

- High-Performance Static CMOS Technology
- Includes the T320C2xx Core CPU
  - Object Compatible With the TMS320C2xx
  - Source Code Compatible With TMS320C25
  - Upwardly Compatible With TMS320C5x
  - 50-ns Instruction Cycle Time
- Industrial Temperature Standard, Automotive Temperature Available
- Memory
  - 544 Words x 16 Bits of On-Chip Data/Program Dual-Access RAM
  - 8K Words x 16 Bits of Flash EEPROM
  - 224K Words x 16 Bits of Total Memory Address Reach (64K Data)
- External Memory Interface
- Event-Manager Module
  - Eight Compare/Pulse-Width Modulation (PWM) Channels
  - Two 16-Bit General-Purpose Timers With Six Modes, Including Continuous Up and Down Counting
  - Three 16-Bit Full Compare Units With Deadband
  - Three Capture Units (Two With Quadrature Encoder-Pulse Interface Capability)
- Single 10-Bit Analog-to-Digital Converter (ADC) Module With 8 Multiplexed Input Channels
- Controller Area Network (CAN) Module
- 32 Individually Programmable, Multiplexed I/O Pins
- Six Dedicated I/O
- Phase-Locked-Loop (PLL)-Based Clock Module
- Watchdog Timer Module
- Serial Communications Interface (SCI) Module
- 16-Bit Serial Peripheral Interface (SPI) Module
- Six External Interrupts (Power Drive Protect, Reset, NMI, and Two Maskable Interrupts)
- Four Power-Down Modes for Low-Power Operation
- Scan-Based Emulation
- Development Tools Available:
  - Texas Instruments (TI™) ANSI C Compiler, Assembler/Linker, and C-Source Debugger
  - Full Range of Emulation Products: Self-Emulation (XDS510™)
  - Third-Party Digital Motor Control and Fuzzy-Logic Development Support
- 144-Pin Plastic Quad Flatpack

## description

The TMS320F243 device is a member of the '24x family of DSP controllers based on the TMS320C2xx generation of 16-bit fixed-point digital signal processors (DSPs). Table 1 provides a comparison of the features of each device. This new family is optimized for digital motor/motion control applications. The DSP controllers combine the enhanced TMS320 architectural design of the 'C2xx core CPU for low-cost, high-performance processing capabilities and several advanced peripherals optimized for motor/motion control applications. These peripherals include the event manager module, which provides general-purpose timers and PWM registers to generate PWM outputs, and a single, 10-bit analog-to-digital converter (ADC), which can perform conversion within 1  $\mu$ s. Refer to the functional block diagram.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

TI and XDS510 are trademarks of Texas Instruments Incorporated.

PRODUCT PREVIEW information concerns products in the formative or design phase of development. Characteristic data and other specifications are design goals. Texas Instruments reserves the right to change or discontinue these products without notice.



POST OFFICE BOX 1443 • HOUSTON, TEXAS 77251-1443

Copyright © 1997, Texas Instruments Incorporated

PRODUCT PREVIEW

TMS320F243  
DSP CONTROLLER

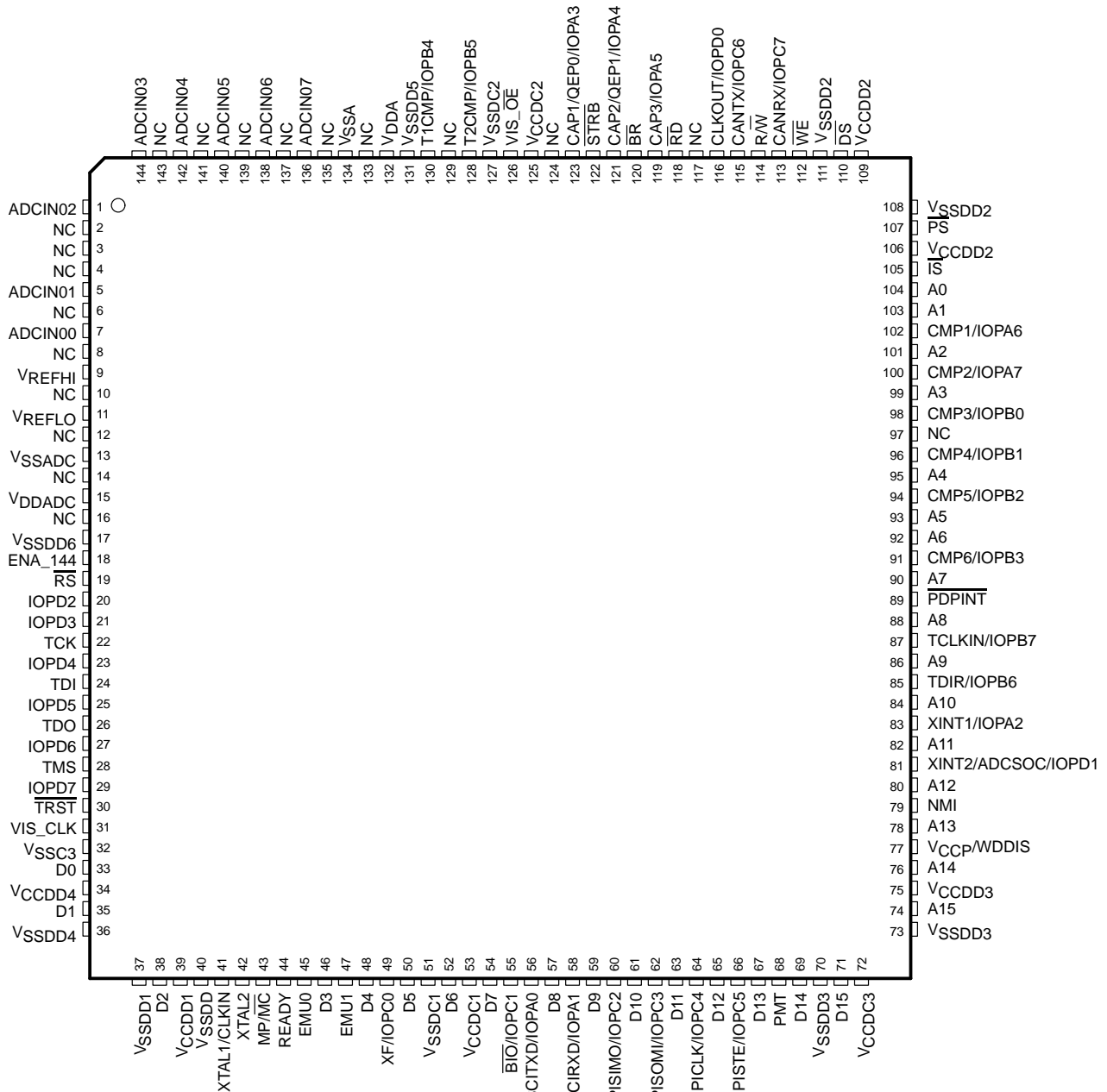
SPRS064 – DECEMBER 1997

description (continued)

Table 1. Characteristics of the TMS320F243 DSP Controller

TMS320x24x DEVICES	ON-CHIP MEMORY (WORDS)				POWER SUPPLY (V)	CYCLE TIME (ns)	PACKAGE TYPE PIN COUNT
	RAM		ROM	FLASH EEPROM			
	DATA	DATA/PROG	PROG	PROG			
TMS320F243	288	256	–	8K	5	50	PGE 144-PQFP

**TMS320F243**  
**PGE PACKAGE†**  
**(TOP VIEW)**



† NC = No connection

**PRODUCT PREVIEW**

# TMS320F243 DSP CONTROLLER

SPRS064 – DECEMBER 1997

## Terminal Functions

TERMINAL NAME	NO.	TYPE†	DESCRIPTION
<b>ANALOG-TO-DIGITAL CONVERTER (ADC) INPUTS</b>			
ADCIN00	7	I	Analog inputs to the ADC
ADCIN01	5		
ADCIN02	1		
ADCIN03	144		
ADCIN04	142		
ADCIN05	140		
ADCIN06	138		
ADCIN07	136		
VDDADC	15	I	Digital supply voltage for ADC (must be separate supply)
VDDA	132	I	Analog supply voltage for ADC
VSSADC	13	I	Digital ground reference for ADC (must be separate supply)
VSSA	134	I	Analog ground reference for ADC
VREFHI	9	I	ADC analog high voltage reference input
VREFLO	11	I	ADC analog low voltage reference input
<b>EVENT MANAGER</b>			
T1CMP/IOPB4	130	I/O	Timer 1 output or general-purpose bidirectional I/O (GPIO)
T2CMP/IOPB5	128	I/O	Timer 2 output or GPIO
TDIR/IOPB6	85	I/O	GP timer direction or GPIO
TCLKIN/IOPB7	87	I/O	GP timer clock input or GPIO
CAP1/QEP0/IOPA3	123	I/O	Capture input #1 or GPIO
CAP2/QEP1/IOPA4	121	I/O	Capture input #2 or GPIO
CAP3/IOPA5	119	I/O	Capture input #3 or GPIO
CMP1/IOPA6	102	I/O	Compare/PWM output pin #1 or GPIO
CMP2/IOPA7	100	I/O	Compare/PWM output pin #2 or GPIO
CMP3/IOPB0	98	I/O	Compare/PWM output pin #3 or GPIO
CMP4/IOPB1	96	I/O	Compare/PWM output pin #4 or GPIO
CMP5/IOPB2	94	I/O	Compare/PWM output pin #5 or GPIO
CMP6/IOPB3	91	I/O	Compare/PWM output pin #6 or GPIO
PDPINT	89	I	Power device protection interrupt input
<b>SERIAL COMMUNICATIONS INTERFACE (SCI) AND BIT I/O PINS</b>			
SCITXD/IOPA0	56	I/O	SCI asynchronous serial port transmit data, or GPIO
SCIRXD/IOPA1	58	I/O	SCI asynchronous serial port receive data, or GPIO
<b>SERIAL PERIPHERAL INTERFACE (SPI) AND BIT I/O PINS</b>			
SPISIMO/IOPC2	60	I/O	SPI slave in, master out or GPIO
SPISOMI/IOPC3	62	I/O	SPI slave out, master in or GPIO
SPICLK/IOPC4	64	I/O	SPI clock or GPIO
SPISTE/IOPC5	66	I/O	SPI slave transmit enable (optional) or GPIO

† I = input, O = output, Z = high impedance



POST OFFICE BOX 1443 • HOUSTON, TEXAS 77251-1443

Terminal Functions (Continued)

TERMINAL NAME	NO.	TYPE†	DESCRIPTION
<b>CONTROLLER AREA NETWORK (CAN)</b>			
CANTX/IOPC6	115	I/O	CAN transmit data or GPIO
CANRX/IOPC7	113	I/O	CAN receive data or GPIO
<b>INITIALIZATION, INTERRUPT, RESET, AND MISCELLANEOUS SIGNALS</b>			
XF/IOPC0	49	I/O	DSP XF output or GPIO
BIO/IOPC1	55	I/O	DSP BIO input or GPIO
CLKOUT/IOPD0	116	I/O	Clock out or GPIO
NMI	79	I	Nonmaskable interrupt
XINT1/IOPA2	83	I/O	External interrupt 1 or GPIO
XINT2/ADCSOC/ IOPD1	81	I/O	External interrupt 2 and ADC start of conversion or GPIO
$\overline{\text{RS}}$	19	I/O	Device reset
TCK	22	I	IEEE Standard 1149.1 (JTAG) test clock with internal pullup
READY	44	I	READY is pulled low to add wait states Data-ready input. READY indicates that an external device is prepared for a bus transaction to be completed. If the device is not ready (READY low), the processor waits one cycle and checks READY again. Note that the processor performs READY-detection if at least two software wait states are programmed. The READY signal is not sampled until the completion of the software wait states.
MP/ $\overline{\text{MC}}$	43	I	Microprocessor/Microcomputer mode select
$\overline{\text{IS}}$	105	O	I/O space strobe
$\overline{\text{DS}}$	110	O	Data space strobe
$\overline{\text{PS}}$	107	O	Program space strobe
$\overline{\text{WE}}$	112	O	Write enable strobe
$\overline{\text{RD}}$	118	O	Read enable strobe
R/ $\overline{\text{W}}$	114	O	Read/write qualifier
$\overline{\text{STRB}}$	122	O	External memory access strobe
$\overline{\text{BR}}$	120	O	Global memory strobe
VIS_CLK	31	O	Visibility clock
ENA_144	18	I	To be supplied (TBS)
VIS_ $\overline{\text{OE}}$	126	O	TBS
V <sub>CCP</sub> /WDDIS	77	I	Flash programming voltage pin and watchdog disable. This is the 5-V supply which is used by the charge pumps to generate the internal 12-V V <sub>PP</sub> . Flash cannot be programmed or erased if this is held at 0 V. This pin also doubles as a hardware watchdog disable, when V <sub>CCP</sub> /WDDIS = +5 V and bit 6 in WDCR is set to 1.
<b>DEDICATED I/O SIGNALS</b>			
IOPD2	20	I/O	Dedicated GPIO – Port D bit 2
IOPD3	21	I/O	Dedicated GPIO – Port D bit 3
IOPD4	23	I/O	Dedicated GPIO – Port D bit 4
IOPD5	25	I/O	Dedicated GPIO – Port D bit 5
IOPD6	27	I/O	Dedicated GPIO – Port D bit 6
IOPD7	29	I/O	Dedicated GPIO – Port D bit 7

† I = input, O = output, Z = high impedance

PRODUCT PREVIEW



# TMS320F243 DSP CONTROLLER

SPRS064 – DECEMBER 1997

## Terminal Functions (Continued)

TERMINAL NAME	NO.	TYPE†	DESCRIPTION
DATA AND ADDRESS BUS SIGNALS			
D0	33	I/O	16-bit Data Bus
D1	35	I/O	
D2	38	I/O	
D3	46	I/O	
D4	48	I/O	
D5	50	I/O	
D6	52	I/O	
D7	54	I/O	
D8	57	I/O	
D9	59	I/O	
D10	61	I/O	
D11	63	I/O	
D12	65	I/O	
D13	67	I/O	
D14	69	I/O	
D15	71	I/O	
A0	104	O	16-bit Address Bus
A1	103		
A2	101		
A3	99		
A4	95		
A5	93		
A6	92		
A7	90		
A8	88		
A9	86		
A10	84		
A11	82		
A12	80		
A13	78		
A14	76		
A15	74		
CLOCK SIGNALS			
XTAL2	42	O	PLL oscillator output pin. XTAL2 is tied to one side of a reference crystal when the device is in PLL mode (CLKMD[1:0] = 1x, CKCR0.7–6). This pin goes in the high-impedance state when EMU1 is active low.
XTAL1/CLKIN	41	I	PLL oscillator input pin. XTAL1/CLKIN is tied to one side of a reference crystal in PLL mode (CLKMD[1:0] = 1x, CKCR0.7–6).

† I = input, O = output, Z = high impedance



POST OFFICE BOX 1443 • HOUSTON, TEXAS 77251-1443

**Terminal Functions (Continued)**

TERMINAL NAME	NO.	TYPE†	DESCRIPTION
<b>IEEE 1149.1 TEST SIGNALS</b>			
TDI	24	I	IEEE standard 1149.1 (JTAG) test data input (TDI) with internal pullup. TDI is clocked into the selected register (instruction or data) on a rising edge of TCK.
TDO	26	O	IEEE standard 1149.1 (JTAG) test data output (TDO). The contents of the selected register (instruction or data) is shifted out of TDO on the falling edge of TCK.
TMS	28	I	IEEE standard 1149.1 (JTAG) test-mode select with internal pullup. This serial control input is clocked into the TAP controller on the rising edge of TCK.
$\overline{\text{TRST}}$	30	I	IEEE standard test reset with internal pulldown. $\overline{\text{TRST}}$ , when active low, gives the scan system control of the operations of the device. If this signal is not connected or driven low, the device operates in its functional mode, and the test reset signals are ignored.
EMU0	45	I/O	Emulator I/O pin 0 with internal pullup. When $\overline{\text{TRST}}$ is driven high, this pin is used as an interrupt to or from the emulator system and is defined as input/output through the JTAG scan.
EMU1	47	I/O	Emulator I/O pin 1 with internal pullup. When $\overline{\text{TRST}}$ is driven high, this pin is used as an interrupt to or from the emulator system and is defined as input/output through JTAG scan.
PMT	68	I	Enables parallel module test (PMT) of Flash memory array
<b>SUPPLY SIGNALS</b>			
VSSDD1	37	I	Digital logic and buffer ground reference
VSSDD2	108, 111	I	Digital logic and buffer ground reference
VSSDD3	70, 73	I	Digital logic and buffer ground reference
VSSDD4	36	I	Digital logic and buffer ground reference
VSSDC1	51	I	Digital core logic ground reference
VSSDC2	127	I	Digital core logic ground reference
VSSC3	32	I	Digital logic ground reference
VCCDD1	39	I	Digital I/O logic and buffer supply voltage
VCCDD2	106, 109	I	Digital I/O logic and buffer supply voltage
VCCDD3	75, 72	I	Digital I/O logic and buffer supply voltage
VCCDD4	34	I	Digital I/O logic and buffer supply voltage
VCCDC1	53	I	Digital core logic supply voltage
VCCDC2	125	I	Digital core logic supply voltage
VSSDD5	131	I	Digital logic and buffer ground reference
VSSDD6	17	I	Digital logic and buffer ground reference
VSSDD	40	I	Digital logic and buffer ground reference
<b>NO CONNECTS</b>			
NC	2	N/A	No connect
NC	3		
NC	4		
NC	6		
NC	8		
NC	10		
NC	12		
NC	14		
NC	16		
NC	97		

† I = input, O = output, Z = high impedance

**PRODUCT PREVIEW**



TMS320F243  
DSP CONTROLLER

SPRS064 – DECEMBER 1997

Terminal Functions (Continued)

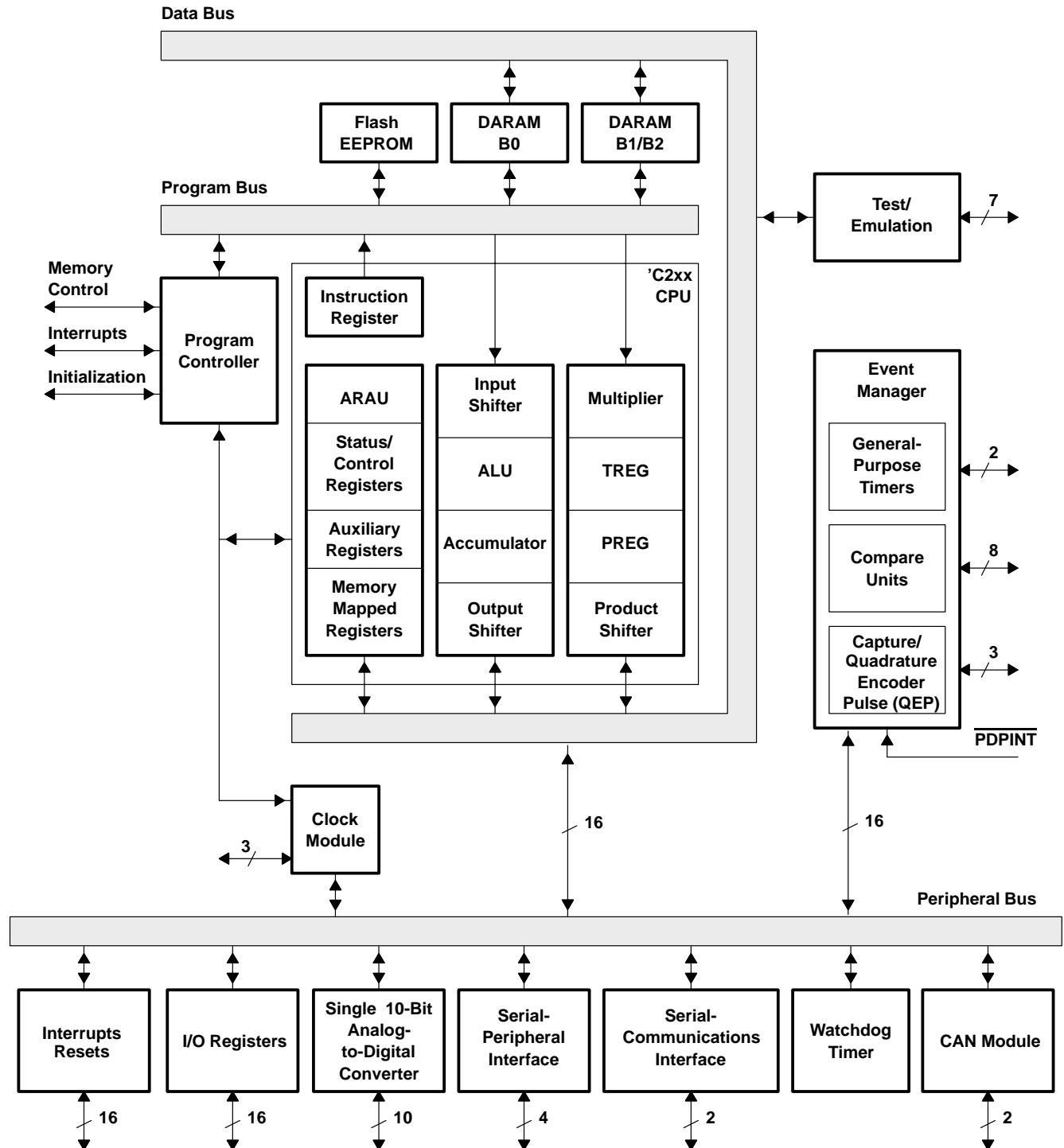
TERMINAL NAME	NO.	TYPE†	DESCRIPTION
NO CONNECTS (CONTINUED)			
NC	117	N/A	No connect
NC	124		
NC	129		
NC	133		
NC	135		
NC	137		
NC	139		
NC	141		
NC	143		

† I = input, O = output, Z = high impedance





# functional block diagram



PRODUCT PREVIEW

## architectural overview

The functional block diagram provides a high level description of each component in the 'x243 DSP controller device. The TMS320x24x devices are composed of three main functional units: a 'C2xx DSP core, internal memory, and peripherals. In addition to these three functional units, there are several system-level features of the 'x243 that are distributed. These system features include the memory map, device reset, interrupts, digital input/output (I/O), clock generation, and low-power operation.

## memory map

The TMS320F243 implement three separate address spaces for program memory, data memory, and I/O space. On the 'x243, the first 96 (0–5Fh) data memory locations are either allocated for memory-mapped registers or reserved. This memory-mapped register space contains various control and status registers, including those for the CPU.

All the on-chip peripherals of the 'x243 devices are mapped into data memory space. Access to these registers is made by the CPU instructions addressing their data memory locations. Figure 1 shows the 'x243 memory map.

memory map (continued)

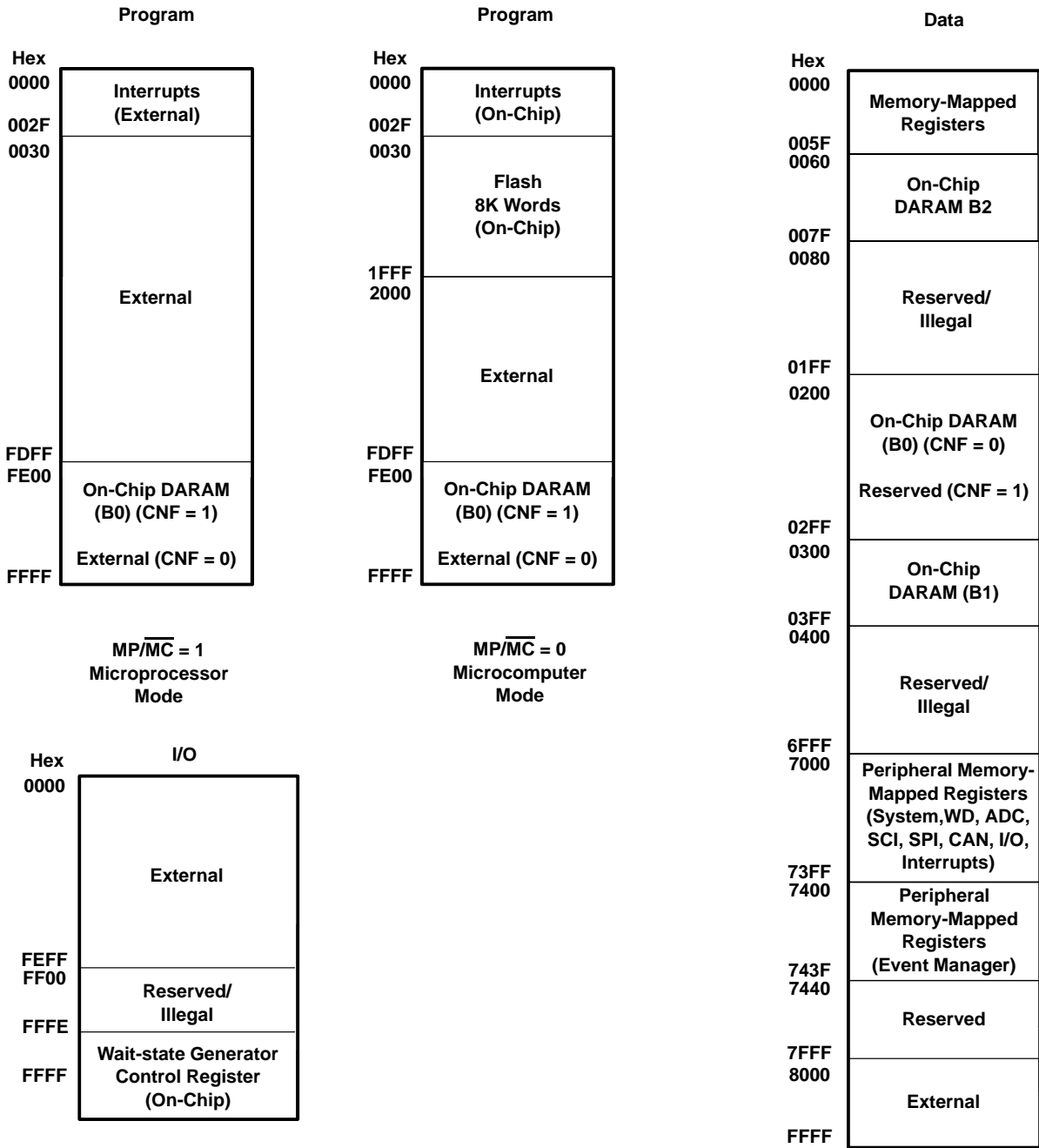


Figure 1. TMS320F243 Memory Map

# TMS320F243 DSP CONTROLLER

SPRS064 – DECEMBER 1997

## peripheral memory map

The TMS320x24x system and peripheral control register frame contains all the data, status, and control bits to operate the system and peripheral modules on the device (excluding the event manager).

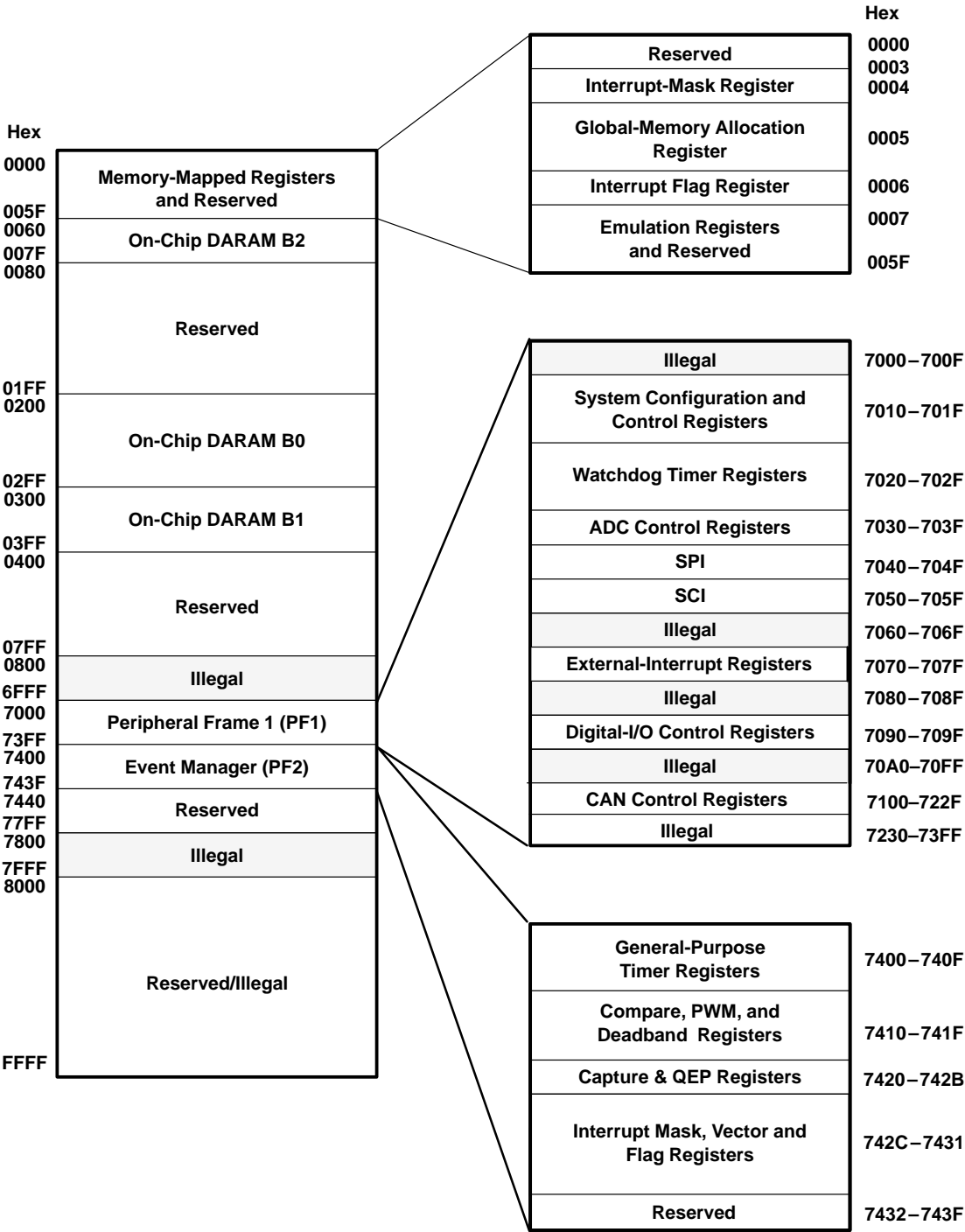


Figure 2. Peripheral Memory Map

## **wait-state generation**

Wait state generation is incorporated in the 'F243 without any external hardware for interfacing the 'F243 with slower off-chip memory and I/O devices. Adding wait states lengthens the time the CPU waits for external memory or an external I/O port to respond when the CPU reads from or writes to that external memory or I/O port. Specifically, the CPU waits one extra cycle (one CLKOUT cycle) for every wait state. The wait states operate on CLKOUT cycle boundaries.

To avoid bus conflicts, writes from the 'F243 always take at least two CLKOUT cycles. The 'F243 offers two options for generating wait states:

- **READY Signal.** With the READY signal, you can externally generate any number of wait states. The READY pin has no effect on accesses to *internal* memory.
- **On-Chip Wait-State Generator.** With this generator, you can generate zero to seven wait states.

### **generating wait states with the READY signal**

When the READY signal is low, the 'F243 waits one CLKOUT cycle and then checks READY again. The 'F243 will not continue executing until the READY signal is driven high; therefore, if the READY signal is not used, it should be pulled high during external accesses.

The READY pin can be used to generate any number of wait states. However, when the 'F243 operates at full speed, it may not respond fast enough to provide a READY-based wait state for the first cycle. For extended wait states using external READY logic, the on-chip wait-state generator should be programmed to generate at least one wait state.

### **generating wait states with the 'F243 on-chip software wait-state generator**

The software wait-state generator can be programmed to generate zero to seven wait states for a given off-chip memory space (program, data, or I/O), regardless of the state of the READY signal. These zero to seven wait states are controlled by the wait-state generator register (WSGR) (FFFFh). For more detailed information on the WSGR and associated bit functions, refer to the *TMS320F243 Reference Set, Volume 2* (literature number TBD).

## digital I/O and shared pin functions

The 'x243 has a total of 26 pins shared between primary functions and I/Os. These pins are divided into two groups:

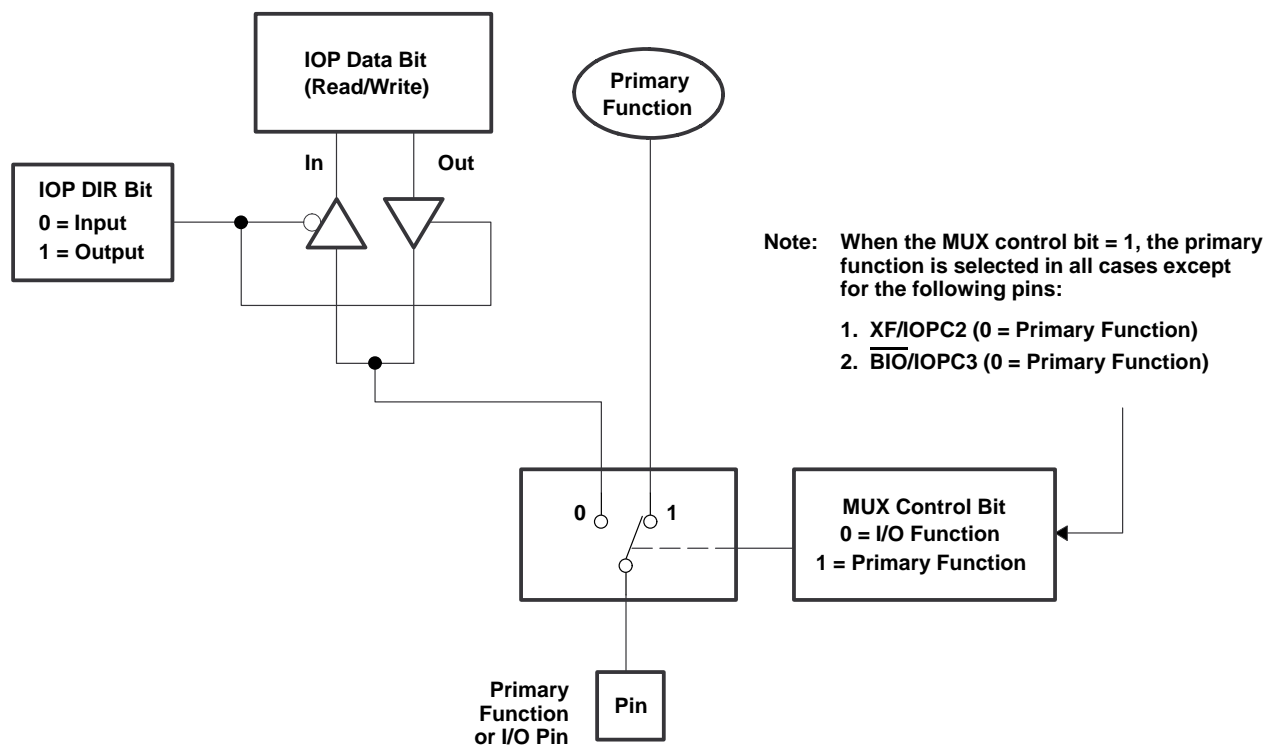
- Group1 — Primary functions shared with I/Os belonging to dedicated I/O ports, Port A, Port B, and Port C.
- Group2 — Primary functions belonging to peripheral modules which also have a built-in I/O feature as a secondary function (for example, SCI, SPI, external interrupts, CAN, and PLL clock modules).

### description of group1 shared I/O pins

The control structure for Group1 type shared I/O pins is shown in Figure 3. The only exception to this configuration is the CLKOUT/IOPC1 pin. In Figure 3, each pin has three bits that define its operation:

- Mux control bit — this bit selects between the primary function (1) and I/O function (0) of the pin.
- I/O direction bit — if the I/O function is selected for the pin (mux control bit is set to 0), this bit determines whether the pin is an input (0) or an output (1).
- I/O data bit — if the I/O function is selected for the pin (mux control bit is set to 0) and the direction selected is an input, data is read from this bit; if the direction selected is an output, data is written to this bit.

The mux control bit, I/O direction bit, and I/O data bit are in the I/O control registers.



**Figure 3. Shared Pin Configuration**

A summary of Group1 pin configurations and associated bits is shown in Table 2.

description of group1 shared I/O pins (continued)

Table 2. Group1 Shared Pin Configurations

PIN #		MUX CONTROL REGISTER (name.bit #)	PIN FUNCTION SELECTED		I/O PORT DATA AND DIRECTION†		
68-PLCC	64-QFP		(CRx.n = 1)	(CRx.n = 0)	REGISTER	DATA BIT #	DIR BIT #
54	43	OCRA.0	SCITXD	IOPA0	PADATDIR	0	8
55	44	OCRA.1	SCIRXD	IOPA1	PADATDIR	1	9
66	55	OCRA.2	XINT1	IOPA2	PADATDIR	2	10
15	8	OCRA.3	CAP1/QEP1	IOPA3	PADATDIR	3	11
14	7	OCRA.4	CAP2/QEP2	IOPA4	PADATDIR	4	12
13	6	OCRA.5	CAP3	IOPA5	PADATDIR	5	13
7	64	OCRA.6	PWM1	IOPA6	PADATDIR	6	14
6	63	OCRA.7	PWM2	IOPA7	PADATDIR	7	15
5	62	OCRA.8	PWM3	IOPB0	PBDATDIR	0	8
4	61	OCRA.9	PWM4	IOPB1	PBDATDIR	1	9
3	60	OCRA.10	PWM5	IOPB2	PBDATDIR	2	10
2	59	OCRA.11	PWM6	IOPB3	PBDATDIR	3	11
19	12	OCRA.12	T1PWM	IOPB4	PBDATDIR	4	12
18	11	OCRA.13	T2PWM	IOPB5	PBDATDIR	5	13
67	56	OCRA.14	TDIR	IOPB6	PBDATDIR	6	14
68	57	OCRA.15	TCLKIN	IOPB7	PBDATDIR	7	15

† Valid only if the I/O function is selected on the pin.

PRODUCT PREVIEW

## description of group2 shared I/O pins

Group2 shared pins belong to peripherals that have built-in general-purpose I/O capability. Control and configuration for these pins are achieved by setting the appropriate bits within the control and configuration registers of the peripherals. Table 3 lists the Group2 shared pins.

**Table 3. Group2 Shared Pin Configurations**

PIN #		MUX CONTROL REGISTER (name.bit #)	PIN FUNCTION SELECTED		I/O PORT DATA AND DIRECTION <sup>†</sup>		
68-PLCC	64-QFP		(CRx.n = 1)	(CRx.n = 0)	REGISTER	DATA BIT #	DIR BIT #
50	39	OCRB.0	IOPC0	XF	PCDATDIR	0	8
53	42	OCRB.1	IOPC1	BIO <sup>‡</sup>	PCDATDIR	1	9
56 <sup>‡</sup>	45	OCRB.2	SPISIMO	IOPC2	PCDATDIR	2	10
57 <sup>‡</sup>	46	OCRB.3	SPISOMI	IOPC3	PCDATDIR	3	11
58 <sup>‡</sup>	47	OCRB.4	SPICLK	IOPC4	PCDATDIR	4	12
59 <sup>‡</sup>	48	OCRB.5	SPISTE	IOPC5	PCDATDIR	5	13
11 <sup>‡</sup>	4	OCRB.6	CANTX	IOPC6	PCDATDIR	6	14
10 <sup>‡</sup>	3	OCRB.7	CANRX	IOPC7	PCDATDIR	7	15
12	5	OCRB.8	IOPD0	CLKOUT	PDDATDIR	0	8
65	54	OCRB.9	XINT2/EXTSOC	IOPD1	PDDATDIR	1	9

<sup>†</sup> Valid only if the I/O function is selected on the pin.

<sup>‡</sup> Dedicated I/O

## digital I/O control registers

Table 4 lists the registers available to the digital I/O module. As with other 'F243 peripherals, the registers are memory-mapped to the data space.

**Table 4. Addresses of Digital I/O Control Registers**

ADDRESS	REGISTER	NAME
7090h	OCRA	I/O mux control register A
7092h	OCRB	I/O mux control register B
7098h	PADATDIR	I/O port A data and direction register
709Ah	PBDATDIR	I/O port B data and direction register
709Ch	PCDATDIR	I/O port C data and direction register
709Eh	PDDATDIR	I/O port D data and direction register



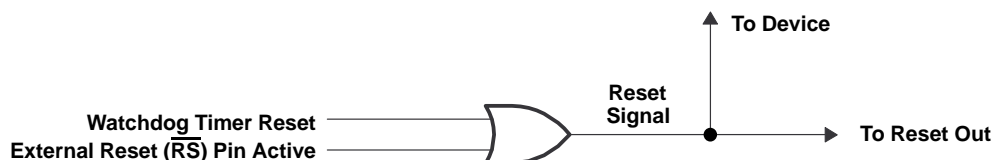
## device reset and interrupts

The TMS320x24x software-programmable interrupt structure supports flexible on-chip and external interrupt configurations to meet real-time interrupt-driven application requirements. The 'x243 recognizes three types of interrupt sources:

- **Reset** (hardware- or software-initiated) is unarbitrated by the CPU and takes immediate priority over any other executing functions. All maskable interrupts are disabled until the reset service routine enables them.  
The 'x243 devices have two sources of reset: an external reset pin and a watchdog timer timeout (reset).
- **Hardware-generated interrupts** are requested by external pins or by on-chip peripherals. There are two types:
  - *External interrupts* are generated by one of four external pins corresponding to the interrupts XINT1, XINT2, PDPINT, and NMI. The first three can be masked both by dedicated enable bits and by the CPU's interrupt mask register (IMR), which can mask each maskable interrupt line at the DSP core. NMI, which is not maskable, takes priority over peripheral interrupts and software-generated interrupts. It can be locked out only by an already executing NMI or a reset.
  - *Peripheral interrupts* are initiated internally by these on-chip peripheral modules: the event manager, SPI, SCI, WD, CAN, and ADC. They can be masked both by enable bits for each event in each peripheral and by the CPU's IMR, which can mask each maskable interrupt line at the DSP core.
- **Software-generated interrupts** for the 'x243 device include:
  - *The INTR instruction.* This instruction allows initialization of any 'x243 interrupt with software. Its operand indicates to which interrupt vector location the CPU branches. This instruction globally disables maskable interrupts (sets the INTM bit to 1).
  - *The NMI instruction.* This instruction forces a branch to interrupt vector location 24h, the same location used for the nonmaskable hardware interrupt NMI. NMI can be initiated by driving the NMI pin low or by executing an NMI instruction. This instruction globally disables maskable interrupts.
  - *The TRAP instruction.* This instruction forces the CPU to branch to interrupt vector location 22h. The TRAP instruction does *not* disable maskable interrupts (INTM is not set to 1); therefore, when the CPU branches to the interrupt service routine, that routine can be interrupted by the maskable hardware interrupts.
  - *An emulator trap.* This interrupt can be generated with either an INTR instruction or a TRAP instruction.

## reset

The reset operation ensures an orderly startup sequence for the device. There are two possible causes of a reset, as shown in Figure 4.



**Figure 4. Reset Signals**

The two possible reset signals are generated as follows:

- **Watchdog timer reset.** A watchdog-timer-generated reset occurs if the watchdog timer overflows or an improper value is written to either the watchdog key register or the watchdog control register. (Note that when the device is powered on, the watchdog timer is automatically active.)
- **Reset pin active.** To generate an external reset pulse on the  $\overline{RS}$  pin, a low-level pulse duration of as little as a few nanoseconds is usually effective; however, pulses of one CPUCLK cycle are necessary to ensure that the device recognizes the reset signal.

Once a reset source is activated, the external  $\overline{RS}$  pin is driven (active) low for a minimum of eight CPUCLK cycles. This allows the TMS320x24x device to reset external system components.

The occurrence of a reset condition causes the TMS320x24x to terminate program execution and affects various registers and status bits. During a reset, RAM contents remain unchanged, and all control bits that are affected by a reset are initialized to their reset state.

## hardware-generated interrupts

The 'C2xx CPU supports one nonmaskable interrupt (NMI) and six maskable prioritized interrupt requests. The 'x243 devices have many peripherals, and each peripheral is capable of generating one or more interrupts in response to many events. Clearly, the 'C2xx CPU does not have sufficient interrupt requests to handle all these peripheral interrupt requests; therefore, a centralized interrupt controller is required to arbitrate the interrupt requests from all the different sources. Throughout this section, refer to Figure 5 .

hardware-generated interrupts (continued)

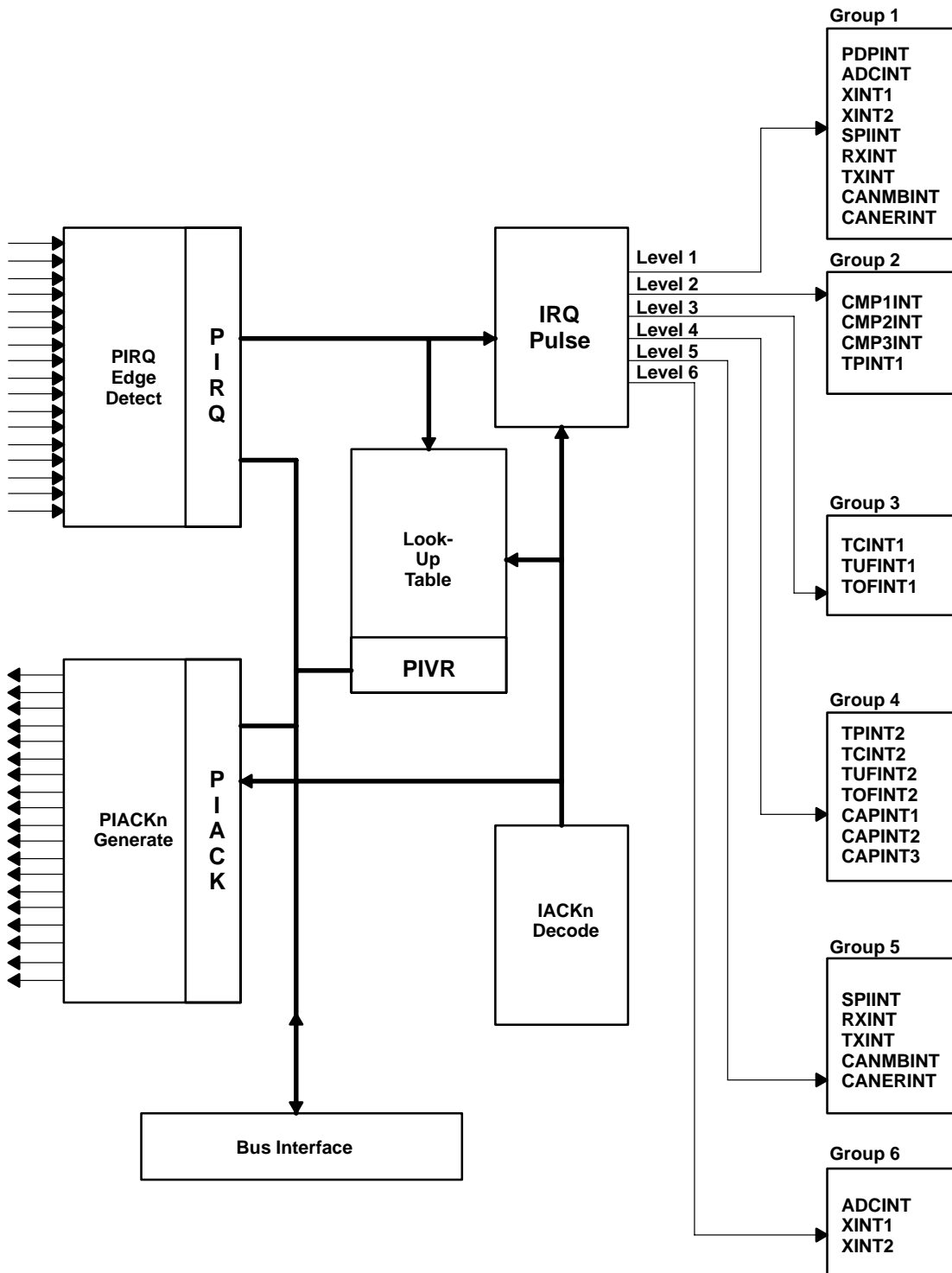


Figure 5. Peripheral Interrupt Expansion Block Diagram

PRODUCT PREVIEW

## ***interrupt hierarchy***

The number of interrupt requests available is expanded by having two levels of hierarchy in the interrupt request system. There are two levels of hierarchy in both the interrupt request/acknowledge hardware and in the interrupt service routine software.

## ***interrupt request structure***

1. At the lower level of the hierarchy, the peripheral interrupt requests (PIRQ) from several peripherals to the interrupt controller are ORed together to generate a request to the CPU. There is an interrupt flag bit and an interrupt enable bit located in the peripheral for each event that can cause a peripheral interrupt request (PIRQ). There is also one PIRQ for each event. If an interrupt-causing event occurs in a peripheral, and the corresponding interrupt enable bit is set, the interrupt request from the peripheral to the interrupt controller is asserted. This interrupt request simply reflects the status of the peripheral's interrupt flag, gated with the interrupt enable bit. When the interrupt flag is cleared, the interrupt request is cleared. Some peripherals have the capability to make either a high-priority or a low-priority interrupt request. If a peripheral has this capability, the value of its interrupt priority bit is transmitted to the interrupt controller. The interrupt request continues to be asserted until it is either automatically cleared by an interrupt acknowledge or cleared by software.
2. At the upper level of the hierarchy, the ORed PIRQs generate interrupt (INT) requests to the CPU. The request to the 'C2xx CPU is a low-going pulse of 2 CPU clock cycles (the pulse duration, etc., should probably be a synthesis parameter to cope with CPUs with different requirements to the 'C2xx). The PIE controller generates an INT pulse when any of the PIRQs controlling that INT go active. If any of the PIRQs capable of asserting that CPU interrupt request are still active in the cycle following an interrupt acknowledge for that INT, another INT pulse is generated (an interrupt acknowledge clears the highest-priority pending PIRQ). Which CPU interrupt requests get asserted by which peripheral interrupt requests, and the relative priority of each peripheral interrupt request, is defined in the interrupt controller and is not part of any of the peripherals. This is shown in Table 5.

*interrupt request structure (continued)*

**Table 5. 'x243-lite Interrupt Source Priority and Vectors**

INTERRUPT NAME	OVERALL PRIORITY	CPU INTERRUPT AND VECTOR ADDRESS	BIT POSITION IN PIRQRx AND PIACKRx	PERIPHERAL INTERRUPT VECTOR	MASKABLE?	SOURCE PERIPHERAL MODULE	DESCRIPTION
Reset	1	RSN 0000h		N/A	N	$\overline{RS}$ pin, Watchdog	Reset from pin, watchdog timeout
Reserved	2	– 0026h		N/A	N	CPU	Emulator Trap
NMI	3	NMI 0024h		N/A	N	Nonmaskable Interrupt	Nonmaskable interrupt
PDPINT	4	INT1 0002h	0.0	0020h	Y	EV	Power device protection interrupt pin
ADCINT	5		0.1	0004h	Y	ADC	ADC interrupt in high-priority mode
XINT1	6		0.2	0001h	Y	External Interrupt Logic	External interrupt pins in high priority
XINT2	7		0.3	0011h	Y	External Interrupt Logic	External interrupt pins in high priority
SPIINT	8		0.4	0005h	Y	SPI	SPI interrupt in high-priority mode
RXINT	9		0.5	0006h	Y	SCI	SCI receiver interrupt in high-priority mode
TXINT	10		0.6	0007h	Y	SCI	SCI transmitter interrupt in high-priority mode
CANMBINT	11		0.7	0040h	Y	CAN	CAN mailbox interrupt (high-priority mode)
CANERINT	12		0.8	0041h	Y	CAN	CAN error interrupt (high-priority mode)
CMP1INT	13	INT2 0004h	0.9	0021h	Y	EV	Compare 1 interrupt
CMP2INT	14		0.10	0022h	Y	EV	Compare 2 interrupt
CMP3INT	15		0.11	0023h	Y	EV	Compare 3 interrupt
TPINT1	16		0.12	0027h	Y	EV	Timer 1 period interrupt
TCINT1	17	INT2 cont. 0004h	0.13	0028h	Y	EV	Timer 1 PWM interrupt
TUFINT1	18		0.14	0029h	Y	EV	Timer 1 underflow interrupt
TOFINT1	19		0.15	002Ah	Y	EV	Timer 1 overflow interrupt
TPINT2	20	INT3 0006h	1.0	002Bh	Y	EV	Timer 2 period interrupt
TCINT2	21		1.1	002Ch	Y	EV	Timer 2 PWM interrupt
TUFINT2	22		1.2	002Dh	Y	EV	Timer 2 underflow interrupt
TOFINT2	23		1.3	002Eh	Y	EV	Timer 2 overflow interrupt
CAPINT1	24	INT4	1.4	0033h	Y	EV	Capture 1 interrupt
CAPINT2	25	0008h	1.5	0034h	Y	EV	Capture 2 interrupt
CAPINT3	26		1.6	0035h	Y	EV	Capture 3 interrupt

**PRODUCT PREVIEW**



**interrupt request structure (continued)**

**Table 5. 'x243-lite Interrupt Source Priority and Vectors (Continued)**

INTERRUPT NAME	OVERALL PRIORITY	CPU INTERRUPT AND VECTOR ADDRESS	BIT POSITION IN PIRQR <sub>x</sub> AND PIACKR <sub>x</sub>	PERIPHERAL INTERRUPT VECTOR	MASKABLE?	SOURCE PERIPHERAL MODULE	DESCRIPTION
SPIINT	27	INT5 000Ah	1.7	0005h	Y	SPI	SPI interrupt (low-priority)
RXINT	28		1.8	0006h	Y	SCI	SCI receiver interrupt (low-priority mode)
TXINT	29		1.9	0007h	Y	SCI	SCI transmitter interrupt (low-priority mode)
CANMBINT	30		1.10	0040h	Y	CAN	CAN mailbox interrupt (low-priority mode)
CANERINT	31		1.11	0041h	Y	CAN	CAN error interrupt (low-priority mode)
ADCINT	32	INT6 000Ch	1.12	0004h	Y	ADC	ADC interrupt (low-priority)
XINT1	33		1.13	0001h	Y	External Interrupt Logic	External interrupt pins (low-priority mode)
XINT2	34		1.14	0011h	Y	External Interrupt Logic	External interrupt pins (low-priority mode)
Reserved		000Eh		N/A	Y	CPU	Analysis interrupt
TRAP	N/A	0022h		N/A	N/A	CPU	TRAP instruction
Phantom Interrupt Vector	N/A	N/A		0000h	N/A	CPU	Phantom interrupt vector

**interrupt acknowledge**

The 'C2xx CPU has one interrupt acknowledge output. The hierarchical interrupt expansion scheme requires one interrupt-acknowledge signal for each peripheral interrupt request (PIRQ) to the interrupt controller. When the CPU asserts its interrupt acknowledge, it simultaneously puts a value on the memory interface program address bus which corresponds to the CPU interrupt being acknowledged (it does this because it is fetching the CPU interrupt vector from program memory, each INT has a vector stored in a dedicated program memory address). This value is shown in Table 5. The PIE controller decodes this value to determine which of the CPU interrupt requests is being acknowledged. It, then, generates a peripheral interrupt acknowledge in response to the highest-priority, currently asserted, peripheral interrupt request associated with that CPU interrupt. The peripheral interrupt acknowledge clears the interrupt flag in the peripheral, and, hence, the peripheral interrupt request.

**interrupt vectors**

When the CPU receives an interrupt request (INT), it does not know which peripheral event caused the request (PIRQ). To enable the CPU to distinguish between all of these events, a unique interrupt vector is generated in response to an active interrupt request getting acknowledged. This vector is loaded into the Peripheral Interrupt Vector Register (PIVR) in the PIE controller and can then be read by the CPU and used to generate a vector to branch to the Interrupt Service Routine (ISR) which corresponds to the event being acknowledged.

In effect, there are two vector tables: The CPU's vector table, which is used to get to the first, General Interrupt Service Routine (GISR) in response to a CPU interrupt request (INT); and the peripheral vector table, which is used to get to the Event-Specific Interrupt Service Routine (SISR) corresponding to the event which caused the peripheral interrupt request (PIRQ). The code in the GISR should read the Peripheral Interrupt Vector Register (PIVR) after saving any necessary context, and use this value to generate a vector to the SISR.

### ***interrupt vectors (continued)***

The peripheral interrupt vector are stored in a table in the peripheral interrupt expansion controller. They can either be hard-coded (potentially ROM), or register locations (RAM), which are programmed by the reset service routine. The peripheral interrupt vectors are all implemented as hard-coded values on the 'x243-lite devices, according to Table 5.

### **phantom interrupt vector**

The phantom interrupt vector is an interrupt system integrity feature. If the CPU's interrupt acknowledge is asserted, but there is no associated peripheral interrupt request asserted, the phantom vector is used so that this fault is handled in a controlled manner. One way the phantom interrupt vector could be required is if the CPU executes a software interrupt instruction with an argument corresponding to a peripheral interrupt (usually INT1–INT6). The other way would be if a peripheral made an interrupt request, but its interrupt request flag was cleared by software before the CPU acknowledged the request. In this case, there may be no peripheral interrupt request asserted to the interrupt controller so the controller would not know which peripheral interrupt vector to load into the PIVR. In these situations, the phantom interrupt vector is loaded into the PIVR in lieu of a peripheral interrupt vector.

### **software hierarchy**

There are two levels of interrupt service routine hierarchy: the General Interrupt Service Routine (GISR), and the Specific Interrupt Service Routine (SISR). There is one GISR for each maskable prioritized request (INT) to the CPU, this performs all necessary context saves before it fetches the peripheral interrupt vector from the PIVR. This vector is used to generate a branch to the SISR. There is one SISR for every interrupt request from a peripheral to the interrupt controller. The SISR performs the actions required in response to the peripheral interrupt request.

### **nonmaskable interrupts**

The PIE controller does not support expansion of nonmaskable interrupts. This is because an ISR must read the peripheral interrupt vector from the PIVR before interrupts are re-enabled (all interrupts are automatically disabled when an interrupt is taken). If the PIVR is not read before interrupts are re-enabled, another interrupt would be acknowledged and a new peripheral interrupt vector would be loaded into the PIVR, causing permanent loss of the original peripheral interrupt vector. Since, by their very nature, nonmaskable interrupts cannot be masked, they cannot be included in the interrupt expansion controller because they could cause the loss of peripheral interrupt vectors.

## *interrupt operation sequence*

1. An interrupt-generating event occurs in a peripheral. The interrupt flag (IF) bit corresponding to that event is set in a register in the peripheral. If the appropriate interrupt enable (IE) bit is set, the peripheral generates an interrupt request to the PIE controller by asserting its PIRQ. If the interrupt is not enabled, the IF remains set until cleared by software. If the interrupt is enabled at a later time, and the interrupt flag is still set, the PIRQ will immediately be asserted.
2. If no unacknowledged CPU interrupt request of the same priority level has previously been sent, the peripheral interrupt request, PIRQ, causes the PIE controller to generate a CPU interrupt request pulse. This pulse is active low for 2 CPU clock cycles (for the 'C2xx CPU).
3. The interrupt request to the CPU sets the corresponding flag in the CPU's interrupt flag register, IFR. If the CPU interrupt has been enabled by setting the appropriate bit in the CPU's Interrupt Mask Register, IMR, the CPU stops what it is doing, mask all other maskable interrupts by setting the INTM bit, save some context, and start executing the General Interrupt Service Routine (GISR) for that interrupt priority level. The CPU generates an interrupt acknowledge automatically, which is accompanied by a value on the Program Address Bus (PAB) that corresponds to the interrupt priority level being responded to. These values are shown in Table 5.
4. The PIE controller decodes the PAB value and it generates a peripheral interrupt acknowledge to clear the peripheral interrupt flag which is asserting the highest-priority PIRQ associated with the CPU interrupt being acknowledged. In addition to asserting the peripheral interrupt vector, the interrupt controller loads the peripheral interrupt vector register (PIVR) with the appropriate peripheral interrupt vector (or the phantom interrupt vector), from the table stored in the PIE controller.
5. When the GISR has completed any necessary context saves, it reads the PIVR and use the interrupt vector as a target (or to generate a target) for a branch to the Specific Interrupt Service Routine (SISR) for the interrupt event which occurred in the peripheral. Interrupts **MUST NOT** be re-enabled until the PIVR has been read; otherwise, its contents can get overwritten by a subsequent interrupt.

## **external interrupts**

The 'x243 has four external interrupts. These interrupts include:

- **XINT1.** The XINT1 control register (at 7070h) provides control and status for this interrupt. XINT1 can be used as a high-priority (Level 1) or low-priority (Level 6) maskable interrupt or as a general-purpose input pin. XINT1 can also be programmed to trigger an interrupt on either the rising or the falling edge.
- **XINT2.** The XINT2 control register (at 7071h) provides control and status for this interrupt. XINT2 can be used as a high-priority (Level 1) or low-priority (Level 6) maskable interrupt or a general-purpose I/O pin. XINT2 can also be programmed to trigger an interrupt on either the rising or the falling edge.
- **NMI.** NMI is a nonmaskable external interrupt or a general-purpose input pin. NMI can also be programmed to trigger an interrupt on either the rising or the falling edge.
- **PDPINT.** This interrupt is provided for safe operation of the power converter and motor drive. This maskable interrupt can put the timers and PWM output pins in high-impedance states and inform the CPU in case of motor drive abnormalities such as overvoltage, overcurrent, and excessive temperature rise. PDPINT is a Level 1 interrupt.

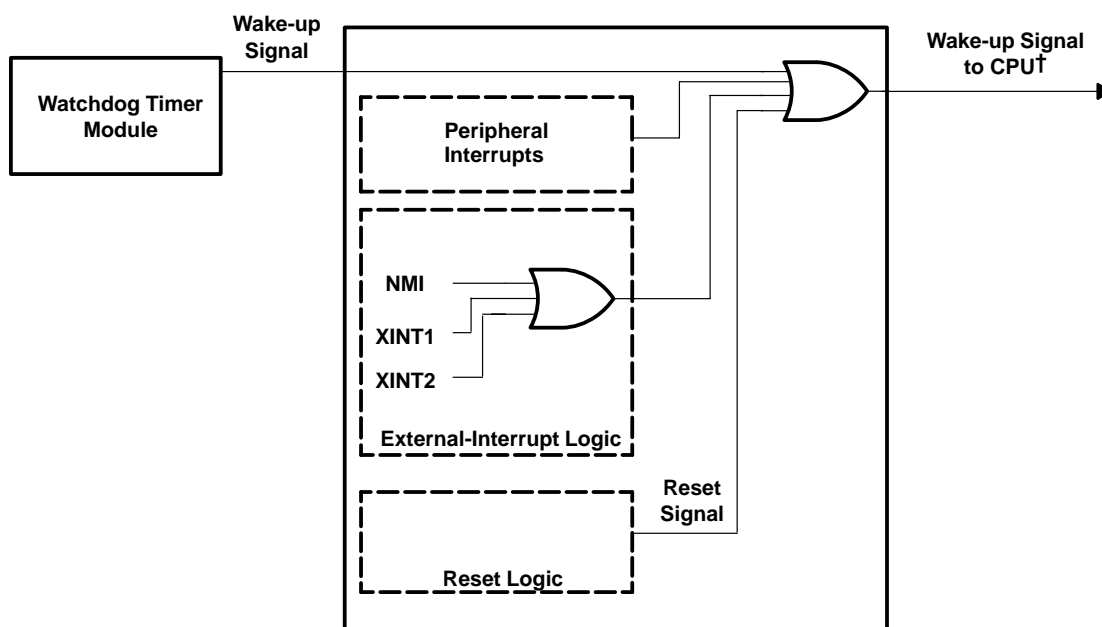


## external interrupts (continued)

Table 6 is a summary of the external interrupt capability of the 'x243.

**Table 6. External Interrupt Types and Functions**

EXTERNAL INTERRUPT	CONTROL REGISTER NAME	CONTROL REGISTER ADDRESS	CAN DO NMI?	DIGITAL I/O PIN	MASKABLE?
XINT1	XINT1CR	7070h	No	Input only	Yes (Level 1 or 6)
XINT2	XINT2CR	7071h	No	I/O	Yes (Level 1 or 6)
NMI	—	—	Yes	Input only	No
$\overline{\text{PDPINT}}$	EVIMRA	742Ch	N/A	N/A	Yes (Level 1)



† The CPU can only exit halt mode (LPM2) with a RESET.

**Figure 6. Waking Up the Device From Power Down**

## clock generation

The TMS320x24x has an on-chip, (x4) PLL-based clock module. This module provides all the necessary clocking signals for the device, as well as control for low-power mode entry. The only external component necessary for this module is an external fundamental crystal.

The PLL-based clock module provides two basic modes of operation: PLL oscillator mode and clock-in mode.

- **Crystal-operation**  
This mode allows the use of a 5-MHz external reference crystal to provide the time base to the device.

## clock generation (continued)

- External clock source operation  
This mode allows the internal crystal oscillator and PLL circuitry to be bypassed. The device clocks are generated from an external clock source input on the XTAL1/CLKIN pin.

The clock module includes two external pins:

1. XTAL1/CLKIN      clock source/crystal input
2. XTAL2              output to crystal

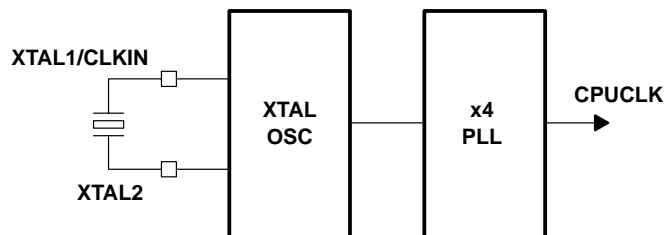


Figure 7. PLL Clock Module Block Diagram

## low-power modes

The 'C2xx has an IDLE instruction. When executed, the IDLE instruction stops the clocks to all circuits in the CPU, but the clock output from the CPU continues to run. With this instruction, the CPU clocks can be shut down to save power while the peripherals (clocked with CLKOUT) continue to run. The CPU exits the IDLE state if it is reset, or, if it receives an interrupt request.

## clock domains

All 'C2xx-based devices have two clock domains:

1. CPU clock domain which consists of the clock for most of the CPU logic
2. System clock domain which consists of the peripheral clock (which is derived from CLKOUT of the CPU) and the clock for the interrupt logic in the CPU.

When the CPU goes into IDLE mode, the CPU clock domain is stopped while the system clock domain continues to run. This mode is also known as IDLE1 mode. The 'C2xx CPU also contains support for a second IDLE mode, IDLE2, to be implemented in external logic. By asserting the IDLE2 input to the 'C2xx CPU, both the CPU clock domain and the system clock domain are stopped, allowing further power savings. A third low-power mode, Halt mode, the deepest, is possible if the oscillator and WDCLK are also shut down when in IDLE2 mode. It is not possible to power down the PLL embedded in the 'C2xx CPU core.

There are two control bits, LPM(1) and LPM(0), which specify which of the three possible low-power modes is entered when the IDLE instruction is executed. See Table 7. These bits are located in the System Control and Status Register (SCSR) described in the *TMS320x243 User's Guide, Volume 2* (literature number TBD).

clock domains (continued)

Table 7. Low-Power Modes Summary

LOW-POWER MODE	LPMx BITS SCSR[12:13]	CPU CLOCK DOMAIN	SYSTEM CLOCK DOMAIN	WDCLK STATUS	PLL STATUS	OSC STATUS	EXIT CONDITION	DESCRIPTION
X + not IDLE	XX	On	On	On	On	On	—	RUN mode
0 + IDLE LPM0	00	Off	On	On	On	On	Peripheral Interrupt, External Interrupt, Reset	IDLE1
1 + IDLE LPM1	01	Off	Off	On	On	On	Wakeup Interrupts, External Interrupt, Reset	IDLE2
2/3 + IDLE LPM2	1X	Off	Off	Off	Off	Off	Reset Only	HALT

wakeup from low-power modes

**reset**

A reset (from any source) causes the device to exit any of the Idle modes. If the device is halted, the reset will first start the oscillator, and there can be a delay while the oscillator powers up before clocks are generated to initiate the CPU reset sequence.

**external interrupts**

The external interrupts, XINTx, can cause the device to exit any of the low-power modes, except Halt. If the device is in IDLE2 mode, the synchronous logic connected to the external interrupt pins is bypassed with combinatorial logic which recognizes the interrupt on the pin, starts the clocks, and then allows the clocked logic to generate an interrupt request to the PIE controller. Note that in Table 7, external interrupts include PDPINT.

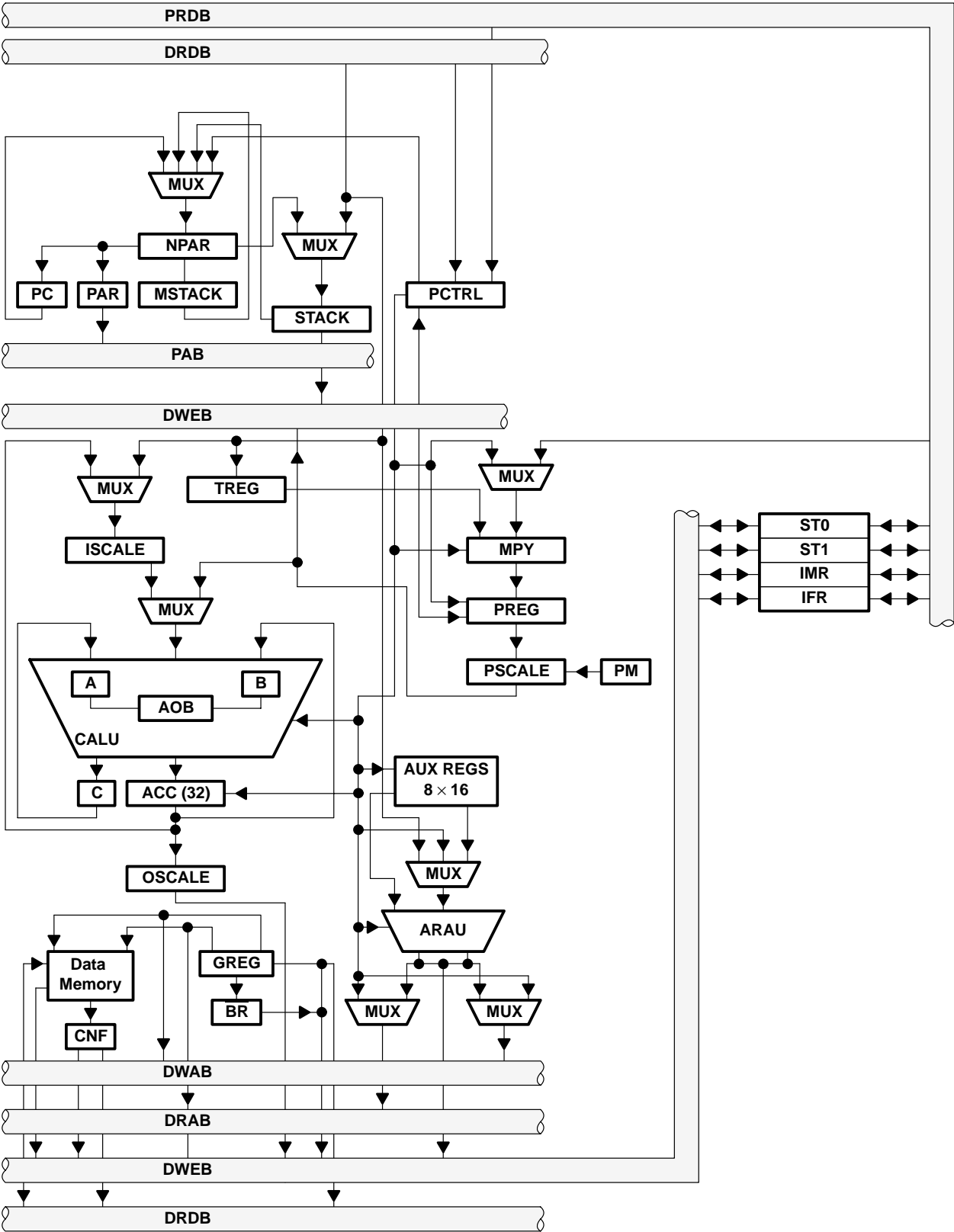
**wakeup interrupts**

Certain peripherals (for example, the CAN wakeup interrupt which can assert the CAN error interrupt request even when there are no clocks running) can have the capability to start the device clocks and then generate an interrupt in response to certain external events, for example, activity on a communication line.

TMS320F243  
DSP CONTROLLER

SPRS064 – DECEMBER 1997

functional block diagram of TMS320x24x DSP CPU



PRODUCT PREVIEW



## 'x243 DSP core CPU

The TMS320x24x devices use an advanced Harvard-type architecture that maximizes processing power by maintaining two separate memory bus structures — program and data — for full-speed execution. This multiple bus structure allows data and instructions to be read simultaneously. Instructions support data transfers between program memory and data memory. This architecture permits coefficients that are stored in program memory to be read in RAM, thereby eliminating the need for a separate coefficient ROM. This, coupled with a four-deep pipeline, allows the 'x243 devices to execute most instructions in a single cycle.

### status and control registers

Two status registers, ST0 and ST1, contain the status of various conditions and modes. These registers can be stored into data memory and loaded from data memory, thus allowing the status of the machine to be saved and restored for subroutines.

The load status register (LST) instruction is used to write to ST0 and ST1. The store status register (SST) instruction is used to read from ST0 and ST1 — except for the INTM bit, which is not affected by the LST instruction. The individual bits of these registers can be set or cleared when using the SETC and CLRC instructions. Figure 8 shows the organization of status registers ST0 and ST1, indicating all status bits contained in each. Several bits in the status registers are reserved and are read as logic 1s. Table 8 lists status register field definitions.

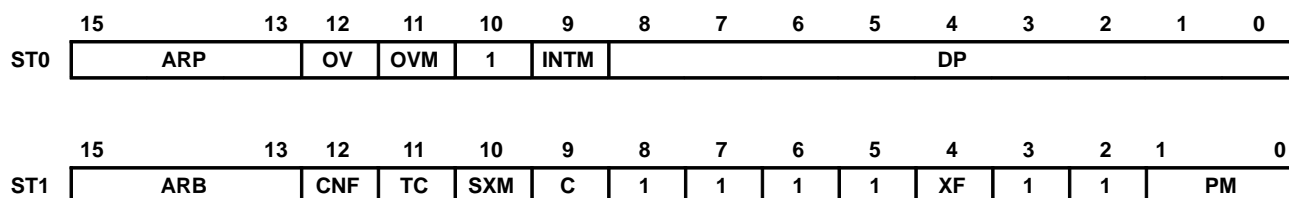


Figure 8. Status and Control Register Organization

Table 8. Status Register Field Definitions

FIELD	FUNCTION
ARB	Auxiliary register pointer buffer. When the ARP is loaded into ST0, the old ARP value is copied to the ARB except during an LST instruction. When the ARB is loaded by way of an LST #1 instruction, the same value is also copied to the ARP.
ARP	Auxiliary register (AR) pointer. ARP selects the AR to be used in indirect addressing. When the ARP is loaded, the old ARP value is copied to the ARB register. ARP can be modified by memory-reference instructions when using indirect addressing, and by the LARP, MAR, and LST instructions. The ARP is also loaded with the same value as ARB when an LST #1 instruction is executed.
C	Carry Bit. C is set to 1 if the result of an addition generates a carry, or reset to 0 if the result of a subtraction generates a borrow. Otherwise, C is reset after an addition or set after a subtraction, except if the instruction is ADD or SUB with a 16-bit shift. In these cases, the ADD can only set and the SUB only reset the carry bit, but cannot affect it otherwise. The single-bit shift and rotate instructions also affect C, as well as the SETC, CLRC, and LST #1 instructions. Branch instructions have been provided to branch on the status of C. C is set to 1 on a reset.
CNF	On-chip RAM configuration control bit. If CNF is set to 0, the reconfigurable data dual-access RAM blocks are mapped to data space; otherwise, they are mapped to program space. The CNF can be modified by the SETC CNF, CLRC CNF, and LST #1 instructions. RS sets the CNF to 0.
DP	Data memory page pointer. The 9-bit DP register is concatenated with the seven LSBs of an instruction word to form a direct memory address of 16 bits. DP can be modified by the LST and LDP instructions.
INTM	Interrupt mode bit. When INTM is set to 0, all unmasked interrupts are enabled. When set to 1, all maskable interrupts are disabled. INTM is set and reset by the SETC INTM and CLRC INTM instructions. RS and IACK also set INTM. INTM has no effect on the unmaskable RS and NMI interrupts. Note that INTM is unaffected by the LST instruction. This bit is set to 1 by reset. It is also set to 1 when a maskable interrupt trap is taken.
OV	Overflow flag bit. As a latched overflow signal, OV is set to 1 when overflow occurs in the arithmetic logic unit (ALU). Once an overflow occurs, the OV remains set until a reset, BCND/D on OV/NOV, or LST instructions clear OV.

# TMS320F243 DSP CONTROLLER

SPRS064 – DECEMBER 1997

## status and control registers (continued)

**Table 8. Status Register Field Definitions (Continued)**

FIELD	FUNCTION
OVM	Overflow mode bit. When OVM is set to 0, overflowed results overflow normally in the accumulator. When set to 1, the accumulator is set to either its most positive or negative value upon encountering an overflow. The SETC and CLRC instructions set and reset this bit, respectively. LST can also be used to modify the OVM.
PM	Product shift mode. If these two bits are 00, the multiplier's 32-bit product is loaded into the ALU with no shift. If PM = 01, the PREG output is left-shifted one place and loaded into the ALU, with the LSB zero-filled. If PM = 10, PREG output is left-shifted by four bits and loaded into the ALU, with the LSBs zero-filled. PM = 11 produces a right shift of six bits, sign-extended. Note that the PREG contents remain unchanged. The shift takes place when transferring the contents of the PREG to the ALU. PM is loaded by the SPM and LST #1 instructions. PM is cleared by RS.
SXM	Sign-extension mode bit. SXM = 1 produces sign extension on data as it is passed into the accumulator through the scaling shifter. SXM = 0 suppresses sign extension. SXM does not affect the definitions of certain instructions; for example, the ADDS instruction suppresses sign extension regardless of SXM. SXM is set by the SETC SXM and reset by the CLRC SXM instructions, and can be loaded by the LST #1. SXM is set to 1 by reset.
TC	Test / control flag bit. TC is affected by the BIT, BITT, CMPR, LST #1, and NORM instructions. TC is set to a 1 if a bit tested by BIT or BITT is a 1, if a compare condition tested by CMPR exists between AR (ARP) and AR0, if the exclusive-OR function of the two MSBs of the accumulator is true when tested by a NORM instruction. The conditional branch, call, and return instructions can execute based on the condition of TC.
XF	XF pin status bit. XF indicates the state of the XF pin, a general-purpose output pin. XF is set by the SETC XF and reset by the CLRC XF instructions. XF is set to 1 by reset.

## central processing unit

The TMS320x24x central processing unit (CPU) contains a 16-bit scaling shifter, a 16 x 16-bit parallel multiplier, a 32-bit central arithmetic logic unit (CALU), a 32-bit accumulator, and additional shifters at the outputs of both the accumulator and the multiplier. This section describes the CPU components and their functions. The functional block diagram shows the components of the CPU.

### input scaling shifter

The TMS320x24x provides a scaling shifter with a 16-bit input connected to the data bus and a 32-bit output connected to the CALU. This shifter operates as part of the path of data coming from program or data space to the CALU and requires no cycle overhead. It is used to align the 16-bit data coming from memory to the 32-bit CALU. This is necessary for scaling arithmetic as well as aligning masks for logical operations.

The scaling shifter produces a left shift of 0 to 16 on the input data. The LSBs of the output are filled with zeros; the MSBs can either be filled with zeros or sign-extended, depending upon the value of the SXM bit (sign-extension mode) of status register ST1. The shift count is specified by a constant embedded in the instruction word or by a value in TREG. The shift count in the instruction allows for specific scaling or alignment operations specific to that point in the code. The TREG base shift allows the scaling factor to be adaptable to the system's performance.

### multiplier

The TMS320x24x devices use a 16 x 16-bit hardware multiplier that is capable of computing a signed or an unsigned 32-bit product in a single machine cycle. All multiply instructions, except the MPYU (multiply unsigned) instruction, perform a signed multiply operation. That is, two numbers being multiplied are treated as 2s-complement numbers, and the result is a 32-bit 2s-complement number. There are two registers associated with the multiplier, as follow:

- 16-bit temporary register (TREG) that holds one of the operands for the multiplier
- 32-bit product register (PREG) that holds the product

Four product shift modes (PM) are available at the PREG output (PSCALE). These shift modes are useful for performing multiply/accumulate operations, performing fractional arithmetic, or justifying fractional products. The PM field of status register ST1 specifies the PM shift mode, as shown in Table 9.

**multiplier (continued)**

**Table 9. PSCALE Product Shift Modes**

PM	SHIFT	DESCRIPTION
00	No shift	Product feed to CALU or data bus with no shift
01	Left 1	Removes the extra sign bit generated in a 2s-complement multiply to produce a Q31 product
10	Left 4	Removes the extra 4 sign bits generated in a 16x13 2s-complement multiply to produce a Q31 product when using the multiply by a 13-bit constant
11	Right 6	Scales the product to allow up to 128 product accumulation without the possibility of accumulator overflow

The product can be shifted one bit to compensate for the extra sign bit gained in multiplying two 16-bit 2s-complement numbers (MPY instruction). A four-bit shift is used in conjunction with MPY instruction with a short immediate value (13 bits or less) to eliminate the four extra sign bits gained in multiplying a 16-bit number by a 13-bit number. Finally, the output of PREG can be right-shifted 6 bits to enable the execution of up to 128 consecutive multiply/accumulates without the possibility of overflow.

The LT (load TREG) instruction normally loads TREG to provide one operand (from the data bus), and the MPY (multiply) instruction provides the second operand (also from the data bus). A multiplication also can be performed with a 13-bit immediate operand when using the MPY instruction. Then a product is obtained every two cycles. When the code is executing multiple multiplies and product sums, the CPU supports the pipelining of the TREG load operations with CALU operations using the previous product. The pipeline operations that run in parallel with loading the TREG include: load ACC with PREG (LTP); add PREG to ACC (LTA); add PREG to ACC and shift TREG input data (DMOV) to next address in data memory (LTD); and subtract PREG from ACC (LTS).

Two multiply/accumulate instructions (MAC and MACD) fully utilize the computational bandwidth of the multiplier, allowing both operands to be processed simultaneously. The data for these operations can be transferred to the multiplier each cycle by way of the program and data buses. This facilitates single-cycle multiply/accumulates when used with the repeat (RPT) instruction. In these instructions, the coefficient addresses are generated by program address generation (PAGEN) logic, while the data addresses are generated by data address generation (DAGEN) logic. This allows the repeated instruction to access the values from the coefficient table sequentially and step through the data in any of the indirect addressing modes.

The MACD instruction, when repeated, supports filter constructs (weighted running averages) so that as the sum-of-products is executed, the sample data is shifted in memory to make room for the next sample and to throw away the oldest sample.

The MPYU instruction performs an unsigned multiplication, which greatly facilitates extended-precision arithmetic operations. The unsigned contents of TREG are multiplied by the unsigned contents of the addressed data memory location, with the result placed in PREG. This process allows the operands of greater than 16 bits to be broken down into 16-bit words and processed separately to generate products of greater than 32 bits. The SQRA (square/add) and SQRS (square/subtract) instructions pass the same value to both inputs of the multiplier for squaring a data memory value.

After the multiplication of two 16-bit numbers, the 32-bit product is loaded into the 32-bit product register (PREG). The product from PREG can be transferred to the CALU or to data memory by way of the SPH (store product high) and SPL (store product low). Note: the transfer of PREG to either the CALU or data bus passes through the PSCALE shifter, and therefore is affected by the product shift mode defined by PM. This is important when saving PREG in an interrupt-service-routine context save as the PSCALE shift effects cannot be modeled in the restore operation. PREG can be cleared by executing the MPY #0 instruction. The product register can be restored by loading the saved low half into TREG and executing a MPY #1 instruction. The high half, then, is loaded using the LPH instruction.

### central arithmetic logic unit

The TMS320x24x central arithmetic logic unit (CALU) implements a wide range of arithmetic and logical functions, the majority of which execute in a single clock cycle. This ALU is referred to as central to differentiate it from a second ALU used for indirect-address generation called the auxiliary register arithmetic unit (ARAU). Once an operation is performed in the CALU, the result is transferred to the accumulator (ACC) where additional operations, such as shifting, can occur. Data that is input to the CALU can be scaled by ISCALE when coming from one of the data buses (DRDB or PRDB) or scaled by PSCALE when coming from the multiplier.

The CALU is a general-purpose arithmetic/logic unit that operates on 16-bit words taken from data memory or derived from immediate instructions. In addition to the usual arithmetic instructions, the CALU can perform Boolean operations, facilitating the bit manipulation ability required for a high-speed controller. One input to the CALU is always provided from the accumulator, and the other input can be provided from the product register (PREG) of the multiplier or the output of the scaling shifter (that has been read from data memory or from the ACC). After the CALU has performed the arithmetic or logical operation, the result is stored in the accumulator.

The TMS320x24x devices support floating-point operations for applications requiring a large dynamic range. The NORM (normalization) instruction is used to normalize fixed-point numbers contained in the accumulator by performing left shifts. The four bits of the TREG define a variable shift through the scaling shifter for the LACT/ADDT/SUBT (load/add to /subtract from accumulator with shift specified by TREG) instructions. These instructions are useful in floating-point arithmetic where a number needs to be denormalized — that is, floating-point to fixed-point conversion. They are also useful in execution of an automatic gain control (AGC) going into a filter. The BITT (bit test) instruction provides testing of a single bit of a word in data memory based on the value contained in the four LSBs of TREG.

The CALU overflow saturation mode can be enabled/disabled by setting/resetting the OVM bit of ST0. When the CALU is in the overflow saturation mode and an overflow occurs, the overflow flag is set and the accumulator is loaded with either the most positive or the most negative value representable in the accumulator, depending on the direction of the overflow. The value of the accumulator at saturation is 07FFFFFFh (positive) or 080000000h (negative). If the OVM (overflow mode) status register bit is reset and an overflow occurs, the overflowed results are loaded into the accumulator with modification. (Note that logical operations cannot result in overflow.)

The CALU can execute a variety of branch instructions that depend on the status of the CALU and the accumulator. These instructions can be executed conditionally based on any meaningful combination of these status bits. For overflow management, these conditions include the OV (branch on overflow) and EQ (branch on accumulator equal to zero). In addition, the BACC (branch to address in accumulator) instruction provides the ability to branch to an address specified by the accumulator (computed goto). Bit test instructions (BIT and BITT), which do not affect the accumulator, allow the testing of a specified bit of a word in data memory.

The CALU also has an associated carry bit that is set or reset depending on various operations within the device. The carry bit allows more efficient computation of extended-precision products and additions or subtractions. It also is useful in overflow management. The carry bit is affected by most arithmetic instructions as well as the single-bit shift and rotate instructions. It is not affected by loading the accumulator, logical operations, or other such non-arithmetic or control instructions.

The ADDC (add to accumulator with carry) and SUBB (subtract from accumulator with borrow) instructions use the previous value of carry in their addition/subtraction operation.

The one exception to the operation of the carry bit is in the use of ADD with a shift count of 16 (add to high accumulator) and SUB with a shift count of 16 (subtract from high accumulator) instructions. This case of the ADD instruction can set the carry bit only if a carry is generated, and this case of the SUB instruction can reset the carry bit only if a borrow is generated; otherwise, neither instruction affects it.



### **central arithmetic logic unit (continued)**

Two conditional operands, C and NC, are provided for branching, calling, returning, and conditionally executing, based upon the status of the carry bit. The SETC, CLRC, and LST #1 instructions also can be used to load the carry bit. The carry bit is set to one on a hardware reset.

### **accumulator**

The 32-bit accumulator is the registered output of the CALU. It can be split into two 16-bit segments for storage in data memory. Shifters at the output of the accumulator provide a left shift of 0 to 7 places. This shift is performed while the data is being transferred to the data bus for storage. The contents of the accumulator remain unchanged. When the post-scaling shifter is used on the high word of the accumulator (bits 16–31), the MSBs are lost and the LSBs are filled with bits shifted in from the low word (bits 0–15). When the post-scaling shifter is used on the low word, the LSBs are zero-filled.

The SFL and SFR (in-place one-bit shift to the left/right) instructions and the ROL and ROR (rotate to the left/right) instructions implement shifting or rotating of the contents of the accumulator through the carry bit. The SXM bit affects the definition of the SFR (shift accumulator right) instruction. When SXM = 1, SFR performs an arithmetic right shift, maintaining the sign of the accumulator data. When SXM = 0, SFR performs a logical shift, shifting out the LSBs and shifting in a zero for the MSB. The SFL (shift accumulator left) instruction is not affected by the SXM bit and behaves the same in both cases, shifting out the MSB and shifting in a zero. Repeat (RPT) instructions can be used with the shift and rotate instructions for multiple-bit shifts.

### **auxiliary registers and auxiliary-register arithmetic unit (ARAU)**

The 'x243 provides a register file containing eight auxiliary registers (AR0–AR7). The auxiliary registers are used for indirect addressing of the data memory or for temporary data storage. Indirect auxiliary-register addressing allows placement of the data memory address of an instruction operand into one of the auxiliary registers. These registers are referenced with a 3-bit auxiliary register pointer (ARP) that is loaded with a value from 0 through 7, designating AR0 through AR7, respectively. The auxiliary registers and the ARP can be loaded from data memory, the ACC, the product register, or by an immediate operand defined in the instruction. The contents of these registers also can be stored in data memory or used as inputs to the CALU.

The auxiliary register file (AR0–AR7) is connected to the ARAU. The ARAU can autoindex the current auxiliary register while the data memory location is being addressed. Indexing either by  $\pm 1$  or by the contents of the AR0 register can be performed. As a result, accessing tables of information does not require the CALU for address manipulation; therefore, the CALU is free for other operations in parallel.

### **memory**

The TMS320x243 devices are configured with the following memory modules:

- Dual-access random-access memory (DARAM)
- Flash

### **dual-access RAM (DARAM)**

There are 544 words  $\times$  16 bits of DARAM on the 'x243 device. The 'x243 DARAM allows writes to and reads from the RAM in the same cycle. The DARAM is configured in three blocks: block 0 (B0), block 1 (B1), and block 2 (B2). Block 1 contains 256 words and Block 2 contains 32 words, and both blocks are located only in data memory space. Block 0 contains 256 words, and can be configured to reside in either data or program memory space.

### ***dual-access RAM (DARAM) (continued)***

When using on-chip RAM, or high-speed external memory, the 'x243 runs at full speed with no wait states. The ability of the DARAM to allow two accesses to be performed in one cycle coupled with the parallel nature of the 'x243 architecture enables the device to perform three concurrent memory accesses in any given machine cycle. Externally, the READY line can be used to interface the 'x243 to slower, less expensive external memory. Downloading programs from slow off-chip memory to on-chip RAM can speed processing while cutting system costs.

### ***flash EEPROM (TMS320F243 only)***

The 'F243 device replaces the mask ROM with 8K words of flash electrically erasable programmable read-only memory (EEPROM). The 'C2xx CPU can fetch data and instructions from the EEPROM. Reading the flash is identical to other one-cycle-access internal memories and does not require any interaction with the flash control registers. Before programming, the flash EEPROM module generates the necessary voltages internally, making it unnecessary to provide the programming or erase voltages externally.

### **peripherals**

The integrated peripherals of the TMS320x243 are described in the following subsections:

- External memory interface
- Event manager (EV)
- Single analog-to-digital converter (ADC)
- Serial peripheral interface (SPI)
- Serial communications interface (SCI)
- Controller area network module (CAN)
- Watchdog timer (WD)

## event-manager (EV) module

The event-manager module includes general-purpose (GP) timers, full compare/PWM units, capture units, and quadrature-encoder pulse (QEP) circuits. Figure 9 shows the functions of the event manager.

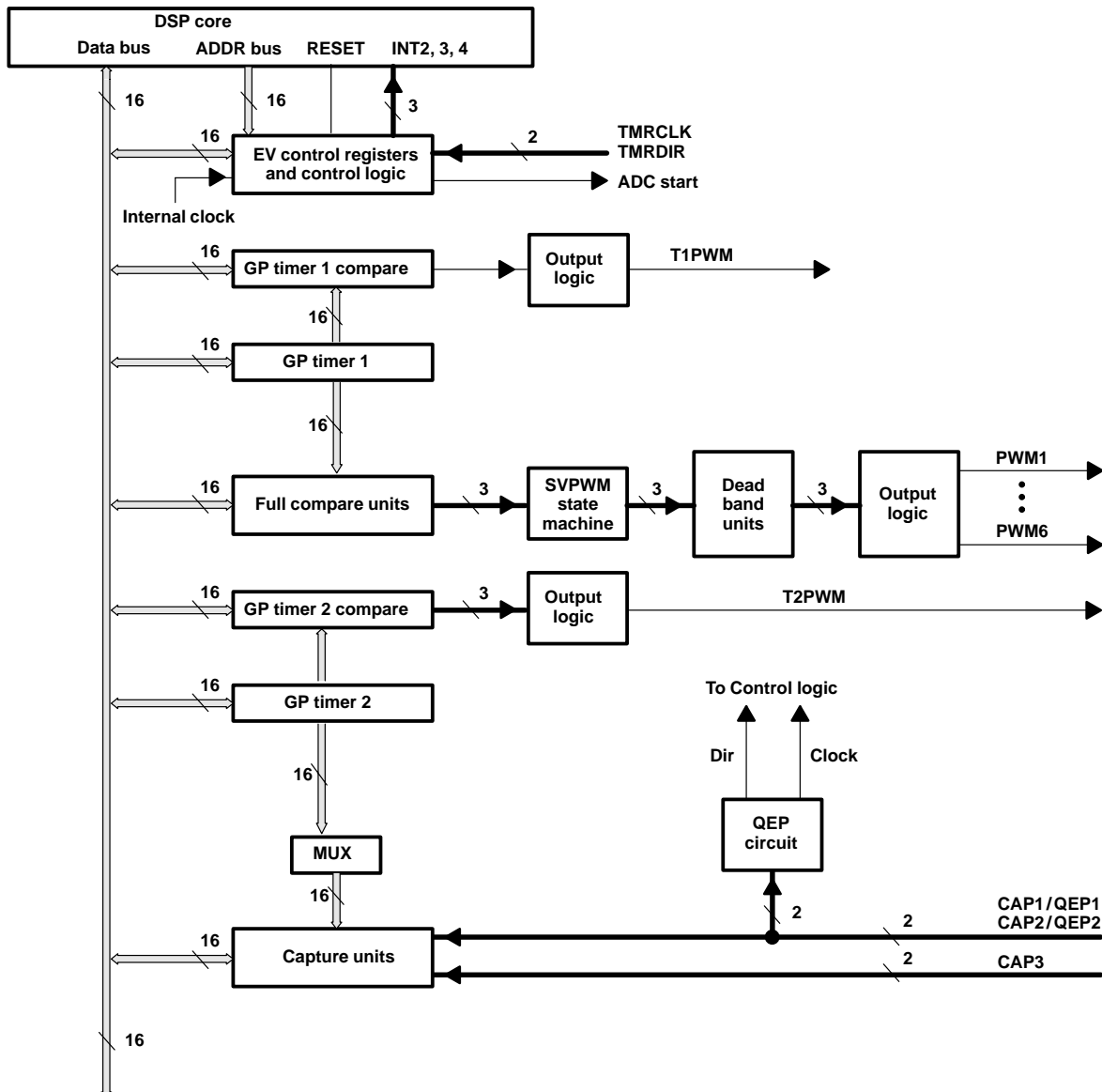


Figure 9. Event-Manager Block Diagram

## ***general-purpose (GP) timers***

There are two GP timers on the TMS320x24x. The GP timer x (for x = 1 or 2) includes:

- A 16-bit timer up-, up/down-counter, TxCNT for reads or writes
- A 16-bit timer-compare register (double-buffered with shadow register), TxCMPR for reads or writes
- A 16-bit timer-period register (double-buffered with shadow register), TxPR for reads or writes
- A 16-bit timer-control register, TxCON for reads or writes
- Selectable internal or external input clocks
- A programmable prescaler for internal or external clock inputs
- Control and interrupt logic, for four maskable interrupts: underflow, overflow, timer compare, and period interrupts
- A selectable direction (TMRDIR) input pin (to count up or down when directional up-/down-count mode is selected)

The GP timers can be operated independently or synchronized with each other. The compare register associated with each GP timer can be used for compare function and PWM-waveform generation. There are three continuous modes of operations for each GP timer in up- or up/down-counting operations. Internal or external input clocks with programmable prescaler is used for each GP timer. GP timers also provide the time base for the other event-manager submodules: GP timer 1 for all the compares and PWM circuits, GP timer 2 for the capture units and the quadrature-pulse counting operations.

Double buffering of the period and compare registers allows programmable change of the timer (PWM) period and the compare/PWM pulse width as needed.

## ***full compare units***

There are three full compare units on TMS320x24x. These compare units use GP timer1 as the timebase and generate six outputs for compare and PWM-waveform generation using programmable deadband circuit. The state of each of the six outputs is configured independently. The compare registers of the compare units are double buffered, allowing programmable change of the compare/PWM pulse widths as needed.

## ***programmable-deadband generator***

The deadband generator circuit includes three 8-bit counters and an 8-bit compare register. Desired deadband value (from 0 to 102  $\mu$ s) can be programmed into the compare register for the outputs of the three compare units. The deadband generation can be enabled/disabled for each compare unit output individually. The deadband-generator circuit produces two outputs (with or without deadband zone) for each compare unit output signal. The output states of the deadband generator are configurable and changeable as needed by way of the double-buffered ACTR register.

## ***PWM waveform generation***

Up to 8 PWM waveforms (outputs) can be generated simultaneously by TMS320x24x: three independent pairs (six outputs) by the three full compare units with *programmable deadbands*, and two independent PWMs by the GP-timer compares.

### ***PWM characteristics***

Characteristics of the PWMs are as follow:

- 16-bit registers
- Programmable deadband for the PWM output pairs, from 0 to 102  $\mu$ s
- Minimum deadband width of 50 ns
- Change of the PWM carrier frequency for PWM frequency wobbling as needed
- Change of the PWM pulse widths within and after each PWM period as needed
- External-maskable power and drive-protection interrupts
- Pulse-pattern-generator circuit, for programmable generation of asymmetric, symmetric, and four-space vector PWM waveforms
- Minimized CPU overhead using auto-reload of the compare and period registers

### ***capture unit***

The capture unit provides a logging function for different events or transitions. The values of the GP timer 2 counter are captured and stored in the two-level FIFO stacks when selected transitions are detected on capture input pins, CAPx for x = 1, 2, or 3. The capture unit of the TMS320x24x consists of four capture circuits.

- Capture units include the following features:
  - One 16-bit capture control register, CAPCON (R/W)
  - One 16-bit capture FIFO status register, CAPFIFO (8 MSBs are read-only, 8 LSBs are write-only)
  - Selection of GP Timer 2 as the time base
  - Three 16-bit 2-level-deep FIFO stacks, one for each capture unit
  - Three Schmitt-triggered capture input pins CAP1, CAP2, and CAP3, one input pin per each capture unit. [All inputs are synchronized with the device (CPU) clock. In order for a transition to be captured, the input must hold at its current level to meet two rising edges of the device clock. The input pins CAP1 and CAP2 can also be used as QEP inputs to the QEP circuit.]
  - User-specified transition (rising edge, falling edge, or both edges) detection
  - Three maskable interrupt flags, one for each capture unit

### ***quadrature-encoder pulse (QEP) circuit***

Two capture inputs (CAP1 and CAP2) can be used to interface the on-chip QEP circuit with a quadrature encoder pulse. Full synchronization of these inputs is performed on-chip. Direction or leading-quadrature pulse sequence is detected, and GP timer 2 is incremented or decremented by the rising and falling edges of the two input signals (four times the frequency of either input pulse).

### analog-to-digital converter (ADC) module

A simplified functional block diagram of the ADC module is shown in Figure 10. The ADC module consists of a 10-bit ADC with a built-in sample-and-hold (S/H) circuit. A total of 8 analog input channels is available on the TMS320x243. Eight analog inputs are provided by way of an 8 to 1 analog multiplexer. Maximum total conversion time for each ADC unit is 1  $\mu$ s. Total accuracy for each converter is plus or minus one and a half LSB. Reference voltage of the ADC module is 0–5 V that can be supplied internally (on-chip) or externally.

Functions of the ADC module include:

- The ADC unit can perform single or continuous S/H and conversion operations. When in continuous conversion mode, the ADC generates two results every 1700 ns (with a 20-MHz clock and a prescale factor of 1). These two results can be two separate analog inputs.
- Two 2-level-deep FIFO result registers.
- Conversion can be started by software, an external signal transition on a device pin, or by certain event manager events.
- The ADC control register is double buffered (with a shadow register) and can be written to at any time. A new conversion can start either immediately or when the previous conversion process is completed.
- In single conversion mode, at the end of each conversion, an interrupt flag is set and the peripheral interrupt request (PIRQ) is generated if it is unmasked/enabled.
- The result of previous conversions stored in data registers will be lost when a third result is stored in the two-deep data FIFO.

### A/D overview

The “pseudo” dual ADC is based around a 10-bit string/capacitor converter with the switched capacitor string providing an inherent S/H function. (Note: There is only one converter with only one inherent S/H circuit.) This peripheral behaves as though there are two analog converters, ADC #1 and ADC #2, but in actual fact, it only uses one converter. This feature makes the A/D software compatible with the C240's A/D and also allows two values (e.g., voltage and current) to be converted almost simultaneously with one conversion request. Minimum conversion time for the ADC unit is 1  $\mu$ s when the converter is clocked at 20 MHz. Total accuracy for the converter is  $\pm 1$  LSB. External high and low reference voltage must be supplied. The upper and lower references can be set to any voltage less than or equal to 5 V by connecting  $V_{REFHI}$  and  $V_{REFLO}$  to external reference voltages.  $V_{CCA}$  and AGND pins have to be connected to 5 V and analog ground, respectively.

The ADC module, shown in Figure 10, has the following features:

- Up to 8 analog inputs, ADCIN0–ADCIN7. The results from converting inputs ADCIN0–ADCIN7 are placed in one of the ADCFIFO results registers (see Table 10).
- Almost simultaneous measurement of two analog inputs, 1700 ns apart.
- Single conversion and continuous conversion modes.
- Conversion can be started by software, an internal event, and/or an external event.
- $V_{REFHI}$  and  $V_{REFLO}$  (high- and low-voltage) reference inputs.
- Two-level-deep digital result registers that contain the digital values of completed conversions.
- Two programmable ADC module control registers (see Table 10)
- Programmable clock prescaler
- Interrupt or polled operation

A/D overview (continued)

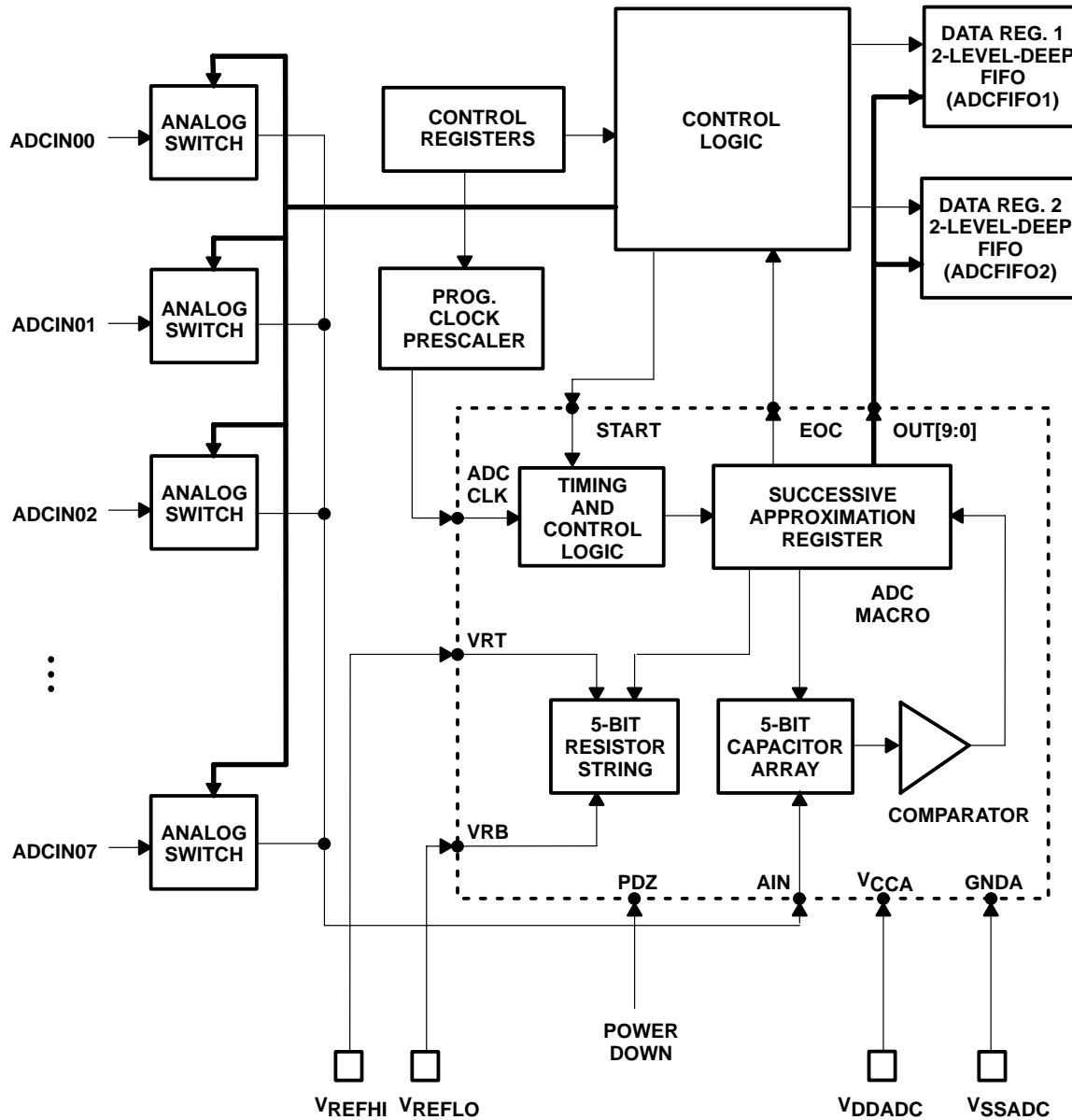


Figure 10. TMS320F243 Pseudo Dual Analog-to-Digital Converter (ADC) Module

Table 10. Addresses of ADC Registers

ADDRESS OFFSET	NAME	DESCRIPTION
7032h	ADCTRL1	ADC Control Register 1
7034h	ADCTRL2	ADC Control Register 2
7036h	ADCFIFO1	ADC 2-Level-Deep Data Register FIFO for pseudo ADC #1
7038h	ADCFIFO2	ADC 2-Level-Deep Data Register FIFO for pseudo ADC #2

## ***shadowed bits***

Many of the control register bits are described as “shadowed”. This means that changing the value of one of these bits does not take effect until the current conversion is complete.

## **serial peripheral interface (SPI) module**

The TMS320x243 devices include the four-pin serial peripheral interface (SPI) module. The SPI is a high-speed synchronous serial-I/O port that allows a serial bit stream of programmed length (one to sixteen bits) to be shifted into and out of the device at a programmable bit-transfer rate. Normally, the SPI is used for communications between the DSP controller and external peripherals or another processor. Typical applications include external I/O or peripheral expansion through devices such as shift registers, display drivers, and ADCs. Multidevice communications are supported by the master/slave operation of the SPI.

The SPI module features include the following:

- Four external pins:
  - SPISOMI: SPI slave-output/master-input pin, or general-purpose bidirectional I/O pin
  - SPISIMO: SPI slave-input/master-output pin, or general-purpose bidirectional I/O pin
  - SPISTE: SPI slave transmit-enable pin, or general-purpose bidirectional I/O pin
  - SPICLK: SPI serial-clock pin, or general-purpose bidirectional I/O pin
- Two operational modes: master and slave
- Baud rate: 125 different programmable rates/5 Mbps at 20 MHz CPUCLK
- Data word format: one to eight data bits
- Four clocking schemes controlled by clock polarity and clock phase bits include:
  - Falling edge without phase delay: SPICLK active high. SPI transmits data on the falling edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.
  - Falling edge with phase delay: SPICLK active high. SPI transmits data one half-cycle ahead of the falling edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
  - Rising edge without phase delay: SPICLK inactive low. SPI transmits data on the rising edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
  - Rising edge with phase delay: SPICLK inactive low. SPI transmits data one half-cycle ahead of the falling edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.
- Simultaneous receive and transmit operation (transmit function can be disabled in software)
- Transmitter and receiver operations are accomplished through either interrupt-driven or polled algorithms.
- Eleven SPI module control registers: Located in control register frame beginning at address 7040h.

NOTE: All registers in this module are 16-bit registers that are connected to the 16-bit peripheral bus. When a register is accessed, the register data is in the lower byte (7–0), and the upper byte (15–8) is read as zeros. Writing to the upper byte has no effect.

Figure 11 is a block diagram of the SPI in slave mode.



‡ The SPISTE pin is shown as being disabled, meaning that data cannot be transmitted in this mode. Note that SW1, SW2, and SW3 are closed in this configuration.

### Figure 11. Four-Pin Serial Peripheral Interface Module Block Diagram†

### serial communications interface (SCI) module

The TMS320x24x devices include a serial communications interface (SCI) module. The SCI module supports digital communications between the CPU and other asynchronous peripherals that use the standard non-return-to-zero (NRZ) format. The SCI receiver and transmitter are double-buffered, and each has its own separate enable and interrupt bits. Both can be operated independently or simultaneously in the full-duplex mode. To ensure data integrity, the SCI checks received data for break detection, parity, overrun, and framing errors. The speed of bit rate (baud) is programmable to over 65000 different speeds through a 16-bit baud-select register. Features of the SCI module include:

- Two external pins
  - SCITXD: SCI transmit-output pin or general-purpose bidirectional I/O pin
  - SCIRXD: SCI receive-input pin or general-purpose bidirectional I/O pin
- Baud rate programmable to 64K different rates
  - Up to 1250 Kbps at 20-MHz CPUCLK
- Data word format
  - One start bit
  - Data word length programmable from one to eight bits
  - Optional even/odd/no parity bit
  - One or two stop bits
- Four error-detection flags: parity, overrun, framing, and break detection
- Two wake-up multiprocessor modes: idle-line and address bit
- Half- or full-duplex operation
- Double-buffered receive and transmit functions
- Transmitter and receiver operations can be accomplished through interrupt-driven or polled algorithms with status flags.
  - Transmitter: TXRDY flag (transmitter-buffer register is ready to receive another character) and TX EMPTY flag (transmitter-shift register is empty)
  - Receiver: RXRDY flag (receiver-buffer register ready to receive another character), BRKDT flag (break condition occurred), and RX ERROR (monitoring four interrupt conditions)
- Separate enable bits for transmitter and receiver interrupts (except BRKDT)
- NRZ (non return-to-zero) format
- Twelve SCI module control registers located in the control register frame beginning at address 7050h

NOTE: All registers in this module are 8-bit registers that are connected to the 16-bit peripheral bus. When a register is accessed, the register data is in the lower byte (7–0), and the upper byte (15–8) is read as zeros. Writing to the upper byte has no effect.

Figure 12 shows the SCI module block diagram.

serial communications interface (SCI) module (continued)

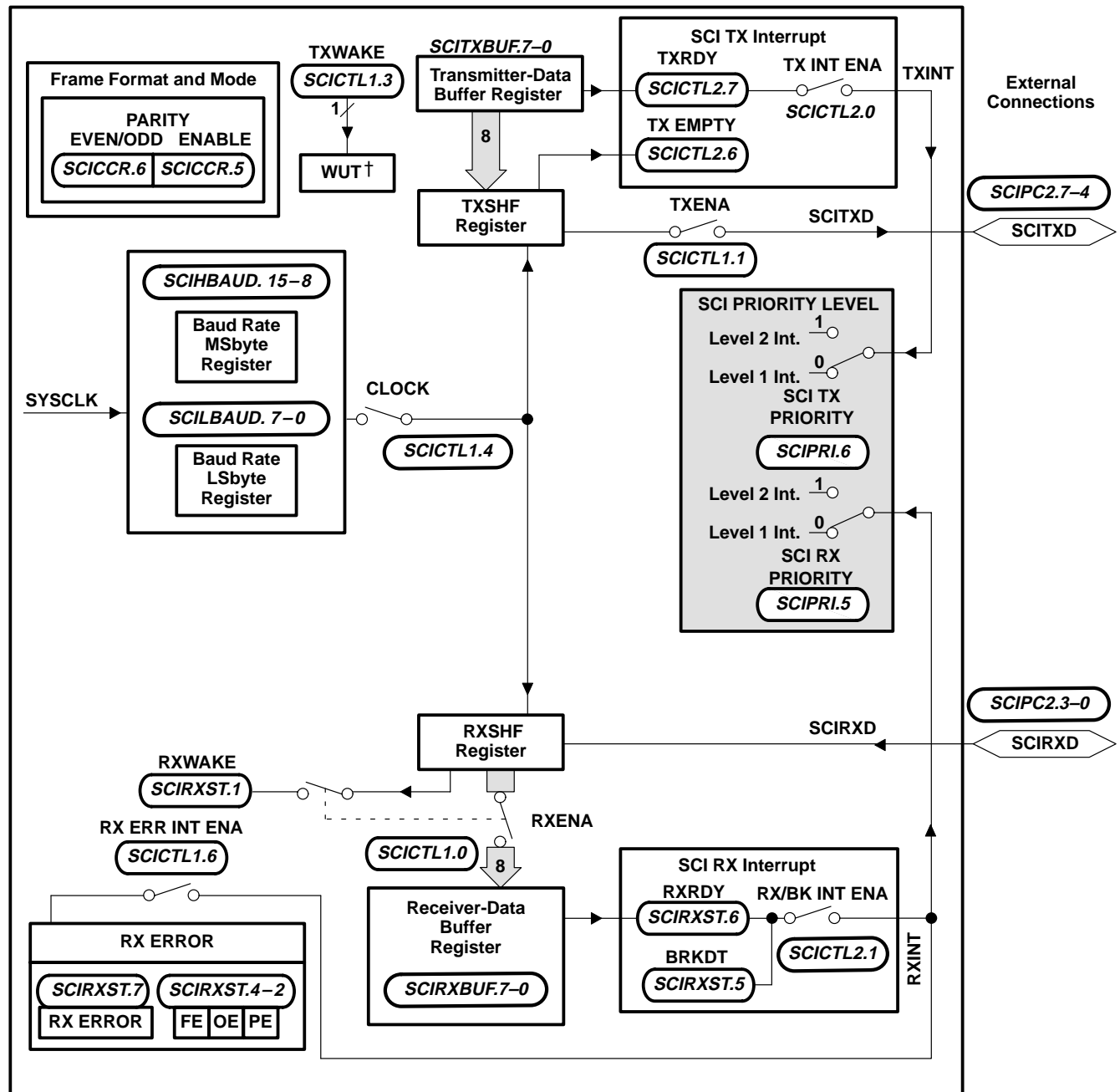


Figure 12. Serial Communications Interface (SCI) Module Block Diagram

controller area network (CAN) module

The CAN module is a 16-bit peripheral. The accesses are split into the control/status registers and accesses to the mailbox RAM.

CAN peripheral registers: The CPU can access the CAN peripheral registers only using 16-bit write accesses. The CAN peripheral always presents full 16-bit data to the CPU bus during read cycles.

# TMS320F243 DSP CONTROLLER

SPRS064 – DECEMBER 1997

## controller area network (CAN) module (continued)

The CAN module is a full-CAN controller designed as a 16-bit peripheral module for the RHEA Bus. The CAN module supports the following features:

- CAN specification 2.0B (active)
  - Standard data and remote frames
  - Extended data and remote frames
- 6 mailboxes for objects of 0 to 8-byte data length
  - 2 receive mailboxes, 2 transmit mailboxes
  - 2 configurable transmit/receive mailboxes
- Programmable global mask for objects 1 and 2 and one for object 3, 4
  - Configurable standard or extended message identifier
- Programmable bit rate
- Programmable interrupt scheme
- Readable error counters
- Self-Test Mode

The CAN module is able to operate in a loop-back mode, receiving its own transmitted message.

### CAN memory map

Table 11 and Table 12 show the register and mailbox locations in the CAN module.

**Table 11. Register Addresses†**

ADDRESS OFFSET	NAME	DESCRIPTION
00h	MDER	Mailbox Direction/Enable Register (bits 7 to 0)
01h	TCR	Transmission Control Register (bits 15 to 0)
02h	RCR	Receive Control Register (bits 15 to 0)
03h	MCR	Master Control Register (bits 13 to 6, 1, 0)
04h	BCR2	Bit Configuration Register 2 (bits 7 to 0)
05h	BCR1	Bit Configuration Register 1 (bits 10 to 0)
06h	ESR	Error Status Register (bits 8 to 0)
07h	GSR	Global Status Register (bits 5 to 0)
08h	CEC	Transmit and Receive Error Counters (bits 15 to 0)
09h	IFR	Interrupt Flag Register (bits 13 to 8, 6 to 0)
0Ah	IMR	Interrupt Mask Register (bits 15, 13 to 0)
0Bh	LAM0_H	Local Acceptance Mask Mailbox 0 and 1 (bits 31, 28 to 16)
0Ch	LAM0_L	Local Acceptance Mask Mailbox 0 and 1 (bits 15 to 0)
0Dh	LAM1_H	Local Acceptance Mask Mailbox 2 and 3 (bits 31, 28 to 16)
0Eh	LAM1_L	Local Acceptance Mask Mailbox 2 and 3 (bits 15 to 0)
0Fh	Reserved	Accesses assert the CAADDRx signal from the CAN peripheral (which asserts an Illegal Address error)

† All unimplemented register bits are read as zero, writes have no effect. Register bits are initialized to zero, unless otherwise stated in the definition.

### CAN memory map (continued)

The mailboxes are situated in one 24 x 32 RAM with 16-bit access. It can be written or read by the CPU or the CAN. The CAN write or read access as well as the CPU read access needs one clock cycle. The CPU write access needs two clock cycles. In these two clock cycles, the SCC performs a read-modify-write cycle and therefore inserts one wait state for the CPU.

The address bit 0 of the address bus used when accessing the RAM decides if the low (0) or the higher (1) 16-bit word of the 32-bit word is taken. The RAM location is determined by the upper bits 5 to 1 of the address bus.

The enable signals for the RAM (EZ and GZ) are always active low.

Table 12 shows the mailbox locations in RAM. One half-word has 16 bits.

**Table 12. Mailbox Addresses<sup>†</sup>**

ADDRESS OFFSET [5:0]	NAME	DESCRIPTION UPPER HALF-WORD ADDRESS BIT 0 = 1	DESCRIPTION LOWER HALF-WORD ADDRESS BIT 0 = 0
00h	MSGID0	Message ID for mailbox 0	Message ID for mailbox 0
02h	MSGCTRL0	Unused	RTR and DLC (bits 4 to 0)
04h	Datalow0	Databyte 0, Databyte 1 (DBO = 0)	Databyte 2, Databyte 3 (DBO = 0)
		Databyte 3, Databyte 2 (DBO = 1)	Databyte 1, Databyte 0 (DBO = 1)
06h	Datahigh0	Databyte 4, Databyte 5 (DBO = 0)	Databyte 6, Databyte 7 (DBO = 0)
		Databyte 7, Databyte 6 (DBO = 1)	Databyte 5, Databyte 4 (DBO = 1)
08h	MSGID1	Message ID for mailbox 1	Message ID for mailbox 1
0Ah	MSGCTRL1	Unused	RTR and DLC (bits 4 to 0)
0Ch	Datalow1	Databyte 0, Databyte 1 (DBO = 0)	Databyte 2, Databyte 3 (DBO = 0)
		Databyte 3, Databyte 2 (DBO = 1)	Databyte 1, Databyte 0 (DBO = 1)
0Eh	Datahigh1	Databyte 4, Databyte 5 (DBO = 0)	Databyte 6, Databyte 7 (DBO = 0)
...	...	...	...
28h	MSGID5	Message ID for mailbox 5	Message ID for mailbox 5
2Ah	MSGCTRL5	Unused	RTR and DLC (bits 4 to 0)
2Ch	Datalow5	Databyte 0, Databyte 1 (DBO = 0)	Databyte 2, Databyte 3 (DBO = 0)
		Databyte 3, Databyte 2 (DBO = 1)	Databyte 3, Databyte 2 (DBO = 1)
2Eh	Datahigh5	Databyte 4, Databyte 5 (DBO = 0)	Databyte 6, Databyte 7 (DBO = 0)
		Databyte 7, Databyte 6 (DBO = 1)	Databyte 5, Databyte 4 (DBO = 1)

<sup>†</sup> The DBO (Data Byte Order) bit is located in the MCR and is used to define the order in which the data bytes are stored in the mailbox when received and in which the data bytes are transmitted. Byte 0 is the first byte in the message and Byte 8 is the last one as shown in the figure of the CAN message.

### CAN interrupt logic

There are two interrupt requests from the CAN module to the Peripheral Interrupt Expansion (PIE) controller, the Mailbox interrupt and the Error interrupt. Both interrupts can assert either a high-priority request or a low-priority request to the CPU. The following events can initiate an interrupt:

- Transmission Interrupt  
A message was transmitted or received successfully—asserts the Mailbox interrupt.
- Abort Acknowledge Interrupt  
A send transmission was aborted—asserts the Error interrupt.

***CAN interrupt logic (continued)***

- Write Denied Interrupt  
The CPU tried to write to a mailbox but was not allowed to—asserts the Error interrupt.
- Wakeup Interrupt  
After wakeup, this interrupt is generated—asserts the Error interrupt, even when clocks are not running.
- Receive Message Lost Interrupt  
An old message was overwritten by a new one—asserts the Error interrupt.
- Bus-Off Interrupt  
The CAN module enters the bus-off state—asserts the Error interrupt.
- Error Passive Interrupt  
The CAN module enters the error passive mode—asserts the Error interrupt.
- Warning Level Interrupt  
One or both of the error counters is greater than or equal to 96—asserts the Error interrupt.

*CAN configuration mode*

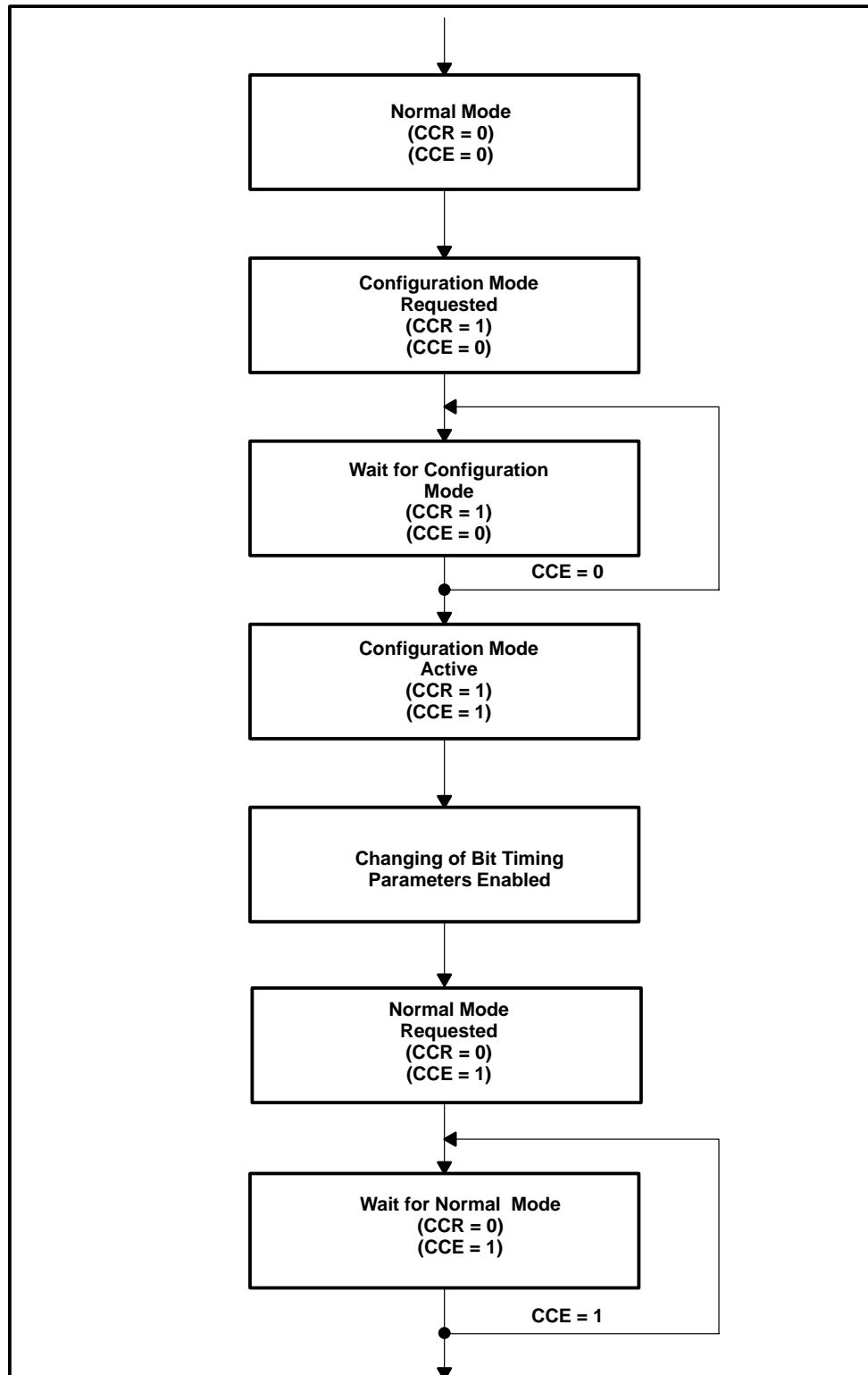


Figure 13. CAN Initialization

## **CAN configuration mode (continued)**

The CAN module must be initialized before activation. This is only possible if the module is in configuration mode. The configuration mode is set by programming CCR with “1”. Only if the status bit CCE confirms the request by getting “1”, the initialization can be performed. Afterwards, the bit configuration registers can be written. The module is activated again by programming the control bit CCR with zero. After a hardware reset, the configuration mode is active.

## **CAN power-down mode (PDM)**

The CAN peripheral's own low-power mode must be requested before a device low-power mode is entered by executing the IDLE instruction, if the device low-power mode is going to shut off the peripheral clocks.

Before the CPU enters its IDLE mode to enter the device low-power mode to potentially shut off ALL device clocks, it must first request a CAN peripheral power-down by writing an “1” to the PDR bit in MCR. If the module is transmitting a message when PDR is set, the transmission is continued until a successful transmission, a lost arbitration, or an error condition on the CAN bus line occurs. Then, the PDA is asserted. Therefore, the module causes no error condition on the CAN bus line. When the module is ready to enter power-down mode, the status bit PDA is set. The CPU must then poll the PDA bit in GSR, and only enter IDLE after PDA is set.

On exiting the power-down mode, the PDR flag in the MCR must be cleared by software or is cleared automatically if the WUBA bit in MCR is set and if there is any bus activity on the CAN bus line. When detecting a dominant signal on the CAN bus, the wakeup interrupt flag WUIF is asserted. The power-down mode is exited as soon as the clock is switched on. There is no internal filtering for the CAN bus line.

The automatic wakeup on bus activity can be enabled or disabled by setting the configuration bit WUBA. If there is any activity on the CAN bus line, the module begins its power up sequence. The module waits until detecting 11 consecutive recessive bits on the RX pin and goes to bus active afterwards. The first message, which initiates the bus activity, *cannot* be received.

When WUBA is enabled, the error interrupt WUIF is asserted automatically to the PIE controller, which handles it as a wakeup interrupt and restart the device clocks if they are stopped.

After leaving the sleep mode with a wakeup, the PDR and PDA are cleared. The CAN error counters remain unchanged.

## **watchdog (WD) timer module**

The TMS320x24x device includes a watchdog (WD) timer module. The WD function of this module monitors software and hardware operation by generating a system reset if it is not periodically serviced by software by having the correct key written. The WD timer operates independently of the CPU and is always enabled. It does not need any CPU initialization to function. When a system reset occurs, the WD timer defaults to the fastest WD timer rate available (6.55 ms for a 39062.5-Hz WDCLK signal). As soon as reset is released internally, the CPU starts executing code, and the WD timer begins incrementing. This means that, to avoid a premature reset, WD setup should occur early in the power-up sequence. See Figure 14 for a block diagram of the WD module. The WD module features include the following:



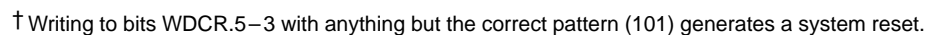
---

**watchdog (WD) timer module (continued)**

- WD Timer
  - Seven different WD overflow rates ranging from 6.55 ms to 1 s
  - A WD-reset key (WDKEY) register that clears the WD counter when a correct value is written, and generates a system reset if an incorrect value is written to the register
  - WD check bits that initiate a system reset if an incorrect value is written to the WD control register (WDCR)
- Automatic activation of the WD timer, once system reset is released
  - Three WD control registers located in control register frame beginning at address 7020h.

NOTE: All registers in this module are 8-bit registers. When a register is accessed, the register data is in the lower byte, the upper byte is read as zeros. Writing to the upper byte has no effect.

Figure 14 shows the WD block diagram. Table 13 shows the different WD overflow (timeout) selections.



**Figure 14. Block Diagram of the WD Module**

**watchdog (WD) timer module (continued)**

**Table 13. WD Overflow (Timeout) Selections**

WD PRESCALE SELECT BITS			WDCLK DIVIDER	39.0625-kHz WDCLK†	
WDPS2	WDPS1	WDPS0		FREQUENCY (Hz)	MINIMUM OVERFLOW (ms)
0	0	X‡	1	152.59	6.55
0	1	0	2	76.29	13.11
0	1	1	4	38.15	26.21
1	0	0	8	19.07	52.43
1	0	1	16	9.54	104.86
1	1	0	32	4.77	209.72
1	1	1	64	2.38	419.43

† Generated by 5.000000-MHz crystal

‡ X = Don't care

**scan-based emulation**

TMS320x24x devices use scan-based emulation for code- and hardware-development support. Serial scan interface is provided by the test-access port. Scan-based emulation allows the emulator to control the processor in the system without the use of intrusive cables to the full pinout of the device.

**TMS320x24x instruction set**

The 'x24x microprocessor implements a comprehensive instruction set that supports both numeric-intensive signal processing operations and general-purpose applications, such as multiprocessing and high-speed control. Source code for the 'C1x and 'C2x DSPs is upwardly compatible with the 'x243 devices.

For maximum throughput, the next instruction is prefetched while the current one is being executed. Because the same data lines are used to communicate to external data, program, or I/O space, the number of cycles an instruction requires to execute varies, depending upon whether the next data operand fetch is from internal or external memory. Highest throughput is achieved by maintaining data memory on chip and using either internal or fast external program memory.

**addressing modes**

The TMS320x24x instruction set provides four basic memory-addressing modes: direct, indirect, immediate, and register.

In direct addressing, the instruction word contains the lower seven bits of the data memory address. This field is concatenated with the nine bits of the data memory page pointer (DP) to form the 16-bit data memory address. Therefore, in the direct-addressing mode, data memory is paged effectively with a total of 512 pages, each page containing 128 words.

Indirect addressing accesses data memory through the auxiliary registers. In this addressing mode, the address of the instruction operand is contained in the currently selected auxiliary register. Eight auxiliary registers (AR0–AR7) provide flexible and powerful indirect addressing. To select a specific auxiliary register, the auxiliary register pointer (ARP) is loaded with a value from 0 to 7 for AR0 through AR7, respectively.

There are seven types of indirect addressing: autoincrement or autodecrement, postindexing by adding or subtracting the contents of AR0, single-indirect addressing with no increment or decrement, and bit-reversed addressing [used in Fast Fourier Transforms (FFTs)] with increment or decrement. All operations are performed on the current auxiliary register in the same cycle as the original instruction, following which the current auxiliary register and ARP can be modified.

## addressing modes (continued)

In immediate addressing, the actual operand data is provided in a portion of the instruction word or words. There are two types of immediate addressing: long and short. In short-immediate addressing, the data is contained in a portion of the bits in a single-word instruction. In long-immediate addressing, the data is contained in the second word of a two-word instruction. The immediate-addressing mode is useful for data that does not need to be stored or used more than once during the course of program execution (for example, initialization values or constants).

The register-addressing mode uses operands in CPU registers either explicitly, such as with a direct reference to a specific register, or implicitly, with instructions that intrinsically reference certain registers. In either case, operand reference is simplified because 16-bit values can be used without specifying a full 16-bit operand address or immediate value.

## repeat feature

The repeat function can be used with instructions (as defined in Table 15) such as multiply/accumulates (MAC and MACD), block moves (BLDD and BLPD), I/O transfers (IN/OUT), and table read/writes (TBLR/TBLW). These instructions, although normally multicycle, are pipelined when the repeat feature is used, and they effectively become single-cycle instructions. For example, the table-read instruction can take three or more cycles to execute, but when the instruction is repeated, a table location can be read every cycle.

The repeat counter (RPTC) is loaded with the addressed data memory location if direct or indirect addressing is used, and with an 8-bit immediate value if short-immediate addressing is used. The RPTC register is loaded by the RPT instruction. This results in a maximum of  $N + 1$  executions of a given instruction. RPTC is cleared by reset. Once a repeat instruction (RPT) is decoded, all interrupts, including NMI (but excluding reset), are masked until the completion of the repeat loop.

## instruction set summary

This section summarizes the operation codes (opcodes) of the instruction set for the 'x24x digital signal processors. This instruction set is a superset of the 'C1x and 'C2x instruction sets. The instructions are arranged according to function and are alphabetized by mnemonic within each category. The symbols in Table 14 are used in the instruction set summary table (Table 15). The TI 'C2xx assembler accepts 'C2x instructions.

The number of words that an instruction occupies in program memory is specified in column 3 of Table 16. Several instructions specify two values separated by a slash mark (/) for the number of words. In these cases, different forms of the instruction occupy a different number of words. For example, the ADD instruction occupies one word when the operand is a short-immediate value or two words if the operand is a long-immediate value.

The number of cycles that an instruction requires to execute is also in column 3 of Table 16. All instructions are assumed to be executed from internal program memory (RAM) and internal data dual-access memory. The cycle timings are for single-instruction execution, not for repeat mode.

**instruction set summary (continued)**

**Table 14. TMS320x24x Opcode Symbols**

SYMBOL	DESCRIPTION
A	Address
ACC	Accumulator
ACCB	Accumulator buffer
ARx	Auxiliary register value (0–7)
BITx	4-bit field that specifies which bit to test for the BIT instruction
BMAR	Block-move address register
DBMR	Dynamic bit-manipulation register
I	Addressing-mode bit
II...II	Immediate operand value
INTM	Interrupt-mode flag bit
INTR#	Interrupt vector number
K	Constant
PREG	Product register
PROG	Program memory
RPTC	Repeat counter
SHF, SHFT	3/4-bit shift value
TC	Test-control bit
T P	Two bits used by the conditional execution instructions to represent the conditions TC, NTC, and BIO. <div> <div>T P</div> <div>Meaning</div> </div> <div> <div>0 0</div> <div><math>\overline{\text{BIO}}</math> low</div> </div> <div> <div>0 1</div> <div>TC = 1</div> </div> <div> <div>1 0</div> <div>TC = 0</div> </div> <div> <div>1 1</div> <div>None of the above conditions</div> </div>
TREGn	Temporary register n (n = 0, 1, or 2)
Z L V C	4-bit field representing the following conditions: <div> <div>Z:</div> <div>ACC = 0</div> </div> <div> <div>L:</div> <div>ACC &lt; 0</div> </div> <div> <div>V:</div> <div>Overflow</div> </div> <div> <div>C:</div> <div>Carry</div> </div> <p>A conditional instruction contains two of these 4-bit fields. The 4-LSB field of the instruction is a 4-bit mask field. A 1 in the corresponding mask bit indicates that the condition is being tested. The second 4-bit field (bits 4–7) indicates the state of the conditions designated by the mask bits as being tested. For example, to test for <math>\text{ACC} \geq 0</math>, the Z and L fields are set while the V and C fields are not set. The next 4-bit field contains the state of the conditions to test. The Z field is set to indicate testing of the condition <math>\text{ACC} = 0</math>, and the L field is reset to indicate testing of the condition <math>\text{ACC} \geq 0</math>. The conditions possible with these 8 bits are shown in the BCND and CC instructions. To determine if the conditions are met, the 4-LSB bit mask is ANDed with the conditions. If any bits are set, the conditions are met.</p>

# TMS320F243 DSP CONTROLLER

SPRS064 – DECEMBER 1997

## instruction set summary (continued)

Table 15. TMS320x24x Instruction Set Summary

'x24x MNEMONIC	DESCRIPTION	WORDS/ CYCLES	OPCODE			
			MSB		LSB	
ABS	Absolute value of accumulator	1/1	1011	1110	0000	0000
ADD	Add to accumulator with shift	1/1	0010	SHFT	IADD	RESS
	Add to high accumulator	1/1	0110	0001	IADD	RESS
	Add to accumulator short immediate	1/1	1011	1000	KKKK	KKKK
	Add to accumulator long immediate with shift	2/2	1011	1111	1001	SHFT
ADDC	Add to accumulator with carry	1/1	0110	0000	IADD	RESS
ADDS	Add to low accumulator with sign extension suppressed	1/1	0110	0010	IADD	RESS
ADDT	Add to accumulator with shift specified by T register	1/1	0110	0011	IADD	RESS
ADRK	Add to auxiliary register short immediate	1/1	0111	1000	KKKK	KKKK
AND	AND with accumulator	1/1	0110	1110	IADD	RESS
	AND immediate with accumulator with shift	2/2	1011	1111	1011	SHFT
	AND immediate with accumulator with shift of 16	2/2	1011	1110	1000	0001
APAC	Add P register to accumulator	1/1	1011	1110	0000	0100
B	Branch unconditionally	2/4	0111	1001	IADD	RESS
BACC	Branch to address specified by accumulator	1/4	1011	1110	0010	0000
BANZ	Branch on auxiliary register not zero	2/4/2	0111	1011	IADD	RESS
BCND	Branch if TC bit $\neq$ 0	2/4/2	1110	0001	0000	0000
	Branch if TC bit = 0	2/4/2	1110	0010	0000	0000
	Branch on carry	2/4/2	1110	0011	0001	0001
	Branch if accumulator $\geq$ 0	2/4/2	1110	0011	1000	1100
	Branch if accumulator $>$ 0	2/4/2	1110	0011	0000	0100
	Branch on I/O status low	2/4/3	1110	0000	0000	0000
	Branch if accumulator $\leq$ 0	2/4/2	1110	0011	1100	1100
	Branch if accumulator $<$ 0	2/4/2	1110	0011	0100	0100
	Branch on no carry	2/4/2	1110	0011	0000	0001
	Branch if no overflow	2/4/2	1110	0011	0000	0010



instruction set summary (continued)

**Table 15. TMS320x24x Instruction Set Summary (Continued)**

'x24x MNEMONIC	DESCRIPTION	WORDS/ CYCLES	OPCODE			
			MSB			LSB
BCND	Branch if accumulator $\neq$ 0	2/4/2	1110	0011	0000	1000 Branch Address
	Branch on overflow	2/4/2	1110	0011	0010	0010 Branch Address
	Branch if accumulator = 0	2/4/2	1110	0011	1000	1000 Branch Address
BIT	Test bit	1/1	0100	BITx	IADD	RESS
BITT	Test bit specified by TREG	1/1	0110	1111	IADD	RESS
BLDD <sup>†</sup>	Block move from data memory to data memory source immediate	2/3	1010	1000	IADD	RESS Branch Address
	Block move from data memory to data memory destination immediate	2/3	1010	1001	IADD	RESS Branch Address
BLPD	Block move from program memory to data memory	2/3	1010	0101	IADD	RESS Branch Address
CALA	Call subroutine indirect	1/4	1011	1110	0011	0000
CALL	Call subroutine	2/4	0111	1010	IADD	RESS Routine Address
CC	Conditional call subroutine	2/4/2	1110	10TP	ZLVC	ZLVC Routine Address
CLRC	Configure block as data memory	1/1	1011	1110	0100	0100
	Enable interrupt	1/1	1011	1110	0100	0000
	Reset carry bit	1/1	1011	1110	0100	1110
	Reset overflow mode	1/1	1011	1110	0100	0010
	Reset sign-extension mode	1/1	1011	1110	0100	0110
	Reset test/control flag	1/1	1011	1110	0100	1010
	Reset external flag	1/1	1011	1110	0100	1100
CMPL	Complement accumulator	1/1	1011	1110	0000	0001
CMPR	Compare auxiliary register with auxiliary register AR0	1/1	1011	1111	0100	01CM
DMOV	Data move in data memory	1/1	0111	0111	IADD	RESS
IDLE	Idle until interrupt	1/1	1011	1110	0010	0010
IN	Input data from port	2/2	1010	1111	IADD	RESS
			16BIT	I/O	PORT	ADRS
INTR	Software-interrupt	1/4	1011	1110	011K	KKKK
LACC	Load accumulator with shift	1/1	0001	SHFT	IADD	RESS
	Load accumulator long immediate with shift	2/2	1011	1111	1000	SHFT 16-Bit Constant
	Zero low accumulator and load high accumulator	1/1	0110	1010	IADD	RESS

<sup>†</sup> In 'x24x devices, the BLDD instruction does not work with memory-mapped registers IMR, IFR, and GREG.

# TMS320F243

## DSP CONTROLLER

SPRS064 – DECEMBER 1997

### instruction set summary (continued)

Table 15. TMS320x24x Instruction Set Summary (Continued)

'x24x MNEMONIC	DESCRIPTION	WORDS/ CYCLES	OPCODE			
			MSB		LSB	
LACL	Load accumulator immediate short	1/1	1011	1001	KKKK	KKKK
	Zero accumulator	1/1	1011	1001	0000	0000
	Zero low accumulator and load high accumulator	1/1	0110	1010	IADD	RESS
	Zero low accumulator and load low accumulator with no sign extension	1/1	0110	1001	IADD	RESS
LACT	Load accumulator with shift specified by T register	1/1	0110	1011	IADD	RESS
LAR	Load auxiliary register	1/2	0000	0ARx	IADD	RESS
	Load auxiliary register short immediate	1/2	1011	0ARx	KKKK	KKKK
	Load auxiliary register long immediate	2/2	1011	1111	0000	1ARx 16-Bit Constant
LDP	Load data-memory page pointer	1/2	0000	1101	IADD	RESS
	Load data-memory page pointer immediate	1/2	1011	110P	AGEP	OINT
LPH	Load high-P register	1/1	0111	0101	IADD	RESS
LST	Load status register ST0	1/2	0000	1110	IADD	RESS
	Load status register ST1	1/2	0000	1111	IADD	RESS
LT	Load TREG	1/1	0111	0011	IADD	RESS
LTA	Load TREG and accumulate previous product	1/1	0111	0000	IADD	RESS
LTD	Load TREG, accumulate previous product, and move data	1/1	0111	0010	IADD	RESS
LTP	Load TREG and store P register in accumulator	1/1	0111	0001	IADD	RESS
LTS	Load TREG and subtract previous product	1/1	0111	0100	IADD	RESS
MAC	Multiply and accumulate	2/3	1010	0010	IADD	RESS
			16-Bit Constant			
MACD	Multiply and accumulate with data move	2/3	1010	0011	IADD	RESS
			16-Bit Constant			
MAR	Load auxiliary register pointer	1/1	1000	1011	1000	1ARx
	Modify auxiliary register	1/1	1000	1011	IADD	RESS
MPY	Multiply (with TREG, store product in P register)	1/1	0101	0100	IADD	RESS
	Multiply immediate	1/1	110C	KKKK	KKKK	KKKK
MPYA	Multiply and accumulate previous product	1/1	0101	0000	IADD	RESS
MPYS	Multiply and subtract previous product	1/1	0101	0001	IADD	RESS
MPYU	Multiply unsigned	1/1	0101	0101	IADD	RESS
NEG	Negate accumulator	1/1	1011	1110	0000	0010
NMI	Nonmaskable interrupt	1/4	1011	1110	0101	0010
NOP	No operation	1/1	1000	1011	0000	0000
NORM	Normalize contents of accumulator	1/1	1010	0000	IADD	RESS
OR	OR with accumulator	1/1	0110	1101	IADD	RESS
	OR immediate with accumulator with shift	2/2	1011	1111	1100	SHFT 16-Bit Constant
	OR immediate with accumulator with shift of 16	2/2	1011	1110	1000	0010 16-Bit Constant
OUT	Output data to port	2/3	0000	1100	IADD	RESS
PAC	Load accumulator with P register	1/1	1011	1110	0000	0011





**instruction set summary (continued)**

**Table 15. TMS320x24x Instruction Set Summary (Continued)**

'x24x MNEMONIC	DESCRIPTION	WORDS/ CYCLES	OPCODE			
			MSB		LSB	
POP	Pop top of stack to low accumulator	1/1	1011	1110	0011	0010
POPD	Pop top of stack to data memory	1/1	1000	1010	IADD	RESS
PSHD	Push data-memory value onto stack	1/1	0111	0110	IADD	RESS
PUSH	Push low accumulator onto stack	1/1	1011	1110	0011	1100
RET	Return from subroutine	1/4	1110	1111	0000	0000
RETC	Conditional return from subroutine	1/4/2	1110	11TP	ZLVC	ZLVC
ROL	Rotate accumulator left	1/1	1011	1110	0000	1100
ROR	Rotate accumulator right	1/1	1011	1110	0000	1101
RPT	Repeat instruction as specified by data-memory value	1/1	0000	1011	IADD	RESS
	Repeat instruction as specified by immediate value	1/1	1011	1011	KKKK	KKKK
SACH	Store high accumulator with shift	1/1	1001	1SHF	IADD	RESS
SACL	Store low accumulator with shift	1/1	1001	0SHF	IADD	RESS
SAR	Store auxiliary register	1/1	1000	0ARx	IADD	RESS
SBRK	Subtract from auxiliary register short immediate	1/1	0111	1100	KKKK	KKKK
SETC	Set carry bit	1/1	1011	1110	0100	1111
	Configure block as program memory	1/1	1011	1110	0100	0101
	Disable interrupt	1/1	1011	1110	0100	0001
	Set overflow mode	1/1	1011	1110	0100	0011
	Set test/control flag	1/1	1011	1110	0100	1011
	Set external flag XF	1/1	1011	1110	0100	1101
	Set sign-extension mode	1/1	1011	1110	0100	0111
SFL	Shift accumulator left	1/1	1011	1110	0000	1001
SFR	Shift accumulator right	1/1	1011	1110	0000	1010
SPAC	Subtract P register from accumulator	1/1	1011	1110	0000	0101
SPH	Store high-P register	1/1	1000	1101	IADD	RESS
SPL	Store low-P register	1/1	1000	1100	IADD	RESS
SPM	Set P register output shift mode	1/1	1011	1111	IADD	RESS
SQRA	Square and accumulate	1/1	0101	0010	IADD	RESS
SQRS	Square and subtract previous product from accumulator	1/1	0101	0011	IADD	RESS
SST	Store status register ST0	1/1	1000	1110	IADD	RESS
	Store status register ST1	1/1	1000	1111	IADD	RESS
SPLK	Store long immediate to data memory	2/2	1010	1110	IADD	RESS 16-Bit Constant
SUB	Subtract from accumulator long immediate with shift	2/2	1011	1111	1010	SHFT 16-Bit Constant
	Subtract from accumulator with shift	1/1	0011	SHFT	IADD	RESS
	Subtract from high accumulator	1/1	0110	0101	IADD	RESS
	Subtract from accumulator short immediate	1/1	1011	1010	KKKK	KKKK

### instruction set summary (continued)

**Table 15. TMS320x24x Instruction Set Summary (Continued)**

'x24x MNEMONIC	DESCRIPTION	WORDS/ CYCLES	OPCODE			
			MSB			LSB
SUBB	Subtract from accumulator with borrow	1/1	0110	0100	IADD	RESS
SUBC	Conditional subtract	1/1	0000	1010	IADD	RESS
SUBS	Subtract from low accumulator with sign extension suppressed	1/1	0110	0110	IADD	RESS
SUBT	Subtract from accumulator with shift specified by TREG	1/1	0110	0111	IADD	RESS
TBLR	Table read	1/3	1010	0110	IADD	RESS
TBLW	Table write	1/3	1010	0111	IADD	RESS
TRAP	Software interrupt	1/4	1011	1110	0101	0001
XOR	Exclusive-OR with accumulator	1/1	0110	1100	IADD	RESS
	Exclusive-OR immediate with accumulator with shift	2/2	1011	1111	1101	SHFT
	Exclusive-OR immediate with accumulator with shift of 16	2/2	1011	1110	1000	0011
ZALR	Zero low accumulator and load high accumulator with rounding	1/1	0110	1000	IADD	RESS

### development support

Texas Instruments offers an extensive line of development tools for the 'x24x generation of DSPs, including tools to evaluate the performance of the processors, generate code, develop algorithm implementations, and fully integrate and debug software and hardware modules.

The following products support development of 'x24x-based applications:

#### Software Development Tools:

- Assembler/linker
- Simulator
- Optimizing ANSI C compiler
- Application algorithms
- C/Assembly debugger and code profiler

#### Hardware Development Tools:

Emulator XDS510 (supports 'x24x multiprocessor system debug)

The *TMS320 Family Development Support Reference Guide* (literature number SPRU011D) contains information about development support products for all TMS320 family member devices, including documentation. Refer to this document for further information about TMS320 documentation or any other TMS320 support products from Texas Instruments. There is also an additional document, the *TMS320 Third Party Support Reference Guide* (literature number SPRU052), which contains information about TMS320-related products from other companies in the industry. To receive copies of TMS320 literature, contact the Literature Response Center at 800/477-8924.

See Table 16 and Table 17 for complete listings of development support tools for the 'x24x. For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

**development support (continued)**

**Table 16. Development Support Tools**

DEVELOPMENT TOOL	PLATFORM	PART NUMBER
<b>Software</b>		
Compiler/Assembler/Linker	SPARC™	TMDS3242555-08
Compiler/Assembler/Linker	PC-DOS™	TMDS3242855-02
Assembler/Linker	PC-DOS, OS/2™	TMDS3242850-02
'C2xx Simulator	PC-DOS, WIN™	TMDX324x851-02
'C2xx Simulator	SPARC	TMDX324x551-09
Digital Filter Design Package	PC-DOS	DFDP
'C2xx Debugger/Emulation Software	PC-DOS, OS/2, WIN	TMDX324012xx
'C2xx Debugger/Emulation Software	SPARC	TMDX324062xx
<b>Hardware</b>		
XDS510 XL Emulator	PC-DOS, OS/2	TMDS00510
XDS510 WS Emulator	SPARC	TMDS00510WS

**Table 17. TMS320x24x-Specific Development Tools**

DEVELOPMENT TOOL	PLATFORM	PART NUMBER
<b>Hardware</b>		
132-pin PQFP Target Cable Kit	—	TMDS30802132PQ
C24x EVM	PC	TMDX326P124x

**device and development support tool nomenclature**

To designate the stages in the product development cycle, Texas Instruments assigns prefixes to the part numbers of all TMS320 devices and support tools. Each TMS320 member has one of three prefixes: TMX, TMP, or TMS. Texas Instruments recommends two of three possible prefix designators for its support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS). This development flow is defined below.

Device development evolutionary flow:

- TMX** Experimental device that is not necessarily representative of the final device's electrical specifications
- TMP** Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification
- TMS** Fully-qualified production device

SPARC is a trademark of SPARC International, Inc.  
PC-DOS and OS/2 are trademarks of International Business Machines Corp.  
WIN is a trademark of Microsoft Corp.

# TMS320F243 DSP CONTROLLER

SPRS064 – DECEMBER 1997

## device and development support tool nomenclature (continued)

Support tool development evolutionary flow:

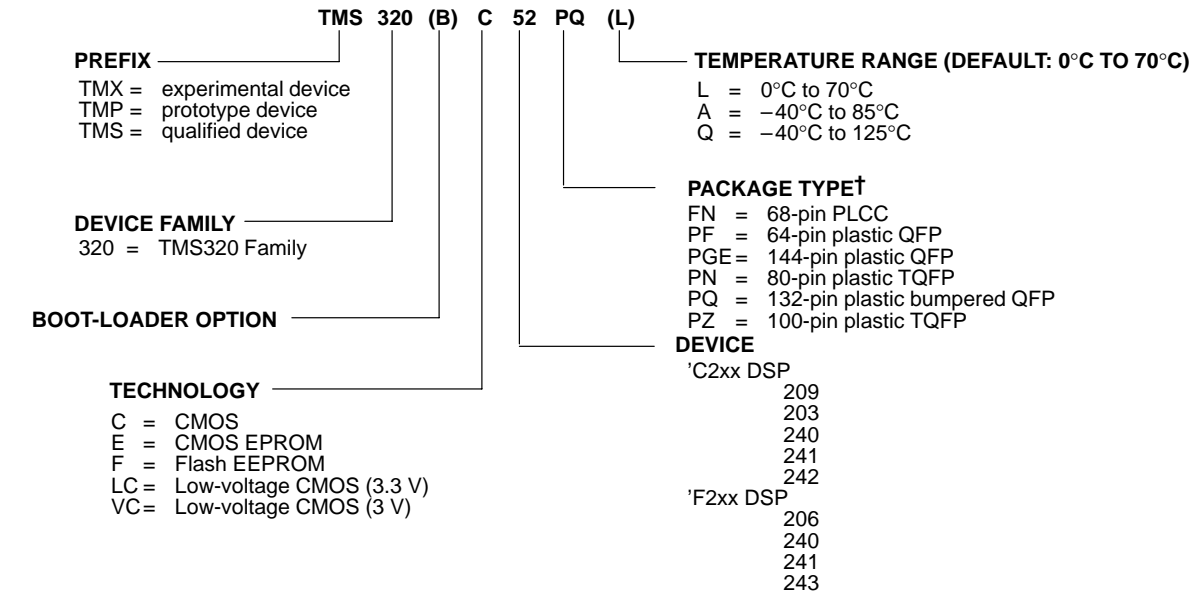
- TMDX** Development support product that has not completed TI's internal qualification testing
- TMDS** Fully qualified development support product

TMX and TMP devices and TMDX development support tools are shipped against the following disclaimer:  
"Developmental product is intended for internal evaluation purposes."

TMS devices and TMDS development support tools have been fully characterized, and the quality and reliability of the device have been fully demonstrated. TI's standard warranty applies.

Predictions show that prototype devices (TMX or TMP) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, PN, PQ, and PZ) and temperature range (for example, L). Figure 15 provides a legend for reading the complete device name for any TMS320x2xx family member.



† PLCC = Plastic J-Leaded Chip Carrier  
QFP = Quad Flatpack  
TQFP = Thin Quad Flat Package

Figure 15. TMS320 Device Nomenclature

---

## **documentation support**

Extensive documentation supports all of the TMS320 family generations of devices from product announcement through applications development. The types of documentation available include: data sheets, such as this document, with design specifications; complete user's guides for all devices and development support tools; and hardware and software applications.

For general background information on DSPs and TI devices, see the three-volume publication *Digital Signal Processing Applications With the TMS320 Family* (literature numbers SPRA012, SPRA016, and SPRA017). Also available is the *Calculation of TMS320C2xx Power Dissipation* application report (literature number SPRA088).

A series of DSP textbooks is published by Prentice-Hall and John Wiley & Sons to support digital signal processing research and education. The TMS320 newsletter, *Details on Signal Processing*, is published quarterly and distributed to update TMS320 customers on product information. The TMS320 DSP bulletin board service (BBS) provides access to a wealth of information pertaining to the TMS320 family, including documentation and source and object code for many DSP algorithms and utilities. The BBS can be reached at 281/274-2323.

Updated information on the TMS320 DSP controllers can be found on the worldwide web at: <http://www.ti.com/dsps>.



# TMS320F243

## DSP CONTROLLER

SPRS064 – DECEMBER 1997

### absolute maximum ratings over operating free-air temperature range (unless otherwise noted)<sup>†</sup>

Supply voltage range, $V_{DD}$ <sup>‡</sup>	–0.3 V to 7 V
Input voltage range	–0.3 V to 7 V
Output voltage range	–0.3 V to 7 V
Operating free-air temperature range, $T_A$ : A version	–40°C to 85°C
Q version	–40°C to 125°C
Storage temperature range, $T_{stg}$	–55°C to 150°C

<sup>†</sup> Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

<sup>‡</sup> All voltage values are with respect to  $V_{SS}$ .

### recommended operating conditions

		MIN	NOM	MAX	UNIT
$V_{DD}$	Supply voltage	4.5	5	5.5	V
$V_{SS}$	Supply ground		0		V
$V_{IH}$	High-level input voltage	XTAL1/CLKIN	3	$V_{DD} + 0.3$	V
		$\overline{RS}$	2	$V_{DD} + 0.3$	
		All other inputs	2	$V_{DD} + 0.3$	
$V_{IL}$	Low-level input voltage	XTAL1/CLKIN	–0.3	0.7	V
		$\overline{RS}$	–0.3	0.8	
		All other inputs	–0.3	0.8	
$I_{OH}$	High-level output current, $V_{OH} = 2.4 V^{\S}$	$\overline{RS}$		–19	mA
		IOPA[0:3]		–17	
		All other inputs		–23	
$I_{OL}$	Low-level output current, $V_{OL} = 0.6 V^{\S}$	$\overline{RS}$		10	mA
		IOPA[0:3]		7	
		All other inputs		13	
$T_A$	Operating free-air temperature	A version	–40	85	°C
		Q version	–40	125	
$\theta_{JA}$	Thermal resistance, junction-to-ambient			48	°C/W
$\theta_{JC}$	Thermal resistance, junction-to-case			8.5	°C/W

<sup>§</sup> Values derived from characterization data and not tested.

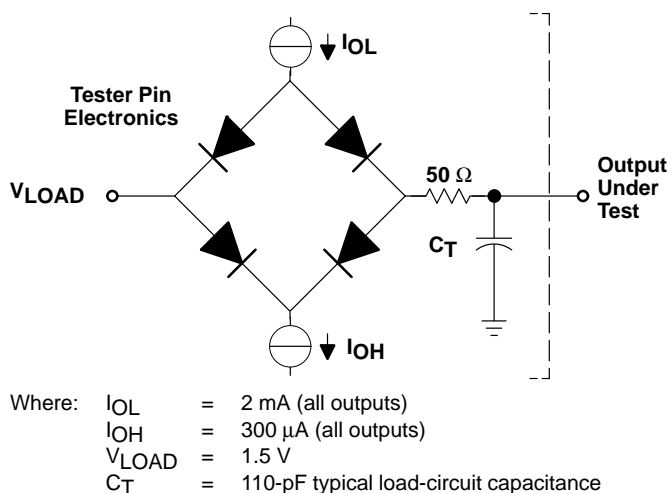
PRODUCT PREVIEW



**electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)**

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$V_{OH}$	High-level output voltage	5-V operation, $I_{OH} = \text{MAX}$	2.4			V
$V_{OL}$	Low-level output voltage	5-V operation, $I_{OL} = \text{MAX}$			0.6	V
$I_I$	Input current ( $V_I = V_{SS}$ or $V_{DD}$ )	$\overline{\text{TRST}}$ and PMTMODE pins with internal pulldown	-10		500	$\mu\text{A}$
		EMU0, EMU1, TMS, TCK, and TDI with internal pullup	-500		10	
		All other input-only pins	-10		10	
$I_{OZ}$	Output current, high-impedance state (off-state)	$V_O = V_{DD}$ or 0 V	-5		5	$\mu\text{A}$
$I_{DD}$	Supply current, operating mode	5-V operation, $t_c(\text{CO}) = 50 \text{ ns}$		80		mA
	Supply current, Idle 1 low-power mode	5-V operation, $t_c(\text{CO}) = 50 \text{ ns}$		50		
	Supply current, Idle 2 low-power mode	5-V operation, $t_c(\text{CO}) = 50 \text{ ns}$		7		
	Supply current, PLL power-down mode	5-V operation, $t_c(\text{CO}) = 50 \text{ ns}$		1		
	Supply current, OSC power-down mode	5-V operation, $t_c(\text{CO}) = 50 \text{ ns}$		400		$\mu\text{A}$
$C_i$	Input capacitance			15		pF
$C_o$	Output capacitance			15		pF

## PARAMETER MEASUREMENT INFORMATION

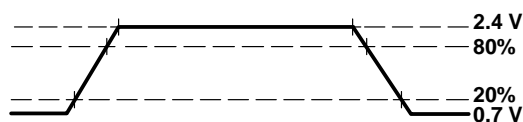


**Figure 16. Test Load Circuit**

## signal transition levels

The data in this section is shown for the 5-V version. Note that some of the signals use different reference voltages, see the recommended operating conditions table. TTL-output levels are driven to a minimum logic-high level of 2.4 V and to a maximum logic-low level of 0.7 V.

Figure 17 shows the TTL-level outputs.



**Figure 17. TTL-Level Outputs**

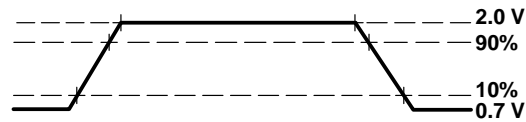
TTL-output transition times are specified as follows:

- For a *high-to-low transition*, the level at which the output is said to be no longer high is below 80% of the total voltage range and lower and the level at which the output is said to be low is 20% of the total voltage range and lower.
- For a *low-to-high transition*, the level at which the output is said to be no longer low is 20% of the total voltage range and higher and the level at which the output is said to be high is 80% of the total voltage range and higher.



## PARAMETER MEASUREMENT INFORMATION

Figure 18 shows the TTL-level inputs.



**Figure 18. TTL-Level Inputs**

TTL-compatible input transition times are specified as follows:

- For a *high-to-low transition* on an input signal, the level at which the input is said to be no longer high is 90% of the total voltage range and lower and the level at which the input is said to be low is 10% of the total voltage range and lower.
- For a *low-to-high transition* on an input signal, the level at which the input is said to be no longer low is 10% of the total voltage range and higher and the level at which the input is said to be high is 90% of the total voltage range and higher.

## PARAMETER MEASUREMENT INFORMATION

### timing parameter symbology

Timing parameter symbols used are created in accordance with JEDEC Standard 100-A. To shorten the symbols, some of the pin names and other related terminology have been abbreviated as follows:

A	A[15:0]	MS	Memory strobe pins $\overline{IS}$ , $\overline{DS}$ , or $\overline{PS}$
CI	XTAL1/CLKIN	R	READY
CO	CLKOUT	RD	Read cycle or $W/\overline{R}$
D	D[15:0]	RS	RESET pins RS or $\overline{RS}$
INT	NMI, XINT1, XINT2/IO	W	Write cycle or $\overline{WE}$

#### Lowercase subscripts and their meanings:

a	access time
c	cycle time (period)
d	delay time
f	fall time
h	hold time
r	rise time
su	setup time
t	transition time
v	valid time
w	pulse duration (width)

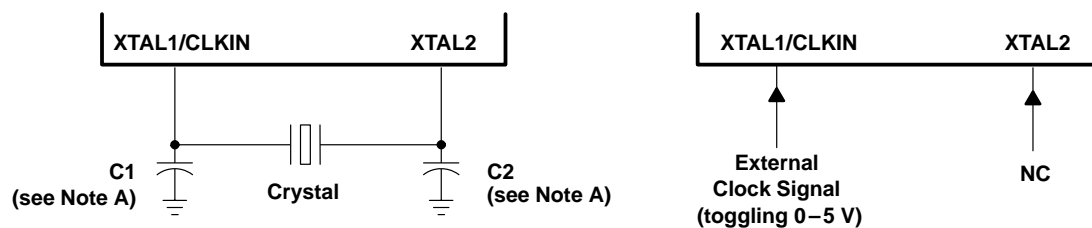
#### Letters and symbols and their meanings:

H	High
L	Low
V	Valid
X	Unknown, changing, or don't care level
Z	High impedance

### general notes on timing parameters

All output signals from the TMS320x24x devices (including CLKOUT) are derived from an internal clock such that all output transitions for a given half-cycle occur with a minimum of skewing relative to each other.

The signal combinations shown in the following timing diagrams may not necessarily represent actual cycles. For actual cycle examples, refer to the appropriate cycle description section of this data sheet.



NOTE A: For the values of C1 and C2, see the manufacturer's specification.

**Figure 19. Recommended Crystal/Clock Connection**

**switching characteristics over recommended operating conditions for clock [ $H = 0.5 t_{c(CO)}$ ]  
(see Note 1 and the recommended operating conditions table)**

PARAMETER		MIN	TYP	MAX	UNIT
$t_{c(CPU)}$	Cycle time, CPUCLK		$t_{c(CI)}$		ns
$t_{c(CO)}$	Cycle time, CLKOUT		$t_{c(CI)}$	†	ns
$t_{d(CIH-CO)}$	Delay time, XTAL1/CLKIN high to CLKOUT high/low	3	18	25	ns
$t_f(CO)$	Fall time, CLKOUT		5		ns
$t_r(CO)$	Rise time, CLKOUT		5		ns
$t_w(COL)$	Pulse duration, CLKOUT low	H–10	H–5	H–3	ns
$t_w(COH)$	Pulse duration, CLKOUT high	H+3	H+5	H+10	ns

† This device utilizes a fully static design and, therefore, can operate with input clock cycle time [ $t_{c(CI)}$ ] approaching infinity. The device is characterized at frequencies approaching 0 Hz, but is tested at  $f_{clk} = 6.7$  MHz to meet device test time requirements.

NOTE 1: Timings assume CLKOUT is set to output CPUCLK. CLKOUT is initialized to CPUCLK by power-on reset.

**timing requirements over recommended operating conditions for clock**

		MIN	MAX	UNIT
$t_{c(CI)}$	Cycle time, XTAL1/CLKIN	50	†	ns
$t_f(CI)$	Fall time, XTAL1/CLKIN		5	ns
$t_r(CI)$	Rise time, XTAL1/CLKIN		5	ns
$t_w(CIL)$	Pulse duration, XTAL1/CLKIN low as a percentage of $t_{c(CI)}$	40	60	%
$t_w(CIH)$	Pulse duration, XTAL1/CLKIN high as a percentage of $t_{c(CI)}$	40	60	%

† This device utilizes a fully static design and, therefore, can operate with input clock cycle time [ $t_{c(CI)}$ ] approaching infinity. The device is characterized at frequencies approaching 0 Hz, but is tested at  $f_{clk} = 6.7$  MHz to meet device test time requirements.

**external reference crystal with PLL-circuit-enabled clock option**

The internal oscillator is enabled by connecting  $\overline{OSCBYP}$  to  $V_{DD}$  and connecting a crystal across XTAL1/CLKIN and XTAL2 pins as shown in Figure 19. The crystal should be in either fundamental or overtone operation and parallel resonant, with an effective series resistance of 30  $\Omega$  and a power dissipation of 1 mW; it should be specified at a load capacitance of 20 pF. Note that overtone crystals require an additional tuned-LC circuit.

**timings with the PLL circuit enabled**

PARAMETER		EXTERNAL REFERENCE CRYSTAL	MIN	TYP	MAX	UNIT
$f_x$	Input clock frequency	5 MHz		5		MHz
C1, C2	Load capacitance			10		pF

**switching characteristics over recommended operating conditions [ $H = 0.5 t_{c(CO)}$ ] (see Figure 20)**

PARAMETER	CLOCK MODE	MIN	TYP	MAX	UNIT
$t_{c(CPU)}$ Cycle time, CPUCLK	before PLL lock, CLKIN divide by 2		$2t_{c(CI)}$	†	ns
	before PLL lock, CLKIN divide by 1		$t_{c(CI)}$		
	after PLL lock		50		
$t_{c(SYS)}$ Cycle time, SYSCLK	CPUCLK divide by 2		$2t_{c(CPU)}$	†	ns
	CPUCLK divide by 4‡		$4t_{c(CPU)}$		
$t_{c(CO)}$ Cycle time, CLKOUT		50		†	ns
$t_f(CO)$ Fall time, CLKOUT			5		ns
$t_r(CO)$ Rise time, CLKOUT			5		ns
$t_w(COL)$ Pulse duration, CLKOUT low		H–10	H–5	H–3	ns
$t_w(COH)$ Pulse duration, CLKOUT high		H+3	H+5	H+10	ns
$t_p$ Transition time, PLL synchronized after PLL enabled	before PLL lock, CLKIN divide by 2			$2000t_{c(CI)}$	ns
	before PLL lock, CLKIN divide by 1			$1000t_{c(CI)}$	

† This device utilizes a fully static design and, therefore, can operate with input clock cycle time [ $t_{c(CI)}$ ] approaching infinity. The device is characterized at frequencies approaching 0 Hz, but is tested at  $f_{clk} = 6.7$  MHz to meet device test time requirements.

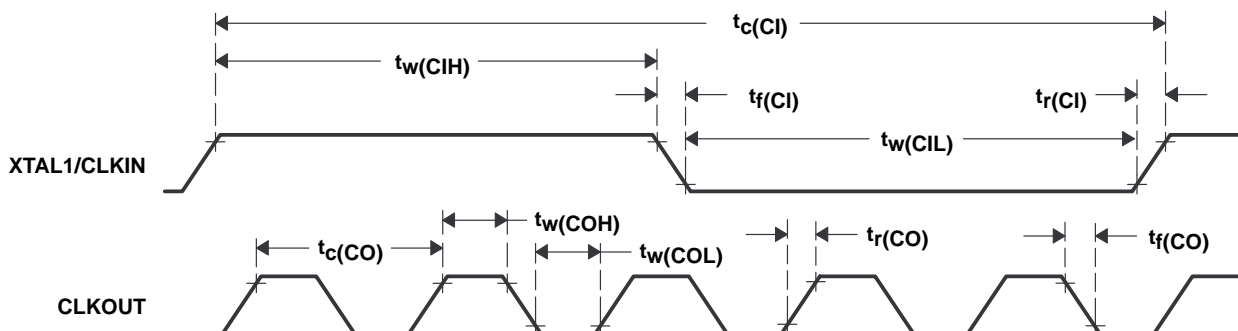
‡ SYSCLK is initialized to divide-by-4 mode by any device reset.

**timing requirements over recommended operating conditions (see Note 1 and Figure 20)**

	EXTERNAL REFERENCE CRYSTAL	MIN	MAX	UNIT
$t_{c(CI)}$ Cycle time, XTAL1/CLKIN	4 MHz	250	†	ns
	6 MHz	167		
	8 MHz	125		
$t_f(CI)$ Fall time, XTAL1/CLKIN			5	ns
$t_r(CI)$ Rise time, XTAL1/CLKIN			5	ns
$t_w(CIL)$ Pulse duration, XTAL1/CLKIN low as a percentage of $t_{c(CI)}$		40	60	%
$t_w(CIH)$ Pulse duration, XTAL1/CLKIN high as a percentage of $t_{c(CI)}$		40	60	%

† This device utilizes a fully static design and, therefore, can operate with input clock cycle time [ $t_{c(CI)}$ ] approaching infinity. The device is characterized at frequencies approaching 0 Hz, but is tested at  $f_{clk} = 6.7$  MHz to meet device test time requirements.

NOTE 1: Timings assume CLKOUT is set to output CPUCLK. CLKOUT is initialized to CPUCLK by power-on reset.



**Figure 20. CLKIN-to-CLKOUT Timing for PLL Oscillator Mode, Multiply-by-5 Option With 4-MHz Crystal**

## low-power mode timings

switching characteristics over recommended operating conditions [ $H = 0.5 t_{c(CO)}$ ]  
(see Figure 21, Figure 22, Figure 23, and Figure 24)<sup>†</sup>

PARAMETER	LOW-POWER MODES	MIN	TYP	MAX	UNIT
$t_d(\text{WAKE-A})$	Idle 1 and Idle 2	650		$15 \times t_{c(CO)}$	ns
	PLL or OSC power down	1500		$15 \times t_{c(CI)}$	
$t_d(\text{IDLE-COH})$	Delay time, Idle instruction executed to CLKOUT high (see Note 1)	300		500	ns
$t_d(\text{WAKE-LOCK})$	Delay time, CLKOUT switching to PLL synchronized (see Note 1)			100	$\mu\text{s}$
$t_d(\text{WAKE-OSC})$	Delay time, wakeup interrupt asserted to oscillator running			10	ms
$t_d(\text{IDLE-OSC})$	Delay time, Idle instruction executed to oscillator power off		60		$\mu\text{s}$

<sup>†</sup> Parameters are specified by design and not tested.

NOTE 1: Timings assume CLKOUT is set to output CPUCLK. CLKOUT is initialized to CPUCLK by power-on reset.

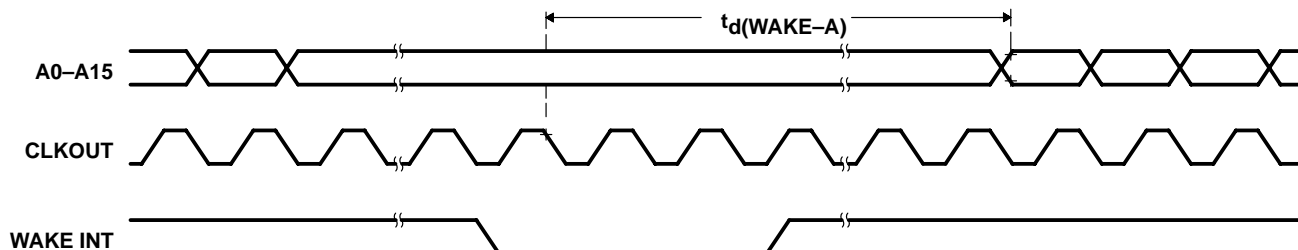


Figure 21. IDLE1 Entry and Exit Timing

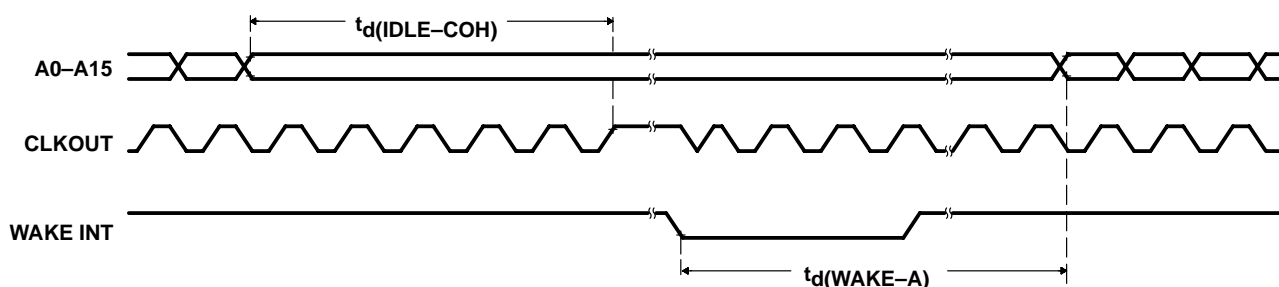


Figure 22. IDLE2 Entry and Exit Timing

PRODUCT PREVIEW

low-power mode timings (continued)

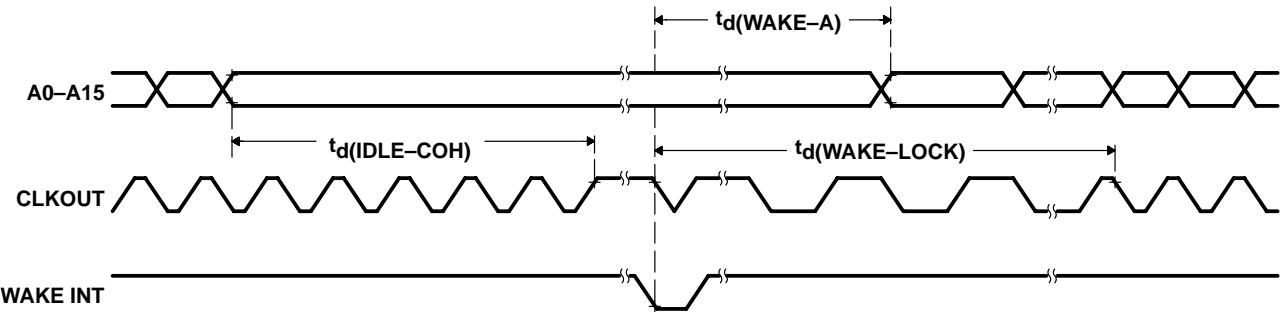


Figure 23. PLL Power-Down Entry and Exit Timing

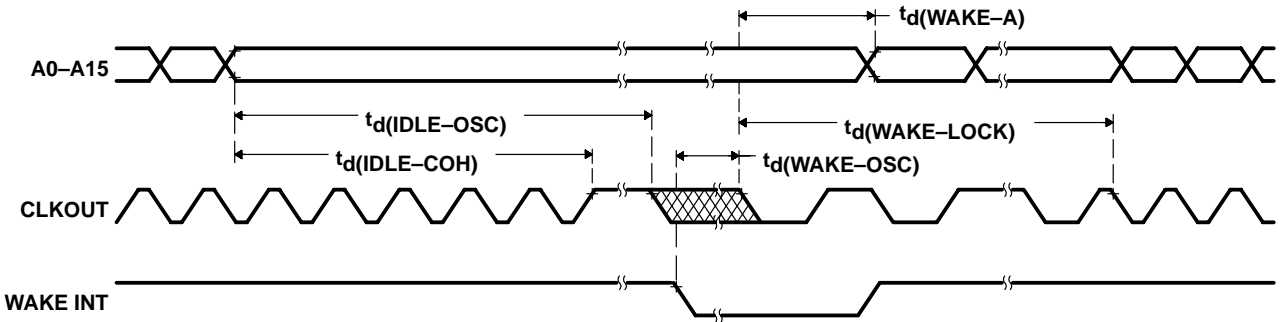


Figure 24. OSC Power-Down Entry and Exit Timing

## $\overline{RS}$ and $\overline{PORESET}$ timings

switching characteristics over recommended operating conditions for a reset [ $H = 0.5t_{c(CO)}$ ]  
(see Figure 25 and Figure 26)<sup>†</sup>

PARAMETER	MIN	MAX	UNIT
$t_w(RSL1)$ Pulse duration, $\overline{RS}$ low <sup>‡</sup>	$8t_{c(SYS)}$		ns
$t_d(RS)$ Delay time, program address at reset vector after $\overline{RS}$ low	4H		ns
$t_d(EX)$ Delay time, reset vector executed after $\overline{RS}$ high	32H		ns

<sup>†</sup> Parameters are specified by design and not tested.

<sup>‡</sup> The parameter  $t_w(RSL1)$  refers to the time  $\overline{RS}$  is an output.

timing requirements over recommended operating conditions for a reset [ $H = 0.5t_{c(CO)}$ ]  
(see Figure 25 and Figure 26)

	MIN	MAX	UNIT
$t_w(RSL)$ Pulse duration, $\overline{RS}$ low <sup>§</sup>	5		ns

<sup>§</sup> The parameter  $t_w(RSL)$  refers to the time  $\overline{RS}$  is an input.

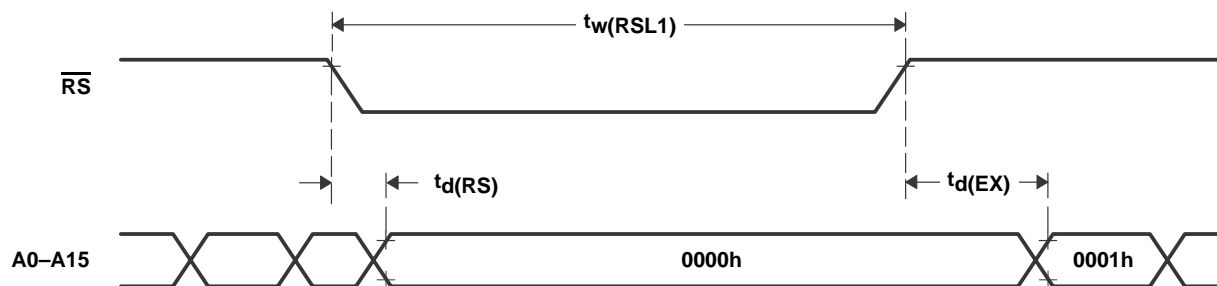


Figure 25. Reset Timing

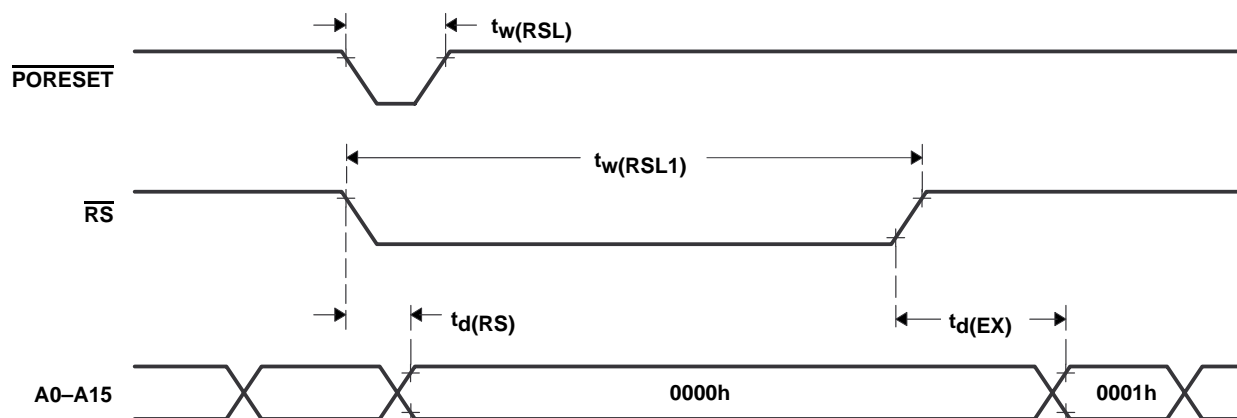


Figure 26. Power-On Reset Timing

TMS320F243  
DSP CONTROLLER

SPRS064 – DECEMBER 1997

XF,  $\overline{\text{BIO}}$ , and MP/ $\overline{\text{MC}}$  timings

switching characteristics over recommended operating conditions [ $H = 0.5t_{c(\text{CO})}$ ]

PARAMETER		MIN	MAX	UNIT
$t_{d(\text{XF})}$	Delay time, XF high/low after CLKOUT high	2	11	ns

timing requirements over recommended operating conditions [ $H = 0.5t_{c(\text{CO})}$ ]

		MIN	MAX	UNIT
$t_{w(\text{BIOL})}$	Pulse duration, $\overline{\text{BIO}}$ or MP/ $\overline{\text{MC}}$ low	$2H + 5$		ns
$t_{su(\text{BIO})\text{CO}}$	Setup time, $\overline{\text{BIO}}$ or MP/ $\overline{\text{MC}}$ low before CLKOUT low	15		ns
$t_{h(\text{BIO})\text{CO}}$	Hold time, $\overline{\text{BIO}}$ or MP/ $\overline{\text{MC}}$ low after CLKOUT low	-6		ns

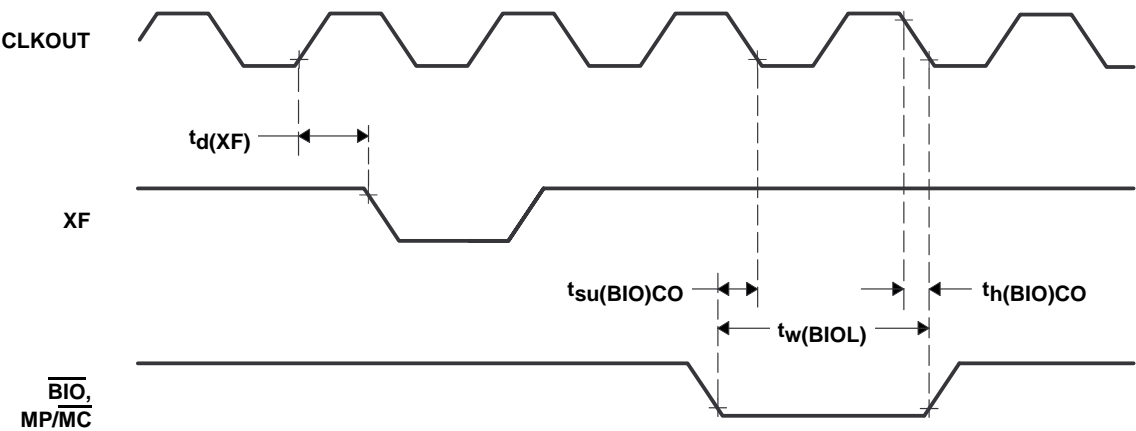


Figure 27. XF and  $\overline{\text{BIO}}$  Timing



## TIMING EVENT MANAGER INTERFACE

### PWM timings

PWM refers to PWM1, PWM2, PWM3, PWM4, PWM5, PWM6, T1PWM, T2PWM, T3PWM, PWM7, PWM8, and PWM9.

### switching characteristics over recommended operating conditions for PWM timing [H = 0.5t<sub>c</sub>(CO)] (see Figure 28)

PARAMETER	MIN	MAX	UNIT
t <sub>w</sub> (PWM) <sup>†</sup> Pulse duration, PWM output high/low	50		ns
t <sub>d</sub> (PWM)CO Delay time, PWM output switching after CLKOUT low	0	8	ns

<sup>†</sup> PWM outputs may be 100%, 0%, or increments of t<sub>c</sub>(CO) with respect to the PWM period.

### timing requirements over recommended operating conditions for PWM timing [H = 0.5t<sub>c</sub>(CO)] (see Figure 29 and Figure 30)

	MIN	MAX	UNIT
t <sub>w</sub> (TMRDIR) Pulse duration, TMRDIR low/high	t <sub>c</sub> (TMRCLK) + 5		ns
t <sub>su</sub> (TMRDIR) Setup time, TMRDIR low/high before CLKOUT low	15		ns
t <sub>w</sub> (TMRCLK) Pulse duration, TMRCLK low as a percentage of TMRCLK cycle time	40	60	%
t <sub>wh</sub> (TMRCLK) Pulse duration, TMRCLK high as a percentage of TMRCLK cycle time	40	60	%
t <sub>c</sub> (TMRCLK) Cycle time, TMRCLK	4 × t <sub>c</sub> (CPU)		ns
t <sub>h</sub> (TMRDIR) Hold time, TMRDIR low/high after CLKOUT low	–6		ns
t <sub>su</sub> (TMRCLK) Setup time, TMRCLK low/high before CLKOUT low	15		ns
t <sub>h</sub> (TMRCLK) Hold time, TMRCLK low/high after CLKOUT low	–6		ns

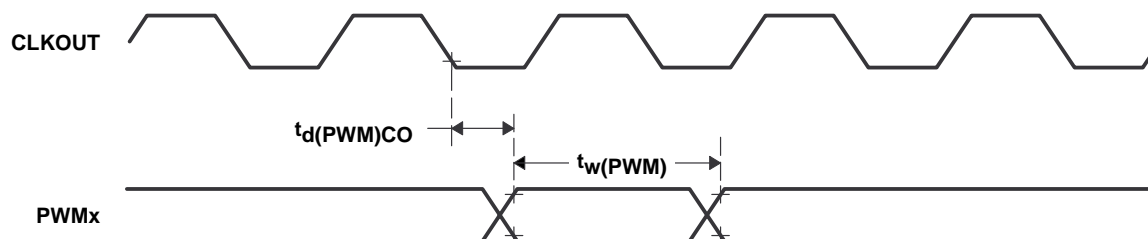


Figure 28. PWM Output Timing

PWM timings (continued)

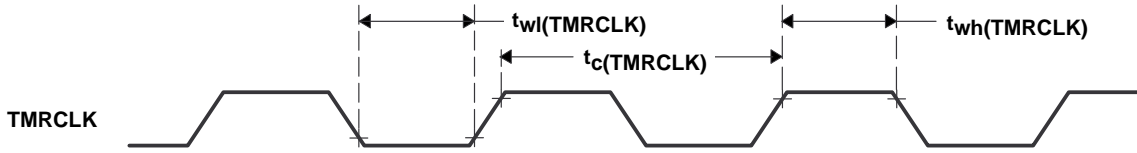


Figure 29. External Timer Clock Input Timing

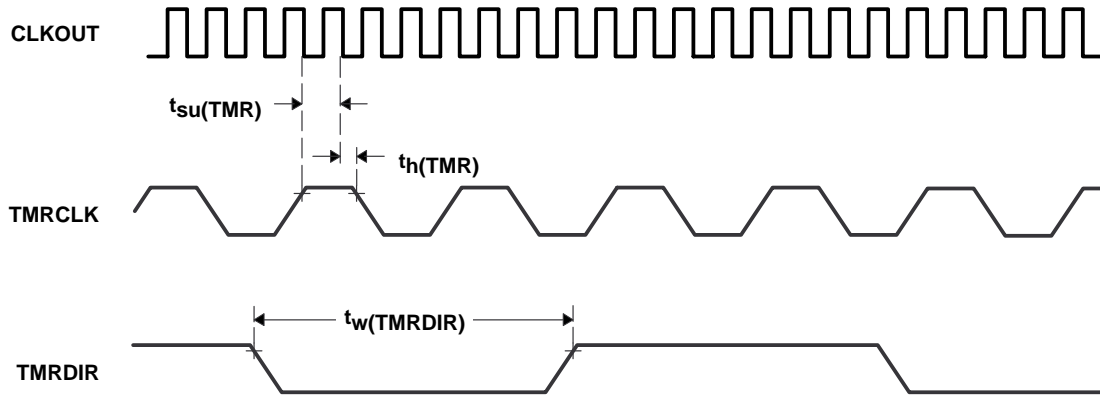


Figure 30. External Timer Direction Input Timing

capture and QEP timings

CAP refers to CAP1/QEP1/IOPC4, CAP2/QEP2/IOPC5, CAP3/IOPC6, and CAP4/IOPC7.

timing requirements over recommended operating conditions [ $H = 0.5t_{c(CO)}$ ] (see Figure 31)

		MIN	MAX	UNIT
$t_w(CAP)$	Pulse duration, CAP input low/high	$t_c(CPU) + 5$		ns
$t_{su}(CAP)$	Setup time, CAP low/high before CLKOUT low	15		ns
$t_h(CAP)CO$	Hold time, CAP low/high after CLKOUT low	-6		ns

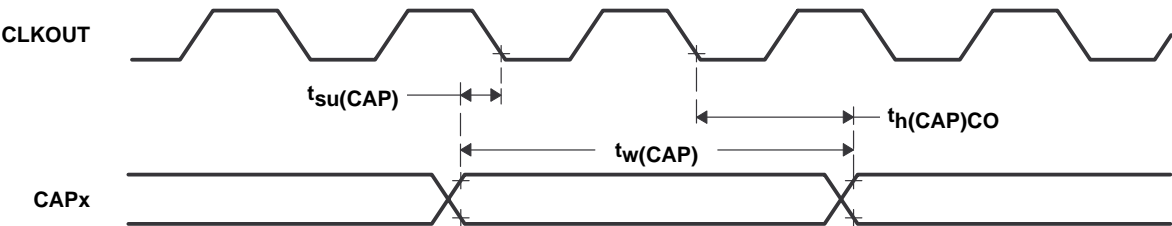


Figure 31. Capture Input and QEP Timing

# TMS320F243 DSP CONTROLLER

SPRS064 – DECEMBER 1997

## interrupt timing

INT refers to NMI, XINT1, and XINT2/IO. PDP refers to  $\overline{\text{PDPINT}}$ .

## switching characteristics over recommended operating conditions interrupts [ $H = 0.5t_{c(CO)}$ ]

PARAMETER	MIN	MAX	UNIT
$t_{hz(PWM)PDP}$ Delay time, PWM to high-impedance state after $\overline{\text{PDPINT}}$ low†	0	5	ns

† Values derived from characterization data and not tested.

## timing requirements over recommended operating conditions interrupts [ $H = 0.5t_{c(CO)}$ ]

	MIN	MAX	UNIT
$t_w(INT)$ Pulse duration, INT input low/high	$t_{c(SYS)} + 5$		ns
$t_w(PDP)$ Pulse duration, $\overline{\text{PDPINT}}$ input low			ns
$t_{su(PDP)CO}$ Setup time, $\overline{\text{PDPINT}}$ input low before CLKOUT low	3		ns
$t_h(PDP)CO$ Hold time, $\overline{\text{PDPINT}}$ input low after CLKOUT low	0		ns
$t_d(INT)$ Delay time, INT low/high to interrupt-vector fetch‡	$2t_{c(SYS)} + 4t_{c(CPU)}$		ns
$t_{su(INT)}$ Setup time, INT input low before CLKOUT low	3		ns
$t_h(INT)$ Hold time, INT input low after CLKOUT low	0		ns

‡ Parameters are specified by design and not tested.

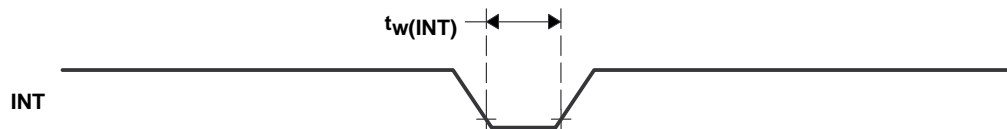


Figure 32. External Interrupt Timing

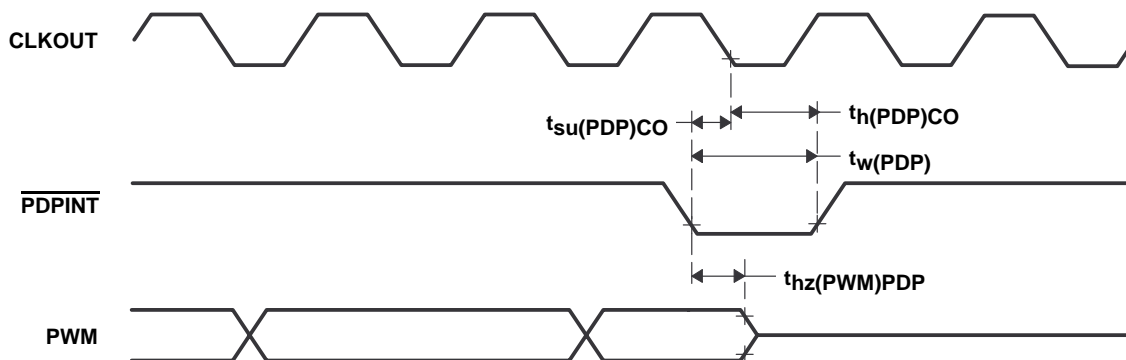


Figure 33. Power Drive Protection Interrupt Timing

## general-purpose input/output timing

GPO refers to the digital output function of shared pins IOPA0–3, IOPB0–7, IOPC0–7, XINT2/IO.

GPI refers to the digital input function of shared pins IOPA0–3, IOPB0–7, IOPC0–7, XINT2/IO.

## switching characteristics over recommended operating conditions for a GPI/O [ $H = 0.5t_{c(CO)}$ ] (see Figure 34)

PARAMETER		MIN	TYP	MAX	UNIT
$t_{d(GPO)CO}$	Delay time, GPO low/high from CLKOUT low	13		50	ns
	All other GPOs			25	
$t_{r(GPO)}$	Rise time, GPO switching low to high		50		ns
$t_{f(GPO)}$	Fall time, GPO switching high to low		50		ns

## timing requirements over recommended operating conditions for a GPI/O [ $H = 0.5t_{c(CO)}$ ] (see Figure 35)

	MIN	MAX	UNIT
$t_{su(GPI)CO}$	Setup time, GPI high/low before CLKOUT high	3	ns
$t_{h(GPI)CO}$	Hold time, GPI high/low after CLKOUT high	0	ns
$t_w(GPI)$	Pulse duration, GPI high/low	$t_{c(SYS)} + 5$	ns

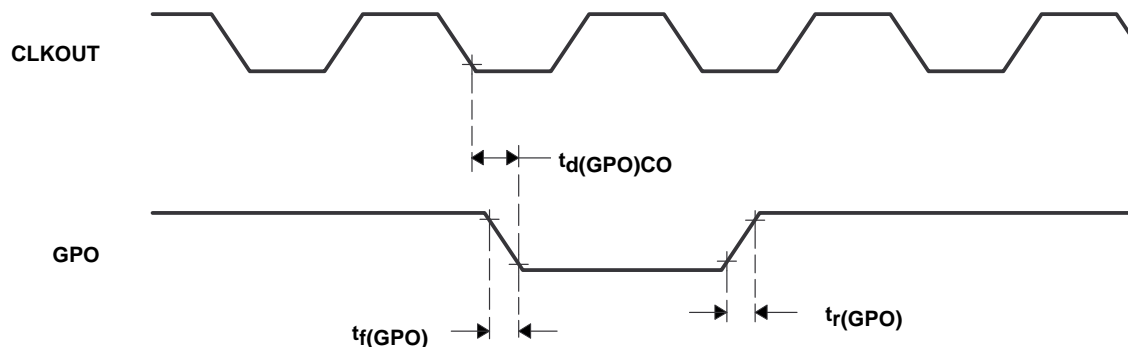


Figure 34. General-Purpose Output Timing

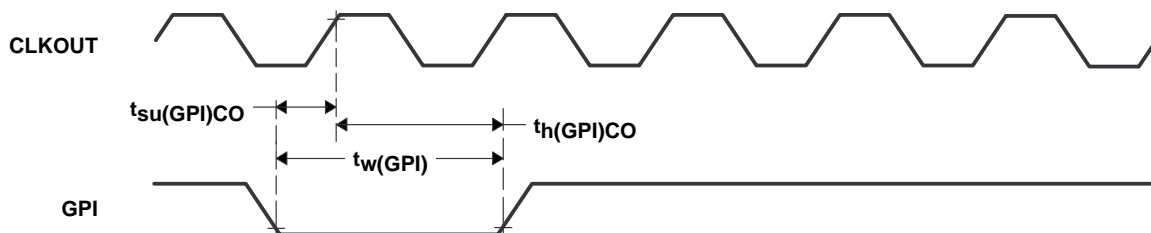


Figure 35. General-Purpose Input Timing

## PRODUCT PREVIEW

## SPI MASTER MODE TIMING PARAMETERS

SPI master mode timing information is listed in the following tables.

SPI master mode external timing parameters (clock phase = 0)<sup>†</sup> (see Figure 36)

NO.			SPI WHEN (SPIBRR + 1) IS EVEN OR SPIBRR = 0 OR 2		SPI WHEN (SPIBRR + 1) IS ODD AND SPIBRR > 3		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{c(SPC)M}$	Cycle time, SPICLK	$4t_c^{\ddagger}$	$128t_c^{\ddagger}$	$5t_c^{\ddagger}$	$127t_c^{\ddagger}$	ns
2§	$t_w(SPCH)M$	Pulse duration, SPICLK high (clock polarity = 0)	$0.5t_{c(SPC)M} - 70$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M} - 0.5t_c - 70$	$0.5t_{c(SPC)M} - 0.5t_c$	ns
	$t_w(SPCL)M$	Pulse duration, SPICLK low (clock polarity = 1)	$0.5t_{c(SPC)M} - 70$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M} - 0.5t_c - 70$	$0.5t_{c(SPC)M} - 0.5t_c$	
3§	$t_w(SPCL)M$	Pulse duration, SPICLK low (clock polarity = 0)	$0.5t_{c(SPC)M} - 70$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M} + 0.5t_c - 70$	$0.5t_{c(SPC)M} + 0.5t_c$	ns
	$t_w(SPCH)M$	Pulse duration, SPICLK high (clock polarity = 1)	$0.5t_{c(SPC)M} - 70$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M} + 0.5t_c - 70$	$0.5t_{c(SPC)M} + 0.5t_c$	
4§	$t_d(SPCH-SIMO)M$	Delay time, SPISIMO valid after SPICLK high (clock polarity = 0)	- 10	10	- 10	10	ns
	$t_d(SPCL-SIMO)M$	Delay time, SPISIMO valid after SPICLK low (clock polarity = 1)	- 10	10	- 10	10	
5§	$t_v(SPCL-SIMO)M$	Valid time, SPISIMO data valid after SPICLK low (clock polarity = 0)	$0.5t_{c(SPC)M} - 70$		$0.5t_{c(SPC)M} + 0.5t_c - 70$		ns
	$t_v(SPCH-SIMO)M$	Valid time, SPISIMO data valid after SPICLK high (clock polarity = 1)	$0.5t_{c(SPC)M} - 70$		$0.5t_{c(SPC)M} + 0.5t_c - 70$		
8§	$t_{su}(SOMI-SPCL)M$	Setup time, SPISOMI to SPICLK low (clock polarity = 0)	0		0		ns
	$t_{su}(SOMI-SPCH)M$	Setup time, SPISOMI to SPICLK high (clock polarity = 1)	0		0		
9§	$t_v(SPCL-SOMI)M$	Valid time, SPISOMI data valid after SPICLK low (clock polarity = 0)	$0.25t_{c(SPC)M} - 70$		$0.5t_{c(SPC)M} - 0.5t_c - 70$		ns
	$t_v(SPCH-SOMI)M$	Valid time, SPISOMI data valid after SPICLK high (clock polarity = 1)	$0.25t_{c(SPC)M} - 70$		$0.5t_{c(SPC)M} - 0.5t_c - 70$		

<sup>†</sup> The MASTER/SLAVE bit (SPICTL.2) is set and the CLOCK PHASE bit (SPICTL.3) is cleared.

<sup>‡</sup>  $t_c$  = system clock cycle time = 1 / SYSCLK =  $t_{c(SYS)}$

<sup>§</sup> The active edge of the SPICLK signal referenced is controlled by the CLOCK POLARITY bit (SPICCR.6).



## PRODUCT PREVIEW

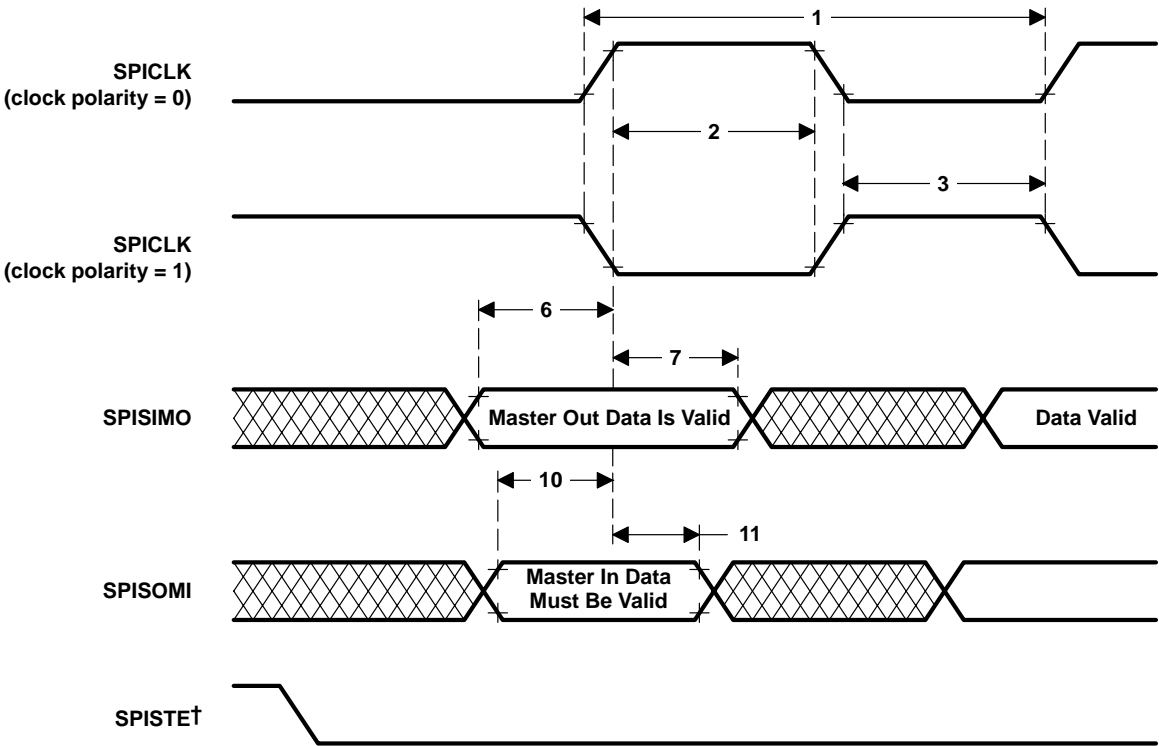
SPI master mode external timing parameters (clock phase = 1)<sup>†</sup> (see Figure 37)

NO.			SPI WHEN (SPIBRR + 1) IS EVEN OR SPIBRR = 0 OR 2		SPI WHEN (SPIBRR + 1) IS ODD AND SPIBRR > 3		UNIT
			MIN	MAX	MIN	MAX	
1	$t_c(\text{SPC})_M$	Cycle time, SPICLK	$4t_c^{\ddagger}$	$128t_c^{\ddagger}$	$5t_c^{\ddagger}$	$127t_c^{\ddagger}$	ns
2§	$t_w(\text{SPCH})_M$	Pulse duration, SPICLK high (clock polarity = 0)	$0.5t_c(\text{SPC})_M - 70$	$0.5t_c(\text{SPC})_M$	$0.5t_c(\text{SPC})_M - 0.5t_c - 70$	$0.5t_c(\text{SPC})_M - 0.5t_c$	ns
	$t_w(\text{SPCL})_M$	Pulse duration, SPICLK low (clock polarity = 1)	$0.5t_c(\text{SPC})_M - 70$	$0.5t_c(\text{SPC})_M$	$0.5t_c(\text{SPC})_M - 0.5t_c - 70$	$0.5t_c(\text{SPC})_M - 0.5t_c$	
3§	$t_w(\text{SPCL})_M$	Pulse duration, SPICLK low (clock polarity = 0)	$0.5t_c(\text{SPC})_M - 70$	$0.5t_c(\text{SPC})_M$	$0.5t_c(\text{SPC})_M + 0.5t_c - 70$	$0.5t_c(\text{SPC})_M + 0.5t_c$	ns
	$t_w(\text{SPCH})_M$	Pulse duration, SPICLK high (clock polarity = 1)	$0.5t_c(\text{SPC})_M - 70$	$0.5t_c(\text{SPC})_M$	$0.5t_c(\text{SPC})_M + 0.5t_c - 70$	$0.5t_c(\text{SPC})_M + 0.5t_c$	
6§	$t_{su}(\text{SIMO-SPCH})_M$	Setup time, SPISIMO data valid before SPICLK high (clock polarity = 0)	$0.5t_c(\text{SPC})_M - 70$		$0.5t_c(\text{SPC})_M - 70$		ns
	$t_{su}(\text{SIMO-SPCL})_M$	Setup time, SPISIMO data valid before SPICLK low (clock polarity = 1)	$0.5t_c(\text{SPC})_M - 70$		$0.5t_c(\text{SPC})_M - 70$		
7§	$t_v(\text{SPCH-SIMO})_M$	Valid time, SPISIMO data valid after SPICLK high (clock polarity = 0)	$0.5t_c(\text{SPC})_M - 70$		$0.5t_c(\text{SPC})_M - 70$		ns
	$t_v(\text{SPCL-SIMO})_M$	Valid time, SPISIMO data valid after SPICLK low (clock polarity = 1)	$0.5t_c(\text{SPC})_M - 70$		$0.5t_c(\text{SPC})_M - 70$		
10§	$t_{su}(\text{SOMI-SPCH})_M$	Setup time, SPISOMI to SPICLK high (clock polarity = 0)	0		0		ns
	$t_{su}(\text{SOMI-SPCL})_M$	Setup time, SPISOMI to SPICLK low (clock polarity = 1)	0		0		
11§	$t_v(\text{SPCH-SOMI})_M$	Valid time, SPISOMI data valid after SPICLK high (clock polarity = 0)	$0.25t_c(\text{SPC})_M - 70$		$0.5t_c(\text{SPC})_M - 70$		ns
	$t_v(\text{SPCL-SOMI})_M$	Valid time, SPISOMI data valid after SPICLK low (clock polarity = 1)	$0.25t_c(\text{SPC})_M - 70$		$0.5t_c(\text{SPC})_M - 70$		

<sup>†</sup> The MASTER/SLAVE bit (SPICTL.2) is set and the CLOCK PHASE bit (SPICTL.3) is set.<sup>‡</sup>  $t_c$  = system clock cycle time = 1 / SYSCLK =  $t_c(\text{SYS})$ <sup>§</sup> The active edge of the SPICLK signal referenced is controlled by the CLOCK POLARITY bit (SPICCR.6).



PARAMETER MEASUREMENT INFORMATION



† The SPISTE signal must be active before the SPI communication stream starts; the SPISTE signal must remain active until the SPI communication stream is complete.

Figure 37. SPI Master Mode External Timing (Clock Phase = 1)

## SPI SLAVE MODE TIMING PARAMETERS

Slave mode timing information is illustrated in the following table and figures.

### SPI slave mode external timing parameters (clock phase = 0)<sup>†</sup> (see Figure 38)

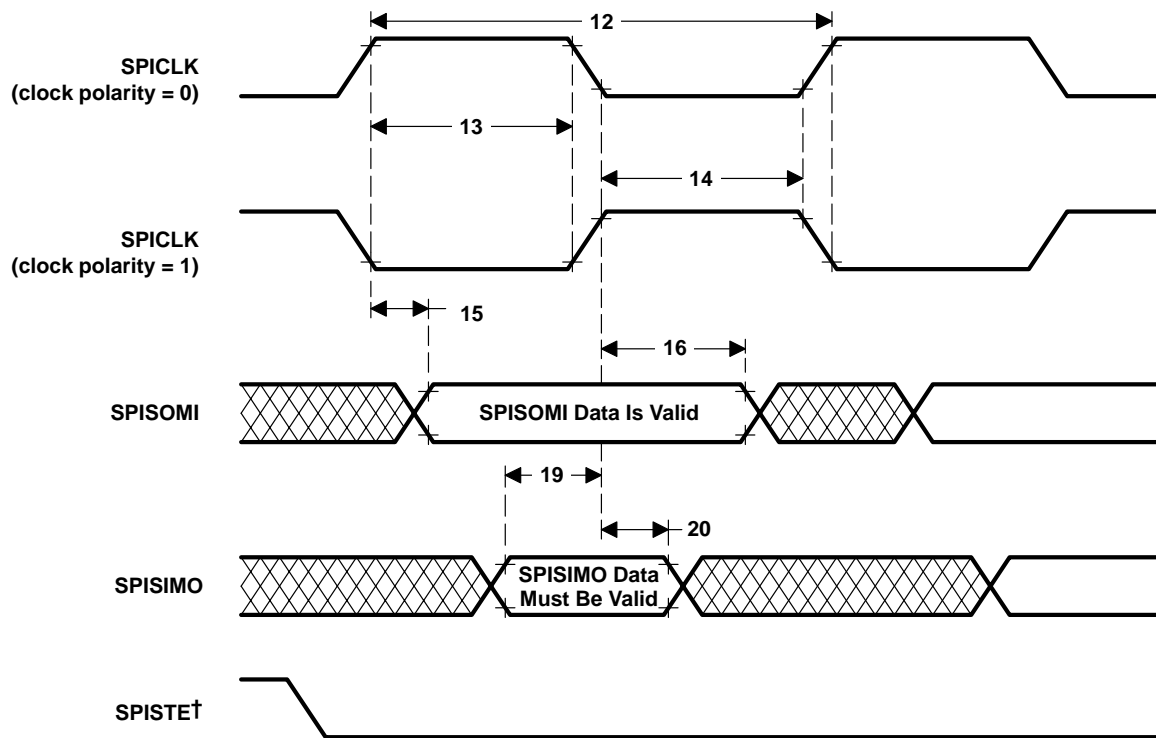
NO.		MIN	MAX	UNIT
12	$t_c(\text{SPC})S$ Cycle time, SPICLK	$8t_c^{\ddagger}$		ns
13§	$t_w(\text{SPCH})S$ Pulse duration, SPICLK high (clock polarity = 0)	$0.5t_c(\text{SPC})S - 70$	$0.5t_c(\text{SPC})S$	ns
	$t_w(\text{SPCL})S$ Pulse duration, SPICLK low (clock polarity = 1)	$0.5t_c(\text{SPC})S - 70$	$0.5t_c(\text{SPC})S$	
14§	$t_w(\text{SPCL})S$ Pulse duration, SPICLK low (clock polarity = 0)	$0.5t_c(\text{SPC})S - 70$	$0.5t_c(\text{SPC})S$	ns
	$t_w(\text{SPCH})S$ Pulse duration, SPICLK high (clock polarity = 1)	$0.5t_c(\text{SPC})S - 70$	$0.5t_c(\text{SPC})S$	
15§	$t_d(\text{SPCH-SOMI})S$ Delay time, SPISOMI valid after SPICLK high (clock polarity = 0)	$0.375t_c(\text{SPC})S - 70$		ns
	$t_d(\text{SPCL-SOMI})S$ Delay time, SPISOMI valid after SPICLK low (clock polarity = 1)	$0.375t_c(\text{SPC})S - 70$		
16§	$t_v(\text{SPCL-SOMI})S$ Valid time, SPISOMI data valid after SPICLK low (clock polarity = 0)	$0.75t_c(\text{SPC})S$		ns
	$t_v(\text{SPCH-SOMI})S$ Valid time, SPISOMI data valid after SPICLK high (clock polarity = 1)	$0.75t_c(\text{SPC})S$		
19§	$t_{su}(\text{SIMO-SPCL})S$ Setup time, SPISIMO to SPICLK low (clock polarity = 0)	0		ns
	$t_{su}(\text{SIMO-SPCH})S$ Setup time, SPISIMO to SPICLK high (clock polarity = 1)	0		
20§	$t_v(\text{SPCL-SIMO})S$ Valid time, SPISIMO data valid after SPICLK low (clock polarity = 0)	$0.5t_c(\text{SPC})S$		ns
	$t_v(\text{SPCH-SIMO})S$ Valid time, SPISIMO data valid after SPICLK high (clock polarity = 1)	$0.5t_c(\text{SPC})S$		

<sup>†</sup> The MASTER/SLAVE bit (SPICTL.2) is cleared and the CLOCK PHASE bit (SPICTL.3) is cleared.

<sup>‡</sup>  $t_c$  = system clock cycle time =  $1/\text{SYSCLK} = t_c(\text{SYS})$

<sup>§</sup> The active edge of the SPICLK signal referenced is controlled by the CLOCK POLARITY bit (SPICCR.6).

# PARAMETER MEASUREMENT INFORMATION



† The SPISIMO signal must be active before the SPI communication stream starts; the SPISIMO signal must remain active until the SPI communication stream is complete.

Figure 38. SPI Slave Mode External Timing (Clock Phase = 0)

# TMS320F243

## DSP CONTROLLER

SPRS064 – DECEMBER 1997

### SPI slave mode external timing parameters (clock phase = 1)<sup>†</sup> (see Figure 39)

NO.		MIN	MAX	UNIT
12	$t_c(\text{SPC})S$ Cycle time, SPICLK	$8t_c^{\ddagger}$		ns
13§	$t_w(\text{SPCH})S$ Pulse duration, SPICLK high (clock polarity = 0)	$0.5t_c(\text{SPC})S - 70$	$0.5t_c(\text{SPC})S$	ns
	$t_w(\text{SPCL})S$ Pulse duration, SPICLK low (clock polarity = 1)	$0.5t_c(\text{SPC})S - 70$	$0.5t_c(\text{SPC})S$	
14§	$t_w(\text{SPCL})S$ Pulse duration, SPICLK low (clock polarity = 0)	$0.5t_c(\text{SPC})S - 70$	$0.5t_c(\text{SPC})S$	ns
	$t_w(\text{SPCH})S$ Pulse duration, SPICLK high (clock polarity = 1)	$0.5t_c(\text{SPC})S - 70$	$0.5t_c(\text{SPC})S$	
17§	$t_{su}(\text{SOMI-SPCH})S$ Setup time, SPISOMI to SPICLK high (clock polarity = 0)	$0.125t_c(\text{SPC})S$		ns
	$t_{su}(\text{SOMI-SPCL})S$ Setup time, SPISOMI to SPICLK low (clock polarity = 1)	$0.125t_c(\text{SPC})S$		
18§	$t_v(\text{SPCH-SOMI})S$ Valid time, SPISOMI data valid after SPICLK high (clock polarity = 0)	$0.75t_c(\text{SPC})S$		ns
	$t_v(\text{SPCL-SOMI})S$ Valid time, SPISOMI data valid after SPICLK low (clock polarity = 1)	$0.75t_c(\text{SPC})S$		
21§	$t_{su}(\text{SIMO-SPCH})S$ Setup time, SPISIMO to SPICLK high (clock polarity = 0)	0		ns
	$t_{su}(\text{SIMO-SPCL})S$ Setup time, SPISIMO to SPICLK low (clock polarity = 1)	0		
22§	$t_v(\text{SPCH-SIMO})S$ Valid time, SPISIMO data valid after SPICLK high (clock polarity = 0)	$0.5t_c(\text{SPC})S$		ns
	$t_v(\text{SPCL-SIMO})S$ Valid time, SPISIMO data valid after SPICLK low (clock polarity = 1)	$0.5t_c(\text{SPC})S$		

<sup>†</sup> The MASTER/SLAVE bit (SPICTL.2) is cleared and the CLOCK PHASE bit (SPICTL.3) is set.

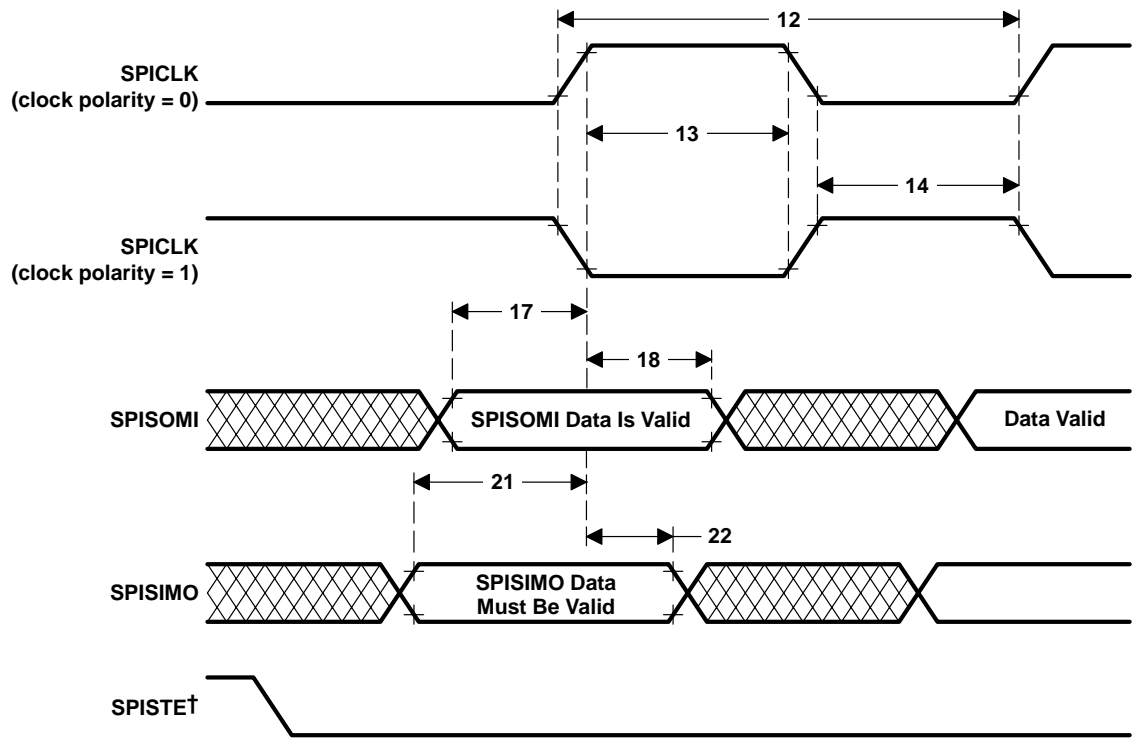
<sup>‡</sup>  $t_c$  = system clock cycle time =  $1/\text{SYSCLK} = t_c(\text{SYS})$

<sup>§</sup> The active edge of the SPICLK signal referenced is controlled by the CLOCK POLARITY bit (SPICCR.6).

PRODUCT PREVIEW



PARAMETER MEASUREMENT INFORMATION



† The SPISTE signal must be active before the SPI communication stream starts; the SPISTE signal must remain active until the SPI communication stream is complete.

Figure 39. SPI Slave Mode External Timing (Clock Phase = 1)

# TMS320F243 DSP CONTROLLER

SPRS064 – DECEMBER 1997

## external memory interface read timings

switching characteristics over recommended operating conditions for an external memory interface read (see Figure 40)

PARAMETER	MIN	MAX	UNIT
$t_d(\text{COL-CNTL})$ Delay time, control valid after CLKOUT low			ns
$t_d(\text{COL-CNTH})$ Delay time, control inactive after CLKOUT low			ns
$t_d(\text{COL-A})$ Delay time, address valid after CLKOUT low			ns
$t_d(\text{COH-RDL})$ Delay time, $\overline{\text{RD}}$ strobe active after CLKOUT high			ns
$t_d(\text{COL-RDH})$ Delay time, $\overline{\text{RD}}$ strobe inactive high after CLKOUT low			ns
$t_d(\text{COL-SL})$ Delay time, $\overline{\text{STRB}}$ active low after CLKOUT low			ns
$t_d(\text{COL-SH})$ Delay time, $\overline{\text{STRB}}$ strobe inactive high after CLKOUT low			ns

timing requirements over recommended operating conditions for an external memory interface read (see Figure 40)

	MIN	MAX	UNIT
$t_h(\text{A})\text{COL}$ Hold time, address valid after CLKOUT low			ns
$t_{su}(\text{A})\text{RD}$ Setup time, address valid before $\overline{\text{RD}}$ strobe active low			ns
$t_h(\text{A})\text{RD}$ Hold time, address valid after $\overline{\text{RD}}$ strobe inactive high			ns
$t_a(\text{A})$ Access time, read data from address valid			ns
$t_{su}(\text{D})\text{RD}$ Setup time, read data before $\overline{\text{RD}}$ strobe inactive high			ns
$t_h(\text{D})\text{RD}$ Hold time, read data from $\overline{\text{RD}}$ strobe inactive high			ns

external memory interface read timings (continued)

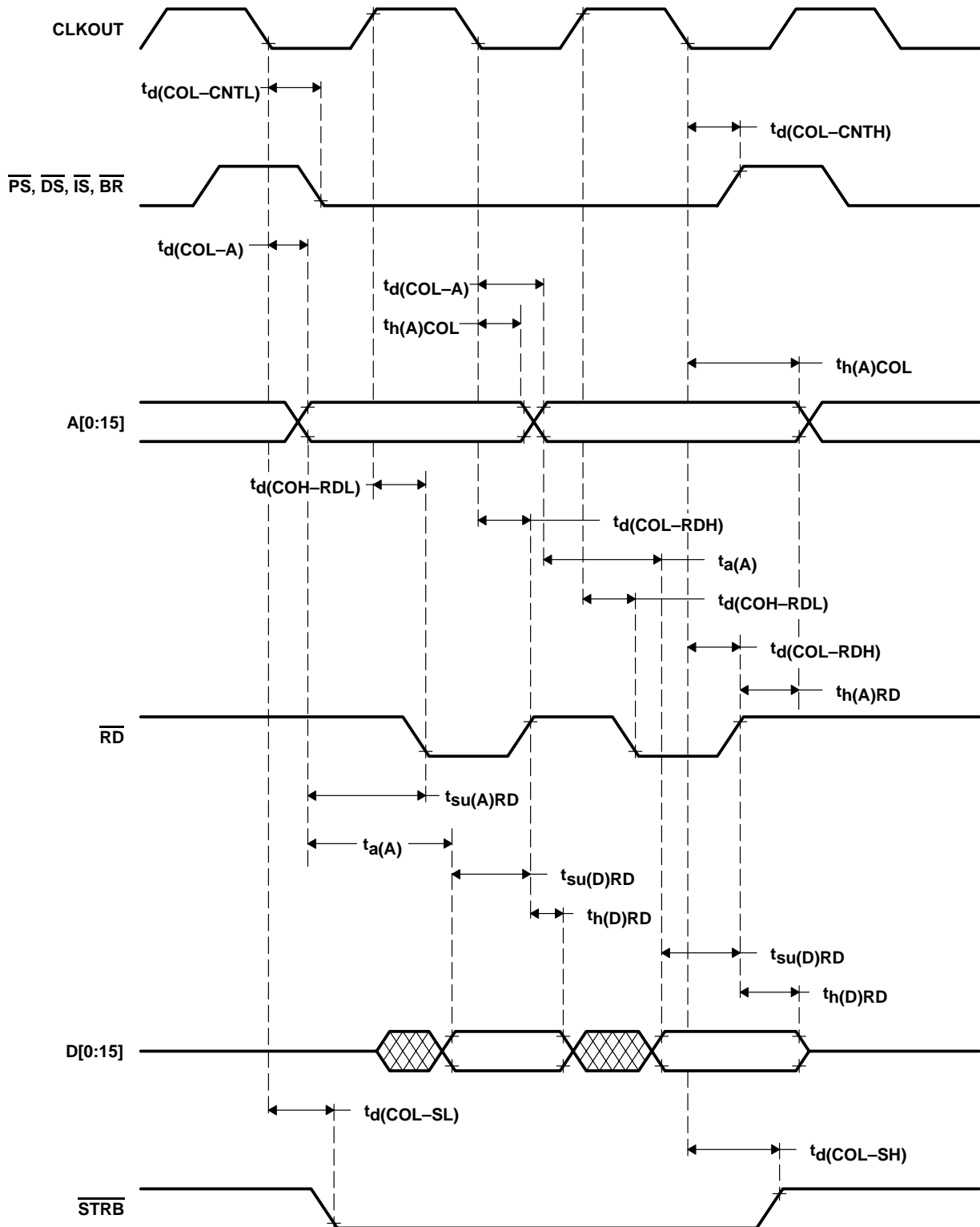


Figure 40. Memory Interface Read/Read Timings

## external memory interface write timings

**switching characteristics over recommended operating conditions for an external memory interface write (see Figure 41)**

PARAMETER	MIN	MAX	UNIT
$t_d(\text{COL-CNTL})$ Delay time, control valid after CLKOUT low			ns
$t_d(\text{COL-CNTH})$ Delay time, control inactive after CLKOUT low			ns
$t_d(\text{COL-A})$ Delay time, address valid after CLKOUT low			ns
$t_d(\text{COH-RWL})$ Delay time, $\overline{\text{R/W}}$ low after CLKOUT high			ns
$t_d(\text{COH-RWH})$ Delay time, $\overline{\text{R/W}}$ high after CLKOUT high			ns
$t_d(\text{COL-WL})$ Delay time, $\overline{\text{WE}}$ strobe active low after CLKOUT low			ns
$t_d(\text{COL-WH})$ Delay time, $\overline{\text{WE}}$ strobe inactive high after CLKOUT low			ns
$t_{\text{en}}(\text{D})\text{COL}$ Enable time, CLKOUT low to data bus driven			ns
$t_d(\text{COL-SL})$ Delay time, $\overline{\text{STRB}}$ active low after CLKOUT low			ns
$t_d(\text{COL-SH})$ Delay time, $\overline{\text{STRB}}$ inactive high after CLKOUT low			ns

**timing requirements over recommended operating conditions for an external memory interface write (see Figure 41)**

	MIN	MAX	UNIT
$t_h(\text{A})\text{COL}$ Hold time, address valid after CLKOUT low			ns
$t_{\text{su}}(\text{A})\text{W}$ Setup time, address valid before $\overline{\text{WE}}$ strobe active low			ns
$t_{\text{su}}(\text{D})\text{W}$ Setup time, write data before $\overline{\text{WR}}$ strobe inactive high			ns
$t_h(\text{D})\text{W}$ Hold time, write data from $\overline{\text{WR}}$ strobe inactive high			ns



external memory interface write timings (continued)

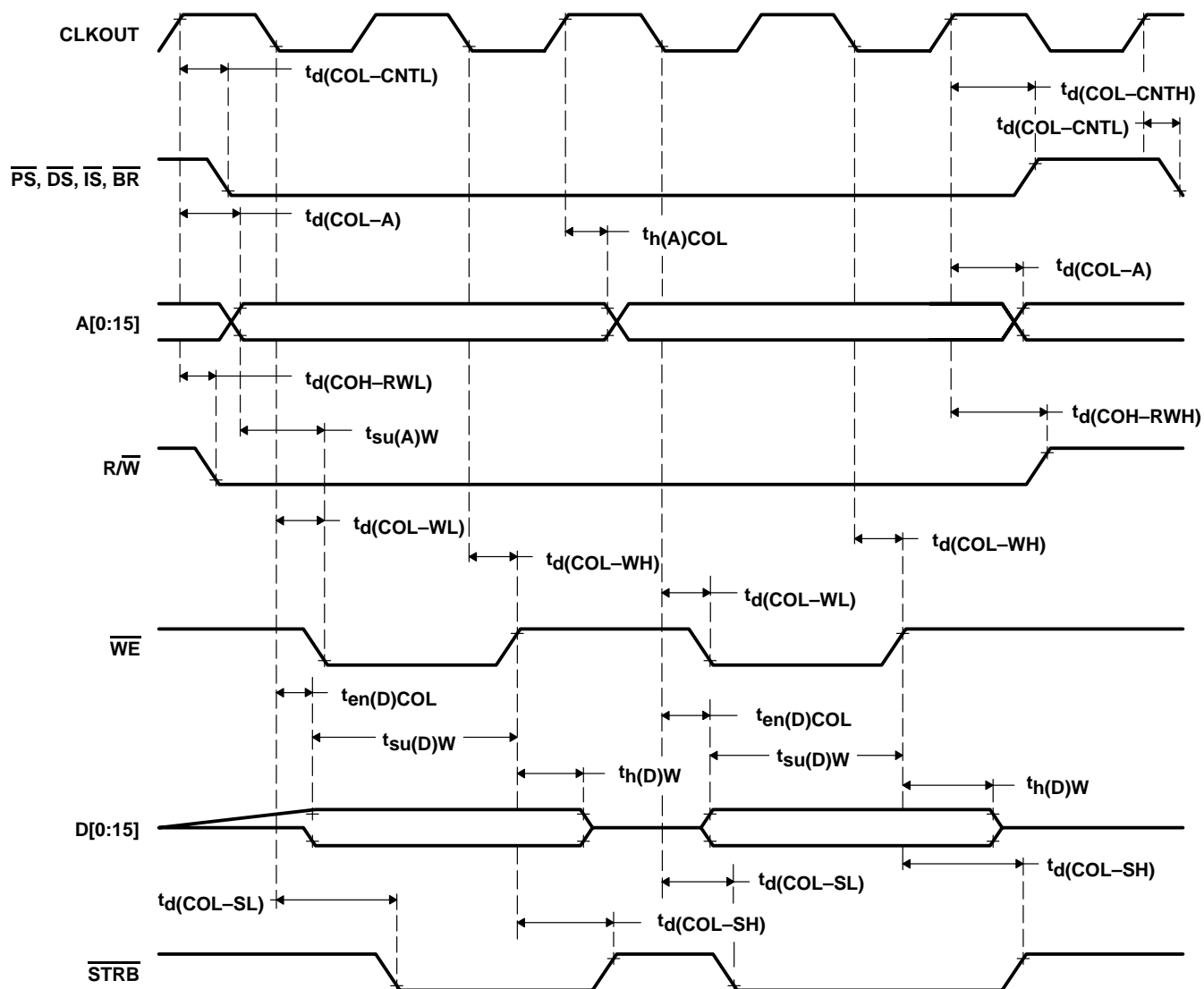


Figure 41. Write/Write Timings

PRODUCT PREVIEW

# TMS320F243

## DSP CONTROLLER

SPRS064 – DECEMBER 1997

### external memory interface ready-on-read timings

switching characteristics over recommended operating conditions for an external memory interface ready-on-read (see Figure 42)

PARAMETER	MIN	MAX	UNIT
$t_d(\text{COL-A})$ Delay time, address valid after CLKOUT low			ns
$t_h(\text{RDY})\text{COH}$ Hold time, READY after CLKOUT high			ns

timing requirements over recommended operating conditions for an external memory interface ready-on-read (see Figure 42)

	MIN	MAX	UNIT
$t_{su}(\text{D})\text{RD}$ Setup time, read data before $\overline{\text{RD}}$ strobe inactive high			ns
$t_v(\text{RDY})\text{ARD}$ Valid time, READY after address valid on read			ns
$t_{su}(\text{RDY})\text{COH}$ Setup time, READY before CLKOUT high			ns

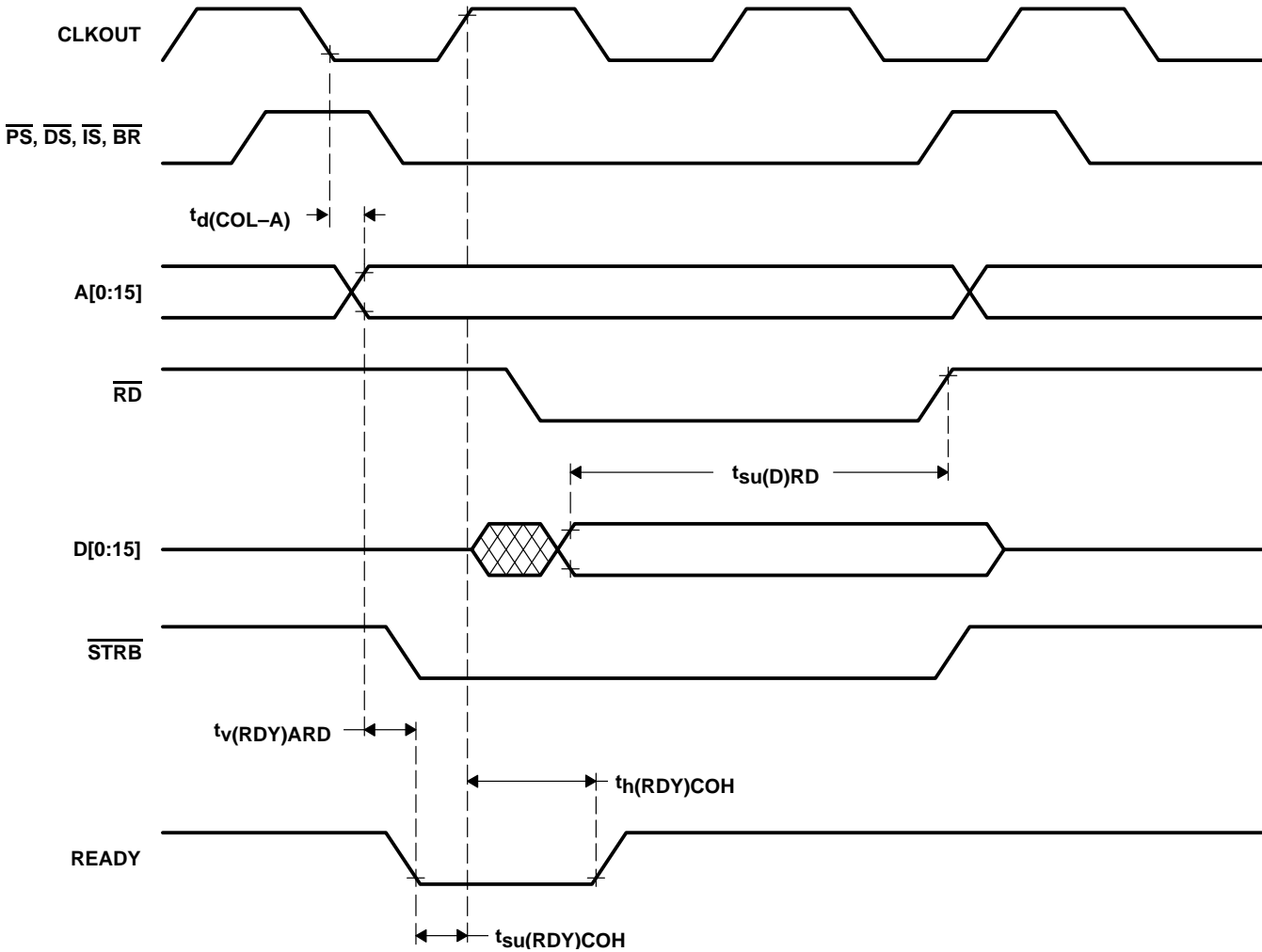


Figure 42. Ready-on-Read Timings

external memory interface ready-on-write timings

switching characteristics over recommended operating conditions for an external memory interface ready-on-write (see Figure 43)

PARAMETER		MIN	MAX	UNIT
$t_d(\text{COH-A})$	Delay time, address valid after CLKOUT high			ns
$t_h(\text{RDY})\text{COH}$	Hold time, READY after CLKOUT high			ns

timing requirements over recommended operating conditions for an external memory interface ready-on-write (see Figure 43)

		MIN	MAX	UNIT
$t_{su(D)W}$	Setup time, write data before $\overline{\text{WR}}$ strobe inactive high			ns
$t_v(\text{RDY})\text{AW}$	Valid time, READY after address valid on write			ns
$t_{su(\text{RDY})}\text{COH}$	Setup time, READY before CLKOUT high			ns

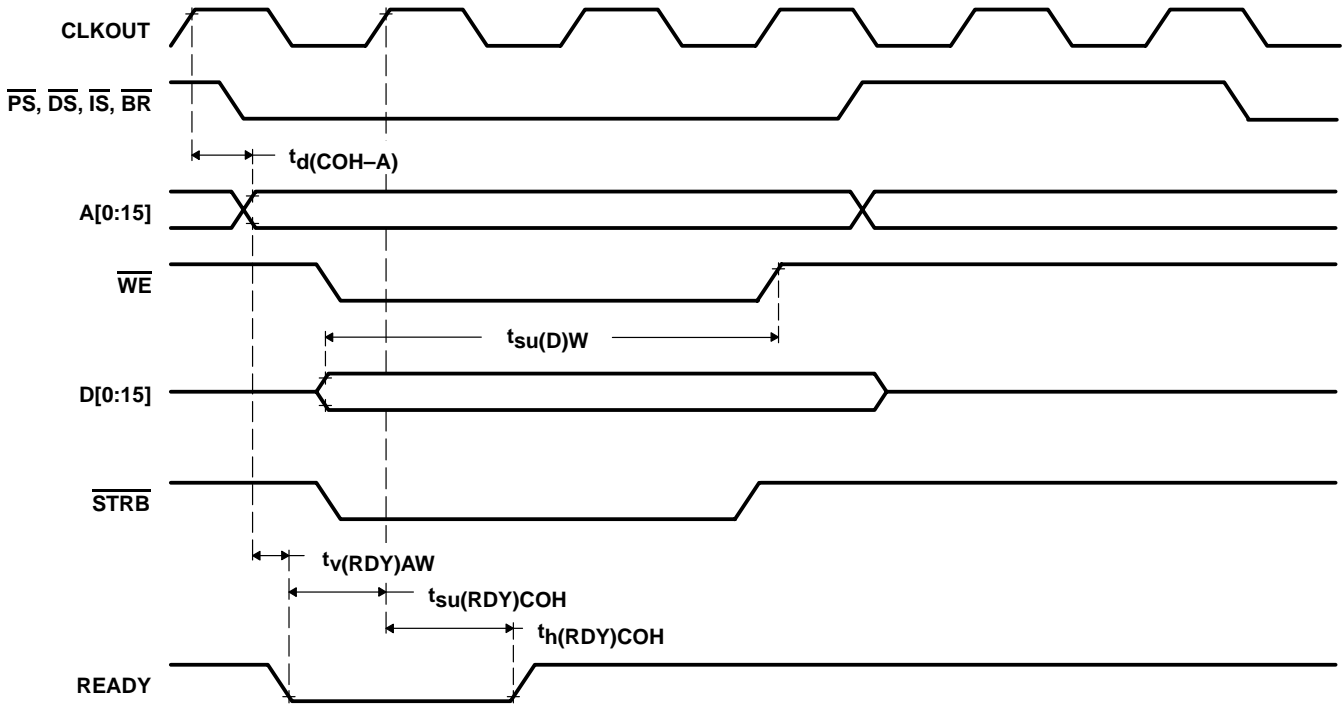


Figure 43. Ready-on-Write Timings

PRODUCT PREVIEW

TMS320F243  
DSP CONTROLLER

SPRS064 – DECEMBER 1997

CPU core miscellaneous timings

switching characteristics over recommended operating conditions for the CPU core  
(see Figure 44)

PARAMETER		MIN	MAX	UNIT
$t_d(XF)$	Delay time, XF high/low after CLKOUT high			ns

timing requirements over recommended operating conditions for the CPU core (see Figure 44)

		MIN	MAX	UNIT
$t_w(BIOL)$	Pulse duration, $\overline{BIO}$ low			ns
$t_w(MPMC)$	Pulse duration, $MP/\overline{MC}$ valid time (high or low)			ns

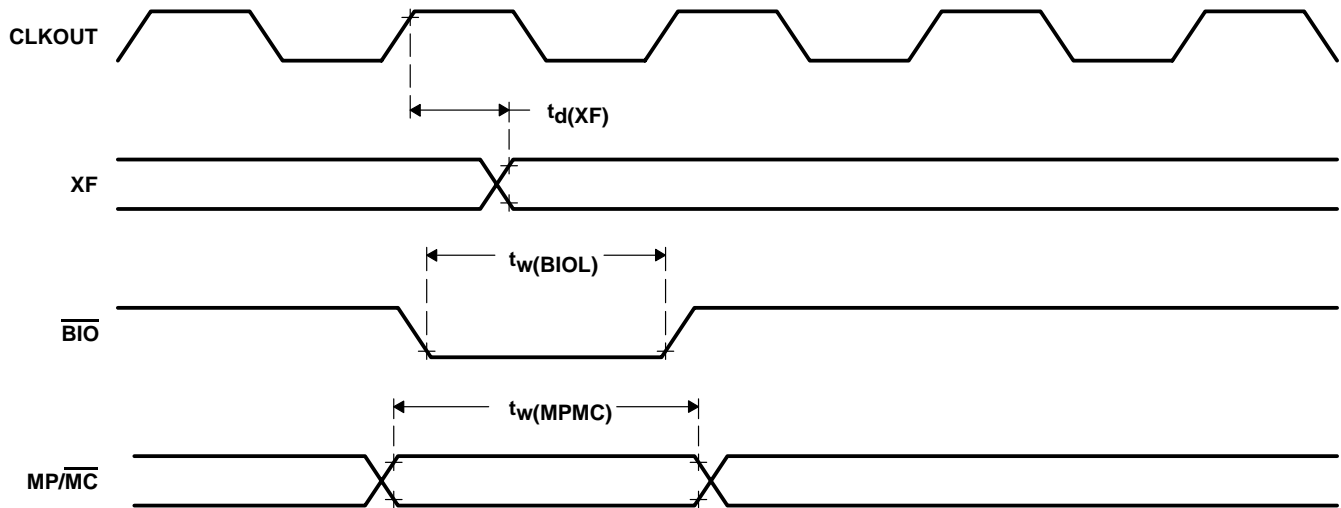


Figure 44. CPU Core Miscellaneous Timings

10-bit dual analog-to-digital converter (ADC)

The 10-bit ADC has a separate power bus for its analog circuitry. These pins are referred to as  $V_{CCA}$  and  $V_{SSA}$ . The purpose is to enhance ADC performance by preventing digital switching noise of the logic circuitry that can be present on  $V_{SS}$  and  $V_{CC}$  from coupling into the ADC analog stage. All ADC specifications are given with respect to  $V_{SSA}$  unless otherwise noted.

Resolution ..... 10-bit (1024 values)  
Monotonic ..... Assured  
Output conversion mode ..... 000h to 3FFh (000h for  $V_I \leq V_{SSA}$ ; 3FFh for  $V_I \geq V_{CCA}$ )  
Conversion time (including sample time) .....  $60t_{c(SYS)}$

recommended operating conditions

		MIN	NOM	MAX	UNIT
$V_{CCA}$	Analog supply voltage	4.5	5	5.5	V
$V_{SSA}$	Analog ground		0		V
$V_{REFHI}$	Analog supply reference source†	$V_{REFLO}^{\ddagger}$		$V_{CCA}$	V
$V_{REFLO}$	Analog ground reference source†	$V_{SSA}$		$V_{REFHI}^{\ddagger}$	V
$V_{AI}$	Analog input voltage, ADCIN00–ADCIN07	$V_{SSA}$		$V_{CCA}$	V

†  $V_{REFHI}$  and  $V_{REFLO}$  must be stable, within  $\pm 1/2$  LSB of the required resolution, during the entire conversion time.

‡ Values derived from characterization data and not tested.

# TMS320F243 DSP CONTROLLER

SPRS064 – DECEMBER 1997

## operating characteristics over recommended operating condition ranges†

PARAMETER		DESCRIPTION	MIN	MAX	UNIT
I <sub>CCA</sub>	Analog supply current	V <sub>CCA</sub> = 5.5 V		5	mA
		converting		2	
		V <sub>CCA</sub> = V <sub>REFHI</sub> = 5.5 V		1	
I <sub>ref</sub>	Input charge current, V <sub>REFHI</sub> or V <sub>REFLO</sub>	V <sub>CCA</sub> = V <sub>CCD</sub> = V <sub>REFHI</sub> = 5.5 V, V <sub>REFLO</sub> = 0 V		2.5	mA
C <sub>ai</sub>	Analog input capacitance	Typical capacitive load on analog input pin		6	pF
		non-sampling		8	
EDNL	Differential nonlinearity error	Difference between the actual step width and the ideal value		± 1.5	LSB
EINL	Integral nonlinearity error	Maximum deviation from the best straight line through the ADC transfer characteristics, excluding the quantization error		± 1.5	LSB
t <sub>d(PU)</sub>	Delay time, power-up to ADC valid	Time to stabilize analog stage after power-up		10	μs

† Absolute resolution = 4.89 mV. At V<sub>REFHI</sub> = 5 V and V<sub>REFLO</sub> = 0 V, this is one LSB. As V<sub>REFHI</sub> decreases, V<sub>REFLO</sub> increases, or both, the LSB size decreases. Therefore, the absolute accuracy and differential/integral linearity errors in terms of LSBs increase.

The ADC module allows complete freedom in the design of the sources for the analog inputs. The period of the sample time is independent of the source impedance. The sample-and-hold period occurs in the first half period of the ADC clock after the ADCIMSTART bit or the ADCSOC bit of the ADC control register 1 (ADCTRL1, bits 13 and 0, respectively) is set to 1. The conversion then occurs during the next six ADC clock cycles. The digital result registers are updated on the next ADC clock cycle once the conversion is completed.

## register file compilation

Table 18 is a collection of all the programmable registers of the TMS320x24x (provided for a quick reference).

**Table 18. Register File Compilation**

ADDR	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	REG
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	
DATA MEMORY SPACE									
CPU STATUS REGISTERS									
	ARP			OV	OVM	1	INTM	DP(8)	ST0
	DP(7)	DP(6)	DP(5)	DP(4)	DP(3)	DP(2)	DP(1)	DP(0)	
	ARB			CNF	TC	SXM	C	1	ST1
	1	1	1	XF	1	1	PM		
GLOBAL MEMORY AND CPU INTERRUPT REGISTERS									
00004h	—	—	—	—	—	—	—	—	IMR
	—	—	INT6 MASK	INT5 MASK	INT4 MASK	INT3 MASK	INT2 MASK	INT1 MASK	
00005h	—	—	—	—	—	—	—	—	GREG
	Global Data Memory Configuration Bits (7–0)								
00006h	—	—	—	—	—	—	—	—	IFR
	—	—	INT6 FLAG	INT5 FLAG	INT4 FLAG	INT3 FLAG	INT2 FLAG	INT1 FLAG	
SYSTEM CONFIGURATION REGISTERS									
07018h	—	CLKSRC	LPM1	LPM0	—	—	—	—	SCSR
	—	—	—	—	—	—	—	ILLADR	
07019h	Reserved								
0701Ah	PORST	—	—	ILLADR	—	SWRST	WDRST	—	SYSSR
	—	—	HPO	—	VCCAOR	—	—	VECRD	
0701Bh to 0701Dh	Reserved								
0701Eh	V15	V14	V13	V12	V11	V10	V9	V8	PIVR
	V7	V6	V5	V4	V3	V2	V1	V0	
0701Fh	Reserved								
WD CONTROL REGISTERS									
07020h to 07022h	Reserved								
07023h	D7	D6	D5	D4	D3	D2	D1	D0	WDCNTR
07024h	Reserved								
07025h	D7	D6	D5	D4	D3	D2	D1	D0	WDKEY
07026h to 07028h	Reserved								
07029h	WD FLAG	WDDIS	WDCHK2	WDCHK1	WDCHK0	WDPS2	WDPS1	WDPS0	WDCR
PLL CLOCK CONTROL REGISTERS									
0702Ah	Reserved								
0702Bh	CLKMD(1)	CLKMD(0)	PLLOCK(1)	PLLOCK(0)	PLLPM(1)	PLLPM(0)	ACLKENA	PLLPS	CKCR0
0702Ch	Reserved								

**register file compilation (continued)**

**Table 18. Register File Compilation (Continued)**

ADDR	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	REG
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	
PLL CLOCK CONTROL REGISTERS (CONTINUED)									
0702Dh	CKINF(3)	CKINF(2)	CKINF(1)	CKINF(0)	PLLDIV(2)	PLLFB(2)	PLLFB(1)	PLLFB(0)	CKCR1
0702Eh to 07031h	Reserved								
A-to-D MODULE CONTROL REGISTERS									
07032h	SUSPEND-SOFT	SUSPEND-FREE	ADCIM-START	ADC2EN	ADC1EN	ADCCON-RUN	ADCINTEN	ADCINTFLAG	ADCTRL1
	ADCEOC	ADC2CHSEL			ADC1CHSEL			ADCSOC	
07033h	Reserved								
07034h	—	—	EVSOC	EXTSOC	INTPRI	ADCEVSOC	ADCEXTSOC	—	ADCTRL2
	ADCFIFO2		—	ADCFIFO1		ADCPSCALE			
07035h	Reserved								
07036h	D9	D8	D7	D6	D5	D4	D3	D2	ADCFIFO1
	D1	D0	0	0	0	0	0	0	
07037h	Reserved								
07038h	D9	D8	D7	D6	D5	D4	D3	D2	ADCFIFO2
	D1	D0	0	0	0	0	0	0	
07039h to 0703Fh	Reserved								
SERIAL PERIPHERAL INTERFACE (SPI) CONFIGURATION CONTROL REGISTERS									
07040h	SPI SW RESET	CLOCK POLARITY	—	—	SPI CHAR3	SPI CHAR2	SPI CHAR1	SPI CHAR0	SPICCR
07041h	—	—	—	OVERRUN INT ENA	CLOCK PHASE	MASTER/SLAVE	TALK	SPI INT ENA	SPICTL
07042h	RECEIVER OVERRUN FLAG	SPI INT FLAG	TX BUF FULL FLAG	—	—	—	—	—	SPISTS
07043h	Reserved								
07044h	—	SPI BIT RATE 6	SPI BIT RATE 5	SPI BIT RATE 4	SPI BIT RATE 3	SPI BIT RATE 2	SPI BIT RATE 1	SPI BIT RATE 0	SPIBRR
07045h	Reserved								
07046h	ERXB15	ERXB14	ERXB13	ERXB12	ERXB11	ERXB10	ERXB9	ERXB8	SPIRXEMU
	ERXB7	ERXB6	ERXB5	ERXB4	ERXB3	ERXB2	ERXB1	ERXB0	
07047h	RXB15	RXB14	RXB13	RXB12	RXB11	RXB10	RXB9	RXB8	SPIRXBUF
	RXB7	RXB6	RXB5	RXB4	RXB3	RXB2	RXB1	RXB0	
07048h	TXB15	TXB14	TXB13	TXB12	TXB11	TXB10	TXB9	TXB8	SPITXBUF
	TXB7	TXB6	TXB5	TXB4	TXB3	TXB2	TXB1	TXB0	
07049h	SDAT15	SDAT14	SDAT13	SDAT12	SDAT11	SDAT10	SDAT9	SDAT8	SPIDAT
	SDAT7	SDAT6	SDAT5	SDAT4	SDAT3	SDAT2	SDAT1	SDAT0	
0704Ah to 0704Ch	Reserved								



**register file compilation (continued)**

**Table 18. Register File Compilation (Continued)**

ADDR	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	REG
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	
SERIAL PERIPHERAL INTERFACE (SPI) CONFIGURATION CONTROL REGISTERS (CONTINUED)									
0704Dh	SPISTE DATA IN	SPISTE DATA OUT	SPISTE FUNCTION	SPISTE DATA DIR	—	—	—	—	SPIPC1
0704Eh	SPISIMO DATA IN	SPISIMO DATA OUT	SPISIMO FUNCTION	SPISIMO DATA DIR	SPISOMI DATA IN	SPISOMI DATA OUT	SPISOMI FUNCTION	SPISOMI DATA DIR	SPIPC2
0704Fh	—	SPI PRIORITY	SPI SUSP SOFT	SPI SUSP FREE	—	—	—	—	SPIPRI
SERIAL COMMUNICATIONS INTERFACE (SCI) CONFIGURATION CONTROL REGISTERS									
07050h	STOP BITS	EVEN/ODD PARITY	PARITY ENABLE	LOOP BACK ENA	ADDR/IDLE MODE	SCI CHAR2	SCI CHAR1	SCI CHAR0	SCICCR
07051h	—	RX ERR INT ENA	SW RESET	—	TXWAKE	SLEEP	TXENA	RXENA	SCICTL1
07052h	BAUD15 (MSB)	BAUD14	BAUD13	BAUD12	BAUD11	BAUD10	BAUD9	BAUD8	SCIHBAUD
07053h	BAUD7	BAUD6	BAUD5	BAUD4	BAUD3	BAUD2	BAUD1	BAUD0 (LSB)	SCILBAUD
07054h	TXRDY	TX EMPTY	—	—	—	—	RX/BK INT ENA	TX INT ENA	SCICTL2
07055h	RX ERROR	RXRDY	BRKDT	FE	OE	PE	RXWAKE	—	SCIRXST
07056h	ERXDT7	ERXDT6	ERXDT5	ERXDT4	ERXDT3	ERXDT2	ERXDT1	ERXDT0	SCIRXEMU
07057h	RXDT7	RXDT6	RXDT5	RXDT4	RXDT3	RXDT2	RXDT1	RXDT0	SCIRXBUF
07058h	Reserved								
07059h	TXDT7	TXDT6	TXDT5	TXDT4	TXDT3	TXDT2	TXDT1	TXDT0	SCITXBUF
0705Ah to 0705Dh	Reserved								
0705Eh	SCITXD DATA IN	SCITXD DATA OUT	SCITXD FUNCTION	SCITXD DATA DIR	SCIRXD DATA IN	SCIRXD DATA OUT	SCIRXD FUNCTION	SCIRXD DATA DIR	SCIPC2
0705Fh	—	SCITX PRIORITY	SCIRX PRIORITY	SCI SUSP SOFT	SCI SUSP FREE	—	—	—	SCIPRI
07060h to 0706Fh	Reserved								
EXTERNAL INTERRUPT CONTROL REGISTERS									
07070h	XINT1 FLAG	—	—	—	—	—	—	—	XINT1CR
	—	—	—	—	—	XINT1 POLARITY	XINT1 PRIORITY	XINT1 ENA	
07071h	XINT2 FLAG	—	—	—	—	—	—	—	XINT2CR
	—	—	—	—	—	XINT2 POLARITY	XINT2 PRIORITY	XINT2 ENA	
07072h to 0708Fh	Reserved								

**register file compilation (continued)**

**Table 18. Register File Compilation (Continued)**

ADDR	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	REG
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	
07090h	DIGITAL I/O CONTROL REGISTERS								OCRA
	CRA.15	CRA.14	CRA.13	CRA.12	CRA.11	CRA.10	CRA.9	CRA.8	
	CRA.7	CRA.6	CRA.5	CRA.4	CRA.3	CRA.2	CRA.1	CRA.0	
07091h	Reserved								OCRB
07092h	—	—	—	—	—	—	CRB.9	CRB.8	
	CRB.7	CRB.6	CRB.5	CRB.4	CRB.3	CRB.2	CRB.1	CRB.0	
07093h to 07097h	Reserved								PADATDIR
07098h	A7DIR	A6DIR	A5DIR	A4DIR	A3DIR	A2DIR	A1DIR	A0DIR	
	IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0	
07099h	Reserved								PBDATDIR
0709Ah	B7DIR	B6DIR	B5DIR	B4DIR	B3DIR	B2DIR	B1DIR	B0DIR	
	IOPB7	IOPB6	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0	
0709Bh	Reserved								PCDATDIR
0709Ch	C7DIR	C6DIR	C5DIR	C4DIR	C3DIR	C2DIR	C1DIR	C0DIR	
	IOPC7	IOPC6	IOPC5	IOPC4	IOPC3	IOPC2	IOPC1	IOPC0	
0709Dh	Reserved								PDDATDIR
0709Eh	—	—	—	—	—	—	D1DIR	D0DIR	
	—	—	—	—	—	—	IOPD1	IOPD0	
0709Fh	Reserved								MDER
CONTROLLER AREA NETWORK (CAN) CONFIGURATION CONTROL REGISTERS									
07100h	—	—	—	—	—	—	—	—	
	MD3	MD2	ME5	ME4	ME3	ME2	ME1	ME0	
07101h	TA5	TA4	TA3	TA2	AA5	AA4	AA3	AA2	TCR
	TRS5	TRS4	TRS3	TRS2	TRR5	TRR4	TRR3	TRR2	
07102h	RFP3	RFP2	RFP1	RFP0	RML3	RML2	RML1	RML0	RCR
	RMP3	RMP2	RMP1	RMP0	OPC3	OPC2	OPC1	OPC0	
07103h	—	—	SUSP	CCR	PDR	DBO	WUBA	CDR	MCR
	ABO	STM	—	—	—	—	MBNR1	MBNR0	
07104h	—	—	—	—	—	—	—	—	BCR2
	BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	
07105h	—	—	—	—	—	SBG	SJW1	SJW0	BCR1
	SAM	TSEG1–3	TSEG1–2	TSEG1–1	TSEG1–0	TSEG2–2	TSEG2–1	TSEG2–0	
07106h	—	—	—	—	—	—	—	FER	ESR
	BEF	SA1	CRCE	SER	ACKE	BO	EP	EW	
07107h	—	—	—	—	—	—	—	—	GSR
	—	—	SMA	CCE	PDA	—	RM	TM	
07108h	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	CEC
	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	
07109h	—	—	MIF5	MIF4	MIF3	MIF2	MIF1	MIF0	IFR
	—	RMLIF	AAIF	WDIF	WUIF	BOIF	EPIF	WLIF	

**register file compilation (continued)**

**Table 18. Register File Compilation (Continued)**

ADDR	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	REG
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	
	CONTROLLER AREA NETWORK (CAN) CONFIGURATION CONTROL REGISTERS (CONTINUED)								
0710Ah	MIL	—	MIM5	MIM4	MIM3	MIM2	MIM1	MIM0	IMR
	EIL	RMLIM	AAIM	WDIM	WUIM	BOIM	EPIM	WLIM	
0710Bh	LAMI	—	—	LAM0–28	LAM0–27	LAM0–26	LAM0–25	LAM0–24	LAM0_H
	LAM0–23	LAM0–22	LAM0–21	LAM0–20	LAM0–19	LAM0–18	LAM0–17	LAM0–16	
0710Ch	LAM0–15	LAM0–14	LAM0–13	LAM0–12	LAM0–11	LAM0–10	LAM0–9	LAM0–8	LAM0_L
	LAM0–7	LAM0–6	LAM0–5	LAM0–4	LAM0–3	LAM0–2	LAM0–1	LAM0–0	
0710Dh	LAMI	—	—	LAM1–28	LAM1–27	LAM1–26	LAM1–25	LAM1–24	LAM1_H
	LAM1–23	LAM1–22	LAM1–21	LAM1–20	LAM1–19	LAM1–18	LAM1–17	LAM1–16	
0710Eh	LAM1–15	LAM1–14	LAM1–13	LAM1–12	LAM1–11	LAM1–10	LAM1–9	LAM1–8	LAM1_L
	LAM1–7	LAM1–6	LAM1–5	LAM1–4	LAM1–3	LAM1–2	LAM1–1	LAM1–0	
0F10Fh to 0F1FFh	Reserved								
	Message Object #0								
07200h	IDL–15	IDL–14	IDL–13	IDL–12	IDL–11	IDL–10	IDL–9	IDL–8	MSGID0L
	IDL–7	IDL–6	IDL–5	IDL–4	IDL–3	IDL–2	IDL–1	IDL–0	
07201h	IDE	AME	AAM	IDH–28	IDH–27	IDH–26	IDH–25	IDH–24	MSGID0H
	IDH–23	IDH–22	IDH–21	IDH–20	IDH–19	IDH–18	IDH–17	IDH–16	
07202h	—	—	—	—	—	—	—	—	MSGCTRL0
	—	—	—	RTR	DLC3	DLC2	DLC1	DLC0	
07203h	Reserved								
07204h	0	0	0	0	0	0	0	0	MBX0A
	D7	D6	D5	D4	D3	D2	D1	D0	
07205h	0	0	0	0	0	0	0	0	MBX0B
	D7	D6	D5	D4	D3	D2	D1	D0	
07206h	0	0	0	0	0	0	0	0	MBX0C
	D7	D6	D5	D4	D3	D2	D1	D0	
07207h	0	0	0	0	0	0	0	0	MBX0D
	D7	D6	D5	D4	D3	D2	D1	D0	
	Message Object #1								
07208h	IDL–15	IDL–14	IDL–13	IDL–12	IDL–11	IDL–10	IDL–9	IDL–8	MSGID1L
	IDL–7	IDL–6	IDL–5	IDL–4	IDL–3	IDL–2	IDL–1	IDL–0	
07209h	IDE	AME	AAM	IDH–28	IDH–27	IDH–26	IDH–25	IDH–24	MSGID1H
	IDH–23	IDH–22	IDH–21	IDH–20	IDH–19	IDH–18	IDH–17	IDH–16	
0720Ah	—	—	—	—	—	—	—	—	MSGCTRL1
	—	—	—	RTR	DLC3	DLC2	DLC1	DLC0	
0720Bh	Reserved								
0720Ch	0	0	0	0	0	0	0	0	MBX1A
	D7	D6	D5	D4	D3	D2	D1	D0	
0720Dh	0	0	0	0	0	0	0	0	MBX1B
	D7	D6	D5	D4	D3	D2	D1	D0	

**register file compilation (continued)**

**Table 18. Register File Compilation (Continued)**

ADDR	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	REG
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	
CONTROLLER AREA NETWORK (CAN) CONFIGURATION CONTROL REGISTERS (CONTINUED)									
0720Eh	0	0	0	0	0	0	0	0	MBX1C
	D7	D6	D5	D4	D3	D2	D1	D0	
0720Fh	0	0	0	0	0	0	0	0	MBX1D
	D7	D6	D5	D4	D3	D2	D1	D0	
Message Object #2									
07210h	IDL-15	IDL-14	IDL-13	IDL-12	IDL-11	IDL-10	IDL-9	IDL-8	MSGID2L
	IDL-7	IDL-6	IDL-5	IDL-4	IDL-3	IDL-2	IDL-1	IDL-0	
07211h	IDE	AME	AAM	IDH-28	IDH-27	IDH-26	IDH-25	IDH-24	MSGID2H
	IDH-23	IDH-22	IDH-21	IDH-20	IDH-19	IDH-18	IDH-17	IDH-16	
07212h	—	—	—	—	—	—	—	—	MSGCTRL2
	—	—	—	RTR	DLC3	DLC2	DLC1	DLC0	
07213h	Reserved								
07214h	0	0	0	0	0	0	0	0	MBX2A
	D7	D6	D5	D4	D3	D2	D1	D0	
07215h	0	0	0	0	0	0	0	0	MBX2B
	D7	D6	D5	D4	D3	D2	D1	D0	
07216h	0	0	0	0	0	0	0	0	MBX2C
	D7	D6	D5	D4	D3	D2	D1	D0	
07217h	0	0	0	0	0	0	0	0	MBX2D
	D7	D6	D5	D4	D3	D2	D1	D0	
Message Object #3									
07218h	IDL-15	IDL-14	IDL-13	IDL-12	IDL-11	IDL-10	IDL-9	IDL-8	MSGID3L
	IDL-7	IDL-6	IDL-5	IDL-4	IDL-3	IDL-2	IDL-1	IDL-0	
07219h	IDE	AME	AAM	IDH-28	IDH-27	IDH-26	IDH-25	IDH-24	MSGID3H
	IDH-23	IDH-22	IDH-21	IDH-20	IDH-19	IDH-18	IDH-17	IDH-16	
0721Ah	—	—	—	—	—	—	—	—	MSGCTRL3
	—	—	—	RTR	DLC3	DLC2	DLC1	DLC0	
0721Bh	Reserved								
0721Ch	0	0	0	0	0	0	0	0	MBX3A
	D7	D6	D5	D4	D3	D2	D1	D0	
0721Dh	0	0	0	0	0	0	0	0	MBX3B
	D7	D6	D5	D4	D3	D2	D1	D0	
0721Eh	0	0	0	0	0	0	0	0	MBX3C
	D7	D6	D5	D4	D3	D2	D1	D0	
0721Fh	0	0	0	0	0	0	0	0	MBX3D
	D7	D6	D5	D4	D3	D2	D1	D0	
Message Object #4									
07220h	IDL-15	IDL-14	IDL-13	IDL-12	IDL-11	IDL-10	IDL-9	IDL-8	MSGID4L
	IDL-7	IDL-6	IDL-5	IDL-4	IDL-3	IDL-2	IDL-1	IDL-0	
07221h	IDE	AME	AAM	IDH-28	IDH-27	IDH-26	IDH-25	IDH-24	MSGID4H
	IDH-23	IDH-22	IDH-21	IDH-20	IDH-19	IDH-18	IDH-17	IDH-16	

**register file compilation (continued)**

**Table 18. Register File Compilation (Continued)**

ADDR	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	REG
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	
	CONTROLLER AREA NETWORK (CAN) CONFIGURATION CONTROL REGISTERS (CONTINUED)								
07222h	—	—	—	—	—	—	—	—	MSGCTRL4
	—	—	—	RTR	DLC3	DLC2	DLC1	DLC0	
07223h	Reserved								
07224h	0	0	0	0	0	0	0	0	MBX4A
	D7	D6	D5	D4	D3	D2	D1	D0	
07225h	0	0	0	0	0	0	0	0	MBX4B
	D7	D6	D5	D4	D3	D2	D1	D0	
07226h	0	0	0	0	0	0	0	0	MBX4C
	D7	D6	D5	D4	D3	D2	D1	D0	
07227h	0	0	0	0	0	0	0	0	MBX4D
	D7	D6	D5	D4	D3	D2	D1	D0	
07228h	Message Object #5								
	IDL-15	IDL-14	IDL-13	IDL-12	IDL-11	IDL-10	IDL-9	IDL-8	MSGID5L
	IDL-7	IDL-6	IDL-5	IDL-4	IDL-3	IDL-2	IDL-1	IDL-0	
07229h	IDE	AME	AAM	IDH-28	IDH-27	IDH-26	IDH-25	IDH-24	MSGID5H
	IDH-23	IDH-22	IDH-21	IDH-20	IDH-19	IDH-18	IDH-17	IDH-16	
0722Ah	—	—	—	—	—	—	—	—	MSGCTRL5
	—	—	—	RTR	DLC3	DLC2	DLC1	DLC0	
0722Bh	Reserved								
0722Ch	0	0	0	0	0	0	0	0	MBX5A
	D7	D6	D5	D4	D3	D2	D1	D0	
0722Dh	0	0	0	0	0	0	0	0	MBX5B
	D7	D6	D5	D4	D3	D2	D1	D0	
0722Eh	0	0	0	0	0	0	0	0	MBX5C
	D7	D6	D5	D4	D3	D2	D1	D0	
0722Fh	0	0	0	0	0	0	0	0	MBX5D
	D7	D6	D5	D4	D3	D2	D1	D0	
07230h to 073FFh	Reserved								
07400h	GENERAL-PURPOSE (GP) TIMER CONFIGURATION CONTROL REGISTERS								
	—	T2STAT	T1STAT	—		T2TOADC		T1TOADC(1)	GPTCON
	T1TOADC(0)	TCOMPOE	T3PIN		T2PIN		T1PIN		
07401h	D15	D14	D13	D12	D11	D10	D9	D8	T1CNT
	D7	D6	D5	D4	D3	D2	D1	D0	
07402h	D15	D14	D13	D12	D11	D10	D9	D8	T1CMPR
	D7	D6	D5	D4	D3	D2	D1	D0	
07403h	D15	D14	D13	D12	D11	D10	D9	D8	T1PR
	D7	D6	D5	D4	D3	D2	D1	D0	
07404h	FREE	SOFT	—	TMODE1	TMODE0	TPS2	TPS1	TPS0	T1CON
	TSWT1	TENABLE	TCLKS1	TCLKS0	TCLD1	TCLD0	TECMPR	SELT1PR	
07405h	D15	D14	D13	D12	D11	D10	D9	D8	T2CNT
	D7	D6	D5	D4	D3	D2	D1	D0	

**register file compilation (continued)**

**Table 18. Register File Compilation (Continued)**

ADDR	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	REG
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	
07406h	GENERAL-PURPOSE (GP) TIMER CONFIGURATION CONTROL REGISTERS (CONTINUED)								T2CMPR
	D15	D14	D13	D12	D11	D10	D9	D8	
	D7	D6	D5	D4	D3	D2	D1	D0	
07407h	D15	D14	D13	D12	D11	D10	D9	D8	T2PR
	D7	D6	D5	D4	D3	D2	D1	D0	
07408h	FREE	SOFT	—	TMODE1	TMODE0	TPS2	TPS1	TPS0	T2CON
	TSWT1	TENABLE	TCLKS1	TCLKS0	TCLD1	TCLD0	TECMPR	SELT1PR	
07409h to 07410h	Reserved								
07411h	FULL AND SIMPLE COMPARE UNIT REGISTERS								COMCON
	CENABLE	CLD1	CLD0	SVENABLE	ACTRLD1	ACTRLD0	FCOMPOE	—	
	—	—	—	—	—	—	—	—	
07412h	Reserved								
07413h	SVRDIR	D2	D1	D0	CMP6ACT1	CMP6ACT0	CMP5ACT1	CMP5ACT0	ACTR
	CMP4ACT1	CMP4ACT0	CMP3ACT1	CMP3ACT0	CMP2ACT1	CMP2ACT0	CMP1ACT1	CMP1ACT0	
07414h	—	—	—	—	—	—	—	—	SACTR
	—	—	SCMP3- ACT1	SCMP3- ACT0	SCMP2- ACT1	SCMP2- ACT0	SCMP1- ACT1	SCMP1- ACT0	
07415h	—	—	—	—	DBT3	DBT2	DBT1	DBT0	DBTCON
	EDBT3	EDBT2	EDBT1	DBTPS2	DBTPS1	DBTPS0	—	—	
07416h	Reserved								
07417h	D15	D14	D13	D12	D11	D10	D9	D8	CMPR1
	D7	D6	D5	D4	D3	D2	D1	D0	
07418h	D15	D14	D13	D12	D11	D10	D9	D8	CMPR2
	D7	D6	D5	D4	D3	D2	D1	D0	
07419h	D15	D14	D13	D12	D11	D10	D9	D8	CMPR3
	D7	D6	D5	D4	D3	D2	D1	D0	
0741Ah	D15	D14	D13	D12	D11	D10	D9	D8	SCMPR1
	D7	D6	D5	D4	D3	D2	D1	D0	
0741Bh	D15	D14	D13	D12	D11	D10	D9	D8	SCMPR2
	D7	D6	D5	D4	D3	D2	D1	D0	
0741Ch	D15	D14	D13	D12	D11	D10	D9	D8	SCMPR3
	D7	D6	D5	D4	D3	D2	D1	D0	
0741Dh to 0741Fh	Reserved								
07420h	CAPTURE UNIT REGISTERS								CAPCON
	CAPRES	CAPQEPN		CAP3EN	—	CAP3TSEL	CAP12TSEL	CAP3TOADC	
	CAP1EDGE		CAP2EDGE		CAP3EDGE		—		
07421h	Reserved								
07422h	—		CAP3FIFO		CAP2FIFO		CAP1FIFO		CAPFIFO
	—	—	—	—	—	—	—	—	

**register file compilation (continued)**

**Table 18. Register File Compilation (Continued)**

ADDR	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	REG
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	
	CAPTURE UNIT REGISTERS (CONTINUED)								
07423h	D15	D14	D13	D12	D11	D10	D9	D8	CAP1FIFO
	D7	D6	D5	D4	D3	D2	D1	D0	
07424h	D15	D14	D13	D12	D11	D10	D9	D8	CAP2FIFO
	D7	D6	D5	D4	D3	D2	D1	D0	
07425h	D15	D14	D13	D12	D11	D10	D9	D8	CAP3FIFO
	D7	D6	D5	D4	D3	D2	D1	D0	
07426h	Reserved								
07427h									CAP1FBOT
07428h									CAP2FBOT
07429h									CAP3FBOT
0742Ah to 0742Bh	Reserved								
	EVENT MANAGER (EV) INTERRUPT CONTROL REGISTERS								
0742Ch	—	—	—	—	—	T1OFINT ENA	T1UFINT ENA	T1CINT ENA	EVIMRA
	T1PINT ENA	—	—	—	CMP3INT ENA	CMP2INT ENA	CMP1INT ENA	PDPINT ENA	
0742Dh	—	—	—	—	—	—	—	—	EVIMRB
	—	—	—	—	T2OFINT ENA	T2UFINT ENA	T2CINT ENA	T2PINT ENA	
0742Eh	—	—	—	—	—	—	—	—	EVIMRC
	—	—	—	—	—	CAP3INT ENA	CAP2INT ENA	CAP1INT ENA	
0742Fh	—	—	—	—	—	T1OFINT FLAG	T1UFINT FLAG	T1CINT FLAG	EVIFRA
	T1PINT FLAG	—	—	—	CMP3INT FLAG	CMP2INT FLAG	CMP1INT FLAG	PDPINT FLAG	
07430h	—	—	—	—	—	—	—	—	EVIFRB
	—	—	—	—	T2OFINT FLAG	T2UFINT FLAG	T2CINT FLAG	T2PINT FLAG	
07431h	—	—	—	—	—	—	—	—	EVIFRC
	—	—	—	—	—	CAP3INT FLAG	CAP2INT FLAG	CAP1INT FLAG	
07432h	0	0	0	0	0	0	0	0	EVIVRA
	0	0	D5	D4	D3	D2	D1	D0	
07433h	0	0	0	0	0	0	0	0	EVIVRB
	0	0	D5	D4	D3	D2	D1	D0	
07434h	0	0	0	0	0	0	0	0	EVIVRC
	0	0	D5	D4	D3	D2	D1	D0	
07435h to 0743Fh	Reserved								

**PRODUCT PREVIEW**

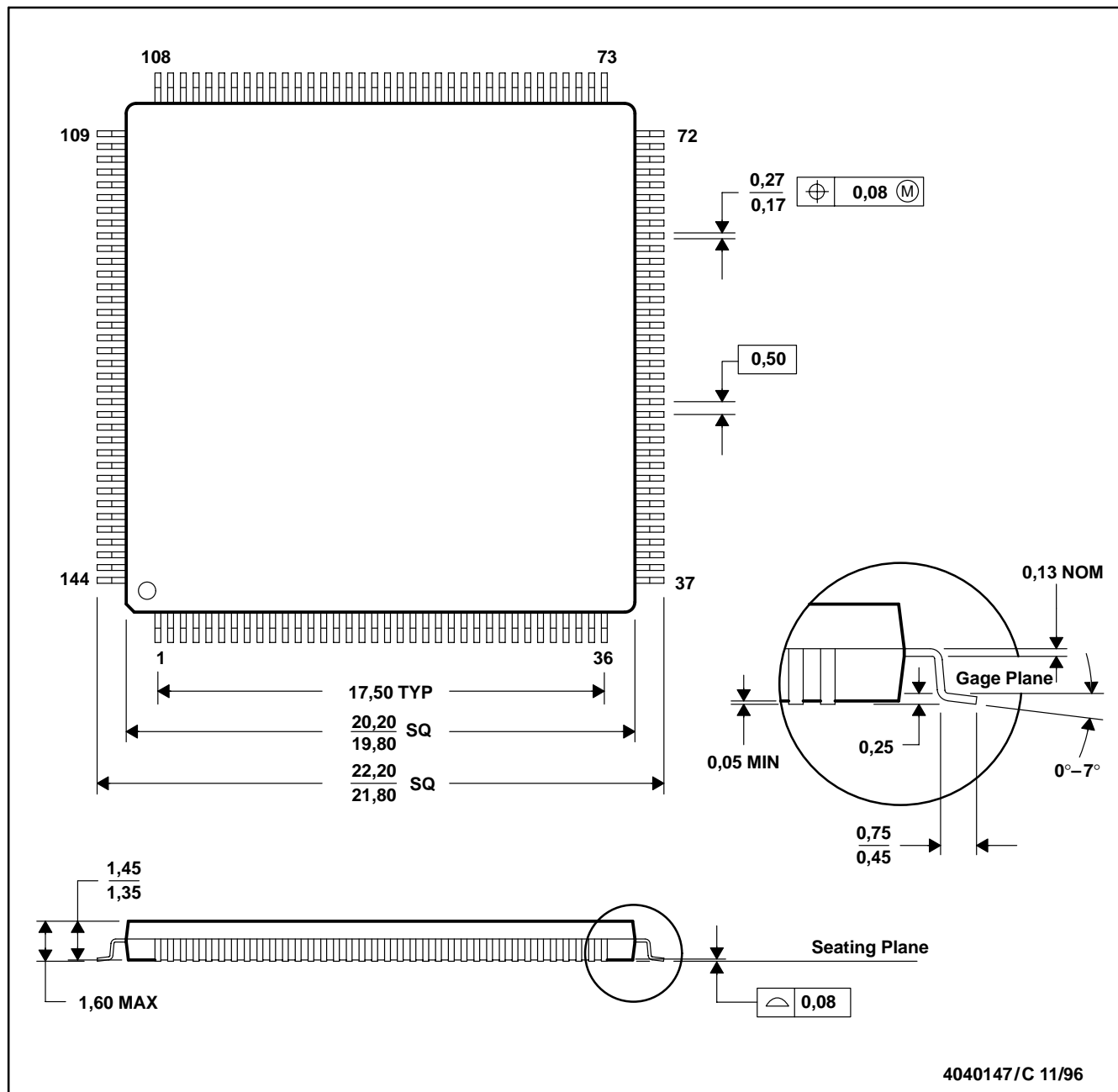
# TMS320F243 DSP CONTROLLER

SPRS064 – DECEMBER 1997

PGE (S-PQFP-G144)

PLASTIC QUAD FLATPACK

PRODUCT PREVIEW



- NOTES: A. All linear dimensions are in millimeters.  
B. This drawing is subject to change without notice.  
C. Falls within JEDEC MS-026



## **IMPORTANT NOTICE**

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

**TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.**

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.