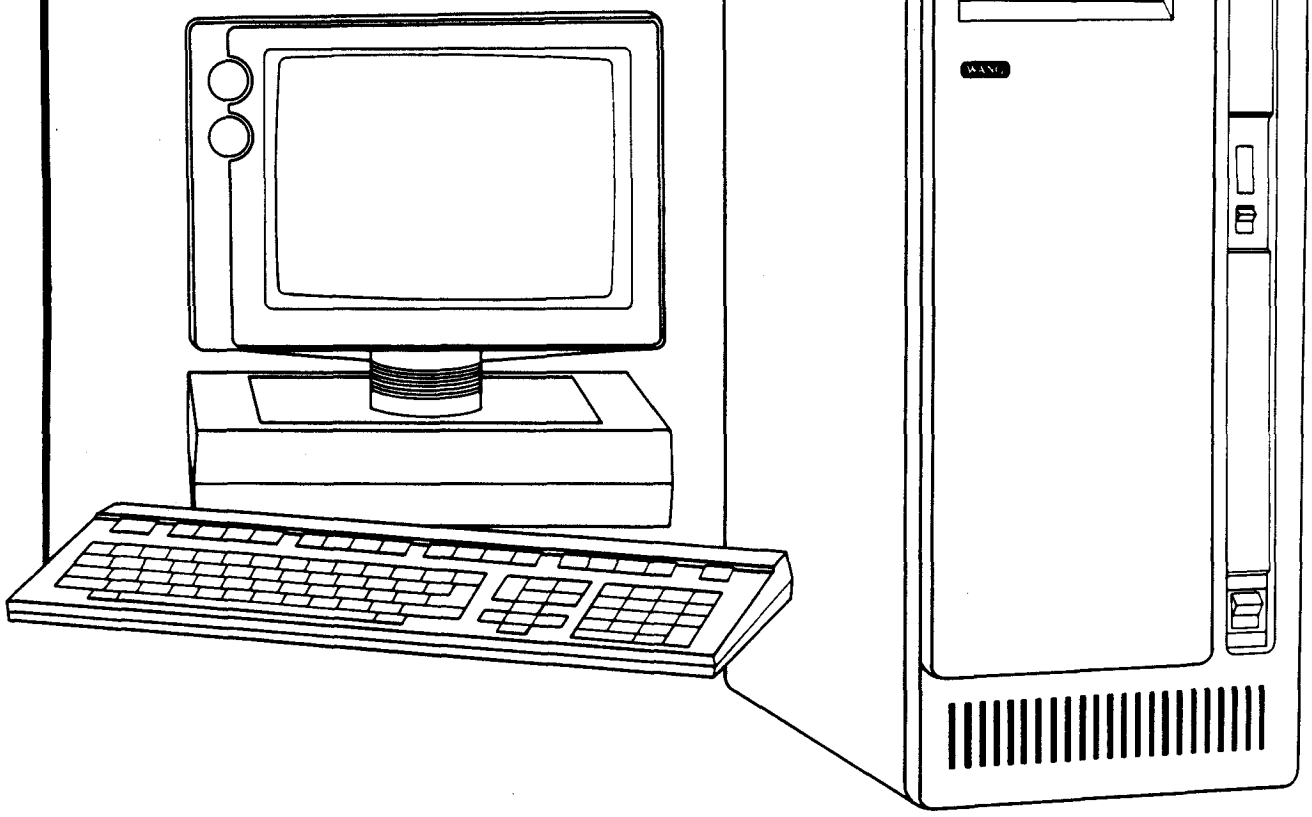
WANG

1403

OIS 40/50/60

BOARD REPAIR WORKBOOK VOLUME 2



CUSTOMER ENGINEERING
TRAINING AND DOCUMENTATION

741-9041

WANG

CUSTOMER ENGINEERING TRAINING CENTER

OIS 40/50/60
BOARD REPAIR WORKBOOK VOLUME 2

PREFACE

This document is intended to be used for TRAINING PURPOSES only. The material contained in this document, while accurate during the developement of this workbook, may not reflect the latest developments or changes to the OIS 40/50/60 system.

TECHNICAL SUPPORT DOCUMENTS

OIS 50 INTERNAL PRINTER CONTROLLER HARDWARE SPECIFICATIONS HM-60
OIS 40/50 RESOURCE MANAGEMENT UNIT THEORY OF OPERATION 751-0902
OIS 40/50 RESOURCE CONTROL UNIT THEORY OF OPERATION 751-0911
OIS 40/50 IWS FULL MATRIX CONTROLLER SPECIFICATION HM-67
WL-2630 OIS/VS COLLECTIVE GATE ARRAY SPECIFICATION HM-37
OIS-50 INTERNAL WISE SPECIFICATION REVISION 3 HM-85
OFFICE INFORMATION SYSTEMS OIS 40/50/60 741-1267
OIS SYSTEM ADMINISTRATION GUIDE 700-5562E

First Edition - December, 1985

OUTLINE OF WORKBOOK

	SECTION #	TITLE
VOLUME 1		
	1.	Introduction to System
	2.	Resource Management Unit (RMU)
	3.	Resource Control Unit (RCU)
	4.	Internal Workstation Controller (IWS)
	5.	Appendices (A-E)
VOLUME 2		
	6.	Internal Printer Controller (IPC)
	7.	Internal WISE Controller (IWISE)
	8.	Diagnostics
	9.	Appendices (F-K)

TABLE OF CONTENTS VOL I

CHAPTER Page

SECTION 1

OIS 40/50/60 WORKBOOK INTRODUCTION

1.1 OIS 40/50/60 SYSTEM INTRODUCTION 1-2

1.2 SYSTEM OPERATION 1-3
1.2.1 Resource Management Unit (RMU) 1-5
1.2.2 Resource Control Unit (RCU) 1-5
1.2.3 The Internal Workstation Controller (IWS) 1-7
1.2.4 The Internal Printer Controller (IPC) 1-8
1.2.5 The Internal WISE Controller (IWISE) 1-8

SECTION 2

Resource Management Unit (RMU)

2.1 Resource Management Unit (RMU) 2-1
2.1.1 Controls and Indicators 2-5
2.1.2 Clock Generation 2-13
2.1.3 Memory Mapped Input/Output 2-14

2.2 Z80A CENTRAL PROCESSOR UNIT 2-18
2.2.1 Z80A CONTROL LINES 2-18
2.2.2 ADDRESS BUS 2-22
2.2.3 DATA BUS 2-22

2.3 COUNTER TIMER CHIP 2-24

2.4 MEMORY 2-26
2.4.1 Z80A and Direct Memory Access Paths 2-26
2.4.2 Memory Parity Generator/Checker 2-30
2.4.3 Memory Refresh 2-33
2.4.4 RAS/CAS Signal Generator 2-35
2.4.5 RAS/CAS Timing 2-37
2.4.6 Read/Write Signal Generator 2-38
2.4.7 Programmable Read Only Memory 2-38

2.5 FLOPPY DISK CONTROLLER 2-40
2.5.1 Read/Write Operation 2-40
2.5.2 Read/Write Enabling 2-43
2.5.3 Control Signal Generator 2-45
2.5.4 Write Data Precompensation 2-46
2.5.5 Motor Control 2-47
2.5.6 Terminal Count 2-47
2.5.7 DMA Acknowledge 2-48
2.5.8 Deadman Timer 2-48
2.5.9 Floppy Disk Controller Reset 2-49
2.5.10 Floppy Disk Drive and Controller Status 2-49
2.5.11 Data Recovery 2-52

CHAPTER		Page
2.6	SERIAL DATA LINK	2-56
2.6.1	Command Decoder	2-56
2.6.2	Receive Operation	2-59
2.6.3	Transmit Operations	2-62
2.6.4	Status Register	2-66

SECTION 3

Resource Control Unit (RCU)

3.1	Resource Control Unit (RCU)	3-1
3.1.1	GENERAL INFORMATION	3-1
3.1.2	RMU-RCU Interaction	3-3
3.1.3	RMU-RCU Interface	3-5
3.2	8X305 MICROCONTROLLER	3-9
3.2.2	Microcontroller Instruction Storage Area	3-13
3.2.3	Left Bank Scratchpad Memory	3-13
3.2.4	Right Bank I/O Decoder Operations	3-19
3.3	RIGHT BANK OPERATIONS	3-22
3.3.1	Status Register File Control I/O Commands	3-22
3.3.2	Parameter Register File Control I/O Comma	3-24
3.3.3	Command Notification Bit Issuance I/O Com	3-25
3.3.4	4K x 8 Data Buffer Control I/O Commands	3-26
3.3.5	50BUS Control I/O Commands	3-27
3.3.6	Winchester Control I/O Commands	3-30
3.3.7	Interrupt Issuance I/O Command	3-35
3.4	4K x 8 DATA BUFFER and 50BUS INTERFACE	3-36
3.4.1	4K x 8 Data Buffer	3-36
3.4.2	Data Buffer I/O Command Functions	3-39
3.4.3	50BUS Interface	3-40
3.4.4	50BUS I/O Command Functions	3-42
3.4.5	Interfacing to a Parallel Device on the 5	3-46
3.4.6	Microcontroller Halt Circuit	3-48
3.5	WINCHESTER DISK DRIVE INTERFACE	3-51
3.5.1	Winchester I/O Command Functions	3-51
3.5.2	Winchester Data Format	3-59
3.5.3	Winchester Write Operation	3-61
3.5.4	Winchester Read Operation	3-63
3.5.5	ECC Generator/Checker	3-65
3.5.6	Winchester Step Logic	3-69
3.5.7	Dead Man Timer	3-70
3.5.8	Phase Locked Loop	3-71
3.5.9	Data Recovery	3-73
3.5.10	Data Recovery Logic Reset Control	3-76

TABLE OF CONTENTS (cont.)

CHAPTER		Page
3.6	STATUS AND PARAMETER REGISTER FILES	3-78
3.6.1	SRF-PRF Control	3-78
3.6.2	Status Register File	3-79
3.6.3	Microcontroller Writes to the SRF	3-80
3.6.4	RMU Reads from the SRF	3-82
3.6.5	Parameter Register File	3-82
3.6.6	Microcontroller Accesses the PRF	3-83
3.6.7	RMU Accesses the PRF	3-86

SECTION 4

Internal Workstation Controller (IWS)

4.1	The CPU and Support Logic.	4-4
4.1.1	Clock Generation	4-4
4.1.2	Z80A Control Inputs	4-4
4.1.3	Control Outputs	4-9
4.1.4	Address Bus	4-9
4.1.5	Data Bus	4-10
4.1.6	MMI/O Decoders	4-10
4.2	Main Memory and Control Logic	4-12
4.2.1	Addressing Main Memory	4-12
4.2.2	Data Transfers	4-14
4.2.3	Parity Circuits	4-15
4.3	Display Memory and Control Logic	4-17
4.3.1	Character/Control Memory	4-19
4.3.2	Font Memory	4-22
4.4	Display Timing	4-24
4.4.1	Display Memory Access	4-29
4.4.2	Display Modification Logic	4-31
4.5	Keyboard Interface	4-37
4.5.1	The 8031 Microcontroller	4-37
4.5.2	Receive Operations	4-39
4.5.3	Transmit Operation	4-39
4.5.4	The 8031 Instruction Cycle	4-40
4.6	OIS 40/50/60 Bus Interface Logic	4-41
4.6.1	Read operation	4-41
4.6.2	Write Operation	4-41
4.6.3	Status Information	4-42
4.6.4	Block Transfers	4-42

TABLE OF CONTENTS (cont.)

CHAPTER Page

SECTION 5
APPENDICIES

A RELATED DOCUMENTATION A-1

B. OIS 40/50/60 MNEMONICS
- Part 1: Resource Management Unit (RMU) B-1
- Part 2: Resource Control Unit (RCU) B-11
- Part 3: Internal Workstation Controller (IWS) B-26

C CHIP LIST
- Part 1: Resource Management Unit (RMU) C-1
- Part 2: Resource Control Unit (RCU) C-15
- Part 3: Internal Workstation Controller (IWS) C-29

D QUIZ ANSWERS
- Section 1 D-1
- Section 2 D-2
- Section 3 D-3
- Section 4 D-4

E SCHEMATICS
Resource Management Unit (RMU)
Resource Control Unit (RCU)
Internal Workstation Controller (IWS)
Mother Board

TABLE OF CONTENTS VOL II

CHAPTER Page

SECTION 6
Internal Printer Controller (IPC)

6.1 INTRODUCTION 6-1

6.2 60BUS Interface 6-3

6.3 Z80A Microprocessor 6-3
6.3.1 NOP Generation 6-3
6.3.2 I/O Operations 6-5

6.4 IPC Memory 6-9

6.5 RS-232C Interface 6-14

6.6 Troublshooting the IPC board 6-20

SECTION 7
Internal WISE Controller (IWISE)

7.1 INTRODUCTION 7-1

7.2 Overview 7-1

7.3 IWISE Protocol 7-4

7.4 Command Structure 7-8
7.4.1 Status Read 7-8
7.4.2 One Byte Read Sequence 7-10
7.4.3 Block Read Sequence 7-11
7.4.4 One Byte Write Sequnce 7-12
7.4.5 Block Write Sequence 7-12
7.4.6 Restart Command sequence 7-13

7.5 Detailed Theroy of Operation 7-14
7.5.1 Central Processing Unit 7-14

7.6 Serial Data Link Protocol Logic 7-20
7.6.1 SDL Receive 7-21
7.6.2 Command Decoder 7-23
7.6.3 Command Sequencer 7-25
7.6.4 Transmit Circuits 7-27
7.6.5 Termination of Transmission 7-32

7.7 50BUS Protocol Logic 7-33

7.8 Access Control 7-35

TABLE OF CONTENTS (cont.)

CHAPTER Page

SECTION 7 (cont.)

7.9 IWISE Memory 7-38
7.9.1 Zero Page Access 7-39

7.10 Hardware Control Ports 7-40

SECTION 8
DIAGNOSTICS

8.1 INTRODUCTION 8-1

8.2 System Diagnostics 8-1
8.2.1 MASTER DTOS 8-2
8.2.2 ONLINE DTOS 8-6
8.2.3 B.I.T. 8-6
8.2.4 SYSEX 8-7

8.3 Individual Board Diagnostic 8-9

CHAPTER Page

SECTION 9
APPENDICIES

F Part 1:
8X305 COMMAND SEQUENCES FOR SDL OPERATIONS F-1
Part 2:
8X305 MICROCONTROLLER COMMAND SEQUENCES F-19

G. RMU - RCU COMMUNICATION PROTOCOL
Introduction G-1
Master Unit Commands G-2
Floppy Unit Commands G-9
Winchester Unit Commands G-11
RCU Port Commands G-20
Status Register File Contents G-22

H MNEMONICS LIST
Part 1: Internal Printer Controller (IPC) H-1
Part 2: Internal WISE Controller (IWISE) H-3

I CHIP LIST
Part 1: Internal Printer Controller (IPC) I-1
Part 2: Internal WISE Controller (IWISE) I-4

J QUIZ ANSWERS
Section 6 J-1
Section 7 J-2

K SCHEMATICS
Internal Printer Controller (IPC)
Internal WISE Controller (IWISE)

LIST OF ILLISTRATIONS

FIGURE NUMBER Page

SECTION 1
OIS 40/50/60 WORKBOOK INTRODUCTION

FIGURE 1-1 OIS 40/50/60 Simplified Block Diagram 1-4

SECTION 2
Resource Management Unit (RMU)

FIGURE 2.1-1 Block Diagram of the Resource Management Unit 2-2

FIGURE 2.1-2 OIS 50 Block Diagram 2-3

FIGURE 2.1-3 RMU CLOCK LOGIC 2-13

FIGURE 2.1-4 RUM TIMING 2-13

FIGURE 2.1-5 MMI/O Logic 2-14

FIGURE 2.2-1 Z80A Access Logic 2-21

FIGURE 2.3-1 Counter Timer Chip Access Logic 2-24

FIGURE 2.4-1 Address Bus and Data Bus Circuitry 2-27

FIGURE 2.4-2 Memory Access Circuitry 2-28

FIGURE 2.4-3 Parity Generator/Checker Logic 2-31

FIGURE 2.4-4 Op-Code Fetch / Memory Refresh Timing Diagram 2-33

FIGURE 2.4-5 RAS/CAS and WRITE Generating Logic 2-36

FIGURE 2.4-6 Memory Read/Write Timing Diagram 2-37

FIGURE 2.4-7 Prom Access Logic 2-39

FIGURE 2.5-1 FDC Access Block Diagram 2-41

FIGURE 2.5-2 FDC Access Logic 2-42

FIGURE 2.5-3 FDC Data Path Logic 2-44

FIGURE 2.5-4 FDC Status Logic 2-51

FIGURE 2.5-5 Data Recovery Block Diagram 2-53

FIGURE 2.5-6 Data Recovery Logic 2-54

FIGURE 2.6-1 SDL Operation Control Logic 2-58

FIGURE 2.6-2 SDL Receive Block Diagram 2-59

LIST OF ILLISTRATIONS (cont.)

FIGURE NUMBER		Page
FIGURE 2.6-3	SDL Receive Logic	2-60
FIGURE 2.6-4	SDL Receive Timing	2-61
FIGURE 2.6-5	SDL Transmit Block Diagram	2-62
FIGURE 2.6-6	SDL Transmit Logic	2-64
FIGURE 2.6-7	SDL Transmit Timing Diagram	2-65

SECTION 3
Resource Control Unit (RCU)

FIGURE 3.1-1	Block Diagram of the Resource Control Unit	3-2
FIGURE 3.1-2	RMU / RCU Interface Block Diagram	3-6
FIGURE 3.2-1	8X305 Microcontroller Block Diagram.	3-10
FIGURE 3.2-2	8X305 Microcontroller Access Logic	3-12
FIGURE 3.2-3	Scratchpad Memory Logic	3-18
FIGURE 3.3-1	Right Bank Decoding Block Diagram	3-23
FIGURE 3.3-2	Command Bus Request Bit Control Logic.	3-25
FIGURE 3.4-1	4K x 8 Data Buffer Block Diagram.	3-37
FIGURE 3.4-2	4K x 8 Data Buffer Access Logic.	3-38
FIGURE 3.4-3	50BUS Control Logic.	3-41
FIGURE 3.4-4	Parallel Device Restart Timing Diagram.	3-47
FIGURE 3.4-5	Parallel Device Status Timing Diagram.	3-47
FIGURE 3.4-6	50BUS Handshaking Timing Diagram.	3-47
FIGURE 3.4-7	Parallel Device Memory Access Timing Diagram.	3-49
FIGURE 3.4-8	Block Read/Write 50BUS Request Timing Diagram.	3-49
FIGURE 3.5-1	Winchester Interface Control Logic.	3-52
FIGURE 3.5-2	Winchester Interface Logic.	3-53
FIGURE 3.5-3	Winchester Disk Data Format.	3-59

LIST OF ILLISTRATIONS (cont.)

FIGURE NUMBER		Page
FIGURE 3.5-4	Winchester Write Block Diagram.	3-64
FIGURE 3.5-5	Winchester Read Block Diagram.	3-64
FIGURE 3.5-6	ECC Generator/Checker Logic.	3-66
FIGURE 3.5-7	Winchester Step Generating Logic.	3-69
FIGURE 3.5-8	Phase Locked Loop Logic.	3-72
FIGURE 3.5-9	Data Recovery Logic.	3-74
FIGURE 3.6-1	Status Register File Block Diagram.	3-80
FIGURE 3.6-2	Status Register File Logic.	3-81
FIGURE 3.6-3	Parameter Register File Block Diagram.	3-83
FIGURE 3.6-4	Parameter Register File Logic.	3-84

SECTION 4
Internal Workstation Controller (IWS)

FIGURE 4-1	Internal Workstation Controller Block Diagram	4-2
FIGURE 4.1-1	Clock Timing Diagram	4-5
FIGURE 4.1-2	Clock Logic	4-6
FIGURE 4.1-3	Wait State Timing Logic	4-8
FIGURE 4.2-1	RAS/CAS Generating Logic	4-13
FIGURE 4.2-2	Parity Generating/Checking Logic	4-16
FIGURE 4.3-1	FONT MATRIX	4-22
FIGURE 4.4-1	Display Timing	4-28
FIGURE 4.4-2	FONT MATRIX	4-35
FIGURE 4.4-3	Line Count to Address Relationship	4-35
FIGURE 4.5-1	8031 Block Diagram	4-38

LIST OF ILLISTRATIONS (cont.)

FIGURE NUMBER Page

SECTION 6
Internal Printer Controller (IPC)

FIGURE 6.1-1 Internal Printer Controller Block Diagram 6-2

FIGURE 6.3-1 I/O Decoders 6-7

FIGURE 6.4-1 REFRESH LOGIC DIAGRAM 6-10

FIGURE 6.4-2 M1 Timing Diagram 6-10

FIGURE 6.4-3 RAS/CAS LOGIC DIAGRAM 6-12

FIGURE 6.5-1 SC2661C Block Diagram 6-15

SECTION 7
Internal WISE Controller (IWISE)

FIGURE 7.2-1 IWISE Block Diagram 7-2

FIGURE 7.3-1 Typicial Transmitted Word 7-6

FIGURE 7.10-1 J1 Pin Assignments 7-44

LIST OF TABLES

TABLE NUMBER Page

SECTION 2
Resource Management Unit (RMU)

Table 2.1-1 LED DISPLAY CODES 2-6

Table 2.1-2 POWER-UP DIAGNOSTIC DISPLAY ERROR 2-7

Table 2.1-3 SYSTEM DISPLAY ERROR CODES 2-11

Table 2.1-4 SOFTWARE CONFIGURATION SWITCH 2-12

Table 2.1-5 RMU MEMORY MAPPED I/O COMMANDS 2-15

Table 2.5-1 WRITE DATA/PRECOMPENSATION SELECT 2-47

Table 2.5-2 FLOPPY DISK DRIVE STATUS BYTE 2-50

Table 2.6-1 SERIAL DATA LINK COMMANDS 2-57

Table 2.6-2 SERIAL DATA LINK STATUS BYTE 2-66

SECTION 3
Resource Control Unit (RCU)

Table 3.1-1 Z80A-GENERATED I/O COMMANDS 3-8

Table 3.2-1 MICROCONTROLLER CONTROL SIGNAL 3-11

Table 3.2-2 SCRATCHPAD MEMORY MAP 3-14

Table 3.2-3 RCU RIGHT BANK I/O DECODER 3-20

Table 3.3-1 STATUS REGISTER FILE I/O COMMANDS 3-24

Table 3.3-2 4K x 8 DATA BUFFER I/O COMMANDS 3-26

Table 3.3-3 50BUS I/O COMMANDS 3-28

Table 3.3-4 WINCHESTER I/O COMMANDS 3-31

LIST OF TABLES (cont.)

TABLE NUMBER		Page
Table 3.5-1	WINCHESTER STATUS BYTE	3-54
Table 3.5-2	WINCHESTER PROGRAM BYTE	3-55
Table 3.5-3	WINCHESTER CONTROL STATUS BYTE	3-56
Table 3.5-4	WINCHESTER INTERFACE CONTROL BYTE	3-57
Table 3.5-5	WINCHESTER DATA MOTION CONTROL	3-59

SECTION 4
Internal Workstation Controller (IWS)

Table 4.1.1	MMI/O COMMANDS	4-10
Table 4.3.1	Main Memory Overlay Assigments	4-18
Table 4.3.2	Display Attribute Decoding	4-20
Table 4.3.3	Attribute Control Bits	4-21
Table 4.4-1	WL2632 Display Signals	4-25
Table 4.6-1	Status Information	4-42

LIST OF TABLES (cont.)

TABLE NUMBER		Page
	SECTION 6 **Internal Printer Controller (IPC)**	
Table 6.3-1	Device and Diagnostic I/O Operations	6-5
Table 6.3-2	SC2661C I/O Operations	6-6
Table 6.5-1	2661 Register Addressing	6-18
	SECTION 7 **Internal WISE Controller (IWISE)**	
Table 7.4-1	STATUS BYTE ASSIGNMENTS	7-9
Table 7.5-1	I/O Port Allocations	7-16
Table 7.5-2	CTC I/O Allocations	7-18
Table 7.5-3	IWRESTART RESET TABLE	7-20
Table 7.6-1	BYTE AQUSISTION AND CHECK TIMING	7-22
Table 7.6-2	Command Decoder Sub Command	7-24
Table 7.6-3	L81 Action Table	7-30
Table 7.7-1	50BUS Signals	7-33
	SECTION 8 **DIAGNOSTICS**	
Table 8.2-1	System Diagnostic	8-2

SECTION 6
INTERNAL PRINTER CONTROLLER (IPC)

SECTION 6
Internal Printer Controller (IPC)

6.1 INTRODUCTION

The Internal Printer Controller, commonaly called the IPC board is designed to control one printer through a modified RS232C serial interface. This board is comprised of four sections.

1. 50BUS Interface
2. Z80A Microprocessor
3. 64K Bytes of Memory
4. RS-232C Interface

The 50BUS interface provides the communication link between the RCU and the Z80A of the IPC. Through this bus, commands and data are transfered to the IPC. This interface is similiar to the 50BUS interface of the other boards in the OIS 40/50/60 system.

The onboard Z80A of the IPC receives instructions and data from the RCU and then goes about transfering the data to the printer through the RS-232C interface. One of the Z80A's responsablities is to monitor the performance of the printer while transferring data to the interface when requested. If a malfuntion should happen the Z80A is informed and it in turn notifies the system of the problem.

In order to increase the speed of the system and reduce the number of transfer operations, the IPC contains 64K of Dynamic RAM. This memory holds the operating codes for the onboard Z80A along with the data to be printed.

The heart of the IPC is the RS-232C interface. This interface consises of a SC2661C Enhanced Programmable Communication Interface (EPCI). The SC2661C is a universal sychronous/asychronous data communication controller chip that provides all the timing and control signals for transfers of data over the RS-232C connector. The onboard Z80A programs the SC2661C upon power up for asynchronous transmition with the proper start/stop and data bits, along with the selected Baud rate. The SC2661C then performs the parallel to serial conversion and inserts the proper number of start and stop bits. At the complection of each byte transfer the SC2661C will inform the onboard Z80A that the next byte can be loaded and status of the current operation can be read through the internal status register of the EPCI.

A block diagram of the IPC showing the four major components, there interconnections with the printer and the OIS 40/50/60 system, is shown in FIGURE 6.1-1. Each of the four section will be described in detail in the following pages.

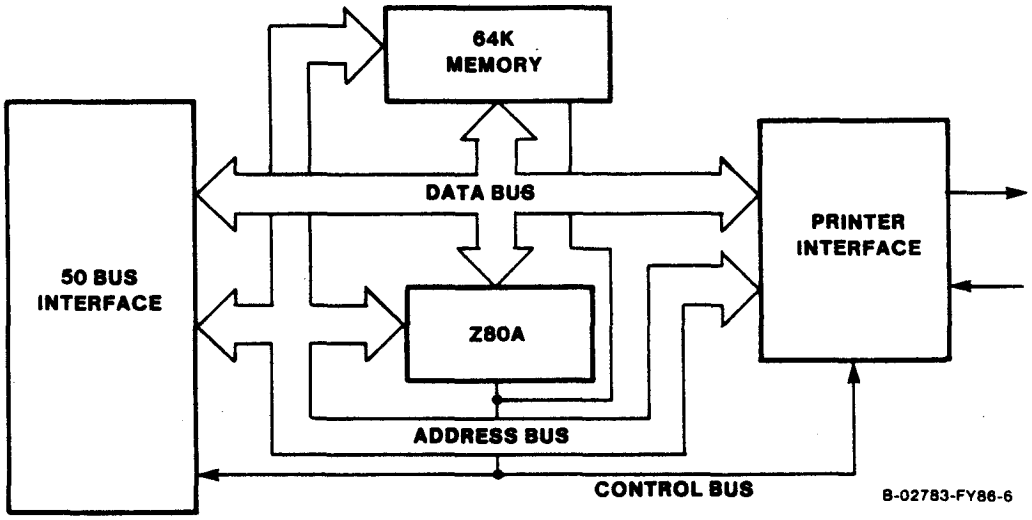


FIGURE 6.1-1
Internal Printer Controller (IPC) Block Diagram

6.2 50BUS Interface

The 50BUS provides a means of communication between the onboard Z80A CPU and the RCU board of the system. When the RCU desires to communicate with the IPC it first generates a BUS REQUEST. Upon receiving this request the onboard Z80A deactivates its busses after the complection of the current instruction, and allows the RCU access to the onboard memory and status registers. This bus request is generated by a combination of the &50BSLCT and &50BUSRQ signal. Both signals are feed through buffer L99 (1A10) to OR gate L74. When both signals are active L74 produces &IPCBSRQ to the Z80A. The Z80A will acknowledge this request by activating pin 23 &BUSAK, thus driving its Data, Address, and Control busses to a tristated condition. &BUSAK is also used to select the 50BUS Address Bus buffers, hence alowing the RCU access. The RCU will then place the appropiate address on the bus and perform the desired transfer operation. The information that the RCU transfers to or from the IPC could be in the form of either data or instructions for the IPC or it could be status information from the IPC. A detailed description of the 50BUS signals and there fuction is given in Sections 3.4.3 and 3.4.4 of Volume 1.

6.3 Z80A Microprocessor

The Z80A of the IPC board has two major tasks to perform. First, upon power up of the system it must receives program instructions from the RCU and then configure the SC2661C chip. Once this has been done the Z80A is responisable for transfering data to the SC2661C for transmition to the printer, and monitoring the status of the printer. The basic structure of the suport chips for the Z80A is the same as the rest of the circuit boards in the OIS 40/50/60 system. The Z80A is driven by a 4Mhz clock supplied by a crystal and associated circuits. If you drsire further information on the structure of the Z80A suport circuits refer to the appropiate section of either the RMU or the IWS controller.

6.3.1 NOP Generation

When the system is first powered on, each of the controller boards must be IPLed. Since each board receives different code and the system can not load all of the boards at the same time each board must be placed in a hold type state until it can be IPLed. This hold state is a function of the NOP generator. When the power is first applied to the system the RCU generates a master reset signal to all boards. This master reset signal known as &MRC is buffered through L99 (1A10) where it is places on the clock input of flipflop L37 (1A8). The "D" input of L37 has +5VDC applied to it via R1. On the trailing edge of &MRC L37 will set causing &NGON to be generated. &NGON will enable data buffer L52 (1B7). The "A" inputs of L52 are all tied to +5VDC, and once L52 is enabled the "Y" outputs will all go low. As you can see from the schematics these outputs are tied diresctly to the Z80A's data bus.

When the &MRC signal was sent to all the boards, the Z80A of the IPC board was placed in a reset state, which caused it to enter into a M1 cycle. During a M1 cycle the Z80A fetches an instruction from the data bus, this instruction is HØØ generated by the NOP generator. A NOP instruction causes the Z80A to perform continuous M1 cycles. When the Z80A is perfroming a NOP instruction the program counter incerments one step for each NOP perfromed. But the Data bus is held at all zero so the Z80A continues to perfrom NOPs. When &NGON was generated it was also sent to L106 the status register. Here it asserted the NOT RUNNING bit. The RCU then works its way through the boards on the system and determines which ones need IPLing. When the RCU finds a NOT RUNNING bit set, it will first generate a restart command to the device to clear the NOT RUNNING bit and disable the NOP generator. Then it will down load a jump to location zero instruction at location zero. This keeps the Z80A in a tight loop, thus keeping the PC register from advancing. Then when the RCU find time it will down load the operation instruction to the board, followed by another restart command. At this time the IPC's Z80A can then go about performing the instructions that it was given.

There are two other times that the NOP generator could be involked. The first is if a parity error is detected during a transfer to or from main memory. If this happens the signal PARERR (Parity Error) is generated and sent to L28 (1B10). It passes through L28 and on to the perset input of L37 the NOP flipflop. This PARERR signal is also sent to the status register to inform the RCU that a problem has occured. The second incident that will invoke the NOP generator is if the pwoer on the printer is turned off and the CTS (Clear to Send) signal is lost. If this happens the &POWERON signal will deactivate, this high level signal will then be places on pin 5 of L28 where it will be inverted and sent to the preset input of L37 the NOP generator flipflop. This &POWERON signal is also sent to the status register for the RCU to read. One final note on the NOP generator operation. If the &POWERON signal is not being generated, either by the printer being turned off or the cable being disconnected, the IPC board will not IPL. This problem can be resolved by placing J4 between pins 2 & 3.

NOTE

If you move J4 from pins 1 & 2 so that you can test the board without a printer connected. Then remember to move it back when you are finished with the test.

6.3.2 I/O Operations

The Z80A of the IPC board communicates to the other devices on the board through memory mapped I/O locations. These operation can be divided into two areas;

1. Device Status and Diagnostic Operations
2. SC2661C Operations

Tables 6.3-1 and 6.3-2 provide a brief explanation of each I/O operation.

TYPE	LOCATION	FUNCTION
IN	07	Device Switch 2
IN	08	Device Switch 1
IN	09	Diagnostic Mode Indicator
IN	0A	Resets the SC2661C
IN	0B	Toggles Diagnostic Mode *
IN	0C	Clears Parity Error Flipflop
IN	0D	Toggles Parity Flipflop *

Table 6.3-1
Device and Diagnostic I/O Operations

* Each read from these locations will cause the state of the flipflop to change states. Further explaination of their operation can be found later in this Section.

The folowing table list the operation associated with the I/O locations that corrisponds to the SC2661C (FPCI) chip. The decoding of these locations are within the SC2661C once the chip has been selected. The SC2661C is selected any time that bit 4 of the address bus is high during an I/O operation.

TYPE	LOCATION	FUNCTION
IN	10	Reads Receive Data Holding Register
OUT	11	Writes to Transmit Holding Register
IN	12	Reads the Status Register
OUT	13	Writes to SYN1/SYN2/DLE Registers *
IN	14	Reads Mode Registers MR1/MR2 *
OUT	15	Writes to Mode Registers MR1/MR2 *
IN	16	Reads Command Register
OUT	17	Writes to Command Register

Table 6.3-2
SC2661C I/O Operations

* Each of these I/O locations have multiple registers assigned to them. Internal circuits in the SC2661C keep track of wich register is to be accessed each time a read or write operation occurs for that location. (i.e. If you perform an OUT 13, you will write to the SYN1 register. Then if you perform another OUT 13, you will write to the SYN2 register. When the maximim number of registers for a given location has been reached the internal circuits will recycle to point to the first register.) The register pointers are reset to their start position when either the reset input of the SC2661C is activated or by reading the Command register. A detailed explaination of each of the registers within the SC2661C can be found in Section 6.5.2.

All I/O operation for the IPC board are selected through three LS138 decoders, L30, 31, and 32 located on sheet 1 of the schematics. Each of these decoders need three enabling signals to alow them to operate, G2A, G2B and G1. The G1 and G2B inputs of each are driven from &BM1 and either &BRD or &BIORQ signals respectivaly. The G1 input is an active high input and will be enabled whenever the Z80A is NOT in a M1 machine cycle. Remembering that the M1 machine cycle is an instruction fetch activity. The G2B input of L31 and L30 are driven by the active low signal of &BRD (Buffered Read). When the Z80A is performing a read operation this signal will be driven low, thus activating the G2B input of these decoders. The G2B input of L32 is driven from the &BIORQ (Buffered I/O request) signal of the Z80A. This signal is activated whenever the Z80A performs an I/O instruction. Before I go any futher let's look at FIGURE 6.3-1 to see how these three decoders are wired.

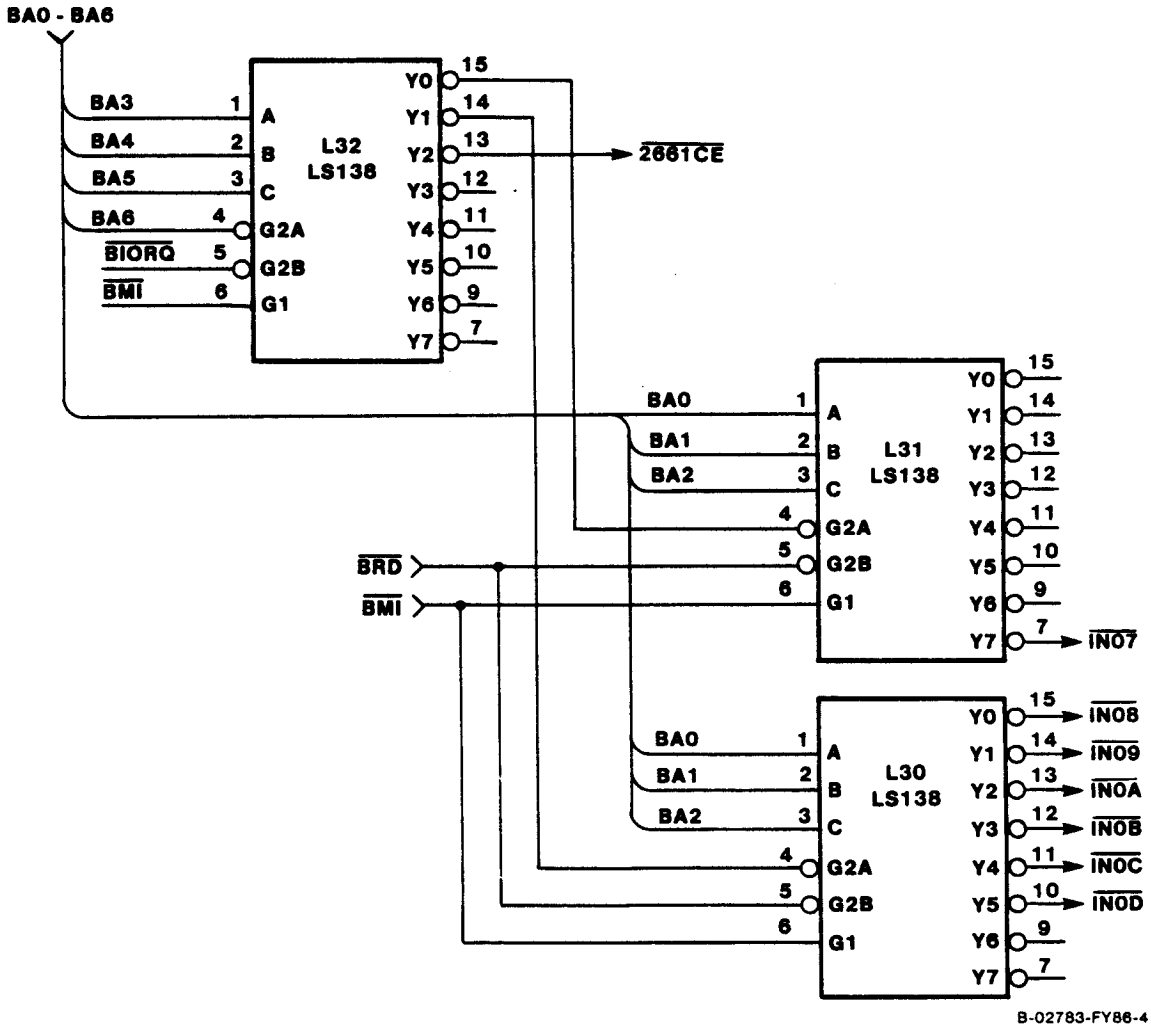


FIGURE 6.3-1
I/O Decoders

The third enable input of both L30 and L31 are driven by one of the outputs of L32. This tells us that L32 must be selected before either L30 or L31 can be enabled. L32 receives it's G2A enable from the BA6 bit of the Address Bus. Whenever this bit is low and the other two enable inputs are activated L32 will decode address bits BA3 - BA5. Only the first three outputs of L32 are used by the IPC board. If BA3 through BA5 are all low then the Y0 output will be driven low. This low out will then enable L31, allowing it to decode BA0 through BA2 of the Address Bus.

The Y1 output of L32, when activated will enable L30, thus allowing it to decode the lower three bits of the Address Bus. Finally when the Y2 output of L32 is active we will select the SC2661. The SC2661 will then decode the lower three bits of the Address Bus.

Let's now take a look at some of the I/O operation and see what functions they perform. The IN07, 08 and 09 operations allow either the system or the onboard Z80A to read the setting of the device switches and the condiction of the Diagnostic Mode Indicator. These three operations are quite simple in that they enable the appropiate buffer, thus placing the contence of the switches or registers on the Data Bus.

The next I/O operation is the IN0A. When the IN0A instruction is performed by the Z80A, L30's Y2 output will be driven low. This high to low transion is inverted and presented to the clock input of flipflop L59 (2B11). The flipflop will set causing the &Q output to go low, generating &WAIT and 2661RST via L58. At the same time the clear input of L57 an LS163 counter is removed. L57 then starts counting. When the QD output goes high on the 8^{th} count flipflop L59 will reset via L67 and L90. This delay is needed to allow the SC2661 time to reset it's internal registers. The &WAIT signal that was generated during this operation was presented to the Z80A placing WAIT states within the instruction cycle.

The IN0B operation allows the system to switch between Normal mode and Diagnostic mode. Each time the IN0B instruction is performed flipflop L35 (1B1) is toggled. When the Normal mode of operation is selected read and write operation to memory occur with even parity. But when the Diagnostic mode of operation is selected read and write operation to memory occur with odd parity. The reason for having odd parity during these transfers is so that the software can check the memory circuits for parity errors.

When a parity error occurs flipflop L59 (3B2) sets, and causes one of two parity error signals to be activated depending on the selected mode of operation. To return this flipflop to the reset condition an IN0C instruction must be preformed. Decoder L30 (1C2) generates IN0C which is gated through AND gate L90 (3B3) to clear flipflop L59.

When the INOD instruction is performed flipflop L89 (2C8) will toggle between Normal parity error responce, and MNI responce. During Normal parity responce, if a parity error occurs the NOP generator is turned on. This causes the Z80A to run in a very tight loop until the problem is recified. For futher information on the NOP generater refer to Section 6.3.1. If the MNI type of resopnce is selected, and a parity error occurs a None Maskable Interrupt will be generated. This type of responce can be used for diagnostic purposes by informing the software that a problem has occured without locking up the micro.

The rest of the I/O operation are all decoded within the SC2661 chip. These operations will be used extencivelly by the Z80A to communicate with the SC2661. The key to these operations is the selection of the SC2661. This is accomplished when BA4 is high and BA3, 5 and 6 are low, and of course the Z80A is performing an I/O operation. When this happens L32 (1F3) generates SC2661CE (2661 chip select). Then the SC2661 decodes the lower three bits of the Address Bus to determine the operation. As I stated before, some of these operations when performed, will selecte different registers within the SC2661 for the same I/O command. It is important to keep track of the number of times a given I/O operation is performed in order to know which register is being accessed. Normally this is done by the software but if during testing the operator tries to read or write to these locations he or she must be aware of the fact that each access will be to a different register. A complete breakdown of the SC2661 operations and commands are descussed in the Section 6.5.

6.4 IPC Memory

The memory of the IPC board consists of nine HM4864-3 64Kx1 memory chips. Eight of these are used for main memory and one for parity. Since these chips are dynamitic they require a 256-row refresh every 2 to 4 Msec. The Z80A provides a 7 bit refresh address at the end of each M1 machine cycle, yielding a total of 128 row addressess. To provide for 256-row addressing a synthetic refresh address bit is multiplexed into the A7R bit of the Z80A Address Bus. The circuits involved in this operation are similar to the ones in the RMU boards.

6.4.1 REFRESH

In a 256-row refresh address operation, the Z80A issues A6 low for the first 128-row refresh and high for the second 128-row refresh. By using this A6 bit as a control input to a "D" type flipflop we can generate a synthetic 256-row refresh address. Referring to FIGURE 6.4-1 and FIGURE 6.4-2 we see the four flipflops that provide this operation. Flipflop L60-1 and L60-2 are used to toggle the synthetic refresh bit while flipflop L51 switches between the synthetic bit and the normal A7 bit, and L43-2 signals the start of refresh.

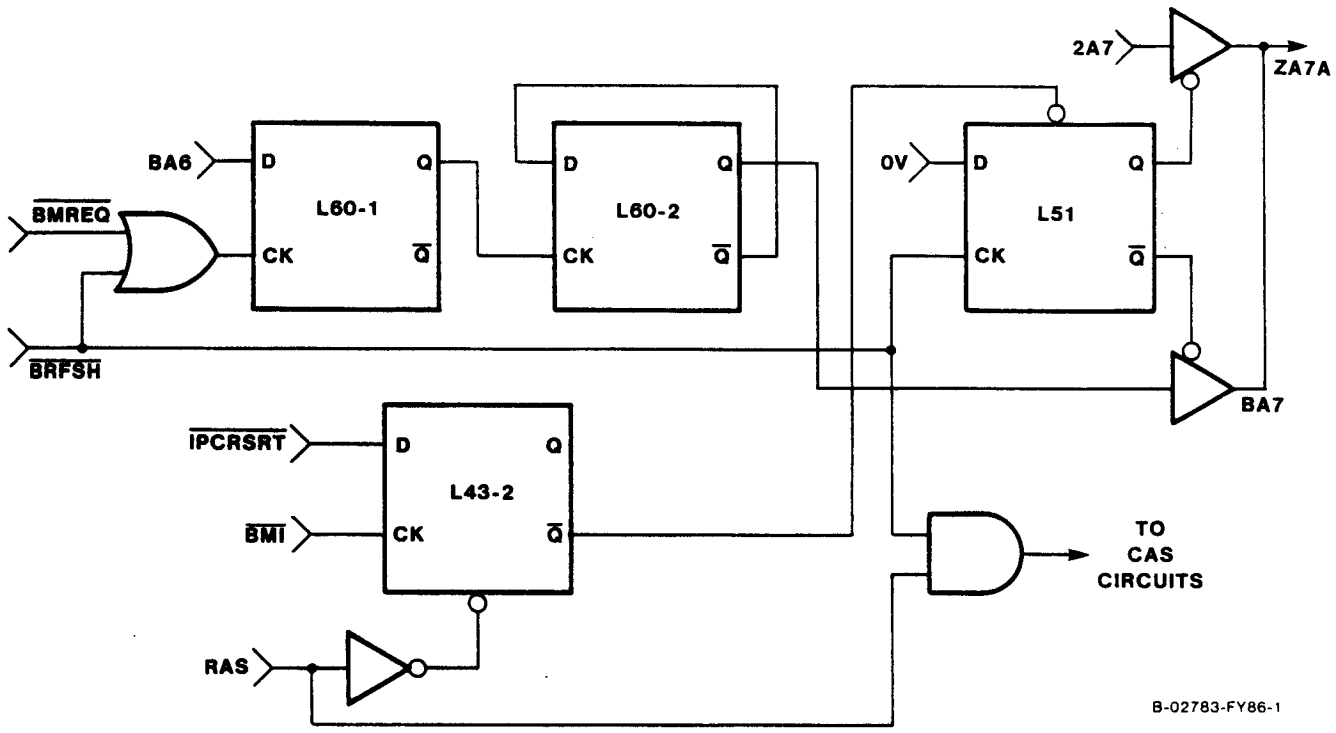


FIGURE 6.4-1
REFRESH LOGIC DIAGRAM

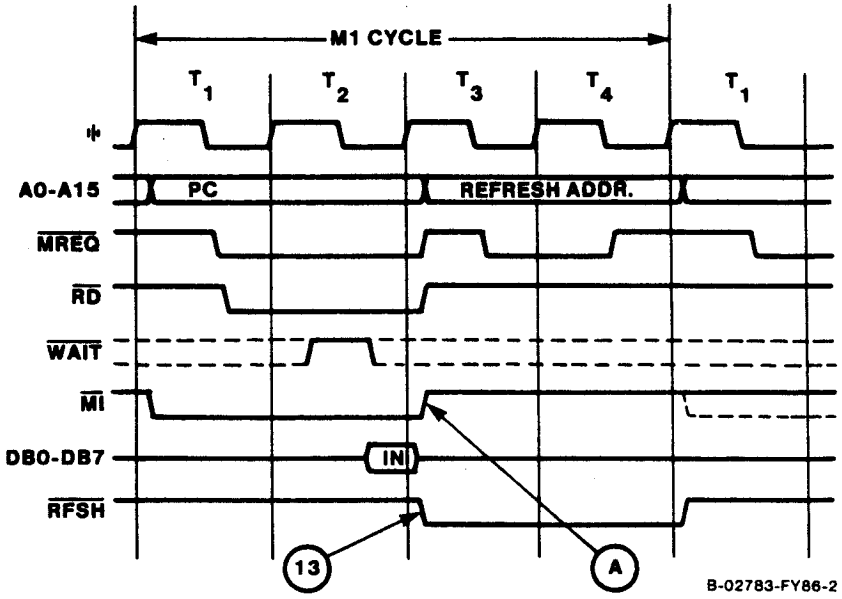


FIGURE 6.4-2
M1 Timing Diagram

Flipflop L51 is normally in a reset state, this enables driver L68 to pass the ZA7 bit from the Z80A to the Address Buss buffer L62. Halfway through the M1 machine cycle &BM1 will deactivate (Refer to FIGURE 6.4-2 Point A) Causing flipflop L43 to set, thus generating Early Refresh (ERFSH) from the "&Q" output. Early Refresh is placed on the preset input of L51, causing it to set and switching the ZA7A output from ZA7 to BA7. The "Q" output of L43 is passed to the "D" input of L94 where on the next 1/2 phi clock pulse L94 will set starting the RAS cycle.

At the same time that &BM1 went high, &BMREQ went high and &BRFSH went low, signaling the start of the refresh cycle (Refer to FIGURE 6.4-2 Point B). In the middle of T3 time &BMREQ became active again, setting up L60-1. During this time (T3 & T4) the Z80A places the refresh address on the Address Bus. Then sometime during T4 state of M1, &BMREQ deactivates again, clocking flipflop L60-1. The state of the BA6 bit of the Address Bus, which is feeding the "D" input of L60-1, will be transfered through to L60-2. Each time the Z80A presents a refresh address on the bus, it (the Z80A) will toggle the state of BA6. When BA6 is in the high state L60-2 will clock and change the state of the synthetic BA7 bit. Using this techeque, a minium of two M1 machine cycle are required to refresh the full 64K bytes of memory. At the end of the refresh cycle &BRFSH will deactivate, this low to high action clocks L51-1 into a reset state, switching the synthetic BA7 bit out of the circuit.

Remember that during a refresh cycle we only need a RAS cycle. While we are in a refresh cycle, &BRFSH signal is low, hence, AND gate L69 is disqualified, blocking the RAS signal from proceeding.

6.4.2 RAS/CAS

There are three different ways that the RAS/CAS cycle can be started. 1.) Normal read or write request, 2.) Instruction fetch, 3.) Refresh cycle. Of these three the Refresh action was decussed in the last section so we will consintrate on the first two.

During a normal read or write operation, this exculdes instruction fetch even though it is in reality a read operation, the Z80A will generate &BMREQ (Buffered Memory Request). This signal will be inverted by L41 (3B11) and passed through gates L50 to generate EPDB (Enable Parity and Data Bus) refer to FIGURE 6.4-3. This signal enables the parity circuits and selects the Memory out buffer. Also the transision of this signal, clocks flipflop L51-2 into a set state causing the "Q" output to go high, hence driving the "D" input of flipflop L94-1 high through OR gate L76. On the next low to high shift of 1/2 phi Z clock L94 will set causing &RAS to generated, signaling

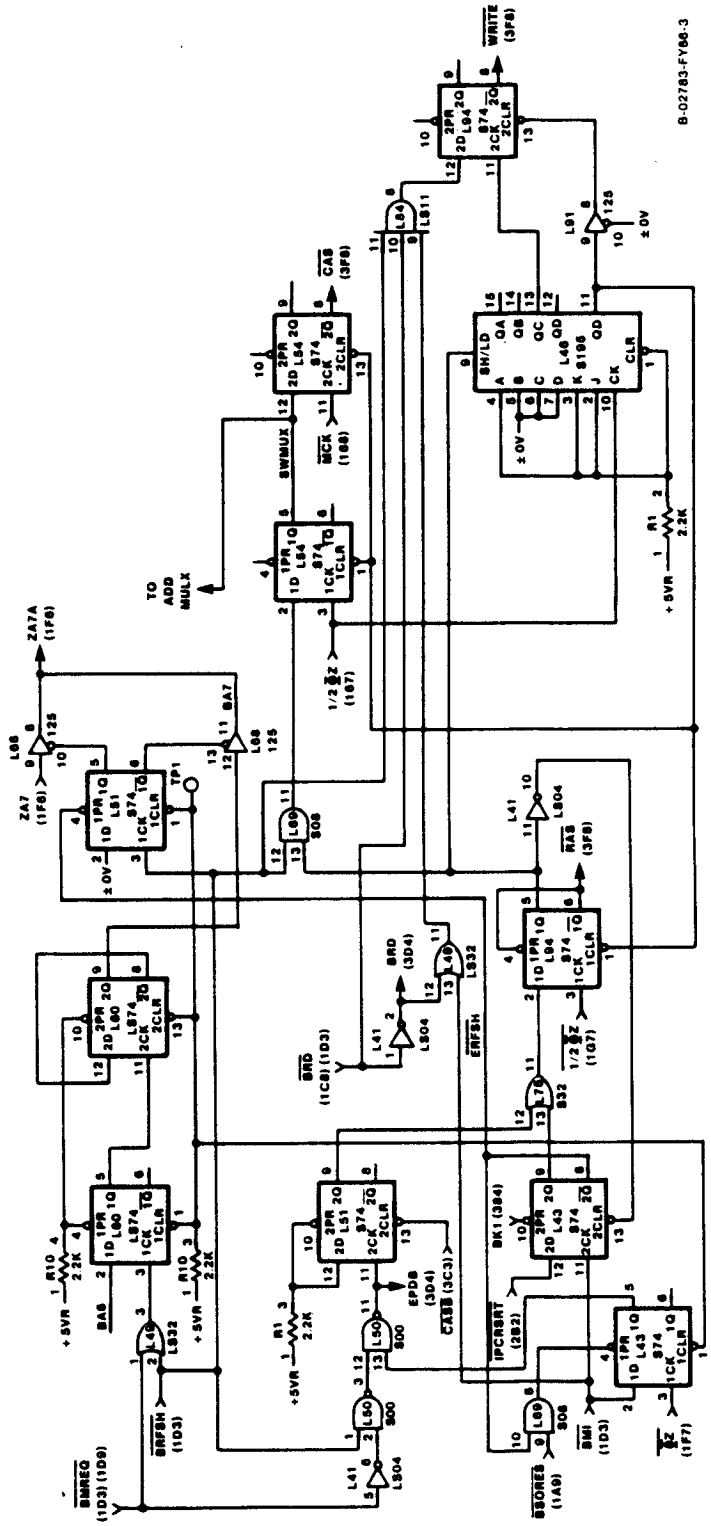


FIGURE 6.4-3
RAS/CAS LOGIC DIAGRAM

the start of the RAS/CAS cycle. The remainder of the RAS/CAS cycle will be descussed later. At the complection of the RAS/CAS cycle L51-2 will be placed in the reset condition by &CASB on its reset input. This will treminate the cycle on the next &1/2 phi z clock.

The RAS/CAS cycle is also started when the Z80A fetches an instruction. When the Z80A begins an instruction fetch cycle it generates &BM1 (Machine 1). This signal is directed to the "D" input of flipflop L43-1. On the next low to high transision of phi Z clock L43 will reset causing the "Q" output to go low. This low is inverted through NAND gate L50 to generate EPDB which then starts the RAS/CAS cycle, just as it did in the last paragraph. Flipflop L43-1 will go back to the set state at T3 time of the M1 cycle, when &BM1 deactivates. The Z80A also generates &BMREQ during the instruction fetch cycle, but it comes at a much later time in the cycle. If the circuits relied on &BMREQ to activate the RAS/CAS cycle during an instruction fetch the address that the PC register of the Z80A placed on the Address Bus would not be stable long enough to generate a reliable address.

To complete the RAS/CAS cycle two more operation must be perfromed. First the address multiplexors must be switched and second the CAS signal must be asserted. Once the RAS flipflop L94 has been set the "Q" output is sent to two locations. The first is the Shift/Load input of shift register L46. This places the register in a shift mode of operation. This register has a two fold operation. One, it determines the complection of the RAS/CAS cycle, and two, it signals the proper time for the write enable signal during a write operation. The second place that the "Q" output of L94-1 is sent is the "D" input of flipflop L54-1, via AND gate L69. Remember, if we are in a refresh cycle AND gate L69 will be disqualified, thus blocking this path. Once the high is placed on the "D" input of L54 and the next low to high transision of 1/2 phi z clock happens L54-1 will set. This generates SWMUX (Swicth Multiplexor) which causes the two address selectors to select the high order bits of the Address Bus. Going back the "Q" output of L54-1 where SWMUX was generated we see that it is also placed on the "D" input of L54-2. On the next low to high transision of &MCK L54-2 will set, generating &CAS from the &"Q" output. At this time the address operation is complete and we are able to perfrom either a read or write.

The only real difference between a read operation and a write operation is that during a write cycle, the Z80A will wait until T4 time before it transfers the data from it's register to the memory. The write enable signal should not be activated until after both RAS and CAS have happened. The timing of this signal is controlled by counter L46. If you recall back when RAS was generated we placed L46 into a shift mode of operation. Then while we were setting up SWMUX and CAS L46 was shifting the single one it had placed in it. When this one