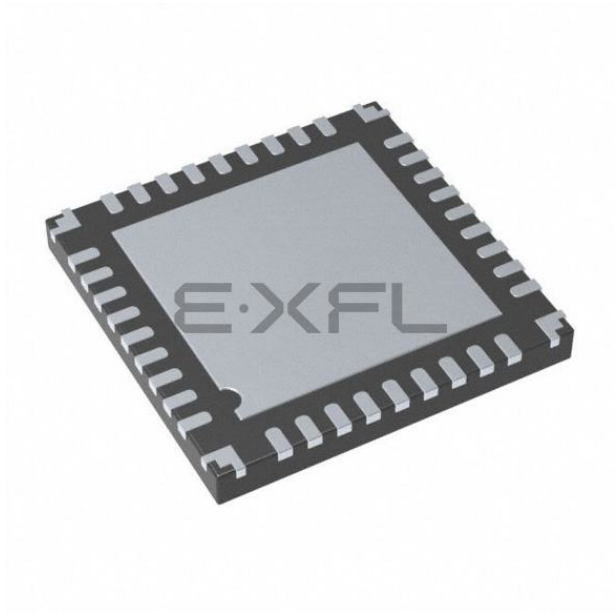
Microchip Technology - DSPIC33CH128MP503-E/M5 Datasheet



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	dsPIC
Core Size	16-Bit Dual-Core
Speed	180MHz, 200MHz
Connectivity	CANbus, I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, Motor Control PWM, POR, PWM, QEI, WDT
Number of I/O	27
Program Memory Size	152KB (152K x 8)
Program Memory Type	FLASH, PRAM
EEPROM Size	-
RAM Size	20K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 31x12b; D/A 4x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	36-UFQFN Exposed Pad
Supplier Device Package	36-UQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33ch128mp503-e-m5

TABLE 3-23: MASTER INTERRUPT VECTOR DETAILS (CONTINUED)

Interrupt Source	Vector #	IRQ #	IVT Address	Interrupt Bit Location		
				Flag	Enable	Priority
QEI1 – QEI Position Counter Compare	56	48	0x000074	IFS3<0>	IEC3<0>	IPC12<2:0>
U1E – UART1 Error	57	49	0x000076	IFS3<1>	IEC3<1>	IPC12<6:4>
U2E – UART2 Error	58	50	0x000078	IFS3<2>	IEC3<2>	IPC12<10:8>
CRC – CRC Generator	59	51	0x00007A	IFS3<3>	IEC3<3>	IPC12<14:12>
C1TX – CAN1 TX Data Request	60	52	0x00007C	IFS3<4>	IEC3<4>	IPC13<2:0>
Reserved	61-68	53-68	0x00007E-0x00008C	—	—	—
ICD – In-Circuit Debugger	69	61	0x00008E	IFS3<13>	IEC3<13>	IPC15<6:4>
JTAG – JTAG Programming	70	62	0x000090	IFS3<14>	IEC3<14>	IPC15<10:8>
PTGSTEP – PTG Step	71	63	0x000092	IFS3<15>	IEC3<15>	IPC15<14:12>
I2C1BC – I2C1 Bus Collision	72	64	0x000094	IFS4<0>	IEC4<0>	IPC16<2:0>
I2C2BC – I2C2 Bus Collision	73	65	0x000096	IFS4<1>	IEC4<1>	IPC16<6:4>
Reserved	74	66	0x000098	—	—	—
PWM1 – PWM Generator 1	75	67	0x00009A	IFS4<3>	IEC4<3>	IPC16<14:12>
PWM2 – PWM Generator 2	76	68	0x00009C	IFS4<4>	IEC4<4>	IPC17<2:0>
PWM3 – PWM Generator 3	77	69	0x00009E	IFS4<5>	IEC4<5>	IPC17<6:4>
PWM4 – PWM Generator 4	78	70	0x0000A0	IFS4<6>	IEC4<6>	IPC17<10:8>
Reserved	79-82	71-74	0x0000A2	—	—	—
CND – Change Notice D	83	75	0x0000AA	IFS4<11>	IEC4<11>	IPC18<14:12>
CNE – Change Notice E	84	76	0x0000AC	IFS4<12>	IEC4<12>	IPC19<2:0>
CMP1 – Comparator 1	85	77	0x0000AE	IFS4<13>	IEC4<13>	IPC19<6:4>
Reserved	86-88	78-80	0x0000B0-0x0000B4	—	—	—
PTGWDT – PTG Watchdog Timer Time-out	89	81	0x0000B6	IFS5<1>	IEC5<1>	IPC20<6:4>
PTG0 – PTG Trigger 0	90	82	0x0000B8	IFS5<2>	IEC5<2>	IPC20<10:8>
PTG1 – PTG Trigger 1	91	83	0x0000BA	IFS5<3>	IEC5<3>	IPC20<14:12>
PTG2 – PTG Trigger 2	92	84	0x0000BC	IFS5<4>	IEC5<4>	IPC21<2:0>
PTG3 – PTG Trigger 3	93	85	0x0000BE	IFS5<5>	IEC5<6>	IPC21<6:4>
SENT1 – SENT1 TX/RX	94	86	0x0000C0	IFS5<6>	IEC5<6>	IPC21<10:8>
SENT1E – SENT1 Error	95	87	0x0000C2	IFS5<7>	IEC5<7>	IPC21<14:12>
SENT2 – SENT2 TX/RX	96	88	0x0000C4	IFS5<8>	IEC5<8>	IPC22<2:0>
SENT2E – SENT2 Error	97	89	0x0000C6	IFS5<9>	IEC5<9>	IPC22<6:4>
ADC – ADC Global Interrupt	98	90	0x0000C8	IFS5<10>	IEC5<10>	IPC22<10:8>
ADCAN0 – ADC AN0 Interrupt	99	91	0x0000CA	IFS5<11>	IEC5<11>	IPC22<14:12>
ADCAN1 – ADC AN1 Interrupt	100	92	0x0000CC	IFS5<12>	IEC5<12>	IPC23<2:0>
ADCAN2 – ADC AN2 Interrupt	101	93	0x0000CE	IFS5<13>	IEC5<13>	IPC23<6:4>
ADCAN3 – ADC AN3 Interrupt	102	94	0x0000D0	IFS5<14>	IEC5<14>	IPC23<10:8>
ADCAN4 – ADC AN4 Interrupt	103	95	0x0000D2	IFS5<15>	IEC5<15>	IPC23<14:12>
ADCAN5 – ADC AN5 Interrupt	104	96	0x0000D4	IFS6<0>	IEC6<0>	IPC24<2:0>
ADCAN6 – ADC AN6 Interrupt	105	97	0x0000D6	IFS6<1>	IEC6<1>	IPC24<6:4>
ADCAN7 – ADC AN7 Interrupt	106	98	0x0000D8	IFS6<2>	IEC6<2>	IPC24<10:8>
ADCAN8 – ADC AN8 Interrupt	107	99	0x0000DA	IFS6<3>	IEC6<3>	IPC24<14:12>
ADCAN9 – ADC AN9 Interrupt	108	100	0x0000DC	IFS6<4>	IEC6<4>	IPC25<2:0>
ADCAN10 – ADC AN10 Interrupt	109	101	0x0000DE	IFS6<5>	IEC6<5>	IPC25<6:4>
ADCAN11 – ADC AN11 Interrupt	110	102	0x0000E0	IFS6<6>	IEC6<6>	IPC25<10:8>
ADCAN12 – ADC AN12 Interrupt	111	103	0x0000E2	IFS6<7>	IEC6<7>	IPC25<14:12>
ADCAN13 – ADC AN13 Interrupt	112	104	0x0000E4	IFS6<8>	IEC6<8>	IPC26<2:0>
ADCAN14 – ADC AN14 Interrupt	113	105	0x0000E6	IFS6<9>	IEC6<9>	IPC26<6:4>
ADCAN15 – ADC AN15 Interrupt	114	106	0x0000E8	IFS6<10>	IEC6<10>	IPC26<10:8>

REGISTER 3-28: CNPUx: CHANGE NOTIFICATION PULL-UP ENABLE FOR PORTx REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNPUx<15:8>							
bit 15							bit 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
CNPUx<7:0>							
bit 7							bit 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 15-0 **CNPUx<15:0>:** Change Notification Pull-up Enable for PORTx bits
1 = The pull-up for PORTx[n] is enabled – takes precedence over the pull-down selection
0 = The pull-up for PORTx[n] is disabled

REGISTER 3-29: CNPDx: CHANGE NOTIFICATION PULL-DOWN ENABLE FOR PORTx REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNPDx<15:8>							
bit 15							bit 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
CNPDx<7:0>							
bit 7							bit 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 15-0 **CNPDx<15:0>:** Change Notification Pull-Down Enable for PORTx bits
1 = The pull-down for PORTx[n] is enabled (if the pull-up for PORTx[n] is not enabled)
0 = The pull-down for PORTx[n] is disabled

TABLE 3-33: OUTPUT SELECTION FOR REMAPPABLE PINS (RPn)

Function	RPnR<5:0>	Output Name
Default PORT	000000	RPn tied to Default Pin
U1TX	000001	RPn tied to UART1 Transmit
$\overline{U1RTS}$	000010	RPn tied to UART1 Request-to-Send
U2TX	000011	RPn tied to UART2 Transmit
$\overline{U2RTS}$	000100	RPn tied to UART2 Request-to-Send
SDO1	000101	RPn tied to SPI1 Data Output
SCK1	000110	RPn tied to SPI1 Clock Output
$\overline{SS1}$	000111	RPn tied to SPI1 Slave Select
SDO2	001000	RPn tied to SPI2 Data Output
SCK2	001001	RPn tied to SPI2 Clock Output
$\overline{SS2}$	001010	RPn tied to SPI2 Slave Select
REFCLKO	001110	RPn tied to Reference Clock Output
OCM1	001111	RPn tied to SCCP1 Output
OCM2	010000	RPn tied to SCCP2 Output
OCM3	010001	RPn tied to SCCP3 Output
OCM4	010010	RPn tied to SCCP4 Output
OCM5	010011	RPn tied to SCCP5 Output
OCM6	010100	RPn tied to SCCP6 Output
CAN1	010101	RPn tied to CAN1 Output
CMP1	010111	RPn tied to Comparator 1 Output
PWM4H	100010	RPn tied to PWM4H Output
PWM4L	100011	RPn tied to PWM4L Output
PWMEA	100100	RPn tied to PWM Event A Output
PWMEB	100101	RPn tied to PWM Event B Output
QEICMP	100110	RPn tied to QEI Comparator Output
CLC1OUT	101000	RPn tied to CLC1 Output
CLC2OUT	101001	RPn tied to CLC2 Output
OCM7	101010	RPn tied to SCCP7 Output
OCM8	101011	RPn tied to SCCP8 Output
PWMEC	101100	RPn tied to PWM Event C Output
PWMED	101101	RPn tied to PWM Event D Output
PTGTRG24	101110	PTG Trigger Output 24
PTGTRG25	101111	PTG Trigger Output 25
SENT1OUT	110000	RPn tied to SENT1 Output
SENT2OUT	110001	RPn tied to SENT2 Output
CLC3OUT	110010	RPn tied to CLC3 Output
CLC4OUT	110011	RPn tied to CLC4 Output
U1DTR	110100	Data Terminal Ready Output 1
U2DTR	110101	Data Terminal Ready Output 2

REGISTER 3-61: RPINR38: PERIPHERAL PIN SELECT INPUT REGISTER 38

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 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
PCI18R7	PCI18R6	PCI18R5	PCI18R4	PCI18R3	PCI18R2	PCI18R1	PCI18R0
bit 7							bit 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 15-8 **Unimplemented:** Read as '0'

bit 7-0 **PCI18R<7:0>:** Assign PWM Input 18 (PCI18) to the Corresponding RPn Pin bits
See Table 3-30.

REGISTER 3-62: RPINR42: PERIPHERAL PIN SELECT INPUT REGISTER 42

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI13R7	PCI13R6	PCI13R5	PCI13R4	PCI13R3	PCI13R2	PCI13R1	PCI13R0
bit 15							bit 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
PCI12R7	PCI12R6	PCI12R5	PCI12R4	PCI12R3	PCI12R2	PCI12R1	PCI12R0
bit 7							bit 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 15-8 **PCI13R<7:0>:** Assign PWM Input 13 (PCI13) to the Corresponding RPn Pin bits
See Table 3-30.

bit 7-0 **PCI12R<7:0>:** Assign PWM Input 12 (PCI12) to the Corresponding RPn Pin bits
See Table 3-30.

REGISTER 3-72: RPOR4: PERIPHERAL PIN SELECT OUTPUT REGISTER 4

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP41R5	RP41R4	RP41R3	RP41R2	RP41R1	RP41R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP40R5	RP40R4	RP40R3	RP40R2	RP40R1	RP40R0
bit 7							bit 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 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP41R<5:0>:** Peripheral Output Function is Assigned to RP41 Output Pin bits (see Table 3-33 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP40R<5:0>:** Peripheral Output Function is Assigned to RP40 Output Pin bits (see Table 3-33 for peripheral function numbers)

REGISTER 3-73: RPOR5: PERIPHERAL PIN SELECT OUTPUT REGISTER 5

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP43R5	RP43R4	RP43R3	RP43R2	RP43R1	RP43R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP42R5	RP42R4	RP42R3	RP42R2	RP42R1	RP42R0
bit 7							bit 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 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP43R<5:0>:** Peripheral Output Function is Assigned to RP43 Output Pin bits (see Table 3-33 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP42R<5:0>:** Peripheral Output Function is Assigned to RP42 Output Pin bits (see Table 3-33 for peripheral function numbers)

REGISTER 3-150: C1BDIAG1L: CAN BUS DIAGNOSTICS REGISTER 1 LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EFMSGCNT<15:8>							
bit 15							bit 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
EFMSGCNT<7:0>							
bit 7							bit 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

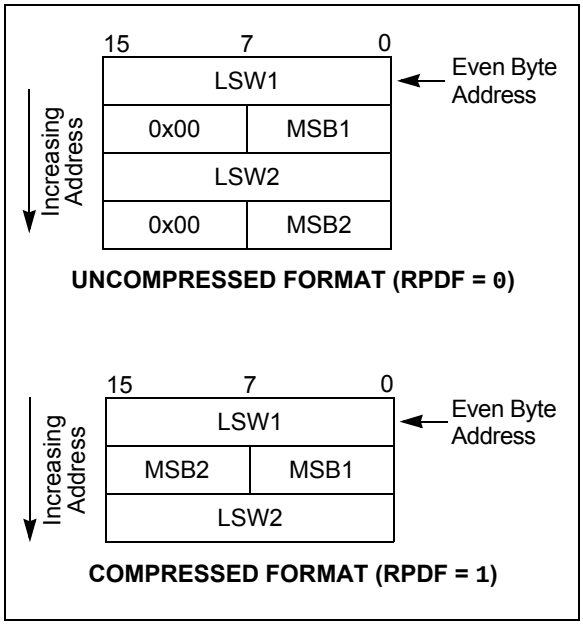
bit 15-0 **EFMSGCNT<15:0>:** Error-Free Message Counter bits

Row programming is performed by first loading 128 instructions into data RAM and then loading the address of the first instruction in that row into the NVMSRCADRL/H register. Once the write has been initiated, the device will automatically load two instructions into the write latches, and write them to the program space destination address defined by the NVMADR/U registers.

The operation will increment the NVMSRCADRL/H and the NVMADR/U registers until all double instruction words have been programmed.

The RPDF bit (NVMCON<9>) selects the format of the stored data in RAM to be either compressed or uncompressed. See Figure 4-14 for data formatting. Compressed data helps to reduce the amount of required RAM by using the upper byte of the second word for the MSB of the second instruction.

FIGURE 4-14: UNCOMPRESSED/ COMPRESSED FORMAT



4.3.3 MASTER TO SLAVE IMAGE LOADING (MSIL)

Master to Slave Image Loading allows the Master user application code to transfer the Slave image stored in the Master Flash to the Slave PRAM. This is the only supported method for programming the Slave PRAM in a final user application.

The LDSLV instruction is executed by the Master user application to transfer a single 24-bit instruction from the Master Flash address, defined by Ws<14:0> (DSRPAG), to the Slave PRAM address, defined by Wd<14:0> (DSWPAG).

The LDSLV instruction should be executed in pairs to ensure correct ECC value generation for each double instruction word that is loaded into the Slave PRAM. The Slave image instruction found at a given even address should be loaded first. This will be the lower instruction word of a 48-bit double instruction word. The upper instruction word should then be loaded from the following odd address. After the pair of LDSLV instructions is executed by the Master user application, both 24-bit Slave image instructions and the generated 7-bit ECC value are actually loaded into the PRAM destination address locations.

The VFSLV instruction allows the Master user application to verify that the PRAM has been loaded correctly. The VFSLV instruction compares the 24-bit instruction word stored in the Master Flash address, defined by Ws<14:0> (DSRPAG), to the 24 bit instruction written to the Slave PRAM address, defined by Wd<14:0> (DSWPAG).

The VFSLV instruction should also be executed in pairs. The lower instruction word found on a given even address should be verified first. The upper instruction word found in the following odd address should then be verified. Then, the Slave image instruction pair read from the Master Flash will have a valid generated ECC value. This full double instruction word with ECC is then compared to the 55-bit value that was actually loaded into the PRAM destination locations. The entire Slave image may be loaded into the PRAM first and then subsequently verified. To make this process simpler, the Microchip `libpic30.h` library has implemented a routine which can be called once to either load or verify the entire Slave image.

The `__program_slave(core#, verify, &slave_image)` routine uses the “verify” parameter to determine if the routine will run using LDSLV instructions or VFSLV instructions. A ‘0’ will load the entire Slave image to the PRAM and a ‘1’ will verify the entire Slave image in the PRAM. An example of how this routine may be used to load and verify the contents of the Slave PRAM is shown in Example 4-2.

TABLE 4-23: SLAVE INTERRUPT PRIORITY REGISTERS (CONTINUED)

Register	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPC36	—	—	—	—	—	—	—	—	—	MSTBRKIP2	MSTBRKIP1	MSTBRKIP0	—	—	—	—
IPC37	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
IPC38	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
IPC39	—	—	—	—	—	—	—	—	—	—	—	—	—	MCLKFIP2	MCLKFIP1	MCLKFIP0
IPC40	—	—	—	—	—	—	—	—	—	ADC1IP2	ADC1IP1	ADC1IP0	—	ADC0IP2	ADC0IP1	ADC0IP0
IPC41	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
IPC42	—	PEVTCIP2	PEVTCIP1	PEVTCIP0	—	PEVTBIP2	PEVTBIP1	PEVTBIP0	—	PEVTAIP2	PEVTAIP1	PEVTAIP0	—	ADFIFOIP2	ADFIFOIP1	ADFIFOIP0
IPC43	—	CLC3PIP2	CLC3PIP1	CLC3PIP0	—	PEVTFIP2	PEVTFIP1	PEVTFIP0	—	PEVTEIP2	PEVTEIP1	PEVTEIP0	—	PEVTDIP2	PEVTDIP1	PEVTDIP0
IPC44	—	CLC3NIP2	CLC3NIP1	CLC3NIP0	—	CLC2NIP2	CLC2NIP1	CLC2NIP0	—	CLC1NIP2	CLC1NIP1	CLC1NIP0	—	CLC4PIP2	CLC4PIP1	CLC4PIP0
IPC45	—	—	—	—	—	—	—	—	—	—	—	—	—	CLC4NIP2	CLC4NIP1	CLC4NIP0
IPC46	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
IPC47	—	—	—	—	—	—	—	—	—	U1EVTIP2	U1EVTIP1	U1EVTIP0	—	—	—	—

REGISTER 4-20: INTCON3: SLAVE INTERRUPT CONTROL REGISTER 3

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	NAE
bit 15							bit 8

U-0	U-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0
—	—	—	DOOVR	—	—	—	APLL
bit 7							bit 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 15-9 **Unimplemented:** Read as '0'

bit 8 **NAE:** NVM Address Error Soft Trap Status bit
1 = NVM address error soft trap has occurred
0 = NVM address error soft trap has not occurred

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **DOOVR:** DO Stack Overflow Soft Trap Status bit
1 = DO stack overflow soft trap has occurred
0 = DO stack overflow soft trap has not occurred

bit 3-1 **Unimplemented:** Read as '0'

bit 0 **APLL:** Auxiliary PLL Loss of Lock Soft Trap Status bit
1 = APLL lock soft trap has occurred
0 = APLL lock soft trap has not occurred

REGISTER 4-21: INTCON4: SLAVE INTERRUPT CONTROL REGISTER 4

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	SGHT
bit 7							bit 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 15-1 **Unimplemented:** Read as '0'

bit 0 **SGHT:** Software Generated Hard Trap Status bit
1 = Software generated hard trap has occurred
0 = Software generated hard trap has not occurred

REGISTER 4-62: RPOR2: PERIPHERAL PIN SELECT OUTPUT REGISTER 2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP37R5	RP37R4	RP37R3	RP37R2	RP37R1	RP37R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP36R5	RP36R4	RP36R3	RP36R2	RP36R1	RP36R0
bit 7							bit 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 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP37R<5:0>:** Peripheral Output Function is Assigned to S1RP37 Output Pin bits (see Table 4-31 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP36R<5:0>:** Peripheral Output Function is Assigned to S1RP36 Output Pin bits (see Table 4-31 for peripheral function numbers)

REGISTER 4-63: RPOR3: PERIPHERAL PIN SELECT OUTPUT REGISTER 3

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP39R5	RP39R4	RP39R3	RP39R2	RP39R1	RP39R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP38R5	RP38R4	RP38R3	RP38R2	RP38R1	RP38R0
bit 7							bit 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 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP39R<5:0>:** Peripheral Output Function is Assigned to S1RP39 Output Pin bits (see Table 4-31 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP38R<5:0>:** Peripheral Output Function is Assigned to S1RP38 Output Pin bits (see Table 4-31 for peripheral function numbers)

REGISTER 5-5: MSI1MBXnD: MSI1 MASTER MAILBOX n DATA REGISTER (n = 0 to 15)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MSIMBXnD<15:8>							
bit 15							bit 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
MSIMBXnD<7:0>							
bit 7							bit 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 15-0 **MSIMBXnD<15:0>:** MSI1 Mailbox n Data bits

When Configuration bit, MBXMx = 1 (programmed):
Mailbox Data Direction: Master read, Slave write; Master MSIMBXnD<15:0> bits become R-0 (a Master write to MSIMBXnD<15:0> will have no effect).

When Configuration bit, MBXMx = 0 (programmed):
Mailbox Data Direction: Master write, Slave read; Master MSIMBXnD<15:0> bits become R/W-0.

REGISTER 6-2: CLKDIV: CLOCK DIVIDER REGISTER (MASTER) (CONTINUED)

bit 3-0 **PLLPRE<3:0>:** PLL Phase Detector Input Divider Select bits (also denoted as 'N1', PLL prescaler)(4)
11111 = Reserved
...
1001 = Reserved
1000 = Input divided by 8
0111 = Input divided by 7
0110 = Input divided by 6
0101 = Input divided by 5
0100 = Input divided by 4
0011 = Input divided by 3
0010 = Input divided by 2
0001 = Input divided by 1 (power-on default selection)
0000 = Reserved

Note 1: The DOZE<2:0> bits can only be written to when the DOZEN bit is clear. If DOZEN = 1, any writes to DOZE<2:0> are ignored.

2: This bit is cleared when the ROI bit is set and an interrupt occurs.

3: The DOZEN bit cannot be set if DOZE<2:0> = 000. If DOZE<2:0> = 000, any attempt by user software to set the DOZEN bit is ignored.

4: PLLPRE<3:0> may be updated while the PLL is operating, but the VCO may overshoot.

REGISTER 9-7: CMBTRIGL: COMBINATIONAL TRIGGER REGISTER LOW

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 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
CTA8EN	CTA7EN	CTA6EN	CTA5EN	CTA4EN	CTA3EN	CTA2EN	CTA1EN
bit 7							bit 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 15-8 **Unimplemented:** Read as '0'

bit 7 **CTA8EN:** Enable Trigger Output from PWM Generator #8 as Source for Combinational Trigger A bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger A signal
0 = Disabled

bit 6 **CTA7EN:** Enable Trigger Output from PWM Generator #7 as Source for Combinational Trigger A bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger A signal
0 = Disabled

bit 5 **CTA6EN:** Enable Trigger Output from PWM Generator #6 as Source for Combinational Trigger A bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger A signal
0 = Disabled

bit 4 **CTA5EN:** Enable Trigger Output from PWM Generator #5 as Source for Combinational Trigger A bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger A signal
0 = Disabled

bit 3 **CTA4EN:** Enable Trigger Output from PWM Generator #4 as Source for Combinational Trigger A bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger A signal
0 = Disabled

bit 2 **CTA3EN:** Enable Trigger Output from PWM Generator #3 as Source for Combinational Trigger A bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger A signal
0 = Disabled

bit 1 **CTA2EN:** Enable Trigger Output from PWM Generator #2 as Source for Combinational Trigger A bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger A signal
0 = Disabled

bit 0 **CTA1EN:** Enable Trigger Output from PWM Generator #1 as Source for Combinational Trigger A bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger A signal
0 = Disabled

REGISTER 9-17: PGxyPCIL: PWM GENERATOR xy PCI REGISTER LOW (x = PWM GENERATOR #; y = F, CL, FF OR S) (CONTINUED)

For Slave:
PWM_PCI<n> Source
00111 = Reserved
00110 = Reserved
00101 = Reserved
00100 = Reserved
00011 = Internally connected to Combo Trigger B
00010 = Internally connected to Combo Trigger A
00001 = Internally connected to the output of PWMPCI<2:0> MUX
00000 = Internally connect to '1'b0'
11111 = Slave CLC1
11110 = Slave Comparator Output 3
11101 = Slave Comparator Output 2
11100 = Slave Comparator Output 1
11011 = Master Comparator Output 1
11010 = Master PWM Event F
11001 = Master PWM Event E
11000 = Master PWM Event D
10111 = Master PWM Event C
10110 = PCI<22> device pin device none PCI<22>
10101 = PCI<21> device pin device none PCI<21>
10100 = PCI<20> device pin device none PCI<20>
10011 = Device pin device none PCI<19>
10010 = Slave S1RPn input Slave PCI18R
10001 = Slave S1RPn input Slave PCI17R
10000 = Slave S1RPn input Slave PCI16R
01111 = Slave S1RPn input Slave PCI15R
01110 = Slave S1RPn input Slave PCI14R
01101 = Slave S1RPn input Slave PCI13R
01100 = Slave S1RPn input Slave PCI12R
01011 = Slave S1RPn input Slave PCI11R
01010 = Slave S1RPn input Slave PCI10R
01001 = Slave S1RPn input Slave PCI9R
01000 = Slave S1RPn input Slave PCI8R

10.4 Input Capture Mode

Input Capture mode is used to capture a timer value from an independent timer base, upon an event, on an input pin or other internal trigger source. The input capture features are useful in applications requiring frequency (time period) and pulse measurement. Figure 10-6 depicts a simplified block diagram of Input Capture mode.

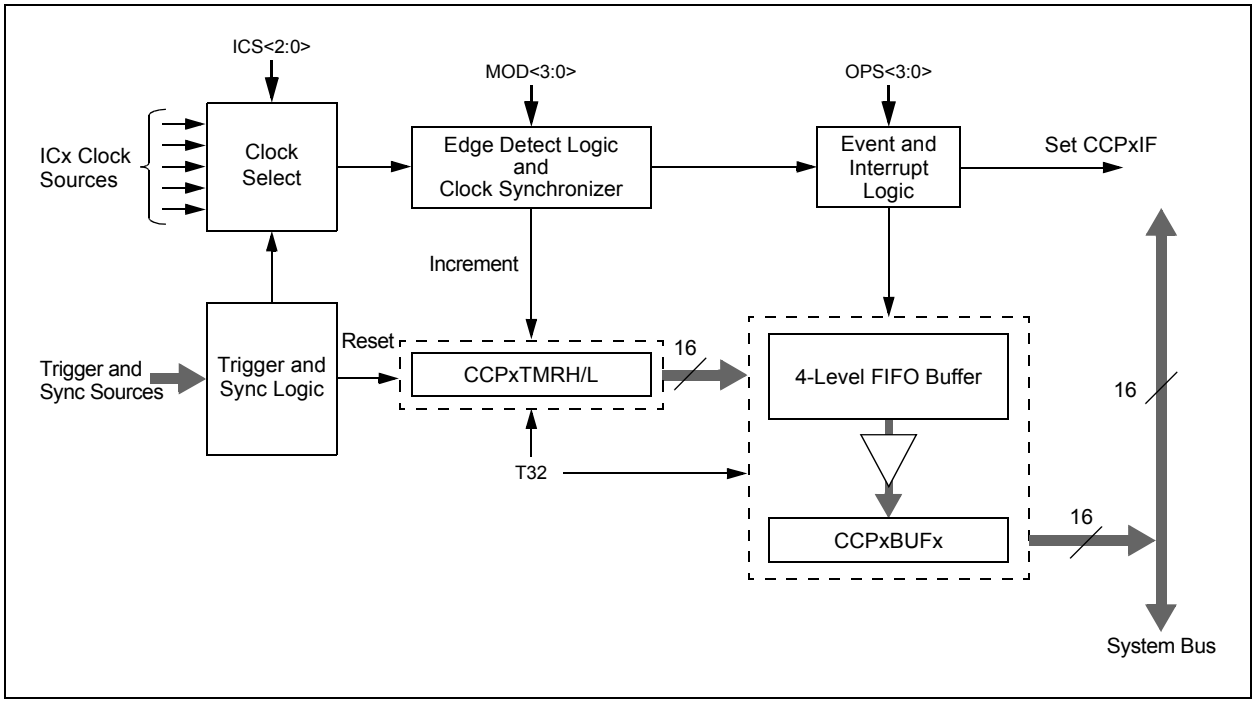
Input Capture mode uses a dedicated 16/32-bit, synchronous, up counting timer for the capture function. The timer value is written to the FIFO when a capture event occurs. The internal value may be read (with a synchronization delay) using the CCPxTMRH/L register.

To use Input Capture mode, the CCSEL bit (CCPxCON1L<4>) must be set. The T32 and the MOD<3:0> bits are used to select the proper Capture mode, as shown in Table 10-4.

TABLE 10-4: INPUT CAPTURE x MODES

MOD<3:0> (CCPxCON1L<3:0>)	T32 (CCPxCON1L<5>)	Operating Mode
0000	0	Edge Detect (16-bit capture)
0000	1	Edge Detect (32-bit capture)
0001	0	Every Rising (16-bit capture)
0001	1	Every Rising (32-bit capture)
0010	0	Every Falling (16-bit capture)
0010	1	Every Falling (32-bit capture)
0011	0	Every Rising/Falling (16-bit capture)
0011	1	Every Rising/Falling (32-bit capture)
0100	0	Every 4th Rising (16-bit capture)
0100	1	Every 4th Rising (32-bit capture)
0101	0	Every 16th Rising (16-bit capture)
0101	1	Every 16th Rising (32-bit capture)

FIGURE 10-6: INPUT CAPTURE x BLOCK DIAGRAM



11.2 Features Overview

- Four Rail-to-Rail Analog Comparators
- Up to Five Selectable Input Sources per Comparator:
  - Three external inputs
  - Two internal inputs from PGA module
- Programmable Comparator Hysteresis
- Programmable Output Polarity
- Interrupt Generation Capability
- Dedicated Pulse Density Modulation DAC for each Analog Comparator:
  - PDM unit followed by a digitally controlled multimode multipole RC filter
- Multimode Multipole RC Output Filter:
  - Transition mode: Provides the fastest response
  - Fast mode: For tracking DAC slopes
  - Steady-State mode: Provides 12-bit resolution
- Slope Compensation along with each DAC:
  - Slope Generation mode
  - Hysteretic Control mode
  - Triangle Wave mode
- Functional Support for the High-Speed PWM module which Includes:
  - PWM duty cycle control
  - PWM period control
  - PWM Fault detect

11.3 DAC Control Registers

The DACCTRL1L and DACCTRL2H/L registers are common configuration registers for Master and Slave DAC modules. The Master and Slave DAC modules are controlled by separate sets of DACCTRL1/2 registers. The DACxCON, DACxDAT, SLPxCON and SLPxDAT registers specify the operation of individual modules. Note that x = 1 for the Master module and x = 1-3 for the Slave modules.

NOTES:

REGISTER 21-13: FDMT CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

U-1	U-1	U-1	U-1	U-1	U-1	U-1	R/PO-1
—	—	—	—	—	—	—	DMTDIS
bit 7							bit 0

Legend:	PO = Program Once 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 23-1 **Unimplemented:** Read as ‘1’

bit 0 **DMTDIS:** DMT Disable bit

1 = DMT is disabled

0 = DMT is enabled

TABLE 24-29: INTERNAL FRC ACCURACY

AC CHARACTERISTICS		Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Characteristic**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
Internal FRC Accuracy @ FRC Frequency = 8 MHz(1)						
F20a	FRC	-3	—	+3	%	-40°C ≤ TA ≤ 0°C
		-1.5	—	+1.5	%	0°C ≤ TA ≤ +85°C
F20b	FRC	-2	—	+2	%	+85°C ≤ TA ≤ +125°C
F22	BFRC	-17	—	+17	%	-40°C ≤ TA ≤ +125°C

Note 1: Frequency is calibrated at +25°C and 3.3V. TUNx bits can be used to compensate for temperature drift.

TABLE 24-30: INTERNAL LPRC ACCURACY

AC CHARACTERISTICS		Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Param No.	**Characteristic**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**	
LPRC @ 32.768 kHz							
F21a	LPRC	-30	—	+30	%	-40°C ≤ TA ≤ -10°C	VDD = 3.0-3.6V
		-20	—	+20	%	-10°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
F21b	LPRC	-30	—	+30	%	+85°C ≤ TA ≤ +125°C	VDD = 3.0-3.6V

T

U

V

W