- High-Performance Static CMOS Technology
- Includes the TMS320C2xx Core CPU
  - Object-Compatible With the TMS320C2xx
  - Source-Code-Compatible With TMS320C25
  - Upwardly Compatible With TMS320C5x™
  - 50-ns Instruction Cycle Time
- Commercial and Industrial Temperature Available
- Memory
  - 544 Words x 16 Bits of On-Chip Data/Program Dual-Access RAM (DARAM)
  - 8K Words x 16 Bits of Flash EEPROM
  - 224K Words x 16 Bits of Total Memory Address Reach (F243 only)
- External Memory Interface (F243 only)
- Event-Manager Module
  - Eight Compare/Pulse-Width Modulation (PWM) Channels
  - Two 16-Bit General-Purpose Timers With Four Modes, Including Continuous Up and Up/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
- 26 Individually Programmable, Multiplexed General-Purpose I/O (GPIO) Pins
- Six Dedicated GPIO Pins (F243 only)
- Phase-Locked-Loop (PLL)-Based Clock Module
- Watchdog (WD) Timer Module
- Serial Communications Interface (SCI) Module
- 16-Bit Serial Peripheral Interface (SPI) Module
- Five External Interrupts (Power Drive Protection, Reset, NMI, and Two Maskable Interrupts)
- Three 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 LQFP PGE Package (F243)
- 68-Pin PLCC FN Package (F241)
- 64-Pin QFP PG Package (F241)

description

The TMS320F243 and TMS320F241 devices are members of the 24x generation of digital signal processor (DSP) controllers based on the TMS320C2000™ platform of 16-bit fixed-point DSPs. The F243 is a superset of the F241. These two devices share similar core and peripherals with some exceptions. For example, the F241 does not have an external memory interface. This new family is optimized for digital motor/motion control applications. The DSP controllers combine the enhanced TMS320™ DSP family 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 µs.

Table of Contents

REVISION HISTORY

REVISION	DATE	PRODUCT STATUS	HIGHLIGHTS
C	September 2000	Production Data	The internal pullup/pulldown information has been corrected for the pins as appropriate. This information can be found in the “Terminal Functions” section of the data sheet. Errors in the F241 memory map have been rectified. Changes have been made to improve the clarity of the memory maps for reserved and illegal addresses. The functional diagram depicting the operation of the GPIO pins has been modified to enhance the understanding of pin behavior. Input and output clamp currents have been added to the “Absolute Maximum Ratings Over Operating Free-Air Temperature Range” table. The input currents (I_I) have been specified separately for pins with pullup and pulldown. The LPM0 current has been changed from 40 mA to 55 mA. The LPM2 current has been changed from 10 μA to 5 mA. The figure corresponding to “IDLE2 Entry and Exit Timing” for LPM1 mode has been removed. Note that the parameters are identical to LPM0. The “Reset Timing” figure now includes the state of the GPIO pins for “Power-on” reset.

device features

Table 1 and Table 2 provide a comparison of the features of the F243 and F241. See the functional block diagram for 24x peripherals and memory.

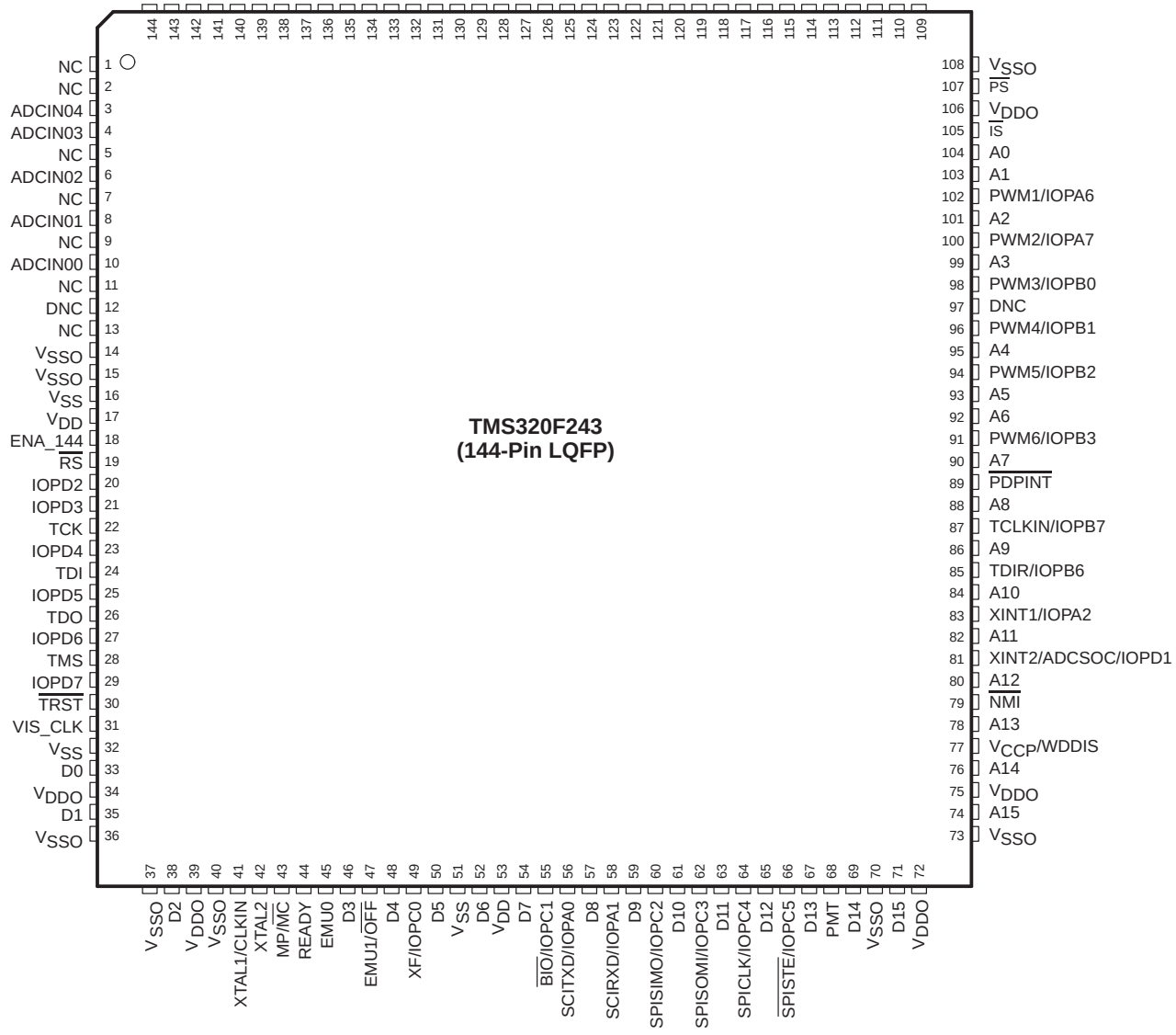
Table 1. Hardware Features of the TMS320F24x DSP Controllers

TMS320F24x DEVICES	ON-CHIP MEMORY (WORDS)		EXTERNAL MEMORY INTERFACE	POWER SUPPLY (V)	CYCLE TIME (ns)
	RAM				
	DATA SPACE	CONFIGURABLE DATA / PROG SPACE			
	(B1 RAM - 256 WORDS) (B2 RAM - 32 WORDS)	(B0 RAM)			
TMS320F243	288	256	√	5	50
TMS320F241			–		

Table 2. Device Specifications of the TMS320F24x DSP Controllers

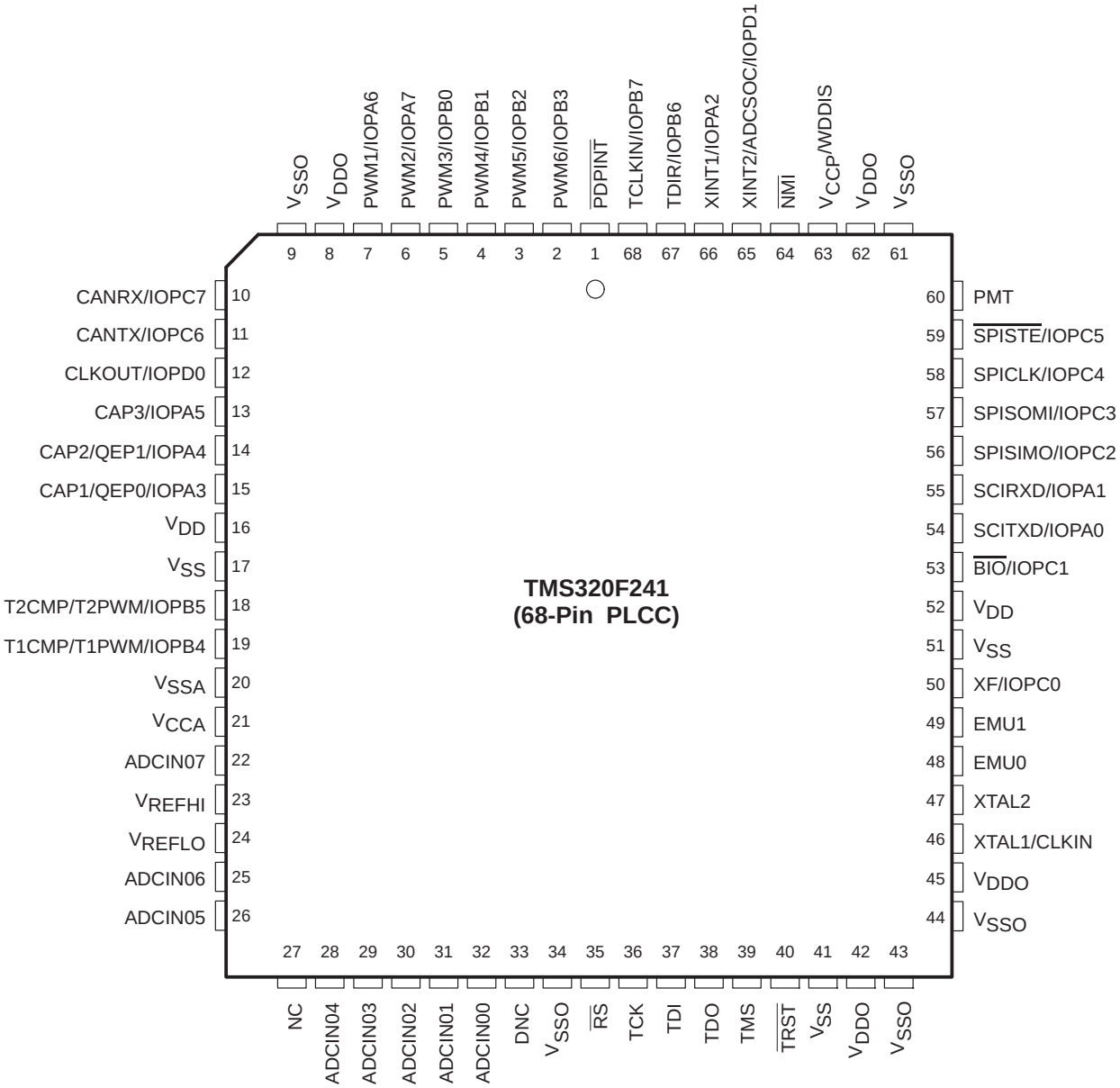
TMS320F24x DEVICES	ON-CHIP MEMORY (WORDS)		ADC CHANNELS	PERIPHERALS		GPIO	PACKAGE TYPE PIN COUNT
	ROM	FLASH EEPROM					
	PROG	PROG		CAN	SPI		
TMS320F243	–	8K	8	√	√	32	PGE 144-PQFP
TMS320F241	–	8K	8	√	√	26	FN 68-PLCC PG 64-PQFP

PGE PACKAGE†‡
(TOP VIEW)



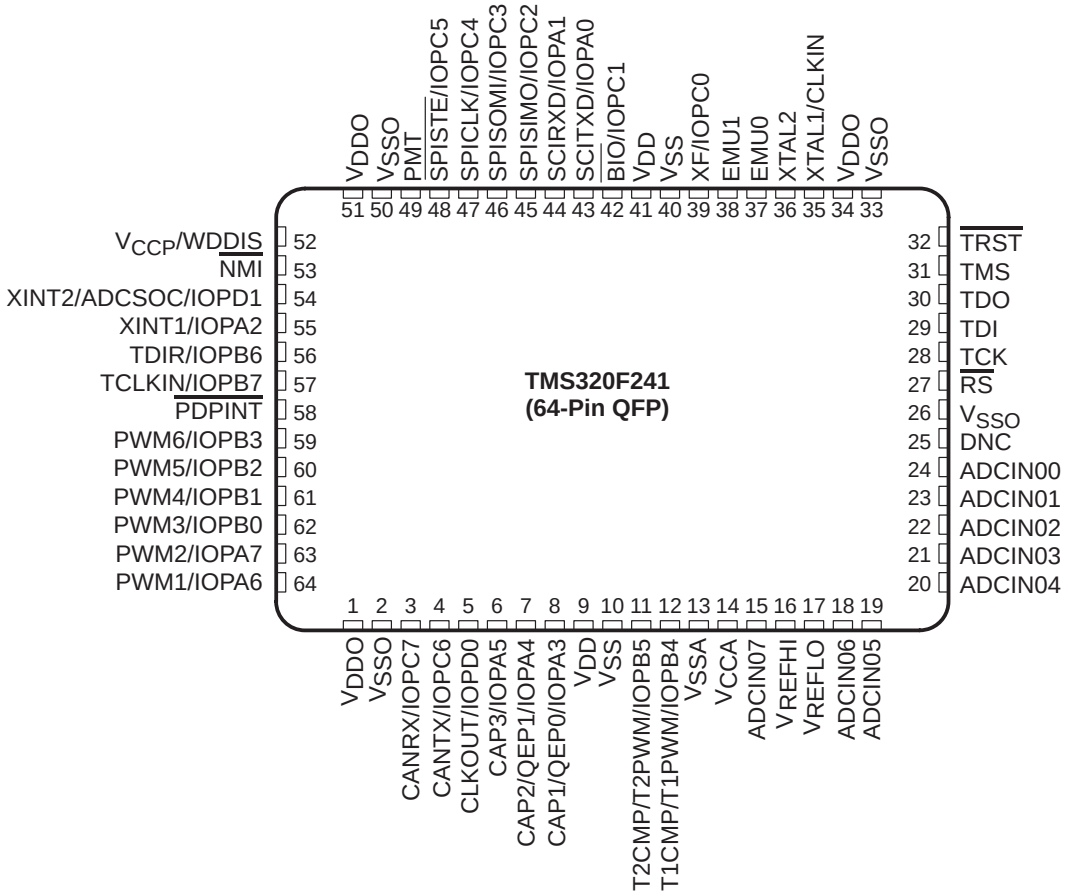
† NC = No connection, DNC = Do not connect
‡ The PMT pin, number 68 in this package drawing, must be connected to ground.

FN PACKAGE†‡
(TOP VIEW)



† NC = No connection, DNC = Do not connect
‡ The PMT pin, number 60 in this package drawing, must be connected to ground.

PG PACKAGE†‡
(TOP VIEW)



† NC = No connection, DNC = Do not connect
‡ The PMT pin, number 49 in this package drawing, must be connected to ground.

Terminal Functions - F243 PGE Package

NAME	144 LQFP NO.	TYPE†	RESET STATE‡	DESCRIPTION
ANALOG-TO-DIGITAL CONVERTER (ADC) INPUTS				
ADCIN00	10	I	I	Analog inputs to the ADC
ADCIN01	8			
ADCIN02	6			
ADCIN03	4			
ADCIN04	3			
ADCIN05	144			
ADCIN06	143			
ADCIN07	139			
V_{CCA}	137	–	–	Analog supply voltage for ADC (5 V). It is highly recommended to isolate V_{CCA} from the digital supply voltage (and V_{SSA} from digital ground) to maintain the specified accuracy and improve the noise immunity of the ADC.
V_{SSA}	135	–	–	Analog ground reference for ADC
V_{REFHI}	141	–	–	ADC analog high-voltage reference input
V_{REFLO}	142	–	–	ADC analog low-voltage reference input
EVENT MANAGER				
T1PWM/T1CMP/***IOPB4***	130	I/O/Z	I	Timer 1 compare output/general-purpose bidirectional digital I/O (GPIO).
T2PWM/T2CMP/***IOPB5***	128	I/O/Z	I	Timer 2 compare output/GPIO
TDIR/***IOPB6***	85	I/O	I	Counting direction for general-purpose (GP) timer/GPIO. If TDIR=1, upward counting is selected. If TDIR=0, downward counting is selected.
TCLKIN/***IOPB7***	87	I/O	I	External clock input for GP timer/GPIO. Note that timer can also use the internal device clock.
CAP1/QEP0/***IOPA3***	123	I/O	I	Capture input #1/quadrature encoder pulse input #0/GPIO
CAP2/QEP1/***IOPA4***	121	I/O	I	Capture input #2/quadrature encoder pulse input #1/GPIO
CAP3/***IOPA5***	119	I/O	I	Capture input #3/GPIO
PWM1/***IOPA6***	102	I/O/Z	I	Compare/PWM output pin #1 or GPIO
PWM2/***IOPA7***	100	I/O/Z	I	Compare/PWM output pin #2 or GPIO
PWM3/***IOPB0***	98	I/O/Z	I	Compare/PWM output pin #3 or GPIO
PWM4/***IOPB1***	96	I/O/Z	I	Compare/PWM output pin #4 or GPIO
PWM5/***IOPB2***	94	I/O/Z	I	Compare/PWM output pin #5 or GPIO
PWM6/***IOPB3***	91	I/O/Z	I	Compare/PWM output pin #6 or GPIO
$\overline{\text{PDPINT}}$	89	I	I	Power drive protection interrupt input. This interrupt, when activated, puts the PWM output pins in the high-impedance state should motor drive/power converter abnormalities, such as overvoltage or overcurrent, etc., arise. $\overline{\text{PDPINT}}$ is level-sensitive and can cause multiple interrupts when held low.

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

‡ The reset state indicates the state of the pin at reset. If the pin is an input, indicated by an I, its state is determined by user design. If the pin is an output, its level at reset is indicated.

§ At reset, the device comes up with AVIS mode enabled. The data bus is in output mode while AVIS is enabled.

NOTE: ***Bold, italicized pin names*** indicate pin function after reset.

LEGEND: ↑ – Internal pullup ↓ – Internal pulldown (Typical active pullup/pulldown value is 150 μA.)

Terminal Functions - F243 PGE Package (Continued)

NAME	144 LQFP NO.	TYPE†	RESET STATE‡	DESCRIPTION
SERIAL PERIPHERAL INTERFACE (SPI) AND BIT I/O PINS				
SPISIMO/***IOPC2***	60	I/O	I	SPI slave in, master out or GPIO
SPISOMI/***IOPC3***	62	I/O	I	SPI slave out, master in or GPIO
SPICLK/***IOPC4***	64	I/O	I	SPI clock or GPIO
$\overline{\text{SPISTE}}$/***IOPC5***	66	I/O	I	SPI slave transmit enable (optional) or GPIO
SERIAL COMMUNICATIONS INTERFACE (SCI) AND BIT I/O PINS				
SCITXD/***IOPA0***	56	I/O	I	SCI asynchronous serial port transmit data or GPIO
SCIRXD/***IOPA1***	58	I/O	I	SCI asynchronous serial port receive data or GPIO
CONTROLLER AREA NETWORK (CAN)				
CANTX/***IOPC6***	115	I/O	I	CAN transmit data or GPIO
CANRX/***IOPC7***	113	I/O	I	CAN receive data or GPIO
INTERRUPT, EXTERNAL ACCESS, AND MISCELLANEOUS SIGNALS				
$\overline{\text{RS}}$	19	I/O	I	Device reset. $\overline{\text{RS}}$ causes the F243/241 to terminate execution and sets PC = 0. When $\overline{\text{RS}}$ is brought to a high level, execution begins at location zero of program memory. $\overline{\text{RS}}$ affects (or sets to zero) various registers and status bits. When the watchdog timer overflows, it initiates a system reset pulse that is reflected on the $\overline{\text{RS}}$ pin. This pulse is eight clock cycles wide.
$\overline{\text{NMI}}$§	79	I	I	Nonmaskable interrupt. When NMI is activated, the device is interrupted regardless of the state of the INTM bit of the status register. $\overline{\text{NMI}}$ is (falling) edge- and low-level-sensitive. To be recognized by the core, this pin must be kept low for at least one clock cycle after the falling edge.
XINT1/***IOPA2***	83	I/O	I	External user interrupt 1 or GPIO. Both XINT1 and XINT2 are edge-sensitive. To be recognized by the core, these pins must be kept high/low for at least one clock cycle after the edge. The edge polarity is programmable.
XINT2/ADCSOC/***IOPD1***	81	I/O	I	External user interrupt 2. External “start-of-conversion” input for ADC/GPIO. Both XINT1 and XINT2 are edge-sensitive. To be recognized by the core, these pins must be kept high/low for at least one clock cycle after the edge. The edge polarity is programmable.
MP/$\overline{\text{MC}}$	43	I	I	Microprocessor/Microcomputer mode select. If this pin is low during reset, the device is put in microcomputer mode and program execution begins at 0000h of internal program memory (flash EEPROM). A high value during reset puts the device in microprocessor mode and program execution begins at 0000h of external program memory. (↓)
READY	44	I	I	READY is pulled low to add wait states for external accesses. READY indicates that an external device is prepared for a bus transaction to be completed. If the device is not ready, it pulls the READY pin low. The processor waits one cycle and checks READY again. Note that the processor performs READY-detection if at least one software wait state is programmed. To meet the external READY timings, the wait-state generator control register (WSGR) should be programmed for at least one wait state. (↑)

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

‡ The reset state indicates the state of the pin at reset. If the pin is an input, indicated by an I, its state is determined by user design. If the pin is an output, its level at reset is indicated.

§ At reset, the device comes up with AVIS mode enabled. The data bus is in output mode while AVIS is enabled.

NOTE: ***Bold, italicized pin names*** indicate pin function after reset.

LEGEND: ↑ – Internal pullup ↓ – Internal pulldown (Typical active pullup/pulldown value is 150 μA.)

Terminal Functions - F243 PGE Package (Continued)

NAME	144 LQFP NO.	TYPE†	RESET STATE‡	DESCRIPTION
INTERRUPT, EXTERNAL ACCESS, AND MISCELLANEOUS SIGNALS (CONTINUED)				
$\overline{IS}$ $\overline{DS}$ $\overline{PS}$	105 110 107	O/Z	1	I/O, data, and program space strobe select signals. $\overline{IS}$, $\overline{DS}$, and $\overline{PS}$ are always high unless low-level asserted for access to the relevant external memory space or I/O. They are placed in the high-impedance state during reset, power down, and when EMU1/$\overline{OFF}$ is active low.
$\overline{WE}$	112	O/Z	1	Write enable strobe. The falling edge of $\overline{WE}$ indicates that the device is driving the external data bus (D15–D0). $\overline{WE}$ is active on all external program, data, and I/O writes. $\overline{WE}$ goes in the high-impedance state when EMU1/$\overline{OFF}$ is active low.
$\overline{RD}$	118	O	1	Read enable strobe. Read-select indicates an active, external read cycle. $\overline{RD}$ is active on all external program, data, and I/O reads. $\overline{RD}$ goes into the high-impedance state when EMU1/$\overline{OFF}$ is active low.
R/$\overline{W}$	114	O/Z	1	Read/write signal. R/$\overline{W}$ indicates transfer direction during communication to an external device. It is normally in read mode (high), unless low level is asserted for performing a write operation. It is placed in the high-impedance state when EMU1/$\overline{OFF}$ is active low and during power down.
$\overline{STRB}$	122	O/Z	1	External memory access strobe. $\overline{STRB}$ is always high unless asserted low to indicate an external bus cycle. $\overline{STRB}$ is active for all off-chip accesses. It is placed in the high-impedance state during power down, and when EMU1/$\overline{OFF}$ is active low.
$\overline{BR}$	120	O/Z	1	Bus request, global memory strobe. $\overline{BR}$ is asserted during access of external global data memory space. $\overline{BR}$ can be used to extend the data memory address space by up to 32K words. $\overline{BR}$ goes in the high-impedance state during reset, power down, and when EMU1/$\overline{OFF}$ is active low.
VIS_CLK	31	O	0	Visibility clock. Same as CLKOUT, but timing is aligned for external buses in visibility mode.
ENA_144	18	I	I	Active high to enable external interface signals. If pulled low, the F243 behaves like an F241—i.e., it has no external memory and generates an illegal address if any of the three external spaces are accessed ($\overline{IS}$, $\overline{DS}$, $\overline{PS}$ asserted). This pin has an internal pulldown. (↓)
$\overline{VIS\_OE}$	126	O	0	This pin is active (low) whenever the external databus is driving as an output during visibility mode. Can be used by external decode logic to prevent data bus contention while running in visibility mode.
XF/IOPC0	49	I/O	O – 1	External flag output (latched software-programmable signal). XF is a general-purpose output pin. It is set/reset by the `SETC XF/CLRC XF` instruction. This pin is configured as an external flag output by all device resets. It can be used as a GPIO, if not used as XF.
$\overline{BIO}$/IOPC1	55	I/O	I	Branch control input. $\overline{BIO}$ is polled by the `BCND pma,BIO` instruction. If $\overline{BIO}$ is low, a branch is executed. If $\overline{BIO}$ is not used, it should be pulled high. This pin is configured as a branch control input by all device resets. It can be used as a GPIO, if not used as a branch control input.

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

‡ The reset state indicates the state of the pin at reset. If the pin is an input, indicated by an I, its state is determined by user design. If the pin is an output, its level at reset is indicated.

§ At reset, the device comes up with AVIS mode enabled. The data bus is in output mode while AVIS is enabled.

NOTE: ***Bold, italicized pin names*** indicate pin function after reset.

LEGEND: ↑ – Internal pullup ↓ – Internal pulldown (Typical active pullup/pulldown value is 150 μA.)

Terminal Functions - F243 PGE Package (Continued)

NAME	144 LQFP NO.	TYPE†	RESET STATE‡	DESCRIPTION
INTERRUPT, EXTERNAL ACCESS, AND MISCELLANEOUS SIGNALS (CONTINUED)				
PMT	68	I	I	Enables parallel module test (PMT). This pin must be connected to ground. (↓)
V_{CCP}/WDDIS	77	I	I	Flash programming voltage pin and watchdog disable. This is the 5-V supply used for flash programming. Flash cannot be programmed if this pin is held at 0 V. This pin also works as a hardware watchdog disable, when V_{CCP}/WDDIS = +5 V and bit 6 in WDCR is set to 1. Do not use a current-limiting resistor on this pin.
DEDICATED I/O SIGNALS				
IOPD2	20	I/O	I	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 (↓)
DATA AND ADDRESS BUS SIGNALS				
D0 (↓)	33	I/O/Z	O§	Bit x of the 16-bit Data Bus
D1 (↓)	35			
D2 (↓)	38			
D3 (↓)	46			
D4 (↓)	48			
D5 (↓)	50			
D6 (↓)	52			
D7 (↓)	54			
D8 (↑)	57			
D9 (↓)	59			
D10 (↓)	61			
D11 (↓)	63			
D12 (↓)	65			
D13 (↓)	67			
D14 (↓)	69			
D15 (↓)	71			

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

‡ The reset state indicates the state of the pin at reset. If the pin is an input, indicated by an I, its state is determined by user design. If the pin is an output, its level at reset is indicated.

§ At reset, the device comes up with AVIS mode enabled. The data bus is in output mode while AVIS is enabled.

NOTE: ***Bold, italicized pin names*** indicate pin function after reset.

LEGEND: ↑ – Internal pullup ↓ – Internal pulldown (Typical active pullup/pulldown value is 150 μA.)

Terminal Functions - F243 PGE Package (Continued)

NAME	144 LQFP NO.	TYPE†	RESET STATE‡	DESCRIPTION
DATA AND ADDRESS BUS SIGNALS (CONTINUED)				
A0	104	O	0	Bit x of the 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				
XTAL1/CLKIN	41	I	I	PLL oscillator input pin. Crystal input to PLL/clock source input to PLL. XTAL1/CLKIN is tied to one side of a reference crystal.
XTAL2	42	O	O	Crystal output. PLL oscillator output pin. XTAL2 is tied to one side of a reference crystal. This pin goes in the high-impedance state when EMU1/$\overline{\text{OFF}}$ is active low.
CLKOUT/IOPD0	116	I/O	O	Clock output. This pin outputs either the CPU clock (CLKOUT) or the watchdog clock (WDCLK). The selection is made by the CLKSRC bit (bit 14) of the System Control and Status Register (SCSR). This pin can be used as a GPIO if not used as a clock output pin.
TEST SIGNALS				
TCK	22	I	I	JTAG test clock with internal pullup (↑)
TDI	24	I	I	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	I/O	I	JTAG scan out, 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	I	JTAG test-mode select (TMS) with internal pullup. This serial control input is clocked into the TAP controller on the rising edge of TCK. (↑)

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

‡ The reset state indicates the state of the pin at reset. If the pin is an input, indicated by an I, its state is determined by user design. If the pin is an output, its level at reset is indicated.

§ At reset, the device comes up with AVIS mode enabled. The data bus is in output mode while AVIS is enabled.

NOTE: ***Bold, italicized pin names*** indicate pin function after reset.

LEGEND: ↑ – Internal pullup ↓ – Internal pulldown (Typical active pullup/pulldown value is 150 μA.)

Terminal Functions - F243 PGE Package (Continued)

NAME	144 LQFP NO.	TYPE†	RESET STATE‡	DESCRIPTION
TEST SIGNALS (CONTINUED)				
$\overline{\text{TRST}}$	30	I	I	JTAG test reset with internal pulldown. $\overline{\text{TRST}}$, when driven high, 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	I	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/$\overline{\text{OFF}}$	47	I/O	I	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. (↑)
SUPPLY SIGNALS				
V_{SSO}	14 15 36 37 40 70 73 108 111 117 124 129 131	–	–	Digital logic and buffer ground reference
V_{DDO}	34 39 72 75 106 109	–	–	Digital logic and buffer supply voltage
V_{DD}	17 53 125	–	–	Digital logic supply voltage
V_{SS}	16 32 51 127	–	–	Digital logic ground reference

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

‡ The reset state indicates the state of the pin at reset. If the pin is an input, indicated by an I, its state is determined by user design. If the pin is an output, its level at reset is indicated.

§ At reset, the device comes up with AVIS mode enabled. The data bus is in output mode while AVIS is enabled.

NOTE: ***Bold, italicized pin names*** indicate pin function after reset.

LEGEND: ↑ – Internal pullup ↓ – Internal pulldown (Typical active pullup/pulldown value is 150 μA.)

Terminal Functions - F243 PGE Package (Continued)

NAME	144 LQFP NO.	TYPE†	RESET STATE‡	DESCRIPTION
NO CONNECTS				
DNC	12 97	–	–	Do not connect. Reserved for test.
NC	1 2 5 7 9 11 13 132 133 134 136 138 140	–	–	No internal connection made to this pin

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

‡ The reset state indicates the state of the pin at reset. If the pin is an input, indicated by an I, its state is determined by user design. If the pin is an output, its level at reset is indicated.

§ At reset, the device comes up with AVIS mode enabled. The data bus is in output mode while AVIS is enabled.

NOTE: ***Bold, italicized pin names*** indicate pin function after reset.

LEGEND: ↑ – Internal pullup ↓ – Internal pulldown (Typical active pullup/pulldown value is 150 μA.)

Terminal Functions - F241 PG and FN Packages

NAME	64 QFP NO.	68 PLCC NO.	TYPE†	RESET STATE‡	DESCRIPTION
ANALOG-TO-DIGITAL CONVERTER (ADC) INPUTS					
ADCIN00	24	32	I	I	Analog inputs to the ADC
ADCIN01	23	31			
ADCIN02	22	30			
ADCIN03	21	29			
ADCIN04	20	28			
ADCIN05	19	26			
ADCIN06	18	25			
ADCIN07	15	22			
V_{CCA}	14	21	–	–	Analog supply voltage for ADC (5 V). It is highly recommended to isolate V_{CCA} from the digital supply voltage (and V_{SSA} from digital ground) to maintain the specified accuracy and improve the noise immunity of the ADC.
V_{SSA}	13	20	–	–	Analog ground reference for ADC
V_{REFHI}	16	23	–	–	ADC analog high-voltage reference input
V_{REFLO}	17	24	–	–	ADC analog low-voltage reference input
EVENT MANAGER					
T1CMP/T1PWM/***IOPB4***	12	19	I/O/Z	I	Timer 1 compare output/general-purpose bidirectional digital I/O (GPIO).
T2CMP/T2PWM/***IOPB5***	11	18	I/O/Z		Timer 2 compare output/GPIO
TDIR/***IOPB6***	56	67	I/O		Counting direction for GP timer/GPIO. If TDIR=1, upward counting is selected. If TDIR=0, downward counting is selected.
TCLKIN/***IOPB7***	57	68	I/O		External clock input for GP timer/GPIO. Note that timer can also use the internal device clock.
CAP1/QEP0/***IOPA3***	8	15	I/O		Capture input #1/quadrature encoder pulse input #0/GPIO
CAP2/QEP1/***IOPA4***	7	14	I/O		Capture input #2/quadrature encoder pulse input #1/GPIO
CAP3/***IOPA5***	6	13	I/O		Capture input #3/GPIO
PWM1/***IOPA6***	64	7	I/O/Z		Compare/PWM output pin #1 or GPIO
PWM2/***IOPA7***	63	6	I/O/Z		Compare/PWM output pin #2 or GPIO
PWM3/***IOPB0***	62	5	I/O/Z		Compare/PWM output pin #3 or GPIO
PWM4/***IOPB1***	61	4	I/O/Z		Compare/PWM output pin #4 or GPIO
PWM5/***IOPB2***	60	3	I/O/Z		Compare/PWM output pin #5 or GPIO
PWM6/**IOPB3**	59	2	I/O/Z		Compare/PWM output pin #6 or GPIO
$\overline{\text{PDPINT}}$	58	1	I	I	Power drive protection interrupt input. This interrupt, when activated, puts the PWM output pins in the high-impedance state, should motor drive/power converter abnormalities, such as overvoltage or overcurrent, etc., arise. $\overline{\text{PDPINT}}$ is level-sensitive and can cause multiple interrupts when held low.

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

‡ The reset state indicates the state of the pin at reset. If the pin is an input, indicated by an I, its state is determined by user design. If the pin is an output, its level at reset is indicated.

NOTE: ***Bold, italicized pin names*** indicate pin function after reset.

LEGEND: ↑ – Internal pullup ↓ – Internal pulldown (Typical active pullup/pulldown value is 150 μA.)

Terminal Functions - F241 PG and FN Packages (Continued)

NAME	64 QFP NO.	68 PLCC NO.	TYPE†	RESET STATE‡	DESCRIPTION
SERIAL PERIPHERAL INTERFACE (SPI) AND BIT I/O PINS					
SPISIMO/***IOPC2***	45	56	I/O	I	SPI slave in, master out or GPIO
SPISOMI/***IOPC3***	46	57	I/O		SPI slave out, master in or GPIO
SPICLK/***IOPC4***	47	58	I/O		SPI clock or GPIO
$\overline{\text{SPISTE}}$/***IOPC5***	48	59	I/O		SPI slave transmit enable (optional) or GPIO
SERIAL COMMUNICATIONS INTERFACE (SCI) AND BIT I/O PINS					
SCITXD/***IOPA0***	43	54	I/O	I	SCI asynchronous serial port transmit data or GPIO
SCIRXD/***IOPA1***	44	55	I/O		SCI asynchronous serial port receive data or GPIO
CONTROLLER AREA NETWORK (CAN)					
CANTX/***IOPC6***	4	11	I/O	I	CAN transmit data or GPIO
CANRX/***IOPC7***	3	10	I/O		CAN receive data or GPIO
INTERRUPT, EXTERNAL ACCESS, AND MISCELLANEOUS SIGNALS					
$\overline{\text{RS}}$	27	35	I/O	I	Device reset. $\overline{\text{RS}}$ causes the F243/241 to terminate execution and sets PC = 0. When $\overline{\text{RS}}$ is brought to a high level, execution begins at location zero of program memory. $\overline{\text{RS}}$ affects (or sets to zero) various registers and status bits. When the watchdog timer overflows, it initiates a system reset pulse reflected on the $\overline{\text{RS}}$ pin. This pulse is eight clock cycles wide.
$\overline{\text{NMI}}$	53	64	I	I	Nonmaskable interrupt. When NMI is activated, the device is interrupted regardless of the state of the INTM bit of the status register. $\overline{\text{NMI}}$ is (falling) edge- and low-level-sensitive. To be recognized by the core, this pin must be kept low for at least one clock cycle after the falling edge.
XINT1/***IOPA2***	55	66	I/O	I	External user interrupt 1 or GPIO. Both XINT1 and XINT2 are edge-sensitive. To be recognized by the core, these pins must be kept low/high for at least one clock cycle after the edge. The edge polarity is programmable.
XINT2/ADCSOC/***IOPD1***	54	65	I/O	I	External user interrupt 2. External “start-of-conversion” input for ADC/GPIO. Both XINT1 and XINT2 are edge-sensitive. To be recognized by the core, these pins must be kept low/high for at least one clock cycle after the edge. The edge polarity is programmable.
XF/IOPC0	39	50	I/O	O – 1	External flag output (latched software-programmable signal). XF is a general-purpose output pin. It is set/reset by the `SETC XF/CLRC XF` instruction. This pin is configured as an external flag output by all device resets. It can be used as a GPIO, if not used as XF.
$\overline{\textbf{\textit{BIO}}}$/IOPC1	42	53	I/O	I	Branch control input. $\overline{\text{BIO}}$ is polled by the `BCND pma,BIO` instruction. If $\overline{\text{BIO}}$ is low, a branch is executed. If $\overline{\text{BIO}}$ is not used, it should be pulled high. This pin is configured as a branch control input by all device resets. It can be used as a GPIO, if not used as a branch control input.
PMT	49	60	I	I	Enables parallel module test (PMT). This pin must be connected to ground.

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

‡ The reset state indicates the state of the pin at reset. If the pin is an input, indicated by an I, its state is determined by user design. If the pin is an output, its level at reset is indicated.

NOTE: ***Bold, italicized pin names*** indicate pin function after reset.

LEGEND: ↑ – Internal pullup ↓ – Internal pulldown (Typical active pullup/pulldown value is 150 μA.)

Terminal Functions - F241 PG and FN Packages (Continued)

NAME	64 QFP NO.	68 PLCC NO.	TYPE†	RESET STATE‡	DESCRIPTION
CLOCK SIGNALS					
XTAL1/CLKIN	35	46	I	I	PLL oscillator input pin. Crystal input to PLL/clock source input to PLL. XTAL1/CLKIN is tied to one side of a reference crystal.
XTAL2	36	47	O	O	Crystal output. PLL oscillator output pin. XTAL2 is tied to one side of a reference crystal. This pin goes in the high-impedance state when EMU1/$\overline{\text{OFF}}$ is active low.
CLKOUT/IOPD0	5	12	I/O	O	Clock output. This pin outputs either the CPU clock (CLKOUT) or the watchdog clock (WDCLK). The selection is made by the CLKSRC bit (bit 14) of the System Status and Control Register (SSCR). This pin can be used as a GPIO if not used as a clock output pin.
TEST SIGNALS					
TCK	28	36	I	I	JTAG test clock with internal pullup (↑)
TDI	29	37	I	I	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	30	38	I/O	I	JTAG scan out, 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	31	39	I	I	JTAG test-mode select (TMS) with internal pullup. This serial control input is clocked into the TAP controller on the rising edge of TCK. (↑)
$\overline{\text{TRST}}$	32	40	I	I	JTAG test reset with internal pulldown. $\overline{\text{TRST}}$, when driven high, 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	37	48	I/O	I	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	38	49	I/O	I	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. (↑)
SUPPLY SIGNALS					
V_{DD}	9	16	–	–	Digital logic supply voltage (5 V)
	41	52	–	–	
V_{DDO}	–	42	–	–	Digital logic and buffer supply voltage (5 V)
	1	8	–	–	
	34	45	–	–	
	51	62	–	–	
V_{SS}	–	41	–	–	Digital logic ground reference
	10	17	–	–	
	40	51	–	–	

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

‡ The reset state indicates the state of the pin at reset. If the pin is an input, indicated by an I, its state is determined by user design. If the pin is an output, its level at reset is indicated.

NOTE: ***Bold, italicized pin names*** indicate pin function after reset.

LEGEND: ↑ – Internal pullup ↓ – Internal pulldown (Typical active pullup/pulldown value is 150 μA.)

Terminal Functions - F241 PG and FN Packages (Continued)

NAME	64 QFP	68 PLCC	TYPE†	RESET STATE‡	DESCRIPTION
SUPPLY SIGNALS (CONTINUED)					
V_{SSO}	–	43	–	–	Digital logic and buffer ground reference
	2	9	–	–	
	26	34	–	–	
	33	44	–	–	
	50	61	–	–	
NO CONNECT					
NC	–	27			No internal connection made to this pin
DNC	25	33	–	–	Do not connect. Reserved for test.
INTERFACE CONTROL SIGNALS					
V_{CCP}/WDDIS	52	63	I	I	Flash programming voltage supply pin. This is the 5-V supply used for flash programming. Flash cannot be programmed if this pin is held at 0 V. This pin also works as a hardware watchdog disable, when V_{CCP}/WDDIS = +5 V and bit 6 in WDCR is set to 1. Note that on ROM devices, only the WDDIS function is valid. Do not use a current-limiting resistor on this pin.

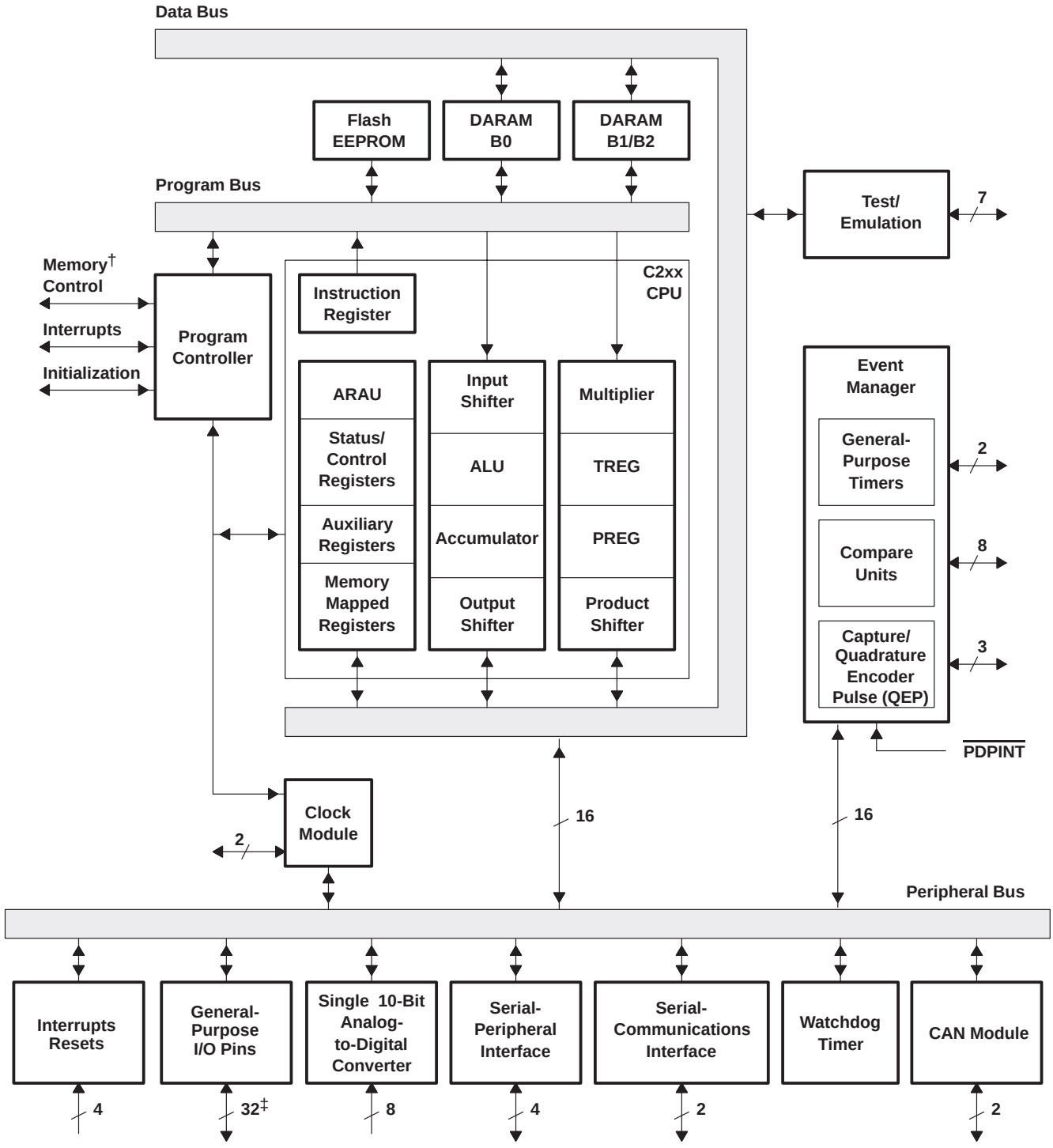
† I = input, O = output, Z = high impedance

‡ The reset state indicates the state of the pin at reset. If the pin is an input, indicated by an I, its state is determined by user design. If the pin is an output, its level at reset is indicated.

NOTE: ***Bold, italicized pin names*** indicate pin function after reset.

LEGEND: ↑ – Internal pullup ↓ – Internal pulldown (Typical active pullup/pulldown value is 150 μA.)

functional block diagram of the 24x DSP controller



† F243 only
‡ 26 in F241

architectural overview

The functional block diagram provides a high-level description of each component in the F243/F241 DSP controllers. 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 F243/F241 that are distributed. These system features include the memory map, device reset, interrupts, digital input/output (I/O), clock generation, and low-power operation.

system-level functions

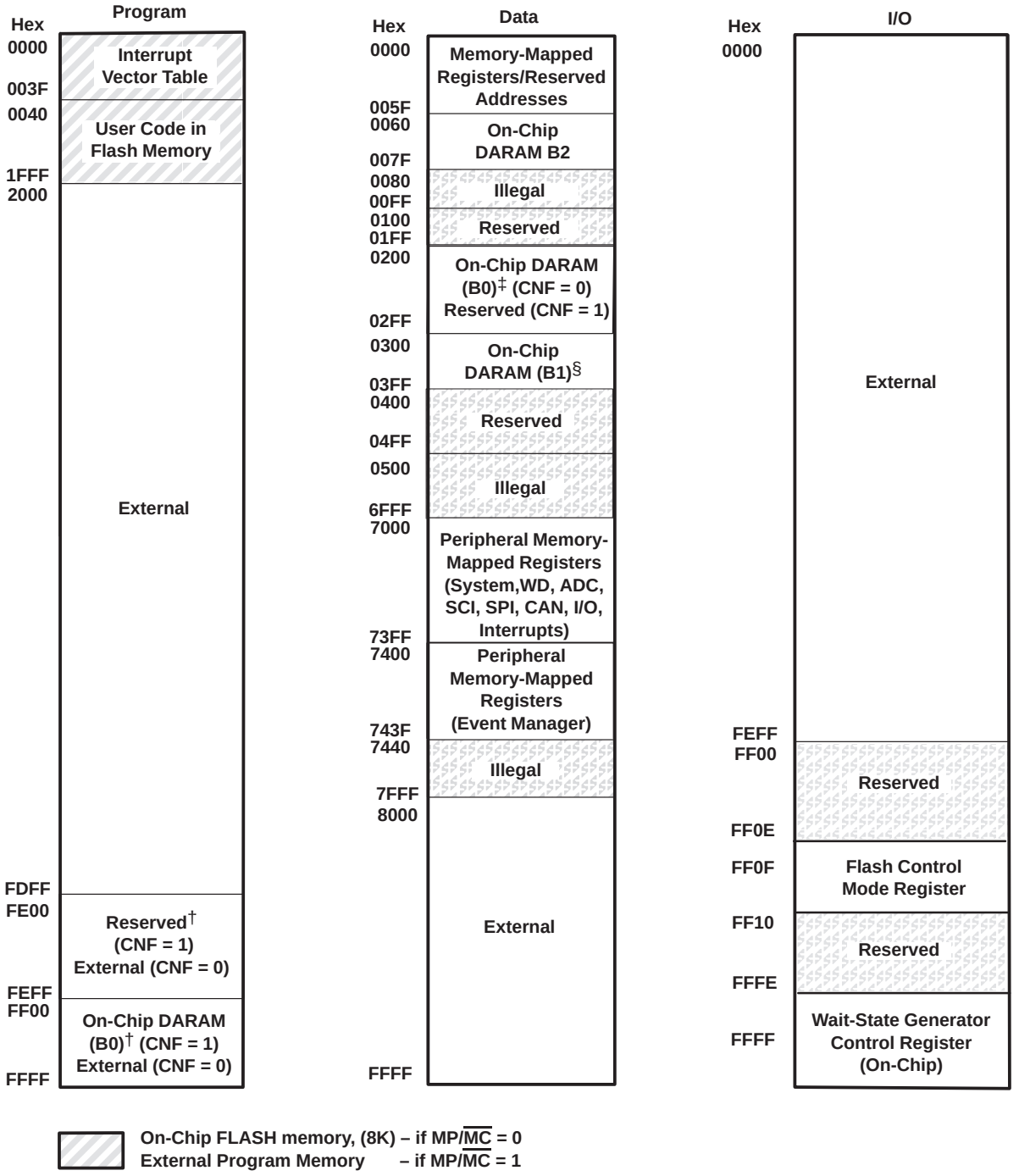
device memory maps

The F243/F241 devices implement three separate address spaces for program memory, data memory, and I/O space. On the F243/F241, 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 F243/F241 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 F243 memory map and Figure 2 shows the F241 memory map.

CAUTION:
Accessing “Reserved” memory locations may cause unpredictable device operation.

memory maps



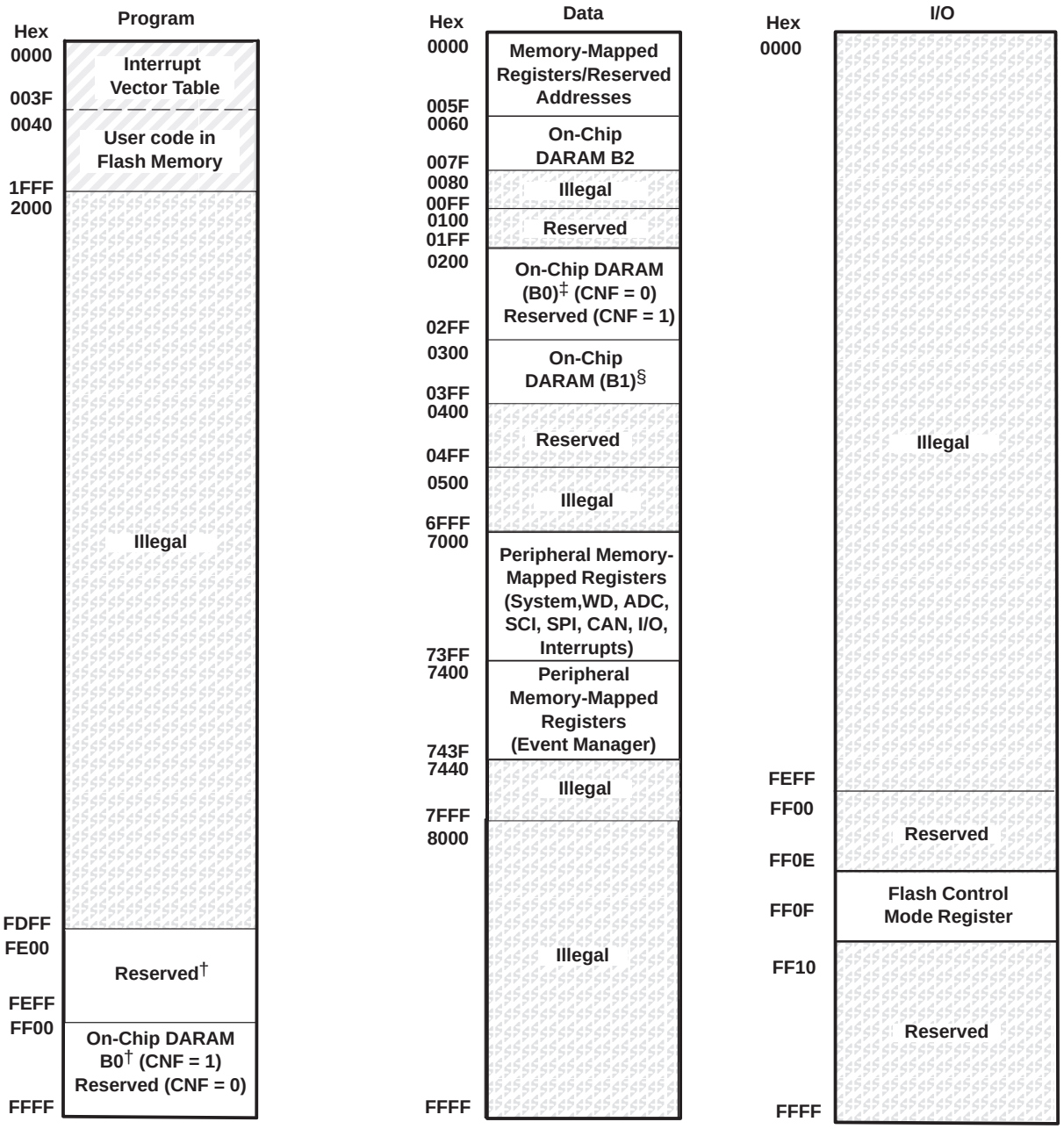
† When CNF = 1, addresses FE00h–FEFFh and FF00h–FFFFh are mapped to the same physical block (B0) in program-memory space. For example, a write to FE00h will have the same effect as a write to FF00h. For simplicity, addresses FE00h–FEFFh are referred to as reserved when CNF = 1.

‡ When CNF = 0, addresses 0100h–01FFh and 0200h–02FFh are mapped to the same physical block (B0) in data-memory space. For example, a write to 0100h will have the same effect as a write to 0200h. For simplicity, addresses 0100h–01FFh are referred to as reserved.

§ Addresses 0300h–03FFh and 0400h–04FFh are mapped to the same physical block (B1) in data-memory space. For example, a write to 0400h has the same effect as a write to 0300h. For simplicity, addresses 0400h–04FFh are referred to as Reserved.

Figure 1. TMS320F243 Memory Map

memory maps (continued)



† When CNF = 1, addresses FE00h–FEFFh and FF00h–FFFFh are mapped to the same physical block (B0) in program-memory space. For example, a write to FE00h will have the same effect as a write to FF00h. For simplicity, addresses FE00h–FEFFh are referred to as reserved when CNF = 1.

‡ When CNF = 0, addresses 0100h–01FFh and 0200h–02FFh are mapped to the same physical block (B0) in data-memory space. For example, a write to 0100h will have the same effect as a write to 0200h. For simplicity, addresses 0100h–01FFh are referred to as reserved.

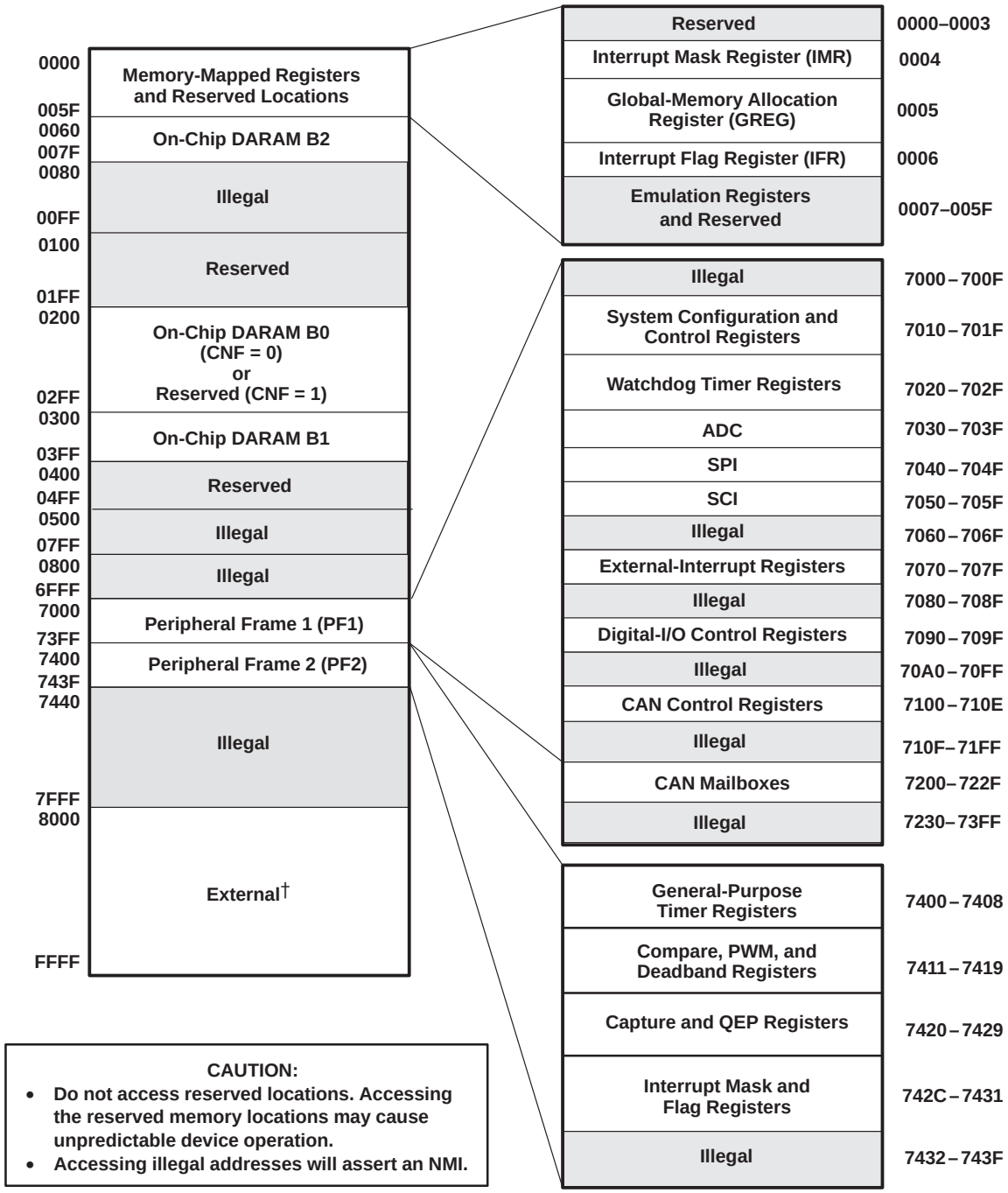
§ Addresses 0300h–03FFh and 0400h–04FFh are mapped to the same physical block (B1) in data-memory space. For example, a write to 0400h has the same effect as a write to 0300h. For simplicity, addresses 0400h–04FFh are referred to as Reserved.

NOTE A: There is no external memory space for program, data, or I/O in the F241.

Figure 2. TMS320F241 Memory Map

peripheral memory map

The 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). The register frame is mapped in the data memory space.



† External memory is available on the F243 only, and it is illegal in the F241.

Figure 3. Peripheral Memory Map for F243/F241

software-controlled wait-state generator

Due to the fast cycle time of the F243 devices, it is often necessary to operate with wait states to interface with external logic or memory. For many systems, one wait state is adequate.

The software wait-state generator can be programmed to generate between 0 and 7 wait states for a given space. Software wait states are configured through the wait-state generator register (WSGR). The WSGR includes three 3-bit fields to configure wait states for the following external memory spaces: data space (DSWS), program space (PSWS), and I/O space (ISWS). The wait-state generator enables wait states for a given memory space based on the value of the corresponding three bits, regardless of the condition of the READY signal. The READY signal can be used to generate additional wait states. All bits of the WSGR are set to 1 at reset so that the device can operate from slow memory at reset. The WSGR register (shown in Table 3, Table 4 and Table 5) resides at I/O location FFFFh. This register should not be accessed in the F241.

Table 3. Wait-State Generator Control Register (WSGR)

15 – 12	11 – 10	9 – 8	7 – 6	5 – 3	2 – 0
Reserved	BVIS	ISWS		DSWS	PSWS
0	R/W–11	R/W–111		R/W–111	R/W–111

LEGEND:
0 = Always read as zeros, R = Read Access, W= Write Access, – n = Value after reset

Table 4. Wait-State(s) Programming

PSWS, DSWS, ISWS BITS	WAIT STATES FOR PROGRAM, DATA, OR I/O
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Table 5. Wait-State Generator Control Register (WSGR)

BITS	NAME	DESCRIPTION
2–0	PSWS	External program space wait states. PSWS determines that between 0 to 7 wait states are applied to all reads and writes to off-chip program space address. The memory cycle can be further extended by using the READY signal. The READY signal does not override the wait states generated by PSWS. These bits are set to 1 (active) by reset ($\overline{RS}$).
5–3	DSWS	External data space wait states. DSWS determines that between 0 to 7 wait states are applied to all reads and writes to off-chip data space. The memory cycle can be further extended by using the READY signal. The READY signal does not override the wait states generated by DSWS. These bits are set to 1 (active) by reset ($\overline{RS}$).
8–6	ISWS	External input / output space wait state. ISWS determines that between 0 to 7 wait states are applied to all reads and writes to off-chip I/O space. The memory cycle can be further extended by using the READY signal. The READY signal does not override the wait states generated by ISWS. These bits are set to 1 (active) by reset ($\overline{RS}$).
10–9	BVIS	Bus visibility modes. Bits 10 and 9 allow selection of various bus visibility modes while running from internal program and/or data memory. These modes provide a method of tracing internal bus activity. These bits are set to 11b by reset ($\overline{RS}$), causing internal program address and program data to be output on the external address and data pins. See Table 6.
15–11	–	Reserved

software-controlled wait-state generator (continued)

Table 6. Visibility Modes

BIT 10	BIT 9	VISIBILITY MODE
0	0	Bus visibility OFF (reduces power consumption and noise)
0	1	Bus visibility OFF (reduces power consumption and noise)
1	0	Data-address bus output to external address bus. Data-data bus output to external data bus.
1	1	Program-address bus output to external address bus. Program-data bus output to external data bus.

digital I/O and shared pin functions

The F243 has a total of 32 general-purpose, bidirectional, digital I/O (GPIO) pins that function as follows: six pins are dedicated I/O pins (see Table 7) and 26 pins are shared between primary functions and I/O. The F241 has 26 I/O pins; all are shared with other functions. The digital I/O ports module provides a flexible method for controlling both dedicated I/O and shared pin functions. All I/O and shared pin functions are controlled using eight 16-bit registers. These registers are divided into two types:

- Output Control Registers — used to control the multiplexer selection that chooses between the primary function of a pin or the general-purpose I/O function.
- Data and Control Registers — used to control the data and data direction of bidirectional I/O pins.

Table 7. Dedicated I/O Pins (F243 Only)

F243 PIN NUMBER	PIN NAME
20	IOPD2
21	IOPD3
23	IOPD4
25	IOPD5
27	IOPD6
29	IOPD7

description of shared I/O pins

The control structure for shared I/O pins is shown in Figure 4, where 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.

description of shared I/O pins (continued)

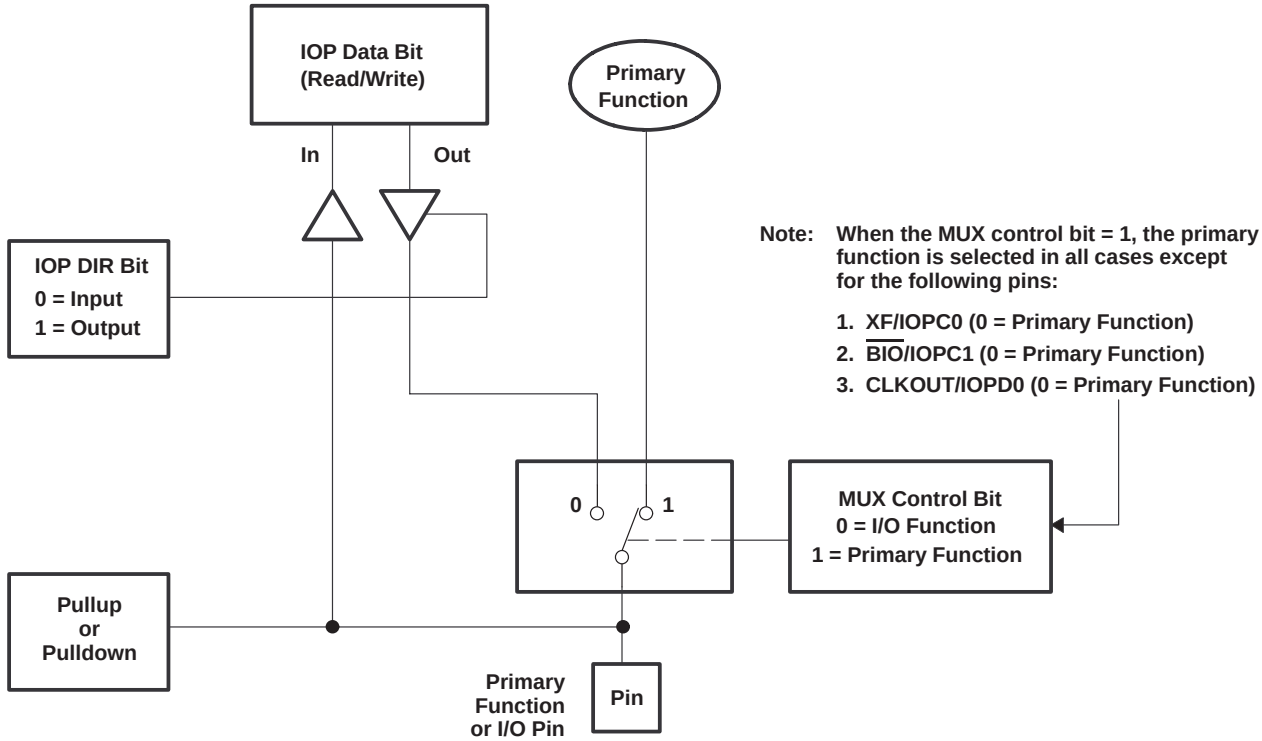


Figure 4. Shared Pin Configuration

A summary of shared pin configurations and associated bits is shown in Table 8.

description of shared I/O pins (continued)

Table 8. Shared Pin Configurations

PIN NO.			MUX CONTROL REGISTER (name.bit #)	PIN FUNCTION SELECTED		I/O PORT DATA AND DIRECTION†		
144 PQFP	68 PLCC	64 QFP		(OCRx.n = 1)	(OCRx.n = 0)	REGISTER	DATA BIT NO.‡	DIR BIT NO.§
F243	F241							
56	54	43	OCRA.0	SCITXD	IOPA0	PADATDIR	0	8
58	55	44	OCRA.1	SCIRXD	IOPA1	PADATDIR	1	9
83	66	55	OCRA.2	XINT1	IOPA2	PADATDIR	2	10
123	15	8	OCRA.3	CAP1/QEP0	IOPA3	PADATDIR	3	11
121	14	7	OCRA.4	CAP2/QEP1	IOPA4	PADATDIR	4	12
119	13	6	OCRA.5	CAP3	IOPA5	PADATDIR	5	13
102	7	64	OCRA.6	PWM1	IOPA6	PADATDIR	6	14
100	6	63	OCRA.7	PWM2	IOPA7	PADATDIR	7	15
98	5	62	OCRA.8	PWM3	IOPB0	PBDATDIR	0	8
96	4	61	OCRA.9	PWM4	IOPB1	PBDATDIR	1	9
94	3	60	OCRA.10	PWM5	IOPB2	PBDATDIR	2	10
91	2	59	OCRA.11	PWM6	IOPB3	PBDATDIR	3	11
130	19	12	OCRA.12	T1PWM/T1CMP	IOPB4	PBDATDIR	4	12
128	18	11	OCRA.13	T2PWM/T2CMP	IOPB5	PBDATDIR	5	13
85	67	56	OCRA.14	TDIR	IOPB6	PBDATDIR	6	14
87	68	57	OCRA.15	TCLKIN	IOPB7	PBDATDIR	7	15
49	50	39	OCRB.0	IOPC0	XF	PCDATDIR	0	8
55	53	42	OCRB.1	IOPC1	$\overline{\text{BIO}}$	PCDATDIR	1	9
60	56	45	OCRB.2	SPISIMO	IOPC2	PCDATDIR	2	10
62	57	46	OCRB.3	SPISOMI	IOPC3	PCDATDIR	3	11
64	58	47	OCRB.4	SPICLK	IOPC4	PCDATDIR	4	12
66	59	48	OCRB.5	$\overline{\text{SPISTE}}$	IOPC5	PCDATDIR	5	13
115	11	4	OCRB.6	CANTX	IOPC6	PCDATDIR	6	14
113	10	3	OCRB.7	CANRX	IOPC7	PCDATDIR	7	15
116	12	5	OCRB.8	IOPD0	CLKOUT	PDDATDIR	0	8
81	65	54	OCRB.9	XINT2/ADCSOC	IOPD1	PDDATDIR	1	9

† Valid only if the I/O function is selected on the pin.
‡ If the GPIO pin is configured as an output, these bits can be written to. If the pin is configured as an input, these bits are read from.
§ If the DIR bit is 0, the GPIO pin functions as an input. For a value of 1, the pin is configured as an output.
NOTE: GPIO pins IOPD2 to IOPD7 are dedicated I/O pins in F243. These pins are not available in the F241.

digital I/O control registers

Table 9 lists the registers available in the digital I/O module. As with other F243/F241 peripherals, the registers are memory-mapped to the data space.

Table 9. 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 F243/F241 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 F243/F241 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, $\overline{\text{PDPINT}}$, and $\overline{\text{NMI}}$. 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. $\overline{\text{NMI}}$, which is not maskable, takes priority over peripheral interrupts and software-generated interrupts. It can be locked out only by an already executing $\overline{\text{NMI}}$ 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 F243/F241 devices include:

  – *The INTR instruction.* This instruction allows initialization of any F243/F241 interrupt with software. Its operand indicates the interrupt vector location to which 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 $\overline{\text{NMI}}$ 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 5.

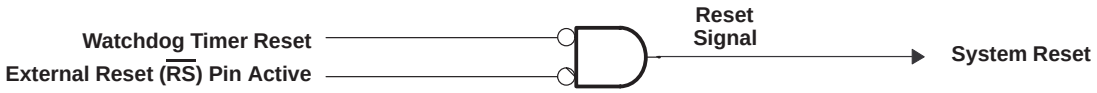


Figure 5. Reset Signals

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.) The watchdog timer reset is reflected on the external $\overline{RS}$ pin also.
- **Reset pin active.** To generate an external reset pulse on the $\overline{RS}$ pin, a low-level pulse duration of at least one CPUCLK cycle is necessary to ensure that the device recognizes the reset signal.

Once watchdog reset 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 24x CPU supports one nonmaskable interrupt (NMI) and six maskable prioritized interrupt requests. The 24x devices have many peripherals, and each peripheral is capable of generating one or more interrupts in response to many events. The 24x CPU does not have sufficient interrupt requests to handle all these peripheral interrupt requests; therefore, a centralized interrupt controller is provided to arbitrate the interrupt requests from all the different sources. Throughout this section, refer to Figure 6 .

hardware-generated interrupts (continued)

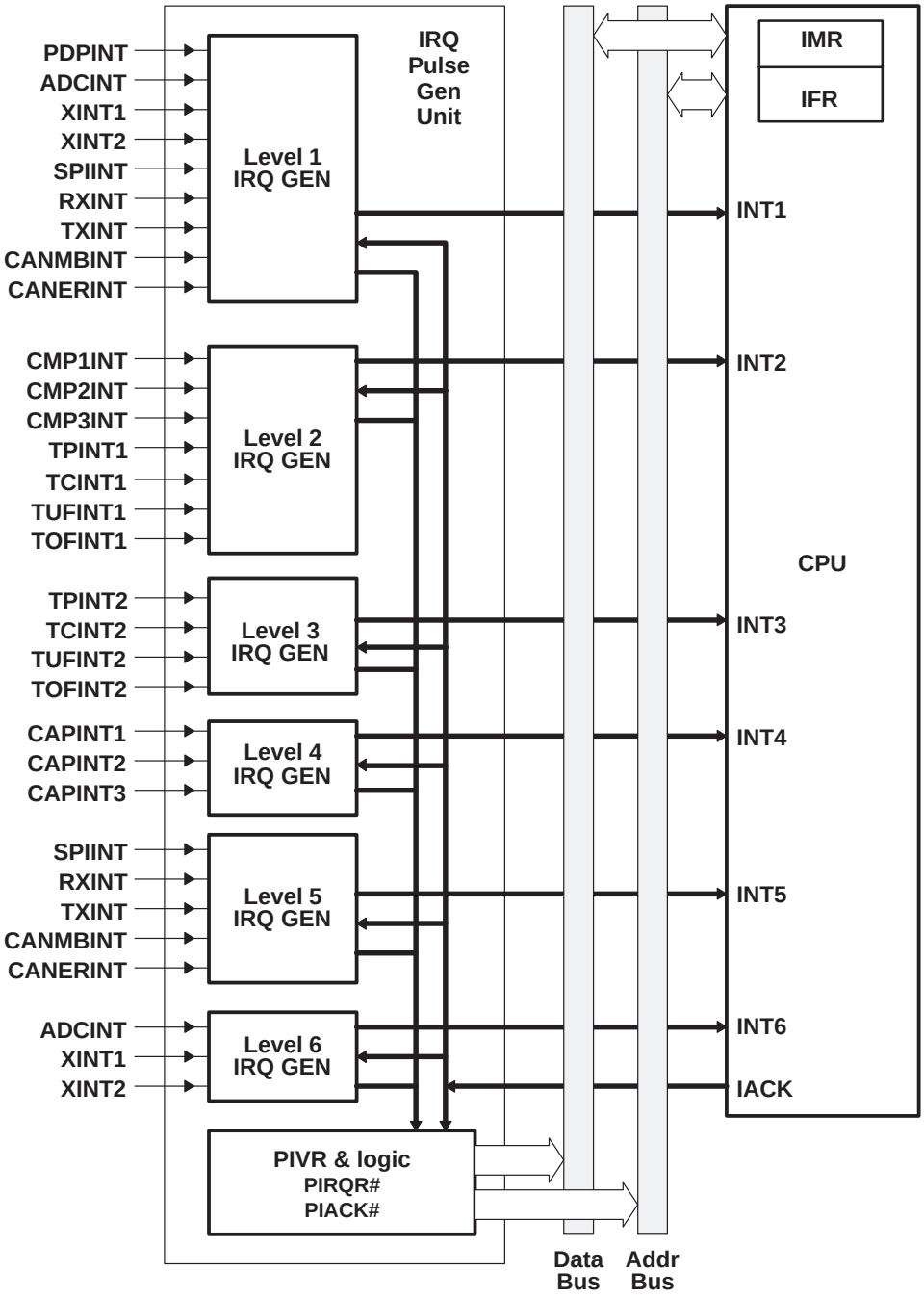


Figure 6. Peripheral Interrupt Expansion Block Diagram

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 (PIRQs) 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. 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 24x CPU is a low-going pulse of 2 CPU clock cycles. The Peripheral Interrupt Expansion (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 10.

interrupt request structure (continued)

Table 10. F243/F241 Interrupt Source Priority and Vectors

INTERRUPT NAME	OVERALL PRIORITY	CPU INTERRUPT AND VECTOR ADDRESS	BIT POSITION IN PIRQRx AND PIACKRx	PERIPHERAL INTERRUPT VECTOR (PIV)	MASKABLE?	SOURCE PERIPHERAL MODULE	DESCRIPTION
Reset	1	RSN 0000h		N/A	N	$\overline{\text{RS}}$ 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			
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			
CANERINT	12		0.8	0041h			
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		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 0008h	1.4	0033h	Y	EV	Capture 1 interrupt
CAPINT2	25		1.5	0034h	Y	EV	Capture 2 interrupt
CAPINT3	26		1.6	0035h	Y	EV	Capture 3 interrupt

interrupt request structure (continued)

Table 10. F243/F241 Interrupt Source Priority and Vectors (Continued)

INTERRUPT NAME	OVERALL PRIORITY	CPU INTERRUPT AND VECTOR ADDRESS	BIT POSITION IN PIRQRx AND PIACKRx	PERIPHERAL INTERRUPT VECTOR (PIV)	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

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 10, column 3, CPU Interrupt and Vector Address. The PIE controller uses the CPU interrupt acknowledge to generate its internal signals to clear the current interrupt requests.

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 PIV is loaded into the Peripheral Interrupt Vector Register (PIVR) in the PIE controller and can then be read by the CPU to generate a branch to the respective Interrupt Service Routine (ISR).

In effect, there are two vector tables: a CPU vector table and a user-specified peripheral vector table. The CPU's vector table, which starts at 0000h, is used to get to the General Interrupt Service Routine (GISR) in response to a CPU interrupt request (INT). A user-specified peripheral vector table is employed 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 PIV to generate a branch to the SISR.

interrupt vectors (continued)

The peripheral interrupt vectors (PIVs) 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 PIVs are all implemented as hard-coded values on the F243/F241 devices, according to Table 10, column 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 Event-Specific Interrupt Service Routine (SISR). There is one GISR for each maskable prioritized request (INT) to the CPU. This can perform necessary context saves before it fetches the PIV from the PIVR. This PIV value 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 any of the INT1 – INT6 interrupts are serviced. 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 in the peripheral register, 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. The interrupt flag (IF) in the peripheral register should be cleared by software only. If the IF bit is not cleared after the respective interrupt service, future interrupts will not be recognized.
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.
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. It then masks all other maskable interrupts by setting the INTM bit, saves some context, clears the respective IFR bit, and starts 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 10, column 3.
4. The PIE controller decodes the PAB value and generates an internal peripheral interrupt acknowledge to load the PIV into the PIVR. The appropriate peripheral interrupt vector (or the phantom interrupt vector), is referenced from the table stored in the PIE controller.
5. When the GISR has completed any necessary context saves, it reads the PIVR and uses the interrupt vector as a target (or to generate a target) for a branch to the Event-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.

NOTE: If an interrupt occurs during the execution of a CLRC INTM instruction, the device always completes CLRC INTM as well as the next instruction before the pending interrupt is processed. This ensures that a return instruction that directly follows CLRC INTM will be executed before an interrupt is processed. The return instruction will pop the previous return address off the top of the stack before the new return address is pushed onto the stack. To allow the CPU to complete the return, interrupts are also blocked after a RET instruction until at least one instruction at the return address is executed. Interrupts may be blocked for more than one instruction if the instruction at the return address requires additional blocking for pipeline protection. If you want an ISR to occur *within* the current ISR rather than *after* the current ISR, place the CLRC INTM instruction more than one instruction before the return (RET) instruction.

external interrupts

The F243/F241 devices have 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 I/O 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.
- **$\overline{\text{NMI}}$.** This is a nonmaskable external interrupt.
- **$\overline{\text{PDPINT}}$.** This interrupt is provided for safe operation of power converters and motor drives controlled by the F243/F241. 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. $\overline{\text{PDPINT}}$ is a Level 1 interrupt.

Table 11 is a summary of the external interrupt capability of the F243/F241.

Table 11. External Interrupt Types and Functions

EXTERNAL INTERRUPT	CONTROL REGISTER NAME	CONTROL REGISTER ADDRESS	MASKABLE?
XINT1	XINT1CR	7070h	Yes (Level 1 or 6)
XINT2	XINT2CR	7071h	Yes (Level 1 or 6)
$\overline{\text{NMI}}$	—	—	No
$\overline{\text{PDPINT}}$	EVIMRA	742Ch	Yes (Level 1)

clock generation

The F243/F241 devices have 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 a fundamental crystal. The “times 4” (x4) option for the F243/F241 PLL is fixed and cannot be changed.

The PLL-based clock module provides two modes of operation:

- Crystal-operation
  This mode allows the use of a 5-MHz external reference crystal to provide the time base to the device.
- External clock source operation
  This mode allows the internal oscillator to be bypassed. The device clocks are generated from an external clock source input on the XTAL1/CLKIN pin. In this case, an external oscillator clock is connected to the XTAL1/CLKIN pin.

The clock module includes two external pins:

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

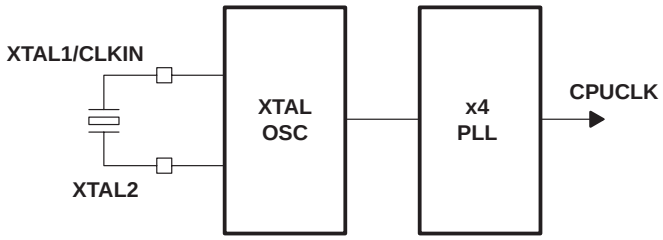


Figure 7. PLL Clock Module Block Diagram

low-power modes

The 24x 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 24x-based devices have two clock domains:

1. CPU clock domain – consists of the clock for most of the CPU logic
2. System clock domain – 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 24x CPU also contains support for a second IDLE mode, IDLE2. By asserting IDLE2 to the 24x 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.

low-power modes (continued)

Two control bits, LPM(1) and LPM(0), specify which of the three possible low-power modes is entered when the IDLE instruction is executed (see Table 12). These bits are located in the System Control and Status Register (SCSR) described in the *TMS320C241/C242/C243 DSP Controllers CPU, System, Instruction Set, and Peripherals Reference Guide* (literature number SPRU276).

Table 12. 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
CPU running normally	XX	On	On	On	On	On	—
IDLE1 – (LPM0)	00	Off	On	On	On	On	Peripheral Interrupt, External Interrupt, Reset
IDLE2 – (LPM1)	01	Off	Off	On	On	On	Wakeup Interrupts, External Interrupt, Reset
HALT – (LPM2) {PLL/OSC power down}	1X	Off	Off	Off	Off	Off	Reset Only

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

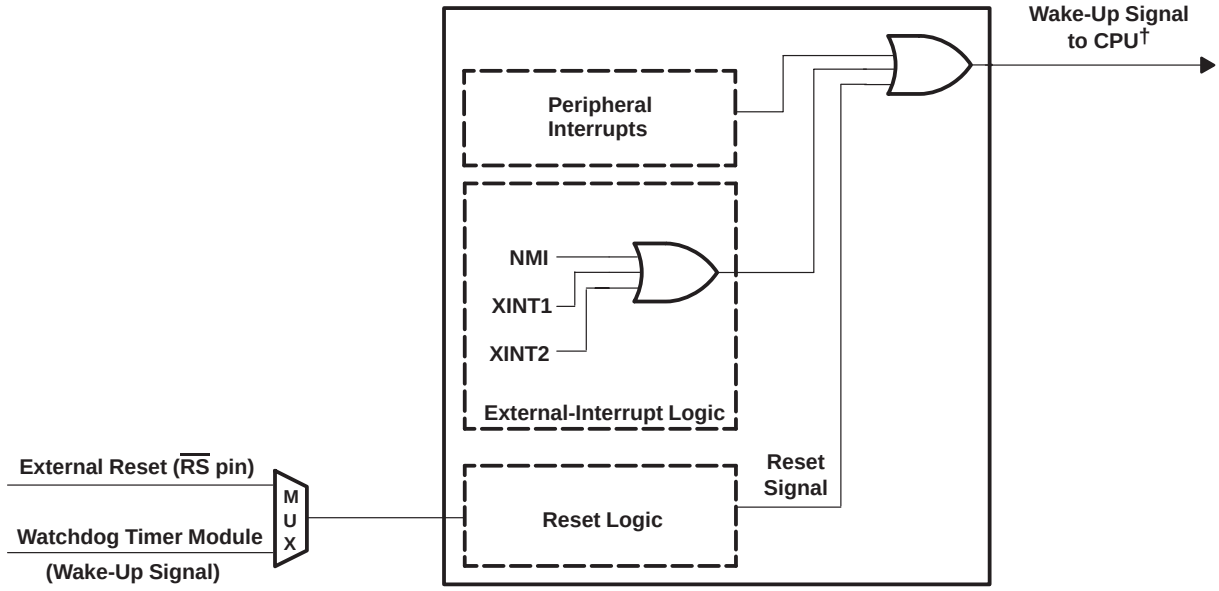
The external interrupts, XINTx, can cause the device to exit any of the low-power modes, except HALT. If the device is in IDLE2 mode, the synchronous logic connected to the external interrupt pins is bypassed with combinatorial logic which recognizes the interrupt on the pin, starts the clocks, and then allows the clocked logic to generate an interrupt request to the PIE controller. Note that in Table 12, external interrupts include $\overline{\text{PDPINT}}$.

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.

peripheral interrupts

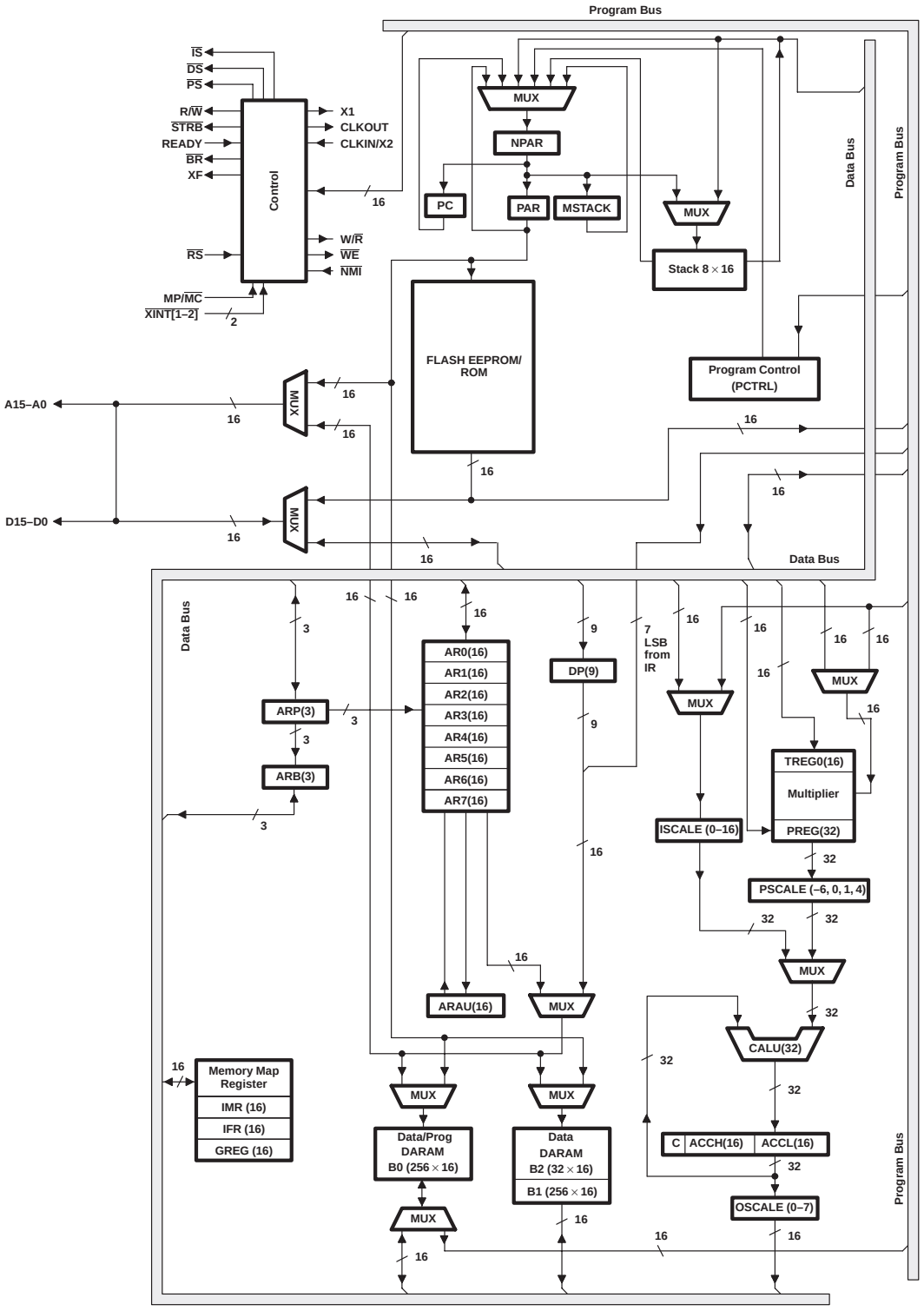
All peripheral interrupts, if enabled locally and globally, can cause the device to exit IDLE1 mode.



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

Figure 8. Waking Up the Device From Power Down

functional block diagram of the 24x DSP CPU



NOTES: A. Symbol descriptions appear in Table 13 and Table 14.
B. For clarity, the data and program buses are shown as single buses although they include address and data bits.

24x legend for the internal hardware

Table 13. Legend for the 24x Internal Hardware

SYMBOL	NAME	DESCRIPTION
ACC	Accumulator	32-bit register that stores the results and provides input for subsequent CALU operations. Also includes shift and rotate capabilities
ARAU	Auxiliary Register Arithmetic Unit	An unsigned, 16-bit arithmetic unit used to calculate indirect addresses using the auxiliary registers as inputs and outputs
AUX REGS	Auxiliary Registers 0–7	These 16-bit registers are used as pointers to anywhere within the data space address range. They are operated upon by the ARAU and are selected by the auxiliary register pointer (ARP). AR0 can also be used as an index value for AR updates of more than one and as a compare value to AR.
$\overline{BR}$	Bus Request Signal	$\overline{BR}$ is asserted during access of the external global data memory space. READY is asserted to the device when the global data memory is available for the bus transaction. $\overline{BR}$ can be used to extend the data memory address space by up to 32K words.
C	Carry	Register carry output from CALU. C is fed back into the CALU for extended arithmetic operation. The C bit resides in status register 1 (ST1), and can be tested in conditional instructions. C is also used in accumulator shifts and rotates.
CALU	Central Arithmetic Logic Unit	32-bit-wide main arithmetic logic unit for the TMS320C2xx core. The CALU executes 32-bit operations in a single machine cycle. CALU operates on data coming from ISCALE or PSCALE with data from ACC, and provides status results to PCTRL.
DARAM	Dual-Access RAM	If the on-chip RAM configuration control bit (CNF) is set to 0, the reconfigurable data dual-access RAM (DARAM) block B0 is mapped to data space; otherwise, B0 is mapped to program space. Blocks B1 and B2 are mapped to data memory space only, at addresses 0300–03FF and 0060–007F, respectively. Blocks 0 and 1 contain 256 words, while block 2 contains 32 words.
DP	Data Memory Page Pointer	The 9-bit DP register is concatenated with the seven least significant bits (LSBs) of an instruction word to form a direct memory address of 16 bits. DP can be modified by the LST and LDP instructions.
GREG	Global Memory Allocation Register	GREG specifies the size of the global data memory space. This register is not useful in the F241, since it has no external memory interface.
IMR	Interrupt Mask Register	IMR individually masks or enables the seven interrupts.
IFR	Interrupt Flag Register	The 7-bit IFR indicates that the TMS320C2xx has latched an interrupt from one of the seven maskable interrupts.
INT#	Interrupt Traps	A total of 32 interrupts by way of hardware and/or software are available.
ISCALE	Input Data-Scaling Shifter	16- to 32-bit barrel left-shifter. ISCALE shifts incoming 16-bit data 0 to16 positions left, relative to the 32-bit output within the fetch cycle; therefore, no cycle overhead is required for input scaling operations.
MPY	Multiplier	16 × 16-bit multiplier to a 32-bit product. MPY executes multiplication in a single cycle. MPY operates either signed or unsigned 2s-complement arithmetic multiply.
MSTACK	Micro Stack	MSTACK provides temporary storage for the address of the next instruction to be fetched when program address-generation logic is used to generate sequential addresses in data space.
MUX	Multiplexer	Multiplexes buses to a common input
NPAR	Next Program Address Register	NPAR holds the program address to be driven out on the PAB on the next cycle.
OSCALE	Output Data-Scaling Shifter	16- to 32-bit barrel left-shifter. OSCALE shifts the 32-bit accumulator output 0 to 7 bits left for quantization management and outputs either the 16-bit high- or low-half of the shifted 32-bit data to the data-write data bus (DWEB).
PAR	Program Address Register	PAR holds the address currently being driven on PAB for as many cycles as it takes to complete all memory operations scheduled for the current bus cycle.
PC	Program Counter	PC increments the value from NPAR to provide sequential addresses for instruction-fetching and sequential data-transfer operations.
PCTRL	Program Controller	PCTRL decodes instruction, manages the pipeline, stores status, and decodes conditional operations.

24x legend for the internal hardware (continued)

Table 13. Legend for the 24x Internal Hardware (Continued)

SYMBOL	NAME	DESCRIPTION
PREG	Product Register	32-bit register holds results of 16 × 16 multiply
PSCALE	Product-Scaling Shifter	0-, 1-, or 4-bit left shift, or 6-bit right shift of multiplier product. The left-shift options are used to manage the additional sign bits resulting from the 2s-complement multiply. The right-shift option is used to scale down the number to manage overflow of product accumulation in the CALU. PSCALE resides in the path from the 32-bit product shifter and from either the CALU or the data-write data bus (DWEB), and requires no cycle overhead.
STACK	Stack	STACK is a block of memory used for storing return addresses for subroutines and interrupt-service routines, or for storing data. The C24x stack is 16-bit wide and eight-level deep.
TREG	Temporary Register	16-bit register holds one of the operands for the multiply operations. TREG holds the dynamic shift count for the LACT, ADDT, and SUBT instructions. TREG holds the dynamic bit position for the BITT instruction.

F243/F241 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 F243/F241 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 9 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 14 lists status register field definitions.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	ARP			OV	OVM	1	INTM	DP								
ST1	ARB			CNF	TC	SXM	C	1	1	1	1	XF	1	1	PM	

Figure 9. Status and Control Register Organization

Table 14. 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.

status and control registers (continued)

Table 14. Status Register Field Definitions (Continued)

FIELD	FUNCTION
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. $\overline{RS}$ 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. $\overline{RS}$ also sets INTM. INTM has no effect on the unmaskable $\overline{RS}$ 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.
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 $\overline{RS}$.
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 instruction. 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 most significant bits (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

input scaling shifter (continued)

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 15.

Table 15. 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 a 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 the 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.

multiplier (continued)

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) instructions. 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.

central arithmetic logic unit (continued)

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.

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/x241 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 ±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.

internal memory

The TMS320x24x devices are configured with the following memory modules:

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

internal memory (continued)

dual-access RAM (DARAM)

There are 544 words × 16 bits of DARAM on the x243/x241 device. The x243/x241 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. The SETC CNF (configure B0 as data memory) and CLRC CNF (configure B0 as program memory) instructions allow dynamic configuration of the memory maps through software.

When using on-chip RAM, or high-speed external memory, the x243/x241 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/x241 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/x241 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

Flash EEPROM provides an attractive alternative to masked program ROM. Like ROM, flash is nonvolatile. However, it has the advantage of “in-target” reprogrammability. The F243/F241 incorporates one 8K × 16-bit flash EEPROM module in program space. Flash devices offer a cost-effective reprogrammable solution for volume production.

Unlike most discrete flash memory, the F243/F241 flash does not require a dedicated state machine, because the algorithms for programming and erasing the flash are executed by the DSP core. This enables several advantages, including: reduced chip size and sophisticated, adaptive algorithms. For production programming, the IEEE Standard 1149.1† (JTAG) scan port provides easy access to the on-chip RAM for downloading the algorithms and flash code. Other key features of the flash include zero-wait-state access rate and single 5-V power supply. Before programming, the flash EEPROM module generates the necessary voltages internally, making it unnecessary to provide the programming or erase voltages externally.

An erased bit in the flash is read as a logic 1, and a programmed bit is read as a logic 0. The flash requires a block-erase of the entire 8K module; however, any combination of bits can be programmed. The following four algorithms are required for flash operations: clear, erase, flash-write, and program. For an explanation of these algorithms and a complete description of the flash EEPROM, see the *TMS320F20x/F24x DSP Embedded Flash Memory Technical Reference* (literature number SPRU282).

illegal access detect

Any access to an illegal address asserts an NMI. This feature is useful to provide a graceful return back to the user code, should an illegal address be accessed inadvertently.

flash serial loader/utilities

The on-chip flash is shipped with a serial bootloader code programmed at the following addresses: 0000–00FFh. All other flash memory locations are in an erased state. The serial bootloader can be used to load flash-programming algorithms or code to any destination RAM through the on-chip serial communications interface (SCI). Refer to the TMS320F240 Serial Bootloader application note (located at **ftp://www.ti.com/**) to understand on-chip flash programming using the serial bootloader code. (Choose **/pub/tms320bbs/c24xfiles** at the main ftp directory to locate the **f240boot.pdf** file.) The latest TMS320F243/241 flash utilities should be available at **http://www.ti.com** which is the external TI web site.

† IEEE Standard 1149.1–1990, IEEE Standard Test Access Port.

peripherals

The integrated peripherals of the TMS320x24x are described in the following subsections:

- External memory interface (F243 only)
- Event-manager (EV2) module
- Analog-to-digital converter (ADC) module
- Serial peripheral interface (SPI) module
- Serial communications interface (SCI) module
- Controller area network (CAN) module
- Watchdog (WD) timer module

external memory interface (F243 only)

The TMS320F243 can address up to 64K × 16 words of memory (or registers) in each of the program, data, and I/O spaces. On-chip memory, when enabled, occupies some of this off-chip range. In data space, the high 32K words can be mapped dynamically either locally or globally using the global memory allocation register (GREG) as described in the *TMS320C241/C242/C243 DSP Controllers CPU, System, Instruction Set, and Peripherals Reference Guide* (literature number SPRU276). Access to a data-memory location, that is mapped as global, asserts the $\overline{BR}$ pin low.

The CPU of the TMS320F243 schedules a program fetch, data read, and data write on the same machine cycle. This is because from on-chip memory, the CPU can execute all three of these operations in the same cycle. However, the external interface multiplexes the internal buses to one address and one data bus. The external interface sequences these operations to complete first the data write, then the data read, and finally the program read.

The F243 supports a wide range of system interfacing requirements. Program, data, and I/O address spaces provide interface to memory and I/O, thereby maximizing system throughput. The full 16-bit address and data bus, along with the $\overline{PS}$, $\overline{DS}$, and $\overline{IS}$ space-select signals, allow addressing of 64K 16-bit words in program, data, and I/O space. Since on-chip peripheral registers occupy positions of data-memory space, the externally addressable data-memory space is 32K 16-bit words.

I/O design is simplified by having I/O treated the same way as memory. I/O devices are accessed in the I/O address space using the processor's external address and data buses in the same manner as memory-mapped devices.

The F243 external parallel interface provides various control signals to facilitate interfacing to the device. The R/$\overline{W}$ output signal is provided to indicate whether the current cycle is a read or a write. The $\overline{STRB}$ output signal provides a timing reference for all external cycles. For convenience, the device also provides the $\overline{RD}$ and the $\overline{WE}$ output signals, which indicate a read and a write cycle, respectively, along with timing information for those cycles. The availability of these signals minimizes external gating necessary for interfacing external devices to the F243.

The bus request ($\overline{BR}$) signal is used in conjunction with other F243 interface signals to arbitrate external global memory accesses. Global memory is external data memory space in which the $\overline{BR}$ signal is asserted at the beginning of the access. When an external global memory device receives the bus request, it responds by asserting the READY signal after the global memory access is arbitrated and the global access is completed.

The TMS320F243 supports zero-wait-state reads on the external interface. However, to avoid bus conflicts, writes take two cycles. This allows the TMS320F243 to buffer the transition of the data bus from input to output (or output to input) by a half cycle. In most systems, TMS320F243 ratio of reads to writes is significantly large to minimize the overhead of the extra cycle on writes.

external memory interface (F243 only) (continued)

Wait states can be generated when accessing slower external resources. The wait states operate on machine-cycle boundaries and are initiated either by using the READY pin or using the software wait-state generator. READY pin can be used to generate any number of wait states. When using the READY pin to communicate with slower devices, the F243 processor waits until the slower device completes its function and signals the processor by way of the READY line. Once a ready indication is provided back to the F243 from the external device, execution continues. For external wait states using the READY pin, the on-chip wait-state generator must be programmed to generate at least one wait state.

wait-state generation (F243 only)

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.

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) (I/O FFFFh). For more detailed information on the WSGR and associated bit functions, refer to the *TMS320C241/C242/C243 DSP Controllers CPU, System, Instruction Set, and Peripherals Reference Guide* (literature number SPRU276).

event-manager (EV2) module

The event-manager module includes general-purpose (GP) timers, full compare/PWM units, capture units, and quadrature-encoder pulse (QEP) circuits. Figure 10 shows the functions of the event manager.

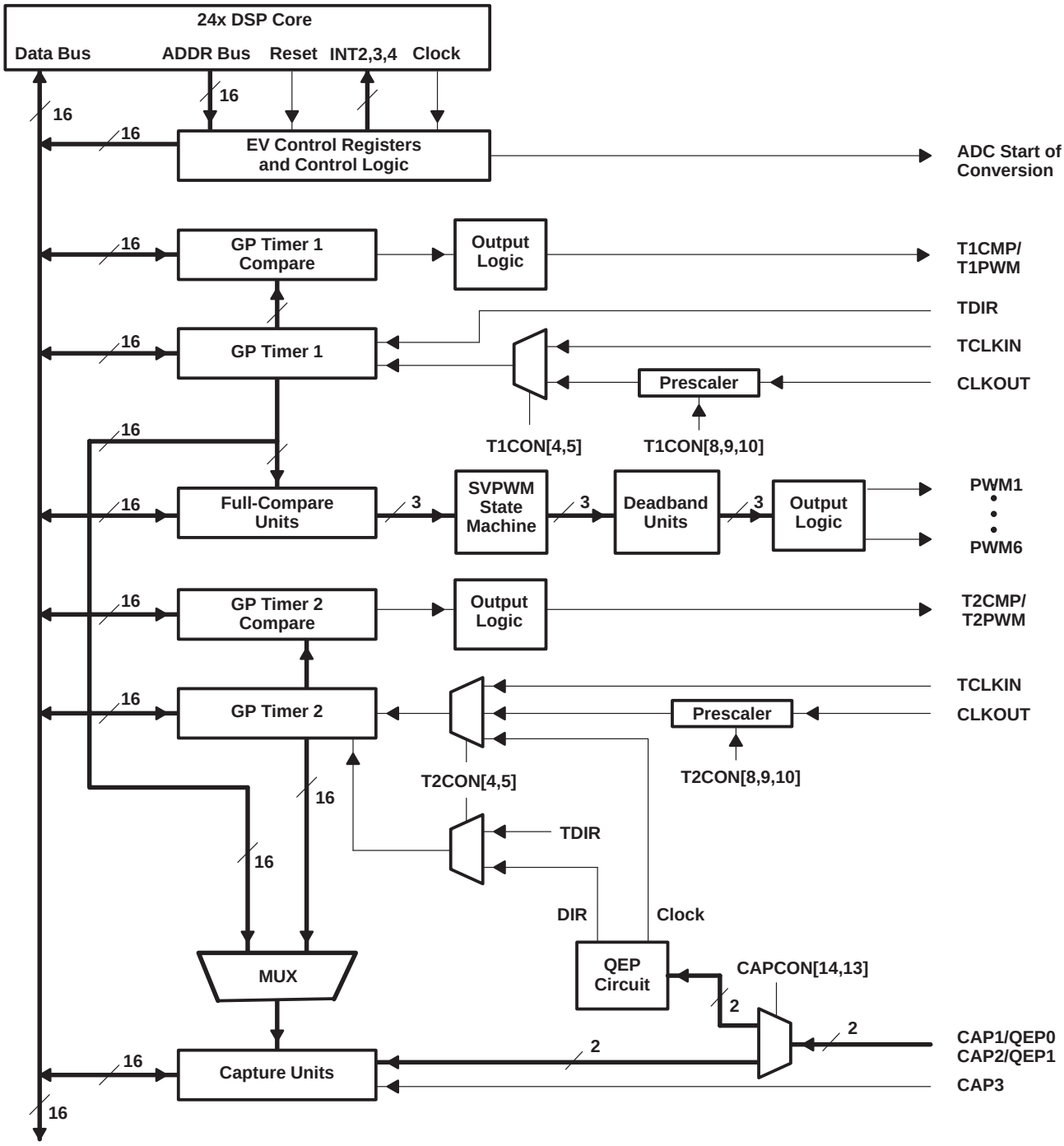


Figure 10. Event-Manager Block Diagram

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-/down-counter, TxCNT, for reads or writes
- A 16-bit timer-compare register, TxCMPR (double-buffered with shadow register), for reads or writes
- A 16-bit timer-period register, TxPR (double-buffered with shadow register), 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 input pin (TDIR) (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 2/1 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 time base 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 values (from 0 to 24 μs) 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 follows:

- 16-bit registers
- Programmable deadband for the PWM output pairs, from 0 to 24 μs
- 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 three 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
  - Selection of GP timer 2 as the time base
  - Three 16-bit 2-level-deep FIFO stacks, one for each capture unit
  - Three 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 11. 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 F243/F241. Eight analog inputs are provided by way of an 8-to-1 analog multiplexer. Maximum total conversion time for each ADC unit is 1 μs. Reference voltage for the ADC module is 0–5 V and is supplied 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 (ADCSOC), 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 2-level-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 fact, it uses only 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 simultaneoulsy with one conversion request. V_{CCA} and V_{SSA} pins must be connected to 5 V and analog ground, respectively. Standard isolation techniques must be used while applying power to the ADC module.

The ADC module, shown in Figure 11, has the following features:

- Up to 8 analog inputs, ADCIN00–ADCIN07. The results from converting the inputs ADCIN00–ADCIN07 are placed in one of the ADCFIFO results registers (see Table 16). The digital value of the input analog voltage is derived by:

$$\text{Digital Value} = 1023 \times \frac{\text{Input Analog Voltage} - V_{REFLO}}{V_{REFHI} - V_{REFLO}}$$

- 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 vaules of completed conversions
- Two programmable ADC module control registers (see Table 16)
- Programmable clock prescaler
- Interrupt or polled operation

A/D overview (continued)

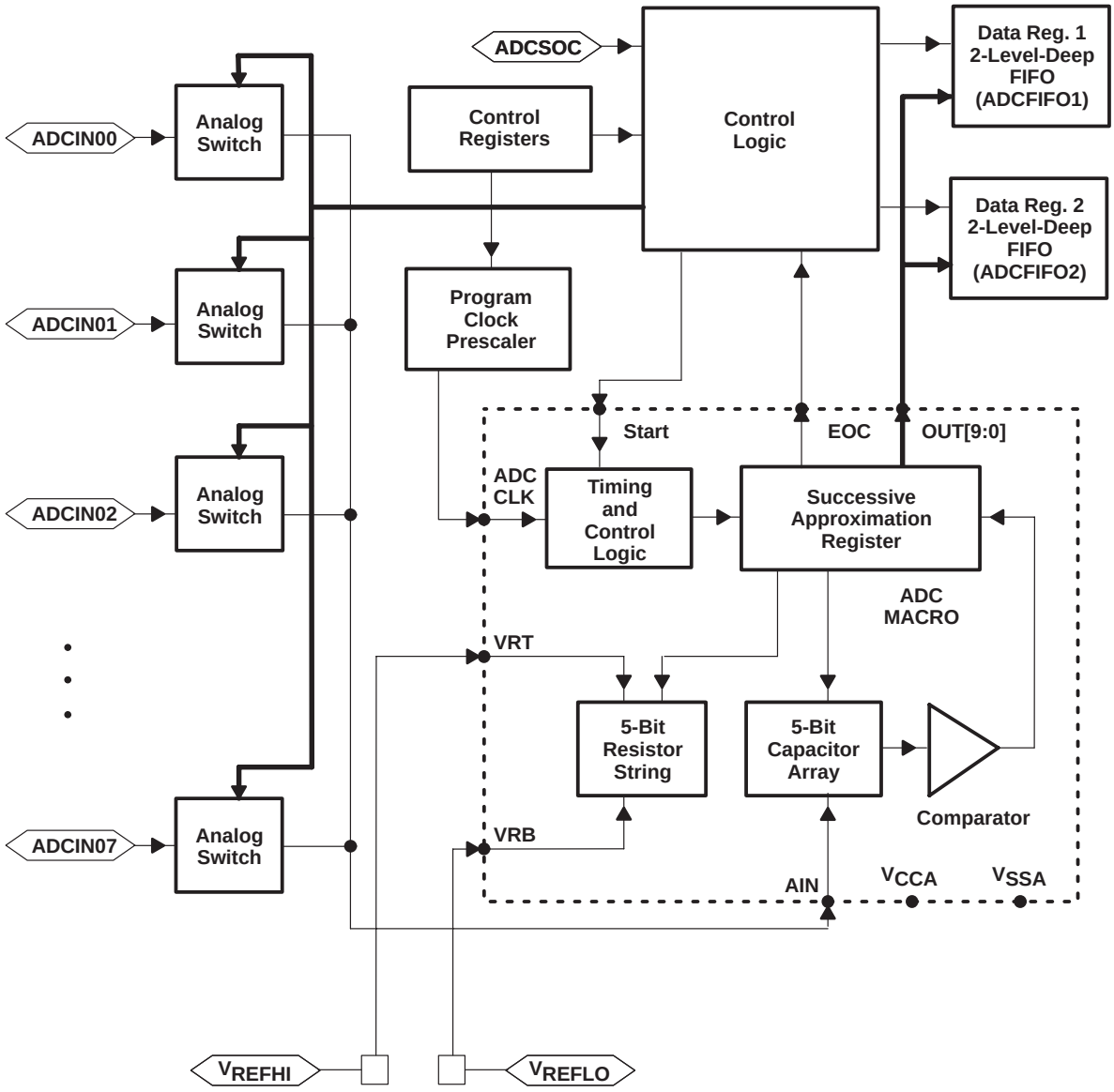


Figure 11. F243/F241 Pseudo Dual Analog-to-Digital Converter (ADC) Module

Table 16. 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 F243/F241 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
  - SPISIMO: SPI slave-input/master-output pin
  - SPISTE: SPI slave transmit-enable pin
  - SPICLK: SPI serial-clock pin

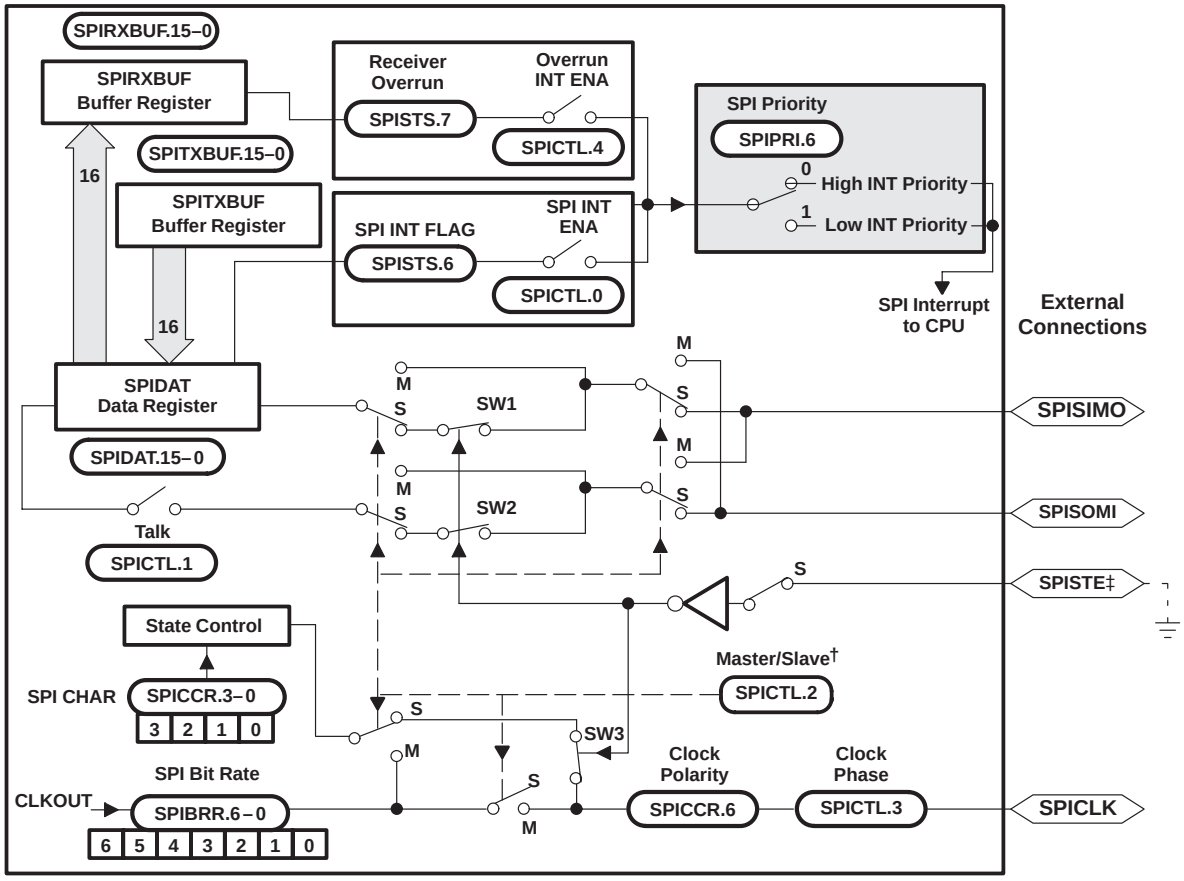
NOTE: All these four pins can be used as GPIO, if the SPI module is not used.

- Two operational modes: master and slave
- Baud rate: 125 different programmable rates/5 Mbps at 20-MHz CPUCLK
- Data word length: one to sixteen 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.
- Eight 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 12 is a block diagram of the SPI in slave mode.

serial peripheral interface (SPI) module (continued)



† The diagram is shown in slave mode.
‡ The SPISTE pin is shown enabled, meaning the data can be transmitted or received in this mode. Note that switches SW1, SW2, and SW3 are closed in this configuration. The “switches” are assumed to close when their “control signal” is high.

Figure 12. Four-Pin Serial Peripheral Interface Module Block Diagram

serial communications interface (SCI) module

The F243/F241 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 bit rate 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
  - SCIRXD: SCI receive-input pin

NOTE: Both pins can be used as GPIO if not used for SCI.

- 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 is 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
- Ten 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 13 shows the SCI module block diagram.

serial communications interface (SCI) module (continued)

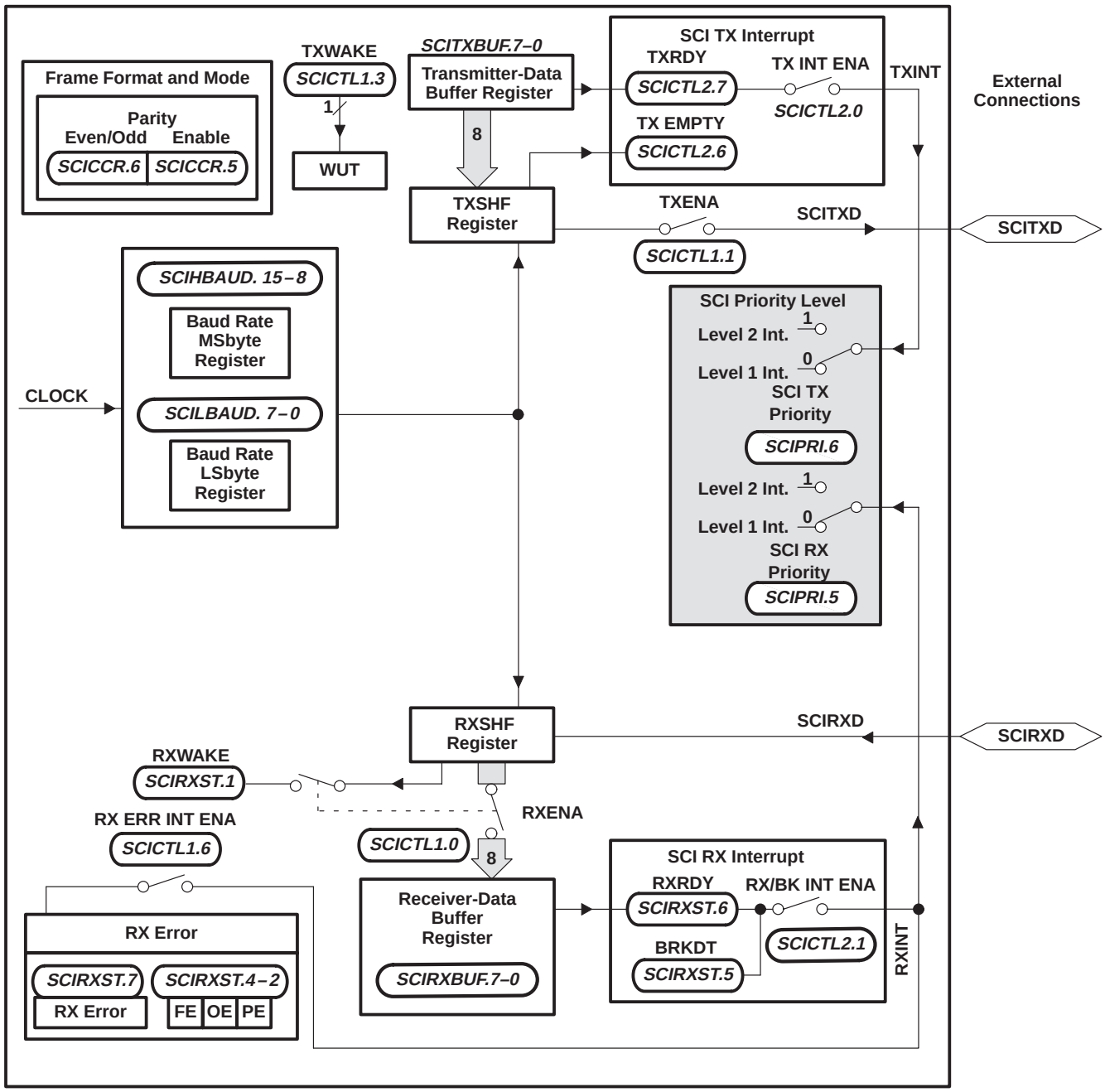


Figure 13. Serial Communications Interface (SCI) Module Block Diagram

controller area network (CAN) module

The CAN peripheral supports the following features:

- Full implementation of CAN protocol, version 2.0B
  - Standard and extended identifiers
  - Data and remote frames
- Six mailboxes for objects of 0- to 8-bytes data length
  - Two receive mailboxes (MBOX0,1), two transmit mailboxes (MBOX4,5)
  - Two configurable transmit/receive mailboxes (MBOX2,3)
- Local acceptance mask registers (LAMn) for mailboxes 0 and 1 and mailboxes 2 and 3
- Programmable bit rate
- Programmable interrupt scheme
- Programmable wake up on bus activity
- Automatic reply to a remote request
- Automatic re-transmission in case of error or loss of arbitration
- Bus failure diagnostic
  - Bus on/off
  - Error passive/active
  - Bus error warning
  - Bus stuck dominant
  - Frame error report
  - Readable error counter
- Self-Test Mode
  - The CAN peripheral operates in a loop back mode
  - Receives its own transmitted message and generates its own acknowledge signal
- Two-Pin Communication
  - The CAN module uses two pins for communication, CANTX and CANRX
  - These two pins are connected to a CAN transceiver chip, which in turn is connected to a CAN bus

controller area network (CAN) module (continued)

Table 17. Register Addresses†

Address	Name	Description
7100h	MDER	Mailbox Direction/Enable Register (bits 7 to 0)
7101h	TCR	Transmission Control Register (bits 15 to 0)
7102h	RCR	Receive Control Register (bits 15 to 0)
7103h	MCR	Master Control Register (bits 13 to 6, 1, 0)
7104h	BCR2	Bit Configuration Register 2 (bits 7 to 0)
7105h	BCR1	Bit Configuration Register 1 (bits 10 to 0)
7106h	ESR	Error Status Register (bits 8 to 0)
7107h	GSR	Global Status Register (bits 5 to 0)
7108h	CEC	Transmit and Receive Error Counters (bits 15 to 0)
7109h	CAN_IFR	Interrupt Flag Register (bits 13 to 8, 6 to 0)
710Ah	CAN_IMR	Interrupt Mask Register (bits 15, 13 to 0)
710Bh	LAM0_H	Local Acceptance Mask for MBOX0 and 1 (bits 31, 28 to 16)
710Ch	LAM0_L	Local Acceptance Mask for MBOX0 and 1 (bits 15 to 0)
710Dh	LAM1_H	Local Acceptance Mask for MBOX2 and 3 (bits 31, 28 to 16)
710Eh	LAM1_L	Local Acceptance Mask for MBOX2 and 3 (bits 15 to 0)
710Fh	Reserved	Accesses assert the CAADDRx signal from the CAN peripheral (which will assert 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 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.
- 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

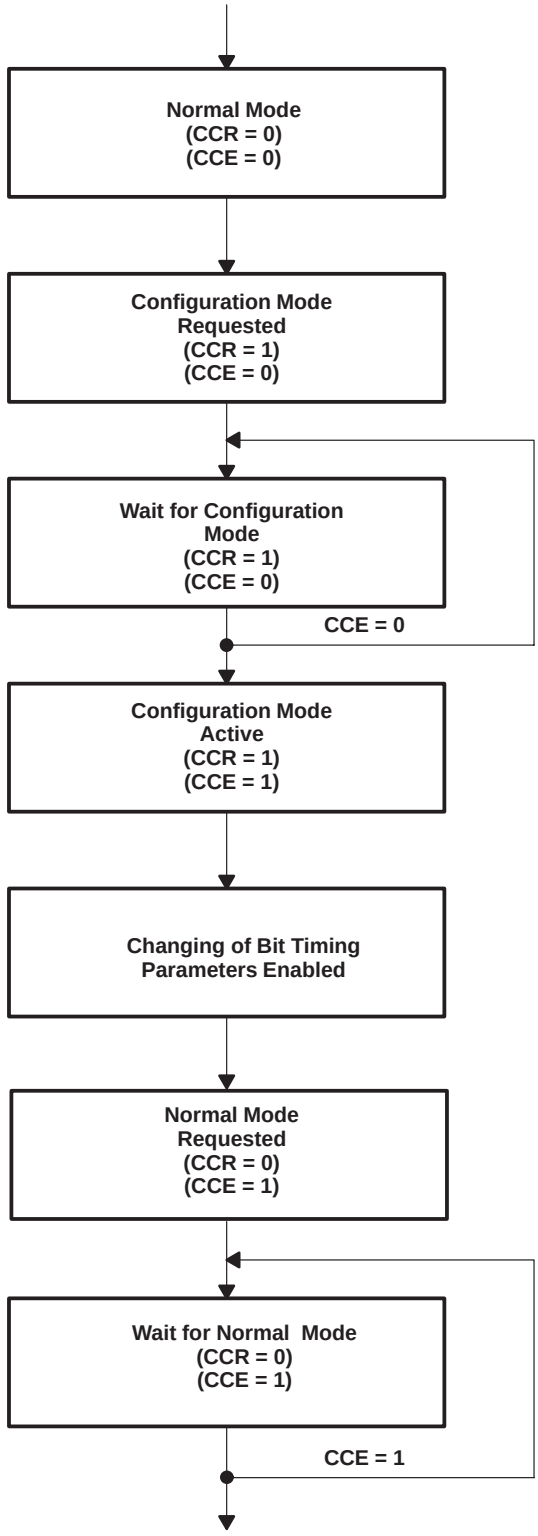


Figure 14. CAN Initialization

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 the CCR bit of the MCR register with “1”. Only if the status bit CCE (GSR.4) 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.

watchdog (WD) timer module

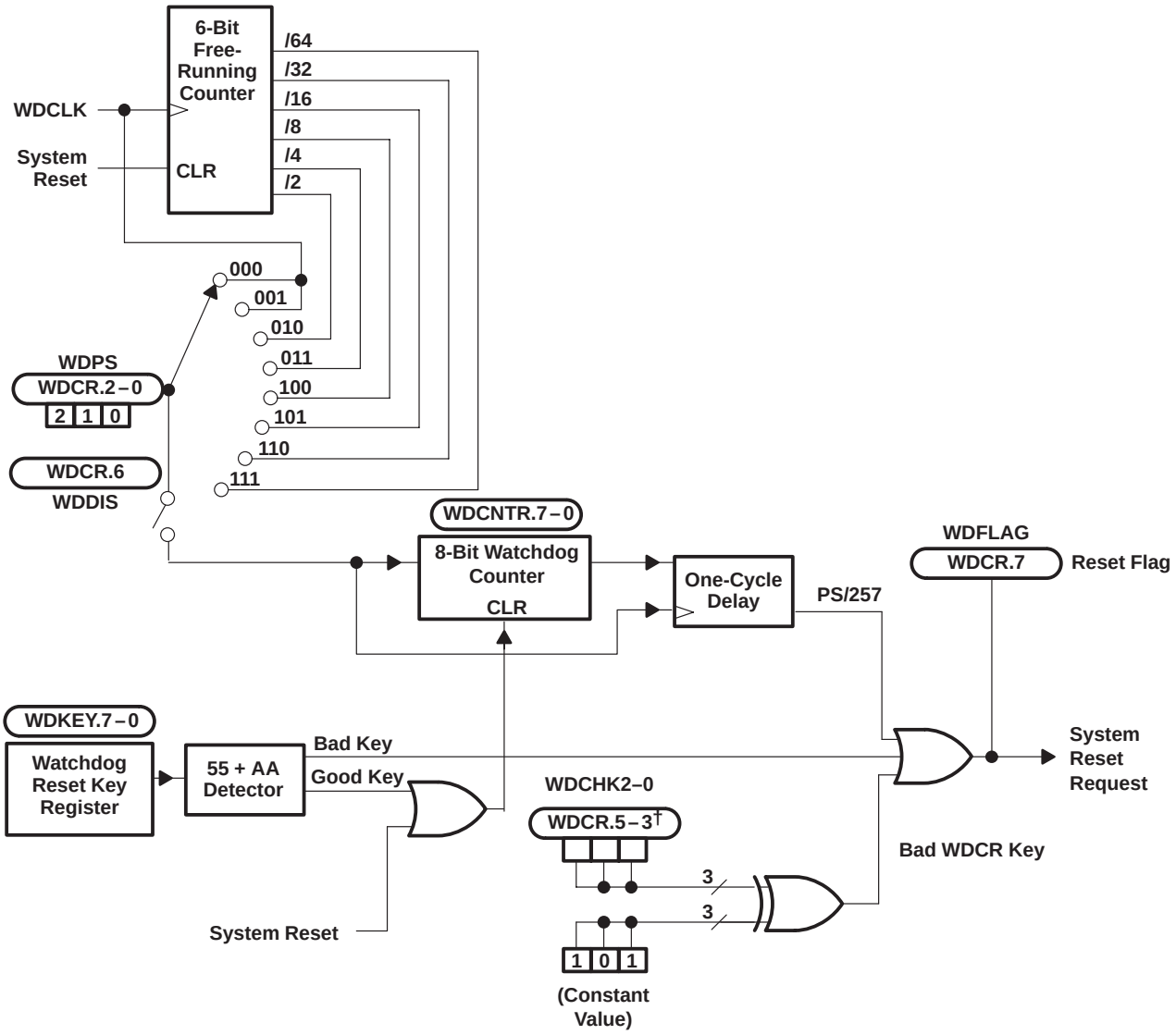
The F243/F241 devices include a watchdog (WD) timer module. The WD function of this module monitors software and hardware operation by generating a system reset if it is not periodically serviced by software by having the correct key written. The WD timer operates independently of the CPU and is always 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 15 for a block diagram of the WD module. The WD module features include the following:

- WD Timer
  - Seven different WD overflow rates ranging from 6.55 ms to 419.43 ms
  - 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 15 shows the WD block diagram. Table 18 shows the different WD overflow (timeout) selections.

watchdog (WD) timer module (continued)



† Writing to bits WDCR.5 – 3 with anything but the correct pattern (101) generates a system reset.

Figure 15. Block Diagram of the WD Module

watchdog (WD) timer module (continued)

Table 18. 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-MHz clock
‡ X = Don't care

scan-based emulation

TMS320x2xx devices incorporate scan-based emulation logic for code-development and hardware-development support. 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. The scan-based emulator communicates with the x2xx by way of the IEEE 1149.1-compatible (JTAG) interface. The F243 and F241 DSPs, like the TMS320F206, TMS320C203, and TMS320LC203, do not include boundary scan. The scan chain of these devices is useful for emulation function only.

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 TMS320C1x™ (C1x™) and TMS320C2x™ (C2x™) generation DSPs is upwardly compatible with the x243/x241 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.

addressing modes (continued)

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.

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 20) 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 internal 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 19 are used in the instruction set summary table (Table 20). 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 21. 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 21. 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 19. 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. T P Meaning 0 0 $\overline{\text{BIO}}$ low 0 1 TC=1 1 0 TC=0 1 1 None of the above conditions
TREGn	Temporary register n (n = 0, 1, or 2)
Z L V C	4-bit field representing the following conditions: Z: ACC = 0 L: ACC < 0 V: Overflow C: Carry 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 ACC ≥ 0, 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 ACC = 0, and the L field is reset to indicate testing of the condition ACC ≥ 0. 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.

instruction set summary (continued)

Table 20. 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 16-Bit Constant
	AND immediate with accumulator with shift of 16	2/2	1011	1110	1000	0001 16-Bit Constant
APAC	Add P register to accumulator	1/1	1011	1110	0000	0100
B	Branch unconditionally	2/4	0111	1001	IADD	RESS Branch Address
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 Branch Address
BCND	Branch if TC bit ≠ 0	2/4/2	1110	0001	0000	0000 Branch Address
	Branch if TC bit = 0	2/4/2	1110	0010	0000	0000 Branch Address
	Branch on carry	2/4/2	1110	0011	0001	0001 Branch Address
	Branch if accumulator ≥ 0	2/4/2	1110	0011	1000	1100 Branch Address
	Branch if accumulator > 0	2/4/2	1110	0011	0000	0100 Branch Address
	Branch on I/O status low	2/4/3	1110	0000	0000	0000 Branch Address
	Branch if accumulator ≤ 0	2/4/2	1110	0011	1100	1100 Branch Address
	Branch if accumulator < 0	2/4/2	1110	0011	0100	0100 Branch Address
	Branch on no carry	2/4/2	1110	0011	0000	0001 Branch Address
	Branch if no overflow	2/4/2	1110	0011	0000	0010 Branch Address

instruction set summary (continued)

Table 20. TMS320x24x Instruction Set Summary (Continued)

x24x MNEMONIC	DESCRIPTION	WORDS/ CYCLES	OPCODE			
			MSB			LSB
BCND	Branch if accumulator ≠ 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†	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

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

instruction set summary (continued)

Table 20. 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
			16BIT	I/O	PORT	ADRS
PAC	Load accumulator with P register	1/1	1011	1110	0000	0011

instruction set summary (continued)

Table 20. 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 20. 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
			16-Bit Constant			
	Exclusive-OR immediate with accumulator with shift of 16	2/2	1011	1110	1000	0011
			16-Bit Constant			
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 DSP Development Support Reference Guide* (literature number SPRU011) contains information about development support products for all TMS320™ DSP family member devices, including documentation. Refer to this document for further information about TMS320™ DSP documentation or any other TMS320™ DSP support products from Texas Instruments. There is also an additional document, the *TMS320 Third-Party Support Reference Guide* (literature number SPRU052), which contains information from other companies in the industry about products related to the TMS320™ DSP. To receive copies of TMS320™ DSP literature, contact the Literature Response Center at 800/477-8924.

See Table 21 and Table 22 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 21. Development Support Tools

DEVELOPMENT TOOL	PLATFORM	PART NUMBER
Software – Code Generation Tools		
Assembler/Linker	PC™, OS/2™, Windows 3.x/Windows™ 95	TMDS3242850-02
C Compiler/Assembler/Linker	PC, OS/2, Windows 3.x/Windows 95	TMDS3242855-02
C Compiler/Assembler/Linker	Open Windows, HP9000, SPARC™	TMDS3242555-08
Software – Simulation		
C2xx Simulator	SPARC, Open Windows	TMDX324x551-09
Software – Emulation Debug Tools		
Code Composer 4.10, Code Generation 7.0	PC, Windows 3.x, OS/2	TMDS324012xx
TI C Source Debugger – WS	SPARC, SunOS™	TMDX324062xx
Hardware – Emulation Debug Tools		
XDS510XL™ Board (ISA card), w/JTAG cable	PC	TMDS00510
XDS510PP™ Pod (Parallel Port) w/JTAG cable	PC	TMDS00510PP
XDS510WS™ Box w/JTAG cable	SPARC	TMDS00510WS

Table 22. TMS320x24x-Specific Development Tools

DEVELOPMENT TOOL	PLATFORM	PART NUMBER
Hardware – Evaluation/Starter Kits		
TMS320LF2407 EVM	PC, Windows 95, Windows 98	TMDX3P701016
TMS320F240 EVM	PC, Windows 3.x	TMDX326P124X
TMS320F243 EVM	PC, Windows 95	TMDS3P604030

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™ DSP devices and support tools. Each TMS320™ DSP family 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

PC and OS/2 are trademarks of International Business Machines Corp.
Windows is a registered trademark of Microsoft Corporation.
SPARC is a trademark of SPARC International, Inc.
SunOS is a trademark of Sun Microsystems, Inc.
XDS510XL, XDS510PP, and XDS510WS are trademarks of Texas Instruments.

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

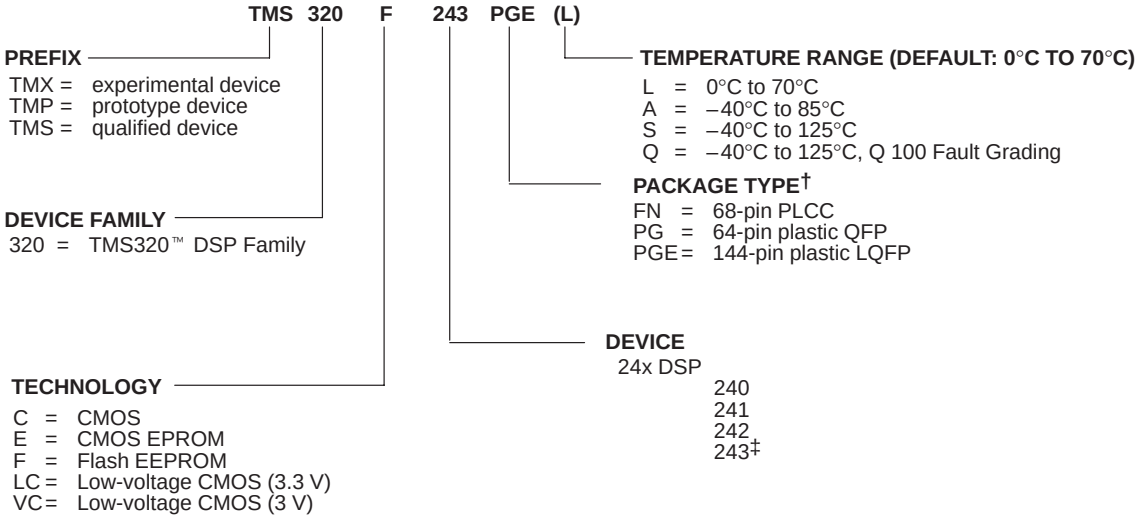
TMX and TMP devices and TMDX development support tools are shipped against the following disclaimer:

"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, FN, PG, and PGE) and temperature range (for example, L). Figure 16 provides a legend for reading the complete device name for any TMS320x2xx family member.



† PLCC = Plastic J-Leaded Chip Carrier
QFP = Quad Flatpack
LQFP = Low-Profile Quad Flatpack

‡ The package dimensions of the 243 device correspond to the LQFP package. This device was stated to be in QFP packaging in previous data sheets. The package dimensions have *not* changed; only the package designation has changed.

Figure 16. TMS320x24x DSP Device Nomenclature

documentation support

Extensive documentation supports all of the TMS320™ DSP 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.

A series of DSP textbooks is published by Prentice-Hall and John Wiley & Sons to support digital signal processing research and education. The TMS320™ DSP newsletter, *Details on Signal Processing*, is published quarterly and distributed to update TMS320™ DSP customers on product information.

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

To send comments regarding the F243/F241 datasheet (SPRS064), use the *comments@books.sc.ti.com* email address, which is a repository for feedback. For questions and support, contact the Product Information Center listed at the **http://www.ti.com/sc/docs/pic/home.htm** site.

Silicon enhancements and errata pertaining to peripherals (such as CAN) are posted on the TI web site and are updated as and when required.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage range, V_{DD}‡ . . . . . –0.3 V to 7 V
Input voltage range . . . . . –0.3 V to 7 V
Output voltage range . . . . . –0.3 V to 7 V
Input clamp current I_{IK} ($V_I < 0$ or $V_I > V_{DD}$) . . . . . ± 20 mA
Output clamp current I_{OK} ($V_O < 0$ or $V_O > V_{DD}$) . . . . . ± 20 mA
Operating free-air temperature range, T_A: L version(F243/F241) . . . . . 0°C to 70°C
A version(F243/F241) . . . . . –40°C to 85°C
S version(F241) . . . . . –40°C to 125°C
Storage temperature range, T_{stg} . . . . . –55°C to 150°C

† 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.
‡ 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
		All other inputs	2		V_{DD} + 0.3	
V_{IL}	Low-level input voltage	XTAL1/CLKIN	–0.3		0.7	V
		All other inputs	–0.3		0.7	
I_{OH}	High-level output current, V_{OH} = 2.4 V	All outputs			8	mA
I_{OL}	Low-level output current, V_{OL} = 0.7 V	All outputs			8	mA
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	
		S version	–40		125	
T_{FP}	Flash programming on flash devices, temperature		–40		85	°C

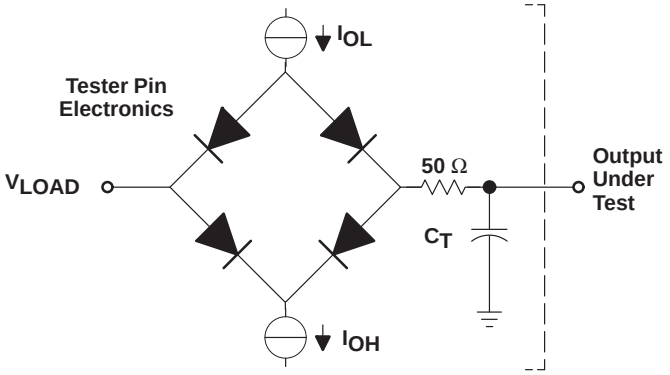
§ Thermal resistance values, Θ_{JA} (junction-to-ambient) and Θ_{JC} (junction-to-case) for the F243/F241 can be found on the mechanical package pages.

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

PARAMETER			TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage		I_{OH} = MAX = 8 mA		2.4			V
V_{OL}	Low-level output voltage		I_{OL} = MAX = 8 mA				0.7	V
I_I	Input current		Pins with pulldown	$V_{IN} = V_{DD}$	65	150	350	µA
				V_{IN} = 0 V	–5		5	
			Pins with pullup	$V_{IN} = V_{DD}$	–5		5	µA
				V_{IN} = 0 V	–350	–150	–65	
			All other input-only pins		–5		5	µA
I_{OZ}	Output current, high-impedance state (off-state)		$V_O = V_{DD}$ or 0 V		–5	1	5	µA
I_{DD}†	Supply current, operating mode		$t_{c(CO)}$ = 50 ns	243		120		mA
				241		90		
	Supply current, Idle 1 low-power mode	LPM0	$t_{c(CO)}$ = 50 ns			55		mA
	Supply current, Idle 2 low-power mode	LPM1	$t_{c(CO)}$ = 50 ns			30		
	Supply current, PLL/OSC power-down mode	LPM2				5		mA
C_i	Input capacitance					15		pF
C_o	Output capacitance					15		pF

† In operating mode, the CPU is running a dummy code in B0 program memory. In all IDLE modes, the CPU is idle in B0 program memory.

PARAMETER MEASUREMENT INFORMATION



Where: I_{OL} = 2 mA (all outputs)
I_{OH} = 300 µA (all outputs)
V_{LOAD} = 1.5 V
C_T = 110-pF typical load-circuit capacitance

Figure 17. 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 18 shows the TTL-level outputs.

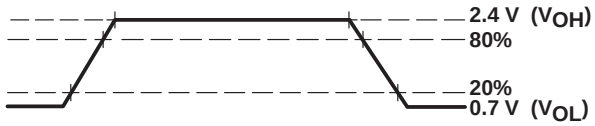


Figure 18. TTL-Level Outputs

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.

Figure 19 shows the TTL-level inputs.

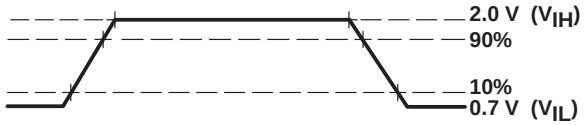


Figure 19. TTL-Level Inputs

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. 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 $\overline{RD}$
D	D[15:0]	RS	RESET pin $\overline{RS}$
INT	$\overline{NMI}$, XINT1, XINT2	W	Write cycle or $\overline{WE}$

Lowercase subscripts and their meanings:		Letters and symbols and their meanings:	
a	access time	H	High
c	cycle time (period)	L	Low
d	delay time	V	Valid
f	fall time	X	Unknown, changing, or don't care level
h	hold time	Z	High impedance
r	rise time		
su	setup time		
t	transition time		
v	valid time		
w	pulse duration (width)		

general notes on timing parameters

All output signals from the F243/F241 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.

CLOCK CHARACTERISTICS AND TIMINGS

clock options

PARAMETER
PLL multiply-by-4

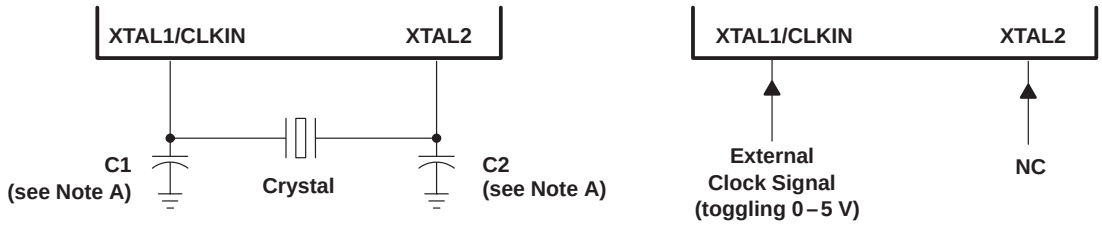
The F243/F241 devices include an on-chip PLL which is hardwired for multiply-by-4 operation. This requires the use of a 5-MHz clock input frequency for 20-MHz device operation. This input clock can be provided from either an external reference crystal or oscillator.

external reference crystal clock option

The internal oscillator is enabled by connecting a crystal across XTAL1/CLKIN and XTAL2 pins as shown in Figure 20a. The crystal should be in fundamental operation and parallel resonant, with an effective series resistance of 30 Ω and a power dissipation of 1 mW; it should be specified at a load capacitance of 20 pF.

external reference oscillator clock option

The internal oscillator is disabled by connecting a TTL-level clock signal to XTAL1/CLKIN and leaving the XTAL2 input pin unconnected as shown in Figure 20b.



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

(a) **(b)**

Figure 20. Recommended Crystal / Clock Connection

external reference crystal/clock with PLL circuit enabled

The internal oscillator is enabled by connecting a crystal across XTAL1/CLKIN and XTAL2 pins as shown in Figure 20a. The crystal should be in fundamental operation and parallel resonant, with an effective series resistance of 30 Ω and a power dissipation of 1 mW; it should be specified at a load capacitance of 20 pF.

timings with the PLL circuit enabled

PARAMETER			MIN	TYP	MAX	UNIT
f_X	Input clock frequency	Oscillator	1		5	MHz
		CLKIN	1		5	MHz
C1, C2	Load capacitance			10		pF

switching characteristics over recommended operating conditions [H = 0.5 $t_{c(CO)}$] (see Figure 21)

PARAMETER		CLOCK MODE	MIN	TYP	MAX	UNIT
$t_{c(CO)}$	Cycle time, CLKOUT		50			ns
$t_{f(CO)}$	Fall time, CLKOUT			4		ns
$t_{r(CO)}$	Rise time, CLKOUT			4		ns
$t_{w(COL)}$	Pulse duration, CLKOUT low		H–3	H	H+3	ns
$t_{w(COH)}$	Pulse duration, CLKOUT high		H–3	H	H+3	ns
t_p	Transition time, PLL synchronized after PLL enabled	before PLL lock, CLKIN multiply by 4			$2500t_{c(CI)}$	ns

timing requirements (see Figure 21)

		EXTERNAL REFERENCE CRYSTAL	MIN	MAX	UNIT
$t_{c(CI)}$	Cycle time, XTAL1/CLKIN	5 MHz	200		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	%

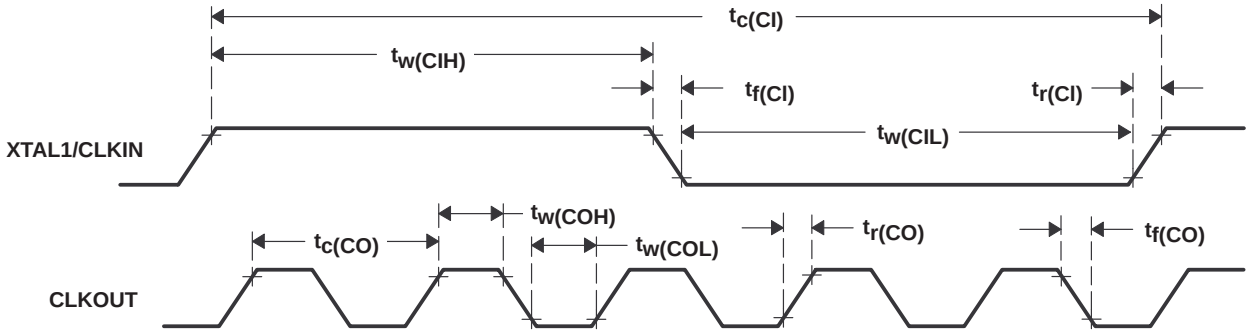


Figure 21. CLKIN-to-CLKOUT Timing for PLL Oscillator Mode, Multiply-by-4 Option with 5-MHz Clock

low-power mode timings

switching characteristics over recommended operating conditions [H = 0.5$t_{c(CO)}$] (see Figure 22 and Figure 23)

PARAMETER		LOW-POWER MODES		MIN	TYP	MAX	UNIT
$t_{d(WAKE-A)}$	Delay time, CLKOUT switching to program execution resume	IDLE1/IDLE2	LPM0 LPM1		4 + 6 $t_{c(CO)}$	15 × $t_{c(CO)}$	ns
$t_{d(IDLE-COH)}$	Delay time, Idle instruction executed to CLKOUT high	HALT {PLL/OSC power down}	LPM2		$4t_{c(CO)}$		ns
$t_{d(WAKE-OSC)}$	Delay time, wakeup interrupt asserted to oscillator running				OSC start-up and PLL lock time		ms
$t_{d(IDLE-OSC)}$	Delay time, Idle instruction executed to oscillator power off				$4t_{c(CO)}$		µs
$t_{d(EX)}$	Delay time, reset vector executed after $\overline{RS}$ high			36H			ns

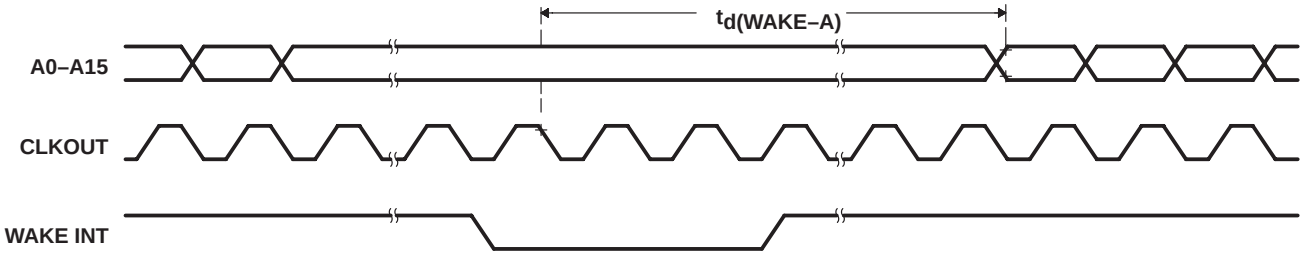


Figure 22. Entry and Exit Timing – LPM0 and LPM1

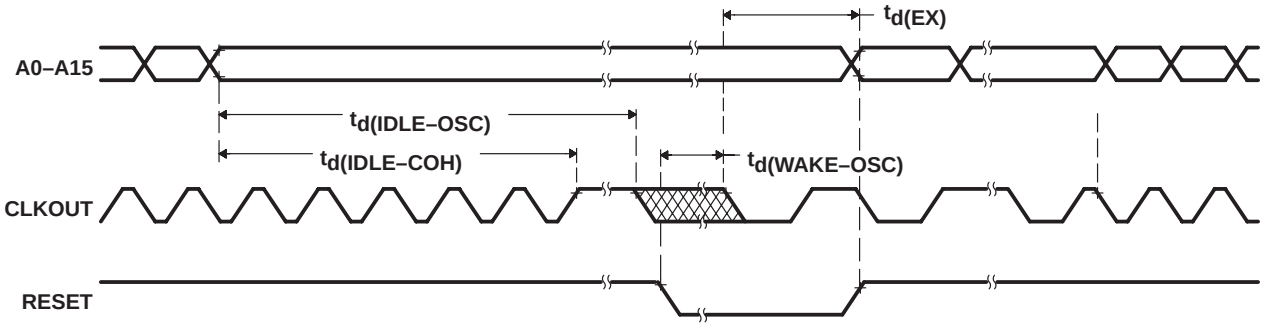


Figure 23. HALT Mode – LPM2

NOTE: WAKE INT can be any valid interrupt or RESET

$\overline{RS}$ timings

switching characteristics over recommended operating conditions for a reset [H = 0.5$t_{c(CO)}$] (see Figure 24)

PARAMETER		MIN	MAX	UNIT
$t_{w(RSL1)}$	Pulse duration, $\overline{RS}$ low†	$8t_{c(CO)}$		ns
$t_{d(EX)}$	Delay time, reset vector executed after $\overline{RS}$ high	36H		ns

† The parameter $t_{w(RSL1)}$ refers to the time $\overline{RS}$ is an output.

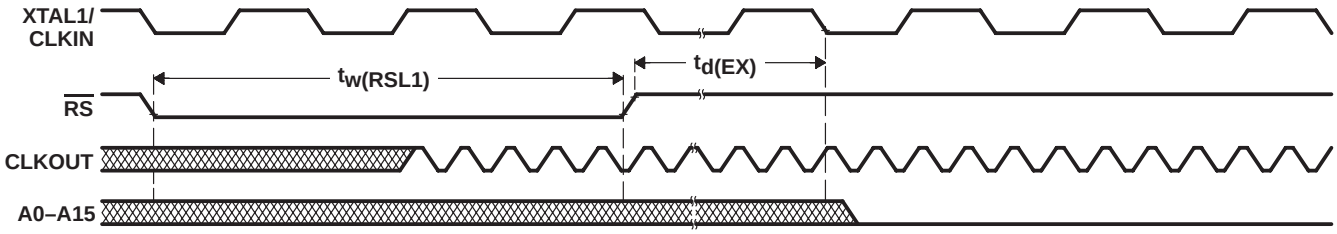
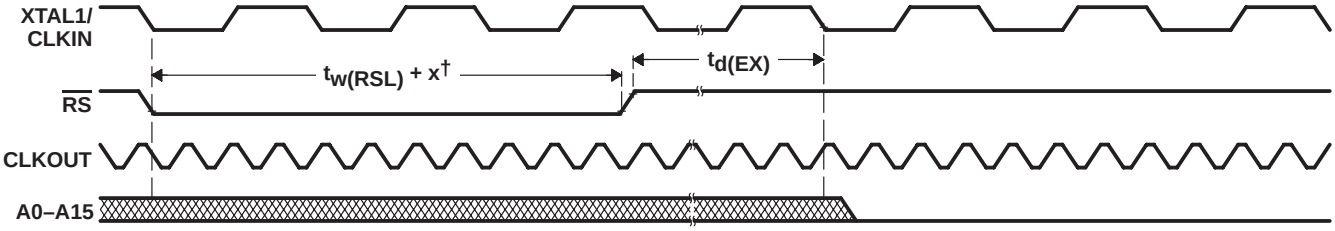


Figure 24. Watchdog Reset Pulse

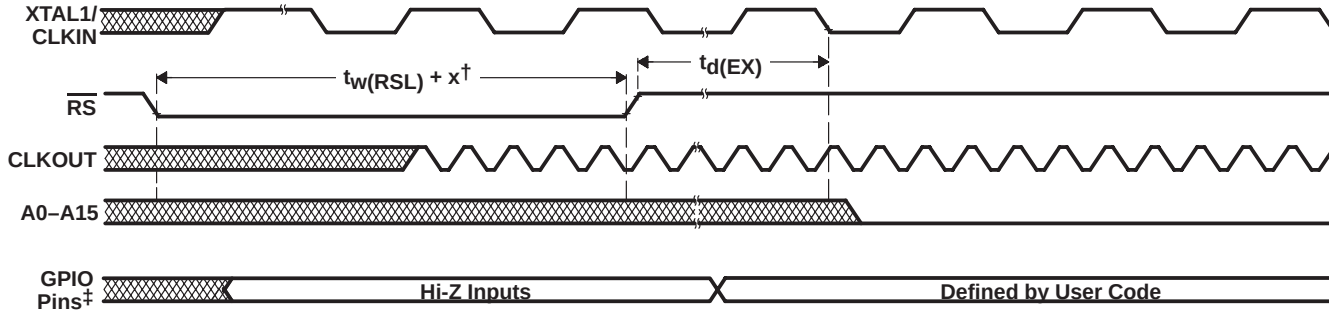
timing requirements for a reset [H = 0.5$t_{c(CO)}$] (see Figure 25)

		MIN	MAX	UNIT
$t_{w(RSL)}$	Pulse duration, $\overline{RS}$ low†	8H		ns
$t_{d(EX)}$	Delay time, reset vector executed after $\overline{RS}$ high	36H		ns

† The parameter $t_{w(RSL)}$ refers to the time $\overline{RS}$ is an input



Case A. External reset after power-on



Case B. Power-on reset

† The value of x depends on the reset condition as follows: **PLL enabled:** Assuming CLKIN is stable, x=PLL lock-up time. If the internal oscillator is used, x=oscillator lock-up time + PLL lock-up time. In case of resets after power on reset, x=0 (i.e., $t_{w(RSL)}$=8H ns only).

‡ (All GPIO pins *except* CLKOUT and XF pins.) GPIO pins are undefined until XTAL1/CLKIN is valid. This behavior is important to consider while using PWM pins for power-electronic circuits.

Figure 25. Reset Timing

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

switching characteristics over recommended operating conditions (see Figure 26)

PARAMETER		MIN	MAX	UNIT
$t_{d(XF)}$	Delay time, CLKOUT high to XF high/low	–3	7	ns

timing requirements (see Figure 26)

		MIN	MAX	UNIT
$t_{su(BIO)CO}$	Setup time, $\overline{\text{BIO}}$ or MP/$\overline{\text{MC}}$ low before CLKOUT low	0		ns
$t_{h(BIO)CO}$	Hold time, $\overline{\text{BIO}}$ or MP/$\overline{\text{MC}}$ low after CLKOUT low	19		ns

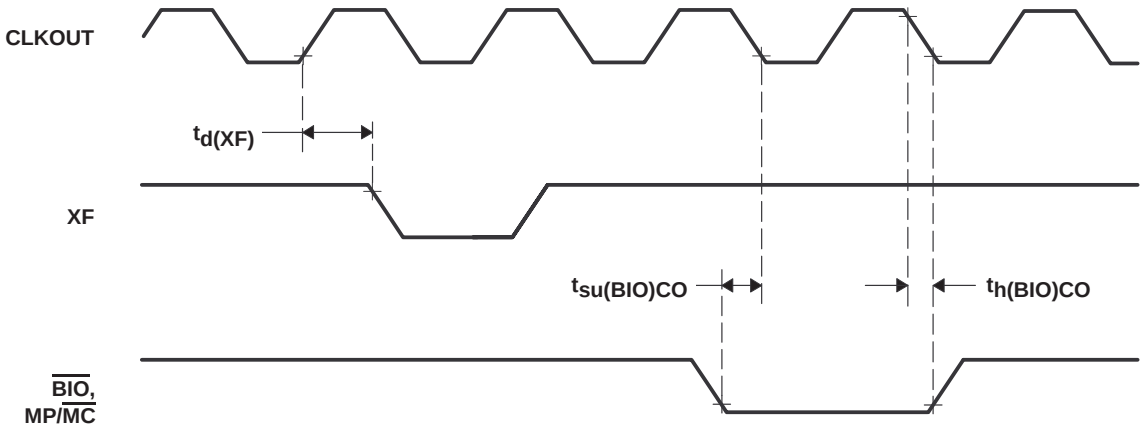


Figure 26. XF and $\overline{\text{BIO}}$ Timing

TIMING EVENT MANAGER INTERFACE

PWM timings

PWM refers to PWM outputs on PWM1, PWM2, PWM3, PWM4, PWM5, PWM6, T1PWM, and T2PWM.

switching characteristics over recommended operating conditions for PWM timing [H = 0.5$t_{c(CO)}$] (see Figure 27)

PARAMETER		MIN	MAX	UNIT
$t_{w(PWM)}$†	Pulse duration, PWM output high/low	2H+5		ns
$t_{d(PWM)CO}$	Delay time, CLKOUT low to PWM output switching		15	ns

† PWM outputs may be 100%, 0%, or increments of $t_{c(CO)}$ with respect to the PWM period.

timing requirements‡ [H = 0.5$t_{c(CO)}$] (see Figure 28)

		MIN	MAX	UNIT
$t_{w(TMRDIR)}$	Pulse duration, TMRDIR low/high	4H+5		ns
$t_{w(TMRCLK)}$	Pulse duration, TMRCLK low as a percentage of TMRCLK cycle time	40	60	%
$t_{wh(TMRCLK)}$	Pulse duration, TMRCLK high as a percentage of TMRCLK cycle time	40	60	%
$t_{c(TMRCLK)}$	Cycle time, TMRCLK	$4 \times t_{c(CO)}$		ns

‡ Parameter TMRDIR is equal to the pin TDIR, and parameter TMRCLK is equal to the pin TCLKIN.

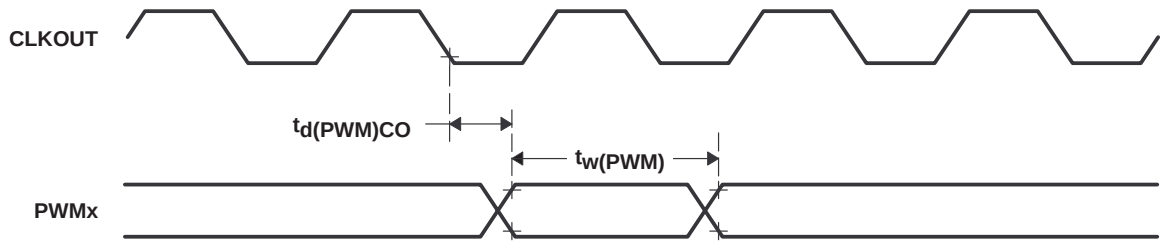


Figure 27. PWM Output Timing

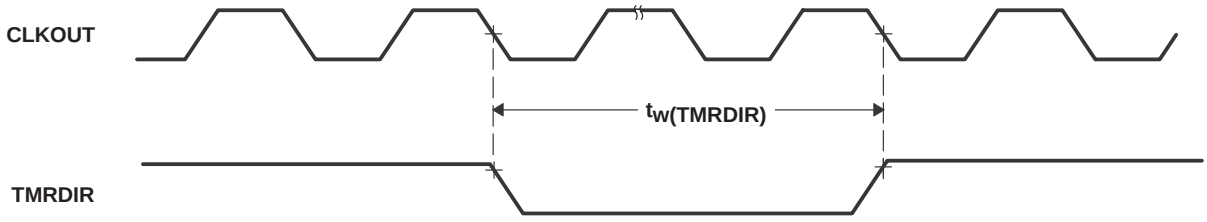


Figure 28. Capture/TMRDIR Timing

capture and QEP timings

CAP refers to CAP1/QEP0/IOPA3, CAP2/QEP1/IOPA4, and CAP3/IOPA5.

timing requirements [H = 0.5$t_{c(CO)}$] (see Figure 29)

		MIN	MAX	UNIT
$t_{w(CAP)}$	Pulse duration, CAP input low/high	4H +15		ns

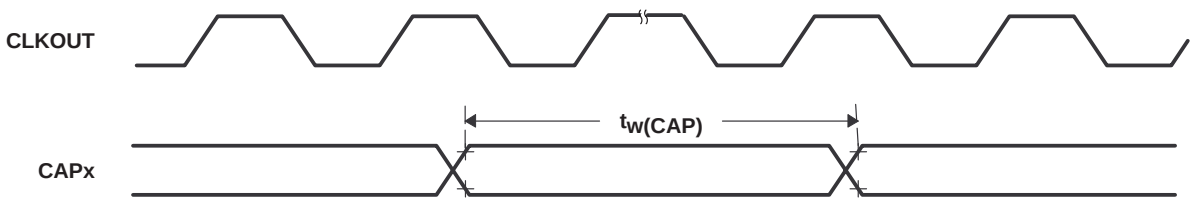


Figure 29. Capture Input and QEP Timing

interrupt timings

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

switching characteristics over recommended operating conditions (see Figure 30)

PARAMETER		MIN	MAX	UNIT
$t_{hz(PWM)PDP}$	Delay time, $\overline{\text{PDPINT}}$ low to PWM to high-impedance state		12	ns
$t_{d(INT)}$	Delay time, INT low/high to interrupt-vector fetch	$10t_{c(CO)}$		ns

timing requirements [H = $0.5t_{c(CO)}$] (see Figure 30)

		MIN	MAX	UNIT
$t_{w(INT)}$	Pulse duration, INT input low/high	2H+15		ns
$t_{w(PDP)}$	Pulse duration, $\overline{\text{PDPINT}}$ input low	4H+5		ns

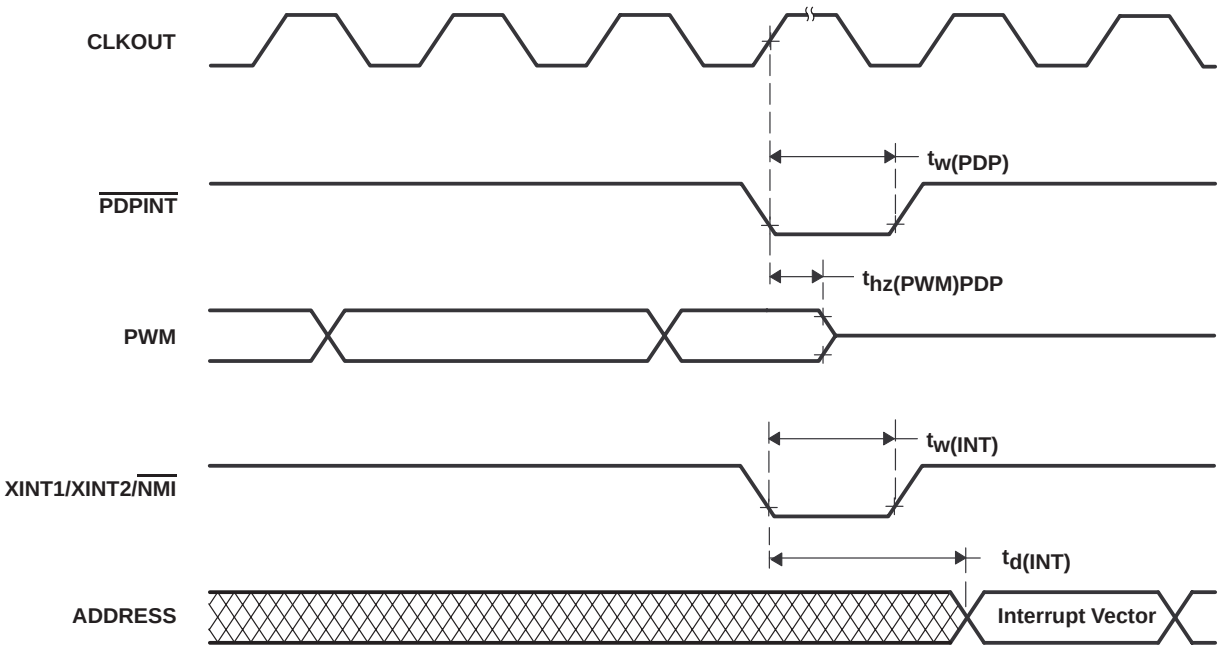


Figure 30. $\overline{\text{PDPINT}}$ and External Interrupt Timings

general-purpose input/output timings

switching characteristics over recommended operating conditions (see Figure 31)

PARAMETER			MIN	MAX	UNIT
$t_{d(GPO)CO}$	Delay time, CLKOUT low to GPIO low/high	All GPIOs		9	ns
$t_{r(GPO)}$	Rise time, GPIO switching low to high	All GPIOs		8	ns
$t_{f(GPO)}$	Fall time, GPIO switching high to low	All GPIOs		6	ns

timing requirements [H = 0.5$t_{c(CO)}$] (see Figure 32)

		MIN	MAX	UNIT
$t_{w(GPI)}$	Pulse duration, GPI high/low	2H+15		ns

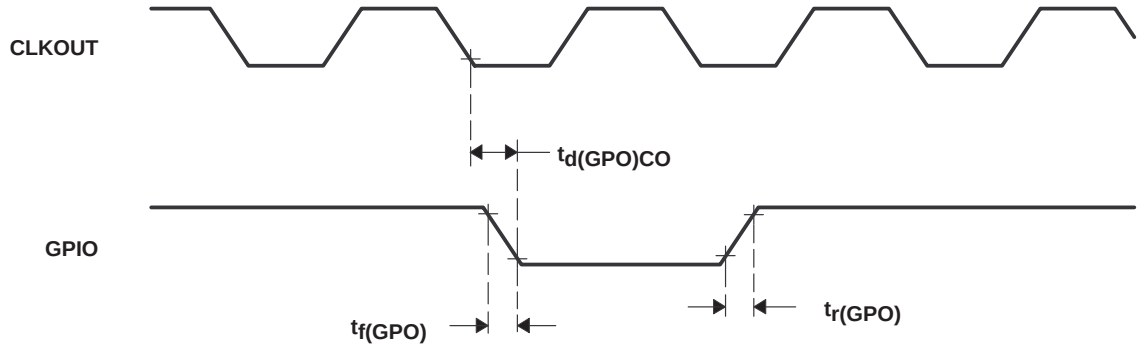


Figure 31. General-Purpose Output Timing

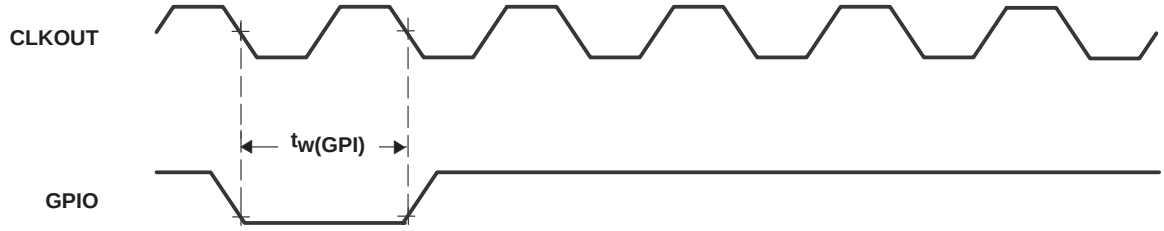


Figure 32. General-Purpose Input Timing

SPI MASTER MODE TIMING PARAMETERS

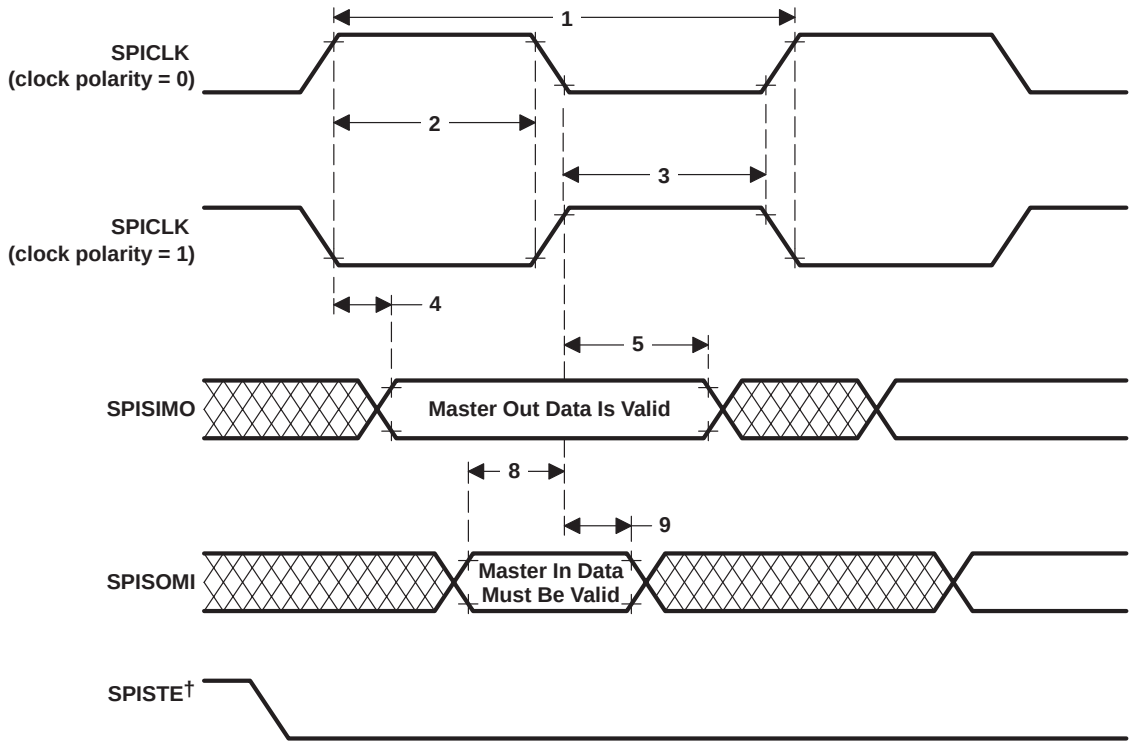
SPI master mode timing information is listed in the following tables.

SPI master mode external timing parameters (clock phase = 0)†‡ (see Figure 33)

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	$4t_{c(CO)}$	$128t_{c(CO)}$	$5t_{c(CO)}$	$127t_{c(CO)}$	ns
2§	$t_{w(SPCH)M}$	Pulse duration, SPICLK high (clock polarity = 0)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}-0.5t_{c(CO)}-10$	$0.5t_{c(SPC)M}-0.5t_{c(CO)}$	ns
	$t_{w(SPCL)M}$	Pulse duration, SPICLK low (clock polarity = 1)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}-0.5t_{c(CO)}-10$	$0.5t_{c(SPC)M}-0.5t_{c(CO)}$	
3§	$t_{w(SPCL)M}$	Pulse duration, SPICLK low (clock polarity = 0)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}+0.5t_{c(CO)}-10$	$0.5t_{c(SPC)M}+0.5t_{c(CO)}$	ns
	$t_{w(SPCH)M}$	Pulse duration, SPICLK high (clock polarity = 1)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}+0.5t_{c(CO)}-10$	$0.5t_{c(SPC)M}+0.5t_{c(CO)}$	
4§	$t_{d(SPCH-SIMO)M}$	Delay time, SPICLK high to SPISIMO valid (clock polarity = 0)	– 10	10	– 10	10	ns
	$t_{d(SPCL-SIMO)M}$	Delay time, SPICLK low to SPISIMO valid (clock polarity = 1)	– 10	10	– 10	10	
5§	$t_{v(SPCL-SIMO)M}$	Valid time, SPISIMO data valid after SPICLK low (clock polarity =0)	$0.5t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}+0.5t_{c(CO)}-10$		ns
	$t_{v(SPCH-SIMO)M}$	Valid time, SPISIMO data valid after SPICLK high (clock polarity =1)	$0.5t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}+0.5t_{c(CO)}-10$		
8§	$t_{su(SOMI-SPCL)M}$	Setup time, SPISOMI before SPICLK low (clock polarity = 0)	0		0		ns
	$t_{su(SOMI-SPCH)M}$	Setup time, SPISOMI before SPICLK high (clock polarity = 1)	0		0		
9§	$t_{v(SPCL-SOMI)M}$	Valid time, SPISOMI data valid after SPICLK low (clock polarity = 0)	$0.25t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-0.5t_{c(CO)}-10$		ns
	$t_{v(SPCH-SOMI)M}$	Valid time, SPISOMI data valid after SPICLK high (clock polarity = 1)	$0.25t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-0.5t_{c(CO)}-10$		

† The MASTER / SLAVE bit (SPICTL.2) is set and the CLOCK PHASE bit (SPICTL.3) is cleared.
‡ t_c = system clock cycle time = 1/CLKOUT = $t_{c(CO)}$
§ 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.

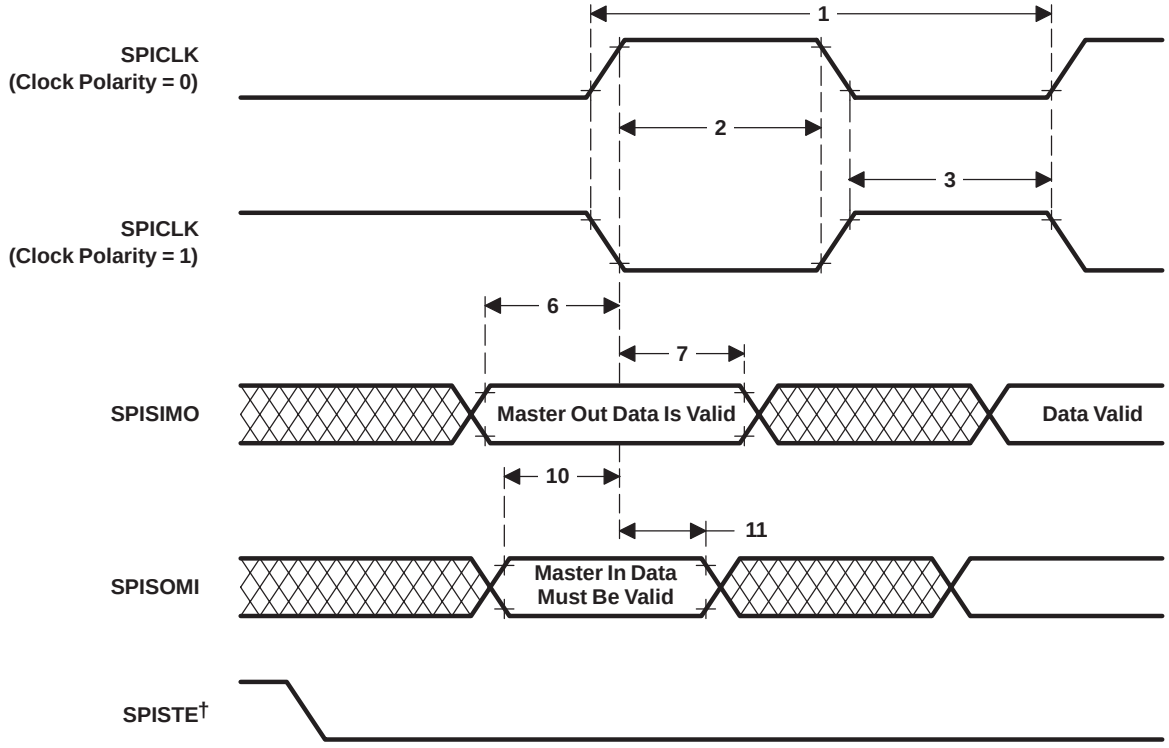
Figure 33. SPI Master Mode External Timing (Clock Phase = 0)

SPI master mode external timing parameters (clock phase = 1)†‡ (see Figure 34)

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	$4t_{c(CO)}$	$128t_{c(CO)}$	$5t_{c(CO)}$	$127t_{c(CO)}$	ns
2§	$t_{w(SPCH)M}$	Pulse duration, SPICLK high (clock polarity = 0)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}-0.5t_{c(CO)}-10$	$0.5t_{c(SPC)M}-0.5t_{c(CO)}$	ns
	$t_{w(SPCL)M}$	Pulse duration, SPICLK low (clock polarity = 1)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}-0.5t_{c(CO)}-10$	$0.5t_{c(SPC)M}-0.5t_{c(CO)}$	
3§	$t_{w(SPCL)M}$	Pulse duration, SPICLK low (clock polarity = 0)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}+0.5t_{c(CO)}-10$	$0.5t_{c(SPC)M}+0.5t_{c(CO)}$	ns
	$t_{w(SPCH)M}$	Pulse duration, SPICLK high (clock polarity = 1)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}+0.5t_{c(CO)}-10$	$0.5t_{c(SPC)M}+0.5t_{c(CO)}$	
6§	$t_{su(SIMO-SPCH)M}$	Setup time, SPISIMO data valid before SPICLK high (clock polarity = 0)	$0.5t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-10$		ns
	$t_{su(SIMO-SPCL)M}$	Setup time, SPISIMO data valid before SPICLK low (clock polarity = 1)	$0.5t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-10$		
7§	$t_{v(SPCH-SIMO)M}$	Valid time, SPISIMO data valid after SPICLK high (clock polarity =0)	$0.5t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-10$		ns
	$t_{v(SPCL-SIMO)M}$	Valid time, SPISIMO data valid after SPICLK low (clock polarity =1)	$0.5t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-10$		
10§	$t_{su(SOMI-SPCH)M}$	Setup time, SPISOMI before SPICLK high (clock polarity = 0)	0		0		ns
	$t_{su(SOMI-SPCL)M}$	Setup time, SPISOMI before SPICLK low (clock polarity = 1)	0		0		
11§	$t_{v(SPCH-SOMI)M}$	Valid time, SPISOMI data valid after SPICLK high (clock polarity = 0)	$0.25t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-10$		ns
	$t_{v(SPCL-SOMI)M}$	Valid time, SPISOMI data valid after SPICLK low (clock polarity = 1)	$0.25t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-10$		

† The MASTER / SLAVE bit (SPICTL.2) is set and the CLOCK PHASE bit (SPICTL.3) is set.
‡ t_c = system clock cycle time = 1/CLKOUT = $t_{c(CO)}$
§ 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 34. SPI Master Mode External Timing (Clock Phase = 1)

SPI SLAVE MODE TIMING PARAMETERS

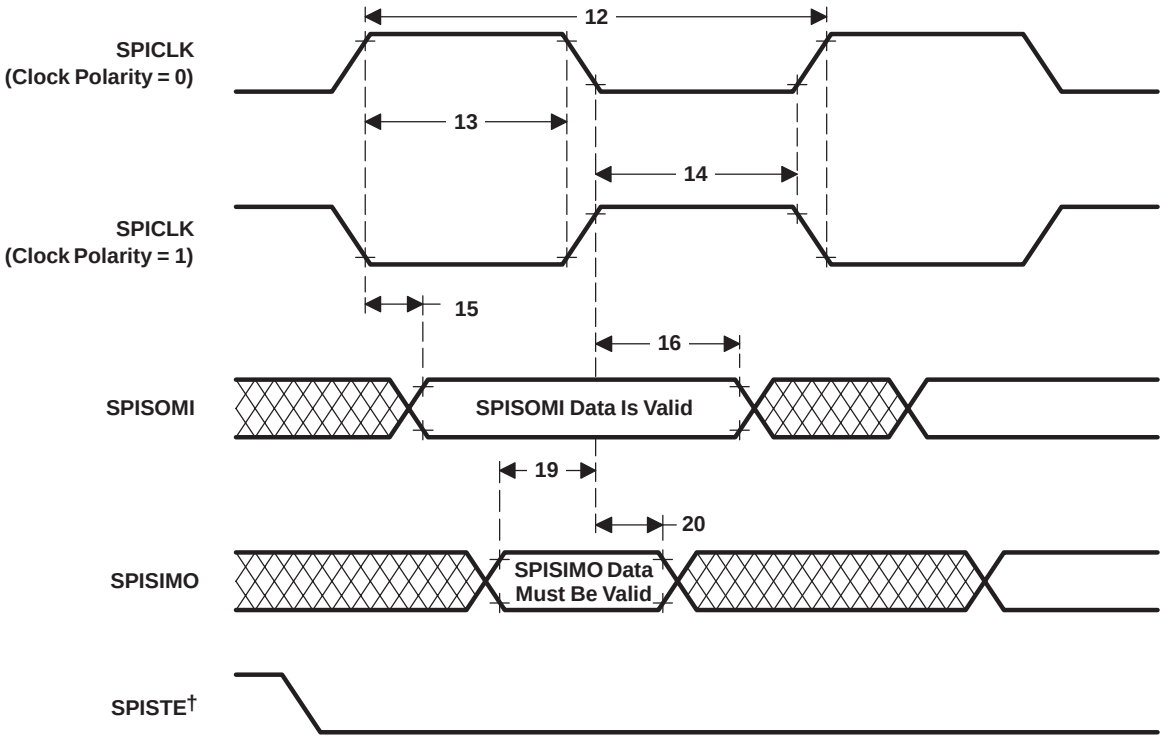
Slave mode timing information is listed in the following tables.

SPI slave mode external timing parameters (clock phase = 0)†‡ (see Figure 35)

NO.			MIN	MAX	UNIT
12	$t_{c(SPC)S}$	Cycle time, SPICLK	$4t_{c(CO)}$‡		ns
13§	$t_{w(SPCH)S}$	Pulse duration, SPICLK high (clock polarity = 0)	$0.5t_{c(SPC)S}-10$	$0.5t_{c(SPC)S}$	ns
	$t_{w(SPCL)S}$	Pulse duration, SPICLK low (clock polarity = 1)	$0.5t_{c(SPC)S}-10$	$0.5t_{c(SPC)S}$	
14§	$t_{w(SPCL)S}$	Pulse duration, SPICLK low (clock polarity = 0)	$0.5t_{c(SPC)S}-10$	$0.5t_{c(SPC)S}$	ns
	$t_{w(SPCH)S}$	Pulse duration, SPICLK high (clock polarity = 1)	$0.5t_{c(SPC)S}-10$	$0.5t_{c(SPC)S}$	
15§	$t_{d(SPCH-SOMI)S}$	Delay time, SPICLK high to SPISOMI valid (clock polarity = 0)	$0.375t_{c(SPC)S}-10$		ns
	$t_{d(SPCL-SOMI)S}$	Delay time, SPICLK low to SPISOMI valid (clock polarity = 1)	$0.375t_{c(SPC)S}-10$		
16§	$t_{v(SPCL-SOMI)S}$	Valid time, SPISOMI data valid after SPICLK low (clock polarity =0)	$0.75t_{c(SPC)S}$		ns
	$t_{v(SPCH-SOMI)S}$	Valid time, SPISOMI data valid after SPICLK high (clock polarity =1)	$0.75t_{c(SPC)S}$		
19§	$t_{su(SIMO-SPCL)S}$	Setup time, SPISIMO before SPICLK low (clock polarity = 0)	0		ns
	$t_{su(SIMO-SPCH)S}$	Setup time, SPISIMO before SPICLK high (clock polarity = 1)	0		
20§	$t_{v(SPCL-SIMO)S}$	Valid time, SPISIMO data valid after SPICLK low (clock polarity = 0)	$0.5t_{c(SPC)S}$		ns
	$t_{v(SPCH-SIMO)S}$	Valid time, SPISIMO data valid after SPICLK high (clock polarity = 1)	$0.5t_{c(SPC)S}$		

† The MASTER / SLAVE bit (SPICTL.2) is cleared and the CLOCK PHASE bit (SPICTL.3) is cleared.
‡ t_c = system clock cycle time = 1/CLKOUT = $t_{c(CO)}$
§ 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.

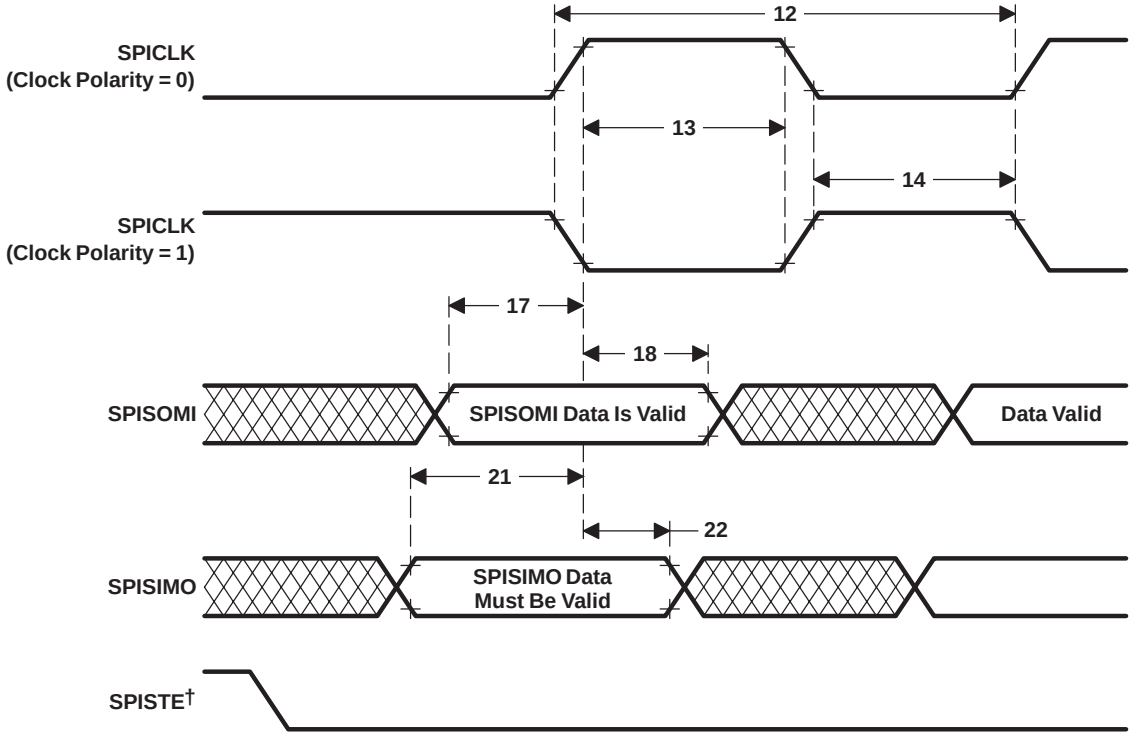
Figure 35. SPI Slave Mode External Timing (Clock Phase = 0)

SPI slave mode external timing parameters (clock phase = 1)†‡ (see Figure 36)

NO.			MIN	MAX	UNIT
12	$t_{c(SPC)S}$	Cycle time, SPICLK	$8t_{c(CO)}$		ns
13§	$t_{w(SPCH)S}$	Pulse duration, SPICLK high (clock polarity = 0)	$0.5t_{c(SPC)S}-10$	$0.5t_{c(SPC)S}$	ns
	$t_{w(SPCL)S}$	Pulse duration, SPICLK low (clock polarity = 1)	$0.5t_{c(SPC)S}-10$	$0.5t_{c(SPC)S}$	
14§	$t_{w(SPCL)S}$	Pulse duration, SPICLK low (clock polarity = 0)	$0.5t_{c(SPC)S}-10$	$0.5t_{c(SPC)S}$	ns
	$t_{w(SPCH)S}$	Pulse duration, SPICLK high (clock polarity = 1)	$0.5t_{c(SPC)S}-10$	$0.5t_{c(SPC)S}$	
17§	$t_{su(SOMI-SPCH)S}$	Setup time, SPISOMI before SPICLK high (clock polarity = 0)	$0.125t_{c(SPC)S}$		ns
	$t_{su(SOMI-SPCL)S}$	Setup time, SPISOMI before SPICLK low (clock polarity = 1)	$0.125t_{c(SPC)S}$		
18§	$t_{v(SPCH-SOMI)S}$	Valid time, SPISOMI data valid after SPICLK high (clock polarity =0)	$0.75t_{c(SPC)S}$		ns
	$t_{v(SPCL-SOMI)S}$	Valid time, SPISOMI data valid after SPICLK low (clock polarity =1)	$0.75t_{c(SPC)S}$		
21§	$t_{su(SIMO-SPCH)S}$	Setup time, SPISIMO before SPICLK high (clock polarity = 0)	0		ns
	$t_{su(SIMO-SPCL)S}$	Setup time, SPISIMO before SPICLK low (clock polarity = 1)	0		
22§	$t_{v(SPCH-SIMO)S}$	Valid time, SPISIMO data valid after SPICLK high (clock polarity = 0)	$0.5t_{c(SPC)S}$		ns
	$t_{v(SPCL-SIMO)S}$	Valid time, SPISIMO data valid after SPICLK low (clock polarity = 1)	$0.5t_{c(SPC)S}$		

† The MASTER / SLAVE bit (SPICTL.2) is cleared and the CLOCK PHASE bit (SPICTL.3) is set.
‡ t_c = system clock cycle time = 1/CLKOUT = $t_{c(CO)}$
§ 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 36. SPI Slave Mode External Timing (Clock Phase = 1)

external memory interface *read* timings

switching characteristics over recommended operating conditions for an external memory interface read (see Figure 37)

PARAMETER		MIN	MAX	UNIT
$t_{d(COL–CNTL)}$	Delay time, CLKOUT low to control valid		3	ns
$t_{d(COL–CNTH)}$	Delay time, CLKOUT low to control inactive		3	ns
$t_{d(COL–A)RD}$	Delay time, CLKOUT low to address valid		5	ns
$t_{d(COH–RDL)}$	Delay time, CLKOUT high to $\overline{RD}$ strobe active		4	ns
$t_{d(COL–RDH)}$	Delay time, CLKOUT low to $\overline{RD}$ strobe inactive high	–4	0	ns
$t_{d(COL–SL)}$	Delay time, CLKOUT low to $\overline{STRB}$ strobe active low		3	ns
$t_{d(COL–SH)}$	Delay time, CLKOUT low to $\overline{STRB}$ strobe inactive high		3	ns
$t_{h(A)COL}$	Hold time, address valid after CLKOUT low	–4		ns
$t_{su(A)RD}$	Setup time, address valid before $\overline{RD}$ strobe active low	22		ns
$t_{h(A)RD}$	Hold time, address valid after $\overline{RD}$ strobe inactive high	–1		ns

timing requirements [H = $0.5t_{c(CO)}$] (see Figure 37)

		MIN	MAX	UNIT
$t_{a(A)}$	Access time, read data from address valid		2H–20	ns
$t_{su(D)RD}$	Setup time, read data before $\overline{RD}$ strobe inactive high	12		ns
$t_{h(D)RD}$	Hold time, read data after $\overline{RD}$ strobe inactive high	0		ns
$t_{h(AIV-D)}$	Hold time, read data after address invalid	–3		ns

external memory interface *read* timings (continued)

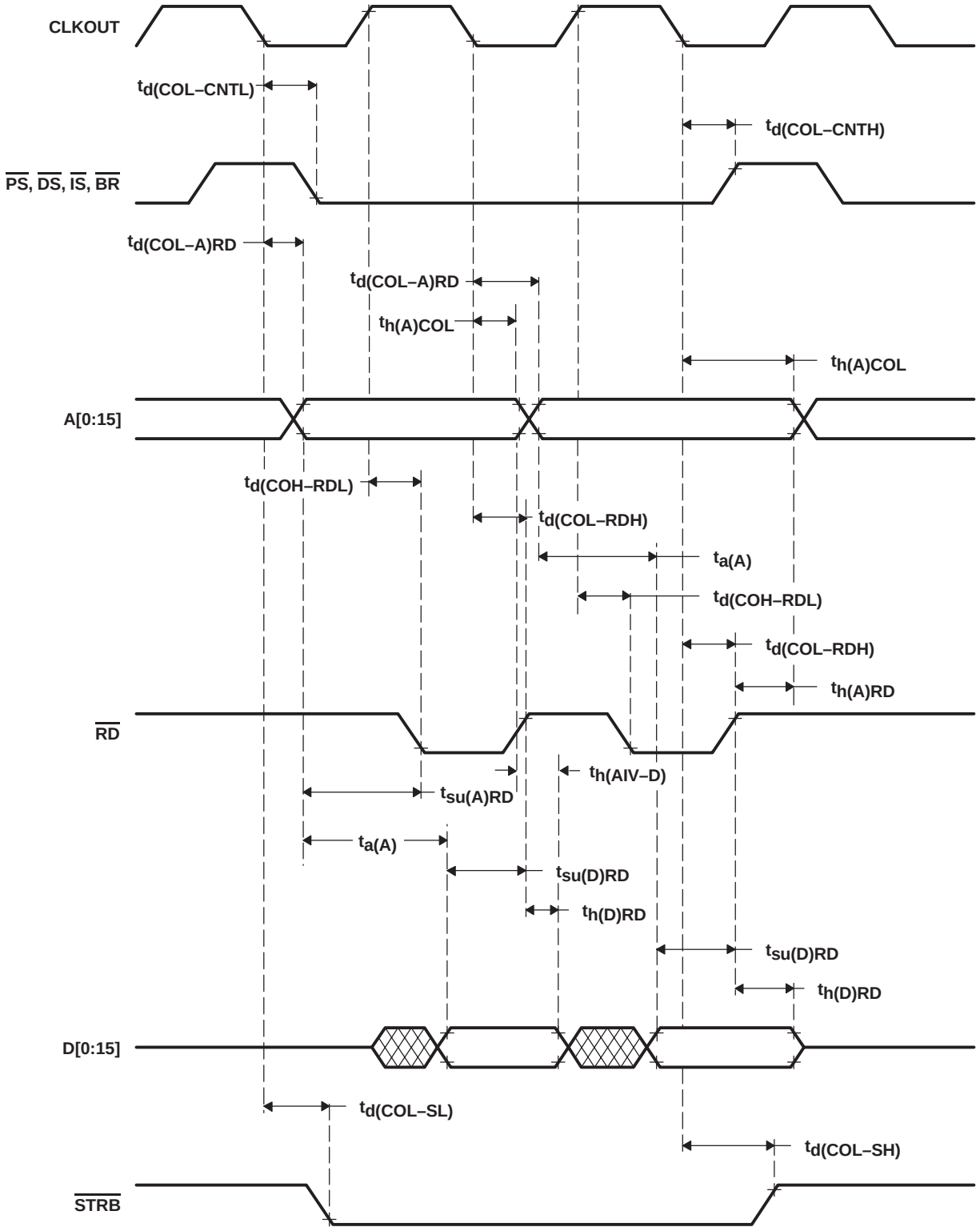


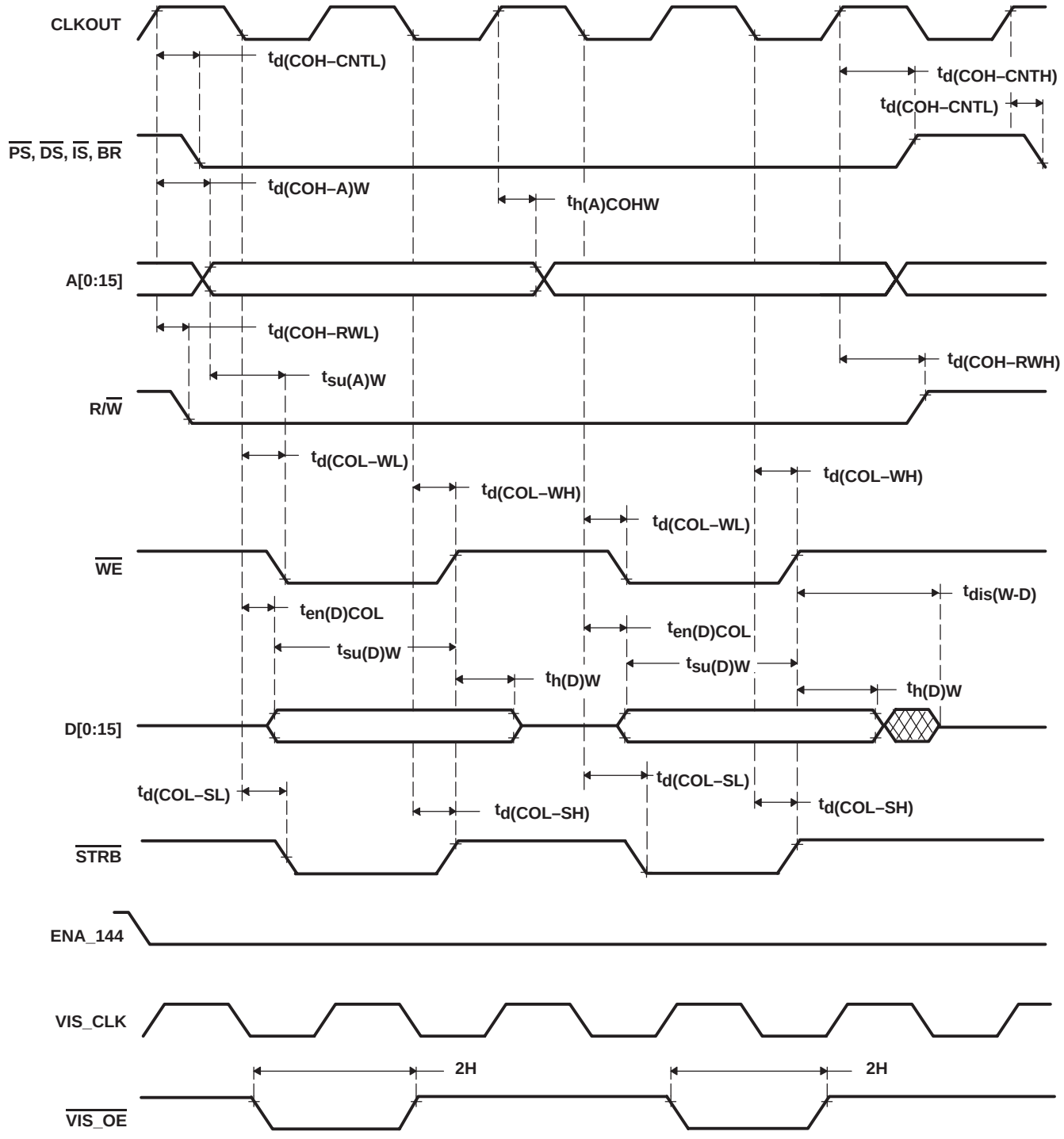
Figure 37. Memory Interface Read/Read Timings

external memory interface *write* timings

switching characteristics over recommended operating conditions for an external memory interface write [H = $0.5t_{c(CO)}$] (see Figure 38)

PARAMETER		MIN	MAX	UNIT
$t_{d(COH–CNTL)}$	Delay time, CLKOUT high to control valid		9	ns
$t_{d(COH–CNTH)}$	Delay time, CLKOUT high to control inactive		9	ns
$t_{d(COH–A)W}$	Delay time, CLKOUT high to address valid		11	ns
$t_{d(COH–RWL)}$	Delay time, CLKOUT high to R/$\overline{W}$ low		6	ns
$t_{d(COH–RWH)}$	Delay time, CLKOUT high to R/$\overline{W}$ high		6	ns
$t_{d(COL–WL)}$	Delay time, CLKOUT low to $\overline{WE}$ strobe active low	–4	0	ns
$t_{d(COL–WH)}$	Delay time, CLKOUT low to $\overline{WE}$ strobe inactive high	–4	0	ns
$t_{en(D)COL}$	Enable time, data bus driven from CLKOUT low	7		ns
$t_{d(COL–SL)}$	Delay time, CLKOUT low to $\overline{STRB}$ active low		3	ns
$t_{d(COL–SH)}$	Delay time, CLKOUT low to $\overline{STRB}$ inactive high		3	ns
$t_{h(A)COHW}$	Hold time, address valid after CLKOUT high	H–1		ns
$t_{su(A)W}$	Setup time, address valid before $\overline{WE}$ strobe active low	H–9		ns
$t_{su(D)W}$	Setup time, write data before $\overline{WE}$ strobe inactive high	2H–1		ns
$t_{h(D)W}$	Hold time, write data after $\overline{WE}$ strobe inactive high	3		ns
$t_{dis(W-D)}$	Disable time, data bus high impedance from $\overline{WE}$ high	4		ns

external memory interface *write* timings (continued)



NOTE A: ENA_144 when low along with BVIS bits (10,9 set to 10 or 11) in register WSGR - IO@FFFFh, VIS_CLK and $\overline{\text{VIS\_OE}}$ will be visible at pins 31 (F243) and 126 (F243) respectively. VIS_CLK and $\overline{\text{VIS\_OE}}$ indicate internal memory write cycles (program/data). During $\overline{\text{VIS\_OE}}$ cycles, the external bus will be driven. VIS_CLK is essentially CLKOUT, to be used along with $\overline{\text{VIS\_OE}}$ for trace capabilities.

Figure 38. Address Visibility Mode

external memory interface *ready-on-read* timings

switching characteristics over recommended operating conditions for an external memory interface ready-on-read (see Figure 39)

PARAMETER		MIN	MAX	UNIT
$t_{d(COL–A)RD}$	Delay time, CLKOUT low to address valid		5	ns

timing requirements for an external memory interface ready-on-read (see Figure 39)

		MIN	MAX	UNIT
$t_{h(RDY)COH}$	Hold time, READY after CLKOUT high	–5		ns
$t_{su(D)RD}$	Setup time, read data before $\overline{RD}$ strobe inactive high	12		ns
$t_{v(RDY)ARD}$	Valid time, READY after address valid on read		4	ns
$t_{su(RDY)COH}$	Setup time, READY before CLKOUT high	17		ns

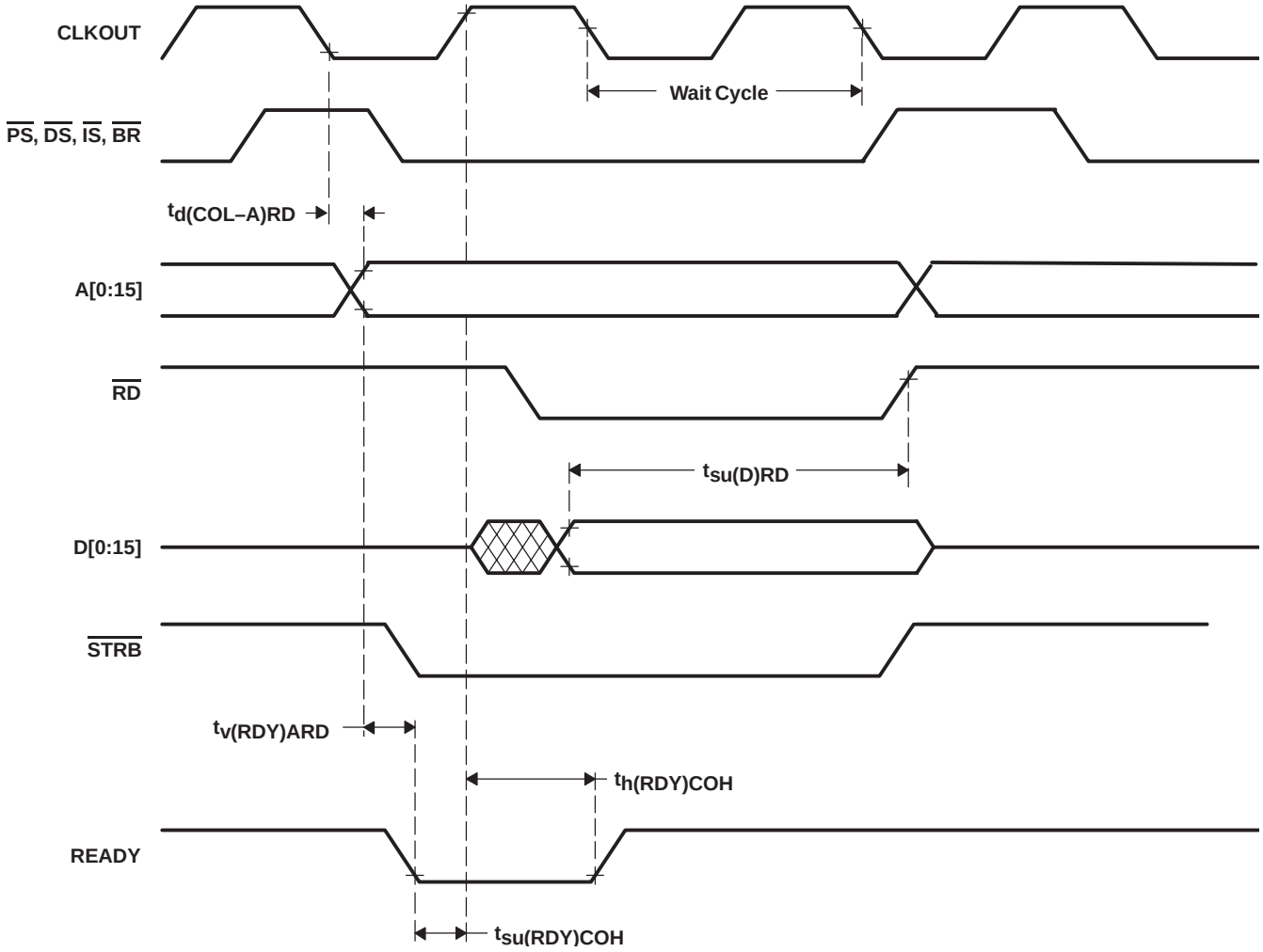


Figure 39. Ready-on-Read Timings

external memory interface *ready-on-write* timings

switching characteristics over recommended operating conditions for an external memory interface ready-on-write (see Figure 40)

PARAMETER		MIN	MAX	UNIT
$t_{d(COH–A)W}$	Delay time, CLKOUT high to address valid		11	ns

timing requirements for an external memory interface ready-on-write [H = 0.5$t_{c(CO)}$] (see Figure 40)

		MIN	MAX	UNIT
$t_{h(RDY)COH}$	Hold time, READY after CLKOUT high	–5		ns
$t_{su(D)W}$	Setup time, write data before $\overline{WE}$ strobe inactive high	2H–1	2H	ns
$t_{v(RDY)AW}$	Valid time, READY after address valid on write		4	ns
$t_{su(RDY)COH}$	Setup time, READY before CLKOUT high	17		ns

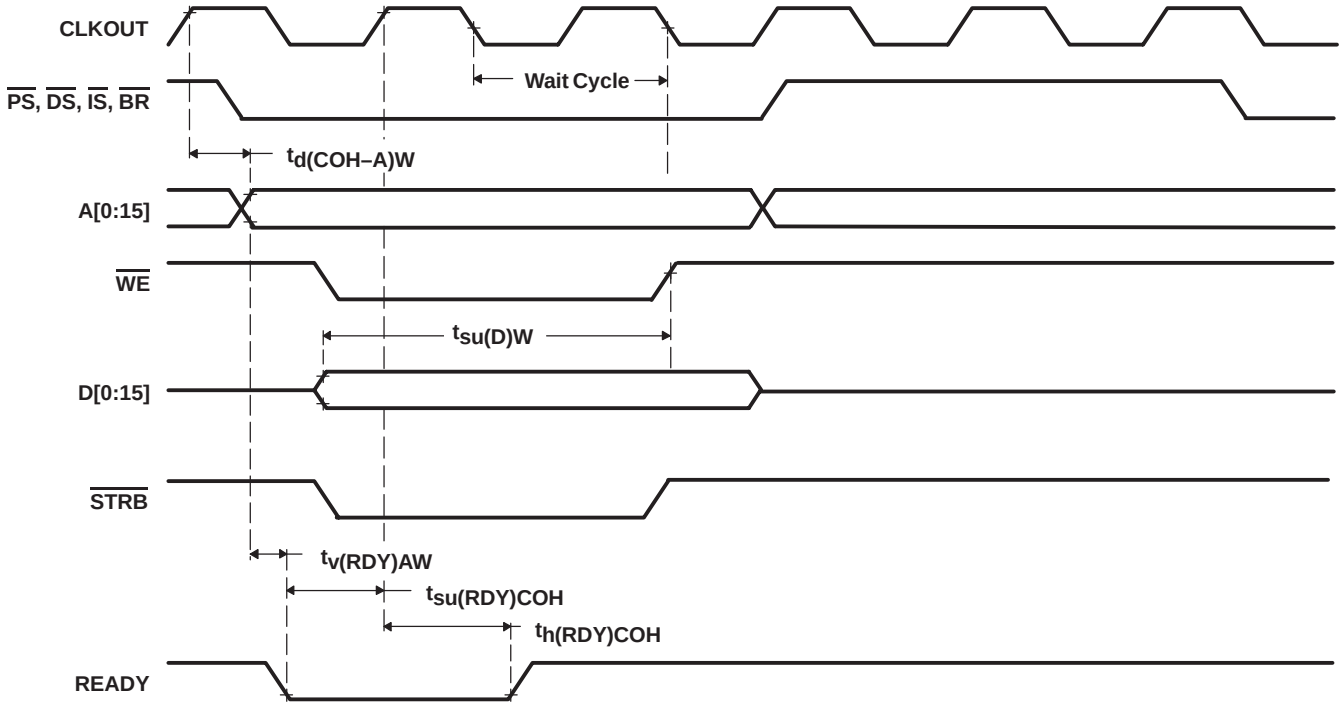


Figure 40. Ready-on-Write Timings

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 power bus isolation 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 $V_I \le V_{SSA}$; 3FFh for $V_I \ge V_{CCA}$)
Conversion time (including sample time) . . . . . 1 μs

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}		V_{CCA}	V
V_{REFLO}	Analog ground reference source†	V_{SSA}		V_{REFHI}	V
V_{AI}	Analog input voltage, ADCIN00–ADCIN07	V_{SSA}		V_{CCA}	V

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

ADC operating frequency

	MIN	MAX	UNIT
ADC operating frequency		20	MHz

operating characteristics over recommended operating condition ranges†

PARAMETER		DESCRIPTION		MIN	MAX	UNIT
I_{CCA}	Analog supply current	V_{CCA} = 5.5 V	Converting		10	mA
			Non-converting		2	
		V_{CCA} = V_{REFHI} = 5.5 V	PLL or OSC power down		1	µA
C_{ai}	Analog input capacitance	Typical capacitive load on analog input pin	Non-sampling		10	pF
			Sampling		30	
E_{DNL}	Differential nonlinearity error	Difference between the actual step width and the ideal value			±2	LSB‡
E_{INL}	Integral nonlinearity error	Maximum deviation from the best straight line through the ADC transfer characteristics, excluding the quantization error			±2	LSB‡
$t_{d(PU)}$	Delay time, power-up to ADC valid	Time to stabilize analog stage after power-up			10	µs
Z_{AI}	Analog input source impedance	Analog input source impedance for conversions to remain within specifications			10	Ω

† Absolute resolution = 4.89 mV. At V_{REFHI} = 5 V and V_{REFLO} = 0 V, this is one LSB. As V_{REFHI} decreases, V_{REFLO} increases, or both, the LSB size decreases. Therefore, the absolute accuracy and differential/integral linearity errors in terms of LSBs increase.
‡ LSB denotes “Least Significant Bits”. For example, an error of 2 LSB corresponds to (2 * 4.89) = 9.78 mV.

ADC input pin circuit

One of the most common A/D application errors is inappropriate source impedance. In practice, minimum source impedance should be used to limit the error as well as to minimize the required sampling time; however, the source impedance must be smaller than Z_{AI}. A typical ADC input pin circuit is shown in Figure 41.

Figure 41. Typical ADC Input Pin Circuit

internal ADC module timings (see Figure 42)

		MIN	MAX	UNIT
$t_{c(AD)}$	Cycle time, ADC prescaled clock	50		ns
$t_{w(SHC)}$	Pulse duration, total sample/hold and conversion time†	900		ns
$t_{w(SH)}$	Pulse duration, sample and hold time	$3t_{c(AD)}$		ns
$t_{w(C)}$	Pulse duration, total conversion time	$10t_{c(AD)}$		ns
$t_{d(SOC-SH)}$	Delay time, start of conversion‡ to beginning of sample and hold	$3t_{c(CO)}$		ns
$t_{d(EOC-FIFO)}$	Delay time, end of conversion to data loaded into result FIFO	$2t_{c(CO)}$		ns
$t_{d(ADCINT)}$	Delay time, ADC flag to ADC interrupt	$2t_{c(CO)}$		ns

† The total sample/hold and conversion time is determined by the summation of $t_{d(SOC-SH)}$, $t_{w(SH)}$, $t_{w(C)}$, and $t_{d(EOC-FIFO)}$.
‡ Start of conversion is signaled by the ADCIMSTART bit (ADCTRL1.13) or the ADCSOC bit (ADCTRL1.0) set in software, the external start signal active (ADCSOC), or internal EVSOC signal active.

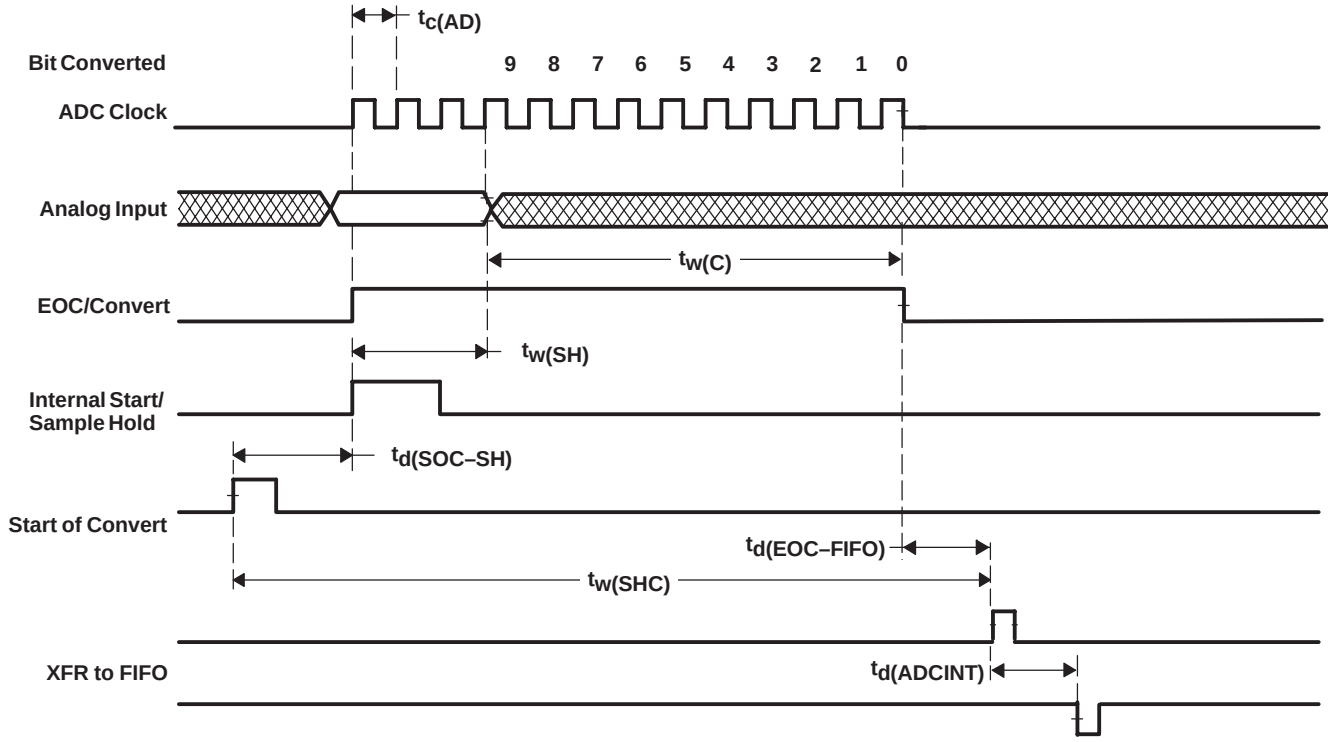


Figure 42. Analog-to-Digital Internal Module Timing

flash EEPROM

switching characteristics over recommended operating conditions

PARAMETER	F243/F241			UNIT
	MIN	TYP	MAX	
Program-erase endurance	10K			Cycles
Program pulses per word†	1	10	150	Pulses
Erase pulses per array†	1	20	1000	Pulses
Flash-write pulses per array†	1	20	6000	Pulses

† These parameters are used in the flash programming algorithms. For a detailed description of the algorithms, see the *TMS320F20x/TMS320F24x DSP Embedded Flash Memory Technical Reference* (literature number SPRU282).

timing requirements

		F243/F241		UNIT
		MIN	MAX	
$t_{d(BUSY)}$	Delay time, after mode deselect to stabilization†	10		µs
$t_{d(RD\text{-}VERIFY)}$	Delay time, verify read mode select to stabilization†	10		µs

† These parameters are used in the flash programming algorithms. For a detailed description of the algorithms, see the *TMS320F20x/TMS320F24x DSP Embedded Flash Memory Technical Reference* (literature number SPRU282).

programming operation

PARAMETER		F243/F241			UNIT
		MIN	NOM	MAX	
$t_{w(PGM)}$	Pulse duration, programming algorithm†	95	100	105	µs
$t_{d(PGM\text{-}MODE)}$	Delay time, program mode select to stabilization†	10			µs

† These parameters are used in the flash programming algorithms. For a detailed description of the algorithms, see the *TMS320F20x/TMS320F24x DSP Embedded Flash Memory Technical Reference* (literature number SPRU282).

erase operation

PARAMETER		F243/F241			UNIT
		MIN	NOM	MAX	
$t_{w(ERASE)}$	Pulse duration, erase algorithm†	6.65	7	7.35	ms
$t_{d(ERASE\text{-}MODE)}$	Delay time, erase mode select to stabilization†	10			µs

† These parameters are used in the flash programming algorithms. For a detailed description of the algorithms, see the *TMS320F20x/TMS320F24x DSP Embedded Flash Memory Technical Reference* (literature number SPRU282).

flash-write operation

PARAMETER		F243/F241			UNIT
		MIN	NOM	MAX	
$t_{w(FLW)}$	Pulse duration, flash-write algorithm†	13.3	14	14.7	ms
$t_{d(FLW\text{-}MODE)}$	Delay time, flash-write mode select to stabilization†	10			µs

† These parameters are used in the flash programming algorithms. For a detailed description of the algorithms, see the *TMS320F20x/TMS320F24x DSP Embedded Flash Memory Technical Reference* (literature number SPRU282).

register file compilation

Table 23 is a collection of all the programmable registers of the TMS320x24x (provided for a quick reference).

Table 23. 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 REGISTERS								
07010h	IRQ0.15	IRQ0.14	IRQ0.13	IRQ0.12	IRQ0.11	IRQ0.10	IRQ0.9	IRQ0.8	PIRQR0
	IRQ0.7	IRQ0.6	IRQ0.5	IRQ0.4	IRQ0.3	IRQ0.2	IRQ0.1	IRQ0.0	
07011h	IRQ1.15	IRQ1.14	IRQ1.13	IRQ1.12	IRQ1.11	IRQ1.10	IRQ1.9	IRQ1.8	PIRQR1
	IRQ1.7	IRQ1.6	IRQ1.5	IRQ1.4	IRQ1.3	IRQ1.2	IRQ1.1	IRQ1.0	
07012h to 07013h	Illegal								
07014h	IAK0.15	IAK0.14	IAK0.13	IAK0.12	IAK0.11	IAK0.10	IAK0.9	IAK0.8	PIACKR0
	IAK0.7	IAK0.6	IAK0.5	IAK0.4	IAK0.3	IAK0.2	IAK0.1	IAK0.0	
07015h	IAK1.15	IAK1.14	IAK1.13	IAK1.12	IAK1.11	IAK1.10	IAK1.9	IAK1.8	PIACKR1
	IAK1.7	IAK1.6	IAK1.5	IAK1.4	IAK1.3	IAK1.2	IAK1.1	IAK1.0	
07016h to 07017h	Illegal								
07018h	—	CLKSRC	LPM1	LPM0	—	—	—	—	SCSR
	—	—	—	—	—	—	—	ILLADR	
07019h to 0701Bh	Illegal								
0701Ch	DIN15	DIN14	DIN13	DIN12	DIN11	DIN10	DIN9	DIN8	DINR
	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0	
0701Dh	Illegal								
0701Eh	V15	V14	V13	V12	V11	V10	V9	V8	PIVR
	V7	V6	V5	V4	V3	V2	V1	V0	
0701Fh	Illegal								

register file compilation (continued)

Table 23. 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	
	WD CONTROL REGISTERS								
07020h to 07022h	Illegal								
07023h	D7	D6	D5	D4	D3	D2	D1	D0	WDCNTR
07024h	Illegal								
07025h	D7	D6	D5	D4	D3	D2	D1	D0	WDKEY
07026h to 07028h	Illegal								
07029h	WDFLAG	WDDIS	WDCHK2	WDCHK1	WDCHK0	WDPS2	WDPS1	WDPS0	WDCR
0702Ah to 07031h	Illegal								
	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	Illegal								
07034h	—	IM	EVSOCP	EXTSOCP	INTPRI	ADCEVSOC	ADCEXTSOC	—	ADCTRL2
	ADCFIFO2		—	ADCFIFO1		ADCPSCALE			
07035h	Illegal								
07036h	D9	D8	D7	D6	D5	D4	D3	D2	ADCFIFO1
	D1	D0	0	0	0	0	0	0	
07037h	Illegal								
07038h	D9	D8	D7	D6	D5	D4	D3	D2	ADCFIFO2
	D1	D0	0	0	0	0	0	0	
07039h to 0703Fh	Illegal								
	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	Illegal								
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	Illegal								
07046h	ERXB15	ERXB14	ERXB13	ERXB12	ERXB11	ERXB10	ERXB9	ERXB8	SPIRXEMU
	ERXB7	ERXB6	ERXB5	ERXB4	ERXB3	ERXB2	ERXB1	ERXB0	

register file compilation (continued)

Table 23. 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)								
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 0704Eh	Illegal								
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	Illegal								
07059h	TXDT7	TXDT6	TXDT5	TXDT4	TXDT3	TXDT2	TXDT1	TXDT0	SCITXBUF
0705Ah to 0705Eh	Illegal								
0705Fh	—	SCITX PRIORITY	SCIRX PRIORITY	SCI SOFT	SCI FREE	—	—	—	SCIPRI
07060h to 0706Fh	Illegal								
	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	Illegal								

register file compilation (continued)

Table 23. 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	
	DIGITAL I/O CONTROL REGISTERS								
07090h	CRA.15	CRA.14	CRA.13	CRA.12	CRA.11	CRA.10	CRA.9	CRA.8	OCRA
	CRA.7	CRA.6	CRA.5	CRA.4	CRA.3	CRA.2	CRA.1	CRA.0	
07091h	Illegal								
07092h	—	—	—	—	—	—	CRB.9	CRB.8	OCRB
	CRB.7	CRB.6	CRB.5	CRB.4	CRB.3	CRB.2	CRB.1	CRB.0	
07093h to 07097h	Illegal								
07098h	A7DIR	A6DIR	A5DIR	A4DIR	A3DIR	A2DIR	A1DIR	A0DIR	PADATDIR
	IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0	
07099h	Illegal								
0709Ah	B7DIR	B6DIR	B5DIR	B4DIR	B3DIR	B2DIR	B1DIR	B0DIR	PBDATDIR
	IOPB7	IOPB6	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0	
0709Bh	Illegal								
0709Ch	C7DIR	C6DIR	C5DIR	C4DIR	C3DIR	C2DIR	C1DIR	C0DIR	PCDATDIR
	IOPC7	IOPC6	IOPC5	IOPC4	IOPC3	IOPC2	IOPC1	IOPC0	
0709Dh	Illegal								
0709Eh	D7DIR	D6DIR	D5DIR	D4DIR	D3DIR	D2DIR	D1DIR	D0DIR	PDDATDIR
	IOPD7	IOPD6	IOPD5	IOPD4	IOPD3	IOPD2	IOPD1	IOPD0	
0709Fh	Illegal								
070A0h to 070FFh	Illegal								
	CONTROLLER AREA NETWORK (CAN) CONFIGURATION CONTROL REGISTERS								
07100h	—	—	—	—	—	—	—	—	MDER
	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	

register file compilation (continued)

Table 23. 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)								
07109h	—	—	MIF5	MIF4	MIF3	MIF2	MIF1	MIF0	CAN_IFR
	—	RMLIF	AAIF	WDIF	WUIF	BOIF	EPIF	WLIF	
0710Ah	MIL	—	MIM5	MIM4	MIM3	MIM2	MIM1	MIM0	CAN_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	
0710Fh to 071FFh	Illegal								
	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	D15	D14	D13	D12	D11	D10	D9	D8	MBX0A
	D7	D6	D5	D4	D3	D2	D1	D0	
07205h	D15	D14	D13	D12	D11	D10	D9	D8	MBX0B
	D7	D6	D5	D4	D3	D2	D1	D0	
07206h	D15	D14	D13	D12	D11	D10	D9	D8	MBX0C
	D7	D6	D5	D4	D3	D2	D1	D0	
07207h	D15	D14	D13	D12	D11	D10	D9	D8	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	D15	D14	D13	D12	D11	D10	D9	D8	MBX1A
	D7	D6	D5	D4	D3	D2	D1	D0	
0720Dh	D15	D14	D13	D12	D11	D10	D9	D8	MBX1B
	D7	D6	D5	D4	D3	D2	D1	D0	

register file compilation (continued)

Table 23. 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	D15	D14	D13	D12	D11	D10	D9	D8	MBX1C
	D7	D6	D5	D4	D3	D2	D1	D0	
0720Fh	D15	D14	D13	D12	D11	D10	D9	D8	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	D15	D14	D13	D12	D11	D10	D9	D8	MBX2A
	D7	D6	D5	D4	D3	D2	D1	D0	
07215h	D15	D14	D13	D12	D11	D10	D9	D8	MBX2B
	D7	D6	D5	D4	D3	D2	D1	D0	
07216h	D15	D14	D13	D12	D11	D10	D9	D8	MBX2C
	D7	D6	D5	D4	D3	D2	D1	D0	
07217h	D15	D14	D13	D12	D11	D10	D9	D8	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	D15	D14	D13	D12	D11	D10	D9	D8	MBX3A
	D7	D6	D5	D4	D3	D2	D1	D0	
0721Dh	D15	D14	D13	D12	D11	D10	D9	D8	MBX3B
	D7	D6	D5	D4	D3	D2	D1	D0	
0721Eh	D15	D14	D13	D12	D11	D10	D9	D8	MBX3C
	D7	D6	D5	D4	D3	D2	D1	D0	
0721Fh	D15	D14	D13	D12	D11	D10	D9	D8	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 23. 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	D15	D14	D13	D12	D11	D10	D9	D8	MBX4A
	D7	D6	D5	D4	D3	D2	D1	D0	
07225h	D15	D14	D13	D12	D11	D10	D9	D8	MBX4B
	D7	D6	D5	D4	D3	D2	D1	D0	
07226h	D15	D14	D13	D12	D11	D10	D9	D8	MBX4C
	D7	D6	D5	D4	D3	D2	D1	D0	
07227h	D15	D14	D13	D12	D11	D10	D9	D8	MBX4D
	D7	D6	D5	D4	D3	D2	D1	D0	
	Message Object #5								
07228h	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	D15	D14	D13	D12	D11	D10	D9	D8	MBX5A
	D7	D6	D5	D4	D3	D2	D1	D0	
0722Dh	D15	D14	D13	D12	D11	D10	D9	D8	MBX5B
	D7	D6	D5	D4	D3	D2	D1	D0	
0722Eh	D15	D14	D13	D12	D11	D10	D9	D8	MBX5C
	D7	D6	D5	D4	D3	D2	D1	D0	
0722Fh	D15	D14	D13	D12	D11	D10	D9	D8	MBX5D
	D7	D6	D5	D4	D3	D2	D1	D0	
07230h to 073FFh	Illegal								
	GENERAL-PURPOSE (GP) TIMER CONFIGURATION CONTROL REGISTERS								
07400h	—	T2STAT	T1STAT	—		T2TOADC		T1TOADC(1)	GPTCON
	T1TOADC(0)	TCOMPOE	—		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 23. 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	
	GENERAL-PURPOSE (GP) TIMER CONFIGURATION CONTROL REGISTERS (CONTINUED)								
07406h	D15	D14	D13	D12	D11	D10	D9	D8	T2CMPR
	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	Illegal								
	FULL AND SIMPLE COMPARE UNIT REGISTERS								
07411h	CENABLE	CLD1	CLD0	SVENABLE	ACTRLD1	ACTRLD0	FCOMPOE	—	COMCON
	—	—	—	—	—	—	—	—	
07412h	Illegal								
07413h	SVRDIR	D2	D1	D0	CMP6ACT1	CMP6ACT0	CMP5ACT1	CMP5ACT0	ACTR
	CMP4ACT1	CMP4ACT0	CMP3ACT1	CMP3ACT0	CMP2ACT1	CMP2ACT0	CMP1ACT1	CMP1ACT0	
07414h	Illegal								
07415h	—	—	—	—	DBT3	DBT2	DBT1	DBT0	DBTCON
	EDBT3	EDBT2	EDBT1	DBTPS2	DBTPS1	DBTPS0	—	—	
07416h	Illegal								
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 to 0741Fh	Illegal								
	CAPTURE UNIT REGISTERS								
07420h	CAPRES	CAPQEPN		CAP3EN	—	CAP3TSEL	CAP12TSEL	CAP3TOADC	CAPCON
	CAP1EDGE		CAP2EDGE		CAP3EDGE		—		
07421h	Illegal								
07422h	—		CAP3FIFO		CAP2FIFO		CAP1FIFO		CAPFIFO
	—	—	—	—	—	—	—	—	
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	Illegal								

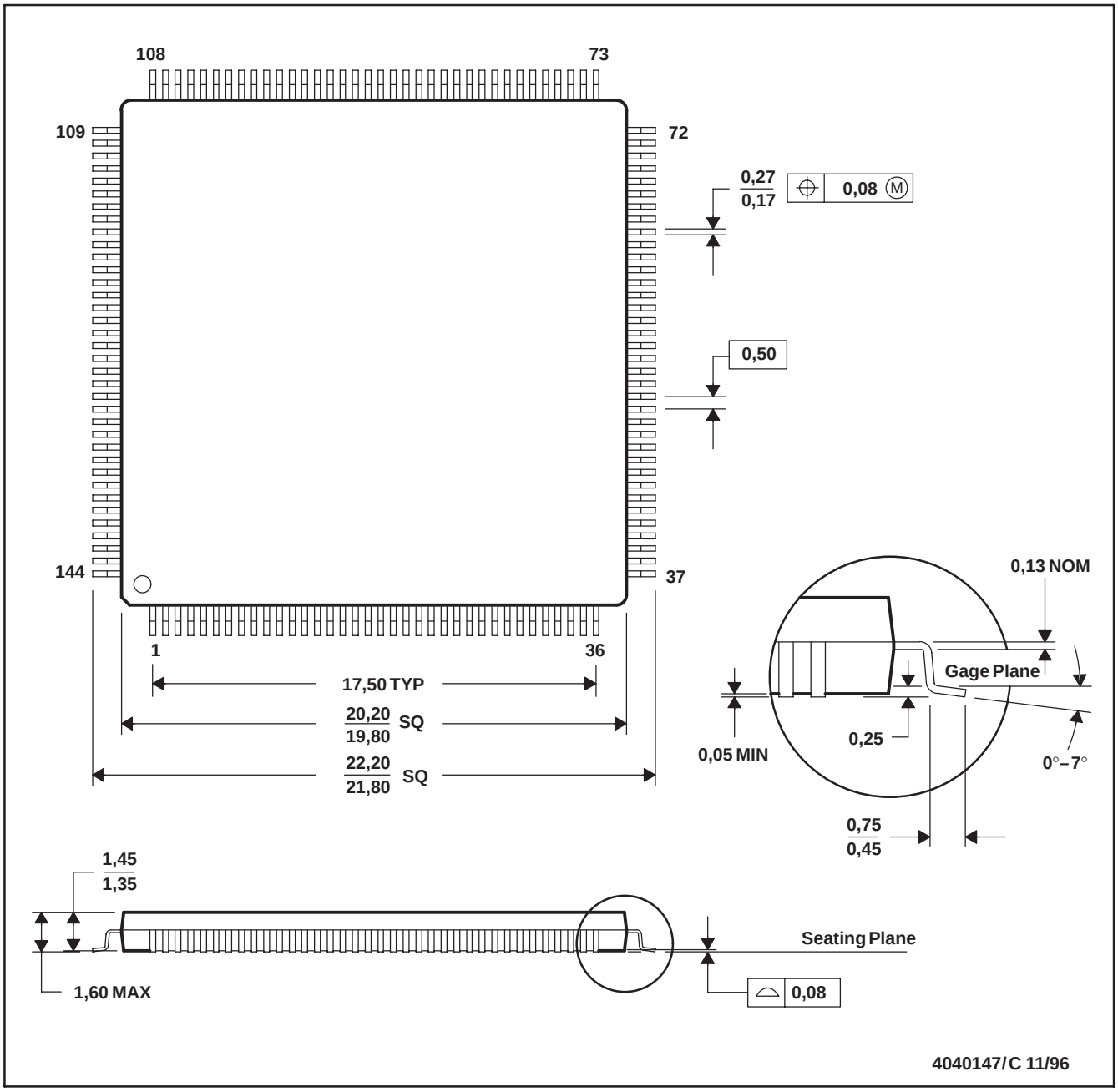
register file compilation (continued)

Table 23. 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)								
07427h	D15	D14	D13	D12	D11	D10	D9	D8	CAP1FBOT
	D7	D6	D5	D4	D3	D2	D1	D0	
07428h	D15	D14	D13	D12	D11	D10	D9	D8	CAP2FBOT
	D7	D6	D5	D4	D3	D2	D1	D0	
07429h	D15	D14	D13	D12	D11	D10	D9	D8	CAP3FBOT
	D7	D6	D5	D4	D3	D2	D1	D0	
0742Ah to 0742Bh	Illegal								
	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 to 0743Fh	Illegal								
	I/O MEMORY SPACE								
	FLASH CONTROL MODE REGISTER								
FF0Fh	—	—	—	—	—	—	—	—	FCMR
	—	—	—	—	—	—	—	—	
	WAIT-STATE GENERATOR CONTROL REGISTER								
0FFFFh	—	—	—	—	—	BVIS.1	BVIS.0	ISWS.2	WSGR
	ISWS.1	ISWS.0	DSWS.2	DSWS.1	DSWS.0	PSWS.2	PSWS.1	PSWS.0	

MECHANICAL DATA

PGE (S-PQFP-G144) **PLASTIC QUAD FLATPACK**



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

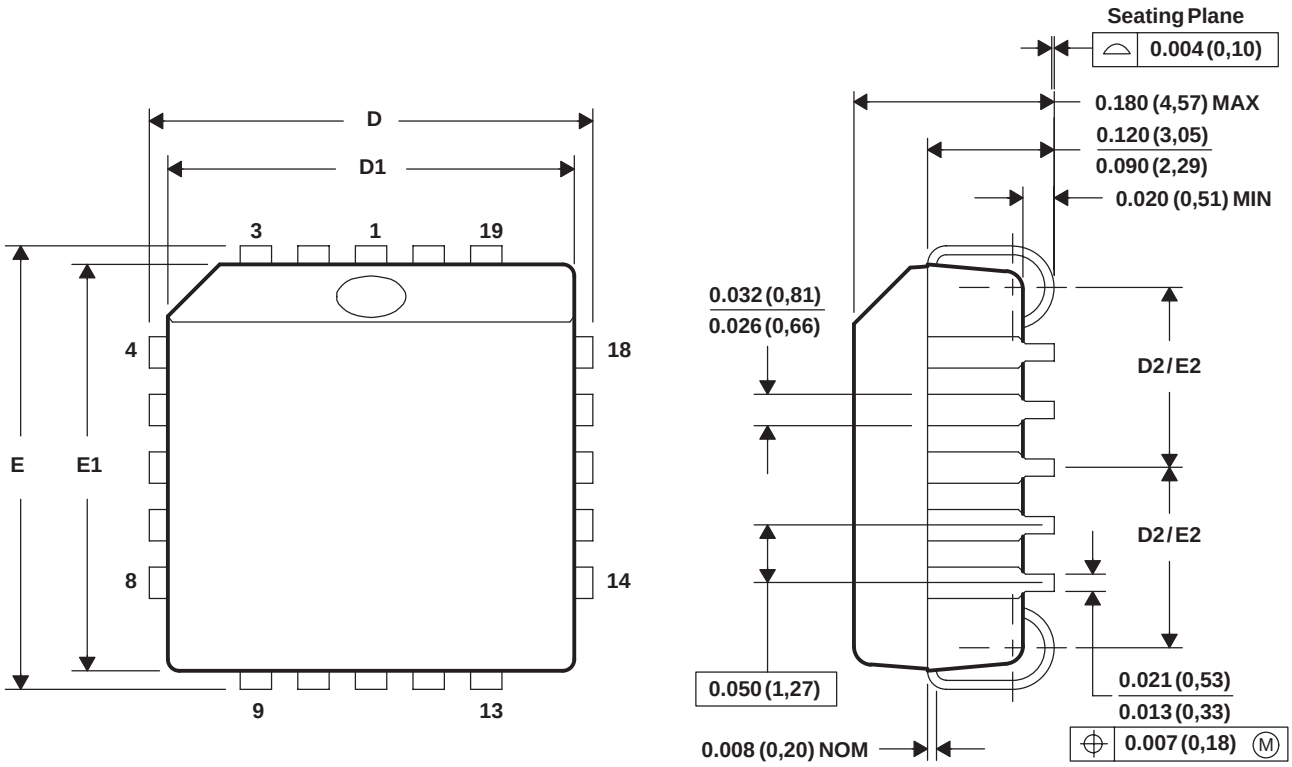
Typical Thermal Resistance Characteristics

PARAMETER	DESCRIPTION	°C/W
Θ_{JA}	Junction-to-ambient	35
Θ_{JC}	Junction-to-case	8.5

MECHANICAL DATA

FN (S-PQCC-J)** **PLASTIC J-LEADED CHIP CARRIER**

20 PIN SHOWN



NO. OF PINS **	D / E		D1 / E1		D2 / E2	
	MIN	MAX	MIN	MAX	MIN	MAX
20	0.385 (9,78)	0.395 (10,03)	0.350 (8,89)	0.356 (9,04)	0.141 (3,58)	0.169 (4,29)
28	0.485 (12,32)	0.495 (12,57)	0.450 (11,43)	0.456 (11,58)	0.191 (4,85)	0.219 (5,56)
44	0.685 (17,40)	0.695 (17,65)	0.650 (16,51)	0.656 (16,66)	0.291 (7,39)	0.319 (8,10)
52	0.785 (19,94)	0.795 (20,19)	0.750 (19,05)	0.756 (19,20)	0.341 (8,66)	0.369 (9,37)
68	0.985 (25,02)	0.995 (25,27)	0.950 (24,13)	0.958 (24,33)	0.441 (11,20)	0.469 (11,91)
84	1.185 (30,10)	1.195 (30,35)	1.150 (29,21)	1.158 (29,41)	0.541 (13,74)	0.569 (14,45)

4040005/ B 03/95

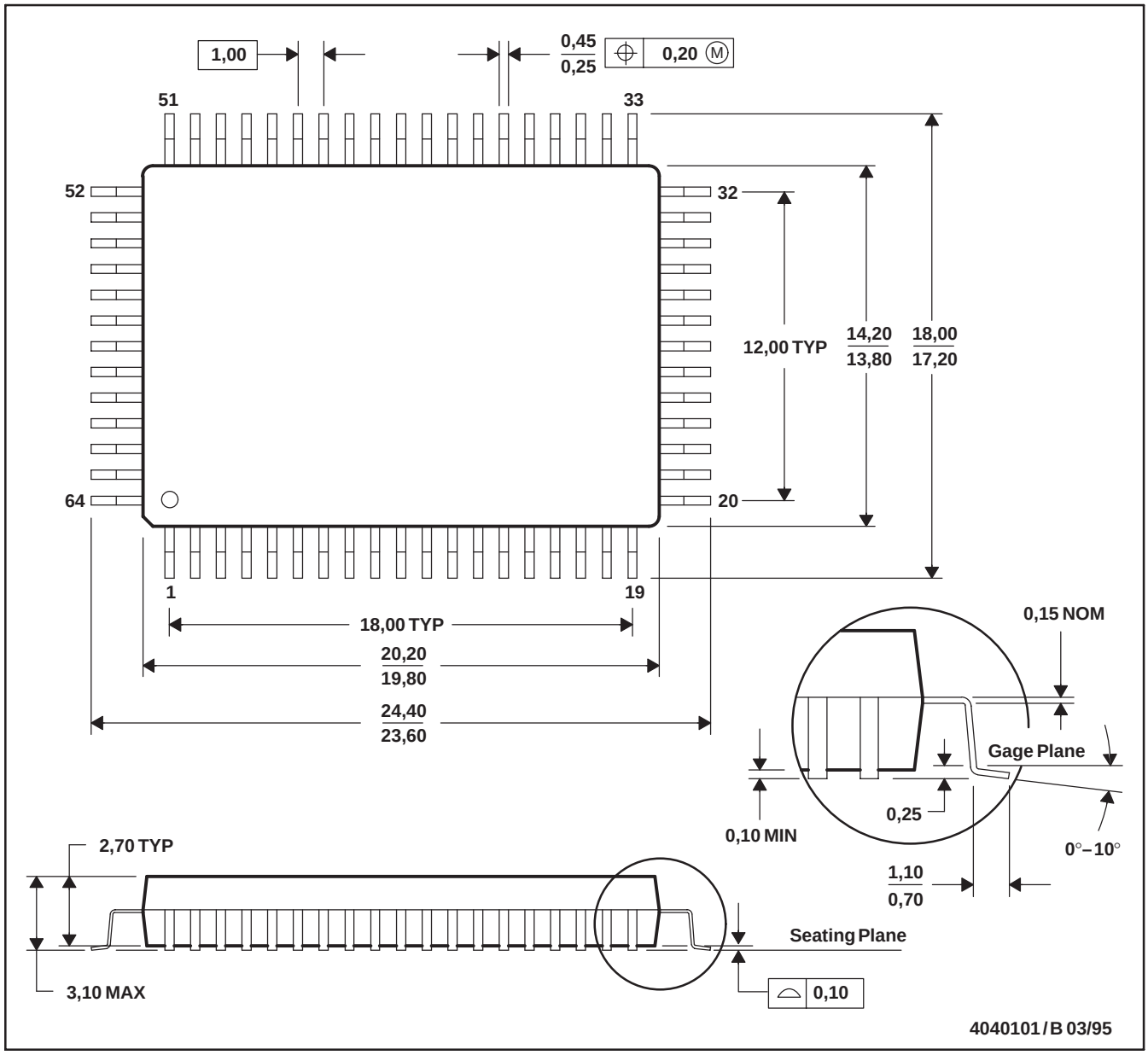
NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-018

Typical Thermal Resistance Characteristics

PARAMETER	DESCRIPTION	°C/W
Θ_{JA}	Junction-to-ambient	48
Θ_{JC}	Junction-to-case	11

MECHANICAL DATA

PG (R-PQFP-G64) **PLASTIC QUAD FLATPACK**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Contact field sales office to determine if a tighter coplanarity requirement is available for this package.

Typical Thermal Resistance Characteristics

PARAMETER	DESCRIPTION	°C/W
Θ_{JA}	Junction-to-ambient	35
Θ_{JC}	Junction-to-case	11

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products 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.

Customers are responsible for their applications using TI components.

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

TI assumes no liability for applications assistance or customer product design. TI does not 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. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.