

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 "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	MIPS32® M4K™
Core Size	32-Bit Single-Core
Speed	80MHz
Connectivity	CANbus, Ethernet, I²C, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	53
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 3.6V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic32mx775f256ht-80v-pt

PIC32MX5XX/6XX/7XX

TABLE 1: PIC32MX5XX USB AND CAN FEATURES

USB and CAN																
Device	Pins	Program Memory (KB)	Data Memory (KB)	USB	CAN	Timers/Capture/Compare	DMA Channels (Programmable/Dedicated)	UART ^(2,3)	SPI ⁽³⁾	I ² C ⁽³⁾	10-bit 1 Msps ADC (Channels)	Comparators	PMP/PSP	JTAG	Trace	Packages ⁽⁴⁾
PIC32MX534F064H	64	64 + 12 ⁽¹⁾	16	1	1	5/5/5	4/4	6	3	4	16	2	Yes	Yes	No	PT, MR
PIC32MX564F064H	64	64 + 12 ⁽¹⁾	32	1	1	5/5/5	4/4	6	3	4	16	2	Yes	Yes	No	PT, MR
PIC32MX564F128H	64	128 + 12 ⁽¹⁾	32	1	1	5/5/5	4/4	6	3	4	16	2	Yes	Yes	No	PT, MR
PIC32MX575F256H	64	256 + 12 ⁽¹⁾	64	1	1	5/5/5	8/4	6	3	4	16	2	Yes	Yes	No	PT, MR
PIC32MX575F512H	64	512 + 12 ⁽¹⁾	64	1	1	5/5/5	8/4	6	3	4	16	2	Yes	Yes	No	PT, MR
PIC32MX534F064L	100	64 + 12 ⁽¹⁾	16	1	1	5/5/5	4/4	6	4	5	16	2	Yes	Yes	Yes	PT, PF, BG
PIC32MX564F064L	100	64 + 12 ⁽¹⁾	32	1	1	5/5/5	4/4	6	4	5	16	2	Yes	Yes	Yes	PT, PF, BG
PIC32MX564F128L	100	128 + 12 ⁽¹⁾	32	1	1	5/5/5	4/4	6	4	5	16	2	Yes	Yes	Yes	PT, PF, BG
PIC32MX575F256L	100	256 + 12 ⁽¹⁾	64	1	1	5/5/5	8/4	6	4	5	16	2	Yes	Yes	Yes	PT, PF, BG
PIC32MX575F512L	100	512 + 12 ⁽¹⁾	64	1	1	5/5/5	8/4	6	4	5	16	2	Yes	Yes	Yes	PT, PF, BG

Legend: PF, PT = TQFP MR = QFN BG = TFBGA TL = VTLA⁽⁵⁾

Note 1: This device features 12 KB boot Flash memory.

Note 2: CTS and RTS pins may not be available for all UART modules. Refer to the “**Device Pin Tables**” section for more information.

Note 3: Some pins between the UART, SPI and I²C modules may be shared. Refer to the “**Device Pin Tables**” section for more information.

Note 4: Refer to **34.0 “Packaging Information”** for more information.

Note 5: 100-pin devices in the VTLA package are available upon request. Please contact your local Microchip Sales Office for details.

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).

PIC32MX5XX/6XX/7XX

REGISTER 5-2: NVMKEY: PROGRAMMING UNLOCK 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	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
	NVMKEY<31:24>							
23:16	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
	NVMKEY<23:16>							
15:8	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
	NVMKEY<15:8>							
7:0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
	NVMKEY<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 **NVMKEY<31:0>**: Unlock Register bits

These bits are write-only, and read as '0' on any read.

Note: This register is used as part of the unlock sequence to prevent inadvertent writes to the PFM.

REGISTER 5-3: NVMADDR: FLASH 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
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	NVMADDR<31:24>							
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	NVMADDR<23:16>							
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
	NVMADDR<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
	NVMADDR<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 **NVMADDR<31:0>**: Flash Address bits

Bulk/Chip/PFM Erase: Address is ignored.

Page Erase: Address identifies the page to erase.

Row Program: Address identifies the row to program.

Word Program: Address identifies the word to program.

PIC32MX5XX/6XX/7XX

REGISTER 10-2: DMASTAT: DMA STATUS 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
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
7:0	U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
	—	—	—	—	RDWR	DMACH<2:0>		

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 31-4 **Unimplemented:** Read as '0'

bit 3 **RDWR:** Read/Write Status bit

1 = Last DMA bus access was a read

0 = Last DMA bus access was a write

bit 2-0 **DMACH<2:0>:** DMA Channel bits

These bits contain the value of the most recent active DMA channel.

REGISTER 10-3: DMAADDR: DMA 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
31:24	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	DMAADDR<31:24>							
23:16	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	DMAADDR<23:16>							
15:8	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	DMAADDR<15:8>							
7:0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	DMAADDR<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 **DMAADDR<31:0>:** DMA Module Address bits

These bits contain the address of the most recent DMA access.

PIC32MX5XX/6XX/7XX

REGISTER 22-1: RTCCON: RTC CONTROL 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	R/W-0	R/W-0
	—	—	—	—	—	—	CAL<9:8>	
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CAL<7:0>							
15:8	R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
	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
	RTSECSEL ⁽³⁾	RTCCLKON	—	—	RTCWREN ⁽⁴⁾	RTCSYNC	HALFSEC ⁽⁵⁾	RTCOE

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 31-26 **Unimplemented:** Read as '0'

bit 25-16 **CAL<9:0>:** RTC Drift Calibration bits, which contain a signed 10-bit integer value

1111111111 = Minimum negative adjustment, subtracts 1 RTC clock pulse every one minute

•

•

•

1000000000 = Maximum negative adjustment, subtracts 512 clock pulses every one minute

0111111111 = Maximum positive adjustment, adds 511 RTC clock pulses every one minute

•

•

•

0000000001 = Minimum positive adjustment, adds 1 RTC clock pulse every one minute

0000000000 = No adjustment

bit 15 **ON:** RTCC On bit^(1,2)

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 CPU enters in Idle mode

0 = Continue normal operation in 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

bit 5-4 **Unimplemented:** Read as '0'

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 only be set 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 only reset on a Power-on Reset (POR).

PIC32MX5XX/6XX/7XX

REGISTER 24-7: C_iRXOVF: CAN RECEIVE FIFO OVERFLOW STATUS 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	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	RXOVF31	RXOVF30	RXOVF29	RXOVF28	RXOVF27	RXOVF26	RXOVF25	RXOVF24
23:16	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	RXOVF23	RXOVF22	RXOVF21	RXOVF20	RXOVF19	RXOVF18	RXOVF17	RXOVF16
15:8	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	RXOVF15	RXOVF14	RXOVF13	RXOVF12	RXOVF11	RXOVF10	RXOVF9	RXOVF8
7:0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	RXOVF7	RXOVF6	RXOVF5	RXOVF4	RXOVF3	RXOVF2	RXOVF1	RXOVF0

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 **RXOVF<31:0>**: FIFO_n Receive Overflow Interrupt Pending bit

1 = FIFO has overflowed
0 = FIFO has not overflowed

REGISTER 24-8: C_iTMR: CAN TIMER 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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CANTS<15:8>							
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CANTS<7:0>							
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
	CANTSPRE<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
	CANTSPRE<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 **CANTS<15:0>**: CAN Time Stamp Timer bits

This is a free-running timer that increments every CANTSPRE system clocks when the CANCAP bit (CiCON<20>) is set.

bit 15-0 **CANTSPRE<15:0>**: CAN Time Stamp Timer Prescaler bits

1111 1111 1111 1111 = CAN time stamp timer (CANTS) increments every 65,535 system clocks

•
•
•

0000 0000 0000 0000 = CAN time stamp timer (CANTS) increments every system clock

Note 1: CiTMR will be paused when CANCAP = 0.

2: The CiTMR prescaler count will be reset on any write to CiTMR (CANTSPRE will be unaffected).

REGISTER 24-15: CifLTCON5: CAN FILTER CONTROL REGISTER 5 (CONTINUED)

- bit 15 **FLTEN21**: Filter 21 Enable bit
 1 = Filter is enabled
 0 = Filter is disabled
- bit 14-13 **MSEL21<1:0>**: Filter 21 Mask Select bits
 11 = Acceptance Mask 3 selected
 10 = Acceptance Mask 2 selected
 01 = Acceptance Mask 1 selected
 00 = Acceptance Mask 0 selected
- bit 12-8 **FSEL21<4:0>**: FIFO Selection bits
 11111 = Message matching filter is stored in FIFO buffer 31
 11110 = Message matching filter is stored in FIFO buffer 30
 •
 •
 •
 00001 = Message matching filter is stored in FIFO buffer 1
 00000 = Message matching filter is stored in FIFO buffer 0
- bit 7 **FLTEN20**: Filter 20 Enable bit
 1 = Filter is enabled
 0 = Filter is disabled
- bit 6-5 **MSEL20<1:0>**: Filter 20 Mask Select bits
 11 = Acceptance Mask 3 selected
 10 = Acceptance Mask 2 selected
 01 = Acceptance Mask 1 selected
 00 = Acceptance Mask 0 selected
- bit 4-0 **FSEL20<4:0>**: FIFO Selection bits
 11111 = Message matching filter is stored in FIFO buffer 31
 11110 = Message matching filter is stored in FIFO buffer 30
 •
 •
 •
 00001 = Message matching filter is stored in FIFO buffer 1
 00000 = Message matching filter is stored in FIFO buffer 0

Note: The bits in this register can only be modified if the corresponding filter enable (FLTENn) bit is '0'.

PIC32MX5XX/6XX/7XX

REGISTER 24-16: CiFLTCON6: CAN FILTER CONTROL REGISTER 6

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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	FLTEN27	MSEL27<1:0>		FSEL27<4:0>				
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	FLTEN26	MSEL26<1:0>		FSEL26<4:0>				
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
	FLTEN25	MSEL25<1:0>		FSEL25<4:0>				
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
	FLTEN24	MSEL24<1:0>		FSEL24<4:0>				

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 **FLTEN27:** Filter 27 Enable bit
1 = Filter is enabled
0 = Filter is disabled
- bit 30-29 **MSEL27<1:0>:** Filter 27 Mask Select bits
11 = Acceptance Mask 3 selected
10 = Acceptance Mask 2 selected
01 = Acceptance Mask 1 selected
00 = Acceptance Mask 0 selected
- bit 28-24 **FSEL27<4:0>:** FIFO Selection bits
11111 = Message matching filter is stored in FIFO buffer 31
11110 = Message matching filter is stored in FIFO buffer 30
.
.
.
00001 = Message matching filter is stored in FIFO buffer 1
00000 = Message matching filter is stored in FIFO buffer 0
- bit 23 **FLTEN26:** Filter 26 Enable bit
1 = Filter is enabled
0 = Filter is disabled
- bit 22-21 **MSEL26<1:0>:** Filter 26 Mask Select bits
11 = Acceptance Mask 3 selected
10 = Acceptance Mask 2 selected
01 = Acceptance Mask 1 selected
00 = Acceptance Mask 0 selected
- bit 20-16 **FSEL26<4:0>:** FIFO Selection bits
11111 = Message matching filter is stored in FIFO buffer 31
11110 = Message matching filter is stored in FIFO buffer 30
.
.
.
00001 = Message matching filter is stored in FIFO buffer 1
00000 = Message matching filter is stored in FIFO buffer 0

Note: The bits in this register can only be modified if the corresponding filter enable (FLTENn) bit is '0'.

PIC32MX5XX/6XX/7XX

REGISTER 24-22: CiFIFOUn: CAN FIFO USER ADDRESS REGISTER 'n' (n = 0 THROUGH 31)

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	R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
	CiFIFOUn<31:24>							
23:16	R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
	CiFIFOUn<23:16>							
15:8	R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
	CiFIFOUn<15:8>							
7:0	R-x	R-x	R-x	R-x	R-x	R-x	R-0 ⁽¹⁾	R-0 ⁽¹⁾
	CiFIFOUn<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 **CiFIFOUn<31:0>**: CAN FIFO User Address bits

TXEN = 1: (FIFO configured as a transmit buffer)

A read of this register will return the address where the next message is to be written (FIFO head).

TXEN = 0: (FIFO configured as a receive buffer)

A read of this register will return the address where the next message is to be read (FIFO tail).

Note 1: This bit will always read '0', which forces byte-alignment of messages.

Note: This register is not guaranteed to read correctly in Configuration mode, and should only be accessed when the module is not in Configuration mode.

REGISTER 24-23: CiFIFOCIN: CAN MODULE MESSAGE INDEX REGISTER 'n' (n = 0 THROUGH 31)

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
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
7:0	U-0	U-0	U-0	R-0	R-0	R-0	R-0	R-0
	—	—	—	CiFIFOCIN<4:0>				

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 31-5 **Unimplemented:** Read as '0'

bit 4-0 **CiFIFOCIN<4:0>**: CAN Side FIFO Message Index bits

TXEN = 1: (FIFO configured as a transmit buffer)

A read of this register will return an index to the message that the FIFO will next attempt to transmit.

TXEN = 0: (FIFO configured as a receive buffer)

A read of this register will return an index to the message that the FIFO will use to save the next message.

PIC32MX5XX/6XX/7XX

REGISTER 25-14: ETHIRQ: ETHERNET CONTROLLER INTERRUPT REQUEST REGISTER

bit 6	<p>PKTPEND: Packet Pending Interrupt bit</p> <p>1 = RX packet pending in memory 0 = RX packet is not pending in memory</p> <p>This bit is set when the BUFCNT counter has a value other than '0'. It is cleared by either a Reset or by writing the BUFCDEC bit to decrement the BUFCNT counter. Writing a '0' or a '1' has no effect.</p>
bit 5	<p>RXACT: Receive Activity Interrupt bit</p> <p>1 = RX packet data was successfully received 0 = No interrupt pending</p> <p>This bit is set whenever RX packet data is stored in the RXBM FIFO. It is cleared by either a Reset or CPU write of a '1' to the CLR register.</p>
bit 4	<p>Unimplemented: Read as '0'</p>
bit 3	<p>TXDONE: Transmit Done Interrupt bit</p> <p>1 = TX packet was successfully sent 0 = No interrupt pending</p> <p>This bit is set when the currently transmitted TX packet completes transmission, and the Transmit Status Vector is loaded into the first descriptor used for the packet. It is cleared by either a Reset or CPU write of a '1' to the CLR register.</p>
bit 2	<p>TXABORT: Transmit Abort Condition Interrupt bit</p> <p>1 = TX abort condition occurred on the last TX packet 0 = No interrupt pending</p> <p>This bit is set when the MAC aborts the transmission of a TX packet for one of the following reasons:</p> <ul style="list-style-type: none">• Jumbo TX packet abort• Underrun abort• Excessive defer abort• Late collision abort• Excessive collisions abort <p>This bit is cleared by either a Reset or CPU write of a '1' to the CLR register.</p>
bit 1	<p>RXBUFNA: Receive Buffer Not Available Interrupt bit</p> <p>1 = RX Buffer Descriptor Not Available condition has occurred 0 = No interrupt pending</p> <p>This bit is set by a RX Buffer Descriptor Overrun condition. It is cleared by either a Reset or a CPU write of a '1' to the CLR register.</p>
bit 0	<p>RXOVFLW: Receive FIFO Over Flow Error bit</p> <p>1 = RX FIFO Overflow Error condition has occurred 0 = No interrupt pending</p> <p>RXOVFLW is set by the RXBM Logic for an RX FIFO Overflow condition. It is cleared by either a Reset or CPU write of a '1' to the CLR register.</p>

Note: 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 only be done for debug/test purposes.

PIC32MX5XX/6XX/7XX

REGISTER 25-26: EMAC1IPGR: ETHERNET CONTROLLER MAC NON-BACK-TO-BACK INTERPACKET GAP 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
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0
	—	NB2BIPKTGP1<6:0>						
7:0	U-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-1	R/W-0
	—	NB2BIPKTGP2<6:0>						

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 31-15 **Unimplemented:** Read as '0'

bit 14-8 **NB2BIPKTGP1<6:0>:** Non-Back-to-Back Interpacket Gap Part 1 bits

This is a programmable field representing the optional carrierSense window referenced in section 4.2.3.2.1 "Deference" of the IEEE 80.23 Specification. If the carrier is detected during the timing of IPGR1, the MAC defers to the carrier. If, however, the carrier comes after IPGR1, the MAC continues timing IPGR2 and transmits, knowingly causing a collision, thus ensuring fair access to the medium. Its range of values is 0x0 to IPGR2. Its recommend value is 0xC (12d).

bit 7 **Unimplemented:** Read as '0'

bit 6-0 **NB2BIPKTGP2<6:0>:** Non-Back-to-Back Interpacket Gap Part 2 bits

This is a programmable field representing the non-back-to-back Inter-Packet-Gap. Its recommended value is 0x12 (18d), which represents the minimum IPG of 0.96 μ s (in 100 Mbps) or 9.6 μ s (in 10 Mbps).

Note: Both 16-bit and 32-bit accesses are allowed to these registers (including the SET, CLR and INV registers). 8-bit accesses are not allowed and are ignored by the hardware.

REGISTER 25-29: EMAC1SUPP: ETHERNET CONTROLLER MAC PHY SUPPORT 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
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	R/W-0	U-0	U-0	R/W-0
	—	—	—	—	RESETRMII ⁽¹⁾	—	—	SPEEDRMII ⁽¹⁾
7:0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-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 31-12 **Unimplemented:** Read as '0'

bit 11 **RESETRMII:** Reset RMII Logic bit⁽¹⁾

1 = Reset the MAC RMII module

0 = Normal operation.

bit 10-9 **Unimplemented:** Read as '0'

bit 8 **SPEEDRMII:** RMII Speed bit⁽¹⁾

This bit configures the Reduced MII logic for the current operating speed.

1 = RMII is running at 100 Mbps

0 = RMII is running at 10 Mbps

bit 7-0 **Unimplemented:** Read as '0'

Note 1: This bit is only used for the RMII module.

Note: Both 16-bit and 32-bit accesses are allowed to these registers (including the SET, CLR and INV registers). 8-bit accesses are not allowed and are ignored by the hardware.

PIC32MX5XX/6XX/7XX

REGISTER 25-37: EMAC1SA0: ETHERNET CONTROLLER MAC STATION ADDRESS 0 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
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P
	STNADDR6<7:0>							
7:0	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P
	STNADDR5<7:0>							

Legend:

R = Readable bit

-n = Value at POR

W = Writable bit

'1' = Bit is set

P = Programmable bit

U = Unimplemented bit, read as '0'

'0' = Bit is cleared

x = Bit is unknown

bit 31-16 **Unimplemented:** Read as '0'

bit 15-8 **STNADDR6<7:0>:** Station Address Octet 6 bits

These bits hold the sixth transmitted octet of the station address.

bit 7-0 **STNADDR5<7:0>:** Station Address Octet 5 bits

These bits hold the fifth transmitted octet of the station address.

- Note 1:** Both 16-bit and 32-bit accesses are allowed to these registers (including the SET, CLR and INV registers). 8-bit accesses are not allowed and are ignored by the hardware.
- 2:** This register is loaded at reset from the factory preprogrammed station address.

REGISTER 26-1: CMxCON: COMPARATOR 'x' CONTROL 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 —
23:16	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —
15:8	R/W-0 ON ⁽¹⁾	R/W-0 COE	R/W-0 CPOL ⁽²⁾	U-0 —	U-0 —	U-0 —	U-0 —	R-0 COUT
7:0	R/W-1 EVPOL<1:0>	R/W-1	U-0 —	R/W-0 CREF	U-0 —	U-0 —	R/W-1	R/W-1 CCH<1:0>

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'

bit 15 **ON:** Comparator ON bit⁽¹⁾

Clearing this bit does not affect the other bits in this register.

1 = Module is enabled. Setting this bit does not affect the other bits in this register

0 = Module is disabled and does not consume current.

bit 14 **COE:** Comparator Output Enable bit

1 = Comparator output is driven on the output CxOUT pin

0 = Comparator output is not driven on the output CxOUT pin

bit 13 **CPOL:** Comparator Output Inversion bit⁽²⁾

1 = Output is inverted

0 = Output is not inverted

bit 12-9 **Unimplemented:** Read as '0'

bit 8 **COUT:** Comparator Output bit

1 = Output of the Comparator is a '1'

0 = Output of the Comparator is a '0'

bit 7-6 **EVPOL<1:0>:** Interrupt Event Polarity Select bits

11 = Comparator interrupt is generated on a low-to-high or high-to-low transition of the comparator output

10 = Comparator interrupt is generated on a high-to-low transition of the comparator output

01 = Comparator interrupt is generated on a low-to-high transition of the comparator output

00 = Comparator interrupt generation is disabled

bit 5 **Unimplemented:** Read as '0'

bit 4 **CREF:** Comparator Positive Input Configure bit

1 = Comparator non-inverting input is connected to the internal CVREF

0 = Comparator non-inverting input is connected to the CxIN+ pin

bit 3-2 **Unimplemented:** Read as '0'

bit 1-0 **CCH<1:0>:** Comparator Negative Input Select bits for Comparator

11 = Comparator inverting input is connected to the IVREF

10 = Comparator inverting input is connected to the C2IN+ pin for C1 and C1IN+ pin for C2

01 = Comparator inverting input is connected to the C1IN+ pin for C1 and C2IN+ pin for C2

00 = Comparator inverting input is connected to the C1IN- pin for C1 and C2IN- pin for C2

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 bit.

2: Setting this bit will invert the signal to the comparator interrupt generator as well. This will result in an interrupt being generated on the opposite edge from the one selected by EVPOL<1:0>.

PIC32MX5XX/6XX/7XX

REGISTER 29-2: DEVCFG1: DEVICE CONFIGURATION WORD 1 (CONTINUED)

- bit 13-12 **FPBDIV<1:0>**: Peripheral Bus Clock Divisor Default Value bits
11 = PBCLK is SYSCLK divided by 8
10 = PBCLK is SYSCLK divided by 4
01 = PBCLK is SYSCLK divided by 2
00 = PBCLK is SYSCLK divided by 1
- bit 11 **Reserved**: Write '1'
- bit 10 **OSCIOFNC**: CLKO Enable Configuration bit
1 = CLKO output is disabled
0 = CLKO output signal is active on the OSCO pin; the Primary Oscillator must be disabled or configured for External Clock mode (EC) for the CLKO to be active (POSCMOD<1:0> = 11 or 00)
- bit 9-8 **POSCMOD<1:0>**: Primary Oscillator Configuration bits
11 = Primary Oscillator is disabled
10 = HS Oscillator mode is selected
01 = XT Oscillator mode is selected
00 = External Clock mode is selected
- bit 7 **IESO**: Internal External Switchover bit
1 = Internal External Switchover mode is enabled (Two-Speed Start-up is enabled)
0 = Internal External Switchover mode is disabled (Two-Speed Start-up is disabled)
- bit 6 **Reserved**: Write '1'
- bit 5 **FSOSCEN**: Secondary Oscillator Enable bit
1 = Enable the Secondary Oscillator
0 = Disable the Secondary Oscillator
- bit 4-3 **Reserved**: Write '1'
- bit 2-0 **FNOSC<2:0>**: Oscillator Selection bits
111 = Fast RC Oscillator with divide-by-N (FRCDIV)
110 = FRCDIV16 Fast RC Oscillator with fixed divide-by-16 postscaler
101 = Low-Power RC Oscillator (LPRC)
100 = Secondary Oscillator (Sosc)
011 = Primary Oscillator (Posc) with PLL module (XT+PLL, HS+PLL, EC+PLL)
010 = Primary Oscillator (XT, HS, EC)⁽¹⁾
001 = Fast RC Oscillator with divide-by-N with PLL module (FRCDIV+PLL)
000 = Fast RC Oscillator (FRC)

Note 1: Do not disable the Posc (POSCMOD = 11) when using this oscillator source.

PIC32MX5XX/6XX/7XX

NOTES:

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

PIC32MX5XX/6XX/7XX

32.2 AC Characteristics and Timing Parameters

The information contained in this section defines PIC32MX5XX/6XX/7XX AC characteristics and timing parameters.

FIGURE 32-1: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS

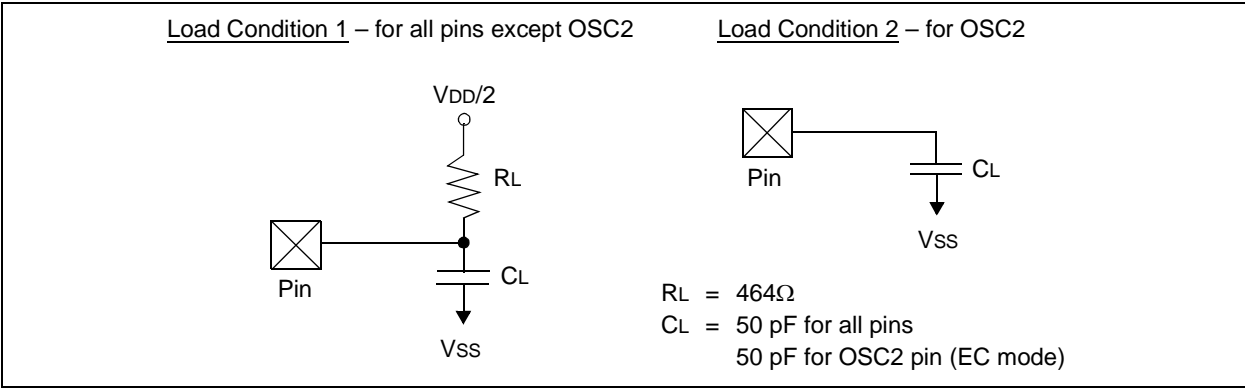


TABLE 32-16: CAPACITIVE LOADING REQUIREMENTS ON OUTPUT PINS

AC CHARACTERISTICS			Standard Operating Conditions: 2.3V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +105^{\circ}\text{C}$ for V-Temp				
Param. No.	Symbol	Characteristics	Min.	Typical ⁽¹⁾	Max.	Units	Conditions
DO50	Cosco	OSC2 pin	—	—	15	pF	In XT and HS modes when an external crystal is used to drive OSC1
DO56	Cio	All I/O pins and OSC2	—	—	50	pF	In EC mode
DO58	Cb	SCLx, SDAx	—	—	400	pF	In I ² C mode

Note 1: Data in "Typical" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

FIGURE 32-2: EXTERNAL CLOCK TIMING

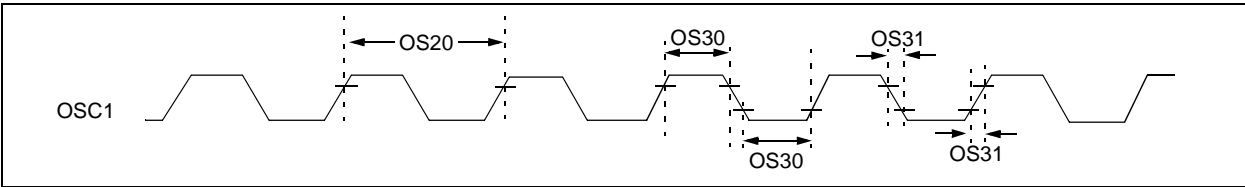


FIGURE 32-14: I2Cx BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)

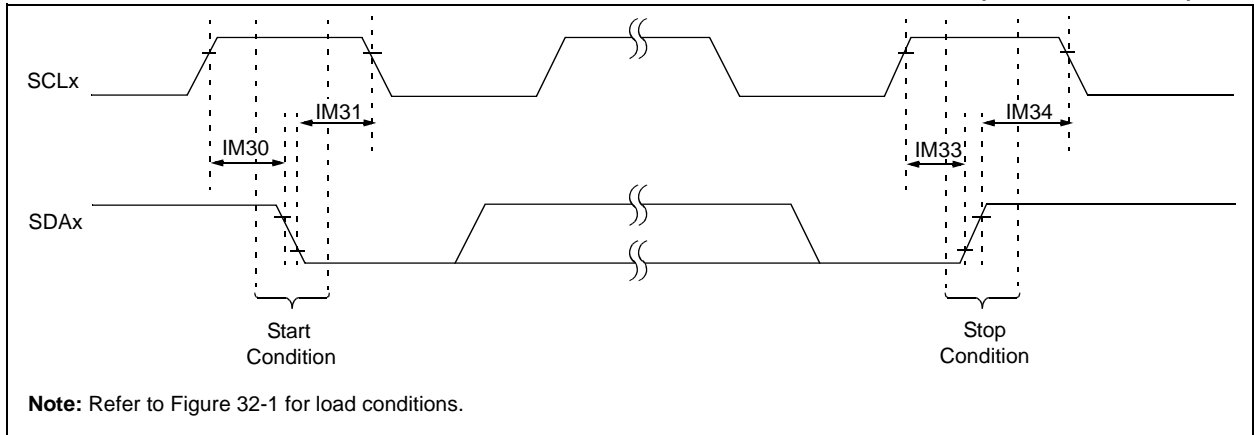


FIGURE 32-15: I2Cx BUS DATA TIMING CHARACTERISTICS (MASTER MODE)

