

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	80MHz
Connectivity	Ethernet, I ² C, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	85
Program Memory Size	64KB (64K 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 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic32mx664f064lt-i-pt

Email: info@E-XFL.COM

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

Referenced Sources

This device data sheet is based on the following individual chapters of the *"PIC32 Family Reference Manual"*. These documents should be considered as the general reference for the operation of a particular module or device feature.

- Note 1: To access the documents listed below, browse to the documentation section of the PIC32MX795F512L product page on the Microchip web site (www.microchip.com) or select a family reference manual section from the following list. In addition to parameters, features, and other documentation, the resulting page provides links to the related family
- Section 1. "Introduction" (DS60001127)
- Section 2. "CPU" (DS60001113)
- Section 4. "Prefetch Cache" (DS60001119)
- Section 3. "Memory Organization" (DS60001115)

reference manual sections.

- Section 5. "Flash Program Memory" (DS60001121)
- Section 6. "Oscillator Configuration" (DS60001112)
- Section 7. "Resets" (DS60001118)
- Section 8. "Interrupt Controller" (DS60001108)
- Section 9. "Watchdog Timer and Power-up Timer (DS60001114)
- Section 10. "Power-Saving Features" (DS60001130)
- Section 12. "I/O Ports" (DS60001120)
- Section 13. "Parallel Master Port (PMP)" (DS60001128)
- Section 14. "Timers" (DS60001105)
- Section 15. "Input Capture" (DS60001122)
- Section 16. "Output Capture" (DS60001111)
- Section 17. "10-bit Analog-to-Digital Converter (ADC)" (DS60001104)
- Section 19. "Comparator" (DS60001110)
- Section 20. "Comparator Voltage Reference (CVREF)" (DS60001109)
- Section 21. "Universal Asynchronous Receiver Transmitter (UART)" (DS60001107)
- Section 23. "Serial Peripheral Interface (SPI)" (DS60001106)
- Section 24. "Inter-Integrated Circuit (I2C)" (DS60001116)
- Section 27. "USB On-The-Go (OTG)" (DS60001126)
- Section 29. "Real-Time Clock and Calendar (RTCC)" (DS60001125)
- Section 31. "Direct Memory Access (DMA) Controller" (DS60001117)
- Section 32. "Configuration" (DS60001124)
- Section 33. "Programming and Diagnostics" (DS60001129)
- Section 34. "Controller Area Network (CAN)" (DS60001154)
- Section 35. "Ethernet Controller" (DS60001155)

FIGURE 4-6: MEMORY MAP ON RESET FOR PIC32MX695F512H, PIC32MX695F512L, PIC32MX795F512H AND PIC32MX795F512L DEVICES



2: The size of this memory region is programmable (see Section 3. "Memory Organization" (DS60001115)) and can be changed by initialization code provided by end user development tools (refer to the specific development tool documentation for information).

5.0 FLASH PROGRAM MEMORY

Note: This data sheet summarizes the features of the PIC32MX5XX/6XX/7XX family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to Section 5. "Flash Program Memory" (DS60001121) in the "PIC32 Family Reference Manual", which is available from the Microchip web site (www.microchip.com/PIC32). PIC32MX5XX/6XX/7XX devices contain an internal Flash program memory for executing user code. There are three methods by which the user can program this memory:

- Run-Time Self-Programming (RTSP)
- EJTAG Programming
- In-Circuit Serial Programming[™] (ICSP[™])

RTSP is performed by software executing from either Flash or RAM memory. Information about RTSP techniques is available in **Section 5. "Flash Program Memory"** (DS60001121) in the *"PIC32 Family Reference Manual"*.

EJTAG is performed using the EJTAG port of the device and an EJTAG capable programmer.

ICSP is performed using a serial data connection to the device and allows much faster programming times than RTSP.

The EJTAG and ICSP methods are described in the "*PIC32 Flash Programming Specification*" (DS60001145), which can be downloaded from the Microchip web site.

Note: For PIC32MX5XX/6XX/7XX devices, the Flash page size is 4 KB and the row size is 512 bytes (1024 IW and 128 IW, respectively).

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
04.04	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	R/W-0
23:16	—	—	—	—	—	-	—	CHECOH
15.0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
10.0	—	—	—	—	—		DCSZ	ː< 1:0>
7.0	U-0	U-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1
7:0	—	—	PREFEN<1:0>		—	PFMWS<2:0>		

REGISTER 9-1: CHECON: CACHE CONTROL REGISTER

Legend:

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

bit 31-17 Unimplemented: Write '0'; ignore read

- bit 16 CHECOH: Cache Coherency Setting on a PFM Program Cycle bit
 - 1 = Invalidate all data and instruction lines
 - 0 = Invalidate all data lnes and instruction lines that are not locked
- bit 15-10 Unimplemented: Write '0'; ignore read
- bit 9-8 DCSZ<1:0>: Data Cache Size in Lines bits
 - Changing these bits causes all lines to be reinitialized to the "invalid" state.
 - 11 = Enable data caching with a size of 4 lines
 - 10 = Enable data caching with a size of 2 lines
 - 01 = Enable data caching with a size of 1 line
 - 00 = Disable data caching
- bit 7-6 Unimplemented: Write '0'; ignore read
- bit 5-4 **PREFEN<1:0>:** Predictive Prefetch Enable bits
 - 11 = Enable predictive prefetch for both cacheable and non-cacheable regions
 - 10 = Enable predictive prefetch only for non-cacheable regions
 - 01 = Enable predictive prefetch only for cacheable regions
 - 00 = Disable predictive prefetch
- bit 3 Unimplemented: Write '0'; ignore read

bit 2-0 PFMWS<2:0>: PFM Access Time Defined in Terms of SYSLK Wait States bits

- 111 = Seven Wait states
- 110 = Six Wait states
- 101 = Five Wait states
- 100 = Four Wait states
- 011 = Three Wait states
- 010 = Two Wait states
- 001 = One Wait state
- 000 = Zero Wait state

REGISTER 10-9: DCHxINT: DMA CHANNEL 'x' INTERRUPT CONTROL REGISTER (CONTINUED)

- bit 5 **CHDDIF:** Channel Destination Done Interrupt Flag bit
 - 1 = Channel Destination Pointer has reached end of destination (CHDPTR = CHDSIZ)
 - 0 = No interrupt is pending
- 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

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	—	—	—	—	—	—	—	—
	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
23:16	—	—	—	—	—	—	—	—
15.0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
15:8	—	—	—	—	—	—	—	—
7.0	U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
7:0						FRMH<2:0>		

REGISTER 11-14: U1FRMH: USB FRAME NUMBER HIGH REGISTER

Legend:

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

bit 31-3 Unimplemented: Read as '0'

bit 2-0 **FRMH<2:0>:** Upper 3 bits of the Frame Numbers bits These register bits are updated with the current frame number whenever a SOF TOKEN is received.

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
	—	—	—	—	—	—	—	—
15.0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
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	PID<3:0>				EP<3:0>			

REGISTER 11-15: U1TOK: USB TOKEN REGISTER

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, re	ead 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 7-4 PID<3:0>: Token Type Indicator bits⁽¹⁾ 1101 = SETUP (TX) token type transaction 1001 = IN (RX) token type transaction 0001 = OUT (TX) token type transaction Note: All other values not listed, are Reserved and must not be used.

bit 3-0 **EP<3:0>:** Token Command Endpoint Address bits The four bit value must specify a valid endpoint.

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
	—	—		—			_	_
15.0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
15:8	—	—		—			_	_
7:0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	LSPD	RETRYDIS		EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK

REGISTER 11-21: U1EP0-U1EP15: USB ENDPOINT CONTROL REGISTER

Legend:

bit 1

R = Readable bit	W = Writable bit	U = Unimplemented bit, re	ad 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 LSPD: Low-Speed Direct Connection Enable bit (Host mode and U1EP0 only)
 - 1 = Direct connection to a low-speed device enabled
 - 0 = Direct connection to a low-speed device disabled; hub required with PRE_PID
- bit 6 **RETRYDIS:** Retry Disable bit (Host mode and U1EP0 only)
 - 1 = Retry NACK'd transactions disabled
 - 0 = Retry NACK'd transactions enabled; retry done in hardware
- bit 5 Unimplemented: Read as '0'
- bit 4 **EPCONDIS:** Bidirectional Endpoint Control bit
 - If EPTXEN = 1 and EPRXEN = 1:
 - 1 = Disable Endpoint 'n' from control transfers; only TX and RX transfers are allowed
 - 0 = Enable Endpoint 'n' for control (SETUP) transfers; TX and RX transfers are also allowed Otherwise, this bit is ignored.
- bit 3 EPRXEN: Endpoint Receive Enable bit
 - 1 = Endpoint 'n' receive is enabled
 - 0 = Endpoint 'n' receive is disabled
- bit 2 EPTXEN: Endpoint Transmit Enable bit
 - 1 = Endpoint 'n' transmit is enabled
 - 0 = Endpoint 'n' transmit is disabled
 - EPSTALL: Endpoint Stall Status bit
 - 1 = Endpoint 'n' was stalled
 - 0 = Endpoint 'n' was not stalled
- bit 0 **EPHSHK:** Endpoint Handshake Enable bit
 - 1 = Endpoint Handshake is enabled
 - 0 = Endpoint Handshake is disabled (typically used for isochronous endpoints)

NOTES:

REGISTER 21-1: PMCON: PARALLEL PORT CONTROL REGISTER (CONTINUED)

- bit 3 **CS1P:** Chip Select 0 Polarity bit⁽²⁾
 - 1 = Active-high (PMCS1)
 - $0 = \text{Active-low}(\overline{\text{PMCS1}})$
- bit 2 Unimplemented: Read as '0'
- bit 1 WRSP: Write Strobe Polarity bit
 - For Slave Modes and Master mode 2 (PMMODE<9:8> = 00,01,10):
 - 1 = Write strobe active-high (PMWR)
 - $0 = Write strobe active-low (\overline{PMWR})$

For Master mode 1 (PMMODE<9:8> = 11):

- 1 = Enable strobe active-high (PMENB)
- 0 = Enable strobe active-low (PMENB)
- bit 0 RDSP: Read Strobe Polarity bit
 - For Slave modes and Master mode 2 (PMMODE<9:8> = 00,01,10):
 - 1 = Read Strobe active-high (PMRD)
 - 0 = Read Strobe active-low (PMRD)

For Master mode 1 (PMMODE<9:8> = 11):

- 1 = Read/write strobe active-high (PMRD/ \overline{PMWR})
- 0 = Read/write strobe active-low (PMRD/PMWR)
- **Note 1:** 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 control bit.
 - 2: These bits have no effect when their corresponding pins are used as address lines.

NOTES:

22.0 REAL-TIME CLOCK AND CALENDAR (RTCC)

Note: This data sheet summarizes the features of the PIC32MX5XX/6XX/7XX family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to Section 29. "Real-Time Clock and Calendar (RTCC)" (DS60001125) in the "PIC32 Family Reference Manual", which is available from the Microchip web site (www.microchip.com/PIC32).

The PIC32 RTCC module is intended for applications in which accurate time must be maintained for extended periods of time with minimal or no CPU intervention. Low-power optimization provides extended battery lifetime while keeping track of time. A simplified block diagram of the RTCC module is illustrated in Figure 22-1. Key features of the RTCC module include:

- Time: hours, minutes and seconds
- 24-hour format (military time)
- · Visibility of one-half second period
- Provides calendar: Weekday, date, month and year
- Alarm intervals are configurable for half of a second, one second, 10 seconds, one minute, 10 minutes, one hour, one day, one week, one month and one year
- Alarm repeat with decrementing counter
- Alarm with indefinite repeat: Chime
- Year range: 2000 to 2099
- Leap year correction
- · BCD format for smaller firmware overhead
- Optimized for long-term battery operation
- Fractional second synchronization
- User calibration of the clock crystal frequency with auto-adjust
- Calibration range: ±0.66 seconds error per month
- · Calibrates up to 260 ppm of crystal error
- Requirements: External 32.768 kHz clock crystal
- Alarm pulse or seconds clock output on RTCC pin



FIGURE 22-1: RTCC BLOCK DIAGRAM

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	—	—	—	—	—	—	—	—
22.10	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
23.10	—	WAKFIL	—	—	—	SEG2PH<2:0> ^(1,4)		
15.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
10.0	SEG2PHTS ⁽¹⁾	SAM ⁽²⁾	:	SEG1PH<2:0	>	P	RSEG<2:0>	
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	SJW<1:0> ⁽³⁾			BRP<5:0>				

REGISTER 24-2: CiCFG: CAN BAUD RATE CONFIGURATION REGISTER

Legend:	HC = Hardware Clear	S = Settable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, re	ead as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 31-23 Unimplemented: Read as '0'

- bit 22 **WAKFIL:** CAN Bus Line Filter Enable bit 1 = Use CAN bus line filter for wake-up 0 = CAN bus line filter is not used for wake-up
- bit 21-19 Unimplemented: Read as '0'

bit 18-16	SEG2PH<2:0>: Phase Buffer Segment 2 bits(1,4)
	111 = Length is 8 x TQ
	•
	•
	•
	000 = Length is 1 x TQ
bit 15	SEG2PHTS: Phase Segment 2 Time Select bit ⁽¹⁾
	1 = Freely programmable
	0 = Maximum of SEG1PH or Information Processing Time, whichever is greater
bit 14	SAM: Sample of the CAN Bus Line bit ⁽²⁾
	1 = Bus line is sampled three times at the sample point
	0 = Bus line is sampled once at the sample point
bit 13-11	SEG1PH<2:0>: Phase Buffer Segment 1 bits ⁽⁴⁾
	111 = Length is 8 x TQ
	•
	•
	•
	000 = Length is 1 x Tq
Note 1:	SEG2PH
2:	3 Time bit sampling is not allowed for BRP < 2.
3:	SJW ≤ SEG2PH.
4:	The Time Quanta per bit must be greater than 7 (that is, $TOBIT > 7$).
••	

Note: This register can only be modified when the CAN module is in Configuration mode (OPMOD<2:0> (CiCON<23:21>) = 100).

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
	—	—	—	—	—	—	—	—
00.40	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
23.10	—	—	—	—	—	—	—	—
15.0	U-0	U-0	U-0	R-0	R-0	R-0	R-0	R-0
15.0	—	—	—			FILHIT<4:0>		
7:0	U-0	R-1	R-0	R-0	R-0	R-0	R-0	R-0
7.0	_				CODE<6:0> ⁽¹)		

REGISTER 24-4: CiVEC: CAN INTERRUPT CODE REGISTER

Legend:

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

bit 31-13 Unimplemented: Read as '0'

bit 12-8	FILHIT<4:0>: Filter Hit Number bit
	11111 = Filter 31
	11110 = Filter 30
	•
	•
	• 00001 = Filter 1
	00000 = Filter 0
bit 7	Unimplemented: Read as '0'
bit 6-0	ICODE<6:0>: Interrupt Flag Code bits ⁽¹⁾
	1111111 = Reserved
	•
	•
	•
	1001001 = Reserved
	1001000 = Invalid message received (IVRIF)
	1000111 = CAN module mode change (MODIF)
	1000110 - Bus bandwidth error (SERRIE)
	1000100 = Address error interrupt (SERRIF)
	1000011 = Receive FIFO overflow interrupt (RBOVIF)
	1000010 = Wake-up interrupt (WAKIF)
	1000001 = Error Interrupt (CERRIF)
	1000000 = No interrupt
	0111111 = Reserved
	•
	•
	010000 = Reserved
	0011111 = FIFO31 Interrupt (CiFSTAT<31> set)
	0011110 = FIFO30 Interrupt (CiFSTAT<30> set)
	•
	•
	• $0.00001 = \text{EIEO1 Interrupt (CIESTAT -1)}$
	$C_{0} = C_{0} = C_{0$
	0000000 = rir O0 interrupt (Or Stat < 0 > Set)

Note 1: These bits are only updated for enabled interrupts.

25.1 Control Registers

TABLE 25-5: ETHERNET CONTROLLER REGISTER SUMMARY FOR PIC32MX664F064H, PIC32MX664F128H, PIC32MX664F064L, PIC32MX664F128L, PIC32MX675F256H, PIC32MX675F512H, PIC32MX695F512H, PIC32MX775F256H, PIC32MX775F512H, PIC32MX795F512H, PIC32MX695F512L, PIC32MX675F256L, PIC32MX675F512L, PIC32MX764F128H, PIC32MX764F128L, PIC32MX775F256L, PIC32MX775F512L AND PIC32MX795F512L DEVICES

ess										В	its								
Virtual Addr (BF88_#)	Register Name ⁽¹⁾	Bit Range	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 Resets
9000	ETHCON1	31:16								PTV<	:15:0>				-				0000
3000	EINOON	15:0	ON	_	SIDL	_	_		TXRTS	RXEN	AUTOFC	—	—	MANFC	_	—	—	BUFCDEC	0000
9010	ETHCON2	31:16	_	—	_	—	_	_	—	_	—	—	—	—	_	—		—	0000
		15:0	—	_	_	—	—			R	XBUFSZ<6:0	>			_	—	_	_	0000
9020	ETHTXST	31:16							TYOTAD	TXSTADE)R<31:16>								0000
		15:0							TASTADL	RYSTAD	R-31-16-						_	—	0000
9030	ETHRXST	15:0							RXSTAD)R<15:2>	//<51.102						_	_	0000
		31:16																	0000
9040	ETHHT0	15:0								HT<	31:0>								0000
9050	ETHHT1	31:16								HT-6	3.32								0000
3030	E111111	15:0								11150	0.022								0000
9060	ETHPMM0	31:16								PMM-	<31:0>								0000
		15:0																	0000
9070	ETHPMM1	15.0								PMM<	63:32>								0000
		31:16		_	_	_	_		_		_	_	_	_	_		_	_	0000
9080	ETHPMCS	15:0								PMCS	<15:0>								0000
0000	ETHOMO	31:16	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	0000
9090	ETHENIO	15:0								PMO.	<15:0>								0000
0040	FTUDVEO	31:16	_	—	_	—	—	_	-	_	—	—		—	—	—	_	—	0000
90A0	ETHRAFC	15:0	HTEN	MPEN	—	NOTPM		PMMO	DE<3:0>		CRC ERREN	CRC OKEN	RUNT ERREN	RUNTEN	UCEN	NOT MEEN	MCEN	BCEN	0000
90B0	ETHRXWM	31:16	—	_	—	—	—	—	_	—				RXFW	M<7:0>				0000
0020		15:0	_		_	_	_	_	_	_				RXEW	M<7:0>				0000
0000		31:16	_	-	-	_	_	_	-	-	-	-	-	_	-	-	-	-	0000
3000		15:0	_	I X BUSEIE	RX BUSEIE	—	—	_	EW MARKIE	FW MARKIE	RX DONEIE	PK TPENDIE	ACTIE	—	I X DONEIE	I X ABORTIE	RX BUFNAIE	RX OVFLWIE	0000
90D0	ETHIRQ	31:16	_	_	—	_	_	_	—	_	—	—	_	_	—	—	_	_	0000
		15:0	—	TXBUSE	RXBUSE	—	—	—	EWMARK	FWMARK	RXDONE	PKTPEND	RXACT	—	TXDONE	TXABORT	RXBUFNA	RXOVFLW	0000

Note 1: All registers in this table (with the exception of ETHSTAT) have corresponding CLR, SET and INV registers at their virtual addresses, plus offsets of 0x4, 0x8 and 0xC, respectively. See Section 12.1.1 "CLR, SET and INV Registers" for more information.

2: Reset values default to the factory programmed value.

REGISTER 25-21: ETHFCSERR: ETHERNET CONTROLLER FRAME CHECK SEQUENCE ERROR STATISTICS 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		
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0		
	—	—	—		—	—	-	—		
00.40	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0		
23.10	—	—	—		—	—	-	—		
15.0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
15.0	FCSERRCNT<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	FCSERRCNT<7:0>									

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, re	ead 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 15-0 **FCSERRCNT<15:0>:** FCS Error Count bits Increment count for frames received with FCS error and the frame length in bits is an integral multiple of 8 bits.

Note 1: This register is only used for RX operations.

2: This register is automatically cleared by hardware after a read operation, unless the byte enables for bytes 0/1 are '0'.

3: It is recommended to use the SET, CLR, or INV registers to set or clear any bit in this register. Setting or clearing any bits in this register should be only done for debug/test purposes.

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	
04-04	R	R	R	R	R	R	R	R	
31:24		VER<3	3:0> ⁽¹⁾			DEVID<2	27:24> ⁽¹⁾		
00.40	R	R	R	R	R	R	R	R	
23:16	DEVID<23:16> ⁽¹⁾								
45.0	R	R	R	R	R	R	R	R	
15:8 DEVID<15:8> ⁽¹⁾									
7.0	R	R	R	R	R	R	R	R	
7:0				DEVID<	7:0> ⁽¹⁾				

REGISTER 29-5: DEVID: DEVICE AND REVISION ID REGISTER

Legend:

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

bit 31-28 VER<3:0>: Revision Identifier bits⁽¹⁾

bit 27-0 DEVID<27:0>: Device ID bits⁽¹⁾

Note 1: See the "PIC32 Flash Programming Specification" (DS60001145) for a list of Revision and Device ID values.

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	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
15:8	—	—						—
7.0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	U-0	R/W-0
7:0	_	_	_	_	JTAGEN	TROEN	_	TDOEN

REGISTER 29-6: DDPCON: DEBUG DATA PORT CONTROL REGISTER

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

bit 31-4 Unimplemented: Read as '0'

- bit 3 JTAGEN: JTAG Port Enable bit
 - 1 = Enable the JTAG port
 - 0 = Disable the JTAG port
- bit 2 **TROEN:** Trace Output Enable bit
 - 1 = Enable the trace port
 - 0 = Disable the trace port
- bit 1 Unimplemented: Read as '0'
- bit 0 TDOEN: TDO Enable for 2-Wire JTAG
 - 1 = 2-wire JTAG protocol uses TDO
 - 0 = 2-wire JTAG protocol does not use TDO

31.0 DEVELOPMENT SUPPORT

The PIC[®] microcontrollers (MCU) and dsPIC[®] digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
- MPLAB[®] X IDE Software
- Compilers/Assemblers/Linkers
 - MPLAB XC Compiler
 - MPASM[™] Assembler
 - MPLINK[™] Object Linker/ MPLIB[™] Object Librarian
 - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
 - MPLAB X SIM Software Simulator
- Emulators
 - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
 - MPLAB ICD 3
 - PICkit™ 3
- Device Programmers
 - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

31.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows[®], Linux and Mac OS[®] X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- · Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- · Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window
- Project-Based Workspaces:
- Multiple projects
- Multiple tools
- Multiple configurations
- · Simultaneous debugging sessions

File History and Bug Tracking:

- · Local file history feature
- Built-in support for Bugzilla issue tracker

31.2 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

31.3 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

31.4 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

31.5 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

100-Lead Plastic Thin Quad Flatpack (PF) - 14x14x1 mm Body 2.00 mm Footprint [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	MILLIMETERS			
Dimensio	Dimension Limits			MAX
Contact Pitch	E	0.50 BSC		
Contact Pad Spacing	C1		15.40	
Contact Pad Spacing	C2		15.40	
Contact Pad Width (X100)	X1			0.30
Contact Pad Length (X100)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2110B

121-Ball Plastic Thin Profile Fine Pitch Ball Grid Array (BG) - 10x10x1.10 mm Body [TFBGA]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-148 Rev F Sheet 1 of 2