PIC10(L)F320/322

6/8-Pin Flash-Based, 8-Bit Microcontrollers

High-Performance RISC CPU:

- Only 35 Instructions to Learn:
  - All single-cycle instructions, except branches
- Operating Speed:
  - DC – 16 MHz clock input
  - DC – 250 ns instruction cycle
- Up to 512 Words of Flash Program Memory
- 64 Bytes Data Memory
- Eight-level Deep Hardware Stack
- Interrupt Capability
- Processor Self-Write/Read access to Program Memory
- Pinout Compatible to other 6-Pin PIC10FXXX Microcontrollers

Special Microcontroller Features:

- Low-Power 16 MHz Internal Oscillator:
  - Software selectable frequency range from 16 MHz to 31 kHz
  - Factory calibrated to ± 1%, typical
- Wide Operating Range:
  - 1.8V to 3.6V (PIC10LF320/322)
  - 2.3V to 5.5V (PIC10F320/322)
- Power-On Reset (POR)
- Power-up Timer (PWRT)
- Brown-Out Reset (BOR)
- Ultra Low-Power Sleep Regulator
- Extended Watchdog Timer (WDT)
- Programmable Code Protection
- Power-Saving Sleep mode
- Selectable Oscillator options (EC mode or Internal Oscillator)
- In-Circuit Serial Programming™ (ICSP™) (via Two Pins)
- In-Circuit Debugger Support
- Fixed Voltage Reference (FVR) with 1.024V, 2.048V and 4.096V (‘F’ variant only) Output Levels
- Integrated Temperature Indicator
- 40-year Flash Data Retention

Low-Power Features (PIC10LF320/322):

- Standby Current:
  - 20 nA @ 1.8V, typical
- Operating Current:
  - 25 μA @ 1 MHz, 1.8V, typical
- Watchdog Timer Current:
  - 500 nA @ 1.8V, typical

Peripheral Features:

- Four I/O Pins:
  - One input-only pin
  - High current sink/source for LED drivers
  - Individually selectable weak pull-ups
  - Interrupt-on-Change
- Timer0: 8-Bit Timer/Counter with 8-Bit Programmable Prescaler
- Timer2: 8-Bit Timer/Counter with 8-Bit Period Register, Prescaler and Postscaler
- Two PWM modules:
  - 10-bit PWM, max. frequency 16 kHz
  - Combined to single 2-phase output
- A/D Converter:
  - 8-bit resolution with 3 channels
- Configurable Logic Cell (CLC):
  - 8 selectable input source signals
  - Two inputs per module
  - Software selectable logic functions including: AND/OR/XOR/D Flop/D Latch/SR/JK
  - External or internal inputs/outputs
  - Operation while in Sleep
- Numerically Controlled Oscillator (NCO):
  - 20-bit accumulator
  - 16-bit increment
  - Linear frequency control
  - High-speed clock input
  - Selectable Output modes
    - Fixed Duty Cycle (FDC)
    - Pulse Frequency (PF) mode
- Complementary Waveform Generator (CWG):
  - Selectable falling and rising edge dead-band control
  - Polarity control
  - Two auto-shutdown sources
  - Multiple input sources: PWM, CLC, NCO

PIC10(L)F320/322 Family Types

Device	Data Sheet Index	Program Memory Flash (words)	Data SRAM (bytes)	I/O's(2)	8-Bit ADC (ch)	Timers (8-Bit)	PWM	Complementary Wave Generator (CWG)	Configurable Logic Cell (CLC)	Fixed Voltage Reference (FVR)	Numerically Controlled Oscillator (NCO)	Debug(1)	XLP
PIC10(L)F320	(1)	256	64	4	3	2	2	1	1	1	1	H	N
PIC10(L)F322	(1)	512	64	4	3	2	2	1	1	1	1	H	N

Note 1: I - Debugging, Integrated on Chip; H - Debugging, Available using Debug Header; E - Emulation, Available using Emulation Header.

2: One pin is input-only.

Data Sheet Index:

1: DS40001585 PIC10(L)F320/322 Data Sheet, 6/8 Pin High Performance, Flash Microcontrollers.

Note: For other small form-factor package availability and marking information, please visit **http://www.microchip.com/packaging** or contact your local sales office.

FIGURE 1: 6-PIN DIAGRAM, PIC10(L)F320/322

SOT-23

FIGURE 2: 8-PIN DIAGRAM, PIC10(L)F320/322

PDIP, DFN

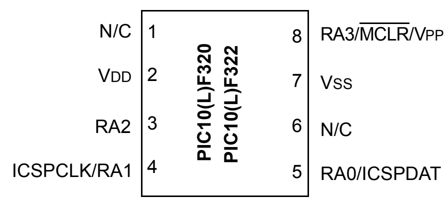


TABLE 1: 6 AND 8-PIN ALLOCATION TABLE, PIC10(L)F320/322

I/O	6-Pin	8-Pin	Analog	Timer	PWM	Interrupts	Pull-ups	CWG	NCO	CLC	Basic	ICSP
RA0	1	5	AN0	—	PWM1	IOC0	Y	CWG1A	—	CLC1IN0	—	ICSPDAT
RA1	3	4	AN1	—	PWM2	IOC1	Y	CWG1B	NCO1CLK	CLC1	CLKIN	ICSPCLK
RA2	4	3	AN2	T0CKI	—	INT/IOC2	Y	$\overline{CWG1FLT}$	NCO1	CLC1IN1	CLKR	
RA3	6	8	—	—	—	IOC3	Y	—	—	—	$\overline{MCLR}$	V_{PP}
N/C	—	1	—	—	—	—	—	—	—	—	—	—
N/C	—	6	—	—	—	—	—	—	—	—	—	—
V_{DD}	5	2	—	—	—	—	—	—	—	—	V_{DD}	—
V_{SS}	2	7	—	—	—	—	—	—	—	—	V_{SS}	—

Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; **http://www.microchip.com**
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

1.0 DEVICE OVERVIEW

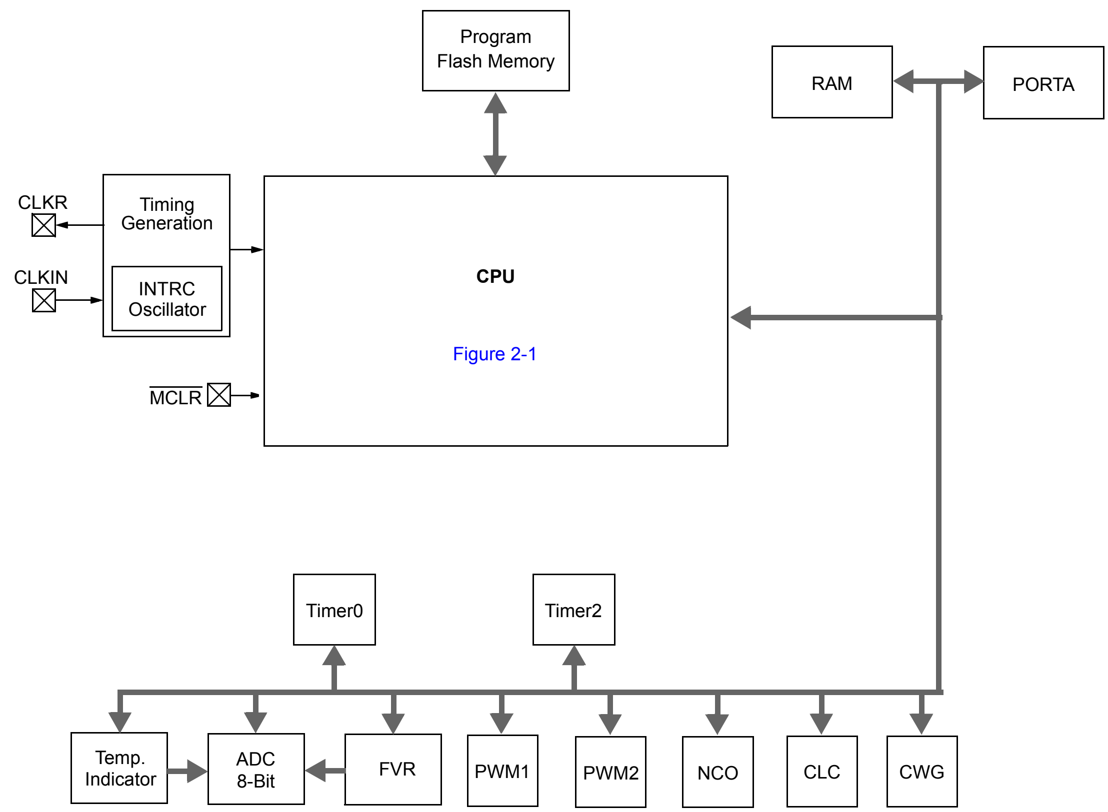
The PIC10(L)F320/322 are described within this data sheet. They are available in 6/8-pin packages. Figure 1-1 shows a block diagram of the PIC10(L)F320/322 devices. Table 1-2 shows the pinout descriptions.

Reference Table 1-1 for peripherals available per device.

TABLE 1-1: DEVICE PERIPHERAL SUMMARY

Peripheral		PIC10(L)F320	PIC10(L)F322
Analog-to-Digital Converter (ADC)		●	●
Configurable Logic Cell (CLC)		●	●
Complementary Wave Generator (CWG)		●	●
Fixed Voltage Reference (FVR)		●	●
Numerically Controlled Oscillator (NCO)		●	●
Temperature Indicator		●	●
PWM Modules			
	PWM1	●	●
	PWM2	●	●
Timers			
	Timer0	●	●
	Timer2	●	●

FIGURE 1-1: PIC10(L)F320/322 BLOCK DIAGRAM



Note 1: See applicable chapters for more information on peripherals.

TABLE 1-2: PIC10(L)F320/322 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/PWM1/CLC1IN0/CWG1A/AN0/ICSPDAT	RA0	TTL	CMOS	General purpose I/O with IOC and WPU.
	PWM1	—	CMOS	PWM output.
	CLC1IN0	ST	—	CLC input.
	CWG1A	—	CMOS	CWG primary output.
	AN0	AN	—	A/D Channel input.
	ICSPDAT	ST	CMOS	ICSP™ Data I/O.
RA1/PWM2/CLC1/CWG1B/AN1/CLKIN/ICSPCLK/NCO1CLK	RA1	TTL	CMOS	General purpose I/O with IOC and WPU.
	PWM2	—	CMOS	PWM output.
	CLC1	—	CMOS	CLC output.
	CWG1B	—	CMOS	CWG complementary output.
	AN1	AN	—	A/D Channel input.
	CLKIN	ST	—	External Clock input (EC mode).
	ICSPCLK	ST	—	ICSP™ Programming Clock.
	NCO1CLK	ST	—	Numerical Controlled Oscillator external clock input.
RA2/INT/T0CKI/NCO1/CLC1IN1/CLKR/AN2/$\overline{\text{CWG1FLT}}$	RA2	TTL	CMOS	General purpose I/O with IOC and WPU.
	INT	ST	—	External interrupt.
	T0CKI	ST	—	Timer0 clock input.
	NCO1	—	CMOS	Numerically Controlled Oscillator output.
	CLC1IN1	ST	—	CLC input.
	CLKR	—	CMOS	Clock Reference output.
	AN2	AN	—	A/D Channel input.
	$\overline{\text{CWG1FLT}}$	ST	—	Complementary Waveform Generator Fault 1 source input.
RA3/$\overline{\text{MCLR}}$/VPP	RA3	TTL	—	General purpose input.
	$\overline{\text{MCLR}}$	ST	—	Master Clear with internal pull-up.
	VPP	HV	—	Programming voltage.
VDD	VDD	Power	—	Positive supply.
VSS	VSS	Power	—	Ground reference.

Legend: AN = Analog input or output; CMOS = CMOS compatible input or output; TTL = CMOS input with TTL levels; ST = CMOS input with Schmitt Trigger levels; HV = High Voltage

2.0 MEMORY ORGANIZATION

These devices contain the following types of memory:

- Program Memory
  - Configuration Word
  - Device ID
  - User ID
  - Flash Program Memory
- Data Memory
  - Core Registers
  - Special Function Registers
  - General Purpose RAM
  - Common RAM

The following features are associated with access and control of program memory and data memory:

- PCL and PCLATH
- Stack
- Indirect Addressing

2.1 Program Memory Organization

The mid-range core has a 13-bit program counter capable of addressing 8K x 14 program memory space. This device family only implements up to 512 words of the 8K program memory space. Table 2-1 shows the memory sizes implemented for the PIC10(L)F320/322 family. Accessing a location above these boundaries will cause a wrap-around within the implemented memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figures 2-1, and 2-2).

TABLE 2-1: DEVICE SIZES AND ADDRESSES

Device	Program Memory Space (Words)	Last Program Memory Address	High-Endurance Flash Memory Address Range (1)
PIC10(L)F320	256	00FFh	0F80h-00FFh
PIC10(L)F322	512	01FFh	1F80h-01FFh

Note 1: High-endurance Flash applies to low byte of each address in the range.

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR PIC10(L)F320

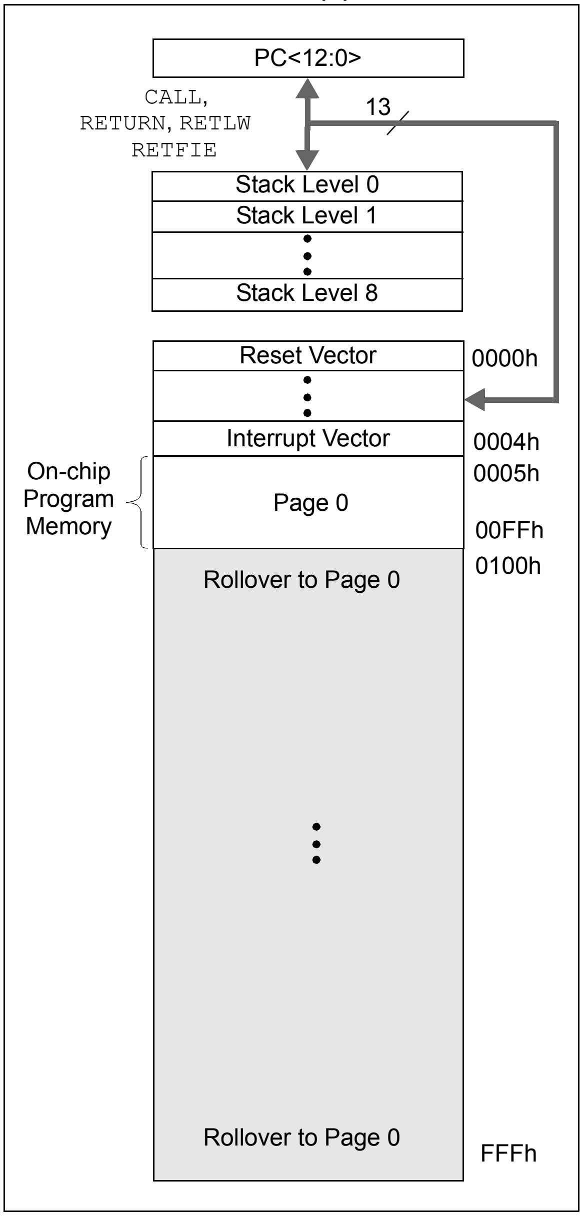
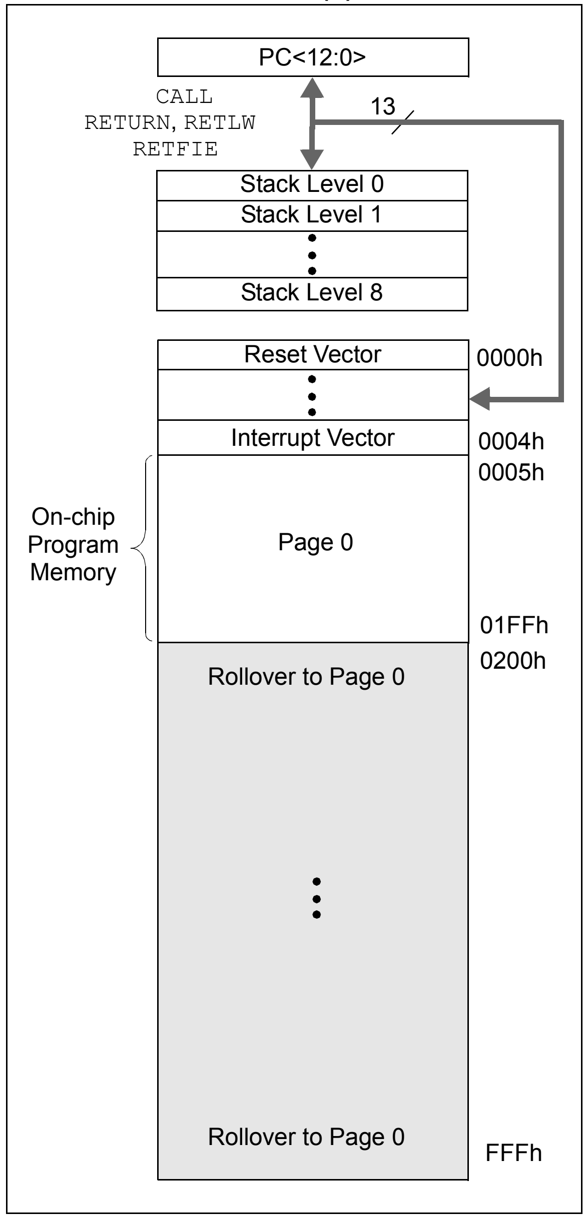


FIGURE 2-2: PROGRAM MEMORY MAP AND STACK FOR PIC10(L)F322



2.2 Data Memory Organization

The data memory is in one bank, which contains the General Purpose Registers (GPR) and the Special Function Registers (SFR). The RP<1:0> bits of the STATUS register are the bank select bits.

RP1 RP0

0 0 → Bank 0 is selected

The bank extends up to 7Fh (128 bytes). The lower locations of the bank are reserved for the Special Function Registers. Above the Special Function Registers are the General Purpose Registers, implemented as Static RAM.

2.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 64 x 8 in the PIC10(L)F320/322. Each register is accessed, either directly or indirectly, through the File Select Register (FSR) (see **Section 2.4 "Indirect Addressing, INDF and FSR Registers"**).

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (see Table 2-3). These registers are static RAM.

The special registers can be classified into two sets: core and peripheral. The Special Function Registers associated with the "core" are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

2.2.2.1 STATUS Register

The STATUS register, shown in Register 2-1, contains:

- the arithmetic status of the ALU
- the Reset status
- the bank select bits for data memory (SRAM)

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as `'000u u1uu'` (where `u` = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits (see **Section 23.0 "Instruction Set Summary"**).

Note 1:	Bits IRP and RP1 of the STATUS register are not used by the PIC10(L)F320 and should be maintained as clear. Use of these bits is not recommended, since this may affect upward compatibility with future products.
2:	The C and DC bits operate as a $\overline{\text{Borrow}}$ and $\overline{\text{Digit Borrow}}$ out bit, respectively, in subtraction.

REGISTER 2-1: STATUS: STATUS REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R-1/q	R-1/q	R/W-x/u	R/W-x/u	R/W-x/u
IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
bit 7							bit 0

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

bit 7 **IRP:** Reserved(2)

bit 6-5 **RP<1:0>:** Reserved(2)

bit 4 **$\overline{\text{TO}}$:** Time-out bit
1 = After power-up, `CLRWDT` instruction or `SLEEP` instruction
0 = A WDT time-out occurred

bit 3 **$\overline{\text{PD}}$:** Power-Down bit
1 = After power-up or by the `CLRWDT` instruction
0 = By execution of the `SLEEP` instruction

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)
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. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low-order bit of the source register.

2: Maintain as ‘0’.

2.2.3 DEVICE MEMORY MAPS

The memory maps for PIC10(L)F320/322 are as shown in Table 2-2.

TABLE 2-2: PIC10(L)F320/322 MEMORY MAP (BANK 0)

Register	Address	Register	Address	Register	Address	Register	Address
INDF(*)	00h	PMADRL	20h	General Purpose Registers 32 Bytes	40h–5Fh	General Purpose Registers 32 Bytes	60h–7Fh
TMR0	01h	PMADRH	21h				
PCL	02h	PMDATL	22h				
STATUS	03h	PMDATH	23h				
FSR	04h	PMCON1	24h				
PORTA	05h	PMCON2	25h				
TRISA	06h	CLKRCON	26h				
LATA	07h	NCO1ACCL	27h				
ANSELA	08h	NCO1ACCH	28h				
WPUA	09h	NCO1ACCU	29h				
PCLATH	0Ah	NCO1INCL	2Ah				
INTCON	0Bh	NCO1INCH	2Bh				
PIR1	0Ch	Reserved	2Ch				
PIE1	0Dh	NCO1CON	2Dh				
OPTION_REG	0Eh	NCO1CLK	2Eh				
PCON	0Fh	Reserved	2Fh				
OSCCON	10h	WDTCON	30h				
TMR2	11h	CLC1CON	31h				
PR2	12h	CLC1SEL1	32h				
T2CON	13h	CLC1SEL2	33h				
PWM1DCL	14h	CLC1POL	34h				
PWM1DCH	15h	CLC1GLS0	35h				
PWM1CON	16h	CLC1GLS1	36h				
PWM2DCL	17h	CLC1GLS2	37h				
PWM2DCH	18h	CLC1GLS3	38h				
PWM2CON	19h	CWG1CON0	39h				
IOCAP	1Ah	CWG1CON1	3Ah				
IOCAN	1Bh	CWG1CON2	3Bh				
IOCAF	1Ch	CWG1DBR	3Ch				
FVRCON	1Dh	CWG1DBF	3Dh				
ADRES	1Eh	VREGCON	3Eh				
ADCON	1Fh	BORCON	3Fh				

Legend: = Unimplemented data memory locations, read as ‘0’.
* = Not a physical register.

TABLE 2-3: SPECIAL FUNCTION REGISTER SUMMARY (BANK 0)

Address	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 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
01h	TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu
02h	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
03h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	—	—	RA3	RA2	RA1	RA0	---- xxxx	---- uuuu
06h	TRISA	—	—	—	—	—(1)	TRISA2	TRISA1	TRISA0	---- 1111	---- 1111
07h	LATA	—	—	—	—	—	LATA2	LATA1	LATA0	---- -xxx	---- -uuu
08h	ANSELA	—	—	—	—	—	ANSA2	ANSA1	ANSA0	---- -111	---- -111
09h	WPUA	—	—	—	—	WPUA3	WPUA2	WPUA1	WPUA0	---- 1111	---- 1111
0Ah	PCLATH	—	—	—	—	—	—	—	PCLH0	---- ---0	---- ---0
0Bh	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 000u
0Ch	PIR1	—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—	-0-0 0-0-	-0-0 0-0-
0Dh	PIE1	—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—	-0-0 0-0-	-0-0 0-0-
0Eh	OPTION_REG	$\overline{WPUEN}$	INTEDG	T0CS	T0SE	PSA	PS<2:0>			1111 1111	uuuu uuuu
0Fh	PCON	—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$	---- --qq	---- --uu
10h	OSCCON	—	IRCF<2:0>			HFIOFR	—	LFIOFR	HFIOFS	-110 0-00	-110 0-00
11h	TMR2	Timer2 Module Register								0000 0000	0000 0000
12h	PR2	Timer2 Period Register								1111 1111	1111 1111
13h	T2CON	—	TOUTPS<3:0>				TMR2ON	T2CKPS<1:0>		-000 0000	-000 0000
14h	PWM1DCL	PWM1DCL<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
15h	PWM1DCH	PWM1DCH<7:0>								xxxx xxxx	uuuu uuuu
16h	PWM1CON	PWM1EN	PWM1OE	PWM1OUT	PWM1POL	—	—	—	—	0000 ----	0000 ----
17h	PWM2DCL	PWM2DCL<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
18h	PWM2DCH	PWM2DCH<7:0>								xxxx xxxx	uuuu uuuu
19h	PWM2CON	PWM2EN	PWM2OE	PWM2OUT	PWM2POL	—	—	—	—	0000 ----	0000 ----
1Ah	IOCAP	—	—	—	—	IOCAP3	IOCAP2	IOCAP1	IOCAP0	---- 0000	---- 0000
1Bh	IOCAN	—	—	—	—	IOCAN3	IOCAN2	IOCAN1	IOCAN0	---- 0000	---- 0000
1Ch	IOCAF	—	—	—	—	IOCAF3	IOCAF2	IOCAF1	IOCAF0	---- 0000	---- 0000
1Dh	FVRCON	FVREN	FVRRDY	TSEN	TSRNG	—	—	ADFVR<1:0>		0x00 --00	0x00 --00
1Eh	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON	ADCS<2:0>			CHS<2:0>			GO/$\overline{DONE}$	ADON	0000 0000	0000 0000

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'.

TABLE 2-3: SPECIAL FUNCTION REGISTER SUMMARY (BANK 0) (CONTINUED)

Address	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 0 (Continued)											
20h	PMADRL	PMADR<7:0>								0000 0000	0000 0000
21h	PMADRH	—	—	—	—	—	—	—	PMADR8	---- ---0	---- ---0
22h	PMDATL	PMDAT<7:0>								xxxx xxxx	uuuu uuuu
23h	PMDATH	—	—	PMDAT<13:8>						--xx xxxx	--uu uuuu
24h	PMCON1	—	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	1000 0000	1000 q000
25h	PMCON2	Program Memory Control Register 2 (not a physical register)								0000 0000	0000 0000
26h	CLKRCON	—	CLKROE	—	—	—	—	—	—	-0-- ----	-0-- ----
27h	NCO1ACCL	NCO1 Accumulator <7:0>								0000 0000	0000 0000
28h	NCO1ACCH	NCO1 Accumulator <15:8>								0000 0000	0000 0000
29h	NCO1ACCU	—				NCO1 Accumulator <19..16>				---- 0000	---- 0000
2Ah	NCO1INCL	NCO1 Increment <7:0>								0000 0001	0000 0001
2Bh	NCO1INCH	NCO1 Increment <15:8>								0000 0000	0000 0000
2Ch	—	Unimplemented								—	—
2Dh	NCO1CON	N1EN	N1OE	N1OUT	N1POL	—	—	—	N1PFM	0000 ---0	00x0 ---0
2Eh	NCO1CLK	N1PWS<2:0>			—	—	—	N1CKS<1:0>		000- --00	000- --00
2Fh	Reserved	Reserved								xxxx xxxx	uuuu uuuu
30h	WDTCON	—	—	WDTPS<4:0>					SWDTEN	--01 0110	--01 0110
31h	CLC1CON	LC1EN	LC1OE	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>			00x0 -000	00x0 -000
32h	CLC1SEL0	—	LC1D2S<2:0>			—	LC1D1S<2:0>			-xxx -xxx	-uuu -uuu
33h	CLC1SEL1	—	LC1D4S<2:0>			—	LC1D3S<2:0>			-xxx -xxx	-uuu -uuu
34h	CLC1POL	LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	0--- xxxx	0--- uuuu
35h	CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	xxxx xxxx	uuuu uuuu
36h	CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	xxxx xxxx	uuuu uuuu
37h	CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	xxxx xxxx	uuuu uuuu
38h	CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	xxxx xxxx	uuuu uuuu
39h	CWG1CON0	G1EN	G1OEB	G1OEA	G1POLB	G1POLA	—	—	G1CS0	0000 0--0	0000 0--0
3Ah	CWG1CON1	G1ASDLB<1:0>		G1ASDLA<1:0>		—	—	G1IS<1:0>		xxxx --xx	uuuu --uu
3Bh	CWG1CON2	G1ASE	G1ARSEN	—	—	—	—	G1ASDCLC1	G1ASDFLT	xx-- --xx	uu-- --uu
3Ch	CWG1DBR	—	—	CWG1DBR<5:0>						--xx xxxx	--uu uuuu
3Dh	CWG1DBF	—	—	CWG1DBF<5:0>						--xx xxxx	--uu uuuu
3Eh	VREGCON	—	—	—	—	—	—	VREGPM1	Reserved	---- --01	---- --01
3Fh	BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	10-- ---q	uu-- ---u

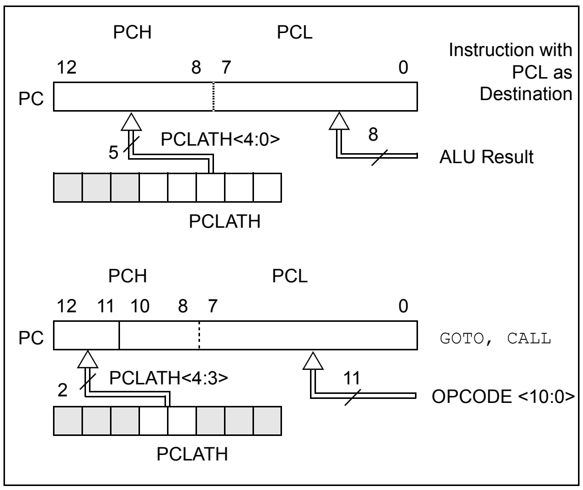
Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: Unimplemented, read as '1'.

2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-3 shows the two situations for the loading of the PC. The upper example in Figure 2-3 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 2-3 shows how the PC is loaded during a `CALL` or `GOTO` instruction (PCLATH<4:3> → PCH).

FIGURE 2-3: LOADING OF PC IN DIFFERENT SITUATIONS



2.3.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<12:8> bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the program counter to be changed by writing the desired upper 5 bits to the PCLATH register. When the lower 8 bits are written to the PCL register, all 13 bits of the program counter will change to the values contained in the PCLATH register and those being written to the PCL register.

A computed `GOTO` is accomplished by adding an offset to the program counter (`ADDWF PCL`). Care should be exercised when jumping into a look-up table or program branch table (computed `GOTO`) by modifying the PCL register. Assuming that PCLATH is set to the table start address, if the table length is greater than 255 instructions or if the lower 8 bits of the memory address rolls over from 0xFF to 0x00 in the middle of the table, then PCLATH must be incremented for each address rollover that occurs between the table beginning and the target location within the table.

For more information refer to Application Note AN556, "*Implementing a Table Read*" (DS00556).

2.3.2 STACK

All devices have an 8-level x 13-bit wide hardware stack (see Figure 2-1). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a `CALL` instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a `RETURN`, `RETLW` or a `RETFIE` instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

> **Note 1:** There are no Status bits to indicate Stack Overflow or Stack Underflow conditions.
>
> **2:** There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the `CALL`, `RETURN`, `RETLW` and `RETFIE` instructions or the vectoring to an interrupt address.

2.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

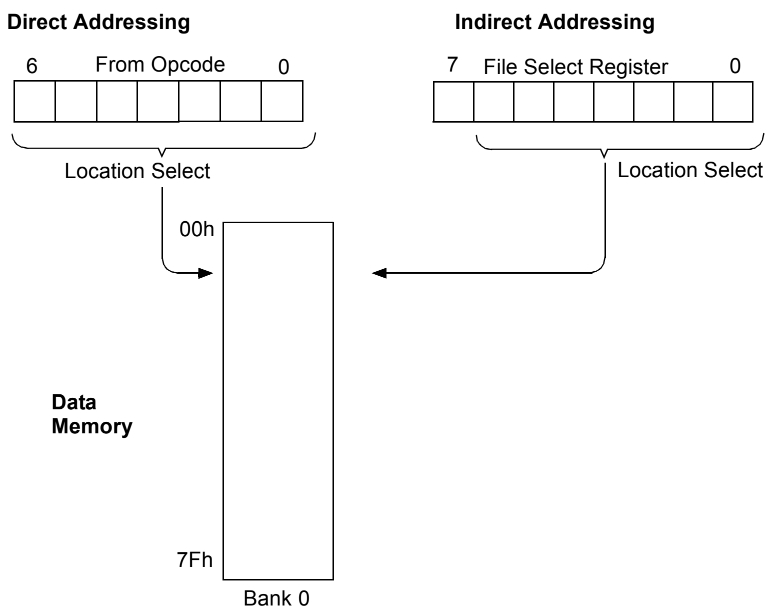
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR and the IRP bit of the STATUS register, as shown in Figure 2-4.

A simple program to clear RAM location 40h-7Fh using indirect addressing is shown in Example 2-1.

EXAMPLE 2-1: INDIRECT ADDRESSING

```
        MOVLW   0x40    ;initialize pointer
        MOVWF   FSR     ;to RAM
NEXT    CLRF    INDF    ;clear INDF register
        INCF    FSR     ;inc pointer
        BTFSS   FSR,7   ;all done?
        GOTO    NEXT    ;no clear next
CONTINUE                ;yes continue
```


FIGURE 2-4: DIRECT/INDIRECT ADDRESSING PIC10(L)F320/322



For memory map detail, see Figure 2-2.

3.0 DEVICE CONFIGURATION

Device configuration consists of Configuration Word and Device ID.

3.1 Configuration Word

There are several Configuration Word bits that allow different oscillator and memory protection options. These are implemented as Configuration Word at 2007h.

3.2 Register Definitions: Configuration Word

REGISTER 3-1: CONFIG: CONFIGURATION WORD

U-1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1
—	WRT<1:0>		BORV	LPBOR	LVP
bit 13					bit 8

R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1
$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		BOREN<1:0>		FOSC
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	P = Programmable bit

bit 13 **Unimplemented**: Read as ‘1’

bit 12-11 **WRT<1:0>:** Flash Memory Self-Write Protection bits
256 W Flash memory: PIC10(L)F320:
- 11 =Write protection off
- 10 =000h to 03Fh write-protected, 040h to 0FFh may be modified by PMCON control
- 01 =000h to 07Fh write-protected, 080h to 0FFh may be modified by PMCON control
- 00 =000h to 0FFh write-protected, no addresses may be modified by PMCON control

512 W Flash memory: PIC10(L)F322:
- 11 =Write protection off
- 10 =000h to 07Fh write-protected, 080h to 1FFh may be modified by PMCON control
- 01 =000h to 0FFh write-protected, 100h to 1FFh may be modified by PMCON control
- 00 =000h to 1FFh write-protected, no addresses may be modified by PMCON control

bit 10 **BORV:** Brown-out Reset Voltage Selection bit
1 = Brown-out Reset voltage (VBOR), low trip point selected.
0 = Brown-out Reset voltage (VBOR), high trip point selected.

bit 9 **LPBOR:** Low-Power Brown-out Reset Enable bit
1 = Low-power Brown-out Reset is enabled
0 = Low-power Brown-out Reset is disabled

bit 8 **LVP:** Low-Voltage Programming Enable bit
1 = Low-Voltage Programming enabled. $\overline{\text{MCLR}}$/VPP pin function is $\overline{\text{MCLR}}$.
0 = High Voltage on $\overline{\text{MCLR}}$/VPP must be used for programming

bit 7 **$\overline{\text{CP}}$:** Code Protection bit(2)
1 = Program memory code protection is disabled
0 = Program memory code protection is enabled

bit 6 **MCLRE:** $\overline{\text{MCLR}}$/VPP Pin Function Select bit
If LVP bit = 1:
This bit is ignored.
If LVP bit = 0:
1 = $\overline{\text{MCLR}}$/VPP pin function is $\overline{\text{MCLR}}$; Weak pull-up enabled.
0 = $\overline{\text{MCLR}}$/VPP pin function is digital input; $\overline{\text{MCLR}}$ internally disabled; Weak pull-up under control of WPUA3 bit.

Note 1: Enabling Brown-out Reset does not automatically enable Power-up Timer.
2: Once enabled, code-protect can only be disabled by bulk erasing the device.
3: See VBOR parameter for specific trip point voltages.

REGISTER 3-1: CONFIG: CONFIGURATION WORD (CONTINUED)

bit 5	**$\overline{\text{PWRTE}}$:** Power-up Timer Enable bit(1) 1 = PWRT disabled 0 = PWRT enabled	
bit 4-3	**WDTE<1:0>:** Watchdog Timer Enable bit 11 = WDT enabled 10 = WDT enabled while running and disabled in Sleep 01 = WDT controlled by the SWDTEN bit in the WDTCON register 00 = WDT disabled	
bit 2-1	**BOREN<1:0>:** Brown-out Reset Enable bits 11 = Brown-out Reset enabled; SBOREN bit is ignored 10 = Brown-out Reset enabled while running, disabled in Sleep; SBOREN bit is ignored 01 = Brown-out Reset controlled by the SBOREN bit in the BORCON register 00 = Brown-out Reset disabled; SBOREN bit is ignored	
bit 0	**FOSC:** Oscillator Selection bit 1 = EC on CLKIN pin 0 = INTOSC oscillator I/O function available on CLKIN pin	

Note 1: Enabling Brown-out Reset does not automatically enable Power-up Timer.
2: Once enabled, code-protect can only be disabled by bulk erasing the device.
3: See V_{BOR} parameter for specific trip point voltages.

3.3 Code Protection

Code protection allows the device to be protected from unauthorized access. Program memory protection and data memory protection are controlled independently. Internal access to the program memory and data memory are unaffected by any code protection setting.

3.3.1 PROGRAM MEMORY PROTECTION

The entire program memory space is protected from external reads and writes by the $\overline{CP}$ bit in Configuration Word. When $\overline{CP}$ = `0`, external reads and writes of program memory are inhibited and a read will return all '`0`'s. The CPU can continue to read program memory, regardless of the protection bit settings. Writing the program memory is dependent upon the write protection setting. See **Section 3.4 "Write Protection"** for more information.

3.4 Write Protection

Write protection allows the device to be protected from unintended self-writes. Applications, such as boot loader software, can be protected while allowing other regions of the program memory to be modified.

The WRT<1:0> bits in Configuration Word define the size of the program memory block that is protected.

3.5 User ID

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are readable and writable during normal execution. See **Section 3.6 "Device ID and Revision ID"** for more information on accessing these memory locations. For more information on checksum calculation, see the "PIC10(L)F320/322 Flash Memory Programming Specification" (DS41572).

3.6 Device ID and Revision ID

The memory location 2006h is where the Device ID and Revision ID are stored. The upper nine bits hold the Device ID. The lower five bits hold the Revision ID. See **Section 9.4 “User ID, Device ID and Configuration Word Access”** for more information on accessing these memory locations.

Development tools, such as device programmers and debuggers, may be used to read the Device ID and Revision ID.

3.7 Register Definitions: Device and Revision

REGISTER 3-2: DEVID: DEVICE ID REGISTER(1)

R	R	R	R	R	R
DEV<8:3>					
bit 13					bit 8

R	R	R	R	R	R	R	R
DEV<2:0>			REV<4:0>				
bit 7							bit 0

Legend:	
R = Readable bit	
‘1’ = Bit is set	‘0’ = Bit is cleared

bit 13-5 **DEV<8:0>:** Device ID bits

Device	DEVID<13:0> Values	
	DEV<8:0>	**REV<4:0>**
PIC10F320	`10 1001 101`	`x xxxx`
PIC10LF320	`10 1001 111`	`x xxxx`
PIC10F322	`10 1001 100`	`x xxxx`
PIC10LF322	`10 1001 110`	`x xxxx`

bit 4-0 **REV<4:0>:** Revision ID bits

These bits are used to identify the revision.

Note 1: This location cannot be written.

4.0 OSCILLATOR MODULE

4.1 Overview

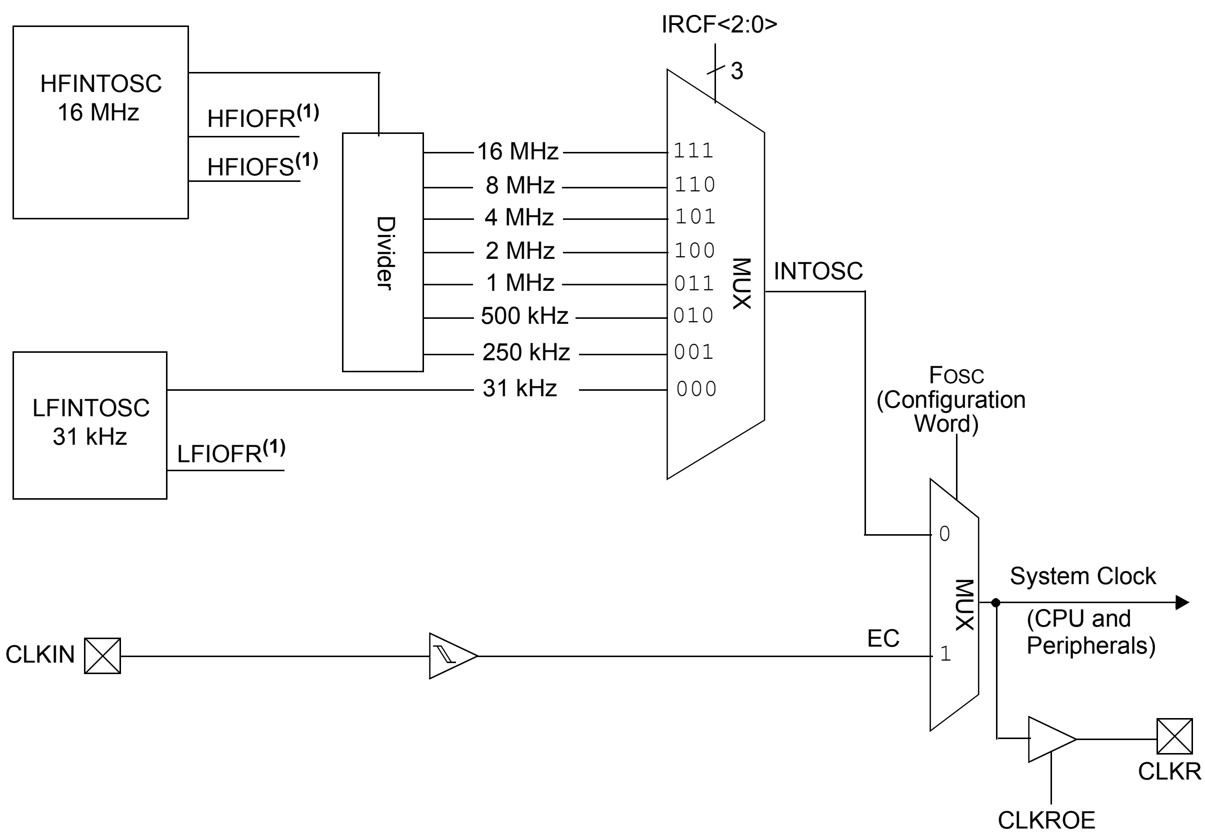
The oscillator module has a variety of clock sources and selection features that allow it to be used in a range of applications while maximizing performance and minimizing power consumption. Figure 4-1 illustrates a block diagram of the oscillator module.

The system can be configured to use an internal calibrated high-frequency oscillator as clock source, with a choice of selectable speeds via software.

Clock source modes are configured by the FOSC bit in Configuration Word (CONFIG).

1. EC oscillator from CLKIN.
2. INTOSC oscillator, CLKIN not enabled.

FIGURE 4-1: PIC10(L)F320/322 CLOCK SOURCE BLOCK DIAGRAM



Note 1: HFIOFR, HFIOFS and LFIOFR are Status bits in the OSCCON register.

4.2 Clock Source Modes

Clock source modes can be classified as external or internal.

- Internal clock source (INTOSC) is contained within the oscillator module, which has eight selectable output frequencies, with a maximum internal frequency of 16 MHz.
- The External Clock mode (EC) relies on an external signal for the clock source.

The system clock can be selected between external or internal clock sources via the FOSC bit of the Configuration Word.

4.3 Internal Clock Modes

The internal clock sources are contained within the oscillator module. The internal oscillator block has two internal oscillators that are used to generate all internal system clock sources: the 16 MHz High-Frequency Internal Oscillator (HFINTOSC) and the 31 kHz (LFINTOSC).

The HFINTOSC consists of a primary and secondary clock. The secondary clock starts first with rapid start-up time, but low accuracy. The secondary clock ready signal is indicated with the HFIOFR bit of the OSCCON register. The primary clock follows with slower start-up time and higher accuracy. The primary clock is stable when the HFIOFS bit of the OSCCON register bit goes high.

4.3.1 INTOSC MODE

When the FOSC bit of the Configuration Word is cleared, the INTOSC mode is selected. When INTOSC is selected, CLKIN pin is available for general purpose I/O. See **Section 3.0 "Device Configuration"** for more information.

4.3.2 FREQUENCY SELECT BITS (IRCF)

The output of the 16 MHz HFINTOSC is connected to a divider and multiplexer (see Figure 4-1). The Internal Oscillator Frequency Select bits (IRCF) of the OSCCON register select the frequency output of the internal oscillator:

- HFINTOSC
  - 16 MHz
  - 8 MHz (default after Reset)
  - 4 MHz
  - 2 MHz
  - 1 MHz
  - 500 kHz
  - 250 kHz
- LFINTOSC
  - 31 kHz

Note:	Following any Reset, the IRCF<2:0> bits of the OSCCON register are set to '110' and the frequency selection is set to 8 MHz. The user can modify the IRCF bits to select a different frequency.

There is no delay when switching between HFINTOSC frequencies with the IRCF bits. This is because the switch involves only a change to the frequency output divider.

Start-up delay specifications are located in **Section 24.0 "Electrical Specifications"**.

4.4 Register Definitions: Reference Clock Control

REGISTER 4-1: CLKRCON – REFERENCE CLOCK CONTROL REGISTER

U-0	R/W-0/0	U-0	U-0	U-0	U-0	U-0	U-0
—	CLKROE	—	—	—	—	—	—
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
q = Value depends on condition			

bit 7 **Unimplemented:** Read as ‘0’

bit 6 **CLKROE:** Reference Clock Output Enable bit
1 = Reference Clock output (CLKR), regardless of TRIS
0 = Reference Clock output disabled

bit 5-0 **Unimplemented:** Read as ‘0’

4.5 Register Definitions: Oscillator Control

REGISTER 4-2: OSCCON: OSCILLATOR CONTROL REGISTER

U-0	R/W-1/1	R/W-1/1	R/W-0/0	R-0/0	U-0	R-0/0	R-0/0
—	IRCF<2:0>			HFIOFR	—	LFIOFR	HFIOFS
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **IRCF<2:0>:** INTOSC (FOSC) Frequency Select bits
111 = 16 MHz
110 = 8 MHz (default value)
101 = 4 MHz
100 = 2 MHz
011 = 1 MHz
010 = 500 kHz
001 = 250 kHz
000 = 31 kHz (LFINTOSC)

bit 3 **HFIOFR:** High-Frequency Internal Oscillator Ready bit
1 = 16 MHz Internal Oscillator (HFINTOSC) is ready
0 = 16 MHz Internal Oscillator (HFINTOSC) is not ready

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **LFIOFR:** Low-Frequency Internal Oscillator Ready bit
1 = 31 kHz Internal Oscillator (LFINTOSC) is ready
0 = 31 kHz Internal Oscillator (LFINTOSC) is not ready

bit 0 **HFIOFS:** High-Frequency Internal Oscillator Stable bit
1 = 16 MHz Internal Oscillator (HFINTOSC) is stable
0 = 16 MHz Internal Oscillator (HFINTOSC) is not stable

4.6 External Clock Mode

4.6.1 EC MODE

The External Clock (EC) mode allows an externally generated logic level as the system clock source. When operating in this mode, an external clock source is connected to the CLKIN input.

TABLE 4-1: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CLKRCON	—	CLKROE	—	—	—	—	—	—	26
OSCCON	—	IRCF<2:0>			HFIOFR	—	LFIOFR	HFIOFS	26

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by ECWG.

TABLE 4-2: SUMMARY OF CONFIGURATION WORD WITH CLOCK SOURCES

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG	13:8	—	—	—	WRT<1:0>		BORV	LPBOR	LVP	20
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		BOREN<1:0>		FOSC	

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.

5.0 RESETS

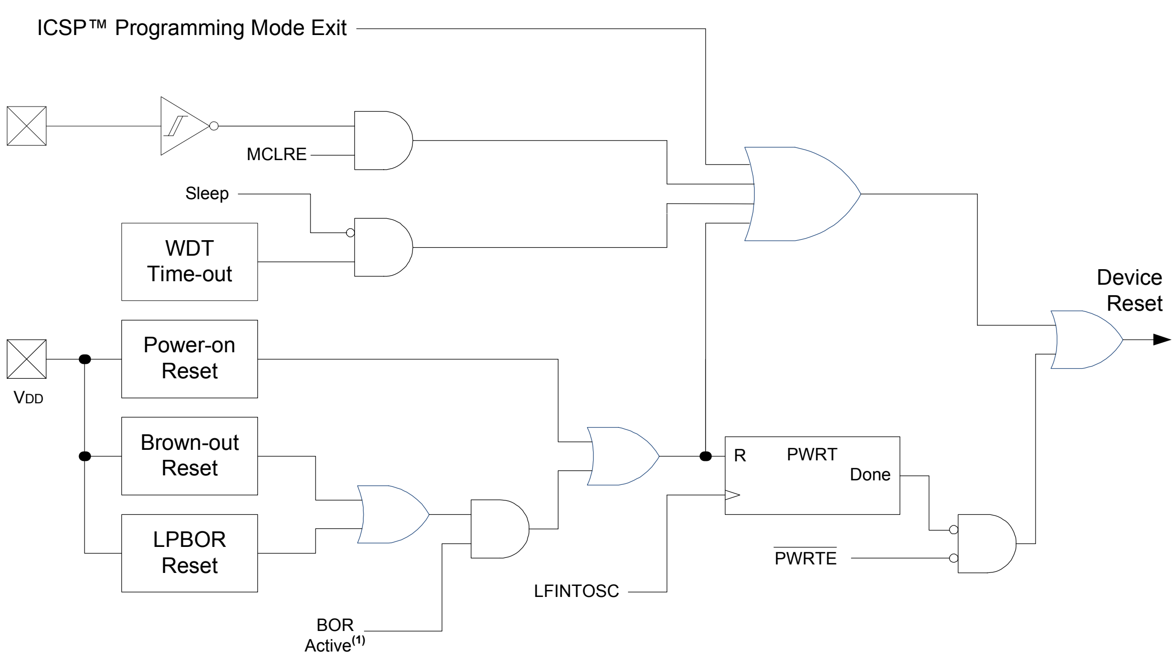
There are multiple ways to reset this device:

- Power-On Reset (POR)
- Brown-Out Reset (BOR)
- Low-Power Brown-Out Reset (LPBOR)
- $\overline{\text{MCLR}}$ Reset
- WDT Reset
- Programming mode exit

To allow VDD to stabilize, an optional Power-up Timer can be enabled to extend the Reset time after a BOR or POR event.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 5-1.

FIGURE 5-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



Note 1: See Table 5-1 for BOR active conditions.

5.1 Power-On Reset (POR)

The POR circuit holds the device in Reset until VDD has reached an acceptable level for minimum operation. Slow rising VDD, fast operating speeds or analog performance may require greater than minimum VDD. The PWRT, BOR or $\overline{\text{MCLR}}$ features can be used to extend the start-up period until all device operation conditions have been met.

5.1.1 POWER-UP TIMER (PWRT)

The Power-up Timer provides a nominal 64 ms time-out on POR or Brown-out Reset.

The device is held in Reset as long as PWRT is active. The PWRT delay allows additional time for the VDD to rise to an acceptable level. The Power-up Timer is enabled by clearing the $\overline{\text{PWRTE}}$ bit in Configuration Word.

The Power-up Timer starts after the release of the POR and BOR.

For additional information, refer to Application Note AN607, *"Power-up Trouble Shooting"* (DS00607).

5.2 Brown-Out Reset (BOR)

The BOR circuit holds the device in Reset when VDD reaches a selectable minimum level. Between the POR and BOR, complete voltage range coverage for execution protection can be implemented.

The Brown-out Reset module has four operating modes controlled by the BOREN<1:0> bits in Configuration Word. The four operating modes are:

- BOR is always on
- BOR is off when in Sleep
- BOR is controlled by software
- BOR is always off

Refer to Table 5-1 for more information.

The Brown-out Reset voltage level is selectable by configuring the BORV bit in Register 3-1.

A VDD noise rejection filter prevents the BOR from triggering on small events. If VDD falls below VBOR for a duration greater than parameter TBORDC, the device will reset. See Figure 5-2 for more information.

TABLE 5-1: BOR OPERATING MODES

BOREN<1:0>	SBOREN	Device Mode	BOR Mode	Device Operation upon: Release of POR/Wake- up from Sleep
11	X	X	Active	Waits for BOR ready(1) (BORRDY = 1)
10	X	Awake	Active	Waits for BOR ready (BORRDY = 1)
		Sleep	Disabled	
01	1	X	Active	Waits for BOR ready(1) (BORRDY = 1)
	0	X	Disabled	Begins immediately (BORRDY = x)
00	X	X	Disabled	

Note 1: In these specific cases, "Release of POR" and "Wake-up from Sleep", there is no delay in start-up. The BOR ready flag, (BORRDY = 1), will be set before the CPU is ready to execute instructions because the BOR circuit is forced on by the BOREN<1:0> bits.

5.2.1 BOR IS ALWAYS ON

When the BOREN bits of Configuration Word are programmed to '11', the BOR is always on. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is active during Sleep. The BOR does not delay wake-up from Sleep.

5.2.2 BOR IS OFF IN SLEEP

When the BOREN bits of Configuration Word are programmed to '10', the BOR is on, except in Sleep. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is not active during Sleep. The device wake-up will be delayed until the BOR is ready.

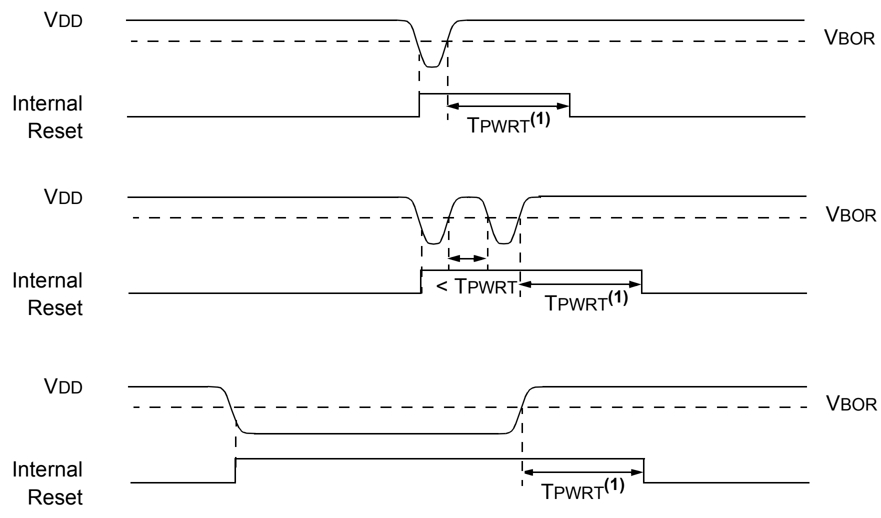
5.2.3 BOR CONTROLLED BY SOFTWARE

When the BOREN bits of Configuration Word are programmed to '01', the BOR is controlled by the SBOREN bit of the BORCON register. The device start-up is not delayed by the BOR ready condition or the VDD level.

BOR protection begins as soon as the BOR circuit is ready. The status of the BOR circuit is reflected in the BORRDY bit of the BORCON register.

BOR protection is unchanged by Sleep.

FIGURE 5-2: BROWN-OUT SITUATIONS



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

5.3 Register Definition: BOR Control

REGISTER 5-1: BORCON: BROWN-OUT RESET CONTROL REGISTER

R/W-1/u	R/W-0/u	U-0	U-0	U-0	U-0	U-0	R-q/u
SBOREN	BORFS(1)	—	—	—	—	—	BORRDY
bit 7							bit 0

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

bit 7 **SBOREN:** Software Brown-out Reset Enable bit
If BOREN <1:0> in Configuration Word ≠ 01:
SBOREN is read/write, but has no effect on the BOR.
If BOREN <1:0> in Configuration Word = 01:
1 = BOR enabled
0 = BOR disabled

bit 6 **BORFS:** Brown-out Reset Fast Start bit(1)
If BOREN<1:0> = 11 (Always on) or BOREN<1:0> = 00 (Always off)
BORFS is Read/Write, but has no effect.
If BOREN <1:0> = 10 (Disabled in Sleep) or BOREN<1:0> = 01 (Under software control):
1 = Band gap is forced on always (covers Sleep/wake-up/operating cases)
0 = Band gap operates normally, and may turn off

bit 5-1 **Unimplemented:** Read as '0'

bit 0 **BORRDY:** Brown-out Reset Circuit Ready Status bit
1 = The Brown-out Reset circuit is active
0 = The Brown-out Reset circuit is inactive

Note 1: BOREN<1:0> bits are located in Configuration Word.

5.4 Low-Power Brown-out Reset (LPBOR)

The Low-Power Brown-Out Reset (LPBOR) is an essential part of the Reset subsystem. Refer to Figure 5-1 to see how the BOR interacts with other modules.

The LPBOR is used to monitor the external VDD pin. When too low of a voltage is detected, the device is held in Reset. When this occurs, a register bit ($\overline{\text{BOR}}$) is changed to indicate that a BOR Reset has occurred. The same bit is set for both the BOR and the LPBOR. Refer to Register 5-2.

5.4.1 ENABLING LPBOR

The LPBOR is controlled by the LPBOR bit of Configuration Word. When the device is erased, the LPBOR module defaults to disabled.

5.4.1.1 LPBOR Module Output

The output of the LPBOR module is a signal indicating whether or not a Reset is to be asserted. This signal is OR'd together with the Reset signal of the BOR module to provide the generic $\overline{\text{BOR}}$ signal which goes to the PCON register and to the power control block.

5.5 $\overline{\text{MCLR}}$

The $\overline{\text{MCLR}}$ is an optional external input that can reset the device. The $\overline{\text{MCLR}}$ function is controlled by the MCLRE and the LVP bit of Configuration Word (Table 5-2).

TABLE 5-2: $\overline{\text{MCLR}}$ CONFIGURATION

MCLRE	LVP	$\overline{\text{MCLR}}$
0	0	Disabled
1	0	Enabled
x	1	Enabled

5.5.1 $\overline{\text{MCLR}}$ ENABLED

When $\overline{\text{MCLR}}$ is enabled and the pin is held low, the device is held in Reset. The $\overline{\text{MCLR}}$ pin is connected to VDD through an internal weak pull-up.

The device has a noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses.

Note: A Reset does not drive the $\overline{\text{MCLR}}$ pin low.

5.5.2 $\overline{\text{MCLR}}$ DISABLED

When $\overline{\text{MCLR}}$ is disabled, the pin functions as a general purpose input and the internal weak pull-up is under software control.

5.6 Watchdog Timer (WDT) Reset

The Watchdog Timer generates a Reset if the firmware does not issue a `CLRWDT` instruction within the time-out period. The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits in the STATUS register are changed to indicate the WDT Reset. See **Section 8.0 "Watchdog Timer"** for more information.

5.7 Programming Mode ICSP Exit

Upon exit of Programming mode, the device will behave as if a POR had just occurred.

5.8 Power-Up Timer

The Power-up Timer optionally delays device execution after a BOR or POR event. This timer is typically used to allow VDD to stabilize before allowing the device to start running.

The Power-up Timer is controlled by the $\overline{\text{PWRTE}}$ bit of Configuration Word.

5.9 Start-up Sequence

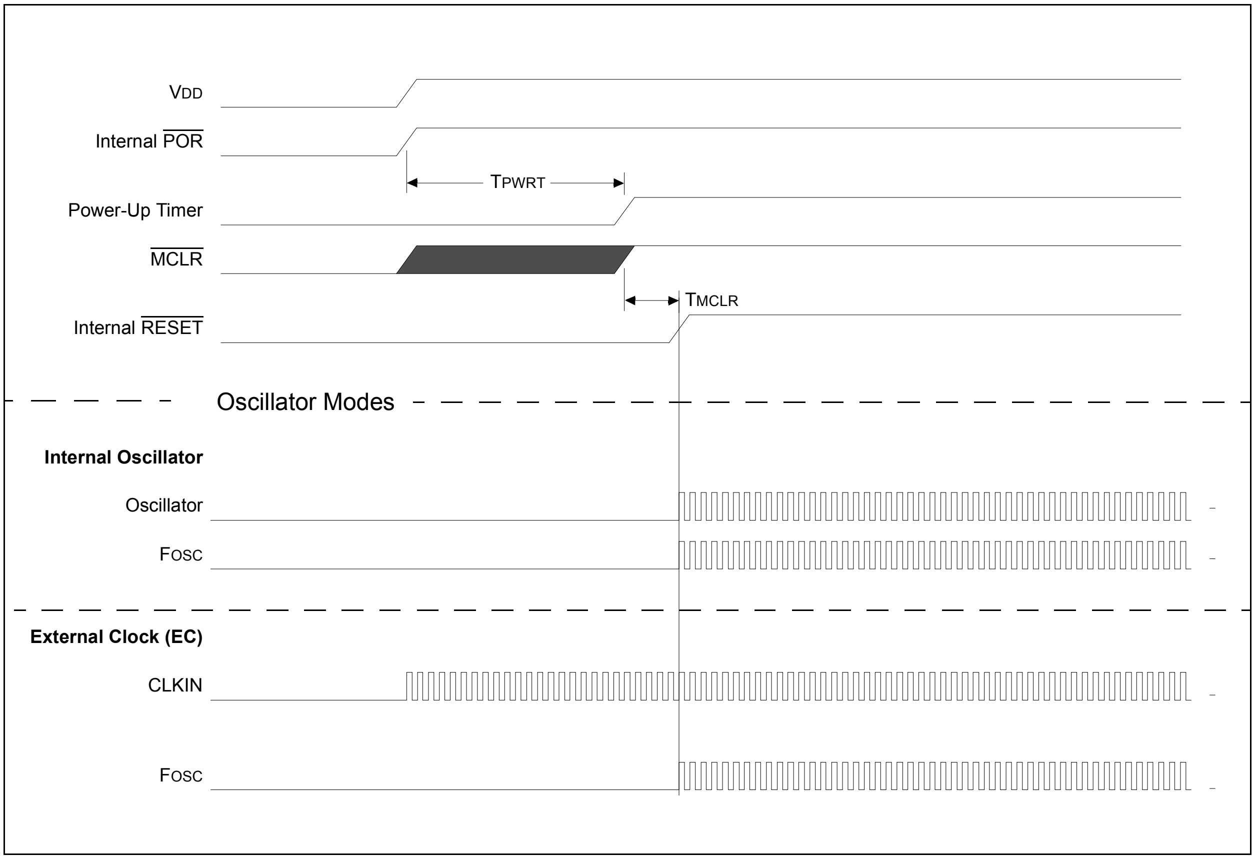
Upon the release of a POR or BOR, the following must occur before the device will begin executing:

1. Power-up Timer runs to completion (if enabled).
2. $\overline{\text{MCLR}}$ must be released (if enabled).

The total time-out will vary based on oscillator configuration and Power-up Timer configuration. See **Section 4.0 "Oscillator Module"** for more information.

The Power-up Timer runs independently of $\overline{\text{MCLR}}$ Reset. If $\overline{\text{MCLR}}$ is kept low long enough, the Power-up Timer will expire. Upon bringing $\overline{\text{MCLR}}$ high, the device will begin execution after 10 FOSC cycles (see Figure 5-3). This is useful for testing purposes or to synchronize more than one device operating in parallel.

FIGURE 5-3: RESET START-UP SEQUENCE



5.10 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON register are updated to indicate the cause of the Reset. Table 5-3 and Table 5-4 show the Reset conditions of these registers.

TABLE 5-3: RESET STATUS BITS AND THEIR SIGNIFICANCE

$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Condition
0	x	1	1	Power-on Reset
u	0	1	1	Brown-out Reset
u	u	0	u	WDT Reset
u	u	0	0	WDT Wake-up from Sleep
u	u	u	u	$\overline{\text{MCLR}}$ Reset during normal operation
u	u	1	0	$\overline{\text{MCLR}}$ Reset during Sleep

TABLE 5-4: RESET CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	0000h	0001 1000	---- --0x
$\overline{\text{MCLR}}$ Reset during normal operation	0000h	000u uuuu	---- --uu
$\overline{\text{MCLR}}$ Reset during Sleep	0000h	0001 0uuu	---- --uu
WDT Reset	0000h	0000 uuuu	---- --uu
WDT Wake-up from Sleep	PC + 1	0000 0uuu	---- --uu
Brown-out Reset	0000h	0001 1uuu	---- --u0
Interrupt Wake-up from Sleep	PC + 1(1)	0001 0uuu	---- --uu

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

Note 1: When the wake-up is due to an interrupt and Global Enable bit (GIE) is set, the return address is pushed on the stack and PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

5.11 Power Control (PCON) Register

The Power Control (PCON) register contains flag bits to differentiate between a:

- Power-On Reset ($\overline{\text{POR}}$)
- Brown-Out Reset ($\overline{\text{BOR}}$)

The PCON register bits are shown in Register 5-2.

5.12 Register Definition: Power Control

REGISTER 5-2: PCON: POWER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W/HC-q/u	R/W/HC-q/u
—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

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

bit 7-2 **Unimplemented:** Read as ‘0’

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

TABLE 5-5: SUMMARY OF REGISTERS ASSOCIATED WITH RESETS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	30
PCON	—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	34
STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	13
WDTCON	—	—	WDTPS<4:0>					SWDTEN	48

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used by Resets.

TABLE 5-6: SUMMARY OF CONFIGURATION WORD WITH RESETS

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG	13:8	—	—	—	WRT<1:0>		BORV	LPBOR	LVP	20
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		BOREN<1:0>		FOSC	

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used by Reset.

6.0 INTERRUPTS

The interrupt feature allows certain events to preempt normal program flow. Firmware is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

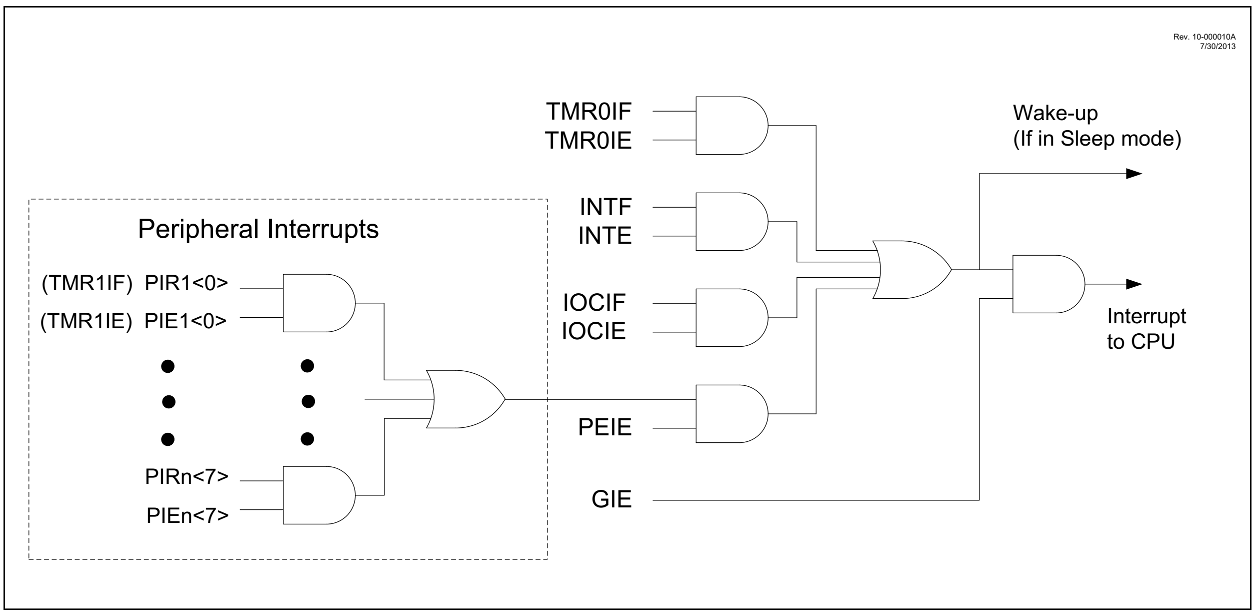
This chapter contains the following information for Interrupts:

- Operation
- Interrupt Latency
- Interrupts During Sleep
- INT Pin
- Context Saving during Interrupts

Many peripherals produce interrupts. Refer to the corresponding chapters for details.

A block diagram of the interrupt logic is shown in Figure 6-1.

FIGURE 6-1: INTERRUPT LOGIC



6.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt Enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PIE1 register)

The INTCON and PIR1 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- PC is loaded with the interrupt vector 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack, and setting the GIE bit.

For additional information on a specific interrupt's operation, refer to its peripheral chapter.

Note 1:	Individual interrupt flag bits are set, regardless of the state of any other enable bits.
2:	All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

6.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three or four instruction cycles. For asynchronous interrupts, the latency is three to five instruction cycles, depending on when the interrupt occurs. See Figure 6-2 and **Section 6.3 "Interrupts During Sleep"** for more details.

FIGURE 6-2: INTERRUPT LATENCY

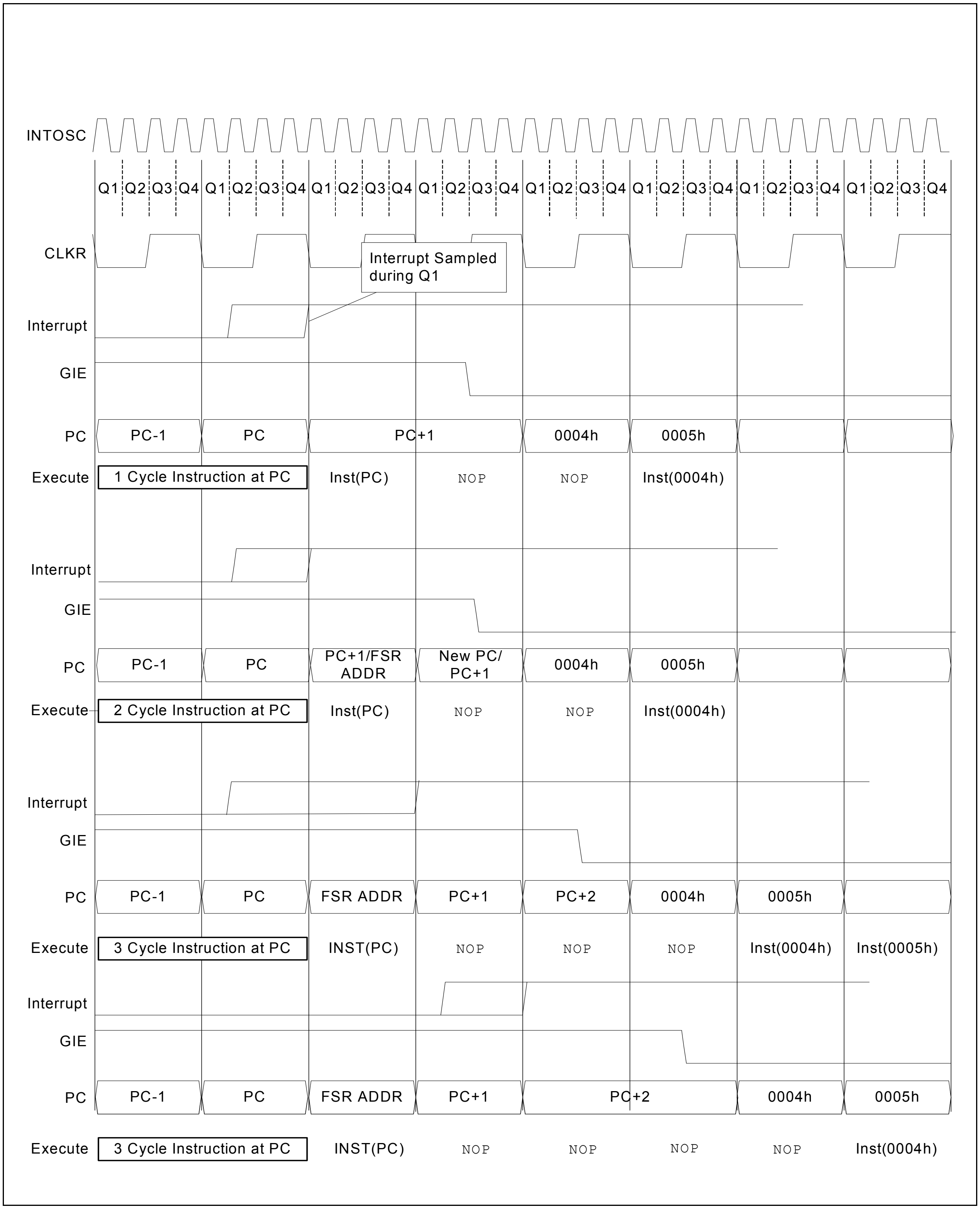
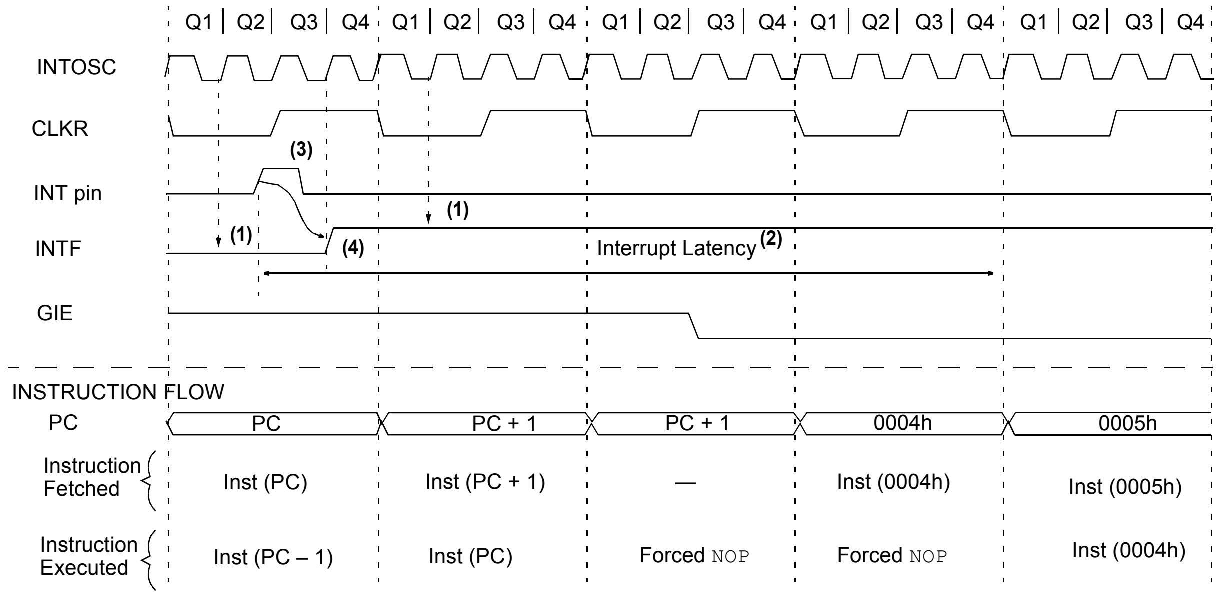


FIGURE 6-3: INT PIN INTERRUPT TIMING



Note 1: INTF flag is sampled here (every Q1).

2: Asynchronous interrupt latency = 3-5 TCY. Synchronous latency = 3-4 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

3: For minimum width of INT pulse, refer to AC specifications in **Section 24.0 "Electrical Specifications"**.

4: INTF is enabled to be set any time during the Q4-Q1 cycles.

6.3 Interrupts During Sleep

Some interrupts can be used to wake from Sleep. To wake from Sleep, the peripheral must be able to operate without the system clock. The interrupt source must have the appropriate Interrupt Enable bit(s) set prior to entering Sleep.

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the `SLEEP` instruction. The instruction directly after the `SLEEP` instruction will always be executed before branching to the ISR. Refer to the **Section 7.0 "Power-Down Mode (Sleep)"** for more details.

6.4 INT Pin

The INT pin can be used to generate an asynchronous edge-triggered interrupt. This interrupt is enabled by setting the INTE bit of the INTCON register. The INTEDG bit of the OPTION_REG register determines on which edge the interrupt will occur. When the INTEDG bit is set, the rising edge will cause the interrupt. When the INTEDG bit is clear, the falling edge will cause the interrupt. The INTF bit of the INTCON register will be set when a valid edge appears on the INT pin. If the GIE and INTE bits are also set, the processor will redirect program execution to the interrupt vector.

6.5 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (e.g., W and STATUS registers). This must be implemented in software.

Temporary holding registers W_TEMP and STATUS_TEMP should be placed in the last 16 bytes of GPR (see Table 1-2). This makes context save and restore operations simpler. The code shown in Example 6-1 can be used to:

- Store the W register
- Store the STATUS register
- Execute the ISR code
- Restore the Status (and Bank Select Bit register)
- Restore the W register

Note:	These devices do not require saving the PCLATH. However, if computed `GOTO`s are used in both the ISR and the main code, the PCLATH must be saved and restored in the ISR.

EXAMPLE 6-1: SAVING STATUS AND W REGISTERS IN RAM

```
MOVWF   W_TEMP              ;Copy W to TEMP register
SWAPF   STATUS,W            ;Swap status to be saved into W
                            ;Swaps are used because they do not affect the status bits
MOVWF   STATUS_TEMP         ;Save status to bank zero STATUS_TEMP register
:
:(ISR)                      ;Insert user code here
:
SWAPF   STATUS_TEMP,W       ;Swap STATUS_TEMP register into W
                            ;(sets bank to original state)
MOVWF   STATUS              ;Move W into STATUS register
SWAPF   W_TEMP,F            ;Swap W_TEMP
SWAPF   W_TEMP,W            ;Swap W_TEMP into W
```


6.6 Interrupt Control Registers

REGISTER 6-1: INTCON: INTERRUPT 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-0/0
GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF(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 **GIE:** Global Interrupt Enable bit
1 = Enables all active interrupts
0 = Disables all interrupts

bit 6 **PEIE:** Peripheral Interrupt Enable bit
1 = Enables all active peripheral interrupts
0 = Disables all peripheral interrupts

bit 5 **TMR0IE:** Timer0 Overflow Interrupt Enable bit
1 = Enables the Timer0 interrupt
0 = Disables the Timer0 interrupt

bit 4 **INTE:** INT External Interrupt Enable bit
1 = Enables the INT external interrupt
0 = Disables the INT external interrupt

bit 3 **IOCIE:** Interrupt-on-Change Interrupt Enable bit
1 = Enables the interrupt-on-change interrupt
0 = Disables the interrupt-on-change interrupt

bit 2 **TMR0IF:** Timer0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed
0 = TMR0 register did not overflow

bit 1 **INTF:** INT External Interrupt Flag bit
1 = The INT external interrupt occurred
0 = The INT external interrupt did not occur

bit 0 **IOCIF:** Interrupt-on-Change Interrupt Flag bit(1)
1 = When at least one of the interrupt-on-change pins changed state
0 = None of the interrupt-on-change pins have changed state

Note 1: The IOCIF Flag bit is read-only and cleared when all the Interrupt-on-Change flags in the IOCAF register have been cleared by software.

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

REGISTER 6-2: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0
—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—
bit 7							bit 0

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

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

bit 6 **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables the A/D converter interrupt
0 = Disables the A/D converter interrupt

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

bit 4 **NCO1IE:** Numerically Controlled Oscillator Interrupt Enable bit
1 = Enables the NCO overflow interrupt
0 = Disables the NCO overflow interrupt

bit 3 **CLC1IE:** Configurable Logic Block Interrupt Enable bit
1 = Enables the CLC interrupt
0 = Disables the CLC interrupt

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

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 Match interrupt
0 = Disables the TMR2 to PR2 Match interrupt

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

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

REGISTER 6-3: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1

U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0
—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—
bit 7							bit 0

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

bit 7 **Unimplemented**: Read as ‘0’

bit 6 **ADIF:** A/D Converter Interrupt Flag bit
1 = The A/D conversion completed
0 = The A/D conversion is not complete

bit 5 **Unimplemented**: Read as ‘0’

bit 4 **NCO1IF:** Numerically Controlled Oscillator Interrupt Flag bit
1 = NCO1 overflow occurred (must be cleared in software)
0 = No NCO1 overflow

bit 3 **CLC1IF:** Configurable Logic Block Rising Edge Interrupt Flag bit
1 = CLC interrupt occurred (must be cleared in software)
0 = No CLC Interrupt

bit 2 **Unimplemented**: Read as ‘0’

bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match
Note: The match must occur the number of times specified by the TMR2 postscaler (Register 17-1).

bit 0 **Unimplemented**: Read as ‘0’

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

TABLE 6-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
IOCAF	—	—	—	—	IOCAF3	IOCAF2	IOCAF1	IOCAF0	76
IOCAN	—	—	—	—	IOCAN3	IOCAN2	IOCAN1	IOCAN0	75
IOCAP	—	—	—	—	IOCAP3	IOCAP2	IOCAP1	IOCAP0	75
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	T0CS	T0SE	PSA	PS<2:0>			95
PIE1	—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—	41
PIR1	—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—	42

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by Interrupts.

7.0 POWER-DOWN MODE (SLEEP)

The Power-down mode is entered by executing a `SLEEP` instruction.

Upon entering Sleep mode, the following conditions exist:

1. WDT will be cleared but keeps running, if enabled for operation during Sleep.
2. $\overline{PD}$ bit of the STATUS register is cleared.
3. $\overline{TO}$ bit of the STATUS register is set.
4. CPU clock is disabled.
5. 31 kHz LFINTOSC is unaffected and peripherals that operate from it may continue operation in Sleep.
6. ADC is unaffected, if the dedicated FRC clock is selected.
7. I/O ports maintain the status they had before `SLEEP` was executed (driving high, low or high-impedance).
8. Resets other than WDT are not affected by Sleep mode.

Refer to individual chapters for more details on peripheral operation during Sleep.

To minimize current consumption, the following conditions should be considered:

- I/O pins should not be floating
- External circuitry sinking current from I/O pins
- Internal circuitry sourcing current from I/O pins
- Current draw from pins with internal weak pull-ups
- Modules using 31 kHz LFINTOSC
- CWG and NCO modules using HFINTOSC

I/O pins that are high-impedance inputs should be pulled to V_{DD} or V_{SS} externally to avoid switching currents caused by floating inputs.

Examples of internal circuitry that might be sourcing current include the FVR module. See **Section 12.0 "Fixed Voltage Reference (FVR)"** for more information on these modules.

7.1 Wake-up from Sleep

The device can wake-up from Sleep through one of the following events:

1. External Reset input on $\overline{MCLR}$ pin, if enabled
2. BOR Reset, if enabled
3. POR Reset
4. Watchdog Timer, if enabled
5. Any external interrupt
6. Interrupts by peripherals capable of running during Sleep (see individual peripheral for more information)

The first three events will cause a device Reset. The last three events are considered a continuation of program execution. To determine whether a device Reset or wake-up event occurred, refer to **Section 5.10 "Determining the Cause of a Reset"**.

When the `SLEEP` instruction is being executed, the next instruction (PC + 1) is prefetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be enabled. Wake-up will occur regardless of the state of the GIE bit. If the GIE bit is disabled, the device continues execution at the instruction after the `SLEEP` instruction. If the GIE bit is enabled, the device executes the instruction after the `SLEEP` instruction, the device will then call the Interrupt Service Routine. In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have a `NOP` after the `SLEEP` instruction.

The WDT is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

The Complementary Waveform Generator (CWG) and the Numerically Controlled Oscillator (NCO) modules can utilize the HFINTOSC oscillator as their respective clock source. Under certain conditions, when the HFINTOSC is selected for use with the CWG or NCO modules, the HFINTOSC will remain active during Sleep. This will have a direct effect on the Sleep mode current. Please refer to **21.0 "Complementary Waveform Generator (CWG) Module"** and **20.0 "Numerically Controlled Oscillator (NCO) Module"** for more information.

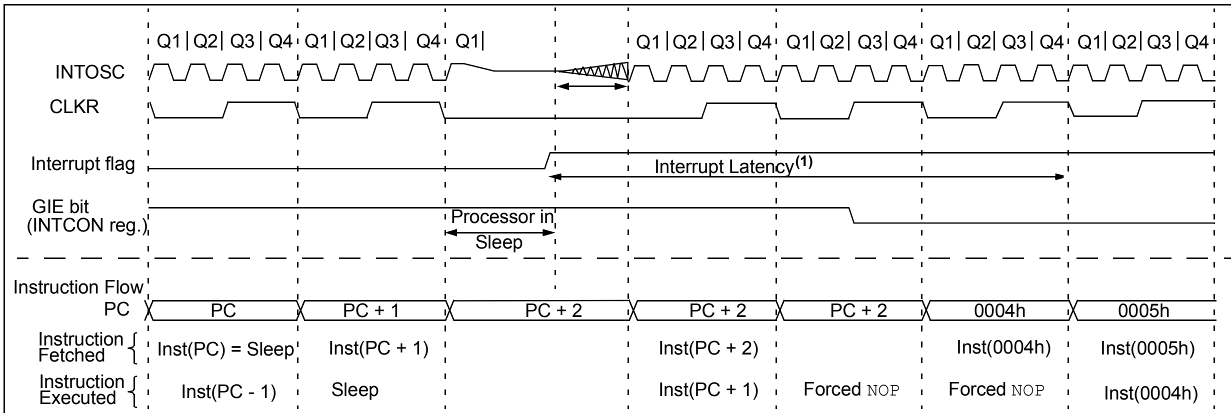
7.1.1 WAKE-UP USING INTERRUPTS

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 7-1: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: GIE = `1` assumed. In this case after wake-up, the processor calls the ISR at 0004h. If GIE = `0`, execution will continue in-line.

TABLE 7-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	13
WDTCON	—	—	WDTPS<4:0>					SWDTEN	48

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used in Power-down mode.

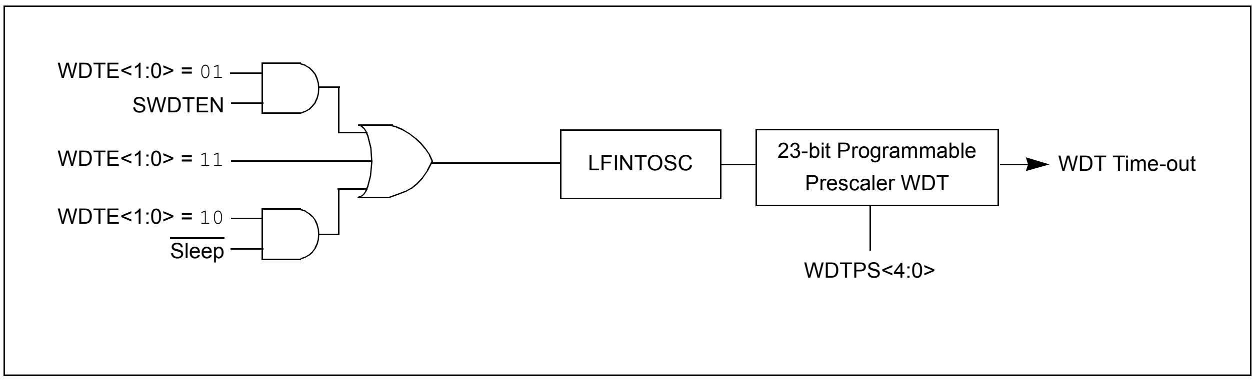
8.0 WATCHDOG TIMER

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 (typical)
- Multiple Reset conditions
- Operation during Sleep

FIGURE 8-1: WATCHDOG TIMER BLOCK DIAGRAM



8.1 Independent Clock Source

The WDT derives its time base from the 31 kHz LFINTOSC internal oscillator. Time intervals in this chapter are based on a nominal interval of 1ms. See **Section 24.0 "Electrical Specifications"** for the LFINTOSC tolerances.

8.2 WDT Operating Modes

The Watchdog Timer module has four operating modes controlled by the WDTE<1:0> bits in Configuration Word. See Table 8-1.

8.2.1 WDT IS ALWAYS ON

When the WDTE bits of Configuration Word are set to '11', the WDT is always on.

WDT protection is active during Sleep.

8.2.2 WDT IS OFF IN SLEEP

When the WDTE bits of Configuration Word are set to '10', the WDT is on, except in Sleep.

WDT protection is not active during Sleep.

8.2.3 WDT CONTROLLED BY SOFTWARE

When the WDTE bits of Configuration Word are set to '01', the WDT is controlled by the SWDTEN bit of the WDTCON register.

WDT protection is unchanged by Sleep. See Table 8-1 for more details.

TABLE 8-1: WDT OPERATING MODES

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

8.3 Time-Out Period

The WDTPS bits of the WDTCON register set the time-out period from 1 ms to 256 seconds (nominal). After a Reset, the default time-out period is 2 seconds.

8.4 Clearing the WDT

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

- Any Reset
- CLRWDT instruction is executed
- Device enters Sleep
- Device wakes up from Sleep
- Oscillator fail
- WDT is disabled

See Table 8-2 for more information.

8.5 Operation During Sleep

When the device enters Sleep, the WDT is cleared. If the WDT is enabled during Sleep, the WDT resumes counting.

When the device exits Sleep, the WDT is cleared again.

When a WDT time-out occurs while the device is in Sleep, no Reset is generated. Instead, the device wakes up and resumes operation. The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits in the STATUS register are changed to indicate the event. See **Section 2.0 "Memory Organization"** and ***Register 2-1*** for more information.

TABLE 8-2: WDT CLEARING CONDITIONS

Conditions	WDT
WDTE<1:0> = 00	Cleared
WDTE<1:0> = 01 and SWDTEN = 0	
WDTE<1:0> = 10 and enter Sleep	
CLRWDT Command	
Exit Sleep	
Change INTOSC divider (IRCF bits)	Unaffected

8.6 Watchdog Control Register

REGISTER 8-1: WDTCON: WATCHDOG TIMER CONTROL REGISTER

U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-1/1	R/W-0/0
—	—	WDTPS<4:0>					SWDTEN
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 5-1 **WDTPS<4:0>:** Watchdog Timer Period Select bits(1)

Bit Value = Prescale Rate

11111 = Reserved. Results in minimum interval (1:32)

•
•
•

10011 = Reserved. Results in minimum interval (1:32)

10010 = 1:8388608 (2^{23}) (Interval 256s nominal)
10001 = 1:4194304 (2^{22}) (Interval 128s nominal)
10000 = 1:2097152 (2^{21}) (Interval 64s nominal)
01111 = 1:1048576 (2^{20}) (Interval 32s nominal)
01110 = 1:524288 (2^{19}) (Interval 16s nominal)
01101 = 1:262144 (2^{18}) (Interval 8s nominal)
01100 = 1:131072 (2^{17}) (Interval 4s nominal)
01011 = 1:65536 (Interval 2s nominal) (Reset value)
01010 = 1:32768 (Interval 1s nominal)
01001 = 1:16384 (Interval 512 ms nominal)
01000 = 1:8192 (Interval 256 ms nominal)
00111 = 1:4096 (Interval 128 ms nominal)
00110 = 1:2048 (Interval 64 ms nominal)
00101 = 1:1024 (Interval 32 ms nominal)
00100 = 1:512 (Interval 16 ms nominal)
00011 = 1:256 (Interval 8 ms nominal)
00010 = 1:128 (Interval 4 ms nominal)
00001 = 1:64 (Interval 2 ms nominal)
00000 = 1:32 (Interval 1 ms nominal)

bit 0 **SWDTEN:** Software Enable/Disable for Watchdog Timer bit

If WDTE<1:0> = 00:
This bit is ignored.
If WDTE<1:0> = 01:
1 = WDT is turned on
0 = WDT is turned off
If WDTE<1:0> = 1x:
This bit is ignored.

Note 1: Times are approximate. WDT time is based on 31 kHz LFINTOSC.

TABLE 8-3: SUMMARY OF REGISTERS ASSOCIATED WITH WATCHDOG TIMER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON	—	IRCF<2:0>			HFIOFR	—	LFIOFR	HFIOFS	26
STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	13
WDTCON	—	—	WDTPS<4:0>					SWDTEN	48

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by Watchdog Timer.

TABLE 8-4: SUMMARY OF CONFIGURATION WORD WITH WATCHDOG TIMER

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG	13:8	—	—	—	WRT<1:0>		BORV	LPBOR	LVP	20
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		BOREN<1:0>		FOSC	

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by Watchdog Timer.

9.0 FLASH PROGRAM MEMORY CONTROL

The Flash program memory is readable and writable during normal operation over the full VDD range. Program memory is indirectly addressed using Special Function Registers (SFRs). The SFRs used to access program memory are:

- PMCON1
- PMCON2
- PMDATL
- PMDATH
- PMADRL
- PMADRH

When accessing the program memory, the PMDATH:PMDATL register pair forms a 2-byte word that holds the 14-bit data for read/write, and the PMADRH:PMADRL register pair forms a 2-byte word that holds the 9-bit address of the program memory location being read.

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

The Flash program memory can be protected in two ways; by code protection ($\overline{\text{CP}}$ bit in Configuration Word) and write protection (WRT<1:0> bits in Configuration Word).

Code protection ($\overline{\text{CP}}$ = 0)(1), disables access, reading and writing, to the Flash program memory via external device programmers. Code protection does not affect the self-write and erase functionality. Code protection can only be reset by a device programmer performing a Bulk Erase to the device, clearing all Flash program memory, Configuration bits and User IDs.

Write protection prohibits self-write and erase to a portion or all of the Flash program memory as defined by the bits WRT<1:0>. Write protection does not affect a device programmers ability to read, write or erase the device.

Note 1:	Code protection of the entire Flash program memory array is enabled by clearing the $\overline{\text{CP}}$ bit of Configuration Word.

9.1 PMADRL and PMADRH Registers

The PMADRH:PMADRL register pair can address up to a maximum of 512 words of program memory. When selecting a program address value, the MSB of the address is written to the PMADRH register and the LSB is written to the PMADRL register.

9.1.1 PMCON1 AND PMCON2 REGISTERS

PMCON1 is the control register for Flash program memory accesses.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared by hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation to occur. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a Reset during normal operation. In these situations, following Reset, the user can check the WRERR bit and execute the appropriate error handling routine.

The PMCON2 register is a write-only register. Attempting to read the PMCON2 register will return all '0's.

To enable writes to the program memory, a specific pattern (the unlock sequence), must be written to the PMCON2 register. The required unlock sequence prevents inadvertent writes to the program memory write latches and Flash program memory.

9.2 Flash Program Memory Overview

It is important to understand the Flash program memory structure for erase and programming operations. Flash program memory is arranged in rows. A row consists of a fixed number of 14-bit program memory words. A row is the minimum size that can be erased by user software.

After a row has been erased, the user can reprogram all or a portion of this row. Data to be written into the program memory row is written to 14-bit wide data write latches. These write latches are not directly accessible to the user, but may be loaded via sequential writes to the PMDATH:PMDATL register pair.

Note:	If the user wants to modify only a portion of a previously programmed row, then the contents of the entire row must be read and saved in RAM prior to the erase. Then, new data and retained data can be written into the write latches to reprogram the row of Flash program memory. However, any unprogrammed locations can be written without first erasing the row. In this case, it is not necessary to save and rewrite the other previously programmed locations.

See Table 9-1 for Erase Row size and the number of write latches for Flash program memory.

TABLE 9-1: FLASH MEMORY ORGANIZATION BY DEVICE

Device	Row Erase (words)	Write Latches (words)
PIC10(L)F320	16	16
PIC10(L)F322		

9.2.1 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must:

1. Write the desired address to the PMADRH:PMADRL register pair.
2. Clear the CFGS bit of the PMCON1 register.
3. Then, set control bit RD of the PMCON1 register.

Once the read control bit is set, the program memory Flash controller will use the second instruction cycle to read the data. This causes the second instruction immediately following the "`BSF PMCON1,RD`" instruction to be ignored. The data is available in the very next cycle, in the PMDATH:PMDATL register pair; therefore, it can be read as two bytes in the following instructions.

PMDATH:PMDATL register pair will hold this value until another read or until it is written to by the user.

> **Note:** The two instructions following a program memory read are required to be `NOP`s. This prevents the user from executing a two-cycle instruction on the next instruction after the RD bit is set.

FIGURE 9-1: FLASH PROGRAM MEMORY READ FLOWCHART

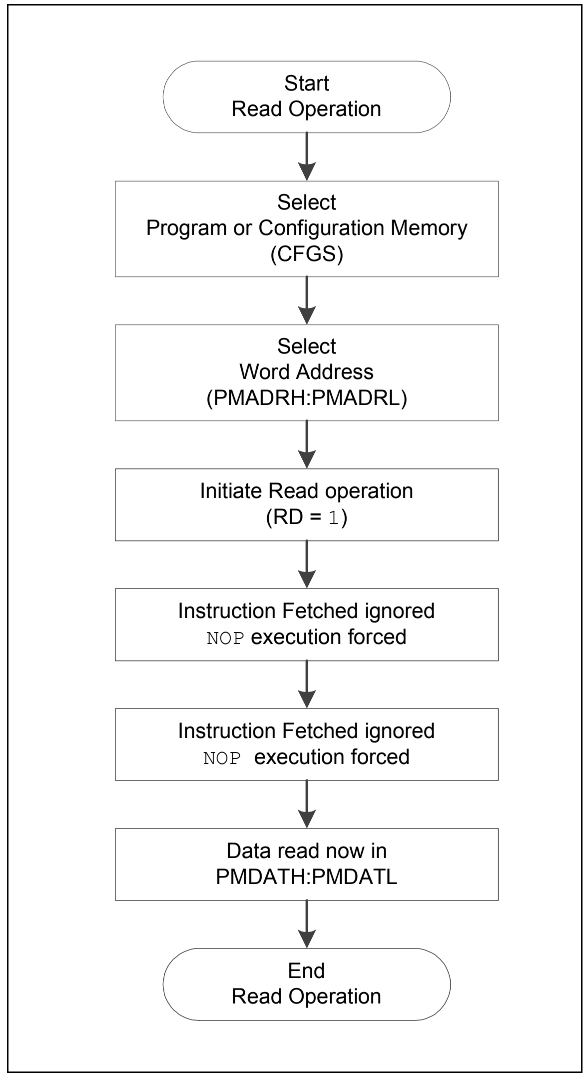
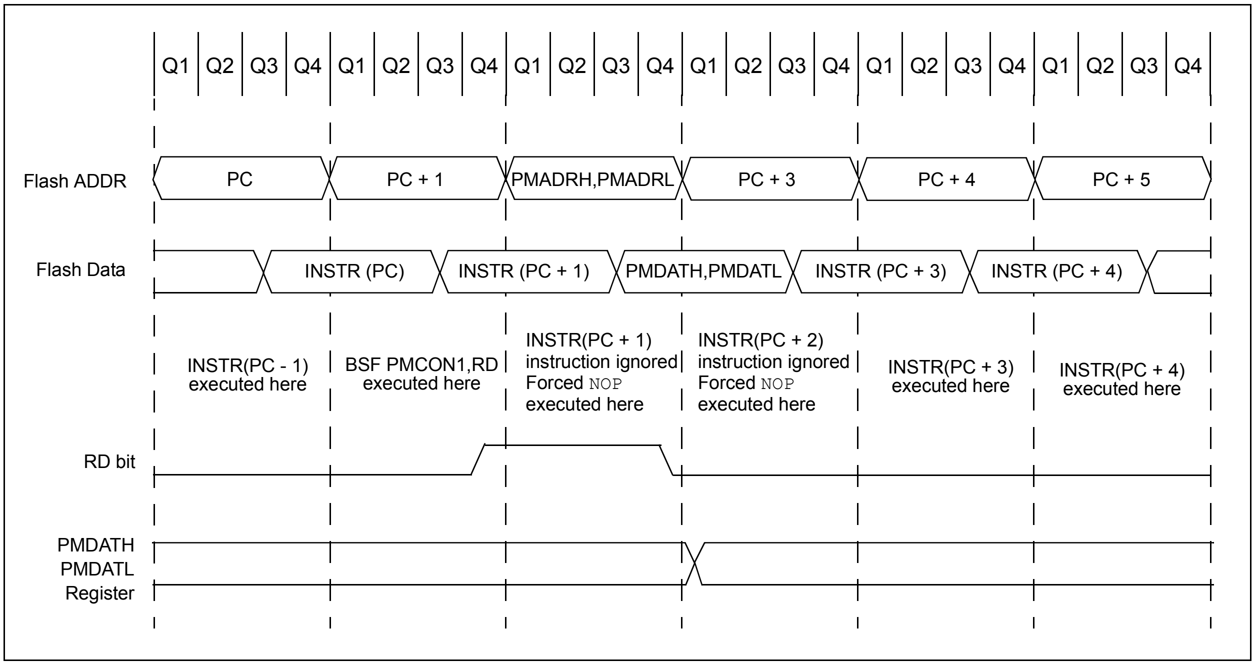


FIGURE 9-2: FLASH PROGRAM MEMORY READ CYCLE EXECUTION



EXAMPLE 9-1: FLASH PROGRAM MEMORY READ

```
* This code block will read 1 word of program
* memory at the memory address:
    PROG_ADDR_HI: PROG_ADDR_LO
*   data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

    BANKSEL   PMADRL            ; not required on devices with 1 Bank of SFRs
    MOVLW     PROG_ADDR_LO      ;
    MOVWF     PMADRL            ; Store LSB of address
    MOVLW     PROG_ADDR_HI      ;
    MOVWF     PMADRH            ; Store MSB of address

    BCF       PMCON1,CFGS       ; Do not select Configuration Space
    BSF       PMCON1,RD         ; Initiate read
    NOP                         ; Ignored (Figure 9-2)
    NOP                         ; Ignored (Figure 9-2)

    MOVF      PMDATL,W          ; Get LSB of word
    MOVWF     PROG_DATA_LO      ; Store in user location
    MOVF      PMDATH,W          ; Get MSB of word
    MOVWF     PROG_DATA_HI      ; Store in user location
```

9.2.2 FLASH MEMORY UNLOCK SEQUENCE

Note:	A delay of at least 100 μs is required after Power-On Reset (POR) before executing a Flash memory unlock sequence.

The unlock sequence is a mechanism that protects the Flash program memory from unintended self-write programming or erasing. The sequence must be executed and completed without interruption to successfully complete any of the following operations:

- Row Erase
- Load program memory write latches
- Write of program memory write latches to program memory
- Write of program memory write latches to User IDs

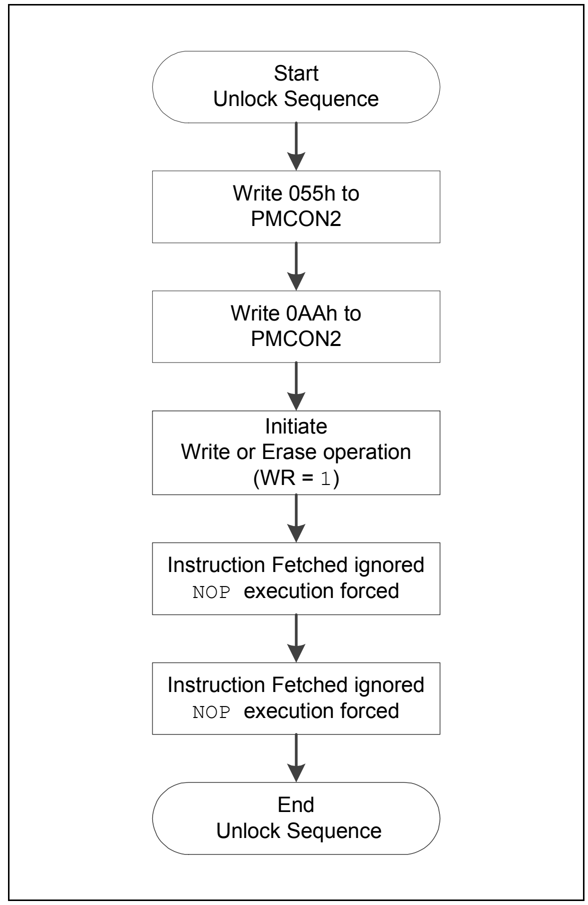
The unlock sequence consists of the following steps:

1. Write 55h to PMCON2
2. Write AAh to PMCON2
3. Set the WR bit in PMCON1
4. `NOP` instruction
5. `NOP` instruction

Once the WR bit is set, the processor will always force two `NOP` instructions. When an Erase Row or Program Row operation is being performed, the processor will stall internal operations (typical 2 ms), until the operation is complete and then resume with the next instruction. When the operation is loading the program memory write latches, the processor will always force the two `NOP` instructions and continue uninterrupted with the next instruction.

Since the unlock sequence must not be interrupted, global interrupts should be disabled prior to the unlock sequence and re-enabled after the unlock sequence is completed.

FIGURE 9-3: FLASH PROGRAM MEMORY UNLOCK SEQUENCE FLOWCHART



9.2.3 ERASING FLASH PROGRAM MEMORY

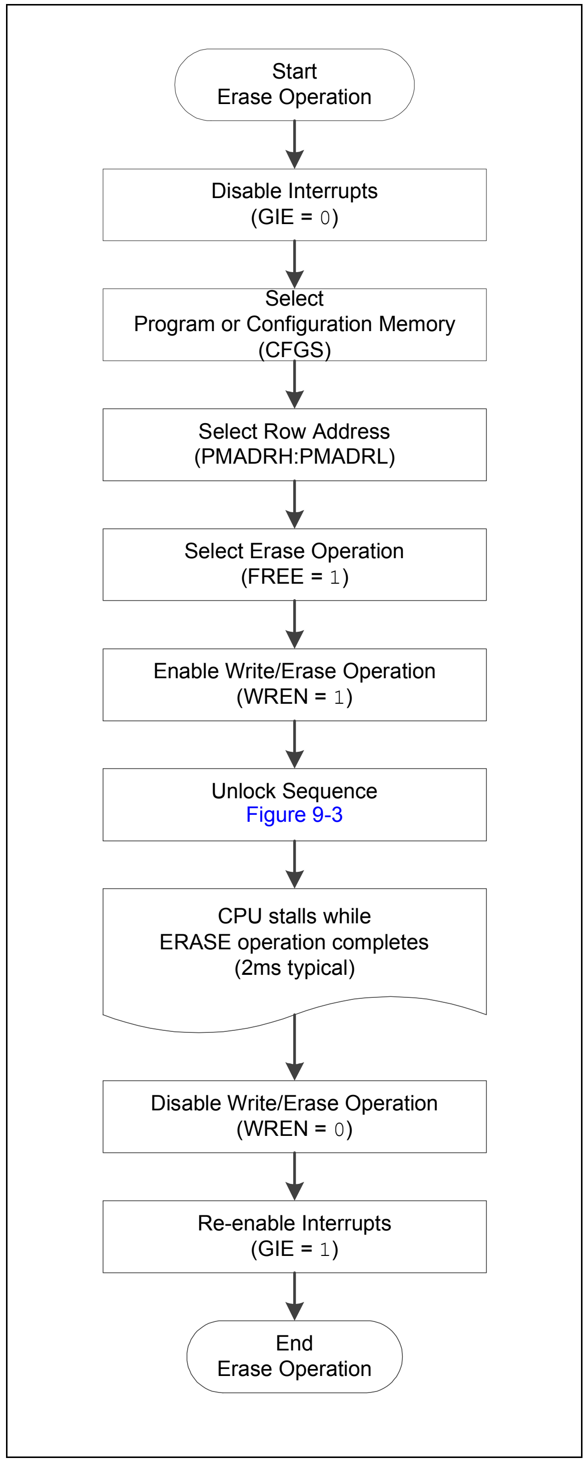
While executing code, program memory can only be erased by rows. To erase a row:

1. Load the PMADRH:PMADRL register pair with any address within the row to be erased.
2. Clear the CFGS bit of the PMCON1 register.
3. Set the FREE and WREN bits of the PMCON1 register.
4. Write 55h, then AAh, to PMCON2 (Flash programming unlock sequence).
5. Set control bit WR of the PMCON1 register to begin the erase operation.

See Example 9-2.

After the “`BSF PMCON1,WR`” instruction, the processor requires two cycles to set up the erase operation. The user must place two `NOP` instructions after the WR bit is set. The processor will halt internal operations for the typical 2 ms erase time. This is not Sleep mode as the clocks and peripherals will continue to run. After the erase cycle, the processor will resume operation with the third instruction after the PMCON1 write instruction.

FIGURE 9-4: FLASH PROGRAM MEMORY ERASE FLOWCHART



EXAMPLE 9-2: ERASING ONE ROW OF PROGRAM MEMORY

```
; This row erase routine assumes the following:
; 1. A valid address within the erase row is loaded in ADDRH:ADDRL
; 2. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)


        BCF         INTCON,GIE      ; Disable ints so required sequences will execute properly
        BANKSEL     PMADRL          ; not required on devices with 1 Bank of SFRs
        MOVF        ADDRL,W         ; Load lower 8 bits of erase address boundary
        MOVWF       PMADRL
        MOVF        ADDRH,W         ; Load upper 6 bits of erase address boundary
        MOVWF       PMADRH
        BCF         PMCON1,CFGS     ; Not configuration space
        BSF         PMCON1,FREE     ; Specify an erase operation
        BSF         PMCON1,WREN     ; Enable writes

Required    MOVLW       55h             ; Start of required sequence to initiate erase
Sequence    MOVWF       PMCON2          ; Write 55h
        MOVLW       0AAh            ;
        MOVWF       PMCON2          ; Write AAh
        BSF         PMCON1,WR       ; Set WR bit to begin erase
        NOP                         ; NOP instructions are forced as processor starts
        NOP                         ; row erase of program memory.
                                    ;
                                    ; The processor stalls until the erase process is complete
                                    ; after erase processor continues with 3rd instruction

        BCF         PMCON1,WREN     ; Disable writes
        BSF         INTCON,GIE      ; Enable interrupts
```

9.2.4 WRITING TO FLASH PROGRAM MEMORY

Program memory is programmed using the following steps:

1. Load the address in PMADRH:PMADRL of the row to be programmed.
2. Load each write latch with data.
3. Initiate a programming operation.
4. Repeat steps 1 through 3 until all data is written.

Before writing to program memory, the word(s) to be written must be erased or previously unwritten. Program memory can only be erased one row at a time. No automatic erase occurs upon the initiation of the write.

Program memory can be written one or more words at a time. The maximum number of words written at one time is equal to the number of write latches. See Figure 9-5 (row writes to program memory with 16 write latches) for more details.

The write latches are aligned to the Flash row address boundary defined by the upper 10-bits of PMADRH:PMADRL, (PMADRH<6:0>:PMADRL<7:5>) with the lower 5-bits of PMADRL, (PMADRL<4:0>) determining the write latch being loaded. Write operations do not cross these boundaries. At the completion of a program memory write operation, the data in the write latches is reset to contain 0x3FFF.

The following steps should be completed to load the write latches and program a row of program memory. These steps are divided into two parts. First, each write latch is loaded with data from the PMDATH:PMDATL using the unlock sequence with LWLO = `1`. When the last word to be loaded into the write latch is ready, the LWLO bit is cleared and the unlock sequence executed. This initiates the programming operation, writing all the latches into Flash program memory.

Note:	The special unlock sequence is required to load a write latch with data or initiate a Flash programming operation. If the unlock sequence is interrupted, writing to the latches or program memory will not be initiated.

1. Set the WREN bit of the PMCON1 register.
2. Clear the CFGS bit of the PMCON1 register.
3. Set the LWLO bit of the PMCON1 register. When the LWLO bit of the PMCON1 register is ‘1’, the write sequence will only load the write latches and will not initiate the write to Flash program memory.
4. Load the PMADRH:PMADRL register pair with the address of the location to be written.
5. Load the PMDATH:PMDATL register pair with the program memory data to be written.
6. Execute the unlock sequence (**Section 9.2.2 “Flash Memory Unlock Sequence”**). The write latch is now loaded.
7. Increment the PMADRH:PMADRL register pair to point to the next location.
8. Repeat steps 5 through 7 until all but the last write latch has been loaded.
9. Clear the LWLO bit of the PMCON1 register. When the LWLO bit of the PMCON1 register is ‘0’, the write sequence will initiate the write to Flash program memory.
10. Load the PMDATH:PMDATL register pair with the program memory data to be written.
11. Execute the unlock sequence (**Section 9.2.2 “Flash Memory Unlock Sequence”**). The entire program memory latch content is now written to Flash program memory.

Note:	The program memory write latches are reset to the blank state (0x3FFF) at the completion of every write or erase operation. As a result, it is not necessary to load all the program memory write latches. Unloaded latches will remain in the blank state.

An example of the complete write sequence is shown in Example 9-3. The initial address is loaded into the PMADRH:PMADRL register pair; the data is loaded using indirect addressing.

FIGURE 9-5: BLOCK WRITES TO FLASH PROGRAM MEMORY WITH 16 WRITE LATCHES

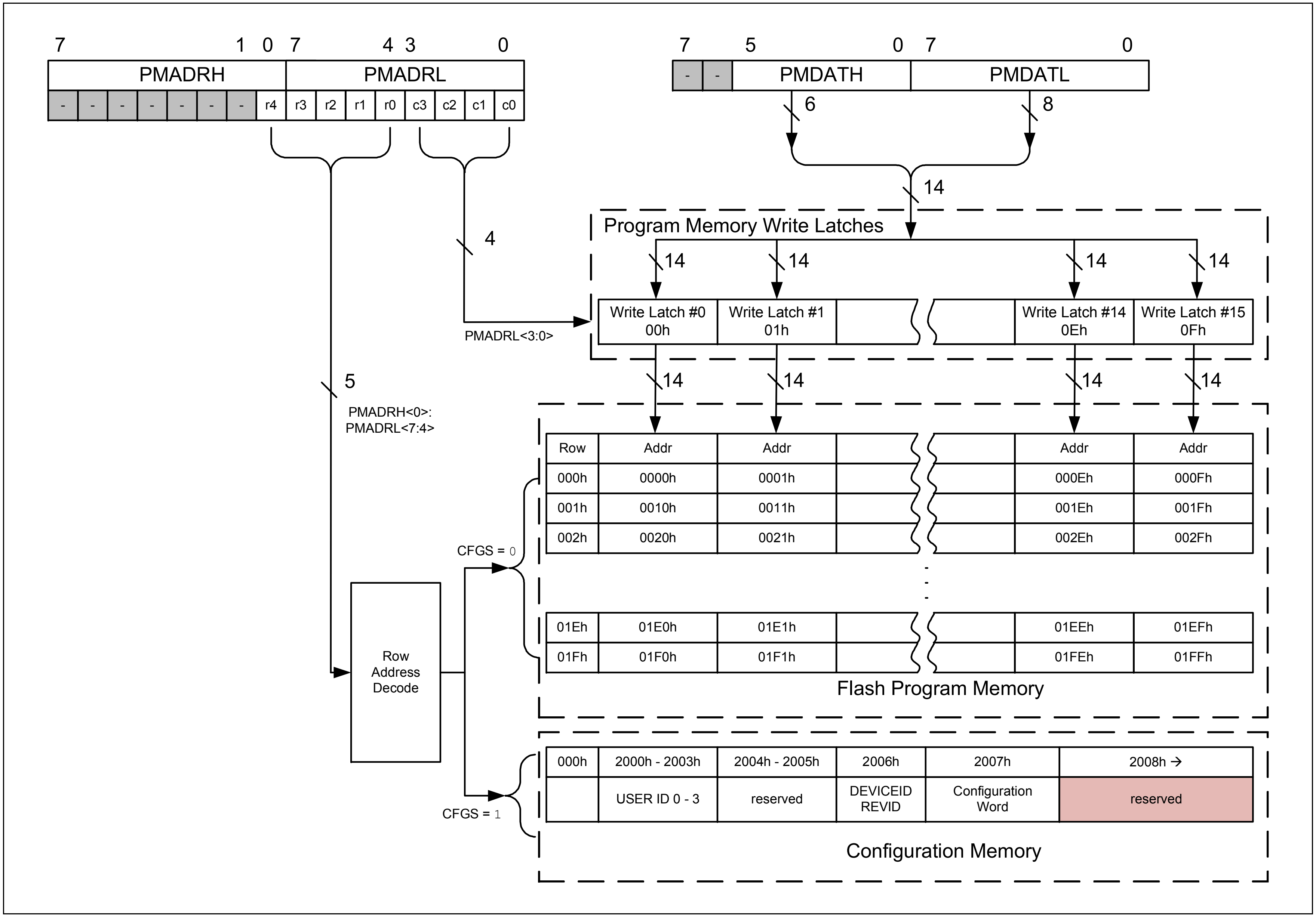
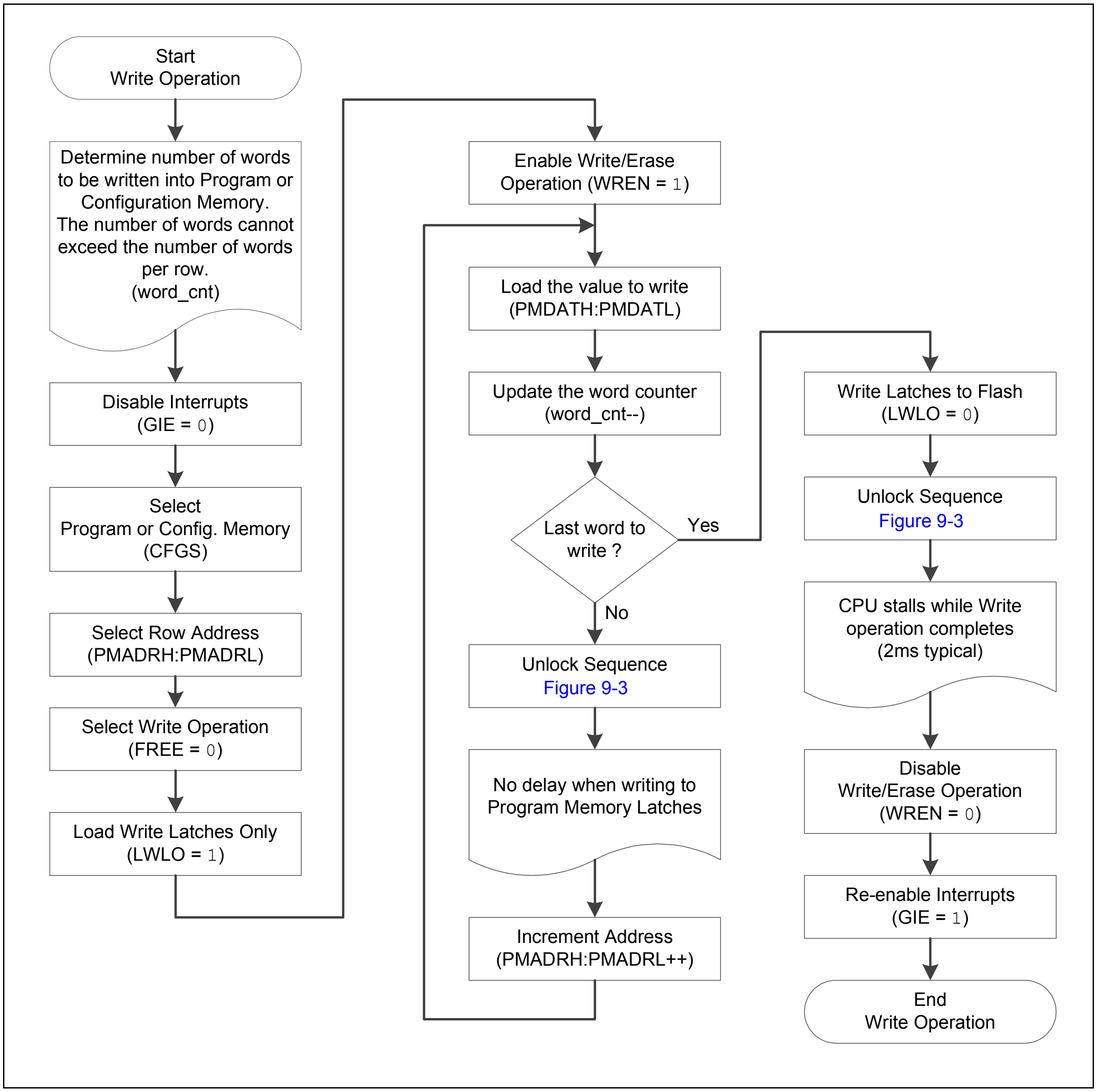


FIGURE 9-6: FLASH PROGRAM MEMORY WRITE FLOWCHART



EXAMPLE 9-3: WRITING TO FLASH PROGRAM MEMORY

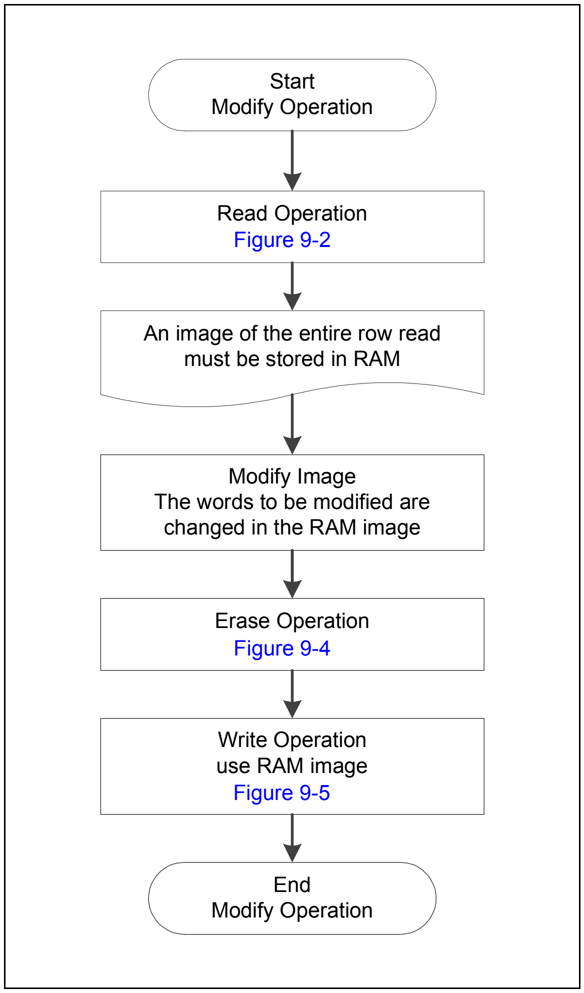
```
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ; This write routine assumes the following:
    ;        A valid starting address (the least significant bits = '00')
    ;        is loaded in ADDRH:ADDRL
    ;        ADDRH, ADDRL and DATADDR are all located in data memory
    ;
   BANKSEL         PMADRH
   MOVF    ADDRH,W      ;Load initial address
   MOVWF   PMADRH       ;
   MOVF    ADDRL,W      ;
   MOVWF   PMADRL       ;
   MOVF    DATAADDR,W   ;Load initial data address
   MOVWF   FSR          ;
LOOP MOVF  INDF,W       ;Load first data byte into lower
   MOVWF   PMDATL       ;
   INCF    FSR,F        ;Next byte
   MOVF    INDF,W       ;Load second data byte into upper
   MOVWF   PMDATH       ;
   INCF    FSR,F        ;
   BANKSEL PMCON1
   BSF     PMCON1,WREN  ;Enable writes
   BCF     INTCON,GIE   ;Disable interrupts (if using)
   BTFSC   INTCON,GIE   ;See AN576
   GOTO    $-2
   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
   ;       Required Sequence
   MOVLW   55h          ;Start of required write sequence:
   MOVWF   PMCON2       ;Write 55h
   MOVLW   0AAh         ;
   MOVWF   PMCON2       ;Write 0AAh
   BSF     PMCON1,WR    ;Set WR bit to begin write
   NOP                  ;Required to transfer data to the buffer
   NOP                  ;registers
   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
   BCF     PMCON1,WREN  ;Disable writes
   BSF     INTCON,GIE   ;Enable interrupts (comment out if not using interrupts)
   BANKSEL PMADRL
   MOVF    PMADRL, W
   INCF    PMADRL,F     ;Increment address
   ANDLW   0x03         ;Indicates when sixteen words have been programmed
   SUBLW   0x03         ;Change value for different size write blocks
                        ;0x0F = 16 words
                        ;0x0B = 12 words
                        ;0x07 = 8 words
                        ;0x03 = 4 words
   BTFSS   STATUS,Z     ;Exit on a match,
   GOTO    LOOP         ;Continue if more data needs to be written
```

9.3 Modifying Flash Program Memory

When modifying existing data in a program memory row, and data within that row must be preserved, it must first be read and saved in a RAM image. Program memory is modified using the following steps:

1. Load the starting address of the row to be modified.
2. Read the existing data from the row into a RAM image.
3. Modify the RAM image to contain the new data to be written into program memory.
4. Load the starting address of the row to be rewritten.
5. Erase the program memory row.
6. Load the write latches with data from the RAM image.
7. Initiate a programming operation.

FIGURE 9-7: FLASH PROGRAM MEMORY MODIFY FLOWCHART



9.4 User ID, Device ID and Configuration Word Access

Instead of accessing program memory, the User ID's, Device ID/Revision ID and Configuration Word can be accessed when CFGS = 1 in the PMCON1 register. This is the region that would be pointed to by PC<13> = 1, but not all addresses are accessible. Different access may exist for reads and writes. Refer to Table 9-2.

When read access is initiated on an address outside the parameters listed in Table 9-2, the PMDATH:PMDATL register pair is cleared, reading back '0's.

TABLE 9-2: USER ID, DEVICE ID AND CONFIGURATION WORD ACCESS (CFGS = 1)

Address	Function	Read Access	Write Access
2000h-2003h	User IDs	Yes	Yes
2006h	Device ID/Revision ID	Yes	No
2007h	Configuration Word	Yes	No

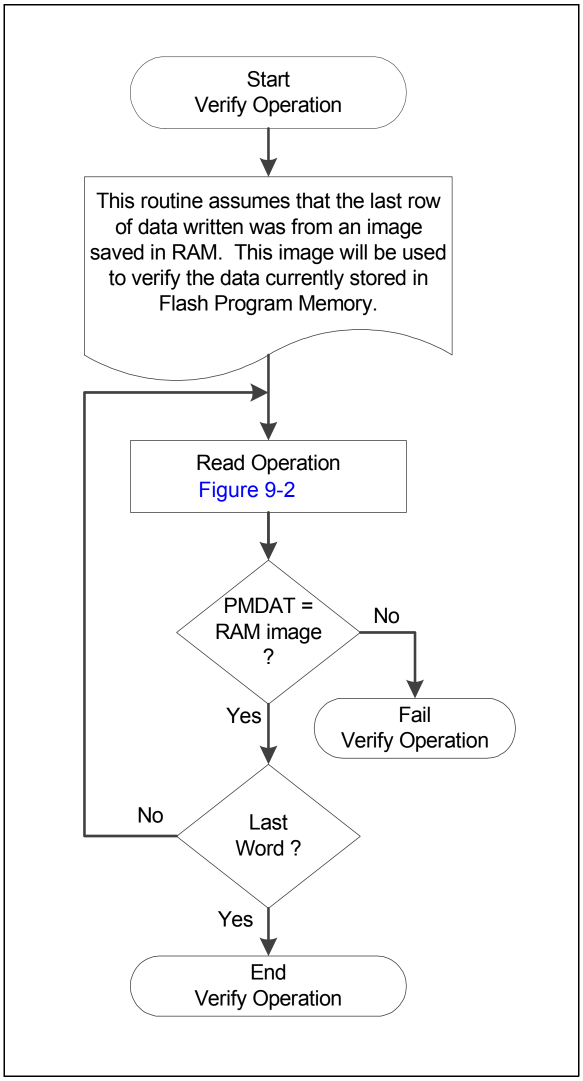
EXAMPLE 9-4: CONFIGURATION WORD AND DEVICE ID ACCESS

```
* This code block will read 1 word of program memory at the memory address:
*   PROG_ADDR_LO (must be 00h-08h) data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

    BANKSEL  PMADRL           ; not required on devices with 1 Bank of SFRs
    MOVLW    PROG_ADDR_LO     ;
    MOVWF    PMADRL           ; Store LSB of address
    CLRF     PMADRH           ; Clear MSB of address

    BSF      PMCON1,CFGS      ; Select Configuration Space
    BCF      INTCON,GIE       ; Disable interrupts
    BSF      PMCON1,RD        ; Initiate read
    NOP                       ; Executed (See Figure 9-2)
    NOP                       ; Ignored (See Figure 9-2)
    BSF      INTCON,GIE       ; Restore interrupts

    MOVF     PMDATL,W         ; Get LSB of word
    MOVWF    PROG_DATA_LO     ; Store in user location
    MOVF     PMDATH,W         ; Get MSB of word
    MOVWF    PROG_DATA_HI     ; Store in user location
```

9.5 Write Verify

It is considered good programming practice to verify that program memory writes agree with the intended value. Since program memory is stored as a full page then the stored program memory contents are compared with the intended data stored in RAM after the last write is complete.

FIGURE 9-8: FLASH PROGRAM MEMORY VERIFY FLOWCHART



9.6 Flash Program Memory Control Registers

REGISTER 9-1: PMDATL: PROGRAM MEMORY DATA LOW

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
PMDAT<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 **PMDAT<7:0>**: The value of the program memory word pointed to by PMADRH and PMADRL after a program memory read command.

REGISTER 9-2: PMDATH: PROGRAM MEMORY DATA HIGH

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	PMDAT<13:8>					
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 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **PMDAT<13:8>**: The value of the program memory word pointed to by PMADRH and PMADRL after a program memory read command.

REGISTER 9-3: PMADRL: PROGRAM MEMORY ADDRESS LOW

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 = 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 **PMADR<7:0>**: Program Memory Read Address low bits

REGISTER 9-4: PMADRH: PROGRAM MEMORY ADDRESS HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	PMADR8
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 **PMADR8:** Program Memory Read Address High bit

REGISTER 9-5: PMCON1: PROGRAM MEMORY CONTROL 1 REGISTER

U-1(1)	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HC-0/q(2)	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
—	CFGS	LWLO	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
S = Bit can only be set	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 **Unimplemented:** Read as ‘1’

bit 6 **CFGS:** Configuration Select bit
1 = Access Configuration, User ID and Device ID Registers
0 = Access Flash program memory

bit 5 **LWLO:** Load Write Latches Only bit(3)
1 = Only the addressed program memory write latch is loaded/updated on the next WR command
0 = The addressed program memory write latch is loaded/updated and a write of all program memory write latches will be initiated on the next WR command

bit 4 **FREE:** Program Flash Erase Enable bit
1 = Performs an erase operation on the next WR command (hardware cleared upon completion)
0 = Performs an write operation on the next WR command

bit 3 **WRERR:** Program/Erase Error Flag bit
1 = Condition indicates an improper program or erase sequence attempt or termination (bit is set automatically on any set attempt (write ‘1’) of the WR bit).
0 = The program or erase operation completed normally.

bit 2 **WREN:** Program/Erase Enable bit
1 = Allows program/erase cycles
0 = Inhibits programming/erasing of program Flash

bit 1 **WR:** Write Control bit
1 = Initiates a program Flash program/erase operation.
The operation is self-timed and the bit is cleared by hardware once operation is complete.
The WR bit can only be set (not cleared) in software.
0 = Program/erase operation to the Flash is complete and inactive.

bit 0 **RD:** Read Control bit
1 = Initiates a program Flash read. Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software.
0 = Does not initiate a program Flash read.

Note 1: Unimplemented bit, read as ‘1’.
2: The WRERR bit is automatically set by hardware when a program memory write or erase operation is started (WR = 1).
3: The LWLO bit is ignored during a program memory erase operation (FREE = 1).

REGISTER 9-6: PMCON2: PROGRAM MEMORY CONTROL 2 REGISTER

W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0
Program Memory Control Register 2							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
S = Bit can only be set	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 **Flash Memory Unlock Pattern bits**
To unlock writes, a 55h must be written first, followed by an AAh, before setting the WR bit of the PMCON1 register. The value written to this register is used to unlock the writes. There are specific timing requirements on these writes.

TABLE 9-3: SUMMARY OF REGISTERS ASSOCIATED WITH FLASH PROGRAM MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
PMCON1	—	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	65
PMCON2	Program Memory Control Register 2								66
PMADRL	PMADR<7:0>								64
PMADRH	—	—	—	—	—	—	—	PMADR8	64
PMDATL	PMDAT<7:0>								63
PMDATH	—	—	PMDAT<13:8>						63

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used by Flash program memory module.

TABLE 9-4: SUMMARY OF CONFIGURATION WORD WITH FLASH PROGRAM MEMORY

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG	13:8	—	—	—	WRT<1:0>		BORV	LPBOR	LVP	20
	7:0	$\overline{\text{CP}}$	$\overline{\text{MCLR}}$	$\overline{\text{PWRTE}}$	WDTE<1:0>		BOREN<1:0>		FOSC	

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used by Flash program memory.

10.0 I/O PORT

Depending on which peripherals are enabled, some or all of the pins may not be available as general purpose I/O. In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output. However, the pin can still be read.

PORTA has three standard registers for its operation. These registers are:

- TRISA register (data direction)
- PORTA register (reads the levels on the pins of the device)
- LATA register (output latch)

Some ports may have one or more of the following additional registers. These registers are:

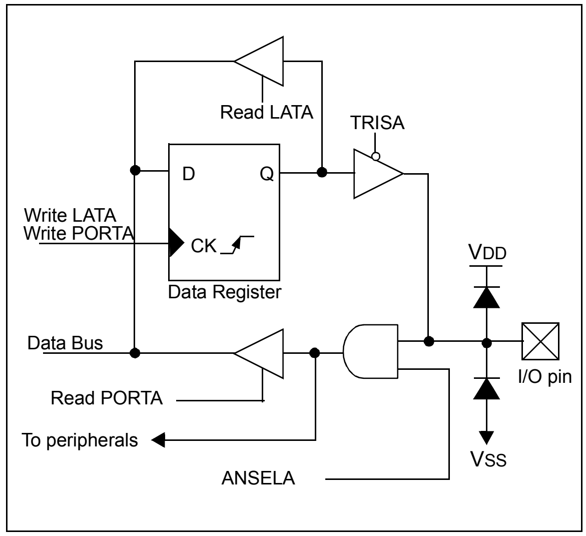
- ANSELA (analog select)
- WPUA (weak pull-up)

The Data Latch (LATA register) is useful for read-modify-write operations on the value that the I/O pins are driving.

A write operation to the LATA register has the same effect as a write to the corresponding PORTA register. A read of the LATA register reads of the values held in the I/O PORT latches, while a read of the PORTA register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELA register. When an ANSEL bit is set, the digital input buffer associated with that bit is disabled. Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 10-1.

FIGURE 10-1: I/O PORT OPERATION



EXAMPLE 10-1: INITIALIZING PORTA

```
; This code example illustrates
; initializing the PORTA register. The
; other ports are initialized in the same
; manner.

BANKSEL    PORTA           ;not required on devices with 1 Bank of SFRs
CLRF       PORTA           ;Init PORTA
BANKSEL    LATA            ;not required on devices with 1 Bank of SFRs
CLRF       LATA            ;
BANKSEL    ANSELA          ;not required on devices with 1 Bank of SFRs
CLRF       ANSELA          ;digital I/O
BANKSEL    TRISA           ;not required on devices with 1 Bank of SFRs
MOVLW      B'00000011'     ;Set RA<1:0> as inputs
MOVWF      TRISA           ;and set RA<2:3> as
                           ;outputs
```


10.1 PORTA Registers

PORTA is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 10-2). Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., disable the output driver). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., enables output driver and puts the contents of the output latch on the selected pin). Example 10-1 shows how to initialize PORTA.

Reading the PORTA register (Register 10-1) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATA).

The TRISA register (Register 10-2) controls the PORTA pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISA register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

10.1.1 WEAK PULL-UPS

Each of the PORTA pins has an individually configurable internal weak pull-up. Control bits WPUA<3:0> enable or disable each pull-up (see Register 10-5). Each weak pull-up is automatically turned off when the port pin is configured as an output. All pull-ups are disabled on a Power-on Reset by the $\overline{\text{WPUEN}}$ bit of the OPTION_REG register.

10.1.2 ANSELA REGISTER

The ANSELA register (Register 10-4) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELA bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELA bits has no effect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

Note:	The ANSELA bits default to the Analog mode after Reset. To use any pins as digital general purpose or peripheral inputs, the corresponding ANSEL bits must be initialized to '0' by user software.

10.1.3 PORTA FUNCTIONS AND OUTPUT PRIORITIES

Each PORTA pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 10-1.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Digital output functions may control the pin when it is in Analog mode with the priority shown in Table 10-1.

TABLE 10-1: PORTA OUTPUT PRIORITY

Pin Name	Function Priority(1)
RA0	ICSPDAT CWG1A PWM1 RA0
RA1	CWG1B PWM2 CLC1 RA1
RA2	NCO1 CLKR RA2
RA3	None

Note 1: Priority listed from highest to lowest.

10.2 Register Definitions: PORTA

REGISTER 10-1: PORTA: PORTA REGISTER

U-0	U-0	U-0	U-0	R-x/x	R/W-x/x	R/W-x/x	R/W-x/x
—	—	—	—	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 7-4 **Unimplemented**: Read as ‘0’

bit 3-0 **RA<3:0>:** PORTA I/O Value bits (RA3 is read-only)

Note 1: Writes to PORTx are actually written to the corresponding LATx register. Reads from PORTx register return actual I/O pin values.

REGISTER 10-2: TRISA: PORTA TRI-STATE REGISTER

U-0	U-0	U-0	U-0	U-1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	—(1)	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 7-4 **Unimplemented**: Read as ‘0’

bit 3 **Unimplemented**: Read as ‘1’

bit 2-0 **TRISA<2:0>:** RA<2:0> Port I/O Tri-State Control bits
1 = Port output driver is disabled
0 = Port output driver is enabled

Note 1: Unimplemented, read as ‘1’.

REGISTER 10-3: LATA: PORTA DATA LATCH REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	—	LATA2	LATA1	LATA0
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 7-3 **Unimplemented**: Read as ‘0’

bit 2-0 **LATA<2:0>:** RA<2:0> Output Latch Value bits

Note 1: Writes to PORTx are actually written to the corresponding LATx register. Reads from LATx register return register values, not I/O pin values.

REGISTER 10-4: ANSELA: PORTA ANALOG SELECT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	—	ANSA2	ANSA1	ANSA0
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 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **ANSA<2:0>**: Analog Select between Analog or Digital Function on Pins RA<2:0>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital Input buffer disabled.
0 = Digital I/O. Pin is assigned to port or Digital special function.

Note 1: Setting a pin to an analog input automatically disables the digital input circuitry. Weak pull-ups, if available, are unaffected. The corresponding TRIS bit must be set to Input mode by the user in order to allow external control of the voltage on the pin.

REGISTER 10-5: WPUA: WEAK PULL-UP PORTA REGISTER

U-0	U-0	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	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 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **WPUA<3:0>:** Weak Pull-up PORTA Control bits
1 = Weak Pull-up enabled(1)
0 = Weak Pull-up disabled.

Note 1: Enabling weak pull-ups also requires that the $\overline{\text{WPUEN}}$ bit of the OPTION_REG register be cleared (Register 16-1).

TABLE 10-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	—	—	ANSA2	ANSA1	ANSA0	70
IOCAF	—	—	—	—	IOCAF3	IOCAF2	IOCAF1	IOCAF0	76
IOCAN	—	—	—	—	IOCAN3	IOCAN2	IOCAN1	IOCAN0	75
IOCAP	—	—	—	—	IOCAP3	IOCAP2	IOCAP1	IOCAP0	75
LATA	—	—	—	—	—	LATA2	LATA1	LATA0	70
PORTA	—	—	—	—	RA3	RA2	RA1	RA0	69
TRISA	—	—	—	—	—(1)	TRISA2	TRISA1	TRISA0	69
WPUA	—	—	—	—	WPUA3	WPUA2	WPUA1	WPUA0	71

Legend: x = unknown, u = unchanged, – = unimplemented locations read as ‘0’. Shaded cells are not used by PORTA.

Note 1: Unimplemented, read as ‘1’.

11.0 INTERRUPT-ON-CHANGE

The PORTA pins can be configured to operate as Interrupt-On-Change (IOC) pins. An interrupt can be generated by detecting a signal that has either a rising edge or a falling edge. Any individual PORTA pin, or combination of PORTA pins, can be configured to generate an interrupt. The Interrupt-on-change module has the following features:

- Interrupt-on-Change enable (Master Switch)
- Individual pin configuration
- Rising and falling edge detection
- Individual pin interrupt flags

Figure 11-1 is a block diagram of the IOC module.

11.1 Enabling the Module

To allow individual PORTA pins to generate an interrupt, the IOCIE bit of the INTCON register must be set. If the IOCIE bit is disabled, the edge detection on the pin will still occur, but an interrupt will not be generated.

11.2 Individual Pin Configuration

For each PORTA pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated IOCAPx bit of the IOCAP register is set. To enable a pin to detect a falling edge, the associated IOCANx bit of the IOCAN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting both the IOCAPx bit and the IOCANx bit of the IOCAP and IOCAN registers, respectively.

11.3 Interrupt Flags

The IOCAFx bits located in the IOCAF register are status flags that correspond to the Interrupt-on-change pins of PORTA. If an expected edge is detected on an appropriately enabled pin, then the status flag for that pin will be set, and an interrupt will be generated if the IOCIE bit is set. The IOCIF bit of the INTCON register reflects the status of all IOCAFx bits.

11.4 Clearing Interrupt Flags

The individual status flags, (IOCAFx bits), can be cleared by resetting them to zero. If another edge is detected during this clearing operation, the associated status flag will be set at the end of the sequence, regardless of the value actually being written.

In order to ensure that no detected edge is lost while clearing flags, only AND operations masking out known changed bits should be performed. The following sequence is an example of what should be performed.

EXAMPLE 11-1:

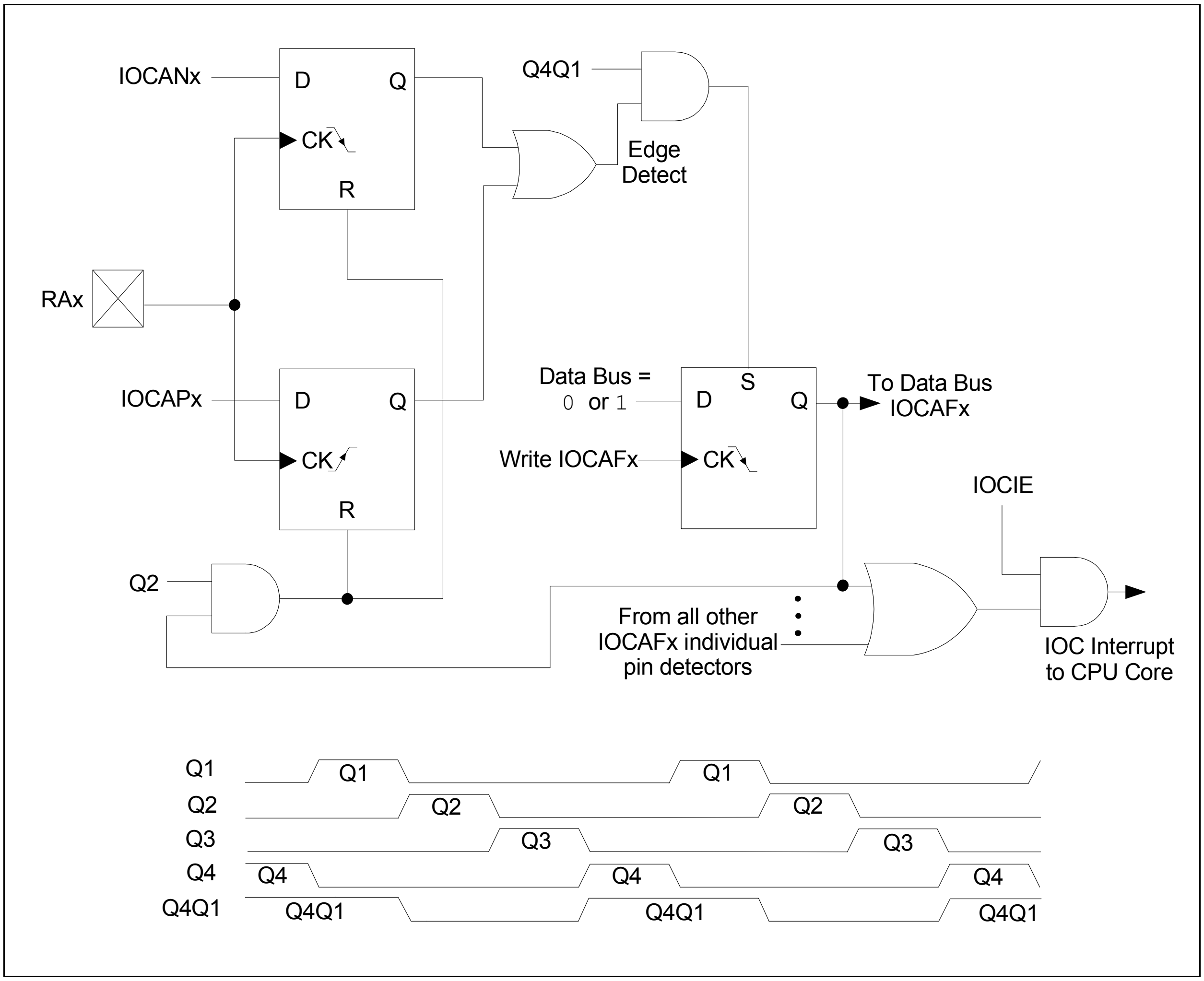
```
MOVLW   0xff
XORWF   IOCAF, W
ANDWF   IOCAF, F
```

11.5 Operation in Sleep

The interrupt-on-change interrupt sequence will wake the device from Sleep mode, if the IOCIE bit is set.

If an edge is detected while in Sleep mode, the IOCAF register will be updated prior to the first instruction executed out of Sleep.

FIGURE 11-1: INTERRUPT-ON-CHANGE BLOCK DIAGRAM



11.6 Interrupt-On-Change Registers

REGISTER 11-1: IOCAP: INTERRUPT-ON-CHANGE PORTA POSITIVE EDGE REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	IOCAP3	IOCAP2	IOCAP1	IOCAP0
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 **IOCAP<3:0>:** Interrupt-on-change PORTA Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.(1)
0 = Interrupt-on-Change disabled for the associated pin.

Note 1: Interrupt-on-change also requires that the IOCIE bit of the INTCON register be set (Register 6-1).

REGISTER 11-2: IOCAN: INTERRUPT-ON-CHANGE PORTA NEGATIVE EDGE REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	IOCAN3	IOCAN2	IOCAN1	IOCAN0
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 **IOCAN<3:0>:** Interrupt-on-change PORTA Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.(1)
0 = Interrupt-on-Change disabled for the associated pin.

Note 1: Interrupt-on-change also requires that the IOCIE bit of the INTCON register be set (Register 6-1).

REGISTER 11-3: IOCAF: INTERRUPT-ON-CHANGE PORTA FLAG REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	IOCAF3	IOCAF2	IOCAF1	IOCAF0
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 - Bit is set in hardware

bit 7-4 **Unimplemented**: Read as ‘0’.

bit 3-0 **IOCAF<3:0>:** Interrupt-on-change PORTA Flag bits

1 = An enable change was detected on the associated pin.
Set when IOCAPx = 1 and a rising edge was detected on RAx, or when IOCANx = 1 and a falling edge was detected on RAx.(1)

0 = No change was detected, or the user cleared the detected change.

Note 1: Interrupt-on-change also requires that the IOCIE bit of the INTCON register be set (Register 6-1).

TABLE 11-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPT-ON-CHANGE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
IOCAF	—	—	—	—	IOCAF3	IOCAF2	IOCAF1	IOCAF0	76
IOCAN	—	—	—	—	IOCAN3	IOCAN2	IOCAN1	IOCAN0	75
IOCAP	—	—	—	—	IOCAP3	IOCAP2	IOCAP1	IOCAP0	75
TRISA	—	—	—	—	—(1)	TRISA2	TRISA1	TRISA0	69

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used by Interrupt-on-Change.

Note 1: Unimplemented, read as ‘1’.

12.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference, or FVR, is a stable voltage reference, independent of VDD, with 1.024V, 2.048V or 4.096V selectable output levels. The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel

The FVR can be enabled by setting the FVREN bit of the FVRCON register.

12.1 Independent Gain Amplifiers

The output of the FVR supplied to the ADC is routed through an independent programmable gain amplifier. The amplifier can be configured to amplify the reference voltage by 1x, 2x or 4x, to produce the three possible voltage levels.

The ADFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the ADC module. Reference **Section 15.0 "Analog-to-Digital Converter (ADC) Module"** for additional information.

To minimize current consumption when the FVR is disabled, the FVR buffers should be turned off by clearing the ADFVR<1:0> bits.

12.2 FVR Stabilization Period

When the Fixed Voltage Reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. Once the circuits stabilize and are ready for use, the FVRRDY bit of the FVRCON register will be set. See **Section 24.0 "Electrical Specifications"** for the minimum delay requirement.

FIGURE 12-1: VOLTAGE REFERENCE BLOCK DIAGRAM

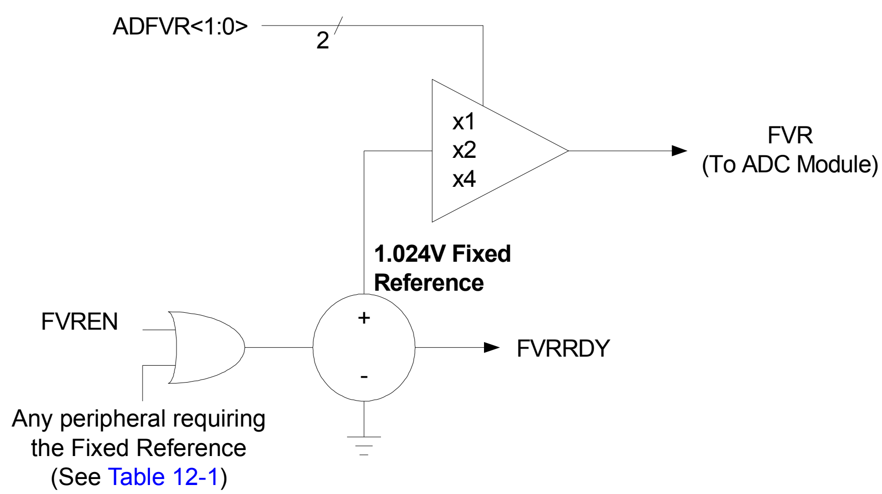


TABLE 12-1: PERIPHERALS REQUIRING THE FIXED VOLTAGE REFERENCE (FVR)

Peripheral	Conditions	Description
HFINTOSC	FOSC = 1	EC on CLKIN pin.
BOR	BOREN<1:0> = 11	BOR always enabled.
	BOREN<1:0> = 10 and BORFS = 1	BOR disabled in Sleep mode, BOR Fast Start enabled.
	BOREN<1:0> = 01 and BORFS = 1	BOR under software control, BOR Fast Start enabled.
IVR	All PIC10F320/322 devices, when VREGPM1 = 1 and not in Sleep	The device runs off of the Power-Save mode regulator when in Sleep mode.

12.3 FVR Control Registers

REGISTER 12-1: FVRCON: FIXED VOLTAGE REFERENCE CONTROL REGISTER

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
FVREN	FVRRDY(1)	TSEN(3)	TSRNG(3)	—	—	ADFVR<1:0>	
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	q = Value depends on condition

bit 7 **FVREN:** Fixed Voltage Reference Enable bit
1 = Fixed Voltage Reference is enabled
0 = Fixed Voltage Reference is disabled

bit 6 **FVRRDY:** Fixed Voltage Reference Ready Flag bit(1)
1 = Fixed Voltage Reference output is ready for use
0 = Fixed Voltage Reference output is not ready or not enabled

bit 5 **TSEN:** Temperature Indicator Enable bit(3)
1 = Temperature Indicator is enabled
0 = Temperature Indicator is disabled

bit 4 **TSRNG:** Temperature Indicator Range Selection bit(3)
1 = $V_{OUT} = V_{DD} - 4V_T$ (High Range)
0 = $V_{OUT} = V_{DD} - 2V_T$ (Low Range)

bit 3-2 **Unimplemented**: Read as ‘0 ‘

bit 1-0 **ADFVR<1:0>:** ADC Fixed Voltage Reference Selection bit
11 = ADC Fixed Voltage Reference Peripheral output is 4x (4.096V)(2)
10 = ADC Fixed Voltage Reference Peripheral output is 2x (2.048V)(2)
01 = ADC Fixed Voltage Reference Peripheral output is 1x (1.024V)
00 = ADC Fixed Voltage Reference Peripheral output is off.

Note 1: FVRRDY indicates the true state of the FVR.
2: Fixed Voltage Reference output cannot exceed V_{DD}.
3: See **Section 14.0 “Temperature Indicator Module”** for additional information.

TABLE 12-2: SUMMARY OF REGISTERS ASSOCIATED WITH FIXED VOLTAGE REFERENCE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	—	—	ADFVR<1:0>		78

Legend: Shaded cells are not used with the Fixed Voltage Reference.

13.0 INTERNAL VOLTAGE REGULATOR (IVR)

The Internal Voltage Regulator (IVR), which provides operation above 3.6V is available on:

- PIC10F320
- PIC10F322

This circuit regulates a voltage for the internal device logic while permitting the VDD and I/O pins to operate at a higher voltage. When VDD approaches the regulated voltage, the IVR output automatically tracks the input voltage.

The IVR operates in one of three power modes based on user configuration and peripheral selection. The operating power modes are:

- High
- Low
- Power Save Sleep mode

Power modes are selected automatically depending on the device operation, as shown in Table 13-1. Tracking mode is selected automatically when VDD drops below the safe operating voltage of the core.

Note: IVR is disabled in Tracking mode, but will consume power. See **Section 24.0 "Electrical Specifications"** for more information.

TABLE 13-1: IVR POWER MODES - REGULATED

VREGPM1 Bit	Sleep Mode	Memory Bias Power Mode	IVR Power Mode
x	No	EC Mode or INTOSC = 16 MHz (HP Bias)	High
		INTOSC = 1 to 8 MHz (MP Bias)	
		INTOSC = 31 kHz to 500 kHz (LP Bias)	Low
0	Yes	Don't Care	Low
1	Yes	No HFINTOSC No Peripherals	Power Save(1)

Note 1: Forced to Low-Power mode by any of the following conditions:

- BOR is enabled
- HFINTOSC is an active peripheral source
- Self-write is active
- ADC is in an active conversion

REGISTER 13-1: VREGCON: VOLTAGE REGULATOR CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-1/1
—	—	—	—	—	—	VREGPM1	Reserved
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 7-2 **Unimplemented:** Read as ‘0’

bit 1 **VREGPM1:** Voltage Regulator Power Mode Selection bit
1 = Power-Save Sleep mode enabled in Sleep. Draws lowest current in Sleep, slower wake-up.
0 = Low-Power mode enabled in Sleep. Draws higher current in Sleep, faster wake-up.

bit 0 **Reserved:** Maintain this bit set.

14.0 TEMPERATURE INDICATOR MODULE

This family of devices is equipped with a temperature circuit designed to measure the operating temperature of the silicon die. The circuit's range of operating temperature falls between of -40°C and +85°C. The output is a voltage that is proportional to the device temperature. The output of the temperature indicator is internally connected to the device ADC.

The circuit may be used as a temperature threshold detector or a more accurate temperature indicator, depending on the level of calibration performed. A one-point calibration allows the circuit to indicate a temperature closely surrounding that point. A two-point calibration allows the circuit to sense the entire range of temperature more accurately. Reference Application Note AN1333, "*Use and Calibration of the Internal Temperature Indicator*" (DS01333) for more details regarding the calibration process.

14.1 Circuit Operation

Figure 14-1 shows a simplified block diagram of the temperature circuit. The proportional voltage output is achieved by measuring the forward voltage drop across multiple silicon junctions.

Equation 14-1 describes the output characteristics of the temperature indicator.

EQUATION 14-1: VOUT RANGES

High Range: VOUT = VDD - 4VT

Low Range: VOUT = VDD - 2VT

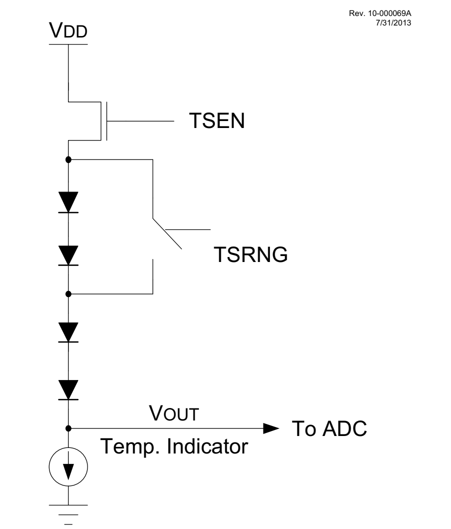
The temperature sense circuit is integrated with the Fixed Voltage Reference (FVR) module. See **Section 12.0 "Fixed Voltage Reference (FVR)"** for more information.

The circuit is enabled by setting the TSEN bit of the FVRCON register. When disabled, the circuit draws no current.

The circuit operates in either high or low range. The high range, selected by setting the TSRNG bit of the FVRCON register, provides a wider output voltage. This provides more resolution over the temperature range, but may be less consistent from part to part. This range requires a higher bias voltage to operate and thus, a higher VDD is needed.

The low range is selected by clearing the TSRNG bit of the FVRCON0 register. The low range generates a lower voltage drop and thus, a lower bias voltage is needed to operate the circuit. The low range is provided for low voltage operation.

FIGURE 14-1: TEMPERATURE CIRCUIT DIAGRAM



14.2 Minimum Operating VDD vs. Minimum Sensing Temperature

When the temperature circuit is operated in low range, the device may be operated at any operating voltage that is within specifications.

When the temperature circuit is operated in high range, the device operating voltage, VDD, must be high enough to ensure that the temperature circuit is correctly biased.

Table 14-1 shows the recommended minimum VDD vs. range setting.

TABLE 14-1: RECOMMENDED VDD VS. RANGE

Min. VDD, TSRNG = 1	Min. VDD, TSRNG = 0
3.6V	1.8V

14.3 Temperature Output

The output of the circuit is measured using the internal Analog-to-Digital Converter. A channel is reserved for the temperature circuit output. Refer to **Section 15.0 "Analog-to-Digital Converter (ADC) Module"** for detailed information.

14.4 ADC Acquisition Time

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

TABLE 14-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE TEMPERATURE INDICATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	—	—	ADFVR<1:0>		78
ADCON	ADCS<2:0>			CHS<2:0>			GO/$\overline{\text{DONE}}$	ADON	88
ADRES	A/D Result Register								89

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the temperature indicator module.

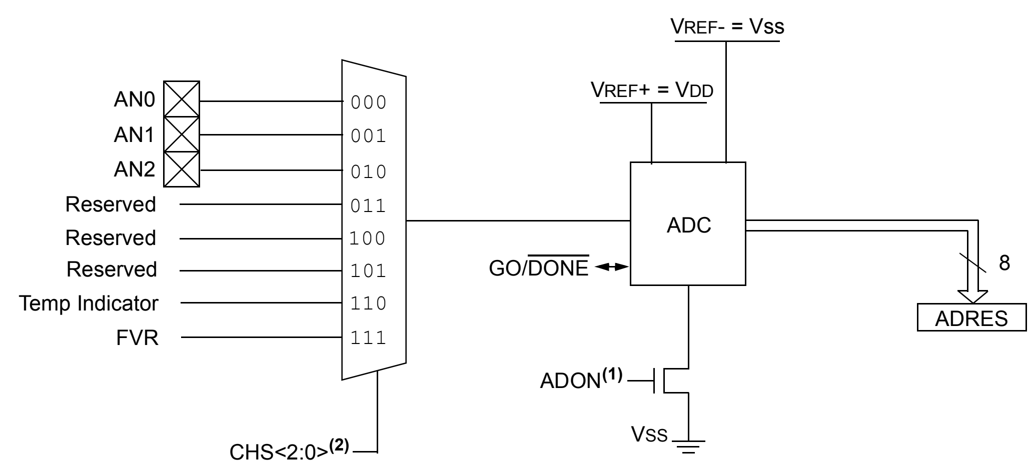
15.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

The Analog-to-Digital Converter (ADC) converts an analog input signal to an 8-bit binary representation of that signal. This device uses three analog input channels, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates an 8-bit binary result via successive approximation and stores the conversion result into the ADC result register (ADRES). Figure 15-1 shows the block diagram of the ADC.

The ADC voltage reference is software selectable to be internally generated.

The ADC can generate an interrupt upon completion of a conversion. This interrupt can be used to wake-up the device from Sleep.

FIGURE 15-1: ADC SIMPLIFIED BLOCK DIAGRAM



Note 1: When ADON = 0, all multiplexer inputs are disconnected.

2: See ADCON register (Register 15-1) for detailed analog channel selection per device.

15.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- Channel selection
- ADC conversion clock source
- Interrupt control

15.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 10.0 "I/O Port"** for more information.

Note:	Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

15.1.2 CHANNEL SELECTION

There are up to 5 channel selections available:

- AN<2:0> pins
- Temperature Indicator
- FVR (Fixed Voltage Reference) Output

Refer to **Section 12.0 "Fixed Voltage Reference (FVR)"** and **Section 14.0 "Temperature Indicator Module"** for more information on these channel selections.

The CHS bits of the ADCON register determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 15.2 "ADC Operation"** for more information.

15.1.3 ADC VOLTAGE REFERENCE

There is no external voltage reference connections to the ADC. Only VDD can be used as a reference source. The FVR is only available as an input channel and not a VREF+ input to the ADC.

15.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS bits of the ADCON register (Register 15-1). There are seven possible clock options:

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC (dedicated internal RC oscillator)

The time to complete one bit conversion is defined as TAD. One full 8-bit conversion requires 9.5 TAD periods as shown in Figure 15-2.

For correct conversion, the appropriate TAD specification must be met. Refer to the A/D conversion requirements in **Section 24.0 "Electrical Specifications"** for more information. Table 15-1 gives examples of appropriate ADC clock selections.

Note:	Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.

TABLE 15-1: ADC CLOCK PERIOD (TAD) VS. DEVICE OPERATING FREQUENCIES

ADC Clock Period (TAD)		Device Frequency (FOSC)			
ADC Clock Source	ADCS<2:0>	16 MHz	8 MHz	4 MHz	1 MHz
FOSC/2	000	125 ns(1)	250 ns(1)	500 ns(1)	2.0 μs
FOSC/4	100	250 ns(1)	500 ns(1)	1.0 μs	4.0 μs
FOSC/8	001	0.5 μs(1)	1.0 μs	2.0 μs	8.0 μs(2)
FOSC/16	101	1.0 μs	2.0 μs	4.0 μs	16.0 μs(2)
FOSC/32	010	2.0 μs	4.0 μs	8.0 μs(2)	32.0 μs(2)
FOSC/64	110	4.0 μs	8.0 μs(2)	16.0 μs(2)	64.0 μs(2)
FRC	x11	1.0-6.0 μs(1,3)	1.0-6.0 μs(1,3)	1.0-6.0 μs(1,3)	1.0-6.0 μs(1,3)

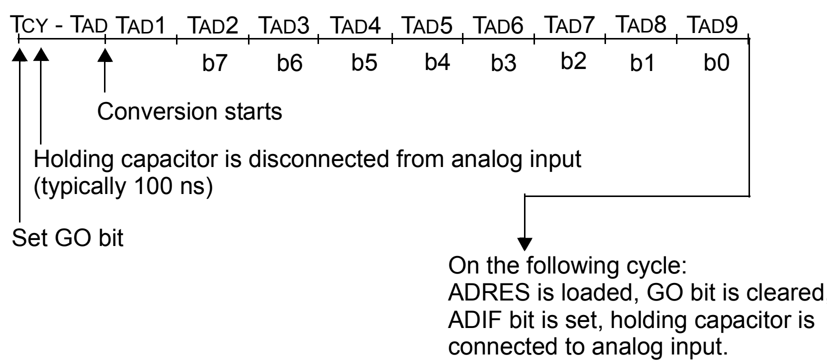
Legend: Shaded cells are outside of recommended range.

Note 1: These values violate the minimum required TAD time.

2: For faster conversion times, the selection of another clock source is recommended.

3: 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 clock source must be used when conversions are to be performed with the device in Sleep mode.

FIGURE 15-2: ANALOG-TO-DIGITAL CONVERSION TAD CYCLES



15.1.5 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital conversion. The ADC Interrupt Flag is the ADIF bit in the PIR1 register. The ADC Interrupt Enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared in software.

Note:	The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled.

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the `SLEEP` instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the GIE and PEIE bits of the INTCON register must be disabled. If the GIE and PEIE bits of the INTCON register are enabled, execution will switch to the Interrupt Service Routine.

15.2 ADC Operation

15.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON register must be set to a '1'. Setting the GO/$\overline{\text{DONE}}$ bit of the ADCON register to a '1' will start the Analog-to-Digital conversion.

Note:	The GO/$\overline{\text{DONE}}$ bit should not be set in the same instruction that turns on the ADC. Refer to **Section 15.2.5 "A/D Conversion Procedure"**.

15.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/$\overline{\text{DONE}}$ bit
- Set the ADIF Interrupt Flag bit
- Update the ADRES register with new conversion result

15.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/$\overline{\text{DONE}}$ bit can be cleared in software. The ADRES register will be updated with the partially complete Analog-to-Digital conversion sample. Incomplete bits will match the last bit converted.

Note:	A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated.

15.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. When the FRC clock source is selected, the ADC waits one additional instruction before starting the conversion. This allows the `SLEEP` instruction to be executed, which can reduce system noise during the conversion. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the ADON bit remains set.

When the ADC clock source is something other than FRC, a `SLEEP` instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

15.2.5 A/D CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

1. Configure Port:
   - Disable pin output driver (Refer to the TRIS register)
   - Configure pin as analog (Refer to the ANSEL register)
2. Configure the ADC module:
   - Select ADC conversion clock
   - Select ADC input channel
   - Turn on ADC module
3. Configure ADC interrupt (optional):
   - Clear ADC interrupt flag
   - Enable ADC interrupt
   - Enable peripheral interrupt
   - Enable global interrupt[(1)]
4. Wait the required acquisition time[(2)].
5. Start conversion by setting the GO/$\overline{\text{DONE}}$ bit.
6. Wait for ADC conversion to complete by one of the following:
   - Polling the GO/$\overline{\text{DONE}}$ bit
   - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result.
8. Clear the ADC interrupt flag (required if interrupt is enabled).

Note 1:	The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.
2:	Refer to **Section 15.4 "A/D Acquisition Requirements"**.

15.3 ADC Register Definitions

The following registers are used to control the operation of the ADC.

REGISTER 15-1: ADCON: A/D CONTROL REGISTER 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	R/W-0/0
ADCS<2:0>			CHS<2:0>			GO/$\overline{\text{DONE}}$	ADON
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 **ADCS<2:0>:** A/D Conversion Clock Select bits
111 = FRC
110 = FOSC/64
101 = FOSC/16
100 = FOSC/4
011 = FRC
010 = FOSC/32
001 = FOSC/8
000 = FOSC/2

bit 4-2 **CHS<2:0>:** Analog Channel Select bits
111 = FVR (Fixed Voltage Reference) Buffer Output(2)
110 = Temperature Indicator(1)
101 = Reserved. No channel connected.
100 = Reserved. No channel connected.
011 = Reserved. No channel connected.
010 = AN2
001 = AN1
000 = AN0

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
If ADON = 1:
1 = A/D conversion in progress (Setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (This bit is automatically cleared by hardware when the A/D conversion is complete.)

If this bit is cleared while a conversion is in progress, the conversion will stop and the results of the conversion up to this point will be transferred to the result registers, but the ADIF interrupt flag bit will not be set.

If ADON = 0:
0 = A/D conversion not in progress

bit 0 **ADON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

Note 1: See **Section 14.0 "Temperature Indicator Module"** for more information.
2: See **Section 12.0 "Fixed Voltage Reference (FVR)"** for more information.

REGISTER 15-2: ADRES: ADC RESULT 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
ADRES<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 **ADRES<7:0>**: ADC Result Register bits
8-bit result

15.4 A/D Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 15-3. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), refer to Figure 15-3. **The maximum recommended impedance for analog sources is 10 kΩ.** As the source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an A/D acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 15-1 may be used. This equation assumes that 1/2 LSb error is used (511 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

EQUATION 15-1: ACQUISITION TIME EXAMPLE

Assumptions: $Temperature = 50°C$ *and external impedance of* $10k\Omega$ $5.0V$ V_{DD}

$$
\begin{aligned}
T_{ACQ} &= \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \\
&= T_{AMP} + T_C + T_{COFF} \\
&= 2\mu s + T_C + [(Temperature - 25°C)(0.05\mu s/°C)]
\end{aligned}
$$

The value for T_C *can be approximated with the following equations:*

$$V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) = V_{CHOLD} \qquad \text{;[1] } V_{CHOLD} \text{ charged to within 1/2 lsb}$$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{CHOLD} \qquad \text{;[2] } V_{CHOLD} \text{ charge response to } V_{APPLIED}$$

$$V_{APPLIED}\left(1 - e^{\frac{-Tc}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) \qquad \text{;combining [1] and [2]}$$

Note: Where n = number of bits of the ADC.

Solving for T_C:

$$
\begin{aligned}
T_C &= -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ ln(1/511) \\
&= -10pF(1k\Omega + 7k\Omega + 10k\Omega)\ ln(0.001957) \\
&= 1.12\mu s
\end{aligned}
$$

Therefore:

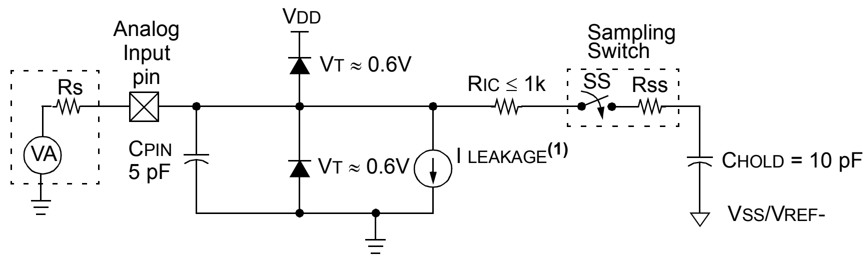
$$
\begin{aligned}
T_{ACQ} &= 2\mu s + 1.12\mu s + [(50°C - 25°C)(0.05\mu s/°C)] \\
&= 4.37\mu s
\end{aligned}
$$

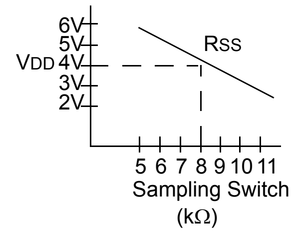
Note 1: The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

2: The charge holding capacitor (CHOLD) is not discharged after each conversion.

3: The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

FIGURE 15-3: 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 **Section 24.0 "Electrical Specifications"**.

FIGURE 15-4: ADC TRANSFER FUNCTION

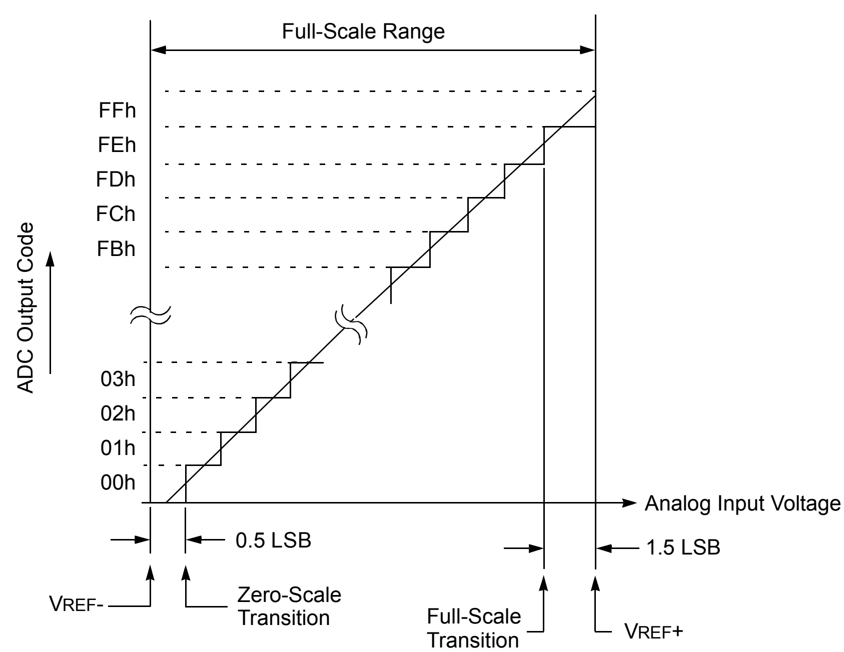


TABLE 15-2: SUMMARY OF REGISTERS ASSOCIATED WITH ADC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON	ADCS<2:0>			CHS<2:0>			GO/$\overline{\text{DONE}}$	ADON	88
ADRES	ADRES<7:0>								89
ANSELA	—	—	—	—	—	ANSA2	ANSA1	ANSA0	70
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	—	—	ADFVR<1:0>		78
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
PIE1	—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—	41
PIR1	—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—	42
TRISA	—	—	—	—	—	TRISA2	TRISA1	TRISA0	69

Legend: x = unknown, u = unchanged, — = unimplemented read as ‘0’, q = value depends on condition. Shaded cells are not used for ADC module.

16.0 TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 8-bit prescaler (independent of Watchdog Timer)
- Programmable internal or external clock source
- Programmable external clock edge selection
- Interrupt on overflow

Figure 16-1 is a block diagram of the Timer0 module.

16.1 Timer0 Operation

The Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

16.1.1 8-BIT TIMER MODE

The Timer0 module will increment every instruction cycle, if used without a prescaler. 8-Bit Timer mode is selected by clearing the T0CS bit of the OPTION_REG register.

When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

Note:	The value written to the TMR0 register can be adjusted, in order to account for the two instruction cycle delay when TMR0 is written.

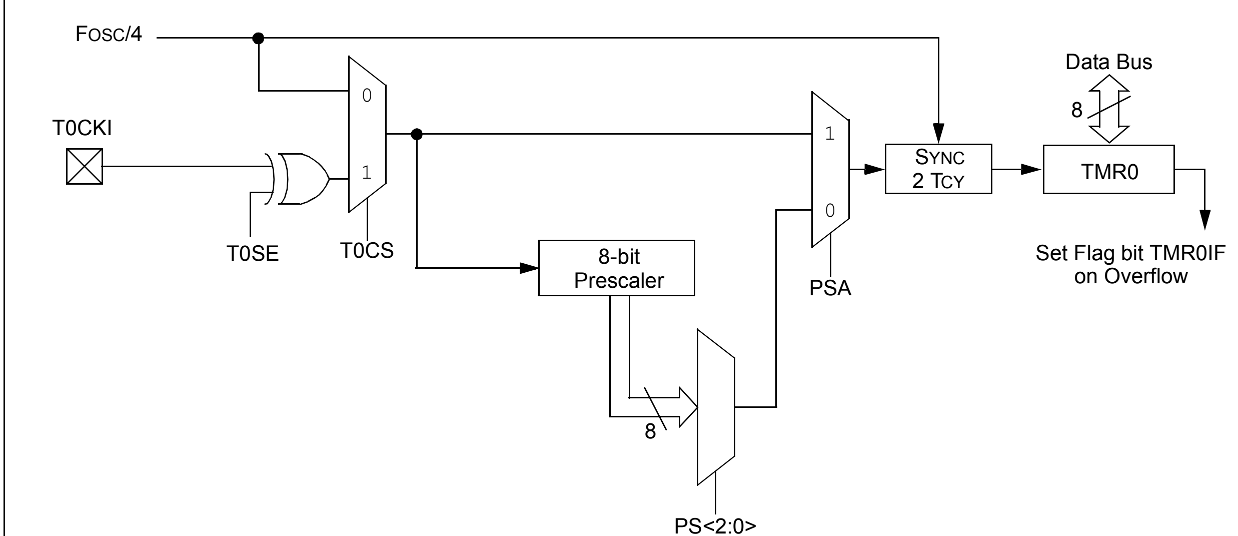
16.1.2 8-BIT COUNTER MODE

In 8-Bit Counter mode, the Timer0 module will increment on every rising or falling edge of the T0CKI pin.

8-Bit Counter mode using the T0CKI pin is selected by setting the T0CS bit in the OPTION_REG register to '1'.

The rising or falling transition of the incrementing edge for the external input source is determined by the T0SE bit in the OPTION_REG register.

FIGURE 16-1: BLOCK DIAGRAM OF THE TIMER0 PRESCALER



16.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A single software programmable prescaler is available for use with Timer0. The prescaler assignment is controlled by the PSA bit of the OPTION_REG register. To assign the prescaler to Timer0, the PSA bit must be cleared to a '0'.

There are 8 prescaler options for the Timer0 module ranging from 1:2 to 1:256. The prescale values are selectable via the PS<2:0> bits of the OPTION_REG register.

The prescaler is not readable or writable. When assigned to the Timer0 module, all instructions writing to the TMR0 register will clear the prescaler.

16.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The TMR0IF interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. The TMR0IF bit can only be cleared in software. The Timer0 interrupt enable is the TMR0IE bit of the INTCON register.

Note:	The Timer0 interrupt cannot wake the processor from Sleep since the timer is frozen during Sleep.

16.1.5 8-BIT COUNTER MODE SYNCHRONIZATION

When in 8-Bit Counter mode, the incrementing edge on the T0CKI pin must be synchronized to the instruction clock. Synchronization can be accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the instruction clock. The high and low periods of the external clocking source must meet the timing requirements as shown in **Section 24.0 "Electrical Specifications"**.

REGISTER 16-1: OPTION_REG: OPTION REGISTER

R/W-1/u	R/W-1/u	R/W-1/u	R/W-1/u	R/W-1/u	R/W-1/u	R/W-1/u	R/W-1/u
$\overline{\text{WPUEN}}$(1)	INTEDG	T0CS	T0SE	PSA	PS<2:0>		
bit 7							bit 0

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

bit 7 **$\overline{\text{WPUEN}}$:** Weak Pull-up Enable bit(1)
1 = Weak pull-ups are disabled
0 = Weak pull-ups are enabled by individual PORT latch values

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of INT pin
0 = Interrupt on falling edge of INT pin

bit 5 **T0CS:** TMR0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (FOSC/4)

bit 4 **T0SE:** TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler is inactive and has no effect on the Timer 0 module
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

Bit Value	TMR0 Rate
000	1 : 2
001	1 : 4
010	1 : 8
011	1 : 16
100	1 : 32
101	1 : 64
110	1 : 128
111	1 : 256

Note 1: $\overline{\text{WPUEN}}$ does not disable the pull-up for the $\overline{\text{MCLR}}$ input when $\overline{\text{MCLR}}$ = 1.

TABLE 16-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER0

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	T0CS	T0SE	PSA	PS<2:0>			95
TMR0	Timer0 module Register								40
TRISA	—	—	—	—	—	TRISA2	TRISA1	TRISA0	69

Legend: – = Unimplemented locations, read as ‘0’, u = unchanged, x = unknown. Shaded cells are not used by the Timer0 module.

17.0 TIMER2 MODULE

The Timer2 module is an 8-bit timer with the following features:

- 8-bit timer register (TMR2)
- 8-bit period register (PR2)
- Interrupt on TMR2 match with PR2
- Software programmable prescaler (1:1, 1:4, 1:16, 1:64)
- Software programmable postscaler (1:1 to 1:16)

See Figure 17-1 for a block diagram of Timer2.

17.1 Timer2 Operation

The clock input to the Timer2 module is the system instruction clock (FOSC/4). The clock is fed into the Timer2 prescaler, which has prescale options of 1:1, 1:4 or 1:64. The output of the prescaler is then used to increment the TMR2 register.

The values of TMR2 and PR2 are constantly compared to determine when they match. TMR2 will increment from 00h until it matches the value in PR2. When a match occurs, two things happen:

- TMR2 is reset to 00h on the next increment cycle.
- The Timer2 postscaler is incremented.

The match output of the Timer2/PR2 comparator is then fed into the Timer2 postscaler. The postscaler has postscale options of 1:1 to 1:16 inclusive. The output of the Timer2 postscaler is used to set the TMR2IF interrupt flag bit in the PIR1 register.

The TMR2 and PR2 registers are both fully readable and writable. On any Reset, the TMR2 register is set to 00h and the PR2 register is set to FFh.

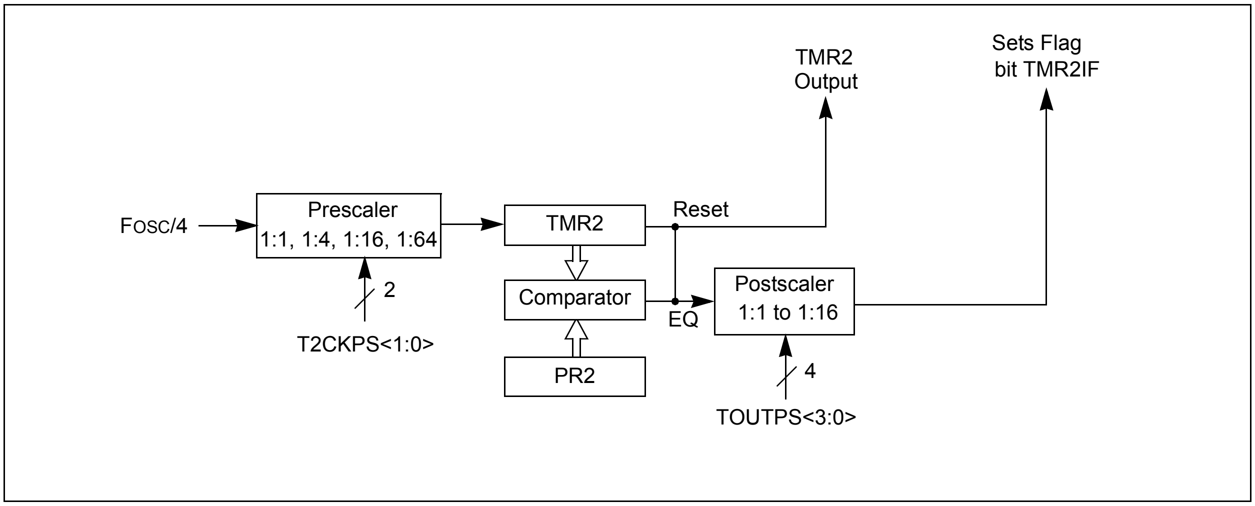
Timer2 is turned on by setting the TMR2ON bit in the T2CON register to a '1'. Timer2 is turned off by clearing the TMR2ON bit to a '0'.

The Timer2 prescaler is controlled by the T2CKPS bits in the T2CON register. The Timer2 postscaler is controlled by the TOUTPS bits in the T2CON register. The prescaler and postscaler counters are cleared when:

- A write to TMR2 occurs.
- A write to T2CON occurs.
- Any device Reset occurs (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset, or Brown-out Reset).

Note:	TMR2 is not cleared when T2CON is written.

FIGURE 17-1: TIMER2 BLOCK DIAGRAM



REGISTER 17-1: T2CON: TIMER2 CONTROL REGISTER

U-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
—	TOUTPS<3:0>				TMR2ON	T2CKPS<1:0>	
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as ‘0’

bit 6-3 **TOUTPS<3:0>:** Timer2 Output Postscaler Select bits
1111 = 1:16 Postscaler
1110 = 1:15 Postscaler
1101 = 1:14 Postscaler
1100 = 1:13 Postscaler
1011 = 1:12 Postscaler
1010 = 1:11 Postscaler
1001 = 1:10 Postscaler
1000 = 1:9 Postscaler
0111 = 1:8 Postscaler
0110 = 1:7 Postscaler
0101 = 1:6 Postscaler
0100 = 1:5 Postscaler
0011 = 1:4 Postscaler
0010 = 1:3 Postscaler
0001 = 1:2 Postscaler
0000 = 1:1 Postscaler

bit 2 **TMR2ON:** Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits
11 = Prescaler is 64
10 = Prescaler is 16
01 = Prescaler is 4
00 = Prescaler is 1

TABLE 17-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
PIE1	—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—	41
PIR1	—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—	42
PR2	Timer2 module Period Register								96
TMR2	Timer2 module Register								96
T2CON	—	TOUTPS<3:0>				TMR2ON	T2CKPS<1:0>		97

Legend: x = unknown, u = unchanged, – = unimplemented read as ‘0’. Shaded cells are not used for Timer2 module.

18.0 PULSE-WIDTH MODULATION (PWM) MODULE

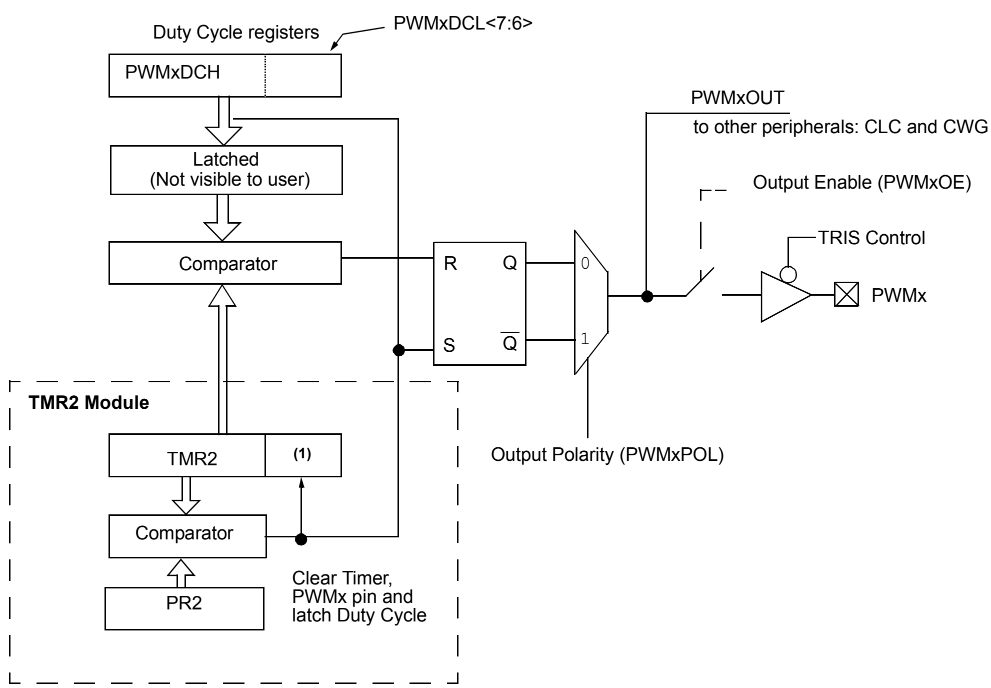
The PWM module generates a Pulse-Width Modulated signal determined by the duty cycle, period, and resolution that are configured by the following registers:

- PR2
- T2CON
- PWMxDCH
- PWMxDCL
- PWMxCON

Figure 18-1 shows a simplified block diagram of PWM operation.

Figure 18-2 shows a typical waveform of the PWM signal.

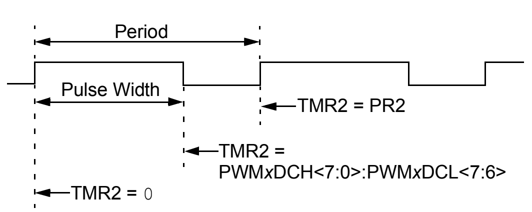
FIGURE 18-1: SIMPLIFIED PWM BLOCK DIAGRAM



Note 1: 8-bit timer is concatenated with the two Least Significant bits of 1/FOSC adjusted by the Timer2 prescaler to create a 10-bit time base.

For a step-by-step procedure on how to set up this module for PWM operation, refer to **Section 18.1.9 "Setup for PWM Operation using PWMx Pins"**.

FIGURE 18-2: PWM OUTPUT



18.1 PWMx Pin Configuration

All PWM outputs are multiplexed with the PORT data latch. The user must configure the pins as outputs by clearing the associated TRIS bits.

> **Note:** Clearing the PWMxOE bit will relinquish control of the PWMx pin.

18.1.1 FUNDAMENTAL OPERATION

The PWM module produces a 10-bit resolution output. Timer2 and PR2 set the period of the PWM. The PWMxDCL and PWMxDCH registers configure the duty cycle. The period is common to all PWM modules, whereas the duty cycle is independently controlled.

> **Note:** The Timer2 postscaler is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

All PWM outputs associated with Timer2 are set when TMR2 is cleared. Each PWMx is cleared when TMR2 is equal to the value specified in the corresponding PWMxDCH (8 MSb) and PWMxDCL<7:6> (2 LSb) registers. When the value is greater than or equal to PR2, the PWM output is never cleared (100% duty cycle).

> **Note:** The PWMxDCH and PWMxDCL registers are double buffered. The buffers are updated when Timer2 matches PR2. Care should be taken to update both registers before the timer match occurs.

18.1.2 PWM OUTPUT POLARITY

The output polarity is inverted by setting the PWMxPOL bit of the PWMxCON register.

18.1.3 PWM PERIOD

The PWM period is specified by the PR2 register of Timer2. The PWM period can be calculated using the formula of Equation 18-1.

EQUATION 18-1: PWM PERIOD

$$PWM\ Period = [(PR2) + 1] \bullet 4 \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

Note: TOSC = 1/FOSC

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The PWM output is active. (Exception: When the PWM duty cycle = 0%, the PWM output will remain inactive.)
- The PWMxDCH and PWMxDCL register values are latched into the buffers.

> **Note:** The Timer2 postscaler has no effect on the PWM operation.

18.1.4 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to the PWMxDCH and PWMxDCL register pair. The PWMxDCH register contains the eight MSbs and the PWMxDCL<7:6>, the two LSbs. The PWMxDCH and PWMxDCL registers can be written to at any time.

Equation 18-2 is used to calculate the PWM pulse width.

Equation 18-3 is used to calculate the PWM duty cycle ratio.

EQUATION 18-2: PULSE WIDTH

$$Pulse\ Width = (PWMxDCH{:}PWMxDCL\langle 7{:}6\rangle) \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

Note: TOSC = 1/FOSC

EQUATION 18-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(PWMxDCH{:}PWMxDCL\langle 7{:}6\rangle)}{4(PR2 + 1)}$$

The 8-bit timer TMR2 register is concatenated with the two Least Significant bits of 1/FOSC, adjusted by the Timer2 prescaler to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

18.1.5 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is 10 bits when PR2 is 255. The resolution is a function of the PR2 register value as shown by Equation 18-4.

EQUATION 18-4: PWM RESOLUTION

$$Resolution = \frac{\log[4(PR2 + 1)]}{\log(2)} \text{ bits}$$

Note:	If the pulse width value is greater than the period the assigned PWM pin(s) will remain unchanged.

TABLE 18-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (F_{OSC} = 20 MHz)

PWM Frequency	0.31 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale (1, 4, 64)	64	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

TABLE 18-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (F_{OSC} = 8 MHz)

PWM Frequency	0.31 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale (1, 4, 64)	64	4	1	1	1	1
PR2 Value	0x65	0x65	0x65	0x19	0x0C	0x09
Maximum Resolution (bits)	8	8	8	6	5	5

18.1.6 OPERATION IN SLEEP MODE

In Sleep mode, the TMR2 register will not increment and the state of the module will not change. If the PWMx pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2 will continue from its previous state.

18.1.7 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency (F_{OSC}). Any changes in the system clock frequency will result in changes to the PWM frequency. Refer to **Section 4.0 "Oscillator Module"** for additional details.

18.1.8 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the PWM registers to their Reset states.

18.1.9 SETUP FOR PWM OPERATION USING PWMx PINS

The following steps should be taken when configuring the module for PWM operation using the PWMx pins:

1. Disable the PWMx pin output driver(s) by setting the associated TRIS bit(s).
2. Clear the PWMxCON register.
3. Load the PR2 register with the PWM period value.
4. Clear the PWMxDCH register and bits <7:6> of the PWMxDCL register.
5. Configure and start Timer2:
   - Clear the TMR2IF interrupt flag bit of the PIR1 register. See Note below.
   - Configure the T2CKPS bits of the T2CON register with the Timer2 prescale value.
   - Enable Timer2 by setting the TMR2ON bit of the T2CON register.
6. Enable PWM output pin and wait until Timer2 overflows, TMR2IF bit of the PIR1 register is set. See Note below.
7. Enable the PWMx pin output driver(s) by clearing the associated TRIS bit(s) and setting the PWMxOE bit of the PWMxCON register.
8. Configure the PWM module by loading the PWMxCON register with the appropriate values.

Note 1: In order to send a complete duty cycle and period on the first PWM output, the above steps must be followed in the order given. If it is not critical to start with a complete PWM signal, then move Step 8 to replace Step 4.

2: For operation with other peripherals only, disable PWMx pin outputs.

18.2 PWM Register Definitions

REGISTER 18-1: PWMxCON: PWM CONTROL REGISTER

R/W-0/0	R/W-0/0	R-0/0	R/W-0/0	U-0	U-0	U-0	U-0
PWMxEN	PWMxOE	PWMxOUT	PWMxPOL	—	—	—	—
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 **PWMxEN:** PWM Module Enable bit
1 = PWM module is enabled
0 = PWM module is disabled

bit 6 **PWMxOE:** PWM Module Output Enable bit
1 = Output to PWMx pin is enabled
0 = Output to PWMx pin is disabled

bit 5 **PWMxOUT:** PWM Module Output Value bit

bit 4 **PWMxPOL:** PWMx Output Polarity Select bit
1 = PWM output is active low.
0 = PWM output is active high.

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

REGISTER 18-2: PWMxDCH: PWM DUTY CYCLE HIGH BITS

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
PWMxDCH<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 **PWMxDCH<7:0>:** PWM Duty Cycle Most Significant bits
These bits are the MSbs of the PWM duty cycle. The two LSbs are found in the PWMxDCL Register.

REGISTER 18-3: PWMxDCL: PWM DUTY CYCLE LOW BITS

R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0	U-0	U-0
PWMxDCL<7:6>		—	—	—	—	—	—
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 **PWMxDCL<7:6>:** PWM Duty Cycle Least Significant bits
These bits are the LSbs of the PWM duty cycle. The MSbs are found in the PWMxDCH Register.

bit 5-0 **Unimplemented:** Read as ‘0’

TABLE 18-3: SUMMARY OF REGISTERS ASSOCIATED WITH PWM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	—	—	ANSA2	ANSA1	ANSA0	70
LATA	—	—	—	—	—	LATA2	LATA1	LATA0	70
PORTA	—	—	—	—	RA3	RA2	RA1	RA0	69
PR2	Timer2 module Period Register								96
PWM1CON	PWM1EN	PWM1OE	PWM1OUT	PWM1POL	—	—	—	—	102
PWM1DCH	PWM1DCH<7:0>								103
PWM1DCL	PWM1DCL<7:6>		—	—	—	—	—	—	103
PWM2CON	PWM2EN	PWM2OE	PWM2OUT	PWM2POL	—	—	—	—	102
PWM2DCH	PWM2DCH<7:0>								103
PWM2DCL	PWM2DCL<7:6>		—	—	—	—	—	—	103
T2CON	—	TOUTPS<3:0>				TMR2ON	T2CKPS<1:0>		97
TMR2	Timer2 module Register								96
TRISA	—	—	—	—	—	TRISA2	TRISA1	TRISA0	69

Legend: – = Unimplemented locations, read as ‘0’, u = unchanged, x = unknown. Shaded cells are not used by the PWM.

19.0 CONFIGURABLE LOGIC CELL (CLC)

The Configurable Logic Cell (CLCx) provides programmable logic that operates outside the speed limitations of software execution. The logic cell selects any combination of the eight input signals and through the use of configurable gates reduces the selected inputs to four logic lines that drive one of eight selectable single-output logic functions.

Input sources are a combination of the following:

- Two I/O pins
- Internal clocks
- Peripherals
- Register bits

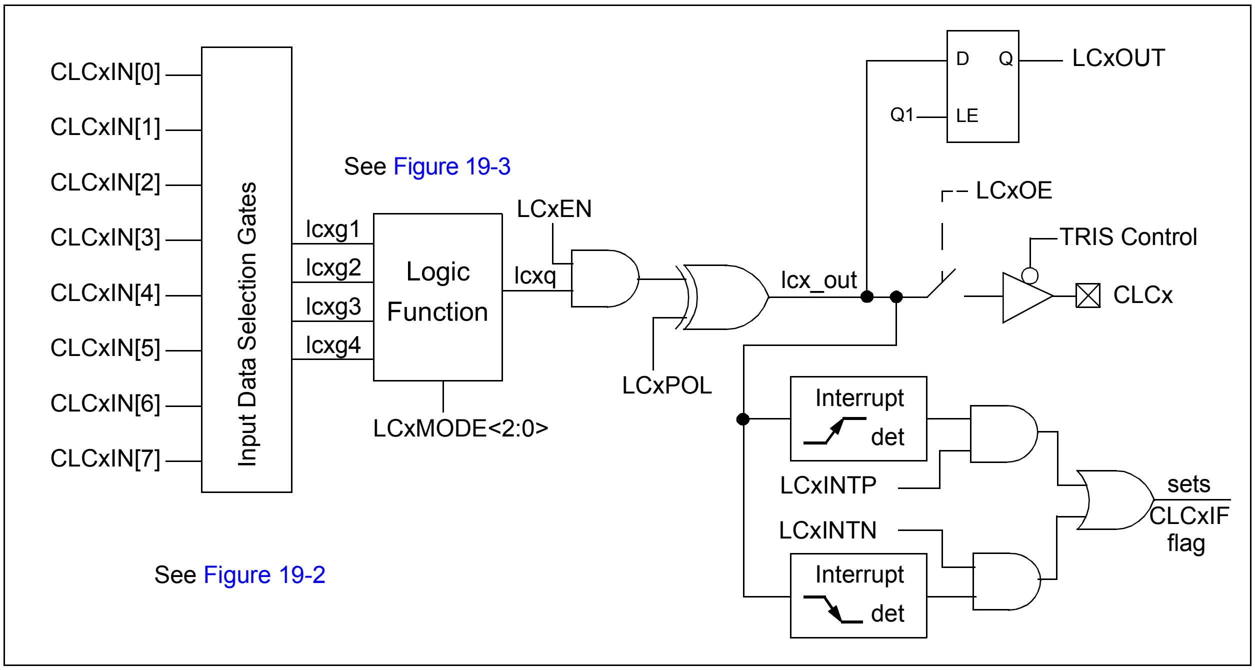
The output can be directed internally to peripherals and to an output pin.

Refer to Figure 19-1 for a simplified diagram showing signal flow through the CLCx.

Possible configurations include:

- Combinatorial Logic
  - AND
  - NAND
  - AND-OR
  - AND-OR-INVERT
  - OR-XOR
  - OR-XNOR
- Latches
  - S-R
  - Clocked D with Set and Reset
  - Transparent D with Set and Reset
  - Clocked J-K with Reset

FIGURE 19-1: CLCx SIMPLIFIED BLOCK DIAGRAM



19.1 CLCx Setup

Programming the CLCx module is performed by configuring the four stages in the logic signal flow. The four stages are:

- Data selection
- Data gating
- Logic function selection
- Output polarity

Each stage is setup at run time by writing to the corresponding CLCx Special Function Registers. This has the added advantage of permitting logic reconfiguration on-the-fly during program execution.

19.1.1 DATA SELECTION

There are eight signals available as inputs to the configurable logic. Four 8-input multiplexers are used to select the inputs to pass on to the next stage.

Data inputs are selected with the CLCxSEL0 and CLCxSEL1 registers (Register 19-3 and Register 19-4, respectively).

Data selection is through four multiplexers as indicated on the left side of Figure 19-2. Data inputs in the figure are identified by a generic numbered input name.

Table 19-1 correlates the generic input name to the actual signal for each CLC module. The columns labeled lcxd1 through lcxd4 indicate the MUX output for the selected data input. D1S through D4S are abbreviations for the MUX select input codes: LCxD1S<2:0> through LCxD4S<2:0>, respectively. Selecting a data input in a column excludes all other inputs in that column.

Note:	Data selections are undefined at power-up.

TABLE 19-1: CLCx DATA INPUT SELECTION

Data Input	lcxd1 D1S	lcxd2 D2S	lcxd3 D3S	lcxd4 D4S	CLC 1
CLCxIN[0]	000	000	000	000	CLCx
CLCxIN[1]	001	001	001	001	CLCxIN1
CLCxIN[2]	010	010	010	010	CLCxIN2
CLCxIN[3]	011	011	011	011	PWM1
CLCxIN[4]	100	100	100	100	PWM2
CLCxIN[5]	101	101	101	101	NCOx
CLCxIN[6]	110	110	110	110	FOSC
CLCxIN[7]	111	111	111	111	LFINTOSC

19.1.2 DATA GATING

Outputs from the input multiplexers are directed to the desired logic function input through the data gating stage. Each data gate can direct any combination of the four selected inputs.

Note:	Data gating is undefined at power-up.

The gate stage is more than just signal direction. The gate can be configured to direct each input signal as inverted or non-inverted data. Directed signals are ANDed together in each gate. The output of each gate can be inverted before going on to the logic function stage.

The gating is in essence a 1-to-4 input AND/NAND/OR/NOR gate. When every input is inverted and the output is inverted, the gate is an OR of all enabled data inputs. When the inputs and output are not inverted, the gate is an AND or all enabled inputs.

Table 19-2 summarizes the basic logic that can be obtained in gate 1 by using the gate logic select bits. The table shows the logic of four input variables, but each gate can be configured to use less than four. If no inputs are selected, the output will be zero or one, depending on the gate output polarity bit.

TABLE 19-2: DATA GATING LOGIC

CLCxGLS0	LCxGyPOL	Gate Logic
0x55	1	AND
0x55	0	NAND
0xAA	1	NOR
0xAA	0	OR
0x00	0	Logic 0
0x00	1	Logic 1

It is possible (but not recommended) to select both the true and negated values of an input. When this is done, the gate output is zero, regardless of the other inputs, but may emit logic glitches (transient-induced pulses). If the output of the channel must be zero or one, the recommended method is to set all gate bits to zero and use the gate polarity bit to set the desired level.

Data gating is configured with the logic gate select registers as follows:

- Gate 1: CLCxGLS0 (Register 19-5)
- Gate 2: CLCxGLS1 (Register 19-6)
- Gate 3: CLCxGLS2 (Register 19-7)
- Gate 4: CLCxGLS3 (Register 19-8)

Register number suffixes are different than the gate numbers because other variations of this module have multiple gate selections in the same register.

Data gating is indicated in the right side of Figure 19-2. Only one gate is shown in detail. The remaining three gates are configured identically with the exception that the data enables correspond to the enables for that gate.

19.1.3 LOGIC FUNCTION

There are eight available logic functions including:

- AND-OR
- OR-XOR
- AND
- S-R Latch
- D Flip-Flop with Set and Reset
- D Flip-Flop with Reset
- J-K Flip-Flop with Reset
- Transparent Latch with Set and Reset

Logic functions are shown in Figure 19-3. Each logic function has four inputs and one output. The four inputs are the four data gate outputs of the previous stage. The output is fed to the inversion stage and from there to other peripherals, an output pin, and back to the CLCx itself.

19.1.4 OUTPUT POLARITY

The last stage in the configurable logic cell is the output polarity. Setting the LCxPOL bit of the CLCxCON register inverts the output signal from the logic stage. Changing the polarity while the interrupts are enabled will cause an interrupt for the resulting output transition.

19.1.5 CLCX SETUP STEPS

The following steps should be followed when setting up the CLCx:

- Disable CLCx by clearing the LCxEN bit.
- Select desired inputs using CLCxSEL0 and CLCxSEL1 registers (See Table 19-1).
- Clear any associated ANSEL bits.
- Set all TRIS bits associated with inputs.
- Clear all TRIS bits associated with outputs.
- Enable the chosen inputs through the four gates using CLCxGLS0, CLCxGLS1, CLCxGLS2, and CLCxGLS3 registers.
- Select the gate output polarities with the LCxPOLy bits of the CLCxPOL register.
- Select the desired logic function with the LCxMODE<2:0> bits of the CLCxCON register.
- Select the desired polarity of the logic output with the LCxPOL bit of the CLCxPOL register. (This step may be combined with the previous gate output polarity step).
- If driving the CLCx pin, set the LCxOE bit of the CLCxCON register and also clear the TRIS bit corresponding to that output.
- If interrupts are desired, configure the following bits:
  - Set the LCxINTP bit in the CLCxCON register for rising event.
  - Set the LCxINTN bit in the CLCxCON register or falling event.
  - Set the CLCxIE bit of the associated PIE registers.
  - Set the GIE and PEIE bits of the INTCON register.
- Enable the CLCx by setting the LCxEN bit of the CLCxCON register.

19.2 CLCx Interrupts

An interrupt will be generated upon a change in the output value of the CLCx when the appropriate interrupt enables are set. A rising edge detector and a falling edge detector are present in each CLC for this purpose.

The CLCxIF bit of the associated PIR registers will be set when either edge detector is triggered and its associated enable bit is set. The LCxINTP enables rising edge interrupts and the LCxINTN bit enables falling edge interrupts. Both are located in the CLCxCON register.

To fully enable the interrupt, set the following bits:

- LCxON bit of the CLCxCON register
- CLCxIE bit of the associated PIE registers
- LCxINTP bit of the CLCxCON register (for a rising edge detection)
- LCxINTN bit of the CLCxCON register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

The CLCxIF bit of the associated PIR registers must be cleared in software as part of the interrupt service. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

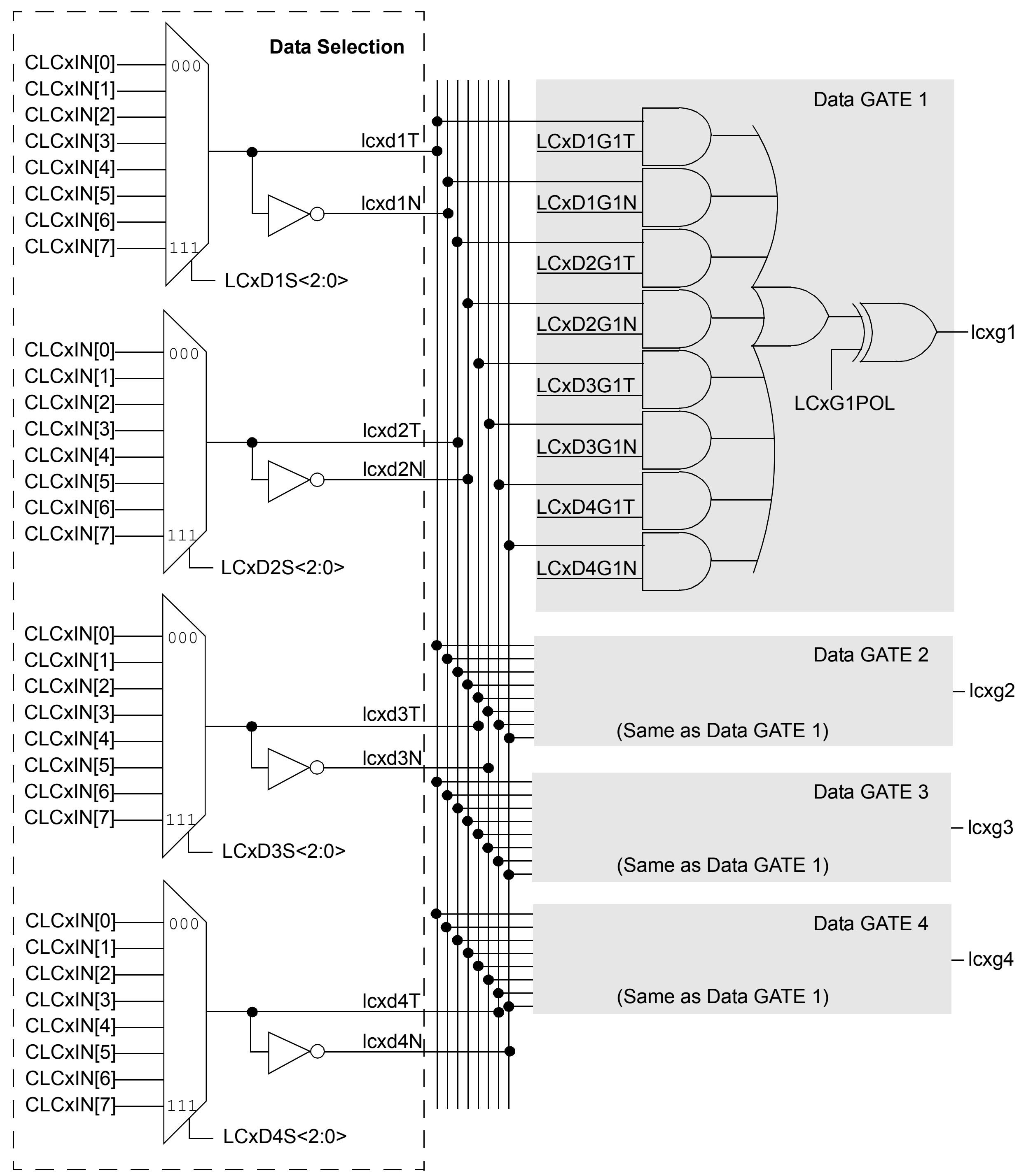
19.3 Effects of a Reset

The CLCxCON register is cleared to zero as the result of a Reset. All other selection and gating values remain unchanged.

19.4 Operation During Sleep

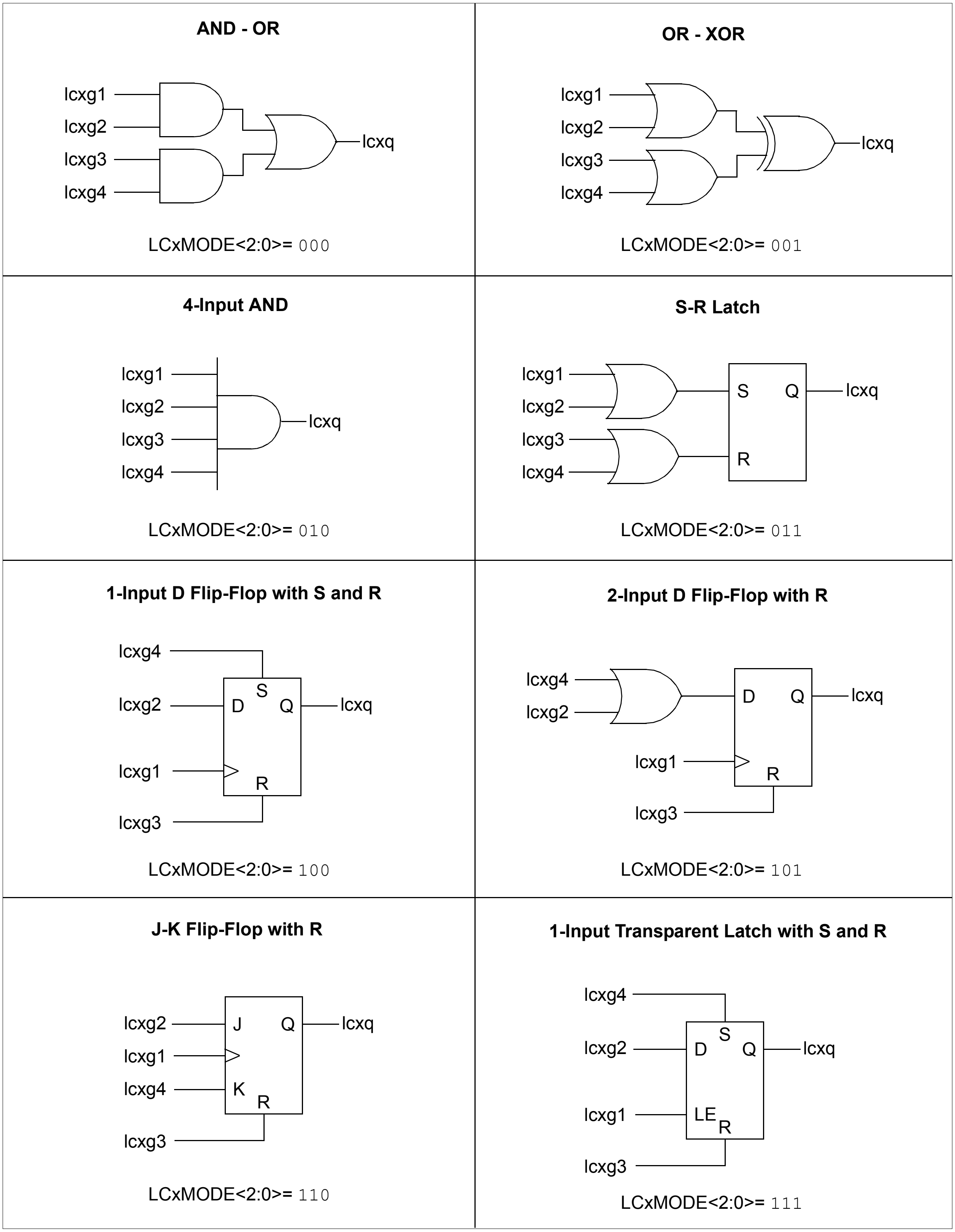
The selection, gating, and logic functions are not affected by Sleep. Operation will continue provided that the source signals are also not affected by Sleep.

FIGURE 19-2: INPUT DATA SELECTION AND GATING



Note: All controls are undefined at power-up.

FIGURE 19-3: PROGRAMMABLE LOGIC FUNCTIONS



19.5 CLC Control Registers

REGISTER 19-1: CLCxCON: CONFIGURABLE LOGIC CELL CONTROL REGISTER

R/W-0/0	R/W-0/0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
LCxEN	LCxOE	LCxOUT	LCxINTP	LCxINTN	LCxMODE<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 Reset
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **LCxEN:** Configurable Logic Cell Enable bit
1 = Configurable Logic Cell is enabled and mixing input signals
0 = Configurable Logic Cell is disabled and has logic zero output

bit 6 **LCxOE:** Configurable Logic Cell Output Enable bit
1 = Configurable Logic Cell port pin output enabled
0 = Configurable Logic Cell port pin output disabled

bit 5 **LCxOUT:** Configurable Logic Cell Data Output bit
Read-only: logic cell output data, after LCxPOL; sampled from lcx_out wire.

bit 4 **LCxINTP:** Configurable Logic Cell Positive Edge Going Interrupt Enable bit
1 = CLCxIF will be set when a rising edge occurs on lcx_out
0 = CLCxIF will not be set

bit 3 **LCxINTN:** Configurable Logic Cell Negative Edge Going Interrupt Enable bit
1 = CLCxIF will be set when a falling edge occurs on lcx_out
0 = CLCxIF will not be set

bit 2-0 **LCxMODE<2:0>:** Configurable Logic Cell Functional Mode bits
111 = Cell is 1-input transparent latch with S and R
110 = Cell is J-K Flip-Flop with R
101 = Cell is 2-input D Flip-Flop with R
100 = Cell is 1-input D Flip-Flop with S and R
011 = Cell is S-R latch
010 = Cell is 4-input AND
001 = Cell is OR-XOR
000 = Cell is AND-OR

REGISTER 19-2: CLCxPOL: SIGNAL POLARITY CONTROL REGISTER

R/W-x/u	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LCxPOL	—	—	—	LCxG4POL	LCxG3POL	LCxG2POL	LCxG1POL
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 Reset
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **LCxPOL:** LCOUT Polarity Control bit
1 = The output of the logic cell is inverted
0 = The output of the logic cell is not inverted

bit 6-4 **Unimplemented:** Read as ‘0’

bit 3 **LCxG4POL:** Gate 4 Output Polarity Control bit
1 = The output of gate 4 is inverted when applied to the logic cell
0 = The output of gate 4 is not inverted

bit 2 **LCxG3POL:** Gate 3 Output Polarity Control bit
1 = The output of gate 3 is inverted when applied to the logic cell
0 = The output of gate 3 is not inverted

bit 1 **LCxG2POL:** Gate 2 Output Polarity Control bit
1 = The output of gate 2 is inverted when applied to the logic cell
0 = The output of gate 2 is not inverted

bit 0 **LCxG1POL:** Gate 1 Output Polarity Control bit
1 = The output of gate 1 is inverted when applied to the logic cell
0 = The output of gate 1 is not inverted

REGISTER 19-3: CLCxSEL0: MULTIPLEXER DATA 1 AND 2 SELECT REGISTER

U-0	R/W-x/u	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	LCxD2S<2:0>(1)			—	LCxD1S<2:0>(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 **Unimplemented:** Read as '0'

bit 6-4 **LCxD2S<2:0>:** Input Data 2 Selection Control bits(1)

111 = CLCxIN[7] is selected for lcxd2.
110 = CLCxIN[6] is selected for lcxd2.
101 = CLCxIN[5] is selected for lcxd2.
100 = CLCxIN[4] is selected for lcxd2.
011 = CLCxIN[3] is selected for lcxd2.
010 = CLCxIN[2] is selected for lcxd2.
001 = CLCxIN[1] is selected for lcxd2.
000 = CLCxIN[0] is selected for lcxd2.

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

bit 2-0 **LCxD1S<2:0>:** Input Data 1 Selection Control bits(1)

111 = CLCxIN[7] is selected for lcxd1.
110 = CLCxIN[6] is selected for lcxd1.
101 = CLCxIN[5] is selected for lcxd1.
100 = CLCxIN[4] is selected for lcxd1.
011 = CLCxIN[3] is selected for lcxd1.
010 = CLCxIN[2] is selected for lcxd1.
001 = CLCxIN[1] is selected for lcxd1.
000 = CLCxIN[0] is selected for lcxd1.

Note 1: See Table 19-1 for signal names associated with inputs.

REGISTER 19-4: CLCxSEL1: MULTIPLEXER DATA 3 AND 4 SELECT REGISTER

U-0	R/W-x/u	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	LCxD4S<2:0>[(1)]			—	LCxD3S<2:0>[(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 **Unimplemented:** Read as ‘0’

bit 6-4 **LCxD4S<2:0>:** Input Data 4 Selection Control bits[(1)]
- `111` = CLCxIN[7] is selected for lcxd4.
- `110` = CLCxIN[6] is selected for lcxd4.
- `101` = CLCxIN[5] is selected for lcxd4
- `100` = CLCxIN[4] is selected for lcxd4.
- `011` = CLCxIN[3] is selected for lcxd4.
- `010` = CLCxIN[2] is selected for lcxd4.
- `001` = CLCxIN[1] is selected for lcxd4.
- `000` = CLCxIN[0] is selected for lcxd4.

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **LCxD3S<2:0>:** Input Data 3 Selection Control bits[(1)]
- `111` = CLCxIN[7] is selected for lcxd3.
- `110` = CLCxIN[6] is selected for lcxd3.
- `101` = CLCxIN[5] is selected for lcxd3.
- `100` = CLCxIN[4] is selected for lcxd3.
- `011` = CLCxIN[3] is selected for lcxd3.
- `010` = CLCxIN[2] is selected for lcxd3.
- `001` = CLCxIN[1] is selected for lcxd3.
- `000` = CLCxIN[0] is selected for lcxd3.

Note 1: See Table 19-1 for signal names associated with inputs.

REGISTER 19-5: CLCxGLS0: GATE 1 LOGIC SELECT 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
LCxG1D4T	LCxG1D4N	LCxG1D3T	LCxG1D3N	LCxG1D2T	LCxG1D2N	LCxG1D1T	LCxG1D1N
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 **LCxG1D4T:** Gate 1 Data 4 True (non-inverted) bit
1 = lcxd4T is gated into lcxg1
0 = lcxd4T is not gated into lcxg1

bit 6 **LCxG1D4N:** Gate 1 Data 4 Negated (inverted) bit
1 = lcxd4N is gated into lcxg1
0 = lcxd4N is not gated into lcxg1

bit 5 **LCxG1D3T:** Gate 1 Data 3 True (non-inverted) bit
1 = lcxd3T is gated into lcxg1
0 = lcxd3T is not gated into lcxg1

bit 4 **LCxG1D3N:** Gate 1 Data 3 Negated (inverted) bit
1 = lcxd3N is gated into lcxg1
0 = lcxd3N is not gated into lcxg1

bit 3 **LCxG1D2T:** Gate 1 Data 2 True (non-inverted) bit
1 = lcxd2T is gated into lcxg1
0 = lcxd2T is not gated into lcxg1

bit 2 **LCxG1D2N:** Gate 1 Data 2 Negated (inverted) bit
1 = lcxd2N is gated into lcxg1
0 = lcxd2N is not gated into lcxg1

bit 1 **LCxG1D1T:** Gate 1 Data 1 True (non-inverted) bit
1 = lcxd1T is gated into lcxg1
0 = lcxd1T is not gated into lcxg1

bit 0 **LCxG1D1N:** Gate 1 Data 1 Negated (inverted) bit
1 = lcxd1N is gated into lcxg1
0 = lcxd1N is not gated into lcxg1

REGISTER 19-6: CLCxGLS1: GATE 2 LOGIC SELECT 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
LCxG2D4T	LCxG2D4N	LCxG2D3T	LCxG2D3N	LCxG2D2T	LCxG2D2N	LCxG2D1T	LCxG2D1N
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 **LCxG2D4T:** Gate 2 Data 4 True (non-inverted) bit
1 = lcxd4T is gated into lcxg2
0 = lcxd4T is not gated into lcxg2

bit 6 **LCxG2D4N:** Gate 2 Data 4 Negated (inverted) bit
1 = lcxd4N is gated into lcxg2
0 = lcxd4N is not gated into lcxg2

bit 5 **LCxG2D3T:** Gate 2 Data 3 True (non-inverted) bit
1 = lcxd3T is gated into lcxg2
0 = lcxd3T is not gated into lcxg2

bit 4 **LCxG2D3N:** Gate 2 Data 3 Negated (inverted) bit
1 = lcxd3N is gated into lcxg2
0 = lcxd3N is not gated into lcxg2

bit 3 **LCxG2D2T:** Gate 2 Data 2 True (non-inverted) bit
1 = lcxd2T is gated into lcxg2
0 = lcxd2T is not gated into lcxg2

bit 2 **LCxG2D2N:** Gate 2 Data 2 Negated (inverted) bit
1 = lcxd2N is gated into lcxg2
0 = lcxd2N is not gated into lcxg2

bit 1 **LCxG2D1T:** Gate 2 Data 1 True (non-inverted) bit
1 = lcxd1T is gated into lcxg2
0 = lcxd1T is not gated into lcxg2

bit 0 **LCxG2D1N:** Gate 2 Data 1 Negated (inverted) bit
1 = lcxd1N is gated into lcxg2
0 = lcxd1N is not gated into lcxg2

REGISTER 19-7: CLCxGLS2: GATE 3 LOGIC SELECT 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
LCxG3D4T	LCxG3D4N	LCxG3D3T	LCxG3D3N	LCxG3D2T	LCxG3D2N	LCxG3D1T	LCxG3D1N
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 **LCxG3D4T:** Gate 3 Data 4 True (non-inverted) bit
1 = lcxd4T is gated into lcxg3
0 = lcxd4T is not gated into lcxg3

bit 6 **LCxG3D4N:** Gate 3 Data 4 Negated (inverted) bit
1 = lcxd4N is gated into lcxg3
0 = lcxd4N is not gated into lcxg3

bit 5 **LCxG3D3T:** Gate 3 Data 3 True (non-inverted) bit
1 = lcxd3T is gated into lcxg3
0 = lcxd3T is not gated into lcxg3

bit 4 **LCxG3D3N:** Gate 3 Data 3 Negated (inverted) bit
1 = lcxd3N is gated into lcxg3
0 = lcxd3N is not gated into lcxg3

bit 3 **LCxG3D2T:** Gate 3 Data 2 True (non-inverted) bit
1 = lcxd2T is gated into lcxg3
0 = lcxd2T is not gated into lcxg3

bit 2 **LCxG3D2N:** Gate 3 Data 2 Negated (inverted) bit
1 = lcxd2N is gated into lcxg3
0 = lcxd2N is not gated into lcxg3

bit 1 **LCxG3D1T:** Gate 3 Data 1 True (non-inverted) bit
1 = lcxd1T is gated into lcxg3
0 = lcxd1T is not gated into lcxg3

bit 0 **LCxG3D1N:** Gate 3 Data 1 Negated (inverted) bit
1 = lcxd1N is gated into lcxg3
0 = lcxd1N is not gated into lcxg3

REGISTER 19-8: CLCxGLS3: GATE 4 LOGIC SELECT 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
LCxG4D4T	LCxG4D4N	LCxG4D3T	LCxG4D3N	LCxG4D2T	LCxG4D2N	LCxG4D1T	LCxG4D1N
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 **LCxG4D4T:** Gate 4 Data 4 True (non-inverted) bit
1 = lcxd4T is gated into lcxg4
0 = lcxd4T is not gated into lcxg4

bit 6 **LCxG4D4N:** Gate 4 Data 4 Negated (inverted) bit
1 = lcxd4N is gated into lcxg4
0 = lcxd4N is not gated into lcxg4

bit 5 **LCxG4D3T:** Gate 4 Data 3 True (non-inverted) bit
1 = lcxd3T is gated into lcxg4
0 = lcxd3T is not gated into lcxg4

bit 4 **LCxG4D3N:** Gate 4 Data 3 Negated (inverted) bit
1 = lcxd3N is gated into lcxg4
0 = lcxd3N is not gated into lcxg4

bit 3 **LCxG4D2T:** Gate 4 Data 2 True (non-inverted) bit
1 = lcxd2T is gated into lcxg4
0 = lcxd2T is not gated into lcxg4

bit 2 **LCxG4D2N:** Gate 4 Data 2 Negated (inverted) bit
1 = lcxd2N is gated into lcxg4
0 = lcxd2N is not gated into lcxg4

bit 1 **LCxG4D1T:** Gate 4 Data 1 True (non-inverted) bit
1 = lcxd1T is gated into lcxg4
0 = lcxd1T is not gated into lcxg4

bit 0 **LCxG4D1N:** Gate 4 Data 1 Negated (inverted) bit
1 = lcxd1N is gated into lcxg4
0 = lcxd1N is not gated into lcxg4

TABLE 19-3: SUMMARY OF REGISTERS ASSOCIATED WITH CLCx

Name	Bit7	Bit6	Bit5	Bit4	BIt3	Bit2	Bit1	Bit0	Register on Page
CLC1CON	LC1EN	LC1OE	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>			110
CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	114
CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	115
CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	116
CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	117
CLC1POL	LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	111
CLC1SEL0	—	LC1D2S<2:0>			—	LC1D1S<2:0>			112
CLC1SEL1	—	LC1D4S<2:0>			—	LC1D3S<2:0>			113
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
PIE1	—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—	41
PIR1	—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—	42
TRISA	—	—	—	—	—	TRISA2	TRISA1	TRISA0	69

Legend: — = unimplemented read as '0'. Shaded cells are not used for CLC module.

20.0 NUMERICALLY CONTROLLED OSCILLATOR (NCO) MODULE

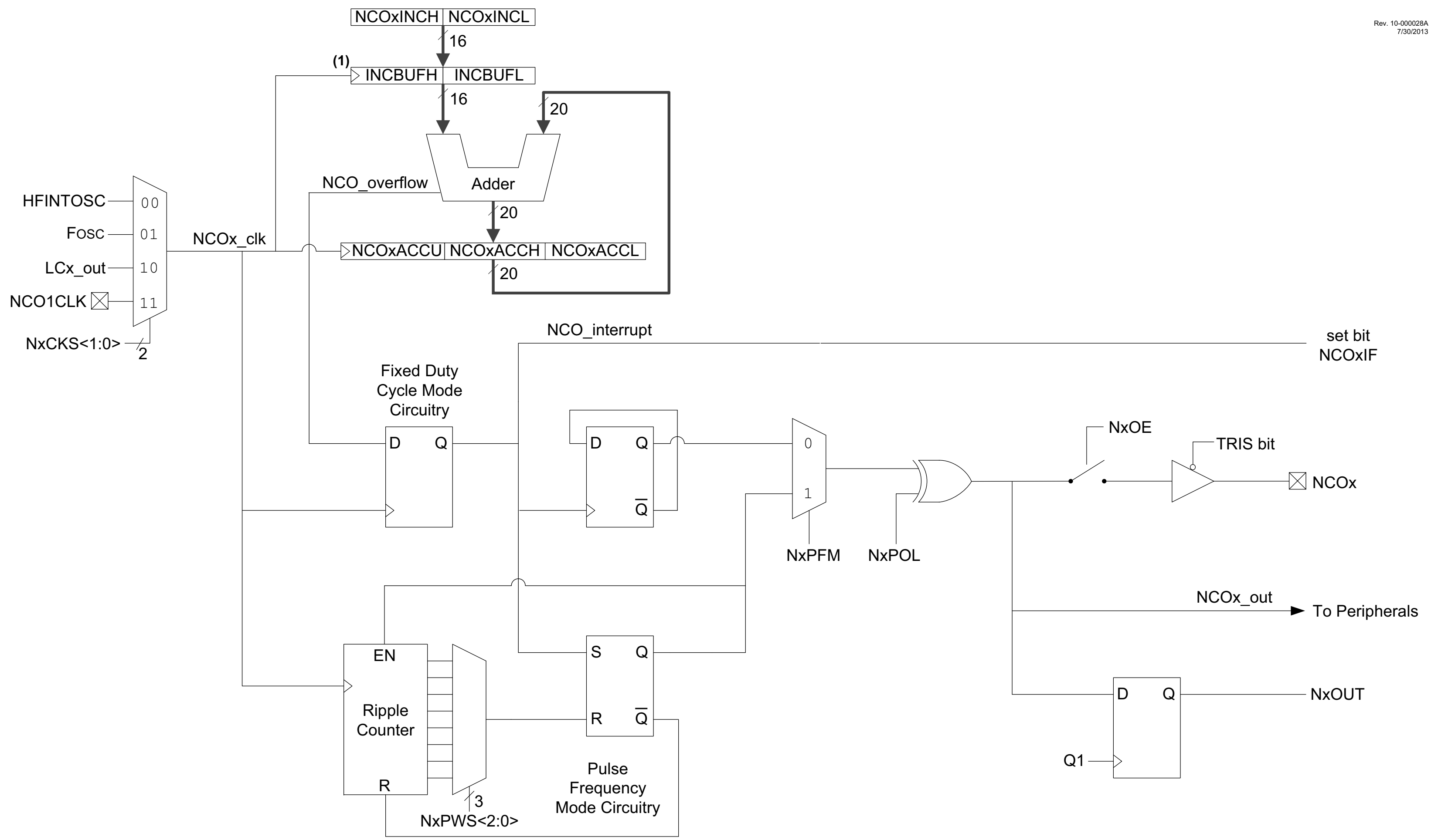
The Numerically Controlled Oscillator (NCOx) module is a timer that uses the overflow from the addition of an increment value to divide the input frequency. The advantage of the addition method over simple counter driven timer is that the resolution of division does not vary with the divider value. The NCOx is most useful for applications that requires frequency accuracy and fine resolution at a fixed duty cycle.

Features of the NCOx include:

- 16-bit increment function
- Fixed Duty Cycle (FDC) mode
- Pulse Frequency (PF) mode
- Output pulse width control
- Multiple clock input sources
- Output polarity control
- Interrupt capability

Figure 20-1 is a simplified block diagram of the NCOx module.

FIGURE 20-1: NUMERICALLY CONTROLLED OSCILLATOR (NCOx) MODULE SIMPLIFIED BLOCK DIAGRAM



Note 1: The increment registers are double-buffered to allow for value changes to be made without first disabling the NCO module. The full increment value is loaded into the buffer registers on the second rising edge of the NCOx_clk signal that occurs immediately after a write to NCOxINCL register. The buffers are not user-accessible and are shown here for reference.

20.1 NCOx OPERATION

The NCOx operates by repeatedly adding a fixed value to an accumulator. Additions occur at the input clock rate. The accumulator will overflow with a carry periodically, which is the raw NCOx output. This effectively reduces the input clock by the ratio of the addition value to the maximum accumulator value. See Equation 20-1.

The NCOx output can be further modified by stretching the pulse or toggling a flip-flop. The modified NCOx output is then distributed internally to other peripherals and optionally output to a pin. The accumulator overflow also generates an interrupt.

The NCOx output creates an instantaneous frequency, which may cause uncertainty. This output depends on the ability of the receiving circuit (i.e., CWG or external resonant converter circuitry) to average the instantaneous frequency to reduce uncertainty.

20.1.1 NCOx CLOCK SOURCES

Clock sources available to the NCOx include:

- HFINTOSC
- FOSC
- LC1OUT
- NCO1CLK pin

The NCOx clock source is selected by configuring the NxCKS<1:0> bits in the NCOxCLK register.

20.1.2 ACCUMULATOR

The Accumulator is a 20-bit register. Read and write access to the Accumulator is available through three registers:

- NCOxACCL
- NCOxACCH
- NCOxACCU

20.1.3 ADDER

The NCOx Adder is a full adder, which operates asynchronously to the clock source selected. The addition of the previous result and the increment value replaces the accumulator value on the rising edge of each input clock.

20.1.4 INCREMENT REGISTERS

The Increment value is stored in two 8-bit registers making up a 16-bit increment. In order of LSB to MSB they are:

- NCOxINCL
- NCOxINCH

Both of the registers are readable and writeable. The Increment registers are double-buffered to allow for value changes to be made without first disabling the NCOx module.

The buffer loads are immediate when the module is disabled. Writing to the MS register first is necessary because then the buffer is loaded synchronously with the NCOx operation after the write is executed on the lower increment register.

Note:	The increment buffer registers are not user-accessible.

EQUATION 20-1:

$$F_{OVERFLOW} = \frac{NCO\ Clock\ Frequency \times Increment\ Value}{2^n}$$

$$n = Accumulator\ width\ in\ bits$$

20.2 FIXED DUTY CYCLE (FDC) MODE

In Fixed Duty Cycle (FDC) mode, every time the Accumulator overflows, the output is toggled. This provides a 50% duty cycle, provided that the increment value remains constant. For more information, see Figure 20-2.

The FDC mode is selected by clearing the NxPFM bit in the NCOxCON register.

20.3 PULSE FREQUENCY (PF) MODE

In Pulse Frequency (PF) mode, every time the Accumulator overflows, the output becomes active for one or more clock periods. See **Section 20.3.1 "OUTPUT PULSE WIDTH CONTROL"** for more information. Once the clock period expires, the output returns to an inactive state. This provides a pulsed output.

The output becomes active on the rising clock edge immediately following the overflow event. For more information, see Figure 20-2.

The value of the active and inactive states depends on the Polarity bit, NxPOL in the NCOxCON register.

The PF mode is selected by setting the NxPFM bit in the NCOxCON register.

20.3.1 OUTPUT PULSE WIDTH CONTROL

When operating in PF mode, the active state of the output can vary in width by multiple clock periods. Various pulse widths are selected with the NxPWS<2:0> bits in the NCOxCLK register.

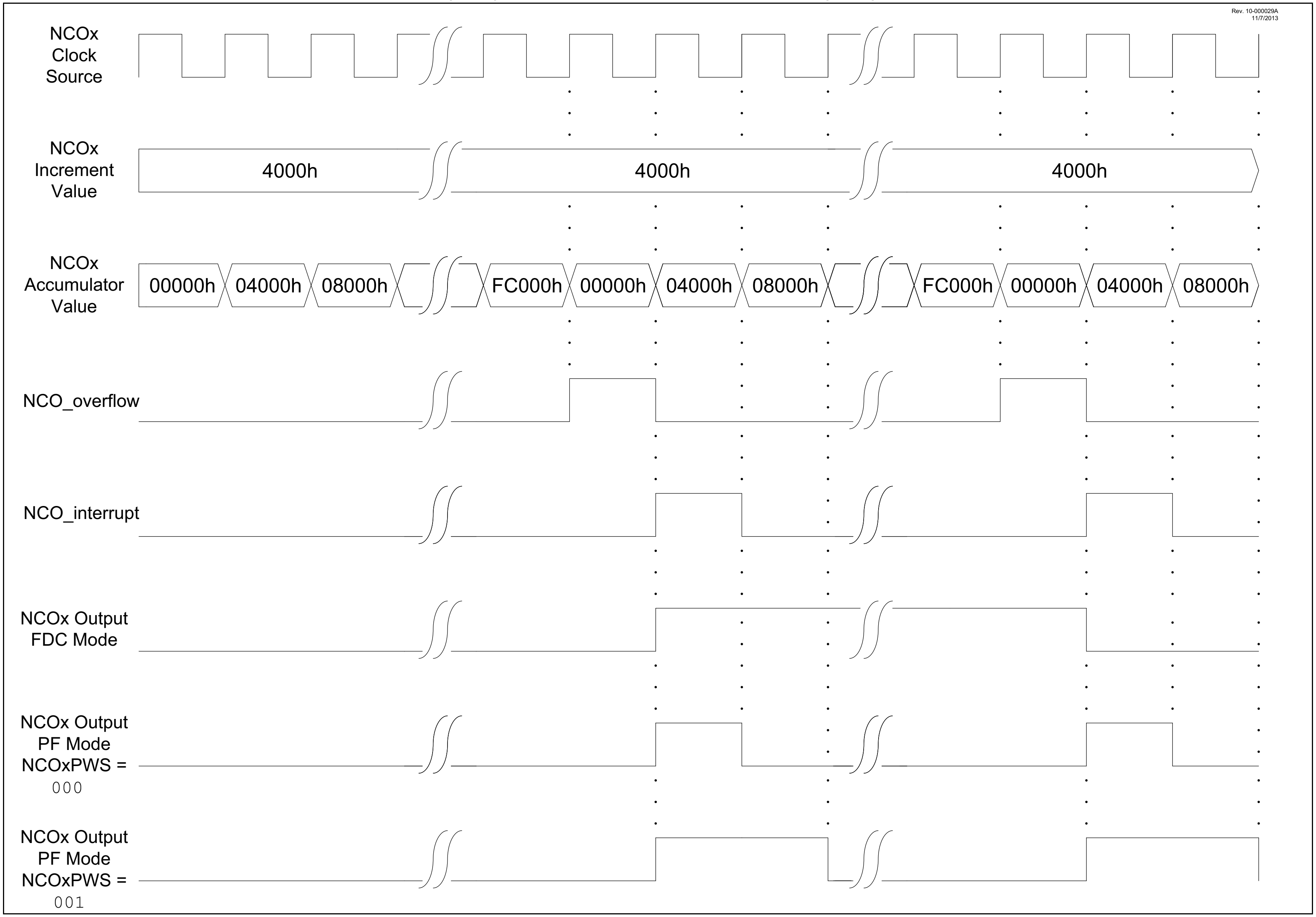
When the selected pulse width is greater than the Accumulator overflow time frame, then NCOx operation is undefined.

20.4 OUTPUT POLARITY CONTROL

The last stage in the NCOx module is the output polarity. The NxPOL bit in the NCOxCON register selects the output polarity. Changing the polarity while the interrupts are enabled will cause an interrupt for the resulting output transition.

The NCOx output can be used internally by source code or other peripherals. This is done by reading the NxOUT (read-only) bit of the NCOxCON register.

FIGURE 20-2: NCO – FIXED DUTY CYCLE (FDC) AND PULSE FREQUENCY MODE (PFM) OUTPUT OPERATION DIAGRAM



20.5 Interrupts

When the Accumulator overflows, the NCOx Interrupt Flag bit, NCOxIF, of the PIR1 register is set. To enable this interrupt event, the following bits must be set:

- NxEN bit of the NCOxCON register
- NCOxIE bit of the PIE1 register
- PEIE bit of the INTCON register
- GIE bit of the INTCON register

The interrupt must be cleared by software by clearing the NCOxIF bit in the Interrupt Service Routine.

20.6 Effects of a Reset

All of the NCOx registers are cleared to zero as the result of a Reset.

20.7 Operation In Sleep

The NCO module operates independently from the system clock and will continue to run during Sleep, provided that the clock source selected remains active.

The HFINTOSC remains active during Sleep when the NCO module is enabled and the HFINTOSC is selected as the clock source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and the NCO clock source, when the NCO is enabled, the CPU will go idle during Sleep, but the NCO will continue to operate and the HFINTOSC will remain active.

This will have a direct effect on the Sleep mode current.

20.8 NCOx Control Registers

REGISTER 20-1: NCOxCON: NCOx CONTROL REGISTER

R/W-0/0	R/W-0/0	R-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
NxEN	NxOE	NxOUT	NxPOL	—	—	—	NxPFM
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 **NxEN:** NCOx Enable bit
1 = NCOx module is enabled
0 = NCOx module is disabled

bit 6 **NxOE:** NCOx Output Enable bit
1 = NCOx output pin is enabled
0 = NCOx output pin is disabled

bit 5 **NxOUT:** NCOx Output bit
1 = NCOx output is high
0 = NCOx output is low

bit 4 **NxPOL:** NCOx Polarity bit
1 = NCOx output signal is active-low (inverted)
0 = NCOx output signal is active-high (non-inverted)

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

bit 0 **NxPFM:** NCOx Pulse Frequency mode bit
1 = NCOx operates in Pulse Frequency mode
0 = NCOx operates in Fixed Duty Cycle mode

REGISTER 20-2: NCOxCLK: NCOx INPUT CLOCK CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
NxPWS<2:0>(1,2)			—	—	—	NxCKS<1:0>	
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 **NxPWS<2:0>:** NCOx Output Pulse Width Select bits(1, 2)
111 = 128 NCOx clock periods
110 = 64 NCOx clock periods
101 = 32 NCOx clock periods
100 = 16 NCOx clock periods
011 = 8 NCOx clock periods
010 = 4 NCOx clock periods
001 = 2 NCOx clock periods
000 = 1 NCOx clock periods

bit 4-2 **Unimplemented:** Read as ‘0’

bit 1-0 **NxCKS<1:0>:** NCOx Clock Source Select bits
11 = LC1OUT
10 = HFINTOSC (16 MHz)
01 = FOSC
00 = NCO1CLK pin

Note 1: NxPWS applies only when operating in Pulse Frequency mode.
2: If NCOx pulse width is greater than NCOx overflow period, operation is undefined.

REGISTER 20-3: NCOxACCL: NCOx ACCUMULATOR REGISTER – LOW BYTE

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
NCOxACC<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 **NCOxACC<7:0>:** NCOx Accumulator, low byte

Note 1: NxPWS applies only when operating in Pulse Frequency mode.
2: If NCOx pulse width is greater than NCOx overflow period, operation is undefined.

REGISTER 20-4: NCOxACCH: NCOx ACCUMULATOR REGISTER – HIGH BYTE

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
NCOxACC<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 **NCOxACC<15:8>:** NCOx Accumulator, high byte

REGISTER 20-5: NCOxACCU: NCOx ACCUMULATOR REGISTER – UPPER BYTE

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	NCOxACC<19:16>			
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 7-4 **Unimplemented:** Read as ‘0’
bit 3-0 **NCOxACC<19:16>:** NCOx Accumulator, upper byte

REGISTER 20-6: NCOxINCL: NCOx INCREMENT REGISTER – LOW BYTE

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-1/1
NCOxINC<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 **NCOxINC<7:0>:** NCOx Increment, low byte

REGISTER 20-7: NCOxINCH: NCOx INCREMENT REGISTER – HIGH BYTE

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
NCOxINC<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 **NCOxINC<15:8>:** NCOx Increment, high byte

TABLE 20-1: SUMMARY OF REGISTERS ASSOCIATED WITH NCOx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CLC1SEL0	—	LC1D2S2	LC1D2S1	LC1D2S0	—	LC1D1S2	LC1D1S1	LC1D1S0	112
CLC1SEL1	—	LC1D4S2	LC1D4S1	LC1D4S0	—	LC1D3S2	LC1D3S1	LC1D3S0	113
CWG1CON1	G1ASDLB<1:0>		G1ASDLA<1:0>		—	—	G1IS<1:0>		140
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
NCO1ACCH	NCO1ACCH<15:8>								126
NCO1ACCL	NCO1ACCL<7:0>								126
NCO1ACCU	—				NCO1ACCU<19:16				126
NCO1CLK	N1PWS<2:0>			—	—	—	N1CKS<1:0>		125
NCO1CON	N1EN	N1OE	N1OUT	N1POL	—	—	—	N1PFM	125
NCO1INCH	NCO1INCH<15:8>								127
NCO1INCL	NCO1INCL<7:0>								127
PIE1	—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—	41
PIR1	—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—	42
TRISA	—	—	—	—	—	TRISA2	TRISA1	TRISA0	*69*

Legend: x = unknown, u = unchanged, – = unimplemented read as ‘0’, q = value depends on condition. Shaded cells are not used for NCO module.

21.0 COMPLEMENTARY WAVEFORM GENERATOR (CWG) MODULE

The Complementary Waveform Generator (CWG) produces a complementary waveform with dead-band delay from a selection of input sources.

The CWG module has the following features:

- Selectable dead-band clock source control
- Selectable input sources
- Output enable control
- Output polarity control
- Dead-band control with Independent 6-bit rising and falling edge dead-band counters
- Auto-shutdown control with:
  - Selectable shutdown sources
  - Auto-restart enable
  - Auto-shutdown pin override control

FIGURE 21-1: CWG BLOCK DIAGRAM

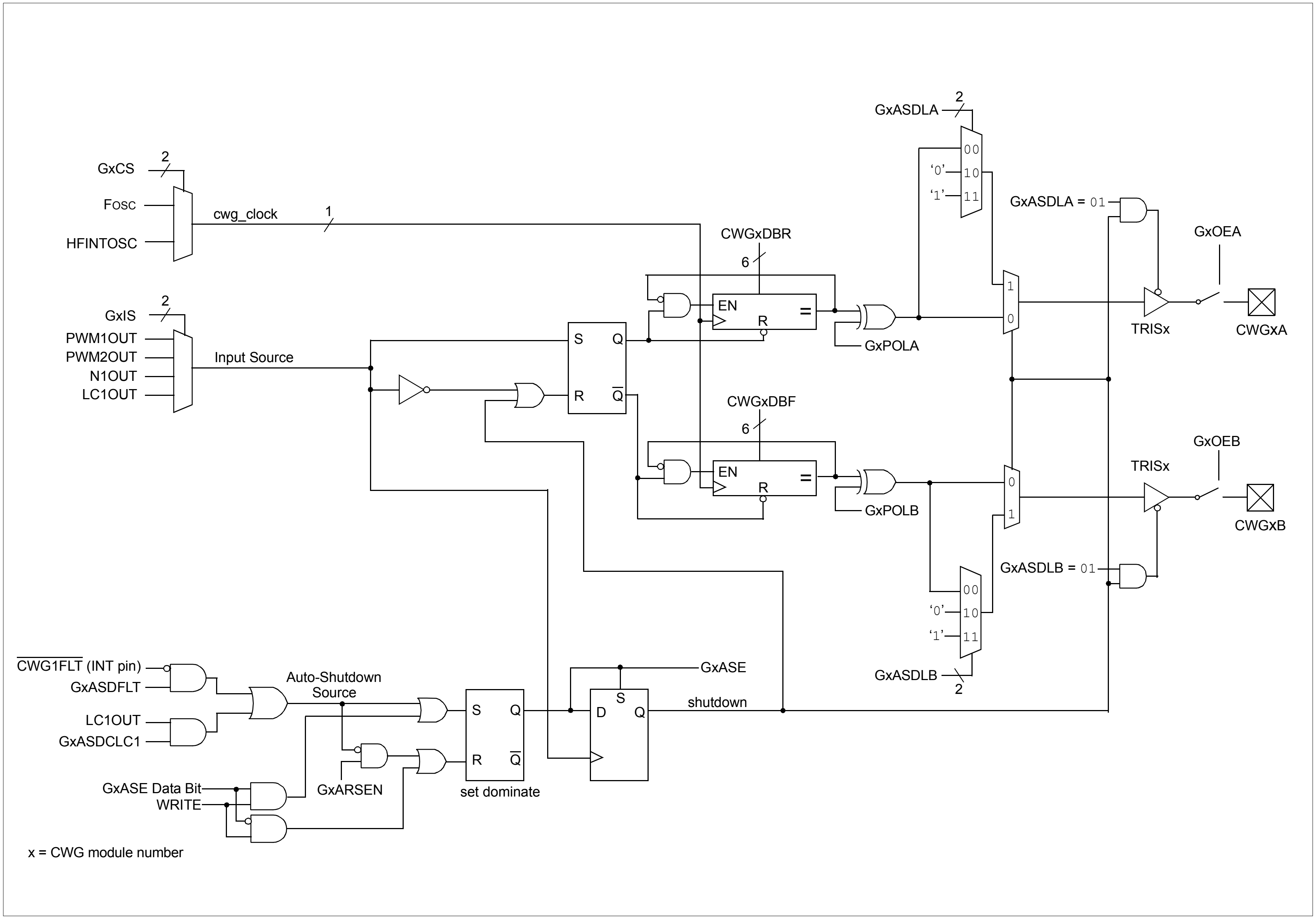
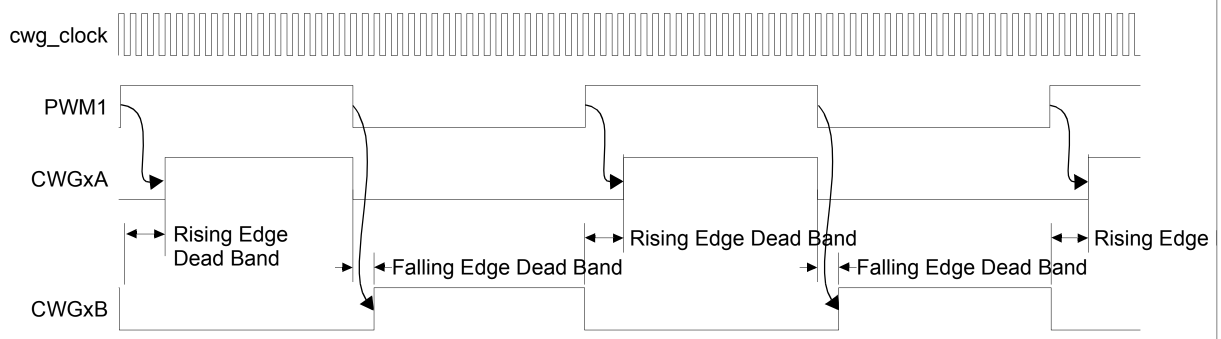


FIGURE 21-2: TYPICAL CWG OPERATION WITH PWM1 (NO AUTO-SHUTDOWN)



21.1 Fundamental Operation

The CWG generates a two output complementary waveform from one of four selectable input sources.

The off-to-on transition of each output can be delayed from the on-to-off transition of the other output, thereby, creating a time delay immediately where neither output is driven. This is referred to as dead time and is covered in **Section 21.5 "Dead-Band Control"**. A typical operating waveform, with dead band, generated from a single input signal is shown in Figure 21-2.

It may be necessary to guard against the possibility of circuit faults or a feedback event arriving too late or not at all. In this case, the active drive must be terminated before the Fault condition causes damage. This is referred to as auto-shutdown and is covered in **Section 21.9 "Auto-shutdown Control"**.

21.2 Clock Source

The CWG module allows the following clock sources to be selected:

- Fosc (system clock)
- HFINTOSC (16 MHz only)

The clock sources are selected using the G1CS0 bit of the CWGxCON0 register (Register 21-1).

21.3 Selectable Input Sources

The CWG can generate the complementary waveform for the following input sources:

- PWM1
- PWM2
- N1OUT
- LC1OUT

The input sources are selected using the GxIS<1:0> bits in the CWGxCON1 register (Register 21-2).

21.4 Output Control

Immediately after the CWG module is enabled, the complementary drive is configured with both CWGxA and CWGxB drives cleared.

21.4.1 OUTPUT ENABLES

Each CWG output pin has individual output enable control. Output enables are selected with the GxOEA and GxOEB bits of the CWGxCON0 register. When an output enable control is cleared, the module asserts no control over the pin. When an output enable is set, the override value or active PWM waveform is applied to the pin per the port priority selection. The output pin enables are dependent on the module enable bit, GxEN. When GxEN is cleared, CWG output enables and CWG drive levels have no effect.

21.4.2 POLARITY CONTROL

The polarity of each CWG output can be selected independently. When the output polarity bit is set, the corresponding output is active high. Clearing the output polarity bit configures the corresponding output as active low. However, polarity does not affect the override levels. Output polarity is selected with the GxPOLA and GxPOLB bits of the CWGxCON0 register.

21.5 Dead-Band Control

Dead-band control provides for non-overlapping output signals to prevent shoot-through current in power switches. The CWG contains two 6-bit dead-band counters. One dead-band counter is used for the rising edge of the input source control. The other is used for the falling edge of the input source control.

Dead band is timed by counting CWG clock periods from zero up to the value in the rising or falling dead-band counter registers. See CWGxDBR and CWGxDBF registers (Register 21-4 and Register 21-5, respectively).

21.6 Rising Edge Dead Band

The rising edge dead band delays the turn-on of the CWGxA output from when the CWGxB output is turned off. The rising edge dead-band time starts when the rising edge of the input source signal goes true. When this happens, the CWGxB output is immediately turned off and the rising edge dead-band delay time starts. When the rising edge dead-band delay time is reached, the CWGxA output is turned on.

The CWGxDBR register sets the duration of the dead-band interval on the rising edge of the input source signal. This duration is from 0 to 64 counts of dead band.

Dead band is always counted off the edge on the input source signal. A count of 0 (zero), indicates that no dead band is present.

If the input source signal is not present for enough time for the count to be completed, no output will be seen on the respective output.

21.7 Falling Edge Dead Band

The falling edge dead band delays the turn-on of the CWGxB output from when the CWGxA output is turned off. The falling edge dead-band time starts when the falling edge of the input source goes true. When this happens, the CWGxA output is immediately turned off and the falling edge dead-band delay time starts. When the falling edge dead-band delay time is reached, the CWGxB output is turned on.

The CWGxDBF register sets the duration of the dead-band interval on the falling edge of the input source signal. This duration is from 0 to 64 counts of dead band.

Dead band is always counted off the edge on the input source signal. A count of 0 (zero), indicates that no dead band is present.

If the input source signal is not present for enough time for the count to be completed, no output will be seen on the respective output.

Refer to Figure 21-3 and Figure 21-4 for examples.

21.8 Dead-Band Uncertainty

When the rising and falling edges of the input source triggers the dead-band counters, the input may be asynchronous. This will create some uncertainty in the dead-band time delay. The maximum uncertainty is equal to one CWG clock period. Refer to Equation 21-1 for more detail.

FIGURE 21-3: DEAD-BAND OPERATION, CWGxDBR = 01H, CWGxDBF = 02H

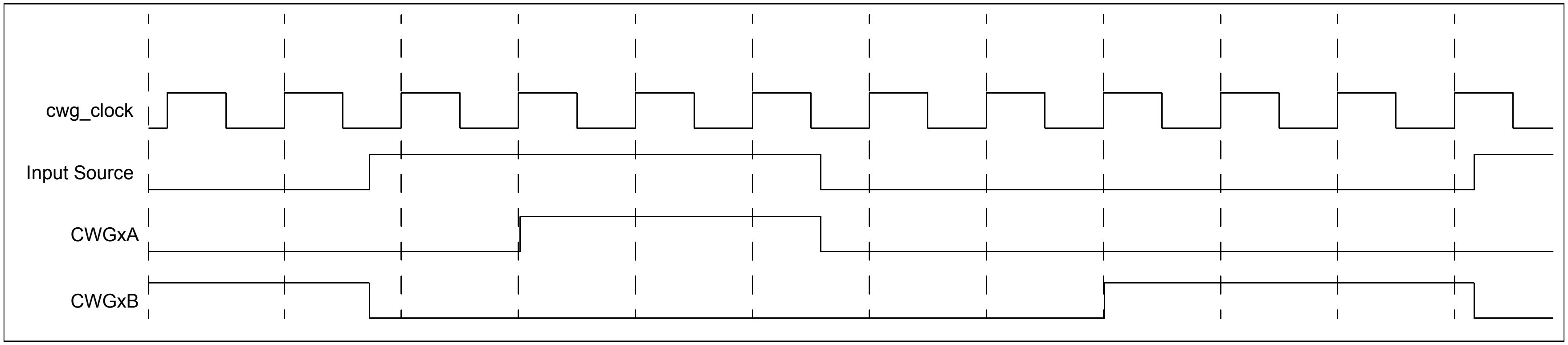
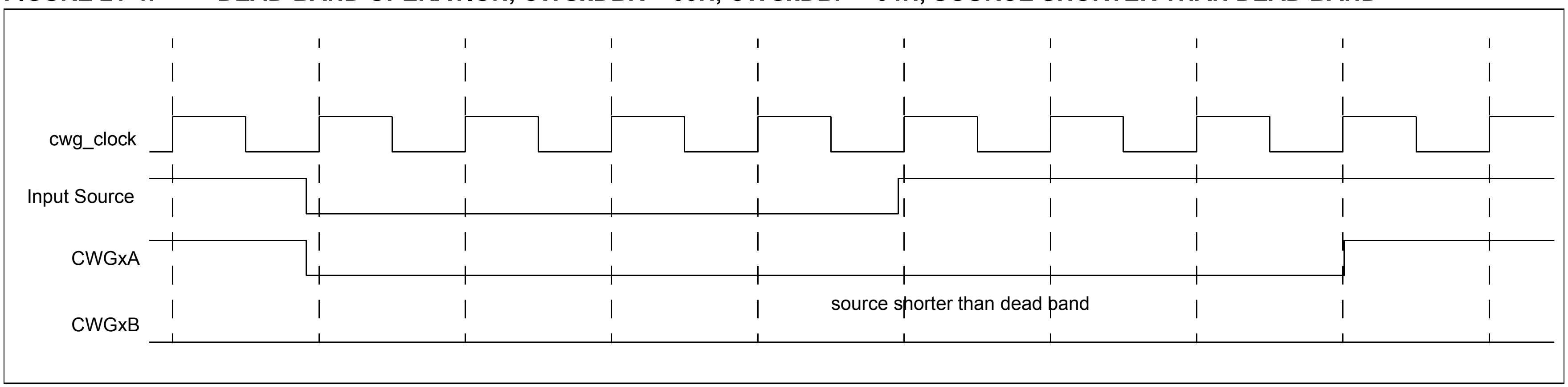


FIGURE 21-4: DEAD-BAND OPERATION, CWGxDBR = 03H, CWGxDBF = 04H, SOURCE SHORTER THAN DEAD BAND



EQUATION 21-1: DEAD-BAND DELAY TIME UNCERTAINTY

$$T_{DEADBAND\_UNCERTAINTY} = \frac{1}{Fcwg\_clock}$$

EXAMPLE 21-1: DEAD-BAND DELAY TIME UNCERTAINTY

$$Fcwg\_clock = 16\ MHz$$

Therefore:

$$T_{DEADBAND\_UNCERTAINTY} = \frac{1}{Fcwg\_clock}$$

$$= \frac{1}{16\ MHz}$$

$$= 625ns$$

21.9 Auto-shutdown Control

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.

21.9.1 SHUTDOWN

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

- Software generated
- External Input

21.9.1.1 Software Generated Shutdown

Setting the GxASE bit of the CWGxCON2 register will force the CWG into the shutdown state.

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

When auto-restart is enabled, the GxASE bit will clear automatically and resume operation on the next rising edge event. See Figure 21-6.

21.9.1.2 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 high, the CWG outputs will immediately go to the selected override levels without software delay. Any combination of two input sources can be selected to cause a shutdown condition. The two sources are:

- LC1OUT
- $\overline{\text{CWG1FLT}}$

Shutdown inputs are selected using the GxASDS0 and GxASDS1 bits of the CWGxCON2 register. (Register 21-3).

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.

21.10 Operation During Sleep

The CWG module operates independently from the system clock and will continue to run during Sleep, provided that the clock and input sources selected remain active.

The HFINTOSC remains active during Sleep, provided that the CWG module is enabled, the input source is active, and the HFINTOSC is selected as the clock source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and the CWG clock source, when the CWG is enabled and the input source is active, the CPU will go idle during Sleep, but the CWG will continue to operate and the HFINTOSC will remain active.

This will have a direct effect on the Sleep mode current.

21.11 Configuring the CWG

The following steps illustrate how to properly configure the CWG to ensure a synchronous start:

1. Ensure that the TRIS control bits corresponding to CWGxA and CWGxB are set so that both are configured as inputs.
2. Clear the GxEN bit, if not already cleared.
3. Set desired dead-band times with the CWGxDBR and CWGxDBF registers.
4. Setup the following controls in CWGxCON2 auto-shutdown register:
   - Select desired shutdown source.
   - Select both output overrides to the desired levels (this is necessary even if not using auto-shutdown because start-up will be from a shutdown state).
   - Set the GxASE bit and clear the GxARSEN bit.
5. Select the desired input source using the CWGxCON1 register.
6. Configure the following controls in CWGxCON0 register:
   - Select desired clock source.
   - Select the desired output polarities.
   - Set the output enables for the outputs to be used.
7. Set the GxEN bit.
8. Clear TRIS control bits corresponding to CWGxA and CWGxB to be used to configure those pins as outputs.
9. If auto-restart is to be used, set the GxARSEN bit and the GxASE bit will be cleared automatically. Otherwise, clear the GxASE bit to start the CWG.

21.11.1 PIN OVERRIDE LEVELS

The levels driven to the output pins, while the shutdown input is true, are controlled by the GxASDLA and GxASDLB bits of the CWGxCON1 register (Register 21-2). GxASDLA controls the CWG1A override level and GxASDLB controls the CWG1B override level. The control bit logic level corresponds to the output logic drive level while in the shutdown state. The polarity control does not apply to the override level.

21.11.2 AUTO-SHUTDOWN RESTART

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

- Software controlled
- Auto-restart

The restart method is selected with the GxARSEN bit of the CWGxCON2 register. Waveforms of software controlled and automatic restarts are shown in Figure 21-5 and Figure 21-6.

21.11.2.1 Software controlled restart

When the GxARSEN bit of the CWGxCON2 register is cleared, the CWG must be restarted after an auto-shutdown event by software.

The CWG will resume operation on the first rising edge event after the GxASE bit is cleared. Clearing the shutdown state requires all selected shutdown inputs to be low, otherwise the GxASE bit will remain set.

21.11.2.2 Auto-Restart

When the GxARSEN bit of the CWGxCON2 register is set, the CWG will restart from the auto-shutdown state automatically.

After the shutdown event clears, the GxASE bit will clear automatically and the CWG will resume operation on the first rising edge event.

FIGURE 21-5: SHUTDOWN FUNCTIONALITY, AUTO-RESTART DISABLED (GxARSEN = 0)

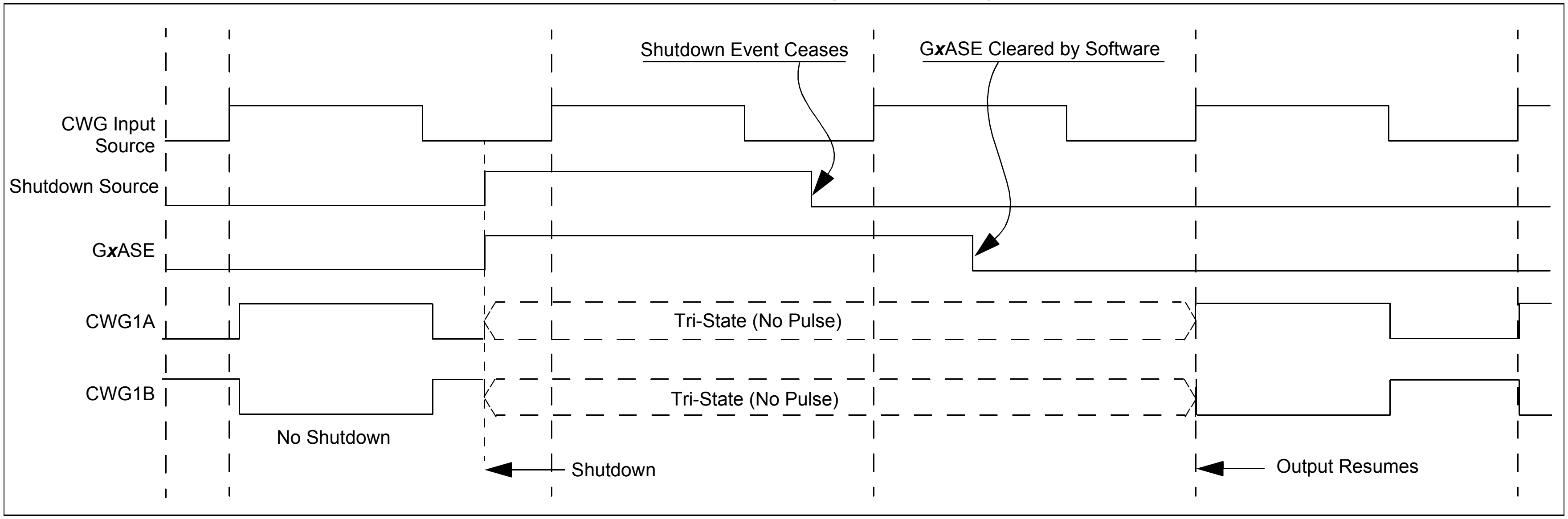
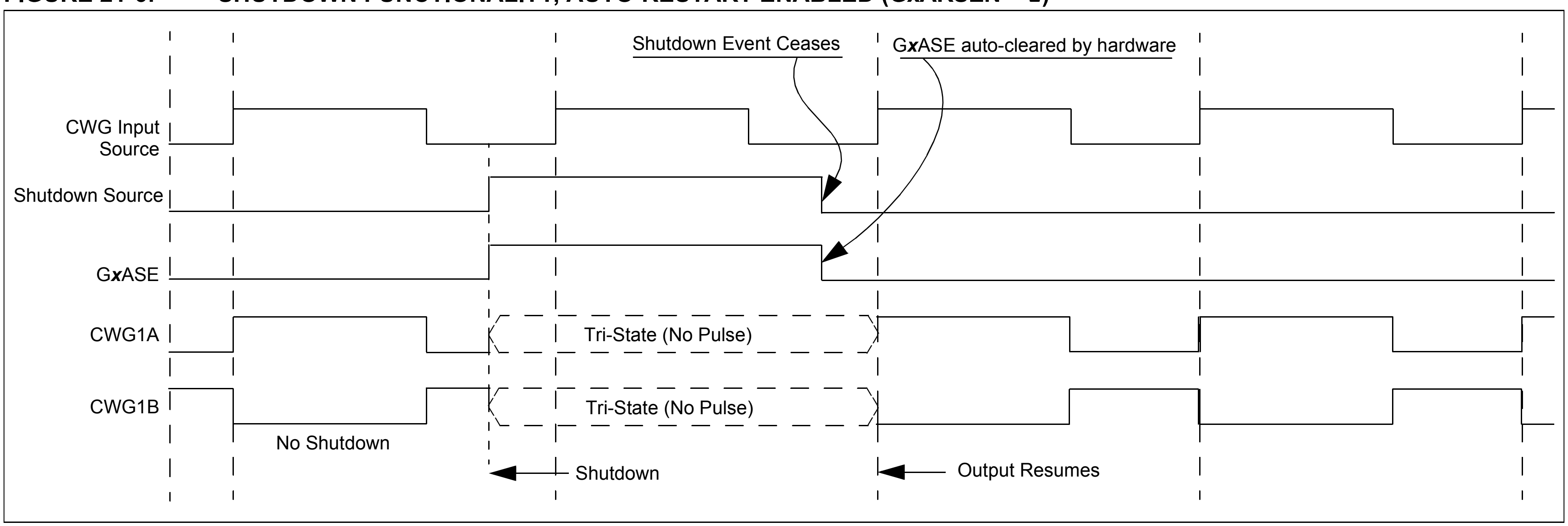


FIGURE 21-6: SHUTDOWN FUNCTIONALITY, AUTO-RESTART ENABLED (GxARSEN = 1)



21.12 CWG Control Registers

REGISTER 21-1: CWGxCON0: CWG CONTROL REGISTER 0

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0
GxEN	GxOEB	GxOEA	GxPOLB	GxPOLA	—	—	GxCS0
bit 7							bit 0

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

bit 7 **GxEN:** CWGx Enable bit
1 = Module is enabled
0 = Module is disabled

bit 6 **GxOEB:** CWGxB Output Enable bit
1 = CWGxB is available on appropriate I/O pin
0 = CWGxB is not available on appropriate I/O pin

bit 5 **GxOEA:** CWGxA Output Enable bit
1 = CWGxA is available on appropriate I/O pin
0 = CWGxA is not available on appropriate I/O pin

bit 4 **GxPOLB:** CWGxB Output Polarity bit
1 = Output is inverted polarity
0 = Output is normal polarity

bit 3 **GxPOLA:** CWGxA Output Polarity bit
1 = Output is inverted polarity
0 = Output is normal polarity

bit 2-1 **Unimplemented**: Read as ‘0’

bit 0 **GxCS0:** CWGx Clock Source Select bit
1 = HFINTOSC
0 = FOSC

REGISTER 21-2: CWGxCON1: CWG CONTROL REGISTER 1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	R/W-0/0	R/W-0/0
GxASDLB<1:0>		GxASDLA<1:0>		—	—	GxIS<1:0>	
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	q = Value depends on condition

bit 7-6 **GxASDLB<1:0>:** CWGx Shutdown State for CWGxB
When an auto shutdown event is present (GxASE = `1`):
`11` = CWGxB pin is driven to ‘`1`’, regardless of the setting of the GxPOLB bit.
`10` = CWGxB pin is driven to ‘`0`’, regardless of the setting of the GxPOLB bit.
`01` = CWGxB pin is tri-stated
`00` = CWGxB pin is driven to its inactive state after the selected dead-band interval. GxPOLB still will control the polarity of the output.

bit 5-4 **GxASDLA<1:0>:** CWGx Shutdown State for CWGxA
When an auto shutdown event is present (GxASE = `1`):
`00` = CWGxA pin is driven to its inactive state after the selected dead-band interval. GxPOLA still will control the polarity of the output.
`01` = CWGxA pin is tri-stated
`10` = CWGxA pin is driven to ‘`0`’, regardless of the setting of the GxPOLA bit.
`11` = CWGxA pin is driven to ‘`1`’, regardless of the setting of the GxPOLA bit.

bit 3-2 **Unimplemented:** Read as ‘`0`’

bit 1-0 **GxIS<1:0>:** CWGx Dead-band Source Select bits
`11` = LC1OUT
`10` = N1OUT
`01` = PWM2OUT
`00` = PWM1OUT

REGISTER 21-3: CWGxCON2: CWG CONTROL REGISTER 2

R/W/HC/HS-0/0	R/W-0/0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
GxASE	GxARSEN	—	—	—	—	GxASDCLC1	GxASDFLT
bit 7							bit 0

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

bit 7 **GxASE:** Auto-Shutdown Event Status bit
1 = An Auto-Shutdown event has occurred. GxOEB/GxOEA Output Controls overridden, Outputs disabled.
0 = No Auto-Shutdown event has occurred, or an Auto-restart has occurred. GxOEB/GxOEA Output Controls enabled.

bit 6 **GxARSEN:** Auto-Restart Enable bit
1 = Auto-restart is enabled
0 = Auto-restart is disabled

bit 5-2 **Unimplemented:** Read as ‘0’

bit 1 **GxASDCLC1:** CWG Auto-shutdown Source Enable bit 1
1 = Shutdown when LC1OUT is high
0 = LC1OUT has no effect on shutdown

bit 0 **GxASDFLT:** CWG Auto-shutdown Source Enable bit 0
1 = Shutdown when $\overline{\text{CWG1FLT}}$ input is low
0 = $\overline{\text{CWG1FLT}}$ input has no effect on shutdown

REGISTER 21-4: CWGxDBR: COMPLEMENTARY WAVEFORM GENERATOR (CWGx) RISING DEAD-BAND COUNT REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	CWG**x**DBR<5:0>					
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	q = Value depends on condition

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

bit 5-0 **CWGxDBR<5:0>:** Complementary Waveform Generator (CWGx) Rising Counts bits
`11 1111` = 63-64 counts of dead band
`11 1110` = 62-63 counts of dead band

•
•
•

`00 0010` = 2-3 counts of dead band
`00 0001` = 1-2 counts of dead band
`00 0000` = 0 counts of dead band

REGISTER 21-5: CWGxDBF: COMPLEMENTARY WAVEFORM GENERATOR (CWGx) FALLING DEAD-BAND COUNT REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	CWGxDBF<5:0>					
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	q = Value depends on condition

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

bit 5-0 **CWGxDBF<5:0>:** Complementary Waveform Generator (CWGx) Falling Counts bits
`11 1111` = 63-64 counts of dead band
`11 1110` = 62-63 counts of dead band

•
•
•

`00 0010` = 2-3 counts of dead band
`00 0001` = 1-2 counts of dead band
`00 0000` = 0 counts of dead band. Dead-band generation is bypassed.

TABLE 21-1: SUMMARY OF REGISTERS ASSOCIATED WITH CWG

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	—	—	ANSA2	ANSA1	ANSA0	70
CWG1CON0	G1EN	G1OEB	G1OEA	G1POLB	G1POLA	—	—	G1CS0	139
CWG1CON1	G1ASDLB<1:0>		G1ASDLA<1:0>		—	—	G1IS<1:0>		140
CWG1CON2	G1ASE	G1ARSEN	—	—	—	—	G1ASDCLC1	G1ASDFLT	141
CWG1DBF	—	—	CWG1DBF<5:0>						142
CWG1DBR	—	—	CWG1DBR<5:0>						142
LATA	—	—	—	—	—	LATA2	LATA1	LATA0	70
TRISA	—	—	—	—	—	TRISA2	TRISA1	TRISA0	69

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by CWG.

22.0 IN-CIRCUIT SERIAL PROGRAMMING™ (ICSP™)

ICSP™ programming allows customers to manufacture circuit boards with unprogrammed devices. Programming can be done after the assembly process allowing the device to be programmed with the most recent firmware or a custom firmware. Five pins are needed for ICSP™ programming:

- ICSPCLK
- ICSPDAT
- $\overline{\text{MCLR}}$/VPP
- VDD
- VSS

In Program/Verify mode the Program Memory, User IDs and the Configuration Words are programmed through serial communications. The ICSPDAT pin is a bidirectional I/O used for transferring the serial data and the ICSPCLK pin is the clock input. For more information on ICSP™ refer to the "*PIC10(L)F320/322 Flash Memory Programming Specification*" (DS41572).

22.1 High-Voltage Programming Entry Mode

The device is placed into High-Voltage Programming Entry mode by holding the ICSPCLK and ICSPDAT pins low then raising the voltage on $\overline{\text{MCLR}}$/VPP to VIHH.

22.2 Low-Voltage Programming Entry Mode

The Low-Voltage Programming Entry mode allows the PIC® Flash MCUs to be programmed using VDD only, without high voltage. When the LVP bit of Configuration Word is set to '1', the low-voltage ICSP programming entry is enabled. To disable the Low-Voltage ICSP mode, the LVP bit must be programmed to '0'.

Entry into the Low-Voltage Programming Entry mode requires the following steps:

1. $\overline{\text{MCLR}}$ is brought to VIL.
2. A 32-bit key sequence is presented on ICSPDAT, while clocking ICSPCLK.

Once the key sequence is complete, $\overline{\text{MCLR}}$ must be held at VIL for as long as Program/Verify mode is to be maintained.

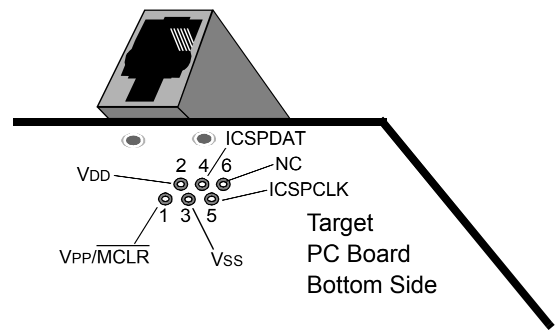
If low-voltage programming is enabled (LVP = 1), the $\overline{\text{MCLR}}$ Reset function is automatically enabled and cannot be disabled. See **Section 5.4 "Low-Power Brown-out Reset (LPBOR)"** for more information.

The LVP bit can only be reprogrammed to '0' by using the High-Voltage Programming mode.

22.3 Common Programming Interfaces

Connection to a target device is typically done through an ICSP™ header. A commonly found connector on development tools is the RJ-11 in the 6P6C (6 pin, 6 connector) configuration. See Figure 22-1.

FIGURE 22-1: ICD RJ-11 STYLE CONNECTOR INTERFACE



Pin Description*

1 = VPP/$\overline{\text{MCLR}}$

2 = VDD Target

3 = VSS (ground)

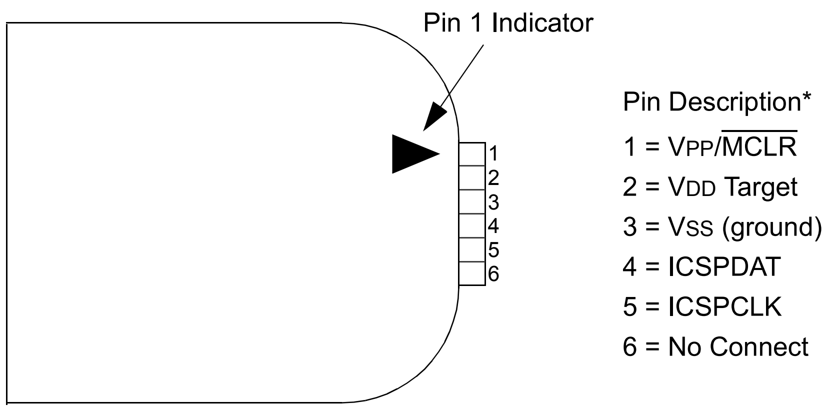
4 = ICSPDAT

5 = ICSPCLK

6 = No Connect

Another connector often found in use with the PICkit™ programmers is a standard 6-pin header with 0.1 inch spacing. Refer to Figure 22-2.

FIGURE 22-2: PICkit™ STYLE CONNECTOR INTERFACE

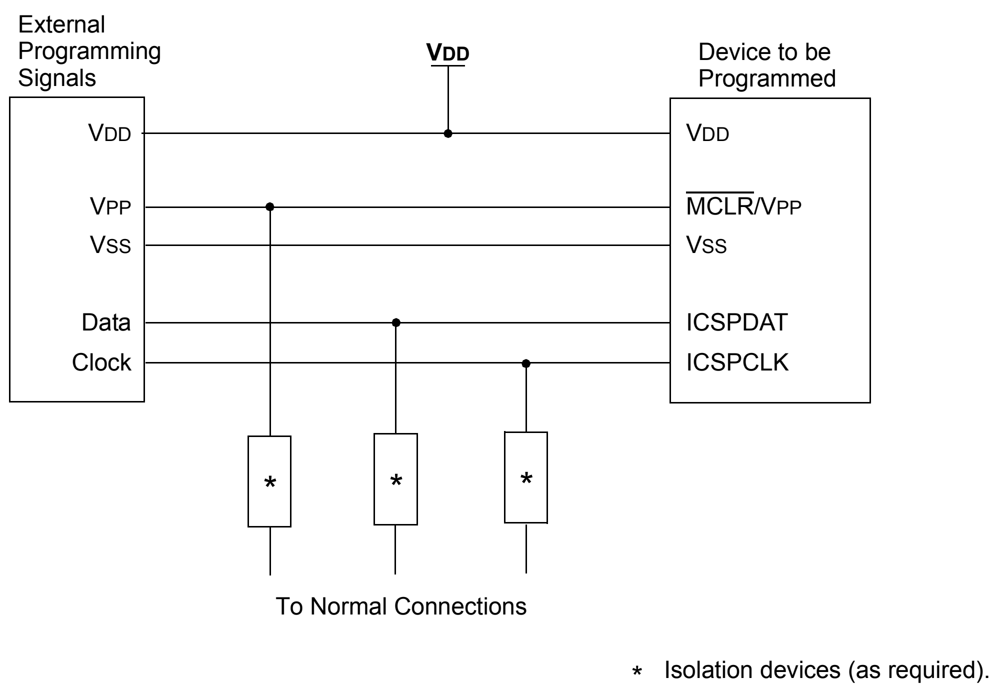


* The 6-pin header (0.100" spacing) accepts 0.025" square pins.

For additional interface recommendations, refer to your specific device programmer manual prior to PCB design.

It is recommended that isolation devices be used to separate the programming pins from other circuitry. The type of isolation is highly dependent on the specific application and may include devices such as resistors, diodes, or even jumpers. See Figure 22-3 for more information.

FIGURE 22-3: TYPICAL CONNECTION FOR ICSP™ PROGRAMMING



23.0 INSTRUCTION SET SUMMARY

The PIC10(L)F320/322 instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 instruction is a 14-bit word divided into an **opcode,** which specifies the instruction type and one or more **operands,** which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 23-1, while the various opcode fields are summarized in Table 23-1.

Table 23-2 lists the instructions recognized by the MPASM™ assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a normal instruction execution time of 1 µs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a `NOP`.

All instruction examples use the format '`0xhh`' to represent a hexadecimal number, where '`h`' signifies a hexadecimal digit.

23.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (RMW) operation. The register is read, the data is modified, and the result is stored according to either the instruction or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a `CLRF PORTA` instruction will read PORTA, clear all the data bits, then write the result back to PORTA. This example would have the unintended consequence of clearing the condition that set the IOCIF flag.

TABLE 23-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= `0` or `1`). The assembler will generate code with x = `0`. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = `0`: store result in W, d = `1`: store result in file register f. Default is d = `1`.
PC	Program Counter
$\overline{TO}$	Time-out bit
C	Carry bit
DC	Digit carry bit
Z	Zero bit
$\overline{PD}$	Power-down bit

FIGURE 23-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations

13 8	7	6 0
OPCODE	d	f (FILE #)

d = `0` for destination W
d = `1` for destination f
f = 7-bit file register address

Bit-oriented file register operations

13 10	9 7	6 0
OPCODE	b (BIT #)	f (FILE #)

b = 3-bit bit address
f = 7-bit file register address

Literal and control operations

General

13 8	7 0
OPCODE	k (literal)

k = 8-bit immediate value

`CALL` and `GOTO` instructions only

13 11	10 0
OPCODE	k (literal)

k = 11-bit immediate value

TABLE 23-2: INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb			LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	**1, 2**
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	**1, 2**
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	**2**
CLRW	–	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	**1, 2**
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	**1, 2**
DECFSZ	f, d	Decrement f, Skip if 0	1**(2)**	00	1011	dfff	ffff		**1, 2, 3**
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	**1, 2**
INCFSZ	f, d	Increment f, Skip if 0	1**(2)**	00	1111	dfff	ffff		**1, 2, 3**
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	**1, 2**
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	**1, 2**
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	–	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	**1, 2**
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	**1, 2**
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	**1, 2**
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		**1, 2**
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	**1, 2**
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		**1, 2**
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		**1, 2**
BTFSC	f, b	Bit Test f, Skip if Clear	1 **(2)**	01	10bb	bfff	ffff		**3**
BTFSS	f, b	Bit Test f, Skip if Set	1 **(2)**	01	11bb	bfff	ffff		**3**
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	–	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$, $\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	–	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	–	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	–	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}$, $\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTA, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is ‘1’ for a pin configured as input and is driven low by an external device, the data will be written back with a ‘0’.

2: If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.

3: If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

23.2 Instruction Descriptions

ADDLW	Add literal and W
Syntax:	[*label*] ADDLW k
Operands:	$0 \leq k \leq 255$
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the eight-bit literal ‘k’ and the result is placed in the W register.

ADDWF	Add W and f
Syntax:	[*label*] ADDWF f,d
Operands:	$0 \leq f \leq 127$ $d \in$ [0,1]
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

ANDLW	AND literal with W
Syntax:	[*label*] ANDLW k
Operands:	$0 \leq k \leq 255$
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND’ed with the eight-bit literal ‘k’. The result is placed in the W register.

ANDWF	AND W with f
Syntax:	[*label*] ANDWF f,d
Operands:	$0 \leq f \leq 127$ $d \in$ [0,1]
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND the W register with register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

BCF	Bit Clear f
Syntax:	[*label*] BCF f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	0 → (f<b>)
Status Affected:	None
Description:	Bit ‘b’ in register ‘f’ is cleared.

BSF	Bit Set f
Syntax:	[*label*] BSF f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	1 → (f<b>)
Status Affected:	None
Description:	Bit ‘b’ in register ‘f’ is set.

BTFSC	Bit Test f, Skip if Clear
Syntax:	[*label*] BTFSC f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit ‘b’ in register ‘f’ is ‘1’, the next instruction is executed. If bit ‘b’ in register ‘f’ is ‘0’, the next instruction is discarded, and a NOP is executed instead, making this a two-cycle instruction.

BTFSS	**Bit Test f, Skip if Set**
Syntax:	[*label*] BTFSS f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b < 7$
Operation:	skip if (f<b>) = `1`
Status Affected:	None
Description:	If bit ‘b’ in register ‘f’ is ‘`0`’, the next instruction is executed. If bit ‘b’ is ‘`1`’, then the next instruction is discarded and a `NOP` is executed instead, making this a two-cycle instruction.

CALL	**Call Subroutine**
Syntax:	[*label*] CALL k
Operands:	$0 \leq k \leq 2047$
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. `CALL` is a two-cycle instruction.

CLRF	**Clear f**
Syntax:	[*label*] CLRF f
Operands:	$0 \leq f \leq 127$
Operation:	00h → (f) `1` → Z
Status Affected:	Z
Description:	The contents of register ‘f’ are cleared and the Z bit is set.

CLRW	**Clear W**
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) `1` → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CLRWDT	**Clear Watchdog Timer**
Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT `0` → WDT prescaler, `1` → $\overline{TO}$ `1` → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	`CLRWDT` instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

COMF	**Complement f**
Syntax:	[*label*] COMF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(\bar{f})$ → (destination)
Status Affected:	Z
Description:	The contents of register ‘f’ are complemented. If ‘d’ is ‘`0`’, the result is stored in W. If ‘d’ is ‘`1`’, the result is stored back in register ‘f’.

DECF	**Decrement f**
Syntax:	[*label*] DECF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register ‘f’. If ‘d’ is ‘`0`’, the result is stored in the W register. If ‘d’ is ‘`1`’, the result is stored back in register ‘f’.

DECFSZ	Decrement f, Skip if 0
Syntax:	[*label*] DECFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) - 1 → (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register ‘f’ are decremented. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed back in register ‘f’. If the result is ‘1’, the next instruction is executed. If the result is ‘0’, then a NOP is executed instead, making it a two-cycle instruction.

GOTO	Unconditional Branch
Syntax:	[*label*] GOTO k
Operands:	$0 \le k \le 2047$
Operation:	k → PC<10:0> PCLATH<4:3> → PC<12:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

INCF	Increment f
Syntax:	[*label*] INCF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 → (destination)
Status Affected:	Z
Description:	The contents of register ‘f’ are incremented. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed back in register ‘f’.

INCFSZ	Increment f, Skip if 0
Syntax:	[*label*] INCFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 → (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register ‘f’ are incremented. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed back in register ‘f’. If the result is ‘1’, the next instruction is executed. If the result is ‘0’, a NOP is executed instead, making it a two-cycle instruction.

IORLW	Inclusive OR literal with W
Syntax:	[*label*] IORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are OR’ed with the eight-bit literal ‘k’. The result is placed in the W register.

IORWF	Inclusive OR W with f
Syntax:	[*label*] IORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .OR. (f) → (destination)
Status Affected:	Z
Description:	Inclusive OR the W register with register ‘f’. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed back in register ‘f’.

MOVF	Move f
Syntax:	[*label*] MOVF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) → (dest)
Status Affected:	Z
Description:	The contents of register 'f' is moved to a destination dependent upon the status of 'd'. If d = 0, destination is W register. If d = 1, the destination is file register 'f' itself. d = 1 is useful to test a file register since Status flag Z is affected.
Words:	1
Cycles:	1
Example:	`MOVF FSR, 0` After Instruction W = value in FSR register Z = 1

MOVWF	Move W to f
Syntax:	[*label*] MOVWF f
Operands:	0 ≤ f ≤ 127
Operation:	(W) → (f)
Status Affected:	None
Description:	Move data from W register to register 'f'.
Words:	1
Cycles:	1
Example:	`MOVW OPTION_REG F` Before Instruction OPTION_REG = 0xFF W = 0x4F After Instruction OPTION_REG = 0x4F W = 0x4F

MOVLW	Move literal to W
Syntax:	[*label*] MOVLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W)
Status Affected:	None
Description:	The eight-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.
Words:	1
Cycles:	1
Example:	`MOVLW 0x5A` After Instruction W = 0x5A

NOP	No Operation
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.
Words:	1
Cycles:	1
Example:	`NOP`

RETFIE	Return from Interrupt
Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None
Description:	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction.
Words:	1
Cycles:	2
Example:	`RETFIE`

After Interrupt
PC = TOS
GIE = 1

RETLW	Return with literal in W
Syntax:	[*label*] RETLW k
Operands:	$0 \leq k \leq 255$
Operation:	k → (W); TOS → PC
Status Affected:	None
Description:	The W register is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.
Words:	1
Cycles:	2

Example:

```
        CALL TABLE;W contains
                  ;table offset
                  ;value
        GOTO DONE
TABLE   •
        •
        ADDWF PC  ;W = offset
        RETLW k1  ;Begin table
        RETLW k2  ;
        •
        •
        •
        RETLW kn  ;End of table
DONE
```

Before Instruction
W = 0x07
After Instruction
W = value of k8

RETURN	Return from Subroutine
Syntax:	[*label*] RETURN
Operands:	None
Operation:	TOS → PC
Status Affected:	None
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

RLF — Rotate Left f through Carry

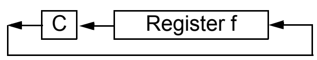
Syntax: [*label*] RLF f,d

Operands: $0 \leq f \leq 127$
$d \in [0,1]$

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is stored back in register 'f'.



Words: 1

Cycles: 1

<u>Example:</u>

```
RLF     REG1,0
```

Before Instruction

```
REG1   =   1110 0110
C      =   0
```

After Instruction

```
REG1   =   1110 0110
W      =   1100 1100
C      =   1
```

RRF — Rotate Right f through Carry

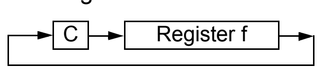
Syntax: [*label*] RRF f,d

Operands: $0 \leq f \leq 127$
$d \in [0,1]$

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.



SLEEP — Enter Sleep mode

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h $\rightarrow$ WDT,
0 $\rightarrow$ WDT prescaler,
1 $\rightarrow$ $\overline{TO}$,
0 $\rightarrow$ $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Description: The power-down Status bit, $\overline{PD}$ is cleared. Time-out Status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.
The processor is put into Sleep mode with the oscillator stopped.

SUBLW — Subtract W from literal

Syntax: [*label*] SUBLW k

Operands: $0 \leq k \leq 255$

Operation: k - (W) $\rightarrow$ (W)

Status Affected: C, DC, Z

Description: The W register is subtracted (2's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

Result	Condition
C = 0	W > k
C = 1	W $\leq$ k
DC = 0	W<3:0> > k<3:0>
DC = 1	W<3:0> $\leq$ k<3:0>

SUBWF	Subtract W from f
Syntax:	[*label*] SUBWF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) - (W) $\rightarrow$ (destination)
Status Affected:	C, DC, Z
Description:	Subtract (2's complement method) W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

C = 0	W > f
C = 1	W ≤ f
DC = 0	W<3:0> > f<3:0>
DC = 1	W<3:0> ≤ f<3:0>

SWAPF	Swap Nibbles in f
Syntax:	[*label*] SWAPF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f<3:0>) $\rightarrow$ (destination<7:4>), (f<7:4>) $\rightarrow$ (destination<3:0>)
Status Affected:	None
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in register 'f'.

XORLW	Exclusive OR literal with W
Syntax:	[*label*] XORLW k
Operands:	$0 \leq k \leq 255$
Operation:	(W) .XOR. k $\rightarrow$ (W)
Status Affected:	Z
Description:	The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.

XORWF	Exclusive OR W with f
Syntax:	[*label*] XORWF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(W) .XOR. (f) $\rightarrow$ (destination)
Status Affected:	Z
Description:	Exclusive OR the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

24.0 ELECTRICAL SPECIFICATIONS

24.1 Absolute Maximum Ratings(†)

Ambient temperature under bias .... -40°C to +125°C
Storage temperature .... -65°C to +150°C
Voltage on pins with respect to VSS
  on VDD pin
    PIC10F320/322 .... -0.3V to +6.5V
    PIC10LF320/322 .... -0.3V to +4.0V
  on $\overline{MCLR}$ pin .... -0.3V to +9.0V
  on all other pins .... -0.3V to (VDD + 0.3V)
Maximum current
  on VSS pin(1)
    -40°C ≤ TA ≤ +85°C .... 150 mA
    -40°C ≤ TA ≤ +125°C .... 100 mA
  on VDD pin(1)
    -40°C ≤ TA ≤ +85°C .... 150 mA
    -40°C ≤ TA ≤ +125°C .... 100 mA
  on any I/O pin .... ±25 mA
Clamp current, IK (VPIN < 0 or VPIN > VDD) .... ±20 mA

Note 1: Maximum current rating requires even load distribution across I/O pins. Maximum current rating may be limited by the device package power dissipation characterizations, see **Section 24.4 "Thermal Considerations"** to calculate device specifications.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure above maximum rating conditions for extended periods may affect device reliability.

24.2 Standard Operating Conditions

The standard operating conditions for any device are defined as:

Operating Voltage: $V_{DDMIN} \le V_{DD} \le V_{DDMAX}$
Operating Temperature: $T_{A\_MIN} \le T_A \le T_{A\_MAX}$

VDD — Operating Supply Voltage[1]

PIC10LF320/322

- VDDMIN (Fosc ≤ 16 MHz) .... +1.8V
- VDDMIN (16 MHz < Fosc ≤ 20 MHz) .... +2.5V
- VDDMAX .... +3.6V

PIC10F320/322

- VDDMIN (Fosc ≤ 16 MHz) .... +2.3V
- VDDMIN (16 MHz < Fosc ≤ 20 MHz) .... +2.5V
- VDDMAX .... +5.5V

TA — Operating Ambient Temperature Range

Industrial Temperature

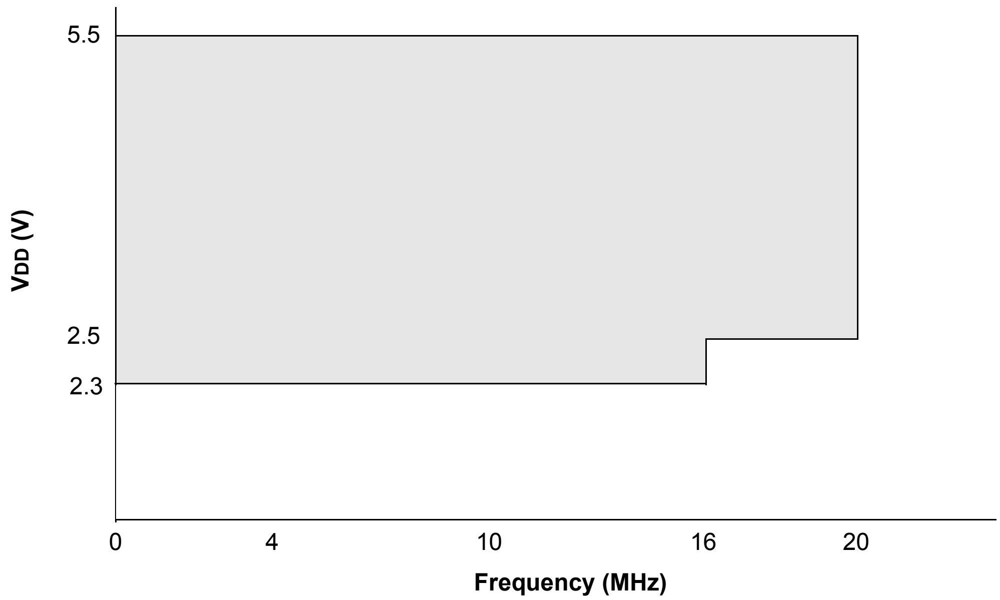
- TA_MIN .... -40°C
- TA_MAX .... +85°C

Extended Temperature

- TA_MIN .... -40°C
- TA_MAX .... +125°C

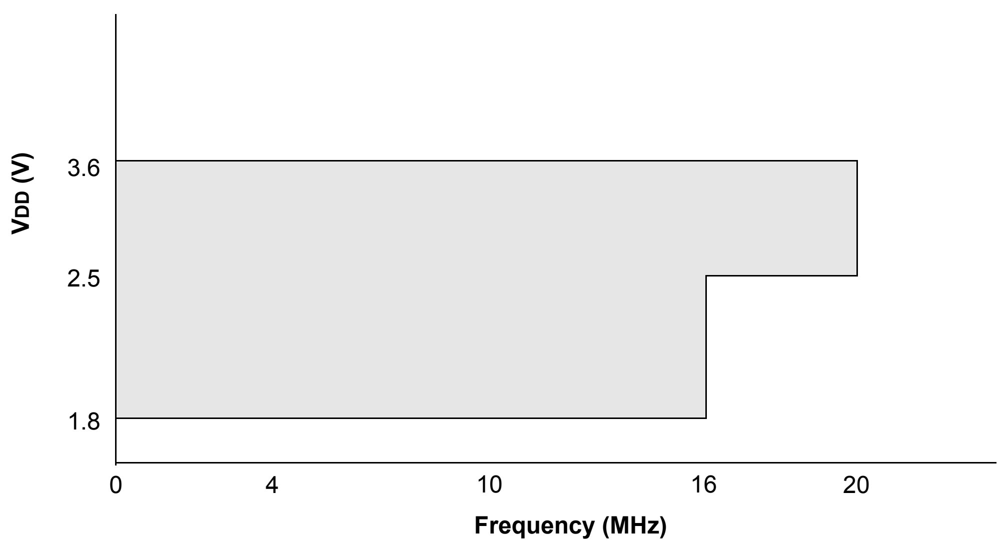
Note 1: See Parameter D001, DC Characteristics: Supply Voltage.

FIGURE 24-1: PIC10F320/322 VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.
2: Refer to Table 24-6 for each Oscillator mode's supported frequencies.

FIGURE 24-2: PIC10LF320/322 VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.
2: Refer to Table 24-6 for each Oscillator mode's supported frequencies.

24.3 DC Characteristics

TABLE 24-1: SUPPLY VOLTAGE

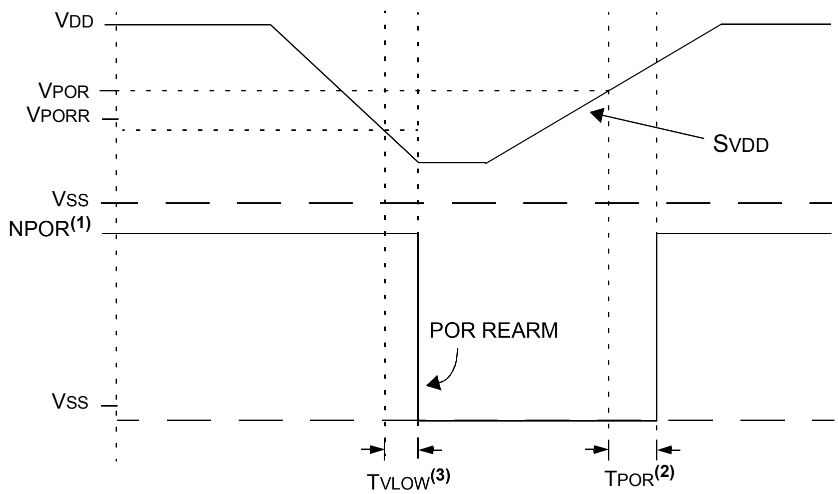
PIC10LF320/322			Standard Operating Conditions (unless otherwise stated)				
PIC10F320/322							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
D001	VDD	**Supply Voltage**					
			1.8 2.5	— —	3.6 3.6	V V	FOSC ≤ 16 MHz: FOSC ≤ 20 MHz
D001			2.3 2.5	— —	5.5 5.5	V V	FOSC ≤ 16 MHz: FOSC ≤ 20 MHz
D002*	VDR	**RAM Data Retention Voltage(1)**					
			1.5	—	—	V	Device in Sleep mode
D002*			1.7	—	—	V	Device in Sleep mode
	VPOR*	**Power-on Reset Release Voltage**	—	1.6	—	V	
	VPORR*	**Power-on Reset Rearm Voltage**	—	0.8	—	V	Device in Sleep mode
			—	1.7	—	V	Device in Sleep mode
D004*	SVDD	**VDD Rise Rate** to ensure internal Power-on Reset signal	0.05	—	—	V/ms	See **Section 5.1 "Power-On Reset (POR)"** for details.

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

FIGURE 24-3: POR AND POR REARM WITH SLOW RISING VDD



Note 1: When NPOR is low, the device is held in Reset.
2: TPOR 1 μs typical.
3: TVLOW 2.7 μs typical.

TABLE 24-2: SUPPLY VOLTAGE (I_{DD})(1,2)

PIC10LF320/322		Standard Operating Conditions (unless otherwise stated)					
PIC10F320/322							
Param No.	Device Characteristics	Min.	Typ†	Max.	Units	Conditions	
						VDD	Note
D013		—	34	—	μA	1.8	FOSC = 500 kHz EC mode
		—	60	—	μA	3.0	
D013		—	76	—	μA	2.3	FOSC = 500 kHz EC mode
		—	110	—	μA	3.0	
		—	153	—	μA	5.0	
D014		—	200	396	μA	1.8	FOSC = 8 MHz EC mode
		—	380	650	μA	3.0	
D014		—	302	430	μA	2.3	FOSC = 8 MHz EC mode
		—	420	655	μA	3.0	
		—	502	775	μA	5.0	
D015		—	0.8	1.3	mA	3.0	FOSC = 20 MHz EC mode
		—	1.1	1.8	mA	3.6	
D015		—	0.8	1.4	mA	3.0	FOSC = 20 MHz EC mode
		—	1.1	1.8	mA	5.0	
D016		—	2.4	—	μA	1.8	FOSC = 32 kHz LFINTOSC mode, 85°C
		—	7	—	μA	3.0	
D016		—	31	—	μA	2.3	FOSC = 32 kHz LFINTOSC mode, 85°C
		—	40	—	μA	3.0	
		—	71	—	μA	5.0	
D016A		—	2.4	—	μA	1.8	FOSC = 32 kHz LFINTOSC mode, 125°C
		—	7	—	μA	3.0	
D016A		—	31	—	μA	2.3	FOSC = 32 kHz LFINTOSC mode,125°C
		—	40	—	μA	3.0	
		—	71	—	μA	5.0	

Note 1: The test conditions for all IDD measurements in active operation mode are: CLKIN = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{\text{MCLR}}$ = VDD; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

TABLE 24-2: SUPPLY VOLTAGE (IDD)(1,2) (CONTINUED)

PIC10LF320/322		Standard Operating Conditions (unless otherwise stated)					
PIC10F320/322							
Param No.	Device Characteristics	Min.	Typ†	Max.	Units	Conditions	
						VDD	Note
D017		—	293	—	μA	1.8	FOSC = 500 kHz HFINTOSC mode
		—	286	—	μA	3.0	
D017		—	272	—	μA	2.3	FOSC = 500 kHz HFINTOSC mode
		—	310	—	μA	3.0	
		—	372	—	μA	5.0	
D018		—	350	1.0	mA	1.8	FOSC = 8 MHz HFINTOSC mode
		—	530	1.1	mA	3.0	
D018		—	0.45	1.0	mA	2.3	FOSC = 8 MHz HFINTOSC mode
		—	0.56	1.1	mA	3.0	
		—	0.64	1.2	mA	5.0	
D019		—	0.46	1.1	mA	1.8	FOSC = 16 MHz HFINTOSC mode
		—	0.73	1.2	mA	3.0	
D019		—	0.60	1.1	mA	2.3	FOSC = 16 MHz HFINTOSC mode
		—	0.76	1.2	mA	3.0	
		—	0.85	1.3	mA	5.0	

Note 1: The test conditions for all IDD measurements in active operation mode are: CLKIN = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{\text{MCLR}}$ = VDD; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

TABLE 24-3: POWER-DOWN CURRENTS (IPD)[1,2]

PIC10LF320/322		Standard Operating Conditions (unless otherwise stated)						
PIC10F320/322								
Param No.	Device Characteristics	Min.	Typ†	Max. +85°C	Max. +125°C	Units	Conditions: VDD	Conditions: Note
D023		—	0.06	1.1	2	μA	1.8	WDT, BOR, and FVR disabled, all Peripherals Inactive
		—	0.08	1.3	2	μA	3.0	
D023		—	0.20	1.1	2	μA	2.3	WDT, BOR, and FVR disabled, all Peripherals Inactive
		—	0.30	1.4	2	μA	3.0	
		—	0.40	2.4	2.4	μA	5.0	
D024		—	0.5	9	11	μA	1.8	WDT Current **(Note 1)**
		—	0.8	11	13	μA	3.0	
D024		—	4.0	10	12	μA	2.3	WDT Current **(Note 1)**
		—	4.2	12	14	μA	3.0	
		—	4.3	14	16	μA	5.0	
D025		—	30	96	120	μA	1.8	FVR current
		—	39	106	123	μA	3.0	
D025		—	32	96	120	μA	2.3	FVR current
		—	39	106	133	μA	3.0	
		—	70	136	170	μA	5.0	
D026		—	7.5	16	18	μA	3.0	BOR Current **(Note 1)**
D026		—	8	18	20	μA	3.0	BOR Current **(Note 1)**
		—	9	20	20.2	μA	5.0	
D026A		—	2.7	10	15	μA	3.0	LPBOR Current
D026A		—	3.0	10	15	μA	3.0	LPBOR Current
		—	3.2	15	20	μA	5.0	
D028		—	0.1	4	5	μA	1.8	A/D Current **(Note 1, Note 3)**, no conversion in progress
		—	0.1	5	6	μA	3.0	
D028		—	3.4	6	7	μA	2.3	A/D Current **(Note 1, Note 3)**, no conversion in progress
		—	3.6	7	8	μA	3.0	
		—	3.8	8	9	μA	5.0	
D029		—	250	—	—	μA	1.8	A/D Current **(Note 1, Note 3)**, conversion in progress
		—	250	—	—	μA	3.0	
D029		—	280	—	—	μA	2.3	A/D Current **(Note 1, Note 3)**, conversion in progress
		—	280	—	—	μA	3.0	
		—	280	—	—	μA	5.0	

* 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: The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

2: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

3: A/D oscillator source is FRC.

TABLE 24-4: I/O PORTS

Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
Standard Operating Conditions (unless otherwise stated)							
	V_{IL}	**Input Low Voltage**					
		I/O PORT:					
D032		with TTL buffer	—	—	0.8	V	$4.5V \le V_{DD} \le 5.5V$
D032A			—	—	0.15 V_{DD}	V	$1.8V \le V_{DD} \le 4.5V$
D033		with Schmitt Trigger buffer	—	—	0.2 V_{DD}	V	$2.0V \le V_{DD} \le 5.5V$
D034		$\overline{MCLR}$	—	—	0.2 V_{DD}	V	
	V_{IH}	**Input High Voltage**					
		I/O ports:					
D040		with TTL buffer	2.0	—	—	V	$4.5V \le V_{DD} \le 5.5V$
D040A			0.25 V_{DD} + 0.8	—	—	V	$1.8V \le V_{DD} \le 4.5V$
D041		with Schmitt Trigger buffer	0.8 V_{DD}	—	—	V	$2.0V \le V_{DD} \le 5.5V$
D042		$\overline{MCLR}$	0.8 V_{DD}	—	—	V	
	I_{IL}	**Input Leakage Current**(2)					
D060		I/O ports	—	± 5 ± 5	± 125 ± 1000	nA nA	$V_{SS} \le V_{PIN} \le V_{DD}$, Pin at high-impedance @ 85°C 125°C
D061		$\overline{MCLR}$	—	± 50	± 200	nA	$V_{SS} \le V_{PIN} \le V_{DD}$ @ 85°C
	I_{PUR}	**Weak Pull-up Current**					
D070*			25 25	100 140	200 300	μA	V_{DD} = 3.3V, V_{PIN} = V_{SS} V_{DD} = 5.0V, V_{PIN} = V_{SS}
	V_{OL}	**Output Low Voltage**					
D080		I/O ports	—	—	0.6	V	I_{OL} = 8mA, V_{DD} = 5V I_{OL} = 6mA, V_{DD} = 3.3V I_{OL} = 1.8mA, V_{DD} = 1.8V
	V_{OH}	**Output High Voltage**					
D090		I/O ports	V_{DD} - 0.7	—	—	V	I_{OH} = 3.5mA, V_{DD} = 5V I_{OH} = 3mA, V_{DD} = 3.3V I_{OH} = 1mA, V_{DD} = 1.8V

* 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: Negative current is defined as current sourced by the pin.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

TABLE 24-5: MEMORY PROGRAMMING REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
		Program Memory Programming Specifications					
D110	VIHH	Voltage on $\overline{MCLR}$/VPP pin	8.0	—	9.0	V	**(Note 2)**
D111	IDDP	Supply Current during Programming	—	—	10	mA	
D112		VDD for Bulk Erase	2.7	—	VDD max.	V	
D113	VPEW	VDD for Write or Row Erase	VDD min.	—	VDD max.	V	
D114	IPPPGM	Current on $\overline{MCLR}$/VPP during Erase/Write	—	—	1.0	mA	
D115	IDDPGM	Current on VDD during Erase/Write	—		5.0	mA	
		Program Flash Memory					
D121	EP	Cell Endurance	10K	—	—	E/W	-40°C to +85°C (**Note 1**)
D122	VPR	VDD for Read	VDD min.	—	VDD max.	V	
D123	TIW	Self-timed Write Cycle Time	—	2	2.5	ms	
D124	TRETD	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated

† 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: Self-write and Block Erase.

2: Required only if single-supply programming is disabled.

24.4 Thermal Considerations

Standard Operating Conditions (unless otherwise stated)					
Param No.	Sym.	Characteristic	Typ.	Units	Conditions
TH01	θ_{JA}	Thermal Resistance Junction to Ambient	60	°C/W	6-pin SOT-23 package
			80	°C/W	8-pin PDIP package
			90	°C/W	8-pin DFN package
TH02	θ_{JC}	Thermal Resistance Junction to Case	31.4	°C/W	6-pin SOT-23 package
			24	°C/W	8-pin PDIP package
			24	°C/W	8-pin DFN package
TH03	T_{JMAX}	Maximum Junction Temperature	150	°C	
TH04	PD	Power Dissipation	—	W	$PD = P_{INTERNAL} + P_{I/O}$
TH05	$P_{INTERNAL}$	Internal Power Dissipation	—	W	$P_{INTERNAL} = I_{DD} \times V_{DD}$(1)
TH06	$P_{I/O}$	I/O Power Dissipation	—	W	$P_{I/O} = \Sigma (I_{OL} * V_{OL}) + \Sigma (I_{OH} * (V_{DD} - V_{OH}))$
TH07	P_{DER}	Derated Power	—	W	$P_{DER} = PD_{MAX} (T_J - T_A)/\theta_{JA}$(2)

Note 1: I_{DD} is current to run the chip alone without driving any load on the output pins.
2: T_A = Ambient Temperature
3: T_J = Junction Temperature

24.5 AC Characteristics

Timing Parameter Symbology has been created with one of the following formats:

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

Lowercase letters (pp) and their meanings:

pp			
cc	CCP1	osc	CLKIN
ck	CLKR	rd	$\overline{\text{RD}}$
cs	$\overline{\text{CS}}$	rw	$\overline{\text{RD}}$ or $\overline{\text{WR}}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{\text{SS}}$
dt	Data in	t0	T0CKI
io	I/O PORT	t1	T1CKI
mc	$\overline{\text{MCLR}}$	wr	$\overline{\text{WR}}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance

FIGURE 24-4: LOAD CONDITIONS

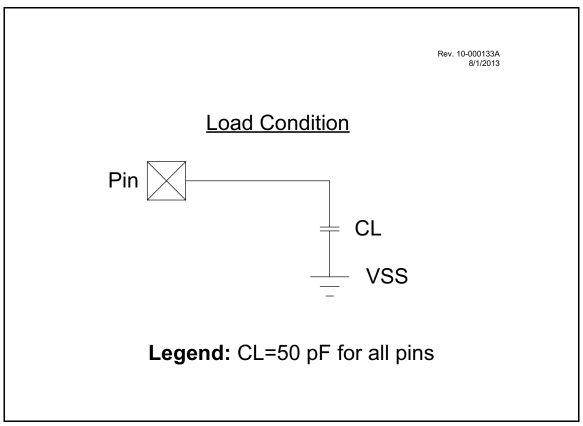


FIGURE 24-5: CLOCK TIMING

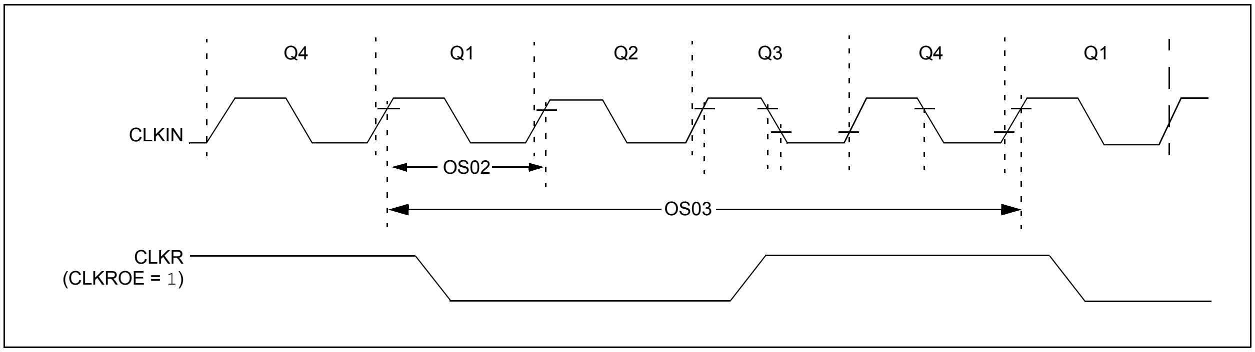


TABLE 24-6: CLOCK OSCILLATOR TIMING REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)

Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
OS01	FOSC	External CLKIN Frequency(1)	DC	—	20	MHz	EC mode
OS02	TOSC	External CLKIN Period(1)	31.25	—	∞	ns	EC Oscillator mode
OS03	TCY	Instruction Cycle Time(1)	200	TCY	DC	ns	TCY = 4/FOSC

* 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 CLKIN pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

TABLE 24-7: OSCILLATOR PARAMETERS

Standard Operating Conditions (unless otherwise stated)

Param No.	Sym.	Characteristic	Freq. Tolerance	Min.	Typ†	Max.	Units	Conditions
OS08	HFOSC	Internal Calibrated HFINTOSC Frequency(1)	±3% -8 to +4%	— —	16.0 16.0	— —	MHz MHz	0°C ≤ TA ≤ +85°C, VDD ≥ 2.3V -40°C ≤ TA ≤ 125°C
OS09	LFOSC	Internal LFINTOSC Frequency	±25%	—	31	—	kHz	
OS10*	TIOSC ST	HFINTOSC Wake-up from Sleep Start-up Time	—	—	5	8	μs	

* 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: To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

FIGURE 24-6: HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE

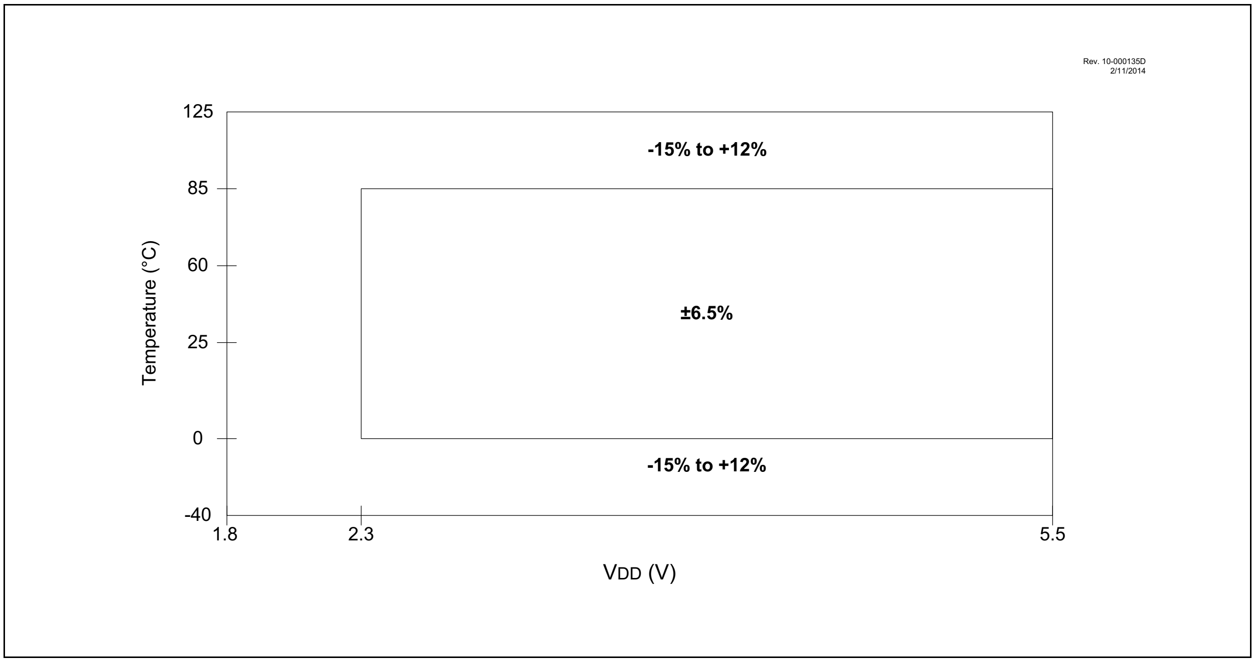


FIGURE 24-7: CLKR AND I/O TIMING

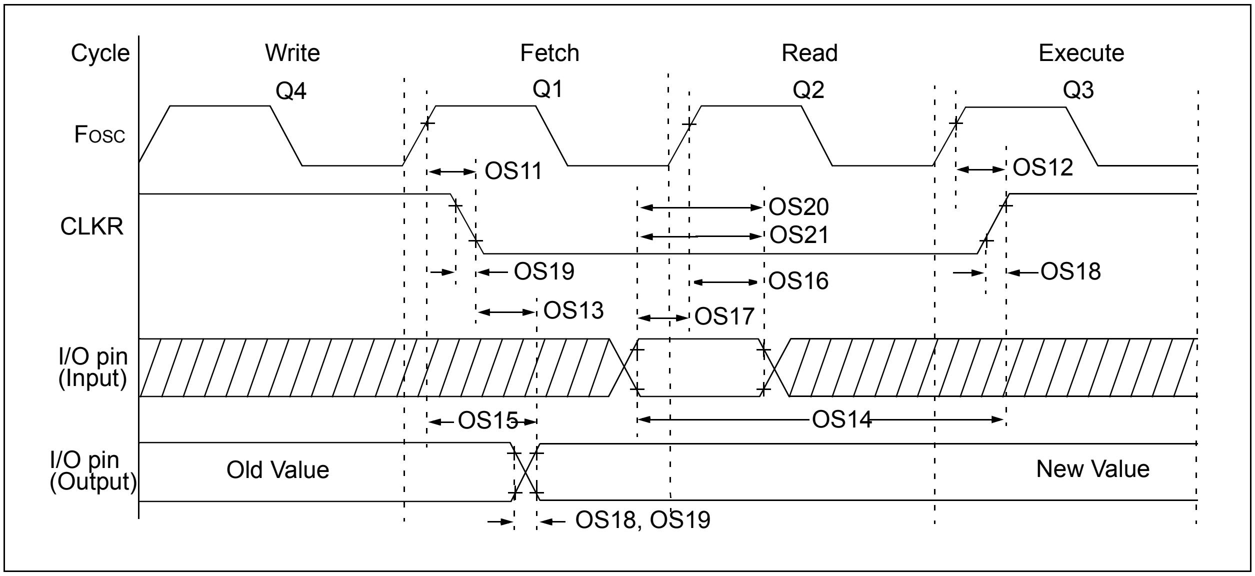
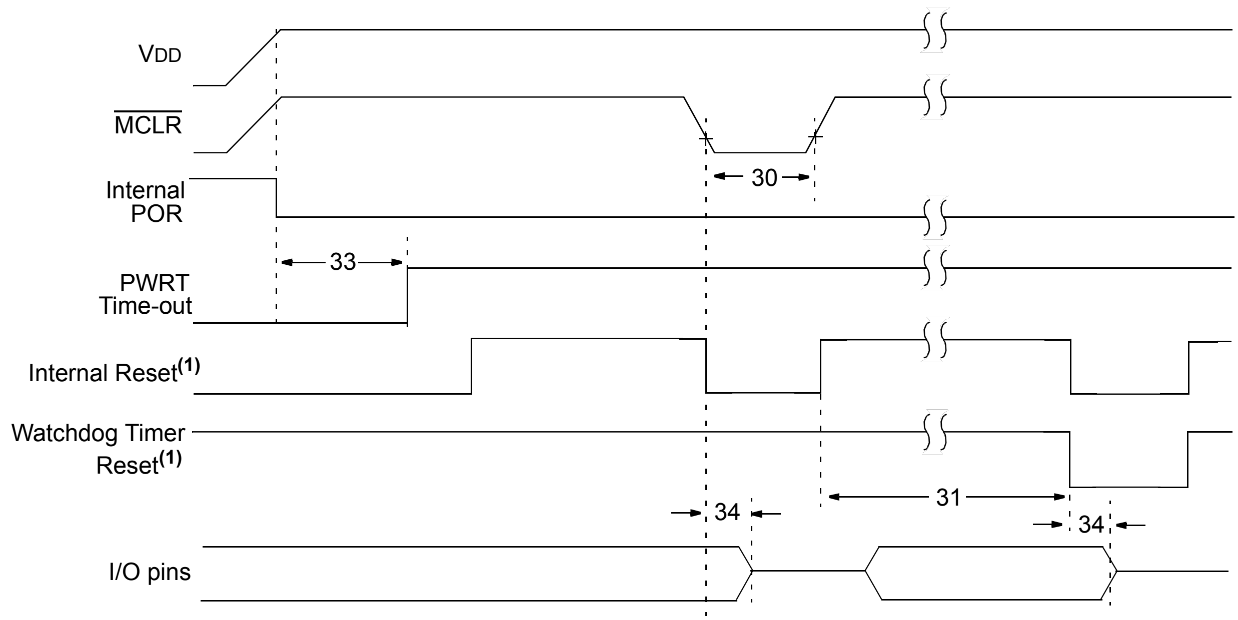


TABLE 24-8: CLKR AND I/O TIMING PARAMETERS

Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
Standard Operating Conditions (unless otherwise stated)							
OS11	TosH2ckL	FOSC↑ to CLKOUT↓(1)	—	—	70	ns	3.3V ≤ VDD ≤ 5.0V
OS12	TosH2ckH	FOSC↑ to CLKOUT↑(1)	—	—	72	ns	3.3V ≤ VDD ≤ 5.0V
OS13	TckL2ioV	CLKOUT↓ to Port out valid(1)	—	—	20	ns	
OS14	TioV2ckH	Port input valid before CLKOUT↑(1)	TOSC + 200 ns	—	—	ns	
OS15	TosH2ioV	Fosc↑ (Q1 cycle) to Port out valid	—	50	70*	ns	3.3V ≤ VDD ≤ 5.0V
OS16	TosH2ioI	Fosc↑ (Q2 cycle) to Port input invalid (I/O in setup time)	50	—	—	ns	3.3V ≤ VDD ≤ 5.0V
OS17	TioV2osH	Port input valid to Fosc↑ (Q2 cycle) (I/O in setup time)	20	—	—	ns	
OS18*	TioR	Port output rise time	— —	40 15	72 32	ns	VDD = 1.8V 3.3V ≤ VDD ≤ 5.0V
OS19*	TioF	Port output fall time	— —	28 15	55 30	ns	VDD = 1.8V 3.3V ≤ VDD ≤ 5.0V
OS20*	Tinp	INT pin input high or low time	25	—	—	ns	
OS21*	Tioc	Interrupt-on-change new input level time	25	—	—	ns	

* These parameters are characterized but not tested.
† Data in “Typ” column is at 3.0V, 25°C unless otherwise stated.
Note 1: Measurements are taken in EXTRC mode where CLKOUT output is 4 x TOSC.

FIGURE 24-8: RESET, WATCHDOG TIMER, AND POWER-UP TIMER TIMING



Note 1: Asserted low.

FIGURE 24-9: BROWN-OUT RESET TIMING AND CHARACTERISTICS

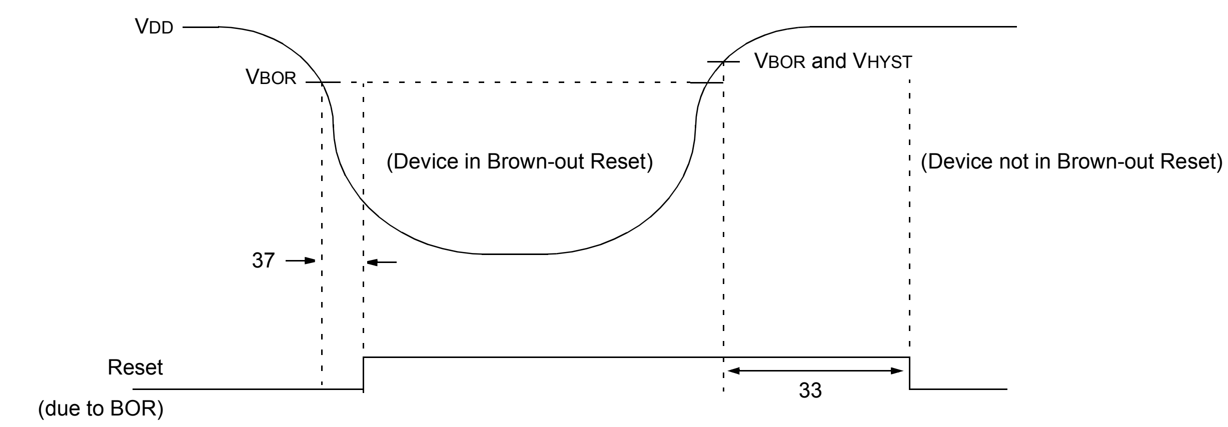


TABLE 24-9: RESET, WATCHDOG TIMER, POWER-UP TIMER AND BROWN-OUT RESET PARAMETERS

Standard Operating Conditions (unless otherwise stated)							
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
30	TMCL	MCLR Pulse Width (low)	2 5	— —	— —	μs μs	VDD = 3.3-5V, -40°C to +85°C VDD = 3.3-5V
31	TWDTLP	Low-Power Watchdog Timer Time-out Period	10	16	27	ms	VDD = 3.3V-5V 1:16 Prescaler used
33*	TPWRT	Power-up Timer Period, PWRTE = 0	40	64	140	ms	
34*	TIOZ	I/O high-impedance from MCLR Low or Watchdog Timer Reset	—	—	2.0	μs	
35	VBOR	Brown-out Reset Voltage[1]	2.55 2.30 1.80	2.70 2.40 1.90	2.85 2.55 2.05	V V V	BORV = 0 BORV = 1 (PIC10F320/322) BORV = 1 (PIC10LF320/322)
36*	VHYST	Brown-out Reset Hysteresis	0	25	50	mV	-40°C to +85°C
37*	TBORDC	Brown-out Reset DC Response Time	1	3	5	μs	VDD ≤ VBOR

* 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: To ensure these voltage tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

FIGURE 24-10: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

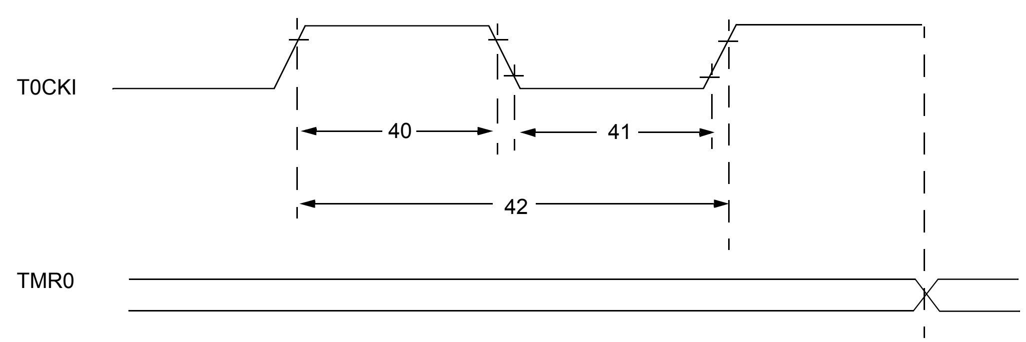


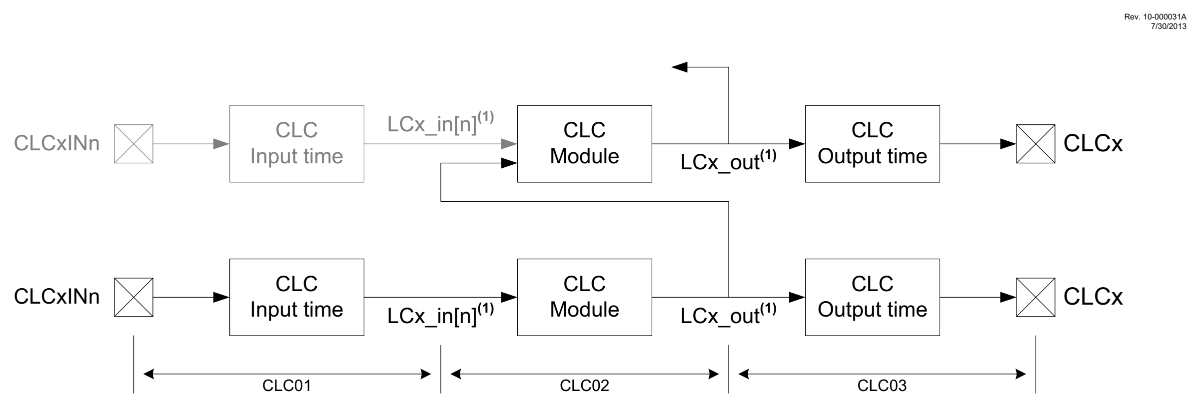
TABLE 24-10: TIMER0 EXTERNAL CLOCK REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)								
Param No.	Sym.	Characteristic		Min.	Typ†	Max.	Units	Conditions
40*	TT0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
41*	TT0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
42*	TT0P	T0CKI Period		Greater of: 20 or $\frac{TCY + 40}{N}$	—	—	ns	N = prescale value (2, 4, ..., 256)

* 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.

FIGURE 24-11: CLC PROPAGATION TIMING



Note 1: See FIGURE 19-1: CLCx Simplified Block Diagram, to identify specific CLC signals.

TABLE 24-11: CONFIGURATION LOGIC CELL (CLC) CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
CLC01*	TCLCIN	CLC input time	—	7	—	ns	
CLC02*	TCLC	CLC module input to output propagation time	— —	24 12	— —	ns ns	VDD = 1.8V VDD > 3.6V
CLC03*	TCLCOUT	CLC output time Rise Time	—	OS18	—	—	**(Note 1)**
		Fall Time	—	OS19	—	—	**(Note 1)**
CLC04*	FCLCMAX	CLC maximum switching frequency	—	45	—	MHz	

* 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: See Table 24-8 for OS18 and OS19 rise and fall times.

TABLE 24-12: A/D CONVERTER (ADC) CHARACTERISTICS:

Standard Operating Conditions (unless otherwise stated)

Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
AD01	NR	Resolution	—	—	8	bit	
AD02	EIL	Integral Error	—	—	±1.7	LSb	VREF = 3.0V
AD03	EDL	Differential Error	—	—	±1	LSb	No missing codes VREF = 3.0V
AD04	EOFF	Offset Error	—	—	±2.5	LSb	VREF = 3.0V
AD05	EGN	Gain Error	—	—	±2.0	LSb	VREF = 3.0V
AD06	VREF	Reference Voltage(3)	1.8	—	VDD	V	VREF = (VREF+ minus VREF-) (**Note 5**)
AD07	VAIN	Full-Scale Range	VSS	—	VREF	V	
AD08	ZAIN	Recommended Impedance of Analog Voltage Source	—	—	10	kΩ	Can go higher if external 0.01μF capacitor is present on input pin.

* 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: Total Absolute Error includes integral, differential, offset and gain errors.

2: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

3: ADC VREF is from external VREF, VDD pin or FVR, whichever is selected as reference input.

4: When ADC is off, it will not consume any current other than leakage current. The power-down current specification includes any such leakage from the ADC module.

5: FVR voltage selected must be 2.048V or 4.096V.

TABLE 24-13: A/D CONVERSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)

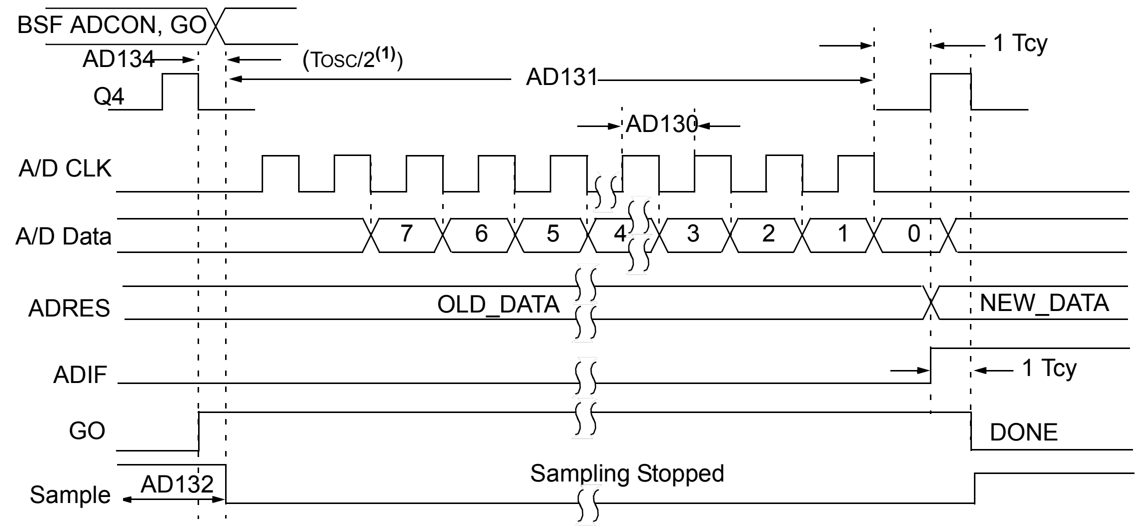
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
AD130*	TAD	A/D Clock Period	1.0	—	6.0	μs	TOSC-based
		A/D Internal FRC Oscillator Period	1.0	1.6	6.0	μs	ADCS<1:0> = 11 (ADRC mode)
AD131	TCNV	Conversion Time (not including Acquisition Time)(1)	—	9.5	—	TAD	Set GO/$\overline{\text{DONE}}$ bit to conversion complete
AD132*	TACQ	Acquisition Time	—	5.0	—	μs	

* 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.

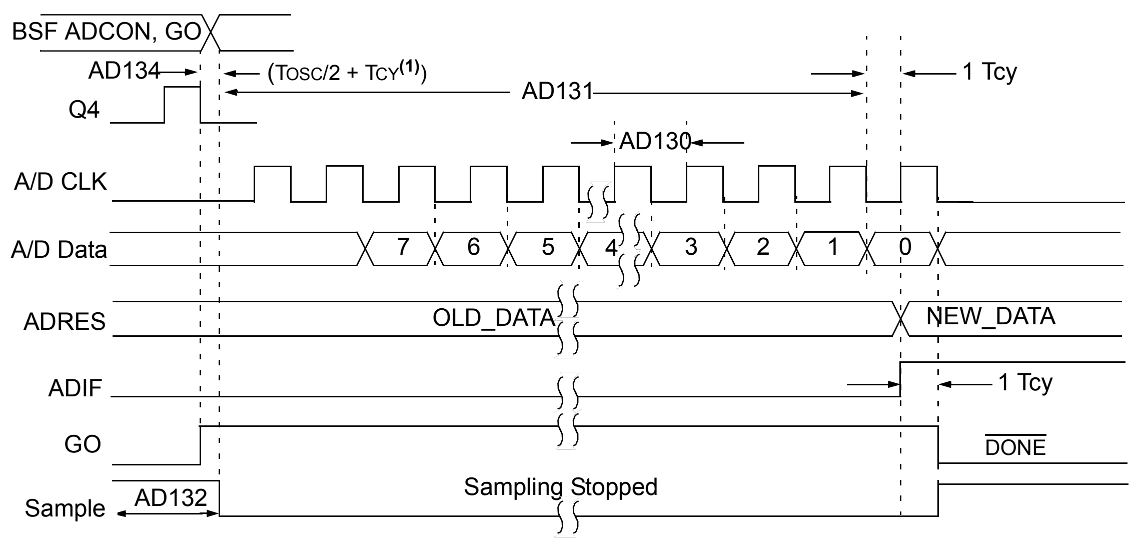
Note 1: The ADRES register may be read on the following TCY cycle.

FIGURE 24-12: A/D CONVERSION TIMING (NORMAL MODE)



Note 1: If the A/D clock source is selected as FRC, a time of TCY is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.

FIGURE 24-13: A/D CONVERSION TIMING (SLEEP MODE)



Note 1: If the A/D clock source is selected as FRC, a time of TCY is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.

25.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

Graphs and charts are not available at this time.

26.0 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

26.1 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

26.2 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

26.3 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

26.4 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

26.5 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

26.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

26.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

26.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

26.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

26.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

26.11 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

26.12 Third-Party Development Tools

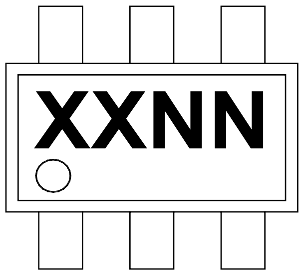
Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

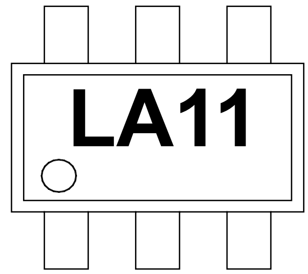
27.0 PACKAGING INFORMATION

27.1 Package Marking Information

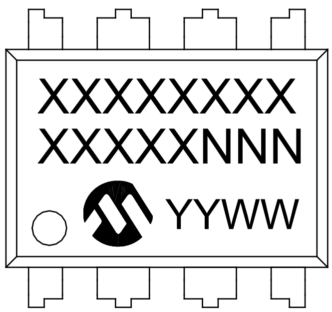
6-Lead SOT-23



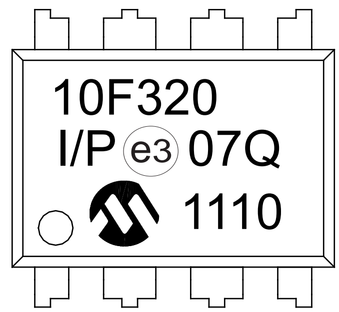
Example



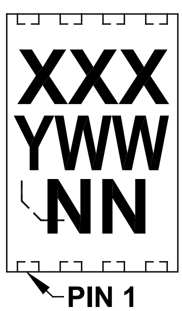
8-Lead PDIP (300 mil)



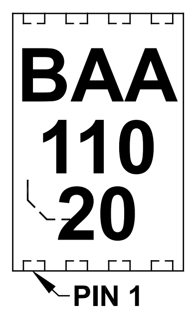
Example



8-Lead DFN (2x3x0.9 mm)



Example



Legend:	XX...X	Product-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	e3	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

TABLE 27-1: 8-LEAD 2x3 DFN (MC) TOP MARKING

Part Number	Marking
PIC10F322(T)-I/MC	BAA
PIC10F322(T)-E/MC	BAB
PIC10F320(T)-I/MC	BAC
PIC10F320(T)-E/MC	BAD
PIC10LF322(T)-I/MC	BAF
PIC10LF322(T)-E/MC	BAG
PIC10LF320(T)-I/MC	BAH
PIC10LF320(T)-E/MC	BAJ

TABLE 27-2: 6-LEAD SOT-23 (OT) PACKAGE TOP MARKING

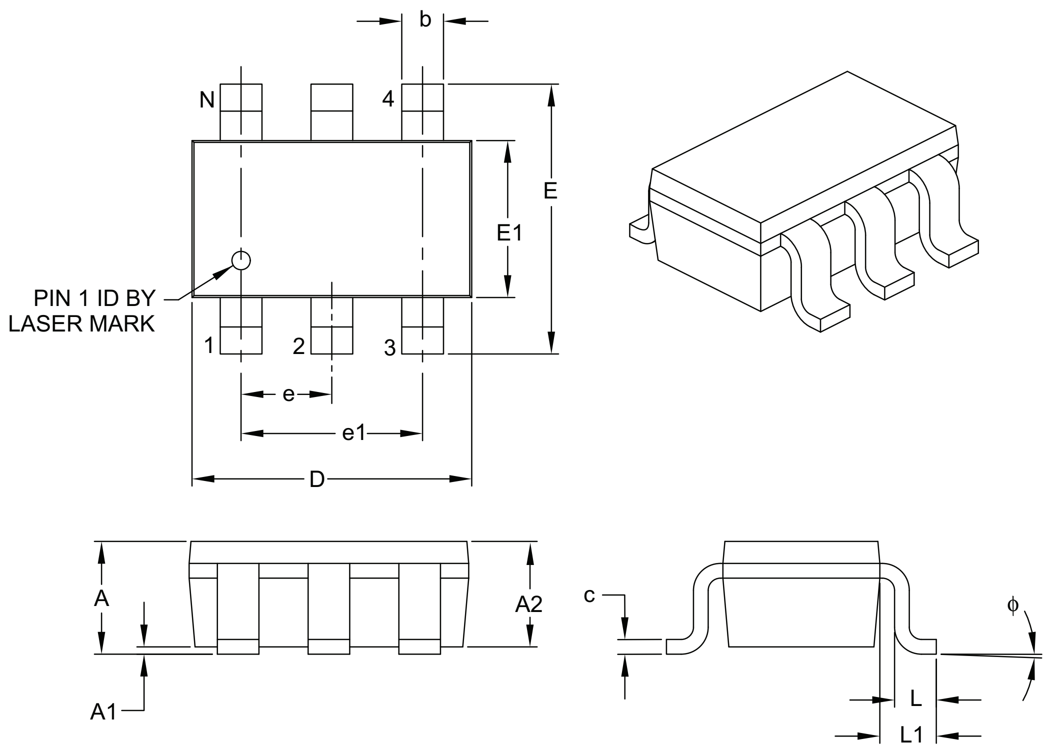
Part Number	Marking
PIC10F322(T)-I/OT	LA/LJ
PIC10F322(T)-E/OT	LB/LK
PIC10F320(T)-I/OT	LC
PIC10F320(T)-E/OT	LD
PIC10LF322(T)-I/OT	LE
PIC10LF322(T)-E/OT	LF
PIC10LF320(T)-I/OT	LG
PIC10LF320(T)-E/OT	LH

27.2 Package Details

The following sections give the technical details of the packages.

6-Lead Plastic Small Outline Transistor (OT) [SOT-23]

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		6	
Pitch	e		0.95 BSC	
Outside Lead Pitch	e1		1.90 BSC	
Overall Height	A	0.90	–	1.45
Molded Package Thickness	A2	0.89	–	1.30
Standoff	A1	0.00	–	0.15
Overall Width	E	2.20	–	3.20
Molded Package Width	E1	1.30	–	1.80
Overall Length	D	2.70	–	3.10
Foot Length	L	0.10	–	0.60
Footprint	L1	0.35	–	0.80
Foot Angle	φ	0°	–	30°
Lead Thickness	c	0.08	–	0.26
Lead Width	b	0.20	–	0.51

Notes:

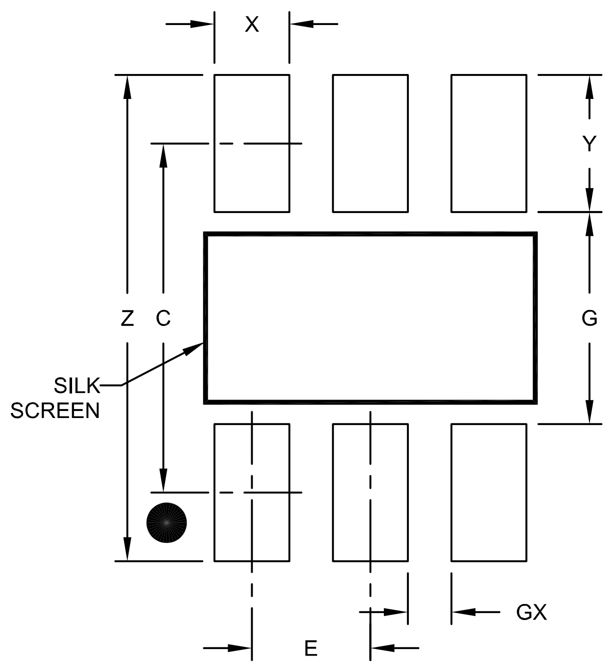
1. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.127 mm per side.
2. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-028B

6-Lead Plastic Small Outline Transistor (OT) [SOT-23]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.95 BSC		
Contact Pad Spacing	C		2.80	
Contact Pad Width (X6)	X			0.60
Contact Pad Length (X6)	Y			1.10
Distance Between Pads	G	1.70		
Distance Between Pads	GX	0.35		
Overall Width	Z			3.90

Notes:

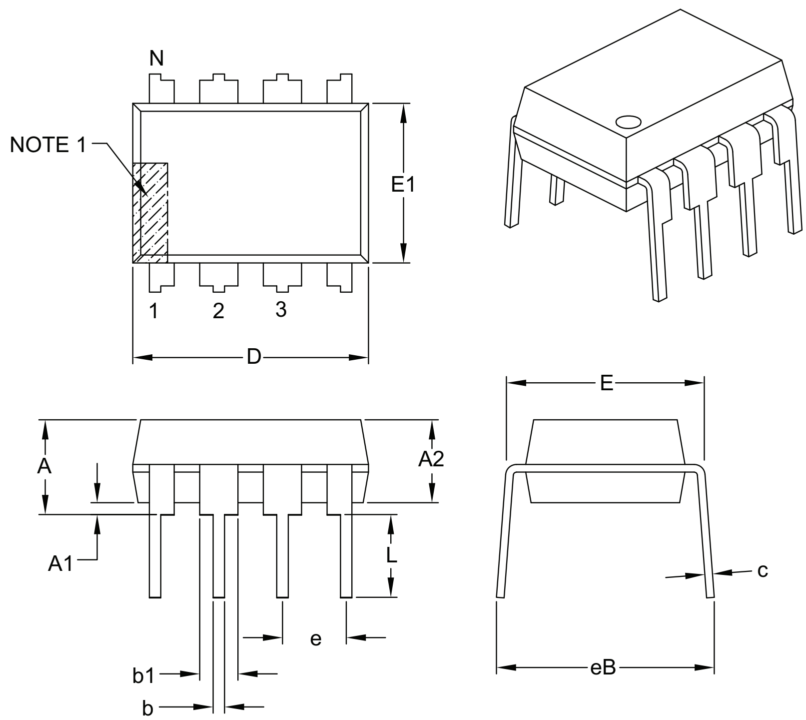
1. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2028A

8-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

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



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	8		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.348	.365	.400
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

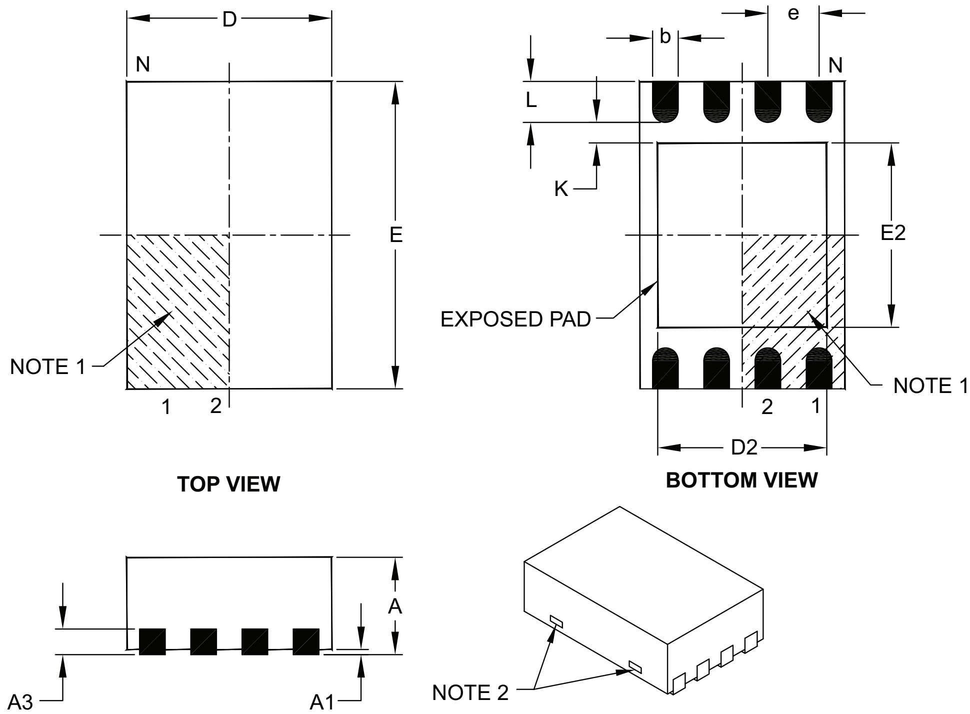
1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-018B

8-Lead Plastic Dual Flat, No Lead Package (MC) – 2x3x0.9 mm Body [DFN]

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		8	
Pitch	e		0.50 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.20 REF	
Overall Length	D		2.00 BSC	
Overall Width	E		3.00 BSC	
Exposed Pad Length	D2	1.30	–	1.55
Exposed Pad Width	E2	1.50	–	1.75
Contact Width	b	0.20	0.25	0.30
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	–	–

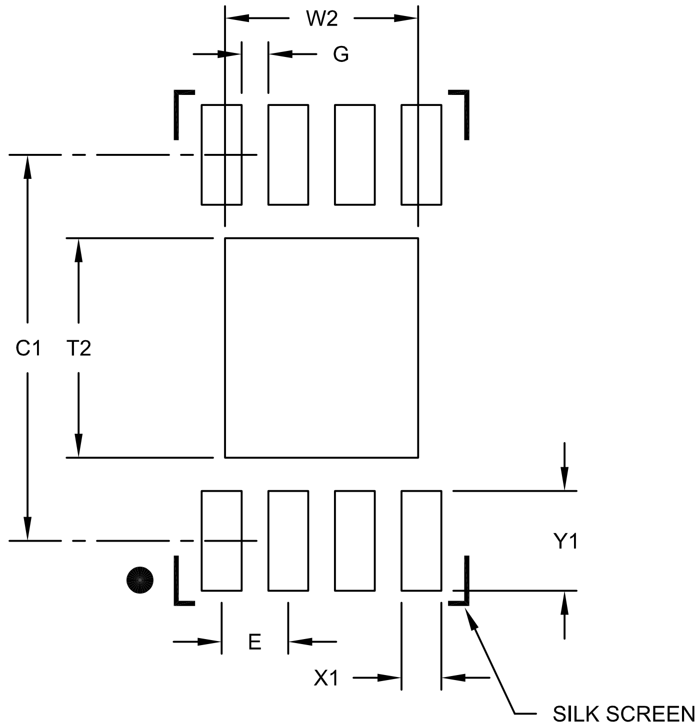
Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package may have one or more exposed tie bars at ends.
3. Package is saw singulated.
4. 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 C04-123C

8-Lead Plastic Dual Flat, No Lead Package (MC) - 2x3x0.9mm Body [DFN]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Optional Center Pad Width	W2			1.45
Optional Center Pad Length	T2			1.75
Contact Pad Spacing	C1		2.90	
Contact Pad Width (X8)	X1			0.30
Contact Pad Length (X8)	Y1			0.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2123B

APPENDIX A: DATA SHEET REVISION HISTORY

Revision A (07/2011)

Original release.

Revision B (02/2014)

Electrical Specifications update and new formats; Minor edits.

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X](1)	-	X	/XX	XXX
Device	Tape and Reel Option		Temperature Range	Package	Pattern

Device:	PIC10F320, PIC10LF320, PIC10F322, PIC10LF322
Tape and Reel Option:	Blank = Standard packaging (tube or tray) T = Tape and Reel(1)
Temperature Range:	I = -40°C to +85°C (Industrial) E = -40°C to +125°C (Extended)
Package:	OT = SOT-23 P = PDIP MC = DFN
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)

Examples:

a) PIC10LF320T - I/OT
Tape and Reel,
Industrial temperature,
SOT-23 package

b) PIC10F322 - I/P
Industrial temperature
PDIP package

c) PIC10F322 - E/MC
Extended temperature,
DFN package

Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

Printed on recycled paper.

ISBN: 9781620779323

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Pforzheim
Tel: 49-7231-424750

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

10/28/13