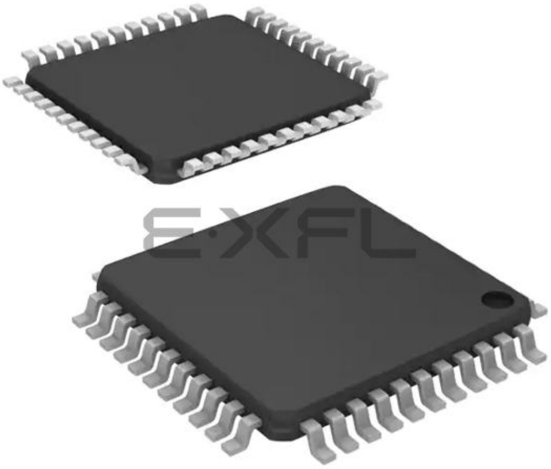


Welcome to E-XFL.COM

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	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	36
Program Memory Size	14KB (8K x 14)
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 35x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f18875-i-pt

TABLE 3-11: PIC16(L)F18875 MEMORY MAP, BANK 30

	Bank 30
F0Ch	—
F0Dh	—
F0Eh	—
F0Fh	—
F10h	RA0PPS
F11h	RA1PPS
F12h	RA2PPS
F13h	RA3PPS
F14h	RA4PPS
F15h	RA5PPS
F16h	RA6PPS
F17h	RA7PPS
F18h	RB0PPS
F19h	RB1PPS
F1Ah	RB2PPS
F1Bh	RB3PPS
F1Ch	RB4PPS
F1Dh	RB5PPS
F1Eh	RB6PPS
F1Fh	RB7PPS
F20h	RC0PPS
F21h	RC1PPS
F22h	RC2PPS
F23h	RC3PPS
F24h	RC4PPS
F25h	RC5PPS
F26h	RC6PPS
F27h	RC7PPS
F28h F37h	—
F38h	ANSELA
F39h	WPUA
F3Ah	ODCONA
F3Bh	SLRCONA
F3Ch	INLVLA
F3Dh	IOCAP
F3Eh	IOCAN
F3Fh	IOCAF

	Bank 30
F40h	CCDNA
F41h	CCDPA
F42h	—
F43h	ANSELB
F44h	WPUB
F45h	ODCONB
F46h	SLRCONB
F47h	INLVLB
F48h	IOCBP
F49h	IOCBN
F4Ah	IOCBF
F4Bh	CCDNB
F4Ch	CCDPB
F4Dh	—
F4Eh	ANSELC
F4Fh	WPUC
F50h	ODCONC
F51h	SLRCONC
F52h	INLVLC
F53h	IOCCP
F54h	IOCCN
F55h	IOCCF
F56h	CCDNC
F57h	CCDPC
F58h	—
F59h	ANSELD
F5Ah	WPUD
F5Bh	ODCOND
F5Ch	SLRCOND
F5Dh	INLVLD
F5Eh	—
F5Fh	—
F60h	—
F61h	CCDND
F62h	CCDPD
F63h	—

	Bank 30
F64h	ANSELE
F65h	WPUE
F66h	ODCONE
F67h	SLRCONE
F68h	INLVLE
F69h	IOCEP
F6Ah	IOCEN
F6Bh	IOCEF
F6Ch	CCDNE
F6Dh	CCDPE
F6Eh	—
F6Fh	—

Legend: ▨ = Unimplemented data memory locations, read as ‘0’.

TABLE 3-13: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18855	PIC16(L)F18875	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
Bank 30 (Continued)													
F24h	RC4PPS			—	—	RC4PPS<5:0>						--00 0000	--uu uuuu
F25h	RC5PPS			—	—	RC5PPS<5:0>						--00 0000	--uu uuuu
F26h	RC6PPS			—	—	RC6PPS<5:0>						--00 0000	--uu uuuu
F27h	RC7PPS			—	—	RC7PPS<5:0>						--00 0000	--uu uuuu
F28h	RD0PPS	—	X	—	—	RD0PPS<5:0>						--00 0000	--uu uuuu
		X	—	Unimplemented								---- ----	---- ----
F29h	RD1PPS	—	X	—	—	RD1PPS<5:0>						--00 0000	--uu uuuu
		X	—	Unimplemented								---- ----	---- ----
F2Ah	RD2PPS	—	X	—	—	RD2PPS<5:0>						--00 0000	--uu uuuu
		X	—	Unimplemented								---- ----	---- ----
F2Bh	RD3PPS	—	X	—	—	RD3PPS<5:0>						--00 0000	--uu uuuu
		X	—	Unimplemented								---- ----	---- ----
F2Ch	RD4PPS	—	X	—	—	RD4PPS<5:0>						--00 0000	--uu uuuu
		X	—	Unimplemented								---- ----	---- ----
F2Dh	RD5PPS	—	X	—	—	RD5PPS<5:0>						--00 0000	--uu uuuu
		X	—	Unimplemented								---- ----	---- ----
F2Eh	RD6PPS	—	X	—	—	RD6PPS<5:0>						--00 0000	--uu uuuu
		X	—	Unimplemented								---- ----	---- ----
F2Fh	RD7PPS	—	X	—	—	RD7PPS<5:0>						--00 0000	--uu uuuu
		X	—	Unimplemented								---- ----	---- ----
F30h	RE0PPS	—	X	—	—	RE0PPS<5:0>						--00 0000	--uu uuuu
		X	—	Unimplemented								---- ----	---- ----

Legend: x = unknown, u = unchanged, q =depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.

Note 1: Register present on PIC16F18855/75 devices only.

2: Unimplemented, read as '1'.

TABLE 3-13: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18855	PIC16(L)F18875	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
Bank 30 (Continued)													
F5Ch	SLRCOND	—	X	SLRD7	SLRD6	SLRD5	SLRD4	SLRD3	SLRD2	SLRD1	SLRD0	1111 1111	1111 1111
		X	—	Unimplemented								---- ----	---- ----
F5Dh	INLVLD	—	X	INLVLD7	INLVLD6	INLVLD5	INLVLD4	INLVLD3	INLVLD2	INLVLD1	INLVLD0	1111 1111	1111 1111
		X	—	Unimplemented								---- ----	---- ----
F5Eh — F60h	—	—		Unimplemented								—	—
F61h	CCDND	—	X	CCDND7	CCDND6	CCDND5	CCDND4	CCDND3	CCDND2	CCDND1	CCDND0	0000 0000	0000 0000
		X	—	Unimplemented								---- ----	---- ----
F62h	CCDPD	—	X	CCDPD7	CCDPD6	CCDPD5	CCDPD4	CCDPD3	CCDPD2	CCDPD1	CCDPD0	0000 0000	0000 0000
		X	—	Unimplemented								---- ----	---- ----
F63h	—	—		Unimplemented								—	—
F64h	ANSELE	—	X	—	—	—	—	—	ANSE2	ANSE1	ANSE0	---- -111	---- -111
		X	—	Unimplemented								---- ----	---- ----
F65h	WPUE	—	X	—	—	—	—	WPUE3	WPUE2	WPUE1	WPUE0	---- 0000	---- 0000
		X	—	—	—	—	—	WPUE3	—	—	—	---- 0---	---- 0---
F66h	ODCONE	—	X	—	—	—	—	—	ODCE2	ODCE1	ODCE0	---- -000	---- -000
		X	—	Unimplemented								---- ----	---- ----
F67h	SLRCONE	—	X	—	—	—	—	—	SLRE2	SLRE1	SLRE0	---- -111	---- -111
		X	—	Unimplemented								---- ----	---- ----
F68h	INLVLE	—	X	—	—	—	—	INLVLE3	INLVLE2	INLVLE1	INLVLE0	---- 1111	---- 1111
		X	—	—	—	—	—	INLVLE3	—	—	—	---- 1---	---- 1---
F69h	IOCEP			—	—	—	—	IOCEP3	—	—	—	---- 0---	---- 0---
F6Ah	IOCEN			—	—	—	—	IOCEN3	—	—	—	---- 0---	---- 0---

Legend: x = unknown, u = unchanged, q =depends on condition, - = unimplemented, read as ‘0’, r = reserved. Shaded locations unimplemented, read as ‘0’.

Note 1: Register present on PIC16F18855/75 devices only.

2: Unimplemented, read as ‘1’.

4.0 DEVICE CONFIGURATION

Device configuration consists of Configuration Words, Code Protection and Device ID.

4.1 Configuration Words

There are several Configuration Word bits that allow different oscillator and memory protection options. These are implemented as shown in Table 4-1.

TABLE 4-1: CONFIGURATION WORD LOCATIONS

Configuration Word	Location
CONFIG1	8007h
CONFIG2	8008h
CONFIG3	8009h
CONFIG4	800Ah
CONFIG5	800Bh

Note: The $\overline{\text{DEBUG}}$ bit in Configuration Words is managed automatically by device development tools including debuggers and programmers. For normal device operation, this bit should be maintained as a ‘1’.

REGISTER 7-11: PIR0: PERIPHERAL INTERRUPT STATUS REGISTER 0

U-0	U-0	R/W/HS-0/0	R-0	U-0	U-0	U-0	R/W/HS-0/0
—	—	TMR0IF	IOCIF	—	—	—	INTF[(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	HS= Hardware Set

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

bit 5 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 4 **IOCIF:** Interrupt-on-Change Interrupt Flag bit (read-only)[(2)]
1 = One or more of the IOCAF-IOCEF register bits are currently set, indicating an enabled edge was detected by the IOC module.
0 = None of the IOCAF-IOCEF register bits are currently set

bit 3-1 **Unimplemented:** Read as ‘0’

bit 0 **INTF:** INT External Interrupt Flag bit[(1)]
1 = The INT external interrupt occurred (must be cleared in software)
0 = The INT external interrupt did not occur

Note 1: The External Interrupt GPIO pin is selected by INTPPS (Register 13-1).
2: The IOCIF bits are the logical OR of all the IOCAF-IOCEF flags. Therefore, to clear the IOCIF flag, application firmware should clear all of the lower level IOCAF-IOCEF register bits.

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

12.5 Register Definitions: PORTA

REGISTER 12-2: PORTA: PORTA 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
RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
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 **RA<7:0>**: PORTA I/O Value bits[(1)]
1 = Port pin is ≥ V_{IH}
0 = Port pin is ≤ V_{IL}

Note 1: Writes to PORTA are actually written to corresponding LATA register. Reads from PORTA register is return of actual I/O pin values.

REGISTER 12-3: TRISA: PORTA TRI-STATE REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
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 **TRISA<7:0>:** PORTA Tri-State Control bit
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

REGISTER 12-6: WPUA: WEAK PULL-UP PORTA 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
WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
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 **WPUA<7:0>**: Weak Pull-up Register bits[(1)]
1 = Pull-up enabled
0 = Pull-up disabled

Note 1: The weak pull-up device is automatically disabled if the pin is configured as an output.

REGISTER 12-7: ODCONA: PORTA OPEN-DRAIN CONTROL 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
ODCA7	ODCA6	ODCA5	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0
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 **ODCA<7:0>:** PORTA Open-Drain Enable bits
For RA<7:0> pins, respectively
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

REGISTER 12-36: WPUD: WEAK PULL-UP PORTD 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
WPUD7	WPUD6	WPUD5	WPUD4	WPUD3	WPUD2	WPUD1	WPUD0
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 **WPUD<7:0>**: WPUD I/O Value bits(1)
1 = Port pin is $\geq$ VIH
0 = Port pin is $\leq$ VIL

Note 1: The weak pull-up device is automatically disabled if the pin is configured as an output.

REGISTER 12-37: ODCOND: PORTD OPEN-DRAIN CONTROL 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
ODCD7	ODCD6	ODCD5	ODCD4	ODCD3	ODCD2	ODCD1	ODCD0
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 **ODCD<7:0>**: ODCD I/O Value bits
1 = Port pin is $\geq$ VIH
0 = Port pin is $\leq$ VIL

REGISTER 12-38: SLRCOND: PORTD SLEW RATE CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
SLRD7	SLRD6	SLRD5	SLRD4	SLRD3	SLRD2	SLRD1	SLRD0
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 **SLRD<7:0>:** PORTD Slew Rate Enable bits
For RD<7:0> pins, respectively
`1` = Port pin slew rate is limited
`0` = Port pin slews at maximum rate

REGISTER 12-39: INLVLD: PORTD INPUT LEVEL CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
INLVLD7	INLVLD6	INLVLD5	INLVLD4	INLVLD3	INLVLD2	INLVLD1	INLVLD0
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 **INLVLD<7:0>:** PORTD Input Level Select bits
For RD<7:0> pins, respectively
`1` = ST input used for PORT reads and interrupt-on-change
`0` = TTL input used for PORT reads and interrupt-on-change

20.1.2 PUSH-PULL MODE

In Push-Pull mode, two output signals are generated, alternating copies of the input as illustrated in Figure 20-2. This alternation creates the push-pull effect required for driving some transformer-based power supply designs.

The push-pull sequencer is reset whenever EN = 0 or if an auto-shutdown event occurs. The sequencer is clocked by the first input pulse, and the first output appears on CWGxA.

The unused outputs CWGxC and CWGxD drive copies of CWGxA and CWGxB, respectively, but with polarity controlled by the POLC and POLD bits of the CWGxCON1 register, respectively.

20.1.3 FULL-BRIDGE MODES

In Forward and Reverse Full-Bridge modes, three outputs drive static values while the fourth is modulated by the input data signal. In Forward Full-Bridge mode, CWGxA is driven to its active state, CWGxB and CWGxC are driven to their inactive state, and CWGxD is modulated by the input signal. In Reverse Full-Bridge mode, CWGxC is driven to its active state, CWGxA and CWGxD are driven to their inactive states, and CWGxB is modulated by the input signal. 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 20.5 “Dead-Band Control”**, with additional details in **Section 20.6 “Rising Edge and Reverse Dead Band”** and **Section 20.7 “Falling Edge and Forward Dead Band”**.

The mode selection may be toggled between forward and reverse toggling the MODE<0> bit of the CWGxCON0 while keeping MODE<2:1> static, without disabling the CWG module.

FIGURE 20-13: SHUTDOWN FUNCTIONALITY, AUTO-RESTART DISABLED (REN = `0`, LSAC = `01`, LSBD = `01`)

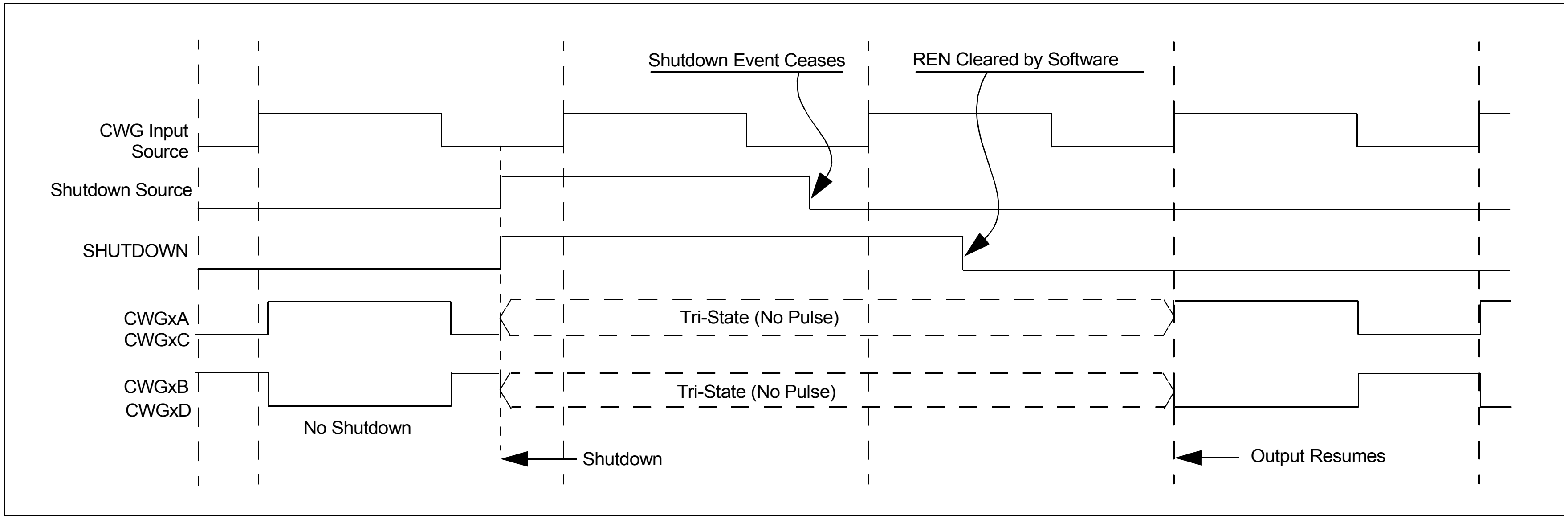
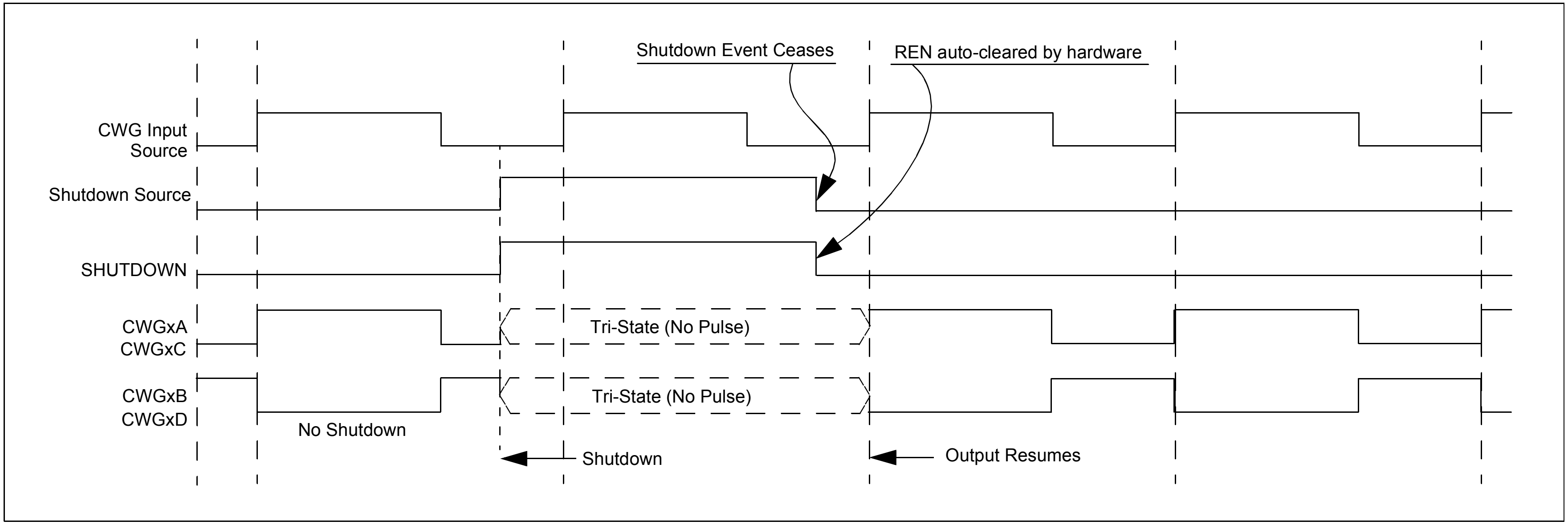


FIGURE 20-14: SHUTDOWN FUNCTIONALITY, AUTO-RESTART ENABLED (REN = `1`, LSAC = `01`, LSBD = `01`)



REGISTER 23-3: ADCON2: ADC CONTROL REGISTER 2

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W/HC-0	R/W-0/0	R/W-0/0	R/W-0/0
ADPSIS	ADCRS<2:0>			ADACLR	ADMD<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 **ADPSIS:** ADC Previous Sample Input Select bits

1 = ADFLTR is transferred to ADPREV at start-of-conversion
0 = ADRES is transferred to ADPREV at start-of-conversion

bit 6-4 **ADCRS<2:0>:** ADC Accumulated Calculation Right Shift Select bits

111 = Reserved
110 = Reserved
101 through 000:
If ADMD = 100:
Low-pass filter time constant is 2^{ADCRS}, filter gain is 1:1
If ADMD = 001, 010 or 011:
The accumulated value is right-shifted by ADCRS (divided by 2^{ADCRS})[2]
Otherwise:
Bits are ignored

bit 3 **ADACLR:** ADC Accumulator Clear Command bit

1 = Initial clear of ADACC, ADAOV, and the sample counter. Bit is cleared by hardware.
0 = Clearing action is complete (or not started)

bit 2-0 **ADMD<2:0>:** ADC Operating Mode Selection bits[1]

111 = Reserved
•
•
•
101 = Reserved
100 = Low-pass Filter mode
011 = Burst Average mode
010 = Average mode
001 = Accumulate mode
000 = Basic (Legacy) mode

Note 1: See Table 23-3 for Full mode descriptions.
2: All results of divisions using the ADCRS bits are truncated, not rounded.

REGISTER 23-30: ADUTHH: ADC UPPER THRESHOLD 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
ADUTH<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 **ADUTH<15:8>**: ADC Upper Threshold MSB. ADLTH and ADUTH are compared with ADERR to set the ADUTHR and ADLTHR bits of ADSTAT. Depending on the setting of ADTMD, an interrupt may be triggered by the results of this comparison.

REGISTER 23-31: ADUTHL: ADC UPPER THRESHOLD 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	R/W-x/x
ADUTH<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 **ADUTH<7:0>**: ADC Upper Threshold LSB. ADLTH and ADUTH are compared with ADERR to set the ADUTHR and ADLTHR bits of ADSTAT. Depending on the setting of ADTMD, an interrupt may be triggered by the results of this comparison.

REGISTER 26-4: MDCARH: MODULATION HIGH CARRIER CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	MDCHS<3:0>[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 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **MDCHS<3:0>** Modulator Data High Carrier Selection bits [1]

1111 =	LC4_out
1110 =	LC3_out
1101 =	LC2_out
1100 =	LC1_out
1011 =	NCO output
1010 =	PWM7_out
1001 =	PWM6_out
1000 =	CCP5 output (PWM Output mode only)
0111 =	CCP4 output (PWM Output mode only)
0110 =	CCP3 output (PWM Output mode only)
0101 =	CCP2 output (PWM Output mode only)
0100 =	CCP1 output (PWM Output mode only)
0011 =	Reference clock module signal (CLKR)
0010 =	HFINTOSC
0001 =	FOSC
0000 =	Pin selected by MDCARHPPS

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

29.0 TIMER2/4/6 MODULE

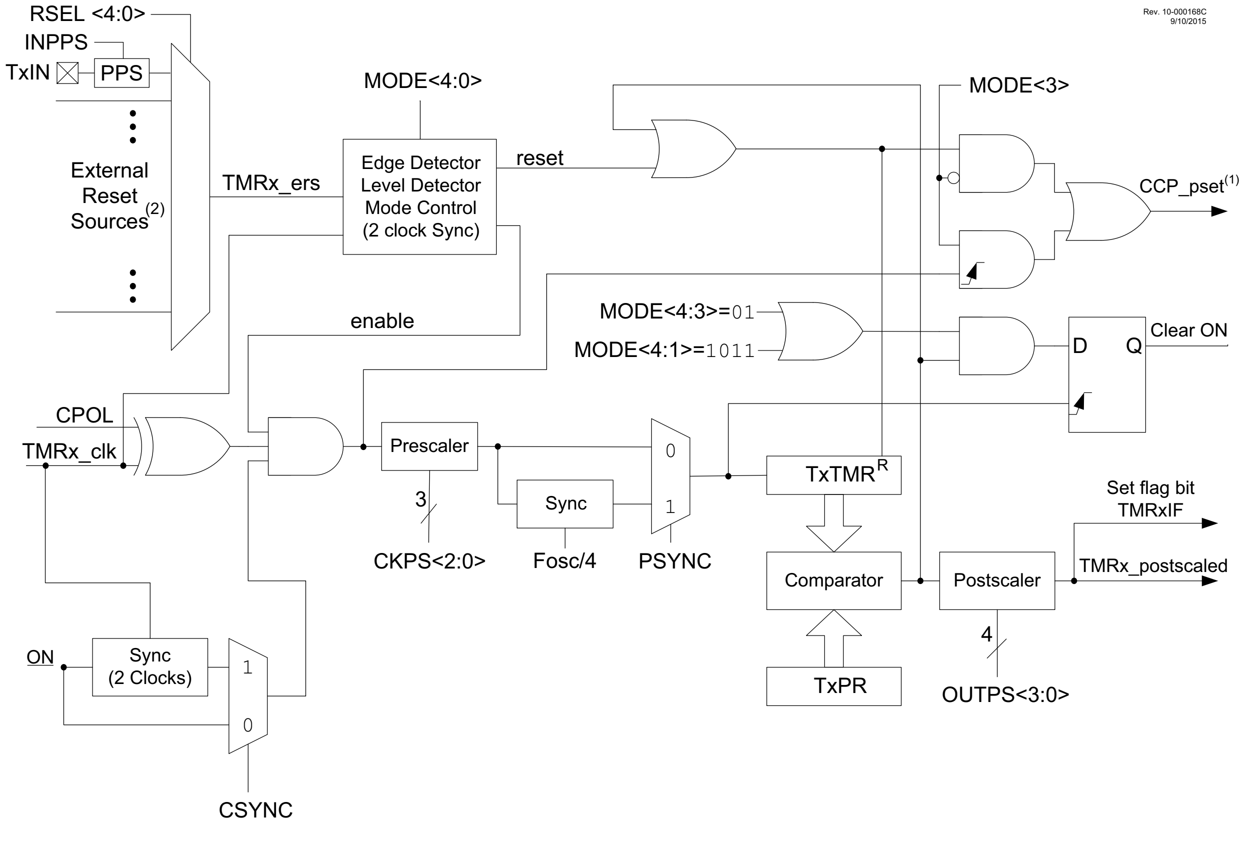
The Timer2/4/6 modules are 8-bit timers that can operate as free-running period counters or in conjunction with external signals that control start, run, freeze, and reset operation in One-Shot and Monostable modes of operation. Sophisticated waveform control such as pulse density modulation are possible by combining the operation of these timers with other internal peripherals such as the comparators and CCP modules. Features of the timer include:

- 8-bit timer register
- 8-bit period register
- Selectable external hardware timer Resets
- Programmable prescaler (1:1 to 1:128)
- Programmable postscaler (1:1 to 1:16)
- Selectable synchronous/asynchronous operation
- Alternate clock sources
- Interrupt-on-period
- Three modes of operation:
  - Free Running Period
  - One-shot
  - Monostable

See Figure 29-1 for a block diagram of Timer2. See Figure 29-2 for the clock source block diagram.

Note:	Three identical Timer2 modules are implemented on this device. The timers are named Timer2, Timer4, and Timer6. All references to Timer2 apply as well to Timer4 and Timer6. All references to T2PR apply as well to T4PR and T6PR.

FIGURE 29-1: TIMER2 BLOCK DIAGRAM



Note 1: Signal to the CCP to trigger the PWM pulse.
2: See Table 29.5 for description of CCP interaction in the different TMR modes.
3: See Register 29-4 for external Reset sources.

REGISTER 30-1: CCPxCON: CCPx CONTROL REGISTER (CONTINUED)

bit 3-0 **MODE<3:0>:** CCPx Mode Select bits[(1)]

1111 = PWM mode
1110 = Reserved
1101 = Reserved
1100 = Reserved

1011 = Compare mode: output will pulse 0-1-0; Clears TMR1
1010 = Compare mode: output will pulse 0-1-0
1001 = Compare mode: clear output on compare match
1000 = Compare mode: set output on compare match

0111 = Capture mode: every 16th rising edge of CCPx input
0110 = Capture mode: every 4th rising edge of CCPx input
0101 = Capture mode: every rising edge of CCPx input
0100 = Capture mode: every falling edge of CCPx input

0011 = Capture mode: every edge of CCPx input
0010 = Compare mode: toggle output on match
0001 = Compare mode: toggle output on match; clear TMR1
0000 = Capture/Compare/PWM off (resets CCPx module)

Note 1: All modes will set the CCPxIF bit, and will trigger an ADC conversion if CCPx is selected as the ADC trigger source.

33.3.1 AUTO-BAUD DETECT

The EUSART module supports automatic detection and calibration of the baud rate.

In the Auto-Baud Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RX signal, the RX signal is timing the BRG. The Baud Rate Generator is used to time the period of a received 55h (ASCII "U") which is the Sync character for the LIN bus. The unique feature of this character is that it has five rising edges including the Stop bit edge.

Setting the ABDEN bit of the BAUD1CON register starts the auto-baud calibration sequence. While the ABD sequence takes place, the EUSART state machine is held in Idle. On the first rising edge of the receive line, after the Start bit, the SPBRG begins counting up using the BRG counter clock as shown in Figure 33-6. The fifth rising edge will occur on the RX pin at the end of the eighth bit period. At that time, an accumulated value totaling the proper BRG period is left in the SPBRGH, SPBRGL register pair, the ABDEN bit is automatically cleared and the RCIF interrupt flag is set. The value in the RCREG needs to be read to clear the RCIF interrupt. RCREG content should be discarded. When calibrating for modes that do not use the SPBRGH register the user can verify that the SPBRGL register did not overflow by checking for 00h in the SPBRGH register.

The BRG auto-baud clock is determined by the BRG16 and BRGH bits as shown in Table 33-1. During ABD, both the SPBRGH and SPBRGL registers are used as a 16-bit counter, independent of the BRG16 bit setting. While calibrating the baud rate period, the SPBRGH and SPBRGL registers are clocked at 1/8th the BRG base clock rate. The resulting byte measurement is the average bit time when clocked at full speed.

Note 1: If the WUE bit is set with the ABDEN bit, auto-baud detection will occur on the byte following the Break character (see **Section 33.3.3 "Auto-Wake-up on Break"**).

2: It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible.

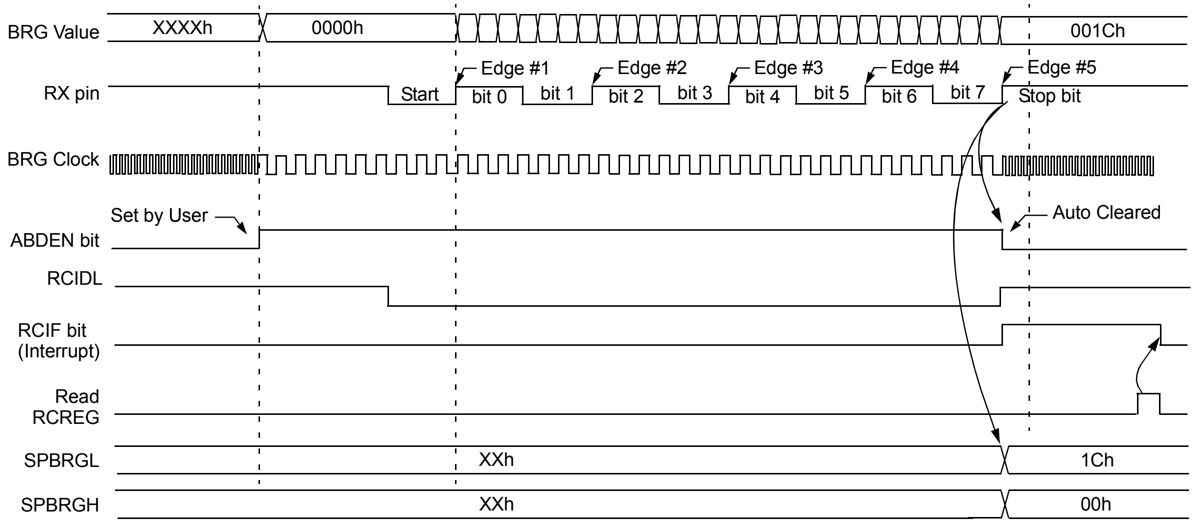
3: During the auto-baud process, the auto-baud counter starts counting at one. Upon completion of the auto-baud sequence, to achieve maximum accuracy, subtract 1 from the SPBRGH:SPBRGL register pair.

TABLE 33-1: BRG COUNTER CLOCK RATES

BRG16	BRGH	BRG Base Clock	BRG ABD Clock
0	0	FOSC/64	FOSC/512
0	1	FOSC/16	FOSC/128
1	0	FOSC/16	FOSC/128
1	1	FOSC/4	FOSC/32

Note: During the ABD sequence, SPBRGL and SPBRGH registers are both used as a 16-bit counter, independent of the BRG16 setting.

FIGURE 33-6: AUTOMATIC BAUD RATE CALIBRATION



Note 1: The ABD sequence requires the **EUSART** module to be configured in Asynchronous mode.

33.4.1.6 Slave Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a slave receives the clock on the TX/CK line. The TX/CK pin output driver is automatically disabled when the device is configured for synchronous slave transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One data bit is transferred for each clock cycle. Only as many clock cycles should be received as there are data bits.

Note:	If the device is configured as a slave and the TX/CK function is on an analog pin, the corresponding ANSEL bit must be cleared.

33.4.1.7 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before RCREG is read to access the FIFO. When this happens the OERR bit of the RC1STA register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear then the error is cleared by reading RCREG. If the overrun occurred when the CREN bit is set then the error condition is cleared by either clearing the CREN bit of the RC1STA register or by clearing the SPEN bit which resets the EUSART.

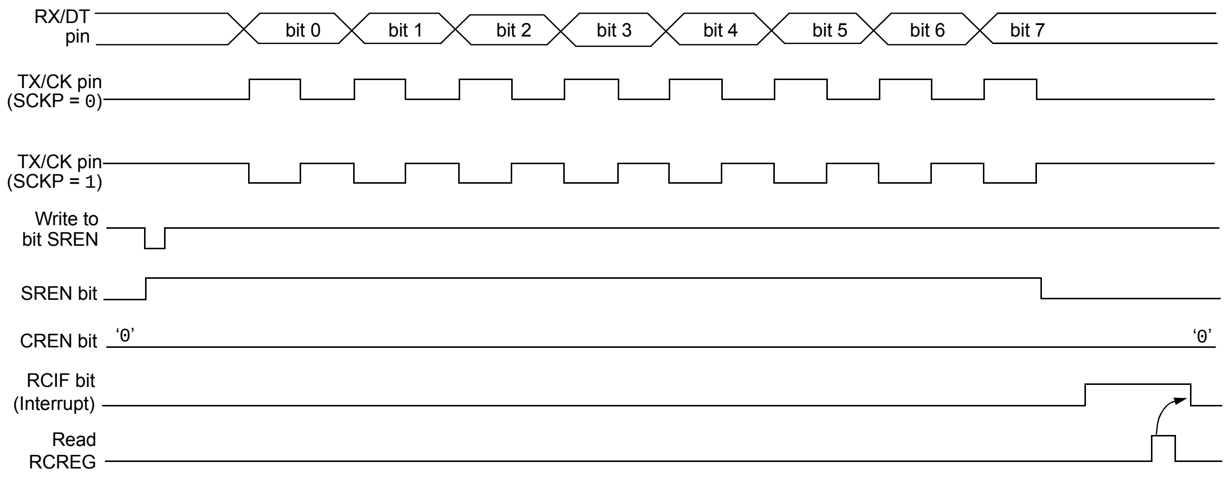
33.4.1.8 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RC1STA register is set the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RC1STA register is the ninth, and Most Significant, data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCREG.

33.4.1.9 Synchronous Master Reception Set-up:

1. Initialize the SPBRGH, SPBRGL register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Clear the ANSEL bit for the RX pin (if applicable).
3. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
4. Ensure bits CREN and SREN are clear.
5. If interrupts are desired, set the RCIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
6. If 9-bit reception is desired, set bit RX9.
7. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
8. Interrupt flag bit RCIF will be set when reception of a character is complete. An interrupt will be generated if the enable bit RCIE was set.
9. Read the RC1STA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
10. Read the 8-bit received data by reading the RCREG register.
11. If an overrun error occurs, clear the error by either clearing the CREN bit of the RC1STA register or by clearing the SPEN bit which resets the EUSART.

FIGURE 33-12: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



Note: Timing diagram demonstrates Sync Master mode with bit SREN = 1 and bit BRGH = 0.

FIGURE 34-1: CLOCK REFERENCE BLOCK DIAGRAM

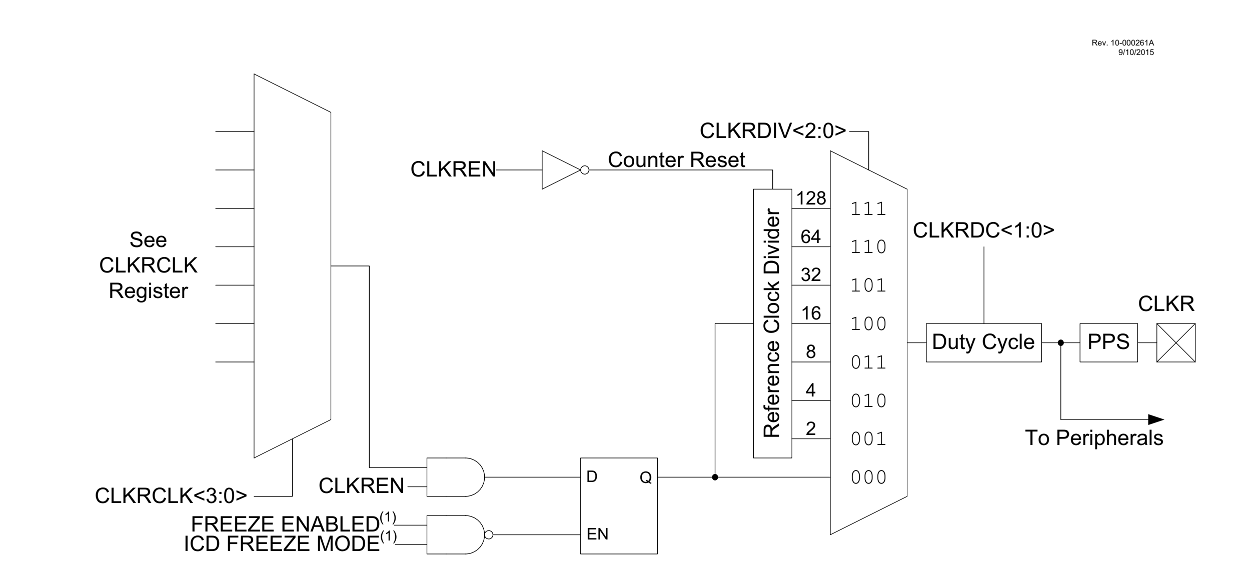
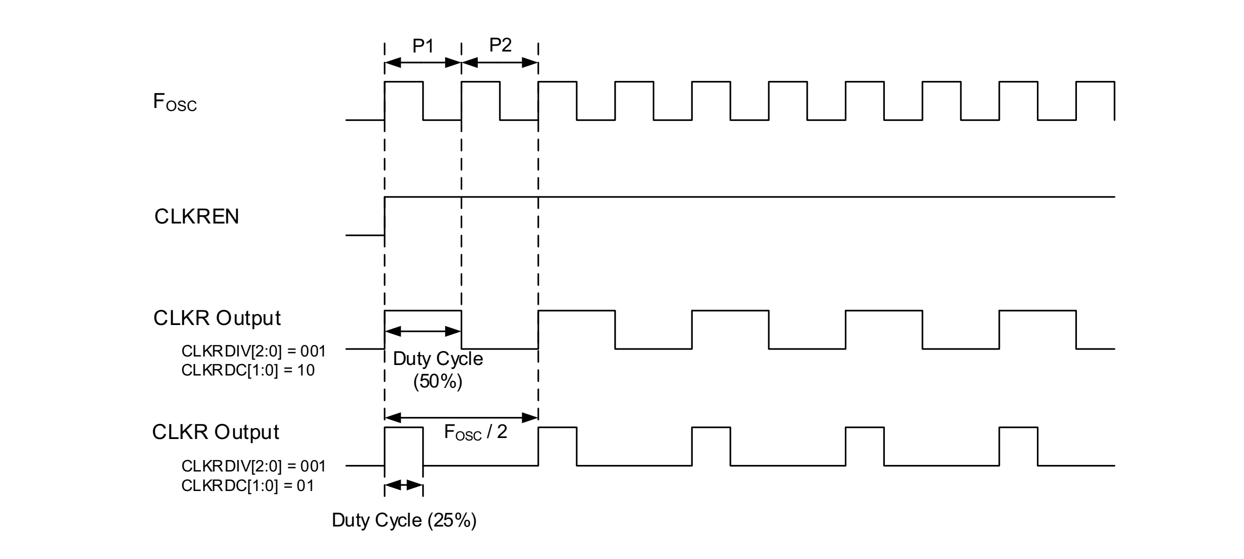


FIGURE 34-2: CLOCK REFERENCE TIMING



39.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

39.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker