

Welcome to [E-XFL.COM](https://www.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, I <sup>2</sup> C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	25
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K 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	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic18lf25k83-i-sp">https://www.e-xfl.com/product-detail/microchip-technology/pic18lf25k83-i-sp</a>

**TABLE 4-5: SPECIAL FUNCTION REGISTER MAP FOR PIC18(L)F25/26K83 DEVICES BANK 61**

3DFFh	—	3DDFh	U2FIFO	3DBFh	—	3D9Fh	—	3D7Fh	—	3D5Fh	I2C2CON2	3D3Fh	—	3D1Fh	—
3DFEh	—	3DDEh	U2BRGH	3DBEh	—	3D9Eh	—	3D7Eh	—	3D5Eh	I2C2CON1	3D3Eh	—	3D1Eh	—
3DFDh	—	3DDDh	U2BRGL	3DBDh	—	3D9Dh	—	3D7Dh	—	3D5Dh	I2C2CON0	3D3Dh	—	3D1Dh	—
3DFCh	—	3DDCh	U2CON2	3DBCh	—	3D9Ch	—	3D7Ch	I2C1BTO	3D5Ch	I2C2ADR3	3D3Ch	—	3D1Ch	SPI1CLK
3DFBh	—	3DDBh	U2CON1	3DBBh	—	3D9Bh	—	3D7Bh	I2C1CLK	3D5Bh	I2C2ADR2	3D3Bh	—	3D1Bh	SPI1INTE
3DFAh	U1ERRIE	3DDAh	U2CON0	3DBAh	—	3D9Ah	—	3D7Ah	I2C1PIE	3D5Ah	I2C2ADR1	3D3Ah	—	3D1Ah	SPI1INTF
3DF9h	U1ERRIR	3DD9h	U2P3H	3DB9h	—	3D99h	—	3D79h	I2C1PIR	3D59h	I2C2ADR0	3D39h	—	3D19h	SPI1BAUD
3DF8h	U1UIR	3DD8h	U2P3L	3DB8h	—	3D98h	—	3D78h	I2C1STAT1	3D58h	I2C2ADB1	3D38h	—	3D18h	SPI1TWIDTH
3DF7h	U1FIFO	3DD7h	U2P2H	3DB7h	—	3D97h	—	3D77h	I2C1STAT0	3D57h	I2C2ADB0	3D37h	—	3D17h	SPI1STATUS
3DF6h	U1BRGH	3DD6h	U2P2L	3DB6h	—	3D96h	—	3D76h	I2C1ERR	3D56h	I2C2CNT	3D36h	—	3D16h	SPI1CON2
3DF5h	U1BRGL	3DD5h	U2P1H	3DB5h	—	3D95h	—	3D75h	I2C1CON2	3D55h	I2C2TXB	3D35h	—	3D15h	SPI1CON1
3DF4h	U1CON2	3DD4h	U2P1L	3DB4h	—	3D94h	—	3D74h	I2C1CON1	3D54h	I2C2RXB	3D34h	—	3D14h	SPI1CON0
3DF3h	U1CON1	3DD3h	U2TXCHK	3DB3h	—	3D93h	—	3D73h	I2C1CON0	3D53h	—	3D33h	—	3D13h	SPI1TCNTH
3DF2h	U1CON0	3DD2h	U2TXB	3DB2h	—	3D92h	—	3D72h	I2C1ADR3	3D52h	—	3D32h	—	3D12h	SPI1TCNTL
3DF1h	U1P3H	3DD1h	U2RXCHK	3DB1h	—	3D91h	—	3D71h	I2C1ADR2	3D51h	—	3D31h	—	3D11h	SPI1TXB
3DF0h	U1P3L	3DD0h	U2RXB	3DB0h	—	3D90h	—	3D70h	I2C1ADR1	3D50h	—	3D30h	—	3D10h	SPI1RXB
3DEFh	U1P2H	3DCFh	—	3DAFh	—	3D8Fh	—	3D6Fh	I2C1ADR0	3D4Fh	—	3D2Fh	—	3D0Fh	—
3DEEh	U1P2L	3DCEh	—	3DAEh	—	3D8Eh	—	3D6Eh	I2C1ADB1	3D4Eh	—	3D2Eh	—	3D0Eh	—
3DEDh	U1P1H	3DCDh	—	3DADh	—	3D8Dh	—	3D6Dh	I2C1ADB0	3D4Dh	—	3D2Dh	—	3D0Dh	—
3DECh	U1P1L	3DCCh	—	3DACH	—	3D8Ch	—	3D6Ch	I2C1CNT	3D4Ch	—	3D2Ch	—	3D0Ch	—
3DEBh	U1TXCHK	3DCBh	—	3DABh	—	3D8Bh	—	3D6Bh	I2C1TXB	3D4Bh	—	3D2Bh	—	3D0Bh	—
3DEAh	U1TXB	3DCAh	—	3DAAh	—	3D8Ah	—	3D6Ah	I2C1RXB	3D4Ah	—	3D2Ah	—	3D0Ah	—
3DE9h	U1RXCHK	3DC9h	—	3DA9h	—	3D89h	—	3D69h	—	3D49h	—	3D29h	—	3D09h	—
3DE8h	U1RXB	3DC8h	—	3DA8h	—	3D88h	—	3D68h	—	3D48h	—	3D28h	—	3D08h	—
3DE7h	—	3DC7h	—	3DA7h	—	3D87h	—	3D67h	—	3D47h	—	3D27h	—	3D07h	—
3DE6h	—	3DC6h	—	3DA6h	—	3D86h	—	3D66h	I2C2BTO	3D46h	—	3D26h	—	3D06h	—
3DE5h	—	3DC5h	—	3DA5h	—	3D85h	—	3D65h	I2C2CLK	3D45h	—	3D25h	—	3D05h	—
3DE4h	—	3DC4h	—	3DA4h	—	3D84h	—	3D64h	I2C2PIE	3D44h	—	3D24h	—	3D04h	—
3DE3h	—	3DC3h	—	3DA3h	—	3D83h	—	3D63h	I2C2PIR	3D43h	—	3D23h	—	3D03h	—
3DE2h	U2ERRIE	3DC2h	—	3DA2h	—	3D82h	—	3D62h	I2C2STAT1	3D42h	—	3D22h	—	3D02h	—
3DE1h	U2ERRIR	3DC1h	—	3DA1h	—	3D81h	—	3D61h	I2C2STAT0	3D41h	—	3D21h	—	3D01h	—
3DE0h	U2UIR	3DC0h	—	3DA0h	—	3D80h	—	3D60h	I2C2ERR	3D40h	—	3D20h	—	3D00h	—

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

# PIC18(L)F25/26K83

## REGISTER 5-5: CONFIGURATION WORD 3L (30 0004h)

U-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	WDTE<1:0>		WDTCPs<4:0>				
bit 7							bit 0

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '1'

-n = Value for blank device

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7 **Unimplemented:** Read as '1'

bit 6-5 **WDTE<1:0>:** WDT Operating Mode bits

00 = WDT is disabled, SWDTEN is ignored

01 = WDT is enabled/disabled by the SWDTEN bit in WDTCON0

10 = WDT is enabled while Sleep = 0, suspended when Sleep = 1; SWDTEN is ignored

11 = WDT is enabled regardless of Sleep; SWDTEN is ignored

bit 4-0 **WDTCPs<4:0>:** WDT Period Select bits

WDTCPs<4:0>	WDTPS at POR				Software Control of WDTPS?
	Value	Divider Ratio		Typical Time-out (F <sub>IN</sub> = 31 kHz)	
00000	00000	1:32	2 <sup>5</sup>	1 ms	No
00001	00001	1:64	2 <sup>6</sup>	2 ms	
00010	00010	1:128	2 <sup>7</sup>	4 ms	
00011	00011	1:256	2 <sup>8</sup>	8 ms	
00100	00100	1:512	2 <sup>9</sup>	16 ms	
00101	00101	1:1024	2 <sup>10</sup>	32 ms	
00110	00110	1:2048	2 <sup>11</sup>	64 ms	
00111	00111	1:4096	2 <sup>12</sup>	128 ms	
01000	01000	1:8192	2 <sup>13</sup>	256 ms	
01001	01001	1:16384	2 <sup>14</sup>	512 ms	
01010	01010	1:32768	2 <sup>15</sup>	1s	
01011	01011	1:65536	2 <sup>16</sup>	2s	
01100	01100	1:131072	2 <sup>17</sup>	4s	
01101	01101	1:262144	2 <sup>18</sup>	8s	
01110	01110	1:524299	2 <sup>19</sup>	16s	
01111	01111	1:1048576	2 <sup>20</sup>	32s	
10000	10000	1:2097152	2 <sup>21</sup>	64s	
10001	10001	1:4194304	2 <sup>22</sup>	128s	
10010	10010	1:8388608	2 <sup>23</sup>	256s	
10011	10011	1:32	2 <sup>5</sup>	1 ms	No
...	...				
11110	11110	1:65536	2 <sup>16</sup>	2s	Yes
11111	01011				

## REGISTER 7-2: OSCCON2: OSCILLATOR CONTROL REGISTER 2

U-0	R-f/f	R-f/f	R-f/f	R-f/f	R-f/f	R-f/f	R-f/f
—	COSC<2:0>			CDIV<3:0>			
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 **Unimplemented:** Read as '0'

bit 6-4 **COSC<2:0>:** Current Oscillator Source Select bits (read-only)<sup>(1)</sup>  
Indicates the current source oscillator and PLL combination per Table 7-1.

bit 3-0 **CDIV<3:0>:** Current Divider Select bits (read-only)<sup>(1)</sup>  
Indicates the current postscaler division ratio per Table 7-1.

**Note 1:** The POR value is the value present when user code execution begins.

## REGISTER 7-3: OSCCON3: OSCILLATOR CONTROL REGISTER 3

R/W/HC-0/0	R/W-0/0	U-0	R-0/0	R-0/0	U-0	U-0	U-0
CSWHOLD	SOSCPWR	—	ORDY	NOSCR	—	—	—
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 **CSWHOLD:** Clock Switch Hold bit  
1 = Clock switch will hold (with interrupt) when the oscillator selected by NOSC is ready  
0 = Clock switch may proceed when the oscillator selected by NOSC is ready; NOSC becomes '1', the switch will occur

bit 6 **SOSCPWR:** Secondary Oscillator Power Mode Select bit  
1 = Secondary oscillator operating in High-Power mode  
0 = Secondary oscillator operating in Low-Power mode

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

bit 4 **ORDY:** Oscillator Ready bit (read-only)  
1 = OSCCON1 = OSCCON2; the current system clock is the clock specified by NOSC  
0 = A clock switch is in progress

bit 3 **NOSCR:** New Oscillator is Ready bit (read-only)<sup>(1)</sup>  
1 = A clock switch is in progress and the oscillator selected by NOSC indicates a "ready" condition  
0 = A clock switch is not in progress, or the NOSC-selected oscillator is not yet ready

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

**Note 1:** If CSWHOLD = 0, the user may not see this bit set because, when the oscillator becomes ready there may be a delay of one instruction clock before this bit is set. The clock switch occurs in the next instruction cycle and this bit is cleared.

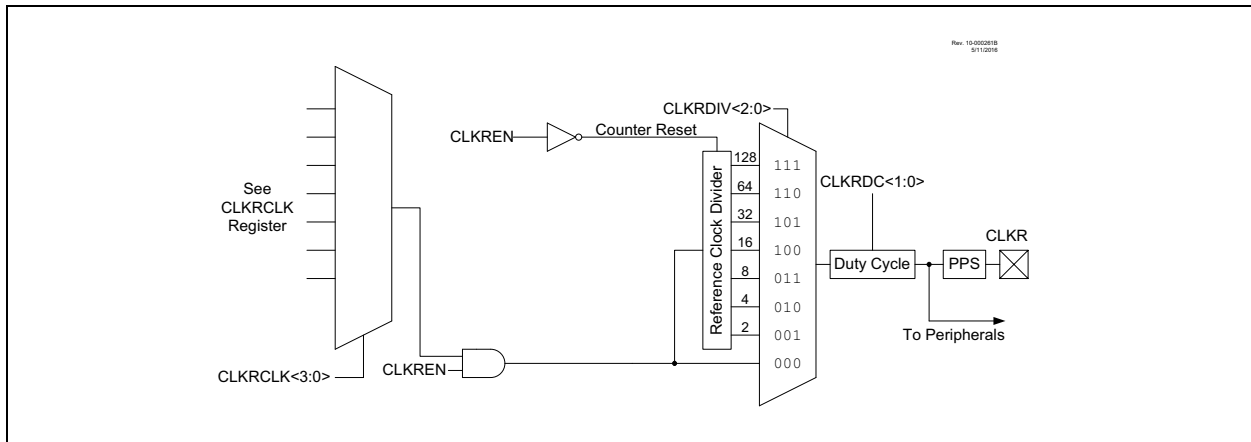
## 8.0 REFERENCE CLOCK OUTPUT MODULE

The reference clock output module provides the ability to send a clock signal to the clock reference output pin (CLKR). The reference clock output can also be used as a signal for other peripherals, such as the Data Signal Modulator (DSM), Memory Scanner and Timer module.

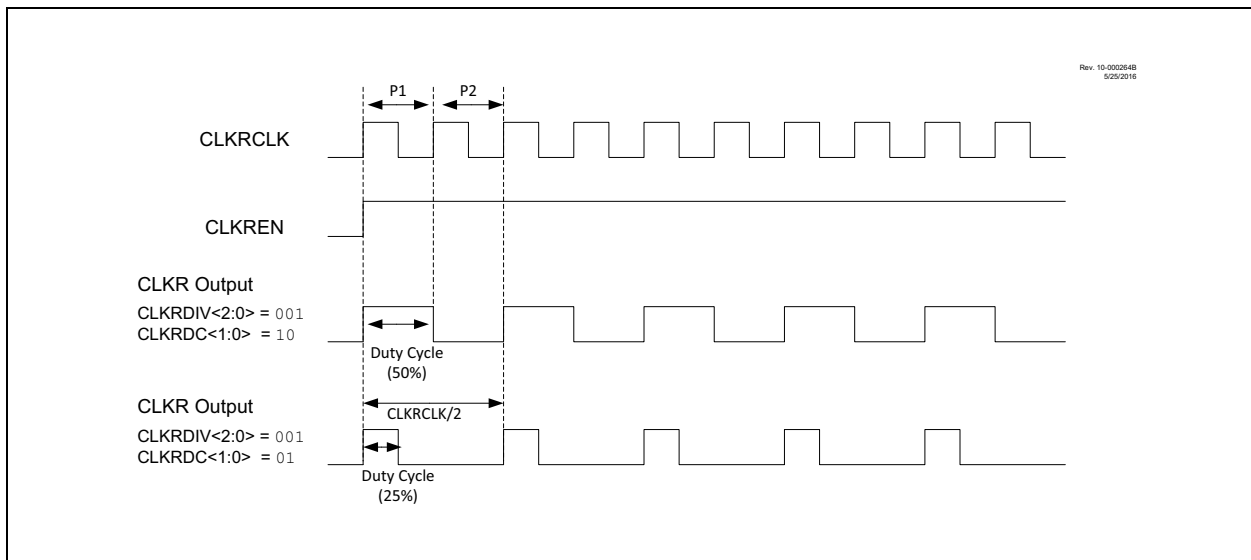
The reference clock output module has the following features:

- Selectable clock source using the CLKRCLK register
- Programmable clock divider
- Selectable duty cycle

**FIGURE 8-1: CLOCK REFERENCE BLOCK DIAGRAM**



**FIGURE 8-2: CLOCK REFERENCE TIMING**



## 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;
}
```

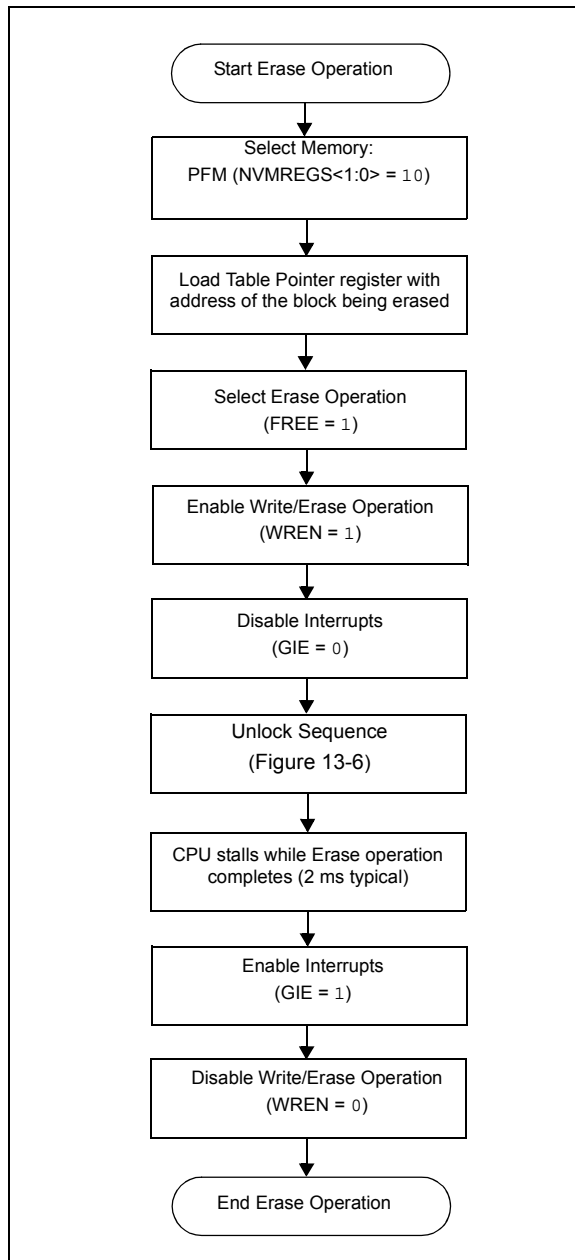
# PIC18(L)F25/26K83

**TABLE 9-3: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON0	GIE/GIEH	GIEL	IPEN	—	—	INT2EDG	INT1EDG	INT0EDG	125
INTCON1	STAT<1:0>		—	—	—	—	—	—	126
PIE0	IOCIE	CRCIE	SCANIE	NVMIE	CSWIE	OSFIE	HLVDIE	SWIE	137
PIE1	SMT1PWAIE	SMT1PRAIE	SMT1IE	C1IE	ADTIE	ADIE	ZCDIE	INT0IE	138
PIE2	I2C1RXIE	SPI1IE	SPI1TXIE	SPI1RXIE	DMA1AIE	DMA1ORIE	DMA1DCNTIE	DMA1SCNTIE	139
PIE3	TMR0IE	U1IE	U1EIE	U1TXIE	U1RXIE	I2C1EIE	I2C1IE	I2C1TXIE	140
PIE4	INT1IE	CLC1IE	CWG1IE	NCO1IE	CCP1IE	TMR2IE	TMR1GIE	TMR1IFE	141
PIE5	IRXIE	WAKIE	ERRIE	TXB2IE/TXBnIE	TXB1IE	TXB0IE	RXB1IE/RXBnIE	RXB0IE/FIFOIE	142
PIE6	DMA2AIE	DMA2ORIE	DMA2DCNTIE	DMA2SCNTIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	C2IE	143
PIE7	U2IE	U2EIE	U2TXIE	U2RXIE	I2C2EIE	I2C2IE	I2C2TXIE	I2C2RXIE	144
PIE8	TMR5IE	INT2IE	CLC2IE	CWG2IE	CCP2IE	TMR4IE	TMR3GIE	TMR3IE	145
PIE9	—	CLC4IE	CCP4IE	CLC3IE	CWG3IE	CCP3IE	TMR6IE	TMR5IE	146
PIR0	IOCIF	CRCIF	SCANIF	NVMIF	CSWIF	OSFIF	HLVDIF	SWIF	127
PIR1	SMT1PWAIF	SMT1PRAIF	SMT1IF	C1IF	ADTIF	ADIF	ZCDIF	INT0IF	128
PIR2	I2C1RXIF	SPI1IF	SPI1TXIF	SPI1RXIF	DMA1AIF	DMA1ORIF	DMA1DCNTIF	DMA1SCNTIF	129
PIR3	TMR0IF	U1IF	U1EIF	U1TXIF	U1RXIF	I2C1EIF	I2C1IF	I2C1TXIF	130
PIR4	INT1IF	CLC1IF	CWG1IF	NCO1IF	CCP1IF	TMR2IF	TMR1GIF	TMR1IF	131
PIR5	IRXIF	WAKIF	ERRIF	TXB2IF/TXBnIF	TXB1IF	TXB0IF	RXB1IF/RXBnIF	RXB0IF/FIFOIF	132
PIR6	DMA2AIF	DMA2ORIF	DMA2DCNTIF	DMA2SCNTIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	C2IF	133
PIR7	U2IF	U2EIF	U2TXIF	U2RXIF	I2C2EIF	I2C2IF	I2C2TXIF	I2C2RXIF	134
PIR8	TMR5IF	INT2IF	CLC2IF	CWG2IF	CCP2IF	TMR4IF	TMR3GIF	TMR3IF	135
PIR9	—	CLC4IF	CCP4IF	CLC3IF	CWG3IF	CCP3IF	TMR6IF	TMR5IF	136
IPR0	IOCIP	CRCIP	SCANIP	NVMIP	CSWIP	OSFIP	HLVDIP	SWIP	147
IPR1	SMT1PWAIP	SMT1PRAIP	SMT1IP	C1IP	ADTIP	ADIP	ZCDIP	INT0IP	148
IPR2	I2C1RIP	SPI1IP	SPI1TIP	SPI1RIP	DMA1AIP	DMA1ORIP	DMA1DCNTIP	DMA1SCNTIP	149
IPR3	TMR0IP	U1IP	U1EIP	U1TXIP	U1RXIP	I2C1EIP	I2C1IP	I2C1TXIP	150
IPR4	INT1IP	CLC1IP	CWG1IP	NCO1IP	CCP1IP	TMR2IP	TMR1GIP	TMR1IP	151
IPR5	IRXIP	WAKIP	ERRIP	TXB2IP/TXBnIP	TXB1IP	TXB0IP	RXB1IP/RXBnIP	RXB0IP/FIFOIP	152
IPR6	DMA2AIP	DMA2ORIP	DMA2DCNTIP	DMA2SCNTIP	SMT2PWAIP	SMT2PRAIP	SMT2IP	C2IP	153
IPR7	U2IP	U2EIP	U2TXIP	U2RXIP	I2C2EIP	I2C2IP	I2C2TXIP	I2C2RXIP	154
IPR8	TMR5IP	INT2IP	CLC2IP	CWG2IP	CCP2IP	TMR4IP	TMR3GIP	TMR3IP	155
IPR9	—	CLC4IP	CCP4IP	CLC3IP	CWG3IP	CCP3IP	TMR6IP	TMR5IP	156
IVTBASEU	—	—	—	BASE<20:16>					157
IVTBASEH	BASE<15:8>								157
IVTBASEL	BASE<7:0>								157
IVTADU	—	—	—	AD<20:16>					158
IVTADH	AD<15:8>								158
IVTADL	AD<7:0>								158
IVTLOCK	—	—	—	—	—	—	—	IVTLOCKED	159

**Legend:** — = unimplemented locations, read as '0'. Shaded bits are not used for interrupts.

**FIGURE 13-7: PFM ROW ERASE FLOWCHART**



## 13.1.6 WRITING TO PROGRAM FLASH MEMORY

The programming write block size is described in Table 5-4. Word or byte programming is not supported. Table writes are used internally to load the holding registers needed to program the memory. There are only as many holding registers as there are bytes in a write block. Refer to Table 5-4 for write latch size.

Since the table latch (TABLAT) is only a single byte, the TBLWT instruction needs to be executed multiple times for each programming operation. The write protection state is ignored for this operation. All of the table write operations will essentially be short writes because only the holding registers are written. NVMIF is not affected while writing to the holding registers.

After all the holding registers have been written, the programming operation of that block of memory is started by configuring the NVMCON1 register for a program memory write and performing the long write sequence.

If the PFM address in the TBLPTR is write-protected or if TBLPTR points to an invalid location, the WR bit is cleared without any effect and the WREER is signaled.

The long write is necessary for programming the program memory. CPU operation is suspended during a long write cycle and resumes when the operation is complete. The long write operation completes in one instruction cycle. When complete, WR is cleared in hardware and NVMIF is set and an interrupt will occur if NVMIE is also set. The latched data is reset to all '1s'. WREN is not changed.

The internal programming timer controls the write time. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device.

**Note:** The default value of the holding registers on device Resets and after write operations is FFh. A write of FFh to a holding register does not modify that byte. This means that individual bytes of program memory may be modified, provided that the change does not attempt to change any bit from a '0' to a '1'. When modifying individual bytes, it is not necessary to load all holding registers before executing a long write operation.



## 15.10 Reset

The DMA registers are set to the default state on any Reset. The registers are also reset to the default state when the enable bit is cleared (DMA1CON1bits.EN=0).

## 15.11 Power Saving Mode Operation

The DMA utilizes system clocks and it is treated as a peripheral when it comes to power-saving operations. Like other peripherals, the DMA also uses Peripheral Module Disable bits to further tailor its operation in low-power states.

### 15.11.1 SLEEP MODE

When the device enters Sleep mode, the system clock to the module is shut down, therefore no DMA operation is supported in Sleep. Once the system clock is disabled, the requisite read and write clocks are also disabled without which the DMA cannot perform any of its tasks.

Any transfers that may be in progress are resumed on exiting from Sleep mode. Register contents are not affected by the device entering or leaving Sleep mode. It is recommended that DMA transactions be allowed to finish before entering Sleep mode.

### 15.11.2 IDLE MODE

In IDLE mode, all of the system clocks (including the read and write clocks) are still operating but the CPU is not using them to save power.

Therefore, every instruction cycle is available to the system arbiter and if the bubble is granted to the DMA, it may be utilized to move data.

### 15.11.3 DOZE MODE

Similar to the Idle mode, the CPU does not utilize all of the available instruction cycles slots that are available to it in order to save power. It only executes instructions based on its settings from the Doze settings.

Therefore, every instruction not used by the CPU is available for system arbitration and may be utilized by the DMA if granted by the arbiter.

### 15.11.4 PERIPHERAL MODULE DISABLE

The Peripheral Module Disable (PMD) registers provide a method to disable DMA by gating all clock sources supplied to it. The respective DMAxMD bit needs to be set in order to disable the DMA.

## 15.12 DMA Register Interfaces

The DMA can transfer data to any GPR or SFR location. For better user accessibility some of the more commonly used SFR spaces have their Mirror registers placed in Bank 64 (0x4000-0x40FF), these Mirror registers can be only accessed through the DMA Source and Destination Address registers.

## 17.8 Register Definitions: PPS Input Selection

**REGISTER 17-1: xxxPPS: PERIPHERAL xxx INPUT SELECTION**

U-0	U-0	R/W-m/u <sup>(1)</sup>	R/W-m/u <sup>(1)</sup>	R/W-m/u <sup>(1)</sup>	R/W-m/u <sup>(1)</sup>	R/W-m/u <sup>(1)</sup>	R/W-m/u <sup>(1)</sup>
—	—	xxxPPS<5:0>					
bit 7							
							bit 0

**Legend:**

R = Readable bit	W = Writable bit	-n/n = Value at POR and BOR/Value at all other Resets
u = Bit is unchanged	x = Bit is unknown	q = value depends on peripheral
'1' = Bit is set	U = Unimplemented bit, read as '0'	m = value depends on default location for that input
'0' = Bit is cleared		

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

bit 5-3 **xxxPPS<5:3>:** Peripheral xxx Input PORTx Pin Selection bits  
See Table 17-1 for the list of available ports and default pin locations.

- 101 = Reserved
- 100 = Reserved
- 011 = Reserved
- 010 = PORTC
- 001 = PORTB
- 000 = PORTA

bit 2-0 **xxxPPS<2:0>:** Peripheral xxx Input PORTx Pin Selection bits

- 111 = Peripheral input is from PORTx Pin 7 (Rx7)
- 110 = Peripheral input is from PORTx Pin 6 (Rx6)
- 101 = Peripheral input is from PORTx Pin 5 (Rx5)
- 100 = Peripheral input is from PORTx Pin 4 (Rx4)
- 011 = Peripheral input is from PORTx Pin 3 (Rx3)
- 010 = Peripheral input is from PORTx Pin 2 (Rx2)
- 001 = Peripheral input is from PORTx Pin 1 (Rx1)
- 000 = Peripheral input is from PORTx Pin 0 (Rx0)

**Note 1:** The Reset value 'm' of this register is determined by device default locations for that input.

## REGISTER 19-6: PMD5: PMD CONTROL REGISTER 5

U-0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	U2MD	U1MD	—	SPI1MD	I2C2MD	I2C1MD
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 7-6	<b>Unimplemented:</b> Read as '0'
bit 5	<b>U2MD:</b> Disable UART2 bit 1 = UART2 module disabled 0 = UART2 module enabled
bit 4	<b>U1MD:</b> Disable UART1 bit 1 = UART1 module disabled 0 = UART1 module enabled
bit 3	<b>Unimplemented:</b> Read as '0'
bit 2	<b>SPI1MD:</b> Disable SPI1 Module bit 1 = SPI1 module disabled 0 = SPI1 module enabled
bit 1	<b>I2C2MD:</b> Disable I <sup>2</sup> C2 Module bit 1 = I <sup>2</sup> C2 module disabled 0 = I <sup>2</sup> C2 module enabled
bit 0	<b>I2C1MD:</b> Disable I <sup>2</sup> C1 Module bit 1 = I <sup>2</sup> C1 module disabled 0 = I <sup>2</sup> C1 module enabled

22.5.7 EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODE

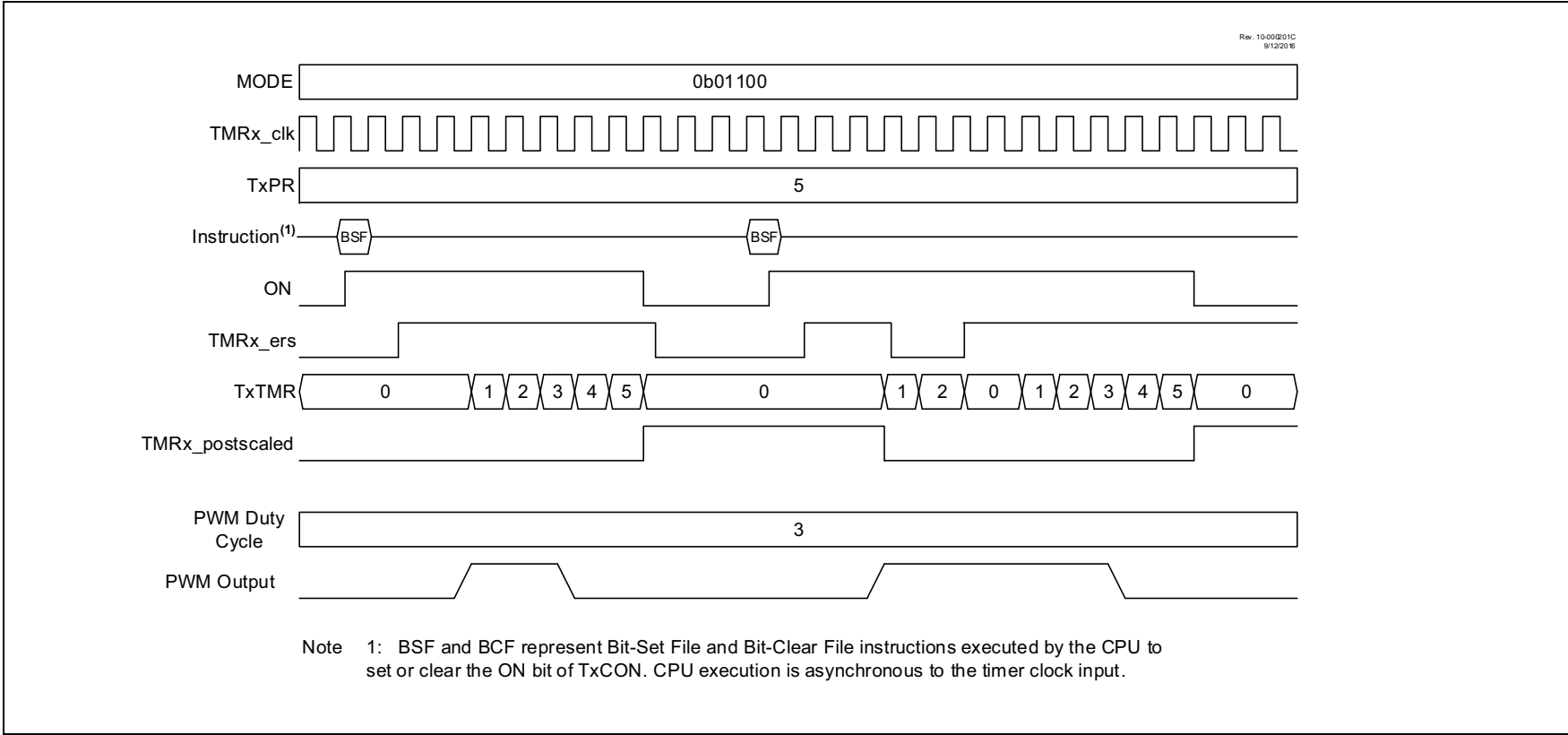
In Edge-Triggered Hardware Limit One-Shot modes the timer starts on the first external signal edge after the ON bit is set and resets on all subsequent edges. Only the first edge after the ON bit is set is needed to start the timer. The counter will resume counting automatically two clocks after all subsequent external Reset edges. Edge triggers are as follows:

- Rising edge Start and Reset (MODE<4:0> = 01100)
- Falling edge Start and Reset (MODE<4:0> = 01101)

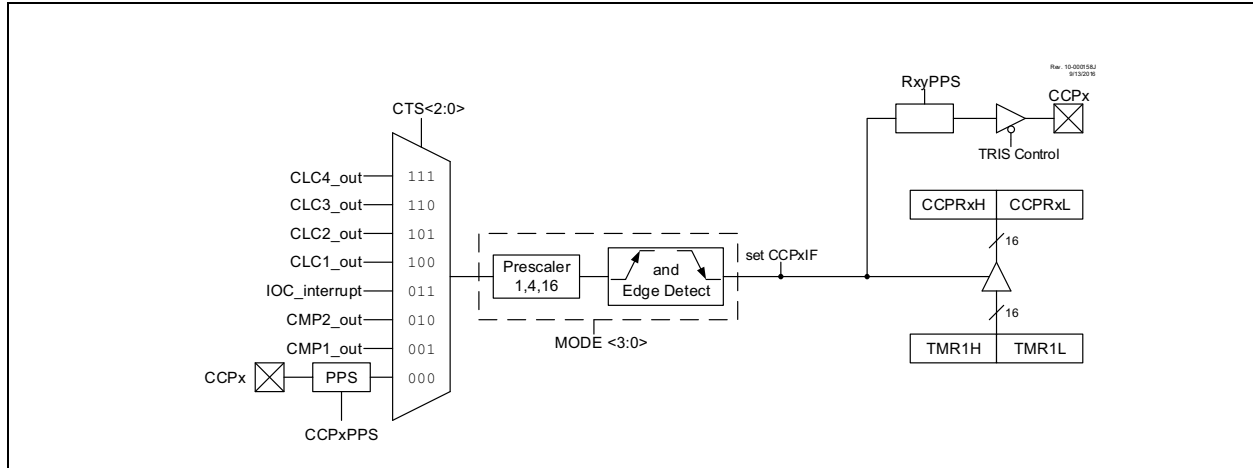
The timer resets and clears the ON bit when the timer value matches the T2PR period value. External signal edges will have no effect until after software sets the ON bit. Figure 22-10 illustrates the rising edge hardware limit one-shot operation.

When this mode is used in conjunction with the CCP then the first starting edge trigger, and all subsequent Reset edges, will activate the PWM drive. The PWM drive will deactivate when the timer matches the CCPRx pulse width value and stay deactivated until the timer halts at the T2PR period match unless an external signal edge resets the timer before the match occurs.

FIGURE 22-10: EDGE TRIGGERED HARDWARE LIMIT ONE-SHOT MODE TIMING DIAGRAM (MODE = 01100)



**FIGURE 23-1: CAPTURE MODE OPERATION BLOCK DIAGRAM**



## 23.4.2 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for standard PWM operation:

1. Use the desired output pin RxyPPS control to select CCPx as the source and disable the CCPx pin output driver by setting the associated TRIS bit.
2. Load the T2PR register with the PWM period value.
3. Configure the CCP module for the PWM mode by loading the CCPxCON register with the appropriate values.
4. Load the CCPRxL register, and the CCPRxH register with the PWM duty cycle value and configure the FMT bit of the CCPxCON register to set the proper register alignment.
5. Configure and start Timer2:
  - Clear the TMR2IF interrupt flag bit of the respective PIR register. See Note below.
  - Select the timer clock source to be as Fosc/4 using the T2CLK register. This is required for correct operation of the PWM module.
  - Configure the CKPS bits of the T2CON register with the Timer prescale value.
  - Enable the Timer by setting the ON bit of the T2CON register.
6. Enable PWM output pin:
  - Wait until the Timer overflows and the TMR2IF bit of the PIR4 register is set. See Note below.
  - Enable the CCPx pin output driver by clearing the associated TRIS bit.

**Note:** In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then step 6 may be ignored.

## 23.4.3 TIMER2 TIMER RESOURCE

The PWM standard mode makes use of the 8-bit Timer2 timer resources to specify the PWM period.

## 23.4.4 PWM PERIOD

The PWM period is specified by the T2PR register of Timer2. The PWM period can be calculated using the formula of Equation 23-1.

### EQUATION 23-1: PWM PERIOD

$$PWM\ Period = [(T2PR) + 1] \cdot 4 \cdot T_{OSC} \cdot (TMR2\ Prescale\ Value)$$

**Note 1:**  $T_{OSC} = 1/F_{OSC}$

When T2TMR is equal to T2PR, the following three events occur on the next increment cycle:

- T2TMR is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is transferred from the CCPRxL/H register pair into a 10-bit buffer.

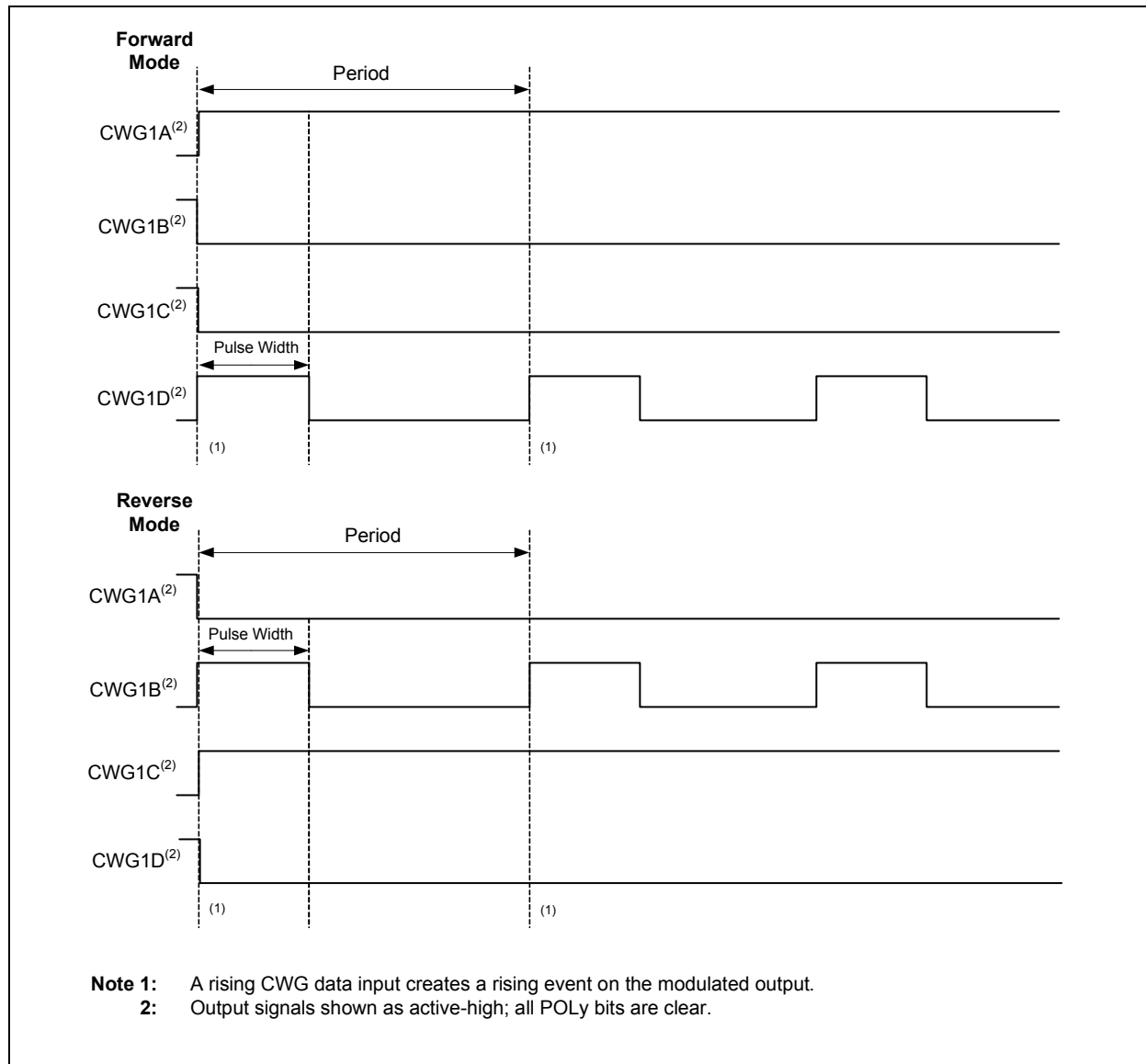
**Note:** The Timer postscaler (see **Section 22.3 “External Reset Sources”**) is not used in the determination of the PWM frequency.

In Forward Full-Bridge mode ( $\text{MODE}\langle 2:0 \rangle = 010$ ), CWGxA is driven to its active state, CWGxB and CWGxC are driven to their inactive state, and CWGxD is modulated by the input signal, as shown in Figure 26-7.

In Reverse Full-Bridge mode ( $\text{MODE}\langle 2:0 \rangle = 011$ ), CWGxC is driven to its active state, CWGxA and CWGxD are driven to their inactive states, and CWGxB is modulated by the input signal, as shown in Figure 26-7.

In Full-Bridge mode, the dead-band period is used when there is a switch from forward to reverse or vice-versa. This dead-band control is described in **Section 26.6 “Dead-Band Control”**, with additional details in **Section 26.7 “Rising Edge and Reverse Dead Band”** and **Section 26.8 “Falling Edge and Forward Dead Band”**. Steering modes are not used with either of the Full-Bridge modes. The mode selection may be toggled between forward and reverse toggling the  $\text{MODE}\langle 0 \rangle$  bit of the CWGxCON0 while keeping  $\text{MODE}\langle 2:1 \rangle$  static, without disabling the CWG module.

**FIGURE 26-7: EXAMPLE OF FULL-BRIDGE OUTPUT**



## 27.7 Register Definitions: CLC Control

**REGISTER 27-1: CLCxCON: CONFIGURABLE LOGIC CELL CONTROL REGISTER**

R/W-0/0	U-0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	—	OUT	INTP	INTN	MODE<2:0>		
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      **EN:** Configurable Logic Cell Enable bit  
1 = Configurable logic cell is enabled and mixing input signals  
0 = Configurable logic cell is disabled and has logic zero output
- bit 6      **Unimplemented:** Read as '0'
- bit 5      **OUT:** Configurable Logic Cell Data Output bit  
Read-only: logic cell output data, after LCPOL; sampled from CLCxOUT
- bit 4      **INTP:** Configurable Logic Cell Positive Edge Going Interrupt Enable bit  
1 = CLCxIF will be set when a rising edge occurs on CLCxOUT  
0 = CLCxIF will not be set
- bit 3      **INTN:** Configurable Logic Cell Negative Edge Going Interrupt Enable bit  
1 = CLCxIF will be set when a falling edge occurs on CLCxOUT  
0 = CLCxIF will not be set
- bit 2-0    **MODE<2:0>:** Configurable Logic Cell Functional Mode bits  
111 = Cell is 1-input transparent latch with S and R  
110 = Cell is J-K flip-flop with R  
101 = Cell is 2-input D flip-flop with R  
100 = Cell is 1-input D flip-flop with S and R  
011 = Cell is S-R latch  
010 = Cell is 4-input AND  
001 = Cell is OR-XOR  
000 = Cell is AND-OR



## REGISTER 31-18: UxTXCHK: UART TRANSMIT CHECKSUM RESULT REGISTER

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	R/W-0/0
TXCHK<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 **TXCHK<7:0>**: Checksum calculated from TX bytes

LIN mode and C0EN = 1:

Sum of all transmitted bytes including PID

LIN mode and C0EN = 0:

Sum of all transmitted bytes except PID

All other modes and C0EN = 1:

Sum of all transmitted bytes since last clear

All other modes and C0EN = 0:

Not used

## REGISTER 31-19: UxRXCHK: UART RECEIVE CHECKSUM RESULT REGISTER

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	R/W-0/0
RXCHK<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 **RXCHK<7:0>**: Checksum calculated from RX bytes

LIN mode and C0EN = 1:

Sum of all received bytes including PID

LIN mode and C0EN = 0:

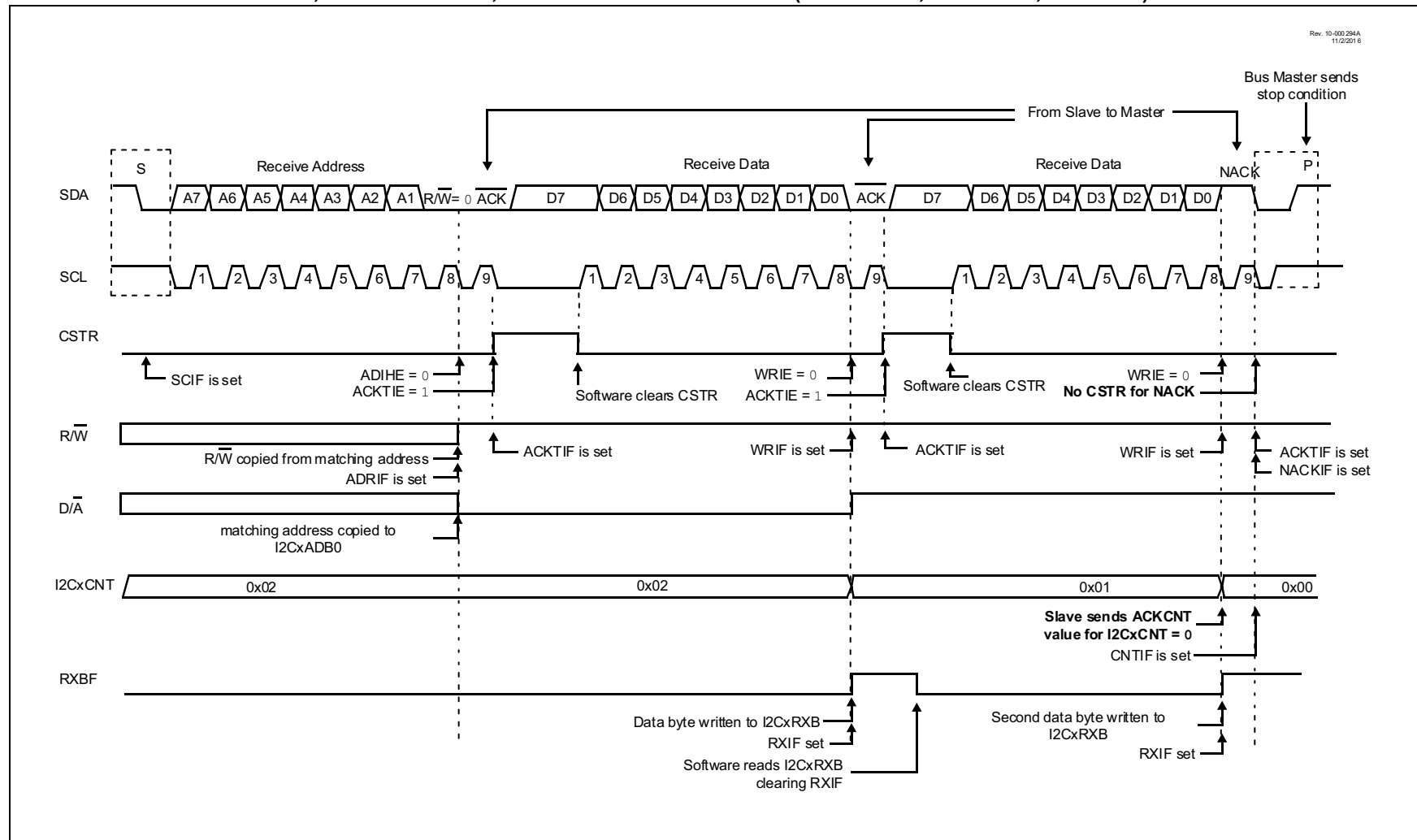
Sum of all received bytes except PID

All other modes and C0EN = 1:

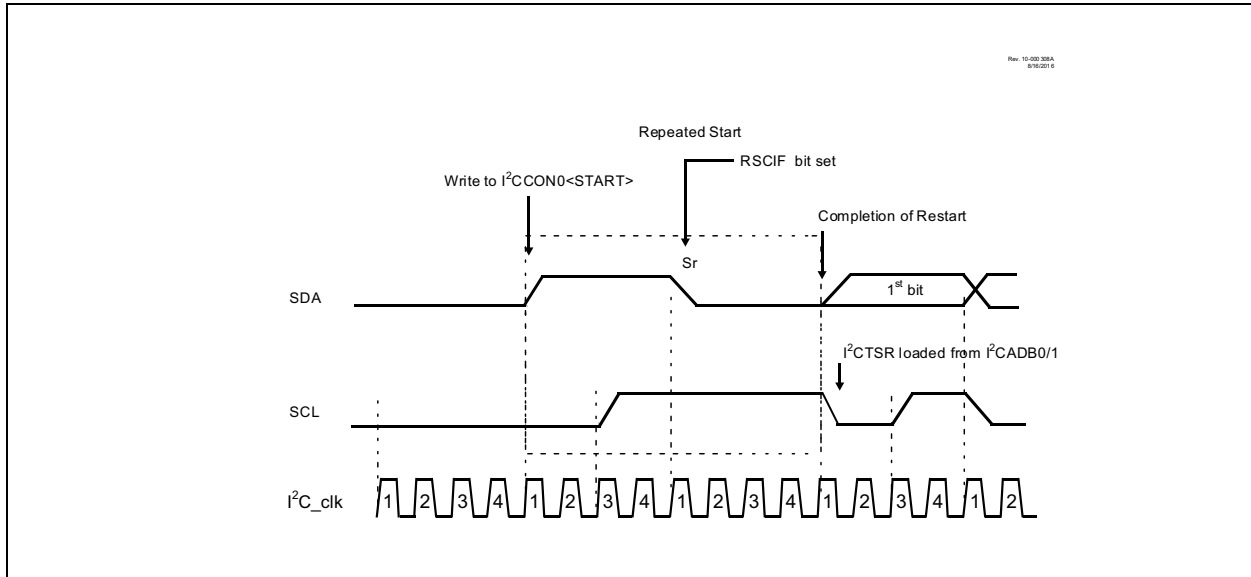
Sum of all received bytes since last clear

All other modes and C0EN = 0:

Not used

**FIGURE 33-7: I<sup>2</sup>C SLAVE, 7-BIT ADDRESS, RECEPTION WITH I2CxCNT (ACKTIE = 1, ADRIE = 0, WRIE = 0)**

**FIGURE 33-16: REPEATED START CONDITION TIMING**

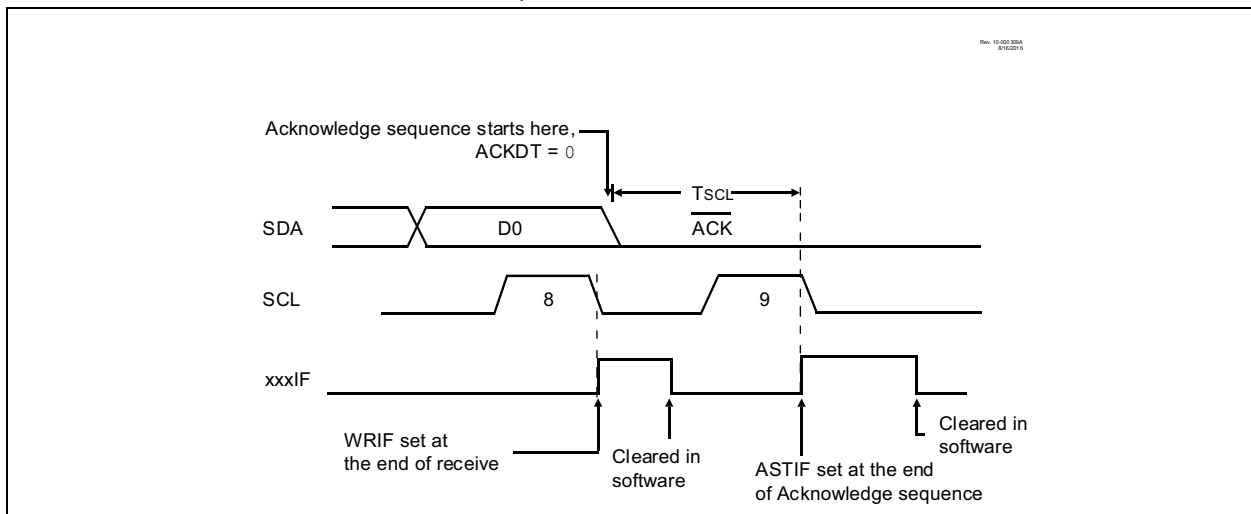


## 33.5.7 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled automatically following an address/data byte transmission. The SCL pin is pulled low and the contents of the Acknowledge Data bits (ACKDT/ACKCNT) are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user

should set the ACKDT bit before starting an Acknowledge sequence. The master then waits one clock period ( $T_{SCL}$ ) and the SCL pin is released high. When the SCL pin is sampled high (clock arbitration), the master counts another  $T_{SCL}$ . The SCL pin is then pulled low. Figure 33-17 shows the timings for Acknowledge sequence.

**FIGURE 33-17: ACKNOWLEDGE SEQUENCE TIMING**



## 33.5.8 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of receive/transmit when  $I2CxCNT = 0$ . After the last byte of a receive/transmit sequence, the SCL line is held low. The master asserts the SDA line low. The SCL pin is then released high  $T_{SCL}/2$  later and is detected high. The SDA pin is then released. When the SDA pin tran-

sitions high while SCL is high, the PCIF bit of the I2CxIF register is set. Figure 33-18 shows the timings for a Stop condition.

## REGISTER 37-15: ADCNT: ADC REPEAT COUNTER REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
CNT<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 **CNT<7:0>**: ADC Repeat Count bits  
Determines the number of times that the ADC is triggered before the threshold is checked when the computation is Low-pass Filter, Burst Average, or Average modes. See Table 37-2 for more details.

## REGISTER 37-16: ADFLTRH: ADC FILTER HIGH BYTE REGISTER

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
FLTR<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 **FLTR<15:8>**: ADC Filter Output Most Significant bits  
In Accumulate, Average, and Burst Average mode, this is equal to ACC right shifted by the ADCRS bits of ADCON2. In LPF mode, this is the output of the Low-pass Filter.

## REGISTER 37-17: ADFLTRL: ADC FILTER LOW BYTE REGISTER

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
FLTR<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 **FLTR<7:0>**: ADC Filter Output Least Significant bits  
In Accumulate, Average, and Burst Average mode, this is equal to ACC right shifted by the ADCRS bits of ADCON2. In LPF mode, this is the output of the Low-pass Filter.

**TABLE 45-23: I<sup>2</sup>C BUS DATA REQUIREMENTS**

Standard Operating Conditions (unless otherwise stated)							
Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
SP100*	THIGH	Clock high time	100 kHz mode	4.0	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	μs	Device must operate at a minimum of 10 MHz
			SSP module	1.5Tcy	—		
SP101*	TLOW	Clock low time	100 kHz mode	4.7	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	μs	Device must operate at a minimum of 10 MHz
			SSP module	1.5Tcy	—		
SP102*	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	CB is specified to be from 10-400 pF
			400 kHz mode	20 + 0.1CB	300	ns	
SP103*	TF	SDA and SCL fall time	100 kHz mode	—	250	ns	CB is specified to be from 10-400 pF
			400 kHz mode	20 + 0.1CB	250	ns	
SP106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
SP107*	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	(Note 2)
			400 kHz mode	100	—	ns	
SP109*	TAA	Output valid from clock	100 kHz mode	—	3500	ns	(Note 1)
			400 kHz mode	—	—	ns	
SP110*	TBUF	Bus free time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
SP111	CB	Bus capacitive loading		—	400	pF	

\* These parameters are characterized but not tested.

**Note 1:** As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

**2:** A Fast mode (400 kHz) I<sup>2</sup>C bus device can be used in a Standard mode (100 kHz) I<sup>2</sup>C bus system, but the requirement TSU:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line T<sub>R max.</sub> + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I<sup>2</sup>C bus specification), before the SCL line is released.