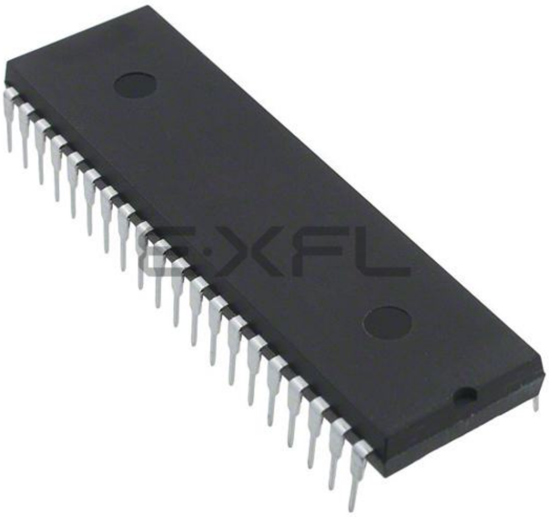
Microchip Technology - PIC18LF45K40-E/P Datasheet



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	36
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 35x10b; 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/pic18lf45k40-e-p

REGISTER 3-5: CONFIGURATION WORD 3L (30 0004h): WINDOWED WATCHDOG TIMER

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
11 = WDT enabled regardless of Sleep; SEN bit in WDTCON0 is ignored
10 = WDT enabled while Sleep = 0, suspended when Sleep = 1; SEN bit in WDTCON0 is ignored
01 = WDT enabled/disabled by SEN bit in WDTCON0
00 = WDT disabled, SEN bit in WDTCON0 is ignored

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

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

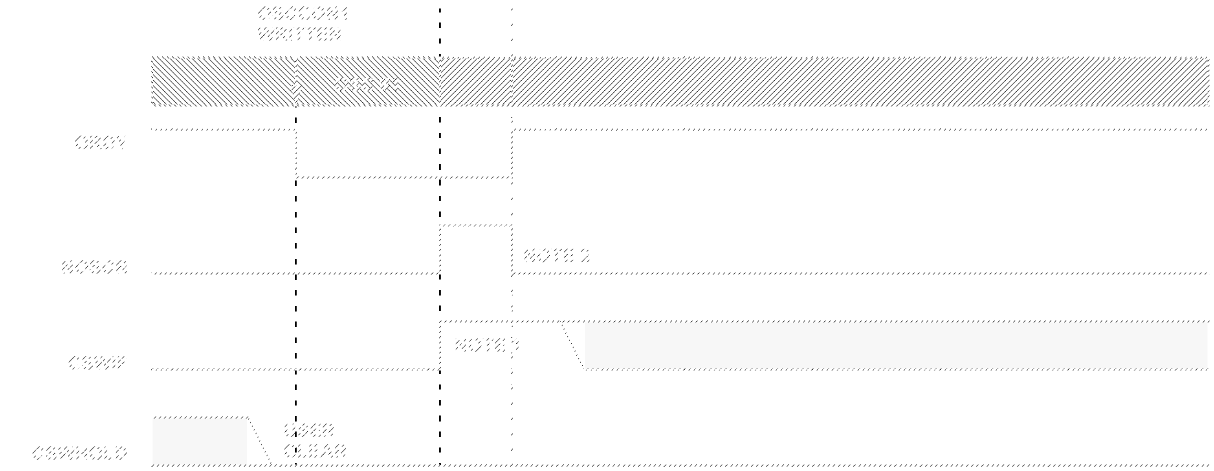
4.4.2 CLOCK SWITCH AND SLEEP

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.

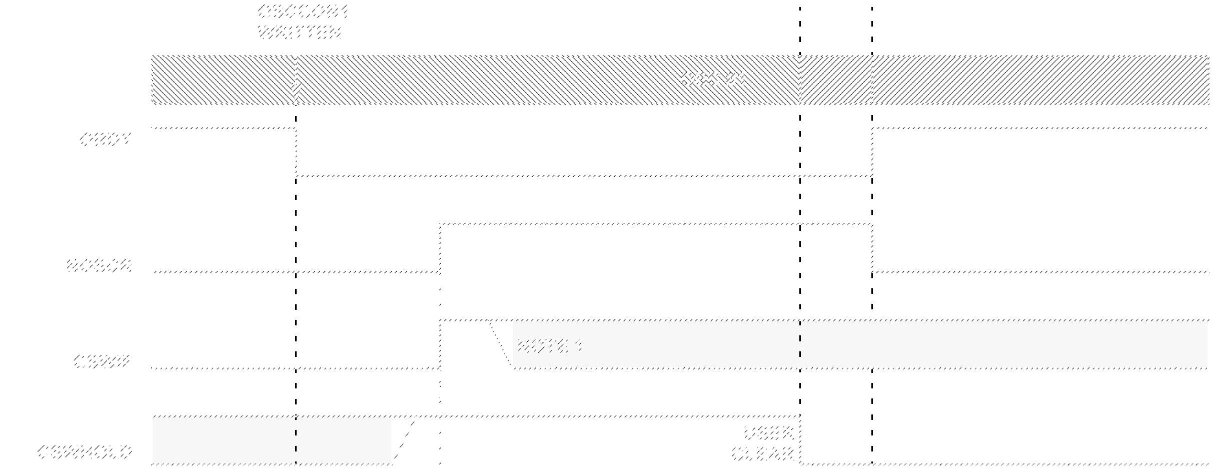
FIGURE 4-6: CLOCK SWITCH (CSWHOLD = 0)



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.

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`
$\overline{\text{MCLR}}$ Reset during normal operation	0	`-uuu uuuu`	`uuuu 0uuu`
$\overline{\text{MCLR}}$ 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).

10.5 Register Definitions: Status

REGISTER 10-2: STATUS: STATUS REGISTER

U-0	R-1/q	R-1/q	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u
—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	N	OV	Z	DC	C
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 **Unimplemented:** Read as ‘0’

bit 6 **$\overline{\text{TO}}$:** Time-Out bit
1 = Set at power-up or by execution of CLRWDT or SLEEP instruction
0 = A WDT time-out occurred

bit 5 **$\overline{\text{PD}}$:** Power-Down bit
1 = Set at power-up or by execution of CLRWDT instruction
0 = Set by execution of the SLEEP instruction

bit 4 **N:** Negative bit used for signed arithmetic (2’s complement); indicates if the result is negative, (ALU MSb = 1).
1 = The result is negative
0 = The result is positive

bit 3 **OV:** Overflow bit used for signed arithmetic (2’s complement); indicates an overflow of the 7-bit magnitude, which causes the sign bit (bit 7) to change state.
1 = Overflow occurred for current signed arithmetic operation
0 = No overflow occurred

bit 2 **Z:** Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit Carry/$\overline{\text{Borrow}}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)[(1)]
1 = A carry-out from the 4th low-order bit of the result occurred
0 = No carry-out from the 4th low-order bit of the result

bit 0 **C:** Carry/$\overline{\text{Borrow}}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)[(1,2)]
1 = A carry-out from the Most Significant bit of the result occurred
0 = No carry-out from the Most Significant bit of the result occurred

Note 1: For $\overline{\text{Borrow}}$, the polarity is reversed. A subtraction is executed by adding the two’s complement of the second operand.

2: For Rotate (RRF, RLF) instructions, this bit is loaded with either the high or low-order bit of the Source register.

REGISTER 13-14: SCANLADRL: SCAN LOW ADDRESS LOW BYTE 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
LADR<7:0>(1, 2)							
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 **LADR<7:0>**: Scan Start/Current Address bits(1, 2)
Least Significant bits of the current address to be fetched from, value increments on each fetch of memory

Note 1: Registers SCANLADRU/H/L form a 22-bit value, but are not guarded for atomic or asynchronous access; registers should only be read or written while SCANGO = 0 (SCANCON0 register).

2: While SCANGO = 1 (SCANCON0 register), writing to this register is ignored.

REGISTER 13-15: SCANHADRU: SCAN HIGH ADDRESS UPPER BYTE REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	HADR<21:16>					
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 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **HADR<21:16>:** Scan End Address bits(1, 2)
Upper bits of the address at the end of the designated scan

Note 1: Registers SCANHADRU/H/L form a 22-bit value but are not guarded for atomic or asynchronous access; registers should only be read or written while SCANGO = 0 (SCANCON0 register).

2: While SCANGO = 1 (SCANCON0 register), writing to this register is ignored.

13.9 Program Memory Scan Configuration

If desired, the program memory scan module may be used in conjunction with the CRC module to perform a CRC calculation over a range of program memory addresses. In order to set up the scanner to work with the CRC you need to perform the following steps:

1. Set the Enable bit in both the CRCCON0 and SCANCON0 registers. If they get disabled, all internal states of the scanner and the CRC are reset (registers are unaffected).
2. Choose which memory access mode is to be used (see **Section 13.11 "Scanning Modes"**) and set the MODE bits of the SCANCON0 register appropriately.
3. Based on the memory access mode, set the INTM bits of the SCANCON0 register to the appropriate interrupt mode (see **Section 13.11.5 "Interrupt Interaction"**)
4. Set the SCANLADRL/H/U and SCANHADRL/H/U registers with the beginning and ending locations in memory that are to be scanned.
5. The CRCGO bit must be set before setting the SCANGO bit. Setting the SCANGO bit starts the scan. Both CRCEN and CRCGO bits must be enabled to use the scanner. When either of these bits are disabled, the scan aborts and the INVALID bit SCANCON0 is set. The scanner will wait for the signal from the CRC that it is ready for the first Flash memory location, then begin loading data into the CRC. It will continue to do so until it either hits the configured end address or an address that is unimplemented on the device, at which point the SCANGO bit will clear, Scanner functions will cease, and the SCANIF interrupt will be triggered. Alternately, the SCANGO bit can be cleared in software if desired.

13.10 Scanner Interrupt

The scanner will trigger an interrupt when the SCANGO bit transitions from '1' to '0'. The SCANIF interrupt flag of PIR7 is set when the last memory location is reached and the data is entered into the CRCDATA registers. The SCANIF bit can only be cleared in software. The SCAN interrupt enable is the SCANIE bit of the PIE7 register.

13.11 Scanning Modes

The memory scanner can scan in four modes: Burst, Peek, Concurrent, and Triggered. These modes are controlled by the MODE bits of the SCANCON0 register. The four modes are summarized in Table 13-2.

13.11.1 BURST MODE

When MODE = 01, the scanner is in Burst mode. In Burst mode, CPU operation is stalled beginning with the operation after the one that sets the SCANGO bit, and the scan begins, using the instruction clock to execute. The CPU is held in its current state until the scan stops. Note that because the CPU is not executing instructions, the SCANGO bit cannot be cleared in software, so the CPU will remain stalled until one of the hardware end-conditions occurs. Burst mode has the highest throughput for the scanner, but has the cost of stalling other execution while it occurs.

13.11.2 CONCURRENT MODE

When MODE = 00, the scanner is in Concurrent mode. Concurrent mode, like Burst mode, stalls the CPU while performing accesses of memory. However, while Burst mode stalls until all accesses are complete, Concurrent mode allows the CPU to execute in between access cycles.

13.11.3 TRIGGERED MODE

When MODE = 11, the scanner is in Triggered mode. Triggered mode behaves identically to Concurrent mode, except instead of beginning the scan immediately upon the SCANGO bit being set, it waits for a rising edge from a separate trigger clock, the source of which is determined by the SCANTRIG register.

13.11.4 PEEK MODE

When MODE = 10, the scanner is in Peek mode. Peek mode waits for an instruction cycle in which the CPU does not need to access the NVM (such as a branch instruction) and uses that cycle to do its own NVM access. This results in the lowest throughput for the NVM access (and can take a much longer time to complete a scan than the other modes), but does so without any impact on execution times, unlike the other modes.

REGISTER 17-3: PPSLOCK: PPS LOCK REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	PPSLOCKED
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 **Unimplemented:** Read as ‘0’

bit 0 **PPSLOCKED:** PPS Locked bit
1 = PPS is locked. PPS selections can not be changed.
0 = PPS is not locked. PPS selections can be changed.

TABLE 17-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE PPS MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	219
INT0PPS	—	—	—	INT0PPS<4:0>					216
INT1PPS	—	—	—	INT1PPS<4:0>					216
INT2PPS	—	—	—	INT2PPS<4:0>					216
T0CKIPPS	—	—	—	T0CKIPPS<4:0>					216
T1CKIPPS	—	—	—	T1CKIPPS<4:0>					216
T1GPPS	—	—	—	T1GPPS<4:0>					216
T3CKIPPS	—	—	—	T3CKIPPS<4:0>					216
T3GPPS	—	—	—	T3GPPS<4:0>					216
T5CKIPPS	—	—	—	T5CKIPPS<4:0>					216
T5GPPS	—	—	—	T5GPPS<4:0>					216
T2INPPS	—	—	—	T2INPPS<4:0>					216
T4INPPS	—	—	—	T4INPPS<4:0>					216
T6INPPS	—	—	—	T6INPPS<4:0>					216
CCP1PPS	—	—	—	CCP1PPS<4:0>					216
CCP2PPS	—	—	—	CCP2PPS<4:0>					216
CWG1PPS	—	—	—	CWG1PPS<4:0>					216
MDCARLPPS	—	—	—	MDCARLPPS<4:0>					216
MDCARHPPS	—	—	—	MDCARHPPS<4:0>					216
MDSRCPPS	—	—	—	MDSRCPPS<4:0>					216
ADACTPPS	—	—	—	ADACTPPS<4:0>					216
SSP1CLKPPS	—	—	—	SSP1CLKPPS<4:0>					216
SSP1DATPPS	—	—	—	SSP1DATPPS<4:0>					216
SSP1SSPPS	—	—	—	SSP1SSPPS<4:0>					216
RX1PPS	—	—	—	RX1PPS<4:0>					218
TX1PPS	—	—	—	TX1PPS<4:0>					216
SSP2CLKPPS	—	—	—	SSP2CLKPPS<4:0>					216
SSP2DATPPS	—	—	—	SSP2DATPPS<4:0>					216

REGISTER 18-2: T0CON1: TIMER0 CONTROL REGISTER 1

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
T0CS<2:0>			T0ASYNC	T0CKPS<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 7-5 **T0CS<2:0>:**Timer0 Clock Source Select bits
111 = Reserved
110 = Reserved
101 = SOSC
100 = LFINTOSC
011 = HFINTOSC
010 = FOSC/4
001 = Pin selected by T0CKIPPS (Inverted)
000 = Pin selected by T0CKIPPS (Non-inverted)

bit 4 **T0ASYNC:** TMR0 Input Asynchronization Enable bit
1 = The input to the TMR0 counter is not synchronized to system clocks
0 = The input to the TMR0 counter is synchronized to FOSC/4

bit 3-0 **T0CKPS<3:0>:** Prescaler Rate Select bit
1111 = 1:32768
1110 = 1:16384
1101 = 1:8192
1100 = 1:4096
1011 = 1:2048
1010 = 1:1024
1001 = 1:512
1000 = 1:256
0111 = 1:128
0110 = 1:64
0101 = 1:32
0100 = 1:16
0011 = 1:8
0010 = 1:4
0001 = 1:2
0000 = 1:1

21.4.1 CCPx PIN CONFIGURATION

The software must configure the CCPx pin as an output by clearing the associated TRIS bit and defining the appropriate output pin through the RxyPPS registers. See **Section 17.0 “Peripheral Pin Select (PPS) Module”** for more details.

The CCP output can also be used as an input for other peripherals.

> **Note:** Clearing the CCPxCON register will force the CCPx compare output latch to the default low level. This is not the PORT I/O data latch.

21.4.2 TIMER1 MODE RESOURCE

In Compare mode, Timer1 must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

See **Section 19.0 “Timer1/3/5 Module with Gate Control”** for more information on configuring Timer1.

> **Note:** Clocking Timer1 from the system clock (FOSC) should not be used in Compare mode. In order for Compare mode to recognize the trigger event on the CCPx pin, TImer1 must be clocked from the instruction clock (FOSC/4) or from an external clock source.

21.4.3 AUTO-CONVERSION TRIGGER

All CCPx modes set the CCP interrupt flag (CCPxIF). When this flag is set and a match occurs, an auto-conversion trigger can take place if the CCP module is selected as the conversion trigger source.

Refer to **Section 31.2.5 “Auto-Conversion Trigger”** for more information.

> **Note:** Removing the match condition by changing the contents of the CCPRxH and CCPRxL register pair, between the clock edge that generates the Auto-conversion Trigger and the clock edge that generates the Timer1 Reset, will preclude the Reset from occurring

21.4.4 COMPARE DURING SLEEP

Since FOSC is shut down during Sleep mode, the Compare mode will not function properly during Sleep, unless the timer is running. The device will wake on interrupt (if enabled).

21.5 PWM Overview

Pulse-Width Modulation (PWM) is a scheme that provides power to a load by switching quickly between fully on and fully off states. The PWM signal resembles a square wave where the high portion of the signal is considered the ON state and the low portion of the signal is considered the OFF state. The high portion, also known as the pulse width, can vary in time and is defined in steps. A larger number of steps applied, which lengthens the pulse width, also supplies more power to the load. Lowering the number of steps applied, which shortens the pulse width, supplies less power. The PWM period is defined as the duration of one complete cycle or the total amount of on and off time combined.

PWM resolution defines the maximum number of steps that can be present in a single PWM period. A higher resolution allows for more precise control of the pulse-width time and in turn the power that is applied to the load.

The term duty cycle describes the proportion of the on time to the off time and is expressed in percentages, where 0% is fully off and 100% is fully on. A lower duty cycle corresponds to less power applied and a higher duty cycle corresponds to more power applied.

Figure 21-3 shows a typical waveform of the PWM signal.

21.5.1 STANDARD PWM OPERATION

The standard PWM function described in this section is available and identical for all CCP modules.

The standard PWM mode generates a Pulse-Width Modulation (PWM) signal on the CCPx pin with up to ten bits of resolution. The period, duty cycle, and resolution are controlled by the following registers:

- PR2 registers
- T2CON registers
- CCPRxL and CCPRxH registers
- CCPxCON registers

It is required to have FOSC/4 as the clock input to TMR2/4/6 for correct PWM operation. Figure 21-4 shows a simplified block diagram of PWM operation.

> **Note:** The corresponding TRIS bit must be cleared to enable the PWM output on the CCPx pin.

FIGURE 21-3: CCP PWM OUTPUT SIGNAL

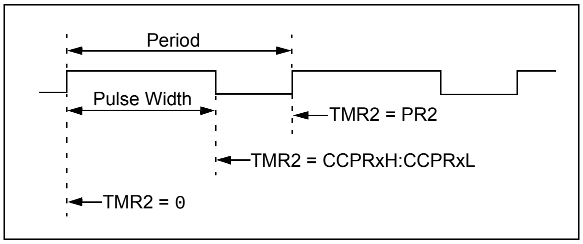
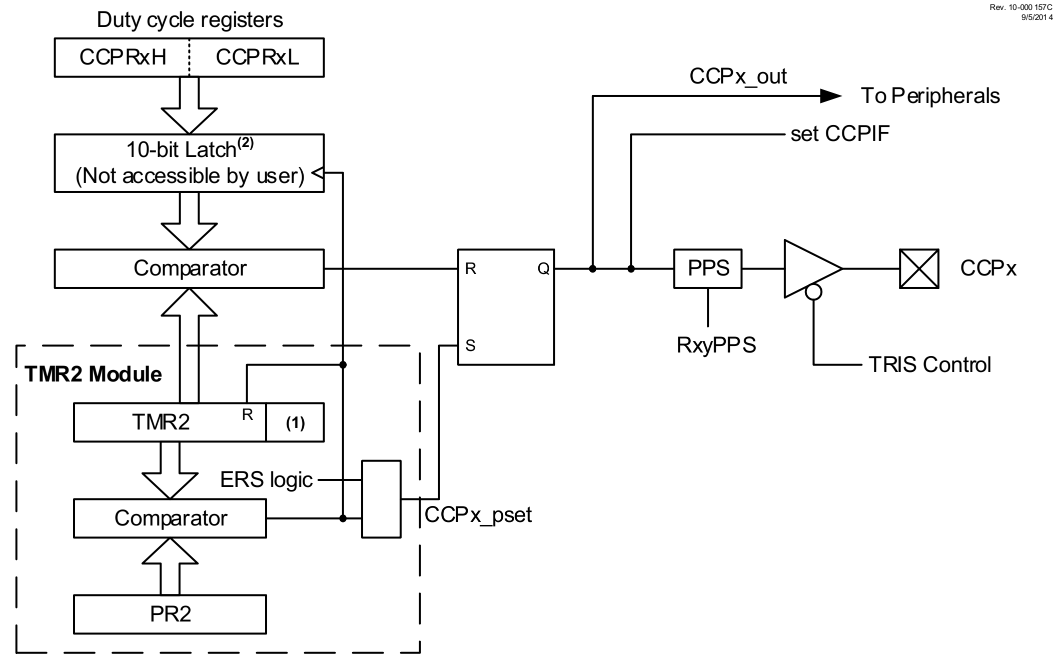


FIGURE 21-4: SIMPLIFIED PWM BLOCK DIAGRAM



Notes: 1. 8-bit timer is concatenated with two bits generated by Fosc or two bits of the internal prescaler to create 10-bit time-base.
2. The alignment of the 10 bits from the CCPR register is determined by the FMT bit.

23.10 Register Definitions: ZCD Control

REGISTER 23-1: 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
ZCDSEN	—	ZCDOUT	ZCDPOL	—	—	ZCDINTP	ZCDINTN
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 **ZCDSEN:** Zero-Cross Detect Software Enable bit

This bit is ignored when ZCDSEN fuse is set.

1= Zero-cross detect is enabled. ZCD pin is forced to output to source and sink current.
0= Zero-cross detect is disabled. ZCD pin operates according to PPS and TRIS controls.

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **ZCDOUT:** 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 **ZCDPOL:** 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 **ZCDINTP:** 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 **ZCDINTN:** 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

24.10 Auto-Shutdown

Auto-shutdown is a method to immediately override the CWG output levels with specific overrides that allow for safe shutdown of the circuit. The shutdown state can be either cleared automatically or held until cleared by software. The auto-shutdown circuit is illustrated in Figure 24-14.

24.10.1 SHUTDOWN

The shutdown state can be entered by either of the following two methods:

- Software generated
- External Input

24.10.1.1 Software Generated Shutdown

Setting the SHUTDOWN bit of the CWG1AS0 register will force the CWG into the shutdown state.

When the auto-restart is disabled, the shutdown state will persist as long as the SHUTDOWN bit is set.

When auto-restart is enabled, the SHUTDOWN bit will clear automatically and resume operation on the next rising edge event. The SHUTDOWN bit indicates when a shutdown condition exists. The bit may be set or cleared in software or by hardware.

24.10.1.2 External Input Source

External shutdown inputs provide the fastest way to safely suspend CWG operation in the event of a Fault condition. When any of the selected shutdown inputs goes active, the CWG outputs will immediately go to the selected override levels without software delay. The override levels are selected by the LSBD<1:0> and LSAC<1:0> bits of the CWG1AS0 register (Register 24-6). Several input sources can be selected to cause a shutdown condition. All input sources are active-low. The sources are:

- Pin selected by CWG1PPS
- Timer2 post-scaled output
- Timer4 post-scaled output
- Timer6 post-scaled output
- Comparator 1 output
- Comparator 2 output

Shutdown input sources are individually enabled by the ASxE bits of the CWG1AS1 register (Register 24-7).

Note:	Shutdown inputs are level sensitive, not edge sensitive. The shutdown state cannot be cleared, except by disabling auto-shutdown, as long as the shutdown input level persists.

24.10.1.3 Pin Override Levels

The levels driven to the CWG outputs during an auto-shutdown event are controlled by the LSBD<1:0> and LSAC<1:0> bits of the CWG1AS0 register (Register 24-6). The LSBD<1:0> bits control CWG1B/D output levels, while the LSAC<1:0> bits control the CWG1A/C output levels.

24.10.1.4 Auto-Shutdown Interrupts

When an auto-shutdown event occurs, either by software or hardware setting SHUTDOWN, the CWG1IF flag bit of the PIR7 register is set (Register 14-5).

24.11 Auto-Shutdown Restart

After an auto-shutdown event has occurred, there are two ways to resume operation:

- Software controlled
- Auto-restart

In either case, the shut-down source must be cleared before the restart can take place. That is, either the shutdown condition must be removed, or the corresponding ASxE bit must be cleared.

24.11.1 SOFTWARE-CONTROLLED RESTART

If the REN bit of the CWG1AS0 register is clear (REN = 0), the CWG module must be restarted after an auto-shutdown event through software.

Once all auto-shutdown sources are removed, the software must clear SHUTDOWN. Once SHUTDOWN is cleared, the CWG module will resume operation upon the first rising edge of the CWG data input.

Note:	The SHUTDOWN bit cannot be cleared in software if the auto-shutdown condition is still present.

24.11.2 AUTO-RESTART

If the REN bit of the CWG1AS0 register is set (REN = 1), the CWG module will restart from the shutdown state automatically.

Once all auto-shutdown conditions are removed, the hardware will automatically clear SHUTDOWN. Once SHUTDOWN is cleared, the CWG module will resume operation upon the first rising edge of the CWG data input.

Note:	The SHUTDOWN bit cannot be cleared in software if the auto-shutdown condition is still present.

FIGURE 26-6: SLAVE SELECT SYNCHRONOUS WAVEFORM

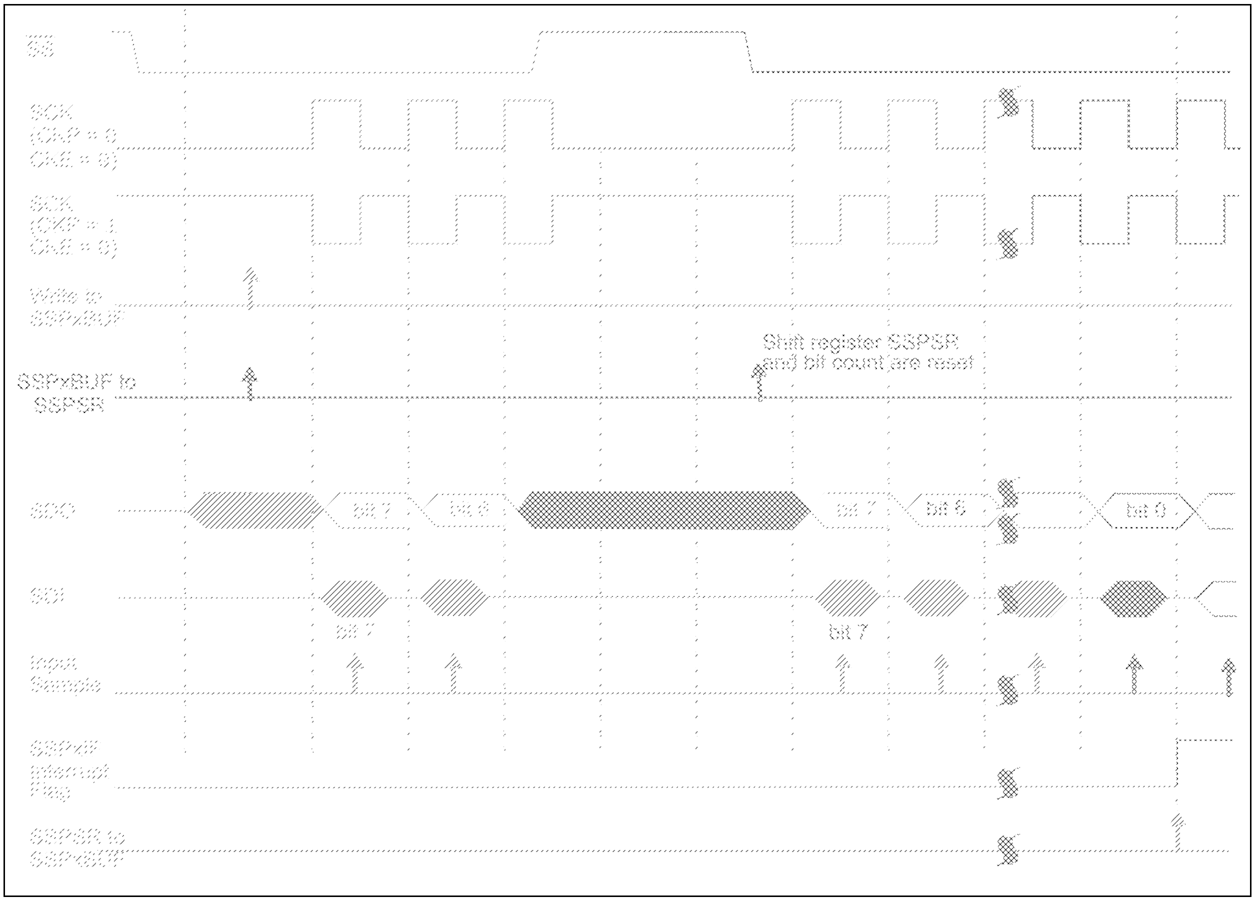
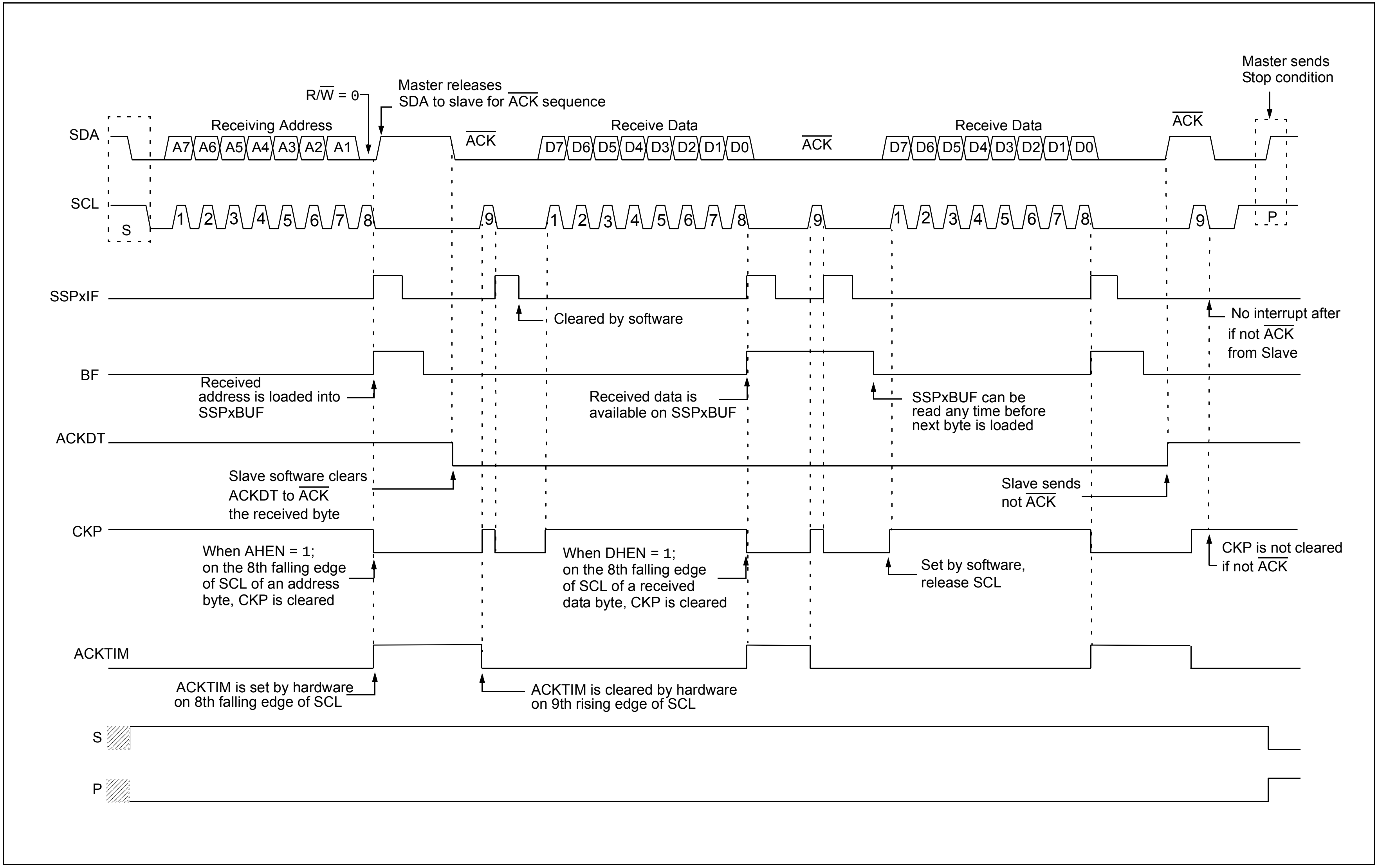


FIGURE 26-17: I^{2}C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 1, DHEN = 1)


SDA
SCL
SSPxIF
BF
ACKDT
CKP
ACKTIM
S
P
R/$\overline{W}$ = 0
Master releases SDA to slave for $\overline{ACK}$ sequence
Receiving Address
A7 A6 A5 A4 A3 A2 A1
$\overline{ACK}$
Receive Data
D7 D6 D5 D4 D3 D2 D1 D0
$\overline{ACK}$
Receive Data
D7 D6 D5 D4 D3 D2 D1 D0
$\overline{ACK}$
Master sends Stop condition
S
1 2 3 4 5 6 7 8
9
1 2 3 4 5 6 7 8
9
1 2 3 4 5 6 7 8
9
P
Cleared by software
No interrupt after if not $\overline{ACK}$ from Slave
Received address is loaded into SSPxBUF
Received data is available on SSPxBUF
SSPxBUF can be read any time before next byte is loaded
Slave software clears ACKDT to $\overline{ACK}$ the received byte
Slave sends not $\overline{ACK}$
When AHEN = 1; on the 8th falling edge of SCL of an address byte, CKP is cleared
When DHEN = 1; on the 8th falling edge of SCL of a received data byte, CKP is cleared
Set by software, release SCL
CKP is not cleared if not $\overline{ACK}$
ACKTIM is set by hardware on 8th falling edge of SCL
ACKTIM is cleared by hardware on 9th rising edge of SCL

26.10.5 I^{2}C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition (Figure 26-27) occurs when the RSEN bit of the SSPxCON2 register is programmed high and the master state machine is no longer active. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. SCL is asserted low. Following this, the RSEN bit of the SSPxCON2 register will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit of the SSPxSTAT register will be set. The SSPxIF bit will not be set until the Baud Rate Generator has timed out.

Note 1: If RSEN is programmed while any other event is in progress, it will not take effect.

2: A bus collision during the Repeated Start condition occurs if:

- SDA is sampled low when SCL goes from low-to-high.
- SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data '1'.

26.10.6 I^{2}C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPxBUF register. This action will set the Buffer Full flag bit, BF, and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted. SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high. When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an $\overline{\text{ACK}}$ bit during the ninth bit time if an address match occurred, or if data was received properly. The status of $\overline{\text{ACK}}$ is written into the ACKSTAT bit on the rising edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPxIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPxBUF, leaving SCL low and SDA unchanged (Figure 26-28).

FIGURE 26-27: REPEATED START CONDITION WAVEFORM

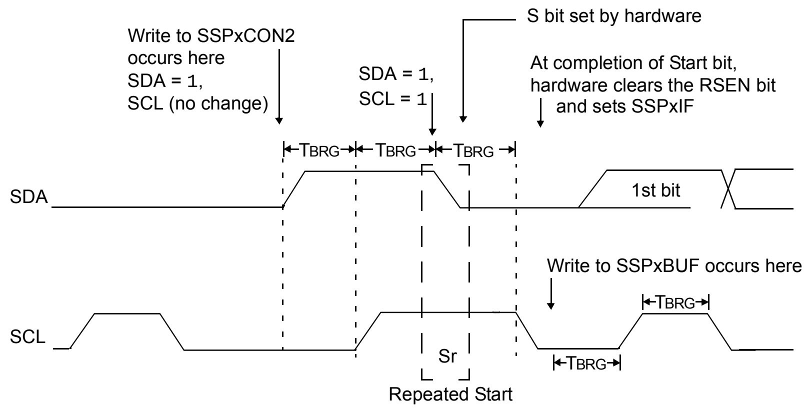


TABLE 31-1: ADC CLOCK PERIOD (TAD) VS. DEVICE OPERATING FREQUENCIES(1,4)

ADC Clock Period (TAD)		Device Frequency (FOSC)						
ADC Clock Source	ADCS<5:0>	64 MHz	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
FOSC/2	000000	31.25 ns(2)	62.5 ns(2)	100 ns(2)	125 ns(2)	250 ns(2)	500 ns(2)	2.0 μs
FOSC/4	000001	62.5 ns(2)	125 ns(2)	200 ns(2)	250 ns(2)	500 ns(2)	1.0 μs	4.0 μs
FOSC/6	000010	125 ns(2)	187.5 ns(2)	300 ns(2)	375 ns(2)	750 ns(2)	1.5 μs	6.0 μs
FOSC/8	000011	187.5 ns(2)	250 ns(2)	400 ns(2)	500 ns(2)	1.0 μs	2.0 μs	8.0 μs(3)
...	...	...	...	...	...	...	...	...
FOSC/16	000100	250 ns(2)	500 ns(2)	800 ns(2)	1.0 μs	2.0 μs	4.0 μs	16.0 μs(3)
...	...	...	...	...	...	...	...	...
FOSC/128	111111	2.0 μs	4.0 μs	6.4 μs	8.0 μs	16.0 μs(3)	32.0 μs(2)	128.0 μs(2)
FRC	ADCS(ADCON0<4>) = 1	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs

Legend: Shaded cells are outside of recommended range.

Note 1: See TAD parameter for FRC source typical TAD value.

2: These values violate the required TAD time.

3: Outside the recommended TAD time.

4: The ADC clock period (TAD) and total ADC conversion time can be minimized when the ADC clock is derived from the system clock FOSC. However, the FRC oscillator source must be used when conversions are to be performed with the device in Sleep mode.

FIGURE 31-2: ANALOG-TO-DIGITAL CONVERSION TAD CYCLES

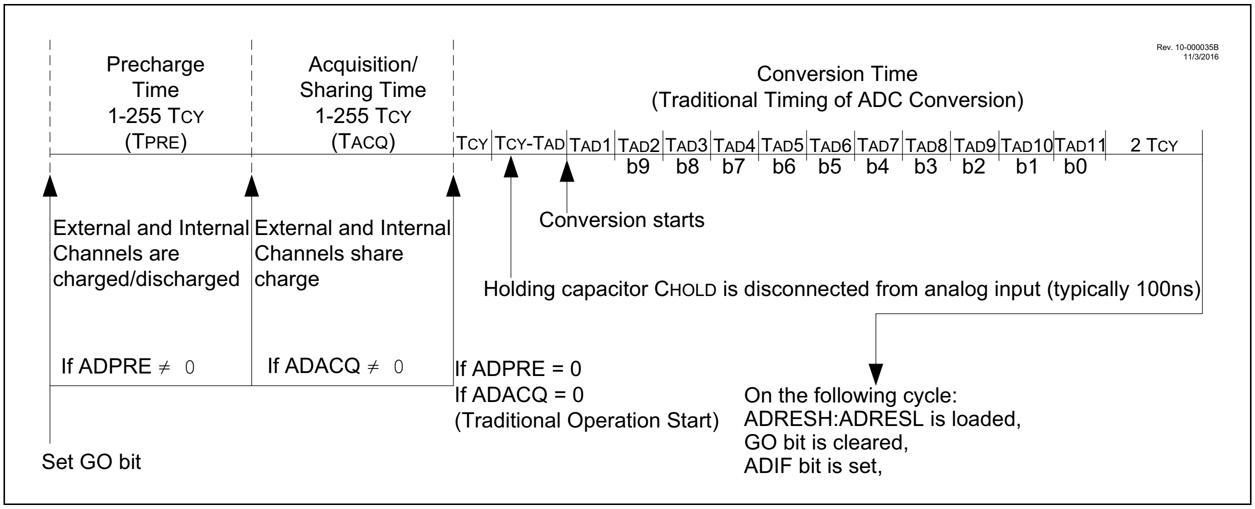
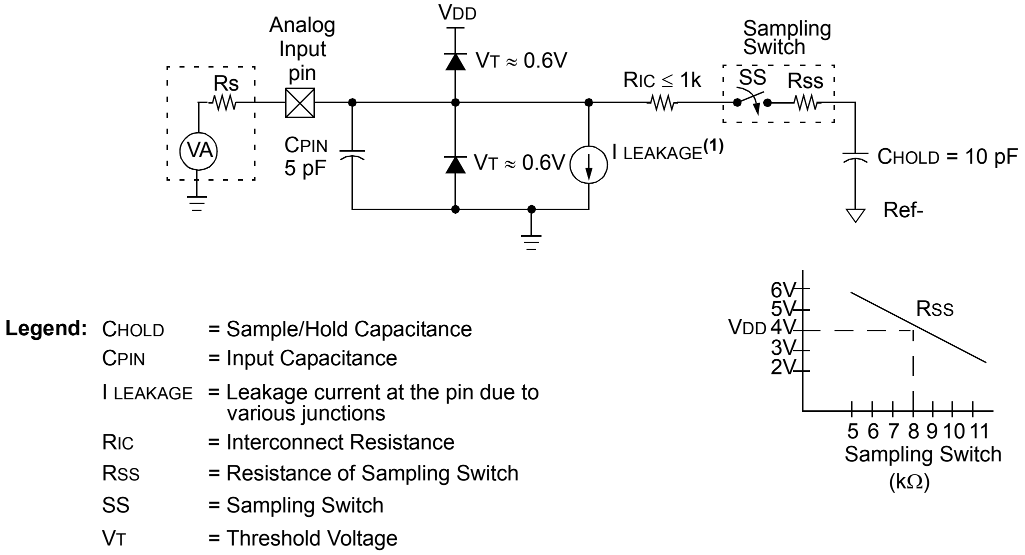


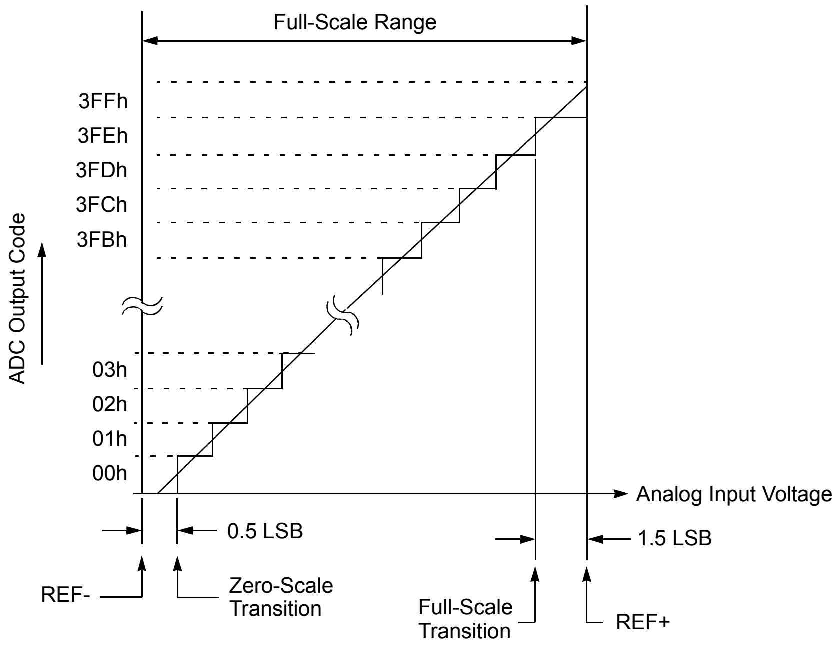
FIGURE 31-4: ANALOG INPUT MODEL



Legend:	CHOLD	= Sample/Hold Capacitance
	CPIN	= Input Capacitance
	I LEAKAGE	= Leakage current at the pin due to various junctions
	RIC	= Interconnect Resistance
	RSS	= Resistance of Sampling Switch
	SS	= Sampling Switch
	VT	= Threshold Voltage

Note 1: Refer to Table 37-4 (parameter D340 and D341).

FIGURE 31-5: ADC TRANSFER FUNCTION



BRA Unconditional Branch

Syntax: BRA n

Operands: -1024 ≤ n ≤ 1023

Operation: (PC) + 2 + 2n → PC

Status Affected: None

Encoding:

`1101`	`0nnn`	`nnnn`	`nnnn`

Description: Add the 2's complement number '2n' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is a 2-cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

Example: `HERE BRA Jump`

Before Instruction
PC = address (`HERE`)
After Instruction
PC = address (`Jump`)

BSF Bit Set f

Syntax: BSF f, b {,a}

Operands: 0 ≤ f ≤ 255
0 ≤ b ≤ 7
a ∈ [0,1]

Operation: 1 → f<b>

Status Affected: None

Encoding:

`1000`	`bbba`	`ffff`	`ffff`

Description: Bit 'b' in register 'f' is set.
If 'a' is '`0`', the Access Bank is selected. If 'a' is '`1`', the BSR is used to select the GPR bank.
If 'a' is '`0`' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 35.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example: `BSF FLAG_REG, 7, 1`

Before Instruction
FLAG_REG = 0Ah
After Instruction
FLAG_REG = 8Ah

SUBWFB Subtract W from f with Borrow

Syntax: SUBWFB f {,d {,a}}

Operands: $0 \leq f \leq 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: (f) – (W) – ($\overline{C}$) → dest

Status Affected: N, OV, C, DC, Z

Encoding:

`0101`	`10da`	`ffff`	`ffff`

Description: Subtract W and the CARRY flag (borrow) from register ‘f’ (2’s complement method). If ‘d’ is ‘`0`’, the result is stored in W. If ‘d’ is ‘`1`’, the result is stored back in register ‘f’ (default).
If ‘a’ is ‘`0`’, the Access Bank is selected. If ‘a’ is ‘`1`’, the BSR is used to select the GPR bank.
If ‘a’ is ‘`0`’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 35.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write to destination

Example 1: `SUBWFB REG, 1, 0`

```
Before Instruction
    REG   =   19h    (0001 1001)
    W     =   0Dh    (0000 1101)
    C     =   1
After Instruction
    REG   =   0Ch    (0000 1100)
    W     =   0Dh    (0000 1101)
    C     =   1
    Z     =   0
    N     =   0      ; result is positive
```

Example 2: `SUBWFB REG, 0, 0`

```
Before Instruction
    REG   =   1Bh    (0001 1011)
    W     =   1Ah    (0001 1010)
    C     =   0
After Instruction
    REG   =   1Bh    (0001 1011)
    W     =   00h
    C     =   1
    Z     =   1      ; result is zero
    N     =   0
```

Example 3: `SUBWFB REG, 1, 0`

```
Before Instruction
    REG   =   03h    (0000 0011)
    W     =   0Eh    (0000 1110)
    C     =   1
After Instruction
    REG   =   F5h    (1111 0101)
                     ; [2's comp]
    W     =   0Eh    (0000 1110)
    C     =   0
    Z     =   0
    N     =   1      ; result is negative
```

SWAPF Swap f

Syntax: SWAPF f {,d {,a}}

Operands: $0 \leq f \leq 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: (f<3:0>) → dest<7:4>,
(f<7:4>) → dest<3:0>

Status Affected: None

Encoding:

`0011`	`10da`	`ffff`	`ffff`

Description: The upper and lower nibbles of register ‘f’ are exchanged. If ‘d’ is ‘`0`’, the result is placed in W. If ‘d’ is ‘`1`’, the result is placed in register ‘f’ (default).
If ‘a’ is ‘`0`’, the Access Bank is selected. If ‘a’ is ‘`1`’, the BSR is used to select the GPR bank.
If ‘a’ is ‘`0`’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 35.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write to destination

Example: `SWAPF REG, 1, 0`

```
Before Instruction
    REG   =   53h
After Instruction
    REG   =   35h
```

TABLE 37-7: EXTERNAL CLOCK/OSCILLATOR TIMING REQUIREMENTS (CONTINUED)

Standard Operating Conditions (unless otherwise stated)

Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
OS21	F_{CY}	Instruction Frequency	—	FOSC/4	—	MHz	
OS22	T_{CY}	Instruction Period	62.5	$1/F_{CY}$	—	ns	

* These parameters are characterized but not tested.

† 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: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min” values with an external clock applied to OSC1 pin. When an external clock input is used, the “max” cycle time limit is “DC” (no clock) for all devices.

2: The system clock frequency (FOSC) is selected by the “main clock switch controls” as described in **Section 6.0 “Power-Saving Operation Modes”**.

3: The system clock frequency (FOSC) must meet the voltage requirements defined in the **Section 37.2 “Standard Operating Conditions”**.

4: LP, XT and HS oscillator modes require an appropriate crystal or resonator to be connected to the device. For clocking the device with the external square wave, one of the EC mode selections must be used.