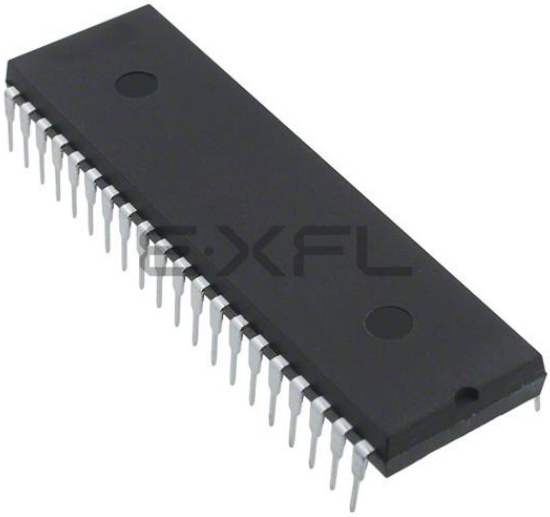


Welcome to E-XFL.COM

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	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	64KB (64K x 8)
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 35x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf46k42-e-p

3.1.1 PRIORITY LOCK

The System arbiter grants memory access to the peripheral selections (DMAx, Scanner) when the PRLOCKED bit (PRLOCK Register) is set.

Priority selections are locked by setting the PRLOCKED bit of the PRLOCK register. Setting and clearing this bit requires a special sequence as an extra precaution against inadvertent changes. Examples of setting and clearing the PRLOCKED bit are shown in Example 3-1 and Example 3-2.

EXAMPLE 3-1: PRIORITY LOCK SEQUENCE

```
; Disable interrupts
BCF INTCON0,GIE

; Bank to PRLOCK register
BANKSEL PRLOCK
MOVLW 55h

; Required sequence, next 4
instructions
MOVWF PRLOCK
MOVLW AAh
MOVWF PRLOCK
; Set PRLOCKED bit to grant memory
access to peripherals
BSF PRLOCK,0

; Enable Interrupts
BSF INTCON0,GIE
```

EXAMPLE 3-2: PRIORITY UNLOCK SEQUENCE

```
; Disable interrupts
BCF INTCON0,GIE

; Bank to PRLOCK register
BANKSEL PRLOCK
MOVLW 55h

; Required sequence, next 4
instructions
MOVWF PRLOCK
MOVLW AAh
MOVWF PRLOCK
; Clear PRLOCKED bit to allow changing
priority settings
BCF PRLOCK,0

; Enable Interrupts
BSF INTCON0,GIE
```

3.2 Memory Access Scheme

The user can assign priorities to both system level and peripheral selections based on which the system arbiter grants memory access. Let us consider the following priority scenarios between ISR, MAIN, and Peripherals.

Note:	It is always required that the ISR priority be higher than Main priority.

3.2.1 ISR PRIORITY > MAIN PRIORITY > PERIPHERAL PRIORITY

When the Peripheral Priority (DMAx, Scanner) is lower than ISR and MAIN Priority, and the peripheral requires:

1. Access to the Program Flash Memory, then the peripheral waits for an instruction cycle in which the CPU does not need to access the PFM (such as a branch instruction) and uses that cycle to do its own Program Flash Memory access, unless a PFM Read/Write operation is in progress.
2. Access to the SFR/GPR, then the peripheral waits for an instruction cycle in which the CPU does not need to access the SFR/GPR (such as `MOVLW`, `CALL`, `NOP`) and uses that cycle to do its own SFR/GPR access.
3. Access to the Data EEPROM, then the peripheral has access to Data EEPROM unless a Data EEPROM Read/Write operation is being performed.

This results in the lowest throughput for the peripheral to access the memory, and does so without any impact on execution times.

3.2.2 PERIPHERAL PRIORITY > ISR PRIORITY > MAIN PRIORITY

When the Peripheral Priority (DMAx, Scanner) is higher than ISR and MAIN Priority, the CPU operation is stalled when the peripheral requests memory.

The CPU is held in its current state until the peripheral completes its operation. Since the peripheral requests access to the bus, the peripheral cannot be disabled until it completes its operation.

This results in the highest throughput for the peripheral to access the memory, but has the cost of stalling other execution while it occurs.

TABLE 4-9: SPECIAL FUNCTION REGISTER MAP FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES BANK 58

3AFFh	—	3ADFh	SPI1SDIPPS	3ABFh	PPSLOCK	3A9Fh	—	3A7Fh	—	3A5Fh	—	3A3Fh	—	3A1Fh	RD7PPS(2)
3AFEh	—	3ADEh	SPI1SCKPPS	3ABEh	—(4)	3A9Eh	—	3A7Eh	—	3A5Eh	—	3A3Eh	—	3A1Eh	RD6PPS(2)
3AFDh	—	3ADDh	ADACTPPS	3ABDh	—	3A9Dh	—	3A7Dh	—	3A5Dh	—	3A3Dh	—	3A1Dh	RD5PPS(2)
3AFCh	—	3ADCh	CLCIN3PPS	3ABCh	—	3A9Ch	—	3A7Ch	—	3A5Ch	—	3A3Ch	—	3A1Ch	RD4PPS(2)
3AFBh	—	3ADBh	CLCIN2PPS	3ABBh	—	3A9Bh	—	3A7Bh	RD1I2C(2)	3A5Bh	RB2I2C	3A3Bh	—	3A1Bh	RD3PPS(2)
3AFAh	—	3ADAh	CLCIN1PPS	3ABAh	—	3A9Ah	—	3A7Ah	RD0I2C(2)	3A5Ah	RB1I2C	3A3Ah	—	3A1Ah	RD2PPS(2)
3AF9h	—	3AD9h	CLCIN0PPS	3AB9h	—	3A99h	—(4)	3A79h	—(4)	3A59h	—(4)	3A39h	—	3A19h	RD1PPS(2)
3AF8h	—	3AD8h	MD1SRCPPS	3AB8h	—	3A98h	—(4)	3A78h	—(4)	3A58h	—(4)	3A38h	—	3A18h	RD0PPS(2)
3AF7h	—	3AD7h	MD1CARHPPS	3AB7h	—	3A97h	—	3A77h	—	3A57h	IOCBF	3A37h	—	3A17h	RC7PPS
3AF6h	—	3AD6h	MD1CARLPPS	3AB6h	—	3A96h	—	3A76h	—	3A56h	IOCBN	3A36h	—	3A16h	RC6PPS
3AF5h	—	3AD5h	CWG3INPPS	3AB5h	—	3A95h	—	3A75h	—	3A55h	IOCBP	3A35h	—	3A15h	RC5PPS
3AF4h	—	3AD4h	CWG2INPPS	3AB4h	—	3A94h	INLVLF(3)	3A74h	INLVLD(2)	3A54h	INLVLB	3A34h	—	3A14h	RC4PPS
3AF3h	—	3AD3h	CWG1INPPS	3AB3h	—	3A93h	SLRCONF(3)	3A73h	SLRCOND(2)	3A53h	SLRCONB	3A33h	—	3A13h	RC3PPS
3AF2h	—	3AD2h	SMT1SIGPPS	3AB2h	—	3A92h	ODCONF(3)	3A72h	ODCOND(2)	3A52h	ODCONB	3A32h	—	3A12h	RC2PPS
3AF1h	—	3AD1h	SMT1WINPPS	3AB1h	—	3A91h	WPUF(3)	3A71h	WPUD(2)	3A51h	WPUB	3A31h	—	3A11h	RC1PPS
3AF0h	—	3AD0h	CCP4PPS	3AB0h	—	3A90h	ANSELF(3)	3A70h	ANSELD(2)	3A50h	ANSELB	3A30h	—	3A10h	RC0PPS
3AEFh	—	3ACFh	CCP3PPS	3AAFh	—	3A8Fh	—	3A6Fh	—	3A4Fh	—	3A2Fh	RF7PPS(3)	3A0Fh	RB7PPS
3AEEh	—	3ACEh	CCP2PPS	3AAEh	—	3A8Eh	—	3A6Eh	—	3A4Eh	—	3A2Eh	RF6PPS(3)	3A0Eh	RB6PPS
3AEDh	—	3ACDh	CCP1PPS	3AADh	—	3A8Dh	—	3A6Dh	—	3A4Dh	—	3A2Dh	RF5PPS(3)	3A0Dh	RB5PPS
3AECh	—	3ACCh	T6INPPS	3AACh	—	3A8Ch	—	3A6Ch	—	3A4Ch	—	3A2Ch	RF4PPS(3)	3A0Ch	RB4PPS
3AEBh	—	3ACBh	T4INPPS	3AABh	—	3A8Bh	—	3A6Bh	RC4I2C	3A4Bh	—	3A2Bh	RF3PPS(3)	3A0Bh	RB3PPS
3AEAh	—	3ACAh	T2INPPS	3AAAh	—	3A8Ah	—	3A6Ah	RC3I2C	3A4Ah	—	3A2Ah	RF2PPS(3)	3A0Ah	RB2PPS
3AE9h	U2CTSPPS	3AC9h	T5GPPS	3AA9h	—	3A89h	—(4)	3A69h	—(4)	3A49h	—(4)	3A29h	RF1PPS(3)	3A09h	RB1PPS
3AE8h	U2RXPPS	3AC8h	T5CKIPPS	3AA8h	—	3A88h	—(4)	3A68h	—(4)	3A48h	—(4)	3A28h	RF0PPS(3)	3A08h	RB0PPS
3AE7h	—	3AC7h	T3GPPS	3AA7h	—	3A87h	IOCEF	3A67h	IOCCF	3A47h	IOCAF	3A27h	—	3A07h	RA7PPS
3AE6h	U1CTSPPS	3AC6h	T3CKIPPS	3AA6h	—	3A86h	IOCEN	3A66h	IOCCN	3A46h	IOCAN	3A26h	—	3A06h	RA6PPS
3AE5h	U1RXPPS	3AC5h	T1GPPS	3AA5h	—	3A85h	IOCEP	3A65h	IOCCP	3A45h	IOCAP	3A25h	—	3A05h	RA5PPS
3AE4h	I2C2SDAPPS	3AC4h	T1CKIPPS	3AA4h	—	3A84h	INLVLE	3A64h	INLVLC	3A44h	INLVLA	3A24h	—	3A04h	RA4PPS
3AE3h	I2C2SCLPPS	3AC3h	T0CKIPPS	3AA3h	—	3A83h	SLRCONE(2)	3A63h	SLRCONC	3A43h	SLRCONA	3A23h	—	3A03h	RA3PPS
3AE2h	I2C1SDAPPS	3AC2h	INT2PPS	3AA2h	—	3A82h	ODCONE(2)	3A62h	ODCONC	3A42h	ODCONA	3A22h	RE2PPS(2)	3A02h	RA2PPS
3AE1h	I2C1SCLPPS	3AC1h	INT1PPS	3AA1h	—	3A81h	WPUE	3A61h	WPUC	3A41h	WPUA	3A21h	RE1PPS(2)	3A01h	RA1PPS
3AE0h	SPI1SSPPS	3AC0h	INT0PPS	3AA0h	—	3A80h	ANSELE(2)	3A60h	ANSELC	3A40h	ANSELA	3A20h	RE0PPS(2)	3A00h	RA0PPS

Legend: Unimplemented data memory locations and registers, read as '0'.

Note 1: Unimplemented in LF devices.

2: Unimplemented in PIC18(L)F26/27K42.

3: Unimplemented in PIC18(L)F26/27/45/46/47K42.

4: Reserved, maintain as '0'.

6.14 Register Definitions: Power Control

REGISTER 6-2: PCON0: POWER CONTROL REGISTER 0

R/W/HS-0/q	R/W/HS-0/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-0/u	R/W/HC-q/u
STKOVF	STKUNF	$\overline{\text{WDTWV}}$	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

Legend:		
HC = Bit is cleared by hardware		HS = Bit is set by hardware
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-m/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 **STKOVF:** Stack Overflow Flag bit
- 1 = A Stack Overflow occurred (more CALLs than fit on the stack)
- 0 = A Stack Overflow has not occurred or set to '0' by firmware

bit 6 **STKUNF:** Stack Underflow Flag bit
- 1 = A Stack Underflow occurred (more RETURNs than CALLs)
- 0 = A Stack Underflow has not occurred or set to '0' by firmware

bit 5 **$\overline{\text{WDTWV}}$:** Watchdog Window Violation bit
- 1 = A WDT window violation has not occurred or set to '1' by firmware
- 0 = A CLRWDT instruction was issued when the WDT Reset window was closed (set to '0' in hardware when a WDT window violation Reset occurs)

bit 4 **$\overline{\text{RWDT}}$:** WDT Reset Flag bit
- 1 = A WDT overflow/time-out Reset has not occurred or set to '1' by firmware
- 0 = A WDT overflow/time-out Reset has occurred (set to '0' in hardware when a WDT Reset occurs)

bit 3 **$\overline{\text{RMCLR}}$:** $\overline{\text{MCLR}}$ Reset Flag bit
- 1 = A $\overline{\text{MCLR}}$ Reset has not occurred or set to '1' by firmware
- 0 = A $\overline{\text{MCLR}}$ Reset has occurred (set to '0' in hardware when a $\overline{\text{MCLR}}$ Reset occurs)

bit 2 **$\overline{\text{RI}}$:** RESET Instruction Flag bit
- 1 = A RESET instruction has not been executed or set to '1' by firmware
- 0 = A RESET instruction has been executed (set to '0' in hardware upon executing a RESET instruction)

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit
- 1 = No Power-on Reset occurred or set to '1' by firmware
- 0 = A Power-on Reset occurred (set to '0' in hardware when a Power-on Reset occurs)

bit 0 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit
- 1 = No Brown-out Reset occurred or set to '1' by firmware
- 0 = A Brown-out Reset occurred (set to '0' in hardware when a Brown-out Reset occurs)

EXAMPLE 9-4: SETTING UP VECTORED INTERRUPTS USING XC8

```
// NOTE 1: If IVTBASE is changed from its default value of 0x000008, then the
// "base(...)" argument must be provided in the ISR. Otherwise the vector
// table will be placed at 0x0008 by default regardless of the IVTBASE value.

// NOTE 2: When MVECEN=0 and IPEN=1, a separate argument as "high_priority"
// or "low_priority" can be used to distinguish between the two ISRs.
// If the argument is not provided, the ISR is considered high priority
// by default.

// NOTE 3: Multiple interrupts can be handled by the same ISR if they are
// specified in the "irq(...)" argument. Ex: irq(IRQ_TMR0, IRQ_CCP1)

void __interrupt(irq(IRQ_TMR0), base(0x4008)) TMR0_ISR(void)
{
        PIR3bits.TMR0IF = 0;                 // Clear the interrupt flag
        LATCbits.LC0 ^= 1;                   // ISR code goes here
}

void __interrupt(irq(default), base(0x4008)) DEFAULT_ISR(void)
{
        // Unhandled interrupts go here
}

void INTERRUPT_Initialize (void)
{
        INTCON0bits.GIEH = 1;                // Enable high priority interrupts
        INTCON0bits.GIEL = 1;                // Enable low priority interrupts
        INTCON0bits.IPEN = 1;                // Enable interrupt priority

        PIE3bits.TMR0IE = 1;                 // Enable TMR0 interrupt
        PIE4bits.TMR1IE = 1;                 // Enable TMR1 interrupt

        IPR3bits.TMR0IP = 0;                 // Make TMR0 interrupt low priority

        // Change IVTBASE if required
        IVTBASEU = 0x00;                     // Optional
        IVTBASEH = 0x40;                     // Default is 0x0008
        IVTBASEL = 0x08;
}
```

REGISTER 14-11: SCANCON0: SCANNER ACCESS CONTROL REGISTER 0

R/W-0/0	R/W-0/0	R/W/HC-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R-0/0
EN	TRIGEN	SGO	—	—	MREG	BURSTMD	BUSY
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	HC = Bit is cleared by hardware

bit 7 **EN:** Scanner Enable bit(1)
1 = Scanner is enabled
0 = Scanner is disabled

bit 6 **TRIGEN:** Scanner Trigger Enable bit(2)
1 = Scanner trigger is enabled
0 = Scanner trigger is disabled
Refer Table 14-1.

bit 5 **SGO:** Scanner GO bit(3, 4)
1 = When the CRC is ready, the Memory region set by the MREG bit will be accessed and data is passed to the CRC peripheral.
0 = Scanner operations will not occur

bit 4-3 **Unimplemented:** Read as '0'

bit 2 **MREG:** Scanner Memory Region Select bit(2)
1 = Scanner address points to Data EEPROM
0 = Scanner address points to Program Flash Memory

bit 1 **BURSTMD:** Scanner Burst Mode bit
1 = Memory access request to the CPU Arbiter is always true
0 = Memory access request to the CPU Arbiter is dependent on the CRC request and Trigger
Refer Table 14-1.

bit 0 **BUSY:** Scanner Busy Indicator bit
1 = Scanner cycle is in process
0 = Scanner cycle is compete (or never started)

Note 1: Setting EN = 1 (SCANCON0 register) does not affect any other register content.
2: Scanner trigger selection can be set using the SCANTRIG register.
3: This bit can be cleared in software. It is cleared in hardware when LADR>HADR (and a data cycle is not occurring) or when CRCGO = 0 (CRCCON0 register).
4: CRCEN and CRCGO bits (CRCCON0 register) must be set before setting the SGO bit.

REGISTER 15-6: DMAxSSAU: DMAx SOURCE START ADDRESS UPPER REGISTER

U-0	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
—	—	SSA<21:16>					
bit 7							bit 0

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

bit 7-0 **SSA<21:16>:** Source Start Address bits

REGISTER 15-7: DMAxSPTRL: DMAx SOURCE POINTER LOW REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SPTR<7:0>							
bit 7							bit 0

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

bit 15-0 **SPTR<7:0>:** Current Source Address Pointer

REGISTER 15-8: DMAxSPTRH: DMAx SOURCE POINTER HIGH REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SPTR<15:8>							
bit 7							bit 0

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

bit 5-0 **SPTR<15:8>:** Current Source Address Pointer

21.1 Timer1/3/5 Operation

The Timer1/3/5 module is a 16-bit incrementing counter which is accessed through the TMRxH:TMRxL register pair. Writes to TMRxH or TMRxL directly update the counter.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or counter and increments on every selected edge of the external source.

Timer1/3/5 is enabled by configuring the ON and GE bits in the TxCON and TxGCON registers, respectively. Table 21-1 displays the Timer1/3/5 enable selections.

TABLE 21-1: TIMER1/3/5 ENABLE SELECTIONS

ON	GE	Timer1/3/5 Operation
1	1	Count Enabled
1	0	Always On
0	1	Off
0	0	Off

21.2 Clock Source Selection

The CS<4:0> bits of the TMRxCLK register (Register 21-3) are used to select the clock source for Timer1/3/5. The TxCLK register allows the selection of several possible synchronous and asynchronous clock sources. Register 21-3 displays the clock source selections.

21.2.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected the TMRxH:TMRxL register pair will increment on multiples of FOSC as determined by the Timer1/3/5 prescaler.

When the FOSC internal clock source is selected, the Timer1/3/5 register value will increment by four counts every instruction clock cycle. Due to this condition, a 2 LSB error in resolution will occur when reading the Timer1/3/5 value. To utilize the full resolution of Timer1/3/5, an asynchronous input signal must be used to gate the Timer1/3/5 clock input.

The following asynchronous sources may be used at the Timer1/3/5 gate:

- Asynchronous event on the TxGPPS pin
- TMR0OUT
- TMR1/3/5OUT (excluding the TMR for which it is being used)
- TMR 2/4/6OUT (postscaled)
- CMP1/2OUT
- SMT1_match
- NCO1OUT
- PWM3/4 OUT
- CCP1/2/3/4 OUT
- CLC1/2/3/4 OUT
- ZCDOUT

Note: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after any one or more of the following conditions:

- Timer1/3/5 enabled after POR
- Write to TMRxH or TMRxL
- Timer1/3/5 is disabled
- Timer1/3/5 is disabled (TMRxON = 0) when TxCKI is high then Timer1/3/5 is enabled (TMRxON = 1) when TxCKI is low.

21.2.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1/3/5 module may work as a timer or a counter.

When enabled to count, Timer1/3/5 is incremented on the rising edge of the external clock input of the TxCKIPPS pin. This external clock source can be synchronized to the microcontroller system clock or it can run asynchronously.

When used as a timer with a clock oscillator, an external 32.768 kHz crystal can be used in conjunction with the dedicated secondary internal oscillator circuit.

21.6.2 TIMER1/3/5 GATE SOURCE SELECTION

The gate source for Timer1/3/5 can be selected using the GSS<4:0> bits of the TMRxGATE register (Register 21-4). The polarity selection for the gate source is controlled by the TxGPOL bit of the TxGCON register (Register 21-2).

Any of the above mentioned signals can be used to trigger the gate. The output of the CMPx can be synchronized to the Timer1/3/5 clock or left asynchronous. For more information see **Section 38.3.1 "Comparator Output Synchronization"**.

21.6.3 TIMER1/3/5 GATE TOGGLE MODE

When Timer1/3/5 Gate Toggle mode is enabled, it is possible to measure the duration between every rising and falling edge of the gate signal.

The Timer1/3/5 gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. See Figure 21-5 for timing details.

Timer1/3/5 Gate Toggle mode is enabled by setting the GTM bit of the TxGCON register. When the GTM bit is cleared, the flip-flop is cleared and held clear. This is necessary in order to control which edge is measured.

Note:	Enabling Toggle mode at the same time as changing the gate polarity may result in indeterminate operation.

21.6.4 TIMER1/3/5 GATE SINGLE-PULSE MODE

When Timer1/3/5 Gate Single-Pulse mode is enabled, it is possible to capture a single-pulse gate event. Timer1/3/5 Gate Single-Pulse mode is first enabled by setting the GSPM bit in the TxGCON register. Next, the GGO/DONE bit in the TxGCON register must be set. The Timer1/3/5 will be fully enabled on the next incrementing edge of the gate signal. On the next trailing edge of the pulse, the GGO/DONE bit will automatically be cleared. No other gate events will be allowed to increment Timer1/3/5 until the GGO/DONE bit is once again set in software.

Clearing the TxGSPM bit of the TxGCON register will also clear the GGO/DONE bit. See Figure 21-6 for timing details.

Enabling the Toggle mode and the Single-Pulse mode simultaneously will permit both sections to work together. This allows the period on the Timer1/3/5 gate source to be measured. See Figure 21-7 for timing details.

21.6.5 TIMER1/3/5 GATE VALUE STATUS

When Timer1/3/5 Gate Value Status is utilized, it is possible to read the most current level of the gate signal. The value is stored in the GVAL bit in the TxGCON register. The GVAL bit is valid even when the Timer1/3/5 gate is not enabled (GE bit is cleared).

21.6.6 TIMER1/3/5 GATE EVENT INTERRUPT

When Timer1/3/5 Gate Event Interrupt is enabled, it is possible to generate an interrupt upon the completion of a gate event. When the falling edge of GVAL occurs, the TMRxGIF flag bit in the respective PIR register will be set. If the TMRxGIE bit in the respective PIE register is set, then an interrupt will be recognized.

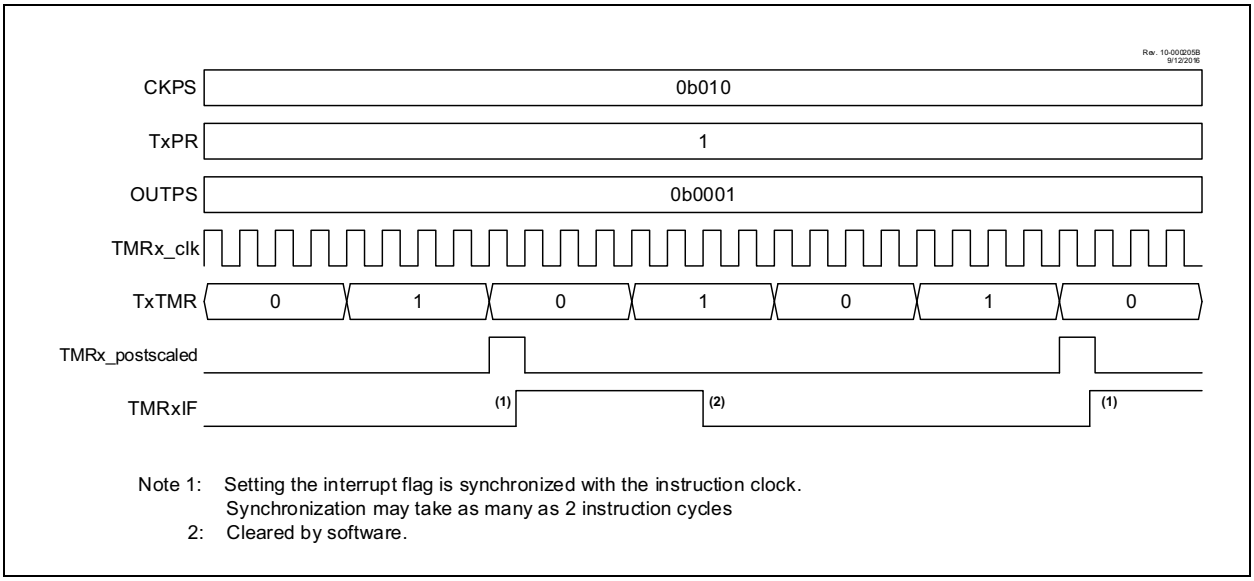
The TMRxGIF flag bit operates even when the Timer1/3/5 gate is not enabled (GE bit is cleared).

For more information on selecting high or low priority status for the Timer1/3/5 Gate Event Interrupt see **Section 9.0 "Interrupt Controller"**.

22.4 Timer2 Interrupt

Timer2 can also generate a device interrupt. The interrupt is generated when the postscaler counter matches one of 16 postscale options (from 1:1 through 1:16), which is selected with the postscaler control bits, OUTPS of the T2CON register. The interrupt is enabled by setting the T2TMR Interrupt Enable bit, TMR2IE, of the respective PIE register. The interrupt timing is illustrated in Figure 22-3.

FIGURE 22-3: TIMER2 PRESCALER, POSTSCALER, AND INTERRUPT TIMING DIAGRAM



Note 1: Setting the interrupt flag is synchronized with the instruction clock. Synchronization may take as many as 2 instruction cycles
2: Cleared by software.

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 CCPTMRS 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 CCPTMRS 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.

25.6.2 GATED TIMER MODE

Gated Timer mode uses the SMTSIGx input to control whether or not the SMT1TMR will increment. Upon a falling edge of the external signal, the SMT1CPW register will update to the current value of the SMT1TMR. Example waveforms for both repeated and single acquisitions are provided in Figure 25-4 and Figure 25-5.

REGISTER 25-10: SMT1CPRL: SMT CAPTURED PERIOD REGISTER – LOW BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMT1CPR<7:0>							
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 7-0 **SMT1CPR<7:0>**: Significant bits of the SMT Period Latch – Low Byte

REGISTER 25-11: SMT1CPRH: SMT CAPTURED PERIOD REGISTER – HIGH BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMT1CPR<15:8>							
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 7-0 **SMT1CPR<15:8>**: Significant bits of the SMT Period Latch – High Byte

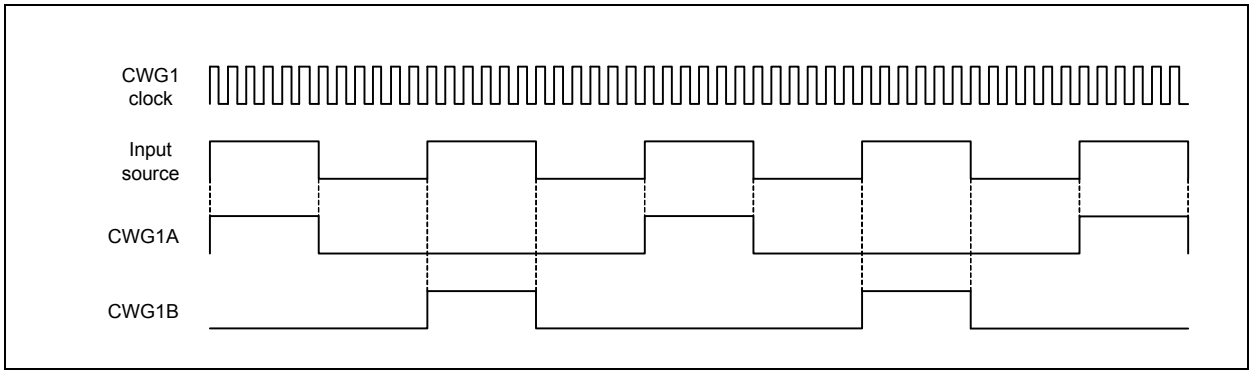
REGISTER 25-12: SMT1CPRU: SMT CAPTURED PERIOD REGISTER – UPPER BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMT1CPR<23:16>							
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 7-0 **SMT1CPR<23:16>**: Significant bits of the SMT Period Latch – Upper Byte

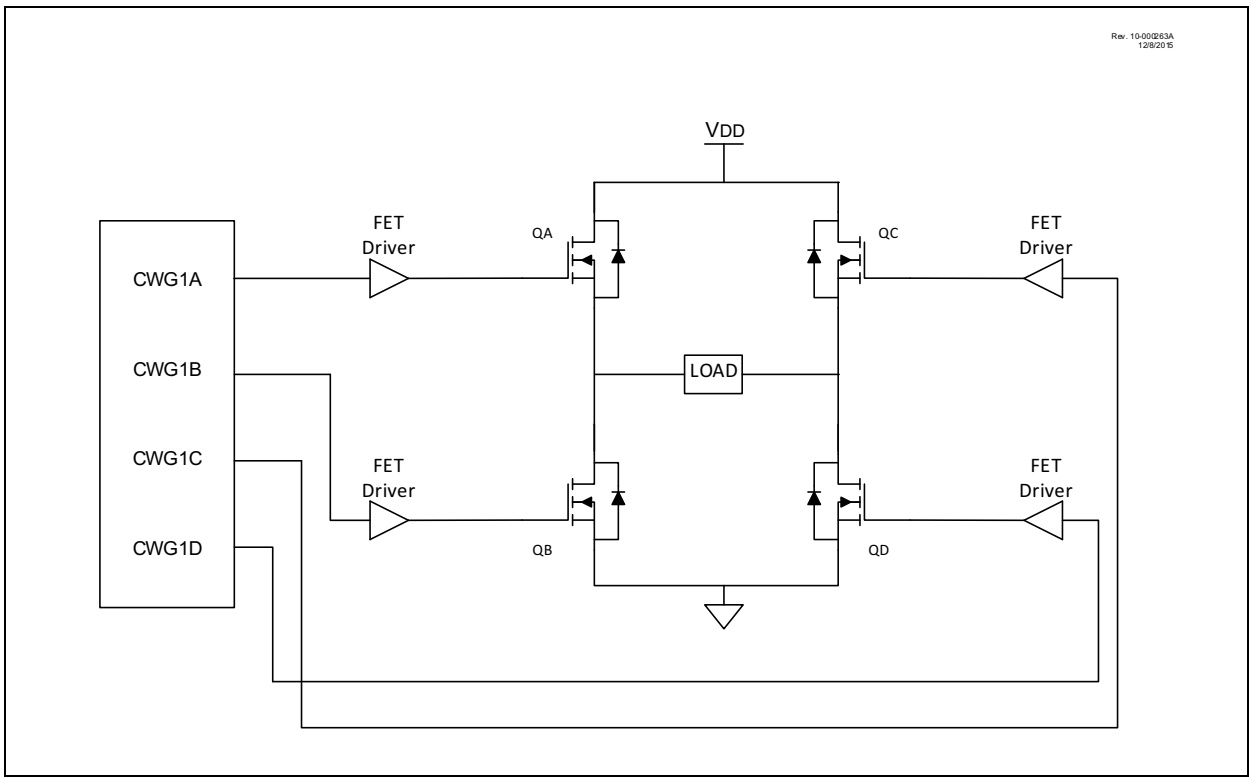
FIGURE 26-4: CWGx PUSH-PULL MODE OPERATION



26.2.3 FULL-BRIDGE MODES

In Forward and Reverse Full-Bridge modes, three outputs drive static values while the fourth is modulated by the input data signal. The mode selection may be toggled between forward and reverse by toggling the MODE<0> bit of the CWGxCON0 while keeping MODE<2:1> static, without disabling the CWG module. When connected as shown in Figure 26-5, the outputs are appropriate for a full-bridge motor driver. Each CWG output signal has independent polarity control, so the circuit can be adapted to high-active and low-active drivers. A simplified block diagram for the Full-Bridge modes is shown in Figure 26-6.

FIGURE 26-5: EXAMPLE OF FULL-BRIDGE APPLICATION



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

29.10 Register Definitions: ZCD Control

REGISTER 29-1: ZCDCON: ZERO-CROSS DETECT CONTROL REGISTER

R/W-0/0	U-0	R-x	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
SEN	—	OUT	POL	—	—	INTP	INTN
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 **SEN:** Zero-Cross Detect Software Enable bit
This bit is ignored when ZCDSEN configuration bit is set.
1= Zero-cross detect is enabled.
0= Zero-cross detect is disabled. ZCD pin operates according to PPS and TRIS controls.

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

bit 5 **OUT:** Zero-Cross Detect Data Output bit
ZCDPOL bit = 0:
1 = ZCD pin is sinking current
0 = ZCD pin is sourcing current
ZCDPOL bit = 1:
1 = ZCD pin is sourcing current
0 = ZCD pin is sinking current

bit 4 **POL:** Zero-Cross Detect Polarity bit
1 = ZCD logic output is inverted
0 = ZCD logic output is not inverted

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

bit 1 **INTP:** Zero-Cross Detect Positive-Going Edge Interrupt Enable bit
1 = ZCDIF bit is set on low-to-high ZCD_output transition
0 = ZCDIF bit is unaffected by low-to-high ZCD_output transition

bit 0 **INTN:** Zero-Cross Detect Negative-Going Edge Interrupt Enable bit
1 = ZCDIF bit is set on high-to-low ZCD_output transition
0 = ZCDIF bit is unaffected by high-to-low ZCD_output transition

TABLE 29-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE ZCD MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
ZCDCON	SEN	—	OUT	POL	—	—	INTP	INTN	462

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the ZCD module.

REGISTER 31-2: UxCON1: UART CONTROL REGISTER 1

R/W-0/0	U-0	U-0	R/W/HC-0/0	R/W-0/0	U-0	R/W-0/0	R/W/HC-0/0
ON	—	—	WUE	RXBIMD	—	BRKOVR	SENDB
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	HC = Hardware clear

bit 7 **ON:** Serial Port Enable bit
- 1 = Serial port enabled
- 0 = Serial port disabled (held in Reset)

bit 6-5 **Unimplemented:** Read as '0'

bit 4 **WUE:** Wake-up Enable bit
- 1 = Receiver is waiting for falling RX input edge which will set the UxIF bit. Cleared by hardware on wake event. Also requires UxIE bit of PIEx to enable wake
- 0 = Receiver operates normally

bit 3 **RXBIMD:** Receive Break Interrupt Mode Select bit
- 1 = Set RXBKIF immediately when RX in has been low for the minimum Break time
- 0 = Set RXBKIF on rising RX input after RX in has been low for the minimum Break time

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

bit 1 **BRKOVR:** Send Break Software Override bit
- 1 = TX output is forced to non-idle state
- 0 = TX output is driven by transmit shift register

bit 0 **SENDB:** Send Break Control bit(1)
- 1 = Output Break upon UxTXB write. Written byte follows Break. Bit is cleared by hardware.
- 0 = Break transmission completed or disabled

Note 1: This bit is read-only in LIN, DMX, and DALI modes.

FIGURE 32-7: CLOCKING DETAIL-MASTER MODE, CKE/SMP = 0/0

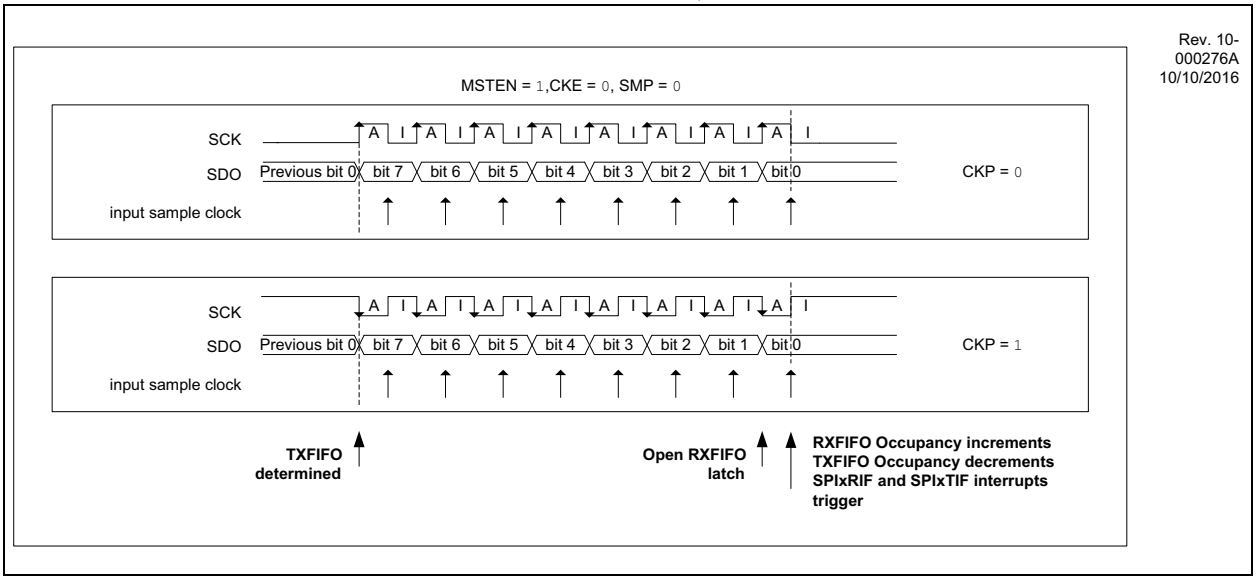
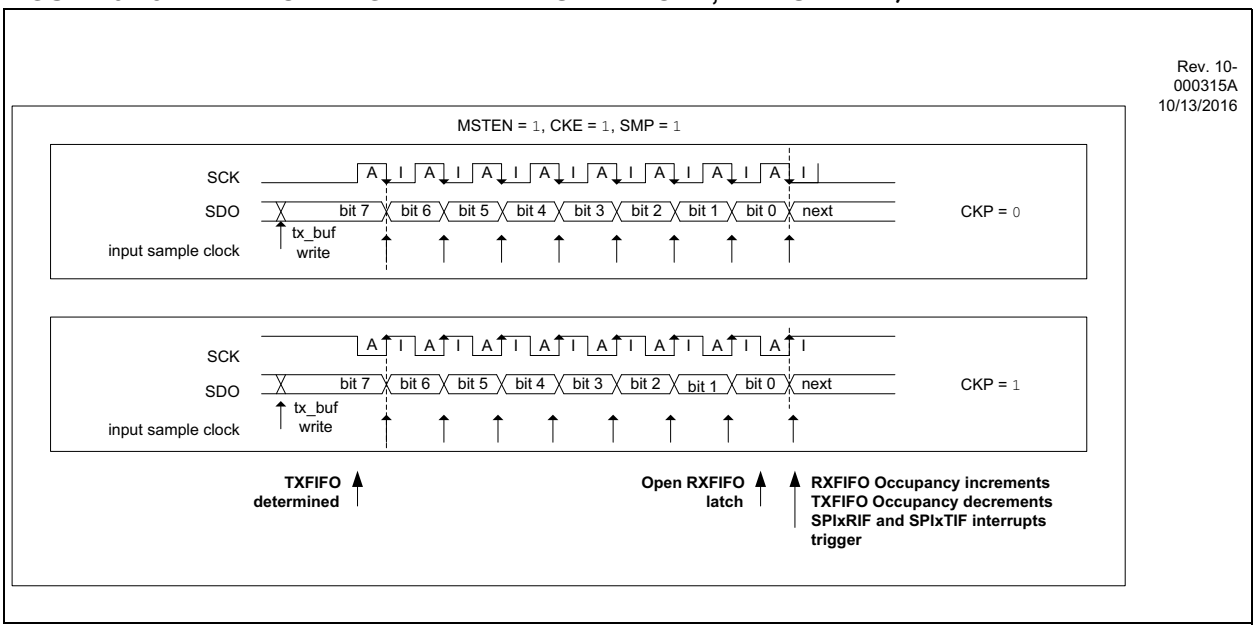


FIGURE 32-8: CLOCKING DETAIL - MASTER MODE, CKE/SMP = 1/1



REGISTER 33-8: I2CxERR: I²C ERROR REGISTER

U-0	R/W/HS-0	R/W/HS-0	R/W/HS-0	U-0	R/W-0	R/W-0	R/W-0
—	BTOIF(1,2)	BCLIF(1)	NACKIF(1)	—	BTOIE	BCLIE	NACKIE
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	HS = Hardware set HC = Hardware clear

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

bit 6 **BTOIF:** Bus Time-Out Interrupt Flag bit(1,2)
1 = Bus Timeout occurred
0 = No bus timeout

bit 5 **BCLIF:** Bus Collision Detect Interrupt Flag bit(1)
1 = Bus collision detected (On the rising edge of SCL input, SDA output is high and input is sampled low)
Slave and Master Mode the module immediately goes idle
Multi-Master Mode attempts to match slave addresses, and/or goes idle
0 = No bus collision detected

bit 4 **NACKIF:** NACK Detect Interrupt Flag bit(1)
1 = When (SMA = 1 || MMA = 1) and a NACK is detected on the bus
NACKIF is also set when any of the TXWRE, RXRDE, TXUF, RXOVR bits are set.
0 = No NACK/Error detected
NACKIF is **not** set by the NACK send for non-matching slave addresses

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

bit 2 **BTOIE:** Bus Time-Out Interrupt Enable bit
1 = Enable interrupt on bus time out
0 = Bus Tim-out not enabled

bit 1 **BCLIE:** Bus Collision Detect Interrupt Enable bit
1 = Enable interrupt on bus collision
0 = Bus collision interrupts are disabled

bit 0 **NACKIE:** NACK Detect Interrupt Enable bit
1 = Enable interrupt on NACKIF
0 = NACKIF interrupt is disabled

Note 1: Enabled error interrupt flags are OR'd to produce the PIRx<I2CEIF> bit.
2: User software must select the Bus Time-out Source in the I2CBTO register.

TABLE 44-4: POWER-DOWN CURRENT (I_{PD})(1,2)

PIC18LF26/45/46/55/56K42				Standard Operating Conditions (unless otherwise stated)					
PIC18F26/45/46/55/56K42				Standard Operating Conditions (unless otherwise stated) VREGPM = 1					
Param. No.	Symbol	Device Characteristics	Min.	Typ.†	Max. +85°C	Max. +125°C	Units	Conditions	
								V_{DD}	Note
D200	I_{PD}	I_{PD} Base	—	0.07	2	6	µA	3.0V	
D200 D200A	I_{PD}	I_{PD} Base	—	0.4	4	8	µA	3.0V	
			—	20	38	42	µA	3.0V	VREGPM = 0
D201	I_{PD_WDT}	Low-Frequency Internal Oscillator/ WDT	—	0.9	3.2	7	µA	3.0V	
D201	I_{PD_WDT}	Low-Frequency Internal Oscillator/ WDT	—	1.1	3.2	9	µA	3.0V	
D202	I_{PD_SOSC}	Secondary Oscillator (S_{OSC})	—	0.75	5	9	µA	3.0V	LP mode
D202	I_{PD_SOSC}	Secondary Oscillator (S_{OSC})	—	1.0	6.5	10	µA	3.0V	LP mode
D203	I_{PD_FVR}	FVR	—	45	74	75	µA	3.0V	FVRCON = 0x81 or 0x84
D203	I_{PD_FVR}	FVR	—	40	70	76	µA	3.0V	FVRCON = 0x81 or 0x84
D204	I_{PD_BOR}	Brown-out Reset (BOR)	—	9.4	14	18	µA	3.0V	
D204	I_{PD_BOR}	Brown-out Reset (BOR)	—	9.4	15	18	µA	3.0V	
D205	I_{PD_LPBOR}	Low-Power Brown-out Reset (LPBOR)	—	0.2	3	6	µA	3.0V	
D206	I_{PD_HLVD}	High/Low Voltage Detect (HLVD)	—	9.5	14.8	18	µA	3.0V	
D206	I_{PD_HLVD}	High/Low Voltage Detect (HLVD)	—	9.7	14.2	17	µA	3.0V	
D207	I_{PD_ADCA}	ADC - Non-Converting	—	0.1	2	6	µA	3.0V	ADC not converting (4)
D207	I_{PD_ADCA}	ADC - Non-Converting	—	0.1	4	8	µA	3.0V	ADC not converting (4)
D208	I_{PD_CMP}	Comparator	—	33	49	50	µA	3.0V	
D208	I_{PD_CMP}	Comparator	—	30	49	50	µA	3.0V	

† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The peripheral current is the sum of the base I_{DD} and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base I_{DD} or I_{PD} current from this limit. Max. values should be used when calculating total current consumption.

2: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode with all I/O pins in high-impedance state and tied to V_{SS}.

3: All peripheral currents listed are on a per-peripheral basis if more than one instance of a peripheral is available.

4: ADC clock source is FRC.