



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

E·XFI

Product Status	Active
Core Processor	MIPS32® M4K™
Core Size	32-Bit Single-Core
Speed	50MHz
Connectivity	I ² C, IrDA, LINbus, PMP, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, DMA, I ² S, POR, PWM, WDT
Number of I/O	19
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 3.6V
Data Converters	A/D 9x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic32mx250f128bt-50i-so

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

		Pin Nu	mber ⁽¹⁾				
Pin Name	28-pin QFN	28-pin SSOP/ SPDIP/ SOIC	36-pin VTLA	44-pin QFN/ TQFP/ VTLA	Pin Type	Buffer Type	Description
OC1	PPS	PPS	PPS	PPS	0	_	Output Compare Output 1
OC2	PPS	PPS	PPS	PPS	0	—	Output Compare Output 2
OC3	PPS	PPS	PPS	PPS	0	—	Output Compare Output 3
OC4	PPS	PPS	PPS	PPS	0	—	Output Compare Output 4
OC5	PPS	PPS	PPS	PPS	0	—	Output Compare Output 5
OCFA	PPS	PPS	PPS	PPS	I	ST	Output Compare Fault A Input
OCFB	PPS	PPS	PPS	PPS	I	ST	Output Compare Fault B Input
INT0	13	16	17	43	I	ST	External Interrupt 0
INT1	PPS	PPS	PPS	PPS	I	ST	External Interrupt 1
INT2	PPS	PPS	PPS	PPS	I	ST	External Interrupt 2
INT3	PPS	PPS	PPS	PPS	I	ST	External Interrupt 3
INT4	PPS	PPS	PPS	PPS	I	ST	External Interrupt 4
RA0	27	2	33	19	I/O	ST	PORTA is a bidirectional I/O port
RA1	28	3	34	20	I/O	ST	1
RA2	6	9	7	30	I/O	ST	1
RA3	7	10	8	31	I/O	ST	1
RA4	9	12	10	34	I/O	ST	1
RA7	_	_	_	13	I/O	ST	1
RA8	_	_	_	32	I/O	ST	
RA9	_	_	_	35	I/O	ST	1
RA10	_	_	_	12	I/O	ST	
RB0	1	4	35	21	I/O	ST	PORTB is a bidirectional I/O port
RB1	2	5	36	22	I/O	ST	7
RB2	3	6	1	23	I/O	ST	7
RB3	4	7	2	24	I/O	ST	
RB4	8	11	9	33	I/O	ST	
RB5	11	14	15	41	I/O	ST	
RB6	12 ⁽²⁾	15 (2)	16 (2)	42 ⁽²⁾	I/O	ST	
RB7	13	16	17	43	I/O	ST	
RB8	14	17	18	44	I/O	ST	
RB9	15	18	19	1	I/O	ST	
RB10	18	21	24	8	I/O	ST	
RB11	19	22	25	9	I/O	ST	
RB12	20 ⁽²⁾	23 ⁽²⁾	26 ⁽²⁾	10 ⁽²⁾	I/O	ST	1
RB13	21	24	27	11	I/O	ST	1
RB14	22	25	28	14	I/O	ST	1
RB15	23	26	29	15	I/O	ST	
Legend:	CMOS = C	MOS compa	atible input	or output		Analog =	Analog input P = Power
	SI = Schmi	tt Irigger in	put with CN	VIOS levels		O = Outp	out I = Input
Note 1.			lod for rofo	ronco only	See the	"Pin Diag	$m_{\text{rem}} = N/A$

DINOUT I/O DESCRIPTIONS (CONTINUED)

2: Pin number for PIC32MX1XX devices only.

3: Pin number for PIC32MX2XX devices only.

3.2 Architecture Overview

The MIPS32 M4K processor core contains several logic blocks working together in parallel, providing an efficient high-performance computing engine. The following blocks are included with the core:

- Execution Unit
- Multiply/Divide Unit (MDU)
- System Control Coprocessor (CP0)
- Fixed Mapping Translation (FMT)
- Dual Internal Bus interfaces
- Power Management
- MIPS16e[®] Support
- · Enhanced JTAG (EJTAG) Controller

3.2.1 EXECUTION UNIT

The MIPS32 M4K processor core execution unit implements a load/store architecture with single-cycle ALU operations (logical, shift, add, subtract) and an autonomous multiply/divide unit. The core contains thirty-two 32-bit General Purpose Registers (GPRs) used for integer operations and address calculation. The register file consists of two read ports and one write port and is fully bypassed to minimize operation latency in the pipeline.

The execution unit includes:

- · 32-bit adder used for calculating the data address
- Address unit for calculating the next instruction address
- Logic for branch determination and branch target address calculation
- · Load aligner
- Bypass multiplexers used to avoid stalls when executing instruction streams where data producing instructions are followed closely by consumers of their results
- Leading Zero/One detect unit for implementing the CLZ and CLO instructions
- Arithmetic Logic Unit (ALU) for performing bitwise logical operations
- Shifter and store aligner

3.2.2 MULTIPLY/DIVIDE UNIT (MDU)

The MIPS32 M4K processor core includes a Multiply/Divide Unit (MDU) that contains a separate pipeline for multiply and divide operations. This pipeline operates in parallel with the Integer Unit (IU) pipeline and does not stall when the IU pipeline stalls. This allows MDU operations to be partially masked by system stalls and/or other integer unit instructions.

The high-performance MDU consists of a 32x16 booth recoded multiplier, result/accumulation registers (HI and LO), a divide state machine, and the necessary multiplexers and control logic. The first number shown ('32' of 32x16) represents the *rs* operand. The second number ('16' of 32x16) represents the *rt* operand. The PIC32 core only checks the value of the latter (*rt*) operand to determine how many times the operation must pass through the multiplier. The 16x16 and 32x16 operations pass through the multiplier once. A 32x32 operation passes through the multiplier twice.

The MDU supports execution of one 16x16 or 32x16 multiply operation every clock cycle; 32x32 multiply operations can be issued every other clock cycle. Appropriate interlocks are implemented to stall the issuance of back-to-back 32x32 multiply operations. The multiply operand size is automatically determined by logic built into the MDU.

Divide operations are implemented with a simple 1 bit per clock iterative algorithm. An early-in detection checks the sign extension of the dividend (*rs*) operand. If *rs* is 8 bits wide, 23 iterations are skipped. For a 16-bit wide *rs*, 15 iterations are skipped and for a 24-bit wide *rs*, 7 iterations are skipped. Any attempt to issue a subsequent MDU instruction while a divide is still active causes an IU pipeline stall until the divide operation is completed.

Table 3-1 lists the repeat rate (peak issue rate of cycles until the operation can be reissued) and latency (number of cycles until a result is available) for the PIC32 core multiply and divide instructions. The approximate latency and repeat rates are listed in terms of pipeline clocks.

TABLE 3-1:MIPS32[®] M4K[®] PROCESSOR CORE HIGH-PERFORMANCE INTEGERMULTIPLY/DIVIDE UNIT LATENCIES AND REPEAT RATES

Opcode	Operand Size (mul rt) (div rs)	Latency	Repeat Rate
MULT/MULTU, MADD/MADDU,	16 bits	1	1
MSUB/MSUBU	32 bits	2	2
MUL	16 bits	2	1
	32 bits	3	2
DIV/DIVU	8 bits	12	11
	16 bits	19	18
	24 bits	26	25
	32 bits	33	32

Coprocessor 0 also contains the logic for identifying and managing exceptions. Exceptions can be caused by a variety of sources, including alignment errors in data, external events or program errors. Table 3-3 lists the exception types in order of priority.

Exception	Description
Reset	Assertion MCLR or a Power-on Reset (POR).
DSS	EJTAG debug single step.
DINT	EJTAG debug interrupt. Caused by the assertion of the external <i>EJ_DINT</i> input or by setting the EjtagBrk bit in the ECR register.
NMI	Assertion of NMI signal.
Interrupt	Assertion of unmasked hardware or software interrupt signal.
DIB	EJTAG debug hardware instruction break matched.
AdEL	Fetch address alignment error. Fetch reference to protected address.
IBE	Instruction fetch bus error.
DBp	EJTAG breakpoint (execution of SDBBP instruction).
Sys	Execution of SYSCALL instruction.
Вр	Execution of BREAK instruction.
RI	Execution of a reserved instruction.
CpU	Execution of a coprocessor instruction for a coprocessor that is not enabled.
CEU	Execution of a CorExtend instruction when CorExtend is not enabled.
Ov	Execution of an arithmetic instruction that overflowed.
Tr	Execution of a trap (when trap condition is true).
DDBL/DDBS	EJTAG Data Address Break (address only) or EJTAG data value break on store (address + value).
AdEL	Load address alignment error. Load reference to protected address.
AdES	Store address alignment error. Store to protected address.
DBE	Load or store bus error.
DDBL	EJTAG data hardware breakpoint matched in load data compare.

TABLE 3-3: MIPS32[®] M4K[®] PROCESSOR CORE EXCEPTION TYPES

3.3 Power Management

The MIPS M4K processor core offers many power management features, including low-power design, active power management and power-down modes of operation. The core is a static design that supports slowing or Halting the clocks, which reduces system power consumption during Idle periods.

3.3.1 INSTRUCTION-CONTROLLED POWER MANAGEMENT

The mechanism for invoking Power-Down mode is through execution of the WAIT instruction. For more information on power management, see Section 26.0 "Power-Saving Features".

3.4 EJTAG Debug Support

The MIPS M4K processor core provides an Enhanced JTAG (EJTAG) interface for use in the software debug of application and kernel code. In addition to standard User mode and Kernel modes of operation, the M4K core provides a Debug mode that is entered after a debug exception (derived from a hardware breakpoint, single-step exception, etc.) is taken and continues until a Debug Exception Return (DERET) instruction is executed. During this time, the processor executes the debug exception handler routine.

The EJTAG interface operates through the Test Access Port (TAP), a serial communication port used for transferring test data in and out of the core. In addition to the standard JTAG instructions, special instructions defined in the EJTAG specification define which registers are selected and how they are used.

REGISTE	R 9-9: DCHxINT: DMA CHANNEL 'x' INTERRUPT CONTROL REGISTER (CONTINUED)
bit 4	CHDHIF: Channel Destination Half Full Interrupt Flag bit
	 1 = Channel Destination Pointer has reached midpoint of destination (CHDPTR = CHDSIZ/2) 0 = No interrupt is pending
bit 3	CHBCIF: Channel Block Transfer Complete Interrupt Flag bit
	 1 = A block transfer has been completed (the larger of CHSSIZ/CHDSIZ bytes has been transferred), or a pattern match event occurs 0 = No interrupt is pending
bit 2	CHCCIF: Channel Cell Transfer Complete Interrupt Flag bit
	1 = A cell transfer has been completed (CHCSIZ bytes have been transferred)0 = No interrupt is pending
bit 1	CHTAIF: Channel Transfer Abort Interrupt Flag bit
	 1 = An interrupt matching CHAIRQ has been detected and the DMA transfer has been aborted 0 = No interrupt is pending
bit 0	CHERIF: Channel Address Error Interrupt Flag bit
	 1 = A channel address error has been detected (either the source or the destination address is invalid) 0 = No interrupt is pending

DS60001168J-page 96

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0		
24.04	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
31.24	CHSSA<31:24>									
00.40	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
23:10	CHSSA<23:16>									
45.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
15:8	CHSSA<15:8>									
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
	CHSSA<7:0>									

REGISTER 9-10: DCHxSSA: DMA CHANNEL 'x' SOURCE START ADDRESS REGISTER

Legend:				
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'		
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown	

 bit 31-0
 CHSSA<31:0> Channel Source Start Address bits

 Channel source start address.

 Note: This must be the physical address of the source.

REGISTER 9-11: DCHxDSA: DMA CHANNEL 'x' DESTINATION START ADDRESS REGISTER

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0		
01.04	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
31:24	1:24 CHDSA<31:24>									
22:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
23.10	CHDSA<23:16>									
45.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
15:8	CHDSA<15:8>									
7.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
7:0	CHDSA<7:0>									

Legend:				
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'		
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown	

bit 31-0 **CHDSA<31:0>:** Channel Destination Start Address bits Channel destination start address.

 $\ensuremath{\textbf{Note:}}$ This must be the physical address of the destination.

PIC32MX1XX/2XX 28/36/44-PIN FAMILY

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	
21.24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	
31:24	—	—	—	—	—	—	—	—	
00.40	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	
23.10	—	—	—	—	—	—	—	—	
45.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
15:8	CHSSIZ<15:8>								
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
				CHSSIZ	<7:0>				

REGISTER 9-12: DCHxSSIZ: DMA CHANNEL 'x' SOURCE SIZE REGISTER

Legend:R = Readable bitW = Writable bitU = Unimplemented bit, read as '0'-n = Value at POR'1' = Bit is set'0' = Bit is clearedx = Bit is unknown

bit 31-16 Unimplemented: Read as '0'

bit 15-0 CHSSIZ<15:0>: Channel Source Size bits

1111111111111111 = 65,535 byte source size

REGISTER 9-13: DCHxDSIZ: DMA CHANNEL 'x' DESTINATION SIZE REGISTER

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
21.24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
31:24	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHDSIZ<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
				CHDSIZ	<u>/</u> <7:0>			

Legend:				
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'		
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown	

bit 31-16 Unimplemented: Read as '0'

10.0 USB ON-THE-GO (OTG)

Note: This data sheet summarizes the features of the PIC32MX1XX/2XX 28/36/44-pin Family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to Section 27. "USB On-The-Go (OTG)" (DS60001126), which is available from the Documentation > Reference Manual section of the Microchip PIC32 web site (www.microchip.com/pic32).

The Universal Serial Bus (USB) module contains analog and digital components to provide a USB 2.0 Full-Speed and Low-Speed embedded host, Full-Speed device or OTG implementation with a minimum of external components. This module in Host mode is intended for use as an embedded host and therefore does not implement a UHCI or OHCI controller.

The USB module consists of the clock generator, the USB voltage comparators, the transceiver, the Serial Interface Engine (SIE), a dedicated USB DMA controller, pull-up and pull-down resistors, and the register interface. A block diagram of the PIC32 USB OTG module is presented in Figure 10-1.

The clock generator provides the 48 MHz clock required for USB Full-Speed and Low-Speed communication. The voltage comparators monitor the voltage on the VBUS pin to determine the state of the bus. The transceiver provides the analog translation between the USB bus and the digital logic. The SIE is a state machine that transfers data to and from the endpoint buffers and generates the hardware protocol for data transfers. The USB DMA controller transfers data between the data buffers in RAM and the SIE. The integrated pull-up and pull-down resistors eliminate the need for external signaling components. The register interface allows the CPU to configure and communicate with the module. The PIC32 USB module includes the following features:

- · USB Full-Speed support for Host and Device
- Low-Speed Host support
- USB OTG support
- · Integrated signaling resistors
- Integrated analog comparators for VBUS monitoring
- Integrated USB transceiver
- · Transaction handshaking performed by hardware
- · Endpoint buffering anywhere in system RAM
- · Integrated DMA to access system RAM and Flash
- Note: The implementation and use of the USB specifications, as well as other third party specifications or technologies, may require licensing; including, but not limited to, USB Implementers Forum, Inc., also referred to as USB-IF (www.usb.org). The user is fully responsible for investigating and satisfying any applicable licensing obligations.

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
21.24	U-0	U-0						
51.24	—	—		—	—	—	_	—
22.16	U-0	U-0						
23.10	—	—		—	—	—	_	—
15.0	U-0	U-0						
15.0	—	—		—	—	—	_	—
	R/WC-0, HS	R-0	R/WC-0, HS					
7:0	STALLE		RESIMEIE(2)		TRNIE(3)	SOFIF	UERRIF ⁽⁴⁾	URSTIF ⁽⁵⁾
	UTALLI		RESUMEIRY					DETACHIF ⁽⁶⁾

REGISTER 10-6: U1IR: USB INTERRUPT REGISTER

Legend:	WC = Write '1' to clear	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit,	read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown	

bit 31-8 Unimplemented: Read as '0'

bit 7		STALLIF: STALL Handshake Interrupt bit
	-	1 = In Host mode a STALL handshake was received during the handshake phase of the transaction
	I	In Device mode a STALL handshake was transmitted during the handshake phase of the transaction
	(0 = STALL handshake has not been sent
bit 6		ATTACHIF: Peripheral Attach Interrupt bit ⁽¹⁾
	-	1 = Peripheral attachment was detected by the USB module
	(0 = Peripheral attachment was not detected
bit 5		RESUMEIF: Resume Interrupt bit ⁽²⁾
	-	$1 = K$ -State is observed on the D+ or D- pin for 2.5 μ s
	(0 = K-State is not observed
bit 4	I	IDLEIF: Idle Detect Interrupt bit
	-	1 = Idle condition detected (constant Idle state of 3 ms or more)
L:1 0	-	U = NO IDE CONDITION DELECTED
DIT 3		IRNIF: Token Processing Complete Interrupt Dit ^{ery}
	-	$\Gamma = \Gamma$ recessing of current token not complete.
hit 2	Ċ	SOFIE: SOF Taken Interrunt hit
		1 = SOE token received by the peripheral or the SOE threshold reached by the host
	(0 = SOF token was not received nor threshold reached
bit 1	I	UERRIF: USB Error Condition Interrupt bit ⁽⁴⁾
		1 = Unmasked error condition has occurred
	(0 = Unmasked error condition has not occurred
bit 0	l	URSTIF: USB Reset Interrupt bit (Device mode) ⁽⁵⁾
	-	1 = Valid USB Reset has occurred
	(0 = No USB Reset has occurred
		DETACHIF: USB Detach Interrupt bit (Host mode) ⁽⁶⁾
	-	1 = Peripheral detachment was detected by the USB module
	(0 = Peripheral detachment was not detected
Note	1:	This bit is valid only if the HOSTEN bit is set (see Register 10-11), there is no activity on the USB for
		2.5 μ s, and the current bus state is not SE0.
	2:	When not in Suspend mode, this interrupt should be disabled.
	3:	Clearing this bit will cause the STAT FIFO to advance.
	4:	Only error conditions enabled through the U1FIF register will set this bit
	5:	
	6.	Host mode
	υ.	nost mode.

13.0 TIMER2/3, TIMER4/5

Note: This data sheet summarizes the features of the PIC32MX1XX/2XX 28/36/44-pin Family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 14. "Timers"** (DS60001105), which is available from the *Documentation* > *Reference Manual* section of the Microchip PIC32 web site (www.microchip.com/pic32).

This family of PIC32 devices features four synchronous 16-bit timers (default) that can operate as a freerunning interval timer for various timing applications and counting external events. The following modes are supported:

- Synchronous internal 16-bit timer
- Synchronous internal 16-bit gated timer
- · Synchronous external 16-bit timer

Two 32-bit synchronous timers are available by combining Timer2 with Timer3 and Timer4 with Timer5. The 32-bit timers can operate in three modes:

- Synchronous internal 32-bit timer
- · Synchronous internal 32-bit gated timer
- Synchronous external 32-bit timer

Note:	In this chapter, references to registers,
	TxCON, TMRx and PRx, use 'x' to
	represent Timer2 through Timer5 in 16-bit
	modes. In 32-bit modes, 'x' represents
	Timer2 or Timer4 and 'y' represents
	Timer3 or Timer5.

13.1 Additional Supported Features

- · Selectable clock prescaler
- Timers operational during CPU idle
- Time base for Input Capture and Output Compare modules (Timer2 and Timer3 only)
- ADC event trigger (Timer3 in 16-bit mode, Timer2/3 in 32-bit mode)
- Fast bit manipulation using CLR, SET and INV registers

Figure 13-1 and Figure 13-2 illustrate block diagrams of Timer2/3 and Timer4/5.

FIGURE 13-1: TIMER2-TIMER5 BLOCK DIAGRAM (16-BIT)



Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
24.24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
31:24	—	—	—	—	_	—	—	—
00.40	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
23:16	—	—	—	—	—	—	—	—
45.0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
15:8	ON ^(1,2)	—	—	—	—	—	—	—
7:0	U-0	R-y	R-y	R-y	R-y	R-y	R/W-0	R/W-0
	_		S	WDTWINEN	WDTCLR			

REGISTER 14-1: WDTCON: WATCHDOG TIMER CONTROL REGISTER

Legend:	y = Values set from Configuration bits on POR				
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'			
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown		

bit 31-16 Unimplemented: Read as '0'

- bit 15 **ON:** Watchdog Timer Enable bit^(1,2)
 - 1 = Enables the WDT if it is not enabled by the device configuration
 - 0 = Disable the WDT if it was enabled in software
- bit 14-7 Unimplemented: Read as '0'
- bit 6-2 **SWDTPS<4:0>:** Shadow Copy of Watchdog Timer Postscaler Value from Device Configuration bits On reset, these bits are set to the values of the WDTPS <4:0> of Configuration bits.
- bit 1 WDTWINEN: Watchdog Timer Window Enable bit
 - 1 = Enable windowed Watchdog Timer
 - 0 = Disable windowed Watchdog Timer
- bit 0 **WDTCLR:** Watchdog Timer Reset bit
 - 1 = Writing a '1' will clear the WDT
 - 0 = Software cannot force this bit to a '0'
- **Note 1:** A read of this bit results in a '1' if the Watchdog Timer is enabled by the device configuration or software.
 - 2: When using the 1:1 PBCLK divisor, the user's software should not read or write the peripheral's SFRs in the SYSCLK cycle immediately following the instruction that clears the module's ON bit.

REGISTER 17-3: SPIxSTAT: SPI STATUS REGISTER

bit 3 SPITBE: SPI Transmit Buffer Empty Status bit 1 = Transmit buffer, SPIxTXB is empty 0 = Transmit buffer, SPIxTXB is not empty Automatically set in hardware when SPI transfers data from SPIxTXB to SPIxSR. Automatically cleared in hardware when SPIxBUF is written to, loading SPIxTXB. bit 2 Unimplemented: Read as '0' bit 1 SPITBF: SPI Transmit Buffer Full Status bit 1 = Transmit not yet started, SPITXB is full 0 = Transmit buffer is not full Standard Buffer Mode: Automatically set in hardware when the core writes to the SPIBUF location, loading SPITXB. Automatically cleared in hardware when the SPI module transfers data from SPITXB to SPISR. Enhanced Buffer Mode: Set when CWPTR + 1 = SRPTR; cleared otherwise bit 0 SPIRBF: SPI Receive Buffer Full Status bit 1 = Receive buffer, SPIxRXB is full

0 = Receive buffer, SPIxRXB is not full

Standard Buffer Mode:

Automatically set in hardware when the SPI module transfers data from SPIxSR to SPIxRXB. Automatically cleared in hardware when SPIxBUF is read from, reading SPIxRXB.

Enhanced Buffer Mode:

Set when SWPTR + 1 = CRPTR; cleared otherwise

REGISTE	R 18-1:	I2CxCON: I ² C CONTROL REGISTER (CONTINUED)
bit 7	GCEN: Ge	neral Call Enable bit (when operating as I ² C slave)
	1 = Enable (module)	interrupt when a general call address is received in the I2CxRSR e is enabled for reception)
	0 = Genera	al call address is disabled
bit 6	STREN: S	CLx Clock Stretch Enable bit (when operating as I ² C slave)
	Used in co	njunction with SCLREL bit.
	1 = Enable	software or receive clock stretching
b:+ F		$\frac{1}{2}$ solution of the constant of $\frac{1}{2}$ constant of the during sector receives
DILS	ACKDI: A	is transmitted when the software initiates on Asknowledge assumes
	1 = Send a	ACK during an Acknowledge sequence
	0 = Send a	an ACK during an Acknowledge sequence
bit 4	ACKEN: A receive)	cknowledge Sequence Enable bit (when operating as I ² C master, applicable during maste
	1 = Initiate Hardwa 0 = Acknow	Acknowledge sequence on SDAx and SCLx pins and transmit ACKDT data bit. are clear at end of master Acknowledge sequence. wledge sequence not in progress
bit 3	RCEN: Re	ceive Enable bit (when operating as I ² C master)
	1 = Enable 0 = Receiv	s Receive mode for I ² C. Hardware clear at end of eighth bit of master receive data byte. re sequence not in progress
bit 2	PEN: Stop	Condition Enable bit (when operating as I ² C master)
	1 = Initiate 0 = Stop co	Stop condition on SDAx and SCLx pins. Hardware clear at end of master Stop sequence. ondition not in progress
bit 1	RSEN: Re	peated Start Condition Enable bit (when operating as I ² C master)
	1 = Initiate master	Repeated Start condition on SDAx and SCLx pins. Hardware clear at end of Repeated Start sequence.
	0 = Repeat	ted Start condition not in progress
bit 0	SEN: Start	Condition Enable bit (when operating as I ² C master)
	1 = Initiate 0 = Start co	Start condition on SDAx and SCLx pins. Hardware clear at end of master Start sequence. ondition not in progress

Note 1: When using 1:1 PBCLK divisor, the user's software should not read/write the peripheral's SFRs in the SYSCLK cycle immediately following the instruction that clears the module's ON bit.

NOTES:

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0		
24.24	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0		
31:24	—	—	—	—	—	—	CAL<9	CAL<9:8>		
22:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
23.10	CAL<7:0>									
45.0	R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0		
15:8	ON ^(1,2)	—	SIDL	_	—	—	—	—		
7.0	R/W-0	R-0	U-0	U-0	R/W-0	R-0	R-0	R/W-0		
7:0	RTSECSEL ⁽³⁾	RTCCLKON	—	—	RTCWREN ⁽⁴⁾	RTCSYNC	HALFSEC ⁽⁵⁾	RTCOE		

REGISTER 21-1: RTCCON: RTC CONTROL REGISTER

Legend:

Logonal				
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'		
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown	

bit 31-26 Unimplemented: Read as '0'

bit 25-16 CAL<9:0>: RTC Drift Calibration bits, which contain a signed 10-bit integer value 0111111111 = Maximum positive adjustment, adds 511 RTC clock pulses every one minute 000000001 = Minimum positive adjustment, adds 1 RTC clock pulse every one minute 000000000 = No adjustment 1111111111 = Minimum negative adjustment, subtracts 1 RTC clock pulse every one minute 100000000 = Maximum negative adjustment, subtracts 512 clock pulses every one minute ON: RTCC On bit^(1,2) bit 15 1 = RTCC module is enabled 0 = RTCC module is disabled bit 14 Unimplemented: Read as '0' bit 13 SIDL: Stop in Idle Mode bit 1 = Disables the PBCLK to the RTCC when the device enters Idle mode 0 = Continue normal operation when the device enters Idle mode bit 12-8 Unimplemented: Read as '0' bit 7 RTSECSEL: RTCC Seconds Clock Output Select bit⁽³⁾ 1 = RTCC Seconds Clock is selected for the RTCC pin 0 = RTCC Alarm Pulse is selected for the RTCC pin bit 6 RTCCLKON: RTCC Clock Enable Status bit 1 = RTCC Clock is actively running 0 = RTCC Clock is not running **Note 1:** The ON bit is only writable when RTCWREN = 1. 2: When using the 1:1 PBCLK divisor, the user's software should not read/write the peripheral's SFRs in the SYSCLK cycle immediately following the instruction that clears the module's ON bit. 3: Requires RTCOE = 1 (RTCCON<0>) for the output to be active. 4: The RTCWREN bit can be set only when the write sequence is enabled. 5: This bit is read-only. It is cleared to '0' on a write to the seconds bit fields (RTCTIME<14:8>).

Note: This register is reset only on a Power-on Reset (POR).

© 2011-2016 Microchip Technology Inc.

PIC32MX1XX/2XX 28/36/44-PIN FAMILY

REGISTER 22-3: AD1CON3: ADC CONTROL REGISTER 3

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	
	—	—	—	—	—	—	—	—	
	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	
23:16	—	—	—	—	_	—	—	—	
45.0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
15:8	ADRC	—	—	SAMC<4:0> ⁽¹⁾					
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W	R/W-0	
				ADCS<	7:0> ⁽²⁾				

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'			
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown		

bit 31-16 Unimplemented: Read as '0'

- bit 15 ADRC: ADC Conversion Clock Source bit
 - 1 = Clock derived from FRC
 - 0 = Clock derived from Peripheral Bus Clock (PBCLK)
- bit 14-13 Unimplemented: Read as '0'
- - 00000001 =TPB • 2 • (ADCS<7:0> + 1) = 4 • TPB = TAD 00000000 =TPB • 2 • (ADCS<7:0> + 1) = 2 • TPB = TAD
- **Note 1:** This bit is only used if the SSRC<2:0> bits (AD1CON1<7:5>) = 111.
 - 2: This bit is not used if the ADRC (AD1CON3<15>) bit = 1.

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
21.24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
31.24	—	—	—	—	—	—	—	—
00.40	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
23:16	—	—	_	—	_	—	—	—
45.0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
15:8	ON ⁽¹⁾	—	—	—	—	—	—	—
7:0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	_	CVROE	CVRR	CVRSS	CVR<3:0>			

REGISTER 24-1: CVRCON: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'						
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown					

bit 31-16 Unimplemented: Read as '0'

- bit 15 **ON:** Comparator Voltage Reference On bit⁽¹⁾
 - 1 = Module is enabled
 - Setting this bit does not affect other bits in the register.
 - 0 = Module is disabled and does not consume current.
 - Clearing this bit does not affect the other bits in the register.
- bit 14-7 Unimplemented: Read as '0'
- bit 6 **CVROE:** CVREFOUT Enable bit
 - 1 = Voltage level is output on CVREFOUT pin
 - 0 = Voltage level is disconnected from CVREFOUT pin
- bit 5 CVRR: CVREF Range Selection bit
 - 1 = 0 to 0.67 CVRSRC, with CVRSRC/24 step size
 - 0 = 0.25 CVRSRC to 0.75 CVRSRC, with CVRSRC/32 step size
- bit 4 **CVRSS:** CVREF Source Selection bit
 - 1 = Comparator voltage reference source, CVRSRC = (VREF+) (VREF-)
 - 0 = Comparator voltage reference source, CVRSRC = AVDD AVSS
- bit 3-0 **CVR<3:0>:** CVREF Value Selection $0 \le CVR<3:0> \le 15$ bits

<u>When CVRR = 1:</u> CVREF = (CVR<3:0>/24) • (CVRSRC) <u>When CVRR = 0:</u> CVREF = 1/4 • (CVRSRC) + (CVR<3:0>/32) • (CVRSRC)

Note 1: When using 1:1 PBCLK divisor, the user's software should not read/write the peripheral's SFRs in the SYSCLK cycle immediately following the instruction that clears the module's ON bit.

25.1 CTMU Control Registers

TABLE 25-1: CTMU REGISTER MAP

Virtual Address (BF80_#)	Register Name ⁽¹⁾	Bit Range		Bits															
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	All Reset
A200	CTMUCON	31:16	EDG1MOD	EDG1POL	EDG1SEL<3:0>				EDG2STAT	EDG1STAT	EDG2MOD	EDG2POL	EDG2SEL<3:0>				_	—	0000
		15:0	ON	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG			ITRIM•	<5:0>			IRNG	<1:0>	0000

Legend: x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: All registers in this table have corresponding CLR, SET and INV registers at its virtual address, plus an offset of 0x4, 0x8 and 0xC, respectively. See Section 11.2 "CLR, SET and INV Registers" for more information.

NOTES:

29.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- · Rich directive set
- Flexible macro language
- · MPLAB X IDE compatibility

29.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel[®] standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- · Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

29.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

29.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- · Support for the entire device instruction set
- · Support for fixed-point and floating-point data
- Command-line interface
- · Rich directive set
- Flexible macro language
- · MPLAB X IDE compatibility

PIC32MX1XX/2XX 28/36/44-PIN FAMILY

NOTES: