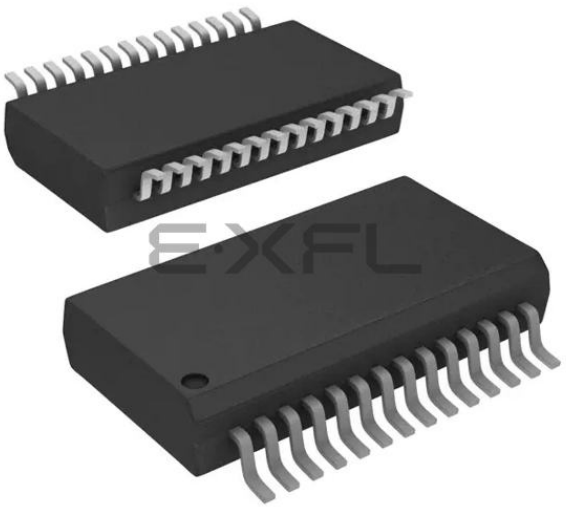


Welcome to E-XFL.COM

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, LVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 24x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f24k40-i-ss

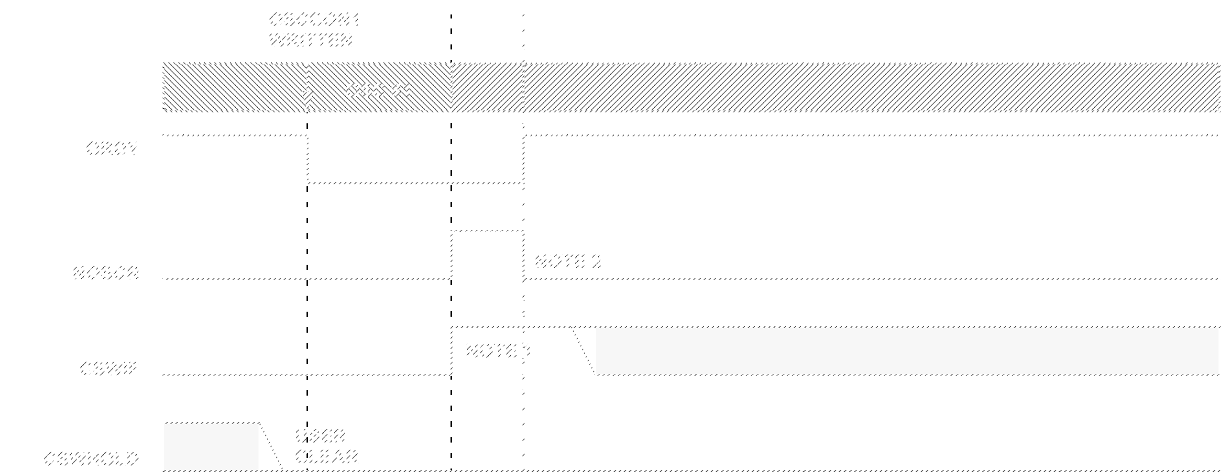
4.4.2 CLOCK SWITCH AND SLEEP

If OSCCON1 is written with a new value and the device is put to Sleep before the switch completes, the switch will not take place and the device will enter Sleep mode.

When the device wakes from Sleep and the CSWHOLD bit is clear, the device will wake with the 'new' clock active, and the clock switch interrupt flag bit (CSWIF) will be set.

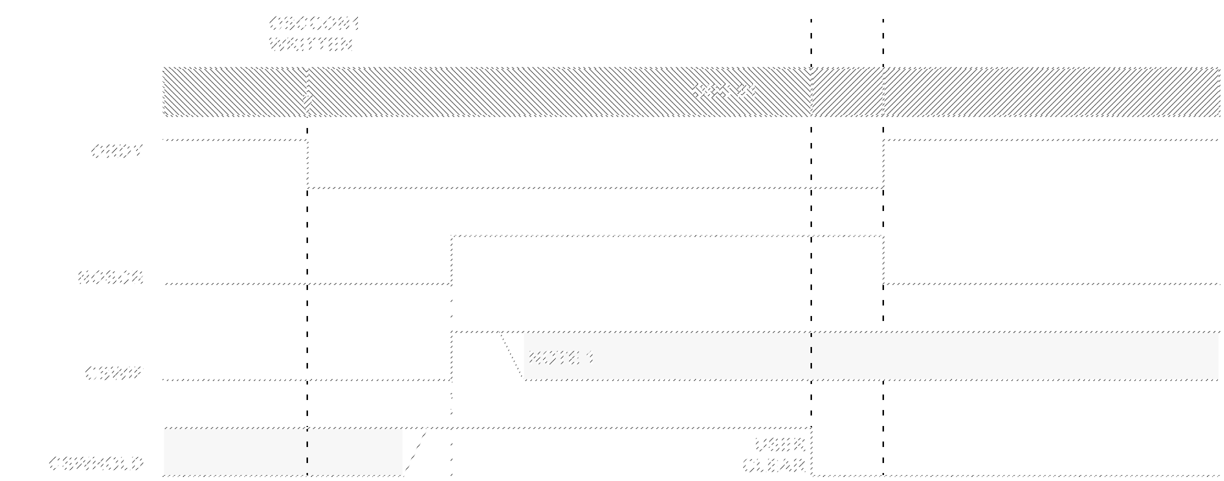
When the device wakes from Sleep and the CSWHOLD bit is set, the device will wake with the 'old' clock active and the new clock will be requested again.

FIGURE 4-6: CLOCK SWITCH (CSWHOLD = 0)



Note 1: CSWIF is asserted coincident with NOSCR; interrupt is serviced at OSC#2 speed.
2: The assertion of NOSCR is hidden from the user because it appears only for the duration of the switch.

FIGURE 4-7: CLOCK SWITCH (CSWHOLD = 1)

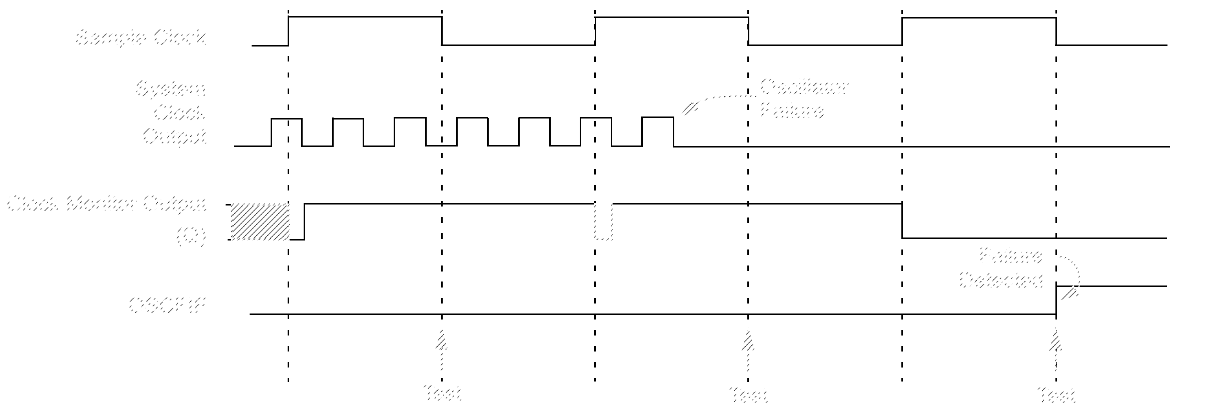


Note 1: CSWIF is asserted coincident with NOSCR, and may be cleared before or after clearing CSWHOLD = 0.

4.5.4 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect an oscillator failure after the Oscillator Start-up Timer (OST) has expired. The OST is used after waking up from Sleep and after any type of Reset. The OST is not used with the EC Clock modes so that the FSCM will be active as soon as the Reset or wake-up has completed.

FIGURE 4-10: FSCM TIMING DIAGRAM



Note: The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

REGISTER 6-2: CPUDOZE: DOZE AND IDLE REGISTER

R/W-0/u	R/W/HC/HS-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
IDLEN	DOZEN	ROI	DOE	—	DOZE<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	HC = Bit is cleared by hardware

bit 7 **IDLEN:** Idle Enable bit
1 = A SLEEP instruction inhibits the CPU clock, but not the peripheral clock(s)
0 = A SLEEP instruction places the device into full Sleep mode

bit 6 **DOZEN:** Doze Enable bit[1,2]
1 = The CPU executes instruction cycles according to DOZE setting
0 = The CPU executes all instruction cycles (fastest, highest power operation)

bit 5 **ROI:** Recover-On-Interrupt bit
1 = Entering the Interrupt Service Routine (ISR) makes DOZEN = 0 bit, bringing the CPU to full-speed operation
0 = Interrupt entry does not change DOZEN

bit 4 **DOE:** Doze-On-Exit bit
1 = Executing RETFIE makes DOZEN = 1, bringing the CPU to reduced speed operation
0 = RETFIE does not change DOZEN

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

bit 2-0 **DOZE<2:0>:** Ratio of CPU Instruction Cycles to Peripheral Instruction Cycles
111 =1:256
110 =1:128
101 =1:64
100 =1:32
011 =1:16
010 =1:8
001 =1:4
000 =1:2

Note 1: When ROI = 1 or DOE = 1, DOZEN is changed by hardware interrupt entry and/or exit.
2: Entering ICD overrides DOZEN, returning the CPU to full execution speed; this bit is not affected.

TABLE 6-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	166
PIE0	—	—	TMR0IE	IOCIE	—	INT2IE	INT1IE	INT0IE	175
PIE1	OSCFIE	CSWIE	—	—	—	—	ADTIE	ADIE	176
PIE2	HLVDIE	ZCDIE	—	—	—	—	C2IE	C1IE	177
PIE3	—	—	RC1IE	TX1IE	—	—	BCL1IE	SSP1IE	178
PIE4	—	—	TMR6IE	TMR5IE	TMR4IE	TMR3IE	TMR2IE	TMR1IE	179
PIE5	—	—	—	—	—	TMR5GIE	TMR3GIE	TMR1GIE	180
PIE6	—	—	—	—	—	—	CCP2IE	CCP1IE	181
PIE7	SCANIE	CRCIE	NVMIE	—	—	—	—	CWG1IE	182
PIR0	—	—	TMR0IF	IOCIF	—	INT2IF	INT1IF	INT0IF	167
PIR1	OSCFIF	CSWIF	—	—	—	—	ADTIF	ADIF	168
PIR2	HLVDIF	ZCDIF(1)	—	—	—	—	C2IF	C1IF	169
PIR3	—	—	RC1IF	TX1IF	—	—	BCL1IF	SSP1IF	170
PIR4	—	—	TMR6IF	TMR5IF	TMR4IF	TMR3IF	TMR2IF	TMR1IF	170
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	206
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	206
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	206
IOCCP(1)	—	—	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	206
IOCCN(1)	—	—	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	206
IOCCF(1)	—	—	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	206
STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	114
VREGCON	—	—	—	—	—	—	VREGPM	Reserved	60
CPUDOZE	IDLEN	DOZEN	ROI	DOE	—	DOZE<2:0>			61
WDTCON0	—	—	WDTPS<4:0>					SEN	81
WDTCON1	—	WDTPS<2:0>			—	WINDOW<2:0>			82

Note 1: — = unimplemented location, read as '0'. Shaded cells are not used in Power-Down mode.

7.0 PERIPHERAL MODULE DISABLE (PMD)

Sleep, Idle and Doze modes allow users to substantially reduce power consumption by slowing or stopping the CPU clock. Even so, peripheral modules still remain clocked, and thus, consume some amount of power. There may be cases where the application needs what these modes do not provide: the ability to allocate limited power resources to the CPU while eliminating power consumption from the peripherals.

The PIC18(L)F2x/4xK40 family addresses this requirement by allowing peripheral modules to be selectively enabled or disabled, placing them into the lowest possible power mode.

For legacy reasons, all modules are ON by default following any Reset.

7.1 Disabling a Module

Disabling a module has the following effects:

- All clock and control inputs to the module are suspended; there are no logic transitions, and the module will not function.
- The module is held in Reset.
- Any SFR becomes "unimplemented"
  - Writing is disabled
  - Reading returns 00h
- I/O functionality is prioritized as per **Section 15.1, I/O Priorities**
- All associated Input Selection registers are also disabled

7.2 Enabling a Module

When the PMD register bit is cleared, the module is re-enabled and will be in its Reset state (Power-on Reset). SFR data will reflect the POR Reset values.

Depending on the module, it may take up to one full instruction cycle for the module to become active. There should be no interaction with the module (e.g., writing to registers) for at least one instruction after it has been re-enabled.

7.3 Effects of a Reset

Following any Reset, each control bit is set to '0', enabling all modules.

7.4 System Clock Disable

Setting SYSCMD (PMD0, Register 7-1) disables the system clock (FOSC) distribution network to the peripherals. Not all peripherals make use of SYSCLK, so not all peripherals are affected. Refer to the specific peripheral description to see if it will be affected by this bit.

REGISTER 7-3: PMD2: PMD CONTROL REGISTER 2

U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	DACMD	ADCMD	—	—	CMP2MD	CMP1MD	ZCDMD(1)
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 **DACMD:** Disable DAC bit
1 = DAC module disabled
0 = DAC module enabled

bit 5 **ADCMD:** Disable ADC bit
1 = ADC module disabled
0 = ADC module enabled

bit 4-3 **Unimplemented:** Read as ‘0’

bit 2 **CMP2MD:** Disable Comparator CMP2 bit
1 = CMP2 module disabled
0 = CMP2 module enabled

bit 1 **CMP1MD:** Disable Comparator CMP1 bit
1 = CMP1 module disabled
0 = CMP1 module enabled

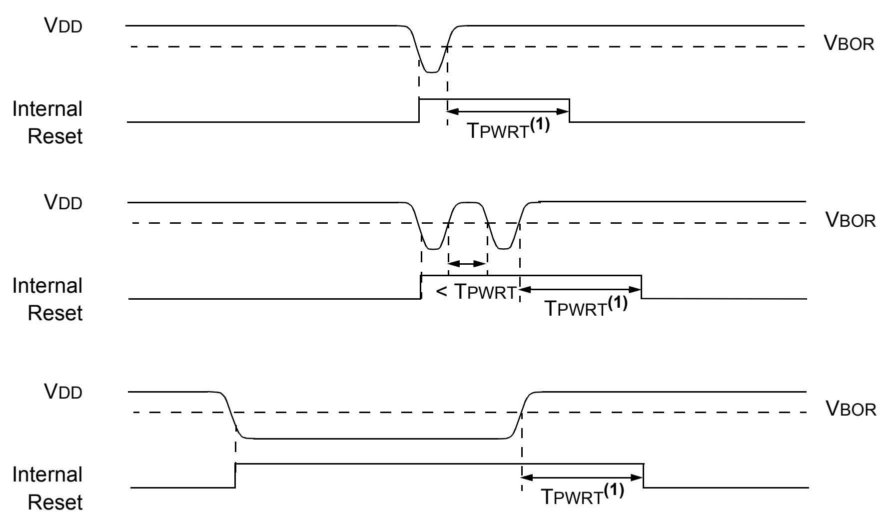
bit 0 **ZCDMD:** Disable Zero-Cross Detect module bit(1)
1 = ZCD module disabled
0 = ZCD module enabled

Note 1: Subject to $\overline{\text{ZCD}}$ bit in CONFIG2H.

TABLE 8-1: BOR OPERATING MODES

<table>
<tr><th rowspan="2">BOREN<1:0></th><th rowspan="2">SBOREN</th><th rowspan="2">Device Mode</th><th rowspan="2">BOR Mode</th><th colspan="2">Instruction Execution upon:</th></tr>
<tr><th>Release of POR</th><th>Wake-up from Sleep</th></tr>
<tr><td>11</td><td>X</td><td>X</td><td>Active</td><td>Wait for release of BOR (BORRDY = 1)</td><td>Begins immediately</td></tr>
<tr><td rowspan="2">10</td><td rowspan="2">X</td><td>Awake</td><td>Active</td><td>Wait for release of BOR (BORRDY = 1)</td><td>N/A</td></tr>
<tr><td>Sleep</td><td>Hibernate</td><td>N/A</td><td>Wait for release of BOR (BORRDY = 1)</td></tr>
<tr><td rowspan="2">01</td><td>1</td><td>X</td><td>Active</td><td rowspan="2">Wait for release of BOR (BORRDY = 1)</td><td rowspan="2">Begins immediately</td></tr>
<tr><td>0</td><td>X</td><td>Hibernate</td></tr>
<tr><td>00</td><td>X</td><td>X</td><td>Disabled</td><td colspan="2">Begins immediately</td></tr>
</table>

FIGURE 8-3: BROWN-OUT SITUATIONS



Note 1: TPWRT delay only if $\overline{\text{PWRTE}}$ bit is programmed to ‘0’.

8.13 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON0 registers are updated to indicate the cause of the Reset. Table 8-3 shows the Reset conditions of these registers.

TABLE 8-3: RESET CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register(2,3)	PCON0 Register
Power-on Reset	0	-110 0000	0011 110x
Brown-out Reset	0	-110 0000	0011 11u0
$\overline{MCLR}$ Reset during normal operation	0	-uuu uuuu	uuuu 0uuu
$\overline{MCLR}$ Reset during Sleep	0	-10u uuuu	uuuu 0uuu
WDT Time-out Reset	0	-0uu uuuu	uuu0 uuuu
WDT Wake-up from Sleep	PC + 2	-00u uuuu	uuuu uuuu
WWDT Window Violation Reset	0	-uuu uuuu	uu0u uuuu
Interrupt Wake-up from Sleep	PC + 2(1)	-10u 0uuu	uuuu uuuu
RESET Instruction Executed	0	-uuu uuuu	uuuu u0uu
Stack Overflow Reset (STVREN = 1)	0	-uuu uuuu	1uuu uuuu
Stack Underflow Reset (STVREN = 1)	0	-uuu uuuu	u1uu uuuu

Legend: u = unchanged, x = unknown, – = unimplemented bit, reads as '0'.

Note 1: When the wake-up is due to an interrupt and Global Interrupt Enable bit (GIE) is set the return address is pushed on the stack and PC is loaded with the corresponding interrupt vector (depending on source, high or low priority) after execution of PC + 2.

2: If a Status bit is not implemented, that bit will be read as '0'.

3: Status bits Z, C, DC are reset by POR/BOR (Register 10-2).

9.2 Independent Clock Source

The WWDT can derive its time base from either the 31 kHz LFINTOSC or 31.25 kHz MFINTOSC internal oscillators, depending on the value of WDTE<1:0> Configuration bits.

If WDTE = `2'b1x`, then the clock source will be enabled depending on the WDTCCS<2:0> Configuration bits.

If WDTE = `2'b01`, the SEN bit should be set by software to enable WWDT, and the clock source is enabled by the WDTCS bits in the WDTCON1 register.

Time intervals in this chapter are based on a minimum nominal interval of 1 ms. See **Section 37.0 "Electrical Specifications"** for LFINTOSC and MFINTOSC tolerances.

9.3 WWDT Operating Modes

The Windowed Watchdog Timer module has four operating modes controlled by the WDTE<1:0> bits in Configuration Words. See Table 9-1.

9.3.1 WWDT IS ALWAYS ON

When the WDTE bits of Configuration Words are set to '`11`', the WWDT is always on.

WWDT protection is active during Sleep.

9.3.2 WWDT IS OFF IN SLEEP

When the WDTE bits of Configuration Words are set to '`10`', the WWDT is on, except in Sleep.

WWDT protection is not active during Sleep.

9.3.3 WWDT CONTROLLED BY SOFTWARE

When the WDTE bits of Configuration Words are set to '`01`', the WWDT is controlled by the SEN bit of the WDTCON0 register.

WWDT protection is unchanged by Sleep. See Table 9-1 for more details.

TABLE 9-1: WWDT OPERATING MODES

WDTE<1:0>	SEN	Device Mode	WWDT Mode
11	X	X	Active
10	X	Awake	Active
		Sleep	Disabled
01	1	X	Active
	0	X	Disabled
00	X	X	Disabled

9.4 Time-out Period

If the WDTCPS<4:0> Configuration bits default to `5'b11111`, then the WDTPS bits of the WDTCON0 register set the time-out period from 1 ms to 256 seconds (nominal). If any value other than the default value is assigned to WDTCPS<4:0> Configuration bits, then the timer period will be based on the WDTCPS<4:0> bits in the CONFIG3L register. After a Reset, the default time-out period is 2s.

9.5 Watchdog Window

The Windowed Watchdog Timer has an optional Windowed mode that is controlled by the WDTCWS<2:0> Configuration bits and WINDOW<2:0> bits of the WDTCON1 register. In the Windowed mode, the `CLRWDT` instruction must occur within the allowed window of the WDT period. Any `CLRWDT` instruction that occurs outside of this window will trigger a window violation and will cause a WWDT Reset, similar to a WWDT time out. See Figure 9-2 for an example.

The window size is controlled by the WINDOW<2:0> Configuration bits, or the WINDOW<2:0> bits of WDTCON1, if WDTCWS<2:0> = `111`.

The five Most Significant bits of the WDTTMR register are used to determine whether the window is open, as defined by the WINDOW<2:0> bits of the WDTCON1 register.

In the event of a window violation, a Reset will be generated and the $\overline{\text{WDTWV}}$ bit of the PCON0 register will be cleared. This bit is set by a POR or can be set in firmware.

9.6 Clearing the WWDT

The WWDT is cleared when any of the following conditions occur:

- Any Reset
- Valid `CLRWDT` instruction is executed
- Device enters Sleep
- Exit Sleep by Interrupt
- WWDT is disabled
- Oscillator Start-up Timer (OST) is running
- Any write to the WDTCON0 or WDTCON1 registers

9.6.1 `CLRWDT` CONSIDERATIONS (WINDOWED MODE)

When in Windowed mode, the WWDT must be armed before a `CLRWDT` instruction will clear the timer. This is performed by reading the WDTCON0 register. Executing a `CLRWDT` instruction without performing such an arming action will trigger a window violation regardless of whether the window is open or not.

See Table 9-2 for more information.

TABLE 10-5: REGISTER FILE SUMMARY FOR PIC18(L)F24/25K40 DEVICES (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
EB1h	CWGINPPS	—	—	—	CWGINPPS<4:0>					---01000
EB0h	CCP2PPS	—	—	—	CCP2PPS<4:0>					---10001
EAFh	CCP1PPS	—	—	—	CCP1PPS<4:0>					---10010
EAEh	ADACTPPS	—	—	—	ADACTPPS<4:0>					---01100
EADh	T6INPPS	—	—	—	T6INPPS<4:0>					---01111
EACh	T4INPPS	—	—	—	T4INPPS<4:0>					---10101
EABh	T2INPPS	—	—	—	T2INPPS<4:0>					---10011
EAAh	T5GPPS	—	—	—	T5GPPS<4:0>					---01100
EA9h	T5CKIPPS	—	—	—	T5CKIPPS<4:0>					---10010
EA8h	T3GPPS	—	—	—	T3GPPS<4:0>					---10000
EA7h	T3CKIPPS	—	—	—	T3CKIPPS<4:0>					---10000
EA6h	T1GPPS	—	—	—	T1GPPS<4:0>					---01101
EA5h	T1CKIPPS	—	—	—	T1CKIPPS<4:0>					---10000
EA4h	T0CKIPPS	—	—	—	T0CKIPPS<4:0>					---00100
EA3h	INT2PPS	—	—	—	INT2PPS<4:0>					---01010
EA2h	INT1PPS	—	—	—	INT1PPS<4:0>					---01001
EA1h	INT0PPS	—	—	—	INT0PPS<4:0>					---01000
EA0h	PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	-------0
E9Fh to E7Eh	—	Unimplemented								—

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition
Note 1: Not available on LF devices.

11.0 NONVOLATILE MEMORY (NVM) CONTROL

Nonvolatile Memory (NVM) is separated into two types: Program Flash Memory (PFM) and Data EEPROM Memory.

PFM, Data EEPROM, User IDs and Configuration bits can all be accessed using the NVMREG<1:0> bits of the NVMCON1 register.

The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump rated to operate over the operating voltage range of the device.

NVM can be protected in two ways, by either code protection or write protection. Code protection ($\overline{\text{CP}}$ and $\overline{\text{CPD}}$ bits in Configuration Word 5L) disables access, reading and writing to both PFM and Data EEPROM Memory via external device programmers. Code protection does not affect the self-write and erase functionality. Code protection can only be reset by a device programmer performing a Bulk Erase to the device, clearing all nonvolatile memory, Configuration bits and User IDs.

Write protection prohibits self-write and erase to a portion or all of the PFM, as defined by the WRT bits of Configuration Word 4H. Write protection does not affect a device programmer's ability to read, write or erase the device.

TABLE 11-1: NVM ORGANIZATION AND ACCESS INFORMATION

Memory	PC<20:0> ICSP™ Addr<21:0> TBLPTR<21:0> NVMADDR<9:0>	Execution	User Access		
		CPU Execution	NVMREG	TABLAT	NVMDAT
User Flash Memory (PFM)	00 0000h • • • 01 FFFFh	Read	10	Read/ Write[1]	—[3]
User IDs[2]	20 0000h • • • 20 000Fh	No Access	x1	Read/ Write	—[3]
Reserved	20 0010h 2F FFFFh	No Access	—[3]		
Configuration	30 0000h • • • 30 000Bh	No Access	x1	Read/ Write	—[3]
Reserved	30 000Ch 30 FFFFh	No Access	—[3]		
User Data Memory (Data EEPROM)	31 0000h • • • 31 03FFh	No Access	00	—[3]	Read/ Write
Reserved	32 0000h 3F FFFBh	No Access	—[3]		
Revision ID/ Device ID	3F FFFCh • • • 3F FFFFh	No Access	x1	Read	—[3]

Note 1: Subject to Memory Write Protection settings.

2: User IDs are eight words ONLY. There is no code protection, table read protection or write protection implemented for this region.

3: Reads as '0', writes clear the WR bit and WRERR bit is set.

REGISTER 13-9: CRCXORH: CRC XOR HIGH BYTE REGISTER

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
X<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 **X<15:8>**: XOR of Polynomial Term XN Enable bits

REGISTER 13-10: CRCXORL: CRC XOR LOW BYTE REGISTER

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	U-1
X<7:1>							—
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 7-1 **X<7:1>**: XOR of Polynomial Term XN Enable bits

bit 0 **Unimplemented:** Read as ‘1’

13.3 CRC Functional Overview

The CRC module can be used to detect bit errors in the Flash memory using the built-in memory scanner or through user input RAM memory. The CRC module can accept up to a 16-bit polynomial with up to a 16-bit seed value. A CRC calculated check value (or checksum) will then be generated into the CRCACC<15:0> registers for user storage. The CRC module uses an XOR shift register implementation to perform the polynomial division required for the CRC calculation.

EXAMPLE 13-1: CRC EXAMPLE

Rev. 10-000206A
1/8/2014

CRC-16-ANSI

$x^{16} + x^{15} + x^{2} + 1$ (17 bits)

Standard 16-bit representation = 0x8005

```
CRCXORH = 0b10000000
CRCXORL = 0b0000010- (1)
```

Data Sequence:
0x55, 0x66, 0x77, 0x88

```
DLEN = 0b0111
PLEN = 0b1111
```

Data entered into the CRC:
SHIFTM = 0:

```
01010101 01100110 01110111 10001000
```

SHIFTM = 1:

```
10101010 01100110 11101110 00010001
```

Check Value (ACCM = 1):

SHIFTM = 0: 0x32D6

```
CRCACCH = 0b00110010
CRCACCL = 0b11010110
```

SHIFTM = 1: 0x6BA2

```
CRCACCH = 0b01101011
CRCACCL = 0b10100010
```

Note 1: Bit 0 is unimplemented. The LSb of any CRC polynomial is always ‘1’ and will always be treated as a ‘1’ by the CRC for calculating the CRC check value. This bit will be read in software as a ‘0’.

14.4 INTCON Registers

The INTCON registers are readable and writable registers, which contain various enable and priority bits.

14.5 PIR Registers

The PIR registers contain the individual flag bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are eight Peripheral Interrupt Request Flag registers (PIR0, PIR1, PIR2, PIR3, PIR4, PIR5, PIR6 and PIR7).

14.6 PIE Registers

The PIE registers contain the individual enable bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are eight Peripheral Interrupt Enable registers (PIE0, PIE1, PIE2, PIE3, PIE4, PIE5, PIE6 and PIE7). When IPEN = 0, the PEIE/GIEL bit must be set to enable any of these peripheral interrupts.

14.7 IPR Registers

The IPR registers contain the individual priority bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are eight Peripheral Interrupt Priority registers (IPR0, IPR1, IPR2, IPR3, IPR4 and IPR5, IPR6 and IPR7). Using the priority bits requires that the Interrupt Priority Enable (IPEN) bit be set.

18.3 Programmable Prescaler

A software programmable prescaler is available for exclusive use with Timer0. There are 16 prescaler options for Timer0 ranging in powers of two from 1:1 to 1:32768. The prescaler values are selected using the T0CKPS<3:0> bits of the T0CON1 register.

The prescaler is not directly readable or writable. Clearing the prescaler register can be done by writing to the TMR0L register or the T0CON0, T0CON1 registers or by any Reset.

18.4 Programmable Postscaler

A software programmable postscaler (output divider) is available for exclusive use with Timer0. There are 16 postscaler options for Timer0 ranging from 1:1 to 1:16. The postscaler values are selected using the T0OUTPS<3:0> bits of the T0CON0 register.

The postscaler is not directly readable or writable. Clearing the postscaler register can be done by writing to the TMR0L register or the T0CON0, T0CON1 registers or by any Reset.

18.5 Operation During Sleep

When operating synchronously, Timer0 will halt. When operating asynchronously, Timer0 will continue to increment and wake the device from Sleep (if Timer0 interrupts are enabled) provided that the input clock source is active.

18.6 Timer0 Interrupts

The Timer0 interrupt flag bit (TMR0IF) is set when either of the following conditions occur:

- 8-bit TMR0L matches the TMR0H value
- 16-bit TMR0 rolls over from 'FFFFh'

When the postscaler bits (T0OUTPS<3:0>) are set to 1:1 operation (no division), the T0IF flag bit will be set with every TMR0 match or rollover. In general, the TMR0IF flag bit will be set every T0OUTPS +1 matches or rollovers.

If Timer0 interrupts are enabled (TMR0IE bit of the PIE0 register = '1'), the CPU will be interrupted and the device may wake from Sleep (see **Section 18.2 "Clock Source Selection"** for more details).

18.7 Timer0 Output

The Timer0 output can be routed to any I/O pin via the RxyPPS output selection register (see **Section 17.0 "Peripheral Pin Select (PPS) Module"** for additional information). The Timer0 output can also be used by other peripherals, such as the auto-conversion trigger of the Analog-to-Digital Converter. Finally, the Timer0 output can be monitored through software via the Timer0 output bit (T0OUT) of the T0CON0 register (Register 18-1).

TMR0_out will be a pulse of one postscaled clock period when a match occurs between TMR0L and PR0 (Period register for TMR0) in 8-bit mode, or when TMR0 rolls over in 16-bit mode. The Timer0 output is a 50% duty cycle that toggles on each TMR0_out rising clock edge.

24.0 COMPLEMENTARY WAVEFORM GENERATOR (CWG) MODULE

The Complementary Waveform Generator (CWG) produces half-bridge, full-bridge, and steering of PWM waveforms. It is backwards compatible with previous CCP functions. The PIC18(L)F2x/4xK40 family has one instance of the CWG module.

The CWG has the following features:

- Six operating modes:
  - Synchronous Steering mode
  - Asynchronous Steering mode
  - Full-Bridge mode, Forward
  - Full-Bridge mode, Reverse
  - Half-Bridge mode
  - Push-Pull mode
- Output polarity control
- Output steering
- Independent 6-bit rising and falling event dead-band timers
  - Clocked dead band
  - Independent rising and falling dead-band enables
- Auto-shutdown control with:
  - Selectable shutdown sources
  - Auto-restart option
  - Auto-shutdown pin override control

24.1 Fundamental Operation

The CWG generates two output waveforms from the selected input source.

The off-to-on transition of each output can be delayed from the on-to-off transition of the other output, thereby, creating a time delay immediately where neither output is driven. This is referred to as dead time and is covered in **Section 24.6 “Dead-Band Control”**.

It may be necessary to guard against the possibility of circuit faults or a feedback event arriving too late or not at all. In this case, the active drive must be terminated before the Fault condition causes damage. This is referred to as auto-shutdown and is covered in **Section 24.10 “Auto-Shutdown”**.

24.2 Operating Modes

The CWG module can operate in six different modes, as specified by the MODE<2:0> bits of the CWG1CON0 register:

- Half-Bridge mode
- Push-Pull mode
- Asynchronous Steering mode
- Synchronous Steering mode
- Full-Bridge mode, Forward
- Full-Bridge mode, Reverse

All modes accept a single pulse data input, and provide up to four outputs as described in the following sections.

All modes include auto-shutdown control as described in **Section 24.10 “Auto-Shutdown”**

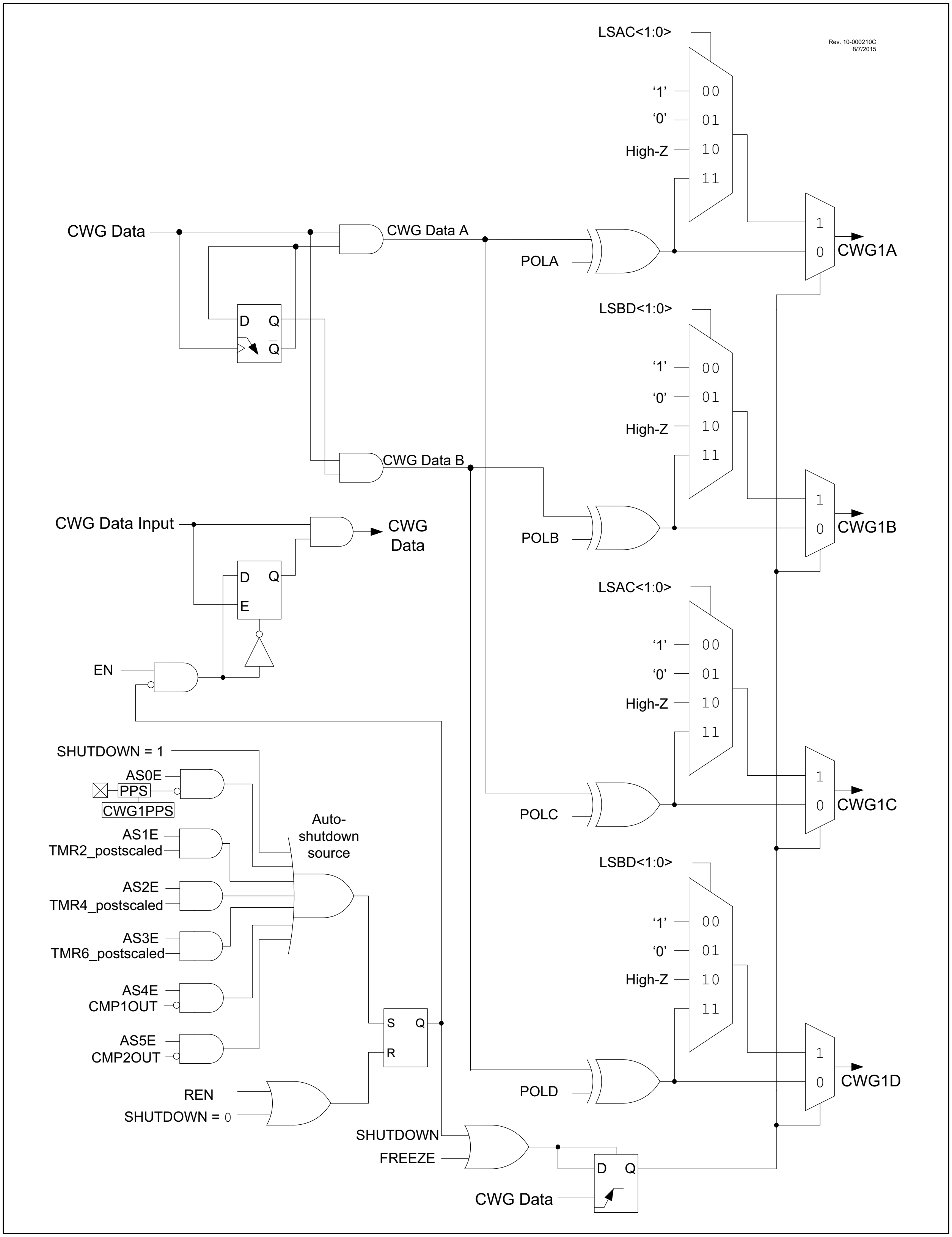
Note:	Except as noted for Full-bridge mode (**Section 24.2.3 “Full-Bridge Modes”**), mode changes should only be performed while EN = `0` (Register 24-1).

24.2.1 HALF-BRIDGE MODE

In Half-Bridge mode, two output signals are generated as true and inverted versions of the input as illustrated in Figure 24-2. A non-overlap (dead-band) time is inserted between the two outputs to prevent shoot through current in various power supply applications. Dead-band control is described in **Section 24.6 “Dead-Band Control”**. The output steering feature cannot be used in this mode. A basic block diagram of this mode is shown in Figure 24-1.

The unused outputs CWG1C and CWG1D drive similar signals, with polarity independently controlled by the POLC and POLD bits of the CWG1CON1 register, respectively.

FIGURE 24-3: SIMPLIFIED CWG BLOCK DIAGRAM (PUSH-PULL MODE, MODE<2:0> = 101)



26.10.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN bit of the SSPxCON2 register. When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit 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 Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 26-30).

26.10.8.1 WCOL Status Flag

If the user writes the SSPxBUF when an Acknowledge sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

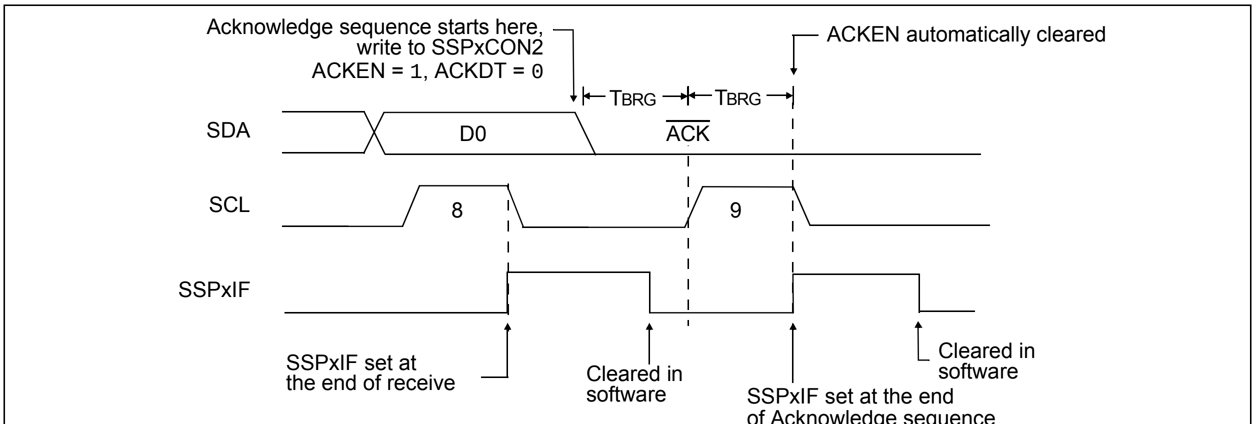
26.10.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN bit of the SSPxCON2 register. At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit of the SSPxSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 26-31).

26.10.9.1 WCOL Status Flag

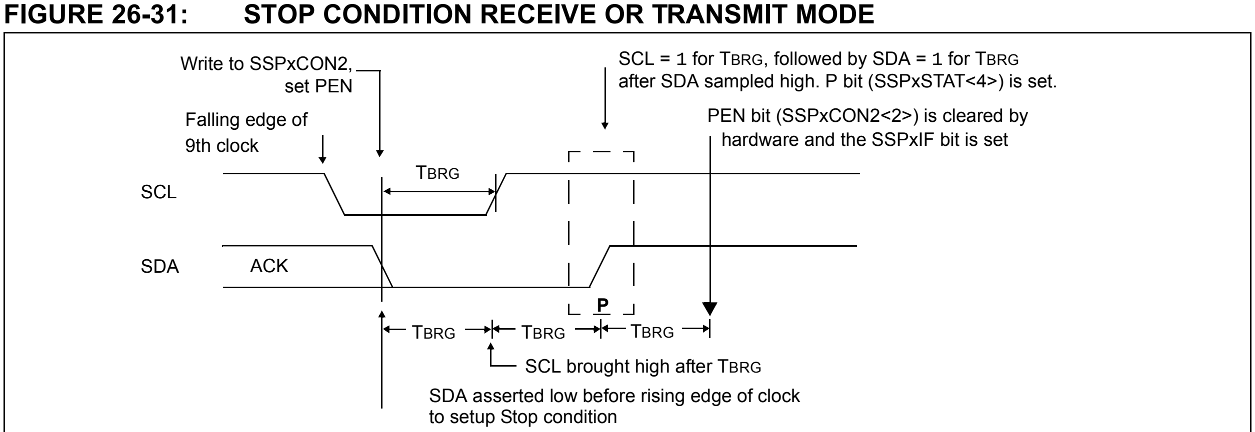
If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

FIGURE 26-30: ACKNOWLEDGE SEQUENCE WAVEFORM



Note: TBRG = one Baud Rate Generator period.

FIGURE 26-31: STOP CONDITION RECEIVE OR TRANSMIT MODE



Note: TBRG = one Baud Rate Generator period.

29.4 ADC Acquisition Time

To ensure accurate temperature measurements, the user must wait at least 200 µs after the ADC input multiplexer is connected to the temperature indicator output before the conversion is performed. In addition, the user must wait 200 µs between consecutive conversions of the temperature indicator output.

TABLE 29-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE TEMPERATURE INDICATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDFVR<1:0>		ADFVR<1:0>		417

Legend: — = Unimplemented location, read as '0'. Shaded cells are unused by the temperature indicator module.

TABLE 31-3: COMPUTATION MODES

		Bit Clear Conditions	Value after Trigger completion		Threshold Operations			Value at ADTIF interrupt		
Mode	ADMD	ADACC and ADCNT	ADACC	ADCNT	Retrigger	Threshold Test	Interrupt	ADAOV	ADFLTR	ADCNT
Basic	0	ADACLR = 1	Unchanged	Unchanged	No	Every Sample	If threshold=true	N/A	N/A	count
Accumulate	1	ADACLR = 1	S + ADACC or (S2-S1) + ADACC	If (ADCNT=FF): ADCNT, otherwise: ADCNT+1	No	Every Sample	If threshold=true	ADACC Overflow	ADACC/2^{ADCRS}	count
Average	2	ADACLR = 1 or ADCNT>=ADRPT at ADGO or retrigger	S + ADACC or (S2-S1) + ADACC	If (ADCNT=FF): ADCNT, otherwise: ADCNT+1	No	If ADCNT>= ADRPT	If threshold=true	ADACC Overflow	ADACC/2^{ADCRS}	count
Burst Average	3	ADACLR = 1 or ADGO set or retrigger	Each repetition: same as Average End with sum of all samples	Each repetition: same as Average End with ADCNT=ADRPT	Repeat while ADCNT<ADRPT	If ADCNT>= ADRPT	If threshold=true	ADACC Overflow	ADACC/2^{ADCRS}	ADRPT
Low-pass Filter	4	ADACLR = 1	S+ADACC-ADACC/2^{ADCRS} or (S2-S1)+ADACC-ADACC/2^{ADCRS}	If (ADCNT=FF): ADCNT, otherwise: ADCNT+1	No	If ADCNT>= ADRPT	If threshold=true	ADACC Overflow	Filtered Value	count

Note: S1 and S2 are abbreviations for Sample 1 and Sample 2, respectively. When ADDSEN = 0, S1 = ADRES; When ADDSEN = 1, S1 = ADPREV and S2 = ADRES.