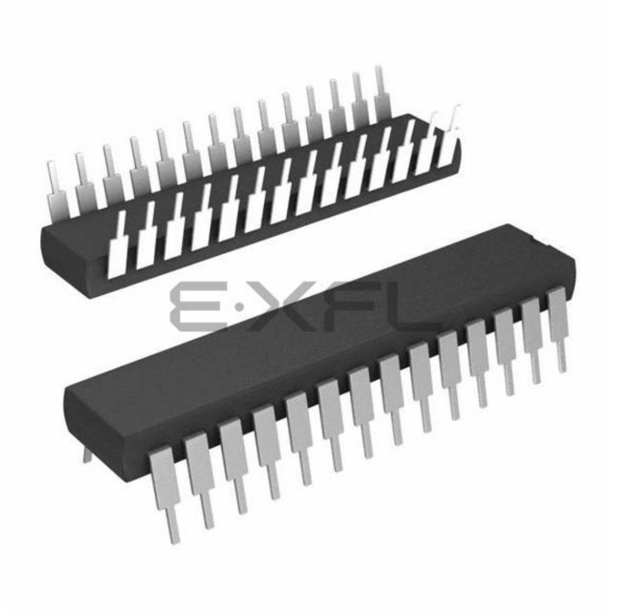


Welcome to E-XFL.COM

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	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	CANbus, I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	25
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf26k83-i-sp

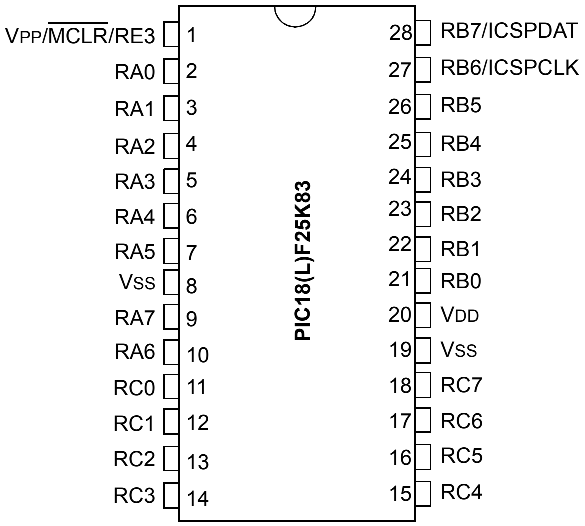
TABLE 2: PACKAGES

Device	SPDIP	SOIC	SSOP	UQFN	QFN
PIC18(L)F25K83	●	●	●	●	●
PIC18(L)F26K83	●	●	●	●	●

Note 1: Pin details are subject to change.

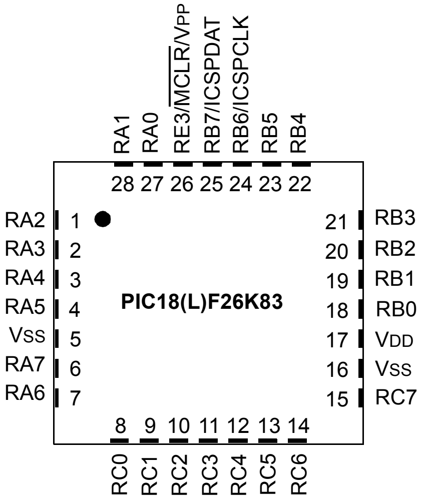
Pin Diagrams

28-pin SPDIP, SOIC, SSOP



Note: See Table 3 for location of all peripheral functions.

28-pin QFN (6x6x0.9mm), UQFN (4x4x0.5mm)



Note 1: See Table 3 for location of all peripheral functions.

2: It is recommended that the exposed bottom pad be connected to VSS, however it must not be the only VSS connection to the device.

5.7 Device Information Area

The Device Information Area (DIA) is a dedicated region in the program memory space. The DIA contains the calibration data for the internal temperature indicator module, stores the Microchip Unique Identifier words and the Fixed Voltage Reference voltage readings measured in mV.

The complete DIA table is shown in Table 5-3: Device Information Area, followed by a description of each region and its functionality. The data is mapped from 3F0000h to 3F003Fh in the PIC18(L)F25/26K83 family. These locations are read-only and cannot be erased or modified by the user. The data is programmed into the device during manufacturing.

TABLE 5-3: DEVICE INFORMATION AREA

Address Range	Name of Region	Standard Device Information
3F0000h-3F000Bh	MUI0	Microchip Unique Identifier (6 Words)
	MUI1	
	MUI2	
	MUI3	
	MUI4	
	MUI5	
3F000Ch-3F000Fh	MUI6	Unassigned (2 Words)
	MUI7	
3F0010h-3F0023h	EUI0	Optional External Unique Identifier (10 Words)
	EUI1	
	EUI2	
	EUI3	
	EUI4	
	EUI5	
	EUI6	
	EUI7	
	EUI8	
	EUI9	
3F0024h-3F0025h	TSLR1	Unassigned (1 Word)
3F0026h-3F0027h	TSLR2	Temperature Indicator ADC reading at @ 90°C (low range setting)
3F0028h-3F0029h	TSLR3	Unassigned (1 word)
3F002Ah-3F002Bh	TSHR1	Unassigned (1 Word)
3F002Ch-3F002Dh	TSHR2	Temperature Indicator ADC reading at @ 90°C (high range setting)
3F002Eh-3F002Fh	TSHR3	Unassigned (1 Word)
3F0030h-3F0031h	FVRA1X	ADC FVR1 Output voltage for 1x setting (in mV)
3F0032h-3F0033h	FVRA2X	ADC FVR1 Output Voltage for 2x setting (in mV)
3F0034h-3F0035h	FVRA4X	ADC FVR1 Output Voltage for 4x setting (in mV)
3F0036h-3F0037h	FVRC1X	Comparator FVR2 output voltage for 1x setting (in mV)
3F0038h-3F0039h	FVRC2X	Comparator FVR2 output voltage for 2x setting (in mV)
3F003Ah-3F003Bh	FVRC4X	Comparator FVR2 output voltage for 4x setting (in mV)
3F003Ch-3F003Fh		Unassigned (2 Words)

7.2.2.3 Internal Oscillator Frequency Adjustment

The internal oscillator is factory-calibrated. This internal oscillator can be adjusted in software by writing to the OSCTUNE register (Register 7-3).

The default value of the OSCTUNE register is 00h. The value is a 6-bit two's complement number. A value of 1Fh will provide an adjustment to the maximum frequency. A value of 20h will provide an adjustment to the minimum frequency.

When the OSCTUNE register is modified, the oscillator frequency will begin shifting to the new frequency. Code execution continues during this shift. There is no indication that the shift has occurred.

OSCTUNE **does not affect** the LFINTOSC frequency. Operation of features that depend on the LFINTOSC clock source frequency, such as the Power-up Timer (PWRT), WWDT, Fail-Safe Clock Monitor (FSCM) and peripherals, are *not* affected by the change in frequency.

7.2.2.4 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is a factory-calibrated 31 kHz internal clock source.

The LFINTOSC is the frequency for the Power-up Timer (PWRT), Windowed Watchdog Timer (WWDT) and Fail-Safe Clock Monitor (FSCM).

The LFINTOSC is enabled through one of the following methods:

- Programming the RSTOSC<2:0> bits of Configuration Word 1 to enable LFINTOSC.
- Write to the NOSC<2:0> bits of the OSCCON1 register during run-time. See **Section 7.3, Clock Switching** for more information.

7.2.2.5 ADCRC

The ADCRC is an oscillator dedicated to the ADC^2 module. The ADCRC oscillator can be manually enabled using the ADOEN bit of the OSCEN register. The ADCRC runs at a fixed frequency of 600 kHz. ADCRC is automatically enabled if it is selected as the clock source for the ADC^2 module.

13.1.1 TABLE READS AND TABLE WRITES

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (`TBLRD`)
- Table Write (`TBLWT`)

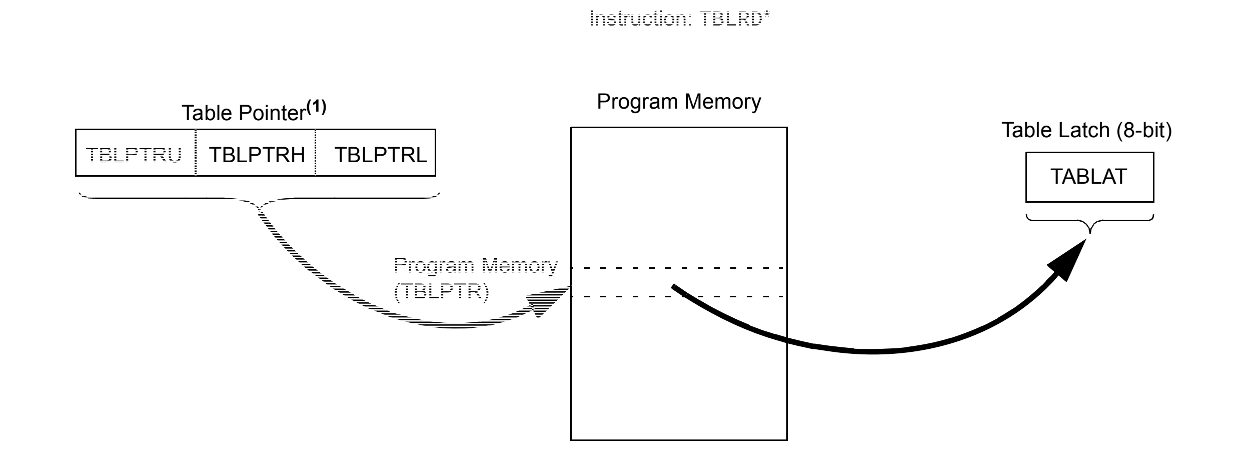
The program memory space is 16 bits wide, while the data RAM space is eight bits wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

The table read operation retrieves one byte of data directly from program memory and places it into the TABLAT register. Figure 13-1 shows the operation of a table read.

The table write operation stores one byte of data from the TABLAT register into a write block holding register. The procedure to write the contents of the holding registers into program memory is detailed in **Section 13.1.6 "Writing to Program Flash Memory"**. Figure 13-2 shows the operation of a table write with program memory and data RAM.

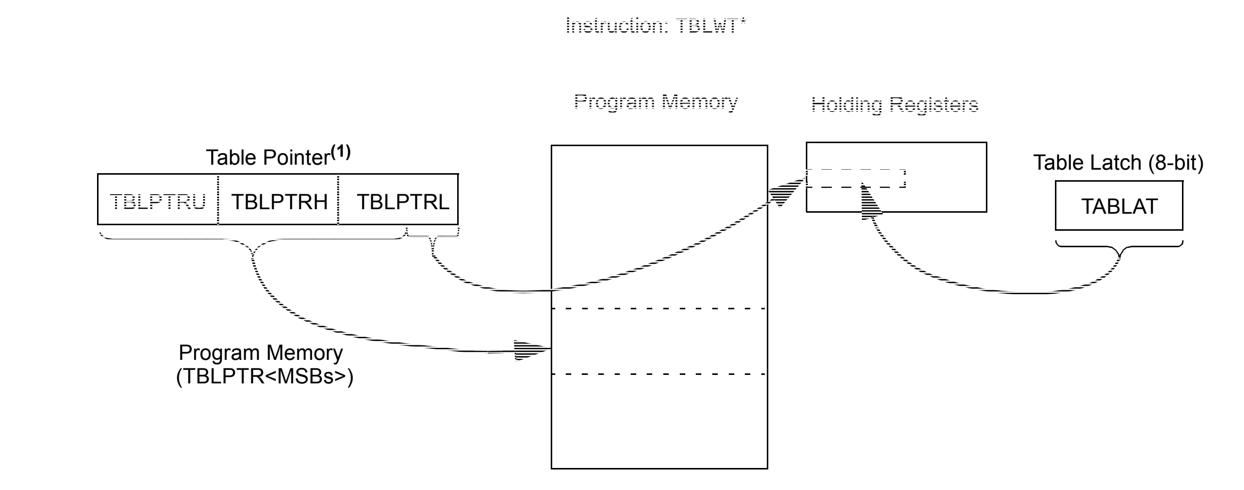
Table operations work with byte entities. Tables containing data, rather than program instructions, are not required to be word aligned. Therefore, a table can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word aligned.

FIGURE 13-1: TABLE READ OPERATION



Note 1: Table Pointer register points to a byte in program memory.

FIGURE 13-2: TABLE WRITE OPERATION



Note 1: During table writes the Table Pointer does not point directly to program memory. The LSBs of TBLPRTL actually point to an address within the write block holding registers. The MSBs of the Table Pointer determine where the write block will eventually be written. The process for writing the holding registers to the program memory array is discussed in **Section 13.1.6 "Writing to Program Flash Memory"**.

14.5 CRC Check Value

The CRC check value will be located in the CRCACC registers after the CRC calculation has finished. The check value will depend on two mode settings of the CRCCON0 register: ACCM and SHIFTM. When the ACCM bit is set, the CRC module augments the data with a number of zeros equal to the length of the polynomial to align the final check value. When the ACCM bit is not set, the CRC will stop at the end of the data. A number of zeros equal to the length of the polynomial can then be entered into CRCDAT to find the same check value as augmented mode. Alternatively the expected check value can be entered at this point to make the final result equal '0'.

When the CRC check value is computed with the SHIFTM bit set, selecting LSb first, and the ACCM bit is also set then the final value in the CRCACC registers will be reversed such that the LSb will be in the MSb position and vice versa. This is the expected check value in bit reversed form. If you are creating a check value to be appended to a data stream then a bit reversal must be performed on the final value to achieve the correct checksum. You can use the CRC to do this reversal by the following method:

- Save the CRCACC value in user RAM space
- Clear the CRCACC registers
- Clear the CRCXOR registers
- Write the saved CRCACC value to the CRCDAT input.

The properly oriented check value will be in the CRCACC registers as the result.

14.6 CRC Interrupt

The CRC will generate an interrupt when the BUSY bit transitions from 1 to 0. The CRCIF Interrupt Flag is set every time the BUSY bit transitions, regardless of whether or not the CRC interrupt is enabled. The CRCIF bit can only be cleared in software.

14.7 Configuring the CRC

The following steps illustrate how to properly configure the CRC.

1. Determine if the automatic program memory scan will be used with the scanner or manual calculation through the SFR interface and perform the actions specified in **Section 14.4 "CRC Data Sources"**, depending on which decision was made.
2. If desired, seed a starting CRC value into the CRCACCH/L registers.
3. Program the CRCXORH/L registers with the desired generator polynomial.
4. Program the DLEN<3:0> bits of the CRCCON1 register with the length of the data word - 1 (refer to Example 14-1). This determines how many times the shifter will shift into the accumulator for each data word.
5. Program the PLEN<3:0> bits of the CRCCON1 register with the length of the polynomial -2 (refer to Example 14-1).
6. Determine whether shifting in trailing zeros is desired and set the ACCM bit of the CRCCON0 register appropriately.
7. Likewise, determine whether the MSb or LSb should be shifted first and write the SHIFTM bit of the CRCCON0 register appropriately.
8. Write the GO bit of the CRCCON0 register to begin the shifting process.

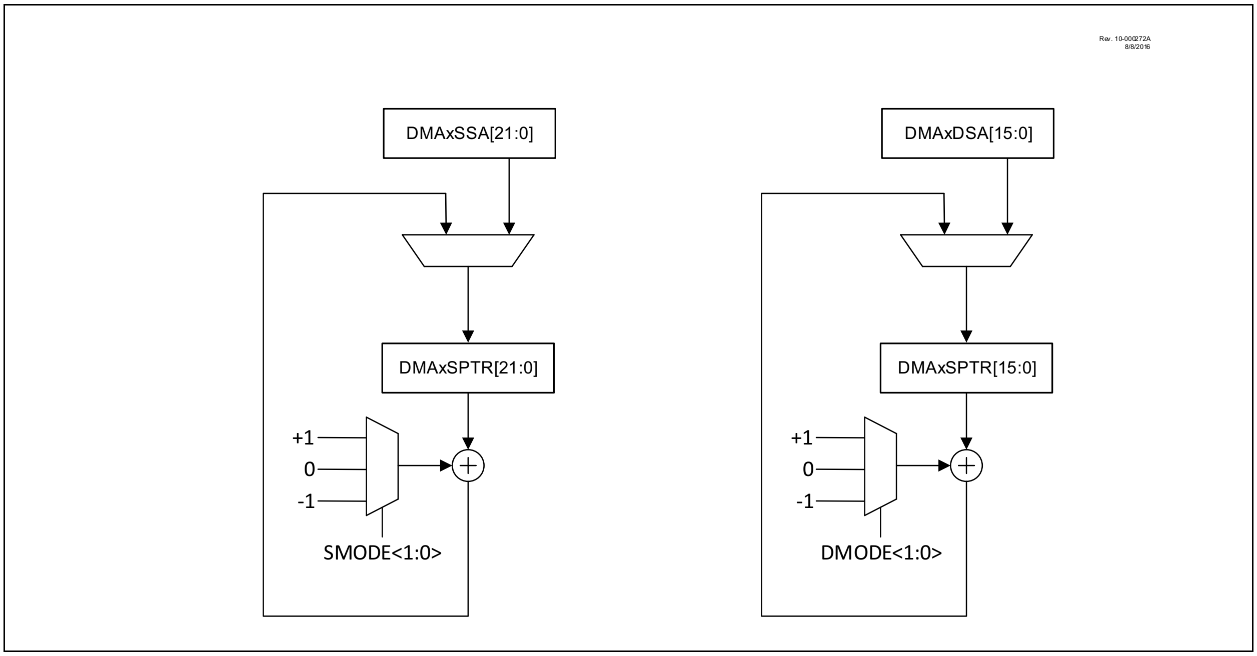
9a. If manual SFR entry is used, monitor the FULL bit of the CRCCON0 register. When FULL = 0, another word of data can be written to the CRCDATH/L registers, keeping in mind that CRCDATH should be written first if the data has more than eight bits, as the shifter will begin upon the CRCDATL register being written.

9b. If the scanner is used, the scanner will automatically load words into the CRCDATH/L registers as needed, as long as the GO bit is set.

10a. If manual entry is used, monitor the CRCIF (and BUSY bit to determine when the completed CRC calculation can be read from CRCACCH/L registers.

10b. If using the memory scanner, monitor the SCANIF (or the GO bit) for the scanner to finish pushing information into the CRCDAT registers. After the scanner is completed, monitor the BUSY bit to determine that the CRC has been completed and the check value can be read from the CRCACC registers. If both the interrupt flags are set and the BUSY and GO bits are cleared, the completed CRC calculation can be read from the CRCACCH/L registers.

FIGURE 15-2: DMA POINTERS BLOCK DIAGRAM



The DMA can initiate data transfers from the PFM, Data EEPROM or SFR/GPR Space. The SMR<1:0> bits in the DMAxCON1 register are used to select the type of memory being pointed to by the Source Address Pointer. The SMR<1.0> bits are required because the PFM and SFR/GPR spaces have overlapping addresses that do not allow the specified address to uniquely define the memory location to be accessed.

> **Note 1:** For proper memory read access to occur, the combination of address and space selection must be valid.
>
> **2:** The destination does not have space selection bits because it can only write to the SFR/GPR space.

15.4.2 DMA MESSAGE SIZE/COUNTERS

A transaction is the transfer of one byte. A message consists of one or more transactions. A complete DMA process consists of one or more messages. The size registers determine how many transactions are in a message. The DMAxSSZ registers determine the source size and DMAxDSZ registers determine the destination size.

When a DMA transfer is initiated, the size registers are copied to corresponding counter registers that control the duration of the message. The DMAxSCNT registers count the source transactions and the DMAxDCNT registers count the destination transactions. Both are simultaneously decremented by one after each transaction.

A message is started by setting the DGO bit of the DMAxCON0 register and terminates when the smaller of the two counters reaches zero.

When either counter reaches zero the DGO bit is cleared and the counter and pointer registers are immediately reloaded with the corresponding size and address data. If the other counter did not reach zero then the next message will continue with the count and address corresponding to that register.

When the source and destination size registers are not equal then the ratio of the largest to the smallest size determines how many messages are in the DMA process. For example, when the destination size is 6 and the source size is 2 then each message will consist of two transactions and the complete DMA process will consist of three messages. When the larger size is not an even integer of the smaller size then the last message in the process will terminate early when the larger count reaches zero. In that case, the larger counter will reset and the smaller counter will have a remainder skewing any subsequent messages by that amount.

> **Note:** Reading the DMAxSCNT or DMAxDCNT registers will never return zero. When either register is decremented from ‘1’ it is immediately reloaded from the corresponding size register.

21.7 Timer1/3/5 Interrupt

The Timer1/3/5 register pair (TMRxH:TMRxL) increments to FFFFh and rolls over to 0000h. When Timer1/3/5 rolls over, the Timer1/3/5 interrupt flag bit of the respective PIR register is set. To enable the interrupt-on-rollover, you must set these bits:

- ON bit of the TxCON register
- TMRxIE bits of the respective PIE register
- GIE/GIEH bit of the INTCON0 register

The interrupt is cleared by clearing the TMRxIF bit in the Interrupt Service Routine.

For more information on selecting high or low priority status for the Timer1/3/5 Overflow Interrupt, see **Section 9.0 “Interrupt Controller”**.

> **Note:** The TMRxH:TMRxL register pair and the TMRxIF bit should be cleared before enabling interrupts.

21.8 Timer1/3/5 Operation During Sleep

Timer1/3/5 can only operate during Sleep when set up in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- ON bit of the TxCON register must be set
- TMRxIE bit of the respective PIE register must be set
- $\overline{\text{SYNC}}$ bit of the TxCON register must be set
- Configure the TMRxCLK register for using secondary oscillator as the clock source
- Enable the SOSCEN bit of the OSCEN register (Register 7-7)

The device will wake-up on an overflow and execute the next instruction. If the GIE/GIEH bit of the INTCON0 register is set, the device will call the Interrupt Service Routine.

The secondary oscillator will continue to operate in Sleep regardless of the $\overline{\text{SYNC}}$ bit setting.

21.9 CCP Capture/Compare Time Base

The CCP modules use the TMRxH:TMRxL register pair as the time base when operating in Capture or Compare mode.

In Capture mode, the value in the TMRxH:TMRxL register pair is copied into the CCPRxH:CCPRxL register pair on a configured event.

In Compare mode, an event is triggered when the value in the CCPRxH:CCPRxL register pair matches the value in the TMRxH:TMRxL register pair. This event can be a Special Event Trigger.

For more information, see **Section 23.0 “Capture/Compare/PWM Module”**.

21.10 CCP Special Event Trigger

When any of the CCP’s are configured to trigger a special event, the trigger will clear the TMRxH:TMRxL register pair. This special event does not cause a Timer1/3/5 interrupt. The CCP module may still be configured to generate a CCP interrupt.

In this mode of operation, the CCPRxH:CCPRxL register pair becomes the period register for Timer1/3/5.

Timer1/3/5 should be synchronized and FOSC/4 should be selected as the clock source in order to utilize the Special Event Trigger. Asynchronous operation of Timer1/3/5 can cause a Special Event Trigger to be missed.

In the event that a write to TMRxH or TMRxL coincides with a Special Event Trigger from the CCP, the write will take precedence.

23.0 CAPTURE/COMPARE/PWM MODULE

The Capture/Compare/PWM module is a peripheral that allows the user to time and control different events, and to generate Pulse-Width Modulation (PWM) signals. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate pulse-width modulated signals of varying frequency and duty cycle.

This family of devices contains four standard Capture/ Compare/PWM modules (CCP1, CCP2, CCP3 and CCP4). Each individual CCP module can select the timer source that controls the module. Each module has an independent timer selection which can be accessed using the CxTSEL bits in the CCPTMRS0 register (Register 23-2). The default timer selection is TMR1 when using Capture/Compare mode and TMR2 when using PWM mode in the CCPx module.

Please note that the Capture/Compare mode operation is described with respect to TMR1 and the PWM mode operation is described with respect to TMR2 in the following sections.

The Capture and Compare functions are identical for all CCP modules.

Note 1: In devices with more than one CCP module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the CCP1CON and CCP2CON control the same operational aspects of two completely different CCP modules.

2: Throughout this section, generic references to a CCP module in any of its operating modes may be interpreted as being equally applicable to CCPx module. Register names, module signals, I/O pins, and bit names may use the generic designator ‘x’ to indicate the use of a numeral to distinguish a particular module, when required.

23.1 CCP Module Configuration

Each Capture/Compare/PWM module is associated with a control register (CCPxCON), a capture input selection register (CCPxCAP) and a data register (CCPRx). The data register, in turn, is comprised of two 8-bit registers: CCPRxL (low byte) and CCPRxH (high byte).

23.1.1 CCP MODULES AND TIMER RESOURCES

The CCP modules utilize Timers 1 through 6 that vary with the selected mode. Various timers are available to the CCP modules in Capture, Compare or PWM modes, as shown in Table 23-1.

TABLE 23-1: CCP MODE – TIMER RESOURCE

CCP Mode	Timer Resource
Capture	Timer1, Timer3 or Timer5
Compare	
PWM	Timer2, Timer4 or Timer6

The assignment of a particular timer to a module is determined by the timer to CCP enable bits in the CCPTMRS0 register (see Register 23-2) All of the modules may be active at once and may share the same timer resource if they are configured to operate in the same mode (Capture/Compare or PWM) at the same time.

23.1.2 OPEN-DRAIN OUTPUT OPTION

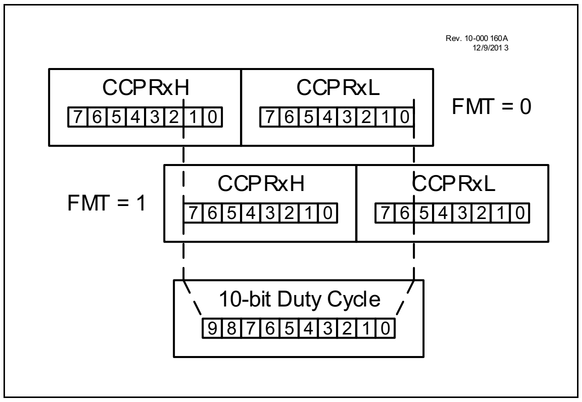
When operating in Output mode (the Compare or PWM modes), the drivers for the CCPx pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor and allows the output to communicate with external circuits without the need for additional level shifters.

23.4.5 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to the CCPRxH:CCPRxL register pair. The alignment of the 10-bit value is determined by the FMT bit of the CCPxCON register (see Figure 23-5). The CCPRxH:CCPRxL register pair can be written to at any time; however the duty cycle value is not latched into the 10-bit buffer until after a match between T2PR and T2TMR.

Equation 23-2 is used to calculate the PWM pulse width. Equation 23-3 is used to calculate the PWM duty cycle ratio.

FIGURE 23-5: PWM 10-BIT ALIGNMENT



EQUATION 23-2: PULSE WIDTH

$$\begin{aligned}\mathit{Pulse\ Width} &= (\mathit{CCPRxH{:}CCPRxL\ register\ pair}) \bullet \\ &\qquad T_{OSC} \bullet (\mathit{TMR2\ Prescale\ Value})\end{aligned}$$

EQUATION 23-3: DUTY CYCLE RATIO

$$\mathit{Duty\ Cycle\ Ratio} = \frac{(\mathit{CCPRxH{:}CCPRxL\ register\ pair})}{4(T2PR + 1)}$$

CCPRxH:CCPRxL register pair are used to double buffer the PWM duty cycle. This double buffering provides glitchless PWM operation.

The 8-bit timer T2TMR register is concatenated with either the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

When the 10-bit time base matches the CCPRxH:CCPRxL register pair, then the CCPx pin is cleared (see Figure 23-4).

23.4.6 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

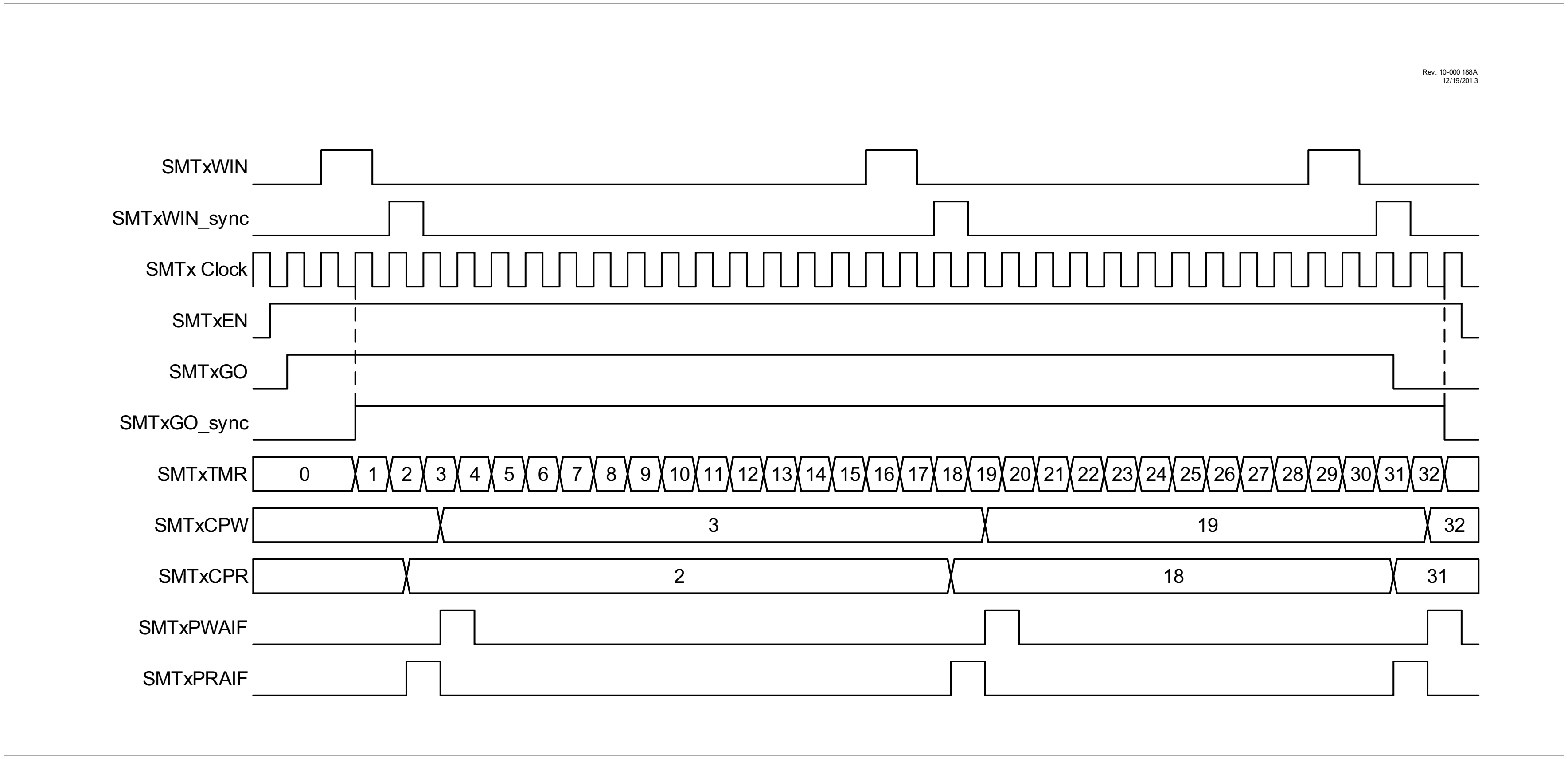
The maximum PWM resolution is ten bits when T2PR is 255. The resolution is a function of the T2PR register value as shown by Equation 23-4.

EQUATION 23-4: PWM RESOLUTION

$$\mathit{Resolution} = \frac{\log[4(T2PR + 1)]}{\log(2)}\ \mathit{bits}$$

Note: If the pulse-width value is greater than the period, the assigned PWM pin(s) will remain unchanged.

FIGURE 25-16: CAPTURE MODE REPEAT ACQUISITION TIMING DIAGRAM



REGISTER 26-7: CWGxAS1: CWG AUTO-SHUTDOWN CONTROL REGISTER 1

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	q = Value depends on condition

bit 7 **Unimplemented** Read as ‘0’

bit 6 **AS6E:** CWG Auto-shutdown Source 6 Enable bit
1 = Auto-shutdown for Source 6 is enabled

CWG Module	CWG1	CWG2	CWG3
Auto-shutdown Source 6	CLC2 OUT	CLC3 OUT	CLC4 OUT

0 = Auto-shutdown for Source 6 is disabled

bit 5 **AS5E:** CWG Auto-shutdown Source 5 (CMP2 OUT) Enable bit
1 = Auto-shutdown for CMP2 OUT is enabled
0 = Auto-shutdown for CMP2 OUT is disabled

bit 4 **AS4E:** CWG Auto-shutdown Source 4 (CMP1 OUT) Enable bit
1 = Auto-shutdown for CMP1 OUT is enabled
0 = Auto-shutdown for CMP1 OUT is disabled

bit 3 **AS3E:** CWG Auto-shutdown Source 3 (TMR6_Postscaled) Enable bit
1 = Auto-shutdown for TMR6_Postscaled is enabled
0 = Auto-shutdown for TMR6_Postscaled is disabled

bit 2 **AS2E:** CWG Auto-shutdown Source 2 (TMR4_Postscaled) Enable bit
1 = Auto-shutdown for TMR4_Postscaled is enabled
0 = Auto-shutdown for TMR4_Postscaled is disabled

bit 1 **AS1E:** CWG Auto-shutdown Source 1 (TMR2_Postscaled) Enable bit
1 = Auto-shutdown for TMR2_Postscaled is enabled
0 = Auto-shutdown for TMR2_Postscaled is disabled

bit 0 **AS0E:** CWG Auto-shutdown Source 0 (Pin selected by CWGxPPS) Enable bit
1 = Auto-shutdown for CWGxPPS Pin is enabled
0 = Auto-shutdown for CWGxPPS Pin is disabled

REGISTER 27-2: CLCxPOL: SIGNAL POLARITY CONTROL REGISTER

R/W-0/0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
POL	—	—	—	G4POL	G3POL	G2POL	G1POL
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **POL:** CLCxOUT Output Polarity Control bit
1 = The output of the logic cell is inverted
0 = The output of the logic cell is not inverted

bit 6-4 **Unimplemented:** Read as ‘0’

bit 3 **G4POL:** Gate 3 Output Polarity Control bit
1 = The output of gate 3 is inverted when applied to the logic cell
0 = The output of gate 3 is not inverted

bit 2 **G3POL:** Gate 2 Output Polarity Control bit
1 = The output of gate 2 is inverted when applied to the logic cell
0 = The output of gate 2 is not inverted

bit 1 **G2POL:** Gate 1 Output Polarity Control bit
1 = The output of gate 1 is inverted when applied to the logic cell
0 = The output of gate 1 is not inverted

bit 0 **G1POL:** Gate 0 Output Polarity Control bit
1 = The output of gate 0 is inverted when applied to the logic cell
0 = The output of gate 0 is not inverted

REGISTER 30-2: MD1CON1: MODULATION CONTROL REGISTER 1

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	CHPOL	CHSYNC	—	—	CLPOL	CLSYNC
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5 **CHPOL:** Modulator High Carrier Polarity Select bit
1 = Selected high carrier signal is inverted
0 = Selected high carrier signal is not inverted

bit 4 **CHSYNC:** Modulator High Carrier Synchronization Enable bit
1 = Modulator waits for a falling edge on the high time carrier signal before allowing a switch to the low time carrier
0 = Modulator output is not synchronized to the high time carrier signal[(1)]

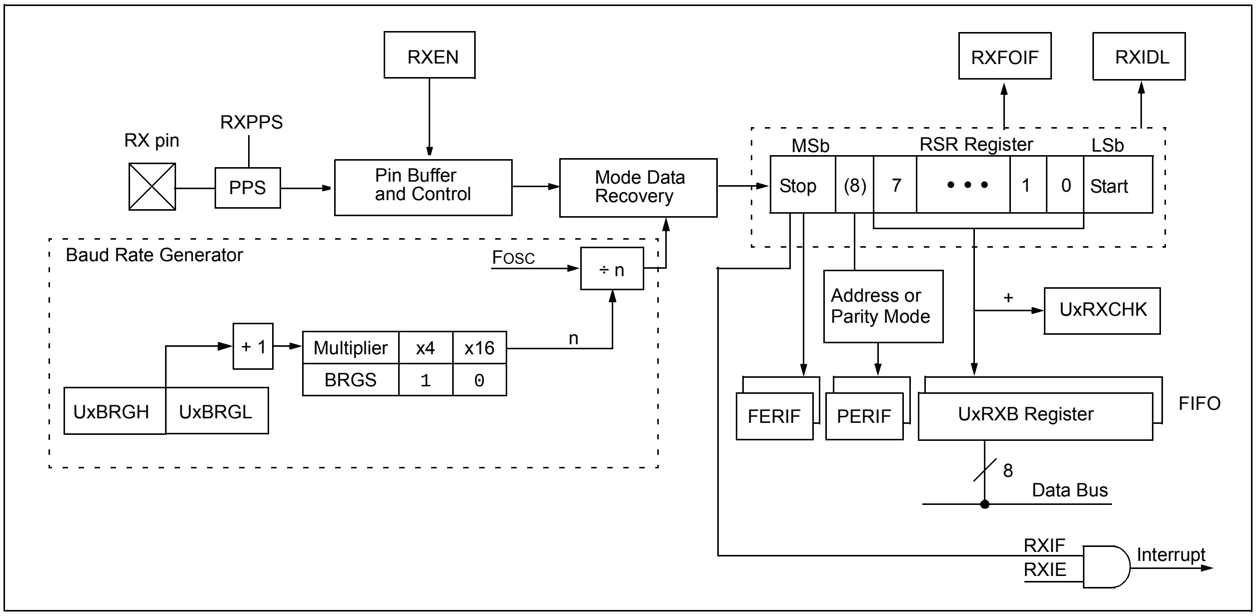
bit 3-2 **Unimplemented:** Read as ‘0’

bit 1 **CLPOL:** Modulator Low Carrier Polarity Select bit
1 = Selected low carrier signal is inverted
0 = Selected low carrier signal is not inverted

bit 0 **CLSYNC:** Modulator Low Carrier Synchronization Enable bit
1 = Modulator waits for a falling edge on the low time carrier signal before allowing a switch to the high time carrier
0 = Modulator output is not synchronized to the low time carrier signal[(1)]

Note 1: Narrowed carrier pulse widths or spurs may occur in the signal stream if the carrier is not synchronized.

FIGURE 31-2: UART RECEIVE BLOCK DIAGRAM



The operation of the UART module is controlled through nineteen registers:

- Three control registers (UxCON0-UxCON2)
- Error enable and status (UxERRIE, UxERRIR, UxUIR)
- UART buffer status and control (UxFIFO)
- Three 9-bit protocol parameters (UxP1-UxP3)
- 16-bit baud rate generator (UxBRGH:L)
- Transmit buffer write (UxTXB)
- Receive buffer read (UxRXB)
- Receive checksum (UxRXCHK)
- Transmit checksum (UxTXCHK)

These registers are detailed in **Section 31.21 “Register Definitions: UART Control”**.

31.1 UART I/O Pin Configuration

The RX input pin is selected with the UxRPPS register. The TX output pin is selected with each pin’s RxyPPS register. When the TRIS control for the pin corresponding to the TX output is cleared, then the UART will maintain control and the logic level on the TX pin. Changing the TXPOL bit in UxCON2 will immediately change the TX pin logic level regardless of the value of EN or TXEN.

31.2 UART Asynchronous Modes

The UART has five asynchronous modes:

- 7-bit
- 8-bit
- 8-bit with even parity in the 9th bit
- 8-bit with odd parity in the 9th bit
- 8-bit with address indicator in the 9th bit

The UART transmits and receives data using the standard Non-Return-to-Zero (NRZ) format. NRZ is implemented with two levels: a VOH Mark state, which represents a ‘1’ data bit, and a VOL Space state, which represents a ‘0’ data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the Mark state. Each character transmission consists of one Start bit followed by seven or eight data bits, one optional parity or address bit, and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is eight bits with no parity. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See **Section 31.17 “UART Baud Rate Generator (BRG)”** for more information.

In all the Asynchronous modes, the UART transmits and receives the LSb first. The UART’s transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is supported by the hardware by even and odd parity modes.

31.2.1 UART ASYNCHRONOUS TRANSMITTER

The UART transmitter block diagram is shown in Figure 31-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the UxTXB register.

31.13 Checksum

This section does not apply to the LIN mode, which handles checksums automatically.

The transmit and receive checksum adders are enabled when the C0EN bit in the UxCON2 register is set. When enabled, the adders accumulate every byte that is transmitted or received. The accumulated sum includes the carry of the addition. Software is responsible for clearing the checksum registers before a transaction and performing the check at the end of the transaction.

The following is an example of how the checksum registers could be used in the Asynchronous modes.

31.13.1 TRANSMIT CHECKSUM METHOD

1. Clear the UxTXCHK register.
2. Set the C0EN bit.
3. Send all bytes of the transaction output.
4. Invert UxTXCHK and send the result as the last byte of the transaction.

31.13.2 RECEIVE CHECKSUM METHOD

1. Clear the UxRXCHK register.
2. Set the C0EN bit.
3. Receive all bytes in the transaction including the checksum byte.
4. Set MSb of UxRXCHK if 7-bit mode is selected.
5. Add 1 to UxRXCHK.
6. If the result is ‘0’, the checksum passes, otherwise it fails.

The CERIF checksum interrupt flag is not active in any mode other than LIN.

31.14 Collision Detection

External forces that interfere with the transmit line are detected in all modes of operation with collision detection. Collision detection is always active when RXEN and TXEN are both set.

When the receive input is connected to the transmit output through either the same I/O pin or external circuitry, a character will be received for every character transmitted. The collision detection circuit provides a warning when the word received does not match the word transmitted.

The TXCIF flag in the UxERRIR register is used to signal collisions. This signal is only useful when the TX output is looped back to the RX input and everything that is transmitted is expected to be received. If more than one transmitter is active at the same time, it can be assumed that the TX word will not match the RX word. The TXCIF detects this mismatch and flags an interrupt. The TXCIF bit will also be set in DALI mode transmissions when the received bit is missing the expected mid-bit transition.

Collision detection is always active, regardless of whether or not the RX input is connected to the TX output. It is up to the user to disable the TXCIE bit when collision interrupts are not required.

The software overhead of unloading the receive buffer of transmitted data is avoided by setting the RUNOVF bit in UxCON2 and ignoring the receive interrupt and letting the receive buffer overflow. When the transmission is complete, prepare for receiving data by flushing the receive buffer (see Section 31.11.2, FIFO Reset) and clearing the RXFOIF overflow flag in the UxERRIR register.

31.15 RX/TX Activity Timeout

The UART works in conjunction with the HLT timers to monitor activity on the RX and TX lines. Use this feature to determine when there has been no activity on the receive or transmit lines for a user specified period of time.

To use this feature, set the HLT to the desired timeout period by a combination of the HLT clock source, timer prescale value, and timer period registers. Configure the HLT to reset on the UART TX or RX line and start the HLT at the same time the UART is started. UART activity will keep resetting the HLT to prevent a full HLT period from elapsing. When there has been no activity on the selected TX or RX line for longer than the HLT period then an HLT interrupt will occur signaling the timeout event.

For example, the following register settings will configure HLT2 for a 5 ms timeout of no activity on U1RX:

- T2PR = 0x9C (156 prescale periods)
- T2CLKCON = 0x05 (500 kHz internal oscillator)
- T2HLT = 0x04 (free running, reset on rising edge)
- T2RST = 0x15 (reset on U1RX)
- T2CON = 0xC0 (Timer2 on with 1:16 prescale)

34.14 CAN Interrupts

The module has several sources of interrupts. Each of these interrupts can be individually enabled or disabled. The PIR5 register contains interrupt flags. The PIE5 register contains the enables for the eight main interrupts. A special set of read-only bits in the CANSTAT register, the ICODE bits, can be used in combination with a jump table for efficient handling of interrupts.

All interrupts have one source, with the exception of the error interrupt and buffer interrupts in Mode 1 and 2. Any of the error interrupt sources can set the error interrupt flag. The source of the error interrupt can be determined by reading the Communication Status register, COMSTAT. In Mode 1 and 2, there are two interrupt enable/disable and flag bits – one for all transmit buffers and the other for all receive buffers.

The interrupts can be broken up into two categories: receive and transmit interrupts.

The receive related interrupts are:

- Receive Interrupts
- Wake-up Interrupt
- Receiver Overrun Interrupt
- Receiver Warning Interrupt
- Receiver Error-Passive Interrupt

The transmit related interrupts are:

- Transmit Interrupts
- Transmitter Warning Interrupt
- Transmitter Error-Passive Interrupt
- Bus-Off Interrupt

34.14.1 INTERRUPT CODE BITS

To simplify the interrupt handling process in user firmware, the ECAN module encodes a special set of bits. In Mode 0, these bits are ICODE<3:1> in the CANSTAT register. In Mode 1 and 2, these bits are EICODE<4:0> in the CANSTAT register. Interrupts are internally prioritized such that the higher priority interrupts are assigned lower values. Once the highest priority interrupt condition has been cleared, the code for the next highest priority interrupt that is pending (if any) will be reflected by the ICODE bits (see Table 34-4). Note that only those interrupt sources that have their associated interrupt enable bit set will be reflected in the ICODE bits.

In Mode 2, when a receive message interrupt occurs, the EICODE bits will always consist of '10000'. User firmware may use FIFO Pointer bits to actually access the next available buffer.

34.14.2 TRANSMIT INTERRUPT

When the transmit interrupt is enabled, an interrupt will be generated when the associated transmit buffer becomes empty and is ready to be loaded with a new message. In Mode 0, there are separate interrupt enable/disable and flag bits for each of the three dedicated transmit buffers. The TXBnIF bit will be set to indicate the source of the interrupt. The interrupt is cleared by the MCU, resetting the TXBnIF bit to a '0'. In Mode 1 and 2, all transmit buffers share one interrupt enable/disable bit and one flag bit. In Mode 1 and 2, TXBnIE in PIE5 and TXBnIF in PIR5 indicate when a transmit buffer has completed transmission of its message. TXBnIF, TXBnIE and TXBnIP in PIR5, PIE5 and IPR5, respectively, are not used in Mode 1 and 2. Individual transmit buffer interrupts can be enabled or disabled by setting or clearing TXBnIE and B0IE register bits. When a shared interrupt occurs, user firmware must poll the TXREQ bit of all transmit buffers to detect the source of interrupt.

34.14.3 RECEIVE INTERRUPT

When the receive interrupt is enabled, an interrupt will be generated when a message has been successfully received and loaded into the associated receive buffer. This interrupt is activated immediately after receiving the End-of-Frame (EOF) field.

In Mode 0, the RXBnIF bit is set to indicate the source of the interrupt. The interrupt is cleared by the MCU, resetting the RXBnIF bit to a '0'.

In Mode 1 and 2, all receive buffers share RXBnIE, RXBnIF and RXBnIP in PIE5, PIR5 and IPR5, respectively. Individual receive buffer interrupts can be controlled by the TXBnIE and BIE0 registers. In Mode 1, when a shared receive interrupt occurs, user firmware must poll the RXFUL bit of each receive buffer to detect the source of interrupt. In Mode 2, a receive interrupt indicates that the new message is loaded into FIFO. FIFO can be read by using FIFO Pointer bits, FP.

REGISTER 34-48: MSEL0: MASK SELECT REGISTER 0[1]

R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
FIL3_1	FIL3_0	FIL2_1	FIL2_0	FIL1_1	FIL1_0	FIL0_1	FIL0_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 7-6 **FIL3_<1:0>:** Filter 3 Select bits 1 and 0

11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0

bit 5-4 **FIL2_<1:0>:** Filter 2 Select bits 1 and 0

11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0

bit 3-2 **FIL1_<1:0>:** Filter 1 Select bits 1 and 0

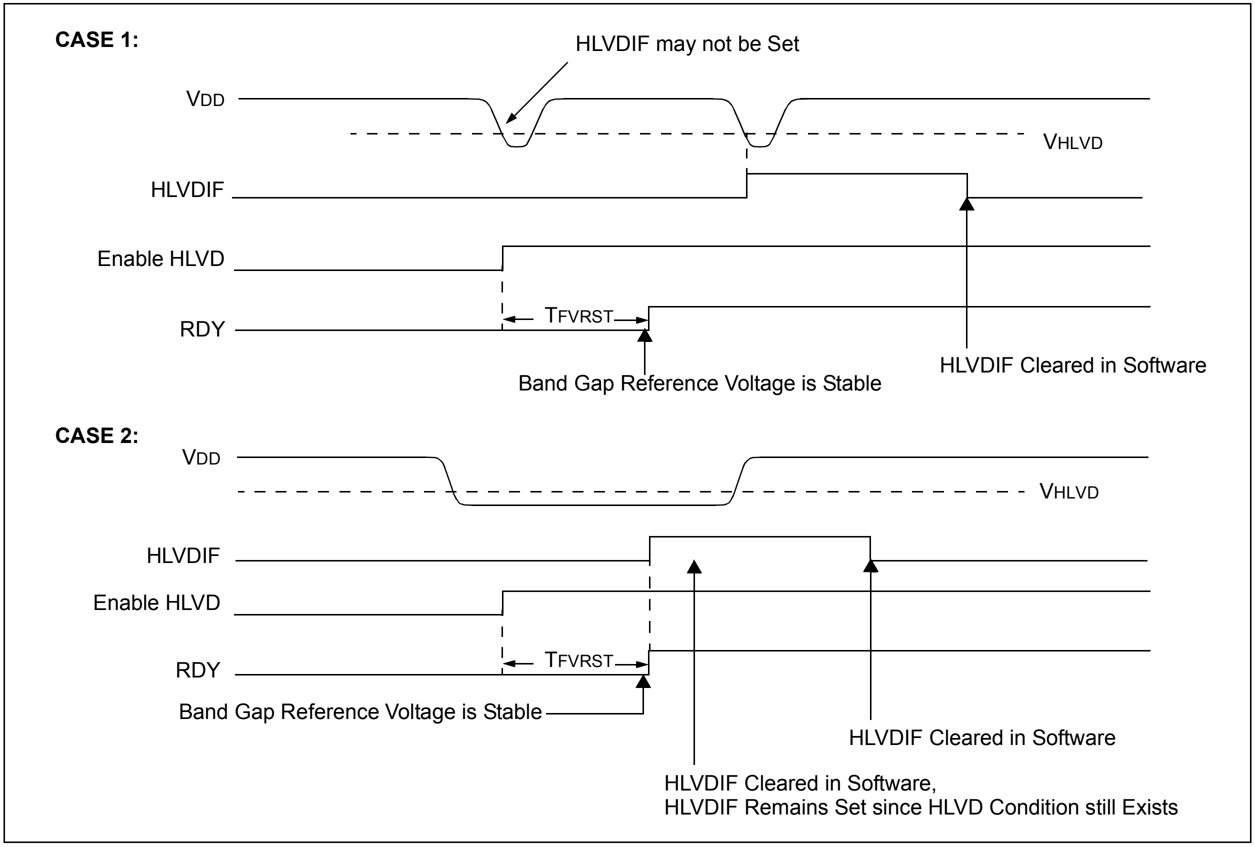
11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0

bit 1-0 **FIL0_<1:0>:** Filter 0 Select bits 1 and 0

11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0

Note 1: This register is available in Mode 1 and 2 only.

FIGURE 40-2: LOW-VOLTAGE DETECT OPERATION (INTL = 1)


CASE 1:
HLVDIF may not be Set
VDD
VHLVD
HLVDIF
Enable HLVD
TFVRST
RDY
Band Gap Reference Voltage is Stable
HLVDIF Cleared in Software
CASE 2:
VDD
VHLVD
HLVDIF
Enable HLVD
TFVRST
RDY
Band Gap Reference Voltage is Stable
HLVDIF Cleared in Software
HLVDIF Cleared in Software,
HLVDIF Remains Set since HLVD Condition still Exists

40.6 Operation During Sleep

When enabled, the HLVD circuitry continues to operate during Sleep. If the device voltage crosses the trip point, the HLVDIF bit will be set and the device will wake up from Sleep. Device execution will continue from the interrupt vector address if interrupts have been globally enabled.

40.7 Operation During Idle and Doze Modes

In both Idle and Doze modes, the module is active and events are generated if peripheral is enabled.

40.8 Operation During Freeze

When in Freeze mode, no new event or interrupt can be generated. The state of the LRDY bit is frozen.

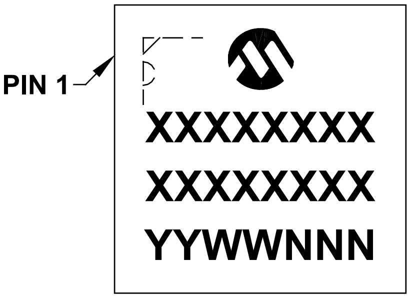
Register reads and writes through the CPU interface are allowed.

40.9 Effects of a Reset

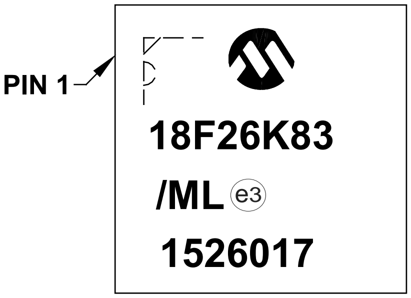
A device Reset forces all registers to their Reset state. This forces the HLVD module to be turned off.

Package Marking Information (Continued)

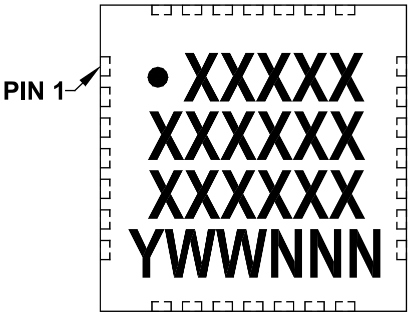
28-Lead QFN (6x6 mm)



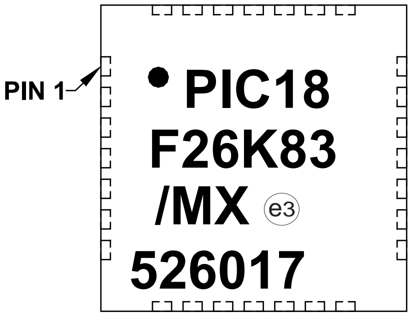
Example



28-Lead UQFN (6x6x0.5 mm)



Example



Legend:	XX...X	Customer-specific information or Microchip part number
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	e3	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.