Microchip Technology - PIC16F1769-I/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	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 12x10b; D/A 2x5b, 2x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1769-i-p

TABLE 3-13: PIC16(L)F1768/9 MEMORY MAP (BANKS 27-30)

	Bank 27		Bank 28		Bank 29		Bank 30
D8Ch	—	E0Ch	—	E8Ch	—	F0Ch	—
D8Dh	—	E0Dh	—	E8Dh	—	F0Dh	—
D8Eh	PWMEN	E0Eh	—	E8Eh	—	F0Eh	—
D8Fh	PWMLD	E0Fh	PPSLOCK	E8Fh	—	F0Fh	CLCDATA
D90h	PWMOUT	E10h	INTPPS	E90h	RA0PPS	F10h	CLC1CON
D91h	PWM5PHL	E11h	T0CKIPPS	E91h	RA1PPS	F11h	CLC1POL
D92h	PWM5PHH	E12h	T1CKIPPS	E92h	RA2PPS	F12h	CLC1SEL0
D93h	PWM5DCL	E13h	T1GPPS	E93h	—	F13h	CLC1SEL1
D94h	PWM5DCH	E14h	CCP1PPS	E94h	RA4PPS	F14h	CLC1SEL2
D95h	PWM5PRL	E15h	CCP2PPS	E95h	RA5PPS	F15h	CLC1SEL3
D96h	PWM5PRH	E16h	COG1INPPS	E96h	—	F16h	CLC1GLS0
D97h	PWM5OFL	E17h	COG2INPPS	E97h	—	F17h	CLC1GLS1
D98h	PWM5OFH	E18h	—	E98h	—	F18h	CLC1GLS2
D99h	PWM5TMRL	E19h	T2INPPS	E99h	—	F19h	CLC1GLS3
D9Ah	PWM5TMRH	E1Ah	T3CKIPPS	E9Ah	—	F1Ah	CLC2CON
D9Bh	PWM5CON	E1Bh	T3GPPS	E9Bh	—	F1Bh	CLC2POL
D9Ch	PWM5INTE	E1Ch	T4INPPS	E9Ch	RB4PPS	F1Ch	CLC2SEL0
D9Dh	PWM5INTF	E1Dh	T5CKIPPS	E9Dh	RB5PPS	F1Dh	CLC2SEL1
D9Eh	PWM5CLKCON	E1Eh	T5GPPS	E9Eh	RB6PPS	F1Eh	CLC2SEL2
D9Fh	PWM5LDCON	E1Fh	T6INPPS	E9Fh	RB7PPS	F1Fh	CLC2SEL3
DA0h	PWM5OFCON	E20h	SSPCLKPPS	EA0h	RC0PPS	F20h	CLC2GLS0
DA1h	PWM6PHL	E21h	SSPDATPPS	EA1h	RC1PPS	F21h	CLC2GLS1
DA2h	PWM6PHH	E22h	SSPSSPPS	EA2h	RC2PPS	F22h	CLC2GLS2
DA3h	PWM6DCL	E23h	—	EA3h	RC3PPS	F23h	CLC2GLS3
DA4h	PWM6DCH	E24h	RXPPS	EA4h	RC4PPS	F24h	CLC3CON
DA5h	PWM6PRL	E25h	CKPPS	EA5h	RC5PPS	F25h	CLC3POL
DA6h	PWM6PRH	E26h	—	EA6h	RC6PPS	F26h	CLC3SEL0
DA7h	PWM6OFL	E27h	—	EA7h	RC7PPS	F27h	CLC3SEL1
DA8h	PWM6OFH	E28h	CLCIN0PPS	EA8h	—	F28h	CLC3SEL2
DA9h	PWM6TMRL	E29h	CLCIN1PPS	EA9h	—	F29h	CLC3SEL3
DAAh	PWM6TMRH	E2Ah	CLCIN2PPS	EAAh	—	F2Ah	CLC3GLS0
DABh	PWM6CON	E2Bh	CLCIN3PPS	EABh	—	F2Bh	CLC3GLS1
DACh	PWM6INTE	E2Ch	PRG1FPPS	EACh	—	F2Ch	CLC3GLS2
DADh	PWM6INTF	E2Dh	PRG1RPPS	EADh	—	F2Dh	CLC3GLS3
DAEh	PWM6CLKCON	E2Eh	PRG2FPPS	EAEh	—	F2Eh	—
DAFh	PWM6LDCON	E2Fh	PRG2RPPS	EAFh	—	F2Fh	—
DB0h	PWM6OFCON	E30h	MD1CHPPS	EB0h	—	F30h	—
DB1h	—	E31h	MD1CLPPS	EB1h	—	F31h	—
DB2h	—	E32h	MD1MODPPS	EB2h	—	F32h	—
DB3h	—	E33h	MD2CHPPS	EB3h	—	F33h	—
DB4h	—	E34h	MD2CLPPS	EB4h	—	F34h	—
DB5h	—	E35h	MD2MODPPS	EB5h	—	F35h	—
DB6h	—	E36h	—	EB6h	—	F36h	—
DB7h	—	E37h	—	EB7h	—	F37h	—
DB8h	—	E38h	—	EB8h	—	F38h	—
DB9h	—	E39h	—	EB9h	—	F39h	—
DBAh	—	E3Ah	—	EBAh	—	F3Ah	—
DBBh	—	E3Bh	—	EBBh	—	F3Bh	—
DBCh	—	E3Ch	—	EBCh	—	F3Ch	—
DBDh	—	E3Dh	—	EBDh	—	F3Dh	—
DBEh	—	E3Eh	—	EBEh	—	F3Eh	—
DBFh	—	E3Fh	—	EBFh	—	F3Fh	—
DC0h	—	E40h	—	EC0h	—	F40h	—
DEFh		E6Fh		EEFh		F6Fh	

Legend: ▨ = Unimplemented data memory locations, read as '0',

TABLE 3-16: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Addr	Name	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 14											
70Ch	—	Unimplemented								—	—
70Dh	COG2PHR(2)	—	—	COG Rising Edge Phase Delay Count Register						`--00 0000`	`--00 0000`
70Eh	COG2PHF(2)	—	—	COG Falling Edge Phase Delay Count Register						`--00 0000`	`--00 0000`
70Fh	COG2BLKR(2)	—	—	COG Rising Edge Blanking Count Register						`--00 0000`	`--00 0000`
710h	COG2BLKF(2)	—	—	COG Falling Edge Blanking Count Register						`--00 0000`	`--00 0000`
711h	COG2DBR(2)	—	—	COG Rising Edge Dead-band Count Register						`--00 0000`	`--00 0000`
712h	COG2DBF(2)	—	—	COG Falling Edge Dead-band Count Register						`--00 0000`	`--00 0000`
713h	COG2CON0(2)	EN	LD	—	CS<1:0>		MD<2:0>			`00-0 0000`	`00-0 0000`
714h	COG2CON1(2)	RDBS	FDBS	—	—	POLD	POLC	POLB	POLA	`00-- 0000`	`00-- 0000`
715h	COG2RIS0(2)	RIS<7:0>								`0000 0000`	`0000 0000`
716h	COG2RIS1(2)	RIS<15:8>								`0000 0000`	`0000 0000`
717h	COG2RSIM0(2)	RSIM<7:0>								`0000 0000`	`0000 0000`
718h	COG2RSIM1(2)	RSIM<15:8>								`0000 0000`	`0000 0000`
719h	COG2FIS0(2)	FIS<7:0>								`0000 0000`	`0000 0000`
71Ah	COG2FIS1(2)	FIS<15:8>								`0000 0000`	`0000 0000`
71Bh	COG2FSIM0(2)	FSIM<7:0>								`0000 0000`	`0000 0000`
71Ch	COG2FSIM1(2)	FSIM<15:8>								`0000 0000`	`0000 0000`
71Dh	COG2ASD0(2)	ASE	ARSEN	ASDBD<1:0>		ASDAC<1:0>		—	—	`0001 01--`	`0001 01--`
71Eh	COG2ASD1(2)	AS7E	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	`0000 0000`	`0000 0000`
71Fh	COG2STR(2)	SDATD	SDATC	SDATB	SDATA	STRD	STRC	STRB	STRA	`0000 0000`	`0000 0000`
Bank 15											
78Ch — 793h	—	Unimplemented								—	—
794h	PRG1RTSS	—	—	—	—	RTSS<3:0>				`---- 0000`	`---- 0000`
795h	PRG1FTSS	—	—	—	—	FTSS<3:0>				`---- 0000`	`---- 0000`
796h	PRG1INS	—	—	—	—	INS<3:0>				`---- 0000`	`---- 0000`
797h	PRG1CON0	EN	—	FEDG	REDG	MODE<1:0>		OS	GO	`0-00 0000`	`0-00 0000`
798h	PRG1CON1	—	—	—	—	—	RDY	FPOL	RPOL	`---- -000`	`---- -000`
799h	PRG1CON2	—	—	—	ISET<4:0>					`---0 0000`	`---0 0000`
79Ah	PRG2RTSS(2)	—	—	—	—	RTSS<3:0>				`---- 0000`	`---- 0000`
79Bh	PRG2FTSS(2)	—	—	—	—	FTSS<3:0>				`---- 0000`	`---- 0000`
79Ch	PRG2INS(2)	—	—	—	—	INS<3:0>				`---- 0000`	`---- 0000`
79Dh	PRG2CON0(2)	EN	—	FEDG	REDG	MODE<1:0>		OS	GO	`0-00 0000`	`0-00 0000`
79Eh	PRG2CON1(2)	—	—	—	—	—	RDY	FPOL	RPOL	`---- -000`	`---- -000`
79Fh	PRG2CON2(2)	—	—	—	ISET<4:0>					`---0 0000`	`---0 0000`
Bank 16-26											
x0Ch/ x8Ch — x1Fh/ x9Fh	—	Unimplemented								—	—

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

Note 1: Unimplemented, read as '`1`'.
2: PIC16(L)F1768/9 only.
3: PIC16(L)F1764/5 only.
4: Unimplemented on PIC16LF1764/5/8/9.

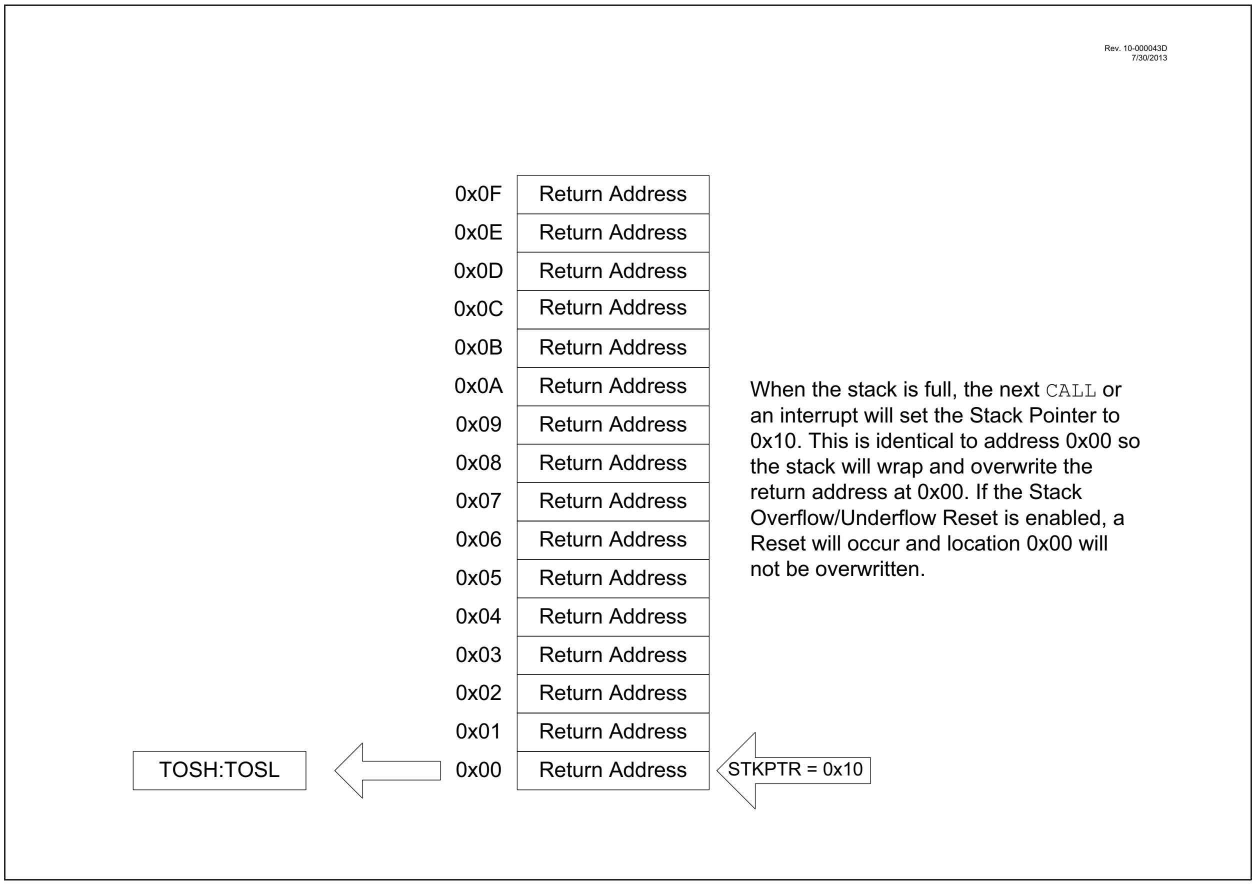
TABLE 3-16: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Addr	Name	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 28											
E0Ch — E0Eh	—	Unimplemented								—	—
E0Fh	PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	`---- ---0`	`---- ---0`
E10h	INTPPS	—	—	—	INTPPS<4:0>					`---0 0010`	`---u uuuu`
E11h	T0CKIPPS	—	—	—	T0CKIPPS<4:0>					`---0 0010`	`---u uuuu`
E12h	T1CKIPPS	—	—	—	T1CKIPPS<4:0>					`---0 0101`	`---u uuuu`
E13h	T1GPPS	—	—	—	T1GPPS<4:0>					`---0 0100`	`---u uuuu`
E14h	CCP1PPS	—	—	—	CCP1PPS<4:0>					`---1 0101`	`---u uuuu`
E15h	CCP2PPS(2)	—	—	—	CCP2PPS<4:0>					`---1 0011`	`---u uuuu`
E16h	COG1INPPS	—	—	—	COG1INPPS<4:0>					`---0 0010`	`---u uuuu`
E17h	COG2INPPS(2)	—	—	—	COG2INPPS<4:0>					`---0 0010`	`---u uuuu`
E18h	—	Unimplemented								—	—
E19h	T2INPPS	—	—	—	T2INPPS<4:0>					`---0 0101`	`---u uuuu`
E1Ah	T3CKIPPS	—	—	—	T3CKIPPS<4:0>					`---1 0101`	`---u uuuu`
E1Bh	T3GPPS	—	—	—	T3GPPS<4:0>					`---1 0100`	`---u uuuu`
E1Ch	T4INPPS	—	—	—	T4INPPS<4:0>					`---1 0001`	`---u uuuu`
E1Dh	T5CKIPPS	—	—	—	T5CKIPPS<4:0>					`---1 0000`	`---u uuuu`
E1Eh	T5GPPS	—	—	—	T5GPPS<4:0>					`---1 0011`	`---u uuuu`
E1Fh	T6INPPS	—	—	—	T6INPPS<4:0>					`---0 0011`	`---u uuuu`
E20h	SSPCLKPPS	—	—	—	SSPCLKPPS<4:0>					`---1 0000`(3)	`---u uuuu`
		—	—	—	SSPCLKPPS<4:0>					`---0 1110`(2)	`---u uuuu`
E21h	SSPDATPPS	—	—	—	SSPDATPPS<4:0>					`---1 0001`(3)	`---u uuuu`
		—	—	—	SSPDATPPS<4:0>					`---0 1100`(2)	`---u uuuu`
E22h	SSPSSPPS	—	—	—	SSPSSPPS<4:0>					`---1 0011`(3)	`---u uuuu`
		—	—	—	SSPSSPPS<4:0>					`---1 0110`(2)	`---u uuuu`
E23h	—	Unimplemented								—	—
E24h	RXPPS	—	—	—	RXPPS<4:0>					`---1 0101`(3)	`---u uuuu`
		—	—	—	RXPPS<4:0>					`---0 1101`(2)	`---u uuuu`
E25h	CKPPS	—	—	—	CKPPS<4:0>					`---1 0100`(3)	`---u uuuu`
		—	—	—	CKPPS<4:0>					`---0 1111`(2)	`---u uuuu`
E26h	—	Unimplemented								—	—
E27h	—	Unimplemented								—	—
E28h	CLCIN0PPS	—	—	—	CLCIN0PPS<4:0>					`---1 0011`	`---u uuuu`
E29h	CLCIN1PPS	—	—	—	CLCIN1PPS<4:0>					`---1 0100`	`---u uuuu`
E2Ah	CLCIN2PPS	—	—	—	CLCIN2PPS<4:0>					`---1 0001`	`---u uuuu`
E2Bh	CLCIN3PPS	—	—	—	CLCIN3PPS<4:0>					`---0 0101`	`---u uuuu`
E2Ch	PRG1RPPS	—	—	—	PRG1RPPS<4:0>					`---1 0100`	`---u uuuu`
E2Dh	PRG1FPPS	—	—	—	PRG1FPPS<4:0>					`---1 0101`	`---u uuuu`
E2Eh	PRG2RPPS(2)	—	—	—	PRG2RPPS<4:0>					`---1 0100`	`---u uuuu`
E2Fh	PRG2FPPS(2)	—	—	—	PRG2FPPS<4:0>					`---1 0101`	`---u uuuu`
E30h	MD1CHPPS	—	—	—	MD1CHPPS<4:0>					`---0 0011`	`---u uuuu`
E31h	MD1CLPPS	—	—	—	MD1CLPPS<4:0>					`---0 0100`	`---u uuuu`

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

Note 1: Unimplemented, read as '`1`'.
2: PIC16(L)F1768/9 only.
3: PIC16(L)F1764/5 only.
4: Unimplemented on PIC16LF1764/5/8/9.

FIGURE 3-7: ACCESSING THE STACK EXAMPLE 4



3.5.2 OVERFLOW/UNDERFLOW RESET

If the STVREN bit in Configuration Words is programmed to '1', the device will be reset if the stack is PUSHed beyond the sixteenth level or POPed beyond the first level, setting the appropriate bits (STKOVF or STKUNF, respectively) in the PCON register.

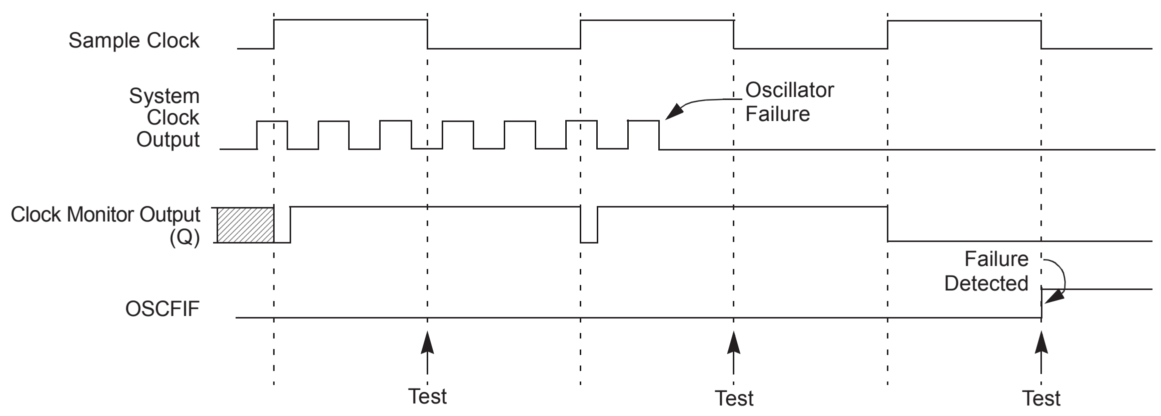
3.6 Indirect Addressing

The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the File Select Registers (FSRs). If the FSRn address specifies one of the two INDFn registers, the read will return '0' and the write will not occur (though Status bits may be affected). The FSRn register value is created by the pair, FSRnH and FSRnL.

The FSRn registers form a 16-bit address that allows an addressing space with 65536 locations. These locations are divided into three memory regions:

- Traditional Data Memory
- Linear Data Memory
- Program Flash Memory

FIGURE 5-10: 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.

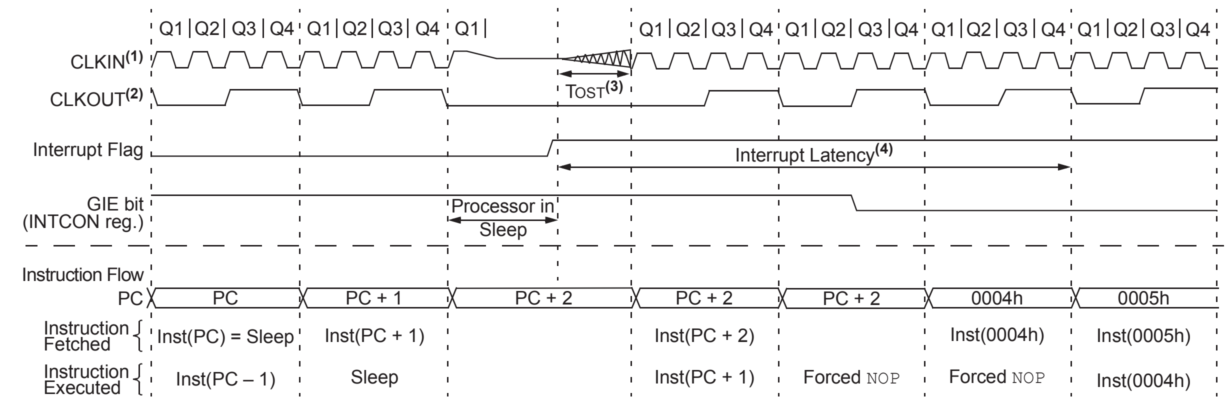
8.1.1 WAKE-UP USING INTERRUPTS

When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a `SLEEP` instruction:
  - `SLEEP` instruction will execute as a `NOP`
  - WDT and WDT prescaler will not be cleared
  - $\overline{\text{TO}}$ bit of the STATUS register will not be set
  - $\overline{\text{PD}}$ bit of the STATUS register will not be cleared
- If the interrupt occurs **during or after** the execution of a `SLEEP` instruction:
  - `SLEEP` instruction will be completely executed
  - Device will immediately wake-up from Sleep
  - WDT and WDT prescaler will be cleared
  - $\overline{\text{TO}}$ bit of the STATUS register will be set
  - $\overline{\text{PD}}$ bit of the STATUS register will be cleared

Even if the flag bits were checked before executing a `SLEEP` instruction, it may be possible for flag bits to become set before the `SLEEP` instruction completes. To determine whether a `SLEEP` instruction executed, test the $\overline{\text{PD}}$ bit. If the $\overline{\text{PD}}$ bit is set, the `SLEEP` instruction was executed as a `NOP`.

FIGURE 8-1: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: External clock. High, Medium, Low mode assumed.
2: CLKOUT is shown here for timing reference.
3: TOST = 1024 TOSC. This delay does not apply to EC, RC and INTOSC Oscillator modes or Two-Speed Start-up (see **Section 5.4 "Two-Speed Clock Start-up Mode"**.
4: GIE = `1` assumed. In this case, after wake-up, the processor calls the ISR at 0004h. If GIE = `0`, execution will continue in-line.

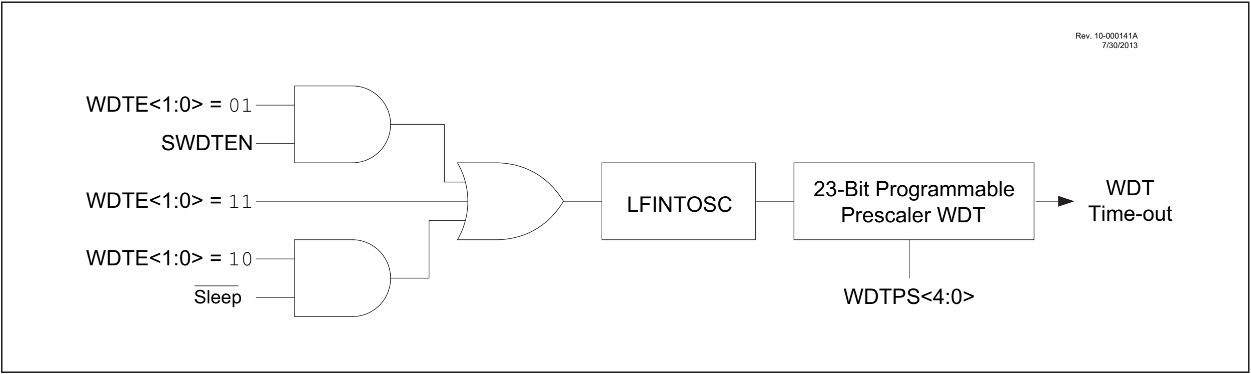
9.0 WATCHDOG TIMER (WDT)

The Watchdog Timer is a system timer that generates a Reset if the firmware does not issue a `CLRWDT` instruction within the time-out period. The Watchdog Timer is typically used to recover the system from unexpected events.

The WDT has the following features:

- Independent clock source
- Multiple operating modes:
  - WDT is always on
  - WDT is off when in Sleep
  - WDT is controlled by software
  - WDT is always off
- Configurable time-out period is from 1 ms to 256 seconds (nominal)
- Multiple Reset conditions
- Operation during Sleep

FIGURE 9-1: WATCHDOG TIMER BLOCK DIAGRAM



REGISTER 10-3: PMADRL: PROGRAM MEMORY 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
PMADR<7:0>							
bit 7							bit 0

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

bit 7-0 **PMADR<7:0>**: Specifies the Least Significant bits for Program Memory Address bits

REGISTER 10-4: PMADRH: PROGRAM MEMORY ADDRESS HIGH BYTE REGISTER

U-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
—[1]	PMADR<14:8>						
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as ‘1’

bit 6-0 **PMADR<14:8>**: Specifies the Most Significant bits for Program Memory Address bits

Note 1: Unimplemented, read as ‘1’.

REGISTER 11-13: WPUB: WEAK PULL-UP PORTB REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
WPUB<7:4>(1,2)				—	—	—	—
bit 7							bit 0

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

bit 7-4	**WPUB<7:4>**: Weak Pull-up PORTB Register bits(1,2) `1` = Pull-up is enabled `0` = Pull-up is disabled
bit 3-0	**Unimplemented:** Read as ‘`0`’

Note 1: The global $\overline{\text{WPUEN}}$ bit of the OPTION_REG register must be cleared for individual pull-ups to be enabled.

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

REGISTER 11-14: ODCONB: PORTB OPEN-DRAIN CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
ODB<7:4>				—	—	—	—
bit 7							bit 0

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

bit 7-4	**ODB<7:4>:** PORTB Open-Drain Enable bits For RB<7:4> Pins: `1` = Port pin operates as an open-drain drive (sink current only) `0` = Port pin operates as a standard push-pull drive (source and sink current)
bit 3-0	**Unimplemented:** Read as ‘`0`’

REGISTER 24-5: CCPxCAP: CCPx CAPTURE INPUT SELECTION REGISTER

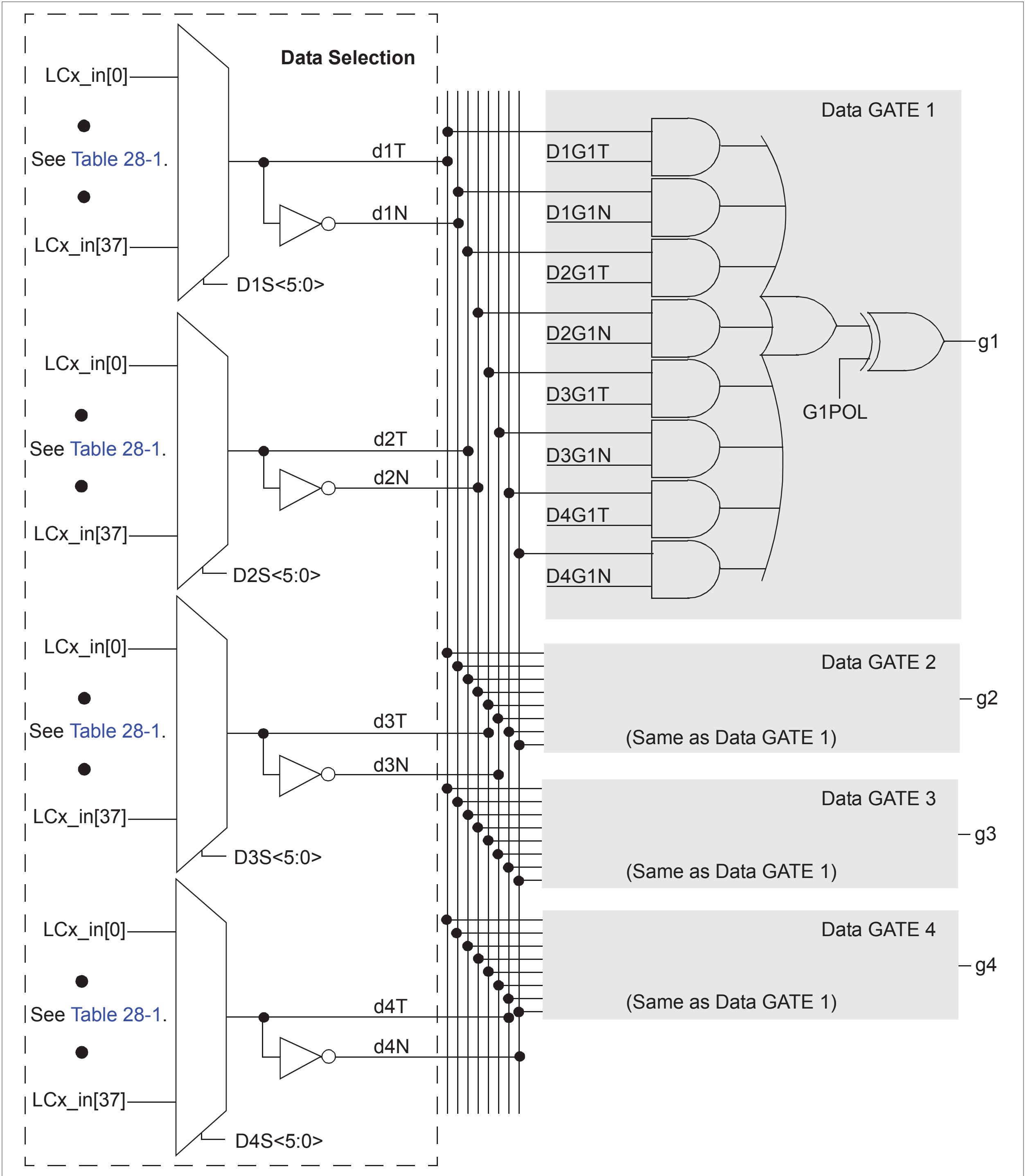
U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CTS<2:0>		
bit 7							bit 0

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

bit 7-3	**Unimplemented:** Read as ‘0’
bit 2-0	**CTS<2:0>:** Capture Trigger Input Selection bits
	111 = IOC_event
	110 = LC3_output
	101 = LC2_output
	100 = C4_sync_out(1)
	011 = C3_sync_out(1)
	010 = C2_sync_out
	001 = C1_sync_out
	000 = Pin selected with the CCPxPPS register

Note 1: PIC16(L)F1768/9 only. Unimplemented on PIC16(L)F1764/5.

FIGURE 28-2: INPUT DATA SELECTION AND GATING



Note: All controls are undefined at power-up.

REGISTER 31-3: MDxSRC: MODULATION x SOURCE CONTROL REGISTER

U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	MS<4:0>				
bit 7							bit 0

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

bit 7-5	**Unimplemented:** Read as ‘0’
bit 4-0	**MS<4:0>** Modulation Source Selection bits See Table 31-3.

TABLE 31-3: MODULATION SOURCE

MS<4:0>	Modulation Source PIC16(L)F1764/5	Modulation Source PIC16(L)F1768/9
11111-10100	Fixed Low	Fixed Low
10011	Fixed Low	sync_C4OUT
10010	Fixed Low	sync_C3OUT
10001	sync_C2OUT	sync_C2OUT
10000	sync_C1OUT	sync_C1OUT
01111	LC3_out	LC3_out
01110	LC2_out	LC2_out
01101	LC1_out	LC1_out
01100	Fixed Low	PWM6_out
01011	PWM5_out	PWM5_out
01010	Fixed Low	PWM4_out
01001	PWM3_out	PWM3_out
01000	Fixed low	CCP2_out
00111	CCP1_out	CCP1_out
00110	SDO_out	SDO_out
00101	Fixed Low	COG2A
00100	DT	DT
00011	TX_out	TX_out
00010	COG1A	COG1A
00001	MDxBIT	MDxBIT
00000	MDxMODPPS Pin Selection	MDxMODPPS Pin Selection

REGISTER 31-4: MDxCARH: MODULATION x CARRIER HIGH 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
—	—	—	—	CH<3:0>(1)			
bit 7							bit 0

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

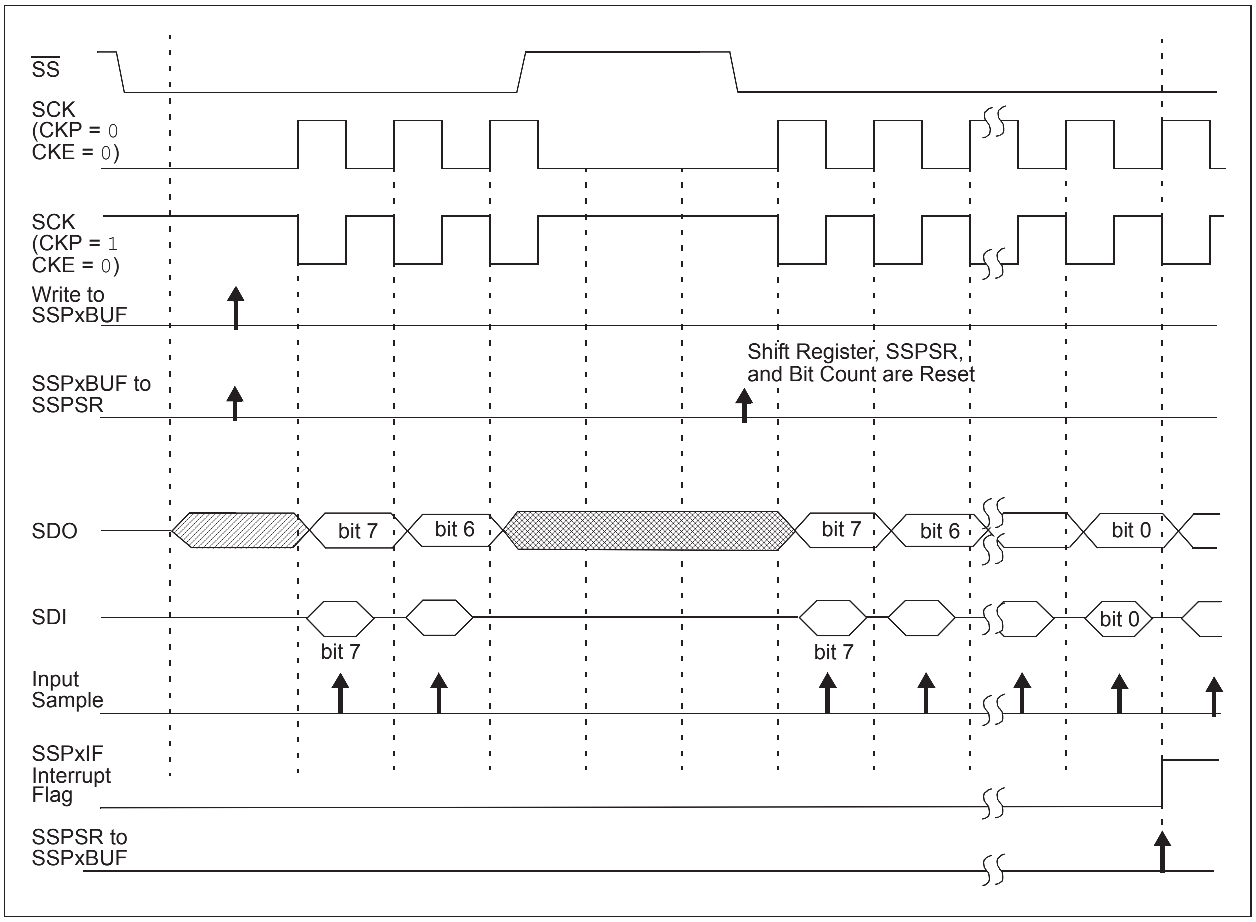
bit 7-4	**Unimplemented:** Read as ‘0’
bit 3-0	**CH<3:0>** Modulator Data High Carrier Selection bits(1) See Table 31-4.

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

TABLE 31-4: HIGH CARRIER SOURCES

CH<3:0>	High Carrier Source PIC16(L)F1764/5	High Carrier Source PIC16(L)F1768/9
1111	LC3_out	LC3_out
1110	LC2_out	LC2_out
1101	LC1_out	LC1_out
1100	Fixed Low	PWM6_out
1011	PWM5_out	PWM5_out
1010	Fixed Low	PWM4_out
1001	PWM3_out	PWM3_out
1000	Fixed Low	CCP2_out
0111	CCP1_out	CCP1_out
0110	Fixed Low	Fixed Low
0101	Fixed Low	Fixed Low
0100	Fixed Low	Fixed Low
0011	Fixed Low	Fixed Low
0010	HFINTOSC	HFINTOSC
0001	FOSC	FOSC
0000	MDxCHPPS Pin Selection	MDxCHPPS Pin Selection

FIGURE 32-8: SLAVE SELECT SYNCHRONOUS WAVEFORM



32.5.6 CLOCK STRETCHING

Clock stretching occurs when a device on the bus holds the SCL line low, effectively pausing communication. The slave may stretch the clock to allow more time to handle data or prepare a response for the master device. A master device is not concerned with stretching, as anytime it is active on the bus and not transferring data, it is stretching. Any stretching done by a slave is invisible to the master software and handled by the hardware that generates SCL.

The CKP bit of the SSPxCON1 register is used to control stretching in software. Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. Setting CKP will release SCL and allow more communication.

32.5.6.1 Normal Clock Stretching

Following an $\overline{\text{ACK}}$, if the R/$\overline{\text{W}}$ bit of SSPxSTAT is set and there is a read request, the slave hardware will clear CKP. This allows the slave time to update SSPxBUF with data to transfer to the master. If the SEN bit of SSPxCON2 is set, the slave hardware will always stretch the clock after the $\overline{\text{ACK}}$ sequence. Once the slave is ready, CKP is set by software and communication resumes.

Note 1: The BF bit has no effect on if the clock will be stretched or not. This is different than previous versions of the module that would not stretch the clock, and cleared CKP if SSPxBUF was read before the 9th falling edge of SCL.

2: Previous versions of the module did not stretch the clock for a transmission if SSPxBUF was loaded before the 9th falling edge of SCL; it is now always cleared for read requests.

32.5.6.2 10-Bit Addressing Mode

In 10-Bit Addressing mode when the UA bit is set, the clock is always stretched. This is the only time the SCL is stretched without CKP being cleared. SCL is released immediately after a write to SSPxADD.

Note: Previous versions of the module did not stretch the clock if the second address byte did not match.

32.5.6.3 Byte NACKing

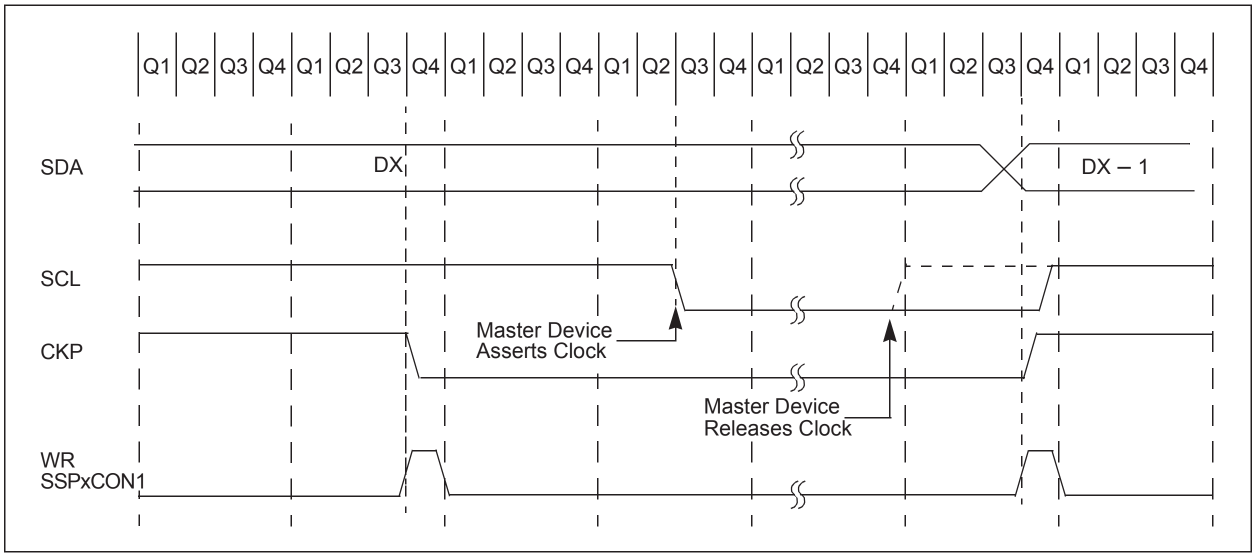
When the AHEN bit of SSPxCON3 is set, CKP is cleared by hardware after the eighth falling edge of SCL for a received matching address byte. When the DHEN bit of SSPxCON3 is set, CKP is cleared after the eighth falling edge of SCL for received data.

Stretching after the eighth falling edge of SCL allows the slave to look at the received address or data and decide if it wants to ACK the received data.

32.5.6.4 Clock Synchronization and the CKP Bit

Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. However, clearing the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I^{2}C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I^{2}C bus have released SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 32-23).

FIGURE 32-23: CLOCK SYNCHRONIZATION TIMING


Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
SDA
DX
DX – 1
SCL
CKP
Master Device Asserts Clock
Master Device Releases Clock
WR SSPxCON1

33.1.1.5 TSR Status

The TRMT bit of the TXxSTA register indicates the status of the TSR register. This is a read-only bit. The TRMT bit is set when the TSR register is empty and is cleared when a character is transferred to the TSR register from the TXxREG. The TRMT bit remains clear until all bits have been shifted out of the TSR register. No interrupt logic is tied to this bit, so the user has to poll this bit to determine the TSR status.

Note:	The TSR register is not mapped in data memory, so it is not available to the user.

33.1.1.6 Transmitting 9-Bit Characters

The EUSART supports 9-bit character transmissions. When the TX9 bit of the TXxSTA register is set, the EUSART will shift nine bits out for each character transmitted. The TX9D bit of the TXxSTA register is the ninth, and Most Significant data bit. When transmitting 9-bit data, the TX9D data bit must be written before writing the eight Least Significant bits into the TXxREG. All nine bits of data will be transferred to the TSR shift register immediately after the TXxREG is written.

A special 9-Bit Address mode is available for use with multiple receivers. See **Section 33.1.2.7 “Address Detection”** for more information on the Address mode.

33.1.1.7 Asynchronous Transmission Setup

1. Initialize the SPxBRGH, SPxBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 33.4 “EUSART Baud Rate Generator (BRG)”**).
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If 9-bit transmission is desired, set the TX9 control bit. A set ninth data bit will indicate that the eight Least Significant data bits are an address when the receiver is set for address detection.
4. Set SCKP bit if inverted transmit is desired.
5. Enable the transmission by setting the TXEN control bit. This will cause the TXIF interrupt bit to be set.
6. If interrupts are desired, set the TXIE interrupt enable bit of the PIE1 register. An interrupt will occur immediately provided that the GIE and PEIE bits of the INTCON register are also set.
7. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D data bit.
8. Load 8-bit data into the TXxREG register. This will start the transmission.

FIGURE 33-3: ASYNCHRONOUS TRANSMISSION

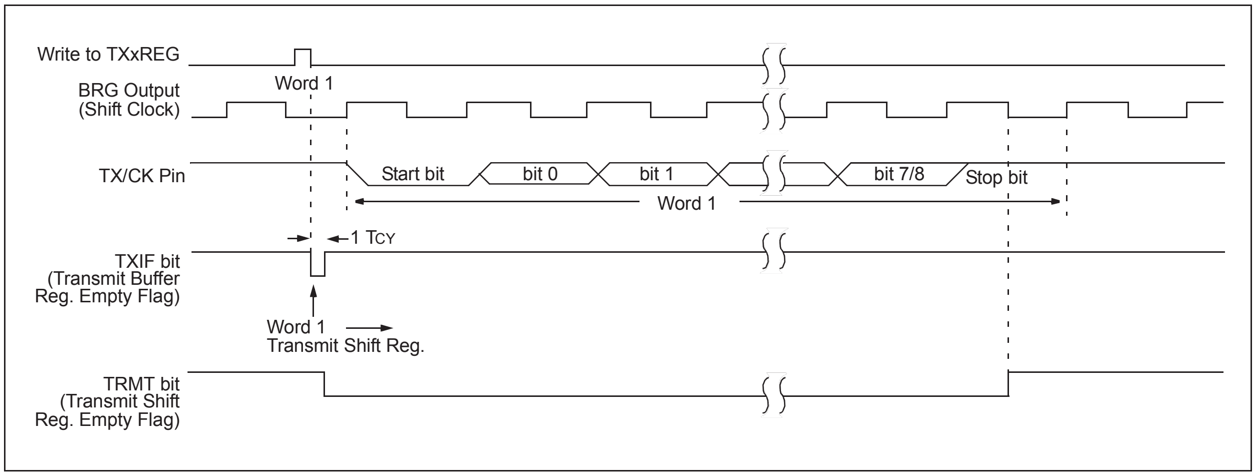


TABLE 36-9: PLL CLOCK TIMING SPECIFICATIONS

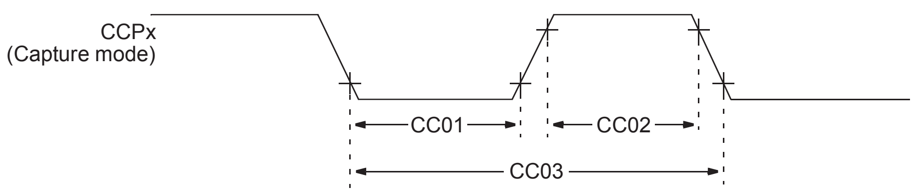
Standard Operating Conditions (unless otherwise stated)

Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
F10	FOSC	Oscillator Frequency Range	4	—	8	MHz	
F11	FSYS	On-Chip VCO System Frequency	16	—	32	MHz	
F12	TRC	PLL Start-up Time (Lock Time)	—	—	2	ms	
F13*	ΔCLK	CLKOUT Stability (Jitter)	-0.25%	—	+0.25%	%	

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5V, +25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 36-11: CAPTURE/COMPARE/PWM TIMINGS (CCP)



Note: Refer to Figure 36-4 for load conditions.

TABLE 36-13: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP)

Standard Operating Conditions (unless otherwise stated)								
Param No.	**Sym.**	**Characteristic**		**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
CC01*	TccL	CCPx Input Low Time	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	20	—	—	ns	
CC02*	TccH	CCPx Input High Time	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	20	—	—	ns	
CC03*	TccP	CCPx Input Period		$\frac{3\,T_{CY} + 40}{N}$	—	—	ns	N = prescale value

* 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.

38.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

38.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

38.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

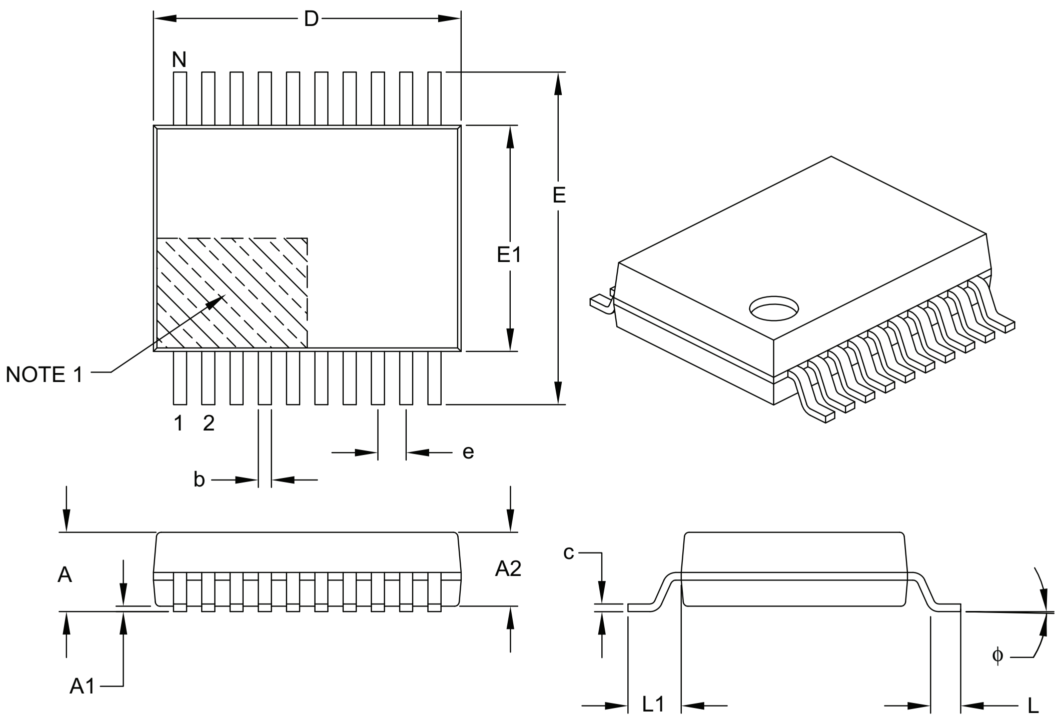
38.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

20-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units	MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		20	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	6.90	7.20	7.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	ϕ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B