Includes Specifications
for the following parts:
Z86C03  Z86C08
Z86E03  Z86E08
Z86C04  Z86C30
Z86E04  Z86E30
Z86C06  Z86C31
Z86E06  Z86E31
Z86C07  Z86C40
Z86E07  Z86E40
Discrete Z8® Microcontrollers

For Peripherals and Consumer Electronic Products

Includes Specifications for the following parts:

- Z86C03
- Z86E03
- Z86C04
- Z86E04
- Z86C06
- Z86E06
- Z86C07
- Z86E07
- Z86C08
- Z86E08
- Z86C30
- Z86E30
- Z86C31
- Z86E31
- Z86C40
- Z86E40

Databook
# Discrete Z8® Microcontrollers

## Table of Contents

<table>
<thead>
<tr>
<th>Title</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Quick Take — A Summary of Parts and Features</strong></td>
<td>A</td>
</tr>
<tr>
<td><strong>Z8® Product Specifications</strong></td>
<td></td>
</tr>
<tr>
<td>Z86C03/C06 CMOS Z8® 8-Bit Consumer Controller Processors</td>
<td>1</td>
</tr>
<tr>
<td>Z86E03/E06 CMOS Z8® 8-Bit OTP Consumer Controller Processors</td>
<td>2</td>
</tr>
<tr>
<td>Z86C04/C08 CMOS Z8® 8-Bit Low Cost 1K/2K ROM Microcontrollers</td>
<td>3</td>
</tr>
<tr>
<td>Z86E04/E08 CMOS Z8® 8-Bit OTP Microcontrollers</td>
<td>4</td>
</tr>
<tr>
<td>Z86C07 CMOS Z8® 8-Bit Microcontroller</td>
<td>5</td>
</tr>
<tr>
<td>Z86E07 CMOS Z8® 8-Bit OTP Microcontroller</td>
<td>6</td>
</tr>
<tr>
<td>Z86C30/C31 CMOS Z8® 8-Bit Consumer Controller Processors</td>
<td>7</td>
</tr>
<tr>
<td>Z86E30/E31 CMOS Z8® 8-Bit OTP Consumer Controller Processors</td>
<td>8</td>
</tr>
<tr>
<td>Z86C40 CMOS Z8® 4K ROM Consumer Controller Processor</td>
<td>9</td>
</tr>
<tr>
<td>Z86E40 CMOS Z8® 8-Bit OTP Consumer Controller Processors</td>
<td>10</td>
</tr>
<tr>
<td><strong>Z8® Microcontrollers Application Notes</strong></td>
<td></td>
</tr>
<tr>
<td>Timekeeping with the Z8®</td>
<td>11-1</td>
</tr>
<tr>
<td>Using the Zilog Z86C06 SPI Bus</td>
<td>11-15</td>
</tr>
<tr>
<td>DTMF Tone Generation Using the Z8® CCP™</td>
<td>11-37</td>
</tr>
<tr>
<td>Serial Communications Using the Z8® CCP™ Software UART</td>
<td>11-45</td>
</tr>
<tr>
<td>The Versatile Z86C08: Three Key Features of this Z8® MCU</td>
<td>11-51</td>
</tr>
<tr>
<td>The Z86C08 Controls a Scrolling LED Message Display</td>
<td>11-59</td>
</tr>
<tr>
<td>Interfacing LCDs to the Z8® Microcontroller</td>
<td>11-75</td>
</tr>
<tr>
<td>On-Chip Oscillator Design</td>
<td>11-89</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

(Continued)

<table>
<thead>
<tr>
<th>SUPPORT PRODUCTS AND THIRD PARTY VENDORS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Z86C0800ZCO EVALUATION BOARD</strong> ..........</td>
</tr>
<tr>
<td><strong>Z86C0800ZDP ADAPTOR Kit</strong> ................</td>
</tr>
<tr>
<td><strong>Z86C1200ZEM EMULATOR</strong> .....................</td>
</tr>
<tr>
<td><strong>Z86E0600ZDP ADAPTOR Kit</strong> ................</td>
</tr>
<tr>
<td><strong>Z86E0700ZDP ADAPTOR Kit</strong> ................</td>
</tr>
<tr>
<td><strong>Z86E3000ZDP ADAPTOR Kit</strong> ................</td>
</tr>
<tr>
<td><strong>Z86E4000ZDF ADAPTOR Kit</strong> .................</td>
</tr>
<tr>
<td><strong>Z86E4000ZDP ADAPTOR Kit</strong> .................</td>
</tr>
<tr>
<td><strong>Z86E4000ZDV ADAPTOR Kit</strong> .................</td>
</tr>
<tr>
<td><strong>Z86E4001ZDF ADAPTOR Kit</strong> .................</td>
</tr>
<tr>
<td><strong>Z86E4001ZDV ADAPTOR Kit</strong> .................</td>
</tr>
<tr>
<td><strong>Z86CCPOOZEM EMULATOR</strong> ...............</td>
</tr>
<tr>
<td><strong>Z86CCPOOZAC EMULATOR KIT</strong> .............</td>
</tr>
<tr>
<td><strong>Z8°S SERIES EMULATORS</strong> ................</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BASE UNITS AND PODS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>THIRD PARTY VENDORS</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SUPERINTEGRATION™ PRODUCTS GUIDE</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZILOG'S SALES OFFICES, REPRESENTATIVES &amp; DISTRIBUTORS</td>
<td>Z</td>
</tr>
<tr>
<td>LITERATURE GUIDE AND ORDERING INFORMATION</td>
<td>L</td>
</tr>
</tbody>
</table>
# Zilog® Microcontrollers

## Quick Take — A Summary of Parts and Features

<table>
<thead>
<tr>
<th>Block Diagram</th>
<th>512 Byte ROM</th>
<th>1K ROM</th>
<th>2K ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>78h CPU</td>
<td>WDT</td>
<td>64 RAM</td>
<td>WDT</td>
</tr>
<tr>
<td>P2</td>
<td>P3</td>
<td>P0</td>
<td>P2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Z86C03</th>
<th>Z86C04/Z86E04</th>
<th>Z86C06</th>
<th>Z86C07/Z86E07</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Consumer Controller Processor (CCP™) with 512 Byte ROM</th>
<th>Z86C04 = 8-Bit Low Cost 1 KByte ROM MCU Z86E04 = OTP Version Consumer Controller Processor (CCP™) with 1 KByte ROM</th>
<th>Z86C07 = 8-Bit 2 KByte ROM MCU Z86E07 = OTP Version</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Process/Speed</th>
<th>CMOS: 8 MHz</th>
<th>CMOS: 8 MHz</th>
<th>CMOS: 12 MHz</th>
<th>CMOS: 8 and 12 MHz</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Features</th>
<th>512 Byte ROM</th>
<th>1 Kbyte ROM</th>
<th>1 Kbyte ROM</th>
<th>2 Kbytes ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-Byte RAM</td>
<td>128-Byte RAM</td>
<td>128-Byte RAM</td>
<td>124-Byte RAM</td>
<td>Two Standby Modes</td>
</tr>
<tr>
<td>Two Standby Modes</td>
<td>Two Standby Modes</td>
<td>Two Standby Modes</td>
<td>Two Standby Modes</td>
<td></td>
</tr>
<tr>
<td>One Counter/Timer</td>
<td>Two Counter/Timer</td>
<td>Two Counter/Timer</td>
<td>Two Counter/Timer</td>
<td></td>
</tr>
<tr>
<td>ROM Protect</td>
<td>ROM Protect</td>
<td>ROM Protect</td>
<td>ROM Protect</td>
<td></td>
</tr>
<tr>
<td>Two Analog Comparator</td>
<td>Two Analog Comparator</td>
<td>Two Analog Comparator</td>
<td>Two Analog Comparator</td>
<td></td>
</tr>
<tr>
<td>Low-Voltage Protection</td>
<td>Low-Voltage Protection (ROM Only)</td>
<td>Low-Voltage Protection (ROM Only)</td>
<td>Low-Voltage Protection (ROM Only)</td>
<td></td>
</tr>
<tr>
<td>14 I/O</td>
<td>14 I/O</td>
<td>14 I/O</td>
<td>14 I/O</td>
<td></td>
</tr>
<tr>
<td>Low-Noise Option</td>
<td>Low-Noise Option</td>
<td>Low-Noise Option</td>
<td>Low Noise Option</td>
<td></td>
</tr>
<tr>
<td>RC Oscillator Option</td>
<td>Serial Peripheral Interface (SPI)</td>
<td>Programmable Interrupt Polarity</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Package</th>
<th>18-Pin DIP 18-Pin SOIC</th>
<th>18-Pin DIP 18-Pin SOIC</th>
<th>18-Pin DIP 18-Pin SOIC</th>
<th>18-Pin DIP 18-Pin SOIC</th>
</tr>
</thead>
</table>

| Support Products | Z86CCP00ZEM - Emulator | Z86CCP00ZCO - Evaluation Board | Z86E000ZDP - Adaptor Kit Z86C000ZDP - Adaptor Kit Z86C1200ZEM - Emulator Z86C1200ZDP - Emulator Z86CCP00ZEM - Emulator Z86CCP00ZAC - Emulator | Z86E000ZDP - Emulator Kit Z86C5000ZEM - Emulator | Z86C5000ZDP - Adaptor Kit Z86CCP00ZEM - Emulator Z86CCP00ZAC - Emulator |
|------------------|-------------------|---------------------|---------------------|---------------------|

Contact Your Local Zilog Sales Office
## Discrete Z8® Microcontrollers

### Quick Take — A Summary of Parts and Features

<table>
<thead>
<tr>
<th>Block Diagram</th>
<th>PART NUMBER</th>
<th>DESCRIPTION</th>
<th>PROCESS/SPEED</th>
<th>FEATURES</th>
<th>PACKAGE</th>
<th>SUPPORT Products</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Z86C08/Z86E08</td>
<td>Z86C08 = Z8® MCU with 2 Kbyte ROM Z86E08 = OTP Version</td>
<td>CMOS: 12 MHz</td>
<td>■ 2 Kbyte ROM ■ 128 Byte RAM ■ Two Standby Modes ■ Two Counter/Timer ■ ROM Protect ■ Two Analog Comparators ■ Auto Power-On Reset ■ Low-Voltage Protection (ROM Only) ■ 14 I/O ■ Low-Noise Option</td>
<td>18-Pin DIP 18-Pin SOIC</td>
<td>Z86C0800ZCO - Evaluation Board Z86C0800ZDP - Adaptor Kit Z86C1200ZEM - Emulator Z86C1200ZDP - Adaptor Kit Z86CCP00ZEM - Emulator Z86CCP00ZAC - Emulator</td>
</tr>
<tr>
<td></td>
<td>Z86C30/Z86E30</td>
<td>Z86C30 = Z8® (CCP®) with 4 Kbyte ROM Z86E30 = OTP Version</td>
<td>CMOS: 12 MHz</td>
<td>■ 4 Kbyte ROM ■ 236 Byte RAM ■ Two Standby Modes ■ Two Counter/Timer ■ ROM Protect ■ Two Analog Comparators ■ Auto Power-On Reset ■ Low-Voltage Protection (ROM Only) ■ 24 I/O ■ RC Oscillator Option ■ Low-Noise Option</td>
<td>28-Pin DIP 28-Pin PCB Chip Carrier</td>
<td>Z86E3000ZDP - Adaptor Kit Z86C3000ZEM - Emulator Z86C3000ZPD - Emulator Pod Z86CCP00ZEM - Emulator Z86CCP00ZAC - Emulator</td>
</tr>
<tr>
<td></td>
<td>Z86C31/Z86E31</td>
<td>Z86C31 = 8-Bit MCU with 2 Kbyte ROM Z86E31 = OTP Version</td>
<td>CMOS: 8 MHz</td>
<td>■ 2 Kbyte ROM ■ 128 Byte RAM ■ Two Standby Modes ■ Two Counter/Timer ■ ROM Protect ■ Two Analog Comparators ■ Auto Power-On Reset ■ Low-Voltage Protection (ROM Only) ■ 24 I/O ■ RC Oscillator Option ■ Low-Noise Option</td>
<td>28-Pin DIP 28-Pin PCB Chip Carrier</td>
<td>Z86E3100ZDP - Adaptor Kit Z86C3100ZEM - Emulator Z86C3100ZPD - Emulator Pod Z86CCP00ZEM - Emulator Z86CCP00ZAC - Emulator</td>
</tr>
<tr>
<td></td>
<td>Z86C40/Z86E40</td>
<td>Z8® Consumer Controller Processor (CCP®) Z86E40 = OTP Version</td>
<td>CMOS: 12 MHz</td>
<td>■ 4K ROM, 236 RAM ■ Two Standby Modes ■ Two Counter/Timers ■ ROM Protect ■ RAM Protect ■ Four Ports ■ Low-Voltage Protection ■ Two Analog Comparators ■ Low-EMI Mode ■ Watch-Dog Timer (WDT) ■ Auto Power-On Reset ■ Low-Power Option</td>
<td>40-Pin DIP 44-Pin PLCC</td>
<td>Z86C4000ZEM - Emulator Z86C5000ZDP - Emulator Pod Z86E4000ZDP - Adaptor Kit Z86E4000ZDV - Adaptor Kit</td>
</tr>
</tbody>
</table>
Z86C03/C06 CMOS Z8® 8-Bit CCP™  
Consumer Controller Processors

Z86E03/E06 CMOS Z8® 8-Bit OTP CCP™  
Consumer Controller Processors

Z86C04/C08 CMOS Z8® Low Cost  
1K /2K ROM Microcontrollers

Z86E04/E08 CMOS Z8®  
8-Bit OTP Microcontrollers

Z86C07 CMOS Z8®  
8-Bit Microcontroller

Z86E07 CMOS Z8®  
8-Bit OTP Microcontroller

Z86C30/C31 CMOS Z8® 8-Bit CCP™  
Consumer Controller Processors
The Zilog's Z8® single-chip microcontroller family with enhanced wake-up circuitry, programmable watch-dog timers and low noise/EMI options. These enhancements result in a more efficient, cost-effective design and provide the user with increased design flexibility over the standard Z8 microcontroller core. With 512 and 1K bytes of ROM and 60 and 124 bytes of general-purpose RAM, respectively, these low cost, low power consumption CMOS microcontrollers offer fast execution, efficient use of memory, sophisticated interrupts, input/output bit manipulation capabilities, and easy hardware/software system expansion.

For applications demanding powerful I/O capabilities, the Z86C03/C06 provides 14 pins dedicated to input and output. These lines are grouped into two ports and are configurable under software control to provide timing, status signals, or parallel I/O.
GENERAL DESCRIPTION (Continued)

Three basic address spaces are available to support this wide range of configurations: Program Memory, Register File, and Expanded Register File. The Register File is composed of 60/124 bytes of General-Purpose Registers, two I/O Port registers, and 13/15 Control and Status registers. The Expanded Register File consists of three control registers in the Z86C03, and four control registers, a SPI Receive Buffer, and a SPI compare register in the Z86C06.

With powerful peripheral features such as on-board comparators, counter/timer(s), Watch-Dog Timer (WDT), and Serial Peripheral Interface (SPI) (C06 only), the Z86C03/ C06 meets the needs of a variety of sophisticated controller applications (Figure 1).

Notes:
All Signals with a preceding front slash, "/", are active Low, e.g.: B/W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

<table>
<thead>
<tr>
<th>Connection</th>
<th>Circuit</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>Vcc</td>
<td>VDD</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
<td>Vss</td>
</tr>
</tbody>
</table>
Table 1. 18-Pin DIP and SOIC Pin Identification

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>P24-27</td>
<td>Port 2, pins 4, 5, 6, 7</td>
<td>In/Output</td>
</tr>
<tr>
<td>5</td>
<td>Vcc</td>
<td>Power Supply</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>XTAL2</td>
<td>Crystal Oscillator Clock</td>
<td>Output</td>
</tr>
<tr>
<td>7</td>
<td>XTAL1</td>
<td>Crystal Oscillator Clock</td>
<td>Input</td>
</tr>
<tr>
<td>8-10</td>
<td>P31-33</td>
<td>Port 3, pins 1, 2, 3</td>
<td>Fixed Input</td>
</tr>
<tr>
<td>11-13</td>
<td>P34-36</td>
<td>Port 3, pins 4, 5, 6</td>
<td>Fixed Output</td>
</tr>
<tr>
<td>14</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>15-18</td>
<td>P20-23</td>
<td>Port 2, pins 0, 1, 2, 3</td>
<td>In/Output</td>
</tr>
</tbody>
</table>
PIN FUNCTIONS

**XTAL1.** Crystal 1 (time-based input). This pin connects a parallel-resonant crystal, ceramic resonator, LC or RC network or an external single-phase clock to the on-chip oscillator input.

**XTAL2.** Crystal 2 (time-based output). This pin connects a parallel-resonant crystal, ceramic resonator, LC or RC network to the on-chip oscillator output.

**Port 2 (P27-P20).** Port 2 is an 8-bit, bi-directional, CMOS compatible I/O port. These eight I/O lines can be configured under software control to be an input or output, independently. Input buffers are Schmitt-triggered and contain Auto Latches. Bits programmed as outputs may be globally programmed as either push-pull or open-drain (Figures 4a, 4b, and 4c). Low EMI output buffers can be globally programmed by the software. In addition, when the SPI is enabled, P20 functions as data-in (DI), and P27 functions as data-out (DO) for the SPI (SPI on the Z86C06 only).

![Port 2 Configuration (Z86C06)](image-url)
Figure 4b. Port 2 Configuration (Z86C06)
Z86C03/C06 CMOS Z8® 8-Bit CCP™
CONSUMER CONTROLLER PROCESSORS

PIN FUNCTIONS (Continued)

Port 2 (I/O)

Figure 4c. Port 2 Configuration (Z86C03)
Auto Latch. The Auto Latch puts valid CMOS levels on all CMOS inputs (except P33, P32, P31) that are not externally driven. Whether this level is 0 or 1 cannot be determined. A valid CMOS level, rather than a floating node, reduces excessive supply current flow in the input buffer.

Port 3 (P36-P31). Port 3 is a 6-bit, CMOS compatible port. These six lines consist of three fixed inputs (P31-P33) and three fixed outputs (P34-P36). Pins P31, P32, and P33 are standard CMOS inputs (no auto latches) and pins P34, P35, and P36 are push-pull outputs. Low EMI output buffers can be globally programmed by the software. Two on-board comparators can process analog signals on P31 and P32 with reference to the voltage on P33. The analog function is enabled by programming Port 3 Mode Register (P3M-bit D1). Pins P31 and P32 are programmable as falling, rising, or both edge triggered interrupts (IRQ register bits 6 and 7). P33 is the comparator reference voltage input when the analog mode is selected. P33 is a falling edge interrupt input only.

**Note:** P33 is available as an interrupt input only in the digital mode. P31 and P32 are valid interrupt inputs and P31 is the T_{IN} input when the analog or digital input mode is selected.

The outputs from the analog comparator can be globally programmed to output from P34 and P35 by setting PCON (F) 00 bit D0 = 1.

Access to Counter/Timer 1 is made through P31 (T_{IN}) and P36 (T_{OUT}).

In the Z86C06, pin P34 can also be configured as SPI clock (SK), input and output, and pin P35 can be configured as Slave Select (SS) in slave mode only, when the SPI is enabled (Figures 5a and 5b).

![Port 3 Configuration](image-url)

**Figure 5a. Port 3 Configuration**
PIN FUNCTIONS (Continued)

Low EMI Emission. The Z86C03/C06 can be programmed to operate in a low EMI emission mode in the PCON register. The oscillator and all I/O ports can be programmed as low EMI emission mode independently. Use of this feature results in:

- The pre-drivers slew rate reduced to 10 ns (typical).
- Low EMI output drivers resistance of 200 ohms (typical).
- Low EMI oscillator.

Comparator Inputs. Port 3 Pin P31 and P32 each have a comparator front end. The comparator reference voltage pin P33 is common to both comparators. In analog mode, the P31 and P32 are the positive inputs to the comparators, and P33 is the reference voltage supplied to both comparators. In digital mode, Pin P33 can be used as a P33 register input or IRQ1 source.

Internal SCLK/TCLK = XTAL operation limited to a maximum of 4 MHz (250 ns cycle time) when the low EMI oscillator is selected and SCLK = External (SMR Register Bit D1=1).

Figure 5b. Port 3 Configuration (Z86C06)
FUNCTIONAL DESCRIPTION

The following special functions have been added to the Z86C03/C06 CCPs to enhance the standard Z8® architecture to provide the user with increased design flexibility.

RESET. The device is reset in one of four ways:

1. Power-On Reset
2. Watch-Dog Timer
3. STOP-Mode Recovery Source
4. Low Voltage Protection

Having the Auto Power-On Reset circuitry built-in, the Z86C03/C06 does not require an external reset circuit. The reset time is 5 ms (typical) plus 18 clock cycles.

The device does not re-initialize the WDTMR, SMR, P2M, or P3M registers to their reset values on a STOP-Mode Recovery operation.

Program Memory. Z86C03/C06 can address up to 512/1K bytes of internal program memory (Figure 6). The first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts. Byte 13 to byte 511/1023 consists of on-chip, user program mask ROM.

ROM Protect. The 512/1K bytes of Program Memory is mask programmable. A ROM protect feature will prevent “dumping” of the ROM contents by inhibiting execution of the LDC and LDCI instructions to program memory in all modes.

ROM protect is mask-programmable. It is selected by the customer when the ROM code is submitted. Selecting ROM protect disables the LDC and LDCI instructions in all modes. ROM look-up tables are not supported in this mode.

Expanded Register File (ERF). The register file has been expanded to allow for additional system control registers and for mapping of additional peripheral devices and input/output ports into the register address area. The Z8 register address space R0 through R15 is implemented as 16 groups of 16 registers per group (Figure 7). These register groups are known as the Expanded Register File (ERF).

Bits 3-0 of the Register Pointer (RP) select the active ERF group. Bits 7-4 of register RP select the working register group (Figure 7). For the Z86C03, three system configuration registers reside in the ERF address space Bank F. For the Z86C06, three system configuration registers reside in the ERF address space Bank F, while three SPI registers reside in Bank C. The rest of the ERF address space is not physically implemented and is open for future expansion.

---

**Figure 6. Program Memory Map**

---

511/1023

<table>
<thead>
<tr>
<th>Location of First Byte of Instruction Executed After RESET</th>
<th>On-Chip ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>IRQ5</td>
</tr>
<tr>
<td>11</td>
<td>IRQ5</td>
</tr>
<tr>
<td>10</td>
<td>IRQ4</td>
</tr>
<tr>
<td>9</td>
<td>IRQ4</td>
</tr>
<tr>
<td>8</td>
<td>IRQ3</td>
</tr>
<tr>
<td>7</td>
<td>IRQ3</td>
</tr>
<tr>
<td>6</td>
<td>IRQ2</td>
</tr>
<tr>
<td>5</td>
<td>IRQ2</td>
</tr>
<tr>
<td>4</td>
<td>IRQ1</td>
</tr>
<tr>
<td>3</td>
<td>IRQ1</td>
</tr>
<tr>
<td>2</td>
<td>IRQ0</td>
</tr>
<tr>
<td>1</td>
<td>IRQ0</td>
</tr>
<tr>
<td>0</td>
<td>IRQ0</td>
</tr>
</tbody>
</table>
FUNCTIONAL DESCRIPTION (Continued)

Z8 STANDARD CONTROL REGISTERS

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>RESET CONDITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF</td>
<td>SPL</td>
</tr>
<tr>
<td>FE</td>
<td>GPR</td>
</tr>
<tr>
<td>FD</td>
<td>RP</td>
</tr>
<tr>
<td>FC</td>
<td>FLAGS</td>
</tr>
<tr>
<td>FB</td>
<td>IMR</td>
</tr>
<tr>
<td>FA</td>
<td>IRQ</td>
</tr>
<tr>
<td>F9</td>
<td>IPR</td>
</tr>
<tr>
<td>F8</td>
<td>P01M</td>
</tr>
<tr>
<td>F7</td>
<td>P3M</td>
</tr>
<tr>
<td>F6</td>
<td>P2M</td>
</tr>
<tr>
<td>F5</td>
<td>Reserved</td>
</tr>
<tr>
<td>F4</td>
<td>Reserved</td>
</tr>
<tr>
<td>F3</td>
<td>PREI</td>
</tr>
<tr>
<td>F2</td>
<td>T1</td>
</tr>
<tr>
<td>F1</td>
<td>TMR</td>
</tr>
<tr>
<td>F0</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

* Will not be reset with a STOP-Mode Recovery

Notes:
General-purpose registers are undefined after power-up, and they are not changed after reset.
* Will not be reset with a STOP-Mode Recovery
U = Unknown
† = Reserved

Figure 7a. Expanded Register File Architecture (Z86C03)
General-purpose registers are undefined after power-up, and they are not changed after reset.

* Will not be reset with a STOP-Mode Recovery.
** Will not be reset with a STOP-Mode Recovery, except D0.
U = Unknown
† = Reserved

Figure 7b. Expanded Register File Architecture (Z86C08)
FUNCTIONAL DESCRIPTION (Continued)

Register File. The Register File consists of two I/O port registers, 60/124 general-purpose registers, and 13/15 control and status registers. The Z86C03 General-Purpose Register file ranges from address 00 to 3F while the Z86C06 General-Purpose Register file ranges from address 00 to 7F (see Figure 9). The instructions can access registers directly or indirectly via an 8-bit address field. This allows a short 4-bit register address using the Register Pointer (Figure 9). In the 4-bit mode, the Register File is divided into 16 working register groups, each occupying 16 continuous locations. The Register Pointer addresses the starting location of the active working-register group.

General-Purpose Registers (GPR). These registers are undefined after the device is powered up. The registers keep their last value after any reset, as long as the reset occurs in the $V_{cc}$ voltage-specified operating range. Note: Register R254 has been designated as a general-purpose register.

Stack. An 8-bit Stack Pointer (R255) used for the internal stack that resides within the 60/124 general-purpose registers.

Figure 8. Register Pointer Register

Figure 9. Register Pointer
Counter/Timers. There are two 8-bit programmable counter/timers (T0-T1), each driven by its own 6-bit programmable prescaler (Z86C03 only has T1). The T1 prescaler can be driven by internal or external clock sources, however, the T0 prescaler is driven by the internal clock only (Figure 10).

Figure 10. Counter/Timer Block Diagram
The 6-bit prescalers divide the input frequency of the clock source by any integer number from 1 to 64. Each prescaler drives its counter, which decrements the value (1 to 256) that has been loaded into the counter. When the counter reaches the end of count, a timer interrupt request, IRQ4 (T0) or IRQ5 (T1), is generated. Note that IRQ4 is software-generated in the Z86C03.

The counters are programmed to start, stop, restart to continue, or restart from the initial value. The counters can also be programmed to stop upon reaching zero (single-pass mode) or to automatically reload the initial value and continue counting (modulo-n continuous mode).

The counters, but not the prescalers, are read at any time without disturbing their value or count mode. The clock source for T1 is user-definable and can be either the internal microprocessor clock divided by four, or an external signal input via Port 3. The Timer Mode register configures the external timer input (P31) as an external clock, a trigger input that can be retriggerable or non-retriggerable, or as a gate input for the internal clock. Port 3, line P36 serves as a timer output (T0out) through which T0 (C06 only), T1, or the internal clock can be output. The counter/timers can be cascaded by connecting the T0 output to the input of T1 (C06 only). The T1n mode is enabled by setting PRE1 bit D1 (R243) to 0.

Interrupts. The Z86C03/C06 has six different interrupts from six different sources. The interrupts are maskable and prioritized (Figure 11). The six sources are divided as follows; three sources are claimed by Port 3 lines P31-P33, two sources in the counter/timers, and one source for the SPI. The Interrupt Mask Register globally or singularly enables or disables the six interrupt requests (Table 2).

![Figure 11. Interrupt Block Diagram](image-url)
Table 2. Interrupt Types, Sources, and Vectors

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Vector Location</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRQ 0</td>
<td>IRQ 0</td>
<td>0, 1</td>
<td>External (P32), Rising/Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ 1</td>
<td>IRQ 1</td>
<td>2, 3</td>
<td>External (P33), Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ 2</td>
<td>IRQ 2, T IN</td>
<td>4, 5</td>
<td>External (P31), Rising/Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ 3*</td>
<td>IRQ 3</td>
<td>6, 7</td>
<td>Software Generated, SPI Receive</td>
</tr>
<tr>
<td>IRQ 4</td>
<td>T0/IRQ 4</td>
<td>8, 9</td>
<td>Internal for C06 and Software Generated for C03</td>
</tr>
<tr>
<td>IRQ 5</td>
<td>TI</td>
<td>10, 11</td>
<td>Internal</td>
</tr>
</tbody>
</table>

Note:
* In the Z86C06, the SPI receive interrupt is mapped to IRQ3 when enabled.

When more than one interrupt is pending, priorities are resolved by a programmable priority encoder that is controlled by the Interrupt Priority register. An interrupt machine cycle is activated when an interrupt request is granted. This disables all subsequent interrupts, saves the Program Counter and Status Flags, and then branches to the program memory vector location reserved for that interrupt. All Z86C03/C06 interrupts are vectored through locations in the program memory. This memory location and the next byte contain the 16-bit starting address of the interrupt service routine for that particular interrupt request.

To accommodate polled interrupt systems, interrupt inputs are masked and the interrupt request register is polled to determine which of the interrupt requests need service. In the Z86C06, when the SPI is disabled, IRQ3 has no hardware source but can be invoked by software (write to IRQ3 Register). When the SPI is enabled, an interrupt will be mapped to IRQ3 after a byte of data has been received by the SPI Shift Register.

An interrupt resulting from AN1 is mapped into IRQ2, and an interrupt from AN2 is mapped into IRQ0. Interrupts IRQ2 and IRQ0 may be rising, falling, or both edge triggered, and are programmable by the user. The software can poll to identify the state of the pin.

The programming bits for the INTERRUPT EDGE SELECT are located in the IRQ register (R250), bits D7 and D6. The configuration is shown in Table 3.

Table 3. IRQ Register

<table>
<thead>
<tr>
<th>IRQ</th>
<th>Interrupt Edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>D7</td>
<td>D6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Notes:
F = Falling Edge
R = Rising Edge
Clock. The Z86C03/C06 on-chip oscillator has a high-gain, parallel-resonant amplifier for connection to a crystal, RC, ceramic resonator, or any suitable external clock source (XTAL1 = Input, XTAL2 = Output). The crystal should be AT cut, 10 kHz to 8 MHz/12 MHz max, with a series resistance (RS) less than or equal to 100 Ohms. The crystal should be connected across XTAL1 and XTAL2 using the vendor's recommended capacitor values (capacitance between 10 pF to 300 pF) from each pin directly to the device ground (pin 14). The layout is important to reduce ground noise injection.

The RC oscillator option is mask-programmable, to be selected by the customer at the time ROM code is submitted. The RC oscillator configuration must be an external resistor connected from XTAL1 to XTAL2, with a frequency-setting capacitor from XTAL1 to ground (Figure 12). The RC value vs. Frequency curves are shown in Figures 57 and 58.

In addition, a special feature has been incorporated into the Z86C03/C06; in low EMI noise mode (bit 7 of PCON register=0) with the RC option selected, the oscillator is targeted to consume considerately less ICC current at frequencies of 10 kHz or less.

Power-On Reset. A timer circuit clocked by a dedicated on-board RC oscillator is used for the Power-On Reset (POR) timer function. The POR time allows Vcc and the oscillator circuit to stabilize before instruction execution begins. The POR timer circuit is a one-shot timer triggered by one of the three conditions:

- Power-Fail to Power-OK Status
- STOP-Mode Recovery (If D5 of SMR=1)
- WDT Timeout

The POR time is a nominal 5 ms. Bit 5 of the STOP Mode Register determines whether the POR timer is bypassed after STOP-Mode Recovery (typical for external clock, and RC/LC oscillators with fast start up time).

HALT. A Halt instructions will turn off the internal CPU clock but not the XTAL oscillation. The counter/timers and external interrupts IRQ0, IRQ1, and IRQ2 remain active. The device may be recovered by interrupts either externally or internally generated.
STOP. This instruction turns off the internal clock and external crystal oscillation and reduces the standby current. The STOP mode is terminated by a RESET only, either by WDT timeout, POR, SPI compare; or SMR recovery. This causes the processor to restart the application program at address 0000C (HEX). Note, the crystal remains active in STOP mode if bits 3 and 4 of the WDTMR are enabled. In this mode, only the Watch-Dog Timer runs in STOP mode.

In order to enter STOP (or HALT) mode, it is necessary to first flush the instruction pipeline to avoid suspending execution in mid-instruction. To do this, the user executes a NOP (opcode=FFH) immediately before the appropriate sleep instruction, i.e.:

```
FF  NOP    ; clear the pipeline
6F  STOP   ; enter STOP mode
or
FF  NOP    ; clear the pipeline
7F  HALT   ; enter HALT mode
```

Serial Peripheral Interface (SPI)—Z86C06 Only. The Z86C06 incorporates a serial peripheral interface for communication with other microcontrollers and peripherals. The SPI does not exist on the Z86C03. The SPI includes features such as STOP-Mode Recovery, Master/Slave selection, and Compare mode. Table 4 contains the pin configuration for the SPI feature when it is enabled. The SPI consists of four registers: SPI Control Register (SCON), SPI Compare Register (SCOMP), SPI Receive/Buffer Register (RxBUF), and SPI Shift Register. SCON is located in bank (C) of the Expanded Register Group at address 02.

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
<th>Pin Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>DI</td>
<td>Data-In</td>
<td>P20</td>
</tr>
<tr>
<td>DO</td>
<td>Data-Out</td>
<td>P27</td>
</tr>
<tr>
<td>SS</td>
<td>Slave Select</td>
<td>P35</td>
</tr>
<tr>
<td>SK</td>
<td>SPI Clock</td>
<td>P34</td>
</tr>
</tbody>
</table>

Table 4. Z86C06 SPI Pin Configuration

The SPI Control Register (SCON) (Figure 13) is a read/write register that controls; Master/Slave selection, interrupts, clock source and phase selection, and error flag. Bit 0 enables/disables the SPI with the default being SPI disabled. A 1 in this location will enable the SPI, and a 0 will disable the SPI. Bits 1 and 2 of the SCON register in Master mode select the clock rate. The user may choose whether internal clock is divide-by-2, -4, -8, or -16. In slave mode, Bit 1 of this register flags the user if an overrun of the RxBUF Register has occurred. The RxCharOverrun flag is only reset by writing a 0 to this bit. In slave mode, bit 2 of the Control Register disables the data-out I/O function. If a 1 is written to this bit, the data-out pin is released to its original port configuration. If a 0 is written to this bit, the SPI shifts out one bit for each bit received. Bit 3 of the SCON Register enables the compare feature of the SPI, with the default being disabled. When the compare feature is enabled, a comparison of the value in the SCOMP Register is made with the value in the RxBUF Register. Bit 4 signals that a receive character is available in the RxBUF Register.

If the associated IRQ3 is enabled, an interrupt is generated. Bit 5 controls the clock phase of the SPI. A 1 in Bit 5 allows for receiving data on the clock's falling edge and transmitting data on the clock's rising edge. A 0 allows receiving data on the clock's rising edge and transmitting on the clock's falling edge. The SPI clock source is defined in bit 6. A 1 uses Timer0 output for the SPI clock, and a 0 uses TCLK for clocking the SPI. Finally, bit 7 determines whether the SPI is used as a Master or a Slave. A 1 puts the SPI into Master mode and a 0 puts the SPI into Slave mode.
FUNCTIONAL DESCRIPTION (Continued)

SPI Operation (Z86C06 only). The SPI is used in one of two modes: either as system slave, or as system master. Several of the possible system configurations are shown in Figure 14. In the slave mode, data transfer starts when the slave select (SS) pin goes active. Data is transferred into the slave's SPI Shift Register through the DI pin, which has the same address as the RxBuf Register. After a byte of data has been received by the SPI Shift Register, a Receive Character Available (RCA/IRQ3) flag and interrupt is generated. The next byte of data will be received at this time. The RxBuf Register must be cleared, or a Receive Character Overrun (RxCharOverrun) flag will be set in the SCON Register, and the data in the RxBuf Register will be overwritten. When the communication between the master and slave is complete, the SS goes inactive.

Unless disconnected, for every bit that is transferred into the slave through the DI pin, a bit is transferred out through the DO pin on the opposite clock edge. During slave operation, the SPI clock pin (SK) is an input. In master mode, the CPU must first activate a SS through one of its I/O ports. Next, data is transferred through the master's DO pin one bit per master clock cycle. Loading data into the shift register initiates the transfer. In master mode, the master's clock will drive the slave's clock. At the conclusion of a transfer, a Receive Character Available (RCA/IRQ3) flag and interrupt is generated. Before data is transferred via the DO pin, the SPI Enable bit in the SCON Register must be enabled.

SPI Compare (Z86C06 only). When the SPI Compare Enable bit, D3 of the SCON Register is set to 1, the SPI Compare feature is enabled. The compare feature is only valid for slave mode. A compare transaction begins when the (SS) line goes active. Data is received as if it were a normal transaction, but there is no data transmitted to avoid bus contention with other slave devices. When the compare byte is received, IRQ3 is not generated. Instead, the data is compared with the contents of the SCOMP Register. If the data does not match, DO remains inactive and the slave ignores all data until the (SS) signal is reset. If the data received matches the data in the SCOMP register, then a SMR signal is generated. DO is activated if it is not tri-stated by D2 in the SCON Register, and data is received the same as any other SPI slave transaction.

When the SPI is activated as a slave, it operates in all system modes; STOP, HALT, and RUN. Slaves' not comparing remain in their current mode, whereas slaves' comparing wake from a STOP or HALT mode by means of an SMR.

SPI Clock (Z86C06 only). The SPI clock maybe driven by three sources: Timer0, a division of the internal system clock, or the external master when in slave mode. Bit D6 of the SCON Register controls what source drives the SPI clock. A 0 in bit D6 of the SCON Register determines the division of the internal system clock if this is used as the SPI clock source. Divide by 2, 4, 8, or 16 is chosen as the scaler.
Figure 14. SPI System Configuration (Z86C06 Only)
Receive Character Available and Overrun (Z86C06 Only). When a complete data stream is received, an interrupt is generated and the RxCharAvail bit in the SCON Register is set. Bit 4 in the SCON Register is for enabling or disabling the RxCharAvail interrupt. The RxCharAvail bit is available for interrupt polling purposes and is reset when the RxBUF Register is read. RxCharAvail is generated in both master and slave modes. While in slave mode, if the RxBUF is not read before the next data stream is received and loaded into the RxBUF Register, Receive Character Overrun (RxCharOverrun) occurs. Since there is no need for clock control in slave mode, bit D1 in the SPI Control Register is used to log any RxCharOverrun (Figure 15 and Figure 16).

<table>
<thead>
<tr>
<th>No</th>
<th>Parameter</th>
<th>Min</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DI to SK Setup</td>
<td>10</td>
<td>ns</td>
</tr>
<tr>
<td>2</td>
<td>SK to DO Valid</td>
<td>15</td>
<td>ns</td>
</tr>
<tr>
<td>3</td>
<td>SS to SK Setup</td>
<td>.5 Tsk</td>
<td>ns</td>
</tr>
<tr>
<td>4</td>
<td>SS to DO Valid</td>
<td>15</td>
<td>ns</td>
</tr>
<tr>
<td>5</td>
<td>SK to DI Hold Time</td>
<td>10</td>
<td>ns</td>
</tr>
</tbody>
</table>

Figure 15. SPI Timing (Z86C06 Only)
PORT Configuration Register (PCON). The PCON configures the ports individually for comparator output on Port 3, low EMI noise on Ports 2 and 3, and low EMI noise oscillator. The PCON Register is located in the Expanded Register File at bank F, location 00 (Figure 17).

Comparator Output Port 3 (D0). Bit 0 controls the comparator used in Port 3. A 1 in this location brings the comparator outputs to P34, and P35 and a 0 releases the Port to its standard I/O configuration.

Bits D4-D1. These bits are reserved and must be 1.

Low EMI Port 2 (D5). Port 2 is configured as a Low EMI Port by resetting this bit (D5=0) or configured as a Standard Port by setting D5=1. The default value is 1.

Low EMI Port 3 (D6). Port 3 is configured as a Low EMI Port by resetting this bit (D6=0) or configured as a Standard Port by setting D6=1. The default value is 1.

Low EMI OSC (D7). This bit of the PCON Register controls the low EMI noise oscillator. A 1 in this location configures the oscillator with standard drive. While a 0 configures the oscillator with low noise drive, it does not affect the relationship of SCLK and XTAL.

Figure 16. SPI Logic (Z86C06 Only)

Figure 17. Port Configuration Register (PCON) (Write Only)
FUNCTIONAL DESCRIPTION (Continued)

STOP-Mode Recovery Register (SMR). This register selects the clock divide value and determines the mode of STOP-Mode Recovery (Figure 18). All bits are Write Only except bit 7, which is Read Only. Bit 7 is a flag bit that is hardware set on the condition of a STOP recovery and reset on a power-on cycle. Bit 6 controls whether a low level or high level is required from the recovery source. The recovery level must be active Low to work with SPI. Bit 5 controls the reset delay after recovery. Bits 2, 3, and 4 of the SMR specify the source of the STOP-Mode Recovery signal. Bit 1 determines whether the XTAL is divided by 1 or 2. A 0 in this location uses XTAL divide-by-two, and a 1 uses XTAL. The default for this bit is XTAL divide-by-two. Bit 0 controls the divide-by-16 prescaler of SCLK/TCLK. The SMR is located in bank F of the Expanded Register Group at address 0BH.

<table>
<thead>
<tr>
<th>SMR (F) 0B</th>
</tr>
</thead>
<tbody>
<tr>
<td>D7 D6 D5 D4 D3 D2 D1 D0</td>
</tr>
<tr>
<td>Reserved (Must be 0)</td>
</tr>
<tr>
<td>External Clock Divide Mode</td>
</tr>
<tr>
<td>0 = SCLK/TCLK = External/2</td>
</tr>
<tr>
<td>1 = SCLK/TCLK = External/1</td>
</tr>
<tr>
<td>Stop Mode Recovery Source</td>
</tr>
<tr>
<td>000 POR Only*</td>
</tr>
<tr>
<td>001-100 Reserved</td>
</tr>
<tr>
<td>101 P27</td>
</tr>
<tr>
<td>110 P2 NOR 0:3</td>
</tr>
<tr>
<td>111 P2 NOR 0:7</td>
</tr>
<tr>
<td>Stop Delay</td>
</tr>
<tr>
<td>0 OFF</td>
</tr>
<tr>
<td>1 ON**</td>
</tr>
<tr>
<td>Stop Recovery Level</td>
</tr>
<tr>
<td>0 Low Level*</td>
</tr>
<tr>
<td>1 High Level</td>
</tr>
<tr>
<td>Stop Flag</td>
</tr>
<tr>
<td>0 POR*</td>
</tr>
<tr>
<td>1 Stop Recovery</td>
</tr>
</tbody>
</table>

* Default setting after RESET
** Default setting after RESET and STOP-Mode Recovery.

Figure 18a. STOP-Mode Recovery Register (Write Only except bit D7, which is Read Only.) (Z86C03)

SCLK/TCLK Divide-by-16 Select (D0)—Z86C06 Only.
D0 of the SMR controls a divide-by-16 prescaler of SCLK/TCLK. The purpose of this control is to selectively reduce device power consumption during normal processor execution (SCLK control) and/or HALT mode (where TCLK sources the counter/timers and interrupt logic).

External Clock Divide Mode (D1). This bit can eliminate the oscillator divide-by-two circuitry. When this bit is 0, SCLK (System Clock) and TCLK (Timer Clock) are equal to the external clock frequency divided by two. The SCLK/TCLK is equal to the external clock frequency when this bit is set (D1=1). Using this bit, together with D7 of PCON, helps further lower EMI [i.e., D7 (PCON)=0, D1 (SMR=1). The default setting is 0.

Notes:
* Default setting after RESET.
** Default setting after RESET and STOP-Mode Recovery.
STOP-Mode Recovery Source (D2,D3,D4). These three bits of the SMR specify the wake-up source of the STOP-Mode Recovery (Figure 19 and Table 5).

### Table 5. STOP-Mode Recovery Source

<table>
<thead>
<tr>
<th>SMR</th>
<th>Operation Description of Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>POR recovery only</td>
</tr>
<tr>
<td>0 0 1</td>
<td>POR recovery only (C03 = Reserved)</td>
</tr>
<tr>
<td>0 1 0</td>
<td>P31 transition (C03 = Reserved)</td>
</tr>
<tr>
<td>0 1 1</td>
<td>P32 transition (C03 = Reserved)</td>
</tr>
<tr>
<td>1 0 0</td>
<td>P33 transition (C03 = Reserved)</td>
</tr>
<tr>
<td>1 0 1</td>
<td>P27 transition</td>
</tr>
<tr>
<td>1 1 0</td>
<td>Logical NOR of Port 2 bits 0:3</td>
</tr>
<tr>
<td>1 1 1</td>
<td>Logical NOR of Port 2 bits 0:7</td>
</tr>
</tbody>
</table>

P31-P33 cannot wake up from STOP Mode if the input lines are configured as analog inputs. In the Z86C06, when the SPI is enabled and the Compare feature is active, a SMR is generated upon a comparison in the SPI Shift Register and SCOMP Register, regardless of the above SMR Register settings. If SPI Compare is used to wake up the part from STOP Mode, it is still possible to have one of the other STOP-Mode Recovery sources active. **Note:** These other STOP-Mode Recovery sources have to be active level Low (bit D6 in SMR set to 0 if P31, P32, P33, and P27 selected, or bit D6 in SMR set to 1 if logical NOR of Port 2 is selected).

STOP-Mode Recovery Delay Select (D5). This bit disables the 5 ms RESET delay after STOP-Mode Recovery. The default condition of this bit is 1. If the 'fast' wake up is selected, the STOP-Mode Recovery source needs to be kept active for at least 5 TpC.

STOP-Mode Recovery Level Select (D6). A 1 in this bit position indicates that a high level on any one of the recovery sources wakes the device from STOP Mode. A 0 indicates low level recovery. The default is 0 on POR (Figure 19).

Cold or Warm Start (D7). This bit is set by the device upon entering STOP Mode. It is active High, and is 0 (cold) on POR/WDT RESET. This bit is Read Only. A 1 in this bit (warm) indicates that the device awakens by a SMR source.

---

**Figure 19a. STOP-Mode Recovery Source (Z86C03)**
FUNCTIONAL DESCRIPTION (Continued)

Watch-Dog Timer Mode Register (WDTMR). The WDT is a retriggerable one-shot timer that resets the Z8 if it reaches its terminal count. The WDT is initially enabled by executing the WDT instruction and refreshed on subsequent executions of the WDT instruction. The WDT cannot be disabled after it has been initially enabled. The WDT circuit is driven by an on-board RC oscillator or external oscillator from XTAL1 pin. The POR clock source is selected with bit 4 of the WDTMR register.

Note: Execution of the WDT instruction affects the Z (zero), S (sign), and V (overflow) flags.

Bits 0 and 1 control a tap circuit that determines the timeout period (on Z86C06 only). Bit 2 determines whether the WDT is active during HALT and bit 3 determines WDT activity during STOP. If bits 3 and 4 of this register are both set to 1, the WDT is only driven by the external clock during STOP mode. This feature makes it possible to wake up from STOP mode from an internal source. Bits 5 through 7 of the WDTMR are reserved (Figure 20). Note: This register is accessible only during the first 64 processor cycles (128 XTAL clocks) from the execution of the first instruction after Power-On Reset, Watch-Dog Reset or a STOP Mode Recovery (Figure 21). After this point, the register cannot be modified by any means, intentional or otherwise. The WDTMR cannot be read and is located in bank F of the Expanded Register Group at address location 0FH.

Figure 19b. STOP-Mode Recovery Source (Z86C06)

Figure 20. Watch-Dog Timer Mode Register (Write Only)
Figure 21. Resets and WDT

* Not available on the Z86C03, WDT fixed at 15 ms/1024τC in the Z86C03.
FUNCTIONAL DESCRIPTION (Continued)

WDT Time Select (D1, D0). Bits 0 and 1 control a tap circuit that determines the time-out period. Table 6 shows the different values that can be obtained. The default value of D0 and D1 are 1 and 0, respectively. These select bits are present in the Z86C06 only.

Table 6. Time-Out Period of the WDT (Z86C06 Only)

<table>
<thead>
<tr>
<th>D1</th>
<th>D0</th>
<th>Time-Out of Internal RC OSC</th>
<th>Time-Out of XTAL Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>5 ms min</td>
<td>256TpC</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>15 ms min</td>
<td>512TpC</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>25 ms min</td>
<td>1024TpC</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>100 ms min</td>
<td>4096TpC</td>
</tr>
</tbody>
</table>

Notes:
TpC = XTAL clock cycle
The default on reset is 15 ms, D0 = 1 and D1 = 0.
See Figures 53 to 56 for details.
The values given are for $V_{cc} = 5.0V$.
For the Z86C03, the WDT time-out value is fixed at 1024 TpC (depending on WDTMR bit D4) period. When writing to the WDTMR in the Z86C03, bit D0 must be 1 and D1 must be 0.

WDT During HALT (D2). This bit determines whether or not the WDT is active during HALT mode. A 1 indicates active during HALT. The default is 1.

WDT During STOP (D3). This bit determines whether or not the WDT is active during STOP mode. Since XTAL clock is stopped during STOP Mode, unless as specified below, the on-board RC must be selected as the clock source to the POR counter. A 1 indicates active during STOP. The default is 1. If bits D3 and D4 are both set to 1, the WDT only, is driven by the external clock during STOP mode.

Clock Source for WDT (D4). This bit determines which oscillator source is used to clock the internal POR and WDT counter chain. If the bit is a 1, the internal RC oscillator is bypassed and the POR and WDT clock source is driven from the external pin, XTAL1. The default configuration of this bit is 0, which selects the internal RC oscillator.

Bits 5, 6 and 7. These bits are reserved.

$V_{cc}$ Voltage Comparator. An on-board Voltage Comparator checks that $V_{cc}$ is at the required level to ensure correct operation of the device. Reset is globally driven if $V_{cc}$ is below the specified voltage (typically 2.1V).
Low Voltage Protection ($V_{LV}$). The Low Voltage Protection trip point ($V_{LV}$) will be less than 3 volts and above 1.4 volts under the following conditions.

Maximum ($V_{LV}$) Conditions:

**Case 1:** $T_A = -40^\circ$ to $+105^\circ$C, Internal Clock (SCLK) Frequency equal or less than 1 MHz

**Case 2:** $T_A = -40^\circ$ to $+85^\circ$C, Internal Clock (SCLK) Frequency equal or less than 2 MHz

**Note:** The internal clock frequency (SCLK) is determined by SMR (F) 0BH bit D1.

The device functions normally at or above 3.0V under all conditions. Below 3.0V, the device is guaranteed to function normally until the Low Voltage Protection trip point ($V_{LV}$) is reached, for the temperatures and operating frequencies in cases 1 and 2 above. The actual low voltage trip point is a function of temperature and process parameters (Figure 22).

![Figure 22. Typical Z86C03/C06 $V_{LV}$ Voltage vs Temperature](image-url)
ABSOLUTE MAXIMUM RATINGS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>V_{CC}</td>
<td>Supply Voltage*</td>
<td>-0.3</td>
<td>+7.0</td>
<td>V</td>
</tr>
<tr>
<td>V_{PM}</td>
<td>Max Input Voltage**</td>
<td>12</td>
<td>12</td>
<td>V</td>
</tr>
<tr>
<td>T_{STG}</td>
<td>Storage Temp</td>
<td>-65</td>
<td>+150</td>
<td>°C</td>
</tr>
<tr>
<td>T_{A}</td>
<td>Oper Ambient Temp</td>
<td>†</td>
<td>†</td>
<td>°C</td>
</tr>
</tbody>
</table>

Notes:
* Voltage on all pins with respect to GND.
** Applies to Port pins only and must limit current going into or out of Port pins to 250 μA maximum.
† See Ordering Information

STANDARD TEST CONDITIONS

The characteristics listed below apply for standard test conditions as noted. All voltages are referenced to Ground. Positive current flows into the referenced pin (Figure 23).

CAPACITANCE

\[ T_{A} = 25^\circ C, \ V_{CC} = \text{GND} = 0V, \ f = 1.0 \ \text{MHz}, \ \text{unmeasured pins returned to GND}. \]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Capacitance</td>
<td>0</td>
<td>12 pF</td>
</tr>
<tr>
<td>Output Capacitance</td>
<td>0</td>
<td>20 pF</td>
</tr>
<tr>
<td>I/O Capacitance</td>
<td>0</td>
<td>25 pF</td>
</tr>
</tbody>
</table>

V_{CC} SPECIFICATION

\[ V_{CC} = 3.0V \ \text{to}\ 5.5V \]
### DC ELECTRICAL CHARACTERISTICS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$</th>
<th>$T_a = 0^\circ C$ to $+70^\circ C$</th>
<th>$T_a = -40^\circ C$ to $+105^\circ C$</th>
<th>Typical @ 25°C</th>
<th>Units</th>
<th>Conditions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{CH}$</td>
<td>Clock Input High Voltage</td>
<td>3.0V</td>
<td>0.9 $V_{cc}$ $V_{cc} + 0.3$</td>
<td>0.9 $V_{cc}$ $V_{cc} + 0.3$</td>
<td>2.4</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.9 $V_{cc}$ $V_{cc} + 0.3$</td>
<td>0.9 $V_{cc}$ $V_{cc} + 0.3$</td>
<td>3.9</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
<td></td>
</tr>
<tr>
<td>$V_{CL}$</td>
<td>Clock Input Low Voltage</td>
<td>3.0V</td>
<td>$V_{ss} - 0.3$ 0.2 $V_{cc}$</td>
<td>$V_{ss} - 0.3$ 0.2 $V_{cc}$</td>
<td>1.6</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{ss} - 0.3$ 0.2 $V_{cc}$</td>
<td>$V_{ss} - 0.3$ 0.2 $V_{cc}$</td>
<td>2.7</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
<td></td>
</tr>
<tr>
<td>$V_{IH}$</td>
<td>Input High Voltage</td>
<td>3.0V</td>
<td>0.7 $V_{cc}$ $V_{cc} + 0.3$</td>
<td>0.7 $V_{cc}$ $V_{cc} + 0.3$</td>
<td>1.8</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.7 $V_{cc}$ $V_{cc} + 0.3$</td>
<td>0.7 $V_{cc}$ $V_{cc} + 0.3$</td>
<td>2.8</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{IL}$</td>
<td>Input Low Voltage</td>
<td>3.0V</td>
<td>$V_{ss} - 0.3$ 0.2 $V_{cc}$</td>
<td>$V_{ss} - 0.3$ 0.2 $V_{cc}$</td>
<td>1.0</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{ss} - 0.3$ 0.2 $V_{cc}$</td>
<td>$V_{ss} - 0.3$ 0.2 $V_{cc}$</td>
<td>1.5</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{OH}$</td>
<td>Output High Voltage</td>
<td>3.0V</td>
<td>$V_{cc} - 0.4$ $V_{cc} - 0.4$</td>
<td>$V_{cc} - 0.4$ $V_{cc} - 0.4$</td>
<td>3.1</td>
<td>V</td>
<td>$I_{oh} = -2.0 mA$ [10]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{cc} - 0.4$ $V_{cc} - 0.4$</td>
<td>$V_{cc} - 0.4$ $V_{cc} - 0.4$</td>
<td>4.8</td>
<td>V</td>
<td>$I_{oh} = -2.0 mA$ [10]</td>
<td></td>
</tr>
<tr>
<td>$V_{OL1}$</td>
<td>Output Low Voltage</td>
<td>3.0V</td>
<td>0.8</td>
<td>0.8</td>
<td>0.2</td>
<td>V</td>
<td>$I_{ol} = +4.0 mA$ [10]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.4</td>
<td>0.4</td>
<td>0.1</td>
<td>V</td>
<td>$I_{ol} = +4.0 mA$ [10]</td>
<td></td>
</tr>
<tr>
<td>$V_{OL2}$</td>
<td>Output Low Voltage</td>
<td>3.0V</td>
<td>1.0</td>
<td>1.0</td>
<td>0.4</td>
<td>V</td>
<td>$I_{ol} = +6 mA$, 3 Pin Max</td>
<td>[10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>1.0</td>
<td>1.0</td>
<td>0.5</td>
<td>V</td>
<td>$I_{ol} = +12 mA$, 3 Pin Max</td>
<td>[10]</td>
</tr>
<tr>
<td>$V_{OFFSET}$</td>
<td>Comparator Input Offset Voltage</td>
<td>3.0V</td>
<td>25</td>
<td>25</td>
<td>10</td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
<td>10</td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{ICR}$</td>
<td>Input Common Mode Voltage Range</td>
<td>3.0V</td>
<td>$OV$ $V_{cc} - 1.0V$</td>
<td>$OV$ $V_{cc} - 1.5V$</td>
<td></td>
<td></td>
<td></td>
<td>[7]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$OV$ $V_{cc} - 1.0V$</td>
<td>$OV$ $V_{cc} - 1.5V$</td>
<td></td>
<td></td>
<td></td>
<td>[7]</td>
</tr>
<tr>
<td>$I_{IL}$</td>
<td>Input Leakage (Input bias current of comparator)</td>
<td>3.0V</td>
<td>$-1.0$ 1.0</td>
<td>$-1.0$ 1.0</td>
<td>$\mu A$</td>
<td>$V_{in} = OV$, $V_{cc}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$-1.0$ 1.0</td>
<td>$-1.0$ 1.0</td>
<td>$\mu A$</td>
<td>$V_{in} = OV$, $V_{cc}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{OL}$</td>
<td>Output Leakage</td>
<td>3.0V</td>
<td>$-1.0$ 1.0</td>
<td>$-1.0$ 1.0</td>
<td>$\mu A$</td>
<td>$V_{in} = OV$, $V_{cc}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$-1.0$ 1.0</td>
<td>$-1.0$ 1.0</td>
<td>$\mu A$</td>
<td>$V_{in} = OV$, $V_{cc}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{CC}$</td>
<td>Supply Current</td>
<td>3.2V</td>
<td></td>
<td></td>
<td>8.0</td>
<td>$\mu A$</td>
<td>$@ 32 kHz$</td>
<td>[13]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>6</td>
<td>6</td>
<td>3.0</td>
<td>mA</td>
<td>$@ 8 MHz$</td>
<td>[4,5,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>11.0</td>
<td>11.0</td>
<td>6.0</td>
<td>mA</td>
<td>$@ 8 MHz$</td>
<td>[4,5,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>8.0</td>
<td>8.0</td>
<td>4.5</td>
<td>mA</td>
<td>$@ 12 MHz$</td>
<td>[4,5,10,11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>15</td>
<td>15</td>
<td>9.0</td>
<td>mA</td>
<td>$@ 12 MHz$</td>
<td>[4,5,10,11]</td>
</tr>
</tbody>
</table>
## DC ELECTRICAL CHARACTERISTICS

### Z86C03/C06

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Note [3]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$V_{cc}$</th>
<th>$T_a = 0^\circ C$ to $+70^\circ C$</th>
<th>$T_a = -40^\circ C$ to $+105^\circ C$</th>
<th>Typical @ $25^\circ C$</th>
<th>Units</th>
<th>Conditions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{CC1}$ Standby Current</td>
<td>3.0V</td>
<td>Min 3.0 Max 1.3 mA</td>
<td>HALT Mode $V_{in} = 0V$, $V_{cc} @$8 MHz</td>
<td>1.0</td>
<td>mA</td>
<td>[4, 7, 10]</td>
</tr>
<tr>
<td></td>
<td>5.5V</td>
<td>Min 5 Max 3.0 mA</td>
<td>HALT Mode $V_{in} = 0V$, $V_{cc} @$8 MHz</td>
<td>1.0</td>
<td>mA</td>
<td>[4, 7, 10]</td>
</tr>
<tr>
<td></td>
<td>3.0V</td>
<td>Min 4.5 Max 2.0 mA</td>
<td>HALT Mode $V_{in} = 0V$, $V_{cc} @$12 MHz</td>
<td>1.0</td>
<td>mA</td>
<td>[4, 7, 10, 11]</td>
</tr>
<tr>
<td></td>
<td>5.5V</td>
<td>Min 7.0 Max 4.0 mA</td>
<td>HALT Mode $V_{in} = 0V$, $V_{cc} @$12 MHz</td>
<td>1.0</td>
<td>mA</td>
<td>[4, 7, 10, 11]</td>
</tr>
<tr>
<td></td>
<td>3.0V</td>
<td>Min 1.4 Max 0.7 mA</td>
<td>Clock Divide-by-16</td>
<td>1.0</td>
<td>mA</td>
<td>[4, 7, 10]</td>
</tr>
<tr>
<td></td>
<td>5.5V</td>
<td>Min 3.5 Max 2.0 mA</td>
<td>Clock Divide-by-16</td>
<td>1.0</td>
<td>mA</td>
<td>[4, 7, 10]</td>
</tr>
<tr>
<td></td>
<td>3.0V</td>
<td>Min 2.0 Max 1.0 mA</td>
<td>Clock Divide-by-16</td>
<td>1.0</td>
<td>mA</td>
<td>[4, 7, 10, 11]</td>
</tr>
<tr>
<td></td>
<td>5.5V</td>
<td>Min 4.5 Max 2.5 mA</td>
<td>Clock Divide-by-16</td>
<td>1.0</td>
<td>mA</td>
<td>[4, 7, 10, 11]</td>
</tr>
<tr>
<td>$I_{CC2}$ Standby Current</td>
<td>3.0V</td>
<td>Min 10 Max 1.0 µA</td>
<td>STOP Mode $V_{in} = 0V$, $V_{cc} WDT$ is not Running</td>
<td>1.0</td>
<td>µA</td>
<td>[6, 9]</td>
</tr>
<tr>
<td></td>
<td>5.5V</td>
<td>Min 10 Max 3.0 µA</td>
<td>STOP Mode $V_{in} = 0V$, $V_{cc} WDT$ is not Running</td>
<td>1.0</td>
<td>µA</td>
<td>[6, 9]</td>
</tr>
<tr>
<td></td>
<td>3.0V</td>
<td>Min 600 Max 400 µA</td>
<td>STOP Mode $V_{in} = 0V$, $V_{cc} WDT$ is Running</td>
<td>1.0</td>
<td>µA</td>
<td>[6, 9, 12]</td>
</tr>
<tr>
<td></td>
<td>5.5V</td>
<td>Min 1000 Max 800 µA</td>
<td>STOP Mode $V_{in} = 0V$, $V_{cc} WDT$ is Running</td>
<td>1.0</td>
<td>µA</td>
<td>[6, 9, 12]</td>
</tr>
<tr>
<td>$I_{ALL}$ Auto Latch Low Current</td>
<td>3.0V</td>
<td>Min 7.0 Max 14.0 µA</td>
<td>$0V &lt; V_{IN} &lt; V_{cc}$</td>
<td>1.0</td>
<td>µA</td>
<td>[6, 9]</td>
</tr>
<tr>
<td></td>
<td>5.5V</td>
<td>Min 20.0 Max 10 µA</td>
<td>$0V &lt; V_{IN} &lt; V_{cc}$</td>
<td>1.0</td>
<td>µA</td>
<td>[6, 9]</td>
</tr>
<tr>
<td>$I_{ALH}$ Auto Latch High Current</td>
<td>3.0V</td>
<td>Min -4.0 Max -8.0 µA</td>
<td>$0V &lt; V_{IN} &lt; V_{cc}$</td>
<td>1.0</td>
<td>µA</td>
<td>[6, 9]</td>
</tr>
<tr>
<td></td>
<td>5.5V</td>
<td>Min -9.0 Max -5.0 µA</td>
<td>$0V &lt; V_{IN} &lt; V_{cc}$</td>
<td>1.0</td>
<td>µA</td>
<td>[6, 9]</td>
</tr>
<tr>
<td>$V_{LV}$ Low Voltage Protection Voltage</td>
<td>1.50</td>
<td>Min 2.95 Max 2.95 V</td>
<td>2 MHz max Ext. CLK Freq.</td>
<td>2.1</td>
<td>V</td>
<td>[3]</td>
</tr>
</tbody>
</table>

### Notes:

1. $I_{CC1}$
   - Clock Driven on XTAL
   - Crystal or Ceramic Resonator

2. $V_{ss} = 0V = GND$

3. $V_{cc} = 3.0V$ to $5.5V$. The $V_{LV}$ increases as the temperature decreases. Typical values measured at $3.3V$ and $5.0V$.

4. All outputs unloaded, I/O pins floating, inputs at rail.

5. $C_{1L} = C_{1U} = 100$ pF

6. Same as note [4] except inputs at $V_{cc}$

7. For analog comparator inputs when analog comparators are enabled.

8. Excludes clock pins.

9. Clock must be forced Low when XTAL1 is clock-driven and XTAL2 is floating.

10. STD mode (not low EMI mode).

11. Z86C06 only.

12. Internal RC is WDT clock source.

13. $C_{1L} = 100$ pF, $C_{1U} = 220$ pF
### AC Electrical Characteristics

(SCLK/TCLK = EXTERNAL/2)

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$</th>
<th>$T_e = 0^\circ C$ to $+70^\circ C$</th>
<th>$T_e = -40^\circ C$ to $+105^\circ C$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>1</td>
<td>$T_{pc}$</td>
<td>Input Clock Period</td>
<td>3.0V</td>
<td>125 DC</td>
<td>83 DC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>125 DC</td>
<td>83 DC</td>
</tr>
<tr>
<td>2</td>
<td>$T_{rc}$</td>
<td>Clock Input Rise</td>
<td>3.0V</td>
<td>25 15</td>
<td>25 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and Fall Times</td>
<td>5.5V</td>
<td>25 15</td>
<td>25 15</td>
</tr>
<tr>
<td>3</td>
<td>$T_{wc}$</td>
<td>Input Clock Width</td>
<td>3.0V</td>
<td>62 41</td>
<td>62 41</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>62 41</td>
<td>62 41</td>
</tr>
<tr>
<td>4</td>
<td>$T_{wtinL}$</td>
<td>Timer Input Low Width</td>
<td>3.0V</td>
<td>100 100</td>
<td>100 100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70 70</td>
<td>70 70</td>
</tr>
<tr>
<td>5</td>
<td>$T_{wtinH}$</td>
<td>Timer Input High Width</td>
<td>3.0V</td>
<td>5$T_{pc}$</td>
<td>5$T_{pc}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5$T_{pc}$</td>
<td>5$T_{pc}$</td>
</tr>
</tbody>
</table>

**Notes:**

[1,7] Indicates specific notes or conditions.
### AC Electrical Characteristics (Continued)
(SCLK/TCLK = EXTERNAL/2)

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>( V_{cc} )</th>
<th>( T_{A} = 0^\circ C ) to +70°</th>
<th>( T_{A} = -40^\circ C ) to +105°</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( \text{Min} )</td>
<td>( \text{Max} )</td>
<td>( \text{Min} )</td>
<td>( \text{Max} )</td>
</tr>
<tr>
<td>6</td>
<td>TpTin</td>
<td>Timer Input Period</td>
<td>3.0V</td>
<td>8TpC</td>
<td>8TpC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>8TpC</td>
<td>8TpC</td>
</tr>
<tr>
<td>7</td>
<td>TtTin,</td>
<td>Timer Input Rise</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>TtTin</td>
<td>and Fall Timer</td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>8</td>
<td>TwL</td>
<td>Int. Request Input Low</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time</td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>9</td>
<td>TwLH</td>
<td>Int. Request Input High</td>
<td>3.0V</td>
<td>5TpC</td>
<td>5TpC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time</td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
</tr>
<tr>
<td>10</td>
<td>Twsm</td>
<td>STOP-Mode Recovery Width</td>
<td>3.0V</td>
<td>5TpC</td>
<td>5TpC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spec</td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
</tr>
<tr>
<td>11</td>
<td>Tost</td>
<td>Oscillator Startup Time</td>
<td>3.0V</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>Twdt</td>
<td>Watch-Dog Timer Refresh</td>
<td>3.0V</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time</td>
<td>5.5V</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.0V</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.0V</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.0V</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td>13</td>
<td>TpOR</td>
<td>Power-On Reset</td>
<td>3.0V</td>
<td>7</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>3</td>
<td>13</td>
</tr>
</tbody>
</table>

**Notes:**

[1] Timing Reference uses 0.7 \( V_{cc} \) for a logic 1 and 0.2 \( V_{cc} \) for a logic 0.
[2] Interrupt request via Port 3 (P31-P33).
[3] \( V_{cc} = 3.0V \) to 5.5V.
[4] SMR-D5 = 0.
[5] WDMR Register
[6] Internal RC Oscillator only.
[7] SMR D1 = 0.
[8] Maximum frequency for internal system clock is 4 MHz when using SCLK = EXTERNAL clock mode.
[10] SMR-D5 = 1, STOP-Mode Recovery delay is on.
[11] Z86C03 = 8 MHz; Z86C06 = 12 MHz.
[12] Z86C06 only.
### AC ELECTRICAL CHARACTERISTICS

Additional Timing Table (Divide-By-One Mode, SCLK/TCLK = EXTERNAL)

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$</th>
<th>$T_A = 0°C$ to $+70°C$ Min</th>
<th>$T_A = 0°C$ to $+70°C$ Max</th>
<th>$T_A = -40°C$ to $+105°C$ Min</th>
<th>$T_A = -40°C$ to $+105°C$ Max</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$T_Pc$</td>
<td>Input Clock Period</td>
<td>3.0V</td>
<td>250 DC</td>
<td>250 DC</td>
<td>4 MHz</td>
<td>250 DC</td>
<td>250 DC</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>250 DC</td>
<td>250 DC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$T_{rc},T_{tc}$</td>
<td>Clock Input Rise and Fall Times</td>
<td>3.0V</td>
<td>25</td>
<td>25</td>
<td>4 MHz</td>
<td>25</td>
<td>25</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$T_{wc}$</td>
<td>Input Clock Width</td>
<td>3.0V</td>
<td>125</td>
<td>125</td>
<td>4 MHz</td>
<td>125</td>
<td>125</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>125</td>
<td>125</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$T_{wTinL}$</td>
<td>Timer Input Low Width</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>4 MHz</td>
<td>100</td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$T_{wTinH}$</td>
<td>Timer Input High Width</td>
<td>3.0V</td>
<td>3$T_Pc$</td>
<td>3$T_Pc$</td>
<td>4 MHz</td>
<td>3$T_Pc$</td>
<td>3$T_Pc$</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>3$T_Pc$</td>
<td>3$T_Pc$</td>
<td></td>
<td></td>
<td></td>
<td>[1,7,8]</td>
</tr>
<tr>
<td>6</td>
<td>$T_{pTin}$</td>
<td>Timer Input Period</td>
<td>3.0V</td>
<td>4$T_Pc$</td>
<td>4$T_Pc$</td>
<td>4 MHz</td>
<td>4$T_Pc$</td>
<td>4$T_Pc$</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>4$T_Pc$</td>
<td>4$T_Pc$</td>
<td></td>
<td></td>
<td></td>
<td>[1,7,8]</td>
</tr>
<tr>
<td>7</td>
<td>$T_{TrTin},T_{TfTin}$</td>
<td>Timer Input Rise &amp; Fall Timer</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>4 MHz</td>
<td>100</td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>$T_{TwL}$</td>
<td>Int. Request Low Time</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>4 MHz</td>
<td>100</td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td>[1,7,8]</td>
</tr>
<tr>
<td>9</td>
<td>$T_{TwH}$</td>
<td>Int. Request High Time</td>
<td>3.0V</td>
<td>3$T_Pc$</td>
<td>3$T_Pc$</td>
<td>4 MHz</td>
<td>3$T_Pc$</td>
<td>2$T_Pc$</td>
<td>[1,2,7,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>3$T_Pc$</td>
<td>2$T_Pc$</td>
<td></td>
<td></td>
<td></td>
<td>[1,2,7,8]</td>
</tr>
<tr>
<td>10</td>
<td>$T_{ws}$</td>
<td>STOP-Mode Recovery Width Spec</td>
<td>3.0V</td>
<td>12</td>
<td>12</td>
<td>4 MHz</td>
<td>12</td>
<td>12</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>12</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td>[4,8]</td>
</tr>
<tr>
<td>11</td>
<td>$T_{os}$</td>
<td>Oscillator Startup Time</td>
<td>3.0V</td>
<td>5$T_Pc$</td>
<td>5$T_Pc$</td>
<td>4 MHz</td>
<td>5$T_Pc$</td>
<td>5$T_Pc$</td>
<td>[3,8,9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5$T_Pc$</td>
<td>5$T_Pc$</td>
<td></td>
<td></td>
<td></td>
<td>[3,8,9]</td>
</tr>
</tbody>
</table>

**Notes:**

[1] Timing Reference uses $0.7V_{cc}$ for a logic 1 and $0.2V_{cc}$ for a logic 0.


[3] SMR-D5 = 0.

[4] SMR-D5 = 1, POR STOP mode delay is on.

[5] Reg. WDTMR.

[6] $V_{cc} = 3.0V$ to $5.5V$.


[8] Maximum frequency for internal system clock is 4 MHz when using XTAL divide-by-one mode.

[9] For RC and LC oscillator, and for oscillator driven by clock driver.
EXPANDED REGISTER FILE CONTROL REGISTERS

**Figure 25. STOP-Mode Recovery Register**
(Write Only except bit D7, which is Read Only)

<table>
<thead>
<tr>
<th>SMR (F) 0B</th>
<th>PCON (F) 00</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>D7 D6 D5 D4 D3 D2 D1 D0</strong></td>
<td><strong>D7 D6 D5 D4 D3 D2 D1 D0</strong></td>
</tr>
<tr>
<td>SCLK/CLK Divide-by-16 (C06)†</td>
<td>Comparator</td>
</tr>
<tr>
<td>0 OFF</td>
<td>Output PORT 3</td>
</tr>
<tr>
<td>1 ON</td>
<td>0 P34, P35 Standard Output*</td>
</tr>
<tr>
<td></td>
<td>1 P34, P35 Comparator Output</td>
</tr>
<tr>
<td>External Clock Divide Mode</td>
<td>Reserved (Must Be 1)</td>
</tr>
<tr>
<td>0 SCLK/CLK = External/2&quot;</td>
<td>Low EMI PORT 2</td>
</tr>
<tr>
<td>1 SCLK/CLK = External/1</td>
<td>1 Standard *</td>
</tr>
<tr>
<td>Stop Mode Recovery Source</td>
<td>Low EMI PORT 3</td>
</tr>
<tr>
<td>000 POR Only††</td>
<td>0 Low EMI Noise</td>
</tr>
<tr>
<td>001 POR Only††</td>
<td>1 Standard *</td>
</tr>
<tr>
<td>010 P31 ††</td>
<td>Low EMI Noise</td>
</tr>
<tr>
<td>011 P32 ††</td>
<td>1 Standard *</td>
</tr>
<tr>
<td>100 P33 ††</td>
<td>Low EMI Noise</td>
</tr>
<tr>
<td>101 P27</td>
<td>1 Standard *</td>
</tr>
<tr>
<td>110 P2 NOR 0:3</td>
<td>Stop Recovery Level</td>
</tr>
<tr>
<td>111 P2 NOR 0:7</td>
<td>0 Low Level *</td>
</tr>
<tr>
<td>Stop Delay</td>
<td>1 High Level</td>
</tr>
<tr>
<td>0 OFF</td>
<td>Stop Recovery</td>
</tr>
<tr>
<td>1 ON †</td>
<td>0 POR</td>
</tr>
<tr>
<td>Stop Recovery Level</td>
<td>1 Stop Recovery</td>
</tr>
<tr>
<td>0 Low Level *</td>
<td>* Default Setting After Reset.</td>
</tr>
<tr>
<td>1 High Level</td>
<td></td>
</tr>
<tr>
<td>Stop Flag</td>
<td></td>
</tr>
<tr>
<td>0 POR</td>
<td>* Default Setting After Reset.</td>
</tr>
<tr>
<td>1 Stop Recovery</td>
<td></td>
</tr>
</tbody>
</table>

* Default setting after RESET.
** Default setting after STOP-Mode Recovery and RESET.
†† C03 Reserved; must be 0.

**Figure 26. Watch-Dog Timer Mode Register**
(Write Only)

<table>
<thead>
<tr>
<th>WDTMR (F) 0F</th>
<th>SCON (C) 02</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>D7 D6 D5 D4 D3 D2 D1 D0</strong></td>
<td><strong>D7 D6 D5 D4 D3 D2 D1 D0</strong></td>
</tr>
<tr>
<td>WDT Tap† INT RC OSC XTAL CLK</td>
<td>SPI Enable</td>
</tr>
<tr>
<td>00 5 512 TpC</td>
<td>0 Disable *</td>
</tr>
<tr>
<td>01* 15* 1024 TpC</td>
<td>1 Enable</td>
</tr>
<tr>
<td>10 25 2048 TpC</td>
<td>RxChar/Overrun (S)</td>
</tr>
<tr>
<td>11 100 8192 TpC</td>
<td>0 Reset</td>
</tr>
<tr>
<td>WDT During HALT</td>
<td>1 Overrun</td>
</tr>
<tr>
<td>0 OFF</td>
<td></td>
</tr>
<tr>
<td>1 ON †</td>
<td></td>
</tr>
<tr>
<td>WDT During STOP</td>
<td></td>
</tr>
<tr>
<td>0 OFF</td>
<td></td>
</tr>
<tr>
<td>1 ON †</td>
<td></td>
</tr>
<tr>
<td>XTAL1/INT RC Select for WDT</td>
<td></td>
</tr>
<tr>
<td>0 On-Board RC *</td>
<td></td>
</tr>
<tr>
<td>1 XTAL</td>
<td></td>
</tr>
<tr>
<td>Reserved (Must be 0)</td>
<td></td>
</tr>
</tbody>
</table>

* Default setting after RESET.
† For C06; C03 Must Be D0 = 1, D1 = 0

**Figure 27. PORT Control Register**
(Write Only)

**Figure 28. SPI Control Register**
(Z86C06 Only)
Z8 CONTROL REGISTER DIAGRAMS

**Figure 29. SPI Compare Register** (Z86C06 Only)

**Figure 30. SPI Receive Buffer** (Z86C06 Only)

**Figure 31. Reserved**

**Figure 32. Timer Mode Register** (F1H: Read/Write)

**Figure 33. Counter Timer 1 Register** (F2H: Read/Write)

**Figure 34. Prescaler 1 Register** (F3H: Write Only)

**Figure 35. Counter/Timer 0 Register** (F4H: Read/Write; Z86C06 Only)

**Figure 36. Prescaler 0 Register** (F5H: Write Only; Z86C06 Only)
Z8 CONTROL REGISTER DIAGRAMS (Continued)

Figure 37. Port 2 Mode Register (F6H: Write Only)

Figure 38. Port 3 Mode Register (F7H: Write Only)

Figure 39. Port 0 and 1 Mode Register (F8H: Write Only)

Figure 40. Interrupt Priority Register (F9H: Write Only)

Figure 41. Interrupt Request Register (FAH: Read/Write)
Figure 42. Interrupt Mask Register (FBH: Read/Write)

Figure 43. Flag Register (FCH: Read/Write)

Figure 44. Register Pointer (FDH: Read/Write)

Figure 45. General Purpose Register (FEH: Read/Write)

Figure 46. Stack Pointer (FFH: Read/Write)
DEVICE CHARACTERISTICS

Figure 47. Typical \( I_{cc} \) vs Frequency

Figure 48. Typical \( V_{OL}, V_{IL} \) vs Temperature

Legend:
- A = \( \text{VII} \) at \( V_{CC} = 3.3\text{V} \)
- B = \( \text{VII} \) at \( V_{CC} = 5.5\text{V} \)
- C = \( \text{Volt} \) at \( V_{CC} = 3.0\text{V} \)
- D = \( \text{Volt} \) at \( V_{CC} = 5.5\text{V} \)
Figure 49. Typical $V_{OH}$, $V_{IH}$ vs Temperature
DEVICE CHARACTERISTICS (Continued)

Figure 50. Typical $V_{\text{OH}}$ vs $I_{\text{OH}}$ Over Temperature

Legend:

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>125°C</td>
<td>A, D</td>
</tr>
<tr>
<td>25°C</td>
<td>B, E</td>
</tr>
<tr>
<td>-55°C</td>
<td>C, F</td>
</tr>
</tbody>
</table>

Note: STD Mode (Not Low EMI Mode)
Figure 51. Typical $I_{OL}$ vs $V_{OL}$ Over Temperature
Figure 52. Typical Power-On Reset Time vs Temperature

Legend:

<table>
<thead>
<tr>
<th>Legend</th>
<th>Vcc/Vee (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3.0V</td>
</tr>
<tr>
<td>B</td>
<td>3.5V</td>
</tr>
<tr>
<td>C</td>
<td>4.0V</td>
</tr>
<tr>
<td>D</td>
<td>4.5V</td>
</tr>
<tr>
<td>E</td>
<td>5.0V</td>
</tr>
<tr>
<td>F</td>
<td>5.5V</td>
</tr>
</tbody>
</table>

Note: Using Internal RC.
Figure 53. Typical 5 ms WDT Setting vs Temperature (Z86C06 Only)
DEVICE CHARACTERISTICS (Continued)

Figure 54. Typical 15 ms WDT Setting vs Temperature

Legend:
- A - Vcc = 3.0V
- B - Vcc = 3.5V
- C - Vcc = 4.0V
- D - Vcc = 4.5V
- E - Vcc = 5.0V
- F - Vcc = 5.5V

Note: Using internal RC.
Figure 55. Typical 25 ms WDT Setting vs Temperature (Z86C06 Only)

Legend:

<table>
<thead>
<tr>
<th>Legend</th>
<th>Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Vcc = 3.0V</td>
</tr>
<tr>
<td>B</td>
<td>Vcc = 3.5V</td>
</tr>
<tr>
<td>C</td>
<td>Vcc = 4.0V</td>
</tr>
<tr>
<td>D</td>
<td>Vcc = 4.5V</td>
</tr>
<tr>
<td>E</td>
<td>Vcc = 5.0V</td>
</tr>
<tr>
<td>F</td>
<td>Vcc = 5.5V</td>
</tr>
</tbody>
</table>

Note: Using internal RC.
Figure 56. Typical 100 ms WDT Setting vs Temperature (Z86C06 Only)
**Figure 57. Typical Frequency vs RC Resistance**

Legend:

A - $V_{cc} = 5.0V \ C = 33 \text{ pF}$

B - $V_{cc} = 3.3V \ C = 33 \text{ pF}$

*Note:* STD Mode (not Low EMI Mode).

*Note:* This chart for reference only. Each process will have a different characteristic curve.
DEVICE CHARACTERISTICS (Continued)

Legend:

<table>
<thead>
<tr>
<th>Curve</th>
<th>Condition</th>
<th>Resistance</th>
<th>Capacitance (pF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Vcc = 5.0V R = 22 K Ohms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Vcc = 5.0V R = 56 K Ohms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Vcc = 5.0V R = 100 K Ohms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Vcc = 5.0V R = 1 M Ohms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Vcc = 5.0V R = 4 M Ohms</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Not in Low EMI Mode

Figure 58. Typical RC Resistance/Capacitance vs Frequency
Figure 59. Auto Latch Characteristics

Legend:

<table>
<thead>
<tr>
<th>Legend</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Vcc = 3.0V</td>
</tr>
<tr>
<td>B</td>
<td>Vcc = 4.5V</td>
</tr>
<tr>
<td>C</td>
<td>Vcc = 5.0V</td>
</tr>
<tr>
<td>D</td>
<td>Vcc = 5.5V</td>
</tr>
</tbody>
</table>
INSTRUCTION SET NOTATION

Addressing Modes. The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRR</td>
<td>Indirect register pair or indirect working-register pair address</td>
</tr>
<tr>
<td>Irr</td>
<td>Indirect working-register pair only</td>
</tr>
<tr>
<td>X</td>
<td>Indexed address</td>
</tr>
<tr>
<td>DA</td>
<td>Direct address</td>
</tr>
<tr>
<td>RA</td>
<td>Relative address</td>
</tr>
<tr>
<td>IM</td>
<td>Immediate</td>
</tr>
<tr>
<td>R</td>
<td>Register or working-register address</td>
</tr>
<tr>
<td>r</td>
<td>Working-register address only</td>
</tr>
<tr>
<td>IR</td>
<td>Indirect-register or indirect working-register address</td>
</tr>
<tr>
<td>Ir</td>
<td>Indirect working-register address only</td>
</tr>
<tr>
<td>RR</td>
<td>Register pair or working register pair address</td>
</tr>
</tbody>
</table>

Symbols. The following symbols are used in describing the instruction set.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst</td>
<td>Destination location or contents</td>
</tr>
<tr>
<td>src</td>
<td>Source location or contents</td>
</tr>
<tr>
<td>cc</td>
<td>Condition code</td>
</tr>
<tr>
<td>@</td>
<td>Indirect address prefix</td>
</tr>
<tr>
<td>SP</td>
<td>Stack Pointer</td>
</tr>
<tr>
<td>PC</td>
<td>Program Counter</td>
</tr>
<tr>
<td>FLAGS</td>
<td>Flag register (Control Register 252)</td>
</tr>
<tr>
<td>RP</td>
<td>Register Pointer (R253)</td>
</tr>
<tr>
<td>IMR</td>
<td>Interrupt mask register (R251)</td>
</tr>
</tbody>
</table>

Flags. Control register (R252) contains the following six flags:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Carry flag</td>
</tr>
<tr>
<td>Z</td>
<td>Zero flag</td>
</tr>
<tr>
<td>S</td>
<td>Sign flag</td>
</tr>
<tr>
<td>V</td>
<td>Overflow flag</td>
</tr>
<tr>
<td>D</td>
<td>Decimal-adjust flag</td>
</tr>
<tr>
<td>H</td>
<td>Half-carry flag</td>
</tr>
</tbody>
</table>

Affected flags are indicated by:

- 0: Clear to zero
- 1: Set to one
- *: Set to clear according to operation
- -: Unaffected
- x: Undefined
### CONDITION CODES

<table>
<thead>
<tr>
<th>Value</th>
<th>Mnemonic</th>
<th>Meaning</th>
<th>Flags Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td></td>
<td>Always True</td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>C</td>
<td>Carry</td>
<td>C = 1</td>
</tr>
<tr>
<td>1111</td>
<td>NC</td>
<td>No Carry</td>
<td>C = 0</td>
</tr>
<tr>
<td>0110</td>
<td>Z</td>
<td>Zero</td>
<td>Z = 1</td>
</tr>
<tr>
<td>1110</td>
<td>NZ</td>
<td>Not Zero</td>
<td>Z = 0</td>
</tr>
<tr>
<td>1101</td>
<td>PL</td>
<td>Plus</td>
<td>S = 0</td>
</tr>
<tr>
<td>0101</td>
<td>MI</td>
<td>Minus</td>
<td>S = 1</td>
</tr>
<tr>
<td>0100</td>
<td>OV</td>
<td>Overflow</td>
<td>V = 1</td>
</tr>
<tr>
<td>1100</td>
<td>NOV</td>
<td>No Overflow</td>
<td>V = 0</td>
</tr>
<tr>
<td>0110</td>
<td>EQ</td>
<td>Equal</td>
<td>Z = 1</td>
</tr>
<tr>
<td>1110</td>
<td>NE</td>
<td>Not Equal</td>
<td>Z = 0</td>
</tr>
<tr>
<td>1001</td>
<td>GE</td>
<td>Greater Than or Equal</td>
<td>(S XOR V) = 0</td>
</tr>
<tr>
<td>0001</td>
<td>LT</td>
<td>Less than</td>
<td>(S XOR V) = 1</td>
</tr>
<tr>
<td>1010</td>
<td>GT</td>
<td>Greater Than</td>
<td>[Z OR (S XOR V)] = 0</td>
</tr>
<tr>
<td>0010</td>
<td>LE</td>
<td>Less Than or Equal</td>
<td>[Z OR (S XOR V)] = 1</td>
</tr>
<tr>
<td>1111</td>
<td>UGE</td>
<td>Unsigned Greater Than or Equal</td>
<td>C = 0</td>
</tr>
<tr>
<td>0111</td>
<td>ULT</td>
<td>Unsigned Less Than</td>
<td>C = 1</td>
</tr>
<tr>
<td>1011</td>
<td>UGT</td>
<td>Unsigned Greater Than</td>
<td>(C = 0 AND Z = 0) = 1</td>
</tr>
<tr>
<td>0011</td>
<td>ULE</td>
<td>Unsigned Less Than or Equal</td>
<td>(C OR Z) = 1</td>
</tr>
<tr>
<td>0000</td>
<td>F</td>
<td>Never True (Always False)</td>
<td></td>
</tr>
</tbody>
</table>
INSTRUCTION FORMATS

One-Byte Instructions

Two-Byte Instructions

Three-Byte Instructions

INSTRUCTION SUMMARY

Note: Assignment of a value is indicated by the symbol "\( \leftarrow \)". For example:

\[ \text{dst} \leftarrow \text{dst} + \text{src} \]

indicates that the source data is added to the destination data and the result is stored in the destination location. The notation "\( \text{addr} (n) \)" is used to refer to bit \( (n) \) of a given operand location. For example:

\[ \text{dst} (7) \]

refers to bit 7 of the destination operand.
### INSTRUCTION SUMMARY (Continued)

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
<th>D</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC dst, src</td>
<td>dst ← dst + src +c</td>
<td>t</td>
<td>1[ ]</td>
<td>* * * * * 0 *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD dst, src</td>
<td>dst ← dst + src</td>
<td>t</td>
<td>0[ ]</td>
<td>* * * * * 0 *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AND dst, src</td>
<td>dst ← dst AND src</td>
<td>t</td>
<td>5[ ]</td>
<td>* * 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALL dst</td>
<td>IR</td>
<td>D6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP ← SP - 2</td>
<td>RR</td>
<td>D4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@SP ← PC,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC ← dst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCF</td>
<td></td>
<td>EF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLR dst</td>
<td>dst ← 0</td>
<td>R</td>
<td>80</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COM dst</td>
<td>dst ← NOT dst</td>
<td>R</td>
<td>60</td>
<td>* * 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP dst, src</td>
<td>dst ← src</td>
<td>t</td>
<td>A[ ]</td>
<td>* * * -</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DA dst</td>
<td>dst ← DA dst</td>
<td>R</td>
<td>40</td>
<td>* * X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEC dst</td>
<td>dst ← dst - 1</td>
<td>R</td>
<td>00</td>
<td>* *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECW dst</td>
<td>dst ← dst - 1</td>
<td>RR</td>
<td>80</td>
<td>* * *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DJNZ r, dst</td>
<td>r ← r - 1</td>
<td>RA</td>
<td>rA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDC dst, src</td>
<td>dst ← src</td>
<td>r</td>
<td>Im</td>
<td>rC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDCI dst, src</td>
<td>dst ← src</td>
<td>Ir</td>
<td>Irr</td>
<td>C3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td></td>
<td>FF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INCM dst</td>
<td>dst ← dst + 1</td>
<td>r</td>
<td>rE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INCW dst</td>
<td>dst ← dst + 1</td>
<td>RR</td>
<td>A0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRET</td>
<td></td>
<td>BF</td>
<td>* * * *</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JP cc, dst</td>
<td>PC ← dst</td>
<td>DA</td>
<td>cD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR cc, dst</td>
<td>PC ← PC + dst</td>
<td>RA</td>
<td>cB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDCI dst, src</td>
<td>dst ← src</td>
<td>Ir</td>
<td>Irr</td>
<td>C3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td></td>
<td>FF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### INSTRUCTION SUMMARY (Continued)

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
</table>
| OR dst, src dst ← dst OR src | R | 4[ ]obile | * * 0 * * | SUB dst, src dst ← dst - src | R | 2[ ]obile | * * 1 * * *
| POP dst dst ← @SP; SP ← SP + 1 | R | 50 | - - - - - - | SWAP dst | °R | F0 | X * X - - |
| PUSH src SP ← SP - 1; @SP ← src | R | 70 | - - - - - - | TCM dst, src (NOT dst) AND src | R | 6[ ]obile | - * 0 - - |
| RCF C ← 0 | CF | 0 | - - - - - - | TM dst, src dst AND src | R | 7[ ]obile | - * 0 - - |
| RET PC ← @SP; SP ← SP + 2 | AF | - - - - - - | WDT | 5F | X X X - - |
| RL dst R | 90 | * * * * - - | RR dst | R | E0 | * * * * - - |
| RLC dst R | 10 | * * * * - - | RRC dst | R | C0 | * * * * - - |
| RR dst R | 9[ ]obile | * * * * - - | SBC dst, src dst ← dst - src - C | R | 3[ ]obile | * * 1 * * |
| SCF C ← 1 | DF | 1 | - - - - - - | SRA dst R | D0 | * * 0 - - |
| SRA dst R | D1 | * * 0 - - | SRP dst Im | 31 | - - - - - - |
| STOP | 6F | 1 | - - - - - - |

* These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a '[ ]' in this table, and its value is found in the following table to the left of the applicable addressing mode pair.

For example, the opcode of an ADC instruction using the addressing modes r (destination) and Ir (source) is 13.
### Opcode Map

#### Lower Nibble (Hex)

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>DEC</td>
<td>DEC</td>
<td>ADD</td>
<td>ADD</td>
<td>ADD</td>
<td>ADD</td>
<td>ADD</td>
<td>ADD</td>
<td>ADD</td>
<td>ADD</td>
<td>ADD</td>
<td>ADD</td>
<td>ADD</td>
<td>ADD</td>
<td>ADD</td>
<td>ADD</td>
</tr>
<tr>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>RLC</td>
<td>RLC</td>
<td>RLC</td>
<td>RLC</td>
<td>RLC</td>
<td>RLC</td>
<td>RLC</td>
<td>RLC</td>
<td>RLC</td>
<td>RLC</td>
<td>RLC</td>
<td>RLC</td>
<td>RLC</td>
<td>RLC</td>
<td>RLC</td>
<td>RLC</td>
</tr>
<tr>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>INC</td>
<td>INC</td>
<td>SUB</td>
<td>SUB</td>
<td>SUB</td>
<td>SUB</td>
<td>SUB</td>
<td>SUB</td>
<td>SUB</td>
<td>SUB</td>
<td>SUB</td>
<td>SUB</td>
<td>SUB</td>
<td>SUB</td>
<td>SUB</td>
<td>SUB</td>
</tr>
<tr>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
</tr>
<tr>
<td>JP</td>
<td>JP</td>
<td>SBC</td>
<td>SBC</td>
<td>SBC</td>
<td>SBC</td>
<td>SBC</td>
<td>SBC</td>
<td>SBC</td>
<td>SBC</td>
<td>SBC</td>
<td>SBC</td>
<td>SBC</td>
<td>SBC</td>
<td>SBC</td>
<td>SBC</td>
</tr>
<tr>
<td>IR1</td>
<td>IR1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
</tr>
<tr>
<td>DA</td>
<td>DA</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
</tr>
<tr>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
</tr>
<tr>
<td>POP</td>
<td>POP</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
</tr>
<tr>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
</tr>
<tr>
<td>COM</td>
<td>COM</td>
<td>TCM</td>
<td>TCM</td>
<td>TCM</td>
<td>TCM</td>
<td>TCM</td>
<td>TCM</td>
<td>TCM</td>
<td>TCM</td>
<td>TCM</td>
<td>TCM</td>
<td>TCM</td>
<td>TCM</td>
<td>TCM</td>
<td>TCM</td>
</tr>
<tr>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>7.0</td>
<td>7.0</td>
<td>7.0</td>
<td>7.0</td>
<td>7.0</td>
<td>7.0</td>
<td>7.0</td>
<td>7.0</td>
<td>7.0</td>
<td>7.0</td>
<td>7.0</td>
<td>7.0</td>
<td>7.0</td>
<td>7.0</td>
<td>7.0</td>
<td>7.0</td>
</tr>
<tr>
<td>10/12.1</td>
<td>10/12.1</td>
<td>10/12.1</td>
<td>10/12.1</td>
<td>10/12.1</td>
<td>10/12.1</td>
<td>10/12.1</td>
<td>10/12.1</td>
<td>10/12.1</td>
<td>10/12.1</td>
<td>10/12.1</td>
<td>10/12.1</td>
<td>10/12.1</td>
<td>10/12.1</td>
<td>10/12.1</td>
<td>10/12.1</td>
</tr>
<tr>
<td>PUSHR2</td>
<td>PUSHR2</td>
<td>PUSHR2</td>
<td>PUSHR2</td>
<td>PUSHR2</td>
<td>PUSHR2</td>
<td>PUSHR2</td>
<td>PUSHR2</td>
<td>PUSHR2</td>
<td>PUSHR2</td>
<td>PUSHR2</td>
<td>PUSHR2</td>
<td>PUSHR2</td>
<td>PUSHR2</td>
<td>PUSHR2</td>
<td>PUSHR2</td>
</tr>
<tr>
<td>8.0</td>
<td>8.0</td>
<td>8.0</td>
<td>8.0</td>
<td>8.0</td>
<td>8.0</td>
<td>8.0</td>
<td>8.0</td>
<td>8.0</td>
<td>8.0</td>
<td>8.0</td>
<td>8.0</td>
<td>8.0</td>
<td>8.0</td>
<td>8.0</td>
<td>8.0</td>
</tr>
<tr>
<td>DECW</td>
<td>DECW</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
<td>AND</td>
</tr>
<tr>
<td>RR1</td>
<td>RR1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>9.0</td>
<td>9.0</td>
<td>9.0</td>
<td>9.0</td>
<td>9.0</td>
<td>9.0</td>
<td>9.0</td>
<td>9.0</td>
<td>9.0</td>
<td>9.0</td>
<td>9.0</td>
<td>9.0</td>
<td>9.0</td>
<td>9.0</td>
<td>9.0</td>
<td>9.0</td>
</tr>
<tr>
<td>RL</td>
<td>RL</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
</tr>
<tr>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>A.0</td>
<td>A.0</td>
<td>A.0</td>
<td>A.0</td>
<td>A.0</td>
<td>A.0</td>
<td>A.0</td>
<td>A.0</td>
<td>A.0</td>
<td>A.0</td>
<td>A.0</td>
<td>A.0</td>
<td>A.0</td>
<td>A.0</td>
<td>A.0</td>
<td>A.0</td>
</tr>
<tr>
<td>INCW</td>
<td>INCW</td>
<td>CP</td>
<td>CP</td>
<td>CP</td>
<td>CP</td>
<td>CP</td>
<td>CP</td>
<td>CP</td>
<td>CP</td>
<td>CP</td>
<td>CP</td>
<td>CP</td>
<td>CP</td>
<td>CP</td>
<td>CP</td>
</tr>
<tr>
<td>RR1</td>
<td>RR1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>B.0</td>
<td>B.0</td>
<td>B.0</td>
<td>B.0</td>
<td>B.0</td>
<td>B.0</td>
<td>B.0</td>
<td>B.0</td>
<td>B.0</td>
<td>B.0</td>
<td>B.0</td>
<td>B.0</td>
<td>B.0</td>
<td>B.0</td>
<td>B.0</td>
<td>B.0</td>
</tr>
<tr>
<td>CLR</td>
<td>CLR</td>
<td>XOR</td>
<td>XOR</td>
<td>XOR</td>
<td>XOR</td>
<td>XOR</td>
<td>XOR</td>
<td>XOR</td>
<td>XOR</td>
<td>XOR</td>
<td>XOR</td>
<td>XOR</td>
<td>XOR</td>
<td>XOR</td>
<td>XOR</td>
</tr>
<tr>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>C.0</td>
<td>C.0</td>
<td>C.0</td>
<td>C.0</td>
<td>C.0</td>
<td>C.0</td>
<td>C.0</td>
<td>C.0</td>
<td>C.0</td>
<td>C.0</td>
<td>C.0</td>
<td>C.0</td>
<td>C.0</td>
<td>C.0</td>
<td>C.0</td>
<td>C.0</td>
</tr>
<tr>
<td>RRC</td>
<td>RRC</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
</tr>
<tr>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>D.0</td>
<td>D.0</td>
<td>D.0</td>
<td>D.0</td>
<td>D.0</td>
<td>D.0</td>
<td>D.0</td>
<td>D.0</td>
<td>D.0</td>
<td>D.0</td>
<td>D.0</td>
<td>D.0</td>
<td>D.0</td>
<td>D.0</td>
<td>D.0</td>
<td>D.0</td>
</tr>
<tr>
<td>SRA</td>
<td>SRA</td>
<td>CALL*</td>
<td>CALL*</td>
<td>CALL*</td>
<td>CALL*</td>
<td>CALL*</td>
<td>CALL*</td>
<td>CALL*</td>
<td>CALL*</td>
<td>CALL*</td>
<td>CALL*</td>
<td>CALL*</td>
<td>CALL*</td>
<td>CALL*</td>
<td>CALL*</td>
</tr>
<tr>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>E.0</td>
<td>E.0</td>
<td>E.0</td>
<td>E.0</td>
<td>E.0</td>
<td>E.0</td>
<td>E.0</td>
<td>E.0</td>
<td>E.0</td>
<td>E.0</td>
<td>E.0</td>
<td>E.0</td>
<td>E.0</td>
<td>E.0</td>
<td>E.0</td>
<td>E.0</td>
</tr>
<tr>
<td>RR</td>
<td>RR</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
</tr>
<tr>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>F.0</td>
<td>F.0</td>
<td>F.0</td>
<td>F.0</td>
<td>F.0</td>
<td>F.0</td>
<td>F.0</td>
<td>F.0</td>
<td>F.0</td>
<td>F.0</td>
<td>F.0</td>
<td>F.0</td>
<td>F.0</td>
<td>F.0</td>
<td>F.0</td>
<td>F.0</td>
</tr>
<tr>
<td>SWAP</td>
<td>SWAP</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
<td>LD</td>
</tr>
<tr>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
<td>R1</td>
</tr>
</tbody>
</table>

### Legend:

- **R** = 8-bit address
- **r** = 4-bit address
- **R1** or **r1** = Dest address
- **R2** or **r2** = Src address

### Sequence:

- Opcode, First Operand, Second Operand

### Note:

- The blank areas are reserved.

* 2-byte instruction appears as a 3-byte instruction
PACKAGE INFORMATION

### SYMBOL INFORMATION

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MILLIMETER</th>
<th>INCH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MIN</td>
<td>MAX</td>
</tr>
<tr>
<td>A1</td>
<td>0.51</td>
<td>0.81</td>
</tr>
<tr>
<td>A2</td>
<td>3.25</td>
<td>3.43</td>
</tr>
<tr>
<td>B</td>
<td>0.38</td>
<td>0.53</td>
</tr>
<tr>
<td>B1</td>
<td>1.14</td>
<td>1.65</td>
</tr>
<tr>
<td>C</td>
<td>0.23</td>
<td>0.38</td>
</tr>
<tr>
<td>D</td>
<td>22.35</td>
<td>23.37</td>
</tr>
<tr>
<td>E</td>
<td>7.62</td>
<td>8.13</td>
</tr>
<tr>
<td>E1</td>
<td>6.22</td>
<td>6.48</td>
</tr>
<tr>
<td>EA</td>
<td>2.54</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>7.87</td>
<td>8.89</td>
</tr>
<tr>
<td>G</td>
<td>3.18</td>
<td>3.81</td>
</tr>
<tr>
<td>G1</td>
<td>1.52</td>
<td>1.65</td>
</tr>
<tr>
<td>H</td>
<td>0.89</td>
<td>1.65</td>
</tr>
</tbody>
</table>

### CONTROLLING DIMENSIONS INCH

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MILLIMETER</th>
<th>INCH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MIN</td>
<td>MAX</td>
</tr>
<tr>
<td>A</td>
<td>2.40</td>
<td>2.63</td>
</tr>
<tr>
<td>A1</td>
<td>0.10</td>
<td>0.30</td>
</tr>
<tr>
<td>A2</td>
<td>2.24</td>
<td>2.44</td>
</tr>
<tr>
<td>B</td>
<td>0.36</td>
<td>0.46</td>
</tr>
<tr>
<td>C</td>
<td>0.23</td>
<td>0.30</td>
</tr>
<tr>
<td>D</td>
<td>11.40</td>
<td>11.75</td>
</tr>
<tr>
<td>E</td>
<td>7.40</td>
<td>7.60</td>
</tr>
<tr>
<td>F</td>
<td>1.27</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>10.00</td>
<td>10.65</td>
</tr>
<tr>
<td>L</td>
<td>0.60</td>
<td>1.00</td>
</tr>
<tr>
<td>G1</td>
<td>0.97</td>
<td>1.07</td>
</tr>
</tbody>
</table>

**18-Pin DIP Package Diagram**

**18-Pin SOIC Package Diagram**
ORDERING INFORMATION

Z86C03 (8 MHz)

<table>
<thead>
<tr>
<th>Standard Temperature</th>
<th>18-Pin DIP</th>
<th>18-Pin SOIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z86C0308PSC</td>
<td>Z86C0308SSC</td>
<td></td>
</tr>
</tbody>
</table>

Extended Temperature

<table>
<thead>
<tr>
<th>18-Pin DIP</th>
<th>18-Pin SOIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z86C0308PEC</td>
<td>Z86C0308SEC</td>
</tr>
</tbody>
</table>

Z86C06 (12 MHz)

<table>
<thead>
<tr>
<th>Standard Temperature</th>
<th>18-Pin DIP</th>
<th>18-Pin SOIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z86C0612PSC</td>
<td>Z86C0612SSC</td>
<td></td>
</tr>
</tbody>
</table>

Extended Temperature

<table>
<thead>
<tr>
<th>18-Pin DIP</th>
<th>18-Pin SOIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z86C0612PEC</td>
<td>Z86C0612SEC</td>
</tr>
</tbody>
</table>

For fast results, contact your local Zilog sales offices for assistance in ordering the part(s) desired.

CODES

Preferred Package
P = Plastic DIP

Longer Lead Time
S = Plastic SOIC

Preferred Temperature
S = 0°C to +70°C

Longer Lead Time
E = −40°C to +105°C

Speeds
08 = 8 MHz
12 = 12 MHz

Environmental
C = Plastic Standard

Example:
Z 86C03 08 P C

is a Z86C03, 8 MHz, DIP, 0°C to +70°C, Plastic Standard Flow

Environmental Flow
Temperature
Package
Speed
Product Number
Zilog Prefix
FEATURES

■ The ZS6E03/E06 Devices Have the Following General Characteristics:

<table>
<thead>
<tr>
<th>Part</th>
<th>ROM</th>
<th>RAM</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZS6E03</td>
<td>512 bytes</td>
<td>60</td>
<td>8 MHz</td>
</tr>
<tr>
<td>ZS6E06</td>
<td>1 Kbyte</td>
<td>124</td>
<td>12 MHz</td>
</tr>
</tbody>
</table>

■ 18-Pin Package (DIP, SOIC)

■ 3.0 to 5.5 Volt Operating Range

■ Operating Temperature: −40°C to +105°C

■ Clock Speeds up to 8 MHz (E03) and 12 MHz (E06)

■ Fast Instruction Pointer: 1.5 µs @ 8 MHz (E03); 1.0 µs @ 12 MHz (E06)

■ Multiple Expanded Register File Control Registers and Two SPI Registers (ZS6E06 only)

■ One/Two Programmable 8-Bit Counter/Timers, Each with a 6-Bit Programmable Prescaler

GENERAL DESCRIPTION

Zilog’s ZS6E03/E06 OTP (One-Time Programmable) CCP™ (Consumer Controller Processors) are members of the Z8® single-chip microcontroller family with enhanced wake-up circuitry, programmable watch-dog timers and low noise/EMI options. These enhancements result in a more efficient, cost effective design and provide the user with increased design flexibility over the standard Z8 microcontroller core. With 512 and 1K bytes of EPROM and 60 and 124 bytes of general-purpose RAM, respectively. These low cost, low power consumption 18-pin CMOS microcontrollers offer fast execution, efficient use of memory, sophisticated interrupts, input/output bit manipulation capabilities, and easy hardware/software system expansion.

The ZS6E03/E06 architecture is characterized by Zilog’s 8-bit microcontroller core with the addition of an Expanded Register File to allow easy access to register mapped peripheral and I/O circuits. The ZS6E03/E06 offers a flexible I/O scheme, an efficient register and address space structure, and a number of ancillary features that are useful in many consumer, automotive, and industrial applications.

For applications demanding powerful I/O capabilities, the ZS6E03/E06 provides 14 pins dedicated to input and output. These lines are grouped into two ports and are configurable under software control to provide timing, status signals, or parallel I/O.
GENERAL DESCRIPTION (Continued)

Three basic address spaces are available to support this wide range of configurations: Program Memory, Register File, and Expanded Register File (Figure 1). The Register File is composed of 60/124 bytes of General-Purpose Registers, two I/O Port registers, and thirteen/fifteen Control and Status registers. The Expanded Register File consists of three control registers in the Z86E03, and four control registers, a SPI Receive Buffer, and a SPI compare register in the Z86E06.

With powerful peripheral features such as on-board comparators, counter/timer(s), Watch-Dog Timer (WDT), and serial peripheral interface (E06 only), the Z86E03/E06 meets the needs of a variety of sophisticated controller applications.

Notes:
All Signals with a preceding front slash, "/", are active Low, e.g.: B/W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

<table>
<thead>
<tr>
<th>Connection</th>
<th>Circuit</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>$V_{CC}$</td>
<td>$V_{CC}$</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
<td>$V_{DD}$</td>
</tr>
<tr>
<td>XTAL</td>
<td>Machine Timing &amp; Inst. Control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WDT, POR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prg. Memory 512/1K x 8-Bit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Register File 60/124 x 8-Bit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Register Pointer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Program Counter</td>
<td></td>
</tr>
</tbody>
</table>

† Z86E03 has one counter/timer.  
* Serial Peripheral Interface for the Z86E06 only.

Figure 1. Z86E03/E06 Functional Block Diagram
Table 1. 18-Pin DIP and SOIC Pin Identification

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>P24-27</td>
<td>Port 2, pins 4, 5, 6, 7</td>
<td>In/Output</td>
</tr>
<tr>
<td>5</td>
<td>Vcc</td>
<td>Power Supply</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>XTAL2</td>
<td>Crystal Oscillator Clock</td>
<td>Output</td>
</tr>
<tr>
<td>7</td>
<td>XTAL1</td>
<td>Crystal Oscillator Clock</td>
<td>Input</td>
</tr>
<tr>
<td>8-10</td>
<td>P31-33</td>
<td>Port 3, pins 1, 2, 3</td>
<td>Fixed Input</td>
</tr>
<tr>
<td>11-13</td>
<td>P34-36</td>
<td>Port 3, pins 4, 5, 6</td>
<td>Fixed Output</td>
</tr>
<tr>
<td>14</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>15-18</td>
<td>P20-23</td>
<td>Port 2, pins 0, 1, 2, 3</td>
<td>In/Output</td>
</tr>
</tbody>
</table>

Figure 2. 18-Pin DIP Pin Configuration

Figure 3. 18-Pin SOIC Pin Configuration
PIN FUNCTIONS

**XTAL1.** Crystal 1 (time-based input). This pin connects a parallel-resonant crystal, ceramic resonator, LC or RC network or an external single-phase clock to the on-chip oscillator input.

**XTAL2.** Crystal 2 (time-based output). This pin connects a parallel-resonant crystal, ceramic resonator, LC or RC network to the on-chip oscillator output.

**Port 2 (P27-P20).** Port 2 is an 8-bit, bidirectional, CMOS compatible I/O port. These eight I/O lines can be configured under software control to be an input or output, independently. Input buffers are Schmitt-triggered and contain Auto Latches. Bits programmed as outputs may be globally programmed as either push-pull or open-drain (Figure 4a., 4b., and 4c.). Low EMI output buffers can be globally programmed by the software. In addition, when the SPI is enabled, P20 functions as data-in (DI), and P27 functions as data-out (DO) for the SPI (SPI on the Z86E06 only).

![Port 2 Configuration (Z86E06)](image-url)

**Figure 4a. Port 2 Configuration (Z86E06)**
Figure 4b. Port 2 Configuration (Z86E06)
Figure 4c. Port 2 Configuration (Z86E03)
**Auto Latch.** The Auto Latch puts valid CMOS levels on all CMOS inputs (except P33, P32, P31) that are not externally driven. Whether this level is 0 or 1 cannot be determined. A valid CMOS level, rather than a floating node, reduces excessive supply current flow in the input buffer.

**Port 3 (P36-P31).** Port 3 is a 6-bit, CMOS compatible port. These six lines consist of three fixed inputs (P31-P33) and three fixed outputs (P34-P36). Pins P31, P32, and P33 are standard CMOS inputs (no auto latches) and pins P34, P35, and P36 are push-pull outputs. Low EMI output buffers can be globally programmed by the software. Two on-board comparators can process analog signals on P31 and P32 with reference to the voltage on P33. The analog function is enabled by programming Port 3 Mode Register (P3M-bit D1). Pins P31 and P32 are programmable as falling, rising, or both edge triggered interrupts (IRQ register bits 6 and 7). P33 is the comparator reference voltage input when the analog mode is selected. P33 is a falling edge interrupt input only.

**Note:** P33 is available as an interrupt input only in the digital mode. P31 and P32 are valid interrupt inputs and P31 is the T<sub>IN</sub> input when the analog or digital input mode is selected.

The outputs from the analog comparator can be globally programmed to output from P34 and P35 by setting PCON (F) 00 bit D0 = 1.

Access to Counter/Timer 1 is made through P31 (T<sub>IN</sub>) and P36 (T<sub>OUT</sub>).

In the Z86E06, pin P34 can also be configured as SPI clock (SK), input and output, and pin P35 can be configured as Slave Select (SS) in slave mode only, when the SPI is enabled (Figures 5a and 5b).

---

![Port 3 Configuration](image-url)

**Figure 5a. Port 3 Configuration**

---

*Z86E03/06 CMOS Z8<sup>®</sup> 8-Bit OTP Consumer Controller Processor*
**PIN FUNCTIONS (Continued)**

**Low EMI Emission.** The Z86E03/E06 can be programmed to operate in a low EMI emission mode in the PCON register. The oscillator and all I/O ports can be programmed as low EMI emission mode independently. Use of this feature results in:

- The pre-drivers slew rate reduced to 10 ns (typical).
- Low EMI output drivers resistance of 200 ohms (typical).
- Low EMI oscillator.

**Comparator Inputs.** Port 3 Pin P31 and P32 each have a comparator front end. The comparator reference voltage pin P33 is common to both comparators. In analog mode, the P31 and P32 are the positive inputs to the comparators, and P33 is the reference voltage supplied to both comparators. In digital mode, Pin P33 can be used as a P33 register input or IRQ1 source.

**Internal SCLK/TCLK = XTAL operation limited to a maximum of 4 MHz (250 ns cycle time) when the low EMI oscillator is selected and SCLK = External (SMR Register Bit D1=1).**
FUNCTIONAL DESCRIPTION

RESET. The device is reset in one of the following conditions:

- Power-On Reset
- Watch-Dog Timer
- STOP-Mode Recovery Source
- Low Voltage Protection

Having the Auto Power-On Reset circuitry built-in, the Z86E03/E06 does not require an external reset circuit. The reset time is 5 ms (typical) plus 18 clock cycles.

The device does not re-initialize the WDTMR, SMR, P2M, or P3M registers to their reset values on a STOP-Mode Recovery operation.

Program Memory. Z86E03/E06 can address up to 512/1K bytes of internal program memory (Figure 6). The first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts. Byte 13 to byte 511/1023 consists of on-chip, user program mask ROM.

EPROM Protect. The 512/1K bytes of Program Memory is mask programmable. A EPROM protect feature will prevent “dumping” of the EPROM contents by inhibiting execution of the LDC and LDCl instructions to program memory in all modes.

EPROM protect is EPROM-programmable. It is selected by the customer when the ROM code is submitted. Selecting ROM protect disables the LDC and LDCl instructions in all modes. ROM lookup tables are not supported in this mode.

Expanded Register File (ERF). The register file has been expanded to allow for additional system control registers and for mapping of additional peripheral devices and input/output ports into the register address area. The Z8 register address space R0 through R15 is implemented as 16 groups of 16 registers per group (Figure 7). These register groups are known as the Expanded Register File (ERF).

Bits 3-0 of the Register Pointer (RP) select the active ERF group. Bits 7-4 of the RP register select the working register group (Figure 7). For the Z86E03, three system configuration registers reside in the ERF address space Bank F. For the Z86E06, three system configuration registers reside in the ERF address space Bank F, while three SPI registers reside in Bank C. The rest of the ERF address space is not physically implemented and is open for future expansion.
FUNCTIONAL DESCRIPTION (Continued)

Z8 STANDARD CONTROL REGISTERS

REGISTER POINTERS

<table>
<thead>
<tr>
<th>Working Register</th>
<th>Expanded Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Pointer</td>
<td>Group Pointer</td>
</tr>
</tbody>
</table>

Z8 Reg. File

Not Implemented

FF
FO
DF
OF

REGISTER

<table>
<thead>
<tr>
<th>FF</th>
<th>SPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FE</td>
<td>GPR</td>
</tr>
<tr>
<td>FD</td>
<td>RP</td>
</tr>
<tr>
<td>FC</td>
<td>FLAGS</td>
</tr>
<tr>
<td>FB</td>
<td>IMR</td>
</tr>
<tr>
<td>FA</td>
<td>INQ</td>
</tr>
<tr>
<td>F9</td>
<td>IPR</td>
</tr>
<tr>
<td>F8</td>
<td>P01M</td>
</tr>
<tr>
<td>F7</td>
<td>P3M</td>
</tr>
<tr>
<td>F6</td>
<td>P2M</td>
</tr>
<tr>
<td>F5</td>
<td>Reserved</td>
</tr>
<tr>
<td>F4</td>
<td>Reserved</td>
</tr>
<tr>
<td>F3</td>
<td>PRE1</td>
</tr>
<tr>
<td>F2</td>
<td>T1</td>
</tr>
<tr>
<td>F1</td>
<td>TMR</td>
</tr>
<tr>
<td>F0</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Z8 STANDARD CONTROL REGISTERS

REGISTER

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF</td>
<td>FE</td>
<td>FD</td>
<td>FC</td>
<td>FB</td>
<td>FA</td>
<td>F9</td>
<td>F8</td>
</tr>
<tr>
<td>EE</td>
<td>PP</td>
<td>RP</td>
<td>IM</td>
<td>IN</td>
<td>IP</td>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>FF</td>
<td>FE</td>
<td>FD</td>
<td>FC</td>
<td>FB</td>
<td>FA</td>
<td>F9</td>
<td>F8</td>
</tr>
<tr>
<td>FF</td>
<td>FE</td>
<td>FD</td>
<td>FC</td>
<td>FB</td>
<td>FA</td>
<td>F9</td>
<td>F8</td>
</tr>
</tbody>
</table>

EXPANDED REG. GROUP (F)

REGISTER

<table>
<thead>
<tr>
<th>(F) 0F</th>
<th>WDTMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>(F) 0E</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F) 0D</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F) 0C</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F) 0B</td>
<td>SMR</td>
</tr>
<tr>
<td>(F) 0A</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F) 09</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F) 08</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F) 07</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F) 06</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F) 05</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F) 04</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F) 03</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F) 02</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F) 01</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F) 00</td>
<td>PCON</td>
</tr>
</tbody>
</table>

EXPANDED REG. GROUP (0)

REGISTER

<table>
<thead>
<tr>
<th>(0) 03</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0) 02</td>
<td>P2</td>
</tr>
<tr>
<td>(0) 01</td>
<td>Reserved</td>
</tr>
<tr>
<td>(0) 00</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Notes:
General-purpose registers are undefined after power-up, and they are not changed after reset.
* Will not be reset with a STOP-Mode Recovery.
U = Unknown
\( ^\dagger \) = Reserved

Figure 7a. Expanded Register File Architecture (Z86E03)
Z8 STANDARD CONTROL REGISTERS

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>RESET CONDITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF</td>
<td>U U U U U U U</td>
</tr>
<tr>
<td>FE</td>
<td>U U U U U U U</td>
</tr>
<tr>
<td>FD</td>
<td>0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>FC</td>
<td>U U U U U U U</td>
</tr>
<tr>
<td>FB</td>
<td>0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>FA</td>
<td>U U U U U U U</td>
</tr>
<tr>
<td>F9</td>
<td>U U U U U U U</td>
</tr>
<tr>
<td>F8</td>
<td>U U U U U U U</td>
</tr>
<tr>
<td>F7</td>
<td>U U U U U U U</td>
</tr>
<tr>
<td>F6</td>
<td>1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>F5</td>
<td>U U U U U U U</td>
</tr>
<tr>
<td>F4</td>
<td>U U U U U U U</td>
</tr>
<tr>
<td>F3</td>
<td>U U U U U U U</td>
</tr>
<tr>
<td>F2</td>
<td>U U U U U U U</td>
</tr>
<tr>
<td>F1</td>
<td>0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>F0</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

EXPANDED REG. GROUP (F)

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>RESET CONDITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>OF</td>
<td>U U U 0 1 0 0</td>
</tr>
<tr>
<td>OF</td>
<td>U U U 0 1 0 0</td>
</tr>
<tr>
<td>OF</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>OF</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>OF</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>OF</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>OD</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

EXPANDED REG. GROUP (C)

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>RESET CONDITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>O2</td>
<td>U U U U U C 0 0</td>
</tr>
<tr>
<td>O1</td>
<td>U U U U U U U U</td>
</tr>
<tr>
<td>O0</td>
<td>0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

EXPANDED REG. GROUP (O)

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>RESET CONDITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>O3</td>
<td>1 1 1 1 1 U U</td>
</tr>
<tr>
<td>O2</td>
<td>U U U U U U U U</td>
</tr>
<tr>
<td>O1</td>
<td>U U U U U U U U</td>
</tr>
<tr>
<td>O0</td>
<td>U U U U U U U U</td>
</tr>
</tbody>
</table>

Notes:
- General-purpose registers are undefined after power-up, and they are not changed after reset.
- * Will not be reset with a STOP-Mode Recovery
- ** Will not be reset with a STOP-Mode Recovery, except Bit D0.
- U = Unknown
- † = Reserved

Figure 7b. Expanded Register File Architecture (Z86E06)
FUNCTIONAL DESCRIPTION (Continued)

R253 RP

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
</table>

Expanded Register Group

Working Register Group

Note: Default Setting After Reset = 00000000

Figure 8. Register Pointer Register

Register File. The Register File consists of two I/O port registers, 60/124 general-purpose registers, and 13/15 control and status registers. The Z86E03 General-Purpose Register file ranges from address 00 to 3F while the Z86E06 General-Purpose Register file ranges from address 00 to 7F (see Figure 9). The instructions can access registers directly or indirectly via an 8-bit address field. This allows a short 4-bit register address using the Register Pointer (Figure 9). In the 4-bit mode, the Register File is divided into 16 working register groups, each occupying 16 contiguous locations. The Register Pointer addresses the starting location of the active working-register group.

General-Purpose Registers (GPR). These registers are undefined after the device is powered up. The registers keep their last value after any reset, as long as the reset occurs in the Vcc voltage-specified operating range. Note: Register R254 has been designated as a general-purpose register.

Stack. An 8-bit Stack Pointer (R255) used for the internal stack that resides within the 60/124 general-purpose registers.

Figures 9. Register Pointer

* Expanded Register Group (0) is selected in this figure by handling bits D3 to D0 as "0" in Register R253 (RP).
Counter/Timers. There are two 8-bit programmable counter/timers (T0-T1), each driven by its own 6-bit programmable prescaler (Z86E03 only has T1). The T1 prescaler can be driven by internal or external clock sources, however, the T0 prescaler is driven by the internal clock only (Figure 10).

* Available only in the Z86C06

Figure 10. Counter/Timer Block Diagram
FUNCTIONAL DESCRIPTION (Continued)

The 6-bit prescalers divide the input frequency of the clock source by any integer number from 1 to 64. Each prescaler drives its counter, which decrements the value (1 to 256) that has been loaded into the counter. When the counter reaches the end of count, a timer interrupt request, RQ4 (T0) or IRQ5 (T1), is generated. Note that IRQ4 is software-generated in the Z86E03.

The counters are programmed to start, stop, restart to continue, or restart from the initial value. The counters can also be programmed to stop upon reaching zero (single-pass mode) or to automatically reload the initial value and continue counting (modulo-n continuous mode).

The counters, but not the prescalers, are read at any time without disturbing their value or count mode. The clock source for T1 is user-definable and can be either the internal microprocessor clock divided by four, or an external signal input via Port 3. The Timer Mode register configures the external timer input (P31) as an external clock, a trigger input that can be retriggerable or non-retriggerable, or as a gate input for the internal clock. Port 3, line P36 serves as a timer output (T_out) through which T0 (E06 only), T1, or the internal clock can be output. The counter/timers can be cascaded by connecting the T0 output to the input of T1 (E06 only). The T_in mode is enabled by setting PRE1 bit D1 (R243) to 0.

Interrupts. The Z86E03/E06 has six different interrupts from six different sources. The interrupts are maskable and prioritized (Figure 11). The six sources are divided as follows; three sources are claimed by Port 3 lines P31-P33, two sources in the counter/timers, and one source for the SPI. The Interrupt Mask Register globally or singularly enables or disables the six interrupt requests (Table 2).

Figure 11. Interrupt Block Diagram
Table 2. Interrupt Types, Sources, and Vectors

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Vector Location</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRQ 0</td>
<td>IRQ 0</td>
<td>0, 1</td>
<td>External (P32), Rising/Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ 1</td>
<td>IRQ 1</td>
<td>2, 3</td>
<td>External (P33), Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ 2</td>
<td>IRQ 2, T_IN</td>
<td>4, 5</td>
<td>External (P31), Rising/Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ 3</td>
<td>IRQ 3</td>
<td>6, 7</td>
<td>Software Generated, SPI Receive</td>
</tr>
<tr>
<td>IRQ 4</td>
<td>TO/IRQ 4</td>
<td>8, 9</td>
<td>Internal for E06 and Software Generated for E03</td>
</tr>
<tr>
<td>IRQ 5</td>
<td>T_IN</td>
<td>10, 11</td>
<td>Internal</td>
</tr>
</tbody>
</table>

Note: When enabled, the SPI receive interrupt is mapped to IRQ3 in the Z86E06.

When more than one interrupt is pending, priorities are resolved by a programmable priority encoder that is controlled by the Interrupt Priority register. An interrupt machine cycle is activated when an interrupt request is granted. This disables all subsequent interrupts, saves the Program Counter and Status Flags, and then branches to the program memory vector location reserved for that interrupt. All Z86E03/E06 interrupts are vectored through locations in the program memory. This memory location and the next byte contain the 16-bit starting address of the interrupt service routine for that particular interrupt request.

To accommodate polled interrupt systems, interrupt inputs are masked and the interrupt request register is polled to determine which of the interrupt requests need service. In the Z86E06, when the SPI is disabled, IRQ3 has no hardware source but can be invoked by software (write to IRQ3 Register). When the SPI is enabled, an interrupt will be mapped to IRQ3 after a byte of data has been received by the SPI Shift Register.

An interrupt resulting from AN1 is mapped into IRQ2, and an interrupt from AN2 is mapped into IRQ0. Interrupts IRQ2 and IRQ0 may be rising, falling, or both edge triggered, and are programmable by the user. The software can poll to identify the state of the pin.

The programming bits for the INTERRUPT EDGE SELECT are located in the IRQ register (R250), bits D7 and D6. The configuration is shown in Table 3.

Table 3. IRQ Register

<table>
<thead>
<tr>
<th>IRQ</th>
<th>Interrupt Edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>D7</td>
<td>D6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Notes:
F = Falling Edge
R = Rising Edge
**FUNCTIONAL DESCRIPTION** (Continued)

**Clock.** The Z86E03/E06 on-chip oscillator has a high-gain, parallel-resonant amplifier for connection to a crystal, RC, ceramic resonator, or any suitable external clock source (XTAL1 = Input, XTAL2 = Output). The crystal should be AT cut, 10 kHz to 8 MHz/12 MHz max, with a series resistance (RS) less than or equal to 100 Ohms.

The crystal should be connected across XTAL1 and XTAL2 using the vendor’s recommended capacitor values (capacitance between 10 pF to 300 pF) from each pin directly to the device ground (pin 14). The layout is important to reduce ground noise injection.

The RC oscillator option is EPROM-programmable, to be selected by the customer at the time the Z8 is EPROM programmed. The RC oscillator configuration must be an external resistor connected from XTAL1 to XTAL2, with a frequency-setting capacitor from XTAL1 to ground (Figure 12).

In addition, a special feature has been incorporated into the Z86E03/E06; in low EMI noise mode (bit 7 of PCON register=0) with the RC option selected, the oscillator is targeted to consume considerably less Icc current at frequencies of 10 kHz or less.

<table>
<thead>
<tr>
<th>Ceramic Resonator or Crystal</th>
<th>LC</th>
<th>RC</th>
<th>External Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1, C2 = 47 pF TYP*</td>
<td>C1, C2 = 22 pF</td>
<td>C1 = 33 pF*</td>
<td></td>
</tr>
<tr>
<td>F = 8 MHz</td>
<td>L = 130 μH*</td>
<td>R = 1 kΩ*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>F = 3 MHz</td>
<td>F = 9.5 MHz</td>
<td></td>
</tr>
</tbody>
</table>

* Preliminary Value Including Pin Parasitics

**Figure 12. Oscillator Configuration**

**Power-On Reset.** A timer circuit clocked by a dedicated on-board RC oscillator is used for the Power-On Reset (POR) timer function. The POR time allows Vcc and the oscillator circuit to stabilize before instruction execution begins. The POR timer circuit is a one-shot timer triggered by one of the three conditions:

- Power-Fail to Power-OK Status
- STOP-Mode Recovery (if D5 of SMR=1)
- WDT Time-out

The POR time is a nominal 5 ms. Bit 5 of the STOP Mode Register determines whether the POR timer is bypassed after STOP mode recovery (typical for external clock, and RC/LC oscillators with fast start up time).

**HALT.** Will turn off the internal CPU clock but not the XTAL oscillation. The counter/timers and external interrupts IRQO, IRQ1, and IRQ2 remain active. The device may be recovered by interrupts either externally or internally generated.
STOP. This instruction turns off the internal clock and external crystal oscillation and reduces the standby current. The STOP mode is terminated by a RESET only, either by WDT time-out, POR, SPI compare, or SMR recovery. This causes the processor to restart the application program at address 0000C (HEX). Note, the crystal remains active in STOP mode if bits 3 and 4 of the WDTMR are enabled. In this mode, only the Watch-Dog Timer runs in STOP mode.

In order to enter STOP or HALT mode, it is necessary to first flush the instruction pipeline to avoid suspending execution in mid-instruction. To do this, the user executes a NOP (opcode=FFH) immediately before the appropriate SLEEP instruction, i.e.:

```
FF NOP ; clear the pipeline
6F STOP ; enter STOP mode
or
FF NOP ; clear the pipeline
7F HALT ; enter HALT mode
```

Serial Peripheral Interface (SPI)—Z86E06 Only. The Z86E06 incorporates a serial peripheral interface for communication with other microcontrollers and peripherals. **The SPI does not exist on the Z86E03.** The SPI includes features such as STOP-Mode Recovery, Master/Slave selection, and Compare mode. Table 4 contains the pin configuration for the SPI feature when it is enabled. The SPI consists of four registers: SPI Control Register (SCON), SPI Compare Register (SCOMP), SPI Receive/Buffer Register (RxBUF), and SPI Shift Register. SCON is located in bank (C) of the Expanded Register Group at address 02.

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
<th>Pin Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>DI</td>
<td>Data-In</td>
<td>P20</td>
</tr>
<tr>
<td>DO</td>
<td>Data-Out</td>
<td>P27</td>
</tr>
<tr>
<td>SS</td>
<td>Slave Select</td>
<td>P35</td>
</tr>
<tr>
<td>SK</td>
<td>SPI Clock</td>
<td>P34</td>
</tr>
</tbody>
</table>

The SPI Control Register (SCON) (Figure 13) is a read/write register that controls; Master/Slave selection, interrupts, clock source and phase selection, and error flag. Bit 0 enables/disables the SPI with the default being SPI disabled. A 1 in this location will enable the SPI, and a 0 will disable the SPI. Bits 1 and 2 of the SCON register in Master mode select the clock rate. The user may choose whether internal clock is divide-by-2, -4, -8, or -16. In slave mode, Bit 1 of this register flags the user if an overrun of the RxBUF Register has occurred. The RxCharOverrun flag is only reset by writing a 0 to this bit. In slave mode, bit 2 of the Control Register disables the data-out I/O function. If a 1 is written to this bit, the data-out pin is released to its original port configuration. If a 0 is written to this bit, the SPI shifts out one bit for each bit received. Bit 3 of the SCON Register enables the compare feature of the SPI, with the default being disabled. When the compare feature is enabled, a comparison of the value in the SCOMP Register is made with the value in the RxBUF Register. Bit 4 signals that a receive character is available in the RxBUF Register.

If the associated IRQ3 is enabled, an interrupt is generated. Bit 5 controls the clock phase of the SPI. A 1 in Bit 5 allows for receiving data on the clock's falling edge and transmitting data on the clock's rising edge. A 0 allows receiving data on the clock's rising edge and transmitting on the clock's falling edge. The SPI clock source is defined in bit 6. A 1 uses Timer0 output for the SPI clock, and a 0 uses TCLK for clocking the SPI. Finally, bit 7 determines whether the SPI is used as a Master or a Slave. A 1 puts the SPI into Master mode and a 0 puts the SPI into Slave mode.
FUNCTIONAL DESCRIPTION (Continued)

SPI Operation (Z86E06 only). The SPI is used in one of two modes: either as system slave, or as system master. Several of the possible system configurations are shown in Figure 14. In the slave mode, data transfer starts when the slave select (SS) pin goes active. Data is transferred into the slave's SPI Shift Register through the DI pin, which has the same address as the RxBUF Register. After a byte of data has been received by the SPI Shift Register, a Receive Character Available (RCA/IRQ3) flag and interrupt is generated. The next byte of data will be received at this time. The RxBUF Register must be cleared, or a Receive Character Overrun (RxCharOverrun) flag will be set in the SCON Register, and the data in the RxBUF Register will be overwritten. When the communication between the master and slave is complete, the SS goes inactive.

Unless disconnected, for every bit that is transferred into the slave through the DI pin, a bit is transferred out through the DO pin on the opposite clock edge. During slave operation, the SPI Clock pin (SK) is an input. In master mode, the CPU must first activate a SS through one of its I/O ports. Next, data is transferred through the master’s DO pin one bit per master clock cycle. Loading data into the shift register initiates the transfer. In master mode, the master’s clock will drive the slave’s clock. At the conclusion of a transfer, a Receive Character Available (RCA/ IRQ3) flag and interrupt is generated. Before data is transferred via the DO pin, the SPI Enable bit in the SCON Register must be enabled.

SPI Compare (Z86E06 only). When the SPI Compare Enable bit, D3 of the SCON Register is set to 1, the SPI Compare feature is enabled. The compare feature is only valid for slave mode. A compare transaction begins when the (SS) line goes active. Data is received as if it were a normal transaction, but there is no data transmitted to avoid bus contention with other slave devices. When the compare byte is received, IRQ3 is not generated. Instead, the data is compared with the contents of the SCOMP Register. If the data does not match, DO remains inactive and the slave ignores all data until the (SS) signal is reset. If the data received matches the data in the SCOMP register, then a SMR signal is generated. DO is activated if it is not tri-stated by D2 in the SCON Register, and data is received the same as any other SPI slave transaction.

When the SPI is activated as a slave, it operates in all system modes; STOP, HALT, and RUN. Slaves’ not comparing remain in their current mode, whereas slaves’ comparing wake from a STOP or HALT mode by means of an SMR.

SPI Clock (Z86E06 only). The SPI clock maybe driven by three sources: Timer0, a division of the internal system clock, or the external master when in slave mode. Bit D6 of the SCON Register controls what source drives the SPI clock. A 0 in bit D6 of the SCON Register determines the division of the internal system clock if this is used as the SPI clock source. Divide by 2, 4, 8, or 16 is chosen as the scaler.
Standard Serial Setup

Master

ss
sk
do
di

Slave

Slave

Slave

Slave

Standard Parallel Setup

Master

ss1
ss2
ss3
ss4
sk
sk
sk
sk

do
di
di
di
di

Slave

Slave

Slave

Slave

Setup For Compare

Master

ss
sk
do
di

(1)

(2)

(255)

(256)

Up to 256 slaves per SS line

Three Wire Compare Setup

Master

ss
sk
do
di

Slave

Slave

Slave

Slave

Multiple slaves may have the same address.

Figure 14. SPI System Configuration (Z86E06 Only)
Receive Character Available and Overrun (Z86E06 Only). When a complete data stream is received, an interrupt is generated and the RxCharAvail bit in the SCON Register is set. Bit 4 in the SCON Register is for enabling or disabling the RxCharAvail interrupt. The RxCharAvail bit is available for interrupt polling purposes and is reset when the RxBUF Register is read. RxCharAvail is generated in both master and slave modes. While in slave mode, if the RxBUF is not read before the next data stream is received and loaded into the RxBUF Register, Receive Character Overrun (RxCharOverrun) occurs. Since there is no need for clock control in slave mode, bit D1 in the SPI Control Register is used to log any RxCharOverrun (Figure 15 and Figure 16).

<table>
<thead>
<tr>
<th>No</th>
<th>Parameter</th>
<th>Min</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DI to SK Setup</td>
<td>10</td>
<td>ns</td>
</tr>
<tr>
<td>2</td>
<td>SK to D0 Valid</td>
<td>15</td>
<td>ns</td>
</tr>
<tr>
<td>3</td>
<td>SS to SK Setup</td>
<td>.5 Tsk</td>
<td>ns</td>
</tr>
<tr>
<td>4</td>
<td>SS to D0 Valid</td>
<td>15</td>
<td>ns</td>
</tr>
<tr>
<td>5</td>
<td>SK to DI Hold Time</td>
<td>10</td>
<td>ns</td>
</tr>
</tbody>
</table>

Figure 15. SPI Timing (Z86E06 Only)
**PORT Configuration Register (PCON).** The PCON configures the ports individually for comparator output on Port 3, low EMI noise on Ports 2 and 3, and low EMI noise oscillator. The PCON Register is located in the Expanded Register File at bank F, location 00 (Figure 17).

**Comparator Output Port 3 (DO).** Bit 0 controls the comparator use in Port 3. A 1 in this location brings the comparator outputs to P34, and P35 and a 0 releases the Port to its standard I/O configuration.

**Bits D4-D1.** These bits are reserved and must be 1.

**Low EMI Port 2 (D5).** Port 2 is configured as a Low EMI Port by resetting this bit (D5=0) or configured as a Standard Port by setting D5=1. The default value is 1.

**Low EMI Port 3 (D6).** Port 3 is configured as a Low EMI Port by resetting this bit (D6=0) or configured as a Standard Port by setting D6=1. The default value is 1.

**Low EMI OSC (D7).** This bit of the PCON Register controls the low EMI noise oscillator. A 1 in this location configures the oscillator with standard drive. While a 0 configures the oscillator with low noise drive, it does not affect the relationship of SCLK and XTAL.

**Figure 16. SPI Logic (Z86E06 Only)**

**Figure 17. Port Configuration Register (PCON) (Write Only)**
FUNCTIONAL DESCRIPTION (Continued)

STOP-Mode Recovery Register (SMR). This register selects the clock divide value and determines the mode of STOP-Mode Recovery (Figure 18). All bits are Write Only except bit 7, which is Read Only. Bit 7 is a flag bit that is hardware set on the condition of a STOP recovery and reset on a power-on cycle. Bit 6 controls whether a low level or high level is required from the recovery source. The recovery level must be active Low to work with SPI. Bit 5 controls the reset delay after recovery. Bits 2, 3, and 4 of the SMR specify the source of the STOP-Mode Recovery signal. Bit 1 determines whether the XTAL is divided by 1 or 2. A 0 in this location uses XTAL divide-by-two, and a 1 uses XTAL. The default for this bit is XTAL divide-by-two. Bit 0 controls the divide-by-16 prescaler of SCLK/TCLK. The SMR is located in bank F of the Expanded Register Group at address 0BH.

Figure 18a. STOP-Mode Recovery Register (Write Only except bit D7, which is Read Only.) (Z86E03)

<table>
<thead>
<tr>
<th>SMR (F) 0B</th>
<th>SCLK/TCLK Divide-by-16 Select (D0)—Z86E06 Only.</th>
<th>External Clock Divide Mode</th>
<th>Stop Mode Recovery Source</th>
<th>Stop Delay</th>
<th>Stop Recovery Level</th>
<th>Stop Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>D7 D6 D5 D4 D3 D2 D1 D0</td>
<td>SCLK/TCLK Divide-by-16</td>
<td>0 OFF **</td>
<td>1 ON</td>
<td>POR Only*</td>
<td>POR Only</td>
<td>POR Only</td>
</tr>
<tr>
<td></td>
<td>External Clock Divide Mode</td>
<td>0 = SCLK/TCLK = External/2*</td>
<td>1 = SCLK/TCLK = External/1</td>
<td>000</td>
<td>001-100</td>
<td>001</td>
</tr>
<tr>
<td></td>
<td>Stop Mode Recovery Source</td>
<td>000 POR Only*</td>
<td>001-100 Reserved</td>
<td>101 P27</td>
<td>110 P2 NOR 0:3</td>
<td>111 P2 NOR 0:7</td>
</tr>
<tr>
<td></td>
<td>Stop Delay</td>
<td>0 OFF</td>
<td>1 ON *</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stop Recovery Level</td>
<td>0 Low Level *</td>
<td>1 High Level</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stop Flag</td>
<td>0 POR*</td>
<td>1 Stop Recovery</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Default setting after RESET.
** Default setting after RESET, and STOP-Mode Recovery.

Figure 18b. STOP-Mode Recovery Register (Write Only except bit D7, which is Read Only.) (Z86E06)

SCLK/TCLK Divide-by-16 Select (D0)—Z86E06 Only. D0 of the SMR controls a divide-by-16 prescaler of SCLK/TCLK. The purpose of this control is to selectively reduce device power consumption during normal processor execution (SCLK control) and/or HALT mode (where TCLK sources the counter/timers and interrupt logic).

External Clock Divide Mode (D1). This bit can eliminate the oscillator divide-by-two circuitry. When this bit is 0, SCLK (System Clock) and TCLK (Timer Clock) are equal to the external clock frequency divided by two. The SCLK/TCLK is equal to the external clock frequency when this bit is set (D1=1). Using this bit, together with D7 of PCON, helps further lower EMI [i.e., D7 (PCON)=0, D1 (SMR)=1]. The default setting is 0.
STOP-Mode Recovery Source (D2,D3,D4). These three bits of the SMR specify the wake-up source of the STOP-Mode Recovery (Figure 19 and Table 5).

### Table 5. STOP-Mode Recovery Source

<table>
<thead>
<tr>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>Operation</th>
<th>Description of Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>POR recovery only</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>POR recovery only (E03 = Reserved)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>P31 transition (E03 = Reserved)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>P32 transition (E03 = Reserved)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>P33 transition (E03 = Reserved)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>P27 transition</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Logical NOR of Port 2 bits 0:3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Logical NOR of Port 2 bits 0:7</td>
</tr>
</tbody>
</table>

P31-P33 cannot wake up from STOP Mode if the input lines are configured as analog inputs. In the Z86E06, when the SPI is enabled and the Compare feature is active, a SMR is generated upon a comparison in the SPI Shift Register and SCOMP Register, regardless of the above SMR Register settings. If SPI Compare is used to wake up the part from STOP Mode, it is still possible to have one of the other STOP-Mode Recovery sources active. **Note:** These other STOP-Mode Recovery sources must be active level Low (bit D6 in SMR set to 0 if P31, P32, P33, and P27 selected, or bit D6 in SMR set to 1 if logical NOR of Port 2 is selected).

STOP-Mode Recovery Delay Select (D5). This bit disables the 5 ms RESET delay after STOP-Mode Recovery. The default condition of this bit is 1. If the "fast" wake up is selected, the STOP-Mode Recovery source needs to be kept active for at least 5 TpC.

STOP-Mode Recovery Level Select (D6). A 1 in this bit position indicates that a high level on any one of the recovery sources wakes the device from STOP Mode. A 0 indicates low level recovery. The default is 0 on POR (Figure 19).

Cold or Warm Start (D7). This bit is set by the device upon entering STOP Mode. It is active High, and is 0 (cold) on POR/WDT RESET. This bit is Read Only. A 1 in this bit (warm) indicates that the device awakens by a SMR source.

Stop Mode Recovery Edge Select (SMR)

![Stop Mode Recovery Edge Select (SMR)](image)

**Figure 19a. STOP Mode Recovery Source (Z86E03)**
FUNCTIONAL DESCRIPTION (Continued)

Watch-Dog Timer Mode Register (WDTMR). The WDT is a retriggerable one-shot timer that resets the Z8 if it reaches its terminal count. The WDT is initially enabled by executing the WDT instruction and refreshed on subsequent executions of the WDT instruction. The WDT cannot be disabled after it has been initially enabled. The WDT circuit is driven by an on-board RC oscillator or external oscillator from XTAL1 pin. The POR clock source is selected with bit 4 of the WDTMR register.

Note: Execution of the WDT instruction affects the Z (zero), S (sign), and V (overflow) flags.

Bits 0 and 1 control a tap circuit that determines the timeout period (on Z86E06 only). Bit 2 determines whether the WDT is active during HALT and bit 3 determines WDT activity during STOP. If bits 3 and 4 of this register are both set to 1, the WDT is only driven by the external clock during STOP mode. This feature makes it possible to wake up from STOP mode from an internal source. Bits 5 through 7 of the WDTMR are reserved (Figure 20). This register is accessible only during the first 64 internal system clock cycles from the execution of the first instruction after Power-On Reset, Watch-Dog Reset or a STOP Mode Recovery (Figure 21). After this point, the register cannot be modified by any means, intentional or otherwise. The WDTMR cannot be read and is located in bank F of the Expanded Register Group at address location 0FH.

Figure 20. Watch-Dog Timer Mode Register (Write Only)
* Not available on the Z86E03, WDT fixed at 15 ms/1024TpC in the Z86E03.

Figure 21. Resets and WDT
FUNCTIONAL DESCRIPTION (Continued)

WDT Time Select (D1,D0). Bits 0 and 1 control a tap circuit that determines the time-out period. Table 6 shows the different values that can be obtained. The default value of D0 and D1 are 1 and 0, respectively. These select bits are present in the Z86E06 only.

Table 6. Time-Out Period of the WDT (Z86E06 Only)

<table>
<thead>
<tr>
<th>D1</th>
<th>D0</th>
<th>Time-Out of Internal RC OSC</th>
<th>Time-Out of XTLAL Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>5 ms min</td>
<td>256Tpd</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>15 ms min</td>
<td>512Tpd</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>25 ms min</td>
<td>1024Tpd</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>100 ms min</td>
<td>4096Tpd</td>
</tr>
</tbody>
</table>

Notes:
Tpd = XTAL clock cycle
The default on reset is 15 ms, D0 = 1 and D1 = 0.
The values given are for Vcc = 5.0V

For the Z86E03, the WDT time-out value is fixed at 1024 Tpd (depending on WDTMR bit D4) period. When writing to the WDTMR in the Z86E03, bit D0 must be 1 and D1 must be 0.

WDT During HALT (D2). This bit determines whether or not the WDT is active during HALT mode. A 1 indicates active during HALT. The default is 1.

WDT During STOP (D3). This bit determines whether or not the WDT is active during STOP mode. Since XTAL clock is stopped during STOP Mode, unless as specified below, the on-board RC must be selected as the clock source to the POR counter. A 1 indicates active during STOP. The default is 1. If bits D3 and D4 are both set to 1, the WDT only, is driven by the external clock during STOP mode.

Clock Source for WDT (D4). This bit determines which oscillator source is used to clock the internal POR and WDT counter chain. If the bit is a 1, the internal RC oscillator is bypassed and the POR and WDT clock source is driven from the external pin, XTAL1. The default configuration of this bit is 0, which selects the internal RC oscillator.

Bits 5, 6 and 7. These bits are reserved.

Vcc Voltage Comparator. An on-board Voltage Comparator checks that Vcc is at the required level to ensure correct operation of the device. Reset is globally driven if Vcc is below the specified voltage (typically 2.6V).

Low Voltage Protection (Vlv). The Low Voltage Protection trip point (Vlv) will be less than 3 volts and above 1.8 volts under the following conditions.

Maximum (Vlv) Conditions:
Case 1: \( T_A = -40^\circ \text{C to } +105^\circ \text{C}, \text{ Internal Clock (SCLK) Frequency equal or less than 1 MHz} \)
Case 2: \( T_A = -40^\circ \text{C to } +85^\circ \text{C}, \text{ Internal Clock (SCLK) Frequency equal or less than 2 MHz} \)

Note: The internal clock frequency (SCLK) is determined by SMR (F) 0BH bit D1.

The device functions normally at or above 3.0V under all conditions. Below 3.0V, the device is guaranteed to function normally until the Low Voltage Protection trip point (Vlv) is reached, for the temperatures and operating frequencies in cases 1 and 2 above. The actual low voltage trip point is a function of temperature and process parameters (Figure 22).
Note: * The typical minimum operating Vcc voltage at that frequency.

Figure 22. Typical Z86E03/E06 V_LV Voltage vs Temperature
SPECIAL FUNCTIONS

EPROM Mode

Besides $V_{cc}$ and GND ($V_{ss}$), the Z86E03/E06 changes all its pin functions in the EPROM mode. XTAL2 has no function, XTAL1 functions as /CE, P31 functions as /OE, P32 functions as EPM, P33 functions as $V_{pp}$, and P02 functions as /PGM.

EPROM Protect. ROM protect is EPROM-programmable. It is selected by the customer at the time the ROM code is EPROM programmed. The selection of ROM Protect disables the LDC and LDCI instructions in all modes. A ROM look-up table cannot be used in this mode.

Application Caution

Please note that when using the device in a noisy environment, it is suggested that the voltages on the EPM, /CE, /OE pins be clamped to $V_{cc}$ through a diode to $V_{cc}$ to prevent accidentally entering the OTP mode. The $V_{pp}$ requires both a diode and a 100 pF capacitor.

User Modes. Table 7 shows the programming voltage of each mode of Z86E06.

<table>
<thead>
<tr>
<th>Programming Modes</th>
<th>$V_{pp}$</th>
<th>EPM</th>
<th>/CE</th>
<th>/OE</th>
<th>/PGM</th>
<th>ADDR</th>
<th>DATA</th>
<th>$V_{cc}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPROM READ1</td>
<td>X</td>
<td>$V_H$</td>
<td>$V_L$</td>
<td>$V_L$</td>
<td>$V_H$</td>
<td>ADDR</td>
<td>Out</td>
<td>4.5V</td>
</tr>
<tr>
<td>EPROM READ2</td>
<td>X</td>
<td>$V_H$</td>
<td>$V_L$</td>
<td>$V_L$</td>
<td>$V_H$</td>
<td>ADDR</td>
<td>Out</td>
<td>5.5V</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>$V_H$</td>
<td>X</td>
<td>$V_L$</td>
<td>$V_L$</td>
<td>$V_H$</td>
<td>ADDR</td>
<td>In</td>
<td>6.0V</td>
</tr>
<tr>
<td>PROGRAM VERIFY</td>
<td>$V_H$</td>
<td>X</td>
<td>$V_L$</td>
<td>$V_L$</td>
<td>$V_H$</td>
<td>ADDR</td>
<td>Out</td>
<td>6.0V</td>
</tr>
<tr>
<td>EPROM PROTECT</td>
<td>$V_H$</td>
<td>$V_H$</td>
<td>$V_H$</td>
<td>$V_H$</td>
<td>$V_L$</td>
<td>NU</td>
<td>NU</td>
<td>6.0V</td>
</tr>
<tr>
<td>PERMANENT WDT ENABLED</td>
<td>$V_H$</td>
<td>$V_H$</td>
<td>$V_H$</td>
<td>$V_H$</td>
<td>$V_L$</td>
<td>NU</td>
<td>NU</td>
<td>6.0V</td>
</tr>
<tr>
<td>GLOBAL AUTO LATCH DISABLED</td>
<td>$V_H$</td>
<td>$V_H$</td>
<td>$V_H$</td>
<td>$V_L$</td>
<td>$V_L$</td>
<td>NU</td>
<td>NU</td>
<td>6.0V</td>
</tr>
<tr>
<td>RC OSCILLATOR</td>
<td>$V_H$</td>
<td>$V_H$</td>
<td>$V_H$</td>
<td>$V_L$</td>
<td>$V_L$</td>
<td>NU</td>
<td>NU</td>
<td>6.0V</td>
</tr>
</tbody>
</table>

Notes:

In EPROM Mode, all Z8 inputs are TTL inputs.

$V_H$ = 12.5V ± 0.5V

$V_{pp}$ = As per specific Z8 DC specification.

$V_{il}$ = As per specific Z8 DC specification.

X = Not used, but must be set to $V_H$, $V_{pp}$, or $V_L$ level.

NU = Not used, but must be set to either $V_H$, $V_{pp}$, or $V_L$ level.

$\text{I}_{\text{cc}}$ during programming = 40 mA maximum.

$\text{I}_{\text{cc}}$ during programming, verify, or read = 40 mA maximum.

$V_{cc}$ has a tolerance of ±0.25V.
**Internal Address Counter.** The address of Z86E03/E06 is generated internally with a counter clocked through pin P01 (Clock). Each clock signal increases the address by one and the high level of pin P00 (Clear) will reset the address to zero. Figure 16 shows the setup time of the serial address input.

**Programming Waveform.** Figures 24, 25 and 26 show the programming waveforms of each mode. Table 7 shows the timing of programming waveforms.

**Programming Algorithm.** Figure 27 shows the flow chart of the Z86E03/E06 programming algorithm.

### Table 8. Timing of Programming Waveforms

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Name</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Address Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>2</td>
<td>Data Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>3</td>
<td>V&lt;sub&gt;pp&lt;/sub&gt; Setup</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>4</td>
<td>V&lt;sub&gt;cc&lt;/sub&gt; Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>5</td>
<td>Chip Enable Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>6</td>
<td>Program Pulse Width</td>
<td>0.95</td>
<td></td>
<td>ms</td>
</tr>
<tr>
<td>7</td>
<td>Data Hold Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>8</td>
<td>/OE Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>9</td>
<td>Data Access Time</td>
<td></td>
<td>200</td>
<td>ns</td>
</tr>
<tr>
<td>10</td>
<td>Data Output Float Time</td>
<td></td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td>11</td>
<td>Overprogram Pulse Width</td>
<td></td>
<td>2.85</td>
<td>ms</td>
</tr>
<tr>
<td>12</td>
<td>EPM Setup Time</td>
<td></td>
<td>2</td>
<td>µs</td>
</tr>
<tr>
<td>13</td>
<td>/PGM Setup Time</td>
<td></td>
<td>2</td>
<td>µs</td>
</tr>
<tr>
<td>14</td>
<td>Address to /OE Setup Time</td>
<td></td>
<td>2</td>
<td>µs</td>
</tr>
<tr>
<td>15</td>
<td>Option Program Pulse Width</td>
<td></td>
<td>78</td>
<td>ms</td>
</tr>
</tbody>
</table>
SPECIAL FUNCTIONS (Continued)
EPROM Mode

Legend:
- T1 Reset Clock Width
- T2 Input Clock High
- T3 Input Clock Period
- T4 Input Clock Low
- T5 Clock to Address Counter Out Delay

Figure 23. Z86E03/E06 Address Counter Waveform
Figure 24. Z86E03/E07 Programming Waveform (EPROM Read)
SPECIAL FUNCTIONS (Continued)

EPROM Mode

Figure 25. Z86E03/E06 Programming Waveform (Program and Verify)
Figure 26. Z86E03/E06 Programming Waveform
(EPROM Protect and Low EMI Program)
SPECIAL FUNCTIONS (Continued)

EPROM Mode

Figure 27. Z86E03/E06 Programming Algorithm

Note:
- To ensure proper operations during the spec., Zilog recommends verification over the $V_{CC}$ range of the device $V_{CC}$ spec.
ABSOLUTE MAXIMUM RATINGS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{CC}$</td>
<td>Supply Voltage*</td>
<td>-0.3</td>
<td>+7.0</td>
<td>V</td>
</tr>
<tr>
<td>$V_{IH}$</td>
<td>Max Input Voltage**</td>
<td>12</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$T_{STG}$</td>
<td>Storage Temp</td>
<td>-65</td>
<td>+150</td>
<td>°C</td>
</tr>
<tr>
<td>$T_A$</td>
<td>Op Amp Ambient Temp</td>
<td>†</td>
<td></td>
<td>°C</td>
</tr>
</tbody>
</table>

Notes:
* Voltage on all pins with respect to GND.
** Applies to Port pins only and must limit current going into or out of Port pins to 250 µA maximum.
† See Ordering Information

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

STANDARD TEST CONDITIONS

The characteristics listed below apply for standard test conditions as noted. All voltages are referenced to Ground. Positive current flows into the referenced pin (Figure 28).

CAPACITANCE

$T_A = 25^\circ C, V_{CC} = GND = 0V, f = 1.0 MHz, unmeasured pins returned to GND.$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Capacitance</td>
<td>0</td>
<td>12 pF</td>
</tr>
<tr>
<td>Output Capacitance</td>
<td>0</td>
<td>20 pF</td>
</tr>
<tr>
<td>I/O Capacitance</td>
<td>0</td>
<td>25 pF</td>
</tr>
</tbody>
</table>

$V_{CC}$ SPECIFICATION

$V_{CC} = 3.0V$ to $5.5V$
## DC ELECTRICAL CHARACTERISTICS

**Z86E03/E06**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$ Note [3]</th>
<th>$T_A = 0°C$ to $+70°C$</th>
<th>$T_A = -40°C$ to $+105°C$</th>
<th>Typical @ 25°C</th>
<th>Units</th>
<th>Conditions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{CH}$</td>
<td>Clock Input High Voltage</td>
<td>3.3V</td>
<td>0.9 $V_{cc}$ $V_{cc}+0.3$</td>
<td>0.9 $V_{cc}$ $V_{cc}+0.3$</td>
<td>2.4</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>0.9 $V_{cc}$ $V_{cc}+0.3$</td>
<td>0.9 $V_{cc}$ $V_{cc}+0.3$</td>
<td>3.9</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
<td></td>
</tr>
<tr>
<td>$V_{CL}$</td>
<td>Clock Input Low Voltage</td>
<td>3.3V</td>
<td>$V_{ss}-0.3$ 0.2 $V_{cc}$</td>
<td>$V_{ss}-0.3$ 0.2 $V_{cc}$</td>
<td>1.6</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>$V_{ss}-0.3$ 0.2 $V_{cc}$</td>
<td>$V_{ss}-0.3$ 0.2 $V_{cc}$</td>
<td>2.7</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
<td></td>
</tr>
<tr>
<td>$V_{IH}$</td>
<td>Input High Voltage</td>
<td>3.3V</td>
<td>0.7 $V_{cc}$ $V_{cc}+0.3$</td>
<td>0.7 $V_{cc}$ $V_{cc}+0.3$</td>
<td>1.8</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>0.7 $V_{cc}$ $V_{cc}+0.3$</td>
<td>0.7 $V_{cc}$ $V_{cc}+0.3$</td>
<td>2.8</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{IL}$</td>
<td>Input Low Voltage</td>
<td>3.3V</td>
<td>$V_{ss}-0.3$ 0.2 $V_{cc}$</td>
<td>$V_{ss}-0.3$ 0.2 $V_{cc}$</td>
<td>1.0</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>$V_{ss}-0.3$ 0.2 $V_{cc}$</td>
<td>$V_{ss}-0.3$ 0.2 $V_{cc}$</td>
<td>1.5</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{OH}$</td>
<td>Output High Voltage</td>
<td>3.3V</td>
<td>$V_{cc}-0.4$ $V_{cc}+0.4$</td>
<td>$V_{cc}-0.4$ $V_{cc}+0.4$</td>
<td>3.1</td>
<td>V</td>
<td>$I_{OH} = -2.0$ mA [10]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>$V_{cc}-0.4$ $V_{cc}+0.4$</td>
<td>$V_{cc}-0.4$ $V_{cc}+0.4$</td>
<td>4.8</td>
<td>V</td>
<td>$I_{OH} = -2.0$ mA [10]</td>
<td></td>
</tr>
<tr>
<td>$V_{OL1}$</td>
<td>Output Low Voltage</td>
<td>3.3V</td>
<td>0.8</td>
<td>0.8</td>
<td>0.2</td>
<td>V</td>
<td>$I_{OH} = +4.0$ mA [10]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>0.4</td>
<td>0.4</td>
<td>0.1</td>
<td>V</td>
<td>$I_{OH} = +4.0$ mA [10]</td>
<td></td>
</tr>
<tr>
<td>$V_{OL2}$</td>
<td>Output Low Voltage</td>
<td>3.3V</td>
<td>1.0</td>
<td>1.0</td>
<td>0.4</td>
<td>V</td>
<td>$I_{OL} = +6$ mA, 3 Pin Max</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>1.0</td>
<td>1.0</td>
<td>0.5</td>
<td>V</td>
<td>$I_{OL} = +12$ mA, 3 Pin Max</td>
<td></td>
</tr>
<tr>
<td>$V_{OFFSET}$</td>
<td>Comparator Input Offset Voltage</td>
<td>3.3V</td>
<td>±10</td>
<td>±10</td>
<td>±5</td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>±10</td>
<td>±10</td>
<td>±5</td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{CCR}$</td>
<td>Input Common Mode Voltage Range</td>
<td>3.3V</td>
<td>OV</td>
<td>$V_{cc} - 1.0V$</td>
<td>$V_{cc} - 1.0V$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>OV</td>
<td>$V_{cc} - 1.5V$</td>
<td>$V_{cc} - 1.5V$</td>
<td></td>
<td></td>
<td>[7]</td>
</tr>
<tr>
<td>$I_{IL}$</td>
<td>Input Leakage</td>
<td>3.3V</td>
<td>−1.0</td>
<td>1.0</td>
<td>−1.0</td>
<td>1.0</td>
<td>μA</td>
<td>$V_{IN} = 0V, V_{CC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>−1.0</td>
<td>1.0</td>
<td>−1.0</td>
<td>1.0</td>
<td>μA</td>
<td>$V_{IN} = 0V, V_{CC}$</td>
</tr>
<tr>
<td>$I_{OL}$</td>
<td>Output Leakage</td>
<td>3.3V</td>
<td>−1.0</td>
<td>1.0</td>
<td>−1.0</td>
<td>1.0</td>
<td>μA</td>
<td>$V_{IN} = 0V, V_{CC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>−1.0</td>
<td>1.0</td>
<td>−1.0</td>
<td>1.0</td>
<td>μA</td>
<td>$V_{IN} = 0V, V_{CC}$</td>
</tr>
<tr>
<td>$I_{CC}$</td>
<td>Supply Current</td>
<td>3.3V</td>
<td>6</td>
<td>6</td>
<td>3.0</td>
<td>mA</td>
<td>@ 8 MHz</td>
<td>[4,5,12]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>11.0</td>
<td>11.0</td>
<td>6.0</td>
<td>mA</td>
<td>@ 8 MHz</td>
<td>[4,5,12]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.3V</td>
<td>8.0</td>
<td>8.0</td>
<td>4.5</td>
<td>mA</td>
<td>@ 12 MHz</td>
<td>[4,5,10,13]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>15</td>
<td>15</td>
<td>9.0</td>
<td>mA</td>
<td>@ 12 MHz</td>
<td>[4,5,10,13]</td>
</tr>
<tr>
<td>$I_{IB}$</td>
<td>Input Bias Current</td>
<td>3.3V</td>
<td>300</td>
<td>300</td>
<td></td>
<td>nA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>300</td>
<td>300</td>
<td></td>
<td>nA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{IO}$</td>
<td>Input Offset Current</td>
<td>3.3V</td>
<td>+150</td>
<td>+150</td>
<td></td>
<td>nA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>+150</td>
<td>+150</td>
<td></td>
<td>nA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Z86E03/E06 CMOS 8-Bit OTP Consumer Controller Processor

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$ to +70°C</th>
<th>$T_s = 0°C$ to +70°C</th>
<th>$T_s = -40°C$ to +105°C</th>
<th>Typical Units</th>
<th>Conditions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
<td>@ 25°C</td>
<td></td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>P36</td>
<td>5.0V</td>
<td>1.0</td>
<td>1.0</td>
<td>V</td>
<td></td>
<td>$I_{OL} = 24$ mA</td>
</tr>
<tr>
<td>$I_{CC1}$</td>
<td>Standby Current</td>
<td>3.3V</td>
<td>3.0</td>
<td>3.0</td>
<td>1.3</td>
<td>mA</td>
<td>HALT Mode $V_{in} = OV$, $V_{CC} \geq 8$ MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>5.0</td>
<td>5.0</td>
<td>3.0</td>
<td>mA</td>
<td>HALT Mode $V_{in} = OV$, $V_{CC} \geq 8$ MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.3V</td>
<td>4.5</td>
<td>4.5</td>
<td>2.0</td>
<td>mA</td>
<td>HALT Mode $V_{in} = OV$, $V_{CC} \geq 12$ MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>7.0</td>
<td>7.0</td>
<td>4.0</td>
<td>mA</td>
<td>HALT Mode $V_{in} = OV$, $V_{CC} \geq 12$ MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.3V</td>
<td>1.4</td>
<td>1.4</td>
<td>0.7</td>
<td>mA</td>
<td>Clock Divide-by-16 @ 8 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>3.5</td>
<td>3.5</td>
<td>0.7</td>
<td>mA</td>
<td>Clock Divide-by-16 @ 8 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.3V</td>
<td>2.0</td>
<td>2.0</td>
<td>1.0</td>
<td>mA</td>
<td>Clock Divide-by-16 @ 12 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>4.5</td>
<td>4.5</td>
<td>2.5</td>
<td>mA</td>
<td>Clock Divide-by-16 @ 12 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>mA</td>
<td>HALT Mode @ 12 kHz</td>
</tr>
<tr>
<td>$I_{CC2}$</td>
<td>Standby Current</td>
<td>3.0V</td>
<td>10</td>
<td>20</td>
<td>1.0</td>
<td>$\mu$A</td>
<td>STOP Mode $V_{in} = OV$, $V_{CC}$ WDT is not Running</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>20</td>
<td>40</td>
<td>3.0</td>
<td>$\mu$A</td>
<td>STOP Mode $V_{in} = OV$, $V_{CC}$ WDT is not Running</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.3V</td>
<td>600</td>
<td>600</td>
<td>400</td>
<td>$\mu$A</td>
<td>STOP Mode $V_{in} = OV$, $V_{CC}$ WDT is Running</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>1000</td>
<td>1000</td>
<td>800</td>
<td>$\mu$A</td>
<td>STOP Mode $V_{in} = OV$, $V_{CC}$ WDT is Running</td>
</tr>
<tr>
<td>$I_{ALL}$</td>
<td>Auto Latch Low Current</td>
<td>3.3V</td>
<td>7.0</td>
<td>14.0</td>
<td>4.0</td>
<td>$\mu$A</td>
<td>$OV &lt; V_{in} &lt; V_{CC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>20.0</td>
<td>30.0</td>
<td>10</td>
<td>$\mu$A</td>
<td>$OV &lt; V_{in} &lt; V_{CC}$</td>
</tr>
<tr>
<td>$I_{ALH}$</td>
<td>Auto Latch High Current</td>
<td>3.3V</td>
<td>4.0</td>
<td>8.0</td>
<td>2.0</td>
<td>$\mu$A</td>
<td>$OV &lt; V_{in} &lt; V_{CC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>9.0</td>
<td>16.0</td>
<td>5.0</td>
<td>$\mu$A</td>
<td>$OV &lt; V_{in} &lt; V_{CC}$</td>
</tr>
<tr>
<td>$T_{POR}$</td>
<td>Power-On Reset</td>
<td>3.3V</td>
<td>7</td>
<td>24</td>
<td>25</td>
<td>13</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>3</td>
<td>13</td>
<td>14</td>
<td>7</td>
<td>ms</td>
</tr>
<tr>
<td>$V_{LV}$</td>
<td>V_{cc} Low Voltage</td>
<td>2.2</td>
<td>2.8</td>
<td>1.7</td>
<td>3.0</td>
<td>2.6</td>
<td>V</td>
</tr>
</tbody>
</table>

### Notes:

1. Clock Driven on XTAL Type: $0.3 \leq 5.0$ $mA$ $8$ MHz
2. Crystal or Ceramic Resonator Type: $3.0 \leq 5.0$ $mA$ $8$ MHz
3. $V_{CC} = 3.0 \text{V to } 5.5 \text{V}$. The $V_{LV}$ increases as the temperature decreases. Typical values measured at $3.3 \text{V}$ and $5.0 \text{V}$.
4. All outputs unloaded, I/O pins floating, inputs at rail.
5. $C_{L} = 100 \text{pF}$
7. For analog comparator inputs when analog comparators are enabled.
8. Excludes clock pins.
9. Clock must be forced Low when XTAL1 is clock-driven and XTAL2 is floating.
10. STD mode (not low EMI mode).
11. Low EMI Oscillator enabled.
12. Z86E03 only.
13. Z86E06 only.
### AC ELECTRICAL CHARACTERISTICS

**Z86E03/E06**

#### Figure 29. Additional Timing

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$</th>
<th>$T_{A} = 0°C$ To $+70°C$</th>
<th>$T_{A} = -40°C$ To $+105°C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TpC</td>
<td>Input Clock Period</td>
<td>3.0V</td>
<td>125 DC, 83 DC</td>
<td>125 DC, 83 DC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>125 DC, 83 DC</td>
<td>125 DC, 83 DC</td>
</tr>
<tr>
<td>2</td>
<td>TrC,TfC</td>
<td>Clock Input Rise</td>
<td>3.0V</td>
<td>25, 15</td>
<td>25, 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and Fall Times</td>
<td>5.5V</td>
<td>25, 15</td>
<td>25, 15</td>
</tr>
<tr>
<td>3</td>
<td>TwC</td>
<td>Input Clock Width</td>
<td>3.0V</td>
<td>62, 41</td>
<td>62, 41</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>62, 41</td>
<td>62, 41</td>
</tr>
<tr>
<td>4</td>
<td>TwTinL</td>
<td>Timer Input Low Width</td>
<td>3.0V</td>
<td>100, 100</td>
<td>100, 100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70, 70</td>
<td>70, 70</td>
</tr>
<tr>
<td>5</td>
<td>TwTinH</td>
<td>Timer Input High Width</td>
<td>3.0V</td>
<td>5TpC, 5TpC, 5TpC</td>
<td>5TpC, 5TpC, 5TpC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC, 5TpC, 5TpC</td>
<td>5TpC, 5TpC, 5TpC</td>
</tr>
</tbody>
</table>
## AC ELECTRICAL CHARACTERISTICS (Continued)
(SCLK/TCLK = EXTERNAL/2)

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$ Note [3]</th>
<th>$T_A = 0^\circ C$ TO $+70^\circ C$</th>
<th>$T_A = -40^\circ C$ TO $+105^\circ C$</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>8 MHz$^{(1)}$</td>
<td>12 MHz$^{(1)}$</td>
<td>8 MHz$^{(1)}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>TpTin</td>
<td>Timer Input Period</td>
<td>3.0V</td>
<td>8TpC</td>
<td>8TpC</td>
<td></td>
<td>[1,7,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>8TpC</td>
<td>8TpC</td>
<td></td>
<td>[1,7,8]</td>
</tr>
<tr>
<td>7</td>
<td>TrTin,</td>
<td>Timer Input Rise and Fall Timer</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td>TtTin</td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td>8</td>
<td>TwIL</td>
<td>Int. Request Input Low Time</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>ns</td>
</tr>
<tr>
<td>9</td>
<td>TwIH</td>
<td>Int. Request Input High Time</td>
<td>3.0V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Twsm</td>
<td>STOP Mode Recovery Width Spec</td>
<td>3.0V</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Tost</td>
<td>Oscillator Startup Time</td>
<td>3.0V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Twdt</td>
<td>Watch-Dog Timer Refresh Time</td>
<td>3.0V</td>
<td>15</td>
<td>15</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.0V</td>
<td>30</td>
<td>30</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>16</td>
<td>16</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.0V</td>
<td>60</td>
<td>60</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.0V</td>
<td>250</td>
<td>250</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>120</td>
<td>120</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

**Notes:**

[1] Timing Reference uses 0.7 $V_{cc}$ for a logic 1 and 0.2 $V_{cc}$ for a logic 0.
[2] Interrupt request via Port 3 (P31-P33).
[3] $V_{cc}$ = 3.0V to 5.5V.
[4] SMR-D5 = 0, POR delay is off.
[5] WDTMR Register
[6] Internal RC Oscillator only.
[7] SMR D1 = 0, SCLK = External/2
[8] Maximum frequency for internal system clock is 4 MHz when using SCLK = EXTERNAL clock mode.
[10] SMR-D5 = 1, STOP-Mode Recovery delay is on.
## AC ELECTRICAL CHARACTERISTICS

Additional Timing Table (Divide-By-One Mode, SCLK/TCLK = EXTERNAL)

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$</th>
<th>4 MHz</th>
<th>4 MHz</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$T_A = 0^\circ C$ to $+70^\circ C$</td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T_A = -40^\circ C$ to $+105^\circ C$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TpC</td>
<td>Input Clock Period</td>
<td>3.0V</td>
<td>250</td>
<td>DC</td>
<td>250</td>
<td>DC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>250</td>
<td>DC</td>
<td>250</td>
<td>DC</td>
</tr>
<tr>
<td>2</td>
<td>TrC,TfC</td>
<td>Clock Input Rise &amp; Fall Times</td>
<td>3.0V</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>ns</td>
</tr>
<tr>
<td>3</td>
<td>TwC</td>
<td>Input Clock Width</td>
<td>3.0V</td>
<td>125</td>
<td></td>
<td>125</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>125</td>
<td></td>
<td>125</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TwTinL</td>
<td>Timer Input Low Width</td>
<td>3.0V</td>
<td>100</td>
<td></td>
<td>100</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td></td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>TwTinH</td>
<td>Timer Input High Width</td>
<td>3.0V</td>
<td>3TpC</td>
<td></td>
<td>3TpC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>3TpC</td>
<td></td>
<td>3TpC</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>TpTin</td>
<td>Timer Input Period</td>
<td>3.0V</td>
<td>4TpC</td>
<td></td>
<td>4TpC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>4TpC</td>
<td></td>
<td>4TpC</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TrTin</td>
<td>Timer Input Rise &amp; Fall Timer</td>
<td>3.0V</td>
<td>100</td>
<td></td>
<td>100</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TtTin</td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td></td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>TwIL</td>
<td>Int. Request Low Time</td>
<td>3.0V</td>
<td>100</td>
<td></td>
<td>100</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td></td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>TwIH</td>
<td>Int. Request Input High Time</td>
<td>3.0V</td>
<td>3TpC</td>
<td></td>
<td>3TpC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>3TpC</td>
<td></td>
<td>3TpC</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Twsm</td>
<td>Stop-Mode Recovery Width Spec</td>
<td>3.0V</td>
<td>12</td>
<td></td>
<td>12</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>12</td>
<td></td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Tosi</td>
<td>Oscillator Startup Time</td>
<td>3.0V</td>
<td>5TpC</td>
<td></td>
<td>5TpC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td></td>
<td>5TpC</td>
<td></td>
</tr>
</tbody>
</table>

Notes:

1. Timing Reference uses 0.7 $V_{cc}$ for a logic 1 and 0.2 $V_{cc}$ for a logic 0.
2. Interrupt request via Port 3 (P33-P31).
3. SMR-D5 = 0.
4. SMR-D5 = 1, POR STOP mode delay is on.
5. Reg. WDTMR.
6. $V_{cc} = 3.0V$ to 5.5V.
7. SMR D1 = 1.
8. Maximum frequency for internal system clock is 4 MHz when using XTL divide-by-one mode.
9. For RC and LC oscillator, and for oscillator driven by clock driver.
EXPANDED REGISTER FILE CONTROL REGISTERS

SMR (F) 0B

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- SCLK/TCLK Divide by 16 (E06)†
  0 OFF**
  1 ON
- External Clock Divide Mode
  0 SCLK/TCLK = External/2*
  1 SCLK/TCLK = External/1
- Stop Mode Recovery Source
  000 POR Only*
  001 POR Only††
  010 P31††
  011 P32††
  100 P33††
  101 P27
  110 P2 NOR 0:3
  111 P2 NOR 0:7
- Stop Delay
  0 OFF
  1 ON
- Stop Recovery Level
  0 Low Level *
  1 High Level
- Stop Flag
  0 POR *
  1 Stop Recovery

* Default setting after RESET.
** Default setting after RESET and STOP-Mode Recovery.
†† E03 reserved; must be 0.
††† E06 only

Figure 30. STOP-Mode Recovery Register (Write Only except bit D7, which is Read Only)

WDTMR (F) 0F

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- WDT TAck†
  00 5 512 TspC
  01 15 * 1024 TspC
  10 25 2048 TspC
  11 100 8192 TspC
- WDT During HALT
  0 OFF
  1 ON *
- WDT During STOP
  0 OFF
  1 ON *
- XTAL1/INT RC Select for WDT
  0 On-Board RC *
  1 XTAL
- Reserved (Must be 0)

* Default setting after RESET.
† For E06; E03 must be D0 = 1, D1 = 0.

Figure 31. Watch-Dog Timer Mode Register (Write Only)

PCON (F) 00

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Comparator
  Output PORT 3
  0 P34, P35 Standard Output*
  1 P34, P35 Comparator Output
- Reserved (Must Be 1)
- Low EMI PORT 2
  0 Low EMI Noise
  1 Standard *
- Low EMI PORT 3
  0 Low EMI Noise
  1 Standard *
- Low EMI Oscillator
  0 Low EMI Noise
  1 Standard *

* Default Setting After Reset.

Figure 32. PORT Control Register (Write Only)

SCON (C) 02

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- SPI Enable
  0 Disable*
  1 Enable
- RxCharOverrun (S)
  0 Reset
  1 Overrun
- CLK Divide (M)
  00 TCLK/2
  01 TCLK/4
  10 TCLK/8
  11 TCLK/16
- DO SPI Port Enable (S)
  0 SPI DO Port Enabled
  1 DO Port to I/O
- Compare Enable
  0 Enable
  1 Disable *
- RxCharAvail
  0 Reset
  1 Char. Avail.
- Clock Phase
  0 Trans/Fail
  1 Trans/Rise
- CLK Source
  0 TCLK
  1 Timer 0 Output
- Master Slave
  0 Slave
  1 Master

(S) Used with Bit D7 equal to 0
(M) Used with Bit D7 equal to 1

* Default Setting After Reset.

Figure 33. SPI Control Register (Z86E06 Only)
EXPANDED REGISTER FILE CONTROL REGISTERS (Continued)

Figure 34. SPI Compare Register (Z86E06 Only)

Figure 35. SPI Receive Buffer (Z86E06 Only)

Z8 CONTROL REGISTER DIAGRAMS

Figure 36. Reserved

Figure 37. Timer Mode Register (F1H: Read/Write)

Figure 38. Counter Timer 1 Register (F2H: Read/Write)

Figure 39. Prescaler 1 Register (F3H: Write Only)

Figure 40. Counter/Timer 0 Register (F4H: Read/Write; Z86E06 Only)

Figure 41. Prescaler 0 Register (F5H: Write Only; Z86E06 Only)
**ADVANCE INFORMATION**

**Z86E031E06 CMOS 8-Bit OTP**

**CONSUMER CONTROLLER PROCESSOR**

---

**Figure 42. Port 2 Mode Register**

(F6H: Write Only)

- **Port 2 I/O Definition**
  - 0: Defines Bit as OUTPUT
  - 1: Defines Bit as INPUT

---

**Figure 43. Port 3 Mode Register**

(F7H: Write Only)

- **Port 3 Inputs**
  - 0: Digital Mode
  - 1: Analog Mode

---

**Figure 44. Port 0 and 1 Mode Register**

(F8H: Write Only)

- **Reserved (Must be 0)**
- **Reserved (Must be 0)**
- **Reserved (Must be 1)**

---

**Figure 45. Interrupt Priority Register**

(F9H: Write Only)

- **Interrupt Group Priority**
  - 000: Reserved
  - 001: C > A > B
  - 010: A > B > C
  - 011: A > C > B
  - 100: B > C > A
  - 101: C > B > A
  - 110: B > A > C
  - 111: Reserved

- **IRQx Priority**
  - 0: IRQx > IRQy
  - 1: IRQy > IRQx

---

**Figure 46. Interrupt Request Register**

(FAH: Read/Write)

- **IRQ0 – IRQ5**
  - 00: P31 ↓ P32 ↓
  - 01: P31 ↓ P32 ↑
  - 10: P31 ↑ P32 ↓
  - 11: P31 ↑ P32 ↑
Z8 CONTROL REGISTER DIAGRAMS (Continued)

**Figure 47. Interrupt Mask Register**  
(FB<sub>H</sub>; Read/Write)

**Figure 48. Flag Register**  
(FC<sub>H</sub>; Read/Write)

**Figure 49. Register Pointer**  
(FD<sub>H</sub>; Read/Write)

**Figure 50. General Purpose Register**  
(FE<sub>H</sub>; Read/Write)

**Figure 51. Stack Pointer**  
(FF<sub>H</sub>; Read/Write)
INSTRUCTION SET NOTATION

Addressing Modes. The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRR</td>
<td>Indirect register pair or indirect working-register pair address</td>
</tr>
<tr>
<td>Irr</td>
<td>Indirect working-register pair only</td>
</tr>
<tr>
<td>X</td>
<td>Indexed address</td>
</tr>
<tr>
<td>DA</td>
<td>Direct address</td>
</tr>
<tr>
<td>RA</td>
<td>Relative address</td>
</tr>
<tr>
<td>IM</td>
<td>Immediate</td>
</tr>
<tr>
<td>R</td>
<td>Register or working-register address</td>
</tr>
<tr>
<td>r</td>
<td>Working-register address only</td>
</tr>
<tr>
<td>IR</td>
<td>Indirect-register or indirect working-register address</td>
</tr>
<tr>
<td>Ir</td>
<td>Indirect working-register address only</td>
</tr>
<tr>
<td>RR</td>
<td>Register pair or working register pair address</td>
</tr>
</tbody>
</table>

Symbols. The following symbols are used in describing the instruction set.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst</td>
<td>Destination location or contents</td>
</tr>
<tr>
<td>src</td>
<td>Source location or contents</td>
</tr>
<tr>
<td>cc</td>
<td>Condition code</td>
</tr>
<tr>
<td>@</td>
<td>Indirect address prefix</td>
</tr>
<tr>
<td>SP</td>
<td>Stack Pointer</td>
</tr>
<tr>
<td>PC</td>
<td>Program Counter</td>
</tr>
<tr>
<td>FLAGS</td>
<td>Flag register (Control Register 252)</td>
</tr>
<tr>
<td>RP</td>
<td>Register Pointer (R253)</td>
</tr>
<tr>
<td>IMR</td>
<td>Interrupt mask register (R251)</td>
</tr>
</tbody>
</table>

Flags. Control register (R252) contains the following six flags:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Carry flag</td>
</tr>
<tr>
<td>Z</td>
<td>Zero flag</td>
</tr>
<tr>
<td>S</td>
<td>Sign flag</td>
</tr>
<tr>
<td>V</td>
<td>Overflow flag</td>
</tr>
<tr>
<td>D</td>
<td>Decimal-adjust flag</td>
</tr>
<tr>
<td>H</td>
<td>Half-carry flag</td>
</tr>
</tbody>
</table>

Affected flags are indicated by:

- 0 Clear to zero
- 1 Set to one
* Set to clear according to operation
- Unaffected
x Undefined
### CONDITION CODES

<table>
<thead>
<tr>
<th>Value</th>
<th>Mnemonic</th>
<th>Meaning</th>
<th>Flags Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>C</td>
<td>Always True</td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>NC</td>
<td>Carry</td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td>Z</td>
<td>No Carry</td>
<td></td>
</tr>
<tr>
<td>0110</td>
<td>NZ</td>
<td>Zero</td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td></td>
<td>Not Zero</td>
<td></td>
</tr>
</tbody>
</table>

| 1101   | PL       | Plus                             |                      |
| 0101   | MI       | Minus                            |                      |
| 0100   | OV       | Overflow                         |                      |
| 1100   | NOV      | No Overflow                      |                      |
| 0110   | EQ       | Equal                            |                      |

| 1110   | NE       | Not Equal                        |                      |
| 1001   | GE       | Greater Than or Equal            | (S XOR V) = 0        |
| 0001   | LT       | Less than                        | (S XOR V) = 1        |
| 1010   | GT       | Greater Than                     | [Z OR (S XOR V)] = 0 |
| 0010   | LE       | Less Than or Equal               | [Z OR (S XOR V)] = 1 |

| 1111   | UGE      | Unsigned Greater Than or Equal   | C = 0                |
| 0111   | ULT      | Unsigned Less Than              | C = 1                |
| 1011   | UGT      | Unsigned Greater Than           | (C = 0 AND Z = 0) = 1|
| 0011   | ULE      | Unsigned Less Than or Equal     | (C OR Z) = 1         |
| 0000   | F        | Never True (Always False)        |                      |
INSTRUCTION FORMATS

### One-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>src</th>
<th>dest</th>
<th>src</th>
<th>dst</th>
<th>src</th>
<th>dest</th>
<th>src</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>VALUE</td>
<td>OPC</td>
<td>VALUE</td>
<td>OPC</td>
<td>VALUE</td>
<td>OPC</td>
<td>VALUE</td>
<td>OPC</td>
<td>VALUE</td>
</tr>
</tbody>
</table>

#### INSTRUCTION SUMMARY

**Note:** Assignment of a value is indicated by the symbol 

\[ \leftarrow \] . For example:

\[ \text{dst} \leftarrow \text{dst} + \text{src} \]

indicates that the source data is added to the destination data and the result is stored in the destination location. The notation "addr (n)" is used to refer to bit (n) of a given operand location. For example:

\[ \text{dst} (7) \]

refers to bit 7 of the destination operand.
### INSTRUCTION SUMMARY (Continued)

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Address Opcode (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ADC</strong> dst, src</td>
<td>°</td>
<td>1[ ]</td>
<td>** * * * 0 **</td>
</tr>
<tr>
<td>dst ← dst + src +C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ADD</strong> dst, src</td>
<td>°</td>
<td>0[ ]</td>
<td>** * * * 0 **</td>
</tr>
<tr>
<td>dst ← dst + src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AND</strong> dst, src</td>
<td>°</td>
<td>5[ ]</td>
<td>- ** 0 - -</td>
</tr>
<tr>
<td>dst ← dst AND src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CALL</strong> dst</td>
<td>DA</td>
<td>D6</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>SP ← SP - 2</td>
<td>IRR</td>
<td>D4</td>
<td></td>
</tr>
<tr>
<td>@SP ← PC,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC ← dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CCF</strong></td>
<td>EF</td>
<td>* - - - - -</td>
<td></td>
</tr>
<tr>
<td>C ← NOT C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CLR</strong> dst</td>
<td>R</td>
<td>80</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>dst ← 0</td>
<td>IR</td>
<td>B1</td>
<td></td>
</tr>
<tr>
<td><strong>COM</strong> dst</td>
<td>R</td>
<td>60</td>
<td>- * 0 - -</td>
</tr>
<tr>
<td>dst ← NOT dst</td>
<td>IR</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td><strong>CP</strong> dst, src</td>
<td>°</td>
<td>A[ ]</td>
<td>** * * * - -</td>
</tr>
<tr>
<td>dst ← src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DA</strong> dst</td>
<td>R</td>
<td>40</td>
<td>** * * X - -</td>
</tr>
<tr>
<td>dst ← DA dst</td>
<td>IR</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td><strong>DEC</strong> dst</td>
<td>R</td>
<td>00</td>
<td>- * * - -</td>
</tr>
<tr>
<td>dst ← dst - 1</td>
<td>IR</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td><strong>DECW</strong> dst</td>
<td>RR</td>
<td>80</td>
<td>- * * - -</td>
</tr>
<tr>
<td>dst ← dst - 1</td>
<td>IR</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td><strong>DI</strong></td>
<td>8F</td>
<td>- - - - - -</td>
<td></td>
</tr>
<tr>
<td>IMR(7) ← 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DJNZ</strong> dst, r</td>
<td>RA</td>
<td>rA</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>r ← r - 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>if r ≠ 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC ← PC + dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range: +127, -128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>EI</strong></td>
<td>9F</td>
<td>- - - - - -</td>
<td></td>
</tr>
<tr>
<td>IMR(7) ← 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HALT</strong></td>
<td>7F</td>
<td>- - - - - -</td>
<td></td>
</tr>
<tr>
<td><strong>INC</strong> dst</td>
<td>r</td>
<td>rE</td>
<td>- * * - -</td>
</tr>
<tr>
<td>dst ← dst + 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>INCW</strong> dst</td>
<td>RR</td>
<td>A0</td>
<td>- * * - -</td>
</tr>
<tr>
<td>dst ← dst + 1</td>
<td>IR</td>
<td>A1</td>
<td></td>
</tr>
<tr>
<td><strong>IRET</strong></td>
<td>BF</td>
<td>** * * * **</td>
<td></td>
</tr>
<tr>
<td>FLAGS ← @SP;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP ← SP + 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC ← @SP;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP ← SP + 2;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMR(7) ← 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>JP</strong> cc, dst</td>
<td>DA</td>
<td>cD</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>if cc is true,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC ← dst</td>
<td>IRR</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td><strong>JR</strong> cc, dst</td>
<td>RA</td>
<td>cB</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>if cc is true,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC ← PC + dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range: +127, -128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LD</strong> dst, src</td>
<td>r</td>
<td>Im</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>dst ← src</td>
<td>R</td>
<td>r8</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>r</td>
<td>r9</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>r ← r - 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC ← PC + dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range: +127, -128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LDC</strong> dst, src</td>
<td>r</td>
<td>Irr</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>dst ← src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LDCI</strong> dst, src</td>
<td>lr</td>
<td>lr</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>dst ← src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r ← r + 1; fr ← fr + 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NOP</strong></td>
<td>FF</td>
<td>- - - - - -</td>
<td></td>
</tr>
</tbody>
</table>
### Instruction and Operation

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR dst, src</td>
<td>t</td>
<td>4[ ]</td>
<td>* * 0</td>
</tr>
<tr>
<td>POP dst</td>
<td>R</td>
<td>50</td>
<td>-</td>
</tr>
<tr>
<td>PUSH src</td>
<td>R</td>
<td>70</td>
<td>-</td>
</tr>
<tr>
<td>RCL dst</td>
<td>R</td>
<td>10</td>
<td>* * * *</td>
</tr>
<tr>
<td>RR dst</td>
<td>R</td>
<td>E0</td>
<td>* * * *</td>
</tr>
<tr>
<td>RRC dst</td>
<td>R</td>
<td>C0</td>
<td>* * * *</td>
</tr>
<tr>
<td>SBC dst, src</td>
<td>t</td>
<td>3[ ]</td>
<td>* * * 1 *</td>
</tr>
<tr>
<td>SCF</td>
<td>D1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>SRA dst</td>
<td>R</td>
<td>D0</td>
<td>* * 0</td>
</tr>
<tr>
<td>SRP dst</td>
<td>im</td>
<td>31</td>
<td>-</td>
</tr>
<tr>
<td>STOP</td>
<td>6F</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>

### Instruction and Operation

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUB dst, src</td>
<td>t</td>
<td>2[ ]</td>
<td>* * * 1 *</td>
</tr>
<tr>
<td>SWAP dst</td>
<td>R</td>
<td>F0</td>
<td>X * X</td>
</tr>
<tr>
<td>TM dst, src</td>
<td>t</td>
<td>7[ ]</td>
<td>* * 0</td>
</tr>
<tr>
<td>WDT</td>
<td>5F</td>
<td>-</td>
<td>X X</td>
</tr>
<tr>
<td>XOR dst, src</td>
<td>t</td>
<td>8[ ]</td>
<td>* * 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address Mode</th>
<th>Opcode Nibble</th>
</tr>
</thead>
<tbody>
<tr>
<td>r r</td>
<td>[2]</td>
</tr>
<tr>
<td>r ir</td>
<td>[3]</td>
</tr>
<tr>
<td>R R</td>
<td>[4]</td>
</tr>
<tr>
<td>R IR</td>
<td>[5]</td>
</tr>
<tr>
<td>R IM</td>
<td>[6]</td>
</tr>
<tr>
<td>IR IM</td>
<td>[7]</td>
</tr>
</tbody>
</table>
## Opcode Map

<table>
<thead>
<tr>
<th>Lower Nibble (Hex)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.65</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>DEC R1</td>
<td>DEC R1</td>
<td>ADD R1, R2</td>
<td>ADD R2, R1</td>
<td>ADD R2, R1</td>
<td>ADD R2, R1</td>
<td>ADD R2, R1</td>
<td>ADD R2, R1</td>
<td>LD R1, R2</td>
<td>LD R1, R2</td>
<td>DJNZ R1, R2</td>
<td>JR cc, R2</td>
<td>LD R1, IM</td>
<td>LD R1, IM</td>
<td>INC R1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>8.0</td>
<td>6.1</td>
<td>6.1</td>
<td>6.1</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8.5</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10/12.1</td>
<td>12/14.1</td>
<td>6.5</td>
<td>6.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>8.5</td>
<td>8.5</td>
<td>8.5</td>
<td>8.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>8.5</td>
<td>8.5</td>
<td>8.5</td>
<td>8.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>10.5</td>
<td>6.5</td>
<td>6.5</td>
<td>12/10.5</td>
<td>12/10.0</td>
<td>6.5</td>
<td>12.10.0</td>
<td>6.5</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- **R** = 8-bit address
- **r** = 4-bit address
- **R1** or **r1** = Dst address
- **R2** or **r2** = Src address

**Sequence:**
Opcode, First Operand, Second Operand

**Note:** The blank areas are reserved.

* 2-byte instruction appears as a 3-byte instruction
18-Pin DIP Package Diagram

18-Pin SOIC Package Diagram
ORDERING INFORMATION

Z86E03 (8 MHz)

<table>
<thead>
<tr>
<th>Standard Temperature</th>
<th>Extended Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>18-Pin DIP</td>
<td>18-Pin SOIC</td>
</tr>
<tr>
<td>Z86E0308PSC</td>
<td>Z86E0308SSC</td>
</tr>
<tr>
<td>18-Pin DIP</td>
<td>18-Pin SOIC</td>
</tr>
<tr>
<td>Z86E0308PEC</td>
<td>Z86E0308SEC</td>
</tr>
</tbody>
</table>

Z86E06 (12 MHz)

<table>
<thead>
<tr>
<th>Standard Temperature</th>
<th>Extended Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>18-Pin DIP</td>
<td>18-Pin SOIC</td>
</tr>
<tr>
<td>Z86E0612PSC</td>
<td>Z86E0612SSC</td>
</tr>
<tr>
<td>18-Pin DIP</td>
<td>18-Pin SOIC</td>
</tr>
<tr>
<td>Z86E0612PEC</td>
<td>Z86E0612SEC</td>
</tr>
</tbody>
</table>

For fast results, contact your local Zilog sales office for assistance in ordering the part(s) desired.

CODES

Preferred Package
P = Plastic DIP

Longer Lead Time
S = Plastic SOIC

Preferred Temperature
S = 0°C to +70°C

Longer Lead Time
E = -40°C to +105°C

Speeds
08 = 8 MHz
12 = 12 MHz

Environmental
C = Plastic Standard

Example:
Z 86E03 08 P S C

is a Z86E03, 8 MHz, DIP, 0°C to +70°C, Plastic Standard Flow
Z86C03/C06 CMOS Z8® 8-Bit CCP™ Consumer Controller Processors

Z86E03/E06 CMOS Z8® 8-Bit OTP CCP™ Consumer Controller Processors

Z86C04/C08 CMOS Z8® Low Cost 1K /2K ROM Microcontrollers

Z86E04/E08 CMOS Z8® 8-Bit OTP Microcontrollers

Z86C07 CMOS Z8® 8-Bit Microcontroller

Z86E07 CMOS Z8® 8-Bit OTP Microcontroller

Z86C30/C31 CMOS Z8® 8-Bit CCP™ Consumer Controller Processors
PRODUCT SPECIFICATION

Z86C04/Z86C08 CMOS Z8® 8-Bit Low-Cost 1K/2K-ROM Microcontrollers

FEATURES

- The Z86C04/C08 Devices Have the Following General Characteristics:

<table>
<thead>
<tr>
<th>Part</th>
<th>ROM</th>
<th>RAM</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z86C04</td>
<td>1 Kbytes</td>
<td>124</td>
<td>8 MHz</td>
</tr>
<tr>
<td>Z86C08</td>
<td>2 Kbytes</td>
<td>124</td>
<td>12 MHz</td>
</tr>
</tbody>
</table>

- Three Digital Inputs at CMOS Levels
- Eleven Digital Inputs at CMOS Levels; Schmitt-Triggered
- Two Programmable 8-Bit Counter/Timers, each with a 6-Bit Programmable Prescaler
- Six Vectored, Priority Interrupts from Six Different Sources
- Software-Enabled Watch-Dog Timer
- Power-On Reset Timer
- Two On-Board Comparators
- On-Chip Oscillator that Accepts a Crystal, Ceramic Resonator, LC, or External Clock Drive
- Programmable Interrupt Polarity
- Auto Latches
- ROM Protect, Low EMI Options

GENERAL DESCRIPTION

The Z86C04/C08 Microcontroller Units (MCUs) are members of the Z8® single-chip microcontroller family with 124 bytes of general-purpose RAM and 1/2 Kbytes of ROM, respectively. The devices are offered in 18-pin DIP and SOIC style packages and are manufactured in CMOS technology. Zilog’s low-cost, low-power consumption Z86C04/C08 offers all the outstanding features of the Z8 family architecture, plus easy software/hardware system expansion.

For applications demanding powerful I/O capabilities, the Z86C04/C08 provides 14 pins dedicated to input and output. These lines are grouped into three ports, and are configurable under software control to provide I/O, timing, and status signals. There are two basic address spaces available to support this configuration: Program Memory, and 124 bytes of general-purpose RAM.

To unburden the system from coping with real-time tasks, such as counting/timing and I/O data communications, the Z86C04/C08 offers two on-chip counter/timers with a large number of user-selectable modes. Additionally, two on-board comparators process analog signals with a common reference voltage (Figure 1).
GENERAL DESCRIPTION (Continued)

Characterized by a flexible I/O scheme, an efficient register and address space structure, and a number of ancillary features, the Z86C04/C08 is well-suited for a variety of consumer, industrial and commercial applications.

Notes:
All Signals with a preceding front slash, "/", are active Low, e.g.: B/W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

<table>
<thead>
<tr>
<th>Connection</th>
<th>Circuit</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>Vcc</td>
<td>VDD</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
<td>VSS</td>
</tr>
</tbody>
</table>

Figure 1. Z86C04/C08 Functional Block Diagram
**PIN DESCRIPTIONS**

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>P24-P27</td>
<td>Port 2, Pins 4, 5, 6, 7</td>
<td>In/Output</td>
</tr>
<tr>
<td>5</td>
<td>V_{cc}</td>
<td>Power Supply</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>XTAL2</td>
<td>Crystal Oscillator Clock</td>
<td>Output</td>
</tr>
<tr>
<td>7</td>
<td>XTAL1</td>
<td>Crystal Oscillator Clock</td>
<td>Input</td>
</tr>
<tr>
<td>8</td>
<td>P31</td>
<td>Port 3, Pin 1, AN1</td>
<td>Input</td>
</tr>
<tr>
<td>9</td>
<td>P32</td>
<td>Port 3, Pin 2, AN2</td>
<td>Input</td>
</tr>
<tr>
<td>10</td>
<td>P33</td>
<td>Port 3, Pin 3, REF</td>
<td>Input</td>
</tr>
<tr>
<td>11-13</td>
<td>P00-P02</td>
<td>Port 0, Pins 0, 1, 2</td>
<td>In/Output</td>
</tr>
<tr>
<td>14</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>15-18</td>
<td>P20-P23</td>
<td>Port 2, Pins 0, 1, 2, 3</td>
<td>In/Output</td>
</tr>
</tbody>
</table>

Figure 2. 18-Pin DIP Configuration

Figure 3. 18-Pin SOIC Pin Configuration
PIN DESCRIPTION (Continued)

**XTAL1, XTAL2** Crystal In, Crystal Out (time-based input and output, respectively). These pins connect a parallel-resonant crystal, LC, or an external single-phase clock to the on-chip clock oscillator and buffer.

**Auto Latch.** The Auto Latch puts valid CMOS levels on all CMOS inputs (except P33, P32, P31) that are not externally driven. Whether this level is 0 or 1 cannot be determined. A valid CMOS level, rather than a floating node, reduces excessive supply current flow in the input buffer.

**Port 0** (P02-P00). Port 0 is a 3-bit I/O, bi-directional, Schmitt-triggered CMOS compatible I/O port. These three I/O lines can be configured under software control to be all inputs or all outputs (Figure 4).

![Figure 4. Port 0 Configuration](image-url)
Port 2 (P27-P20). Port 2 is an 8-bit I/O, bit programmable, bi-directional, Schmitt-triggered CMOS compatible I/O port. These eight I/O lines can be configured under software control to be an input or output, independently. Bits programmed as outputs may be globally programmed as either push-pull or open-drain (Figure 5).

Figure 5. Port 2 Configuration
PIN DESCRIPTION (Continued)

Port 3 (P33-P31). Port 3 is a 3-bit, CMOS compatible port with three fixed input (P33-P31) lines. These three input lines can be configured under software control as digital inputs or analog inputs. These three input lines can also be used as the interrupt sources IRQ0-IRQ3 and as the timer input signal ($T_{IN}$) (Figure 6).

![Port 3 Configuration Diagram](image)

**Figure 6. Port 3 Configuration**

**Comparator Inputs.** Two analog comparators are added to Port 3 inputs for interface flexibility. Typical applications for these on-board comparators are: Zero crossing detection, A/D conversion, voltage scaling, and threshold detection.

The dual comparator (common inverting terminal) features a single power supply which discontinues power in STOP mode. The common voltage range is 0-4V when the $V_{CC}$ is 5.0V.

Interrupts are generated on either edge of Comparator 2’s output, or on the falling edge of Comparator 1’s output. The comparator output may be used for interrupt generation, Port 3 data inputs, or $T_{IN}$ through P31. Alternately, the comparators may be disabled, freeing the reference input (P33) for use as IRQ1 and/or P33 input.
FUNCTIONAL DESCRIPTION

RESET. Upon power-up the Power-On Reset circuit waits for $T_{POR}$ ms, plus 18 clock cycles, and then starts program execution at address %000C (Hex) (Figure 7). The Z86C04/C08 control registers’ reset value is shown in Table 2.

![Figure 7. Internal Reset Configuration](image)

Table 2. Z86C04/C08 Control Registers

<table>
<thead>
<tr>
<th>Addr. Reg.</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1 TMR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>F2 T1</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>F3 PRE1</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>F4 TO</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>F5 PRE0</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>F6 P2M</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Inputs after reset</td>
</tr>
<tr>
<td>F7 P3M</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>F8 P01M</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>F9 IPR</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>FA IRQ</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>IRQ3 is used for positive edge detection</td>
</tr>
<tr>
<td>FB IMR</td>
<td>0</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>FC FLAGS</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>FD RP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>FF SPL</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>*00 Port 0</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>*02 Port 2</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>*03 Port 3</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
</tbody>
</table>

Note:
* Registers are not reset after a STOP-Mode Recovery using P27 pin. A subsequent reset will cause these control registers to be reconfigured as shown in Table 2 and the user must avoid bus contention on the port pins or it may affect device reliability.

Program Memory. The Z86C04/C08 can address up to 1K/2K bytes of internal program memory (Figure 8). The first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts. Bytes 0-1023/2047 are on-chip mask-programmed ROM.

![Figure 8. Program Memory Map](image)
FUNCTIONAL DESCRIPTION (Continued)

Register File. The Register File consists of three I/O port registers, 124 general-purpose registers, and 15 control and status registers (R0, R2-R3, R4-R127, and R241-R255, respectively; see Figure 9). Note that R254 is available for general purpose use. The Z86C04/C08 instructions can access registers directly or indirectly through an 8-bit address field. This allows short 4-bit register addressing using the Register Pointer. In the 4-bit mode, the register file is divided into eight working register groups, each occupying 16 continuous locations. The Register Pointer (Figure 10) addresses the starting location of the active working-register group. Upon power-up, the general purpose registers are undefined.

<table>
<thead>
<tr>
<th>Location</th>
<th>Stack Pointer (Bits 7-0)</th>
<th>SPL</th>
<th>r7 r6 r5 r4</th>
<th>r3 r2 r1 r0</th>
</tr>
</thead>
<tbody>
<tr>
<td>255</td>
<td>General Purpose Register</td>
<td>RP</td>
<td>SF</td>
<td>SF</td>
</tr>
<tr>
<td>254</td>
<td>Register Pointer</td>
<td>Flags</td>
<td>7F</td>
<td>7F</td>
</tr>
<tr>
<td>253</td>
<td>Program Control Flags</td>
<td>IMR</td>
<td>6F</td>
<td>6F</td>
</tr>
<tr>
<td>252</td>
<td>Interrupt Mask Register</td>
<td>IRQ</td>
<td>5F</td>
<td>5F</td>
</tr>
<tr>
<td>251</td>
<td>Interrupt Request Register</td>
<td>IPR</td>
<td>4F</td>
<td>4F</td>
</tr>
<tr>
<td>250</td>
<td>Interrupt Priority Register</td>
<td>P01M</td>
<td>3F</td>
<td>3F</td>
</tr>
<tr>
<td>249</td>
<td>Ports 0-1 Mode</td>
<td>P3M</td>
<td>2F</td>
<td>2F</td>
</tr>
<tr>
<td>248</td>
<td>Port 3 Mode</td>
<td>P2M</td>
<td>1F</td>
<td>1F</td>
</tr>
<tr>
<td>247</td>
<td>Port 2 Mode</td>
<td>PRE0</td>
<td>0F</td>
<td>0F</td>
</tr>
<tr>
<td>246</td>
<td>Not Implemented</td>
<td>T0</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>245</td>
<td>T0 Prescaler</td>
<td>PRE1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>244</td>
<td>Timer/Counter 0</td>
<td>T1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>243</td>
<td>T1 Prescaler</td>
<td>TMR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>242</td>
<td>Timer/Counter 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>241</td>
<td>Timer Mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>240</td>
<td>Not Implemented</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>General Purpose Registers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>Port 3</td>
<td>P3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>126</td>
<td>Port 2</td>
<td>P2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>Reserved</td>
<td>P1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>124</td>
<td>Port 0</td>
<td>P0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 9. Register File

Figure 10. Register Pointer

*Expanded Register Group (0) is selected in this figure by handling bits D3 to D0 as "0" in Register R253(RP).
Stack Pointer. The Z86C04/C08 has an 8-bit Stack Pointer (R255) used for the internal stack that resides within the 124 general-purpose registers.

General-Purpose Register (GPR). The general-purpose register upon device power-up is undefined. The general-purpose register upon a STOP-Mode Recovery and reset stays in its last state. It will not keep its last state from a V_{CC} droop below 1.8V. Note: Register R254 has been designated as a general-purpose register.

Counter/Timer. There are two 8-bit programmable counter/timers (T0 and T1), each driven by its own 6-bit programmable prescaler. The T1 prescaler can be driven by internal or external clock sources, however the T0 can be driven by the internal clock source only (Figure 11).

The 6-bit prescalers can divide the input frequency of the clock source by any integer number from 1 to 64. Each prescaler drives its counter, which decrements the value (1 to 256) that has been loaded into the counter. When both counter and prescaler reach the end of count, a timer interrupt request, IRQ4 (T0) or IRQ5 (T1), is generated.

The counter can be programmed to start, stop, restart to continue, or restart from the initial value. The counters can also be programmed to stop upon reaching zero (single pass mode) or to automatically reload the initial value and continue counting (modulo-n continuous mode).

The counters, but not the prescalers are read at any time without disturbing their value or count mode. The clock source for T1 is user-definable and can be either the internal microprocessor clock divided by four, or an external signal input through Port 3. The Timer Mode register configures the external timer input (P31) as an external clock, a trigger input that is retriggerable or non-retriggerable, or as a gate input for the internal clock.

* Note: Divide-by-two is not used in Low EMI Mode.
FUNCTIONAL DESCRIPTION (Continued)

Interrupts. The Z86C04/C08 has six interrupts from six different sources. These interrupts are maskable and prioritized (Figure 12). The six sources are divided as follows: the falling edge of P31 (AN1), P32 (AN2), P33 (REF), the rising edge of P32 (AN2), and the two counter/timers. The Interrupt Mask Register globally or individually enables or disables the six interrupt requests (Table 3).

When more than one interrupt is pending, priorities are resolved by a programmable priority encoder that is controlled by the Interrupt Priority register. All Z86C04/C08 interrupts are vectored through locations in program memory. When an Interrupt machine cycle is activated, an interrupt request is granted. This disables all subsequent interrupts, saves the Program Counter and Status Flags, and then branches to the program memory vector location reserved for that interrupt. This memory location and the next byte contain the 16-bit starting address of the interrupt service routine for that particular interrupt request.

To accommodate polled interrupt systems, interrupt inputs are masked and the interrupt request register is polled to determine which of the interrupt requests needs service.

**Note:** User must select any Z86C08 mode in Zilog's C12 ICEBOX™ emulator. The rising edge interrupt is not supported on the Z86CCP00ZEM emulator.

**Table 3. Interrupt Types, Sources, and Vectors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Vector Location</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRQ0</td>
<td>AN2(P32)</td>
<td>0,1</td>
<td>External (F)Edge</td>
</tr>
<tr>
<td>IRQ1</td>
<td>REF(P33)</td>
<td>2,3</td>
<td>External (F)Edge</td>
</tr>
<tr>
<td>IRQ2</td>
<td>AN1(P31)</td>
<td>4,5</td>
<td>External (F)Edge</td>
</tr>
<tr>
<td>IRQ3</td>
<td>AN2(P32)</td>
<td>6,7</td>
<td>External (R)Edge</td>
</tr>
<tr>
<td>IRQ4</td>
<td>T0</td>
<td>8,9</td>
<td>Internal</td>
</tr>
<tr>
<td>IRQ5</td>
<td>T1</td>
<td>10,11</td>
<td>Internal</td>
</tr>
</tbody>
</table>

**Notes:**
F = Falling edge triggered
R = Rising edge triggered

**Figure 12. Interrupt Block Diagram**
Clock. The Z86C04/C08 on-chip oscillator has a high-gain, parallel-resonant amplifier for connection to a crystal, ceramic resonator, or any suitable external clock source (XTAL1 = Input, XTAL2 = Output). The crystal should be AT cut, 12 MHz max, with a series resistance (RS) less than or equal to 100 Ohms. Note: C04 is 8 MHz max.

The crystal should be connected across XTAL1 and XTAL2 using the vendor's crystal recommended capacitors (capacitance is between 10 pF to 250 pF, which depends on the crystal manufacturer, ceramic resonator and PCB layout) from each pin directly to device Ground pin 14 (Figure 13).

Note that the crystal capacitor loads should be connected to VSS pin 14 to reduce ground noise injection.

HALT Mode. This instruction turns off the internal CPU clock but not the crystal oscillation. The counter/timers and external interrupts IRQ0, IRQ1, IRQ2, and IRQ3 remain active. The device can be recovered by interrupts, either externally or internally generated. An interrupt request must be executed (enabled) to exit HALT mode. After the interrupt service routine, the program continues from the instruction after the HALT.

STOP Mode. This instruction turns off the internal clock and external crystal oscillation and reduces the standby current. The STOP mode can be released by two methods. The first method is a RESET of the device by removing VCC. The second method is if P27 is configured as an input line when the device executes the STOP instruction. A low input condition on P27 releases the STOP mode.

Program execution under both conditions begins at location 000C (Hex). However, when P27 is used to release the STOP mode, the I/O port mode registers are not reconfigured to their default power-on conditions. This prevents any I/O configured as output when the STOP instruction was executed, from glitching to an unknown state. To use the P27 release approach with STOP mode, use the following instruction:

```
LD P2M, #1XXX XXXXB
NOP
STOP
```

(X = dependent upon user's application.)

In order to enter STOP or HALT mode, it is necessary to first flush the instruction pipeline to avoid suspending execution in mid-instruction. To do this, the user must execute a NOP (opcode = FFH) immediately before the appropriate sleep instruction, that is, as follows:

```
FF NOP ; clear the pipeline
6F STOP ; enter STOP mode
FF NOP ; clear the pipeline
7F HALT ; enter HALT mode
```

Watch-Dog Timer (WDT). The Watch-Dog Timer is enabled by instruction WDT. When the WDT is enabled, it cannot be stopped by the instruction. With the WDT instruction, the WDT should be refreshed once the WDT is enabled within every Twdt period; otherwise, the Z86C04/C08 resets itself. The WDT instruction affects the Flags accordingly: Z = 1, S = 0, V = 0.

```
WDT = 5F (Hex)
```

![Figure 13. Oscillator Configuration](image-url)
FUNCTIONAL DESCRIPTION (Continued)

Opcode WDT (5FH). The first time opcode 5FH is executed, the WDT is enabled, and subsequent execution clears the WDT counter. This has to be done within the maximum $T_{WDT}$ period; otherwise, the WDT times out and generates a Reset. The generated Reset is the same as a Power-On Reset of $T_{POR}$ plus 18 XTAL clock cycles. The WDT does not work in STOP mode. The WDT is disabled during and after a Reset, until the WDT is enabled again.

Opcode WDH (4FH). When this instruction is executed it will enable the WDT during HALT. If not, the WDT will stop when entering HALT. This instruction does not clear the counters, it facilitates running the WDT function during HALT mode. A WDH instruction executed without executing WOT (5FH) has no effect.

Low Voltage Protection ($V_{LV}$). Maximum ($V_{LV}$) Conditions:

Case 1: $T_A = -40\degree\text{C}, +85\degree\text{C}$, Internal Clock Frequency equal or less than 2 MHz

Case 2: $T_A = -40\degree\text{C}, +105\degree\text{C}$, Internal Clock Frequency equal or less than 1 MHz

Note: The internal clock frequency is one-half the external clock frequency in standard mode.

The device will function normally at or above 3.0V under all conditions. Below 3.0V, the device functions normally until the Low Voltage Protection trip point ($V_{LV}$) is reached. The device is guaranteed to function normally at supply voltages above the low voltage trip point for the temperatures and operating frequencies in Cases 1 and 2. The actual low voltage trip point is a function of temperature and process parameters (Figure 14).

### 2 MHz (Typical)

<table>
<thead>
<tr>
<th>Temp</th>
<th>$-40\degree\text{C}$</th>
<th>$0\degree\text{C}$</th>
<th>$+25\degree\text{C}$</th>
<th>$+70\degree\text{C}$</th>
<th>$+105\degree\text{C}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{LV}$</td>
<td>2.55</td>
<td>2.4</td>
<td>2.1</td>
<td>1.7</td>
<td>1.6</td>
</tr>
</tbody>
</table>

ROM Protect. ROM Protect fully protects the Z86C04/C08 ROM code from being read internally. When ROM Protect is selected, the Z86C04/C08 will disable the instructions LDC and LDCI (Z86C04/C08 and Z86E04/E08 do not support the instructions of LDE and LDEI) in all modes. ROM look-up tables cannot be used in this mode.

![Figure 14. Typical Z86C04/C08 $V_{LV}$ vs. Temperature](image)
ABSOLUTE MAXIMUM RATINGS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient Temperature under Bias</td>
<td>-40</td>
<td>+105</td>
<td>C</td>
</tr>
<tr>
<td>Storage Temperature</td>
<td>-65</td>
<td>+150</td>
<td>C</td>
</tr>
<tr>
<td>Voltage on any Pin with Respect to V&lt;sub&gt;ss&lt;/sub&gt; [Note 1]</td>
<td>-0.6</td>
<td>+12</td>
<td>V</td>
</tr>
<tr>
<td>Voltage on V&lt;sub&gt;dd&lt;/sub&gt; Pin with Respect to V&lt;sub&gt;ss&lt;/sub&gt;</td>
<td>-0.3</td>
<td>+7</td>
<td>V</td>
</tr>
<tr>
<td>Voltage on Pin 7 with Respect to V&lt;sub&gt;ss&lt;/sub&gt; [Note 2]</td>
<td>-0.6</td>
<td>V&lt;sub&gt;dd&lt;/sub&gt;+1</td>
<td>V</td>
</tr>
<tr>
<td>Total Power Dissipation</td>
<td>462</td>
<td></td>
<td>mW</td>
</tr>
<tr>
<td>Maximum Current out of V&lt;sub&gt;ss&lt;/sub&gt;</td>
<td>84</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Maximum Current into V&lt;sub&gt;dd&lt;/sub&gt;</td>
<td>84</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Maximum Current into an Input Pin [Note 3]</td>
<td>-600</td>
<td>+600</td>
<td>µA</td>
</tr>
<tr>
<td>Maximum Current into an Open-Drain Pin [Note 4]</td>
<td>-600</td>
<td>+600</td>
<td>µA</td>
</tr>
<tr>
<td>Maximum Output Current Sunked by Any I/O Pin</td>
<td>12</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Maximum Output Current Sourced by Any I/O Pin</td>
<td>12</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Total Maximum Output Current Sunked by Port 2</td>
<td>70</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Total Maximum Output Current Sourced by Port 2</td>
<td>70</td>
<td></td>
<td>mA</td>
</tr>
</tbody>
</table>

Notice:
Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for an extended period may affect device reliability. Total power dissipation should not exceed 462 mW for the package. Power dissipation is calculated as follows:

Total power dissipation = \( V_{dd} \times (I_{dd} - (\text{sum of } I_{ow})) + \text{sum of } [(V_{dd} - V_{ow}) \times I_{ow}] + \text{sum of } (V_{ox} \times I_{ox}) \)

STANDARD TEST CONDITIONS

The characteristics listed below apply for standard test conditions as noted. All voltages are referenced to Ground. Positive current flows into the referenced pin (Figure 15).

CAPACITANCE

\( T_A = 25^\circ C, V_{cc} = \text{GND} = 0V, f = 1.0 \text{ MHz}, \) unmeasured pins returned to GND.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input capacitance</td>
<td>0</td>
<td>10 pF</td>
</tr>
<tr>
<td>Output capacitance</td>
<td>0</td>
<td>20 pF</td>
</tr>
<tr>
<td>I/O capacitance</td>
<td>0</td>
<td>25 pF</td>
</tr>
</tbody>
</table>
## DC ELECTRICAL CHARACTERISTICS

<table>
<thead>
<tr>
<th>Sym</th>
<th>Parameter</th>
<th>( V_{CC} ) [4]</th>
<th>( T_A = 0^\circ C \text{ to } +70^\circ C )</th>
<th>( T_A = -40^\circ C \text{ to } +105^\circ C )</th>
<th>Typical @ 25°C Units</th>
<th>Conditions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
<td>Units</td>
<td></td>
</tr>
<tr>
<td>( V_{CH} )</td>
<td>Clock Input High Voltage</td>
<td>3.0V</td>
<td>0.8 ( V_{CC} )</td>
<td>( V_{CC}+0.3 )</td>
<td>0.8 ( V_{CC} )</td>
<td>( V_{CC}+0.3 )</td>
<td>1.7 V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.8 ( V_{CC} )</td>
<td>( V_{CC}+0.3 )</td>
<td>0.8 ( V_{CC} )</td>
<td>( V_{CC}+0.3 )</td>
<td>2.8 V</td>
</tr>
<tr>
<td>( V_{CL} )</td>
<td>Clock Input Low Voltage</td>
<td>3.0V</td>
<td>( V_{SS}+0.3 )</td>
<td>0.2 ( V_{CC} )</td>
<td>( V_{SS}+0.3 )</td>
<td>0.2 ( V_{CC} )</td>
<td>0.8 V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>( V_{SS}+0.3 )</td>
<td>0.2 ( V_{CC} )</td>
<td>( V_{SS}+0.3 )</td>
<td>0.2 ( V_{CC} )</td>
<td>1.7 V</td>
</tr>
<tr>
<td>( V_{IH} )</td>
<td>Input High Voltage</td>
<td>3.0V</td>
<td>0.7 ( V_{CC} )</td>
<td>( V_{CC}+0.3 )</td>
<td>0.7 ( V_{CC} )</td>
<td>( V_{CC}+0.3 )</td>
<td>1.8 V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.7 ( V_{CC} )</td>
<td>( V_{CC}+0.3 )</td>
<td>0.7 ( V_{CC} )</td>
<td>( V_{CC}+0.3 )</td>
<td>2.8 V</td>
</tr>
<tr>
<td>( V_{IL} )</td>
<td>Input Low Voltage</td>
<td>3.0V</td>
<td>( V_{SS}+0.3 )</td>
<td>0.2 ( V_{CC} )</td>
<td>( V_{SS}+0.3 )</td>
<td>0.2 ( V_{CC} )</td>
<td>0.8 V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>( V_{SS}+0.3 )</td>
<td>0.2 ( V_{CC} )</td>
<td>( V_{SS}+0.3 )</td>
<td>0.2 ( V_{CC} )</td>
<td>1.5 V</td>
</tr>
<tr>
<td>( V_{OH} )</td>
<td>Output High Voltage</td>
<td>3.0V</td>
<td>( V_{CC}+0.4 )</td>
<td>( V_{CC}+0.4 )</td>
<td>( V_{CC}+0.4 )</td>
<td>( V_{CC}+0.4 )</td>
<td>3.0 V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>( V_{CC}+0.4 )</td>
<td>( V_{CC}+0.4 )</td>
<td>( V_{CC}+0.4 )</td>
<td>( V_{CC}+0.4 )</td>
<td>4.8 V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>( V_{CC}+0.4 )</td>
<td>( V_{CC}+0.4 )</td>
<td>( V_{CC}+0.4 )</td>
<td>( V_{CC}+0.4 )</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>( V_{CC}+0.4 )</td>
<td>( V_{CC}+0.4 )</td>
<td>( V_{CC}+0.4 )</td>
<td>( V_{CC}+0.4 )</td>
<td>V</td>
</tr>
<tr>
<td>( V_{OL1} )</td>
<td>Output Low Voltage</td>
<td>3.0V</td>
<td>0.8</td>
<td>0.8</td>
<td>0.2 V</td>
<td>( I_{OL} = +4.0 \text{ mA} ) [5]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.4</td>
<td>0.4</td>
<td>0.1 V</td>
<td>( I_{OL} = +4.0 \text{ mA} ) [5]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>0.4</td>
<td>0.4</td>
<td>V</td>
<td>Low Noise @ ( I_{OL} = 1.0 \text{ mA} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.4</td>
<td>0.4</td>
<td>V</td>
<td>Low Noise @ ( I_{OL} = 1.0 \text{ mA} )</td>
<td></td>
</tr>
<tr>
<td>( V_{OL2} )</td>
<td>Output Low Voltage</td>
<td>3.0V</td>
<td>1.0</td>
<td>1.0</td>
<td>0.8 V</td>
<td>( I_{OL} = +12 \text{ mA,} ) 3 Pin Max [5]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.8</td>
<td>0.8</td>
<td>0.3 V</td>
<td>( I_{OL} = +12 \text{ mA,} ) 3 Pin Max [5]</td>
<td></td>
</tr>
<tr>
<td>( V_{OFFSET} )</td>
<td>Comparator Input Offset Voltage</td>
<td>3.0V</td>
<td>25</td>
<td>25</td>
<td>10 mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
<td>10 mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( V_{LV} )</td>
<td>( V_{CC} ) Low Voltage</td>
<td>2.7</td>
<td>2.95</td>
<td>2.1 V</td>
<td>( \oplus ) 1 MHz Max, Int. CLK Freq</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( I_{IL} )</td>
<td>Input Leakage (Input Bias Current of Comparator)</td>
<td>3.0V</td>
<td>-1.0</td>
<td>1.0</td>
<td>-1.0</td>
<td>1.0</td>
<td>( \mu A )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>-1.0</td>
<td>1.0</td>
<td>-1.0</td>
<td>1.0</td>
<td>( \mu A )</td>
</tr>
<tr>
<td>( I_{OL} )</td>
<td>Output Leakage</td>
<td>3.0V</td>
<td>-1.0</td>
<td>1.0</td>
<td>-1.0</td>
<td>1.0</td>
<td>( \mu A )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>-1.0</td>
<td>1.0</td>
<td>-1.0</td>
<td>1.0</td>
<td>( \mu A )</td>
</tr>
<tr>
<td>( V_{ICR} )</td>
<td>Comparator Input Common Mode Voltage Range</td>
<td>0</td>
<td>( V_{CC} -1.0 )</td>
<td>0</td>
<td>( V_{CC} -1.5 )</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Sym</td>
<td>Parameter</td>
<td>$V_{CC}$ [V]</td>
<td>$T_{A} = 0^\circ C$ to $+70^\circ C$</td>
<td>$T_{A} = -40^\circ C$ to $+105^\circ C$</td>
<td>Typical @ $25^\circ C$</td>
<td>Units</td>
<td>Conditions</td>
</tr>
<tr>
<td>------</td>
<td>----------------------------</td>
<td>-------------</td>
<td>----------------------------------------</td>
<td>------------------------------------------</td>
<td>-------------------------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>I_{CC}</td>
<td>Supply Current</td>
<td>3.2V</td>
<td>-</td>
<td>-</td>
<td>80 µA</td>
<td>mA</td>
<td>All output and I/O Pins Floating @ 32 kHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>3.5</td>
<td>3.5</td>
<td>1.5 mA</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 2 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>7.0</td>
<td>7.0</td>
<td>3.8 mA</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 2 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>8.0</td>
<td>8.0</td>
<td>3.0 mA</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 8 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>11.0</td>
<td>11.0</td>
<td>4.4 mA</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 8 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>10</td>
<td>10</td>
<td>3.6 mA</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 12 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>15</td>
<td>15</td>
<td>9.0 mA</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 12 MHz</td>
</tr>
<tr>
<td>I_{CC}</td>
<td>Standby Current</td>
<td>3.0V</td>
<td>2.5</td>
<td>2.5</td>
<td>0.7 mA</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V$, $V_{CC} @ 2 MHz$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>4.0</td>
<td>5.0</td>
<td>2.5 mA</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V$, $V_{CC} @ 2 MHz$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>4.0</td>
<td>4.0</td>
<td>1.0 mA</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V$, $V_{CC} @ 8 MHz$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>5.0</td>
<td>5.0</td>
<td>3.0 mA</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V$, $V_{CC} @ 8 MHz$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>4.5</td>
<td>4.5</td>
<td>1.5 mA</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V$, $V_{CC} @ 12 MHz$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>7.0</td>
<td>7.0</td>
<td>4.0 mA</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V$, $V_{CC} @ 12 MHz$</td>
</tr>
<tr>
<td>I_{CC}</td>
<td>Supply Current (Low Noise Mode)</td>
<td>3.0V</td>
<td>3.5</td>
<td>3.5</td>
<td>1.5 mA</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 1 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>7.0</td>
<td>7.0</td>
<td>3.8 mA</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 1 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>5.8</td>
<td>5.8</td>
<td>2.5 mA</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 2 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>9.0</td>
<td>9.0</td>
<td>4.0 mA</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 2 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>8.0</td>
<td>8.0</td>
<td>3.0 mA</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 4 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>11.0</td>
<td>11.0</td>
<td>4.4 mA</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 4 MHz</td>
</tr>
</tbody>
</table>
## DC Electrical Characteristics (Continued)

<table>
<thead>
<tr>
<th>Sym</th>
<th>Parameter</th>
<th>$V_{CC}$ [4]</th>
<th>$T_A = 0°C$ to $+70°C$</th>
<th>$T_A = -40°C$ to $+105°C$</th>
<th>Typical @ 25°C</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$V_{CC}$</td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
<td>mA</td>
</tr>
<tr>
<td>$I_{CC1}$</td>
<td>Standby Current (Low Noise Mode)</td>
<td>3.0V</td>
<td>1.2</td>
<td>1.2</td>
<td>0.4</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V$, $V_{CC}$ @ 1 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>1.6</td>
<td>1.6</td>
<td>0.9</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V$, $V_{CC}$ @ 1 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>1.5</td>
<td>1.5</td>
<td>0.5</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V$, $V_{CC}$ @ 2 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>1.9</td>
<td>1.9</td>
<td>1</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V$, $V_{CC}$ @ 2 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>2.0</td>
<td>2.0</td>
<td>0.8</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V$, $V_{CC}$ @ 4 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>2.4</td>
<td>2.4</td>
<td>0.3</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V$, $V_{CC}$ @ 4 MHz</td>
</tr>
<tr>
<td>$I_{CC2}$</td>
<td>Standby Current</td>
<td>3.0V</td>
<td>10</td>
<td>20</td>
<td>1.0</td>
<td>µA</td>
<td>STOP mode $V_{IN} = 0V$, $V_{CC}$ WDT is not Running</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>10</td>
<td>20</td>
<td>1.0</td>
<td>µA</td>
<td>STOP mode $V_{IN} = 0V$, $V_{CC}$ WDT is not Running</td>
</tr>
<tr>
<td>$I_{ALH}$</td>
<td>Auto Latch High Current</td>
<td>3.0V</td>
<td>6.0</td>
<td>8.0</td>
<td>3.0</td>
<td>µA</td>
<td>$OV &lt; V_{IN} &lt; V_{CC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>22</td>
<td>30</td>
<td>16</td>
<td>µA</td>
<td>$OV &lt; V_{IN} &lt; V_{CC}$</td>
</tr>
<tr>
<td>$I_{ALH}$</td>
<td>Auto Latch High Current</td>
<td>3.0V</td>
<td>−4.0</td>
<td>−5.0</td>
<td>−1.5</td>
<td>µA</td>
<td>$OV &lt; V_{IN} &lt; V_{CC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>−12.0</td>
<td>−20</td>
<td>−8.0</td>
<td>µA</td>
<td>$OV &lt; V_{IN} &lt; V_{CC}$</td>
</tr>
</tbody>
</table>

### Notes:

1. $I_{CC1}$
   - For 2.75V operating, the device operates down to $V_{LV}$. The minimum operational $V_{CC}$ is determined on the value of the voltage $V_{LV}$ at the ambient temperature. The $V_{LV}$ increases as the temperature decreases.
   - $V_{CC} = 3.0V$ to 5.5V, typical values measured at $V_{CC} = 3.3V$ and $V_{CC} = 5.0V$.
   - Standard Mode (not Low EMI mode)
   - Z86C08 only.

2. $V_{SS} = 0V = GND$

3. $V_{CC} = 3.0V$ to 5.5V, typical values measured at $V_{CC} = 3.3V$ and $V_{CC} = 5.0V$.

4. $V_{CC} = 3.0V$ to 5.5V, typical values measured at $V_{CC} = 3.3V$ and $V_{CC} = 5.0V$.

5. Standard Mode (not Low EMI mode)

6. Z86C08 only.

7. CL1 = 100 pF, CL2 = 220 pF, RF = 30 kOhm
Figure 16. AC Electrical Timing Diagram
### AC ELECTRICAL CHARACTERISTICS

Timing Table (Standard Mode for SCLK/TCLK = XTAL/2)

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$</th>
<th>$T_a = 0°C$ to $+70°C$</th>
<th>$T_a = -40°C$ to $+105°C$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>8 MHz (C04)</td>
<td>12 MHz (C08)</td>
<td>8 MHz (C04)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>1</td>
<td>TpC</td>
<td>Input Clock Period</td>
<td>3.0V</td>
<td>125</td>
<td>DC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>125</td>
<td>DC</td>
</tr>
<tr>
<td>2</td>
<td>TrC,TIC</td>
<td>Clock Input Rise and Fall Times</td>
<td>3.0V</td>
<td>25</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>TwC</td>
<td>Input Clock Width</td>
<td>3.0V</td>
<td>62</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>62</td>
<td>41</td>
</tr>
<tr>
<td>4</td>
<td>TwTinL</td>
<td>Timer Input Low Width</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>5</td>
<td>TwTinH</td>
<td>Timer Input High Width</td>
<td>3.0V</td>
<td>5TpC</td>
<td>5TpC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
</tr>
<tr>
<td>6</td>
<td>TpTin</td>
<td>Timer Input Period</td>
<td>3.0V</td>
<td>8TpC</td>
<td>8TpC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>8TpC</td>
<td>8TpC</td>
</tr>
<tr>
<td>7</td>
<td>TrTin, TtTin</td>
<td>Timer Input Rise and Fall Timer</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>8</td>
<td>TwL</td>
<td>Int. Request Input Low Time</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>9</td>
<td>TwH</td>
<td>Int. Request Input High Time</td>
<td>3.0V</td>
<td>5TpC</td>
<td>5TpC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>Tpor</td>
<td></td>
<td>3.0V</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

**Notes:**

[1] Timing Reference uses 0.7 $V_{cc}$ for a logic 1 and 0.2 $V_{cc}$ for a logic 0.

[2] Interrupt request through Port 3 (P33-P31).
## AC ELECTRICAL CHARACTERISTICS
### Low Noise Mode

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{CC}$</th>
<th>1 MHz Min</th>
<th>1 MHz Max</th>
<th>4 MHz Min</th>
<th>4 MHz Max</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TPC</td>
<td>Input Clock Period</td>
<td>3.0V</td>
<td>1000</td>
<td>DC</td>
<td>250</td>
<td>DC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>1000</td>
<td>DC</td>
<td>250</td>
<td>DC</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TrC</td>
<td>Clock Input Rise</td>
<td>3.0V</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>and Fall Times</td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TwC</td>
<td>Input Clock Width</td>
<td>3.0V</td>
<td>500</td>
<td>125</td>
<td>500</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>500</td>
<td>125</td>
<td>500</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TwTinL</td>
<td>Timer Input Low Width</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>TwTinH</td>
<td>Timer Input High Width</td>
<td>3.0V</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>TpTin</td>
<td>Timer Input Period</td>
<td>3.0V</td>
<td>4TpC</td>
<td>4TpC</td>
<td>4TpC</td>
<td>4TpC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>4TpC</td>
<td>4TpC</td>
<td>4TpC</td>
<td>4TpC</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TrTin,</td>
<td>Timer Input Rise</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TtTin</td>
<td>and Fall Timer</td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>TwIL</td>
<td>Int. Request Input Low</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time</td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>TwIH</td>
<td>Int. Request Input High</td>
<td>3.0V</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time</td>
<td>5.5V</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Twdt</td>
<td>Watch-Dog Timer</td>
<td>3.0V</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Delay Time</td>
<td>5.5V</td>
<td>12</td>
<td>12</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

### Notes:
[1] Timing Reference uses $0.7V_{CC}$ for a logic 1 and $0.2V_{CC}$ for a logic 0.
[2] Interrupt request through Port 3 (P33-P31).
LOW NOISE VERSION

Low EMI Emission

The Z86C04/C08 can be programmed to operate in a Low EMI emission mode by means of a mask ROM bit option. Use of this feature results in:

- All pre-driver slew rates reduced to 10 ns typical.
- Internal SCLK/TCLK operation limited to a maximum of 4 MHz - 250 ns cycle time.
- Output drivers have resistances of 200 ohms (typical).
- Oscillator divide-by-two circuitry eliminated.

The Low EMI mode is mask-programmable to be selected by the customer at the time the ROM code is submitted.

EMI Characteristics

The Z86C04/C08 operating in the Low EMI mode generates EMI as measured in the following chart:

The measurements, shown in Figure 17, were made while operating the Z86C04/C08 in three states: (1) idle condition; (2) static output; (3) switched output.

Near Field EMI Analysis

Z86C04/C08 Low Noise

Figure 17. Low Noise Analysis
Z8® CONTROL REGISTER DIAGRAMS

Figure 18. Timer Mode Register (F1H: Read/Write)

Figure 19. Counter Time 1 Register (F2H: Read/Write)

Figure 20. Prescaler 1 Register (F3H: Write Only)

Figure 21. Counter/Timer 0 Register (F4H: Read/Write)

Figure 22. Prescaler 0 Register (F5H: Write Only)

Figure 23. Port 2 Mode Register (F6H: Write Only)

Figure 24. Port 3 Mode Register (F7H: Write Only)
Z8 CONTROL REGISTER DIAGRAMS (Continued)

Figure 25. Port 0 and 1 Mode Register (F8H: Write Only)

Figure 26. Interrupt Priority Register (F9H: Write Only)

Figure 27. Interrupt Request Register (FAH: Read/Write)

Figure 28. Interrupt Mask Register (FBH: Read/Write)

Figure 29. Flag Register (FCH: Read/Write)

Figure 30. Register Pointer (FDH: Read/Write)

Figure 31. Stack Pointer (FFH: Read/Write)
DEVICE CHARACTERISTICS
Standard Mode

Figure 32. Typical Icc vs Frequency

Figure 33. VIL, VOH vs. Temperature
DEVICE CHARACTERISTICS (Continued)

Standard Mode

Figure 34. $V_{IH}, V_{OH}$ vs. Temperature

Figure 35. Typical $I_{OH}$ vs. $V_{OH}$
Figure 36. Typical $I_{OL}$ vs. $V_{OL}$

Figure 37. Typical WDT Time Out Period vs $V_{CC}$ Over Temperature
INSTRUCTION SET NOTATION

Addressing Modes. The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRR</td>
<td>Indirect register pair or indirect working-</td>
</tr>
<tr>
<td></td>
<td>register pair address</td>
</tr>
<tr>
<td>Ir</td>
<td>Indirect working-register pair only</td>
</tr>
<tr>
<td>X</td>
<td>Indexed address</td>
</tr>
<tr>
<td>DA</td>
<td>Direct address</td>
</tr>
<tr>
<td>RA</td>
<td>Relative address</td>
</tr>
<tr>
<td>IM</td>
<td>Immediate</td>
</tr>
<tr>
<td>R</td>
<td>Register or working-register address</td>
</tr>
<tr>
<td>r</td>
<td>Working register address only</td>
</tr>
<tr>
<td>IR</td>
<td>Indirect-register or indirect working-</td>
</tr>
<tr>
<td></td>
<td>register address</td>
</tr>
<tr>
<td>Ir</td>
<td>Indirect working-register address only</td>
</tr>
<tr>
<td>RR</td>
<td>Register pair or working register pair</td>
</tr>
<tr>
<td></td>
<td>address</td>
</tr>
</tbody>
</table>

Symbols. The following symbols are used in describing the instruction set.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst</td>
<td>Destination location or contents</td>
</tr>
<tr>
<td>src</td>
<td>Source location or contents</td>
</tr>
<tr>
<td>cc</td>
<td>Condition code</td>
</tr>
<tr>
<td>@</td>
<td>Indirect address prefix</td>
</tr>
<tr>
<td>SP</td>
<td>Stack pointer</td>
</tr>
<tr>
<td>PC</td>
<td>Program counter</td>
</tr>
<tr>
<td>FLAGS</td>
<td>Flag register (Control Register 252)</td>
</tr>
<tr>
<td>RP</td>
<td>Register Pointer (R253)</td>
</tr>
<tr>
<td>IMR</td>
<td>Interrupt mask register (R251)</td>
</tr>
</tbody>
</table>

Flags. Control register (R252) contains the following six flags.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Carry flag</td>
</tr>
<tr>
<td>Z</td>
<td>Zero flag</td>
</tr>
<tr>
<td>S</td>
<td>Sign flag</td>
</tr>
<tr>
<td>V</td>
<td>Overflow flag</td>
</tr>
<tr>
<td>D</td>
<td>Decimal-adjust flag</td>
</tr>
<tr>
<td>H</td>
<td>Half-carry flag</td>
</tr>
</tbody>
</table>

Affected flags are indicated by:

- 0 Clear to zero
- 1 Set to one
- * Set to clear according to operation
- - Unaffected
- X Undefined
## CONDITION CODES

<table>
<thead>
<tr>
<th>Value</th>
<th>Mnemonic</th>
<th>Meaning</th>
<th>Flags Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>—</td>
<td>Always true</td>
<td>—</td>
</tr>
<tr>
<td>0111</td>
<td>C</td>
<td>Carry</td>
<td>C = 1</td>
</tr>
<tr>
<td>1111</td>
<td>NC</td>
<td>No Carry</td>
<td>C = 0</td>
</tr>
<tr>
<td>0110</td>
<td>Z</td>
<td>Zero</td>
<td>Z = 1</td>
</tr>
<tr>
<td>1110</td>
<td>NZ</td>
<td>Not zero</td>
<td>Z = 0</td>
</tr>
<tr>
<td>1101</td>
<td>PL</td>
<td>Plus</td>
<td>S = 1</td>
</tr>
<tr>
<td>0101</td>
<td>MI</td>
<td>Minus</td>
<td>S = 0</td>
</tr>
<tr>
<td>0100</td>
<td>OV</td>
<td>Overflow</td>
<td>V = 1</td>
</tr>
<tr>
<td>1100</td>
<td>NOV</td>
<td>No overflow</td>
<td>V = 0</td>
</tr>
<tr>
<td>0110</td>
<td>EQ</td>
<td>Equal</td>
<td>Z = 1</td>
</tr>
<tr>
<td>1110</td>
<td>NE</td>
<td>Not equal</td>
<td>Z = 0</td>
</tr>
<tr>
<td>1001</td>
<td>GE</td>
<td>Greater than or equal</td>
<td>(S XOR V) = 0</td>
</tr>
<tr>
<td>0001</td>
<td>LT</td>
<td>Less than</td>
<td>(S XOR V) = 1</td>
</tr>
<tr>
<td>1010</td>
<td>GT</td>
<td>Greater than</td>
<td>[Z OR (S XOR V)] = 0</td>
</tr>
<tr>
<td>0010</td>
<td>LE</td>
<td>Less than or equal</td>
<td>[Z OR (S XOR V)] = 1</td>
</tr>
<tr>
<td>1111</td>
<td>UGE</td>
<td>Unsigned greater than or equal</td>
<td>C = 0</td>
</tr>
<tr>
<td>0111</td>
<td>ULT</td>
<td>Unsigned less than</td>
<td>C = 1</td>
</tr>
<tr>
<td>1011</td>
<td>UGT</td>
<td>Unsigned greater than</td>
<td>(C = 0 AND Z = 0) = 1</td>
</tr>
<tr>
<td>0011</td>
<td>ULE</td>
<td>Unsigned less than or equal</td>
<td>(C OR Z) = 1</td>
</tr>
<tr>
<td>0000</td>
<td>F</td>
<td>Never true (always false)</td>
<td>—</td>
</tr>
</tbody>
</table>
INSTRUCTION FORMATS

One-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst/src</th>
<th>1110</th>
<th>dst/src</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>MODE</td>
<td>dst</td>
<td>1110</td>
<td>dst</td>
</tr>
<tr>
<td>OPC</td>
<td>VALUE</td>
<td>src</td>
<td>1110</td>
<td>src</td>
</tr>
<tr>
<td>OPC</td>
<td>src</td>
<td>1110</td>
<td>src</td>
<td></td>
</tr>
</tbody>
</table>

Two-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst/src</th>
<th>1110</th>
<th>dst/src</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>MODE</td>
<td>dst</td>
<td>1110</td>
<td>dst</td>
</tr>
<tr>
<td>OPC</td>
<td>VALUE</td>
<td>src</td>
<td>1110</td>
<td>src</td>
</tr>
<tr>
<td>OPC</td>
<td>src</td>
<td>1110</td>
<td>src</td>
<td></td>
</tr>
</tbody>
</table>

Three-Byte Instructions

INSTRUCTION SUMMARY

Note: Assignment of a value is indicated by the symbol "←". For example:

dst ← dst + src

indicates that the source data is added to the destination data and the result is stored in the destination location. The
### INSTRUCTION SUMMARY (Continued)

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
<th>Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC dst, src</td>
<td>+</td>
<td>1[ ]</td>
<td>* * * * 0 *</td>
<td>dst ← dst + src +C</td>
<td>INC dst</td>
<td>r</td>
<td>rE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>dst ← dst + 1</td>
<td></td>
<td>r = 0 - F</td>
</tr>
<tr>
<td>ADD dst, src</td>
<td>+</td>
<td>0[ ]</td>
<td>* * * * 0 *</td>
<td>dst ← dst + src</td>
<td>R</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>AND dst, src</td>
<td>+</td>
<td>5[ ]</td>
<td>- * * 0</td>
<td>dst ← dst AND src</td>
<td>IR</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>CALL dst</td>
<td>DA</td>
<td>D6</td>
<td>- - - - -</td>
<td>SP ← SP - 2</td>
<td>CALL dst</td>
<td>DA</td>
<td>cD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D4</td>
<td></td>
<td>@SP ← PC,</td>
<td></td>
<td></td>
<td>c = 0 - F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PC ← dst</td>
<td></td>
<td></td>
<td>PC ← dst</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Range: 127, -128</td>
</tr>
<tr>
<td>CCF</td>
<td>EF</td>
<td>* - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLR dst</td>
<td>R</td>
<td>80</td>
<td>- - - - -</td>
<td>dst ← 0</td>
<td>CLR dst</td>
<td>R</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>IR</td>
<td>B1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COM dst</td>
<td>R</td>
<td>60</td>
<td>- * * 0</td>
<td>dst ← NOT dst</td>
<td>COM dst</td>
<td>R</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>IR</td>
<td>61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP dst, src</td>
<td>+</td>
<td>A[ ]</td>
<td>* * * * - -</td>
<td>dst ← src</td>
<td>CP dst, src</td>
<td>+</td>
<td>A[ ]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DA dst</td>
<td>R</td>
<td>40</td>
<td>* * * X</td>
<td>dst ← DA dst</td>
<td>DA dst</td>
<td>R</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEC dst</td>
<td>R</td>
<td>00</td>
<td>- * * - -</td>
<td>dst ← dst - 1</td>
<td>DEC dst</td>
<td>R</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>IR</td>
<td>01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECW dst</td>
<td>RR</td>
<td>80</td>
<td>- * * - -</td>
<td>dst ← dst - 1</td>
<td>DECW dst</td>
<td>RR</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>IR</td>
<td>81</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DJNZ r, dst</td>
<td>ra</td>
<td>rA</td>
<td>- - - - -</td>
<td>r ← r - 1</td>
<td>DJNZ r, dst</td>
<td>ra</td>
<td>rA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>if r ≠ 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PC ← PC + dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Range: +127, -128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EL</td>
<td></td>
<td>9F</td>
<td>- - - - -</td>
<td></td>
<td>EL</td>
<td>9F</td>
<td>- - - - -</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HALT</td>
<td></td>
<td>7F</td>
<td>- - - - -</td>
<td></td>
<td>HALT</td>
<td>7F</td>
<td>- - - - -</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRET</td>
<td>BF</td>
<td>* * * * * *</td>
<td></td>
<td></td>
<td>IRET</td>
<td>BF</td>
<td>* * * * * *</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JP cc, dst</td>
<td>DA</td>
<td>cD</td>
<td>- - - - -</td>
<td></td>
<td>JP cc, dst</td>
<td>DA</td>
<td>cD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>if cc is true,</td>
<td></td>
<td></td>
<td>cc = 0 - F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PC ← dst</td>
<td></td>
<td></td>
<td>PC ← dst</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Range: 127, -128</td>
</tr>
<tr>
<td>JR cc, dst</td>
<td>RA</td>
<td>cB</td>
<td>- - - - -</td>
<td></td>
<td>JR cc, dst</td>
<td>RA</td>
<td>cB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>if cc is true,</td>
<td></td>
<td></td>
<td>cc = 0 - F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PC ← PC + dst</td>
<td></td>
<td></td>
<td>PC ← PC + dst</td>
</tr>
<tr>
<td>LD dst, src</td>
<td>r</td>
<td>rIm</td>
<td>rC</td>
<td>dst ← src</td>
<td>LD dst, src</td>
<td>r</td>
<td>rIm</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### INSTRUCTION SUMMARY (Continued)

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected C Z S V D H</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR dst, src dst ← dst OR src</td>
<td>†</td>
<td>4[</td>
<td>– * * 0 – –</td>
</tr>
<tr>
<td>POP dst dst ← @SP; SP ← SP + 1</td>
<td>R</td>
<td>50</td>
<td>– – – – – –</td>
</tr>
<tr>
<td>PUSH src SP ← SP – 1; @SP ← src</td>
<td>R</td>
<td>70</td>
<td>– – – – – –</td>
</tr>
<tr>
<td>RCF</td>
<td>CF</td>
<td>0</td>
<td>– – – – – –</td>
</tr>
<tr>
<td>RET</td>
<td>AF</td>
<td>– – – – – –</td>
<td></td>
</tr>
<tr>
<td>PC ← @SP; SP ← SP + 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RL dst R</td>
<td>90</td>
<td>* * * – –</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IR</td>
<td>91</td>
<td></td>
</tr>
<tr>
<td>RLC dst R</td>
<td>10</td>
<td>* * * – –</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IR</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>RR dst R</td>
<td>E0</td>
<td>* * * – –</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IR</td>
<td>E1</td>
<td></td>
</tr>
<tr>
<td>RRC dst R</td>
<td>C0</td>
<td>* * * – –</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IR</td>
<td>C1</td>
<td></td>
</tr>
<tr>
<td>SBC dst, src dst ← dst – src – C</td>
<td>†</td>
<td>3[</td>
<td>* * * 1 *</td>
</tr>
<tr>
<td>SCF</td>
<td>DF</td>
<td>1</td>
<td>– – – – – –</td>
</tr>
<tr>
<td>SRA dst R</td>
<td>D0</td>
<td>* * * 0 – –</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IR</td>
<td>D1</td>
<td></td>
</tr>
<tr>
<td>SRP dst Im</td>
<td>31</td>
<td>– – – – – –</td>
<td></td>
</tr>
<tr>
<td>RP ← src</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected C Z S V D H</th>
</tr>
</thead>
<tbody>
<tr>
<td>STOP</td>
<td>6F</td>
<td>1</td>
<td>– – – – – –</td>
</tr>
<tr>
<td>SUB dst, src dst ← dst – src</td>
<td>†</td>
<td>2[</td>
<td>* * * 1 *</td>
</tr>
<tr>
<td>SWAP dst R</td>
<td>F0</td>
<td>X * X – –</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IR</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>TCM dst, src (NOT dst) AND src</td>
<td>†</td>
<td>6[</td>
<td>* * 0 – –</td>
</tr>
<tr>
<td>TM dst, src dst AND src</td>
<td>†</td>
<td>7[</td>
<td>* * 0 – –</td>
</tr>
<tr>
<td>WDH</td>
<td>4F</td>
<td>– – – – – –</td>
<td></td>
</tr>
<tr>
<td>WDT</td>
<td>5F</td>
<td>– X X X – –</td>
<td></td>
</tr>
<tr>
<td>XOR dst, src dst ← dst XOR src</td>
<td>†</td>
<td>B[</td>
<td>* * 0 – –</td>
</tr>
</tbody>
</table>

† These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a ‘[ ]’ in this table, and its value is found in the following table to the left of the applicable addressing mode pair.

For example, the opcode of an ADC instruction using the addressing modes r (destination) and Ir (source) is 13.

<table>
<thead>
<tr>
<th>Address Mode</th>
<th>Lower Opcode Nibble</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst src</td>
<td></td>
</tr>
<tr>
<td>r r</td>
<td>[2]</td>
</tr>
<tr>
<td>r Ir</td>
<td>[3]</td>
</tr>
<tr>
<td>R R</td>
<td>[4]</td>
</tr>
<tr>
<td>R Ir</td>
<td>[5]</td>
</tr>
<tr>
<td>R IM</td>
<td>[6]</td>
</tr>
<tr>
<td>IR IM</td>
<td>[7]</td>
</tr>
</tbody>
</table>
### OPCODE MAP

#### Execution Cycles

<table>
<thead>
<tr>
<th>Upper Nibble (Hex)</th>
<th>Lower Nibble (Hex)</th>
<th>Bytes per Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Upper Nibble (Hex)</th>
<th>Lower Nibble (Hex)</th>
<th>Bytes per Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>

#### Legend:
- R = 8-bit Address
- r = 4-bit Address
- R1 or r1 = Dest Address
- R2 or r2 = Src Address

#### Sequence:
- Opcode, First Operand, Second Operand

#### Note:
- Blank areas are reserved.
- *2-byte instruction appears as a 3-byte instruction
### PACKAGE INFORMATION

**18-Pin DIP Package Diagram**

**18-Pin SOIC Package Diagram**

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MILLIMETER</th>
<th>INCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>0.51</td>
<td>0.02</td>
</tr>
<tr>
<td>A2</td>
<td>3.25</td>
<td>1.28</td>
</tr>
<tr>
<td>B</td>
<td>0.30</td>
<td>0.12</td>
</tr>
<tr>
<td>B1</td>
<td>1.14</td>
<td>0.45</td>
</tr>
<tr>
<td>C</td>
<td>0.23</td>
<td>0.01</td>
</tr>
<tr>
<td>D</td>
<td>22.35</td>
<td>8.80</td>
</tr>
<tr>
<td>E</td>
<td>7.62</td>
<td>3.00</td>
</tr>
<tr>
<td>E1</td>
<td>6.22</td>
<td>2.45</td>
</tr>
<tr>
<td>mA</td>
<td>2.54</td>
<td>1.00</td>
</tr>
<tr>
<td>L</td>
<td>3.10</td>
<td>1.25</td>
</tr>
<tr>
<td>GI</td>
<td>1.52</td>
<td>0.60</td>
</tr>
<tr>
<td>S</td>
<td>0.89</td>
<td>0.35</td>
</tr>
</tbody>
</table>

**CONTROLLING DIMENSIONS: INCH**

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MILLIMETER</th>
<th>INCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2.50</td>
<td>0.99</td>
</tr>
<tr>
<td>A1</td>
<td>0.10</td>
<td>0.04</td>
</tr>
<tr>
<td>A2</td>
<td>2.24</td>
<td>0.89</td>
</tr>
<tr>
<td>B</td>
<td>0.36</td>
<td>0.14</td>
</tr>
<tr>
<td>C</td>
<td>0.23</td>
<td>0.01</td>
</tr>
<tr>
<td>D</td>
<td>1.14</td>
<td>0.44</td>
</tr>
<tr>
<td>E</td>
<td>7.40</td>
<td>2.91</td>
</tr>
<tr>
<td>mA</td>
<td>1.27</td>
<td>0.50</td>
</tr>
<tr>
<td>H</td>
<td>10.00</td>
<td>3.94</td>
</tr>
<tr>
<td>h</td>
<td>0.30</td>
<td>0.12</td>
</tr>
<tr>
<td>L</td>
<td>0.60</td>
<td>0.24</td>
</tr>
<tr>
<td>GI</td>
<td>0.97</td>
<td>0.38</td>
</tr>
</tbody>
</table>

**CONTROLLING DIMENSIONS: MM**

Leads are coplanar within .004 inch.
### ORDERING INFORMATION

#### Z86C04 (8 MHz)

<table>
<thead>
<tr>
<th>Standard Temperature</th>
<th>Extended Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>18-Pin DIP</td>
<td>18-Pin DIP</td>
</tr>
<tr>
<td>Z86C0408PSC</td>
<td>Z86C0408PEC</td>
</tr>
<tr>
<td>18-Pin SOIC</td>
<td>18-Pin SOIC</td>
</tr>
<tr>
<td>Z86C0408SSC</td>
<td>Z86C0408SEC</td>
</tr>
</tbody>
</table>

#### Z86C08 (12 MHz)

<table>
<thead>
<tr>
<th>Standard Temperature</th>
<th>Extended Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>18-Pin DIP</td>
<td>18-Pin DIP</td>
</tr>
<tr>
<td>Z86C0812PSC</td>
<td>Z86C0812PEC</td>
</tr>
<tr>
<td>18-Pin SOIC</td>
<td>18-Pin SOIC</td>
</tr>
<tr>
<td>Z86C0812SSC</td>
<td>Z86C0812SEC</td>
</tr>
</tbody>
</table>

For fast results, contact your local Zilog sale offices for assistance in ordering the part(s) desired.

### CODES

**Preferred Package**  
P = DIP

**Longer Lead Time**  
S = SOIC

**Preferred Temperature**  
S = 0°C to +70°C

**Longer Lead Time**  
E = -40°C to +105°C

**Speeds**  
08 = 8 MHz  
12 = 12 MHz

**Environmental**  
C = Plastic Standard

**Example:**  
Z86C04 08 P S C

is a Z86C04, 8 MHz, DIP, 0°C to +70°C, Plastic Standard Flow

- Environmental Flow  
- Temperature  
- Package  
- Speed  
- Product Number  
- Zilog Prefix
Z86C03/C06 CMOS Z8® 8-Bit CCP™ Consumer Controller Processors

Z86E03/E06 CMOS Z8® 8-Bit OTP CCP™ Consumer Controller Processors

Z86C04/C08 CMOS Z8® Low Cost 1K /2K ROM Microcontrollers

Z86E04/E08 CMOS Z8® 8-Bit OTP Microcontrollers

Z86C07 CMOS Z8® 8-Bit Microcontroller

Z86E07 CMOS Z8® 8-Bit OTP Microcontroller

Z86C30/C31 CMOS Z8® 8-Bit CCP™ Consumer Controller Processors
FEATURES

- The Z86E04/E08 Devices Have the Following General Characteristics:

<table>
<thead>
<tr>
<th>Part</th>
<th>ROM</th>
<th>RAM</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z86E04</td>
<td>1 Kbyte</td>
<td>124</td>
<td>8 MHz</td>
</tr>
<tr>
<td>Z86E08</td>
<td>2 Kbyte</td>
<td>124</td>
<td>12 MHz</td>
</tr>
</tbody>
</table>

- 18-Pin Package (DIP, SOIC)
- Clock Speeds up to 8 MHz (E04), 12 MHz (E08)
- Low Noise Programmable
- ROM Protect Programmable
- 4.5V to 5.5V Operating Range
- Low Power Consumption: 50 mW (Typical)
- Fast Instruction Pointer: 1 µs @ 12 MHz (E08), 1.25 µs @ 8 MHz (E04)
- Two Standby Modes: STOP and HALT

14 Input/Output Lines
All Digital Inputs, CMOS Levels, Schmitt-Triggered.
Two Programmable 8-Bit Counter/Timers Each with a 6-Bit Programmable Prescaler.
Six Vectored, Priority Interrupts from Six Different Sources.
Programmable Watch-Dog Timer
Power-On Reset Timer
Two On-Board Comparators
On-Chip Oscillator that Accepts a Crystal, Ceramic Resonator, LC, or External Clock Drive.
Programmable Interrupt Polarity
Auto Latches

GENERAL DESCRIPTION

The Z86E04/E08 8-bit One-Time-Programmable (OTP) Microcontrollers (MCUs) are members of Zilog’s single-chip Z8® microcontroller family with 1K/2K bytes of one-time PROM, respectively. Offered in 18-pin DIP or SOIC style packages, and manufactured in CMOS technology, the Z86E04/E08 allow easy software development, debug and prototyping, and are ideal for small production runs not economically desirable with a masked ROM version.

The Z86E04/E08 are characterized by a flexible I/O scheme, an efficient register, and address space structure, in addition to a number of ancillary features useful in many consumer, industrial, and commercial applications.

For applications demanding powerful I/O capabilities, the Z86E04/E08 provides 14 pins dedicated to input and output. These lines are grouped into three ports, and are configurable under software control to provide I/O, timing, and status signals. There are two basic address spaces available to support this configuration: program memory and 124 bytes of general-purpose registers.

The Z86E04/E08 each offer programmable EPROM Protect and programmable Low Noise. When the part is programmed for EPROM Protect, the Low Noise feature will automatically be enabled. When programmed for Low Noise, the EPROM Protect feature is optional.
GENERAL DESCRIPTION (Continued)

To unburden the system from coping with real-time tasks, such as counting/timing and I/O data communications, the Z86E04/E08 offers two on-chip counter/timers with a large number of user selectable modes. Included, are two on-board comparators that can process analog signals with a common reference voltage (Figures 1 and 2).

Notes:
All Signals with a preceding front slash, "/", are active Low, e.g.: B/W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

<table>
<thead>
<tr>
<th>Connection</th>
<th>Circuit</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>V_{cc}</td>
<td>V_{dd}</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
<td>V_{ss}</td>
</tr>
</tbody>
</table>

Figure 1. Z86E04/E08 Functional Block Diagram
Figure 2. Z86E04/E08 EPROM Programming Mode Block Diagram

PIN DESCRIPTION

Table 1. 18-Pin DIP/SOIC Pin Identification

<table>
<thead>
<tr>
<th>EPROM Programming Mode</th>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin #</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-4</td>
<td>1-4</td>
<td>D7-D4</td>
<td>Data 4, 5, 6, 7</td>
<td>In/Output</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Vcc</td>
<td>Power Supply</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>N/C</td>
<td>No Connection</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>/CE</td>
<td>Chip Enable</td>
<td>Input</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>/OE</td>
<td>Output Enable</td>
<td>Input</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>EPM</td>
<td>EPROM Prog Mode</td>
<td>Input</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>Vpp</td>
<td>Prog Voltage</td>
<td>Input</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>Clear</td>
<td>Clear Clock</td>
<td>Input</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>Clock</td>
<td>Address</td>
<td>Input</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>/PGM</td>
<td>Prog Mode</td>
<td>Input</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>15-18</td>
<td>15-18</td>
<td>D3-D0</td>
<td>Data 0, 1, 2, 3</td>
<td>In/Output</td>
</tr>
</tbody>
</table>

Figure 3. 18-Pin DIP/SOIC Pin Configuration
EPROM Programming Mode
PIN DESCRIPTION

Figure 4. 18-Pin DIP Pin Configuration

Figure 5. 18-Pin SOIC Pin Configuration

Table 2. 18-Pin DIP/SOIC Pin Identification

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>P24-P27</td>
<td>Port 2, Pins 4,5,6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>5</td>
<td>Vcc</td>
<td>Power Supply</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>XTAL2</td>
<td>Crystal Osc. Clock</td>
<td>Output</td>
</tr>
<tr>
<td>7</td>
<td>XTAL1</td>
<td>Crystal Osc. Clock</td>
<td>Input</td>
</tr>
<tr>
<td>8</td>
<td>P31</td>
<td>Port 3, Pin 1, AN1</td>
<td>Input</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>P32</td>
<td>Port 3, Pin 2, AN2</td>
<td>Input</td>
</tr>
<tr>
<td>10</td>
<td>P33</td>
<td>Port 3, Pin 3, REF</td>
<td>Input</td>
</tr>
<tr>
<td>11-13</td>
<td>P00-P02</td>
<td>Port 0, Pins 0,1,2</td>
<td>In/Output</td>
</tr>
<tr>
<td>14</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>15-18</td>
<td>P20-P23</td>
<td>Port 2, Pins 0,1,2,3</td>
<td>In/Output</td>
</tr>
</tbody>
</table>
PIN FUNCTIONS

OTP Programming Mode

**D7-D0 Data Bus.** Data can be read from, or written to the EPROM through this data bus.

**V<sub>cc</sub> Power Supply.** It is 5V during EPROM Read Mode and 6V during the other modes (Program, Program Verify, etc.).

**/CE Chip Enable (active Low).** This pin is active during EPROM Read Mode, Program Mode, and Program Verify Mode.

**/OE Output Enable (active Low).** This pin drives the Data Bus direction. When this pin is Low, the Data Bus is output. When High, the Data Bus is input.

**EPM EPROM Program Mode.** This pin controls the different EPROM Program Modes by applying different voltages.

**V<sub>pp</sub> Program Voltage.** This pin supplies the program voltage.

**Clear Clear (active High).** This pin resets the internal address counter at the High Level.

**Clock Address Clock.** This pin is a clock input. The internal address counter increases by one with one clock cycle.

**/PGM Program Mode (active Low).** A Low level at this pin programs the data to the EPROM through the Data Bus.

Application Precaution

The production test-mode environment may be enabled accidentally during normal operation if *excessive noise* surges above V<sub>cc</sub> occur on the XTAL1 pin.

In addition, processor operation of Z8 OTP devices may be affected by *excessive noise* surges on the V<sub>pp</sub>, /CE, /EPM, /OE pins while the microcontroller is in Standard Mode.

Recommendations for dampening voltage surges in both test and OTP mode include the following:

- Using a clamping diode to V<sub>cc</sub>
- Adding a capacitor to the affected pin.
Z86E04/E08 STANDARD MODE

XTAL1, XTAL2 Crystal In, Crystal Out (time-based input and output, respectively). These pins connect a parallel-resonant crystal, LC, or an external single-phase clock (8 MHz or 12 MHz max) to the on-chip clock oscillator and buffer.

Port 0, P02-P00. Port 0 is a 3-bit bi-directional, Schmitt-triggered CMOS compatible I/O port. These three I/O lines can be globally configured under software control to be inputs or outputs (Figure 6).

Auto Latch. The Auto Latch puts valid CMOS levels on all CMOS inputs (except P33, P32, P31) that are not externally driven. Whether this level is 0 or 1 cannot be determined. A valid CMOS level, rather than a floating node, reduces excessive supply current flow in the input buffer.

Figure 6. Port 0 Configuration
**Port 2, P27-P20.** Port 2 is an 8-bit, bit programmable, bi-directional, Schmitt-triggered CMOS compatible I/O port. These eight I/O lines can be configured under software control to be inputs or outputs, independently. Bits programmed as outputs can be globally programmed as either push-pull or open-drain (Figure 7).

**Figure 7. Port 2 Configuration**
Port 3, P33-P31. Port 3 is a 3-bit, CMOS compatible port with three fixed input (P32-P30) lines. These three input lines can be configured under software control as digital inputs or analog inputs. These three input lines are also used as the interrupt sources IRQ0-IRQ3 and as the timer input signal (T1N - Figure 8).

**Figure 8. Port 3 Configuration**
Comparator Inputs. Two analog comparators are added to input of Port 3, P31 and P32, for interface flexibility. The comparators reference voltage P3 (REF) is common to both comparators.

Typical applications for the on-board comparators; Zero crossing detection, A/D conversion, voltage scaling, and threshold detection. In analog mode, P33 input functions serve as a reference voltage to the comparators.

The dual comparator (common inverting terminal) features a single power supply which discontinues power in STOP mode. The common voltage range is 0-4 V when the \( V_{CC} \) is 5.0 V; the power supply and common mode rejection ratios are 90 dB and 60 dB, respectively.

Interrupts are generated on either edge of Comparator 2’s output, or on the falling edge of Comparator 1’s output. The comparator output is used for interrupt generation, Port 3 data inputs, or \( T_{P} \) through P31. Alternatively, the comparators can be disabled, freeing the reference input (P33) for use as IRQ1 and/or P33 input.

FUNCTIONAL DESCRIPTION

The following special functions have been incorporated into the Z86E04/E08 devices to enhance the standard Z8 core architecture to provide the user with increased design flexibility.

RESET. This function is accomplished by means of a Power-On Reset or a Watch-Dog Timer Reset. Upon power-up, the Power-On Reset circuit waits for \( T_{POR} \) ms, plus 18 clock cycles, then starts program execution at address 000C (Hex) (Figure 9). The Z86E04/E08 control registers’ reset value is shown in Table 3.

![Figure 9. Internal Reset Configuration](image)

Power-On Reset (POR). A timer circuit clocked by a dedicated on-board RC oscillator is used for a POR timer function. The POR time allows \( V_{CC} \) and the oscillator circuit to stabilize before instruction execution begins. The POR timer circuit is a one-shot timer triggered by one of the four following conditions:

- Power bad to power good status
- Stop-Mode Recovery
- WDT time-out
- WDH time-out

Watch-Dog Timer Reset. The WDT is a retriggerable one-shot timer that resets the Z8 if it reaches its terminal count. The WDT is initially enabled by executing the WDT instruction and is retriggered on subsequent execution of the WDT instruction. The timer circuit is driven by an on-board RC oscillator.
**FUNCTIONAL DESCRIPTION (Continued)**

**Table 3. Z86E04/E08 Control Registers**

<table>
<thead>
<tr>
<th>Addr.</th>
<th>Reg.</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>TMR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>F2</td>
<td>T1</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>F3</td>
<td>PRE1</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>F4</td>
<td>T0</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>F5</td>
<td>PRE0</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>F6*</td>
<td>P2M</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Inputs after</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>reset</td>
</tr>
<tr>
<td>F7*</td>
<td>P3M</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>F8*</td>
<td>P01M</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>F9</td>
<td>IPR</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>FA</td>
<td>IRQ</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>IRQ3 is used for</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>positive edge</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>detection</td>
</tr>
<tr>
<td>FB</td>
<td>IMR</td>
<td>0</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>FC</td>
<td>FLAGS</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>FD</td>
<td>RP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>FF</td>
<td>SPL</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

- Registers are not reset after a STOP-Mode Recovery using P27 pin. A subsequent reset will cause these control registers to be reconfigured as shown in Table 4 and the user must avoid bus contention on the port pins or it may affect device reliability.

**Program Memory.** The Z86E04/E08 addresses up to 1K/2K bytes of internal program memory (Figure 10). The first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts. Bytes 0-1024/2048 are on-chip one-time programmable ROM.

![Program Memory Map](image-url)

**Figure 10. Program Memory Map**
Register File. The Register File consists of three I/O port registers, 124 general-purpose registers, and 14 control and status registers R0-R3, R4-R127 and R241-R255, respectively (Figure 11). General-purpose registers occupy the 04H to 7FH address space. I/O ports are mapped as per the existing CMOS Z8. The Z86E04/E08 instructions can access registers directly or indirectly through an 8-bit address field. This allows short 4-bit register addressing using the Register Pointer. In the 4-bit mode, the register file is divided into eight working register groups, each occupying 16 continuous locations. The Register Pointer (Figure 12) addresses the starting location of the active working-register group.

<table>
<thead>
<tr>
<th>Location</th>
<th>Identifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>255</td>
<td>Stack Pointer (Bits 7-0)</td>
</tr>
<tr>
<td>254</td>
<td>General-Purpose Register</td>
</tr>
<tr>
<td>253</td>
<td>Register Pointer</td>
</tr>
<tr>
<td>252</td>
<td>Program Control Flags</td>
</tr>
<tr>
<td>251</td>
<td>Interrupt Mask Register</td>
</tr>
<tr>
<td>250</td>
<td>Interrupt Request Register</td>
</tr>
<tr>
<td>249</td>
<td>Interrupt Priority Register</td>
</tr>
<tr>
<td>248</td>
<td>Ports 0-1 Mode</td>
</tr>
<tr>
<td>247</td>
<td>Port 3 Mode</td>
</tr>
<tr>
<td>246</td>
<td>Port 2 Mode</td>
</tr>
<tr>
<td>245</td>
<td>T0 Prescaler</td>
</tr>
<tr>
<td>244</td>
<td>Timer/Counter0</td>
</tr>
<tr>
<td>243</td>
<td>T1 Prescaler</td>
</tr>
<tr>
<td>242</td>
<td>Timer/Counter1</td>
</tr>
<tr>
<td>241</td>
<td>Timer Mode</td>
</tr>
<tr>
<td>128</td>
<td>Not Implemented</td>
</tr>
<tr>
<td>127</td>
<td>General-Purpose Registers</td>
</tr>
<tr>
<td>4</td>
<td>Port 3</td>
</tr>
<tr>
<td>3</td>
<td>Port 2</td>
</tr>
<tr>
<td>2</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>Port 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>Identifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Port 0</td>
</tr>
</tbody>
</table>

Figure 11. Register File
FUNCTIONAL DESCRIPTION (Continued)

Stack Pointer. The Z86E04/E08 has an 8-bit Stack Pointer (R255) used for the internal stack that resides within the 124 general-purpose registers.

General-Purpose Registers (GPR). These registers are undefined after the device is powered up. The registers keep their last value after any reset, as long as the reset occurs in the \( V_{cc} \) voltage-specified operating range. Note: Register R254 has been designated as a general-purpose register.

Counter/Timer. There are two 8-bit programmable counter/timers (T0 and T1), each driven by its own 6-bit programmable prescaler. The T1 prescaler is driven by internal or external clock sources; however, the T0 can be driven by the internal clock source only (Figure 13).

The 6-bit prescalers divide the input frequency of the clock source by any integer number from 1 to 64. Each prescaler drives its counter, which decrements the value (1 to 256) that has been loaded into the counter. When both counter and prescaler reach the end of count, a timer interrupt request IRQ4 (T0) or IRQ5 (T1) is generated.

The counter can be programmed to start, stop, restart to continue, or restart from the initial value. The counters are also programmed to stop upon reaching zero (single pass mode) or to automatically reload the initial value and continue counting (modulo-n continuous mode).

The counters, but not the prescalers, are read at any time without disturbing their value or count mode. The clock source for T1 is user-definable and is either the internal microprocessor clock divided by four, or an external signal input through Port 3. The Timer Mode register configures the external timer input (P30) as an external clock, a trigger input that is retrigergerable or non-retrigergerable, or used as a gate input for the internal clock.

![Counter/Timers Block Diagram](image-url)
Interrupts. The Z86E04/E08 has six interrupts from five different sources. These interrupts are maskable and prioritized (Figure 14). The sources are divided as follows: the rising edge of P31 (AN1), P32 (AN2), P33 (REF), the rising edge of P32 (AN2), and two counter/timers. The Interrupt Mask Register globally or individually enables or disables the six interrupt requests (Table 4).

When more than one interrupt is pending, priorities are resolved by a programmable priority encoder that is controlled by the Interrupt Priority register. All Z86E04/E08 interrupts are vectored through locations in program memory. When an Interrupt machine cycle is activated, an Interrupt Request is granted. This disables all subsequent interrupts, saves the Program Counter and Status Flags, and then branches to the program memory vector location reserved for that interrupt. This memory location and the next byte contain the 16-bit starting address of the interrupt service routine for that particular interrupt request.

To accommodate polled interrupt systems, interrupt inputs are masked and the interrupt request register is polled to determine which of the interrupt requests needs service.

Note: User must select any Z86E08 mode in Zilog's C12 ICEBOX™ emulator. The rising edge interrupt is not supported on the Z86CCP002EM emulator.

Table 4. Interrupt Types, Sources, and Vectors

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Vector Location</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRQ0</td>
<td>AN2(P32)</td>
<td>0, 1</td>
<td>External (F)Edge</td>
</tr>
<tr>
<td>IRQ1</td>
<td>REF(P33)</td>
<td>2, 3</td>
<td>External (F)Edge</td>
</tr>
<tr>
<td>IRQ2</td>
<td>AN1(P31)</td>
<td>4, 5</td>
<td>External (F)Edge</td>
</tr>
<tr>
<td>IRQ3</td>
<td>AN2(P32)</td>
<td>6, 7</td>
<td>Internal</td>
</tr>
<tr>
<td>IRQ4</td>
<td>T0</td>
<td>8, 9</td>
<td>Internal</td>
</tr>
<tr>
<td>IRQ5</td>
<td>T1</td>
<td>10, 11</td>
<td>Internal</td>
</tr>
</tbody>
</table>

Notes:
F = Falling edge triggered
R = Rising edge triggered

Figure 14. Interrupt Block Diagram
**FUNCTIONAL DESCRIPTION (Continued)**

**Clock.** The Z86E04/E08 on-chip oscillator has a high-gain, parallel-resonant amplifier for connection to a crystal, ceramic resonator, or any suitable external clock source (XTAL1 = INPUT, XTAL2 = OUTPUT). The crystal should be AT cut, 8 MHz or 12 MHz max, with a series resistance (RS) of less than or equal to 100 Ohms.

The crystal should be connected across XTAL1 and XTAL2 using the vendor's crystal recommended capacitors from each pin directly to device ground pin 14 (Figure 15). Note that the crystal capacitor loads should be connected to Vss, Pin 14 to reduce ground noise injection.

**Figure 15. Oscillator Configuration**

**HALT Mode.** This instruction turns off the internal CPU clock but not the crystal oscillation. The counter/timers and external interrupts IRQ0, IRQ1, IRQ2 and IRQ3 remain active. The device is recovered by interrupts, either externally or internally generated. An interrupt request must be executed (enabled) to exit HALT mode. After the interrupt service routine, the program continues from the instruction after the HALT.

**STOP Mode.** This instruction turns off the internal clock and external crystal oscillation and reduces the standby current to 10 µA. The STOP mode is released by a RESET through a Stop-Mode Recovery (pin P27). A Low input condition on P27 releases the STOP mode. Program execution begins at location 000C(Hex). However, when P27 is used to release the STOP mode, the I/O port mode registers are not reconfigured to their default power-on conditions. This prevents any I/O, configured as output when the STOP instruction was executed, from glitching to an unknown state. To use the P27 release approach with STOP mode, use the following instruction:

```
LD P2M, #1XXX XXXXB
NOP
STOP
```

X = Dependent on user's application.

In order to enter STOP or HALT mode, it is necessary to first flush the instruction pipeline to avoid suspending execution in mid-instruction. To do this, the user executes a NOP (opcode=FFH) immediately before the appropriate SLEEP instruction, i.e.:

```
FF NOP ; clear the pipeline
6F STOP ; enter STOP mode
or
FF NOP ; clear the pipeline
7F HALT ; enter HALT mode
```

**Watch-Dog Timer (WDT).** The Watch-Dog Timer is enabled by instruction WDT. When the WDT is enabled, it cannot be stopped by the instruction. With the WDT instruction, the WDT is refreshed when it is enabled within every 1 Twdt period; otherwise, the Z86E04 resets itself. The WDT instruction affects the flags accordingly; Z=1, S=0, V=0.

```
WDT = 5F (Hex)
```

**Opcode WDT (5FH).** The first time opcode 5FH is executed, the WDT is enabled and subsequent execution clears the WDT counter. This must be done at least every T_wdt; otherwise, the WDT times out and generates a reset. The generated reset is the same as a power-on reset of T_POR, plus 18 XTAL clock cycles.
Opcode **WDH** (4FH). When this instruction is executed it enables the WDT during HALT. If not, the WDT stops when entering HALT. This instruction does not clear the counters, it just makes it possible to have the WDT running during HALT mode. A WDH instruction executed without executing WDT (5FH) has no effect.

**Auto Reset Voltage** ($V_{\text{RST}}$). The Z86E04/E08 has an auto-reset built-in. The auto-reset circuit resets the Z86E04/E08 when it detects the $V_{\text{CC}}$ below $V_{\text{RST}}$. Figure 16 shows the Auto Reset Voltage vs temperature. The Z86E04/E08 does not function from $V_{\text{RST}}$ to below 4.5V. Upon power-up of the device, the $V_{\text{CC}}$ rise time must reach 4.5V before the $T_{\text{POR}}$ expires so that program execution begins with the $V_{\text{CC}}$ in the range 4.5V to 5.5V.

If the $V_{\text{CC}}$ drops below 4.5V while the device is in operation, the device must be powered down and then re-powered up again.

![Figure 16. Typical Auto Reset Voltage ($V_{\text{RST}}$) vs Temperature](image-url)
FUNCTIONAL DESCRIPTION (Continued)

**Low EMI Emission**

The Z86E04/E08 can be programmed to operate in a low EMI emission mode by means of an EPROM programmable bit option. Use of this feature results in:

- Less than 1 mA consumed during HALT mode.
- All drivers slew rates reduced to 10 ns (typical).
- Internal SCLK/TCLK = XTAL operation limited to a maximum of 4 MHz - 250 ns cycle time.
- Output drivers have resistances of 200 ohms (typical).
- Oscillator divide-by-two circuitry eliminated.

The Z86E04/E08 offers programmable ROM Protect and programmable Low Noise features. When programmed for Low Noise, the ROM Protect feature is optional.

In addition to $V_{DD}$ and GND ($V_{SS}$), the Z86E04/E08 changes all its pin functions in the EPROM mode. XTAL2 has no function, XTAL1 functions as /CE, P31 functions as /OE, P32 functions as EPM, P33 functions as $V_{PP}$, and P02 functions as /PGM.

**ROM Protect.** ROM Protect fully protects the Z86E04/E08 ROM code from being read externally. When ROM Protect is selected, the Z86E04/E08 will disable the instructions LDC and LDCI (Z86E04/E08 and Z86C04/C08 do not support the instructions of LDE and LDEI). When the device is programmed for ROM Protect, the Low Noise feature will automatically be enabled.

Please note that when using the device in a noisy environment, it is suggested that the voltages on the EPM and CE pins be clamped to $V_{cc}$ through a diode to $V_{cc}$ to prevent accidentally entering the OTP mode. The $V_{PP}$ requires both a diode and a 100 pF capacitor.

**User Modes.** Table 5 shows the programming voltage of each mode of Z86E04/E08.

### Table 5. OTP Programming Table

<table>
<thead>
<tr>
<th>Programming Modes</th>
<th>Device</th>
<th>$V_{PP}$</th>
<th>EPM</th>
<th>/CE</th>
<th>/OE</th>
<th>/PGM</th>
<th>ADDR</th>
<th>DATA</th>
<th>$V_{CC}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPROM READ1</td>
<td>All</td>
<td>X</td>
<td>$V_H$</td>
<td>$V_L$</td>
<td>$V_L$</td>
<td>$V_{IH}$</td>
<td>ADDR</td>
<td>Out</td>
<td>4.5V</td>
</tr>
<tr>
<td>EPROM READ2</td>
<td>All</td>
<td>X</td>
<td>$V_H$</td>
<td>$V_L$</td>
<td>$V_L$</td>
<td>$V_{IH}$</td>
<td>ADDR</td>
<td>Out</td>
<td>5.5V</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>All</td>
<td>$V_H$</td>
<td>X</td>
<td>$V_L$</td>
<td>$V_{IH}$</td>
<td>$V_L$</td>
<td>ADDR</td>
<td>In</td>
<td>6.0V</td>
</tr>
<tr>
<td>PROGRAM VERIFY</td>
<td>All</td>
<td>$V_H$</td>
<td>X</td>
<td>$V_L$</td>
<td>$V_{IH}$</td>
<td>$V_{IL}$</td>
<td>ADDR</td>
<td>Out</td>
<td>6.0V</td>
</tr>
<tr>
<td>EPROM PROTECT</td>
<td>All</td>
<td>$V_H$</td>
<td>$V_H$</td>
<td>$V_{IH}$</td>
<td>$V_{IL}$</td>
<td>NU</td>
<td>NU</td>
<td>6.0V</td>
<td></td>
</tr>
<tr>
<td>LOW NOISE SELECT</td>
<td>E04/E08</td>
<td>$V_H$</td>
<td>$V_{IH}$</td>
<td>$V_{IH}$</td>
<td>$V_{IL}$</td>
<td>NU</td>
<td>NU</td>
<td>6.0V</td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**

- $V_H = 12.5V \pm 0.5V$
- $V_{PP}$ = As per specific Z8 DC specification.
- $V_{IL}$ = As per specific Z8 DC specification.
- X = Not used, but must be set to $V_H$, $V_{PP}$ or $V_{IL}$ level.
- NU = Not used, but must be set to either $V_H$ or $V_{IL}$ level.
- $I_{PP}$ during programming = 40 mA maximum.
- $I_{CC}$ during programming, verify, or read = 40 mA maximum.

$V_{CC}$ has a tolerance of ±0.25V.
**Internal Address Counter.** The address of Z86E04/E08 is generated internally with a counter clocked through pin P01 (Clock). Each clock signal increases the address by one and the “high” level of pin P00 (Clear) will reset the address to zero. Figure 17 shows the setup time of the serial address input.

**Programming Waveform.** Figures 18, 19 and 20 show the programming waveforms of each mode. Table 6 shows the timing of programming waveforms.

**Programming Algorithm.** Figure 21 shows the flow chart of the Z86E04/E08 programming algorithm.

### Table 6. Timing of Programming Waveforms

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Name</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Address Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>2</td>
<td>Data Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>3</td>
<td>V&lt;sub&gt;PP&lt;/sub&gt; Setup</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>4</td>
<td>V&lt;sub&gt;CC&lt;/sub&gt; Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>5</td>
<td>Chip Enable Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>6</td>
<td>Program Pulse Width</td>
<td>0.95</td>
<td></td>
<td>ms</td>
</tr>
<tr>
<td>7</td>
<td>Data Hold Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>8</td>
<td>/OE Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>9</td>
<td>Data Access Time</td>
<td></td>
<td>200</td>
<td>ns</td>
</tr>
<tr>
<td>10</td>
<td>Data Output Float Time</td>
<td></td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td>11</td>
<td>Overprogram Pulse Width</td>
<td></td>
<td>2.85</td>
<td>ms</td>
</tr>
<tr>
<td>12</td>
<td>EPM Setup Time</td>
<td></td>
<td>2</td>
<td>µs</td>
</tr>
<tr>
<td>13</td>
<td>/PGM Setup Time</td>
<td></td>
<td>2</td>
<td>µs</td>
</tr>
<tr>
<td>14</td>
<td>Address to /OE Setup Time</td>
<td></td>
<td>2</td>
<td>µs</td>
</tr>
<tr>
<td>15</td>
<td>Option Program Pulse Width</td>
<td></td>
<td>78</td>
<td>ms</td>
</tr>
</tbody>
</table>
FUNCTIONAL DESCRIPTION (Continued)

Figure 17. Z86E04/E08 Address Counter Waveform
Figure 18. Z86E04/E08 Programming Waveform (EPROM Read)
Figure 19. Z86E04/E08 Programming Waveform (Program and Verify)
Figure 20. Z86E04/E08 Programming Waveform (EPROM Protect and Low EMI Program)
FUNCTIONAL DESCRIPTION (Continued)

Figure 21. Z86E04/E08 Programming Algorithm
ABSOLUTE MAXIMUM RATINGS

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for an extended period may affect device reliability. Total power dissipation should not exceed 462 mW for the package. Power dissipation is calculated as follows:

\[
\text{Total Power Dissipation} = V_{DD} \times \left[ \left( I_{DD} - \text{sum of } I_{OH} \right) \right] + \text{sum of } (V_{DD} - V_{OL} \times I_{OL}) + \text{sum of } (V_{OL} \times I_{OL})
\]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient Temperature under Bias</td>
<td>-40</td>
<td>+105</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>Storage Temperature</td>
<td>-65</td>
<td>+150</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>Voltage on any Pin with Respect to V\text{ss}</td>
<td>-0.6</td>
<td>+12</td>
<td>V</td>
<td>[1]</td>
</tr>
<tr>
<td>Voltage on V\text{DD} Pin with Respect to V\text{ss}</td>
<td>-0.3</td>
<td>+7</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Voltage on Pins 7, 8, 9, 10 with Respect to V\text{ss}</td>
<td>-0.6</td>
<td>V_{DD}+1</td>
<td>V</td>
<td>[2]</td>
</tr>
<tr>
<td>Total Power Dissipation</td>
<td>462</td>
<td></td>
<td>mW</td>
<td></td>
</tr>
<tr>
<td>Maximum Current out of V\text{ss}</td>
<td>84</td>
<td></td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td>Maximum Current into V\text{DD}</td>
<td>84</td>
<td></td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td>Maximum Current into an Input Pin</td>
<td>-600</td>
<td>+600</td>
<td>μA</td>
<td>[3]</td>
</tr>
<tr>
<td>Maximum Current into an Open-Drain Pin</td>
<td>-600</td>
<td>+600</td>
<td>μA</td>
<td>[4]</td>
</tr>
<tr>
<td>Maximum Output Current Sensed by Any I/O Pin</td>
<td>12</td>
<td></td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td>Maximum Output Current Sensed by Port 2</td>
<td>70</td>
<td></td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td>Total Maximum Output Current Sensed by Port 2</td>
<td>70</td>
<td></td>
<td>mA</td>
<td></td>
</tr>
</tbody>
</table>

Notes:
[1] This applies to all pins except where otherwise noted.
Maximum current into pin must be ±600 μA.
[2] There is no input protection diode from pin to VDD
(not applicable to EPROM Mode).
[4] Device pin is not at an output Low state.

STANDARD TEST CONDITIONS

The characteristics listed below apply for standard test conditions as noted. All voltages are referenced to Ground. Positive current flows into the referenced pin (Figure 22).

![Figure 22. Test Load Diagram](image)

CAPACITANCE

\( T_A = 25^\circ\text{C}, V_{CC} = \text{GND} = 0\text{V}, f = 1.0 \text{ MHz}, \) unmeasured pins returned to GND.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input capacitance</td>
<td>0</td>
<td>10 pF</td>
</tr>
<tr>
<td>Output capacitance</td>
<td>0</td>
<td>20 pF</td>
</tr>
<tr>
<td>I/O capacitance</td>
<td>0</td>
<td>25 pF</td>
</tr>
</tbody>
</table>
## DC ELECTRICAL CHARACTERISTICS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{CC}$</th>
<th>$T_A = 0°C$ to $+70°C$</th>
<th>Typical @ 25°C</th>
<th>Units</th>
<th>Conditions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{CH}$</td>
<td>Clock Input High Voltage</td>
<td>4.5V</td>
<td>0.8 $V_{CC}$</td>
<td>$V_{CC}+0.3$</td>
<td>2.4</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.8 $V_{CC}$</td>
<td>$V_{CC}+0.3$</td>
<td>2.6</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
</tr>
<tr>
<td>$V_{CL}$</td>
<td>Clock Input Low Voltage</td>
<td>4.5V</td>
<td>$V_{SS}-0.3$</td>
<td>0.2 $V_{CC}$</td>
<td>1.6</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{SS}-0.3$</td>
<td>0.2 $V_{CC}$</td>
<td>2.3</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
</tr>
<tr>
<td>$V_{IH}$</td>
<td>Input High Voltage</td>
<td>4.5V</td>
<td>0.7 $V_{CC}$</td>
<td>$V_{CC}+0.3$</td>
<td>2.1</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.7 $V_{CC}$</td>
<td>$V_{CC}+0.3$</td>
<td>2.7</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>$V_{IL}$</td>
<td>Input Low Voltage</td>
<td>4.5V</td>
<td>$V_{SS}-0.3$</td>
<td>0.2 $V_{CC}$</td>
<td>1.2</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{SS}-0.3$</td>
<td>0.2 $V_{CC}$</td>
<td>1.7</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>$V_{OH}$</td>
<td>Output High Voltage</td>
<td>4.5V</td>
<td>$V_{CC}-0.4$</td>
<td>3.9</td>
<td>V</td>
<td>$I_{OH} = -2.0 mA$</td>
<td>[3]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{CC}-0.4$</td>
<td>5.4</td>
<td>V</td>
<td>$I_{OH} = -2.0 mA$</td>
<td>[3]</td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>Output Low Voltage</td>
<td>4.5V</td>
<td>0.4</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.4</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{OL1}$</td>
<td>Output Low Voltage</td>
<td>4.5V</td>
<td>0.8</td>
<td>0.2</td>
<td>V</td>
<td>$I_{OL} = +4.0 mA$</td>
<td>[3]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.4</td>
<td>0.2</td>
<td>V</td>
<td>$I_{OL} = +4.0 mA$</td>
<td>[3]</td>
</tr>
<tr>
<td>$V_{OL2}$</td>
<td>Output Low Voltage</td>
<td>4.5V</td>
<td>TBD</td>
<td>0.7</td>
<td>V</td>
<td>$I_{OL} = +12 mA$, 3 Pin Max</td>
<td>[3]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.8</td>
<td>0.5</td>
<td>V</td>
<td>$I_{OL} = +12 mA$, 3 Pin Max</td>
<td>[3]</td>
</tr>
<tr>
<td>$V_{OFFSET}$</td>
<td>Comparator Input Offset Voltage</td>
<td>4.5V</td>
<td>10</td>
<td>6</td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>7</td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{RST}$</td>
<td>Auto Reset Voltage</td>
<td>1.55</td>
<td>2.7</td>
<td>2.4</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{IL}$</td>
<td>Input Leakage (Input Bias Current of Comparator)</td>
<td>4.5V</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>$\mu A$</td>
<td>$V_{IN} = 0V, V_{CC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>$\mu A$</td>
<td>$V_{IN} = 0V, V_{CC}$</td>
</tr>
<tr>
<td>$I_{OL}$</td>
<td>Output Leakage</td>
<td>4.5V</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>$\mu A$</td>
<td>$V_{IN} = 0V, V_{CC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>$\mu A$</td>
<td>$V_{IN} = 0V, V_{CC}$</td>
</tr>
<tr>
<td>$V_{ICR}$</td>
<td>Input Common Mode Voltage Range</td>
<td>0</td>
<td>$V_{CC}-1.0$</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Symbol** | **Parameter** | **Symbol** | **Parameter** | **$V_{CC}$** | **$I_{CC}$** | **Symbol** | **Parameter** | **Symbol** | **Parameter** | **$V_{CC}$** | **$I_{CC}$** | **Symbol** | **Parameter** | **Symbol** | **Parameter** | **$V_{CC}$** | **$I_{CC}$** |
| $I_{cc}$ | Supply Current (Standard Mode) | | | 4.5V | 4.0 | 2.2 | mA | All Output and I/O Pins Floating @ 2 MHz |
| | | | | 5.5V | 7.0 | 5.0 | mA | All Output and I/O Pins Floating @ 2 MHz |
| | | | | 4.5V | 9.0 | 4.5 | mA | All Output and I/O Pins Floating @ 8 MHz |
| | | | | 5.5V | 11.0 | 8.3 | mA | All Output and I/O Pins Floating @ 8 MHz |
| | | | | 4.5V | 10 | 6.1 | mA | All Output and I/O Pins Floating @ 12 MHz (E08) |
| | | | | 5.5V | 15 | 10.8 | mA | All Output and I/O Pins Floating @ 12 MHz (E08) |
| $I_{cc1}$ | Standby Current (Standard Mode) | | | 4.5V | 2.5 | 0.5 | mA | HALT mode $V_{cc} = 0V$, $V_{CC} @ 2 MHz$ |
| | | | | 5.5V | 4.0 | 1.0 | mA | HALT mode $V_{cc} = 0V$, $V_{CC} @ 2 MHz$ |
| | | | | 4.5V | 4.0 | 1.0 | mA | HALT mode $V_{cc} = 0V$, $V_{CC} @ 8 MHz$ |
| | | | | 5.5V | 5.0 | 2.0 | mA | HALT mode $V_{cc} = 0V$, $V_{CC} @ 8 MHz$ |
| | | | | 4.5V | 5.0 | 1.3 | mA | HALT mode $V_{cc} = 0V$, $V_{CC} @ 12 MHz (E08)$ |
| | | | | 5.5V | 7.0 | 2.3 | mA | HALT mode $V_{cc} = 0V$, $V_{CC} @ 12 MHz (E08)$ |
| $I_{cc}$ | Supply Current (Low Noise Mode) | | | 4.5V | 4.0 | 2.2 | mA | All Output and I/O Pins Floating @ 1 MHz |
| | | | | 5.5V | 7.0 | 4.2 | mA | All Output and I/O Pins Floating @ 1 MHz |
| | | | | 4.5V | 6.0 | 2.9 | mA | All Output and I/O Pins Floating @ 2 MHz |
| | | | | 5.5V | 9.0 | 5.5 | mA | All Output and I/O Pins Floating @ 2 MHz |
| | | | | 4.5V | 8.0 | 4.4 | mA | All Output and I/O Pins Floating @ 4 MHz |
| | | | | 5.5V | 11.0 | 7.9 | mA | All Output and I/O Pins Floating @ 4 MHz |
### DC ELECTRICAL CHARACTERISTICS (Continued)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Voltage</th>
<th>@ 25°C</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>I(_{CC1})</td>
<td>Standby Current (Low Noise Mode)</td>
<td>4.5V</td>
<td>1.2</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>1.6</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.5V</td>
<td>1.5</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>1.9</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.5V</td>
<td>2.0</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>2.4</td>
<td>1.3</td>
</tr>
<tr>
<td>I(_{CC2})</td>
<td>Standby Current</td>
<td>4.5V</td>
<td>10</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>10</td>
<td>1.0</td>
</tr>
<tr>
<td>I(_{AL})</td>
<td>Auto Latch Low Current</td>
<td>4.5V</td>
<td>10</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>15</td>
<td>11.5</td>
</tr>
<tr>
<td>I(_{AH})</td>
<td>Auto Latch High Current</td>
<td>4.5V</td>
<td>-7.0</td>
<td>-3.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>-7.0</td>
<td>-6.5</td>
</tr>
</tbody>
</table>

**Notes:**

1. \(I_{CC1}\): Typ Mix Unit
   - Clock Driven: 0.3 5.0 mA 8 MHz
   - Crystal or Resonator: 3.5 5.0 mA 8 MHz
2. \(V_{CC} = 0V = GND\)
AC ELECTRICAL CHARACTERISTICS

Figure 23. Electrical Timing Diagram
## AC ELECTRICAL CHARACTERISTICS

### Low Noise Mode

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>V&lt;sub&gt;CC&lt;/sub&gt;</th>
<th>1 MHz (Min)</th>
<th>1 MHz (Max)</th>
<th>4 MHz (Min)</th>
<th>4 MHz (Max)</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TpC</td>
<td>Input Clock Period</td>
<td>4.5V</td>
<td>1000 DC</td>
<td>250 DC</td>
<td>250 DC</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>1000 DC</td>
<td>250 DC</td>
<td>250 DC</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TrC, TfC</td>
<td>Clock Input Rise and Fall Times</td>
<td>4.5V</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TwC</td>
<td>Input Clock Width</td>
<td>4.5V</td>
<td>500</td>
<td>125</td>
<td>125</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>500</td>
<td>125</td>
<td>125</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TwTinL</td>
<td>Timer Input Low Width</td>
<td>4.5V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>TwTinh</td>
<td>Timer Input High Width</td>
<td>4.5V</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>TpTin</td>
<td>Timer Input Period</td>
<td>4.5V</td>
<td>4TpC</td>
<td>4TpC</td>
<td>4TpC</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>4TpC</td>
<td>4TpC</td>
<td>4TpC</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TrTin, TtTin</td>
<td>Timer Input Rise and Fall Timer</td>
<td>4.5V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>TwI L</td>
<td>Int. Request Input Low Time</td>
<td>4.5V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1,2]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>ns</td>
<td>[1,2]</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>TwI H</td>
<td>Int. Request Input High Time</td>
<td>4.5V</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>ns</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>ns</td>
<td>[1,2]</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Twdt</td>
<td>Watch-Dog Timer Delay Time</td>
<td>4.5V</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>ms</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>ms</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>TPOR</td>
<td>Power-On Reset Time</td>
<td>4.5V</td>
<td>15</td>
<td>10</td>
<td>10</td>
<td>ms</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>15</td>
<td>10</td>
<td>10</td>
<td>ms</td>
<td>[1]</td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**

[1] Timing Reference uses 0.9 V<sub>CC</sub> for a logic 1 and 0.1 V<sub>CC</sub> for a logic 0.

[2] Interrupt request through Port 3 (P33-P31)
### AC ELECTRICAL CHARACTERISTICS
Standard Mode, Standard Temperature

<table>
<thead>
<tr>
<th>No.</th>
<th>Symbol</th>
<th>Parameter Description</th>
<th>$V_{cc}$</th>
<th>8 MHz (E04) Min</th>
<th>8 MHz (E04) Max</th>
<th>12 MHz (E08) Min</th>
<th>12 MHz (E08) Max</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TpC</td>
<td>Input Clock Period</td>
<td></td>
<td>4.5V 125 DC</td>
<td>4.5V 125 DC</td>
<td>5.5V 83 DC</td>
<td>5.5V 83 DC</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>3</td>
<td>TwC</td>
<td>Input Clock Width</td>
<td></td>
<td>4.5V 62 ns</td>
<td>4.5V 62 ns</td>
<td>5.5V 41 ns</td>
<td>5.5V 41 ns</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>4</td>
<td>TwTinL</td>
<td>Timer Input Low Width</td>
<td></td>
<td>4.5V 100 ns</td>
<td>4.5V 100 ns</td>
<td>5.5V 70 ns</td>
<td>5.5V 70 ns</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>5</td>
<td>TwTinH</td>
<td>Timer Input High Width</td>
<td></td>
<td>4.5V 5Tpc</td>
<td>4.5V 5Tpc</td>
<td>5.5V 5Tpc</td>
<td>5.5V 5Tpc</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>6</td>
<td>TpTin</td>
<td>Timer Input Period</td>
<td></td>
<td>4.5V 8Tpc</td>
<td>4.5V 8Tpc</td>
<td>5.5V 8Tpc</td>
<td>5.5V 8Tpc</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>7</td>
<td>TrTin, TttIn</td>
<td>Timer Input Rise and Fall Timer</td>
<td></td>
<td>4.5V 100 ns</td>
<td>4.5V 100 ns</td>
<td>5.5V 70 ns</td>
<td>5.5V 70 ns</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>8</td>
<td>TwL</td>
<td>Int. Request Input Low Time</td>
<td></td>
<td>4.5V 100 ns</td>
<td>4.5V 100 ns</td>
<td>5.5V 70 ns</td>
<td>5.5V 70 ns</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td>9</td>
<td>TwH</td>
<td>Int. Request Input High Time</td>
<td></td>
<td>4.5V 5Tpc</td>
<td>4.5V 5Tpc</td>
<td>5.5V 5Tpc</td>
<td>5.5V 5Tpc</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td>10</td>
<td>Twdt</td>
<td>Watch-Dog Timer Delay Time</td>
<td></td>
<td>4.5V 15 ms</td>
<td>4.5V 15 ms</td>
<td>5.5V 10 ms</td>
<td>5.5V 10 ms</td>
<td>ms</td>
<td>[1]</td>
</tr>
<tr>
<td>11</td>
<td>TPOR</td>
<td>Power-On Reset Timer</td>
<td></td>
<td>4.5V 60 ms</td>
<td>4.5V 60 ms</td>
<td>5.5V 45 ms</td>
<td>5.5V 45 ms</td>
<td>ms</td>
<td>[1]</td>
</tr>
</tbody>
</table>

**Notes:**

[1] Timing Reference uses 0.7 $V_{cc}$ for a logic 1 and 0.2 $V_{cc}$ for a logic 0.

[2] Interrupt request through Port 3 (P33-P31)
Z8 CONTROL REGISTERS

**Figure 24. Timer Mode Register**  
(F1H: Read/Write)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>T1 Initial Value (When Written)</td>
</tr>
<tr>
<td></td>
<td>(Range 1-256 Decimal 01-00 Hex)</td>
</tr>
<tr>
<td>6</td>
<td>T1 Current Value (When READ)</td>
</tr>
<tr>
<td>5</td>
<td>Reserved (Must be 0)</td>
</tr>
<tr>
<td>4</td>
<td>T1 Modes</td>
</tr>
<tr>
<td>3</td>
<td>External Clock Input</td>
</tr>
<tr>
<td>2</td>
<td>Gate Input</td>
</tr>
<tr>
<td>1</td>
<td>Trigger Input (Non-retriggerable)</td>
</tr>
<tr>
<td>0</td>
<td>Trigger Input (Retriggerable)</td>
</tr>
</tbody>
</table>

**Figure 25. Counter Timer 1 Register**  
(F2H: Read/Write)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>T1 Initial Value (When Written)</td>
</tr>
<tr>
<td></td>
<td>(Range 1-256 Decimal 01-00 Hex)</td>
</tr>
<tr>
<td>6</td>
<td>T1 Current Value (When READ)</td>
</tr>
</tbody>
</table>

**Figure 26. Prescaler 1 Register**  
(F3H: Write Only)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Prescaler Modulo (Range: 1-64 Decimal 01-00 Hex)</td>
</tr>
<tr>
<td></td>
<td>Reserved (Must be 0.)</td>
</tr>
<tr>
<td>6</td>
<td>Count Mode</td>
</tr>
<tr>
<td>5</td>
<td>0 = T0 Single Pass</td>
</tr>
<tr>
<td>4</td>
<td>1 = T0 Modulo-N</td>
</tr>
</tbody>
</table>

**Figure 27. Counter/Timer 0 Register**  
(F4H: Read/Write)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>T0 Initial Value (When Written)</td>
</tr>
<tr>
<td></td>
<td>(Range: 1-256 Decimal 01-00 Hex)</td>
</tr>
<tr>
<td>6</td>
<td>T0 Current Value (When READ)</td>
</tr>
</tbody>
</table>

**Figure 28. Prescaler 0 Register**  
(F5H: Write Only)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>P27 - P20 I/O Definition</td>
</tr>
<tr>
<td></td>
<td>0 Defines Bit as OUTPUT</td>
</tr>
<tr>
<td></td>
<td>1 Defines Bit as INPUT</td>
</tr>
</tbody>
</table>

**Figure 29. Port 2 Mode Register**  
(F6H: Write Only)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>0 Port 2 Pull-Ups Open-Drain</td>
</tr>
<tr>
<td>6</td>
<td>1 Port 2 Pull-Ups Active</td>
</tr>
<tr>
<td>5</td>
<td>0 Port 3 Pull-Ups Active</td>
</tr>
<tr>
<td>4</td>
<td>1 Digital</td>
</tr>
<tr>
<td>3</td>
<td>1 Analog</td>
</tr>
<tr>
<td>2</td>
<td>Reserved (Must be 0.)</td>
</tr>
</tbody>
</table>

**Figure 30. Port 3 Mode Register**  
(F7H: Write Only)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Reserved (Must be 0.)</td>
</tr>
</tbody>
</table>
Figure 31. Port 0 and 1 Mode Register (F8H; Write Only)

P03-P00 Mode
00 = Output
01 = Input
Reserved (Must be 1.)
Reserved (Must be 0.)

Figure 32. Interrupt Priority Register (F9H; Write Only)

Interrupt Group Priority
Reserved = 000
C > A > B = 001
A > B > C = 010
A > C > B = 011
B > C > A = 100
B > A > C = 101
Reserved = 111
IRQ1, IRQ4 Priority (Group C)
0 = IRQ1 > IRQ4
1 = IRQ4 > IRQ1
IRQ0, IRQ2 Priority (Group B)
0 = IRQ2 > IRQ0
1 = IRQ0 > IRQ2
IRQ3, IRQ5 Priority (Group A)
0 = IRQ5 > IRQ3
1 = IRQ3 > IRQ5
Reserved (Must be 0.)

Figure 33. Interrupt Request Register (FAH; Read/Write)

IRQ0 = P32 Input ↓
IRQ1 = P33 Input ↓
IRQ2 = P31 Input ↓
IRQ3 = P32 Input ↑
IRQ4 = T0
IRQ5 = T1
Reserved (Must be 0.)

Figure 34. Interrupt Mask Register (FBH; Read/Write)

1 Enables IRQ0-IRQ5
(D0 = IRQ0)
Reserved (Must be 0.)
1 Enables Interrupts

Figure 35. Flag Register (FC; Read/Write)

User Flag F1
User Flag F2
Half Carry Flag
Decimal Adjust Flag
Overflow Flag
Sign Flag
Zero Flag
Cary Flag

Figure 36. Register Pointer (FDH; Read/Write)

Reserved (Must be 0)
Register Pointer

Figure 37. Stack Pointer (FFH; Read/Write)

Stack Pointer Lower Byte (SP7 - SP0)
OPERATING MODES

Figure 38. Maximum \( I_{CC} \) and \( I_{CC1} \) vs Frequency in Standard Mode
Figure 39. Typical $I_{cc}$ and $I_{cc1}$ vs Frequency in Standard Mode
Figure 40. Typical \( I_{cc} \) and \( I_{cc1} \) vs Frequency in Low EMI Mode

Figure 41. Maximum \( I_{cc} \) and \( I_{cc1} \) vs Frequency in Low EMI Mode
Figure 42. Typical POR Time Out Period vs Temperature

Figure 43. Typical WDT Time Out Period vs Temperature
### INSTRUCTION SET NOTATION

**Addressing Modes.** The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRR</td>
<td>Indirect register pair or indirect working-register pair address</td>
</tr>
<tr>
<td>Ir</td>
<td>Indirect working-register pair only</td>
</tr>
<tr>
<td>X</td>
<td>Indexed address</td>
</tr>
<tr>
<td>DA</td>
<td>Direct address</td>
</tr>
<tr>
<td>RA</td>
<td>Relative address</td>
</tr>
<tr>
<td>IM</td>
<td>Immediate</td>
</tr>
<tr>
<td>R</td>
<td>Register or working-register address</td>
</tr>
<tr>
<td>r</td>
<td>Working register address only</td>
</tr>
<tr>
<td>IR</td>
<td>Indirect-register or indirect working-register address</td>
</tr>
<tr>
<td>Ir</td>
<td>Indirect working-register address only</td>
</tr>
<tr>
<td>RR</td>
<td>Register pair or working register pair address</td>
</tr>
</tbody>
</table>

**Symbols.** The following symbols are used in describing the instruction set.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst</td>
<td>Destination location or contents</td>
</tr>
<tr>
<td>src</td>
<td>Source location or contents</td>
</tr>
<tr>
<td>cc</td>
<td>Condition code</td>
</tr>
<tr>
<td>@</td>
<td>Indirect address prefix</td>
</tr>
<tr>
<td>SP</td>
<td>Stack pointer</td>
</tr>
<tr>
<td>PC</td>
<td>Program counter</td>
</tr>
<tr>
<td>FLAGS</td>
<td>Flag register (Control Register 252)</td>
</tr>
<tr>
<td>RP</td>
<td>Register Pointer (R253)</td>
</tr>
<tr>
<td>IMR</td>
<td>Interrupt mask register (R251)</td>
</tr>
</tbody>
</table>

**Flags.** Control register (R252) contains the following six flags.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Carry flag</td>
</tr>
<tr>
<td>Z</td>
<td>Zero flag</td>
</tr>
<tr>
<td>S</td>
<td>Sign flag</td>
</tr>
<tr>
<td>V</td>
<td>Overflow flag</td>
</tr>
<tr>
<td>D</td>
<td>Decimal-adjust flag</td>
</tr>
<tr>
<td>H</td>
<td>Half-carry flag</td>
</tr>
</tbody>
</table>

Affected flags are indicated by:

0: Clear to zero
1: Set to one
*: Set to clear according to operation
-: Unaffected
X: Undefined
## CONDITION CODES

<table>
<thead>
<tr>
<th>Value</th>
<th>Mnemonic</th>
<th>Meaning</th>
<th>Flags Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>—</td>
<td>Always true</td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>C</td>
<td>Carry</td>
<td>C=1</td>
</tr>
<tr>
<td>1111</td>
<td>NC</td>
<td>No Carry</td>
<td>C=0</td>
</tr>
<tr>
<td>0110</td>
<td>Z</td>
<td>Zero</td>
<td>Z=1</td>
</tr>
<tr>
<td>1110</td>
<td>NZ</td>
<td>Not zero</td>
<td>Z=0</td>
</tr>
<tr>
<td>1101</td>
<td>PL</td>
<td>Plus</td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td>MI</td>
<td>Minus</td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>OV</td>
<td>Overflow</td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td>NOV</td>
<td>No overflow</td>
<td></td>
</tr>
<tr>
<td>0110</td>
<td>EQ</td>
<td>Equal</td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>NE</td>
<td>Not equal</td>
<td>Z=0</td>
</tr>
<tr>
<td>1001</td>
<td>GE</td>
<td>Greater than or equal</td>
<td>(S XOR V)=0</td>
</tr>
<tr>
<td>0001</td>
<td>LT</td>
<td>Less than</td>
<td>(S XOR V)=1</td>
</tr>
<tr>
<td>1010</td>
<td>GT</td>
<td>Greater than</td>
<td>[Z OR (S XOR V)]=0</td>
</tr>
<tr>
<td>0010</td>
<td>LE</td>
<td>Less than or equal</td>
<td>[Z OR (S XOR V)]=1</td>
</tr>
<tr>
<td>1111</td>
<td>UGE</td>
<td>Unsigned greater than or equal</td>
<td>C=0</td>
</tr>
<tr>
<td>0111</td>
<td>ULT</td>
<td>Unsigned less than</td>
<td>C=1</td>
</tr>
<tr>
<td>1011</td>
<td>UGT</td>
<td>Unsigned greater than</td>
<td>(C = 0 AND Z=0)=1</td>
</tr>
<tr>
<td>0011</td>
<td>ULE</td>
<td>Unsigned less than or equal</td>
<td>(C OR Z)=1</td>
</tr>
<tr>
<td>0000</td>
<td>F</td>
<td>Never true (Always False)</td>
<td></td>
</tr>
</tbody>
</table>
### INSTRUCTION FORMATS

#### One-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>src</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>1110</td>
<td>dst</td>
<td>src</td>
</tr>
</tbody>
</table>

#### OPC MODE

- **CLF, CPL, DA, DEC, DECW, INC, INCW, POP, PUSH, RL, RLC, RR, RRC, SRA, SWAP**

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>src</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>1110</td>
<td>dst</td>
<td>src</td>
</tr>
</tbody>
</table>

####Callable Instructions

- **JP, CALL (Indirect)**

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>src</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>1110</td>
<td>dst</td>
<td>src</td>
</tr>
</tbody>
</table>

#### SRP

- **LD, LDE, LDEI, LDC, LCDI**

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>src</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>1110</td>
<td>dst</td>
<td>src</td>
</tr>
</tbody>
</table>

#### VALUE

- **DJNZ, JR**

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>src</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>1110</td>
<td>dst</td>
<td>src</td>
</tr>
</tbody>
</table>

#### STOP/HALT

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>src</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>FFH</td>
<td>7FH</td>
<td>7FH</td>
</tr>
</tbody>
</table>

### Two-Byte Instructions

#### OPC MODE

- **ADC, ADD, AND, CP, LD, OR, SBC, SUB, TCM, TM, XOR**

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>src</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>1110</td>
<td>dst</td>
<td>src</td>
</tr>
</tbody>
</table>

#### VALUE

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>src</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>1110</td>
<td>dst</td>
<td>src</td>
</tr>
</tbody>
</table>

#### ADDRESS

- **LD**

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>src</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>1110</td>
<td>dst</td>
<td>src</td>
</tr>
</tbody>
</table>

### Three-Byte Instructions

#### OPC MODE

- **CCF, DI, EI, IRET, NOP, RCF, RET, SCF**

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>src</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>1110</td>
<td>dst</td>
<td>src</td>
</tr>
</tbody>
</table>

#### VALUE

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>src</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>1110</td>
<td>dst</td>
<td>src</td>
</tr>
</tbody>
</table>

### INSTRUCTION SUMMARY

**Note:** Assignment of a value is indicated by the symbol *"←"*. For example:

\[
\text{dst} \leftarrow \text{dst} + \text{src}
\]

indicates that the source data is added to the destination data and the result is stored in the destination location. The notation "addr (n)" is used to refer to bit (n) of a given operand location. For example:

\[
\text{dst(7)}
\]

refers to bit 7 of the destination operand.
# INSTRUCTION SUMMARY

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ADC</strong> dst, src</td>
<td><strong>t</strong></td>
<td>1[ ]</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>dst←dst + src +C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ADD</strong> dst, src</td>
<td><strong>t</strong></td>
<td>0[ ]</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>dst←dst + src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AND</strong> dst, src</td>
<td><strong>t</strong></td>
<td>5[ ]</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>dst←dst AND src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CALL</strong> dst</td>
<td>DA</td>
<td>D6</td>
<td></td>
</tr>
<tr>
<td>SP←SP – 2</td>
<td>IRR</td>
<td>D4</td>
<td></td>
</tr>
<tr>
<td>@SP←PC, PC←dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CCF</strong></td>
<td>EF</td>
<td></td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>C←NOT C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CLR</strong> dst</td>
<td>R</td>
<td>B0</td>
<td></td>
</tr>
<tr>
<td>dst←0</td>
<td>IR</td>
<td>B1</td>
<td></td>
</tr>
<tr>
<td><strong>COM</strong> dst</td>
<td>R</td>
<td>60</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>dst←NOT dst</td>
<td>IR</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td><strong>CP</strong> dst, src</td>
<td><strong>t</strong></td>
<td>A[ ]</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>dst ← src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DA</strong> dst</td>
<td>R</td>
<td>40</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>dst←DA dst</td>
<td>IR</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td><strong>DEC</strong> dst</td>
<td>R</td>
<td>00</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>dst←dst – 1</td>
<td>IR</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td><strong>DECW</strong> dst</td>
<td>RR</td>
<td>80</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>dst←dst – 1</td>
<td>IR</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td><strong>DI</strong></td>
<td>8F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMR(7)←0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DJNZr, dst</strong></td>
<td>RA</td>
<td>rA</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>r←r – 1</td>
<td>r = 0 – F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>if r ≠ 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←PC + dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range: +127, –128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>EI</strong></td>
<td>9F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMR(7)←1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HALT</strong></td>
<td>7F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INC</strong> dst</td>
<td>r</td>
<td>rE</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>dst←dst + 1</td>
<td>r = 0 – F</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>INCW</strong> dst</td>
<td>RR</td>
<td>A0</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>dst←dst + 1</td>
<td>IR</td>
<td>A1</td>
<td></td>
</tr>
<tr>
<td><strong>IRET</strong></td>
<td>BF</td>
<td><strong>C Z S V D H</strong></td>
<td></td>
</tr>
<tr>
<td>FLAGS←@SP,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP←SP + 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←@SP,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP←SP + 2; IMR(7)←1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>** JP cc, dst**</td>
<td>DA</td>
<td>cD</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>if cc is true, PC←dst</td>
<td>IRR</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td><strong>JR cc, dst</strong></td>
<td>RA</td>
<td>cB</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>if cc is true, PC←PC + dst</td>
<td>IRR</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Range: +127, –128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LD</strong> dst, src</td>
<td>r</td>
<td>lm</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>dst←src</td>
<td>r</td>
<td>r8</td>
<td></td>
</tr>
<tr>
<td><strong>LDI</strong> dst, src</td>
<td>r</td>
<td>r9</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>dst←src</td>
<td>r</td>
<td>r9</td>
<td></td>
</tr>
<tr>
<td>Range: +127, –128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LDC</strong> dst, src</td>
<td>r</td>
<td>lrr</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>dst←src</td>
<td>lrr</td>
<td>C3</td>
<td></td>
</tr>
<tr>
<td><strong>LDCI</strong> dst, src</td>
<td>lrr</td>
<td>C3</td>
<td><strong>C Z S V D H</strong></td>
</tr>
<tr>
<td>dst←src</td>
<td>lrr</td>
<td>C3</td>
<td></td>
</tr>
</tbody>
</table>

---

4-39
### INSTRUCTION SUMMARY (Continued)

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOP</td>
<td>dst, src</td>
<td>FF</td>
<td></td>
<td>STOP</td>
<td>dst</td>
<td>6F</td>
<td>C</td>
</tr>
<tr>
<td>OR</td>
<td>dst, src</td>
<td>OR dst OR src</td>
<td>4[ ]</td>
<td>0</td>
<td>STOP</td>
<td>dst</td>
<td>6F</td>
</tr>
<tr>
<td>POP</td>
<td>dst</td>
<td>POP dst</td>
<td>50</td>
<td></td>
<td>STOP</td>
<td>dst</td>
<td>6F</td>
</tr>
<tr>
<td>R</td>
<td>src</td>
<td>R</td>
<td>70</td>
<td>SWAP</td>
<td>dst</td>
<td>F0</td>
<td>X</td>
</tr>
<tr>
<td>R</td>
<td>SP</td>
<td>SP</td>
<td>71</td>
<td>TCM</td>
<td>dst</td>
<td>6[ ]</td>
<td>0</td>
</tr>
<tr>
<td>R</td>
<td>SP</td>
<td>SP</td>
<td>71</td>
<td>TM</td>
<td>dst</td>
<td>7[ ]</td>
<td>0</td>
</tr>
<tr>
<td>R</td>
<td>SP</td>
<td>SP</td>
<td>71</td>
<td>WDT</td>
<td>dst</td>
<td>5F</td>
<td>X</td>
</tr>
<tr>
<td>R</td>
<td>SP</td>
<td>SP</td>
<td>71</td>
<td>XOR</td>
<td>dst</td>
<td>B[ ]</td>
<td>0</td>
</tr>
<tr>
<td>R</td>
<td>SP</td>
<td>SP</td>
<td>71</td>
<td></td>
<td>dst</td>
<td>dst XOR src</td>
<td></td>
</tr>
</tbody>
</table>

† These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a '[ ]' in this table, and its value is found in the following table to the left of the applicable addressing mode pair.

For example, the opcode of an ADC instruction using the addressing modes r (destination) and Ir (source) is 13.

<table>
<thead>
<tr>
<th>Address Mode dst</th>
<th>Lower Opcode Nibble</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>r</td>
<td>Ir</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>R</td>
<td>IR</td>
</tr>
<tr>
<td>R</td>
<td>IM</td>
</tr>
<tr>
<td>IR</td>
<td>IM</td>
</tr>
</tbody>
</table>
### OPCODE MAP

<table>
<thead>
<tr>
<th>Lower Nibble (Hex)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6.5 DEC</td>
<td>6.5 DEC</td>
<td>6.5 ADD</td>
<td>6.5 ADD</td>
<td>10.5 ADD</td>
<td>10.5 ADD</td>
<td>10.5 ADD</td>
<td>10.5 ADD</td>
<td>10.5 ADD</td>
<td>12.10.5 LD,r1,r2</td>
<td>12.10.0 LD,r1,cc</td>
<td>6.5 LD</td>
<td>12.10.0 JP</td>
<td>6.5 INC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6.5 RLC</td>
<td>6.5 RLC</td>
<td>6.5 RLC</td>
<td>6.5 RLC</td>
<td>10.5 RLC</td>
<td>10.5 RLC</td>
<td>10.5 RLC</td>
<td>10.5 RLC</td>
<td>10.5 RLC</td>
<td>12/14.1 TM,r1,r2</td>
<td>10.5 TM</td>
<td>10.5 TM</td>
<td>10.5 TM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6.5 INC</td>
<td>6.5 INC</td>
<td>6.5 INC</td>
<td>6.5 INC</td>
<td>10.5 INC</td>
<td>10.5 INC</td>
<td>10.5 INC</td>
<td>10.5 INC</td>
<td>10.5 INC</td>
<td>10.5 INC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>6.5 JP</td>
<td>6.5 JPB</td>
<td>6.5 JPB</td>
<td>6.5 JPB</td>
<td>10.5 JPB</td>
<td>10.5 JPB</td>
<td>10.5 JPB</td>
<td>10.5 JPB</td>
<td>10.5 JPB</td>
<td>10.5 JPB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6.5 DA</td>
<td>6.5 DA</td>
<td>6.5 DA</td>
<td>6.5 DA</td>
<td>10.5 DA</td>
<td>10.5 DA</td>
<td>10.5 DA</td>
<td>10.5 DA</td>
<td>10.5 DA</td>
<td>10.5 DA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6.5 POP</td>
<td>6.5 POP</td>
<td>6.5 POP</td>
<td>6.5 POP</td>
<td>10.5 POP</td>
<td>10.5 POP</td>
<td>10.5 POP</td>
<td>10.5 POP</td>
<td>10.5 POP</td>
<td>10.5 POP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6.5 COM</td>
<td>6.5 COM</td>
<td>6.5 COM</td>
<td>6.5 COM</td>
<td>10.5 COM</td>
<td>10.5 COM</td>
<td>10.5 COM</td>
<td>10.5 COM</td>
<td>10.5 COM</td>
<td>10.5 COM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10.5 PUSH</td>
<td>10.5 PUSH</td>
<td>10.5 PUSH</td>
<td>10.5 PUSH</td>
<td>10.5 PUSH</td>
<td>10.5 PUSH</td>
<td>10.5 PUSH</td>
<td>10.5 PUSH</td>
<td>10.5 PUSH</td>
<td>10.5 PUSH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10.5 DECW</td>
<td>10.5 DECW</td>
<td>10.5 DECW</td>
<td>10.5 DECW</td>
<td>10.5 DECW</td>
<td>10.5 DECW</td>
<td>10.5 DECW</td>
<td>10.5 DECW</td>
<td>10.5 DECW</td>
<td>10.5 DECW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6.5 RL</td>
<td>6.5 RL</td>
<td>6.5 RL</td>
<td>6.5 RL</td>
<td>10.5 RL</td>
<td>10.5 RL</td>
<td>10.5 RL</td>
<td>10.5 RL</td>
<td>10.5 RL</td>
<td>10.5 RL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>6.5 INCW</td>
<td>6.5 INCW</td>
<td>6.5 INCW</td>
<td>6.5 INCW</td>
<td>10.5 INCW</td>
<td>10.5 INCW</td>
<td>10.5 INCW</td>
<td>10.5 INCW</td>
<td>10.5 INCW</td>
<td>10.5 INCW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>6.5 CLR</td>
<td>6.5 CLR</td>
<td>6.5 CLR</td>
<td>6.5 CLR</td>
<td>10.5 CLR</td>
<td>10.5 CLR</td>
<td>10.5 CLR</td>
<td>10.5 CLR</td>
<td>10.5 CLR</td>
<td>10.5 CLR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>6.5 RRC</td>
<td>6.5 RRC</td>
<td>6.5 RRC</td>
<td>6.5 RRC</td>
<td>10.5 RRC</td>
<td>10.5 RRC</td>
<td>10.5 RRC</td>
<td>10.5 RRC</td>
<td>10.5 RRC</td>
<td>10.5 RRC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>6.5 SRA</td>
<td>6.5 SRA</td>
<td>6.5 SRA</td>
<td>6.5 SRA</td>
<td>10.5 SRA</td>
<td>10.5 SRA</td>
<td>10.5 SRA</td>
<td>10.5 SRA</td>
<td>10.5 SRA</td>
<td>10.5 SRA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>6.5 RR</td>
<td>6.5 RR</td>
<td>6.5 RR</td>
<td>6.5 RR</td>
<td>10.5 RR</td>
<td>10.5 RR</td>
<td>10.5 RR</td>
<td>10.5 RR</td>
<td>10.5 RR</td>
<td>10.5 RR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>8.5 SWAP</td>
<td>8.5 SWAP</td>
<td>6.5 SWAP</td>
<td>6.5 SWAP</td>
<td>10.5 SWAP</td>
<td>10.5 SWAP</td>
<td>10.5 SWAP</td>
<td>10.5 SWAP</td>
<td>10.5 SWAP</td>
<td>10.5 SWAP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Bytes per Instruction

- **Execution Cycles:**
  - Upper Opcode Nibble
  - Execution Cycles
  - Pipeline Cycles

- **Mnemonic:**
  - First Operand
  - Second Operand

### Legend:
- **R** = 8-bit address
- **r** = 4-bit address
- **R1** or **r1** = Dist address
- **R2** or **r2** = Src address

### Sequence:
- Opcode, First Operand, Second Operand

### Note:
- Blank areas not defined.
- *2-byte instruction appears as a 3-byte instruction
PACKAGE INFORMATION

18-Pin DIP Package Diagram

18-Pin SOIC Package Diagram
ORDERING INFORMATION

Z86E04 (8 MHz)
Standard Temperature
18-Pin DIP 18-Pin SOIC
Z86E0408PSC Z86E0408SSC

Z86E08 (12 MHz)
Standard Temperature
18-Pin DIP 18-Pin SOIC
Z86E0812PSC Z86E0812SSC

For fast results, contact your local Zilog sales office for assistance in ordering the part(s) desired.

CODES

Preferred Package
P = Plastic DIP

Longer Lead Time
S = SOIC

Preferred Temperature
S = 0°C to +70°C

Speeds
08 = 8 MHz
12 = 12 MHz

Environmental
C = Plastic Standard

Example:
Z 86E04 08 P S C
is a Z86E04, 8 MHz, DIP, 0°C to +70°C, Plastic Standard Flow
Z86C03/C06 CMOS Z8® 8-Bit CCP™
Consumer Controller Processors

Z86E03/E06 CMOS Z8® 8-Bit OTP CCP™
Consumer Controller Processors

Z86C04/C08 CMOS Z8® Low Cost
1K /2K ROM Microcontrollers

Z86E04/E08 CMOS Z8®
8-Bit OTP Microcontrollers

Z86C07 CMOS Z8®
8-Bit Microcontroller

Z86E07 CMOS Z8®
8-Bit OTP Microcontroller

Z86C30/C31 CMOS Z8® 8-Bit CCP™
Consumer Controller Processors
FEATURES

- Low Cost, 8-Bit CMOS MCU
- 2 Kbytes of ROM
- 124 Bytes of RAM
- 18-Pin Package (DIP, SOIC)
- 3.0 to 5.5 Volt Operating Range
- Low Power Consumption: 50 mW (typical)
- Low Voltage Protection
- ROM Protection
- Fast Instruction Pointer: 1 μs @ 12 MHz
- Two Standby Modes: STOP and HALT
- Two Programmable 8-Bit Counter/Timers Each with a 6-Bit Programmable Prescaler.

GENERAL DESCRIPTION

The Z86C07 Microcontroller Unit (MCU) is a member of the Z8® single-chip microcontroller family with 2 Kbytes of ROM and 124 bytes of general-purpose RAM. Offered in an 18-pin (DIP, SOIC) package style and manufactured in CMOS technology, Zilog's low cost, low power consumption Z86C07 offers all the outstanding features of the Z8 family architecture, including easy software/hardware system expansion.

For applications demanding powerful I/O capabilities, the Z86C07 provides 14 pins dedicated to input and output. These lines are grouped into three ports, and are configurable under software control to provide I/O, timing, and status signals. The Z86C07 also features two on-board comparators that can process analog signals with a common reference voltage. There are two basic address spaces available to support this configuration: Program Memory, and 124 bytes of general-purpose registers.

The Z86C07 is characterized by a flexible I/O scheme, an efficient register and address space structure, and a number of ancillary features that are useful in many consumer and industrial applications.
GENERAL DESCRIPTION (Continued)

To unburden the system from coping with real-time tasks, such as counting/timing and I/O data communications, the Z86C07 offers two on-chip counter/timers with a large number of user selectable modes (Figure 1).

With powerful peripheral features, such as on-board comparators, counter/timer(s) and permanently enabled Watch-Dog Timer (WDT), the Z86C07 meets the needs of a variety of sophisticated controller applications.

Notes:
All Signals with a preceding front slash, "/", are active Low, e.g.: B/W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

<table>
<thead>
<tr>
<th>Connection</th>
<th>Circuit</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>Vcc</td>
<td>Vdd</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
<td>Vss</td>
</tr>
</tbody>
</table>

Figure 1. Z86C07 Functional Block Diagram
Table 1. 18-Pin DIP and SOIC Pin Identification

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>P24-P27</td>
<td>Port 2, Pins 4, 5, 6, 7</td>
<td>In/Output</td>
</tr>
<tr>
<td>5</td>
<td>V</td>
<td>Power Supply</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>XTAL2</td>
<td>Crystal Oscillator Clock</td>
<td>Input</td>
</tr>
<tr>
<td>7</td>
<td>XTAL1</td>
<td>Crystal Oscillator Clock</td>
<td>Output</td>
</tr>
<tr>
<td>8</td>
<td>P31</td>
<td>Port 3, Pin 1, AN1</td>
<td>Input</td>
</tr>
<tr>
<td>9</td>
<td>P32</td>
<td>Port 3, Pin 2, AN2</td>
<td>Input</td>
</tr>
<tr>
<td>10</td>
<td>P33</td>
<td>Port 3, Pin 3, REF</td>
<td>Input</td>
</tr>
<tr>
<td>11-13</td>
<td>P00-P02</td>
<td>Port 0, Pins 0, 1, 2</td>
<td>In/Output</td>
</tr>
<tr>
<td>14</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>15-18</td>
<td>P20-P23</td>
<td>Port 2, Pins 0, 1, 2, 3</td>
<td>In/Output</td>
</tr>
</tbody>
</table>

Figure 2. 18-Pin DIP Pin Configuration

Figure 3. 18-Pin SOIC Pin Configuration
PIN FUNCTIONS

**XTAL1, XTAL2 Crystal In, Crystal Out** (time-based input and output, respectively). These pins connect a parallel-resonant crystal, LC, or an external single-phase clock (12 MHz max) to the on-chip clock oscillator and buffer.

**Port 0** (P02-P00). Port 0 is a 3-bit I/O, nibble programmable, bidirectional, CMOS compatible I/O port. These three I/O lines can be configured under software control to be inputs or outputs (Figure 4). Inputs are Schmitt-triggered.

![Port 0 Configuration Diagram](image-url)

**Figure 4. Port 0 Configuration**
**Port 2** (P27-P20). Port 2 is an 8-bit I/O, bit programmable, bidirectional, CMOS compatible I/O port. These eight I/O lines can be configured under software control to be inputs or outputs, independently. Bits programmed as outputs may be globally programmed as either push-pull or open-drain (Figure 5). Inputs are Schmitt-triggered.

**Figure 5. Port 2 Configuration**
PIN FUNCTIONS (Continued)

Port 3 (P33-P31). Port 3 is a 3-bit, CMOS compatible port with three fixed input (P33-P31) lines. These three input lines can be configured under software control as digital inputs or analog inputs. These three input lines can also be used as the interrupt sources IRQ0-IRQ3 and as the timer input signal (TIN) (Figure 6).

![Port 3 Configuration Diagram](image)

**Figure 6. Port 3 Configuration**

**Comparator Inputs.** Two analog comparators are added to Port 3 inputs for interface flexibility.

Typical applications for the on-board comparators are: Zero crossing detection, A/D conversion, voltage scaling, and threshold detection.

The dual comparator (common inverting terminal) features a single power supply which discontinues power in STOP Mode. The common voltage range is 0-4V; the power supply and common mode rejection ratios are 90 dB and 60 dB, respectively.

Interrupts are generated on either edge of Comparator 2’s output, or on the falling edge of Comparator 1’s output. The comparator output may be used for interrupt generation, Port 3 data inputs, or TIN through P31. Alternatively, the comparators may be disabled, freeing the reference input (P33) for use as IRQ1 and/or P33 input.
FUNCTIONAL DESCRIPTION

Reset. Upon power-up the Power-On Reset circuit waits for $T_{POR}$ plus 18 clock cycles, and then starts program execution at address %000C (HEX) (Figure 7). Reference the Z86C07 control registers' Reset value (Table 2).

![Diagram of Internal Reset Configuration](image)

Table 2. Z86C07 Control Registers

<table>
<thead>
<tr>
<th>Addr.</th>
<th>Reg.</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>TMR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>F2</td>
<td>T1</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>F3</td>
<td>PRE1</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>F4</td>
<td>T0</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>F5</td>
<td>PRE0</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>F6*</td>
<td>P2M</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Inputs after reset</td>
</tr>
<tr>
<td>F7*</td>
<td>P3M</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>F8*</td>
<td>P01M</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>O</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>F9</td>
<td>IPR</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>FA</td>
<td>IRQ</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>IRQ3 is used for positive edge detection</td>
</tr>
<tr>
<td>PB</td>
<td>IMR</td>
<td>O</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td>FLAGS</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>FD</td>
<td>RP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>FE</td>
<td>SPH</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>Not used, stack always internal</td>
</tr>
<tr>
<td>FF</td>
<td>SPL</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
</tbody>
</table>

Note:
* Registers are not reset after a STOP-Mode Recovery using P27 pin. A subsequent reset will cause these control registers to be reconfigured as shown in Table 3 and the user must avoid bus contention on the port pins or it may affect device reliability.

Program Memory. The Z86C07 can address up to 2 Kbytes of internal program memory (Figure 8). The first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts. Bytes 0-2048 are on-chip mask-programmed ROM.

![Program Memory Map](image)
**FUNCTIONAL DESCRIPTION** (Continued)

**Register File.** The Register File consists of three I/O port registers, 124 general-purpose registers, and 15 control and status registers (R3-R0, R127-R4 and R255-R241, respectively - Figure 9). The Z86C07 instructions can access registers directly or indirectly through an 8-bit address field. This allows short, 4-bit register addressing using the Register Pointer. In the 4-bit mode, the register file is divided into eight working register groups, each occupying 16 continuous locations. The Register Pointer (Figure 10) addresses the starting location of the active working-register group.

**General-Purpose Registers (GPR).** These registers are undefined after the device is powered up. The registers keep their last value after any reset, as long as the reset occurs in the Vcc voltage-specified operating range. Note: Register R254 has been designated as a general-purpose register.

<table>
<thead>
<tr>
<th>Location</th>
<th>Identifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>255</td>
<td>Stack Pointer (Bits 7-0)</td>
</tr>
<tr>
<td>254</td>
<td>General Purpose Register</td>
</tr>
<tr>
<td>253</td>
<td>Register Pointer</td>
</tr>
<tr>
<td>252</td>
<td>Program Control Flags</td>
</tr>
<tr>
<td>251</td>
<td>Interrupt Mask Register</td>
</tr>
<tr>
<td>250</td>
<td>Interrupt Request Register</td>
</tr>
<tr>
<td>249</td>
<td>Interrupt Priority Register</td>
</tr>
<tr>
<td>248</td>
<td>Ports 0-1 Mode</td>
</tr>
<tr>
<td>247</td>
<td>Port 3 Mode</td>
</tr>
<tr>
<td>246</td>
<td>Port 2 Mode</td>
</tr>
<tr>
<td>245</td>
<td>T0 Prescaler</td>
</tr>
<tr>
<td>244</td>
<td>Timer/Counter 0</td>
</tr>
<tr>
<td>243</td>
<td>T1 Prescaler</td>
</tr>
<tr>
<td>242</td>
<td>Timer/Counter 1</td>
</tr>
<tr>
<td>241</td>
<td>Timer Mode</td>
</tr>
<tr>
<td>240</td>
<td>Not Implemented</td>
</tr>
<tr>
<td>128</td>
<td>General Purpose Registers</td>
</tr>
<tr>
<td>127</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Port 3</td>
</tr>
<tr>
<td>3</td>
<td>Port 2</td>
</tr>
<tr>
<td>2</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>P1</td>
</tr>
<tr>
<td>0</td>
<td>P0</td>
</tr>
</tbody>
</table>

**Figure 9. Register File**
Stack Pointer. The Z86C07 has an 8-bit Stack Pointer (R255) used for the internal stack that resides within the 124 general-purpose registers.

Counter/Timer. There are two 8-bit programmable counter/timers (T0 and T1), each driven by its own 6-bit programmable prescaler. The T1 prescaler can be driven by internal or external clock sources, however the T0 can be driven by the internal clock source only (Figure 11).

The 6-bit prescalers can divide the input frequency of the clock source by any integer number from 1 to 64. Each prescaler drives its counter, which decrements the value (1 to 256) that has been loaded into the counter. When both counter and prescaler reach the end of count, a timer interrupt request IRQ4 (T0) or IRQ5 (T1) is generated.

The counter can be programmed to start, stop, restart to continue, or restart from the initial value. The counters can also be programmed to stop upon reaching zero (single pass mode) or to automatically reload the initial value and continue counting (modulo-n continuous mode).

The counters, but not the prescalers, are read at any time without disturbing their value or count mode. The clock source for T1 is user-definable and can be either the internal microprocessor clock divided by four, or an external signal input through Port 3. The Timer Mode register configures the external timer input (P30) as an external clock, a trigger input that is retriggerable or not retriggerable, or as a gate input for the internal clock.

Figure 11. Counter/Timers Block Diagram

* Note: Divide-by-two is not used in Low EMI Mode.
FUNCTIONAL DESCRIPTION (Continued)

Interrupts. The Z86C07 has six interrupts from six different sources. These interrupts are maskable and prioritized (Figure 12). The six sources are divided as follows: the falling edge of P31 (AN1), P32 (AN2), P33 (REF), the rising edge of P32 (AN2), and the two counter/timers. The Interrupt Mask Register globally or individually enables or disables the six interrupt requests (Table 3).

When more than one interrupt is pending, priorities are resolved by a programmable priority encoder that is controlled by the Interrupt Priority register. All Z86C07 interrupts are vectored through locations in program memory. When an interrupt machine cycle is activated, an interrupt request is granted. This disables all subsequent interrupts, saves the Program Counter and Status Flags, and then branches to the program memory vector location reserved for that interrupt. This memory location and the next byte contain the 16-bit starting address of the interrupt service routine for that particular interrupt request.

To accommodate polled interrupt systems, interrupt inputs are masked and the interrupt request register is polled to determine which of the interrupt requests needs service.

Note: User must select any Z86C08 mode in Zilog’s C12 ICEBOX™ emulator. The rising edge interrupt is not supported on the Z86CCP00ZEM emulator.

Table 3. Interrupt Types, Sources, and Vectors

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Vector Location</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRQ0</td>
<td>AN2(P32)</td>
<td>0,1</td>
<td>External (F)Edge</td>
</tr>
<tr>
<td>IRQ1</td>
<td>REF(P33)</td>
<td>2,3</td>
<td>External (F)Edge</td>
</tr>
<tr>
<td>IRQ2</td>
<td>AN1(P31)</td>
<td>4,5</td>
<td>External (F)Edge</td>
</tr>
<tr>
<td>IRQ3</td>
<td>AN2(P32)</td>
<td>6,7</td>
<td>External (R)Edge</td>
</tr>
<tr>
<td>IRQ4</td>
<td>T0</td>
<td>8,9</td>
<td>Internal</td>
</tr>
<tr>
<td>IRQ5</td>
<td>T1</td>
<td>10,11</td>
<td>Internal</td>
</tr>
</tbody>
</table>

Notes:
F = Falling edge triggered
R = Rising edge triggered

Figure 12. Interrupt Block Diagram
Clock. The Z86C07 on-chip oscillator has a high-gain, parallel-resonant amplifier for connection to a crystal, ceramic resonator, or any suitable external clock source (XTAL1 = Input, XTAL2 = Output). The crystal should be AT cut, 12 MHz max, with a series resistance (RS) less than or equal to 100 Ohms.

The crystal should be connected across XTAL1 and XTAL2 using the vendor's crystal recommended capacitors from each pin directly to device ground pin 14 (Figure 13).

Note that the crystal capacitor loads should be connected to Vss, pin 14 to reduce Ground noise injection.

HALT Mode. This instruction turns off the internal CPU clock, but not the crystal oscillation. The counter/timers and external interrupts IRQO, IRQ1, IRQ2, and IRQ3 remain active. The device can be recovered by interrupts, either externally or internally generated. The program execution begins at location 000C (HEX). An interrupt request must be executed (enabled) to exit HALT mode. After the interrupt service routine, the program continues from the instruction after the HALT.

STOP Mode. This instruction turns off the internal clock and external crystal oscillation and reduces the standby current to less than 10 µA. The STOP Mode can be released by two methods. The first method is a RESET of the device by removing Vcc. The second method is if P27 is configured as an input line when the device executes the STOP instruction. A low input level on P27 releases the STOP Mode.

Program execution under both conditions begins at location 000C (HEX). However, when P27 is used to release the STOP Mode, the I/O port mode registers are not reconfigured to their default power-on conditions. This prevents any I/O, configured as output when the STOP instruction was executed, from glitching to an unknown state.

To use the P27 release approach with STOP Mode, use the following instruction:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD P2M, #1XXXXXXXB</td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td></td>
</tr>
<tr>
<td>STOP</td>
<td></td>
</tr>
<tr>
<td>X = depends on user's application</td>
<td></td>
</tr>
</tbody>
</table>

In order to enter STOP or HALT mode, it is necessary to first flush the instruction pipeline to avoid suspending execution in mid-instruction. To do this, the user must execute a NOP (opcode = FFH) immediately before the appropriate SLEEP instruction. i.e.:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF NOP</td>
<td>clear the pipeline</td>
</tr>
<tr>
<td>6F STOP</td>
<td>enter STOP mode</td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>FF NOP</td>
<td>clear the pipeline</td>
</tr>
<tr>
<td>7F HALT</td>
<td>enter HALT mode</td>
</tr>
</tbody>
</table>

Watch-Dog Timer (WDT). The Watch-Dog Timer is permanently enabled. The WDT should be refreshed at least every Twdt; otherwise, the Z86C07 resets itself.

WDT = SF (HEX).

Opcode WDT (5FH). Execution of this command clears the WDT counter. This has to be done at least every Twdt. Otherwise, the WDT times out and generates a Reset. The generated Reset is the same as a Power-On Reset of T_{POR}, plus 18 XTAL clock cycles. The WDT instruction affects the Flags accordingly: Z = 1, S = 0, V = 0.
FUNCTIONAL DESCRIPTION (Continued)

ROM Protect. ROM Protect fully protects the Z86C07 ROM code from being read internally. When ROM Protect is selected, the Z86C07 will disable the instructions LDC and LOCI (Z86C04/C08 and Z86E04/E08 do not support the instructions of LDE and LDEI) in all modes. ROM look-up tables cannot be used in this mode.

Low Voltage Protection (VLV). The Low Voltage trip voltage ($V_{LV}$) is less than 3 volts and above 1.4 volts under the following conditions:

Maximum ($V_{LV}$) Conditions:

Case 1: $T_A = -40^\circ C$, $+105^\circ C$, Internal Clock Frequency equal or less than 1 MHz

Case 2: $T_A = -40^\circ C$, $+85^\circ C$, Internal Clock Frequency equal or less than 2 MHz

Note: The internal clock frequency is one-half the external clock frequency.

The device will function normally at or above 3.0V under all conditions. Below 3.0V, the device functions normally until the Low Voltage Protection trip point ($V_{LV}$) is reached. The device is guaranteed to function normally at supply voltages above the low voltage trip point for the temperatures and operating frequencies in Case 1 and Case 2 above. The actual low voltage trip point is a function of temperature and process parameters (Figure 14).

<table>
<thead>
<tr>
<th>Temp</th>
<th>-40°C</th>
<th>0°C</th>
<th>+25°C</th>
<th>+70°C</th>
<th>+105°C</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{LV}$</td>
<td>2.55</td>
<td>2.4</td>
<td>2.1</td>
<td>1.7</td>
<td>1.6</td>
</tr>
</tbody>
</table>

2 MHz (Typical)

![Power-On Reset threshold for $V_{CC}$ and 4 MHz $V_{LV}$ overlap](Figure 14. Typical Z86C07 $V_{LV}$ vs Temperature)
### Absolute Maximum Ratings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient Temperature under Bias</td>
<td>-40</td>
<td>+105</td>
<td>°C</td>
</tr>
<tr>
<td>Storage Temperature</td>
<td>-65</td>
<td>+150</td>
<td>°C</td>
</tr>
<tr>
<td>Voltage on any Pin with Respect to V_{SS} (Note 1)</td>
<td>-0.6</td>
<td>+12</td>
<td>V</td>
</tr>
<tr>
<td>Voltage on V_{PP} Pin with Respect to V_{SS}</td>
<td>-0.3</td>
<td>+7</td>
<td>V</td>
</tr>
<tr>
<td>Voltage on Pin 7 with Respect to V_{SS} (Note 2)</td>
<td>-0.6</td>
<td>V_{PP} + 1</td>
<td>V</td>
</tr>
<tr>
<td>Total Power Dissipation</td>
<td>462</td>
<td>mW</td>
<td></td>
</tr>
<tr>
<td>Maximum Current out of V_{SS}</td>
<td>85</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td>Maximum Current into V_{PP}</td>
<td>85</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td>Maximum Current into an Input Pin (Note 3)</td>
<td>±600</td>
<td>µA</td>
<td></td>
</tr>
<tr>
<td>Maximum Current into an Open-Drain Pin (Note 4)</td>
<td>±600</td>
<td>µA</td>
<td></td>
</tr>
<tr>
<td>Maximum Output Current Sunked by any I/O Pin</td>
<td>12</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td>Maximum Output Current Sourced by any I/O Pin</td>
<td>12</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td>Total Maximum Output Current Sunked by Port 2</td>
<td>70</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td>Total Maximum Output Current Sourced by Port 2</td>
<td>70</td>
<td>mA</td>
<td></td>
</tr>
</tbody>
</table>

**Notice:**

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability. Total power dissipation should not exceed 616 mW for the package. Power dissipation is calculated as follows:

\[
\text{Total Power Dissipation} = V_{DD} \times I_{DD} - \text{(sum of } I_{IH}) + \text{sum of } [(V_{DD} - V_{OL}) \times I_{OL}] + \text{sum of } (V_{OH} \times I_{OH})
\]
STANDARD TEST CONDITIONS

The characteristics listed below apply for standard test conditions as noted. All voltages are referenced to Ground. Positive current flows into the referenced pin (Figure 15).

CAPACITANCE

$T_a = 25^\circ C, V_{cc} = GND = 0V, f = 1.0 \text{ MHz}, \text{ unmeasured pins returned to GND}$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input capacitance</td>
<td>0</td>
<td>10 pF</td>
</tr>
<tr>
<td>Output capacitance</td>
<td>0</td>
<td>20 pF</td>
</tr>
<tr>
<td>I/O capacitance</td>
<td>0</td>
<td>25 pF</td>
</tr>
</tbody>
</table>

$V_{cc}$ SPECIFICATION

$V_{cc} = 3.0V \text{ to } 5.0V$

Typicals are at 3.3V and 5.0V.
## DC ELECTRICAL CHARACTERISTICS

<table>
<thead>
<tr>
<th>Sym</th>
<th>Parameter</th>
<th>$V_{cc}$ [4] $T_a = 0^\circ$C to $+70^\circ$C</th>
<th>$T_a = -40^\circ$C to $+105^\circ$C</th>
<th>Typical @ 25°C</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{CH}$</td>
<td>Clock Input High Voltage</td>
<td>3.0V</td>
<td>0.8 $V_{cc}$ $V_{cc}+0.3$</td>
<td>0.8 $V_{cc}$ $V_{cc}+0.3$</td>
<td>1.7</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.8 $V_{cc}$ $V_{cc}+0.3$</td>
<td>0.8 $V_{cc}$ $V_{cc}+0.3$</td>
<td>2.75</td>
<td>V</td>
</tr>
<tr>
<td>$V_{CL}$</td>
<td>Clock Input Low Voltage</td>
<td>3.0V</td>
<td>$V_{ss}+0.3$</td>
<td>0.2 $V_{cc}$ $V_{ss}+0.3$</td>
<td>0.2 $V_{cc}$</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{ss}+0.3$</td>
<td>0.2 $V_{cc}$ $V_{ss}+0.3$</td>
<td>0.2 $V_{cc}$</td>
<td>1.5</td>
</tr>
<tr>
<td>$V_{IH}$</td>
<td>Input High Voltage</td>
<td>3.0V</td>
<td>0.7 $V_{cc}$ $V_{cc}+0.3$</td>
<td>0.7 $V_{cc}$ $V_{cc}+0.3$</td>
<td>1.8</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.7 $V_{cc}$ $V_{cc}+0.3$</td>
<td>0.7 $V_{cc}$ $V_{cc}+0.3$</td>
<td>2.8</td>
<td>V</td>
</tr>
<tr>
<td>$V_{IL}$</td>
<td>Input Low Voltage</td>
<td>3.0V</td>
<td>$V_{ss}+0.3$</td>
<td>0.2 $V_{cc}$ $V_{ss}+0.3$</td>
<td>0.2 $V_{cc}$</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{ss}+0.3$</td>
<td>0.2 $V_{cc}$ $V_{ss}+0.3$</td>
<td>0.2 $V_{cc}$</td>
<td>1.5</td>
</tr>
<tr>
<td>$V_{OH}$</td>
<td>Output High Voltage</td>
<td>3.0V</td>
<td>$V_{cc}+0.4$</td>
<td>$V_{cc}+0.4$</td>
<td>2.7</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{cc}+0.4$</td>
<td>$V_{cc}+0.4$</td>
<td>5.5</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>$V_{cc}+0.4$</td>
<td>$V_{cc}+0.4$</td>
<td>V</td>
<td>Low Noise @ 0.5 mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{cc}+0.4$</td>
<td>$V_{cc}+0.4$</td>
<td>V</td>
<td>Low Noise @ 0.5 mA</td>
</tr>
<tr>
<td>$V_{OL1}$</td>
<td>Output Low Voltage</td>
<td>3.0V</td>
<td>0.8</td>
<td>0.8</td>
<td>0.3</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.4</td>
<td>0.4</td>
<td>0.2</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>0.4</td>
<td>0.4</td>
<td>V</td>
<td>Low Noise @ 0.5 mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.4</td>
<td>0.4</td>
<td>V</td>
<td>Low Noise @ 0.5 mA</td>
</tr>
<tr>
<td>$V_{OL2}$</td>
<td>Output Low Voltage</td>
<td>3.0V</td>
<td>1.0</td>
<td>1.0</td>
<td>0.8</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.8</td>
<td>0.8</td>
<td>0.3</td>
<td>V</td>
</tr>
<tr>
<td>$V_{OFFSET}$</td>
<td>Comparator Input Offset Voltage</td>
<td>3.0V</td>
<td>25</td>
<td>25</td>
<td>10</td>
<td>mV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
<td>10</td>
<td>mV</td>
</tr>
<tr>
<td>$V_{LV}$</td>
<td>$V_{cc}$ Low Voltage</td>
<td></td>
<td>2.7</td>
<td>2.95</td>
<td>2.1</td>
<td>V</td>
</tr>
<tr>
<td>$I_{IL}$</td>
<td>Input Leakage</td>
<td>3.0V</td>
<td>$-1.0$</td>
<td>1.0</td>
<td>$-1.0$</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$-1.0$</td>
<td>1.0</td>
<td>$-1.0$</td>
<td>1.0</td>
</tr>
<tr>
<td>$I_{OL}$</td>
<td>Output Leakage</td>
<td>3.0V</td>
<td>$-1.0$</td>
<td>1.0</td>
<td>$-1.0$</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$-1.0$</td>
<td>1.0</td>
<td>$-1.0$</td>
<td>1.0</td>
</tr>
<tr>
<td>$V_{ICMR}$</td>
<td>Input Common Mode Range</td>
<td></td>
<td>0</td>
<td>$V_{cc}+1.0$</td>
<td>0</td>
<td>$V_{cc}+1.5$</td>
</tr>
<tr>
<td>Sym</td>
<td>Parameter</td>
<td>$V_{CC}[4]$</td>
<td>$T_A = 0°C$ to $+70°C$</td>
<td>$T_A = -40°C$ to $+105°C$</td>
<td>Typical @ 25°C</td>
<td>Units</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------------------</td>
<td>-------------</td>
<td>------------------------</td>
<td>---------------------------</td>
<td>-----------------</td>
<td>-------</td>
</tr>
<tr>
<td>I_{CC}</td>
<td>Supply Current</td>
<td>3.2V</td>
<td>70</td>
<td>70</td>
<td>3.5</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>7.0</td>
<td>7.0</td>
<td>3.5</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>5.0</td>
<td>8.0</td>
<td>2.5</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>11.0</td>
<td>11.0</td>
<td>5.3</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>7.5</td>
<td>10</td>
<td>3.0</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>15</td>
<td>15</td>
<td>7.5</td>
<td>mA</td>
</tr>
<tr>
<td>I_{CC1}</td>
<td>Standby Current</td>
<td>3.0V</td>
<td>2.5</td>
<td>2.5</td>
<td>0.7</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>4.0</td>
<td>5.0</td>
<td>2.0</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>3.0</td>
<td>4.0</td>
<td>1.0</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>5.0</td>
<td>5.0</td>
<td>2.8</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>4.5</td>
<td>4.5</td>
<td>1.5</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>7.0</td>
<td>7.0</td>
<td>4.0</td>
<td>mA</td>
</tr>
<tr>
<td>I_{CC}</td>
<td>Supply Current (Low Noise Mode)</td>
<td>3.0V</td>
<td>3.5</td>
<td>3.5</td>
<td>1.0</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>7.0</td>
<td>7.0</td>
<td>2.8</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>5.8</td>
<td>5.8</td>
<td>1.5</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>9.0</td>
<td>9.0</td>
<td>3.5</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>8.0</td>
<td>8.0</td>
<td>2.0</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>11.0</td>
<td>11.0</td>
<td>5.4</td>
<td>mA</td>
</tr>
</tbody>
</table>
### Sym Parameter

<table>
<thead>
<tr>
<th>Sym</th>
<th>Parameter</th>
<th>$V_{cc}$[4]</th>
<th>$T_a = 0^\circ C$ to $+70^\circ C$</th>
<th>$T_a = -40^\circ C$ to $+105^\circ C$</th>
<th>Typical @ 25°C</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{cc1}$</td>
<td>Standby Current (Low Noise Mode)</td>
<td>3.0V</td>
<td>1.2</td>
<td>1.2</td>
<td>0.3</td>
<td>mA</td>
<td>HALT Mode $V_{in} = 0V$, $V_{cc} @ 1$ MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>1.6</td>
<td>1.6</td>
<td>0.5</td>
<td>mA</td>
<td>HALT Mode $V_{in} = 0V$, $V_{cc} @ 1$ MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>1.5</td>
<td>1.5</td>
<td>0.5</td>
<td>mA</td>
<td>HALT Mode $V_{in} = 0V$, $V_{cc} @ 2$ MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>1.9</td>
<td>1.9</td>
<td>1</td>
<td>mA</td>
<td>HALT Mode $V_{in} = 0V$, $V_{cc} @ 2$ MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>2.0</td>
<td>2.0</td>
<td>0.9</td>
<td>mA</td>
<td>HALT Mode $V_{in} = 0V$, $V_{cc} @ 2$ MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>2.4</td>
<td>2.4</td>
<td>1.6</td>
<td>mA</td>
<td>HALT Mode $V_{in} = 0V$, $V_{cc} @ 4$ MHz</td>
</tr>
<tr>
<td>$I_{cc2}$</td>
<td>Standby Current</td>
<td>3.0V</td>
<td>10</td>
<td>20</td>
<td>1.2</td>
<td>$\mu$A</td>
<td>STOP Mode $V_{in} = 0V$, $V_{cc}$ WDT is not Running</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>10</td>
<td>20</td>
<td>2.0</td>
<td>$\mu$A</td>
<td>STOP Mode $V_{in} = 0V$, $V_{cc}$ WDT is not Running</td>
</tr>
</tbody>
</table>

**Notes:**

[1] $I_{cc1}$

<table>
<thead>
<tr>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock Driven</td>
<td>0.3</td>
<td>5.0</td>
<td>mA</td>
</tr>
<tr>
<td>Crystal/Resonator</td>
<td>3.5</td>
<td>5.0</td>
<td>mA</td>
</tr>
</tbody>
</table>

[2] $V_{cc} = 0V = GND$

[3] For 2.75V operating, the device operates down to $V_{L}$. The minimum operational $V_{cc}$ is determined on the value of the voltage $V_{L}$ at the ambient temperature. The $V_{L}$ increases as the temperature decreases.

[4] The $V_{cc}$ voltage specification of 3.0V guarantees 3.3V ±0.3V and $V_{cc}$ voltage specification of 5.5V guarantees 5.0V ±0.5V.

[5] Standard Mode (not Low EMI Mode)


DC ELECTRICAL CHARACTERISTICS
Timing Diagrams

Figure 16. Electrical Timing Diagram
### AC ELECTRICAL CHARACTERISTICS
Timing Table (Standard Mode)

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}[3]$</th>
<th>8 MHz</th>
<th>12 MHz</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TpC</td>
<td>Input Clock Period</td>
<td>3.0V</td>
<td>125 DC</td>
<td>83 DC</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>125 DC</td>
<td>83 DC</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>2</td>
<td>TrC, TfC</td>
<td>Clock Input Rise and Fall Times</td>
<td>3.0V</td>
<td>25</td>
<td>15</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>15</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>3</td>
<td>TwC</td>
<td>Input Clock Width</td>
<td>3.0V</td>
<td>62</td>
<td>41</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>62</td>
<td>41</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>4</td>
<td>TwTinL</td>
<td>Timer Input Low Width</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>5</td>
<td>TwTinH</td>
<td>Timer Input High Width</td>
<td>3.0V</td>
<td>5 TpC</td>
<td>5 TpC</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5 TpC</td>
<td>5 TpC</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>6</td>
<td>TpTin</td>
<td>Timer Input Period</td>
<td>3.0V</td>
<td>8 TpC</td>
<td>8 TpC</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>8 TpC</td>
<td>8 TpC</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>7</td>
<td>TrTin, TfTin</td>
<td>Timer Input Rise and Fall Timer</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>8</td>
<td>TwlL</td>
<td>Int. Request Input Low Time</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td>9</td>
<td>TwlH</td>
<td>Int. Request Input High Time</td>
<td>3.0V</td>
<td>5 TpC</td>
<td>5 TpC</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5 TpC</td>
<td>5 TpC</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td>10</td>
<td>Twdt</td>
<td>Watch-Dog Timer Delay Time</td>
<td>3.0V</td>
<td>25</td>
<td>25</td>
<td>ms</td>
<td>[1,4]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5</td>
<td>5</td>
<td>ms</td>
<td>[1,4]</td>
</tr>
<tr>
<td>11</td>
<td>Tpor</td>
<td></td>
<td>3.0V</td>
<td>24</td>
<td>24</td>
<td>ms</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>12</td>
<td>12</td>
<td>ms</td>
<td>[1]</td>
</tr>
</tbody>
</table>

**Notes:**

[1] Timing Reference uses 0.7 $V_{cc}$ for a logic 1 and 0.2 $V_{cc}$ for a logic 0.
[2] Interrupt request through Port 3 (P33-P31).
[3] The $V_{cc}$ Voltage specification of 3.0V guarantees 3.3V ±0.3V and $V_{cc}$ voltage specification of 5.5V guarantees 5.0V ±0.5V.
[4] Length of time before WDT times out.
Low Noise Version

Low EMI Emission
The Z86C07 can be programmed to operate in a Low EMI emission mode by means of a mask ROM bit option. Use of this feature results in:

- Less than 1 mA consumed during HALT mode, -0°C to +70°C.
- All pre-driver slew rates reduced to 10 ns typical.
- Internal SCLK/TCLK = XTAL operation limited to a maximum of 4 MHz - 250 ns cycle time.
- Output drivers have resistances of 200 ohms (typical).
- Oscillator divide-by-two circuitry eliminated.

The Low EMI mode is mask-programmable to be selected by the customer at the time the ROM code is submitted.

EMI Characteristics
The Z86C07 operating in the Low EMI mode generates EMI as measured in the following chart:

The measurements shown in Figure 17 were made while operating the Z86C07 in three states: (1) Idle condition; (2) static output; (3) switched output.

![Near Field EMI Analysis](image-url)

*Figure 17. Low Noise Analysis*
### AC ELECTRICAL CHARACTERISTICS

**Low Noise Mode**

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$[3]</th>
<th>1 MHz</th>
<th>4 MHz</th>
<th>1 MHz</th>
<th>4 MHz</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>TwTinH</td>
<td>Timer Input High Width</td>
<td>3.0V</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td></td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td></td>
<td>[1]</td>
</tr>
<tr>
<td>6</td>
<td>TpTin</td>
<td>Timer Input Period</td>
<td>3.0V</td>
<td>4TpC</td>
<td>4TpC</td>
<td>4TpC</td>
<td>4TpC</td>
<td></td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>4TpC</td>
<td>4TpC</td>
<td>4TpC</td>
<td>4TpC</td>
<td></td>
<td>[1]</td>
</tr>
<tr>
<td>7</td>
<td>TrTin, TtTin</td>
<td>Timer Input Rise and Fall Timer</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>8</td>
<td>TwIL</td>
<td>Int. Request Input</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low Time</td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td>9</td>
<td>TwIH</td>
<td>Int. Request Input</td>
<td>3.0V</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td></td>
<td>[1,2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High Time</td>
<td>5.5V</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td>2.5TpC</td>
<td></td>
<td>[1,2]</td>
</tr>
<tr>
<td>10</td>
<td>Twdt</td>
<td>Watch-Dog Timer</td>
<td>3.0V</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>ms</td>
<td>[1,4]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Delay Time</td>
<td>5.5V</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>ms</td>
<td>[1,4]</td>
</tr>
</tbody>
</table>

**Notes:**

1. Timing Reference uses 0.7 $V_{cc}$ for a logic 1 and 0.2 $V_{cc}$ for a logic 0.
2. Interrupt request through Port 3 (P33-P31).
3. The $V_{cc}$ Voltage specification of 3.0V guarantees 3.3V ±0.3V and $V_{cc}$ Voltage specification of 5.5V guarantees 5.0V ±0.5V.
4. Length of time before WDT times out.
Z8 CONTROL REGISTER DIAGRAMS

Figure 18. Timer Mode Register (F1n: Read/Write)

Figure 19. Counter Time 1 Register (F2n: Read/Write)

Figure 20. Prescaler 1 Register (F3n: Write Only)

Figure 21. Counter/Timer 0 Register (F4n: Read/Write)

Figure 22. Prescaler 0 Register (F5n: Write Only)

Figure 23. Port 2 Mode Register (F6n: Write Only)

Figure 24. Port 3 Mode Register (F7n: Write Only)
Figure 25. Port 0 and 1 Mode Register (F8H: Write Only)

Figure 26. Interrupt Priority Register (F9H: Write Only)

Figure 28. Interrupt Mask Register (FBH: Read/Write)

Figure 29. Flag Register (FCH: Read/Write)

Figure 30. Register Pointer (FDH: Read/Write)

Figure 31. Stack Pointer (FFH: Read/Write)
DEVICE CHARACTERISTICS

Figure 32. Maximum $I_{cc}$ vs Frequency

Figure 33. Typical $I_{cc}$ vs Frequency
Figure 34. $V_{OL}$, $V_{OL}$ vs Temperature

Figure 35. $V_{IH}$, $V_{OH}$ vs Temperature
DEVICE CHARACTERISTICS (Continued)

Figure 36. Typical $I_{on}$ vs $V_{on}$

Figure 37. Typical $I_{ol}$ vs $V_{ol}$
INSTRUCTION SET NOTATION

Addressing Modes. The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRR</td>
<td>Indirect register pair or indirect working-register pair address</td>
</tr>
<tr>
<td>Ir</td>
<td>Indirect working-register pair only</td>
</tr>
<tr>
<td>X</td>
<td>Indexed address</td>
</tr>
<tr>
<td>DA</td>
<td>Direct address</td>
</tr>
<tr>
<td>RA</td>
<td>Relative address</td>
</tr>
<tr>
<td>IM</td>
<td>Immediate</td>
</tr>
<tr>
<td>R</td>
<td>Register or working-register address</td>
</tr>
<tr>
<td>r</td>
<td>Working register address only</td>
</tr>
<tr>
<td>IR</td>
<td>Indirect-register or indirect working-register address</td>
</tr>
<tr>
<td>Ir</td>
<td>Indirect working-register address only</td>
</tr>
<tr>
<td>RR</td>
<td>Register pair or working register pair address</td>
</tr>
</tbody>
</table>

Symbols. The following symbols are used in describing the instruction set.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst</td>
<td>Destination location or contents</td>
</tr>
<tr>
<td>src</td>
<td>Source location or contents</td>
</tr>
<tr>
<td>cc</td>
<td>Condition code</td>
</tr>
<tr>
<td>@</td>
<td>Indirect address prefix</td>
</tr>
<tr>
<td>SP</td>
<td>Stack pointer</td>
</tr>
<tr>
<td>PC</td>
<td>Program counter</td>
</tr>
<tr>
<td>FLAGS</td>
<td>Flag register (Control Register 252)</td>
</tr>
<tr>
<td>RP</td>
<td>Register Pointer (R253)</td>
</tr>
<tr>
<td>IMR</td>
<td>Interrupt mask register (R251)</td>
</tr>
</tbody>
</table>

Flags. Control register (R252) contains the following six flags.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Carry flag</td>
</tr>
<tr>
<td>Z</td>
<td>Zero flag</td>
</tr>
<tr>
<td>S</td>
<td>Sign flag</td>
</tr>
<tr>
<td>V</td>
<td>Overflow flag</td>
</tr>
<tr>
<td>D</td>
<td>Decimal-adjust flag</td>
</tr>
<tr>
<td>H</td>
<td>Half-carry flag</td>
</tr>
</tbody>
</table>

Affected flags are indicated by:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Clear to zero</td>
</tr>
<tr>
<td>1</td>
<td>Set to one</td>
</tr>
<tr>
<td>*</td>
<td>Set to clear according to operation</td>
</tr>
<tr>
<td>-</td>
<td>Unaffected</td>
</tr>
<tr>
<td>X</td>
<td>Undefined</td>
</tr>
</tbody>
</table>
### Condition Codes

<table>
<thead>
<tr>
<th>Value</th>
<th>Mnemonic</th>
<th>Meaning</th>
<th>Flags Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>---</td>
<td>Always true</td>
<td>---</td>
</tr>
<tr>
<td>0111</td>
<td>C</td>
<td>Carry</td>
<td>C = 1</td>
</tr>
<tr>
<td>1111</td>
<td>NC</td>
<td>No Carry</td>
<td>C = 0</td>
</tr>
<tr>
<td>0110</td>
<td>Z</td>
<td>Zero</td>
<td>Z = 1</td>
</tr>
<tr>
<td>1110</td>
<td>NZ</td>
<td>Not zero</td>
<td>Z = 0</td>
</tr>
<tr>
<td>1101</td>
<td>PL</td>
<td>Plus</td>
<td>S = 0</td>
</tr>
<tr>
<td>0101</td>
<td>MI</td>
<td>Minus</td>
<td>S = 1</td>
</tr>
<tr>
<td>0100</td>
<td>OV</td>
<td>Overflow</td>
<td>V = 1</td>
</tr>
<tr>
<td>1100</td>
<td>NOV</td>
<td>No overflow</td>
<td>V = 0</td>
</tr>
<tr>
<td>0110</td>
<td>EQ</td>
<td>Equal</td>
<td>Z = 1</td>
</tr>
<tr>
<td>1110</td>
<td>NE</td>
<td>Not equal</td>
<td>Z = 0</td>
</tr>
<tr>
<td>1001</td>
<td>GE</td>
<td>Greater than or equal</td>
<td>(S XOR V) = 0</td>
</tr>
<tr>
<td>0001</td>
<td>LT</td>
<td>Less than</td>
<td>(S XOR V) = 1</td>
</tr>
<tr>
<td>1010</td>
<td>GT</td>
<td>Greater than</td>
<td>[Z OR (S XOR V)] = 0</td>
</tr>
<tr>
<td>0010</td>
<td>LE</td>
<td>Less than or equal</td>
<td>[Z OR (S XOR V)] = 1</td>
</tr>
<tr>
<td>1111</td>
<td>UGE</td>
<td>Unsigned greater than or equal</td>
<td>C = 0</td>
</tr>
<tr>
<td>0111</td>
<td>ULT</td>
<td>Unsigned less than</td>
<td>C = 1</td>
</tr>
<tr>
<td>1011</td>
<td>UGT</td>
<td>Unsigned greater than</td>
<td>(C = 0 AND Z = 0) = 1</td>
</tr>
<tr>
<td>0011</td>
<td>ULE</td>
<td>Unsigned less than or equal</td>
<td>(C OR Z) = 1</td>
</tr>
<tr>
<td>0000</td>
<td>F</td>
<td>Never True (Always False)</td>
<td>---</td>
</tr>
</tbody>
</table>
### INSTRUCTION FORMATS

One-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>OPC</th>
<th>MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dst</td>
<td>src</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dst</td>
<td>src</td>
<td></td>
</tr>
<tr>
<td>CLR, CPL, DA, DEC,</td>
<td>ADC, ADD, AND, CP,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSW, INC, INCW,</td>
<td>LD, OR, SBC, SUB,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POP, PUSH, RL, RLC,</td>
<td>TCM, TM, XOR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RR, RRC, SRA, SWAP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JP, CALL (Indirect)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Two-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>OPC</th>
<th>MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>src</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>dst</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRCP</td>
<td>VALUE</td>
<td>VALUE</td>
<td></td>
</tr>
<tr>
<td>ADC, ADD, AND, CP,</td>
<td>ADC, ADD, AND, CP,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OR, SBC, SUB, TCM,</td>
<td>LD, OR, SBC, SUB,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TM, XOR</td>
<td>TCM, TM, XOR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD, LDE, LDEI,</td>
<td>LD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDC, LDCI</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Three-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>addr</td>
</tr>
<tr>
<td>JP</td>
<td></td>
</tr>
<tr>
<td>CALL</td>
<td></td>
</tr>
</tbody>
</table>

### INSTRUCTION SUMMARY

**Note:** Assignment of a value is indicated by the symbol "←". For example:

- `dst ← dst + src`

indicates that the source data is added to the destination data and the result is stored in the destination location. The notation "addr (n)" is used to refer to bit (n) of a given operand location. For example:

- `dst(7)`

refers to bit 7 of the destination operand.
## INSTRUCTION SUMMARY (Continued)

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode dst src</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected C Z S V D H</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ADC</strong> dst, src</td>
<td>↑</td>
<td>1[ ]</td>
<td>* * * * 0 *</td>
</tr>
<tr>
<td>dst←dst + src +C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ADD</strong> dst, src</td>
<td>↑</td>
<td>0[ ]</td>
<td>* * * * 0 *</td>
</tr>
<tr>
<td>dst←dst + src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AND</strong> dst, src</td>
<td>↑</td>
<td>5[ ]</td>
<td>* * 0 - -</td>
</tr>
<tr>
<td>dst←dst AND src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CALL</strong> dst</td>
<td>DA</td>
<td>D6</td>
<td>- - - - -</td>
</tr>
<tr>
<td>SP←SP – 2</td>
<td>IRR</td>
<td>D4</td>
<td></td>
</tr>
<tr>
<td>@SP←PC,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CCF</strong></td>
<td></td>
<td>EF</td>
<td>* - - - -</td>
</tr>
<tr>
<td>C←NOT C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CLR</strong> dst</td>
<td>R</td>
<td>B0</td>
<td>- - - - -</td>
</tr>
<tr>
<td>dst←0</td>
<td>IR</td>
<td>B1</td>
<td></td>
</tr>
<tr>
<td><strong>COM</strong> dst</td>
<td>R</td>
<td>60</td>
<td>- * 0 -</td>
</tr>
<tr>
<td>dst←NOT dst</td>
<td>IR</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td><strong>CP</strong> dst, src</td>
<td>↑</td>
<td>A[ ]</td>
<td>* * * * -</td>
</tr>
<tr>
<td>dst – src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DA</strong> dst</td>
<td>R</td>
<td>40</td>
<td>* * * X -</td>
</tr>
<tr>
<td>dst←-DA dst</td>
<td>IR</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td><strong>DEC</strong> dst</td>
<td>R</td>
<td>00</td>
<td>- * * - -</td>
</tr>
<tr>
<td>dst←-dst – 1</td>
<td>IR</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td><strong>DECW</strong> dst</td>
<td>RR</td>
<td>80</td>
<td>- * * * -</td>
</tr>
<tr>
<td>dst←-dst – 1</td>
<td>IR</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td><strong>DI</strong></td>
<td></td>
<td>8F</td>
<td>- - - - -</td>
</tr>
<tr>
<td>IMR(7)←-0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DJNZ</strong> dst, src</td>
<td>RA</td>
<td>rA</td>
<td>- - - - -</td>
</tr>
<tr>
<td>r←r – 1</td>
<td></td>
<td>r = 0 – F</td>
<td></td>
</tr>
<tr>
<td>if r ≠ 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←-PC + dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range: +127, -128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>EI</strong></td>
<td></td>
<td>9F</td>
<td>- - - - -</td>
</tr>
<tr>
<td>IMR(7)←-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HALT</strong></td>
<td></td>
<td>7F</td>
<td>- - - - -</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode dst src</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected C Z S V D H</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INC</strong> dst</td>
<td>r</td>
<td>rE</td>
<td>- * * * -</td>
</tr>
<tr>
<td>dst←dst + 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>INCW</strong> dst</td>
<td>RR</td>
<td>A0</td>
<td>- * * * -</td>
</tr>
<tr>
<td>dst←dst + 1</td>
<td>IR</td>
<td>A1</td>
<td></td>
</tr>
<tr>
<td><strong>IRET</strong></td>
<td></td>
<td>BF</td>
<td>* * * * *</td>
</tr>
<tr>
<td>FLAGS←@SP;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP←SP + 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←@SP;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP←SP + 2;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMR(7)←-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>JP</strong> cc, dst</td>
<td>DA</td>
<td>cD</td>
<td>- - - - -</td>
</tr>
<tr>
<td>if cc is true,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←dst</td>
<td>IRR</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td><strong>JR</strong> cc, dst</td>
<td>RA</td>
<td>cB</td>
<td>- - - - -</td>
</tr>
<tr>
<td>if cc is true,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←PC + dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range: +127, -128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LD</strong> dst, src</td>
<td>r</td>
<td>Im</td>
<td>rC</td>
</tr>
<tr>
<td>dst←src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LDI</strong> dst, src</td>
<td>Ir</td>
<td>Irr</td>
<td>C2</td>
</tr>
<tr>
<td>dst←src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LDCI</strong> dst, src</td>
<td>Ir</td>
<td>Irr</td>
<td>C3</td>
</tr>
<tr>
<td>dst←src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LDC</strong> dst, src</td>
<td>r</td>
<td>Im</td>
<td>C2</td>
</tr>
<tr>
<td>dst←src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NOP</strong></td>
<td></td>
<td>FF</td>
<td>- - - - -</td>
</tr>
</tbody>
</table>

5-30
<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
<th>C</th>
<th>Z</th>
<th>S</th>
<th>V</th>
<th>D</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR dst, src, dst ← dst OR src</td>
<td>t</td>
<td>4[ ]</td>
<td>- * * 0 - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POP dst, dst ← @SP; SP ← SP + 1</td>
<td>R</td>
<td>50</td>
<td>- - - - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PUS dst, SP ← SP - 1; @SP ← src</td>
<td>R</td>
<td>70</td>
<td>- - - - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCF</td>
<td>CF</td>
<td>0</td>
<td>- - - - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RET PC ← @SP; SP ← SP + 2</td>
<td>AF</td>
<td>- - - - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RL dst</td>
<td>R</td>
<td>90</td>
<td>* * * * - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RLC dst</td>
<td>R</td>
<td>10</td>
<td>* * * * - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RR dst</td>
<td>R</td>
<td>E0</td>
<td>* * * * - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RRC dst</td>
<td>R</td>
<td>C0</td>
<td>* * * * - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SBC dst, src, dst ← dst – src – C</td>
<td>t</td>
<td>3[ ]</td>
<td>* * * * 1 *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCF</td>
<td>DF</td>
<td>1</td>
<td>- - - - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRA dst</td>
<td>R</td>
<td>D0</td>
<td>* * * 0 - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRP dst</td>
<td>Im</td>
<td>31</td>
<td>- - - - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
<th>C</th>
<th>Z</th>
<th>S</th>
<th>V</th>
<th>D</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>STOP</td>
<td>6F</td>
<td>1</td>
<td>- - - - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUB dst, src, dst ← dst – src</td>
<td>t</td>
<td>2[ ]</td>
<td>* * * * 1 *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWAP dst</td>
<td>R</td>
<td>F0</td>
<td>X * * X - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCM dst, src</td>
<td>6[ ]</td>
<td>* * 0 - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TM dst, src</td>
<td>7[ ]</td>
<td>* * 0 - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WDH</td>
<td>4F</td>
<td>- - - - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WDT</td>
<td>5F</td>
<td>- X X X - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XOR dst, src</td>
<td>B[ ]</td>
<td>* * 0 - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

† These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a ‘[ ]’ in this table, and its value is found in the following table to the left of the applicable addressing mode pair.

For example, the opcode of an ADC instruction using the addressing modes r (destination) and Ir (source) is 13.

<table>
<thead>
<tr>
<th>Address Mode</th>
<th>Opcode Nibble</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>r</td>
<td>Ir</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>R</td>
<td>IR</td>
</tr>
<tr>
<td>R</td>
<td>IM</td>
</tr>
<tr>
<td>IR</td>
<td>IM</td>
</tr>
</tbody>
</table>

5-31
**OPCODE MAP**

<table>
<thead>
<tr>
<th>Upper Nibble (Hex)</th>
<th>Lower Nibble (Hex)</th>
<th>Bytes per instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>E</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>3</td>
</tr>
</tbody>
</table>

**Execution Cycles**

1. **Lower Nibble**
   - Mnemonic
   - First Operand
   - Second Operand

2. **Execution Cycles**
   - 10.5

3. **Pipeline Cycles**

**Legend:**
- **R** = 8-bit Address
- **r** = 4-bit Address
- R1 or r1 = Dat Address
- R2 or r2 = Src Address

**Sequence:**
- Opcode, First Operand, Second Operand

**Note:** Blank areas reserved.

*2-byte instruction appears as a 3-byte instruction*
### PACKAGE INFORMATION

#### 18-Pin DIP Package Diagram

#### 18-Pin SOIC Package Diagram

### SYMBOL | MILLIMETER | INCH
--- | --- | ---
A | 0.81 | 0.032 | 0.032
A2 | 3.43 | 0.135 | 0.135
B | 0.03 | 0.012 | 0.012
B1 | 1.65 | 0.065 | 0.065
C | 0.38 | 0.015 | 0.015
D | 23.37 | 0.920 | 0.920
E | 0.13 | 0.050 | 0.050
E1 | 6.48 | 0.255 | 0.255
F | 2.54 | 0.100 | 0.100
eA | 0.89 | 0.035 | 0.035
G1 | 1.65 | 0.065 | 0.065
S | 1.65 | 0.065 | 0.065

### CONTROLLING DIMENSIONS: INCH

#### 18-Pin DIP Package Diagram

### SYMBOL | MILLIMETER | INCH
--- | --- | ---
A | 0.81 | 0.032 | 0.032
A1 | 0.03 | 0.012 | 0.012
A2 | 3.43 | 0.135 | 0.135
B | 0.03 | 0.012 | 0.012
C | 0.38 | 0.015 | 0.015
D | 23.37 | 0.920 | 0.920
E | 0.13 | 0.050 | 0.050
E1 | 6.48 | 0.255 | 0.255
F | 2.54 | 0.100 | 0.100
eA | 0.89 | 0.035 | 0.035
G1 | 1.65 | 0.065 | 0.065
S | 1.65 | 0.065 | 0.065

### CONTROLLING DIMENSIONS: MM

#### 18-Pin SOIC Package Diagram

### SYMBOL | MILLIMETER | INCH
--- | --- | ---
A | 2.65 | .104 | .104
A1 | 0.30 | 0.012 | 0.012
A2 | 2.44 | 0.096 | 0.096
B | 0.46 | 0.018 | 0.018
C | 0.20 | 0.008 | 0.008
D | 1.15 | 0.045 | 0.045
E | 0.76 | 0.030 | 0.030
F | 1.67 | 0.065 | 0.065
H | 0.10 | 0.004 | 0.004
h | 0.10 | 0.004 | 0.004
L | 0.80 | 0.031 | 0.031
Q1 | 0.97 | 0.038 | 0.038
ORDERING INFORMATION

Z86C07 (8 MHz)

Standard Temperature

- 18-Pin DIP: Z86C0708PSC
- 18-Pin SOIC: Z86C0708SSC

Extended Temperature

- 18-Pin DIP: Z86C0708PEC
- 18-Pin SOIC: Z86C0708SEC

Z86C07 (12 MHz)

Standard Temperature

- 18-Pin DIP: Z86C0712PSC
- 18-Pin SOIC: Z86C0712SSC

Extended Temperature

- 18-Pin DIP: Z86C0712PEC
- 18-Pin SOIC: Z86C0712SEC

For fast results, contact your local Zilog sale offices for assistance in ordering the part desired.

CODES

Preferred Package
P = DIP

Longer Lead Time
S = SOIC

Preferred Temperature
S = 0°C to +70°C

Longer Lead Time
E = -40°C to +105°C

Speeds
08 = 8 MHz
12 = 12 MHz

Environmental
C = Plastic Standard

Example:

Z 86C07 08 P S C

is a Z86C07, 8 MHz, DIP, 0°C to +70°C, Plastic Standard Flow

Environmental Flow
Temperature
Package
Speed
Product Number
Zilog Prefix
Z86C03/C06 CMOS Z8® 8-Bit CCP™ Consumer Controller Processors

Z86E03/E06 CMOS Z8® 8-Bit OTP CCP™ Consumer Controller Processors

Z86C04/C08 CMOS Z8® Low Cost 1K /2K ROM Microcontrollers

Z86E04/E08 CMOS Z8® 8-Bit OTP Microcontrollers

Z86C07 CMOS Z8® 8-Bit Microcontroller

Z86E07 CMOS Z8® 8-Bit OTP Microcontroller

Z86C30/C31 CMOS Z8® 8-Bit CCP™ Consumer Controller Processors
FEATURES

- Low Cost, 8-Bit CMOS MCU (OTP Support for Z86C07)
- 18-Pin Package (DIP, SOIC)
- 2 Kbytes of One-Time-PROM
- 124 Bytes of General-Purpose RAM
- 4.0V to 5.5V Operating Range
- Clock Speed: 12 MHz
- Low Power Consumption: 50 mW (Typical)
- Low Noise Programmable
- Programmable ROM Protect
- Two Programmable 8-Bit Counter/Timers Each with a 6-Bit Programmable Prescaler.
- Fast Instruction Pointer: 1 µs @ 12 MHz
- Two Standby Modes: STOP and HALT
- 14 Input/Output Lines
- 11 Digital Inputs, CMOS Levels, Schmitt-Triggered.
- Six Vectored, Priority Interrupts from Six Different Sources.
- Programmable Interrupt Polarity
- Software-Programmable Watch-Dog Timer
- Power-On Reset Timer
- Two On-Board Comparators
- On-Chip Oscillator that Accepts a Crystal, Ceramic Resonator, LC, or External Clock Drive.

GENERAL DESCRIPTION

The Z86E07 8-bit One-Time-Programmable (OTP) Microcontroller (MCU) is a member of the Z8® single-chip microcontroller family with 2 Kbytes of one-time PROM and 124 Bytes of General-Purpose RAM. The device is housed in an 18-pin DIP or SOIC style package and is manufactured in CMOS technology. The Z86E07 allows easy software development and debug, prototyping, and is ideal for small production runs not economically desirable with a masked ROM version.

For applications demanding powerful I/O capabilities, the Z86E07 provides 14 pins dedicated to input and output. These lines are grouped into three ports, and are configurable under software control to provide I/O, timing, and status signals. There are two basic address spaces available to support this configuration: program memory and 124 bytes of general-purpose registers.

The Z86E07 offers programmable EPROM Protect and programmable Low Noise. When the part is programmed for EPROM Protect, the Low Noise feature will automatically be enabled. When programmed for Low Noise, the EPROM Protect feature is optional.

With a flexible I/O scheme, an efficient register and address space structure, and a number of ancillary features, the Z86E07 is well-suited for a variety of consumer, industrial and commercial applications.
To unburden the system from coping with real-time tasks, such as counting/timing and I/O data communications, the Z86E07 offers two on-chip counter/timers with a large number of user selectable modes. The device also features two on-board comparators that process analog signals with a common reference voltage (Figures 1 and 2).

Notes:
All Signals with a preceding front slash, "/", are active Low, e.g.: B/W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

<table>
<thead>
<tr>
<th>Connection</th>
<th>Circuit</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>Vcc</td>
<td>Vcc</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
<td>Vss</td>
</tr>
</tbody>
</table>

Figure 1. Z86E07 Functional Block Diagram
Figure 2. Z86E07 EPROM Mode Block Diagram

PIN DESCRIPTION

Table 1. Z86E07 18-Pin DIP Pin Identification

<table>
<thead>
<tr>
<th>EPROM Mode</th>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin #</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-4</td>
<td>1-4</td>
<td>D7-D4</td>
<td>Data 4, 5, 6, 7</td>
<td>In/Output</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Vcc</td>
<td>Power Supply</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>N/C</td>
<td>No Connection</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>/CE</td>
<td>Chip Enable</td>
<td>Input</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>/OE</td>
<td>Output Enable</td>
<td>Input</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>EPM</td>
<td>EPROM Prog Mode</td>
<td>Input</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>Vpp</td>
<td>Prog Voltage</td>
<td>Input</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>Clear</td>
<td>Clear Clock</td>
<td>Input</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>Clock</td>
<td>Address</td>
<td>Input</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>/PGM</td>
<td>Prog Mode</td>
<td>Input</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>15-18</td>
<td>15-18</td>
<td>D3-D0</td>
<td>Data 0,1, 2, 3</td>
<td>In/Output</td>
</tr>
</tbody>
</table>

Figure 3. Z86E07 18-Pin DIP Pin Configuration

EPROM Mode
PIN DESCRIPTION

Table 2. Z86E07 18-Pin DIP Pin Identification

<table>
<thead>
<tr>
<th>Standard Mode</th>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-4</td>
<td>P27-P24</td>
<td>Port 2, Pins 4, 5, 6, 7</td>
<td>In/Output</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Vcc</td>
<td>Power Supply</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>XTAL2</td>
<td>Crystal Osc. Clock Output</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>XTAL1</td>
<td>Crystal Osc. Clock Input</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>P31</td>
<td>Port 3, Pin 1</td>
<td>Input</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>P32</td>
<td>Port 3, Pin 2</td>
<td>Input</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>P33</td>
<td>Port 3, Pin 3</td>
<td>Input</td>
</tr>
<tr>
<td></td>
<td>11-13</td>
<td>P02-P00</td>
<td>Port 0, Pins 0, 1, 2</td>
<td>In/Output</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>15-18</td>
<td></td>
<td>P23-P20</td>
<td>Port 2, Pins 0, 1, 2, 3</td>
<td>In/Output</td>
</tr>
</tbody>
</table>

Note:
* Pin Identification and Configuration identical on DIP and SOIC style packages.

PIN FUNCTIONS

OTP Programming Mode

D7-D0 Data Bus. The data can be read from, or written to the EPROM through this data bus.

Vcc Power Supply. It is 5V during the EPROM Read mode and 6V during the other mode.

/CE Chip Enable (active Low). This pin is active during EPROM Read Mode, Program Mode, and Program Verify Mode (Vil, ViH, Vih = 12V ±0.5V).

/OE Output Enable (active Low). This pin drives the Data Bus direction. When this pin is Low, the Data Bus is output. When High, the Data Bus is input.

EPM EPROM Program Mode. This pin controls the different EPROM Program Modes by applying different voltages (Vil, Vih, Vih = 12V ±0.5V).

Vpp Program Voltage. This pin supplies the program voltage (Vih = 12V ±0.5V).

Clock Address Clock. This pin is a clock input. The internal address counter increases by one with one clock signal.

/PGM Program Mode (active Low). Low Level at this pin programs the data to the EPROM through the Data Bus.

Application Precaution

The production test-mode environment may be enabled accidentally during normal operation if excessive noise surges above Vcc occur on the XTAL1 pin.

In addition, processor operation of Z8 One-Time Programmable devices may be affected by excessive noise surges on the Vpp, /CE, /EPM, /OE pins while the microcontroller is in standard mode.

Recommendations for dampening voltage surges in both test and OTP mode include the following:

- Using a clamping diode to Vcc.
- Adding a capacitor to the affected pin.
**Z86E07 Standard Mode**

**XTAL1, XTAL2** *Crystal In, Crystal Out* (time-based input and output, respectively). These pins connect a parallel-resonant crystal, LC, or an external single-phase clock (12 MHz max) to the on-chip clock oscillator and buffer.

**Port 0, PO2-PO0.** Port 0 is a 3-bit bi-directional, Schmitt-triggered CMOS compatible I/O port. These three I/O lines can be globally configured under software control to be inputs or outputs (Figure 5).

![Schema](image)

**Figure 5. Port 0 Configuration**
Port 2, P27-P20. Port 2 is an 8-bit, bit programmable, bi-directional, Schmitt-triggered CMOS compatible I/O port. These eight I/O lines can be configured under software control to be inputs or outputs, independently. Bits programmed as outputs can be globally programmed as either push-pull or open-drain (Figure 6).

Figure 6. Port 2 Configuration
Port 3, P33-P31. Port 3 is a 3-bit, CMOS compatible port with three fixed input (P32-P30) lines. These three input lines can be configured under software control as digital inputs or analog inputs. These three input lines are also used as the interrupt sources IRQ0-IRQ3 and as the timer input signal $T_{in}$ (Figure 7).

Figure 7. Port 3 Configuration
Z86E07 Standard Mode (Continued)

Comparator Inputs. Two analog comparators are added to input of Port 3, P31 and P32, for interface flexibility. The comparators reference voltage P3 (REF) is common to both comparators.

Typical applications for the on-board comparators; Zero crossing detection, A/D conversion, voltage scaling, and threshold detection. In analog mode, P33 input functions serve as a reference voltage to the comparators.

The dual comparator (common inverting terminal) features a single power supply which discontinues power in STOP mode. The common voltage range is 0-4 V when the Vcc is 5.0 V; the power supply and common mode rejection ratios are 90 dB and 60 dB, respectively. Interrupts are generated on either edge of Comparator 2's output, or on the falling edge of Comparator 1's output. The comparator output is used for interrupt generation, Port 3 data inputs, or T1N through P31. Alternatively, the comparators can be disabled, freeing the reference input (P33) for use as IRQ1 and/or P33 input.

FUNCTIONAL DESCRIPTION

RESET is accomplished through Power-On or a Watch-Dog Timer Reset. Upon power-up, the Power-On Reset circuit waits for T_{POR} plus 18 clock cycles, then starts program execution at address 000C (Hex). Reference Table 3 for the Z86E07 control registers' reset values (Figure 8).

Power-On Reset (POR). A timer circuit, clocked by a dedicated on-board RC oscillator, is used for a POR timer function. The POR time allows Vcc and the oscillator circuit to stabilize before instruction execution begins. The POR timer circuit is a one-shot timer triggered by one of the four following conditions:

- Power bad to power good status
- STOP-Mode Recovery
- WDT time-out
- WDH time-out

Watch-Dog Timer Reset. The WDT is a retriggerable one-shot timer that resets the Z8 if it reaches its terminal count. The WDT is initially enabled by executing the WDT instruction and is retriggered on subsequent execution of the WDT instruction. The timer circuit is driven by an on-board RC oscillator.
Table 2. Z86C07 Control Registers

<table>
<thead>
<tr>
<th>Addr</th>
<th>Reg.</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>TMR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Timers Off</td>
</tr>
<tr>
<td>F2</td>
<td>T1</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>F3</td>
<td>PRE1</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>F4</td>
<td>T0</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>F5</td>
<td>PRE0</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>F6*</td>
<td>P2M</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Inputs after reset.</td>
</tr>
<tr>
<td>F7*</td>
<td>P3M</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td>Standard Port 3 inputs.</td>
</tr>
<tr>
<td>F8*</td>
<td>P01M</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>F9</td>
<td>IPR</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>FA</td>
<td>IRQ</td>
<td>U</td>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>IRQ3 is used for positive edge detection.</td>
</tr>
<tr>
<td>PB</td>
<td>IMR</td>
<td>0</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td>FLAGS</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>FD</td>
<td>RP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>FF</td>
<td>SPL</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

* Registers are not reset after a STOP-Mode Recovery using P27 pin. A subsequent reset will cause these control registers to be reconfigured as shown in Table 3 and the user must avoid bus contention on the port pins or it may affect device reliability.

Program Memory. The Z86E07 addresses up to 2 Kbytes of internal program memory (Figure 9). The first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts. Bytes 0-2047 are on-chip one-time programmable ROM.
**FUNCTIONAL DESCRIPTION** (Continued)

**Register File.** The Register File consists of three I/O port registers, 124 general-purpose registers, and 14 control and status registers, RO-R3, R4-R127 and R241-R255, respectively (Figure 10). General-purpose registers occupy the 04H to 7FH address space. I/O ports are mapped as per the existing CMOS Z8. The Mode and Configuration Registers are the same as the Z86C07. The Z86E07 instructions can access registers directly or indirectly through an 8-bit address field. This allows short, 4-bit register addressing using the Register Pointer. In the 4-bit mode, the register file is divided into eight working register groups, each occupying 16 continuous locations. The Register Pointer (Figure 11) addresses the starting location of the active working-register group.

**Stack Pointer.** The Z86E07 has an 8-bit Stack Pointer (R255) used for the internal stack that resides within the 124 general-purpose registers.

---

**Figure 10. Register File**

**Figure 11. Register Pointer**
General-Purpose Registers (GPR). These registers are undefined after the device is powered up. The registers keep their last value after any reset, as long as the reset occurs in the $V_{CC}$ voltage-specified operating range. **Note:** Register R254 has been designated as a general-purpose register.

Counter/Timer. There are two 8-bit programmable counter/timers (T0 and T1), each driven by its own 6-bit programmable prescaler. The T1 prescaler is driven by internal or external clock sources; however, the T0 can be driven by the internal clock source only (Figure 12).

The 6-bit prescalers divide the input frequency of the clock source by any integer number from 1 to 64. Each prescaler drives its counter, which decrements the value (1 to 256) that has been loaded into the counter. When both counter and prescaler reach the end of count, a timer interrupt request IRQ4 (T0) or IRQ5 (T1) is generated.

The counter can be programmed to start, stop, restart to continue, or restart from the initial value. The counters are also programmed to stop upon reaching zero (single pass mode) or to automatically reload the initial value and continue counting (modulo-n continuous mode).

The counters, but not the prescalers, are read at any time without disturbing their value or count mode. The clock source for T1 is user-definable and is either the internal microprocessor clock divided by four, or an external signal input through Port 3. The Timer Mode register configures the external timer input (P30) as an external clock, a trigger input that is retriggerable or not retriggerable, or used as a gate input for the internal clock.

![Figure 12. Counter/Timers Block Diagram](image-url)
Interrupts. The Z86E07 has six interrupts from five different sources. These interrupts are maskable and prioritized (Figure 13). The sources are divided as follows: the falling edge of P31 (AN1), P32 (AN2), P33 (REF), the rising edge of P32 (AN2), and two counters/timers. The Interrupt Mask Register globally or individually enables or disables the six interrupt requests (Table 4).

When more than one interrupt is pending, priorities are resolved by a programmable priority encoder that is controlled by the Interrupt Priority register. All Z86E07 interrupts are vectored through locations in program memory. When an Interrupt machine cycle is activated, an Interrupt Request is granted. This disables all subsequent interrupts, saves the Program Counter and Status Flags, and then branches to the program memory vector location reserved for that interrupt. This memory location and the next byte contain the 16-bit starting address of the interrupt service routine for that particular interrupt request.

To accommodate polled interrupt systems, interrupt inputs are masked and the interrupt request register is polled to determine which of the interrupt requests needs service.

Note: User must select any Z86C08 mode in Zilog’s C12 ICEBOX™ emulator. The rising edge interrupt is not supported on the Z86CCP00ZEM emulator.

### Table 4. Interrupt Types, Sources, and Vectors

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Vector Location</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRQ0</td>
<td>AN2(P32)</td>
<td>0, 1</td>
<td>External (F)Edge</td>
</tr>
<tr>
<td>IRQ1</td>
<td>REF(P33)</td>
<td>2, 3</td>
<td>External (F)Edge</td>
</tr>
<tr>
<td>IRQ2</td>
<td>AN1(P31)</td>
<td>4, 5</td>
<td>External (F)Edge</td>
</tr>
<tr>
<td>IRQ3</td>
<td>AN2(P32)</td>
<td>6, 7</td>
<td>External (R)Edge</td>
</tr>
<tr>
<td>IRQ4</td>
<td>T0</td>
<td>8, 9</td>
<td>Internal</td>
</tr>
<tr>
<td>IRQ5</td>
<td>T1</td>
<td>10, 11</td>
<td>Internal</td>
</tr>
</tbody>
</table>

**Notes:**

F = Falling edge triggered
R = Rising edge triggered

**Figure 13. Interrupt Block Diagram**
Clock. The Z86E07 on-chip oscillator has a high-gain, parallel-resonant amplifier for connection to a crystal, ceramic resonator, or any suitable external clock source (XTAL1 = INPUT, XTAL2 = OUTPUT). The crystal should be AT cut, 12 MHz max, with a series resistance (RS) of less than or equal to 100 Ohms.

The crystal should be connected across XTAL1 and XTAL2 using the vendors crystal recommended capacitors (capacitance values depend upon the crystal manufacturer, ceramic resonator and PCB layout) from each pin directly to device ground pin 14 (Figure 14). Note that the crystal capacitor loads should be connected to Vss, Pin 14 to reduce Ground noise injection.

HALT Mode. This instruction turns off the internal CPU clock, but not the crystal oscillation. The counter/timers and external interrupts IRQ0, IRQ1, IRQ2 and IRQ3 remain active. The device is recovered by interrupts, either externally or internally generated. An interrupt request must be executed (enabled) to exit HALT mode. After the interrupt service routine, the program continues from the instruction after the HALT.

STOP Mode. This instruction turns off the internal clock and external crystal oscillation and reduces the standby current to 10 µA. The STOP mode is released by a RESET through a STOP-Mode Recovery (pin P27). A Low input condition on P27 releases the STOP mode. Program execution begins at location 000C(Hex). However, when P27 is used to release the STOP mode, the I/O port mode registers are not reconfigured to their default power-on conditions. This prevents any I/O, configured as output when the STOP instruction was executed, from glitching to an unknown state. To use the P27 release approach with STOP mode, use the following instruction:

LD P2M, #1XXX XXXXB
NOP
STOP

X = Dependent on user's application.

Watch-Dog Timer (WDT). The Watch-Dog Timer is enabled by instruction WDT. When the WDT is enabled, it cannot be stopped by the instruction. With the WDT instruction, the WDT is refreshed when it is enabled within every Twdt period; otherwise, the Z86E07 resets itself. The WDT instruction affects the flags accordingly; Z=1, S=0, V=0.

WDT = SF (Hex)

Opcode WDT (5FH). The first time opcode 5FH is executed, the WDT is enabled and subsequent execution clears the WDT counter. This must be done at least every Twdt period; otherwise, the WDT times out and generates a reset. The generated reset is the same as a Power-On Reset of Tpor, plus 18 XTAL clock cycles. The WDT does not work in STOP Mode.
FUNCTIONAL DESCRIPTION (Continued)

Opcode WDH (4FH). When this instruction is executed it enables the WDT during HALT. If not, the WDT stops when entering HALT. This instruction does not clear the counters, it just makes it possible to have the WDT running during HALT mode. A WDH instruction executed without executing WDT (5FH) has no effect.

Auto Reset Voltage ($V_{\text{RST}}$). The Z86E07 has an auto-reset built-in. The auto-reset circuit resets the Z86E07 when it detects the $V_{\text{CC}}$ below $V_{\text{RST}}$. Figure 15 shows the Auto Reset Voltage vs temperature. The Z86E07 does not function from $V_{\text{RST}}$ to below 4.5V. Upon power-up of the device, the $V_{\text{CC}}$ rise time must reach 4.5V before the $T_{\text{POR}}$ expires so that program execution begins with the $V_{\text{CC}}$ in the range 4.5V to 5.5V.

If the $V_{\text{CC}}$ drops below 4.5V while the device is in operation, the device must be powered down and then re-powered up again.

![Figure 15. Typical Auto Reset Voltage ($V_{\text{RST}}$) vs Temperature](image)
Low EMI Emission

The Z86E07 can be programmed to operate in a low EMI emission mode by means of an EPROM programmable bit option. Use of this feature results in:

- Less than 1 mA consumed during HALT mode.
- All drivers slew rates reduced to 10 ns (typical).
- Internal SCLK/TCLK = XTAL operation limited to a maximum of 4 MHz - 250 ns cycle time.
- Output drivers have resistances of 200 ohms (typical).
- Oscillator divide-by-two circuitry eliminated.

The Z86E07 offers programmable ROM Protect and programmable Low Noise features. When programmed for Low Noise, the ROM Protect feature is optional.

Besides $V_{DD}$ and GND ($V_{SS}$), the Z86E07 changes all its pin functions in the EPROM mode. XTAL2 has no function, XTAL1 functions as /CE, P31 functions as /OE, P32 functions as EPM, P33 functions as $V_{PP}$, and P02 functions as /PGM.

**EPROM Protect.** ROM Protect fully protects the Z86E07 ROM code from being read externally. When ROM Protect is selected, the Z86E07 will disable the instructions LDC and LDCI (Z86E07 and Z86C08 do not support the instructions of LDE and LDEI). When the device is programmed for ROM Protect, the Low Noise feature will automatically be enabled. A ROM look-up table cannot be used when EPROM Protect is selected.

Please note that when using the device in a noisy environment, it is suggested that the voltages on the EPM, /CE, /OE pins be clamped to $V_{CC}$ through a diode to $V_{CC}$ to prevent accidentally entering the OTP mode. The $V_{PP}$ requires both a diode and a 100 pF capacitor.

**User Modes.** Table 5 shows the programming voltage of each mode of Z86E07.

<table>
<thead>
<tr>
<th>Programming Modes</th>
<th>$V_{PP}$</th>
<th>EPM</th>
<th>/CE</th>
<th>/OE</th>
<th>/PGM</th>
<th>ADDR</th>
<th>DATA</th>
<th>$V_{CC}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPROM READ1</td>
<td>X</td>
<td>$V_H$</td>
<td>$V_{IL}$</td>
<td>$V_{IL}$</td>
<td>$V_{IH}$</td>
<td>ADDR</td>
<td>Out</td>
<td>4.5V</td>
</tr>
<tr>
<td>EPROM READ2</td>
<td>X</td>
<td>$V_H$</td>
<td>$V_{IL}$</td>
<td>$V_{IL}$</td>
<td>$V_{IH}$</td>
<td>ADDR</td>
<td>Out</td>
<td>5.5V</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>$V_H$</td>
<td>X</td>
<td>$V_{IL}$</td>
<td>$V_{IL}$</td>
<td>$V_{IH}$</td>
<td>ADDR</td>
<td>In</td>
<td>6.0V</td>
</tr>
<tr>
<td>PROGRAM VERIFY</td>
<td>$V_H$</td>
<td>X</td>
<td>$V_{IL}$</td>
<td>$V_{IL}$</td>
<td>$V_{IH}$</td>
<td>ADDR</td>
<td>Out</td>
<td>6.0V</td>
</tr>
<tr>
<td>EPROM PROTECT</td>
<td>$V_H$</td>
<td>$V_{IH}$</td>
<td>$V_{IH}$</td>
<td>$V_{IL}$</td>
<td>NU</td>
<td>NU</td>
<td>6.0V</td>
<td></td>
</tr>
<tr>
<td>LOW NOISE SELECT</td>
<td>$V_H$</td>
<td>$V_{IH}$</td>
<td>$V_{IH}$</td>
<td>$V_{IL}$</td>
<td>NU</td>
<td>NU</td>
<td>6.0V</td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**

$V_{H}$ = 12.5V ±0.5V

$V_{IL}$ = As per specific Z8 DC specification.

$X$ = Not used, but must be set to $V_{ss}$, $V_{mm}$, or $V_{cc}$ level.

NU = Not used, but must be set to either $V_{ss}$ or $V_{cc}$ level.

$I_{PP}$ during programming = 40 mA maximum.

$I_{PP}$ during programming, verify, or read = 40 mA maximum.

$V_{PP}$ has a tolerance of ±0.25V.

6-15
SPECIAL FUNCTIONS

Internal Address Counter. The address of Z86E07 is generated internally with a counter clocked through pin P01 (Clock). Each clock signal increases the address by one and the 'high' level of pin P00 (Clear) will reset the address to zero. Figure 16 shows the set-up time of the serial address input.

Programming Waveform. Figures 17, 18 and 19 show the programming waveforms of each mode. Table 6 shows the timing of programming waveforms.

Programming Algorithm. Figure 20 shows the flow chart of the Z86E07 programming algorithm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Address Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>2</td>
<td>Data Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>3</td>
<td>V&lt;sub&gt;pp&lt;/sub&gt; Setup</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>4</td>
<td>V&lt;sub&gt;cc&lt;/sub&gt; Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>5</td>
<td>Chip Enable Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>6</td>
<td>Program Pulse Width</td>
<td>0.95</td>
<td></td>
<td>ms</td>
</tr>
<tr>
<td>7</td>
<td>Data Hold Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>8</td>
<td>/OE Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>9</td>
<td>Data Access Time</td>
<td>200</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>10</td>
<td>Data Output Float Time</td>
<td>100</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>11</td>
<td>Overprogram Pulse Width</td>
<td>2.85</td>
<td></td>
<td>ms</td>
</tr>
<tr>
<td>12</td>
<td>EPM Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>13</td>
<td>/PGM Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>14</td>
<td>Address to /OE Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>15</td>
<td>Option Program Pulse Width</td>
<td>78</td>
<td></td>
<td>ms</td>
</tr>
</tbody>
</table>
Figure 16. Z86E07 Address Counter Waveform

Legend:

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset Clock Width</td>
<td>Input Clock High</td>
<td>Input Clock Period</td>
<td>Input Clock Low</td>
<td>Clock to Address Counter Out Delay</td>
</tr>
<tr>
<td>30 ns Min</td>
<td>30 ns Min</td>
<td>70 ns Min</td>
<td>30 ns Min</td>
<td>15 ns Max</td>
</tr>
</tbody>
</table>
Figure 17. Z86E07 Programming Waveform (EPROM Read)
Figure 18. Z86E07 Programming Waveform (Program and Verify)
Z86E07 CMOS 8-Bit OTP MICROCONTROLLER

SPECIAL FUNCTIONS (Continued)

Address

Vih

Vil

Data

Vih

Vil

Vpp

Vh

Vih

6V

4.5V

Vcc

/CE

Vh

Vih

/CE

Vh

Vih

/OE

Vh

Vih

EPM

VIH

VIL

VIH

VIH

VIH

VIH

Vth

/PGM

VIH

VIL

Figure 19. Z86E07 Programming Waveform
(EPROM Protect and Low EMI Program)
Note:
* To ensure proper operations during the spec., Zilog recommends verification over the Vcc range of the device Vcc spec.

Figure 20. Z86E07 Programming Algorithm
ABSOLUTE MAXIMUM RATINGS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient Temperature under Bias</td>
<td>-40</td>
<td>+105</td>
<td>C</td>
</tr>
<tr>
<td>Storage Temperature</td>
<td>-65</td>
<td>+150</td>
<td>C</td>
</tr>
<tr>
<td>Voltage on any Pin with Respect to Vss [Note 1]</td>
<td>-0.6</td>
<td>+12</td>
<td>V</td>
</tr>
<tr>
<td>Voltage on VDD Pin with Respect to Vss</td>
<td>-0.3</td>
<td>+7</td>
<td>V</td>
</tr>
<tr>
<td>Voltage on Pin 7, 8, 9, 10 with Respect to Vss [Note 2]</td>
<td>-0.6</td>
<td>VDD+1</td>
<td>V</td>
</tr>
<tr>
<td>Total Power Dissipation</td>
<td>462</td>
<td></td>
<td>mW</td>
</tr>
<tr>
<td>Maximum Current out of Vss</td>
<td>84</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Maximum Current into VDD</td>
<td>84</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Maximum Current into an Input Pin [Note 3]</td>
<td>-600</td>
<td>+600</td>
<td>µA</td>
</tr>
<tr>
<td>Maximum Current into an Open-Drain Pin [Note 4]</td>
<td>-600</td>
<td>+600</td>
<td>µA</td>
</tr>
<tr>
<td>Maximum Output Current Sunked by Any I/O Pin</td>
<td>12</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Maximum Output Current Sourced by Any I/O Pin</td>
<td>12</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Total Maximum Output Current Sunked by Port 2</td>
<td>70</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Total Maximum Output Current Sourced by Port 2</td>
<td>70</td>
<td></td>
<td>mA</td>
</tr>
</tbody>
</table>

Notice:

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for an extended period may affect device reliability. Total power dissipation should not exceed 462 mW for the package.

Power dissipation is calculated as follows:

Total Power dissipation = VDD x [I DD - (sum of I OH)] + sum of [(VDD - VOH) x I OH] + sum of (VOL x IOL)

STANDARD TEST CONDITIONS

The characteristics listed below apply for standard test conditions as noted. All voltages are referenced to Ground. Positive current flows into the referenced pin (Figure 21).

From Output Under Test

CAPACITANCE

T A = 25°C, VCC = GND = 0V, f = 1.0 MHz, unmeasured pins returned to GND.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input capacitance</td>
<td>0</td>
<td>10 pF</td>
</tr>
<tr>
<td>Output capacitance</td>
<td>0</td>
<td>20 pF</td>
</tr>
<tr>
<td>I/O capacitance</td>
<td>0</td>
<td>25 pF</td>
</tr>
</tbody>
</table>
### DC Electrical Characteristics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter Description</th>
<th>$V_{CC}$[3]</th>
<th>$T_A = 0°C$ to $+70°C$</th>
<th>Typical [11] @ $25°C$</th>
<th>Units</th>
<th>Conditions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{CH}$</td>
<td>Clock Input High Voltage</td>
<td>4.5V</td>
<td>0.8 $V_{CC}$</td>
<td>$V_{CC} + 0.3$</td>
<td>2.4</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.8 $V_{CC}$</td>
<td>$V_{CC} + 0.3$</td>
<td>2.6</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
</tr>
<tr>
<td>$V_{CL}$</td>
<td>Clock Input Low Voltage</td>
<td>4.5V</td>
<td>$V_{SS} - 0.3$</td>
<td>0.2 $V_{CC}$</td>
<td>1.6</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{SS} - 0.3$</td>
<td>0.2 $V_{CC}$</td>
<td>2.3</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
</tr>
<tr>
<td>$V_{IH}$</td>
<td>Input High Voltage</td>
<td>4.5V</td>
<td>0.7 $V_{CC}$</td>
<td>$V_{CC} + 0.3$</td>
<td>2.1</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.7 $V_{CC}$</td>
<td>$V_{CC} + 0.3$</td>
<td>2.7</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>$V_{IL}$</td>
<td>Input Low Voltage</td>
<td>4.5V</td>
<td>$V_{SS} - 0.3$</td>
<td>0.2 $V_{CC}$</td>
<td>1.2</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{SS} - 0.3$</td>
<td>0.2 $V_{CC}$</td>
<td>1.7</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>$V_{OH}$</td>
<td>Output High Voltage</td>
<td>4.5V</td>
<td>$V_{CC} - 0.4$</td>
<td>3.9</td>
<td>V</td>
<td>$I_{OH} = -2.0 mA$</td>
<td>[9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{CC} - 0.4$</td>
<td>5.4</td>
<td>V</td>
<td>$I_{OH} = -2.0 mA$</td>
<td>[9]</td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>Output Low Voltage</td>
<td>4.5V</td>
<td>0.4</td>
<td>V</td>
<td>Low Noise @ $I_{OL} = 0.5 mA$</td>
<td>[10]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.4</td>
<td>V</td>
<td>Low Noise @ $I_{OL} = 0.5 mA$</td>
<td>[10]</td>
<td></td>
</tr>
<tr>
<td>$V_{OL1}$</td>
<td>Output Low Voltage</td>
<td>4.5V</td>
<td>0.4</td>
<td>0.2</td>
<td>V</td>
<td>$I_{OL} = +4.0 mA$</td>
<td>[9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.4</td>
<td>0.2</td>
<td>V</td>
<td>$I_{OL} = +4.0 mA$</td>
<td>[9]</td>
</tr>
<tr>
<td>$V_{OL2}$</td>
<td>Output Low Voltage</td>
<td>4.5V</td>
<td>1.0</td>
<td>0.7</td>
<td>V</td>
<td>$I_{OL} = +12 mA$, 3 Pin Max</td>
<td>[9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.8</td>
<td>0.5</td>
<td>V</td>
<td>$I_{OL} = +12 mA$, 3 Pin Max</td>
<td>[9]</td>
</tr>
<tr>
<td>$V_{OFFSET}$</td>
<td>Comparator Input Offset Voltage</td>
<td>4.5V</td>
<td>25</td>
<td>6</td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>7</td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{RST}$</td>
<td>Auto Reset Voltage</td>
<td>1.55</td>
<td>2.7</td>
<td>2.4</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{IL}$</td>
<td>Input Leakage (Input Bias Current of Comparator)</td>
<td>4.5V</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>$\mu A$</td>
<td>$0V &lt; V_{IN} &lt; V_{CC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>$\mu A$</td>
<td>$0V &lt; V_{IN} &lt; V_{CC}$</td>
</tr>
<tr>
<td>$I_{OL}$</td>
<td>Output Leakage</td>
<td>4.5V</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>$\mu A$</td>
<td>$V_{IN} = 0V$, $V_{CC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>$\mu A$</td>
<td>$V_{IN} = 0V$, $V_{CC}$</td>
</tr>
<tr>
<td>$V_{CM}$</td>
<td>Input Common Mode Voltage Range</td>
<td>0</td>
<td>$V_{CC} - 1.0$</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## DC ELECTRICAL CHARACTERISTICS (Continued)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$</th>
<th>$T_a = 0^\circ C$ to $+70^\circ C$</th>
<th>Typical[11] @ $25^\circ C$</th>
<th>Units</th>
<th>Conditions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{cc}$</td>
<td>Supply Current</td>
<td></td>
<td>Min</td>
<td>Max</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 2 MHz</td>
<td>[4,5,8,9]</td>
</tr>
<tr>
<td></td>
<td>(Standard Mode)</td>
<td>4.5V</td>
<td>4.0</td>
<td>2.2</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 2 MHz</td>
<td>[4,5,8,9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>7.0</td>
<td>5.0</td>
<td>mA</td>
<td>Floating @ 2 MHz</td>
<td>[8,9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.5V</td>
<td>9.0</td>
<td>4.5</td>
<td>mA</td>
<td>Floating @ 8 MHz</td>
<td>[8,9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>11.0</td>
<td>8.3</td>
<td>mA</td>
<td>Floating @ 8 MHz</td>
<td>[8,9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.5V</td>
<td>10</td>
<td>6.1</td>
<td>mA</td>
<td>Floating @ 12 MHz</td>
<td>[8,9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>15</td>
<td>10.8</td>
<td>mA</td>
<td>Floating @ 12 MHz</td>
<td>[8,9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>11</td>
<td>10.2</td>
<td>mA</td>
<td>Floating @ 12 MHz</td>
<td>[8,9]</td>
</tr>
<tr>
<td>$I_{cc1}$</td>
<td>Standby Current</td>
<td></td>
<td>Min</td>
<td>Max</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 2 MHz</td>
<td>[4,5,8,9]</td>
</tr>
<tr>
<td></td>
<td>(Standard Mode)</td>
<td>4.5V</td>
<td>2.5</td>
<td>0.5</td>
<td>mA</td>
<td>HALT mode $V_{in} = 0V$, $V_{cc} = 2 MHz$</td>
<td>[4,5,8,9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>4.0</td>
<td>1.0</td>
<td>mA</td>
<td>HALT mode $V_{in} = 0V$, $V_{cc} = 2 MHz$</td>
<td>[8,9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.5V</td>
<td>4.0</td>
<td>1.0</td>
<td>mA</td>
<td>HALT mode $V_{in} = 0V$, $V_{cc} = 8 MHz$</td>
<td>[8,9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>5.0</td>
<td>2.0</td>
<td>mA</td>
<td>HALT mode $V_{in} = 0V$, $V_{cc} = 8 MHz$</td>
<td>[8,9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.5V</td>
<td>5.0</td>
<td>1.3</td>
<td>mA</td>
<td>HALT mode $V_{in} = 0V$, $V_{cc} = 12 MHz$</td>
<td>[8,9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>7.0</td>
<td>2.3</td>
<td>mA</td>
<td>HALT mode $V_{in} = 0V$, $V_{cc} = 12 MHz$</td>
<td>[8,9]</td>
</tr>
<tr>
<td>$I_{cc}$</td>
<td>Supply Current</td>
<td></td>
<td>Min</td>
<td>Max</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 1 MHz</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td></td>
<td>(Low Noise Mode)</td>
<td>4.5V</td>
<td>4.0</td>
<td>2.2</td>
<td>mA</td>
<td>Floating @ 2 MHz</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>7.0</td>
<td>4.2</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 2 MHz</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.5V</td>
<td>6.0</td>
<td>2.9</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 4 MHz</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>9.0</td>
<td>5.5</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 4 MHz</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.5V</td>
<td>8.0</td>
<td>4.4</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 4 MHz</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>11.0</td>
<td>7.9</td>
<td>mA</td>
<td>All Output and I/O Pins Floating @ 4 MHz</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td>Symbol</td>
<td>Parameter</td>
<td>$V_{cc}$</td>
<td>$T_a = 0^\circ$C to +70$^\circ$C</td>
<td>Typical @ 25$^\circ$C</td>
<td>Units</td>
<td>Conditions</td>
<td>Notes</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td>---------</td>
<td>-------------------------------</td>
<td>-----------------------</td>
<td>-------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>$I_{CC1}$</td>
<td>Standby Current (Low Noise Mode)</td>
<td>4.5V</td>
<td>1.2</td>
<td>1.0</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V, V_{cc}$</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>1.6</td>
<td>1.9</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V, V_{cc}$</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.5V</td>
<td>1.5</td>
<td>0.5</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V, V_{cc}$</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>1.9</td>
<td>1</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V, V_{cc}$</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.5V</td>
<td>2.0</td>
<td>0.8</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V, V_{cc}$</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>2.4</td>
<td>1.3</td>
<td>mA</td>
<td>HALT mode $V_{IN} = 0V, V_{cc}$</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td>$I_{CC2}$</td>
<td>Standby Current</td>
<td>4.5V</td>
<td>10</td>
<td>1.0</td>
<td>µA</td>
<td>STOP mode $V_{IN} = 0V, V_{cc}$</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>10</td>
<td>1.0</td>
<td>µA</td>
<td>STOP mode $V_{IN} = 0V, V_{cc}$</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td>$I_{ALL}$</td>
<td>Auto Latch Low Current</td>
<td>4.5V</td>
<td>10</td>
<td>6.0</td>
<td>µA</td>
<td>$0V &lt; V_{IN} &lt; V_{cc}$</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>15</td>
<td>11.5</td>
<td>µA</td>
<td>$0V &lt; V_{IN} &lt; V_{cc}$</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td>$I_{ALH}$</td>
<td>Auto Latch High Current</td>
<td>4.5V</td>
<td>-7.0</td>
<td>-3.3</td>
<td>µA</td>
<td>$0V &lt; V_{IN} &lt; V_{cc}$</td>
<td>[4,5,8,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>-7.0</td>
<td>-6.5</td>
<td>µA</td>
<td>$0V &lt; V_{IN} &lt; V_{cc}$</td>
<td>[4,5,8,10]</td>
</tr>
</tbody>
</table>

**Notes:**

1. $I_{CC1}$: Clock Driven
   - Typ: 0.3
   - Max: 5.0
   - Unit: mA
   - Freq: 6 MHz
2. $V_{ee} = 0V = \text{GND}$
3. $V_{cc}$ must be in the allowed operating range (4.5V to 5.5V) prior to the minimum $T_{POR}$ time-out. $V_{cc}$ is specified at 4.5V to 5.5V.
4. All outputs unloaded, I/O pins floating, inputs at rail.
5. CL1 = CL2 = 100 pF.
7. Except clock pins and Port 3 input pins unless in EPROM Mode.
8. Using resonator/or crystal (not by Clock Driver).
9. Standard Mode (Low EMI not selected).
10. Low EMI selected.
11. Typical is $V_{cc} = 5.0V$; Temperature = 25$^\circ$C.
12. For comparator inputs, the inputs must be in the common-mode range.
13. A 10 Megohm pull-down resistor may be required on the XTAL1 clock input pin.
AC ELECTRICAL CHARACTERISTICS

Figure 22. Electrical Timing Diagram
## AC ELECTRICAL CHARACTERISTICS

### Low Noise Mode

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{CC}$[3]</th>
<th>1 MHz</th>
<th>4 MHz</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.5V</td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>DC</td>
<td>250</td>
<td>DC</td>
<td>250</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input Clock Period</td>
<td></td>
<td>1000</td>
<td>DC</td>
<td>250</td>
<td>DC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clock Input Rise and Fall Times</td>
<td>4.5V</td>
<td>25</td>
<td>25</td>
<td>ns [1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input Clock Width</td>
<td></td>
<td>4.5V</td>
<td>500</td>
<td>125</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>500</td>
<td>125</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Timer Input Low Width</td>
<td></td>
<td>4.5V</td>
<td>100</td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Timer Input High Width</td>
<td></td>
<td>4.5V</td>
<td>2.5$T_pC$</td>
<td>2.5$T_pC$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>2.5$T_pC$</td>
<td>2.5$T_pC$</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Timer Input Period</td>
<td></td>
<td>4.5V</td>
<td>4$T_pC$</td>
<td>4$T_pC$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>4$T_pC$</td>
<td>4$T_pC$</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Timer Input Rise and Fall Timer</td>
<td>4.5V</td>
<td>100</td>
<td>100</td>
<td>ns [1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int. Request Input Low Time</td>
<td>4.5V</td>
<td>100</td>
<td>100</td>
<td>ns [1,2]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Int. Request Input High Time</td>
<td>4.5V</td>
<td>2.5$T_pC$</td>
<td>2.5$T_pC$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>2.5$T_pC$</td>
<td>2.5$T_pC$</td>
<td>[1,2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Watch-Dog Timer Delay Time</td>
<td>4.5V</td>
<td>15</td>
<td>15</td>
<td>ms [1,3]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>10</td>
<td>10</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Power-On Reset Time</td>
<td></td>
<td>4.5V</td>
<td>15</td>
<td>10</td>
<td>ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>15</td>
<td>10</td>
<td>ms</td>
</tr>
</tbody>
</table>

### Notes:

1. Timing Reference uses 0.7 $V_{CC}$ for a logic 1 and 0.2 $V_{CC}$ for a logic 0.
2. Interrupt request through Port 3 (P33-P31).
3. Delay time between WDT refresh.
## AC ELECTRICAL CHARACTERISTICS
### Standard Mode, Standard Temperature

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$[3]</th>
<th>8 MHz</th>
<th>12 MHz</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TpC</td>
<td>Input Clock Period</td>
<td>4.5V</td>
<td>Min</td>
<td>Max</td>
<td>4.5V</td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>125</td>
<td>DC</td>
<td>5.5V</td>
<td>125</td>
</tr>
<tr>
<td>2</td>
<td>TrC,TfC</td>
<td>Clock Input Rise and Fall Times</td>
<td>4.5V</td>
<td>Min</td>
<td>Max</td>
<td>4.5V</td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>15</td>
<td>5.5V</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>TwC</td>
<td>Input Clock Width</td>
<td>4.5V</td>
<td>Min</td>
<td>Max</td>
<td>4.5V</td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>62</td>
<td>41</td>
<td>5.5V</td>
<td>62</td>
</tr>
<tr>
<td>4</td>
<td>TwTinL</td>
<td>Timer Input Low Width</td>
<td>4.5V</td>
<td>Min</td>
<td>Max</td>
<td>4.5V</td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>70</td>
<td>5.5V</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>TwTinH</td>
<td>Timer Input High Width</td>
<td>4.5V</td>
<td>Min</td>
<td>Max</td>
<td>4.5V</td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5.5V</td>
<td>5TpC</td>
</tr>
<tr>
<td>6</td>
<td>TpTin</td>
<td>Timer Input Period</td>
<td>4.5V</td>
<td>Min</td>
<td>Max</td>
<td>4.5V</td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>8TpC</td>
<td>8TpC</td>
<td>5.5V</td>
<td>8TpC</td>
</tr>
<tr>
<td>7</td>
<td>TrTin,</td>
<td>Timer Input Rise and Fall Timer</td>
<td>4.5V</td>
<td>Min</td>
<td>Max</td>
<td>4.5V</td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td>TtTin</td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
<td>5.5V</td>
<td>100</td>
</tr>
<tr>
<td>8</td>
<td>TwIL</td>
<td>Int. Request Input Low Time</td>
<td>4.5V</td>
<td>Min</td>
<td>Max</td>
<td>4.5V</td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
<td>5.5V</td>
<td>100</td>
</tr>
<tr>
<td>9</td>
<td>TwIH</td>
<td>Int. Request Input High Time</td>
<td>4.5V</td>
<td>Min</td>
<td>Max</td>
<td>4.5V</td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5.5V</td>
<td>5TpC</td>
</tr>
<tr>
<td>10</td>
<td>Twdt</td>
<td>Watch-Dog Timer Delay Time</td>
<td>4.5V</td>
<td>Min</td>
<td>Max</td>
<td>4.5V</td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>15</td>
<td>15</td>
<td>5.5V</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>TPOR</td>
<td>Power-On Reset Timer</td>
<td>4.5V</td>
<td>Min</td>
<td>Max</td>
<td>4.5V</td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>15</td>
<td>10</td>
<td>5.5V</td>
<td>15</td>
</tr>
</tbody>
</table>

**Notes:**

[1] Timing Reference uses 0.7 $V_{cc}$ for a logic 1 and 0.2 $V_{cc}$ for a logic 0.
[2] Interrupt request through Port 3 (P33-P31)
Z8® CONTROL REGISTERS

Figure 23. Timer Mode Register (F1H: Read/Write)

Figure 24. Counter Timer 1 Register (F2H: Read/Write)

Figure 25. Prescaler 1 Register (F3H: Write Only)

Figure 26. Counter/Timer 0 Register (F4H: Read/Write)

Figure 27. Prescaler 0 Register (F5H: Write Only)

Figure 28. Port 2 Mode Register (F6H: Write Only)

Figure 29. Port 3 Mode Register (F7H: Write Only)
Z8 CONTROL REGISTERS (Continued)

R248 P01M

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- P02, P03 Mode
- 00 = Output
- 01 = Input
- Reserved (Must be 1.)
- Reserved (Must be 0.)

Figure 30. Port 0 and 1 Mode Register
(F8H: Write Only)

R249 IPR

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Interrupt Group Priority
- Reserved = 000
- C > A > B = 001
- A > B > C = 010
- A > C > B = 011
- B > C > A = 100
- C > B > A = 101
- B > A > C = 110
- Reserved = 111

- IRQ1, IRQ4 Priority (Group C)
- 0 = IRQ1 > IRQ4
- 1 = IRQ4 > IRQ1

- IRQ0, IRQ2 Priority (Group B)
- 0 = IRQ2 > IRQ0
- 1 = IRQ0 > IRQ2

- IRQ3, IRQ5 Priority (Group A)
- 0 = IRQ5 > IRQ3
- 1 = IRQ3 > IRQ5
- Reserved (Must be 0.)

Figure 31. Interrupt Priority Register
(F9H: Write Only)

R251 IMR

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 1 Enables IRQ0-IRQ5
- (P0 = IRQ0)
- Reserved (Must be 0.)
- 1 Enables Interrupts

Figure 33. Interrupt Mask Register
(FBH: Write Only)

R252 Flags

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- User Flag F1
- User Flag F2
- Half Carry Flag
- Decimal Adjust Flag
- Overflow Flag
- Sign Flag
- Zero Flag
- Carry Flag

Figure 34. Flag Register
(FCH: Read/Write)

R253 RP

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Register (Must be 0.)
- Register Pointer

Figure 35. Register Pointer
(FDH: Read/Write)

R255 SPL

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Stack Pointer Lower Byte (SP7 - SP0)

Figure 36. Stack Pointer
(FFH: Read/Write)
OPERATING MODES

Figure 37. Maximum $I_{cc}$ and $I_{cc1}$ vs Frequency in Standard Mode

Figure 38. Typical $I_{cc}$ and $I_{cc1}$ vs Frequency in Standard Mode
OPERATING MODES (Continued)

Figure 39. Typical $I_{cc}$ and $I_{cc1}$ vs Frequency in Low EMI Mode

Figure 40. Maximum $I_{cc}$ and $I_{cc1}$ vs Frequency in Low EMI Mode
Figure 41. Typical POR Time Out Period vs Temperature

Figure 42. Typical WDT Time Out Period vs Temperature
### INSTRUCTION SET NOTATION

**Addressing Modes.** The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRR</td>
<td>Indirect register pair or indirect working-register pair address</td>
</tr>
<tr>
<td>Ir</td>
<td>Indirect working-register pair only</td>
</tr>
<tr>
<td>X</td>
<td>Indexed address</td>
</tr>
<tr>
<td>DA</td>
<td>Direct address</td>
</tr>
<tr>
<td>RA</td>
<td>Relative address</td>
</tr>
<tr>
<td>IM</td>
<td>Immediate</td>
</tr>
<tr>
<td>R</td>
<td>Register or working-register address</td>
</tr>
<tr>
<td>r</td>
<td>Working register address only</td>
</tr>
<tr>
<td>IR</td>
<td>Indirect-register or indirect working-register address</td>
</tr>
<tr>
<td>Ir</td>
<td>Indirect working-register address only</td>
</tr>
<tr>
<td>RR</td>
<td>Register pair or working register pair address</td>
</tr>
</tbody>
</table>

**Symbols.** The following symbols are used in describing the instruction set.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst</td>
<td>Destination location or contents</td>
</tr>
<tr>
<td>src</td>
<td>Source location or contents</td>
</tr>
<tr>
<td>cc</td>
<td>Condition code</td>
</tr>
<tr>
<td>@</td>
<td>Indirect address prefix</td>
</tr>
<tr>
<td>SP</td>
<td>Stack pointer</td>
</tr>
<tr>
<td>PC</td>
<td>Program counter</td>
</tr>
<tr>
<td>FLAGS</td>
<td>Flag register (Control Register 252)</td>
</tr>
<tr>
<td>RP</td>
<td>Register Pointer (R253)</td>
</tr>
<tr>
<td>IMR</td>
<td>Interrupt mask register (R251)</td>
</tr>
</tbody>
</table>

**Flags.** Control register (R252) contains the following six flags.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Carry flag</td>
</tr>
<tr>
<td>Z</td>
<td>Zero flag</td>
</tr>
<tr>
<td>S</td>
<td>Sign flag</td>
</tr>
<tr>
<td>V</td>
<td>Overflow flag</td>
</tr>
<tr>
<td>D</td>
<td>Decimal-adjust flag</td>
</tr>
<tr>
<td>H</td>
<td>Half-carry flag</td>
</tr>
</tbody>
</table>

Affected flags are indicated by:

- 0: Clear to zero
- 1: Set to one
- *: Set to clear according to operation
- -: Unaffected
- X: Undefined
## CONDITION CODES

<table>
<thead>
<tr>
<th>Value</th>
<th>Mnemonic</th>
<th>Meaning</th>
<th>Flags Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>---</td>
<td>Always true</td>
<td>---</td>
</tr>
<tr>
<td>0111</td>
<td>C</td>
<td>Carry</td>
<td>C=1</td>
</tr>
<tr>
<td>1111</td>
<td>NC</td>
<td>No Carry</td>
<td>C=0</td>
</tr>
<tr>
<td>0110</td>
<td>Z</td>
<td>Zero</td>
<td>Z=1</td>
</tr>
<tr>
<td>1110</td>
<td>NZ</td>
<td>Not zero</td>
<td>Z=0</td>
</tr>
<tr>
<td>1101</td>
<td>PL</td>
<td>Plus</td>
<td>S=0</td>
</tr>
<tr>
<td>0101</td>
<td>MI</td>
<td>Minus</td>
<td>S=1</td>
</tr>
<tr>
<td>0100</td>
<td>OV</td>
<td>Overflow</td>
<td>V=1</td>
</tr>
<tr>
<td>1100</td>
<td>NOV</td>
<td>No overflow</td>
<td>V=0</td>
</tr>
<tr>
<td>0110</td>
<td>EQ</td>
<td>Equal</td>
<td>Z=1</td>
</tr>
<tr>
<td>1110</td>
<td>NE</td>
<td>Not equal</td>
<td>Z=0</td>
</tr>
<tr>
<td>1001</td>
<td>GE</td>
<td>Greater than or equal</td>
<td>(S XOR V)=0</td>
</tr>
<tr>
<td>0001</td>
<td>LT</td>
<td>Less than</td>
<td>(S XOR V)=1</td>
</tr>
<tr>
<td>1010</td>
<td>GT</td>
<td>Greater than</td>
<td>[Z OR (S XOR V)]=0</td>
</tr>
<tr>
<td>0010</td>
<td>LE</td>
<td>Less than or equal</td>
<td>[Z OR (S XOR V)]=1</td>
</tr>
<tr>
<td>1111</td>
<td>UGE</td>
<td>Unsigned greater than or equal</td>
<td>C=0</td>
</tr>
<tr>
<td>0111</td>
<td>ULT</td>
<td>Unsigned less than</td>
<td>C=1</td>
</tr>
<tr>
<td>1011</td>
<td>UGT</td>
<td>Unsigned greater than</td>
<td>(C = 0 AND Z=0)=1</td>
</tr>
<tr>
<td>0011</td>
<td>ULE</td>
<td>Unsigned less than or equal</td>
<td>(C OR Z)=1</td>
</tr>
<tr>
<td>0000</td>
<td>F</td>
<td>Never true (Always False)</td>
<td>---</td>
</tr>
</tbody>
</table>
# Instruction Formats

## One-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>opc</th>
<th>src</th>
<th>addr</th>
<th>cc</th>
<th>opc</th>
<th>dst</th>
<th>opc</th>
<th>src</th>
<th>addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLR, CPL, DA, DEC, DECW, INC, INCW, POP, PUSH, RL, RLC, RR, RRC, SRA, SWAP</td>
<td>JP, CALL (Indirect)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Two-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>src</th>
<th>opc</th>
<th>addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC, ADD, AND, CP, LD, OR, SBC, SUB, TCM, TM, XOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Three-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst</th>
<th>src</th>
<th>opc</th>
<th>addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Instruction Summary

### One-Byte Instructions

- Assignment of a value is indicated by the symbol "←". For example:
  - `dst ← dst + src`

### Two-Byte Instructions

- Notation "addr (n)" is used to refer to bit (n) of a given operand location. For example:
  - `dst(7)` refers to bit 7 of the destination operand.
## INSTRUCTION SUMMARY

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
<th>C</th>
<th>Z</th>
<th>S</th>
<th>V</th>
<th>D</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC dst, src</td>
<td>†</td>
<td>1[ ]</td>
<td>* * * * 0 *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD dst, src</td>
<td>†</td>
<td>0[ ]</td>
<td>* * * * 0 *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AND dst, src</td>
<td>†</td>
<td>5[ ]</td>
<td>- * * 0 -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALL dst</td>
<td>DA</td>
<td>D6</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALL dst</td>
<td>IRR</td>
<td>D4</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCF</td>
<td>EF</td>
<td>* * * * -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLR dst</td>
<td>R</td>
<td>B0</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COM dst</td>
<td>R</td>
<td>60</td>
<td>- * * 0 -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP dst, src</td>
<td>†</td>
<td>A[ ]</td>
<td>* * * * -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DA dst</td>
<td>R</td>
<td>40</td>
<td>* * * X -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEC dst</td>
<td>R</td>
<td>00</td>
<td>- * * * -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECW dst</td>
<td>RR</td>
<td>80</td>
<td>- * * * -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DI</td>
<td>8F</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DJNZR, dst r←r - 1</td>
<td>RA</td>
<td>rA</td>
<td>r = 0 - F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DJNZR, dst r←r - 1</td>
<td>IF r ≠ 0</td>
<td>PC←PC + dst</td>
<td>Range: +127, -128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EI</td>
<td>9F</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMR(7)←1</td>
<td>1</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HALT</td>
<td>7F</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
<th>C</th>
<th>Z</th>
<th>S</th>
<th>V</th>
<th>D</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>INC dst</td>
<td>r</td>
<td>rE</td>
<td>- * * * -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INC dst</td>
<td>dst←dst + 1</td>
<td>R</td>
<td>0 - F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INCW dst</td>
<td>dst←dst + 1</td>
<td>RR</td>
<td>A0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRET</td>
<td>BF</td>
<td>* * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JP cc, dst</td>
<td>DA</td>
<td>cD</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR cc, dst</td>
<td>RA</td>
<td>cB</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD dst, src</td>
<td>r</td>
<td>lr</td>
<td>rC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD dst, src</td>
<td>dst←src</td>
<td>r</td>
<td>r8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDC dst, src</td>
<td>r</td>
<td>lr</td>
<td>C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDCI dst, src</td>
<td>lr</td>
<td>lr</td>
<td>C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### INSTRUCTION SUMMARY (Continued)

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOP</td>
<td></td>
<td>FF</td>
<td></td>
</tr>
<tr>
<td>OR dst, src</td>
<td>†</td>
<td>4[ ]</td>
<td></td>
</tr>
<tr>
<td>POP dst</td>
<td>R</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>PUSH src</td>
<td>R</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>RCF</td>
<td>CF</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>RET</td>
<td>AF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RL dst</td>
<td>R</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>RLC dst</td>
<td>R</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>RR dst</td>
<td>R</td>
<td>E0</td>
<td></td>
</tr>
<tr>
<td>RRC dst</td>
<td>R</td>
<td>C0</td>
<td></td>
</tr>
<tr>
<td>SBC dst, src</td>
<td>†</td>
<td>3[ ]</td>
<td></td>
</tr>
<tr>
<td>SCF</td>
<td></td>
<td>DF</td>
<td></td>
</tr>
<tr>
<td>SRA dst</td>
<td>R</td>
<td>D0</td>
<td></td>
</tr>
<tr>
<td>SRP dst</td>
<td>IR</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>STOP</td>
<td></td>
<td>6F</td>
<td></td>
</tr>
<tr>
<td>SUB dst, src</td>
<td>†</td>
<td>2[ ]</td>
<td></td>
</tr>
<tr>
<td>SWAP dst</td>
<td>R</td>
<td>F0</td>
<td></td>
</tr>
<tr>
<td>TCM dst, src</td>
<td>†</td>
<td>6[ ]</td>
<td></td>
</tr>
<tr>
<td>TM dst, src</td>
<td>†</td>
<td>7[ ]</td>
<td></td>
</tr>
<tr>
<td>WDH</td>
<td></td>
<td>4F</td>
<td></td>
</tr>
<tr>
<td>WDT</td>
<td></td>
<td>5F</td>
<td></td>
</tr>
<tr>
<td>XOR dst, src</td>
<td>†</td>
<td>B[ ]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>STOP</td>
<td></td>
<td>6F</td>
<td></td>
</tr>
<tr>
<td>SUB dst, src</td>
<td>†</td>
<td>2[ ]</td>
<td></td>
</tr>
<tr>
<td>SWAP dst</td>
<td>R</td>
<td>F0</td>
<td></td>
</tr>
<tr>
<td>TCM dst, src</td>
<td>†</td>
<td>6[ ]</td>
<td></td>
</tr>
<tr>
<td>TM dst, src</td>
<td>†</td>
<td>7[ ]</td>
<td></td>
</tr>
<tr>
<td>WDH</td>
<td></td>
<td>4F</td>
<td></td>
</tr>
<tr>
<td>WDT</td>
<td></td>
<td>5F</td>
<td></td>
</tr>
<tr>
<td>XOR dst, src</td>
<td>†</td>
<td>B[ ]</td>
<td></td>
</tr>
</tbody>
</table>

† These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a ‘[ ]’ in this table, and its value is found in the following table to the left of the applicable addressing mode pair.

For example, the opcode of an ADC instruction using the addressing modes r (destination) and Ir (source) is 13.

<table>
<thead>
<tr>
<th>Address Mode</th>
<th>Lower Opcode Nibble</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>[2]</td>
</tr>
<tr>
<td>Ir</td>
<td>[3]</td>
</tr>
<tr>
<td>R</td>
<td>[4]</td>
</tr>
<tr>
<td>IR</td>
<td>[5]</td>
</tr>
<tr>
<td>IM</td>
<td>[6]</td>
</tr>
<tr>
<td>IM</td>
<td>[7]</td>
</tr>
</tbody>
</table>
### Opcode Map

#### Upper Nibble (Hex)

<table>
<thead>
<tr>
<th>Upper Nibble (Hex)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
<td>06.5 DEC R1</td>
<td>06.5 DEC R1</td>
<td>06.5 ADD R1</td>
<td>06.5 ADD R1</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1</td>
<td>10.5 ADD R1, R1</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1</td>
<td>10.5 ADD R1, R1</td>
<td>12/10.5 LD R2, R1</td>
<td>12/10.5 JR cc, R1</td>
<td>12/10.0 LD R1, R1</td>
</tr>
<tr>
<td><strong>B</strong></td>
<td>06.5 RLC R1</td>
<td>06.5 RLC R1</td>
<td>06.5 ADC R1, R1</td>
<td>06.5 ADC R1, R1</td>
<td>10.5 ADC R2, R1</td>
<td>10.5 ADC R2, R1</td>
<td>10.5 ADC R1, R1</td>
<td>10.5 ADC R1, R1</td>
<td>10.5 ADC R2, R1</td>
<td>10.5 ADC R2, R1</td>
<td>10.5 ADC R1, R1</td>
<td>10.5 ADC R1, R1</td>
<td>4.0 WDH R1 R1</td>
<td>5.0 WDT R1 R1</td>
<td>6.0 STOP R1 R1</td>
</tr>
<tr>
<td><strong>C</strong></td>
<td>06.5 INC R1</td>
<td>06.5 INC R1</td>
<td>06.5 SUB R1</td>
<td>06.5 SUB R1</td>
<td>10.5 SUB R2, R1</td>
<td>10.5 SUB R2, R1</td>
<td>10.5 SUB R1, R1</td>
<td>10.5 SUB R1, R1</td>
<td>10.5 SUB R2, R1</td>
<td>10.5 SUB R2, R1</td>
<td>10.5 SUB R1, R1</td>
<td>10.5 SUB R1, R1</td>
<td>6.1 EI R1 R1</td>
<td>14.0 RET R1 R1</td>
<td>16.0 IRET R1 R1</td>
</tr>
<tr>
<td><strong>D</strong></td>
<td>06.5 COM R1</td>
<td>06.5 COM R1</td>
<td>10.5 AND R1</td>
<td>06.5 AND R1</td>
<td>10.5 AND R2, R1</td>
<td>10.5 AND R2, R1</td>
<td>10.5 AND R1, R1</td>
<td>10.5 AND R1, R1</td>
<td>10.5 AND R2, R1</td>
<td>10.5 AND R2, R1</td>
<td>10.5 AND R1, R1</td>
<td>10.5 AND R1, R1</td>
<td>14.0 CALL* IR1 R1</td>
<td>20.0 CALL* IR1 R1</td>
<td>20.0 CALL* DA R2, R1</td>
</tr>
<tr>
<td><strong>E</strong></td>
<td>06.5 CLR R1</td>
<td>06.5 CLR R1</td>
<td>06.5 OR R1</td>
<td>06.5 OR R1</td>
<td>10.5 OR R2, R1</td>
<td>10.5 OR R2, R1</td>
<td>10.5 OR R1, R1</td>
<td>10.5 OR R1, R1</td>
<td>10.5 OR R2, R1</td>
<td>10.5 OR R2, R1</td>
<td>10.5 OR R1, R1</td>
<td>10.5 OR R1, R1</td>
<td>6.5 SRA R1</td>
<td>6.5 SRA R1</td>
<td>6.5 SRA R1</td>
</tr>
<tr>
<td><strong>F</strong></td>
<td>06.5 RRC R1</td>
<td>06.5 RRC R1</td>
<td>10.5 LD R1</td>
<td>06.5 LD R1</td>
<td>10.5 LD R2, R1</td>
<td>10.5 LD R2, R1</td>
<td>10.5 LD R1, R1</td>
<td>10.5 LD R1, R1</td>
<td>10.5 LD R2, R1</td>
<td>10.5 LD R2, R1</td>
<td>10.5 LD R1, R1</td>
<td>10.5 LD R1, R1</td>
<td>6.5 SWAP R1</td>
<td>6.5 SWAP R1</td>
<td>6.5 SWAP R1</td>
</tr>
</tbody>
</table>

#### Lower Nibble (Hex)

<table>
<thead>
<tr>
<th>Lower Nibble (Hex)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0</strong></td>
<td>06.5 DEC R1</td>
<td>06.5 DEC R1</td>
<td>06.5 ADD R1</td>
<td>06.5 ADD R1</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1</td>
<td>10.5 ADD R1, R1</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1</td>
<td>10.5 ADD R1, R1</td>
<td>4.0 WDH R1 R1</td>
<td>5.0 WDT R1 R1</td>
<td>6.0 STOP R1 R1</td>
</tr>
<tr>
<td><strong>1</strong></td>
<td>06.5 RLC R1</td>
<td>06.5 RLC R1</td>
<td>06.5 ADC R1, R1</td>
<td>06.5 ADC R1, R1</td>
<td>10.5 ADC R2, R1</td>
<td>10.5 ADC R2, R1</td>
<td>10.5 ADC R1, R1</td>
<td>10.5 ADC R1, R1</td>
<td>10.5 ADC R2, R1</td>
<td>10.5 ADC R2, R1</td>
<td>10.5 ADC R1, R1</td>
<td>10.5 ADC R1, R1</td>
<td>14.0 CALL* IR1 R1</td>
<td>20.0 CALL* IR1 R1</td>
<td>20.0 CALL* DA R2, R1</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>06.5 INC R1</td>
<td>06.5 INC R1</td>
<td>06.5 SUB R1</td>
<td>06.5 SUB R1</td>
<td>10.5 SUB R2, R1</td>
<td>10.5 SUB R2, R1</td>
<td>10.5 SUB R1, R1</td>
<td>10.5 SUB R1, R1</td>
<td>10.5 SUB R2, R1</td>
<td>10.5 SUB R2, R1</td>
<td>10.5 SUB R1, R1</td>
<td>10.5 SUB R1, R1</td>
<td>6.5 SRA R1</td>
<td>6.5 SRA R1</td>
<td>6.5 SRA R1</td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>06.5 COM R1</td>
<td>06.5 COM R1</td>
<td>10.5 AND R1</td>
<td>06.5 AND R1</td>
<td>10.5 AND R2, R1</td>
<td>10.5 AND R2, R1</td>
<td>10.5 AND R1, R1</td>
<td>10.5 AND R1, R1</td>
<td>10.5 AND R2, R1</td>
<td>10.5 AND R2, R1</td>
<td>10.5 AND R1, R1</td>
<td>10.5 AND R1, R1</td>
<td>6.5 SRA R1</td>
<td>6.5 SRA R1</td>
<td>6.5 SRA R1</td>
</tr>
<tr>
<td><strong>4</strong></td>
<td>06.5 CLR R1</td>
<td>06.5 CLR R1</td>
<td>06.5 OR R1</td>
<td>06.5 OR R1</td>
<td>10.5 OR R2, R1</td>
<td>10.5 OR R2, R1</td>
<td>10.5 OR R1, R1</td>
<td>10.5 OR R1, R1</td>
<td>10.5 OR R2, R1</td>
<td>10.5 OR R2, R1</td>
<td>10.5 OR R1, R1</td>
<td>10.5 OR R1, R1</td>
<td>6.5 SRA R1</td>
<td>6.5 SRA R1</td>
<td>6.5 SRA R1</td>
</tr>
<tr>
<td><strong>5</strong></td>
<td>06.5 RRC R1</td>
<td>06.5 RRC R1</td>
<td>10.5 LD R1</td>
<td>06.5 LD R1</td>
<td>10.5 LD R2, R1</td>
<td>10.5 LD R2, R1</td>
<td>10.5 LD R1, R1</td>
<td>10.5 LD R1, R1</td>
<td>10.5 LD R2, R1</td>
<td>10.5 LD R2, R1</td>
<td>10.5 LD R1, R1</td>
<td>10.5 LD R1, R1</td>
<td>6.5 SRA R1</td>
<td>6.5 SRA R1</td>
<td>6.5 SRA R1</td>
</tr>
<tr>
<td><strong>6</strong></td>
<td>06.5 SWAP R1</td>
<td>06.5 SWAP R1</td>
<td>06.5 LD R1</td>
<td>06.5 LD R1</td>
<td>10.5 LD R2, R1</td>
<td>10.5 LD R2, R1</td>
<td>10.5 LD R1, R1</td>
<td>10.5 LD R1, R1</td>
<td>10.5 LD R2, R1</td>
<td>10.5 LD R2, R1</td>
<td>10.5 LD R1, R1</td>
<td>10.5 LD R1, R1</td>
<td>6.5 SRA R1</td>
<td>6.5 SRA R1</td>
<td>6.5 SRA R1</td>
</tr>
</tbody>
</table>

#### Legend:
- **R** = 8-bit address
- **r** = 4-bit address
- **R1** or **r1** = Dst address
- **R2** or **r2** = Src address

#### Sequence:
- Opcode, First Operand, Second Operand

#### Note:
- Blank areas are reserved.
- * 2-byte instruction appears as a 3-byte instruction

#### Byte per Instruction:
- **2** bytes per Instruction
- **3** bytes per Instruction

---

**Execution Cycle:**
- Upper Opcode Nibble
- Lower Opcode Nibble
- Mnemonic
- First Operand
- Second Operand

**CPU Cycles:**
- Pipeline Cycles
- Lower Nibble

**Note:**
- Blank areas are reserved.
18-Pin DIP Package Diagram

CONTROLLING DIMENSIONS • MILLIMETER

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>0.51</td>
<td>0.61</td>
</tr>
<tr>
<td>A2</td>
<td>2.25</td>
<td>3.43</td>
</tr>
<tr>
<td>B</td>
<td>0.30</td>
<td>0.53</td>
</tr>
<tr>
<td>B1</td>
<td>1.14</td>
<td>1.63</td>
</tr>
<tr>
<td>C</td>
<td>0.02</td>
<td>0.30</td>
</tr>
<tr>
<td>D</td>
<td>2.25</td>
<td>2.37</td>
</tr>
<tr>
<td>E</td>
<td>7.62</td>
<td>8.13</td>
</tr>
<tr>
<td>E1</td>
<td>6.22</td>
<td>6.48</td>
</tr>
</tbody>
</table>

18-Pin SOIC Package Diagram

CONTROLLING DIMENSIONS • MILLIMETER

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.94</td>
<td>1.04</td>
</tr>
<tr>
<td>A1</td>
<td>0.30</td>
<td>0.40</td>
</tr>
<tr>
<td>A2</td>
<td>1.14</td>
<td>1.63</td>
</tr>
<tr>
<td>B</td>
<td>0.45</td>
<td>0.86</td>
</tr>
<tr>
<td>B1</td>
<td>0.09</td>
<td>0.12</td>
</tr>
<tr>
<td>C</td>
<td>1.17</td>
<td>1.27</td>
</tr>
<tr>
<td>D</td>
<td>1.49</td>
<td>1.54</td>
</tr>
<tr>
<td>E</td>
<td>7.40</td>
<td>7.60</td>
</tr>
</tbody>
</table>

CONTROLLING DIMENSIONS • INCH

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.40</td>
<td>0.63</td>
</tr>
<tr>
<td>A1</td>
<td>0.20</td>
<td>0.30</td>
</tr>
<tr>
<td>A2</td>
<td>0.24</td>
<td>0.44</td>
</tr>
<tr>
<td>B</td>
<td>0.24</td>
<td>0.40</td>
</tr>
<tr>
<td>B1</td>
<td>0.10</td>
<td>0.16</td>
</tr>
<tr>
<td>C</td>
<td>0.10</td>
<td>0.16</td>
</tr>
<tr>
<td>D</td>
<td>1.10</td>
<td>1.10</td>
</tr>
<tr>
<td>E</td>
<td>0.74</td>
<td>0.76</td>
</tr>
<tr>
<td>E1</td>
<td>0.61</td>
<td>0.64</td>
</tr>
</tbody>
</table>

CONTROLLING DIMENSIONS • MM

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2.40</td>
<td>2.65</td>
</tr>
<tr>
<td>A1</td>
<td>0.10</td>
<td>0.30</td>
</tr>
<tr>
<td>A2</td>
<td>2.84</td>
<td>2.88</td>
</tr>
<tr>
<td>B</td>
<td>0.16</td>
<td>0.18</td>
</tr>
<tr>
<td>B1</td>
<td>0.20</td>
<td>0.22</td>
</tr>
<tr>
<td>C</td>
<td>0.20</td>
<td>0.22</td>
</tr>
<tr>
<td>D</td>
<td>1.10</td>
<td>1.10</td>
</tr>
<tr>
<td>E</td>
<td>7.40</td>
<td>7.60</td>
</tr>
<tr>
<td>E1</td>
<td>6.20</td>
<td>6.48</td>
</tr>
</tbody>
</table>

CONTROLLING DIMENSIONS • LEADS ARE COPLANAR WITHIN .004 INCH.

18-Pin SOIC Package Diagram

CONTROLLING DIMENSIONS • MILLIMETER

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>10.00</td>
<td>10.65</td>
</tr>
<tr>
<td>h</td>
<td>0.30</td>
<td>0.40</td>
</tr>
<tr>
<td>L</td>
<td>0.60</td>
<td>0.80</td>
</tr>
<tr>
<td>Q1</td>
<td>0.57</td>
<td>0.87</td>
</tr>
</tbody>
</table>

CONTROLLING DIMENSIONS • INCH

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>10.00</td>
<td>10.65</td>
</tr>
<tr>
<td>h</td>
<td>0.30</td>
<td>0.40</td>
</tr>
<tr>
<td>L</td>
<td>0.60</td>
<td>0.80</td>
</tr>
<tr>
<td>Q1</td>
<td>0.57</td>
<td>0.87</td>
</tr>
</tbody>
</table>
ORDERING INFORMATION

Z86E07 (12 MHz)

Standard Temperature

DIP     SOIC

Z86E0712PSC   Z86E0712SSC

For fast results, contact your local Zilog sales office or technical center for assistance in ordering the part desired.

CODES

Preferred Package

P = DIP

Longer Lead Time

S = SOIC

Preferred Temperature

S = 0°C to +70°C

Speed

12 = 12 MHz

Environmental

C= Plastic Standard

Example:

Z 86E07 12 P S C

is a Z86E07, 12 MHz, DIP, 0°C to +70°C, Plastic Standard Flow
Z86C03/C06 CMOS Z8® 8-Bit CCP™ Consumer Controller Processors

Z86E03/E06 CMOS Z8® 8-Bit OTP CCP™ Consumer Controller Processors

Z86C04/C08 CMOS Z8® Low Cost 1K /2K ROM Microcontrollers

Z86E04/E08 CMOS Z8® 8-Bit OTP Microcontrollers

Z86C07 CMOS Z8® 8-Bit Microcontroller

Z86E07 CMOS Z8® 8-Bit OTP Microcontroller

Z86C30/C31 CMOS Z8® 8-Bit CCP™ Consumer Controller Processors
FEATURES

The Z86C30/C31 Devices Have the Following General Characteristics:

<table>
<thead>
<tr>
<th>Part</th>
<th>ROM</th>
<th>RAM</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z86C30</td>
<td>4 Kbyte</td>
<td>236 Bytes</td>
<td>12 MHz</td>
</tr>
<tr>
<td>Z86C31</td>
<td>2 Kbyte</td>
<td>124 Bytes</td>
<td>8 MHz</td>
</tr>
</tbody>
</table>

- 28-Pin Package Styles (DIP, SOIC, PCB Chip Carrier)
- 3.0 to 5.5 Volt Operating Range
- Operating Temperature: -40° to +105°C
- Low Power Consumption: 50 mW (Typical)
- Fast Instruction Pointer: 1.5 µs @ 8 MHz (Z86C31), 1.0 µs @ 12 MHz (Z86C30)
- Two Programmable 8-Bit Counter/Timers Each with a 6-Bit Programmable Prescaler
- Two Standby Modes: STOP and HALT
- ROM Protect Option
- RAM Protect Option (Z86C30 Only)
- 24 Input/Output Lines (Two with Comparator Inputs)
- Seven Digital Inputs CMOS Levels, Schmitt-Triggered
- Three Digital Inputs CMOS Levels
- Three Expanded Register File Control Registers
- Low Voltage Protection
- Watch-Dog/Power-On Reset Timers
- Two Comparators with Programmable Interrupt Polarity
- Six Vectored, Priority Interrupts from Six Different Sources
- On-Chip Oscillator that Accepts a Crystal, Ceramic Resonator, LC, RC, or External Clock Drive.
- Software Programmable Low EMI Mode
- Open-Drain Mode on Three Ports
- Auto Latches

GENERAL DESCRIPTION

The Z86C30/C31 CCP™ (Consumer Controller Processors) are members of Zilog's Z8® single-chip microcontroller family with enhanced wake-up circuitry, programmable watch-dog timers and low noise/EMI options. These enhancements result in a more efficient, cost-effective design and provide the user with increased design flexibility over the standard Z8 microcontroller core. With 4K/2K bytes of ROM and 236/124 bytes of RAM for the Z86C30 and Z86C31, respectively, these low cost, low power consumption CMOS microcontrollers offer fast execution, efficient use of memory, sophisticated interrupts, input/output bit manipulation capabilities, and easy hardware/software system expansion.

The Z86C30/C31 architecture is characterized by Zilog's 8-bit microcontroller core with an Expanded Register File to allow easy access to register mapped peripheral and I/O circuits. These devices offer a flexible I/O scheme, an efficient register and address space structure, and a number of ancillary features that are useful in many industrial, automotive, and industrial applications.

For applications demanding powerful I/O capabilities, the Z86C30/C31 provides 24 pins dedicated to input and output. These lines are grouped into three ports, eight lines per port, and are configurable under software control to provide timing, status signals, and parallel I/O with or without handshake.
GENERAL DESCRIPTION (Continued)

Three basic address spaces are available to support this wide range of configurations: Program Memory, Register File, and Expanded Register File. The Register File is composed of 236/124 bytes of general-purpose registers, three I/O port registers and 15 control and status registers. The Expanded Register File consists of three Control registers.

To unburden the system from coping with real-time tasks, such as counting/timing and input/output data communication, the Z86C30/C31 offers two on-chip counter/timers with a large number of user-selectable modes, and on-board comparators to process analog signals with a common reference voltage (Figure 1).

With powerful peripheral features such as on-board comparators, counter/timer(s), Watch-Dog Timer (WDT), the Z86C30/C31 meets the needs of a variety of sophisticated controller applications.

Notes:
All Signals with a preceding front slash, '/', are active Low, e.g., B/W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

<table>
<thead>
<tr>
<th>Connection</th>
<th>Circuit</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>V$_{cc}$</td>
<td>V$_{DD}$</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
<td>V$_{SS}$</td>
</tr>
</tbody>
</table>

Figure 1. Z86C30/C31 Functional Block Diagram
**PIN DESCRIPTION**

Table 1. 28-Pin DIP* Pin Identification

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>P25-27</td>
<td>Port 2, Pins 5,6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>4-7</td>
<td>P04-07</td>
<td>Port 0, Pins 4,5,6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>8</td>
<td>Vcc</td>
<td>Power Supply</td>
<td>In/Output</td>
</tr>
<tr>
<td>9</td>
<td>XTAL2</td>
<td>Crystal Oscillator</td>
<td>Output</td>
</tr>
<tr>
<td>10</td>
<td>XTAL1</td>
<td>Crystal Oscillator</td>
<td>Input</td>
</tr>
<tr>
<td>11-13</td>
<td>P04-07</td>
<td>Port 3, Pins 1,2,3</td>
<td>Fixed Input</td>
</tr>
<tr>
<td>14-15</td>
<td>P34-35</td>
<td>Port 3, Pins 4,5</td>
<td>Fixed Output</td>
</tr>
<tr>
<td>16</td>
<td>P37</td>
<td>Port 3, Pin 7</td>
<td>Fixed Output</td>
</tr>
<tr>
<td>17</td>
<td>P36</td>
<td>Port 3, Pin 6</td>
<td>Fixed Output</td>
</tr>
<tr>
<td>18</td>
<td>P30</td>
<td>Port 3, Pin 0</td>
<td>Fixed Input</td>
</tr>
<tr>
<td>19-21</td>
<td>P00-02</td>
<td>Port 0, Pins 0,1,2</td>
<td>In/Output</td>
</tr>
<tr>
<td>22</td>
<td>GND</td>
<td>Ground</td>
<td>In/Output</td>
</tr>
<tr>
<td>23</td>
<td>P03</td>
<td>Port 0, Pins 3</td>
<td>In/Output</td>
</tr>
<tr>
<td>24-28</td>
<td>P20-24</td>
<td>Port 2, Pins 0,1,2,3,4</td>
<td>In/Output</td>
</tr>
</tbody>
</table>

Note: * SOIC style package is identical in pin identification and configuration.

Figure 2. 28-Pin DIP* Pin Configuration

Table 2. 28-Pin PCB Chip Carrier Pin Identification

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>P25-27</td>
<td>Port 2, Pins 5,6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>4-7</td>
<td>P04-07</td>
<td>Port 0, Pins 4,5,6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>8</td>
<td>Vcc</td>
<td>Power Supply</td>
<td>In/Output</td>
</tr>
<tr>
<td>9</td>
<td>XTAL2</td>
<td>Crystal Oscillator</td>
<td>Output</td>
</tr>
<tr>
<td>10</td>
<td>XTAL1</td>
<td>Crystal Oscillator</td>
<td>Input</td>
</tr>
<tr>
<td>11-13</td>
<td>P04-07</td>
<td>Port 3, Pins 1,2,3</td>
<td>Fixed Input</td>
</tr>
<tr>
<td>14-15</td>
<td>P34-35</td>
<td>Port 3, Pins 4,5</td>
<td>Fixed Output</td>
</tr>
<tr>
<td>16</td>
<td>P37</td>
<td>Port 3, Pin 7</td>
<td>Fixed Output</td>
</tr>
<tr>
<td>17</td>
<td>P36</td>
<td>Port 3, Pin 6</td>
<td>Fixed Output</td>
</tr>
<tr>
<td>18</td>
<td>P30</td>
<td>Port 3, Pin 0</td>
<td>Fixed Input</td>
</tr>
<tr>
<td>19-21</td>
<td>P00-02</td>
<td>Port 0, Pins 0,1,2</td>
<td>In/Output</td>
</tr>
<tr>
<td>22</td>
<td>GND</td>
<td>Ground</td>
<td>In/Output</td>
</tr>
<tr>
<td>23</td>
<td>P03</td>
<td>Port 0, Pins 3</td>
<td>In/Output</td>
</tr>
<tr>
<td>24-28</td>
<td>P20-24</td>
<td>Port 2, Pins 0,1,2,3,4</td>
<td>In/Output</td>
</tr>
</tbody>
</table>

Figure 3. 28-Pin PCB Chip Carrier Pin Configuration
**PIN FUNCTIONS**

**XTAL1. Crystal 1 (time-based input).** This pin connects a parallel-resonant crystal, ceramic resonator, LC or RC network or external single-phase clock to the on-chip oscillator input.

**XTAL2. Crystal 2 (time-based output).** This pin connects a parallel-resonant crystal, ceramic resonator, LC or RC network to the on-chip oscillator output.

**Port 0 (P07-P00).** Port 0 is an 8-bit, bi-directional, CMOS compatible I/O port. These eight I/O lines can be nibble programmed as P03-P00 input/output and P07-P04 input/output, separately. The input buffers are Schmitt-Triggered and nibbles programmed as outputs can be globally programmed as either push-pull or open-drain. Low EMI output buffers can be globally programmed by the software. Port 0 can also be used as a handshake I/O port.

In Handshake Mode, Port 3 lines P32 and P35 are used as handshake control lines. The handshake direction is determined by the configuration (input or output) assigned to Port 0's upper nibble. The lower nibble must have the same direction as the upper nibble (Figure 4).

---

**Figure 4. Port 0 Configuration**
Port 2 (P27-P20). Port 2 is an 8-bit, bi-directional, CMOS compatible I/O port. These eight I/O lines can be configured under software control as inputs or outputs, independently. All input buffers are Schmitt-Triggered. Bits programmed as outputs may be globally programmed as either push-pull or open-drain. Low EMI output buffers can be globally programmed by the software. When used as an I/O port, Port 2 can be placed under handshake control. In Handshake Mode, Port 3 lines P31 and P36 are used as handshake control lines. The handshake direction is determined by the configuration (input or output) assigned to bit 7 of Port 2 (Figure 5).

![Diagram of Port 2 Configuration]

**Figure 5. Port 2 Configuration**
**PIN FUNCTIONS** (Continued)

**Port 3 (P37-P30).** Port 3 is an 8-bit, CMOS compatible port. These eight lines consist of four fixed inputs (P33-P30) and four fixed outputs (P37-P34), and can be configured under software for interrupt and port handshake functions. Port 3 pin 0 input is Schmitt-triggered. Pins P31, P32 and P33 are standard CMOS inputs (no auto latches) and pins P34, P35, P36, P37 are push-pull output lines. Two on-board comparators can process analog signals on P31 and P32 with reference to the voltage on P33. The comparator output can be outputted from P34 and P37, respectively, by setting PCON register (PCON) bit D0 to 1. The analog function is enabled by programming the Port 3 Mode Register (P3M) (bit D1) for interrupt function. P30 and P33 are falling edge interrupt inputs. P31 and P32 are programmable as falling, rising, or both edge triggered interrupts (IRQ register bits 6 and 7). In Analog Mode, P33 is the comparator reference voltage input.

Access to Counter/Timer 1 is made through P31 (T1IN) and P36 (T1OVRL). Handshake lines for Ports 0 and 2 are available on P3 pin 1 through 6 (Figure 6).

![Port 3 Configuration Diagram](image-url)

**Figure 6. Port 3 Configuration**
### Table 3. Port 3 Pin Assignments

<table>
<thead>
<tr>
<th>Pin</th>
<th>I/O</th>
<th>CTC1</th>
<th>Analog</th>
<th>Int.</th>
<th>P0 HS</th>
<th>P2 HS</th>
</tr>
</thead>
<tbody>
<tr>
<td>P30</td>
<td>IN</td>
<td></td>
<td></td>
<td></td>
<td>IRQ3</td>
<td></td>
</tr>
<tr>
<td>P31</td>
<td>IN</td>
<td>T_In</td>
<td>AN1</td>
<td></td>
<td>IRQ2</td>
<td>D/R</td>
</tr>
<tr>
<td>P32</td>
<td>IN</td>
<td></td>
<td>AN2</td>
<td></td>
<td>IRQ0</td>
<td>D/R</td>
</tr>
<tr>
<td>P33</td>
<td>IN</td>
<td></td>
<td>REF</td>
<td></td>
<td>IRQ1</td>
<td></td>
</tr>
<tr>
<td>P34</td>
<td>OUT</td>
<td></td>
<td>AN1-OUT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P35</td>
<td>OUT</td>
<td></td>
<td></td>
<td></td>
<td>R/D</td>
<td></td>
</tr>
<tr>
<td>P36</td>
<td>OUT</td>
<td>T_out</td>
<td></td>
<td></td>
<td>R/D</td>
<td></td>
</tr>
<tr>
<td>P37</td>
<td>OUT</td>
<td></td>
<td>AN2-OUT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- HS = Handshake Signals
- D = DAV
- R = RDY

**Comparator Inputs.** Port 3 Pin P31 and P32 each have a comparator front end. The comparator reference voltage, Pin P33, is common to both comparators. In analog mode, the P31 and P32 are the positive inputs to the comparators, and P33 is the reference voltage supplied to both comparators. In analog mode, the P31 and P32 are the positive inputs to the comparators, and P33 is the reference voltage supplied to both comparators. In digital mode, pin P33 can be used as a P33 register input or IRQ1 source. The comparator outputs can be programmed out on P34 and P37 by setting the PCON register bit D0 to a 1 state.

**Auto Latch.** The Auto Latch puts valid CMOS levels on all CMOS inputs (except P33, P32, P31) that are not externally driven. Whether this is 0 or 1, cannot be determined. A valid CMOS level, rather than a floating mode, reduces excessive supply current flow in the input buffer.

**Note:** Deletion of all port pin auto latches is available as a ROM mask option. The auto latch delete option is selected by the customer when the ROM code is submitted. P01M reg. bit D4 and D3 must be "0" with the Auto Latch Delete option.

**Low EMI Emission.** The Z86C30/C31 can be programmed to operate in a low EMI emission mode in the PCON register. The oscillator and all I/O ports can be programmed as low EMI emission mode independently. Use of this feature results in:
- The pre-drivers slew rate reduced to 10 ns typical.
- Low EMI output drivers have resistance of 200 Ohms (typical).
- Low EMI Oscillator.
- Internal SCLK/TCLK = XTAL operation limited to a maximum of 4 MHz - 250 ns cycle time, when Low EMI Oscillator is selected and system clock (SCLK = XTAL, SMR Reg. Bit D1 = 1).
FUNCTIONAL DESCRIPTION

The following special functions have been incorporated into the Z86C30/C31 CCPs to enhance the standard Z8® core architecture to provide the user with increased design flexibility.

**Reset.** The device is reset in one of the following conditions:

- Power-On Reset
- Watch-Dog Timer
- Stop-Mode Recovery Source
- Low Voltage Recovery

Having the auto Power-on Reset circuitry built-in, the Z86C30/C31 does not require an external reset circuit. The reset time is 5 ms (typical), plus 18 clock cycles.

The Z86C30/C31 does not re-initialize WDTMR, SMR, P2M, PCON and P3M registers to their reset values on a Stop-Mode Recovery operation.

**Program Memory.** The Z86C30/C31 can address up to 4K/2K bytes of internal program memory (Figure 7). The first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six, 16-bit vectors that correspond to the six available interrupts. Address 12 to 4095/2047 are reserved for the user ROM Program. After reset, the program counter points to the program start address at 000CH.

**ROM Protect.** The 4K/2K bytes of program memory is mask programmable. A ROM protect feature prevents “dumping” of the ROM contents by inhibiting execution of LDC and LDCl instructions to program memory in ALL modes. A ROM look up table cannot be used with this feature selected.

The ROM protect option is mask-programmable and is selected by the customer when the ROM code is submitted.

**Expanded Register File (ERF).** The register file has been expanded to allow for additional system control registers, mapping of additional peripheral devices and input/output ports into the register address area. The Z8 register address space R0 through R15 is implemented as 16 groups of 16 registers per group (Figure 8). These register groups are known as the Expanded Register File (ERF).

Bits 3-0 of the Register Pointer (RP) select the active ERF group. Bits 7-4 of register RP select the working register group (Figure 9). Three system configuration registers reside in the Expanded Register File at bank F (PCON, SMR, WDTMR). The rest of the Expanded Register is not physically implemented and is open for future expansion.
Figure 8. Expanded Register File Architecture

Notes:
* Will not be reset with a STOP-Mode Recovery.
** Will not be reset with a STOP-Mode Recovery, except Bit D0.
U = Unknown
FUNCTIONAL DESCRIPTION (Continued)

**Expanded Register Group**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
</table>

| Working Register Group |

**Default setting after RESET = 00000000**

**Figure 9. Register Pointer Register**

**Register File.** The register file consists of three I/O port registers, 236/124 general-purpose registers and 15 control and status registers and three system configuration registers in the expanded register group (See Figure 8). The instructions can access registers directly or indirectly through an 8-bit address field. This allows a short 4-bit register address using the Register Pointer (Figure 9 and 10). In the 4-bit mode, the register file is divided into 16 working register groups, each occupying 16 continuous locations. The Register Pointer addresses the starting location of the active working register group. The general-purpose registers on device power-up are undefined.

**Note:** Register Bank E0-EF can only be accessed through working register and indirect addressing modes. (This bank is available in Z86C30 only.)

**General Purpose Register (GPR).** The general purpose registers are undefined after the device is powered-up. The registers keep their last value after any reset, as long as the reset occurs in the Vcc voltage-specified operating range. **Note:** Register R254 has been designated as a general purpose register.

**Figure 10. Register Pointer**
**RAM Protect (Z86C30 Only).** The upper portion of the RAM's address spaces %80 to %EF (excluding the control registers) can be protected from reading and writing. The RAM Protect bit option is mask-programmable and is selected by the customer when the ROM code is submitted. After the mask option is selected, the user can activate from the internal ROM code to turn off/on the RAM Protect by loading a bit D6 in the IMR register to either a 0 or a 1, respectively. A 1 in D6 indicates RAM Protect enabled.

**Stack.** An 8-bit Stack Pointer (R255) is used for the internal stack that resides within the 236/124 general-purpose registers.

**Counter/Timers.** There are two 8-bit programmable counter/timers (T0–T1), each driven by its own 6-bit programmable prescaler. The T1 prescaler can be driven by internal or external clock sources; however, the T0 prescaler is driven by the internal clock only (Figure 11).

---

**Figure 11. Counter/Timer Block Diagram**
The 6-bit prescalers can divide the input frequency of the clock source by any integer number from 1 to 64. Each prescaler drives its counter, which decrements the value (1 to 256) that has been loaded into the counter. When the counter reaches the end of count, a timer interrupt request, IRQ4 (T0) or IRQ5 (T1), is generated.

The counters can be programmed to start, stop, restart to continue, or restart from the initial value. The counters can also be programmed to stop upon reaching zero (single pass mode) or to automatically reload the initial value and continue counting (modulo-n continuous mode).

The counters, but not the prescalers, can be read at any time without disturbing their value or count mode. The clock source for T1 is user-definable and can be either the internal microprocessor clock divided by four, or an external signal input through Port 3. The Timer Mode register configures the external timer input (P31) as an external clock; a trigger input that can be retriggerable or not-retriggerable; or as a gate input for the internal clock. Port 3 line P36 serves as a timer output (TOUT) through which T0, T1 or the internal clock are output. The counter/timers can be cascaded by connecting the T0 output to the input of T1. T1 Mode is enabled by setting R243 PRE1 Bit D1 to 0.

**Interrupts.** The Z86C30/C31 has six different interrupts from six different sources. The interrupts are maskable and prioritized (Figure 12). The six sources are divided as follows: four sources are claimed by Port 3 lines P33-P30 and two in counter/timers. The Interrupt Mask Register globally or individually enables or disables the six interrupt requests (Table 4).

![Figure 12. Interrupt Block Diagram](image-url)
Table 4. Interrupt Types, Sources, and Vectors

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Vector Location</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRQ0</td>
<td>/DAV0, IRQ0</td>
<td>0, 1</td>
<td>External (P32), Rising/Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ1,</td>
<td>IRQ1</td>
<td>2, 3</td>
<td>External (P33), Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ2</td>
<td>/DAV2, IRQ2, T_N</td>
<td>4, 5</td>
<td>External (P31), Rising/Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ3</td>
<td>IRQ3</td>
<td>6, 7</td>
<td>External (P30), Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ4</td>
<td>T0</td>
<td>8, 9</td>
<td>Internal</td>
</tr>
<tr>
<td>IRQ5</td>
<td>T1</td>
<td>10, 11</td>
<td>Internal</td>
</tr>
</tbody>
</table>

When more than one interrupt is pending, priorities are resolved by a programmable priority encoder that is controlled by the Interrupt Priority register. An interrupt machine cycle is activated when an interrupt request is granted; it disables all subsequent interrupts, saves the Program Counter and Status Flags, and then branches to the program memory vector location reserved for that interrupt. All Z86C30/C31 interrupts are vectored through locations in the program memory. This memory location and the next byte contain the 16-bit starting address of the interrupt service routine for that particular interrupt request.

To accommodate polled interrupt systems, interrupt inputs are masked and the interrupt request register is polled to determine which of the interrupt requests need service.

An interrupt resulting from AN1 is mapped into IRQ2, and an interrupt from AN2 is mapped into IRQ0. Interrupts IRQ2 and IRQ0 may be rising, falling or both edge triggered, and are programmable by the user. The software can poll to identify the state of the pin.

Programming bits for the Interrupt Edge Select are located in the IRQ Register (R250), bits D7 and D6. The configuration is shown in Table 5.

Table 5. IRQ Register

<table>
<thead>
<tr>
<th>IRQ</th>
<th>Interrupt Edge</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D7</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Notes:
F = Falling Edge
R = Rising Edge
Clock. The Z86C30/C31 on-chip oscillator has a high-gain, parallel-resonant amplifier for connection to a crystal, RC, ceramic resonator, or any suitable external clock source (XTAL1 = Input, XTAL2 = Output). The crystal should be AT cut, 10 kHz to 12 MHz max., with a series resistance (RS) less than, or equal to, 100 Ohms. (Note: The Z86C31 is 8 MHz max.)

![Ceramic Resonator or Crystal](C1, C2 = 47 pf TYP)

![LC](C1, C2 = 22 pf)

L = 130 uH TYP

F = 3 MHz TYP

* Preliminary Value Including Pin Parasitics

The crystal should be connected across XTAL1 and XTAL2 using the vendor's recommended capacitors values from each pin directly to Ground, pin 22. This is to reduce ground noise injection. The RC oscillator option is mask-programmable, to be selected by the customer at the time ROM code is submitted. The RC oscillator configuration must be an external resistor connected from XTAL1 to XTAL2, with a frequency-setting capacitor from XTAL1 to Ground (Figure 13).

Power-On Reset (POR). A timer circuit clocked by a dedicated on-board RC oscillator is used for the Power-On Reset (POR) timer function. The POR timer allows $V_{CC}$ and the oscillator circuit to stabilize before instruction execution begins.

The POR timer circuit is a one-shot timer triggered by one of three conditions:

1. Power-fail to Power-OK status
2. Stop-Mode Recovery (if D5 of SMR=1)
3. WDT time-out

The POR time is $T_{POR}$. Bit 5 of the STOP mode register determines whether the POR timer is bypassed after Stop-Mode Recovery (typical for external clock, and RC/LC oscillators with fast start up time).

HALT. Turns off the internal CPU clock, but not the XTAL oscillation. The counter/timers and external interrupt IRQ0, IRQ1, IRQ2 and IRQ3, remain active. The device may be recovered by interrupts, either external or internal generated.

In order to enter STOP or HALT mode, it is necessary to first flush the instruction pipeline to avoid suspending execution in mid-instruction. To do this, the user must execute a NOP (opcode=FFH) immediately before the appropriate sleep instruction, i.e.:

```
FF NOP ; clear the pipeline
6F STOP ; enter STOP mode
or
FF NOP ; clear the pipeline
7F HALT ; enter HALT mode
```

STOP. This instruction turns off the internal clock and external crystal oscillation and reduces the standby current. The STOP mode is terminated by a RESET only, either by WDT time-out, POR; or SMR recovery. This causes the processor to restart the application program at address 000C (HEX).
Port Configuration Register (PCON). The Port Configuration Register (PCON) configures the ports individually: Comparator Output on Port 3, Open-Drain on Port 0, Low EMI Noise on Ports 0, 2, and 3, and Low EMI Noise Oscillator. The PCON Register is located in the Expanded Register File at bank F, location 00 (Figure 14).

Comparator Output Port 3 (D0). Bit 0 controls the comparator use in Port 3. A 1 in this location brings the comparator outputs to P34 and P37 and a 0 releases the Port to its standard I/O configuration.

Port 0 Open-Drain (D2). Port 0 is configured as an open-drain by resetting this bit (D2=0) and configured as Pull-up Active by setting D2 = 1. The default value is 1.

Low EMI Port 0 (D3). Port 0 is configured as a Low EMI Port by resetting this bit (D3=0) and configured as a Standard Port by setting D3=1. The default value is 1.

Low EMI Port 2 (D5). Port 2 is configured as a Low EMI Port by resetting this bit (D5=0) and configured as a Standard Port by setting D5=1. The default values is 1.

Low EMI Port 3 (D6). Port 3 is configured as a Low EMI Port by resetting this bit (D6=0) and configured as a Standard Port by setting D6=1. The default values is 1.

Low EMI OSC (D7). This bit of the PCON Register controls the low EMI noise oscillator. A 1 in this location configures the oscillator with standard drive, while a 0 configures the oscillator with low-noise drive, it does not affect the relationship of SCLK and XTAL.

Low EMI OSC (D7). This bit of the PCON Register controls the low EMI noise oscillator. A 1 in this location configures the oscillator with standard drive, while a 0 configures the oscillator with low-noise drive, it does not affect the relationship of SCLK and XTAL.
FUNCTIONAL DESCRIPTION (Continued)

STOP-Mode Recovery Register (SMR). This register selects the clock divide value and determines the mode of Stop-Mode Recovery (Figure 15). All bits are Write Only, except Bit 7 which is a Read Only. Bit 7 is a flag bit that is hardware set on the condition of STOP Recovery and reset by a power-on cycle. Bit 6 controls whether a low level or high level is required from the recovery source. Bit 5 controls the reset delay after recovery. Bits 2, 3, and 4 of the SMR register specify the source of the STOP-Mode Recovery signal. Bits 0 and 1 determine the time-out period of the WDT (Table 7). The SMR is located in bank F of the Expanded Register Group at address 0BH.

SCLK/TCLK Divide-by-16 Select (D0). D0 of the SMR controls a divide-by-16 prescaler of SCLK/TCLK. The purpose of this control is to selectively reduce device power consumption during normal processor execution (SCLK control) and/or HALT mode (TCLK sources, counter/timers, and interrupt logic). The default setting after either a Reset or a Stop-Mode Recovery is 0.

External Clock Divide-by-2 (D1). This bit can eliminate the oscillator divide-by-2 circuitry. When this bit is 0, SCLK (System Clock) and TCLK (Timer Clock) are equal to the external clock frequency divided by two. The SCLK/TCLK is equal to the external clock frequency when this bit is set (D1 = 1). Using this bit, together with D7 of PCON, further helps lower EMI [i.e., D7 (PCON) = 0, D1 (SMR) = 1]. The default setting is 0.

STOP-Mode Recovery Source (D2, D3, and D4). These three bits of the SMR specify the wake-up source of the STOP-Mode Recovery (Figure 16).

![Diagram of STOP-Mode Recovery Source](image-url)

Figure 16. STOP-Mode Recovery Source
Table 6. STOP-Mode Recovery Source

<table>
<thead>
<tr>
<th>SMR</th>
<th>D4 D3 D2</th>
<th>Operation</th>
<th>Description of action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0 0</td>
<td>POR recovery</td>
<td>only</td>
</tr>
<tr>
<td>0</td>
<td>0 0 1</td>
<td>P30 transition</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1 0 0</td>
<td>P31 transition</td>
<td>(Not in Analog Mode.)</td>
</tr>
<tr>
<td>0</td>
<td>1 0 1</td>
<td>P32 transition</td>
<td>(Not in Analog Mode.)</td>
</tr>
<tr>
<td>1</td>
<td>0 0 0</td>
<td>P33 transition</td>
<td>(Not in Analog Mode.)</td>
</tr>
<tr>
<td>1</td>
<td>0 1 0</td>
<td>Logical NOR of Port 2, bits 0-3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1 1 0</td>
<td>Logical NOR of Port 2, bits 0-7</td>
<td></td>
</tr>
</tbody>
</table>

STOP-Mode Recovery Delay Select (D5). This bit disables the 5 ms RESET delay after STOP-Mode Recovery. The default condition of this bit is 1. If the "fast" wake up is selected, the STOP-Mode Recovery source needs to be kept active for at least 5 TpC.

STOP-Mode Recovery Level Select (D6). A 1 in this bit position indicates that a high level on any one of the recovery sources wakes the Z86C30/C31 from STOP mode. A 0 indicates low level recovery. The default is 0 on POR (Figure 16).

Cold or Warm Start (D7). This bit is set by the device upon entering STOP mode. A 0 in this bit (cold) indicates that the device was reset by POR RESET. A 1 in this bit (warm) indicates that the device awakens by a SMR source.

Watch-Dog Timer Mode Register (WDTMR). The WDT is a retriggerable one-shot timer that will reset the Z8 if it reaches its terminal count. The WDT is initially enabled by executing the WDT instruction and refreshed on subsequent executions of the WDT instruction. The WDT cannot be disabled after it has been initially enabled. The WDT circuit is driven by an on-board RC oscillator or external oscillator from XTAL1 pin. The POR clock source is selected with bit 4 of the WDT register.

Note: Execution of the WDT instruction affects the Z (zero), S (sign), V (overflow) flags.

Table 7. Time-out Period of the WDT

<table>
<thead>
<tr>
<th>D1 D0</th>
<th>Time-out of Internal RC OSC</th>
<th>Time-out of XTAL clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>5 ms min</td>
<td>256 TpC</td>
</tr>
<tr>
<td>0 1</td>
<td>15 ms min</td>
<td>512 TpC</td>
</tr>
<tr>
<td>1 0</td>
<td>25 ms min</td>
<td>1024 TpC</td>
</tr>
<tr>
<td>1 1</td>
<td>100 ms min</td>
<td>4096 TpC</td>
</tr>
</tbody>
</table>

Notes:
TpC = XTAL clock cycle
The default on reset is 15 ms.
The values given are for $V_{cc} = 5.0V$

WDT During HALT (D2). This bit determines whether or not the WDT is active during HALT mode. A 1 indicates active during HALT. The default is 1.

WDT During STOP (D3). This bit determines whether or not the WDT is active during STOP mode. A 1 indicates active during STOP. A 0 will disable the WDT during STOP mode. Since the on-board OSC is stopped during STOP mode, the WDT clock source has to select the on-board RC OSC for the WDT to recover from STOP mode. The default is 1.

Clock Source for WDT (D4). This bit determines which oscillator source is used to clock the internal POR and WDT counter chain. If the bit is a 1, the internal RC oscillator is bypassed and the POR and WDT clock source is driven from the external pin, XTAL1. The default configuration of this bit is 0, which selects the RC oscillator.
FUNCTIONAL DESCRIPTION (Continued)

WDTMR Register Accessibility. The WDTMR register (Figure 17) is accessible only during the first 64 system clock cycles from the execution of the first instruction after Power-On Reset, Watch-Dog Timer reset or a STOP-Mode Recovery. After this point, the register cannot be modified by any means, intentional or otherwise. The WDTMR cannot be read and is located in Bank F of the Expanded Register Group at address location 0FH (Figure 18).

Note: The WDT can be permanently enabled through a mask programming option on the Z86C30/C31. This option is selected by the customer at the time of ROM code submittal. In this mode, WDT is always activated when the device comes out of reset. Execution of the WDT instruction serves to refresh the WDT time-out period. WDT operation in the HALT and STOP modes is controlled by WDTMR programming. If this mask option is not selected at the time of ROM code submission, the WDT must be activated by the user through the WDT instruction and is always disabled by any reset to the device.

Figure 17. Watch-Dog Timer Mode Register (Write Only)
Figure 18. Resets and WDT
FUNCTIONAL DESCRIPTION (Continued)

Low Voltage Protection. An on-board Voltage Comparator checks that $V_{CC}$ is at the required level to ensure correct operation of the device. Reset is globally driven if $V_{CC}$ is below the referenced Low Voltage Protection trip point voltage. The minimum operating voltage for functionality varies with temperature and operating frequency, while the Low Voltage Protection trip point voltage ($V_{LV}$) varies with temperature only.

The Low Voltage Protection trip voltage ($V_{LV}$) is less than 3 volts and above 1.4 volts under the following conditions.

Maximum ($V_{LV}$) Conditions:

Case 1: $T_A = -40^\circ C$ to $+105^\circ C$, Internal Clock (SCLK)
Frequency equal or less than 1 MHz

Case 2: $T_A = -40^\circ C$ to $+85^\circ C$, Internal Clock (SCLK)
Frequency equal to or less than 2 MHz

Note: The internal clock frequency (SCLK) is determined by SMR (F) 0B bit D1.

The Z86C30/C31 functions normally at or above 3.0V under all conditions. Below 3.0V, the devices are guaranteed to function normally until the Low Voltage Protection trip point ($V_{LV}$) is reached for the temperatures and operating frequencies in Case 1 and Case 2 above. The actual Low Voltage Protection trip point is a function of temperature and process parameters (Figure 19).

![Diagram showing $V_{CC}$ vs $T_A$ with $V_{LV}$ as a function of temperature with 1 MHz and 2 MHz lines.](attachment:Figure_19.png)

* Typical minimum operating $V_{CC}$ voltage at that frequency.
## ABSOLUTE MAXIMUM RATINGS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient Temperature under Bias</td>
<td>-40</td>
<td>+105</td>
<td>°C</td>
</tr>
<tr>
<td>Storage Temperature</td>
<td>-65</td>
<td>+150</td>
<td>°C</td>
</tr>
<tr>
<td>Voltage on any Pin with Respect to V&lt;sub&gt;SS&lt;/sub&gt; [Note 1]</td>
<td>-0.6</td>
<td>+7</td>
<td>V</td>
</tr>
<tr>
<td>Voltage on V&lt;sub&gt;DD&lt;/sub&gt; Pin with Respect to V&lt;sub&gt;SS&lt;/sub&gt;</td>
<td>-0.3</td>
<td>+7</td>
<td>V</td>
</tr>
<tr>
<td>Voltage on XTAL1 and /RESET Pins with Respect to V&lt;sub&gt;SS&lt;/sub&gt; [Note 2]</td>
<td>-0.6</td>
<td>V&lt;sub&gt;DD&lt;/sub&gt;+1</td>
<td>V</td>
</tr>
<tr>
<td>Total Power Dissipation</td>
<td>770</td>
<td></td>
<td>mW</td>
</tr>
<tr>
<td>Maximum Current out of V&lt;sub&gt;SS&lt;/sub&gt;</td>
<td>140</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td>Maximum Current into V&lt;sub&gt;DD&lt;/sub&gt;</td>
<td>125</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td>Maximum Current into an Input Pin [Note 3]</td>
<td>-600</td>
<td>+600</td>
<td>µA</td>
</tr>
<tr>
<td>Maximum Current into an Open-Drain Pin [Note 4]</td>
<td>-600</td>
<td>+600</td>
<td>µA</td>
</tr>
<tr>
<td>Maximum Output Current Sunked by Any I/O Pin</td>
<td>25</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td>Maximum Output Current Sourced by Any I/O Pin</td>
<td>25</td>
<td>mA</td>
<td></td>
</tr>
</tbody>
</table>

### Notes:

[1] This applies to all pins except XTAL pins and where otherwise noted.

[2] There is no input protection diode from pin to V<sub>DD</sub>.

[3] This excludes XTAL pins.

[4] Device pin is not at an output Low state.

## Notice:

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for an extended period may affect device reliability. Total power dissipation should not exceed 770 mW for the package. Power dissipation is calculated as follows:

\[
\text{Total Power Dissipation} = V_{\text{DD}} \times (I_{\text{DD}} - \text{sum of } I_{\text{OH}}) + \text{sum of } [(V_{\text{DD}} - V_{\text{OL}}) \times I_{\text{OH}}] + \text{sum of } (V_{\text{IL}} \times I_{\text{OL}})
\]

## STANDARD TEST CONDITIONS

The characteristics listed below apply for standard test conditions as noted. All voltages are referenced to Ground. Positive current flows into the referenced pin (Figure 20).

![Figure 20. Test Load Diagram](image)

### CAPACITANCE

\( T_A = 25^\circ C; \ V_{\text{CC}} = \text{GND} = 0V, \ f = 1.0 \text{ MHz}, \) unmeasured pins returned to GND.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input capacitance</td>
<td>0</td>
<td>12 pF</td>
</tr>
<tr>
<td>Output capacitance</td>
<td>0</td>
<td>12 pF</td>
</tr>
<tr>
<td>I/O capacitance</td>
<td>0</td>
<td>12 pF</td>
</tr>
</tbody>
</table>
### DC ELECTRICAL CHARACTERISTICS

#### Z86C30/C31

<table>
<thead>
<tr>
<th>Sym</th>
<th>Parameter</th>
<th>$V_H$</th>
<th>$V_L$</th>
<th>$V_{OH}$</th>
<th>$V_{OL}$</th>
<th>$V_{IH}$</th>
<th>$V_{IL}$</th>
<th>$V_{OH}$</th>
<th>$V_{OL}$</th>
<th>$V_{IL}$</th>
<th>$V_{OH}$</th>
<th>$V_{OL}$</th>
<th>$V_{IL}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$V_{CC}$ notation</td>
<td></td>
<td>3.0V</td>
<td>0.7 $V_{CC}$</td>
<td>0.2 $V_{CC}$</td>
<td>0.2 $V_{CC}$</td>
<td>0.2 $V_{CC}$</td>
<td>3.3V</td>
<td>0.7 $V_{CC}$</td>
<td>0.2 $V_{CC}$</td>
<td>3.3V</td>
<td>0.7 $V_{CC}$</td>
<td>0.2 $V_{CC}$</td>
</tr>
<tr>
<td></td>
<td>$V_{CC}$ to $+70^\circ C$</td>
<td></td>
<td>0.3</td>
<td>0.1</td>
<td>0.3</td>
<td>0.1</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>$V_{CC}$ to $+105^\circ C$</td>
<td></td>
<td>1.3</td>
<td>1.3</td>
<td>0.7</td>
<td>0.7</td>
<td>1.3</td>
<td>1.3</td>
<td>0.7</td>
<td>0.7</td>
<td>1.3</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>$@ 25^\circ C$</td>
<td></td>
<td>2.5</td>
<td>2.5</td>
<td>1.5</td>
<td>1.5</td>
<td>2.5</td>
<td>2.5</td>
<td>1.5</td>
<td>1.5</td>
<td>2.5</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td>Units</td>
<td></td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>Conditions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Notes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **$V_{CH}$**: Clock Input High Voltage
- **$V_{CL}$**: Clock Input Low Voltage
- **$V_{HI}$**: Input High Voltage
- **$V_{IL}$**: Input Low Voltage
- **$V_{OH}$**: Output High Voltage
- **$V_{OL}$**: Output Low Voltage
- **$V_{OH1}$**: Output High Voltage (Low EMI Mode)
- **$V_{OL1}$**: Output Low Voltage (Low EMI Mode)
- **$V_{OH2}$**: Output Low Voltage
- **$V_{IH}$**: Input High Voltage
- **$V_{IL}$**: Input Low Voltage
- **$V_{OH}$**: Output High Voltage
- **$V_{OL}$**: Output Low Voltage
- **$V_{OH1}$**: Output High Voltage (Low EMI Mode)
- **$V_{OL1}$**: Output Low Voltage (Low EMI Mode)
- **$V_{OH2}$**: Output Low Voltage
- **$V_{IH}$**: Input High Voltage
- **$V_{IL}$**: Input Low Voltage
- **$V_{OH}$**: Output High Voltage
- **$V_{OL}$**: Output Low Voltage
- **$V_{OH1}$**: Output High Voltage (Low EMI Mode)
- **$V_{OL1}$**: Output Low Voltage (Low EMI Mode)
- **$V_{OH2}$**: Output Low Voltage
- **$V_{IH}$**: Input High Voltage
- **$V_{IL}$**: Input Low Voltage
- **$V_{OH}$**: Output High Voltage
- **$V_{OL}$**: Output Low Voltage
<table>
<thead>
<tr>
<th>Sym</th>
<th>Parameter</th>
<th>$V_{cc}$ Note[3]</th>
<th>$T_A = 0^\circ C$ to $+70^\circ C$ Min Max</th>
<th>$T_A = -40^\circ C$ to $+105^\circ C$ Min Max</th>
<th>Typical @ 25$^\circ C$ Units</th>
<th>Conditions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HALT mode $V_{in} = OV, V_{cc} \Leftrightarrow 8 \text{ MHz}$ [4,5]</td>
<td></td>
</tr>
<tr>
<td>Icc1</td>
<td>Standby Current</td>
<td>3.0V</td>
<td>3 3 1 mA</td>
<td></td>
<td></td>
<td>HALT mode $V_{in} = OV, V_{cc} \Leftrightarrow 8 \text{ MHz}$ [4,5]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>5 5 2.4 mA</td>
<td></td>
<td></td>
<td>HALT mode $V_{in} = OV, V_{cc} \Leftrightarrow 12 \text{ MHz}$ [4,5]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>4 4 1.5 mA</td>
<td></td>
<td></td>
<td>HALT mode $V_{in} = OV, V_{cc} \Leftrightarrow 12 \text{ MHz}$ [4,5,15]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>6 6 3.2 mA</td>
<td></td>
<td></td>
<td>HALT mode $V_{in} = OV, V_{cc} \Leftrightarrow 12 \text{ MHz}$ [4,5,15]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>2 2 0.8 mA</td>
<td></td>
<td></td>
<td>Clock Divide by 16 @ 8 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>4 4 1.8 mA</td>
<td></td>
<td></td>
<td>Clock Divide by 16 @ 8 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>3 3 1.2 mA</td>
<td></td>
<td></td>
<td>Clock Divide by 16 @ 12 MHz</td>
<td>[4,5,15]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>5 5 2.5 mA</td>
<td></td>
<td></td>
<td>Clock Divide by 16 @ 12 MHz</td>
<td>[4,5,15]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>8 15 1 $\mu$A</td>
<td></td>
<td></td>
<td>STOP mode $V_{in} = OV, V_{cc}$</td>
<td>[6,11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>10 20 2 $\mu$A</td>
<td></td>
<td></td>
<td>STOP mode $V_{in} = OV, V_{cc}$</td>
<td>[6,11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>500 600 310 $\mu$A</td>
<td></td>
<td></td>
<td>STOP mode $V_{in} = OV, V_{cc}$</td>
<td>[6,11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>800 1000 600 $\mu$A</td>
<td></td>
<td></td>
<td>STOP mode $V_{in} = OV, V_{cc}$</td>
<td>[6,11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>8 25 16 $\mu$A</td>
<td></td>
<td></td>
<td>WDT is not Running</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>15 42 23 $\mu$A</td>
<td></td>
<td></td>
<td>WDT is not Running</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>-5 -18 -13 $\mu$A</td>
<td></td>
<td></td>
<td>WDT is Running</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>-8 -26 -17 $\mu$A</td>
<td></td>
<td></td>
<td>WDT is Running</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>8 25 16 $\mu$A</td>
<td></td>
<td></td>
<td>0V &lt; $V_{in} &lt; V_{cc}$</td>
<td>[9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>15 42 23 $\mu$A</td>
<td></td>
<td></td>
<td>0V &lt; $V_{in} &lt; V_{cc}$</td>
<td>[9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>-5 -18 -13 $\mu$A</td>
<td></td>
<td></td>
<td>0V &lt; $V_{in} &lt; V_{cc}$</td>
<td>[9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>-8 -26 -17 $\mu$A</td>
<td></td>
<td></td>
<td>0V &lt; $V_{in} &lt; V_{cc}$</td>
<td>[9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>7 24 7 25 8.5 ms</td>
<td></td>
<td></td>
<td>Power-On Reset</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>3 13 3 14 5 ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VCC</td>
<td>$V_{cc}$ Low Voltage</td>
<td>1.5 2.8</td>
<td>1.5 3.0 2.1 V</td>
<td></td>
<td></td>
<td>2 MHz max Int. CLK Freq.</td>
<td>[7]</td>
</tr>
<tr>
<td></td>
<td>Protection Voltage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes:
[1] $I_{CC1}$
Clock Driven 0.3 mA 5 mA 8 MHz
Crystal or Resonator 24 mA 5 mA 8 MHz

[2] GND = 0V.

[3] The $V_{cc}$ voltage specification of 3.0 guarantees 3.3V ±0.3 V and the $V_{cc}$ voltage specification of 5.5 V guarantees 5.0 V ±0.5 V.


[5] CL1 = CL2 = 100 pF


[7] The $V_L$ increases as the temperature decreases.

[8] Standard Mode (not Low EMI Mode)

[9] Auto Latch (mask option) selected

[10] For analog comparator inputs when analog comparators are enabled.

[11] Clock must be forced Low, when XTAL1 is clock-driven and XTAL2 is floating.


[13] Temperature is 0°C to +70°C.

[14] Auto Latch Delete option is not selected.

[15] Z86C30 only.

7-23
AC ELECTRICAL CHARACTERISTICS
Additional Timing Diagram

Figure 21. Additional Timing
### AC Electrical Characteristics

**Additional Timing Table (For SCLK/TCLK = XTAL/2)**

<table>
<thead>
<tr>
<th>No</th>
<th>Sym</th>
<th>Parameter</th>
<th>$V_{CC}$</th>
<th>$T_A = 0°C$ to $+70°C$</th>
<th>$T_A = -40°C$ to $+105°C$</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>8 MHz (C31)</td>
<td></td>
<td>12 MHz (C30)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TpC</td>
<td>Input Clock Period</td>
<td>3.0V</td>
<td>125 DC</td>
<td>83 DC</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>125 DC</td>
<td>83 DC</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td>2</td>
<td>TrC,TIC</td>
<td>Clock Input Rise and Fall Times</td>
<td>3.0V</td>
<td>25 DC</td>
<td>15 DC</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>25 DC</td>
<td>15 DC</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td>3</td>
<td>TwC</td>
<td>Input Clock Width</td>
<td>3.0V</td>
<td>62.5</td>
<td>62.5</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>62.5</td>
<td>62.5</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td>4</td>
<td>TwTinL</td>
<td>Timer Input Low Width</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td>5</td>
<td>TwTinH</td>
<td>Timer Input High Width</td>
<td>3.0V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td>6</td>
<td>TpTin</td>
<td>Timer Input Period</td>
<td>3.0V</td>
<td>8TpC</td>
<td>8TpC</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>8TpC</td>
<td>8TpC</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td>7</td>
<td>TrTin, Tffin</td>
<td>Timer Input Rise and Fall Timer</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td>8A</td>
<td>TwIL</td>
<td>Int. Request Low Time</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td>ns</td>
<td>[1,7,8]</td>
</tr>
<tr>
<td>8B</td>
<td>TwIL</td>
<td>Int. Request Low Time</td>
<td>3.0V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>ns</td>
<td>[1,3,7,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>ns</td>
<td>[1,3,7,8]</td>
</tr>
<tr>
<td>9</td>
<td>TwIH</td>
<td>Int. Request Input High Time</td>
<td>3.0V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>ns</td>
<td>[1,2,7,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>ns</td>
<td>[1,2,7,8]</td>
</tr>
<tr>
<td>10</td>
<td>Twsm</td>
<td>Stop-Mode Recovery Width Spec</td>
<td>3.0V</td>
<td>12</td>
<td>12</td>
<td>ns</td>
<td>[4,8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>12</td>
<td>12</td>
<td>ns</td>
<td>[4,8]</td>
</tr>
<tr>
<td>11</td>
<td>Tost</td>
<td>Oscillator Start-up Time</td>
<td>3.0V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>ns</td>
<td>[4,9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>ns</td>
<td>[4,9]</td>
</tr>
<tr>
<td>12</td>
<td>Twdt</td>
<td>Watch-Dog Timer Delay Time</td>
<td>3.0V</td>
<td>6.0</td>
<td>6.0</td>
<td>ms</td>
<td>D1=0[5,11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>3.0</td>
<td>3.0</td>
<td>ms</td>
<td>D1=0[5,11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.0V</td>
<td>20</td>
<td>20</td>
<td>ms</td>
<td>D0=0[5,11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>10</td>
<td>10</td>
<td>ms</td>
<td>D1=0[5,11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.0V</td>
<td>33</td>
<td>33</td>
<td>ms</td>
<td>D0=0[5,11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>16</td>
<td>16</td>
<td>ms</td>
<td>D1=0[5,11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.0V</td>
<td>132</td>
<td>132</td>
<td>ms</td>
<td>D0=0[5,11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>66</td>
<td>66</td>
<td>ms</td>
<td>D1=0[5,11]</td>
</tr>
</tbody>
</table>

**Notes:**

[1] Timing Reference uses 0.7 $V_{CC}$ for a logic 1 and 0.2 $V_{CC}$ for a logic 0.
[2] Interrupt request through Port 3 (P33-P30).
[3] Interrupt request through Port 3 (P30).
[5] Reg. WDTMR.
[6] The $V_{CC}$ voltage specification of 3.0 guarantees 3.3V ±0.3V and the $V_{CC}$ voltage specification of 5.5V guarantees 5.0V ±0.5V.
[7] SMR D1=0.
[8] Maximum frequency for internal system clock is 4 MHz when using XTAL divide-by-1 mode.
[9] For RC and LC oscillator, and for oscillator driven by clock driver.
[10] Using internal RC
### AC ELECTRICAL CHARACTERISTICS

**Additional Timing Table (Divide-By-One Mode)**

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>( V_{cc} )</th>
<th>( T_A = 0^\circ \text{C} ) to (+70^\circ \text{C} )</th>
<th>( T_A = -40^\circ \text{C} ) to (+105^\circ \text{C} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TpC</td>
<td>Input Clock Period</td>
<td>3.0V</td>
<td>250 DC</td>
<td>250 DC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>250 DC</td>
<td>250 DC</td>
</tr>
<tr>
<td>2</td>
<td>TrC,TfC</td>
<td>Clock Input Rise and Fall Times</td>
<td>3.0V</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>TwC</td>
<td>Input Clock Width</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>TwTinL</td>
<td>Timer Input Low Width</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>5</td>
<td>TwTinH</td>
<td>Timer Input High Width</td>
<td>3.0V</td>
<td>3Tpc</td>
<td>3Tpc</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>3Tpc</td>
<td>3Tpc</td>
</tr>
<tr>
<td>6</td>
<td>TpTin</td>
<td>Timer Input Period</td>
<td>3.0V</td>
<td>4Tpc</td>
<td>4Tpc</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>4Tpc</td>
<td>4Tpc</td>
</tr>
<tr>
<td>7</td>
<td>TrTin, Tfitin</td>
<td>Timer Input Rise and Fall Timer</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>8A</td>
<td>Twl</td>
<td>Int. Request Low Time</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>8B</td>
<td>Twl</td>
<td>Int. Request Low Time</td>
<td>3.0V</td>
<td>3Tpc</td>
<td>3Tpc</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>3Tpc</td>
<td>3Tpc</td>
</tr>
<tr>
<td>9</td>
<td>TwIH</td>
<td>Int. Request Input High Time</td>
<td>3.0V</td>
<td>3Tpc</td>
<td>3Tpc</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>3Tpc</td>
<td>2Tpc</td>
</tr>
<tr>
<td>10</td>
<td>Twsm</td>
<td>STOP-Mode Recovery Width Spec</td>
<td>3.0V</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>Tost</td>
<td>Oscillator Startup Time</td>
<td>3.0V</td>
<td>5Tpc</td>
<td>5Tpc</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5Tpc</td>
<td>5Tpc</td>
</tr>
</tbody>
</table>

**Notes:**

1. Timing Reference uses 0.7 \( V_{cc} \) for a logic 1 and 0.2 \( V_{cc} \) for a logic 0.
2. Interrupt request via Port 3 (P33-P31).
3. Interrupt request via Port 3 (P30).
4. SMR-D5 = 1, POR STOP mode delay is on.
5. Reg. WDTMR.
6. The \( V_{cc} \) voltage specification of 3.0 guarantees 3.3V ±0.3V and the \( V_{cc} \) voltage specification of 5.5V guarantees 5.0V ±0.5V.
7. SMR-D1 = 0.
8. Maximum frequency for internal system clock is 4 MHz when using XTAL divide-by-1 mode.
9. For RC and LC oscillator, and for oscillator driven by clock driver.
AC ELECTRICAL CHARACTERISTICS
Handshake Timing Table

Data In

<table>
<thead>
<tr>
<th>Data In Valid</th>
<th>Next Data In Valid</th>
</tr>
</thead>
</table>

/DAV (Input)

<table>
<thead>
<tr>
<th>Delayed DAV</th>
</tr>
</thead>
</table>

RDY (Output)

<table>
<thead>
<tr>
<th>Delayed RDY</th>
</tr>
</thead>
</table>

Figure 22. Input Handshake Timing

Data Out

<table>
<thead>
<tr>
<th>Data Out Valid</th>
<th>Next Data Out Valid</th>
</tr>
</thead>
</table>

/DAV (Output)

<table>
<thead>
<tr>
<th>Delayed DAV</th>
</tr>
</thead>
</table>

RDY (Input)

<table>
<thead>
<tr>
<th>Delayed RDY</th>
</tr>
</thead>
</table>

Figure 23. Output Handshake Timing
### AC ELECTRICAL CHARACTERISTICS (Continued)

**Handshake Timing Table**

<table>
<thead>
<tr>
<th>No</th>
<th>Sym</th>
<th>Parameter</th>
<th>$V_{cc}$ Note[1]</th>
<th>$T_A = 0°C$ to $+70°C$</th>
<th>$T_A = -40°C$ to $+105°C$</th>
<th>Data Direction</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$V_{cc}$</td>
<td>8 MHz (C31)</td>
<td>12 MHz (C30)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TsDi(DAV)</td>
<td>Data In Setup Time</td>
<td>3.0V</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>IN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>IN</td>
</tr>
<tr>
<td>2</td>
<td>ThDi(DAV)</td>
<td>Data In Hold Time</td>
<td>3.0V</td>
<td>160</td>
<td>160</td>
<td>160</td>
<td>IN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>115</td>
<td>115</td>
<td>115</td>
<td>IN</td>
</tr>
<tr>
<td>3</td>
<td>TwDAV</td>
<td>Data Available Width</td>
<td>3.0V</td>
<td>155</td>
<td>155</td>
<td>155</td>
<td>IN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>110</td>
<td>110</td>
<td>110</td>
<td>IN</td>
</tr>
<tr>
<td>4</td>
<td>TdDAV(RDY)</td>
<td>DAV Fall to RDY Fall Delay</td>
<td>3.0V</td>
<td>160</td>
<td>160</td>
<td>160</td>
<td>IN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>115</td>
<td>115</td>
<td>115</td>
<td>IN</td>
</tr>
<tr>
<td>5</td>
<td>TdDAV(RDY)</td>
<td>DAV Rise to RDY Rise Delay</td>
<td>3.0V</td>
<td>120</td>
<td>120</td>
<td>120</td>
<td>IN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>IN</td>
</tr>
<tr>
<td>6</td>
<td>RDY0d(DAV)</td>
<td>RDY Rise to DAV Fall Delay</td>
<td>3.0V</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>IN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>IN</td>
</tr>
<tr>
<td>7</td>
<td>TdDO(DAV)</td>
<td>Data Out to DAV Fall Delay</td>
<td>3.0V</td>
<td>63</td>
<td>42</td>
<td>42</td>
<td>OUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>63</td>
<td>42</td>
<td>42</td>
<td>OUT</td>
</tr>
<tr>
<td>8</td>
<td>TdDAVO(RDY)</td>
<td>DAV Fall to RDY Fall Delay</td>
<td>3.0V</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OUT</td>
</tr>
<tr>
<td>9</td>
<td>TdRDY0(DAV)</td>
<td>RDY Fall to DAV Rise Delay</td>
<td>3.0V</td>
<td>160</td>
<td>160</td>
<td>160</td>
<td>OUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>115</td>
<td>115</td>
<td>115</td>
<td>OUT</td>
</tr>
<tr>
<td>10</td>
<td>TwRDY</td>
<td>RDY Width</td>
<td>3.0V</td>
<td>110</td>
<td>110</td>
<td>110</td>
<td>OUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>OUT</td>
</tr>
<tr>
<td>11</td>
<td>TdRDY0d(DAV)</td>
<td>RDY Rise to DAV Fall Delay</td>
<td>3.0V</td>
<td>110</td>
<td>110</td>
<td>110</td>
<td>OUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>OUT</td>
</tr>
</tbody>
</table>

**Notes:**

[1] The $V_{cc}$ voltage specification of 3.0 guarantees 3.3V ±0.3V and the $V_{cc}$ voltage specification of 5.5V guarantees 5.0V ±0.5V.


[3] Z86C31 max. freq. = 8 MHz; Z86C30 max. freq. = 12 MHz.
EXPANDED REGISTER FILE CONTROL REGISTERS

SMR (F) 08

- SCLK/TCLK Divide by 16
  0: OFF
  1: ON
- External Clock Divide by 2
  0: SCLK/TCLK = XTAL/2
  1: SCLK/TCLK = XTAL
- Stop-Mode Recovery Source
  - POR Only
  - POR
  - P30
  - P31
  - P32
  - P33
  - P27
  - P2 NOR 0/3
  - P2 NOR 0/7
- Stop Delay
  0: OFF
  1: ON
- Stop Recovery Level
  0: Low
  1: High
- Stop Flag
  0: POR
  1: Stop Recovery

* Default setting after RESET
† Default setting after reset and Stop-Mode Recovery.

WDTMR (F) 0F

- WDT TAP
  0: INT RC OSC
  1: XTAL CLK
- XTAL/INT RC Select for WDT
  0: On-Board RC
  1: XTAL

Stop-Moc:le Recovery Source

- POR Only
- POR
- P30
- P31
- P32
- P33
- P27
- P2 NOR 0/3
- P2 NOR 0/7

Stop Delay

- 0: OFF
- 1: ON

Stop Recovery Level

- 0: Low
- 1: High

Stop Flag

- 0: POR
- 1: Stop Recovery

Figure 24. Stop-Mode Recovery Register
(Write only Except Bit D7 Which is Read Only)

Figure 25. Watch-Dog Timer Mode Register
(Write Only)

Figure 26. Port Configuration Register
(Write Only)
Z8® CONTROL REGISTER DIAGRAMS

**Figure 27. Reserved**

**Figure 28. Timer Mode Register (F1H: Read/Write)**

**Figure 29. Counter Timer 1 Register (F2H: Read/Write)**

**Figure 30. Prescaler 1 Register (F3H: Write Only)**

**Figure 31. Counter/Timer 0 Register (F4H: Read/Write)**

**Figure 32. Prescaler 0 Register (F5H: Write Only)**

**Figure 33. Port 2 Mode Register (F6H: Write Only)**
Z8® CONTROL REGISTER DIAGRAMS (Continued)

Figure 34. Port 3 Mode Register (F7H: Write Only)

Figure 35. Port 0 and 1 Mode Register (F8H: Write Only)

Figure 36. Interrupt Priority Register (F9H: Write Only)
Z8® CONTROL REGISTER DIAGRAMS (Continued)

**Figure 37. Interrupt Request Register**

(FA<sub>H</sub>: Read/Write)

**Figure 38. Interrupt Mask Register**

(FB<sub>H</sub>: Read/Write)

**Figure 39. Flag Register**

(FC<sub>H</sub>: Read/Write)

**Figure 40. Register Pointer**

(FD<sub>H</sub>: Read/Write)

**Figure 41. General-Purpose Register**

**Figure 42. Stack Pointer**

(FF<sub>H</sub>: Read/Write)
INSTRUCTION SET NOTATION

Addressing Modes. The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRR</td>
<td>Indirect register pair or indirect working-register pair address</td>
</tr>
<tr>
<td>Irr</td>
<td>Indirect working-register pair only</td>
</tr>
<tr>
<td>X</td>
<td>Indexed address</td>
</tr>
<tr>
<td>DA</td>
<td>Direct address</td>
</tr>
<tr>
<td>RA</td>
<td>Relative address</td>
</tr>
<tr>
<td>IM</td>
<td>Immediate</td>
</tr>
<tr>
<td>R</td>
<td>Register or working-register address</td>
</tr>
<tr>
<td>r</td>
<td>Working-register address only</td>
</tr>
<tr>
<td>IR</td>
<td>Indirect-register or indirect working-register address</td>
</tr>
<tr>
<td>Ir</td>
<td>Indirect working-register address only</td>
</tr>
<tr>
<td>RR</td>
<td>Register pair or working register pair address</td>
</tr>
</tbody>
</table>

Symbols. The following symbols are used in describing the instruction set.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst</td>
<td>Destination location or contents</td>
</tr>
<tr>
<td>src</td>
<td>Source location or contents</td>
</tr>
<tr>
<td>cc</td>
<td>Condition code</td>
</tr>
<tr>
<td>@</td>
<td>Indirect address prefix</td>
</tr>
<tr>
<td>SP</td>
<td>Stack Pointer</td>
</tr>
<tr>
<td>PC</td>
<td>Program Counter</td>
</tr>
<tr>
<td>FLAGS</td>
<td>Flag register (Control Register 252)</td>
</tr>
<tr>
<td>RP</td>
<td>Register Pointer (R253)</td>
</tr>
<tr>
<td>IMR</td>
<td>Interrupt mask register (R251)</td>
</tr>
</tbody>
</table>

Flags. Control register (R252) contains the following six flags:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Carry flag</td>
</tr>
<tr>
<td>Z</td>
<td>Zero flag</td>
</tr>
<tr>
<td>S</td>
<td>Sign flag</td>
</tr>
<tr>
<td>V</td>
<td>Overflow flag</td>
</tr>
<tr>
<td>D</td>
<td>Decimal-adjust flag</td>
</tr>
<tr>
<td>H</td>
<td>Half-carry flag</td>
</tr>
</tbody>
</table>

Affected flags are indicated by:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Clear to zero</td>
</tr>
<tr>
<td>1</td>
<td>Set to one</td>
</tr>
<tr>
<td>*</td>
<td>Set to clear according to operation</td>
</tr>
<tr>
<td>-</td>
<td>Unaffected</td>
</tr>
<tr>
<td>x</td>
<td>Undefined</td>
</tr>
</tbody>
</table>
## CONDITION CODES

<table>
<thead>
<tr>
<th>Value</th>
<th>Mnemonic</th>
<th>Meaning</th>
<th>Flags Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td></td>
<td>Always True</td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>C</td>
<td>Carry</td>
<td>C = 1</td>
</tr>
<tr>
<td>1111</td>
<td>NC</td>
<td>No Carry</td>
<td>C = 0</td>
</tr>
<tr>
<td>0110</td>
<td>Z</td>
<td>Zero</td>
<td>Z = 1</td>
</tr>
<tr>
<td>1110</td>
<td>NZ</td>
<td>Not Zero</td>
<td>Z = 0</td>
</tr>
<tr>
<td>1101</td>
<td>PL</td>
<td>Plus</td>
<td>S = 0</td>
</tr>
<tr>
<td>0101</td>
<td>MI</td>
<td>Minus</td>
<td>S = 1</td>
</tr>
<tr>
<td>0100</td>
<td>OV</td>
<td>Overflow</td>
<td>V = 1</td>
</tr>
<tr>
<td>1100</td>
<td>NOV</td>
<td>No Overflow</td>
<td>V = 0</td>
</tr>
<tr>
<td>0110</td>
<td>EQ</td>
<td>Equal</td>
<td>Z = 1</td>
</tr>
<tr>
<td>1110</td>
<td>NE</td>
<td>Not Equal</td>
<td>Z = 0</td>
</tr>
<tr>
<td>1001</td>
<td>GE</td>
<td>Greater Than or Equal</td>
<td>(S XOR V) = 0</td>
</tr>
<tr>
<td>0001</td>
<td>LT</td>
<td>Less than</td>
<td>(S XOR V) = 1</td>
</tr>
<tr>
<td>1010</td>
<td>GT</td>
<td>Greater Than</td>
<td>[Z OR (S XOR V)] = 0</td>
</tr>
<tr>
<td>0010</td>
<td>LE</td>
<td>Less Than or Equal</td>
<td>[Z OR (S XOR V)] = 1</td>
</tr>
<tr>
<td>1111</td>
<td>UGE</td>
<td>Unsigned Greater Than or Equal</td>
<td>C = 0</td>
</tr>
<tr>
<td>0111</td>
<td>ULT</td>
<td>Unsigned Less Than</td>
<td>C = 1</td>
</tr>
<tr>
<td>1011</td>
<td>UGT</td>
<td>Unsigned Greater Than</td>
<td>(C = 0 AND Z = 0) = 1</td>
</tr>
<tr>
<td>0011</td>
<td>ULE</td>
<td>Unsigned Less Than or Equal</td>
<td>(C OR Z) = 1</td>
</tr>
<tr>
<td>0000</td>
<td>F</td>
<td>Never True (Always False)</td>
<td></td>
</tr>
</tbody>
</table>
### INSTRUCTION FORMATS

**One-Byte Instructions**

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>SRC</th>
<th>DST</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>\text{OPC}</td>
<td>\text{MODE}</td>
<td>\text{SRC}</td>
</tr>
<tr>
<td>\text{OPC}</td>
<td>\text{1110}</td>
<td>\text{OPC}</td>
<td>\text{MODE}</td>
</tr>
<tr>
<td>\text{OPC}</td>
<td>\text{1110}</td>
<td>\text{OPC}</td>
<td>\text{MODE}</td>
</tr>
<tr>
<td>\text{OPC}</td>
<td>\text{1110}</td>
<td>\text{OPC}</td>
<td>\text{MODE}</td>
</tr>
<tr>
<td>\text{OPC}</td>
<td>\text{1110}</td>
<td>\text{OPC}</td>
<td>\text{MODE}</td>
</tr>
</tbody>
</table>

**Two-Byte Instructions**

<table>
<thead>
<tr>
<th>OPC</th>
<th>VALUE</th>
<th>SRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>\text{VALUE}</td>
<td>\text{SRC}</td>
</tr>
<tr>
<td>OPC</td>
<td>\text{VALUE}</td>
<td>\text{SRC}</td>
</tr>
</tbody>
</table>

**Three-Byte Instructions**

<table>
<thead>
<tr>
<th>OPC</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC</td>
<td>\text{VALUE}</td>
</tr>
</tbody>
</table>

### INSTRUCTION SUMMARY

**Note:** Assignment of a value is indicated by the symbol "\( \leftarrow \). For example:

\[
\text{dst} \leftarrow \text{dst} + \text{src}
\]

indicates that the source data is added to the destination data and the result is stored in the destination location.

The notation "addr (n)" is used to refer to bit (n) of a given operand location. For example:

\[
\text{dst} (7)
\]

refers to bit 7 of the destination operand.
### INSTRUCTION SUMMARY (Continued)

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ADC dst, src</strong></td>
<td>†</td>
<td>1[ ]</td>
<td>* * * * 0 *</td>
</tr>
<tr>
<td>dst←dst + src + C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ADD dst, src</strong></td>
<td>†</td>
<td>0[ ]</td>
<td>* * * * 0 *</td>
</tr>
<tr>
<td>dst←dst + src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AND dst, src</strong></td>
<td>†</td>
<td>5[ ]</td>
<td>- * * 0 -</td>
</tr>
<tr>
<td>dst←dst AND src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CALL dst</strong></td>
<td>DA</td>
<td>D6</td>
<td></td>
</tr>
<tr>
<td>SP←SP - 2</td>
<td>IRR</td>
<td>D4</td>
<td></td>
</tr>
<tr>
<td>@SP←PC,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CCF</strong></td>
<td>EF</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>C←NOT C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CLR dst</strong></td>
<td>R</td>
<td>B0</td>
<td></td>
</tr>
<tr>
<td>dst←0</td>
<td>IR</td>
<td>B1</td>
<td></td>
</tr>
<tr>
<td><strong>COM dst</strong></td>
<td>R</td>
<td>60</td>
<td>- * * 0 -</td>
</tr>
<tr>
<td>dst←NOT dst</td>
<td>IR</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td><strong>CP dst, src</strong></td>
<td>†</td>
<td>A[ ]</td>
<td>* * * * -</td>
</tr>
<tr>
<td>dst←src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DA dst</strong></td>
<td>R</td>
<td>40</td>
<td>* * * X -</td>
</tr>
<tr>
<td>dst←DA dst</td>
<td>IR</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td><strong>DEC dst</strong></td>
<td>R</td>
<td>00</td>
<td>- * * * -</td>
</tr>
<tr>
<td>dst←dst - 1</td>
<td>IR</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td><strong>DECW dst</strong></td>
<td>RR</td>
<td>80</td>
<td>- * * * -</td>
</tr>
<tr>
<td>dst←dst - 1</td>
<td>IR</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td><strong>DI</strong></td>
<td></td>
<td>8F</td>
<td></td>
</tr>
<tr>
<td>IMR(7)←0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DJNZr, dst</strong></td>
<td>RA</td>
<td>rA</td>
<td>-</td>
</tr>
<tr>
<td>r←r - 1</td>
<td>r</td>
<td>0 - F</td>
<td></td>
</tr>
<tr>
<td>if r ≠ 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←PC + dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range: +127, -128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>EI</strong></td>
<td></td>
<td>9F</td>
<td>-</td>
</tr>
<tr>
<td>IMR(7)←1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HALT</strong></td>
<td></td>
<td>7F</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INC dst</strong></td>
<td>r</td>
<td>rE</td>
<td>- * * * -</td>
</tr>
<tr>
<td>dst←dst + 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>INCW dst</strong></td>
<td>RR</td>
<td>A0</td>
<td>- * * * -</td>
</tr>
<tr>
<td>dst←dst + 1</td>
<td>IR</td>
<td>A1</td>
<td></td>
</tr>
<tr>
<td><strong>IRET</strong></td>
<td>BF</td>
<td>* * * * *</td>
<td></td>
</tr>
<tr>
<td>FLAGS←@SP;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP←SP + 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←@SP;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP←SP + 2;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMR(7)←1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>JP cc, dst</strong></td>
<td>DA</td>
<td>cD</td>
<td>-</td>
</tr>
<tr>
<td>if cc is true</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←dst</td>
<td>IRR</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>if cc is true</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>JR cc, dst</strong></td>
<td>RA</td>
<td>cB</td>
<td>-</td>
</tr>
<tr>
<td>if cc is true,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←PC + dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range: +127, -128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LD dst, src</strong></td>
<td>r</td>
<td>rIm</td>
<td>-</td>
</tr>
<tr>
<td>dst←src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LDC dst, src</strong></td>
<td>r</td>
<td>Irr</td>
<td>-</td>
</tr>
<tr>
<td>dst←src</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7-36
<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOP</td>
<td></td>
<td>FF</td>
<td></td>
</tr>
<tr>
<td>OR dst, src</td>
<td>R</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>POP dst</td>
<td>R</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>PUSH src</td>
<td>R</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>RCF</td>
<td></td>
<td>CF</td>
<td></td>
</tr>
<tr>
<td>RET</td>
<td></td>
<td>AF</td>
<td></td>
</tr>
<tr>
<td>RL dst</td>
<td>R</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>RLC dst</td>
<td>R</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>RR dst</td>
<td>R</td>
<td>E0</td>
<td></td>
</tr>
<tr>
<td>RRC dst</td>
<td>R</td>
<td>C0</td>
<td></td>
</tr>
<tr>
<td>SBC dst, src</td>
<td>R</td>
<td>D0</td>
<td></td>
</tr>
<tr>
<td>SCF</td>
<td></td>
<td>CF</td>
<td></td>
</tr>
<tr>
<td>SRA dst</td>
<td>R</td>
<td>D0</td>
<td></td>
</tr>
<tr>
<td>SRP src</td>
<td></td>
<td>Im</td>
<td></td>
</tr>
</tbody>
</table>

The table continues with similar entries for other instructions, each with its own address mode, opcode byte, and flags affected.

Instruction and Operation | Address Mode | Opcode Byte (Hex) | Flags Affected |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>STOP</td>
<td></td>
<td>6F</td>
<td></td>
</tr>
<tr>
<td>SUB dst, src</td>
<td>R</td>
<td>5F</td>
<td></td>
</tr>
<tr>
<td>SWAP dst</td>
<td>R</td>
<td>F0</td>
<td></td>
</tr>
<tr>
<td>TCM dst, src</td>
<td>R</td>
<td>7F</td>
<td></td>
</tr>
<tr>
<td>XOR dst, src</td>
<td>R</td>
<td>B0</td>
<td></td>
</tr>
<tr>
<td>WDT</td>
<td></td>
<td>5F</td>
<td></td>
</tr>
</tbody>
</table>

For example, the opcode of an ADC instruction using the addressing modes r (destination) and Ir (source) is 13.
### OPCODE MAP

<table>
<thead>
<tr>
<th>Upper Nibble (Hex)</th>
<th>Lower Nibble (Hex)</th>
<th>Bytes per Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>6.5</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>6.5</td>
<td>1</td>
</tr>
</tbody>
</table>

### Legend:
- **R** = 8-bit address
- **r** = 4-bit address
- **R1** or **r1** = Dest address
- **R2** or **r2** = Src address

### Sequence:
- Opcode, First Operand, Second Operand

### Note:
- The blanks are reserved.
- *2-byte instruction appears as a 3-byte instruction
PACKAGE INFORMATION

28-Pin DIP Package Diagram

28-Pin SOIC Package Diagram
NOTES:
1. DIMENSIONAL TOLERANCE ± .001 UNLESS SPECIFIED
2. ALL DIMENSIONS IN INCH
ORDERING INFORMATION

Z86C30 (12 MHz)
Standard Temperature
28-Pin DIP
Z86C3012PSC

Extended Temperature
28-Pin DIP
Z86C3012PEC

Standard Temperature
28-Pin SOIC
Z86C3012SSC

Extended Temperature
28-Pin SOIC
Z86C3012SEC

Standard Temperature
28-Pin PCB Chip Carrier
Z86C3012TSC

Extended Temperature
28-Pin PCB Chip Carrier
Z86C3012TEC

Z86C31 (8 MHz)
Standard Temperature
28-Pin DIP
Z86C3108PSC

Extended Temperature
28-Pin DIP
Z86C3108PEC

Standard Temperature
28-Pin SOIC
Z86C3108SSC

Extended Temperature
28-Pin SOIC
Z86C3108SEC

Standard Temperature
28-Pin PCB Chip Carrier
Z86C3108TSC

Extended Temperature
28-Pin PCB Chip Carrier
Z86C3108TEC

For fast results, contact your local Zilog sales office for assistance in ordering the part desired.

CODES

Preferred Package
P = Plastic DIP

Longer Lead Time
S = SOIC
T = PCB Chip Carrier

Preferred Temperature
S = 0°C to +70°C

Longer Lead Time
E = -40°C to +105°C

Speeds
08 = 8 MHz
12 = 12 MHz

Environmental
C = Plastic Standard

Example:
Z 86C30 12 P S C

is a Z86C30, 12 MHz, DIP, 0°C to +70°C, Plastic Standard Flow

Environmental Flow
Temperature
Package
Speed
Product Number
Zilog Prefix
FEATURES

- The Z86E30 and Z86E31 have the following general characteristics:

<table>
<thead>
<tr>
<th>Part</th>
<th>EPROM</th>
<th>RAM</th>
<th>Speeds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z86E30</td>
<td>4K</td>
<td>236</td>
<td>12 MHz</td>
</tr>
<tr>
<td>Z86E31</td>
<td>2K</td>
<td>124</td>
<td>8 MHz</td>
</tr>
</tbody>
</table>

- 28-pin packages (DIP, Cerdip Window Lid)
- 4.5V to 5.5V operating range
- Clock speeds up to 8 MHz (E31) and 12 MHz (E30)
- Software programmable low EMI mode
- Pull-up active/open-drain programmable on ports 0 and 2
- EPROM protect option
- RAM protect programmable
- RC oscillator programmable
- Low power consumption: 60 mW

GENERAL DESCRIPTION

The Z86E30/E31 CCP™ (Consumer Controller Processors) are members of Zilog's Z8® single-chip microcontroller family with enhanced wake-up circuitry, programmable watch-dog timers and low noise/EMI options. These enhancements result in a more efficient, cost-effective design and provide the user with increased design flexibility over the standard Z8 microcontroller core. With 4K/2K bytes of EPROM and 236/124 bytes of general-purpose RAM, respectively, these low cost, low power consumption CMOS microcontrollers offer fast execution, efficient use of memory, sophisticated interrupts, input/output bit manipulation capabilities, and easy hardware/software system expansion.

Manufactured in CMOS technology and offered in 28-pin DIP and Cerdip Window Lid package styles, these devices allow easy software development and debug, prototyping, and small production runs not economically desirable with a masked ROM version.

For applications demanding powerful I/O capabilities, the Z86E30/E31 provides 24 pins dedicated to input and output. These lines are grouped into three ports, eight lines per port, and are configurable under software control to provide timing, status signals, and parallel I/O with or without handshake.
GENERAL DESCRIPTION (Continued)

Three basic address spaces are available to support this wide range of configurations: Program Memory, Register File, and Expanded Register File (ERF). The Register File is composed of 236/124 bytes of general-purpose registers, three I/O port registers and 15 control and status registers. The Expanded Register File consists of three control registers.

To unburden the system from coping with the real-time tasks such as counting/timing and input/output data communication, the Z86E30/31 offers two on-chip counter/timers with a large number of user-selectable modes, and two on-board comparators to process analog signals with a common reference voltage (Figures 1 and 2).

Notes:
All Signals with a preceding front slash, \(^{/}\), are active Low, e.g.: B/W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

<table>
<thead>
<tr>
<th>Connection</th>
<th>Circuit</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>(V_{CC} )</td>
<td>(V_{DD} )</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
<td>(V_{SS} )</td>
</tr>
</tbody>
</table>

---

Figure 1. Z86E30/E31 Functional Block Diagram
Figure 2. Z86E30/E31 EPROM Programming Block Diagram
### PIN DESCRIPTION

#### Table 1. Z86E30/E31 28-Pin DIP Pin Identification*

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>P25-P27</td>
<td>Port 2, Pins 5,6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>4-7</td>
<td>P04-P07</td>
<td>Port 0, Pins 4,5,6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>8</td>
<td>VCC</td>
<td>Power Supply</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>XTAL2</td>
<td>Crystal Oscillator</td>
<td>Output</td>
</tr>
<tr>
<td>10</td>
<td>XTAL1</td>
<td>Crystal Oscillator</td>
<td>Input</td>
</tr>
<tr>
<td>11-13</td>
<td>P31-P33</td>
<td>Port 3, Pins 1,2,3</td>
<td>Input</td>
</tr>
<tr>
<td>14-15</td>
<td>P34-P35</td>
<td>Port 3, Pins 4,5</td>
<td>Output</td>
</tr>
<tr>
<td>16</td>
<td>P37</td>
<td>Port 3, Pin 7</td>
<td>Output</td>
</tr>
<tr>
<td>17</td>
<td>P36</td>
<td>Port 3, Pin 6</td>
<td>Output</td>
</tr>
<tr>
<td>18</td>
<td>P30</td>
<td>Port 3, Pin 0</td>
<td>Input</td>
</tr>
<tr>
<td>19-21</td>
<td>P00-P02</td>
<td>Port 0, Pins 0,1,2</td>
<td>In/Output</td>
</tr>
<tr>
<td>22</td>
<td>VSS</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>P03</td>
<td>Port 0, Pin 3</td>
<td>In/Output</td>
</tr>
<tr>
<td>24-28</td>
<td>P20-P24</td>
<td>Port 2, Pins 0,1,2,3,4</td>
<td>In/Output</td>
</tr>
</tbody>
</table>

**Note:**
* Pin Identification and Configuration identical on DIP and Cerdip Window Lid style packages.

#### Table 2. Z86E30/E31 28-Pin DIP Pin Identification*

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>D5-D7</td>
<td>Data 5,6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>4-7</td>
<td>A4-A7</td>
<td>Address 4,5,6,7</td>
<td>Input</td>
</tr>
<tr>
<td>8</td>
<td>VPP</td>
<td>Prog. Voltage</td>
<td>Input</td>
</tr>
<tr>
<td>9</td>
<td>NC</td>
<td>No connection</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>/CE</td>
<td>Chip Select</td>
<td>Input</td>
</tr>
<tr>
<td>11</td>
<td>/OE</td>
<td>Output Enable</td>
<td>Input</td>
</tr>
<tr>
<td>12</td>
<td>EPM</td>
<td>EPROM Prog. Mode</td>
<td>Input</td>
</tr>
<tr>
<td>13</td>
<td>VPP</td>
<td>Prog. Voltage</td>
<td>Input</td>
</tr>
<tr>
<td>14-15</td>
<td>A8-A9</td>
<td>Address 8,9</td>
<td>Input</td>
</tr>
<tr>
<td>16</td>
<td>A11</td>
<td>Address 11</td>
<td>Input</td>
</tr>
<tr>
<td>17</td>
<td>A10</td>
<td>Address 10</td>
<td>Input</td>
</tr>
<tr>
<td>18</td>
<td>/PGM</td>
<td>Prog. Mode</td>
<td>Input</td>
</tr>
<tr>
<td>19-21</td>
<td>A0-A2</td>
<td>Address 0,1,2</td>
<td>Input</td>
</tr>
<tr>
<td>22</td>
<td>VSS</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>A3</td>
<td>Address 3</td>
<td>Input</td>
</tr>
<tr>
<td>24-28</td>
<td>D0-D4</td>
<td>Data 0,1,2,3,4</td>
<td>In/Output</td>
</tr>
</tbody>
</table>

**Note:**
* Pin Identification and Configuration identical on DIP and Cerdip Window Lid style packages.

---

**Figure 3. Z86E30/31 Standard Mode 28-Pin DIP Pin Configuration***

**Figure 4. Z86E30/31 EPROM Programming Mode 28-Pin DIP Pin Configuration***
PIN FUNCTIONS

Z86E30/31 Standard Mode

XTAL1 Crystal 1 (time-based input). This pin connects a parallel-resonant crystal, ceramic resonator, LC, RC network or external single-phase clock to the on-chip oscillator input.

XTAL2 Crystal 2 (time-based output). This pin connects a parallel-resonant crystal, ceramic resonator, LC or RC network to the on-chip oscillator output.

Port 0 (P07-P00). Port 0 is an 8-bit, bi-directional, CMOS compatible I/O port. These eight I/O lines can be nibble programmed as P03-P00 input/output and P07-P04 input/output, separately. The input buffers are Schmitt-triggered and nibbles programmed as outputs can be globally programmed as either push-pull or open-drain. Low EMI output buffers can be globally programmed by the software. Port 0 can also be used as a handshake I/O port.

In Handshake Mode, Port 3 lines P32 and P35 are used as handshake control lines. The handshake direction is determined by the configuration (input or output) assigned to Port 0’s upper nibble. The lower nibble must have the same direction as the upper nibble (Figure 5).

EPROM Programming Mode

D7-D0 Data Bus. The data can be read from or written to the EPROM through the data bus.

A11-A0 Address Bus. During programming, the EPROM address is written to the address bus.

Vcc Power Supply. This pin must be supply 5V during the EPROM Read Mode and 6V during other modes.

/CE Chip Enable (active Low). This pin is active during EPROM Read Mode, Program Mode and Program Verify Mode.

/OE Output Enable (active Low). This pin drives the direction of the Data Bus. When this pin is Low, the Data Bus is output, when High, the Data Bus is input.

EPM EPROM Program Mode. This pin controls the different EPROM Program Mode by applying different voltages.

Vpp Program Voltage. This pin supplies the program voltage.

/PGM Program Mode (active Low). When this pin is Low, the data is programmed to the EPROM through the Data Bus.

Application Precaution

The production test-mode environment may be enabled accidentally during normal operation if excessive noise surges above Vcc occur on the XTAL1 pin.

In addition, processor operation of Z80TP devices may be affected by excessive noise surges on the Vpp, /CE, /EPM, /OE pins while the microcontroller is in Standard Mode.

Recommendations for dampening voltage surges in both test and OTP mode include the following:

- Using a clamping diode to Vcc
- Adding a capacitor to the affected pin.
PIN FUNCTIONS (Continued)

Z86E30/E31 CMOS 8-Bit Z8® CCP™
CONSUMER CONTROLLER PROCESSORS

Figure 5. Port 0 Configuration
Port 2 (P27-P20). Port 2 is an 8-bit, bi-directional, CMOS compatible I/O port. These eight I/O lines can be configured under software control as an input or output, independently. All input buffers are Schmitt-triggered. Bits programmed as outputs can be globally programmed as either push-pull or open-drain. Low EMI output buffers can be globally programmed by the software. When used as an I/O port, Port 2 can be placed under handshake control. In Handshake Mode, Port 3 lines P31 and P36 are used as handshake control lines. The handshake direction is determined by the configuration (input or output) assigned to bit 7 of Port 2 (Figure 6).
PIN FUNCTIONS (Continued)

**Port 3** (P37-P30). Port 3 is an 8-bit, CMOS compatible port with four fixed inputs (P33-P30) and four fixed outputs (P37-P34), and can be configured under software for interrupt and handshake control functions. Port 3, pin 0 is Schmitt-triggered. Pins P31, P32, and P33 are standard CMOS inputs (no Auto Latches) and pins P34, P35, P36, and P37 are push-pull output lines. Low EMI output buffers can be globally programmed by software. Two on-board comparators can process analog signals on P31 and P32 with reference to the voltage on P33. The comparator output can be outputted from P34 and P37, respectively, by setting PCON register (PCON) bit D0 to 1.

The analog function is enabled by setting the D1 of Port 3 Mode Register (P3M). For the interrupt function, P30 and P33 are falling edge triggered interrupt inputs. P31 and P32 can be programmed as falling, rising or both edge triggered interrupt inputs (Figure 7). Access to Counter/Timer 1 is made through P31 (T1N) and P36 (Tout). Handshake lines for Ports 0 and 2 are also available on Port 3 (Table 3). T1N Modes are enabled by setting R243 PRE1 Bit D1 to 0.

**Note:** P33-P30 inputs differ from the Z86C30/31 in that there is no clamping diode to VCC due to the EPROM high-voltage detection circuits. Exceeding the Vih maximum specification during standard operating mode may cause the device to enter EPROM mode.

**Comparator Inputs.** Port 3, pins P31 and P32 each have a comparator front end. The comparator reference voltage (pin P33) is common to both comparators. In analog mode, P31 and P32 are the positive inputs, and P33 is the reference voltage of the comparators.

**Auto Latch.** The Auto Latch puts valid CMOS levels on all CMOS inputs (except P33-P31) that are not externally driven. Whether this is 0 or 1, cannot be determined.

A valid CMOS level, rather than a floating node, reduces excessive supply current flow in the input buffer.

**Low EMI Emission.** The Z86E30/31 can be programmed to operate in a low EMI emission mode in the PCON register. The oscillator and all I/O ports can be programmed as low EMI emission mode independently. Use of this feature results in:

- The pre-drivers slew rate reduced to 10 ns typical.
- Low EMI output drivers have resistance of 200 ohms (typical).
- Low EMI Oscillator.
- Internal SCLK/TCLK = XTAL operation limited to a maximum of 4 MHz - 250 ns cycle time, when Low EMI Oscillator is selected and system clock (SCLK = XTAL, SMR Reg. Bit D1 = 1).

**Table 3. Port 3 Pin Assignments**

<table>
<thead>
<tr>
<th>Pin</th>
<th>I/O</th>
<th>CTC1</th>
<th>AN IN</th>
<th>Int.</th>
<th>P0 HS</th>
<th>P2 HS</th>
</tr>
</thead>
<tbody>
<tr>
<td>P30</td>
<td>IN</td>
<td></td>
<td></td>
<td>IRQ3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P31</td>
<td>IN</td>
<td>T_IN</td>
<td>AN1</td>
<td>IRQ2</td>
<td>D/R</td>
<td></td>
</tr>
<tr>
<td>P32</td>
<td>IN</td>
<td></td>
<td>AN2</td>
<td>IRQ0</td>
<td>D/R</td>
<td></td>
</tr>
<tr>
<td>P33</td>
<td>IN</td>
<td></td>
<td>REF</td>
<td>IRQ1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P34</td>
<td>OUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P35</td>
<td>OUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P36</td>
<td>OUT</td>
<td>T_OUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P37</td>
<td>OUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 7. Port 3 Configuration
FUNCTIONAL DESCRIPTION

The Z86E30/E31 CCP™s incorporate the following special functions to enhance the standard Z8® architecture to provide the user with increased design flexibility.

RESET. The device is reset in one of the following conditions:

- Power-On Reset
- Watch-Dog Timer
- STOP-Mode Recovery Source

Having the Auto Power-on Reset circuitry built in, the Z86E30/E31 does not need to be connected to an external power-on reset circuit. The reset time is 5 ms (typical) plus 18 clock cycles. The Z86E30/31 does not re-initialize WDTMR, SMR, P2M, and P3M registers to their reset values on a STOP-Mode Recovery operation.

Program Memory. The Z86E30/E31 can address up to 4K/2K bytes of internal program memory (Figure 8). The first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts. Address 12 (000CH) to address 4095 (0FFFH)/2047 (07FFH) are reserved for the user program. After reset, the program counter points at the address 000CH which is the starting address of the user program.

EPROM Protect. The 4K/2K bytes program memory is a One-Time-PROM. An EPROM protect feature prevents "dumping" of the ROM contents by inhibiting execution of LDC and LDCI instructions (LDE and LDEI instructions are not available in Z86E30/E31) to program memory in all modes. In EPROM protect mode, the instructions of LDC and LDCI are disabled globally. ROM look-up tables cannot be used with this option.

Expanded Register File (ERF). The register file has been expanded to allow for additional system control registers, mapping of additional peripheral devices, and input/output ports into the register address area. The Z8 register address space R0 through R15 is implemented as 16 groups of 16 registers per group (Figure 11). These register groups are known as the Expanded Register File (ERF).

The low nibble (D3-D0) of the Register Pointer (RP) selects the active ERF group, and the high nibble (D7-D4) of register RP selects the working register group (Figure 9).

Figure 8. Program Memory Map

Figure 9. Register Pointer Register

Three system configuration registers reside in the Expanded Register File at bank FH: PCON, SMR, and WDTMR. The rest of the Expanded Register is not physically implemented and is reserved for future expansion.
Register File. The register file consists of three I/O port registers, 236/124 general-purpose registers, 15 control and status registers, and three system configuration registers in the expanded register group. The instructions can access registers directly or indirectly through an 8-bit address field. This allows a short 4-bit register address using the Register Pointer (Figure 10). In the 4-bit mode, the register file is divided into 16 working register groups, each occupying 16 continuous locations. The Register Pointer addresses the starting location of the active working-register group.

Note: Register Bank E0-EF can only be accessed through working register and indirect addressing modes. (This bank is available in Z86C30 only.)

General Purpose Register (GPR). The general purpose registers are undefined after the device is powered-up. The registers keep their last value after any reset, as long as the reset occurs in the Vcc voltage-specified operating range. Note: Register R254 has been designated as a general purpose register.

Figure 10. Register Pointer
FUNCTIONAL DESCRIPTION (Continued)

### Z8® STANDARD CONTROL REGISTERS

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>RESET CONDITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF</td>
<td>SPL</td>
</tr>
<tr>
<td>FE</td>
<td>GPR</td>
</tr>
<tr>
<td>FD</td>
<td>RP</td>
</tr>
<tr>
<td>FC</td>
<td>FLAGS</td>
</tr>
<tr>
<td>FB</td>
<td>IMR</td>
</tr>
<tr>
<td>FA</td>
<td>IRQ</td>
</tr>
<tr>
<td>F9</td>
<td>IPR</td>
</tr>
<tr>
<td>F6</td>
<td>P01M</td>
</tr>
<tr>
<td>F7</td>
<td>P2M</td>
</tr>
<tr>
<td>F5</td>
<td>PRE0</td>
</tr>
<tr>
<td>F4</td>
<td>T0</td>
</tr>
<tr>
<td>F3</td>
<td>PRE1</td>
</tr>
<tr>
<td>F2</td>
<td>T1</td>
</tr>
<tr>
<td>F1</td>
<td>TMR</td>
</tr>
<tr>
<td>F0</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

**Notes:**
- U = Unknown
- * Will not be reset with a STOP-Mode Recovery.
- ** Will not be reset with a STOP-Mode Recovery, except D0.

### EXPANDED REG. GROUP (F)

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>RESET CONDITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(F)0F</td>
<td>WDTMR</td>
</tr>
<tr>
<td>(F)0E</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F)0D</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F)0C</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F)0B</td>
<td>SMR</td>
</tr>
<tr>
<td>(F)0A</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F)09</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F)08</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F)07</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F)06</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F)05</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F)04</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F)03</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F)02</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F)01</td>
<td>Reserved</td>
</tr>
<tr>
<td>(F)00</td>
<td>PCON</td>
</tr>
</tbody>
</table>

### EXPANDED REG. GROUP (0)

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>RESET CONDITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0)03</td>
<td>P3</td>
</tr>
<tr>
<td>(0)02</td>
<td>P2</td>
</tr>
<tr>
<td>(0)01</td>
<td>Reserved</td>
</tr>
<tr>
<td>(0)00</td>
<td>P0</td>
</tr>
</tbody>
</table>

**Figure 11. Expanded Register File Architecture**
**RAM Protect (Z86E30 Only).** The upper portion of the RAM's address spaces %7F to %EF (excluding the control registers) can be protected from reading and writing. The RAM Protect bit option is EPROM-programmable. After the EPROM option is selected, the user can activate from the internal ROM code to turn off/on the RAM Protect by loading a bit D6 in the IMR register to either a 0 or 1, respectively. A 1 in D6 indicates RAM Protect enabled. This option is only available in the Z86E30.

**Stack.** The Z86E30/E31 has 236/124 general-purpose registers. An 8-bit Stack Pointer (R255) is used for the internal stack that resides within the general-purpose registers.

**Counter/Timers.** There are two 8-bit programmable counter/timers (T0 and T1), each driven by its own 6-bit programmable prescaler. The T1 prescaler can be driven by internal or external clock sources; however, the T0 prescaler is driven by the internal clock only (Figure 12).

The 6-bit prescalers can divide the input frequency of the clock source by any integer number from 1 to 64. Each prescaler drives its counter, which decrements the value (1 to 256) that has been loaded into the counter. When the counter reaches the end of count, a timer interrupt request, IRQ4 (T0) or IRQ5 (T1), is generated.

The counters can be programmed to start, stop, restart to continue, or restart from the initial value. The counters can also be programmed to stop upon reaching zero (single pass mode) or to automatically reload the initial value and continue counting (modulo-n continuous mode).

The counters, but not the prescalers, can be read at any time without disturbing their value or count mode. The clock source for T1 is user-definable and can be either the internal microprocessor clock divided by four, or an external signal input through Port 3. The Timer Mode register configures the external timer input (P30) as an external clock, a trigger input that can be retrigergable or not-retrigergable, or as a gate input for the internal clock. Port 3 line P36 serves as a timer output (TOUT) through which T0, T1 or the internal clock is output. The counter/timers are cascaded by connecting the T0 output to the input of T1.

![Counter/Timer Block Diagram](image)
FUNCTIONAL DESCRIPTION (Continued)

Interrupts. The Z86E30/E31 has six different interrupts from six different sources. The interrupts are maskable and prioritized (Figure 13). The six sources are divided as follows: four sources are claimed by Port 3 lines P33-P30, and two in counter/timers. The Interrupt Mask Register globally or individually enables or disables the six interrupt requests (Table 4).

Figure 13. Interrupt Block Diagram

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Vector Location</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRQ0</td>
<td>/DAV0, IRQ0</td>
<td>0, 1</td>
<td>External (P32), Rising/Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ1</td>
<td>IRQ1</td>
<td>2, 3</td>
<td>External (P33), Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ2</td>
<td>/DAV2, IRQ2, T_in</td>
<td>4, 5</td>
<td>External (P31), Rising/Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ3</td>
<td>IRQ3</td>
<td>6, 7</td>
<td>External (P30), Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ4</td>
<td>T0</td>
<td>8, 9</td>
<td>Internal</td>
</tr>
<tr>
<td>IRQ5</td>
<td>T1</td>
<td>10, 11</td>
<td>Internal</td>
</tr>
</tbody>
</table>
When more than one interrupt is pending, priorities are resolved by a programmable priority encoder that is controlled by the Interrupt Priority register. An interrupt machine cycle is activated when an interrupt request is granted. Thus, disabling all subsequent interrupts, saves the Program Counter and Status Flags, and then branches to the program memory vector location reserved for that interrupt. All Z86E30/E31 interrupts are vectored through locations in the program memory. This memory location and the next byte contain the 16-bit starting address of the interrupt service routine for that particular interrupt request.

To accommodate polled interrupt systems, interrupt inputs are masked and the interrupt request register is polled to determine which of the interrupt requests need service.

An interrupt resulting from AN1 is mapped into IRQ2, and an interrupt from AN2 is mapped into IRQ0. Interrupts IRQ2 and IRQ0 may be rising, falling or both edge triggered, and are programmable by the user. The software may poll to identify the state of the pin.

Programming bits for the Interrupt Edge Select are located in bits D7 and D6 of the IRQ Register (R250). The configuration is shown in Table 5.

Table 5. IRQ Register Configuration

<table>
<thead>
<tr>
<th>IRQ</th>
<th>Interrupt Edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>D7</td>
<td>D6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Notes:
F = Falling Edge
R = Rising Edge

Clock. The Z86E30/E31 on-chip oscillator has a high-gain, parallel-resonant amplifier for connection to a crystal, RC, ceramic resonator, or any suitable external clock source (XTAL1 = Input, XTAL2 = Output). The crystal should be AT cut, 10 kHz to 12 MHz max, with a series resistance (RS) less than or equal to 100 Ohms.

The crystal should be connected across XTAL1 and XTAL2 using the vendor’s recommended capacitors from each pin directly to device pin 22 to reduce injection of system ground noise. The RC oscillator option is selected in the programming mode. The RC oscillator configuration must be an external resistor connected from XTAL1 to XTAL2, with a frequency-setting capacitor from XTAL1 to ground (Figure 14).

Note: RC OSC does not support 12 MHz.

![C1 XTAL1 C2 XTAL2](image1)

C1, C2 = 47 pF TYP
f = 8 MHz

![C1 XTAL1 L XTAL2](image2)

C1, C2 = 22 pF
L = 130 uH
f = 3 MHz

![C1 XTAL1 R XTAL2](image3)

RC
@ 5V VCC (TYP)

C1 = 33 pF
R = 1k
f = 6.6 MHz

* Typical value including pin parasitics

Figure 14. Oscillator Configuration
FUNCTIONAL DESCRIPTION (Continued)

**Power-On Reset (POR).** A timer circuit clocked by a dedicated on-board RC oscillator is used for the Power-On Reset (POR) timer function. The POR timer allows $V_{CC}$ and the oscillator circuit to stabilize before instruction execution begins.

The POR timer circuit is a one-shot timer triggered by one of three conditions:

1. Power bad to Power OK status
2. STOP-Mode Recovery (if D5 of SMR=0)
3. WDT time-out

The POR time is a nominal 5 ms. Bit 5 of the STOP Mode Register (SMR) determines whether the POR timer is bypassed after STOP-Mode Recovery (typical for external clock, and RC/LC oscillators with fast start up time).

**HALT.** This instruction turns off the internal CPU clock, but not the XTAL oscillation. The counter/timers and external interrupts IRQ0, IRQ1, and IRQ2 remain active. The device is recovered by interrupts, either externally or internally generated. An interrupt request must be executed (enabled) to exit HALT mode. After the interrupt service routine, the program continues from the instruction after the HALT.

In order to enter STOP or HALT mode, it is necessary to first flush the instruction pipeline to avoid suspending execution in mid-instruction. To do this, the user must execute a NOP instruction (opcode=FFH) immediately before the appropriate SLEEP instruction, i.e.:

```
FF NOP ; clear the pipeline
6F STOP ; enter STOP mode
```

or

```
FF NOP ; clear the pipeline
7F HALT ; enter HALT mode
```

**STOP.** This instruction turns off the internal clock and external crystal oscillation and reduces the standby current to 10 $\mu A$ or less. STOP mode is terminated by one of the following resets: WDT time-out, POR, or STOP-Mode Recovery Source which is defined by SMR register. This causes the processor to restart the application program at address 000C (HEX).

**Port Configuration Register (PCON).** The PORT Configuration Register (PCON) configures the ports individually: Comparator Output on Port 3, Open-Drain on Port 0, Low EMI Noise on Ports 0, 2, and 3, and Low EMI Noise Oscillator. The PCON Register is located in the Expanded Register File at bank F, location 00 (Figure 15).

![Figure 15. Port Configuration Register (PCON) (Write Only)](image-url)
Comparator Output Port 3 (D0). Bit 0 controls the comparator use in Port 3. A 1 in this location brings the comparator outputs to P34 and P35 and a 0 releases the Port to its standard I/O configuration.

Port 0 Open-Drain (D1). Port 0 is configured as an open-drain by resetting this bit (D1 = 0) and configured as pull-up active by setting D1 = 1. The default value is 1.

Low EMI Port 0 (D3). Port 0 is configured as a Low EMI Port by resetting this bit (D3 = 0) and configured as a Standard Port by setting D3 = 1. The default value is 1.

Low EMI Port 2 (D5). Port 2 is configured as a Low EMI Port by resetting this bit (D5 = 0) and configured as a Standard Port by setting D5 = 1. The default value is 1.

Low EMI Port 3 (D6). Port 3 is configured as a Low EMI Port by resetting this bit (D6 = 0) and configured as a Standard Port by setting D6 = 1. The default value is 1.

Low EMI OSC (D7). This bit of the PCON Register controls the low EMI noise oscillator. A 1 in this location configures the oscillator with standard drive. While a 0 configures the oscillator with low noise drive; however, it does not affect the relationship of SCLK and XTAL.

STOP-Mode Recovery Register (SMR). This register selects the clock divide value and determines the mode of STOP mode recovery (Figure 16). All bits are Write Only except Bit 7 which is a Read Only. Bit 7 is a flag bit that is hardware set on the condition of STOP Recovery and reset by a power-on cycle. Bit 6 controls whether a low or high level is required from the recovery source. Bit 5 controls the reset delay after recovery. Bits 2, 3, and 4 of the SMR register specify the STOP-Mode Recovery Source (Table 7). The SMR is located in Bank F of the Expanded Register Group at address 0BH.

---

**Figure 16. STOP-Mode Recovery Register**

(Write Only Except D7 Which is Read Only)
FUNCTIONAL DESCRIPTION (Continued)

SCLK/TCLK Divide-by-16 Select (D0). This bit of the SMR controls a divide-by-16 prescaler of SCLK/TCLK. The purpose of this control is to selectively reduce device power consumption during normal processor execution (SCLK control) and/or HALT mode (where TCLK sources counter/timers and interrupt logic).

External Clock Divide-By-Two (D1). This bit can eliminate the oscillator divide-by-two circuitry. When this bit is 0, SCLK (System Clock) and TCLK (Timer Clock) are equal to the external clock frequency divided by two. The SCLK/TCLK is equal to the external clock frequency when this bit is set (D1 = 1). Using this bit, together with D7 of PCON, further helps lower EMI [i.e., D7 (PCON) = 0, D1 (SMR) = 1]. The default setting is 0. Maximum frequency is 4 MHz with D1 = 1

STOP-Mode Recovery Source (D2, D3, and D4). These three bits of the SMR register specify the wake-up source of the STOP-Mode Recovery (Figure 17). Table 6 shows the SMR source selected with the setting of D2 to D4. P33-P31 cannot be used to wake up from STOP mode when programmed as analog inputs.

Table 6. STOP-Mode Recovery Source

<table>
<thead>
<tr>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>SMR Source selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>POR recovery only</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>P30 transition</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>P31 transition (Not in analog mode)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>P32 transition (Not in analog mode)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>P33 transition (Not in analog mode)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>P27 transition</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Logical NOR of Port 2 bits 0-3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Logical NOR of Port 2 bits 0-7</td>
</tr>
</tbody>
</table>

STOP-Mode Recovery Delay Select (D5). The 5 ms RESET delay after STOP-Mode Recovery is disabled by programming this bit to a zero. A1 in this bit causes a 5 ms RESET delay after STOP-Mode Recovery. The default condition of this bit is 1. If the fast wake up mode is selected, the STOP-Mode Recovery source must be kept active for at least 5 TpC.

STOP-Mode Recovery Level Select (D6). A 1 in this bit defines that a high level on any one of the recovery sources wakes the Z86E30/E31 from STOP Mode. A 0 defines the low level recovery. The default value is 0.
Cold or Warm Start (D7). This bit is set by the device upon entering STOP Mode. A 0 in this bit indicates that the device has been reset by POR (cold). A 1 in this bit indicates the device was awakened by a SMR source (warm). This bit is read only.

Watch-Dog Timer Mode Register (WDTMR). The WDT is a retriggerable one-shot timer that resets the Z8 if it reaches terminal count (Figure 18). The WDT is disabled after Power-On Reset and initially enabled by executing the WDT instruction. It is refreshed on subsequent executions of the WDT instruction. The WDT cannot be disabled when it has been enabled. The WDT is driven either by an on-board RC oscillator or external oscillator from XTAL1 pin. The POR clock source is selected with bit 4 of the WDT register (Figure 19).

Note: Execution of the WDT instruction affects the Z (Zero), S (Sign), and V (Overflow) flags.

WDTMR Register Accessibility. The WDTMR register is accessible only during the first 64 processor cycles (128 XTAL clock cycles) from the execution of the first instruction after Power-On Reset, Watch-Dog Reset, or a STOP Mode Recovery. After this point, the register cannot be modified by any means, intentional or otherwise. The WDTMR cannot be read and is located in Bank F of the Expanded Register Group at address location 0FH.

WDT Time-out Period (D0 and D1). Bits 0 and 1 control a tap circuit that determines the time-out periods that can be obtained. Table 7 shows the time-out period. The default value of D0 and D1 are 1 and 0, respectively.

Table 7. Time-out Period of the WDT

<table>
<thead>
<tr>
<th>D1</th>
<th>D0</th>
<th>Time-Out of Internal RC OSC</th>
<th>Time-Out of the Crystal Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>5 ms</td>
<td>256 TpC</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>15 ms</td>
<td>512 TpC</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>25 ms</td>
<td>1024 TpC</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>100 ms</td>
<td>4096 TpC</td>
</tr>
</tbody>
</table>

WDT During the HALT Mode (D2). This bit determines whether or not the WDT is active during HALT mode. A 1 indicates that the WDT is active during HALT. A 0 disables the WDT in HALT mode. The default value is 1.

WDT During STOP Mode (D3). This bit determines whether or not the WDT is active during STOP mode. A 1 indicates active during STOP mode. A 0 disables the WDT during STOP mode. Since the on-board OSC is stopped during STOP mode, the WDT clock source has to select the on-board RC OSC for the WDT to recover from STOP mode. The default is 1.

Clock Source For WDT (D4). This bit determines which oscillator source is used to clock the internal POR and WDT counter chain. If the bit is a 1, the internal RC oscillator is bypassed and the POR and WDT clock source is driven from the external pin, XTAL1. The default configuration of the bit is 0, which selects the RC oscillator.

Note: Execution of the WDT instruction affects the Z (Zero), S (Sign), and V (Overflow) flags.

WDTMR Register Accessibility. The WDTMR register is accessible only during the first 64 processor cycles (128 XTAL clock cycles) from the execution of the first instruction after Power-On Reset, Watch-Dog Reset, or a STOP Mode Recovery. After this point, the register cannot be modified by any means, intentional or otherwise. The WDTMR cannot be read and is located in Bank F of the Expanded Register Group at address location 0FH.

WDT Time-out Period (D0 and D1). Bits 0 and 1 control a tap circuit that determines the time-out periods that can be obtained. Table 7 shows the time-out period. The default value of D0 and D1 are 1 and 0, respectively.

Table 7. Time-out Period of the WDT

<table>
<thead>
<tr>
<th>D1</th>
<th>D0</th>
<th>Time-Out of Internal RC OSC</th>
<th>Time-Out of the Crystal Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>5 ms</td>
<td>256 TpC</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>15 ms</td>
<td>512 TpC</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>25 ms</td>
<td>1024 TpC</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>100 ms</td>
<td>4096 TpC</td>
</tr>
</tbody>
</table>

WDT During the HALT Mode (D2). This bit determines whether or not the WDT is active during HALT mode. A 1 indicates that the WDT is active during HALT. A 0 disables the WDT in HALT mode. The default value is 1.

WDT During STOP Mode (D3). This bit determines whether or not the WDT is active during STOP mode. A 1 indicates active during STOP mode. A 0 disables the WDT during STOP mode. Since the on-board OSC is stopped during STOP mode, the WDT clock source has to select the on-board RC OSC for the WDT to recover from STOP mode. The default is 1.

Clock Source For WDT (D4). This bit determines which oscillator source is used to clock the internal POR and WDT counter chain. If the bit is a 1, the internal RC oscillator is bypassed and the POR and WDT clock source is driven from the external pin, XTAL1. The default configuration of the bit is 0, which selects the RC oscillator.

Figure 18. Watch-Dog Timer Mode Register (Write Only)
FUNCTIONAL DESCRIPTION (Continued)

Figure 19. Resets and WDT
**Auto Reset Voltage.** An on-board Voltage Comparator checks that $V_{CC}$ is at the required level to ensure correct operation of the device. Reset is globally driven if $V_{CC}$ is below $V_{RST}$ (Figure 20).

If the $V_{CC}$ drops below 4.5V while the device is in operation, the device must be powered down and then re-powered up again. **Note:** $V_{CC}$ must be in the allowed operating range (4.5V to 5.5V) prior to the minimum Power-On Reset time-out ($T_{PoR}$).

---

**Figure 20. Typical Z86E30/E31 $V_{RST}$ Voltage vs Temperature**
**FUNCTIONAL DESCRIPTION (Continued)**

### EPROM Programming Mode

Table 8 shows the programming voltages of each programming mode. Table 9, Figures 21, 22, and 23 show the programming timing of each programming mode. Figure 24 shows the flow-chart of an Intelligent Programming Algorithm, which is compatible with a 2764A EPROM (Z86E30/E31 is 4K/2K EPROM, 2764A is 8K EPROM).

Table 25 shows the circuit diagram of the Z86E30/E31 programming adaptor which adapts from 2764A to Z86E30/E31. Since the EPROM size of Z86E30/E31 differs from 2764A, the programming address range should be set from 0000H to 0FFFH.

#### Table 8. EPROM Programming Table

<table>
<thead>
<tr>
<th>Programming Modes</th>
<th>V&lt;sub&gt;PP&lt;/sub&gt;</th>
<th>EPM</th>
<th>/CE</th>
<th>/OE</th>
<th>/PGM</th>
<th>ADDR</th>
<th>DATA</th>
<th>V&lt;sub&gt;CC&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPROM READ1</td>
<td>X</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>ADDR</td>
<td>Out</td>
<td>4.5V†</td>
</tr>
<tr>
<td>EPROM READ2</td>
<td>X</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>ADDR</td>
<td>Out</td>
<td>5.5V†</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>X</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>ADDR</td>
<td>In</td>
<td>6.0V</td>
</tr>
<tr>
<td>PROGRAM VERIFY</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>X</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>ADDR</td>
<td>Out</td>
<td>6.0V</td>
</tr>
<tr>
<td>EPROM PROTECT</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>NU</td>
<td>NU</td>
<td>6.0V</td>
</tr>
<tr>
<td>RC OSCILLATOR SELECT</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>NU</td>
<td>NU</td>
<td>6.0V</td>
</tr>
<tr>
<td>RAM PROTECT</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>V&lt;sub&gt;H&lt;/sub&gt;</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>NU</td>
<td>NU</td>
<td>6.0V</td>
</tr>
</tbody>
</table>

**Notes:**

- V<sub>H</sub> = 1.25V
- V<sub>IL</sub> = As per specific Z8 DC specification.
- X = Not used, but must be set to V<sub>H</sub>, V<sub>IL</sub>, or V<sub>L</sub> level.
- NU = Not used, but must be set to either V<sub>H</sub> or V<sub>L</sub> level.
- I<sub>PP</sub> during programming = 40 mA maximum.
- I<sub>CC</sub> during programming, verify, or read = 40 mA maximum.

† Although most programmers do an EPROM read at V<sub>CC</sub> = 5.0V, Zilog recommends an EPROM read at V<sub>CC</sub> = 4.5V and 5.0V to ensure proper device operations during the V<sub>CC</sub> after programming.

#### Table 9. EPROM Programming Timing

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Name</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Address Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>2</td>
<td>Data Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>3</td>
<td>V&lt;sub&gt;PP&lt;/sub&gt; Setup</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>4</td>
<td>V&lt;sub&gt;CC&lt;/sub&gt; Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>5</td>
<td>Chip Enable Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>6</td>
<td>Program Pulse Width</td>
<td>0.95</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>7</td>
<td>Data Hold Time</td>
<td>2</td>
<td></td>
<td>ms</td>
</tr>
<tr>
<td>8</td>
<td>/OE Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>9</td>
<td>Data Access Time</td>
<td></td>
<td>200</td>
<td>ns</td>
</tr>
<tr>
<td>10</td>
<td>Data Output Float Time</td>
<td></td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td>11</td>
<td>Overprogram Pulse Width</td>
<td></td>
<td>2.85</td>
<td>ms</td>
</tr>
<tr>
<td>12</td>
<td>EPM Setup Time</td>
<td></td>
<td>2</td>
<td>µs</td>
</tr>
<tr>
<td>13</td>
<td>/PGM Setup Time</td>
<td></td>
<td>2</td>
<td>µs</td>
</tr>
<tr>
<td>14</td>
<td>Address to /OE Setup Time</td>
<td></td>
<td>2</td>
<td>µs</td>
</tr>
<tr>
<td>15</td>
<td>Option Program Pulse Width</td>
<td></td>
<td>78</td>
<td>ms</td>
</tr>
</tbody>
</table>
Figure 21. EPROM READ Mode Timing Diagram
Figure 22. Timing Diagram of EPROM Program and Verify Modes
Figure 23. Timing Diagram of EPROM Protect, RAM Protect, and RC OSC Modes
FUNCTIONAL DESCRIPTION (Continued)

Note: To ensure proper operations during the spec., Zilog recommends verification over the $V_{CC}$ range of the device $V_{CC}$ spec.

Figure 24. Z86E30/E31 Programming Algorithm
Figure 25. Z86E30/E31 Programming Adaptor Circuitry

Note: The programming address must be set to 0000H - 0FFFH (Lower 4K Byte Memory).
**ABSOLUTE MAXIMUM RATINGS**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{CC}$</td>
<td>Supply Voltage (*)</td>
<td>-0.3</td>
<td>7.0</td>
<td>V</td>
</tr>
<tr>
<td>$V_{IHM}$</td>
<td>Max Input Voltage</td>
<td>7</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$T_{STG}$</td>
<td>Storage Temp</td>
<td>-65</td>
<td>150</td>
<td>C</td>
</tr>
<tr>
<td>$T_A$</td>
<td>Oper Ambient Temp</td>
<td>†</td>
<td></td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Power Dissipation</td>
<td>2.2</td>
<td></td>
<td>W</td>
</tr>
</tbody>
</table>

Notes:
* Voltage on all pins with respect to Ground.
** Applies to all Port pins, except Port 31, 32, 33 and must limit current going into and out of Port pin to 250 µA maximum.
† See Ordering Information.

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**STANDARD TEST CONDITIONS**

The characteristics listed below apply for standard test conditions as noted. All voltages are referenced to Ground. Positive current flows into the referenced pin (Figure 26).

**CAPACITANCE**

$T_A = 25°C; V_{CC} = \text{GND} = 0V; f = 1.0 \text{ MHz};$ unmeasured pins returned to GND.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input capacitance</td>
<td>0</td>
<td>12 pF</td>
</tr>
<tr>
<td>Output capacitance</td>
<td>0</td>
<td>12 pF</td>
</tr>
<tr>
<td>I/O capacitance</td>
<td>0</td>
<td>12 pF</td>
</tr>
</tbody>
</table>

**$V_{CC}$ SPECIFICATION**

$V_{CC} = 5.0V \pm 0.5V$
## DC ELECTRICAL CHARACTERISTICS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>( V_{CC} ) Note[3]</th>
<th>( T_A = 0°C ) to +70°C</th>
<th>Typical</th>
<th>Units</th>
<th>Conditions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max Input Voltage</td>
<td>5.0V</td>
<td>( V_{CC} + 0.5V )</td>
<td>( V_{CC} + 0.3V )</td>
<td>2.5</td>
<td>V</td>
<td>( I_{IN} &lt; 250 \mu A )</td>
</tr>
<tr>
<td>( V_{CH} )</td>
<td>Clock Input High Voltage</td>
<td>5.0V</td>
<td>0.7 ( V_{CC} )</td>
<td>0.2 ( V_{CC} )</td>
<td>1.5</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
</tr>
<tr>
<td>( V_{CL} )</td>
<td>Clock Input Low Voltage</td>
<td>5.0V</td>
<td>( V_{SS} - 0.3 )</td>
<td>0.2 ( V_{CC} )</td>
<td>1.5</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
</tr>
<tr>
<td>( V_{IH} )</td>
<td>Input High Voltage</td>
<td>5.0V</td>
<td>0.7 ( V_{CC} )</td>
<td>( V_{CC} + 0.3 )</td>
<td>2.5</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>( V_{IL} )</td>
<td>Input Low Voltage</td>
<td>5.0V</td>
<td>( V_{SS} - 0.3 )</td>
<td>0.2 ( V_{CC} )</td>
<td>1.5</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>( V_{OH} )</td>
<td>Output High Voltage (Low EMI Mode)</td>
<td>5.0V</td>
<td>( V_{CC} - 0.4 )</td>
<td>( V_{CC} - 0.4 )</td>
<td>4.8</td>
<td>V</td>
<td>( I_{OH} = -2.0 , mA )</td>
</tr>
<tr>
<td>( V_{OL} )</td>
<td>Output Low Voltage (Low EMI Mode)</td>
<td>5.0V</td>
<td>0.4</td>
<td>0.1</td>
<td>V</td>
<td>( I_{OL} = +4.0 , mA )</td>
<td>[9]</td>
</tr>
<tr>
<td>( V_{OL2} )</td>
<td>Output Low Voltage</td>
<td>5.0V</td>
<td>1.5</td>
<td>0.3</td>
<td>V</td>
<td>( I_{OL} = +12 , mA )</td>
<td>3 Pin Max</td>
</tr>
<tr>
<td>( V_{OFFSET} )</td>
<td>Comparator Input Offset Voltage</td>
<td>5.0V</td>
<td>50</td>
<td>10</td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( I_{IL} )</td>
<td>Input Leakage</td>
<td>5.0V</td>
<td>-10</td>
<td>+10</td>
<td>&lt; 1</td>
<td>( \mu A )</td>
<td>( V_{IN} = 0V, V_{CC} )</td>
</tr>
<tr>
<td>( I_{OL} )</td>
<td>Output Leakage</td>
<td>5.0V</td>
<td>-10</td>
<td>+10</td>
<td>&lt; 1</td>
<td>( \mu A )</td>
<td>( V_{IN} = 0V, V_{CC} )</td>
</tr>
<tr>
<td>( I_{CC} )</td>
<td>Supply Current (Standard Mode)</td>
<td>5.0V</td>
<td>16</td>
<td>15</td>
<td>mA</td>
<td>( \geq 8 , MHz )</td>
<td>[4,5,11]</td>
</tr>
</tbody>
</table>

### Notes:

[1] \( I_{CC} \)  
- Clock-driven XTAL: 0.3 mA to 6.0 mA at 8 MHz  
- Crystal or resonator: 3.5 mA to 6.0 mA at 8 MHz

[2] \( V_{SS} = 0V = GND \).

[3] \( V_{CC} \) must be in the allowed operating range (4.5V to 5.5V) prior to the minimum TPOA timeout. \( V_{CC} \) specified at 4.5V to 5.5V.


[5] \( CL1 = CL2 = 100 \, pF \).

[6] Same as note [4] except inputs at \( V_{CC} \).

[7] Except clock pins and Port 3 input pins in EPROM mode.

[8] Port Low EMI mode.

[9] Port STD mode.

[10] SMR Reg Bit D1=1.

### DC ELECTRICAL CHARACTERISTICS (Continued)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{CC}$</th>
<th>$T_A = 0°C$ to $+70°C$</th>
<th>Typical @ $25°C$</th>
<th>Units</th>
<th>Conditions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{CC1}$</td>
<td>Standby Current (Standard Mode)</td>
<td>5.0V</td>
<td>6.0</td>
<td>3.5</td>
<td>mA</td>
<td>HALT mode $V_W = 0V$, $V_{CC}$ @ 8 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>3.0</td>
<td>1.50</td>
<td>mA</td>
<td>Clock Divide-by-16 @ 8 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td>$I_{CC}$</td>
<td>Supply Current (SCLK/TCLK = XTAL)</td>
<td>5.0V</td>
<td>7.5</td>
<td>5.0</td>
<td>mA</td>
<td>@ 2 MHz</td>
<td>[4,5,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>12.0</td>
<td>8.0</td>
<td>mA</td>
<td>@ 4 MHz</td>
<td>[4,5,10]</td>
</tr>
<tr>
<td>$I_{CC1}$</td>
<td>Standby Current (SCLK/TCLK = XTAL)</td>
<td>5.0V</td>
<td>2.0</td>
<td>1.0</td>
<td>mA</td>
<td>Clock Divide-by-16 @ 2 MHz</td>
<td>[4,5,10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>3.0</td>
<td>1.5</td>
<td>mA</td>
<td>Clock Divide-by-16 @ 4 MHz</td>
<td>[4,5,10]</td>
</tr>
<tr>
<td>$I_{CC1}$ (Standard Mode)</td>
<td></td>
<td>5.0V</td>
<td>2.0</td>
<td>0.75</td>
<td>mA</td>
<td></td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>2.0</td>
<td>1.0</td>
<td>mA</td>
<td></td>
<td>[4,5]</td>
</tr>
<tr>
<td>$I_{CC2}$</td>
<td>Standby Current</td>
<td>5.0V</td>
<td>10</td>
<td>2</td>
<td>$\mu$A</td>
<td>STOP mode $V_W = 0V$, $V_{CC}$ WDT is not Running</td>
<td>[6]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>800</td>
<td>450</td>
<td>$\mu$A</td>
<td>STOP mode $V_W = 0V$, $V_{CC}$ WDT is Running</td>
<td>[6]</td>
</tr>
<tr>
<td>$I_{ALL}$</td>
<td>Auto Latch Low Current</td>
<td>5.0V</td>
<td>-10</td>
<td>-5</td>
<td>$\mu$A</td>
<td>$OV &lt; V_W &lt; V_{CC}$</td>
<td></td>
</tr>
<tr>
<td>$I_{ALH}$</td>
<td>Auto Latch High Current</td>
<td>5.0V</td>
<td>20</td>
<td>10</td>
<td>$\mu$A</td>
<td>$OV &lt; V_W &lt; V_{CC}$</td>
<td></td>
</tr>
<tr>
<td>$T_{POR}$</td>
<td>Power-On Reset</td>
<td>5.0V</td>
<td>2.5</td>
<td>4.5</td>
<td>ms</td>
<td></td>
<td>[3]</td>
</tr>
<tr>
<td>$V_{ref}$</td>
<td>Auto Reset Voltage</td>
<td>3.0</td>
<td>2.6</td>
<td></td>
<td>V</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**

1. $I_{CC1}$
2. $V_{SS}=OV=GND$.
3. $V_{CC}$ must be in the allowed operating range (4.5V to 5.5V) prior to the minimum $T_{POR}$ timeout. $V_{CC}$ specified at 4.5V to 5.5V.
4. All outputs unloaded, I/O pins floating, inputs at rail.
5. CL1=CL2=100 pF.
6. Same as note 4 except inputs at $V_{CC}$.
7. Except clock pins and Port 3 input pins in EPROM mode.
8. Port Low EMI mode.
9. Port STD mode.
10. SMR Reg Bit D1=1.
11. Z86E30 only.
AC ELECTRICAL CHARACTERISTICS
Additional Timing Diagram (Standard Mode for SCLK/TCLK + XTAL/2)

Figure 27. Additional Timing
## AC ELECTRICAL CHARACTERISTICS

### Additional Timing Table (Standard Mode)

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{CC}$</th>
<th>$T_A = 0^\circ$C to +70°C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TpC</td>
<td>Input Clock Period</td>
<td>5.0V</td>
<td>125 DC</td>
</tr>
<tr>
<td>2</td>
<td>TrC,TfC</td>
<td>Clock Input Rise and Fall Times</td>
<td>5.0V</td>
<td>25 15 ns</td>
</tr>
<tr>
<td>3</td>
<td>TwC</td>
<td>Input Clock Width</td>
<td>5.0V</td>
<td>62.5 41.6</td>
</tr>
<tr>
<td>4</td>
<td>TwTinL</td>
<td>Timer Input Low Width</td>
<td>5.0V</td>
<td>70 70 ns</td>
</tr>
<tr>
<td>5</td>
<td>TwTinH</td>
<td>Timer Input High Width</td>
<td>5.0V</td>
<td>5TpC 5TpC</td>
</tr>
<tr>
<td>6</td>
<td>TpTin</td>
<td>Timer Input Period</td>
<td>5.0V</td>
<td>8TpC 8TpC</td>
</tr>
<tr>
<td>7</td>
<td>TpTin</td>
<td>Timer Input Rise and Fall Times</td>
<td>5.0V</td>
<td>100 100 ns</td>
</tr>
<tr>
<td>8A</td>
<td>TwiL</td>
<td>Int. Request Low Time</td>
<td>5.0V</td>
<td>70 70 ns</td>
</tr>
<tr>
<td>8B</td>
<td>TwiL</td>
<td>Int. Request Low Time</td>
<td>5.0V</td>
<td>5TpC 5TpC</td>
</tr>
<tr>
<td>9</td>
<td>TwiH</td>
<td>Int. Request Input High Time</td>
<td>5.0V</td>
<td>5TpC 5TpC</td>
</tr>
<tr>
<td>10</td>
<td>Twsm</td>
<td>STOP-Mode Recovery Width Spec</td>
<td>5.0V</td>
<td>12 12 ms</td>
</tr>
<tr>
<td>11</td>
<td>Tosi</td>
<td>Oscillator Start-up Time</td>
<td>5.0V</td>
<td>5TpC 5TpC</td>
</tr>
<tr>
<td>12</td>
<td>Twdt</td>
<td>Watch-Dog Timer Delay Time</td>
<td>5.0V</td>
<td>5 5 ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.0V</td>
<td>15 15 ms</td>
</tr>
</tbody>
</table>

### Notes:

[1] Timing Reference uses 0.7 $V_{CC}$ for a logic 1 and 0.2 $V_{CC}$ for a logic 0.
[2] Interrupt request through Port 3 (P33-P31).
[3] Interrupt request through Port 3 (P30).
[5] Reg. WDTMR.
[6] 5.0V ±0.5V
[7] Reg. WDTMR D1 = 0, D0 = 0.
[8] Reg. WDTMR D1 = 0, D0 = 1.
[9] Reg. WDTMR D1 = 1, D0 = 0.
[10] Reg. WDTMR D1 = 1, D0 = 1.
[11] Z86E30 max frequency = 12 MHz; Z86E31 max frequency = 8 MHz.
AC ELECTRICAL CHARACTERISTICS
Handshake Timing Diagrams

Data In

/DAV (Input)

RDY (Output)

Data In Valid

Next Data In Valid

Delayed DAV

Delayed RDY

Figure 28. Input Handshake Timing

Data Out

/DAV (Output)

RDY (Input)

Data Out Valid

Next Data Out Valid

Delayed DAV

Delayed RDY

Figure 29. Output Handshake Timing
AC ELECTRICAL CHARACTERISTICS
Handshake Timing Table - Standard Mode

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$ Note[1]</th>
<th>$T_A = 0^\circ$C to $+70^\circ$C 8 MHz, 12 MHz [2]</th>
<th>Data Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TSDI(DAV)</td>
<td>Data In Setup Time</td>
<td>5.0V</td>
<td>Min 0 Max 0</td>
<td>IN</td>
</tr>
<tr>
<td>2</td>
<td>TDI(DAV)</td>
<td>Data In Hold Time</td>
<td>5.0V</td>
<td>Min 115 Max 115</td>
<td>IN</td>
</tr>
<tr>
<td>3</td>
<td>TwDAV</td>
<td>Data Available Width</td>
<td>5.0V</td>
<td>Min 110 Max 110</td>
<td>IN</td>
</tr>
<tr>
<td>4</td>
<td>TdDAVI(RDY)</td>
<td>DAV Fall to RDY Fall Delay</td>
<td>5.0V</td>
<td>Min 115 Max 115</td>
<td>IN</td>
</tr>
<tr>
<td>5</td>
<td>TdDAVID(RDY)</td>
<td>DAV Rise to RDY Rise Delay</td>
<td>5.0V</td>
<td>Min 80 Max 80</td>
<td>IN</td>
</tr>
<tr>
<td>6</td>
<td>TdRDYO(DAV)</td>
<td>RDY Rise to DAV Fall Delay</td>
<td>5.0V</td>
<td>Min 0 Max 0</td>
<td>IN</td>
</tr>
<tr>
<td>7</td>
<td>TdDO(DAV)</td>
<td>Data Out to DAV Fall Delay</td>
<td>5.0V</td>
<td>Min 63 Max 63</td>
<td>OUT</td>
</tr>
<tr>
<td>8</td>
<td>TdDAVO(RDY)</td>
<td>DAV Fall to RDY Fall Delay</td>
<td>5.0V</td>
<td>Min 0 Max 0</td>
<td>OUT</td>
</tr>
<tr>
<td>9</td>
<td>TdRDYO(DAV)</td>
<td>RDY Fall to DAV Rise Delay</td>
<td>5.0V</td>
<td>Min 115 Max 115</td>
<td>OUT</td>
</tr>
<tr>
<td>10</td>
<td>TwRDY</td>
<td>RDY Width</td>
<td>5.0V</td>
<td>Min 80 Max 80</td>
<td>OUT</td>
</tr>
<tr>
<td>11</td>
<td>TdRDYOd(DAV)</td>
<td>RDY Rise to DAV Fall Delay</td>
<td>5.0V</td>
<td>Min 80 Max 80</td>
<td>OUT</td>
</tr>
</tbody>
</table>

Note:
[1] 5.0V ±0.5V Standard operating temperature range 0°C to +70°C.
[2] Z86E30 max frequency = 12 MHz; Z86E31 max frequency = 8 MHz.
EXPANDED REGISTER FILE CONTROL REGISTERS

Figure 30. Port Configuration Register (Write Only)

* Default setting after RESET

Figure 31. STOP-Mode Recovery Register (Write Only Except Bit D7, Which is Read Only)

* Default setting after RESET.
** Default setting after RESET and STOP-Mode Recovery.

Figure 32. Watch-Dog Timer Mode Register (Write Only)

* Default setting after RESET
Z8® CONTROL REGISTER DIAGRAMS

**Figure 33. Reserved**

R240

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>

R241 TMR

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No Function</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Load T0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Disable T0 Count</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Enable T0 Count</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No Function</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Load T1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Disable T1 Count</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Enable T1 Count</td>
</tr>
</tbody>
</table>

**Figure 34. Timer Mode Register (F1H: Read/Write)**

R242 T1

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T1 Initial Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(When Written)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Range: 1-256 Decimal 01-00 HEX)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T1 Current Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(When Read)</td>
</tr>
</tbody>
</table>

**Figure 35. Counter Timer 1 Register (F2H: Read/Write)**

R243 PRE1

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Count Mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 T0 Single Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 T1 Modulo N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Clock Source</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 T1 Internal</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 T1 External Timing Input (TIN) Mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Prescaler Modulo</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Range: 1-64 Decimal 01-00 HEX)</td>
</tr>
</tbody>
</table>

**Figure 36. Prescaler 1 Register (F3H: Write Only)**

R244 T0

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T0 Initial Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(When Written)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Range: 1-256 Decimal 01-00 HEX)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T0 Current Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(When Read)</td>
</tr>
</tbody>
</table>

**Figure 37. Counter/Timer 0 Register (F4H: Read/Write)**

R245 PRED

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Count Mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 T0 Single Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 T0 Modulo N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reserved (Must Be 0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Prescaler Modulo</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Range: 1-64 Decimal 01-00 HEX)</td>
</tr>
</tbody>
</table>

**Figure 38. Prescaler 0 Register (F5H: Write Only)**

R246 P2M

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>P27 - P20 I/O Definition</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 Defines Bit as Output</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 Defines Bit as Input</td>
</tr>
</tbody>
</table>

**Figure 39. Port 2 Mode Register (F6H: Write Only)**
<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 40. Port 3 Mode Register**
(F7H: Write Only)

- **0** Port 2 Open-Drain
- **1** Port 2 Push-pull
- **0** P31, P32 Digital Mode
- **1** P31, P32 Analog Mode
- **0** P32 = Input
- **1** P32 = Output
- P32 = /Dilo/RDY0
- P35 = RDY0/DAY0
- Reserved (Must be 0)
- **0** P31 = Input
- **1** P31 = Output
- P31 = /Dilo/2/RDY2
- P36 = RDY2/DAY2
- Reserved (Must be 0)

**Figure 42. Interrupt Priority Register**
(F9H: Write Only)

- Interrupt Group Priority
- Reserved = 000
- C > A > B = 001
- A > C > B = 010
- A > B > C = 011
- B > C > A = 100
- C > B > A = 101
- Reserved = 111
- IRQ1, IRQ4 Priority (Group C)
- **0** IRQ1 > IRQ4
- **1** IRQ4 > IRQ1
- IRQ0, IRQ2 Priority (Group B)
- **0** IRQ2 > IRQ0
- **1** IRQ0 > IRQ2
- IRQ3, IRQ5 Priority (Group A)
- **0** IRQ5 > IRQ3
- **1** IRQ3 > IRQ5
- Reserved (Must be 0)

**Figure 43. Interrupt Request Register**
(FAH: Read/Write)

- IRQ0 = P32 Input
- IRQ1 = P33 Input
- IRQ2 = P31 Input
- IRQ3 = P30 Input
- IRQ4 = T0
- IRQ5 = T1
- Inter Edge
- P31 ↓ P32 ↓ = 00
- P31 ↓ P32 ↑ = 01
- P31 ↑ P32 ↓ = 10
- P31 ↑ P32 ↑ = 11

**Figure 44. Interrupt Mask Register**
(FBH: Read/Write)

- **1** Enables IRQ5-IRQ0
- **0** (D0 = IRQ0)
- **1** Enables RAM Protect
- **1** Enables Interrupts

† RAM Protect option must be previously selected.
Z8® CONTROL REGISTER DIAGRAMS (Continued)

**Figure 45. Flag Register**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Flag F1</td>
<td>User Flag F2</td>
<td>Half Carry Flag</td>
<td>Decimal Adjust Flag</td>
<td>Overflow Flag</td>
<td>Sign Flag</td>
<td>Zero Flag</td>
<td>Carry Flag</td>
</tr>
</tbody>
</table>

**Figure 46. Register Pointer**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expanded Register File</td>
<td>Register Pointer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 47. General-Purpose Register**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 = 0 State</td>
<td>1 = 1 State</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 48. Stack Pointer**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stack Pointer Lower Byte (SP7 - SP0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
INSTRUCTION SET NOTATION

Addressing Modes. The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRR</td>
<td>Indirect register pair or indirect working-</td>
</tr>
<tr>
<td></td>
<td>register pair address</td>
</tr>
<tr>
<td>Irr</td>
<td>Indirect working-register pair only</td>
</tr>
<tr>
<td>X</td>
<td>Indexed address</td>
</tr>
<tr>
<td>DA</td>
<td>Direct address</td>
</tr>
<tr>
<td>RA</td>
<td>Relative address</td>
</tr>
<tr>
<td>IM</td>
<td>Immediate</td>
</tr>
<tr>
<td>R</td>
<td>Register or working-register address</td>
</tr>
<tr>
<td>r</td>
<td>Working-register address only</td>
</tr>
<tr>
<td>IR</td>
<td>Indirect-register or indirect working-register</td>
</tr>
<tr>
<td></td>
<td>address</td>
</tr>
<tr>
<td>Ir</td>
<td>Indirect working-register address only</td>
</tr>
<tr>
<td>RR</td>
<td>Register pair or working register pair address</td>
</tr>
</tbody>
</table>

Symbols. The following symbols are used in describing the instruction set.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst</td>
<td>Destination location or contents</td>
</tr>
<tr>
<td>src</td>
<td>Source location or contents</td>
</tr>
<tr>
<td>cc</td>
<td>Condition code</td>
</tr>
<tr>
<td>@</td>
<td>Indirect address prefix</td>
</tr>
<tr>
<td>SP</td>
<td>Stack Pointer</td>
</tr>
<tr>
<td>PC</td>
<td>Program Counter</td>
</tr>
<tr>
<td>FLAGS</td>
<td>Flag register (Control Register 252)</td>
</tr>
<tr>
<td>RP</td>
<td>Register Pointer (R253)</td>
</tr>
<tr>
<td>IMR</td>
<td>Interrupt mask register (R251)</td>
</tr>
</tbody>
</table>

Flags. Control register (R252) contains the following six flags:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Carry flag</td>
</tr>
<tr>
<td>Z</td>
<td>Zero flag</td>
</tr>
<tr>
<td>S</td>
<td>Sign flag</td>
</tr>
<tr>
<td>V</td>
<td>Overflow flag</td>
</tr>
<tr>
<td>D</td>
<td>Decimal-adjust flag</td>
</tr>
<tr>
<td>H</td>
<td>Half-carry flag</td>
</tr>
</tbody>
</table>

Affected flags are indicated by:

- **0**: Clear to zero  
- **1**: Set to one  
- *****: Set to clear according to operation  
- **-**: Unaffected  
- **x**: Undefined
## Condition Codes

<table>
<thead>
<tr>
<th>Value</th>
<th>Mnemonic</th>
<th>Meaning</th>
<th>Flags Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td></td>
<td>Always True</td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>C</td>
<td>Carry</td>
<td>C = 1</td>
</tr>
<tr>
<td>1111</td>
<td>NC</td>
<td>No Carry</td>
<td>C = 0</td>
</tr>
<tr>
<td>0110</td>
<td>Z</td>
<td>Zero</td>
<td>Z = 1</td>
</tr>
<tr>
<td>1110</td>
<td>NZ</td>
<td>Not Zero</td>
<td>Z = 0</td>
</tr>
<tr>
<td>1101</td>
<td>PL</td>
<td>Plus</td>
<td>S = 0</td>
</tr>
<tr>
<td>0101</td>
<td>MI</td>
<td>Minus</td>
<td>S = 1</td>
</tr>
<tr>
<td>0100</td>
<td>OV</td>
<td>Overflow</td>
<td>V = 1</td>
</tr>
<tr>
<td>1100</td>
<td>NOV</td>
<td>No Overflow</td>
<td>V = 0</td>
</tr>
<tr>
<td>0110</td>
<td>EQ</td>
<td>Equal</td>
<td>Z = 1</td>
</tr>
<tr>
<td>1110</td>
<td>NE</td>
<td>Not Equal</td>
<td>Z = 0</td>
</tr>
<tr>
<td>1001</td>
<td>GE</td>
<td>Greater Than or Equal</td>
<td>(S XOR V) = 0</td>
</tr>
<tr>
<td>0001</td>
<td>LT</td>
<td>Less than</td>
<td>(S XOR V) = 1</td>
</tr>
<tr>
<td>1010</td>
<td>GT</td>
<td>Greater Than</td>
<td>[Z OR (S XOR V)] = 0</td>
</tr>
<tr>
<td>0010</td>
<td>LE</td>
<td>Less Than or Equal</td>
<td>[Z OR (S XOR V)] = 1</td>
</tr>
<tr>
<td>1111</td>
<td>UGE</td>
<td>Unsigned Greater Than or Equal</td>
<td>C = 0</td>
</tr>
<tr>
<td>0111</td>
<td>ULT</td>
<td>Unsigned Less Than</td>
<td>C = 1</td>
</tr>
<tr>
<td>1011</td>
<td>UGT</td>
<td>Unsigned Greater Than</td>
<td>(C = 0 AND Z = 0) = 1</td>
</tr>
<tr>
<td>0011</td>
<td>ULE</td>
<td>Unsigned Less Than or Equal</td>
<td>(C OR Z) = 1</td>
</tr>
<tr>
<td>0000</td>
<td>F</td>
<td>Never True (Always False)</td>
<td></td>
</tr>
</tbody>
</table>
INSTRUCTION FORMATS

One-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst/src</th>
<th>src/dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>LDEI</td>
<td>dst</td>
<td>src</td>
</tr>
<tr>
<td>OP</td>
<td>LOE</td>
<td>dst</td>
<td>src</td>
</tr>
<tr>
<td>OP</td>
<td>LDCI</td>
<td>dst</td>
<td>src</td>
</tr>
<tr>
<td>OP</td>
<td>LDC</td>
<td>dst</td>
<td>src</td>
</tr>
</tbody>
</table>

Two-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst/src</th>
<th>src/dst</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP</td>
<td>LDAU</td>
<td>dst</td>
<td>src</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>DAL</td>
<td>dst</td>
<td>src</td>
<td></td>
</tr>
</tbody>
</table>

Three-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dst/src</th>
<th>src/dst</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP</td>
<td>JPAU</td>
<td>dst</td>
<td>src</td>
<td></td>
</tr>
<tr>
<td>OP</td>
<td>DAI</td>
<td>dst</td>
<td>src</td>
<td></td>
</tr>
</tbody>
</table>

INSTRUCTION SUMMARY

Note: Assignment of a value is indicated by the symbol "←". For example:

dst ← dst + src

indicates that the source data is added to the destination data and the result is stored in the destination location. The notation "addr (n)" is used to refer to bit (n) of a given operand location. For example:

dst (7)

refers to bit 7 of the destination operand.
# INSTRUCTION SUMMARY (Continued)

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC dst, src</td>
<td>†</td>
<td>1[ ]</td>
<td>* * * * 0 *</td>
<td>INC dst</td>
<td>dst=dst</td>
<td>rE</td>
<td>r = 0 – F</td>
</tr>
<tr>
<td>dst – dst + src + C</td>
<td>dst – dst + 1</td>
<td>R</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD dst, src</td>
<td>†</td>
<td>0[ ]</td>
<td>* * * * 0 *</td>
<td>INCW dst</td>
<td>RR</td>
<td>A0</td>
<td>* * * *</td>
</tr>
<tr>
<td>dst – dst</td>
<td>dst – dst + 1</td>
<td>IR</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AND dst, src</td>
<td>†</td>
<td>5[ ]</td>
<td>* * 0 *</td>
<td>IRET</td>
<td>BF</td>
<td>* * * *</td>
<td></td>
</tr>
<tr>
<td>dst – dst AND src</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALL dst</td>
<td>DA</td>
<td>D6</td>
<td>- - - - - -</td>
<td>JP cc, dst</td>
<td>DA</td>
<td>cD</td>
<td>- - - - -</td>
</tr>
<tr>
<td>SP – SP – 2</td>
<td>IRR</td>
<td>D4</td>
<td></td>
<td>PC – dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@SP – PC,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC – dst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCF</td>
<td>EF</td>
<td>* - - - -</td>
<td>IF cc is true</td>
<td></td>
<td>c = 0 – F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C – NOT C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLR dst</td>
<td>R</td>
<td>B0</td>
<td>- - - - - -</td>
<td>JR cc, dst</td>
<td>RA</td>
<td>cB</td>
<td>- - - - -</td>
</tr>
<tr>
<td>dst = 0</td>
<td>IR</td>
<td>B1</td>
<td></td>
<td>PC – PC + dst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COM dst</td>
<td>R</td>
<td>60</td>
<td>- * * 0</td>
<td></td>
<td>+ dst</td>
<td>+ 0 – F</td>
<td></td>
</tr>
<tr>
<td>dst – NOT dst</td>
<td>IR</td>
<td>61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP dst, src</td>
<td>†</td>
<td>A[ ]</td>
<td>* * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst – src</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DA dst</td>
<td>R</td>
<td>40</td>
<td>* * * X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst – DA dst</td>
<td>IR</td>
<td>41</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEC dst</td>
<td>R</td>
<td>00</td>
<td>- * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst – dst – 1</td>
<td>IR</td>
<td>01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECW dst</td>
<td>RR</td>
<td>80</td>
<td>- * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst – dst – 1</td>
<td>IR</td>
<td>81</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DI</td>
<td>8F</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DJNZ r, dst</td>
<td>RA</td>
<td>rA</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r – r – 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>if r = 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC – PC + dst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range: +127, –128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EI</td>
<td>9F</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMR(7) ← 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HALT</td>
<td>7F</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Instruction and Operation Table

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOP</td>
<td>FF</td>
<td>- - - - - -</td>
<td></td>
</tr>
<tr>
<td>OR dst, src</td>
<td>4[ ]</td>
<td>- * * 0 - -</td>
<td></td>
</tr>
<tr>
<td>POP dst</td>
<td>R 50</td>
<td>- - - - - -</td>
<td></td>
</tr>
<tr>
<td>PUSH src</td>
<td>R 70</td>
<td>- - - - - -</td>
<td></td>
</tr>
<tr>
<td>RCF</td>
<td>CF</td>
<td>0 - - - - -</td>
<td></td>
</tr>
<tr>
<td>RET</td>
<td>AF</td>
<td>- - - - - -</td>
<td></td>
</tr>
<tr>
<td>RL dst</td>
<td>R 90</td>
<td>* * * - -</td>
<td></td>
</tr>
<tr>
<td>RLC dst</td>
<td>R 10</td>
<td>* * * - -</td>
<td></td>
</tr>
<tr>
<td>RR dst</td>
<td>R 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RRC dst</td>
<td>R C0</td>
<td>* * * - -</td>
<td></td>
</tr>
<tr>
<td>SBC dst, src</td>
<td>3[ ]</td>
<td>* * * 1 *</td>
<td></td>
</tr>
<tr>
<td>SCF</td>
<td>DF</td>
<td>1 - - - - -</td>
<td></td>
</tr>
<tr>
<td>SRA dst</td>
<td>R D0</td>
<td>* * * 0 - -</td>
<td></td>
</tr>
<tr>
<td>SRP src</td>
<td>Im 31</td>
<td>- - - - - -</td>
<td></td>
</tr>
</tbody>
</table>

### Instruction and Operation Table

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>STOP</td>
<td>6F</td>
<td>- - - - - -</td>
<td></td>
</tr>
<tr>
<td>SUB dst, src</td>
<td>2[ ]</td>
<td>* * * 1 *</td>
<td></td>
</tr>
<tr>
<td>SWAP dst</td>
<td>R F0</td>
<td>X * * X - -</td>
<td></td>
</tr>
<tr>
<td>TCM dst, src</td>
<td>6[ ]</td>
<td>- * * 0 - -</td>
<td></td>
</tr>
<tr>
<td>TM dst, src</td>
<td>7[ ]</td>
<td>- * * 0 - -</td>
<td></td>
</tr>
<tr>
<td>WDT</td>
<td>5F</td>
<td>X X X X - -</td>
<td></td>
</tr>
<tr>
<td>XOR dst, src</td>
<td>B[ ]</td>
<td>- * * 0 - -</td>
<td></td>
</tr>
</tbody>
</table>

† These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a '[ ]' in this table, and its value is found in the following table to the left of the applicable addressing mode pair.

For example, the opcode of an ADC instruction using the addressing modes r (destination) and Ir (source) is 13.

### Address Mode Table

<table>
<thead>
<tr>
<th>Address Mode</th>
<th>Lower OpcodeNibble</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>[2]</td>
</tr>
<tr>
<td>r Ir</td>
<td>[3]</td>
</tr>
<tr>
<td>R R</td>
<td>[4]</td>
</tr>
<tr>
<td>R Ir</td>
<td>[5]</td>
</tr>
<tr>
<td>R IM</td>
<td>[6]</td>
</tr>
<tr>
<td>IM Im</td>
<td>[7]</td>
</tr>
</tbody>
</table>

8-43
### OPCODE MAP

<table>
<thead>
<tr>
<th>Upper Nibble (Hex)</th>
<th>Lower Nibble (Hex)</th>
<th>Bytes per Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2 or 3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2 or 3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2 or 3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2 or 3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2 or 3</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2 or 3</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>2 or 3</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>2 or 3</td>
</tr>
</tbody>
</table>

#### Legend:
- **R** = 8-bit address
- **r** = 4-bit address
- **R_1** or **r_1** = Dst address
- **R_2** or **r_2** = Src address

#### Note:
The blanks are reserved.

#### Sequence:
Opcode, First Operand, Second Operand

* 2-byte instruction appears as a 3-byte instruction.
## PACKAGE INFORMATION

### OPTION TABLE

<table>
<thead>
<tr>
<th>OPTION #</th>
<th>PACKAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>STANDARD</td>
</tr>
<tr>
<td>02</td>
<td>IDF</td>
</tr>
</tbody>
</table>

### SYMBOL OPT # MILLIMETER INCH

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>OPT #</th>
<th>MILLIMETER</th>
<th>INCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>01</td>
<td>0.51</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>0.81</td>
<td>0.032</td>
</tr>
<tr>
<td>A2</td>
<td>01</td>
<td>3.18</td>
<td>1.25</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>3.94</td>
<td>1.55</td>
</tr>
<tr>
<td>B</td>
<td>01</td>
<td>0.38</td>
<td>0.015</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>0.53</td>
<td>0.021</td>
</tr>
<tr>
<td>B1</td>
<td>01</td>
<td>1.52</td>
<td>0.060</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>1.78</td>
<td>0.070</td>
</tr>
<tr>
<td></td>
<td>03</td>
<td>1.87</td>
<td>0.090</td>
</tr>
<tr>
<td></td>
<td>04</td>
<td>1.52</td>
<td>0.090</td>
</tr>
<tr>
<td></td>
<td>05</td>
<td>1.87</td>
<td>0.090</td>
</tr>
<tr>
<td>C</td>
<td>01</td>
<td>0.23</td>
<td>0.009</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>0.38</td>
<td>0.015</td>
</tr>
<tr>
<td>D</td>
<td>01</td>
<td>3.43</td>
<td>1.35</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>3.94</td>
<td>1.55</td>
</tr>
<tr>
<td>E</td>
<td>01</td>
<td>15.24</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>15.75</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td>03</td>
<td>15.94</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>04</td>
<td>16.10</td>
<td>0.66</td>
</tr>
<tr>
<td>E1</td>
<td>01</td>
<td>13.59</td>
<td>0.53</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>14.10</td>
<td>0.55</td>
</tr>
<tr>
<td></td>
<td>03</td>
<td>13.08</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>04</td>
<td>13.08</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>05</td>
<td>12.83</td>
<td>0.515</td>
</tr>
<tr>
<td></td>
<td>06</td>
<td>13.08</td>
<td>0.515</td>
</tr>
<tr>
<td>eA</td>
<td>01</td>
<td>13.49</td>
<td>0.610</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>16.51</td>
<td>0.650</td>
</tr>
<tr>
<td></td>
<td>03</td>
<td>16.51</td>
<td>0.650</td>
</tr>
<tr>
<td></td>
<td>04</td>
<td>16.51</td>
<td>0.650</td>
</tr>
<tr>
<td></td>
<td>05</td>
<td>16.51</td>
<td>0.650</td>
</tr>
<tr>
<td>L</td>
<td>01</td>
<td>3.18</td>
<td>0.125</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>3.81</td>
<td>0.150</td>
</tr>
<tr>
<td>Q1</td>
<td>01</td>
<td>1.52</td>
<td>0.060</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>1.78</td>
<td>0.075</td>
</tr>
<tr>
<td>S</td>
<td>01</td>
<td>1.52</td>
<td>0.060</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>1.52</td>
<td>0.060</td>
</tr>
</tbody>
</table>

### CONTROLLING DIMENSIONS INCH

28-Pin DIP Package Diagram
28-Pin Window Cerdip Package Diagram
ORDERING INFORMATION

Z86E30 (12 MHz)
28-Pin DIP  28-Pin Cerdip Window Lid
Z86E3012PSC  Z86E3012KSE

Z86E31 (8 MHz)
28-Pin DIP  28-Pin Cerdip Window Lid
Z86E3108PSC  Z86E3108KSE

For fast results, contact your local Zilog sales office for assistance in ordering the part(s) desired.

CODES

Preferred Package
P = Plastic DIP

Longer Lead Time
K= Cerdip Window Lid

Temperature
S = 0°C to +70°C

Speeds
08 = 8 MHz
12 = 12 MHz

Environmental
C = Plastic Standard
E = Hermetic Standard

Example:
Z 86E30 12 P S C
is a Z86E30, 12 MHz, DIP, 0°C to +70°C, Plastic Standard Flow
**FEATURES**

- 8-Bit, CMOS MCU with 4 Kbytes of ROM and 256 Bytes of RAM (236 Bytes for General Purpose)
- Package Styles: 40-Pin DIP, 44-Pin PLCC, 44-Pin QFP
- Software Programmable Low EMI Modes
- Programmable Open-Drain Mode on Port 0, Port 1, and Port 2
- Low-Power Consumption: 40 mW (Typical @ 5.0V)
- Fast Instruction Pointer: 750 ns @ 16 MHz
- Two Standby Modes: STOP and HALT
- 32 Input/Output Lines (Three with Comparator Inputs)
- 25 Digital CMOS Level, Schmitt-Triggered Inputs
- Three Digital CMOS Level Inputs
- Three Expanded Register File Control Registers
- Two Programmable 8-Bit Counter/Timers, Each with a 6-Bit Programmable Prescaler
- Six Vectored, Priority Interrupts from Six Different Sources
- Clock Speeds up to 12 MHz and 16 MHz
- Low Voltage Protection
- Watch-Dog/Power-On Reset Timers
- Permanently Enabled WDT Option
- Two Comparators with Programmable Interrupt Polarity
- On-Chip Oscillator that Accepts a Crystal, Ceramic Resonator, LC, RC, or External Clock Drive
- RAM and ROM Protect
- Programmable Interrupt Polarity
- Auto Latches

**GENERAL DESCRIPTION**

The Z86C40 CCP™ (Consumer Controller Processor) is a member of Zilog's Z8® single-chip microcontroller family with enhanced wake-up circuitry, programmable watchdog timers and low noise/EMI options. These enhancements result in a more efficient, cost-effective design and provide the user with increased design flexibility over the standard Z8 microcontroller core. With 4 Kbytes of ROM and 236 bytes of general-purpose RAM, this low cost, low power consumption CMOS microcontroller offers fast execution, efficient use of memory, sophisticated interrupts, input/output bit manipulation capabilities, and easy hardware/software system expansion.

The Z86C40 architecture is characterized by Zilog's 8-bit microcontroller core with an Expanded Register File (ERF) to allow access to register mapped peripheral and I/O circuits. The Z86C40 offers a flexible I/O scheme, an efficient register and address space structure, and a number of ancillary features that are useful in many industrial, automotive, computer peripherals, and consumer applications.

With ROM/ROMless selectivity, the Z86C40 provides both external memory and pre-programmed ROM, which enables this Z8 microcontroller to be used in high-volume applications, or where code flexibility is required.
GENERAL DESCRIPTION (Continued)

For applications demanding powerful I/O capabilities, the Z86C40 provides 32 pins dedicated to input and output. These lines are grouped into four ports with eight lines each, and are configurable under software control to provide timing, status signals, parallel I/O with or without handshake, and address/data bus for interfacing external memory (Figure 1).

Four basic address spaces are available to support this wide range of configurations: Program Memory, Register File, Data Memory, and Expanded Register File (ERF). The Register File is composed of 236 bytes of general-purpose registers, four I/O port registers, and 15 control and status registers. The Expanded Register File consists of two control registers.

To unburden the system from coping with the real-time tasks, such as counting/timing and data communication, the Z86C40 offers two on-chip counter/timers with a large number of user-selectable modes. Additionally, two on-board comparators allow analog signals to be processed using a common reference voltage.

**Note:** All signals with a preceding front slash, "/", are active Low, e.g., B/W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

<table>
<thead>
<tr>
<th>Connection</th>
<th>Circuit</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>$V_{dd}$</td>
<td>$V_{dd}$</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
<td>$V_{ss}$</td>
</tr>
</tbody>
</table>

![Figure 1. Functional Block Diagram](image-url)
## PIN DESCRIPTION

![Figure 2. 40-Pin DIP Pin Configuration](image)

### Table 1. 40-Pin DIP Pin Identification

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R/W</td>
<td>Read/Write</td>
<td>Output</td>
</tr>
<tr>
<td>2-4</td>
<td>P25-P27</td>
<td>Port 2, Pins 5,6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>5-7</td>
<td>P04-P06</td>
<td>Port 0, Pins 4,5,6</td>
<td>In/Output</td>
</tr>
<tr>
<td>8-9</td>
<td>P14-P15</td>
<td>Port 1, Pins 4,5</td>
<td>In/Output</td>
</tr>
<tr>
<td>11</td>
<td>V&lt;sub&gt;cc&lt;/sub&gt;</td>
<td>Power Supply</td>
<td></td>
</tr>
<tr>
<td>12-13</td>
<td>P16-P17</td>
<td>Port 1, Pins 6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>14</td>
<td>XTAL2</td>
<td>Crystal Oscillator</td>
<td>Output</td>
</tr>
<tr>
<td>15</td>
<td>XTAL1</td>
<td>Crystal Oscillator</td>
<td>Input</td>
</tr>
<tr>
<td>16-18</td>
<td>P31-P33</td>
<td>Port 3, Pins 1,2,3</td>
<td>Input</td>
</tr>
<tr>
<td>19</td>
<td>P34</td>
<td>Port 3, Pin 4</td>
<td>Output</td>
</tr>
<tr>
<td>20</td>
<td>/AS</td>
<td>Address Strobe</td>
<td>Output</td>
</tr>
<tr>
<td>21</td>
<td>/RESET</td>
<td>Reset</td>
<td>Input</td>
</tr>
<tr>
<td>22</td>
<td>P35</td>
<td>Port 3, Pin 5</td>
<td>Output</td>
</tr>
<tr>
<td>23</td>
<td>P37</td>
<td>Port 3, Pin 7</td>
<td>Output</td>
</tr>
<tr>
<td>24</td>
<td>P36</td>
<td>Port 3, Pin 6</td>
<td>Output</td>
</tr>
<tr>
<td>25</td>
<td>P30</td>
<td>Port 3, Pin 0</td>
<td>Input</td>
</tr>
<tr>
<td>26-27</td>
<td>P00-P01</td>
<td>Port 0, Pins 0,1</td>
<td>In/Output</td>
</tr>
<tr>
<td>28-29</td>
<td>P10-P11</td>
<td>Port 1, Pins 0,1</td>
<td>In/Output</td>
</tr>
<tr>
<td>30</td>
<td>P02</td>
<td>Port 0, Pin 2</td>
<td>In/Output</td>
</tr>
<tr>
<td>31</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>32-33</td>
<td>P12-P13</td>
<td>Port 1, Pins 2,3</td>
<td>In/Output</td>
</tr>
<tr>
<td>34</td>
<td>P03</td>
<td>Port 0,Pin 3</td>
<td>In/Output</td>
</tr>
<tr>
<td>35-39</td>
<td>P20-P24</td>
<td>Port 2, Pins 0,1,2,3,4</td>
<td>In/Output</td>
</tr>
<tr>
<td>40</td>
<td>/DS</td>
<td>Data Strobe</td>
<td>Output</td>
</tr>
</tbody>
</table>
Table 2. 44-Pin PLCC Pin Identification

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>NC</td>
<td>Not Connected</td>
<td></td>
</tr>
<tr>
<td>3-4</td>
<td>P12-P13</td>
<td>Port 1, Pins 2,3</td>
<td>In/Output</td>
</tr>
<tr>
<td>5</td>
<td>P03</td>
<td>Port 0, Pin 3</td>
<td>In/Output</td>
</tr>
<tr>
<td>6-10</td>
<td>P20-P24</td>
<td>Port 2, Pins 0,1,2,3,4</td>
<td>In/Output</td>
</tr>
<tr>
<td>11</td>
<td>/DS</td>
<td>Data Strobe</td>
<td>Output</td>
</tr>
<tr>
<td>12</td>
<td>NC</td>
<td>Not Connected</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>R/W</td>
<td>Read/Write</td>
<td>Output</td>
</tr>
<tr>
<td>14-16</td>
<td>P25-P27</td>
<td>Port 2, Pins 5,6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>17-19</td>
<td>P04-P06</td>
<td>Port 0, Pins 4,5,6</td>
<td>In/Output</td>
</tr>
<tr>
<td>20-21</td>
<td>P14-P05</td>
<td>Port 1, Pins 4,5</td>
<td>In/Output</td>
</tr>
<tr>
<td>22</td>
<td>P07</td>
<td>Port 0, Pin 7</td>
<td>In/Output</td>
</tr>
<tr>
<td>23</td>
<td>Vcc</td>
<td>Power Supply</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>NC</td>
<td>Not Connected</td>
<td></td>
</tr>
<tr>
<td>25-26</td>
<td>P16-P17</td>
<td>Port 1, Pins 6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>27</td>
<td>XTAL1</td>
<td>Crystal Oscillator</td>
<td>Output</td>
</tr>
<tr>
<td>28</td>
<td>XTAL2</td>
<td>Crystal Oscillator</td>
<td>Input</td>
</tr>
<tr>
<td>29-31</td>
<td>P31-P33</td>
<td>Port 3, Pins 1,2,3</td>
<td>Input</td>
</tr>
<tr>
<td>32</td>
<td>P34</td>
<td>Port 3, Pin 4</td>
<td>Output</td>
</tr>
<tr>
<td>33</td>
<td>/AS</td>
<td>Address Strobe</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>R/RL</td>
<td>ROM/ROMless select</td>
<td>Input</td>
</tr>
<tr>
<td>35</td>
<td>/RESET</td>
<td>Reset</td>
<td>Input</td>
</tr>
<tr>
<td>36</td>
<td>P35</td>
<td>Port 3, Pin 5</td>
<td>Output</td>
</tr>
<tr>
<td>37</td>
<td>P36</td>
<td>Port 3, Pin 6</td>
<td>Output</td>
</tr>
<tr>
<td>39</td>
<td>P30</td>
<td>Port 3, Pin 0</td>
<td>Input</td>
</tr>
<tr>
<td>40-41</td>
<td>P00-P01</td>
<td>Port 0, Pins 0,1</td>
<td>In/Output</td>
</tr>
<tr>
<td>42-43</td>
<td>P10-P11</td>
<td>Port 1, Pins 0,1</td>
<td>In/Output</td>
</tr>
<tr>
<td>44</td>
<td>P02</td>
<td>Port 0, Pin 2</td>
<td>In/Output</td>
</tr>
</tbody>
</table>
Table 3. 44-Pin QFP Pin Identification

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>P05-P06</td>
<td>Port 0, Pins 5,6</td>
<td>In/Output</td>
</tr>
<tr>
<td>3-4</td>
<td>P14-P05</td>
<td>Port 1, Pins 4,5</td>
<td>In/Output</td>
</tr>
<tr>
<td>5</td>
<td>P07</td>
<td>Port 0, Pin 7</td>
<td>In/Output</td>
</tr>
<tr>
<td>6</td>
<td>Vcc</td>
<td>Power Supply</td>
<td>Not Connected</td>
</tr>
<tr>
<td>7</td>
<td>NC</td>
<td>Ground</td>
<td>Not Connected</td>
</tr>
<tr>
<td>8-9</td>
<td>P16-P17</td>
<td>Port 1, Pins 6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>10</td>
<td>XTAL2</td>
<td>Crystal Oscillator</td>
<td>Output</td>
</tr>
<tr>
<td>11</td>
<td>XTAL1</td>
<td>Crystal Oscillator</td>
<td>Input</td>
</tr>
<tr>
<td>12-14</td>
<td>P31-P33</td>
<td>Port 3, Pins 1,2,3</td>
<td>Input</td>
</tr>
<tr>
<td>15</td>
<td>P34</td>
<td>Port 3, Pin 4</td>
<td>Output</td>
</tr>
<tr>
<td>16</td>
<td>/AS</td>
<td>Address Strobe</td>
<td>Output</td>
</tr>
<tr>
<td>17</td>
<td>R/RL</td>
<td>ROM/ROMStrobe</td>
<td>Input</td>
</tr>
<tr>
<td>18</td>
<td>/RESET</td>
<td>Reset</td>
<td>Input</td>
</tr>
<tr>
<td>19</td>
<td>P35</td>
<td>Port 3, Pin 5</td>
<td>Output</td>
</tr>
<tr>
<td>20</td>
<td>P37</td>
<td>Port 3, Pin 7</td>
<td>Output</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>P36</td>
<td>Port 3, Pin 6</td>
<td>Output</td>
</tr>
<tr>
<td>22</td>
<td>P30</td>
<td>Port 3, Pin 0</td>
<td>Input</td>
</tr>
<tr>
<td>23-24</td>
<td>P00-P01</td>
<td>Port 0, Pin 0,1</td>
<td>In/Output</td>
</tr>
<tr>
<td>25-26</td>
<td>P10-P11</td>
<td>Port 1, Pins 0,1</td>
<td>In/Output</td>
</tr>
<tr>
<td>27</td>
<td>P02</td>
<td>Port 0, Pin 2</td>
<td>In/Output</td>
</tr>
<tr>
<td>28</td>
<td>GND</td>
<td>Ground</td>
<td>Input</td>
</tr>
<tr>
<td>29</td>
<td>NC</td>
<td>Not Connected</td>
<td>Input</td>
</tr>
<tr>
<td>30-31</td>
<td>P12-P13</td>
<td>Port 1, Pins 2,3</td>
<td>Input</td>
</tr>
<tr>
<td>32</td>
<td>P03</td>
<td>Port 0, Pin 3</td>
<td>Output</td>
</tr>
<tr>
<td>33-37</td>
<td>P20-24</td>
<td>Port 2, Pins 0,1,2,3,4</td>
<td>In/Output</td>
</tr>
<tr>
<td>38</td>
<td>/DS</td>
<td>Strobe</td>
<td>Output</td>
</tr>
<tr>
<td>39</td>
<td>NC</td>
<td>Not Connected</td>
<td>Output</td>
</tr>
<tr>
<td>40</td>
<td>R/WW</td>
<td>Read/Write</td>
<td>Output</td>
</tr>
<tr>
<td>41-43</td>
<td>P25-P27</td>
<td>Port 2, Pins 5,6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>44</td>
<td>P04</td>
<td>Port 0, Pin 4</td>
<td>In/Output</td>
</tr>
</tbody>
</table>

Figure 4. 44-Pin QFP Pin Configuration
PIN FUNCTIONS

/ROMless (input, active Low). This pin, when connected to GND, disables the internal ROM and forces the device to function as a Z86C90/C89 ROMless Z8. (Note that, when left unconnected or pulled High to $V_{cc}$, the device functions normally as a Z8 ROM version).

**Note:** When using in ROM Mode in High EMI (noisy) environment, the ROMless pins should be connected directly to $V_{cc}$.

/DS (output, active Low). Data Strobe is activated once for each external memory transfer. For a READ operation, data must be available prior to the trailing edge of /DS. For WRITE operations, the falling edge of /DS indicates that output data is valid.

/AS (output, active Low). Address Strobe is pulsed once at the beginning of each machine cycle for external memory transfer. Address output is from Port 0/Port 1 for all external programs. Memory address transfers are valid at the trailing edge of /AS. Under program control, /AS is placed in the high-impedance state along with Ports 0 and 1, Data Strobe, and Read/Write.

XTAL1 Crystal 1 (time-based input). This pin connects a parallel-resonant crystal, ceramic resonator, LC, or RC network, or an external single-phase clock to the on-chip oscillator input.

XTAL2 Crystal 2 (time-based output). This pin connects a parallel-resonant, crystal, ceramic resonant, LC, or RC network to the on-chip oscillator output.

R/W (output, write Low). Read/Write, the R/W signal is Low when the Z86C40 is writing to the external program or data memory.

Port 0 (P00-P07). Port 0 is an 8-bit, bidirectional, CMOS compatible port. These eight I/O lines are configured under software control as a nibble I/O port (P03-P00 input/output and P07-P04 input/output), or as an address port for interfacing external memory. The input buffers are Schmitt-triggered and nibble-programmed as outputs and can be globally programmed as either push-pull or open-drain. Low EMI output buffers can be globally programmed by the software. Port 0 is placed under handshake control. In this configuration, Port 3, lines P32 and P35 are used as the handshake control /DAV0 and RDY0. Handshake signal direction is dictated by the I/O direction (input or output) of Port 0 of the upper nibble P04-P07. The lower nibble must have the same direction as the upper nibble.

For external memory references, Port 0 provides address bits A11-A8 (lower nibble) or A15-A8 (lower and upper nibble) depending on the required address space. If the address range requires 12 bits or less, the upper nibble of Port 0 can be programmed independently as I/O while the lower nibble is used for addressing. If one or both nibbles are needed for I/O operation, they are configured by writing to the Port 0 mode register.

In ROMless mode, after a hardware reset, Port 0 is configured as address lines A15-A8, and extended timing is set to accommodate slow memory access. The initialization routine can include reconfiguration to eliminate this extended timing mode. (In ROM mode, Port 0 is defined as input after reset.)

Port 0 is set in the high-impedance mode if selected as an address output state along with Port 1 and the control signals /AS, /DS, and R/W (Figure 5).
Figure 5. Port 0 Configuration
PIN FUNCTIONS (Continued)

Port 1 (P10-P17). Port 1 is an 8-bit, bidirectional, CMOS compatible port (Figure 6), with multiplexed Address (A7-A0) and Data (D7-D0) ports. For the Z86C40 ROM device, these eight I/O lines are programmed as inputs or outputs, or can be configured under software control as an Address/Data port for interfacing external memory. The input buffers are Schmitt-triggered and byte-programmed as outputs and can be globally programmed as either push-pull or open-drain. Low EMI output buffers can be globally programmed by the software.

Port 1 may be placed under handshake control. In this configuration, Port 3, lines P33 and P34 are used as the handshake controls RDY1 and /DAV1 (Ready and Data Available). Memory locations greater than 4095 are referenced through Port 1. To interface external memory, Port 1 must be programmed for the multiplexed Address/Data mode. If more than 256 external locations are required, Port 0 outputs the additional lines.

Port 1 can be placed in the high-impedance state along with Port 0, /AS, /DS, and R/W, allowing the Z86C40 to share common resources in multiprocessor and DMA applications.

Figure 6. Port 1 Configuration
Port 2 (P20-P27). Port 2 is an 8-bit, bidirectional, CMOS compatible I/O port. These eight I/O lines are configured under software control as an input or output, independently. Port 2 is always available for I/O operation. The input buffers are Schmitt-triggered. Bits programmed as outputs may be globally programmed as either push-pull or open-drain. Low EMI output buffers can be globally programmed by the software.

Port 2 may be placed under handshake control. In this Handshake Mode, Port 3 lines P31 and P36 are used as the handshake controls lines /DAV2 and RDY2. The handshake signal assignment for Port 3 lines P31 and P36 is dictated by the direction (input or output) assigned to bit 7, Port 2 (Figure 7).

Figure 7. Port 2 Configuration
PIN FUNCTIONS (Continued)

Port 3 (P30-P37). Port 3 is an 8-bit, CMOS compatible, four fixed inputs (P30-P33) and four fixed outputs (P34-P37), and is configured under software control for Input/Output, Counter/Timers, interrupt, port handshake, and Data Memory functions. Port 3, pin 0 input is Schmitt-triggered, and pins P31, P32, and P33 are standard CMOS inputs (no Auto Latches). Pins P34, P35, P36, P37 are push-pull output lines. Low EMI output buffers can be globally programmed by the software.

Two on-board comparators can process analog signals on P31 and P32 with reference to the voltage on P33. The analog function is enabled by programming Port 3 Mode Register (P3M bit 1). For Interrupt functions, Port 3, pin 0 and pin 3 are falling edge interrupt inputs. P31 and P32 are programmable as rising, falling, or both edge triggered interrupts (IRQ register bits 6 and bit 7). P33 is the comparator reference voltage input when in Analog Mode. Access to Counter/Timers 1 is made through P31 (T_in) and P36 (T_out). Handshake lines for Ports 0, 1, and 2 are available on P31 through P36.

Port 3 also provides the following control functions: handshake for Ports 0, 1, and 2 (/DAV and RDY); four external interrupt request signals (IRQ3-IRQ0); timer input and output signals (T_in and T_out); Data Memory Select (/DM, see Table 4, Figure 37).

P34 output can be software-programmed to function as a Data Memory Select (DM). The Port 3 mode register (P3M) bit D3, D4 selects this function. When accessing external Data Memory, the P34 goes active Low; when accessing external program memory, the P34 goes High.

### Table 4. Port 3 Pin Assignments

<table>
<thead>
<tr>
<th>Pin</th>
<th>I/O</th>
<th>CTC1</th>
<th>Analog</th>
<th>Int.</th>
<th>P0 HS</th>
<th>P1 HS</th>
<th>P2 HS</th>
<th>Ext</th>
</tr>
</thead>
<tbody>
<tr>
<td>P30</td>
<td>IN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P31</td>
<td>IN</td>
<td></td>
<td>T_in</td>
<td>AN1</td>
<td>IRQ3</td>
<td></td>
<td></td>
<td>D/R</td>
</tr>
<tr>
<td>P32</td>
<td>IN</td>
<td></td>
<td>AN2</td>
<td>IRQ0</td>
<td></td>
<td>D/R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P33</td>
<td>IN</td>
<td></td>
<td>REF</td>
<td>IRQ1</td>
<td></td>
<td></td>
<td></td>
<td>D/R</td>
</tr>
<tr>
<td>P34</td>
<td>OUT</td>
<td></td>
<td>AN1-OUT</td>
<td></td>
<td></td>
<td></td>
<td>R/D</td>
<td>/DM</td>
</tr>
<tr>
<td>P35</td>
<td>OUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P36</td>
<td>OUT</td>
<td></td>
<td>T_out</td>
<td></td>
<td></td>
<td></td>
<td>R/D</td>
<td></td>
</tr>
<tr>
<td>P37</td>
<td>OUT</td>
<td></td>
<td>AN2-OUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- HS = Handshake Signals
- D = /DAV
- R = RDY

**Auto Latch.** The Auto-Latch instruction puts valid CMOS levels on all CMOS inputs (except P33-P31) that are not externally driven. Whether this level is 0 or 1, cannot be determined. A valid CMOS level, rather than a floating node, reduces excessive supply current flow in the input buffer.

**Comparator Inputs.** Port 3, Pins P31 and P32 each have a comparator front end. The comparator reference voltage, pin P33, is common to both comparators. In analog mode, the P31 and P32 are the positive inputs to the comparators and P33 is the reference voltage supplied to both comparators. In digital mode, pin P33 can be used as a P33 register input or IRQ1 source. P34 and P37 outputs the comparator outputs by software-programming the PCON Reg. bit D0 to 1.

**Note:** Deletion of all Port Auto Latches is available as a ROM Mask option. The Auto Latch Delete option is selected by the customer when the ROM code is submitted.
/RESET (input, active Low). Initializes the MCU. Reset is accomplished either through Power-On Reset, Watch-Dog Timer reset, STOP-Mode Recovery, or external reset. During Power-On Reset and Watch-Dog Reset, the internally generated reset is driving the reset pin Low for the POR time. Any devices driving the reset line must be open-drain to avoid damage from a possible conflict during reset conditions. Pull-up is provided internally.

After the POR time, /RESET is a Schmitt-triggered input. To avoid asynchronous and noisy reset problems, the Z86C40 is equipped with a reset filter of four external clocks (4 TpC). If the external reset signal is less than 4 TpC in duration, no reset occurs. On the fifth clock after the reset is detected, an internal RST signal is latched and held for an internal register count of 18 external clocks, or for the duration of the external reset, whichever is longer. During the reset cycle, /DS is held active Low while /AS cycles at a rate of TpC/2. Program execution begins at location 000C (HEX), 5-10 TpC cycles after the RST is released. For Power-On Reset, the reset output time is T_{POR} ms.

Once program execution begins, /AS and /DS toggles only for external memory accesses. The Z86C40 does not reset WDTMR, SMR, P2M, PCON, and P3M registers on a STOP-Mode Recovery operation.

![Diagram of Port 3 Configuration](image-url)
PIN FUNCTIONS (Continued)

Figure 8b. Port 3 Configuration
FUNCTIONAL DESCRIPTION

The Z86C40 MCU incorporates the following special functions to enhance the standard Z8 architecture to provide the user with increased design flexibility.

RESET. The device is reset in one of the following conditions:

- Power-On Reset
- Watch-Dog Timer
- Stop-Mode Recovery Source
- External Reset
- Low Voltage Recovery

Auto Power-On Reset circuitry is built into the Z86C40, eliminating the need for an external reset circuit to reset upon power-up. The internal pull-up resistor is on the Reset pin, so a pull-up resistor is not required; however, in high EMI (noisy) environment, it is recommended that a small value pull-up resistor be used.

Program Memory. The Z86C40 addresses up to 4 Kbytes of internal program memory and 60 Kbytes of external memory (Figure 9). The first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts. For ROM mode, address 12 to address 4095 consists of on-chip mask-programmed ROM. At addresses 4096 and greater, the Z86C40 executes external program memory fetches through Port 0 and Port 1 in Address/Data mode.

The 4 Kbyte program memory is mask programmable. A ROM protect feature prevents "dumping" of the ROM contents by inhibiting execution of LDC, LCDI, LDE, and LDEI instructions to Program Memory in all modes. ROM look-up tables cannot be used with this feature.

The ROM Protect option is mask-programmable, to be selected by the customer when the ROM code is submitted.

Figure 9. Program Memory Map
**FUNCTIONAL DESCRIPTION (Continued)**

**Data Memory (/DM).** The Z86C40 ROM version can address up to 60 Kbytes of external data memory beginning at location 4096. External data memory may be included with, or separated from, the external program memory space. /DM, an optional I/O function that can be programmed to appear on pin P34, is used to distinguish between data and program memory space (Figure 10). The state of the /DM signal is controlled by the type of instruction being executed. An LDC opcode references PROGRAM (/DM inactive) memory, and an LDE instruction references data (/DM active Low) memory. The user must configure Port 3 Mode Register (P3M) bits D3 and D4 for this mode.

**Expanded Register File (ERF).** The Z86C40 register file has been expanded to allow for additional system control registers, and for mapping of additional peripheral devices along with I/O ports into the register address area. The Z8 register address space R0 through R15 has now been implemented as 16 groups of 16 registers per group (Figure 11). These register groups are known as the Expanded Register File (ERF). Bits 7-4 of register RP select the working register group. Bits 3-0 of register RP select the expanded register group (Figure 12). Three system configuration registers reside in the Expanded Register File at Bank F (PCON, SMR, WDTMR). The rest of the Expanded Register is not physically implemented, and is open for future expansion.

![Figure 10. Data Memory Map](image-url)
Figure 11. Expanded Register File Architecture
**FUNCTIONAL DESCRIPTION (Continued)**

Register File. The register file consists of four I/O port registers, 236 general-purpose registers and 15 control and status registers (R0-R3, R4-239 and R240-R255, respectively), plus three system configuration registers in the expanded register group. The instructions access registers directly or indirectly through an 8-bit address field. This allows a short, 4-bit register address using the Register Pointer (Figure 13). In the 4-bit mode, the register file is divided into 16 working register groups, each occupying 16 continuous locations. The Register Pointer addresses the starting location of the active working register group.

**Figure 12. Register Pointer Register**

**Figure 13. Register Pointer**

*Expanded Register Group (0) is selected in this figure by handling bits D3 to D0 as "0" in Register R253 (RP).*
General-Purpose Registers (GPR). These registers are undefined after the device is powered up. The registers keep their last value after any reset, as long as the reset occurs in the $V_{cc}$ voltage-specified operating range. It will not keep its last state from a $V_{LV}$ reset if the $V_{cc}$ drops below 1.8V.

Note: Register Bank EO-EF is only accessed through working register and indirect addressing modes.

RAM Protect. The upper portion of the RAM's address spaces %80F to %EF (excluding the control registers) are protected from reading and writing. The RAM Protect bit option is mask-programmable and is selected by the customer when the ROM code is submitted. After the mask option is selected, the user activates this feature from the internal ROM code to turn off/on the RAM Protect by loading either a 0 or 1 into the IMR register, bit D6. A 1 in D6 enables RAM Protect.

Stack. The Z86C40 external data memory or the internal register file is used for the stack. The 16-bit Stack Pointer (R254-R255) is used for the external stack which can reside anywhere in the data memory for ROMless mode, but only from 4096 to 65535 in ROM mode. An 8-bit Stack Pointer (R255) is used for the internal stack that resides within the 236 general-purpose registers (R4-R239). SPH is used as a general-purpose register when using internal stack only.

Counter/Timers. There are two 8-bit programmable counter/timers (T0-T1), each driven by its own 6-bit programmable prescaler. The T1 prescaler is driven by internal or external clock sources; however, the T0 prescaler is driven by the internal clock only (Figure 14).

The 6-bit prescalers can divide the input frequency of the clock source by any integer number from 1 to 64. Each prescaler drives its counter, which decrements the value (1 to 256) that has been loaded into the counter. When the counter reaches the end of the count, a timer interrupt request, IRQ4 (T0) or IRQ5 (T1), is generated.

The counters can be programmed to start, stop, restart to continue, or restart from the initial value. The counters can also be programmed to stop upon reaching zero (single pass mode) or to automatically reload the initial value and continue counting (modulo-n continuous mode).

The counters, but not the prescalers, are read at any time without disturbing their value or count mode. The clock source for T1 is user-definable and is either the internal microprocessor clock divide-by-four, or an external signal input through Port 3. The Timer Mode register configures the external timer input (P31) as an external clock, a trigger input that can be retriggerable or non-retriggerable, or as a gate input for the internal clock. The counter/timers can be cascaded by connecting the TO output to the input of T1. TIN Mode is enabled by setting R243 PRE1 Bit D1 to 0.
Figure 14. Counter/Timer Block Diagram
Interrupts. The Z86C40 has six different interrupts from six different sources. These interrupts are maskable, prioritized (Figure 15) and the six sources are divided as follows: four sources are claimed by Port 3 lines P33-P30, and two in counter/timers (Table 5). The Interrupt Mask Register globally or individually enables or disables the six interrupt requests.

![Figure 15. Interrupt Block Diagram](image)

**Table 5. Interrupt Types, Sources, and Vectors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Vector Location</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRQ0</td>
<td>/DAV0, IRQ0</td>
<td>0, 1</td>
<td>External (P32), Rise Fall Edge Triggered</td>
</tr>
<tr>
<td>IRQ1</td>
<td>IRQ1</td>
<td>2, 3</td>
<td>External (P33), Fall Edge Triggered</td>
</tr>
<tr>
<td>IRQ2</td>
<td>/DAV2, IRQ2, Tₕ</td>
<td>4, 5</td>
<td>External (P31), Rise Fall Edge Triggered</td>
</tr>
<tr>
<td>IRQ3</td>
<td>IRQ3</td>
<td>6, 7</td>
<td>External (P30), Fall Edge Triggered</td>
</tr>
<tr>
<td>IRQ4</td>
<td>T₀</td>
<td>8, 9</td>
<td>Internal</td>
</tr>
<tr>
<td>IRQ5</td>
<td>T₁</td>
<td>10, 11</td>
<td>Internal</td>
</tr>
</tbody>
</table>
FUNCTIONAL DESCRIPTION (Continued)

When more than one interrupt is pending, priorities are resolved by a programmable priority encoder that is controlled by the Interrupt Priority register. An interrupt machine cycle is activated when an interrupt request is granted. This action disables all subsequent interrupts, saves the Program Counter and Status Flags, and then branches to the program memory vector location reserved for that interrupt.

All Z86C40 interrupts are vectored through locations in the program memory. This memory location and the next byte contain the 16-bit address of the interrupt service routine for that particular interrupt request. To accommodate polled interrupt systems, interrupt inputs are masked and the Interrupt Request register is polled to determine which of the interrupt requests need service.

An interrupt resulting from AN1 is mapped into IRQ2, and an interrupt from AN2 is mapped into IRQ0. Interrupts IRQ2 and IRQ0 may be rising, falling, or both edge triggered, and are programmable by the user. The software may poll to identify the state of the pin.

Programming bits for the Interrupt Edge Select is located in the IRQ Register (R250), bits D7 and D6. The configuration is shown in Table 6.

<table>
<thead>
<tr>
<th>IRQ</th>
<th>Interrupt Edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>D7</td>
<td>D6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Notes:
F = Falling Edge
R = Rising Edge

Clock. The Z86C40 on-chip oscillator has a high-gain, parallel-resonant amplifier for connection to a crystal, LC, RC, ceramic resonator, or any suitable external clock source (XTAL1 = Input, XTAL2 = Output). The crystal should be AT cut, 16 MHz max., with a series resistance (RS) of less than or equal to 100 Ohms.

The crystal should be connected across XTAL1 and XTAL2 using the vendor’s recommended capacitor values from each pin directly to the device Ground pin to reduce ground noise injection into the oscillator. The RC oscillator option is mask-programmable on the Z86C40 and is selected by the customer at the time when the ROM code is submitted. (Note that the RC option is available up to 8 MHz.) The RC oscillator configuration must be an external resistor connected from XTAL1 to XTAL2, with a frequency-setting capacitor from XTAL1 to Ground (Figure 16).

Note: For better noise immunity, the capacitors should be tied directly to the device Ground pin (VSS).

Figure 16. Oscillator Configuration
**Power-On-Reset (POR)**. A timer circuit clocked by a dedicated on-board RC oscillator is used for the Power-On Reset (POR) timer function. The POR time allows \( V_{cc} \) and the oscillator circuit to stabilize before instruction execution begins.

The POR timer circuit is a one-shot timer triggered by one of three conditions:

1. Power fail to Power OK status.
2. STOP-Mode Recovery (if \( D_5 \) of SMR=1).
3. WDT timeout.

The POR time is specified as \( T_{POR} \). Bit 5 of the STOP-Mode Register determines whether the POR timer is bypassed after STOP-Mode Recovery (typical for external clock, RC/LC oscillators).

**HALT**. HALT turns off the internal CPU clock, but not the XTAL oscillation. The counter/timers and external interrupts IRQ0, IRQ1, IRQ2, and IRQ3 remain active. The devices are recovered by interrupts, either externally or internally generated. An interrupt request must be enabled and executed to exit HALT mode. After the interrupt service routine, the program continues from the instruction after the HALT.

In order to enter STOP (or HALT) mode, it is necessary to first flush the instruction pipeline to avoid suspending execution in mid-instruction. To do this, the user must execute a NOP (opcode=FFH) immediately before the appropriate sleep instruction, i.e.:

```
FF NOP ; clear the pipeline
6F STOP ; enter STOP mode
```

or

```
FF NOP ; clear the pipeline
7F HALT ; enter HALT mode
```

**STOP**. This instruction turns off the internal clock and external crystal oscillation. It also reduces the standby current to 10 \( \mu A \) or less. The STOP mode is terminated by a reset only, either by WDT timeout, POR, SMR recovery, or external reset. This causes the processor to restart the application program at address 000C (HEX).

**Port Configuration Register (PCON)**. The PCON register configures the ports individually; comparator output on Port 3, open-drain on Port 0 and Port 1, low EMI on Ports 0, 1, 2, and 3, and low EMI oscillator. The PCON register is located in the expanded register file at Bank F, location 00 (Figure 17).

**Figure 17. Port Configuration Register (PCON) (Write Only)**
FUNCTIONAL DESCRIPTION (Continued)

Comparator Output Port 3 (D0). Bit 0 controls the comparator use in Port 3. A 1 in this location brings the comparator outputs to P34 and P37, and a 0 releases the Port to its standard I/O configuration. The default value is 0.

Port 1 Open-Drain (D1). Port 1 can be configured as an open-drain by resetting this bit (D1=0) or configured as push-pull active by setting this bit (D1=1). The default value is 1.

Port 0 Open-Drain (D2). Port 0 can be configured as an open-drain by resetting this bit (D2=0) or configured as push-pull active by setting this bit (D2=1). The default value is 1.

Low EMI Port 0 (D3). Port 0 can be configured as a Low EMI Port by resetting this bit (D3=0) or configured as a Standard Port by setting this bit (D3=1). The default value is 1.

Low EMI Port 1 (D4). Port 1 can be configured as a Low EMI Port by resetting this bit (D4=0) or configured as a Standard Port by setting this bit (D4=1). The default value is 1.

Low EMI Port 2 (D5). Port 2 can be configured as a Low EMI Port by resetting this bit (D5=0) or configured as a Standard Port by setting this bit (D5=1). The default value is 1.

Low EMI Port 3 (D6). Port 3 can be configured as a Low EMI Port by resetting this bit (D6=0) or configured as a Standard Port by setting this bit (D6=1). The default value is 1.

Low EMI OSC (D7). This bit of the PCON Register controls the low EMI noise oscillator. A 1 in this location configures the oscillator with standard drive, while a 0 configures the oscillator with low noise drive. The low EMI mode will reduce the drive of the oscillator (OSC). The default value is 1. Note: Maximum external clock frequency of 4 MHz when running in the low EMI oscillator mode.

Low EMI Emission. The Z86C40 can be programmed to operate in a low EMI emission mode in the PCON register. The oscillator and all I/O ports can be programmed as low EMI emission mode independently. Use of this feature results in:

- The pre-drivers slew rate reduced to 10 ns typical.
- Low EMI output drivers have resistance of 200 Ohms (typical).
- Low EMI Oscillator
- Internal SCLK/TCLK=XTAL operation limited to a maximum of 4 MHz - 250 ns cycle time, when Low EMI Oscillator is selected and system clock (SCLK = XTAL, SMR Reg. Bit D1 = 1).
STOP-Mode Recovery Register (SMR). This register selects the clock divide value and determines the mode of STOP-Mode Recovery (Figure 18). All bits are Write Only, except bit 7 which is Read Only. Bit 7 is a flag bit that is hardware set on the condition of STOP recovery and reset by a power-on cycle. Bit 6 controls whether a low level or a high level is required from the recovery source. Bit 5 controls the reset delay after recovery. Bits 2, 3, and 4, or the SMR register, specify the source of the STOP-Mode Recovery signal. Bits 0 and 1 determine the timeout period of the WDT. The SMR is located in Bank F of the Expanded Register Group at address 0BH.

---

**Figure 18. STOP-Mode Recovery Register**

(Write Only Except Bit D7, Which Is Read Only)
**FUNCTIONAL DESCRIPTION (Continued)**

**SCLK/TCLK Divide-by-16 Select (D0).** D0 of the SMR controls a divide-by-16 prescaler of SCLK/TCLK. The purpose of this control is to selectively reduce device power consumption during normal processor execution (SCLK control) and/or HALT mode (where TCLK sources counter/timers and interrupt logic). This bit is reset to D0 = 0 after a Stop-Mode Recovery.

**External Clock Divide-by-Two (D1).** This bit can eliminate the oscillator divide-by-two circuitry. When this bit is 0, the System Clock (SCLK) and Timer Clock (TCLK) are equal to the external clock frequency divided by two. The SCLK/TCLK is equal to the external clock frequency when this bit is set (D1=1). Using this bit together with D7 of PCON further helps lower EMI (i.e., D7 (PCON) = 0, D1 (SMR) = 1). The default setting is zero. Maximum external clock frequency is 4 MHz when SMR Bit D1 = 1 where SCLK/TCLK = XTAL.

**STOP-Mode Recovery Source (D2, D3, and D4).** These three bits of the SMR specify the wake-up source of the STOP recovery (Figure 19 and Table 7).

---

**Figure 19. STOP-Mode Recovery Source**

---

9-24
Table 7. STOP-Mode Recovery Source

<table>
<thead>
<tr>
<th>SMR:432</th>
<th>Operation Description of Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>D4 D3 D2</td>
<td>POR and/or external reset recovery</td>
</tr>
<tr>
<td>0 0 0</td>
<td>P30 transition</td>
</tr>
<tr>
<td>0 0 1</td>
<td>P31 transition (not in Analog Mode)</td>
</tr>
<tr>
<td>0 1 0</td>
<td>P32 transition (not in Analog Mode)</td>
</tr>
<tr>
<td>1 0 0</td>
<td>P33 transition (not in Analog Mode)</td>
</tr>
<tr>
<td>1 0 1</td>
<td>P27 transition</td>
</tr>
<tr>
<td>1 1 0</td>
<td>Logical NOR of P20 through P23</td>
</tr>
<tr>
<td>1 1 1</td>
<td>Logical NOR of P20 through P27</td>
</tr>
</tbody>
</table>

STOP-Mode Recovery Delay Select (D5). This bit, if High, enables the \( T_{POR} \)/RESET delay after Stop-Mode Recovery. The default configuration of this bit is 1. If the "fast" wake up is selected, the Stop-Mode Recovery source is kept active for at least 5 \( T_pC \).

STOP-Mode Recovery Edge Select (D6). A 1 in this bit position indicates that a high level on any one of the recovery sources wakes the Z86C40 from STOP mode. A 0 indicates low-level recovery. The default is 0 on POR (Figure 19).

Cold or Warm Start (D7). This bit is set by the device upon entering STOP mode. A 0 in this bit (cold) indicates that the device resets by POR/WDT RESET. A 1 in this bit (warm) indicates that the device awakens by a SMR source.

Watch-Dog Timer Mode Register (WDTMR). The WDT is a retriggerable one-shot timer that resets the Z8 if it reaches its terminal count. The WDT is initially enabled by executing the WDT instruction and refreshed on subsequent executions of the WDT instruction. The WDT circuit is driven by an on-board RC oscillator or external oscillator from the XTAL1 pin. The POR clock source is selected with bit 4 of the WDT register (Figure 20).

WDT instruction affects the Z (Zero), S (Sign), and V (Overflow) flags. The WDTMR must be written to within 64 internal system clocks. After that, the WDTMR is write protected.

Figure 20. Watch-Dog Timer Mode Register (Write Only)
FUNCTIONAL DESCRIPTION (Continued)

![Diagram of resets and WDT](image)

Figure 21. Resets and WDT
WDT Time Select. (D0,D1). Selects the WDT time period and is configured as shown in Table 8.

Table 8. WDT Time Select

<table>
<thead>
<tr>
<th>D1</th>
<th>D0</th>
<th>Timeout of Internal RC OSC</th>
<th>Timeout of XTAL Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>5 ms min</td>
<td>256 TpC</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>15 ms min</td>
<td>512 TpC</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>25 ms min</td>
<td>1024 TpC</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>100 ms min</td>
<td>4096 TpC</td>
</tr>
</tbody>
</table>

Notes:
TpC = XTAL clock cycle
The default on reset is 15 ms.
Values given are for Vcc = 5.0V.

WDTMR During HALT (D2). This bit determines whether or not the WDT is active during HALT mode. A 1 indicates active during HALT. The default is 1.

WDTMR During STOP (D3). This bit determines whether or not the WDT is active during STOP mode. Since XTAL clock is stopped during STOP mode, the on-board RC must be selected as the clock source to the POR counter. A 1 indicates active during STOP. The default is 1.

Clock Source for WDT (D4). This bit determines which oscillator source is used to clock the internal POR and WDT counter chain. If the bit is a 1, the internal RC oscillator is bypassed and the POR and WDT clock source is driven from the external pin, XTAL1. The default configuration of this bit is 0 which selects the Internal RC oscillator.

WDTMR Register Accessibility. The WDTMR register is accessible only during the first 64 internal system clock cycles from the execution of the first instruction after Power-On Reset, watchdog reset or a Stop-Mode Recovery. After this point, the register cannot be modified by any means, intentional or otherwise. The WDTMR cannot be read and is located in bank F of the Expanded Register Group at address location 0FH (Figure 20).

Note: The WDT can be permanently enabled through a mask programming option. The option is selected by the customer at the time of ROM code submittal. In this mode, WDT is always activated when the device comes out of reset. Execution of the WDT instruction serves to refresh the WDT time-out period. WDT operation in the HALT and STOP modes is controlled by WDTMR programming. If this mask option is not selected at the time of ROM code submission, the WDT must be activated by the user through the WDT instruction and is always disabled by any reset to the device.
FUNCTIONAL DESCRIPTION (Continued)

Low Voltage Protection. An on-board Voltage Comparator checks that \( V_{cc} \) is at the required level to ensure correct operation of the device. Reset is globally driven if \( V_{cc} \) is below the specified voltage (Low Voltage Protection voltage). The minimum operating voltage is varying with the temperature and operating frequency, while the Low Voltage Protection voltage \( (V_{LV}) \) varies with temperature only.

The Low Voltage Protection trip voltage \( (V_{LV}) \) is less than 3V and above 1.4V under the following conditions.

Maximum \( (V_{LV}) \) Conditions:

**Case 1:** \( T_A = -40^\circ C, +105^\circ C \), Internal Clock Frequency equal or less than 1 MHz

**Case 2:** \( T_A = -40^\circ C, +85^\circ C \), Internal Clock Frequency equal or less than 2 MHz

Note: The internal clock frequency is one-half the external clock frequency (SMR D1 = 0).

The device functions normally at or above 3.0V under all conditions. Below 3.0V, the device functions normally until the Low Voltage Protection trip point \( (V_{LV}) \) is reached, for the temperatures and operating frequencies in Case 1 and Case 2, above. The device is guaranteed to function normally at supply voltages above the Low Voltage Protection trip point. The actual Low Voltage Protection trip point is a function of temperature and process parameters (Figure 22).

![Figure 22. Typical Z86C40 Low Voltage Protection Voltage vs Temperature](image-url)
ABSOLUTE MAXIMUM RATINGS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient Temperature under Bias</td>
<td>-40</td>
<td>+105</td>
<td>C</td>
</tr>
<tr>
<td>Storage Temperature</td>
<td>-65</td>
<td>+150</td>
<td>C</td>
</tr>
<tr>
<td>Voltage on any Pin with Respect to Vss [Note 1]</td>
<td>-0.6</td>
<td>+7</td>
<td>V</td>
</tr>
<tr>
<td>Voltage on VDD Pin with Respect to Vss</td>
<td>-0.3</td>
<td>+7</td>
<td>V</td>
</tr>
<tr>
<td>Voltage on XTAL1 and /RESET Pins with Respect to Vss [Note 2]</td>
<td>-0.6</td>
<td>VDD+1</td>
<td>V</td>
</tr>
<tr>
<td>Total Power Dissipation</td>
<td>770</td>
<td></td>
<td>mW</td>
</tr>
<tr>
<td>Maximum Current out of Vss</td>
<td>140</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Maximum Current into VDD</td>
<td>125</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Maximum Current into an Input Pin [Note 3]</td>
<td>-600</td>
<td>+600</td>
<td>µA</td>
</tr>
<tr>
<td>Maximum Current into an Open-Drain Pin [Note 4]</td>
<td>-600</td>
<td>+600</td>
<td>µA</td>
</tr>
<tr>
<td>Maximum Output Current Sunked by Any I/O Pin</td>
<td>25</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Maximum Output Current Sourced by Any I/O Pin</td>
<td>25</td>
<td></td>
<td>mA</td>
</tr>
</tbody>
</table>

Notes:
[1] This applies to all pins except XTAL pins and where otherwise noted.
[2] There is no input protection diode from pin to VDD.
[3] This excludes XTAL pins.
[4] Device pin is not at an output Low state.

Notice:
Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for an extended period may affect device reliability.

STANDARD TEST CONDITIONS

The characteristics listed below apply for standard test conditions as noted. All voltages are referenced to Ground. Positive current flows into the referenced pin (Figure 23).

From Output Under Test

![Test Load Diagram](Image)

Figure 23. Test Load Diagram

CAPACITANCE

T_A = 25°C, VDD = GND = 0V, f = 1.0 MHz; unmeasured pins returned to GND.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input capacitance</td>
<td>0</td>
<td>12 pF</td>
</tr>
<tr>
<td>Output capacitance</td>
<td>0</td>
<td>12 pF</td>
</tr>
<tr>
<td>I/O capacitance</td>
<td>0</td>
<td>12 pF</td>
</tr>
</tbody>
</table>
## DC ELECTRICAL CHARACTERISTICS

<table>
<thead>
<tr>
<th>Sym</th>
<th>Parameter</th>
<th>$V_{cc}$</th>
<th>$T_a = 0^\circ C$ to +70$^\circ C$</th>
<th>$T_a = -40^\circ C$ to +105$^\circ C$</th>
<th>Typical [13] $@ 25^\circ C$</th>
<th>Units</th>
<th>Conditions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{ch}$</td>
<td>Clock Input High Voltage</td>
<td>3.0V</td>
<td>0.7 $V_{cc}$</td>
<td>$V_{cc}$+0.3</td>
<td>0.7 $V_{cc}$</td>
<td>$V_{cc}$+0.3</td>
<td>1.3</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.7 $V_{cc}$</td>
<td>$V_{cc}$+0.3</td>
<td>0.7 $V_{cc}$</td>
<td>$V_{cc}$+0.3</td>
<td>2.5</td>
<td>V</td>
</tr>
<tr>
<td>$V_{cl}$</td>
<td>Clock Input Low Voltage</td>
<td>3.0V</td>
<td>GND-0.3</td>
<td>0.2 $V_{cc}$</td>
<td>GND-0.3</td>
<td>0.2 $V_{cc}$</td>
<td>0.7</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>GND-0.3</td>
<td>0.2 $V_{cc}$</td>
<td>GND-0.3</td>
<td>0.2 $V_{cc}$</td>
<td>1.5</td>
<td>V</td>
</tr>
<tr>
<td>$V_{in}$</td>
<td>Input High Voltage</td>
<td>3.0V</td>
<td>0.7 $V_{cc}$</td>
<td>$V_{cc}$+0.3</td>
<td>0.7 $V_{cc}$</td>
<td>$V_{cc}$+0.3</td>
<td>1.3</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.7 $V_{cc}$</td>
<td>$V_{cc}$+0.3</td>
<td>0.7 $V_{cc}$</td>
<td>$V_{cc}$+0.3</td>
<td>2.5</td>
<td>V</td>
</tr>
<tr>
<td>$V_{il}$</td>
<td>Input Low Voltage</td>
<td>3.0V</td>
<td>GND-0.3</td>
<td>0.2 $V_{cc}$</td>
<td>GND-0.3</td>
<td>0.2 $V_{cc}$</td>
<td>0.7</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>GND-0.3</td>
<td>0.2 $V_{cc}$</td>
<td>GND-0.3</td>
<td>0.2 $V_{cc}$</td>
<td>1.5</td>
<td>V</td>
</tr>
<tr>
<td>$V_{oh}$</td>
<td>Output High Voltage</td>
<td>3.0V</td>
<td>$V_{cc}$-0.4</td>
<td>$V_{cc}$-0.4</td>
<td>$V_{cc}$-0.4</td>
<td>3.1</td>
<td>V</td>
<td>$I_{oh} = -2.0 mA$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{cc}$-0.4</td>
<td>$V_{cc}$-0.4</td>
<td>$V_{cc}$-0.4</td>
<td>4.8</td>
<td>V</td>
<td>$I_{oh} = -2.0 mA$</td>
</tr>
<tr>
<td>$V_{ol1}$</td>
<td>Output Low Voltage</td>
<td>3.0V</td>
<td>0.6</td>
<td>0.6</td>
<td>0.2</td>
<td>V</td>
<td>$I_{ol} = +4.0 mA$</td>
<td>[8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>0.4</td>
<td>0.4</td>
<td>0.1</td>
<td>V</td>
<td>$I_{ol} = +4.0 mA$</td>
<td>[8]</td>
</tr>
<tr>
<td>$V_{ol2}$</td>
<td>Output Low Voltage</td>
<td>3.0V</td>
<td>1.2</td>
<td>1.2</td>
<td>0.3</td>
<td>V</td>
<td>$I_{ol} = +6 mA$</td>
<td>[8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>1.2</td>
<td>1.2</td>
<td>0.3</td>
<td>V</td>
<td>$I_{ol} = +12 mA$</td>
<td>[8]</td>
</tr>
<tr>
<td>$V_{rh}$</td>
<td>Reset Input High Voltage</td>
<td>3.0V</td>
<td>$V_{cc}$</td>
<td>$V_{cc}$</td>
<td>$V_{cc}$</td>
<td>1.5</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>$V_{cc}$</td>
<td>$V_{cc}$</td>
<td>$V_{cc}$</td>
<td>2.1</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
</tr>
<tr>
<td>$V_{rl}$</td>
<td>Reset Input Low Voltage</td>
<td>3.0V</td>
<td>GND-0.3</td>
<td>0.2 $V_{cc}$</td>
<td>GND-0.3</td>
<td>0.2 $V_{cc}$</td>
<td>1.1</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>GND-0.3</td>
<td>0.2 $V_{cc}$</td>
<td>GND-0.3</td>
<td>0.2 $V_{cc}$</td>
<td>1.7</td>
<td>V</td>
</tr>
<tr>
<td>$V_{offset}$</td>
<td>Comparator Input Offset Voltage</td>
<td>3.0V</td>
<td>25</td>
<td>25</td>
<td>10</td>
<td>mV</td>
<td>[10]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
<td>10</td>
<td>mV</td>
<td>[10]</td>
<td></td>
</tr>
<tr>
<td>$I_{il}$</td>
<td>Input Leakage</td>
<td>3.0V</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>2</td>
<td>$&lt; 1$</td>
<td>$\mu A$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>2</td>
<td>$&lt; 1$</td>
<td>$\mu A$</td>
</tr>
<tr>
<td>$I_{ol}$</td>
<td>Output Leakage</td>
<td>3.0V</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>2</td>
<td>$&lt; 1$</td>
<td>$\mu A$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>2</td>
<td>$&lt; 1$</td>
<td>$\mu A$</td>
</tr>
<tr>
<td>$I_{ir}$</td>
<td>Reset Input Current</td>
<td>3.0V</td>
<td>-130</td>
<td>-130</td>
<td>-25</td>
<td>$\mu A$</td>
<td>[6,11]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>-180</td>
<td>-180</td>
<td>-40</td>
<td>$\mu A$</td>
<td>[6,11]</td>
<td></td>
</tr>
<tr>
<td>$I_{cc}$</td>
<td>Supply Current</td>
<td>3.0V</td>
<td>20</td>
<td>20</td>
<td>7</td>
<td>mA</td>
<td>@ 16 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
<td>12</td>
<td>mA</td>
<td>@ 16 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>15</td>
<td>15</td>
<td>5</td>
<td>mA</td>
<td>@ 12 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>20</td>
<td>20</td>
<td>15</td>
<td>mA</td>
<td>@ 12 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td>$I_{cc1}$</td>
<td>Standby Current</td>
<td>3.0V</td>
<td>4.5</td>
<td>4.5</td>
<td>2.0</td>
<td>mA</td>
<td>HALT Mode $V_{in} = 0 V, V_{cc}$ @ 16 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>8</td>
<td>8</td>
<td>3.7</td>
<td>mA</td>
<td>HALT Mode $V_{in} = 0 V, V_{cc}$ @ 16 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>3.4</td>
<td>3.4</td>
<td>1.5</td>
<td>mA</td>
<td>Clock Divide-by-16 @ 16 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>7.0</td>
<td>7.0</td>
<td>2.9</td>
<td>mA</td>
<td>Clock Divide-by-16 @ 16 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td>$I_{cc2}$</td>
<td>Standby Current</td>
<td>3.0V</td>
<td>8</td>
<td>15</td>
<td>1</td>
<td>$\mu A$</td>
<td>STOP Mode $V_{in} = 0 V, V_{cc}$ WDT is not Running</td>
<td>[6,11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>10</td>
<td>20</td>
<td>2</td>
<td>$\mu A$</td>
<td>STOP Mode $V_{in} = 0 V, V_{cc}$ WDT is not Running</td>
<td>[6,11]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0V</td>
<td>500</td>
<td>600</td>
<td>310</td>
<td>$\mu A$</td>
<td>STOP Mode $V_{in} = 0 V, V_{cc}$ WDT is Running</td>
<td>[6,11,14]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>800</td>
<td>1000</td>
<td>600</td>
<td>$\mu A$</td>
<td>STOP Mode $V_{in} = 0 V, V_{cc}$ WDT is Running</td>
<td>[6,11,14]</td>
</tr>
<tr>
<td>Sym</td>
<td>Parameter</td>
<td>$V_{cc}$</td>
<td>$T_a = 0,^\circ C$ to +70$^\circ C$</td>
<td>$T_a = -40,^\circ C$ to +105$^\circ C$</td>
<td>Typical [13] @ 25$^\circ C$</td>
<td>Units</td>
<td>Conditions</td>
<td>Notes</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------</td>
<td>----------</td>
<td>--------------------------------------</td>
<td>---------------------------------</td>
<td>---------------------------------</td>
<td>-------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>Note [3]</td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{CM}$</td>
<td>Input Common Mode</td>
<td>3.0</td>
<td>0</td>
<td>$V_{cc}$-1.0V</td>
<td>0</td>
<td>$V_{cc}$-1.5V</td>
<td>V</td>
<td>[10]</td>
</tr>
<tr>
<td></td>
<td>Voltage Range</td>
<td>5.5</td>
<td>0</td>
<td>$V_{cc}$-1.0V</td>
<td>0</td>
<td>$V_{cc}$-1.5V</td>
<td>V</td>
<td>[10]</td>
</tr>
<tr>
<td>$I_{ALC}$</td>
<td>Auto Latch Low Current</td>
<td>3.0V</td>
<td>8</td>
<td>10</td>
<td>5</td>
<td>$\mu A$</td>
<td>$O V &lt; V_{in} &lt; V_{cc}$</td>
<td>[9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>15</td>
<td>20</td>
<td>11</td>
<td>$\mu A$</td>
<td>$O V &lt; V_{in} &lt; V_{cc}$</td>
<td>[9]</td>
</tr>
<tr>
<td>$I_{ALH}$</td>
<td>Auto Latch High Current</td>
<td>3.0V</td>
<td>-5</td>
<td>-7</td>
<td>-3</td>
<td>$\mu A$</td>
<td>$O V &lt; V_{in} &lt; V_{cc}$</td>
<td>[9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>-8</td>
<td>-10</td>
<td>-6</td>
<td>$\mu A$</td>
<td>$O V &lt; V_{in} &lt; V_{cc}$</td>
<td>[9]</td>
</tr>
<tr>
<td>$T_{POR}$</td>
<td>Power On Reset</td>
<td>3.0V</td>
<td>7</td>
<td>24</td>
<td>7</td>
<td>25</td>
<td>8.5 $mS$</td>
<td>[7]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5V</td>
<td>3</td>
<td>13</td>
<td>3</td>
<td>14</td>
<td>5.0 $mS$</td>
<td>[7]</td>
</tr>
<tr>
<td>$V_{LV}$</td>
<td>$V_{CC}$ Low Voltage</td>
<td>1.7</td>
<td>2.95</td>
<td>1.7</td>
<td>2.6</td>
<td>$V$</td>
<td>$2 , MHz \max \ Int. , CLK , Freq.$</td>
<td>[7]</td>
</tr>
<tr>
<td></td>
<td>Protection Voltage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{OH}$</td>
<td>Output High Voltage</td>
<td>3.3V</td>
<td>$V_{cc}$-0.4</td>
<td>$V_{cc}$-0.4</td>
<td>4.8</td>
<td>$V$</td>
<td>$I_{OH} = -0.5mA$</td>
<td></td>
</tr>
<tr>
<td>(Low EMI Mode)</td>
<td></td>
<td>5.0V</td>
<td>$V_{cc}$-0.4</td>
<td>$V_{cc}$-0.4</td>
<td></td>
<td>$V$</td>
<td>$I_{OH} = -0.5mA$</td>
<td></td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>Output Low Voltage</td>
<td>3.3V</td>
<td>0.6</td>
<td>0.6</td>
<td>0.2</td>
<td>$V$</td>
<td>$I_{OL} = 1.0mA$</td>
<td></td>
</tr>
<tr>
<td>(Low EMI Mode)</td>
<td></td>
<td>5.0V</td>
<td>0.4</td>
<td>0.4</td>
<td>0.1</td>
<td>$V$</td>
<td>$I_{OL} = 1.0mA$</td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**

[1] $I_{CC}$: Clock-Driven, Typ 0.3 $mA$, Max 5 $mA$; 8 $MHz$; Resonator or Crystal, Typ 3.0 $mA$, Max 5 $mA$; 8 $MHz$.


[3] The $V_{cc}$ voltage specification of 3.0V guarantees 3.3V ±0.3V, and the $V_{cc}$ voltage specification of 5.5V guarantees 5.0V ±0.5V.


[7] The $V_{LV}$ increases as the temperature decreases.


[10] For analog comparator, inputs when analog comparators are enabled.

[11] Clock must be forced Low, when XTAL 1 is clock-driven and XTAL2 is floating.


[13] Typical are at $V_{cc}$ = 5.0V and 3.3V.

[14] Internal RC selected.
AC CHARACTERISTICS
External I/O or Memory Read and Write Timing Diagram

Figure 24. External I/O or Memory Read/Write Timing
### AC CHARACTERISTICS

External I/O or Memory Read and Write Timing Table

(SCLK/TCLK = XTAL/2)

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$</th>
<th>$T_A = 0°C$ to $+70°C$</th>
<th>$T_A = -40°C$ to $+105°C$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>12 MHz</td>
<td>16 MHz</td>
<td>12 MHz</td>
</tr>
<tr>
<td>1</td>
<td>TdA(AS)</td>
<td>Address Valid to /AS Rise Delay</td>
<td>3.0 35</td>
<td>25 35</td>
<td>35 25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 35</td>
<td>25 35</td>
<td>35 25</td>
</tr>
<tr>
<td>2</td>
<td>TdAS(A)</td>
<td>/AS Rise to Address Float Delay</td>
<td>3.0 45</td>
<td>35 45</td>
<td>45 35</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 45</td>
<td>35 45</td>
<td>45 35</td>
</tr>
<tr>
<td>3</td>
<td>TdAS(DR)</td>
<td>/AS Rise to Read Data Req'd Valid</td>
<td>3.0 250</td>
<td>180 250</td>
<td>250 180</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 250</td>
<td>180 250</td>
<td>250 180</td>
</tr>
<tr>
<td>4</td>
<td>TwAS</td>
<td>/AS Low Width</td>
<td>3.0 55</td>
<td>40 55</td>
<td>55 40</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 55</td>
<td>40 55</td>
<td>55 40</td>
</tr>
<tr>
<td>5</td>
<td>Td</td>
<td>Address Float to /DS Fall</td>
<td>3.0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>6</td>
<td>TwDSR</td>
<td>/DS (Read) Low Width</td>
<td>3.0 200</td>
<td>135 200</td>
<td>200 135</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 200</td>
<td>135 200</td>
<td>200 135</td>
</tr>
<tr>
<td>7</td>
<td>TwDSW</td>
<td>/DS (Write) Low Width</td>
<td>3.0 110</td>
<td>80 110</td>
<td>110 80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 110</td>
<td>80 110</td>
<td>110 80</td>
</tr>
<tr>
<td>8</td>
<td>TdDSR(DR)</td>
<td>/DS Fall to Read Data Req'd Valid</td>
<td>3.0 150</td>
<td>75 150</td>
<td>150 75</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 150</td>
<td>75 150</td>
<td>150 75</td>
</tr>
<tr>
<td>9</td>
<td>ThDR(DS)</td>
<td>Read Data to /DS Rise Hold Time</td>
<td>3.0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>10</td>
<td>TdDS(A)</td>
<td>/DS Rise to Address Active Delay</td>
<td>3.0 45</td>
<td>50 45</td>
<td>45 50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 55</td>
<td>50 55</td>
<td>55 50</td>
</tr>
<tr>
<td>11</td>
<td>TdDS(AS)</td>
<td>/DS Rise to /AS Fall Delay</td>
<td>3.0 30</td>
<td>35 30</td>
<td>30 35</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 45</td>
<td>35 45</td>
<td>45 55</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 45</td>
<td>25 45</td>
<td>45 25</td>
</tr>
<tr>
<td>13</td>
<td>TdDS(R/W)</td>
<td>/DS Rise to R/W Not Valid</td>
<td>3.0 45</td>
<td>35 45</td>
<td>45 35</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 45</td>
<td>35 45</td>
<td>45 35</td>
</tr>
<tr>
<td>14</td>
<td>TdDW(DSW)</td>
<td>Write Data Valid to /DS Fall (Write)</td>
<td>3.0 55</td>
<td>25 55</td>
<td>55 25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 55</td>
<td>25 55</td>
<td>55 25</td>
</tr>
<tr>
<td>15</td>
<td>TdDS(DW)</td>
<td>/DS Rise to Write Data Not Valid</td>
<td>3.0 45</td>
<td>35 45</td>
<td>45 35</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 55</td>
<td>35 55</td>
<td>55 35</td>
</tr>
<tr>
<td>16</td>
<td>TdA(DR)</td>
<td>Address Valid to Read Data Req'd Valid</td>
<td>3.0 55</td>
<td>35 55</td>
<td>55 35</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 310</td>
<td>230 310</td>
<td>310 230</td>
</tr>
<tr>
<td>17</td>
<td>TdAS(ES)</td>
<td>/AS Rise to /DS Fall Delay</td>
<td>3.0 65</td>
<td>45 65</td>
<td>65 45</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 65</td>
<td>45 65</td>
<td>65 45</td>
</tr>
<tr>
<td>18</td>
<td>TdDI(DS)</td>
<td>Data Input Setup to /DS Rise</td>
<td>0.0 115</td>
<td>60 115</td>
<td>115 60</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 75</td>
<td>60 75</td>
<td>75 60</td>
</tr>
<tr>
<td>19</td>
<td>TdDM(AS)</td>
<td>/DM Valid to /AS Fall Delay</td>
<td>3.0 35</td>
<td>30 35</td>
<td>35 30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 35</td>
<td>30 35</td>
<td>35 30</td>
</tr>
</tbody>
</table>

**Notes:**

[3] The $V_{cc}$ voltage specification of 3.0V guarantees 3.3V ±0.3V, and the $V_{cc}$ voltage specification of 5.5V guarantees 5.0V ±0.5V.

---

**Notes:**

Standard Test Load
All timing references use 0.7 $V_{cc}$ for a logic 1 and 0.2 $V_{cc}$ for a logic 0.
AC ELECTRICAL CHARACTERISTICS

Additional Timing Diagram

Figure 25. Additional Timing
AC ELECTRICAL CHARACTERISTICS
Additional Timing Table (SCLK/TCLK = XTAL/2)

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$</th>
<th>12 MHz</th>
<th>16 MHz</th>
<th>12 MHz</th>
<th>16 MHz</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note[6]</td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TpC</td>
<td>Input Clock Period</td>
<td>3.0V</td>
<td>83</td>
<td>DC</td>
<td>62.5</td>
<td>DC</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>83</td>
<td>DC</td>
<td>62.5</td>
<td>DC</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>2</td>
<td>TrC,TfC</td>
<td>Clock Input Rise &amp; Fall Times</td>
<td>3.0V</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>3</td>
<td>TwC</td>
<td>Input Clock Width</td>
<td>3.0V</td>
<td>41</td>
<td>31</td>
<td>41</td>
<td>31</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>41</td>
<td>31</td>
<td>41</td>
<td>31</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>4</td>
<td>TwTinL</td>
<td>Timer Input Low Width</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>5</td>
<td>TwTinH</td>
<td>Timer Input High Width</td>
<td>3.0V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>TpTin</td>
<td>Timer Input Period</td>
<td>3.0V</td>
<td>8TpC</td>
<td>8TpC</td>
<td>8TpC</td>
<td>8TpC</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>8TpC</td>
<td>8TpC</td>
<td>8TpC</td>
<td>8TpC</td>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TrTin,</td>
<td>Timer Input Rise &amp; Fall Timer</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td>TfTin</td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1]</td>
</tr>
<tr>
<td>8A</td>
<td>TwL</td>
<td>Int. Request Low Time</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td>8B</td>
<td>TwL</td>
<td>Int. Request Low Time</td>
<td>3.0V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>[1,3]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>[1,3]</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>TwH</td>
<td>Int. Request Input High Time</td>
<td>3.0V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>[1,2]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>[1,2]</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Twsm</td>
<td>STOP-Mode Recovery Width Spec</td>
<td>3.0V</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Tost</td>
<td>Oscillator Startup Time</td>
<td>3.0V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>[4]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>5TpC</td>
<td>[4]</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Twdt</td>
<td>Watch-Dog Timer Delay Time</td>
<td>3.0V</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>ms</td>
<td>D0 = 0 [5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>ms</td>
<td>D1 = 0 [5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.0V</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>ms</td>
<td>D0 = 1 [5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>ms</td>
<td>D1 = 0 [5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.0V</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>ms</td>
<td>D0 = 0 [5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>ms</td>
<td>D1 = 1 [5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.0V</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>ms</td>
<td>D0 = 1 [5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>ms</td>
<td>D1 = 1 [5]</td>
</tr>
</tbody>
</table>

Notes:
[1] Timing Reference uses 0.7 $V_{cc}$ for a logic 1 and 0.2 $V_{cc}$ for a logic 0.
[5] Reg. WDTMR.
[6] The $V_{cc}$ voltage specification of 3.0V guarantees 3.3V ±0.3V, and the $V_{cc}$ voltage specification of 5.5V guarantees 5.0V ±0.5V.
### AC ELECTRICAL CHARACTERISTICS
Additional Timing Table (Divide-By-One Mode, SCLK/TCLK = XTAL)

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$</th>
<th>$T_a = 0°C$ to $+70°C$</th>
<th>$T_a = -40°C$ to $+105°C$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note [6]</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>1</td>
<td>$T_{PC}$</td>
<td>Input Clock Period</td>
<td>3.0V</td>
<td>250 DC</td>
<td>250 DC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>250 DC</td>
<td>250 DC</td>
</tr>
<tr>
<td>2</td>
<td>$T_{IC}$</td>
<td>Clock Input Rise &amp; Fall Times</td>
<td>3.0V</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>$T_{WC}$</td>
<td>Input Clock Width</td>
<td>3.0V</td>
<td>125</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>125</td>
<td>125</td>
</tr>
<tr>
<td>4</td>
<td>$T_{TINL}$</td>
<td>Timer Input Low Width</td>
<td>3.0V</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>5</td>
<td>$T_{TINH}$</td>
<td>Timer Input High Width</td>
<td>3.0V</td>
<td>3$T_{PC}$</td>
<td>3$T_{PC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>3$T_{PC}$</td>
<td>3$T_{PC}$</td>
</tr>
<tr>
<td>6</td>
<td>$T_{PTIN}$</td>
<td>Timer Input Period</td>
<td>3.0V</td>
<td>4$T_{PC}$</td>
<td>4$T_{PC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>4$T_{PC}$</td>
<td>4$T_{PC}$</td>
</tr>
<tr>
<td>7</td>
<td>$T_{TRIN}$, $T_{TTRIN}$</td>
<td>Timer Input Rise &amp; Fall Timer</td>
<td>3.0V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>8A</td>
<td>$T_{TIL}$</td>
<td>Int. Request Low Time</td>
<td>3.0V</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>8B</td>
<td>$T_{TIL}$</td>
<td>Int. Request Low Time</td>
<td>3.0V</td>
<td>3$T_{PC}$</td>
<td>3$T_{PC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>3$T_{PC}$</td>
<td>3$T_{PC}$</td>
</tr>
<tr>
<td>9</td>
<td>$T_{TIIH}$</td>
<td>Int. Request Input High Time</td>
<td>3.0V</td>
<td>3$T_{PC}$</td>
<td>2$T_{PC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>3$T_{PC}$</td>
<td>2$T_{PC}$</td>
</tr>
<tr>
<td>10</td>
<td>$T_{TWSM}$</td>
<td>STOP-Mode Recovery Width Spec</td>
<td>3.0V</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>$T_{OST}$</td>
<td>Oscillator Startup Time</td>
<td>3.0V</td>
<td>5$T_{PC}$</td>
<td>5$T_{PC}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>5$T_{PC}$</td>
<td>5$T_{PC}$</td>
</tr>
</tbody>
</table>

Notes:

[1] Timing Reference uses 0.7 $V_{cc}$ for a logic 1 and 0.2 $V_{cc}$ for a logic 0.
[4] SMR=D5 = 1, POR STOP mode delay is on.
[5] Reg. WDTMR.
[6] The $V_{cc}$ voltage specification of 3.0V guarantees 3.3V ±0.3V, and the $V_{cc}$ voltage specification of 5.5V guarantees 5.5V ±0.5V.
[7] SMR D1 = 0.
[8] Maximum frequency for internal system clock is 4 MHz when using XTAL divide-by-one mode.
[9] For RC and LC oscillator, and for oscillator driven by clock driver.
AC ELECTRICAL CHARACTERISTICS
Handshake Timing Diagrams

Figure 26. Input Handshake Timing

Figure 27. Output Handshake Timing
## AC ELECTRICAL CHARACTERISTICS

### Handshake Timing Table

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{CC}$</th>
<th>$T_a = 0°C$ to $+70°C$</th>
<th>$T_a = -40°C$ to $+105°C$</th>
<th>Data Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>12 MHz</td>
<td>16 MHz</td>
<td>12 MHz</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>1</td>
<td>TsDI(DAV)</td>
<td>Data In Setup Time</td>
<td>3.0V</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>ThDI(DAV)</td>
<td>Data In Hold Time</td>
<td>3.0V</td>
<td>160</td>
<td>160</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>115</td>
<td>115</td>
<td>115</td>
</tr>
<tr>
<td>3</td>
<td>TwDAV</td>
<td>Data Available Width</td>
<td>3.0V</td>
<td>155</td>
<td>155</td>
<td>155</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>110</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>4</td>
<td>TdDAV(RDY)</td>
<td>DAV Fall to RDY Fall Delay</td>
<td>3.0V</td>
<td>160</td>
<td>160</td>
<td>160</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>115</td>
<td>115</td>
<td>115</td>
</tr>
<tr>
<td>5</td>
<td>TdDAVd(RDY)</td>
<td>DAV Rise to RDY Rise Delay</td>
<td>3.0V</td>
<td>120</td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>6</td>
<td>TdRDY0(DAV)</td>
<td>RDY Rise to DAV Fall Delay</td>
<td>3.0V</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>TdDO(DAV)</td>
<td>Data Out to DAV Fall Delay</td>
<td>3.0V</td>
<td>42</td>
<td>31</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>42</td>
<td>31</td>
<td>42</td>
</tr>
<tr>
<td>8</td>
<td>TdDAV0(RDY)</td>
<td>DAV Fall to RDY Fall Delay</td>
<td>3.0V</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>TdRDY0(DAV)</td>
<td>RDY Fall to DAV Rise Delay</td>
<td>3.0V</td>
<td>160</td>
<td>160</td>
<td>160</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>115</td>
<td>115</td>
<td>115</td>
</tr>
<tr>
<td>10</td>
<td>TwRDY</td>
<td>RDY Width</td>
<td>3.0V</td>
<td>110</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>11</td>
<td>TdRDY0d(DAV)</td>
<td>RDY Rise to DAV Fall Delay</td>
<td>3.0V</td>
<td>110</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5V</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
</tbody>
</table>

**Notes:**

1. Timing Reference uses 0.7 $V_{CC}$ for a logic 1 and 0.2 $V_{CC}$ for a logic 0.
2. The $V_{CC}$ voltage specification of 3.0V guarantees 3.3V ±0.3V and the $V_{CC}$ voltage specification of 5.5V guarantees 5.0V ±0.5V.
EXPANDED REGISTER FILE CONTROL REGISTERS

**SMR (FH) 08**

- **D7**
  - SCLK/TCLK Divide-by-16
    - 0: OFF
    - 1: ON
  - External Clock Divide by 2
    - 0: SCLK/TCLK = XTAL/2
    - 1: SCLK/TCLK = XTAL

- **D6: D5**
  - STOP-Mode Recovery Source
    - 000: POR Only and/or External Reset
    - 001: P30
    - 010: P31
    - 011: P32
    - 100: P27
    - 101: P2 NOR 0-3
    - 110: P2 NOR 0-7

- **D4**
  - Stop Delay
    - 0: OFF
    - 1: ON

- **D3**
  - Stop Recovery Level
    - 0: Low
    - 1: High

- **D2: D1**
  - Stop Flag (Read only)
    - 0: POR
    - 1: Stop Recovery

- **D0**
  - Default setting after RESET.
  - **Default setting after RESET and STOP-Mode Recovery.**

* Figure 28. Stop-Mode Recovery Register
  (Write Only Except Bit D7, Which Is Read Only)

**WDTMR (F) 0F**

- **D7**
  - WDT TAP
    - 00: 5 ms
    - 01: 15 ms
    - 10: 25 ms
    - 11: 100 ms

- **D6: D5**
  - INT RC OSC
    - External Clock
    - 00: 256 TpC
    - 01: 512 TpC
    - 10: 1024 TpC
    - 11: 4096 TpC

- **D4**
  - WDT During HALT
    - 0: OFF
    - 1: ON

- **D3**
  - WDT During STOP
    - 0: OFF
    - 1: ON

- **D2**
  - XTAL1/INT RC Select for WDT
    - 0: On-Board RC
    - 1: XTAL

- **D1**
  - Reserved (Must be 0)

* Default setting after RESET

* Figure 29. Watch-Dog Timer Mode Register
  (Write Only)
**Z8® CONTROL REGISTERS**

**Figure 30. Port Configuration Register (PCON) (Write Only)**

**Figure 31. Timer Mode Register (F1H: Read/Write)**

**Figure 32. Counter/Timer 1 Register (F2H: Read/Write)**

**Figure 33. Prescaler 1 Register (F3H: Write Only)**

**Figure 34. Counter/Timer 0 Register (F4H: Read/Write)**

**Figure 35. Prescaler 0 Register (F5H: Write Only)**
Figure 36. Port 2 Mode Register (F6H: Write Only)

Figure 37. Port 3 Mode Register (F7H: Write Only)

Figure 38. Port 0 and 1 Mode Register (F8H: Write Only)

Figure 39. Interrupt Priority Register (F9H: Write Only)
Z8 CONTROL REGISTERS (Continued)

R250 IRQ

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRQ0 = P32 Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRQ1 = P33 Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRQ2 = P31 Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRQ3 = P30 Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRQ4 = T0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRQ5 = T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Inter Edge
P31 \ P32 = 00
P31 \ P32 = 01
P31 \ P32 = 10
P31 \ P32 = 11

* This option must be selected when ROM code is submitted for ROM Masking; otherwise this control bit is disabled permanently.

Figure 40. Interrupt Request Register
(FAH: Read/Write)

R251 IMR

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enables IRQ0-IRQ5 (D0 = IRQ0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enables RAM Protect *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enables Interrupts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 41. Interrupt Mask Register
(FBH: Read/Write)

R252 FLAGS

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Flag F1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Flag F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Half Carry Flag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decimal Adjust Flag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overflow Flag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sign Flag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zero Flag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Carry Flag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 42. Flag Register
(FCBH: Read/Write)

R253 RP

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expanded Register File</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Working Register Pointer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 43. Register Pointer
(FDH: Read/Write)

R254 SPH

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stack Pointer Upper Byte (SP0 - SP15)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 44. Stack Pointer High
(FFE: Read/Write)

R255 SPL

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stack Pointer Lower Byte (SP0 - SP7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 45. Stack Pointer Low
(FFF: Read/Write)
INSTRUCTION SET NOTATION

Addressing Modes. The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRR</td>
<td>Indirect register pair or indirect working-register pair address</td>
</tr>
<tr>
<td>Irr</td>
<td>Indirect working-register pair only</td>
</tr>
<tr>
<td>X</td>
<td>Indexed address</td>
</tr>
<tr>
<td>DA</td>
<td>Direct address</td>
</tr>
<tr>
<td>RA</td>
<td>Relative address</td>
</tr>
<tr>
<td>IM</td>
<td>Immediate</td>
</tr>
<tr>
<td>R</td>
<td>Register or working-register address</td>
</tr>
<tr>
<td>r</td>
<td>Working-register address only</td>
</tr>
<tr>
<td>IR</td>
<td>Indirect-register or indirect working-register address</td>
</tr>
<tr>
<td>Ir</td>
<td>Indirect working-register address only</td>
</tr>
<tr>
<td>RR</td>
<td>Register pair or working register pair address</td>
</tr>
</tbody>
</table>

Symbols. The following symbols are used in describing the instruction set.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst</td>
<td>Destination location or contents</td>
</tr>
<tr>
<td>src</td>
<td>Source location or contents</td>
</tr>
<tr>
<td>cc</td>
<td>Condition code</td>
</tr>
<tr>
<td>@</td>
<td>Indirect address prefix</td>
</tr>
<tr>
<td>SP</td>
<td>Stack Pointer</td>
</tr>
<tr>
<td>PC</td>
<td>Program Counter</td>
</tr>
<tr>
<td>FLAGS</td>
<td>Flag register (Control Register 252)</td>
</tr>
<tr>
<td>RP</td>
<td>Register Pointer (R253)</td>
</tr>
<tr>
<td>IMR</td>
<td>Interrupt mask register (R251)</td>
</tr>
</tbody>
</table>

Flags. Control register (R252) contains the following six flags:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Carry flag</td>
</tr>
<tr>
<td>Z</td>
<td>Zero flag</td>
</tr>
<tr>
<td>S</td>
<td>Sign flag</td>
</tr>
<tr>
<td>V</td>
<td>Overflow flag</td>
</tr>
<tr>
<td>D</td>
<td>Decimal-adjust flag</td>
</tr>
<tr>
<td>H</td>
<td>Half-carry flag</td>
</tr>
</tbody>
</table>

Affected flags are indicated by:

- 0: Clear to zero
- 1: Set to one
- *: Set to clear according to operation
- -: Unaffected
- x: Undefined
## CONDITION CODES

<table>
<thead>
<tr>
<th>Value</th>
<th>Mnemonic</th>
<th>Meaning</th>
<th>Flags Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td></td>
<td>Always True</td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>C</td>
<td>Carry</td>
<td>C = 1</td>
</tr>
<tr>
<td>1111</td>
<td>NC</td>
<td>No Carry</td>
<td>C = 0</td>
</tr>
<tr>
<td>0110</td>
<td>Z</td>
<td>Zero</td>
<td>Z = 1</td>
</tr>
<tr>
<td>1110</td>
<td>NZ</td>
<td>Not Zero</td>
<td>Z = 0</td>
</tr>
<tr>
<td>1101</td>
<td>PL</td>
<td>Plus</td>
<td>S = 0</td>
</tr>
<tr>
<td>0101</td>
<td>MI</td>
<td>Minus</td>
<td>S = 1</td>
</tr>
<tr>
<td>0100</td>
<td>OV</td>
<td>Overflow</td>
<td>V = 1</td>
</tr>
<tr>
<td>1100</td>
<td>NOV</td>
<td>No Overflow</td>
<td>V = 0</td>
</tr>
<tr>
<td>0110</td>
<td>EQ</td>
<td>Equal</td>
<td>Z = 1</td>
</tr>
<tr>
<td>1110</td>
<td>NE</td>
<td>Not Equal</td>
<td>Z = 0</td>
</tr>
<tr>
<td>1001</td>
<td>GE</td>
<td>Greater Than or Equal</td>
<td>(S XOR V) = 0</td>
</tr>
<tr>
<td>0001</td>
<td>LT</td>
<td>Less than</td>
<td>(S XOR V) = 1</td>
</tr>
<tr>
<td>1010</td>
<td>GT</td>
<td>Greater Than</td>
<td>[Z OR (S XOR V)] = 0</td>
</tr>
<tr>
<td>0010</td>
<td>LE</td>
<td>Less Than or Equal</td>
<td>[Z OR (S XOR V)] = 1</td>
</tr>
<tr>
<td>1111</td>
<td>UGE</td>
<td>Unsigned Greater Than or Equal</td>
<td>C = 0</td>
</tr>
<tr>
<td>0111</td>
<td>ULT</td>
<td>Unsigned Less Than</td>
<td>C = 1</td>
</tr>
<tr>
<td>1011</td>
<td>UGT</td>
<td>Unsigned Greater Than</td>
<td>(C = 0 AND Z = 0) = 1</td>
</tr>
<tr>
<td>0011</td>
<td>ULE</td>
<td>Unsigned Less Than or Equal</td>
<td>(C OR Z) = 1</td>
</tr>
<tr>
<td>0000</td>
<td>F</td>
<td>Never True (Always False)</td>
<td></td>
</tr>
</tbody>
</table>
INSTRUCTION FORMATS

**One-Byte Instructions**

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>SRC</th>
<th>DEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLR, CPL, DA, DEC, DECW, INC, INCW, POP, PUSH, RL, RLC, RR, RRC, SRA, SWAP</td>
<td>JP, CALL (Indirect)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPC</td>
<td>VALUE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPC</td>
<td>LOW, LOWE, LOEI, LDEI, LDC, LDCl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPC</td>
<td>DJnz, JR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Two-Byte Instructions**

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>SRC</th>
<th>DEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC, ADD, AND, CP, LD, OR, SBC, SUB, TM, TM, XOR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADC, ADD, AND, CP, LD, OR, SBC, SUB, TM, TM, XOR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPC</td>
<td>VALUE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Three-Byte Instructions**

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>SRC</th>
<th>DEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOAD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**INSTRUCTION SUMMARY**

Note: Assignment of a value is indicated by the symbol "<-". For example:

- \( \text{dst} \leftarrow \text{dst} + \text{src} \)

indicates that the source data is added to the destination data and the result is stored in the destination location. The notation "addr (n)" is used to refer to bit (n) of a given operand location. For example:

- \( \text{dst} (7) \)

refers to bit 7 of the destination operand.
### INSTRUCTION SUMMARY (Continued)

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ADC dst, src</strong></td>
<td>†</td>
<td>1[ ]</td>
<td>* * * 0 *</td>
<td><strong>INC dst</strong></td>
<td></td>
<td>r</td>
<td>rE</td>
</tr>
<tr>
<td>dst←dst + src + C</td>
<td></td>
<td></td>
<td></td>
<td>dst←dst + 1</td>
<td></td>
<td>r</td>
<td>0 - F</td>
</tr>
<tr>
<td><strong>ADD dst, src</strong></td>
<td>†</td>
<td>0[ ]</td>
<td>* * * 0 *</td>
<td><strong>INCW dst</strong></td>
<td></td>
<td>RR</td>
<td>A0</td>
</tr>
<tr>
<td>dst←dst + src</td>
<td></td>
<td></td>
<td></td>
<td>dst←dst + 1</td>
<td></td>
<td></td>
<td>A1</td>
</tr>
<tr>
<td><strong>AND dst, src</strong></td>
<td>†</td>
<td>5[ ]</td>
<td>* * 0</td>
<td><strong>IRET</strong></td>
<td></td>
<td>BF</td>
<td>* * * * *</td>
</tr>
<tr>
<td>dst←dst AND src</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CALL dst</strong></td>
<td>DA</td>
<td>D6</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP←SP - 2</td>
<td>IRR</td>
<td>D4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@SP←PC,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CCF</strong></td>
<td>EF</td>
<td>* * * *</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C←NOT C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CLR dst</strong></td>
<td>R</td>
<td>B0</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst←0</td>
<td>IR</td>
<td>B1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>COM dst</strong></td>
<td>R</td>
<td>60</td>
<td>* * 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst←NOT dst</td>
<td>IR</td>
<td>61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CP dst, src</strong></td>
<td>†</td>
<td>A[ ]</td>
<td>* * * * -</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst ← src</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DA dst</strong></td>
<td>R</td>
<td>40</td>
<td>* * * X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst←DA dst</td>
<td>IR</td>
<td>41</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DEC dst</strong></td>
<td>R</td>
<td>00</td>
<td>* * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst←dst - 1</td>
<td>IR</td>
<td>01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DECW dst</strong></td>
<td>RR</td>
<td>80</td>
<td>* * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst←dst - 1</td>
<td>IR</td>
<td>81</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DI</strong></td>
<td>8F</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMR(7)←0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DJNZ r, dst</strong></td>
<td>RA</td>
<td>rA</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r←r - 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>if r ≠ 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←PC + dst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range: +127, -128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>EI</strong></td>
<td>9F</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMR(7)←1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HALT</strong></td>
<td>7F</td>
<td>- - - - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instruction and Operation</td>
<td>Address and Opcode</td>
<td>Address Mode and Operation</td>
<td>Address Mode and Operation</td>
<td>Address Mode and Operation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------</td>
<td>---------------------------</td>
<td>---------------------------</td>
<td>---------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OR dst, src</td>
<td>4[ ]</td>
<td>R</td>
<td>50</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POP dst</td>
<td>50</td>
<td>IR</td>
<td>51</td>
<td>IR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PUSH src, SP+1;</td>
<td>50</td>
<td>IR</td>
<td>51</td>
<td>IR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWAP dst</td>
<td>R</td>
<td>F0</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SBC dst, src</td>
<td>3[ ]</td>
<td>R</td>
<td>D0</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCF</td>
<td>DF</td>
<td>R</td>
<td>IR</td>
<td>IR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRA dst</td>
<td>D0</td>
<td>R</td>
<td>IR</td>
<td>IR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRP src</td>
<td>31</td>
<td>Im</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

† These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a ‘[ ]’ in this table, and its value is found in the following table to the left of the applicable addressing mode pair.

For example, the opcode of an ADC instruction using the addressing modes r (destination) and Ir (source) is 13.
## OPCODE MAP

<table>
<thead>
<tr>
<th>Lower Nibble (Hex)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DEC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DEC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>INC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>INC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JP</td>
<td>SRP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>POP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>POP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PUSH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PUSH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R2</td>
<td>IR2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>IR1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Legend:
- **R** = 8-bit Address
- **r** = 4-bit Address
- **R1** or **r1** = Dest Address
- **R2** or **r2** = Src Address

**Sequence:**
- Opcode, First Operand, Second Operand

**Note:** Blanks are reserved.

*2-byte instruction appears as a 3-byte instruction*
Z86C40 CMOS Z8'4K ROM CCP™
CONSUMER CONTROLLER PROCESSOR

PACKAGE INFORMATION (Continued)

44-Pin QFP Package Diagram
# ORDERING INFORMATION

## Z86C40 (12 MHz)

<table>
<thead>
<tr>
<th>Package Type</th>
<th>Standard Temperature</th>
<th>Extended Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>40-Pin DIP</td>
<td>Z86C4012PSC</td>
<td>Z86C4012PEC</td>
</tr>
<tr>
<td>40-Pin PLCC</td>
<td>Z86C4012VSC</td>
<td>Z86C4012VEC</td>
</tr>
<tr>
<td>44-Pin QFP</td>
<td>Z86C4012FSC</td>
<td>Z86C4012FEC</td>
</tr>
</tbody>
</table>

## Z86C40 (16 MHz)

<table>
<thead>
<tr>
<th>Package Type</th>
<th>Standard Temperature</th>
<th>Extended Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>40-Pin DIP</td>
<td>Z86C4016PSC</td>
<td>Z86C4016PEC</td>
</tr>
<tr>
<td>40-Pin PLCC</td>
<td>Z86C4016VSC</td>
<td>Z86C4016VEC</td>
</tr>
<tr>
<td>44-Pin QFP</td>
<td>Z86C4016FSC</td>
<td>Z86C4016FEC</td>
</tr>
</tbody>
</table>

For fast results, contact your local Zilog sales office for assistance in ordering the part desired.

## CODES

### Preferred Package
- **P** = Plastic DIP
- **V** = Plastic Chip Carrier

### Longer Lead Time
- **F** = Plastic Quad Flat Pack

### Preferred Temperature
- **S** = 0°C to +70°C

### Longer Lead Time
- **E** = -40°C to +105°C

### Speeds
- **08** = 8 MHz
- **16** = 16 MHz

### Environmental
- **C** = Plastic Standard

### Example:

```
| Z | 86C40 | 16 | P | S | C |
```

is a Z86C40, 16 MHz, DIP, 0°C to +70°C, Plastic Standard Flow

- Environmental Flow
- Temperature
- Package
- Speed
- Product Number
- Zilog Prefix
FEATURES

- Low Cost, 8-Bit CMOS OTP Microcontroller, with 4 Kbytes of One-Time PROM and 236 Bytes of RAM
- Package Styles: 40-Pin DIP, 44-Pin PLCC, 44-Pin QFP, 40-Pin CerDIP Window Lid
- 4.5V to 5.5V Operating Range
- Clock Speeds up to 12 MHz
- Software Programmable Low EMI Mode
- Software Enabled Watch-Dog Timer
- Pull-Up Active/Open-Drain Programmable on Port 0, Port 1 and Port 2
- Programmable RC Oscillator, EPROM Protect, and RAM Protect
- Low Power Consumption: 60 mW
- Fast Instruction Pointer: 0.6 µs

GENERAL DESCRIPTION

The Z86E40 OTP (One-Time-Programmable) CCP™ (Consumer Controller Processors) is a member of Zilog's Z8® single-chip microcontroller family with enhanced wake-up circuitry, programmable watch-dog timers and low noise/EMI options. These enhancements result in a more efficient, cost-effective design and provide the user with increased design flexibility over the standard Z8 microcontroller core. With 4 Kbytes of One-Time-PROM and 236 bytes of general-purpose RAM, this low cost, low power consumption CMOS microcontroller offers fast execution, efficient use of memory, sophisticated interrupts, input/output bit manipulation capabilities, and easy hardware/software system expansion.

The Z86E40 architecture is characterized by Zilog's 8-bit microcontroller core with an expanded register file to allow easy access to register mapped peripheral and I/O circuits.

The Z86E40 has 32 pins dedicated to input and output for applications demanding powerful I/O capabilities. These lines are grouped into four ports, eight lines per port, and are configurable under software control to provide timing, status signals, and parallel I/O with or without handshake, and address/data bus for interfacing external memory.
GENERAL DESCRIPTION (Continued)

Four basic address spaces are available to support this wide range of configurations: Program Memory, Data Memory, Register File, and Expanded Register File (ERF). The Register File is composed of 236 bytes of general-purpose registers, four I/O port registers, and 15 control and status registers. The Expanded Register File consists of three control registers.

To unburden the system from coping with the real-time tasks such as counting/timing and input/output data communication, the Z86E40 offers two on-chip counter/timers with a large number of user selectable modes, and two on-board comparators to process analog signals with a common reference voltage (Figures 1 and 2).

Offered in a variety of package styles, such as 40-pin DIP, 44-pin PLCC, 44-pin QFP and 40-pin Cerdp Window Lid, the Z86E40 is well-suited for a wide range of applications.

Notes:
All Signals with a preceding front slash, "/", are active Low, e.g., B/W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

<table>
<thead>
<tr>
<th>Connection</th>
<th>Circuit</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>V_{cc}</td>
<td>V_{cc}</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
<td>V_{ss}</td>
</tr>
</tbody>
</table>

Figure 1. Z86E40 Functional Block Diagram
Figure 2. Z86E40 EPROM Programming Block Diagram
Table 1. 40-Pin DIP Pin Identification

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R/W</td>
<td>Read/Write</td>
<td>Output</td>
</tr>
<tr>
<td>2-4</td>
<td>P25-P27</td>
<td>Port 2, Pins 5,6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>5-7</td>
<td>P04-P06</td>
<td>Port 0, Pins 4,5,6</td>
<td>In/Output</td>
</tr>
<tr>
<td>8-9</td>
<td>P14-P15</td>
<td>Port 1, Pins 4,5</td>
<td>In/Output</td>
</tr>
<tr>
<td>10</td>
<td>P07</td>
<td>Port 0, Pin 7</td>
<td>In/Output</td>
</tr>
<tr>
<td>11</td>
<td>Vcc</td>
<td>Power Supply</td>
<td></td>
</tr>
<tr>
<td>12-13</td>
<td>P16-P17</td>
<td>Port 1, Pins 6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>14</td>
<td>XTAL2</td>
<td>Crystal Oscillator</td>
<td>Output</td>
</tr>
<tr>
<td>15</td>
<td>XTAL1</td>
<td>Crystal Oscillator</td>
<td>Input</td>
</tr>
<tr>
<td>16-18</td>
<td>P31-P33</td>
<td>Port 3, Pins 1,2,3</td>
<td>Input</td>
</tr>
<tr>
<td>19</td>
<td>P34</td>
<td>Port 3, Pin 4</td>
<td>Output</td>
</tr>
<tr>
<td>20</td>
<td>/AS</td>
<td>Address Strobe</td>
<td>Output</td>
</tr>
<tr>
<td>21</td>
<td>/RESET</td>
<td>Reset</td>
<td>Input</td>
</tr>
<tr>
<td>22</td>
<td>P35</td>
<td>Port 3, Pin 5</td>
<td>Output</td>
</tr>
<tr>
<td>23</td>
<td>P37</td>
<td>Port 3, Pin 7</td>
<td>Output</td>
</tr>
<tr>
<td>24</td>
<td>P36</td>
<td>Port 3, Pin 6</td>
<td>Output</td>
</tr>
<tr>
<td>25</td>
<td>P30</td>
<td>Port 3, Pin 0</td>
<td>Input</td>
</tr>
<tr>
<td>26-27</td>
<td>P00-P01</td>
<td>Port 0, Pins 0,1</td>
<td>In/Output</td>
</tr>
<tr>
<td>28-29</td>
<td>P10-P11</td>
<td>Port 1, Pins 0,1</td>
<td>In/Output</td>
</tr>
<tr>
<td>30</td>
<td>P02</td>
<td>Port 0, Pin 2</td>
<td>In/Output</td>
</tr>
<tr>
<td>31</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>32-33</td>
<td>P12-P13</td>
<td>Port 1, Pins 2,3</td>
<td>In/Output</td>
</tr>
<tr>
<td>34</td>
<td>P03</td>
<td>Port 0, Pin 3</td>
<td>In/Output</td>
</tr>
<tr>
<td>35-39</td>
<td>P20-P24</td>
<td>Port 2, Pins 0,1,2,3,4</td>
<td>In/Output</td>
</tr>
<tr>
<td>40</td>
<td>/DS</td>
<td>Data Strobe</td>
<td>Output</td>
</tr>
</tbody>
</table>

Note:
* Pin Configuration and Identification identical on DIP and Cerdip Window Lid style packages.
Figure 4. 44-Pin PLCC Pin Configuration (Standard Mode)

Table 2. 44-Pin PLCC Pin Identification

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</td>
<td>Ground</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>P12-P13</td>
<td>Port 1, Pins 2,3</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>P03</td>
<td>Port 0, Pin 3</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>P20-P24</td>
<td>Port 2, Pins 0,1,2,3,4</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>/DS</td>
<td>Data Strobe</td>
<td>Output</td>
<td></td>
</tr>
<tr>
<td>NC</td>
<td>No Connection</td>
<td>Output</td>
<td></td>
</tr>
<tr>
<td>R/W</td>
<td>Read/Write</td>
<td>Output</td>
<td></td>
</tr>
<tr>
<td>P25-P27</td>
<td>Port 2, Pins 5,6,7</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>P04-P06</td>
<td>Port 0, Pins 4,5,6</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>P14-P05</td>
<td>Port 1, Pins 4,5</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>P07</td>
<td>Port 0, Pin 7</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>Vcc</td>
<td>Power Supply</td>
<td>Output</td>
<td></td>
</tr>
<tr>
<td>P16-P17</td>
<td>Port 1, Pins 6,7</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>XTAL2</td>
<td>Crystal Oscillator</td>
<td>Output</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>XTAL1</td>
<td>Crystal Oscillator</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>P31-P33</td>
<td>Port 3, Pins 1,2,3</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>P34</td>
<td>Port 3, Pin 4</td>
<td>Output</td>
<td></td>
</tr>
<tr>
<td>/AS</td>
<td>Address Strobe</td>
<td>Output</td>
<td></td>
</tr>
<tr>
<td>R/RL</td>
<td>ROM/ROMless select</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>/RESET</td>
<td>Reset</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>P0-P1</td>
<td>Port 0, Pins 0,1</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>P00-P01</td>
<td>Port 0, Pins 0,1</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>P10-P11</td>
<td>Port 1, Pins 0,1</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>P02</td>
<td>Port 0, Pin 2</td>
<td>In/Output</td>
<td></td>
</tr>
</tbody>
</table>

10-5
### Table 3. 44-Pin QFP Pin Identification

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>P05-P06</td>
<td>Port 0, Pins 5,6</td>
<td>In/Output</td>
</tr>
<tr>
<td>3-4</td>
<td>P14-P05</td>
<td>Port 1, Pins 4,5</td>
<td>In/Output</td>
</tr>
<tr>
<td>5</td>
<td>P07</td>
<td>Port 0, Pin 7</td>
<td>In/Output</td>
</tr>
<tr>
<td>6-7</td>
<td>Vcc</td>
<td>Power Supply</td>
<td>In/Output</td>
</tr>
<tr>
<td>8-9</td>
<td>P16-P17</td>
<td>Port 1, Pins 6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>10</td>
<td>XTAL2</td>
<td>Crystal Oscillator</td>
<td>Output</td>
</tr>
<tr>
<td>11</td>
<td>XTAL1</td>
<td>Crystal Oscillator</td>
<td>Input</td>
</tr>
<tr>
<td>12-14</td>
<td>P31-P13</td>
<td>Port 3, Pins 1,2,3</td>
<td>Input</td>
</tr>
<tr>
<td>15</td>
<td>P34</td>
<td>Port 3, Pin 4</td>
<td>Output</td>
</tr>
<tr>
<td>16</td>
<td>/AS</td>
<td>Address Strobe</td>
<td>Output</td>
</tr>
<tr>
<td>17</td>
<td>/R/RL</td>
<td>ROM/ROMless select</td>
<td>Input</td>
</tr>
<tr>
<td>18</td>
<td>/RESET</td>
<td>Reset</td>
<td>Input</td>
</tr>
<tr>
<td>19</td>
<td>P35</td>
<td>Port 3, Pin 5</td>
<td>Output</td>
</tr>
<tr>
<td>20</td>
<td>P37</td>
<td>Port 3, Pin 7</td>
<td>Output</td>
</tr>
<tr>
<td>21</td>
<td>P36</td>
<td>Port 3, Pin 6</td>
<td>Output</td>
</tr>
<tr>
<td>22</td>
<td>P30</td>
<td>Port 3, Pin 0</td>
<td>Input</td>
</tr>
<tr>
<td>23-24</td>
<td>P00-P01</td>
<td>Port 0, Pin 0,1</td>
<td>In/Output</td>
</tr>
<tr>
<td>25-26</td>
<td>P10-P11</td>
<td>Port 1, Pins 0,1,2,3,4</td>
<td>In/Output</td>
</tr>
<tr>
<td>27</td>
<td>P02</td>
<td>Port 0, Pin 2</td>
<td>In/Output</td>
</tr>
<tr>
<td>28-29</td>
<td>GND</td>
<td>Ground</td>
<td>Input</td>
</tr>
<tr>
<td>30-31</td>
<td>P12-P13</td>
<td>Port 1, Pins 2,3</td>
<td>In/Output</td>
</tr>
<tr>
<td>32</td>
<td>P36</td>
<td>Port 0, Pin 3</td>
<td>Input</td>
</tr>
<tr>
<td>33-37</td>
<td>P20-4</td>
<td>Port 2, Pins 0,1,2,3,4</td>
<td>In/Output</td>
</tr>
<tr>
<td>38</td>
<td>/DS</td>
<td>Data Strobe</td>
<td>Output</td>
</tr>
<tr>
<td>39</td>
<td>NC</td>
<td>No Connection</td>
<td>Input</td>
</tr>
<tr>
<td>40</td>
<td>R/W</td>
<td>Read/Write</td>
<td>Output</td>
</tr>
<tr>
<td>41-43</td>
<td>P25-P27</td>
<td>Port 2, Pins 5,6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>44</td>
<td>P04</td>
<td>Port 0, Pin 4</td>
<td>In/Output</td>
</tr>
</tbody>
</table>
Table 4. 40-Pin DIP Package Pin Identification*

<table>
<thead>
<tr>
<th>EPROM Mode</th>
<th>Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NC</td>
<td></td>
<td>No Connection</td>
<td></td>
</tr>
<tr>
<td>2-4</td>
<td>D5-D7</td>
<td>Data 5,6,7</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>5-7</td>
<td>A4-A6</td>
<td>Address 4,5,6</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>8-9</td>
<td>NC</td>
<td>No Connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>A7</td>
<td>Address 7</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>V&lt;sub&gt;CC&lt;/sub&gt;</td>
<td>Power Supply</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12-14</td>
<td>NC</td>
<td>No Connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>/CE</td>
<td>Chip Select</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>/OE</td>
<td>Output Enable</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>EPM</td>
<td>EPROM Prog. Mode</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>V&lt;sub&gt;pp&lt;/sub&gt;</td>
<td>Prog. Voltage</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>A8</td>
<td>Address 8</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>20-21</td>
<td>NC</td>
<td>No Connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>A9</td>
<td>Address 9</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>A11</td>
<td>Address 11</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>A10</td>
<td>Address 10</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>/PGM</td>
<td>Prog. Mode</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>26-27</td>
<td>A0-A1</td>
<td>Address 0,1</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>28-29</td>
<td>NC</td>
<td>No Connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>A2</td>
<td>Address 2</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>GND</td>
<td>Ground</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32-33</td>
<td>NC</td>
<td>No Connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>A3</td>
<td>Address 3</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>35-39</td>
<td>D0-D4</td>
<td>Data 0,1,2,3,4</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>NC</td>
<td>No Connection</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note:
* Pin Configuration and Description identical on DIP and Cerdin Window Lid style packages.
Table 5. 44-Pin PLCC Pin Identification

<table>
<thead>
<tr>
<th>EPROM Programming Mode Pin</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>3-4</td>
<td>NC</td>
<td>No Connection</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>A3</td>
<td>Address 3</td>
<td>Input</td>
</tr>
<tr>
<td>6-10</td>
<td>D0-D4</td>
<td>Data 0,1,2,3,4</td>
<td>In/Output</td>
</tr>
<tr>
<td>11-13</td>
<td>NC</td>
<td>No Connection</td>
<td></td>
</tr>
<tr>
<td>14-16</td>
<td>D5-D7</td>
<td>Data 5,6,7</td>
<td>In/Output</td>
</tr>
<tr>
<td>17-19</td>
<td>A4-A6</td>
<td>Address 4,5,6</td>
<td>Input</td>
</tr>
<tr>
<td>20-21</td>
<td>NC</td>
<td>No Connection</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>A7</td>
<td>Address 7</td>
<td>Input</td>
</tr>
<tr>
<td>23-24</td>
<td>V_{PP}</td>
<td>Power Supply</td>
<td></td>
</tr>
<tr>
<td>25-27</td>
<td>NC</td>
<td>No Connection</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>/CE</td>
<td>Chip Select</td>
<td>Input</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EPROM Programming Mode Pin</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>/OE</td>
<td>Output Enable</td>
<td>Input</td>
</tr>
<tr>
<td>30</td>
<td>EPM</td>
<td>EPROM Prog. Mode</td>
<td>Input</td>
</tr>
<tr>
<td>31</td>
<td>V_{pp}</td>
<td>Prog. Voltage</td>
<td>Input</td>
</tr>
<tr>
<td>32</td>
<td>A8</td>
<td>Address 8</td>
<td>Input</td>
</tr>
<tr>
<td>33-35</td>
<td>NC</td>
<td>No Connection</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>A9</td>
<td>Address 9</td>
<td>Input</td>
</tr>
<tr>
<td>37</td>
<td>A11</td>
<td>Address 11</td>
<td>Input</td>
</tr>
<tr>
<td>38</td>
<td>A10</td>
<td>Address 10</td>
<td>Input</td>
</tr>
<tr>
<td>39</td>
<td>/PGM</td>
<td>Prog. Mode</td>
<td>Input</td>
</tr>
<tr>
<td>40-41</td>
<td>A0,A1</td>
<td>Address 0,1</td>
<td>Input</td>
</tr>
<tr>
<td>42-43</td>
<td>NC</td>
<td>No Connection</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>A2</td>
<td>Address 2</td>
<td>Input</td>
</tr>
</tbody>
</table>
Figure 8. 44-Pin QFP Pin Configuration (EPROM Programming Mode)

Table 6. 44-pin QFP Pin Identification

<table>
<thead>
<tr>
<th>EPROM Programming Mode Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2 A5-A6</td>
<td>Address 5,6</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>3-4 NC</td>
<td>No Connection</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>5 A7</td>
<td>Address 7</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>6-7 Vcc</td>
<td>Power Supply</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-10 NC</td>
<td>No Connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 /CE</td>
<td>Chip Select</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>12 /OE</td>
<td>Output Enable</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>13 EPM</td>
<td>EPROM Prog. Mode</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>14 Vpp</td>
<td>Prog. Voltage</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>15 A8</td>
<td>Address 8</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>16-18 NC</td>
<td>No Connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19 A9</td>
<td>Address 9</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>20 A11</td>
<td>Address 11</td>
<td>Input</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EPROM Programming Mode Pin #</th>
<th>Symbol</th>
<th>Function</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>21 A10</td>
<td>Address 10</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>22 /PGM</td>
<td>Prog. Mode</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>23-24 A0,A1</td>
<td>Address 0,1</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>25-26 NC</td>
<td>No Connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27 A2</td>
<td>Address 2</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>28-29 GND</td>
<td>Ground</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30-31 NC</td>
<td>No Connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 A3</td>
<td>Address 3</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>33-37 D0-D4</td>
<td>Data 0,1,2,3,4</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>38-40 NC</td>
<td>No Connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41-43 D5-D7</td>
<td>Data 5,6,7</td>
<td>In/Output</td>
<td></td>
</tr>
<tr>
<td>44 A4</td>
<td>Address 4</td>
<td>Input</td>
<td></td>
</tr>
</tbody>
</table>
PIN FUNCTIONS

EPROM Programming Mode

D7-D0 Data Bus. The data can be read from or written to external memory through the data bus.

A11-A0 Address Bus. During programming, the EPROM address is written to the address bus.

VCC Power Supply. This pin must supply 5V during the EPROM read mode and 6V during other modes.

/CE Chip Enable (active Low). This pin is active during EPROM Read Mode, Program Mode and Program Verify Mode.

/OE Output Enable (active Low). This pin drives the direction of the Data Bus. When this pin is Low, the Data Bus is output, when High, the Data Bus is input.

EPM EPROM Program Mode. This pin controls the different EPROM Program Mode by applying different voltages.

Vpp Program Voltage. This pin supplies the program voltage.

/PGM Program Mode (active Low). When this pin is Low, the data is programmed to the EPROM through the Data Bus.

Application Precaution

The production test-mode environment may be enabled accidentally during normal operation if excessive noise surges above VCC occur on pins XTAL1 and /RESET.

In addition, processor operation of Z8 OTP devices may be affected by excessive noise surges on the Vpp, /CE, /EPM, /OE pins while the microcontroller is in standard mode.

Recommendations for dampening voltage surges in both test and OTP mode include the following:

- Using a clamping diode to VCC;
- Adding a capacitor to the affected pin.

Z86E40 Standard Mode

XTAL Crystal 1 (time-based input). This pin connects a parallel-resonant crystal, ceramic resonator, LC, RC network, or external single-phase clock to the on-chip oscillator input.

XTAL2 Crystal 2 (time-based output). This pin connects a parallel-resonant crystal, ceramic resonator, LC, or RC network to the on-chip oscillator output.

R/W Read/Write (output, write Low). The R/W signal is Low when the CCP is writing to the external program or data memory.

/RESET Reset (input, active Low). Reset will initialize the MCU. Reset is accomplished either through Power-On, Watch-Dog Timer reset, STOP-Mode Recovery, or external reset. During Power-On Reset and Watch-Dog Timer Reset, the internally generated reset drives the reset pin low for the POR time. Any devices driving the reset line must be open-drain in order to avoid damage from a possible conflict during reset conditions. Pull-up is provided internally. After the POR time, /RESET is a Schmitt-triggered input.

/ROMless (input, active Low). This pin, when connected to GND, disables the internal ROM and forces the device to function as a Z86C90/C89 ROMless Z8. (Note that, when left unconnected or pulled High to VCC, the device functions normally as a Z8 ROM version).

Note: When using in ROM Mode in High EMI (noisy) environment, the ROMless pins should be connected directly to VCC.

To avoid asynchronous and noisy reset problems, the Z86E40 is equipped with a reset filter of four external clocks (4TPc). If the external reset signal is less than 4TPc in duration, no reset occurs. On the fifth clock after the reset is detected, an internal RST signal is latched and held for an internal register count of 18 external clocks, or for the duration of the external reset, whichever is longer. During the reset cycle, /DS is held active Low while /AS cycles at a rate of TPc/2. Program execution begins at location 000CH, 5-10 TPc cycles after /RESET is released. For Power-On Reset, the reset output time is 5 ms. The Z86E40 does not reset WDTMR, SMR, P2M, and P3M registers on a STOP-Mode Recovery operation.
Port 0 (P07-P00). Port 0 is an 8-bit, bi-directional, CMOS compatible I/O port. These eight I/O lines can be configured under software control as a nibble I/O port, or as an address port for interfacing external memory. The input buffers are Schmitt-triggered and nibble programmed. Either nibble output that can be globally programmed as push-pull or open-drain. Low EMI output buffers can be globally programmed by the software. Port 0 can be placed under handshake control. In Handshake mode, Port 3 lines P32 and P35 are used as handshake control lines. The handshake direction is determined by the configuration (input or output) assigned to Port 0's upper nibble. The lower nibble must have the same direction as the upper nibble.

For external memory references, Port 0 provides address bits A11-A8 (lower nibble) or A15-A8 (lower and upper nibble) depending on the required address space. If the address range requires 12 bits or less, the upper nibble of Port 0 can be programmed independently as I/O while the lower nibble is used for addressing. If one or both nibbles are needed for I/O operation, they must be configured by writing to the Port 0 mode register. In ROMless mode, after a hardware reset, Port 0 is configured as address lines A15-A8, and extended timing is set to accommodate slow memory access. The initialization routine can include reconfiguration to eliminate this extended timing mode. In ROM mode, Port 0 is defined as input after reset.

Port 0 can be set in the high-impedance mode if selected as an address output state, along with Port 1 and the control signals /AS, /DS, and R/W (Figure 9).
**PIN FUNCTIONS (Continued)**

**Port 1 (P17-P10).** Port 1 is an 8-bit, bi-directional, CMOS compatible port with multiplexed Address (A7-A0) and Data (D7-D0) ports. These eight I/O lines can be programmed as inputs or outputs or can be configured under software control as an Address/Data port for interfacing external memory. The input buffers are Schmitt-triggered and the output buffers can be globally programmed as either push-pull or open-drain. Low EMI output buffers can be globally programmed by the software. Port 1 can be placed under handshake control. In this configuration, Port 3, lines P33 and P34 are used as the handshake controls RDY1 and /DAV1 (Ready and Data Available). To interface external memory, Port 1 must be programmed for the multiplexed Address/Data mode. If more than 256 external locations are required, Port 0 outputs the additional lines (Figure 10).

Port 1 can be placed in the high-impedance state along with Port 0, /AS, /DS, and R/W, allowing the Z86E40 to share common resources in multiprocessor and DMA applications.

![Z86E40 CMOS Z8* OTP, 8-Bit CCP™](image)

**Figure 10. Port 1 Configuration**
Port 2 (P27-P20). Port 2 is an 8-bit, bi-directional, CMOS compatible I/O port. These eight I/O lines can be configured under software control as an input or output, independently. All input buffers are Schmitt-triggered. Bits programmed as outputs can be globally programmed as either push-pull or open-drain. Low EMI output buffers can be globally programmed by the software. When used as an I/O port, Port 2 can be placed under handshake control.

In Handshake Mode, Port 3 lines P31 and P36 are used as handshake control lines. The handshake direction is determined by the configuration (input or output) assigned to bit 7 of Port 2 (Figure 11).

![Port 2 Configuration Diagram](image)

**Figure 11. Port 2 Configuration**
PIN FUNCTIONS (Continued)

Port 3 (P37-P30). Port 3 is an 8-bit, CMOS compatible port with our fixed inputs (P33-P30) and four fixed outputs (P37-P34). These eight lines can be configured by software for interrupt and handshake control functions. Port 3, Pin 0 is Schmitt-triggered. P31, P32 and P33 are standard CMOS inputs (no Auto Latches) and P34, P35, P36 and P37 are push-pull output lines. Low EMI output buffers can be globally programmed by the software. Two on-board comparators can process analog signals on P31 and P32 with reference to the voltage on P33. The analog function is enabled by setting the D1 of Port 3 Mode Register (P3M). The comparator output can be outputted from P34 and P37, respectively, by setting PCON register Bit D0 to 1 state. For the interrupt function, P30 and P33 are falling edge triggered interrupt inputs. P31 and P32 can be programmed as falling, rising or both edges triggered interrupt inputs (Figure 12). Access to Counter/Timer 1 is made through P31 (T1N) and P36 (TOUT). Handshake lines for Port 0, Port 1, and Port 2 are also available on Port 3 (Table 7).

Note: P33-P30 differs from the Z86C40 in that there is no clamping diode to Vcc due to the EPROM high-voltage circuits. Exceeding the Vih maximum specification during standard operating mode may cause the device to enter EPROM mode.

Figure 12. Port 3 Configuration
Table 7. Port 3 Pin Assignments

<table>
<thead>
<tr>
<th>Pin</th>
<th>I/O</th>
<th>CTC1</th>
<th>Analog</th>
<th>Interrupt</th>
<th>P0 HS</th>
<th>P1 HS</th>
<th>P2 HS</th>
<th>Ext</th>
</tr>
</thead>
<tbody>
<tr>
<td>P30</td>
<td>IN</td>
<td></td>
<td></td>
<td>IRQ3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P31</td>
<td>IN</td>
<td>T_IN</td>
<td>AN1</td>
<td>IRQ2</td>
<td>D/R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P32</td>
<td>IN</td>
<td></td>
<td>AN2</td>
<td>IRQ0</td>
<td></td>
<td>D/R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P33</td>
<td>IN</td>
<td></td>
<td>REF</td>
<td>IRQ1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P34</td>
<td>OUT</td>
<td></td>
<td>AN1-Out</td>
<td>R/D</td>
<td></td>
<td>/DM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P35</td>
<td>OUT</td>
<td></td>
<td>R/D</td>
<td>R/D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P36</td>
<td>OUT</td>
<td>T_OUT</td>
<td>An2-Out</td>
<td>R/D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Comparator Inputs.** Port 3, P31 and P32, each have a comparator front end. The comparator reference voltage P33 is common to both comparators. In analog mode, P31 and P32 are the positive input of the comparators and P33 is the reference voltage of the comparators.

**Auto Latch.** The Auto Latch puts valid CMOS levels on all CMOS inputs (except P33-P31) that are not externally driven. Whether this level is 0 or 1, cannot be determined. A valid CMOS level, rather than a floating node, reduces excessive supply current flow in the input buffer. Auto Latches are available on Port 0, Port 2 and P30. There are no Auto Latches on P31, P32, and P33.

**Low EMI Emission.** The Z86E40 can be programmed to operate in a low EMI emission mode in the PCON register. The oscillator and all I/O ports can be programmed as low EMI emission mode independently. Use of this feature results in:

- The pre-drivers slew rate reduced to 10 ns typical.
- Low EMI output drivers have resistance of 200 Ohms (typical).
- Low EMI Oscillator.
- Internal SCLK/TCLK = XTAL operation limited to a maximum of 4 MHz - 250 ns cycle time, when Low EMI Oscillator is selected and system clock (SCLK = XTAL, SMR Reg. Bit D1 = 1)
FUNCTIONAL DESCRIPTION

The Z86E40 MCU incorporates the following special functions to enhance the standard Z8 architecture to provide the user with increased design flexibility.

RESET. The device is reset in one of three ways:

1. Power-On Reset
2. Watch-Dog Timer
3. STOP-Mode Recovery Source

Having the Auto Power-on Reset circuitry built-in, the Z86E40 does not need to be connected to an external power-on reset circuit. The reset time is 5 ms (typical) plus 18 clock cycles. The Z86E40 does not re-initialize WDTMR, SMR, P2M, and P3M registers to their reset values on a STOP-Mode Recovery operation. Note: The device \( V_{cc} \) must rise up to the operating \( V_{cc} \) specification before the \( T_{POR} \) expires.

Program Memory. The Z86E40 can address up to 4 Kbytes of internal program memory (Figure 13). The first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts. For EPROM mode, byte 12 (000CH) to address 4095 (0FFFH) consists of programmable EPROM. After reset, the program counter points at the address 000CH which is the starting address of the user program. In ROMless mode, the Z86E40 can address up to 64 Kbytes of external program memory. The ROM/ROMless option is only available on the 44-pin devices.

![Figure 13. Program Memory Map](image-url)

**Table:**

<table>
<thead>
<tr>
<th>EPROM</th>
<th>ROMless</th>
</tr>
</thead>
<tbody>
<tr>
<td>External ROM and RAM</td>
<td>External ROM and RAM</td>
</tr>
<tr>
<td>On-Chip One Time PROM</td>
<td></td>
</tr>
<tr>
<td>Location of First Byte of Instruction Executed After RESET</td>
<td>_irqs</td>
</tr>
<tr>
<td>12</td>
<td>irq5</td>
</tr>
<tr>
<td>11</td>
<td>irq5</td>
</tr>
<tr>
<td>10</td>
<td>irq4</td>
</tr>
<tr>
<td>9</td>
<td>irq4</td>
</tr>
<tr>
<td>8</td>
<td>irq3</td>
</tr>
<tr>
<td>7</td>
<td>irq3</td>
</tr>
<tr>
<td>6</td>
<td>irq2</td>
</tr>
<tr>
<td>5</td>
<td>irq2</td>
</tr>
<tr>
<td>4</td>
<td>irq1</td>
</tr>
<tr>
<td>3</td>
<td>irq1</td>
</tr>
<tr>
<td>2</td>
<td>irq0</td>
</tr>
<tr>
<td>1</td>
<td>irq0</td>
</tr>
<tr>
<td>0</td>
<td>irq0</td>
</tr>
</tbody>
</table>
EPROM Protect. The 4 Kbytes of program memory is a one-time PROM. An EPROM protect feature prevents dumping of the ROM contents by inhibiting execution of LDC, LDCI, LDE, and LDEI instructions to program memory in all modes. ROM look-up tables cannot be used with this feature.

Data Memory (/DM). In EPROM mode, the Z86E40 can address up to 60 Kbytes of external data memory beginning at location 4096. In ROMless mode, the Z86E40 can address up to 64 Kbytes of data memory. External data memory may be included with, or separated from, the external program memory space. /DM, an optional I/O function that can be programmed to appear on pin P34, is used to distinguish between data and program memory space (Figure 14). The state of the /DM signal is controlled by the type of instruction being executed. An LDC opcode references PROGRAM (/DM inactive) memory, and an LDE instruction references data (/DM active Low) memory.

Expanded Register File (ERF). The register file has been expanded to allow for additional system control registers, mapping of additional peripheral devices and input/output ports into the register address area. The Z8 register address space R0 through R15 is implemented as 16 groups of 16 registers per group (Figure 15). These register groups are known as the Expanded Register File (ERF).

The low nibble (D3-D0) of the Register Pointer (RP) select the active ERF group, and the high nibble (D7-D4) of register RP select the working register group (Figure 16). Three system configuration registers reside in the Expanded Register File at bank FH: PCON, SMR, and WDTMR. The rest of the Expanded Register is not physically implemented and is reserved for future expansion.
Figure 15. Expanded Register File Architecture

Notes:
U = Unknown
† For Z86E40 (ROMless) reset condition: "10110110"
* Will not be reset with a STOP Mode Recovery
** Will not be reset with a STOP Mode Recovery, except Bit D0.
Register File. The 256 byte register file consists of four I/O port registers, 236 general-purpose registers and 15 control and status registers (R240 is reserved). The instructions can access registers directly or indirectly through an 8-bit address field. This allows short, 4-bit register addresses using the Register Pointer (Figures 16 and 17). In the 4-bit mode, the register file is divided into 16 working register groups, each occupying 16 continuous locations. The Register Pointer addresses the starting location of the active working-register group.

Note: Register Bank E0H-EFH can only be accessed through working registers and indirect addressing modes.

Figure 16. Register Pointer Register

Figure 17. Register Pointer
FUNCTIONAL DESCRIPTION (Continued)

**General-Purpose Registers (GPR).** These registers are undefined after the device is powered up. The registers keep their last value after any reset, as long as the reset occurs in the $V_{cc}$ voltage-specified operating range.

**RAM Protect.** The upper portion of the RAM's address spaces 80H to EFH (excluding the control registers) can be protected from reading and writing. This option can be selected during the EPROM Programming Mode. After this option is selected, the user can activate this feature from the internal EPROM. D6 of the IMR control register (R251) is used to turn off/on the RAM protect by loading a 0 or 1, respectively. A 1 in D6 indicates RAM Protect enabled.

**Stack.** The Z86E40 external data memory or the internal register file can be used for the stack. The 16-bit Stack Pointer (R254-R255) is used for the external stack which can reside anywhere in the data memory for ROMless mode, but only from 4096 to 65535 in ROM mode. An 8-bit Stack Pointer (R255) is used for the internal stack that resides within the 236 general-purpose registers (R4-R239). SPH can be used as a general-purpose register when using internal stack only.

**Counter/Timers.** There are two 8-bit programmable counter/timers (T0 and T1), each driven by its own 6-bit programmable prescaler. The T1 prescaler is driven by internal or external clock sources; however, the T0 prescaler is driven by the internal clock only (Figure 18).

The 6-bit prescalers can divide the input frequency of the clock source by any integer number from 1 to 64. Each prescaler drives its counter, which decrements the value (1 to 256) that has been loaded into the counter. When the counter reaches the end of count, a timer interrupt request, IRQ4 (T0) or IRQ5 (T1), is generated.

The counters can be programmed to start, stop, restart to continue, or restart from the initial value. The counters can also be programmed to stop upon reaching zero (single pass mode) or to automatically reload the initial value and continue counting (modulo-n continuous mode).

The counters, but not the prescalers, can be read at any time without disturbing their value or count mode. The clock source for T1 is user-definable and can be either the internal microprocessor clock divided by four, or an external signal input through Port 3. The Timer Mode register configures the external timer input (P31) as an external clock, a trigger input that can be retriggerable or non-retriggerable, or as a gate input for the internal clock. Port 3 line P36 serves as a timer output ($T_{out}$) through which T0, T1 or the internal clock can be output. The counter/timers can be cascaded by connecting the T0 output to the input of T1.
Figure 18. Counter/Timer Block Diagram
FUNCTIONAL DESCRIPTION (Continued)

Interrupts. The Z86E40 has six different interrupts from six different sources. The interrupts are maskable and prioritized (Figure 19). The six sources are divided as follows: four sources are claimed by Port 3 lines P33-P30 and two in counter/timers. The Interrupt Mask Register globally or individually enables or disables the six interrupt requests (Table 8).

---

**Figure 19. Interrupt Block Diagram**

**Table 8. Interrupt Types, Sources, and Vectors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Vector Location</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRQ0</td>
<td>/DAV0, IRQ0</td>
<td>0, 1</td>
<td>External (P32), Rising/Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ1</td>
<td>IRQ1</td>
<td>2, 3</td>
<td>External (P33), Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ2</td>
<td>/DAV2, IRQ2, $T_{IN}$</td>
<td>4, 5</td>
<td>External (P31), Rising/Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ3</td>
<td>IRQ3</td>
<td>6, 7</td>
<td>External (P30), Falling Edge Triggered</td>
</tr>
<tr>
<td>IRQ4</td>
<td>$T_0$</td>
<td>8, 9</td>
<td>Internal</td>
</tr>
<tr>
<td>IRQ5</td>
<td>$T_1$</td>
<td>10, 11</td>
<td>Internal</td>
</tr>
</tbody>
</table>
When more than one interrupt is pending, priorities are resolved by a programmable priority encoder that is controlled by the Interrupt Priority Register (IPR). An interrupt machine cycle is activated when an interrupt request is granted. Thus, disabling all subsequent interrupts, saves the Program Counter and Status Flags, and then branches to the program memory vector location reserved for that interrupt. All Z86E40 interrupts are vectored through locations in the program memory. This memory location and the next byte contain the 16-bit starting address of the interrupt service routine for that particular interrupt request.

To accommodate polled interrupt systems, interrupt inputs are masked and the interrupt request register is polled to determine which of the interrupt requests need service.

An interrupt resulting from AN1 is mapped into IRQ2, and an interrupt from AN2 is mapped into IRQ0. Interrupts IRQ2 and IRQ0 may be rising, falling or both edge triggered, and are programmable by the user. The software may poll to identify the state of the pin.

Programming bits for the Interrupt Edge Select are located in bits D7 and D6 of the IRQ Register (R250). The configuration is shown in Table 9.

<table>
<thead>
<tr>
<th>IRQ</th>
<th>D7</th>
<th>D6</th>
<th>P31</th>
<th>P32</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>F</td>
<td>R</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>R</td>
<td>F</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>R/F</td>
<td>R/F</td>
</tr>
</tbody>
</table>

Notes:
F = Falling Edge
R = Rising Edge

Clock. The Z86E40 on-chip oscillator has a high-gain, parallel-resonant amplifier for connection to a crystal, RC, ceramic resonator, or any suitable external clock source (XTAL1 = Input, XTAL2 = Output). The crystal should be AT cut, 10 kHz to 12 MHz max, with a series resistance (RS) less than or equal to 100 Ohms.

The crystal should be connected across XTAL1 and XTAL2 using the vendor's recommended capacitor values from each pin directly to device pin Ground. The RC oscillator option can be selected in the programming mode. The RC oscillator configuration must be an external resistor connected from XTAL1 to XTAL2, with a frequency-setting capacitor from XTAL1 to Ground (Figure 20).

Note: RC OSC does not support 12 MHz.
**FUNCTIONAL DESCRIPTION (Continued)**

**Power-On Reset (POR).** A timer circuit clocked by a dedicated on-board RC oscillator is used for the Power-On Reset (POR) timer function. The POR timer allows $V_{cc}$ and the oscillator circuit to stabilize before instruction execution begins.

The POR timer circuit is a one-shot timer triggered by one of three conditions:

1. Power fail to Power OK status
2. STOP-Mode Recovery (if D5 of SMR=0)
3. WDT time-out

The POR time is a nominal 5 ms. Bit 5 of the STOP mode Register (SMR) determines whether the POR timer is bypassed after STOP-Mode Recovery (typical for an external clock and RC/LC oscillators with fast start up times).

**HALT.** Turns off the internal CPU clock, but not the XTAL oscillation. The counter/timers and external interrupt IRQ0, IRQ1, and IRQ2 remain active. The device is recovered by interrupts, either externally or internally generated. An interrupt request must be executed (enabled) to exit HALT mode. After the interrupt service routine, the program continues from the instruction after the HALT.

In order to enter STOP or HALT mode, it is necessary to first flush the instruction pipeline to avoid suspending execution in mid-instruction. To do this, the user must execute a NOP (opcode=FFH) immediately before the appropriate sleep instruction, i.e.:

```
FF NOP ; clear the pipeline
6F STOP ; enter STOP mode
```  

or

```
FF NOP ; clear the pipeline
7F HALT ; enter HALT mode
```  

**STOP.** This instruction turns off the internal clock and external crystal oscillation and reduces the standby current to 10 microamperes or less. STOP mode is terminated by one of the following resets: either by WDT time-out, POR, a STOP-Mode Recovery Source which is defined by the SMR register or external reset. This causes the processor to restart the application program at address 0000H.

**Port Configuration Register (PCON).** The PCON register configures the ports individually; comparator output on Port 3, open-drain on Port 0 and Port 1, low EMI on Ports 0, 1, 2 and 3, and low EMI oscillator. The PCON register is located in the expanded register file at Bank F, location 00 (Figure 21).

![Figure 21. Port Configuration Register (PCON) (Write Only)](image-url)
Comparator Output Port 3 (DO). Bit 0 controls the comparator use in Port 3. A 1 in this location brings the comparator outputs to P34 and P37, and a 0 releases the Port to its standard I/O configuration.

Port 1 Open-Drain (D1). Port 1 can be configured as an open-drain by resetting this bit (D1=0) or configured as push-pull active by setting this bit (D1=1). The default value is 1.

Port 0 Open-Drain (D2). Port 0 can be configured as an open-drain by resetting this bit (D2=0) or configured as push-pull active by setting this bit (D2=1). The default value is 1.

Low EMI Port 0 (D3). Port 0 can be configured as a Low EMI Port by resetting this bit (D3=0) or configured as a Standard Port by setting this bit (D3=1). The default value is 1.

Low EMI Port 1 (D4). Port 1 can be configured as a Low EMI Port by resetting this bit (D4=0) or configured as a Standard Port by setting this bit (D4=1). The default value is 1.

Low EMI Port 2 (D5). Port 2 can be configured as a Low EMI Port by resetting this bit (D5=0) or configured as a Standard Port by setting this bit (D5=1). The default value is 1.

Low EMI Port 3 (D6). Port 3 can be configured as a Low EMI Port by resetting this bit (D6=0) or configured as a Standard Port by setting this bit (D6=1). The default value is 1.

Low EMI OSC (D7). This bit of the PCON Register controls the low EMI noise oscillator. A 1 in this location configures the oscillator with standard drive. While a 0 configures the oscillator with low noise drive, however, it does not affect the relationship of SCLK and XTAL. The low EMI mode will reduce the drive of the oscillator (OSC). The default value is 1. **Note:** 4 MHz is the maximum external clock frequency when running in the low EMI oscillator mode.

STOP-Mode Recovery Register (SMR). This register selects the clock divide value and determines the mode of STOP-Mode Recovery (Figure 22). All bits are Write Only except bit 7 which is a Read Only. Bit 7 is a flag bit that is hardware set on the condition of STOP Recovery and reset by a power-on cycle. Bit 6 controls whether a low or high level is required from the recovery source. Bit 5 controls the reset delay after recovery. Bits 2, 3, and 4 of the SMR register specify the STOP-Mode Recovery Source. The SMR is located in Bank F of the Expanded Register Group at address OBH.

**Figure 22. STOP-Mode Recovery Register**
(Write-Only Except Bit D7, Which Is Read-Only)
SCLK/TCLK Divide-by-16 Select (D0). This bit of the SMR controls a divide-by-16 prescaler of SCLK/TCLK. The purpose of this control is to selectively reduce device power consumption during normal processor execution (SCLK control) and/or HALT mode (where TCLK sources counter/timers and interrupt logic).

External Clock Divide-by-Two (D1). This bit can eliminate the oscillator divide-by-two circuitry. When this bit is 0, the System Clock (SCLK) and Timer Clock (TCLK) are equal to the external clock frequency divided by two. The SCLK/TCLK is equal to the external clock frequency when this bit is set (D1 = 1). Using this bit together with D7 of PCON further helps lower EMI (i.e., D7 (PCON) = 0, D1 (SMR) = 1). The default setting is zero.

STOP-Mode Recovery Source (D2, D3, and D4). These three bits of the SMR register specify the wake up source of the STOP-Mode Recovery (Figure 23). Table 10 shows the SMR source selected with the setting of D2 to D4. P33-P31 cannot be used to wake up from STOP mode when programmed as analog inputs.

Figure 23. STOP-Mode Recovery Source

<p>| Table 10. STOP-Mode Recovery Source |</p>
<table>
<thead>
<tr>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>SMR Source selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>POR recovery only</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>P30 transition</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>P31 transition (Not in analog mode)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>P32 transition (Not in analog mode)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>P33 transition (Not in analog mode)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>P27 transition</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Logical NOR of Port 2 bits 0-3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Logical NOR of Port 2 bits 0-7</td>
</tr>
</tbody>
</table>

STOP-Mode Recovery Delay Select (D5). The 5 ms RESET delay after STOP-Mode Recovery is disabled by programming this bit to a zero. A 1 in this bit will cause a 5 ms RESET delay after STOP-Mode Recovery. The default condition of this bit is 1. If the fast wake up mode is selected, the STOP-Mode Recovery source needs to be kept active for at least 5T磐C.

STOP-Mode Recovery Level Select (D6). A 1 in this bit defines that a high level on any one of the recovery sources wakes the Z86E40 from STOP mode. A 0 defines low level recovery. The default value is 0.
Cold or Warm Start (D7). This bit is set by the device upon entering STOP mode. A 0 in this bit indicates that the device has been reset by POR (cold). A 1 in this bit indicates the device was awakened by a SMR source (warm).

Watch-Dog Timer Mode Register (WDTMR). The WDT is a retriggerable one-shot timer that resets the Z8 if it reaches its terminal count. The WDT is disabled after Power-On Reset and initially enabled by executing the WDT instruction and refreshed on subsequent executions of the WDT instruction. The WDT cannot be disabled when it has been enabled. The WDT is driven either by an on-board RC oscillator or an external oscillator from XTAL1 pin. The POR clock source is selected with bit 4 of the WDT register.

Note: Execution of the WDT instruction affects the Z (Zero), S (Sign), and V (Overflow) flags.

WDT Time-out Period (D0 and D1). Bits 0 and 1 control a tap circuit that determines the time-out periods that can be obtained (Table 11). The default value of D0 and D1 are 1 and 0, respectively.

Table 11. Time-out Period of the WDT

<table>
<thead>
<tr>
<th>D1</th>
<th>D0</th>
<th>Time-out of the Internal RC OSC</th>
<th>Time-out of the External Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>5 ms</td>
<td>256 TpC</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>15 ms*</td>
<td>512 TpC*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>25 ms</td>
<td>1024 TpC</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>100 ms</td>
<td>4096 TpC</td>
</tr>
</tbody>
</table>

Notes:
TpC = External clock cycle.
* The default setting is 15 ms.

WDT During HALT Mode (D2). This bit determines whether or not the WDT is active during HALT mode. A 1 indicates that the WDT is active during HALT. A 0 disables the WDT in HALT mode. The default value is 1.

WDT During STOP Mode (D3). This bit determines whether or not the WDT is active during STOP mode. A 1 indicates active during STOP. A 0 disables the WDT during STOP mode.

Clock Source For WDT (D4). This bit determines which oscillator source is used to clock the internal POR and WDT counter chain. If the bit is a 1, the internal RC oscillator is bypassed and the POR and WDT clock source is driven from the external pin, XTAL1. The default configuration of this bit is 0, which selects the RC oscillator.

WDTMR Register Accessibility. The WDTMR register is accessible only during the first 64 internal system clock cycles from the execution of the first instruction after Power-On Reset, Watch-Dog reset or a STOP-Mode Recovery (Figures 24 and 25). After this point, the register cannot be modified by any means, intentional or otherwise. The WDTMR cannot be read and is located in Bank F of the Expanded Register Group at address location OFH.

Figure 24. Watch-Dog Timer Mode Register (Write Only)
Figure 25. Resets and WDT
Auto Reset Voltage. An on-board Voltage Comparator checks that $V_{cc}$ is at the required level to ensure correct operation of the device. Reset is globally driven if $V_{cc}$ is below $V_{rst}$ (Figure 26). If the $V_{cc}$ drops below 4.5V while the device is operating, the device must be powered down, then powered up again.

Note: $V_{cc}$ must be in the allowed operating range (4.5V to 5.5V) prior to the minimum Power-On Reset time-out ($T_{POR}$).

Figure 26. Typical Z86E40 $V_{rst}$ Voltage vs Temperature
FUNCTIONAL DESCRIPTION (Continued)

EPROM Programming Mode

Table 12 shows the programming voltages of each programming mode. Table 13, Figures 27, 28, and 29 show the programming timing of each programming mode. Figure 30 shows the circuit diagram of a Z86E40 programming adaptor, which adapts from 2764A to Z86E40. Figure 31 shows the flow-chart of an Intelligent Programming Algorithm, which is compatible with 2764A EPROM (Z86E40 is 4K EPROM, 2764A is 8K EPROM). Since the EPROM size of Z86E40 differs from 2764A, the programming address range has to be set from 0000H to 0FFFH. Otherwise, the upper 4K of data (1000H-1FFFH) will overwrite the lower 4K of data.

Note: EPROM Protect feature disables the LDC, LDCI, LDE, and LDEI instructions, and a ROM look-up table cannot be used with this feature.

Table 12. EPROM Programming Table

<table>
<thead>
<tr>
<th>Programming Modes</th>
<th>Vpp</th>
<th>EPM</th>
<th>/CE</th>
<th>/OE</th>
<th>/PGM</th>
<th>ADDR</th>
<th>DATA</th>
<th>Vcc *</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPROM READ1</td>
<td>X</td>
<td>VH</td>
<td>VIL</td>
<td>VIL</td>
<td>VH</td>
<td>ADDR</td>
<td>Out</td>
<td>4.5V†</td>
</tr>
<tr>
<td>EPROM READ2</td>
<td>VH</td>
<td>X</td>
<td>VIL</td>
<td>VIL</td>
<td>VH</td>
<td>ADDR</td>
<td>Out</td>
<td>5.5V†</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>VH</td>
<td>X</td>
<td>VIL</td>
<td>VIL</td>
<td>VIL</td>
<td>ADDR</td>
<td>In</td>
<td>6.0V</td>
</tr>
<tr>
<td>PROGRAM VERIFY</td>
<td>VH</td>
<td>X</td>
<td>VIL</td>
<td>VIL</td>
<td>VIL</td>
<td>ADDR</td>
<td>Out</td>
<td>6.0V</td>
</tr>
<tr>
<td>EPROM PROTECT</td>
<td>VH</td>
<td>VH</td>
<td>VIL</td>
<td>VIL</td>
<td>VIL</td>
<td>NU</td>
<td>NU</td>
<td>6.0V</td>
</tr>
<tr>
<td>RC OSCILLATOR SELECT</td>
<td>VH</td>
<td>VH</td>
<td>VIL</td>
<td>VIL</td>
<td>VIL</td>
<td>NU</td>
<td>NU</td>
<td>6.0V</td>
</tr>
<tr>
<td>RAM PROTECT</td>
<td>VH</td>
<td>VH</td>
<td>VIL</td>
<td>VIL</td>
<td>VIL</td>
<td>NU</td>
<td>NU</td>
<td>6.0V</td>
</tr>
</tbody>
</table>

Notes:  
VH = 12.5V ± 12.5V  
VIL = As per specific Z8 DC specification.  
X = Not used, but must be set to VH, VIL, or VIL level.  
NU = Not used, but must be set to either VH or VIL level.

Notes (Continued):  
lpp during programming = 40 mA maximum.  
lcc during programming, verify, or read = 40 mA maximum.  
* Vcc has a tolerance of ±0.25V.  
† Although most programmers do an EPROM read at Vcc = 5.0, Zilog recommends an EPROM read at Vcc = 4.5 and 5.0 to ensure proper device operations during the Vcc after programming.

Table 13. EPROM Programming Timing

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Name</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Address Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>2</td>
<td>Data Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>3</td>
<td>Vpp Setup</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>4</td>
<td>Vcc Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>5</td>
<td>Chip Enable Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>6</td>
<td>Program Pulse Width</td>
<td>0.95</td>
<td>1.05</td>
<td>ms</td>
</tr>
<tr>
<td>7</td>
<td>Data Hold Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>8</td>
<td>/OE Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>9</td>
<td>Data Access Time</td>
<td>2</td>
<td>200</td>
<td>ns</td>
</tr>
<tr>
<td>10</td>
<td>Data Output Float Time</td>
<td></td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td>11</td>
<td>Overprogram Pulse Width</td>
<td>2.85</td>
<td></td>
<td>ms</td>
</tr>
<tr>
<td>12</td>
<td>EPM Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>13</td>
<td>/PGM Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>14</td>
<td>Address to /OE Setup Time</td>
<td>2</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>15</td>
<td>Option Program Pulse Width</td>
<td>78</td>
<td></td>
<td>ms</td>
</tr>
</tbody>
</table>
Figure 27. EPROM READ Mode Timing Diagram
Figure 28. Timing Diagram of EPROM Program and Verify Modes
Figure 29. Timing Diagram of EPROM Protect, RAM Protect and RC OSC Modes
Figure 30. Z86E40 Z8 OTP Programming Adapter
Figure 31. Z86E40 Programming Algorithm

Note:
* To ensure proper operation, Zilog recommends Vcc range of the device Vcc specification.
ABSOLUTE MAXIMUM RATINGS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>V_{cc}</td>
<td>Supply Voltage (*)</td>
<td>-0.3</td>
<td>+7.0</td>
<td>V</td>
</tr>
<tr>
<td>V_{PM}</td>
<td>Max. Input Voltage (**)</td>
<td>7</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>T_{STG}</td>
<td>Storage Temp</td>
<td>-65°</td>
<td>+150°</td>
<td>C</td>
</tr>
<tr>
<td>T_{A}</td>
<td>Oper Ambient Temp</td>
<td>†</td>
<td>†</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Power Dissipation</td>
<td>2.2</td>
<td>W</td>
<td></td>
</tr>
</tbody>
</table>

Notes:
* Voltage on all pins with respect to GND.
** Applies to all Port pins only, except Port 31, 32, 33 and must limit current going into or out of port pin to 250 µA max.
† See Ordering Information.

STANDARD TEST CONDITIONS

The characteristics listed below apply for standard test conditions as noted. All voltages are referenced to Ground. Positive current flows into the referenced pin (Figure 32).

CAPACITANCE

\[ T_a = 25°C; \ V_{cc} = \text{GND} = 0V; \ f = 1.0 \text{ MHz}; \text{ unmeasured pins returned to GND}. \]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input capacitance</td>
<td>0</td>
<td>12 pF</td>
</tr>
<tr>
<td>Output capacitance</td>
<td>0</td>
<td>12 pF</td>
</tr>
<tr>
<td>I/O capacitance</td>
<td>0</td>
<td>12 pF</td>
</tr>
</tbody>
</table>

V_{cc} SPECIFICATION

\[ V_{cc} = 5.0V \pm 0.5V \]
## DC ELECTRICAL CHARACTERISTICS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>( V_{cc} )</th>
<th>( T_{a} = 0°C ) to +70°C</th>
<th>Typical ( @ 25°C )</th>
<th>Units</th>
<th>Conditions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max Input Voltage</td>
<td>5.0V</td>
<td>( V_{cc} + 0.5V )</td>
<td>2.5</td>
<td>V</td>
<td>( I_n ) &lt;250 ( \mu )A</td>
<td>[7]</td>
</tr>
<tr>
<td>( V_{CH} )</td>
<td>Clock Input High Voltage</td>
<td>5.0V</td>
<td>0.7 ( V_{cc} )</td>
<td>0.2 ( V_{cc} )</td>
<td>1.5</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
</tr>
<tr>
<td>( V_{CL} )</td>
<td>Clock Input Low Voltage</td>
<td>5.0V</td>
<td>( V_{ss} ) -0.3</td>
<td>0.2 ( V_{cc} )</td>
<td>1.5</td>
<td>V</td>
<td>Driven by External Clock Generator</td>
</tr>
<tr>
<td>( V_{IH} )</td>
<td>Input High Voltage</td>
<td>5.0V</td>
<td>0.7 ( V_{cc} )</td>
<td>( V_{cc} +0.3 )</td>
<td>2.5</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>( V_{IL} )</td>
<td>Input Low Voltage</td>
<td>5.0V</td>
<td>( V_{ss} ) -0.3</td>
<td>0.2 ( V_{cc} )</td>
<td>1.5</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>( V_{OH} )</td>
<td>Output High Voltage (Low EMI Mode)</td>
<td>5.0V</td>
<td>( V_{cc} ) -0.4</td>
<td>0.4</td>
<td>0.1</td>
<td>V</td>
<td>( I_{OH} ) =-2.0 mA</td>
</tr>
<tr>
<td>( V_{OL} )</td>
<td>Output Low Voltage (Low EMI Mode)</td>
<td>5.0V</td>
<td>( V_{ss} ) -0.3</td>
<td>0.2 ( V_{cc} )</td>
<td>1.5</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>( V_{RST} )</td>
<td>Output Low Voltage</td>
<td>5.0V</td>
<td>1.5</td>
<td>0.3</td>
<td>V</td>
<td>( I_{OL} ) =+12 mA, 3 Pin Max</td>
<td></td>
</tr>
<tr>
<td>( V_{RH} )</td>
<td>Reset Input High Voltage</td>
<td>5.0V</td>
<td>0.7 ( V_{cc} )</td>
<td>( V_{cc} +0.3 )</td>
<td>2.1</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>( V_{RL} )</td>
<td>Reset Input Low Voltage</td>
<td>5.0V</td>
<td>( V_{ss} ) -0.3</td>
<td>0.2 ( V_{cc} )</td>
<td>1.7</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>( V_{OFFSET} )</td>
<td>Comparator Input Offset Voltage</td>
<td>5.0V</td>
<td>50</td>
<td>10</td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( I_{IL} )</td>
<td>Input Leakage</td>
<td>5.0V</td>
<td>-10</td>
<td>+10</td>
<td>&lt;1</td>
<td>( \mu )A</td>
<td>( V_{in} ) = OV, ( V_{cc} )</td>
</tr>
<tr>
<td>( I_{OL} )</td>
<td>Output Leakage</td>
<td>5.0V</td>
<td>-10</td>
<td>+10</td>
<td>&lt;1</td>
<td>( \mu )A</td>
<td>( V_{in} ) = OV, ( V_{IL} ) = 0</td>
</tr>
<tr>
<td>( I_{R} )</td>
<td>Reset Input Current</td>
<td>5.0V</td>
<td>60</td>
<td>45</td>
<td>( \mu )A</td>
<td>( V_{cc} ) = 5.0V,</td>
<td></td>
</tr>
<tr>
<td>( I_{CC} )</td>
<td>Supply Current (Standard Mode)</td>
<td>5.0V</td>
<td>16</td>
<td>15.0</td>
<td>mA</td>
<td>( \phi 8 ) MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>18</td>
<td>mA</td>
<td>( \phi 12 ) MHz</td>
<td>[4,5]</td>
<td></td>
</tr>
</tbody>
</table>
## DC ELECTRICAL CHARACTERISTICS (Continued)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$</th>
<th>$T_{a} = 0^\circ C$ to $+70^\circ C$</th>
<th>Typical @ 25$^\circ C$</th>
<th>Conditions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{CC1}$</td>
<td>Standby Current (Standard Mode)</td>
<td>5.0V</td>
<td>6.0</td>
<td>3.5 mA</td>
<td>HALT mode $V_{IN} = OV, V_{CC}$</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>7.5</td>
<td>4.5 mA</td>
<td>HALT mode $V_{IN} = OV, V_{CC}$</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>3.0</td>
<td>1.5 mA</td>
<td>Clock Divide by 16</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>3.0</td>
<td>1.7 mA</td>
<td>Clock Divide by 16</td>
<td>[4,5]</td>
</tr>
<tr>
<td>$I_{CC}$</td>
<td>Supply Current (Low EMI Mode)</td>
<td>5.0V</td>
<td>7.5</td>
<td>5.0 mA</td>
<td>@ 2 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>12.0</td>
<td>8.0 mA</td>
<td>@ 4 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td>$I_{CC1}$</td>
<td>Standby Current (Low EMI Mode)</td>
<td>5.0V</td>
<td>2.0</td>
<td>1.0 mA</td>
<td>@ 2 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>3.0</td>
<td>1.5 mA</td>
<td>@ 2 MHz</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>2.0</td>
<td>0.75 mA</td>
<td>Clock Divide-by-16</td>
<td>[4,5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>2.0</td>
<td>1.0 mA</td>
<td>Clock Divide-by-16</td>
<td>[4,5]</td>
</tr>
<tr>
<td>$I_{CC2}$</td>
<td>Standby Current</td>
<td>5.0V</td>
<td>10</td>
<td>2 µA</td>
<td>STOP mode $V_{IN} = OV, V_{CC}$</td>
<td>[6]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.0V</td>
<td>800</td>
<td>450 µA</td>
<td>STOP mode $V_{IN} = OV, V_{CC}$</td>
<td>[6,10]</td>
</tr>
<tr>
<td>$I_{ALL}$</td>
<td>Auto Latch Low Current</td>
<td>5.0V</td>
<td>-10</td>
<td>-5 µA</td>
<td>$0V &lt; V_{IN} &lt; V_{CC}$</td>
<td></td>
</tr>
<tr>
<td>$I_{ALH}$</td>
<td>Auto Latch High Current</td>
<td>5.0V</td>
<td>20</td>
<td>10 µA</td>
<td>$0V &lt; V_{IN} &lt; V_{CC}$</td>
<td></td>
</tr>
<tr>
<td>$T_{POR}$</td>
<td>Power-On Reset</td>
<td>5.0V</td>
<td>2.5</td>
<td>4.5 ms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{RST}$</td>
<td>Auto Reset Voltage</td>
<td>2.6 V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**

1. $I_{CC1}$
   - Typ: Clock-driven crystal
   - Max: 3.5 mA
   - Unit: 8 MHz
   - Freq: 0.3 mA
2. $V_{SS}=OV=GND$.
3. $V_{CC}$ must be in the allowed operating range (4.5V to 5.5V) prior to the minimum $T_{POR}$ timeout. $V_{CC}$ specified at 4.5V to 5.5V.
4. All outputs unloaded, I/O pins floating, inputs at rail.
5. $CL_1 = CL_2 = 100 pF$.
7. Except clock pins and Port 3 input pins in EPROM mode.
8. Port Low EMI mode.
9. Port STD mode.
10. Internal RC.
AC ELECTRICAL CHARACTERISTICS
External I/O or Memory Read/Write Timing Diagrams (Standard Mode)

Figure 33. External I/O or Memory Read/Write Timing
### AC ELECTRICAL CHARACTERISTICS

External I/O or Memory Read/Write Timing (Standard Mode)

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>$V_{cc}$</th>
<th>8 MHz</th>
<th>12 MHz</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$T_{dA(AS)}$</td>
<td>Address Valid to /AS Rise Delay</td>
<td>$5.0V$</td>
<td>35</td>
<td>35</td>
<td>ns</td>
<td>[2]</td>
</tr>
<tr>
<td>2</td>
<td>$T_{dAS(A)}$</td>
<td>/AS Rise to Address Float Delay</td>
<td>$5.0V$</td>
<td>70</td>
<td>45</td>
<td>ns</td>
<td>[2]</td>
</tr>
<tr>
<td>3</td>
<td>$T_{dAS(DR)}$</td>
<td>/AS Rise to Read Data Req'd Valid</td>
<td>$5.0V$</td>
<td>400</td>
<td>250</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td>4</td>
<td>$T_{wAS}$</td>
<td>/AS Low Width</td>
<td>$5.0V$</td>
<td>80</td>
<td>55</td>
<td>ns</td>
<td>[2]</td>
</tr>
<tr>
<td>5</td>
<td>$T_{dAS(DS)}$</td>
<td>Address Float to /DS Fall</td>
<td>$5.0V$</td>
<td>300</td>
<td>200</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td>6</td>
<td>$T_{wDSR}$</td>
<td>/DS (Read) Low Width</td>
<td>$5.0V$</td>
<td>200</td>
<td>100</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td>7</td>
<td>$T_{wDSW}$</td>
<td>/DS (Write) Low Width</td>
<td>$5.0V$</td>
<td>165</td>
<td>110</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td>8</td>
<td>$T_{dDSR(DR)}$</td>
<td>/DS Fall to Read Data Req'd Valid</td>
<td>$5.0V$</td>
<td>260</td>
<td>160</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td>9</td>
<td>$T_{hDR(DS)}$</td>
<td>Read Data /DS Rise Hold Time</td>
<td>$5.0V$</td>
<td>95</td>
<td>55</td>
<td>ns</td>
<td>[2]</td>
</tr>
<tr>
<td>10</td>
<td>$T_{dDS(A)}$</td>
<td>/DS Rise to Address Active Delay</td>
<td>$5.0V$</td>
<td>70</td>
<td>45</td>
<td>ns</td>
<td>[2]</td>
</tr>
<tr>
<td>11</td>
<td>$T_{dDS(AS)}$</td>
<td>/DS Rise to /AS Fall Delay</td>
<td>$5.0V$</td>
<td>70</td>
<td>45</td>
<td>ns</td>
<td>[2]</td>
</tr>
<tr>
<td>12</td>
<td>$T_{dR/W(AS)}$</td>
<td>R/W Valid to /AS Rise Delay</td>
<td>$5.0V$</td>
<td>70</td>
<td>45</td>
<td>ns</td>
<td>[2]</td>
</tr>
<tr>
<td>13</td>
<td>$T_{dDS(R/W)}$</td>
<td>/DS Rise to R/W Not Valid</td>
<td>$5.0V$</td>
<td>70</td>
<td>45</td>
<td>ns</td>
<td>[2]</td>
</tr>
<tr>
<td>14</td>
<td>$T_{dD(W/AS)}$</td>
<td>Write Data Valid to /DS Fall (Write) Delay</td>
<td>$5.0V$</td>
<td>80</td>
<td>55</td>
<td>ns</td>
<td>[2]</td>
</tr>
<tr>
<td>15</td>
<td>$T_{dDS(DW)}$</td>
<td>/DS Rise to Write Data Not Valid</td>
<td>$5.0V$</td>
<td>80</td>
<td>55</td>
<td>ns</td>
<td>[2]</td>
</tr>
<tr>
<td>16</td>
<td>$T_{dA(DR)}$</td>
<td>Address Valid to Read Data Req'd Valid</td>
<td>$5.0V$</td>
<td>475</td>
<td>310</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td>17</td>
<td>$T_{dAS(DS)}$</td>
<td>/AS Rise to /DS Fall Delay</td>
<td>$5.0V$</td>
<td>100</td>
<td>65</td>
<td>ns</td>
<td>[2]</td>
</tr>
<tr>
<td>18</td>
<td>$T_{dDI(DS)}$</td>
<td>Data Output Setup to /DS Rise</td>
<td>$5.0V$</td>
<td>75</td>
<td>75</td>
<td>ns</td>
<td>[1,2]</td>
</tr>
<tr>
<td>19</td>
<td>$T_{dDM(AS)}$</td>
<td>/DM Valid to /AS Fall Delay</td>
<td>$5.0V$</td>
<td>55</td>
<td>35</td>
<td>ns</td>
<td>[2]</td>
</tr>
</tbody>
</table>

**Notes:**

1. When using extended memory timing add 2$T_{pC}$.
2. Timing numbers given are for minimum $T_{pC}$.
3. $5.0V ±0.5V$ Standard Test Load. All timing references use $0.7V_{cc}$ for a logic 1 and $0.2V_{cc}$ for a logic 0. Standard operating temperature range 0°C to +70°C.
AC ELECTRICAL CHARACTERISTICS
Additional Timing Diagrams (Standard Mode)

Figure 34. Additional Timing
### AC ELECTRICAL CHARACTERISTICS

#### Additional Timing Table (Standard Mode)

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>( V_{cc} ) Min</th>
<th>8 MHz Min</th>
<th>8 MHz Max</th>
<th>12 MHz Min</th>
<th>12 MHz Max</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( T_{PCI} )</td>
<td>Input Clock Period</td>
<td>5.0V</td>
<td>125 DC</td>
<td>83 DC</td>
<td>ns</td>
<td>[1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>( T_{RICI} )</td>
<td>Clock Input Rise &amp; Fall Times</td>
<td>5.0V</td>
<td>25</td>
<td>15</td>
<td>ns</td>
<td>[1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>( T_{WC} )</td>
<td>Input Clock Width</td>
<td>5.0V</td>
<td>62.5 DC</td>
<td>41.5 DC</td>
<td>ns</td>
<td>[1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>( T_{W Tin L} )</td>
<td>Timer Input Low Width</td>
<td>5.0V</td>
<td>70</td>
<td>70</td>
<td>ns</td>
<td>[1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>( T_{W Tin H} )</td>
<td>Timer Input High Width</td>
<td>5.0V</td>
<td>5( T_{PCI} )</td>
<td>5( T_{PCI} )</td>
<td></td>
<td>[1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>( T_{P Tin} )</td>
<td>Timer Input Period</td>
<td>5.0V</td>
<td>8( T_{PCI} )</td>
<td>8( T_{PCI} )</td>
<td></td>
<td>[1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>( T_{R Tin,TITin} )</td>
<td>Timer Input Rise &amp; Fall Times</td>
<td>5.0V</td>
<td>100</td>
<td>100</td>
<td>ns</td>
<td>[1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8A</td>
<td>( T_{W L} )</td>
<td>Int. Request Low Time</td>
<td>5.0V</td>
<td>70</td>
<td>70</td>
<td>ns</td>
<td>[1,2]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8B</td>
<td>( T_{W H} )</td>
<td>Int. Request Low Time</td>
<td>5.0V</td>
<td>5( T_{PCI} )</td>
<td>5( T_{PCI} )</td>
<td></td>
<td>[1,3]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>( T_{Wsh} )</td>
<td>Int. Request High Time</td>
<td>5.0V</td>
<td>5( T_{PCI} )</td>
<td>5( T_{PCI} )</td>
<td></td>
<td>[1,2]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>( T_{wsm} )</td>
<td>STOP-Mode Recovery Width Spec</td>
<td>5.0V</td>
<td>12</td>
<td>12</td>
<td>ns</td>
<td>[11]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>( T_{o q} )</td>
<td>Oscillator Startup Time</td>
<td>5.0V</td>
<td>5( T_{PCI} )</td>
<td>5( T_{PCI} )</td>
<td></td>
<td>[4]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>( T_{w d t} )</td>
<td>Watch-Dog Timer Delay Time</td>
<td>5.0V</td>
<td>5</td>
<td>5</td>
<td>ms</td>
<td>[6]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Notes:

1. Timing Reference uses 0.7 \( V_{cc} \) for a logic 1 and 0.2 \( V_{cc} \) for a logic 0.
2. Interrupt request through Port 3 (P33-P31).
3. Interrupt request through Port 3 (P30).
4. SMR-D5 = 0
5. 5.0V ±0.5V
6. Reg. WDTMR D1=0, D0=0
7. Reg. WDTMR D1=0, D0=1
8. Reg. WDTMR D1=1, D0=0
9. Reg. WDTMR D1=1, D0=1
10. Reg. SMR-D5=0. No Delay.
AC ELECTRICAL CHARACTERISTICS

Handshake Timing Diagrams

Data In

/DataV (Input)

RDY (Output)

Figure 34. Input Handshake Timing

Data Out

/DataV (Output)

RDY (Input)

Figure 35. Output Handshake Timing
## AC ELECTRICAL CHARACTERISTICS
Handshake Timing Table - (Standard Modes)

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Parameter</th>
<th>V&lt;sub&gt;cc&lt;/sub&gt; Note[1]</th>
<th>8 MHz Min</th>
<th>8 MHz Max</th>
<th>12 MHz Min</th>
<th>12 MHz Max</th>
<th>Data Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TsDI(DAV)</td>
<td>Data In Setup Time</td>
<td>5.0V</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>IN</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ThDI(DAV)</td>
<td>Data In Hold Time</td>
<td>5.0V</td>
<td>115</td>
<td>115</td>
<td>115</td>
<td>IN</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TwDAV</td>
<td>Data Available Width</td>
<td>5.0V</td>
<td>110</td>
<td>110</td>
<td>110</td>
<td>IN</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TdDAV(RDY)</td>
<td>DAV Fall to RDY Fall Delay</td>
<td>5.0V</td>
<td>115</td>
<td>115</td>
<td>115</td>
<td>IN</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>TdDAVid(RDY)</td>
<td>DAV Rise to RDY Rise Delay</td>
<td>5.0V</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>IN</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>TdRDYO(DAV)</td>
<td>RDY Rise to DAV Fall Delay</td>
<td>5.0V</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>IN</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TdDO(DAV)</td>
<td>Data Out to DAV Fall Delay</td>
<td>5.0V</td>
<td>63</td>
<td>42</td>
<td>42</td>
<td>OUT</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>TdDAVO(RDY)</td>
<td>DAV Fall to RDY Fall Delay</td>
<td>5.0V</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OUT</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>TdRDYO(DAV)</td>
<td>RDY Fall to DAV Rise Delay</td>
<td>5.0V</td>
<td>115</td>
<td>115</td>
<td>115</td>
<td>OUT</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>TwRDY</td>
<td>RDY Width</td>
<td>5.0V</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>OUT</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>TdRDYOd(DAV)</td>
<td>RDY Rise to DAV Fall Delay</td>
<td>5.0V</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>OUT</td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**

[1] 5.0 V ±0.5V

Standard operating temperature range 0°C to +70°C
EXPANDED REGISTER FILE CONTROL REGISTERS

**PCON (FH) 00H**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparator Output Port 3  0: P34, P37 Standard*  1: P34, P37 Comparator Output</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port 1 Open Drain  0  1: Port 1 Push-pull Active*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port 0 Open Drain  0  1: Port 0 Push-pull Active*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port 0 Low EMI  0  1: Port 0 Standard*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port 1 Low EMI  0  1: Port 1 Standard*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port 2 Low EMI  0  1: Port 2 Standard*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port 3 Low EMI  0  1: Port 3 Standard*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low EMI Oscillator  0  1: Low EMI Standard*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Default Setting After Reset

**SMR (FH) 0B**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCLK/TCLK Divide-by-16  0: OFF**  1: ON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>External Clock Divide by 2  0: SCLK/TCLK = XTAL/2*  1: SCLK/TCLK = XTAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stop Mode Recovery Source  000: POR Only and/or External Reset*  001: P30  010: P31  011: P32  100: P33  101: P27  110: P2 NOR 0-3  111: P2 NOR 0-7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stop Delay  0: OFF  1: ON*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stop Recovery Level  0: Low*  1: High</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stop Flag  0: POR*  1: Stop Recovery</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Default Setting After RESET.
** Default Setting After RESET and STOP-Mode Recovery.

**Figure 36. Port Configuration Register (Write Only)**

**Figure 37. STOP-Mode Recovery Register (Write Only Except Bit D7, Which Is Read Only)**

**WDTMR (FH) 0F**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>WDT Tap  INT RC OSC  External Clock  00: 5 ms  256 TPC  01: 15 ms  512 TPC  10: 25 ms  1024 TPC  11: 100 ms  4096 TPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WDT During HALT  0: OFF  1: ON*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WDT During STOP  0: OFF  1: ON*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XTAL/INT RC Select for WDT  0: On-Board RC*  1: XTAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Default setting after RESET

**Figure 38. Watch-Dog Timer Mode Register (Write Only)**

10-45
Z8 CONTROL REGISTER DIAGRAMS

**Figure 39. Reserved**

- **R240**
  - **D7 D6 D5 D4 D3 D2 D1 D0**
  - Reserved (Must be 0)

**Figure 40. Timer Mode Register (F1H: Read/Write)**

- **R241 TMR**
  - **D7 D6 D5 D4 D3 D2 D1 D0**
  - 0: No Function
  - 1: Load T0
  - 0: Disable T0 Count
  - 1: Enable T0 Count
  - 0: No Function
  - 1: Load T1
  - 0: Disable T1 Count
  - 1: Enable T1 Count

  **TIN Modes**
  - 00: External Clock Input
  - 01: Gate Input
  - 10: Trigger Input (Non-retriggerable)
  - 11: Trigger Input (Retriggerable)

  **TOUT Modes**
  - 00: Not Used
  - 01: T0 Out
  - 10: T1 Out
  - 11: Internal Clock Out

  Default After Reset = 00H

**Figure 41. Counter/Timer 1 Register (F2H: Read/Write)**

- **R242 T1**
  - **D7 D6 D5 D4 D3 D2 D1 D0**
  - **T1 Initial Value**
    - (When Written)
    - (Range: 1-256 Decimal 01-00 HEX)
  - **T1 Current Value**
    - (When Read)

**Figure 42. Prescaler 1 Register (F3H: Write Only)**

- **R243 PRE1**
  - **D7 D6 D5 D4 D3 D2 D1 D0**
  - Count Mode
    - 0: T1 Single Pass
    - 1: T1 Modulo N
  - Clock Source
    - 1: T1 Internal
    - 0: T1 External Timing Input (TIN Mode)
  - Prescaler Modulo
    - (Range: 1-64 Decimal 01-00 HEX)

*Default After Reset

**Figure 43. Counter/Timer 0 Register (F4H: Read/Write)**

- **R244 T0**
  - **D7 D6 D5 D4 D3 D2 D1 D0**
  - **T0 Initial Value**
    - (When Written)
    - (Range: 1-256 Decimal 01-00 HEX)
  - **T0 Current Value**
    - (When Read)

**Figure 44. Prescaler 0 Register (F5H: Write Only)**

- **R245 PRE0**
  - **D7 D6 D5 D4 D3 D2 D1 D0**
  - Count Mode
    - 0: T1 Single Pass
    - 1: T1 Modulo N
  - Reserved (Must be 0)
  - Prescaler Modulo
    - (Range: 1-64 Decimal 01-00 HEX)
Figure 45. Port 2 Mode Register (F6H; Write Only)

Figure 46. Port 3 Mode Register (F7H; Write Only)

Figure 47. Port 0 and 1 Mode Register (F8H; Write Only)

Figure 48. Interrupt Priority Register (F9H; Write Only)
Z8 CONTROL REGISTER DIAGRAMS (Continued)

**Figure 49. Interrupt Request Register (FA<sub>H</sub>: Read/Write)**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRQ0 = P32 Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRQ1 = P33 Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRQ2 = P31 Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRQ3 = P30 Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRQ4 = T0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRQ5 = T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Inter Edge

P31 ↓ P32 ↓ = 00
P31 ↓ P32 ↑ = 01
P31 ↑ P32 ↓ = 10
P31 ↑ P32 ↑ = 11

Default After Reset = 00H

**Figure 50. Interrupt Mask Register (FB<sub>H</sub>: Read/Write)**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enables IRQ0-IRQ5 (D0 = IRQ0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enables RAM Protect †</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enables Interrupts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

† RAM Protect option must be previously selected.

**Figure 51. Flag Register (FC<sub>H</sub>: Read/Write)**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Flag F1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Flag F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Half Carry Flag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decimal Adjust Flag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overflow Flag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sign Flag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zero Flag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Carry Flag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 52. Register Pointer (FD<sub>H</sub>: Read/Write)**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expanded Register File</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Working Register Pointer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Default After Reset = 00H

**Figure 53. Stack Pointer High (FE<sub>H</sub>: Read/Write)**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stack Pointer Upper Byte (SP8 - SP15)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 54. Stack Pointer Low (FF<sub>H</sub>: Read/Write)**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stack Pointer Lower Byte (SP0 - SP7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
INSTRUCTION SET NOTATION

Addressing Modes. The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRR</td>
<td>Indirect register pair or indirect working-register pair address</td>
</tr>
<tr>
<td>Irr</td>
<td>Indirect working-register pair only</td>
</tr>
<tr>
<td>X</td>
<td>Indexed address</td>
</tr>
<tr>
<td>DA</td>
<td>Direct address</td>
</tr>
<tr>
<td>RA</td>
<td>Relative address</td>
</tr>
<tr>
<td>IM</td>
<td>Immediate</td>
</tr>
<tr>
<td>R</td>
<td>Register or working-register address</td>
</tr>
<tr>
<td>r</td>
<td>Working-register address only</td>
</tr>
<tr>
<td>IR</td>
<td>Indirect-register or indirect working-register address</td>
</tr>
<tr>
<td>Ir</td>
<td>Indirect working-register address only</td>
</tr>
<tr>
<td>RR</td>
<td>Register pair or working register pair address</td>
</tr>
</tbody>
</table>

Symbols. The following symbols are used in describing the instruction set.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst</td>
<td>Destination location or contents</td>
</tr>
<tr>
<td>src</td>
<td>Source location or contents</td>
</tr>
<tr>
<td>cc</td>
<td>Condition code</td>
</tr>
<tr>
<td>®</td>
<td>Indirect address prefix</td>
</tr>
<tr>
<td>SP</td>
<td>Stack Pointer</td>
</tr>
<tr>
<td>PC</td>
<td>Program Counter</td>
</tr>
<tr>
<td>FLAGS</td>
<td>Flag register (Control Register 252)</td>
</tr>
<tr>
<td>RP</td>
<td>Register Pointer (R253)</td>
</tr>
<tr>
<td>IMR</td>
<td>Interrupt mask register (R251)</td>
</tr>
</tbody>
</table>

Flags. Control register (R252) contains the following six flags:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Carry flag</td>
</tr>
<tr>
<td>Z</td>
<td>Zero flag</td>
</tr>
<tr>
<td>S</td>
<td>Sign flag</td>
</tr>
<tr>
<td>V</td>
<td>Overflow flag</td>
</tr>
<tr>
<td>D</td>
<td>Decimal-adjust flag</td>
</tr>
<tr>
<td>H</td>
<td>Half-carry flag</td>
</tr>
</tbody>
</table>

Affected flags are indicated by:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Clear to zero</td>
</tr>
<tr>
<td>1</td>
<td>Set to one</td>
</tr>
<tr>
<td>*</td>
<td>Set to clear according to operation</td>
</tr>
<tr>
<td>–</td>
<td>Unaffected</td>
</tr>
<tr>
<td>x</td>
<td>Undefined</td>
</tr>
</tbody>
</table>
## CONDITION CODES

<table>
<thead>
<tr>
<th>Value</th>
<th>Mnemonic</th>
<th>Meaning</th>
<th>Flags Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>C</td>
<td>Always True</td>
<td>—</td>
</tr>
<tr>
<td>0111</td>
<td>NC</td>
<td>No Carry</td>
<td>C = 1</td>
</tr>
<tr>
<td>1111</td>
<td>Z</td>
<td>Zero</td>
<td>Z = 1</td>
</tr>
<tr>
<td>0110</td>
<td>NZ</td>
<td>Not Zero</td>
<td>Z = 0</td>
</tr>
<tr>
<td>1101</td>
<td>PL</td>
<td>Plus</td>
<td>S = 0</td>
</tr>
<tr>
<td>0101</td>
<td>MI</td>
<td>Minus</td>
<td>S = 1</td>
</tr>
<tr>
<td>0100</td>
<td>OV</td>
<td>Overflow</td>
<td>V = 1</td>
</tr>
<tr>
<td>1100</td>
<td>NOV</td>
<td>No Overflow</td>
<td>V = 0</td>
</tr>
<tr>
<td>0110</td>
<td>EQ</td>
<td>Equal</td>
<td>Z = 1</td>
</tr>
<tr>
<td>1110</td>
<td>NE</td>
<td>Not Equal</td>
<td>Z = 0</td>
</tr>
<tr>
<td>1000</td>
<td>GE</td>
<td>Greater Than or Equal</td>
<td>(S XOR V) = 0</td>
</tr>
<tr>
<td>0001</td>
<td>LT</td>
<td>Less than</td>
<td>(S XOR V) = 1</td>
</tr>
<tr>
<td>1010</td>
<td>GT</td>
<td>Greater Than</td>
<td>[Z OR (S XOR V)] = 0</td>
</tr>
<tr>
<td>0010</td>
<td>LE</td>
<td>Less Than or Equal</td>
<td>[Z OR (S XOR V)] = 1</td>
</tr>
<tr>
<td>1111</td>
<td>UGE</td>
<td>Unsigned Greater Than or Equal</td>
<td>C = 0</td>
</tr>
<tr>
<td>0111</td>
<td>ULT</td>
<td>Unsigned Less Than</td>
<td>C = 1</td>
</tr>
<tr>
<td>1011</td>
<td>UGT</td>
<td>Unsigned Greater Than</td>
<td>(C = 0 AND Z = 0) = 1</td>
</tr>
<tr>
<td>0011</td>
<td>ULE</td>
<td>Unsigned Less Than or Equal</td>
<td>(C OR Z) = 1</td>
</tr>
<tr>
<td>0000</td>
<td>F</td>
<td>Never True (Always False)</td>
<td>—</td>
</tr>
</tbody>
</table>
### INSTRUCTION FORMATS

#### One-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>dest</th>
<th>src</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOP</td>
<td></td>
<td>1110</td>
<td>src</td>
<td></td>
</tr>
<tr>
<td>CCF</td>
<td>DI, EI, IRET, NOP, RCF, RET, SCF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Two-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC, ADD, AND, CP, LD, OR, SBC, SUB, TCM, TM, XOR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADC, ADD, AND, CP, LD, OR, SBC, SUB, TCM, TM, XOR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Three-Byte Instructions

<table>
<thead>
<tr>
<th>OPC</th>
<th>MODE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>JP, CALL (Indirect)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### INSTRUCTION SUMMARY

**Note:** Assignment of a value is indicated by the symbol "\( \rightarrow \). For example:

\[
dst \leftarrow dst + src
\]

indicates that the source data is added to the destination data and the result is stored in the destination location. The notation "addr (n)" is used to refer to bit (n) of a given operand location. For example:

\[
dst (7)
\]

refers to bit 7 of the destination operand.
## INSTRUCTION SUMMARY (Continued)

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
<th>Instruction and Operation</th>
<th>Address Mode</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC dst, src</td>
<td>†</td>
<td>1[ ]</td>
<td>* * * 0</td>
<td>INC dst</td>
<td>r</td>
<td>rE</td>
<td>* * *</td>
</tr>
<tr>
<td>dst←dst + src + C</td>
<td></td>
<td></td>
<td></td>
<td>dst←dst + 1</td>
<td>r</td>
<td>r = 0 – F</td>
<td></td>
</tr>
<tr>
<td>ADD dst, src</td>
<td>†</td>
<td>0[ ]</td>
<td>* * * 0</td>
<td>INCW dst</td>
<td>RR</td>
<td>AO</td>
<td>* * *</td>
</tr>
<tr>
<td>dst←dst + src</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IR</td>
<td>A1</td>
<td></td>
</tr>
<tr>
<td>AND dst, src</td>
<td>†</td>
<td>5[ ]</td>
<td>* * 0</td>
<td>IRET</td>
<td>BF</td>
<td>* * * * * *</td>
<td></td>
</tr>
<tr>
<td>dst←dst AND src</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALL dst</td>
<td>DA</td>
<td>D6</td>
<td></td>
<td>JP cc, dst</td>
<td>DA</td>
<td>cD</td>
<td></td>
</tr>
<tr>
<td>SP←SP – 2</td>
<td>IRR</td>
<td>D4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@SP←PC,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←dst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCF</td>
<td>EF</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C←NOT C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLR dst</td>
<td>R</td>
<td>B0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst←0</td>
<td>IR</td>
<td>B1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COM dst</td>
<td>R</td>
<td>60</td>
<td>* * 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst←NOT dst</td>
<td>IR</td>
<td>61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP dst, src</td>
<td>†</td>
<td>A[ ]</td>
<td>* * * –</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst←src</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DA dst</td>
<td>R</td>
<td>40</td>
<td>* * X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst←DA dst</td>
<td>IR</td>
<td>41</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEC dst</td>
<td>R</td>
<td>00</td>
<td>* * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst←dst – 1</td>
<td>IR</td>
<td>01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECW dst</td>
<td>RR</td>
<td>80</td>
<td>* * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst←dst – 1</td>
<td>IR</td>
<td>81</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DI</td>
<td>8F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMR(7)←0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DJNZr, dst</td>
<td>RA</td>
<td>rA</td>
<td>–</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r←r – 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>if r ≠ 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC←PC + dst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range: +127, –128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EI</td>
<td>9F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMR(7)←1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HALT</td>
<td>7F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTES:**
- Addresses are in little-endian format, with 0 being the least significant byte.
- Operands are in little-endian format, with 0 being the least significant byte.
- Flags affected are indicated by the following:
  - C: Carry
  - Z: Zero
  - S: Sign
  - V: Overflow
  - D: Half-Carry
  - H: Auxiliary Carry
<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode (dst src)</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected C Z S V D H</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR dst, src</td>
<td>t</td>
<td>4[ ]</td>
<td>* * 0 - - - -</td>
</tr>
<tr>
<td>dst ← dst OR src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POP dst</td>
<td>R</td>
<td>50</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>dst ← @SP; SP ← SP + 1</td>
<td>IR</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>PUSH src</td>
<td>R</td>
<td>70</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>SP ← SP - 1; @SP ← src</td>
<td>IR</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>RCF</td>
<td>CF</td>
<td>0</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>RET</td>
<td>AF</td>
<td>- - - - - -</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>70</td>
<td>- - - - - -</td>
<td></td>
</tr>
<tr>
<td>RR dst</td>
<td>R</td>
<td>E0</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>RRC dst</td>
<td>R</td>
<td>C0</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>SBC dst, src</td>
<td>t</td>
<td>3[ ]</td>
<td>* * 1 *</td>
</tr>
<tr>
<td>dst ← dst - src - C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCF</td>
<td>DF</td>
<td>1</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>SRA dst</td>
<td>R</td>
<td>D0</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>RP ← src</td>
<td>IR</td>
<td>D1</td>
<td></td>
</tr>
<tr>
<td>SRP dst</td>
<td>Im</td>
<td>31</td>
<td>- - - - - -</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction and Operation</th>
<th>Address Mode (dst src)</th>
<th>Opcode Byte (Hex)</th>
<th>Flags Affected C Z S V D H</th>
</tr>
</thead>
<tbody>
<tr>
<td>STOP</td>
<td>6F</td>
<td>1</td>
<td>- - - - - -</td>
</tr>
<tr>
<td>SUB dst, src</td>
<td>t</td>
<td>2[ ]</td>
<td>* * * 1 *</td>
</tr>
<tr>
<td>dst ← dst - src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWAP dst</td>
<td>R</td>
<td>F0</td>
<td>X * * X - -</td>
</tr>
<tr>
<td>SP ← SP - 1; @SP ← src</td>
<td>IR</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>TCM dst, src</td>
<td>t</td>
<td>6[ ]</td>
<td>* * 0 - -</td>
</tr>
<tr>
<td>(NOT dst) AND src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TM dst, src</td>
<td>t</td>
<td>7[ ]</td>
<td>- * 0 - -</td>
</tr>
<tr>
<td>dst AND src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XOR dst, src</td>
<td>t</td>
<td>B[ ]</td>
<td>- * 0 - -</td>
</tr>
<tr>
<td>dst ← dst XOR src</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WDT</td>
<td>5F</td>
<td>- X X X - -</td>
<td></td>
</tr>
</tbody>
</table>

† These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a '[' ]' in this table, and its value is found in the following table to the left of the applicable addressing mode pair.

For example, the opcode of an ADC instruction using the addressing modes r (destination) and Ir (source) is 13.

<table>
<thead>
<tr>
<th>Address Mode (dst src)</th>
<th>Lower Opcode Nibble</th>
</tr>
</thead>
<tbody>
<tr>
<td>r r</td>
<td>[2]</td>
</tr>
<tr>
<td>r Ir</td>
<td>[3]</td>
</tr>
<tr>
<td>R R</td>
<td>[4]</td>
</tr>
<tr>
<td>R IR</td>
<td>[5]</td>
</tr>
<tr>
<td>R IM</td>
<td>[6]</td>
</tr>
<tr>
<td>IR IM</td>
<td>[7]</td>
</tr>
</tbody>
</table>
## OPCODE MAP

### Lower Nibble (Hex)

<table>
<thead>
<tr>
<th>Upper Nibble (Hex)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6.5 DEC R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 DEC R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>1</td>
<td>6.5 RLC R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 DEC R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>2</td>
<td>6.5 INC R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 SUB R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>3</td>
<td>8.0 JP R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 SUB R1, R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>4</td>
<td>8.5 DA R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 OR R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>5</td>
<td>10.5 POP R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 AND R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>6</td>
<td>6.5 COM R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 REG R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>7</td>
<td>10.5 PUSH R2</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 TM R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>8</td>
<td>10.5 DECW R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 TM R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>9</td>
<td>6.5 RL R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 TM R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>A</td>
<td>10.5 INCW R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 TM R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>B</td>
<td>6.5 CLR R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 TM R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>C</td>
<td>6.5 RRC R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 TM R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>D</td>
<td>6.5 SRA R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 TM R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>E</td>
<td>6.5 RR R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 TM R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
<tr>
<td>F</td>
<td>6.5 SWAP R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 TM R1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6.5 ADD R1, r2</td>
<td>10.5 ADD R2, R1</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
<td>10.5 ADD R1, R1, IM</td>
</tr>
</tbody>
</table>

### Pipeline Cycles

- **Upper Opcode Nibble**: 4 cycles
- **Execution Cycles**: 3 cycles
- **Lower Opcode Nibble**: 2 cycles
- **First Operand**: 1 cycle
- **Second Operand**: 1 cycle

### Legend:

- **R** = 8-bit Address
- **r** = 4-bit Address
- **R1 or r1** = Dst Address
- **R2 or r2** = Src Address

### Sequence:

- Opcode, First Operand, Second Operand

### Note:

- Blanks are reserved.

*2-byte instruction appears as a 3-byte instruction
### 40-Pin DIP Package Diagram

**CONTRollING DIMENSIONS : INCH**

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MILLIMETER</th>
<th>INCH</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.51</td>
<td>.81</td>
<td>.020</td>
<td>.032</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>3.18</td>
<td>3.94</td>
<td>.125</td>
<td>.155</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>0.36</td>
<td>0.53</td>
<td>.015</td>
<td>.021</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BI</td>
<td>1.02</td>
<td>1.52</td>
<td>.040</td>
<td>.060</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.23</td>
<td>0.38</td>
<td>.009</td>
<td>.015</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>52.07</td>
<td>52.58</td>
<td>.205</td>
<td>.207</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>15.24</td>
<td>15.75</td>
<td>.600</td>
<td>.820</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>13.59</td>
<td>14.22</td>
<td>.535</td>
<td>.580</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ea</td>
<td>15.49</td>
<td>16.51</td>
<td>.610</td>
<td>.850</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>3.18</td>
<td>3.81</td>
<td>.125</td>
<td>.150</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q1</td>
<td>1.92</td>
<td>1.91</td>
<td>.060</td>
<td>.075</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>1.52</td>
<td>2.29</td>
<td>.060</td>
<td>.090</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTED:**
1. CONTROLLING DIMENSIONS : INCH
2. LEADS ARE COPLANAR WITHIN .004 IN.
3. DIMENSION : MILLIMETER

### 44-Pin PLCC Package Diagram

**DIM FROM CENTER TO CENTER OF RADIUS**

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MILLIMETER</th>
<th>INCH</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4.27</td>
<td>4.37</td>
<td>.169</td>
<td>.180</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>2.67</td>
<td>2.92</td>
<td>.105</td>
<td>.115</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D/E</td>
<td>17.40</td>
<td>17.65</td>
<td>.685</td>
<td>.695</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1/E1</td>
<td>16.51</td>
<td>16.66</td>
<td>.650</td>
<td>.656</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>15.24</td>
<td>16.00</td>
<td>.600</td>
<td>.630</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>1.27 TYP</td>
<td>.050 TYP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTED:**
1. CONTROLLING DIMENSIONS : INCH
2. LEADS ARE COPLANAR WITHIN .004 IN.
3. DIMENSION : MILLIMETER
ORDERING INFORMATION

Z86E40 (12 MHz)

<table>
<thead>
<tr>
<th>Package Type</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>40-Pin DIP</td>
<td>Z86E4012PSC</td>
</tr>
<tr>
<td>40-Pin PLCC</td>
<td>Z86E4012VSC</td>
</tr>
<tr>
<td>44-Pin QFP</td>
<td>Z86E4012FSC</td>
</tr>
<tr>
<td>40-Pin Cerdip Window Lid</td>
<td>Z86E4012KSE</td>
</tr>
</tbody>
</table>

For fast results, contact your local Zilog sales office for assistance in ordering the part desired.

CODES

Preferred Package
- P = Plastic DIP
- V = Plastic Chip Carrier

Longer Lead Time
- F = Plastic Quad Flat Pack
- K = Cerdip Window Lid

Temperature
- S = 0°C to +70°C

Speed
- 12 = 12 MHz

Environmental
- C = Plastic Standard
- E = Hermetic Standard

Example:

Z 86E40 12 P S C

is a Z86E40, 16 MHz, DIP, 0°C to +70°C, Plastic Standard Flow

Environmental Flow
Temperature
Package
Speed
Product Number
Zilog Prefix
Timekeeping with the Z8®

K

eep track of the current time with the Z8®, using minimal hardware. This Application Note describes several software-based methods of timekeeping, with complete code listings.

INTRODUCTION

Many controller applications require ongoing tracking of the current time — at a specific time, for instance, the controller may have to take a specific action, like turning on a solenoid. This is accomplished by using a Real Time Clock, or RTC, the sole purpose of which is to keep track of time. A microprocessor-based system may use a RTC integrated circuit for timekeeping, or accomplish the same results through software, using the microprocessor’s internal clock as a timebase. There are trade-offs associated with each approach, but the software method is by far the most economical, provided the micro can handle the added software overhead.

This Application Note covers two methods of timekeeping in software, with display of the current time on seven-segment LEDS. Time set switches are provided to advance the time. One method uses the microprocessor crystal as a timebase, while the other uses the highly-accurate 60 Hz line frequency as the timebase.

TIMEKEEPING BASED ON THE 8 MHZ CRYSTAL

![Diagram of Z86C21 circuit](image-url)
TIMEKEEPING BASED ON THE 60 HZ LINE FREQUENCY

[Diagram showing components and circuit connections, including Z86C08, 4.00 MHz XTAL, IN4001 diodes, 27 pF capacitors, and a 100 mF capacitor.]

III V SYSTEMS TSM5034T
This program configures the Zilog Z8 to keep time using the 12.00 MHz crystal frequency as a timebase. The time is displayed on four (4) seven-segment, common-anode LEDs. The cathodes of the displays are driven by P2. The anodes of the displays are connected to P0, and a pull-up resistor. The colon, made up of two discrete LEDs, flashes at a 1 Hz rate, and is driven by P2-7. The set switches are connected to P3. This program may be adapted to any 28-pin or 40-pin Z8. Written by Don Owen Newquist, Zilog, on 5-1-91.

WORK_REG .equ 10h
address_hi .equ r0
address_lo .equ r1
address .equ r0
pointer .equ r2
data .equ r3
col_count .equ r4
ring_counter .equ r5
HALF_SECOND .equ 16h

TIME_REG .equ 00h
millisec .equ r5
seconds .equ r6
minutes .equ r7
hours .equ r8
seconds_lo .equ r9
seconds_hi .equ r10
minutes_lo .equ r11
minutes_hi .equ r12
hours_lo .equ r13
hours_hi .equ r14
sw_count .equ r15
STATUS .equ $04
MILLISEC .equ $05
SECONDS .equ $06
MINUTES .equ $07
HOURS .equ $08
SECONDS_LO .equ $09
SECONDS_HI .equ $0a
MINUTES_LO .equ $0b
MINUTES_HI .equ $0c
HOURS_LO .equ $0d
HOURS_HI .equ $0e

.org 00
.word 00
.word 00
.word time
.word load_time
.org 000ch
**Initialization**

;******************************************************************************;
; Initialization
;******************************************************************************;
init: srp #WORK_REG
      di
      ld t0,#250 ; disable int
      ld pre0,#01010001b ; 250 decimal
      ld pre1,#%13
      ld t1,#00 ; set to for 5 mS period
      ld p2m,#0
      ld p3m,#0 ; outputs on p2
      ld p0lm,#04 ; open drain on p2
      clr p0
      clr p3
      clr status ; p0 low
      clr half_second
      clr status reg
      clr ipr,#00001000b ; clear status reg
      ld imr,#%30 ; make irq5 > irq3
      ld spl,#%80 ; enable irq4,irq5
      clr sp
      ld tmr,#%0f ; load and enable counters
      ld pointer,#04 ; point to time regs
      ld r15,#12 ; six locations
      clr @pointer ; clear ram
      inc pointer
      djnz r15,clear_reg ; continue until all zero
      ld hours,#%12 ; start time at 12:00
      ld pointer,#hours_hi ; start at hours reg
      ld ring_counter,#%88 ; enable first digit
      ei ; enable interrupts
      jr main_loop

;******************************************************************************;
; This routine converts the seconds, minutes, and hours bcd
; data into units and tens-of-units for displaying
;******************************************************************************;
time_convert: ld SECONDS_LO,SECONDS ; transfer contents
      ld SECONDS_HI,SECONDS
      and SECONDS_LO,#%0f ; keep only lower bits
      swap SECONDS_HI ; swap nibbles
      and SECONDS_HI,#%0f ; keep only lower bits
      ld MINUTES_LO,MINUTES ; transfer contents
      ld MINUTES_HI,MINUTES
      and MINUTES_LO,#%0f ; keep only lower bits
      swap MINUTES_HI ; swap nibbles
      and MINUTES_HI,#%0f ; keep only lower bits
      ld hours_lo,hours ; transfer contents
      ld hours_hi,hours
      and hours_lo,#%0f ; keep only lower bits
      swap hours_hi ; return
      and hours_hi,#%0f ; keep only lower bits
;**********************************************************************
; This interrupt routine updates the time
;**********************************************************************

; save current reg pointer
push rp

; point to time reg group
srp #TIME_REG

; look at time-set switches
call test_sw

; increment millisec reg
inc millisec

; half second?
jr ult,exit_time

; don't toggle colon
jr z,test_sw

; sw 1 pressed?
add MINUTES,#1

doa MINUTES

cp MINUTES,#60

; sixty minutes?
jr ult,inc_half_sec

; convert to bcd
clr MINUTES

; sw 2 pressed?
jr inc_half_sec

; 1:00?
add HOURS,#1

da HOURS

cp HOURS,#13

; set to zero
jr ult,inc_half_sec

; convert to bcd
ld HOURS,#1

; toggle colon bit now
inc HALF_SECOND

clr millisec

; one second?
cp HALF_SECOND,#2

; exit if not
jp ult,exit_time

; set to zero
clr HALF_SECOND

; increment sec
add seconds,#1

da seconds

; convert to bcd
cp seconds,#60

; sixty seconds?
jr ult,exit_time

; set to zero
clr seconds

; inc minutes
add minutes,#1

da minutes

; convert to bcd
cp minutes,#60

; sixty minutes?
jr ult,exit_time

; not yet
clr minutes

; set to zero
add hours,#1

da hours

; convert to bcd
cp hours,#13

; 1:00?
jr ult,exit_time

; exit
ld hours,#1

; set to 1:00
set hrs:
s

; convert to individual bcd
exit_time:
call time_convert
e

; return to orig reg pointer
pop rp

; return from int
iret
This subroutine checks to see if the time-set switches are pressed.

```Assembly
; test_sw:
push rp
srp #WORK_REG
ld data,p3 ; get switch data
com data
and data,#$03
; only first two bits
cp data,#0
jr eq,clear_sw
tm data,#1 ; min pressed?
jr z,test_hrs ; no
inc sw_count ; inc counter
cp sw_count,#2 ; debounced?
jr ult,exit_sw ; not yet
or STATUS,#$0000001b ; set bit
jr exit_sw ; exit
test_hrs:

; test_hrs:

; exit_sw:
pop rp
ret

; clear_sw:
clr STATUS ; reset sw status bits
clr sw_count ; reset debounce counter
pop rp
ret

; load_time:

; load_time:

; load_table:

; load_table:

; index_num:

; index_num:

; no_index:

; no_index:

; load_time_ret:

; load_time_ret:
```

This subroutine loads the time data into the RAM buffer.

```Assembly
load_time:
push rp ; save current reg pointer
srp #WORK_REG ; point to working reg
load_table:
ld r12,#pointer ; load contents
ld address_hi,#$hb led_table ; load hi address of table
ld address_lo,#$lb led_table ; load lo address of table
cp r12,#0 ; is it zero?
jr eq,no_index ; if yes, don’t step thru table
incw address ; step thru table
djnz r12,index_num ; index if not zero
no_index:
1de data,@address ; load segments
and p2,#$80
or p2,data
ld p3,ring_counter ; turn on digit
dec pointer ; inc reg location
cp pointer,#SECONDS_HI ; at ending reg?
jr ugt,load_time_ret ; exit
ld pointer,#HOURS_HI ; start at beginning
cp @pointer,#0
jr ne,load_time_ret
dec pointer ; don’t display leading zero
rr ring_counter ; rotate counter
pop rp ; return to time regs
iret
```
led_table:

| .byte   | 00111111B | ZERO          |
| .byte   | 00000110B | ONE           |
| .byte   | 01011011B | TWO           |
| .byte   | 01001111B | THREE         |
| .byte   | 01100110B | FOUR          |
| .byte   | 01101101B | FIVE          |
| .byte   | 01111101B | SIX           |
| .byte   | 00000111B | SEVEN         |
| .byte   | 01111111B | EIGHT         |
| .byte   | 01100111B | NINE          |

.end
TIMEKEEPING BASED ON THE 60 Hz LINE FREQUENCY

; This clock routine uses the Zilog Z86E08 to keep time. The 60 Hz line
; frequency causes an interrupt to the CPU every half cycle using one of the
; on-board comparators. The time registers are then incremented, and the dis-
; play refreshed at 1/60 second intervals. The time is displayed on a Three-
; Five Systems TSM6X34 Four Digit Display. The display has serial data and
; clock inputs, along with on-board display drivers. Port P0 provides the
; clock, data, and reset lines for the display. Port P2 is available for
; user options. The time set switch is connected to the second comparator
; input, and when pressed, advances the time at a 60 Hz rate.
; This program was written by Don Owen Newquist on May 16, 1992.

;-----------------------------------------------------------------------------
TIME_REG   .equ  10h
counter     .equ  r0
bit_count   .equ  r1
address_hi  .equ  r4
address_lo  .equ  r5
address     .equ  r4
pointer     .equ  r6
data        .equ  r7
milliseconds .equ  r8
seconds     .equ  r9
minutes     .equ  r10
hours       .equ  r11
minutes_lo  .equ  r12
minutes_hi  .equ  r13
hours_lo    .equ  r14
hours_hi    .equ  r15
BUFFER     .equ  04h
START      .EQU  02
ENABLE_HI  .EQU  01
ONE        .EQU  02
ZERO       .EQU  %FD
CLOCK_HI   .EQU  04
CLOCK_LO   .EQU  %FB

.org  00
.word  00
.word  00
.word  00
.word  time
.word  00
.word  00
.org  000ch

11-8
;******************************************************************************
; Initialization
;******************************************************************************
init:          srp  #TIME_REG
            ld   p2m,#0       ; disable int
            di
            ld   p3m,#0       ; outputs on p2
            ld   p0m,#04      ; open drain on p2
            clr  p0
            clr  p3
            clr  irq
            ld   imr,#%08     ; outputs on p0, int stack
            ld   spl,#%80     ; p0 low
            clr  p0
            ld   imr,#%08     ; p3 low
            ld   sph          ; enable irq3
            inc  pointer      ; set stack pointer
            djnz  counter,clear_reg
            ld   @pointer,#6  ; point to time regs
            inc  pointer      ; six locations
            clr  @pointer     ; clear ram
            djnz  counter,clear_reg
            ld   hours,#%12    ; continue until all zero
            ld   @pointer,#18h ; start time at 12:00
            call  time_convert
            call  load_time
main_loop:     ei       ; enable interrupts
                jr   main_loop
;******************************************************************************
; This interrupt routine updates the time
;******************************************************************************
time:          tm   p3,#2       ; sw pressed?
                jr   nz,inc_minutes
                inc  milliseconds
                cp   milliseconds,#60
                jr   ult,exit_time
                clr  milliseconds
                inc  seconds
                cp   seconds,#60
                jr   ult,exit_time
                clr  seconds
inc_minutes:   add  minutes,#1
                da   minutes
                cp   minutes,#%60
                jr   ult,exit_time
                clr  minutes
                set_hrs:   add  hours,#1
                da   hours
                cp   hours,#%13
                jr   ult,exit_time
                ld   hours,#1
                exit_time:  call  time_convert
                              call  load_time
                              iret
                              ; convert to individual bcd
                              ; return from int
This routine converts the seconds, minutes, and hours BCD data into units and tens-of-units for displaying:

```
time_convert:  ld minutes_lo,minutes; transfer contents
               ld minutes_hi,minutes;
               and minutes_lo,#%0f; keep only lower bits
               swap minutes_hi; swap nibbles
               and minutes_hi,#%0f; keep only lower bits
               ld hours_lo,hours; transfer contents
               ld hours_hi,hours;
               and hours_lo,#%0f; keep only lower bits
               swap hours_hi;
               and hours_hi,#%0f; keep only lower bits
               ret; return
```

This subroutine loads the time data into the RAM buffer, then to the display:

```
load_time:     ld pointer,#minutes_lo; load contents
               ld BUFFER,#04; load buffer add
               ld counter,#4;
               ld address_hi,#^hb led_table; load hi address of table
load_table:    ld address_lo,#^lb led_table; load lo address of table
               ld data,@pointer;
               add address_lo,data;
no_index:      ldc data,@address; load segments
               ld @BUFFER,data;
               inc pointer;
               inc BUFFER;
               djnz counter,load_table;
               ld BUFFER,#4;
               ld counter,#3;
               ld p0,#START; clock & enable low, data hi
send_start:    call clock_out; clock the data
               djnz counter,send_start;
next_digit:    ld bit_count,#8;
               ld data,@BUFFER;
rotate:        rcf data;
               rrc nc,zero;
               or p0,#ONE;
               jr clock_it;
zero:          and p0,#ZERO
               call clock_out;
               djnz bit_count,rotate;
               inc BUFFER;
               djnz counter,next_digit;
               call clock_out; one more to load data
               ld p0,#ENABLE_HI; take enable line high
ret
```
;---------------------------------------------
; Clock Out LED Data
;---------------------------------------------
clock_out: or p0,#CLOCK_HI
        nop
        nop
        and p0,#CLOCK_LO
        nop
        nop
        nop
        ret

        .org $00f0

led_table:
        .byte 00111111B  ; ZERO
        .byte 00000110B  ; ONE
        .byte 01011011B  ; TWO
        .byte 01001111B  ; THREE
        .byte 01100110B  ; FOUR
        .byte 01101101B  ; FIVE
        .byte 01111101B  ; SIX
        .byte 00000111B  ; SEVEN
        .byte 01111111B  ; EIGHT
        .byte 01100111B  ; NINE

.end
SINE TABLE

05 ' Listing 1
10 '---------------------------------------------------------------
20 ' *** SIN-TAB.BAS ***
30 '
40 ' Purpose= to generate a complete sine table
50 ' of hex numbers in format... [TAB].byte[TAB]01h,02h,...,0Ch
60 '---------------------------------------------------------------
70 '
80 ON ERROR GOTO 590
90 DIM B$(256)
100 DIM C$(22)
110 DEFINT A,B
120 PI = 3.141593
130 C=360/256
140 GOTO 390
150 FOR I=0 TO 255
160 H=B+A*SIN((C*I*PI)/180)
170 A$=HEX$(H) 'HEX$ function rounds off H autom'ly
180 IF LEN(A$)>2 THEN PRINT "--Bad data: some bytes bigger than FFh":GOTO 430
190 IF LEN(A$)=1 THEN A$="0"+A$ 'ADD LEADING ZERO, IF NECESS.
200 B$(I+1)=A$+"h"
210 NEXT I
220 I=1 __
230 FOR J=1 TO 22
240 C$(J)=CHR$(9)+".byte"+CHR$(9) 'START W/[TAB].byte[TAB]
250 FOR K=1 TO 12
260 C$(J)=C$(J)+B$(I)+"," I=I+1:IF I>256 GOTO 290
270 NEXT K
280 L=LEN(C$(J))-1
290 C$(J)=LEFT$(C$(J),L) 'KILL LAST COMMA
300 NEXT J
310 OPEN "TABLE.ASC" FOR OUTPUT AS #1
320 FOR J=1 TO 22:PRINT #1,C$(J):NEXT J
330 CLOSE #1
340 BEEP:PRINT "SINE DATA STORED IN CURRENT DRIVE & DIRECTORY"
350 PRINT "IN ASCII FILE CALLED TABLE.ASC ":PRINT
360 PRINT "( Press any key )"
370 PRINT "$=INKEY$:IF $=" THEN 380 ELSE SYSTEM
380 CLS:LOCATE 10,35:PRINT "$= SIN-TAB ":PRINT
390 PRINT TAB(10);"Generates a 256-byte sine table in the current"
400 PRINT TAB(10);"directory by creating an ASCII file called: TABLE.ASC"
410 PRINT TAB(10);"(If the file already exists, it will be overwritten !!!)
420 PRINT TAB(10);"Sine wave is of the form: A sin(xt) where A= amplitude."
430 PRINT TAB(10);"What is amplitude? (range= 1 - 127) A= ";
440 LINE INPUT $;
450 A=VAL($):IF A<1 OR A>127 THEN GOTO 430
460 PRINT TAB(10);"An offset is needed to keep all values positive."
470 PRINT TAB(15);"Suggest:";A+1
480 PRINT TAB(20);"Is this okay? (Y/N)"

11-12
500 T$=INKEY$:IF T$="" GOTO 500
510 IF T$=CHR$(27) THEN SYSTEM
520 IF T$="Y" OR T$="y" THEN B=A+1:PRINT "WAIT...";GOTO 150
530 IF T$="N" OR T$="n" THEN 540 ELSE 500
540 PRINT TAB(10);"Enter new value (0-255): ";
550 LINE INPUT T$
560 B=VAL(T$):IF B<0 OR B>255 THEN PRINT "--Illegal value--";GOTO 470
570 PRINT "WAIT...";GOTO 150
580 ' *** ERROR TRAP ***
590 IF ERL<>320 GOTO 630
600 BEEP:PRINT:PRINT "Unable to create (or write to) file: TABLE.ASC"
610 PRINT " ( Press any key )"
620 RESUME 380
630 IF ERL=460 THEN PRINT "--Illegal value--";PRINT:RESUME 440
640 IF ERL=560 THEN PRINT "--Illegal value--";PRINT:RESUME 470
650 BEEP:PRINT:PRINT "*** ERROR ";ERR;" OCCURRED AT LINE";ERL:PRINT
660 PRINT " ( Press any key )"
670 RESUME 380
Using the Zilog Z86C06 SPI Bus

The SPI stuffs a lot of power into a small package. Utilized in either Master or Slave mode with the ability to talk to many different processors and work with a wide range of peripherals, it fits readily into the low cost/high performance bracket.

Introduction

The Zilog Z86C06 SPI (Serial Peripheral Interface) is a compact, powerful and cost effective microcontroller. Its small form factor (18 pins) does not limit the full power of the Z8 core. The Z86C06 provides 1K (on-board) of program memory, 124 general purpose registers and an SPI bus that can be used in master or slave mode. This combines to overcome any pin I/O limitations and unleash the full Z8 capabilities.

Zilog's Superintegration also reduces system cost because two analog comparators are provided. No external interrupt reset circuitry is required because an interrupt Power-On-Reset along with Low Voltage (brownout) protection has been incorporated internally. A crystal may be optionally bypassed because the part is capable of using an RC or LC clocking source. It is also ideal for battery applications because the part can be placed in SLEEP mode and draw less than 10 microamps (2 microamps typical) of supply current. The user has a number of options available both internal and external, to recover from SLEEP mode by using the Stop Mode Recovery Register.

SPI Bus Operation

The SPI bus is useful because many peripherals exist to directly support it. Among these are E2 Serial PROMS, Real Time Clock chips, A/D converters, Frequency Generators, and display drivers for LED, LCD and VF displays. The other facet of the SPI bus that makes it useful is its ability to communicate with other processors. This is a requirement for distributed processing systems. SPI communications between processors is done in a Master/Slave type of arrangement.

In order to understand how the SPI bus is utilized to communicate with some of the peripherals mentioned, the architecture of the SPI bus itself must first be understood. In the simplest application, the SPI bus consists of three lines: a serial clock line, a serial in line, and a serial out line. Most peripherals also require a /CS line so that multiple peripherals can be utilized on the bus. A simplified diagram is shown in Figure 1.
Figure 1. Simplified SPI Bus Configuration
The serial clock (SCLK) line synchronizes the transfer of data between the master and the peripheral or slave device. SCLK is generated and controlled by the master device. Typically, the serial data output is transmitted on the falling edge of SCLK and the receive data is captured on the rising edge of SCLK. The Z8 has the capability of altering this relationship by manipulating bit D5 of the SCON register. When the data is transmitted, the most significant bit is transmitted first.

Control of the SPI bus in the Z8 is done in the SCON register (location (C) 02 in the expanded register file). There are also two other SPI bus registers in the Z8 which are necessary: the TX/RCV buffer and the SPI compare register. A diagram of the bit manipulation of the SPI Control Register is given in Figure 2. The Master/Slave mode is controlled by bit D7 in the SCON register. In the master mode, the definition of the bits is defined as follows:

![SPI Control Register (SCON)](image)

(S) Used with Bit D7 equal to 0
(M) Used with Bit D7 equal to 1

* Default Setting After Reset.
Master Mode

D7 - Master/Slave Select: Programmed as a 1 to select master mode.

D6 - CLK Source Select: This is used to select the SCLK source for SPI bus. A 1 selects the TO output as the control clocking source; a 0 selects the internal clock (XTAL/2). If the internal clock is selected, the actual clocking source is modified according to bits D2 and D1.

D5 - Clock Phase: This bit controls the phase relationship of the latching of the receive data and the transmitted data. A 1 transmits data on the rising edge of SCLK and latches the receive data on the falling edge. A 0 performs the opposite.

D4 - Rx Character Available: This bit can be used to determine if the receive data buffer is full. If activated, this also activates IRQ3.

D3 - Compare Enable: This bit has no effect in Master Mode.

D2,D1 - Clock Divide: If the SCLK source in bit D6 was programmed as a 0, then these bits control how the internal clock is manipulated to generate the SCLK. In the non-low noise mode, the SCLK can be programmed to be 1/4, 1/8, 1/16, or 1/32 of the external XTAL1 frequency. Note that PCLK is 1/2 of the XTAL1 frequency when the part is not operating in low noise mode. TCLK is equal to the XTAL1 frequency if low noise mode is selected (bit D7 of the PCON register).

DO - SPI Enable: This bit controls the activation of the SPI bus. When the SPI bus is enabled, the SPI signals get mapped according to Table 1.

Table 1. SPI Pin Configuration

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
<th>Port Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>DI</td>
<td>Data In</td>
<td>P20</td>
</tr>
<tr>
<td>DO</td>
<td>Data Out</td>
<td>P27</td>
</tr>
<tr>
<td>SS</td>
<td>Slave Select</td>
<td>P35</td>
</tr>
<tr>
<td>SLCKS</td>
<td>PL Clock</td>
<td>P34</td>
</tr>
</tbody>
</table>

Slave Mode

In slave mode these same bits change their function and are defined as follows:

D7 - Master/Slave Select: Programmed as a 0 to select slave mode. No slave mode transactions are initiated unless the Slave Select input is active low.

D6 - CLK Source Select: This has no effect in slave mode since the SCLK is controlled by the SPI Master.

D5 - Clock Phase: This bit controls the phase relationship of the latching of the receive data and the transmitted data described above.

D4 - Rx Character Available: Same as above.

D3 - Compare Enable: This bit can be used to enable a stop mode recovery source. When the processor is in SLEEP mode, and a character has been transferred into the receive buffer, and if it matches the SPI compare register, then the controller recovers from stop mode.

D2 - DO SPI Port Enable: When the slave mode is active, this bit controls the activation of the Data-out signal. If a 1 is written in this location, then P27 is controlled by the P2M register and P2. A 0 enables data to be shifted out with the SCLK.

D1 - Rx Character Overrun: When a 1 is read in this location, then the receive buffer has been overrun. This condition must be reset by writing a 1 to this bit.

D0 - SPI Enable: Same as Master Mode.
Compare Mode

Another special mode, commonly used in high end communication devices, is the SPI's internal Compare/Wakeup mode (Figure 3). This mode, commonly known in communications terminology as "wake up," is a compare register and logic which monitors the received data when in the low power SLEEP modes. If a "match" occurs between the compare register and the received data, the Z86C06 "wakes up" and processes according to the programmed, "waking up" state or condition.

Figure 3. SPI Internal Compare/Wakeup Mode
CONNECTING THE Z8 SPI BUS TO THE XICOR X25C02 SPI E2PROM

One of the most popular peripherals for microcontrollers is the serial E²PROM. Economics is the obvious reason for this. It costs significantly less to add a serial E²PROM to a microcontroller than to purchase a microcontroller with embedded E² capability.

Connecting SPI peripherals to the Z8 is an easy task. An example of this is demonstrated by connecting the ZB6C06 to the XICOR X25C02 (Figure 4). Note that the X25C02 is in the standby mode, the standby current is specified to be less than 150 microamps. The sleep mode current of the Z8 is less than 10 microamps (2 microamps typical). Since the active current of the X25C02 is less than 2 milliamps, it may be advantageous to use a port pin to supply power to the serial E² part in order to minimize application sleep mode current.

An example of the source code required to interface the X25C02 to the Z86C06 is given in Appendix A (Listing 1). This example, given in the listing, exercises the X25C02 by writing to it and reading from it. The listing also has three general-purpose routines for SPI utilization; SPI_MASTER_INIT, SPI_ENABLE and SPI. They are used to initialize the bus as a master, enable the SPI bus, and write/read data from the SPI peripheral, respectively.

Figure 4. Z86C06 and XICOR X25C02 Application Interconnection
CONNECTING THE Z8 SPI BUS TO AN A/D CONVERTER

Port 3 of the Z86C06 includes two analog comparators (when the port is programmed to operate in the analog mode in the P3M register). In addition to using the internal comparators as general purpose comparators, they can also be configured and used as a 3-bit A/D (Figure 5).

If more resolution than three bits is required, there are two alternatives. First, use an output port pin to charge a capacitor, start a timer, discharge the capacitor, then use the timer and an analog comparator to calculate the input voltage. This method is shown in Figure 5. The required amount of resolution is achieved by adjusting R1 and C1 to provide a maximum time constant for the Z8 timers. The most common way to utilize this technique is to provide a precalculated lookup table based on the timer results.

While the method shown in Figure 6 is cost effective, it may not be accurate enough for some applications. When additional accuracy is required, it is possible to find A/D converters with SPI interfaces. One such part is a Motorola MC145053 that provides 10 bits of conversion resolution. The hardware interface for the Z86C06 and the MC145053 is shown in Figure 7. An example of the source code required to interface the MC145053 to the Z86C06 is given in Appendix B (Listing 2). The general purpose SPI routines used in the E2 application are also used to interface to the MC145053.

![Figure 5. Three-Bit Analog/Digital Converter](image)
Figure 6. Low Cost Analog/Digital Converter

Figure 7. Z86C06/MC145053 Hardware Interface

Notes:
When P20 is switched from '1' to '0'
\[ V_{\text{ANALOG}} = (V_{\text{CC}} - 0.7)e^{-t/RC} \]
An interrupt can be used to trigger the \( V_{\text{REF}} \) crossing point to capture a timer. \( V_{\text{ANALOG}} \) can be calculated from the timer value.
CONCLUSION

The SPI bus is a simple, three-wire serial interface that can be used to communicate with a number of different specialized peripherals. While not discussed in detail in this App Note, it also lends itself as an excellent communications protocol controller in a distributed processing architecture. The Z86C06 is a very powerful and highly integrated processor that supports the SPI bus.

Control of the SPI bus is simple as demonstrated with each of the examples. Because control of the SPI bus is easy with the Z8, there is little hardware and software overhead required to use it. This also makes it an excellent cost-effective choice as a processor in a distributed processing environment even where an analog interface is involved.
APPENDIX A
Listing 1

<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>LINE#</th>
<th>SOURCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00000000</td>
<td>;***********************************************************************</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>ZILOG / SPI INTERFACE</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Z86C06 AND X25C02</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>BY: LYN ZASTROW</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>COPYWRITE 1991</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>ASSEMBLER: ZILOG ASMS8 ASSEMBLER</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Z86C06 I/O UTILIZATION</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>PORT DEFINITIONS:</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>P20: SPI PROCESSOR INPUT DATA</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>P21: NOT USED</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>P22: NOT USED</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>P23: NOT USED</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>P24: NOT USED</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>P25: NOT USED</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>P26: X25C02 POWER CONTROL</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>P27: SPI PROCESSOR OUTPUT DATA</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>P31: NOT USED</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td>P32: NOT USED</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>P33: NOT USED</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td>P34: SCLK OUTPUT FROM PROCESSOR</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>P35: /CS OUTPUT FROM PROCESSOR</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td>P36: NOT USED</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>TIME CONSTANTS ARE BASED ON 8MHz OPERATIONAL FREQUENCY</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>;***********************************************************************</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td>------- REGISTER POINTER DECLARATIONS -------</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
<td>------- REGISTER DECLARATIONS -------</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>abs 00000000</td>
<td>37</td>
<td>TDATA</td>
<td>.EQU R0</td>
</tr>
<tr>
<td>abs 00000001</td>
<td>38</td>
<td>RXBUF</td>
<td>.EQU R1</td>
</tr>
<tr>
<td>abs 00000002</td>
<td>39</td>
<td>SCON</td>
<td>.EQU R2</td>
</tr>
<tr>
<td>abs 00000004</td>
<td>40</td>
<td>MCHR</td>
<td>.EQU R4</td>
</tr>
<tr>
<td>abs 00000005</td>
<td>41</td>
<td>TREAD</td>
<td>.EQU R5</td>
</tr>
<tr>
<td>abs 0000000e</td>
<td>42</td>
<td>MADDR</td>
<td>.EQU RR14</td>
</tr>
<tr>
<td>abs 0000000f</td>
<td>43</td>
<td>MDR</td>
<td>.EQU R15</td>
</tr>
<tr>
<td>abs 0000000b</td>
<td>44</td>
<td>SMR</td>
<td>.EQU R11</td>
</tr>
<tr>
<td>45</td>
<td></td>
<td>XDATA</td>
<td>.EQU %10</td>
</tr>
<tr>
<td>46</td>
<td></td>
<td>RXDATA</td>
<td>.EQU %11</td>
</tr>
<tr>
<td>47</td>
<td></td>
<td>CS</td>
<td>.EQU %12</td>
</tr>
<tr>
<td>48</td>
<td></td>
<td>DCS</td>
<td>.EQU %13</td>
</tr>
<tr>
<td>49</td>
<td></td>
<td>XADDR</td>
<td>.EQU %1C</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
0000000000000000df 000000000000000020 0000000000000000ef 000000000000000010 000000000000000006 000000000000000004 000000000000000005 000000000000000001 000000000000000003 000000000000000002 000000000000000002

000000000000000002 000000000000000002 000000000000000002 000000000000000002 000000000000000002 000000000000000002 000000000000000002 000000000000000002

53 ; __________________________________________ PROGRAM CONSTANTS __________________________________________
54 XCS .EQU 11011111b ;P35 = /CS FOR THE X25C02
55 XCSNOT .EQU 00100000b ;P35 = DEACTIVEm X25C02 /CS
56 RCARESET .EQU 11101111b ;RECEIVED CHAR AVAIL. RESET
57 RCAMASK .EQU 00010000b ;RECEIVED CHAR AVAIL. MASK
58
000000000000000006
59 WREN .EQU 00000110b ;SET WRITE ENABLE LATCH
60 WRDI .EQU 00000100b ;DISABLE WRITE ENABLE LATCH
61 RDSR .EQU 00000101b ;READ STATUS REGISTER
62 WRSR .EQU 00000001b ;WRITE STATUS REGISTER
63 READ .EQU 00000011b ;READ MEMORY LOCATION
64 WRITE .EQU 00000010b ;WRITE MEMORY LOCATION
65
000000000000000002 000000000000000002 000000000000000002 000000000000000002

000000000000000002 000000000000000002 000000000000000002 000000000000000002

66 WEIMASK .EQU 00000010b ;WRITE ENABLE LATCH MASK OF STATUS REG
67 NOBP .EQU 00000010b ;WRSR MASK FOR NO BLOCK PROTECT
68 BMPASK .EQU 00000110b ;BLOCK PROTECT MASK FOR RDSR
69 ;
70 ;
71 ;
72 ;
73 ; LOAD INTERRUPT VECTOR JUMP TABLE:
74 .ORG 0000!
75 ;
76 ;
77 .WORD INTRET ;IRQ0 - PORT P32 - PROGRAMMABLE EDGE
78 .WORD INTRET ;IRQ1 - PORT P33 - NEGATIVE EDGE ONLY
79 .WORD INTRET ;IRQ2 - PORT P31 - PROGRAMMABLE EDGE
80 .WORD INTRET ;IRQ3 - SPI
81 .WORD INTRET ;IRQ4 - TO - INTERNAL
82 .WORD INTRET ;IRQ5 - Tl - INTERNAL
83 ;
84 ;******************************************************
85 ;
86 ; INITIALIZE THE OPERATION OF THE CONTROLLER
87 ;
88 ;******************************************************
89 ;
90 JP START
91
92 MESSAGE:
93 .ASCIC 'THIS ASCII DATA WILL BE STORED IN THE X25C02.'
APPENDIX A
Listing 1
(Continued)

.ASCII 'IN REAL APPLICATIONS, IT COULD BE USED TO STORE SERIAL NUMBERS.

.ASCII 'PROGRAMMABLE PRESETS, OR CONFIGURATION INFORMATION.

000000ad 4f4e2ea0

000000b1 9f
000000b2 8f
000000b3 b002
000000b5 e6f601
000000b8 e603e0
000000bb e6f701
000000be e6f92f
000000c1 e6f804
000000c4 3110
000000c6 e6ff7f
000000c9 d6
000000cc d6
000000cf e612df
000000d2 e61320

000000c5 96 START:
000000c6 98 E1
000000c7 99 DI
000000c8 100
000000c9 101
000000ca 102
000000cb 103
000000cc 104 CLR P2
000000cd 105
000000ce 106
000000cf 107 LD P2M,#00000001B
000000d0 108
000000d1 109 LD P3,##80
000000d2 110
000000d3 111 LD P3M,#81
000000d4 112
000000d5 113 LD IPR,#00101111B
000000d6 114
000000d7 115 LD POLK,#404
000000d8 116 SRP #A10
000000d9 117 LD SPL,#7F
000000da 118 CALL SPI_MASTER_INIT
000000db 119 CALL SPI_ENABLE
000000dc 120
000000dd 121 LD CS,#XCS
000000de 122 LD DCS,#XCSNOT
000000ef 123
000000f0 124 ;
000000f1 125 ;****************************************************************************** 1
000000f2 126 ;
000000f3 127 ; WRITE MABLE AND STATUS BEFORE PROCEEDING - ALSO CODE TO
000000f4 128 ; CHECK STATUS OF THE BLOCK PROTECT BITS AND UNPROTECT THE CHIP. BEFORE
000000f5 129 ; DOING ANYTHING, TURN ON THE X25C02 POWER
000000f6 130 ;
000000f7 131 ;******************************************************************************
132 ;
133 POWER_ON:
134 OR P2,#01000000B ;TURN ON THE POWER TO THE X25C02. THERE
135 NOP ; MAY NEED TO BE SOME NOPs INSERTED TO
136 NOP ; ACCOUNT FOR CAPACITANCE CHARGING.
137 ;
138 WRITE_ENABLE:
139 LD XDATA,#WREN ;PROVIDE WRITE ENABLE CAPABILITY
140 CALL SPI ;
141 OR P3,DCS ;DE-ACTIVATE THE CHIP SELECT OUTPUT
142
143 READ_STATUS:
144 LD XDATA,#RDSR ;READ THE STATUS REGISTER
145 CALL SPI ;
146 LD XDATA,#00 ;TRANSMIT DUMMY ADDRESS AND
147 CALL SPI ;GET DATA FROM PERIPHERAL
148 OR P3,DCS ;DE-ACTIVATE THE CHIP SELECT OUTPUT
149
150 VERIFY_WE:
151 TM RDATA,#WREN ;CHECK THE WRITE ENABLE BIT IN STATUS
152 JR EQ,WRITE_ENABLE ;WRITE ENABLE MASK WAS NOT SET
153
154 BLOCK_PROTECT:
155 TM RDATA,#BPMASK ;CHECK BLOCK PROTECT BITS IN STATUS
156 JR EQ,WR_BYTE_MESS ;NO PROTECTION ACTIVATED
157 LD XDATA,#NOPR ;SELECT NO BLOCK PROTECT
158 CALL SPI ;
159 OR P3,DCS ;DE-ACTIVATE THE CHIP SELECT
160
161 ;
162 ;******************************************************************************
163 ;
164 ; CODE TO DEMONSTRATE TWO DIFFERENT METHODS OF WRITING DATA TO THE X25C02. *
165 ; THE FIRST METHOD WRITES ONE BYTE AT A TIME, AND THE SECOND METHOD WILL *
166 ; WRITE FOUR BYTES AT A TIME. *
167 ;
168 ;******************************************************************************
169 ;
170 WR_BYTE_MESS:
171 ;WRITE THE FIRST LINE OF THE MESSAGE ONE
172 ;BYTE AT A TIME
173 ;STORE THE MESSAGE ADDRESS POINTER
174 ;STORE THE MESSAGE STARTING AT LOCATION 00H
175 NEXT_BYTE:
176 ;SEND THE WRITE COMMAND TO THE X25C02
177 ;SEND THE ADDRESS TO STORE THE WRITE DATA
178 ;SEND THE MESSAGE DATA TO THE WRITE BUFFER
179 ;DE-ACTIVATE THE CHIP SELECT OUTPUT
180 ;INCREMENT THE MESSAGE ADDRESS POINTER
181 ;INCREMENT X25C02 ADDRESS
182 ;CHECK IF END OF CHARACTER STRING
183 ;INCREMENT X25C02 ADDRESS
184 ;
185 ;******************************************************************************
186 ;
187 ;******************************************************************************
188 ;
189 ;******************************************************************************
190 ;
191 ;******************************************************************************
192 ;
193 ;******************************************************************************
194 ;
195 ;******************************************************************************
196 ;
197 ;******************************************************************************
198 ;
199 ;******************************************************************************
200 ;
201 ;******************************************************************************
202 ;
203 ;******************************************************************************
204 ;
205 ;******************************************************************************
206 ;
APPENDIX A
Listing 1 (Continued)

00000125 eb** 185 JR NE, WR_4BYTE ; DONE WITH FIRST STRING
00000127 8be1 186 JR NEXT_BYTE ; REPEAT LOOP UNTIL MESSAGE IS LOADED

00000129 4e04 188 WR_4BYTE: ; WRITE THE NEXT LINE OF THE MESSAGE 4 BYTES
0000012b 61002 LCD MCHR, #4 ; AT A TIME
0000012d 760000 189 XDATA, #WRITE ; X25C02 CAN PROCESS UP TO 4 BYTES IN A ROW
0000012f e41003 190 LD XDATA, XADDR ; SEND THE WRITE COMMAND TO THE X25C02
00000131 411000 191 LD XDATA, XADDR ; SEND THE ADDRESS TO STORE THE WRITE DATA

00000133 303 192 CALL SPI ;
00000135 910000 193 CALL SPI ;

00000137 c20e 194 NEXT_CHAR: ; GET THE NEXT BYTE OF THE MESSAGE

00000139 d61000 195 INC XADDR ; MOVE THE MESSAGE POINTER

0000013b 760080 196 INC XADDR ; INCREMENT THE MESSAGE ADDRESS POINTER

0000013d 760080 197 INC XADDR ; INCREMENT X25C02 ADDRESS

0000013f 4e05 198 INC XADDR ; CHECK IF END OF CHARACTER STRING

00000141 eb** 199 LD XDATA, XADDR ; DONE WITH WRITING SECOND STRING

00000143 4a1000 200 CALL SPI ; GET THE NEXT BYTE OF THE MESSAGE

00000145 441303 201 JR NE, RD_MESSAGE ; DE-ACTIVATE THE CHIP SELECT OUTPUT

00000147 8d0000+0129, 202 RD_MESSAGE: ; READ A LINE OF MESSAGE FROM THE EEPROM:

0000014a 440000 203 JP WR_4BYTE ; REPEAT UNTIL MESSAGE STORED

0000014d 441303 204 OR P3, DCS ;
0000014f 441303 205 OR P3, DCS ;

00000151 441303 206 INC MDR, P3, DCS ;
00000153 441303 207 ;
00000155 441303 208 ; CODE TO DEMONSTRATE HOW TO READ DATA FROM THE X25C02. BEFORE READING,
00000157 441303 209 ; THE WRITE DISABLE INSTRUCTION IS SENT TO PROTECT THE INFORMATION
00000159 441303 210 ;

0000015b d61000 211 ;*******************************************************************************

0000015d 441303 212 ; *

0000015f 441303 213 ; CODE TO DEMONSTRATE HOW TO READ DATA FROM THE X25C02. BEFORE READING,

00000160 441303 214 ; THE WRITE DISABLE INSTRUCTION IS SENT TO PROTECT THE INFORMATION

00000162 441303 215 ;

00000164 441303 216 OR P3, DCS ;
00000166 441303 217 OR P3, DCS ;
00000168 441303 218 OR P3, DCS ;

0000016a 441303 219 OR P3, DCS ;
0000016c 441303 220 OR P3, DCS ;

0000016e 441303 221 OR P3, DCS ;
00000170 441303 222 OR P3, DCS ;
00000172 441303 223 OR P3, DCS ;
00000174 441303 224 OR P3, DCS ;

00000176 441303 225 OR P3, DCS ;
00000178 441303 226 OR P3, DCS ;
0000017a 441303 227 OR P3, DCS ;

0000017c 441303 228 OR P3, DCS ;
0000017e 441303 229 OR P3, DCS ;

00000180 441303 230 OR P3, DCS ;
00000182 441303 231 OR P3, DCS ;

00000184 441303 232 OR P3, DCS ;
00000186 441303 233 OR P3, DCS ;
00000188 441303 234 OR P3, DCS ;

0000018a 441303 235 OR P3, DCS ;
0000018c 441303 236 OR P3, DCS ;

0000018e 441303 237 ;

**USING THE Z86C06 SPI Bus**

**APPLICATION NOTE**

---

**TURN OFF POWER TO THE X25C02, AND PLACE THE Z86C06 TO SLEEP**

---

```
00000179 b002
0000017b 10fd
0000011d e6fd0f
00000180 bc28
00000182 50fd
00000184 ff
00000185 6f
00000186 70fd
00000188 e6fd0c
0000018d 50fd
```

---

```
238 ;************************************************************************************
239 ;
240 ; TURN OFF POWER TO THE X25C02
241 ;
242 ;************************************************************************************
243 ;
244 CLR P2 ;TURN OFF POWER TO X25C02
245
246 PUSH RP ;SAVE THE CURRENT REGISTER POINTER
247 LD RP,#160F ;ACCESS THE STOP MODE RECOVERY REGISTER
248 LD SHC,#0101000B ;CONFIGURE STOP MODE FOR A HIGH LEVEL
249
250 POP RP ;RESET THE REGISTER POINTER
251
252 NOP ;CLEAR THE PIPELINE
253 STOP ;STOP THE PROCESSOR
254
255 ; NOTE THAT WHEN THE PROCESSOR WAKES UP, IT
256 ; WILL RUN EVERYTHING ALL OVER AGAIN AND
257 ; THEN GOES BACK TO SLEEP. A DECISION LOOP
258 ; COULD BE PUT AT THE BEGINNING OF THE CODE
259 ; TO CHECK THE PROCESSOR WAS ACTIVATED AS
260 ; THE RESULT OF A FOR, OR AS THE RESULT OF
261 ; A STOP MODE RECOVERY
262
263 ;************************************************************************************
264 ;
265 ;************************************************************************************
266 ; SPI_MASTER_INIT
267
268 ; INITIALIZE THE SPI BUS FOR MASTER MODE OF OPERATION
269 ; SPECIAL NOTE: THIS ROUTINE DOES NOT ENABLE THE SPI BUS
270 ;
271 ;************************************************************************************
272 ;
273 ; SPI_MASTER_INIT:
274 ;************************************************************************************
275 ;
276 ;************************************************************************************
277 ;
278 ;************************************************************************************
279 ;
280 ;************************************************************************************
281 ;
282 ;************************************************************************************
283 ;
284 ;************************************************************************************
285 ;
286 ;************************************************************************************
287 ;
288 ;************************************************************************************
289 ;
```

---

**INITIALIZE SPI BUS FOR MASTER MODE OF OPERATION**

---

```
00000179 b002
0000017b 70fd
0000017d e6fd0f
00000180 bc28
00000182 50fd
00000184 ff
00000185 6f
00000186 70fd
00000188 e6fd0c
0000018d 50fd
```

---

```
238 ;************************************************************************************
239 ;
240 ; TURN OFF POWER TO THE X25C02, AND PLACE THE Z86C06 TO SLEEP
241 ;
242 ;************************************************************************************
243 ;
244 CLR P2 ;TURN OFF POWER TO X25C02
245
246 PUSH RP ;SAVE THE CURRENT REGISTER POINTER
247 LD RP,#160F ;ACCESS THE STOP MODE RECOVERY REGISTER
248 LD SHC,#0101000B ;CONFIGURE STOP MODE FOR A HIGH LEVEL
249
250 POP RP ;RESET THE REGISTER POINTER
251
252 NOP ;CLEAR THE PIPELINE
253 STOP ;STOP THE PROCESSOR
254
255 ; NOTE THAT WHEN THE PROCESSOR WAKES UP, IT
256 ; WILL RUN EVERYTHING ALL OVER AGAIN AND
257 ; THEN GOES BACK TO SLEEP. A DECISION LOOP
258 ; COULD BE PUT AT THE BEGINNING OF THE CODE
259 ; TO CHECK THE PROCESSOR WAS ACTIVATED AS
260 ; THE RESULT OF A FOR, OR AS THE RESULT OF
261 ; A STOP MODE RECOVERY
262
263 ;************************************************************************************
264 ;
265 ;************************************************************************************
266 ; SPI_MASTER_INIT
267
268 ; INITIALIZE THE SPI BUS FOR MASTER MODE OF OPERATION
269 ; SPECIAL NOTE: THIS ROUTINE DOES NOT ENABLE THE SPI BUS
270 ;
271 ;************************************************************************************
272 ;
273 ; SPI_MASTER_INIT:
274 ;************************************************************************************
275 ;
276 ;************************************************************************************
277 ;
278 ;************************************************************************************
279 ;
280 ;************************************************************************************
281 ;
282 ;************************************************************************************
283 ;
284 ;************************************************************************************
285 ;
286 ;************************************************************************************
287 ;
288 ;************************************************************************************
289 ;
```

---

**INTIALIZE SPI BUS FOR MASTER MODE OF OPERATION**

---

```
00000186 70fd
00000186 e6fd0c
0000011b 2c8a
00000186 50fd
0000018f af
```

---

```
272 PUSH RP ;SAVE THE CURRENT REGISTER POINTER
273 LD RP,#160C ;ACCESS THE SPI EXPANDED REGISTER FILE
274 LD SCON,#1001010B ;ENABLE MASTER MODE
275
276 ; SET CLOCK SOURCE TO TCLK
277 ; XMIT ON FALLING EDGE, RECEIVE ON RISING
278 ; RESET THE RCV CHAR AVAILABLE
279 ; DISABLE THE COMPARE WAKE-UP FEATURE
280 POP RP ;Retrieve Working Register Pointer
281 RET
282
283 ;************************************************************************************
284 ;
285 ;************************************************************************************
286 ;
287 ;************************************************************************************
288 ;
289 ;************************************************************************************

---

**INITIALIZE SPI BUS FOR MASTER MODE OF OPERATION**

---

```
00000186 70fd
00000186 e6fd0c
0000011b 2c8a
00000186 50fd
0000018f af
```

---

```
272 PUSH RP ;SAVE THE CURRENT REGISTER POINTER
273 LD RP,#160C ;ACCESS THE SPI EXPANDED REGISTER FILE
274 LD SCON,#1001010B ;ENABLE MASTER MODE
275
276 ; SET CLOCK SOURCE TO TCLK
277 ; XMIT ON FALLING EDGE, RECEIVE ON RISING
278 ; RESET THE RCV CHAR AVAILABLE
279 ; DISABLE THE COMPARE WAKE-UP FEATURE
280 POP RP ;Retrieve Working Register Pointer
281 RET
282
283 ;************************************************************************************
284 ;
285 ;************************************************************************************
286 ;
287 ;************************************************************************************
288 ;
289 ;************************************************************************************

---

**INITIALIZE SPI BUS FOR MASTER MODE OF OPERATION**

---

```
272 PUSH RP ;SAVE THE CURRENT REGISTER POINTER
273 LD RP,#160C ;ACCESS THE SPI EXPANDED REGISTER FILE
274 LD SCON,#1001010B ;ENABLE MASTER MODE
275
276 ; SET CLOCK SOURCE TO TCLK
277 ; XMIT ON FALLING EDGE, RECEIVE ON RISING
278 ; RESET THE RCV CHAR AVAILABLE
279 ; DISABLE THE COMPARE WAKE-UP FEATURE
280 POP RP ;Retrieve Working Register Pointer
281 RET
282
283 ;************************************************************************************
284 ;
285 ;************************************************************************************
286 ;
287 ;************************************************************************************
288 ;
289 ;************************************************************************************

---

**INITIALIZE SPI BUS FOR MASTER MODE OF OPERATION**

---

```
272 PUSH RP ;SAVE THE CURRENT REGISTER POINTER
273 LD RP,#160C ;ACCESS THE SPI EXPANDED REGISTER FILE
274 LD SCON,#1001010B ;ENABLE MASTER MODE
275
276 ; SET CLOCK SOURCE TO TCLK
277 ; XMIT ON FALLING EDGE, RECEIVE ON RISING
278 ; RESET THE RCV CHAR AVAILABLE
279 ; DISABLE THE COMPARE WAKE-UP FEATURE
280 POP RP ;Retrieve Working Register Pointer
281 RET
282
283 ;************************************************************************************
284 ;
285 ;************************************************************************************
286 ;
287 ;************************************************************************************
288 ;
289 ;************************************************************************************
```

---

**INITIALIZE SPI BUS FOR MASTER MODE OF OPERATION**

---

```
272 PUSH RP ;SAVE THE CURRENT REGISTER POINTER
273 LD RP,#160C ;ACCESS THE SPI EXPANDED REGISTER FILE
274 LD SCON,#1001010B ;ENABLE MASTER MODE
275
276 ; SET CLOCK SOURCE TO TCLK
277 ; XMIT ON FALLING EDGE, RECEIVE ON RISING
278 ; RESET THE RCV CHAR AVAILABLE
279 ; DISABLE THE COMPARE WAKE-UP FEATURE
280 POP RP ;Retrieve Working Register Pointer
281 RET
282
283 ;************************************************************************************
284 ;
285 ;************************************************************************************
286 ;
287 ;************************************************************************************
288 ;
289 ;************************************************************************************
```

---

**INITIALIZE SPI BUS FOR MASTER MODE OF OPERATION**

---

```
272 PUSH RP ;SAVE THE CURRENT REGISTER POINTER
273 LD RP,#160C ;ACCESS THE SPI EXPANDED REGISTER FILE
274 LD SCON,#1001010B ;ENABLE MASTER MODE
275
276 ; SET CLOCK SOURCE TO TCLK
277 ; XMIT ON FALLING EDGE, RECEIVE ON RISING
278 ; RESET THE RCV CHAR AVAILABLE
279 ; DISABLE THE COMPARE WAKE-UP FEATURE
280 POP RP ;Retrieve Working Register Pointer
281 RET
282
283 ;************************************************************************************
284 ;
285 ;************************************************************************************
286 ;
287 ;************************************************************************************
288 ;
289 ;************************************************************************************
```
APPENDIX B

Listing 2

290 ;
291 SPI_ENABLE:
292  PUSH RP ;SAVE THE CURRENT REGISTER POINTER
293  LD RP,#40C
294  OR SCON,#0000001B
295  POP RP
296  RET
297 ;
298 ;******************************************************************************
299 ;
300 ;  SPI
301 ;
302 ;  XMIT A BYTE OF DATA ON THE SPI BUS FOUND IN REGISTER XDATA. ALSO, PLACE
303 ;  A RECEIVED CHARACTER IN REGISTER RDATA. THIS ROUTINE WILL ACTIVATE THE
304 ;  /CS LINE, BUT WILL NOT DE-ACTIVATE THE LINE IN CASE MULTIPLE WRITES OR
305 ;  READS ARE GOING TO TAKE PLACE.
306 ;
307 ;  XDATA - REGISTER LOCATION 10h
308 ;  RDATA - REGISTER LOCATION 11h
309 ;
310 ;******************************************************************************
311 ;
312 SPI:
313  PUSH RP ;SAVE THE CURRENT REGISTER POINTER
314  AND P3,CS
315  LD RP,#40C
316  AND SCON,#RCARESET
317  LD RXBUF,XDATA
318  RCV_CHECK:
319  TM SCON,#RCMASK
320  JR EQ.RCV_CHECK ;CHECK IF CHARACTER AVAILALBE IN RCVR
321  LD RXDATA,RXBUF
322  POP RP
323  RET
324 ;
325 ;******************************************************************************
326 ;
327 ;  INTERRUPTS
328 ;
329 ;  THERE WERE NO INTERRUPTS USED BY THIS PROGRAM. AN INTERRUPT COULD HAVE
330 ;  BEEN USED TO READ THE SPI DATA (IRQ3).
331 ;
332 ;******************************************************************************
333 ;
334 ;
335 .END

APPENDIX B

Using the Z86C06 SPI Bus

Application Note
LOC OBJ

LINE | SOURCE |
--- | --- |
1 | *********************************************** |
2 | |
3 | ZILOG / SPI INTERFACE |
4 | Z86C06 AND MC145053 |
5 | |
6 | BY: LYN ZASTROW |
7 | COPYRIGHT 1991 |
8 | |
9 | ASSEMBLER: ZILOG ASM8 ASSEMBLER |
10 | |
11 | Z86C06 I/O UTILIZATION |
12 | PORT DEFINITIONS: |
13 | P20: SPI PROCESSOR INPUT DATA |
14 | P21: NOT USED |
15 | P22: NOT USED |
16 | P23: NOT USED |
17 | P24: NOT USED |
18 | P25: NOT USED |
19 | P26: NOT USED |
20 | P27: SPI PROCESSOR OUTPUT DATA |
21 | P31: END OF CONVERSION SIGNAL |
22 | P32: NOT USED |
23 | P33: NOT USED |
24 | P34: SCLK OUTPUT FROM PROCESSOR |
25 | P35: NOT USED |
26 | P36: /CS OUTPUT FROM PROCESSOR |
27 | |
28 | TIME CONSTANTS ARE BASED ON 8MHz OPERATIONAL FREQUENCY |
29 | |
30 | *********************************************** |
31 | |
32 | |
33 | --- REGISTER POINTER DECLARATIONS --- |
34 | |
35 | |
36 | |

abs 00000000  TDATA .EQU R0 :SAME AS XDATA, BUT WORKING REGISTER (%10)
abs 00000001  RXBUF .EQU R1 :SPI RX & XMIT BUFFER (%11)
abs 00000002  SCON .EQU R2 :SPI CONTROL REGISTER (EXTENDED REG) (%12)
abs 00000004  HHEAD .EQU R4 :HIGH BYTE OF CONVERSION (%14)
abs 00000005  LOAD .EQU R5 :LOW BYTE OF A/D CONVERSION (%15)
abs 00000006  SLOOP .EQU R6 :SHIFT LOOP COUNTER FOR CONVERSION (%16)

000000000000000000000010  XDATA .EQU %10 :SPI XMIT DATA
000000000000000000000011  RXDATA .EQU %11 :SPI RECEIVED DATA
000000000000000000000012  CS .EQU %12 :REGISTER CONTAINING /CS
000000000000000000000013  DCS .EQU %13 :REGISTER DE-ACTIVATING CS

11-31
APPENDIX B
Listing 2
(Continued)

50 ; PROGRAM CONSTANTS
51 XCS .EQU 10111111B ;P36 = /CS FOR THE MC145053
52 XCNOT .EQU 01000000B ;P36 = DEACTIVE MC145053 /CS
53 RCARESET .EQU 11101111B ;RECEIVED CHAR AVAIL RESET
54 RCAMASK .EQU 00010000B ;RECEIVED CHAR AVAIL MASK
55 ECAMASK .EQU 00000010B ;MASK FOR P31 - END OF CONVERSION

56 57 ; ANALOG MULTIPLEXOR ADDRESSES
58 ANO .EQU 900 ; ANALOG CHANNEL 0
59 AN1 .EQU 910 ; ANALOG CHANNEL 1
60 AN2 .EQU 920 ; ANALOG CHANNEL 2
61 AN3 .EQU 930 ; ANALOG CHANNEL 3
62 AN4 .EQU 940 ; ANALOG CHANNEL 4
63 HSAMEL .EQU 950 ; HALF SCALE CHANNEL
64 ZSCALE .EQU 9C0 ; ZERO SCALE CHANNEL
65 FSCALE .EQU 9D0 ; FULL SCALE CHANNEL

66 67 ; INTERRUPT VECTORS
68 ; LOAD INTERRUPT VECTOR JUMP TABLE:
69 70 .ORG 0000H
71 72 ;
73 ;
74 .WORLD INTRET ; IRQO - PORT P32 - PROGRAMMABLE EDGE
75 .WORLD INTRET ; IRQ1 - PORT P33 - NEGATIVE EDGE ONLY
76 .WORLD INTRET ; IRQ2 - PORT P31 - PROGRAMMABLE EDGE
77 .WORLD INTRET ; IRQ3 - SPI
78 .WORLD INTRET ; IRQ4 - TO - INTERNAL
79 .WORLD INTRET ; IRQ5 - T1 - INTERNAL
80 ;
81 ;******************************************************************************
82 ; INITIALIZE THE OPERATION OF THE CONTROLLER
83 ;******************************************************************************
84 ;
85 ;******************************************************************************
86 ;
87 START:
88 EI ;RESET THE INTERRUPT FLIP-FLOP
89 DI ;
90 ; UPON POWER-UP, THE STOP MODE RECOVERY REGISTER IS SET FOR A POR SOURCE ONLY.
91 ; THE WDT IS NOT ACTIVE UNTIL A WDT INSTRUCTION IS EXECUTED.
92 ;
93 94 CLR P2 ; CLEAR THE P2 OUTPUT BUFFER
94 95 LD P2H, #00000001B ; SET ALL P2 BITS TO OUTPUT EXCEPT SPI DATA INPUT LINE
96 97 LD P3H, #F00 ; SET THE P3 OUTPUTS (P34-P36)
98 99 LD P3M, #F01 ; DEFAULT SCLK LOW, REST ARE HIGH
100 101 LD IFR, #00101111B ; SET THE P3 INPUTS TO BE IN ANALOG MODE
102 ; PRIORITY THE INTERRUPTS AS FOLLOWS: ; IRQ3>IRQ5>IRQ4>IRQ1>IRQ0>IRQ2

11-32
000001c e6f804 103 : LD  PO1M, #04 ;SET INTERNAL STACK
000001f 3110 104 : SRP  #10 ;SET A DEFAULT WORKING REGISTER SET
0000021 e6ff71 105 : LD  SPL, #7F ;
0000024 d61www 106 :  
0000027 d61www 107 : CALL  SPI_MASTER_INIT ;INITIALIZE THE SPI REGISTERS
000002a e612bf 108 : CALL  SPI_ENABLE ;ENABLE THE SPI BUS
000002d e61340 109 :  
000002e e610b0 110 : LD  CS, #XCS ;LOAD THE CHIP SELECT VARIABLE
0000030 111 : LD  DCS, #XCSNOT ;LOAD THE CHIP DE-SELECT VARIABLE

;**********************************************************************************************
0000030 120 TEST_HALF:
0000033 d610b0 121 : LD  XDATA, #HSCALE ;PROVIDE ADDRESS FOR THE HALF SCALE TEST
0000036 a6e402 122 : CALL  CONVERT ;PERFORM THE A/D CONVERSION
0000039 ff 123 : CP  HIAD, #402 ;CHECK THE HI BYTE
000003a e6e500 124 : NOP ;THIS WOULD NORMALLY BE A CONDITIONAL JUMP
000003c ff 125 : CP  LOAD, #000 ;CHECK THE LO BYTE
000003d 126 : NOP ;THIS WOULD NORMALLY BE A CONDITIONAL JUMP

;**********************************************************************************************
000003e 128 TEST_ZERO:
0000041 d610c0 129 : LD  XDATA, #ZSCALE ;PROVIDE ADDRESS FOR THE ZERO SCALE TEST
0000044 a6e400 130 : CALL  CONVERT ;PERFORM THE A/D CONVERSION
0000047 ff 131 : CP  HIAD, #400 ;CHECK THE HI BYTE
0000048 a6e500 132 : NOP ;THIS WOULD NORMALLY BE A CONDITIONAL JUMP
000004b ff 133 : CP  LOAD, #000 ;CHECK THE LO BYTE
000004c 134 : NOP ;THIS WOULD NORMALLY BE A CONDITIONAL JUMP

;**********************************************************************************************
000004c 136 TEST_FULL:
0000050 e610d0 137 : LD  XDATA, #FSCALE ;PROVIDE ADDRESS FOR THE FULL SCALE TEST
0000054 d61000 138 : CALL  CONVERT ;PERFORM THE A/D CONVERSION
0000058 a6e03 139 : CP  HIAD, #03 ;CHECK THE HI BYTE
000005b ff 140 : NOP ;THIS WOULD NORMALLY BE A CONDITIONAL JUMP
000005e a6e5ff 141 : CP  LOAD, #FF ;CHECK THE LO BYTE
000005f 142 : NOP ;THIS WOULD NORMALLY BE A CONDITIONAL JUMP

;**********************************************************************************************
0000005a 144 :
0000005a e61000 152 : LD  XDATA, #0 ;START AT ANALOG CHANNEL 0
0000005d d61www 153 : CALL  CONVERT ;PERFORM THE A/D CONVERSION
APPENDIX B
Listing 2
(Continued)

00000060 154 LOOP:
00000060 e61000 155 LD XDATA,#0 ;START THE LOOP AGAIN BECAUSE OF CONVERSION
00000063 d6www 156 ; DELAY IN MULTIPLEXER
00000066 ff 157 CALL CONVERT ;PERFORM THE A/D CONVERSION
00000067 061010 158 NOP ; THIS WOULD NORMALLY BE CODE TO UTILIZE
0000006a a61050 159 ; HEAD AND LOAD FOR THE CHANNEL BEING
0000006d 1bf1 160 ; CONVERTED
0000006e 1bf1 161 ADD XDATA,#10 ;DO THE NEXT CHANNEL
00000072 86000030 162 CP XDATA,#60 ;CHECK IF ALL CHANNELS CONVERTED
00000075 d61www 163 JR LT, LOOP ;CONTINUE UNTIL ALL CHANNELS CONVERTED
00000078 441303 164
0000007b d6www 165
0000007e d61www 166 ;
00000081 4811 167
00000083 5811 168 ;
00000086 5811 169;
00000088 441303 170 JP TEST_HALF ;DO IT ALL OVER AGAIN!
0000008b af 171 ;*************************************************************************************
0000008d 172 ;
0000008f 173 ;*******************************************************************************
00000092 174 ;
00000094 175 ; CONVERT
00000096 176 ;
00000098 177 ; SEND THE CHANNEL TO CONVERT, WAIT FOR THE EOC SIGNAL, AND READ THE
0000009c 178 ; DATA. THE ANALOG CHANNEL MUST BE STORED IN XDATA. THE 10 BIT DATA IS
000000a0 179 ; IS RETURNED HIAD AND LOAD.
000000a3 d6www 180 ;
000000a6 181 ;*******************************************************************************
000000a8 182 ;
000000ab d6www 183 CONVERT:
000000ac 184 CALL SPI ;TRANSMIT DATA IN 16 BIT FORMAT
000000af 185 CALL SPI ;
000000be 186 OR P3,DCS ;DE-ACTIVATE THE CHIP SELECT OUTPUT
000000c1 187 CALL CONVERT_END ;WAIT FOR END OF CONVERSION
000000c7 e61000 188 CALL SPI ;READ THE DATA - HI BYTE
000000cc d61www 189 LD HIAD,RDATA ;STORE THE HIGH BYTE
000000d2 d6www 190 CALL SPI ;READ THE LAST TWO BITS
000000d6 191 LD LOAD,RDATA ;STORE THE LOW BYTE
000000d8 e61000 192 OR P3,DCS ;DEACTIVATE THE CHIP SELECT OUTPUT
000000de 193 LD SLOOP,#6 ;SHIFT THE DATA TO MAKE IT LOOK LIKE TRUE
000000e4 194 ; 16 BIT DATA WITH THE 6 MSBs SET TO 0. IF
000000e7 195 IF 8 BIT DATA IS DESIRED, THEN THIS DOES
000000e9 196 ; NOT NEED TO BE DONE AND THE HIAD RESULT
000000eb 197 ; COULD BE USED AS IS.
000000f5 198 SRA HIAD ;SHIFT THE HI BYTE, PUT LSB IN CARRY FLAG
000000f8 199 RRC LOAD ;SHIFT THE LO BYTE PUTTING THE CARRY FLAG
000000fa 200 ; IN THE MSB
000000fd 201 DJNZ SLOOP,SHIFT ;CONTINUE THE SHIFTING UNTIL COMPLETE
00000100 202 AND HIAD,#0000011B ;MASK OFF THE UPPER 6 BITS AFTER THE SHIFT
00000103 203 ;
00000106 af 204 RET
00000106 af 205 ;

11-34
206 ;******************************************************************************
207 ; CONVERT_END
208 ;
209 ;
210 ; POLL THE END OF CONVERT LINE UNTIL A/D CONVERSION IS DONE
211 ;
212 ;******************************************************************************
213 ;
214 CONVERT_END:
215 TM P3,#ECMASK ; CHECK THE EC BIT
216 JR EQ.CONVERT_END ; WAIT UNTIL THE EC BIT IS HIGH
217 RET
218 ;
219 ;******************************************************************************
220 ;
221 ; SPI_MASTER_INIT
222 ;
223 ; INITIALIZE THE SPI BUS FOR MASTER MODE OF OPERATION
224 ; SPECIAL NOTE: THIS ROUTINE DOES NOT ENABLE THE SPI BUS
225 ;
226 ;******************************************************************************
227 ;
228 SPI_MASTER_INIT:
229 PUSH RP ; SAVE THE CURRENT REGISTER POINTER
230 LD RP,#%0c ; ACCESS THE SPI EXPANDED REGISTER FILE
231 LD SCON,#10001010b ; ENABLE MASTER MODE
232 ; SET CLOCK SOURCE TO TCLK
233 ; XMIT ON FALLING EDGE, RECEIVE ON RISING
234 ; RESET THE RCV CHAR AVAILABLE
235 ; DISABLE THE COMPARE WAKE-UP FEATURE
236 ; SET CLOCK RATE (1 MHz WITH 8 MHz CRYSTAL)
237 POP RP ; RETRIEVE WORKING REGISTER POINTER
238 RET
239 ;
240 ;******************************************************************************
241 ;
242 ; SPI_ENABLE
243 ;
244 ; ENABLE THE SPI BUS - THIS STARTS THE SCLK
245 ;
246 ;******************************************************************************
247 ;
248 SPI_ENABLE:
249 PUSH RP ; SAVE THE CURRENT REGISTER POINTER
250 LD RP,#%0c ; ACCESS THE SPI EXPANDED REGISTER FILE
251 OR SCON,#0000001b ; ENABLE THE SPI BUS
252 POP RP ; RETURN TO NORMAL REGISTERS
253 RET
254 ;
DTMF Tone Generation Using the Z8® CCP

In many applications a microprocessor must access phone lines for communications or data exchange. This has traditionally been accomplished by adding a DTMF tone encoder and a 3.58 MHz crystal; however, the Z8's pulse width modulation capabilities allow DTMF tone generation in software.

INTRODUCTION AND THEORY OF OPERATION

The program outlined below generates DTMF tones using a pulse width modulation (PWM) algorithm. PWM is used to vary the DC level of the output by varying the duty cycle (the "on" time divided by the cycle period) of the square wave. Varying the "on" time by a sine function and then feeding the output through a lowpass filter yields a sine wave.

Sine values are determined by the Basic program in Listing 1. The sine table contains 256 entries, which in turn contain hexadecimal values representing a sine function for 360 degrees. These values are indexed and loaded into T1 at a sampling rate that, according to Nyquist, must be at least twice the highest frequency tone that we want to reproduce. Since the highest frequency for this application is 1477 Hz, the sampling rate must be at least twice this, or 2954 samples per second. The higher the sampling rate, the greater the accuracy. In the example illustrated here a sample rate of 12000 samples per second is used both for higher accuracy and ease in filtering.

Since we are, in effect, producing two tones, two pointers are used to fetch the next value in the look-up table: one for row frequencies, and one for column frequencies. The frequency of the resulting sine wave can be calculated by multiplying the number of steps in the sine look-up table (256) by the desired frequency, divided by the sampling rate. This offset value is added to the current pointer, which then fetches the next hex number from the look-up table. This is done for both the row and column frequencies. The two are added, then loaded into the timer register (T1).

The lowpass filter was chosen to have a corner frequency of the lowest column frequency, or 1209 Hz. At this point, the column frequencies will be at least 3 dB below the row frequencies. However, telephone lines themselves act as a large-scale lowpass filter, and by the time the tones reach the telephone switching equipment, the amplitude should be the same. The spec therefore calls for the column frequencies to be 3 dB higher than the row frequencies. In the telephone industry this is known as "twist."

This adjustment can be made in software by taking the hex value from the look-up table for the column frequency and doing a "rotate left." This results in twice the amplitude for the column frequency — 6 dB gain, or 3 dB up from the row frequency, just where we want to be. Overall dB level from the output of the lowpass filter can be adjusted with a potentiometer.
DTMF TONE GENERATION

Z86C06

12.00 MHz

+5V

VCC

GND

TELEPHONE KEYPAD
3x4 MATRIX

1 2 3
4 5 6
7 8 9
* 0 #

13 1K
.1

14 1K
.1

15 1K
.1

16 1K
.1

17 1K
.1

18 1K
.1

+5V

27 pF

27 pF
This program, using the Zilog Z8, is designed to produce DTMF tones without an external DAC. The tones are produced using PWM, using one of the timers to vary the duty cycle of the output pulses. A sine look-up table is indexed at a sample rate well above the highest tone (1447 Hz). Both column and row values from the look-up table are added in software to produce the PWM output at the sample frequency. These hex values are loaded into a timer, which determines the pulse width. At Terminal Count (TC), the port pin is taken low, and the difference between this period and the sample period is loaded into the timer. When run through a low-pass filter, the result is an accurate DTMF tone. Crystal frequency is 12.0000 MHz. The keypad columns are output from P24-6, while the row inputs are connected to P20-3.

This program was written on 6-2-92 by Don Owen Newquist, Zilog, Inc.
DTMF TONE GENERATION USING
THE Z8® CCP® APPLICATION NOTE

BOUNCE .EQU %04
COUNTER_1 .EQU %05
KEY_CNT .EQU %06
KEY_TEMP .EQU %07
TEMP_1 .EQU %08
SCAN .EQU %09

.org 0000h
.word 0
.word 0
.word 0
.word 0
.word key_scan
.word timer_1

;--------------------------------------------------------------------------;
;                INITIALIZATION                                          ;
;--------------------------------------------------------------------------;

.org 000ch       ; disable int
di
ld rp,#0f0h      ; point to
ld r0,#0feh      ; pcon reg
ld r11,#0       ; crystal div by 2
srp #WORK_REG1   ; lowest bank
ld p3m,#0       ; default
ld p01m,#04h    ; int stack
ld sp1,#80h     ; stack at highest ram
ld irq,#00h
clr imr          ; clear int mask
ld pre0,#05h     ; cont mode
t0,#00           ; sampling time = 125 microsec
ld pre1,#6h      ; one shot mode
clr p3           ; clear port 3
ld imr,#10h      ; enable irq4
ld ipr,#0bh      ; irq5 > irq4
ld tmr,#03h      ; load & en t0
ld pointer_hi,#^hb sine ; get lookup table
ld pointer_lo,#^1b sine ; address
ld r_freq_hi,#1fh ; 697
ld r_freq_lo,#0efh
ld c_freq_hi,#17h ; 1209
ld c_freq_lo,#26h
ld row_inc_hi,#00 ; 697
ld row_inc_lo,#01
ld col_inc_hi,#00 ; 1209
ld col_inc_lo,#01
ld row_val,#10h
ld col_val,#10h
clr BOUNCE       ; debounce counter
clr KEY_CNT
clr KEY_TEMP
clr TEMP_1
clr SCAN
ei

check_bounce:    jr check_bounce ;
KEYBOARD SCAN ROUTINE

; KEYBOARD SCAN ROUTINE

key_scan: push rp
        srp #WORK_REG0
        ld p2m,#00001111b ; out for p27-p24, in p23-p20
        ld scan,#11101111b ; load scan byte

load_scan: ld p2,scan
        nop
        nop
        ld temp_l,p2 ; load contents of p2
        ld counter_l,#0 ; load counter

row_loop: inc key_cnt
        scf
        rrc temp_l ; rotate right
        jr c,no_keys
        cp key_temp,key_cnt ; same key?
        jr ne,load_keys ; not the same
        inc bounce ; increment bounce counter
        cp bounce,#2 ; bounce = 2 ?
        jr ult,load_keys
        call dtmf_out ; call transmit routine
        jr exit

load_keys: ld key_temp,key_cnt ; transfer key data
        clr key_cnt
        pop rp
        iret

no_keys: inc counter_l ; looking for a one
        cp counter_l,#4 ; keep looking if not all rows
        jr ult,row_loop ; look again at next row
        scf ; set carry flag
        rlc scan ; rotate to scan next column
        jr c,load_scan ; scan next column if no carry

exit: clr bounce
        clr key_cnt
        clr key_temp
        clr counter_l
        pop rp
        iret

no_irq: iret
DTMF Tone Generation Using
The Z8 CCP® Application Note

Get Keyswitch Offset Values

```assembly
;-----------------------------------------------------------------------------
dtmf_out:
    push rp
    srp #WORK_REG1
    ld tmr,#00h ; reset t0 bits
    ld imr,#20h ; reset t0 vector
    ld pre0,#4 ; one-shot mode
    ld offset_hi,#^hb offset_tbl
    ld offset_lo,#^lb offset_tbl
    sub KEY_CNT,#1
    cp KEY_CNT,#0
    jr eq,no_index

index_loop:
    incw offset
    incw offset
    incw offset
    dec KEY_CNT
    cp KEY_CNT,#0
    jr ne,index_loop

no_index:
    ldc r_freq_hi,@offset
    incw offset ;
    ldc r_freq_lo,@offset
    incw offset ;
    ldc c_freq_hi,@offset
    incw offset
    ldc c_freq_lo,@offset
    ld t0,#ctval

enable:
    ei

;-----------------------------------------------------------------------------

; DTMF Generation Routine
;-----------------------------------------------------------------------------
start:
    or p3,#40h ;
    or tmr,#0fh ;
tcm p2,#0fh ;
jr z,exit_dtmf ;
dtmf_loop:
    tm irq,#10h ;
jr z,dtmf_loop ;
    and irq,#0efh ;
jr start ;
exit_dtmf:
    ld imr,#10h ; timer 0 int
    ld pre0,#01h ; cont mode, full count
    ld imr,#10h ; t0 only
    ld tmr,#03h ;
    ld t0,#00 ;
    pop rp ;
    ret ;
```

11-42
;-----------------------------------------------------------------------------
; Timer 1 Interrupt Routine
;-----------------------------------------------------------------------------
timer_1:  xor p3,#40h ; toggle port pin
         rcf
         add row_inc_lo,r_freq_lo
         adc row_inc_hi,r_freq_hi
         ld pointer_lo,row_inc_hi
         ldc row_val,@pointer
         add col_inc_lo,c_freq_lo
         adc col_inc_hi,c_freq_hi
         ld pointer_lo,col_inc_hi
         ldc col_val,@pointer
         rl col_val
         add row_val,col_val
         iret

load_t1:  ld t1,row_val
         iret

革新:

.org 180h

offset_tbl:

.byte 1fh,0bfh,37h,70h
.byte 23h,00h,37h,70h
.byte 26h,0cfh,37h,70h
.byte 2ah,0d7h,37h,70h
.byte 1fh,0bfh,3dh,30h
.byte 23h,00h,3dh,30h
.byte 26h,0cfh,3dh,30h
.byte 2ah,0d7h,3dh,30h
.byte 1fh,0bfh,43h,0afh
.byte 23h,00h,43h,0afh
.byte 26h,0cfh,43h,0afh
.byte 2ah,0d7h,43h,0afh

.sine:

.org 200h

.byte 18h,18h,18h,18h,18h,19h,19h,19h,19h,19h,19h,19h,19h,19h,19h,19h,19h
.byte 1ah,1ah,1ah,1ah,1ah,1ah,1ah,1ah,1ah,1ah,1ah
.byte 1ch,1ch,1ch,1ch,1ch,1ch,1ch,1ch,1ch,1ch,1ch,1ch,1ch,1ch,1ch,1ch,1ch
.byte 1eh,1eh,1eh,1eh,1eh,1eh,1eh,1eh,1eh,1eh,1eh
.byte 1fh,1fh,1fh,1fh,1fh,1fh,1fh,1fh,1fh,1fh,1fh,1fh,1fh,1fh,1fh,1fh,1fh
.byte 1ff,1ff,1ff,1ff,1ff,1ff,1ff,1ff,1ff,1ff,1ff,1ff,1ff,1ff,1ff,1ff,1ff
.byte 1eh,1eh,1eh,1eh,1eh,1eh,1eh,1eh,1eh,1eh,1eh
.byte 1dh,1dh,1dh,1dh,1dh,1dh,1dh,1dh,1dh,1dh,1dh
.byte 1bh,1bh,1bh,1bh,1bh,1bh,1bh,1bh,1bh,1bh,1bh
.byte 1ah,1ah,1ah,1ah,1ah,1ah,1ah,1ah,1ah,1ah,1ah
.byte 19h,19h,19h,19h,19h,19h,19h,19h,19h,19h,19h
.byte 17h,17h,17h,17h,17h,17h,17h,17h,17h,17h,17h
.byte 15h,15h,15h,15h,15h,15h,15h,15h,15h,15h,15h
.byte 13h,13h,13h,13h,13h,13h,13h,13h,13h,13h,13h
.byte 11h,11h,11h,11h,11h,11h,11h,11h,11h,11h,11h
.byte 19h,19h,19h,19h,19h,19h,19h,19h,19h,19h
.byte 17h,17h,17h,17h,17h,17h,17h,17h,17h,17h
.byte 15h,15h,15h,15h,15h,15h,15h,15h,15h,15h,15h
.byte 13h,13h,13h,13h,13h,13h,13h,13h,13h,13h,13h
.byte 11h,11h,11h,11h,11h,11h,11h,11h,11h,11h,11h
.byte 19h,19h,19h,19h,19h,19h,19h,19h,19h
.byte 17h,17h,17h,17h,17h,17h,17h,17h,17h
.
.end
Many applications require asynchronous communications with the outside world. This Application Note presents one method for accomplishing serial communications through software rather than hardware by taking advantage of the flexibility of the Z8® CCP™.

**INTRODUCTION**

Straightforward serial communications between Zilog Z8® CCP™ processors and the outside world are possible using the 9600 baud software UART described in this Application Note. This technique is particularly well suited to projects which require asynchronous communications, but where a hardware UART is cost- or space-prohibitive. Designed to be used with the Zilog CCP family, running at an 8 MHz clock frequency, this approach requires the use of only one counter/timer and two port pins.

**THEORY OF OPERATION**

Essentially, the UART remains in an idle loop until it either (a) senses an interrupt request on port P3-1 (Receive), or (b) it senses a character in the BUFFER register to be transmitted. Eleven bits in total are counted: one start bit, eight data bits, one parity bit, and one stop bit. A typical schematic is provided in Figure 1.

---

![Figure 1. Serial CCP UART](image-url)
RECEIVE MODE

The UART goes into operation upon sensing a receive character via an interrupt (Low-going edge) request on P3-3. It then prepares to receive one start bit (R10), eight data bits (R10), one parity bit, and one stop bit (R10). Subsequently, it programs the T0 timer as a down-counter which reaches a terminal count in 0.104 ms. This is the sampling rate. The interrupt vector is loaded with the receive subroutine address, which is jumped to when an IRQ 4 interrupt is generated. This routine decrements the bit counters, and rotates the data in register R10. When a byte is assembled, it waits for the parity and stop bit, which then completes the transfer.

TRANSMIT MODE

The UART is ready to transmit at any time, and senses a transmit character when BUFFER is not zero. When BUFFER is loaded with data to be transmitted, the jump vector is loaded with the transmit subroutine address. The counter/timer T0 is then loaded for a terminal count at 0.104 ms. This is the sampling rate for the transmission. When T0 times out, an IRQ 4 interrupt is generated. The program jumps to the IRQ 4 interrupt routine, and then immediately jumps to the transmit routine. The transmit routine decrements the bit counter and rotates out the transmit data.
;********************************************************************
;** SERIAL DATATRANSFER APPLICATION FOR CCP-SERIES                     *
;** WITHOUT HARDWARE UART.                                            *
;** WRITTEN BY W. MANSFELD, ZILOG GERMANY / feb. 1991/ dec.91      *
;**                                                                 *
;********************************************************************

TO_VECTOR_H: .EQU %20
TO_VECTOR_L: .EQU %21
BYTE_BUFF: .EQU %22 ;CONTAINS Rx RETURN BYTE ( RP=20 )

.ORG %00

.WORD DUMMY
.WORD DUMMY
.WORD DUMMY
.WORD SER_IN     ;RX BYTE INTERRUPT
.WORD TO_INT     ;TO JUMP VECTOR INTERRUPT
.WORD DUMMY

.ORG %0C

DI
LD  SPL,#%7E
CLR  SPH
LD  IPR,#1   ;INT. PRIORITY
SRP  #%20  ;SET REG. POINTER
         ;refer to upper EQU’s
LD  P3M,#0000001B ;PORT 3 NORMAL MODE
LD  P01M,#00000100B ;SELECT INTERNAL STACK

; PORTS 0,1,2 NOT USED IN THIS APP.
LD  IMR,#00011000B ;ENABLE T0,RX INT
EI

; SEND A ENDLESS BYTE STREAM VIA Tx LINE

TEST_LOOP:
LD  R4,#'5'   ;ASCII 5 TEST BYTE
CALL SER_OUT  ;SEND BYTE
JR TEST_LOOP
;**************************************************************
; SERIAL RECEIVE ON INTERRUPT P3,0
; INTERRUPT WHEN A CHARACTER COMES FROM HOST
; input: none
;**************************************************************
SER_IN:
; SAVE USED REGISTERS
PUSH R15 ; TRANSPARENT /DATA BUFFER
PUSH R6 ; SERIAL BIT COUNT ++
PUSH IMR ;SAVE INT. MODE REGISTER
LD IMR,#00010000B ;SELECTE TO INT. ONLY

; PUT A DELAY LOOP TO THAT POINT IF BAUDRATE IS BELOW OR
; 9600 BAUD, TO REACH MIDDLE OF START BIT BEFORE STARTING
; RX BIT WINDOW

; SERIAL BAUDRATE
; USE TO FOR BIT-CENTRE INTERRUPT
LD PREO,#00000111B ;MODIFY PRESCALER PREO
LD T0,#104 ;BAUD RATE = 9600 / ON 8MHz CRYSTAL
LD TMR,#00000011B ;START/PRESET TO

; MODIFY TO INT. JUMP VECTOR
LD R7,#^HB SERIN_TO_INT
LD T0_VECTOR_H,R7
LD R7,#^LB SERIN_TO_INT
LD T0_VECTOR_L,R7
EI
LD R6,#8 ;READ 8 DATABYTES
CLR R15 ;CLEAR RECEIVESHIFT REGISTER(BUFFER)
SER_IN1:
CP R6,#0 ;8 BITS RECEIVED ?
JR NE,SER_IN1 ;WAIT FOR BITS RECEIVED
LD BYTE_BUFF,R15 ;SAVE RX BYTE IN BUFFER
LD R6,#1 ;READ 1 STOPBIT
CLR R15
SER_IN2:
CP R6,#0 ;READ DATA DONE?
JR NE,SER_IN2 ;WAIT FOR BIT RECEIVED
DI
AND TMR,#1111100B ;TO OFF
POP IMR ;INTERRUPT MASK BACK
CLR IRQ
POP R6
POP R15
IRET ;RX DONE
; OUTPUT A BYTE THROUGH THE SERIAL
; INPUT : R4 CONTAINS TX-DAT
; OUTPUT: R4 HOLDS INPUT DATA
;
SER_OUT:

; SAVE USED REGISTERS
PUSH R5 ; TO INT. VECTOR POINTER LB
PUSH R7 ; TX DATA
PUSH R10 ; BIT COUNTER
PUSH IMR ; FOR TO int. ONLY ACTIVE HEREIN
PUSH R4 ; SAVE DATA
LD IMR,#00010000B ; TO int. ON
;
SERIAL BAUDRATE, USE TO FOR TIMING
LD T0,#104 ; BAUD RATE = 9600
LD PRE0,#00000101B ; PRESCALER VALUE 1
LD TMR,#03 ; LOAD + ENABLE TO
;
MODIFY TO INT. JUMP VECTOR FOR Tx
LD R7,#11B SER_OUT_TO_INT
LD T0_VECTOR_H,R7
LD R7,#0B SER_OUT_TO_INT
LD T0_VECTOR_L,R7
EI
;
NOW SEND DUMMY 0-bit and one START BIT
LD R10,#2 ; SEND 2 BIT: 0 + START-BIT
LD R7,#11111101B ; VALUE for the 2 bits
;
SER_OUT1:
CP R10,#0 ; DONE?
JR NZ,SER_OUT1
LD R10,#8 ; SEND 8 BITS OF DATA
POP R7 ; TAKE DATABYTE FROM STACK
PUSH R7 ; CORRECT THE STACK ( PUT BACK THAT BYTE )
;
CHECK IF ALL DATABITS SENT
SER_OUT2:
CP R10,#0 ; DONE?
JR NZ,SER_OUT2
;
NOW SEND STOP-BIT
LD R10,#1 ; SEND ONE BIT ( STOP BIT )
LD R7,#%FF ; VALUE
;
SER_OUT3:
CP R10,#0 ; DONE?
JR NZ,SER_OUT3
DI
AND TMR,#11111100B ; STOP TO
POP R4
POP IMR ; OLD IMR BACK
POP R10
POP R7
POP R5
EI
RET ; RET ONLY, WAS NOT AN INT. SERVICE
;************************************************
;* TRANSMIT TO INTERRUPT HANDLER                *
;* FOR BIT GENERATION TO P3,7                   *
;************************************************
SER_OUT_TO_INT:
  DEC    R10   ;BIT CNT DOWN
  RR     R7    ;SHIFT BIT
  JR     C,SER_O_TO_INT1  ;BIT = 1
  AND    P3,#%7F  ;BIT = 0
  IRET
SER_O_TO_INT1:
  OR      P3,#%80  ;BIT = 1
  IRET

; PROGRAMMABLE USE OF T0 ( RX or TX )
T0_INT:
  JP     @T0_VECTOR_H  ;INDIRECT TO int JUMP
          ;FOR Tx,RX bit read
DUMMY:
  IRET

;************************************************
;* RECEIVE TO INTERRUPT HANDLER                 *
;* FOR BIT READ FROM P3,0                       *
;************************************************
SERIN_TO_INT:
  DEC    R6     ;BIT COUNT
  PUSH   R15    ;SAVE FOR TEMPORARY USE
  LD     R15,P3 ;READ THE PORT / BIT Rx
  RR     R15    ;CHECK if 0 or 1
  POP    R15    ;NOW USE AS RECEIVE SHIFT BUFFER
  RRC    R15    ;SHIFT THE CARRY =0 OR 1 INTO BUFFER
  IRET

.END
THE VERSATILE Z86C08: THREE KEY FEATURES OF THIS Z8® MICROCONTROLLER

If you need D/A conversion, or a zero crossing detector, or a current sensing device ... Use the Z86C08’s dual comparator.

DUAL ANALOG COMPARATOR

Using the dual analog comparators on the Z86C08 in conjunction with several on-chip features, provides a cost effective way to monitor power failures and frequency excursions (comparator used as a zero crossing detector), as a blood pressure tester and digital readout (comparator used as an A/D converter), or as a current sensing device in automotive design to detect and subsequently shut off any short circuiting of relays, lights, monitors, etc.

In many microcontroller applications, the digital designer is often concerned with sampling and controlling non-digital elements within the system. However, when the designer is forced to deviate from the precise world of TTL logic and regulated 5 volt supplies, frequently, microcontroller architectures and specifications fall short in the areas of cost sensitivity and consumer orientation. Therefore, using the analog comparators in these specific areas are a few of the reliable, inexpensive design applications for the Z86C08.

Comparator Basics

The dual comparators share a common inverting terminal with non-inverting terminals bonded directly to external I/O ports (Figure 1). The comparators are enabled by a bit in the I/O port mode/control register. If bit D1 of R247 is zero, then the comparators are in digital mode. If D1 is one, then they are in analog mode. With the comparators disabled, the I/O ports are available for normal activities. These particular I/O ports can be used to generate external interrupt requests to the Z8®. With the comparators enabled, interrupts can also be generated.

The ideal comparator is a three terminal device (Figure 2). V1 is a non-inverting terminal. Signals entering at V2, the inverting terminal, exit V_{out} 180° out of phase. Since a comparator is essentially an operational amplifier, it has an associated gain. The open loop gain (no feedback) of a comparator is defined as the Voltage Out (V_{out}) over the Differential Input Voltage. The Differential Input Voltage is the voltage at the non-inverting input with respect to the inverting input. Thus gain is:

\[ G = \frac{V_{out}}{V_{1} - (V2)} = \frac{Voltage\ Out}{Differential\ Input\ Voltage} \]

The Inset Offset Voltage, the difference between V1 and V2, forces V_{out} to a specified level. The Input Offset Voltage is typically below 50 mV.

Zero Crossing Detect Applications

The dual comparator can be used as a zero crossing detector to monitor 110 VAC (or other power line parameters) and its frequency (Figure 3). Each time the voltage passes through zero an interrupt is generated. The outputs of the comparators on the Z86C08 connect directly to the on-chip CPU. When using the comparators to detect zero crossing of the signal, interrupts are generated at every crossing of a signal, interrupts are generated at every crossing. Interrupt subroutines can then calculate period and phase angle relationships between any two analog signals. The phase angle being critical when calculating power factor in power line circuits.

In the case of 110 VAC, 60 Hz power line, an interrupt is generated every 1/120 of a second. This means that whenever the monitor stops (no interrupts), there is a power fail or other problem which can be translated by a control device recovery action (Figure 4).

Frequency checks can also be made by zero crossing detection. Whenever frequency drifts from the normal monitoring zero points, interrupts are either increased (higher frequency) or decreased (lower frequency) from the norm. If necessary, appropriate action is then taken.
Zero Crossing Detect Applications (Cont’d.)
Another application is threshold detection for low voltage battery operated devices. Whenever the VBB drops below the Zener reference voltage level, an interrupt is generated to alert a control device or alarm.

The addition of two on-chip counter/timers further complement the above mentioned applications. Crystal precision timing is done on the period of zero crossings. The sum or difference of two separate analog signals can then be calculated. For example, negative or positive feedback is returned from the Z86C08 in closed loop calculations. In power circuits, a time-of-day clock could be implemented with a timer. Then, date and time of power failures and frequency excursions can be recorded. CMOS technology allows for battery backup.

Analog to Digital (A/D) Conversion
Accurate low speed A/D conversion is implemented with the Z86C08 using the dual slope or ratiometric method. With this method, a dv/dt is applied to the inverting terminal of a comparator. The analog input (V_{input}) signal is applied to the non-inverting terminal. The charge rate of the RC circuit is a dv/dt (Figure 5). As V_{REF} ramps upward from zero volts during time T_1, V_{REF} will exceed V_{INPUT}. This causes the comparator to change state and produce an interrupt. By using the on-chip timer, time T_1 can be quickly determined.

The RC circuit is immediately discharged over fixed time T_2 (Figure 6), where T_2 is determined by the time constant T_c = RC. Since the product of RC is only an approximate indicator of discharge time, a value of n should be multiplied to improve accuracy. A general guideline should equate n to 1.4. Then, T_2 = 1.4 RC. The dual slope A/D converter measures voltage by converting voltage into time intervals. Or,

\[
\frac{T_2}{T_1} = \frac{V_{INPUT}}{V_{REF}}, \text{ then, } V_{INPUT} = V_{REF} = \frac{T_2}{T_1}
\]

By using an I/O port on the Z86C08 as the V_{REF} input, interrupts generated by the comparators can alternately switch V_{REF} ON or OFF to perform the conversions.

Example: Blood Pressure Tester
A pressure transducer in a blood pressure tester is a good example of the dual slope A/D conversion method. A minimum system consists of display logic, Z86C08 circuitry and a transducer signal input (Figure 7). P00 outputs the appropriate signal to the RC ramp circuit of the V_{REF} input. The output from the pressure transducer (Figure 8) is a linear voltage response to the applied pressure. This signal is input to An2, the non-inverting terminal of the comparator.

In this configuration, the sampling cycle for the A/D conversion begins when a logical 1 is output on P00 and a timer is enabled. When the comparator transitions, an interrupt is generated, the timer is stopped and P00 is toggled to discharge the RC circuit. By storing the count T_1 and resetting the timer, the converter is now ready to take another sample. The value of V_{IN} is mathematically determined later and software algorithms are used to determine corresponding pressure.

The display is driven from a simple multiplexer circuit. The Z86C08 can sink large I_{OC} currents which reduces or eliminates buffering.

Current Sensing
The dual comparator is used as a current sensing device in many application areas, e.g., in automotive relays, lights, monitors, etc. In the automotive arena, current sensing is used in a typical case as shown in Figure 9a. If the functional block shorts, then current (I) surges causing voltage (V) to fall. When V reaches 2.5V, the comparator triggers an interrupt which allows software to enable an emergency shut off.
Comparator Basics

Figure 1. Dual Analog Comparator

Figure 2. Ideal Comparator
Zero Crossing Detect Applications

Figure 3. Zero Crossing Detector

110 VAC

Power Fail= No interrupts

● INTERRUPTS every 1/120 sec @ 60 Hz

Figure 4. Interrupt After Power Failure
Analog to Digital (A/D) Conversion

Figure 5. A/D Converter

Figure 6. Voltage vs. Time
Blood Pressure Tester

![Diagram of Blood Pressure Tester]

- **Cap = 30pf**
- **Enables Diastolic/Systolic Segments**
- **7 segment enable lines**
- **** Selects Diastolic/Systolic Segments

Figure 7. A/D Blood Pressure Test and Readout
Figure 8. Silicon Pressure Transducer
Current Sensing

Note:
R is large compared to the equivalent impedance of the Functional Block input. R1 and R2 are user-selectable and are generally in a 10K to 100K range of power dissipation considerations. R1 and R2 are determined from the following formula (Figure 9b.).

Figure 9a. Current Sensing

Figure 9b. Power Dissipation Formula
The Z86C08 controls a scrolling LED message display

Display text and graphics while scrolling the LED message with a minimum of hardware. The characters are displayed using a time-division multiplex technique with more than six characters easily added by software.

Introduction

Designed around the Zilog Z86C08, 18-pin microcontroller, this LED display is capable of displaying text as well as graphics, with hardware being kept to a minimum. The present design is configured to display up to six characters, but additional characters are easily added with minimal software changes.

The display uses a TDM (Time-Division Multiplexed) technique to display the characters. That is, each character segment is turned on for a few hundred microseconds at a time, then is "refreshed" every 18 ms.

For demonstration purposes, the software routine displays the time in hours, minutes, and seconds. Once every sixty seconds, a "secret message" scrolls across the display. Then, after the message is displayed, the program reverts back to displaying the time.

The Hardware

The Z86C08 (Figure 1) uses Port 2 for the row data and clocks the column data out of Port 0. PNP transistors are used to drive the rows, since the Z86C08 cannot source the required current directly. The characters are displayed in a 5x7 format, so only seven lines are needed out of Port 2. A logic Low turns on the transistors, while a logic High turns them off.
THE HARDWARE (Continued)

Figure 1. Z86C08 Circuit Functions
The columns are driven by six 74HCT164 shift registers (Figure 2). The 74HCT164s do not have the necessary sink capability to drive the LEDs, so the outputs of the shift registers drive six 75492 high-current buffers. These are capable of sinking 250 mA per pin continuously, with instantaneous current per column approaching 1.5 A.

Each 74HCT164 drives six columns; five character segments, plus one space. The last Q output from the shift register is then fed to the DATA input of the next shift register, while all CLEAR and CLOCK lines are wired in parallel.

Figure 2. LED Circuitry
THE HARDWARE (Continued)

To scan the display, a logic 1 is output at P0-2. Next, P0-1 is taken High, then immediately taken Low, along with P0-2. This clocks a logic 1 into the first shift register (column 1). Next a character segment is output at P2. The transistors that are turned on will source current for the LEDs with the column providing a sink path. The columns are left on for about 500 µs and then turned off. Now, the column data is shifted one and a new character segment is output. After the last column has been scanned, the display is "refreshed" starting at the first column again.

To set the time, two push-button switches are connected to Port 3 to adjust the hours and minutes.

THE SOFTWARE

The initialization part of the program configures the ports, timers, interrupts, etc. The size of the display buffer (FIFO) is determined by the number of columns in the display. The bottom of the display buffer starts at 20H. The upper limit of the display buffer can extend to 70H. This translates into a sixteen-character display. There is no need to have a display buffer larger than the display itself, since only that many characters can be viewed at a time. A power-up, the display is configured for showing the time.

The flowchart for the display appears in Figures 3a and 3b. To keep time, the internal clock was divided down by T0 to provide an interrupt every 5 ms. The interrupt routine increments a counter, and when 200 counts is reached, the seconds register is incremented by one. Also, when ten seconds is reached, tens-of-seconds is incremented. This counter continues to increment minutes, tens-of-minutes, hours, and tens-of-hours. Upon power-up, the display shows 12:00. When it is around 9:00, the leading hours digit is blanked. Time is adjusted by two push-button switches. When pressed, one increments the minutes register every second, while the other increments the hours register every second. The time data is stored in locations 09H-0EH.
1. Load T0 for TC of 5 ms (Time)
2. Load T1 for TC of .5 ms (Scan)
3. Load Hours and Minutes with 12:00
4. Load Buffer (FIFO) With Hours, Minutes and Seconds Segment Data
5. Take Data and Reset Lines High
6. Take Clock Line Low
7. Enable Interrupts

**Initialization**

Wait for Interrupts

**Time Interrupt Service Routine**

- SW1 Pressed?
- Yes: Set SW1 Bit In Status Reg
- No: Increment Milliseconds

- SW2 Pressed?
- Yes: Set SW2 Bit In Status Reg
- No: Milliseconds = 200?
  - No: SW1 Bit Set?
    - Yes: Increment Minutes
    - No: Return From Time Interrupt
  - Yes: Return From Time Interrupt

- Seconds = 60?
  - Yes: Increment Minutes
  - No: Minutes = 60?
    - Yes: Increment Hours
    - No: Hours = 13?
      - Yes: Set Hours to 01
      - No: Return From Time Interrupt

1. Clear Milliseconds
2. Increment Seconds

1. Clear Minutes
2. Increment Hours

1. Increment Minutes
2. Clear Seconds

**Figure 3a. Display Flowchart**
Figure 3b. Display Flowchart
At sixty-second intervals, the time display is blanked, and the internal message is scrolled across the display. The message is stored in an ASCII format. The individual ASCII characters index a look-up table, which converts the characters to a 6x7 format (first segment is a space). The software checks to see if all of the segments have been indexed at the beginning of the scan. If so, it then indexes the next character (Figures 4a and 4b).

**Figure 4a. Scrolling the Letter Z.**
THE SOFTWARE (Continued)

For scrolling messages, the display buffer acts like a FIFO (First In - First Out). The FIFO is cleared at power-up. When the internal message is being indexed, the character segments are queued up in the FIFO. The FIFO size is determined by the size of the display. At the end of each scan, the next character segment is indexed, and is stored at the bottom of the FIFO. The character segments are then shifted up the FIFO one location. This process continues until the entire message is displayed. At this time, a 0 is loaded into the FIFO at the beginning of each scan allowing the columns trailing the message to blank out. As the display is being scanned, the byte at each FIFO location is output at P2 as each column is turned on. The scrolling effect is created by shifting the FIFO data at the start of each scan (Figure 5).

![Diagram showing FIFO data shifting during scrolling](image)

Figure 4b. Scrolling the Letter Z.
Figure 5. Shifting FIFO Data
APPENDIX

SCROLL

LINE#  --- SOURCE ---
1;**********************************************************************************************;
2;                                                                                           ;
3;    This is a scrolling LED display routine using the Zilog Z86C08, 'C17                 ;
4;    18-pin CMOS processor. The processor's P2 port outputs the row data, and           ;
5;    the column data is clocked out of P0 into cascaded shift registers. It          ;
6;    has a real-time clock in software, and displays the time in hours, min-          ;
7;    utes, and seconds. At every 60-second interval, it blanks the screen and         ;
8;    displays an internal message. The message is stored in ROM, and can be up       ;
9;    to 127 characters in length. After scrolling the message, the program re-       ;
10;    sumes the time display, until the next 60 seconds.                               ;
11;    The size of the display is 36-columns long, enough to display six char-          ;
12;    acters. The display can be made longer, but the refresh time may need ad-       ;
13;    justing to eliminate flicker. Current refresh time is 500 microseconds,       ;
14;    using a crystal frequency of 8.00 MHz.                                         ;
15;    There are two pushbutton switches that are read to adjust the hours and        ;
16;    minutes. On power-up, the display will show 12:00.                             ;
17;    This program was written by Don Owen Newquist on 12-1-91.                      ;
18;**********************************************************************************************;

000000000000000010 19 WORK_REG  .equ 10h
abs 00000000 20 address_hi .equ r0
abs 00000001 21 address_lo .equ r1
abs 00000002 22 address .equ rr0
abs 00000004 23 pointer .equ r2
abs 00000005 24 zero_count .equ r4
abs 00000006 25 temp_1 .equ r5
abs 00000007 26 temp_2 .equ r6
abs 00000008 27 temp_3 .equ r7
abs 00000009 28 col_count .equ r8
abs 0000000a 29 seq_count .equ r9
000000000000000000 30 TIME_REG .equ 00h
abs 00000005 31 millisec .equ r5
abs 00000006 32 seconds .equ r6
abs 00000007 33 minutes .equ r7
abs 00000008 34 hours .equ r8
abs 00000009 35 seconds_lo .equ r9
abs 0000000a 36 seconds_hi .equ r10
abs 0000000b 37 minutes_lo .equ r11
abs 0000000c 38 minutes_hi .equ r12
abs 0000000d 39 hours_lo .equ r13
abs 0000000e 40 hours_hi .equ r14
abs 0000000f 41 sw_count .equ r15
000000000000000004 42 STATUS .equ 04h
000000000000000005 43 KILLSEC .equ 05h
000000000000000006 44 SECONDS .equ 06h
000000000000000007 45 MINUTES .equ 07h
000000000000000008 46 HOURS .equ 08h
000000000000000009 47 SECONDS LO .equ 09h
00000000000000000a 48 SECONDS HI .equ 0ah
00000000000000000b 49 MINUTES LO .equ 0bh
00000000000000000c 50 MINUTES HI .equ 0ch
00000000000000000d 51 HOURS LO .equ 0dh
00000000000000000e 52 HOURS HI .equ 0eh
000000000000000020 53 BUFFER .equ 20h
54;
55;
56;    STATUS REG:     d4 d3 d2 d1 d0  
57;    | | | | | displaying message
58;    | | | | | buffer cleared
58 ;
59 ;
60 ;
61 ;

00000000 62 .org 00
00000000 0000 63 .word 00
00000002 0000 64 .word 00
00000004 0000 65 .word 00
00000006 0000 66 .word 00
00000008 0000 67 .word time
0000000a 0000 68 .word shift
0000000c 69 .org 000c

70 ;********************************************************************;
71 ; Initialization
72 ;********************************************************************;

0000000c 3110 73 init: srp #WORK_REG ; disable int
0000000e 8f 74 di
00000010 e6f464 75 ld t0,#64 ; 100 decimal
00000012 e6f5c9 76 ld pre0,#1000001b ; set t0 for 5 mS period
00000015 e6f27d 77 ld t1,#125 ;
00000017 e6f313 78 ld prel,#00000011b ; set t1 for .5 mS period
0000001a e6f600 79 ld p2n,#0 ; outputs on p2
0000001d e6f701 80 ld p3n,#1 ; active on p2
00000020 e6f800 81 ld p0m,#0 ; outputs on p0
00000023 b000 82 clr p0 ; p0 low
00000026 b0ef 83 clr sv ; sv count= 0
00000029 b000 84 ld ipr,#0000000b ; make irq5 > irq3
0000002c b00f30 85 ld imr,#30 ; enable irq4,irq5
0000002e b00f80 86 ld spl,#80 ; set stack pointer
00000031 b010f 87 ld tmr,#0f ; load and enable counters
00000033 2c04 88 ld pointer,#04 ; point to time regs
00000036 bdc0 89 ld r15,#12 ; six locations
00000038 b1e2 90 clear_reg: clr @pointer ; clear ram
0000003a 2e 91 inc pointer ;
0000003b fafb 92 djmr r15,clear_reg ; continue until all zero
0000003d 0000 93 call clear_buffer ; clear buffer
00000040 e60800 94 ld HOURS,#12 ; start time at 12:00
00000043 0000 95 call load_time ; start loading time
00000046 2c20 96 ld pointer,#BUFFER ; start at top of buffer
00000048 0000 97 ld p0,#00000011b ; take data and clear hi
0000004a 9f 98 main_loop: ei ; enable interrupts
0000004c 8fd 99 jr main_loop ;

100 ;********************************************************************;
101 ; This interrupt routine updates the time
102 ;********************************************************************;

00000050 70fd 103 time: push rp ; save current req pointer
00000050 3100 104 srp #TIME_REG ; point to time req group
00000052 0000 105 call test_sw ; look at time->set switches
00000055 e6 106 inc millisec ; increment millisec req
00000056 a6e5c8 107 cp millisec,#200 ; one second?
00000059 0000 108 clr millisec ; set to zero
0000005c b0e5 109 jp ultr.exit_time ; exit if not
0000005e 0000 110 call test_sw ; exit if not
00000061 a6408 111 jr r,tim e;test_sw2 ; no
00000063 00701 112 add MINUTES,#1 ;
00000066 4007 113 da MINUTES ; convert to bcd
00000068 a60760 114 cp MINUTES,#60 ; sixty minutes?
0000006b 7b4a 115 jr ul t,inc_seconds ;
0000006e b007 116 clr MINUTES ;
APPENDIX (Continued)

000006f db**
0000071 760410
0000074 eb**
0000076 060801
0000079 4008
000007b e50813
000007e b**
0000080 e50801
0000083 06e501
0000086 a60410
0000089 4008
000008b a6013
000008e 7b**
0000090 e6010
0000093 06e601
0000095 06e701
0000098 a60401
00000a0 e50410
00000a3 e50410
00000a6 b**
00000a9 06e501
00000ab c0e7
00000ad e6010
00000b0 50fd
00000b2 bf
00000b4 06e401
00000b6 06e501
00000b8 06e601
00000ba 06e701
00000be 06e801
00000c0 06e901
00000c4 06e901
00000c8 e50e09
00000ca e50e0a
00000cb e50e09
00000d0 f00a
00000d2 56000f
00000d4 b00a
00000d6 e50e0f
00000d8 e50e0b
00000dc e4070b
00000e0 56000f
00000e4 e50e0c
00000e6 56000f
00000e8 56000f
00000ea 56000f
00000ed af
00000f0 50fd
00000f3 d00a
00000f4 56000f
00000f6 56000f
00000f8 56000f
00000fa 56000f
00000fc 56000f
00000fe 56000f
00000ff 56000f
0000100 db**
0000102 760410
0000105 760410
0000108 760410
000010b 760410
000010e 760410
0000111 db**
0000114 760410

0000118 118 jr inc_seconds
000011c 119 test_sw2: tm STATUS,#0000000b
0000120 120 jr t,inc_seconds
0000124 121 add HOURS,#1
0000127 122 da HOURS
000012a 123 cp HOURS,#113
000012d 124 jr ult,inc_seconds
000012f 125 ld HOURS,#1
0000132 126 inc_seconds: add seconds,#1
0000135 127 da seconds
0000138 128 cp seconds,#60
000013b 129 jr ult,exit_time
000013e 130 or STATUS,#00000001b
0000141 131 clr seconds
0000144 132 add minutes,#1
0000147 133 da minutes
000014a 134 cp minutes,#60
000014d 135 jr ult,exit_time
000014f 136 clr minutes
0000152 137 set_hrs: add hours,#1
0000155 138 da hours
0000158 139 cp hours,#113
000015b 140 jr ult,exit_time
000015e 141 ld hours,#1
0000161 142 exit_time: call time_convert
0000164 143 call load_time
0000167 144 pop rp
000016a 145 iret
000016d 146
000016f 147 ;******************************************************************************;
0000171 148 ; This routine converts the seconds, minutes, and hours bcd
0000174 149 ; data into units and tens-of-units for displaying
0000177 150 ;******************************************************************************;
0000179 151 time_convert: ld SECONDS Lo,SECONDS
000017d 152 1d SECONDS Hi,SECONDS
000017f 153 and SECONDS Lo,#10f
0000182 154 swap SECONDS Hi
0000185 155 and SECONDS Hi,#10f
0000188 156 1d MINUTES Lo,MINUTES
000018b 157 1d MINUTES Hi,MINUTES
000018f 158 and MINUTES Lo,#10f
0000192 159 swap MINUTES Hi
0000195 15a and MINUTES Hi,#10f
0000198 15b 1d HOURS Lo,HOURS
000019c 15c 1d HOURS Hi,HOURS
000019f 15d and HOURS Lo,#10f
00001a2 15e swap HOURS Hi
00001a5 15f and HOURS Hi,#10f
00001a8 160 ret
00001ab 161
00001ac 162
00001ad 163
00001ae 164
00001af 165
00001b0 166
00001b1 167
00001b2 168 ;******************************************************************************;
00001b4 169 ; This subroutine loads the time data into the RAM buffer
00001b7 170 ;******************************************************************************;
00001b9 171 load_time: push rp
00001bc 172 173 srp #WORK_REG
00001c0 174 175 ld r10,#d09
00001c3 176 177 ld r11,#BUFFER
00001c6 178 179 load_table: ld r12,#r10
00001c9 180 181 load_table: ld address_hi,#tb number_table; load hi address of table

11-70
 ;*****************************************************************************;
 ; This subroutine checks to see if the time-set switches are pressed. ;
 ;*****************************************************************************;
 211 test_sw: ld r10,p3 ; load sw data  
 212 cp r10,p3 ; l's complement  
 213 and r10,#03 ; mask off upper bits  
 214 tm p3,#1 ; min pressed?  
 215 jr t, test_hrs ; no  
 216 inc sw_count ; inc counter  
 217 cp sw_count,#2 ; debounced?  
 218 jr ult, exit_sw ; not yet  
 219 or STATUS,#00000100b ; set bit  
 220 jr exit_sw ; exit  
 221 test_hrs: tm p3,#4 ; hrs pressed?  
 222 jr t, clear_sw ; no  
 223 inc sw_count ; inc debounce counter  
 224 cp sw_count,#2 ; debounced?  
 225 jr ult, exit_sw ; not yet  
 226 or STATUS,#0000000b ; set bit  
 227 jr exit_sw ; exit  
 228 clear_sw: and STATUS,#1110011b ; reset sw status bits  
 229 clr sw_count ; reset debounce counter  
 230 ret ; return  
 231  
 232 ;*****************************************************************************;
 233 ; This is the timer interrupt routine. When T1 hits TC, column data is ;
 234 ; shifted one. ;
 235 ;*****************************************************************************;
 236 shift: push rp ; save reg pointer  
 237 srp #WORK_REC ; point to working reg
APPENDIX (Continued)

00000154 2e 238 inc pointer ; point to next location in ram
00000155 8e 239 inc col_count ;
00000156 a6e824 240 cp col_count,#36 ; thirty-six columns?
00000159 7b4# 241 jr ult,test_flag ; end of screen?
0000015b 0e8 242 clr col_count ; reset col number
0000015d 2c20 243 ld pointer,#BUFFER ; start at beginning of buffer
0000015f 460002 244 or po,#00000010b ; take data line high
00000162 760001 245 test_flag: tm STATUS,#0000001b ; time to scroll message?
00000165 61>2 246 jr r,continue : no, display time
00000167 248 247 call load_message ; load message data
0000016a bOe4 248 jr load_row ; load row data
0000016c d61iWv 250 load_row: ld temp_1,pointer ; load contents
0000016e d60401 251 com temp_1 ;
00000171 560001 252 ld p2,temp_1 ; out to port
00000173 5902 253 clock_it: or po,#00000100b ; take clock hi
00000175 460004 254 and po,#0000001b ; take clock and data low
00000178 560001 255 pop rp ; restore reg pointer
0000017b 50fd 256 int ;
0000017d bf 257 ;*******************************************************************************;
0000017e 760002 258 ; This routine loads the message into the buffer ;
00000181 6e# 259 ;*******************************************************************************;
00000183 d60400 260 load_message: tm STATUS,#00000010b ; clear buffer?
00000186 460002 261 jr nr,req_scroll ;
00000188 d60400 262 call clear_buffer ; clear contents of buffer
0000018a 660400 263 or STATUS,#00000010b ; set bit
0000018c 0c6 264 ld address_hi,#"buf mess_beg" ;
0000018e 1c6 265 ld address_lo,#"buf mess_beg" ;
00000190 9c06 266 call get_ascii ; get first character
00000192 e352 267 ld seq_count,#6 ; start out with six segs
00000195 6e600 268 jr load_next_seq ;
00000197 e6400 269 req_scroll: cp seq_count,#0 ; six segments loaded?
00000199 760404 270 jr ne,load_next_seq ; no
0000019c 6b# 271 tm STATUS,#00000100b ; end of message?
0000019e 460004 272 jr r,load_next_char ; load next ascii char
000001a1 6c00 273 ld temp_2,#0 ;
000001a4 d60400 274 call load_fifo ; load data
000001a6 460002 275 inc zero_count ; inc no of locations cleared
000001a8 a6e424 276 cp zero_count,#36 ; 36 locations yet?
000001a9 7b4# 277 jr ult,scroll_return ; load segment data
000001ba d60400 278 clr STATUS ; display time now
000001b1 0b04 279 clr zero_count ; clear counter
000001b3 0e8 280 jr scroll_return ; return
000001b6 9c06 281 load_next_char: ld seq_count,#6 ; load 6 segments/character
000001b9 a0e0 282 incw address ; point to next ascii char
000001bc d60400 283 call get_ascii ; get next character
000001c5 260 284 load_next_seq: ldc temp_2,rr12 ; load seq from ascii table
000001c8 d60400 285 call load_fifo ; shift the data
000001cb a6e424 286 incw rr12 ; point to next seq in table
000001cc 0e8 287 dec seq_count ; decrement segment count
000001cf 0e9 288 scroll_return: ret ;
000001f6 af 289 ;*******************************************************************************;
000001fe 0e8 290 ; This subroutine loads and shifts the RAM buffer ;
000001ff 0e8 291 ;*******************************************************************************;
00000200 fc24 292 load_fifo: ld r15,#36 ; load number of columns
00000203 2c20 293 call get_ascii ; get next character
00000206 a372 294 ld pointer,#BUFFER ; load starting buffer reg
00000209 660400 295 call load_fifo ; shift the data
0000020c f326 296 ld @pointer,temp_2 ; load new data
0000020f 68e7 297 ld temp_2,temp_3 ; transfer bytes
; This subroutine indexes the character table and fetches the segment data.
;*************************************************************;
00001ce ccc**
00001de dcc**
00001e2 c2aa
00001e4 a6000
00001e7 eb**
00001e9 460404
00001eb 8b**
00001ec 26ee20
00001ed a6000
00001ee aoec
00001ef aoec
00001f0 aoec
00001f2 eaf2
00001f4 af

;*************************************************************;
301 get_ascii: ld r12,#1h char_table ; load starting add of table
302 ld r13,#1b char_table ;
303 1dc r14,$err10 ; load ascii data
304 cp r14,#0 ; end of message?
305 jr ne,load_next ; no
306 or STATUS,$0000100b ; set bit to mark mess end
307 jr mess_return ; return
308
309 load_next: sub r14,$120 ; subtract 20h
310 ldc rl4,@rl0 ; load ascii data
311 cp rl4,#0 ; is it a space?
312 jr eq,lleSS _retlllll ; if yes, don't index table
313 index_table: incv rl2 ; index table
314
315
316
317
318
319
320
321

;*************************************************************;
322 mess_return: ret ;
323
324 ;*************************************************************;
325 clear_buffer: ld r15,#36 ; get no of columns
326 ld pointer,#BUFFER ; starting point of buffer
327 clear_loop: clr @pointer ; clear contents
328 incv pointer ; next location
329
djnz rl4,clear_loop ; till out of columns
330
331
332

;*************************************************************;
333 mess beg: .asciz 'THIS DISPLAY IS POWERED BY ILLOG!' .org 200h
334
335

;*************************************************************;
338 char_table: .byte 0,0,0,0,0,0,0,0 : space
339 .byte 0,0,0,0,0,0,0,0 : !
340 .byte 0,0,0,0,0,0,0,0 : "
341 .byte 0.1,4E,7F,14E,7F,14E : /
342 .byte 0.1,2E,2AE,7F,2AE,2AE ;
343 .byte 0.62E,64E,4AE,13E,23E ;
344 .byte 0.36E,49E,35E,02E,05E ;
345 .byte 0.0,0,0,0,0,0,0,0 ;
346 .byte 0.1CE,22E,41E,0,0 ;
347 .byte 0.1CE,22E,41E,0,0 ;
348 .byte 0.1CE,22E,41E,0,0 ;
349 .byte 0.1CE,22E,41E,0,0 ;
350 .byte 0.1CE,22E,41E,0,0 ;
351 .byte 0.1CE,22E,41E,0,0 ;
352 .byte 0.1CE,22E,41E,0,0 ;
353 .byte 0.1CE,22E,41E,0,0 ;
APPENDIX (Continued)

```assembly
00000225 00000041221c 354 .byte 0,0,0,41H,22H,1CH ; )
0000022b 0022147f1422 355 .byte 0,22H,14H,7FH,14H,22H ; *
0000022c 0008036e0f08 356 .byte 0,08H,08H,3EH,08H,08H ; +
00000227 000000060000 357 .byte 0,0,1,6,0,0 ; ,
0000022d 000808080808 358 .byte 0,8,8,8,8 ; -
0000022e 000000000000 359 .byte 0,0,0,1,0,0 ; ,
0000022f 000000408010 360 .byte 0,2,4,8,10H,20H ; /
361 :numbers
000002df 003e4549593e 362 number_table: .byte 0,3EH,45H,49H,51H,3EH ; 0
000002e5 00000027f010 363 .byte 0,0,21H,7FH,01H,0 ; 1
000002e6 002345494931 364 .byte 0,23H,45H,49H,49H,31H ; 2
000002e7 004241495966 365 .byte 0,42H,41H,49H,59H,66H ; 3
000002e8 00014247f004 366 .byte 0,0CH,14H,24H,7FH,04H ; 4
000002e9 00724515154e 367 .byte 0,72H,51H,51H,4EH ; 5
000002ea 001629494946 368 .byte 0,1EH,29H,49H,49H,46H ; 6
000002eb 004047485060 369 .byte 0,40H,47H,48H,50H,60H ; 7
000002ec 003649494936 370 .byte 0,36H,49H,49H,49H,36H ; 8
000002ed 003149494943c 371 .byte 0,31H,49H,49H,49H,3CH ; 9
372 : MORE SPECIAL CHARACTERS
0000031b 000000014000 373 .byte 0,0,0,14H,0,0 ; : :
00000321 000000116000 374 .byte 0,0,1,16H,0,0 ; ;
00000322 000814224100 375 .byte 0,8,14H,22H,41H,0 ; <
0000032d 001414141414 376 .byte 0,14H,14H,14H,14H ; =
00000333 000002214080 377 .byte 0,0,41H,22H,14H,0 ; >
00000339 000204040502 378 .byte 0,20H,40H,40H,50H,20H ; ?
379 : AT SIGN AND UPPERCASE LETTERS
0000033f 003e4549593e 380 .byte 0,3EH,45H,49H,51H,3EH ; @
00000345 001f2444441f 381 .byte 0,1FH,24H,44H,44H,1FH ; A
00000346 004f49494936 382 .byte 0,4FH,49H,49H,49H,36H ; B
00000351 003e41414122 383 .byte 0,3EH,41H,41H,41H,22H ; C
00000357 00714141413e 384 .byte 0,71H,41H,41H,41H,3EH ; D
0000035d 004f49494941 385 .byte 0,4FH,49H,49H,49H,41H ; E
00000363 004f48484840 386 .byte 0,4FH,48H,48H,48H,40H ; F
00000369 003e41414147 387 .byte 0,3EH,41H,41H,41H,47H ; G
0000036f 00708080807f 388 .byte 0,70H,08H,08H,08H,7FH ; H
00000375 0000017f4100 389 .byte 0,00H,41H,7FH,1FH,00H ; I
0000037b 00020101017e 390 .byte 0,02H,01H,01H,7EH ; J
00000381 007081422421 391 .byte 0,70H,08H,42H,24H,21H ; K
00000387 007001010101 392 .byte 0,70H,01H,01H,01H,01H ; L
0000038d 00702182027f 393 .byte 0,70H,21H,82H,27H,7FH ; M
00000393 00700084047f 394 .byte 0,70H,00H,84H,04H,7FH ; N
00000399 003e41414143e 395 .byte 0,3EH,41H,41H,41H,3EH ; O
0000039f 004f48484830 396 .byte 0,4FH,48H,48H,48H,30H ; P
000003a5 003e41454230 397 .byte 0,3EH,41H,45H,42H,30H ; Q
000003ab 004f44443a31 398 .byte 0,4FH,44H,44H,3A,31H ; R
000003b1 003249494926 399 .byte 0,32H,49H,49H,49H,26H ; S
000003b7 0040407f4f04 400 .byte 0,40H,40H,7FH,4F,04H ; T
000003bd 007e0010117e 401 .byte 0,7EH,01H,01H,11H,7EH ; U
000003ce 00702001027c 402 .byte 0,70H,02H,01H,02H,7CH ; V
000003cf 00702002c027f 403 .byte 0,7EH,02H,02C,02H,7FH ; W
000003d0 006314041463 404 .byte 0,63H,14H,04H,14H,63H ; X
000003d5 0060100f1060 405 .byte 0,60H,10H,0F,10H,60H ; Y
000003db 004345495161 406 .byte 0,43H,45H,49H,51H,61H ; Z
000003de 00774f411414 407 .byte 0,77H,4F,41H,14H,14H ; 
00000367 002010000402 408 .byte 0,20H,10H,04H,02H,02H ; \ 
0000036d 004141414141 409 .byte 0,41H,41H,41H,7FH,7FH ; ]
00000373 000401000000 410 .byte 0,04H,01H,00H,00H,00H ; ^
00000379 000101010101 411 .byte 0,01H,01H,01H,01H,01H ; _
0000037f 000101010101 412 .byte 0,01H,01H,01H,01H,01H ; 
        .end
```

INTERFACING LCDs TO THE Z8®

By trading hardware approaches for software solutions, interfacing a Z8 Microcontroller to a M1641 LCD module becomes a practical and simplified design methodology.

INTRODUCTION

There has been an increasing demand for interfacing Liquid Crystal Displays (LCDs) to low-end microcontrollers in recent years. Unfortunately, little has been offered to address real-world applications and to help the design engineer understand how to make LCDs work. This App Note (Application Note) explains and shows a software method of interfacing a Z8 to an LCD module. The challenge to the programmer is the fixed amount of ROM space. Although almost any Z8 device is usable, the CCP™ (Consumer Controller Processor) Family is referenced; For example, the Z86C40, Z86C96/61, etc.

OVERVIEW

Since the Z8 architecture is so flexible, it would be very difficult to include all possible applications in the spec sheet example Figures and Tables. Since the purpose of this App Note is to reduce the complexity of interfacing Z8s with LCDs, the software routines are intended to be "cut and paste," so choose the ones that meet your needs (caution, do not forget the initialization routines). Also, remember that not all possible functions inside the LCD were utilized. In many applications, it is only necessary to transfer ASCII for display in 16-character chunks. For messages containing more than 16 characters, the message may be broken down into two 16-character fields, each alternately being displayed. The details on special functions are in the manufacturers data sheets.

M1641 LCD MODULE

The M1641 LCD Module is a 1-line by 16-character display with an on-board controller; the HD44780. The HD44780 divides the 16 characters into two lines of eight characters each. Even though this is a two-line device, physically, all characters appear on the same line. The controller has an on-board character generator in ROM capable of displaying 192 ASCII characters, along with eight user programmable characters. All characters are displayed in a 5x7 font.
The LCD module can be connected to either an 8-bit or 4-bit data bus. There are three control lines: RS, R/W, and E (Enable). The RS line selects either an instruction (Low), or data (High). The R/W line (write active Low), allows data to be written to the LCD (Low) or read from the LCD (High). The Enable line (E) is used to latch data to and from the LCD (Figure 1a and 1b and Table 1a and 1b). The V1c line is used for adjusting the contrast, but for most applications may be tied to ground.

### Table 1a. Read Characteristics

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Standard</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable cycle time</td>
<td>$t_{\text{CYC}E}$</td>
<td>1000</td>
<td>ns</td>
</tr>
<tr>
<td>Enable pulse width</td>
<td>High level</td>
<td>$P_{\text{EH}}$</td>
<td>450 ns</td>
</tr>
<tr>
<td>Enable rise and fall time</td>
<td>$t_{\text{Er}}$, $t_{\text{Ef}}$</td>
<td>-</td>
<td>25 ns</td>
</tr>
<tr>
<td>Setup time</td>
<td>RS, R/W→E</td>
<td>$t_{\text{AS}}$</td>
<td>140 ns</td>
</tr>
<tr>
<td>Address hold time</td>
<td>$t_{\text{AH}}$</td>
<td>10 ns</td>
<td></td>
</tr>
<tr>
<td>Data delay time</td>
<td>$t_{\text{DOR}}$</td>
<td>-</td>
<td>320 ns</td>
</tr>
<tr>
<td>Data hold time</td>
<td>$t_{H}$</td>
<td>20 ns</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

*V_{dd}=5.0V \pm 5\%, V_{ss}=0V, T_{A}=0^\circ C \text{ to } 50^\circ C*
### Table 1b. Write Characteristics

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Standard</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable cycle time</td>
<td></td>
<td>t\textsubscript{Cyc}E</td>
<td>1000  ns</td>
</tr>
<tr>
<td>Enable pulse width</td>
<td></td>
<td>PW\textsubscript{EH}</td>
<td>450  ns</td>
</tr>
<tr>
<td>Enable rise and fall time</td>
<td></td>
<td>t\textsubscript{Ef}, t\textsubscript{Ei}</td>
<td>- 25  ns</td>
</tr>
<tr>
<td>Setup time</td>
<td></td>
<td>t\textsubscript{AS}</td>
<td>140  ns</td>
</tr>
<tr>
<td>Address hold time</td>
<td></td>
<td>t\textsubscript{AH}</td>
<td>10   ns</td>
</tr>
<tr>
<td>Data delay time</td>
<td></td>
<td>t\textsubscript{CDR}</td>
<td>195  ns</td>
</tr>
<tr>
<td>Data hold time</td>
<td></td>
<td>t\textsubscript{H}</td>
<td>10   ns</td>
</tr>
</tbody>
</table>

Note:

\[ V_{dd} = 5.0V \pm 5\%, \ V_{ss} = 0V, \ T_{a} = 0^\circ C \text{ to } 50^\circ C \]

### Figure 1b. LCD Write Timing

*Valid Data*
INTERFACE (Continued)

V_{ss} is tied to ground while V_{cc} is tied to the +5 Volt supply. Unless your application is heavily "I/O bound," it is easiest to use one of the Z8's 8-bit ports for data, and use two lines from Port 3 for control. Figure 2 shows a typical Z8 interface.

![Figure 2. Typical Z8/LCD Interface](image)

INITIALIZATION

After power up, initialize the LCD before sending data. Note that the LCD module is very slow. Therefore, it is necessary to write a delay loop program in between instructions. Again, the LCD can be configured for either 8-bit or 4-bit data transfer. When operating in 4-bit mode, the upper nibble gets transferred first, followed by the lower nibble.

Table 2 shows complete instruction codes. In order to write the instruction codes to the LCD module, the RS line must be Low. Figure 3 shows an initialization sequence for an 8-bit transfer operation. The starting address for the DD RAM is 80H for the first eight characters. For the next eight characters, the starting address is COH. If the DD RAM is programmed for auto increment, then the DD RAM address is automatically incremented after each character write.
### Table 2. LCD Instructions Codes

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Instruction Code</th>
<th>Description</th>
<th>Execution Time (when fCP or fOSC is 250 kHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear Display</td>
<td>0 0 0 0 0 0 0 0 1</td>
<td>Clears all display memory and returns the cursor to the home position (Address 0).</td>
<td>82 µs - 1.64 ms</td>
</tr>
<tr>
<td>Return Home</td>
<td>0 0 0 0 0 0 0 0 1</td>
<td>Returns the cursor to the home position (Address 0). Also returns the display being shifted to the original position. DD RAM contents remain unchanged.</td>
<td>40 µs - 1.6 ms</td>
</tr>
<tr>
<td>Entry Mode Set</td>
<td>0 0 0 0 0 0 0 1 1</td>
<td>Sets the cursor move direction and specifies or not to shift the display. These operations are performed during data write and read.</td>
<td>40 µs - 1.64 ms</td>
</tr>
<tr>
<td>Display ON/OFF Control</td>
<td>0 0 0 0 0 0 1 0 0</td>
<td>(D) is display ON/OFF control, memory remains unchanged in OFF condition. (C) cursor ON/OFF (B) blinking cursor.</td>
<td>40 µs</td>
</tr>
<tr>
<td>Cursor or Display Shift</td>
<td>0 0 0 0 0 0 1 0 0</td>
<td>Moves the cursor and shifts the display without changing DD RAM contents.</td>
<td>40 µs</td>
</tr>
<tr>
<td>Function Set</td>
<td>0 0 0 0 0 0 1 0 0</td>
<td>Sets interface data length (DL), number of display lines (N), and character font (F).</td>
<td>40 µs</td>
</tr>
<tr>
<td>Set CG RAM Address</td>
<td>0 0 0 0 0 0 0 1</td>
<td>Sets the CG RAM address. CG RAM data is sent and received after this setting.</td>
<td>40 µs</td>
</tr>
<tr>
<td>Set DD RAM Address</td>
<td>0 0 0 0 0 0 0 0 1</td>
<td>Sets the DD RAM address. DD RAM data is sent and received after this setting.</td>
<td>40 µs</td>
</tr>
<tr>
<td>Read Busy Flag &amp; Address</td>
<td>0 0 0 0 0 0 1 0 0</td>
<td>Reads Busy Flag (BF) indicating internal operation is being performed and reads address counter contents.</td>
<td>1 µs</td>
</tr>
<tr>
<td>Write Data to CG or DD RAM</td>
<td>0 0 0 0 0 0 0 1</td>
<td>Writes into DD RAM or CG RAM.</td>
<td>40 µs</td>
</tr>
<tr>
<td>Read Data from CG or DD RAM</td>
<td>0 0 0 0 0 0 0 1</td>
<td>Reads data from DD RAM or CG RAM.</td>
<td>40 µs</td>
</tr>
</tbody>
</table>
INITIALIZATION (Continued)

Notes to the previous table:

1. *Doesn't Matter

2. DD RAM: Display data RAM
   CG RAM: Character generator RAM
   A₀₀₅: CG RAM address
   A₀₀₆: DD RAM address
   Corresponds to cursor address
   AC: Address counter used for both of DD and CG RAM

<table>
<thead>
<tr>
<th></th>
<th>DD RAM</th>
<th>CG RAM</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/D</td>
<td>I/D=1: Increment I/D=0: Decrement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>C=1: Cursor On C=0: Cursor Off</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R/L</td>
<td>R/L=1: Right shift R/L=0: Left shift</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. Execution times in Table 2 indicate the maximum values when operating frequency is 250 kHz.

4. When \( f_{osc} \) is 270 kHz: \( 40 \mu s \times \frac{250}{270} = 37 \mu s \)
Display Initialization

Each time the module is turned on or reset, an initialization procedure must be executed. The procedure consists of sending a sequence of hex codes from the microprocessor or parallel I/O port. The initialization sequence turns on the cursor, clears the display, and sets the module onto an auto-increment mode.

The initial hex code 30, 34, or 38 is sent two or more times to ensure the module enters the 8-bit or 4-bit data mode. All the initialization sequences are performed under the condition of Register Select (RS) = 0 (Low) and Read/Write (R/W) = 0 (Low).

4-bit data bus microcontroller may operate the display module by sending the initialization sequence in 4-bit format. Since 4-bit operation requires the data to be sent twice over the higher 4-bit bus lines (D4-D7), memory requirements are doubled.

Example for the module with 5x7 character format under 8-bit data transfer.


---

**Notes:**
1. Both RS and R/W terminals shall be "0" in this sequence.
2. RS, RW and Data are latched at the falling edge of the Enable signal, (falling edge is typically 10 ns; MAX: 25 ns).
3. M4024 has to be initialized on E1 and E2, respectively.
   Refer to (2) in Sample Programs.
Appendix A (Listing 1) has all the routines for communicating to an LCD module. For this App Note exercise, there is no cursor, auto-increment for the DD RAM address, and no display shift. Since there is no read instruction from the LCD, the R/W is tied Low.

Transferring ASCII to the Z8 via the UART requires buffering before writing to the LCD, since the LCD is not able to keep up with the Z8 at high baud rates. Therefore, the 16 ASCII are first stored within one of the working register groups, then transferred to the LCD. Actually, the Z8 has the ability to store several messages within the register file. This is useful for applications that require a handful of commonly used messages that are downloaded from a master MPU after initialization.

Of course, messages can be stored in the internal Z8 ROM space. Since the program occupies less than 750 bytes, there is plenty of room to display other custom messages.

This program allows the Z8 user to interface to a 1-line by 16-character Liquid Crystal Display (LCD). It includes a routine to load serial data from the UART to the register file, then out to the display. For displaying "canned" messages send a control character followed by the message number. Another control character is used for accepting ASCII from the outside world followed by sixteen ASCII characters.

This program is intended to be used with the industry-standard 1-line by sixteen-character LCD module (Seiko M1641, etc.). Certain modifications are necessary for 2-line modules (see manufacturers data sheets).
APPENDIX A

LCD Program
(Listing 1)

: M1641 Pinout
:     -------------------
:     1 - Vss
:     2 - Vdd
:     3 - V1c
:     4 - RS
:     5 - R/W
:     6 - E
:     7 - DB0
:     8 - DB1
:     9 - DB2
:    10 - DB3
:    11 - DB4
:    12 - DB5
:    13 - DB6
:    14 - DB7

CUR_HOME .EQU 02H
DIS_CLEAR .EQU 01H
CG_RAM .EQU 40H
DD_RAM_1 .EQU 80H
DD_RAM_2 .EQU 08H
PORT_2 .EQU 02H
PORT_0 .EQU 00H
CONST_1 .EQU 1FH
CONST_2 .EQU OFFH
PDLL_SID .EQU 0BH
IRQ3_RES .EQU OF7H
STAC < I .EQU 0BH
CLDCK_LO .EQU 0FEH
CLDCK_HI .EQU 01H
RS_LOW .EQU 0FDH
RS_HIGH .EQU 02H
E_LOW .EQU 0FBH
E_HIGH .EQU 04H
MYREG_O .EQU 00H
E_BITS .EQU 38H
AI_NS .EQU 06H
DO_NC .EQU 0CH
BUFF_BEG .EQU 10H
BUFF_END .EQU 20H
MASK .EQU 01H

MYREG2 .EQU 20H

: .ORG 0000H
: .WORD 0,0,0
: .word RXD
: .word 0
: .word 0
: .org 000ch
APPENDIX A (Continued)

INTERFACEN LCDs TO THE Z8 APPLICATION NOTE

INITIALIZATION

BEGIN:

; DISABLE INTERRUPTS
DI #MYREG_0
; POINT TO REGS 10 - 1F HEX
SRP #STACK
; INITIALIZE STACK POINTER
LD PO1M,#00
; CONFIGURE P0 FOR OUTPUT
LD P2M,#0
; CONFIGURE P2 FOR OUTPUT
LD P3M,#%41
; ACTIVE PULL-UPS FOR P2
LD IPR,#%28
; IRQ3 IS HIGHEST
LD IRQ,#%88
; ENABLE IRQ3
CALL LCD_INIT
; INITIALIZE THE LCD
EI
; ENABLE INTERRUPTS

ENABLE:
;
JR ENABLE
; WAIT FOR INTERRUPTS

------------------------------------------------------------------------

INITIALIZE THE LIQUID CRYSTAL DISPLAY

------------------------------------------------------------------------

LCD_INIT:

AND PORT_0,#RS_LOW
LD PORT_2,#%00
OR PORT_0,#E_HIGH
AND PORT_0,#E_LOW
CALL WAIT_1
LD PORT_2,#8_BITS
OR PORT_0,#E_HIGH
AND PORT_0,#E_LOW
CALL WAIT_1
OR PORT_0,#E_HIGH
AND PORT_0,#E_LOW
CALL WAIT_1
OR PORT_0,#E_HIGH
AND PORT_0,#E_LOW
CALL WAIT_1
OR PORT_0,#E_HIGH
AND PORT_0,#E_LOW
CALL WAIT_1
LD PORT_2,#DO_NC
OR PORT_0,#E_HIGH
AND PORT_0,#E_LOW
CALL WAIT_1
LD PORT_2,#AI_NS
OR PORT_0,#E_HIGH
AND PORT_0,#E_LOW
CALL WAIT_1

; MAKE SURE P3-4 IS 0
; MAKE P2 ZERO
; TAKE ENABLE HIGH
; BRING ENABLE BACK LOW
; WAIT AWHILE
; SET DATA LENGTH FOR 8 BITS
; TAKE ENABLE HIGH
; BRING ENABLE BACK LOW
; WAIT AWHILE
; TAKE ENABLE HIGH
; BRING ENABLE BACK LOW
; WAIT AWHILE
; TAKE ENABLE HIGH
; BRING ENABLE BACK LOW
; WAIT AWHILE
; TURN ON DISPLAY, NO CURSOR!
; TAKE ENABLE HIGH
; BRING ENABLE BACK LOW
; WAIT AWHILE
; ENTRY MODE: INC ADDRESS, NO SHIFT
; TAKE ENABLE HIGH
; BRING ENABLE BACK LOW
; WAIT AWHILE
LCD_RES:

LD PORT_2,#DIS_CLEAR ; CLEAR DISPLAY
OR PORT_0,#E_HIGH ; TAKE ENABLE HIGH
AND PORT_0,#E_LOW ; BRING ENABLE BACK LOW
CALL WAIT_1 ; WAIT AWHILE
LD PORT_2,#CUR_HOME ; CURSOR HOME
OR PORT_0,#E_HIGH ; TAKE ENABLE HIGH
AND PORT_0,#E_LOW ; BRING ENABLE BACK LOW
CALL WAIT_1 ; WAIT AWHILE
LD PORT_2,#CG_RAM ; SET CG RAM
OR PORT_0,#E_HIGH ; TAKE ENABLE HIGH
AND PORT_0,#E_LOW ; BRING ENABLE BACK LOW
CALL WAIT_2 ; WAIT AWHILE
LD PORT_2,#DD_RAM_1 ; SET DD RAM
OR PORT_0,#E_HIGH ; TAKE E HIGH
AND PORT_0,#E_LOW ; TAKE E LOW
CALL WAIT_2 ; WAIT AWHILE
RET

---------

RECEIVE INTERRUPT ROUTINE - RECEPTION OF 1B HEX IS FOLLOWED
BY SIXTEEN ASCII CHARACTERS TO BE DISPLAYED ON LCD MODULE.
RECEPTION OF 1C HEX, FOLLOWED BY A MESSAGE NUMBER (e.g. 0-9),
DISPLAYS ONE OF THE INTERNAL MESSAGES.
---------

RXD:

DI ; DISABLE INTERRUPTS
PUSH RP ; SAVE REG POINTER
SRP #MYREG2 ; POINT TO WORKING REGS
LD R7,SIO ; GET BYTE FROM SIO
CF R7,#%1B ; ESCAPE CHARACTER?
JR NE,FS ; NO, TRY FS CHARACTER
CALL ASCII ; GET READY FOR TEXT
JR RXDOUT ; EXIT

FS:

CF R7,#%1C ; IF FS, THEN CANNED MESSAGE
JR NE,RXDOUT ; IF NOT, EXIT
CALL INT_MSG ; CALL CANNED MESSAGE ROUTINE

RXDOUT:

POP RP ; RESTORE POINTER
IRET ; RETURN FROM INTERRUPT

---------

ACCESS ONE OF THE INTERNAL LCD MESSAGES
---------

INT_MSG:

PUSH RP ; SAVE CURRENT REG POINTER
SRP #MYREG_0 ; POINT TO REG 0 - FH
PUSH IMR ; SAVE CURRENT IMR
DI ; DISABLE INTERRUPTS
LD IMR,#%20 ; ENABLE IRQ5 ONLY
CLR IRQ ; CLEAR ANY PENDING
EI ; ENABLE INTERRUPTS
APPENDIX A (Continued)

MSG_NUM:

```
TM IRQ,#POLL_SIO ; POLL SIO FOR NEXT BYTE
JR Z,MSG_NUM ; KEEP POLLING
AND IRQ,#IRQ3_RES ; CLEAR IRQ3
LD R4,SIO ; LOAD BYTE FROM SIO
LD R8,^HB_LCD_MSG ; LOAD LOOKUP TABLE ADD
LD R9,^LB_LCD_MSG ;
LD R7,#BUFF_BEG ; GET BUFFER START ADD
```

LOOKUP_1:

```
LD R6,#16 ; SET BYTE COUNTER FOR 16
```

LOOKUP_2:

```
INCW RR8 ; STEP TO DESIRED MESSAGE
DJNZ R6,LOOKUP_2 ; IF NOT 0, KEEP DECREMENTING
DJNZ R4,LOOKUP_1 ; INDEX MESSAGES
```

LOAD_MSG:

```
LDEI @R7,@RR8 ; LOAD INT BUFFER, INCREMENT
CP R7,#BUFF_END ; END OF BUFFER?
JR LT,LOAD_MSG ; NO, KEEP LOADING
CALL LCD_RES ; SETUP FOR LCD TRANSFER
CALL LCD_LOAD ; GO TO LOAD ROUTINE
POP IMR ; RESTORE INTERRUPT STRUCTURE
POP RP ; RESTORE REG POINTER
RET ; RETURN TO CALLER
```

---------------------------------------------------------------------
LOAD ASCII CHARACTERS FOR DISPLAY ON LCD MODULE
---------------------------------------------------------------------

ASCII:

```
PUSH IMR ; SAVE CONTENTS OF IMR
DI ; DISABLE INTERRUPTS
LD IMR,#%20 ; ENABLE IRQ 5 ONLY
CLR IRQ ; CLEAR IRQ
EI ; ENABLE INTERRUPTS
CALL TXT_LOAD ; LOAD TEXT INTO BUFFER
CALL LCD_RES ; CLEAR THE LCD
POP IMR ; RESTORE IMR
RET ; RETURN FROM INTERRUPT
```

---------------------------------------------------------------------
TAKE THE ASCII TEXT AND LOAD IT INTO THE BUFFER
---------------------------------------------------------------------

TXT_LOAD:

```
CLR R6 ; RESET BYTE COUNTER
LD R7,#BUFF_BEG ; POINT TO ASCII BUFFER
```

LOAD:

```
TM IRQ,#POLL_SIO ; BYTE IN SIO?
JR Z,LOAD ; LOAD SOME MORE
LD @R7,SIO ; STORE AT THIS BUFFER LOCATION
AND IRQ,#IRQ3_RES ; RESET IRQ3
INC R7 ; INC BUFFER ADDRESS
INC R6 ; INC BYTE COUNTER
CP R6,#16 ; SIXTEEN BYTES YET?
JR LT,LOAD ; NO, KEEP GOING
RET ; RETURN TO CALLER
```
INTERFACING LCDs TO THE '1J9

APPLICATION NOTE

TAKE THE ASCII FROM THE BUFFER AND LOAD THE LCD

-----------------------------

LCD_LOAD:
CALL LCD_OUT ; OUTPUT CONTENTS OF BUFFER TO LCD
AND PORT_0,#RS_LOW ; TAKE RS LOW
LD PORT_2,#DD_RAM_2 ; LOAD STARTING ADDRESS FOR LINE 2
OR PORT_0,#E_HIGH ; TAKE ENABLE HIGH
AND PORT_0,#E_LOW ; TAKE ENABLE LOW
CALL LCD_OUT ; OUTPUT TO LCD
RET

LCD_OUT:
OR PORT_0,#RS_HIGH ; TAKE RS HIGH
CLR R6 ; RESET BYTE COUNTER

LCD_LOOP:
LD PORT_2,#R7 ; FETCH ASCII FROM BUFFER
OR PORT_0,#E_HIGH ; TAKE ENABLE HIGH
AND PORT_0,#E_LOW ; TAKE ENABLE LOW
CALL WAIT_2 ; WAIT AWHILE
INC R6 ; INC BYTE COUNTER
INC R7 ; INC BUFFER ADDRESS
CF R6,#8 ; EIGHT BYTES YET?
JR LT, LCD_LOOP ; NO, GO AGAIN
RET ; RETURN

-----------------------------

DELAY LOOP FOR LCD WRITES

-----------------------------

WAIT_2:
LD R4,#CONST_1 ; THIS DELAY NOT

BUSY:
DJNZ R5,BUSY ; QUITE AS LONG
RET ; RETURN TO CALLER

-----------------------------

DELAY LOOP FOR LCD INITIALIZATION

-----------------------------

WAIT_1:
PUSH RP ; SAVE RP
SRP #MYREG_0 ; POINT TO OOH
LD R4,#CONST_1 ; LOAD FOR DELAY

LCDELY_1:
LD R5,#CONST_2 ; LCD'S ARE SLOW!

LCDELY_2:
DJNZ R5,LCDELY_2 ; DONE YET?
DJNZ R4,LCDELY_1 ;
POP RP ; RESTORE RP
RET ; RETURN TO CALLER

-----------------------------

INTERNAL LCD MESSAGES

-----------------------------

LCD_MSG:
.BLOCK 16
.ASCII 'ZILOG Z8s ARE OK'
.END
INTRODUCTION

This Application Note (App Note) is written for designers using Zilog Integrated Circuits with on-chip oscillators; circuits in which the amplifier portion of a feedback oscillator is contained on the IC. This App Note covers common theory of oscillators, and requirements of the circuitry (both internal and external to the IC) which comes from the theory for crystal and ceramic resonator based circuits.

Purpose and Benefits

The purposes and benefits of this App Note include:

1. Providing designers with greater understanding of how oscillators work and how to design them to avoid problems.
2. To eliminate field failures and other complications resulting from an unawareness of critical on-chip oscillator design constraints and requirements.

Problem Background

Inadequate understanding of the theory and practice of oscillator circuit design, especially concerning oscillator start-up, has resulted in an unreliable design and subsequent field problems (See on page 10 for reference materials and acknowledgments).

OSCILLATOR THEORY OF OPERATION

The circuit under discussion is called the Pierce Oscillator (Figures 1, 2). The configuration used is in all Zilog on-chip oscillators. Advantages of this circuit are low power consumption, low cost, large output signal, low power level in the crystal, stability with respect to $V_{cc}$ and temperature, and low impedances (not disturbed by stray effects). One drawback is the need for high gain in the amplifier to compensate for feedback path losses.

![Figure 1. Basic Circuit and Loop Gain](image1)

![Figure 2. Zilog Pierce Oscillator](image2)
Pierce Oscillator (Feedback Type)

The basic circuit and loop gain is shown in Figure 1. The concept is straightforward; gain of the amplifier is $A = V_o/V_i$. The gain of the passive feedback element is $B = V_i/V_o$. Combining these equations gives the equality $AB = 1$. Therefore, the total gain around the loop is unity. Also, since the gain factors $A$ and $B$ are complex numbers, they have phase characteristics. It is clear that the total phase shift around the loop is forced to zero (i.e., 360 degrees), since $V_{in}$ must be in phase with itself. In this circuit, the amplifier ideally provides 180 degrees of phase shift (since it is an inverter). Hence, the feedback element is forced to provide the other 180 degrees of phase shift.

Additionally, these gain and phase characteristics of both the amplifier and the feedback element vary with frequency. Thus, the above relationships must apply at the frequency of interest. Also, in this circuit the amplifier is an active element and the feedback element is passive. Thus, by definition, the gain of the amplifier at frequency must be greater than unity, if the loop gain is to be unity.

The described oscillator amplifies its own noise at start-up until it settles at the frequency which satisfies the gain/phase requirement $AB = 1$. This means loop gain equals one, and loop phase equals zero (360 degrees). To do this, the loop gain at points around the frequency of oscillation must be greater than one. This achieves an average loop gain of one at the operating frequency.

The amplifier portion of the oscillator provides gain $> 1$ plus 180 degrees of phase shift. The feedback element provides the additional 180 degrees of phase shift without attenuating the loop gain to $< 1$. To do this the feedback element is inductive, i.e., it must have a positive reactance at the frequency of operation. The feedback elements discussed are quartz crystals and ceramic resonators.

Quartz Crystals

A quartz crystal is a piezoelectric device; one which transforms electrical energy to mechanical energy and vice versa. The transformation occurs at the resonant frequency of the crystal. This happens when the applied AC electric field is sympathetic in frequency with the mechanical resonance of the slice of quartz. Since this characteristic can be made very accurate, quartz crystals are normally used where frequency stability is critical. Typical frequency tolerance is .005 to 0.3%.

The advantage of a quartz crystal in this application is its wide range of positive reactance values (i.e., it looks inductive) over a narrow range of frequencies (Figure 3).
However, there are several ranges of frequencies where the reactance is positive; these are the fundamental (desired frequency of operation), and the third and fifth mechanical overtones (approximately 3 and 5 times the fundamental frequency). Since the desired frequency range in this application is always the fundamental, the overtones must be suppressed. This is done by reducing the loop gain at these frequencies. Usually, the amplifier's gain roll off, in combination with the crystal parasitics and load capacitors, is sufficient to reduce gain and prevent oscillation at the overtone frequencies.

The following parameters are for an equivalent circuit of a quartz crystal (Figure 4):

\[ \begin{align*}
L & \text{ - motional inductance (typ 120 mH @ 4 MHz)} \\
C & \text{ - motional capacitance (typ .01 pf @ 4 MHz)} \\
R & \text{ - motional resistance (typ 36 ohm @ 4 MHz)} \\
Cs & \text{ - shunt capacitance resulting from the sum of the capacitor formed by the electrodes (with the quartz as a dielectric) and the parasitics of the contact wires and holder (typ 3 pf @ 4 MHz).}
\end{align*} \]

The series resonant frequency is given by:

\[ F_s = \frac{1}{2\pi \sqrt{LC}}, \]
where \( X_c \) and \( X_l \) are equal.

Thus, they cancel each other and the crystal is then \( R \) shunted by \( Cs \) with zero phase shift.

The parallel resonant frequency is given by:

\[ F_p = \frac{1}{2\pi \sqrt{L \left( C \frac{C_t}{C+C_t} \right)}}, \]
where: \( C_t = C_s + C \)

**Series vs. Parallel Resonance.** There is very little difference between series and parallel resonance frequencies (Figure 3). A series resonant crystal (operating at zero phase shift) is desired for non-inverting amplifiers. A parallel resonant crystal (operating at or near 180 degrees of phase shift) is desired for inverting amps. Figure 3 shows that the difference between these two operating modes is small. Actually, all crystals have operating points in both serial and parallel modes. A series resonant circuit will NOT have load caps \( C1 \) and \( C2 \). A data sheet for a crystal designed for series operation does not have a load cap spec. A parallel resonant crystal data sheet specifies a load cap value which is the series combination of \( C1 \) and \( C2 \). For this App Note discussion, since all the circuits of interest are inverting amplifier based, only the parallel mode of operation is considered.
OSCILLATOR THEORY OF OPERATION

Ceramic Resonators

Ceramic resonators are similar to quartz crystals, but are used where frequency stability is less critical and low cost is desired. They operate on the same basic principle as quartz crystals as they are piezoelectric devices and have a similar equivalent circuit. The frequency tolerance is wider (0.3 to 3%), but the ceramic costs less than quartz.

Typical values of parameters are $L = 0.092 \text{ mH}$, $C = 4.6 \text{ pf}$, $R = 7 \text{ ohms}$ and $C_s = 40 \text{ pf}$, all at 8 MHz. Generally, ceramic resonators tend to start up faster but have looser frequency tolerance than quartz. This means that external circuit parameters are more critical with resonators.

Figure 5 shows reactance vs. frequency and Figure 6 shows the equivalent circuit.

Figure 5. Ceramic Resonator Reactance
Load Capacitors

The effects/purposes of the load caps are:

Cap C2 combined with the amp output resistance provides a small phase shift. It also provides some attenuation of overtones.

Cap C1 combined with the crystal resistance provides additional phase shift.

These two phase shifts place the crystal in the parallel resonant region of Figure 3.

Crystal manufacturers specify a load capacitance number. This number is the load seen by the crystal which is the series combination of C1 and C2, including all parasitics (PCB and holder). This load is specified for crystals meant to be used in a parallel resonant configuration. The effect on start-up time; if C1 and C2 increase, start-up time increases to the point at which the oscillator will not start. Hence, for fast and reliable start-up, over manufacture of large quantities, the load caps should be sized as low as possible without resulting in overtone operation.

Amplifier Characteristics

The following text discusses open loop gain vs. frequency, open loop phase vs. frequency, and internal bias.

Open Loop Gain vs. Frequency over lot, VCC, Process Split, and Temp. Closed loop gain must be adequate to start the oscillator and keep it running at the desired frequency. This means that the amplifier open loop gain must be equal to one plus the gain required to overcome the losses in the feedback path, across the frequency band and up to the frequency of operation. This is over full process, lot, Vcc, and temperature ranges. Therefore, measuring the open loop gain is not sufficient; the losses in the feedback path (crystal and load caps) must be factored in.

Open Loop Phase vs. Frequency. Amplifier phase shift at and near the frequency of interest must be 180 degrees plus some, minus zero. The parallel configuration allows for some phase delay in the amplifier. The crystal adjusts to this by moving slightly down the reactance curve (Figure 3).

Internal Bias. Internal to the IC, there is a resistor placed from output to input of the amplifier. The purpose of this feedback is to bias the amplifier in its linear region and to provide the start-up transition. Typical values are 1M to 20M ohms.
PRACTICE: CIRCUIT ELEMENT AND LAYOUT CONSIDERATIONS

The discussion now applies prior theory to the practical application.

**Amplifier and Feedback Resistor**

The elements of the circuit, internal to the IC, include the amplifier, feedback resistor, and output resistance. The amplifier is modeled as a transconductance amplifier with a gain specified as $I_{out}/V_{in}$ (amps per volt).

**Transconductance/Gain.** The loop gain $A_B = g_m \times Z_1$, where $g_m$ is amplifier transconductance (gain) in amps/volt and $Z_1$ is the load seen by the output. $A_B$ must be greater than unity at and about the frequency of operation to sustain oscillation.

**Gain Measurement Circuit.** The gain of the amplifier can be measured using the circuits of Figures 6 & 7. This may be necessary to verify adequate gain at the frequency of interest and in determining design margin.

**Gain Requirement vs. Temperature, Frequency and Supply Voltage.** The gain to start and sustain oscillation (Figure 8) must comply with:

$$g_m > 4\pi^2 f^2 R_q C_{in} C_{out} \times M$$

where: $M$ is a quartz form factor = $(1 + C_{out}/C_{in} + C_{out}/C_{out})^2$

**Output Impedance.** The output impedance limits power to the XTAL and provides small phase shift with load cap $C_2$.

**DC Bias**

1. $V_{in}$
2. $V_{out}$
3. $V_{b}$

$$I_{out} = (V_{out} - V_{b})/33$$

* Figure 7. Transconductance (gm) Measurement

** Figure 8. Quartz Oscillator Configuration

* Inside chip, feedback resistor biases the amplifier in the high gm region.

** External components typically: $C_{in} = C_{out} = 30$ to $50$ pf (add 10 pf pin cap).
Load Capacitors

In the selection of load caps it is understood that parasitics are always included.

Upper Limits. If the load caps are too large, the oscillator will not start because the loop gain is too low at the operating frequency. This is due to the impedance of the load capacitors. Larger load caps produce a longer start-up.

Lower Limits. If the load caps are too small, either the oscillator will not start (due to inadequate phase shift around the loop), or it will run at a 3rd, 5th, or 7th overtone frequency (due to inadequate suppression of higher overtones).

Capacitor Type and Tolerance. Ceramic caps of ±10% tolerance should be adequate for most applications.

Ceramic vs. Quartz. Manufacturers of ceramic resonators generally specify larger load cap values than quartz crystals. Quartz C is typically 15 to 30 pf and ceramic typically 100 pf.

Summary. For reliable and fast start-up, capacitors should be as small as possible without resulting in overtone operation. The selection of these capacitors is critical and all of the factors covered in this note should be considered.

Feedback Element

The following text describes the specific parameters of a typical crystal:

Drive Level. There is no problem at frequencies greater than 1 MHz and V \text{cc} = 5V since high frequency AT cut crystals are designed for relatively high drive levels (5-10 mw max).

A typical calculation for the approximate power dissipated in a crystal is:

\[ P = 2R \left( \pi f C V_{\text{cc}} \right)^2 \]

Where. R = crystal resistance of 40 ohms, C = C1 + C0 = 20 pf. The calculation gives a power dissipation of 2 mW at 16 MHz.

Series Resistance. Lower series resistance gives better performance but costs more. Higher R results in more power dissipation and longer start-up, but can be compensated by reduced C1 and C2. This value ranges from 200 ohms at 1 MHz down to 15 ohms at 20 MHz.

Frequency. The frequency of oscillation in parallel resonant circuits is mostly determined by the crystal (99.5%). The external components have a negligible effect (0.5%) on frequency. The external components (C1,C2) and layout are chosen primarily for good start-up and reliability reasons.

Frequency Tolerance (initial temperature and aging). Initial tolerance is typically ±0.01%. Temperature tolerance is typically ±0.005% over the temp range (-30 to +100 degrees C). Aging tolerance is also given, typically ±.005%.

Holder. Typical holder part numbers are HC6, 18, 25, 33, 44.

Shunt Capacitance. (Cs) typically <7 pf.

Mode. Typically the mode (fundamental, 3rd or 5th overtone) is specified as well as the loading configuration (series vs. parallel).

The ceramic resonator equivalent circuit is the same as shown in Figure 4. The values differ from those specified in the theory section. Note that the ratio of L/C is much lower than with quartz crystals. This gives a lower Q which allows a faster start-up and looser frequency tolerance (typically ±0.9% over time and temperature) than quartz.

Layout

The following text explains trace layout as it affects the various stray capacitance parameters (Figure 9).

Traces and Placement. Traces connecting crystal, caps, and the IC oscillator pins should be as short and wide as possible (this helps reduce parasitic inductance and resistance). Therefore, the components (caps and crystal) should be placed as close to the oscillator pins of the IC as possible.

Grounding/Guarding. The traces from the oscillator pins of the IC should be guarded from all other traces (clock, V \text{cc}, address/data lines) to reduce crosstalk. This is usually accomplished by keeping other traces away from the oscillator circuit and by placing a ground ring around the traces/components (Figure 9).

Measurement and Observation

Connection of a scope to either of the circuit nodes is likely to affect operation because the scope adds 3-30 pf of capacitance and 1M-10Mohms of resistance to the circuit.
Indications of an Unreliable Design

There are two major indicators which are used in working designs to determine their reliability over full lot and temperature variations. They are:

Start Up Time. If start up time is excessive, or varies widely from unit to unit, there is probably a gain problem. C1/C2 needs to be reduced; the amplifier gain is not adequate at frequency, or crystal Rs is too large.

Output Level. The signal at the amplifier output should swing from ground to \( V_{cc} \). This indicates there is adequate gain in the amplifier. As the oscillator starts up, the signal amplitude grows until clipping occurs, at which point, the loop gain is effectively reduced to unity and constant oscillation is achieved. A signal of less than 2.5 Vp-p is an indication that low gain may be a problem. Either C1/C2 should be made smaller or a low R crystal should be used.

Figure 9. Circuit Board Design Rules
SUMMARY

Understanding the Theory of Operation of oscillators, combined with practical applications, should give designers enough information to design reliable oscillator circuits. Proper selection of crystals and load capacitors, along with good layout practices, results in a cost effective, trouble free design. Reference the following text for Zilog products with on-chip oscillators and their general/specific requirements.

ZILOG PRODUCT USING ON-CHIP OSCILLATORS

Zilog products that have on-chip oscillators:

Z8® Family: All
Z80®: C01, C11, C13, C15, C50, C90, 180, 181, 280
Z8000®: 8581
Communications Products: SCC™, ISCC™, ESCC™

ZILOG CHIP PARAMETERS

The following are some recommendations on values/parameters of components for use with Zilog on-chip oscillators. These are only recommendations; no guarantees are made by performance of components outside of Zilog ICs. Finally, the values/parameters chosen depend on the application. This App Note is meant as a guideline to making these decisions. Selection of optimal components is always a function of desired cost/performance trade-offs.

Note: All load capacitance specs include stray capacitance.

Z8® Family

General Requirements:
Crystal Cut: AT cut, parallel resonant, fundamental mode.
Crystal Co: < 7 pf for all frequencies.
Crystal Rs: < 100 ohms for all frequencies.
Load Capacitance: 10 to 22 pf, 15 pf typical.

Specific Requirements:
8604: xtal or ceramic, f = 1 - 8 MHz.
8600/10: f = 8 MHz.
8601/03/11/13: f = 12.5 MHz.
8602: xtal or ceramic, f = 4 MHz.
8680/81/82/84/91: f = 8, 12, 16, 16 MHz.
8671: f = 8 MHz.
8612: f = 12, 16 MHz.
86C08/E08: f = 8, 12 MHz.
86C09/19: xtal/resonator, f = 8 MHz, C = 47 pf max.
86C00/10/20/30: f = 8, 12, 16 MHz.
86C11/21/91/40/90: f = 12, 16, 20 MHz.
86C27/97: f = 4, 8 MHz.
86C12: f = 12, 16 MHz.
Super8 (all): f = 1 - 20 MHz.

Z8000® Family (8581 only)

General Requirements:
Crystal cut: AT cut, parallel resonant, fundamental mode.
Crystal Co: < 7 pf for all frequencies.
Crystal Rs: < 150 ohms for all frequencies.
Load capacitance: 10 to 33 pf.

Z80® Family

General Requirements:
Crystal cut: AT cut, parallel resonant, fundamental mode.
Crystal Co: < 7 pf for all frequencies.
Crystal Rs: < 60 ohms for all frequencies.
Load capacitance: 10 to 22 pf.

Specific Requirements:
84C01: C1 = 22 pf, C2 = 33 pf (typ); f = DC to 10 MHz.
84C90: DC to 8 MHz.
84C50: same as 84C01.
84C11/13/15: C1 = C2 = 20 - 33 pf; f = 6 -10 MHz
80180: f = 12, 16, 20 MHz (Fxtal = 2 x sys. clock).
80280: f = 20 MHz (Fxtal = 2 x Fsyclick).
80181: TBD.
Communications Family

General Requirements:
- Crystal cut: AT cut, parallel resonant, fundamental mode.
- Crystal Co: < 7 pf for all frequencies.
- Crystal Rs: < 150 ohms for all frequencies.
- Load capacitance: 20 to 33 pf.
- Frequency: cannot exceed PCLK.

Specific Requirements:
- 8530/85C30/SCC: f = 1 - 6 MHz (10 MHz SCC), 1 - 8.5 MHz (8 MHz SCC).
- 85130/ESCC (16/20 MHz), f = 1 - 16.384 MHz.
- 16C35/ISC: f = 1 - 10 MHz.

REFERENCES MATERIALS AND ACKNOWLEDGMENTS


National Semiconductor Corp., App Notes 326 and 400.

Zilog, Inc., Steve German; Figures 4 and 8.


Data Sheets; CTS Corp. Knights Div., Crystal Oscillators.
# Z86C0800ZCO Evaluation Board
## Product Specification

### Devices Supported:
Z86C08, Z86C04

### Description
The Z86C0800ZCO Evaluation Board kit contains an assembled circuit board, software and documentation to help the user become familiar with the features of the Z86C08 microcontroller.

The Z86C0800ZCO Evaluation Board is used to demonstrate the advantages and versatility of the 18-pin Z8 device. The kit contains hardware and software that demonstrates the implementation of WDT, HALT, and STOP mode, low cost D to A, and A to D conversion techniques.

### Specifications
#### Power Requirements
+5 Vdc @ 50 mA

#### Dimensions
- Width: 4.4 in. (11.2 cm)
- Length: 4.8 in. (12.2 cm)

### Kit Contents
- Z86C08 Evaluation Board
- CMOS Z86C08 MPU
- 4 MHz Crystal
- Four 7-Segment LED Displays
- 17-Key Keypad

- Software (IBM® PC Platform)
  - Application Source Code
  - Z8®/Z80®/Z8000® Cross Assembler
  - MOBJ Link/Loader

- Documentation
  - Discrete Z8® Databook
  - Z8 Cross Assembler User's Guide
  - MOBJ Link/Loader User's Guide
  - Z86C08 Evaluation Board User's Guide

### Ordering Information
- Part No: Z86C0800ZCO
Z86C08 Adaptor Kit

PRODUCT SPECIFICATION

DEVICE SUPPORTED: Z86C08

DESCRIPTION
The Z86C08 Adaptor Kit is used to convert a Z8® MCU 40-pin package to an 18-pin package. This adaptor board allows a standard Z8 emulation device to emulate the Z86C08. The Z86C08 Adaptor Board is placed between the Z8 emulator and the user's target socket. The board does not emulate the watch-dog timer function.

SPECIFICATIONS
Dimensions
Width: 2.5 in. (6.4 cm)
Length: 2.9 in. (7.4 cm)

KIT CONTENTS
Z86C08 Adaptor Board
40-Pin Z8 MPU Socket
18-Pin Z86C08 Socket
12 MHz Crystal

Cables
18-Pin Z86C08 Emulation Cable

Documentation
Z86C08 Adaptor Kit User's Guide

ORDERING INFORMATION
Part No: Z86C0800ZDP
Z86C1200ZEM EMULATOR
PRODUCT SPECIFICATION

DEVICES SUPPORTED: Z86C04/E04, Z86C07, Z86C08/E08, Z86C11, Z86C20,
Z86C21/E21[1], Z86E22[1], Z86E23[2], Z86C61, Z86C63, Z86C65, Z86C91

DESCRIPTION
The Z86C1200ZEM Z8® Emulator is a member of Zilog's ICEBOX™ product family of in-circuit emulators. The Z86C1200ZEM provides emulation and OTP programming support for Zilog's Z8 microcontrollers. The Emulator provides all the essential MCU timing and I/O circuitry which simplifies user emulation of the prototype hardware/software product. The data entering, program debugging, and OTP programming are performed by the monitor ROM and the Host Package which communicates through a RS-232C serial interface with a fixed 19200 baud rate. The user program can be downloaded directly from the host computer through the RS-232C connector. The user code may then be executed using various debugging commands in the monitor. The Emulator can be connected to a serial port COM 1 or COM 2 of the host computer (IBM® XT, AT 386, 486Compatible).

SPECIFICATIONS
Emulation Specification
Maximum Emulation Speed 16 MHz

Power Requirements
+5 Vdc @ 1.0 A

Dimensions
Width: 6.0 in. (15.2 cm)
Length: 8.8 in. (22.4 cm)

Serial Interface
RS-232C @ 19200 baud

KIT CONTENTS
Z86C12 Emulator
Z8 Emulation Base Board (Revision B)
CMOS Z86C9120PSC
8K X 8 EPROM (Programmed with Debug Monitor)
EPM5128 EPLD
32K X 8 STATIC RAM
3.64K X 4 STATIC RAM
RS-232C Interface
Reset Switch
Z86C12 Emulation Daughter Board
EPM5032 EPLD
16 MHz CMOS Z86C1216GSE ICE Chip
40/18 Pin ZIF OTP Sockets
80/60/40 Pin Target Connectors

Cables
12', 40-Pin DIP Emulation Cable
12', 28-Pin DIP Emulation Cable
12', 18-Pin DIP Emulation Cable
15', Power Cable with Banana Plugs
48', Power Cable
60', DB 25 RS-232C Cable

Software (IBM®-PC Platform)
Z8/Z80/Z8000 Cross Assembler
MOBJ Link/Loader
Host Package (Revision 1.5)
Includes Windows and non-Windows

Documentation
Emulator User’s Guide
Support Products Catalog
Z8 Cross Assembler User’s Guide
MOBJ Link/Loader User’s Guide
Registration Card

ORDERING INFORMATION
Part No: Z86C1200ZEM

Notes:
[2] With Z86E2300ZDP Programming Adaptor, Rev. 1.0
**DEVICES SUPPORTED:**  Z86C06/09/19

**DESCRIPTION**
The Z86E06 Adaptor Kit converts the 28-pin footprint of Zilog's Z86E30 OTP chip to the 18-pin DIP configuration of the Z86E06/09/19 OTP chip. The board supports all the functions of the Z86C06/09/19 except for Serial Peripheral Interface.

**SPECIFICATIONS**
**Dimensions**
- Width: 0.8 in. (2.0 cm)
- Length: 1.5 in. (3.8 cm)

**KIT CONTENTS**
- Z86E06 Adaptor Kit
- 28-Pin Z86E30 MCU Socket
- 18-Pin Z86C06/09/19 Connector

**Documentation**
- Z86E06 OTP Conversion Kit User's Guide

**ORDERING INFORMATION**
- Part No: Z86E0600ZDP
DEVICE SUPPORTED: Z86E07 SOIC

DESCRIPTION
The Z86E07 Adaptor Kit converts an 18-pin SOIC package to an 18-pin DIP package, allowing a Z86E07 DIP OTP programmer to program the 18-pin SOIC Z86E07 OTP microcontroller.

SPECIFICATIONS
Dimensions
- Width: 0.95 in.
- Length: 1.10 in.

Operating Temperature
- 0 to 50°C

Operating Humidity
- 10-90% RH (non-condensing)

KIT CONTENTS
Z86E07 Adaptor Board
- 18-Pin SOIC ZIF Package
- 18-Pin DIP Connector

Documentation
Z86E07 OTP Adaptor Kit User's Guide

ORDERING INFORMATION
Part No: Z86E0700ZDP
DEVICES SUPPORTED:  Z86E30, Z86E31

DESCRIPTION
The Z86E30 DIP OTP Adaptor Kit allows a standard EPROM programmer to program the Z86E30 OTP microcontroller.

SPECIFICATIONS
Power Requirements
+12.5 Vdc @ .5A

Dimensions
Width:  1.45 in. (3.68 cm)
Length: 2.0 in. (5.08 cm)

KIT CONTENTS
Z86E30 OTP Program Adaptor Board
28-Pin DIP ZIF Socket
28-Pin Connector

Documentation
OTP Program Adaptor User’s Guide

ORDERING INFORMATION
Part No:  Z86E3000ZDP
DESCRIPTION
The Z86E40 QFP OTP Adaptor Kit allows a standard EPROM programmer to program the Z86E40 OTP microcontroller.

SPECIFICATIONS
Power Requirements
+12.5 Vdc @ .5 A

Dimensions
Width: 1.75 in. (4.4 cm)
Length: 2.20 in. (5.6 cm)

KIT CONTENTS
Z86E40 QFP OTP Adaptor Board
44-Pin QFP ZIF Socket
28-Pin Connector

Documentation
OTP Program Adaptor User's Guide

ORDERING INFORMATION
Part No: Z86E4000ZDF
**DEVICE SUPPORTED:** Z86E40

**DESCRIPTION**
The Z86E40 DIP OTP Adaptor Kit allows a standard EPROM programmer to program the Z86E40 OTP microcontroller.

**SPECIFICATIONS**

**Power Requirements**
+12.5 Vdc @ .5 A

**Dimensions**
- Width: 1.4 in. (3.6 cm)
- Length: 2.6 in. (6.6 cm)

**KIT CONTENTS**
- Z86E40 DIP OTP Adaptor Board
- 40-Pin DIP ZIF Socket
- 28-Pin Connector
- Documentation
- OTP Program Adaptor User's Guide

**ORDERING INFORMATION**
- Part No: Z86E4000ZDP
**DEVICE SUPPORTED:**  Z86E40

**DESCRIPTION**
The Z86E40 PLCC OTP Adaptor Kit allows a standard EPROM programmer to program the Z86E40 OTP microcontroller.

**SPECIFICATIONS**

<table>
<thead>
<tr>
<th>Power Requirements</th>
<th>+12.5 Vdc @ .5 A</th>
</tr>
</thead>
</table>

| Dimensions          | Width: 1.6 in. (4.1 cm)  |
|                     | Length: 2.0 in. (5.1 cm) |

**KIT CONTENTS**

- Z86E40 PLCC OTP Adaptor Board
- 40-Pin ZIF Socket
- 28-Pin Connector

**Documentation**

- OTP Program Adaptor User's Guide

**ORDERING INFORMATION**

- Part No: Z86E4000ZDV
**DESCRIPTION**
The Z86E40 OTP Adaptor Kit converts a 44-pin QFP package to a 40-pin DIP package, allowing the C50 ICEBOX™ to program the 44-pin QFP Z86E40 OTP microcontroller.

**SPECIFICATIONS**

<table>
<thead>
<tr>
<th>Power Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>+12.5 Vdc @ .5 A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width: 2.05 in. (5.2 cm)</td>
</tr>
<tr>
<td>Length: 2.10 in. (5.3 cm)</td>
</tr>
</tbody>
</table>

**KIT CONTENTS**
- Z86E40 OTP Adaptor Board
- 44-Pin QFP ZIF Socket
- 40-Pin Connector
- Documentation
  - OTP Program Adaptor User's Guide

**ORDERING INFORMATION**
- Part No: Z86E4001ZDF
Z86E4001ZDV Adaptor Kit
Product Specification

DEVICE SUPPORTED: Z86E40

DESCRIPTION
The Z86E40 OTP Adaptor Kit converts a 44-pin PLCC package to a 40-pin DIP package, allowing the C50 ICEBOX" to program the 44-pin PLCC Z86E40 OTP microcontroller.

SPECIFICATIONS
Power Requirements
+12.5 Vdc @ .5 A

Dimensions
Width: 1.8 in. (4.6 cm)
Length: 2.1 in. (5.3 cm)

KIT CONTENTS
Z86E40 OTP Adaptor Board
40-Pin PLCC ZIF Socket
40-Pin Connector

Documentation
OTP Program Adaptor User's Guide

ORDERING INFORMATION
Part No: Z86E4001ZDV
DESCRIPTION
The Z86CCPO0ZEM is a member of Zilog's family of in-circuit emulators. The Z8 CCP emulator provides emulation and OTP programming support for Zilog's Consumer Controller Processor (CCP™) microcontroller. The Emulator provides all the essential MCU timing and I/O circuitry which simplifies user emulation of the prototype hardware/software product.

The data entering, program debugging, and OTP programming are performed by the monitor ROM and the Host Package which communicates through RS-232C serial interface with a fixed 19200 baud rate. The user program can be downloaded directly from the host computer via an RS-232C connector. The user code may then be executed using various debugging commands in the monitor. The Emulator can be connected to any serial port (COM 1, 2, 3 or 4) of the host computer.

The Z86CCP00ZEM supports stand-alone mode (without host mode) through special jumper options. In stand-alone mode, the C50 ICE chip is used in conjunction with a PROM and can support speeds of DC to 12 MHz at Vcc of 3V to 5.5V.

SPECIFICATIONS
Emulation Specification
Maximum Emulation Speed 8 MHz
Minimum Emulation Speed 1 MHz

Power Requirements
+8V Vdc @ 0.5 A

Dimensions
Width: 7.0 in. (17.7 cm)
Length: 9.0 in. (22.9 cm)

Serial Interface
RS-232C @ 19200 baud

System Requirements
IBM-Compatible 286 @ 12 MHz or Newer
CPU Running DOS 5.0 or Higher
Windows Version 3.0 or Higher
720 Kbytes of Disk Space
512 Kbytes of RAM

KIT CONTENTS
Z8 CCP Emulator
Z8 CCP Emulation Board (Revision A)
CMOS Z86C9320VSC
RS-232C Interface
Reset Switch
20 MHz CMOS Z86C5020FSE ICE Chip
8K x 8 STATIC RAM for Code Memory
18-Pin ZIF OTP Socket
Socket Available for 18-Pin Target Connector
18-Pin Target Connector Cable
Holes Available for 40-Pin ZIF Socket
Socket Available for 40-Pin Target Connector
Holes Available for 28-Pin ZIF Socket
Socket Available for 28-Pin Target Connector

Software (IBM PC platform)
Z8/Z80/Z8000 Cross Assembler
MOBJ Link/Loader
Emulator GUI Host Package

Documentation
Emulator User's Guide
Z8 Cross Assembler User's Guide
MOBJ Link/Loader User's Guide
Registration Card
Z8 CCP Emulator GUI User's Guide
Discrete Z8 Data Book
Z8 Microcontroller Technical Manual

ORDERING INFORMATION
Part No: Z86CCP00ZEM
Z86CCP00ZAC EMULATOR KIT
PRODUCT SPECIFICATION

DEVICES SUPPORTED: Z86C31/E31, Z86C30/E30, Z86C40/E40

DESCRIPTION
The Z86CCP00ZAC is the accessory kit for the Z86CCP00ZEM. The kit contains all accessories to fully populate and operate all functions of the Z86CCP00ZEM.

SPECIFICATIONS
Emulation Specification
- Maximum Emulation Speed: 8 MHz
- Minimum Emulation Speed: 1 MHz
- Maximum Emulation Speed
  - Stand-Alone Mode: 16 MHz
- Minimum Emulation Speed
  - Stand-Alone Mode: DC

Power Requirements
- +8V Vdc @ 0.5 A

Dimensions
- Width: 7.0 in. (17.7 cm)
- Length: 9.0 in. (22.9 cm)

Serial Interface
- RS-232C @ 19200 baud

System Requirements
- IBM-Compatilbe 286 @ 12 MHz or Newer
- CPU Running DOS 5.0 or Higher
- Windows Version 3.0 or Higher
- 720 Kbytes of Disk Space
- 512 Kbytes of RAM

KIT CONTENTS
Z8 CCP Emulator Kit
- 28-Pin ZIF Socket
- 28-Pin Target Connector Cable
- 40-Pin ZIF Socket
- 40-Pin Target Connector Cable
- RS-232 Cable
- Power Cable

Software (IBM PC platform)
- Z8®/Z80®/Z8000® Cross Assembler
- MOBJ Link/Loader
- Emulator GUI Host Package

Documentation
- Emulator User's Guide
- Z8 Cross Assembler User's Guide
- MOBJ Link/Loader User's Guide
- Registration Card
- Z8 CCP Emulator GUI User's Guide
- Discrete Z8 Data Book
- Z8 Microcontroller Technical Manual

ORDERING INFORMATION
Part No: Z86CCP00ZAC
Z8® S SERIES EMULATORS
BASE UNITS AND PODS

DESCRIPTION
The system comprises 24 MHz or 33 MHz base unit options, and pod options which allow the emulation of various Z8 microcontrollers. Features include real-time transparent emulation up to 33 MHz, in-line symbolic assembler and disassembler, real-time hardware breakpoints, eight channel user logic analyzer, external trigger input and outputs, trace display and memory display/edit during execution, and window or command driven user interface.

SPECIFICATIONS
Microcontrollers Emulated:
- Z86C1200ZPD
  - Z86C00, Z86C10, Z86C11, Z86C20, Z86C21, Z86C21, Z86C91, Z86C61, Z86C63
- Z86C5000ZPD
  - Z86C03, Z86C06, Z86C09, Z86C19, Z86C30, Z86C31, Z86C40, Z86C90
- Z86C9300ZPD
  - Z86C93 (24 MHz)
- Z86C931ZPD
  - Z86C93 (33 MHz)
- Z86C9500ZPD
  - Z86C95 (24 MHz)
- Z86C951ZPD
  - Z86C95 (33 MHz)

Maximum Emulation Speed:
- Up to 24 MHz (microcontroller dependent)
- Up to 33 MHz (Z86C93 and Z86C95)

Size:
- 260 mm wide, 260 mm deep, 64 mm high

Operating Temperature:
- 0°C to +40°C

Storage Temperature:
- -10°C to +65°C

Operating Humidity:
- 0 to 90%

Maximum Emulation Program and Data Memory:
- 64 Kbytes

Program Memory Mapping:
- 1K blocks

Pass Counters:
- Two, 16-bit each

Trace Buffer:
- 32K-80 bits

Sequencer:
- Hardware, 8 levels

User Probe:
- Eight channel logic input
- One trigger input
- Seven trigger outputs (Events, Pass Counters, Sequencer)

Host Interface:
- Asynchronous RS-232C
  - 9600/115 Kbaud
  - XON/XOFF support

File Upward/Downward Format:
- Zilog MUFOM (EEE 695-1985)
- Intel® HEX
- Intel AOMF
- 2500AD® Software

MINIMUM HOST REQUIREMENTS
- IBM® compatible PC/XT/AT/386/486 or PS-2
- 640 Kbyte memory
- 20 Mbyte hard disk
- RS-232 serial port (COM 1 or COM 2)
- Mouse (serial or bus)
- MDA, CGA, EGA, or VGA video adaptor

MINIMUM EMULATION SUPPORT
- One base unit
- One emulation pod

ORDERING INFORMATION:

<table>
<thead>
<tr>
<th>Base Unit</th>
<th>Emulation Pod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z86C00002ZUSP064</td>
<td>Z86C1200ZPD</td>
</tr>
<tr>
<td>Z86C00012ZUSP064 (33 MHz)</td>
<td>Z86C5000ZPD</td>
</tr>
<tr>
<td>Z86C9500ZUSP064</td>
<td>Z86C93002ZPD (24 MHz)</td>
</tr>
<tr>
<td>Z86C9512ZUSP064 (33 MHz)</td>
<td>Z86C9301ZPD (33 MHz)</td>
</tr>
<tr>
<td>Z86C9500ZPD (24 MHz)</td>
<td>Z86C9502ZPD (24 MHz)</td>
</tr>
<tr>
<td>Z86C9501ZPD (33 MHz)</td>
<td>Z86C9501ZPD (33 MHz)</td>
</tr>
</tbody>
</table>
## Z8® Hardware and Software Third Party Support

### Z8 Support

<table>
<thead>
<tr>
<th>Company</th>
<th>Assembler</th>
<th>C Compiler</th>
<th>Simulator</th>
<th>Operating System</th>
<th>Phone Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allen Ashley</td>
<td>X</td>
<td></td>
<td>X</td>
<td>DOS, CP/M</td>
<td>(818) 793-5748</td>
</tr>
<tr>
<td>Avocet Systems</td>
<td>X</td>
<td></td>
<td></td>
<td>DOS</td>
<td>(800) 448-8500</td>
</tr>
<tr>
<td>Byte Craft</td>
<td></td>
<td>X</td>
<td></td>
<td>DOS</td>
<td>(519) 888-6911</td>
</tr>
<tr>
<td>Cybernetic Micro</td>
<td>X</td>
<td></td>
<td></td>
<td>DOS</td>
<td>(415) 726-3000</td>
</tr>
<tr>
<td>Micro Computer Control</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>DOS</td>
<td>(609) 466-1751</td>
</tr>
<tr>
<td>Micro Dialects</td>
<td>X</td>
<td></td>
<td></td>
<td>Macintosh</td>
<td>(513) 271-9100</td>
</tr>
<tr>
<td>Production Language Corp.</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>DOS (386+)</td>
<td>(817) 599-8363</td>
</tr>
<tr>
<td>Pseudo Corp.</td>
<td>X</td>
<td></td>
<td>X</td>
<td>DOS</td>
<td>(804) 873-1947</td>
</tr>
<tr>
<td>Software Development Systems</td>
<td>X</td>
<td></td>
<td></td>
<td>DOS UNIX</td>
<td>(800) 448-7733</td>
</tr>
<tr>
<td>Western Wares</td>
<td>X</td>
<td></td>
<td></td>
<td>DOS CP/M-80 ISIS-II</td>
<td>(303) 327-4888</td>
</tr>
<tr>
<td>2500AD Software</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>DOS UNIX CP/M VAX VMS</td>
<td>(719) 395-8683</td>
</tr>
</tbody>
</table>

### Super8® Support

<table>
<thead>
<tr>
<th>Company</th>
<th>Assembler</th>
<th>C Compiler</th>
<th>Simulator</th>
<th>Operating System</th>
<th>Phone Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allen Ashley</td>
<td>X</td>
<td></td>
<td>X</td>
<td>Macintosh</td>
<td>(818) 793-5748</td>
</tr>
<tr>
<td>Micro Computer Control</td>
<td></td>
<td>X</td>
<td></td>
<td>DOS</td>
<td>(609) 466-1751</td>
</tr>
<tr>
<td>Pseudo Corp.</td>
<td>X</td>
<td></td>
<td>X</td>
<td>DOS</td>
<td>(804) 873-1947</td>
</tr>
<tr>
<td>2500AD Software</td>
<td>X</td>
<td></td>
<td>X</td>
<td>DOS</td>
<td>(719) 395-8683</td>
</tr>
</tbody>
</table>
### Emulators

<table>
<thead>
<tr>
<th>Emulator</th>
<th>Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development System</td>
<td></td>
</tr>
<tr>
<td>Creative Technology</td>
<td></td>
</tr>
<tr>
<td>iSystems</td>
<td></td>
</tr>
<tr>
<td>JK Board V.3.8</td>
<td></td>
</tr>
<tr>
<td>MicroTime</td>
<td></td>
</tr>
<tr>
<td>Orion Instruments</td>
<td></td>
</tr>
<tr>
<td>Signum Systems</td>
<td></td>
</tr>
<tr>
<td>Wytec</td>
<td></td>
</tr>
</tbody>
</table>

A = Emulate with Z86C0800ZDP Adaptor  
B = Emulate with Z8612 Board  
C = Emulate with Z86C0800ZDP and Z8612 Board or Z86C0800ZEM  
D = Emulate with Z8612 Board  
E = Emulate with Z86C90 Board

### OTP Programmers

<table>
<thead>
<tr>
<th>OTP Programmers</th>
<th>Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development System</td>
<td></td>
</tr>
<tr>
<td>Data I/O, Inc.</td>
<td></td>
</tr>
<tr>
<td>Logical Devices, Inc.*</td>
<td></td>
</tr>
<tr>
<td>Needham, Inc.</td>
<td></td>
</tr>
<tr>
<td>Smart Access, Inc.</td>
<td></td>
</tr>
</tbody>
</table>

* Single and Gang Programming Available
## Z80® & Z80180 Hardware and Software Third Party Support

### Hardware Support

<table>
<thead>
<tr>
<th>Company</th>
<th>Product</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>American Automation</td>
<td>Emulator</td>
<td>(714) 731-1661</td>
</tr>
<tr>
<td>Applied Microsystems</td>
<td>Emulator</td>
<td>(206) 882-2000</td>
</tr>
<tr>
<td>Hewlett-Packard</td>
<td>HP 64000/UJ/PC Emulator</td>
<td>(800) 4HP-DATA</td>
</tr>
<tr>
<td>Huntsville Microsystems</td>
<td>Emulator</td>
<td>(205) 881-6005</td>
</tr>
<tr>
<td>iSystems (Germany)</td>
<td>Emulator</td>
<td>08131-25083</td>
</tr>
<tr>
<td>Micromint</td>
<td>SB180,SB180FX,</td>
<td>(800) 635-3355</td>
</tr>
<tr>
<td></td>
<td>BCC180,RTC180</td>
<td></td>
</tr>
<tr>
<td>MicroWorks</td>
<td>Prototyping board</td>
<td>(408) 997-1644</td>
</tr>
<tr>
<td>Orion Instruments†</td>
<td>Emulator</td>
<td>(415) 327-8800</td>
</tr>
<tr>
<td>Pentica Systems, Inc.</td>
<td>Emulator, ICEBOX, ICE</td>
<td>(617) 577-1101</td>
</tr>
<tr>
<td>Softaid†</td>
<td>Analyzer, (symbolic</td>
<td></td>
</tr>
<tr>
<td></td>
<td>debug)</td>
<td></td>
</tr>
<tr>
<td>Sophia Systems</td>
<td>Emulator, SA2000</td>
<td>(415) 493-6700</td>
</tr>
<tr>
<td>Versalogic</td>
<td>Z80 STD Bus</td>
<td>(503) 485-8575</td>
</tr>
<tr>
<td></td>
<td>circuit board</td>
<td></td>
</tr>
<tr>
<td>Z-World</td>
<td>IBM PC</td>
<td>(916) 753-3722</td>
</tr>
<tr>
<td></td>
<td>Development Bd.</td>
<td></td>
</tr>
<tr>
<td>Zaxtek</td>
<td>Emulator</td>
<td>(714) 474-1170</td>
</tr>
<tr>
<td>Zilog</td>
<td>S180+ESCC (Z8S18000ZCO)</td>
<td><em>Call Zilog</em></td>
</tr>
<tr>
<td></td>
<td>Application Board</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Z80181 Eval. Kit</td>
<td><em>Call Zilog</em></td>
</tr>
<tr>
<td></td>
<td>(Z8018100ZCO)</td>
<td></td>
</tr>
<tr>
<td>Zilog</td>
<td>Z84C15 Eval. Kit</td>
<td><em>Call Zilog</em></td>
</tr>
<tr>
<td></td>
<td>(Z84C1600ZCO)</td>
<td></td>
</tr>
<tr>
<td>Zilog</td>
<td>Z84C50+KIO Application Bd.</td>
<td><em>Call Zilog</em></td>
</tr>
<tr>
<td></td>
<td>(Z84C5000ZCO)</td>
<td></td>
</tr>
<tr>
<td>Zilog</td>
<td>Z84C01+KIO Development Bd.</td>
<td><em>Call Zilog</em></td>
</tr>
<tr>
<td></td>
<td>(Z84C9000ZCO)</td>
<td></td>
</tr>
<tr>
<td>Emulation Technology†‡</td>
<td>Emulator</td>
<td>(408) 982-0660</td>
</tr>
</tbody>
</table>

**Note:**
Z80180 Emulators can be utilized for Z80182 in Eval Mode 1.

† Supports Z182 in Mode 0 also.
‡ Supports Z182 in Mode 2 also.

### Assemblers and Cross Assemblers

<table>
<thead>
<tr>
<th>Company</th>
<th>Host/Comments</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>2500AD</td>
<td>C; IBM PC, CP/M, VAX, Sun</td>
<td>(800) 843-8144</td>
</tr>
<tr>
<td>American Automation</td>
<td>C; IBM PC</td>
<td>(714) 731-1661</td>
</tr>
<tr>
<td>Archimedes</td>
<td>C; IBM PC, Sun, VAX, HP</td>
<td>(415) 567-4010</td>
</tr>
<tr>
<td>Avocet Systems</td>
<td>C; IBM PC</td>
<td>(800) 448-8500</td>
</tr>
<tr>
<td>Laboratory</td>
<td>Forth; IBM PC</td>
<td>(213) 306-7412</td>
</tr>
<tr>
<td>Micromint</td>
<td>Z-System OS</td>
<td>(800) 635-3355</td>
</tr>
<tr>
<td>Softaid</td>
<td>Z80180 Guide, IBM PC diskette</td>
<td>(800) 433-8812</td>
</tr>
<tr>
<td>Z-World Dynamic</td>
<td>C; Uniware, IBM PC, VAX, UNIX/VMX, Apollo</td>
<td>(916) 753-3722</td>
</tr>
</tbody>
</table>

### Simulators/Applications Software

<table>
<thead>
<tr>
<th>Company</th>
<th>Host/Comments</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avocet Systems</td>
<td>Simulator/IBM PC</td>
<td>(800) 448-8500</td>
</tr>
<tr>
<td>Lear Com Company</td>
<td>Simulator/IBM PC</td>
<td>(303) 232-2226</td>
</tr>
<tr>
<td>Logisof</td>
<td>8080 to Z80 Translator</td>
<td>(408) 773-8465</td>
</tr>
<tr>
<td>Micromint</td>
<td>Z-System OS</td>
<td>(800) 635-3355</td>
</tr>
<tr>
<td>Softaid</td>
<td>Z80180 Guide, IBM PC diskette</td>
<td>(800) 433-8812</td>
</tr>
<tr>
<td>The AG Group</td>
<td>LLAP Dvmnt/Apple</td>
<td>(510) 937-7900</td>
</tr>
</tbody>
</table>

### PLCC Socket Manufacturers:
Methode, TNB, ITT, Cannon, Precicontact

### Shrink DIP Socket Manufacturers:
TI, Bevar, Yamaichi

### 44/80/100 Pin QFP:
ZIF (Zero Insertion Force) sockets for prototyping may be obtained from Yamaichi Electronics, (408) 450-0797.

### 100-Pin QFP Clip:
Emulation Technology, 408-982-0660

---

68 PLCC Socket Manufacturers:
Methode, TNB, ITT, Cannon, Precicontact

64 Shrink DIP Socket Manufacturers:
TI, Bevar, Yamaichi

44/80/100 Pin QFP:
ZIF (Zero Insertion Force) sockets for prototyping may be obtained from Yamaichi Electronics, (408) 450-0797.

100-Pin QFP Clip:
Emulation Technology, 408-982-0660
## Z80, Z80180, Z80280, & Z80380 Hardware and Software Third Party Support

### Z80 & Z80180 High Level Language Compilers

<table>
<thead>
<tr>
<th>Company</th>
<th>Language Host</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>2500AD</td>
<td>C, IBM PC, CP/M, VAX</td>
<td>(800) 843-8144</td>
</tr>
<tr>
<td>American Automation</td>
<td>C, IBM PC</td>
<td>(714) 731-1661</td>
</tr>
<tr>
<td>Archimedes</td>
<td>C, IBM PC, Sun, VAX, HP</td>
<td>(415) 567-4010</td>
</tr>
<tr>
<td>Avocet Systems</td>
<td>C, IBM PC</td>
<td>(800) 448-8500</td>
</tr>
<tr>
<td>Laboratory</td>
<td>Forth IBM PC</td>
<td>(213) 306-7412</td>
</tr>
<tr>
<td>Microtek Lab, Inc.</td>
<td>C, Pascal IBM PC, Microsystems</td>
<td>(213) 321-2121</td>
</tr>
<tr>
<td>Microtek Research</td>
<td>Sun Micro, VAX</td>
<td>(408) 980-1300</td>
</tr>
<tr>
<td>MPE</td>
<td>Forth IBM PC</td>
<td>(716) 461-9187</td>
</tr>
<tr>
<td>Softaid</td>
<td>MT-Basic IBM PC</td>
<td>(800) 433-8812</td>
</tr>
<tr>
<td>Software Development Systems</td>
<td>C, IBM PC, VAX, Sun, Apollo</td>
<td>(708) 971-8170</td>
</tr>
<tr>
<td>Z-World</td>
<td>Dynamic C, IBM PC</td>
<td>(916) 753-3722</td>
</tr>
</tbody>
</table>

**Note:** Z80/64180 software is also compatible with the Z80180.

### Simulators

<table>
<thead>
<tr>
<th>Company</th>
<th>Host/Comments</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro Methods, Inc.</td>
<td>IBM PC (ZRPM)</td>
<td>(503) 861-1765</td>
</tr>
</tbody>
</table>

### Communications Software Support

#### SCC Physical Layer Drivers and Upper Layer Software

<table>
<thead>
<tr>
<th>Company</th>
<th>Software</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT Barret Assoc.</td>
<td>Software drivers</td>
<td>(713) 728-8688</td>
</tr>
<tr>
<td>Forward Technology</td>
<td>for SCCs/ESCCs</td>
<td>(516) 496-9033</td>
</tr>
<tr>
<td>GCOM</td>
<td>Drivers for SCCs/ESCCs</td>
<td>(217) 337-4471</td>
</tr>
<tr>
<td>Probitas</td>
<td>LLAP Driver, AppleTalk protocol stack, custom projects</td>
<td>(415) 941-2090</td>
</tr>
</tbody>
</table>

**Note:** Z80 software is object code compatible with the Z80.

### High Level Language Compilers

<table>
<thead>
<tr>
<th>Company</th>
<th>Language Host</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>2500AD</td>
<td>C, IBM PC, CP/M, VAX</td>
<td>(800) 843-8144</td>
</tr>
<tr>
<td>Computer Design Solutions</td>
<td>C, IBM PC</td>
<td>(704) 876-2346</td>
</tr>
</tbody>
</table>

**Note:** This is NOT a complete list of hardware and software vendors who support Zilog products. Please contact the Zilog sales office nearest you if what you are looking for is not on this list. This list is for reference only and is not an endorsement for any company.

### Z80280 Hardware Support

<table>
<thead>
<tr>
<th>Company</th>
<th>System Name</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Design Solutions</td>
<td>STD Buscard &amp; Z80 Dvmt. Board</td>
<td>(704) 876-2346</td>
</tr>
<tr>
<td>Softaid</td>
<td>Z808 ICE Analyzer</td>
<td>(800) 433-8812</td>
</tr>
</tbody>
</table>

### 68 PLCC Socket Manufacturers:

Methods, TNB, ITT, Cannon, Precicontact

### Z80280 Assemblers and Cross Assemblers:

<table>
<thead>
<tr>
<th>Company</th>
<th>Host/Comments</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>2500AD</td>
<td>IBM PC, CP/M, VAX</td>
<td>(800) 843-8144</td>
</tr>
<tr>
<td>Computer Design Solutions</td>
<td>IBM PC</td>
<td>(704) 876-2346</td>
</tr>
</tbody>
</table>

### Z80380 Hardware Support

<table>
<thead>
<tr>
<th>Company</th>
<th>System Name</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signum</td>
<td>In-Circuit Emulator</td>
<td>(805) 371-4608</td>
</tr>
<tr>
<td>Zilog</td>
<td>Z80380 Evaluation Kit</td>
<td></td>
</tr>
</tbody>
</table>

### Z80 Assemblers and Cross Assemblers:

<table>
<thead>
<tr>
<th>Company</th>
<th>Host/Comments</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLC</td>
<td>IBM PC, Unix</td>
<td>(817) 599-8365</td>
</tr>
<tr>
<td>2500AD</td>
<td>IBM PC, CP/M, VAX</td>
<td>(800) 843-8144</td>
</tr>
<tr>
<td>Company</td>
<td>Product</td>
<td>Company</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Allen Ashley</td>
<td>Assembler</td>
<td>Logical Devices, Inc.</td>
</tr>
<tr>
<td>395 Sierra Madre Villa</td>
<td>Disassembler</td>
<td>1201 NW 65th Place</td>
</tr>
<tr>
<td>Pasadena, CA 91107-2902</td>
<td>Simulator</td>
<td>Fort Lauderdale, FL 33309</td>
</tr>
<tr>
<td>(818) 793-5748</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avocet Systems</td>
<td>Assembler</td>
<td>Micro Computer Control</td>
</tr>
<tr>
<td>120 Union Street</td>
<td></td>
<td>P.O. Box 275 / 17 Model Ave.</td>
</tr>
<tr>
<td>Rockport, ME 04856</td>
<td></td>
<td>Hopewell, NJ 08525</td>
</tr>
<tr>
<td>(800) 448-8500</td>
<td></td>
<td>(609) 466-1751</td>
</tr>
<tr>
<td>Byte Craft Limited</td>
<td>C Compiler</td>
<td>Micro Dialects</td>
</tr>
<tr>
<td>421 King Street North</td>
<td></td>
<td>P.O. Box 30014</td>
</tr>
<tr>
<td>Waterloo, Ontario</td>
<td></td>
<td>Cincinnati, OH 45230</td>
</tr>
<tr>
<td>Canada N2J4E4</td>
<td></td>
<td>(513) 271-9100</td>
</tr>
<tr>
<td>(519) 888-6911</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creative Technology</td>
<td>Emulator</td>
<td>MicroTime</td>
</tr>
<tr>
<td>5144 Peachtree Road</td>
<td></td>
<td>10F No. 1180 Chen-De Rd.</td>
</tr>
<tr>
<td>Suite 301</td>
<td></td>
<td>11148 Taipei, Taiwan, R.O.C.</td>
</tr>
<tr>
<td>Atlanta, GA 30341</td>
<td></td>
<td>11-886-2-881-1791</td>
</tr>
<tr>
<td>(404) 455-8255</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cybernetic Micro Systems</td>
<td>Assembler</td>
<td>MPE</td>
</tr>
<tr>
<td>P.O. Box 3000</td>
<td></td>
<td>2604 Elmwood Ave.</td>
</tr>
<tr>
<td>San Gregorio, CA 94074</td>
<td></td>
<td>Rochester, NY 14618</td>
</tr>
<tr>
<td>(415) 726-3000</td>
<td></td>
<td>(716) 461-9187</td>
</tr>
<tr>
<td>Dantrol</td>
<td>Emulator</td>
<td>Needham Electronics</td>
</tr>
<tr>
<td>1910 Rena Ln.</td>
<td></td>
<td>4539 Orange Grove Ave.</td>
</tr>
<tr>
<td>Dalton, GA 30720</td>
<td></td>
<td>Sacramento, CA 95841</td>
</tr>
<tr>
<td>(404) 226-3714</td>
<td></td>
<td>(916) 924-8037</td>
</tr>
<tr>
<td>Data I/O, Inc.</td>
<td>OTP Programmer</td>
<td>Orion Instruments</td>
</tr>
<tr>
<td>10525 Willows Road N.E.</td>
<td>(Z86E21)</td>
<td>180 Independence Dr.</td>
</tr>
<tr>
<td>P.O. Box 97046</td>
<td></td>
<td>Menlo Park, CA 94025</td>
</tr>
<tr>
<td>Redmond, WA 98073-9746</td>
<td></td>
<td>(415) 327-8800</td>
</tr>
<tr>
<td>(206) 867-6829</td>
<td></td>
<td>Production Languages Corp.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.O. Box 109</td>
</tr>
<tr>
<td>iSystems GmbH</td>
<td>Emulator</td>
<td>Weatherford, TX 76086-0109</td>
</tr>
<tr>
<td>Einsteinstr. 5</td>
<td></td>
<td>(817) 599-8363</td>
</tr>
<tr>
<td>W8050 Dachau, Germany</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(49) 8131-25085</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J K Engineering</td>
<td>Emulator</td>
<td>Pseudo Corp.</td>
</tr>
<tr>
<td>Tong Lee Bldg. #08-03</td>
<td></td>
<td>Newport News, VA 23606</td>
</tr>
<tr>
<td>Singapore 1334</td>
<td></td>
<td>(804) 873-1947</td>
</tr>
<tr>
<td>011-65-7448418</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laboratory Microsystems</td>
<td>Forth Compiler</td>
<td>Signum Systems</td>
</tr>
<tr>
<td>12555 West Jefferson Blvd.</td>
<td></td>
<td>171 E. Thousand Oaks Blvd.</td>
</tr>
<tr>
<td>Los Angeles, CA 90060</td>
<td></td>
<td>Thousand Oaks, CA 91360</td>
</tr>
<tr>
<td>(213) 306-7412</td>
<td></td>
<td>(805) 371-4608</td>
</tr>
<tr>
<td>Company</td>
<td>Product</td>
<td>Company</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>--------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Smart Access, Inc.</td>
<td>OTP Programmer (Z86E21, Z86E22)</td>
<td>Western Wares</td>
</tr>
<tr>
<td>124 Robin Road</td>
<td></td>
<td>P.O. Box C</td>
</tr>
<tr>
<td>Altamonte Springs, FL 32701</td>
<td></td>
<td>Norwood, CO 81423</td>
</tr>
<tr>
<td>(407) 331-4724</td>
<td></td>
<td>(303) 327-4898</td>
</tr>
<tr>
<td>Software Development Systems</td>
<td>Assembler</td>
<td>Wytec</td>
</tr>
<tr>
<td>4248 Belle Aire Lane</td>
<td></td>
<td>185C East Lake Street Ste. 140</td>
</tr>
<tr>
<td>Downers Grove, IL 60515</td>
<td></td>
<td>Bloomingdale, IL 60108</td>
</tr>
<tr>
<td>(800) 448-7733</td>
<td></td>
<td>(708) 894-1440</td>
</tr>
<tr>
<td>Software Science</td>
<td>Z8® Prototyping System</td>
<td>2500AD Software, Inc.</td>
</tr>
<tr>
<td>3750 Round Bottom Road</td>
<td></td>
<td>109 Brookdale Ave.</td>
</tr>
<tr>
<td>Cincinnati, OH 45244</td>
<td></td>
<td>P.O. Box 480</td>
</tr>
<tr>
<td>(513) 561-2060</td>
<td></td>
<td>Buena Vista, CO 81211</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(719) 395-8683, or 800-843-8144</td>
</tr>
</tbody>
</table>
# Telephone Answering Devices

## Block Diagram

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Z8600/Z8611</th>
<th>Z86C30/E30/C31/E31</th>
<th>Z89C65/Z89C66</th>
<th>Z89165/Z89166</th>
</tr>
</thead>
</table>

## Description

- **Z8** NMOS (CCP®)
  - Z8600 = 2K ROM
  - Z8611 = 4K ROM

- **Z8** Consumer Controller Processor (CCP®)
  - Z86C30 = 28-Pin, 4K ROM
  - Z86C31 = 28-Pin, 2K ROM
  - Z86C40 = 40-Pin, 4K ROM
  - Z86E30, Z86E31, Z86E40 = OTP Version

- Telephone Answering Controller
  - Z89C66 = ROMLess with 31 I/O Pins

- Low-Cost DTAD Controller
  - Z89166 = ROMLess with 31 I/O Pins

## Process/Speed

- NMOS: 8,12 MHz
- CMOS: 12 MHz
- CMOS: 20 MHz
- CMOS: 20 MHz

## Features

- 2K/4K ROM
- 128 Bytes RAM
- 22/32 I/O Lines
- On-Chip Oscillator
- Two Counter/Timers
- Six Vectored, Priority Interrupts
- UART (Z8611 Only)
- 4K ROM/236 RAM
- Two Standby Modes
- Two Counter/Timers
- ROM/RAM Protect
- Four Ports (Z86C40/E40)
- Three Ports (Z86C30/E30/C31/E31)
- Low-Voltage Protection
- Two Analog Comparators
- Low-EMI Option
- Watch-Dog Timer (WDT)
- Auto Power-On Reset
- Low-Power Option
- 24K ROM (Z89C65 Only)
- 16-Bit DSP
- 4K Word ROM
- 8-Bit A/D with Automatic Gain Control (AGC)
- DTMF Macro Available
- LPC Macro Available
- 10-Bit PWM D/A
- Other DSP Software Options Available
- 47 I/O Pins (Z89C65 Only)
- 24K ROM (Z89165 Only)
- 16-Bit DSP
- 6K Word DSP ROM
- 8-Bit A/D with Automatic Gain Control (AGC)
- DTMF Macro Available
- LPC Macro Available
- 10-Bit PWM D/A
- Other DSP Software Options Available
- 47 I/O Pins (Z89165 Only)

## Package

- 28-Pin DIP
- 40-Pin DIP
- 44-Pin PLCC
- 68-Pin PLCC
- 80-Pin QFP

## Support Products

- Z86C1200ZEM - Emulator
- Z8660000ZCO - Evaluation Board
- Z8660000ZDP - Adaptor Kit
- Z86CCP00ZEM - Emulator
- Z86CCP00ZAC - Emulator
- Z86C5000ZEM - Emulator
- Z86E3000ZDP - Adaptor Kit
- Z86E4000ZDP - Program Adaptor Kit
- Z89C6501ZEM - Emulator
- Z89C6500ZDB - Emulator
- Z89C6500ZCO - Evaluation Board
<table>
<thead>
<tr>
<th>BLOCK DIAGRAM</th>
<th>PART NUMBER</th>
<th>DESCRIPTION</th>
<th>PROCESS/SPEED</th>
<th>FEATURES</th>
<th>PACKAGE</th>
<th>SUPPORT PRODUCTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Z89C67/Z89C68/Z89C69</td>
<td>Telephone Answering Controller</td>
<td>CMOS: 20 MHz</td>
<td>16-Bit DSP, 6K Word ROM, DTMF Macro Available, LPC Macro Available, 10-Bit PWM D/A, Other DSP Software Options Available, ARAM/DRAM-ROM Controller and Interface, Dual CODEC Interface, 43 I/O (Z89C67 Only)</td>
<td>84-Pin PLCC</td>
<td>Z89C5900ZEM - Emulator, Z89C6700ZEM - Emulator, Z89C6700ZDB - Emulator, Z8916902ZCO - Evaluation Board</td>
</tr>
<tr>
<td></td>
<td>Z89167/Z89168</td>
<td>Enhanced Telephone Answering Controller</td>
<td>CMOS: 24 MHz</td>
<td>24K ROM (Z89167 Only), 16-Bit DSP, 8K Word ROM, DTMF Macro Available, LPC Macro Available, 10-Bit PWM D/A, Other DSP Software Options Available, ARAM/DRAM-ROM Controller and Interface, Dual CODEC Interface, 43 I/O (Z89167 Only)</td>
<td>84-Pin PLCC</td>
<td>Z89C5900ZEM - Emulator, Z89C6700ZEM - Emulator, Z89C6700ZDB - Emulator, Z8916902ZCO - Evaluation Board</td>
</tr>
<tr>
<td></td>
<td>Z89169</td>
<td>Enhanced Telephone Answering Controller</td>
<td>CMOS: 24 MHz</td>
<td>32K ROM, 16-Bit DSP, 8K Word ROM, DTMF Macro Available, LPC Macro Available, 10-Bit PWM D/A, Other DSP Software Options Available, ARAM/DRAM-ROM Controller and Interface, Dual CODEC Interface, 43 I/O</td>
<td>84-Pin PLCC</td>
<td>Z89C5900ZEM - Emulator, Z89C6700ZEM - Emulator, Z89C6700ZDB - Emulator, Z8916902ZCO - Evaluation Board</td>
</tr>
</tbody>
</table>
## Zilog TV/Video Products

### Superintegration™ Products Guide

<table>
<thead>
<tr>
<th>Block Diagram</th>
<th>16K/8K ROM</th>
<th>6K ROM</th>
<th>CHAR ROM</th>
<th>1K/6K ROM</th>
<th>2K/8K/16K ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>16K/8K ROM</strong></td>
<td>4K CHAR ROM</td>
<td>3K CHAR ROM</td>
<td>COMMAND INTERPRETER</td>
<td>2K CPU RAM</td>
<td>Z8 CPU</td>
</tr>
<tr>
<td><strong>Z8 CPU</strong></td>
<td><strong>RAM</strong></td>
<td><strong>OSD</strong></td>
<td><strong>PWM</strong></td>
<td><strong>WDT</strong></td>
<td><strong>P2</strong></td>
</tr>
<tr>
<td><strong>OSD</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>13 TIMER</strong></td>
<td><strong>5 WDT PORTS</strong></td>
<td><strong>7 TIMER</strong></td>
<td><strong>3 WDT PORTS</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Part Number

| **Z86C27/127/917/E47** | **Z86227** | **Z86128/Z86228/Z86129** | **Z86L06/Z86L29** | **Z86L70/71/72/73/74 75/76/77/78** |

### Description

- Digital Television Controller (DTC™) Television, VCRs, and Cable
- Z86E47 = OTP Version
- Standard DTC™ Features with Reduced ROM, RAM, PWM Outputs for Greater Economy
- Z86128/228 = Line 21 Closed Caption Controller (L2IC™)
- Z86129/229 = Line 21 Closed Caption and EDS Controller
- Z86L06 = Low-Voltage CMOS Consumer Controller Processor
- Z86L29 = 6K Infrared Remote Controller
- Zilog Infrared Remote Controllers (ZIRC™) for IR Remote/Battery Operated Applications Ranging in ROM: L70=2K, L71=4K, L72&78=16K, L73&74=32K, L75=4K, L76=12K, L77=24K

### PROCESS/SPEED

- CMOS: 4 MHz
- CMOS: 12 MHz
- Low-Voltage CMOS: 8 MHz
- Low-Voltage CMOS: 8 MHz

### FEATURES

- 8K/16K/OTP ROM
- 256 Byte RAM
- 16x7-Bit Video RAM
- On-Screen Display (OSD) Video Controller
- Programmable
  - Color
  - Size
  - Position Attributes
- 13 PWMs for D/A Conversion
- 128-Character Set
- 4Kx6-Bit Char. Gen. ROM
- Watch-Dog Timer (WDT)
- Low-Voltage Protection
- Five Ports/36 Pins
- Two Standby Modes
- Low-EMI Mode
- 6K ROM, 256 Byte RAM
- 120x7-Bit Video RAM
- OSD On-Board Programmable
  - Color
  - Size
  - Position Attributes
- 7 PWMs
- 96 Character Set
- 3Kx6-Bit Char. Gen. ROM
- Watch-Dog Timer (WDT)
- Low-Voltage Protection
- Three Ports/20 Pins
- Two Standby Modes
- Low-EMI Mode
- Conforms to FCC Line 21 Format
- Parallel or Serial Modes
- Stand-Alone Operation
- On-Board Data Sync and Slicer
- On-Board Character Generator
  - Color
  - Blinking
  - Italic
  - Underline
  - Extended Data Services
- 1K ROM and 6K ROM
- Watch-Dog Timer (WDT)
- Two Analog Comparators with Output Option
- Two Standby Modes
- Two Counter/Timers
- Auto Power-On Reset
- 2V Operation
- RC Oscillator Option
- Low-Voltage Protection
- High-Current Drivers (2, 4)
- Watch-Dog Timer (WDT)
- Two Analog Comparators with Output Option
- Two Standby Modes
- Two Counter/Timers
- Auto Pulse
- Reception/Generation
- Auto Power-On Reset
- 2V Operation
- RC Oscillator Option
- Low-Voltage Protection
- High-Current Drivers
  - Three OTP Versions Available
    - Z86E70/71/72/73/74

### PACKAGE

- 64-Pin DIP
- 40-Pin DIP
- 18-Pin DIP
- 18-Pin SOIC
- Z86L71=20-Pin DIP/SOIC
- Z86L70/75=18-Pin DIP, SOIC
- Z86L72/76, 77=40, 44-Pin DIP, PLCC, QFP
- Z86L74=64/68-Pin

### SUPPORT PRODUCTS

- Z86C2700ZCO - Evaluation Board
- Z86C2700ZDB - Emulator
- Z86C2700ZEM - Emulator
- Z86C2700ZDZ - Emulator
- Z86C2700ZCO - Evaluation Board
- Support Documentation Provided with the device
- Z86C5000ZEM - Emulator
- Z86L7200TSC - Emulator
- Z86L7200ZEM - Emulator
- Z86L7100ZDB - Emulator
<table>
<thead>
<tr>
<th>PART NUMBER</th>
<th>DESCRIPTION</th>
<th>PROCESS/SPEED</th>
<th>FEATURES</th>
<th>PACKAGE</th>
<th>SUPPORT PRODUCTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z86C40/Z86E40</td>
<td>Z8® Consumer Controller Processor (CCP)</td>
<td>CMOS: 12 MHz</td>
<td>4K ROM, 236 RAM, Two Standby Modes, Two Counter/Timers, RAM Protect, RAM Protect, Four Ports, Low-Voltage Protection, Two Analog Comparators, Low-EMI Mode, Watch-Dog Timer (WDT), Auto Power-On Reset, Low-Power Option</td>
<td>40-Pin DIP</td>
<td>Z86C5000ZEM - Emulator</td>
</tr>
<tr>
<td></td>
<td>Z86C5000ZEM - Emulator</td>
<td></td>
<td></td>
<td>44-Pin PLCC</td>
<td>Z86CCP00ZEM - Emulator</td>
</tr>
<tr>
<td></td>
<td>Z86E4000ZEM - Emulator</td>
<td></td>
<td></td>
<td></td>
<td>Z86E4000ZEM - Adaptor Kit</td>
</tr>
<tr>
<td></td>
<td>Z86E4000ZDV - Adaptor Kit</td>
<td></td>
<td></td>
<td></td>
<td>Z86E4000ZDV - Adaptor Kit</td>
</tr>
<tr>
<td>Z86C61/Z86C62</td>
<td>Z8MCU with Expanded I/Os</td>
<td>CMOS: 16, 20 MHz</td>
<td>16K ROM, Full-Duplex UART, Two Standby Modes (STOP and HALT), Two Counter/Timers, ROM Protect Option, RAM Protect Option, Pin Compatible to Z86C21, Z86C61 = Four Ports, Z86C62 = Seven Ports</td>
<td>40-Pin DIP</td>
<td>Z86C5000ZEM - Emulator</td>
</tr>
<tr>
<td></td>
<td>Z86C5000ZEM - Emulator</td>
<td></td>
<td></td>
<td>44-Pin PLCC</td>
<td>Z86CCP00ZEM - Emulator</td>
</tr>
<tr>
<td></td>
<td>Z86E4000ZEM - Emulator</td>
<td></td>
<td></td>
<td></td>
<td>Z86E4000ZEM - Adaptor Kit</td>
</tr>
<tr>
<td></td>
<td>Z86E4000ZDV - Adaptor Kit</td>
<td></td>
<td></td>
<td></td>
<td>Z86E4000ZDV - Adaptor Kit</td>
</tr>
<tr>
<td>Z83300/02/04/14</td>
<td>Advanced TV Controller with Closed Caption Decoder (CCD), StarSight®, OSD for TV, VCR, Cable, Satellite</td>
<td>CMOS: 12 MHz</td>
<td>12K/16K/24K ROM, DSP Core, RAM PC, Programmable OSD, PC: 7 PWM, 3-Channel ADC, Watch-Dog Timer (WDT), Auto Power-On Reset, Low-Power Option</td>
<td>40-Pin SDIP</td>
<td>Z86C5000ZEM - Emulator</td>
</tr>
<tr>
<td></td>
<td>Z83301/03/05/13 Advanced TV Controller with Closed Caption Decoder (CCD), StarSight®, OSD for TV, VCR, Cable, Satellite</td>
<td></td>
<td></td>
<td>52-Pin SDIP</td>
<td>Z86C5000ZEM - Emulator</td>
</tr>
<tr>
<td></td>
<td>Z83331/Z83336 Advanced TV Controller with Closed Caption Decoder (CCD), StarSight®, OSD for TV, VCR, Cable, Satellite</td>
<td></td>
<td></td>
<td>42-Pin SDIP</td>
<td>Z86C5000ZEM - Emulator</td>
</tr>
</tbody>
</table>
### Discrete Z8® Microcontroller

<table>
<thead>
<tr>
<th>PART NUMBER</th>
<th>DESCRIPTION</th>
<th>PROCESS/SPEED</th>
<th>FEATURES</th>
<th>PACKAGE</th>
<th>SUPPORT PRODUCTS</th>
</tr>
</thead>
</table>
| Z86C03      | Consumer Controller Processor (CCP°) with 512 Byte ROM | CMOS: 8 MHz | ■ 512 Byte ROM  
■ 64 Byte RAM  
■ Two Standby Modes  
■ One Counter/Timer  
■ ROM Protect  
■ Two Analog Comparator  
■ Auto Power-On Reset  
■ Low-Voltage Protection  
■ 14 I/O  
■ RC Oscillator Option  
■ Low-Noise Option | 18-Pin DIP  
18-Pin SOIC | Z86CCP00ZEM - Emulator  
Z86CCP00ZAC - Emulator |
| Z86C04/Z86E04 | Z86C04 = 8-Bit Low Cost 1 Kbyte ROM MCU  
Z86E04 = OTP Version | CMOS: 8 MHz | ■ 1 Kbyte ROM  
■ 128 Byte RAM  
■ Two Standby Modes  
■ Two Counter/Timer  
■ ROM Protect  
■ Two Analog Comparator  
■ Auto Power-On Reset  
■ Low-Voltage Protection (ROM Only)  
■ 14 I/O  
■ Low-Noise Option | 18-Pin DIP  
18-Pin SOIC | Z86CCP00ZCO - Evaluation Board  
Z86CCP00ZDP - Adaptor Kit  
Z86CC120ZEM - Emulator  
Z86CC120ZPD - Adaptor Kit  
Z86CCP00ZEM - Emulator  
Z86CCP00ZAC - Emulator |
| Z86C06      | Consumer Controller Processor (CCP°) with 1 Kbyte ROM | CMOS: 12 MHz | ■ 1 Kbyte ROM  
■ 128-Byte RAM  
■ Two Standby Modes  
■ Two Counter/Timer  
■ ROM Protect  
■ Two Analog Comparator  
■ Auto Power-On Reset  
■ Low-Voltage Protection (ROM Only)  
■ 14 I/O  
■ RC Oscillator Option  
■ Serial Peripheral Interface (SPI) | 18-Pin DIP  
18-Pin SOIC | Z86E0600ZDP - Adaptor Kit  
Z86CCP00ZEM - Emulator  
Z86CCP00ZAC - Emulator |
## Discrete Z8® Microcontroller

<table>
<thead>
<tr>
<th>Block Diagram</th>
<th>2K ROM</th>
<th>4K ROM</th>
<th>2K ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Z8® CPU</td>
<td>Z8® CPU</td>
<td>Z8® CPU</td>
</tr>
<tr>
<td></td>
<td>WDT</td>
<td>128 RAM</td>
<td>WDT</td>
</tr>
<tr>
<td></td>
<td>P0</td>
<td>P2</td>
<td>P0</td>
</tr>
</tbody>
</table>

### Part Number

<table>
<thead>
<tr>
<th>2K ROM</th>
<th>4K ROM</th>
<th>2K ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z86C08/Z86E08</td>
<td>Z86C30/Z86E30</td>
<td>Z86C31/Z86E31</td>
</tr>
</tbody>
</table>

### Description

- **Z86C08** = Z8® MCU with 2 Kbyte ROM
- **Z86E08** = OTP Version
- **Z86C30** = Z8® (CCP*) with 4 Kbyte ROM
- **Z86E30** = OTP Version
- **Z86C31** = 8-Bit MCU with 2 Kbyte ROM
- **Z86E31** = OTP Version

### Process/Speed

- CMOS: 12 MHz
- CMOS: 12 MHz
- CMOS: 8 MHz

### Features

- 2 Kbyte ROM
- 128 Byte RAM
- Two Standby Modes
- Two Counter/Timer
- ROM Protect
- Two Analog Comparators
- Auto Power-On Reset
- Low-Voltage Protection (ROM Only)
- 14 I/O
- Low-Noise Option
- 4 Kbyte ROM
- 236 Byte RAM
- Two Standby Modes
- Two Counter/Timer
- ROM Protect
- Two Analog Comparators
- Auto Power-On Reset
- Low-Voltage Protection (ROM Only)
- 24 I/O
- RC Oscillator Option
- Low-Noise Option
- 2 Kbyte ROM
- 128 Byte RAM
- Two Standby Modes
- Two Counter/Timer
- ROM Protect
- Two Analog Comparators
- Auto Power-On Reset
- Low-Voltage Protection (ROM Only)
- 24 I/O
- RC Oscillator Option
- Low-Noise Option

### Package

- 18-Pin DIP
- 18-Pin SOIC
- 28-Pin DIP
- 28-Pin PLCC

### Support Products

- Z86C0800ZCO - Evaluation Board
- Z86C0800ZDP - Adaptor Kit
- Z86C1200ZEM - Emulator
- Z86C1200ZDP - Adaptor Kit
- Z86CCP00ZEM - Emulator
- Z86CCP00ZAC - Emulator
- Z86C3000ZDP - Adaptor Kit
- Z86C5000ZEM - Emulator
- Z86C5000ZPD - Emulator Pod
- Z86CCP00ZEM - Emulator
- Z86CCP00ZAC - Emulator
### Block Diagram

<table>
<thead>
<tr>
<th></th>
<th><strong>Bus IF</strong></th>
<th><strong>DAC IF</strong></th>
<th><strong>Sample Rate Generator</strong></th>
<th><strong>Sound Blaster Command Set Interpreter</strong></th>
<th><strong>MIDI Interface</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DAC</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>512 RAM</strong></td>
<td>4K ROM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>16-BIT MAC</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Peripherals Interface</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Part Numbers

<table>
<thead>
<tr>
<th><strong>Part Number</strong></th>
<th><strong>Description</strong></th>
<th><strong>Process/Speed</strong></th>
<th><strong>Features</strong></th>
<th><strong>Package</strong></th>
<th><strong>Support Products</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Z86321</td>
<td>8-Bit Digital Audio Processor</td>
<td>CMOS: 12 MHz</td>
<td>• Sound Blaster™ Compatible • ADPCM Decompression • 8-Bit DAC Interface • Successive Approximation ADC Algorithm • MIDI Interface</td>
<td>40-Pin DIP</td>
<td>Support Documentation Provided with Device</td>
</tr>
<tr>
<td>Z89320</td>
<td>16-Bit Digital Signal Processor</td>
<td>CMOS: 10 MHz</td>
<td>• 16-Bit Multiply/Accumulate • 100 ns • 512 Word RAM • 4K Word RAM • Peripherals Interface Bus • 74 Instruction Set</td>
<td>40-Pin DIP</td>
<td>Z89C0000ZEM - Emulator</td>
</tr>
<tr>
<td>Z89321/Z89371</td>
<td>16-Bit Digital Signal Processor</td>
<td>CMOS: 20 MHz</td>
<td>• 16-Bit Multiply/Accumulate • 50 μs • 512 Word RAM • 4K Word ROM • Peripherals Interface Bus • CODEC Interface</td>
<td>40-Pin DIP</td>
<td>Z8937100ZEM - Emulator</td>
</tr>
<tr>
<td>Z5380</td>
<td>Small Computer System Interface (SCSI)</td>
<td>Clock: 1.5 Mb/s</td>
<td>• Compatible 5380 Pin-out • CMOS • Asynchronous VF Supports 1.5 Mb/s • 48 mA Drivers • Arbitration Support • Support Normal or Block Mode DMA</td>
<td>40-Pin DIP</td>
<td>Support Documentation Provided with Device</td>
</tr>
</tbody>
</table>

### Support Documentation

- Z86321 Emulator
- Z89320 Emulator
- Z5380 Emulator

*SoundBlaster™ is a Trademark of Creative Labs, Inc.*
## MULTIMEDIA/PC AUDIO

<table>
<thead>
<tr>
<th>Block Diagram</th>
<th>Multimedia/PC Audio</th>
<th>Wireless Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISA Bus I/F</td>
<td>ISAbus/F OMA Interface Logic</td>
<td>Modulator</td>
</tr>
<tr>
<td>DMA Logic</td>
<td>Interrupt Logic</td>
<td>Clock &amp; Synchronizer</td>
</tr>
<tr>
<td>Registers</td>
<td>Register Logic</td>
<td>Receiver &amp; Transmitter</td>
</tr>
<tr>
<td>Host I/F</td>
<td>Command Control</td>
<td>FSK Modulator</td>
</tr>
<tr>
<td>DMA Control Logic</td>
<td>ROM I/F</td>
<td>Modem Control Logic</td>
</tr>
<tr>
<td>Parameter</td>
<td>Acquire Bank</td>
<td>DFF Encoder</td>
</tr>
<tr>
<td>Logic</td>
<td>Waveform Data Input</td>
<td>Matched Filter</td>
</tr>
<tr>
<td>Bus Interface Control</td>
<td>Direct Converter</td>
<td>Direct Converter</td>
</tr>
<tr>
<td>Data Transfer Control</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### PART NUMBER

- **Z53C80**
- **Z89341/Z89342**
- **Z2000**
- **Z87000**

### DESCRIPTION

- **SCSI Adaptor**
- **Wave Synthesis Chip Set**
- **Spread Spectrum Burst Processor**
- **Cordless Phone Transceiver/Controller**

### SPEED MHz

- **Clock: 3 Mb/s**
- **CMOS: 36 MHz**
- **CMOS: 45 MHz**
- **CMOS: 16.384 MHz**

### FEATURES

- **ANSI X3, 131-1986 Standard**
- **DMA or Programmed I/O Data Transfers**
- **Asynchronous Interface Support**
- **3 Mb/s**
- **ISA Bus I/F**
- **Glitch Eater**
- **4-Channel**
- **16-Bit Linear**
- **PCM Sound Generator**
- **Sampling Rates 20 kHz to 44.1 kHz**
- **Support 16-, 18-, and 20-Bit DAC**
- **Audio Bandwidth 0 Hz to 20,000 Hz**
- **Direct Interface with PC ISA Bus**
- **Direct Support 4Mx16 ROM**
- **Operates up to 11.1264 Mchips Second in Transmit and Receive Modes**
- **Maximum Data Rate of 2.048 Mbps in Conformance with FCC Regulations**
- **Supports DifferentiablyEncoded BPSK or QPSK Modulation**
- **Full- or Half-Duplex Operation for FDD or TDD Implementations**
- **Two Independent PN Sequences**
- **Power Management Features**
- **Supports 900 MHz Spread Spectrum Cordless Phone Design**
- **Adaptive Frequency Hopping**
- **Transmit Power Control**
- **Bus Interface to ADPCM Processor**
- **12K Words of RAM for Transceiver and Phone Control Software**
- **32 Pins of Program I/O**
- **ROM Code, OTP and ICEBOX™ Version to be Available Q3/94**

### PACKAGE

- **40-Pin DIP**
- **44-Pin PLCC**
- **84-Pin PLCC**
- **100-Pin VQFP**
- **84-Pin PLCC**

### SUPPORT PRODUCTS

- **Support Documentation Provided with Device**
- **Support Documentation Provided with Device**
- **Z02000002CO - Evaluation Board**
- **Z870000ZEM - Emulator**

*Z2000 is sold under license from Stanford Telecommunications, Inc. ASIC and Custom Products Division*
<table>
<thead>
<tr>
<th>PART NUMBER</th>
<th>DESCRIPTION</th>
<th>PROCESS/SPEED</th>
<th>FEATURES</th>
<th>PACKAGE</th>
<th>SUPPORT PRODUCTS</th>
</tr>
</thead>
</table>
| Z8615       | Keyboard MCU | NMOS: 4, 5 MHz | • 4K ROM  
  • 124-Byte RAM  
  • 32 I/O Lines  
  • Two Counter/Timers  
  • Watch-Dog Timer (WDT)  
  • RC Oscillator  
  • Dedicated Row Column Pins  
  • Data/Clock Pins  
  • Direct Connect LED Pins | 40-Pin DIP  
  44-Pin PLCC | Z0861500ZCO - Evaluation Board  
 Z86C1202ZEM - Emulator  
 Z0861500ZDP - Adaptor Kit |
| Z8614/Z8602 | Z8602 = 2K ROM Keyboard MCU  
 Z8614 = 4K ROM Keyboard MCU | NMOS: 4 MHz | • 4K ROM  
  • 124-Byte RAM  
  • 32 I/O Lines  
  • Two Counter/Timers  
  • Dedicated Row Column Pins  
  • Data/Clock Pins  
  • Direct Connect LED Pins | 40-Pin DIP  
  44-Pin PLCC | Z0860200ZCO - Evaluation Board  
 Z86C1202ZEM - Emulator  
 Z0860200ZDP - Adaptor Kit |
| Z86E23      | Keyboard OTP MCU | CMOS: 4 MHz | • 8K ROM  
  • 256-Byte RAM  
  • 32 I/O Lines  
  • Two Counter/Timers  
  • Dedicated Row Column Pins | 40-Pin DIP  
  44-Pin PLCC | Z086E2300ZCO - Evaluation Board  
 Z86C1202ZEM - Emulator  
 Z086E2300ZDP - Adaptor Kit |
| Z86C17      | Mouse MCU | CMOS: 4 MHz | • 2K ROM  
  • 124-Byte RAM  
  • 14 I/O Lines  
  • Two Counter/Timers  
  • Dedicated Opto-Transistor Pins  
  • Integrated Pull-up Resistors  
  • Power-Down Modes  
  • Power-On Reset (POR)  
  • Watch-Dog Timer (WDT) | 18-Pin DIP  
  18-Pin SOIC | Z86C1700ZEM - Emulator |
<table>
<thead>
<tr>
<th>BLOCK DIAGRAM</th>
<th>2K ROM</th>
<th>1K ROM</th>
<th>4K ROM</th>
<th>4K ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z86C08/Z86C07/Z86E08</td>
<td>Z86D04/Z86E04</td>
<td>Z89321/Z89371</td>
<td>Z86C30/Z86E30</td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>CMOS: 4, 8, 12 MHz</td>
<td>CMOS: 4 MHz</td>
<td>CMOS: 15, 20 MHz</td>
<td>CMOS: 8, 12 MHz</td>
</tr>
<tr>
<td>FEATURES</td>
<td>18-Pin DIP</td>
<td>Z86C1200ZEM - Emulator</td>
<td>Z86C1200ZEM - Emulator</td>
<td>Z86C5000ZEM - Emulator</td>
</tr>
<tr>
<td>PACKAGE</td>
<td>18-Pin SOIC</td>
<td>Z8937100ZEM - Emulator</td>
<td>Z8937100TSC - Emulator</td>
<td>Z86C5000ZEM - Emulator</td>
</tr>
<tr>
<td>SUPPORT PRODUCTS</td>
<td>18-Pin DIP</td>
<td>Z86C1200ZEM - Emulator</td>
<td>Z86C5000ZEM - Emulator</td>
<td>Z86C5000ZEM - Emulator</td>
</tr>
<tr>
<td>Z86C1200ZEM - Emulator</td>
<td>Z86C1200ZEM - Emulator</td>
<td>Z86C1200ZEM - Emulator</td>
<td>Z86C1200ZEM - Emulator</td>
<td></td>
</tr>
<tr>
<td>Z86C5000ZEM - Emulator</td>
<td>Z86C5000ZEM - Emulator</td>
<td>Z86C5000ZEM - Emulator</td>
<td>Z86C5000ZEM - Emulator</td>
<td></td>
</tr>
<tr>
<td>PART NUMBER</td>
<td>DESCRIPTION</td>
<td>PROCESS/SPEED</td>
<td>FEATURES</td>
<td>PACKAGE</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
<td>------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Z84C01</td>
<td>Z80® CPU with Clock Generator/Clock</td>
<td>CMOS: 10 MHz</td>
<td>■ Clock Generator/Controller</td>
<td>44-Pin QFP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Four Power Down Modes</td>
<td>44-Pin PLCC</td>
</tr>
<tr>
<td>Z84C90</td>
<td>Killer I/O (Three Z80® Peripherals)</td>
<td>CMOS: 8, 10, 12 MHz</td>
<td>■ Serial Input/Output (SIO)</td>
<td>84-Pin PLCC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Counter/Timer Circuit (CTC)</td>
<td>80-Pin QFP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Plus Eight I/O Lines</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Three 8-Bit Ports</td>
<td></td>
</tr>
<tr>
<td>Z84013/Z84C13</td>
<td>Intelligent Peripheral Controller</td>
<td>Z84013 = CMOS: 6, 10 MHz</td>
<td>■ Serial Input/Output (SIO)</td>
<td>84-Pin PLCC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z84C13 = CMOS: 6, 10 MHz</td>
<td>■ Counter/Timer Circuit (CTC)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Watch-Dog Timer (WDT)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Clock Generator Circuit (CGC)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Wait State Generator (WSG)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Power-On Reset (POR)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Two Chip Selects</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Evaluation Mode</td>
<td></td>
</tr>
<tr>
<td>Z84015/Z84C15</td>
<td>Enhanced Intelligent Peripheral</td>
<td>Z84015 = CMOS: 6, 10 MHz</td>
<td>■ Serial Input/Output (SIO)</td>
<td>100-Pin QFP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z84C15 = CMOS: 16 MHz</td>
<td>■ Counter/Timer Circuit (CTC)</td>
<td>100-Pin QFP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Watch-Dog Timer (WDT)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Clock Generator Circuit (CGC)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Four Power-Down Modes</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Power-On Reset</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Two Chip Selects</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ 32-Bit CRC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Wait State Generator (WSG)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Evaluation Mode</td>
<td></td>
</tr>
</tbody>
</table>

**Z80® EMBEDDED CONTROLLERS**

**SUPERINTEGRATION™ PRODUCTS GUIDE**
<table>
<thead>
<tr>
<th>PART NUMBER</th>
<th>DESCRIPTION</th>
<th>PROCESS/SPEED</th>
<th>FEATURES</th>
<th>PACKAGE</th>
<th>SUPPORT PRODUCTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z80180/Z8S180/Z8L180</td>
<td>High-Performance Z80® CPU with Peripherals</td>
<td>Z80180 = CMOS: 6, 8, 10 MHz&lt;br&gt;Z8S180 = CMOS: 16 MHz&lt;br&gt;Z8L180 = CMOS: 20, 33 MHz</td>
<td>▪ Enhanced Z80® CPU&lt;br▪ 1 Mbyte MMU&lt;br▪ 2 DMAs&lt;br▪ 2 UARTs with Baud Rate Generators&lt;br▪ C/Serial I/O Port Oscillator&lt;br▪ Z8S180 Includes;&lt;br- Power-Down&lt;br- Programable EMI&lt;br- Divide-By-One&lt;br- Clock Option&lt;br- 3.3V and 5V Version</td>
<td>64-Pin DIP&lt;br68-Pin PLCC&lt;br80-Pin QFP</td>
<td>Z8S18000ZCO - Evaluation Board&lt;brZEP2MIP0001 - EPM® Manual</td>
</tr>
<tr>
<td>Z80181</td>
<td>Smart Access Controller</td>
<td>CMOS: 10, 12 MHz</td>
<td>▪ Complete Z180® plus SCC/2&lt;br- Counter/Timer Circuit&lt;br- 16 I/O Lines&lt;br- Emulation Mode</td>
<td>100-Pin QFP</td>
<td>Z8018100ZCO - Evaluation Board&lt;brZEP2MIP0002 - EPM® Manual</td>
</tr>
<tr>
<td>Z80182/Z8L182</td>
<td>Zilog Intelligent Peripheral (ZIP™)&lt;brZ8L182 = Low-Voltage Version</td>
<td>Z80182 = CMOS: 16, 33 MHz&lt;brZ8L182 = CMOS: 20 MHz</td>
<td>▪ Static Version of Z180® plus ESCC&lt;br- (2 Channels of Z85230 with 32-Bit CRC Not Available for 16 MHz)&lt;br- 16550 MIMIC&lt;br- 24 Parallel I/O&lt;br- Emulation Mode&lt;br- 3.3V and 5V Version</td>
<td>100-Pin QFP&lt;br100-Pin VQFP</td>
<td>Z8018200ZCO - Evaluation Board&lt;brZEP2MIP0003 - EPM® Manual</td>
</tr>
<tr>
<td>Z80380/Z8L380</td>
<td>Z390® Microprocessor&lt;brZ8L380 = Low-Voltage Z390</td>
<td>Z8L380 = CMOS: 10 MHz&lt;brZ80380 = CMOS: 16, 18 MHz</td>
<td>▪ 16/32-Bit MPU&lt;br▪ Internal 32-Bit Datapaths and ALU&lt;br▪ 2 Clocks/Cycle Instruction Execution up to 4 Gbytes of Linear Addressing&lt;br▪ Enhanced Instruction Set&lt;br▪ 4 Banks of On-Chip Register Files&lt;br▪ Object-Code Compatible with Z80/Z180 Microprocessors up to 6 Programmable Memory Chip Selects&lt;br▪ 3.3V and 5V Version</td>
<td>100-Pin QFP</td>
<td>Z8038000ZCO - Evaluation Board&lt;brZEP2MIP0004 - EPM® Manual</td>
</tr>
<tr>
<td>PART NUMBER</td>
<td>DESCRIPTION</td>
<td>PROCESS/SPEED</td>
<td>SUPPORT PRODUCTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------</td>
<td>------------------------</td>
<td>-------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z89C00</td>
<td>16-Bit Digital Signal Processor</td>
<td>CMOS: 10, 15 MHz</td>
<td>Z89C6500ZEM - Emulator</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Z89C6500ZCC - Emulator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z89120</td>
<td>Zilog Modem/Fax Controller</td>
<td>CMOS: 20 MHz</td>
<td>Z89C6501ZEM - Emulator</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Z89C6500ZDP - Emulator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z89920</td>
<td>Zilog Modem/Fax Controller</td>
<td>CMOS: 20 MHz</td>
<td>Z89C6500ZDB - Emulator</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Z8601700ZCO - Evaluation Board</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86017</td>
<td>PCMCIA Interface Adaptor</td>
<td>CMOS: 20 MHz</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PACKAGE</th>
<th>68-Pin PLCC</th>
<th>68-Pin PLCC</th>
<th>68-Pin PLCC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>60-Pin VQFP</td>
<td></td>
<td>100-Pin VQFP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEATURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-Bit Multiply/Accumulate</td>
</tr>
<tr>
<td>75 ns</td>
</tr>
<tr>
<td>Two Data RAMs (256 Words each)</td>
</tr>
<tr>
<td>4K Word ROM</td>
</tr>
<tr>
<td>64Kx16 Ext. ROM</td>
</tr>
<tr>
<td>16-Bit I/O Port</td>
</tr>
<tr>
<td>74 Instructions</td>
</tr>
<tr>
<td>Most Single Cycle</td>
</tr>
<tr>
<td>Two Conditional Branch Inputs, Two User Outputs</td>
</tr>
<tr>
<td>Library of Macros</td>
</tr>
<tr>
<td>Zero Overhead Pointers</td>
</tr>
<tr>
<td>Z8® with 24 Kbyte ROM</td>
</tr>
<tr>
<td>16-Bit DSP with 4K Word ROM</td>
</tr>
<tr>
<td>8-Bit A/D</td>
</tr>
<tr>
<td>10-Bit D/A (PWM)</td>
</tr>
<tr>
<td>Library of Macros</td>
</tr>
<tr>
<td>47 I/O Pins</td>
</tr>
<tr>
<td>Two Comparators Independent Z8® and DSP Operations Power-Down Mode</td>
</tr>
<tr>
<td>Z8 with 64K External Memory</td>
</tr>
<tr>
<td>DSP with 4K Word ROM</td>
</tr>
<tr>
<td>8-Bit A/D</td>
</tr>
<tr>
<td>10-Bit D/A</td>
</tr>
<tr>
<td>Library of Macros</td>
</tr>
<tr>
<td>47 I/O Pins</td>
</tr>
<tr>
<td>Two Comparators Independent Z8® and DSP Operations Power-Down Mode</td>
</tr>
<tr>
<td>256 Bytes of Attribute Memory</td>
</tr>
<tr>
<td>Five Configuration Registers</td>
</tr>
<tr>
<td>EEPROM Sequencer or SPI Interface</td>
</tr>
<tr>
<td>PCMCIA to I/O, Memory or Both</td>
</tr>
<tr>
<td>PCMCIA to ATA/IDE</td>
</tr>
<tr>
<td>ATA/IDE to ATA/IDE</td>
</tr>
<tr>
<td>3.0V to 5.5V Operation</td>
</tr>
<tr>
<td>8- or 16-Bit Peripheral Support</td>
</tr>
<tr>
<td><strong>PART NUMBER</strong></td>
</tr>
<tr>
<td>----------------</td>
</tr>
</tbody>
</table>
| Z84C15/Z84015  | Enhanced Peripheral Controller | Z84015 = CMOS: 6, 10 MHz  
Z84C15 = CMOS: 16 MHz | Z84015 includes;  
- Enhanced Z80® CPU  
- 2 DMA  
- 1 Mbyte MMU  
- 2 UARTs with baud rate generators  
- C/Serial I/O port oscillator  
- Z8S180 includes;  
  - Power Down  
  - Programmable EMI  
  - Divide-By-One  
  - Clock Option  
  - 3.3V and 5V version | 100-Pin QFP  
100-Pin VQFP | Z84C1500ZCO - Evaluation Board |
| Z80180/Z8S180/Z8L180 | High-Performance Z80® CPU with Peripherals  
Z8S180 = Static Version  
Z8L180 = Low-Voltage Version | Z80180 = CMOS: 6, 8, 10, MHz  
Z8S180 = CMOS: 16 MHz  
Z8L180 = CMOS: 20, 33 MHz | Static Version of Z180® plus ESCC  
(Two channels of Z85230 with 32-bit CRC not available for 16 MHz)  
16550 MIMIC  
24 Parallel I/O  
Emulation Mode  
3.3V and 5V version | 64-Pin DIP  
68-Pin PLCC  
80-Pin QFP | Z8S18000ZCO - Evaluation Board  
Z8018000ZCO - Evaluation Board  
Z8523000ZCO - Evaluation Board |
| Z80182/Z8L182 | Zilog Intelligent Peripheral (ZIP®)  
Z8L182 = Low-Voltage Version | Z80182 = CMOS: 16, 18, 33 MHz  
Z8L182 = CMOS: 20 MHz | Full-Dual-Channel  
SCC Plus Deeper FIFOs:  
- 4 Bytes on Transceivers  
- 8 Bytes on Receivers  
DPLL Counter Per Channel  
Software Compatible to SCC | 100-Pin QFP  
100-Pin VQFP | Z8018200ZCO - Evaluation Board  
Z8018600ZCO - Evaluation Board  
ZEPDC00002 - EPM® Manual |
| Z85230 | Enhanced Serial Communication Controller | CMOS: 8, 10, 16, 20 MHz | - | 40-Pin DIP  
44-Pin PLCC | Z8S18000ZCO - Evaluation Board  
Z8038000ZCO - Evaluation Board  
Z8523000ZCO - Evaluation Board |

**package**
- 100-Pin QFP
- 100-Pin VQFP
- 64-Pin DIP
- 68-Pin PLCC
- 80-Pin QFP

**Support products**
- Z84C1500ZCO - Evaluation Board
- Z8S18000ZCO - Evaluation Board
- Z8018000ZCO - Evaluation Board
- Z8523000ZCO - Evaluation Board
- Z8018600ZCO - Evaluation Board
- ZEPDC00002 - EPM® Manual
## Serial Communications

### Block Diagram

- **SCC**

### Part Number

- **Z8030/Z80C30**
- **Z8530/Z85C30**

### Description

**Serial Communication Controller**
- Z8030/Z80C30 = Multiplexed Bus
- Z8530/Z85C30 = Non-Multiplexed Bus

**Enhanced Serial Communication Controller**
- Z8230/Z80230 = Dual Channel
- Z85233 = Single Channel

**Integrated Serial Communication Controller**

### Process/Speed

**Z8030/Z8530 = NMOS:** 4, 6, 8 MHz

**Z80C30/Z85C30 = CMOS:** 8, 10, 16 MHz

- Clocks: 2, 2.5, 4 MHz

### Features

- Two Independent Full-Duplex Channels
- Enhanced DMA Support:
  - 10x19 Status FIFO
  - 14-Bit Byte Counter
  - NRZ/NRZI/FM Encoding Modes
- Full Dual-Channel SCC Plus Deeper FIFOs:
  - 4 Bytes on Transmitters
  - 8 Bytes on Receivers
  - DPLL Counter Per Channel
  - Software Compatible to SCC

### Package

- 40-Pin DIP
- 44-Pin CERDIP
- 44-Pin PLCC
- 40-Pin DIP
- 44-Pin PLCC
- 44-Pin QFP (Z85233 Only)
- 68-Pin PLCC
- 68-Pin PLCC
- 100-Pin VQFP

### Support Products

- Z80186002ZCO - Evaluation Board
- Z85230002ZCO - Evaluation Board
- Z80180002ZCO - Evaluation Board
- Z80380002ZCO - Evaluation Board
- Z85230002ZCO - Evaluation Board
- ZEPMD00002 - EPM™ Manual

### Superintegration™ Products Guide

- **85C30 SCC**
- **53C80 SCSI**

**SCSI Serial Communication and Small Computer Interface**

**Integrated Serial Communication Controller**

**Two Independent Full-Duplex Channels**

**Four DMA Controllers**

**Bus Interface Unit**

**Direct SCSI Bus Interface**

**Supports SCSI ANSI-X3.131-1986 Standard**
<table>
<thead>
<tr>
<th><strong>Part Number</strong></th>
<th>Z80181</th>
<th>Z80182/Z8L182</th>
<th>Z16C30</th>
<th>Z16C32</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Smart Access Controller</td>
<td>Zilog Intelligent Peripheral (ZIP™) Z8L182 = Low-Voltage Version</td>
<td>Universal Serial Controller (USC™)</td>
<td>Integrated Universal Serial Controller</td>
</tr>
<tr>
<td><strong>Process/Speed</strong></td>
<td>CMOS: 10, 12 MHz</td>
<td>Z80182 = CMOS: 16, 18, 33 MHz ZBL182 = CMOS: 20 MHz</td>
<td>CMOS: 10 MHz CPU Bus 10 Mb/s</td>
<td>CMOS: 20 MHz DMA Clock 20 Mb/s</td>
</tr>
<tr>
<td><strong>Features</strong></td>
<td>Complete Z180™ plus SCC/2CTC 16 I/O Lines Emulation Mode</td>
<td>Complete Static Version of Z180™ plus ESCC (2 Channels of Z85230 with 32-Bit CRC not available for 16 MHz) 16550 MIMIC 24 Parallel I/O Emulation Mode 3.3V and 5V Version</td>
<td>Two Dual-Channel 32-Byte Receive and Transmit FIFOs 16-Bit Bus B/W: 18.2 Mb/s Two BRGs Per Channel Flexible B/16-Bit Bus Interface 12 Serial Protocols Eight Data Encoding Bits</td>
<td>Single-Channel (Half of USC) plus two DMA Controllers Array Chained and Linked-List Modes with Ring Buffer Support</td>
</tr>
<tr>
<td><strong>Package</strong></td>
<td>100-Pin QFP</td>
<td>100-Pin QFP 100-Pin QFP</td>
<td>68-Pin PLCC</td>
<td>68-Pin PLCC</td>
</tr>
</tbody>
</table>

**Diagram:**

- CTC
- SCC/2 (85C30/2)
- 16 I/O
- Z180
- 24 I/O
- 85230 ESCC (2 CH)
- 16550 MIMIC
- S180
- USC
- USC/2
- DMA
- DMA
<table>
<thead>
<tr>
<th>PART NUMBER</th>
<th>DESCRIPTION</th>
<th>PROCESS/SPEED</th>
<th>FEATURES</th>
<th>PACKAGE</th>
<th>SUPPORT PRODUCTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z86C91/Z8691</td>
<td>ROMLess Z8*</td>
<td>Z86C91 = CMOS: 16 MHz Z8691 = NMOS: 12 MHz</td>
<td><strong>Features</strong></td>
<td>40-Pin DIP 44-Pin PLCC 44-Pin QFP</td>
<td>Z0860000ZCO - Evaluation Board Z86C0000ZUSP064 - Signum Emulator Z86C1200ZPD - Signum Emulator Pod</td>
</tr>
<tr>
<td>Z86E21/Z86C21</td>
<td>Z86E21 = 8K OTP Z86C21 = 8K ROM</td>
<td>CMOS: 12, 16 MHz</td>
<td><strong>Features</strong></td>
<td>40-Pin DIP 44-Pin PLCC 44-Pin QFP</td>
<td>Z0860000ZCO - Evaluation Board Z86C0000ZUSP064 - Signum Emulator Z86C1200ZPD - Signum Emulator Pod</td>
</tr>
<tr>
<td>Z89C00</td>
<td>16-Bit Digital Signal Processor</td>
<td>CMOS: 10, 15 MHz</td>
<td><strong>Features</strong></td>
<td>68-Pin PLCC</td>
<td>Z89C00ZEM - Emulator</td>
</tr>
<tr>
<td>Z86C93</td>
<td>ROMLess Enhanced Z8* Multi/Div</td>
<td>CMOS: 20, 25, 33 MHz</td>
<td><strong>Features</strong></td>
<td>40-Pin DIP 44-Pin PLCC 44-Pin QFP</td>
<td>Z0860000ZCO - Evaluation Board Z86C0000ZUSP064 - Signum Emulator Z86C3300ZPD - Signum Emulator Pod Z86C3301ZPD - Signum Emulator Pod</td>
</tr>
</tbody>
</table>

**Features**:
- Full-Duplex UART
- Two Standby Modes (STOP and HALT)
- 2x8 Bit
- Counter/Timer
- 256 Byte RAM
- Full-Duplex UART
- Two Standby Modes (STOP and HALT)
- Two Counter/Timers
- ROM Protect Option
- RAM Protect Option
- Low-EMI Option
- 16-Bit Multiply/Accumulate
- 75 ns
- Two Data RAMs (256 Words Each)
- 4K Word ROM
- 64Kx16 Ext. ROM
- 16-Bit I/O Port
- 74 Instructions
- Most Single Cycle
- Two Conditional Branch Inputs, Two User Outputs
- Library of Macros
- Zero Overhead Pointers
- 16x16 Multiply 17 Clocks
- 32x16 Divide 20 Clocks
- Full-Duplex UART
- Two Standby Modes (STOP and HALT)
- Three 16-Bit Counter/Timers
<table>
<thead>
<tr>
<th>BLOCK DIAGRAM</th>
<th>MULTI/DIV UART</th>
<th>88-BIT R-S ECC</th>
<th>SRAM/DRAM CTRL</th>
<th>AT/IDE HOST INTERFACE</th>
<th>MULTI/DIV UART</th>
<th>CPU</th>
<th>OSC</th>
<th>464 RAM CLOCK</th>
<th>Search</th>
<th>Merge</th>
<th>P2</th>
<th>P3</th>
<th>A15-A0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>DSP</td>
<td>DAC</td>
<td>PWM</td>
<td>MCU INTERFACE</td>
<td>CPU</td>
<td>DSP</td>
<td>DAC</td>
<td>SPI</td>
<td>MCU</td>
<td>SPI</td>
<td>P2</td>
<td>P3</td>
<td>A15-A0</td>
</tr>
<tr>
<td>P2</td>
<td>P3</td>
<td>A15-0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PART NUMBER</th>
<th>Z86C95</th>
<th>Z86018</th>
<th>Z86193</th>
<th>Z86295</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCRIPTION</td>
<td>ROMLess Enhanced Z8® with DSP</td>
<td>Zilog Datapath Controller</td>
<td>ROMLess Enhanced Z8® Multiply/Divide</td>
<td>ROMLess Enhanced Z8® DSP Servo Timer</td>
</tr>
<tr>
<td>PROCESS/SPEED</td>
<td>CMOS: 24, 33 MHz</td>
<td>CMOS: 40 MHz</td>
<td>CMOS: 40 MHz</td>
<td>CMOS: 40 MHz</td>
</tr>
<tr>
<td>FEATURES</td>
<td>Eight Channel</td>
<td>Full-Track Read</td>
<td>16x16 Multiply 17 Clocks</td>
<td>Eight Channel</td>
</tr>
<tr>
<td></td>
<td>8-Bit ADC</td>
<td>Automatic Data Transfer (Point &amp; Go®)</td>
<td>32x16 Divide 38 Clocks</td>
<td>8-Bit ADC</td>
</tr>
<tr>
<td></td>
<td>8-Bit DAC</td>
<td>88-Bit Reed Solomon ECC &quot;On The Fly&quot;</td>
<td>Full-Duplex UART</td>
<td>8-Bit DAC</td>
</tr>
<tr>
<td></td>
<td>16-Bit Multiply/Divide</td>
<td>Full AT/IDE Bus Interface</td>
<td>Two Standby Modes (STOP &amp; HALT)</td>
<td>Serial Peripheral Interface (SPI)</td>
</tr>
<tr>
<td></td>
<td>Full-Duplex UART</td>
<td>64 Kbytes SRAM Buffer</td>
<td>Three 16-Bit Counter/Timers</td>
<td>Pulse Width Modulator (PWM)</td>
</tr>
<tr>
<td></td>
<td>Serial Peripheral Interface (SPI)</td>
<td>1 Mbytes DRAM Buffer</td>
<td>SEARCH Machine</td>
<td>Three 16-Bit Counter/Timer</td>
</tr>
<tr>
<td></td>
<td>Three Standby Modes (STOP/HALT/PAUSE)</td>
<td>Split Data Field Support</td>
<td>MERGE Machine</td>
<td>Full-Duplex UART</td>
</tr>
<tr>
<td></td>
<td>Pulse Width Modulator (PWM)</td>
<td>Joint Test Action Group (JTAG)</td>
<td>Bus Request Mode</td>
<td>16-Bit DSP</td>
</tr>
<tr>
<td></td>
<td>3x16-Bit Timer</td>
<td>Boundary Scan Option</td>
<td>Evaluation Mode</td>
<td>Programmable Servo Timer</td>
</tr>
<tr>
<td></td>
<td>16-Bit DSP Slave Processor</td>
<td>8 Kbytes Buffer RAM Reserved for MCU</td>
<td>Z8® - DSP Mail Box</td>
<td>16-Bit DSP</td>
</tr>
<tr>
<td></td>
<td>83 ns Multiply/Accumulate</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PACKAGE</th>
<th>80-Pin QFP</th>
<th>100-Pin VQFP</th>
<th>64-Pin VQFP</th>
<th>100-Pin VQFP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>84-Pin PLCC</td>
<td>100-Pin VQFP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SUPPORT PRODUCTS</th>
<th>Z86C9500ZCO - Evaluation Board</th>
<th>Z86C9900ZCO - Evaluation Board</th>
<th>Z8619200ZME - Emulator</th>
<th>Z862IA01ZCO - Evaluation Board</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Z86C9502ZUSP064 - Signum Emulator</td>
<td>Z86C9501ZUSP064 - Signum Emulator</td>
<td>Z86C9500ZPD - Signum Emulator POD</td>
<td>Z86C9501ZPD - Signum Emulator POD</td>
</tr>
<tr>
<td></td>
<td>Z86C9500ZPD - Signum Emulator POD</td>
<td>Z86C9500ZPD - Signum Emulator POD</td>
<td>Z86C9500ZPD - Signum Emulator POD</td>
<td>Z86C9500ZPD - Signum Emulator POD</td>
</tr>
<tr>
<td></td>
<td>Z86C9500ZCO - Evaluation Board</td>
<td>Z86C9500ZCO - Evaluation Board</td>
<td>Z86C9500ZCO - Evaluation Board</td>
<td>Z86C9500ZCO - Evaluation Board</td>
</tr>
</tbody>
</table>
## Bus Interface

<table>
<thead>
<tr>
<th><strong>Block Diagram</strong></th>
<th><strong>Superintegration™ Products Guide</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Block Diagram" /></td>
<td><img src="image" alt="Superintegration™ Products Guide" /></td>
</tr>
</tbody>
</table>

### Part Number

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Z86016</th>
<th>Z86017</th>
<th>Z86M17</th>
<th>Z86020</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>8-Bit PCMCIA Interface Adaptor</td>
<td>PCMCIA Interface Adaptor</td>
<td>PCMCIA Interface Adaptor</td>
<td>PCI/Multifunction Bridge</td>
</tr>
<tr>
<td><strong>Process/Speed</strong></td>
<td>CMOS: 20 MHz</td>
<td>CMOS: 20 MHz</td>
<td>CMOS: 20 MHz</td>
<td>CMOS: 33 MHz</td>
</tr>
<tr>
<td><strong>Features</strong></td>
<td><img src="image" alt="Features" /></td>
<td><img src="image" alt="Features" /></td>
<td><img src="image" alt="Features" /></td>
<td><img src="image" alt="Features" /></td>
</tr>
<tr>
<td><strong>Package</strong></td>
<td>48-Pin VQFP</td>
<td>100-Pin VQFP</td>
<td>100-Pin VQFP</td>
<td>160-Pin QFP</td>
</tr>
<tr>
<td><strong>Support Products</strong></td>
<td>Z86016002ZCO - Evaluation Board (Available Q494)</td>
<td>Z8601700ZCO - Evaluation Board</td>
<td>Z8601700ZCO - Evaluation Board</td>
<td>Available Q494</td>
</tr>
</tbody>
</table>
ZILOG DOMESTIC SALES OFFICES
AND TECHNICAL CENTERS

CALIFORNIA
Agoura .......................................................... 818-707-2160
Campbell ........................................................ 408-370-8120
Irvine ............................................................ 714-453-9701
San Diego ...................................................... 619-658-0391

COLORADO
Boulder .......................................................... 303-494-2905

FLORIDA
Clearwater ...................................................... 813-725-8400

GEORGIA
Duluth ........................................................... 404-931-4022

ILLINOIS
Schaumburg .................................................... 708-517-8080

MINNESOTA
Minneapolis ..................................................... 612-944-0737

NEW HAMPSHIRE
Nashua ........................................................... 603-888-8590

OHIO
Independence ................................................ 216-447-1480

OREGON
Portland .......................................................... 503-274-6250

PENNSYLVANIA
Horsham .......................................................... 215-764-0805

TEXAS
Austin ............................................................. 512-343-8976
Dallas .............................................................. 214-987-9987

INTERNATIONAL SALES OFFICES

CANADA
Toronto ............................................................ 905-850-2377

CHINA
Shenzhen ...................................................... 86-755-2236089
Shanghai ........................................................ 86-21-4370050, x5204
86-21-4331020

GERMANY
Munich ........................................................... 49-89-67-2045
Sömmerda ..................................................... 49-3634-23906

JAPAN
Tokyo ............................................................ 81-3-3587-0528

HONG KONG
Kowloon .......................................................... 852-7238979

KOREA
Seoul ............................................................. 82-2-577-3272

SINGAPORE
Singapore ........................................................ 65-2357155

TAIWAN
Taipei ............................................................ 886-2-741-3125

UNITED KINGDOM
Maidenhead .................................................... 44-628-392-00

© 1994 by Zilog, Inc. All rights reserved. No part of this document may be copied or reproduced in any form or by any means without the prior written consent of Zilog, Inc. The information in this document is subject to change without notice. Devices sold by Zilog, Inc. are covered by warranty and patent indemnification provisions appearing in Zilog, Inc. Terms and Conditions of Sale only. Zilog, Inc. makes no warranty, express, statutory, implied or by description, regarding the information set forth herein or regarding the freedom of the described devices from intellectual property infringement. Zilog, Inc. makes no warranty of merchantability or fitness for any purpose. Zilog, Inc. shall not be responsible for any errors that may appear in this document. Zilog, Inc. makes no commitment to update or keep current the information contained in this document.

Zilog’s products are not authorized for use as critical components in life support devices or systems unless a specific written agreement pertaining to such intended use is executed between the customer and Zilog prior to use. Life support devices or systems are those which are intended for surgical implantation into the body, or which sustains life whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.

Zilog, Inc. 210 East Hacienda Ave.
Campbell, CA 95008-6600
Telephone (408) 370-8000
Telex 910-338-7621
FAX 408 370-6056
U.S., CANADIAN & PUERTO RICAN REPRESENTATIVES

ARIZONA
Scottsdale
Thom Luke Sales, Inc. ............................................ (602) 451-5400

CALIFORNIA
Santa Clara
Phase II Technical Sales .................................... (408) 980-0414
Irvine
Infinity Sales .................................................. (714) 833-0300

COLORADO
Englewood
Thorson Rocky Mountain .................................. (303) 773-6300

CONNECTICUT
Wallingford
Advanced Technical Sales .................................. (508) 664-0888

FLORIDA
Altamonte Springs
Semtronic Associates, Inc. .............................. (407) 831-8233
Clearwater
Semtronic Associates, Inc. .............................. (813) 461-4675
Fort Lauderdale
Semtronic Associates, Inc. .............................. (305) 731-2484

GEORGIA
Norcross
BITS .......................................................... (404) 564-5599

ILLINOIS
Hoffman Estates
Victory Sales, Inc. ............................................ (708) 490-0300

IOWA
Cedar Rapids
Advanced Technical Sales .............................. (319) 393-8280

KANSAS
Olathe
Advanced Technical Sales .............................. (913) 782-8702

MARYLAND
Pasadena
Electronic Engineering & Sales ........................ (410) 255-9686

Massachusetts
North Reading
Advanced Technical Sales .............................. (508) 664-0888

MICHIGAN
Novi
Rathsburg Associates, Inc. .............................. (810) 615-4000

MINNESOTA
Minneapolis
Professional Sales for Industry ....................... (612) 944-8545

MISSOURI
Bridgeton
Advanced Technical Sales .............................. (314) 291-5003

NORTH CAROLINA
Huntsville
BITS .......................................................... (205) 881-2900
Raleigh
BITS .......................................................... (919) 676-1880

NEW JERSEY
Cherry Hill
Tritek .......................................................... (609) 667-0200

NEW MEXICO
Albuquerque
Quatra & Associates ........................................ (505) 296-6781

NEW YORK
Fairport
L-Mar Associates, Inc. ..................................... (716) 425-9100

OHIO
Centerville
Q-Mark, Inc. .................................................. (513) 438-1129
Independence
Rathsburg Associates, Inc. .............................. (216) 447-8825

OKLAHOMA
Tulsa
Nova Marketing, Inc. ....................................... (918) 660-5105
SALES REPRESENTATIVES AND DISTRIBUTORS

U.S., CANADIAN & PUERTO RICAN REPRESENTATIVES

OREGON
Portland
Phase II Technical Sales ..................................... (503) 643-6455

TEXAS
Austin
Nova Marketing, Inc. ......................................... (512) 343-2321
Dallas
Nova Marketing, Inc. ......................................... (214) 265-4630
Houston
Nova Marketing, Inc. ......................................... (713) 240-6082

UTAH
Salt Lake City
Thorson Rocky Mountain ...................................... (801) 942-1683

WASHINGTON
Kirkland
Phase II Technical Sales ...................................... (206) 823-3874

WISCONSIN
Brookfield
Victory Sales, Inc. ........................................... (414) 789-5770

CANADA
British Columbia
J-Squared Technologies, Inc. ............................... (604) 473-4666
Ontario
J-Squared Technologies, Inc. ............................... (905) 672-2030
Ottawa
J-Squared Technologies, Inc. ............................... (613) 592-9540
Quebec
J-Squared Technologies, Inc. ............................... (514) 694-8330

PUERTO RICO
San Juan
Semtronic Associates, Inc. ................................. (809) 766-0700
# Sales Representatives and Distributors

## U.S. and Canadian Distributors

### Nationwide
- **Newark Electronics** ........................................................................... 1-800-367-3573
- **Zeus Electronics** ................................................................................. 1-800-524-4735

### Alabama
- **Hamilton Hallmark Electronics**
  - Inside Alabama .......................................................................................... 800-572-7236
  - Outside Alabama ....................................................................................... 800-633-2918
- **Huntsville**
  - Arrow Electronics ....................................................................................... (205) 837-6955
  - Hamilton Hallmark Electronics ......................................................... (205) 837-8700

### Arizona
- **Hamilton Hallmark Electronics**
  - Inside Arizona .......................................................................................... 800-352-8489
  - Outside Arizona ......................................................................................... 800-528-8471
- **Phoenix**
  - Hamilton Hallmark Electronics ............................................................... (602) 437-1200
- **Tempe**
  - Anthem Electronics .................................................................................... (602) 966-6600
  - Arrow Electronics ....................................................................................... (602) 431-0030

### California
- **Calabasas**
  - Arrow Electronics ....................................................................................... (818) 880-9686
- **Chatsworth**
  - Anthem Electronics .................................................................................... (818) 700-1000
- **Costa Mesa**
  - Hamilton Hallmark Electronics ............................................................... (714) 641-4100
- **Fremont**
  - Arrow Electronics ....................................................................................... (510) 490-4977
- **Irvine**
  - Anthem Electronics ..................................................................................... (714) 768-4444
  - Arrow Electronics ....................................................................................... (714) 587-0404
  - Zeus Electronics ......................................................................................... (714) 581-4622
- **Rocklin**
  - Anthem Electronics .................................................................................... (916) 624-9744
  - Hamilton Hallmark Electronics ............................................................... (916) 624-9781
- **San Diego**
  - Anthem Electronics .................................................................................... (619) 453-9005
  - Arrow Electronics ....................................................................................... (619) 565-4800
  - Hamilton Hallmark Electronics ............................................................... (619) 571-7540
- **San Jose**
  - Anthem Electronics .................................................................................... (408) 453-1200
  - Arrow Electronics ....................................................................................... (408) 441-9700
  - Hamilton Hallmark Electronics ............................................................... (408) 435-3500
  - Zeus Electronics ......................................................................................... (408) 629-4789
- **Woodland Hills**
  - Hamilton Hallmark Electronics ............................................................... (818) 594-0404

### Colorado
- **Colorado Springs**
  - Hamilton Hallmark Electronics ............................................................... (719) 637-0055
- **Englewood**
  - Anthem Electronics .................................................................................... (303) 790-4500
  - Arrow Electronics ....................................................................................... (303) 799-0258
  - Hamilton Hallmark Electronics ............................................................... (303) 790-1662

### Connecticut
- **Cheshire**
  - Hamilton Hallmark Electronics ............................................................... (203) 271-2844
- **Wallingford**
  - Arrow Electronics ....................................................................................... (203) 265-7741
- **Waterbury**
  - Anthem Electronics .................................................................................... (203) 596-3200

### Florida
- **Deerfield Beach**
  - Arrow Electronics ....................................................................................... (305) 429-8200
- **Lake Mary**
  - Arrow Electronics ....................................................................................... (407) 333-9300
  - Zeus Electronics ......................................................................................... (407) 333-3055
- **Largo**
  - Hamilton Hallmark Electronics ............................................................... (813) 541-7440
  - (800) 282-9350
- **Fort Lauderdale**
  - Hamilton Hallmark Electronics ............................................................... (305) 484-5482
- **Winter Park**
  - Hamilton Hallmark Electronics ............................................................... (407) 657-3300

### Georgia
- **Duluth**
  - Arrow Electronics ....................................................................................... (404) 497-1300
  - Hamilton Hallmark Electronics ............................................................... (404) 623-5475
  - (404) 623-4400

### Illinois
- **Bensenville**
  - Hamilton Hallmark Electronics ............................................................... (708) 860-7780
- **Itasca**
  - Arrow Electronics ....................................................................................... (708) 250-0500
  - Zeus Electronics ......................................................................................... (708) 595-9730
- **Schaumburg**
  - Anthem Electronics .................................................................................... (708) 884-0200
# Sales Representatives and Distributors

## U.S. and Canadian Distributors

### Indiana

**Indianapolis**
- Arrow Electronics: (317) 299-2071
- Hamilton Hallmark Electronics: (317) 872-8875, (800) 829-0146

### Iowa

**Cedar Rapids**
- Arrow Electronics: (319) 395-7230

### Kansas

**Lenexa**
- Arrow Electronics: (913) 541-9542
- Hamilton Hallmark Electronics: (913) 888-4747, (800) 332-4375

### Kentucky

**Lexington**
- Hamilton Hallmark Electronics: (800) 235-6039, (800) 525-0069

### Maryland

**Columbia**
- Anthem Electronics: (410) 995-6640
- Arrow Electronics: (410) 596-7000
- Hamilton Hallmark Electronics: (410) 988-9800

### Massachusetts

**Peabody**
- Hamilton Hallmark Electronics: (508) 532-9808

**Wilmington**
- Anthem Electronics: (508) 657-5170
- Arrow Electronics: (508) 658-0900
- Zeus Electronics: (508) 658-4776

### Michigan

**Livonia**
- Arrow Electronics: (313) 462-2290
- Norl
  - Hamilton Hallmark Electronics: (313) 347-4271

**Plymouth**
- Hamilton Hallmark Electronics: (313) 416-5800, (800) 767-9654

### Minnesota

**Bloomington**
- Hamilton Hallmark Electronics: (612) 881-2600

**Eden Prairie**
- Anthem Electronics: (612) 944-5454
- Arrow Electronics: (612) 941-5280

### Missouri

**Earth City**
- Hamilton Hallmark Electronics: (314) 291-5350

**St. Louis**
- Arrow Electronics: (314) 567-6888

### Nevada

**Sparks**
- Arrow Electronics: (702) 331-5000

### New Jersey

**Cherry Hill**
- Hamilton Hallmark Electronics: (609) 235-1900

**Marlton**
- Arrow Electronics: (609) 596-8000

**Pinebrook**
- Anthem Electronics: (201) 227-7960
- Arrow Electronics: (201) 227-7880

**Parsippany**
- Hamilton Hallmark Electronics: (201) 575-4415

### New York

**Commack**
- Anthem Electronics: (516) 864-6600

**Hauppauge**
- Arrow Electronics: (516) 231-2500
- Hamilton Hallmark Electronics: (516) 737-0600

**Melville**
- Arrow Electronics: (516) 391-1300

**Rochester**
- Arrow Electronics: (716) 427-0300
- Hamilton Hallmark Electronics: (716) 475-9130

**Ronkonkoma**
- Hamilton Hallmark Electronics: (516) 737-0600

**Port Chester**
- Zeus Electronics: (914) 937-7400

### North Carolina

**Raleigh**
- Arrow Electronics: (919) 876-3132
- Hamilton Hallmark Electronics: (919) 872-0712

---

Z-5
## SALES REPRESENTATIVES AND DISTRIBUTORS

### U.S. AND CANADIAN DISTRIBUTORS

#### OHIO

<table>
<thead>
<tr>
<th>Location</th>
<th>Company/Contact</th>
<th>Phone(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centerville</td>
<td>Arrow Electronics</td>
<td>(513) 435-5563</td>
</tr>
<tr>
<td>Dayton</td>
<td>Hamilton Hallmark Electronics</td>
<td>(513) 439-6735</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(800) 423-4688</td>
</tr>
<tr>
<td>Solon</td>
<td>Arrow Electronics</td>
<td>(216) 248-3990</td>
</tr>
<tr>
<td></td>
<td>Hamilton Hallmark Electronics</td>
<td>(216) 498-1100</td>
</tr>
<tr>
<td>Worthington</td>
<td>Hamilton Hallmark Electronics</td>
<td>(614) 888-3313</td>
</tr>
</tbody>
</table>

#### OKLAHOMA

<table>
<thead>
<tr>
<th>Location</th>
<th>Company/Contact</th>
<th>Phone(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tulsa</td>
<td>Arrow Electronics</td>
<td>(918) 252-7537</td>
</tr>
<tr>
<td></td>
<td>Hamilton Hallmark Electronics</td>
<td>(918) 459-6000</td>
</tr>
</tbody>
</table>

#### OREGON

<table>
<thead>
<tr>
<th>Location</th>
<th>Company/Contact</th>
<th>Phone(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beaverton</td>
<td>ALMAC/Arrow Electronics</td>
<td>(503) 629-8090</td>
</tr>
<tr>
<td></td>
<td>Anthem Electronics</td>
<td>(503) 643-1114</td>
</tr>
<tr>
<td></td>
<td>Hamilton Hallmark Electronics</td>
<td>(503) 526-6200</td>
</tr>
</tbody>
</table>

#### PENNSYLVANIA

<table>
<thead>
<tr>
<th>Location</th>
<th>Company/Contact</th>
<th>Phone(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horsham</td>
<td>Anthem Electronics</td>
<td>(215) 443-5150</td>
</tr>
<tr>
<td>Pittsburgh</td>
<td>Arrow Electronics</td>
<td>(412) 963-6807</td>
</tr>
</tbody>
</table>

#### TEXAS

<table>
<thead>
<tr>
<th>Location</th>
<th>Company/Contact</th>
<th>Phone(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austin</td>
<td>Arrow Electronics</td>
<td>(512) 835-4180</td>
</tr>
<tr>
<td></td>
<td>Hamilton Hallmark Electronics</td>
<td>(512) 259-8848</td>
</tr>
<tr>
<td>Carrollton</td>
<td>Arrow Electronics</td>
<td>(214) 380-6464</td>
</tr>
<tr>
<td></td>
<td>Zeus Electronics</td>
<td>(214) 380-4330</td>
</tr>
<tr>
<td>Dallas</td>
<td>Hamilton Hallmark Electronics</td>
<td>(214) 553-4300</td>
</tr>
<tr>
<td>Houston</td>
<td>Arrow Electronics</td>
<td>(713) 530-4700</td>
</tr>
<tr>
<td></td>
<td>Hamilton Hallmark Electronics</td>
<td>(713) 781-6100</td>
</tr>
<tr>
<td>Richardson</td>
<td>Anthem Electronics</td>
<td>(214) 238-7100</td>
</tr>
<tr>
<td>San Antonio</td>
<td>Hamilton Hallmark Electronics</td>
<td>(210) 828-2246</td>
</tr>
</tbody>
</table>

#### UTAH

<table>
<thead>
<tr>
<th>Location</th>
<th>Company/Contact</th>
<th>Phone(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salt Lake City</td>
<td>Anthem Electronics</td>
<td>(801) 973-8555</td>
</tr>
<tr>
<td></td>
<td>Arrow Electronics</td>
<td>(801) 973-6913</td>
</tr>
<tr>
<td></td>
<td>Hamilton Hallmark Electronics</td>
<td>(801) 266-2022</td>
</tr>
</tbody>
</table>

#### WASHINGTON

<table>
<thead>
<tr>
<th>Location</th>
<th>Company/Contact</th>
<th>Phone(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bellevue</td>
<td>ALMAC/Arrow Electronics</td>
<td>(206) 643-9992</td>
</tr>
<tr>
<td>Bothell</td>
<td>Anthem Electronics</td>
<td>(206) 483-1700</td>
</tr>
<tr>
<td>Redmond</td>
<td>Hamilton Hallmark Electronics</td>
<td>(206) 881-6697</td>
</tr>
<tr>
<td>Spokane</td>
<td>ALMAC/Arrow Electronics</td>
<td>(509) 924-9500</td>
</tr>
</tbody>
</table>

#### PENNSYLVANIA

<table>
<thead>
<tr>
<th>Location</th>
<th>Company/Contact</th>
<th>Phone(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brookfield</td>
<td>Arrow Electronics</td>
<td>(414) 792-0150</td>
</tr>
<tr>
<td>New Berlin</td>
<td>Hamilton Hallmark Electronics</td>
<td>(414) 797-7844</td>
</tr>
</tbody>
</table>

#### CANADA

<table>
<thead>
<tr>
<th>Location</th>
<th>Company/Contact</th>
<th>Phone(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alberta</td>
<td>Future Electronics</td>
<td>(403) 250-5550</td>
</tr>
<tr>
<td></td>
<td>Future Electronics</td>
<td>(403) 438-2858</td>
</tr>
<tr>
<td>British Columbia</td>
<td>Arrow Electronics</td>
<td>(604) 421-2333</td>
</tr>
<tr>
<td></td>
<td>Future Electronics</td>
<td>(604) 294-1166</td>
</tr>
<tr>
<td></td>
<td>Hamilton Hallmark Electronics</td>
<td>(604) 420-4101</td>
</tr>
<tr>
<td>Manitoba</td>
<td>Future Electronics</td>
<td>(204) 786-7711</td>
</tr>
<tr>
<td>Ontario</td>
<td>Arrow Electronics</td>
<td>(613) 226-6903</td>
</tr>
<tr>
<td></td>
<td>Future Electronics</td>
<td>(905) 670-7769</td>
</tr>
<tr>
<td></td>
<td>Future Electronics</td>
<td>(905) 612-9200</td>
</tr>
<tr>
<td></td>
<td>Hamilton Hallmark Electronics</td>
<td>(613) 820-8313</td>
</tr>
<tr>
<td></td>
<td>Hamilton Hallmark Electronics</td>
<td>(416) 564-6060</td>
</tr>
<tr>
<td></td>
<td>Hamilton Hallmark Electronics</td>
<td>(613) 226-1700</td>
</tr>
<tr>
<td>Quebec</td>
<td>Arrow Electronics</td>
<td>(514) 421-7411</td>
</tr>
<tr>
<td></td>
<td>Future Electronics</td>
<td>(514) 694-7710</td>
</tr>
<tr>
<td></td>
<td>Hamilton Hallmark Electronics</td>
<td>(514) 335-1000</td>
</tr>
</tbody>
</table>
SALES REPRESENTATIVES AND DISTRIBUTORS

CENTRAL AND SOUTH AMERICA

MEXICO
Semiconductores ............................................. 525-524-6123
Profesionales ............................................. 525-524-6123
Proyeccion Electronica ..................................... 525-264-7482

ARGENTINA
Buenos Aires
YEL SRL .................................................. 011-541-440-1532

BRAZIL
Sao Paulo
Nishicom .................................................. 011-55-11-535-1755

ASIA-PACIFIC

AUSTRALIA
R&D Electronics ............................................. 61-3-558-0444
GEC Electronics Division ............................. 61-2-638-1888

CHINA
Beijing
Lestina International Ltd. ............. 86-1-849-8888
China Electronics Appliance Corp. ....... 86-755-335-4214
TLG Electronics, Ltd. ......................... 85-2-388-7613

Guang Zhou
Lestina International Ltd. ............. 86-20-885-0613
.......................................................... 86-20-886-1615

HONG KONG
Lestina International Ltd. ............. 852-735-1736
Electrocon Products Ltd. ............... 852-481-6022
Components Agent, Ltd. ............... 852-487-8826

INDIA
Bangalore
Maxvale (S) Pte. Ltd. ......................... 91-80-568369
Zenith Technologies Pvt. Ltd. .......... 91-812-586782

Bombay
Zenith Technologies Pvt. Ltd. .......... 91-22-4947457

New Delhi
Maxvale (S) Pte. Ltd. ......................... 91-11-685-3180

INDONESIA
Jakarta
Cinergi Asiamaju ...................................... 62-21-7982762

JAPAN
Tokyo
Teksel Co., Ltd. ...................................... 81-3-5467-9000
Internix Incorporated ......................... 81-3-3369-1101
Kanematsu Elec. Components Corp. ...... 81-3-3779-7611

Osaka
Teksel Co., Ltd. ...................................... 81-6-6368-9000

KOREA
ENC-Korea ................................................. 822-523-2220

MALAYSIA
Eltee Electronics Ltd. ......................... 60-3-7038498

NEW ZEALAND
GEC Electronics Division ..................... 64-25-971057

PHILIPPINES
Alexan Commercial .................................. 63-2-402223

SINGAPORE
Eltee Electronics Ltd. ......................... 65-2830888

TAIWAN (ROC)
Acer Sertek, Inc. ................................. 886-2-501-0055
Promate Electronics Co. Ltd. .......... 886-2-659-0303

THAILAND
Eltee Electronics Ltd. ......................... 66-2-538-4600
<table>
<thead>
<tr>
<th>Sales Representatives and Distributors</th>
</tr>
</thead>
</table>

**EUROPE**

**AUSTRIA**
- **Vienna**
  - EBV Elektronik GMBH ........................... 0043-1-8941774
  - Avnet/Electronic 2000 ....................... 0043-1-9112847

**BELGIUM**
- **Antwerp**
  - D & D Electronics PVBA ........................ 32-3-8277934
  - EBV Elektronik ............................................. 322-7160010

**DENMARK**
- **Brondby**
  - Ditz Schweitzer AS .............................. 4542-453044
  - Lynge
  - Rep Delco .................................................. 45-35-821200

**ENGLAND**
- **Berkshire**
  - Future Electronics .............................. 44-753-687000
  - Gothic Crellon ................................ 44-734-787848
  - Macro Marketing ..................................... 44-628-604383
  - **Lancashire**
  - Complementary Technologies Ltd. / ........ 44-942-274731

**FINLAND**
- **Espoo**
  - OY SW Instruments AB ........................... 358-0-522-122

**FRANCE**
- **Antony**
  - CCI Electronique .................................... 331-46744700
  - Cedex
  - A2M ......................................................... 331-46232425
  - **Champs sur Marne**
  - EBV Elektronik .............................................. 331-64688600
  - Massy
  - Reptronic SA ........................................... 331-60139300

**GERMANY**
- **Berlin**
  - EBV Elektronik GMBH ................................. 030-3421041
  - Avnet/Electronic 2000 ................................. 030-2110761

- **Burgwedel**
  - EBV Elektronik GMBH ................................... 05139-80870

- **Dortmund**
  - Future GMBH ............................................. 02305-42051
  - **Duesseldorf**
  - Avnet/Electronic 2000 ............................... 0211-92003-0
  - Thesys/AE ..................................................... 0211-53602-0
  - Efurt
  - Thesys ..................................................... 0361-4278100

- **Frankfurt**
  - EBV Elektronik GMBH ................................... 069-785037
  - Avnet/Electronic 2000 ................................. 069-973840
  - Future GMBH .............................................. 06126-54020
  - Thesys/AE ..................................................... 06434-5041

- **Hamburg**
  - Avnet/Electronic 2000 ................................... 040-64557021

- **Leonberg**
  - EBV Elektronik GMBH ................................... 07152-30090

- **Muenchen**
  - EBV Elektronik GMBH ................................... 089-9571950
  - Future GMBH .............................................. 089-9571950

- **Nuernberg**
  - Avnet/Electronic 2000 ................................. 0911-9951610

- **Neuss**
  - EBV Elektronik GMBH ................................... 02131-96770

- **Stuttgart**
  - Avnet/Electronic 2000 ............................... 07156-356190
  - Future GMBH .............................................. 0711-830830
  - Thesys/AE ..................................................... 0711-9889100

- **Weissbach**
  - EBV Elektronik GMBH ................................... 036-426486
SALES REPRESENTATIVES AND DISTRIBUTORS

ISRAEL
RDT .................................................. 972-36450707

ITALY
Milano
Avmir De Mico ..................................... 0039-295-343600
EBV Elektronik ................................... 0039-2-66017111

Firenze
EBV Elektronik ...................................... 0039-55-350792

Roma
EBV Elektronik ...................................... 0039-6-2253367

Modena
EBV Elektronik ...................................... 0039-59-344752

Napoli
EBV Elektronik ...................................... 0039-81-2395540

Torino
EBV Elektronik ...................................... 0039-11-2161531

Vicenza
EBV Elektronik ...................................... 0039-444-572366

NETHERLANDS
EBV Elektronik ...................................... 313-46562353

NORWAY
Bexab Norge ......................................... 47-63833800

POLAND
Warsaw
Gamma Ltd. ........................................... 004822-330853

PORTUGAL
Amadora
Amitron-Arrow ....................................... 0035-1-4714806

RUSSIA
Woronosh
Thesys/Intertechna ................................ 0732593697

Vyborg
Gamma Ltd. ........................................... 081278-31509

St. Petersburg
Gamma Ltd. ........................................... 0812-5131402

SPAIN
Barcelona
Amitron-Arrow S.A. .................................. 0034-3-4907494

Madrid
Amitron-Arrow S.A. .................................. 0034-1-3043040

SWEDEN
Bexab Sweden AB .................................... 46-8-630-8800

SWITZERLAND
Dietikon
EBV Elektronik GMBH ................................ 0041-1-7401090

Lausanne
EBV Elektronik AG ................................... 0041-21-3112804

Regensdorf
Eurodis AG ............................................ 0041-1-8433111

UKRAINE
Kiev
Thesys/Mikropribor .................................. 04434-9533
Z86E30/E31 CMOS Z8® OTP CCP™
Consumer Controller Processor

Z86C40 CMOS Z8® 4K ROM CCP™
Consumer Controller Processor

Z86E40 CMOS Z8® 8-Bit OTP CCP™
Consumer Controller Processor

Z8® Microcontrollers
Application Notes

Z8® Support Products
and Third Party Vendors

Superintegration™
Products Guide

Zilog Sales Offices,
Representatives & Distributors

Literature Guide and
Ordering Information
LITERATURE GUIDE

Z8® MICROCONTROLLERS - CONSUMER FAMILY OF PRODUCTS

Databooks By Market Niche

<table>
<thead>
<tr>
<th>Product Specifications</th>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z8® Microcontrollers Databook</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Product Specifications</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86C07 CMOS Z8 8-Bit Microcontroller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86C08 CMOS Z8 8-Bit Microcontroller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86E08 CMOS Z8 8-Bit OTP Microcontroller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86C11 CMOS Z8 Microcontroller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86C12 CMOS Z8 In-Circuit Microcontroller Emulator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86C21 8K ROM Z8 CMOS Microcontroller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86E21 CMOS Z8 8K OTP Microcontroller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86C61/62/96 CMOS Z8 Microcontroller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86C63/64 32K ROM Z8 CMOS Microcontroller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86C91 CMOS Z8 ROMless Microcontroller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86C93 CMOS Z8 Multiply/Divide Microcontroller</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Support Product Specifications</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z08600000ZCO Development Kit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86C0800ZCO Applications Board</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86C0800ZDP Adaptor Board</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86E2100ZDF Adaptor Kit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86E2100ZDP Adaptor Kit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86E2100ZDV Adaptor Kit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86E2100ZDV Adaptor Kit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86C6100TSC Z86C61/63 MCU OTP Emulation Board</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86C6200ZEM In-Circuit Emulator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86C1200ZEM Z8® In-Circuit Emulator -C12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z8® S Series Emulators, Base Units and Pods</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Additional Information</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zilog's Superintegration™ Products Guide</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Literature Guide</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Third Party Support Vendors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zilog's Sales Offices, Representatives and Distributors</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Infrared Remote (IR) Controllers Databook**

<table>
<thead>
<tr>
<th>Product Specifications</th>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Specifications</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86L06 Low Voltage CMOS Consumer Controller Processor (Preliminary)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86L29 6K Infrared (IR) Remote (ZIRC™) Controller (Advance Information)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86L70/L71/L72/L75/L76 Zilog IR (ZIRC™) CCP™ Controller Family (Preliminary)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86E72/E73/E74 Zilog IR (ZIRC™) CCP™ Controller Family (Preliminary)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Application Note</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beyond the 3 Volt Limit</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Support Product Specifications</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86L7100ZDB Emulator Board</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z86L7100ZEM ICEBOX™ In-Circuit Emulator Board</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Additional Information</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zilog's Superintegration™ Products Guide</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Literature Ordering Guide</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zilog's Sales Offices, Representatives and Distributors</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**LITERATURE GUIDE**

**Z8® MICROCONTROLLERS - CONSUMER FAMILY OF PRODUCTS**

**Databooks By Market Niche**

**Discrete Z8® Microcontrollers**

**Product Specifications**
- Z86C03/C06 CMOS Z8® 8-Bit Consumer Controller Processors
- Z86E03/E06 CMOS Z8® 8-Bit OTP Consumer Controller Processors
- Z86C04/C08 CMOS Z8® 8-Bit Low Cost 1K/2K ROM Microcontrollers
- Z86E04/E08 CMOS Z8® 8-Bit OTP Microcontrollers
- Z86C07 CMOS Z8® 8-Bit Microcontroller
- Z86E07 CMOS Z8® 8-Bit OTP Microcontroller
- Z86C30/C31 CMOS Z8® 8-Bit Consumer Controller Processors
- Z86E30/E31 CMOS Z8® 8-Bit OTP Consumer Controller Processors
- Z86C40 CMOS Z8® 4K ROM Consumer Controller Processor
- Z86E40 CMOS Z8® 8-Bit OTP Consumer Controller Processor

**Z8® Microcontrollers Application Notes**
- Timekeeping with the Z8®
- Using The Zilog Z86C06 SPI Bus
- DTMF Tone Generation Using the Z8® CCP™
- Serial Communications Using the Z8® CCP™ Software UART
- The Versatile Z86C08: Three Key Features of this Z8® MCU
- The Z86C08 Controls a Scrolling LED Message Display
- Interfacing LCDs to the Z8® Microcontroller

**Support Product Specifications and Third Party Vendors**
- Z86C0800ZC0 Evaluation Board
- Z86C0800ZDP Adaptor Kit
- Z86C1200ZEM Emulator
- Z86E0600ZDP Adaptor Kit
- Z86E0700ZDP Adaptor Kit
- Z86E3000ZDP Adaptor Kit
- Z86E4000ZDF Adaptor Kit
- Z86E4000ZDP Adaptor Kit
- Z86E4000ZDV Adaptor Kit
- Z86E4001ZDF Adaptor Kit
- Z86E4001ZDV Adaptor Kit
- Z86CCP00ZEM Emulator
- Z86CCP00ZAC Emulator Kit
- Z8® S Series Emulators, Base Units and Pods

**Additional Information**
- Zilog’s Superintegration™ Products Guide
- Literature Guide and Ordering Information
- Zilog’s Sales Offices, Representatives and Distributors

**Part No** | **Unit Cost**
---|---
DC 8318-02 | $5.00
Z8® MICROCONTROLLERS - CONSUMER FAMILY OF PRODUCTS

Databooks By Market Niche

Digital Television Controllers

**Product Specifications**
- Z89300 Series Digital Television Controller
- Z86C27/97 CMOS Z8® Digital Signal Processor
- Z86C47/E47 CMOS Z8® Digital Signal Processor
- Z86127 Low Cost Digital Television Controller
- Z86128/228 Line 21 Closed-Caption Controller (L21C™)
- Z86227 40-Pin Low Cost (4LDTC™) Digital Television Controller

**Support Product Specifications**
- Z86C2700ZCO Application Kit
- Z86C2700ZDB Emulation Board
- Z86C2702ZEM In-Circuit Emulator

**Additional Information**
- Zilog's Superintegration™ Products Guide
- Literature Guide and Ordering Information
- Zilog's Sales Offices, Representatives and Distributors

### Telephone Answering Device Databook

**Product Specifications**
- Z89C65, Z89C66 (ROMless) Dual Processor T.A.M. Controller (Preliminary)
- Z89C67, Z89C68/C69 (ROMless) Dual Processor Tapeless T.A.M. Controller (Preliminary)

**Development Guides**
- Z89C65 Software Development Guide
- Z89C67/C69 Software Development Guide

**Technical Notes**
- Using Samsung KT8554 Codec on the ZTAD Development Board
- Z89C67/C69 Design Guidelines
- Z89C67/C69 ARAM Bit-Rate Measurements
- Z89C67 Codec Interfacing (Preliminary)
- Controlling the Out-5V and Codec Clock Signals for Low-Power Halt Mode

**Support Product Specifications**
- Z89C5900ZEM Emulation Module
- Z89C6500ZDB Emulation Board
- Z89C6501ZEM ICEBOX™ In-Circuit Emulator
- Z89C6700ZDB Emulator Board
- Z89C6700ZEM ICEBOX™ Emulator Board

**Additional Information**
- Zilog's Superintegration™ Products Guide
- Literature Ordering Guide
- Zilog's Sales Offices, Representatives and Distributors

<table>
<thead>
<tr>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC-8308-01</td>
<td>$ 5.00</td>
</tr>
<tr>
<td>DC-8300-02</td>
<td>$ 5.00</td>
</tr>
</tbody>
</table>
**Z8® MICROCONTROLLERS - PERIPHERALS MULTIMEDIA FAMILY OF PRODUCTS**

<table>
<thead>
<tr>
<th>Databooks By Market Niche</th>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Signal Processor Databook</td>
<td>DC-8299-04</td>
<td>$ 5.00</td>
</tr>
</tbody>
</table>

**Product Specifications**
- Z89321/371 16-Bit Digital Signal Processor (Preliminary)
- Z89C00 16-Bit Digital Signal Processor (Preliminary)
- Z89320 16-Bit Digital Signal Processor (Preliminary)
- Z86C95 Z8® Digital Signal Processor (Preliminary)
- Z89120, Z89920 (ROMless) 16-Bit Mixed Signal Processor (Preliminary)
- Z89121, Z89921 (ROMless) 16-Bit Mixed Signal Processor (Preliminary)

**Application Note**
- Using the Z89371/321 CODEC Interface
- Z89371 Inter Processor Communication
- Understanding Q15 Two's Complement Fractional Multiplication (Z89C00 DSP)

**Support Product Specifications**
- Z8937100ZEM In-Circuit Emulator -C00
- Z8937100TSC Emulation Module
- Z89C0000ZAS Z89C00 Assembler, Linker and Librarian
- Z89C0000ZCC Z89C00 C Cross Compiler
- Z89C0000ZEM In-Circuit Emulator -C00
- Z89C0000ZHP Logic Analyzer Adaptor Board
- Z89C0000ZSD Z89C00 Simulator/Debugger
- Z89C0000ZTR Z89C00 Translator

**Additional Information**
- Zilog's Superintegration™ Products Guide
- Literature Guide and Third Party Support
- Zilog's Sales Offices, Representatives and Distributors
Z8® MICROCONTROLLERS - PERIPHERALS MULTIMEDIA FAMILY OF PRODUCTS

Databooks By Market Niche

<table>
<thead>
<tr>
<th>Product Specifications</th>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZS602/14 NMOS Z8® 8-Bit Keyboard Controller</td>
<td>DC-8304-01</td>
<td>$5.00</td>
</tr>
<tr>
<td>ZS615 NMOS Z8® 8-Bit Keyboard Controller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZS6C15 CMOS Z8® 8-Bit MCU Keyboard Controller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZS6E23 Z8® 8-Bit Keyboard Controller with 8K OTP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZS6C04/C08 CMOS Z8® 8-Bit Microcontroller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZS6E08 CMOS Z8® 8-Bit Microcontroller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZS8C17 CMOS Z8® 8-Bit Microcontroller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZS6C11/717 Z8® 8-Bit Microcontroller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZS6217 Z8® 8-Bit Microcontroller</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Application Notes

- ZS602 Keyboard
- ZS6C17 In-Mouse Applications

Support Product Specifications and Third Party Support

- Z0860200ZCO Evaluation Board
- Z0860200ZDP Adaptor Kit
- ZS6C0800ZCO Evaluation Board
- ZS6C0800ZDP Adaptor Kit
- ZS6C1200ZEM Emulator
- ZS6E2300ZDP Adaptor Kit
- ZS6E2301ZDP Adaptor Kit
- ZS6E2300ZDV Adaptor Kit
- ZS6E2301ZDV Adaptor Kit

Additional Information

- Zilog's Superintegration™ Products Guide
- Literature Guide and Ordering Information
- Zilog's Sales Offices, Representatives and Distributors

PC Audio Databook

<table>
<thead>
<tr>
<th>Product Specifications</th>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZS6321 Digital Audio Processor (Preliminary)</td>
<td>DC-8317-00</td>
<td>$5.00</td>
</tr>
<tr>
<td>ZS9320 16-Bit Digital Signal Processor (Preliminary)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZS9321/371 16-Bit Digital Signal Processor (Preliminary)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZS9331 16-Bit PC ISA Bus Interface (Advance Information)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZS9541/42/43 Wave Synthesis Chip Set (Advance Information)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZS580 Small Computer System Interface</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Additional Information

- Zilog's Superintegration™ Products Guide
- Literature Guide
Z8® MICROCONTROLLERS - PERIPHERALS MEMORY FAMILY OF PRODUCTS

Databooks By Market Niche

<table>
<thead>
<tr>
<th>Mass Storage Solutions</th>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC-8303-01</td>
<td>$5.00</td>
<td></td>
</tr>
</tbody>
</table>

**Product Specifications**
- Z86C21 8K ROM Z8 CMOS Microcontroller
- Z86E21 CMOS Z8 8K OTP Microcontroller
- Z86C91 CMOS Z8 ROMless Microcontroller
- Z86C93 CMOS Z8 Multiply/Divide Microcontroller
- Z86C95 Z8 Digital Signal Processor
- Z86018 Data Path Controller
- Z89C00 16-Bit Digital Signal Processor

**Application Note**
Understanding Q15 Two's Complement Fractional Multiplication (Z89C00 DSP)

**Support Product Specifications**
- Z8060000ZCO Development Kit
- Z86C1200ZEM In-Circuit Emulator
- Z86E2100ZDF Adaptor Kit
- Z86E2100ZDP Adaptor Kit
- Z86E2100ZDV Adaptor Kit
- Z86E2101ZDF Conversion Kit
- Z86E2101ZDV Conversion Kit
- Z86C9300ZEM ICEBOX™ Emulator
- Z86C9500ZCO Evaluation Board
- Z8® S Series Emulators, Base Units and Pods
- Z89C0000ZAS Z89C00 Assembler, Linker and Librarian
- Z89C0000ZCC Z89C00 C Cross Compiler
- Z89C0000ZEM In-Circuit Emulator-C00
- Z89C0000ZSD Z89C00 Simulator/Debugger
- ZPMCMIAOZDP PCMCIA Extender Card

**Additional Information**
- Zilog's Superintegration™ Products Guide
- Zilog's Literature Guide
- Zilog's Sales Offices, Representatives and Distributors
Z8® MICROCONTROLLERS LITERATURE (Continued)

Technical Manuals and Users Guides

<table>
<thead>
<tr>
<th>Part No.</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC-8291-02</td>
<td>5.00</td>
</tr>
<tr>
<td>DC-8296-00</td>
<td>N/C</td>
</tr>
<tr>
<td>DC-8284-01</td>
<td>5.00</td>
</tr>
<tr>
<td>DC-8294-02</td>
<td>5.00</td>
</tr>
<tr>
<td>DC-8595-00</td>
<td>5.00</td>
</tr>
<tr>
<td>DC-8298-03</td>
<td>5.00</td>
</tr>
<tr>
<td>DC-5538-01</td>
<td>N/C</td>
</tr>
</tbody>
</table>

Z8® Application Notes

<table>
<thead>
<tr>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC-2516-01</td>
<td>N/C</td>
</tr>
<tr>
<td>DC-2539-01</td>
<td>N/C</td>
</tr>
<tr>
<td>DC-2541-01</td>
<td>N/C</td>
</tr>
<tr>
<td>DC-2496-01</td>
<td>N/C</td>
</tr>
<tr>
<td>DC-2584-01</td>
<td>N/C</td>
</tr>
<tr>
<td>DC-2592-01</td>
<td>N/C</td>
</tr>
<tr>
<td>DC-2591-01</td>
<td>N/C</td>
</tr>
<tr>
<td>DC-3001-01</td>
<td>N/C</td>
</tr>
<tr>
<td>DC-2604-01</td>
<td>N/C</td>
</tr>
<tr>
<td>DC-2605-01</td>
<td>N/C</td>
</tr>
<tr>
<td>DC-2639-01</td>
<td>N/C</td>
</tr>
<tr>
<td>DC-2645-01</td>
<td>N/C</td>
</tr>
</tbody>
</table>
LITERATURE GUIDE

Z80®/Z8000® DATACOMMUNICATIONS FAMILY OF PRODUCTS

Databooks By Market Niche

High-Speed Serial Communication Controllers

**Product Specifications**
- Z16C30 CMOS Universal Serial Controller (USC™) (Preliminary)
- Z16C32 Integrated Universal Serial Controller (IUSC™) (Preliminary)

**Application Notes**
- Using the Z16C30 Universal Serial Controller with MIL-STD-1553B
- Design a Serial Board to Handle Multiple Protocols
- Datacommunications IUSC™/MUSC™ Time Slot Assigner

**Support Products**
- Z16C3001ZCO Evaluation Board Product Specification
- Z8018600ZCO Evaluation Board Product Specification

**Additional Information**
- Zilog’s Superintegration™ Products Guide
- Literature Guide
- Third Party Support Vendors

---

Serial Communication Controllers

**Product Specifications**
- Z8030/Z8530 Z-Bus® SCC Serial Communication Controller
- Z80C30/Z85C30 CMOS Z-Bus® SCC Serial Communication Controller
- Z80230 Z-Bus® ESCC™ Enhanced Serial Communication Controller (Preliminary)
- Z85230 ESCC™ Enhanced Serial Communication Controller
- Z85233 EMSCC™ Enhanced Mono Serial Communication Controller
- Z85C80 SCSCI™ Serial Communications and Small Computer Interface
- Z16C35/Z85C35 CMOS ISCC™ Integrated Serial Communications Controller

**Application Notes**
- Interfacing Z8500 Peripherals to the 68000
- SCC In Binary Synchronous Communications
- Zilog SCC Z8030/Z8530 Questions and Answers
- Integrating Serial Data and SCSI Peripheral Control on One Chip
- Zilog ISCC™ Controller Questions and Answers
- Boost Your System Performance Using the Zilog ESCC™
- Zilog ESCC™ Controller Questions and Answers
- The Zilog Datacom Family with the 80186 CPU
- On-Chip Oscillator Design

**Support Products**
- Z8S18000ZCO Evaluation Board Product Specification
- Z8523000ZCO Evaluation Board Product Specification
- Z8018600ZCO Evaluation Board Product Specification
- ZEPMDC00002 Electronic Programmer's Manual Software

**Additional Information**
- Zilog’s Superintegration™ Products Guide
- Literature Guide
Z80®/Z8000® Datacommunications Family of Products

### Databooks

<table>
<thead>
<tr>
<th>Databook</th>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z80 Family Databook</td>
<td>DC-8321-00</td>
<td>5.00</td>
</tr>
</tbody>
</table>

#### Discrete Z80® Family
- Z8400/C00 NMOS/CMOS Z80® CPU Product Specification
- Z8410/C10 NMOS/CMOS Z80 DMA Product Specification
- Z8420/C20 NMOS/CMOS Z80 PIO Product Specification
- Z8430/C30 NMOS/CMOS Z80 CTC Product Specification
- Z8440/C40 NMOS/CMOS Z80 SIO Product Specification

#### Embedded Controllers
- Z84C01 Z80 CPU with CGC Product Specification
- Z8470 Z80 DART Product Specification
- Z84C90 CMOS Z80 KIO™ Product Specification
- Z84013/015 Z84C13/C15 IPC/EIPC Product Specification

#### Application Notes and Technical Articles
- Z80® Family Interrupt Structure
- Using the Z80® SIO with SDLC
- Using the Z80® SIO in Asynchronous Communications
- Binary Synchronous Communication Using the Z80® SIO
- Serial Communication with the Z80A DART
- Interfacing Z80® CPUs to the Z8500 Peripheral Family
- Timing in an Interrupt-Based System with the Z80® CTC
- A Z80-Based System Using the DMA with the SIO
- Using the Z84C11/C13/C15 in Place of the Z84011/013/015
- On-Chip Oscillator Design
- A Fast Z80® Embedded Controller
- Z80® Questions and Answers

#### Additional Information
- Zilog's Superintegration™ Products Guide
- Literature Guide
- Third Party Support Vendors
- Zilog's Sales Offices, Representatives and Distributors
Z80®/Z8000® Datacommunications Family of Products

Databooks

<table>
<thead>
<tr>
<th>Product Specification</th>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z80180/Z8S180/Z8L180 Z180™ Microprocessor</td>
<td>DC-8322-00</td>
<td>5.00</td>
</tr>
<tr>
<td>Z80181 Z181™ Smart Access Controller (SAC™)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z80182/Z8L182 Zilog Intelligent Peripheral Controller (ZIP™)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Application Notes and Technical Articles

- Z180™ Questions and Answers
- Z180™/SCC Serial Communication Controller Interface at 10 MHz
- Interfacing Memory and I/O to the 20 MHz Z8S180 System
- Break Detection on the Z80180 and Z181™
- Z182 Programming the MIMIC Autoecho ECHOZ182 Sample Code
- Local Talk Link Access Protocol Using the Z80181

Support Products

- Z8S18000ZC0 Evaluation Board
- Z8018100ZC0 Evaluation Board
- Z8018101ZC0 Evaluation Board
- Z8018101ZA6 Driver Software
- Z8018100ZDP Adaptor Board
- Z8018200ZC0 Evaluation Board
- Z80® and Z80180 Hardware and Software Support
- Third Party Support Vendors

Additional Information

- Zilog's Superintegration™ Products Guide
- Literature Guide
- Zilog's Sales Offices, Representatives and Distributors
# LITERATURE GUIDE

## Z80®/Z8000® DATACOMMUNICATIONS FAMILY OF PRODUCTS

### Databooks and User's Manuals

<table>
<thead>
<tr>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC-8319-00</td>
<td>5.00</td>
</tr>
</tbody>
</table>

#### Z8000 Family of Products

**Z8000 Family Databook**
- Zilog's Z8000 Family Architecture
- Z8001/Z8002 Z8000 CPU Product Specification
- Z8016 Z8000 Z-DTC Product Specification
- Z8036 Z8000 Z-CIO Product Specification
- Z8536 CIO Counter/Timer and Parallel I/O Unit Product Specification
- Z8038/Z8538 FIO FIFO Input/Output Interface Unit Product Specification
- Z8060/Z8560 FIFO Buffer Unit
- Z8581 Clock Generator and Controller Product Specification

#### User's Manuals

- Z8000 CPU Central Processing Unit User's Manual
- Z8010 Memory Management Unit (MMU) User's Manual
- Z8000 Application Notes and Military Products

#### Application Notes

- Using SCC with Z8000 in SDLC Protocol
- SCC in Binary Synchronous Communication
- Zilog's Military Products Overview

#### Additional Information

- Zilog's Superintegration™ Products Guide
- Literature Guide
- Zilog's Sales Offices, Representatives and Distributors

<table>
<thead>
<tr>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC-8309-00</td>
<td>5.00</td>
</tr>
<tr>
<td>DC-8276-04</td>
<td>5.00</td>
</tr>
<tr>
<td>DC-8224-03</td>
<td>5.00</td>
</tr>
<tr>
<td>DC-6003-03</td>
<td>N/C</td>
</tr>
<tr>
<td>DC-8297-03</td>
<td>5.00</td>
</tr>
<tr>
<td>DC-6021-00</td>
<td>N/C</td>
</tr>
<tr>
<td>DC-8315-00</td>
<td>N/C</td>
</tr>
</tbody>
</table>

### Z80 Family Technical Manual
- Z80180 Z180 MPU Microprocessor Unit Technical Manual
- Z280 MPU Microprocessor Unit Technical Manual
- Z380™ Preliminary Product Specification
- Z2000 Spread-Spectrum Transceiver Advance Information Product Specification
- ZNW2000 User's Manual for PC WAN Adaptor Board Development Kit

<table>
<thead>
<tr>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC-8293-02</td>
<td>5.00</td>
</tr>
<tr>
<td>DC-8280-04</td>
<td>5.00</td>
</tr>
<tr>
<td>DC-8292-02</td>
<td>5.00</td>
</tr>
<tr>
<td>DC-8286-01</td>
<td>5.00</td>
</tr>
<tr>
<td>DC-8286-01A</td>
<td>N/C</td>
</tr>
</tbody>
</table>
## MILITARY COMPONENTS FAMILY

<table>
<thead>
<tr>
<th>Military Product Specifications</th>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z8681 ROMless Microcomputer</td>
<td>DC-2392-02</td>
<td>N/C</td>
</tr>
<tr>
<td>Z8001/8002 Military Z8000 CPU Central Processing Unit</td>
<td>DC-2342-03</td>
<td>N/C</td>
</tr>
<tr>
<td>Z8581 Military CGC Clock Generator and Controller</td>
<td>DC-2346-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Z8030 Military Z8000 Z-SCC Serial Communications Controller</td>
<td>DC-2388-02</td>
<td>N/C</td>
</tr>
<tr>
<td>Z8530 Military SCC Serial Communications Controller</td>
<td>DC-2397-02</td>
<td>N/C</td>
</tr>
<tr>
<td>Z8036 Military Z8000 Z-CIO Counter/Timer Controller and Parallel I/O</td>
<td>DC-2389-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Z8038/8538 Military FIO FIFO Input/Output Interface Unit</td>
<td>DC-2463-02</td>
<td>N/C</td>
</tr>
<tr>
<td>Z8536 Military CIO Counter/Timer Controller and Parallel I/O</td>
<td>DC-2396-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Z8400 Military Z80 CPU Central Processing Unit</td>
<td>DC-2351-02</td>
<td>N/C</td>
</tr>
<tr>
<td>Z8420 Military PIO Parallel Input/Output Controller</td>
<td>DC-2384-02</td>
<td>N/C</td>
</tr>
<tr>
<td>Z8430 Military CTC Counter/Timer Circuit</td>
<td>DC-2385-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Z8440/1/2/4 Z80 SIO Serial Input/Output Controller</td>
<td>DC-2386-02</td>
<td>N/C</td>
</tr>
<tr>
<td>Z80C30/65C30 Military CMOS SCC Serial Communications Controller</td>
<td>DC-2478-02</td>
<td>N/C</td>
</tr>
<tr>
<td>Z84C00 CMOS Z80 CPU Central Processing Unit</td>
<td>DC-2441-02</td>
<td>N/C</td>
</tr>
<tr>
<td>Z84C20 CMOS Z80 PIO Parallel Input/Output</td>
<td>DC-2384-02</td>
<td>N/C</td>
</tr>
<tr>
<td>Z84C30 CMOS Z80 CTC Counter/Timer Circuit</td>
<td>DC-2481-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Z84C40/1/2/4 CMOS Z80 SIO Serial Input/Output</td>
<td>DC-2482-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Z16C30 CMOS USC Universal Serial Controller (Preliminary)</td>
<td>DC-2531-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Z80180 Z180 MPU Microprocessor Unit</td>
<td>DC-2538-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Z84C90 CMOS KIO Serial/Parallel/Counter Timer (Preliminary)</td>
<td>DC-2502-00</td>
<td>N/C</td>
</tr>
<tr>
<td>Z85230 ESCC Enhanced Serial Communication Controller</td>
<td>DC-2595-00</td>
<td>N/C</td>
</tr>
</tbody>
</table>
### General Literature

<table>
<thead>
<tr>
<th>Catalogs, Handbooks, Product Flyers and Users Guides</th>
<th>Part No</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Superintegration Master Selection Guide 1994-1995</td>
<td>DC-5634-00</td>
<td>N/C</td>
</tr>
<tr>
<td>Superintegration Products Guide</td>
<td>DC-5676-00</td>
<td>N/C</td>
</tr>
<tr>
<td>Quality and Reliability Report</td>
<td>DC-8329-00</td>
<td>N/C</td>
</tr>
<tr>
<td>ZIA® 3.3-5.5V Matched Chip Set for AT Hard Disk Drives Datasheet</td>
<td>DC-5556-01</td>
<td>N/C</td>
</tr>
<tr>
<td>ZIA ZIA00ZCO Disk Drive Development Kit Datasheet</td>
<td>DC-5593-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog Hard Disk Controllers - Z86C93/C95 Datasheet</td>
<td>DC-5560-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog Infrared (IR) Controllers - ZIRC™ Datasheet</td>
<td>DC-5558-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog V. Fast Modem Controller Solutions</td>
<td>DC-5525-02</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog Digital Signal Processing - Z89320 Datasheet</td>
<td>DC-5547-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog Keyboard Controllers Datasheet</td>
<td>DC-5600-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Z80® - Next Generation Z80®/Z180® Datasheet</td>
<td>DC-5580-02</td>
<td>N/C</td>
</tr>
<tr>
<td>Fault Tolerant Z8® Microcontroller Datasheet</td>
<td>DC-5603-01</td>
<td>N/C</td>
</tr>
<tr>
<td>32K ROM Z8® Microcontrollers Datasheet</td>
<td>DC-5601-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog Datacommunications Brochure</td>
<td>DC-5519-00</td>
<td>N/C</td>
</tr>
<tr>
<td>Z89300 DTC Controller Family Brochure</td>
<td>DC-5608-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog Digital Signal Processing Brochure</td>
<td>DC-5536-02</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog ASSPs - Partnering With You Product Brochure</td>
<td>DC-5553-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog Wireless Products Datasheet</td>
<td>DC-5630-00</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog Z8604 Cost Efficient Datasheet</td>
<td>DC-5662-00</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog Chip Carrier Device Packaging Datasheet</td>
<td>DC-5672-00</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog Database of IR Codes Datasheet</td>
<td>DC-5631-00</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog PCMCIA Adaptor Chip Z86017 Datasheet</td>
<td>DC-5585-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog Television/Video Controllers Datasheet</td>
<td>DC-5567-01</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog TAD Controllers - Z89C65/C67/C69 Datasheet</td>
<td>DC-5561-02</td>
<td>N/C</td>
</tr>
<tr>
<td>Zilog Z87000 Z-Phone Datasheet</td>
<td>DC-5632-00</td>
<td>D/C</td>
</tr>
<tr>
<td>Zilog 1993 Annual Report</td>
<td>DC-1993-AR</td>
<td>N/C</td>
</tr>
</tbody>
</table>
**ORDERING INFORMATION**

Complete the attached literature order form. Be sure to enclose the proper payment or supply a purchase order. Please reference specific order requirements.

**MINIMUM ORDER REQUIREMENTS**

Orders under $300.00 must be prepaid by check, money order or credit card. Canadian and foreign orders must be accompanied by a cashier's check in U.S. dollars, drawn on a correspondent U.S. bank only. Orders over $300.00 may be submitted with a Purchase Order.

**SHIPMENT**

Orders will be shipped after your check is cashed or credit is checked via the most economical method. Please allow four weeks for delivery.

RETURNS ARE NOT ACCEPTED.

---

<table>
<thead>
<tr>
<th>NAME</th>
<th>COMPANY</th>
<th>ADDRESS</th>
<th>CITY</th>
<th>STATE</th>
<th>ZIP</th>
<th>COUNTRY</th>
</tr>
</thead>
</table>

**PLEASE PRINT OR TYPE**

<table>
<thead>
<tr>
<th>PHONE ( )</th>
<th>METHOD OF PAYMENT (CHECK ONE)</th>
<th>COUNTRY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>□ Check □ Money Order</td>
<td></td>
</tr>
<tr>
<td>Credit Card □ VISA □ MC □ P.O. (over $300.00)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PART NUMBER</th>
<th>DOCUMENT TITLE</th>
<th>UNIT COST</th>
<th>QTY.</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$</td>
<td>$</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$</td>
<td>$</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$</td>
<td>$</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$</td>
<td>$</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$</td>
<td>$</td>
<td>$</td>
</tr>
</tbody>
</table>

**Mail To:**

ZiLOG
210 E. HACIENDA AVE. M/S C1-0
CAMPBELL, CA 95008-6600
Phone: (408)376-8016
Fax: (408)376-8056

Credit Card or Purchase Order # ________________ SUBTOTAL
Expiration Date ___________ ADD APPLICABLE SALES TAX (CA ONLY)
Signature ________________ ADD 10% SHIPPING AND HANDLING

TOTAL
Thank you for your interest in Zilog products. To receive prompt attention and information, please fill out this reply card and mail it to Zilog.

Please send me more information on Zilog products in the following areas:

- General-Purpose
- Mass Storage
- Low Voltage Controllers
- Wireless Communications
- Multimedia
- Fax/Modem Control
- Datacommunications
- Speech Processing/Synthesis

The primary product/industry in which my company is involved is: ____________________________

My specific application is: ____________________________________________________________

My need is:

- Immediate
- Within 6-12 months
- Long-term
- For information only

Name __________________________________________ Title ________________________________

Company ____________________________________________________

Address __________________________________________________________________________

City ____________________________ ST _____ ZIP _________

Telephone (___)_____________ Ext. _________

Fax (___)_____________________

Please tell us where you heard about Zilog:

- Colleague's recommendation
- Direct sales call
- Advertisement (where?____________________)
- Prior experience with company
- Distributor or rep
- Conference Presentation (where?__________