PIC16(L)F18325/18345

Full-Featured, Low Pin Count Microcontrollers with XLP

Description

PIC16(L)F18325/18345 microcontrollers feature Analog, Core Independent Peripherals and Communication Peripherals, combined with eXtreme Low Power (XLP) for a wide range of general purpose and low-power applications. The Peripheral Pin Select (PPS) functionality enables pin mapping when using the digital peripherals (CLC, CWG, CCP, PWM and communications) to add flexibility to the application design.

Core Features

- C Compiler Optimized RISC Architecture
- Only 49 Instructions
- Operating Speed:
  - DC – 32 MHz clock input
  - 125 ns minimum instruction cycle
- Interrupt Capability
- 16-Level Deep Hardware Stack
- Up to Four 8-bit Timers
- Up to Three 16-bit Timers
- Low-Current Power-on Reset (POR)
- Configurable Power-up Timer (PWRTE)
- Brown-out Reset (BOR) with Fast Recovery
- Low-Power BOR (LPBOR) Option
- Extended Watchdog Timer (WDT) with Dedicated On-Chip Oscillator for Reliable Operation
- Programmable Code Protection

Memory

- 14 Kbytes Program Flash Memory
- 1 KB Data SRAM Memory
- 256B of EEPROM
- Direct, Indirect and Relative Addressing Modes

Operating Characteristics

- Operating Voltage Range:
  - 1.8V to 3.6V (PIC16LF18325/18345)
  - 2.3V to 5.5V (PIC16F18325/18345)
- Temperature Range:
  - Industrial: -40°C to 85°C
  - Extended: -40°C to 125°C

eXtreme Low-Power (XLP) Features

- Sleep mode: 40 nA @ 1.8V, typical
- Watchdog Timer: 250 nA @ 1.8V, typical
- Secondary Oscillator: 300 nA @ 32 kHz
- Operating Current:
  - 8 μA @ 32 kHz, 1.8V, typical
  - 37 μA/MHz @ 1.8V, typical

Power-Saving Functionality

- IDLE mode: ability to put the CPU core to Sleep while internal peripherals continue operating from the system clock
- DOZE mode: ability to run the CPU core slower than the system clock used by the internal peripherals
- SLEEP mode: Lowest Power Consumption
- Peripheral Module Disable (PMD): peripheral power disable hardware module to minimize power consumption of unused peripherals

Digital Peripherals

- Configurable Logic Cell (CLC):
  - Four CLCs
  - Integrated combinational and sequential logic
- Complementary Waveform Generator (CWG):
  - Two CWGs
  - Rising and falling edge dead-band control
  - Full-bridge, half-bridge, 1-channel drive
  - Multiple signal sources
- Capture/Compare/PWM (CCP) modules:
  - Four CCPs
  - 16-bit resolution for Capture/Compare modes
  - 10-bit resolution for PWM mode
- Pulse-Width Modulators (PWM)
  - Two 10-bit PWMs
- Numerically Controlled Oscillator (NCO):
  - Precision linear frequency generator (@50% duty cycle) with 0.0001% step size of source input clock
  - Input Clock: 0 Hz < F_{NCO} < 32 MHz
  - Resolution: $F_{NCO}/2^{20}$
- Serial Communications:
  - EUSART
    - RS-232, RS-485, LIN compatible
    - Auto-Baud Detect, auto-wake-up on start
  - Master Synchronous Serial Port (MSSP)
    - SPI
    - I^2C™, SMBus, PMBus™ compatible
- Data Signal Modulator (DSM):
  - Modulates a carrier signal with digital data to create custom carrier synchronized output waveforms

- Up to 18 I/O Pins:
  - Individually programmable pull-ups
  - Slew rate control
  - Interrupt-on-change with edge-select
  - Input level selection control (ST or TTL)
  - Digital open-drain enable
- Peripheral Pin Select (PPS):
  - I/O pin remapping of digital peripherals
- Timer modules:
  - Timer0:
    - 8/16-bit timer/counter
    - Synchronous or asynchronous operation
    - Programmable prescaler/postscaler
    - Time base for capture/compare function
  - Timer1/3/5 with gate control:
    - 16-bit timer/counter
    - Programmable internal or external clock sources
    - Multiple gate sources
    - Multiple gate modes
    - Time base for capture/compare function
  - Timer2/4/6:
    - 8-bit timers
    - Programmable prescaler/postscaler
    - Time base for PWM function

Analog Peripherals

- 10-bit Analog-to-Digital Converter (ADC):
  - 17 external channels
  - Conversion available during Sleep
- Comparator:
  - Two comparators
  - Fixed Voltage Reference at non-inverting input(s)
  - Comparator outputs externally accessible
- 5-bit Digital-to-Analog Converter (DAC):
  - 5-bit resolution, rail-to-rail
  - Positive Reference Selection
  - Unbuffered I/O pin output
  - Internal connections to ADCs and comparators
- Voltage Reference:
  - Fixed Voltage Reference with 1.024V, 2.048V and 4.096V output levels

Flexible Oscillator Structure

- High-Precision Internal Oscillator:
  - Software-selectable frequency range up to 32 MHz
  - ±2% at nominal 4 MHz calibration point
- 4x PLL with External Sources
- Low-Power Internal 31 kHz Oscillator (LFINTOSC)
- External Low-Power 32 kHz Crystal Oscillator (SOSC)
- External Oscillator Block with:
  - Three Crystal/Resonator modes up to 20 MHz
  - Three External Clock modes up to 20 MHz
  - Fail-Safe Clock Monitor
    - Allows for safe shutdown if peripheral clock stops
  - Oscillator Start-up Timer (OST)
    - Ensures stability of crystal oscillator sources

PIC16(L)F183XX Family Types

Device	Data Sheet Index	Program Flash Memory (Words)	Program Flash Memory (Kbytes)	Data Memory (bytes)	Data SRAM (bytes)	I/Os(2)	10-bit ADC (ch)	5-bit DAC	High-Speed/ Comparators	CWG	Clock Ref	Timers (8/16-bit)	CCP	10-bit PWM	NCO	EUSART	MSSP (I^2C™ /SPI)	CLC	DSM	PPS	XLP	PMD	Idle and Doze	Debug(1)
PIC16(L)F18313	(1)	2048	3.5	256	256	6	5	1	1	1	1	2/1	2	2	1	1	1	2	1	Y	Y	Y	Y	I
PIC16(L)F18323	(1)	2048	3.5	256	256	12	11	1	2	1	1	2/1	2	2	1	1	1	2	1	Y	Y	Y	Y	I
PIC16(L)F18324	(2)	4096	7	256	512	12	11	1	2	2	1	4/3	4	2	1	1	1	4	1	Y	Y	Y	Y	I
PIC16(L)F18325	(3)	8192	14	256	1024	12	11	1	2	2	1	4/3	4	2	1	1	2	4	1	Y	Y	Y	Y	I
PIC16(L)F18344	(2)	4096	7	256	512	18	17	1	2	2	1	4/3	4	2	1	1	1	4	1	Y	Y	Y	Y	I
PIC16(L)F18345	(3)	8192	14	256	1024	18	17	1	2	2	1	4/3	4	2	1	1	2	4	1	Y	Y	Y	Y	I

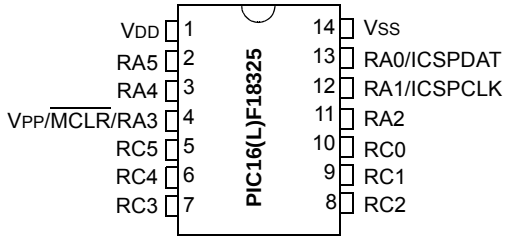
Note 1: Debugging Methods: (I) – Integrated on Chip;
2: One pin is input-only.

Data Sheet Index: (Unshaded devices are described in this document.)
1: DS40001799 PIC16(L)F18313/18323 Data Sheet, Full-Featured, Low Pin Count Microcontrollers with XLP
2: DS40001800 PIC16(L)F18324/18344 Data Sheet, Full Featured, Low Pin Count Microcontrollers with XLP
3: DS40001795 PIC16(L)F18325/18345 Data Sheet, Full Featured, Low Pin Count Microcontrollers with XLP

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

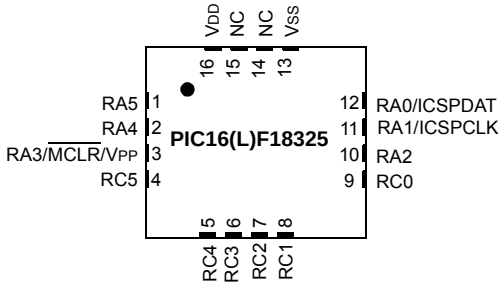
Pin Diagrams

FIGURE 1: 14-PIN PDIP, SOIC, TSSOP



Note: See Table 1 for location of all peripheral functions.

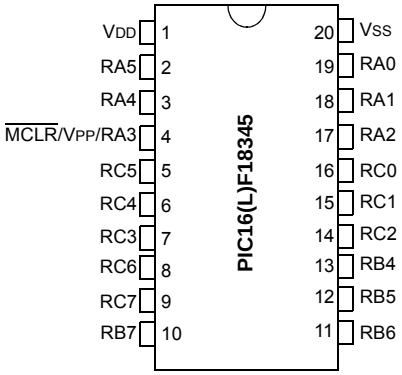
FIGURE 2: 16-PIN UQFN (4x4)



Note 1: See Table 1 for location of all peripheral functions.

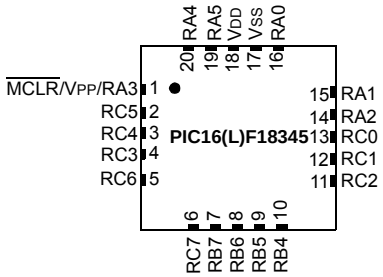
2: It is recommended that the exposed bottom pad be connected to VSS, but must not be the main VSS connection to the device.

FIGURE 3: 20-PIN PDIP, SOIC, SSOP



Note: See Table 2 for location of all peripheral functions.

FIGURE 4: 20-PIN UQFN (4x4)



Note 1: See Table 2 for location of all peripheral functions.

2: It is recommended that the exposed bottom pad be connected to VSS, but must not be the main VSS connection to the device.

Pin Allocation Tables

TABLE 1: 14/16-PIN ALLOCATION TABLE (PIC16(L)F18325)

I/O(2)	14-Pin PDIP/SOIC/TSSOP	16-Pin UQFN	ADC	Reference	Comparator	NCO	DAC	DSM	Timers	CCP	PWM	CWG	MSSP	EUSART	CLC
RA0	13	12	ANA0	—	C1IN0+	—	DAC1OUT	—	—	—	—	—	$\overline{SS2}$(1)	—	—
RA1	12	11	ANA1	VREF+	C1IN0- C2IN0-	—	DAC1REF+	—	—	—	—	—	—	—	—
RA2	11	10	ANA2	VREF-	—	—	DAC1REF-	—	T0CKI(1)	CCP3(1)	—	CWG1IN(1) CWG2IN(1)	—	—	—
RA3	4	3	—	—	—	—	—	—	—	—	—	—	—	—	—
RA4	3	2	ANA4	—	—	—	—	—	T1G(1) SOSCO	—	—	—	—	—	—
RA5	2	1	ANA5	—	—	—	—	—	T1CKI(1) SOSCIN SOSCI	—	—	—	—	—	CLCIN3(1)
RC0	10	9	ANC0	—	C2IN0+	—	—	—	T5CKI(1)	—	—	—	SCK1(1) SCL1(1,3,4)	—	—
RC1	9	8	ANC1	—	C1IN1- C2IN1-	—	—	—	—	CCP4(1)	—	—	SDI1(1) SDA1(1,3,4)	—	CLCIN2(1)
RC2	8	7	ANC2	—	C1IN2- C2IN2-	—	—	MDCIN1(1)	—	—	—	—	—	—	—
RC3	7	6	ANC3	—	C1IN3- C2IN3-	—	—	MDMIN(1)	T5G(1)	CCP2(1)	—	—	$\overline{SS1}$(1)	—	CLCIN0(1)
RC4	6	5	ANC4	—	—	—	—	—	T3G(1)	—	—	—	SCK2(1) SCL2(1,3,4)	—	CLCIN1(1)
RC5	5	4	ANC5	—	—	—	—	MDCIN2(1)	T3CKI(1)	CCP1(1)	—	—	SDI2(1) SDA2(1,3,4)	RX(1) DT(1,3)	—
VDD	1	16	—	—	—	—	—	—	—	—	—	—	—	—	—
VSS	14	13	—	—	—	—	—	—	—	—	—	—	—	—	—

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.
2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers.
3: These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.
4: These pins are configured for I^2C™ logic levels; clock and data signals may be assigned to any of these pins. Assignments to the other pins (e.g., RA5) will operate
TTL/ST as selected by the INLVL register.

TABLE 1: 14/16-PIN ALLOCATION TABLE (PIC16(L)F18325) (CONTINUED)

I/O(2)	14-Pin PDIP/SOIC/TSSOP	16-Pin UQFN	ADC	Reference	Comparator	NCO	DAC	DSM	Timers	CCP	PWM	CWG	MSSP	EUSART	CLC
OUT(2)	—	—	—	—	C1OUT	NCO1	—	DSM	TMR0	CCP1	PWM5	CWG1A CWG2A	SDA1(3) SDA2(3)	CK	CLC1OUT
	—	—	—	—	C2OUT	—	—	—	—	CCP2	PWM6	CWG1B CWG2B	SCL1(3) SCL2(3)	DT(3)	CLC2OUT
	—	—	—	—	—	—	—	—	—	CCP3	—	CWG1C CWG2C	SDO1 SDO2	TX	CLC3OUT
	—	—	—	—	—	—	—	—	—	CCP4	—	CWG1D CWG2D	SCK1 SCK2	—	CLC4OUT

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.
2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers.
3: These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.
4: These pins are configured for I^2C™ logic levels; clock and data signals may be assigned to any of these pins. Assignments to the other pins (e.g., RA5) will operate
TTL/ST as selected by the INLVL register.

TABLE 2: 20-PIN ALLOCATION TABLE (PIC16(L)F18345)

I/O(2)	20-Pin PDIP/SOIC/SSOP	20-Pin UQFN	ADC	Reference	Comparator	NCO	DAC	DSM	Timers	CCP	PWM	CWG	MSSP	EUSART	CLC
RA0	19	16	ANA0	—	C1IN0+	—	DAC1OUT	—	—	—	—	—	—	—	—
RA1	18	15	ANA1	VREF+	C1IN0- C2IN0-	—	DAC1REF+	—	—	—	—	—	$\overline{SS2}$	—	—
RA2	17	14	ANA2	VREF-	—	—	DAC1REF-	—	T0CKI(1)	CCP3(1)	—	CWG1IN(1) CWG2IN(1)	—	—	CLCIN0(1)
RA3	4	1	—	—	—	—	—	—	—	—	—	—	—	—	—
RA4	3	20	ANA4	—	—	—	—	—	T1G(1) T3G(1) T5G(1) SOSCO	CCP4(1)	—	—	—	—	—
RA5	2	19	ANA5	—	—	—	—	—	T1CKI(1) T3CKI(1) T5CKI(1) SOSCIN SOSCI	—	—	—	—	—	—
RB4	13	10	ANB4	—	—	—	—	—	—	—	—	—	SDI1(1) SDA1(1,3,4)	—	CLCIN2(1)
RB5	12	9	ANB5	—	—	—	—	—	—	—	—	—	SDI2(1) SDA2(1,3,4)	RX(1) DT(1)	CLCIN3(1)
RB6	11	8	ANB6	—	—	—	—	—	—	—	—	—	SCK1(1) SCL1(1,3,4)	—	—
RB7	10	7	ANB7	—	—	—	—	—	—	—	—	—	SCK2(1) SCL2(1,3,4)	—	—
RC0	16	13	ANC0	—	C2IN0+	—	—	—	—	—	—	—	—	—	—
RC1	15	12	ANC1	—	C1IN1- C2IN1-	—	—	—	—	—	—	—	—	—	—
RC2	14	11	ANC2	—	C1IN2- C2IN2-	—	—	MDCIN1(1)	—	—	—	—	—	—	—

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers.

3: These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.

4: These pins are configured for I^2C™ logic levels; clock and data signals may be assigned to any of these pins. Assignments to other pins (e.g. RA5) will operate, bu TTL/ST as selected by the INLVL register.

TABLE 2: 20-PIN ALLOCATION TABLE (PIC16(L)F18345) (CONTINUED)

I/O(2)	20-Pin PDIP/SOIC/SSOP	20-Pin UQFN	ADC	Reference	Comparator	NCO	DAC	DSM	Timers	CCP	PWM	CWG	MSSP	EUSART	CLC	
RC3	7	4	ANC3	—	C1IN3- C2IN3-	—	—	MDMIN(1)	—	CCP2(1)	—	—	—	—	CLCIN1(1)	
RC4	6	3	ANC4	—	—	—	—	—	—	—	—	—	—	—	—	
RC5	5	2	ANC5	—	—	—	—	MDCIN2(1)	—	CCP1(1)	—	—	—	—	—	
RC6	8	5	ANC6	—	—	—	—	—	—	—	—	—	$\overline{SS1}$(1)	—	—	
RC7	9	6	ANC7	—	—	—	—	—	—	—	—	—	—	—	—	
VDD	1	18	—	—	—	—	—	—	—	—	—	—	—	—	—	
VSS	20	17	—	—	—	—	—	—	—	—	—	—	—	—	—	
OUT(2)	—	—	—	—	C1OUT	NCO1	—	DSM	TMR0	CCP1	PWM5	CWG1A CWG2A	SDO1 SDO2	DT(3)	CLC1OUT	
	—	—	—	—	C2OUT	—	—	—	—	CCP2	PWM6	CWG1B CWG2B	SCK1 SCK2	CK	CLC2OUT	
	—	—	—	—	—	—	—	—	—	CCP3	—	CWG1C CWG2C	SCL1(3) SCL2(3)	TX	CLC3OUT	
	—	—	—	—	—	—	—	—	—	CCP4	—	CWG1D CWG2D	SDA1(3) SDA2(3)	—	CLC4OUT	

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers.

3: These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.

4: These pins are configured for I^2C™ logic levels; clock and data signals may be assigned to any of these pins. Assignments to other pins (e.g. RA5) will operate, bu
TTL/ST as selected by the INLVL register.

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**. 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., DS30000000A is version A of document DS30000000).

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 PIC16(L)F18325/18345 devices are described within this data sheet. PIC16(L)F18325 is available in 14-pin PDIP, SOIC, TSSOP and 16-pin UQFN packages. PIC16(L)F18345 is available in 20-pin PDIP, SOIC, SSOP and UQFN packages. See **Section 37.0 "Packaging Information"** for further packaging information. Figure 1-1 shows a block diagram of the PIC16(L)F18325/18345 devices. Table 1-2 shows the pinout descriptions.

Reference Table 1-1 for peripherals available per device.

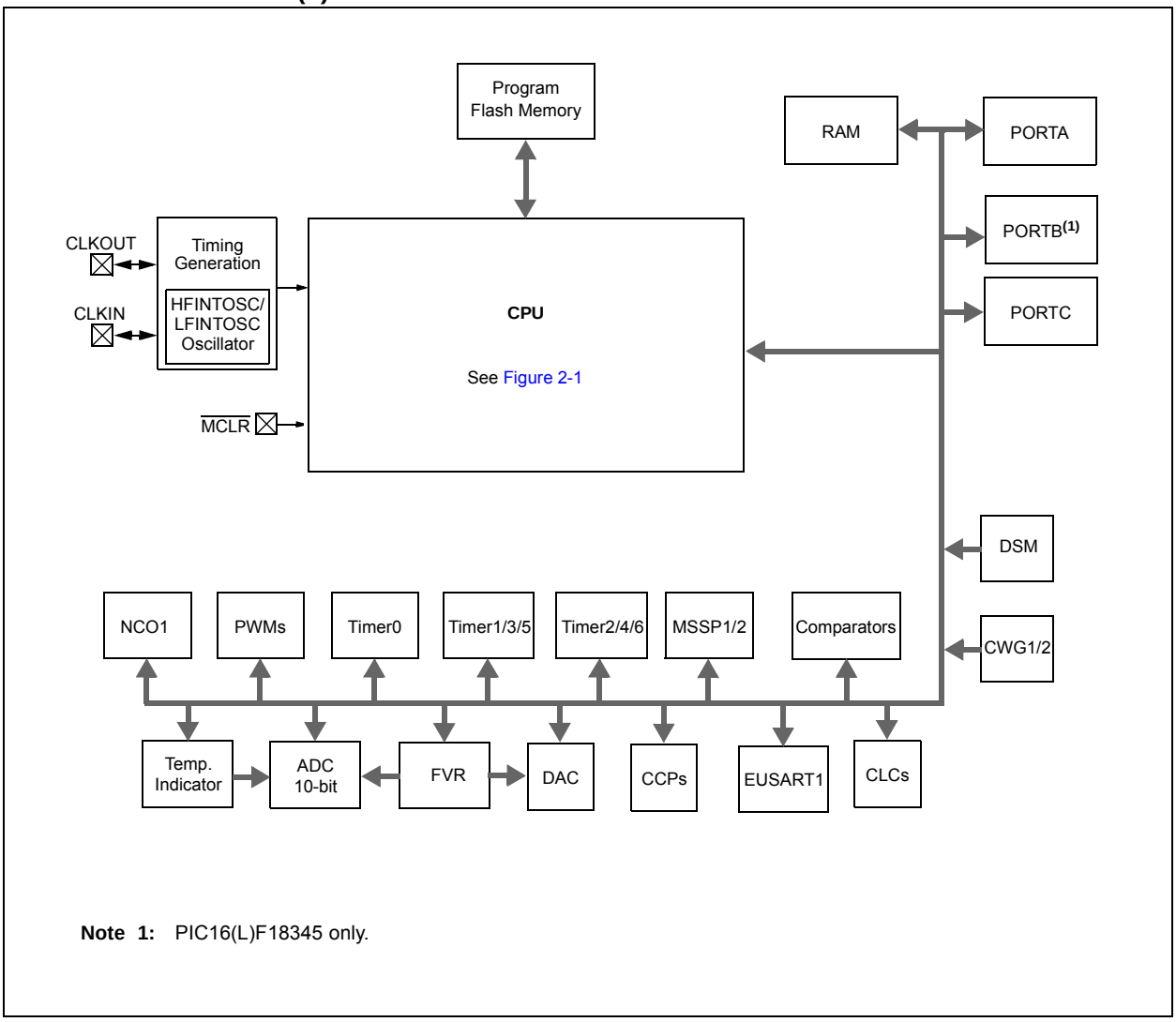
TABLE 1-1: DEVICE PERIPHERAL SUMMARY

Peripheral		PIC16(L)F18325	PIC16(L)F18345
Analog-to-Digital Converter (ADC)		•	•
Temperature Indicator		•	•
Digital-to-Analog Converter (DAC)			
	DAC1	•	•
Fixed Voltage Reference (FVR)			
	ADCFVR	•	•
	CDAFVR	•	•
Digital Signal Modulator (DSM)			
	DSM1	•	•
Numerically Controlled Oscillator (NCO)			
	NCO1	•	•
Capture/Compare/PWM (CCP/ECCP) Modules			
	CCP1	•	•
	CCP2	•	•
	CCP3	•	•
	CCP4	•	•
Comparators			
	C1	•	•
	C2	•	•
Complementary Waveform Generator (CWG)			
	CWG1	•	•
	CWG2	•	•
Configurable Logic Cell (CLC)			
	CLC1	•	•
	CLC2	•	•
	CLC3	•	•
	CLC4	•	•
Enhanced Universal Synchronous/Asynchronous Receiver/ Transmitter (EUSART)			
	EUSART1	•	•
Master Synchronous Serial Port (MSSP)			
	MSSP1	•	•
	MSSP2	•	•
Pulse-Width Modulator (PWM)			
	PWM5	•	•
	PWM6	•	•

TABLE 1-1: DEVICE PERIPHERAL SUMMARY (CONTINUED)

Peripheral		PIC16(L)F18325	PIC16(L)F18345
Timers			
	Timer0	•	•
	Timer1	•	•
	Timer2	•	•
	Timer3	•	•
	Timer4	•	•
	Timer5	•	•
	Timer6	•	•

FIGURE 1-1: PIC16(L)F18325/18345 BLOCK DIAGRAM



Note 1: PIC16(L)F18345 only.

TABLE 1-2: PIC16(L)F18325 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/ANA0/C1IN0+/DAC1OUT/ $\overline{SS2}$(1)/ ICDDAT/ICSPDAT	RA0	TTL/ST	CMOS	General purpose I/O.
	ANA0	AN	—	ADC Channel A0 input.
	C1IN0+	AN	—	Comparator C1 positive input.
	DAC1OUT	—	AN	Digital-to-Analog Converter output.
	$\overline{SS2}$	TTL/ST	—	Slave Select 2 input.
	ICDDAT	TTL/ST	CMOS	In-Circuit Debug Data I/O.
	ICSPDAT	TTL/ST	CMOS	ICSP™ Data I/O.
RA1/ANA1/VREF+/C1IN0-/ C2IN0-/DAC1REF+/ ICDCLK/ ICSPCLK	RA1	TTL/ST	CMOS	General purpose I/O.
	ANA1	AN	—	ADC Channel A1 input.
	VREF+	AN	—	ADC positive voltage reference input.
	C1IN0-	AN	—	Comparator C1 negative input.
	C2IN0-	AN	—	Comparator C2 negative input.
	DAC1REF+	—	AN	Digital-to-Analog Converter positive reference input.
	ICDCLK	TTL/ST	CMOS	In-Circuit Debug Clock I/O.
	ICSPCLK	TTL/ST	CMOS	ICSP™ Clock I/O.
RA2/ANA2/VREF-/ DAC1REF-/ T0CKI(1)/ CCP3(1)/CWG1IN(1)/ CWG2IN(1)/INT(1)	RA2	TTL/ST	CMOS	General purpose I/O.
	ANA2	AN	—	ADC Channel A2 input.
	VREF-	AN	—	ADC negative voltage reference input.
	DAC1REF-	—	AN	Digital-to-Analog Converter negative reference input.
	T0CKI	TTL/ST	—	Timer0 Clock input.
	CCP3	TTL/ST	CMOS	Capture/Compare/PWM 3 input.
	CWG1IN	TTL/ST	—	Complementary Waveform Generator 1 input.
	CWG2IN	TTL/ST	—	Complementary Waveform Generator 2 input.
	INT	TTL/ST	—	External interrupt input.
RA3/$\overline{MCLR}$/VPP	RA3	TTL/ST	CMOS	General purpose I/O.
	$\overline{MCLR}$	TTL/ST	—	Master Clear with internal pull-up.
	VPP	HV	—	Programming voltage.
RA4/ANA4/T1G(1)/ SOSCO/ CLKOUT/OSC2	RA4	TTL/ST	CMOS	General purpose I/O.
	ANA4	AN	—	ADC Channel A4 input.
	T1G	ST	—	Timer1 gate input.
	SOSCO	—	XTAL	Secondary Oscillator connection.
	CLKOUT	—	CMOS	FOSC/4 output.
	OSC2	—	XTAL	Crystal/Resonator (LP, XT, HS modes).

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C = Schmitt Trigger input with I^2C
HV = High Voltage XTAL = Crystal levels

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. See Register 12-1.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 12-2.

3: These I^2C functions are bidirectional. The output pin selections must be the same as the input pin selections.

TABLE 1-2: PIC16(L)F18325 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RA5/ANA5/T1CKI(1)/ SOSCIN/ SOSCI/ CLCIN3(1)/CLKIN/ OSC1	RA5	TTL/ST	CMOS	General purpose I/O.
	ANA5	AN	—	ADC Channel A5 input.
	T1CKI	TTL/ST	—	Timer1 Clock input.
	SOSCIN	TTL/ST	—	Secondary Oscillator input connection.
	SOSCI	XTAL	—	Secondary Oscillator connection.
	CLCIN3	TTL/ST	—	Configurable Logic Cell 3 input.
	CLKIN	TTL/ST	—	External clock input.
	OSC1	XTAL	—	Crystal/Resonator (LP, XT, HS modes).
RC0/ANC0/C2IN0+/ T5CKI(1)/ SCK1(1)/ SCL1(1,3)	RC0	TTL/ST	CMOS	General purpose I/O.
	ANC0	AN	—	ADC Channel C0 input.
	C2IN0+	AN	—	Comparator C2 positive input.
	T5CKI	TTL/ST	—	Timer5 Clock input.
	SCK1	TTL/ST	CMOS	SPI Clock 1.
	SCL1	I^2C™	OD	I^2C™ Clock 1.
RC1/ANC1/C1IN1-/C2IN1-/ CCP4(1)/SDI1(1)/ SDA1(1,3)/ CLCIN2(1)	RC1	TTL/ST	CMOS	General purpose I/O.
	ANC1	AN	—	ADC Channel C1 input.
	C1IN1-	AN	—	Comparator C1 negative input.
	C2IN1-	AN	—	Comparator C2 negative input.
	CCP4	TTL/ST	CMOS	Capture/Compare/PWM 4 input.
	SDI1	TTL/ST	CMOS	SPI Data input 1.
	SDA1	I^2C	OD	I^2C Data 1.
	CLCIN2	TTL/ST	—	Configurable Logic Cell 2 input.
RC2/ANC2/C1IN2-/C2IN2-/ MDCIN1(1)	RC2	TTL/ST	CMOS	General purpose I/O.
	ANC2	AN	—	ADC Channel C2 input.
	C1IN2-	AN	—	Comparator C1 negative input.
	C2IN2-	AN	—	Comparator C2 negative input.
	MDCIN1	TTL/ST	—	Modular Carrier input 1.
RC3/ANC3/C1IN3-/C2IN3-/ MDMIN(1)/T5G(1)/ CCP2(1)/ $\overline{SS1}$(1)/CLCIN0(1)	RC3	TTL/ST	CMOS	General purpose I/O.
	ANC3	AN	—	ADC Channel C3 input.
	C1IN3-	AN	—	Comparator C1 negative input.
	C2IN3-	AN	—	Comparator C2 negative input.
	MDMIN	TTL/ST	—	Modular Source input.
	T5G	TTL/ST	—	Timer5 gate input.
	CCP2	TTL/ST	CMOS	Capture/Compare/PWM 2 input.
	$\overline{SS1}$	TTL/ST	—	Slave Select 1 input.
	CLCIN0	TTL/ST	—	Configurable Logic Cell 0 input.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C = Schmitt Trigger input with I^2C
HV = High Voltage XTAL = Crystal levels

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. See Register 12-1.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 12-2.

3: These I^2C functions are bidirectional. The output pin selections must be the same as the input pin selections.

TABLE 1-2: PIC16(L)F18325 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RC4/ANC4/T3G(1)/ SCK2(1)/ SCL2(1,3)/ CLCIN1(1)	RC4	TTL/ST	CMOS	General purpose I/O.
	ANC4	AN	—	ADC Channel C4 input.
	T3G	TTL/ST	—	Timer3 gate input.
	SCK2	TTL/ST	CMOS	SPI Clock 2.
	SCL2	I^2C	OD	I^2C Clock 2.
	CLCIN1	TTL/ST	—	Configurable Logic Cell 1 input.
RC5/ANC5/MDCIN2(1)/ T3CKI(1)/CCP1(1)/SDI2(1)/ SDA2(1,3)/RX(1)/DT	RC5	TTL/ST	CMOS	General purpose I/O.
	ANC5	AN	—	ADC Channel C5 input.
	MDCIN2	TTL/ST	—	Modular Carrier input 2.
	T3CKI	TTL/ST	—	Timer3 Clock input.
	CCP1	TTL/ST	CMOS	Capture/Compare/PWM 1 input.
	SDI2	TTL/ST	CMOS	SPI Data 2.
	SDA2	I^2C	OD	I^2C Data 2.
	RX	TTL/ST	CMOS	EUSART asynchronous input.
	DT	TTL/ST	CMOS	EUSART synchronous data output.
VDD	VDD	Power	—	Positive supply.
VSS	VSS	Power	—	Ground reference.

Legend: AN = Analog input or output; CMOS = CMOS compatible input or output; OD = Open-Drain
TTL = TTL compatible input; ST = Schmitt Trigger input with CMOS levels; I^2C = Schmitt Trigger input with I^2C
HV = High Voltage; XTAL = Crystal levels

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. See Register 12-1.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 12-2.

3: These I^2C functions are bidirectional. The output pin selections must be the same as the input pin selections.

TABLE 1-2: PIC16(L)F18325 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
OUT(2)	C1	—	CMOS	Comparator C1 output.
	C2	—	CMOS	Comparator C2 output.
	NCO1	—	CMOS	Numerically Controlled Oscillator output.
	DSM	—	CMOS	Digital Signal Modulator output.
	TMR0	—	CMOS	Timer0 clock output.
	CCP1	—	CMOS	Capture/Compare/PWM 1 output.
	CCP2	—	CMOS	Capture/Compare/PWM 2 output.
	CCP3	—	CMOS	Capture/Compare/PWM 3 output.
	CCP4	—	CMOS	Capture/Compare/PWM 4 output.
	PWM5	—	CMOS	Pulse-Width Modulator 5 output.
	PWM6	—	CMOS	Pulse-Width Modulator 6 output.
	CWG1A	—	CMOS	Complementary Waveform Generator 1 output A.
	CWG2A	—	CMOS	Complementary Waveform Generator 2 output A.
	CWG1B	—	CMOS	Complementary Waveform Generator 1 output B.
	CWG2B	—	CMOS	Complementary Waveform Generator 2 output B.
	CWG1C	—	CMOS	Complementary Waveform Generator 1 output C.
	CWG2C	—	CMOS	Complementary Waveform Generator 2 output C.
	CWG1D	—	CMOS	Complementary Waveform Generator 1 output D.
	CWG2D	—	CMOS	Complementary Waveform Generator 2 output D.
	SDA1(3)	I^2C	OD	I^2C data output.
	SDA2(3)	I^2C	OD	I^2C data output.
	SCL1(3)	I^2C	OD	I^2C clock output.
	SCL2(3)	I^2C	OD	I^2C clock output.
	SDO1	—	CMOS	SPI1 data output.
	SD02	—	CMOS	SPI2 data output.
	SCK1	—	CMOS	SPI1 clock output.
	SCK2	—	CMOS	SPI2 clock output.
	TX/CK	—	CMOS	Asynchronous TX data/synchronous clock output.
	DT(3)	—	CMOS	EUSART synchronous data output.
	CLC1OUT	—	CMOS	Configurable Logic Cell 1 source output.
	CLC2OUT	—	CMOS	Configurable Logic Cell 2 source output.
	CLC3OUT	—	CMOS	Configurable Logic Cell 3 source output.
	CLC4OUT	—	CMOS	Configurable Logic Cell 4 source output.
	CLKR	—	CMOS	Clock Reference output.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C = Schmitt Trigger input with I^2C
HV = High Voltage XTAL = Crystal levels

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. See Register 12-1.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 12-2.

3: These I^2C functions are bidirectional. The output pin selections must be the same as the input pin selections.

TABLE 1-3: PIC16(L)F18345 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/ANA0/C1IN0+/DAC1OUT/ ICDDAT/ICSPDAT	RA0	TTL/ST	CMOS	General purpose I/O.
	ANA0	AN	—	ADC Channel A0 input.
	C1IN0+	AN	—	Comparator C1 positive input.
	DAC1OUT	—	AN	Digital-to-Analog Converter output.
	ICDDAT	TTL/ST	CMOS	In-Circuit Debug Data I/O.
	ICSPDAT	TTL/ST	CMOS	ICSP™ Data I/O.
RA1/ANA1/VREF+/C1IN0-/ C2IN0-/ DAC1REF+/$\overline{SS2}$(1)/ ICDCLK/ ICSPCLK	RA1	TTL/ST	CMOS	General purpose I/O.
	ANA1	AN	—	ADC Channel A1 input.
	VREF+	AN	—	ADC positive voltage reference input.
	C1IN0-	AN	—	Comparator C1 negative input.
	C2IN0-	AN	—	Comparator C2 negative input.
	DAC1REF+	AN	—	Digital-to-Analog Converter positive reference input.
	$\overline{SS2}$	TTL/ST	—	Slave Select 2 input.
	ICDCLK	TTL/ST	CMOS	In-Circuit Debug Clock I/O.
	ICSPCLK	TTL/ST	CMOS	ICSPTM Clock I/O.
RA2/ANA2/VREF-/ DAC1REF-/ T0CKI(1)/ CCP3(1)/CWG1IN(1)/ CWG2IN(1)/CLCIN0(1)/ INT(1)	RA2	TTL/ST	CMOS	General purpose I/O.
	ANA2	AN	—	ADC Channel A2 input.
	VREF-	AN	—	ADC negative voltage reference input.
	DAC1REF-	AN	—	Digital-to-Analog Converter negative reference input.
	T0CKI	TTL/ST	—	Timer0 Clock input.
	CCP3	TTL/ST	CMOS	Capture/Compare/PWM 3 input.
	CWG1IN	TTL/ST	—	Complementary Waveform Generator 1 input.
	CWG2IN	TTL/ST	—	Complementary Waveform Generator 2 input.
	CLCIN0	TTL/ST	—	Configurable Logic Cell 0 input.
	INT	TTL/ST	—	External interrupt input.
RA3/$\overline{MCLR}$/VPP	RA3	TTL/ST	CMOS	General purpose I/O.
	$\overline{MCLR}$	TTL/ST	—	Master Clear with internal pull-up.
	VPP	HV	—	Programming voltage.
RA4/ANA4/T1G(1)/T3G(1)/ T5G(1)/SOSCO/CCP4(1)/ CLKOUT/OSC2	RA4	TTL/ST	CMOS	General purpose I/O.
	ANA4	AN	—	ADC Channel A4 input.
	T1G	TTL/ST	—	Timer1 gate input.
	T3G	TTL/ST	—	Timer3 gate input.
	T5G	TTL/ST	—	Timer5 gate input.
	SOSCO	—	XTAL	Secondary Oscillator connection.
	CCP4	TTL/ST	CMOS	Capture/Compare/PWM 4 input.
	CLKOUT	—	CMOS	FOSC/4 output.
	OSC2	—	XTAL	Crystal/Resonator (LP, XT, HS modes).

Legend: AN = Analog input or output CMOS= CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C = Schmitt Trigger input with I^2C
HV = High Voltage XTAL = Crystal levels

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. See Register 12-2.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 12-2.

3: These I^2C functions are bidirectional. The output pin selections must be the same as the input pin selections.

TABLE 1-3: PIC16(L)F18345 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RA5/ANA5/T1CKI(1)/ T3CKI(1)/ T5CKI(1)/ SOSCIN/SOSCI/ CLKIN/OSC1	RA5	TTL/ST	CMOS	General purpose I/O.
	ANA5	AN	—	ADC Channel A5 input.
	T1CKI	TTL/ST	—	Timer1 Clock input.
	T3CKI	TTL/ST	—	Timer3 Clock input.
	T5CKI	TTL/ST	—	Timer5 Clock input.
	SOSCIN	TTL/ST	—	Secondary Oscillator input connection.
	SOSCI	XTAL	—	Secondary Oscillator connection.
	CLKIN	TTL/ST	—	External clock input.
	OSC1	XTAL	—	Crystal/Resonator (LP, XT, HS modes).
RB4/ANB4/SDI1(1)/ SDA1(1,3)/ CLCIN2(1)	RB4	TTL/ST	CMOS	General purpose I/O.
	ANB4	AN	—	ADC Channel B4 input.
	SDI1	TTL/ST	CMOS	SPI Data input 1.
	SDA1	I^2C™	OD	I^2C Data 1.
	CLCIN2	TTL/ST	—	Configurable Logic Cell 2 input.
RB5/ANB5/SDI2(1)/ SDA2(1,3)/ RX(1)/DT/CLCIN3(1)	RB5	TTL/ST	CMOS	General purpose I/O.
	ANB5	AN	—	ADC Channel B5 input.
	SDI2	TTL/ST	CMOS	SPI Data input 2.
	SDA2	I^2C	OD	I^2C Data 2.
	RX	TTL/ST	CMOS	EUSART asynchronous input.
	DT	TTL/ST	CMOS	EUSART synchronous data output.
	CLCIN3	TTL/ST	—	Configurable Logic Cell 3 input.
RB6/ANB6/SCK1(1)/ SCL1(1,3)	RB6	TTL/ST	CMOS	General purpose I/O.
	ANB6	AN	—	ADC Channel B6 input.
	SCK1	TTL/ST	CMOS	SPI Clock 1.
	SCL1	I^2C	OD	I^2C Clock 1.
RB7/ANB7/SCK2(1)/ SCL2(1,3)	RB7	TTL/ST	CMOS	General purpose I/O.
	ANB7	AN	—	ADC Channel B7 input.
	SCK2	TTL/ST	CMOS	SPI Clock 2.
	SCL2	I^2C	OD	I^2C Clock 2.
RC0/ANC0/C2IN0+	RC0	TTL/ST	CMOS	General purpose I/O.
	ANC0	AN	—	ADC Channel C0 input.
	C2IN0+	AN	—	Comparator C2 positive input.
RC1/ANC1/C1IN1-/C2IN1-	RC1	TTL/ST	CMOS	General purpose I/O.
	ANC1	AN	—	ADC Channel C1 input.
	C1IN1-	AN	—	Comparator C1 negative input.
	C2IN1-	AN	—	Comparator C2 negative input.
RC2/ANC2/C1IN2-/C2IN2-/ MDCIN1(1)	RC2	TTL/ST	CMOS	General purpose I/O.
	ANC2	AN	—	ADC Channel C2 input.
	C1IN2-	AN	—	Comparator C1 negative input.
	C2IN2-	AN	—	Comparator C2 negative input.
	MDCIN1	TTL/ST	—	Modular Carrier input 1.

Legend: AN = Analog input or output CMOS= CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C = Schmitt Trigger input with I^2C
HV = High Voltage XTAL = Crystal levels

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. See Register 12-2.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 12-2.

3: These I^2C functions are bidirectional. The output pin selections must be the same as the input pin selections.

TABLE 1-3: PIC16(L)F18345 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RC3/ANC3/C1IN3-/C2IN3-/ MDMIN(1)/ CCP2(1)/CLCIN1(1)/	RC3	TTL/ST	CMOS	General purpose I/O.
	ANC3	AN	—	ADC Channel C3 input.
	C1IN3-	AN	—	Comparator C1 negative input.
	C2IN3-	AN	—	Comparator C2 negative input.
	MDMIN	TTL/ST	—	Modular Source input.
	CCP2	TTL/ST	CMOS	Capture/Compare/PWM 2 input.
	CLCIN1	TTL/ST	—	Configurable Logic Cell 1 input.
RC4/ANC4	RC4	TTL/ST	CMOS	General purpose I/O.
	ANC4	AN	—	ADC Channel C4 input.
RC5/ANC5/MDCIN2(1)/ CCP1(1)	RC5	TTL/ST	CMOS	General purpose I/O.
	ANC5	AN	—	ADC Channel C5 input.
	MDCIN2	TTL/ST	—	Modular Carrier input 2.
	CCP1	TTL/ST	CMOS	Capture/Compare/PWM 1 input.
RC6/ANC6/$\overline{SS1}$(1)	RC6	TTL/ST	CMOS	General purpose I/O.
	ANC6	AN	—	ADC Channel C6 input.
	$\overline{SS1}$	TTL/ST	—	Slave Select 1 input.
RC7/ANC7	RC7	TTL/ST	CMOS	General purpose I/O.
	ANC7	AN	—	ADC Channel C7 input.
VDD	VDD	Power	—	Positive supply.
VSS	VSS	Power	—	Ground reference.

Legend: AN = Analog input or output CMOS= CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C = Schmitt Trigger input with I^2C
HV = High Voltage XTAL = Crystal levels

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. See Register 12-2.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 12-2.

3: These I^2C functions are bidirectional. The output pin selections must be the same as the input pin selections.

TABLE 1-3: PIC16(L)F18345 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
OUT(2)	C1	—	CMOS	Comparator C1 output.
	C2	—	CMOS	Comparator C2 output.
	NCO1	—	CMOS	Numerically Controlled Oscillator output.
	DSM	—	CMOS	Digital Signal Modulator output.
	TMR0	—	CMOS	Timer0 clock output.
	CCP1	—	CMOS	Capture/Compare/PWM 1 output.
	CCP2	—	CMOS	Capture/Compare/PWM 2 output.
	CCP3	—	CMOS	Capture/Compare/PWM 3 output.
	CCP4	—	CMOS	Capture/Compare/PWM 4 output.
	PWM5	—	CMOS	Pulse-Width Modulator 5 output.
	PWM6	—	CMOS	Pulse-Width Modulator 6 output.
	CWG1A	—	CMOS	Complementary Waveform Generator 1 output A.
	CWG2A	—	CMOS	Complementary Waveform Generator 2 output A.
	CWG1B	—	CMOS	Complementary Waveform Generator 1 output B.
	CWG2B	—	CMOS	Complementary Waveform Generator 2 output B.
	CWG1C	—	CMOS	Complementary Waveform Generator 1 output C.
	CWG2C	—	CMOS	Complementary Waveform Generator 2 output C.
	CWG1D	—	CMOS	Complementary Waveform Generator 1 output D.
	CWG2D	—	CMOS	Complementary Waveform Generator 2 output D.
	SDA1(3)	I^2C	OD	I^2C data output.
	SDA2(3)	I^2C	OD	I^2C data output.
	SCL1(3)	I^2C	OD	I^2C clock output.
	SCL2(3)	I^2C	OD	I^2C clock output.
	SDO1	—	CMOS	SPI1 data output.
	SD02	—	CMOS	SPI2 data output.
	SCK1	—	CMOS	SPI1 clock output.
	SCK2	—	CMOS	SPI2 clock output.
	TX/CK	—	CMOS	Asynchronous TX data/synchronous clock output.
	DT(3)	—	CMOS	EUSART synchronous data output.
	CLC1OUT	—	CMOS	Configurable Logic Cell 1 source output.
	CLC2OUT	—	CMOS	Configurable Logic Cell 2 source output.
	CLC3OUT	—	CMOS	Configurable Logic Cell 3 source output.
	CLC4OUT	—	CMOS	Configurable Logic Cell 4 source output.
	CLKR	—	CMOS	Clock Reference output.

Legend: AN = Analog input or output CMOS= CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C = Schmitt Trigger input with I^2C
HV = High Voltage XTAL = Crystal levels

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. See Register 12-2.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 12-2.

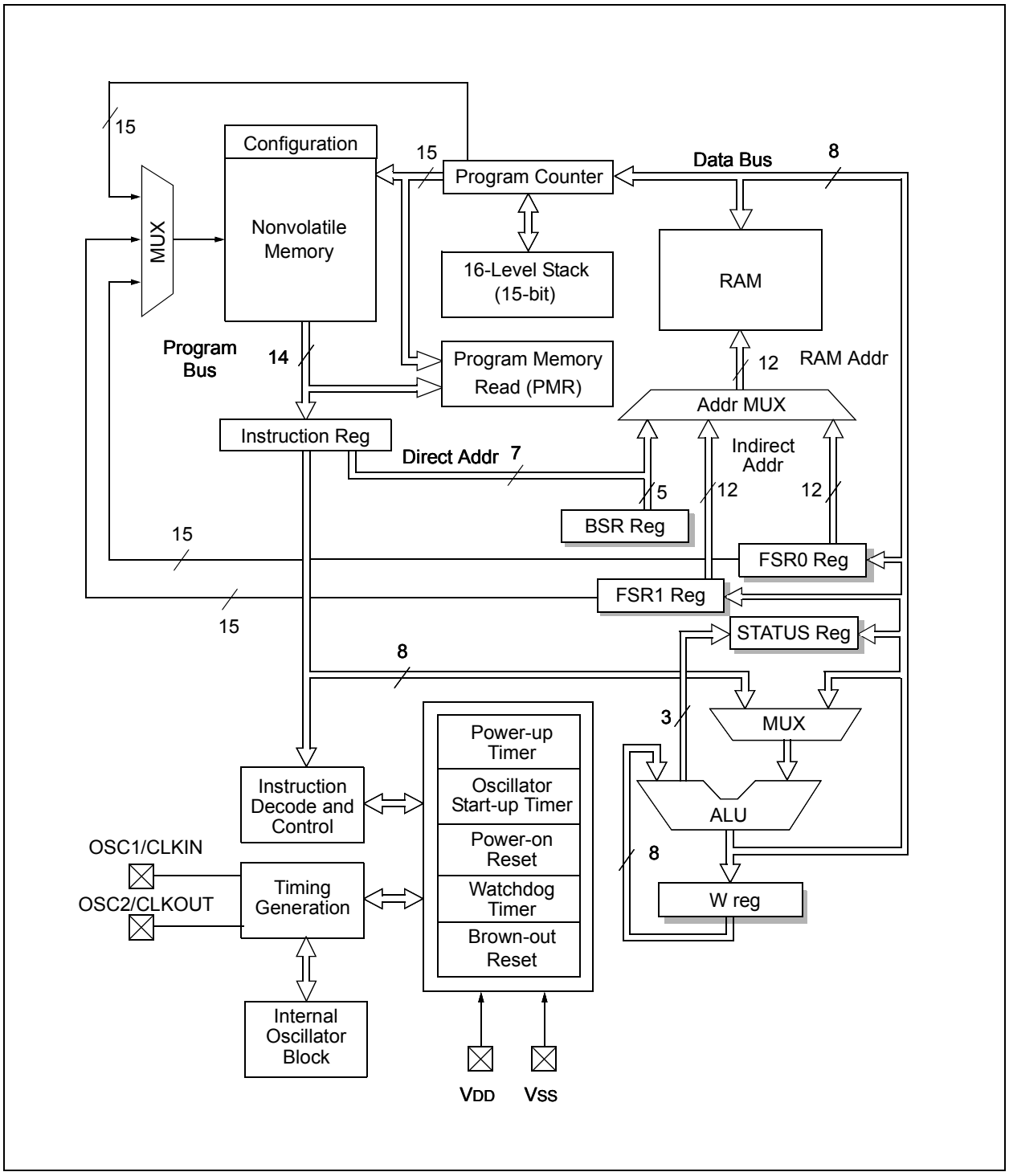
3: These I^2C functions are bidirectional. The output pin selections must be the same as the input pin selections.

2.0 ENHANCED MID-RANGE CPU

This family of devices contains an enhanced mid-range 8-bit CPU core. The CPU has 49 instructions. Interrupt capability includes automatic context saving. The hardware stack is 16-levels deep and has Overflow and Underflow Reset capability. Direct, Indirect, and Relative addressing modes are available. Two File Select Registers (FSRs) provide the ability to read program and data memory.

- Automatic Interrupt Context Saving
- 16-level Stack with Overflow and Underflow
- File Select Registers
- Instruction Set

FIGURE 2-1: CORE BLOCK DIAGRAM



2.1 Automatic Interrupt Context Saving

During interrupts, certain registers are automatically saved in shadow registers and restored when returning from the interrupt. This saves stack space and user code. See **Section 7.5 "Automatic Context Saving"** for more information.

2.2 16-Level Stack with Overflow and Underflow

These devices have a hardware stack memory 15 bits wide and 16 words deep. A Stack Overflow or Underflow will set the appropriate bit (STKOVF or STKUNF) in the PCON register, and if enabled, will cause a software Reset. See **Section 3.4 "Stack"** for more details.

2.3 File Select Registers

There are two 16-bit File Select Registers (FSR). FSRs can access all file registers, program memory, and data EEPROM, which allows one Data Pointer for all memory. When an FSR points to program memory, there is one additional instruction cycle in instructions using INDF to allow the data to be fetched. General purpose memory can now also be addressed linearly, providing the ability to access contiguous data larger than 80 bytes. There are also new instructions to support the FSRs. See **Section 3.5 "Indirect Addressing"** for more details.

2.4 Instruction Set

There are 49 instructions for the enhanced mid-range CPU to support the features of the CPU. See **Section 33.0 "Instruction Set Summary"** for more details.

3.0 MEMORY ORGANIZATION

These devices contain the following types of memory:

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

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

- PCL and PCLATH
- Stack
- Indirect Addressing
- NVMREG Access

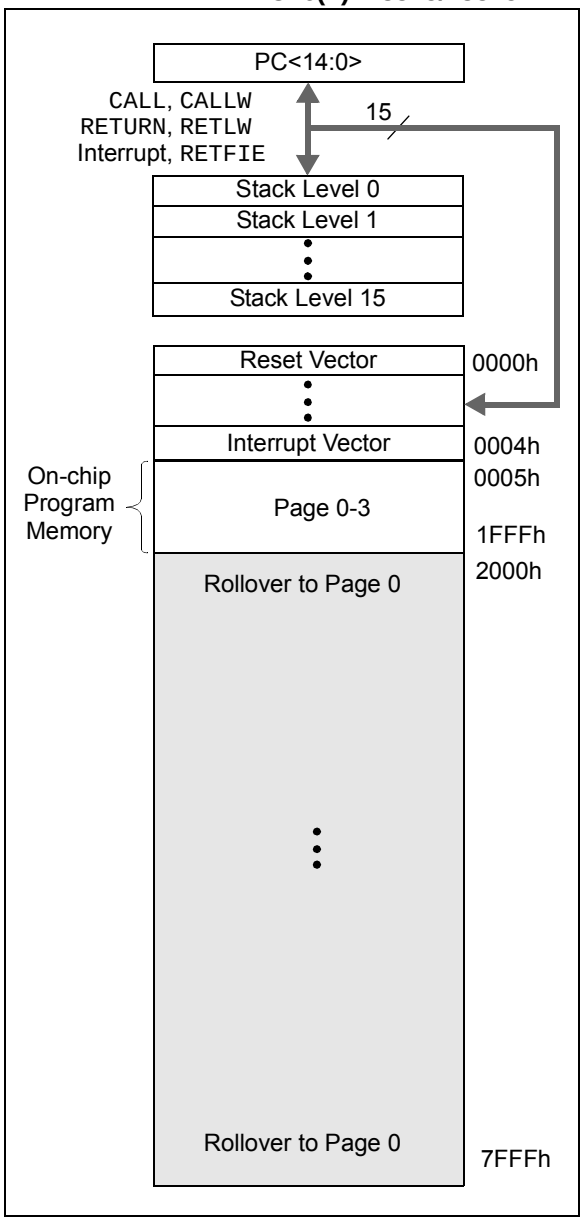
3.1 Program Memory Organization

The enhanced mid-range core has a 15-bit program counter capable of addressing 32K x 14 program memory space. Table 3-1 shows the memory sizes implemented. 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 Figure 3-1).

TABLE 3-1: DEVICE SIZES AND ADDRESSES

Device	Program Memory Size (Words)	Last Program Memory Address
PIC16(L)F18325/18345	8192	1FFFh

FIGURE 3-1: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F18325/18345



3.1.1 READING PROGRAM MEMORY AS DATA

There are two methods of accessing constants in program memory. The first method is to use tables of RETLW instructions. The second method is to set an FSR to point to the program memory.

3.1.1.1 RETLW Instruction

The RETLW instruction can be used to provide access to tables of constants. The recommended way to create such a table is shown in Example 3-1.

EXAMPLE 3-1: RETLW INSTRUCTION

```
constants
    BRW                 ;Add Index in W to
                        ;program counter to
                        ;select data
    RETLW DATA0         ;Index0 data
    RETLW DATA1         ;Index1 data
    RETLW DATA2
    RETLW DATA3


my_function
    ;… LOTS OF CODE…
    MOVLW       DATA_INDEX
    call constants
    ;… THE CONSTANT IS IN W
```

The BRW instruction makes this type of table very simple to implement. If your code must remain portable with previous generations of microcontrollers, the older table read method must be used because the BRW instruction is not available in some devices, such as the PIC16F6XX, PIC16F7XX, PIC16F8XX, and PIC16F9XX devices.

3.1.1.2 Indirect Read with FSR

The program memory can be accessed as data by setting bit 7 of the FSRxH register and reading the matching INDFx register. The MOVIW instruction will place the lower eight bits of the addressed word in the W register. Writes to the program memory cannot be performed via the INDF registers. Instructions that access the program memory via the FSR require one extra instruction cycle to complete. Example 3-2 demonstrates accessing the program memory via an FSR.

The HIGH directive will set bit 7 if a label points to a location in the program memory.

EXAMPLE 3-2: ACCESSING PROGRAM MEMORY VIA FSR

```
constants
    RETLW DATA0        ;Index0 data
    RETLW DATA1        ;Index1 data
    RETLW DATA2
    RETLW DATA3
my_function
    ;… LOTS OF CODE…
    MOVLW   LOW constants
    MOVWF   FSR1L
    MOVLW   HIGH constants
    MOVWF   FSR1H
    MOVIW   0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

3.2 Data Memory Organization

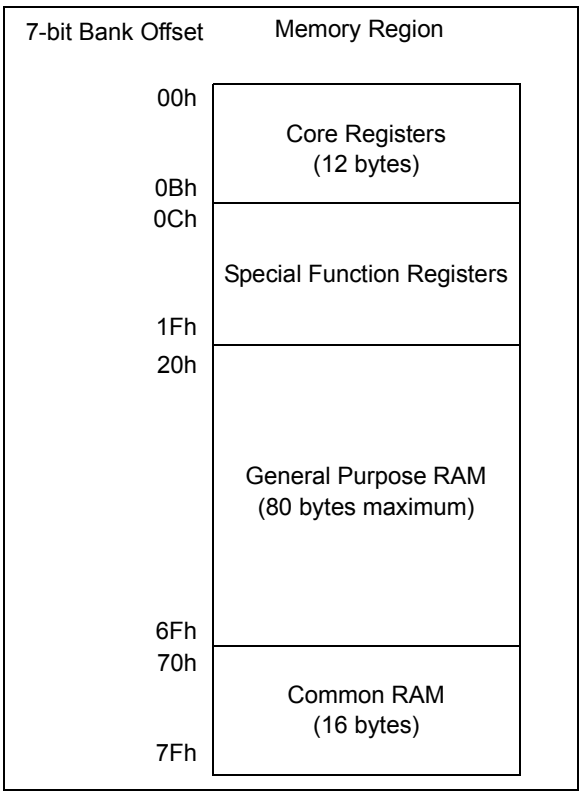
The data memory is partitioned into 32 memory banks with 128 bytes in each bank. Each bank consists of (Figure 3-2):

- 12 core registers
- Special Function Registers (SFR)
- Up to 80 bytes of General Purpose RAM (GPR)
- 16 bytes of common RAM

The active bank is selected by writing the bank number into the Bank Select Register (BSR). Unimplemented memory will read as '0'. All data memory can be accessed either directly (via instructions that use the file registers) or indirectly via the two File Select Registers (FSR). See **Section 3.5 "Indirect Addressing""** for more information.

Data memory uses a 12-bit address. The upper seven bits of the address define the Bank address and the lower five bits select the registers/RAM in that bank.

FIGURE 3-2: BANKED MEMORY PARTITIONING



3.2.1 CORE REGISTERS

The core registers contain the registers that directly affect the basic operation. The core registers occupy the first 12 addresses of every data memory bank (addresses x00h/x80h through x0Bh/x8Bh). These registers are listed below in Table 3-2. For detailed information, see Table 3-4.

TABLE 3-2: CORE REGISTERS

Addresses	BANKx
x00h or x80h	INDF0
x01h or x81h	INDF1
x02h or x82h	PCL
x03h or x83h	STATUS
x04h or x84h	FSR0L
x05h or x85h	FSR0H
x06h or x86h	FSR1L
x07h or x87h	FSR1H
x08h or x88h	BSR
x09h or x89h	WREG
x0Ah or x8Ah	PCLATH
x0Bh or x8Bh	INTCON

3.2.1.1 STATUS Register

The STATUS register, shown in Register 3-1, contains:

- The arithmetic status of the ALU
- The Reset status

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 $\overline{TO}$ and $\overline{PD}$ 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 (Refer to **Section 3.0 "Memory Organization"**).

Note 1: The C and DC bits operate as $\overline{Borrow}$ and $\overline{Digit\ Borrow}$ out bits, respectively, in subtraction.

REGISTER 3-1: STATUS: STATUS REGISTER

U-0	U-0	U-0	R-1/q	R-1/q	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC(1)	C(1)
bit 7							bit 0

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

bit 7-5 **Unimplemented:** Read as '0'

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

bit 3 **$\overline{PD}$:** 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/$\overline{Digit\ Borrow}$ 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/$\overline{Borrow}$ bit(1) (`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 $\overline{Borrow}$, 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-order or low-order bit of the source register.

3.2.2 SPECIAL FUNCTION REGISTER

The Special Function Registers are registers used by the application to control the desired operation of peripheral functions in the device. The Special Function Registers occupy the 20 bytes after the core registers of every data memory bank (addresses x0Ch/x8Ch through x1Fh/x9Fh), with the exception of banks 27, 28, and 29 (PPS and CLC registers). The registers associated with the operation of the peripherals are described in the appropriate peripheral chapter of this data sheet.

3.2.3 GENERAL PURPOSE RAM

There are up to 80 bytes of GPR in each data memory bank. The Special Function Registers occupy the 20 bytes after the core registers of every data memory bank (addresses x0Ch/x8Ch through x1Fh/x9Fh), with the exception of banks 27, 28, and 29 (PPS and CLC registers).

3.2.3.1 Linear Access to GPR

The general purpose RAM can be accessed in a non-banked method via the FSRs. This can simplify access to large memory structures. See **Section 3.5.2 "Linear Data Memory"** for more information.

3.2.4 COMMON RAM

There are 16 bytes of common RAM accessible from all banks.

3.2.5 DEVICE MEMORY MAPS

The memory maps for PIC16(L)F18325/18345 are as shown in Table 3-4.

TABLE 3-3: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (ALL BANKS)(1)

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
All Banks											
000h	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
001h	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
002h	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
003h	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu
004h	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
005h	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
006h	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
007h	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
008h	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000
009h	WREG	Working Register								0000 0000	uuuu uuuu
00Ah	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
00Bh	INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	00-- ---1	00-- ---1

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

Note 1: These registers can be accessed from any bank.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 0											
				CPU CORE REGISTERS; see Table 3-2 for specifics							
00Ch	PORTA			—	—	RA5	RA4	RA3	RA2	RA1	RA0
00Dh	PORTB	X	—	Unimplemented							
		—	X	RB7	RB6	RB5	RB4	—	—	—	—
00Eh	PORTC	X	—	—	—	RC5	RC4	RC3	RC2	RC1	RC0
		—	X	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
00Fh	—	—		Unimplemented							
010h	PIR0			—	—	TMR0IF	IOCIF	—	—	—	INTF
011h	PIR1			TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF
012h	PIR2			TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BCL2IF	TMR4IF	NCO1IF
013h	PIR3			OSFIF	CSWIF	TMR3GIF	TMR3IF	CLC4IF	CLC3IF	CLC2IF	CLC1IF
014h	PIR4			CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF
015h	TMR0L			TMR0L<7:0>							
016h	TMR0H			TMR0H<7:0>							
017h	T0CON0			T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>			
018h	T0CON1			T0CS<2:0>			T0ASYNC	T0CKPS<3:0>			
019h	TMR1L			TMR1L<7:0>							
01Ah	TMR1H			TMR1H<7:0>							
01Bh	T1CON			TMR1CS<1:0>		T1CKPS<1:0>		T1SOSC	$\overline{\text{T1SYNC}}$	—	TMR1ON
01Ch	T1GCON			TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>	
01Dh	TMR2			TMR2<7:0>							
01Eh	PR2			PR2<7:0>							
01Fh	T2CON			—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>	

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 1											
	CPU CORE REGISTERS; see Table 3-2 for specifics										
08Ch	TRISA			—	—	TRISA5	TRISA4	—	TRISA2	TRISA1	TRISA0
08Dh	TRISB	X	—	Unimplemented							
		—	X	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—
08Eh	TRISC	X	—	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
		—	X	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
08Fh	—	—		Unimplemented							
090h	PIE0			—	—	TMR0IE	IOCIE	—	—	—	INTE
091h	PIE1			TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE
092h	PIE2			TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BCL2IE	TMR4IE	NCO1IE
093h	PIE3			OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE
094h	PIE4			CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE
095h	—	—		Unimplemented							
096h	—	—		Unimplemented							
097h	WDTCON			—	—	WDTPS<4:0>					SWDTEN
098h	—	—		Unimplemented							
099h	—	—		Unimplemented							
09Ah	—	—		Unimplemented							
09Bh	ADRESL			ADRESL<7:0>							
09Ch	ADRESH			ADRESH<7:0>							
09Dh	ADCON0			CHS<5:0>						GO/$\overline{\text{DONE}}$	ADON
09Eh	ADCON1			ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>	
09Fh	ADACT			—	—	—	ADACT<4:0>				

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 2											
				CPU CORE REGISTERS; see Table 3-2 for specifics							
10Ch	LATA			—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0
10Dh	LATB	X	—	Unimplemented							
		—	X	LATB7	LATB6	LATB5	LATB4	—	—	—	—
10Eh	LATC	X	—	—	—	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
		—	X	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
10Fh	—	—		Unimplemented							
110h	—	—		Unimplemented							
111h	CM1CON0			C1ON	C1OUT	—	C1POL	—	C1SP	C1HYS	C1SYNC
112h	CM1CON1			C1INTP	C1INTN	C1PCH<2:0>			C1NCH<2:0>		
113h	CM2CON0			C2ON	C2OUT	—	C2POL	—	C2SP	C2HYS	C2SYNC
114h	CM2CON1			C2INTP	C2INTN	C2PCH<2:0>			C2MCH<2:0>		
115h	CMOUT			—	—	—	—	—	—	MC2OUT	MC1OUT
116h	BORCON			SBOREN	—	—	—	—	—	—	BORRDY
117h	FVRCON			FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>	
118h	DACCON0			DAC1EN	—	DAC1OE	—	DAC1PSS<1:0>		—	DAC1NS
119h	DACCON1			—	—	—	DAC1R<4:0>				
11Ah to 11Fh	—	—		Unimplemented							

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 3											
				CPU CORE REGISTERS; see Table 3-2 for specifics							
18Ch	ANSELA			—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0
18Dh	ANSELB	X	—	Unimplemented							
		—		ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—
18Eh	ANSELC		—	—	—	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0
		—	X	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0
18Fh	—	—		Unimplemented							
190h	—	—		Unimplemented							
191h	—	—		Unimplemented							
192h	—	—		Unimplemented							
193h	—	—		Unimplemented							
194h	—	—		Unimplemented							
195h	—	—		Unimplemented							
196h	—	—		Unimplemented							
197h	VREGCON(1)			—	—	—	—	—	—	VREGPM<1:0>	
198h	—	—		Unimplemented							
199h	RC1REG			RC1REG<7:0>							
19Ah	TX1REG			TX1REG<7:0>							
19Bh	SP1BRGL			SP1BRG<7:0>							
19Ch	SP1BRGH			SP1BRG<15:8>							
19Dh	RC1STA			SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
19Eh	TX1STA			CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TMRT	TX9D
19Fh	BAUD1CON			ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 4											
				CPU CORE REGISTERS; see Table 3-2 for specifics							
20Ch	WPUA			—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
20Dh	WPUB	X	—	Unimplemented							
		—	X	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—
20Eh	WPUC	X	—	—	—	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0
		—	X	WPUC7	WPUC6	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0
20Fh	—	—		Unimplemented							
210h	—	—		Unimplemented							
211h	SSP1BUF			SSP1BUF<7:0>							
212h	SSP1ADD			SSP1ADD<7:0>							
213h	SSP1MSK			SSP1MSK<7:0>							
214h	SSP1STAT			SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
215h	SSP1CON1			WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>			
216h	SSP1CON2			GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
217h	SSP1CON3			ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
218h	—	—		Unimplemented							
219h	SSP2BUF			SSP2BUF<7:0>							
21Ah	SSP2ADD			SSP2ADD<7:0>							
21Bh	SSP2MSK			SSP2MSK<7:0>							
21Ch	SSP2STAT			SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
21Dh	SSP2CON1			WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>			
21Eh	SSP2CON2			GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
21Fh	SSP2CON3			ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 5											
				CPU CORE REGISTERS; see Table 3-2 for specifics							
28Ch	ODCONA			—	—	ODCA5	ODCA4	—	ODCA2	ODCA1	ODCA0
28Dh	ODCONB	X	—	Unimplemented							
		—	X	ODCB7	ODCB6	ODCB5	ODCB4	—	—	—	—
28Eh	ODCONC	X	—	—	—	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0
		—	X	ODCC7	ODCC6	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0
28Fh	—	—		Unimplemented							
290h	—	—		Unimplemented							
291h	CCPR1L			CCPR1<7:0>							
292h	CCPR1H			CCPR1<15:8>							
293h	CCP1CON			CCP1EN	—	CCP1OUT	CCP1FMT	CCP1MODE<3:0>			
294h	CCP1CAP			—		—	—	CCP1CTS<3:0>			
295h	CCPR2L			CCPR2<7:0>							
296h	CCPR2H			CCPR2<15:8>							
297h	CCP2CON			CCP2EN	—	CCP2OUT	CCP2FMT	CCP2MODE<3:0>			
298h	CCP2CAP			—	—	—	—	CCP2CTS<3:0>			
299h	—	—		Unimplemented							
29Ah	—	—		Unimplemented							
29Bh	—	—		Unimplemented							
29Ch	—	—		Unimplemented							
29Dh	—	—		Unimplemented							
29Eh	—	—		Unimplemented							
29Fh	CCPTMRS			C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>	

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 6											
				CPU CORE REGISTERS; see Table 3-2 for specifics							
30Ch	SLRCONA			—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0
30Dh	SLRCONB	X	—	Unimplemented							
		—	X	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—
30Eh	SLRCONC	X	—	—	—	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0
		—	X	SLRC7	SLRC6	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0
30Fh	—	—		Unimplemented							
310h	—	—		Unimplemented							
311h	CCPR3L			CCPR3<7:0>							
312h	CCPR3H			CCPR3<15:8>							
313h	CCP3CON			CCP3EN	—	CCP3OUT	CCP3FMT	CCP3MODE<3:0>			
314h	CCP3CAP			—	—	—	—	CCP3CTS<3:0>			
315h	CCPR4L			CCPR4<7:0>							
316h	CCPR4H			CCPR4<15:8>							
317h	CCP4CON			CCP4EN	—	CCP4OUT	CCP4FMT	CCP4MODE<3:0>			
318h	CCP4CAP			—	—	—	—	CCP4CTS<3:0>			
319h to 31Fh	—	—		Unimplemented							

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 7											
				CPU CORE REGISTERS; see Table 3-2 for specifics							
38Ch	INLVLA			—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
38Dh	INLVLB	X	—	Unimplemented							
		—	X	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—
38Eh	INLVLC	X	—	—	—	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0
		—	X	INLVLC7	INLVLC6	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0
38Fh	—	—		Unimplemented							
390h	—	—		Unimplemented							
391h	IOCAP			—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0
392h	IOCAN			—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0
393h	IOCAF			—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0
394h	IOCBP	X	—	Unimplemented							
		—	X	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—
395h	IOCBN	X	—	Unimplemented							
		—	X	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—
396h	IOCBF	X	—	Unimplemented							
		—	X	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—
397h	IOCCP	X	—	—	—	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0
		—	X	IOCCP7	IOCCP6	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0
398h	IOCCN	X	—	—	—	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0
		—	X	IOCCN7	IOCCN6	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0
399h	IOCCF	X	—	—	—	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0
		—	X	IOCCF7	IOCCF6	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 7											
CPU CORE REGISTERS; see Table 3-2 for specifics											
39Ah	CLKRCON			CLKREN	—	—	CLKRDC<1:0>		CLKRDIV<2:0>		
39Bh	—	—		Unimplemented							
39Ch	MDCON			MDEN	—	—	MDOPOL	MDOUT	—	—	MDBIT
39Dh	MDSRC			—	—	—	—	MDMS<3:0>			
39Eh	MDCARH			—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>			
39Fh	MDCARL			—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>			

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 8											
CPU CORE REGISTERS; see Table 3-2 for specifics											
40Ch to 410h	—	—		Unimplemented							
411h	TMR3L			TMR3L<7:0>							
412h	TMR3H			TMR3H<7:0>							
413h	T3CON			TMR3CS<1:0>		T3CKPS<1:0>		T3SOSC	$\overline{\text{T3SYNC}}$	—	TMR3ON
414h	T3GCON			TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/$\overline{\text{DONE}}$	T3GVAL	T3GSS<1:0>	
415h	TMR4			TMR4<7:0>							
416h	PR4			PR4<7:0>							
417h	T4CON			—	T4OUTPS<3:0>				TMR4ON	T4CKPS<1:0>	
418h	TMR5L			TMR5L<7:0>							
419h	TMR5H			TMR5H<7:0>							
41Ah	T5CON			TMR5CS<1:0>		T5CKPS<1:0>		T5SOSC	$\overline{\text{T5SYNC}}$	—	TMR5ON
41Bh	T5GCON			TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/$\overline{\text{DONE}}$	T5GVAL	T5GSS<1:0>	
41Ch	TMR6			TMR6<7:0>							
41Dh	PR6			PR6<7:0>							
41Eh	T6CON			—	T6OUTPS<3:0>				TMR6ON	T6CKPS<1:0>	
41Fh	—	—		Unimplemented							

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 9											
	CPU CORE REGISTERS; see Table 3-2 for specifics										
48Ch to 497h	—	—		Unimplemented							
498h	NCO1ACCL			NCO1ACC<7:0>							
499h	NCO1ACCH			NCO1ACC<15:8>							
49Ah	NCO1ACCU			—	—	—	—	NCO1ACC<19:16>			
49Bh	NCO1INCL			NCO1INC<7:0>							
49Ch	NCO1INCH			NCO1INC<15:8>							
49Dh	NCO1INCU			—	—	—	—	NCO1INC<19:16>			
49Eh	NCO1CON			N1EN	—	N1OUT	N1POL	—	—	—	N1PFM
49Fh	NCO1CLK			N1PWS<2:0>			—	—	—	N1CKS<1:0>	

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 10-11											
CPU CORE REGISTERS; see Table 3-2 for specifics											
50Ch to 51Fh	—	—		Unimplemented							
58Ch to 59Fh	—	—		Unimplemented							
Bank 12											
60Ch	—	—		Unimplemented							
60Dh	—	—		Unimplemented							
60Eh	—	—		Unimplemented							
60Fh	—	—		Unimplemented							
610h	—	—		Unimplemented							
611h	—	—		Unimplemented							
612h	—	—		Unimplemented							
613h	—	—		Unimplemented							
614h	—	—		Unimplemented							
615h	—	—		Unimplemented							
616h	—	—		Unimplemented							
617h	PWM5DCL			PWM5DC<1:0>		—	—	—	—	—	—
618h	PWM5DCH			PWM5DC<9:2>							
619h	PWM5CON			PWM5EN	—	PWM5OUT	PWM5POL	—	—	—	—
61Ah	PWM6DCL			PWM6DC<1:0>		—	—	—	—	—	—
61Bh	PWM6DCH			PWM6DC<9:2>							
61Ch	PWM6CON			PWM6EN	—	PWM6OUT	PWM6POL	—	—	—	—
61Dh to 61Eh	—	—		Unimplemented							
61Fh	PWMTMRS			—	—	—	—	P6TSEL<1:0>		P5TSEL<1:0>	

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 13											
CPU CORE REGISTERS; see Table 3-2 for specifics											
68Ch	—	—		Unimplemented							
68Dh	—	—		Unimplemented							
68Eh	—	—		Unimplemented							
68Fh	—	—		Unimplemented							
690h	—	—		Unimplemented							
691h	CWG1CLKCON			—	—	—	—	—	—	—	CS
692h	CWG1DAT			—	—	—	—	DAT<3:0>			
693h	CWG1DBR			—	—	DBR<5:0>					
694h	CWG1DBF			—	—	DBF<5:0>					
695h	CWG1CON0			EN	LD	—	—	—	MODE<2:0>		
696h	CWG1CON1			—	—	IN	—	POLD	POLC	POLB	POLA
697h	CWG1AS0			SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—
698h	CWG1AS1			—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E
699h	CWG1STR			OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA
69Ah to 69Fh	—	—		Unimplemented							

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 14											
CPU CORE REGISTERS; see Table 3-2 for specifics											
70Ch	—	—		Unimplemented							
70Dh	—	—		Unimplemented							
70Eh	—	—		Unimplemented							
70Fh	—	—		Unimplemented							
710h	—	—		Unimplemented							
711h	CWG2CLKCON			—	—	—	—	—	—	—	CS
712h	CWG2DAT			—	—	—	—	DAT<3:0>			
713h	CWG2DBR			—	—	DBR<5:0>					
714h	CWG2DBF			—	—	DBF<5:0>					
715h	CWG2CON0			EN	LD	—	—	—	MODE<2:0>		
716h	CWG2CON1			—	—	IN	—	POLD	POLC	POLB	POLA
717h	CWG2AS0			SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—
718h	CWG2AS1			—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E
719h	CWG2STR			OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA
71Ah to 71Fh	—	—		Unimplemented							

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Banks 15-16											
CPU CORE REGISTERS; see Table 3-2 for specifics											
78Ch to 79FH	—	—		Unimplemented							
80Ch to 81Fh	—	—		Unimplemented							
Bank 17											
88Ch	—	—		Unimplemented							
88Dh	—	—		Unimplemented							
88Eh	—	—		Unimplemented							
88Fh	—	—		Unimplemented							
890h	—	—		Unimplemented							
891h	NVMADRL			NVMADR<7:0>							
892h	NVMADRH			—	NVMADR<14:8>						
893h	NVMDATL			NVMDAT<7:0>							
894h	NVMDATH			—	—	NVMDAT<13:8>					
895h	NVMCON1			—	NVMREGS	LWLO	FREE	WRERR	WREN	WR	RD
896h	NVMCON2			NVMCON2							
897h	—	—		Unimplemented							
898h	—	—		Unimplemented							
899h	—	—		Unimplemented							
89Ah	—	—		Unimplemented							
89Bh	PCON0			STKOVF	STKUNF	—	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
89Ch to 89Fh	—	—		Unimplemented							

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 18											
	CPU CORE REGISTERS; see Table 3-2 for specifics										
90Ch	—	—		Unimplemented							
90Dh	—	—		Unimplemented							
90Eh	—	—		Unimplemented							
90Fh	—	—		Unimplemented							
910h	—	—		Unimplemented							
911h	PMD0			SYSCMD	FVRMD	—	—	—	NVMMD	CLKRMD	IOCMD
912h	PMD1			NCOMD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD
913h	PMD2			—	DACMD	ADCMD	—	—	CMP2MD	CMP1MD	—
914h	PMD3			CWG2MD	CWG1MD	PWM6MD	PWM5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD
915h	PMD4			—	—	UART1MD	—	—	MSSP2MD	MSSP1MD	—
916h	PMD5			—	—	—	CLC4MD	CLC3MD	CLC2MD	CLC1MD	DSMMD
917h	—	—		Unimplemented							
918h	CPUDOZE			IDLEN	DOZEN	ROI	DOE	—	DOZE<2:0>		
919h	OSCCON1			—	NOSC<2:0>			NDIV<3:0>			
91Ah	OSCCON2			—	COSC<2:0>			CDIV<3:0>			
91Bh	OSCCON3			CSWHOLD	SOSCPWR	SOSCBE	ORDY	NOSCR	—	—	—
91Ch	OSCSTAT1			EXTOR	HFOR	—	LFOR	SOR	ADOR	—	PLLR
91Dh	OSCEN			EXTOEN	HFOEN	—	LFOEN	SOSCEN	ADOEN	—	—
91Eh	OSCTUNE			—	—	HFTUN<5:0>					
91Fh	OSCFRQ			—	—	—	—	—	HFFRQ<2:0>		

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Banks 19-27											
	CPU CORE REGISTERS; see Table 3-2 for specifics										
98Ch to 9EFh	—	—		Unimplemented							
A0Ch to A6Fh	—	—		Unimplemented							
A8Ch to AEFh	—	—		Unimplemented							
B0Ch to B6Fh	—	—		Unimplemented							
B8Ch to BEFh	—	—		Unimplemented							
C0Ch to C6Fh	—	—		Unimplemented							
C8Ch to CEFh	—	—		Unimplemented							
D0Ch to D6Fh	—	—		Unimplemented							
D8Ch to DEFh	—	—		Unimplemented							

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 28											
	CPU CORE REGISTERS; see Table 3-2 for specifics										
E0Ch	—	—		Unimplemented							
E0Dh	—	—		Unimplemented							
E0Eh	—	—		Unimplemented							
E0Fh	PPSLOCK			—	—	—	—	—	—	—	PPSLOCK
E10h	INTPPS			—	—	—	INTPPS<4:0>				
E11h	T0CKIPPS			—	—	—	T0CKIPPS<4:0>				
E12h	T1CKIPPS			—	—	—	T1CKIPPS<4:0>				
E13h	T1GPPS			—	—	—	T1GPPS<4:0>				
E14h	CCP1PPS			—	—	—	CCP1PPS<4:0>				
E15h	CCP2PPS			—	—	—	CCP2PPS<4:0>				
E16h	CCP3PPS			—	—	—	CCP3PPS<4:0>				
E17h	CCP4PPS	X	—	—	—	—	CCP4PPS<4:0>				
		—	X	—	—	—	CCP4PPS<4:0>				
E18h	CWG1PPS			—	—	—	CWG1PPS<4:0>				
E19h	CWG2PPS			—	—	—	CWG2PPS<4:0>				
E1Ah	MDCIN1PPS			—	—	—	MDCIN1PPS<4:0>				
E1Bh	MDCIN2PPS			—	—	—	MDCIN2PPS<4:0>				
E1Ch	MDMINPPS			—	—	—	MDMINPPS<4:0>				
E1Dh	SSP2CLKPPS	X	—	—	—	—	SSP2CLKPPS<4:0>				
		—	X	—	—	—	SSP2CLKPPS<4:0>				
E1Eh	SSP2DATPPS	X	—	—	—	—	SSP2DATPPS<4:0>				
		—	X	—	—	—	SSP2DATPPS<4:0>				
E1Fh	SSP2SSPPS	X	—	—	—	—	SSP2SSPPS<4:0>				
		—	X	—	—	—	SSP2SSPPS<4:0>				
E20h	SSP1CLKPPS	X	—	—	—	—	SSP1CLKPPS<4:0>				
		—	X	—	—	—	SSP1CLKPPS<4:0>				

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 28											
				CPU CORE REGISTERS; see Table 3-2 for specifics							
E21h	SSP1DATPPS	X	—	—	—	—	SSP1DATPPS<4:0>				
		—	X	—	—	—	SSP1DATPPS<4:0>				
E22h	SSP1SSPPS	X	—	—	—	—	SSP1SSPPS<4:0>				
		—	X	—	—	—	SSP1SSPPS<4:0>				
E23h	—	—		Unimplemented							
E24h	RXPPS	X	—	—	—	—	RXPPS<4:0>				
		—	X	—	—	—	RXPPS<4:0>				
E25h	TXPPS	X	—	—	—	—	TXPPS<4:0>				
		—	X	—	—	—	TXPPS<4:0>				
E26h	—	—		Unimplemented							
E27h	—	—		Unimplemented							
E28h	CLCIN0PPS	X	—	—	—	—	CLCIN0PPS<4:0>				
		—	X	—	—	—	CLCIN0PPS<4:0>				
E29h	CLCIN1PPS	X	—	—	—	—	CLCIN1PPS<4:0>				
		—	X	—	—	—	CLCIN1PPS<4:0>				
E2Ah	CLCIN2PPS	X	—	—	—	—	CLCIN2PPS<4:0>				
		—	X	—	—	—	CLCIN2PPS<4:0>				
E2Bh	CLCIN3PPS	X	—	—	—	—	CLCIN3PPS<4:0>				
		—	X	—	—	—	CLCIN3PPS<4:0>				
E2Ch	T3CKIPPS	X	—	—	—	—	T3CKIPPS<4:0>				
		—	X	—	—	—	T3CKIPPS<4:0>				
E2Dh	T3GPPS	X	—	—	—	—	T3GPPS<4:0>				
		—	X	—	—	—	T3GPPS<4:0>				
E2Eh	T5CKIPPS	X	—	—	—	—	T5CKIPPS<4:0>				
		—	X	—	—	—	T5CKIPPS<4:0>				
E2Fh	T5GPPS	X	—	—	—	—	T5GPPS<4:0>				
		—	X	—	—	—	T5GPPS<4:0>				

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 29											
CPU CORE REGISTERS; see Table 3-2 for specifics											
E8Dh	—	—		Unimplemented							
E8Eh	—	—		Unimplemented							
E8Fh	—	—		Unimplemented							
E90h	RA0PPS			—	—	—	RA0PPS<4:0>				
E91h	RA1PPS			—	—	—	RA1PPS<4:0>				
E92h	RA2PPS			—	—	—	RA2PPS<4:0>				
E93h	—	—		Unimplemented							
E94h	RA4PPS			—	—	—	RA4PPS<4:0>				
E95h	RA5PPS			—	—	—	RA5PPS<4:0>				
E96h	—	—		Unimplemented							
E97h	—	—		Unimplemented							
E98h	—	—		Unimplemented							
E99h	—	—		Unimplemented							
E9Ah	—	—		Unimplemented							
E9Bh	—	—		Unimplemented							
E9Ch	RB4PPS	X	—	Unimplemented							
		—	X	—	—	—	RB4PPS<4:0>				
E9Dh	RB5PPS	X	—	Unimplemented							
		—	X	—	—	—	RB5PPS<4:0>				
E9Eh	RB6PPS	X	—	Unimplemented							
		—	X	—	—	—	RB6PPS<4:0>				
E9Fh	RB7PPS	X	—	Unimplemented							
		—	X	—	—	—	RB7PPS<4:0>				

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 29											
				CPU CORE REGISTERS; see Table 3-2 for specifics							
EA0h	RC0PPS			—	—	—	RC0PPS<4:0>				
EA1h	RC1PPS			—	—	—	RC1PPS<4:0>				
EA2h	RC2PPS			—	—	—	RC2PPS<4:0>				
EA3h	RC3PPS			—	—	—	RC3PPS<4:0>				
EA4h	RC4PPS			—	—	—	RC4PPS<4:0>				
EA5h	RC5PPS			—	—	—	RC5PPS<4:0>				
EA6h	RC6PPS	X	—	Unimplemented							
		—	X	—	—	—	RC6PPS<4:0>				
EA7h	RC7PPS	X	—	Unimplemented							
		—	X	—	—	—	RC7PPS<4:0>				

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 30											
				CPU CORE REGISTERS; see Table 3-2 for specifics							
F0Ch	—	—		Unimplemented							
F0Dh	—	—		Unimplemented							
F0Eh	—	—		Unimplemented							
F0Fh	CLCDATA			—	—	—	—	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OU
F10h	CLC1CON			LC1EN	—	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>		
F11h	CLC1POL			LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1PO
F12h	CLC1SEL0			—	—	LC1D1S<5:0>					
F13h	CLC1SEL1			—	—	LC1D2S<5:0>					
F14h	CLC1SEL2			—	—	LC1D3S<5:0>					
F15h	CLC1SEL3			—	—	LC1D4S<5:0>					
F16h	CLC1GLS0			LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1
F17h	CLC1GLS1			LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1
F18h	CLC1GLS2			LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1
F19h	CLC1GLS3			LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1
F1Ah	CLC2CON			LC2EN	—	LC2OUT	LC2INTP	LC2INTN	LC2MODE<2:0>		
F1Bh	CLC2POL			LC2POL	—	—	—	LC2G4POL	LC2G3POL	LC2G2POL	LC2G1PO
F1Ch	CLC2SEL0			—	—	LC2D1S<5:0>					
F1Dh	CLC2SEL1			—	—	LC2D2S<5:0>					
F1Eh	CLC2SEL2			—	—	LC2D3S<5:0>					
F1Fh	CLC2SEL3			—	—	LC2D4S<5:0>					

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325 / PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 30										
			CPU CORE REGISTERS; see Table 3-2 for specifics							
F20h	CLC2GLS0		LC2G1D4T	LC2G1D4N	LC2G1D3T	LC2G1D3N	LC2G1D2T	LC2G1D2N	LC2G1D1T	LC2G1D1
F21h	CLC2GLS1		LC2G2D4T	LC2G2D4N	LC2G2D3T	LC2G2D3N	LC2G2D2T	LC2G2D2N	LC2G2D1T	LC2G2D1
F22h	CLC2GLS2		LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1
F23h	CLC2GLS3		LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1
F24h	CLC3CON		LC3EN	—	LC3OUT	LC3INTP	LC3INTN	LC3MODE<2:0>		
F25h	CLC3POL		LC3POL	—	—	—	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1PO
F26h	CLC3SEL0		—	—	LC3D1S<5:0>					
F27h	CLC3SEL1		—	—	LC3D2S<5:0>					
F28h	CLC3SEL2		—	—	LC3D3S<5:0>					
F29h	CLC3SEL3		—	—	LC3D4S<5:0>					
F2Ah	CLC3GLS0		LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1
F2Bh	CLC3GLS1		LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1
F2Ch	CLC3GLS2		LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1
F2Dh	CLC3GLS3		LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1
F2Eh	CLC4CON		LC4EN	—	LC4OUT	LC4INTP	LC4INTN	LC4MODE<2:0>		
F2Fh	CLC4POL		LC4POL	—	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1PO
F30h	CLC4SEL0		—	—	LC4D1S<5:0>					
F31h	CLC4SEL1		—	—	LC4D2S<5:0>					
F32h	CLC4SEL2		—	—	LC4D3S<5:0>					
F33h	CLC4SEL3		—	—	LC4D4S<5:0>					
F34h	CLC4GLS0		LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1
F35h	CLC4GLS1		LC4G2D4T	LC4G2D4N	LC4G2D3T	LC4G2D3N	LC4G2D2T	LC4G2D2N	LC4G2D1T	LC4G2D1
F36h	CLC4GLS2		LC4G3D4T	LC4G3D4N	LC4G3D3T	LC4G3D3N	LC4G3D2T	LC4G3D2N	LC4G3D1T	LC4G3D1
F37h	CLC4GLS3		LC4G4D4T	LC4G4D4N	LC4G4D3T	LC4G4D3N	LC4G4D2T	LC4G4D2N	LC4G4D1T	LC4G4D1

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 3-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 31 — only accessible from Debug Executive, unless otherwise specified											
CPU CORE REGISTERS; see Table 3-2 for specifics											
F8Ch to FE3h	—	—	—	Unimplemented							
FE4h(2)	STATUS_SHAD			—	—	—	—	—	Z	DC	C
FE5h(2)	WREG_SHAD			Working Register Normal (Non-ICD) Shadow							
FE6h(2)	BSR_SHAD			—	—	—	Bank Select Register Normal (Non-ICD) Shadow				
FE7h(2)	PCLATH_SHAD			—	Program Counter Latch High Register Normal (Non-ICD) Shadow						
FE8h(2)	FSR0L_SHAD			Indirect Data Memory Address 0 Low Pointer Normal (Non-ICD) Shadow							
FE9h(2)	FSR0H_SHAD			Indirect Data Memory Address 0 High Pointer Normal (Non-ICD) Shadow							
FEAh(2)	FSR1L_SHAD			Indirect Data Memory Address 1 Low Pointer Normal (Non-ICD) Shadow							
FEBh(2)	FSR1H_SHAD			Indirect Data Memory Address 1 High Pointer Normal (Non-ICD) Shadow							
FECh	—	—		Unimplemented							
FEDh(2)	STKPTR			—	—	—	Current Stack pointer				
FEEh(2)	TOSL			Top of Stack Low byte							
FEFh(2)	TOSH			—	Top of Stack High byte						

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

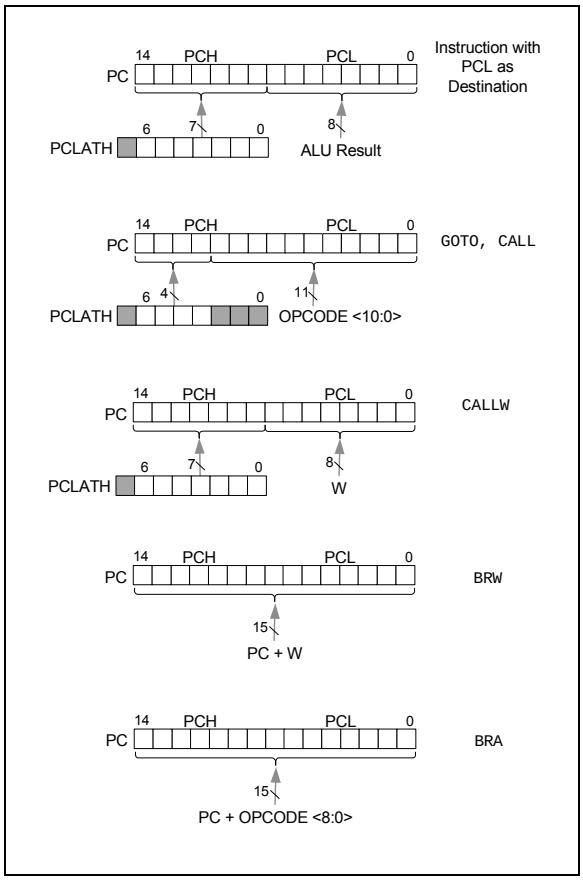
Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

3.3 PCL and PCLATH

The Program Counter (PC) is 15-bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<14:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 3-3 shows the five situations for the loading of the PC.

FIGURE 3-3: LOADING OF PC IN DIFFERENT SITUATIONS



3.3.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<14:8> 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 seven bits to the PCLATH register. When the lower eight bits are written to the PCL register, all 15 bits of the program counter will change to the values contained in the PCLATH register and those being written to the PCL register.

3.3.2 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When performing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to Application Note AN556, *"Implementing a Table Read"* (DS00556).

3.3.3 COMPUTED FUNCTION CALLS

A computed function CALL allows programs to maintain tables of functions and provide another way to execute state machines or look-up tables. When performing a table read using a computed function CALL, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block).

If using the CALL instruction, the PCH<2:0> and PCL registers are loaded with the operand of the CALL instruction. PCH<6:3> is loaded with PCLATH<6:3>.

The CALLW instruction enables computed calls by combining PCLATH and $\overline{W}$ to form the destination address. A computed CALLW is accomplished by loading the $\overline{W}$ register with the desired address and executing CALLW. The PCL register is loaded with the value of $\overline{W}$ and PCH is loaded with PCLATH.

3.3.4 BRANCHING

The branching instructions add an offset to the PC. This allows relocatable code and code that crosses page boundaries. There are two forms of branching, BRW and BRA. The PC will have incremented to fetch the next instruction in both cases. When using either branching instruction, a PCL memory boundary may be crossed.

If using BRW, load the $\overline{W}$ register with the desired unsigned address and execute BRW. The entire PC will be loaded with the address PC + 1 + $\overline{W}$.

If using BRA, the entire PC will be loaded with PC + 1 + the signed value of the operand of the BRA instruction.

3.4 Stack

All devices have a 16-level x 15-bit wide hardware stack (refer to Figure 3-4 through Figure 3-7). The stack space is not part of either program or data space. The PC is PUSHed onto the stack when CALL or CALLW instructions are 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 if the STVREN bit is programmed to '0' (Configuration Words). This means that after the stack has been PUSHed sixteen times, the seventeenth PUSH overwrites the value that was stored from the first PUSH. The eighteenth PUSH overwrites the second PUSH (and so on). The STKOVF and STKUNF flag bits will be set on an Overflow/Underflow, regardless of whether the Reset is enabled.

Note 1:	There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, CALLW, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

3.4.1 ACCESSING THE STACK

The stack is available through the TOSH, TOSL and STKPTR registers. STKPTR is the current value of the Stack Pointer. TOSH:TOSL register pair points to the TOP of the stack. Both registers are read/writable. TOS is split into TOSH and TOSL due to the 15-bit size of the PC. To access the stack, adjust the value of STKPTR, which will position TOSH:TOSL, then read/write to TOSH:TOSL. STKPTR is five bits to allow detection of overflow and underflow.

Note:	Care should be taken when modifying the STKPTR while interrupts are enabled.

During normal program operation, CALL, CALLW and Interrupts will increment STKPTR while RETLW, RETURN, and RETFIE will decrement STKPTR. At any time, STKPTR can be inspected to see how much stack is left. The STKPTR always points at the currently used place on the stack. Therefore, a CALL or CALLW will increment the STKPTR and then write the PC, and a return will unload the PC and then decrement the STKPTR.

Reference Figure 3-4 through Figure 3-7 for examples of accessing the stack.

FIGURE 3-4: ACCESSING THE STACK EXAMPLE 1

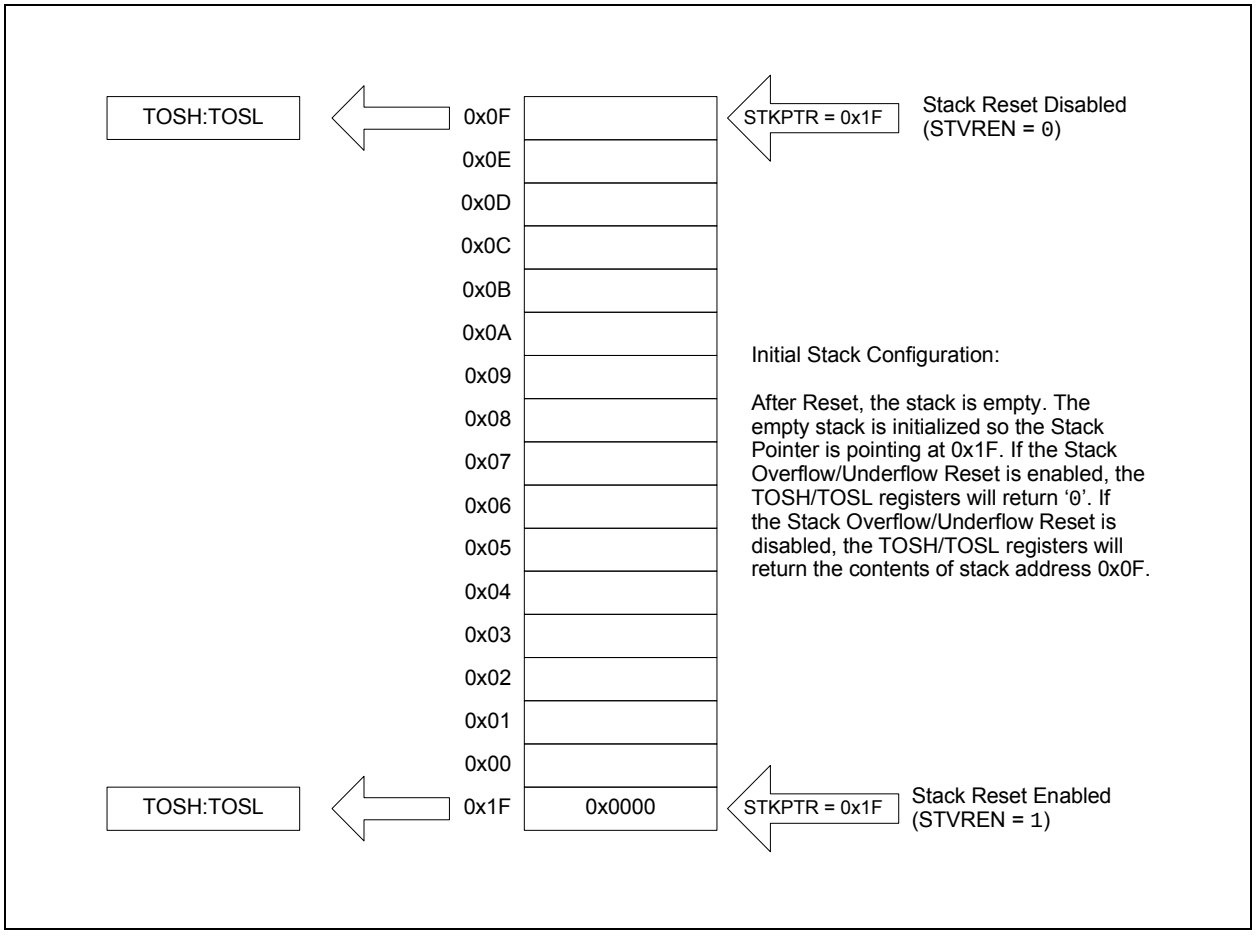


FIGURE 3-5: ACCESSING THE STACK EXAMPLE 2

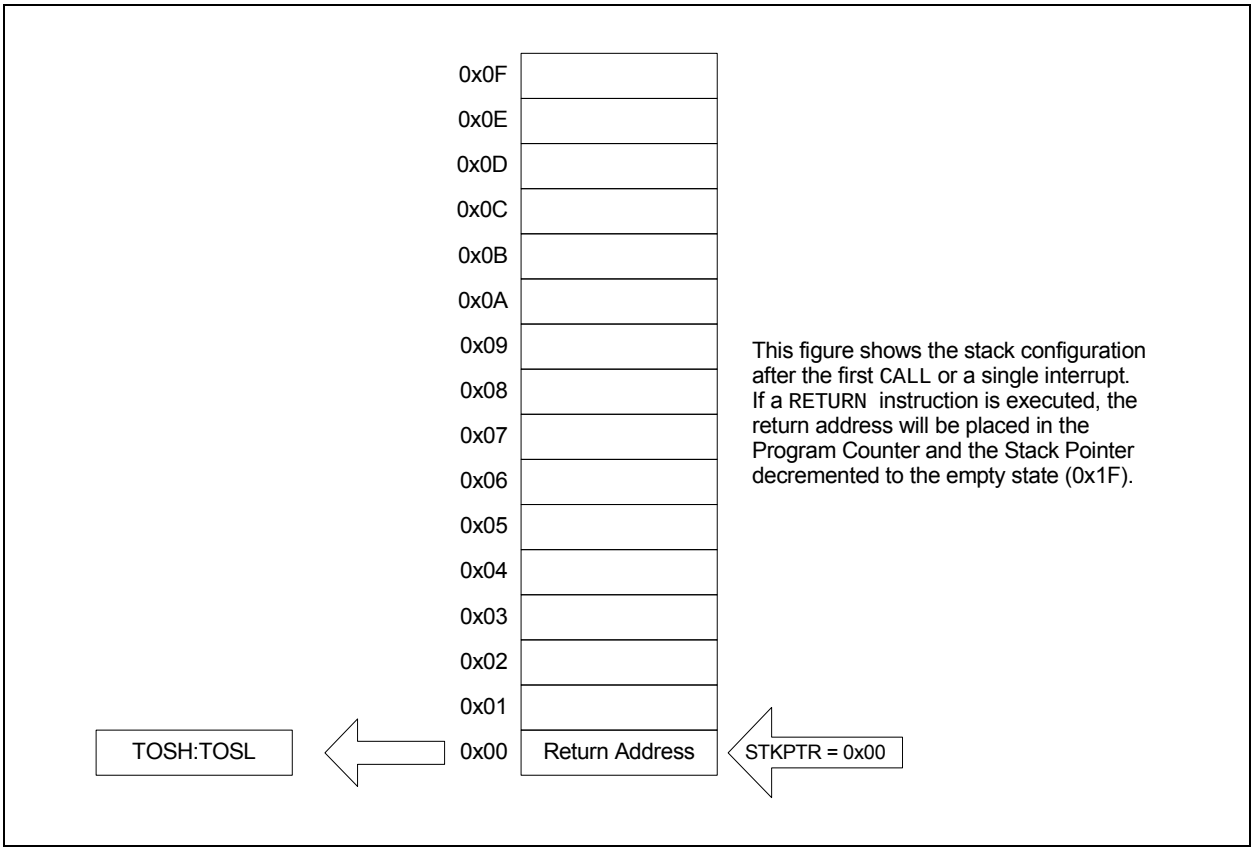


FIGURE 3-6: ACCESSING THE STACK EXAMPLE 3

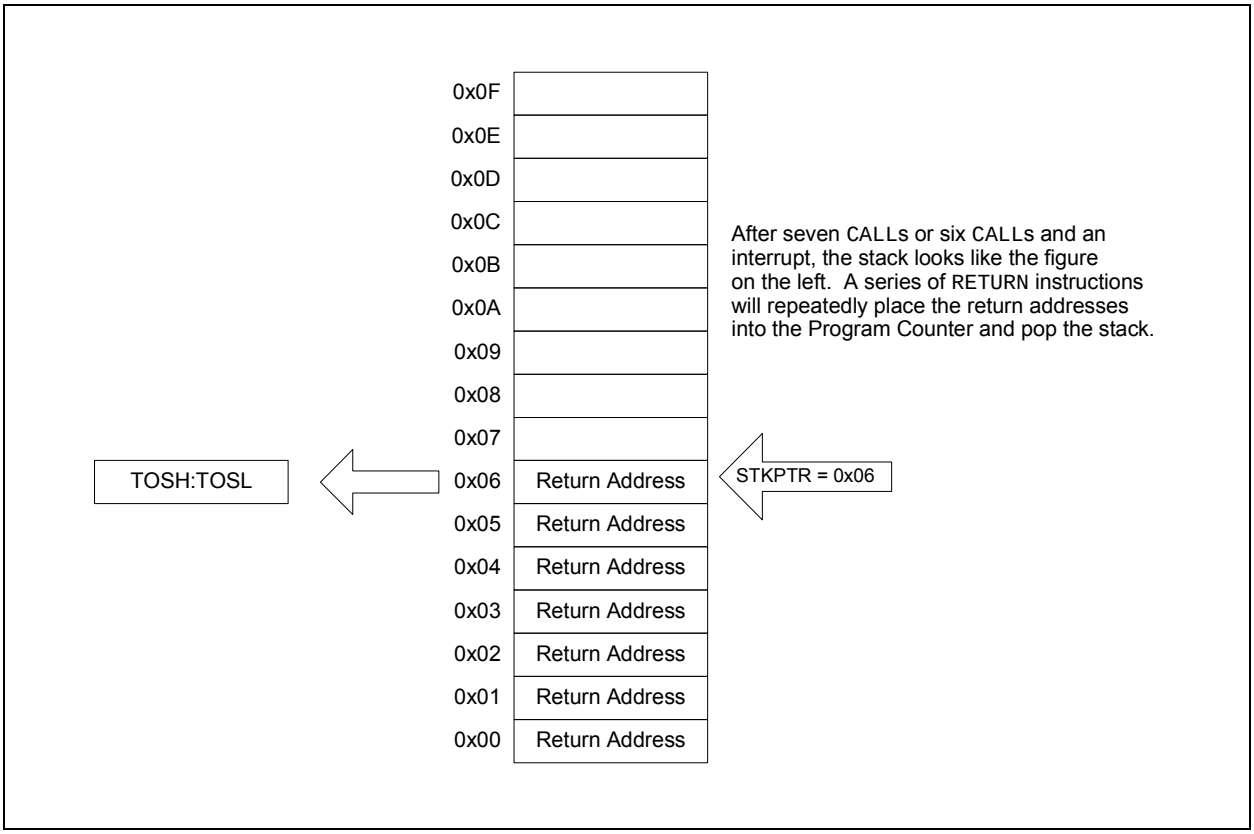
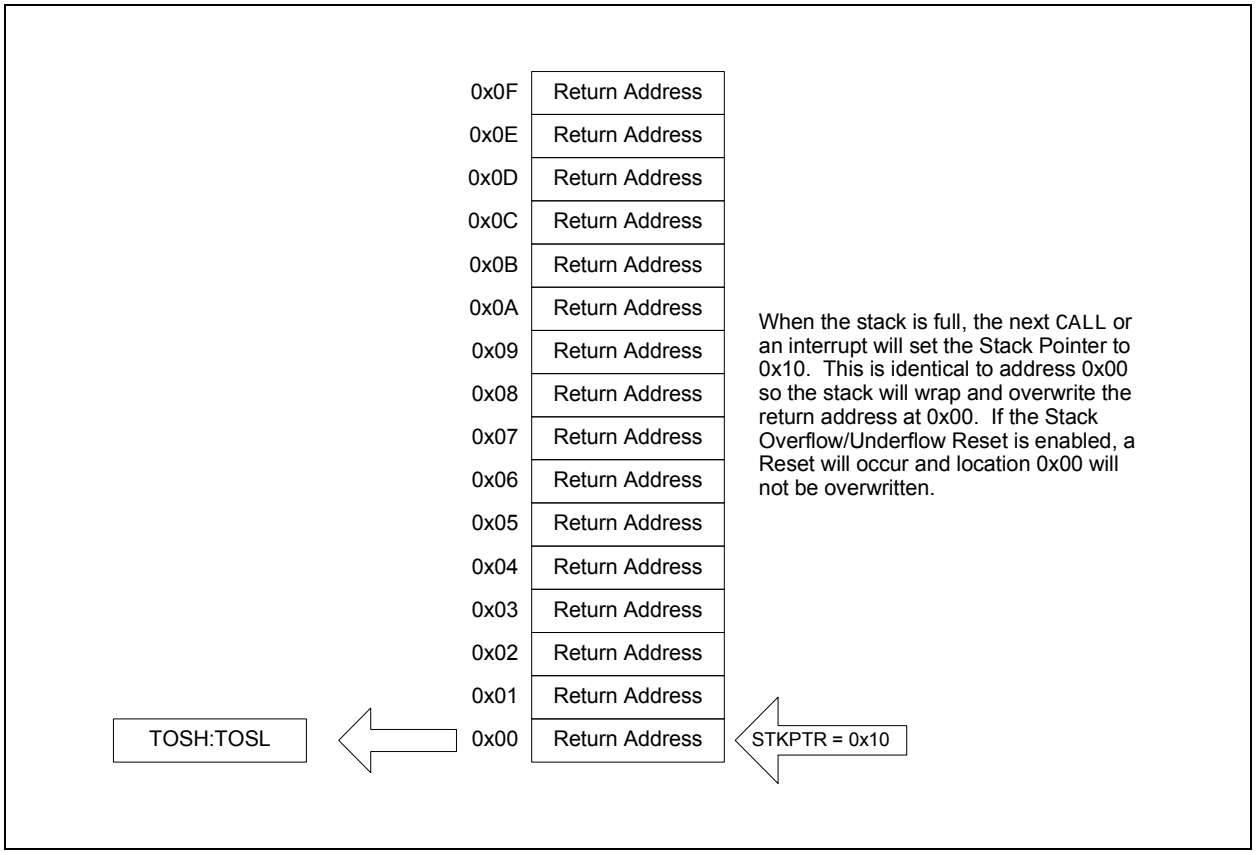


FIGURE 3-7: ACCESSING THE STACK EXAMPLE 4



3.4.2 OVERFLOW/UNDERFLOW RESET

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

3.5 Indirect Addressing

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

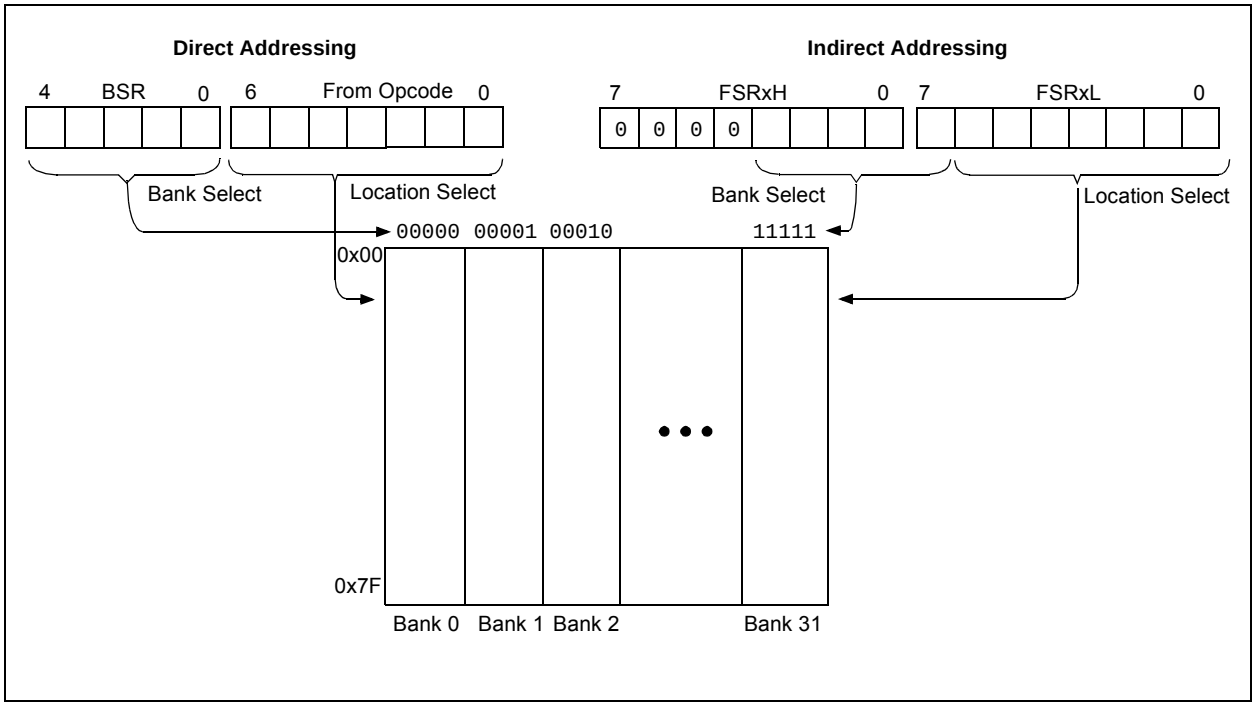
The FSR registers form a 16-bit address that allows an addressing space with 65536 locations. These locations are divided into four memory regions:

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

3.5.1 TRADITIONAL DATA MEMORY

The traditional data memory is a region from FSR address 0x000 to FSR address 0xFFF. The addresses correspond to the absolute addresses of all SFR, GPR and common registers.

FIGURE 3-8: TRADITIONAL DATA MEMORY MAP



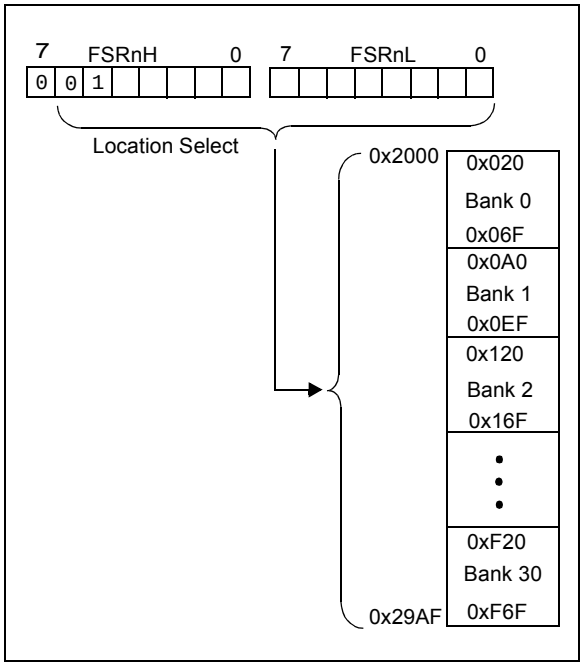
3.5.2 LINEAR DATA MEMORY

The linear data memory is the region from FSR address 0x2000 to FSR address 0x29AF. This region is a virtual region that points back to the 80-byte blocks of GPR memory in all the banks.

Unimplemented memory reads as 0x00. Use of the linear data memory region allows buffers to be larger than 80 bytes because incrementing the FSR beyond one bank will go directly to the GPR memory of the next bank.

The 16 bytes of common memory are not included in the linear data memory region.

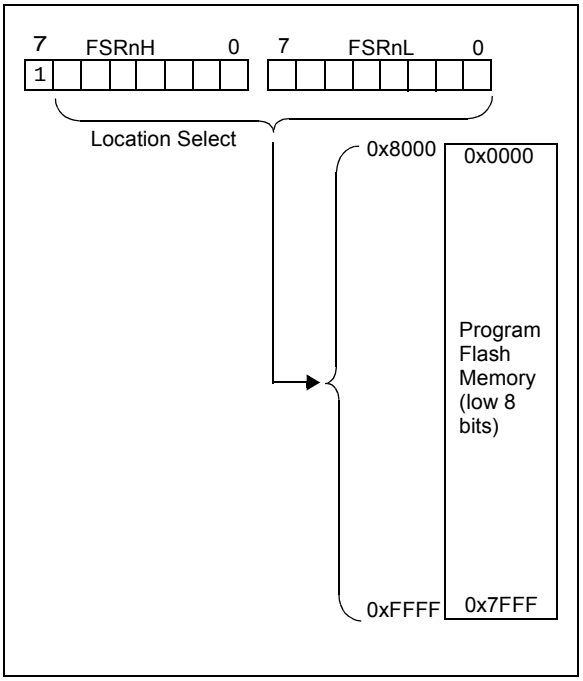
FIGURE 3-9: LINEAR DATA MEMORY MAP



3.5.3 PROGRAM FLASH MEMORY

To make constant data access easier, the entire program Flash memory is mapped to the upper half of the FSR address space. When the MSB of FSRnH is set, the lower 15 bits are the address in program memory which will be accessed through INDF. Only the lower eight bits of each memory location is accessible via INDF. Writing to the program Flash memory cannot be accomplished via the FSR/INDF interface. All instructions that access program Flash memory via the FSR/INDF interface will require one additional instruction cycle to complete.

FIGURE 3-10: PROGRAM FLASH MEMORY MAP



4.0 DEVICE CONFIGURATION

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

4.1 Configuration Words

There are several Configuration Word bits that allow different oscillator and memory protection options. These are implemented as Configuration Word 1 at 8007h, Configuration Word 2 at 8008h, Configuration Word 3 at 8009h, and Configuration Word 4 at 800Ah.

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

4.2 Register Definitions: Configuration Words

REGISTER 4-1: CONFIGURATION WORD 1: OSCILLATORS

R/P-1	U-1	R/P-1	U-1	U-1	R/P-1
FCMEN	—	CSWEN	—	—	$\overline{\text{CLKOUTEN}}$
bit 13					bit 8

U-1	R/P-1	R/P-1	R/P-1	U-1	R/P-1	R/P-1	R/P-1
—	RSTOSC2	RSTOSC1	RSTOSC0	—	FEXTOSC2	FEXTOSC1	FEXTOSC0
bit 7							bit 0

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '1'
'0' = Bit is cleared	'1' = Bit is set	n = Value when blank or after Bulk Erase

bit 13 **FCMEN:** Fail-Safe Clock Monitor Enable bit
1 = ON FSCM timer enabled
0 = OFF FSCM timer disabled

bit 12 **Unimplemented:** Read as '1'

bit 11 **CSWEN:** Clock Switch Enable bit
1 = ON Writing to NOSC and NDIV is allowed
0 = OFF The NOSC and NDIV bits cannot be changed by user software

bit 10-9 **Unimplemented:** Read as '1'

bit 8 **$\overline{\text{CLKOUTEN}}$:** Clock Out Enable bit
If FEXTOSC = EC, HS, HT or LP, then this bit is ignored; otherwise:
1 = OFF CLKOUT function is disabled; I/O or oscillator function on OSC2
0 = ON CLKOUT function is enabled; FOSC/4 clock appears at OSC2

bit 7 **Unimplemented:** Read as '1'

bit 6-4 **RSTOSC<2:0>:** Power-up Default Value for COSC bits
This value is the Reset default value for COSC, and selects the oscillator first used by user software
111 = EXT1X EXTOSC operating per FEXTOSC<2:0> bits
110 = HFINT1 HFINTOSC (1 MHz)
101 = Reserved
100 = LFINT LFINTOSC
011 = SOSC SOSC (32.768 kHz)
010 = Reserved
001 = EXT4X EXTOSC with 4x PLL; EXTOSC operating per FEXTOSC<2:0> bits
000 = HFINT32 HFINTOSC (32 MHz)

bit 3 **Unimplemented:** Read as '1'

bit 2-0 **FEXTOSC<2:0>:** FEXTOSC External Oscillator Mode Selection bits
111 = ECH EC (External Clock) above 8 MHz
110 = ECM EC (External Clock) for 100 kHz to 8 MHz
101 = ECL EC (External Clock) below 100 kHz
100 = OFF Oscillator not enabled
011 = Unimplemented
010 = HS HS (Crystal oscillator) above 8 MHz
001 = XT HT (Crystal oscillator) above 100 kHz, below 8 MHz
000 = LP LP (Crystal oscillator) optimized for 32.768 kHz

REGISTER 4-2: CONFIGURATION WORD 2: SUPERVISORS

R/P-1	R/P-1	R/P-1	U-1	R/P-1	U-1
$\overline{\text{DEBUG}}$	STVREN	PPS1WAY	—	BORV	—
bit 13					bit 8

R/P-1	R/P-1	R/P-1	U-1	R/P-1	R/P-1	R/P-1	R/P-1
BOREN1	BOREN0	$\overline{\text{LPBOREN}}$	—	WDTE1	WDTE0	$\overline{\text{PWRTE}}$	MCLRE
bit 7							bit 0

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '1'
'0' = Bit is cleared	'1' = Bit is set	n = Value when blank or after Bulk Erase

bit 13 **$\overline{\text{DEBUG}}$:** Debugger Enable bit(1)
1 = OFF Background debugger disabled; ICSPCLK and ICSPDAT are general purpose I/O pins
0 = ON Background debugger enabled; ICSPCLK and ICSPDAT are dedicated to the debugger

bit 12 **STVREN:** Stack Overflow/Underflow Reset Enable bit
1 = ON Stack Overflow or Underflow will cause a Reset
0 = OFF Stack Overflow or Underflow will not cause a Reset

bit 11 **PPS1WAY:** PPSLOCK One-Way Set Enable bit
1 = ON The PPSLOCK bit can be cleared and set only once; PPS registers remain locked after one clear/set cycle
0 = OFF The PPSLOCK bit can be set and cleared repeatedly (subject to the unlock sequence)

bit 10 **Unimplemented:** Read as '1'

bit 9 **BORV:** Brown-out Reset Voltage Selection bit(2)
1 = LOW Brown-out Reset voltage (V_{BOR}) set to 1.9V on LF, and 2.45V on F devices
0 = HIGH Brown-out Reset voltage (V_{BOR}) set to 2.7V
The higher voltage setting is recommended for operation at or above 16 MHz.

bit 8 **Unimplemented:** Read as '1'

bit 7-6 **BOREN<1:0>:** Brown-out Reset Enable bits
When enabled, Brown-out Reset Voltage (V_{BOR}) is set by the BORV bit
11 = ON Brown-out Reset is enabled; SBOREN bit is ignored
10 = SLEEP Brown-out Reset is enabled while running, disabled in Sleep; SBOREN bit is ignored
01 = SBOREN Brown-out Reset is enabled according to SBOREN
00 = OFF Brown-out Reset is disabled

bit 5 **$\overline{\text{LPBOREN}}$:** Low-Power BOR Enable bit
1 = OFF ULPBOR is disabled
0 = ON ULPBOR is enabled

bit 4 **Unimplemented:** Read as '1'

bit 3-2 **WDTE<1:0>:** Watchdog Timer Enable bit
11 = ON WDT is enabled; SWDTEN is ignored
10 = SLEEP WDT is enabled while running and disabled in Sleep/Idle; SWDTEN is ignored
01 = SWDTEN WDT is controlled by the SWDTEN bit in the WDTCON register
00 = OFF WDT is disabled; SWDTEN is ignored

bit 1 **$\overline{\text{PWRTE}}$:** Power-up Timer Enable bit
1 = OFF PWRT is disabled
0 = ON PWRT is enabled

bit 0 **MCLRE:** Master Clear ($\overline{\text{MCLR}}$) Enable bit
If LVP = 1:
RA3 pin function is $\overline{\text{MCLR}}$.
If LVP = 0:
1 = ON $\overline{\text{MCLR}}$ pin is $\overline{\text{MCLR}}$.
0 = OFF $\overline{\text{MCLR}}$ pin function is port-defined function.

Note 1: The DEBUG bit in Configuration Words is managed automatically by device development tools including debuggers and programmers. For normal device operation, this bit should be maintained as a '1'.
2: See V_{BOR} parameter for specific trip point voltages.

REGISTER 4-3: CONFIGURATION WORD 3: MEMORY

R/P-1	U-1	U-1	U-1	U-1	U-1
LVP(1)	—	—	—	—	—
bit 13					bit 8

U-1	U-1	U-1	U-1	U-1	U-1	R/P-1	R/P-1
—	—	—	—	—	—	WRT1	WRT0
bit 7							bit 0

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '1'
'0' = Bit is cleared	'1' = Bit is set	n = Value when blank or after Bulk Erase

bit 13 **LVP:** Low-Voltage Programming Enable bit(1)
- 1 = ON Low-Voltage Programming is enabled. $\overline{MCLR}$/VPP pin function is $\overline{MCLR}$. MCLRE Configuration bit is ignored.
- 0 = OFF HV on $\overline{MCLR}$/VPP must be used for programming.

bit 12-2 **Unimplemented:** Read as '1'

bit 1-0 **WRT<1:0>:** User NVM Self-Write Protection bits
- 11 = OFF Write protection off
- 10 = BOOT 0000h to 01FFh write-protected, 0200h to 1FFFh may be modified
- 01 = HALF 0000h to 0FFFh write-protected, 1000h to 1FFFh may be modified
- 00 = ALL 0000h to 1FFFh write-protected, no addresses may be modified

WRT applies only to the self-write feature of the device; writing through ICSP™ is never protected.

Note 1: The LVP bit cannot be programmed to '0' when Programming mode is entered via LVP.

REGISTER 4-4: CONFIGURATION WORD 4: CODE PROTECTION

U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—
bit 13					bit 8

U-1	U-1	U-1	U-1	U-1	U-1	R/P-1	R/P-1
—	—	—	—	—	—	$\overline{CPD}$	$\overline{CP}$
bit 7							bit 0

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '1'
'0' = Bit is cleared	'1' = Bit is set	n = Value when blank or after Bulk Erase

bit 13-2 **Unimplemented:** Read as '1'

bit 1 **$\overline{CPD}$:** Data EEPROM Memory Code Protection bit
- 1 = OFF Data EEPROM code protection disabled
- 0 = ON Data EEPROM code protection enabled

bit 0 **$\overline{CP}$:** Program Memory Code Protection bit
- 1 = OFF Program Memory code protection disabled
- 0 = ON Program Memory code protection enabled

4.3 Code Protection

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

4.3.1 PROGRAM MEMORY PROTECTION

The entire program memory space is protected from external reads and writes by the $\overline{CP}$ bit in Configuration Words. 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. Self-write writing the program memory is dependent upon the write protection setting. See **Section 4.4 "Write Protection"** for more information.

4.3.2 DATA MEMORY PROTECTION

The entire data EEPROM is protected from external reads and writes by the $\overline{CPD}$ bit in the Configuration Words. When $\overline{CPD}$ = 0, external reads and writes of EEPROM memory are inhibited and a read will return all '0's. The CPU can continue to read and write EEPROM memory, regardless of the protection bit settings.

4.4 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 Words define the size of the program memory block that is protected.

4.5 User ID

Four memory locations (8000h-8003h) 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 10.4.7 "NVMREG EEPROM, User ID, Device ID and Configuration Word Access"** for more information on accessing these memory locations. For more information on checksum calculation, see the *"PIC16(L)F183XX Memory Programming Specification"* (DS40001738).

4.6 Device ID and Revision ID

The 14-bit device ID word is located at 8006h and the 14-bit revision ID is located at 8005h. These locations are read-only and cannot be erased or modified. See **Section 10.4 "NVMREG 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.

4.7 Register Definitions: Device and Revision

REGISTER 4-5: DEVID: DEVICE ID REGISTER

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

R	R	R	R	R	R	R	R
DEV<7:0>							
bit 7							bit 0

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

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

Device	DEVID<13:0> Values
PIC16F18325	11 0000 0011 1110 (303Eh)
PIC16LF18325	11 0000 0100 0000 (3040h)
PIC16F18345	11 0000 0011 1111 (303Fh)
PIC16LF18345	11 0000 0100 0001 (3041h)

REGISTER 4-6: REVID: REVISION ID REGISTER

R-1	R-0	R	R	R	R
REV<13:8>					
bit 13					bit 8

R	R	R	R	R	R	R	R
REV<7:0>							
bit 7							bit 0

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

bit 13-0 **REV<13:0>:** Revision ID bits

Note: The upper two bits of the Revision ID Register will always read '10'.

5.0 RESETS

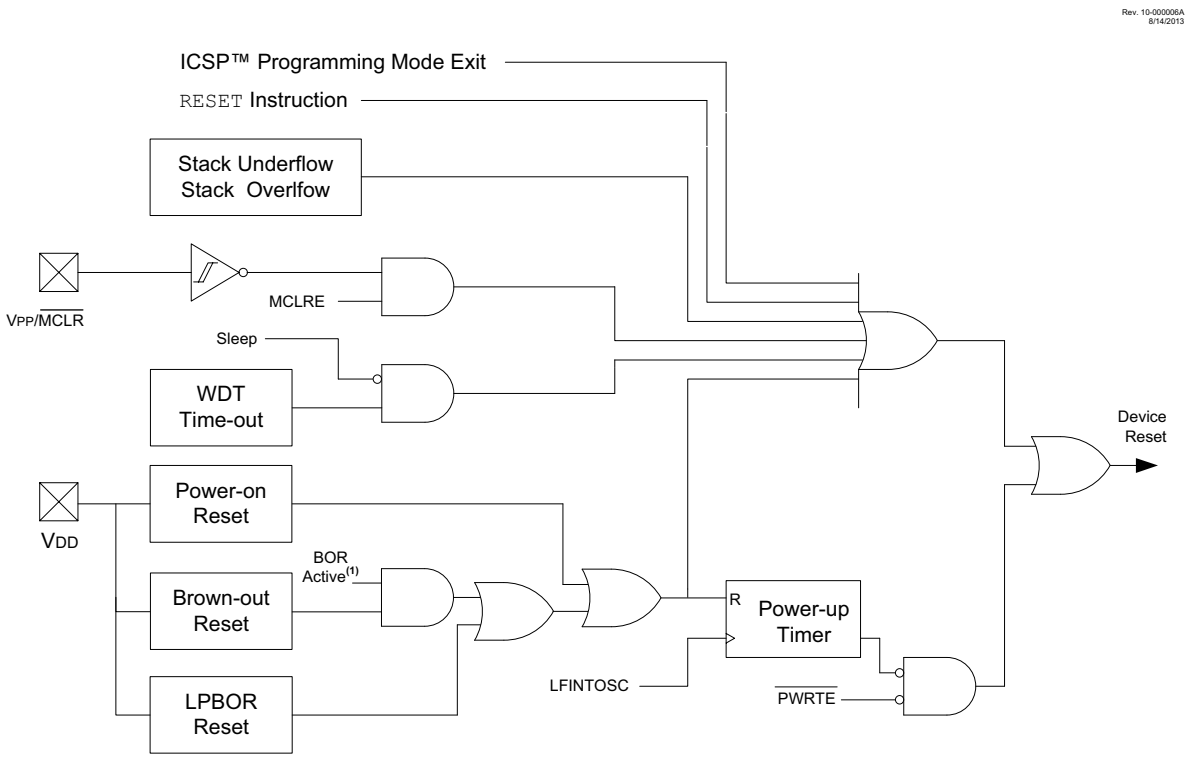
There are multiple ways to reset this device:

- Power-On Reset (POR)
- Brown-Out Reset (BOR)
- Low-Power Brown-Out Reset (LPBOR)
- $\overline{\text{MCLR}}$ Reset
- WDT Reset
- RESET instruction
- Stack Overflow
- Stack Underflow
- 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 MCLR features can be used to extend the start-up period until all device operation conditions have been met.

5.2 Brown-out Reset (BOR)

The BOR circuit holds the device in Reset while VDD is below 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 Words. 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 Configuration Words.

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	Instruction Execution upon: Release of POR or Wake-up from Sleep
11	X	X	Active	Waits for release of BOR(1) (BORRDY = 1)
10	X	Awake	Active	Waits for release of BOR (BORRDY = 1)
		Sleep	Disabled	BOR ignored when asleep
01	1	X	Active	Waits for release of BOR(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 Words 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 Words 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, but device wake-up will be delayed until the BOR can determine that VDD is higher than the BOR threshold. The device wake-up will be delayed until the BOR is ready.

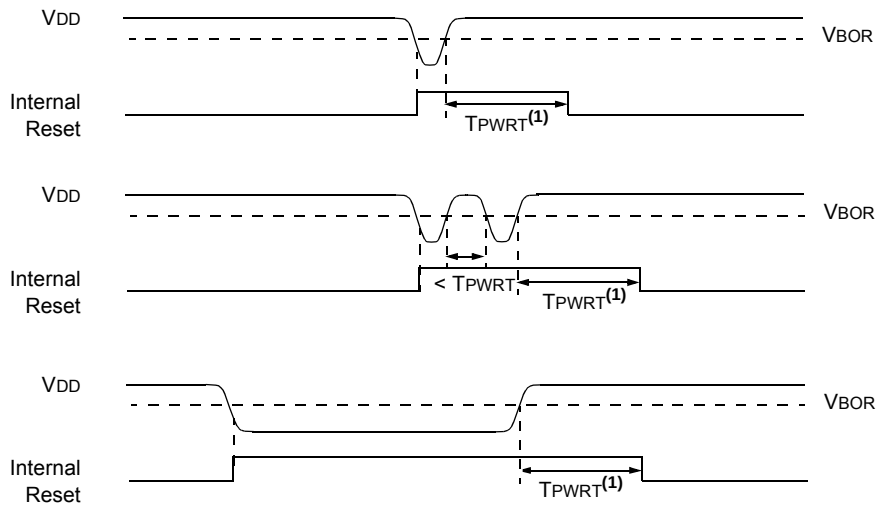
5.2.3 BOR CONTROLLED BY SOFTWARE

When the BOREN bits of Configuration Words are programmed to '01', the BOR is controlled by the SBOREN bit of the BORCON register. The device wake from Sleep 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'.

5.2.4 BOR ALWAYS OFF

When the BOREN bits of Configuration Word 2 are programmed to '00', the BOR is always disable. In the configuration, setting the SWBOREN bit will have no affect on BOR operation.

5.3 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 ($\overline{BOR}$) 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.

5.3.1 ENABLING LPBOR

The LPBOR is controlled by the $\overline{LPBOR}$ bit of Configuration Words. When the device is erased, the LPBOR module defaults to disabled.

5.3.1.1 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 $\overline{BOR}$ signal, which goes to the PCON register and to the power control block.

5.4 $\overline{MCLR}$

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

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

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

5.4.1 $\overline{MCLR}$ ENABLED

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

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

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

5.4.2 $\overline{MCLR}$ DISABLED

When $\overline{MCLR}$ is disabled, the pin functions as a general purpose input and the internal weak pull-up is under software control. See **Section 11.2 "PORTA Registers"** for more information.

5.5 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 $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register as well as the $\overline{RWDT}$ bit in the PCON register, are changed to indicate the WDT Reset. See **Section 9.0 "Watchdog Timer (WDT)"** for more information.

5.6 RESET Instruction

A RESET instruction will cause a device Reset. The $\overline{RI}$ bit in the PCON register will be set to '0'. See Table 5-4 for default conditions after a RESET instruction has occurred.

5.7 Stack Overflow/Underflow Reset

The device can reset when the Stack Overflows or Underflows. The STKOVF or STKUNF bits of the PCON register indicate the Reset condition. These Resets are enabled by setting the STVREN bit in Configuration Words. See **Section 3.4.2 "Overflow/Underflow Reset"** for more information.

5.8 Programming Mode Exit

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

5.9 Power-up Timer

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 $\overline{PWRTE}$ bit in Configuration Words.

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).

5.10 Start-up Sequence

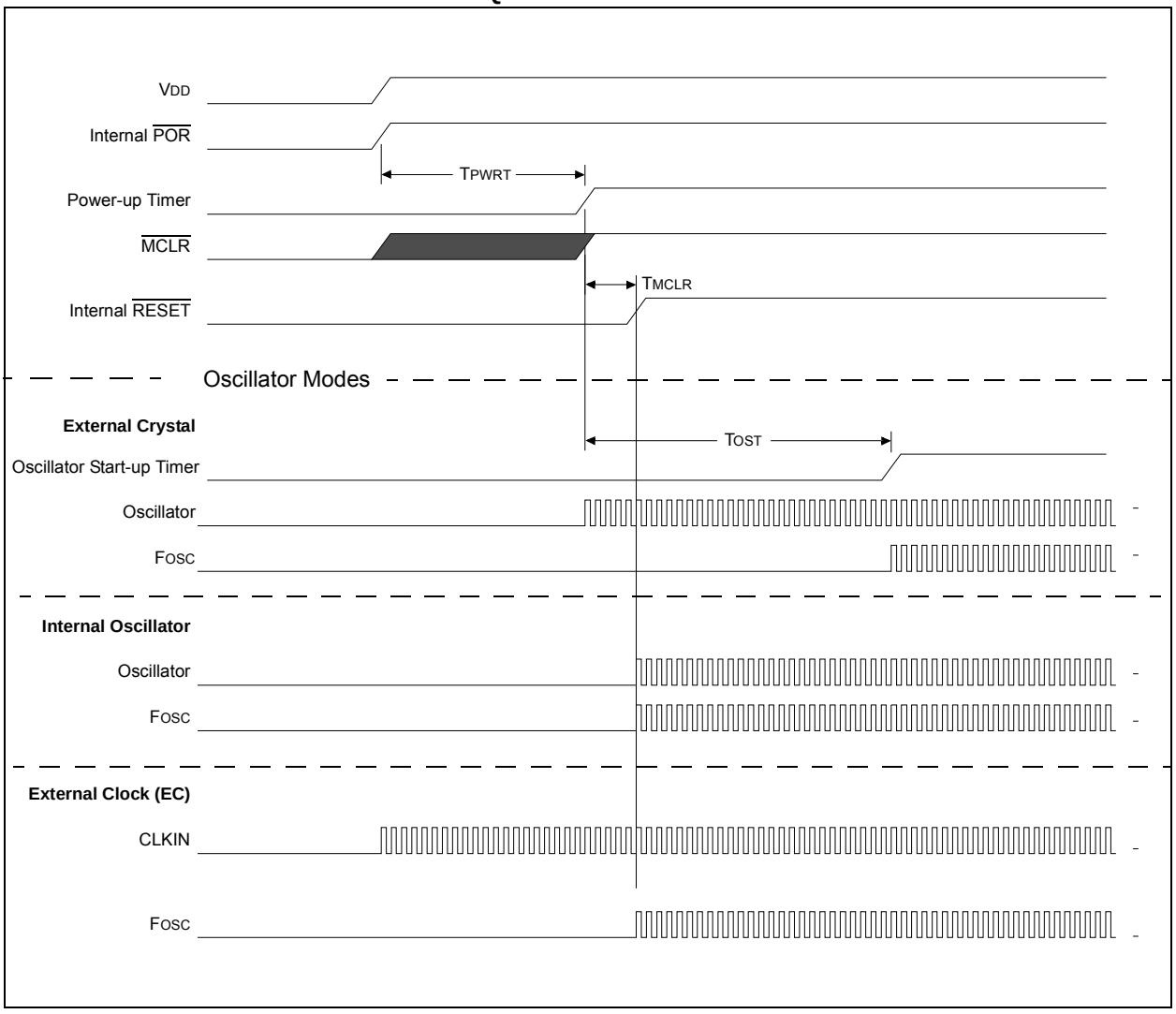
Upon the release of a POR or BOR, the following must occur before the device will begin executing:

1. Power-up Timer runs to completion (if enabled).
2. $\overline{\text{MCLR}}$ must be released (if enabled).
3. Oscillator start-up timer runs to completion (if required for oscillator source).

The total time-out will vary based on oscillator configuration and Power-up Timer Configuration. See **Section 6.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for more information.

The Power-up Timer and oscillator start-up timer run 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



5.11 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

STKOVF	STKUNF	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Condition
0	0	1	1	1	0	x	1	1	Power-on Reset
0	0	1	1	1	0	x	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	0	1	1	1	0	x	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
0	0	u	1	1	u	0	1	1	Brown-out Reset
u	u	0	u	u	u	u	0	u	WDT Reset
u	u	u	u	u	u	u	0	0	WDT Wake-up from Sleep
u	u	u	u	u	u	u	1	0	Interrupt Wake-up from Sleep
u	u	u	0	u	u	u	u	u	$\overline{\text{MCLR}}$ Reset during Normal Operation
u	u	u	0	u	u	u	1	0	$\overline{\text{MCLR}}$ Reset during Sleep
u	u	u	u	0	u	u	u	u	RESET Instruction Executed
1	u	u	u	u	u	u	u	u	Stack Overflow Reset (STVREN = 1)
u	1	u	u	u	u	u	u	u	Stack Underflow Reset (STVREN = 1)

TABLE 5-4: RESET CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON0 Register
Power-on Reset	0000h	---1 1000	00-- 110x
$\overline{\text{MCLR}}$ Reset during Normal Operation	0000h	---u uuuu	uu-- 0uuu
$\overline{\text{MCLR}}$ Reset during Sleep	0000h	---1 0uuu	uu-- 0uuu
WDT Reset	0000h	---0 uuuu	uu-0 uuuu
WDT Wake-up from Sleep	PC + 1	---0 0uuu	uu-u uuuu
Brown-out Reset	0000h	---1 1000	00-1 11u0
Interrupt Wake-up from Sleep	PC + 1(1)	---1 0uuu	uu-u uuuu
RESET Instruction Executed	0000h	---u uuuu	uu-u u0uu
Stack Overflow Reset (STVREN = 1)	0000h	---u uuuu	1u-u uuuu
Stack Underflow Reset (STVREN = 1)	0000h	---u uuuu	u1-u uuuu

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.

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

R/W-1/u	R/W-0-0	U-0	U-0	U-0	U-0	U-0	R-q/u
SBOREN(1)	Reserved	—	—	—	—	—	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(1)
If BOREN <1:0> in Configuration Words ≠ 01:
SBOREN is read/write, but has no effect on the BOR.
If BOREN <1:0> in Configuration Words = 01:
1 = BOR Enabled
0 = BOR Disabled

bit 6 Reserved

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 Words.

5.12 Power Control (PCON) Register

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

- Power-on Reset ($\overline{POR}$)
- Brown-out Reset ($\overline{BOR}$)
- Reset Instruction Reset ($\overline{RI}$)
- $\overline{MCLR}$ Reset ($\overline{RMCLR}$)
- Watchdog Timer Reset ($\overline{RWDT}$)
- Stack Underflow Reset (STKUNF)
- Stack Overflow Reset (STKOVF)

The PCON0 register bits are shown in Register 5-2.

Hardware will change the corresponding register bit during the Reset process; if the Reset was not caused by the condition, the bit remains unchanged (Table 5-4).

Software should reset the bit to the inactive state after the restart (hardware will not reset the bit).

Software may also set any PCON bit to the active state, so that user code may be tested, but no Reset action will be generated.

5.13 Register Definitions: Power Control

REGISTER 5-2: PCON0: POWER CONTROL REGISTER 0

R/W/HS-0/q	R/W/HS-0/q	U-0	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-q/u	R/W/HC-q/u
STKOVF	STKUNF	—	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
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	-m/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 **STKOVF:** Stack Overflow Flag bit
1 = A Stack Overflow occurred
0 = A Stack Overflow has not occurred or cleared by firmware

bit 6 **STKUNF:** Stack Underflow Flag bit
1 = A Stack Underflow occurred
0 = A Stack Underflow has not occurred or cleared by firmware

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

bit 4 **$\overline{\text{RWDT}}$:** Watchdog Timer Reset Flag bit
1 = A Watchdog Timer Reset has not occurred or set to '1' by firmware
0 = A Watchdog Timer Reset has occurred (cleared by hardware)

bit 3 **$\overline{\text{RMCLR}}$:** $\overline{\text{MCLR}}$ Reset Flag bit
1 = A $\overline{\text{MCLR}}$ Reset has not occurred or set to '1' by firmware
0 = A $\overline{\text{MCLR}}$ Reset has occurred (cleared by hardware)

bit 2 **$\overline{\text{RI}}$:** RESET Instruction Flag bit
1 = A RESET instruction has not been executed or set to '1' by firmware
0 = A RESET instruction has been executed (cleared by hardware)

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	—	—	—	—	—	—	BORRDY	72
PCON0	STKOVF	STKUNF	—	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	73
STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	27
WDTCON	—	—	WDTPS<4:0>					SWDTEN	118

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

6.0 OSCILLATOR MODULE (WITH FAIL-SAFE CLOCK MONITOR)

6.1 Overview

The oscillator module has a wide variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 6-1 illustrates a block diagram of the oscillator module.

Clock sources can be supplied from external oscillators, quartz-crystal resonators and ceramic resonators. In addition, the system clock source can be supplied from one of two internal oscillators and PLL circuits, with a choice of speeds selectable via software. Additional clock features include:

- Selectable system clock source between external or internal sources via software.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, ECH, ECM, ECL) and switch automatically to the internal oscillator.
- Oscillator Start-up Timer (OST) ensures stability of crystal oscillator sources.

The RSTOSC bits of Configuration Word 1 determine the type of oscillator that will be used when the device is reset, including when it is first powered-up.

The internal clock modes, LFINTOSC, HFINTOSC (set at 1 MHz), or HFINTOSC (set at 32 MHz) can be set through the RSTOSC bits.

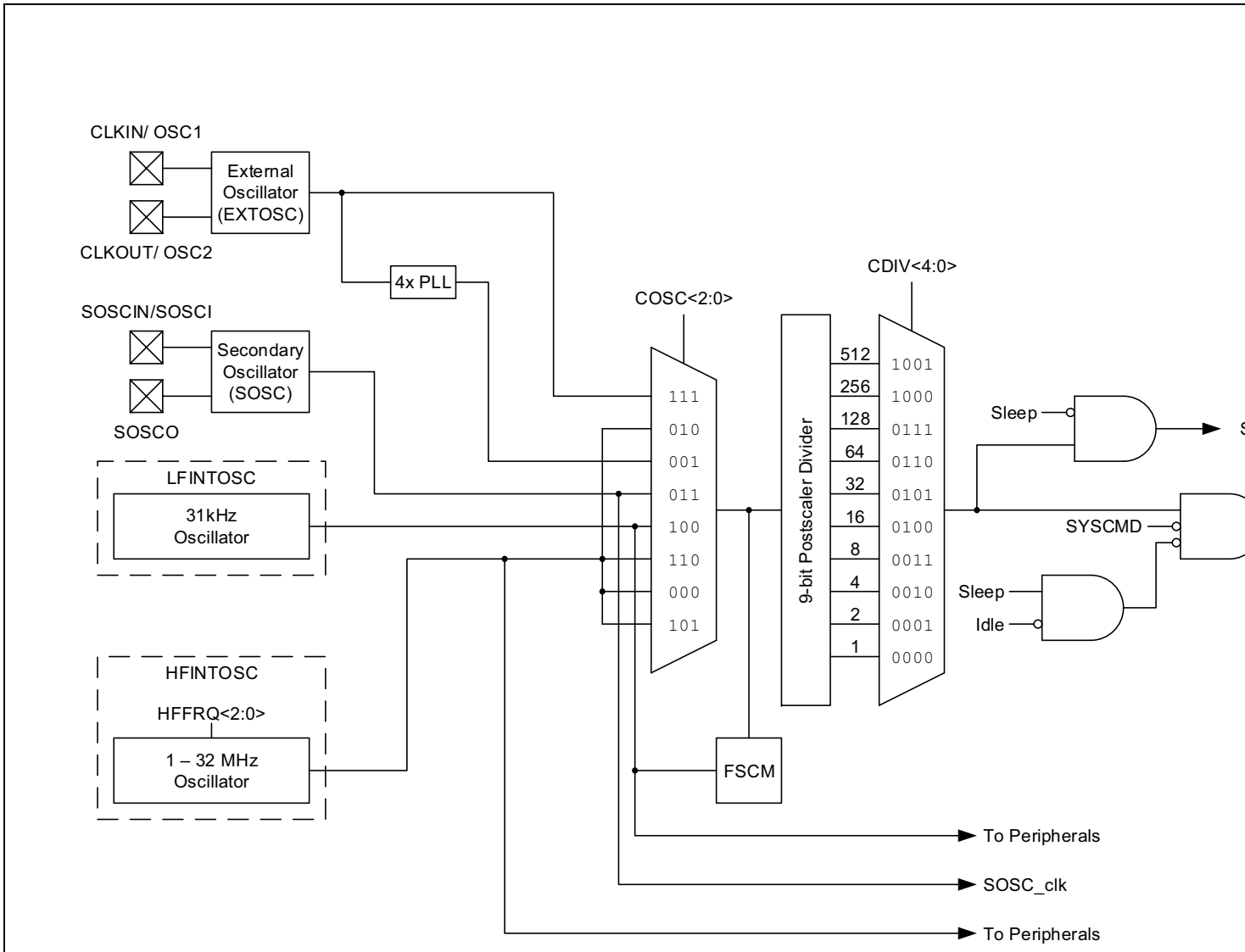
If an external clock source is selected, the FEXTOSC bits of Configuration Word 1 must be used in conjunction with the RSTOSC bits to select the External Clock mode.

The external oscillator module can be configured in one of the following clock modes by setting the FEXTOSC<2:0> bits of Configuration Word 1:

1. ECL – External Clock Low-Power mode (below 100 kHz)
2. ECM – External Clock Medium-Power mode (100 kHz to 8 MHz)
3. ECH – External Clock High-Power mode (above 8 MHz)
4. LP – 32 kHz Low-Power Crystal mode.
5. XT – Medium Gain Crystal or Ceramic Resonator Oscillator mode (between 100 kHz and 4 MHz)
6. HS – High Gain Crystal or Ceramic Resonator mode (above 4 MHz)

The ECH, ECM, and ECL Clock modes rely on an external logic level signal as the device clock source. The LP, XT, and HS Clock modes require an external crystal or resonator to be connected to the device. Each mode is optimized for a different frequency range. The INTOSC internal oscillator block produces low and high-frequency clock sources, designated LFINTOSC and HFINTOSC. (see Internal Oscillator Block, Figure 6-1).

FIGURE 6-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM



6.2 Clock Source Types

Clock sources can be classified as external or internal.

External clock sources rely on external circuitry for the clock source to function. Examples are: oscillator modules (ECH, ECM, ECL mode), quartz crystal resonators or ceramic resonators (LP, XT and HS modes).

There is also a secondary oscillator block which is optimized for a 32.768 kHz external clock source, which can be used as an alternate clock source.

There are two internal oscillator blocks:

- HFINTOSC
- LFINTOSC

The HFINTOSC can produce clock frequencies from 1-16 MHz. The LFINTOSC generates a 31 kHz clock frequency.

There is a PLL that can be used by the external oscillator. See **6.2.1.4 "4x PLL"** for more details. Additionally, there is a PLL that can be used by the HFINTOSC at certain frequencies. See **Section 6.2.2.2 "2x PLL"** for more details.

6.2.1 EXTERNAL CLOCK SOURCES

An external clock source can be used as the device system clock by performing one of the following actions:

- Program the RSTOSC<2:0> bits in the Configuration Words to select an external clock source that will be used as the default system clock upon a device Reset.
- Write the NOSC<2:0> and NDIV<3:0> bits in the OSCCON1 register to switch the system clock source.

See **Section 6.3 "Clock Switching"** for more information.

6.2.1.1 EC Mode

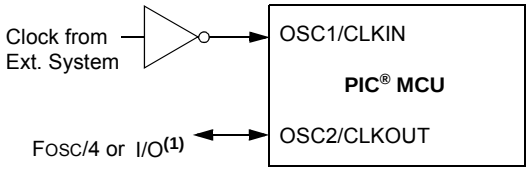
The External Clock (EC) mode allows an externally generated logic level signal to be the system clock source. When operating in this mode, an external clock source is connected to the CLKIN input. OSC2/CLKOUT is available for general purpose I/O or CLKOUT. Figure 6-2 shows the pin connections for EC mode.

EC mode has three power modes to select from through Configuration Words:

- ECH – High power, 8-32 MHz
- ECM – Medium power, 0.1-8 MHz
- ECL – Low power, 0-0.1 MHz

The Oscillator Start-up Timer (OST) is disabled when EC mode is selected. Therefore, there is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the PIC® MCU design is fully static, stopping the external clock input will have the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.

FIGURE 6-2: EXTERNAL CLOCK (EC) MODE OPERATION



Note 1: Output depends upon $\overline{\text{CLKOUTEN}}$ bit of the Configuration Words.

6.2.1.2 LP, XT, HS Modes

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to OSC1 and OSC2 (Figure 6-3). The three modes select a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

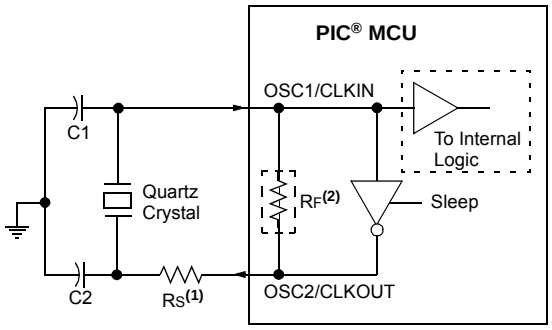
LP Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is designed to drive only 32.768 kHz tuning-fork type crystals (watch crystals).

XT Oscillator mode selects the intermediate gain setting of the internal inverter-amplifier. XT mode current consumption is the medium of the three modes. This mode is best suited to drive resonators with a medium drive level specification.

HS Oscillator mode selects the highest gain setting of the internal inverter-amplifier. HS mode current consumption is the highest of the three modes. This mode is best suited for resonators that require a high drive setting.

Figure 6-3 and Figure 6-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

FIGURE 6-3: QUARTZ CRYSTAL OPERATION (LP, XT OR HS MODE)



Note 1: A series resistor (RS) may be required for quartz crystals with low drive level.

2: The value of RF varies with the Oscillator mode selected (typically between 2 MΩ to 10 MΩ).

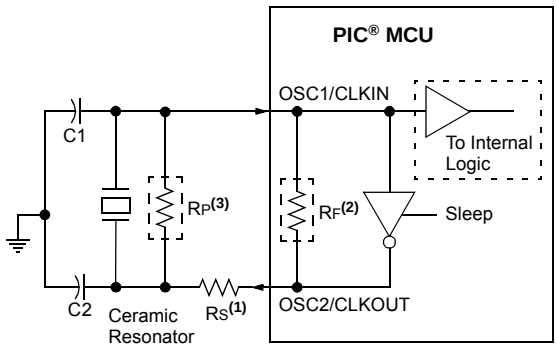
Note 1: Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

2: Always verify oscillator performance over the VDD and temperature range that is expected for the application.

3: For oscillator design assistance, reference the following Microchip Application Notes:

- AN826, "*Crystal Oscillator Basics and Crystal Selection for rfPIC® and PIC® Devices*" (DS00826)
- AN849, "*Basic PIC® Oscillator Design*" (DS00849)
- AN943, "*Practical PIC® Oscillator Analysis and Design*" (DS00943)
- AN949, "*Making Your Oscillator Work*" (DS00949)

FIGURE 6-4: CERAMIC RESONATOR OPERATION (XT OR HS MODE)



Note 1: A series resistor (RS) may be required for ceramic resonators with low drive level.

2: The value of RF varies with the Oscillator mode selected (typically between 2 MΩ to 10 MΩ).

3: An additional parallel feedback resistor (RP) may be required for proper ceramic resonator operation.

6.2.1.3 Oscillator Start-up Timer (OST)

If the oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations from OSC1. This occurs following a Power-on Reset (POR) or a wake-up from Sleep. The OST ensures that the oscillator circuit, using a quartz crystal resonator or ceramic resonator, has started and is providing a stable system clock to the oscillator module.

6.2.1.4 4x PLL

The oscillator module contains a PLL that can be used with external clock sources to provide a system clock source. The input frequency for the PLL must fall within specifications. See the PLL Clock Timing Specifications in Table 34-9.

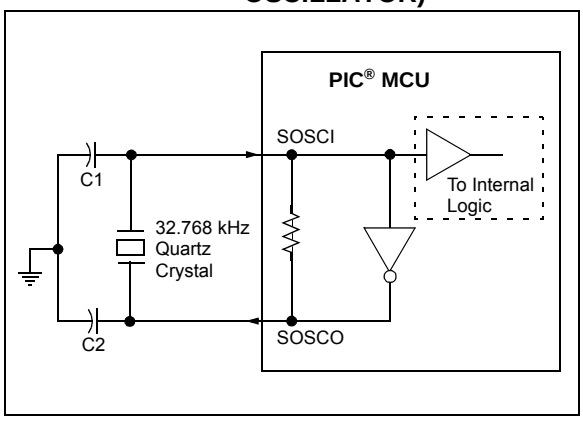
The PLL may be enabled for use by one of two methods:

1. Program the RSTOSC bits in the Configuration Word 1 to enable the EXTOSC with 4x PLL.
2. Write the NOSC<2:0> bits in the OSCCON1 register to enable the EXTOSC with 4x PLL.

6.2.1.5 Secondary Oscillator

The secondary oscillator is a separate oscillator block that can be used as an alternate system clock source. The secondary oscillator is optimized for 32.768 kHz, and can be used with an external crystal oscillator connected to the SOSCI and SOSCO device pins, or an external clock source connected to the SOSCIN pin. The secondary oscillator can be selected during run-time using clock switching. Refer to **Section 6.3 "Clock Switching"** for more information.

FIGURE 6-5: QUARTZ CRYSTAL OPERATION (SECONDARY OSCILLATOR)



Note 1: Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

2: Always verify oscillator performance over the VDD and temperature range that is expected for the application.

3: For oscillator design assistance, reference the following Microchip Application Notes:

- AN826, "*Crystal Oscillator Basics and Crystal Selection for rfPIC® and PIC® Devices*" (DS00826)
- AN849, "*Basic PICmicro® Oscillator Design*" (DS00849)
- AN943, "*Practical PICmicro® Oscillator Analysis and Design*" (DS00943)
- AN949, "*Making Your Oscillator Work*" (DS00949)
- TB097, "*Interfacing a Micro Crystal MS1V-T1K 32.768 kHz Tuning Fork Crystal to a PIC16F690/SS*" (DS91097)
- AN1288, "*Design Practices for Low-Power External Oscillators*" (DS01288)

6.2.2 INTERNAL CLOCK SOURCES

The device may be configured to use the internal oscillator block as the system clock by performing one of the following actions:

- Program the RSTOSC<2:0> bits in Configuration Words to select the INTOSC clock source, which will be used as the default system clock upon a device Reset.
- Write the NOSC<2:0> bits in the OSCCON1 register to switch the system clock source to the internal oscillator during run-time. See **Section 6.3 "Clock Switching"** for more information.

The function of the OSC2/CLKOUT pin is determined by the CLKOUTEN bit in Configuration Words.

The internal oscillator block has two independent oscillators that can produce two internal system clock sources.

1. The HFINTOSC (High-Frequency Internal Oscillator) is factory calibrated and operates up to 32 MHz. The frequency of HFINTOSC can be selected through the OSCFRQ Frequency Selection register, and fine-tuning can be done via the OSCTUNE register.
2. The LFINTOSC (Low-Frequency Internal Oscillator) is factory calibrated and operates at 31 kHz.

6.2.2.1 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a precision digitally-controlled internal clock source that produces a stable clock up to 32 MHz. The HFINTOSC can be enabled through one of the following methods:

- Programming the RSTOSC<2:0> bits in Configuration Word 1 to '110' (1 MHz) or '000' (32 MHz) to set the oscillator upon device Power-up or Reset
- Write to the NOSC<2:0> bits of the OSCCON1 register during run-time

The HFINTOSC frequency can be selected by setting the HFFRQ<2:0> bits of the OSCFRQ register.

The NDIV<3:0> bits of the OSCCON1 register allow for division of the output of the selected clock source by a range between 1:1 and 1:512.

6.2.2.2 2x PLL

The oscillator module contains a PLL that can be used with the HFINTOSC clock source to provide a system clock source. The input frequency to the PLL is limited to 8, 12, or 16 MHz, which will yield a system clock source of 16, 24, or 32 MHz, respectively.

The PLL may be enabled for use by one of two methods:

1. Program the RSTOSC bits in the Configuration Word 1 to '000' to enable the HFINTOSC (32 MHz). This setting configures the HFFRQ<2:0> bits to '110' (16 MHz) and activates the 2x PLL.
2. Write '000' the NOSC<2:0> bits in the OSCCON1 register to enable the 2x PLL, and write the correct value into the HFFRQ<2:0> bits of the OSCFRQ register to select the desired system clock frequency. See Register 6-6 for more information.

6.2.2.3 Internal Oscillator Frequency Adjustment

The internal oscillator is factory-calibrated. This internal oscillator can be adjusted in software by writing to the OSCTUNE register (Register 6-3).

The default value of the OSCTUNE register is 00h. The value is a 6-bit two's complement number. A value of 3Fh will provide an adjustment to the maximum frequency. A value of 0h will provide an adjustment to the minimum frequency.

When the OSCTUNE register is modified, the oscillator frequency will begin shifting to the new frequency. Code execution continues during this shift. There is no indication that the shift has occurred.

OSCTUNE does not affect the LFINTOSC frequency. Operation of features that depend on the LFINTOSC clock source frequency, such as the Power-up Timer (PWRT), Watchdog Timer (WDT), Fail-Safe Clock Monitor (FSCM) and peripherals, are *not* affected by the change in frequency.

6.2.2.4 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is a factory calibrated 31 kHz internal clock source.

The LFINTOSC is the clock source for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

The LFINTOSC is selected as the clock source through one of the following methods:

- Programming the RSTOSC<2:0> bits of Configuration Word 1 to enable LFINTOSC.
- Write to the NOSC<2:0> bits of the OSCCON1 register.

6.2.2.5 Oscillator Status and Manual Enable

The 'ready' status of each oscillator is displayed in the OSCSTAT1 register (Register 6-4). The oscillators can also be manually enabled through the OSCEN register (Register 6-5). Manual enables make it possible to verify the operation of the EXTOSC or SOSC crystal oscillators. This can be achieved by enabling the selected oscillator, then watching the corresponding 'ready' state of the oscillator in the OSCSTAT1 register.

6.3 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the New Oscillator Source (NOSC) and New Divider Selection Request (NDIV) bits of the OSCCON1 register. The following clock sources can be selected:

- External Oscillator (EXTOSC)
- High-Frequency Internal Oscillator (HFINTOSC)
- Low-Frequency Internal Oscillator (LFINTOSC)
- Secondary Oscillator (SOSC)
- EXTOSC with 4x PLL
- HFINTOSC with 2x PLL

6.3.1 NEW OSCILLATOR SOURCE (NOSC) AND NEW DIVIDER SELECTION REQUEST (NDIV) BITS

The New Oscillator Source (NOSC) and New Divider Selection Request (NDIV) bits of the OSCCON1 register select the system clock source that is used for the CPU and peripherals.

When the new values of NOSC<2:0> and NDIV<3:0> are written to OSCCON1, the current oscillator selection will continue to operate as the system clock while waiting for the new source to indicate that it is stable and ready. In some cases, the newly requested source may already be in use, and is ready immediately. In the case of a divider-only change, the new and old sources are the same, so the source will be ready immediately, as well. The device may enter Sleep while waiting for the switch as described in **Section 6.3.3 “Clock Switch and Sleep”**.

When the new oscillator is ready, the New Oscillator is Ready (NOSCR) bit of OSCCON3 and the Clock Switch Interrupt Flag (CSWIF) bit of PIR3 become set (CSWIF = 1). If Clock Switch Interrupts are enabled (CLKSIE = 1), an interrupt will be generated at that time. The Oscillator Ready (ORDY) bit of OSCCON3 can also be polled to determine when the oscillator is ready in lieu of an interrupt.

If the Clock Switch Hold (CSWHOLD) bit of OSCCON3 is clear, the oscillator switch will occur when the New Oscillator Ready bit (NOSCR) is set and the interrupt (if enabled) will be serviced at the new oscillator setting.

If CSWHOLD is set, the oscillator switch is suspended, while execution continues using the current (old) clock source. When the NOSCR bit is set, software should:

- Set CSWHOLD = 0 so the switch can complete, or
- Copy COSC into NOSC<2:0> to abandon the switch.

If DOZE is in effect, the switch occurs on the next clock cycle, whether or not the CPU is operating during that cycle.

Changing the clock post-divider without changing the clock source (i.e., changing FOSC from 1 MHz to 2 MHz) is handled in the same manner as a clock source change, as described previously. The clock source will already be active, so the switch is relatively quick. CSWHOLD must be clear (CSWHOLD = 0) for the switch to complete.

The current COSC and CDIV are indicated in the OSCCON2 register up to the moment when the switch actually occurs, at which time OSCCON2 is updated and ORDY is set. NOSCR is cleared by hardware to indicate that the switch is complete.

6.3.2 PLL INPUT SWITCH

Switching between the PLL and any non-PLL source is managed as described above. The input to the PLL is established when NOSC<2:0> selects the PLL, and maintained by the COSC setting.

When NOSC<2:0> and COSC select the PLL with different input sources, the system continues to run using the COSC setting, and the new source is enabled per NOSC<2:0>. When the new oscillator is ready (and CSWHOLD = 0), system operation is suspended while the PLL input is switched and the PLL acquires lock.

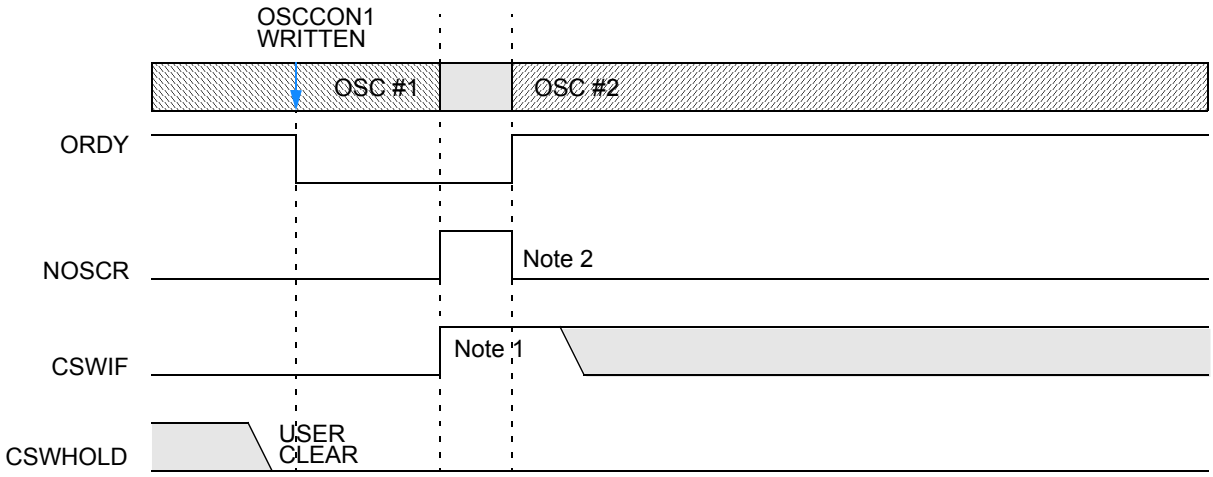
6.3.3 CLOCK SWITCH AND SLEEP

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

When the device wakes from Sleep and the CSWHOLD bit is clear, the device will wake with the ‘new’ clock active, and the Clock Switch Interrupt Flag bit (CSWIF) will be set.

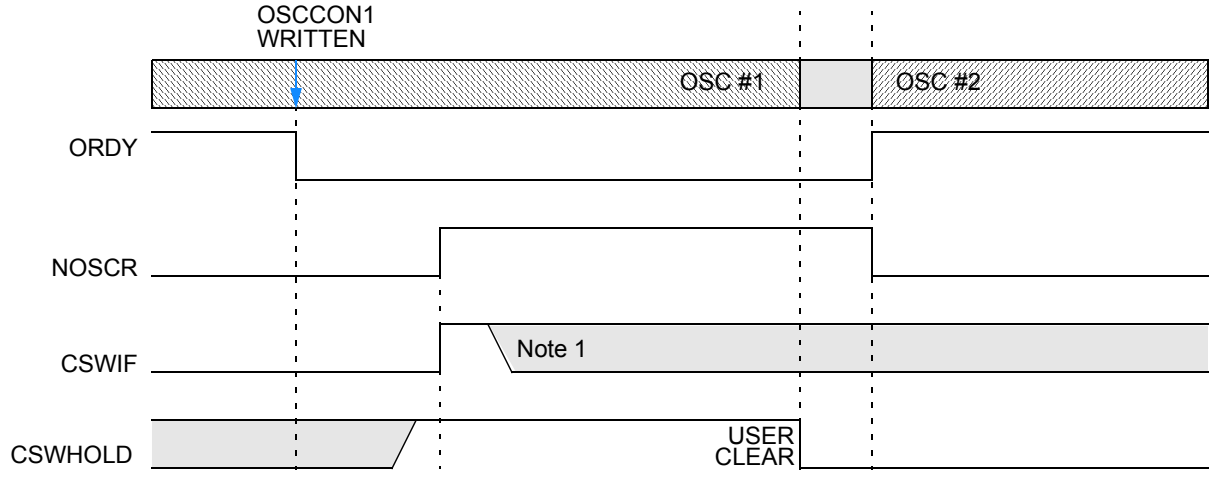
When the device wakes from Sleep and the CSWHOLD bit is set, the device will wake with the ‘old’ clock active and the new clock will be requested again.

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



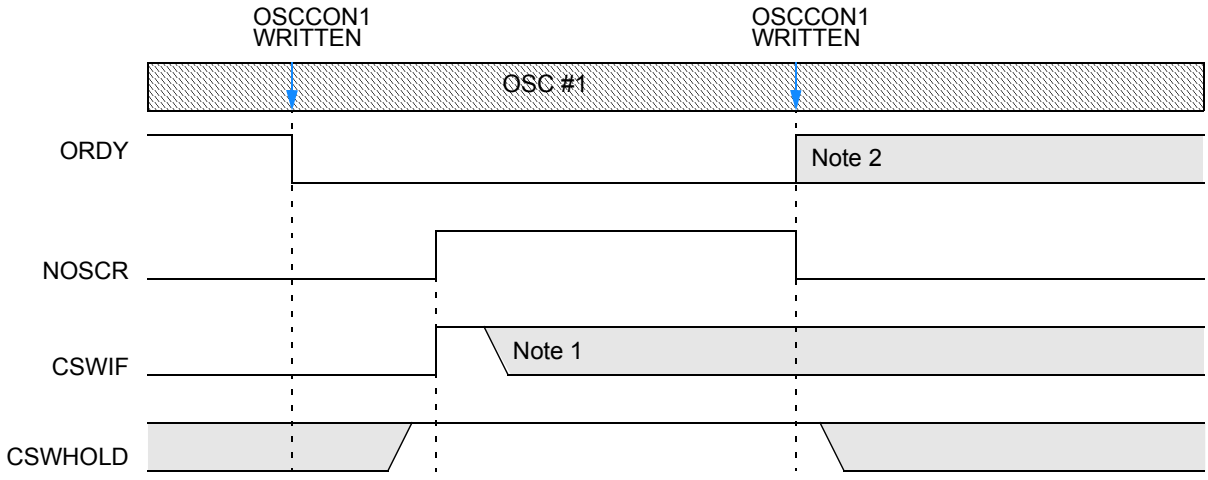
Note 1: CSWIF is asserted coincident with NOSCR; interrupt is serviced at OSC#2 speed.
2: The assertion of NOSCR is hidden from the user because it appears only for the duration of the switch.

FIGURE 6-7: CLOCK SWITCH (CSWHOLD = 1)



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

FIGURE 6-8: CLOCK SWITCH ABANDONED

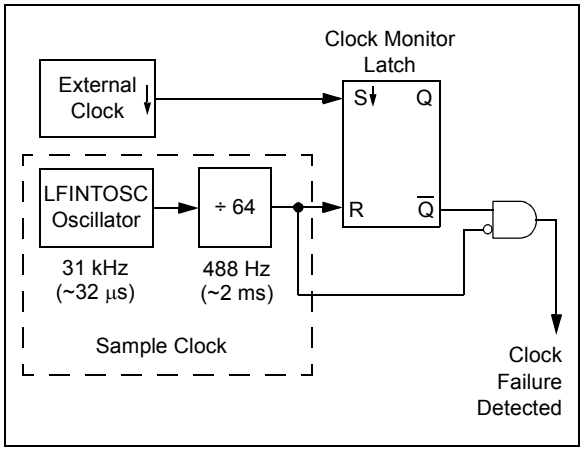


Note 1: CSWIF may be cleared before or after rewriting OSCCON1; CSWIF is not automatically cleared.
2: ORDY = 0 if OSCCON1 does not match OSCCON2; a new switch will begin.

6.4 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM is enabled by setting the FCMEN bit in the Configuration Words. The FSCM is applicable to all external Oscillator modes (LP, XT, HS, EC and Secondary Oscillator).

FIGURE 6-9: FSCM BLOCK DIAGRAM



6.4.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by comparing the external oscillator to the FSCM sample clock. The sample clock is generated by dividing the LFINTOSC by 64. See Figure 6-9. Inside the fail detector block is a latch. The external clock sets the latch on each falling edge of the external clock. The sample clock clears the latch on each rising edge of the sample clock. A failure is detected when an entire half-cycle of the sample clock elapses before the external clock goes low.

6.4.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM switches the device clock to the HFINTOSC at 1 MHz clock frequency and sets the bit flag OSFIF of the PIR3 register. Setting this flag will generate an interrupt if the OSFIE bit of the PIE3 register is also set. The device firmware can then take steps to mitigate the problems that may arise from a failed clock. The system clock will continue to be sourced from the internal clock source until the device firmware successfully restarts the external oscillator and switches back to external operation, by writing to the NOSC<2:0> and NDIV<3:0>bits of the OSCCON1 register.

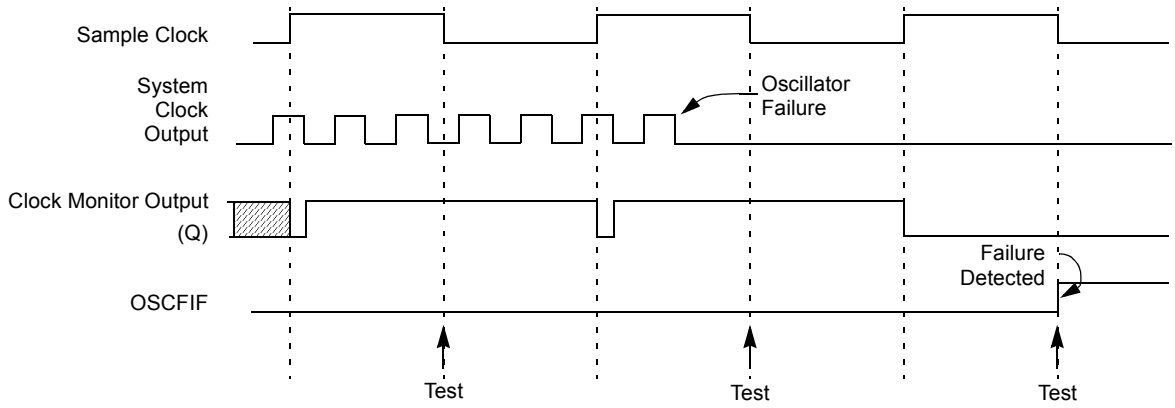
6.4.3 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared after a Reset, executing a SLEEP instruction or changing the NOSC<2:0> and NDIV<3:0> bits of the OSCCON1 register. When switching to the external oscillator or PLL, the OST is restarted. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON1. When the OST times out, the Fail-Safe condition is cleared after successfully switching to the external clock source. The OSFIF bit should be cleared prior to switching to the external clock source. If the Fail-Safe condition still exists, the OSFIF flag will again become set by hardware.

6.4.4 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect an oscillator failure after the Oscillator Start-up Timer (OST) has expired. The OST is used after waking up from Sleep and after any type of Reset. The OST is not used with the EC Clock modes so that the external clock signal can be stopped if required. Therefore, the device will always be executing code while the OST is operating.

FIGURE 6-10: FSCM TIMING DIAGRAM



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

6.5 Register Definitions: Oscillator Control

REGISTER 6-1: OSCCON1: OSCILLATOR CONTROL REGISTER 1

U-0	R/W-f/f(1)	R/W-f/f(1)	R/W-f/f(1)	R/W-q/q(4)	R/W-q/q(4)	R/W-q/q(4)	R/W-q/q(4)
—	NOSC<2:0>(2,3)			NDIV<3:0>(2,3)			
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	f = determined by fuse setting

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **NOSC<2:0>:** New Oscillator Source Request bits
The setting requests a source oscillator and PLL combination per Table 6-1.
POR value = RSTOSC (Register 4.2).

bit 3-0 **NDIV<3:0>:** New Divider Selection Request bits
The setting determines the new postscaler division ratio per Table 6-2.

Note 1: The default value (f/f) is set equal to the RSTOSC Configuration bits.
2: If NOSC is written with a reserved value (Table 6-1), the HFINTOSC will be automatically selected as the clock source.
3: When CSWEN = 0, this register is read-only and cannot be changed from the POR value.
4: When RSTOSC = 110 (HFINTOSC 1 MHz) the NDIV bits will default to '0010' upon Reset; for all other NOSC settings the NVID bits will default to '0000' upon Reset.

REGISTER 6-2: OSCCON2: OSCILLATOR CONTROL REGISTER 2

U-0	R-q/q(1)	R-q/q(1)	R-q/q(1)	R-q/q(1)	R-q/q(1)	R-q/q(1)	R-q/q(1)
—	COSC<2:0>			CDIV<3:0>			
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **COSC<2:0>:** Current Oscillator Source Select bits (read-only)
Indicates the current source oscillator and PLL combination per Table 6-1.

bit 3-0 **CDIV<3:0>:** Current Divider Select bits (read-only)
Indicates the current postscaler division ratio per Table 6-2.

Note 1: The Reset value (n/n) will match the NOSC<2:0>/NDIV<3:0> bits.

TABLE 6-1: NOSC/COSC BIT SETTINGS

NOSC<2:0> COSC<2:0>	Clock Source
111	EXTOSC(1)
110	HFINTOSC (1 MHz)
101	Reserved
100	LFINTOSC
011	SOSC
010	Reserved
001	EXTOSC with 4xPLL(1)
000	HFINTOSC with 2x PLL (32 MHz)

Note 1: EXTOSC configured by the FEXTOSC bits of Configuration Word 1 (Register 4-1).

TABLE 6-2: NDIV/CDIV BIT SETTINGS

NDIV<3:0> CDIV<3:0>	Clock Divider
1111–1010	Reserved
1001	512
1000	256
0111	128
0110	64
0101	32
0100	16
0011	8
0010	4
0001	2
0000	1

REGISTER 6-3: OSCCON3: OSCILLATOR CONTROL REGISTER 3

R/W/HC-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	U-0	U-0	U-0
CSWHOLD	SOSCPWR	SOSCBE	ORDY	NOSCR	—	—	—
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 **CSWHOLD:** Clock Switch Hold bit
- 1 = Clock switch will hold (with interrupt) when the oscillator selected by NOSC is ready
- 0 = Clock switch may proceed when the oscillator selected by NOSC is ready; if this bit is set at the time that NOSCR becomes ‘1’, the switch and interrupt will occur.

bit 6 **SOSCPWR:** Secondary Oscillator Power Mode Select bit

If SOSCBE = 0
- 1 = Secondary oscillator operating in High-Power mode
- 0 = Secondary oscillator operating in Low-Power mode

If SOSCBE = 0
- x = Bit is ignored

bit 5 **SOSCBE:** Secondary Oscillator Bypass Enable bit
- 1 = Secondary oscillator SOSCI is configured as an external clock input (ST-buffer); SOSCO is not used.
- 0 = Secondary oscillator is configured as a crystal oscillator using SOSCO and SOSCI pins.

bit 4 **ORDY:** Oscillator Ready bit (read-only)
- 1 = OSCCON1 = OSCCON2; the current system clock is the clock specified by NOSC
- 0 = A clock switch is in progress

bit 3 **NOSCR:** New Oscillator is Ready bit (read-only)
- 1 = A clock switch is in progress and the oscillator selected by NOSC indicates a Ready condition
- 0 = A clock switch is not in progress, or the NOSC-selected oscillator is not yet ready

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

REGISTER 6-4: OSCSTAT1: OSCILLATOR STATUS REGISTER 1

R-q/q	R-q/q	U-0	R-q/q	R-q/q	R-q/q	U-0	R-q/q
EXTOR	HFOR	—	LFOR	SOR	ADOR	—	PLLR
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 **EXTOR:** EXTOSC (external) Oscillator Ready
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used.

bit 6 **HFOR:** HFINTOSC Oscillator Ready
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used.

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **LFOR:** LFINTOSC Oscillator Ready
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used.

bit 3 **SOR:** Secondary Oscillator Ready
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used.

bit 2 **ADOR:** ADCRC Oscillator Ready
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **PLLR:** PLL is ready
1 = The PLL is ready to be used
0 = The PLL is not enabled, the required input source is not ready, or the PLL is not ready.

REGISTER 6-5: OSCEN: OSCILLATOR MANUAL ENABLE REGISTER

R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0
EXTOEN	HFOEN	—	LFOEN	SOSCEN	ADOEN	—	—
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 **EXTOEN:** External Oscillator Manual Request Enable bit
1 = EXTOSC is explicitly enabled, operating as specified by FEXTOSC
0 = EXTOSC could be enabled by another module

bit 6 **HFOEN:** HFINTOSC Oscillator Manual Request Enable bit
1 = HFINTOSC is explicitly enabled, operating as specified by OSCFRQ (Register 6-6)
0 = HFINTOSC could be enabled by another module

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **LFOEN:** LFINTOSC (31 kHz) Oscillator Manual Request Enable bit
1 = LFINTOSC is explicitly enabled
0 = LFINTOSC could be enabled by another module

bit 3 **SOSCEN:** Secondary (Timer1) Oscillator Manual Request Enable bit
1 = Secondary Oscillator is explicitly enabled, operating as specified by SOSCBE and SOSCPWR
0 = Secondary Oscillator could be enabled by another module

bit 2 **ADOEN:** ADOSC (600 kHz) Oscillator Manual Request Enable bit
1 = ADOSC is explicitly enabled
0 = ADOSC could be enabled by another module

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **Unimplemented:** Read as ‘0’

REGISTER 6-6: OSCFRQ: HFINTOSC FREQUENCY SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-q/q	R/W-q/q	R/W-q/q
—	—	—	—	—	HFFRQ<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 7-3 **Unimplemented:** Read as '0'.

bit 2-0 **HFFRQ<2:0>:** HFINTOSC Frequency Selection bits(1)

HFFRQ<2:0>	Nominal Freq. (MHz) (NOSC = 110)	2xPLL Freq. (MHz) (NOSC = 000)
000	1	N/A
001	2	
010	Reserved	
011	4	
100	8	16
101	12	24
110	16	32
111	Reserved	Reserved

Note 1: When RSTOSC = 110 (HFINTOSC 1 MHz), the HFFRQ bits will default to '011' upon Reset; when RSTOSC = 000 (HFINTOSC 32 MHz), the HFFRQ bits will default to '110' upon Reset.

REGISTER 6-7: OSCTUNE: HFINTOSC TUNING REGISTER

U-0	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
—	—	HFTUN<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	

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

bit 5-0 **HFTUN<5:0>:** HFINTOSC Frequency Tuning bits

01 1111 = Maximum frequency
01 1110
•
•
•
00 0001
00 0000 = Center frequency. Oscillator module is running at the calibrated frequency (default value).
11 1111
•
•
•
10 0000 = Minimum frequency.

TABLE 6-3: 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
OSCCON1	—	NOSC<2:0>			NDIV<3:0>				86
OSCCON2	—	COSC<2:0>			CDIV<3:0>				86
OSCCON3	CWSHOLD	SOSCPWR	SOSCBE	ORDY	NOSCR	—	—	—	87
OSCSTAT1	EXTOR	HFOR	—	LFOR	SOR	ADOR	—	PLLR	88
OSCEN	EXTOEN	HFOEN	—	LFOEN	SOSCEN	ADOEN	—	—	89
OSCFRQ	—	—	—	—	—	HFFRQ<2:0>			90
OSCTUNE	—	—	HFTUN<5:0>						91

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

TABLE 6-4: 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
CONFIG1	13:8	—	—	FCMEN	—	CSWEN	—	—	$\overline{\text{CLKOUTEN}}$	60
	7:0	—	RSTOSC2	RSTOSC1	RSTOSC0	—	FEXTOSC2	FEXTOSC1	FEXTOSC0	

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

7.0 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.

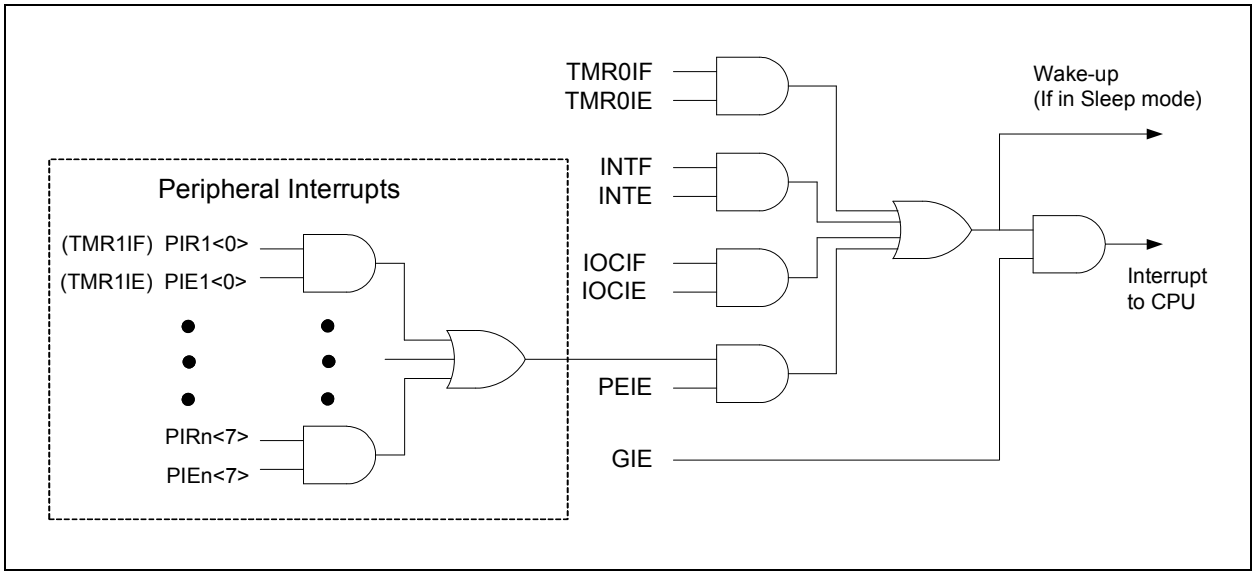
This chapter contains the following information for Interrupts:

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

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

A block diagram of the interrupt logic is shown in Figure 7-1.

FIGURE 7-1: INTERRUPT LOGIC



7.1 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 PIEx registers)

The PIR1, PIR2, PIR3 and PIR4 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
- Critical registers are automatically saved to the shadow registers (See **"Section 7.5 "Automatic Context Saving"**)
- 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, restoring the saved context from the shadow registers 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.

7.2 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 7-2 and Figure 7-3 for more details.

FIGURE 7-2: INTERRUPT LATENCY

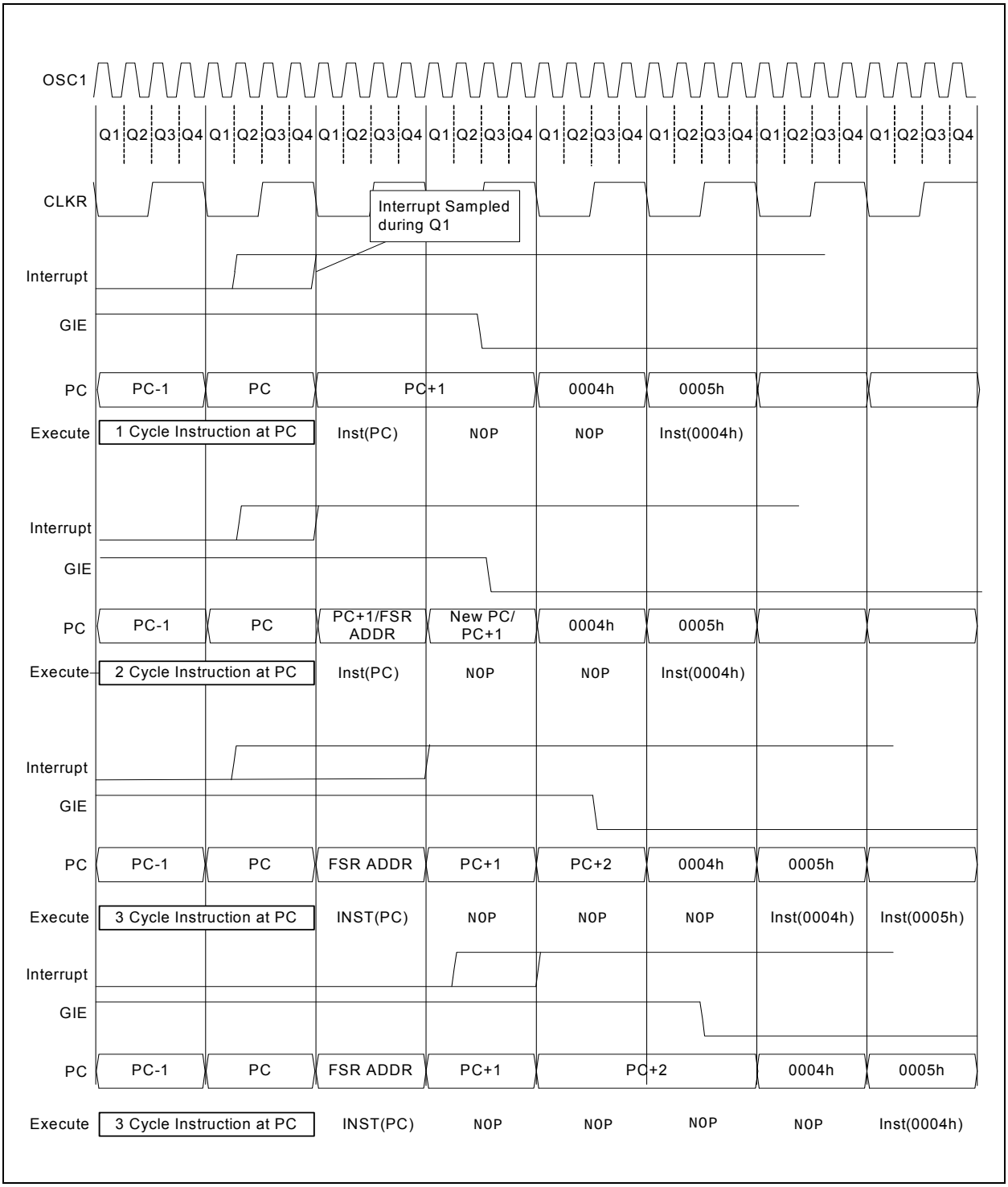
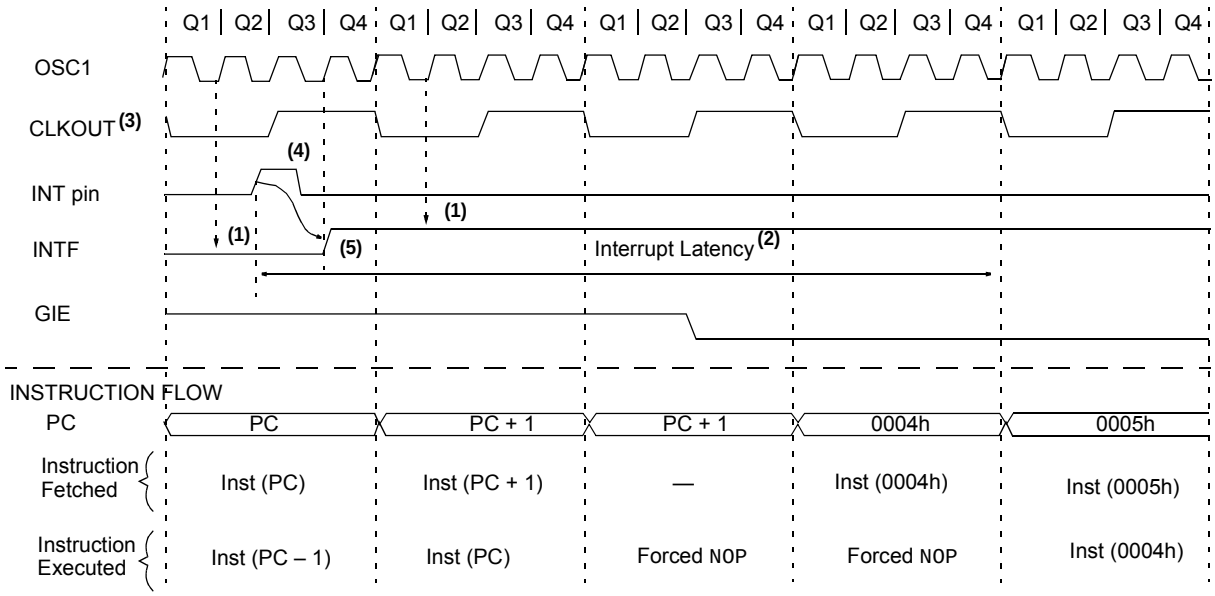


FIGURE 7-3: INT PIN INTERRUPT TIMING



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: CLKOUT not available in all oscillator modes.

4: For minimum width of INT pulse, refer to AC specifications in **Section 34.0 "Electrical Specifications"**".

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

7.3 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 **Section 8.0 "Power-Saving Operation Modes"** for more details.

7.4 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 PIE0 register. The INTEDG bit of the INTCON 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 PIR0 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.

7.5 Automatic Context Saving

Upon entering an interrupt, the return PC address is saved on the stack. Additionally, the following registers are automatically saved in the shadow registers:

- W register
- STATUS register (except for $\overline{TO}$ and $\overline{PD}$)
- BSR register
- FSR registers
- PCLATH register

Upon exiting the Interrupt Service Routine, these registers are automatically restored. Any modifications to these registers during the ISR will be lost. If modifications to any of these registers are desired, the corresponding shadow register should be modified and the value will be restored when exiting the ISR. The shadow registers are available in Bank 31 and are readable and writable. Depending on the user's application, other registers may also need to be saved.

7.6 Register Definitions: Interrupt Control

REGISTER 7-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	U-0	R-1/1
GIE	PEIE	—	—	—	—	—	INTEDG
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 5-1 **Unimplemented:** Read as ‘0’.

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

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

REGISTER 7-2: PIE0: PERIPHERAL INTERRUPT ENABLE REGISTER 0

U-0	U-0	R/W/HS-0/0	R-0	U-0	U-0	U-0	R/W/HS-0/0
—	—	TMR0IE	IOCIE	—	—	—	INTE
bit 7							bit 0

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

bit 7-6 **Unimplemented**: Read as '0'.

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
- 1 = Enables the TMR0 interrupt
- 0 = Disables the TMR0 interrupt

bit 4 **IOCIE:** Interrupt-on-Change Interrupt Enable bit
- 1 = Enables the IOC change interrupt
- 0 = Disables the IOC change interrupt

bit 3-1 **Unimplemented**: Read as '0'.

bit 0 **INTE:** INT External Interrupt Flag bit(1)
- 1 = Enables the INT external interrupt
- 0 = Disables the INT external interrupt

Note 1: The external interrupt GPIO pin is selected by INTPPS (Register 12-1).

REGISTER 7-3: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE
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 **TMR1GIE:** Timer1 Gate Interrupt Enable bit
1 = Enables the Timer1 gate acquisition interrupt
0 = Disables the Timer1 gate acquisition interrupt

bit 6 **ADIE:** Analog-to-Digital Converter (ADC) Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

bit 5 **RCIE:** EUSART Receive Interrupt Enable bit
1 = Enables the EUSART receive interrupt
0 = Disables the EUSART receive interrupt

bit 4 **TXIE:** EUSART Transmit Interrupt Enable bit
1 = Enables the EUSART transmit interrupt
0 = Disables the EUSART transmit interrupt

bit 3 **SSP1IE:** Synchronous Serial Port (MSSP) Interrupt Enable bit
1 = Enables the MSSP interrupt
0 = Disables the MSSP interrupt

bit 2 **BCL1IE:** MSSP1 Bus Collision Interrupt Enable bit
1 = MSSP bus collision interrupt enabled
0 = MSSP bus collision interrupt not enabled

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the Timer2 to PR2 match interrupt
0 = Disables the Timer2 to PR2 match interrupt

bit 0 **TMR1IE:** Timer1 Overflow Interrupt Enable bit
1 = Enables the Timer1 overflow interrupt
0 = Disables the Timer1 overflow interrupt

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

REGISTER 7-4: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2

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
TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BCL2IE	TMR4IE	NCO1IE
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 **TMR6IE:** TMR6 to PR6 Match Interrupt Enable bit
1 = TMR6 to PR6 match interrupt is enabled
0 = TMR6 to PR6 match is not enabled

bit 6 **C2IE:** Comparator C2 Interrupt Enable bit
1 = Enables the Comparator C2 interrupt
0 = Disables the Comparator C2 interrupt

bit 5 **C1IE:** Comparator C1 Interrupt Enable bit
1 = Enables the Comparator C1 interrupt
0 = Disables the Comparator C1 interrupt

bit 4 **NVMIE:** NVM Interrupt Enable Bit
1 = ENVM task complete interrupt enabled
0 = NVM interrupt not enabled

bit 3 **SSP2IE:** Master Synchronous Serial Port (MSSP2) Interrupt Enable bit
1 = Enables the MSSP2 interrupt
0 = Disables the MSSP2 interrupt

bit 2 **BCL2IE:** MSSP2 Bus Collision Interrupt Enable bit
1 = MSSP bus collision interrupt enabled
0 = MSSP bus collision interrupt not enabled

bit 1 **TMR4IE:** TMR4 to PR4 Match Interrupt Enable bit
1 = TMR4 to PR4 match interrupt is enabled
0 = TMR4 to PR4 match is not enabled

bit 0 **NCO1IE:** NCO Interrupt Enable bit
1 = NCO rollover interrupt enabled
0 = NCO rollover interrupt not enabled

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

REGISTER 7-5: PIE3: PERIPHERAL INTERRUPT ENABLE REGISTER 3

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
OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE
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 **OSFIE**: Oscillator Fail Interrupt Enable bit.
1 = Enables the Oscillator Fail interrupt
0 = Disables the Oscillator Fail interrupt

bit 6 **CSWIE:** Clock Switch Complete Interrupt Enable bit
1 = The clock switch module interrupt is enabled
0 = The clock switch module interrupt is not enabled

bit 5 **TMR3GIE:** Timer3 Gate Interrupt Enable bit
1 = Timer3 Gate interrupt is enabled
0 = Timer3 Gate interrupt is not enabled

bit 4 **TMR3IE:** TMR3 Overflow Interrupt Enable bit
1 = TMR3 overflow interrupt is enabled
0 = TMR3 overflow interrupt is not enabled

bit 3 **CLC4IE:** CLC4 Interrupt Flag bit
1 = CLC4 interrupt is enabled
0 = CLC4 interrupt is not enabled

bit 2 **CLC3IE:** CLC3 Interrupt Flag bit
1 = CLC3 interrupt is enabled
0 = CLC3 interrupt is not enabled

bit 1 **CLC2IE:** CLC2 Interrupt Enable bit
1 = CLC2 interrupt enabled
0 = CLC2 interrupt disabled

bit 0 **CLC1IE:** CLC1 Interrupt Enable bit
1 = CLC1 interrupt enabled
0 = CLC1 interrupt disabled

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

REGISTER 7-6: PIE4: PERIPHERAL INTERRUPT ENABLE REGISTER 4

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
CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE
bit 7							bit 0

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

bit 7 **CWG2IE:** CWG 2 Interrupt Enable bit
1 = CWG2 interrupt enabled
0 = CWG2 interrupt not enabled

bit 6 **CWG1IE:** CWG 1 Interrupt Enable bit
1 = CWG1 interrupt enabled
0 = CWG1 interrupt not enabled

bit 5 **TMR5GIE:** Timer5 Gate Interrupt Enable bit
1 = TMR5 Gate interrupt is enabled
0 = TMR5 Gate interrupt is not enabled

bit 4 **TMR5IE:** TMR5 Overflow Interrupt Enable bit
1 = TMR5 overflow interrupt is enabled
0 = TMR5 overflow interrupt is not enabled

bit 3 **CCP4IE:** CCP4 Interrupt Enable bit
1 = CCP4 interrupt is enabled
0 = CCP4 interrupt is not enabled

bit 2 **CCP3IE:** CCP3 Interrupt Enable bit
1 = CCP3 interrupt is enabled
0 = CCP3 interrupt is not enabled

bit 1 **CCP2IE:** CCP2 Interrupt Enable bit
1 = CCP2 interrupt is enabled
0 = CCP2 interrupt is not enabled

bit 0 **CCP1IE:** CCP1 Interrupt Enable bit
1 = CCP1 interrupt is enabled
0 = CCP1 interrupt is not enabled

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

REGISTER 7-7: PIR0: PERIPHERAL INTERRUPT REQUEST REGISTER 0

U-0	U-0	R/W/HS-0/0	R-0	U-0	U-0	U-0	R/W/HS-0/0
—	—	TMR0IF	IOCIF	—	—	—	INTF(1)
bit 7							bit 0

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

bit 7-6 **Unimplemented:** Read as '0'

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

bit 4 **IOCIF:** Interrupt-on-Change Interrupt Flag bit (read-only)
- 1 = An enabled edge was detected by the IOC module. One of the IOCF bits is set.
- 0 = No enabled edge is was detected by the IOC module. None of the IOCF bits is set.

Pins are individually masked via IOCxP and IOCxN.

bit 3-1 **Unimplemented:** Read as '0'

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

Note 1: The External Interrupt GPIO pin is selected by INTPPS (Register 12-1).

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

REGISTER 7-8: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1

R/W/HS-0/0	R/W/HS-0/0	R/W-0/0	R/W-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF
bit 7							bit 0

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

bit 7 **TMR1GIF**: Timer1 Gate Interrupt Flag bit
1 = The Timer1 gate has gone inactive (the gate is closed).
0 = The Timer1 gate has not gone inactive.

bit 6 **ADIF:** Analog-to-Digital Converter (ADC) Interrupt Flag bit
1 = The A/D conversion completed
0 = The A/D conversion is not completed

bit 5 **RCIF:** EUSART Receive Interrupt Flag bit
1 = The EUSART1 receive buffer is not empty
0 = The EUSART1 receive buffer is empty

bit 4 **TXIF**: EUSART Transmit Interrupt Flag bit
1 = The EUSART1 receive buffer is not empty
0 = The EUSART1 receive buffer is empty

bit 3 **SSP1IF**: Synchronous Serial Port (MSSP) Interrupt Flag bit
1 = The Transmission/Reception/Bus Condition is complete (must be cleared in software)
0 = Waiting for the Transmission/Reception/Bus Condition in progress

bit 2 **BCL1IF:** MSSP Bus Collision Interrupt Flag bit
1 = A bus collision was detected (must be cleared in software)
0 = No bus collision was detected

bit 1 **TMR2IF:** Timer2 to PR2 Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF:** Timer1 Overflow Interrupt Flag bit
1 = TMR1 overflow occurred (must be cleared in software)
0 = No TMR1 overflow occurred

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

REGISTER 7-9: PIR2: PERIPHERAL INTERRUPT REQUEST REGISTER 2

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BCL2IF	TMR4IF	NCO1IF
bit 7							bit 0

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

bit 7 **TMR6IF:** TMR6 to PR6 Match Interrupt Flag bit
1 = TMR6 to PR6 match occurred (must be cleared in software)
0 = No TMR6 to PR6 match occurred

bit 6 **C2IF:** Comparator C2 Interrupt Flag bit
1 = Comparator 2 interrupt asserted
0 = Comparator 2 interrupt not asserted

bit 5 **C1IF:** Comparator C1 Interrupt Flag bit
1 = Comparator 1 interrupt asserted
0 = Comparator 1 interrupt not asserted

bit 4 **NVMIF**: NVM Interrupt Flag bit
1 = The NVM has completed a programming task
0 = NVM interrupt not asserted

bit 3 **SSP2IF:** Master Synchronous Serial Port (MSSP2) Interrupt Flag bit
1 = The Transmission/Reception/Bus Condition is complete (must be cleared in software)
0 = Waiting for the Transmission/Reception/Bus Condition in progress

bit 2 **BCL2IF:** MSSP2 Bus Collision Interrupt Flag bit
1 = A bus collision was detected (must be cleared in software)
0 = No bus collision was detected

bit 1 **TMR4IF:** TMR4 to PR4 Match Interrupt Flag bit
1 = TMR4 to PR4 match occurred (must be cleared in software)
0 = No TMR4 to PR4 match occurred

bit 0 **NCO1IF:** NCO Interrupt Flag bit
1 = The NCO has rolled over.
0 = No NCO interrupt is asserted.

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

REGISTER 7-10: PIR3: PERIPHERAL INTERRUPT REQUEST REGISTER 3

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
OSFIF	CSWIF	TMR3GIF	TMR3IF	CLC4IF	CLC3IF	CLC2IF	CLC1IF
bit 7							bit 0

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

bit 7 **OSFIF:** Oscillator Fail-Safe Interrupt Flag bit
1 = Oscillator fail-safe interrupt has occurred
0 = No oscillator fail-safe interrupt

bit 6 **CSWIF:** Clock Switch Complete Interrupt Flag bit
1 = The clock switch module indicates an interrupt condition
0 = The clock switch module does not indicate an interrupt condition

bit 5 **TMR3GIF:** Timer3 Gate Interrupt Flag bit
1 = The TMR3 gate has gone inactive (the gate is closed).
0 = The TMR3 gate has not gone inactive.

bit 4 **TMR3IF:** TMR3 Overflow Interrupt Flag bit
1 = TMR3 overflow occurred (must be cleared in software)
0 = No TMR3 overflow occurred

bit 3 **CLC4IF:** CLC4 Interrupt Flag bit
1 = The CLC4OUT interrupt condition has been met
0 = No CLC4 interrupt

bit 2 **CLC3IF:** CLC3 Interrupt Flag bit
1 = The CLC3OUT interrupt condition has been met
0 = No CLC3 interrupt

bit 1 **CLC2IF:** CLC2 Interrupt Flag bit
1 = The CLC2OUT interrupt condition has been met
0 = No CLC2 interrupt

bit 0 **CLC1IF:** CLC1 Interrupt Flag bit
1 = The CLC1OUT interrupt condition has been met
0 = No CLC1 interrupt

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

REGISTER 7-11: PIR4: PERIPHERAL INTERRUPT REQUEST REGISTER 4

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF
bit 7							bit 0

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

bit 7 **CWG2IF:** CWG 2 Interrupt Flag bit
1 = CWG2 has gone into shutdown
0 = CWG2 is operating normally, or interrupt cleared

bit 6 **CWG1IF:** CWG1 Interrupt Flag bit
1 = CWG1 has gone into shutdown
0 = CWG1 is operating normally, or interrupt cleared

bit 5 **TMR5GIF:** Timer5 Gate Interrupt Flag bit
1 = The TMR5 gate has gone inactive (the gate is closed).
0 = The TMR5 gate has not gone inactive.

bit 4 **TMR5IF:** Timer5 Overflow Interrupt Flag bit
1 = TMR5 overflow occurred (must be cleared in software)
0 = No TMR5 overflow occurred

bit 3 **CCP4IF:** CCP4 Interrupt Flag bit

Value	CCPM Mode		
	Capture	Compare	PWM
1	Capture occurred (must be cleared in software)	Compare match occurred (must be cleared in software)	Output trailing edge occurred (must be cleared in software)
0	Capture did not occur	Compare match did not occur	Output trailing edge did not occur

bit 2 **CCP3IF:** CCP3 Interrupt Flag bit

Value	CCPM Mode		
	Capture	Compare	PWM
1	Capture occurred (must be cleared in software)	Compare match occurred (must be cleared in software)	Output trailing edge occurred (must be cleared in software)
0	Capture did not occur	Compare match did not occur	Output trailing edge did not occur

REGISTER 7-11: PIR4: PERIPHERAL INTERRUPT REQUEST REGISTER 4 (CONTINUED)

bit 1 **CCP2IF:** CCP2 Interrupt Flag bit

Value	CCPM Mode		
	Capture	Compare	PWM
1	Capture occurred (must be cleared in software)	Compare match occurred (must be cleared in software)	Output trailing edge occurred (must be cleared in software)
0	Capture did not occur	Compare match did not occur	Output trailing edge did not occur

bit 0 **CCP1IF:** CCP1 Interrupt Flag bit

Value	CCPM Mode		
	Capture	Compare	PWM
1	Capture occurred (must be cleared in software)	Compare match occurred (must be cleared in software)	Output trailing edge occurred (must be cleared in software)
0	Capture did not occur	Compare match did not occur	Output trailing edge did not occur

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

TABLE 7-1: 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	—	—	—	—	—	INTEDG	98
PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	99
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	100
PIE2	TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BCL2IE	TMR4IE	NCO1IE	101
PIE3	OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	102
PIE4	CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	103
PIR0	—	—	TMR0IF	IOCIF	—	—	—	INTF	104
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	105
PIR2	TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BCL2IF	TMR4IF	NCO1IF	106
PIR3	OSFIF	CSWIF	TMR3GIF	TMR3IF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	107
PIR4	CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	108

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

8.0 POWER-SAVING OPERATION MODES

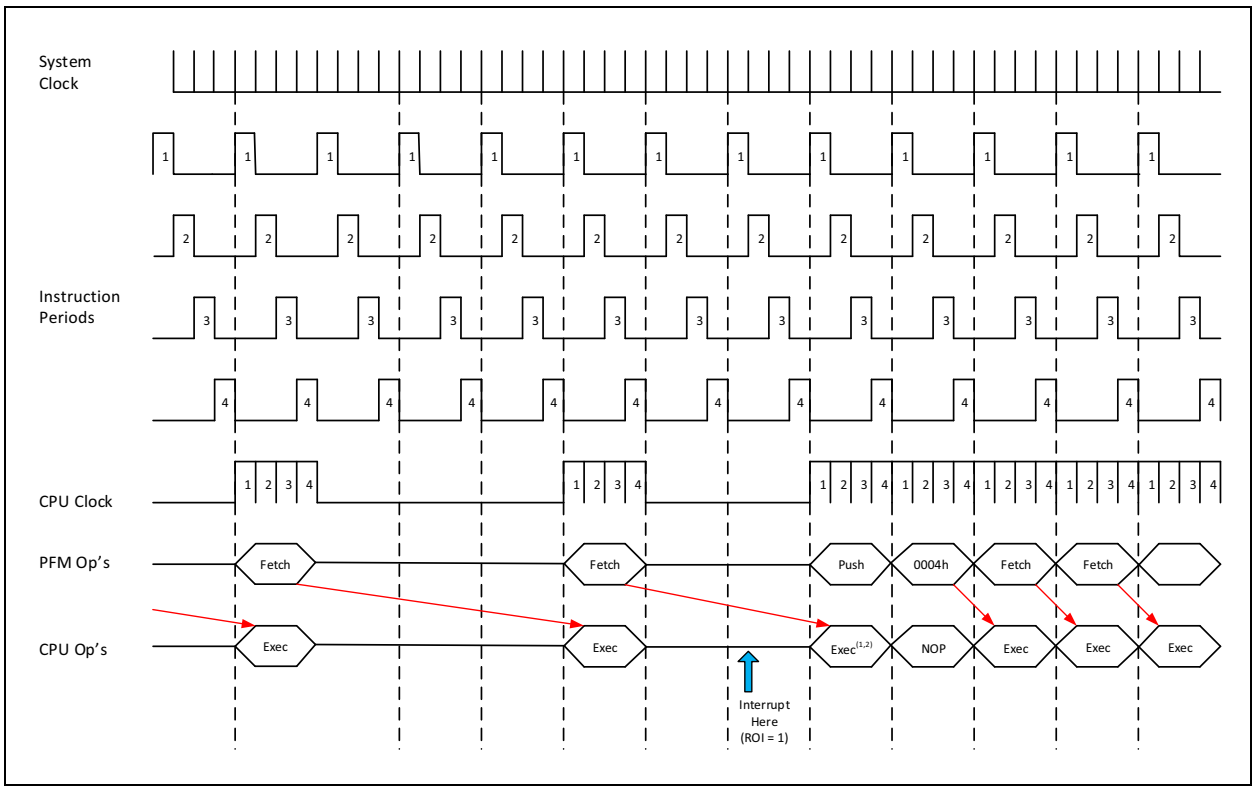
The purpose of the Power-Down modes is to reduce power consumption. There are two Power-Down modes: Doze mode and Sleep mode.

8.1 Doze Mode

Doze mode allows for power savings by reducing CPU operation and program memory access, without affecting peripheral operation. Doze mode differs from Sleep mode because the system oscillators continue to operate, while only the CPU and program memory are affected. The reduced execution saves power by eliminating unnecessary operations within the CPU and memory.

When the Doze Enable (DOZEN) bit is set (DOZEN = 1), the CPU executes only one instruction cycle out of every N cycles as defined by the DOZE<2:0> bits of the CPUDOZE register. For example, if DOZE<2:0> = 100, the instruction cycle ratio is 1:32. The CPU and memory execute for one instruction cycle and then lay idle for 31 instruction cycles. During the unused cycles, the peripherals continue to operate at the system clock speed.

FIGURE 8-1: DOZE MODE OPERATION EXAMPLE



8.1.1 DOZE OPERATION

The Doze operation is illustrated in Figure 8-1. For this example:

- Doze enable (DOZEN) bit set (DOZEN = 1)
- DOZE<2:0> = 001 (1:4) ratio
- Recover-on-Interrupt (ROI) bit set (ROI = 1)

As with normal operation, the program memory fetches for the next instruction cycle. The instruction clocks to the peripherals continue throughout.

8.1.2 INTERRUPTS DURING DOZE

If an interrupt occurs and the Recover-on-Interrupt (ROI) bit is clear (ROI = 0) at the time of the interrupt, the Interrupt Service Routine (ISR) continues to execute at the rate selected by DOZE<2:0>. Interrupt latency is extended by the DOZE<2:0> ratio.

If an interrupt occurs and the ROI bit is set (ROI = 1) at the time of the interrupt, the DOZEN bit is cleared and the CPU executes at full speed. The prefetched instruction is executed and then the interrupt vector sequence is executed. In Figure 8-1, the interrupt occurs during the 2nd instruction cycle of the Doze period, and immediately brings the CPU out of Doze. If the Doze-on-Exit (DOE) bit is set (DOE = 1) when the RETFIE operation is executed, DOZEN is set, and the CPU executes at the reduced rate based on the DOZE<2:0> ratio.

8.2 Sleep Mode

Sleep mode is entered by executing the SLEEP instruction, while the Idle Enable (IDLEN) bit of the CPUDOZE register is clear (IDLEN = 0). If the SLEEP instruction is executed while the IDLEN bit is set (IDLEN = 1), the CPU will enter the Idle mode (**Section 8.2.3 “Low-Power Sleep Mode”**).

Upon entering Sleep mode, the following conditions exist:

1. WDT will be cleared but keeps running if enabled for operation during Sleep
2. The $\overline{PD}$ bit of the STATUS register is cleared
3. The $\overline{TO}$ bit of the STATUS register is set
4. The CPU clock is disabled
5. 31 kHz LFINTOSC, HFINTOSC and SOSC are unaffected and peripherals using them may continue operation in Sleep.
6. Timer1 and peripherals that use it continue to operate in Sleep when the Timer1 clock source selected is:

- LFINTOSC
- T1CKI
- Secondary Oscillator

7. ADC is unaffected if the dedicated ADCRC oscillator is selected
8. I/O ports maintain the status they had before SLEEP was executed (driving high, low, or high-impedance)
9. 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 any oscillator

I/O pins that are high-impedance inputs should be pulled to VDD or VSS externally to avoid switching currents caused by floating inputs.

Examples of internal circuitry that might be sourcing current include modules such as the DAC and FVR modules. See **Section 23.0 “5-bit Digital-to-Analog Converter (DAC1) Module”** and **Section 15.0 “Fixed Voltage Reference (FVR)”** for more information on these modules.

8.2.1 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 5.11 "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.

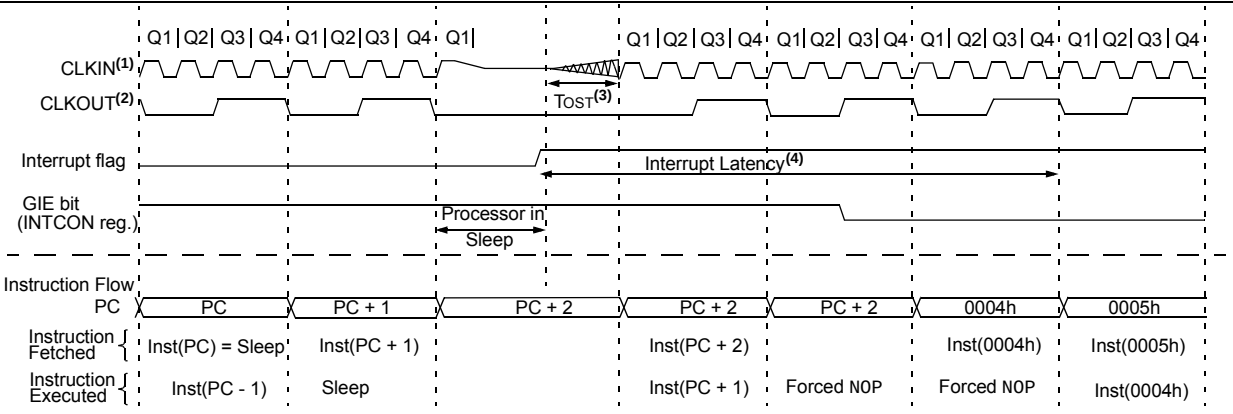
8.2.2 WAKE-UP USING INTERRUPTS

When global interrupts are disabled (GIE cleared) and any interrupt source, with the exception of the clock switch interrupt, 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
  - $\overline{TO}$ bit of the STATUS register will not be set
  - $\overline{PD}$ 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
  - $\overline{TO}$ bit of the STATUS register will be set
  - $\overline{PD}$ 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 $\overline{PD}$ bit. If the $\overline{PD}$ bit is set, the SLEEP instruction was executed as a NOP.

FIGURE 8-2: WAKE-UP FROM SLEEP THROUGH INTERRUPT



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

8.2.3 LOW-POWER SLEEP MODE

The PIC16F18325/18345 device contains an internal Low Dropout (LDO) voltage regulator, which allows the device I/O pins to operate at voltages up to 5.5V while the internal device logic operates at a lower voltage. The LDO and its associated reference circuitry must remain active when the device is in Sleep mode.

The PIC16F18325/18345 allows the user to optimize the operating current in Sleep, depending on the application requirements.

Low-Power Sleep mode can be selected by setting the VREGPM<1:0> bits of the VREGCON register. Depending on the configuration of these bits, the LDO and reference circuitry are placed in a low-power state when the device is in Sleep.

8.2.3.1 Sleep Current vs. Wake-up Time

In the default operating mode, the LDO and reference circuitry remain in the normal configuration while in Sleep. The device is able to exit Sleep mode quickly since all circuits remain active. In Low-Power Sleep mode, when waking-up from Sleep, an extra delay time is required for these circuits to return to the normal configuration and stabilize.

The Low-Power Sleep mode is beneficial for applications that stay in Sleep mode for long periods of time. The Normal mode is beneficial for applications that need to wake from Sleep quickly and frequently.

8.2.3.2 Peripheral Usage in Sleep

Some peripherals that can operate in Sleep mode will not operate properly with the Low-Power Sleep mode selected. The Low-Power Sleep mode is intended for use with these peripherals:

- Brown-out Reset (BOR)
- Watchdog Timer (WDT)
- External interrupt pin/Interrupt-on-change pins
- Timer 1 (with external clock source)

It is the responsibility of the end user to determine what is acceptable for their application when setting the VREGPM settings in order to ensure operation in Sleep.

Note:	The PIC16LF18325/18345 does not have a configurable Low-Power Sleep mode. PIC16LF18325/18345 is an unregulated device and is always in the lowest power state when in Sleep, with no wake-up time penalty. This device has a lower maximum VDD and I/O voltage than the PIC16F18325/18345. See **Section 34.0 "Electrical Specifications"** for more information.

8.2.4 IDLE MODE

When the Idle Enable (IDLEN) bit is clear (IDLEN = 0), the SLEEP instruction will put the device into full Sleep mode (see **Section 8.2 "Sleep Mode"**). When IDLEN is set (IDLEN = 1), the SLEEP instruction will put the device into Idle mode. In Idle mode, the CPU and memory operations are halted, but the peripheral clocks continue to run. This mode is similar to Doze mode, except that in IDLE both the CPU and program memory are shut off.

Note:	Peripherals using FOSC will continue running while in Idle (but not in Sleep).

Note:	If CLKOUT is enabled (CLKOUT = 0, Configuration Word 1), the output will continue operating while in Idle.

8.2.4.1 Idle and Interrupts

Idle mode ends when an interrupt occurs (even if GIE = 0), but IDLEN is not changed. The device can re-enter IDLE by executing the SLEEP instruction.

If Recover-on-Interrupt is enabled (ROI = 1), the interrupt that brings the device out of Idle also restores full-speed CPU execution when DOZE is also enabled.

8.2.4.2 Idle and WDT

When in Idle, the WDT Reset is blocked and will instead wake the device. The WDT wake-up is not an interrupt, therefore ROI does not apply.

Note:	The WDT can bring the device out of Idle, in the same way it brings the device out of Sleep. The DOZEN bit is not affected.

8.3 Register Definitions: Voltage Regulator Control

REGISTER 8-1: VREGCON: VOLTAGE REGULATOR CONTROL REGISTER(1)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-1/1
—	—	—	—	—	—	VREGPM<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 7-2 **Unimplemented:** Read as '0'

bit 1-0 **VREGPM<1:0>:** Voltage Regulator Power Mode Selection bits
11 = Lowest Power mode; LDO is off; Band gap generator is on only if needed by peripherals; longest wake-up time
10 = Low-Power mode; LDO is off; Band gap generator is on
01 = Normal-Power mode (Reset default); LDO supplying low power
00 = High-Power mode; LDO supplying highest power; fastest wake-up time

Note 1: PIC16F18325/18345 only.

REGISTER 8-2: CPUDOZE: DOZE AND IDLE REGISTER

R/W-0/u	R/W/HC/HS-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
IDLEN	DOZEN(1,2)	ROI	DOE	—	DOZE<2:0>		
bit 7							bit 0

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

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

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

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

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

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

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

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

TABLE 8-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	99
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	100
PIE2	TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BCL2IE	TMR4IE	NCO1IE	101
PIE3	OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	102
PIE4	CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	103
PIR0	—	—	TMR0IF	IOCIF	—	—	—	INTF	104
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	105
PIR2	TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BCL2IF	TMR4IF	NCO1IF	106
PIR3	OSFIF	CSWIF	TMR3GIF	TMR3IF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	107
PIR4	CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	108
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	172
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	172
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	173
IOCBP(1)	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	173
IOCBN(1)	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	174
IOCBF(1)	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	174
IOCCP	IOCCP7(1)	IOCCP6(1)	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	175
IOCCN	IOCCN7(1)	IOCCN6(1)	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	175
IOCCF	IOCCF7(1)	IOCCF6(1)	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	176
STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	27
VREGCON(2)	—	—	—	—	—	—	VREGPM<1:0>		114
CPUDOZE	IDLEN	DOZEN	ROI	DOE	—	DOZE<2:0>			114
WDTCON	—	—	WDTPS<4:0>					SWDTEN	118

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

Note 1: PIC16(L)F18345 only.

2: PIC16F18325/18345 only.

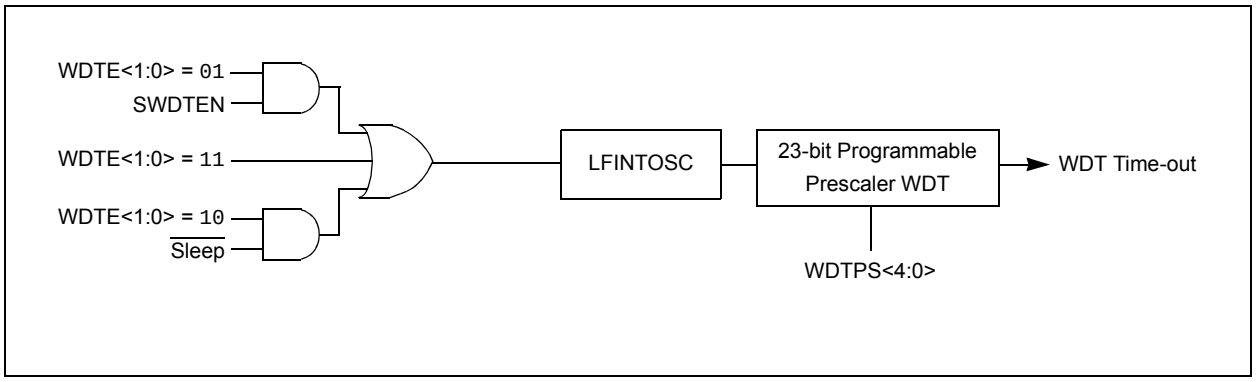
9.0 WATCHDOG TIMER (WDT)

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

The WDT has the following features:

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

FIGURE 9-1: WATCHDOG TIMER BLOCK DIAGRAM



9.1 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 1 ms. See Table 34-8 for the LFINTOSC specification.

9.2 WDT Operating Modes

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

9.2.1 WDT IS ALWAYS ON

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

WDT protection is active during Sleep.

9.2.2 WDT IS OFF IN SLEEP

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

WDT protection is not active during Sleep.

9.2.3 WDT CONTROLLED BY SOFTWARE

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

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

TABLE 9-1: 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

9.3 Time-out Period

The WDTPS<4:0> 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 two seconds.

9.4 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
- Oscillator Start-up Timer (OST) is running

See Table 9-2 for more information.

9.5 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. The WDT remains clear until the OST, if enabled, completes. See **Section 6.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for more information on the OST.

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 $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register are changed to indicate the event. See STATUS Register (Register 3-1) for more information.

TABLE 9-2: WDT CLEARING CONDITIONS

Conditions	WDT
WDTE = 00	Cleared and Disabled
WDTE = 01 and SWDTEN = 0	
Exit Sleep due to a Reset + System Clock = XT, HS, LP	Cleared until the end of OST
Exit Sleep due to a Reset + System Clock = HFINTOSC, LFINTOSC, EC, SOSC	
Exit Sleep due to an interrupt	Cleared
Enter Sleep	
CLRWDT Command	
Oscillator Failure (see **Section 6.4 "Fail-Safe Clock Monitor"**)	
System Reset	
Any clock switch or divider change (see **Section 6.3 "Clock Switching"**)	Unaffected

9.6 Register Definitions: Watchdog Control

REGISTER 9-1: 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>(1)					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	-m/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> = 1x:
This bit is ignored.
If WDTE<1:0> = 01:
1 = WDT is turned on
0 = WDT is turned off
If WDTE<1:0> = 00:
This bit is ignored.

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

TABLE 9-3: 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
STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	27
WDTCON	—	—	WDTPS<4:0>					SWDTEN	118

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

TABLE 9-4: 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
CONFIG2	13:8	—	—	$\overline{DEBUG}$	STVREN	PPS1WAY	—	BORV	—	61
	7:0	BOREN1	BOREN0	$\overline{LPBOREN}$	—	WDTE1	WDTE0	$\overline{PWRTE}$	MCLRE	

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

10.0 NONVOLATILE MEMORY (NVM) CONTROL

NVM is separated into two types: Program Flash Memory and Data EEPROM.

NVM is accessible by using both the FSR and INDF registers, or through the NVMREG register interface.

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

NVM can be protected in two ways; by either code protection or write protection.

Code protection ($\overline{CP}$ and $\overline{CPD}$ bits in Configuration Word 4) disables access, reading and writing, to both the program Flash memory and EEPROM via external device programmers. Code protection does not affect the self-write and erase functionality. Code protection can only be Reset by a device programmer performing a Bulk Erase to the device, clearing all nonvolatile memory, Configuration bits, and user IDs.

Write protection prohibits self-write and erase to a portion or all of the program Flash memory, as defined by the WRT<1:0> bits of Configuration Word 3. Write protection does not affect a device programmer's ability to read, write, or erase the device.

10.1 Program Flash Memory

Program Flash memory consists of 8192 14-bit words as user memory, with additional words for user ID information, Configuration Words, and interrupt vectors. Program Flash memory provides storage locations for:

- User program instructions
- User defined data

Program Flash memory data can be read and/or written to through:

- CPU instruction fetch (read-only)
- FSR/INDF indirect access (read-only) (**Section 10.3 "FSR and INDF Access"**)
- NVMREG access (**Section 10.4 "NVMREG Access"**
- In-Circuit Serial Programming (ICSP™)

Read operations return a single word of memory. When write and erase operations are done on a row basis, the row size is defined in Table 10-1. Program Flash memory will erase to a logic '1' and program to a logic '0'.

TABLE 10-1: FLASH MEMORY ORGANIZATION BY DEVICE

Device	Row Erase (words)	Write Latches (words)
PIC16(L)F18325	32	32
PIC16(L)F18345		

It is important to understand the program Flash memory structure for erase and programming operations. Program Flash memory is arranged in rows. A row consists of 32 14-bit program memory words. A row is the minimum size that can be erased by user software.

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

Note:	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, the new data and retained data can be written into the write latches to reprogram the row of program Flash 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

10.1.1 PROGRAM MEMORY VOLTAGES

The program Flash memory is readable and writable during normal operation over the full VDD range.

10.1.1.1 Programming Externally

The program memory cell and control logic support write and Bulk Erase operations down to the minimum device operating voltage. Special BOR operation is enabled during Bulk Erase (**Section "BOR protection is unchanged by Sleep."**).

10.1.1.2 Self-Programming

The program memory cell and control logic will support write and row erase operations across the entire VDD range. Bulk Erase is not supported when self-programming.

10.2 Data EEPROM

Data EEPROM consists of 256 bytes of user data memory. The EEPROM provides storage locations for 8-bit user defined data.

EEPROM can be read and/or written through:

- FSR/INDF indirect access (**Section 10.3 "FSR and INDF Access"**)
- NVMREG access (**Section 10.4 "NVMREG Access"**)
- In-Circuit Serial Programming (ICSP)

Unlike program Flash memory, which must be written to by row, EEPROM can be written to word by word.

10.3 FSR and INDF Access

The FSR and INDF registers allow indirect access to the program Flash memory or EEPROM.

10.3.1 FSR READ

With the intended address loaded into an FSR register, a `MOVIW` instruction or read of INDF will read data from the program Flash memory or EEPROM.

Reading from NVM requires one instruction cycle. The CPU operation is suspended during the read, and resumes immediately after. Read operations return a single word of memory.

10.3.2 FSR WRITE

Writing/erasing the NVM through the FSR registers (ex. `MOVWI` instruction) is not supported in the PIC16(L)F18325/18345 devices.

10.4 NVMREG Access

The NVMREG interface allows read/write access to all the locations accessible by FSRs, and also read/write access to the user ID locations, and read-only access to the device identification, revision, and Configuration data.

Reading, writing, or erasing of NVM via the NVMREG interface is prevented when the device is code-protected.

10.4.1 NVMREG READ OPERATION

To read a NVM location using the NVMREG interface, the user must:

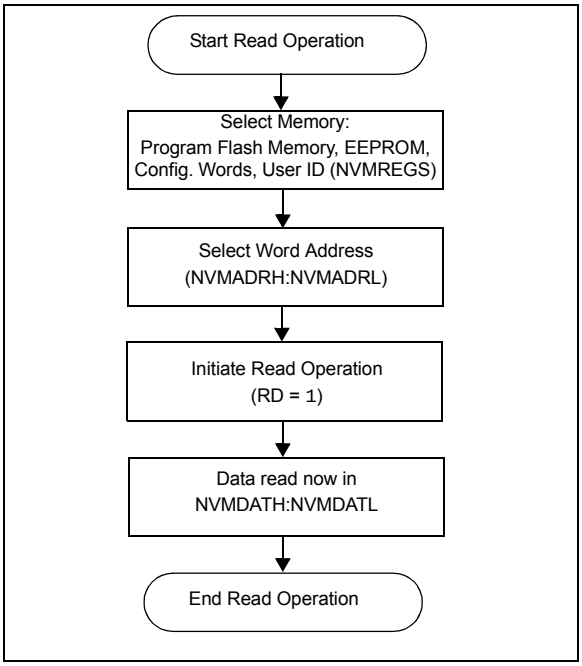
1. Clear the NVMREGS bit of the NVMCON1 register if the user intends to access program Flash memory locations, or set NVMREGS if the user intends to access user ID, Configuration, or EEPROM locations.
2. Write the desired address into the NVMADRH:NVMADRL register pair (Table 10-2).
3. Set the RD bit of the NVMCON1 register to initiate the read.

Once the read control bit is set, the CPU operation is suspended during the read, and resumes immediately after. The data is available in the very next cycle, in the NVMDATH:NVMDATL register pair; therefore, it can be read as two bytes in the following instructions.

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

Upon completion, the RD bit is cleared by hardware.

FIGURE 10-1: PROGRAM FLASH MEMORY READ FLOWCHART



EXAMPLE 10-1: PROGRAM FLASH 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  NVMADRL          ; Select Bank for NVMCON registers
    MOVLW    PROG_ADDR_LO     ;
    MOVWF    NVMADRL          ; Store LSB of address
    MOVLW    PROG_ADDR_HI     ;
    MOVWF    NVMADRH          ; Store MSB of address

    BCF      NVMCON1,NVMREGS  ; Do not select Configuration Space
    BSF      NVMCON1,RD       ; Initiate read

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


10.4.2 NVM UNLOCK SEQUENCE

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

- Program Flash Memory Row Erase
- Load of Program Flash Memory write latches
- Write of Program Flash Memory write latches to Program Flash Memory memory
- Write of Program Flash Memory write latches to user IDs
- Write to EEPROM

The unlock sequence consists of the following steps and must be completed in order:

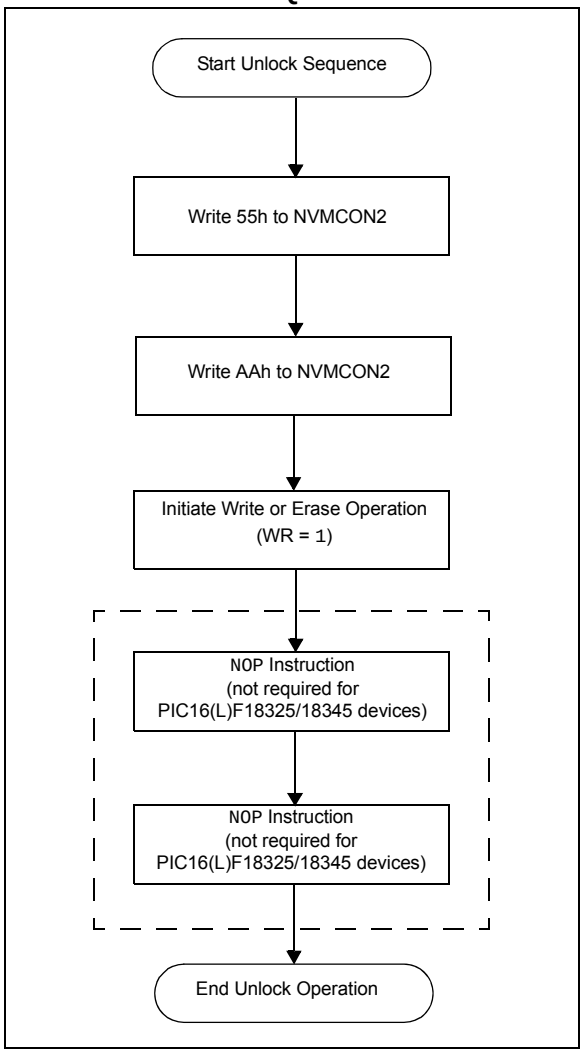
- Write 55h to NVMCON2
- Write AAh to NMVCON2
- Set the WR bit of NVMCON1

Once the WR bit is set, the processor will stall internal operations until the operation is complete and then resume with the next instruction.

Note:	The two NOP instructions after setting the WR bit, which were required in previous devices, are not required for PIC16(L)F18325/18345 devices. See Figure 10-2.

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 10-2: NVM UNLOCK SEQUENCE FLOWCHART



EXAMPLE 10-2: NVM UNLOCK SEQUENCE

```
BANKSEL     NVMCON1
BSF         NVMCON1,WREN    ; Enable write/erase
MOVLW       55h             ; Load 55h
BCF         INTCON,GIE      ; Recommended so sequence is not interrupted

MOVWF       NVMCON2         ; Step 1: Load 55h into NVMCON2
MOVLW       AAh             ; Step 2: Load W with AAh
MOVWF       NVMCON2         ; Step 3: Load AAh into NVMCON2
BSF         NVMCON1,WR      ; Step 4: Set WR bit to begin write/erase

BSF         INTCON,GIE      ; Re-enable interrupts
```

Note 1: Sequence begins when NVMCON2 is written; steps 1-4 must occur in the cycle-accurate order shown.

2: Opcodes shown are illustrative; any instruction that has the indicated effect may be used.

10.4.3 NVMREG WRITE TO EEPROM

Writing to the EEPROM is accomplished by the following steps:

1. Set the NVMREGS and WREN bits of the NVMCON1 register.
2. Write the desired address (address +7000h) into the NVMADRH:NVMADRL register pair (Table 10-2).
3. Perform the unlock sequence as described in **Section 10.4.2 "NVM Unlock Sequence"**.

A single EEPROM word is written with NVMDATA. The operation includes an implicit erase cycle for that word (it is not necessary to set the FREE bit), and requires many instruction cycles to finish. CPU execution continues in parallel and, when complete, WR is cleared by hardware, NVMIF is set, and an interrupt will occur if NVMIE is also set. Software must poll the WR bit to determine when writing is complete, or wait for the interrupt to occur. WREN will remain unchanged.

Once the EEPROM write operation begins, clearing the WR bit will have no effect; the operation will continue to run to completion.

10.4.4 NVMREG ERASE OF PROGRAM FLASH MEMORY

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

To erase a program Flash memory row:

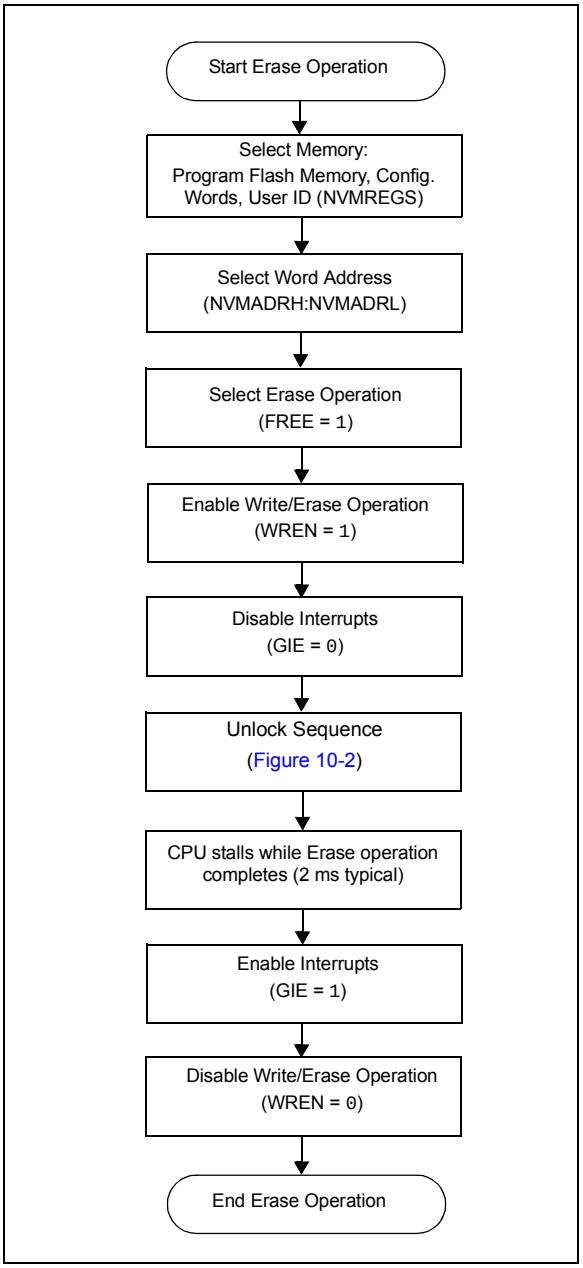
1. Clear the NVMREGS bit of the NVMCON1 register to erase program Flash memory locations, or set the NVMREGS bit to erase user ID locations.
2. Write the desired address into the NVMADRH:NVMADRL register pair (Table 10-2).
3. Set the FREE and WREN bits of the NVMCON1 register.
4. Perform the unlock sequence as described in **Section 10.4.2 "NVM Unlock Sequence"**.

If the program Flash memory address is write-protected, the WR bit will be cleared and the erase operation will not take place.

While erasing program Flash memory, CPU operation is suspended, and resumes when the operation is complete. Upon completion, the NVMIF is set, and an interrupt will occur if the NVMIE bit is also set.

Write latch data is not affected by erase operations, and WREN will remain unchanged.

FIGURE 10-3: NVM ERASE FLOWCHART



EXAMPLE 10-3: ERASING ONE ROW OF PROGRAM FLASH MEMORY

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

BANKSEL       NVMADRL
MOVF          ADDRL,W
MOVWF         NVMADRL              ; Load lower 8 bits of erase address boundary
MOVF          ADDRH,W
MOVWF         NVMADRH              ; Load upper 6 bits of erase address boundary
BCF           NVMCON1,NVMREGS      ; Choose Program Flash Memory area
BSF           NVMCON1,FREE         ; Specify an erase operation
BSF           NVMCON1,WREN         ; Enable writes
BCF           INTCON,GIE           ; Disable interrupts during unlock sequence

; ------------------------------REQUIRED UNLOCK SEQUENCE:-----------------------------

MOVLW         55h                  ; Load 55h to get ready for unlock sequence
MOVWF         NVMCON2              ; First step is to load 55h into NVMCON2
MOVLW         AAh                  ; Second step is to load AAh into W
MOVWF         NVMCON2              ; Third step is to load AAh into NVMCON2
BSF           NVMCON1,WR           ; Final step is to set WR bit

; ------------------------------------------------------------------------------------

BSF           INTCON,GIE           ; Re-enable interrupts, erase is complete
BCF           NVMCON1,WREN         ; Disable writes
```

TABLE 10-2: NVM ORGANIZATION AND ACCESS INFORMATION

Master Values			NVMREG Access			FSR Access	
Memory Function	**Program Counter (PC), ICSP™ Address**	**Memory Type**	**NVMREGS bit (NVMCON1)**	**NVMADR <14:0>**	**Allowed Operations**	**FSR Address**	**FSR Programming Address**
Reset Vector	0000h	Program Flash Memory	0	0000h	READ WRITE	8000h	READ-ONLY
User Memory	0001h		0	0001h		8001h	
	0003h			0003h		8003h	
INT Vector	0004h		0	0004h		8004h	
User Memory	0005h		0	0005h		8005h	
	17FFh			17FFh		FFFFh	
User ID	No PC Address	Program Flash Memory	1	0000h	READ	No Access	
				0003h			
Reserved		—	—	0004h	—		
Rev ID		Program Flash Memory	1	0005h	READ		
Device ID			1	0006h			
CONFIG1			1	0007h			
CONFIG2			1	0008h			
CONFIG3			1	0009h			
CONFIG4				000Ah			
User Memory		EEPROM	1	7000h	READ	F000h	READ-ONLY
				70FFh	WRITE	F0FFh	

10.4.5 NVMREG WRITE TO PROGRAM FLASH MEMORY

Program memory is programmed using the following steps:

1. Load the address of the row to be programmed into NVMADRH:NVMADRL.
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 10-4 (row writes to program memory with 32 write latches) for more details.

The write latches are aligned to the Flash row address boundary defined by the upper ten bits of NVMADRH:NVMADRL, (NVMADRH<6:0>:NVMADRL<7:5>) with the lower five bits of NVMADRL, (NVMADRL<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 NVMDATH:NVMDATL 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 NVMCON1 register.
2. Clear the NVMREGS bit of the NVMCON1 register.
3. Set the LWLO bit of the NVMCON1 register. When the LWLO bit of the NVMCON1 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 NVMADRH:NVMADRL register pair with the address of the location to be written.
5. Load the NVMDATH:NVMDATL register pair with the program memory data to be written.
6. Execute the unlock sequence (**Section 10.4.2 "NVM Unlock Sequence"**). The write latch is now loaded.
7. Increment the NVMADRH:NVMADRL 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 NVMCON1 register. When the LWLO bit of the NVMCON1 register is '0', the write sequence will initiate the write to Flash program memory.
10. Load the NVMDATH:NVMDATL register pair with the program memory data to be written.
11. Execute the unlock sequence (**Section 10.4.2 "NVM 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 10-4. The initial address is loaded into the NVMADRH:NVMADRL register pair; the data is loaded using indirect addressing.

FIGURE 10-4: BLOCK WRITES TO PROGRAM FLASH MEMORY WITH 32 WRITE LATCHES

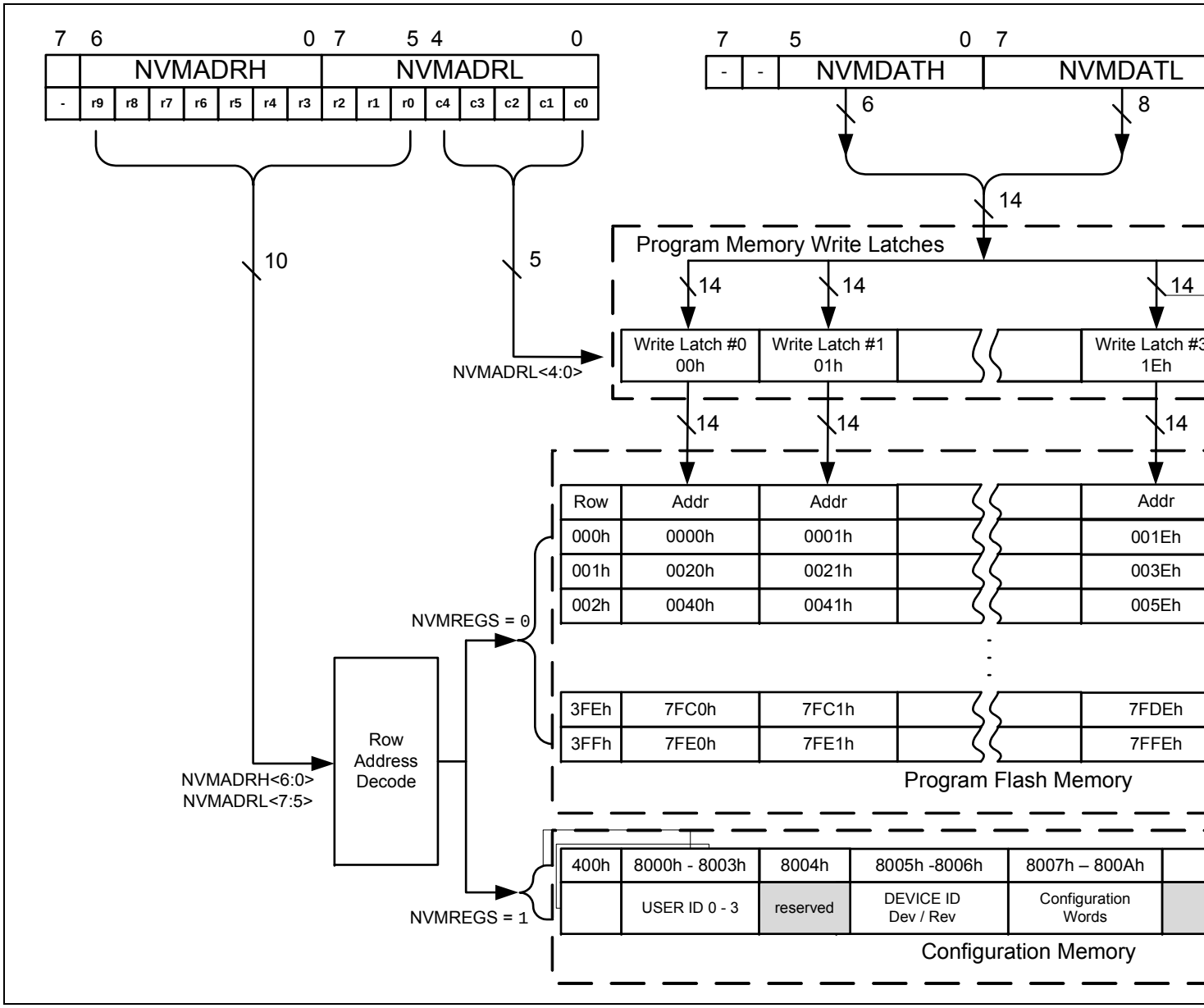
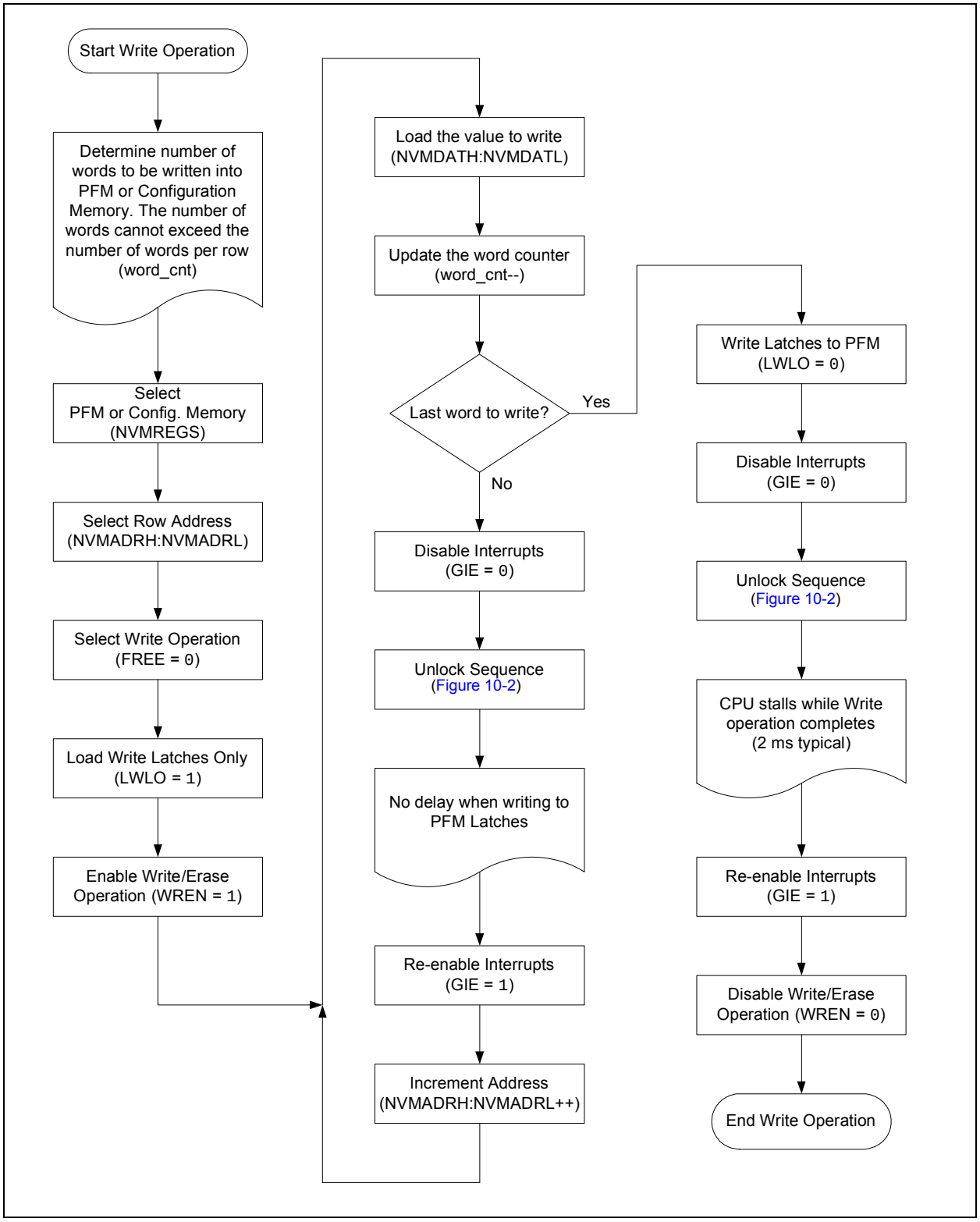


FIGURE 10-5: PROGRAM FLASH MEMORY WRITE FLOWCHART



EXAMPLE 10-4: WRITING TO PROGRAM FLASH MEMORY

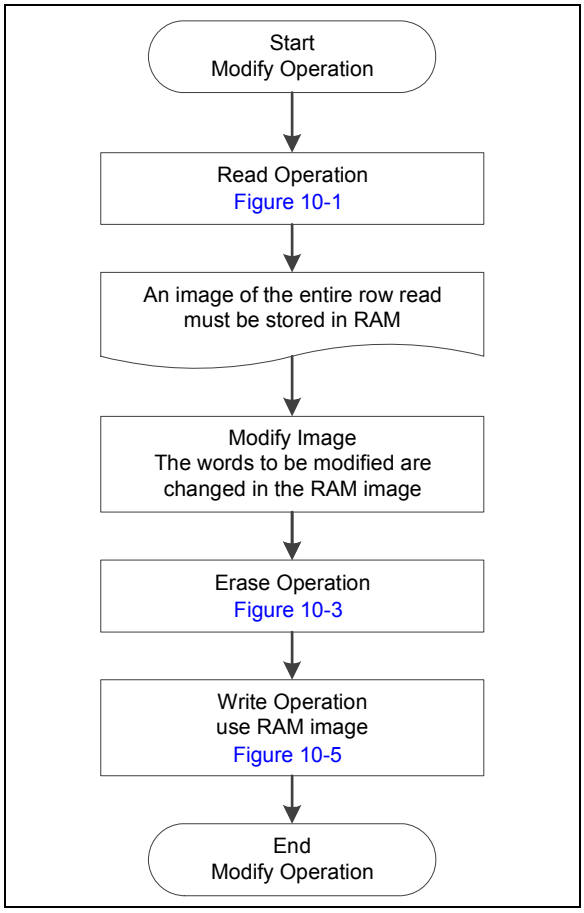
```
; This write routine assumes the following:
; 1. 64 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
;    stored in little endian format
; 3. A valid starting address (the least significant bits = 00000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)
; 5. NVM interrupts are not taken into account

        BANKSEL     NVMADRH
        MOVF        ADDRH,W
        MOVWF       NVMADRH             ; Load initial address
        MOVF        ADDRL,W
        MOVWF       NVMADRL
        MOVLW       LOW DATA_ADDR       ; Load initial data address
        MOVWF       FSR0L
        MOVLW       HIGH DATA_ADDR
        MOVWF       FSR0H
        BCF         NVMCON1,NVMREGS     ; Set Program Flash Memory as write location
        BSF         NVMCON1,WREN        ; Enable writes
        BSF         NVMCON1,LWLO        ; Load only write latches

LOOP
        MOVIW       FSR0++
        MOVWF       NVMDATL             ; Load first data byte
        MOVIW       FSR0++
        MOVWF       NVMDATH             ; Load second data byte

        MOVF        NVMADRL,W
        XORLW       0x1F                ; Check if lower bits of address are 00000
        ANDLW       0x1F                ; and if on last of 32 addresses
        BTFSC       STATUS,Z            ; Last of 32 words?
        GOTO        START_WRITE         ; If so, go write latches into memory

        CALL        UNLOCK_SEQ          ; If not, go load latch
        INCF        NVMADRL,F           ; Increment address
        GOTO        LOOP

START_WRITE
        BCF         NVMCON1,LWLO        ; Latch writes complete, now write memory
        CALL        UNLOCK_SEQ          ; Perform required unlock sequence
        BCF         NVMCON1,WREN        ; Disable writes

UNLOCK_SEQ
        MOVLW       55h
        BCF         INTCON,GIE          ; Disable interrupts
        MOVWF       NVMCON2             ; Begin unlock sequence
        MOVLW       AAh
        MOVWF       NVMCON2
        BSF         NVMCON1,WR
        BSF         INTCON,GIE          ; Unlock sequence complete, re-enable interrupts
        return
```

10.4.6 MODIFYING PROGRAM FLASH 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 10-6: PROGRAM FLASH MEMORY MODIFY FLOWCHART



10.4.7 NVMREG EEPROM, USER ID, DEVICE ID AND CONFIGURATION WORD ACCESS

Instead of accessing program Flash memory, the EEPROM, the user ID's, Device ID/Revision ID and Configuration Words can be accessed when NVMREGS = 1 in the NVMCON1 register. This is the region that would be pointed to by PC<15> = 1, but not all addresses are accessible. Different access may exist for reads and writes. Refer to Table 10-3.

When read access is initiated on an address outside the parameters listed in Table 10-3, the NVMDATH: NVMDATL register pair is cleared, reading back '0's.

TABLE 10-3: EEPROM, USER ID, DEV/REV ID AND CONFIGURATION WORD ACCESS (NVMREGS = 1)

Address	Function	Read Access	Write Access
8000h-8003h	User IDs	Yes	Yes
8005h-8006h	Device ID/Revision ID	Yes	No
8007h-800Ah	Configuration Words 1-4	Yes	No
F000h-F0FFh	EEPROM	Yes	Yes

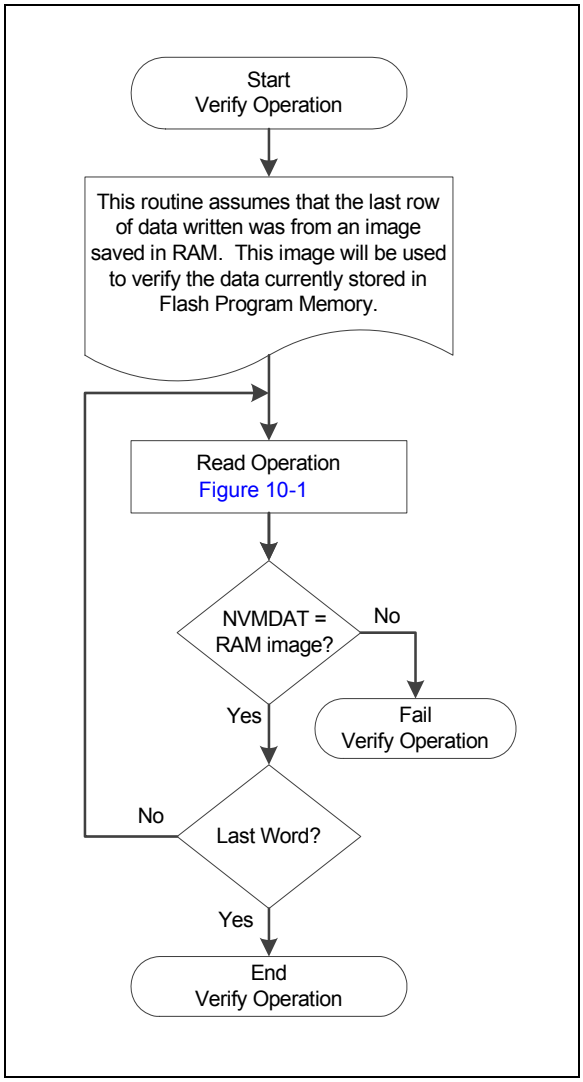
EXAMPLE 10-5: DEVICE ID ACCESS

```
; This write routine assumes the following:
; 1.  64 bytes of data are loaded, starting at the address in DATA_ADDR
; 2.  Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
; stored in little endian format
; 3.  A valid starting address (the least significant bits = 00000) is loaded in ADDRH:ADDRL
; 4.  ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)
; 5.  NVM interrupts are not taken into account

          BANKSEL             NVMADRH
          MOVF                ADDRH,W
          MOVWF               NVMADRH               ; Load initial address
          MOVF                ADDRL,W
          MOVWF               NVMADRL
          MOVLW               LOW DATA_ADDR         ; Load initial data address
          MOVWF               FSR0L
          MOVLW               HIGH DATA_ADDR
          MOVWF               FSR0H
          BCF                 NVMCON1,NVMREGS       ; Set Program Flash Memory as write location
          BSF                 NVMCON1,WREN          ; Enable writes
          BSF                 NVMCON1,LWLO          ; Load only write latches

LOOP
          MOVIW               FSR0++
          MOVWF               NVMDATL               ; Load first data byte
          MOVIW               FSR0++
          MOVWF               NVMDATH               ; Load second data byte

          MOVF                NVMADRL,W
          XORLW               0x1F                  ; Check if lower bits of address are 00000
          ANDLW               0x1F                  ; and if on last of 32 addresses
          BTFSC               STATUS,Z              ; Last of 32 words?
          GOTO                START_WRITE           ; If so, go write latches into memory

          CALL                UNLOCK_SEQ            ; If not, go load latch
          INCF                NVMADRL,F             ; Increment address
          GOTO                LOOP

START_WRITE
          BCF                 NVMCON1,LWLO          ; Latch writes complete, now write memory
          CALL                UNLOCK_SEQ            ; Perform required unlock sequence
          BCF                 NVMCON1,WREN          ; Disable writes

UNLOCK_SEQ
          MOVLW               55h
          BCF                 INTCON,GIE            ; Disable interrupts
          MOVWF               NVMCON2               ; Begin unlock sequence
          MOVLW               AAh
          MOVWF               NVMCON2
          BSF                 NVMCON1,WR
          BSF                 INTCON,GIE            ; Unlock sequence complete, re-enable interrupts
          return
```


10.4.8 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 10-7: PROGRAM FLASH MEMORY VERIFY FLOWCHART



10.4.9 WRERR BIT

The WRERR bit can be used to determine if a write error occurred.

WRERR will be set if one of the following conditions occurs:

- If WR is set while the NVMADRH:NMVADRL points to a write-protected address
- A Reset occurs while a self-write operation was in progress
- An unlock sequence was interrupted

The WRERR bit is normally set by hardware, but can be set by the user for test purposes. Once set, WRERR must be cleared in software.

TABLE 10-4: ACTIONS FOR PROGRAM FLASH MEMORY WHEN WR = 1

Free	LWLO	Actions for Program Flash Memory when WR = 1	Comments
0	0	Write the write latch data to program Flash memory row. See **Section 10.4.4 "NVMREG Erase of Program Flash Memory"**	• If WP is enabled, WR is cleared and WRERR is set • Write latches are reset to 3FFh • NVMDATH:NVMDATL is ignored
0	1	Copy NVMDATH:NVMDATL to the write latch corresponding to NVMADR LSBs. See **Section 10.4.4 "NVMREG Erase of Program Flash Memory"**	• Write protection is ignored • No memory access occurs
1	x	Erase the 32-word row of NVMADRH:NVMADRL location. See **Section 10.4.3 "NVMREG Write to EEPROM"**	• If WP is enabled, WR is cleared and WRERR is set • All 32 words are erased • NVMDATH:NVMDATL is ignored

10.5 Register Definitions: Program Flash Memory Control

REGISTER 10-1: NVMDATL: NONVOLATILE MEMORY DATA LOW BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NVMDAT<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 **NVMDAT<7:0>**: Read/write value for Least Significant bits of program memory

REGISTER 10-2: NVMDATH: NONVOLATILE MEMORY DATA HIGH BYTE REGISTER

U-0	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
—	—	NVMDAT<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 **NVMDAT<13:8>**: Read/write value for Most Significant bits of program memory

REGISTER 10-3: NVMADRL: NONVOLATILE MEMORY ADDRESS LOW BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NVMADR<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 **NVMADR<7:0>**: Specifies the Least Significant bits for program memory address

REGISTER 10-4: NVMADRH: NONVOLATILE MEMORY ADDRESS HIGH BYTE REGISTER

U-1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	NVMADR<14:8>						
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 **Unimplemented:** Read as ‘1’

bit 6-0 **NVMADR<14:8>**: Specifies the Most Significant bits for program memory address

REGISTER 10-5: NVMCON1: NONVOLATILE MEMORY CONTROL 1 REGISTER

U-0	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HC-x/q	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
—	NVMREGS	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 '0'

bit 6 **NVMREGS:** Configuration Select bit
1 = Access EEPROM, Configuration, user ID and device ID registers
0 = Access program Flash memory

bit 5 **LWLO:** Load Write Latches Only bit
When FREE = 0:
1 = The next WR command updates the write latch for this word within the row; no memory operation is initiated.
0 = The next WR command writes data or erases.
Otherwise: The bit is ignored.

bit 4 **FREE:** Program Flash Memory Erase Enable bit
When NVMREGS:NVMADR points to a program Flash memory location:
1 = Performs an erase operation with the next WR command; the 32-word pseudo-row containing the indicated address is erased (to all 1s) to prepare for writing.
0 = All write operations have completed normally.

bit 3 **WRERR:** Program/Erase Error Flag bit **(1,2,3)**
This bit is normally set by hardware.
1 = A write operation was interrupted by a Reset, interrupted unlock sequence, or WR was written to one while NVMADR points to a write-protected address.
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**(4,5,6)**
When NVMREG:NVMADR points to a EEPROM location:
1 = Initiates an erase/program cycle at the corresponding EEPROM location
0 = NVM program/erase operation is complete and inactive
When NVMREG:NVMADR points to a program Flash memory location:
1 = Initiates the operation indicated by Table 10-5
0 = NVM program/erase operation is complete and inactive
Otherwise: This bit is ignored.

bit 0 **RD:** Read Control bit**(7)**
1 = Initiates a read at address = NVMADR1, and loads data to NVMDAT Read takes one instruction cycle and the bit is cleared when the operation is complete. The bit can only be set (not cleared) in software.
0 = NVM read operation is complete and inactive.

Note 1: Bit is undefined while WR = 1 (during the EEPROM write operation it may be '0' or '1').
2: Bit must be cleared by software; hardware will not clear this bit.
3: Bit may be written to '1' by software in order to implement test sequences.
4: This bit can only be set by following the unlock sequence of **Section 10.4.2 "NVM Unlock Sequence"**.
5: Operations are self-timed, and the WR bit is cleared by hardware when complete.
6: Once a write operation is initiated, setting this bit to zero will have no effect.
7: Reading from EEPROM loads only NVMDATL<7:0> (Register 10-1).

REGISTER 10-6: NVMCON2: NONVOLATILE 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
NVMCON2							
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 **NVMCON2<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 NVMCON1 register. The value written to this register is used to unlock the writes.

TABLE 10-5: SUMMARY OF REGISTERS ASSOCIATED WITH NONVOLATILE MEMORY (NVM)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
PIR2	TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BLC2IF	TMR4IF	NCO1IF	106
PIE2	TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BLC2IE	TMR4IE	NCO1IE	101
NVMCON1	—	NVMREGS	LWLO	FREE	WRERR	WREN	WR	RD	136
NVMCON2	NVMCON2								137
NVMADRL	NVMADR<7:0>								135
NVMADRH	—(1)	NVMADR<14:8>							135
NVMDATL	NVMDAT<7:0>								135
NVMDATH	—	—	NVMDAT<13:8>						135

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by NVM.
Note 1: Unimplemented, read as '1'.

TABLE 10-6: SUMMARY OF CONFIGURATION WORD WITH NONVOLATILE MEMORY (NVM)

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
CONFIG3	13:8	—	—	LVP	—	—	—	—	—	62
	7:0	—	—	—	—	—	—	WRT<1:0>		
CONFIG4	13:8	—	—	—	—	—	—	—	—	63
	7:0	—	—	—	—	—	—	$\overline{CPD}$	$\overline{CP}$	

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

11.0 I/O PORTS

TABLE 11-1: PORT AVAILABILITY PER DEVICE

Device	PORTA	PORTB	PORTC
PIC16(L)F18325	•		•
PIC16(L)F18345	•	•	•

Each port has ten standard registers for its operation. These registers are:

- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)
- TRISx registers (data direction)
- ANSELx registers (analog select)
- WPUx registers (weak pull-up)
- INLVLx (input level control)
- SLRCONx registers (slew rate)
- ODCONx registers (open-drain)

Most port pins share functions with device peripherals, both analog and digital. 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.

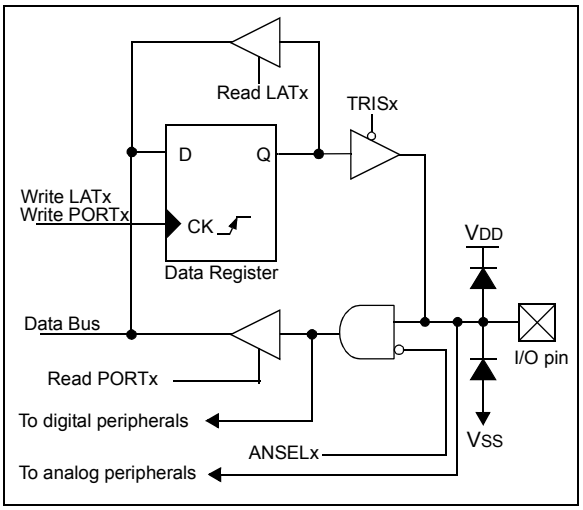
The Data Latch (LATx registers) is useful for read-modify-write operations on the value that the I/O pins are driving.

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

Ports that support analog inputs have an associated ANSELx 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 11-1.

FIGURE 11-1: GENERIC I/O PORT OPERATION



11.1 I/O Priorities

Each pin defaults to the PORT data latch after Reset. Other functions are selected with the peripheral pin select logic. See **Section 12.0 "Peripheral Pin Select (PPS) Module"** for more information.

Analog input functions, such as ADC and comparator inputs, are not shown in the peripheral pin select lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELx register. Digital output functions may continue to control the pin when it is in Analog mode.

Analog outputs, when enabled, take priority over the digital outputs and force the digital output driver to the high-impedance state.

11.2 PORTA Registers

11.2.1 DATA REGISTER

PORTA is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 11-2). 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). The exception is RA3, which is input-only and its TRIS bit will always read as '1'. Example 11-1 shows how to initialize PORTA.

Reading the PORTA register (Register 11-1) 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 PORT data latch LATA (Register 11-3) holds the output port data, and contains the latest value of a LATA or PORTA write.

EXAMPLE 11-1: INITIALIZING PORTA

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

BANKSEL  PORTA        ;
CLRF     PORTA        ;Init PORTA
BANKSEL  LATA         ;Data Latch
CLRF     LATA         ;
BANKSEL  ANSELA       ;
CLRF     ANSELA       ;digital I/O
BANKSEL  TRISA        ;
MOVLW    B'00111000'  ;Set RA<5:3> as inputs
MOVWF    TRISA        ;and set RA<2:0> as
                      ;outputs
```

11.2.2 DIRECTION CONTROL

The TRISA register (Register 11-2) 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 inputs always read '0'.

11.2.3 OPEN-DRAIN CONTROL

The ODCONA register (Register 11-6) controls the open-drain feature of the port. Open-drain operation is independently selected for each pin. When an ODCONA bit is set, the corresponding port output becomes an open-drain driver capable of sinking current only. When an ODCONA bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

Note:	It is not necessary to set open-drain control when using the pin for I^2C™; the I^2C module controls the pin and makes the pin open-drain.

11.2.4 SLEW RATE CONTROL

The SLRCONA register (Register 11-7) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONA bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONA bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

11.2.5 INPUT THRESHOLD CONTROL

The INLVLA register (Register 11-8) controls the input voltage threshold for each of the available PORTA input pins. A selection between the Schmitt Trigger CMOS or the TTL Compatible thresholds is available. The input threshold is important in determining the value of a read of the PORTA register and also the level at which an interrupt-on-change occurs, if that feature is enabled. See Table 34-4 for more information on threshold levels.

Note:	Changing the input threshold selection should be performed while all peripheral modules are disabled. Changing the threshold level during the time a module is active may inadvertently generate a transition associated with an input pin, regardless of the actual voltage level on that pin.

11.2.6 ANALOG CONTROL

The ANSELA register (Register 11-4) 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.

11.2.7 WEAK PULL-UP CONTROL

The WPUA register (Register 11-5) controls the individual weak pull-ups for each PORT pin.

PORTA pin RA3 includes the $\overline{MCLR}$/VPP input. The $\overline{MCLR}$ input allows the device to be reset, and can be disabled by the MCLRE bit of Configuration Word 2. A weak pull-up is present on the RA3 port pin. This weak pull-up is enabled when $\overline{MCLR}$ is enabled (MCLRE = 1) or the WPUA3 bit is set. The weak pull-up is disabled when is disabled and the WPUA3 bit is clear.

11.2.8 PORTA FUNCTIONS AND OUTPUT PRIORITIES

Each PORTA pin is multiplexed with other functions.

Each pin defaults to the PORT latch data after Reset. Other output functions are selected with the peripheral pin select logic. See **Section 12.0 "Peripheral Pin Select (PPS) Module"** for more information.

Analog input functions, such as ADC and comparator inputs are not shown in the peripheral pin select lists. Digital output functions may continue to control the pin when it is in Analog mode.

11.3 Register Definitions: PORTA

REGISTER 11-1: PORTA: PORTA REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	RA5	RA4	RA3(2)	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 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RA<5:0>**: PORTA I/O Value bits(1)
1 = Port pin is ≥ VIH
0 = Port pin is ≤ VIL

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

2: Bit RA3 is read-only, and will read '1' when MCLRE = 1 (master clear enabled).

REGISTER 11-2: TRISA: PORTA TRI-STATE REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	U-1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	TRISA5	TRISA4	—	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 7-6 **Unimplemented:** Read as '0'

bit 5-4 **TRISA<5:4>:** PORTA Tri-State Control bit
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

bit 3 **Unimplemented:** Read as '1'

bit 2-0 **TRISA<2:0>:** PORTA Tri-State Control bit
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

REGISTER 11-3: LATA: PORTA DATA LATCH REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LATA5	LATA4	—	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 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **LATA<5:4>**: RA<5:4> Output Latch Value bits(1)

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **LATA<2:0>**: RA<2:0> Output Latch Value bits(1)

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

REGISTER 11-4: ANSELA: PORTA ANALOG SELECT REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	ANSA5	ANSA4	—	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 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **ANSA<5:4>**: Analog Select between Analog or Digital Function on pins RA<5:4>, 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.

bit 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: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

REGISTER 11-5: WPUA: WEAK PULL-UP PORTA REGISTER

U-0	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
—	—	WPUA5	WPUA4	WPUA3(1)	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 7-6 **Unimplemented:** Read as '0'

bit 5-0 **WPUA<5:0>**: Weak Pull-up Register bits(2)
1 = Pull-up enabled
0 = Pull-up disabled

Note 1: If MCLRE = 1, the weak pull-up in RA3 is always enabled; bit WPUA3 is not affected.
2: The weak pull-up device is automatically disabled if the pin is configured as an output.

REGISTER 11-6: ODCONA: PORTA OPEN-DRAIN CONTROL REGISTER

U-0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	ODCA5	ODCA4	—	ODCA2	ODCA1	ODCA0
bit 7							bit 0

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

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **ODCA<5:4>:** PORTA Open-Drain Enable bits
For RA<5:4> pins, respectively
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

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

bit 2-0 **ODCA<2:0>:** PORTA Open-Drain Enable bits
For RA<2:0> pins, respectively
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

REGISTER 11-7: SLRCONA: PORTA SLEW RATE CONTROL REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0
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 5-4 **SLRA<5:4>:** PORTA Slew Rate Enable bits
For RA<5:4> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

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

bit 2-0 **SLRA<2:0>:** PORTA Slew Rate Enable bits
For RA<2:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

REGISTER 11-8: INLVLA: PORTA INPUT LEVEL CONTROL REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
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 **INLVLA<5:0>:** PORTA Input Level Select bits
For RA<5:0> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

TABLE 11-2: 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
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	141
TRISA	—	—	TRISA5	TRISA4	—	TRISA2	TRISA1	TRISA0	141
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	142
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	143
ODCONA	—	—	ODCA5	ODCA4	—	ODCA2	ODCA1	ODCA0	143
SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	144
INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	144

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

TABLE 11-3: SUMMARY OF CONFIGURATION WORD WITH PORTA

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
CONFIG2	13:8	—	—	$\overline{\text{DEBUG}}$	STVREN	PPS1WAY	—	BORV	—	61
	7:0	BOREN1	BOREN0	$\overline{\text{LPBOREN}}$	—	WDTE1	WDTE0	$\overline{\text{PWRTE}}$	MCLRE	

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

11.4 PORTB Registers (PIC16(L)F18345 Only)

11.4.1 DATA REGISTER

PORTB is a 4-bit wide bidirectional port and is only available in the PIC16(L)F18345 devices. The corresponding data direction register is TRISB (Register 11-10). Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 11-1 shows how to initialize an I/O port.

Reading the PORTB register (Register 11-9) 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 (LATB).

The PORT data latch LATB (Register 11-11) holds the output port data, and contains the latest value of a LATB or PORTB write.

11.4.2 DIRECTION CONTROL

The TRISB register (Register 11-10) controls the PORTB pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISB register are maintained set when using them as analog inputs. I/O pins configured as analog inputs always read '0'.

11.4.3 INPUT THRESHOLD CONTROL

The INLVLB register (Register 11-16) controls the input voltage threshold for each of the available PORTB input pins. A selection between the Schmitt Trigger CMOS or the TTL Compatible thresholds is available. The input threshold is important in determining the value of a read of the PORTB register and also the level at which an interrupt-on-change occurs, if that feature is enabled. See Table 34-4 for more information on threshold levels.

Note:	Changing the input threshold selection should be performed while all peripheral modules are disabled. Changing the threshold level during the time a module is active may inadvertently generate a transition associated with an input pin, regardless of the actual voltage level on that pin.

11.4.4 OPEN-DRAIN CONTROL

The ODCONB register (Register 11-14) controls the open-drain feature of the port. Open-drain operation is independently selected for each pin. When an ODCONB bit is set, the corresponding port output becomes an open-drain driver capable of sinking current only. When an ODCONB bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

Note:	It is not necessary to set open-drain control when using the pin for I^2C™; the I^2C module controls the pin and makes the pin open-drain.

11.4.5 SLEW RATE CONTROL

The SLRCONB register (Register 11-15) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONB bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONB bit is cleared, the corresponding port pin drive slews at the maximum rate possible.

11.4.6 ANALOG CONTROL

The ANSELB register (Register 11-12) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELB 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 ANSELB bits has no effect on digital output functions. A pin with TRIS clear and ANSELB 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 ANSELB 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.

11.4.7 WEAK PULL-UP CONTROL

The WPUB register (Register 11-13) controls the individual weak pull-ups for each PORT pin.

11.4.8 PORTB FUNCTIONS AND OUTPUT PRIORITIES

Each pin defaults to the PORT latch data after Reset. Other output functions are selected with the peripheral pin select logic. See **Section 12.0 "Peripheral Pin Select (PPS) Module"** for more information.

Analog input functions, such as ADC and comparator inputs, are not shown in the peripheral pin select lists. Digital output functions may continue to control the pin when it is in Analog mode.

11.5 Register Definitions: PORTB

REGISTER 11-9: PORTB: PORTB REGISTER

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	U-0	U-0	U-0	U-0
RB7	RB6	RB5	RB4	—	—	—	—
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 **RB<7:4>**: PORTB I/O Value bits(1)
1 = Port pin is ≥ VIH
0 = Port pin is ≤ VIL

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

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

REGISTER 11-10: TRISB: PORTB TRI-STATE REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—
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 **TRISB<7:4>:** PORTB I/O Tri-State Control bits
0 = PORTB output driver is enabled
1 = PORTB output driver is disabled

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

REGISTER 11-11: LATB: PORTB DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0
LATB7	LATB6	LATB5	LATB4	—	—	—	—
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 **LATB<7:4>**: RB<5:4> Output Latch Value bits(1)

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

Note 1: Writes to LATB are equivalent with writes to the corresponding PORTB register.Reads from LATB register return register values, not I/O pin values.

REGISTER 11-12: ANSELB: PORTB ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—
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 **ANSB<7:4>**: Analog Select between Analog or Digital Function
0 = Digital I/O. Pin is assigned to port or digital special function.
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.

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

Note 1: Setting ANSB[n] = 1 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 to allow external control of the voltage on the pin.

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

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—
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 **WPUB<7:4>**: Weak Pull-up Register bits
1 = Weak Pull-up enabled
0 = Weak Pull-up disabled

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

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

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
ODCB7	ODCB6	ODCB5	ODCB4	—	—	—	—
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 **ODCB<7:4>:** PORTB Open-Drain Configuration bits
For RB<7:4> pins, respectively:
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

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

REGISTER 11-15: SLRCONB: PORTB SLEW RATE CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—
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 **SLRB<7:4>:** PORTB Slew Rate Control on pins RB<7:4>, respectively
0 = Slew rate disabled
1 = Slew rate enabled

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

REGISTER 11-16: INLVLB: PORTB INPUT LEVEL CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—
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 **INLVLB<7:4>:** PORTB Input Level Select on pins RB<7:4>, respectively
0 = TTL input used for PORT reads
1 = ST input used for PORT reads

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

TABLE 11-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PORTB	RB7	RB6	RB5	RB4	—	—	—	—	147
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
LATB	LATB7	LATB6	LATB5	LATB4	—	—	—	—	148
ANSELB	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
WPUB	WPUB7	WPU6	WPUB5	WPUB4	—	—	—	—	149
ODCONB	ODCB7	ODCB6	ODCB5	ODCB4	—	—	—	—	149
SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	150
INLVLB	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	150

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

11.6 PORTC Registers

11.6.1 DATA REGISTER

PORTC is a bidirectional port that is either 6-bit wide (PIC16(L)F18325) or 8-bit wide (PIC16(L)F18345). The corresponding data direction register is TRISC (Register 11-18). Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 11-1 shows how to initialize an I/O port.

Reading the PORTC register (Register 11-17) 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 (LATC).

The PORT data latch LATC (Register 11-19) holds the output port data, and contains the latest value of a LATC or PORTC write.

11.6.2 DIRECTION CONTROL

The TRISC register (Register 11-18) controls the PORTC pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISC register are maintained set when using them as analog inputs. I/O pins configured as analog inputs always read '0'.

11.6.3 INPUT THRESHOLD CONTROL

The INLVLC register (Register 11-24) controls the input voltage threshold for each of the available PORTC input pins. A selection between the Schmitt Trigger CMOS or the TTL Compatible thresholds is available. The input threshold is important in determining the value of a read of the PORTC register and also the level at which an interrupt-on-change occurs, if that feature is enabled. See Table 34-4 for more information on threshold levels.

Note:	Changing the input threshold selection should be performed while all peripheral modules are disabled. Changing the threshold level during the time a module is active may inadvertently generate a transition associated with an input pin, regardless of the actual voltage level on that pin.

11.6.4 OPEN-DRAIN CONTROL

The ODCONC register (Register 11-22) controls the open-drain feature of the port. Open-drain operation is independently selected for each pin. When an ODCONC bit is set, the corresponding port output becomes an open-drain driver capable of sinking current only. When an ODCONC bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

Note:	It is not necessary to set open-drain control when using the pin for I^2C™; the I^2C module controls the pin and makes the pin open-drain.

11.6.5 SLEW RATE CONTROL

The SLRCONC register (Register 11-23) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONC bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONC bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

11.6.6 ANALOG CONTROL

The ANSELC register (Register 11-20) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELC 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 ANSELC bits has no effect on digital output functions. A pin with TRIS clear and ANSELC 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 ANSELC 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.

11.6.7 WEAK PULL-UP CONTROL

The WPUC register (Register 11-21) controls the individual weak pull-ups for each PORT pin.

11.6.8 PORTC FUNCTIONS AND OUTPUT PRIORITIES

Each pin defaults to the PORT latch data after Reset. Other functions are selected with the peripheral pin select logic. See **Section 12.0 "Peripheral Pin Select (PPS) Module"** for more information.

Analog output functions, such as ADC and comparator inputs, are not shown in the peripheral pin select lists. Digital output functions may continue to control the pin when it is in Analog mode.

11.7 Register Definitions: PORTC

REGISTER 11-17: PORTC: PORTC 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
RC7(1)	RC6(1)	RC5	RC4	RC3	RC2	RC1	RC0
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 **RC<7:6>**: PORTC I/O Value bits(1,2)
1 = Port pin is ≥ VIH
0 = Port pin is ≤ VIL

bit 5-0 **RC<5:0>**: PORTC General Purpose I/O Pin bits(2)
1 = Port pin is ≥ VIH
0 = Port pin is ≤ VIL

Note 1: PIC16(L)F18345 only; otherwise read as '0'.
2: Writes to PORTC are actually written to corresponding LATC register. Reads from PORTC register is return of actual I/O pin values.

REGISTER 11-18: TRISC: PORTC TRI-STATE REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
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 **TRISC<7:6>**: PORTC Tri-State Control bits(1)
1 = PORTC pin configured as an input (tri-stated)
0 = PORTC pin configured as an output

bit 5-0 **TRISC<5:0>:** PORTC Tri-State Control bits
1 = PORTC pin configured as an input (tri-stated)
0 = PORTC pin configured as an output

Note 1: PIC16(L)F18345 only; otherwise read as '0'.

REGISTER 11-19: LATC: PORTC DATA LATCH 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
LATC7(1)	LATC6(1)	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
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 **LATC<7:6>**: PORTC Output Latch Value bits(1)

bit 5-0 **LATC<5:0>**: PORTC Output Latch Value bits

Note 1: PIC16(L)F18345 only; otherwise read as '0'.

REGISTER 11-20: ANSELC: PORTC ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0
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 **ANSC<7:6>**: Analog Select between Analog or Digital Function on pins RC<7:6>, respectively(1)
0 = Digital I/O. Pin is assigned to port or digital special function.
1 = Analog input. Pin is assigned as analog input(2). Digital input buffer disabled.

bit 5-0 **ANSC<5:0>**: Analog Select between Analog or Digital Function on pins RC<5:0>, respectively
0 = Digital I/O. Pin is assigned to port or digital special function.
1 = Analog input. Pin is assigned as analog input(2). Digital input buffer disabled.

Note 1: PIC16(L)F18345 only; otherwise read as '0'.
2: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

REGISTER 11-21: WPUC: WEAK PULL-UP PORTC REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
WPUC7(1)	WPUC6(1)	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0
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 **WPUC<7:6>(1)**: Weak Pull-up Register bits(2)
1 = Pull-up enabled
0 = Pull-up disabled

bit 5-0 **WPUC<5:0>**: Weak Pull-up Register bits(2)
1 = Pull-up enabled
0 = Pull-up disabled

Note 1: PIC16(L)F18345 only; otherwise read as '0'.
2: The weak pull-up device is automatically disabled if the pin is configured as an output.

REGISTER 11-22: ODCONC: PORTC OPEN-DRAIN CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ODCC7(1)	ODCC6(1)	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0
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 **ODCC<7:6>**: PORTC Open-Drain Enable bits(1)
For RC<7:6> pins, respectively
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

bit 5-0 **ODCC<5:0>:** PORTC Open-Drain Enable bits
For RC<5:0> pins, respectively
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

Note 1: PIC16(L)F18345 only; otherwise read as '0'.

REGISTER 11-23: SLRCONC: PORTC SLEW RATE CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
SLRC7(1)	SLRC6(1)	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0
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 **SLRC<7:6>:** PORTC Slew Rate Enable bits(1)
For RC<7:6> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

bit 5-0 **SLRC<5:0>:** PORTC Slew Rate Enable bits
For RC<5:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

Note 1: PIC16(L)F18345 only; otherwise read as '0'.

REGISTER 11-24: INLVLC: PORTC INPUT LEVEL CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
INLVLC7(1)	INLVLC6(1)	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0
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 **INLVLC<7:6>:** PORTC Input Level Select bits(1)
For RC<7:6> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

bit 5-0 **INLVLC<5:0>:** PORTC Input Level Select bits
For RC<5:0> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

Note 1: PIC16(L)F18345 only; otherwise read as '0'.

TABLE 11-5: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PORTC	RC7(1)	RC6(1)	RC5	RC4	RC3	RC2	RC1	RC0	153
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
LATC	LATC7(1)	LATC6(1)	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	154
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
WPUC	WPUC7(1)	WPUC6(1)	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	155
ODCONC	ODCC7(1)	ODCC6(1)	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0	156
SLRCONC	SLRC7(1)	SLRC6(1)	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	156
INLVLC	INLVLC7(1)	INLVLC6(1)	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	157

Note 1: PIC16(L)F18345 only.

12.0 PERIPHERAL PIN SELECT (PPS) MODULE

The Peripheral Pin Select (PPS) module connects peripheral inputs and outputs to the device I/O pins. Only digital signals are included in the selections. All analog inputs and outputs remain fixed to their assigned pins. Input and output selections are independent as shown in the simplified block diagram Figure 12-1.

12.1 PPS Inputs

Each peripheral has a PPS register with which the inputs to the peripheral are selected. Inputs include the device pins.

Although every peripheral has its own PPS input selection register, the selections are identical for every peripheral as shown in Register 12-1.

Note:	The notation “xxx” in the register name is a place holder for the peripheral identifier. For example, CLC1PPS.

12.2 PPS Outputs

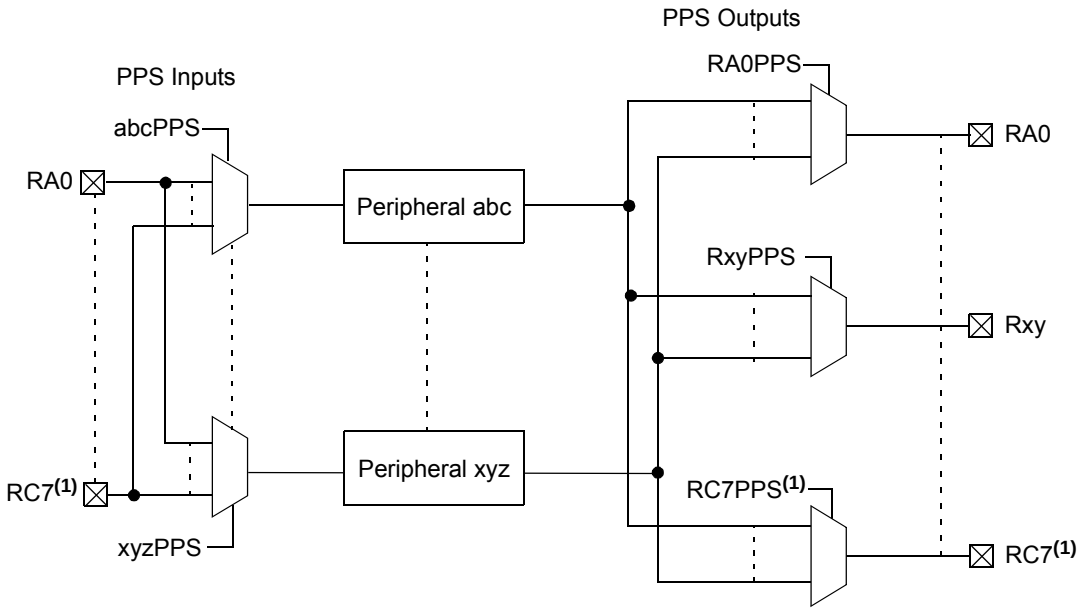
Each I/O pin has a PPS register with which the pin output source is selected. With few exceptions, the port TRIS control associated with that pin retains control over the pin output driver. Peripherals that control the pin output driver as part of the peripheral operation will override the TRIS control as needed. These peripherals include:

- EUSART1 (synchronous operation)
- MSSP (I^2C)

Although every pin has its own PPS peripheral selection register, the selections are identical for every pin as shown in Register 12-2.

Note:	The notation “Rxy” is a place holder for the pin identifier. For example, RA0PPS.

FIGURE 12-1: SIMPLIFIED PPS BLOCK DIAGRAM



Note 1: RB<7:4> and RC<7:6> are available on PIC16(L)F18345 only.

12.3 Bidirectional Pins

PPS selections for peripherals with bidirectional signals on a single pin must be made so that the PPS input and PPS output select the same pin. Peripherals that have bidirectional signals include:

- EUSART1 (synchronous operation)
- MSSP (I^2C)

Note:	The I^2C™ default input pins are I^2C and SMBus compatible and are the only pins on the PIC16(L)F18325 with this compatibility. For the PIC16(L)F18345, in addition to the default pins as described above, RC0, RC1, RC4, and RC5 are also I^2C and SMBus compatible. Clock and data signals can be routed to any pin, however pins without I^2C compatibility will operate at standard TTL/ST logic levels as selected by the INVLV register.

12.4 PPS Lock

The PPS includes a mode in which all input and output selections can be locked to prevent inadvertent changes. PPS selections are locked by setting the PPSLOCKED bit of the PPSLOCK register. Setting and clearing this bit requires a special sequence as an extra precaution against inadvertent changes. Examples of setting and clearing the PPSLOCKED bit are shown in Example 12-1.

EXAMPLE 12-1: PPS LOCK/UNLOCK SEQUENCE

```
; suspend interrupts
    bcf     INTCON,GIE
;   BANKSEL PPSLOCK     ; set bank
; required sequence, next 5 instructions
    movlw   0x55
    movwf   PPSLOCK
    movlw   0xAA
    movwf   PPSLOCK
; Set PPSLOCKED bit to disable writes or
; Clear PPSLOCKED bit to enable writes
    bsf     PPSLOCK,PPSLOCKED
; restore interrupts
    bsf     INTCON,GIE
```

12.5 PPS Permanent Lock

The PPS can be permanently locked by setting the PPS1WAY Configuration bit. When this bit is set, the PPSLOCKED bit can only be cleared and set one time after a device Reset. This allows for clearing the PPSLOCKED bit so that the input and output selections can be made during initialization. When the PPSLOCKED bit is set after all selections have been made, it will remain set and cannot be cleared until after the next device Reset event.

12.6 Operation During Sleep

PPS input and output selections are unaffected by Sleep.

12.7 Effects of a Reset

A device Power-On-Reset (POR) clears all PPS input and output selections to their default values. All other Resets leave the selections unchanged. Default input selections are shown in pin allocation Table 1 and Table 2.

12.8 Register Definitions: PPS Input Selection

REGISTER 12-1: xxxPPS: PERIPHERAL xxx INPUT SELECTION

U-0	U-0	U-0	R/W-q/u	R/W-q/u	R/W-q/u	R/W-q/u	R/W-q/u
—	—	—	xxxPPS<4:0>				
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 peripheral

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **xxxPPS<4:0>:** Peripheral xxx Input Selection bits

11xxx = Reserved. Do not use.

10111 = Peripheral input is RC7(1)
10110 = Peripheral input is RC6(1)
10101 = Peripheral input is RC5
10100 = Peripheral input is RC4
10011 = Peripheral input is RC3
10010 = Peripheral input is RC2
10001 = Peripheral input is RC1
10000 = Peripheral input is RC0
...
01111 = Peripheral input is RB7(1)
01110 = Peripheral input is RB6(1)
01101 = Peripheral input is RB5(1)
01100 = Peripheral input is RB4(1)
...
0011x = Reserved. Do not use.
00101 = Peripheral input is RA5
00100 = Peripheral input is RA4
00011 = Peripheral input is RA3
00010 = Peripheral input is RA2
00001 = Peripheral input is RA1
00000 = Peripheral input is RA0

Note 1: PIC16(L)F18345 only.

REGISTER 12-2: RxyPPS: PIN Rxy OUTPUT SOURCE SELECTION REGISTER

U-0	U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	RxyPPS<4:0>				
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 **Unimplemented:** Read as '0'

bit 4-0 **RxyPPS<4:0>:** Pin Rxy Output Source Selection bits

11111 = Rxy source is DSM
11110 = Rxy source is CLKR
11101 = Rxy source is NCO1
11100 = Rxy source is TMR0
11011 = Rxy source is SDO2/SDA2(1)
11010 = Rxy source is SCK2/SCL2(1)
11001 = Rxy source is SDO1/SDA1
11000 = Rxy source is SCK1/SCL1(1)
10111 = Rxy source is C2
10110 = Rxy source is C1
10101 = Rxy source is DT(1)
10100 = Rxy source is TX/CK(1)
10011 = Rxy source is CWG2D(1)
10010 = Rxy source is CWG2C(1)
10001 = Rxy source is CWG2B(1)
10000 = Rxy source is CWG2A(1)
01111 = Rxy source is CCP4
01110 = Rxy source is CCP3
01101 = Rxy source is CCP2
01100 = Rxy source is CCP1
01011 = Rxy source is CWG1D(1)
01010 = Rxy source is CWG1C(1)
01001 = Rxy source is CWG1B(1)
01000 = Rxy source is CWG1A(1)
00111 = Rxy source is CLC4OUT
00110 = Rxy source is CLC3OUT
00101 = Rxy source is CLC2OUT
00100 = Rxy source is CLC1OUT
00011 = Rxy source is PWM6
00010 = Rxy source is PWM5
00001 = Reserved
00000 = Rxy source is LATxy

Note 1: TRIS control is overridden by the peripheral as required.

REGISTER 12-3: PPSLOCK: PPS LOCK REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	PPSLOCKED
bit 7							bit 0

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

bit 7-1 **Unimplemented:** Read as '0'

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

TABLE 12-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE PPS MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	162
INTPPS	—	—	—	INTPPS<4:0>					160
T0CKIPPS	—	—	—	T0CKIPPS<4:0>					160
T1CKIPPS	—	—	—	T1CKIPPS<4:0>					160
T1GPPS	—	—	—	T1GPPS<4:0>					160
T3CKIPPS	—	—	—	T3CKIPPS<4:0>					160
T3GPPS	—	—	—	T3GPPS<4:0>					160
T5CKIPPS	—	—	—	T5CKIPPS<4:0>					160
T5GPPS	—	—	—	T5GPPS<4:0>					160
CCP1PPS	—	—	—	CCP1PPS<4:0>					160
CCP2PPS	—	—	—	CCP2PPS<4:0>					160
CCP3PPS	—	—	—	CCP3PPS<4:0>					160
CCP4PPS	—	—	—	CCP4PPS<4:0>					160
CWG1PPS	—	—	—	CWG1PPS<4:0>					160
CWG2PPS	—	—	—	CWG2PPS<4:0>					160
MDCIN1PPS	—	—	—	MDCIN1PPS<4:0>					160
MDCIN2PPS	—	—	—	MDCIN2PPS<4:0>					160
MDMINPPS	—	—	—	MDMINPPS<4:0>					160
SSP1CLKPPS	—	—	—	SSP1CLKPPS<4:0>					160
SSP1DATPPS	—	—	—	SSP1DATPPS<4:0>					160
SSP1SSPPS	—	—	—	SSP1SSPPS<4:0>					160
SSP2CLKPPS	—	—	—	SSP2CLKPPS<4:0>					160
SSP2DATPPS	—	—	—	SSP2DATPPS<4:0>					160
SSP2SSPPS	—	—	—	SSP2SSPPS<4:0>					160

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the PPS module.
Note 1: PIC16(L)F18345 only.

TABLE 12-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE PPS MODULE (CONTINUED)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
RXPPS	—	—	—	RXPPS<4:0>					161
TXPPS	—	—	—	TXPPS<4:0>					160
CLCIN0PPS	—	—	—	CLCIN0PPS<4:0>					160
CLCIN1PPS	—	—	—	CLCIN1PPS<4:0>					160
CLCIN2PPS	—	—	—	CLCIN2PPS<4:0>					160
CLCIN3PPS	—	—	—	CLCIN3PPS<4:0>					160
RA0PPS	—	—	—	RA0PPS<4:0>					161
RA1PPS	—	—	—	RA1PPS<4:0>					161
RA2PPS	—	—	—	RA2PPS<4:0>					161
RA4PPS	—	—	—	RA4PPS<4:0>					161
RA5PPS	—	—	—	RA5PPS<4:0>					161
RB4PPS(1)	—	—	—	RB4PPS<4:0>					161
RB5PPS(1)	—	—	—	RB5PPS<4:0>					161
RB6PPS(1)	—	—	—	RB6PPS<4:0>					161
RB7PPS(1)	—	—	—	RB7PPS<4:0>					161
RC0PPS	—	—	—	RC0PPS<4:0>					161
RC1PPS	—	—	—	RC1PPS<4:0>					161
RC2PPS	—	—	—	RC2PPS<4:0>					161
RC3PPS	—	—	—	RC3PPS<4:0>					161
RC4PPS	—	—	—	RC4PPS<4:0>					161
RC5PPS	—	—	—	RC5PPS<4:0>					161
RC6PPS(1)	—	—	—	RC6PPS<4:0>					161
RC7PPS(1)	—	—	—	RC7PPS<4:0>					161

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the PPS module.

Note 1: PIC16(L)F18345 only.

13.0 PERIPHERAL MODULE DISABLE

The PIC16(L)F18325/18345 provides the ability to disable selected modules, placing them into the lowest possible power mode.

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

13.1 Disabling a Module

Disabling a module has the following effects:

- All clock and control inputs to the module are suspended; there are no logic transitions, and the module will not function.
- The module is held in Reset.
- Any SFRs become “unimplemented”
  - Writing is disabled
  - Reading returns 00h
- Module outputs are disabled; I/O goes to the next module according to pin priority

13.2 Enabling a Module

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

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

13.3 Disabling a Module

When a module is disabled, any and all associated input selection registers (ISMs) are also disabled.

13.4 System Clock Disable

Setting SYSCMD (PMD0, Register 13-1) disables the system clock (FOSC) distribution network to the peripherals. Not all peripherals make use of SYSCLK, so not all peripherals are affected. Refer to the specific peripheral description to see if it will be affected by this bit.

REGISTER 13-1: PMD0: PMD CONTROL REGISTER 0

R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
SYSCMD	FVRMD	—	—	—	NVMMD	CLKRMD	IOCMD
7							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 **SYSCMD:** Disable Peripheral System Clock Network bit
See description in **Section 13.4 “System Clock Disable”**.
1 = System Clock network disabled (a.k.a. FOSC)
0 = System Clock network enabled

bit 6 **FVRMD:** Disable Fixed Voltage Reference FVR bit
1 = FVR module disabled
0 = FVR module enabled

bit 5-3 **Unimplemented:** Read as ‘0’

bit 2 **NVMMD:** NVM Module Disable bit(1)
1 = Data EEPROM (a.k.a. user memory, EEPROM) reading and writing is disabled; NVMCON registers cannot be written; FSR access to EEPROM returns zero.
0 = NVM module enabled

bit 1 **CLKRMD:** Disable Clock Reference CLKR bit
1 = CLKR module disabled
0 = CLKR module enabled

bit 0 **IOCMD:** Disable Interrupt-on-Change bit, All Ports
1 = IOC module(s) disabled
0 = IOC module(s) enabled

Note 1: When enabling NVM, a delay of up to 1 µs may be required before accessing data.

REGISTER 13-2: PMD1: PMD CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NCOMD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD
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 **NCOMD:** Disable Numerically Control Oscillator bit
1 = NCO1 module disabled
0 = NCO1 module enabled

bit 6 **TMR6MD:** Disable Timer TMR6 bit
1 = TMR6 module disabled
0 = TMR6 module enabled

bit 5 **TMR5MD:** Disable Timer TMR5 bit
1 = TMR5 module disabled
0 = TMR5 module enabled

bit 4 **TMR4MD:** Disable Timer TMR4 bit
1 = TMR4 module disabled
0 = TMR4 module enabled

bit 3 **TMR3MD:** Disable Timer TMR3 bit
1 = TMR3 module disabled
0 = TMR3 module enabled

bit 2 **TMR2MD:** Disable Timer TMR2 bit
1 = TMR2 module disabled
0 = TMR2 module enabled

bit 1 **TMR1MD:** Disable Timer TMR1 bit
1 = TMR1 module disabled
0 = TMR1 module enabled

bit 0 **TMR0MD:** Disable Timer TMR0 bit
1 = TMR0 module disabled
0 = TMR0 module enabled

REGISTER 13-3: PMD2: PMD CONTROL REGISTER 2

U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	U-0
—	DACMD	ADCMD	—	—	CMP2MD	CMP1MD	—
bit 7							bit 0

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

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

bit 6 **DACMD:** Disable DAC bit
1 = DAC module disabled
0 = DAC module enabled

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

bit 4-3 **Unimplemented:** Read as '0'

bit 2 **CMP2MD:** Disable Comparator C2 bit
1 = C2 module disabled
0 = C2 module enabled

bit 1 **CMP1MD:** Disable Comparator C1 bit
1 = C1 module disabled
0 = C1 module enabled

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

REGISTER 13-4: PMD3: PMD CONTROL REGISTER 3

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
CWG2MD	CWG1MD	PWM6MD	PWM5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD
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 **CWG2MD:** Disable CWG2 bit
1 = CWG2 module disabled
0 = CWG2 module enabled

bit 6 **CWG1MD:** Disable CWG1 bit
1 = CWG1 module disabled
0 = CWG1 module enabled

bit 5 **PWM6MD:** Disable Pulse-Width Modulator PWM6 bit
1 = PWM6 module disabled
0 = PWM6 module enabled

bit 4 **PWM5MD:** Disable Pulse-Width Modulator PWM5 bit
1 = PWM5 module disabled
0 = PWM5 module enabled

bit 3 **CCP4MD:** Disable Pulse-Width Modulator CCP4 bit
1 = CCP4 module disabled
0 = CCP4 module enabled

bit 2 **CCP3MD:** Disable Pulse-Width Modulator CCP3 bit
1 = CCP3 module disabled
0 = CCP3 module enabled

bit 1 **CCP2MD:** Disable Pulse-Width Modulator CCP2 bit
1 = CCP2 module disabled
0 = CCP2 module enabled

bit 0 **CCP1MD:** Disable Pulse-Width Modulator CCP1 bit
1 = CCP1 module disabled
0 = CCP1 module enabled

REGISTER 13-5: PMD4: PMD CONTROL REGISTER 4

U-0	U-0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	U-0
—	—	UART1MD	—	—	MSSP2MD	MSSP1MD	—
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 **UART1MD:** Disable EUSART1 bit
1 = EUSART1 module disabled
0 = EUSART1 module enabled

bit 4-3 **Unimplemented:** Read as '0'

bit 2 **MSSP2MD:** Disable MSSP2 bit
1 = MSSP2 module disabled
0 = MSSP2 module enabled

bit 1 **MSSP1MD:** Disable MSSP1 bit
1 = MSSP1 module disabled
0 = MSSP1 module enabled

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

REGISTER 13-6: PMD5: PMD CONTROL REGISTER 5

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	CLC4MD	CLC3MD	CLC2MD	CLC1MD	DSMMD
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 7-5 **Unimplemented:** Read as ‘0’

bit 4 **CLC4MD:** Disable CLC4 bit
1 = CLC4 module disabled
0 = CLC4 module enabled

bit 3 **CLC3MD:** Disable CLC3 bit
1 = CLC3 module disabled
0 = CLC3 module enabled

bit 2 **CLC2MD:** Disable CLC2 bit
1 = CLC2 module disabled
0 = CLC2 module enabled

bit 1 **CLC1MD:** Disable CLC1 bit
1 = CLC1 module disabled
0 = CLC1 module enabled

bit 0 **DSMMD:** Disable Data Signal Modulator bit
1 = DSM module disabled
0 = DSM module enabled

14.0 INTERRUPT-ON-CHANGE

All pins on all ports 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 pin, or combination of 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 14-1 is a block diagram of the IOC module.

14.1 Enabling the Module

To allow individual pins to generate an interrupt, the IOCIE bit of the PIE0 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.

14.2 Individual Pin Configuration

For each pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated bit of the IOCxP register is set. To enable a pin to detect a falling edge, the associated bit of the IOCxN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting the associated bits in both of the IOCxP and IOCxN registers.

14.3 Interrupt Flags

The bits located in the IOCxF registers are status flags that correspond to the interrupt-on-change pins of each port. 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 PIR0 register reflects the status of all IOCxF bits.

14.4 Clearing Interrupt Flags

The individual status flags, (IOCxF register 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 14-1: CLEARING INTERRUPT FLAGS (PORTA EXAMPLE)

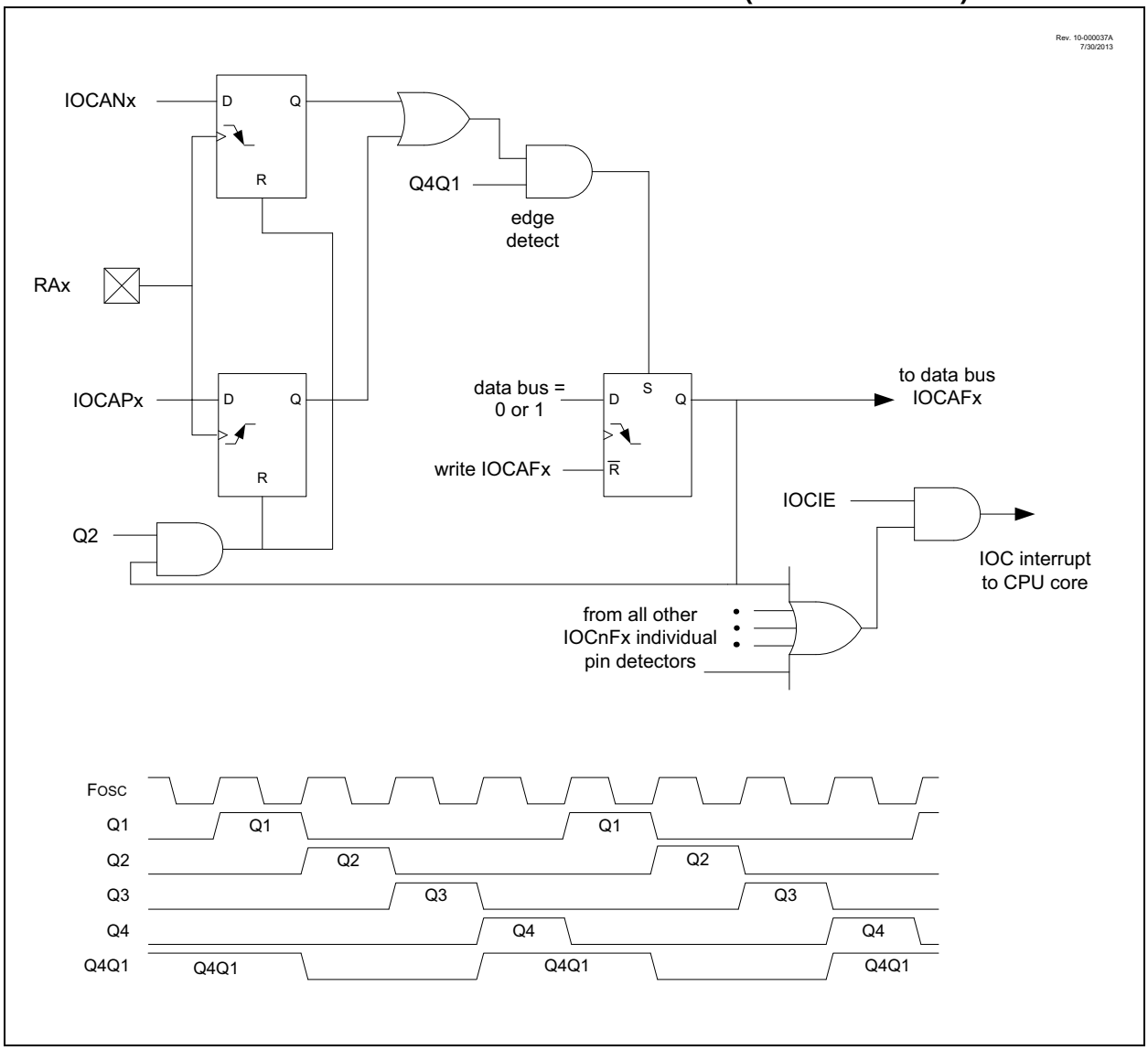
```
MOVLW  0xff
XORWF  IOCAF, W
ANDWF  IOCAF, F
```

14.5 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 affected IOCxF register will be updated prior to the first instruction executed out of Sleep.

FIGURE 14-1: INTERRUPT-ON-CHANGE BLOCK DIAGRAM (PORTA EXAMPLE)



14.6 Register Definitions: Interrupt-on-Change Control

REGISTER 14-1: IOCAP: INTERRUPT-ON-CHANGE PORTA POSITIVE EDGE REGISTER

U-0	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
—	—	IOCAP5	IOCAP4	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 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **IOCAP<5:0>:** Interrupt-on-Change PORTA Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin

REGISTER 14-2: IOCAN: INTERRUPT-ON-CHANGE PORTA NEGATIVE EDGE REGISTER

U-0	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
—	—	IOCAN5	IOCAN4	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 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **IOCAN<5:0>:** Interrupt-on-Change PORTA Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin

REGISTER 14-3: IOCAF: INTERRUPT-ON-CHANGE PORTA FLAG REGISTER

U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	IOCAF5	IOCAF4	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 7-6 **Unimplemented:** Read as '0'

bit 5-0 **IOCAF<5:0>:** Interrupt-on-Change PORTA Flag bits
1 = An enabled 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.
0 = No change was detected, or the user cleared the detected change.

REGISTER 14-4: IOCBP: INTERRUPT-ON-CHANGE PORTB POSITIVE EDGE REGISTER(1)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—
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 **IOCBP<7:4>:** Interrupt-on-Change PORTB Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin

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

Note 1: PIC16(L)F18345 only.

REGISTER 14-5: IOCBN: INTERRUPT-ON-CHANGE PORTB NEGATIVE EDGE REGISTER(1)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—
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 **IOCBN<7:4>:** Interrupt-on-Change PORTB Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin

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

Note 1: PIC16(L)F18345 only.

REGISTER 14-6: IOCBF: INTERRUPT-ON-CHANGE PORTB FLAG REGISTER(1)

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	U-0	U-0	U-0	U-0
IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—
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 **IOCBF<7:4>:** Interrupt-on-Change PORTB Flag bits
1 = An enabled change was detected on the associated pin
Set when IOCBPx = 1 and a rising edge was detected on RBx, or when IOCBNx = 1 and a falling edge was detected on RBx.
0 = No change was detected, or the user cleared the detected change.

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

Note 1: PIC16(L)F18345 only.

REGISTER 14-7: IOCCP: INTERRUPT-ON-CHANGE PORTC POSITIVE EDGE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCCP7(1)	IOCCP6(1)	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0
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 **IOCCP<7:6>:** Interrupt-on-Change PORTC Positive Edge Enable bits(1)
1 = Interrupt-on-Change enabled on the pin for a positive going edge. IOCCFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin

bit 5-0 **IOCCP<5:0>:** Interrupt-on-Change PORTC Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive going edge. IOCCFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin

Note 1: PIC16(L)F18345 only.

REGISTER 14-8: IOCCN: INTERRUPT-ON-CHANGE PORTC NEGATIVE EDGE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCCN7(1)	IOCCN6(1)	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0
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 **IOCCN<7:6>:** Interrupt-on-Change PORTC Negative Edge Enable bits(1)
1 = Interrupt-on-Change enabled on the pin for a negative going edge. IOCCFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin

bit 5-0 **IOCCN<5:0>:** Interrupt-on-Change PORTC Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative going edge. IOCCFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin

Note 1: PIC16(L)F18345 only.

REGISTER 14-9: IOCCF: INTERRUPT-ON-CHANGE PORTC FLAG REGISTER

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
IOCCF7(1)	IOCCF6(1)	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0
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 7-6 **IOCCF<7:6>:** Interrupt-on-Change PORTC Flag bits
- 1 = An enabled change was detected on the associated pin. Set when IOCCPx = 1 and a rising edge was detected on RCx, or when IOCCNx = 1 and a falling edge was detected on RCx.
- 0 = No change was detected, or the user cleared the detected change.

bit 5-0 **IOCCF<5:0>:** Interrupt-on-Change PORTC Flag bits
- 1 = An enabled change was detected on the associated pin. Set when IOCCPx = 1 and a rising edge was detected on RCx, or when IOCCNx = 1 and a falling edge was detected on RCx.
- 0 = No change was detected, or the user cleared the detected change.

Note 1: PIC16(L)F18345 only.

TABLE 14-1: 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
ANSELA	—	—	ANSA4	ANSA4	—	ANSA2	ANSA1	ANSA0	142
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	141
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	99
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	172
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	172
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	173
IOCBP(1)	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	173
IOCBN(1)	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	174
IOCBF(1)	IOCBF7	IOCBF6	IOCBF5	IIOCBF4	—	—	—	—	174
IOCCP	IOCCP7(1)	IOCCP6(1)	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	175
IOCCN	IOCCN7(1)	IOCCN6(1)	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	175
IOCCF	IOCCF7(1)	IOCCF6(1)	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	176

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by interrupt-on-change.

Note 1: PIC16(L)F18345 only.

2: Unimplemented, read as '1'.

15.0 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
- ADC positive reference
- Comparator positive input
- Digital-to-Analog Converter (DAC)

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

Note:	Fixed Voltage Reference output cannot exceed VDD.

15.1 Independent Gain Amplifiers

The output of the FVR, which is supplied to the ADC, Comparators and DAC, is routed through two independent programmable gain amplifiers. Each amplifier can be programmed for a gain of 1x, 2x or 4x, to produce the three possible voltage levels.

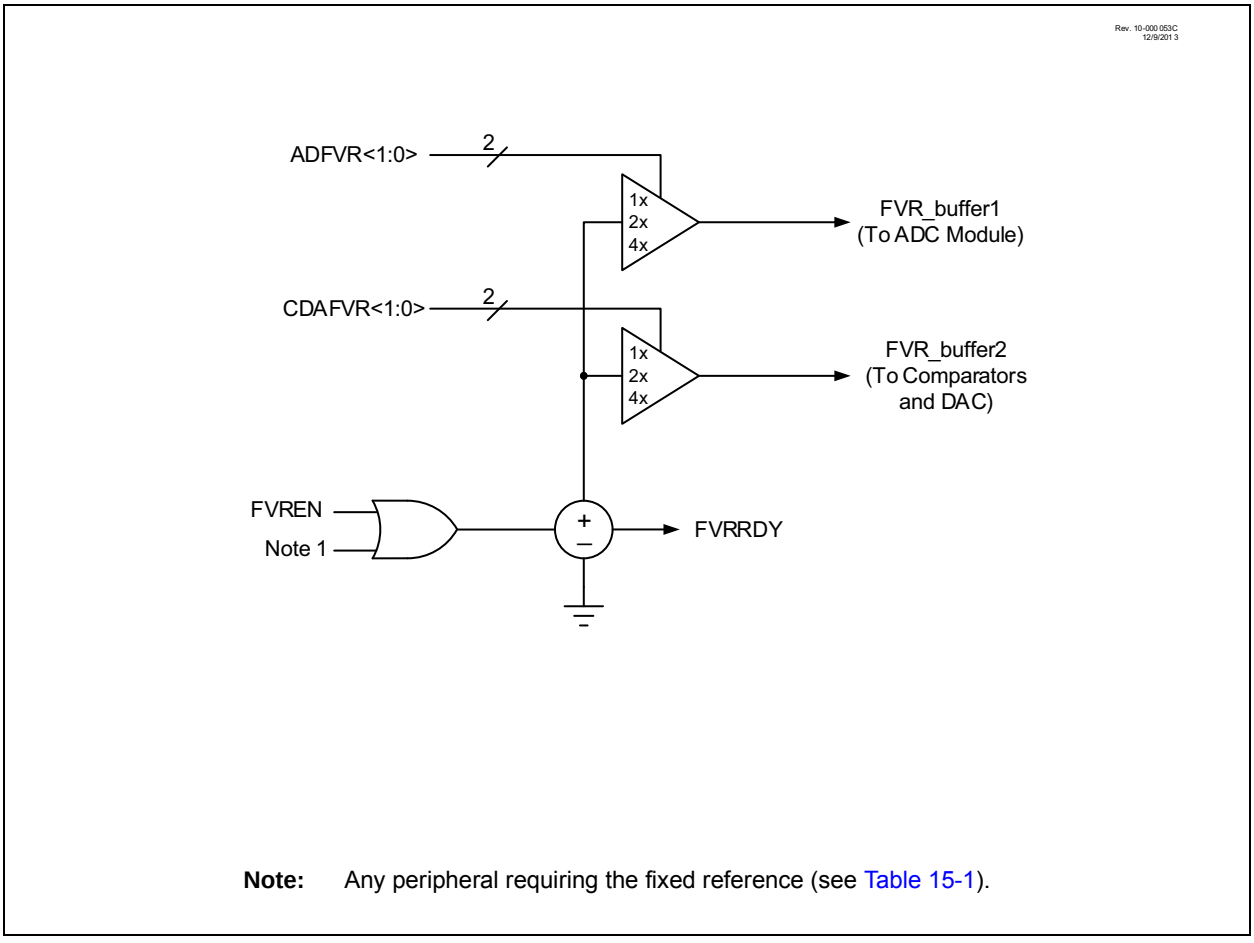
The ADFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the ADC module. Reference **Section 21.0 "Analog-to-Digital Converter (ADC) Module"** for additional information.

The CDAFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the DAC and comparator module. Reference **Section 23.0 "5-bit Digital-to-Analog Converter (DAC1) Module"** and **Section 17.0 "Comparator Module"** for additional information.

15.2 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.

FIGURE 15-1: VOLTAGE REFERENCE BLOCK DIAGRAM



Note: Any peripheral requiring the fixed reference (see Table 15-1).

15.3 Register Definitions: FVR Control

REGISTER 15-1: FVRCON: FIXED VOLTAGE REFERENCE CONTROL REGISTER

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
FVREN	FVRRDY(1)	TSEN(3)	TSRNG(3)	CDAFVR<1:0>		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 = VOUT = VDD - 4VT (High Range)
0 = VOUT = VDD - 2VT (Low Range)

bit 3-2 **CDAFVR<1:0>:** Comparator FVR Buffer Gain Selection bits
11 = Comparator FVR Buffer Gain is 4x, (4.096V)(2)
10 = Comparator FVR Buffer Gain is 2x, (2.048V)(2)
01 = Comparator FVR Buffer Gain is 1x, (1.024V)
00 = Comparator FVR Buffer is off

bit 1-0 **ADFVR<1:0>:** ADC FVR Buffer Gain Selection bit
11 = ADC FVR Buffer Gain is 4x, (4.096V)(2)
10 = ADC FVR Buffer Gain is 2x, (2.048V)(2)
01 = ADC FVR Buffer Gain is 1x, (1.024V)
00 = ADC FVR Buffer is off

Note 1: FVRRDY is always '1'.
2: Fixed Voltage Reference output cannot exceed VDD.
3: See **Section 16.0 "Temperature Indicator Module"** for additional information.

TABLE 15-1: 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	CDAFVR<1:0>		ADFVR<1:0>		178
ADCON0	CHS<5:0>						GO/$\overline{\text{DONE}}$	ADON	242
ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		243
CMxCON1	CxINTP	CxINTN	CxPCH<2:0>			CxNCH<2:0>			189
DAC1CON0	DAC1EN	—	DAC1OE	—	DAC1PPS<1:0>		—	DAC1NSS	261

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

16.0 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 -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.

16.1 Circuit Operation

Figure 16-1 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 16-1 describes the output characteristics of the temperature indicator.

EQUATION 16-1: VOUT RANGES

High Range: $V_{OUT} = V_{DD} - 4V_T$

Low Range: $V_{OUT} = V_{DD} - 2V_T$

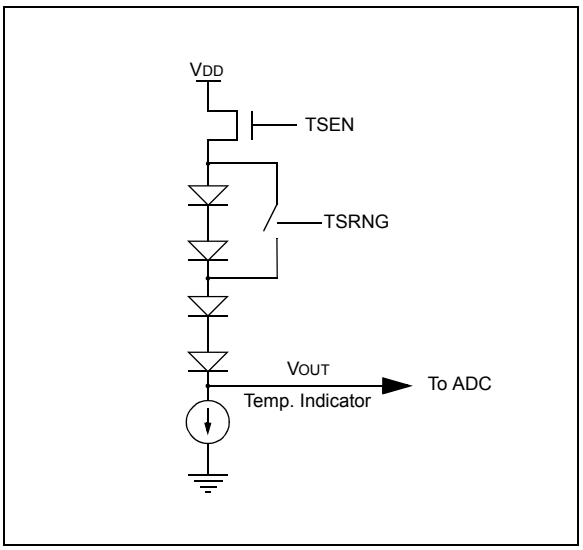
The temperature sense circuit is integrated with the Fixed Voltage Reference (FVR) module. See **Section 15.0 "Fixed Voltage Reference (FVR)"** for more information.

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 FVRCON 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 16-1: TEMPERATURE CIRCUIT DIAGRAM



16.2 Minimum Operating VDD

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 16-1 shows the recommended minimum VDD vs. range setting.

TABLE 16-1: RECOMMENDED VDD VS. RANGE

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

16.3 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 21.0 "Analog-to-Digital Converter (ADC) Module"** for detailed information.

16.4 ADC Acquisition Time

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

TABLE 16-2: 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	CDAFVR<1:0>		ADFVR<1:0>		178

Legend: Shaded cells are unused by the temperature indicator module.

17.0 COMPARATOR MODULE

Comparators are used to interface analog circuits to a digital circuit by comparing two analog voltages and providing a digital indication of their relative magnitudes. Comparators are very useful mixed signal building blocks because they provide analog functionality independent of program execution. The analog comparator module includes the following features:

- Programmable input selection
- Programmable output polarity
- Rising/falling output edge interrupts
- Wake-up from Sleep
- Programmable Speed/Power optimization
- CWG Auto-shutdown source
- Selectable voltage reference

17.1 Comparator Overview

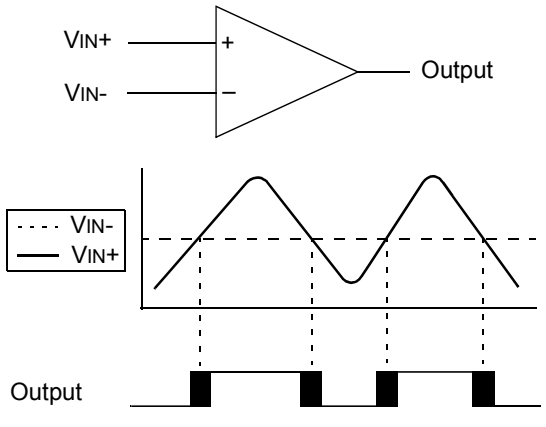
A single comparator is shown in Figure 17-1 along with the relationship between the analog input levels and the digital output. When the analog voltage at VIN+ is less than the analog voltage at VIN-, the output of the comparator is a digital low level. When the analog voltage at VIN+ is greater than the analog voltage at VIN-, the output of the comparator is a digital high level.

The comparators available for this device are located in Table 17-1.

TABLE 17-1: AVAILABLE COMPARATORS

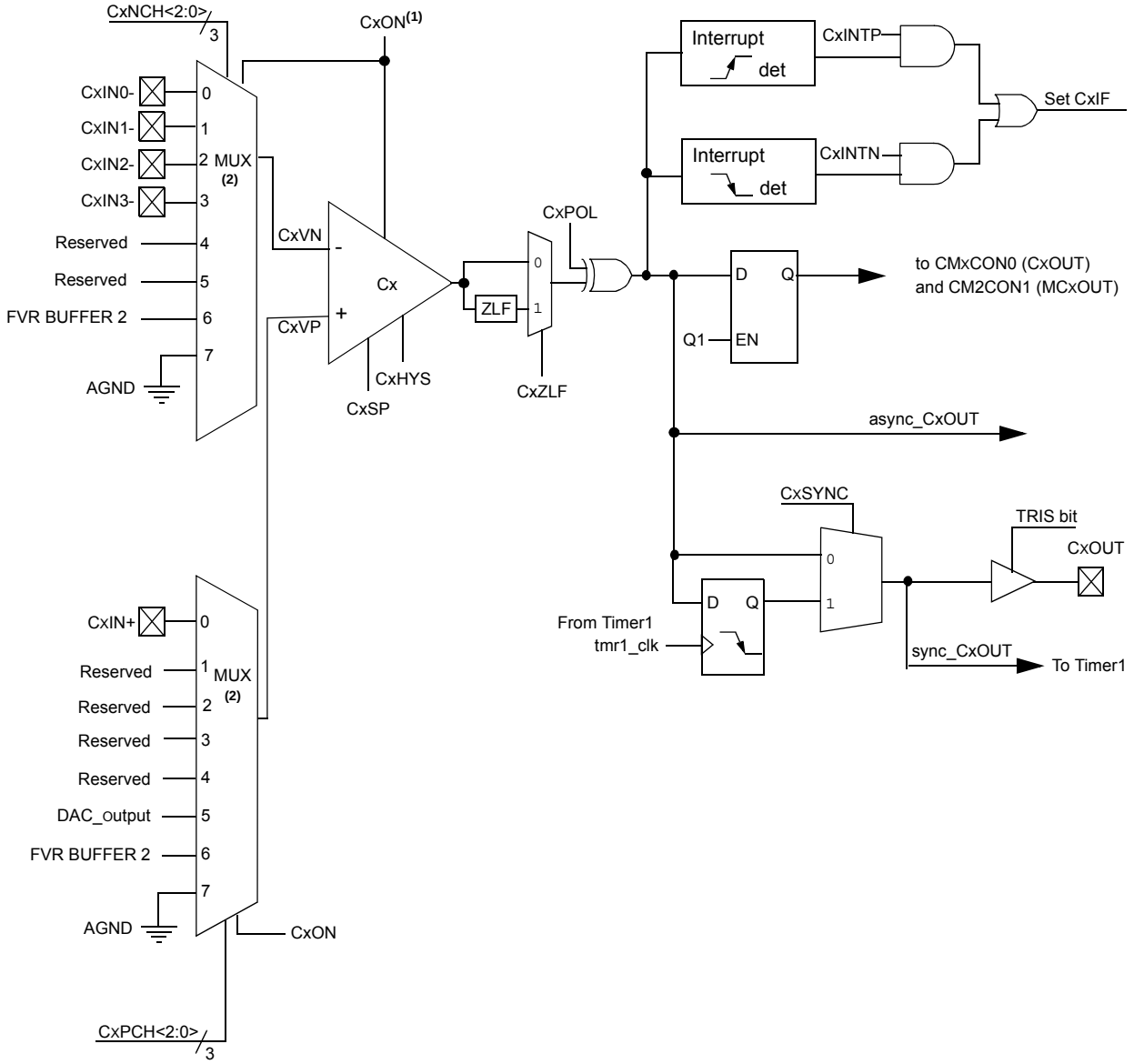
Device	C1	C2
PIC16(L)F18325	•	•
PIC16(L)F18345	•	•

FIGURE 17-1: SINGLE COMPARATOR



Note: The black areas of the output of the comparator represents the uncertainty due to input offsets and response time.

FIGURE 17-2: COMPARATOR MODULE SIMPLIFIED BLOCK DIAGRAM



Note 1: When CxON = 0, the comparator will produce a '0' at the output.
2: When CxON = 0, all multiplexer inputs are disconnected.

17.2 Comparator Control

Each comparator has two control registers: CMxCON0 and CMxCON1.

The CMxCON0 register (see Register 17-1) contains Control and Status bits for the following:

- Enable
- Output
- Output polarity
- Speed/Power selection
- Hysteresis enable
- Timer1 output synchronization

The CMxCON1 register (see Register 17-2) contains Control bits for the following:

- Interrupt on positive/negative edge enables
- Positive input channel selection
- Negative input channel selection

17.2.1 COMPARATOR ENABLE

Setting the CxON bit of the CMxCON0 register enables the comparator for operation. Clearing the CxON bit disables the comparator resulting in minimum current consumption.

17.2.2 COMPARATOR OUTPUT

The output of the comparator can be monitored by reading either the CxOUT bit of the CMxCON0 register or the MCxOUT bit of the CMOUT register.

The comparator output can also be routed to an external pin through the RxyPPS register (Register 12-2). The corresponding TRIS bit must be clear to enable the pin as an output.

Note 1:	The internal output of the comparator is latched with each instruction cycle. Unless otherwise specified, external outputs are not latched.

17.2.3 COMPARATOR OUTPUT POLARITY

Inverting the output of the comparator is functionally equivalent to swapping the comparator inputs. The polarity of the comparator output can be inverted by setting the CxPOL bit of the CMxCON0 register. Clearing the CxPOL bit results in a non-inverted output.

Table 17-2 shows the output state versus input conditions, including polarity control.

TABLE 17-2: COMPARATOR OUTPUT STATE VS. INPUT CONDITIONS

Input Condition	CxPOL	CxOUT
CxVN > CxVP	0	0
CxVN < CxVP	0	1
CxVN > CxVP	1	1
CxVN < CxVP	1	0

17.3 Comparator Hysteresis

A selectable amount of separation voltage can be added to the input pins of each comparator to provide a hysteresis function to the overall operation. Hysteresis is enabled by setting the CxHYS bit of the CMxCON0 register.

See Comparator Specifications in Table 34-14 for more information.

17.4 Timer1 Gate Operation

The output resulting from a comparator operation can be used as a source for gate control of Timer1. See **Section 26.6 "Timer1 Gate"** for more information. This feature is useful for timing the duration or interval of an analog event.

It is recommended that the comparator output be synchronized to Timer1. This ensures that Timer1 does not increment while a change in the comparator is occurring.

17.4.1 COMPARATOR OUTPUT SYNCHRONIZATION

The output from a comparator can be synchronized with Timer1 by setting the CxSYNC bit of the CMxCON0 register.

Once enabled, the comparator output is latched on the falling edge of the Timer1 source clock. If a prescaler is used with Timer1, the comparator output is latched after the prescaling function. To prevent a race condition, the comparator output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. See the Comparator Block Diagram (Figure 17-2) and the Timer1 Block Diagram (Figure 26-1) for more information.

17.5 Comparator Interrupt

An interrupt can be generated upon a change in the output value of the comparator for each comparator, a rising edge detector and a falling edge detector are present.

When either edge detector is triggered and its associated enable bit is set (CxINTP and/or CxINTN bits of the CMxCON1 register), the Corresponding Interrupt Flag bit (CxIF bit of the PIR2 register) will be set.

To enable the interrupt, you must set the following bits:

- CxON, CxPOL and CxSP bits of the CMxCON0 register
- CxIE bit of the PIE2 register
- CxINTP bit of the CMxCON1 register (for a rising edge detection)
- CxINTN bit of the CMxCON1 register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

The associated interrupt flag bit, CxIF bit of the PIR2 register, must be cleared in software. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

Note:	Although a comparator is disabled, an interrupt can be generated by changing the output polarity with the CxPOL bit of the CMxCON0 register, or by switching the comparator on or off with the CxON bit of the CMxCON0 register.

17.6 Comparator Positive Input Selection

Configuring the CxPCH<2:0> bits of the CMxCON1 register directs an internal voltage reference or an analog pin to the non-inverting input of the comparator:

- CxIN0+ analog pin
- DAC output
- FVR (Fixed Voltage Reference)
- VSS (Ground)

See **Section 15.0 “Fixed Voltage Reference (FVR)”** for more information on the Fixed Voltage Reference module.

See **Section 23.0 “5-bit Digital-to-Analog Converter (DAC1) Module”** for more information on the DAC input signal.

Any time the comparator is disabled (CxON = 0), all comparator inputs are disabled.

17.7 Comparator Negative Input Selection

The CxNCH<2:0> bits of the CMxCON1 register direct an analog input pin and internal reference voltage or analog ground to the inverting input of the comparator:

- CxIN- pin
- FVR (Fixed Voltage Reference)
- Analog Ground

Some inverting input selections share a pin with the operational amplifier output function. Enabling both functions at the same time will direct the operational amplifier output to the comparator inverting input.

Note:	To use CxINy+ and CxINy- pins as analog input, the appropriate bits must be set in the ANSEL register and the corresponding TRIS bits must also be set to disable the output drivers.

17.8 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in Table 34-14 for more details.

17.9 Analog Input Connection Considerations

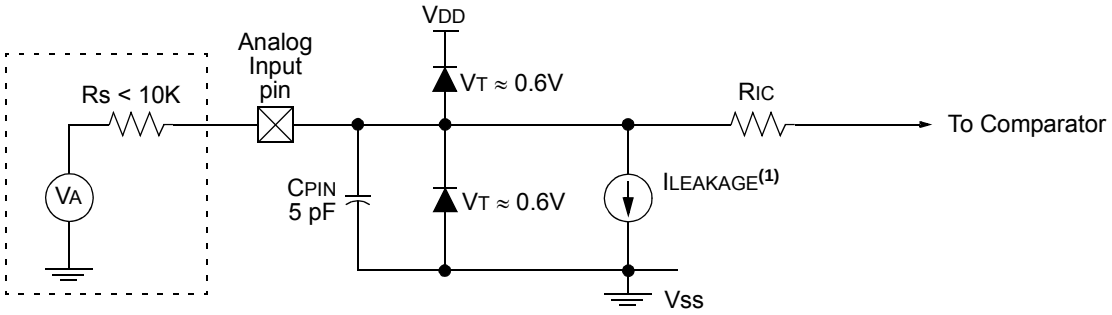
A simplified circuit for an analog input is shown in Figure 17-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to V_{DD} and V_{SS}. The analog input, therefore, must be between V_{SS} and V_{DD}. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

Note 1: When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.

2: Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

FIGURE 17-3: ANALOG INPUT MODEL



Legend: C_{PIN} = Input Capacitance
$I_{LEAKAGE}$ = Leakage Current at the pin due to various junctions
R_{IC} = Interconnect Resistance
R_S = Source Impedance
V_A = Analog Voltage
V_T = Threshold Voltage

Note 1: See I/O Ports in Table 34-4.

17.10 CWG Auto-shutdown Source

The output of the comparator module can be used as an auto-shutdown source for the CWG module. When the output of the comparator is active and the corresponding ASxE is enabled, the CWG operation will be suspended immediately (**Section 19.7.1.2 "External Input Source Shutdown"**).

17.11 Operation in Sleep Mode

The comparator module can operate during Sleep. The comparator clock source is based on the Timer1 clock source. If the Timer1 clock source is either the system clock (FOSC) or the instruction clock (FOSC/4), Timer1 will not operate during Sleep, and synchronized comparator outputs will not operate.

A comparator interrupt will wake the device from Sleep. The CxIE bits of the PIE2 register must be set to enable comparator interrupts.

17.12 Register Definitions: Comparator Control

REGISTER 17-1: CMxCON0: COMPARATOR Cx CONTROL REGISTER 0

R/W-0/0	R-0/0	U-0	R/W-0/0	U-0	R/W-1/1	R/W-0/0	R/W-0/0
CxON	CxOUT	—	CxPOL	—	CxSP	CxHYS	CxSYNC
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 **CxON:** Comparator Enable bit
1 = Comparator is enabled
0 = Comparator is disabled and consumes no active power

bit 6 **CxOUT:** Comparator Output bit
If CxPOL = 1 (inverted polarity):
1 = CxVP < CxVN
0 = CxVP > CxVN
If CxPOL = 0 (non-inverted polarity):
1 = CxVP > CxVN
0 = CxVP < CxVN

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

bit 4 **CxPOL:** Comparator Output Polarity Select bit
1 = Comparator output is inverted
0 = Comparator output is not inverted

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

bit 2 **CxSP:** Comparator Speed/Power Select bit
1 = Comparator operates in Normal-Power, High-Speed mode
0 = Reserved. (do not use)

bit 1 **CxHYS:** Comparator Hysteresis Enable bit
1 = Comparator hysteresis enabled
0 = Comparator hysteresis disabled

bit 0 **CxSYNC:** Comparator Output Synchronous Mode bit
1 = Comparator output to Timer1 and I/O pin is synchronous to changes on Timer1 clock source. Output updated on the falling edge of Timer1 clock source.
0 = Comparator output to Timer1 and I/O pin is asynchronous

REGISTER 17-2: CMxCON1: COMPARATOR Cx CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CxINTP	CxINTN	CxPCH<2:0>			CxNCH<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 **CxINTP:** Comparator Interrupt on Positive Going Edge Enable bits
1 = The CxIF interrupt flag will be set upon a positive going edge of the CxOUT bit
0 = No interrupt flag will be set on a positive going edge of the CxOUT bit

bit 6 **CxINTN:** Comparator Interrupt on Negative Going Edge Enable bits
1 = The CxIF interrupt flag will be set upon a negative going edge of the CxOUT bit
0 = No interrupt flag will be set on a negative going edge of the CxOUT bit

bit 5-3 **CxPCH<2:0>:** Comparator Positive Input Channel Select bits
111 = CxVP connects to AVSS
110 = CxVP connects to FVR Buffer 2
101 = CxVP connects to DAC output
100 = CxVP unconnected
011 = CxVP unconnected
010 = CxVP unconnected
001 = CxVN unconnected
000 = CxVP connects to CxIN0+ pin

bit 2-0 **CxNCH<2:0>:** Comparator Negative Input Channel Select bits
111 = CxVN connects to AVSS
110 = CxVN connects to FVR Buffer 2
101 = CxVN unconnected
100 = CxVN unconnected
011 = CxVN connects to CxIN3- pin
010 = CxVN connects to CxIN2- pin
001 = CxVN connects to CxIN1- pin
000 = CxVN connects to CxIN0- pin

REGISTER 17-3: CMOUT: COMPARATOR OUTPUT REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
—	—	—	—	—	—	MC2OUT	MC1OUT
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 **MC2OUT:** Mirror Copy of C2OUT bit

bit 0 **MC1OUT:** Mirror Copy of C1OUT bit

TABLE 17-3: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
TRISA	—	—	TRISA5	TRISA4	—	TRISA2	TRISA1	TRISA0	141
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
CMxCON0	CxON	CxOUT	—	CxPOL	—	CxSP	CxHYS	CxSYNC	188
CMxCON1	CxINTP	CxINTN	CxPCH<2:0>			CxNCH<2:0>			189
CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	190
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		178
DACCON0	DAC1EN	—	DAC1OE	—	DAC1PSS<1:0>		—	DAC1NSS	261
DACCON1	—	—	—	DAC1R<4:0>					262
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
PIE2	TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BLC2IE	TMR4IE	NCO1IE	101
PIR2	TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BLC2IF	TMR4IF	NCO1IF	106
CMPxPPS	—	—	—	CMPxPPS<4:0>					160
CLCINxPPS	—	—	—	CLCINxPPS<4:0>					160
MDMINPPS	—	—	—	MDMINPPS<4:0>					160
T1GPPS	—	—	—	T1GPPS<4:0>					160
CWGxAS1	—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	216

Legend: — = unimplemented location, read as '0'. Shaded cells are unused by the comparator module.
Note 1: PIC16(L)F18345 only.

18.0 PULSE-WIDTH MODULATION (PWM)

The PWMx modules generate Pulse Width Modulated (PWM) signals of varying frequency and duty cycle.

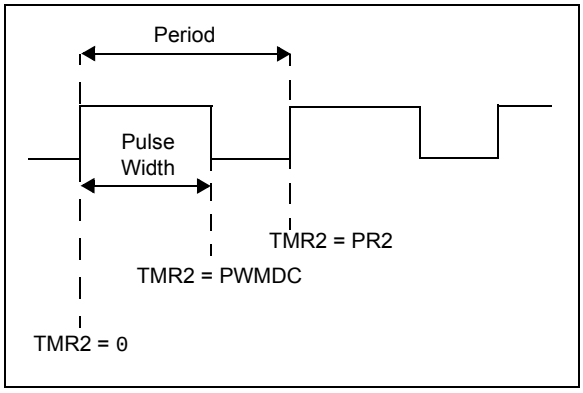
In addition to the CCP modules, the PIC16(L)F18325/18345 devices contain two PWM modules. These modules are essentially the same as the CCP modules without the Capture or Compare functionality.

Pulse-Width Modulation (PWM) is a scheme that provides power to a load by switching quickly between fully on and fully off states. The PWM signal resembles a square wave where the high portion of the signal is considered the 'on' state (pulse width), and the low portion of the signal is considered the 'off' state. The term duty cycle describes the proportion of the 'on' time to the 'off' time and is expressed in percentages, where 0% is fully off and 100% is fully on. A lower duty cycle corresponds to less power applied and a higher duty cycle corresponds to more power applied. The PWM period is defined as the duration of one complete cycle or the total amount of on and off time combined.

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

Figure 18-1 shows a typical waveform of the PWM signal.

FIGURE 18-1: PWM OUTPUT



18.1 Standard PWM Mode

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

- TMR2, TMR4 or TMR6 registers
- PR2, PR4 or PR6 registers
- PWMxCON registers
- PWMxDCH registers
- PWMxDCL registers

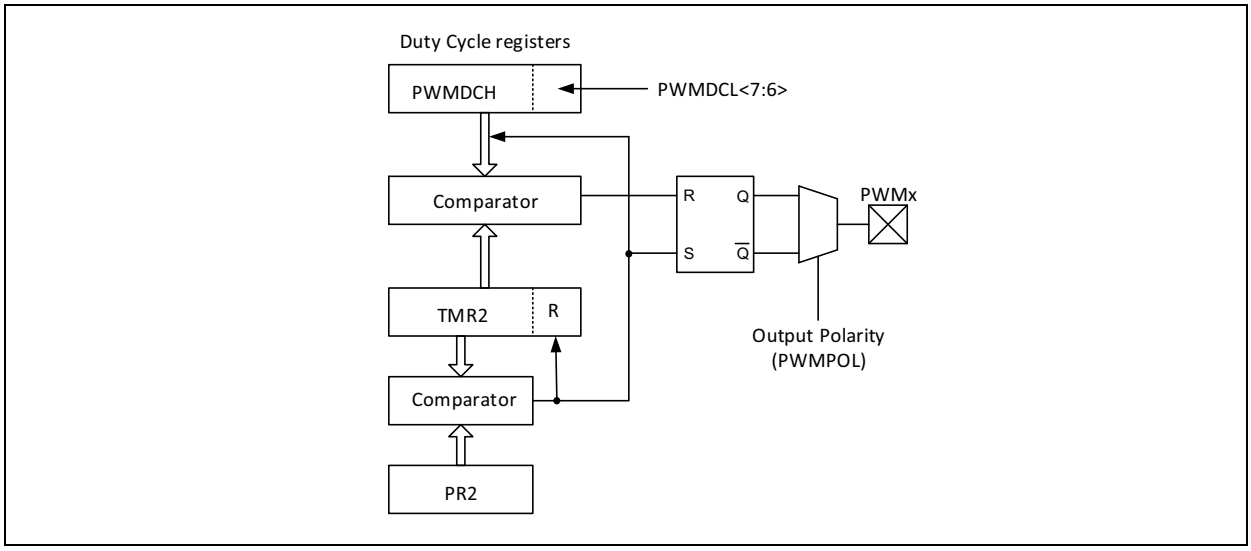
Figure 28-2, "Compare Mode Operation Block Diagram" shows a simplified block diagram of the PWM operation.

If PWMPOL = 0, the default state of the output is '0'. If PWMPOL = 1, the default state is '1'. If PWMEN = '0', the output will be the default state.

Note:	The corresponding TRIS bit must be cleared to enable the PWM output on the PWMx pin

Note:	The formulas and text refer to TMR2 and PR2, for simplicity. The same formulas and text apply to TMR4/6 and PR4/6. The timer sources can be selected in Register 18-4. For additional information on TMR2/4/6, please refer to **Section 27.0 "Timer2/4/6 Module"**

FIGURE 18-2: SIMPLIFIED PWM BLOCK DIAGRAM



18.1.1 PWM PERIOD

Referring to Figure 18-1, the PWM output has a period and a pulse width. The frequency of the PWM is the inverse of the period (1/period).

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

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 PWMx pin is set (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM pulse width is latched from PWMxDC.

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

18.1.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to the PWMxDC register. The PWMxDCH contains the eight MSbs and bits <7:6> of the PWMxDCL register contain the two LSbs.

The PWMDC register is double-buffered and can be updated at any time. This double buffering is essential for glitch-free PWM operation. New values take effect when TMR2 = PR2. Note that PWMDC is left-justified.

The 8-bit timer TMR2 register is concatenated with either the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

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

$$Pulse\ Width = (PWMxDC) \bullet T_{OSC} \bullet \bullet (TMR2\ Prescale\ Value)$$

EQUATION 18-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(PWMxDC)}{4(PR2 + 1)}$$

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

EQUATION 18-4:

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

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

18.1.4 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.

18.1.5 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 6.0, Oscillator Module (with Fail-Safe Clock Monitor)** for additional details.

18.1.6 EFFECTS OF RESET

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

18.1.7 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the module for using the PWMx outputs:

1. Disable the PWMx pin output driver(s) by setting the associated TRIS bit(s).
2. Configure the PWM output polarity by configuring the PWMxPOL bit of the PWMxCON register.
3. Load the PR2 register with the PWM period value, as determined by Equation 18-1.
4. Load the PWMxDCH register and bits <7:6> of the PWMxDCL register with the PWM duty cycle value, as determined by Equation 18-2.
5. Configure and start Timer2:
   - Clear the TMR2IF interrupt flag bit of the PIR1 register.
   - Select the Timer2 prescale value by configuring the T2CKPS bit of the T2CON register.
   - Enable Timer2 by setting the TMR2ON bit of the T2CON register.
6. Wait until the TMR2IF is set.
7. When the TMR2IF flag bit is set:
   - Clear the associated TRIS bit(s) to enable the output driver.
   - Route the signal to the desired pin by configuring the RxyPPS register.
   - Enable the PWMx module by setting the PWMxEN bit of the PWMxCON register.

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 the PWM module can be enabled during Step 2 by setting the PWMxEN bit of the PWMxCON register.

18.2 Register Definitions: PWM Control

REGISTER 18-1: PWMxCON: PWM CONTROL REGISTER

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

bit 5 **PWMxOUT:** PWM Module Output Level when bit is read.

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
PWMxDC<9:2>							
bit 7							bit 0

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

bit 7-0 **PWMxDC<9:2>:** PWM Duty Cycle Most Significant bits
These bits are the MSbs of the PWM duty cycle. The two LSbs are found in 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
PWMxDC<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 7-6 **PWMxDC<1:0>:** PWM Duty Cycle Least Significant bits
These bits are the LSbs of the PWM duty cycle. The MSbs are found in PWMxDCH Register.

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

TABLE 18-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 20 MHz)

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale	16	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 (FOSC = 8 MHz)

PWM Frequency	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

REGISTER 18-4: PWMTMRS: PWM TIMERS CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1
—	—	—	—	P6TSEL<1:0>		P5TSEL<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 7-4 **Unimplemented:** Read as ‘0’

bit 3-2 **P6TSEL<1:0>:** PWM6 Mode Timer Selection

- 00 = Reserved
- 01 = PWM6 is based on TMR2
- 10 = PWM6 is based on TMR4
- 11 = PWM6 is based on TMR6

bit 1-0 **P5TSEL<1:0>:** PWM5 Mode Timer Selection

- 00 = Reserved
- 01 = PWM5 is based on TMR2
- 10 = PWM5 is based on TMR4
- 11 = PWM5 is based on TMR6

TABLE 18-3: SUMMARY OF REGISTERS ASSOCIATED WITH PWMx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
PWM5CON	PWM5EN	—	PWM5OUT	PWM5POL	—	—	—	—	194
PWM5DCH	PWM5DC<9:2>								194
PWM5DCL	PWM5DC<1:0>		—	—	—	—	—	—	194
PWM6CON	PWM6EN	—	PWM6OUT	PWM6POL	—	—	—	—	194
PWM6DCH	PWM6DC<9:2>								194
PWM6DCL	PWM6DC<1:0>		—	—	—	—	—	—	194
PWMTMRS	—	—	—	—	P6TSEL<1:0>		P5TSEL<1:0>		195
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	105
PIR2	TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BCL2IF	TMR4IF	NCO1IF	106
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	100
PIE2	TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BCL2IE	TMR4IE	NCO1IE	101
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		296
T4CON	—	T4OUTPS<3:0>				TMR4ON	T4CKPS<1:0>		290
T6CON	—	T6OUTPS<3:0>				TMR6ON	T6CKPS<1:0>		290
TMR2	TMR2<7:0>								297
TMR4	TMR4<7:0>								297
TMR6	TMR6<7:0>								297
PR2	PR2<7:0>								297
PR4	PR4<7:0>								297
PR6	PR6<7:0>								297
RxyPPS	—	—	—	RxyPPS<4:0>					161
CWGxDAT	—	—	—	—	DAT<3:0>				213
CLCxSELy	—	—	LCxDyS<5:0>						227
MDSRC	—	—	—	—	MDMS<3:0>				270
MDCARH	—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>				271
MDCARL	—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				272

Legend: - = Unimplemented locations, read as '0'. Shaded cells are not used by the PWM module.

Note 1: PIC16(L)F18345 only.

2: Unimplemented, read as '1'.

19.0 COMPLEMENTARY WAVEFORM GENERATOR (CWG) MODULE

The Complementary Waveform Generator (CWGx) produces complementary waveforms with dead-band delay from a selection of input sources.

The CWGx 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

19.1 Fundamental Operation

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

The off-to-on transition of each output can be delayed from the on-to-off transition of the other output, thereby, creating a time delay immediately where neither output is driven. This is referred to as dead time and is covered in **Section 19.6 "Dead-Band Control"**.

It may be necessary to guard against the possibility of circuit faults or a feedback event arriving too late or not at all. In this case, the active drive must be terminated before the Fault condition causes damage. This is referred to as auto-shutdown and is covered in **Section 19.7 "Auto-Shutdown Control"**.

19.2 Operating Modes

The CWGx module can operate in six different modes, as specified by the MODE<2:0> bits of the CWGxCON0 register:

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

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

All modes include auto-shutdown control as described in **Section 19.11 "Register Definitions: CWG Control"**

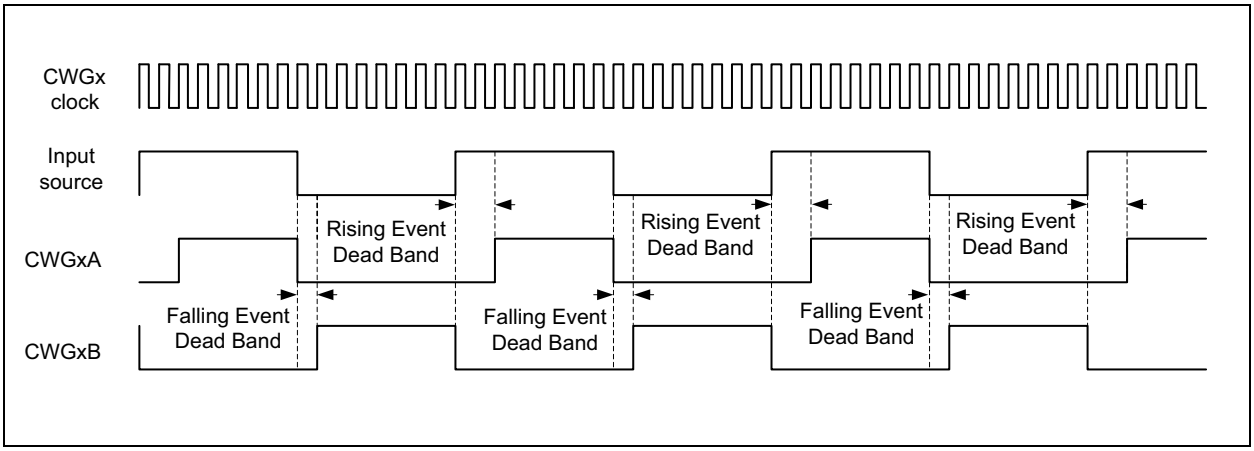
Note:	Except as noted for Full-bridge mode (**Section 19.2.4 "Full-Bridge Modes"**), mode changes should only be performed while EN = 0 (Register 19-1).

19.2.1 HALF-BRIDGE MODE

In Half-Bridge mode, two output signals are generated as true and inverted versions of the input as illustrated in Figure 19-1. A non-overlap (dead-band) time is inserted between the two outputs to prevent shoot-through current in various power supply applications. Dead-band control is described in **Section 19.6 "Dead-Band Control"**. Steering modes are not used in Half-Bridge mode.

The unused outputs, CWGxC and CWGxD, drive similar signals with polarity independently controlled by POLC and POLD, respectively.

FIGURE 19-1: CWGx HALF-BRIDGE MODE OPERATION



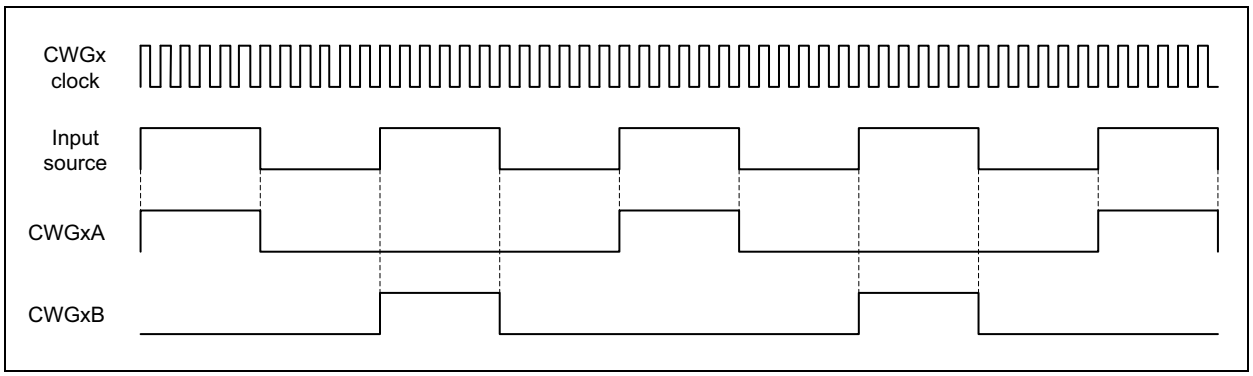
19.2.2 PUSH-PULL MODE

In Push-Pull mode, two output signals are generated, alternating copies of the input as illustrated in Figure 19-2. This alternation creates the push-pull effect required for driving some transformer based power supply designs. Dead-band control is not used in Push-Pull mode. Steering modes are not used in Push-Pull mode.

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

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

FIGURE 19-2: CWGx PUSH-PULL MODE OPERATION



19.2.3 STEERING MODES

In both Synchronous and Asynchronous Steering modes, the modulated input signal can be steered to any combination of four CWG outputs and a fixed-value will be presented on all the outputs not used for the PWM output. Each output has independent polarity, steering, and shutdown options. Dead-band control is not used in either Steering mode.

When STRy = 0 (Register 19-5), the corresponding pin is held at the level defined by SDATy (Register 19-5). When STRy = 1, the pin is driven by the modulated input signal.

The POLy bits (Register 19-2) control the signal polarity only when STRy = 1.

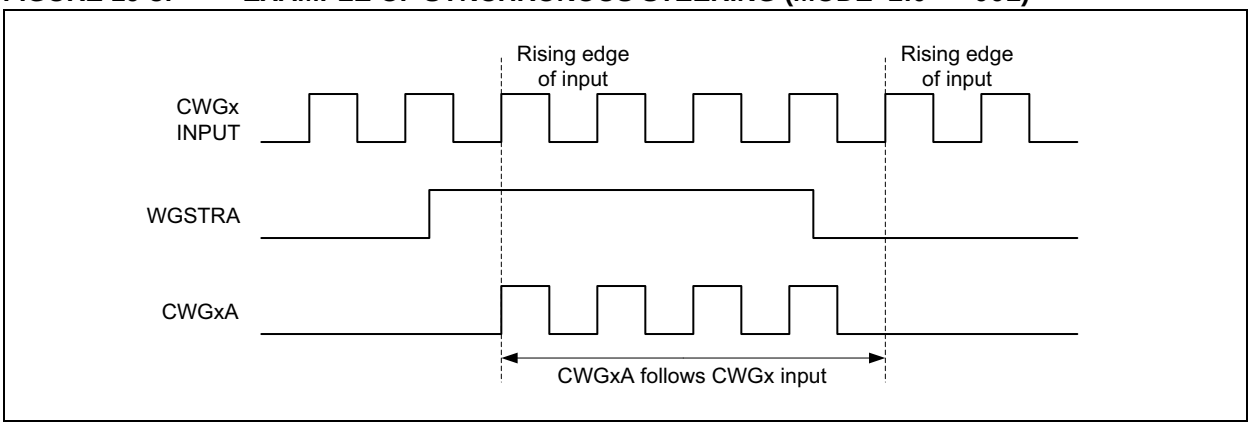
The CWG auto-shutdown operation also applies to Steering modes as described in **Section 19.11 "Register Definitions: CWG Control"**.

Note:	Only the WGSTRy bits are synchronized; the WGSDATy (data) bits are not synchronized.

19.2.3.1 Synchronous Steering Mode

In Synchronous Steering mode (MODE<2:0> bits = 001, Register 19-1), changes to steering selection registers take effect on the next rising edge of the modulated data input (Figure 19-3). In Synchronous Steering mode, the output will always produce a complete waveform.

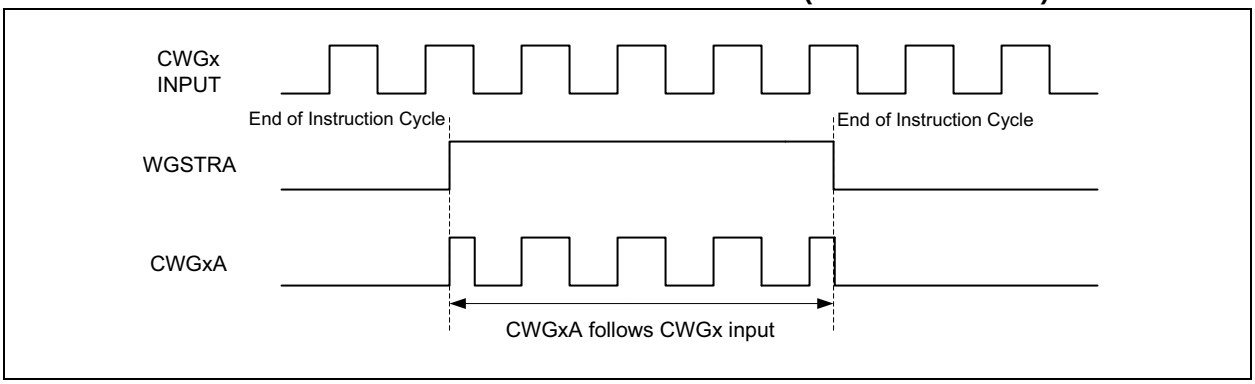
FIGURE 19-3: EXAMPLE OF SYNCHRONOUS STEERING (MODE<2:0> = 001)



19.2.3.2 Asynchronous Steering Mode

In Asynchronous mode (MODE<2:0> bits = 000, Register 19-1), steering takes effect at the end of the instruction cycle that writes to WGxSTR. In Asynchronous Steering mode, the output signal may be an incomplete waveform (Register 19-4). This operation may be useful when the user firmware needs to immediately remove a signal from the output pin.

FIGURE 19-4: EXAMPLE OF ASYNCHRONOUS STEERING (MODE<2:0> = 000)



19.2.3.3 Start-up Considerations

The application hardware must use the proper external pull-up and/or pull-down resistors on the CWG output pins. This is required because all I/O pins are forced to high-impedance at Reset.

The POLy bits (Register 19-2) allow the user to choose whether the output signals are active-high or active-low.

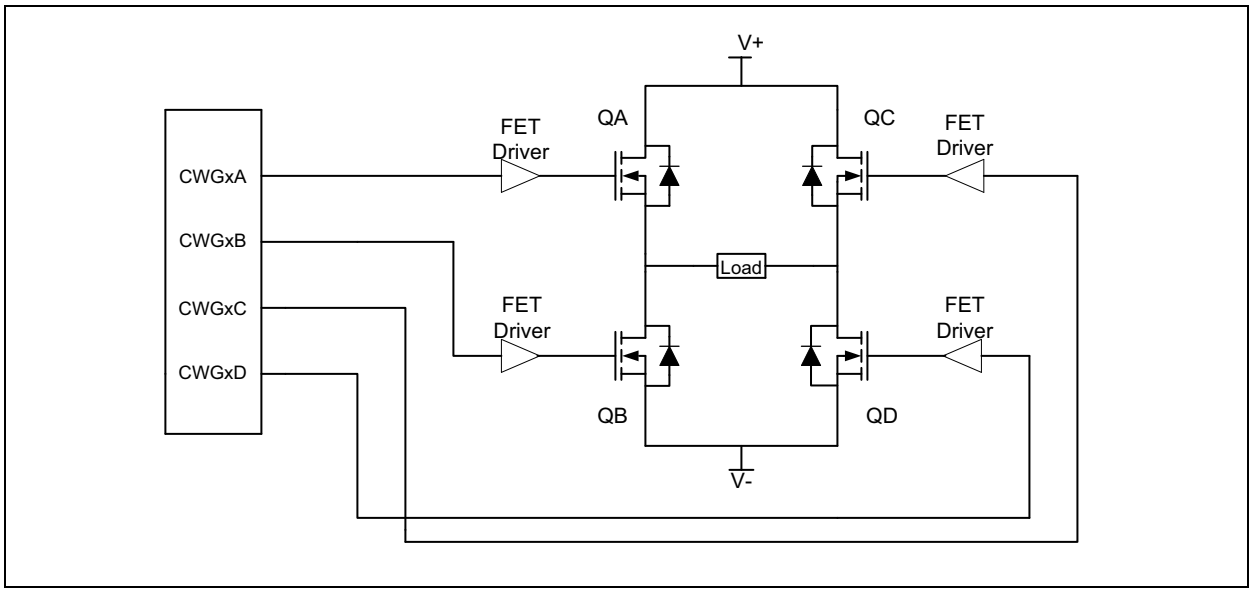
19.2.4 FULL-BRIDGE MODES

In Forward and Reverse Full-Bridge modes, three outputs drive static values while the fourth is modulated by the data input. Dead-band control is described in **Section 19.2.3 "Steering Modes"** and **Section 19.6 "Dead-Band Control"**. Steering modes are not used with either of the Full-Bridge modes.

The mode selection may be toggled between forward and reverse (changing MODE<2:0>) without clearing EN.

When connected as shown in Figure 19-5, the outputs are appropriate for a full-bridge motor driver. Each CWG output signal has independent polarity control, so the circuit can be adapted to high-active and low-active drivers.

FIGURE 19-5: EXAMPLE OF FULL-BRIDGE APPLICATION



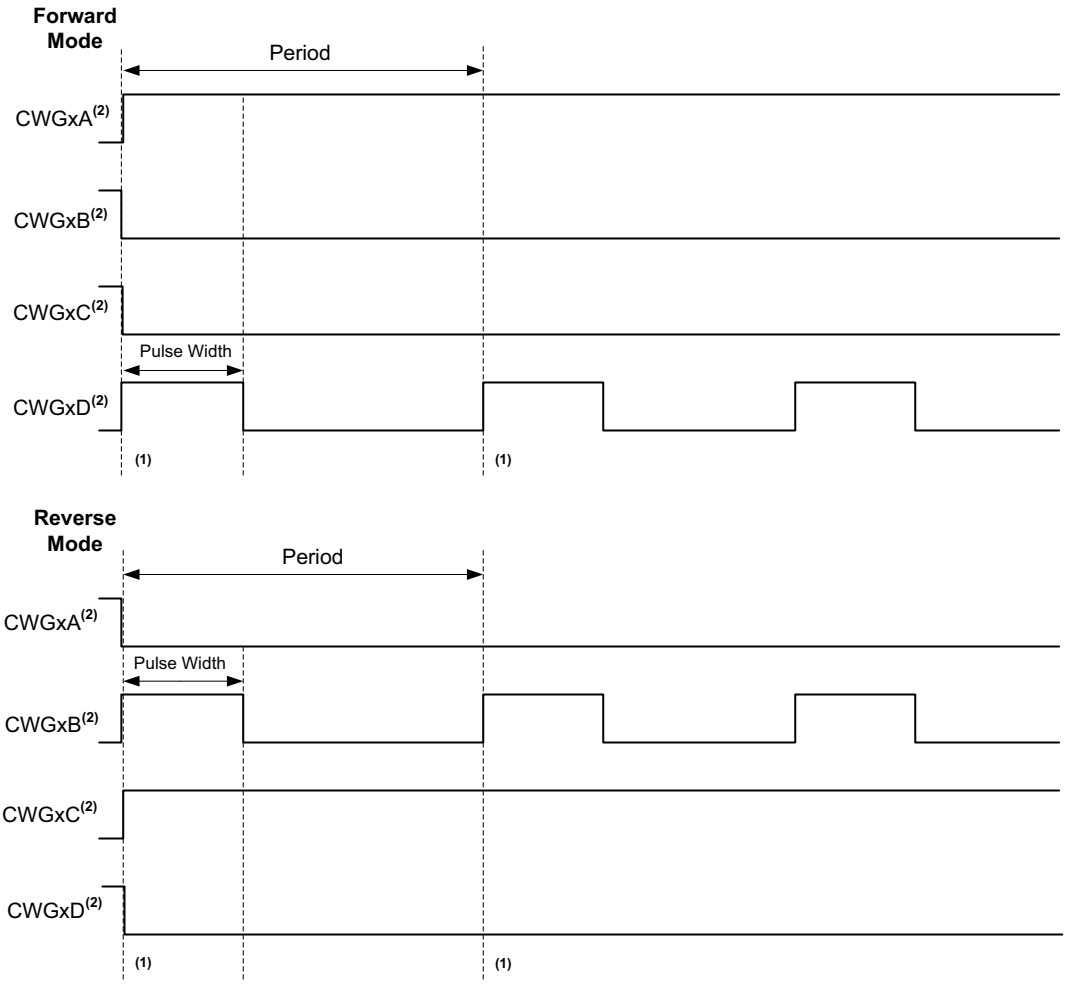
19.2.4.1 Full-Bridge Forward Mode

In Full-Bridge Forward mode (MODE<2:0> = 010), CWGxA is driven to its active state and CWGxD is modulated while CWGxB and CWGxC are driven to their inactive state, as illustrated at the top of Figure 19-6.

19.2.4.2 Full-Bridge Reverse Mode

In Full-Bridge Reverse mode (MODE<2:0> = 011), CWGxC is driven to its active state and CWGxB is modulated while CWGxA and CWGxD are driven to their inactive state, as illustrated at the bottom of Figure 19-6.

FIGURE 19-6: EXAMPLE OF FULL-BRIDGE OUTPUT



Note 1: A rising CWG data input creates a rising event on the modulated output.

2: Output signals shown as active-high; all POLy bits are clear.

19.2.4.3 Direction Change in Full-Bridge Mode

In Full-Bridge mode, changing MODE<2:0> controls the forward/reverse direction. Changes to MODE<2:0> change to the new direction on the next rising edge of the modulated input.

A direction change is initiated in software by changing the MODE<2:0> bits of the WGxCON0 register. The sequence is illustrated in Figure 19-7.

- The associated active output CWGxA and the inactive output CWGxC are switched to drive in the opposite direction.
- The previously modulated output CWGxD is switched to the inactive state, and the previously inactive output CWGxB begins to modulate.
- CWG modulation resumes after the direction-switch dead band has elapsed.

19.2.4.4 Dead-Band Delay in Full-Bridge Mode

Dead-band delay is important when either of the following conditions is true:

1. The direction of the CWG output changes when the duty cycle of the data input is at or near 100%, or
2. The turn-off time of the power switch, including the power device and driver circuit, is greater than the turn-on time.

The dead-band delay is inserted only when changing directions, and only the modulated output is affected. The statically-configured outputs (CWGxA and CWGxC) are not afforded dead band, and switch essentially simultaneously.

Figure 19-7 shows an example of the CWG outputs changing directions from forward to reverse, at near 100% duty cycle. In this example, at time t1, the output of CWGxA and CWGxD become inactive, while output CWGxC becomes active. Since the turn-off time of the power devices is longer than the turn-on time, a shoot-through current will flow through power devices QC and QD for the duration of 't'. The same phenomenon will occur to power devices QA and QB for the CWG direction change from reverse to forward.

If changing the CWG direction at high duty cycle is required for an application, two possible solutions for eliminating the shoot-through current are:

1. Reduce the CWG duty cycle for one period before changing directions.
2. Use switch drivers that can drive the switches off faster than they can drive them on.

FIGURE 19-7: EXAMPLE OF PWM DIRECTION CHANGE AT NEAR 100% DUTY CYCLE

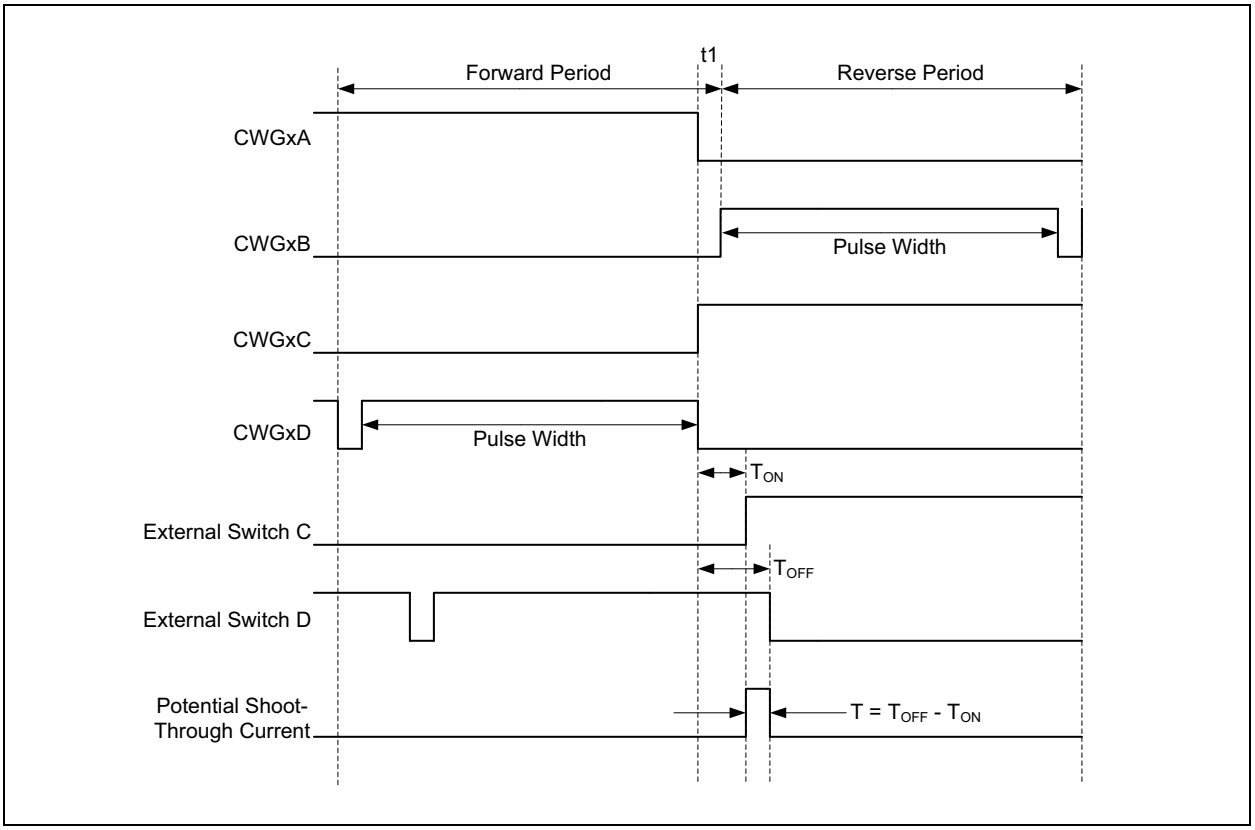


FIGURE 19-8: SIMPLIFIED CWGx BLOCK DIAGRAM (HALF-BRIDGE MODE, MODE<2:0> = 100)

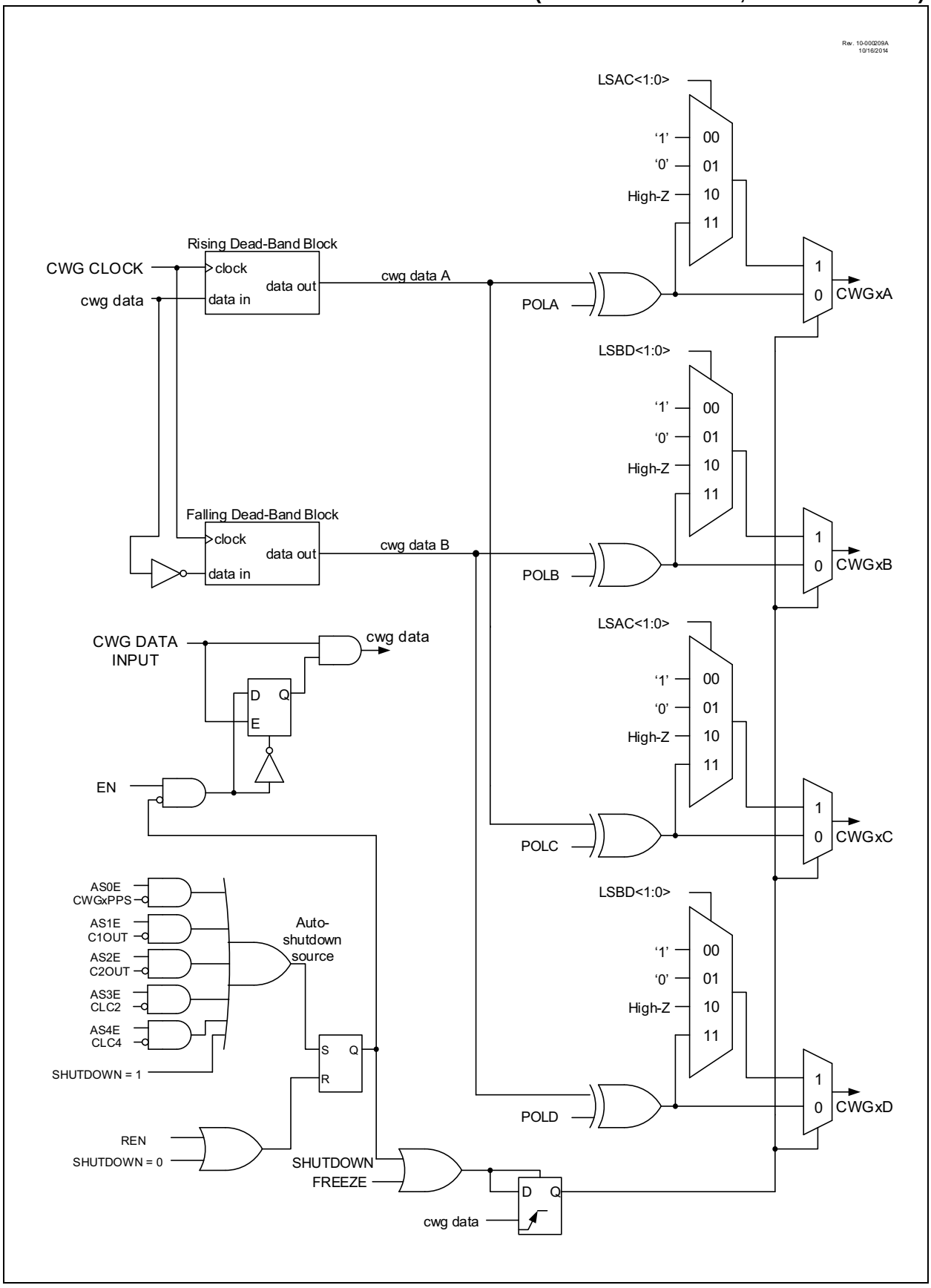


FIGURE 19-9: SIMPLIFIED CWG BLOCK DIAGRAM (PUSH-PULL MODE, MODE <2:0> = 101)

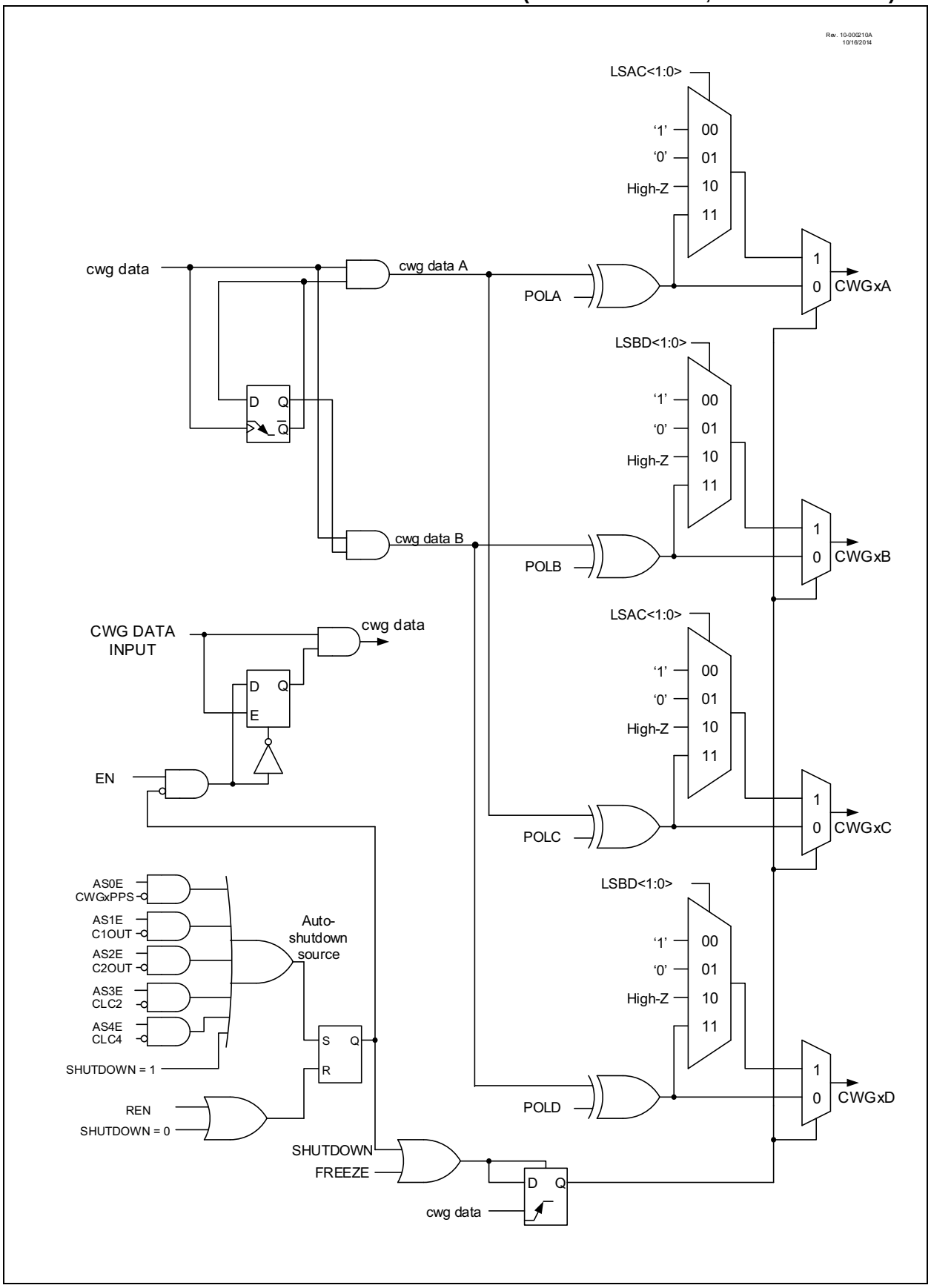


FIGURE 19-10: SIMPLIFIED CWG BLOCK DIAGRAM (OUTPUT STEERING MODES)

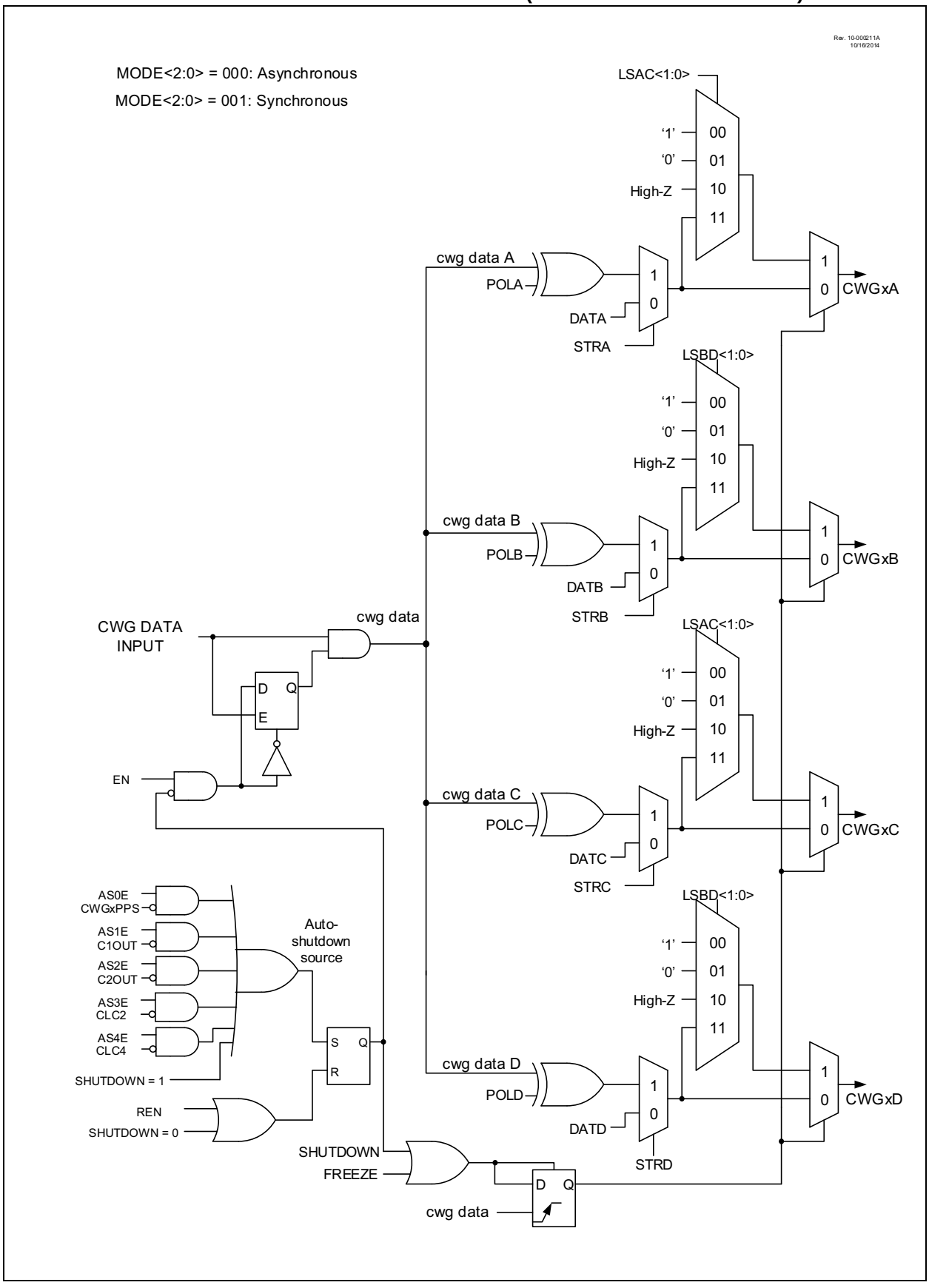
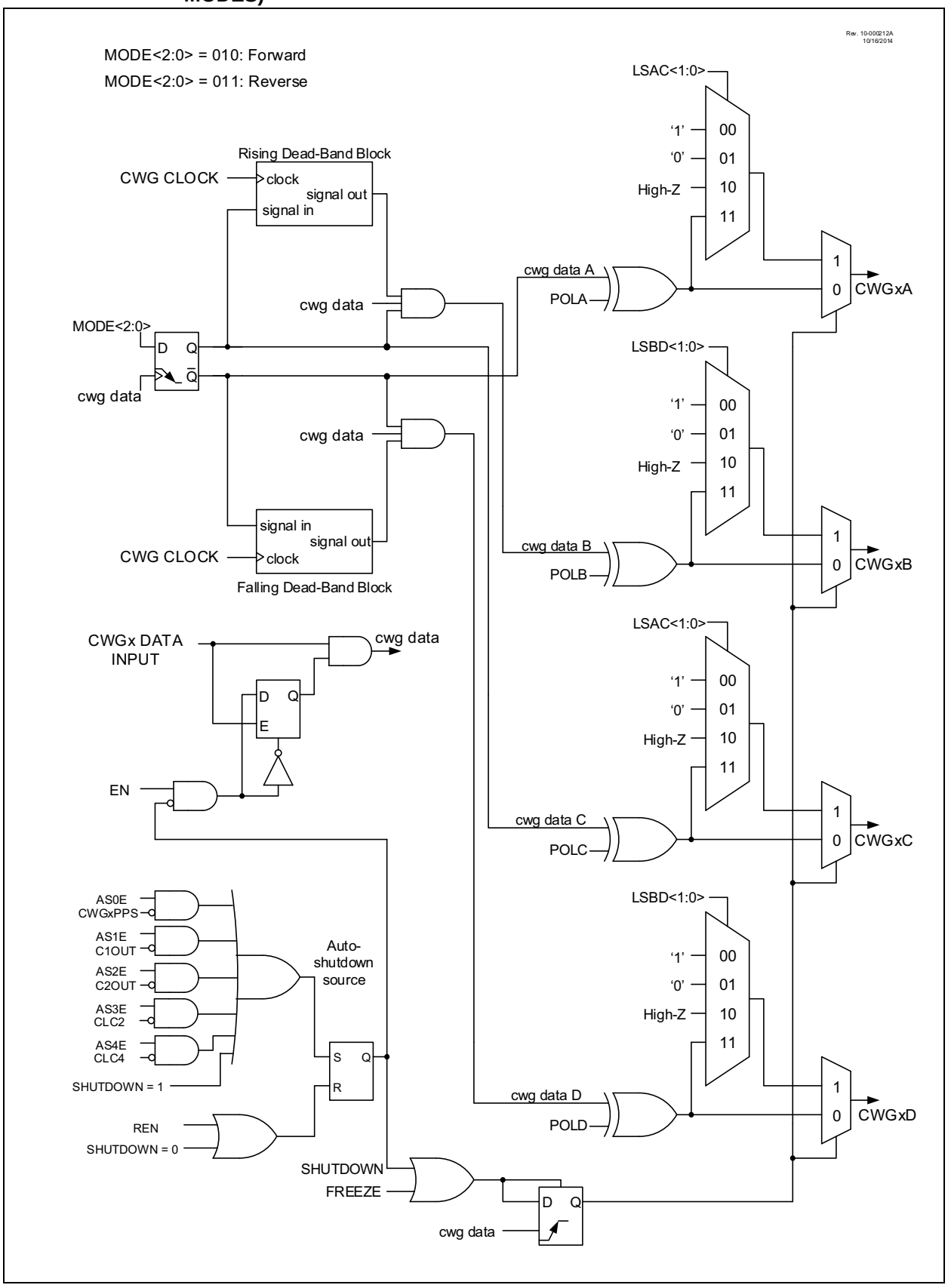


FIGURE 19-11: SIMPLIFIED CWG BLOCK DIAGRAM (FORWARD AND REVERSE FULL-BRIDGE MODES)



19.3 Clock Source

The clock source is used to drive the dead-band timing circuits. The CWGx module allows the following clock sources to be selected:

- FOSC (system clock)
- HFINTOSC (16 MHz only)

When the HFINTOSC is selected the HFINTOSC will be kept running during Sleep. Therefore, CWG modes requiring dead band can operate in Sleep provided that the CWG data input is also active during Sleep. The clock sources are selected using the CS bit of the CWGxCLKCON register (Register 19-3).

19.4 Selectable Input Sources

The CWG generates the output waveforms from the input sources in Table 19-1.

TABLE 19-1: SELECTABLE INPUT SOURCES

Source Peripheral	Signal Name
CWGxPPS	CWG PPS input connection
C1OUT	Comparator 1 output
C2OUT	Comparator 2 output
CCP1	Capture/Compare/PWM output
CCP2	Capture/Compare/PWM output
CCP3	Capture/Compare/PWM output
CCP4	Capture/Compare/PWM output
PWM5	PWM5 output
PWM6	PWM6 output
NCO1	Numerically Controlled Oscillator (NCO) output
CLC1	Configurable Logic Cell 1 output
CLC2	Configurable Logic Cell 2 output
CLC3	Configurable Logic Cell 3 output
CLC4	Configurable Logic Cell 4 output

The input sources are selected using the DAT<3:0> bits in the CWGxDAT register (Register 19-4).

19.5 Output Control

Immediately after the CWG module is enabled, the complementary drive is configured with all output drives cleared.

19.5.1 CWGx OUTPUTS

Each CWG output can be routed to a Peripheral Pin Select (PPS) output via the RxyPPS register (see **Section 12.0 "Peripheral Pin Select (PPS) Module"**).

19.5.2 POLARITY CONTROL

The polarity of each CWG output can be selected independently. When the output polarity bit is set, the corresponding output is active-low. Clearing the output polarity bit configures the corresponding output as active-high. However, polarity does not affect the override levels. Output polarity is selected with the POLy bits of the CWGxCON1 register.

19.6 Dead-Band Control

Dead-band control provides for non-overlapping output signals to prevent current shoot-through in power switches. The CWGx modules contain two 6-bit dead-band counters. These counters can be loaded with values that will determine the length of the dead band initiated on either the rising or falling edges of the input source. Dead-band control is used in either Half-Bridge or Full-Bridge modes.

The rising-edge dead-band delay is determined by the rising dead-band count register (Register 19-8, CWGxDBR) and the falling-edge dead-band delay is determined by the falling dead-band count register (Register 19-9, CWGxDBF). Dead-band duration is established by counting the CWG clock periods from zero up to the value loaded into either the rising or falling dead-band counter registers. The dead-band counters are incremented on every rising edge of the CWG clock source.

19.6.1 RISING EDGE AND REVERSE DEAD BAND

In Half-Bridge mode, the rising edge dead band delays the turn-on of the CWGxA output after the rising edge of the CWG data input. In Full-Bridge mode, the reverse dead-band delay is only inserted when changing directions from Forward mode to Reverse mode, and only the modulated output CWGxB is affected.

The CWGxDBR register determines the duration of the dead-band interval on the rising edge of the input source signal. This duration is from 0 to 64 periods of the CWG clock.

Dead band is always initiated on the edge of the input source signal. A count of zero indicates that no dead band is present.

If the input source signal reverses polarity before the dead-band count is completed then no output will be seen on the respective output.

The CWGxDBR register value is double-buffered. If EN = 0 (Register 19-1), the buffer is loaded when CWGxDBR is written. If EN = 1, then the buffer will be loaded at the rising edge, following the first falling edge of the data input after the LD bit (Register 19-1) is set.

19.6.2 FALLING EDGE AND FORWARD DEAD BAND

In Half-Bridge mode, the falling edge dead band delays the turn-on of the CWGxB output at the falling edge of the CWGx data input. In Full-Bridge mode, the forward dead-band delay is only inserted when changing directions from Reverse mode to Forward mode, and only the modulated output CWGxD is affected.

The CWGxDBF register determines the duration of the dead-band interval on the falling edge of the input source signal. This duration is from zero to 64 periods of the CWG clock.

Dead band is always initiated on the edge of the input source signal. A count of zero indicates that no dead band is present.

If the input source signal reverses polarity before the dead-band count is completed, then no output will be seen on the respective output.

The CWGxDBF register value is double-buffered. When EN = 0 (Register 19-1), the buffer is loaded when CWGxDBF is written. If EN = 1, then the buffer will be loaded at the rising edge following the first falling edge of the data input after the LD (Register 19-1) is set.

19.6.3 DEAD-BAND JITTER

The CWG input data signal may be asynchronous to the CWG input clock, so some jitter may occur in the observed dead band in each cycle. The maximum jitter is equal to one CWG clock period. See Equation 19-1 for details and an example.

EQUATION 19-1: DEAD-BAND DELAY TIME CALCULATION

$$T_{DEAD-BAND\_MIN} = \frac{1}{F_{CWG\_CLOCK}} \bullet DBx<4:0>$$

$$T_{DEAD-BAND\_MAX} = \frac{1}{F_{CWG\_CLOCK}} \bullet DBx<4:0> + 1$$

$$T_{JITTER} = T_{DEAD-BAND\_MAX} - T_{DEAD-BAND\_MIN}$$

$$T_{JITTER} = \frac{1}{F_{CWG\_CLOCK}}$$

$$T_{DEAD-BAND\_MAX} = T_{DEAD-BAND\_MIN} + T_{JITTER}$$

Example:

$$DBR<4:0> = 0x0A = 10$$

$$F_{CWG\_CLOCK} = 8\ MHz$$

$$T_{JITTER} = \frac{1}{8\ MHz}$$

$$T_{DEAD-BAND\_MIN} = 125\ ns \bullet 10 = 1.25\ \mu s$$

$$T_{DEAD-BAND\_MAX} = 1.25\ \mu s + 0.125\ \mu s = 1.37\ \mu s$$

19.7 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.

19.7.1 SHUTDOWN

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

- Software generated
- External input

The SHUTDOWN bit indicates when a Shutdown condition exists. The bit may be set or cleared in software or by hardware.

19.7.1.1 Software-Generated Shutdown

Setting the SHUTDOWN bit of the CWGxAS0 register will force the CWG into the shutdown state.

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

When auto-restart is enabled, the SHUTDOWN bit will clear automatically and resume operation on the next rising edge event.

19.7.1.2 External Input Source Shutdown

Any of the auto-shutdown external inputs can be selected to suspend the CWG operation. These sources are individually enabled by the ASxE bits of the CWGxAS1 register (Register 19-7). When any of the selected inputs goes active (pins are active-low), the CWG outputs will immediately switch to the override levels selected by the LSBD<1:0> and LSAC<1:0> bits without any software delay (**Section 19.7.1.3 "Pin Override Levels"**). Any of the following external input sources can be selected to cause a Shutdown condition:

- Comparator C1
- Comparator C2
- CLC2
- CWGxPPS

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.

19.7.1.3 Pin Override Levels

The levels driven to the CWG outputs during an auto-shutdown event are controlled by the LSBD<1:0> and LSAC<1:0> bits of the CWGxAS0 register (Register 19-6). The LSBD<1:0> bits control CWGxB/D output levels, while the LSAC<1:0> bits control the CWGxA/C output levels.

19.7.1.4 Auto-Shutdown Interrupts

When an auto-shutdown event occurs, either by software or hardware setting SHUTDOWN, the CWGxIF flag bit of the PIR4 register is set (Register 7-11).

19.8 Auto-Shutdown Restart

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

- Software controlled
- Auto-restart

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

19.8.1 SOFTWARE-CONTROLLED RESTART

If the REN bit of the CWGxASD0 register is clear (REN = 0), the CWGx module must be restarted after an auto-shutdown event through software.

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

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

19.8.2 AUTO-RESTART

If the REN bit of the CWGxASD0 register is set (REN = 1), the CWGx module will restart from the shutdown state automatically.

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

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

19.9 Operation During Sleep

The CWGx module will operate during Sleep, provided that the input sources remain active.

If the HFINTOSC is selected as the module clock source, dead-band generation will remain active. This will have a direct effect on the Sleep mode current.

19.10 Configuring the CWG

1. Ensure that the TRIS control bits corresponding to CWG outputs are set so that all are configured as inputs, ensuring that the outputs are inactive during setup. External hardware should ensure that pin levels are held to safe levels.
2. Clear the EN bit, if not already cleared.
3. Configure the MODE<2:0> bits of the CWGxCON0 register to set the output operating mode.
4. Configure the POLy bits of the CWGxCON1 register to set the output polarities.
5. Configure the DAT<3:0> bits of the CWGxDAT register to select the data input source.
6. If a Steering mode is selected, configure the STRy bits to select the desired output on the CWG outputs.
7. Configure the LSBD<1:0> and LSAC<1:0> bits of the CWGxAS0 register to select the auto-shutdown output override states (this is necessary even if not using auto-shutdown because start-up will be from a shutdown state).
8. If auto-restart is desired, set the REN bit of CWGxAS0.
9. If auto-shutdown is desired, configure the ASxE bits of the CWGxAS1 register to select the shutdown source.
10. Set the desired rising and falling dead-band times with the CWGxDBR and CWGxDBF registers.
11. Select the clock source in the CWGxCLKCON register.
12. Set the EN bit to enable the module.
13. Clear the TRIS bits that correspond to the CWG outputs to set them as outputs.
14. If auto-restart is to be used, set the REN bit and the SHUTDOWN bit will be cleared automatically. Otherwise, clear the SHUTDOWN bit in software to start the CWG.

19.11 Register Definitions: CWG Control

REGISTER 19-1: CWGxCON0: CWGx CONTROL REGISTER 0

R/W-0/0	R/W/HC-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
EN	LD(1)	—	—	—	MODE<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	HS/HC = Bit is set/cleared by hardware

bit 7 **EN:** CWGx Enable bit
- 1 = CWGx is enabled
- 0 = CWGx is disabled

bit 6 **LD:** CWG Load Buffers bit(1)
- 1 = Dead-band count buffers to be loaded on CWG data rising edge following first falling edge after this bit is set.
- 0 = Buffers remain unchanged

bit 5-3 **Unimplemented**: Read as '0'

bit 2-0 **MODE<2:0>**: CWGx Mode bits
- 111 = Reserved
- 110 = Reserved
- 101 = CWG outputs operate in Push-Pull mode
- 100 = CWG outputs operate in Half-Bridge mode
- 011 = CWG outputs operate in Reverse Full-Bridge mode
- 010 = CWG outputs operate in Forward Full-Bridge mode
- 001 = CWG outputs operate in Synchronous Steering mode
- 000 = CWG outputs operate in Asynchronous Steering mode

Note 1: This bit can only be set after EN = 1; it cannot be set in the same cycle when EN is set.

REGISTER 19-2: CWGxCON1: CWGx CONTROL REGISTER 1

U-0	U-0	R-x	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IN	—	POLD	POLC	POLB	POLA
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 **IN:** CWGx Data Input Signal (read-only)

bit 4 **Unimplemented:** Read as ‘0’

bit 3 **POLD:** WGxD Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

bit 2 **POLC:** WGxC Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

bit 1 **POLB:** WGxB Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

bit 0 **POLA:** WGxA Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

REGISTER 19-3: CWGxCLKCON: CWGx CLOCK INPUT SELECTION REGISTER

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

bit 0 **CS:** CWG Clock Source Selection Select bits

WGCLK	Clock Source
0	FOSC
1	HFINTOSC (remains operating during Sleep)

REGISTER 19-4: CWGxDAT: CWGx DATA INPUT SELECTION 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
—	—	—	—	DAT<3:0>			
bit 7							bit 0

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

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

bit 3-0 **DAT<3:0>:** CWG Data Input Selection bits

WGDAT	Data Source
0000	CWGxPPS
0001	C1OUT
0010	C2OUT
0011	CCP1
0100	CCP2
0101	CCP3
0110	CCP4
0111	PWM5
1000	PWM6
1001	NCO1
1010	CLC1
1011	CLC2
1100	CLC3
1101	CLC4
1110	Reserved
1111	Reserved

REGISTER 19-5: CWGxSTR(1): CWG STEERING CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
OVRD	OVRC	OVRB	OVRA	STRD(2)	STRC(2)	STRB(2)	STRA(2)
bit 7							bit 0

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

bit 7 **OVRD:** Steering Data D bit

bit 6 **OVRC:** Steering Data C bit

bit 5 **OVRB:** Steering Data B bit

bit 4 **OVRA:** Steering Data A bit

bit 3 **STRD:** Steering Enable bit D(2)
- 1 = CWGxD output has the CWGx data input waveform with polarity control from POLD bit
- 0 = CWGxD output is assigned to value of OVRD bit

bit 2 **STRC:** Steering Enable bit C(2)
- 1 = CWGxC output has the CWGx data input waveform with polarity control from POLC bit
- 0 = CWGxC output is assigned to value of OVRC bit

bit 1 **STRB:** Steering Enable bit B(2)
- 1 = CWGxB output has the CWGx data input waveform with polarity control from POLB bit
- 0 = CWGxB output is assigned to value of OVRB bit

bit 0 **STRA:** Steering Enable bit A(2)
- 1 = CWGxA output has the CWGx data input waveform with polarity control from POLA bit
- 0 = CWGxA output is assigned to value of OVRA bit

Note 1: The bits in this register apply only when MODE<2:0> = 00x (Register 19-1, steering modes).

2: This bit is double-buffered when MODE<2:0> = 001.

REGISTER 19-6: CWGxAS0: CWG AUTO-SHUTDOWN CONTROL REGISTER 0

R/W/HS/SC-0/0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	U-0	U-0
SHUTDOWN	REN	LSBD<1:0>		LSAC<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 **SHUTDOWN:** Auto-Shutdown Event Status bit(1,2)
- 1 = An auto-shutdown state is in effect
- 0 = No auto-shutdown event has occurred

bit 6 **REN:** Auto-Restart Enable bit
- 1 = Auto-restart is enabled
- 0 = Auto-restart is disabled

bit 5-4 **LSBD<1:0>:** CWGxB and CWGxD Auto-Shutdown State Control bits
- 11 = A logic '1' is placed on CWGxB/D when an auto-shutdown event occurs.
- 10 = A logic '0' is placed on CWGxB/D when an auto-shutdown event occurs.
- 01 = Pin is tri-stated on CWGxB/D when an auto-shutdown event occurs.
- 00 = The inactive state of the pin, including polarity, is placed on CWGxB/D after the required dead-band interval when an auto-shutdown event occurs.

bit 3-2 **LSAC<1:0>:** CWGxA and CWGxC Auto-Shutdown State Control bits
- 11 = A logic '1' is placed on CWGxA/C when an auto-shutdown event occurs.
- 10 = A logic '0' is placed on CWGxA/C when an auto-shutdown event occurs.
- 01 = Pin is tri-stated on CWG1A/C when an auto-shutdown event occurs.
- 00 = The inactive state of the pin, including polarity, is placed on CWGxA/C after the required dead-band interval when an auto-shutdown event occurs.

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

Note 1: This bit may be written while EN = 0 (Register 19-1), to place the outputs into the shutdown configuration.

2: The outputs will remain in auto-shutdown state until the next rising edge of the CWG data input after this bit is cleared.

REGISTER 19-7: CWGxAS1: CWG AUTO-SHUTDOWN CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E
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 7-5 **Unimplemented:** Read as '0'

bit 4 **AS4E:** CWG Auto-Shutdown Source 4 (CLC4) Enable bit
1 = Auto-shutdown for CLC4 is enabled
0 = Auto-shutdown for CLC4 is disabled

bit 3 **AS3E:** CWG Auto-Shutdown Source 3 (CLC2) Enable bit
1 = Auto-shutdown from CLC2 is enabled
0 = Auto-shutdown from CLC2 is disabled

bit 2 **AS2E:** CWG Auto-Shutdown Source 2 (C2) Enable bit
1 = Auto-shutdown from Comparator 2 is enabled
0 = Auto-shutdown from Comparator 2 is disabled

bit 1 **AS1E:** CWG Auto-Shutdown Source 1 (C1) Enable bit
1 = Auto-shutdown from Comparator 1 is enabled
0 = Auto-shutdown from Comparator 1 is disabled

bit 0 **AS0E:** CWG Auto-Shutdown Source 0 (CWGxPPS) Enable bit
1 = Auto-shutdown from CWGxPPS is enabled
0 = Auto-shutdown from CWGxPPS is disabled

REGISTER 19-8: CWGxDBR: CWGx RISING DEAD-BAND COUNT REGISTER

U-0	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
—	—	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 **DBR<5:0>:** CWG Rising Edge Triggered Dead-Band Count bits
11 1111 = 63-64 CWG clock periods
11 1110 = 62-63 CWG clock periods
.
.
.
00 0010 = 2-3 CWG clock periods
00 0001 = 1-2 CWG clock periods
00 0000 = 0 CWG clock periods. Dead-band generation is bypassed.

REGISTER 19-9: CWGxDBF: CWGx FALLING DEAD-BAND COUNT REGISTER

U-0	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
—	—	DBF<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 **DBF<5:0>:** CWG Falling Edge Triggered Dead-Band Count bits

11 1111 = 63-64 CWG clock periods
11 1110 = 62-63 CWG clock periods
.
.
.
00 0010 = 2-3 CWG clock periods
00 0001 = 1-2 CWG clock periods
00 0000 = 0 CWG clock periods. Dead-band generation is bypassed.

TABLE 19-2: SUMMARY OF REGISTERS ASSOCIATED WITH CWGx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
PIR4	CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	108
PIE4	CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	103
CWG1CON0	EN	LD	—	—	—	MODE<2:0>			211
CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	212
CWG1CLKCON	—	—	—	—	—	—	—	CS	212
CWG1DAT	—	—	—	—	DAT<3:0>				213
CWG1STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	214
CWG1AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	215
CWG1AS1	—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	216
CWG1DBR	—	—	DBR<5:0>						216
CWG1DBF	—	—	DBF<5:0>						217
CWG1PPS	—	—	—	CWG1PPS<4:0>					160
CWG2CON0	EN	LD	—	—	—	MODE<2:0>			211
CWG2CON1	—	—	IN	—	POLD	POLC	POLB	POLA	212
CWG2CLKCON	—	—	—	—	—	—	—	CS	212
CWG2DAT	—	—	—	—	DAT<3:0>				213
CWG2STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	214
CWG2AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	215
CWG2AS1	—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	216
CWG2DBR	—	—	DBR<5:0>						216
CWG2DBF	—	—	DBF<5:0>						217
CWG2PPS	—	—	—	CWG2PPS<4:0>					160
RxyPPS	—	—	—	RxyPPS<4:0>					161

Note 1: PIC16(L)F18345 only.
2: Unimplemented, read as '0'.

20.0 CONFIGURABLE LOGIC CELL (CLC)

The Configurable Logic Cell (CLCx) provides programmable logic that operates outside the speed limitations of software execution. The logic cell takes up to 32 input signals and, through the use of configurable gates, reduces the 32 inputs to four logic lines that drive one of eight selectable single-output logic functions.

Input sources are a combination of the following:

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

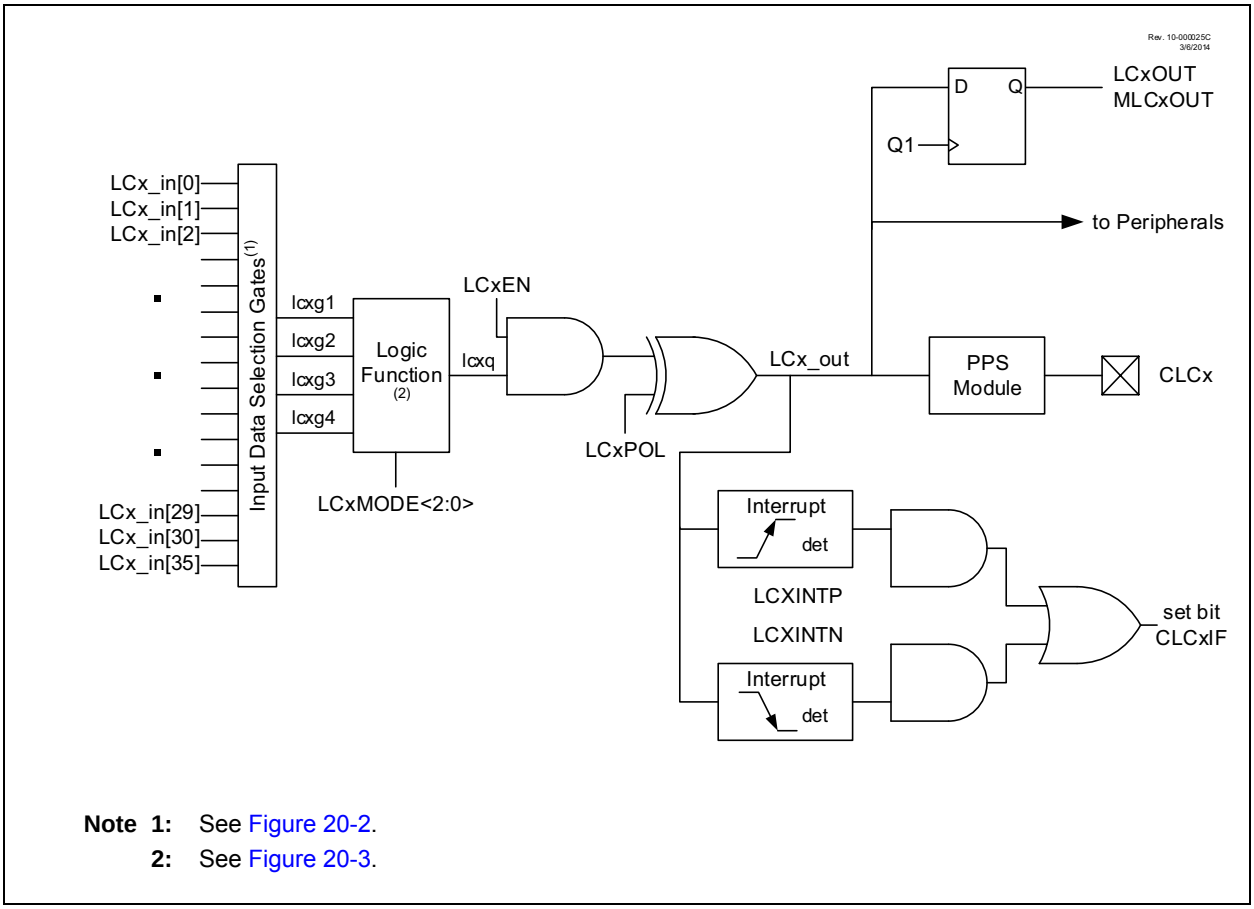
The output can be directed internally to peripherals and to an output pin.

Refer to Figure 20-1 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 20-1: CLCx SIMPLIFIED BLOCK DIAGRAM



Note 1: See Figure 20-2.
2: See Figure 20-3.

20.1 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.

20.1.1 DATA SELECTION

There are 36 signals available as inputs to the configurable logic. Four 36-input multiplexers are used to select the inputs to pass on to the next stage.

Data selection is through four multiplexers as indicated on the left side of Figure 20-2. Data inputs in the figure are identified by a generic numbered input name.

Table 20-1 correlates the generic input name to the actual signal for each CLC module. The column labeled 'LCxDyS<5:0> Value' indicates the MUX selection code for the selected data input. LCxDyS is an abbreviation for the MUX select input codes: LCxD1S<5:0> through LCxD4S<5:0>.

Data inputs are selected with CLCxSEL0 through CLCxSEL3 registers (Register 20-3 through Register 20-6).

Note:	Data selections are undefined at power-up.

TABLE 20-1: CLCx DATA INPUT SELECTION

LCxDyS<5:0> Value	CLCx Input Source
100011 [35]	TMR6/PR6 match
100010 [34]	TMR5 overflow
100001 [33]	TMR4/PR4 match
100000 [32]	TMR3 overflow
11111 [31]	FOSC
11110 [30]	HFINTOSC
11101 [29]	LFINTOSC
11100 [28]	ADCRC
11011 [27]	IOCIF int flag bit
11010 [26]	TMR2/PR2 match
11001 [25]	TMR1 overflow
11000 [24]	TMR0 overflow
10111 [23]	EUSART1 (DT) output
10110 [22]	EUSART1 (TX/CK) output
10101 [21]	SDA2
10100 [20]	SCL2
10011 [19]	SDA1
10010 [18]	SCL1
10001 [17]	PWM6 output
10000 [16]	PWM5 output
01111 [15]	CCP4 output
01110 [14]	CCP3 output
01101 [13]	CCP2 output
01100 [12]	CCP1 output
01011 [11]	CLKR output
01010 [10]	DSM output
01001 [9]	C2 output
01000 [8]	C1 output
00111 [7]	CLC4 output
00110 [6]	CLC3 output
00101 [5]	CLC2 output
00100 [4]	CLC1 output
00011 [3]	CLCIN3PPS
00010 [2]	CLCIN2PPS
00001 [1]	CLCIN1PPS
00000 [0]	CLCIN0PPS

20.1.2 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 20-2 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 20-2: DATA GATING LOGIC

CLCxGLSy	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 20-7)
- Gate 2: CLCxGLS1 (Register 20-8)
- Gate 3: CLCxGLS2 (Register 20-9)
- Gate 4: CLCxGLS3 (Register 20-10)

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 20-2. 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.

20.1.3 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 20-3. 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.

20.1.4 OUTPUT POLARITY

The last stage in the configurable logic cell is the output polarity. Setting the LCxPOL bit of the CLCxPOL 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.

20.2 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 PIR3 register will be set when either edge detector is triggered and its associated enable bit is set. The LCxINTP bit 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:

- CLCxIE bit of the PIE3 register
- 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 PIR3 register, 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.

20.3 Output Mirror Copies

Mirror copies of all LCxCON output bits are contained in the CLCDATA register. Reading this register reads the outputs of all CLCs simultaneously. This prevents any reading skew introduced by testing or reading the LCxOUT bits in the individual CLCxCON registers.

20.4 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.

20.5 Operation During Sleep

The CLC module operates independently from the system clock and will continue to run during Sleep, provided that the input sources selected remain active.

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

In other words, if the HFINTOSC is simultaneously selected as the system clock and as a CLC input source, when the CLC is enabled, the CPU will go idle during Sleep, but the CLC will continue to operate and the HFINTOSC will remain active.

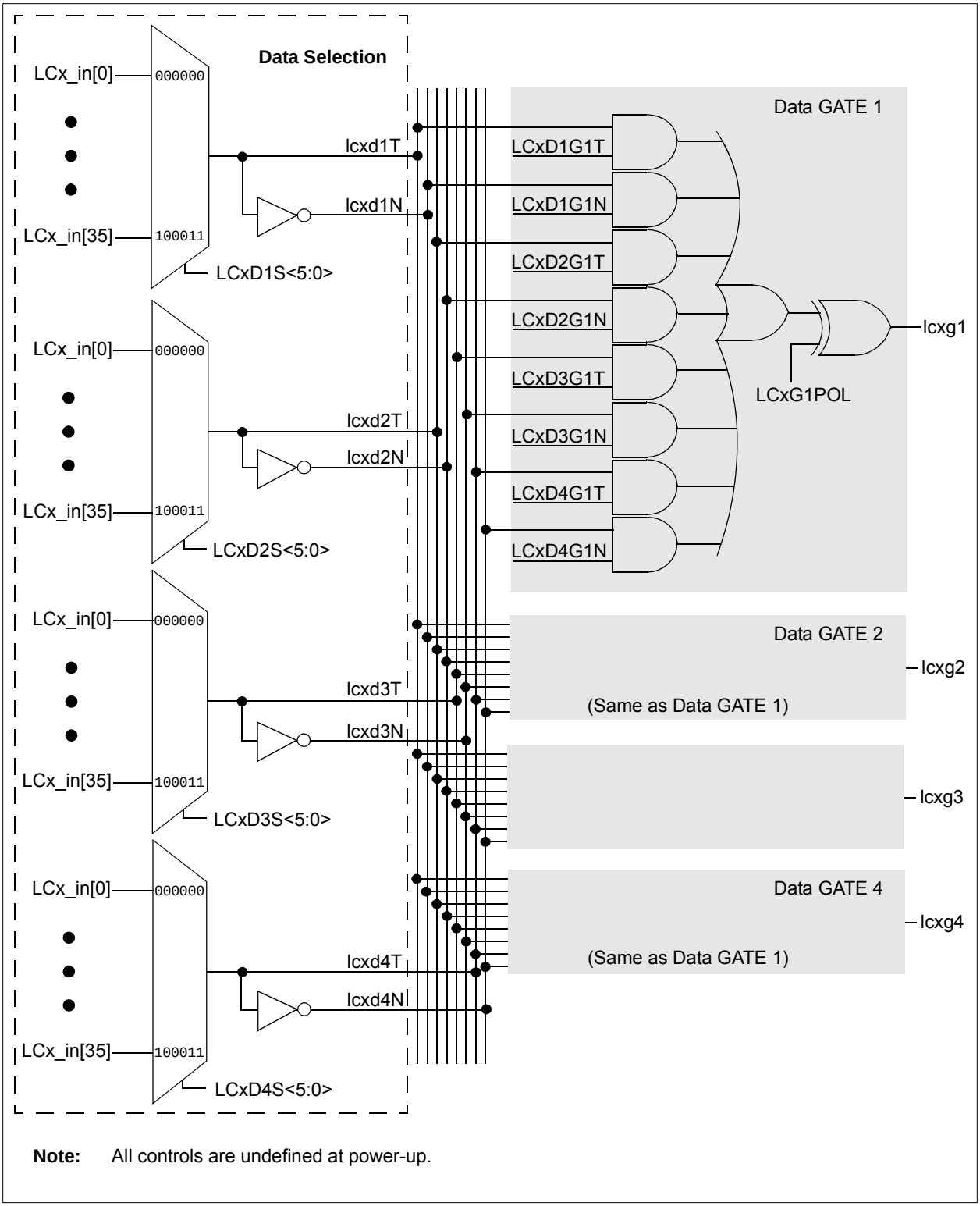
This will have a direct effect on the Sleep mode current.

20.6 CLCx Setup Steps

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

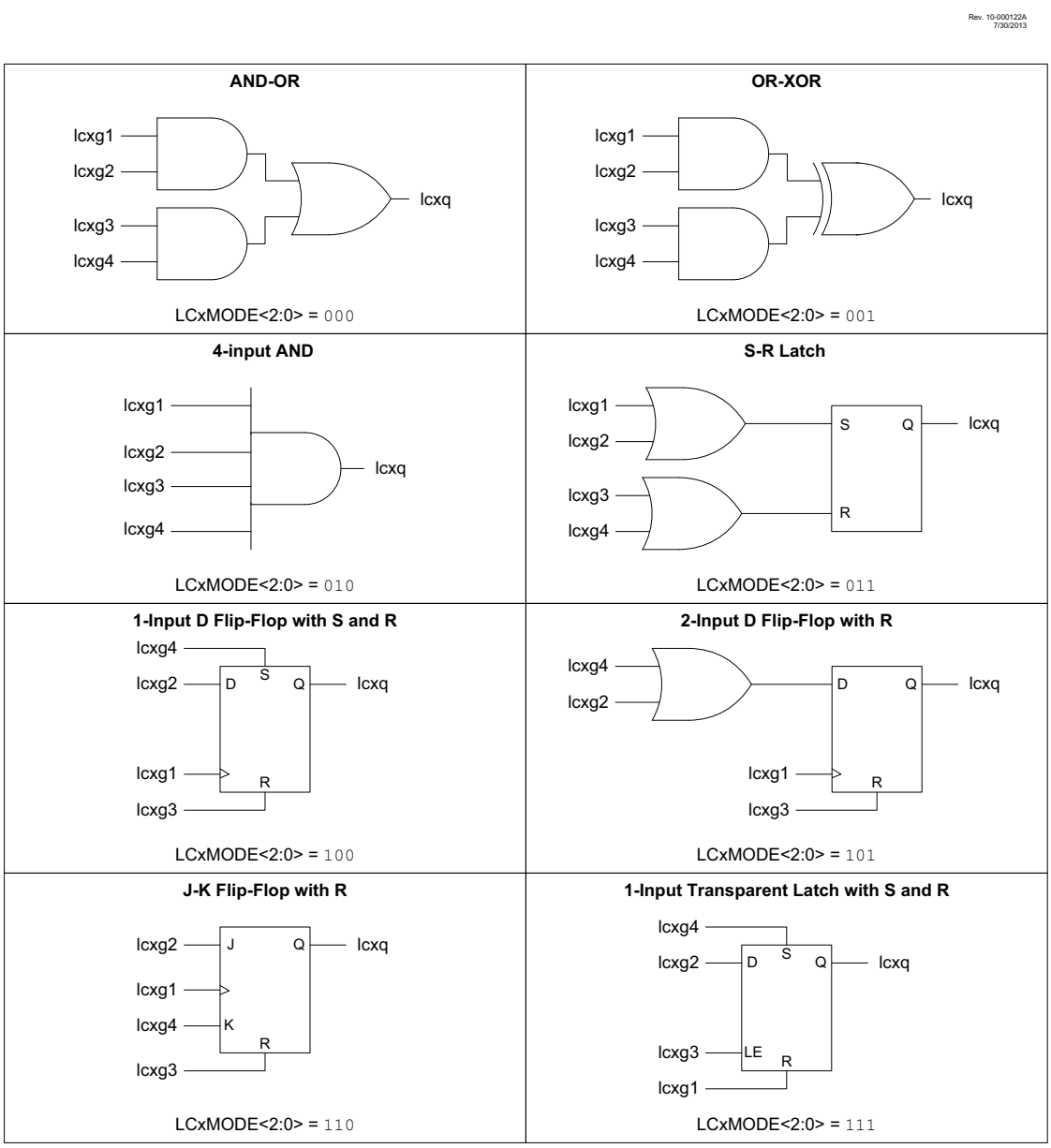
- Disable CLCx by clearing the LCxEN bit.
- Select desired inputs using CLCxSEL0 through CLCxSEL3 registers (See Table 20-1).
- Clear any associated ANSEL bits.
- Set all TRIS bits associated with inputs.
- Clear all TRIS bits associated with outputs.
- Enable the chosen inputs through the four gates using CLCxGLS0, CLCxGLS1, CLCxGLS2, and CLCxGLS3 registers.
- Select the gate output polarities with the LCxGyPOL 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 a device pin, set the desired pin PPS control 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 for falling event.
  - Set the CLCxIE bit of the PIE3 register.
  - Set the GIE and PEIE bits of the INTCON register.
- Enable the CLCx by setting the LCxEN bit of the CLCxCON register.

FIGURE 20-2: INPUT DATA SELECTION AND GATING



Note: All controls are undefined at power-up.

FIGURE 20-3: PROGRAMMABLE LOGIC FUNCTIONS



20.7 Register Definitions: CLC Control

REGISTER 20-1: CLCxCON: CONFIGURABLE LOGIC CELL CONTROL REGISTER

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

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

bit 4 **LCxINTP:** Configurable Logic Cell Positive Edge Going Interrupt Enable bit
1 = CLCxIF will be set when a rising edge occurs on CLCxOUT
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 CLCxOUT
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 20-2: CLCxPOL: SIGNAL POLARITY CONTROL REGISTER

R/W-0/0	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 Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **LCxPOL:** CLCxOUT Output 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 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 2 **LCxG3POL:** 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 1 **LCxG2POL:** 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

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

REGISTER 20-3: CLCxSEL0: GENERIC CLCx DATA 0 SELECT 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
—	—	LCxD1S<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	

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

bit 5-0 **LCxD1S<5:0>:** CLCx Data1 Input Selection bits
See Table 20-1.

REGISTER 20-4: CLCxSEL1: GENERIC CLCx DATA 1 SELECT 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
—	—	LCxD2S<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	

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

bit 5-0 **LCxD2S<5:0>:** CLCx Data 2 Input Selection bits
See Table 20-1.

REGISTER 20-5: CLCxSEL2: GENERIC CLCx DATA 2 SELECT 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
—	—	LCxD3S<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	

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

bit 5-0 **LCxD3S<5:0>:** CLCx Data 3 Input Selection bits
See Table 20-1.

REGISTER 20-6: CLCxSEL3: GENERIC CLCx DATA 3 SELECT 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
—	—	LCxD4S<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	

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **LCxD4S<5:0>:** CLCx Data 4 Input Selection bits
See Table 20-1.

REGISTER 20-7: CLCxGLS0: GATE 0 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 0 Data 4 True (non-inverted) bit
1 = CLCIN3 (true) is gated into CLCx Gate 0
0 = CLCIN3 (true) is not gated into CLCx Gate 0

bit 6 **LCxG1D4N:** Gate 0 Data 4 Negated (inverted) bit
1 = CLCIN3 (inverted) is gated into CLCx Gate 0
0 = CLCIN3 (inverted) is not gated into CLCx Gate 0

bit 5 **LCxG1D3T:** Gate 0 Data 3 True (non-inverted) bit
1 = CLCIN2 (true) is gated into CLCx Gate 0
0 = CLCIN2 (true) is not gated into CLCx Gate 0

bit 4 **LCxG1D3N:** Gate 0 Data 3 Negated (inverted) bit
1 = CLCIN2 (inverted) is gated into CLCx Gate 0
0 = CLCIN2 (inverted) is not gated into CLCx Gate 0

bit 3 **LCxG1D2T:** Gate 0 Data 2 True (non-inverted) bit
1 = CLCIN1 (true) is gated into CLCx Gate 0
0 = CLCIN1 (true) is not gated into CLCx Gate 0

bit 2 **LCxG1D2N:** Gate 0 Data 2 Negated (inverted) bit
1 = CLCIN1 (inverted) is gated into CLCx Gate 0
0 = CLCIN1 (inverted) is not gated into CLCx Gate 0

bit 1 **LCxG1D1T:** Gate 0 Data 1 True (non-inverted) bit
1 = CLCIN0 (true) is gated into CLCx Gate 0
0 = CLCIN0 (true) is not gated into CLCx Gate 0

bit 0 **LCxG1D1N:** Gate 0 Data 1 Negated (inverted) bit
1 = CLCIN0 (inverted) is gated into CLCx Gate 0
0 = CLCIN0 (inverted) is not gated into CLCx Gate 0

REGISTER 20-8: CLCxGLS1: 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
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 1 Data 4 True (non-inverted) bit
1 = CLCIN3 (true) is gated into CLCx Gate 1
0 = CLCIN3 (true) is not gated into CLCx Gate 1

bit 6 **LCxG2D4N:** Gate 1 Data 4 Negated (inverted) bit
1 = CLCIN3 (inverted) is gated into CLCx Gate 1
0 = CLCIN3 (inverted) is not gated into CLCx Gate 1

bit 5 **LCxG2D3T:** Gate 1 Data 3 True (non-inverted) bit
1 = CLCIN2 (true) is gated into CLCx Gate 1
0 = CLCIN2 (true) is not gated into CLCx Gate 1

bit 4 **LCxG2D3N:** Gate 1 Data 3 Negated (inverted) bit
1 = CLCIN2 (inverted) is gated into CLCx Gate 1
0 = CLCIN2 (inverted) is not gated into CLCx Gate 1

bit 3 **LCxG2D2T:** Gate 1 Data 2 True (non-inverted) bit
1 = CLCIN1 (true) is gated into CLCx Gate 1
0 = CLCIN1 (true) is not gated into CLCx Gate 1

bit 2 **LCxG2D2N:** Gate 1 Data 2 Negated (inverted) bit
1 = CLCIN1 (inverted) is gated into CLCx Gate 1
0 = CLCIN1 (inverted) is not gated into CLCx Gate 1

bit 1 **LCxG2D1T:** Gate 1 Data 1 True (non-inverted) bit
1 = CLCIN0 (true) is gated into CLCx Gate 1
0 = CLCIN0 (true) is not gated into CLCx Gate 1

bit 0 **LCxG2D1N:** Gate 1 Data 1 Negated (inverted) bit
1 = CLCIN0 (inverted) is gated into CLCx Gate 1
0 = CLCIN0 (inverted) is not gated into CLCx Gate 1

REGISTER 20-9: CLCxGLS2: 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
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 2 Data 4 True (non-inverted) bit
1 = CLCIN3 (true) is gated into CLCx Gate 2
0 = CLCIN3 (true) is not gated into CLCx Gate 2

bit 6 **LCxG3D4N:** Gate 2 Data 4 Negated (inverted) bit
1 = CLCIN3 (inverted) is gated into CLCx Gate 2
0 = CLCIN3 (inverted) is not gated into CLCx Gate 2

bit 5 **LCxG3D3T:** Gate 2 Data 3 True (non-inverted) bit
1 = CLCIN2 (true) is gated into CLCx Gate 2
0 = CLCIN2 (true) is not gated into CLCx Gate 2

bit 4 **LCxG3D3N:** Gate 2 Data 3 Negated (inverted) bit
1 = CLCIN2 (inverted) is gated into CLCx Gate 2
0 = CLCIN2 (inverted) is not gated into CLCx Gate 2

bit 3 **LCxG3D2T:** Gate 2 Data 2 True (non-inverted) bit
1 = CLCIN1 (true) is gated into CLCx Gate 2
0 = CLCIN1 (true) is not gated into CLCx Gate 2

bit 2 **LCxG3D2N:** Gate 2 Data 2 Negated (inverted) bit
1 = CLCIN1 (inverted) is gated into CLCx Gate 2
0 = CLCIN1 (inverted) is not gated into CLCx Gate 2

bit 1 **LCxG3D1T:** Gate 2 Data 1 True (non-inverted) bit
1 = CLCIN0 (true) is gated into CLCx Gate 2
0 = CLCIN0 (true) is not gated into CLCx Gate 2

bit 0 **LCxG3D1N:** Gate 2 Data 1 Negated (inverted) bit
1 = CLCIN0 (inverted) is gated into CLCx Gate 2
0 = CLCIN0 (inverted) is not gated into CLCx Gate 2

REGISTER 20-10: CLCxGLS3: 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
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 3 Data 4 True (non-inverted) bit
1 = CLCIN3 (true) is gated into CLCx Gate 3
0 = CLCIN3 (true) is not gated into CLCx Gate 3

bit 6 **LCxG4D4N:** Gate 3 Data 4 Negated (inverted) bit
1 = CLCIN3 (inverted) is gated into CLCx Gate 3
0 = CLCIN3 (inverted) is not gated into CLCx Gate 3

bit 5 **LCxG4D3T:** Gate 3 Data 3 True (non-inverted) bit
1 = CLCIN2 (true) is gated into CLCx Gate 3
0 = CLCIN2 (true) is not gated into CLCx Gate 3

bit 4 **LCxG4D3N:** Gate 3 Data 3 Negated (inverted) bit
1 = CLCIN2 (inverted) is gated into CLCx Gate 3
0 = CLCIN2 (inverted) is not gated into CLCx Gate 3

bit 3 **LCxG4D2T:** Gate 3 Data 2 True (non-inverted) bit
1 = CLCIN1 (true) is gated into CLCx Gate 3
0 = CLCIN1 (true) is not gated into CLCx Gate 3

bit 2 **LCxG4D2N:** Gate 3 Data 2 Negated (inverted) bit
1 = CLCIN1 (inverted) is gated into CLCx Gate 3
0 = CLCIN1 (inverted) is not gated into CLCx Gate 3

bit 1 **LCxG4D1T:** Gate 3 Data 1 True (non-inverted) bit
1 = CLCIN0 (true) is gated into CLCx Gate 3
0 = CLCIN0 (true) is not gated into CLCx Gate 3

bit 0 **LCxG4D1N:** Gate 3 Data 1 Negated (inverted) bit
1 = CLCIN0 (inverted) is gated into CLCx Gate 3
0 = CLCIN0 (inverted) is not gated into CLCx Gate 3

REGISTER 20-11: CLCDATA: CLC DATA OUTPUT

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT
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 **MLC4OUT:** Mirror copy of LC4OUT bit

bit 2 **MLC3OUT:** Mirror copy of LC3OUT bit

bit 1 **MLC2OUT:** Mirror copy of LC2OUT bit

bit 0 **MLC1OUT:** Mirror copy of LC1OUT bit

TABLE 20-3: SUMMARY OF REGISTERS ASSOCIATED WITH CLCx

Name	Bit7	Bit6	Bit5	Bit4	Blt3	Bit2	Bit1	Bit0	Register on Page
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	141
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
PIR3	OSFIF	CSWIF	TMR3GIF	TMR3IF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	107
PIE3	OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	102
CLC1CON	LC1EN	—	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>			225
CLC1POL	LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	226
CLC1SEL0	—	—	LC1D1S<5:0>						227
CLC1SEL1	—	—	LC1D2S<5:0>						227
CLC1SEL2	—	—	LC1D3S<5:0>						227
CLC1SEL3	—	—	LC1D4S<5:0>						228
CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	228
CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	229
CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	230
CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	231
CLC2CON	LC2EN	—	LC2OUT	LC2INTP	LC2INTN	LC2MODE<2:0>			225
CLC2POL	LC2POL	—	—	—	LC2G4POL	LC2G3POL	LC2G2POL	LC2G1POL	226
CLC2SEL0	—	—	LC2D1S<5:0>						227
CLC2SEL1	—	—	LC2D2S<5:0>						227
CLC2SEL2	—	—	LC2D3S<5:0>						227
CLC2SEL3	—	—	LC2D4S<5:0>						228

TABLE 20-3: SUMMARY OF REGISTERS ASSOCIATED WITH CLCx (CONTINUED)

Name	Bit7	Bit6	Bit5	Bit4	Blt3	Bit2	Bit1	Bit0	Register on Page
CLC2GLS0	LC2G1D4T	LC2G1D4N	LC2G1D3T	LC2G1D3N	LC2G1D2T	LC2G1D2N	LC2G1D1T	LC2G1D1N	228
CLC2GLS1	LC2G2D4T	LC2G2D4N	LC2G2D3T	LC2G2D3N	LC2G2D2T	LC2G2D2N	LC2G2D1T	LC2G2D1N	229
CLC2GLS2	LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1N	230
CLC2GLS3	LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1N	231
CLC3CON	LC3EN	—	LC3OUT	LC3INTP	LC3INTN	LC3MODE<2:0>			225
CLC3POL	LC3POL	—	—	—	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1POL	226
CLC3SEL0	—	—	LC3D1S<5:0>						227
CLC3SEL1	—	—	LC3D2S<5:0>						227
CLC3SEL2	—	—	LC3D3S<5:0>						227
CLC3SEL3	—	—	LC3D4S<5:0>						228
CLC3GLS0	LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1N	228
CLC3GLS1	LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N	229
CLC3GLS2	LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N	230
CLC3GLS3	LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N	231
CLC4CON	LC4EN	—	LC4OUT	LC4INTP	LC4INTN	LC4MODE<2:0>			225
CLC4POL	LC4POL	—	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL	226
CLC4SEL0	—	—	LC4D1S<5:0>						227
CLC4SEL1	—	—	LC4D2S<5:0>						227
CLC4SEL2	—	—	LC4D3S<5:0>						227
CLC4SEL3	—	—	LC4D4S<5:0>						228
CLC4GLS0	LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N	228
CLC4GLS1	LC4G2D4T	LC4G2D4N	LC4G2D3T	LC4G2D3N	LC4G2D2T	LC4G2D2N	LC4G2D1T	LC4G2D1N	229
CLC4GLS2	LC4G3D4T	LC4G3D4N	LC4G3D3T	LC4G3D3N	LC4G3D2T	LC4G3D2N	LC4G3D1T	LC4G3D1N	230
CLC4GLS3	LC4G4D4T	LC4G4D4N	LC4G4D3T	LC4G4D3N	LC4G4D2T	LC4G4D2N	LC4G4D1T	LC4G4D1N	231
CLCDATA	—	—	—	—	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT	232
CLCIN0PPS	—	—	—	CLCIN0PPS<4:0>					160
CLCIN1PPS	—	—	—	CLCIN1PPS<4:0>					160
CLCIN2PPS	—	—	—	CLCIN2PPS<4:0>					160
CLCIN3PPS	—	—	—	CLCIN3PPS<4:0>					160
CLC1OUTPPS	—	—	—	CLC1OUTPPS<4:0>					160
CLC2OUTPPS	—	—	—	CLC2OUTPPS<4:0>					160
CLC3OUTPPS	—	—	—	CLC3OUTPPS<4:0>					160
CLC4OUTPPS	—	—	—	CLC4OUTPPS<4:0>					160

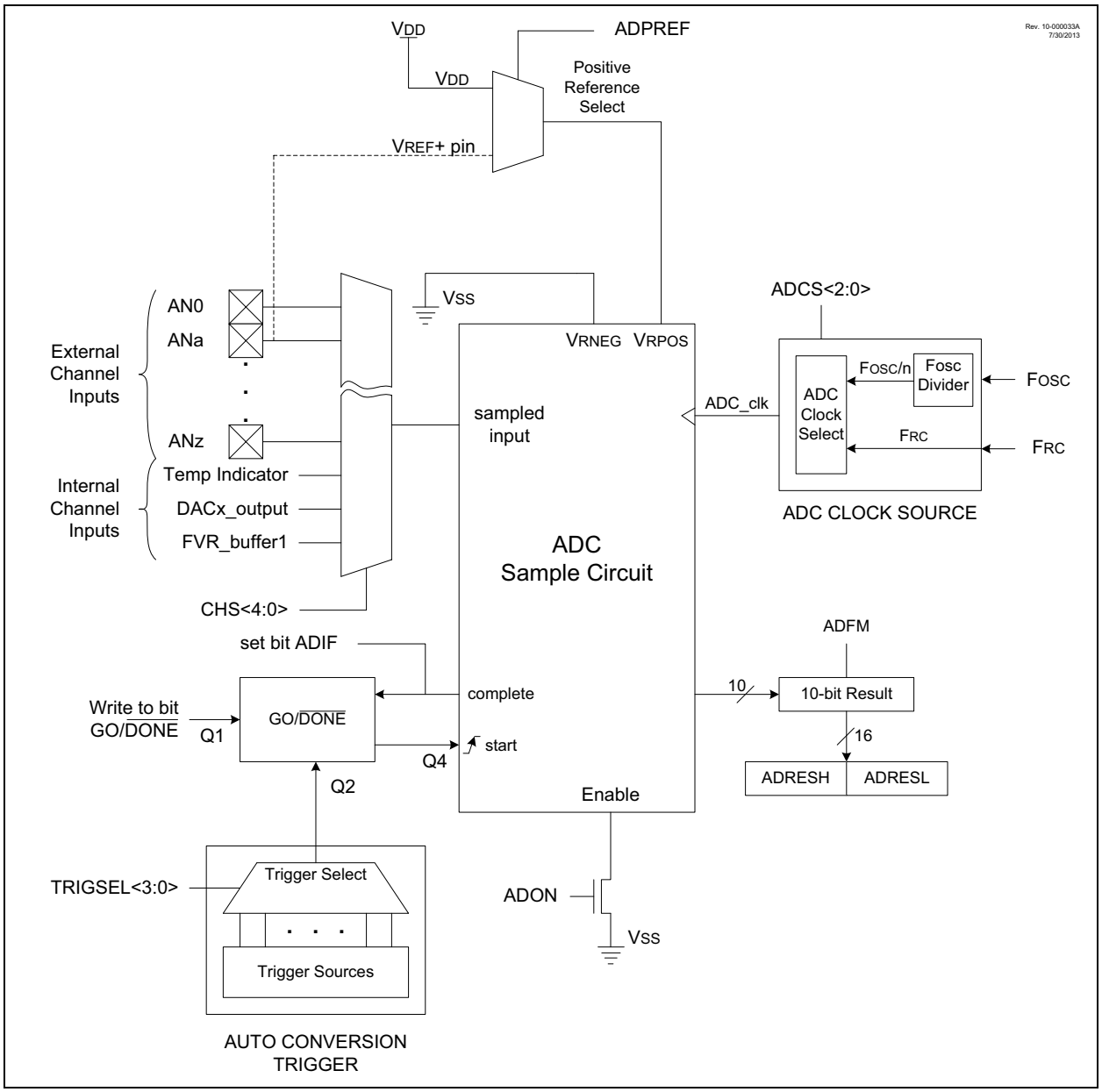
21.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, 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 a 10-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADRESH:ADRESL register pair). Figure 21-1 shows the block diagram of the ADC.

The ADC voltage reference is software selectable to be either internally generated or externally supplied.

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

FIGURE 21-1: ADC BLOCK DIAGRAM



21.1 ADC Configuration

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

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Result formatting

21.1.1 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 11.0 "I/O Ports"** 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.

21.1.2 CHANNEL SELECTION

There are several channel selections available:

- Five PORTA pins (RA0-RA2, RA4-RA5)
- Four PORTB pins (RB4-RB7, PIC16(L)F18345 only)
- Six PORTC pins (RC0-RC5, PIC16(L)F18325)
- Eight PORTC pins (RC0-RC7, PIC16(L)F18345 only)
- Temperature Indicator
- DAC output
- Fixed Voltage Reference (FVR)
- AVSS (ground)

The CHS<5:0> bits of the ADCON0 register (Register 21-1) 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 21.2 "ADC Operation"** for more information.

21.1.3 ADC VOLTAGE REFERENCE

The ADPREF<1:0> bits of the ADCON1 register provides control of the positive voltage reference. The positive voltage reference can be:

- VREF+ pin
- VDD
- FVR 2.048V
- FVR 4.096V (Not available on LF devices)

The ADNREF bit of the ADCON1 register provides control of the negative voltage reference. The negative voltage reference can be:

- VREF- pin
- VSS

See **Section 21.0 "Analog-to-Digital Converter (ADC) Module"** for more details on the Fixed Voltage Reference.

21.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS<2:0> bits of the ADCON1 register. There are seven possible clock options:

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

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11.5 TAD periods as shown in Figure 21-2.

For correct conversion, the appropriate TAD specification must be met. Refer to Table 34-13 for more information. Table 21-1 gives examples of appropriate ADC clock selections.

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

TABLE 21-1: ADC CLOCK PERIOD (T_{AD}) VS. DEVICE OPERATING FREQUENCIES

ADC Clock Period (T_{AD})		Device Frequency (F_{OSC})					
ADC Clock Source	ADCS<2:0>	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
F_{OSC}/2	000	62.5ns(2)	100 ns(2)	125 ns(2)	250 ns(2)	500 ns(2)	2.0 μs
F_{OSC}/4	100	125 ns(2)	200 ns(2)	250 ns(2)	500 ns(2)	1.0 μs	4.0 μs
F_{OSC}/8	001	0.5 μs(2)	400 ns(2)	0.5 μs(2)	1.0 μs	2.0 μs	8.0 μs(3)
F_{OSC}/16	101	800 ns	800 ns	1.0 μs	2.0 μs	4.0 μs	16.0 μs(3)
F_{OSC}/32	010	1.0 μs	1.6 μs	2.0 μs	4.0 μs	8.0 μs(3)	32.0 μs(2)
F_{OSC}/64	110	2.0 μs	3.2 μs	4.0 μs	8.0 μs(3)	16.0 μs(2)	64.0 μs(2)
ADCRC	x11	1.0-6.0 μs(1,4)	1.0-6.0 μs(1,4)	1.0-6.0 μs(1,4)	1.0-6.0 μs(1,4)	1.0-6.0 μs(1,4)	1.0-6.0 μs(1,4)

Legend: Shaded cells are outside of recommended range.

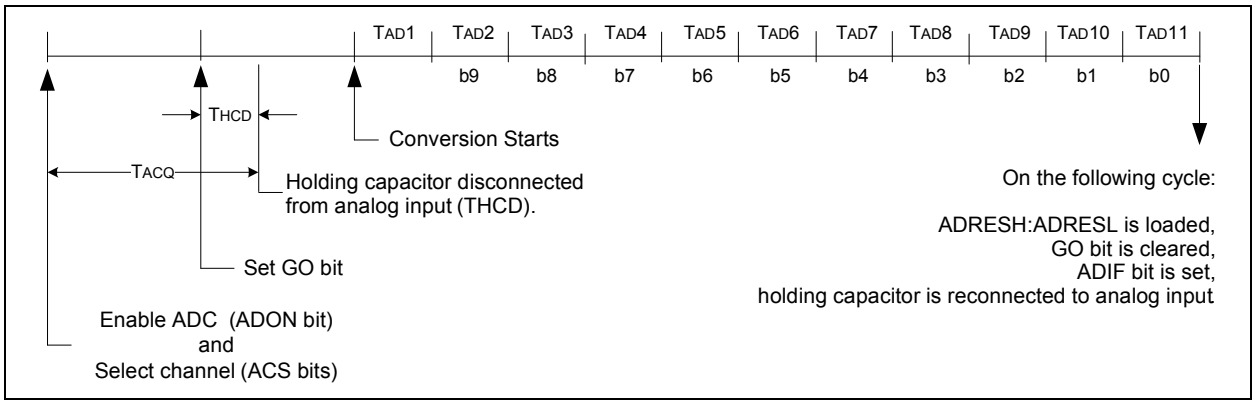
Note 1: See T_{AD} parameter for ADCRC source typical T_{AD} value.

2: These values violate the required T_{AD} time.

3: Outside the recommended T_{AD} time.

4: The ADC clock period (T_{AD}) and total ADC conversion time can be minimized when the ADC clock is derived from the system clock F_{OSC}. However, the ADCRC oscillator source must be used when conversions are to be performed with the device in Sleep mode.

FIGURE 21-2: ANALOG-TO-DIGITAL CONVERSION T_{AD} CYCLES



21.1.5 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 1:	The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled.
2:	The ADC operates during Sleep only when the ADCRC oscillator is selected.

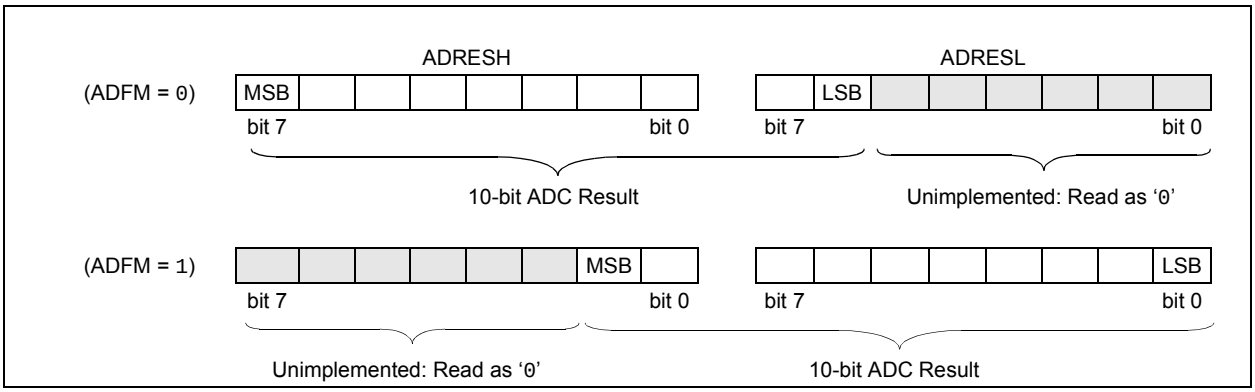
This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the SLEEP instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the ADIE bit of the PIE1 register and the PEIE bit of the INTCON register must both be set and the GIE bit of the INTCON register must be cleared. If all three of these bits are set, the execution will switch to the Interrupt Service Routine.

21.1.6 RESULT FORMATTING

The 10-bit ADC conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON1 register controls the output format.

Figure 21-3 shows the two output formats.

FIGURE 21-3: 10-BIT ADC CONVERSION RESULT FORMAT



21.2 ADC Operation

21.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/DONE bit of the ADCON0 register to a '1' will start the Analog-to-Digital conversion.

Note:	The GO/DONE bit should not be set in the same instruction that turns on the ADC. Refer to **Section 21.2.6 "ADC Conversion Procedure"**.

21.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/DONE bit
- Set the ADIF Interrupt Flag bit
- Update the ADRESH and ADRESL registers with new conversion result

21.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/DONE bit can be cleared in software. The ADRESH and ADRESL registers 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.

21.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the ADCRC option. When the ADCRC oscillator 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 ADCRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

21.2.5 AUTO-CONVERSION TRIGGER

The Auto-conversion Trigger allows periodic ADC measurements without software intervention. When a rising edge of the selected source occurs, the GO/DONE bit is set by hardware.

The Auto-conversion Trigger source is selected with the ADACT<4:0> bits of the ADACT register.

Using the Auto-conversion Trigger does not assure proper ADC timing. It is the user's responsibility to ensure that the ADC timing requirements are met.

See Table 21-2 for auto-conversion sources.

TABLE 21-2: ADC AUTO-CONVERSION TABLE

Source Peripheral	Description
TMR0	Timer0 Overflow condition
TMR1	Timer1 Overflow condition
TMR3	Timer3 Overflow condition
TMR5	Timer5 Overflow condition
TMR2	Match between Timer2 and PR2
TMR4	Match between Timer4 and PR4
TMR6	Match between Timer6 and PR6
C1	Comparator C1 output
C2	Comparator C2 output
CLC1	CLC1 output
CLC2	CLC2 output
CLC3	CLC3 output
CLC4	CLC4 output
CCP1	CCP1 output
CCP2	CCP2 output
CCP3	CCP3 output
CCP4	CCP4 output

21.2.6 ADC 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
   - Configure voltage reference
   - 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 21.3 "ADC Acquisition Requirements"**.

EXAMPLE 21-1: ADC CONVERSION

```
;This code block configures the ADC
;for polling, Vdd and Vss references, ADCRC
;oscillator and AN0 input.
;
;Conversion start & polling for completion ;
are included.
;
BANKSEL    ADCON1        ;
MOVLW      B'11110000'   ;Right justify, ADCRC
                         ;oscillator
MOVWF      ADCON1        ;Vdd and Vss Vref
BANKSEL    TRISA         ;
BSF        TRISA,0       ;Set RA0 to input
BANKSEL    ANSEL         ;
BSF        ANSEL,0       ;Set RA0 to analog
BANKSEL    ADCON0        ;
MOVLW      B'00000001'   ;Select channel AN0
MOVWF      ADCON0        ;Turn ADC On
CALL       SampleTime    ;Acquisiton delay
BSF        ADCON0,ADGO   ;Start conversion
BTFSC      ADCON0,ADGO   ;Is conversion done?
GOTO       $-1           ;No, test again
BANKSEL    ADRESH        ;
MOVF       ADRESH,W      ;Read upper 2 bits
MOVWF      RESULTHI      ;store in GPR space
BANKSEL    ADRESL        ;
MOVF       ADRESL,W      ;Read lower 8 bits
MOVWF      RESULTLO      ;Store in GPR space
```

21.3 ADC 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 21-4. 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 21-4. **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 ADC acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 21-1 may be used. This equation assumes that 1/2 LSb error is used (1,024 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

EQUATION 21-1: ACQUISITION TIME EXAMPLE

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

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

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

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

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

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

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

Solving for T_C*:*

$$T_C = -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ ln(1/2047)$$
$$= -10pF(1k\Omega + 7k\Omega + 10k\Omega)\ ln(0.0004885)$$
$$= 1.37\mu s$$

Therefore:

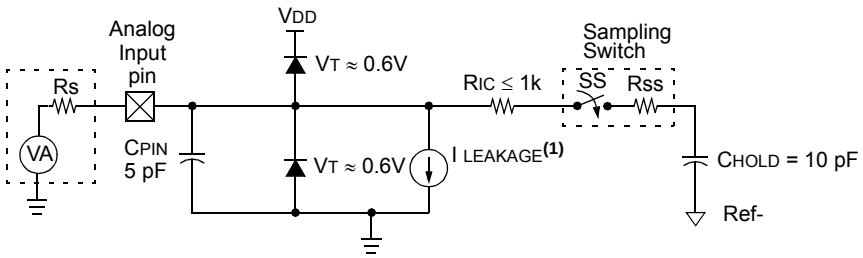
$$T_{ACQ} = 2\mu s + 892ns + [(50°C - 25°C)(0.05\mu s/°C)]$$
$$= 4.62\mu s$$

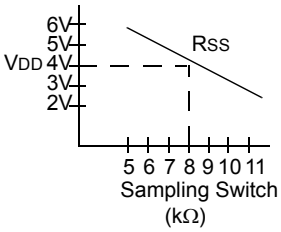
Note 1: The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

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 21-4: ANALOG INPUT MODEL



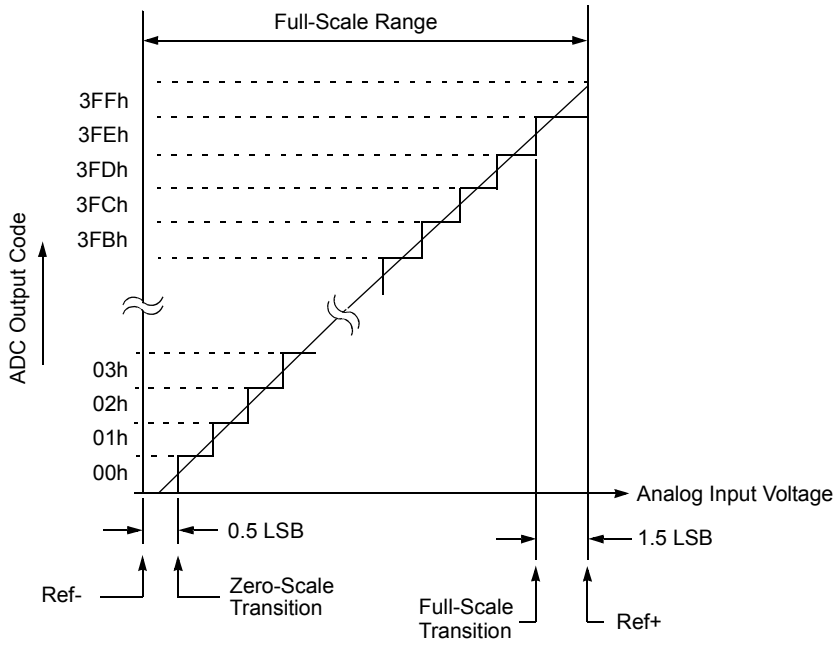


Legend:

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

Note 1: Refer to Table 34-4 (parameter D060).

FIGURE 21-5: ADC TRANSFER FUNCTION



21.4 Register Definitions: ADC Control

REGISTER 21-1: ADCON0: ADC 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
CHS<5:0>						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 7-2 **CHS<5:0>:** Analog Channel Select bits

111111 = FVR (Fixed Voltage Reference)(2)
111110 = DAC1 output(1)
111101 = Temperature Indicator(3)
111100 = AVSS (Analog Ground)
111011 = Reserved. No channel connected.
•
•
•
010111 = ANC7(4)
010110 = ANC6(4)
010101 = ANC5
010100 = ANC4
010011 = ANC3
010010 = ANC2
010001 = ANC1
010000 = ANC0
001111 = ANB7(4)
001110 = ANB6(4)
001101 = ANB5(4)
001100 = ANB4(4)
001011 = Reserved. No channel connected.
•
•
•
000101 = ANA5
000100 = ANA4
000011 = Reserved. No channel connected.
000010 = ANA2
000001 = ANA1
000000 = ANA0

bit 1 **GO/$\overline{\text{DONE}}$:** ADC Conversion Status bit

1 = ADC conversion cycle in progress. Setting this bit starts an ADC conversion cycle.
This bit is automatically cleared by hardware when the ADC conversion has completed.
0 = ADC conversion completed/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 23.0 "5-bit Digital-to-Analog Converter (DAC1) Module"** for more information.
2: See **Section 15.0 "Fixed Voltage Reference (FVR)"** for more information.
3: See **Section 16.0 "Temperature Indicator Module"** for more information.
4: PIC16(L)F18345 only.

REGISTER 21-2: ADCON1: ADC CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
ADFM	ADCS<2:0>			—	ADNREF	ADPREF<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 **ADFM:** ADC Result Format Select bit
1 = Right justified. Six Most Significant bits of ADRESH are set to '0' when the conversion result is loaded.
0 = Left justified. Six Least Significant bits of ADRESL are set to '0' when the conversion result is loaded.

bit 6-4 **ADCS<2:0>:** ADC Conversion Clock Select bits
111 = ADCRC (dedicated RC oscillator)
110 = FOSC/64
101 = FOSC/16
100 = FOSC/4
011 = ADCRC (dedicated RC oscillator)
010 = FOSC/32
001 = FOSC/8
000 = FOSC/2

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

bit 2 **ADNREF:** A/D Negative Voltage Reference Configuration bit
When ADON = 0, all multiplexer inputs are disconnected.
0 = VREF- is connected to AVSS
1 = VREF- is connected to external VREF-

bit 1-0 **ADPREF<1:0>:** ADC Positive Voltage Reference Configuration bits
11 = VREF+ is connected to internal Fixed Voltage Reference (FVR) module(1)
10 = VREF+ is connected to external VREF+ pin(1)
01 = Reserved
00 = VREF+ is connected to VDD

Note 1: When selecting the VREF+ pin as the source of the positive reference, be aware that a minimum voltage specification exists. See Table 34-13 for details.

REGISTER 21-3: ADACT: A/D AUTO-CONVERSION TRIGGER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	ADACT<4:0>				
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 **Unimplemented:** Read as '0'

bit 4-0 **ADACT<4:0>:** Auto-Conversion Trigger Selection bits(1)

10001 = Timer5 overflow(2)
10000 = Timer3 overflow(2)
1111 = CCP4
1110 = CCP3
1101 = CCP2
1100 = CCP1
1011 = CLC4
1010 = CLC3
1001 = CLC2
1000 = CLC1
0111 = Comparator C2
0110 = Comparator C1
0101 = Timer2-PR2 match
0100 = Timer1 overflow(2)
0011 = Timer0 overflow(2)
0010 = Timer6-PR6 match
0001 = Timer4-PR4 match
0000 = No auto-conversion trigger selected

Note 1: This is a rising edge sensitive input for all sources.

2: Trigger corresponds to when the peripheral's interrupt flag is set.

REGISTER 21-4: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 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	R/W-x/u	R/W-x/u
ADRES<9:2>							
bit 7							bit 0

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

bit 7-0 **ADRES<9:2>**: ADC Result Register bits
Upper eight bits of 10-bit conversion result

REGISTER 21-5: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 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	R/W-x/u	R/W-x/u
ADRES<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 7-6 **ADRES<1:0>**: ADC Result Register bits
Lower two bits of 10-bit conversion result

bit 5-0 **Reserved**: Do not use.

REGISTER 21-6: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1

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<9:8>	
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 7-2 **Reserved**: Do not use.

bit 1-0 **ADRES<9:8>**: ADC Result Register bits
Upper two bits of 10-bit conversion result

REGISTER 21-7: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 1

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
Lower eight bits of 10-bit conversion result

TABLE 21-3: 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
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	100
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	105
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	141
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
ADCON0	CHS<5:0>						GO/$\overline{\text{DONE}}$	ADON	242
ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		243
ADACT	—	—	—	ADACT<4:0>					244
ADRESH	ADRESH<7:0>								245
ADRESL	ADRESL<7:0>								245
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		178
DAC1CON1	—	—	—	DAC1R<4:0>					262
OSCSTAT1	EXTOR	HFOR	—	LFOR	SOR	ADOR	—	PLLR	88

Legend: – = unimplemented read as '0'. Shaded cells are not used for the ADC module.

Note 1: PIC16(L)F18345 only.

2: Unimplemented, read as '1'.

22.0 NUMERICALLY CONTROLLED OSCILLATOR (NCO1) MODULE

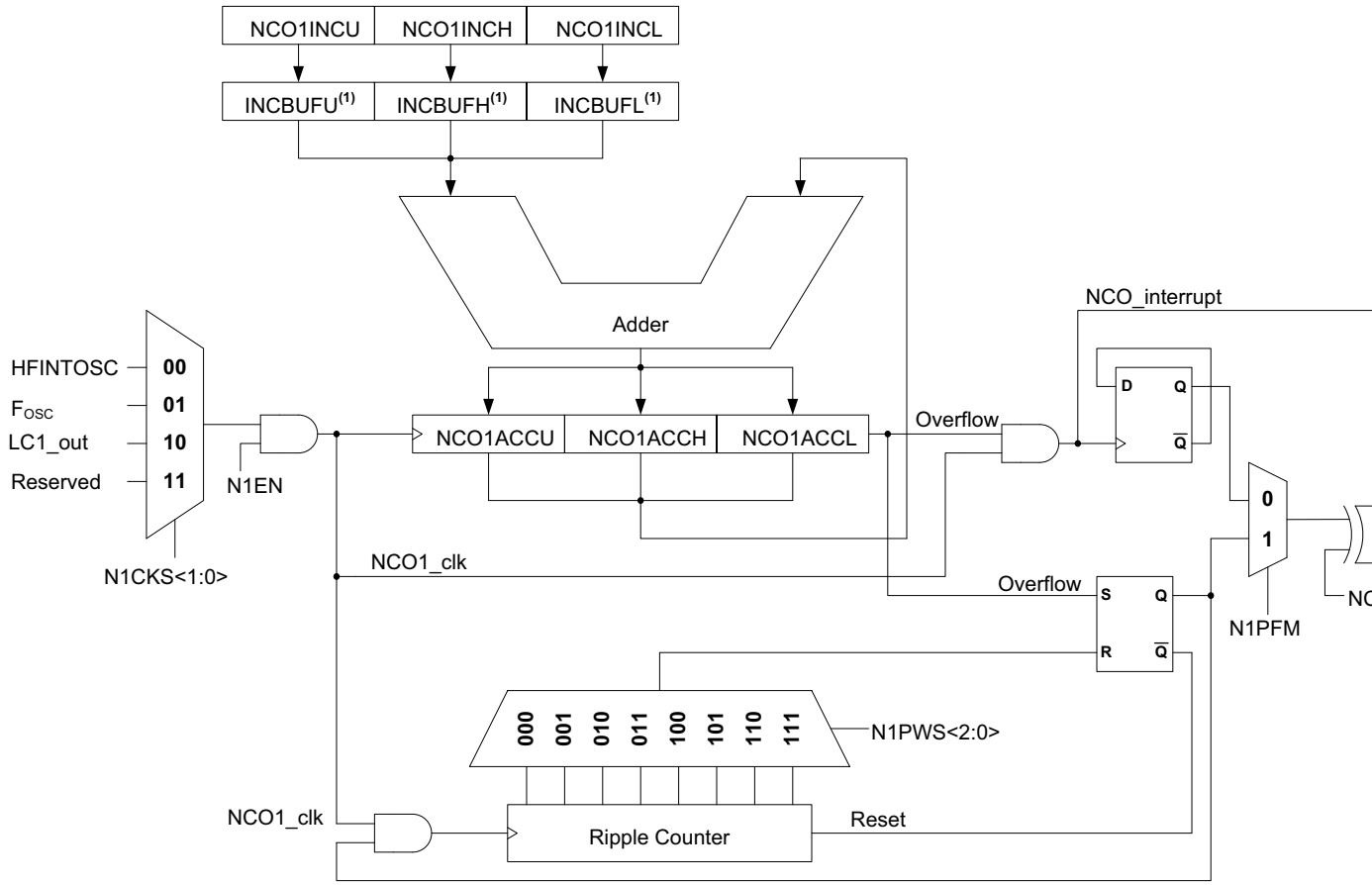
The Numerically Controlled Oscillator (NCO1) module is a timer that uses the overflow from the addition of an increment value to divide the input frequency. The advantage of the addition method over simple counter-driven timer is that the output frequency resolution does not vary with the divider value. The NCO1 is most useful for applications that require frequency accuracy and fine resolution at a fixed duty cycle.

Features of the NCO1 include:

- 20-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 22-1 is a simplified block diagram of the NCO1 module.

FIGURE 22-1: NUMERICALLY CONTROLLED OSCILLATOR MODULE SIMPLIFIED BLOCK DIAGRAM



Note 1: The increment registers are double-buffered to allow for value changes to be made without first disabling the NCO1 mo reference only and are not user accessible.

22.1 NCO1 Operation

The NCO1 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 NCO1 output (NCO_overflow). This effectively reduces the input clock by the ratio of the addition value to the maximum accumulator value. See Equation 22-1.

The NCO1 output can be further modified by stretching the pulse or toggling a flip-flop. The modified NCO1 output is then distributed internally to other peripherals and can optionally be output to a pin. The accumulator overflow also generates an interrupt (NCO_interrupt).

The NCO1 period changes in discrete steps to create an average frequency. This output depends on the ability of the receiving circuit (i.e., CWG or external resonant converter circuitry) to average the NCO1 output to reduce uncertainty.

EQUATION 22-1:

$$F_{OVERFLOW} = \frac{NCO1\ Clock\ Frequency \times Increment\ Value}{2^{20}}$$

22.1.1 NCO1 CLOCK SOURCES

Clock sources available to the NCO1 include:

- HFINTOSC
- FOSC
- LC1_out

The NCO1 clock source is selected by configuring the N1CKS<2:0> bits in the NCO1CLK register.

22.1.2 ACCUMULATOR

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

- NCO1ACCL
- NCO1ACCH
- NCO1ACCU

22.1.3 ADDER

The NCO1 adder is a full adder, which operates independently from the system clock. The addition of the previous result and the increment value replaces the accumulator value on the rising edge of each input clock.

22.1.4 INCREMENT REGISTERS

The increment value is stored in three registers making up a 20-bit increment. In order of LSB to MSB they are:

- NCO1INCL
- NCO1INCH
- NCO1INCU

When the NCO1 module is enabled, the NCO1INCU and NCO1INCH registers should be written first, then the NCO1INCL register. Writing to the NCO1INCL register initiates the increment buffer registers to be loaded simultaneously on the second rising edge of the NCO_clk signal.

The registers are readable and writable. The increment registers are double-buffered to allow value changes to be made without first disabling the NCO1 module.

When the NCO1 module is disabled, the increment buffers are loaded immediately after a write to the increment registers.

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

22.2 Fixed Duty Cycle (FDC) Mode

In Fixed Duty Cycle (FDC) mode, every time the accumulator overflows (NCO_overflow), the output is toggled. This provides a 50% duty cycle, provided that the increment value remains constant. For more information, see Figure 22-2.

The FDC mode is selected by clearing the N1PFM bit in the NCO1CON register.

22.3 Pulse Frequency (PF) Mode

In Pulse Frequency (PF) mode, every time the accumulator overflows (NCO_overflow), the output becomes active for one or more clock periods. 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 22-2.

The value of the active and inactive states depends on the polarity bit, N1POL, in the NCO1CON register.

The PF mode is selected by setting the N1PFM bit in the NCO1CON register.

22.3.1 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 N1PWS<2:0> bits in the NCO1CLK register.

When the selected pulse width is greater than the accumulator overflow time frame, the output of the NCO1 operation is indeterminate.

22.4 Output Polarity Control

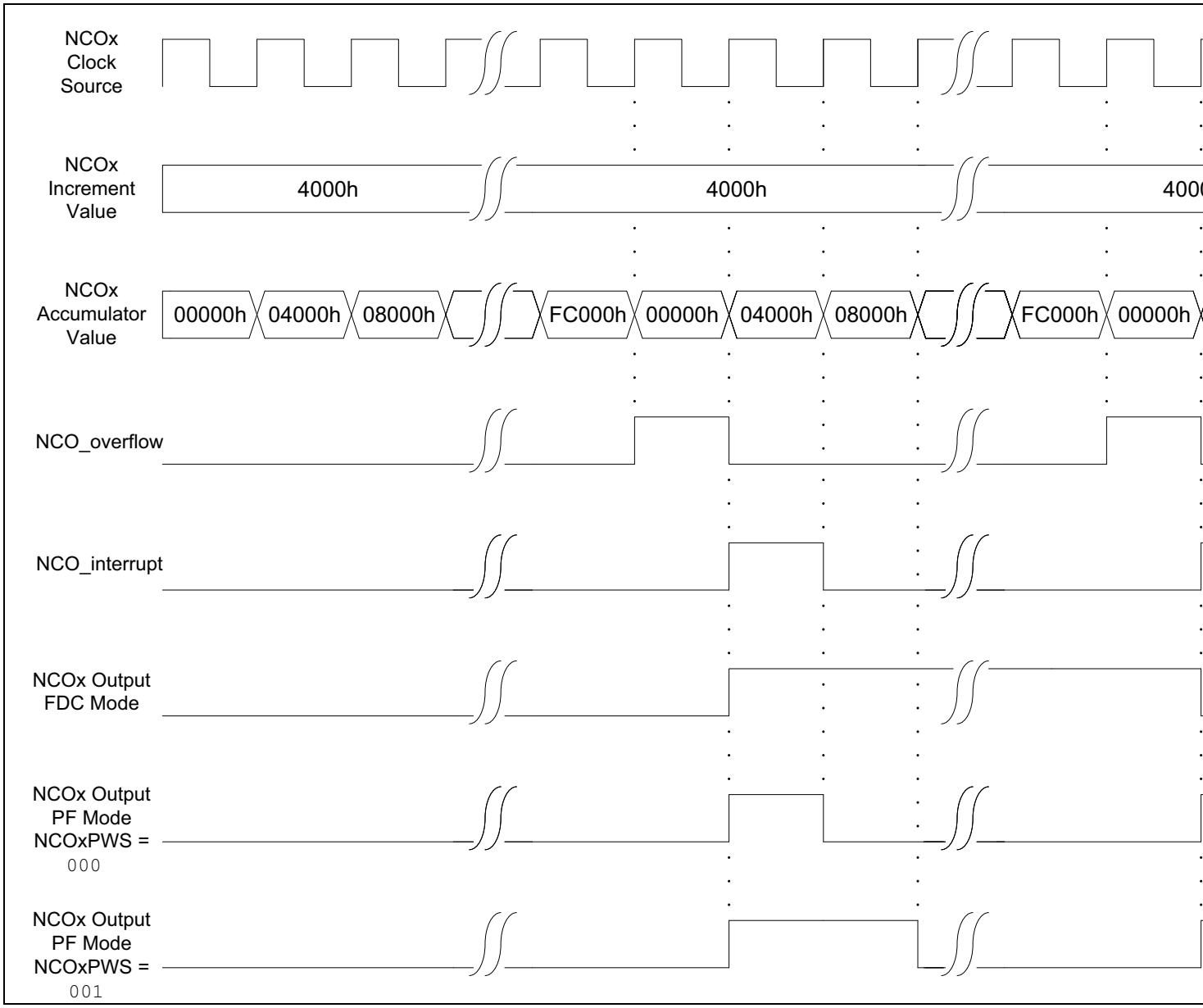
The last stage in the NCO1 module is the output polarity. The N1POL bit in the NCO1CON register selects the output polarity. Changing the polarity while the interrupts are enabled will cause an interrupt for the resulting output transition.

The NCO1 output can be used internally by source code or other peripherals. Accomplish this by reading the N1OUT (read-only) bit of the NCO1CON register.

The NCO1 output signal is available to the following peripherals:

- CLC
- CWG

FIGURE 22-2: FDC OUTPUT MODE OPERATION DIAGRAM



22.5 Interrupts

When the accumulator overflows (NCO_overflow), the NCO1 Interrupt Flag bit, NCO1IF, of the PIR2 register is set. To enable the interrupt event (NCO_interrupt), the following bits must be set:

- N1EN bit of the NCO1CON register
- NCO1IE bit of the PIE2 register
- PEIE bit of the INTCON register
- GIE bit of the INTCON register

The interrupt must be cleared by software by clearing the NCO1IF bit in the Interrupt Service Routine.

22.6 Effects of a Reset

All of the NCO1 registers are cleared to zero as the result of a Reset.

22.7 Operation in Sleep

The NCO1 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 NCO1 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 NCO1 clock source, when the NCO1 is enabled, the CPU will go idle during Sleep, but the NCO1 will continue to operate and the HFINTOSC will remain active.

This will have a direct effect on the Sleep mode current.

22.8 NCO1 Control Registers

REGISTER 22-1: NCO1CON: NCO1 CONTROL REGISTER

R/W-0/0	U-0	R-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
N1EN	—	N1OUT	N1POL	—	—	—	N1PFM
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 **N1EN:** NCO1 Enable bit
1 = NCO1 module is enabled
0 = NCO1 module is disabled

bit 6 **Unimplemented**: Read as ‘0’.

bit 5 **N1OUT:** NCO1 Output bit
Displays the current output value of the NCO1 module

bit 4 **N1POL:** NCO1 Polarity
1 = NCO1 output signal is inverted
0 = NCO1 output signal is not inverted

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

bit 0 **N1PFM:** NCO1 Output Divider mode
1 = NCO1 operates in Pulse Frequency mode
0 = NCO1 operates in Fixed Duty Cycle mode, divide by 2

REGISTER 22-2: NCO1CLK: NCO1 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
N1PWS<2:0>			—	—	—	N1CKS<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 **N1PWS<2:0>:** NCO1 Output Pulse Width Select(1, 2)
- 000 = NCO1 output is active for 1 input clock period
- 001 = NCO1 output is active for 2 input clock periods
- 010 = NCO1 output is active for 4 input clock periods
- 011 = NCO1 output is active for 8 input clock periods
- 100 = NCO1 output is active for 16 input clock periods
- 101 = NCO1 output is active for 32 input clock periods
- 110 = NCO1 output is active for 64 input clock periods
- 111 = NCO1 output is active for 128 input clock periods

bit 4-2 **Unimplemented:** Read as ‘0’

bit 1-0 **N1CKS<1:0>:** NCO1 Clock Source Select bits
- 00 = HFINTOSC (16 MHz)
- 01 = FOSC
- 10 = CLC1OUT
- 11 =Reserved

Note 1: N1PWS applies only when operating in Pulse Frequency mode.

2: If NCO1 pulse width is greater than NCO1 overflow period, operation is undefined.

REGISTER 22-3: NCO1ACCL: NCO1 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
NCO1ACC<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 **NCO1ACC<7:0>:** NCO1 Accumulator, low byte

REGISTER 22-4: NCO1ACCH: NCO1 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
NCO1ACC<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 **NCO1ACC<15:8>:** NCO1 Accumulator, high byte

REGISTER 22-5: NCO1ACCU: NCO1 ACCUMULATOR REGISTER – UPPER BYTE(1)

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	NCO1ACC<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 **NCO1ACC<19:16>:** NCO1 Accumulator, upper byte

Note 1: The accumulator spans registers NCO1ACCU:NCO1ACCH:NCO1ACCL. The 24 bits are reserved but not all are used.This register updates in real-time, asynchronously to the CPU; there is no provision to guarantee atomic access to this 24-bit space using an 8-bit bus. Writing to this register while the module is operating will produce undefined results.

REGISTER 22-6: NCO1INCL(1,2): NCO1 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
NCO1INC<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 **NCO1INC<7:0>:** NCO1 Increment, low byte

Note 1: The logical increment spans NCO1INCU:NCO1INCH:NCO1INCL.

2: NCO1INC is double-buffered as INCBUF; INCBUF is updated on the next falling edge of NCOCLK after writing to NCO1INCL;NCO1INCU and NCO1INCH should be written prior to writing NCO1INCL.

REGISTER 22-7: NCO1INCH(1): NCO1 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
NCO1INC<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 **NCO1INC<15:8>:** NCO1 Increment, high byte

Note 1: The logical increment spans NCO1INCU:NCO1INCH:NCO1INCL.

REGISTER 22-8: NCO1INCU(1): NCO1 INCREMENT 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
—	—	—	—	NCO1INC<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 **NCO1INC<19:16>:** NCO1 Increment, upper byte

Note 1: The logical increment spans NCO1INCU:NCO1INCH:NCO1INCL.

TABLE 22-1: SUMMARY OF REGISTERS ASSOCIATED WITH NCO1

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
PIR2	TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BCL2IF	TMR4IF	NCO1IF	106
PIE2	TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BCL2IE	TMR4IE	NCO1IE	101
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
NCO1CON	N1EN	—	N1OUT	N1POL	—	—	—	N1PFM	254
NCO1CLK	N1PWS<2:0>			—	—	—	N1CKS<1:0>		255
NCO1ACCL	NCO1ACC <7:0>								255
NCO1ACCH	NCO1ACC <15:8>								256
NCO1ACCU	—	—	—	—	NCO1ACC <19:16>				256
NCO1INCL	NCO1INC<7:0>								256
NCO1INCH	NCO1INC<15:8>								257
NCO1INCU	—	—	—	—	NCO1INC<19:16>				257
RxyPPS	—	—	—	RxyPPS<4:0>					161
CWG1DAT	—	—	—	—	DAT<3:0>				213
MDSRC	—	—	—	—	MDMS<3:0>				270
MDCARH	—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>				271
MDCARL	—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				272
CCPxCAP	—	—	—	—	CCPxCTS<3:0>				308

Legend: – = unimplemented read as '0'. Shaded cells are not used for NCO1 module.

Note 1: PIC16(L)F18345 only.

2: Unimplemented, read as '1'.

23.0 5-BIT DIGITAL-TO-ANALOG CONVERTER (DAC1) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 32 selectable output levels.

The input of the DAC can be connected to:

- External VREF pins
- VDD supply voltage
- FVR (Fixed Voltage Reference)

The output of the DAC can be configured to supply a reference voltage to the following:

- Comparator positive input
- ADC input channel
- DAC1OUT pin

The Digital-to-Analog Converter (DAC) is enabled by setting the DAC1EN bit of the DAC1CON0 register.

23.1 Output Voltage Selection

The DAC has 32 voltage level ranges. The 32 levels are set with the DAC1R<4:0> bits of the DAC1CON1 register.

The DAC output voltage is determined by Equation 23-1:

EQUATION 23-1: DAC OUTPUT VOLTAGE

$$V_{OUT} = \left((V_{SOURCE+}) - (V_{SOURCE-}) \times \frac{DAC1R\langle 4:0\rangle}{2^5} \right) + (V_{SOURCE-})$$

$$V_{SOURCE+} = V_{DD} \text{ or } V_{REF+} \text{ or } FVR$$

$$V_{SOURCE-} = V_{SS} \text{ or } V_{REF-}$$

23.2 Ratiometric Output Level

The DAC output value is derived using a resistor ladder with each end of the ladder tied to a positive and negative voltage reference input source. If the voltage of either input source fluctuates, a similar fluctuation will result in the DAC output value.

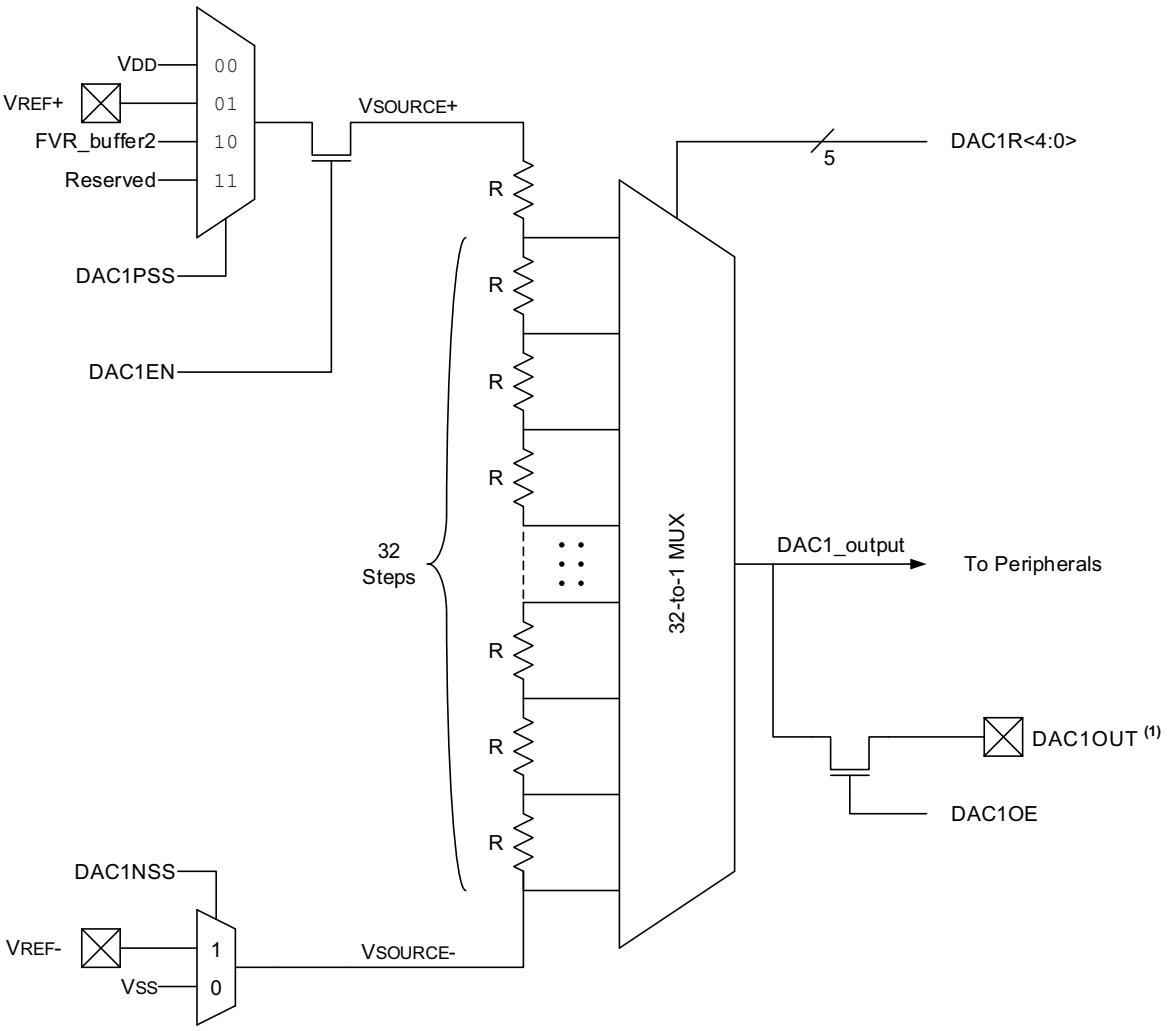
The value of the individual resistors within the ladder can be found in Table 34-15.

23.3 DAC Voltage Reference Output

The DAC voltage can be output to the DAC1OUT pin by setting the DAC1OE bit of the DAC1CON0 register. Selecting the DAC reference voltage for output on the DAC1OUT pin automatically overrides the digital output buffer and digital input threshold detector functions, it disables the weak pull-up and the constant-current drive function of that pin. Reading the DAC1OUT pin when it has been configured for DAC reference voltage output will always return a '0'.

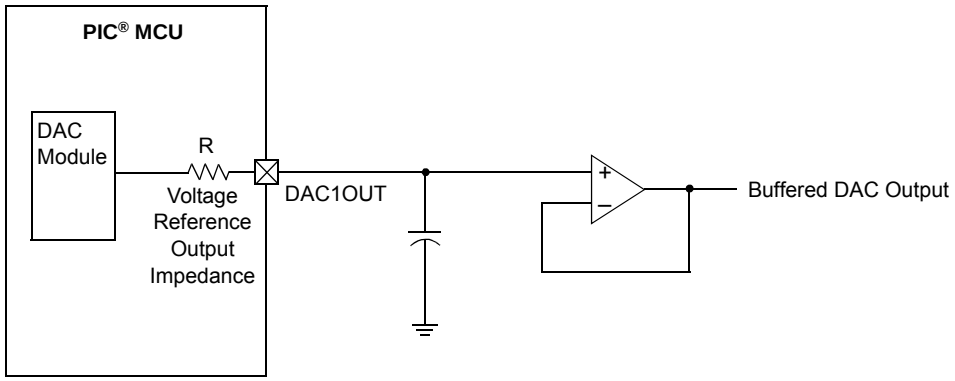
Due to the limited current drive capability, a buffer must be used on the DAC voltage reference output for external connections to the DAC1OUT pin. Figure 23-2 shows an example buffering technique.

FIGURE 23-1: DIGITAL-TO-ANALOG CONVERTER BLOCK DIAGRAM



Note 1: The unbuffered DAC1_output is provided on the DAC1OUT pin(s).

FIGURE 23-2: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE



23.4 Operation During Sleep

The DAC continues to function during Sleep. When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the DAC1CON0 register are not affected.

23.5 Effects of a Reset

A device Reset affects the following:

- DAC is disabled.
- DAC output voltage is removed from the DAC1OUT pin.
- The DAC1R<4:0> range select bits are cleared.

23.6 Register Definitions: DAC Control

REGISTER 23-1: DACCON0: VOLTAGE REFERENCE CONTROL REGISTER 0

R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0
DAC1EN	—	DAC1OE	—	DAC1PSS<1:0>		—	DAC1NSS
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 **DAC1EN:** DAC1 Enable bit
1 = DAC is enabled
0 = DAC is disabled

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **DAC1OE:** DAC1 Voltage Output 1 Enable bit
1 = DAC voltage level is also an output on the DAC1OUT pin
0 = DAC voltage level is disconnected from the DAC1OUT pin

bit 4 **Unimplemented:** Read as ‘0’

bit 3-2 **DAC1PSS<1:0>:** DAC1 Positive Source Select bits
11 = Reserved, do not use
10 = FVR output
01 = V_{REF+} pin
00 = V_{DD}

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **DAC1NSS:** DAC1 Negative Source Select bits
1 = V_{REF-} pin
0 = V_{SS}

REGISTER 23-2: DACCON1: VOLTAGE REFERENCE CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	DAC1R<4:0>				
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 **Unimplemented:** Read as '0'

bit 4-0 **DAC1R<4:0>:** DAC1 Voltage Output Select bits

$V_{OUT} = (V_{SRC+} - V_{SRC-})*(DAC1R<4:0>/32) + V_{SRC-}$

TABLE 23-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE DAC1 MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
DACCON0	DAC1EN	—	DAC1OE	—	DAC1PSS<1:0>		—	DAC1NSS	261
DACCON1	—	—	—	DAC1R<4:0>					262
CMxCON1	CxINTP	CxINTN	CxPCH<2:0>			CxNCH<2:0>			189
ADCON0	CHS<5:0>						GO/$\overline{DONE}$	ADON	242

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used with the DAC module.

24.0 DATA SIGNAL MODULATOR (DSM) MODULE

The Data Signal Modulator (DSM) is a peripheral which allows the user to mix a data stream, also known as a modulator signal, with a carrier signal to produce a modulated output.

Both the carrier and the modulator signals are supplied to the DSM module either internally, from the output of a peripheral, or externally through an input pin.

The modulated output signal is generated by performing a logical “AND” operation of both the carrier and modulator signals and then provided to the MDOUT pin.

The carrier signal is comprised of two distinct and separate signals. A carrier high (CARH) signal and a carrier low (CARL) signal. During the time in which the modulator (MOD) signal is in a logic high state, the DSM mixes the carrier high signal with the modulator signal. When the modulator signal is in a logic low state, the DSM mixes the carrier low signal with the modulator signal.

Using this method, the DSM can generate the following types of key modulation schemes:

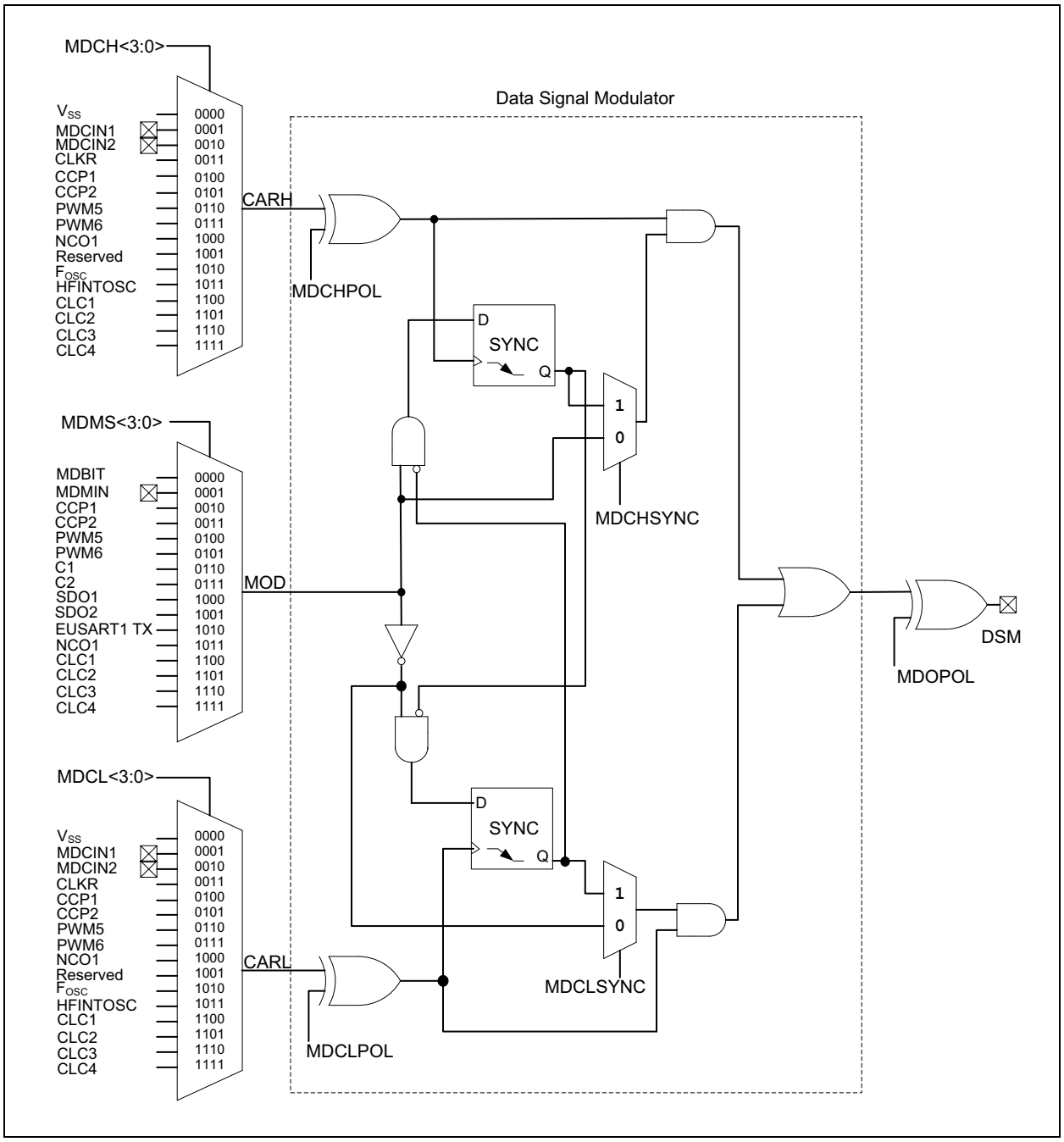
- Frequency-Shift Keying (FSK)
- Phase-Shift Keying (PSK)
- On-Off Keying (OOK)

Additionally, the following features are provided within the DSM module:

- Carrier Synchronization
- Carrier Source Polarity Select
- Carrier Source Pin Disable
- Programmable Modulator Data
- Modulator Source Pin Disable
- Modulated Output Polarity Select
- Slew Rate Control

Figure 24-1 shows a simplified block diagram of the Data Signal Modulator peripheral.

FIGURE 24-1: SIMPLIFIED BLOCK DIAGRAM OF THE DATA SIGNAL MODULATOR



24.1 DSM Operation

The DSM module can be enabled by setting the MDEN bit in the MDCON register. Clearing the MDEN bit in the MDCON register, disables the DSM module by automatically switching the carrier high and carrier low signals to the VSS signal source. The modulator signal source is also switched to the MDBIT in the MDCON register. This not only assures that the DSM module is inactive, but that it is also consuming the least amount of current.

The values used to select the carrier high, carrier low, and modulator sources held by the Modulation Source, Modulation High Carrier, and Modulation Low Carrier control registers are not affected when the MDEN bit is cleared and the DSM module is disabled. The values inside these registers remain unchanged while the DSM is inactive. The sources for the carrier high, carrier low and modulator signals will once again be selected when the MDEN bit is set and the DSM module is again enabled and active.

The modulated output signal can be disabled without shutting down the DSM module. The DSM module will remain active and continue to mix signals, but the output value will not be sent to the DSM pin. During the time that the output is disabled, the DSM pin will remain low. The modulated output can be disabled by clearing the MDEN bit in the MDCON register.

24.2 Modulator Signal Sources

The modulator signal can be supplied from the following sources:

- CCP1 Signal
- CCP2 Signal
- PWM5 Output
- PWM6 Output
- MSSP1 SDO1 Signal (SPI mode only)
- MSSP2 SDO2 Signal (SPI mode only)
- Comparator C1 Signal
- Comparator C2 Signal
- EUSART1 TX Signal
- External Signal on MDMIN pin
- NCO1 output
- CLC1 output
- CLC2 output
- CLC3 output
- CLC4 output
- MDBIT bit in the MDCON register

The modulator signal is selected by configuring the MDMS <3:0> bits in the MDSRC register.

24.3 Carrier Signal Sources

The carrier high signal and carrier low signal can be supplied from the following sources:

- CCP1 Signal
- CCP2 Signal
- PWM5 Output
- PWM6 Output
- NCO1 Output
- FOSC (System Clock)
- HFINTOSC
- CLC1 Output
- CLC2 Output
- CLC3 Output
- CLC4 Output
- Reference Clock Module Signal
- External Signal on MDCIN1 pin
- External Signal on MDCIN2 pin
- VSS

The carrier high signal is selected by configuring the MDCH <3:0> bits in the MDCARH register. The carrier low signal is selected by configuring the MDCL <3:0> bits in the MDCARL register.

24.4 Carrier Synchronization

During the time when the DSM switches between carrier high and carrier low signal sources, the carrier data in the modulated output signal can become truncated. To prevent this, the carrier signal can be synchronized to the modulator signal. When the modulator signal transitions away from the synchronized carrier, the unsynchronized carrier source is immediately active, while the synchronized carrier remains active until its next falling edge. When the modulator signal transitions back to the synchronized carrier, the unsynchronized carrier is immediately disabled, and the modulator waits until the next falling edge of the synchronized carrier before the synchronized carrier becomes active.

Synchronization is enabled separately for the carrier high and carrier low signal sources. Synchronization for the carrier high signal is enabled by setting the MDCHSYNC bit in the MDCARH register. Synchronization for the carrier low signal is enabled by setting the MDCLSYNC bit in the MDCARL register.

Figure 24-1 through Figure 24-6 show timing diagrams of using various synchronization methods.

FIGURE 24-2: ON OFF KEYING (OOK) SYNCHRONIZATION

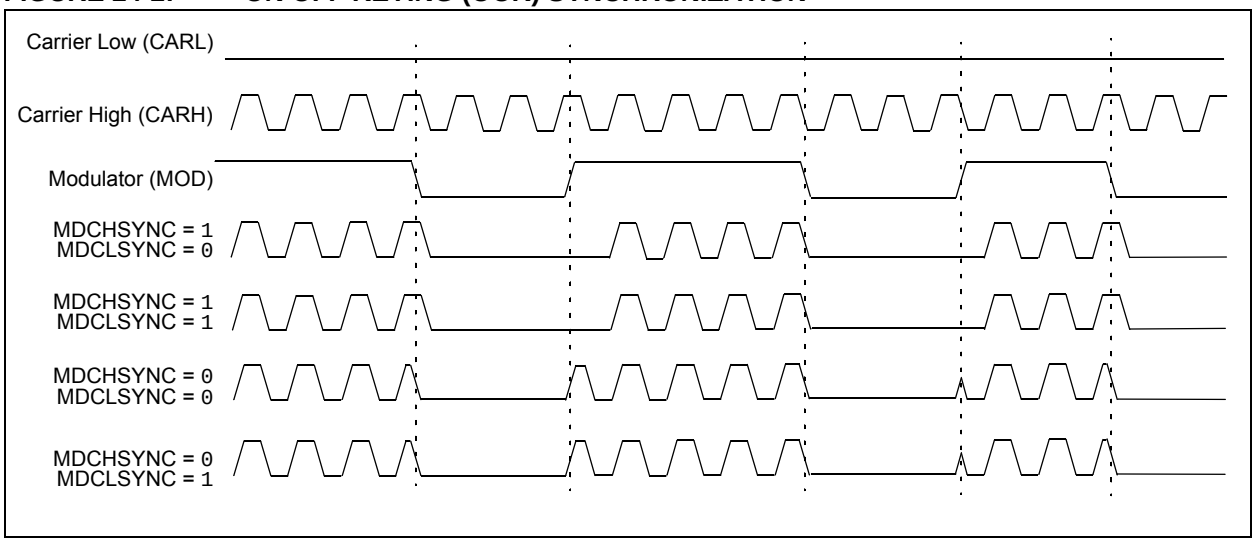


FIGURE 24-3: NO SYNCHRONIZATION (MDSHSYNC = 0, MDCLSYNC = 0)

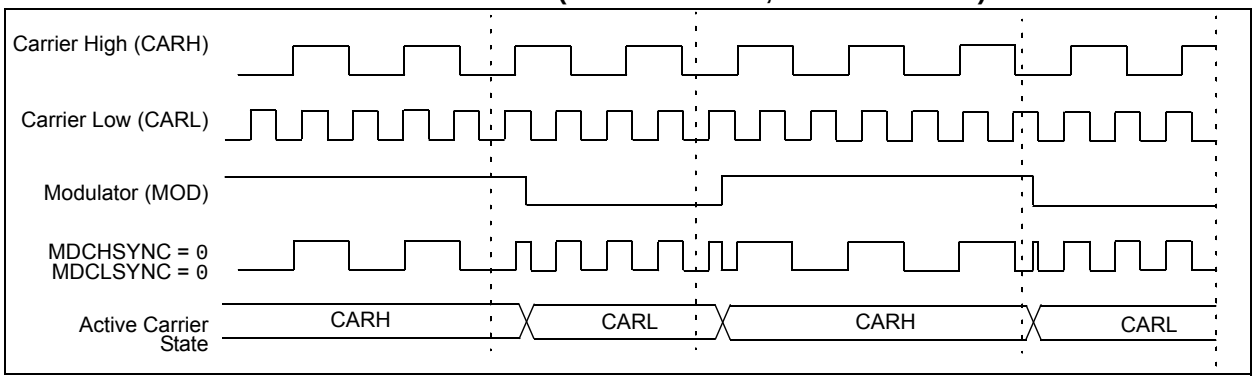


FIGURE 24-4: CARRIER HIGH SYNCHRONIZATION (MDSHSYNC = 1, MDCLSYNC = 0)

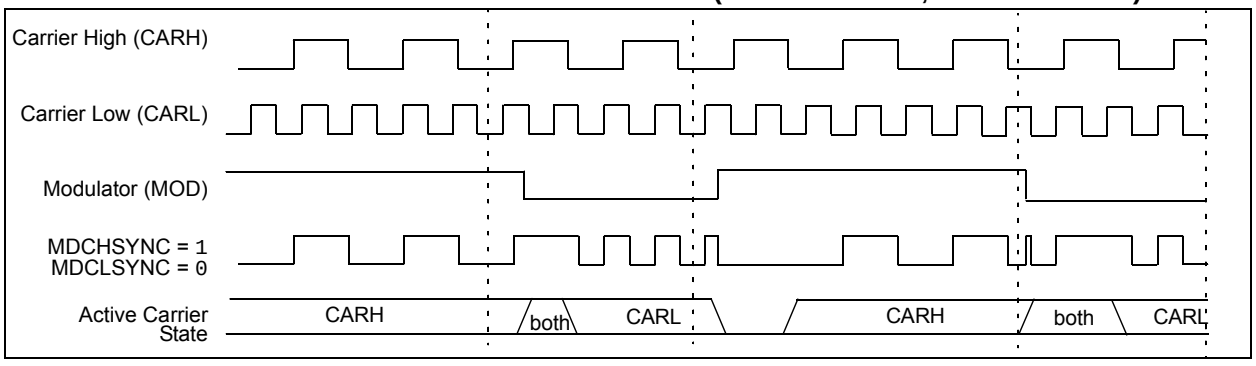


FIGURE 24-5: CARRIER LOW SYNCHRONIZATION (MDSHSYNC = 0, MDCLSYNC = 1)

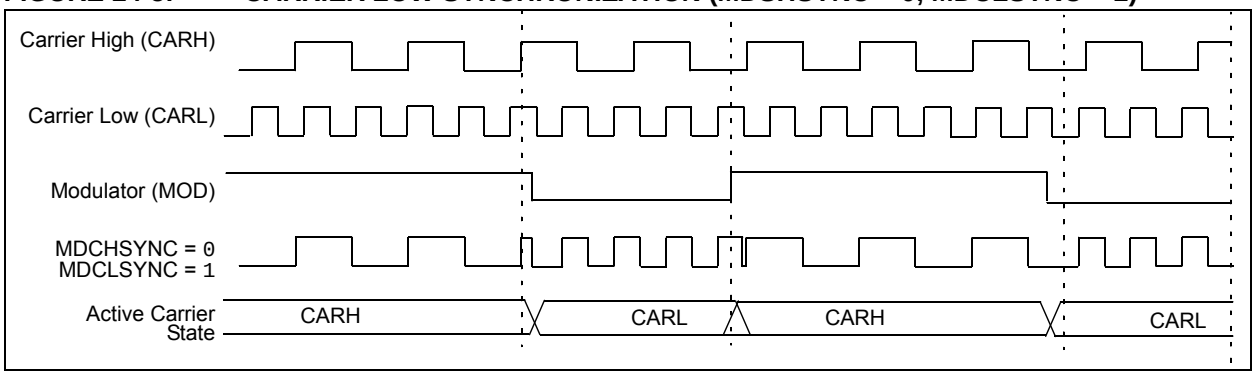
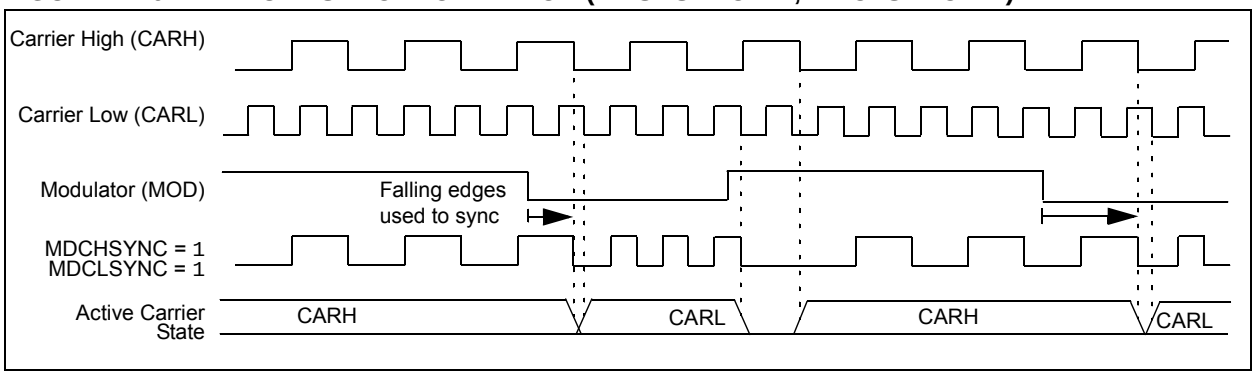


FIGURE 24-6: FULL SYNCHRONIZATION (MDSHSYNC = 1, MDCLSYNC = 1)



24.5 Carrier Source Polarity Select

The signal provided from any selected input source for the carrier high and carrier low signals can be inverted. Inverting the signal for the carrier high source is enabled by setting the MDCHPOL bit of the MDCARH register. Inverting the signal for the carrier low source is enabled by setting the MDCLPOL bit of the MDCARL register.

24.6 Programmable Modulator Data

The MDBIT of the MDCON register can be selected as the source for the modulator signal. This gives the user the ability to program the value used for modulation.

24.7 Modulated Output Polarity

The modulated output signal provided on the DSM pin can also be inverted. Inverting the modulated output signal is enabled by setting the MDOPOL bit of the MDCON register.

24.8 Slew Rate Control

The slew rate limitation on the output port pin can be disabled. The slew rate limitation can be removed by clearing the SLR bit of the SLRCON register associated with that pin. For example, clearing the slew rate limitation for pin RA5 would require clearing the SLRA5 bit of the SLRCONA register.

24.9 Operation in Sleep Mode

The DSM module is not affected by Sleep mode. The DSM can still operate during Sleep, if the Carrier and Modulator input sources are also still operable during Sleep.

24.10 Effects of a Reset

Upon any device Reset, the DSM module is disabled. The user's firmware is responsible for initializing the module before enabling the output. The registers are reset to their default values.

24.11 Register Definitions: Modulation Control

REGISTER 24-1: MDCON: MODULATION CONTROL REGISTER

R/W-0/0	U-0	U-0	R/W-0/0	R-0/0	U-0	U-0	R/W-0/0
MDEN	—	—	MDOPOL	MDOUT	—	—	MDBIT(2)
bit 7							bit 0

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

bit 7 **MDEN:** Modulator Module Enable bit
- 1 = Modulator module is enabled and mixing input signals
- 0 = Modulator module is disabled and has no output

bit 6-5 **Unimplemented:** Read as '0'

bit 4 **MDOPOL:** Modulator Output Polarity Select bit
- 1 = Modulator output signal is inverted; idle high output
- 0 = Modulator output signal is not inverted; idle low output

bit 3 **MDOUT:** Modulator Output bit
Displays the current output value of the modulator module.(1)

bit 2-1 **Unimplemented:** Read as '0'

bit 0 **MDBIT:** Allows software to manually set modulation source input to module(2)

Note 1: The modulated output frequency can be greater and asynchronous from the clock that updates this register bit, the bit value may not be valid for higher speed modulator or carrier signals.

2: MDBIT must be selected as the modulation source in the MDSRC register for this operation.

REGISTER 24-2: MDSRC: MODULATION SOURCE CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	MDMS<3:0>			
bit 7							bit 0

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

bit 7-4 **Unimplemented:** Read as '0'

bit 3-0 **MDMS<3:0>** Modulation Source Selection bits

1111 = CLC4 output
1110 = CLC3 output
1101 = CLC2 output
1100 = CLC1 output
1011 = NCO1 output
1010 = EUSART1 TX output
1001 = MSSP2 SDO2 output
1000 = MSSP1 SDO1 output
0111 = C2 (Comparator 2) output
0110 = C1 (Comparator 1) output
0101 = PWM6 output
0100 = PWM5 output
0011 = CCP2 output (PWM Output mode only)
0010 = CCP1 output (PWM Output mode only)
0001 = MDMINPPS
0000 = MDBIT bit of MDCON register is modulation source

REGISTER 24-3: MDCARH: MODULATION HIGH CARRIER CONTROL REGISTER

U-0	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>(1)			
bit 7							bit 0

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

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

bit 6 **MDCHPOL:** Modulator High Carrier Polarity Select bit
- 1 = Selected high carrier signal is inverted
- 0 = Selected high carrier signal is not inverted

bit 5 **MDCHSYNC:** Modulator High Carrier Synchronization Enable bit
- 1 = Modulator waits for a falling edge on the high time carrier signal before allowing a switch to the low time carrier
- 0 = Modulator output is not synchronized to the high time carrier signal(1)

bit 4 **Unimplemented:** Read as '0'

bit 3-0 **MDCH<3:0>** Modulator Data High Carrier Selection bits (1)
- 1111 = CLC4 output
- 1110 = CLC3 output
- 1101 = CLC2 output
- 1100 = CLC1 output
- 1011 = HFINTOSC
- 1010 = FOSC
- 1001 = Reserved. No channel connected.
- 1000 = NCO1 output
- 0111 = PWM6 output
- 0110 = PWM5 output
- 0101 = CCP2 output (PWM Output mode only)
- 0100 = CCP1 output (PWM Output mode only)
- 0011 = Reference clock module signal (CLKR)
- 0010 = MDCIN2PPS
- 0001 = MDCIN1PPS
- 0000 = VSS

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

REGISTER 24-4: MDCARL: MODULATION LOW CARRIER CONTROL REGISTER

U-0	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>(1)			
bit 7							bit 0

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

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

bit 6 **MDCLPOL:** Modulator Low Carrier Polarity Select bit
- 1 = Selected low carrier signal is inverted
- 0 = Selected low carrier signal is not inverted

bit 5 **MDCLSYNC:** Modulator Low Carrier Synchronization Enable bit
- 1 = Modulator waits for a falling edge on the low time carrier signal before allowing a switch to the high time carrier
- 0 = Modulator output is not synchronized to the low time carrier signal(1)

bit 4 **Unimplemented:** Read as '0'

bit 3-0 **MDCL<3:0>** Modulator Data High Carrier Selection bits (1)
- 1111 = CLC4 output
- 1110 = CLC3 output
- 1101 = CLC2 output
- 1100 = CLC1 output
- 1011 = HFINTOSC
- 1010 = FOSC
- 1001 = Reserved. No channel connected.
- 1000 = NCO1 output
- 0111 = PWM6 output
- 0110 = PWM5 output
- 0101 = CCP2 output (PWM Output mode only)
- 0100 = CCP1 output (PWM Output mode only)
- 0011 = Reference clock module signal (CLKR)
- 0010 = MDCIN2PPS
- 0001 = MDCIN1PPS
- 0000 = VSS

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

TABLE 24-1: SUMMARY OF REGISTERS ASSOCIATED WITH DATA SIGNAL MODULATOR MODE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	144
INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	144
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
SLRCONB(1)	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	150
INLVLB(1)	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	150
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
SLRCONC	SLRC7(1)	SLRC6(1)	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	156
INLVLC	INLVLC7(1)	INLVLC6(1)	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	157
MDCON	MDEN	—	—	MDOPOL	MDOUT	—	—	MDBIT	269
MDSRC	—	—	—	—	MDMS<3:0>				270
MDCARH	—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>				271
MDCARL	—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				272
MDCIN1PPS	—	—	—	MDCIN1PPS<4:0>					160
MDCIN2PPS	—	—	—	MDCIN2PPS<4:0>					160
MDMINPPS	—	—	—	MDMINPPS<4:0>					160

Legend: — = unimplemented, read as '0'. Shaded cells are not used in the Data Signal Modulator mode.

Note 1: PIC16(L)F18345 only.

2: Unimplemented. Read as '1'.

25.0 TIMER0 MODULE

The Timer0 module is an 8/16-bit timer/counter with the following features:

- 16-bit timer/counter
- 8-bit timer/counter with programmable period
- Synchronous or asynchronous operation
- Selectable clock sources
- Programmable prescaler (independent of Watchdog Timer)
- Programmable postscaler
- Operation during Sleep mode
- Interrupt on match or overflow
- Output on I/O pin (via PPS) or to other peripherals

25.1 Timer0 Operation

Timer0 can operate as either an 8-bit timer/counter or a 16-bit timer/counter. The mode is selected with the T016BIT bit of the T0CON register.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or a counter and increments on every rising edge of the external source.

25.1.1 16-BIT MODE

In normal operation, TMR0 increments on the rising edge of the clock source. A 15-bit prescaler on the clock input gives several prescale options (see prescaler control bits, T0CKPS<3:0> in the T0CON1 register).

25.1.1.1 Timer0 Reads and Writes in 16-bit Mode

TMR0H is not the actual high byte of Timer0 in 16-bit mode. It is actually a buffered version of the real high byte of Timer0, which is neither directly readable nor writable (see Figure 25-1). TMR0H is updated with the contents of the high byte of Timer0 during a read of TMR0L. This provides the ability to read all 16 bits of Timer0 without having to verify that the read of the high and low byte was valid, due to a rollover between successive reads of the high and low byte.

Similarly, a write to the high byte of Timer0 must also take place through the TMR0H Buffer register. The high byte is updated with the contents of TMR0H when a write occurs to TMR0L. This allows all 16 bits of Timer0 to be updated at once.

25.1.2 8-BIT MODE

In normal operation, TMR0 increments on the rising edge of the clock source. A 15-bit prescaler on the clock input gives several prescale options (see prescaler control bits, T0CKPS<3:0> in the T0CON1 register).

The value of TMR0L is compared to that of the Period buffer, a copy of TMR0H, on each clock cycle. When the two values match, the following events happen:

- TMR0_out goes high for one prescaled clock period
- TMR0L is reset
- The contents of TMR0H are copied to the period buffer

In 8-bit mode, the TMR0L and TMR0H registers are both directly readable and writable. The TMR0L register is cleared on any device Reset, while the TMR0H register initializes at FFh.

Both the prescaler and postscaler counters are cleared on the following events:

- A write to the TMR0L register
- A write to either the T0CON0 or T0CON1 registers.
- Any device Reset – Power-on Reset (POR),$\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset (WDTR) or Brown-out Reset (BOR)

25.1.3 COUNTER MODE

In Counter mode, the prescaler is normally disabled by setting the T0CKPS bits of the T0CON1 register to '0000'. Each rising edge of the clock input (or the output of the prescaler if the prescaler is used) increments the counter by '1'.

25.1.4 TIMER MODE

In Timer mode, the Timer0 module will increment every instruction cycle as long as there is a valid clock signal and the T0CKPS bits of the T0CON1 register (Register 25-4) are set to '0000'. When a prescaler is added, the timer will increment at the rate based on the prescaler value.

25.1.5 ASYNCHRONOUS MODE

When the T0ASYNC bit of the T0CON1 register is set (T0ASYNC = 1), the counter increments with each rising edge of the input source (or output of the prescaler, if used). Asynchronous mode allows the counter to continue operation during Sleep mode provided that the clock also continues to operate during Sleep.

25.1.6 SYNCHRONOUS MODE

When the T0ASYNC bit of the T0CON1 register is clear (T0ASYNC = 0), the counter clock is synchronized to the system oscillator (FOSC/4). When operating in Synchronous mode, the counter clock frequency cannot exceed FOSC/4.

25.2 Clock Source Selection

The T0CS<2:0> bits of the T0CON1 register are used to select the clock source for Timer0. Register 25-4 displays the clock source selections.

25.2.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected, Timer0 operates as a timer and will increment on multiples of the clock source, as determined by the Timer0 prescaler.

25.2.2 EXTERNAL CLOCK SOURCE

When an external clock source is selected, Timer0 can operate as either a timer or a counter. Timer0 will increment on multiples of the rising edge of the external clock source, as determined by the Timer0 prescaler.

25.3 Programmable Prescaler

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

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

25.4 Programmable Postscaler

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

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

25.5 Operation During Sleep

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

25.6 Timer0 Interrupts

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

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

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

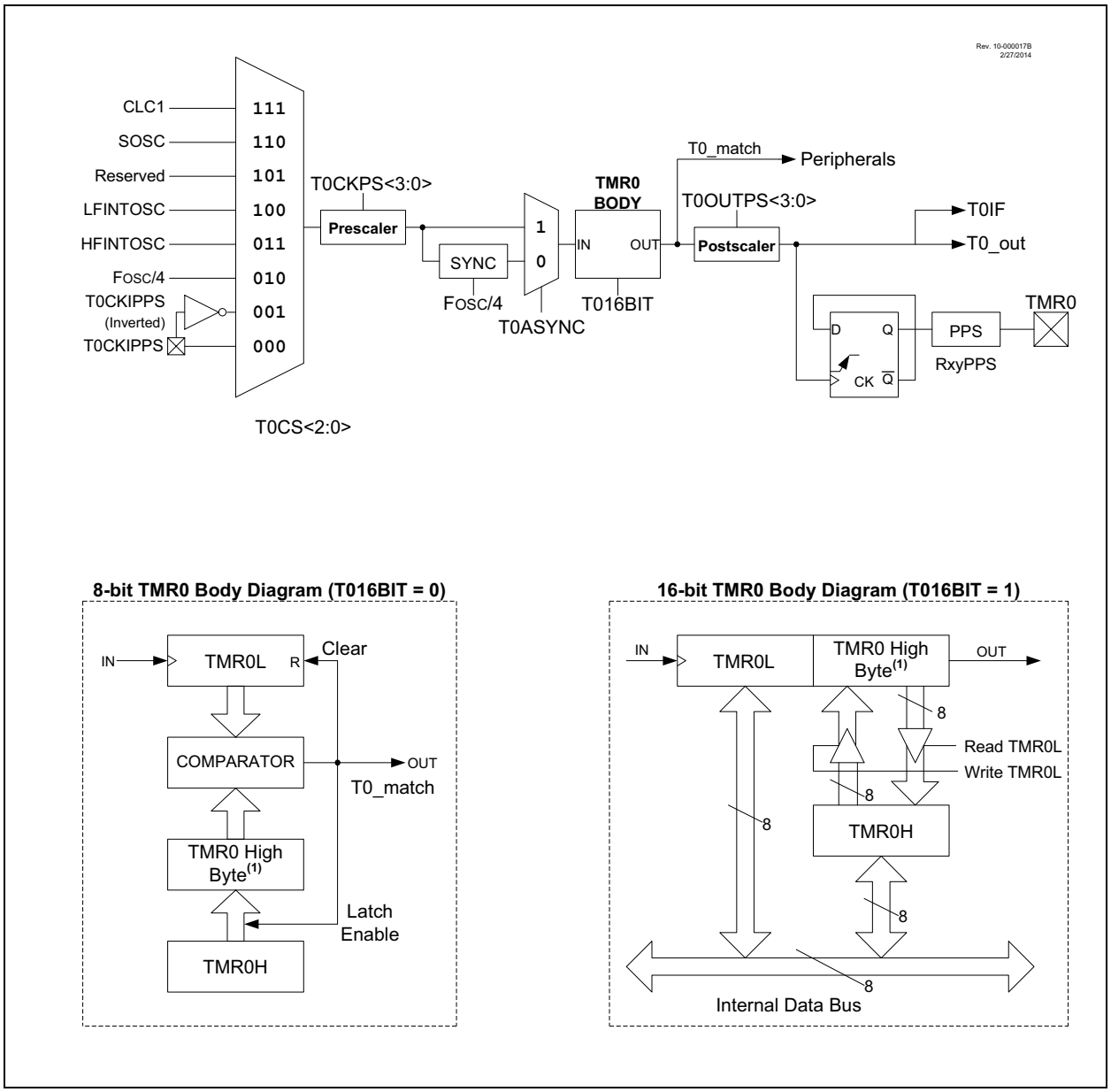
If Timer0 interrupts are enabled (TMR0IE bit of the PIE0 register = 1), the CPU will be interrupted and the device may wake from Sleep (see **Section 25.5 "Operation During Sleep"** for more details).

25.7 Timer0 Output

The Timer0 output can be routed to any I/O pin via the RxyPPS output selection register (see **Section 12.0 "Peripheral Pin Select (PPS) Module"** for additional information). The Timer0 output can also be used by other peripherals, such as the auto-conversion trigger of the Analog-to-Digital Converter. Finally, the Timer0 output can be monitored through software via the Timer0 Output bit (T0OUT) of the T0CON0 register (Register 25-3).

TMR0_out will be one postscaled clock period when a match occurs between TMR0L and TMR0H in 8-bit mode, or when TMR0 rolls over in 16-bit mode. The Timer0 output is a 50% duty cycle that toggles on each TMR0_out rising clock edge.

FIGURE 25-1: BLOCK DIAGRAM OF TIMER0



25.8 Register Definitions: Timer0 Register

REGISTER 25-1: TMR0L: TIMER0 COUNT REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR0L<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 **TMR0L<7:0>:** TMR0 Counter bits 7..0

REGISTER 25-2: TMR0H: TIMER0 PERIOD REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TMR0H<7:0> or TMR0<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 When T016BIT = 0
TMR0H<7:0>: TMR0 Period Register Bits 7..0
When T016BIT = 1
TMR0<15:8>: TMR0 Counter bits 15..8

REGISTER 25-3: T0CON0: TIMER0 CONTROL REGISTER 0

R/W-0/0	U-0	R-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>			
bit 7							bit 0

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

bit 7 **T0EN:** TMR0 Enable bit
1 = The module is enabled and operating
0 = The module is disabled and in the lowest power mode

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **T0OUT:** TMR0 Output (read-only)
TMR0 output bit

bit 4 **T016BIT:** TMR0 Operating as 16-bit Timer Select bit
1 = TMR0 is a 16-bit timer
0 = TMR0 is an 8-bit timer

bit 3-0 **T0OUTPS<3:0>:** TMR0 Output Postscaler (divider) Select bits
0000 = 1:1 Postscaler
0001 = 1:2 Postscaler
0010 = 1:3 Postscaler
0011 = 1:4 Postscaler
0100 = 1:5 Postscaler
0101 = 1:6 Postscaler
0110 = 1:7 Postscaler
0111 = 1:8 Postscaler
1000 = 1:9 Postscaler
1001 = 1:10 Postscaler
1010 = 1:11 Postscaler
1011 = 1:12 Postscaler
1100 = 1:13 Postscaler
1101 = 1:14 Postscaler
1110 = 1:15 Postscaler
1111 = 1:16 Postscaler

REGISTER 25-4: T0CON1: TIMER0 CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
T0CS<2:0>			T0ASYNC	T0CKPS<3:0>			
bit 7							bit 0

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

bit 7-5 **T0CS<2:0>:** Timer0 Clock Source Select bits
000 = T0CKIPPS (True)
001 = T0CKIPPS (Inverted)
010 = FOSC/4
011 = HFINTOSC
100 = LFINTOSC
101 = Reserved
110 = SOSC
111 = CLC1

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

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

TABLE 25-1: 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
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
TMR0L	TMR0L<7:0>								277
TMR0H	TMR0H<7:0> or TMR0<15:8>								277
T0CON0	T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>				278
T0CON1	T0CS<2:0>			T0ASYNC	T0CKPS<3:0>				279
T0CKIPPS	—	—	—	T0CKIPPS<4:0>					160
TMR0PPS	—	—	—	TMR0PPS<4:0>					160
ADACT	—	—	—	ADACT<4:0>					244
CLCxSELy	—	—	LCxDyS<5:0>						227
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		291
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
PIR0	—	—	TMR0IF	IOCIF	—	—	—	INTF	104
PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	99

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by the Timer0 module.

Note 1: PIC16(L)F18345 only.

2: Unimplemented, read as '1'.

26.0 TIMER1/3/5 MODULE WITH GATE CONTROL

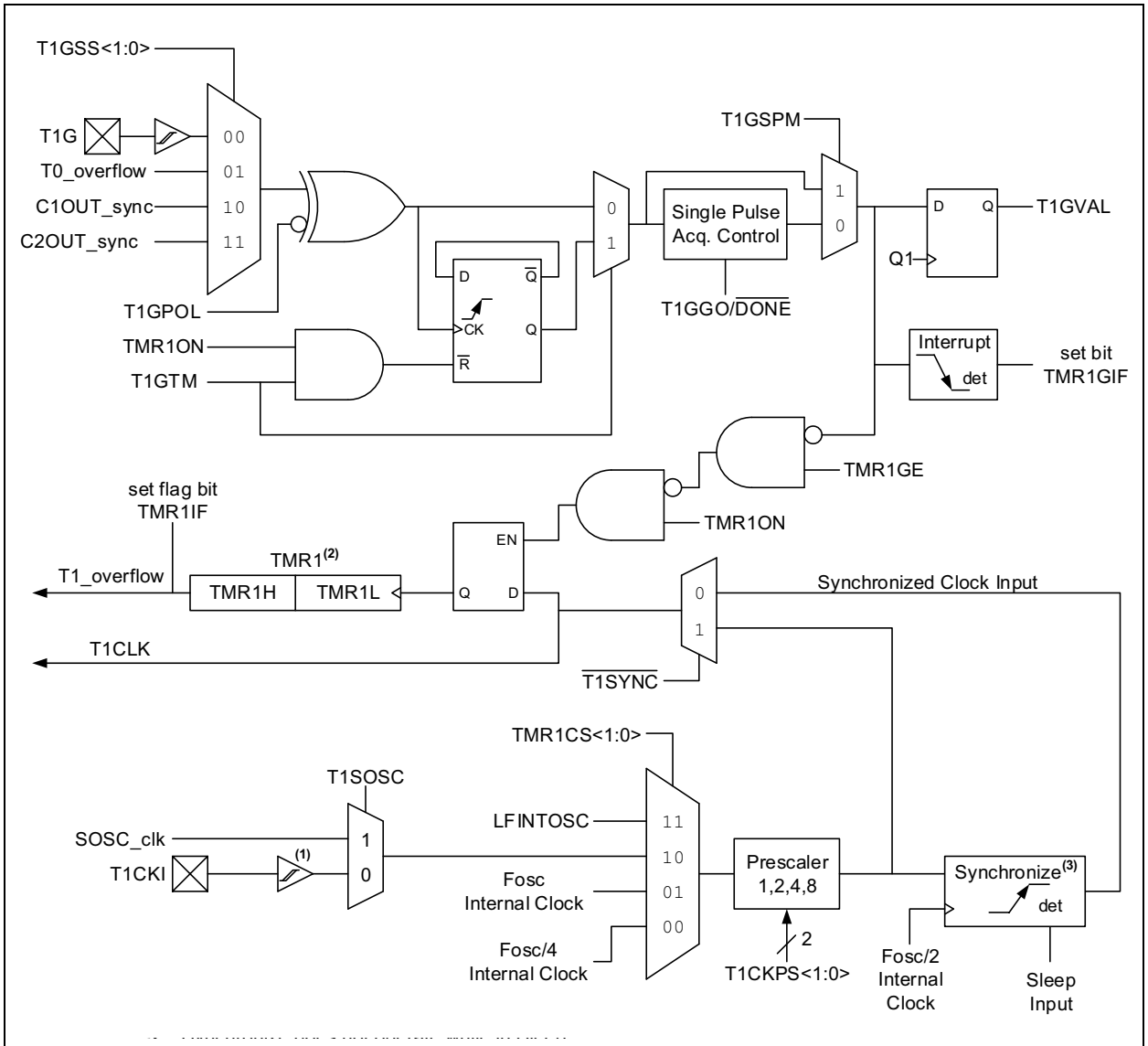
Timer1/3/5 modules are 16-bit timers/counters, each with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 2-bit prescaler
- Optionally synchronized comparator out
- Multiple Timer1 gate (count enable) sources
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- Time base for the Capture/Compare function
- Auto-Conversion Trigger (with CCP)
- Selectable Gate Source Polarity
- Gate Toggle mode
- Gate Single-Pulse mode
- Gate Value Status
- Gate Event Interrupt

Figure 26-1 is a block diagram of the Timer1 module.

Note 1:	In devices with more than one Timer module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the T1CON and T3CON control the same operational aspects of two completely different Timer modules.
2:	Throughout this section, generic references to Timer1 module in any of its operating modes may be interpreted as being equally applicable to Timerx module. Register names, module signals, I/O pins and bit names may use the generic designator ‘x’ to indicate the use of a numeral to distinguish a particular module, when required.

FIGURE 26-1: TIMER1 BLOCK DIAGRAM



Note 1: ST Buffer is high speed type when using T1CKI.
2: Timer1 register increments on rising edge.
3: Synchronize does not operate while in Sleep.

26.1 Timer1 Operation

The Timer1 module is a 16-bit incrementing counter which is accessed through the TMR1H:TMR1L register pair. Writes to TMR1H or TMR1L directly update the counter.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or counter and increments on every selected edge of the external source.

Timer1 is enabled by configuring the TMR1ON and TMR1GE bits in the T1CON and T1GCON registers, respectively. Table 26-1 displays the Timer1 enable selections.

TABLE 26-1: TIMER1 ENABLE SELECTIONS

TMR1ON	TMR1GE	Timer1 Operation
0	0	Off
0	1	Off
1	0	Always On
1	1	Count Enabled

26.2 Clock Source Selection

The TMR1CS<1:0> and T1SOSC bits of the T1CON register are used to select the clock source for Timer1. Table 26-2 displays the clock source selections.

26.2.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected, the TMR1H:TMR1L register pair will increment on multiples of FOSC as determined by the Timer1 prescaler.

When the FOSC internal clock source is selected, the Timer1 register value will increment by four counts every instruction clock cycle. Due to this condition, a 2 LSB error in resolution will occur when reading the Timer1 value. To utilize the full resolution of Timer1, an asynchronous input signal must be used to gate the Timer1 clock input.

The following asynchronous sources may be used:

- Asynchronous event on the T1G pin to Timer1 gate
- C1 or C2 comparator input to Timer1 gate

26.2.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When enabled to count, Timer1 is incremented on the rising edge of the external clock input, T1CKI, which can be either synchronized to the microcontroller system clock or run asynchronously.

When used as a timer with a clock oscillator, an external 32.768 kHz crystal can be connected to the SOSCI/SOSCO pins.

Note: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after any one or more of the following conditions:

- Timer1 enabled after POR
- Write to TMR1H or TMR1L
- Timer1 is disabled
- Timer1 is disabled (TMR1ON = 0) when T1CKI is high then Timer1 is enabled (TMR1ON=1) when T1CKI is low.

TABLE 26-2: CLOCK SOURCE SELECTIONS

TMR1CS<1:0>	Clock Source
11	LFINTOSC
10	External Clocking on T1CKI Pin
01	System Clock (FOSC)
00	Instruction Clock (FOSC/4)

26.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits of the T1CON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

26.4 Timer1 (Secondary) Oscillator

A dedicated low-power 32.768 kHz oscillator circuit is built-in between pins SOSCI (input) and SOSCO (amplifier output). This internal circuit is designed to be used in conjunction with an external 32.768 kHz crystal.

The oscillator circuit is enabled by setting the T1SOSC bit of the T1CON register. The oscillator will continue to run during Sleep.

Note:	The oscillator requires a start-up and stabilization time before use. Thus, T1SOSC should be set and a suitable delay observed prior to using Timer1. A suitable delay similar to the OST delay can be implemented in software by clearing the TMR1IF bit then presetting the TMR1H:TMR1L register pair to FC00h. The TMR1IF flag will be set when 1024 clock cycles have elapsed, thereby indicating that the oscillator is running and reasonably stable.

26.5 Timer1 Operation in Asynchronous Counter Mode

If the control bit $\overline{\text{T1SYNC}}$ of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If the external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 26.5.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

Note:	When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

26.5.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMR1H:TMR1L register pair.

26.6 Timer1 Gate

Timer1 can be configured to count freely or the count can be enabled and disabled using Timer1 gate circuitry. This is also referred to as Timer1 Gate Enable.

Timer1 gate can also be driven by multiple selectable sources.

26.6.1 TIMER1 GATE ENABLE

The Timer1 Gate Enable mode is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 Gate Enable mode is configured using the T1GPOL bit of the T1GCON register.

When Timer1 Gate Enable mode is enabled, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 Gate Enable mode is disabled, no incrementing will occur and Timer1 will hold the current count. See Figure 26-3 for timing details.

TABLE 26-3: TIMER1 GATE ENABLE SELECTIONS

T1CLK	T1GPOL	T1G	Timer1 Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
↑	1	1	Counts

26.6.2 TIMER1 GATE SOURCE SELECTION

Timer1 gate source selections are shown in Table 26-4. Source selection is controlled by the T1GSS bits of the T1GCON register. The polarity for each available source is also selectable. Polarity selection is controlled by the T1GPOL bit of the T1GCON register.

TABLE 26-4: TIMER1 GATE SOURCES

T1GSS	Timer1 Gate Source
00	Timer1 Gate Pin
01	Overflow of Timer0 (TMR0 increments from FFh to 00h)
10	Comparator 1 Output (optionally Timer1 synchronized output)
11	Comparator 2 Output (optionally Timer1 synchronized output)

26.6.2.1 T1G Pin Gate Operation

The T1G pin is one source for Timer1 gate control. It can be used to supply an external source to the Timer1 gate circuitry.

26.6.2.2 Timer0 Overflow Gate Operation

When Timer0 increments from FFh to 00h, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 gate circuitry.

26.6.2.3 Comparator C1 Gate Operation

The output resulting from a Comparator 1 operation can be selected as a source for Timer1 gate control. The Comparator 1 output can be synchronized to the Timer1 clock or left asynchronous. For more information see **Section 17.4.1 "Comparator Output Synchronization"**.

26.6.2.4 Comparator C2 Gate Operation

The output resulting from a Comparator 2 operation can be selected as a source for Timer1 gate control. The Comparator 2 output can be synchronized to the Timer1 clock or left asynchronous. For more information see **Section 17.4.1 "Comparator Output Synchronization"**.

26.6.3 TIMER1 GATE TOGGLE MODE

When Timer1 Gate Toggle mode is enabled, it is possible to measure the full-cycle length of a Timer1 gate signal, as opposed to the duration of a single level pulse.

The Timer1 gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. See Figure 26-4 for timing details.

Timer1 Gate Toggle mode is enabled by setting the T1GTM bit of the T1GCON register. When the T1GTM bit is cleared, the flip-flop is cleared and held clear. This is necessary in order to control which edge is measured.

Note:	Enabling Toggle mode at the same time as changing the gate polarity may result in indeterminate operation.

26.6.4 TIMER1 GATE SINGLE-PULSE MODE

When Timer1 Gate Single-Pulse mode is enabled, it is possible to capture a single-pulse gate event. Timer1 Gate Single-Pulse mode is first enabled by setting the T1GSPM bit in the T1GCON register. Next, the T1GGO/$\overline{\text{DONE}}$ bit in the T1GCON register must be set. The Timer1 will be fully enabled on the next incrementing edge. On the next trailing edge of the pulse, the T1GGO/$\overline{\text{DONE}}$ bit will automatically be cleared. No other gate events will be allowed to increment Timer1 until the T1GGO/$\overline{\text{DONE}}$ bit is once again set in software. See Figure 26-5 for timing details.

If the Single-Pulse Gate mode is disabled by clearing the T1GSPM bit in the T1GCON register, the T1GGO/$\overline{\text{DONE}}$ bit should also be cleared.

Enabling the Toggle mode and the Single-Pulse mode simultaneously will permit both sections to work together. This allows the cycle times on the Timer1 gate source to be measured. See Figure 26-6 for timing details.

26.6.5 TIMER1 GATE VALUE STATUS

When Timer1 Gate Value Status is utilized, it is possible to read the most current level of the gate control value. The value is stored in the T1GVAL bit in the T1GCON register. The T1GVAL bit is valid even when the Timer1 gate is not enabled (TMR1GE bit is cleared).

26.6.6 TIMER1 GATE EVENT INTERRUPT

When Timer1 Gate Event Interrupt is enabled, it is possible to generate an interrupt upon the completion of a gate event. When the falling edge of T1GVAL occurs, the TMR1GIF flag bit in the PIR1 register will be set. If the TMR1GIE bit in the PIE1 register is set, then an interrupt will be recognized.

The TMR1GIF flag bit operates even when the Timer1 gate is not enabled (TMR1GE bit is cleared).

26.7 Timer1 Interrupt

The Timer1 register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When Timer1 rolls over, the Timer1 interrupt flag bit of the PIR1 register is set. To enable the interrupt on rollover, you must set these bits:

- TMR1ON bit of the T1CON register
- TMR1IE bit of the PIE1 register
- PEIE bit of the INTCON register
- GIE bit of the INTCON register

The interrupt is cleared by clearing the TMR1IF bit in the Interrupt Service Routine.

Note:	The TMR1H:TMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts.

26.8 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when setup in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- TMR1ON bit of the T1CON register must be set
- TMR1IE bit of the PIE1 register must be set
- PEIE bit of the INTCON register must be set
- $\overline{\text{T1SYNC}}$ bit of the T1CON register must be set
- TMR1CS bits of the T1CON register must be configured
- T1SOSC bit of the T1CON register must be configured

The device will wake-up on an overflow and execute the next instructions. If the GIE bit of the INTCON register is set, the device will call the Interrupt Service Routine.

Secondary oscillator will continue to operate in Sleep regardless of the $\overline{\text{T1SYNC}}$ bit setting.

26.9 CCP Capture/Compare Time Base

The CCP modules use the TMR1H:TMR1L register pair as the time base when operating in Capture or Compare mode.

In Capture mode, the value in the TMR1H:TMR1L register pair is copied into the CCPR1H:CCPR1L register pair on a configured event.

In Compare mode, an event is triggered when the value CCPR1H:CCPR1L register pair matches the value in the TMR1H:TMR1L register pair. This event can be an Auto-conversion Trigger.

For more information, see **Section 28.0 "Capture/Compare/PWM Modules"**.

26.10 CCP Auto-Conversion Trigger

When any of the CCP's are configured to trigger an auto-conversion, the trigger will clear the TMR1H:TMR1L register pair. This auto-conversion does not cause a Timer1 interrupt. The CCP module may still be configured to generate a CCP interrupt.

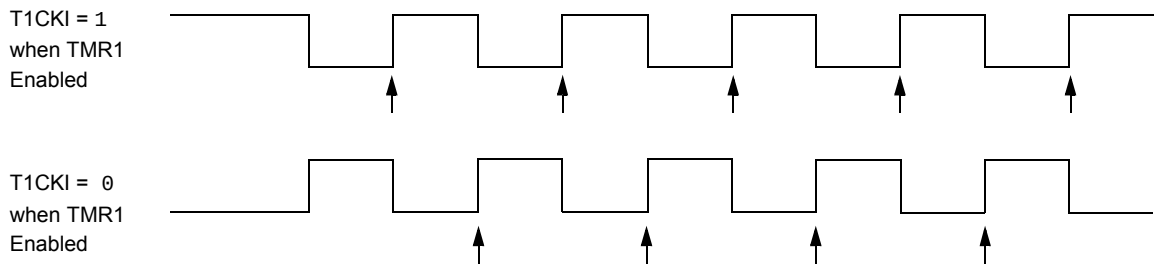
In this mode of operation, the CCPR1H:CCPR1L register pair becomes the period register for Timer1.

Timer1 should be synchronized and FOSC/4 should be selected as the clock source in order to utilize the Auto-conversion Trigger. Asynchronous operation of Timer1 can cause an Auto-conversion Trigger to be missed.

In the event that a write to TMR1H or TMR1L coincides with an Auto-conversion Trigger from the CCP, the write will take precedence.

For more information, see **Section 28.3.3 "Auto-Conversion Trigger"**.

FIGURE 26-2: TIMER1 INCREMENTING EDGE



Note 1: Arrows indicate counter increments.

2: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge of the clock.

FIGURE 26-3: TIMER1 GATE ENABLE MODE

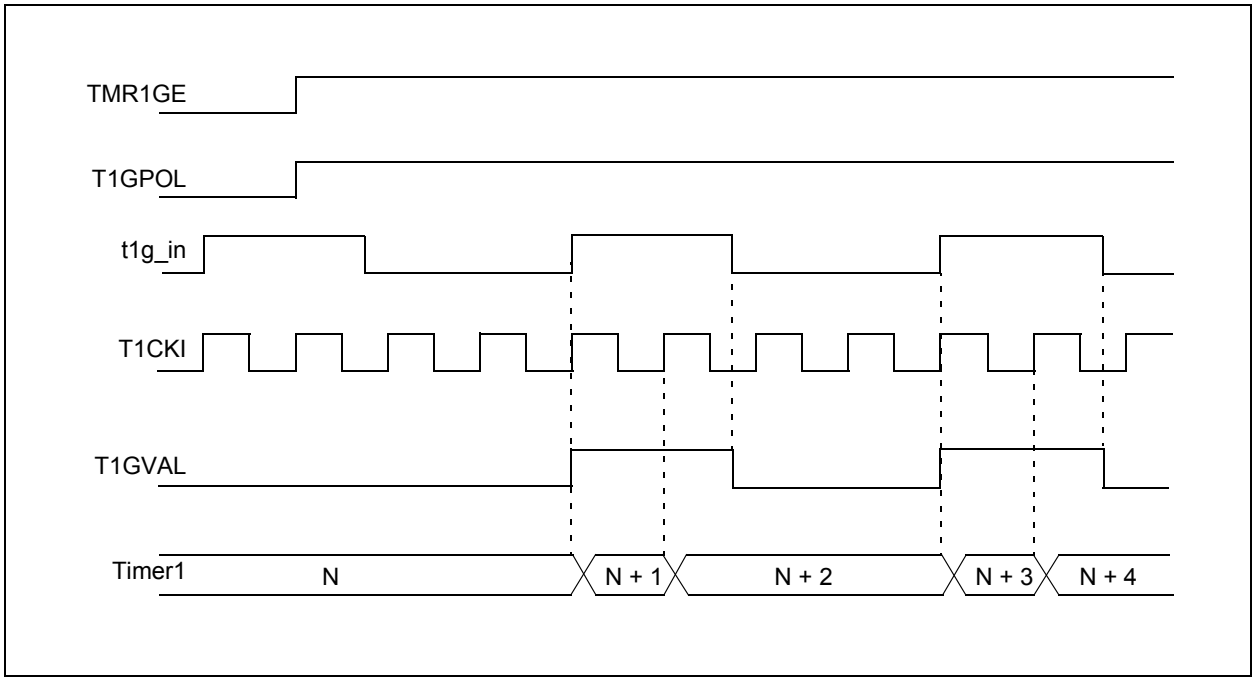


FIGURE 26-4: TIMER1 GATE TOGGLE MODE

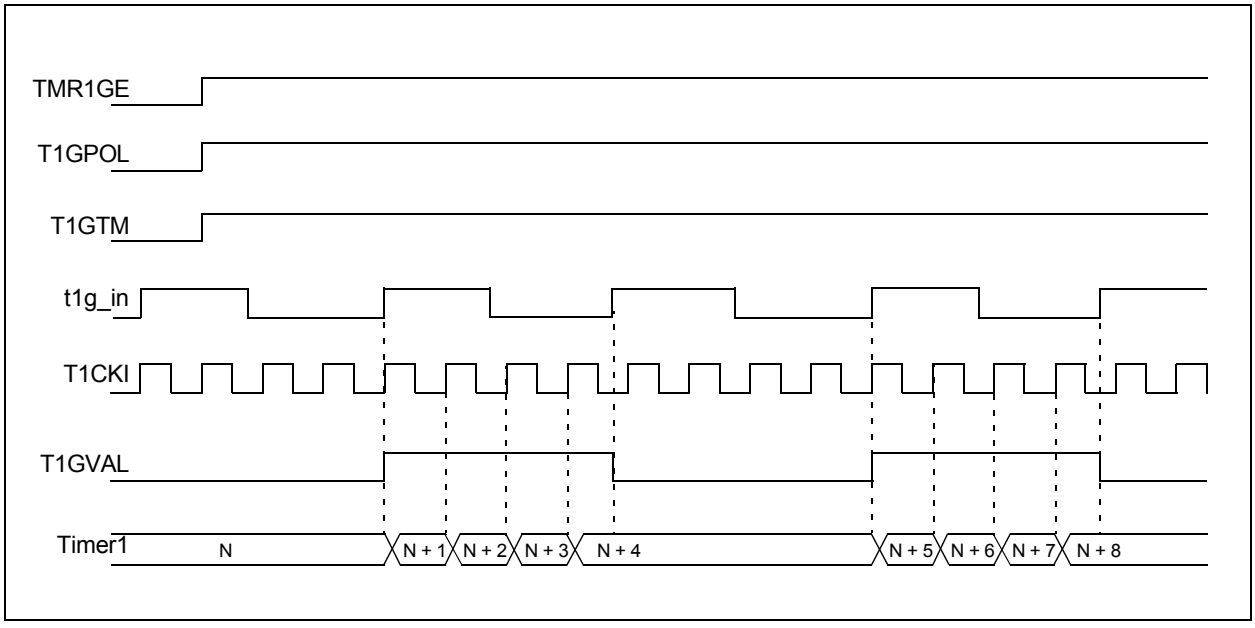


FIGURE 26-5: TIMER1 GATE SINGLE-PULSE MODE

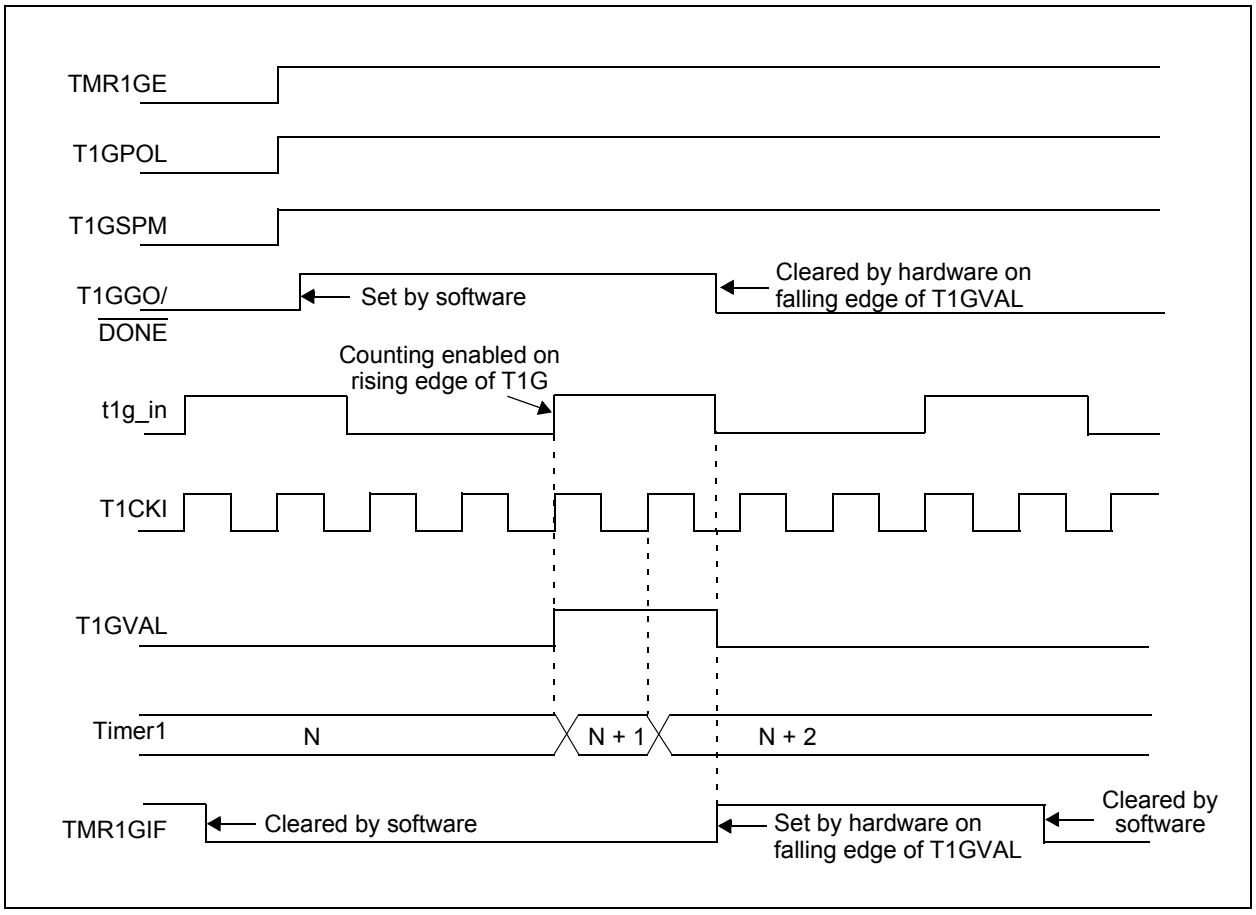
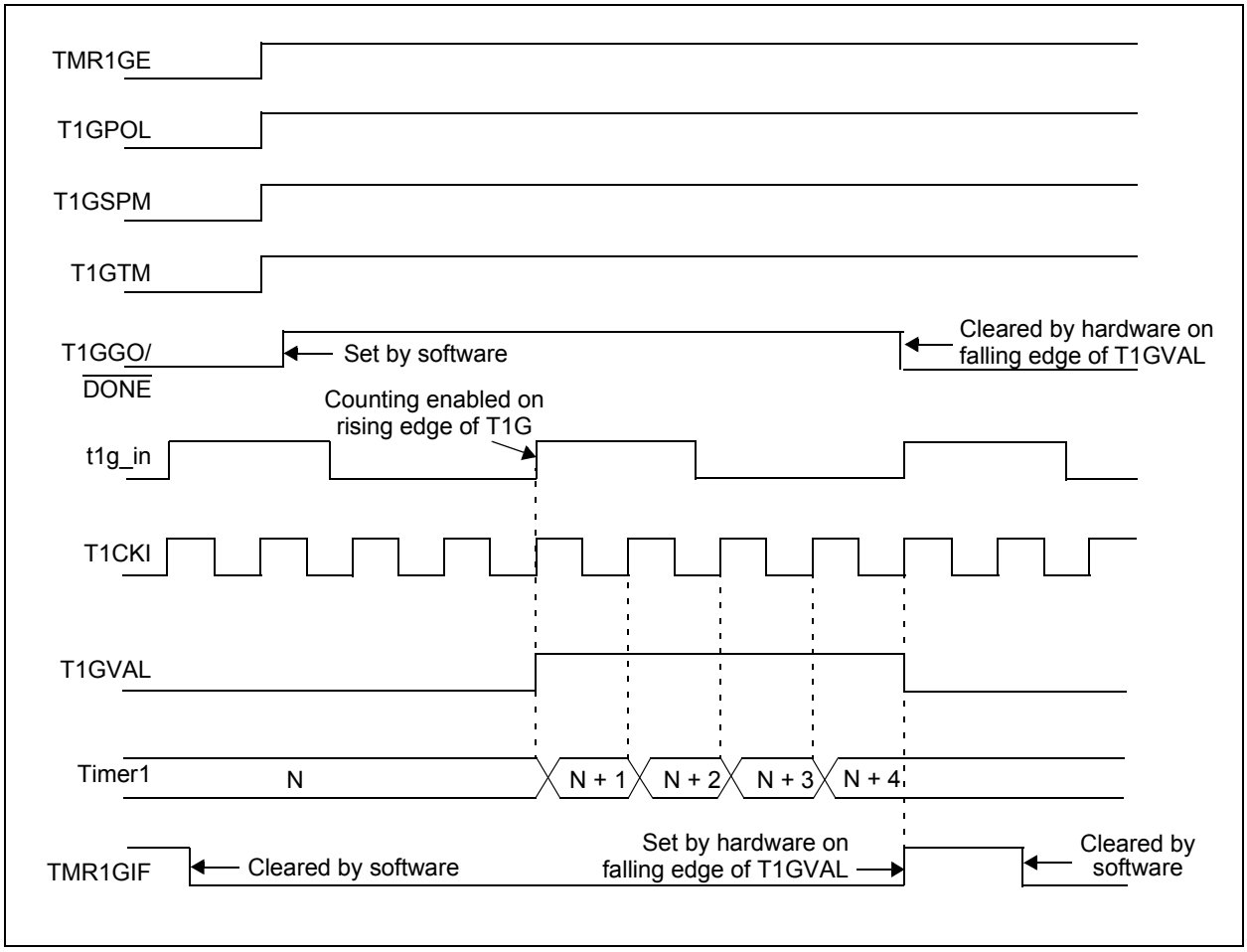


FIGURE 26-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE



26.11 Register Definitions: Timer1/3/5 Control

REGISTER 26-1: TxCON(1): TIMERx CONTROL REGISTER

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	U-0	R/W-0/u
TMRxCS<1:0>		TxCKPS<1:0>		TxSOSC	$\overline{\text{TxSYNC}}$	—	TMRxON
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 **TMRxCS<1:0>:** Timerx Clock Source Select bits
- 11 = Timerx clock Source is LFINTOSC
- 10 = Timerx clock source is pin or oscillator:
  - If TxSOSC = 0:
  - External clock from TxCKIPPS pin (on the rising edge)
  - If TxSOSC = 1:
  - Clock from SOSC, either crystal oscillator on TxSOSCI/TxSOSCO pins, or SOSCIN input
- 01 = Timerx clock source is system clock (FOSC)
- 00 = Timerx clock source is instruction clock (FOSC/4)

bit 5-4 **TxCKPS<1:0>:** Timerx Input Clock Prescale Select bits
- 11 = 1:8 Prescale value
- 10 = 1:4 Prescale value
- 01 = 1:2 Prescale value
- 00 = 1:1 Prescale value

bit 3 **TxSOSC:** LP Oscillator Enable Control bit
- 1 = SOSC requested as the clock source
- 0 = TxCKI enabled as the clock source

bit 2 **$\overline{\text{TxSYNC}}$:** Timer1 Synchronization Control bit

TMRxCS<1:0> = 1x
- 1 = Do not synchronize external clock input
- 0 = Synchronize external clock input with system clock

TMRxCS<1:0> = 0x

This bit is ignored. Timer1 uses the internal clock and no additional synchronization is performed.

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

bit 0 **TMRxON:** Timer1 On bit
- 1 = Enables Timerx
- 0 = Stops Timerx and clears Timerx gate flip-flop

Note 1: 'x' refers to either '1', '3' or '5' for the respective Timer1/3/5 registers.

REGISTER 26-2: TxGCON(1): TIMERx GATE CONTROL REGISTER

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W/HC-0/u	R-x/x	R/W-0/u	R/W-0/u
TMRxGE	TxGPOL	TxGTM	TxGSPM	TxGGO/$\overline{\text{DONE}}$	TxGVAL	TxGSS<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	HC = Bit is cleared by hardware

bit 7 **TMRxGE:** Timer1 Gate Enable bit

If TMRxON = 0:
This bit is ignored
If TMRxON = 1:
1 = Timerx counting is controlled by the Timer1 gate function
0 = Timerx is always counting

bit 6 **TxGPOL:** Timerx Gate Polarity bit

1 = Timerx gate is active-high (Timerx counts when gate is high)
0 = Timerx gate is active-low (Timerx counts when gate is low)

bit 5 **TxGTM:** Timerx Gate Toggle Mode bit

1 = Timerx Gate Toggle mode is enabled
0 = Timerx Gate Toggle mode is disabled and toggle flip-flop is cleared
Timerx gate flip-flop toggles on every rising edge.

bit 4 **TxGSPM:** Timerx Gate Single-Pulse Mode bit

1 = Timerx Gate Single-Pulse mode is enabled and is controlling Timerx gate
0 = Timerx Gate Single-Pulse mode is disabled

bit 3 **TxGGO/$\overline{\text{DONE}}$:** Timerx Gate Single-Pulse Acquisition Status bit

1 = Timerx gate single-pulse acquisition is ready, waiting for an edge
0 = Timerx gate single-pulse acquisition has completed or has not been started
This bit is automatically cleared when TxGSPM is cleared

bit 2 **TxGVAL:** Timerx Gate Value Status bit

Indicates the current state of the Timerx gate that could be provided to TMRxH:TMRxL
Unaffected by Timerx Gate Enable (TMRxGE)

bit 1-0 **TxGSS<1:0>:** Timerx Gate Source Select bits

11 = Comparator 2 optionally synchronized output
10 = Comparator 1 optionally synchronized output
01 = Timer0 overflow output
00 = Timerx gate pin

Note 1: 'x' refers to either '1', '3' or '5' for the respective Timer1/3/5 registers.

REGISTER 26-3: TMRxL(1): TIMERx LOW BYTE 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
TMRxL<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 **TMRxL<7:0>:** TMRx Low Byte bits

Note 1: 'x' refers to either '1', '3' or '5' for the respective Timer1/3/5 registers.

REGISTER 26-4: TMRxH(1): TIMERx HIGH BYTE 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
TMRxH<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 **TMRxH<7:0>:** TMRx High Byte bits

Note 1: 'x' refers to either '1', '3' or '5' for the respective Timer1/3/5 registers.

TABLE 26-5: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	100
PIR3	OSFIF	CSWIF	TMR3GIF	TMR3IF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	107
PIE3	OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	102
PIR4	CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	108
PIE4	CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	103
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1SOSC	$\overline{\text{T1SYNC}}$	—	TMR1ON	290
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		291
TMR1L	TMR1L<7:0>								292
TMR1H	TMR1H<7:0>								292
T1CKIPPS	—	—	—	T1CKIPPS<4:0>					160
T1GPPS	—	—	—	T1GPPS<4:0>					160
T3CON	TMR3CS<1:0>		T3CKPS<1:0>		T3SOSC	$\overline{\text{T3SYNC}}$	—	TMR3ON	290
T3GCON	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/$\overline{\text{DONE}}$	T3GVAL	T3GSS<1:0>		291
TMR3L	TMR3L<7:0>								292
TMR3H	TMR3H<7:0>								292
T3CKIPPS	—	—	—	T3CKIPPS<4:0>					160
T3GPPS	—	—	—	T3GPPS<4:0>					160
T5CON	TMR5CS<1:0>		T5CKPS<1:0>		T5SOSC	$\overline{\text{T5SYNC}}$	—	TMR5ON	290
T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/$\overline{\text{DONE}}$	T5GVAL	T5GSS<1:0>		291
TMR5L	TMR5L<7:0>								292
TMR5H	TMR5H<7:0>								292
T5CKIPPS	—	—	—	T5CKIPPS<4:0>					160
T5GPPS	—	—	—	T5GPPS<4:0>					160
T0CON0	T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>				278
CMxCON0	CxON	CxOUT	—	CxPOL	—	CxSP	CxHYS	CxSYNC	188
CCPTMRS	C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		310
CCPxCON	CCPxEN	—	CCPxOUT	CCPxFMT	CCPxMODE<3:0>				307
CLCxSELy	—	—	LCxDyS<5:0>						227
ADACT	—	—	—	ADACT<4:0>					244

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the Timer1 module.

Note 1: PIC16(L)F18345 only.

2: Unimplemented, read as '1'.

27.0 TIMER2/4/6 MODULE

Timer2/4/6 modules are 8-bit timers that incorporate the following features:

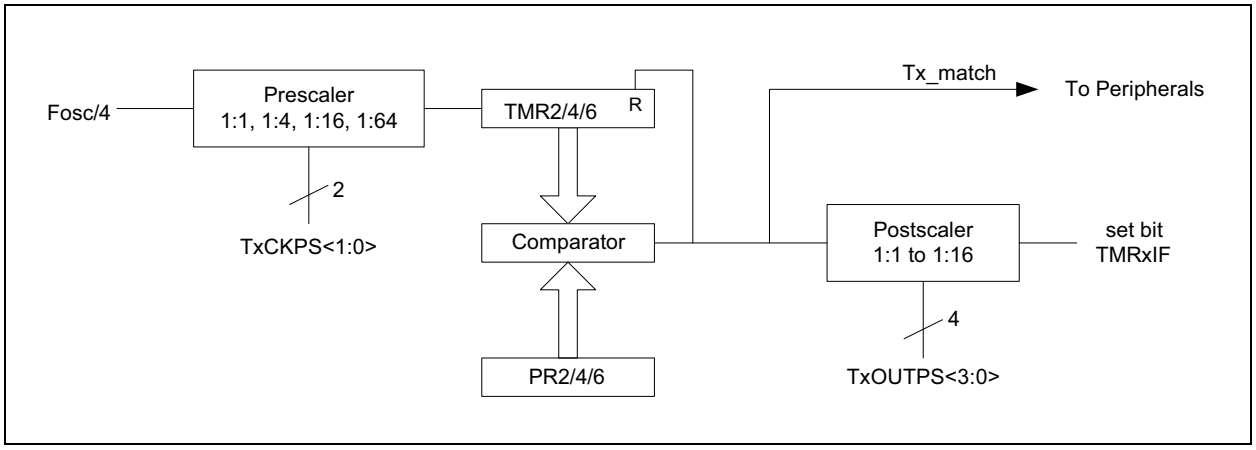
- 8-bit Timer and Period registers (TMR2/4/6 and PR2/4/6, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16, and 1:64)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR2/4/6 match with PR2/4/6
- Optional use as the shift clock for the MSSPx module

See Figure 27-1 for a block diagram of Timer2/4/6.

Note 1: In devices with more than one Timer module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the T2CON and T4CON control the same operational aspects of two completely different Timer modules.

2: Throughout this section, generic references to Timer2 module in any of its operating modes may be interpreted as being equally applicable to Timerx module. Register names, module signals, I/O pins and bit names may use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module, when required.

FIGURE 27-1: TIMER2/4/6 BLOCK DIAGRAM



27.1 Timer2 Operation

The clock input to the Timer2 modules is the system instruction clock (FOSC/4).

A 4-bit counter/prescaler on the clock input allows direct input, divide-by-4 and divide-by-16 prescale options. These options are selected by the prescaler control bits, T2CKPS<1:0> of the T2CON register. The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscaler (see **Section 27.2 "Timer2 Interrupt"**).

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, whereas the PR2 register initializes to FFh. Both the prescaler and postscaler counters are cleared on the following events:

- A write to the TMR2 register
- A write to the T2CON register
- Power-on Reset (POR)
- Brown-out Reset (BOR)
- $\overline{\text{MCLR}}$ Reset
- Watchdog Timer (WDT) Reset
- Stack Overflow Reset
- Stack Underflow Reset
- RESET Instruction

Note:	TMR2 is not cleared when T2CON is written.

27.2 Timer2 Interrupt

Timer2 can also generate an optional device interrupt. The Timer2 output signal (TMR2-to-PR2 match) provides the input for the 4-bit counter/postscaler. This counter generates the TMR2 match interrupt flag which is latched in TMR2IF of the PIR1 register. The interrupt is enabled by setting the TMR2 Match Interrupt Enable bit, TMR2IE, of the PIE1 register.

A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, T2OUTPS<3:0>, of the T2CON register.

27.3 Timer2 Output

The unscaled output of TMR2 is available primarily to the CCP modules, where it is used as a time base for operations in PWM mode.

Timer2 can be optionally used as the shift clock source for the MSSPx module operating in SPI mode. Additional information is provided in **Section 29.0 "Master Synchronous Serial Port (MSSPx) Module"**

27.4 Timer2 Operation During Sleep

The Timer2 timers cannot be operated while the processor is in Sleep mode. The contents of the TMR2 and PR2 registers will remain unchanged while the processor is in Sleep mode.

27.5 Register Definitions: Timer2/4/6 Control

REGISTER 27-1: TxCON(1): TIMERx 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
—	TxOUTPS<3:0>				TMRxON	TxCKPS<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 **TxOUTPS<3:0>:** Timerx 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 **TMRxON:** Timer2 On bit

1 = Timerx is on
0 = Timerx is off

bit 1-0 **TxCKPS<1:0>:** Timerx Clock Prescale Select bits

11 = Prescaler is 64
10 = Prescaler is 16
01 = Prescaler is 4
00 = Prescaler is 1

Note 1: 'x' refers to either '2,' 4' or '6' for the respective Timer2/4/6 registers.

REGISTER 27-2: TMRx(1): TIMERx COUNT REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMRx<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 **TMRx<7:0>:** TMRx Counter bits 7..0

Note 1: 'x' refers to either '2,' 4' or '6' for the respective Timer2/4/6 registers.

REGISTER 27-3: PRx: TIMERx PERIOD REGISTER(1)

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
PRx<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 **PRx<7:0>:** TMRx Counter bits 7..0
When TMRx = PRx, the next clock will reset the counter; counter period is (PRx+1)

Note 1: 'x' refers to either '2,' 4' or '6' for the respective Timer2/4/6 registers.

TABLE 27-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2/4/6

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	100
PIR2	TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BCL2IF	TMR4IF	NCO1IF	106
PIE2	TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BCL2IE	TMR4IE	NCO1IE	101
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		296
TMR2	TMR2<7:0>								297
PR2	PR2<7:0>								297
T4CON	—	T4OUTPS<3:0>				TMR4ON	T4CKPS<1:0>		296
TMR4	TMR4<7:0>								297
PR4	PR4<7:0>								297
T6CON	—	T6OUTPS<3:0>				TMR6ON	T6CKPS<1:0>		296
TMR6	TMR6<7:0>								297
PR6	PR6<7:0>								297
ADACT	—	—	—	ADACT<4:0>					244
PWMTMRS	—	—	—	—	P6TSEL<1:0>		P5TSEL<1:0>		195
CLCxSELy	—	—	LCxDyS<5:0>						227

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for Timer2/4/6 module.

28.0 CAPTURE/COMPARE/PWM MODULES

The Capture/Compare/PWM module is a peripheral that allows the user to time and control different events and to generate Pulse-Width Modulation (PWM) signals. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate Pulse-Width Modulated signals of varying frequency and duty cycle.

This family of devices contains four standard Capture/Compare/PWM modules (CCP1, CCP2, CCP3 and CCP4).

The Capture and Compare functions are identical for all CCP modules.

28.1 CCP/PWM Clock Selection

The PIC16(L)F18325/18345 devices allow each individual CCP and PWM module to select the timer source that controls the module. Each module has an independent selection.

As there are up to three 8-bit timers with auto-reload (Timer2, Timer4, and Timer6), PWM mode on the CCP and PWM modules can use any of these timers.

The CCPTMRS register is used to select which timer is used.

Note 1:	In devices with more than one CCP module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the CCP1CON and CCP2CON control the same operational aspects of two completely different CCP modules.
2:	Throughout this section, generic references to a CCP module in any of its operating modes may be interpreted as being equally applicable to CCPx module. Register names, module signals, I/O pins, and bit names may use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module, when required.

28.2 Capture Mode

The Capture mode function described in this section is available and identical for all CCP modules.

Capture mode makes use of either the 16-bit Timer0 or Timer1 resource. When an event occurs on the capture source, the 16-bit CCPRxH:CCPRxL register pair captures and stores the 16-bit value of the TMR0H:TMR0L or TMR1H:TMR1L register pair, respectively. An event is defined as one of the following and is configured by the CCPxMODE<3:0> bits of the CCPxCON register:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

When a capture is made, the Interrupt Request Flag bit CCPxIF of the PIR4 register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPRxH, CCPRxL register pair is read, the old captured value is overwritten by the new captured value.

Figure 28-1 shows a simplified diagram of the capture operation.

28.2.1 CAPTURE SOURCES

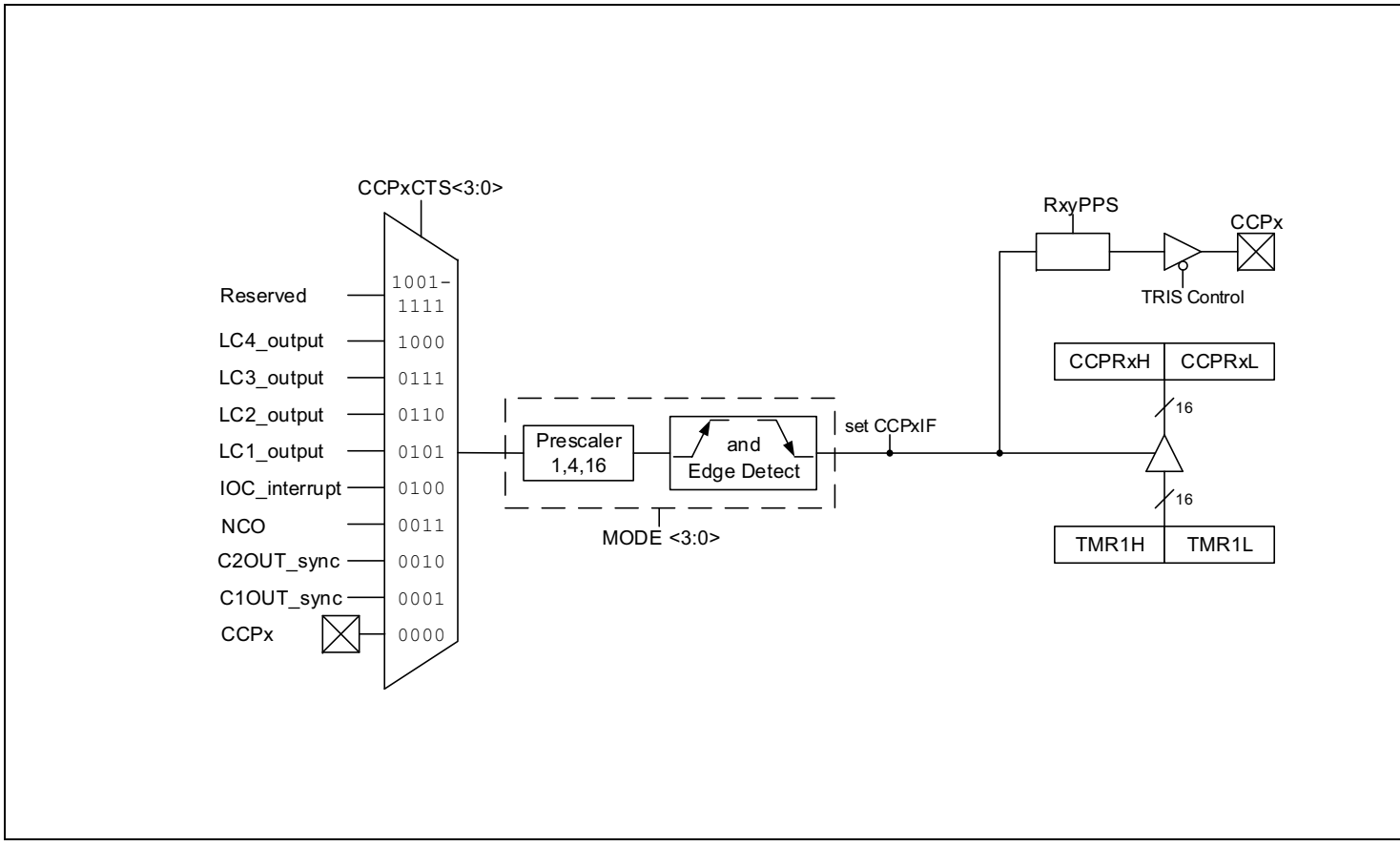
In Capture mode, the CCPx pin should be configured as an input by setting the associated TRIS control bit.

Note:	If the CCPx pin is configured as an output, a write to the port can cause a Capture condition.

The capture source is selected by configuring the CCPxCTS<3:0> bits of the CCPxCAP register. The following sources can be selected:

- CCPxPPS input
- C1_output
- C2_output
- NCO_output
- IOC_interrupt
- LC1_output
- LC2_output
- LC3_output
- LC4_output

FIGURE 28-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



28.2.2 TIMER1/3/5 MODE RESOURCE

Timer1/3/5 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

See **Section 26.0 "Timer1/3/5 Module with Gate Control"** for more information on configuring Timer1/3/5.

28.2.3 SOFTWARE INTERRUPT MODE

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE interrupt enable bit of the PIE4 register clear to avoid false interrupts. Additionally, the user should clear the CCPxIF interrupt flag bit of the PIR4 register following any change in Operating mode.

Note:	Clocking Timer1/3/5 from the system clock (FOSC) should not be used in Capture mode. In order for Capture mode to recognize the trigger event on the CCPx pin, Timer1/3/5 must be clocked from the instruction clock (FOSC/4) or from an external clock source.

28.2.4 CCP PRESCALER

There are four prescaler settings specified by the CCPxMODE<3:0> bits of the CCPxCON register. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. Any Reset will clear the prescaler counter.

Switching from one capture prescaler to another does not clear the prescaler and may generate a false interrupt. To avoid this unexpected operation, turn the module off by clearing the CCPxCON register before changing the prescaler. Example 28-1 demonstrates the code to perform this function.

EXAMPLE 28-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
BANKSEL CCPxCON     ;Set Bank bits to point
                    ;to CCPxCON
CLRF    CCPxCON     ;Turn CCP module off
MOVLW   NEW_CAPT_PS ;Load the W reg with
                    ;the new prescaler
                    ;move value and CCP ON
MOVWF   CCPxCON     ;Load CCPxCON with this
                    ;value
```

28.2.5 CAPTURE DURING SLEEP

Capture mode depends upon the Timer1/3/5 module for proper operation. There are two options for driving the Timer1/3/5 module in Capture mode. It can be driven by the instruction clock (FOSC/4), or by an external clock source.

When Timer1/3/5 is clocked by FOSC/4, Timer1/3/5 will not increment during Sleep. When the device wakes from Sleep, Timer1/3/5 will continue from its previous state.

Capture mode will operate during Sleep when Timer1/3/5 is clocked by an external clock source.

28.3 Compare Mode

The Compare mode function described in this section is available and identical for all CCP modules.

Compare mode makes use of the 16-bit Timer1/3/5 resource. The 16-bit value of the CCPRxH:CCPRxL register pair is constantly compared·against the 16-bit value of the TMR1/3/5H:TMR1/3/5L register pair. When a match occurs, one of the following events can occur:

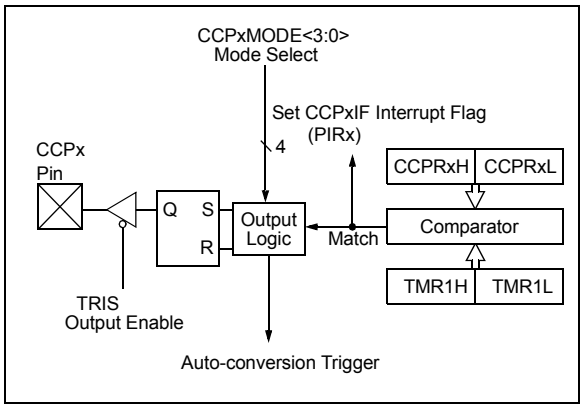
- Toggle the CCPx output
- Set the CCPx output
- Clear the CCPx output
- Generate an Auto-conversion Trigger
- Generate a Software Interrupt

The action on the pin is based on the value of the CCPxMODE<3:0> control bits of the CCPxCON register. At the same time, the interrupt flag CCPxIF bit is set, and an ADC conversion can be triggered, if selected.

All Compare modes can generate an interrupt and trigger an ADC conversion.

Figure 28-2 shows a simplified diagram of the compare operation.

FIGURE 28-2: COMPARE MODE OPERATION BLOCK DIAGRAM



28.3.1 CCPX PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the associated TRIS bit and defining the appropriate output pin through the RxyPPS registers. See **Section 12.0 "Peripheral Pin Select (PPS) Module"** for more details.

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

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

28.3.2 TIMER1/3/5 MODE RESOURCE

In Compare mode, Timer1/3/5 must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

See **Section 26.0 "Timer1/3/5 Module with Gate Control"** for more information on configuring Timer1/3/5.

Note:	Clocking Timer1/3/5 from the system clock (FOSC) should not be used in Compare mode. In order for Compare mode to recognize the trigger event on the CCPx pin, Timer1/3/5 must be clocked from the instruction clock (FOSC/4) or from an external clock source.

28.3.3 AUTO-CONVERSION TRIGGER

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

Refer to **Section 21.2.5 "Auto-Conversion Trigger"** for more information.

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

28.3.4 COMPARE DURING SLEEP

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

28.4 PWM Overview

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

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

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

Figure 28-3 shows a typical waveform of the PWM signal.

28.4.1 STANDARD PWM OPERATION

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

The standard PWM mode generates a Pulse-Width Modulation (PWM) signal on the CCPx pin with up to 10 bits of resolution. The period, duty cycle, and resolution are controlled by the following registers:

- PR2/4/6 registers
- T2/4/6CON registers
- CCPRxL registers
- CCPxCON registers

Figure 28-4 shows a simplified block diagram of PWM operation.

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

FIGURE 28-3: CCP PWM OUTPUT SIGNAL

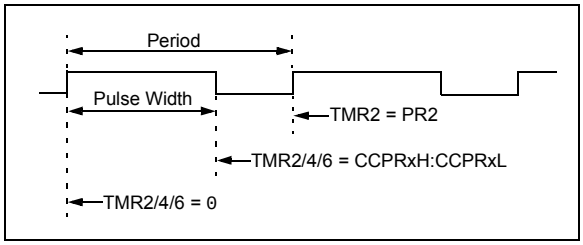
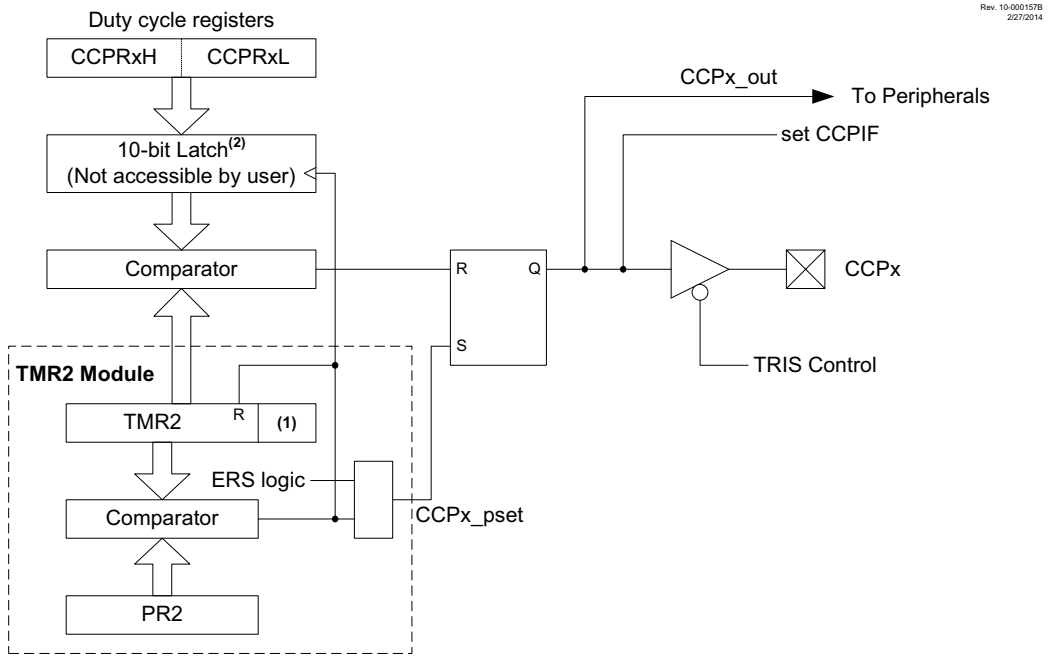


FIGURE 28-4: SIMPLIFIED PWM BLOCK DIAGRAM



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

28.4.2 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for standard PWM operation:

1. Use the desired output pin RxyPPS control to select CCPx as the source and disable the CCPx pin output driver by setting the associated TRIS bit.
2. Load the PR2 register with the PWM period value.
3. Configure the CCP module for the PWM mode by loading the CCPxCON register with the appropriate values.
4. Load the CCPRxL register and the CCPRxH register bits, with the PWM duty cycle value and configure the CCPxFMT bit of the CCPxCON register to set the proper register alignment.
5. Configure and start Timer2, 4 or 6.
   - Clear the TMR2/4/6IF interrupt flag bits of the PIR4 register. See Note below.
   - Configure the T2/4/6CKPS bits of the T2/4/6CON register with the Timer prescale value.
   - Enable the Timer by setting the TMR2/4/6ON bit of the T2/4/6CON register.
6. Enable PWM output pin:
   - Wait until the Timer overflows and the TMR2/4/6IF bits of the PIR4 register is set. See Note below.
   - Enable the CCPx pin output driver by clearing the associated TRIS bit.

Note:	In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then step 6 may be ignored.

28.4.3 TIMER2/4/6 TIMER RESOURCE

The PWM standard mode makes use of the 8-bit Timer2/4/6 timer resources to specify the PWM period.

28.4.4 PWM PERIOD

The PWM period is specified by the PRx register of Timer2/4/6. The PWM period can be calculated using the formula of Equation 28-1.

EQUATION 28-1: PWM PERIOD

$$PWM\ Period = [(PR2x) + 1] \bullet 4 \bullet TOSC \bullet (TMR2/4/6\ Prescale\ Value)$$

Note: TOSC = 1/FOSC

When TMR2/4/6 is equal to PR2/4/6, the following three events occur on the next increment cycle:

- TMR2/4/6 is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is transferred from the CCPRxL/H register pair into a 10-bit buffer.

Note:	The Timer postscaler (see **Section 27.1 "Timer2 Operation"**) is not used in the determination of the PWM frequency.

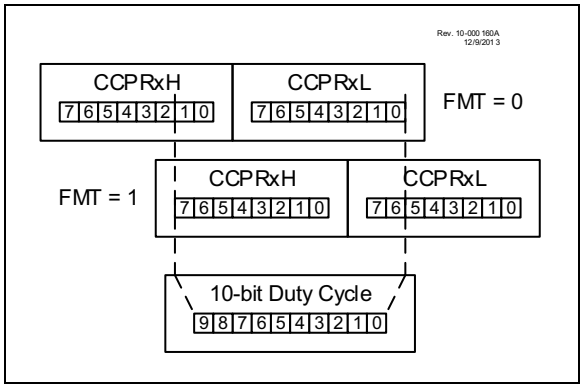
28.4.5 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to the CCPRxH:CCPRxL register pair. The alignment of the 10-bit value is determined by the CCPRxFMT bit of the CCPxCON register (see Figure 28-5). The CCPRxH:CCPRxL register pair can be written to at any time; however, the duty cycle value is not latched into the 10-bit buffer until after a match between PR2/4/6 and TMR2/4/6.

Equation 28-2 is used to calculate the PWM pulse width.

Equation 28-3 is used to calculate the PWM duty cycle ratio.

FIGURE 28-5: PWM 10-BIT ALIGNMENT BLOCK DIAGRAM



EQUATION 28-2: PULSE WIDTH

$$Pulse\ Width = (CCPRxH{:}CCPRxL\ register\ pair) \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

EQUATION 28-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(CCPRxH{:}CCPRxL\ register\ pair)}{4(PR2 + 1)}$$

The CCPRxH:CCPRxL register pair and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

The 8-bit timer TMR2/4/6 register is concatenated with either the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2/4/6 prescaler is set to 1:1.

When the 10-bit time base matches the CCPRxH:CCPRxL register pair, then the CCPx pin is cleared (see Figure 28-4).

28.4.6 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 ten bits when PRx is 255. The resolution is a function of the PRx register value as shown by Equation 28-4.

EQUATION 28-4: PWM RESOLUTION

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

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

TABLE 28-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 20 MHz)

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale	16	4	1	1	1	1
PRx Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

TABLE 28-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 8 MHz)

PWM Frequency	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale	16	4	1	1	1	1
PRx Value	0x65	0x65	0x65	0x19	0x0C	0x09
Maximum Resolution (bits)	8	8	8	6	5	5

28.4.7 OPERATION IN SLEEP MODE

In Sleep mode, the TMR2/4/6 register will not increment and the state of the module will not change. If the CCPx pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2/4/6 will continue from its previous state.

28.4.8 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 6.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for additional details.

28.4.9 EFFECTS OF RESET

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

28.5 Register Definitions: CCP Control

REGISTER 28-1: CCPxCON: CCPx CONTROL REGISTER

R/W-0/0	U-0	R-x/x	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CCPxEN	—	CCPxOUT	CCPxFMT	CCPxMODE<3:0>			
bit 7							bit 0

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

bit 7 **CCPxEN:** CCP Module Enable bit
0 = CCP is disabled
1 = CCP is enabled

bit 6 **Unimplemented:** Read as '0'

bit 5 **CCPxOUT:** CCPx Output Data (read-only) bit

bit 4 **CCPxFMT:** CCPW (pulse width) Alignment bit
CCPxMODE = Capture mode
Unused
CCPxMODE = Compare mode
Unused
CCPxMODE = PWM mode
0 = Right-aligned format
1 = Left-aligned format

bit 3-0 **CCPxMODE<3:0>:** CCPx Mode Select bits(1)
1111 = PWM mode
1110 = Reserved
1101 = Reserved
1100 = Reserved

1011 = Compare mode: output will pulse 0-1-0; Clears TMR1/3/5
1010 = Compare mode: output will pulse 0-1-0
1001 = Compare mode: clear output on compare match
1000 = Compare mode: set output on compare match

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

0011 = Capture mode: every edge of CCPx input
0010 = Compare mode: toggle output on match
0001 = Compare mode: toggle output on match; clear TMR1/3/5
0000 = Capture/Compare/PWM off (resets CCPx module)

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

REGISTER 28-2: CCPxCAP: CAPTURE INPUT SELECTION REGISTER

U-0	U-0	U-0	U-0	R/W-0/x	R/W-0/x	R/W-0/x	R/W-0/x
—	—	—	—	CCPxCTS<3:0>			
bit 7							bit 0

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

bit 7-4 **Unimplemented:** Read as '0'

bit 3-0 **CCPxCTS<3:0>:** CCPx Capture Mode Data Select bits

CCAP<3:0>	CCP1CAP CAPTURE INPUT	CCP2CAP CAPTURE INPUT	CCP3CAP CAPTURE INPUT	CCP4CAP CAPTURE INPUT
0000	CCP1PPS	CCP2PPS	CCP3PPS	CCP4PPS
0001	C1OUT			
0010	C2OUT			
0011	NCO1			
0100	IOC_interrupt			
0101	LC1_output			
0110	LC2_output			
0111	LC3_output			
1000	LC4_output			
1001 … 1111	Reserved			

REGISTER 28-3: CCPRxL REGISTER: CCPx REGISTER LOW BYTE

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
CCPRxL<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 Reset
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0

CCPxMODE = Capture mode
CCPRxL<7:0>: Captured value of TMR1/3/5L
CCPxMODE = Compare mode
CCPRxL<7:0>: LS Byte compared to TMR1/3/5L
CCPxMODE = PWM modes when CCPxFMT = 0
CCPRxL<7:0>: CCPW<7:0> – Pulse-width Least Significant eight bits
CCPxMODE = PWM modes when CCPxFMT = 1
CCPRxL<7:6>: CCPW<1:0> – Pulse-width Least Significant two bits
CCPRxL<5:0>: Not used.

REGISTER 28-4: CCPRxH REGISTER: CCPx REGISTER HIGH BYTE

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
CCPRxH<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 Reset
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0

CCPxMODE = Capture mode
CCPRxH<7:0>: Captured value of TMR1/3/5H
CCPxMODE = Compare mode
CCPRxH<7:0>: MS Byte compared to TMR1/3/5H
CCPxMODE = PWM modes when CCPxFMT = 0
CCPRxH<7:2>: Not used
CCPRxH<1:0>: CCPW<9:8> – Pulse-width Most Significant two bits
CCPxMODE = PWM modes when CCPxFMT = 1
CCPRxH<7:0>: CCPW<9:2> – Pulse-width Most Significant eight bits

REGISTER 28-5: CCPTMRS: CCP TIMERS CONTROL REGISTER

R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1
C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<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 Reset
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **C4TSEL<1:0>:** CCP4 Capture, Compare and PWM mode Timer Selection bits
Selection as show in Table 28-4.

bit 5-4 **C3TSEL<1:0>:** CCP3 Capture, Compare and PWM mode Timer Selection bits
Selection as show in Table 28-4.

bit 3-2 **C2TSEL<1:0>:** CCP2 Capture, Compare and PWM mode Timer Selection bits
Selection as show in Table 28-4.

bit 1-0 **C1TSEL<1:0>:** CCP1 Capture, Compare and PWM mode Timer Selection bits
Selection as show in Table 28-4.

TABLE 28-3: TIMER SELECTIONS

CxTSEL<1:0>	**Operating mode based on CCPxMODE<3:0>**	
	Capture Compare	**PWM**
00	TMR0	TMR2
01	TMR1	
10	TMR3	TMR4
11	TMR5	TMR6

TABLE 28-4: SUMMARY OF REGISTERS ASSOCIATED WITH CCPx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
PIR4	CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	108
PIE4	CWG2IE	CWGIE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	103
CCPxCON	CCPxEN	—	CCPxOUT	CCPxFMT	CCPxMODE<3:0>				307
CCPxCAP	—	—	—	—	—	CCPxCTS<2:0>			308
CCPRxL	CCPRx<7:0>								309
CCPRxH	CCPRx<15:8>								309
CCPTMRS	C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		310
CCP1PPS	—	—	—	CCP1PPS<4:0>					160
CCP2PPS	—	—	—	CCP2PPS<4:0>					160
CCP3PPS	—	—	—	CCP3PPS<4:0>					160
CCP4PPS	—	—	—	CCP4PPS<4:0>					160
RxyPPS	—	—	—	RxyPPS<4:0>					161
ADACT	—	—	—	ADACT<4:0>					244
CLCxSELy	—	—	LCxDyS<5:0>						227
CWGxDAT	—	—	—	—	DAT<3:0>				213
MDSRC	—	—	—	—	MDMS<3:0>				270
MDCARH	—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>				271
MDCARL	—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				272

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by the CCP module.

Note 1: PIC16(L)F18345 only.

2: Unimplemented, read as '1'.

29.0 MASTER SYNCHRONOUS SERIAL PORT (MSSPx) MODULE

29.1 MSSPx Module Overview

The Master Synchronous Serial Port (MSSPx) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D Converters, etc. The MSSPx module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)

The SPI interface supports the following modes and features:

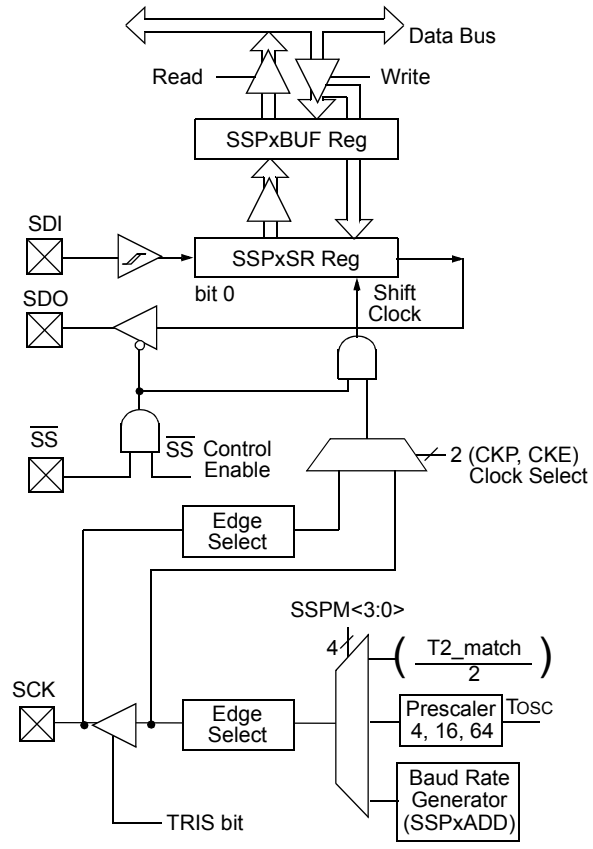
- Master mode
- Slave mode
- Clock Parity
- Slave Select Synchronization (Slave mode only)
- Daisy-chain connection of slave devices

Figure 29-1 is a block diagram of the SPI interface module.

Note 1: In devices with more than one MSSP module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the SSP1STAT and SSP2STAT control the same operational aspects of two completely different MSSP modules.

2: Throughout this section, generic references to the MSSP1 module in any of its operating modes may be interpreted as being equally applicable to MSSPx module. Register names, module signals, I/O pins, and bit names may use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module, when required.

FIGURE 29-1: MSSP BLOCK DIAGRAM (SPI MODE)



The I²C interface supports the following modes and features:

- Master mode
- Slave mode
- Byte NACKing (Slave mode)
- Limited multi-master support
- 7-bit and 10-bit addressing
- Start and Stop interrupts
- Interrupt masking
- Clock stretching
- Bus collision detection
- General call address matching
- Address masking
- Address Hold and Data Hold modes
- Selectable SDA hold times

Figure 29-2 is a block diagram of the I²C interface module in Master mode. Figure 29-3 is a diagram of the I²C interface module in Slave mode.

FIGURE 29-2: MSSP BLOCK DIAGRAM (I²C™ MASTER MODE)

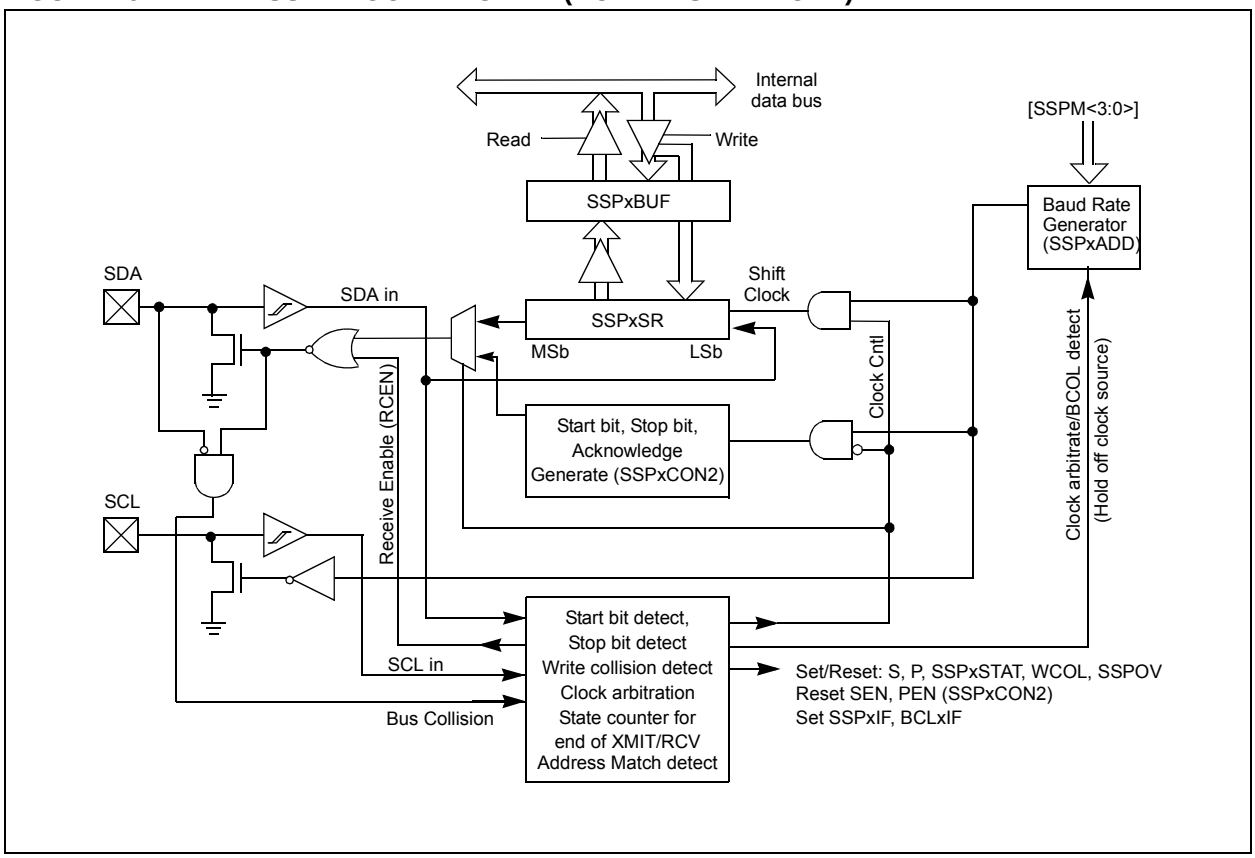
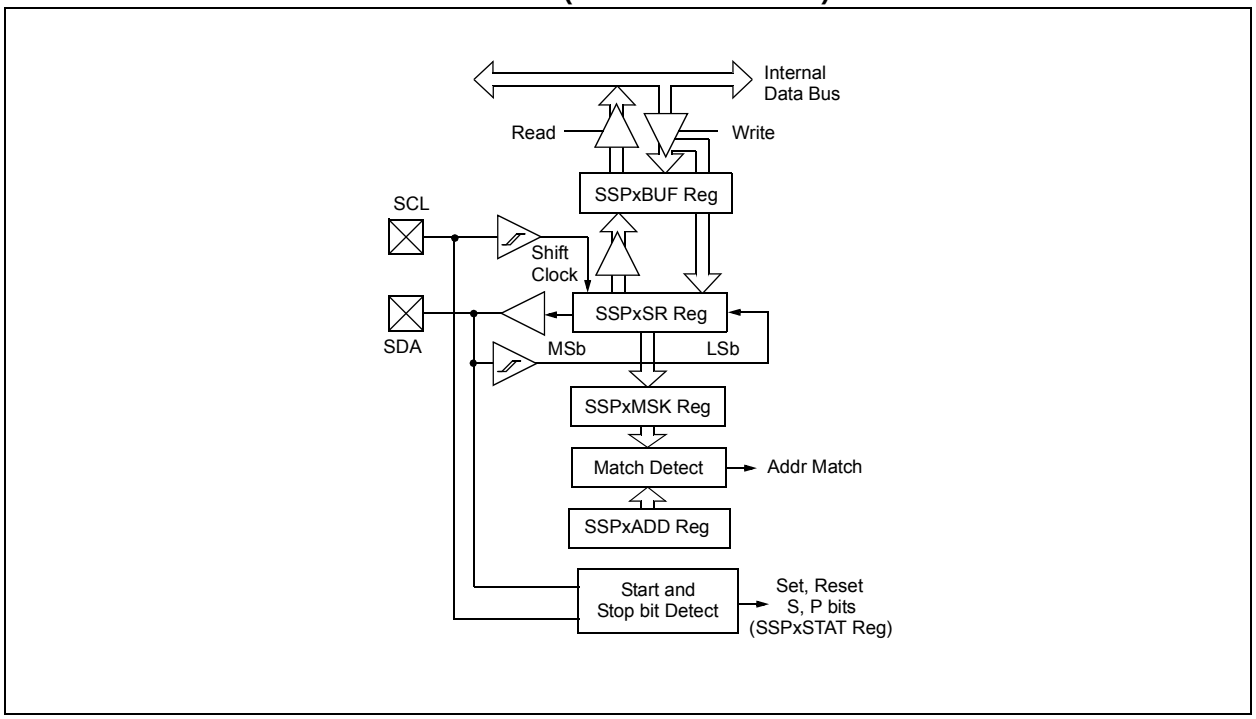


FIGURE 29-3: MSSP BLOCK DIAGRAM (I^2C™ SLAVE MODE)



29.2 SPI Mode Overview

The Serial Peripheral Interface (SPI) bus is a synchronous serial data communication bus that operates in Full-Duplex mode. Devices communicate in a master/slave environment where the master device initiates the communication. A slave device is controlled through a Chip Select known as Slave Select.

The SPI bus specifies four signal connections:

- Serial Clock (SCK)
- Serial Data Out (SDO)
- Serial Data In (SDI)
- Slave Select ($\overline{SS}$)

Figure 29-1 shows the block diagram of the MSSPx module when operating in SPI mode.

The SPI bus operates with a single master device and one or more slave devices. When multiple slave devices are used, an independent Slave Select connection is required from the master device to each slave device.

Figure 29-4 shows a typical connection between a master device and multiple slave devices.

The master selects only one slave at a time. Most slave devices have tri-state outputs so their output signal appears disconnected from the bus when they are not selected.

Transmissions involve two shift registers, eight bits in size, one in the master and one in the slave. With either the master or the slave device, data is always shifted out one bit at a time, with the Most Significant bit (MSb) shifted out first. At the same time, a new Least Significant bit (LSb) is shifted into the same register.

Figure 29-5 shows a typical connection between two processors configured as master and slave devices.

Data is shifted out of both shift registers on the programmed clock edge and latched on the opposite edge of the clock.

The master device transmits information out on its SDO output pin which is connected to, and received by, the slave's SDI input pin. The slave device transmits information out on its SDO output pin, which is connected to, and received by, the master's SDI input pin.

To begin communication, the master device first sends out the clock signal. Both the master and the slave devices should be configured for the same clock polarity.

The master device starts a transmission by sending out the MSb from its shift register. The slave device reads this bit from that same line and saves it into the LSb position of its shift register.

During each SPI clock cycle, a full-duplex data transmission occurs. This means that while the master device is sending out the MSb from its shift register (on its SDO pin) and the slave device is reading this bit and saving it as the LSb of its shift register, that the slave device is also sending out the MSb from its shift register (on its SDO pin) and the master device is reading this bit and saving it as the LSb of its shift register.

After eight bits have been shifted out, the master and slave have exchanged register values.

If there is more data to exchange, the shift registers are loaded with new data and the process repeats itself.

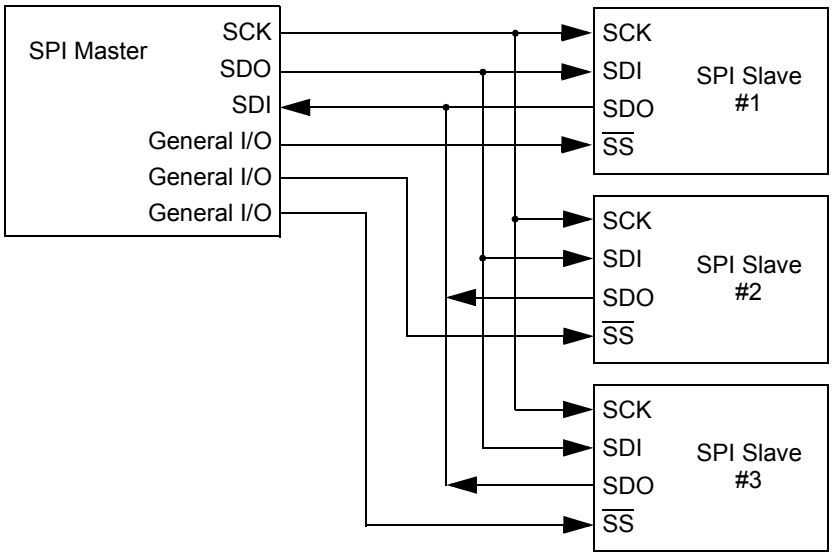
Whether the data is meaningful or not (dummy data), depends on the application software. This leads to three scenarios for data transmission:

- Master sends useful data and slave sends dummy data.
- Master sends useful data and slave sends useful data.
- Master sends dummy data and slave sends useful data.

Transmissions may involve any number of clock cycles. When there is no more data to be transmitted, the master stops sending the clock signal and it deselects the slave.

Every slave device connected to the bus that has not been selected through its slave select line must disregard the clock and transmission signals and must not transmit out any data of its own.

FIGURE 29-4: SPI MASTER AND MULTIPLE SLAVE CONNECTION



29.2.1 SPI MODE REGISTERS

The MSSPx module has five registers for SPI mode operation. These are:

- MSSPx STATUS register (SSPxSTAT)
- MSSPx Control register 1 (SSPxCON1)
- MSSPx Control register 3 (SSPxCON3)
- MSSPx Data Buffer register (SSPxBUF)
- MSSPx Address register (SSPxADD)
- MSSPx Shift register (SSPxSR) (Not directly accessible)

SSPxCON1 and SSPSTAT are the control and STATUS registers in SPI mode operation. The SSPxCON1 register is readable and writable. The lower six bits of the SSPxSTAT are read-only. The upper two bits of the SSPxSTAT are read/write.

In one SPI Master mode, SSPxADD can be loaded with a value used in the Baud Rate Generator. More information on the Baud Rate Generator is available in **Section 29.7 "Baud Rate Generator"**.

SSPxSR is the shift register used for shifting data in and out. SSPxBUF provides indirect access to the SSPxSR register. SSPxBUF is the buffer register to which data bytes are written, and from which data bytes are read.

In receive operations, SSPxSR and SSPxBUF together create a buffered receiver. When SSPxSR receives a complete byte, it is transferred to SSPxBUF and the SSPxIF interrupt is set.

During transmission, the SSPxBUF is not buffered. A write to SSPxBUF will write to both SSPxBUF and SSPxSR.

29.2.2 SPI MODE OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPxCON1<5:0> and SSPxSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

To enable the serial port, SSP Enable bit, SSPEN of the SSPxCON1 register, must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPxCONy registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

- SDI must have corresponding TRIS bit set
- SDO must have corresponding TRIS bit cleared
- SCK (Master mode) must have corresponding TRIS bit cleared
- SCK (Slave mode) must have corresponding TRIS bit set
- $\overline{SS}$ must have corresponding TRIS bit set

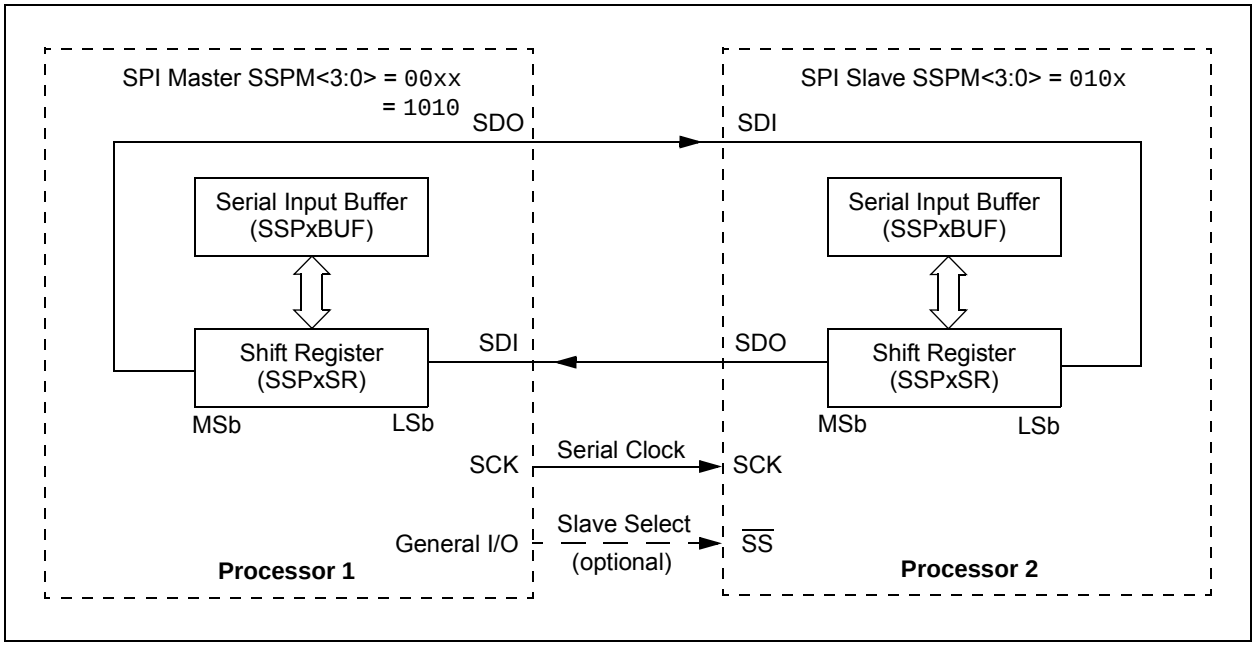
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

The MSSPx consists of a transmit/receive shift register (SSPxSR) and a buffer register (SSPxBUF). The SSPxSR shifts the data in and out of the device, MSb first. The SSPxBUF holds the data that was written to the SSPxSR until the received data is ready. Once the eight bits of data have been received, that byte is moved to the SSPxBUF register. Then, the Buffer Full Detect bit, BF of the SSPxSTAT register, and the interrupt flag bit, SSPxIF, are set. This double-buffering of the received data (SSPxBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPxBUF register during transmission/reception of data will be ignored and the write collision detect bit, WCOL, of the SSPxCON1 register, will be set. User software must clear the WCOL bit to allow the following write(s) to the SSPxBUF register to complete successfully.

When the application software is expecting to receive valid data, the SSPxBUF should be read before the next byte of data to transfer is written to the SSPxBUF. The Buffer Full bit, BF of the SSPxSTAT register, indicates when SSPxBUF has been loaded with the received data (transmission is complete). When the SSPxBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur.

The SSPxSR is not directly readable or writable and can only be accessed by addressing the SSPxBUF register. Additionally, the SSPxSTAT register indicates the various Status conditions.

FIGURE 29-5: SPI MASTER/SLAVE CONNECTION



29.2.3 SPI MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK line. The master determines when the slave (Processor 2, Figure 29-5) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPxBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPxSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPxBUF register as if a normal received byte (interrupts and Status bits appropriately set).

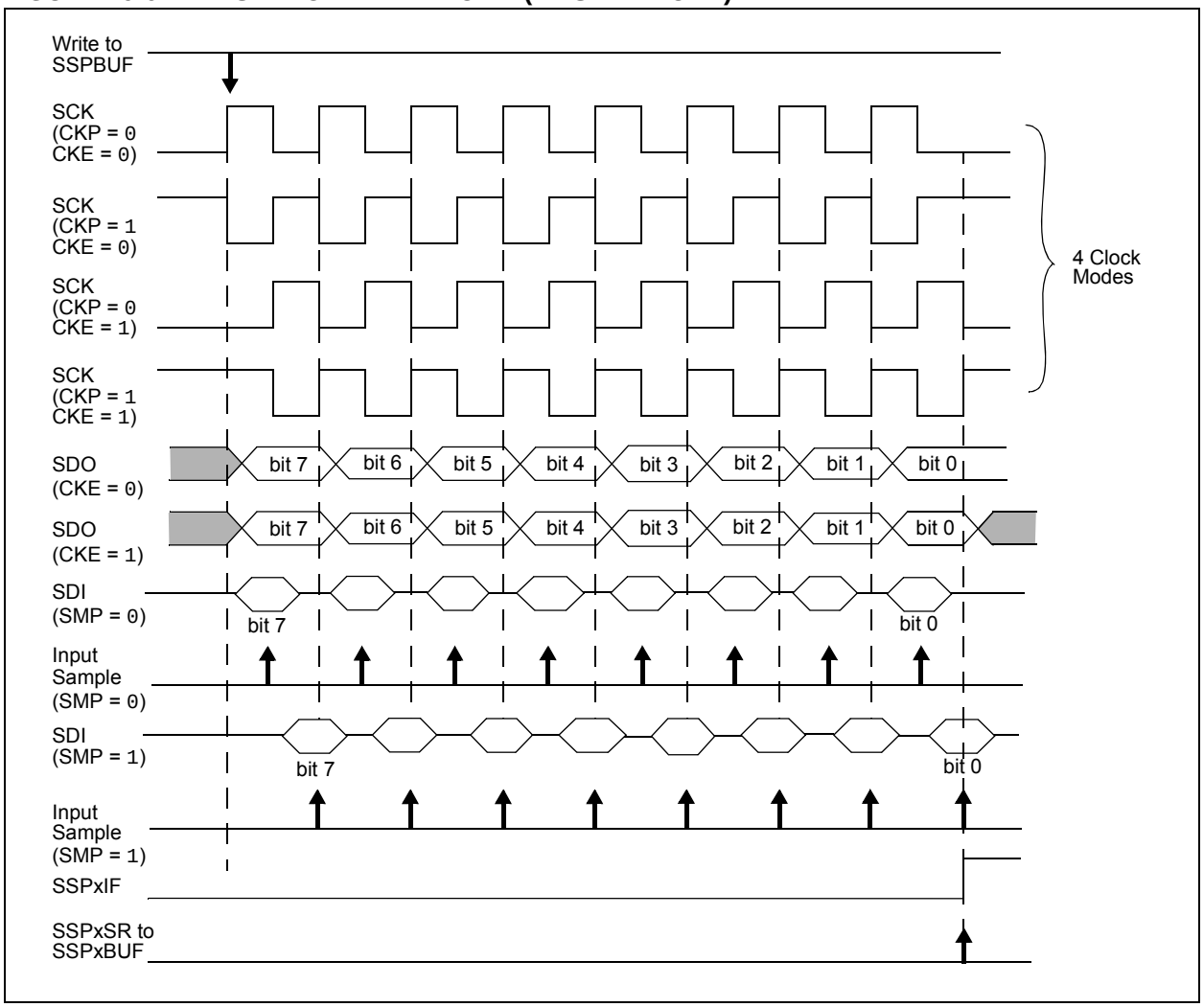
The clock polarity is selected by appropriately programming the CKP bit of the SSPxCON1 register and the CKE bit of the SSPxSTAT register. This then, would give waveforms for SPI communication as shown in Figure 29-6, Figure 29-8, Figure 29-9 and Figure 29-10, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- FOSC/4 (or TCY)
- FOSC/16 (or 4 * TCY)
- FOSC/64 (or 16 * TCY)
- Timer2 output/2
- FOSC/(4 * (SSPADD + 1))

Figure 29-6 shows the waveforms for Master mode.

When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPxBUF is loaded with the received data is shown.

FIGURE 29-6: SPI MODE WAVEFORM (MASTER MODE)



29.2.4 SPI SLAVE MODE

In Slave mode, the data is transmitted and received as external clock pulses appear on SCK. When the last bit is latched, the SSPxIF interrupt flag bit is set.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit of the SSPxCON1 register.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. The shift register is clocked from the SCK pin input and when a byte is received, the device will generate an interrupt. If enabled, the device will wake-up from Sleep.

29.2.4.1 Daisy-Chain Configuration

The SPI bus can sometimes be connected in a daisy-chain configuration. The first slave output is connected to the second slave input, the second slave output is connected to the third slave input, and so on. The final slave output is connected to the master input. Each slave sends out, during a second group of clock pulses, an exact copy of what was received during the first group of clock pulses. The whole chain acts as one large communication shift register. The daisy-chain feature only requires a single Slave Select line from the master device.

Figure 29-7 shows the block diagram of a typical daisy-chain connection when operating in SPI mode.

In a daisy-chain configuration, only the most recent byte on the bus is required by the slave. Setting the BOEN bit of the SSPxCON3 register will enable writes to the SSPxBUF register, even if the previous byte has not been read. This allows the software to ignore data that may not apply to it.

29.2.5 SLAVE SELECT SYNCHRONIZATION

The Slave Select can also be used to synchronize communication. The Slave Select line is held high until the master device is ready to communicate. When the Slave Select line is pulled low, the slave knows that a new transmission is starting.

If the slave fails to receive the communication properly, it will be reset at the end of the transmission, when the Slave Select line returns to a high state. The slave is then ready to receive a new transmission when the Slave Select line is pulled low again. If the Slave Select line is not used, there is a risk that the slave will eventually become out of sync with the master. If the slave misses a bit, it will always be one bit off in future transmissions. Use of the Slave Select line allows the slave and master to align themselves at the beginning of each transmission.

The $\overline{SS}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{SS}$ pin control enabled (SSPxCON1<3:0> = 0100).

When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven.

When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

Note 1:	When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPxCON1<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to VDD.
2:	When the SPI is used in Slave mode with CKE set; the user must enable $\overline{SS}$ pin control.
3:	While operated in SPI Slave mode the SMP bit of the SSPxSTAT register must remain clear.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the $\overline{SS}$ pin to a high level or clearing the SSPEN bit.

FIGURE 29-7: SPI DAISY-CHAIN CONNECTION

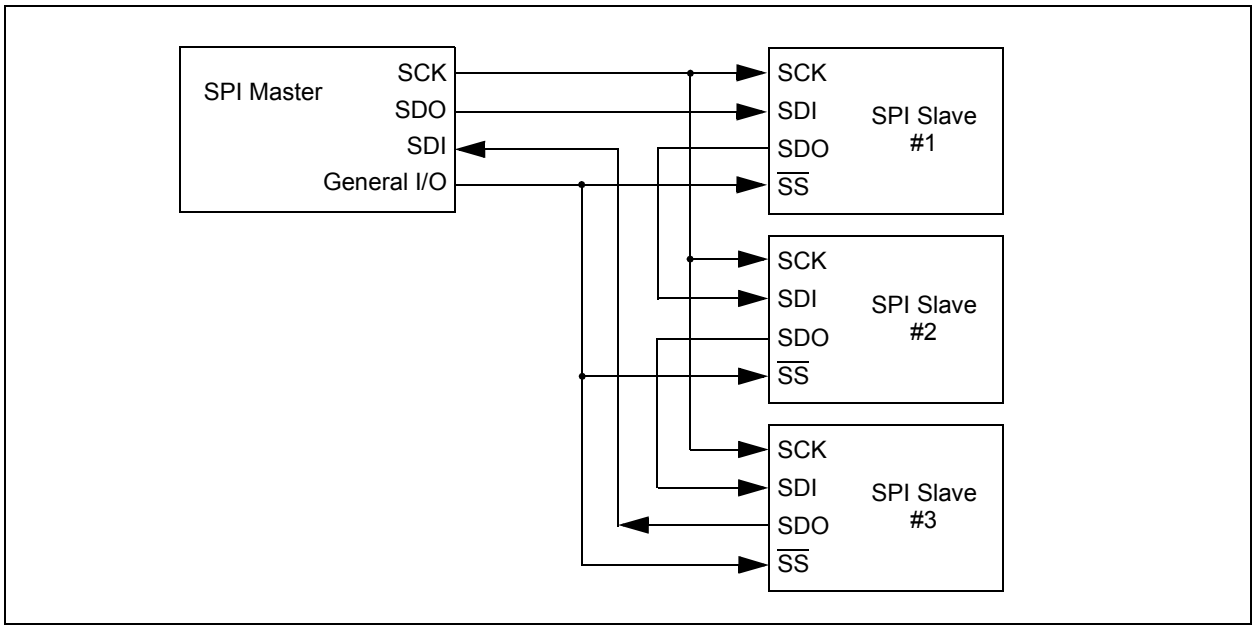


FIGURE 29-8: SLAVE SELECT SYNCHRONOUS WAVEFORM

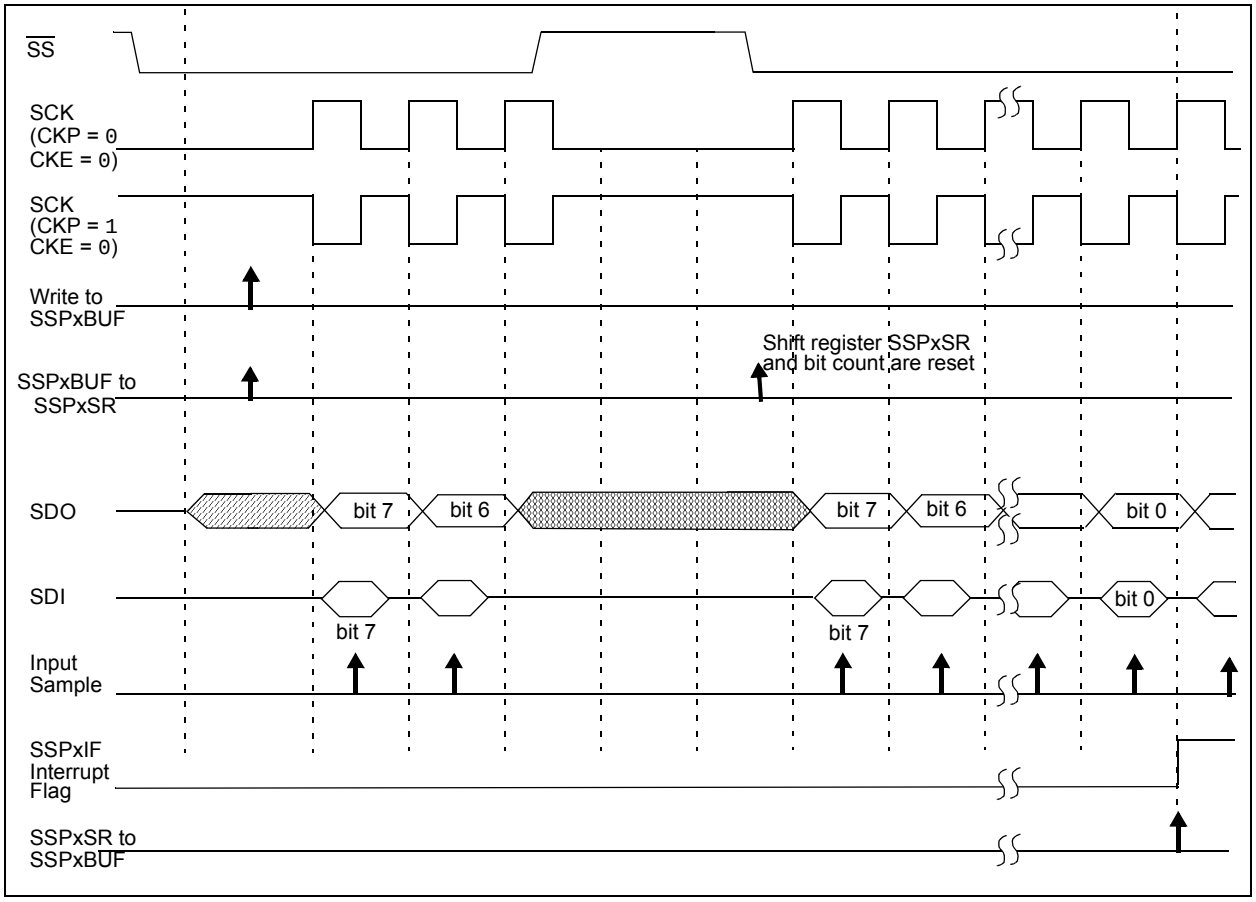


FIGURE 29-9: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

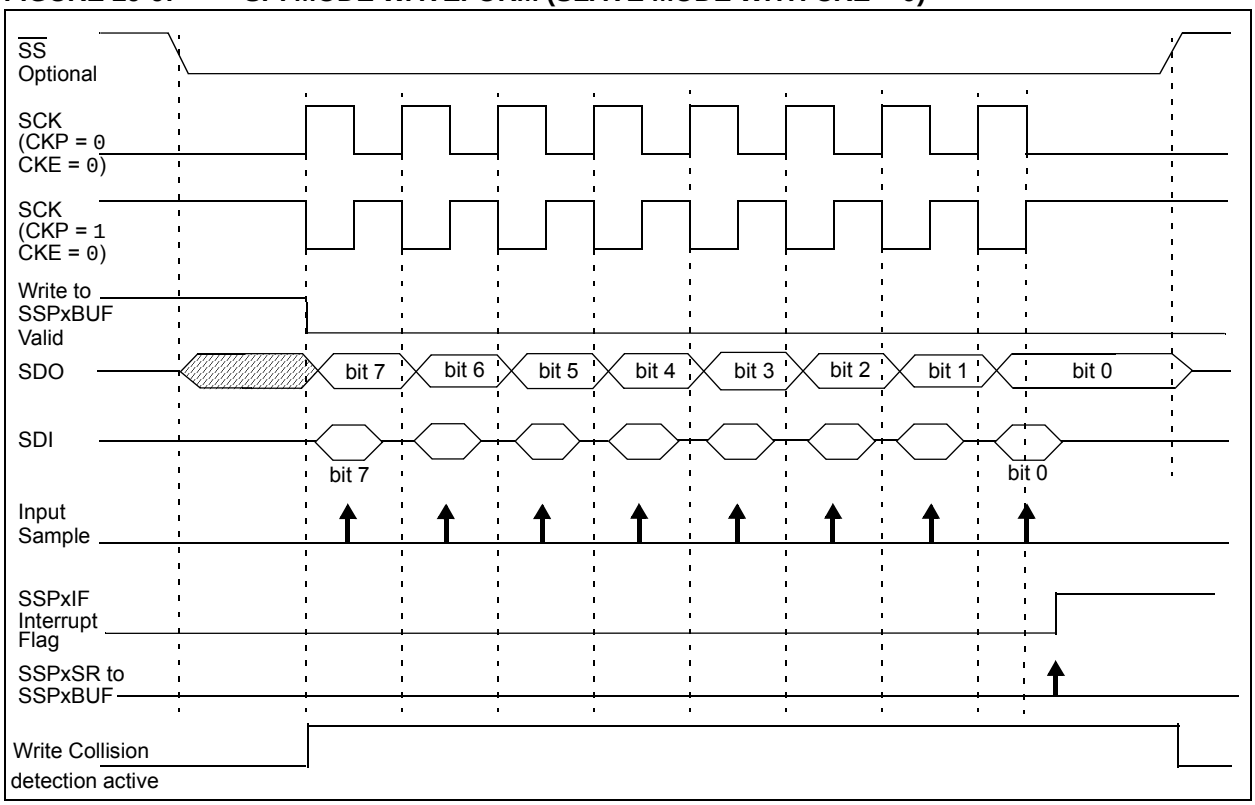
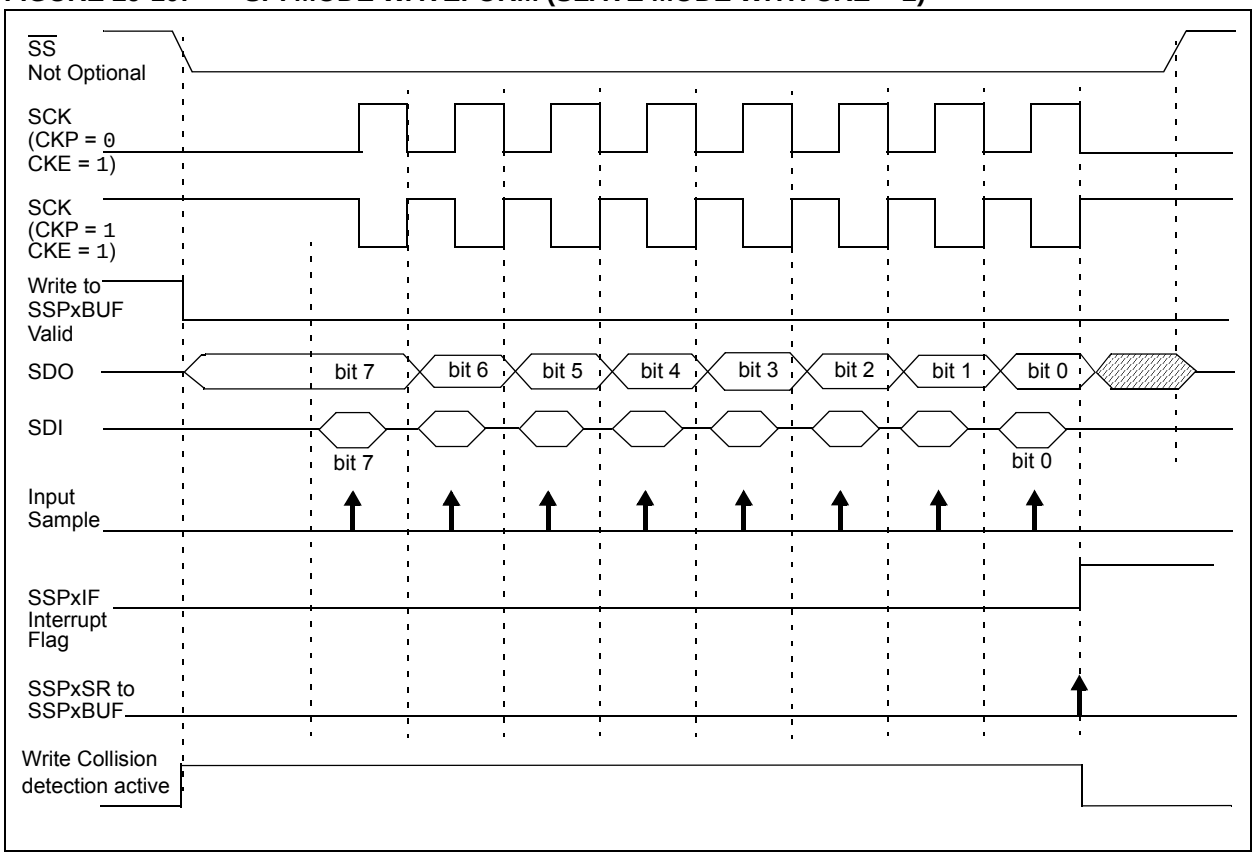


FIGURE 29-10: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



29.2.6 SPI OPERATION IN SLEEP MODE

In SPI Master mode, module clocks may be operating at a different speed than when in Full-Power mode; in the case of the Sleep mode, all clocks are halted.

Special care must be taken by the user when the MSSP clock is much faster than the system clock.

In Slave mode, when MSSP interrupts are enabled, after the master completes sending data, an MSSP interrupt will wake the controller from Sleep.

If an exit from Sleep mode is not desired, MSSP interrupts should be disabled.

In SPI Master mode, when the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all eight bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

29.3 I²C Mode Overview

The Inter-Integrated Circuit (I^2C) bus is a multi-master serial data communication bus. Devices communicate in a master/slave environment where the master devices initiate the communication. A slave device is controlled through addressing.

The I^2C bus specifies two signal connections:

- Serial Clock (SCL)
- Serial Data (SDA)

Figure 29-11 shows the block diagram of the MSSPx module when operating in I^2C mode.

Both the SCL and SDA connections are bidirectional open-drain lines, each requiring pull-up resistors for the supply voltage. Pulling the line to ground is considered a logical zero and letting the line float is considered a logical one.

Figure 29-11 shows a typical connection between two processors configured as master and slave devices.

The I^2C bus can operate with one or more master devices and one or more slave devices.

There are four potential modes of operation for a given device:

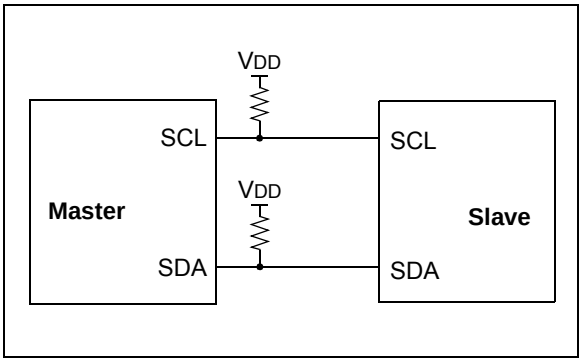
- Master Transmit mode
  (master is transmitting data to a slave)
- Master Receive mode
  (master is receiving data from a slave)
- Slave Transmit mode
  (slave is transmitting data to a master)
- Slave Receive mode
  (slave is receiving data from the master)

To begin communication, a master device starts out in Master Transmit mode. The master device sends out a Start bit followed by the address byte of the slave it intends to communicate with. This is followed by a single Read/Write bit, which determines whether the master intends to transmit to or receive data from the slave device.

If the requested slave exists on the bus, it will respond with an Acknowledge bit, otherwise known as an $\overline{ACK}$. The master then continues in either Transmit mode or Receive mode and the slave continues in the complement, either in Receive mode or Transmit mode, respectively.

A Start bit is indicated by a high-to-low transition of the SDA line while the SCL line is held high. Address and data bytes are sent out, Most Significant bit (MSb) first. The Read/Write bit is sent out as a logical one when the master intends to read data from the slave, and is sent out as a logical zero when it intends to write data to the slave.

FIGURE 29-11: I²C™ MASTER/ SLAVE CONNECTION



The Acknowledge bit ($\overline{ACK}$) is an active-low signal, which holds the SDA line low to indicate to the transmitter that the slave device has received the transmitted data and is ready to receive more.

The transition of a data bit is always performed while the SCL line is held low. Transitions that occur while the SCL line is held high are used to indicate Start and Stop bits.

If the master intends to write to the slave, then it repeatedly sends out a byte of data, with the slave responding after each byte with an $\overline{ACK}$ bit. In this example, the master device is in Master Transmit mode and the slave is in Slave Receive mode.

If the master intends to read from the slave, then it repeatedly receives a byte of data from the slave, and responds after each byte with an $\overline{ACK}$ bit. In this example, the master device is in Master Receive mode and the slave is Slave Transmit mode.

On the last byte of data communicated, the master device may end the transmission by sending a Stop bit. If the master device is in Receive mode, it sends the Stop bit in place of the last $\overline{ACK}$ bit. A Stop bit is indicated by a low-to-high transition of the SDA line while the SCL line is held high.

In some cases, the master may want to maintain control of the bus and re-initiate another transmission. If so, the master device may send another Start bit in place of the Stop bit or last $\overline{ACK}$ bit when it is in Receive mode.

The I^2C bus specifies three message protocols;

- Single message where a master writes data to a slave.
- Single message where a master reads data from a slave.
- Combined message where a master initiates a minimum of two writes, or two reads, or a combination of writes and reads, to one or more slaves.

When one device is transmitting a logical one, or letting the line float, and a second device is transmitting a logical zero, or holding the line low, the first device can detect that the line is not a logical one. This detection, when used on the SCL line, is called clock stretching. Clock stretching gives slave devices a mechanism to control the flow of data. When this detection is used on the SDA line, it is called arbitration. Arbitration ensures that there is only one master device communicating at any single time.

29.3.1 CLOCK STRETCHING

When a slave device has not completed processing data, it can delay the transfer of more data through the process of clock stretching. An addressed slave device may hold the SCL clock line low after receiving or sending a bit, indicating that it is not yet ready to continue. The master that is communicating with the slave will attempt to raise the SCL line in order to transfer the next bit, but will detect that the clock line has not yet been released. Because the SCL connection is open-drain, the slave has the ability to hold that line low until it is ready to continue communicating.

Clock stretching allows receivers that cannot keep up with a transmitter to control the flow of incoming data.

29.3.2 ARBITRATION

Each master device must monitor the bus for Start and Stop bits. If the device detects that the bus is busy, it cannot begin a new message until the bus returns to an Idle state.

However, two master devices may try to initiate a transmission on or about the same time. When this occurs, the process of arbitration begins. Each transmitter checks the level of the SDA data line and compares it to the level that it expects to find. The first transmitter to observe that the two levels do not match, loses arbitration, and must stop transmitting on the SDA line.

For example, if one transmitter holds the SDA line to a logical one (lets it float) and a second transmitter holds it to a logical zero (pulls it low), the result is that the SDA line will be low. The first transmitter then observes that the level of the line is different than expected and concludes that another transmitter is communicating.

The first transmitter to notice this difference is the one that loses arbitration and must stop driving the SDA line. If this transmitter is also a master device, it also must stop driving the SCL line. It then can monitor the lines for a Stop condition before trying to reissue its transmission. In the meantime, the other device that has not noticed any difference between the expected and actual levels on the SDA line continues with its original transmission. It can do so without any complications, because so far, the transmission appears exactly as expected with no other transmitter disturbing the message.

Slave Transmit mode can also be arbitrated, when a master addresses multiple slaves, but this is less common.

If two master devices are sending a message to two different slave devices at the address stage, the master sending the lower slave address always wins arbitration. When two master devices send messages to the same slave address, and addresses can sometimes refer to multiple slaves, the arbitration process must continue into the data stage.

Arbitration usually occurs very rarely, but it is a necessary process for proper multi-master support.

29.4 I²C MODE OPERATION

All MSSP I²C communication is byte oriented and shifted out MSb first. Six SFR registers and two interrupt flags interface the module with the PIC® microcontroller and user software. Two pins, SDA and SCL, are exercised by the module to communicate with other external I²C devices.

29.4.1 BYTE FORMAT

All communication in I²C is done in 9-bit segments. A byte is sent from a master to a slave or vice-versa, followed by an Acknowledge bit sent back. After the eighth falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an acknowledge value on the next clock pulse.

The clock signal, SCL, is provided by the master. Data is valid to change while the SCL signal is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define special conditions on the bus, explained below.

29.4.2 DEFINITION OF I²C TERMINOLOGY

There is language and terminology in the description of I²C communication that have definitions specific to I²C. That word usage is defined below and may be used in the rest of this document without explanation. This table was adapted from the Philips I²C specification.

29.4.3 SDA AND SCL PINS

Selection of any I²C mode with the SSPEN bit set, forces the SCL and SDA pins to be open-drain. These pins should be set by the user to inputs by setting the appropriate TRIS bits.

Note 1: Data is tied to output zero when an I²C™ mode is enabled.

2: Any device pin can be selected for SDA and SCL functions with the PPS peripheral. These functions are bidirectional. The SDA input is selected with the SSPDATPPS registers. The SCL input is selected with the SSPCLKPPS registers. Outputs are selected with the RxyPPS registers. It is the user's responsibility to make the selections so that both the input and the output for each function is on the same pin.

29.4.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT bit of the SSPxCON3 register. Hold time is the time SDA is held valid after the falling edge of SCL. Setting the SDAHT bit selects a longer 300 ns minimum hold time and may help on buses with large capacitance.

TABLE 29-1: I²C™ BUS TERMS

TERM	Description
Transmitter	The device which shifts data out onto the bus.
Receiver	The device which shifts data in from the bus.
Master	The device that initiates a transfer, generates clock signals and terminates a transfer.
Slave	The device addressed by the master.
Multi-master	A bus with more than one device that can initiate data transfers.
Arbitration	Procedure to ensure that only one master at a time controls the bus. Winning arbitration ensures that the message is not corrupted.
Synchronization	Procedure to synchronize the clocks of two or more devices on the bus.
Idle	No master is controlling the bus, and both SDA and SCL lines are high.
Active	Any time one or more master devices are controlling the bus.
Addressed Slave	Slave device that has received a matching address and is actively being clocked by a master.
Matching Address	Address byte that is clocked into a slave that matches the value stored in SSPADD.
Write Request	Slave receives a matching address with R/$\overline{W}$ bit clear, and is ready to clock in data.
Read Request	Master sends an address byte with the R/$\overline{W}$ bit set, indicating that it wishes to clock data out of the Slave. This data is the next and all following bytes until a Restart or Stop.
Clock Stretching	When a device on the bus hold SCL low to stall communication.
Bus Collision	Any time the SDA line is sampled low by the module while it is outputting and expected high state.

29.4.5 START CONDITION

The I^2C specification defines a Start condition as a transition of SDA from a high to a low state while SCL line is high. A Start condition is always generated by the master and signifies the transition of the bus from an Idle to an Active state. Figure 29-12 shows wave forms for Start and Stop conditions.

A bus collision can occur on a Start condition if the module samples the SDA line low before asserting it low. This does not conform to the I^2C Specification that states no bus collision can occur on a Start.

29.4.6 STOP CONDITION

A Stop condition is a transition of the SDA line from low-to-high state while the SCL line is high.

Note: At least one SCL low time must appear before a Stop is valid, therefore, if the SDA line goes low then high again while the SCL line stays high, only the Start condition is detected.

29.4.7 RESTART CONDITION

A Restart is valid any time that a Stop would be valid. A master can issue a Restart if it wishes to hold the bus after terminating the current transfer. A Restart has the same effect on the slave that a Start would, resetting all slave logic and preparing it to clock in an address. The master may want to address the same or another slave. Figure 29-13 shows the wave form for a Restart condition.

In 10-bit Addressing Slave mode a Restart is required for the master to clock data out of the addressed slave. Once a slave has been fully addressed, matching both high and low address bytes, the master can issue a Restart and the high address byte with the R/$\overline{W}$ bit set. The slave logic will then hold the clock and prepare to clock out data.

After a full match with R/$\overline{W}$ clear in 10-bit mode, a prior match flag is set and maintained until a Stop condition, a high address with R/$\overline{W}$ clear, or high address match fails.

29.4.8 START/STOP CONDITION INTERRUPT MASKING

The SCIE and PCIE bits of the SSPxCON3 register can enable the generation of an interrupt in Slave modes that do not typically support this function. Slave modes where interrupt on Start and Stop detect are already enabled, these bits will have no effect.

FIGURE 29-12: I^2C™ START AND STOP CONDITIONS

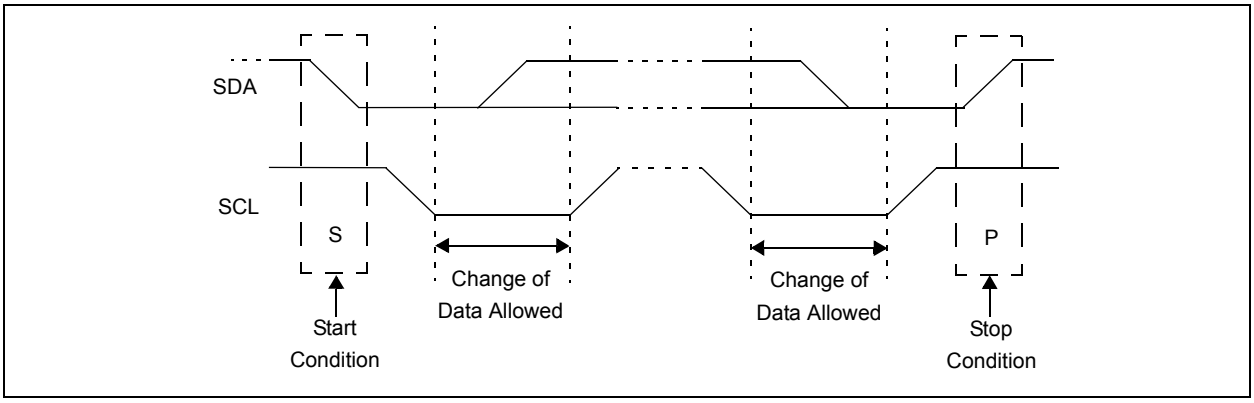
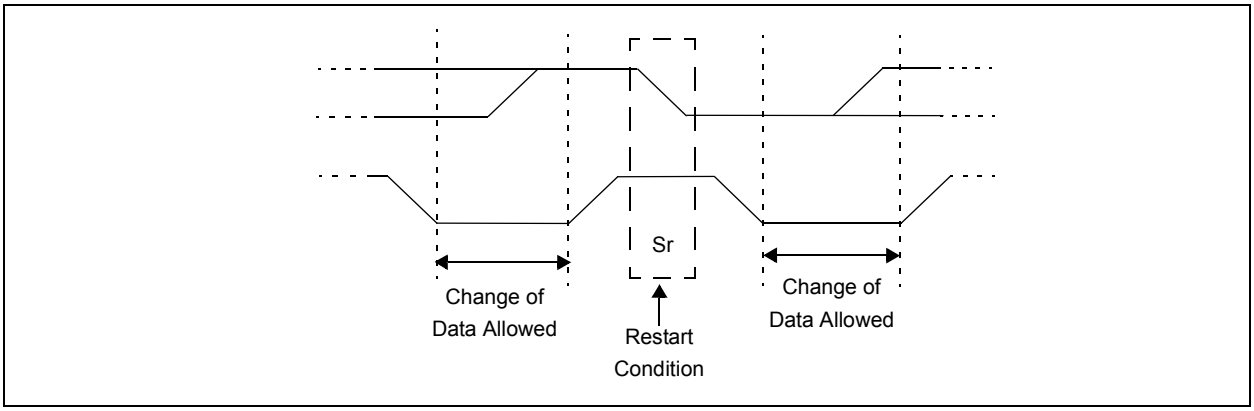


FIGURE 29-13: I^2C™ RESTART CONDITION



29.4.9 ACKNOWLEDGE SEQUENCE

The ninth SCL pulse for any transferred byte in I^2C is dedicated as an Acknowledge. It allows receiving devices to respond back to the transmitter by pulling the SDA line low. The transmitter must release control of the line during this time to shift in the response. The Acknowledge ($\overline{ACK}$) is an active-low signal, pulling the SDA line low indicates to the transmitter that the device has received the transmitted data and is ready to receive more.

The result of an $\overline{ACK}$ is placed in the ACKSTAT bit of the SSPxCON2 register.

Slave software, when the AHEN and DHEN bits are set, allow the user to set the $\overline{ACK}$ value sent back to the transmitter. The ACKDT bit of the SSPxCON2 register is set/cleared to determine the response.

Slave hardware will generate an $\overline{ACK}$ response if the AHEN and DHEN bits of the SSPxCON3 register are clear.

There are certain conditions where an $\overline{ACK}$ will not be sent by the slave. If the BF bit of the SSPxSTAT register or the SSPOV bit of the SSPxCON1 register are set when a byte is received.

When the module is addressed, after the eighth falling edge of SCL on the bus, the ACKTIM bit of the SSPxCON3 register is set. The ACKTIM bit indicates the acknowledge time of the active bus. The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is enabled.

29.5 I^2C SLAVE MODE OPERATION

The MSSP Slave mode operates in one of four modes selected by the SSPM bits of SSPxCON1 register. The modes can be divided into 7-bit and 10-bit Addressing mode. 10-bit Addressing modes operate the same as 7-bit with some additional overhead for handling the larger addresses.

Modes with Start and Stop bit interrupts operate the same as the other modes with SSPxIF additionally getting set upon detection of a Start, Restart, or Stop condition.

29.5.1 SLAVE MODE ADDRESSES

The SSPxADD register (Register 29-6) contains the Slave mode address. The first byte received after a Start or Restart condition is compared against the value stored in this register. If the byte matches, the value is loaded into the SSPxBUF register and an interrupt is generated. If the value does not match, the module goes idle and no indication is given to the software that anything happened.

The SSP Mask register (Register 29-5) affects the address matching process. See **Section 29.5.9 "SSP Mask Register"** for more information.

29.5.1.1 I^2C Slave 7-bit Addressing Mode

In 7-bit Addressing mode, the LSb of the received data byte is ignored when determining if there is an address match.

29.5.1.2 I^2C Slave 10-bit Addressing Mode

In 10-bit Addressing mode, the first received byte is compared to the binary value of '1 1 1 1 0 A9 A8 0'. A9 and A8 are the two MSb's of the 10-bit address and stored in bits 2 and 1 of the SSPxADD register.

After the acknowledge of the high byte the UA bit is set and SCL is held low until the user updates SSPxADD with the low address. The low address byte is clocked in and all eight bits are compared to the low address value in SSPxADD. Even if there is not an address match; SSPIF and UA are set, and SCL is held low until SSPxADD is updated to receive a high byte again. When SSPxADD is updated the UA bit is cleared. This ensures the module is ready to receive the high address byte on the next communication.

A high and low address match as a write request is required at the start of all 10-bit addressing communication. A transmission can be initiated by issuing a Restart once the slave is addressed, and clocking in the high address with the R/$\overline{W}$ bit set. The slave hardware will then acknowledge the read request and prepare to clock out data. This is only valid for a slave after it has received a complete high and low address byte match.

29.5.2 SLAVE RECEPTION

When the R/$\overline{W}$ bit of a matching received address byte is clear, the R/$\overline{W}$ bit of the SSPxSTAT register is cleared. The received address is loaded into the SSPxBUF register and acknowledged.

When the Overflow condition exists for a received address, then not Acknowledge is given. An Overflow condition is defined as either bit BF of the SSPxSTAT register is set, or bit SSPOV of the SSPxCON1 register is set. The BOEN bit of the SSPxCON3 register modifies this operation. For more information see Register 29-4.

An MSSP interrupt is generated for each transferred data byte. Flag bit, SSPxIF, must be cleared by software.

When the SEN bit of the SSPxCON2 register is set, SCL will be held low (clock stretch) following each received byte. The clock must be released by setting the CKP bit of the SSPxCON1 register, except sometimes in 10-bit mode. See **Section 29.5.6.2 "10-bit Addressing Mode"** for more detail.

29.5.2.1 7-bit Addressing Reception

This section describes a standard sequence of events for the MSSPx module configured as an I^2C slave in 7-bit Addressing mode. Figure 29-14 and Figure 29-15 is used as a visual reference for this description.

This is a step-by-step process of what typically must be done to accomplish I^2C communication.

1. Start bit detected.
2. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Matching address with R/$\overline{W}$ bit clear is received.
4. The slave pulls SDA low sending an $\overline{ACK}$ to the master, and sets SSPxIF bit.
5. Software clears the SSPxIF bit.
6. Software reads received address from SSPxBUF clearing the BF flag.
7. If SEN = 1; Slave software sets CKP bit to release the SCL line.
8. The master clocks out a data byte.
9. Slave drives SDA low sending an $\overline{ACK}$ to the master, and sets SSPxIF bit.
10. Software clears SSPxIF.
11. Software reads the received byte from SSPxBUF clearing BF.
12. Steps 8-12 are repeated for all received bytes from the master.
13. Master sends Stop condition, setting P bit of SSPxSTAT, and the bus goes idle.

29.5.2.2 7-bit Reception with AHEN and DHEN

Slave device reception with AHEN and DHEN set operate the same as without these options with extra interrupts and clock stretching added after the eighth falling edge of SCL. These additional interrupts allow the slave software to decide whether it wants to $\overline{ACK}$ the receive address or data byte, rather than the hardware. This functionality adds support for PMBus™ that was not present on previous versions of this module.

This list describes the steps that need to be taken by slave software to use these options for I^2C communication. Figure 29-16 displays a module using both address and data holding. Figure 29-17 includes the operation with the SEN bit of the SSPxCON2 register set.

1. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
2. Matching address with R/$\overline{W}$ bit clear is clocked in. SSPxIF is set and CKP cleared after the eighth falling edge of SCL.
3. Slave clears the SSPxIF.
4. Slave can look at the ACKTIM bit of the SSPxCON3 register to determine if the SSPxIF was after or before the $\overline{ACK}$.
5. Slave reads the address value from SSPxBUF, clearing the BF flag.
6. Slave sets $\overline{ACK}$ value clocked out to the master by setting ACKDT.
7. Slave releases the clock by setting CKP.
8. SSPxIF is set after an $\overline{ACK}$, not after a NACK.
9. If SEN = 1 the slave hardware will stretch the clock after the $\overline{ACK}$.
10. Slave clears SSPxIF.

Note: SSPxIF is still set after the ninth falling edge of SCL even if there is no clock stretching and BF has been cleared. Only if NACK is sent to master is SSPIF not set

11. SSPxIF set and CKP cleared after eighth falling edge of SCL for a received data byte.
12. Slave looks at ACKTIM bit of SSPxCON3 to determine the source of the interrupt.
13. Slave reads the received data from SSPxBUF clearing BF.
14. Steps 7-14 are the same for each received data byte.
15. Communication is ended by either the slave sending an $\overline{ACK}$ = 1, or the master sending a Stop condition. If a Stop is sent and Interrupt on Stop Detect is disabled, the slave will only know by polling the P bit of the SSPxSTAT register.

FIGURE 29-14: I²C™ SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 0, DHEN = 0)

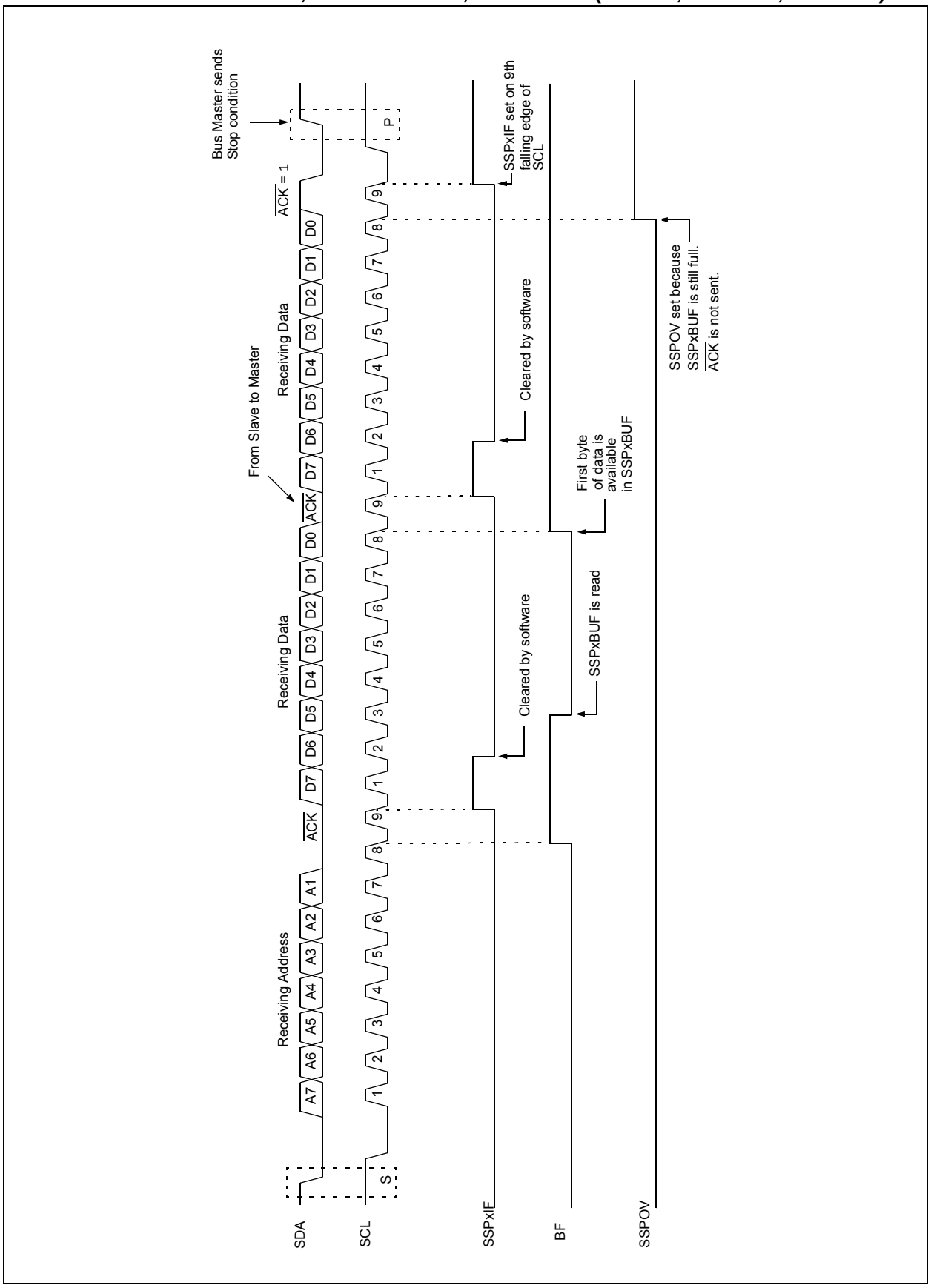


FIGURE 29-15: I²C™ SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)

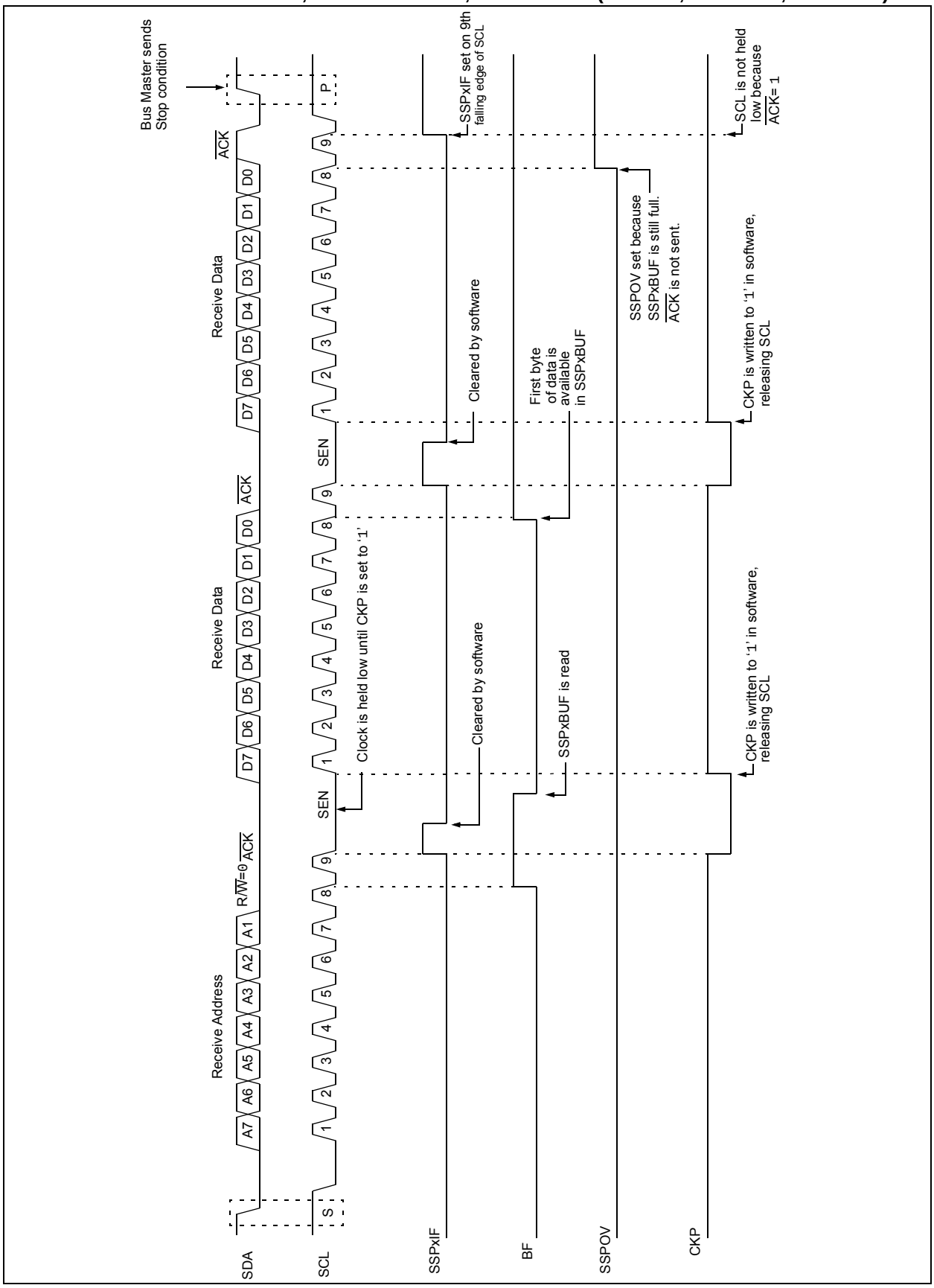


FIGURE 29-16: I²C™ SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 1)

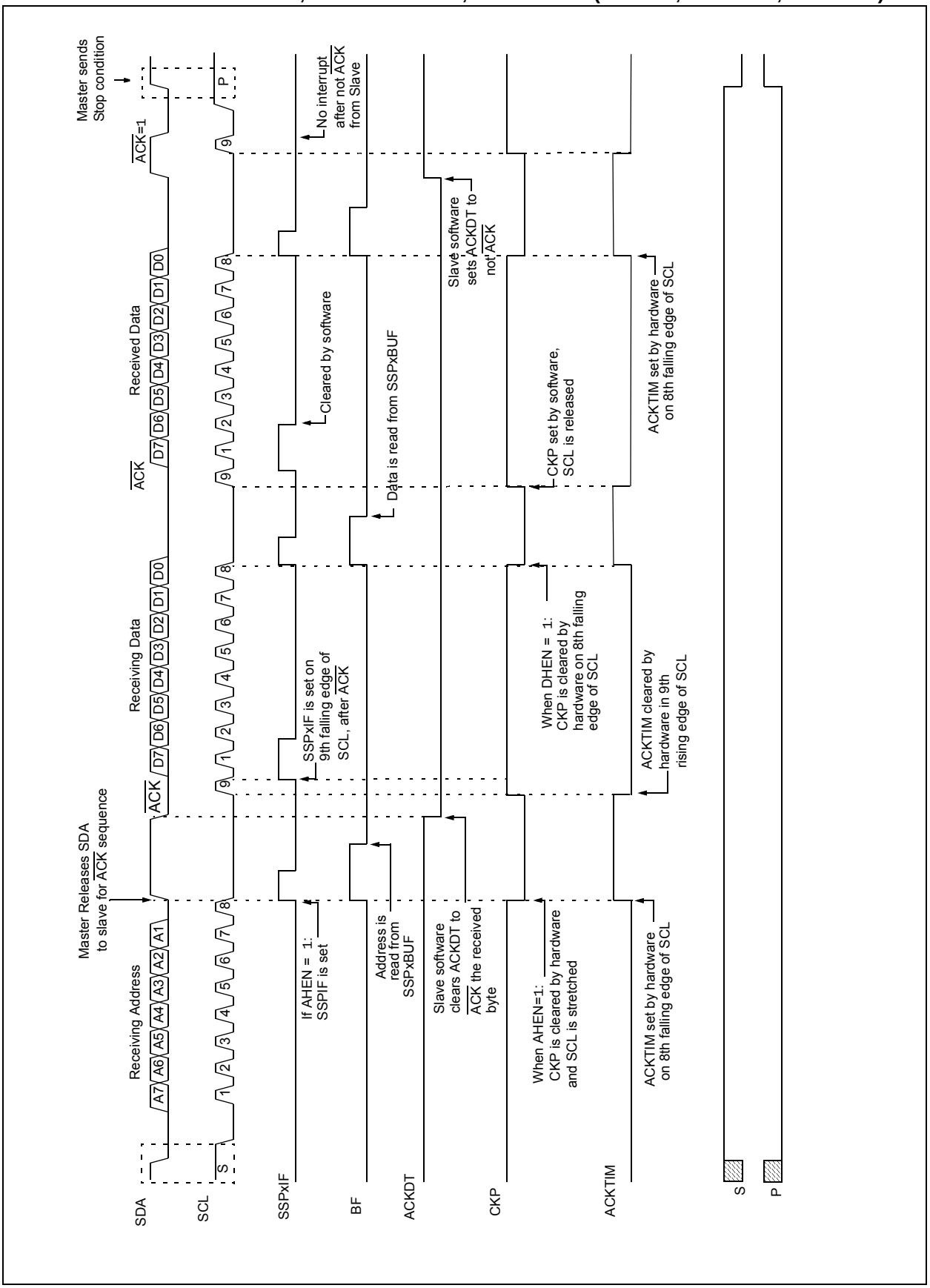
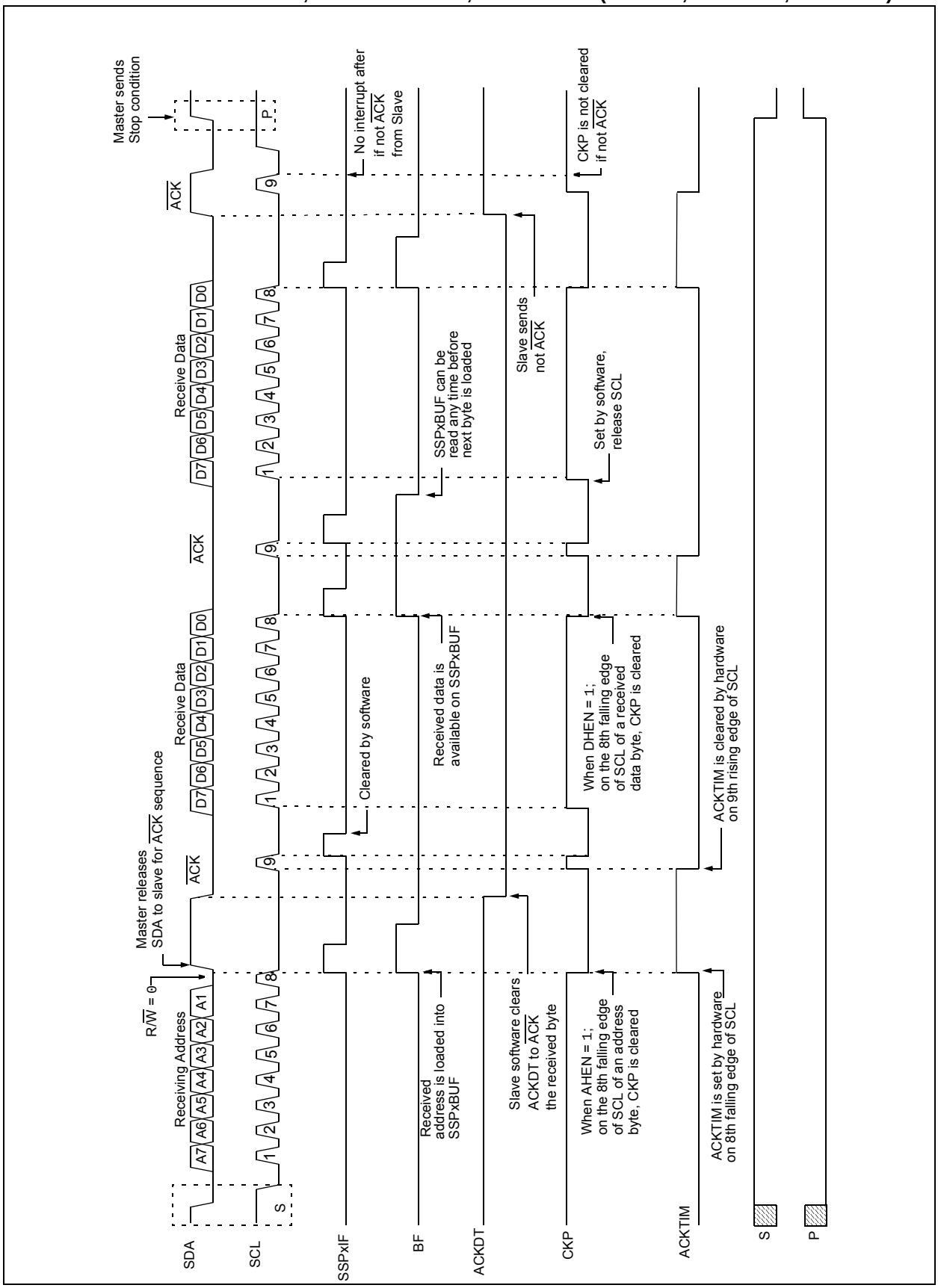


FIGURE 29-17: I²C™ SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 1, DHEN = 1)



29.5.3 SLAVE TRANSMISSION

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPxSTAT register is set. The received address is loaded into the SSPxBUF register, and an $\overline{ACK}$ pulse is sent by the slave on the ninth bit.

Following the $\overline{ACK}$, slave hardware clears the CKP bit and the SCL pin is held low (see **Section 29.5.6 "Clock Stretching"** for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data.

The transmit data must be loaded into the SSPxBUF register which also loads the SSPxSR register. Then the SCL pin should be released by setting the CKP bit of the SSPxCON1 register. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time.

The $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. This $\overline{ACK}$ value is copied to the ACKSTAT bit of the SSPxCON2 register. If ACKSTAT is set (not $\overline{ACK}$), then the data transfer is complete. In this case, when the not $\overline{ACK}$ is latched by the slave, the slave goes idle and waits for another occurrence of the Start bit. If the SDA line was low ($\overline{ACK}$), the next transmit data must be loaded into the SSPxBUF register. Again, the SCL pin must be released by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPxIF bit must be cleared by software and the SSPxSTAT register is used to determine the status of the byte. The SSPxIF bit is set on the falling edge of the ninth clock pulse.

29.5.3.1 Slave Mode Bus Collision

A slave receives a Read request and begins shifting data out on the SDA line. If a bus collision is detected and the SBCDE bit of the SSPxCON3 register is set, the BCLIF bit of the PIR register is set. Once a bus collision is detected, the slave goes idle and waits to be addressed again. User software can use the BCLIF bit to handle a slave bus collision.

29.5.3.2 7-bit Transmission

A master device can transmit a read request to a slave, and then clock data out of the slave. The list below outlines what software for a slave will need to do to accomplish a standard transmission. Figure 29-18 can be used as a reference to this list.

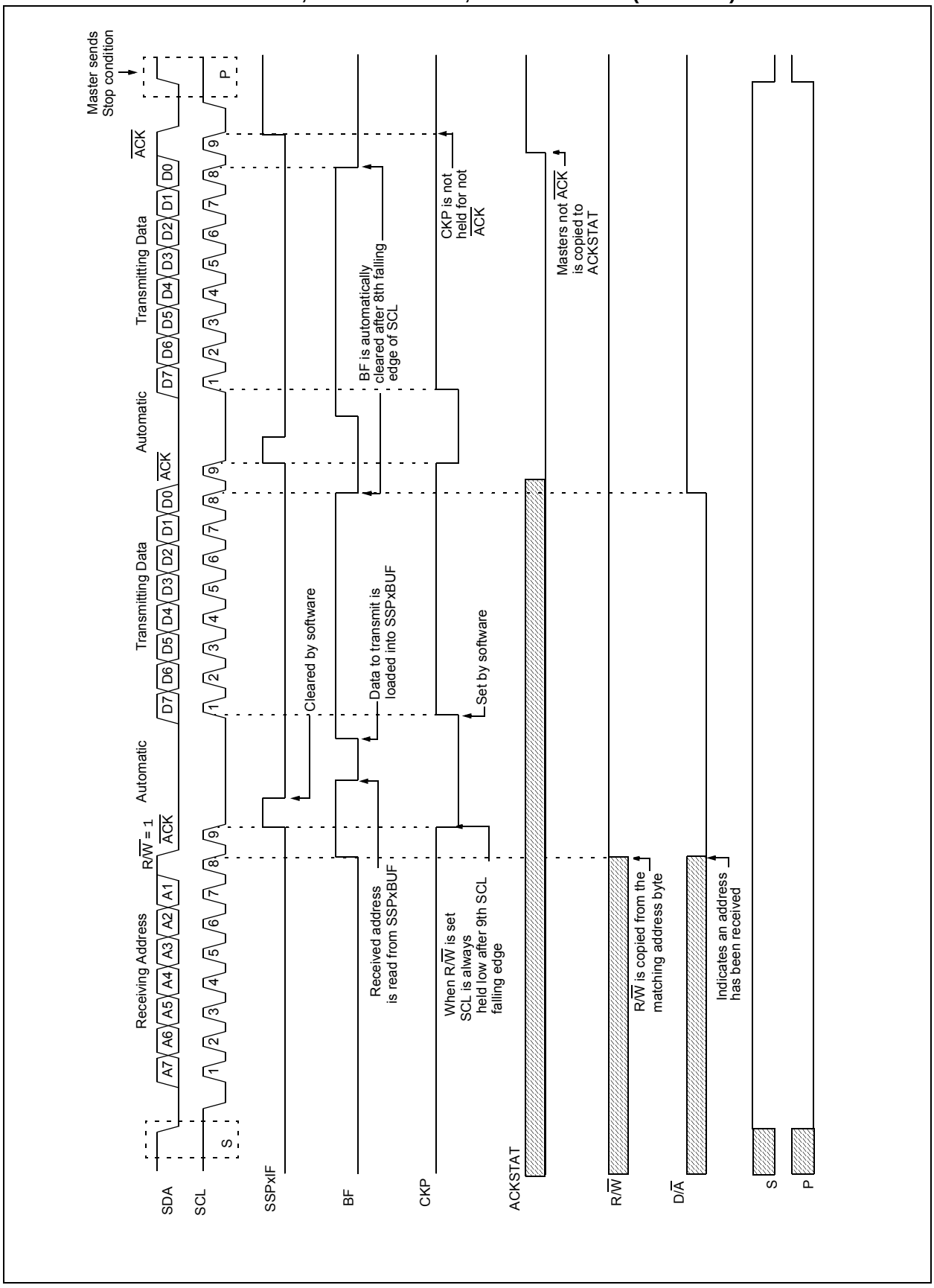
1. Master sends a Start condition on SDA and SCL.
2. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Matching address with R/$\overline{W}$ bit set is received by the Slave setting SSPxIF bit.
4. Slave hardware generates an $\overline{ACK}$ and sets SSPxIF.
5. SSPxIF bit is cleared by user.
6. Software reads the received address from SSPxBUF, clearing BF.
7. R/$\overline{W}$ is set so CKP was automatically cleared after the $\overline{ACK}$.
8. The slave software loads the transmit data into SSPxBUF.
9. CKP bit is set releasing SCL, allowing the master to clock the data out of the slave.
10. SSPxIF is set after the $\overline{ACK}$ response from the master is loaded into the ACKSTAT register.
11. SSPxIF bit is cleared.
12. The slave software checks the ACKSTAT bit to see if the master wants to clock out more data.

Note 1: If the master $\overline{ACK}$s the clock will be stretched.

2: ACKSTAT is the only bit updated on the rising edge of SCL (9th) rather than the falling.

13. Steps 9-13 are repeated for each transmitted byte.
14. If the master sends a not $\overline{ACK}$; the clock is not held, but SSPxIF is still set.
15. The master sends a Restart condition or a Stop.
16. The slave is no longer addressed.

FIGURE 29-18: I²C™ SLAVE, 7-BIT ADDRESS, TRANSMISSION (AHEN = 0)



29.5.3.3 7-bit Transmission with Address Hold Enabled

Setting the AHEN bit of the SSPxCON3 register enables additional clock stretching and interrupt generation after the eighth falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSPxIF interrupt is set.

Figure 29-19 displays a standard waveform of a 7-bit address slave transmission with AHEN enabled.

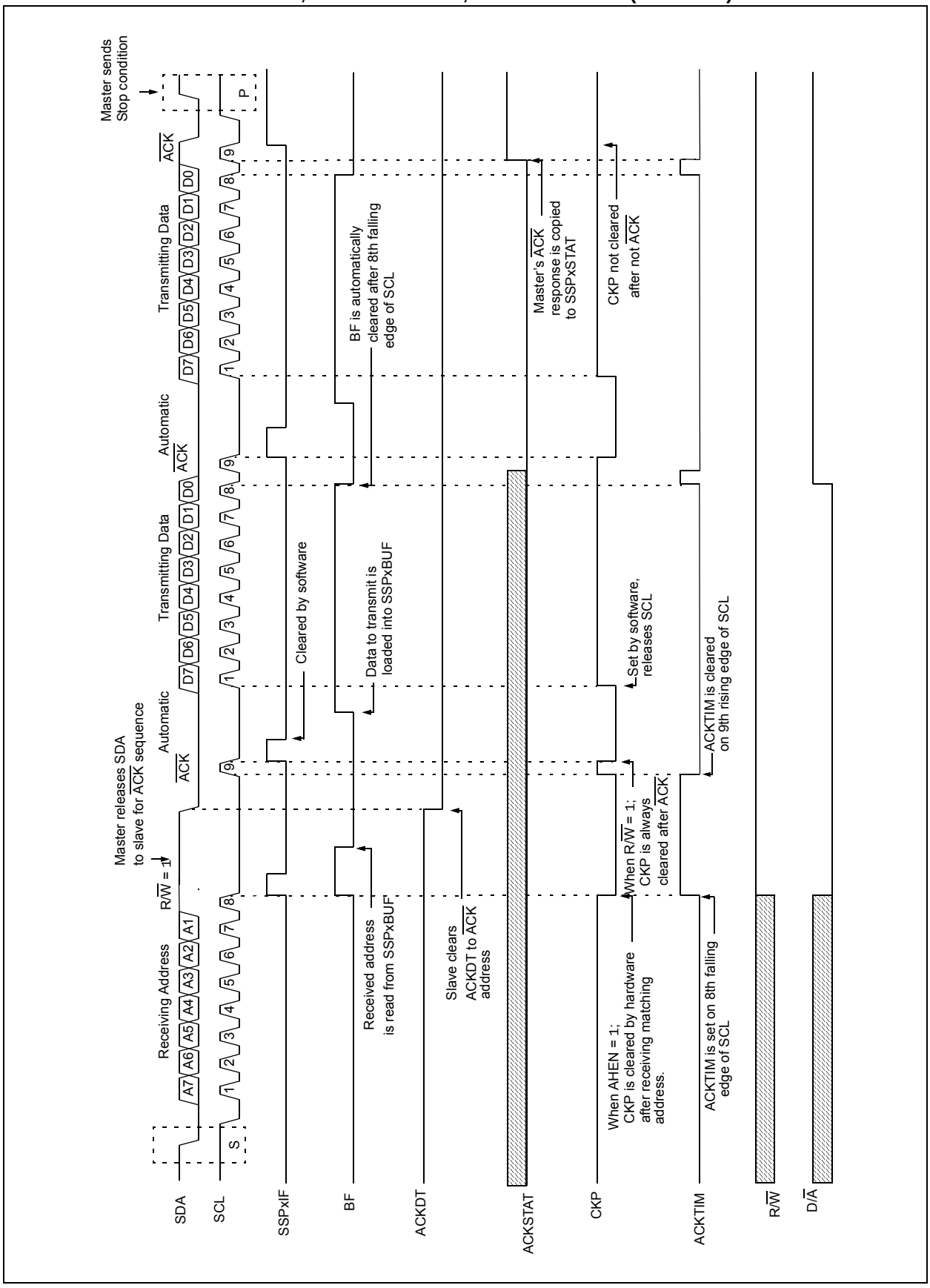
1. Bus starts Idle.
2. Master sends Start condition; the S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Master sends matching address with R/$\overline{W}$ bit set. After the eighth falling edge of the SCL line the CKP bit is cleared and SSPxIF interrupt is generated.
4. Slave software clears SSPxIF.
5. Slave software reads ACKTIM bit of SSPxCON3 register, and R/$\overline{W}$ and D/$\overline{A}$ of the SSPxSTAT register to determine the source of the interrupt.
6. Slave reads the address value from the SSPxBUF register clearing the BF bit.
7. Slave software decides from this information if it wishes to $\overline{ACK}$ or not $\overline{ACK}$ and sets the ACKDT bit of the SSPxCON2 register accordingly.
8. Slave sets the CKP bit releasing SCL.
9. Master clocks in the $\overline{ACK}$ value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSPxIF after the $\overline{ACK}$ if the R/$\overline{W}$ bit is set.
11. Slave software clears SSPxIF.
12. Slave loads value to transmit to the master into SSPxBUF setting the BF bit.

Note: SSPxBUF cannot be loaded until after the $\overline{ACK}$.

13. Slave sets the CKP bit releasing the clock.
14. Master clocks out the data from the slave and sends an $\overline{ACK}$ value on the ninth SCL pulse.
15. Slave hardware copies the $\overline{ACK}$ value into the ACKSTAT bit of the SSPxCON2 register.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not $\overline{ACK}$ the slave releases the bus allowing the master to send a Stop and end the communication.

Note: Master must send a not $\overline{ACK}$ on the last byte to ensure that the slave releases the SCL line to receive a Stop.

FIGURE 29-19: I²C™ SLAVE, 7-BIT ADDRESS, TRANSMISSION (AHEN = 1)



29.5.4 SLAVE MODE 10-BIT ADDRESS RECEPTION

This section describes a standard sequence of events for the MSSPx module configured as an I^2C slave in 10-bit Addressing mode.

Figure 29-20 is used as a visual reference for this description.

This is a step-by-step process of what must be done by slave software to accomplish I^2C communication.

1. Bus starts Idle.
2. Master sends Start condition; S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Master sends matching high address with R/$\overline{W}$ bit clear; UA bit of the SSPxSTAT register is set.
4. Slave sends $\overline{ACK}$ and SSPxIF is set.
5. Software clears the SSPxIF bit.
6. Software reads received address from SSPxBUF clearing the BF flag.
7. Slave loads low address into SSPxADD, releasing SCL.
8. Master sends matching low address byte to the slave; UA bit is set.

Note: Updates to the SSPxADD register are not allowed until after the $\overline{ACK}$ sequence.

9. Slave sends $\overline{ACK}$ and SSPxIF is set.

Note: If the low address does not match, SSPxIF and UA are still set so that the slave software can set SSPxADD back to the high address. BF is not set because there is no match. CKP is unaffected.

10. Slave clears SSPxIF.
11. Slave reads the received matching address from SSPxBUF clearing BF.
12. Slave loads high address into SSPxADD.
13. Master clocks a data byte to the slave and clocks out the slaves $\overline{ACK}$ on the ninth SCL pulse; SSPxIF is set.
14. If SEN bit of SSPxCON2 is set, CKP is cleared by hardware and the clock is stretched.
15. Slave clears SSPxIF.
16. Slave reads the received byte from SSPxBUF clearing BF.
17. If SEN is set the slave sets CKP to release the SCL.
18. Steps 13-17 repeat for each received byte.
19. Master sends Stop to end the transmission.

29.5.5 10-BIT ADDRESSING WITH ADDRESS OR DATA HOLD

Reception using 10-bit addressing with AHEN or DHEN set is the same as with 7-bit modes. The only difference is the need to update the SSPxADD register using the UA bit. All functionality, specifically when the CKP bit is cleared and SCL line is held low are the same. Figure 29-21 can be used as a reference of a slave in 10-bit addressing with AHEN set.

Figure 29-22 shows a standard waveform for a slave transmitter in 10-bit Addressing mode.

FIGURE 29-20: I²C™ SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)

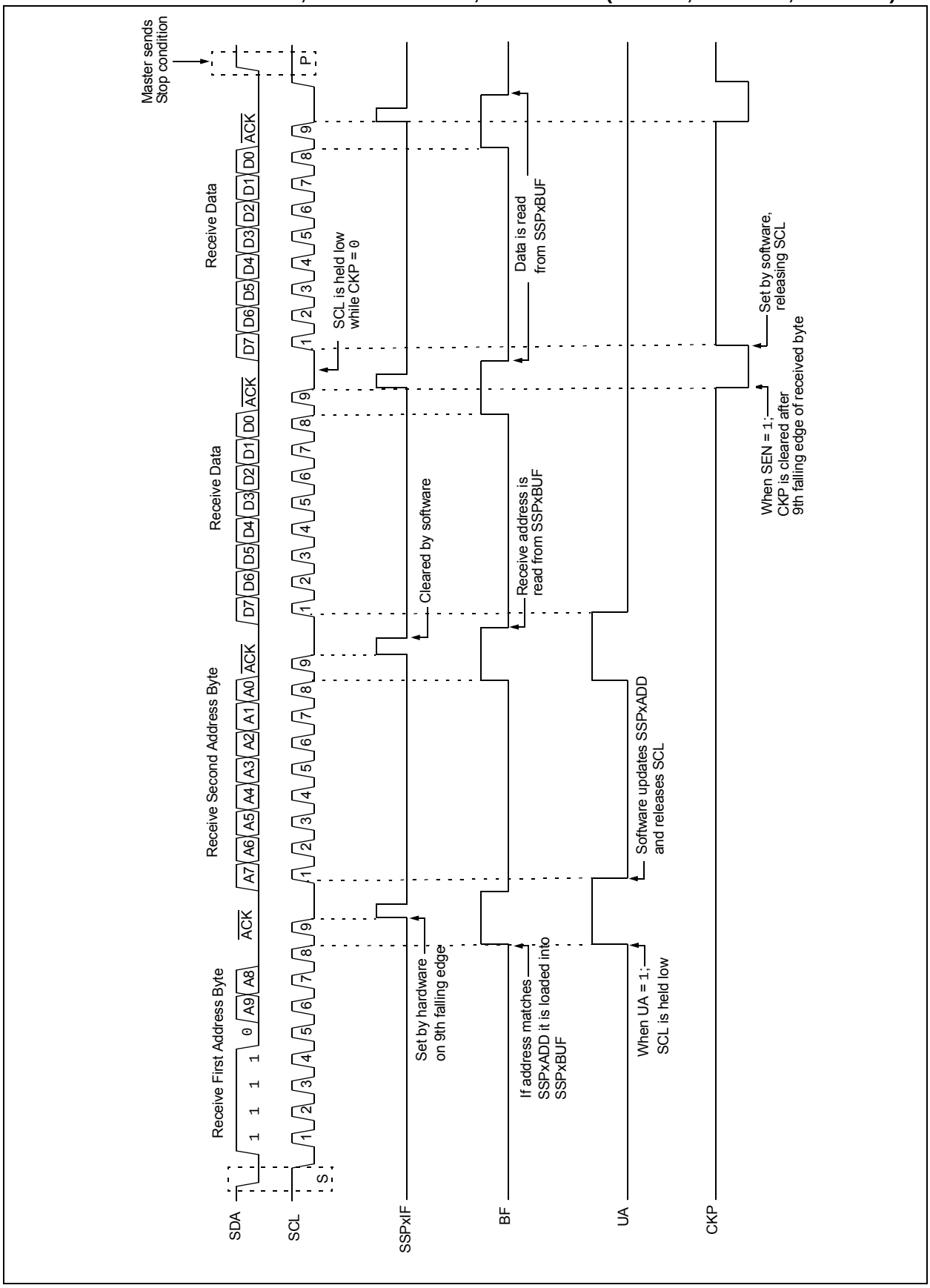


FIGURE 29-21: I²C™ SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 0)

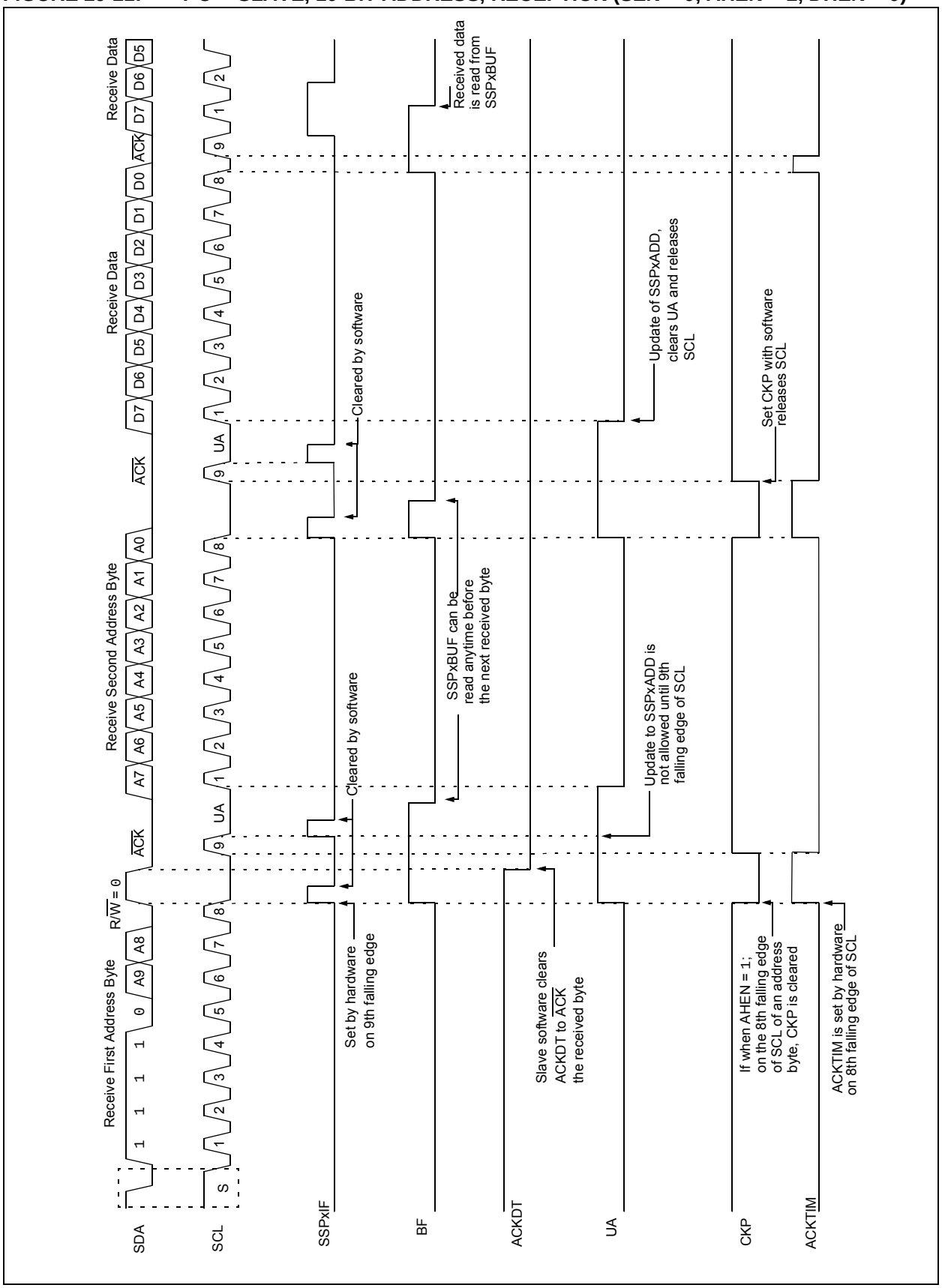
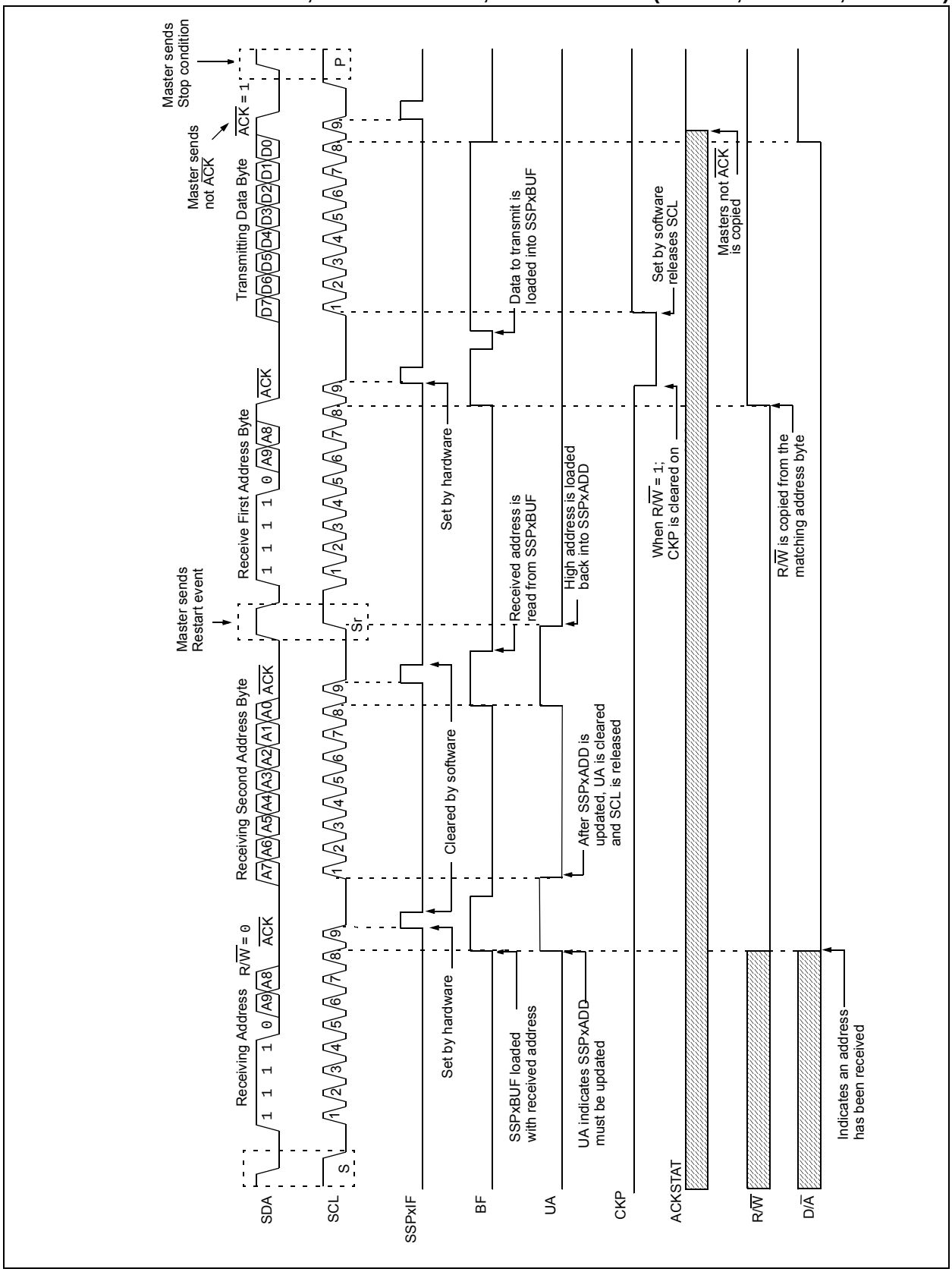


FIGURE 29-22: I²C™ SLAVE, 10-BIT ADDRESS, TRANSMISSION (SEN = 0, AHEN = 0, DHEN = 0)



29.5.6 CLOCK STRETCHING

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

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

29.5.6.1 Normal Clock Stretching

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

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

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

29.5.6.2 10-bit Addressing Mode

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

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

29.5.6.3 Byte NACKing

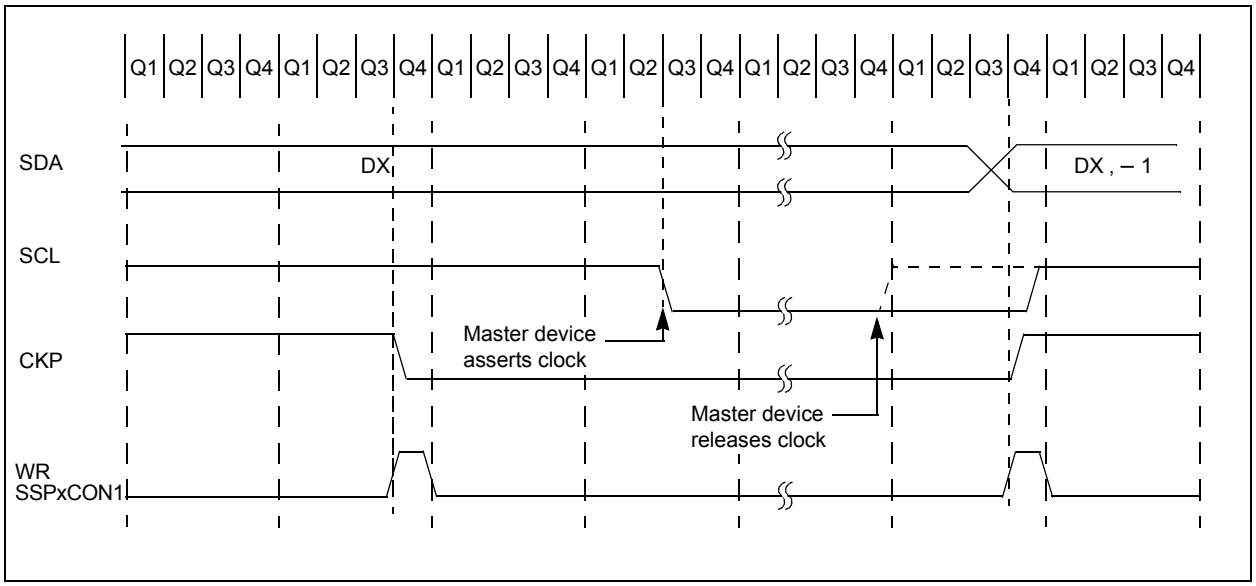
When AHEN bit of SSPxCON3 is set; CKP is cleared by hardware after the eighth falling edge of SCL for a received matching address byte. When DHEN bit of SSPxCON3 is set; CKP is cleared after the eighth falling edge of SCL for received data.

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

29.5.7 CLOCK SYNCHRONIZATION AND THE CKP BIT

Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. However, clearing the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I^2C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I^2C bus have released SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 29-23).

FIGURE 29-23: CLOCK SYNCHRONIZATION TIMING



29.5.8 GENERAL CALL ADDRESS SUPPORT

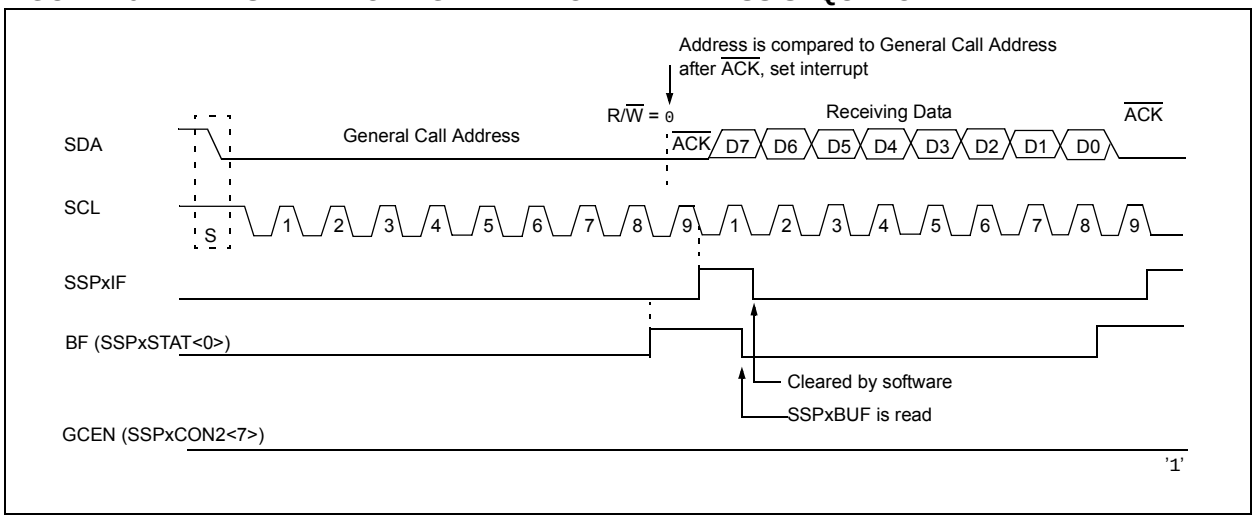
The addressing procedure for the I^2C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is a reserved address in the I^2C protocol, defined as address 0x00. When the GCEN bit of the SSPxCON2 register is set, the slave module will automatically $\overline{ACK}$ the reception of this address regardless of the value stored in SSPxADD. After the slave clocks in an address of all zeros with the R/$\overline{W}$ bit clear, an interrupt is generated and slave software can read SSPxBUF and respond. Figure 29-24 shows a general call reception sequence.

In 10-bit Address mode, the UA bit will not be set on the reception of the general call address. The slave will prepare to receive the second byte as data, just as it would in 7-bit mode.

If the AHEN bit of the SSPxCON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the eighth falling edge of SCL. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

FIGURE 29-24: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE



29.5.9 SSP MASK REGISTER

An SSP Mask (SSPMSK) register (Register 29-5) is available in I^2C Slave mode as a mask for the value held in the SSPxSR register during an address comparison operation. A zero ('0') bit in the SSPxMSK register has the effect of making the corresponding bit of the received address a "don't care".

This register is reset to all '1's upon any Reset condition and, therefore, has no effect on standard SSP operation until written with a mask value.

The SSP Mask register is active during:

- 7-bit Address mode: address compare of A<7:1>.
- 10-bit Address mode: address compare of A<7:0> only. The SSP mask has no effect during the reception of the first (high) byte of the address.

29.6 I²C Master Mode

Master mode is enabled by setting and clearing the appropriate SSPM<3:0> bits in the SSPxCON1 register and by setting the SSPEN bit. In Master mode, the SDA and SCK pins must be configured as inputs. The MSSP peripheral hardware will override the output driver TRIS controls when necessary to drive the pins low.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSPx module is disabled. Control of the I²C bus may be taken when the P bit is set, or the bus is Idle.

In Firmware Controlled Master mode, user code conducts all I²C bus operations based on Start and Stop bit condition detection. Start and Stop condition detection is the only active circuitry in this mode. All other communication is done by the user software directly manipulating the SDA and SCL lines.

The following events will cause the SSP Interrupt Flag bit, SSPxIF, to be set (SSP interrupt, if enabled):

- Start condition detected
- Stop condition detected
- Data transfer byte transmitted/received
- Acknowledge transmitted/received
- Repeated Start generated

Note 1: The MSSPx module, when configured in I²C™ Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur

2: When in Master mode, Start/Stop detection is masked and an interrupt is generated when the SEN/PEN bit is cleared and the generation is complete.

29.6.1 I²C MASTER MODE OPERATION

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/$\overline{W}$) bit. In this case, the R/$\overline{W}$ bit will be logic '0'. Serial data is transmitted eight bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

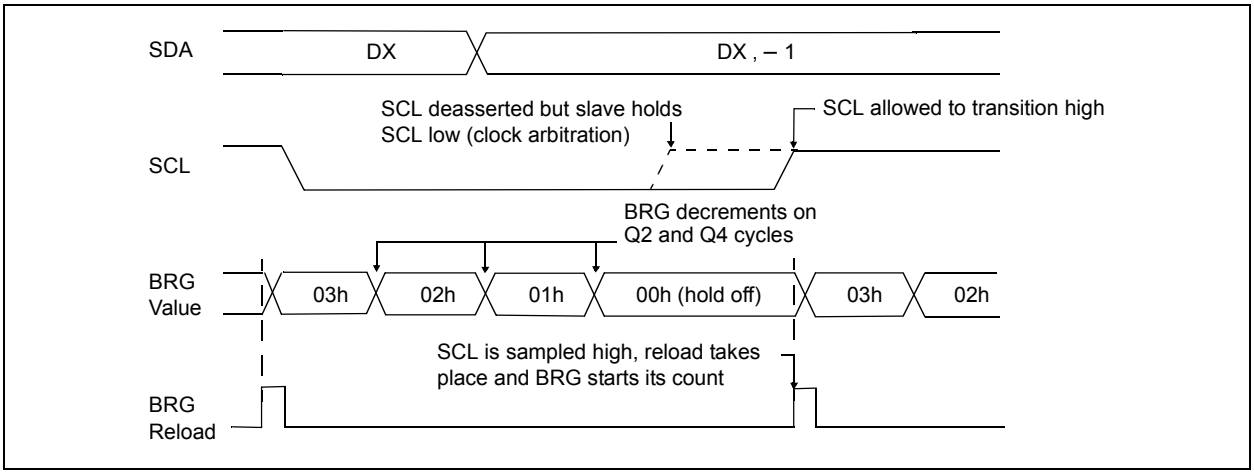
In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/$\overline{W}$ bit. In this case, the R/$\overline{W}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received eight bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

A Baud Rate Generator is used to set the clock frequency output on SCL. See **Section 29.7 "Baud Rate Generator"** for more detail.

29.6.2 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, releases the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 29-25).

FIGURE 29-25: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



29.6.3 WCOL STATUS FLAG

If the user writes the SSPxBUF when a Start, Restart, Stop, Receive or Transmit sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write does not occur). Any time the WCOL bit is set it indicates that an action on SSPxBUF was attempted while the module was not idle.

Note:	Because queuing of events is not allowed, writing to the lower five bits of SSPxCON2 is disabled until the Start condition is complete.

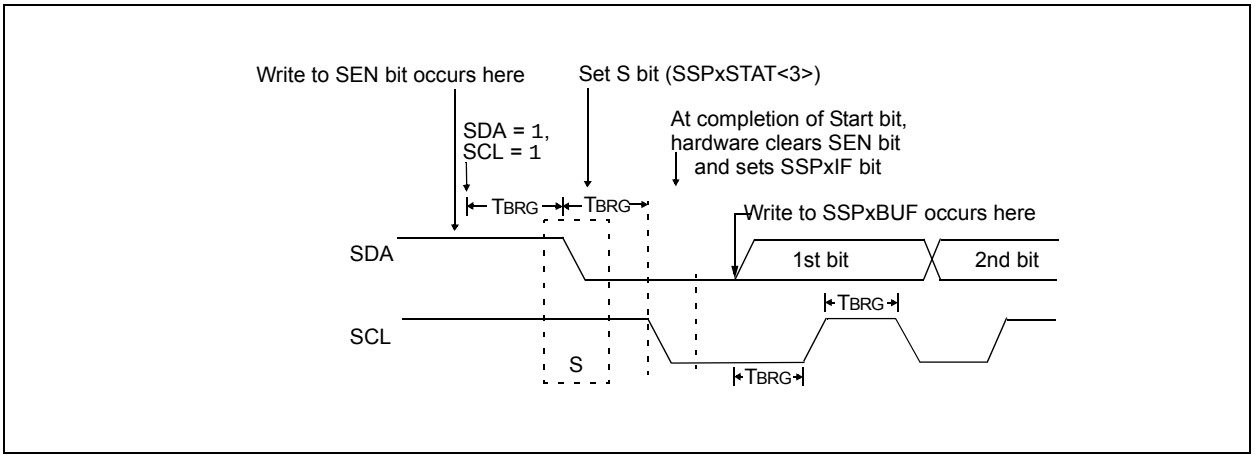
29.6.4 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition (Figure 29-26), the user sets the Start Enable bit, SEN bit of the SSPxCON2 register. If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the Start condition and causes the S bit of the SSPxSTAT register to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit of the SSPxCON2 register will be automatically cleared by hardware; the Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

Note 1: If at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLIF, is set, the Start condition is aborted and the I²C™ module is reset into its Idle state.

2: The Philips I²C specification states that a bus collision cannot occur on a Start.

FIGURE 29-26: FIRST START BIT TIMING



29.6.5 I²C MASTER MODE REPEATED START CONDITION TIMING

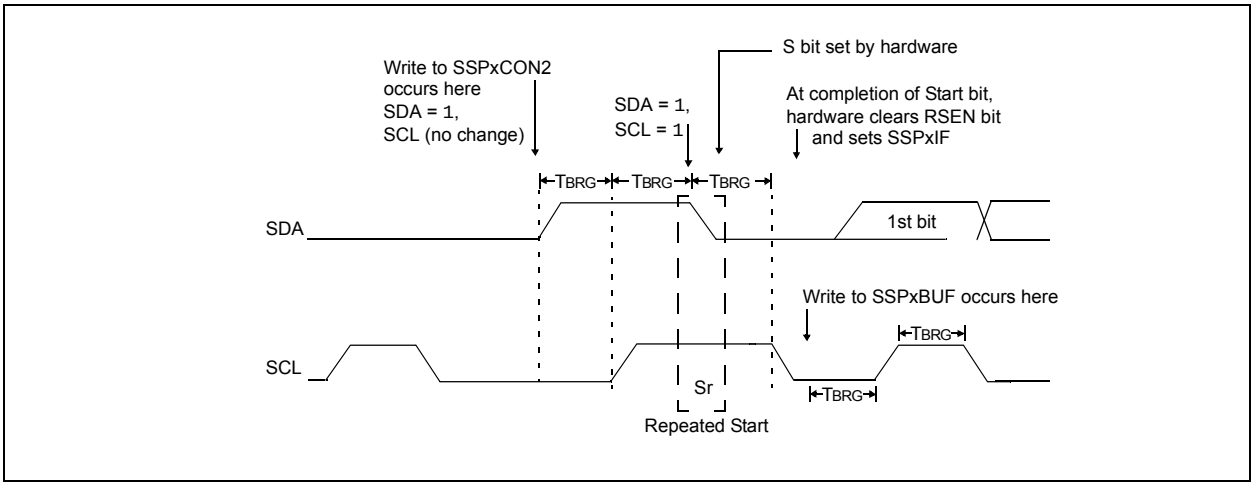
A Repeated Start condition (Figure 29-27) occurs when the RSEN bit of the SSPxCON2 register is programmed high and the master state machine is no longer active. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. SCL is asserted low. Following this, the RSEN bit of the SSPxCON2 register will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit of the SSPxSTAT register will be set. The SSPxIF bit will not be set until the Baud Rate Generator has timed out.

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

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

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

FIGURE 29-27: REPEATED START CONDITION WAVEFORM



29.6.6 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPxBUF register. This action will set the Buffer Full flag bit, BF, and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted. SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high. When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an $\overline{ACK}$ bit during the ninth bit time if an address match occurred, or if data was received properly. The status of $\overline{ACK}$ is written into the ACKSTAT bit on the rising edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPxBUF, leaving SCL low and SDA unchanged (Figure 29-28).

After the write to the SSPxBUF, each bit of the address will be shifted out on the falling edge of SCL until all seven address bits and the R/$\overline{W}$ bit are completed. On the falling edge of the eighth clock, the master will release the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the $\overline{ACK}$ bit is loaded into the ACKSTAT Status bit of the SSPxCON2 register. Following the falling edge of the ninth clock transmission of the address, the SSPxIF is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPxBUF takes place, holding SCL low and allowing SDA to float.

29.6.6.1 BF Status Flag

In Transmit mode, the BF bit of the SSPxSTAT register is set when the CPU writes to SSPxBUF and is cleared when all eight bits are shifted out.

29.6.6.2 WCOL Status Flag

If the user writes the SSPxBUF when a transmit is already in progress (i.e., SSPxSR is still shifting out a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

WCOL must be cleared by software before the next transmission.

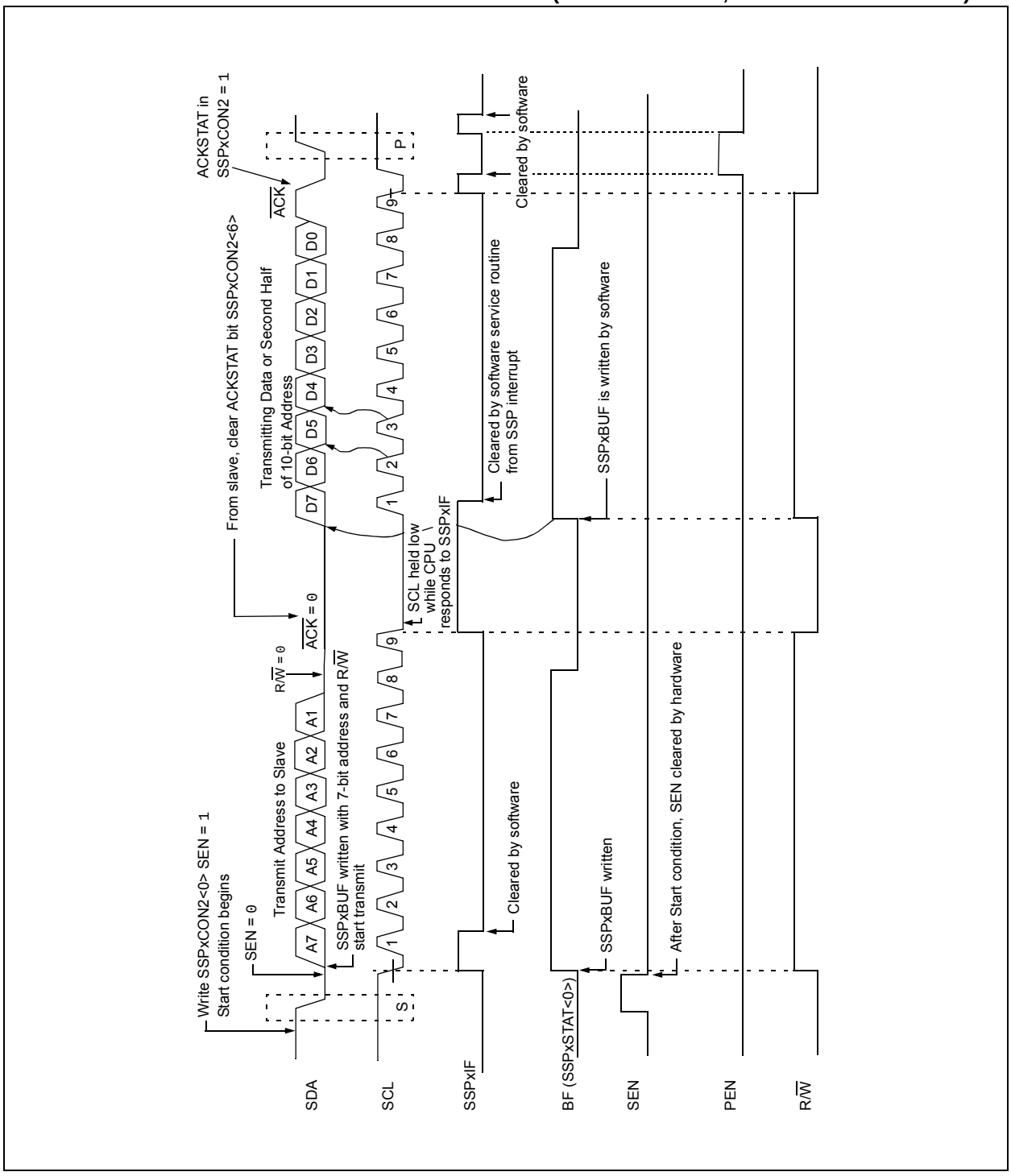
29.6.6.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit of the SSPxCON2 register is cleared when the slave has sent an Acknowledge ($\overline{ACK}$ = 0) and is set when the slave does not Acknowledge ($\overline{ACK}$ = 1). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

29.6.6.4 Typical Transmit Sequence

1. The user generates a Start condition by setting the SEN bit of the SSPxCON2 register.
2. SSPxIF is set by hardware on completion of the Start.
3. SSPxIF is cleared by software.
4. The MSSPx module will wait the required start time before any other operation takes place.
5. The user loads the SSPxBUF with the slave address to transmit.
6. Address is shifted out the SDA pin until all eight bits are transmitted. Transmission begins as soon as SSPxBUF is written to.
7. The MSSPx module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPxCON2 register.
8. The MSSPx module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
9. The user loads the SSPxBUF with eight bits of data.
10. Data is shifted out the SDA pin until all eight bits are transmitted.
11. The MSSPx module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPxCON2 register.
12. Steps 8-11 are repeated for all transmitted data bytes.
13. The user generates a Stop or Restart condition by setting the PEN or RSEN bits of the SSPxCON2 register. Interrupt is generated once the Stop/Restart condition is complete.

FIGURE 29-28: I²C™ MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)



29.6.7 I²C MASTER MODE RECEPTION

Master mode reception (Figure 29-29) is enabled by programming the Receive Enable bit, RCEN bit of the SSPxCON2 register.

Note:	The MSSPx module must be in an Idle state before the RCEN bit is set or the RCEN bit will be disregarded.

The Baud Rate Generator begins counting and on each rollover, the state of the SCL pin changes (high-to-low/low-to-high) and data is shifted into the SSPxSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPxSR are loaded into the SSPxBUF, the BF flag bit is set, the SSPxIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSPx is now in Idle state awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable, ACKEN bit of the SSPxCON2 register.

29.6.7.1 BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPxBUF from SSPxSR. It is cleared when the SSPxBUF register is read.

29.6.7.2 SSPOV Status Flag

In receive operation, the SSPOV bit is set when eight bits are received into the SSPxSR and the BF flag bit is already set from a previous reception.

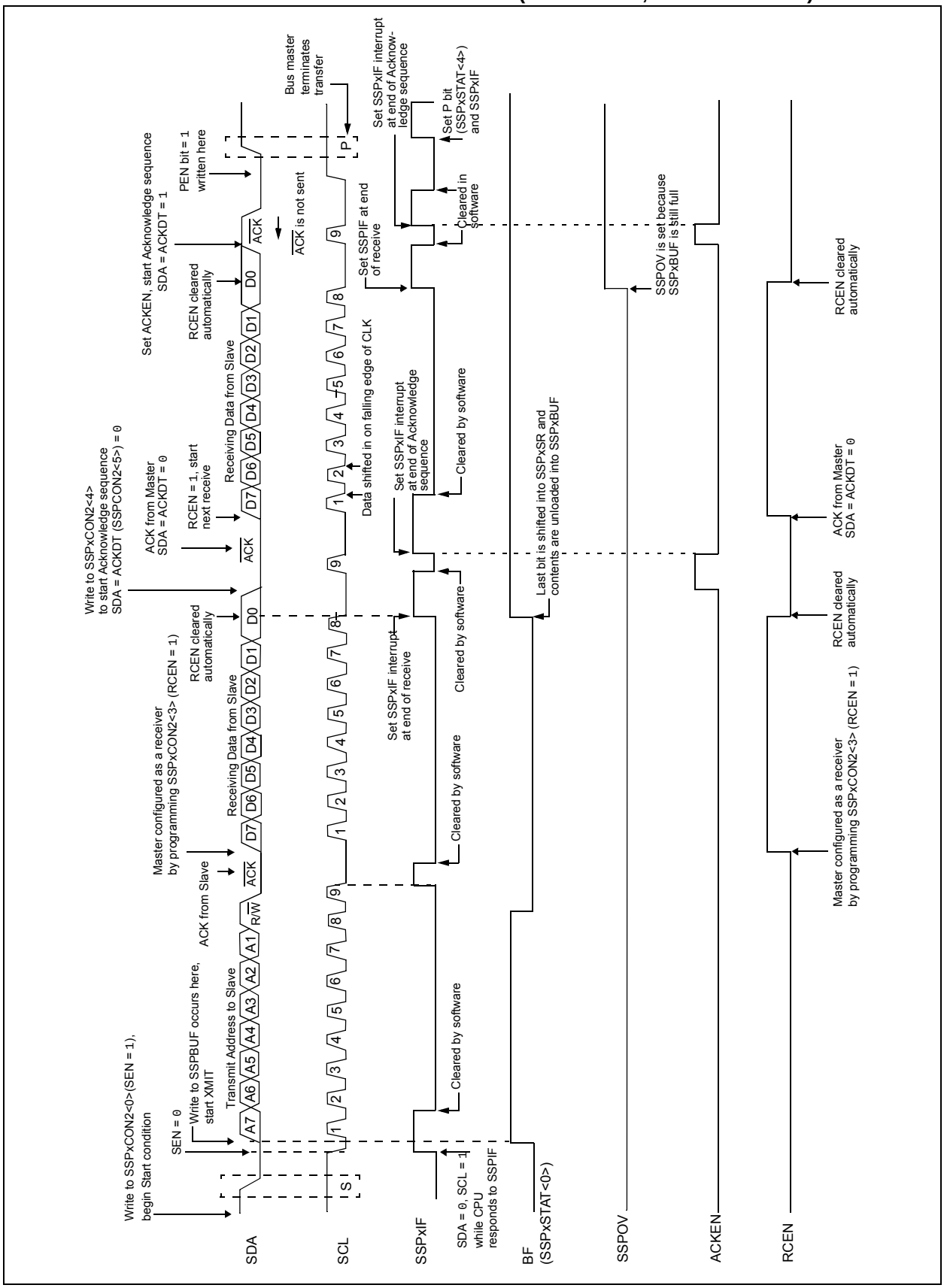
29.6.7.3 WCOL Status Flag

If the user writes the SSPxBUF when a receive is already in progress (i.e., SSPxSR is still shifting in a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

29.6.7.4 Typical Receive Sequence:

1. The user generates a Start condition by setting the SEN bit of the SSPxCON2 register.
2. SSPxIF is set by hardware on completion of the Start.
3. SSPxIF is cleared by software.
4. User writes SSPxBUF with the slave address to transmit and the R/$\overline{W}$ bit set.
5. Address is shifted out the SDA pin until all eight bits are transmitted. Transmission begins as soon as SSPxBUF is written to.
6. The MSSPx module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPxCON2 register.
7. The MSSPx module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
8. User sets the RCEN bit of the SSPxCON2 register and the master clocks in a byte from the slave.
9. After the eighth falling edge of SCL, SSPxIF and BF are set.
10. Master clears SSPxIF and reads the received byte from SSPxBUF, clears BF.
11. Master sets $\overline{ACK}$ value sent to slave in ACKDT bit of the SSPxCON2 register and initiates the $\overline{ACK}$ by setting the ACKEN bit.
12. Master's $\overline{ACK}$ is clocked out to the slave and SSPxIF is set.
13. User clears SSPIF.
14. Steps 8-13 are repeated for each received byte from the slave.
15. Master sends a not $\overline{ACK}$ or Stop to end communication.

FIGURE 29-29: I²C™ MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)



29.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN bit of the SSPxCON2 register. When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSPx module then goes into Idle mode (Figure 29-30).

29.6.8.1 WCOL Status Flag

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

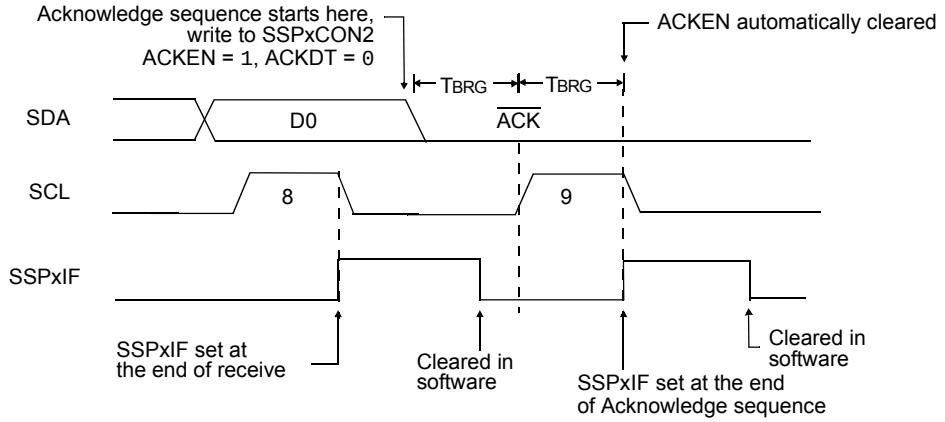
29.6.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN bit of the SSPxCON2 register. At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit of the SSPxSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 29-31).

29.6.9.1 WCOL Status Flag

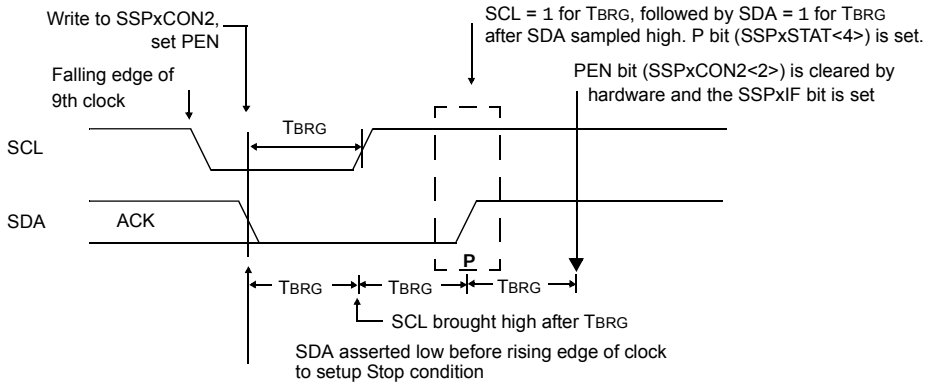
If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

FIGURE 29-30: ACKNOWLEDGE SEQUENCE WAVEFORM



Note: TBRG = one Baud Rate Generator period.

FIGURE 29-31: STOP CONDITION RECEIVE OR TRANSMIT MODE



Note: TBRG = one Baud Rate Generator period.

29.6.10 SLEEP OPERATION

While in Sleep mode, the I^2C slave module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

29.6.11 EFFECTS OF A RESET

A Reset disables the MSSPx module and terminates the current transfer.

29.6.12 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSPx module is disabled. Control of the I^2C bus may be taken when the P bit of the SSPxSTAT register is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the SSP interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed by hardware with the result placed in the BCL1IF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

29.6.13 MULTI -MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin is '0', then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLIF and reset the I^2C port to its Idle state (Figure 29-32).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPxBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

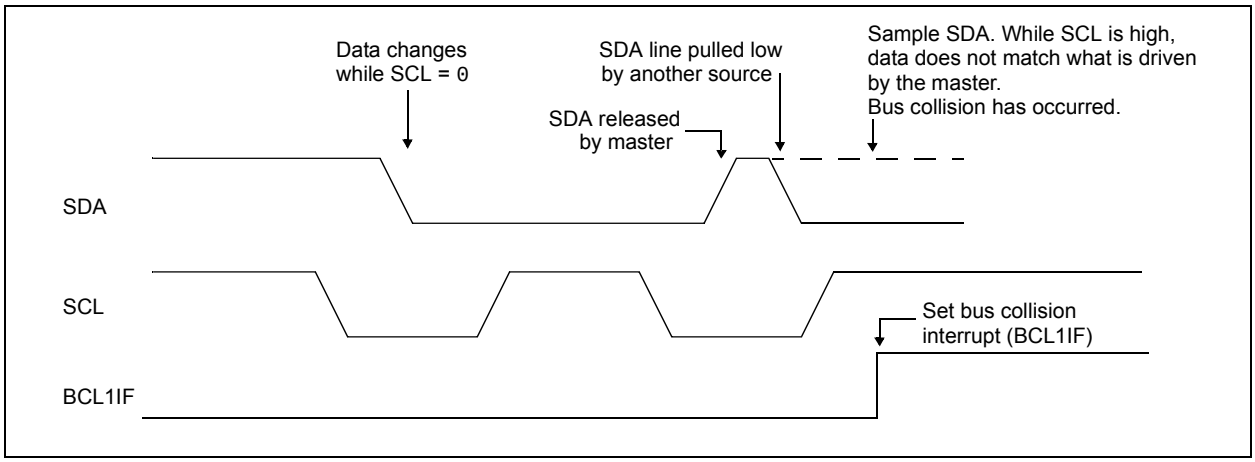
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPxCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPxIF bit will be set.

A write to the SSPxBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I^2C bus can be taken when the P bit is set in the SSPxSTAT register, or the bus is Idle and the S and P bits are cleared.

FIGURE 29-32: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE



29.6.13.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 29-33).

b) SCL is sampled low before SDA is asserted low (Figure 29-34).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

- the Start condition is aborted,
- the BCL1IF flag is set and
- the MSSPx module is reset to its Idle state (Figure 29-33).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded and counts down. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 29-35). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to zero; if the SCL pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

Note:	The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

FIGURE 29-33: BUS COLLISION DURING START CONDITION (SDA ONLY)

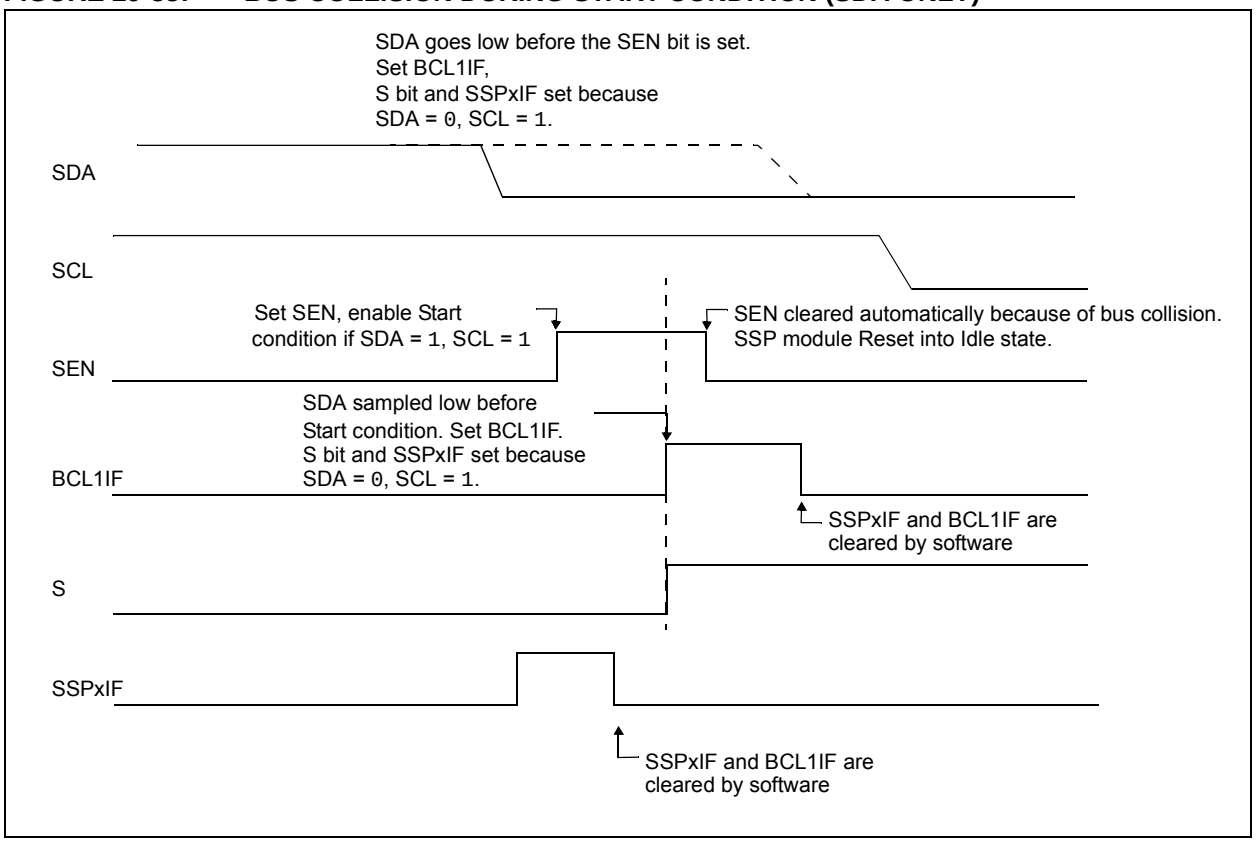


FIGURE 29-34: BUS COLLISION DURING START CONDITION (SCL = 0)

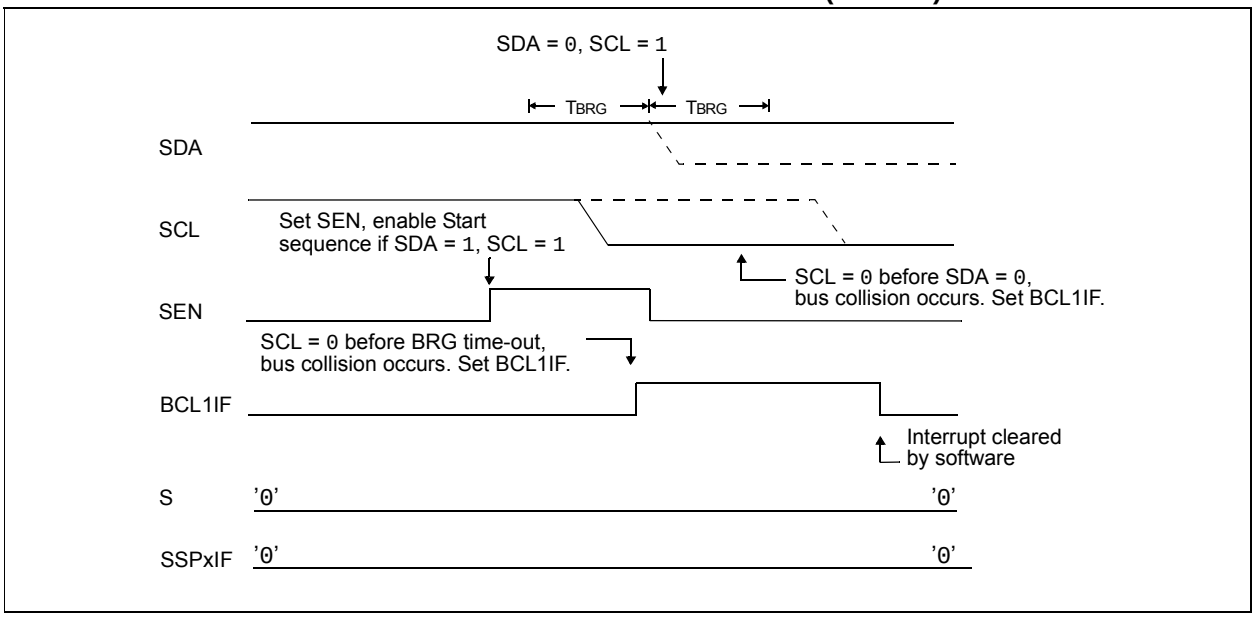
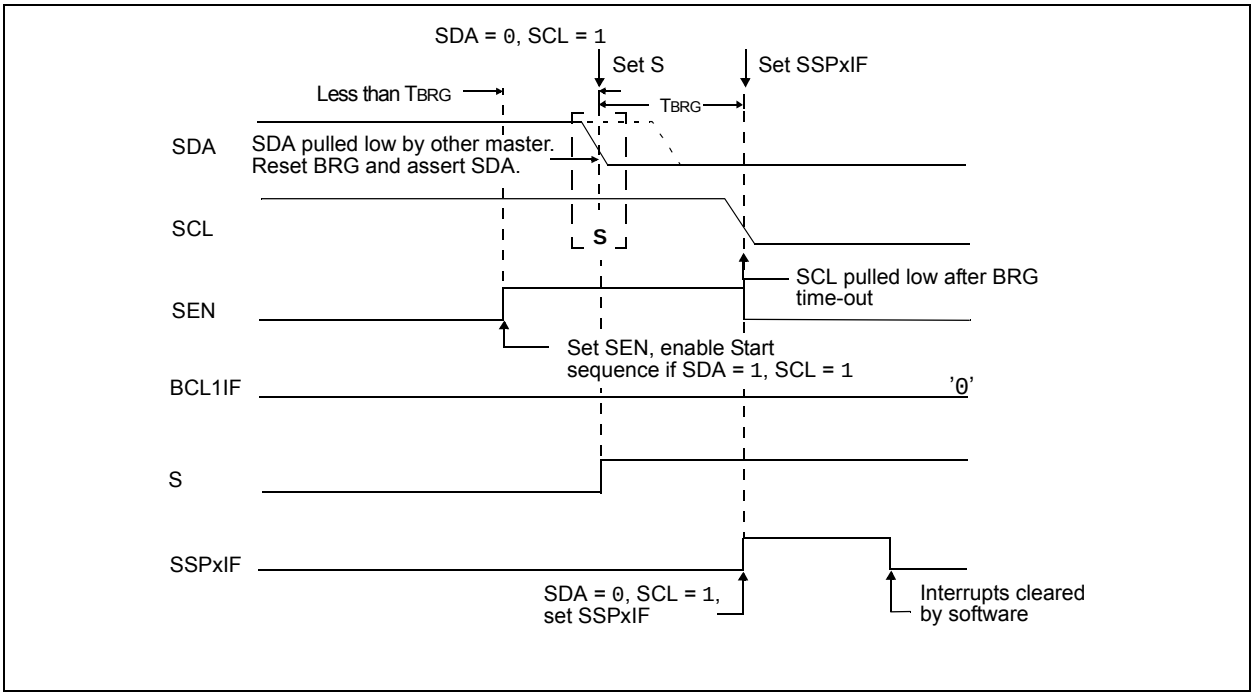


FIGURE 29-35: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION



29.6.13.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

a) A low level is sampled on SDA when SCL goes from low level to high level (Case 1).

b) SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1' (Case 2).

When the user releases SDA and the pin is allowed to float high, the BRG is loaded with SSPxADD and counts down to zero. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 29-36). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition, see Figure 29-37.

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

FIGURE 29-36: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

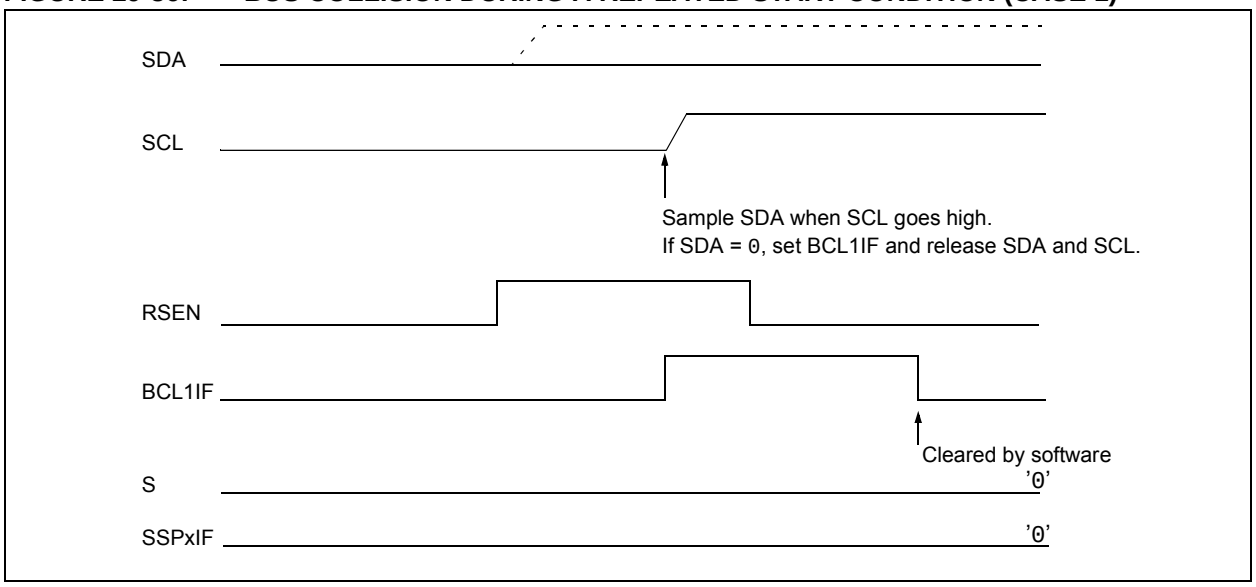
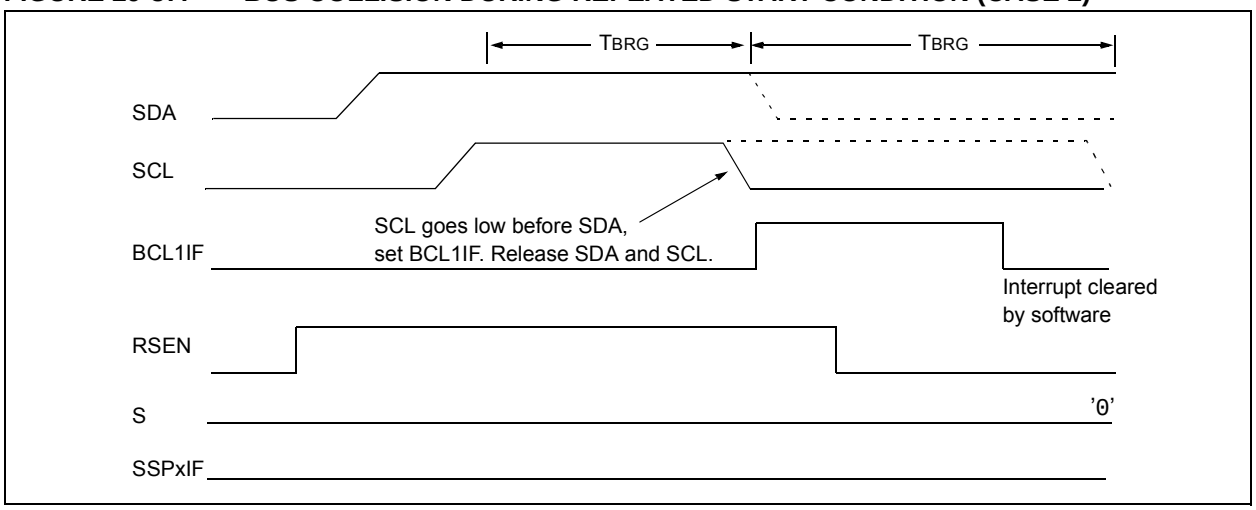


FIGURE 29-37: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



29.6.13.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

a) After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out (Case 1).

b) After the SCL pin is deasserted, SCL is sampled low before SDA goes high (Case 2).

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPxADD and counts down to zero. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 29-38). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 29-39).

FIGURE 29-38: BUS COLLISION DURING A STOP CONDITION (CASE 1)

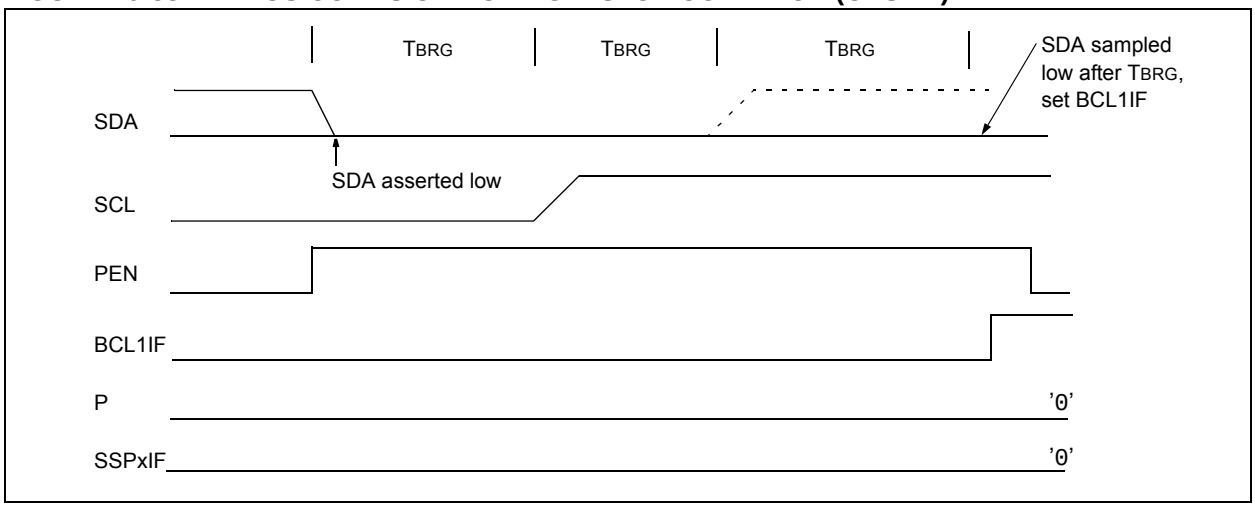
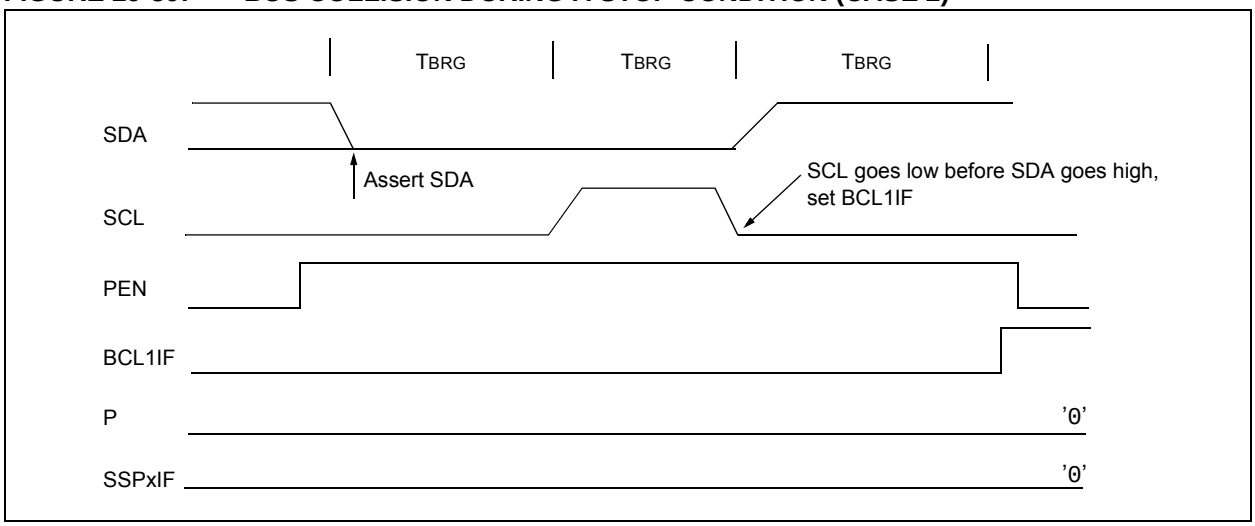


FIGURE 29-39: BUS COLLISION DURING A STOP CONDITION (CASE 2)



29.7 Baud Rate Generator

The MSSPx module has a Baud Rate Generator available for clock generation in both I^2C and SPI Master modes. The Baud Rate Generator (BRG) reload value is placed in the SSPxADD register (Register 29-6). When a write occurs to SSPxBUF, the Baud Rate Generator will automatically begin counting down.

Once the given operation is complete, the internal clock will automatically stop counting and the clock pin will remain in its last state.

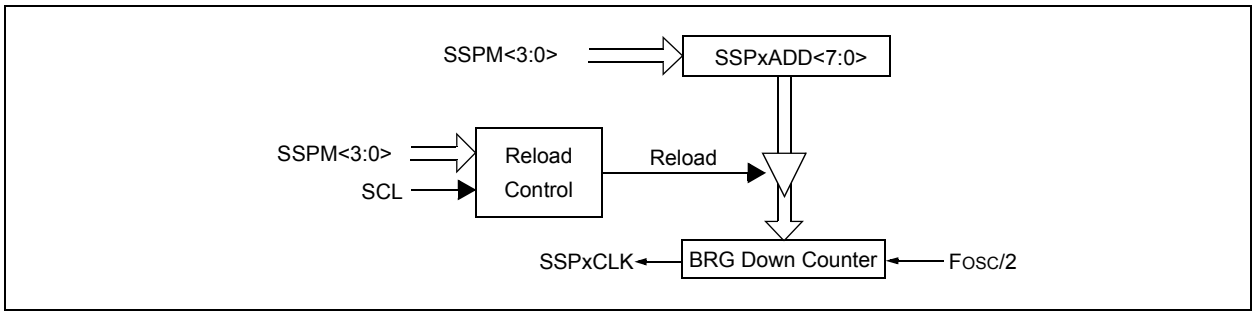
An internal signal “Reload” in Figure 29-40 triggers the value from SSPxADD to be loaded into the BRG counter. This occurs twice for each oscillation of the module clock line. The logic dictating when the reload signal is asserted depends on the mode the MSSPx is being operated in.

Table 29-4 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPADD.

EQUATION 29-1:

$$F_{CLOCK} = \frac{F_{OSC}}{(SSPxADD + 1)(4)}$$

FIGURE 29-40: BAUD RATE GENERATOR BLOCK DIAGRAM



Note: Values of 0x00, 0x01 and 0x02 are not valid for SSPxADD when used as a Baud Rate Generator for I^2C™. This is an implementation limitation.

TABLE 29-2: MSSP CLOCK RATE W/BRG

FOSC	FCY	BRG Value	FCLOCK (2 Rollovers of BRG)
32 MHz	8 MHz	13h	400 kHz
32 MHz	8 MHz	19h	308 kHz
32 MHz	8 MHz	4Fh	100 kHz
16 MHz	4 MHz	09h	400 kHz
16 MHz	4 MHz	0Ch	308 kHz
16 MHz	4 MHz	27h	100 kHz
4 MHz	1 MHz	09h	100 kHz

Note: Refer to the I/O port electrical specifications in Table 34-4 to ensure the system is designed to support IOL requirements.

29.8 Register Definitions: MSSP Control

REGISTER 29-1: SSPxSTAT: SSP STATUS REGISTER

R/W-0/0	R/W-0/0	R/HS/HC-0/0	R/HS/HC-0/0	R/HS/HC-0/0	R/HS/HC-0/0	R/HS/HC-0/0	R/HS/HC-0/0
SMP	CKE(1)	D/$\overline{A}$	P(2)	S(2)	R/$\overline{W}$	UA	BF
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/HC = Hardware set/clear

bit 7 **SMP:** SPI Data Input Sample bit
SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time
SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode
In I^2C Master or Slave mode:
1 = Slew rate control disabled for Standard Speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for High-Speed mode (400 kHz)

bit 6 **CKE:** SPI Clock Edge Select bit (SPI mode only)(1)
In SPI Master or Slave mode:
1 = Transmit occurs on transition from active to Idle clock state
0 = Transmit occurs on transition from Idle to active clock state
In I^2C™ mode only:
1 = Enable input logic so that thresholds are compliant with SMBus specification
0 = Disable SMBus specific inputs

bit 5 **D/$\overline{A}$:** Data/$\overline{Address}$ bit (I^2C™ mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit(2)
(I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)
1 = Indicates that a Stop bit has been detected last (this bit is '0' on Reset)
0 = Stop bit was not detected last

bit 3 **S:** Start bit (2)
(I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)
1 = Indicates that a Start bit has been detected last (this bit is '0' on Reset)
0 = Start bit was not detected last

bit 2 **R/$\overline{W}$:** Read/$\overline{Write}$ bit information (I^2C mode only)
This bit holds the R/$\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit, or not ACK bit.
In I^2C Slave mode:
1 = Read
0 = Write
In I^2C Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
OR-ing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Idle mode.

bit 1 **UA:** Update Address bit (10-bit I^2C mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit
Receive (SPI and I^2C modes):
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty
Transmit (I^2C mode only):
1 = Data transmit in progress (does not include the $\overline{ACK}$ and Stop bits), SSPBUF is full
0 = Data transmit complete (does not include the $\overline{ACK}$ and Stop bits), SSPBUF is empty

Note 1: Polarity of clock state is set by the CKP bit of the SSPCON register.
2: This bit is cleared on Reset and when SSPEN is cleared.

REGISTER 29-2: SSPxCON1: SSP CONTROL REGISTER 1

R/C/HS-0/0	R/C/HS-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
WCOL	SSPOV(1)	SSPEN	CKP	SSPM<3:0>			
bit 7							bit 0

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

bit 7 **WCOL:** Write Collision Detect bit (Transmit mode only)
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit(1)
In SPI mode:
1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. In Slave mode, the user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register (must be cleared in software).
0 = No overflow
In I^2C mode:
1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in Transmit mode (must be cleared in software).
0 = No overflow

bit 5 **SSPEN:** Synchronous Serial Port Enable bit
In both modes, when enabled, the following pins must be properly configured as input or output
In SPI mode:
1 = Enables serial port and configures SCK, SDO, SDI and $\overline{SS}$ as the source of the serial port pins(2)
0 = Disables serial port and configures these pins as I/O port pins
In I^2C mode:
1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins(3)
0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** Clock Polarity Select bit
In SPI mode:
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level
In I^2C Slave mode:
SCL release control
1 = Enable clock
0 = Holds clock low (clock stretch). (Used to ensure data setup time.)
In I^2C Master mode:
Unused in this mode

REGISTER 29-2: SSPxCON1: SSP CONTROL REGISTER 1 (CONTINUED)

bit 3-0 **SSPM<3:0>:** Synchronous Serial Port Mode Select bits
1111 = I^2C Slave mode, 10-bit address with Start and Stop bit interrupts enabled
1110 = I^2C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1101 = Reserved
1100 = Reserved
1011 = I^2C firmware controlled Master mode (slave idle)
1010 = SPI Master mode, clock = FOSC/(4 * (SSPADD+1))**(5)**
1001 = Reserved
1000 = I^2C Master mode, clock = FOSC / (4 * (SSPADD+1))**(4)**
0111 = I^2C Slave mode, 10-bit address
0110 = I^2C Slave mode, 7-bit address
0101 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control disabled, $\overline{SS}$ can be used as I/O pin
0100 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control enabled
0011 = SPI Master mode, clock = T2_match/2
0010 = SPI Master mode, clock = FOSC/64
0001 = SPI Master mode, clock = FOSC/16
0000 = SPI Master mode, clock = FOSC/4

Note 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.
2: When enabled, these pins must be properly configured as input or output. Use SSP1SSPPS, SSP1CLKPPS, SSP1DATPPS, and RxyPPS to select the pins.
3: When enabled, the SDA and SCL pins must be configured as inputs. Use SSPxCLKPPS, SSPxDATPPS, and RxyPPS to select the pins.
4: SSPADD values of 0, 1 or 2 are not supported for I^2C mode.
5: SSPADD value of '0' is not supported. Use SSPM = 0000 instead.

REGISTER 29-3: SSPxCON2: SSPx CONTROL REGISTER 2 (I²C™ MODE ONLY)(1)

R/W-0/0	R/HS/HC-0	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0	R/S/HC-0/0	R/S/HC-0/0	R/S/HC-0/0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

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

bit 7 **GCEN:** General Call Enable bit (in I²C Slave mode only)
1 = Enable interrupt when a general call address (0x00 or 00h) is received in the SSPSR
0 = General call address disabled

bit 6 **ACKSTAT:** Acknowledge Status bit (in I²C mode only)
1 = Acknowledge was not received
0 = Acknowledge was received

bit 5 **ACKDT:** Acknowledge Data bit (in I²C mode only)
In Receive mode:
Value transmitted when the user initiates an Acknowledge sequence at the end of a receive
1 = Not Acknowledge
0 = Acknowledge

bit 4 **ACKEN:** Acknowledge Sequence Enable bit (in I²C Master mode only)
In Master Receive mode:
1 = Initiate Acknowledge sequence on SDA and SCL pins, and transmit ACKDT data bit. Automatically cleared by hardware.
0 = Acknowledge sequence idle

bit 3 **RCEN:** Receive Enable bit (in I²C Master mode only)
1 = Enables Receive mode for I²C
0 = Receive idle

bit 2 **PEN:** Stop Condition Enable bit (in I²C Master mode only)
SCKMSSP Release Control:
1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Stop condition Idle

bit 1 **RSEN:** Repeated Start Condition Enable bit (in I²C Master mode only)
1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Repeated Start condition Idle

bit 0 **SEN:** Start Condition Enable/Stretch Enable bit
In Master mode:
1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Start condition Idle
In Slave mode:
1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)
0 = Clock stretching is disabled

Note 1: For bits ACKEN, RCEN, PEN, RSEN, SEN: If the I²C module is not in the Idle mode, this bit may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

REGISTER 29-4: SSPxCON3: SSP CONTROL REGISTER 3

R-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
ACKTIM(3)	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
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 **ACKTIM:** Acknowledge Time Status bit (I^2C™ mode only)(3)
1 = Indicates the I^2C bus is in an Acknowledge sequence, set on eighth falling edge of SCL clock
0 = Not an Acknowledge sequence, cleared on ninth rising edge of SCL clock

bit 6 **PCIE**: Stop Condition Interrupt Enable bit (I^2C mode only)
1 = Enable interrupt on detection of Stop condition
0 = Stop detection interrupts are disabled(2)

bit 5 **SCIE**: Start Condition Interrupt Enable bit (I^2C mode only)
1 = Enable interrupt on detection of Start or Restart conditions
0 = Start detection interrupts are disabled(2)

bit 4 **BOEN:** Buffer Overwrite Enable bit
In SPI Slave mode:(1)
1 = SSPBUF updates every time that a new data byte is shifted in ignoring the BF bit
0 = If new byte is received with BF bit of the SSPSTAT register already set, SSPOV bit of the SSPCON1 register is set, and the buffer is not updated
In I^2C Master mode and SPI Master mode:
This bit is ignored.
In I^2C Slave mode:
1 = SSPBUF is updated and $\overline{ACK}$ is generated for a received address/data byte, ignoring the state of the SSPOV bit only if the BF bit = 0.
0 = SSPBUF is only updated when SSPOV is clear

bit 3 **SDAHT:** SDA Hold Time Selection bit (I^2C mode only)
1 = Minimum of 300 ns hold time on SDA after the falling edge of SCL
0 = Minimum of 100 ns hold time on SDA after the falling edge of SCL

bit 2 **SBCDE:** Slave Mode Bus Collision Detect Enable bit (I^2C Slave mode only)
If, on the rising edge of SCL, SDA is sampled low when the module is outputting a high state, the BCL1IF bit of the PIR1 register is set, and bus goes idle
1 = Enable slave bus collision interrupts
0 = Slave bus collision interrupts are disabled

bit 1 **AHEN:** Address Hold Enable bit (I^2C Slave mode only)
1 = Following the eighth falling edge of SCL for a matching received address byte; CKP bit of the SSPCON1 register will be cleared and the SCL will be held low.
0 = Address holding is disabled

bit 0 **DHEN:** Data Hold Enable bit (I^2C Slave mode only)
1 = Following the eighth falling edge of SCL for a received data byte; slave hardware clears the CKP bit of the SSPCON1 register and SCL is held low.
0 = Data holding is disabled

Note 1: For daisy-chained SPI operation; allows the user to ignore all but the last received byte. SSPOV is still set when a new byte is received and BF = 1, but hardware continues to write the most recent byte to SSPBUF.
2: This bit has no effect in Slave modes that Start and Stop condition detection is explicitly listed as enabled.
3: The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is set.

REGISTER 29-5: SSPxMSK: SSP MASK REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
SSPxMSK<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 7-1 **SSPxMSK<7:1>:** Mask bits
1 = The received address bit n is compared to SSPADD<n> to detect I²C™ address match
0 = The received address bit n is not used to detect I²C address match

bit 0 **SSPxMSK<0>:** Mask bit for I²C Slave mode, 10-bit Address
I²C Slave mode, 10-bit address (SSPM<3:0> = 0111 or 1111):
1 = The received address bit 0 is compared to SSPADD<0> to detect I²C address match
0 = The received address bit 0 is not used to detect I²C address match
I²C Slave mode, 7-bit address:
MSK0 bit is ignored.

REGISTER 29-6: SSPxADD: MSSP ADDRESS AND BAUD RATE REGISTER (I²C™ MODE)

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
SSPxADD<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	

Master mode:

bit 7-0 **SSPxADD<7:0>:** Baud Rate Clock Divider bits
SCL pin clock period = ((ADD<7:0> + 1) *4)/FOSC

10-bit Slave mode – Most Significant Address Byte:

bit 7-3 **Not used:** Unused for Most Significant Address Byte. Bit state of this register is a "don't care". Bit pattern sent by master is fixed by I²C specification and must be equal to '11110'. However, those bits are compared by hardware and are not affected by the value in this register.

bit 2-1 **SSPxADD<2:1>:** Two Most Significant bits of 10-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a "don't care".

10-bit Slave mode – Least Significant Address Byte:

bit 7-0 **SSPxADD<7:0>:** Eight Least Significant bits of 10-bit address

7-bit Slave mode:

bit 7-1 **SSPxADD<7:1>:** 7-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a "don't care".

REGISTER 29-7: SSPxBUF: MSSP BUFFER 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
SSPxBUF<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 **SSPxBUF<7:0>:** MSSP Buffer bits

TABLE 29-3: SUMMARY OF REGISTERS ASSOCIATED WITH MSSPx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(3)	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
INLVLA(1)	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	144
TRISB(2)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB(2)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
INLVLB(2)	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	150
TRISC	TRISC7(2)	TRISC6(2)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7(2)	ANSC6(2)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
INLVLC(1)	INLVLC7(2)	INLVLC6(2)	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	157
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	100
PIR2	TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BCL2IF	TMR4IF	NCO1IF	106
PIE2	TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BCL2IE	TMR4IE	NCO1IE	101
SSPxSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	358
SSPxCON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				359
SSPxCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	361
SSPxCON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	362
SSPxMSK	SSPxMSK<7:0>								363
SSPxADD	SSPxADD<7:0>								363
SSPxBUF	SSPxBUF<7:0>								364
SSPxCLKPPS	—	—	—	SSPxCLKPPS<4:0>					160
SSPxDATPPS	—	—	—	SSPxDATPPS<4:0>					160
SSPxSSPPS	—	—	—	SSPxSSPPS<4:0>					160
RxyPPS	—	—	—	RxyPPS<4:0>					161

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by the MSSP module

Note 1: When using designated I^2C™ pins, the associated pin values in INLVLx will be ignored.

2: PIC16(L)F18345 only.

3: Unimplemented, read as '1'.

30.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART1)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART1) module is a serial I/O communications peripheral. It contains all the clock generators, shift registers and data buffers necessary to perform an input or output serial data transfer independent of device program execution. The EUSART1, also known as a Serial Communications Interface (SCI), can be configured as a full-duplex asynchronous system or half-duplex synchronous system. Full-Duplex mode is useful for communications with peripheral systems, such as CRT terminals and personal computers. Half-Duplex Synchronous mode is intended for communications with peripheral devices, such as A/D or D/$\overline{A}$ integrated circuits, serial EEPROMs or other microcontrollers. These devices typically do not have internal clocks for baud rate generation and require the external clock signal provided by a master synchronous device.

The EUSART1 module includes the following capabilities:

- Full-duplex asynchronous transmit and receive
- Two-character input buffer
- One-character output buffer
- Programmable 8-bit or 9-bit character length
- Address detection in 9-bit mode
- Input buffer overrun error detection
- Received character framing error detection
- Half-duplex synchronous master
- Half-duplex synchronous slave
- Programmable clock polarity in synchronous modes
- Sleep operation

The EUSART1 module implements the following additional features, making it ideally suited for use in Local Interconnect Network (LIN) bus systems:

- Automatic detection and calibration of the baud rate
- Wake-up on Break reception
- 13-bit Break character transmit

Block diagrams of the EUSART1 transmitter and receiver are shown in Figure 30-1 and Figure 30-2.

The EUSART1 transmit output (TX_out) is available to the TX/CK pin and internally to the following peripherals:

- Configurable Logic Cell (CLC)

FIGURE 30-1: EUSART1 TRANSMIT BLOCK DIAGRAM

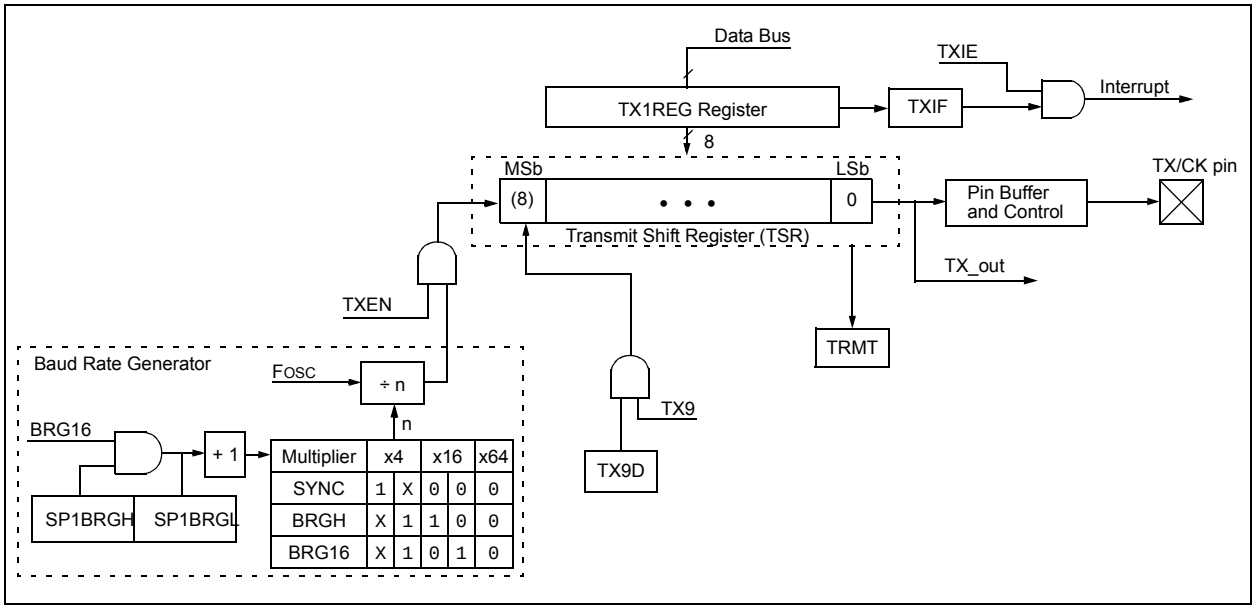
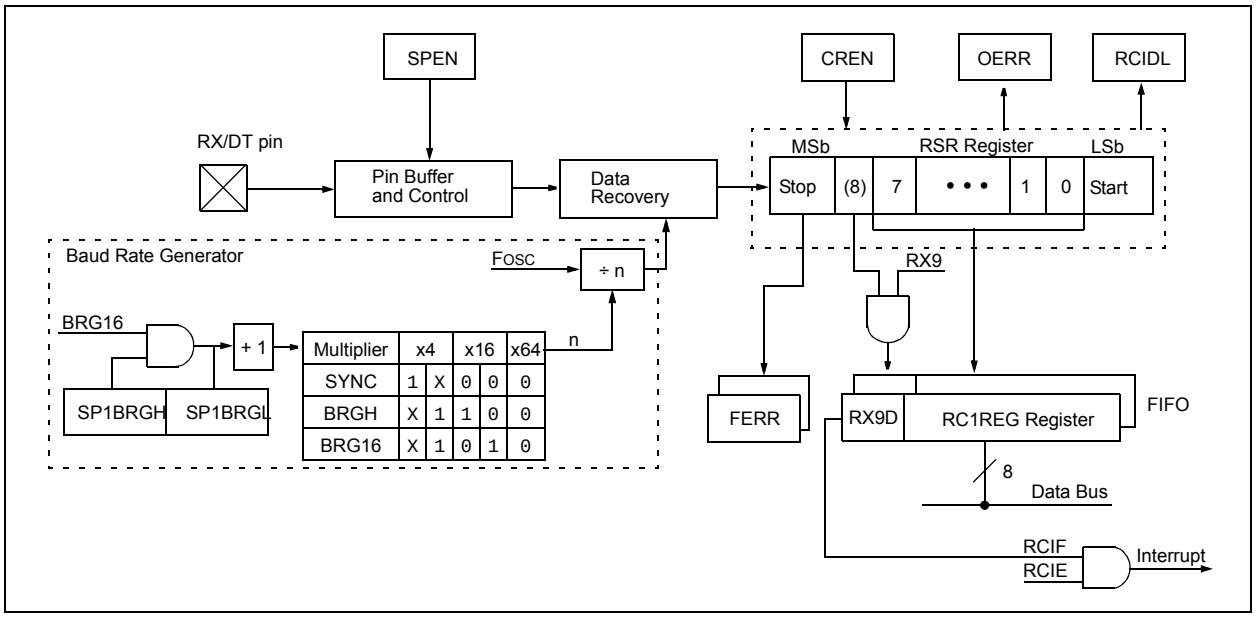


FIGURE 30-2: EUSART1 RECEIVE BLOCK DIAGRAM



The operation of the EUSART1 module is controlled through three registers:

- Transmit Status and Control (TX1STA)
- Receive Status and Control (RC1STA)
- Baud Rate Control (BAUD1CON)

These registers are detailed in Register 30-1, Register 30-2 and Register 30-3, respectively.

The RX and CK input pins are selected with the RXPPS and CKPPS registers, respectively. TX, CK, and DT output pins are selected with each pin's RxyPPS register. Since the RX input is coupled with the DT output in Synchronous mode, it is the user's responsibility to select the same pin for both of these functions when operating in Synchronous mode. The EUSART1 control logic will control the data direction drivers automatically.

30.1 EUSART1 Asynchronous Mode

The EUSART1 transmits and receives data using the standard non-return-to-zero (NRZ) format. NRZ is implemented with two levels: a VOH mark state which represents a '1' data bit, and a VOL space state which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the Mark state. Each character transmission consists of one Start bit followed by eight or nine data bits and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is eight bits. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 8-bit/16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See Table 30-3 for examples of baud rate configurations.

The EUSART1 transmits and receives the LSb first. The EUSART1's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

30.1.1 EUSART1 ASYNCHRONOUS TRANSMITTER

The EUSART1 transmitter block diagram is shown in Figure 30-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TX1REG register.

30.1.1.1 Enabling the Transmitter

The EUSART1 transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN = 1
- SYNC = 0
- SPEN = 1

All other EUSART1 control bits are assumed to be in their default state.

Setting the TXEN bit of the TX1STA register enables the transmitter circuitry of the EUSART1. Clearing the SYNC bit of the TX1STA register configures the EUSART1 for asynchronous operation. Setting the SPEN bit of the RC1STA register enables the EUSART1 and automatically configures the TX/CK I/O pin as an output. If the TX/CK pin is shared with an analog peripheral, the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

Note:	The TXIF Transmitter Interrupt flag is set when the TXEN enable bit is set.

30.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TX1REG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TX1REG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TX1REG until the Stop bit of the previous character has been transmitted. The pending character in the TX1REG is then transferred to the TSR in one TCY immediately following the Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the transfer of the data to the TSR from the TX1REG.

30.1.1.3 Transmit Data Polarity

The polarity of the transmit data can be controlled with the SCKP bit of the BAUD1CON register. The default state of this bit is '0' which selects high true transmit idle and data bits. Setting the SCKP bit to '1' will invert the transmit data resulting in low true idle and data bits. The SCKP bit controls transmit data polarity in Asynchronous mode only. In Synchronous mode, the SCKP bit has a different function. See **Section 30.4.1.2 "Clock Polarity"**.

30.1.1.4 Transmit Interrupt Flag

The TXIF interrupt flag bit of the PIR1 register is set whenever the EUSART1 transmitter is enabled and no character is being held for transmission in the TX1REG. In other words, the TXIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the TX1REG. The TXIF flag bit is not cleared immediately upon writing TX1REG. TXIF becomes valid in the second instruction cycle following the write execution. Polling TXIF immediately following the TX1REG write will return invalid results. The TXIF bit is read-only, it cannot be set or cleared by software.

The TXIF interrupt can be enabled by setting the TXIE interrupt enable bit of the PIE1 register. However, the TXIF flag bit will be set whenever the TX1REG is empty, regardless of the state of TXIE enable bit.

To use interrupts when transmitting data, set the TXIE bit only when there is more data to send. Clear the TXIE interrupt enable bit upon writing the last character of the transmission to the TX1REG.

30.1.1.5 TSR Status

The TRMT bit of the TX1STA register indicates the status of the TSR register. This is a read-only bit. The TRMT bit is set when the TSR register is empty and is cleared when a character is transferred to the TSR register from the TX1REG. The TRMT bit remains clear until all bits have been shifted out of the TSR register. No interrupt logic is tied to this bit, so the user has to poll this bit to determine the TSR status.

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

30.1.1.6 Transmitting 9-bit Characters

The EUSART1 supports 9-bit character transmissions. When the TX9 bit of the TX1STA register is set, the EUSART1 will shift nine bits out for each character transmitted. The TX9D bit of the TX1STA register is the ninth, and Most Significant data bit. When transmitting 9-bit data, the TX9D data bit must be written before writing the eight Least Significant bits into the TX1REG. All nine bits of data will be transferred to the TSR shift register immediately after the TX1REG is written.

A special 9-bit Address mode is available for use with multiple receivers. See **Section 30.1.2.7 "Address Detection"** for more information on the Address mode.

30.1.1.7 Asynchronous Transmission Set-up

1. Initialize the SP1BRGH, SP1BRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 30.3 "EUSART1 Baud Rate Generator (BRG)"**).
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If 9-bit transmission is desired, set the TX9 control bit. A set ninth data bit will indicate that the eight Least Significant data bits are an address when the receiver is set for address detection.
4. Set SCKP bit if inverted transmit is desired.
5. Enable the transmission by setting the TXEN control bit. This will cause the TXIF interrupt bit to be set.
6. If interrupts are desired, set the TXIE interrupt enable bit of the PIE1 register. An interrupt will occur immediately provided that the GIE and PEIE bits of the INTCON register are also set.
7. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D data bit.
8. Load 8-bit data into the TX1REG register. This will start the transmission.

FIGURE 30-3: ASYNCHRONOUS TRANSMISSION

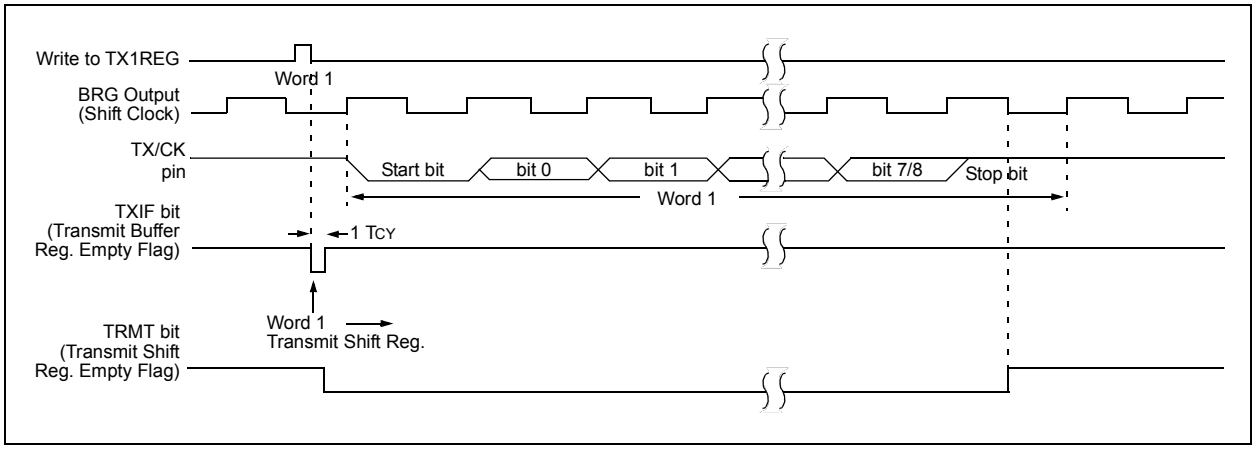
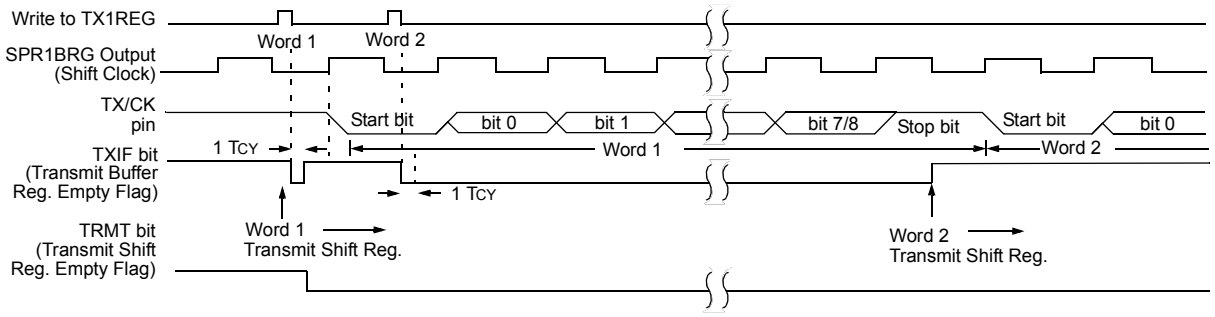


FIGURE 30-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)



Note: This timing diagram shows two consecutive transmissions.

30.1.2 EUSART1 ASYNCHRONOUS RECEIVER

The Asynchronous mode is typically used in RS-232 systems. The receiver block diagram is shown in Figure 30-2. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all eight or nine bits of the character have been shifted in, they are immediately transferred to a two character First-In-First-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third character before software must start servicing the EUSART1 receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RC1REG register.

30.1.2.1 Enabling the Receiver

The EUSART1 receiver is enabled for asynchronous operation by configuring the following three control bits:

- CREN = 1
- SYNC = 0
- SPEN = 1

All other EUSART1 control bits are assumed to be in their default state.

Setting the CREN bit of the RC1STA register enables the receiver circuitry of the EUSART1. Clearing the SYNC bit of the TX1STA register configures the EUSART1 for asynchronous operation. Setting the SPEN bit of the RC1STA register enables the EUSART1. The programmer must set the corresponding TRIS bit to configure the RX/DT I/O pin as an input.

Note:	If the RX/DT function is on an analog pin, the corresponding ANSEL bit must be cleared for the receiver to function.

30.1.2.2 Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data recovery circuit samples a '0' in the Stop bit position then a framing error is set for this character, otherwise the framing error is cleared for this character. See **Section 30.1.2.4 "Receive Framing Error"** for more information on framing errors.

Immediately after all data bits and the Stop bit have been received, the character in the RSR is transferred to the EUSART1 receive FIFO and the RCIF interrupt flag bit of the PIR1 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RC1REG register.

Note:	If the receive FIFO is overrun, no additional characters will be received until the Overrun condition is cleared. See **Section 30.1.2.5 "Receive Overrun Error"** for more information on overrun errors.

30.1.2.3 Receive Interrupts

The RCIF interrupt flag bit of the PIR1 register is set whenever the EUSART1 receiver is enabled and there is an unread character in the receive FIFO. The RCIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RCIF interrupts are enabled by setting all of the following bits:

- RCIE, Interrupt Enable bit of the PIE1 register
- PEIE, Peripheral Interrupt Enable bit of the INTCON register
- GIE, Global Interrupt Enable bit of the INTCON register

The RCIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

30.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error Status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RC1STA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RC1REG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = 1) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RC1STA register which resets the EUSART1. Clearing the CREN bit of the RC1STA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

Note:	If all receive characters in the receive FIFO have framing errors, repeated reads of the RC1REG will not clear the FERR bit.

30.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RC1STA register is set. The characters already in the FIFO buffer can be read but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RC1STA register or by resetting the EUSART1 by clearing the SPEN bit of the RC1STA register.

30.1.2.6 Receiving 9-bit Characters

The EUSART1 supports 9-bit character reception. When the RX9 bit of the RC1STA register is set the EUSART1 will shift nine bits into the RSR for each character received. The RX9D bit of the RC1STA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RC1REG.

30.1.2.7 Address Detection

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems. Address detection is enabled by setting the ADDEN bit of the RC1STA register.

Address detection requires 9-bit character reception. When address detection is enabled, only characters with the ninth data bit set will be transferred to the receive FIFO buffer, thereby setting the RCIF interrupt bit. All other characters will be ignored.

Upon receiving an address character, user software determines if the address matches its own. Upon address match, user software must disable address detection by clearing the ADDEN bit before the next Stop bit occurs. When user software detects the end of the message, determined by the message protocol used, software places the receiver back into the Address Detection mode by setting the ADDEN bit.

30.1.2.8 Asynchronous Reception Setup

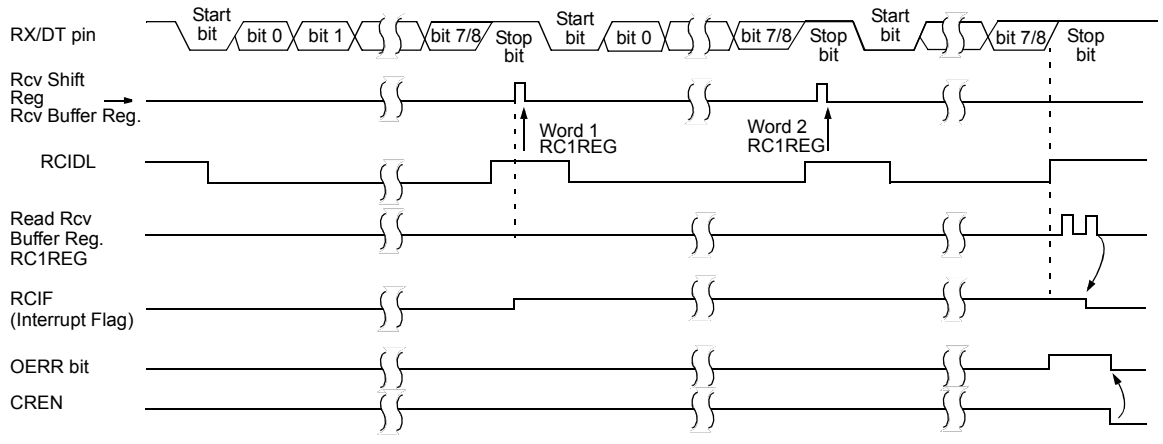
1. Initialize the SP1BRGH, SP1BRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 30.3 "EUSART1 Baud Rate Generator (BRG)"**).
2. Clear the ANSEL bit for the RX pin (if applicable).
3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
4. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
5. If 9-bit reception is desired, set the RX9 bit.
6. Enable reception by setting the CREN bit.
7. The RCIF interrupt flag bit will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
8. Read the RC1STA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.
9. Get the received eight Least Significant data bits from the receive buffer by reading the RC1REG register.
10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

30.1.2.9 9-bit Address Detection Mode Setup

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

1. Initialize the SP1BRGH, SP1BRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 30.3 "EUSART1 Baud Rate Generator (BRG)"**).
2. Clear the ANSEL bit for the RX pin (if applicable).
3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
4. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
5. Enable 9-bit reception by setting the RX9 bit.
6. Enable address detection by setting the ADDEN bit.
7. Enable reception by setting the CREN bit.
8. The RCIF interrupt flag bit will be set when a character with the ninth bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
9. Read the RC1STA register to get the error flags. The ninth data bit will always be set.
10. Get the received eight Least Significant data bits from the receive buffer by reading the RC1REG register. Software determines if this is the device's address.
11. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
12. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.

FIGURE 30-5: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX input. The RC1REG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

30.2 Clock Accuracy with Asynchronous Operation

The factory calibrates the internal oscillator block output (INTOSC). However, the INTOSC frequency may drift as VDD or temperature changes, and this directly affects the asynchronous baud rate. Two methods may be used to adjust the baud rate clock, but both require a reference clock source of some kind.

The first (preferred) method uses the OSCTUNE register to adjust the INTOSC output. Adjusting the value in the OSCTUNE register allows for fine resolution changes to the system clock source. See **Section 6.2.2.3 "Internal Oscillator Frequency Adjustment"** for more information.

The other method adjusts the value in the Baud Rate Generator. This can be done automatically with the Auto-Baud Detect feature (see **Section 30.3.1 "Auto-Baud Detect"**). There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change in the peripheral clock frequency.

30.3 EUSART1 Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is an 8-bit or 16-bit timer that is dedicated to the support of both the asynchronous and synchronous EUSART1 operation. By default, the BRG operates in 8-bit mode. Setting the BRG16 bit of the BAUD1CON register selects 16-bit mode.

The SP1BRGH, SP1BRGL register pair determines the period of the free running baud rate timer. In Asynchronous mode the multiplier of the baud rate period is determined by both the BRGH bit of the TX1STA register and the BRG16 bit of the BAUD1CON register. In Synchronous mode, the BRGH bit is ignored.

Table 30-1 contains the formulas for determining the baud rate. Example 30-1 provides a sample calculation for determining the baud rate and baud rate error.

Typical baud rates and error values for various asynchronous modes have been computed for your convenience and are shown in Table 30-3. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

Writing a new value to the SP1BRGH, SP1BRGL register pair causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit to make sure that the receive operation is idle before changing the system clock.

EXAMPLE 30-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit SPR1BRG:

$$\text{Desired Baud Rate} = \frac{FOSC}{64([SPBRGH:SPBRGL] + 1)}$$

Solving for SP1BRGH:SP1BRGL:

$$X = \frac{\frac{FOSC}{\text{Desired Baud Rate}}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

$$\text{Calculated Baud Rate} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$Error = \frac{\text{Calc. Baud Rate} - \text{Desired Baud Rate}}{\text{Desired Baud Rate}}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

30.3.1 AUTO-BAUD DETECT

The EUSART1 module supports automatic detection and calibration of the baud rate.

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

Setting the ABDEN bit of the BAUD1CON register starts the auto-baud calibration sequence. While the ABD sequence takes place, the EUSART1 state machine is held in Idle. On the first rising edge of the receive line, after the Start bit, the SPBRG begins counting up using the BRG counter clock as shown in Figure 30-6. The fifth rising edge will occur on the RX pin at the end of the eighth bit period. At that time, an accumulated value totaling the proper BRG period is left in the SP1BRGH, SP1BRGL register pair, the ABDEN bit is automatically cleared and the RCIF interrupt flag is set. The value in the RC1REG needs to be read to clear the RCIF interrupt. RC1REG content should be discarded. When calibrating for modes that do not use the SP1BRGH register the user can verify that the SP1BRGL register did not overflow by checking for 00h in the SP1BRGH register.

The BRG auto-baud clock is determined by the BRG16 and BRGH bits as shown in Table 30-1. During ABD, both the SP1BRGH and SP1BRGL registers are used as a 16-bit counter, independent of the BRG16 bit setting. While calibrating the baud rate period, the SP1BRGH and SP1BRGL registers are clocked at 1/8th the BRG base clock rate. The resulting byte measurement is the average bit time when clocked at full speed.

Note 1: If the WUE bit is set with the ABDEN bit, auto-baud detection will occur on the byte following the Break character (see **Section 30.3.3 "Auto-Wake-up on Break"**).

2: It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART1 baud rates are not possible.

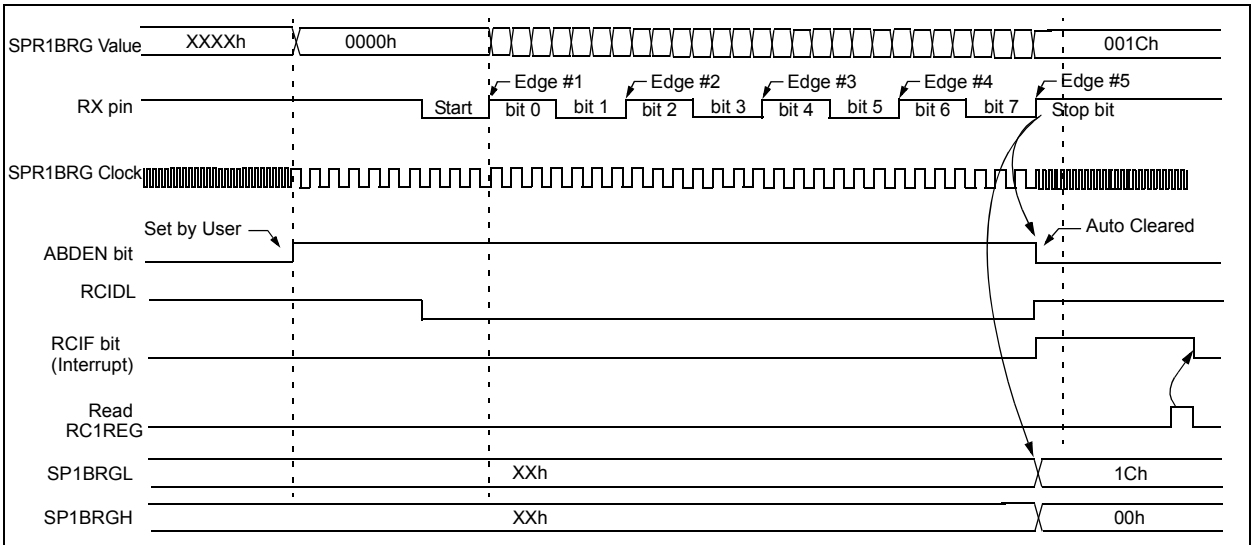
3: During the auto-baud process, the auto-baud counter starts counting at one. Upon completion of the auto-baud sequence, to achieve maximum accuracy, subtract 1 from the SP1BRGH:SP1BRGL register pair.

TABLE 30-1: BRG COUNTER CLOCK RATES

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

Note: During the ABD sequence, SP1BRGL and SP1BRGH registers are both used as a 16-bit counter, independent of the BRG16 setting.

FIGURE 30-6: AUTOMATIC BAUD RATE CALIBRATION



Note 1: The ABD sequence requires the **EUSART1** module to be configured in Asynchronous mode.

30.3.2 AUTO-BAUD OVERFLOW

During the course of automatic-baud detection, the ABDOVF bit of the BAUDxCON register will be set if the baud rate counter overflows before the fifth rising edge is detected on the RX pin. The ABDOVF bit indicates that the counter has exceeded the maximum count that can fit in the 16 bits of the SPxBRGH:SPxBRGL register pair. The Overflow condition will set the RCIF flag. The counter continues to count until the fifth rising edge is detected on the RX pin. The RCIDL bit will remain false ('0') until the fifth rising edge at which time the RCIDL bit will be set. If the RCREG is read after the overflow occurs but before the fifth rising edge, then the fifth rising edge will set the RCIF again.

Terminating the auto-baud process early to clear an Overflow condition will prevent proper detection of the sync character fifth rising edge. If any falling edges of the sync character have not yet occurred when the ABDEN bit is cleared, then those will be falsely detected as Start bits. The following steps are recommended to clear the Overflow condition:

1. Read RCREG to clear RCIF
2. If RCIDL is zero, then wait for RCIF and repeat step 1
3. Clear the ABDOVF bit

30.3.3 AUTO-WAKE-UP ON BREAK

During Sleep mode, all clocks to the EUSART1 are suspended. Because of this, the Baud Rate Generator is inactive and a proper character reception cannot be performed. The Auto-Wake-up feature allows the controller to wake-up due to activity on the RX/DT line. This feature is available only in Asynchronous mode.

The Auto-Wake-up feature is enabled by setting the WUE bit of the BAUD1CON register. Once set, the normal receive sequence on RX/DT is disabled, and the EUSART1 remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RX/DT line. (This coincides with the start of a Sync Break or a wake-up signal character for the LIN protocol.)

The EUSART1 module generates an RCIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 30-7), and asynchronously if the device is in Sleep mode (Figure 30-8). The Interrupt condition is cleared by reading the RC1REG register.

The WUE bit is automatically cleared by the low-to-high transition on the RX line at the end of the Break. This signals to the user that the Break event is over. At this point, the EUSART1 module is in Idle mode waiting to receive the next character.

30.3.3.1 Special Considerations

Break Character

To avoid character errors or character fragments during a wake-up event, the wake-up character must be all zeros.

When the wake-up is enabled the function works independent of the low time on the data stream. If the WUE bit is set and a valid non-zero character is received, the low time from the Start bit to the first rising edge will be interpreted as the wake-up event. The remaining bits in the character will be received as a fragmented character and subsequent characters can result in framing or overrun errors.

Therefore, the initial character in the transmission must be all '0's. This must be ten or more bit times, 13-bit times recommended for LIN bus, or any number of bit times for standard RS-232 devices.

Oscillator Start-up Time

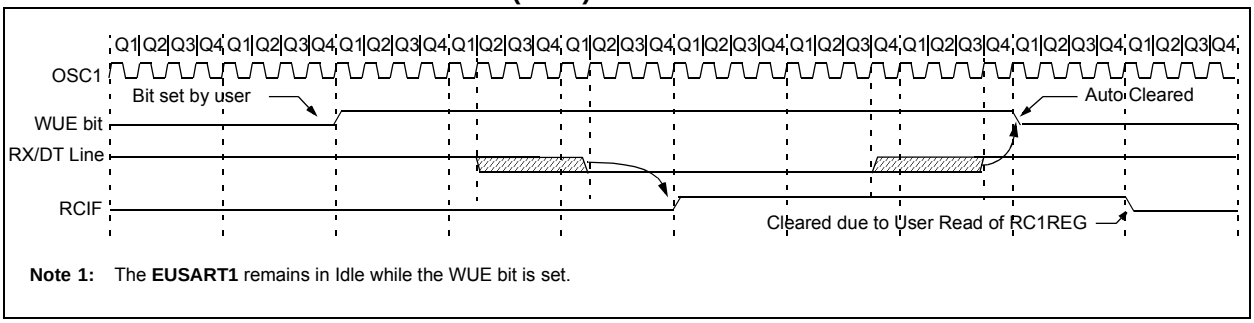
Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., LP, XT or HS/PLL mode). The Sync Break (or wake-up signal) character must be of sufficient length, and be followed by a sufficient interval, to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART1.

WUE Bit

The wake-up event causes a receive interrupt by setting the RCIF bit. The WUE bit is cleared in hardware by a rising edge on RX/DT. The Interrupt condition is then cleared in software by reading the RC1REG register and discarding its contents.

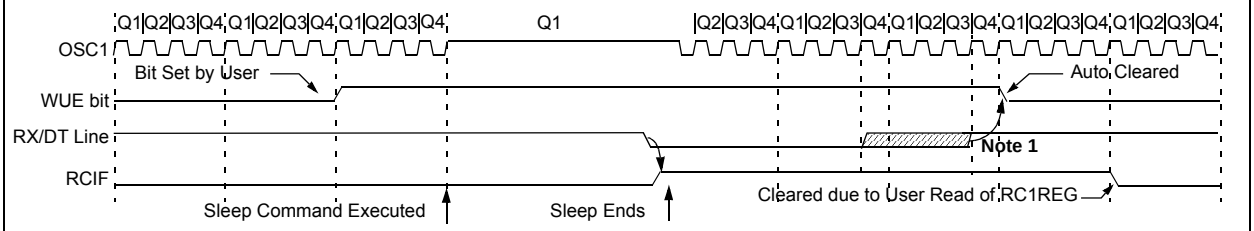
To ensure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process before setting the WUE bit. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

FIGURE 30-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION



Note 1: The **EUSART1** remains in Idle while the WUE bit is set.

FIGURE 30-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP



Note 1: If the wake-up event requires long oscillator warm-up time, the automatic clearing of the WUE bit can occur while the `stposc` signal is still active. This sequence should not depend on the presence of Q clocks.

2: The **EUSART1** remains in Idle while the WUE bit is set.

30.3.4 BREAK CHARACTER SEQUENCE

The EUSART1 module has the capability of sending the special Break character sequences that are required by the LIN bus standard. A Break character consists of a Start bit, followed by 12 '0' bits and a Stop bit.

To send a Break character, set the SENDB and TXEN bits of the TX1STA register. The Break character transmission is then initiated by a write to the TX1REG. The value of data written to TX1REG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

The TRMT bit of the TX1STA register indicates when the transmit operation is active or idle, just as it does during normal transmission. See Figure 30-9 for the timing of the Break character sequence.

30.3.4.1 Break and Sync Transmit Sequence

The following sequence will start a message frame header made up of a Break, followed by an auto-baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART1 for the desired mode.
2. Set the TXEN and SENDB bits to enable the Break sequence.
3. Load the TX1REG with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TX1REG to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware and the Sync character is then transmitted.

When the TX1REG becomes empty, as indicated by the TXIF, the next data byte can be written to TX1REG.

30.3.5 RECEIVING A BREAK CHARACTER

The Enhanced EUSART1 module can receive a Break character in two ways.

The first method to detect a Break character uses the FERR bit of the RC1STA register and the received data as indicated by RC1REG. The Baud Rate Generator is assumed to have been initialized to the expected baud rate.

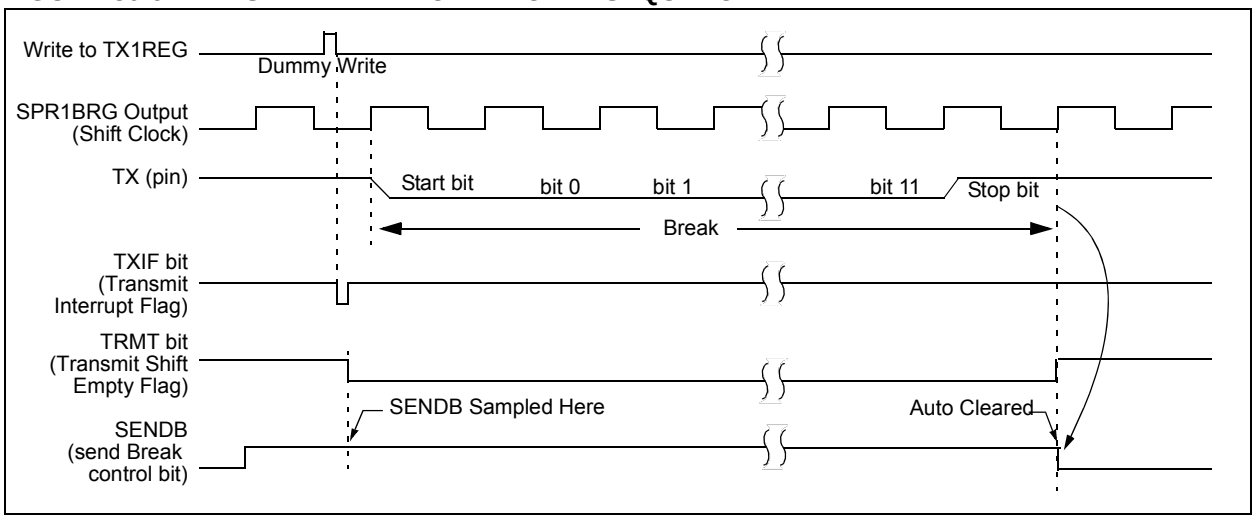
A Break character has been received when;

- RCIF bit is set
- FERR bit is set
- RC1REG = 00h

The second method uses the Auto-Wake-up feature described in **Section 30.3.3 "Auto-Wake-up on Break"**. By enabling this feature, the EUSART1 will sample the next two transitions on RX/DT, cause an RCIF interrupt, and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Detect feature. For both methods, the user can set the ABDEN bit of the BAUD1CON register before placing the EUSART1 in Sleep mode.

FIGURE 30-9: SEND BREAK CHARACTER SEQUENCE



30.4 EUSART1 Synchronous Mode

Synchronous serial communications are typically used in systems with a single master and one or more slaves. The master device contains the necessary circuitry for baud rate generation and supplies the clock for all devices in the system. Slave devices can take advantage of the master clock by eliminating the internal clock generation circuitry.

There are two signal lines in Synchronous mode: a bidirectional data line and a clock line. Slaves use the external clock supplied by the master to shift the serial data into and out of their respective receive and transmit shift registers. Since the data line is bidirectional, synchronous operation is half-duplex only. Half-duplex refers to the fact that master and slave devices can receive and transmit data but not both simultaneously. The EUSART1 can operate as either a master or slave device.

Start and Stop bits are not used in synchronous transmissions.

30.4.1 SYNCHRONOUS MASTER MODE

The following bits are used to configure the EUSART1 for synchronous master operation:

- SYNC = 1
- CSRC = 1
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TX1STA register configures the device for synchronous operation. Setting the CSRC bit of the TX1STA register configures the device as a master. Clearing the SREN and CREN bits of the RC1STA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RC1STA register enables the EUSART1.

30.4.1.1 Master Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a master transmits the clock on the TX/CK line. The TX/CK pin output driver is automatically enabled when the EUSART1 is configured for synchronous transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One clock cycle is generated for each data bit. Only as many clock cycles are generated as there are data bits.

30.4.1.2 Clock Polarity

A clock polarity option is provided for Microwire compatibility. Clock polarity is selected with the SCKP bit of the BAUD1CON register. Setting the SCKP bit sets the clock Idle state as high. When the SCKP bit is set, the data changes on the falling edge of each clock. Clearing the SCKP bit sets the Idle state as low. When the SCKP bit is cleared, the data changes on the rising edge of each clock.

30.4.1.3 Synchronous Master Transmission

Data is transferred out of the device on the RX/DT pin. The RX/DT and TX/CK pin output drivers are automatically enabled when the EUSART1 is configured for synchronous master transmit operation.

A transmission is initiated by writing a character to the TX1REG register. If the TSR still contains all or part of a previous character the new character data is held in the TX1REG until the last bit of the previous character has been transmitted. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TX1REG is immediately transferred to the TSR. The transmission of the character commences immediately following the transfer of the data to the TSR from the TX1REG.

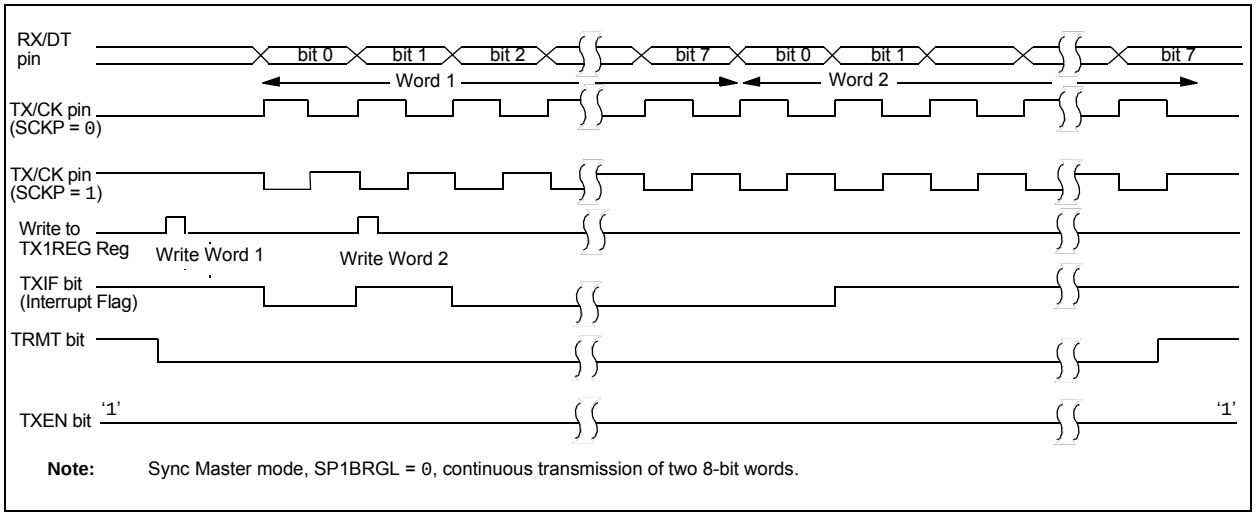
Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

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

30.4.1.4 Synchronous Master Transmission Set-up

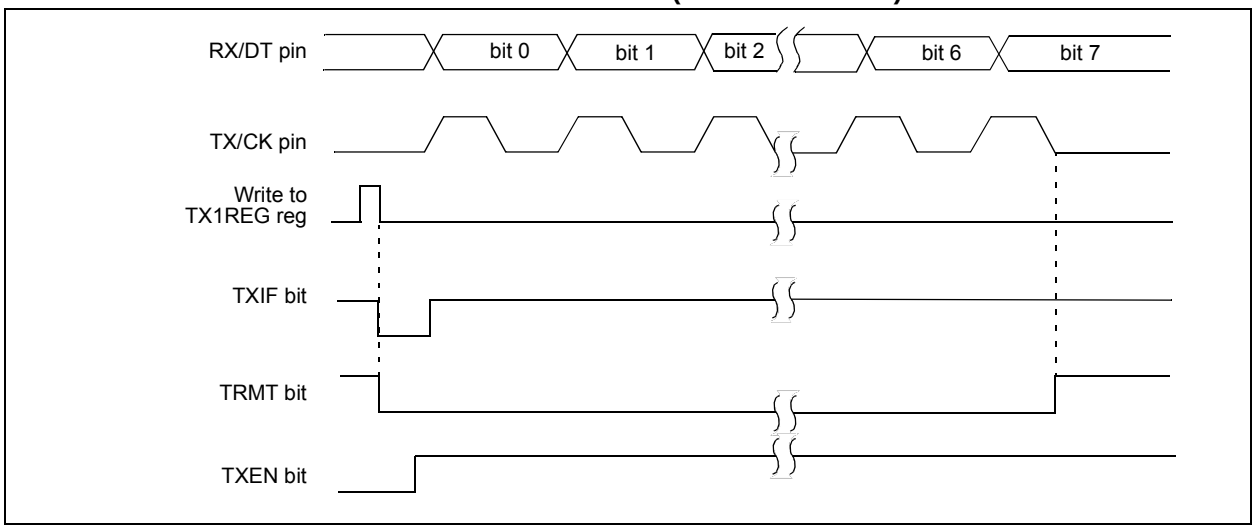
1. Initialize the SP1BRGH, SP1BRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 30.3 "EUSART1 Baud Rate Generator (BRG)"**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Disable Receive mode by clearing bits SREN and CREN.
4. Enable Transmit mode by setting the TXEN bit.
5. If 9-bit transmission is desired, set the TX9 bit.
6. If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
7. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
8. Start transmission by loading data to the TX1REG register.

FIGURE 30-10: SYNCHRONOUS TRANSMISSION



Note: Sync Master mode, SP1BRGL = 0, continuous transmission of two 8-bit words.

FIGURE 30-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



30.4.1.5 Synchronous Master Reception

Data is received at the RX/DT pin. The RX/DT pin output driver is automatically disabled when the EUSART1 is configured for synchronous master receive operation.

In Synchronous mode, reception is enabled by setting either the Single Receive Enable bit (SREN of the RC1STA register) or the Continuous Receive Enable bit (CREN of the RC1STA register).

When SREN is set and CREN is clear, only as many clock cycles are generated as there are data bits in a single character. The SREN bit is automatically cleared at the completion of one character. When CREN is set, clocks are continuously generated until CREN is cleared. If CREN is cleared in the middle of a character the CK clock stops immediately and the partial character is discarded. If SREN and CREN are both set, then SREN is cleared at the completion of the first character and CREN takes precedence.

To initiate reception, set either SREN or CREN. Data is sampled at the RX/DT pin on the trailing edge of the TX/CK clock pin and is shifted into the Receive Shift Register (RSR). When a complete character is received into the RSR, the RCIF bit is set and the character is automatically transferred to the two character receive FIFO. The Least Significant eight bits of the top character in the receive FIFO are available in RC1REG. The RCIF bit remains set as long as there are unread characters in the receive FIFO.

Note:	If the RX/DT function is on an analog pin, the corresponding ANSEL bit must be cleared for the receiver to function.

30.4.1.6 Slave Clock

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

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

30.4.1.7 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before RC1REG is read to access the FIFO. When this happens the OERR bit of the RC1STA register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the Overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear then the error is cleared by reading RC1REG. If the overrun occurred when the CREN bit is set then the Error condition is cleared by either clearing the CREN bit of the RC1STA register or by clearing the SPEN bit which resets the EUSART1.

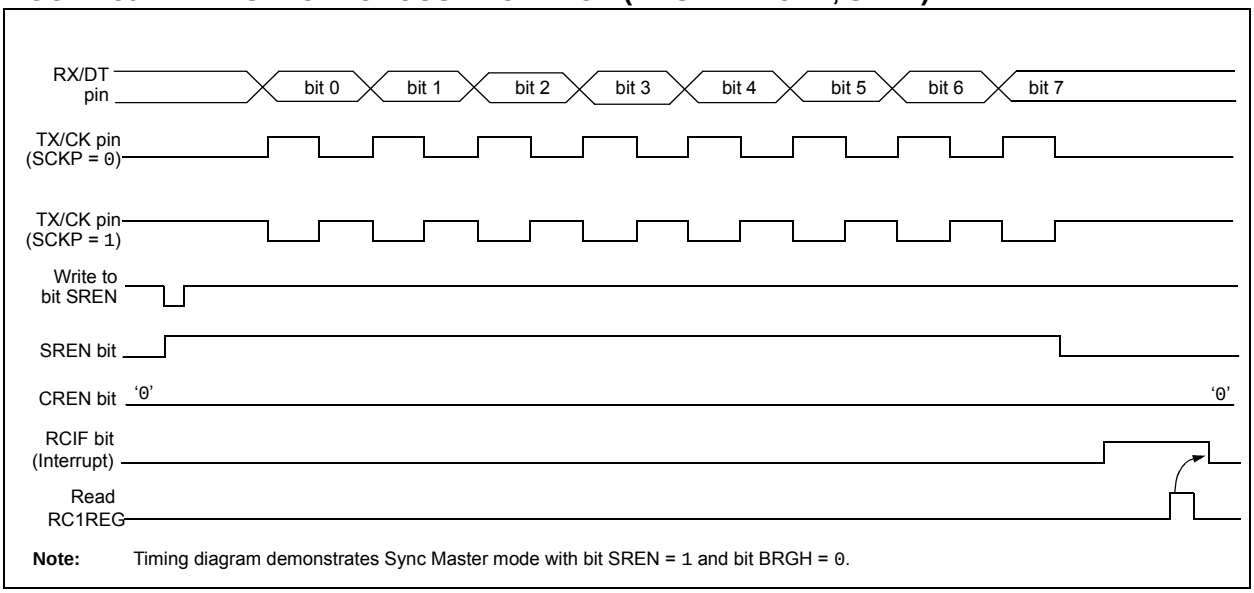
30.4.1.8 Receiving 9-bit Characters

The EUSART1 supports 9-bit character reception. When the RX9 bit of the RC1STA register is set the EUSART1 will shift nine bits into the RSR for each character received. The RX9D bit of the RC1STA register is the ninth, and Most Significant, data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RC1REG.

30.4.1.9 Synchronous Master Reception Set-up

1. Initialize the SP1BRGH, SP1BRGL register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Clear the ANSEL bit for the RX pin (if applicable).
3. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
4. Ensure bits CREN and SREN are clear.
5. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
6. If 9-bit reception is desired, set bit RX9.
7. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
8. Interrupt flag bit RCIF will be set when reception of a character is complete. An interrupt will be generated if the enable bit RCIE was set.
9. Read the RC1STA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
10. Read the 8-bit received data by reading the RC1REG register.
11. If an overrun error occurs, clear the error by either clearing the CREN bit of the RC1STA register or by clearing the SPEN bit which resets the EUSART1.

FIGURE 30-12: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



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

30.4.2 SYNCHRONOUS SLAVE MODE

The following bits are used to configure the EUSART1 for synchronous slave operation:

- SYNC = 1
- CSRC = 0
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TX1STA register configures the device for synchronous operation. Clearing the CSRC bit of the TX1STA register configures the device as a slave. Clearing the SREN and CREN bits of the RC1STA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RC1STA register enables the EUSART1.

30.4.2.1 EUSART1 Synchronous Slave Transmit

The operation of the Synchronous Master and Slave modes are identical (see **Section 30.4.1.3 "Synchronous Master Transmission"**), except in the case of the Sleep mode.

If two words are written to the TX1REG and then the SLEEP instruction is executed, the following will occur:

1. The first character will immediately transfer to the TSR register and transmit.
2. The second word will remain in the TX1REG register.
3. The TXIF bit will not be set.
4. After the first character has been shifted out of TSR, the TX1REG register will transfer the second character to the TSR and the TXIF bit will now be set.
5. If the PEIE and TXIE bits are set, the interrupt will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will call the Interrupt Service Routine.

30.4.2.2 Synchronous Slave Transmission Set-up

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Clear the ANSEL bit for the CK pin (if applicable).
3. Clear the CREN and SREN bits.
4. If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
5. If 9-bit transmission is desired, set the TX9 bit.
6. Enable transmission by setting the TXEN bit.
7. If 9-bit transmission is selected, insert the Most Significant bit into the TX9D bit.
8. Start transmission by writing the Least Significant eight bits to the TX1REG register.

30.4.2.3 EUSART1 Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (**Section 30.4.1.5 "Synchronous Master Reception"**), with the following exceptions:

- Sleep
- CREN bit is always set, therefore the receiver is never idle
- SREN bit, which is a "don't care" in Slave mode

A character may be received while in Sleep mode by setting the CREN bit prior to entering Sleep. Once the word is received, the RSR register will transfer the data to the RC1REG register. If the RCIE enable bit is set, the interrupt generated will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will branch to the interrupt vector.

30.4.2.4 Synchronous Slave Reception Set-up

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Clear the ANSEL bit for both the CK and DT pins (if applicable).
3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
4. If 9-bit reception is desired, set the RX9 bit.
5. Set the CREN bit to enable reception.
6. The RCIF bit will be set when reception is complete. An interrupt will be generated if the RCIE bit was set.
7. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RC1STA register.
8. Retrieve the eight Least Significant bits from the receive FIFO by reading the RC1REG register.
9. If an overrun error occurs, clear the error by either clearing the CREN bit of the RC1STA register or by clearing the SPEN bit which resets the EUSART1.

30.5 EUSART1 Operation During Sleep

The EUSART1 will remain active during Sleep only in the Synchronous Slave mode. All other modes require the system clock and therefore cannot generate the necessary signals to run the Transmit or Receive Shift registers during Sleep.

Synchronous Slave mode uses an externally generated clock to run the Transmit and Receive Shift registers.

30.5.1 SYNCHRONOUS RECEIVE DURING SLEEP

To receive during Sleep, all the following conditions must be met before entering Sleep mode:

- RC1STA and TX1STA Control registers must be configured for Synchronous Slave Reception (see **Section 30.4.2.4 "Synchronous Slave Reception Set-up"**).
- If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- The RCIF interrupt flag must be cleared by reading RC1REG to unload any pending characters in the receive buffer.

Upon entering Sleep mode, the device will be ready to accept data and clocks on the RX/DT and TX/CK pins, respectively. When the data word has been completely clocked in by the external device, the RCIF interrupt flag bit of the PIR1 register will be set. Thereby, waking the processor from Sleep.

Upon waking from Sleep, the instruction following the SLEEP instruction will be executed. If the Global Interrupt Enable (GIE) bit of the INTCON register is also set, then the Interrupt Service Routine at address 004h will be called.

30.5.2 SYNCHRONOUS TRANSMIT DURING SLEEP

To transmit during Sleep, all the following conditions must be met before entering Sleep mode:

- The RC1STA and TX1STA Control registers must be configured for synchronous slave transmission (see **Section 30.4.2.2 "Synchronous Slave Transmission Set-up"**).
- The TXIF interrupt flag must be cleared by writing the output data to the TX1REG, thereby filling the TSR and transmit buffer.
- If interrupts are desired, set the TXIE bit of the PIE1 register and the PEIE bit of the INTCON register.
- Interrupt enable bits TXIE of the PIE1 register and PEIE of the INTCON register must set.

Upon entering Sleep mode, the device will be ready to accept clocks on TX/CK pin and transmit data on the RX/DT pin. When the data word in the TSR has been completely clocked out by the external device, the pending byte in the TX1REG will transfer to the TSR and the TXIF flag will be set. Thereby, waking the processor from Sleep. At this point, the TX1REG is available to accept another character for transmission, which will clear the TXIF flag.

Upon waking from Sleep, the instruction following the SLEEP instruction will be executed. If the Global Interrupt Enable (GIE) bit is also set then the Interrupt Service Routine at address 0004h will be called.

30.6 Register Definitions: EUSART1 Control

REGISTER 30-1: TX1STA: TRANSMIT STATUS AND 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-1/1	R/W-0/0
CSRC	TX9	TXEN(1)	SYNC	SENDB	BRGH	TRMT	TX9D
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 **CSRC:** Clock Source Select bit
Asynchronous mode:
Unused in this mode – value ignored
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)
1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC:** EUSART1 Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit
Asynchronous mode:
1 = Send SYNCH BREAK on next transmission – Start bit, followed by 12 '0' bits, followed by Stop bit; cleared by hardware upon completion
0 = SYNCH BREAK transmission disabled or completed
Synchronous mode:
Unused in this mode – value ignored

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode – value ignored

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** Ninth bit of Transmit Data
Can be address/data bit or a parity bit.

Note 1: SREN/CREN overrides TXEN in Sync mode.

REGISTER 30-2: RC1STA: RECEIVE STATUS AND CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-x/x
SPEN(1)	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
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 **SPEN:** Serial Port Enable bit(1)
- 1 = Serial port enabled
- 0 = Serial port disabled (held in Reset)

bit 6 **RX9:** 9-bit Receive Enable bit
- 1 = Selects 9-bit reception
- 0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit

Asynchronous mode:

Unused in this mode – value ignored

Synchronous mode – Master:
- 1 = Enables single receive
- 0 = Disables single receive

This bit is cleared after reception is complete.

Synchronous mode – Slave

Unused in this mode – value ignored

bit 4 **CREN:** Continuous Receive Enable bit

Asynchronous mode:
- 1 = Enables continuous receive until enable bit CREN is cleared
- 0 = Disables continuous receive

Synchronous mode:
- 1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
- 0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit

Asynchronous mode 9-bit (RX9 = 1):
- 1 = Enables address detection – enable interrupt and load of the receive buffer when the ninth bit in the receive buffer is set
- 0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit

Asynchronous mode 8-bit (RX9 = 0):

Unused in this mode – value ignored

bit 2 **FERR:** Framing Error bit
- 1 = Framing error (can be updated by reading RC1REG register and receive next valid byte)
- 0 = No framing error

bit 1 **OERR:** Overrun Error bit
- 1 = Overrun error (can be cleared by clearing bit CREN)
- 0 = No overrun error

bit 0 **RX9D:** Ninth bit of Received Data

This can be address/data bit or a parity bit and must be calculated by user firmware.

Note 1: The EUSART1 module automatically changes the pin from tri-state to drive as needed. Configure the associated TRIS bits for TX/CK and RX/DT to 1.

REGISTER 30-3: BAUD1CON: BAUD RATE CONTROL REGISTER

R-0/0	R-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
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 **ABDOVF:** Auto-Baud Detect Overflow bit
Asynchronous mode:
1 = Auto-baud timer overflowed
0 = Auto-baud timer did not overflow
Synchronous mode:
Don't care

bit 6 **RCIDL:** Receive Idle Flag bit
Asynchronous mode:
1 = Receiver is Idle
0 = Start bit has been received and the receiver is receiving
Synchronous mode:
Don't care

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

bit 4 **SCKP:** Clock/Transmit Polarity Select bit
Asynchronous mode:
1 = Idle state for transmit (TX) is a low level
0 = Idle state for transmit (TX) is a high level
Synchronous mode:
1 = Idle state for clock (CK) is a high level
0 = Idle state for clock (CK) is a low level

bit 3 **BRG16:** 16-bit Baud Rate Generator bit
1 = 16-bit Baud Rate Generator is used
0 = 8-bit Baud Rate Generator is used

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

bit 1 **WUE:** Wake-up Enable bit
Asynchronous mode:
1 = EUSART will continue to sample the Rx pin – interrupt generated on falling edge; bit cleared in hardware on following rising edge.
0 = RX pin not monitored nor rising edge detected
Synchronous mode:
Unused in this mode – value ignored

bit 0 **ABDEN**: Auto-Baud Detect Enable bit
Asynchronous mode:
1 = Enable baud rate measurement on the next character – requires reception of a SYNCH field (55h);cleared in hardware upon completion
0 = Baud rate measurement disabled or completed
Synchronous mode:
Unused in this mode – value ignored

REGISTER 30-4: RC1REG(1): RECEIVE DATA REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
RC1REG<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 **RC1REG<7:0>:** Lower eight bits of the received data; read-only; see also RX9D (Register 30-2)

Note 1: RC1REG (including the ninth bit) is double buffered, and data is available while new data is being received.

REGISTER 30-5: TX1REG(1): TRANSMIT DATA REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TX1REG<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 **TX1REG<7:0>:** Lower eight bits of the received data; read-only; see also RX9D (Register 30-1)

Note 1: TX1REG (including the ninth bit) is double buffered, and can be written when previous data has started shifting.

REGISTER 30-6: SP1BRGL(1): BAUD RATE GENERATOR REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SP1BRG<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 **SP1BRG<7:0>:** Lower eight bits of the Baud Rate Generator

Note 1: Writing to SP1BRG resets the BRG counter.

REGISTER 30-7: SP1BRGH(1, 2): BAUD RATE GENERATOR HIGH REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SP1BRG<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 **SP1BRG<15:8>:** Upper eight bits of the Baud Rate Generator

Note 1: SP1BRGH value is ignored for all modes unless BAUD1CON<BRG16> is active.

2: Writing to SP1BRGH resets the BRG counter.

TABLE 30-2: SUMMARY OF REGISTERS ASSOCIATED WITH EUSART1

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	100
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	383
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	382
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	384
RC1REG	RC1REG<7:0>								385
TX1REG	TX1REG<7:0>								385
SP1BRGL	SP1BRG<7:0>								385
SP1BRGH	SP1BRG<15:8>								386
RXPPS	—	—	—	RXPPS<4:0>					161
TXPPS	—	—	—	TXPPS<4:0>					160
RxyPPS	—	—	—	RxyPPS<4:0>					161
CLCxSELy	—	—	—	LCxDyS<4:0>					227
MDSRC	—	—	—	—	MDMS<3:0>				270

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for the EUSART1 module.

Note 1: PIC16(L)F18345 only.

2: Unimplemented, read as '1'.

TABLE 30-3: BAUD RATE FORMULAS

Configuration Bits			BRG/EUSART1 Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	Fosc/[64 (n+1)]
0	0	1	8-bit/Asynchronous	Fosc/[16 (n+1)]
0	1	0	16-bit/Asynchronous	
0	1	1	16-bit/Asynchronous	Fosc/[4 (n+1)]
1	0	x	8-bit/Synchronous	
1	1	x	16-bit/Synchronous	

Legend: x = Don't care, n = value of SP1BRGH, SP1BRGL register pair.

TABLE 30-4: BAUD RATE FOR ASYNCHRONOUS MODES

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	1221	1.73	255	1200	0.00	239	1200	0.00	143
2400	2404	0.16	207	2404	0.16	129	2400	0.00	119	2400	0.00	71
9600	9615	0.16	51	9470	-1.36	32	9600	0.00	29	9600	0.00	17
10417	10417	0.00	47	10417	0.00	29	10286	-1.26	27	10165	-2.42	16
19.2k	19.23k	0.16	25	19.53k	1.73	15	19.20k	0.00	14	19.20k	0.00	8
57.6k	55.55k	-3.55	3	—	—	—	57.60k	0.00	7	57.60k	0.00	2
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	300	0.16	207	300	0.00	191	300	0.16	51
1200	1202	0.16	103	1202	0.16	51	1200	0.00	47	1202	0.16	12
2400	2404	0.16	51	2404	0.16	25	2400	0.00	23	—	—	—
9600	9615	0.16	12	—	—	—	9600	0.00	5	—	—	—
10417	10417	0.00	11	10417	0.00	5	—	—	—	—	—	—
19.2k	—	—	—	—	—	—	19.20k	0.00	2	—	—	—
57.6k	—	—	—	—	—	—	57.60k	0.00	0	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

TABLE 30-4: BAUD RATE FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 0											
	FOSC = 32.000 MHz			FOSC = 20.000 MHz			FOSC = 18.432 MHz			FOSC = 11.0592 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	—	—	—
9600	9615	0.16	207	9615	0.16	129	9600	0.00	119	9600	0.00	71
10417	10417	0.00	191	10417	0.00	119	10378	-0.37	110	10473	0.53	65
19.2k	19.23k	0.16	103	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35
57.6k	57.14k	-0.79	34	56.82k	-1.36	21	57.60k	0.00	19	57.60k	0.00	11
115.2k	117.64k	2.12	16	113.64k	-1.36	10	115.2k	0.00	9	115.2k	0.00	5

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 0											
	FOSC = 8.000 MHz			FOSC = 4.000 MHz			FOSC = 3.6864 MHz			FOSC = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	300	0.16	207
1200	—	—	—	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19231	0.16	25	19.23k	0.16	12	19.2k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 1											
	FOSC = 32.000 MHz			FOSC = 20.000 MHz			FOSC = 18.432 MHz			FOSC = 11.0592 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	300.0	0.00	6666	300.0	-0.01	4166	300.0	0.00	3839	300.0	0.00	2303
1200	1200	-0.02	3332	1200	-0.03	1041	1200	0.00	959	1200	0.00	575
2400	2401	-0.04	832	2399	-0.03	520	2400	0.00	479	2400	0.00	287
9600	9615	0.16	207	9615	0.16	129	9600	0.00	119	9600	0.00	71
10417	10417	0.00	191	10417	0.00	119	10378	-0.37	110	10473	0.53	65
19.2k	19.23k	0.16	103	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35
57.6k	57.14k	-0.79	34	56.818	-1.36	21	57.60k	0.00	19	57.60k	0.00	11
115.2k	117.6k	2.12	16	113.636	-1.36	10	115.2k	0.00	9	115.2k	0.00	5

TABLE 30-4: BAUD RATE FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 1											
	FOSC = 8.000 MHz			FOSC = 4.000 MHz			FOSC = 3.6864 MHz			FOSC = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	299.9	-0.02	1666	300.1	0.04	832	300.0	0.00	767	300.5	0.16	207
1200	1199	-0.08	416	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19.23k	0.16	25	19.23k	0.16	12	19.20k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	FOSC = 32.000 MHz			FOSC = 20.000 MHz			FOSC = 18.432 MHz			FOSC = 11.0592 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	300.0	0.00	26666	300.0	0.00	16665	300.0	0.00	15359	300.0	0.00	9215
1200	1200	0.00	6666	1200	-0.01	4166	1200	0.00	3839	1200	0.00	2303
2400	2400	0.01	3332	2400	0.02	2082	2400	0.00	1919	2400	0.00	1151
9600	9604	0.04	832	9597	-0.03	520	9600	0.00	479	9600	0.00	287
10417	10417	0.00	767	10417	0.00	479	10425	0.08	441	10433	0.16	264
19.2k	19.18k	-0.08	416	19.23k	0.16	259	19.20k	0.00	239	19.20k	0.00	143
57.6k	57.55k	-0.08	138	57.47k	-0.22	86	57.60k	0.00	79	57.60k	0.00	47
115.2k	115.9k	0.64	68	116.3k	0.94	42	115.2k	0.00	39	115.2k	0.00	23

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	FOSC = 8.000 MHz			FOSC = 4.000 MHz			FOSC = 3.6864 MHz			FOSC = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	300.0	0.00	6666	300.0	0.01	3332	300.0	0.00	3071	300.1	0.04	832
1200	1200	-0.02	1666	1200	0.04	832	1200	0.00	767	1202	0.16	207
2400	2401	0.04	832	2398	0.08	416	2400	0.00	383	2404	0.16	103
9600	9615	0.16	207	9615	0.16	103	9600	0.00	95	9615	0.16	25
10417	10417	0	191	10417	0.00	95	10473	0.53	87	10417	0.00	23
19.2k	19.23k	0.16	103	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	12
57.6k	57.14k	-0.79	34	58.82k	2.12	16	57.60k	0.00	15	—	—	—
115.2k	117.6k	2.12	16	111.1k	-3.55	8	115.2k	0.00	7	—	—	—

31.0 REFERENCE CLOCK OUTPUT MODULE

The Reference Clock Output module provides the ability to send a clock signal to the clock reference output pin (CLKR). The Reference Clock Output can also be used as a signal for other peripherals, such as the Data Signal Modulator (DSM).

The Reference Clock Output module has the following features:

- System clock is the module source clock
- Programmable clock divider
- Selectable duty cycle

31.1 Clock Source

The Reference Clock Output module uses the system clock (FOSC) as the clock source. Any device clock switching will be reflected in the clock output.

31.1.1 CLOCK SYNCHRONIZATION

Once the reference clock enable (CLKREN) is set, the module is ensured to be glitch-free at start-up.

When the Reference Clock Output is disabled, the output signal will be disabled immediately.

Clock dividers and clock duty cycles can be changed while the module is enabled, but glitches may occur on the output. To avoid possible glitches, clock dividers and clock duty cycles should be changed only when the CLKREN is clear.

31.2 Programmable Clock Divider

The module takes the system clock input and divides it based on the value of the CLKRDIV<2:0> bits of the CLKRCON register (Register 31-1).

The following configurations can be made based on the CLKRDIV<2:0> bits:

- Base FOSC value
- FOSC divided by 2
- FOSC divided by 4
- FOSC divided by 8
- FOSC divided by 16
- FOSC divided by 32
- FOSC divided by 64
- FOSC divided by 128

The clock divider values can be changed while the module is enabled; however, in order to prevent glitches on the output, the CLKRDIV<2:0> bits should only be changed when the module is disabled (CLKREN = 0).

31.3 Selectable Duty Cycle

The CLKRDC<1:0> bits of the CLKRCON register can be used to modify the duty cycle of the output clock. A duty cycle of 25%, 50%, or 75% can be selected for all clock rates, with the exception of the undivided base FOSC value.

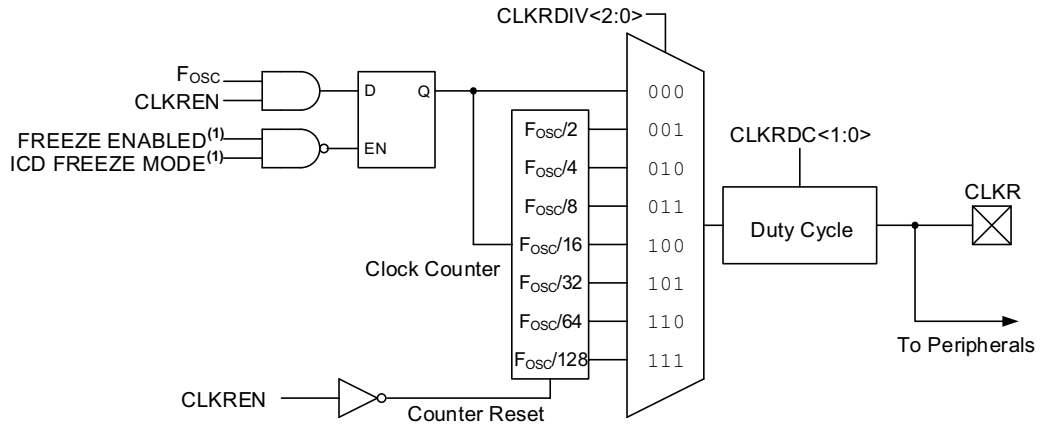
The duty cycle can be changed while the module is enabled; however, in order to prevent glitches on the output, the CLKRDC<1:0> bits should only be changed when the module is disabled (CLKREN = 0).

Note:	The CLKRDC1 bit is reset to '1'. This makes the default duty cycle 50% and not 0%.

31.4 Operation in Sleep Mode

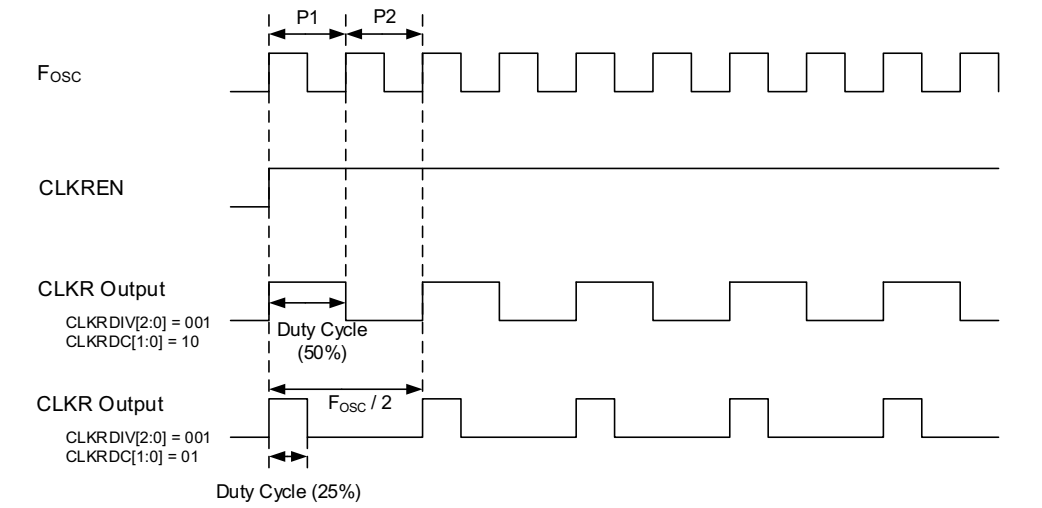
The Reference Clock Output module clock is based on the system clock. When the device goes to Sleep, the module outputs will remain in their current state. This will have a direct effect on peripherals using the Reference Clock Output as an input signal.

FIGURE 31-1: CLOCK REFERENCE BLOCK DIAGRAM



Note 1: Freeze is used in Debug Mode only; otherwise read as '0'

FIGURE 31-2: CLOCK REFERENCE TIMING



REGISTER 31-1: CLKRCON: REFERENCE CLOCK CONTROL REGISTER

R/W-0/0	U-0	U-0	R/W-1/1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CLKREN	—	—	CLKRDC<1:0>		CLKRDIV<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 **CLKREN:** Reference Clock Module Enable bit
1 = Reference clock module enabled
0 = Reference clock module is disabled

bit 6-5 **Unimplemented:** Read as '0'.

bit 4-3 **CLKRDC<1:0>:** Reference Clock Duty Cycle bits (1)
11 = Clock outputs duty cycle of 75%
10 = Clock outputs duty cycle of 50%
01 = Clock outputs duty cycle of 25%
00 = Clock outputs duty cycle of 0%

bit 2-0 **CLKRDIV<2:0>:** Reference Clock Divider bits
111 = FOSC divided by 128
110 = FOSC divided by 64
101 = FOSC divided by 32
100 = FOSC divided by 16
011 = FOSC divided by 8
010 = FOSC divided by 4
001 = FOSC divided by 2
000 = FOSC

Note 1: Bits are valid for Reference Clock divider values of two or larger, the base clock cannot be further divided.

TABLE 31-1: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK REFERENCE OUTPUT

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	141
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
CLKRCON	CLKREN	—	—	CLKRDC<1:0>		CLKRDIV<2:0>			225
CLCxSELy	—	—	LCxDyS<5:0>						227
MDCARH	—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>				271
MDCARL	—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				272
RxyPPS	—	—	—	RxyPPS<4:0>					161

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the CLKR module.
Note 1: PIC16(L)F18345 only.
2: Unimplemented, read as '1'.

32.0 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
- $\overline{MCLR}/V_{PP}$
- V_{DD}
- V_{SS}

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 **"***PIC16(L)F183XX Memory Programming Specification***"** (DS40001738).

32.1 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 $\overline{MCLR}/V_{PP}$ to V_{IHH}.

32.2 Low-Voltage Programming Entry Mode

The Low-Voltage Programming Entry mode allows the PIC® Flash MCUs to be programmed using V_{DD} only, without high voltage. When the LVP bit of Configuration Words 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. $\overline{MCLR}$ is brought to V_{IL}.
2. A 32-bit key sequence is presented on ICSPDAT, while clocking ICSPCLK.

Once the key sequence is complete, $\overline{MCLR}$ must be held at V_{IL} for as long as Program/Verify mode is to be maintained.

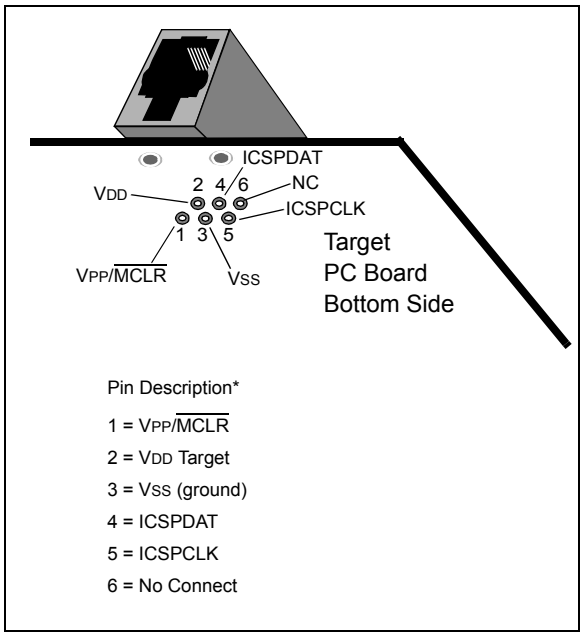
If low-voltage programming is enabled (LVP = 1), the $\overline{MCLR}$ Reset function is automatically enabled and cannot be disabled. See **Section 5.4 "MCLR"** for more information.

The LVP bit can only be reprogrammed to '0' by using the High-Voltage Programming mode.

32.3 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 32-1.

FIGURE 32-1: ICD RJ-11 STYLE CONNECTOR INTERFACE

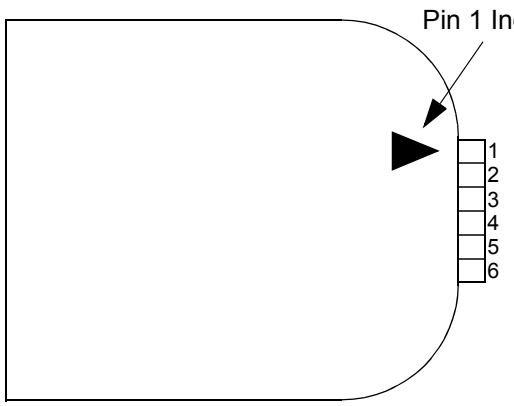


Another connector often found in use with the PICkit™ programmers is a standard 6-pin header with 0.1 inch spacing. Refer to Figure 32-2.

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 32-3 for more information.

FIGURE 32-2: PICkit™ PROGRAMMER STYLE CONNECTOR INTERFACE



Pin Description*

1 = VPP/$\overline{\text{MCLR}}$

2 = VDD Target

3 = VSS (ground)

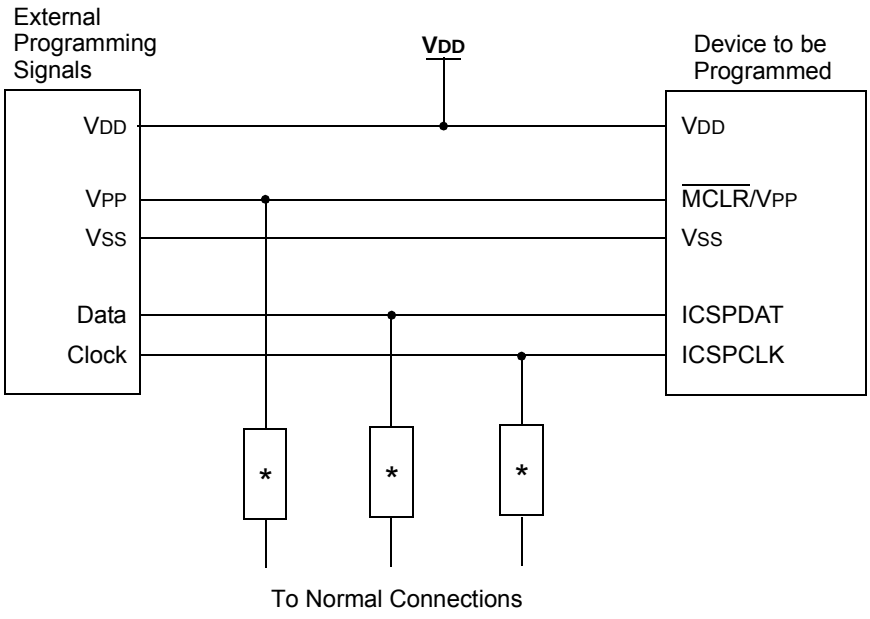
4 = ICSPDAT

5 = ICSPCLK

6 = No Connect

* The 6-pin header (0.100" spacing) accepts 0.025" square pins.

FIGURE 32-3: TYPICAL CONNECTION FOR ICSP™ PROGRAMMING



* Isolation devices (as required).

33.0 INSTRUCTION SET SUMMARY

Each instruction is a 14-bit word containing the operation code (opcode) and all required operands. The opcodes are broken into three broad categories.

- Byte Oriented
- Bit Oriented
- Literal and Control

The literal and control category contains the most varied instruction word format.

Table 33-3 lists the instructions recognized by the MPASM™ assembler.

All instructions are executed within a single instruction cycle, with the following exceptions, which may take two or three cycles:

- Subroutine takes two cycles (CALL, CALLW)
- Returns from interrupts or subroutines take two cycles (RETURN, RETLW, RETFIE)
- Program branching takes two cycles (GOTO, BRA, BRW, BTFSS, BTFSC, DECFSZ, INCSFZ)
- One additional instruction cycle will be used when any instruction references an indirect file register and the file select register is pointing to program memory.

One instruction cycle consists of 4 oscillator cycles; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution rate of 1 MHz.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

33.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) 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.

TABLE 33-1: 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.
n	FSR or INDF number. (0-1)
mm	Pre-post increment-decrement mode selection

TABLE 33-2: ABBREVIATION DESCRIPTIONS

Field	Description
PC	Program Counter
$\overline{TO}$	Time-Out bit
C	Carry bit
DC	Digit Carry bit
Z	Zero bit
$\overline{PD}$	Power-Down bit

FIGURE 33-1: GENERAL FORMAT FOR INSTRUCTIONS

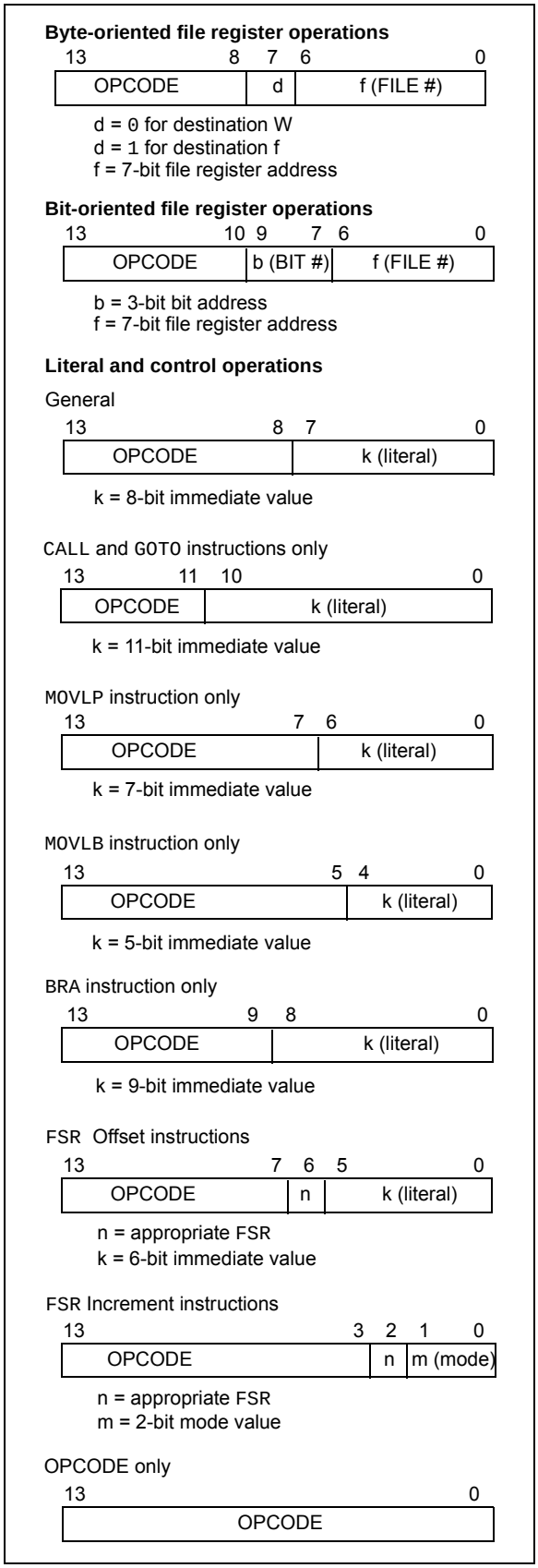


TABLE 33-3: PIC16(L)F18325/18345 INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	**2**
ADDWFC	f, d	Add with Carry W and f	1	11	1101	dfff	ffff	C, DC, Z	**2**
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	**2**
ASRF	f, d	Arithmetic Right Shift	1	11	0111	dfff	ffff	C, Z	**2**
LSLF	f, d	Logical Left Shift	1	11	0101	dfff	ffff	C, Z	**2**
LSRF	f, d	Logical Right Shift	1	11	0110	dfff	ffff	C, Z	**2**
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	**2**
CLRW	–	Clear W	1	00	0001	0000	00xx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	**2**
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	**2**
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	**2**
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	**2**
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	**2**
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		**2**
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	**2**
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	**2**
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	**2**
SUBWFB	f, d	Subtract with Borrow W from f	1	11	1011	dfff	ffff	C, DC, Z	**2**
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		**2**
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	**2**
BYTE ORIENTED SKIP OPERATIONS									
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		**1, 2**
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		**1, 2**
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		**2**
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		**2**
BIT-ORIENTED SKIP OPERATIONS									
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		**1, 2**
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		**1, 2**
LITERAL OPERATIONS									
ADDLW	k	Add literal and W	1	11	1110	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	Move literal to BSR	1	00	0000	001k	kkkk		
MOVLP	k	Move literal to PCLATH	1	11	0001	1kkk	kkkk		
MOVLW	k	Move literal to W	1	11	0000	kkkk	kkkk		
SUBLW	k	Subtract W from literal	1	11	1100	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	
CONTROL OPERATIONS									
BRA	k	Relative Branch	2	11	001k	kkkk	kkkk		
BRW	–	Relative Branch with W	2	00	0000	0000	1011		
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CALLW	–	Call Subroutine with W	2	00	0000	0000	1010		
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
RETFIE	k	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	0100	kkkk	kkkk		
RETURN	–	Return from Subroutine	2	00	0000	0000	1000		

Note 1: 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.

2: If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

3: See table in the MOVIW and MOVWI instruction descriptions.

TABLE 33-3: PIC16(L)F18325/18345 INSTRUCTION SET (CONTINUED)

Mnemonic, Operands		Description	Cycles	14-bit Opcode MSb LSb	Status Affected	Notes
INHERENT OPERATIONS						
CLRWDT	–	Clear Watchdog Timer	1	00 0000 0110 0100	$\overline{TO}$, $\overline{PD}$	
NOP	–	No Operation	1	00 0000 0000 0000		
OPTION	–	Load OPTION_REG register with W	1	00 0000 0110 0010		
RESET	–	Software device Reset	1	00 0000 0000 0001		
SLEEP	–	Go into Standby mode	1	00 0000 0110 0011	$\overline{TO}$, $\overline{PD}$	
TRIS	f	Load TRIS register with W	1	00 0000 0110 0fff		
C-COMPILER OPTIMIZED						
ADDFSR	n, k	Add Literal k to FSRn	1	11 0001 0nkk kkkk		
MOVIW	n mm	Move Indirect FSRn to W with pre/post inc/dec modifier, mm	1	00 0000 0001 0nmm	Z	**2, 3**
	k[n]	Move INDFn to W, Indexed Indirect.	1	11 1111 0nkk kkkk	Z	**2**
MOVWI	n mm	Move W to Indirect FSRn with pre/post inc/dec modifier, mm	1	00 0000 0001 1nmm		**2, 3**
	k[n]	Move W to INDFn, Indexed Indirect.	1	11 1111 1nkk kkkk		**2**

Note 1: 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.

2: If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

3: See table in the MOVIW and MOVWI instruction descriptions.

33.2 Instruction Descriptions

ADDFSR — Add Literal to FSRn

Syntax:	[*label*] ADDFSR FSRn, k
Operands:	-32 ≤ k ≤ 31 n ∈ [0, 1]
Operation:	FSR(n) + k → FSR(n)
Status Affected:	None
Description:	The signed 6-bit literal 'k' is added to the contents of the FSRnH:FSRnL register pair. FSRn is limited to the range 0000h-FFFFh. Moving beyond these bounds will cause the FSR to wrap-around.

ADDLW — Add literal and W

Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the 8-bit literal 'k' and the result is placed in the W register.

ADDWF — Add W and f

Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [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'.

ADDWFC — ADD W and CARRY bit to f

Syntax:	[*label*] ADDWFC f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) + (C) → dest
Status Affected:	C, DC, Z
Description:	Add W, the Carry flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'.

ANDLW — AND literal with W

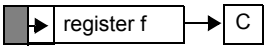
Syntax:	[*label*] ANDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND'ed with the 8-bit literal 'k'. The result is placed in the W register.

ANDWF — AND W with f

Syntax:	[*label*] ANDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [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'.

ASRF — Arithmetic Right Shift

Syntax:	[*label*] ASRF f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f<7>)→ dest<7> (f<7:1>) → dest<6:0>, (f<0>) → C,
Status Affected:	C, Z
Description:	The contents of register 'f' are shifted one bit to the right through the Carry flag. The MSb remains unchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.



BCF **Bit Clear f**

Syntax:	[*label*] BCF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	0 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

BRA **Relative Branch**

Syntax:	[*label*] BRA label [*label*] BRA $+k
Operands:	-256 ≤ label - PC + 1 ≤ 255 -256 ≤ k ≤ 255
Operation:	(PC) + 1 + k → PC
Status Affected:	None
Description:	Add the signed 9-bit literal 'k' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + k. This instruction is a 2-cycle instruction. This branch has a limited range.

BRW **Relative Branch with W**

Syntax:	[*label*] BRW
Operands:	None
Operation:	(PC) + (W) → PC
Status Affected:	None
Description:	Add the contents of W (unsigned) to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + (W). This instruction is a 2-cycle instruction.

BSF **Bit Set f**

Syntax:	[*label*] BSF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 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 ≤ f ≤ 127 0 ≤ b ≤ 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 2-cycle instruction.

BTFSS **Bit Test f, Skip if Set**

Syntax:	[*label*] BTFSS f,b
Operands:	0 ≤ f ≤ 127 0 ≤ 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 2-cycle instruction.

CALL **Call Subroutine**

Syntax:	[*label*] CALL k
Operands:	$0 \le k \le 2047$
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<6:3>) → PC<14:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The 11-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a 2-cycle instruction.

CALLW **Subroutine Call With W**

Syntax:	[*label*] CALLW
Operands:	None
Operation:	(PC) +1 → TOS, (W) → PC<7:0>, (PCLATH<6:0>) → PC<14:8>
Status Affected:	None
Description:	Subroutine call with W. First, the return address (PC + 1) is pushed onto the return stack. Then, the contents of W is loaded into PC<7:0>, and the contents of PCLATH into PC<14:8>. CALLW is a 2-cycle instruction.

CLRF **Clear f**

Syntax:	[*label*] CLRF f
Operands:	$0 \le f \le 127$
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:	$0 \le f \le 127$ $d \in [0,1]$
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:	$0 \le f \le 127$ $d \in [0,1]$
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 ≤ f ≤ 127 d ∈ [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 2-cycle instruction.

GOTO **Unconditional Branch**

Syntax:	[*label*] GOTO k
Operands:	0 ≤ k ≤ 2047
Operation:	k → PC<10:0> PCLATH<6:3> → PC<14:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 11-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a 2-cycle instruction.

INCF **Increment f**

Syntax:	[*label*] INCF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [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 ≤ f ≤ 127 d ∈ [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 2-cycle instruction.

IORLW **Inclusive OR literal with W**

Syntax:	[*label*] IORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are OR'ed with the 8-bit literal 'k'. The result is placed in the W register.

IORWF **Inclusive OR W with f**

Syntax:	[*label*] IORWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [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'.

LSLF **Logical Left Shift**

Syntax: [*label*] LSLF f {,d}

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f<7>) → C
(f<6:0>) → dest<7:1>
0 → dest<0>

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the left through the Carry flag. A '0' is shifted into the LSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.

LSRF **Logical Right Shift**

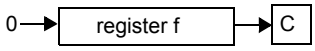
Syntax: [*label*] LSRF f {,d}

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: 0 → dest<7>
(f<7:1>) → dest<6:0>,
(f<0>) → C,

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the right through the Carry flag. A '0' is shifted into the MSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.



MOVF **Move f**

Syntax: [*label*] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

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

MOVIW **Move INDFn to W**

Syntax: [*label*] MOVIW ++FSRn
[*label*] MOVIW --FSRn
[*label*] MOVIW FSRn++
[*label*] MOVIW FSRn--
[*label*] MOVIW k[FSRn]

Operands: n ∈ [0,1]
mm ∈ [00,01, 10, 11]
-32 ≤ k ≤ 31

Operation: INDFn → W
Effective address is determined by
- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:
- FSR + 1 (all increments)
- FSR - 1 (all decrements)
- Unchanged

Status Affected: Z

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h - FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

MOVLB **Move literal to BSR**

Syntax: [*label*] MOVLB k

Operands: 0 ≤ k ≤ 31

Operation: k → BSR

Status Affected: None

Description: The 5-bit literal 'k' is loaded into the Bank Select Register (BSR).

MOVLP **Move literal to PCLATH**

Syntax: [*label*] MOVLP k

Operands: 0 ≤ k ≤ 127

Operation: k → PCLATH

Status Affected: None

Description: The 7-bit literal 'k' is loaded into the PCLATH register.

MOVLW **Move literal to W**

Syntax: [*label*] MOVLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W)

Status Affected: None

Description: The 8-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

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:
```
MOVWF   OPTION_REG
```
Before Instruction
OPTION_REG = 0xFF
W = 0x4F
After Instruction
OPTION_REG = 0x4F
W = 0x4F

MOVWI **Move W to INDFn**

Syntax: [*label*] MOVWI ++FSRn
[*label*] MOVWI --FSRn
[*label*] MOVWI FSRn++
[*label*] MOVWI FSRn--
[*label*] MOVWI k[FSRn]

Operands: $n \in [0,1]$
$mm \in [00,01, 10, 11]$
$-32 \leq k \leq 31$

Operation: $W \rightarrow INDFn$
Effective address is determined by
- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:
- FSR + 1 (all increments)
- FSR - 1 (all decrements)

Unchanged

Status Affected: None

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h-FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

The increment/decrement operation on FSRn WILL NOT affect any Status bits.

NOP **No Operation**

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Description: No operation.

Words: 1

Cycles: 1

Example:
```
NOP
```

OPTION **Load OPTION_REG Register with W**

Syntax: [*label*] OPTION

Operands: None

Operation: (W) → OPTION_REG

Status Affected: None

Description: Move data from W register to OPTION_REG register.

Words: 1

Cycles: 1

Example:
```
OPTION
```
Before Instruction
OPTION_REG = 0xFF
W = 0x4F
After Instruction
OPTION_REG = 0x4F
W = 0x4F

RESET **Software Reset**

Syntax: [*label*] RESET

Operands: None

Operation: Execute a device Reset. Resets the $\overline{RI}$ flag of the PCON register.

Status Affected: None

Description: This instruction provides a way to execute a hardware Reset by software.

RETFIE **Return from Interrupt**

Syntax: [*label*] RETFIE k

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 2-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 ≤ k ≤ 255

Operation: k → (W);
TOS → PC

Status Affected: None

Description: The W register is loaded with the 8-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.

Words: 1

Cycles: 2

Example:

```
CALL TABLE;W contains table
    ;offset value
•   ;W now has table value
•
•
ADDWF PC  ;W = offset
RETLW k1  ;Begin table
RETLW k2  ;
•
•
•
RETLW kn  ; End of table
```

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 2-cycle instruction.

RLF **Rotate Left f through Carry**

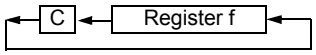
Syntax: [*label*] RLF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [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

Example:

```
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**

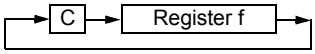
Syntax: [*label*] RRF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [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 → WDT,
0 → WDT prescaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

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.
See **Section 8.2 "Sleep Mode"** for more information.

SUBLW **Subtract W from literal**

Syntax: [*label*] SUBLW k

Operands: 0 ≤ k ≤ 255

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Description: The W register is subtracted (2's complement method) from the 8-bit literal 'k'. The result is placed in the W register.

C = 0	W > k
C = 1	W ≤ k
DC = 0	W<3:0> > k<3:0>
DC = 1	W<3:0> ≤ k<3:0>

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) - (W) → (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>

SUBWFB **Subtract W from f with Borrow**

Syntax: SUBWFB f {,d}

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

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

Status Affected: C, DC, Z

Description: Subtract W and the BORROW flag (CARRY) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

SWAPF **Swap Nibbles in f**

Syntax:	[*label*] SWAPF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f<3:0>) → (destination<7:4>), (f<7:4>) → (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'.

TRIS **Load TRIS Register with W**

Syntax:	[*label*] TRIS f
Operands:	$5 \le f \le 7$
Operation:	(W) → TRIS register 'f'
Status Affected:	None
Description:	Move data from W register to TRIS register. When 'f' = 5, TRISA is loaded. When 'f' = 6, TRISB is loaded. When 'f' = 7, TRISC is loaded.

XORLW **Exclusive OR literal with W**

Syntax:	[*label*] XORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .XOR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are XOR'ed with the 8-bit literal 'k'. The result is placed in the W register.

XORWF **Exclusive OR W with f**

Syntax:	[*label*] XORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .XOR. (f) → (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'.

34.0 ELECTRICAL SPECIFICATIONS

34.1 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 V_{SS}
- on V_{DD} pin
  - PIC16F18325/18345 ..... -0.3V to +6.5V
  - PIC16LF18325/18345 ..... -0.3V to +4.0V
- on $\overline{MCLR}$ pin ..... -0.3V to +9.0V
- on all other pins ..... -0.3V to (V_{DD} + 0.3V)

Maximum current
- on V_{SS} pin(1)
  - -40°C ≤ T_A ≤ +85°C ..... 250 mA
  - +85°C < T_A ≤ +125°C ..... 85 mA
- on V_{DD} pin(1)
  - -40°C ≤ T_A ≤ +85°C ..... 250 mA
  - +85°C < T_A ≤ +125°C ..... 85 mA
- on any I/O pin ..... ±50 mA

Clamp current, I_K (V_{PIN} < 0 or V_{PIN} > V_{DD}) ..... ±20 mA
Total power dissipation(2) ..... 800 mW

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 Table 34-3 to calculate device specifications.

2: Power dissipation is calculated as follows:
$P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$.

† 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.

34.2 Standard Operating Conditions

The standard operating conditions for any device are defined as:

Operating Voltage: $V_{DDMIN} \leq V_{DD} \leq V_{DDMAX}$
Operating Temperature: $T_{A\_MIN} \leq T_A \leq T_{A\_MAX}$

V_{DD} — Operating Supply Voltage(1)

PIC16LF18325/18345

V_{DDMIN} (Fosc ≤ 16 MHz) ..... +1.8V
V_{DDMIN} (Fosc ≤ 32 MHz) ..... +2.5V
V_{DDMAX} ..... +3.6V

PIC16F18325/18345

V_{DDMIN} (Fosc ≤ 16 MHz) ..... +2.3V
V_{DDMIN} (Fosc ≤ 32 MHz) ..... +2.5V
V_{DDMAX} ..... +5.5V

T_A — Operating Ambient Temperature Range

Industrial Temperature

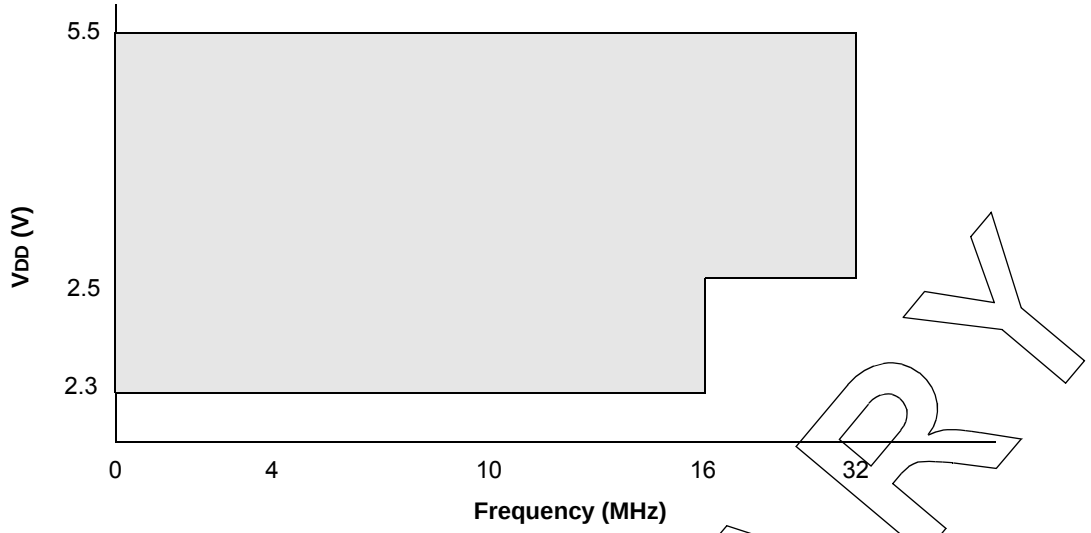
T_{A_MIN} ..... -40°C
T_{A_MAX} ..... +85°C

Extended Temperature

T_{A_MIN} ..... -40°C
T_{A_MAX} ..... +125°C

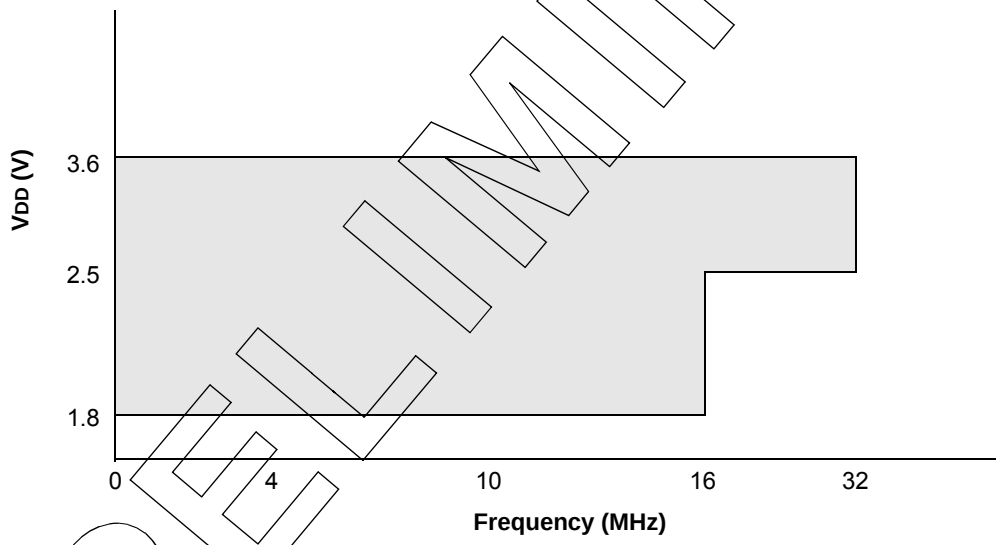
Note 1: See Parameter D002, DC Characteristics: Supply Voltage.

FIGURE 34-1: VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C, PIC16F18325/18345 ONLY



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.
2: Refer to Table 34-7 for each Oscillator mode's supported frequencies.

FIGURE 34-2: VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C, PIC16LF18325/18345 ONLY



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.
2: Refer to Table 34-7 for each Oscillator mode's supported frequencies.

34.3 DC Characteristics

TABLE 34-1: SUPPLY VOLTAGE

PIC16LF18325/18345			Standard Operating Conditions (unless otherwise stated)				
PIC16F18325/18345			Standard Operating Conditions (unless otherwise stated)				
Param. No.	Sym.	Characteristic	Min.	Typ.†	Max.	Units	Conditions
Supply Voltage							
D002	VDD		1.8 2.5	— —	3.6 3.6	V V	FOSC ≤ 16 MHz FOSC > 16 MHz
D002	VDD		2.3 2.5	— —	5.5 5.5	V V	FOSC ≤ 16 MHz: FOSC > 16 MHz
RAM Data Retention(1)							
D003	VDR		1.5	—	—	V	Device in Sleep mode
D003	VDR		1.7	—	—	V	Device in Sleep mode
Power-on Reset Release Voltage(2)							
D004	VPOR		—	1.6	—	V	BOR and LPBOR disabled(3)
D004	VPOR		—	1.6	—	V	BOR and LPBOR disabled(3)
Power-on Reset ReARM Voltage(2)							
D005	VPORR		—	0.8	—	V	BOR and LPBOR disabled(3)
D005	VPORR		—	1.5	—	V	BOR and LPBOR disabled(3)
VDD Rise Rate to ensure Internal Power-on Reset Signal(2)							
D006	SVDD		0.05	—	—	V/ms	BOR and LPBOR disabled(3)
D006	SVDD		0.05	—	—	V/ms	BOR and LPBOR disabled(3)

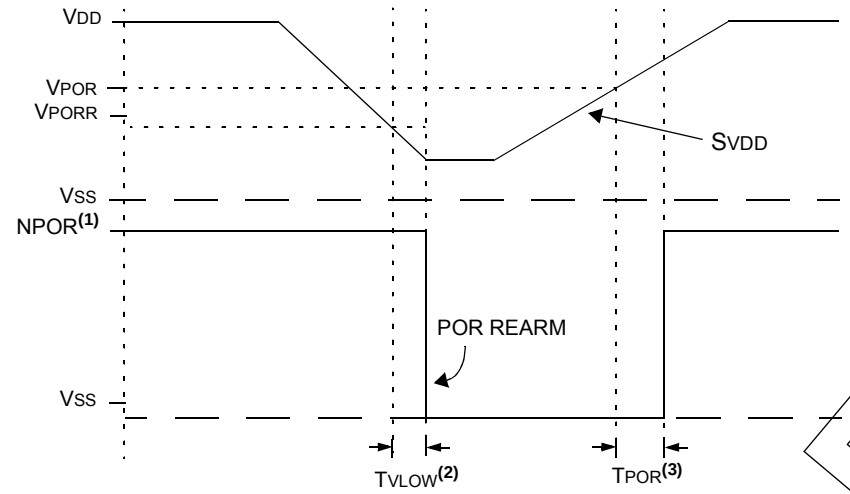
† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in Sleep mode or during a device Reset, without losing RAM data.

2: See Figure 34-3.

3: Please see Table 34-11 for BOR and LPBOR trip point information.

FIGURE 34-3: 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 34-2: SUPPLY CURRENT (I_{DD})(1,2)

PIC16LF18325/18345	Standard Operating Conditions (unless otherwise stated)							
PIC16F18325/18345	Standard Operating Conditions (unless otherwise stated)							
Param. No.	Symbol	Device Characteristics	Min.	Typ.†	Max.	Units	Conditions	
							V_{DD}	Note
D100	IDD_{XT4}	XT = 4 MHz	—	321	455	uA	3.0V	
D100	IDD_{XT4}	XT = 4 MHz	—	332	479	uA	3.0V	
D101	IDD_{HFO16}	HFINTOSC = 16 MHz	—	1.3	1.8	mA	3.0V	
D101	IDD_{HFO16}	HFINTOSC = 16 MHz	—	1.4	1.9	mA	3.0V	
D102	IDD_{HFOPLL}	HFINTOSC = 32 MHz	—	2.2	2.8	mA	3.0V	
D102	IDD_{HFOPLL}	HFINTOSC = 32 MHz	—	2.3	2.9	mA	3.0V	
D103	$IDD_{HSPLL32}$	HS+PLL = 32 MHz	—	2.2	2.8	mA	3.0V	
D103	$IDD_{HSPLL32}$	HS+PLL = 32 MHz	—	2.3	2.9	mA	3.0V	
D104	IDD_{IDLE}	IDLE Mode, HFINTOSC = 16 MHz	—	804	1283	uA	3.0V	
D104	IDD_{IDLE}	IDLE Mode, HFINTOSC = 16 MHz	—	816	1284	uA	3.0V	
D105	IDD_{DOZE}(3)	DOZE mode, HFINTOSC = 16 MHz, DOZE Ratio = 16	—	863	—	uA	3.0V	
D105	IDD_{DOZE}(3)	DOZE mode, HFINTOSC = 16 MHz, DOZE Ratio = 16	—	875	—	uA	3.0V	

† 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 test conditions for all I_{DD} measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD}; $\overline{MCLR}$ = V_{DD}; 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.

3: IDD_{DOZE} = [IDD_{IDLE}*(N-1)/N] + IDD_{HFO}16/N where N = DOZE Ration (see Register 8-2).

TABLE 34-3: POWER-DOWN CURRENTS (I_{PD})(1,2,3)

PIC16LF18325/18345			Standard Operating Conditions (unless otherwise stated)						
PIC16F18325/18345			Standard Operating Conditions (unless otherwise stated) VREGPM = 1						
Param. No.	Symbol	Device Characteristics	Min.	Typ.†	Max. +85°C	Max. +125°C	Units	Conditions	
								V_{DD}	Note
D200	I_{PD}	I_{PD} Base	—	0.05	2	9	µA	3.0V	
D200	I_{PD}	I_{PD} Base	—	0.8	4	12	µA	3.0V	
			—	13	22	27	µA	3.0V	VREGPM = 0
D201	I_{PD_WDT}	Low-Frequency Internal Oscillator/WDT	—	0.8	5	13	µA	3.0V	
D201	I_{PD_WDT}	Low-Frequency Internal Oscillator/WDT	—	0.9	5	13	µA	3.0V	
D202	I_{PD_SOSC}	Secondary Oscillator (SOSC)	—	0.6	5	13	µA	3.0V	
D202	I_{PD_SOSC}	Secondary Oscillator (SOSC)	—	0.8	9	15	µA	3.0V	
D203	I_{PD_FVR}	FVR	—	40	47	47	µA	3.0V	
D203	I_{PD_FVR}	FVR	—	33	44	44	µA	3.0V	
D204	I_{PD_BOR}	Brown-out Reset (BOR)	—	12	17	19	µA	3.0V	
D204	I_{PD_BOR}	Brown-out Reset (BOR)	—	12	18	20	µA	3.0V	
D205	I_{PD_LPBOR}	Low Power Brown-out Reset (LPBOR)	—	3	5	13	µA	3.0V	
D205	I_{PD_LPBOR}	Low Power Brown-out Reset (LPBOR)	—	4	5	13	µA	3.0V	
D207	I_{PD_ADCA}	ADC - Active	—	0.9	5	13	µA	3.0V	ADC is converting(4)
D207	I_{PD_ADCA}	ADC - Active	—	0.9	5	13	µA	3.0V	ADC is converting(4)
D208	I_{PD_CMP}	Comparator	—	32	43	45	µA	3.0V	
D208	I_{PD_CMP}	Comparator	—	31	42	44	µA	3.0V	

* 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 I_{PD} and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base I_{DD} or I_{PD} 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 V_{SS}.

3: All peripheral currents listed are on a per-peripheral basis if more than one instance of a peripheral is available.

4: ADC clock source is ADCRC.

TABLE 34-4: I/O PORTS

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated)				
Param. No.	Sym.	Characteristic	Min.	Typ.†	Max.	Units	Conditions
	VIL	**Input Low Voltage**					
		I/O PORT:					
D300		with TTL buffer	—	—	0.8	V	4.5V ≤ VDD ≤ 5.5V
D301			—	—	0.15 VDD	V	1.8V ≤ VDD ≤ 4.5V
D302		with Schmitt Trigger buffer	—	—	0.2 VDD	V	2.0V ≤ VDD ≤ 5.5V
D303		with I²C™ levels	—	—	0.3 VDD	V	
D304		with SMBus levels	—	—	0.8	V	2.7V ≤ VDD ≤ 5.5V
D305		$\overline{MCLR}$	—	—	0.2 VDD	V	
	VIH	**Input High Voltage**					
		I/O PORT:					
D320		with TTL buffer	2.0	—	—	V	4.5V ≤ VDD ≤ 5.5V
D321			0.25 VDD + 0.8	—	—	V	1.8V ≤ VDD ≤ 4.5V
D322		with Schmitt Trigger buffer	0.8 VDD	—	—	V	2.0V ≤ VDD ≤ 5.5V
D323		with I²C™ levels	0.7 VDD	—	—	V	
D324		with SMBus levels	2.1	—	—	V	2.7V ≤ VDD ≤ 5.5V
D325		$\overline{MCLR}$	0.7 VDD	—	—	V	
	IIL	**Input Leakage Current(2)**					
D340		I/O Ports	—	± 5	± 125	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C
D341			—	± 5	± 1000	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 125°C
D342		$\overline{MCLR}$(2)	—	± 50	± 200	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C
	IPUR	**Weak Pull-up Current**					
D350			25	120	200	µA	VDD = 3.0V, VPIN = VSS
	VOL	**Output Low Voltage(4)**					
D360		I/O ports	—	—	0.6	V	IOL = 10.0 mA, VDD = 3.0V
	VOH	**Output High Voltage(4)**					
D370		I/O ports	VDD - 0.7	—	—	V	IOH = 6.0 mA, VDD = 3.0V
D380	CIO	**All I/O pins**	—	5	50	pF	

* 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 34-5: I/O AND CLOCK TIMING SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
High Voltage Entry Programming Mode Specifications							
MEM01	VIHH	Voltage on $\overline{MCLR}$/VPP pin to enter Programming mode	—	—	—	V	**Note 2**
MEM02	IPPGM	Current on $\overline{MCLR}$/VPP pin during Programming mode	—	—	—	uA	**Note 2**
Programming Mode Specifications							
MEM10	VBE	VDD for Bulk Erase	—	2.7	—	V	
MEM11	IDDPGM	Supply Current during Programming Operation	—	—	—	V	
Data EEPROM Memory Specifications							
MEM20	ED	DataEE Byte Endurance	100k	—	—	E/W	-40°C ≤ TA ≤ 85°C
MEM21	TD_RET	Characteristic Retention	—	40	—	Year	Provided no other specifications are violated
MEM22	ND_REF	Total Erase/Write Cycles before Refresh	—	—	100k	E/W	
MEM23	VD_RW	VDD for Read or Erase/Write Operation	VDDMIN	—	VDDMAX	V	
MEM24	TD_BEW	Byte Erase and Write Cycle Time	—	4.0	5.0	ms	
Program Flash Memory Specifications							
MEM30	EP	Flash Memory Cell Endurance	10k	—	—	E/W	-40°C ≤ Ta ≤ 85°C **(Note 1)**
MEM31	EPHEF	High-Endurance Flash Memory Cell Endurance	100k	—	—	E/W	TBD
MEM32	TP_RET	Characteristic Retention	—	40	—	Year	Provided no other specifications are violated
MEM33	VP_RD	VDD for Read Operation	VDDMIN	—	VDDMAX	V	
MEM34	VP_REW	VDD for Row Erase or Write Operation	VDDMIN	—	VDDMAX	V	
MEM35	TP_REW	Self-Timed Row Erase or Self-Timed Write	—	2.0	2.5	ms	

† 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: Flash Memory Cell Endurance for the Flash memory is defined as: One Row Erase operation and one Self-Timed Write.

2: Required only if CONFIG3.LVP is disabled.

TABLE 34-6: THERMAL CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)					
Param. No.	**Sym.**	**Characteristic**	**Typ.**	**Units**	**Conditions**
TH01	θJA	Thermal Resistance Junction to Ambient	70.0	°C/W	14-pin PDIP package
			95.3	°C/W	14-pin SOIC package
			100.0	°C/W	14-pin TSSOP package
			51.5	°C/W	16-pin UQFN 4x4mm package
			62.2	°C/W	20-pin PDIP package
			87.3	°C/W	20-pin SSOP package
			77.7	°C/W	20-pin SOIC package
			43.0	°C/W	20-pin UQFN 4x4mm package
TH02	θJC	Thermal Resistance Junction to Case	32.75	°C/W	14-pin PDIP package
			31.0	°C/W	14-pin SOIC package
			24.4	°C/W	14-pin TSSOP package
			5.4	°C/W	16-pin UQFN 4x4mm package
			27.5	°C/W	20-pin PDIP package
			31.1	°C/W	20-pin SSOP package
			23.1	°C/W	20-pin SOIC package
			5.3	°C/W	20-pin UQFN 4x4mm package
TH03	TJMAX	Maximum Junction Temperature	150	°C	
TH04	PD	Power Dissipation	0.800	W	PD = PINTERNAL + PI/O
TH05	PINTERNAL	Internal Power Dissipation	—	W	PINTERNAL = IDD x VDD(1)
TH06	PI/O	I/O Power Dissipation	—	W	PI/O = Σ (IOL * VOL) + Σ (IOH * (VDD - VOH))
TH07	PDER	Derated Power	—	W	PDER = PDMAX (TJ - TA)/θJA(2)

Note 1: IDD is current to run the chip alone without driving any load on the output pins.
2: TA = Ambient Temperature, TJ = Junction Temperature

34.4 AC Characteristics

FIGURE 34-4: LOAD CONDITIONS

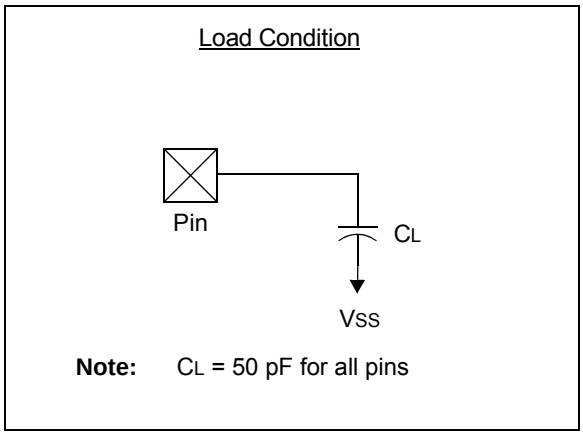


FIGURE 34-5: CLOCK TIMING

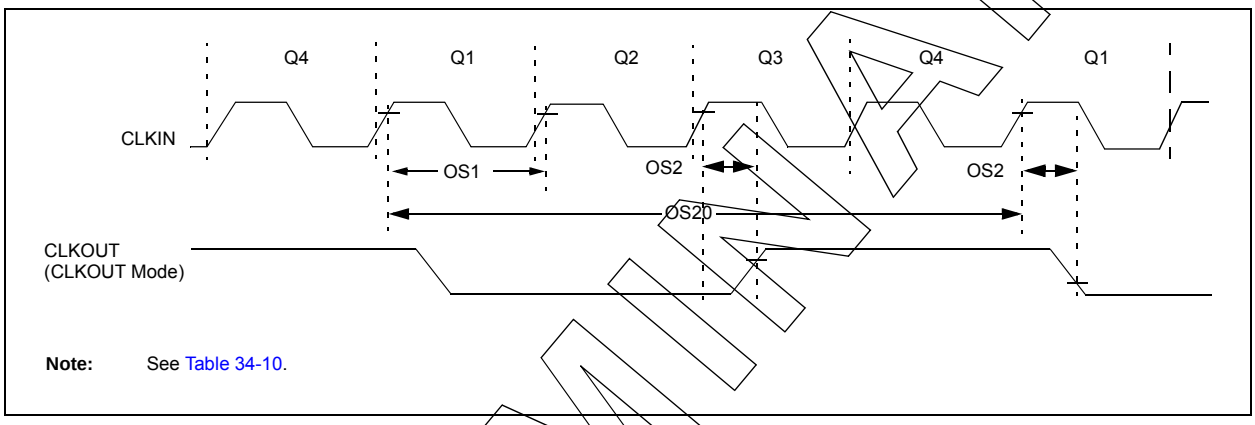


TABLE 34-7: EXTERNAL CLOCK/OSCILLATOR TIMING REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
ECL Oscillator							
OS1	FECL	Clock Frequency	—	—	500	kHz	
OS2	TECL_DC	Clock Duty Cycle	40	—	60	%	
ECM Oscillator							
OS3	FECM	Clock Frequency	—	—	4	MHz	**Note 4**
OS4	TECM_DC	Clock Duty Cycle	40	—	60	%	
ECH Oscillator							
OS5	FECH	Clock Frequency	—	—	32	MHz	
OS6	TECH_DC	Clock Duty Cycle	40	—	60	%	
LP Oscillator							
OS7	FLP	Clock Frequency	—	—	100	kHz	**Note 4**
XT Oscillator							
OS8	FXT	Clock Frequency	—	—	4	MHz	**Note 4**
HS Oscillator							
OS9	FHS	Clock Frequency	—	—	20	MHz	**Note 4**
System Clock							
OS20	FOSC	System Clock Frequency	—	—	32	MHz	**Note 2, Note 3**
OS21	FCY	Instruction Frequency	—	FOSC/4	—	MHz	
OS22	TCY	Instruction Period	125	1/FCY	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

2: The system clock frequency (FOSC) is selected by the "main clock switch controls" as described in **Section 6.3 "Clock Switching"**.

3: The system clock frequency (FOSC) must meet the voltage requirements defined in the **Section 34.2 "Standard Operating Conditions"**. LP, XT and HS oscillator modes require an appropriate crystal or resonator to be connected to the device.

4: For clocking the device with an external square wave, one of the EC mode selections must be used.

TABLE 34-8: OSCILLATOR PARAMETERS(1)

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
OS20	FHFOSC	Precision Calibrated HFINTOSC Frequency	3.92	4	4.08	MHz	25°C
OS20	FHFOSC	Precision Calibrated HFINTOSC Frequency	—	4 8 12 16 32	—	MHz	-40°C to 125°C (2)
OS21	FHFOSCLP	Low-Power Optimized HFINTOSC Frequency	0.93 1.86	1 2	1.07 2.14	MHz MHz	
OS23	FLFOSC	Internal LFINTOSC Frequency	—	31	—	kHz	
OS24	THFOSCST	HFINTOSC Wake-up from Sleep Start-up Time	—	11 50	20 —	µs µs	VREGPM = 0 VREGPM = 1
OS26	TLFOSCST	LFINTOSC Wake-up from Sleep Start-up Time	—	0.2	—	ms	

* 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.

2: See Figure 34-6.

FIGURE 34-6: PRECISION CALIBRATED HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE

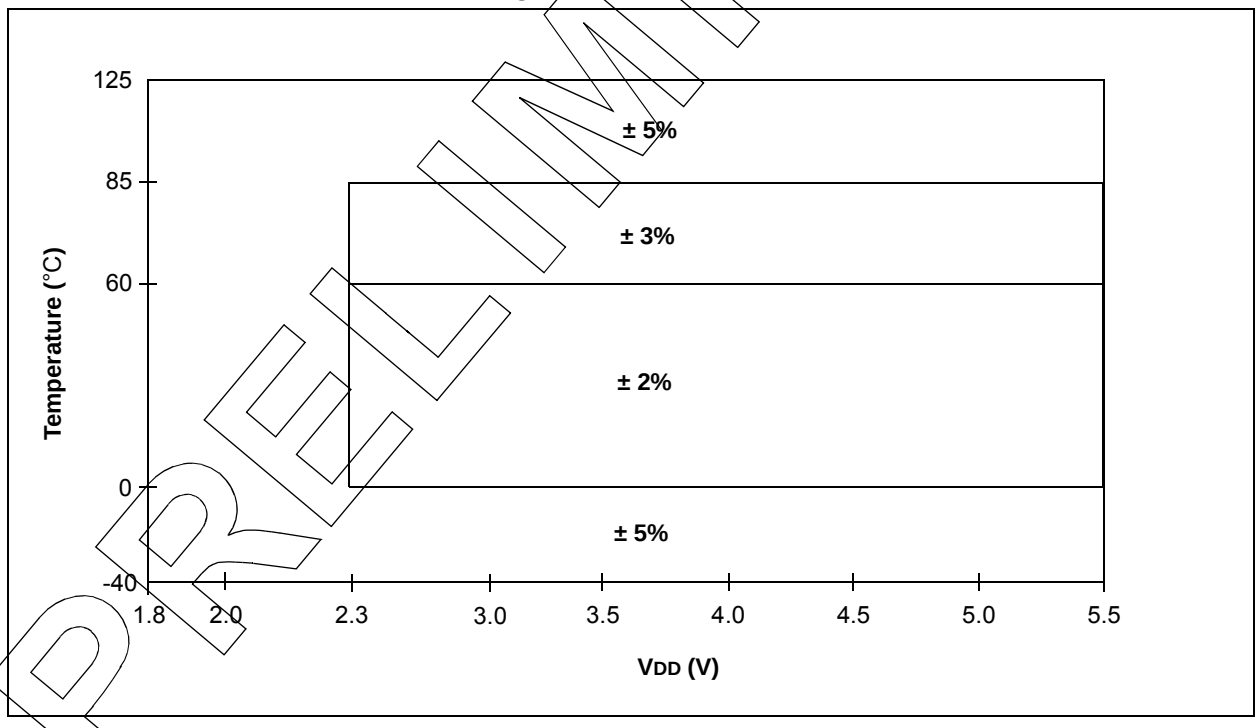


TABLE 34-9: PLL CLOCK TIMING SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
PLL01	FPLLIN	PLL Input Frequency Range	4	—	8	MHz	
PLL02	FPLLOUT	PLL Output Frequency Range	16	—	32	MHz	
PLL03	TPLLST	PLL Lock Time from Start-up	—	200	—	μs	
PLL04	FPLLJIT	PLL Output Frequency Stability (Jitter)	-0.25	—	0.25	%	

* These parameters are characterized but not tested.

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

FIGURE 34-7: CLKOUT AND I/O TIMING

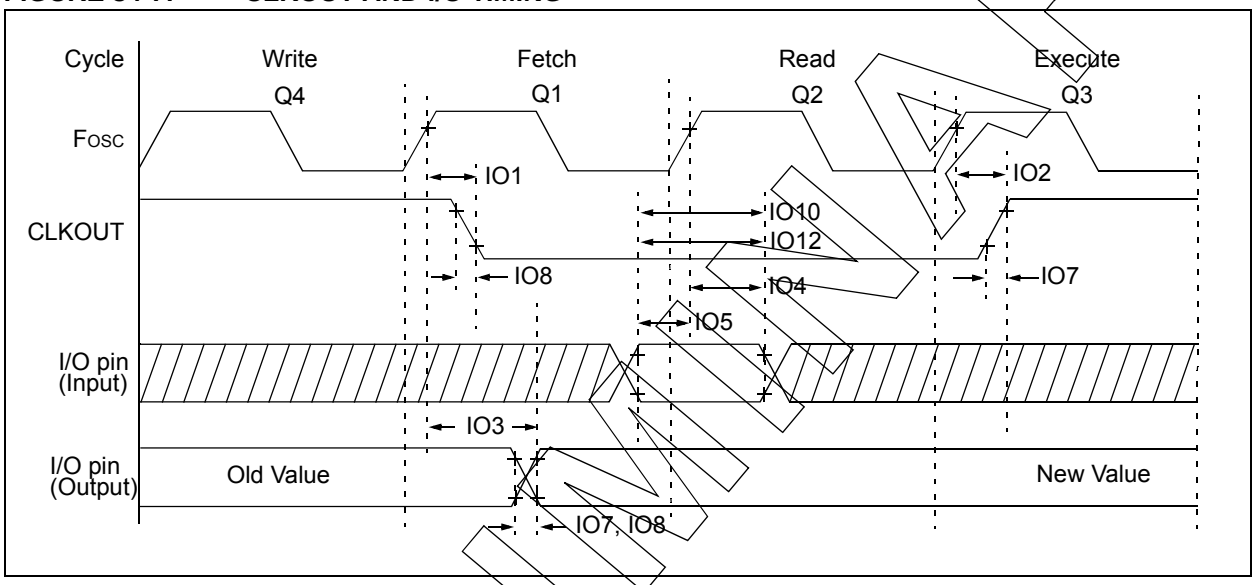


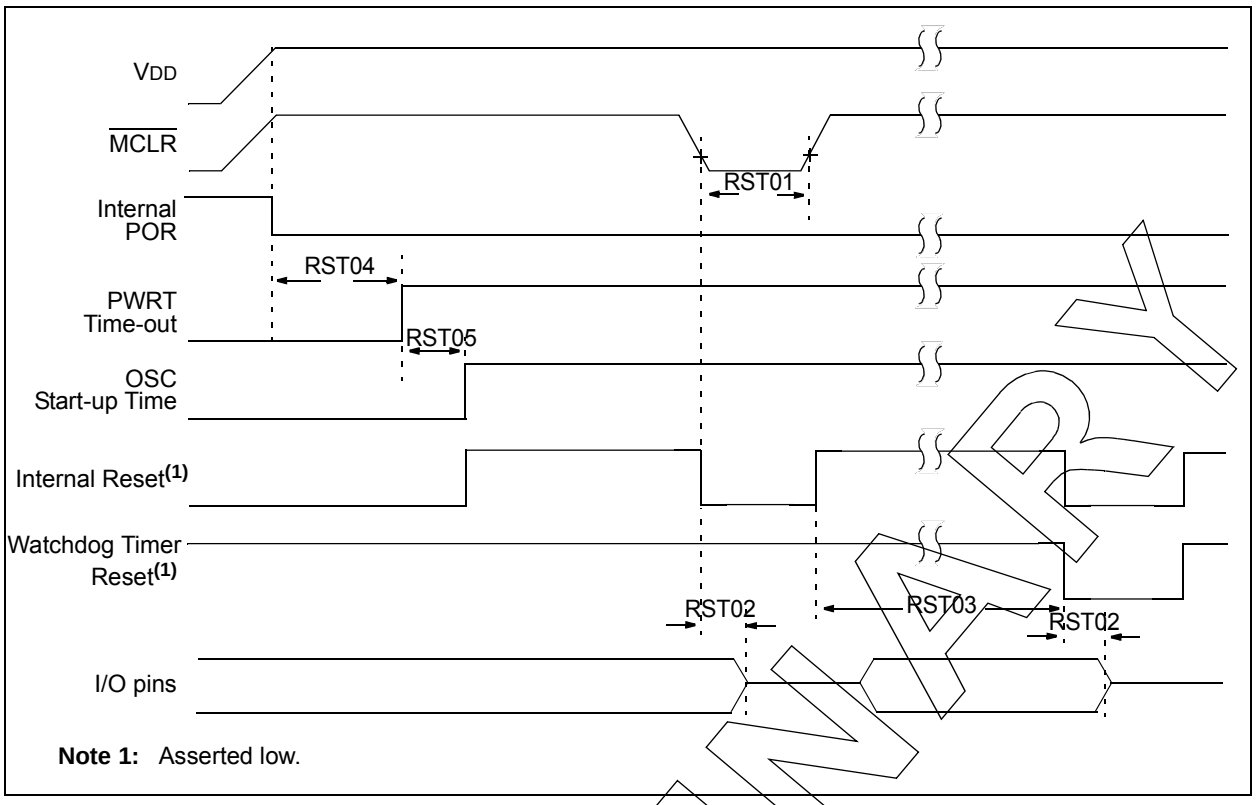
TABLE 34-10: CLKOUT AND I/O TIMING SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
IO1	TCLKOUTH	CLKOUT rising edge delay (rising edge FOSC (Q1 cycle) to falling edge CLKOUT	—	—	—	ns	
IO2	TCLKOUTL	CLKOUT falling edge delay (rising edge FOSC (Q3 cycle) to rising edge CLKOUT	—	—	—	ns	
IO3	TIO_VALID	Port output valid time (rising edge FOSC (Q1 cycle) to port valid)	—	—	—	ns	
IO4	TIO_SETUP	Port input setup time (Setup time before rising edge FOSC - Q2 cycle)	—	—	—	ns	
IO5	TIO_HOLD	Port input hold time (Hold time after rising edge FOSC - Q2 cycle)	—	—	—	ns	
IO6	TIOR_SLREN	Port I/O rise time, slew rate enabled	—	—	—	ns	VDD = 3.0V, Load conditions
IO7	TIOR_SLRDIS	Port I/O rise time, slew rate disabled	—	—	—	ns	VDD = 3.0V, Load conditions
IO8	TIOF_SLREN	Port I/O fall time, slew rate enabled	—	—	—	ns	VDD = 3.0V, Load conditions
IO9	TIOF_SLRDIS	Port I/O fall time, slew rate disabled	—	—	—	ns	VDD = 3.0V, Load conditions
IO10	TINT	INT pin high or low time to trigger an interrupt	—	—	—	ns	
IO11	TIOC	Interrupt-on-Change minimum high or low time to trigger interrupt	—	—	—	ns	

* These parameters are characterized but not tested.

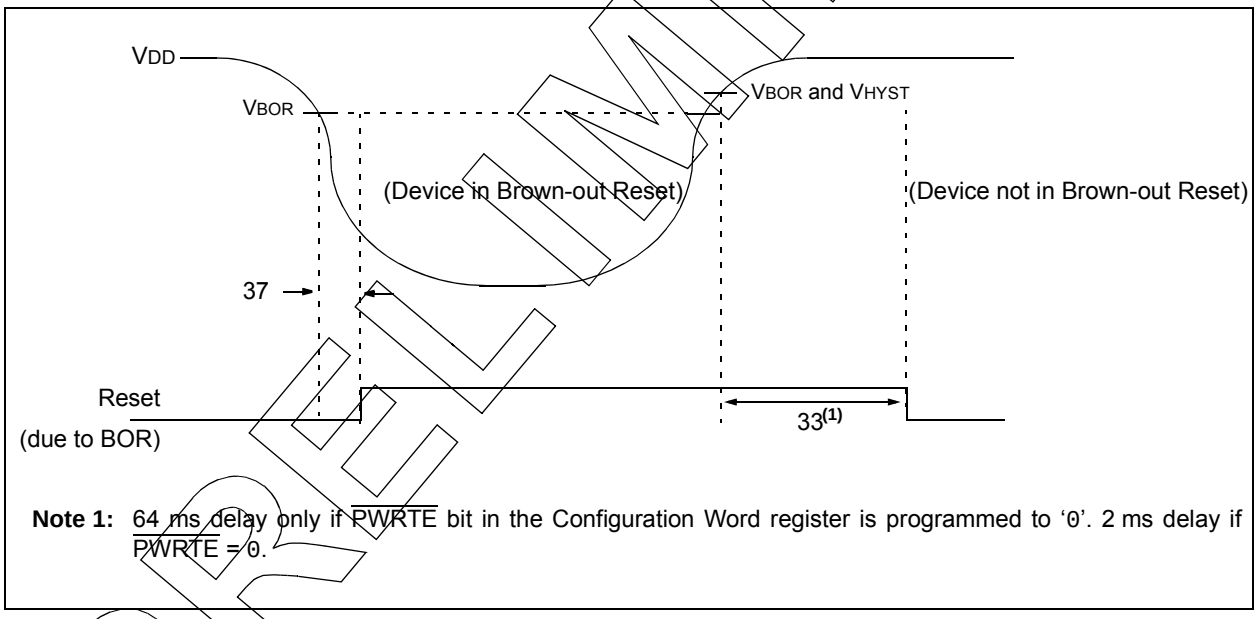
† Data in “Typ.” column is at 3.0V, 25°C unless otherwise stated.

FIGURE 34-8: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Note 1: Asserted low.

FIGURE 34-9: BROWN-OUT RESET TIMING AND CHARACTERISTICS



Note 1: 64 ms delay only if $\overline{PWRTE}$ bit in the Configuration Word register is programmed to '0'. 2 ms delay if $\overline{PWRTE}$ = 0.

TABLE 34-11: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, BROWN-OUT RESET AND LOW POWER BROWN-OUT RESET SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
RST01	TMCLR	MCLR Pulse Width Low to ensure Reset	2	—	—	µs	
RST02	TIOZ	I/O high-impedance from Reset detection	—	—	2	µs	
RST03	TWDT	Watchdog Timer Time-out Period	10	16	27	ms	16 ms Nominal Reset Time
RST04*	TPWRT	Power-up Timer Period	40	65	140	ms	
RST05	TOST	Oscillator Start-up Timer Period(1,2)	—	1024	—	TOSC	**(Note3)**
RST06	VBOR	Brown-out Reset Voltage(4)	2.55 2.30 1.80	2.70 2.45 1.90	2.85 2.60 2.10	V V V	BORV = 0 BORV = 1 (PIC16F18325/18345) BORV = 1 (PIC16LF18325/18345)
RST07	VBORHYS	Brown-out Reset Hysteresis	0	25	75	mV	
RST08	TBORDC	Brown-out Reset Response Time	1	3	35	µs	
RST09	VLPBOR	Low-Power Brown-out Reset Voltage	2.3	2.45	2.7	V	PIC16F18325/18345
			—	—	—	V	PIC16LF18325/18345

* 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 the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

2: By design.

3: Period of the slower clock.

4: 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.

TABLE 34-12: ANALOG-TO-DIGITAL CONVERTER (ADC) CHARACTERISTICS(1,2)

Standard Operating Conditions (unless otherwise stated) VDD = 3.0V, TA = 25°C							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
AD01	NR	Resolution	—	—	10	bit	
AD02	EIL	Integral Error	—	±0.1	±1.0	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD03	EDL	Differential Error	—	±0.1	±1.0	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD04	EOFF	Offset Error	—	0.5	2	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD05	EGN	Gain Error	—	±0.2	±1.0	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD06	VADREF	ADC Reference Voltage (ADREF+)(3)	1.8	—	VDD	V	
AD07	VAIN	Full-Scale Range	VSS	—	ADREF+	V	
AD06	VADREF	ADC Reference Voltage (ADREF+ - ADREF-)(3)	1.8	—	VDD	V	
AD07	VAIN	Full-Scale Range	ADREF-	—	ADREF+	V	
AD08	ZAIN	Recommended Impedance of Analog Voltage Source	—	10	—	kΩ	
AD09	RVREF	ADC Voltage Reference Ladder Impedance	—	—	—	kΩ	

* 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 is the sum of the offset, gain and integral non-linearity (INL) errors.

2: The ADC conversion result never decreases with an increase in the input and has no missing codes.

TABLE 34-13: ANALOG-TO DIGITAL CONVERTER (ADC) CONVERSION TIMING SPECIFICATIONS(1,2)

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
AD20	TAD	ADC Clock Period	1	—	9	us	Using FOSC as the ADC clock source; ADCS ! = x11
AD21			1	2	6	us	Using ADCRC as the ADC clock source; ADCS = x11
AD22	TCNV	Conversion Time	—	11	—	TAD	Set of GO/DONE bit to Clear of GO/DONE bit
AD23	TACQ	Acquisition Time	—	2	—	us	
AD24	THCD	Sample and Hold Capacitor Disconnect Time	—	—	—	us	FOSC based clock source
			—	—	—	us	ADCRC based clock source

* 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 34-10: ADC CONVERSION TIMING (ADC CLOCK FOSC-BASED)

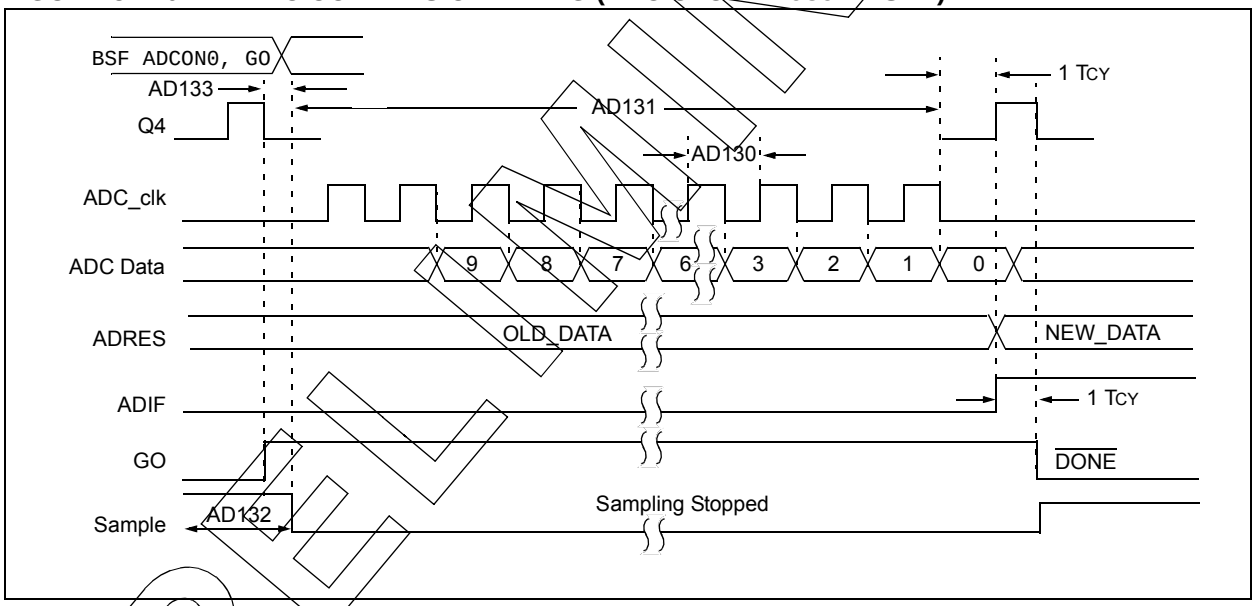
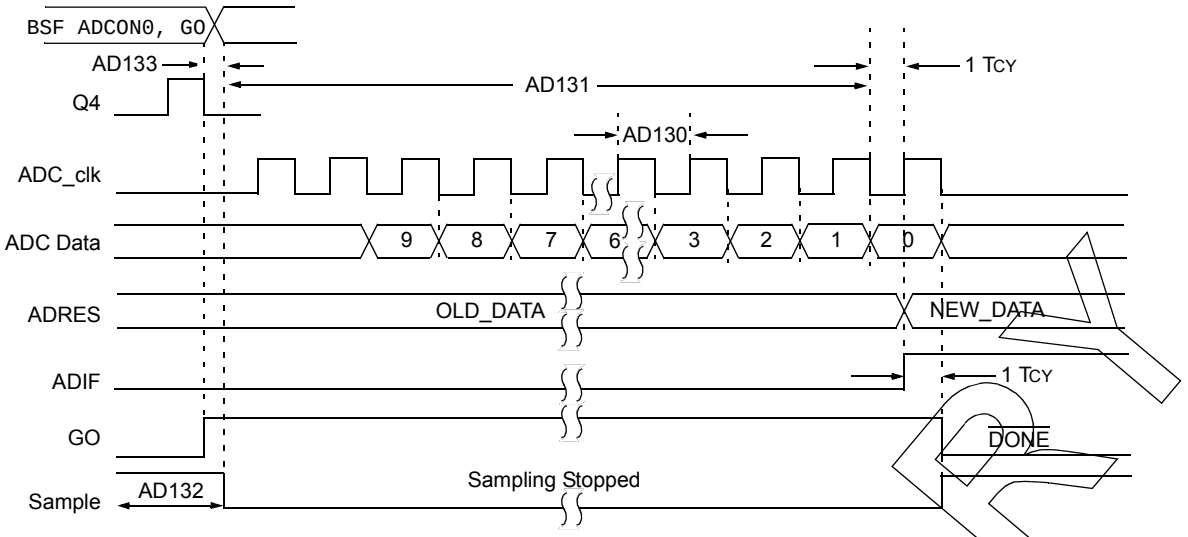


FIGURE 34-11: ADC CONVERSION TIMING (ADC CLOCK FROM ADCRC)



Note: If the ADC clock source is selected as ADCRC, a time of TCY is added before the ADC clock starts. This allows the SLEEP instruction to be executed.

TABLE 34-14: COMPARATOR SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated) V_{DD} = 3.0V, T_A = 25°C See Section 35.0 "DC and AC Characteristics Graphs and Charts" for operating characterization.							
Param No.	**Sym.**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Comments**
CM01	V_{IOFF}	Input Offset Voltage	—	—	±40	mV	V_{ICM} = V_{DD}/2
CM02	V_{ICM}	Input Common Mode Voltage	GND	—	V_{DD}	V	
CM03	CMRR	Common Mode Input Rejection Ratio	—	50	—	dB	
CM04	C_{HYST}	Comparator Hysteresis	15	25	35	mV	
CM05	T_{RESP}(1)	Response Time, Rising Edge	—	300	600	ns	
		Response Time, Falling Edge	—	220	500	ns	
CM06*	T_{MCV2VO}(2)	Mode Change to Valid Output	—	—	10	us	

* These parameters are characterized but not tested.

Note 1: Response time measured with one comparator input at V_{DD}/2, while the other input transitions from V_{SS} to V_{DD}.

2: A mode change includes changing any of the control register values, including module enable.

TABLE 34-15: DIGITAL-TO-ANALOG CONVERTER (DAC) SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated) V_{DD} = 3.0V, T_A = 25°C							
Param No.	**Sym.**	**Characteristics**	**Min.**	**Typ.†**	**Max.**	**Units**	**Comments**
DSB01	V_{LSB}	Step Size	—	V_{DD}/32	—	V	
DSB01	V_{ACC}	Absolute Accuracy	—	—	± 0.5	LSb	
DSB03*	R_{UNIT}	Unit Resistor Value	—	6000	—	Ω	
DSB04*	T_{ST}	Settling Time(1)	—	—	10	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: Settling time measured while DACR<4:0> transitions from '00000' to '01111'.

TABLE 34-16: FIXED VOLTAGE REFERENCE (FVR) SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Symbol**	**Characteristic**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
FVR01	V_{FVR1}	1x Gain (1.024V nominal)	-4	—	4	%	V_{DD} ≥ 2.5V, -40°C to 85°C
FVR02	V_{FVR2}	2x Gain (2.048V nominal)	-4	—	4	%	V_{DD} ≥ 2.5V, -40°C to 85°C
FVR03	V_{FVR4}	4x Gain (4.096V nominal)	-5	—	5	%	V_{DD} ≥ 4.75V, -40°C to 85°C
FVR04	T_{FVRST}	FVR Start-up Time	—	—	—	µs	

FIGURE 34-12: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

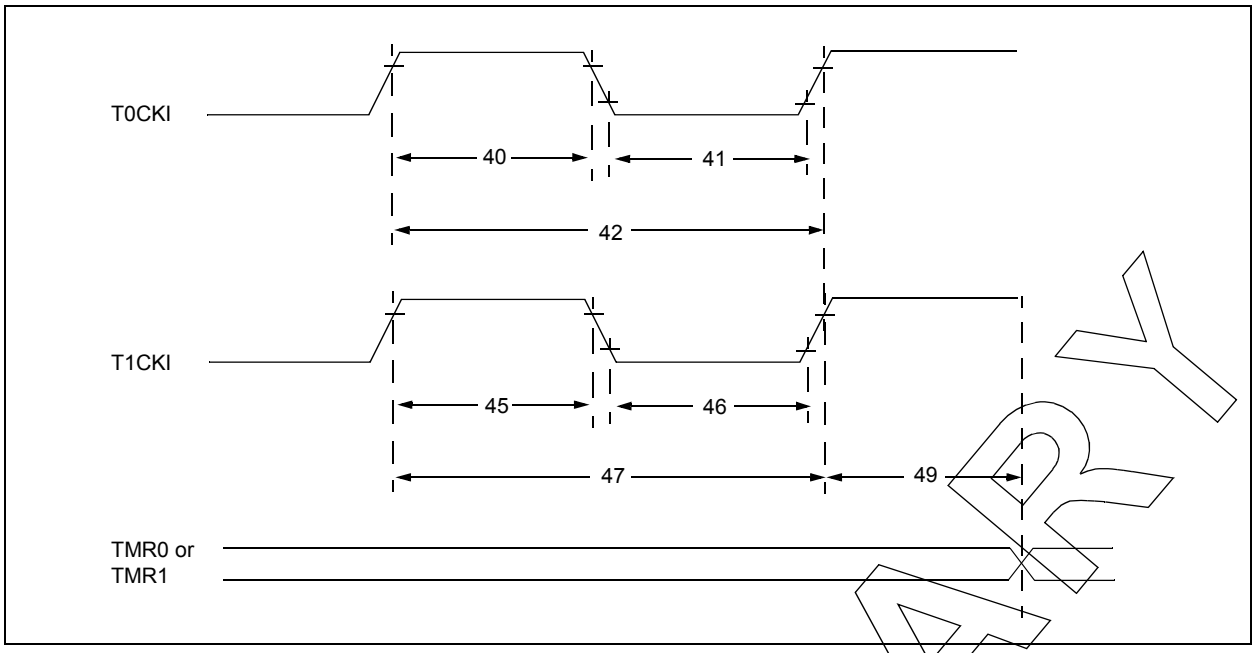


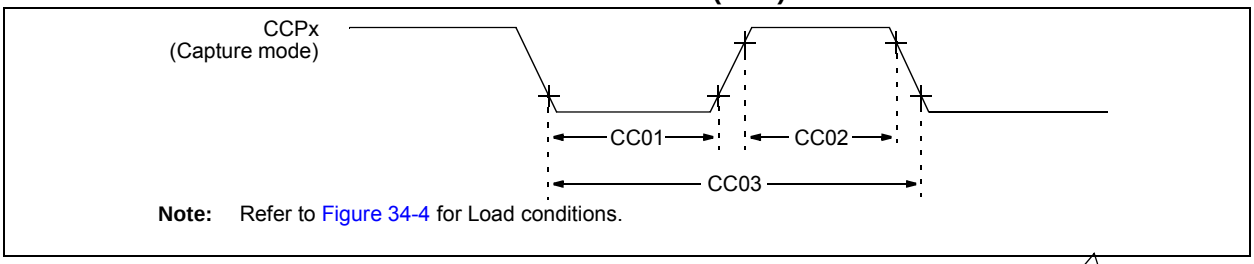
TABLE 34-17: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

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 TCY + 40 / N	—	—	ns	N = prescale value
45*	TT1H	T1CKI High Time	Synchronous, No Prescaler	0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
46*	TT1L	T1CKI Low Time	Synchronous, No Prescaler	0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
47*	TT1P	T1CKI Input Period	Synchronous	Greater of: 30 or TCY + 40 / N	—	—	ns	N = prescale value
			Asynchronous	60	—	—	ns	
48	FT1	Secondary Oscillator Input Frequency Range (oscillator enabled by setting bit T1OSCEN)		32.4	32.768	33.1	kHz	
49*	TCKEZTMR1	Delay from External Clock Edge to Timer Increment		2 TOSC	—	7 TOSC	—	Timers in Sync mode

* 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 34-13: CAPTURE/COMPARE/PWM TIMINGS (CCP)



Note: Refer to Figure 34-4 for Load conditions.

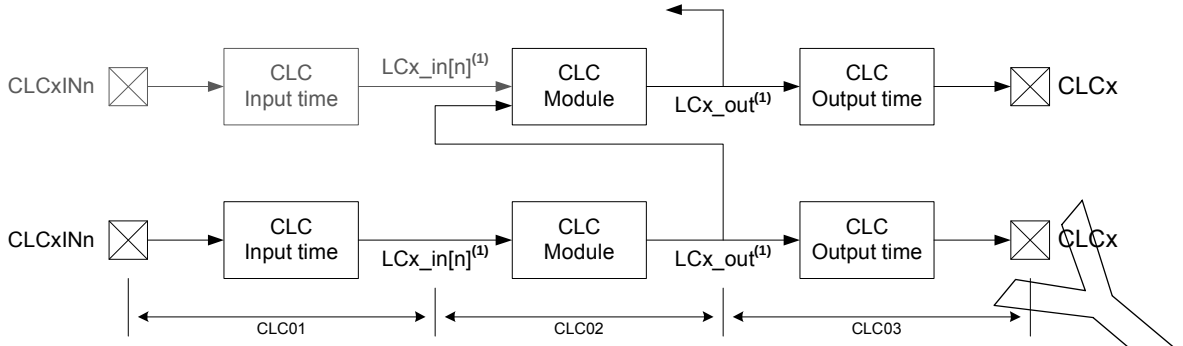
TABLE 34-18: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP)

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

* These parameters are characterized but not tested.

† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 34-14: CLC PROPAGATION TIMING



Note 1: See Figure 20-1, "CLCx Simplified Block Diagram" to identify specific CLC signals.

TABLE 34-19: CONFIGURABLE LOGIC CELL (CLC) CHARACTERISTICS

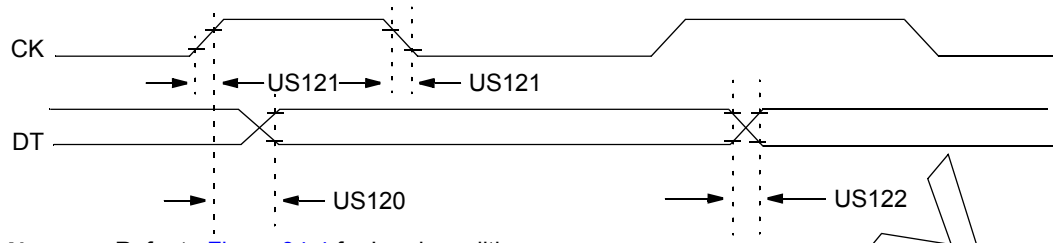
Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
CLC01*	TCLCIN	CLC input time	—	7	OS17	ns	**(Note 1)**
CLC02*	TCLC	CLC module input to output progagation time	— —	24 12	— —	ns ns	VDD = 1.8V VDD > 3.6V
CLC03*	TCLCOUT	CLC output time Rise Time	—	OS18	—	—	**(Note 1)**
		Fall Time	—	OS19	—	—	**(Note 1)**
CLC04*	FCLCMAX	CLC maximum switching frequency	—	32	FOSC	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 34-10 for OS17, OS18 and OS19 rise and fall times.

FIGURE 34-15: EUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

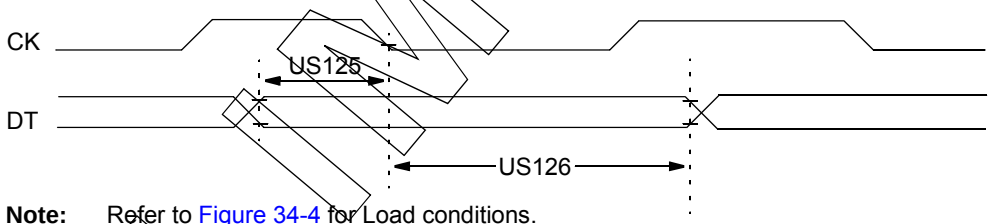


Note: Refer to Figure 34-4 for Load conditions.

TABLE 34-20: EUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)						
Param. No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
US120	TCKH2DTV	SYNC XMIT (Master and Slave) Clock high to data-out valid	—	80	ns	3.0V ≤ VDD ≤ 5.5V
			—	100	ns	1.8V ≤ VDD ≤ 5.5V
US121	TCKRF	Clock out rise time and fall time (Master mode)	—	45	ns	3.0V ≤ VDD ≤ 5.5V
			—	50	ns	1.8V ≤ VDD ≤ 5.5V
US122	TDTRF	Data-out rise time and fall time	—	45	ns	3.0V ≤ VDD ≤ 5.5V
			—	50	ns	1.8V ≤ VDD ≤ 5.5V

FIGURE 34-16: EUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

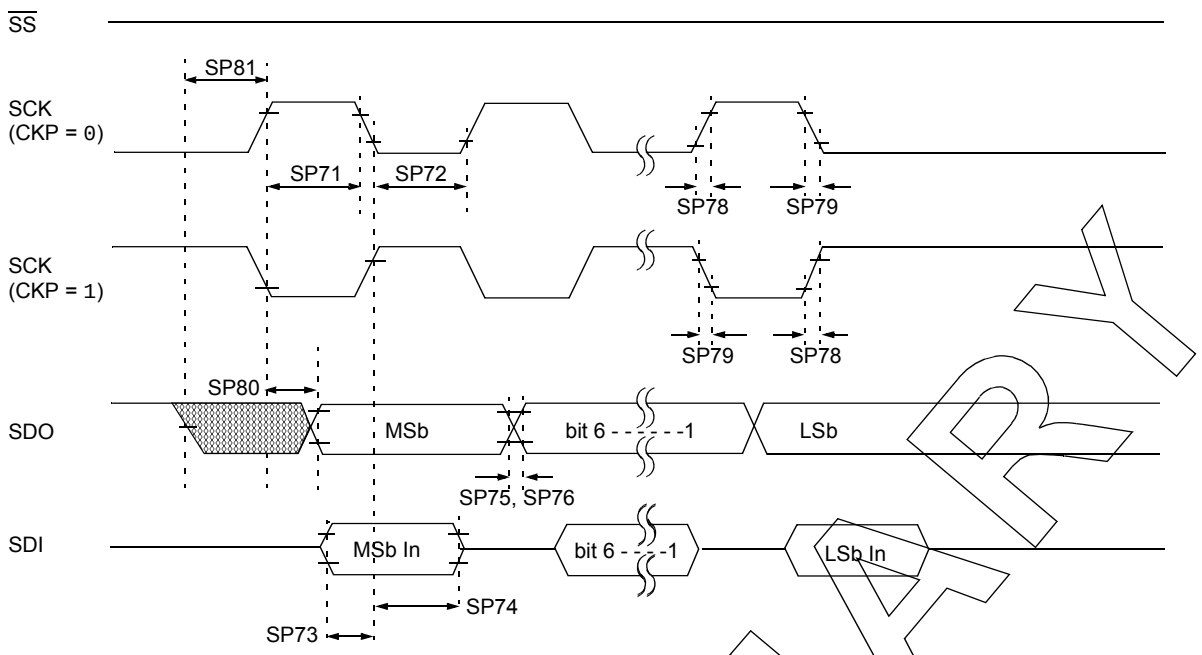


Note: Refer to Figure 34-4 for Load conditions.

TABLE 34-21: EUSART SYNCHRONOUS RECEIVE REQUIREMENTS

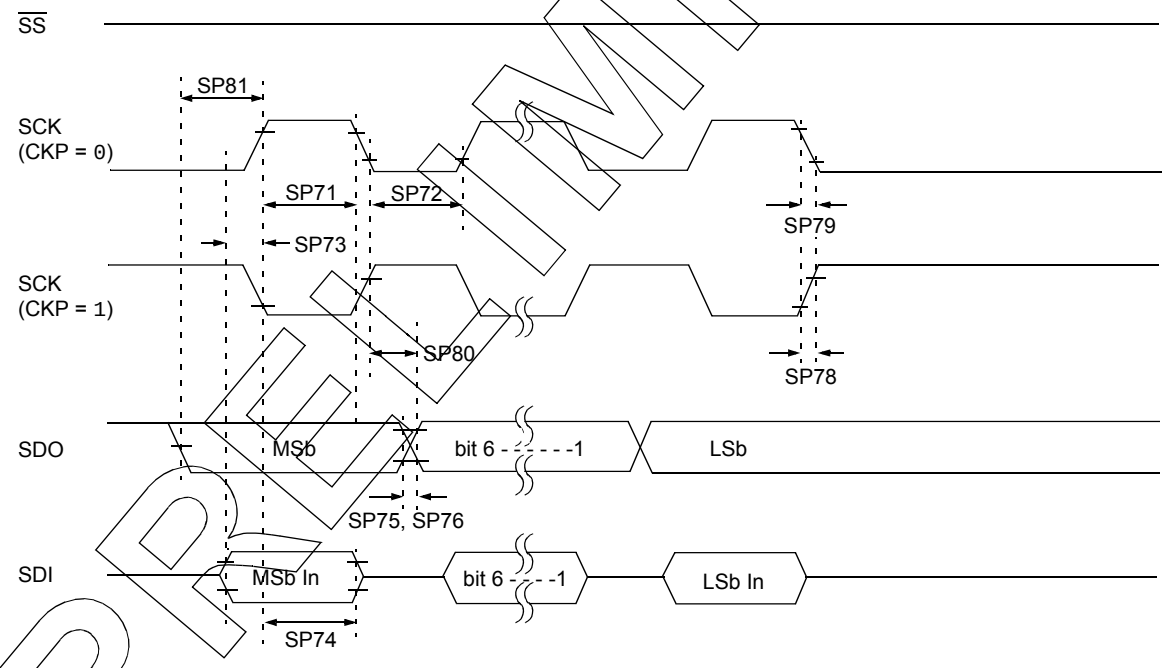
Standard Operating Conditions (unless otherwise stated)						
Param. No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
US125	TDTV2CKL	SYNC RCV (Master and Slave) Data-setup before CK ↓ (DT hold time)	10	—	ns	
US126	TCKL2DTL	Data-hold after CK ↓ (DT hold time)	15	—	ns	

FIGURE 34-17: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)



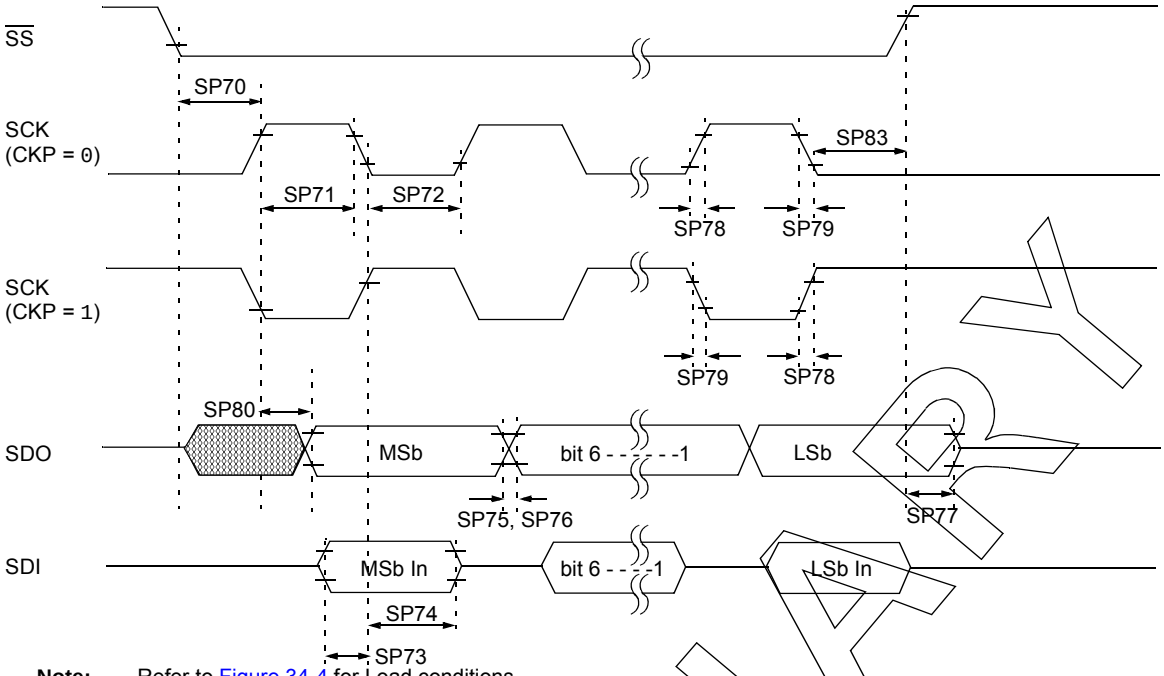
Note: Refer to Figure 34-4 for Load conditions.

FIGURE 34-18: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)



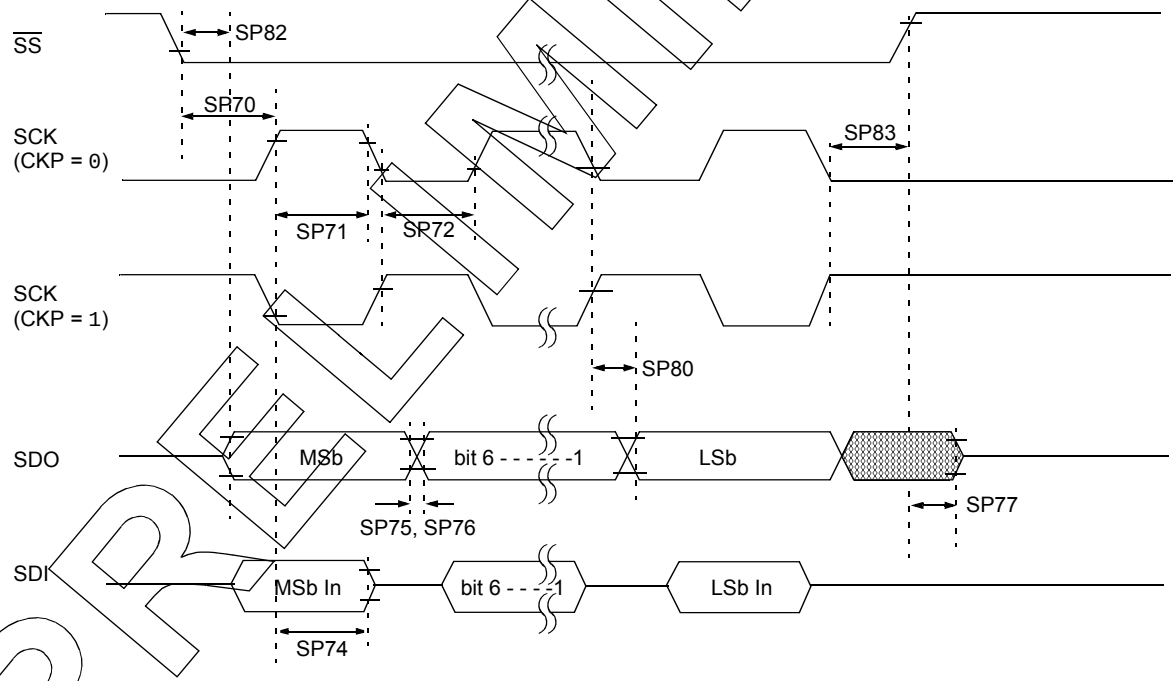
Note: Refer to Figure 34-4 for Load conditions.

FIGURE 34-19: SPI SLAVE MODE TIMING (CKE = 0)



Note: Refer to Figure 34-4 for Load conditions.

FIGURE 34-20: SPI SLAVE MODE TIMING (CKE = 1)



Note: Refer to Figure 34-4 for Load conditions.

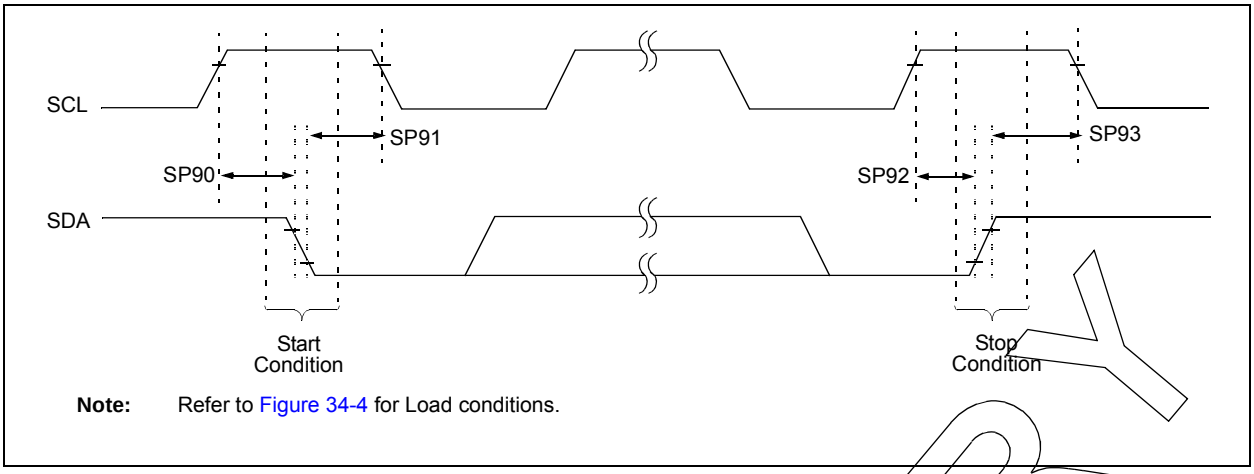
TABLE 34-22: SPI MODE REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Symbol**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
SP70*	TssL2scH, TssL2scL	$\overline{SS}$↓ to SCK↓ or SCK↑ input	2.25*Tcy	—	—	ns	
SP71*	TscH	SCK input high time (Slave mode)	Tcy + 20	—	—	ns	
SP72*	TscL	SCK input low time (Slave mode)	Tcy + 20	—	—	ns	
SP73*	TdiV2scH, TdiV2scL	Setup time of SDI data input to SCK edge	100	—	—	ns	
SP74*	TscH2diL, TscL2diL	Hold time of SDI data input to SCK edge	100	—	—	ns	
SP75*	TdoR	SDO data output rise time	—	10	25	ns	3.0V ≤ VDD ≤ 5.5V
			—	25	50	ns	1.8V ≤ VDD ≤ 5.5V
SP76*	TdoF	SDO data output fall time	—	10	25	ns	
SP77*	TssH2doZ	$\overline{SS}$↑ to SDO output high-impedance	10	—	50	ns	
SP78*	TscR	SCK output rise time (Master mode)	—	10	25	ns	3.0V ≤ VDD ≤ 5.5V
			—	25	50	ns	1.8V ≤ VDD ≤ 5.5V
SP79*	TscF	SCK output fall time (Master mode)	—	10	25	ns	
SP80*	TscH2doV, TscL2doV	SDO data output valid after SCK edge	—	—	50	ns	3.0V ≤ VDD ≤ 5.5V
			—	—	145	ns	1.8V ≤ VDD ≤ 5.5V
SP81*	TdoV2scH, TdoV2scL	SDO data output setup to SCK edge	1 Tcy	—	—	ns	
SP82*	TssL2doV	SDO data output valid after $\overline{SS}$↓ edge	—	—	50	ns	
SP83*	TscH2ssH, TscL2ssH	$\overline{SS}$ ↑ after SCK edge	1.5 Tcy + 40	—	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 34-21: I²C™ BUS START/STOP BITS TIMING



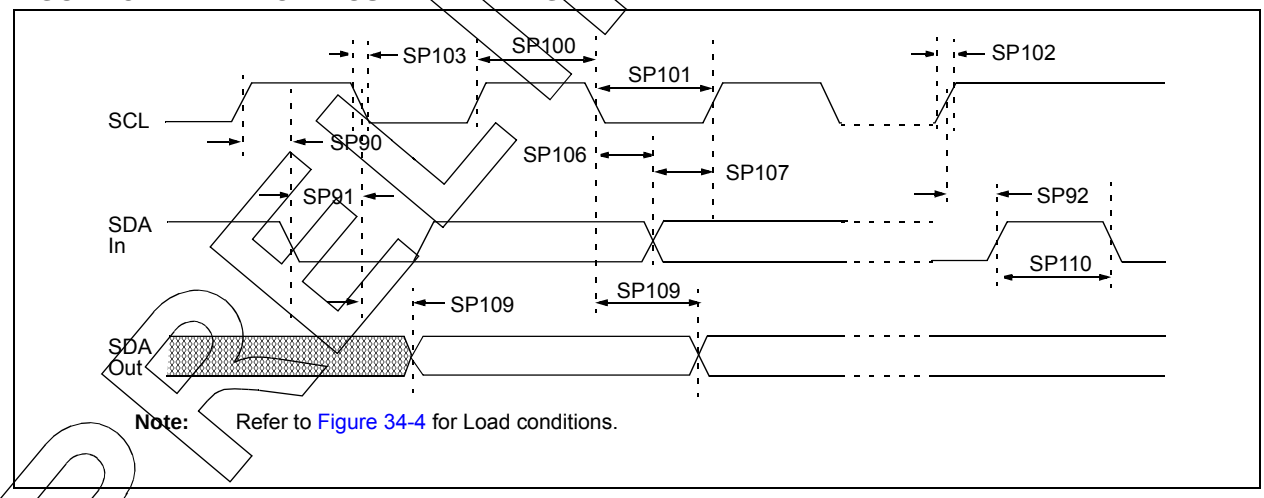
Note: Refer to Figure 34-4 for Load conditions.

TABLE 34-23: I²C™ BUS START/STOP BITS REQUIREMENT S

Standard Operating Conditions (unless otherwise stated)								
Param. No.	**Symbol**	**Characteristic**		**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
SP90*	TSU:STA	Start condition Setup time	100 kHz mode	4700	—	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	600	—	—		
SP91*	THD:STA	Start condition Hold time	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	600	—	—		
SP92*	TSU:STO	Stop condition Setup time	100 kHz mode	4700	—	—	ns	
			400 kHz mode	600	—	—		
SP93	THD:STO	Stop condition Hold time	100 kHz mode	4000	—	—	ns	
			400 kHz mode	600	—	—		

* These parameters are characterized but not tested.

FIGURE 34-22: I²C™ BUS DATA TIMING



Note: Refer to Figure 34-4 for Load conditions.

TABLE 34-24: I²C™ BUS DATA REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Symbol**	**Characteristic**		**Min.**	**Max.**	**Units**	**Conditions**
SP100*	THIGH	Clock high time	100 kHz mode	4.0	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	µs	Device must operate at a minimum of 10 MHz
			SSP module	1.5TCY	—		
SP101*	TLOW	Clock low time	100 kHz mode	4.7	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	µs	Device must operate at a minimum of 10 MHz
			SSP module	1.5TCY	—		
SP102*	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1CB	300	ns	CB is specified to be from 10-400 pF
SP103*	TF	SDA and SCL fall time	100 kHz mode	—	250	ns	
			400 kHz mode	20 + 0.1CB	250	ns	CB is specified to be from 10-400 pF
SP106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	µs	
SP107*	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	**(Note 2)**
			400 kHz mode	100	—	ns	
SP109*	TAA	Output valid from clock	100 kHz mode	—	3500	ns	**(Note 1)**
			400 kHz mode	—	—	ns	
SP110*	TBUF	Bus free time	100 kHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	µs	
SP111	CB	Bus capacitive loading		—	400	pF	

* These parameters are characterized but not tested.

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

2: A Fast mode (400 kHz) I²C™ bus device can be used in a Standard mode (100 kHz) I²C bus system, but the requirement TSU:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

35.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

The graphs and tables provided in this section are for **design guidance** and are **not tested**.

In some graphs or tables, the data presented are **outside specified operating range** (i.e., outside specified V_{DD} range). This is for **information only** and devices are ensured to operate properly only within the specified range.

Unless otherwise noted, all graphs apply to both the L and LF devices.

Note:	The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

"Typical" represents the mean of the distribution at 25°C. "MAXIMUM", "Max.", "MINIMUM" or "Min." represents (mean + 3σ) or (mean - 3σ) respectively, where σ is a standard deviation, over each temperature range.

Charts and graphs are not available at this time.

36.0 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

36.1 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

36.2 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

36.3 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

36.4 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

36.5 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

36.6 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.

36.7 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 upgradeable 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.

36.8 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.

36.9 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™).

36.10 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.

36.11 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.

36.12 Third-Party Development Tools

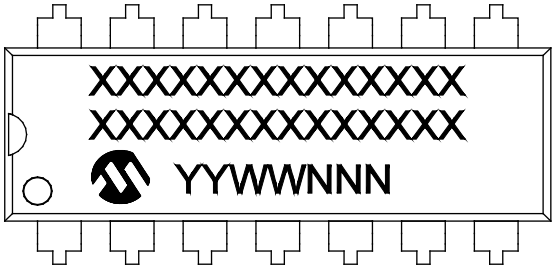
Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- 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®

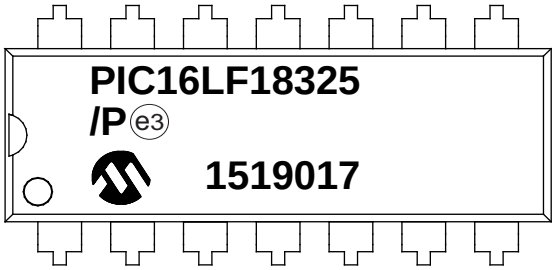
37.0 PACKAGING INFORMATION

37.1 Package Marking Information

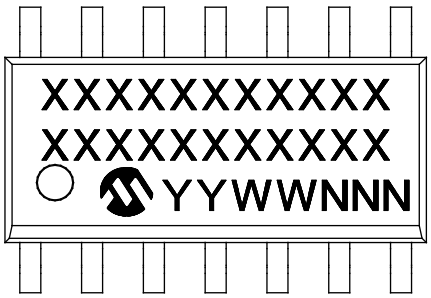
14-Lead PDIP (300 mil)



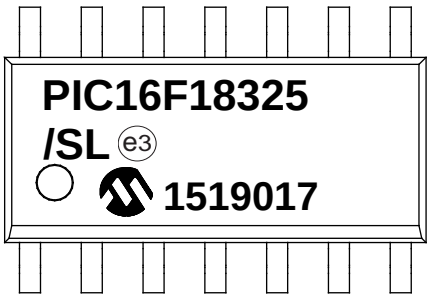
Example



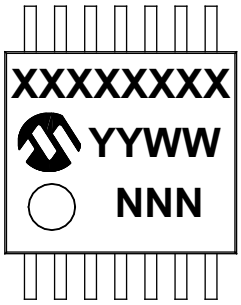
14-Lead SOIC (3.90 mm)



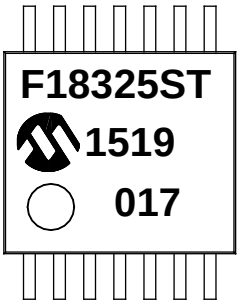
Example



14-Lead TSSOP (4.4 mm)



Example

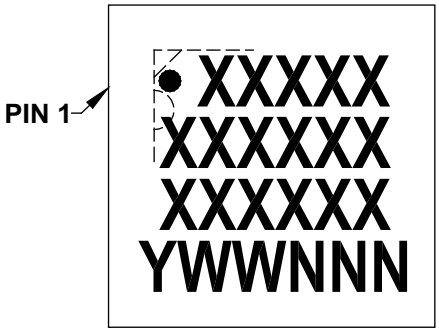


Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

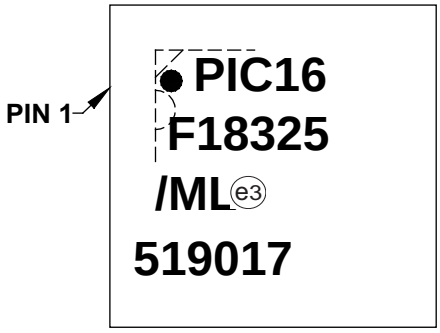
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

Package Marking Information (Continued)

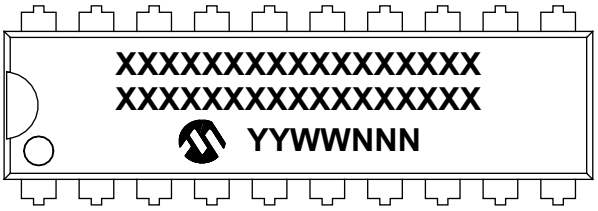
16-Lead UQFN (4x4x0.5mm)



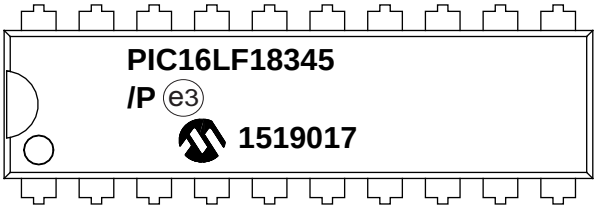
Example



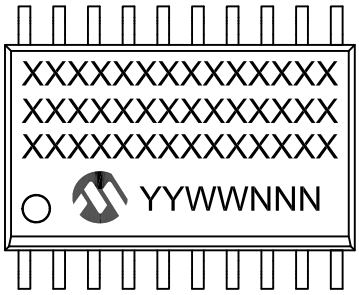
20-Lead PDIP (300 mil)



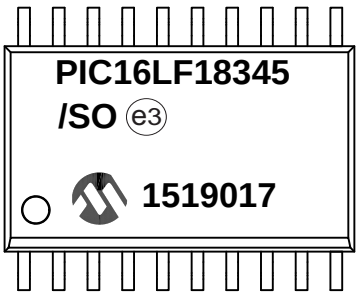
Example



20-Lead SOIC (7.50 mm)



Example

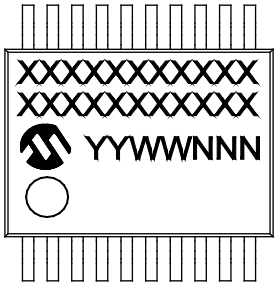


Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

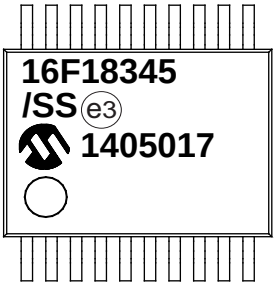
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

Package Marking Information (Continued)

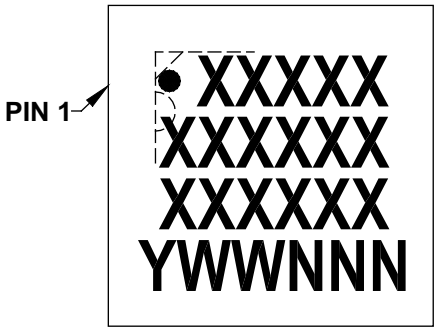
20-Lead SSOP (5.30 mm)



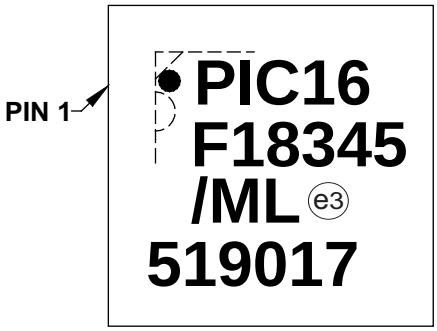
Example



20-Lead UQFN (4x4x0.5 mm)



Example



Legend:	XX...X	Customer-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.

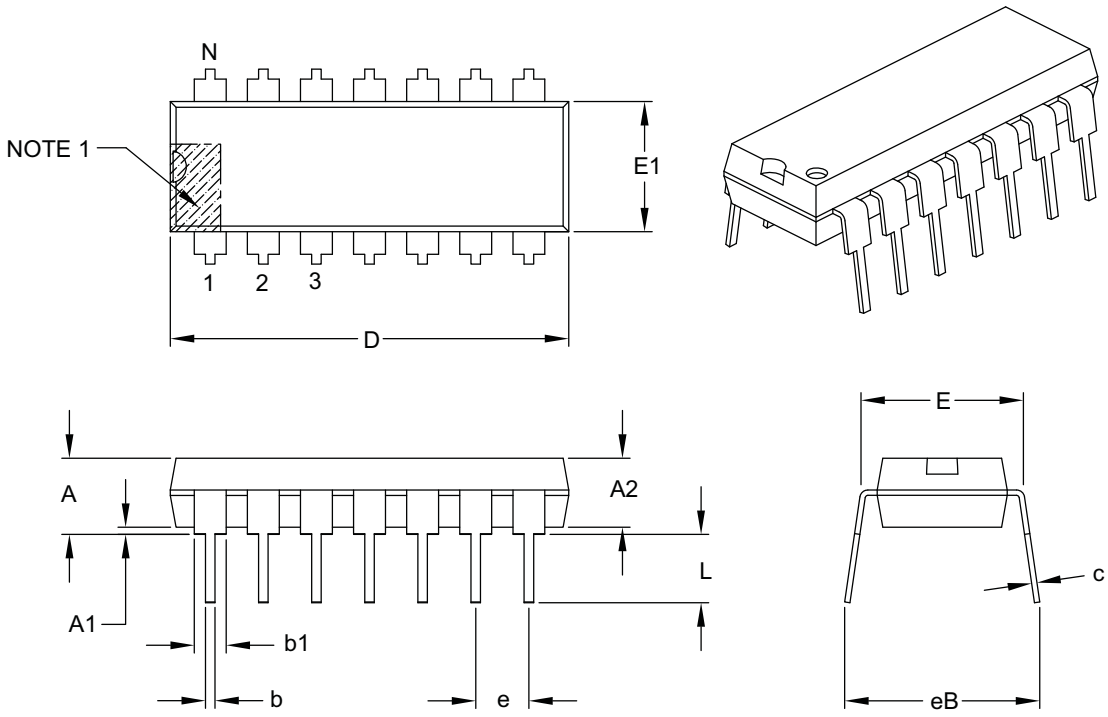
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

37.2 Package Details

The following sections give the technical details of the packages.

14-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		14	
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	.735	.750	.775
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

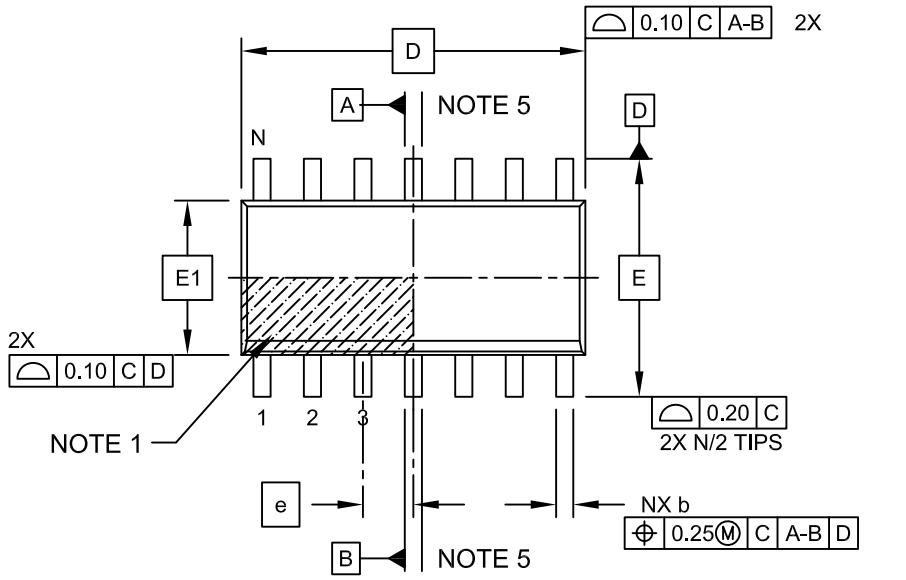
1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

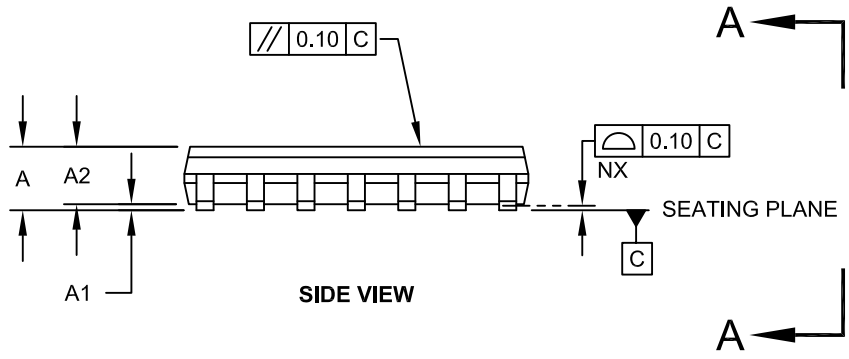
Microchip Technology Drawing C04-005B

14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

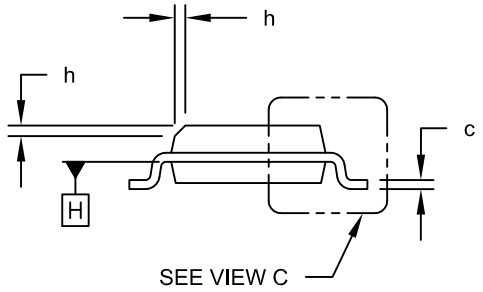
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW

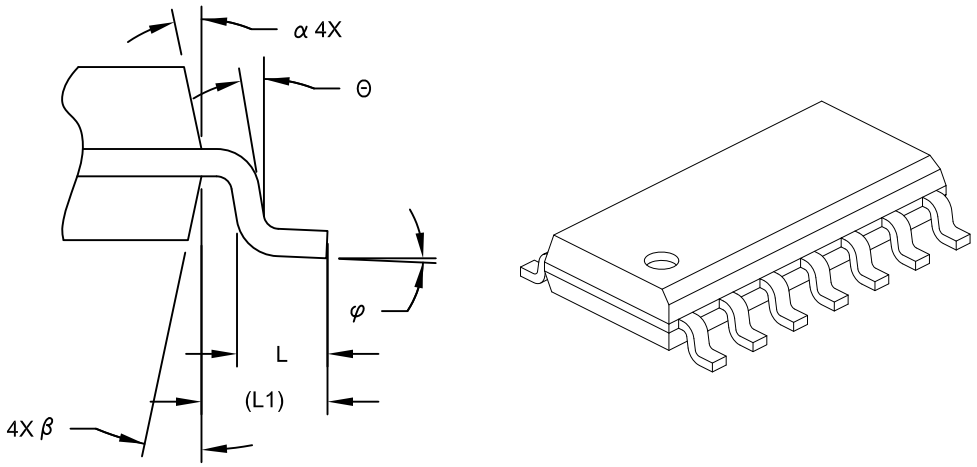


VIEW A-A

Microchip Technology Drawing No. C04-065C Sheet 1 of 2

14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

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



VIEW C

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		14	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	1.75
Molded Package Thickness	A2	1.25	-	-
Standoff §	A1	0.10	-	0.25
Overall Width	E		6.00 BSC	
Molded Package Width	E1		3.90 BSC	
Overall Length	D		8.65 BSC	
Chamfer (Optional)	h	0.25	-	0.50
Foot Length	L	0.40	-	1.27
Footprint	L1		1.04 REF	
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.10	-	0.25
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

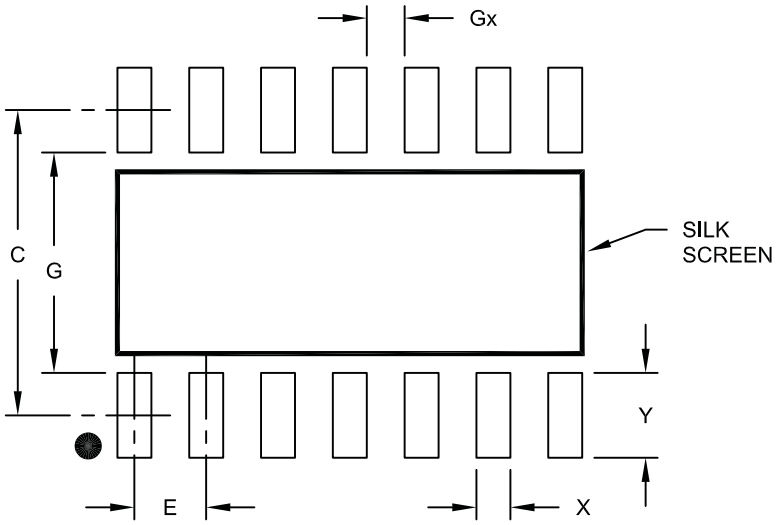
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-065C Sheet 2 of 2

14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

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		1.27 BSC	
Contact Pad Spacing	C		5.40	
Contact Pad Width	X			0.60
Contact Pad Length	Y			1.50
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	3.90		

Notes:

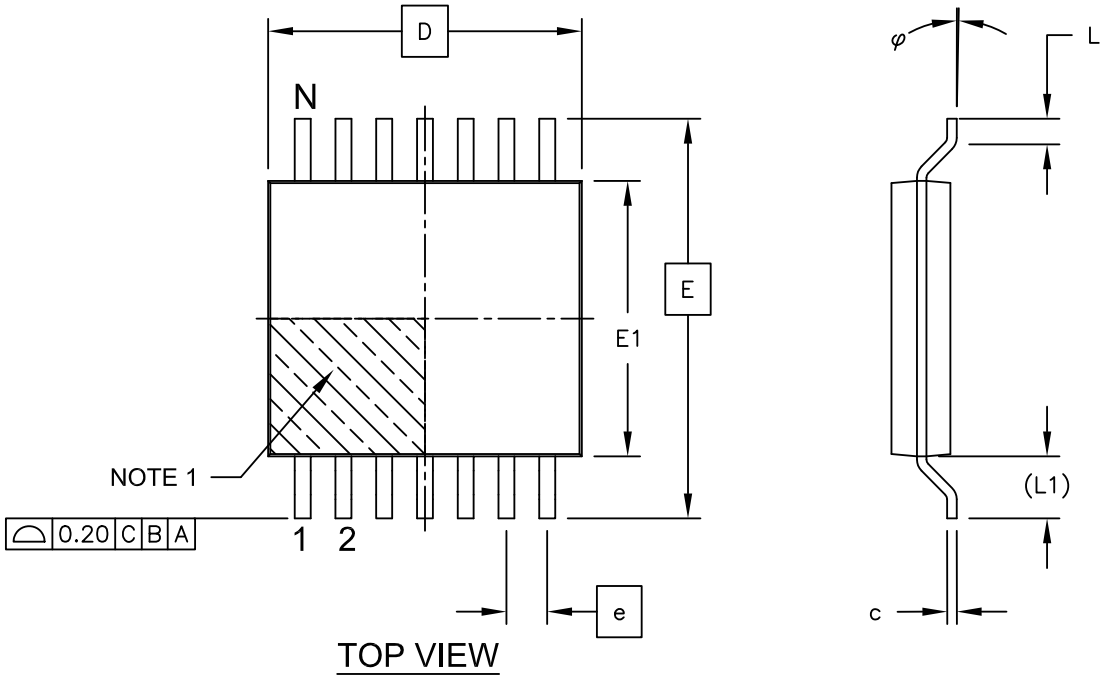
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

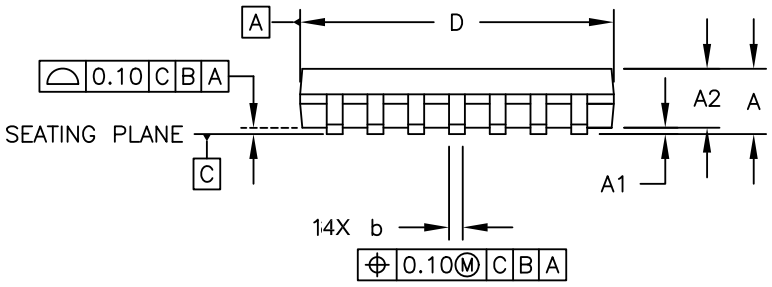
Microchip Technology Drawing No. C04-2065A

14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

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



TOP VIEW

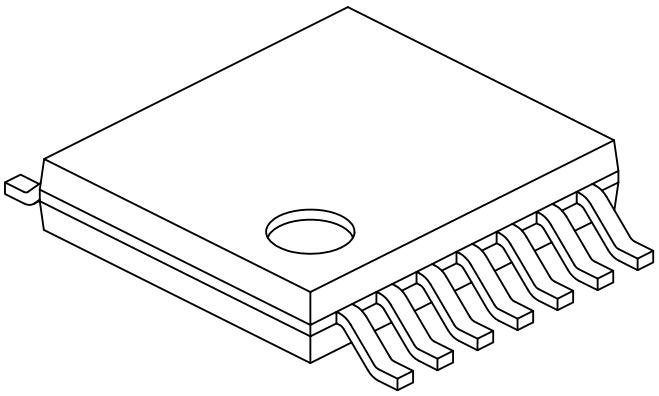


SIDE VIEW

Microchip Technology Drawing C04-087C Sheet 1 of 2

14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

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		14	
Pitch	e		0.65 BSC	
Overall Height	A	-	-	1.20
Molded Package Thickness	A2	0.80	1.00	1.05
Standoff	A1	0.05	-	0.15
Overall Width	E		6.40 BSC	
Molded Package Width	E1	4.30	4.40	4.50
Molded Package Length	D	4.90	5.00	5.10
Foot Length	L	0.45	0.60	0.75
Footprint	(L1)		1.00 REF	
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.09	-	0.20
Lead Width	b	0.19	-	0.30

Notes:

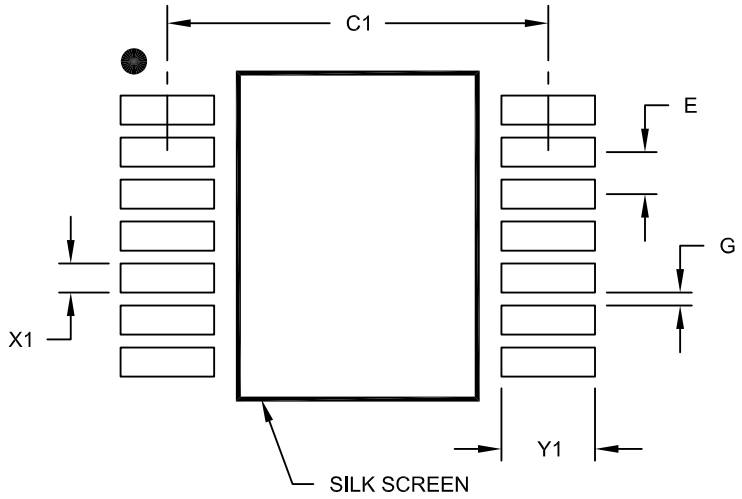
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-087C Sheet 2 of 2

14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

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		0.65 BSC	
Contact Pad Spacing	C1		5.90	
Contact Pad Width (X14)	X1			0.45
Contact Pad Length (X14)	Y1			1.45
Distance Between Pads	G	0.20		

Notes:

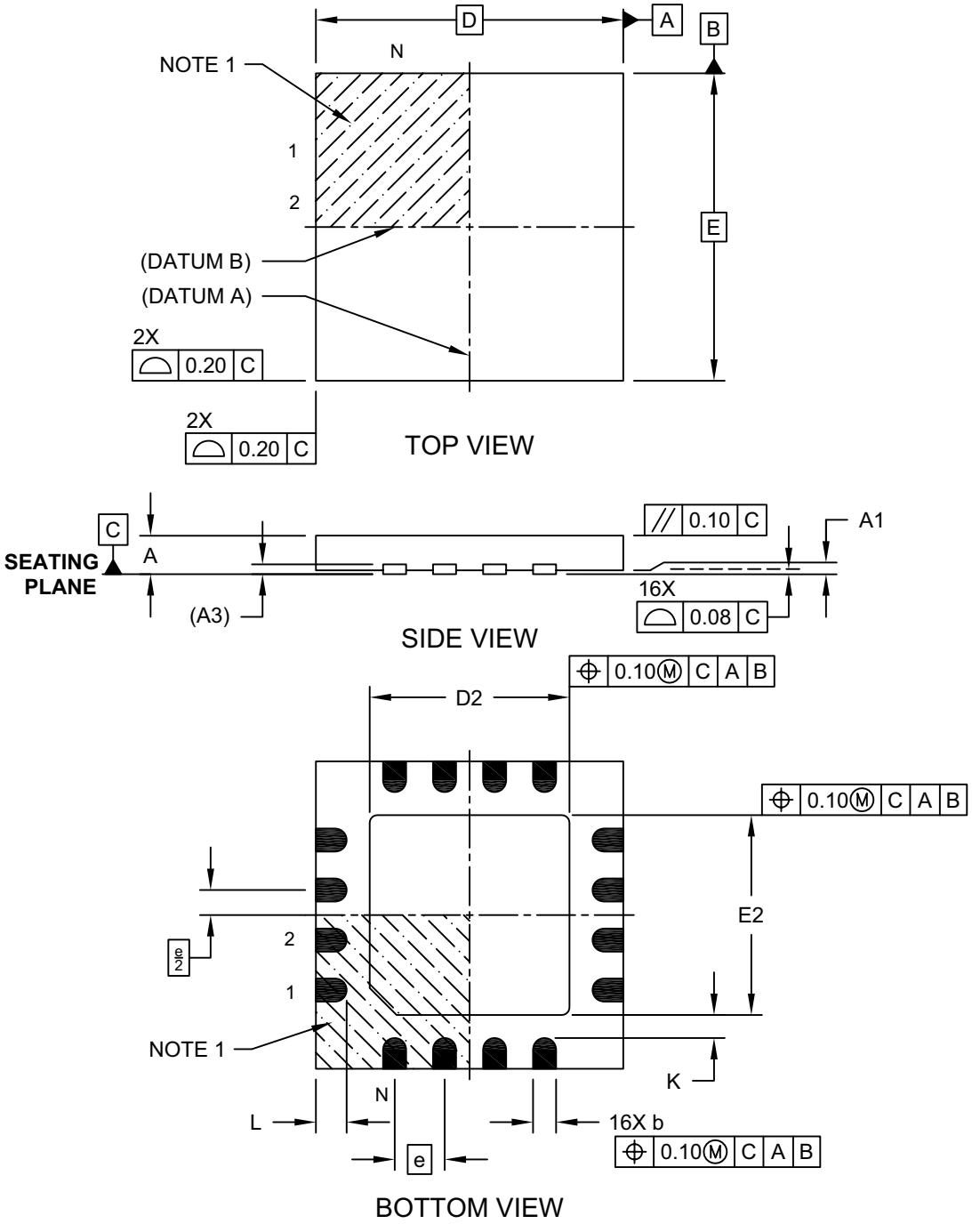
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2087A

16-Lead Ultra Thin Plastic Quad Flat, No Lead Package (JQ) - 4x4x0.5 mm Body [UQFN]

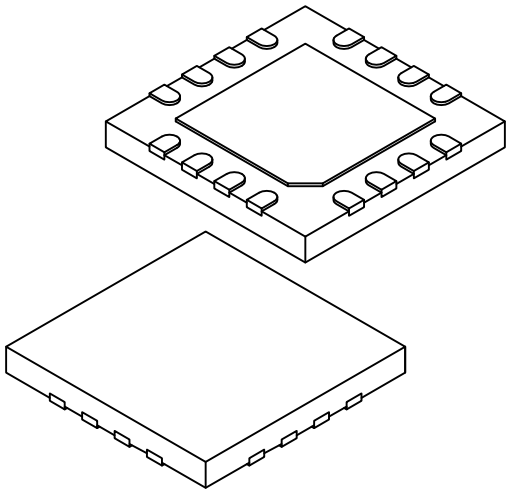
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-257A Sheet 1 of 2

16-Lead Ultra Thin Plastic Quad Flat, No Lead Package (JQ) - 4x4x0.5 mm Body [UQFN]

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		16	
Pitch	e		0.65 BSC	
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.127 REF	
Overall Width	E		4.00 BSC	
Exposed Pad Width	E2	2.50	2.60	2.70
Overall Length	D		4.00 BSC	
Exposed Pad Length	D2	2.50	2.60	2.70
Terminal Width	b	0.25	0.30	0.35
Terminal Length	L	0.30	0.40	0.50
Terminal-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 is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

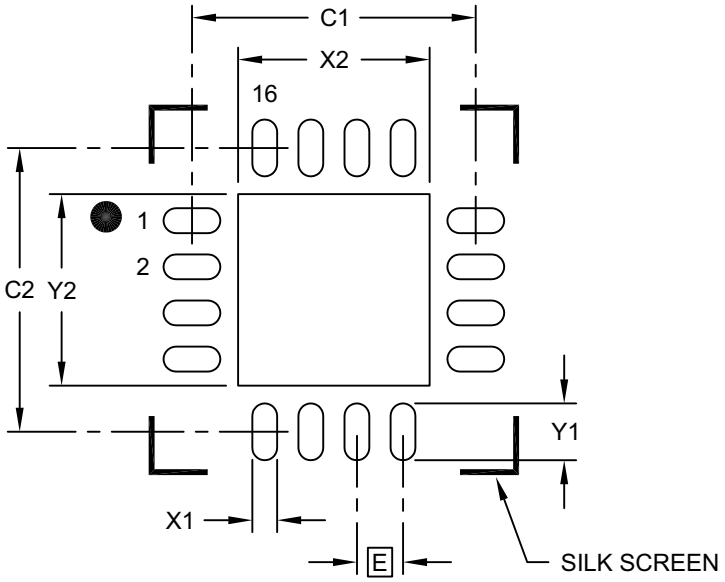
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-257A Sheet 2 of 2

16-Lead Ultra Thin Plastic Quad Flat, No Lead Package (JQ) - 4x4x0.5 mm Body [UQFN]

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		0.65 BSC	
Optional Center Pad Width	X2			2.70
Optional Center Pad Length	Y2			2.70
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X16)	X1			0.35
Contact Pad Length (X16)	Y1			0.80

Notes:

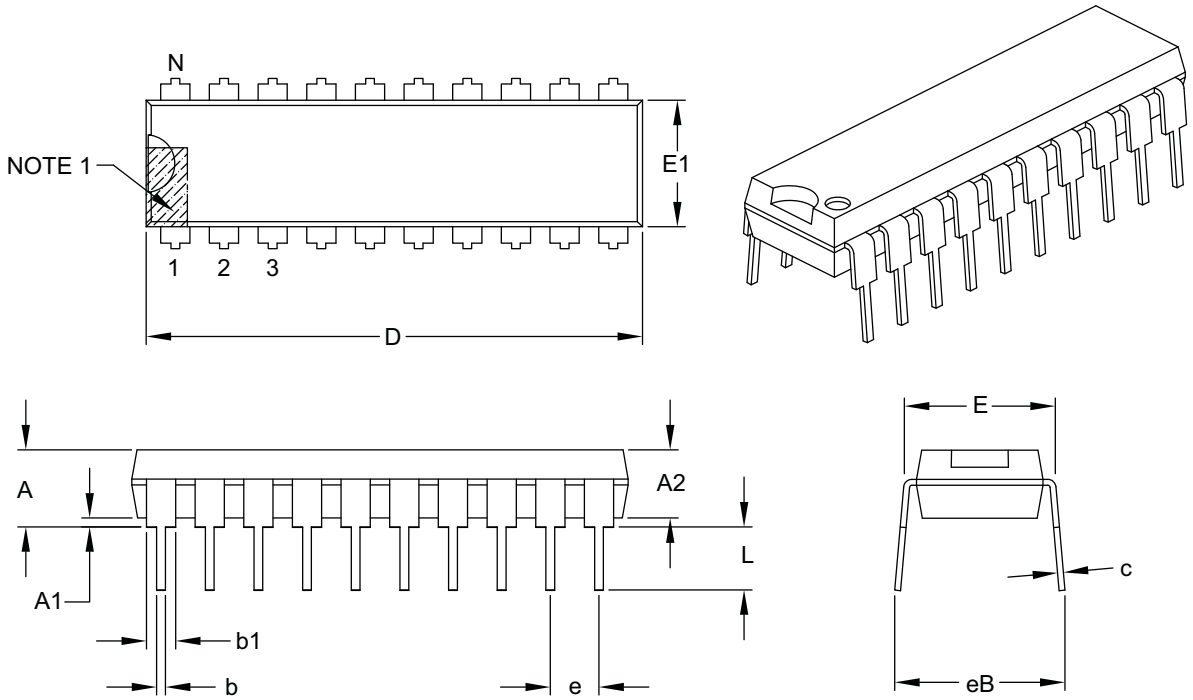
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2257A

20-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		20	
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	.300	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.980	1.030	1.060
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

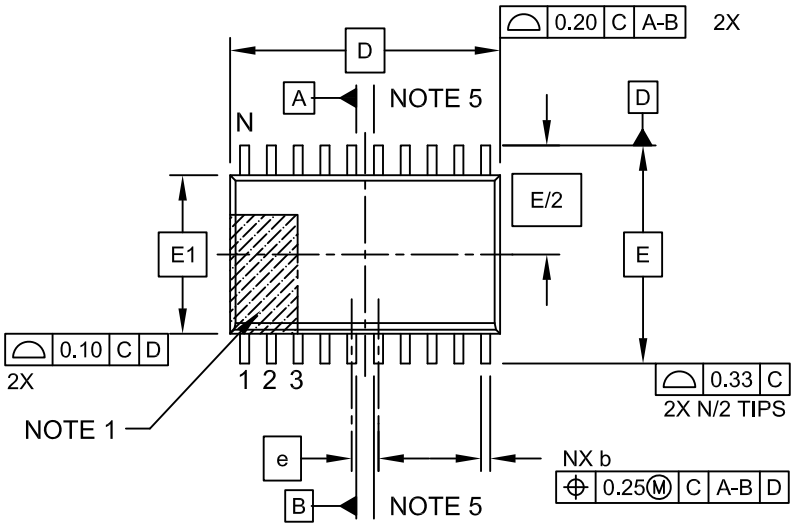
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

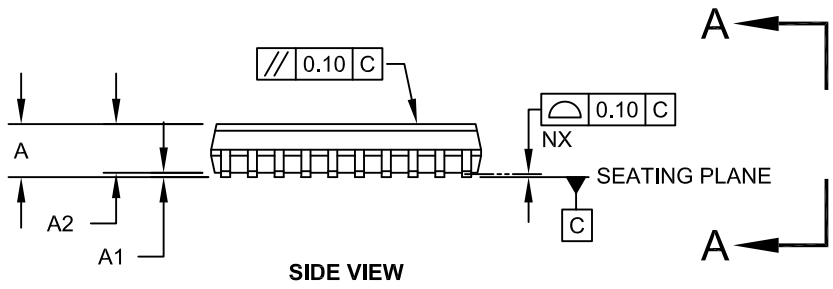
Microchip Technology Drawing C04-019B

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

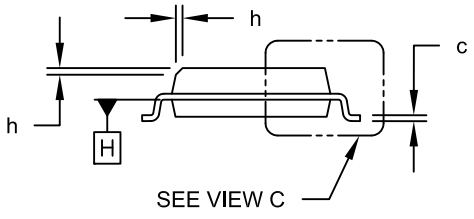
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW

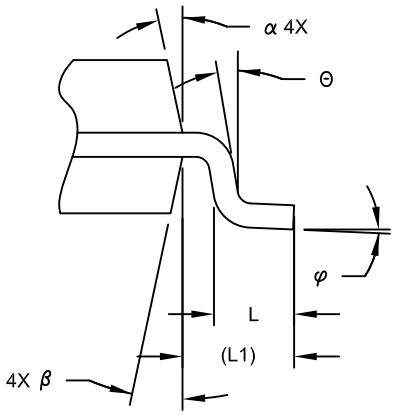


VIEW A-A

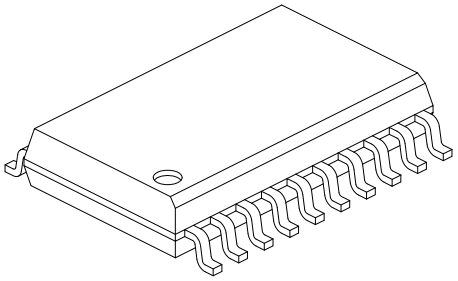
Microchip Technology Drawing C04-094C Sheet 1 of 2

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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



VIEW C



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		20	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		12.80 BSC	
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1		1.40 REF	
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.20	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

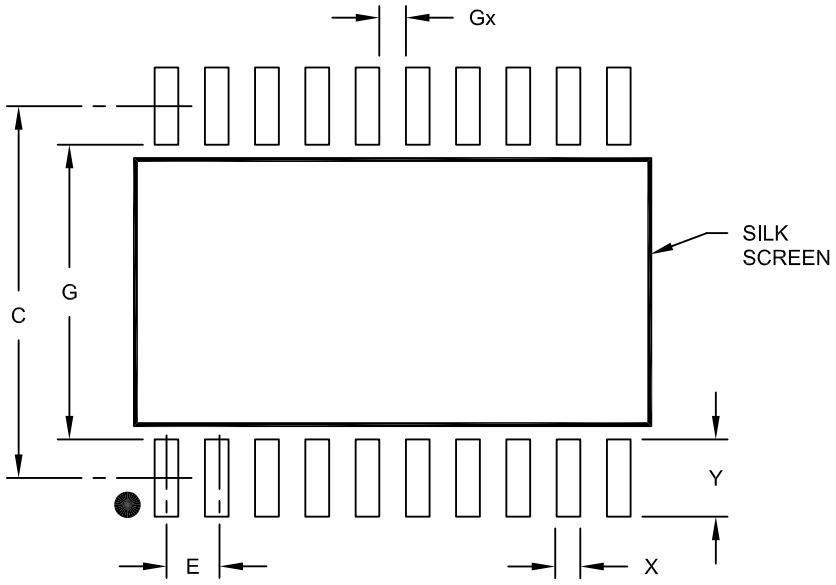
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-094C Sheet 2 of 2

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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		1.27 BSC	
Contact Pad Spacing	C		9.40	
Contact Pad Width (X20)	X			0.60
Contact Pad Length (X20)	Y			1.95
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.45		

Notes:

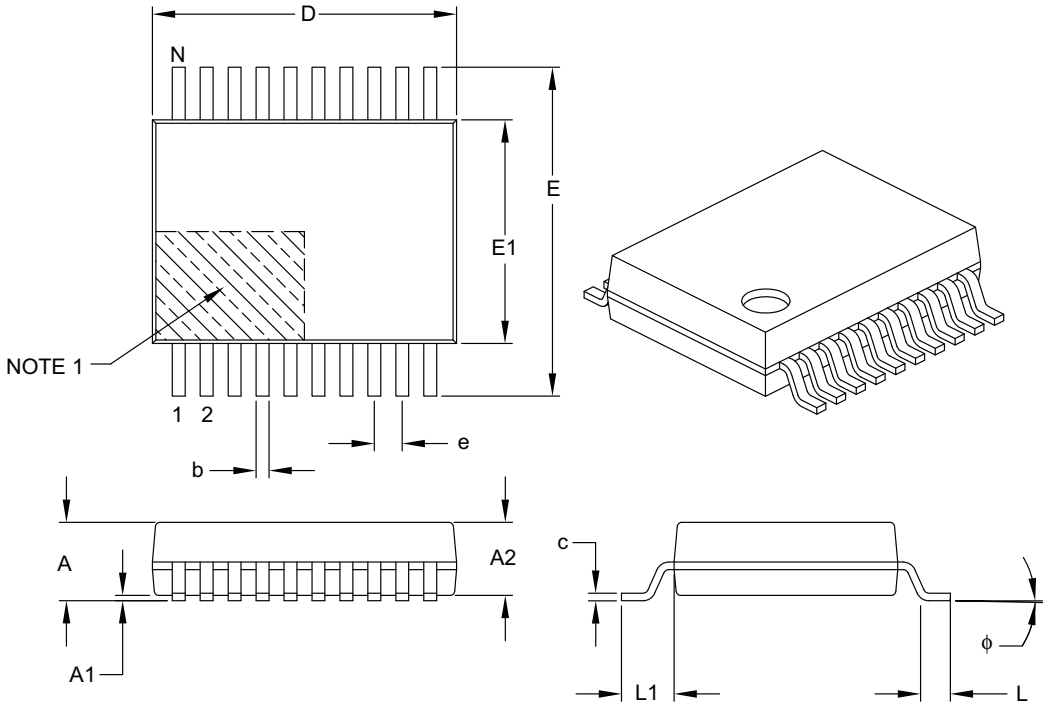
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2094A

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

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



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

Notes:

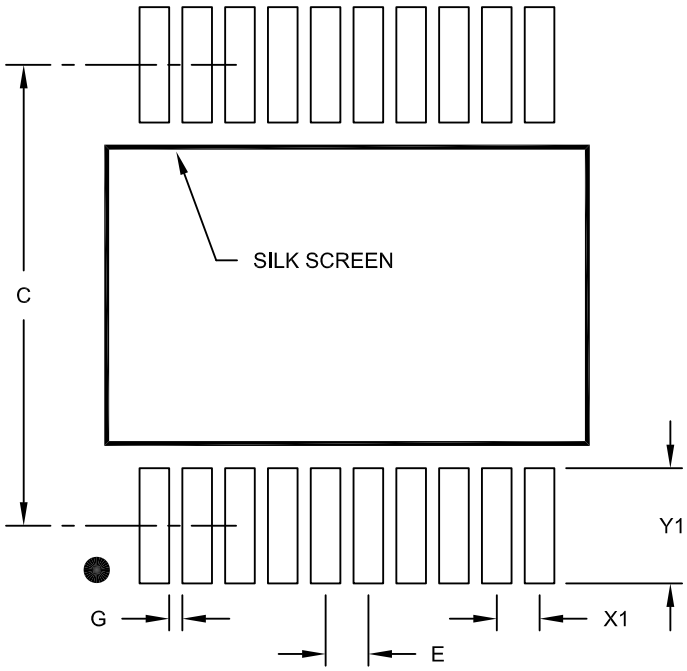
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-072B

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

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X20)	X1			0.45
Contact Pad Length (X20)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

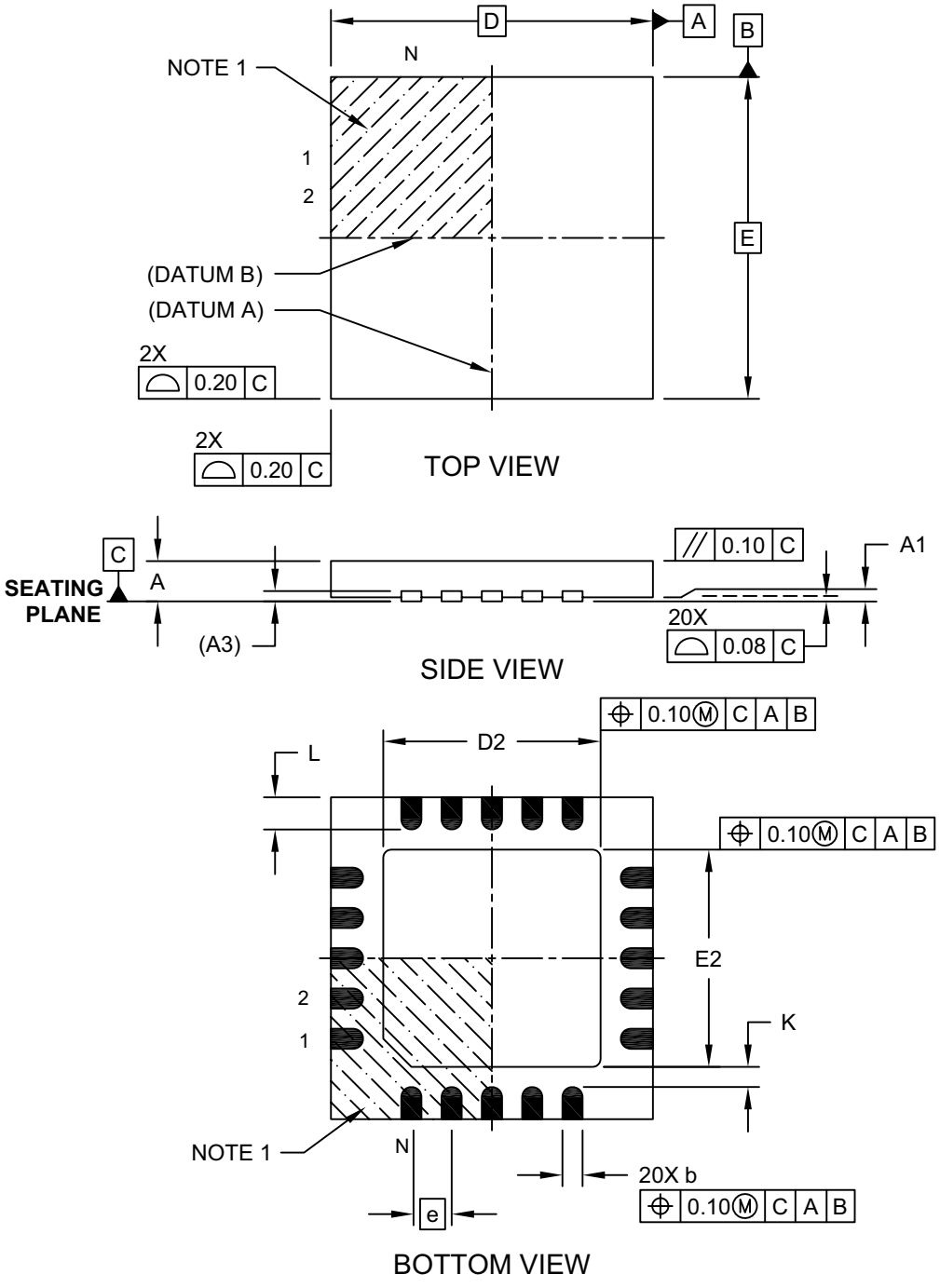
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072A

20-Lead Ultra Thin Plastic Quad Flat, No Lead Package (GZ) - 4x4x0.5 mm Body [UQFN]

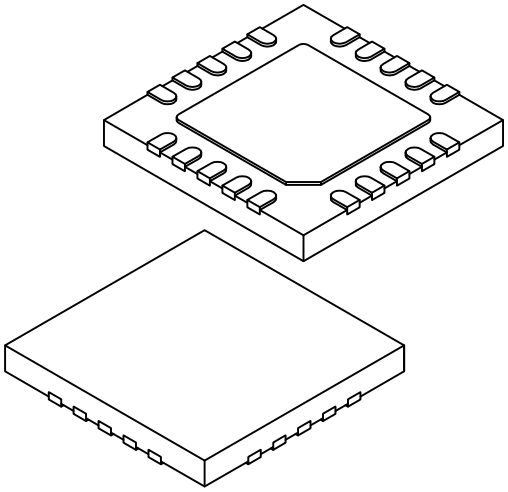
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-255A Sheet 1 of 2

20-Lead Ultra Thin Plastic Quad Flat, No Lead Package (GZ) - 4x4x0.5 mm Body [UQFN]

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 Terminals	N		20	
Pitch	e		0.50 BSC	
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.127 REF	
Overall Width	E		4.00 BSC	
Exposed Pad Width	E2	2.60	2.70	2.80
Overall Length	D		4.00 BSC	
Exposed Pad Length	D2	2.60	2.70	2.80
Terminal Width	b	0.20	0.25	0.30
Terminal Length	L	0.30	0.40	0.50
Terminal-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 is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

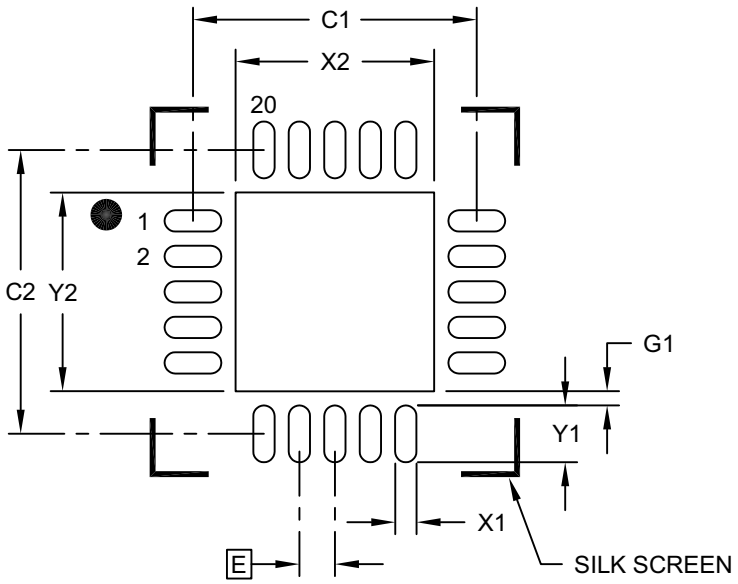
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-255A Sheet 2 of 2

20-Lead Ultra Thin Plastic Quad Flat, No Lead Package (GZ) - 4x4x0.5 mm Body [UQFN]

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	X2			2.80
Optional Center Pad Length	Y2			2.80
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X20)	X1			0.30
Contact Pad Length (X20)	Y1			0.80
Contact Pad to Center Pad (X20)	G1	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2255A

APPENDIX A: DATA SHEET REVISION HISTORY

Revision A (06/2015)

Initial release of the document.

Revision B (07/2015)

Updated the eXtreme Low-Power (XLP) Features section on page 1; Updated the Data Sheet Index in the Family Types table; Updated Sections 2.3, 3.2, 3.2.1 through 3.2.3, 3.5, 4.3.2, 5.2, 5.2.2, 5.2.3, 5.10, 6.1, 6.2.2, 6.2.2.3, 6.3, 6.4.4; 34.0 (Electrical Specifications); Removed note in Section 6.3.2; Updated Tables 1-2, 3-3, 5-1, 6-4, 19-1, 34-4, 34-7, Registers 4-1, 4-3, 4-4, 6-3, 6-4, and Figures 3-2, 34-5, 34-7; Removed Figure 3-8 (Indirect Addressing); Updated note 2 in Register 6-1 and Table 34-5; Updated notes in Register 6-2; Removed note 1 in Register 6-5 and note 2 in Register 6-2; Split table 6-1 in Tables 6-1 and 6-2; Changed data sheet status from Advance Information to Preliminary; Other minor corrections.

THE MICROCHIP WEB SITE

Microchip provides online support via our web 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://www.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:	PIC16F18325, PIC16LF18325, PIC16F18345, PIC16LF18345.	
Tape and Reel Option:	Blank T	= Standard packaging (tube or tray) = Tape and Reel(1)
Temperature Range:	I E	= -40°C to +85°C (Industrial) = -40°C to +125°C (Extended)
Package:(2)	JQ GZ P ST SL SO SS	= 16-lead UQFN (4x4) = 20-lead UQFN (4x4) = PDIP = TSSOP = 14-lead SOIC = 20-lead SOIC = SSOP
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)	

Examples:

a) PIC16LF18325- E/P
Extended temperature
PDIP package

b) PIC16LF18345- E/SO
Extended temperature,
SOIC 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.

2: Small form-factor packaging options may be available. Please check www.microchip.com/packaging for small-form factor package availability, or contact your local Sales Office.

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

ISBN: 978-1-63277-621-1

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: 852-2943-5100
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 - Dongguan
Tel: 86-769-8702-9880

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

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

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

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: 886-7-213-7828

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

01/27/15

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 8-bit Microcontrollers - MCU ***category:***

Click to view products by Microchip ***manufacturer:***

Other Similar products are found below :

009936B CY8C20524-12PVXIT CY8C28433-24PVXIT MB95F012KPFT-G-SNE2 MB95F013KPMC-G-SNE2 MB95F263KPF-G-SNE2 MB95F264KPFT-G-SNE2 MB95F398KPMC-G-SNE2 MB95F478KPMC2-G-SNE2 MB95F562KPF-G-SNE2 MB95F564KPF-G-SNE2 MB95F634KPMC-G-SNE2 MB95F636KWQN-G-SNE1 MB95F696KPMC-G-SNE2 MB95F698KPMC1-G-SNE2 MB95F698KPMC2-G-SNE2 MB95F698KPMC-G-SNE2 MB95F818KPMC1-G-SNE2 MC908JK1ECDWER MC9S08PA32AVLD MC9S08PT60AVLD R5F1076CMSPV0 R5F5631ECDFBV0 C8051F389-B-GQ C8051F392-A-GMR ISD-ES1600_USB_PROG 901015X S9S08SL8F1CTJR STM8TL53G4U6 PIC16F877-04/P-B R5F10Y17ASP#30 CY8C3MFIDOCK-125 403708R MB95F354EPF-G-SNE2 MB95F564KPFT-G-SNE2 MB95F564KWQN-G-SNE1 MB95F636KP-G-SH-SNE2 MB95F636KPMC-G-SNE2 MB95F694KPMC-G-SNE2 MB95F778JPMC1-G-SNE2 MB95F818KPMC-G-SNE2 MC908QY8CDWER MC9S08PT16AVLD MC9S08PT32AVLH MC9S08PT60AVLC MC9S08PT60AVLH C8051F500-IQR 400801H LC87F0G08AUJA-AH 026923G