

Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	12
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 11x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-UQFN Exposed Pad
Supplier Device Package	16-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f18324-i-jq

4.1.1.2 Indirect Read with FSR

The program memory can be accessed as data by setting bit 7 of an 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 read the program memory via the FSR require one extra instruction cycle to complete. Example 4-2 demonstrates reading 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 4-2: 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
```

4.1.1.3 NVMREG Access

The NVMREG interface allows read/write access to all locations accessible by the FSRs, User ID locations, and EEPROM. The NVMREG interface also provides read-only access to Device ID, Revision ID, and Configuration data. See **Section 11.4 “NVMREG Access”** for more information.

4.2 Data Memory Organization

The data memory is partitioned into 32 memory banks with 128 bytes in each bank. Each bank consists of (Figure 4-2):

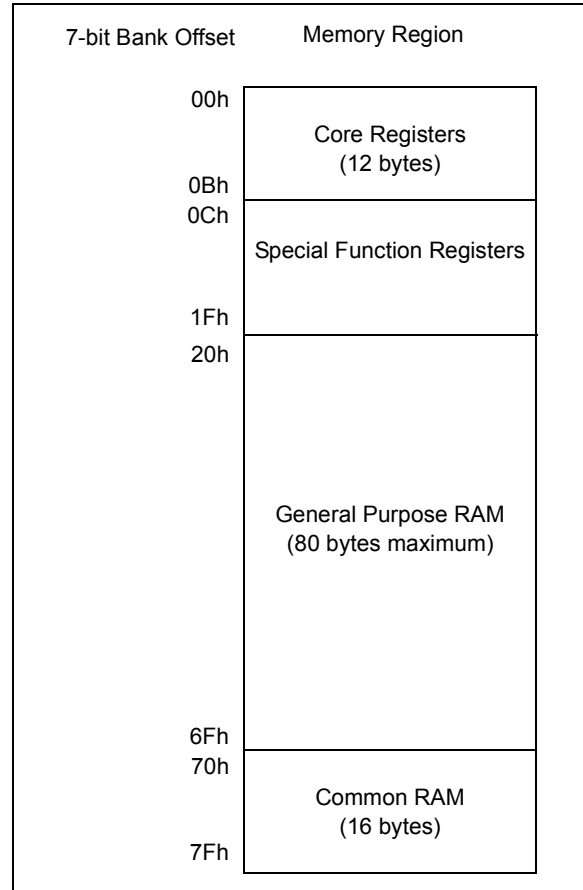
- 12 core registers
- Special Function Registers (SFR)
- Up to 80 bytes of General Purpose RAM (GPR)
- 16 bytes of common RAM

4.2.1 BANK SELECTION

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 4.5 “Indirect Addressing”** for more information.

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

FIGURE 4-2: BANKED MEMORY PARTITIONING



4.2.2 CORE REGISTERS

The core registers contain the registers that directly affect 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 4-2. For detailed information, see Table 4-4.

TABLE 4-2: CORE REGISTERS

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

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

Address	Name	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
Bank 7													
CPU CORE REGISTERS; see Table 4-2 for specifics													
38Ch	INLVLA			—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	--11 1111	--11 1111
38Dh	INVLVB	X	—	Unimplemented								—	—
		—	X	INVLVB7	INVLVB6	INVLVB5	INVLVB4	—	—	—	—	1111 ----	1111 ----
38Eh	INLVLC	X	—	—	—	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	--11 1111	--11 1111
		—	X	INLVLC7	INLVLC6	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	1111 1111	1111 1111
38Fh	—	—	—	Unimplemented								—	—
390h	—	—	—	Unimplemented								—	—
391h	IOCAP			—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	--00 0000	--00 0000
392h	IOCAN			—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	--00 0000	--00 0000
393h	IOCAF			—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	--00 0000	--00 0000
394h	IOCBP	X	—	Unimplemented								—	—
		—	X	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	0000 ----	0000 ----
395h	IOCBN	X	—	Unimplemented								—	—
		—	X	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	0000 ----	0000 ----
396h	IOCBF	X	—	Unimplemented								—	—
		—	X	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	0000 ----	0000 ----
397h	IOCCP	X	—	—	—	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	--00 0000	--00 0000
		—	X	IOCCP7	IOCCP6	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	0000 0000	0000 0000
398h	IOCCN	X	—	—	—	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	--00 0000	--00 0000
		—	X	IOCCN7	IOCCN6	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	0000 0000	0000 0000
399h	IOCCF	X	—	—	—	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	--00 0000	--00 0000
		—	X	IOCCF7	IOCCF6	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	0000 0000	0000 0000

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 PIC16F18324/18344.

Note 2: Register accessible from both User and ICD Debugger.

4.4 Stack

All devices have a 16-level x 15-bit wide hardware stack (refer to Figure 4-4 through Figure 4-7). 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 and does not cause a Reset when either a Stack Overflow or Underflow occur 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.

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.

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.

4.4.1 ACCESSING THE STACK

The stack is accessible 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 read to see how many levels remain available on the stack. 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 write the PC and then decrement the `STKPTR`.

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

5.3 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.

5.3.1 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 5.4 "Write Protection"** for more information.

5.3.2 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.

5.4 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.

5.5 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 11.4.7 "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).

5.6 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 11.4.7 "NVMREG EEPROM, User ID, Device ID and Configuration Word Access"** for more information on accessing these memory locations.

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

PIC16(L)F18324/18344

REGISTER 7-7: 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

q = Reset value is determined by hardware

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 7-3: 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>				87
OSCCON2	—	COSC<2:0>			CDIV<3:0>				87
OSCCON3	CWSHOLD	SOSCPWR	SOSCBE	ORDY	NOSCR	—	—	—	89
OSCSTAT1	EXTOR	HFOR	—	LFOR	SOR	ADOR	—	PLLR	90
OSCEN	EXTOEN	HFOEN	—	LFOEN	SOSCEN	ADOEN	—	—	91
OSCFRQ	—	—	—	—	HFFRQ<3:0>				92
OSCTUNE	—	—	HFTUN<5:0>						93

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

TABLE 7-4: 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	—	—	CLKOUTEN	62
	7:0	—	RSTOSC2	RSTOSC1	RSTOSC0	—	FEXTOSC2	FEXTOSC1	FEXTOSC0	

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

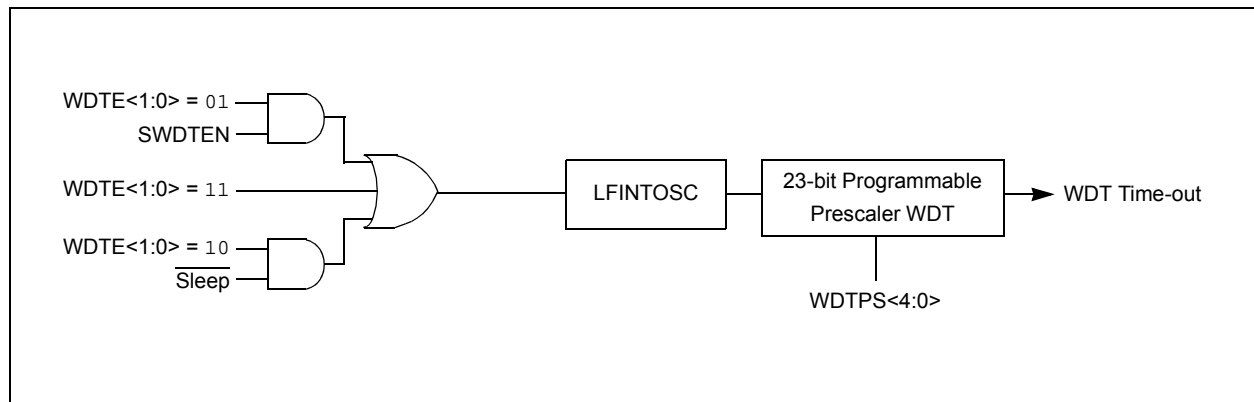
10.0 WATCHDOG TIMER (WDT)

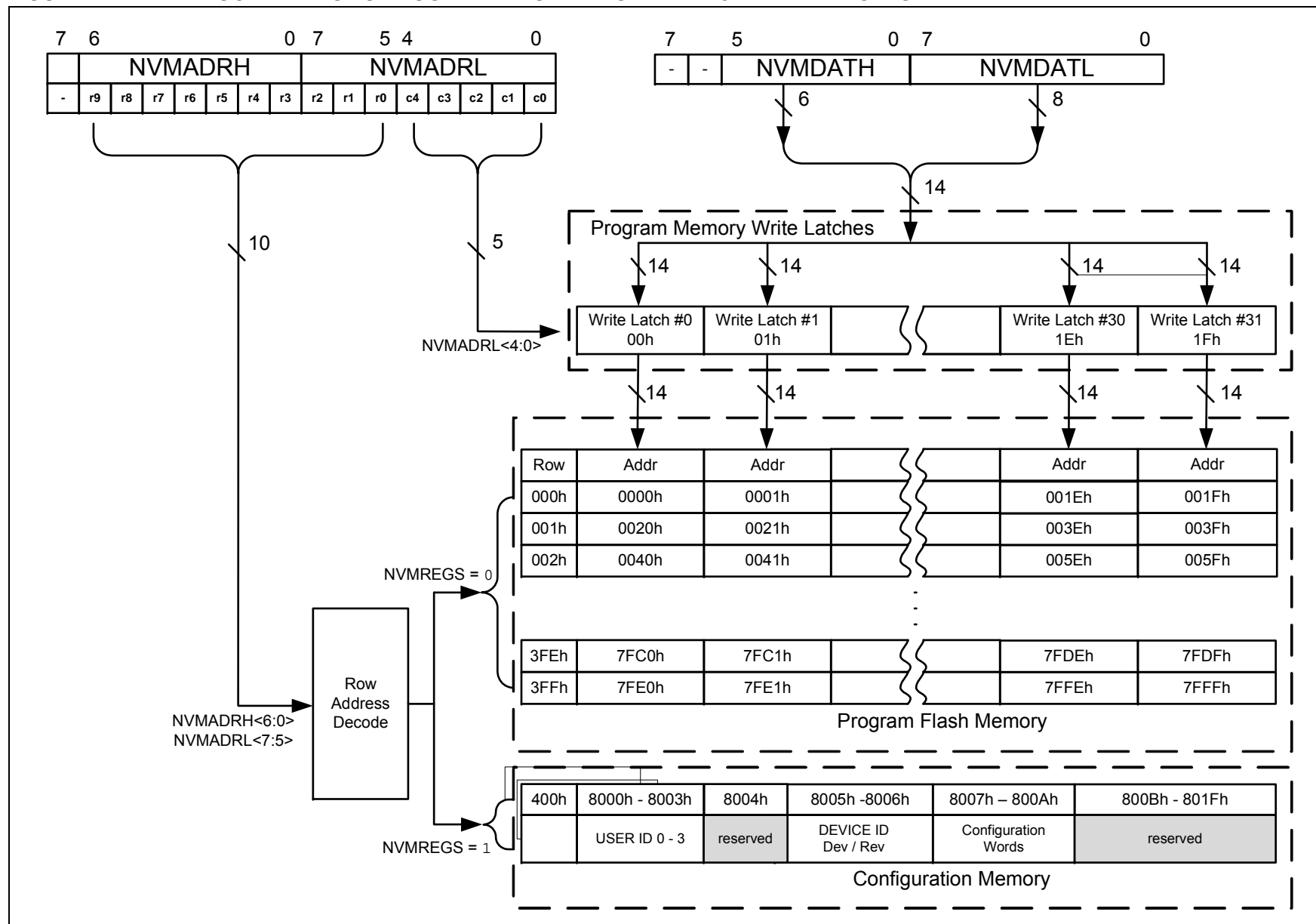
The Watchdog Timer is a system timer that generates a Reset if the firmware does not issue a `CLRWDT` instruction within the time-out period. The Watchdog Timer is typically used to recover the system from unexpected events.

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 WDT clearing conditions
- Operation during Sleep

FIGURE 10-1: WATCHDOG TIMER BLOCK DIAGRAM





REGISTER 12-3: 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 12-4: ANSALA: 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.

18.9 Analog Input Connection Considerations

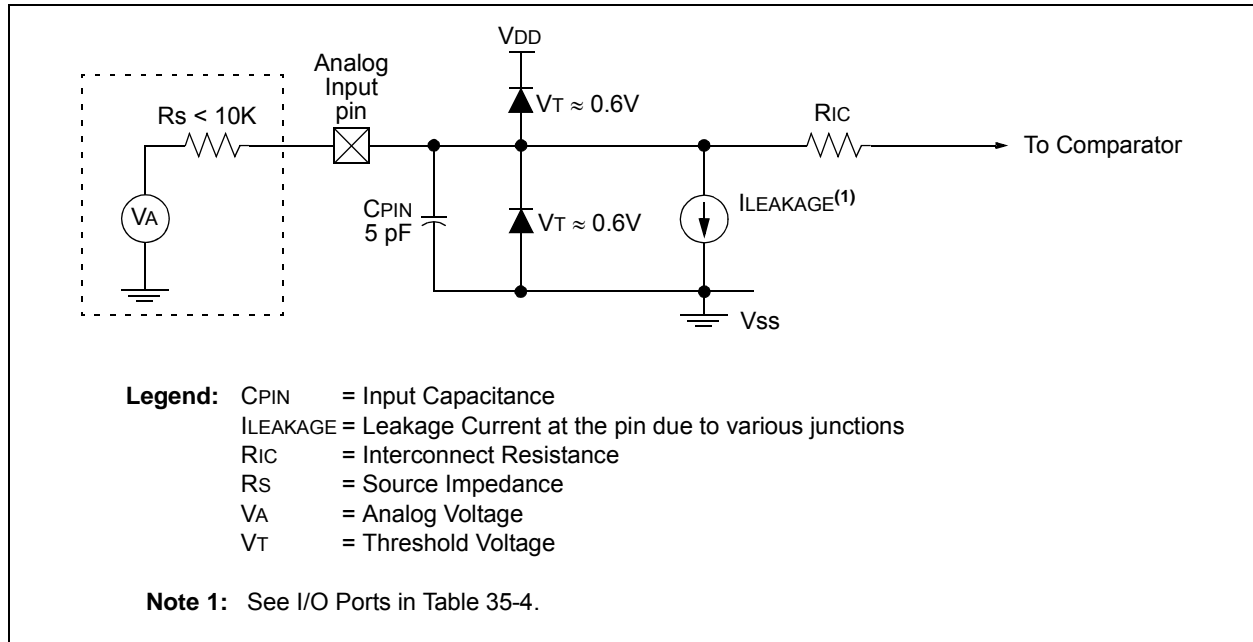
A simplified circuit for an analog input is shown in Figure 18-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to V_{DD} and V_{SS} . The analog input, therefore, must be between V_{SS} and V_{DD} . If the input voltage deviates 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 k Ω 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 provide an input based on their level as either a TTL or ST input buffer.

2: Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

FIGURE 18-3: ANALOG INPUT MODEL



18.10 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 20.7.1.2 “External Input Source Shutdown”**).

18.11 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.

TABLE 21-3: SUMMARY OF REGISTERS ASSOCIATED WITH CLCx (CONTINUED)

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Register on Page
CLC2GLS0	LC2G1D4T	LC2G1D4N	LC2G1D3T	LC2G1D3N	LC2G1D2T	LC2G1D2N	LC2G1D1T	LC2G1D1N	227
CLC2GLS1	LC2G2D4T	LC2G2D4N	LC2G2D3T	LC2G2D3N	LC2G2D2T	LC2G2D2N	LC2G2D1T	LC2G2D1N	228
CLC2GLS2	LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1N	229
CLC2GLS3	LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1N	230
CLC3CON	LC3EN	—	LC3OUT	LC3INTP	LC3INTN	LC3MODE<2:0>			224
CLC3POL	LC3POL	—	—	—	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1POL	225
CLC3SEL0	—	—	LC3D1S<5:0>						226
CLC3SEL1	—	—	LC3D2S<5:0>						226
CLC3SEL2	—	—	LC3D3S<5:0>						226
CLC3SEL3	—	—	LC3D4S<5:0>						227
CLC3GLS0	LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1N	227
CLC3GLS1	LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N	228
CLC3GLS2	LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N	229
CLC3GLS3	LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N	230
CLC4CON	LC4EN	—	LC4OUT	LC4INTP	LC4INTN	LC4MODE<2:0>			224
CLC4POL	LC4POL	—	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL	225
CLC4SEL0	—	—	LC4D1S<5:0>						226
CLC4SEL1	—	—	LC4D2S<5:0>						226
CLC4SEL2	—	—	LC4D3S<5:0>						226
CLC4SEL3	—	—	LC4D4S<5:0>						227
CLC4GLS0	LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N	227
CLC4GLS1	LC4G2D4T	LC4G2D4N	LC4G2D3T	LC4G2D3N	LC4G2D2T	LC4G2D2N	LC4G2D1T	LC4G2D1N	228
CLC4GLS2	LC4G3D4T	LC4G3D4N	LC4G3D3T	LC4G3D3N	LC4G3D2T	LC4G3D2N	LC4G3D1T	LC4G3D1N	229
CLC4GLS3	LC4G4D4T	LC4G4D4N	LC4G4D3T	LC4G4D3N	LC4G4D2T	LC4G4D2N	LC4G4D1T	LC4G4D1N	230
CLCDATA	—	—	—	—	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT	231
CLCIN0PPS	—	—	—	CLCIN0PPS<4:0>					159
CLCIN1PPS	—	—	—	CLCIN1PPS<4:0>					159
CLCIN2PPS	—	—	—	CLCIN2PPS<4:0>					159
CLCIN3PPS	—	—	—	CLCIN3PPS<4:0>					159
CLC1OUTPPS	—	—	—	CLC1OUTPPS<4:0>					159
CLC2OUTPPS	—	—	—	CLC2OUTPPS<4:0>					159
CLC3OUTPPS	—	—	—	CLC3OUTPPS<4:0>					159
CLC4OUTPPS	—	—	—	CLC4OUTPPS<4:0>					159

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the CLC module.

30.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP1) MODULE

30.1 MSSP1 Module Overview

The Master Synchronous Serial Port (MSSP) 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 MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I²C)

The SPI interface supports the following modes and features:

- Master mode
- Slave mode
- Clock Polarity
- Slave Select Synchronization (Slave mode only)
- Daisy-chain connection of slave devices

Figure 30-1 is a block diagram of the SPI interface module.

FIGURE 30-1: MSSP BLOCK DIAGRAM (SPI MODE)

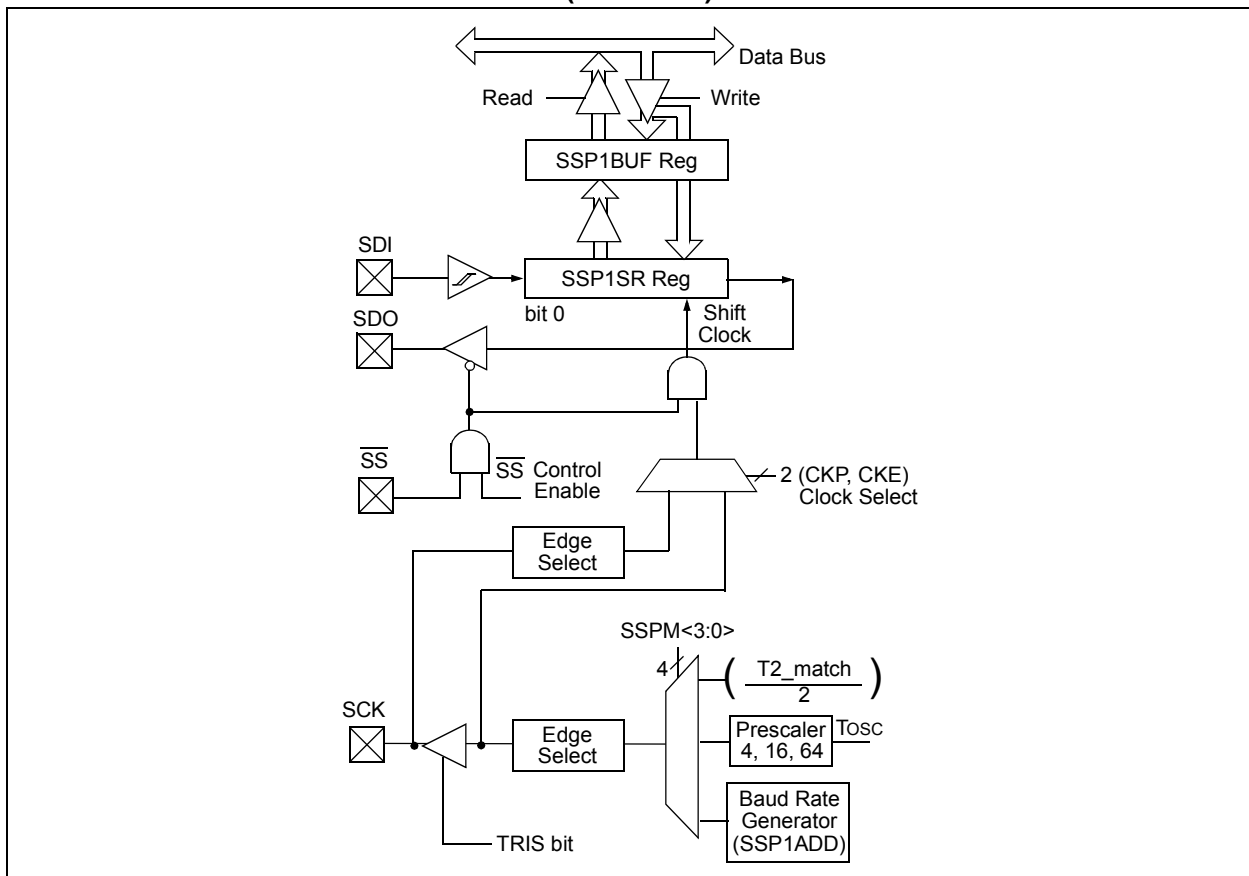


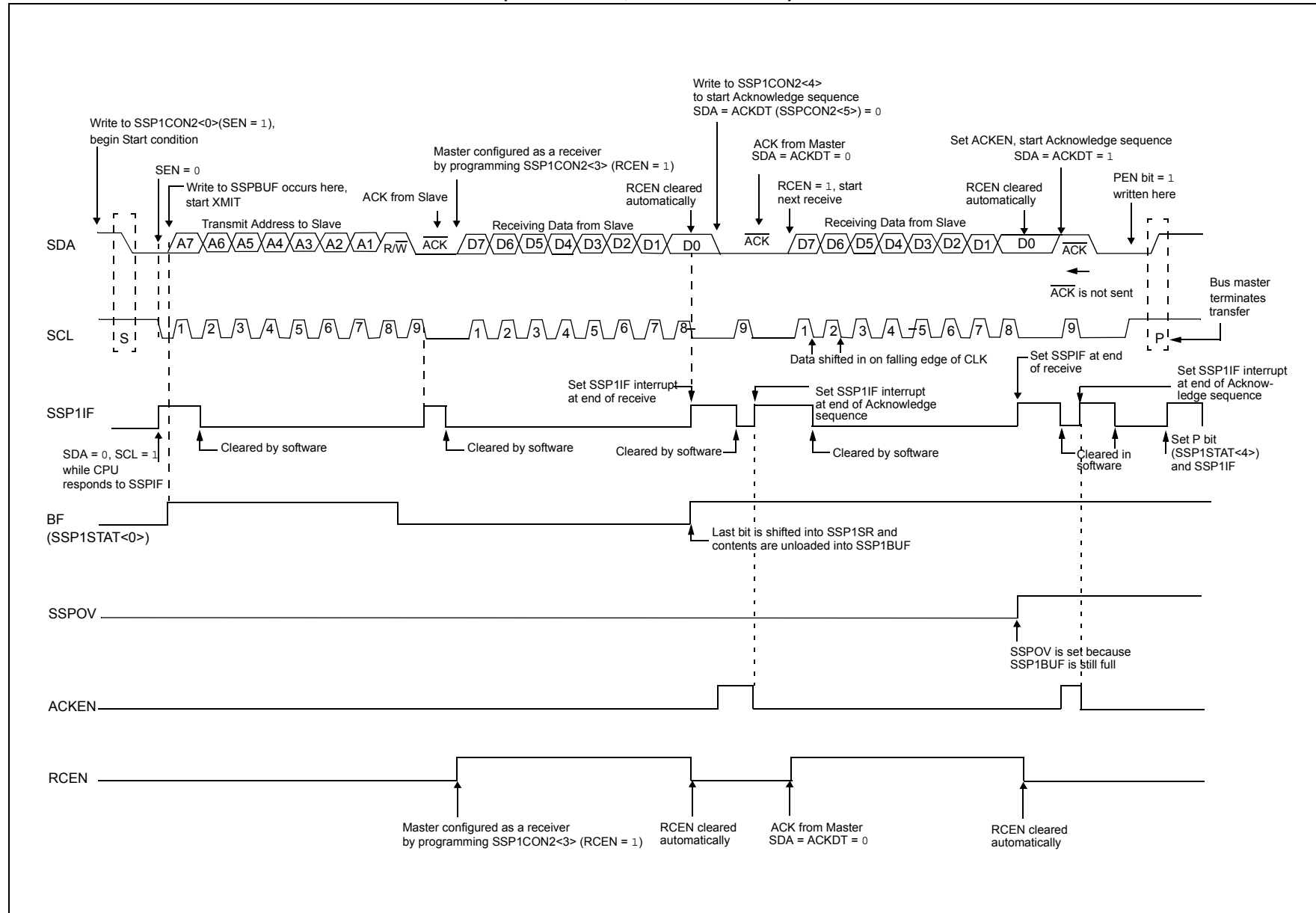
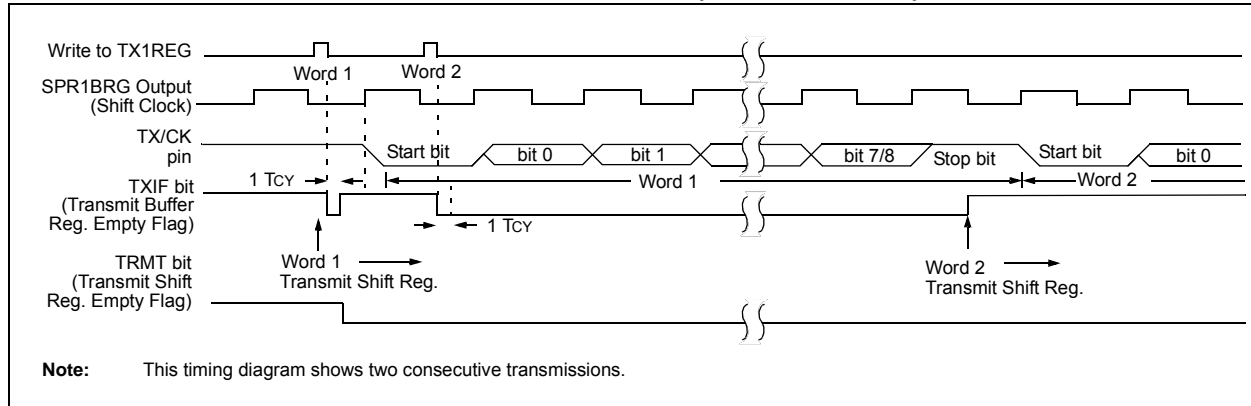
FIGURE 30-29: I²C MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)

FIGURE 31-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)



31.1.2 EUSART1 ASYNCHRONOUS RECEIVER

The Asynchronous mode is typically used in RS-232 systems. The receiver block diagram is shown in Figure 31-2. 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.

31.1.2.1 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.

31.1.2.2 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 31.1.2.4 "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 31.1.2.5 "Receive Overrun Error"** for more information on overrun errors.

31.4.2 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.

31.4.2.1 EUSART1 Synchronous Slave Transmit

The operation of the Synchronous Master and Slave modes are identical (see **Section 31.4.1.3 “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.

31.4.2.2 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.

31.4.2.3 EUSART1 Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (**Section 31.4.1.5 “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.

31.4.2.4 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.

TABLE 32-1: 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	140
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	146
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	153
CLKRCON	CLKREN	—	—	CLKRDC<1:0>		CLKRDIV<2:0>			224
CLCxSEly	—	—	LCxDyS<5:0>						226
MDCARH	—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>				270
MDCARL	—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				271

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

Note 1: PIC16(L)F18344 only.

2: Unimplemented, read as '1'.

34.0 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 34-3 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 entry 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.

34.1 Read-Modify-Write Operations

Any write 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 working (W) register, or the originating file register, depending on the state of the destination designator 'd' (see Table 34-1 for more information). A read operation is performed on a register even if the instruction writes to that register.

TABLE 34-1: 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.
n	FSR or INDF number. (0-1)
mm	Pre-post increment-decrement mode selection

TABLE 34-2: ABBREVIATION DESCRIPTIONS

Field	Description
PC	Program Counter
TO	Time-Out bit
C	Carry bit
DC	Digit Carry bit
Z	Zero bit
PD	Power-Down bit

35.0 ELECTRICAL SPECIFICATIONS

35.1 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	
PIC16F18324/18344	-0.3V to +6.5V
PIC16LF18324/18344	-0.3V to +4.0V
on 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 35-3 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.

35.3 DC Characteristics

TABLE 35-1: SUPPLY VOLTAGE

PIC16LF18324/18344			Standard Operating Conditions (unless otherwise stated)				
PIC16F18324/18344			Standard Operating Conditions (unless otherwise stated)				
Param. No.	Sym.	Characteristic	Min.	Typ.†	Max.	Units	Conditions
Supply Voltage							
D002	VDD		1.8	—	3.6	V	FOSC ≤ 16 MHz
			2.5	—	3.6	V	FOSC > 16 MHz
D002	VDD		2.3	—	5.5	V	FOSC ≤ 16 MHz:
			2.5	—	5.5	V	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 35-3.
- 3:** See Table 35-11 for BOR and LPBOR trip point information.

FIGURE 35-21: I²C BUS START/STOP BITS TIMING

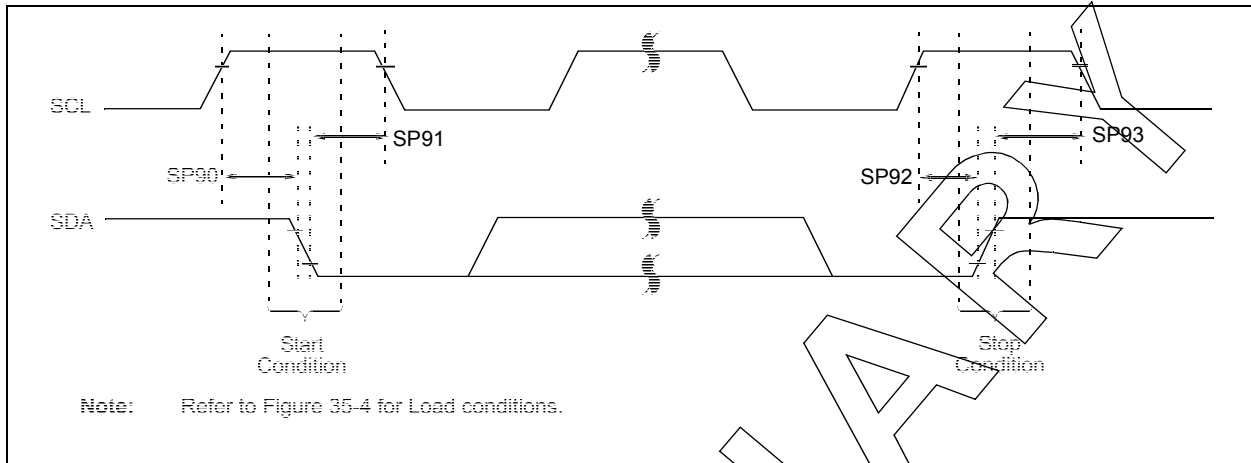


TABLE 35-23: I²C BUS START/STOP BITS CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)								
Param. No.	Symbol	Characteristic		Min.	Typ.	Max.	Units	Conditions
SP90*	TSU:STA	Start condition	100 kHz mode	4700	—	—	ns	Only relevant for Repeated Start condition
		Setup time	400 kHz mode	600	—	—		
SP91*	THD:STA	Start condition	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
		Hold time	400 kHz mode	600	—	—		
SP92*	TSU:STO	Stop condition	100 kHz mode	4700	—	—	ns	
		Setup time	400 kHz mode	600	—	—		
SP93	THD:STO	Stop condition	100 kHz mode	4000	—	—	ns	
		Hold time	400 kHz mode	600	—	—		

* These parameters are characterized but not tested.

FIGURE 35-22: I²C BUS DATA TIMING

