



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	5
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 4x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	8-VDFN Exposed Pad
Supplier Device Package	8-DFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic12f1840-i-mf

FIGURE 1: 8-PIN DIAGRAM FOR PIC12(L)F1840

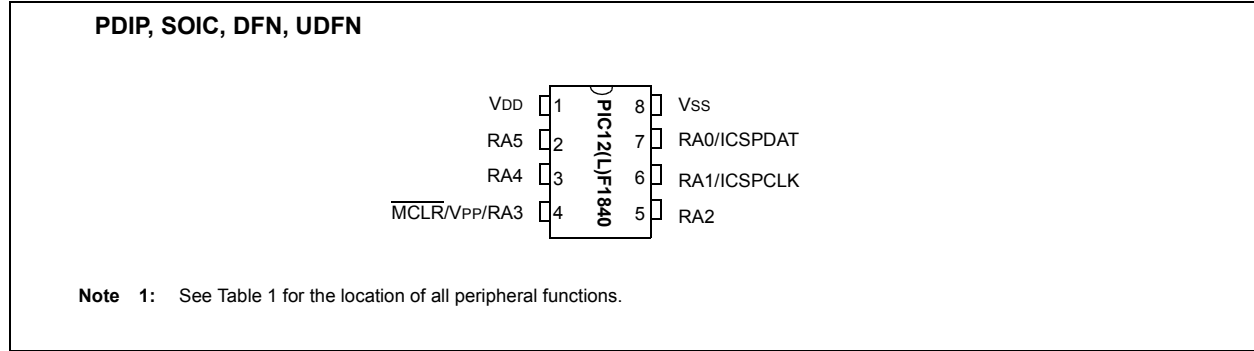


TABLE 1: 8-PIN ALLOCATION TABLE (PIC12(L)F1840)

I/O	8-Pin PDIP/SOIC/DFN/UDFN	ADC	Reference	Cap Sense	Comparator	SR Latch	Timers	ECCP	EUSART	MSSP	Interrupt	Modulator	Pull-up	Basic
RA0	7	AN0	DACOUT	CPS0	C1IN+	—	—	P1B	TX CK	SDO SS ⁽¹⁾	IOC	MDOUT	Y	ICSPDAT ICDDAT
RA1	6	AN1	VREF	CPS1	C1IN0-	SRI	—	—	RX DT	SCL SCK	IOC	MDMIN	Y	ICSPCLK ICPCLK
RA2	5	AN2	—	CPS2	C1OUT	SRQ	T0CKI	CCP1 P1A FLT0	—	SDA SDI	INT/ IOC	MDCIN1	Y	—
RA3	4	—	—	—	—	—	T1G ⁽¹⁾	—	—	SS	IOC	—	Y	MCLR VPP
RA4	3	AN3	—	CPS3	C1IN1-	—	T1G T1OSO	P1B ⁽¹⁾	TX ⁽¹⁾ CK ⁽¹⁾	SDO ⁽¹⁾	IOC	MDCIN2	Y	OSC2 CLKOUT CLKR
RA5	2	—	—	—	—	SRNQ	T1CKI T1OSI	CCP1 ⁽¹⁾ P1A ⁽¹⁾	RX ⁽¹⁾ DT ⁽¹⁾	—	IOC	—	Y	OSC1 CLKIN
VDD	1	—	—	—	—	—	—	—	—	—	—	—	—	VDD
VSS	8	—	—	—	—	—	—	—	—	—	—	—	—	VSS

Note 1: Alternate pin function selected with the APFCON (Register 12-1) register.

5.2.2.5 Internal Oscillator Frequency Selection

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register.

The output of the 16 MHz HFINTOSC postscaler and LFINTOSC connects to a multiplexer (see Figure 5-1). The Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register select the frequency output of the internal oscillators. One of the following frequencies can be selected via software:

- 32 MHz (requires 4x PLL)
- 16 MHz
- 8 MHz
- 4 MHz
- 2 MHz
- 1 MHz
- 500 kHz (default after Reset)
- 250 kHz
- 125 kHz
- 62.5 kHz
- 31.25 kHz
- 31 kHz (LFINTOSC)

Note: Following any Reset, the IRCF<3:0> bits of the OSCCON register are set to '0111' and the frequency selection is set to 500 kHz. The user can modify the IRCF bits to select a different frequency.

The IRCF<3:0> bits of the OSCCON register allow duplicate selections for some frequencies. These duplicate choices can offer system design trade-offs. Lower power consumption can be obtained when changing oscillator sources for a given frequency. Faster transition times can be obtained between frequency changes that use the same oscillator source.

5.2.2.6 32 MHz Internal Oscillator Frequency Selection

The Internal Oscillator Block can be used with the 4x PLL associated with the External Oscillator Block to produce a 32 MHz internal system clock source. The following settings are required to use the 32 MHz internal clock source:

- The FOSC bits in Configuration Words must be set to use the INTOSC source as the device system clock (FOSC<2:0> = 100).
- The SCS bits in the OSCCON register must be cleared to use the clock determined by FOSC<2:0> in Configuration Words (SCS<1:0> = 00).
- The IRCF bits in the OSCCON register must be set to the 8 MHz HFINTOSC set to use (IRCF<3:0> = 1110).
- The SPLLEN bit in the OSCCON register must be set to enable the 4xPLL, or the PLEN bit of the Configuration Words must be programmed to a '1'.

Note: When using the PLEN bit of the Configuration Words, the 4x PLL cannot be disabled by software and the 8 MHz HFINTOSC option will no longer be available.

The 4x PLL is not available for use with the internal oscillator when the SCS bits of the OSCCON register are set to '1x'. The SCS bits must be set to '00' to use the 4x PLL with the internal oscillator.

5.2.2.7 Internal Oscillator Clock Switch Timing

When switching between the HFINTOSC, MFINTOSC and the LFINTOSC, the new oscillator may already be shut down to save power (see Figure 5-7). If this is the case, there is a delay after the IRCF<3:0> bits of the OSCCON register are modified before the frequency selection takes place. The OSCSTAT register will reflect the current active status of the HFINTOSC, MFINTOSC and LFINTOSC oscillators. The sequence of a frequency selection is as follows:

1. IRCF<3:0> bits of the OSCCON register are modified.
2. If the new clock is shut down, a clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. The current clock is held low and the clock switch circuitry waits for a rising edge in the new clock.
5. The new clock is now active.
6. The OSCSTAT register is updated as required.
7. Clock switch is complete.

See Figure 5-7 for more details.

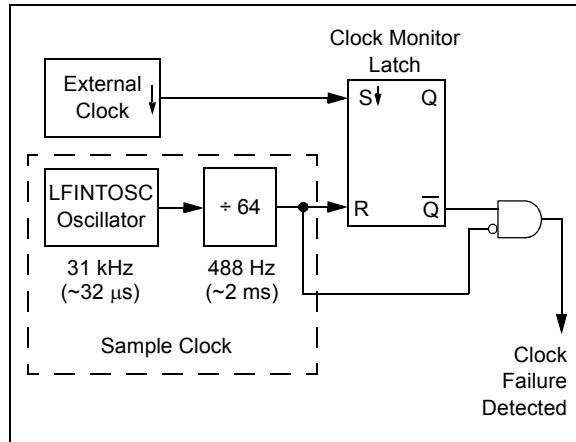
If the internal oscillator speed is switched between two clocks of the same source, there is no start-up delay before the new frequency is selected. Clock switching time delays are shown in Table 5-1.

Start-up delay specifications are located in the oscillator tables of **Section 30.0 “Electrical Specifications”**

5.5 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM can detect oscillator failure any time after the Oscillator Start-up Timer (OST) has expired. 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, Timer1 Oscillator and RC).

FIGURE 5-9: FSCM BLOCK DIAGRAM



5.5.1 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 5-9. 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.

5.5.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM switches the device clock to an internal clock source and sets the bit flag OSFIF of the PIR2 register. Setting this flag will generate an interrupt if the OSFIE bit of the PIE2 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.

The internal clock source chosen by the FSCM is determined by the IRCF<3:0> bits of the OSCCON register. This allows the internal oscillator to be configured before a failure occurs.

5.5.3 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared after a Reset, executing a *SLEEP* instruction or changing the SCS bits of the OSCCON register. When the SCS bits are changed, the OST is restarted. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON. 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.

5.5.4 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 or RC Clock modes so that the FSCM will be active as soon as the Reset or wake-up has completed. When the FSCM is enabled, the Two-Speed Start-up is also enabled. Therefore, the device will always be executing code while the OST is operating.

Note: Due to the wide range of oscillator start-up times, the Fail-Safe circuit is not active during oscillator start-up (i.e., after exiting Reset or Sleep). After an appropriate amount of time, the user should check the Status bits in the OSCSTAT register to verify the oscillator start-up and that the system clock switchover has successfully completed.

10.6 Register Definitions: Watchdog Control

REGISTER 10-1: WDTCON: WATCHDOG TIMER CONTROL REGISTER

U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-1/1	R/W-0/0
—	—	WDTPS<4:0>					SWDTEN
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-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

Bit Value = Prescale Rate

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

•
•
•

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

10010 = 1:8388608 (2^{23}) (Interval 256s nominal)

10001 = 1:4194304 (2^{22}) (Interval 128s nominal)

10000 = 1:2097152 (2^{21}) (Interval 64s nominal)

01111 = 1:1048576 (2^{20}) (Interval 32s nominal)

01110 = 1:524288 (2^{19}) (Interval 16s nominal)

01101 = 1:262144 (2^{18}) (Interval 8s nominal)

01100 = 1:131072 (2^{17}) (Interval 4s nominal)

01011 = 1:65536 (Interval 2s nominal) (Reset value)

01010 = 1:32768 (Interval 1s nominal)

01001 = 1:16384 (Interval 512 ms nominal)

01000 = 1:8192 (Interval 256 ms nominal)

00111 = 1:4096 (Interval 128 ms nominal)

00110 = 1:2048 (Interval 64 ms nominal)

00101 = 1:1024 (Interval 32 ms nominal)

00100 = 1:512 (Interval 16 ms nominal)

00011 = 1:256 (Interval 8 ms nominal)

00010 = 1:128 (Interval 4 ms nominal)

00001 = 1:64 (Interval 2 ms nominal)

00000 = 1:32 (Interval 1 ms nominal)

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

If WDTE<1:0> = 00:

This bit is ignored.

If WDTE<1:0> = 01:

1 = WDT is turned on

0 = WDT is turned off

If WDTE<1:0> = 1x:

This bit is ignored.

11.2 Using the Data EEPROM

The data EEPROM is a high-endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). When variables in one section change frequently, while variables in another section do not change, it is possible to exceed the total number of write cycles to the EEPROM without exceeding the total number of write cycles to a single byte. Refer to **Section 30.0 “Electrical Specifications”**. If this is the case, then a refresh of the array must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.

11.2.1 READING THE DATA EEPROM MEMORY

To read a data memory location, the user must write the address to the EEADRL register, clear the EEPGD and CFGS control bits of the EECON1 register, and then set control bit RD. The data is available at the very next cycle, in the EEDATL register; therefore, it can be read in the next instruction. EEDATL will hold this value until another read or until it is written to by the user (during a write operation).

EXAMPLE 11-1: DATA EEPROM READ

```
BANKSEL EEADRL      ;
MOVLW  DATA_EE_ADDR ;
MOVWF  EEADRL        ;Data Memory
                        ;Address to read
BCF     EECON1, CFGS ;Deselect Config space
BCF     EECON1, EEPGD;Point to DATA memory
BSF     EECON1, RD    ;EE Read
MOVF    EEDATL, W     ;W = EEDATL
```

Note: Data EEPROM can be read regardless of the setting of the CPD bit.

11.2.2 WRITING TO THE DATA EEPROM MEMORY

To write an EEPROM data location, the user must first write the address to the EEADRL register and the data to the EEDATL register. Then the user must follow a specific sequence to initiate the write for each byte.

The write will not initiate if the above sequence is not followed exactly (write 55h to EECON2, write AAh to EECON2, then set the WR bit) for each byte. Interrupts should be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

11.2.3 PROTECTION AGAINST SPURIOUS WRITE

There are conditions when the user may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built-in. On power-up, WREN is cleared. Also, the Power-up Timer (64 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during:

- Brown-out
- Power Glitch
- Software Malfunction

11.2.4 DATA EEPROM OPERATION DURING CODE-PROTECT

Data memory can be code-protected by programming the CPD bit in the Configuration Words to '0'.

When the data memory is code-protected, only the CPU is able to read and write data to the data EEPROM. It is recommended to code-protect the program memory when code-protecting data memory. This prevents anyone from replacing your program with a program that will access the contents of the data EEPROM.

EXAMPLE 11-3: FLASH PROGRAM MEMORY READ

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

  BANKSEL  EEADRL          ; Select Bank for EEPROM registers
  MOVLW    PROG_ADDR_LO    ;
  MOVWF    EEADRL          ; Store LSB of address
  MOVLW    PROG_ADDR_HI    ;
  MOVWL    EEADRH          ; Store MSB of address

  BCF      EECON1,CFGSS     ; Do not select Configuration Space
  BSF      EECON1,EEPGD     ; Select Program Memory
  BCF      INTCON,GIE       ; Disable interrupts
  BSF      EECON1,RD        ; Initiate read
  NOP                      ; Executed (Figure 11-1)
  NOP                      ; Ignored (Figure 11-1)
  BSF      INTCON,GIE       ; Restore interrupts

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


13.6 Register Definitions: Interrupt-on-Change Control

REGISTER 13-1: 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. Associated Status bit and interrupt flag will be set upon detecting an edge.
- 0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 13-2: 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. Associated Status bit and interrupt flag will be set upon detecting an edge.
- 0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 13-3: 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.

23.0 DATA SIGNAL MODULATOR

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 23-1 shows a Simplified Block Diagram of the Data Signal Modulator peripheral.

PIC12(L)F1840

FIGURE 23-1: SIMPLIFIED BLOCK DIAGRAM OF THE DATA SIGNAL MODULATOR

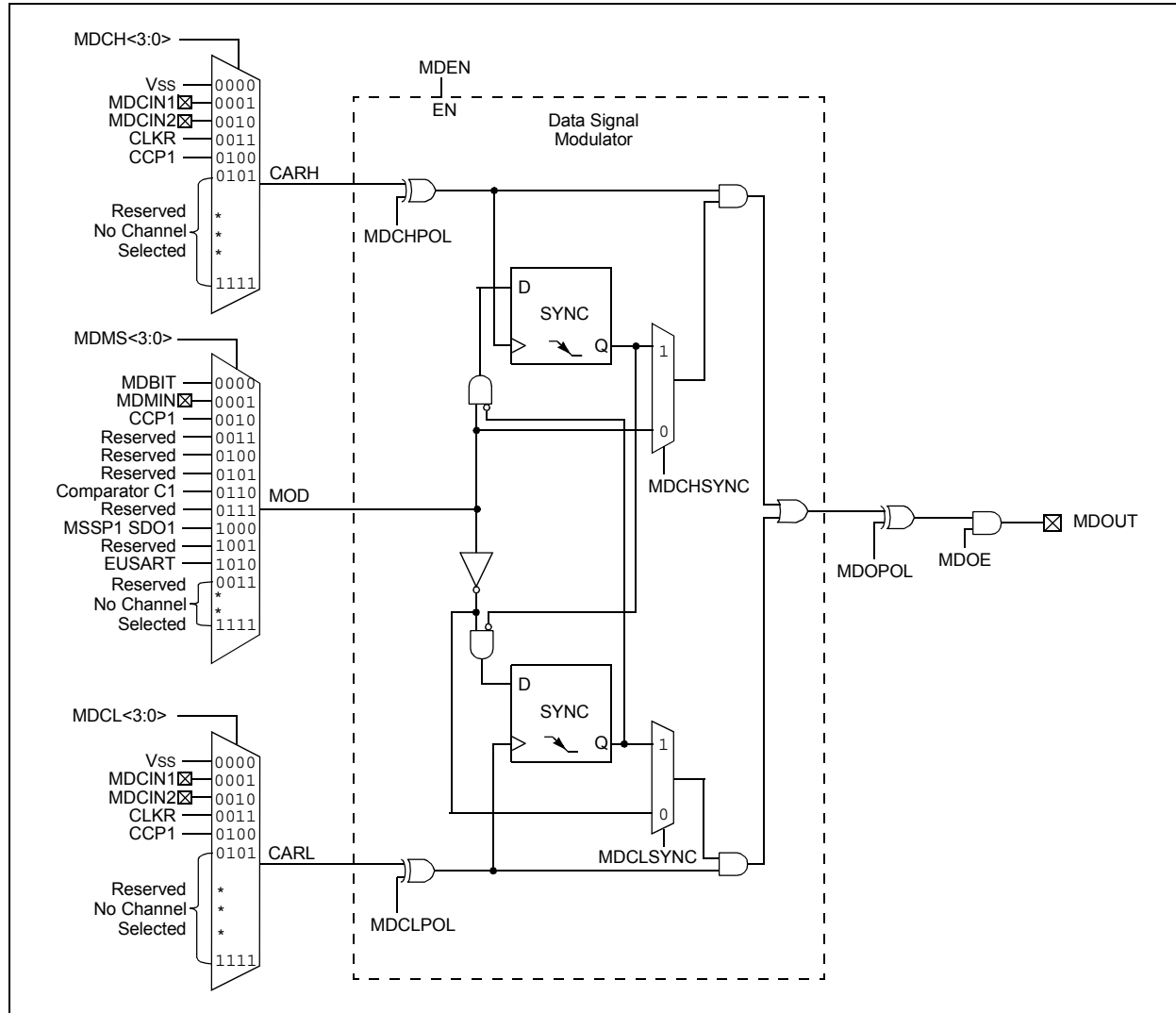
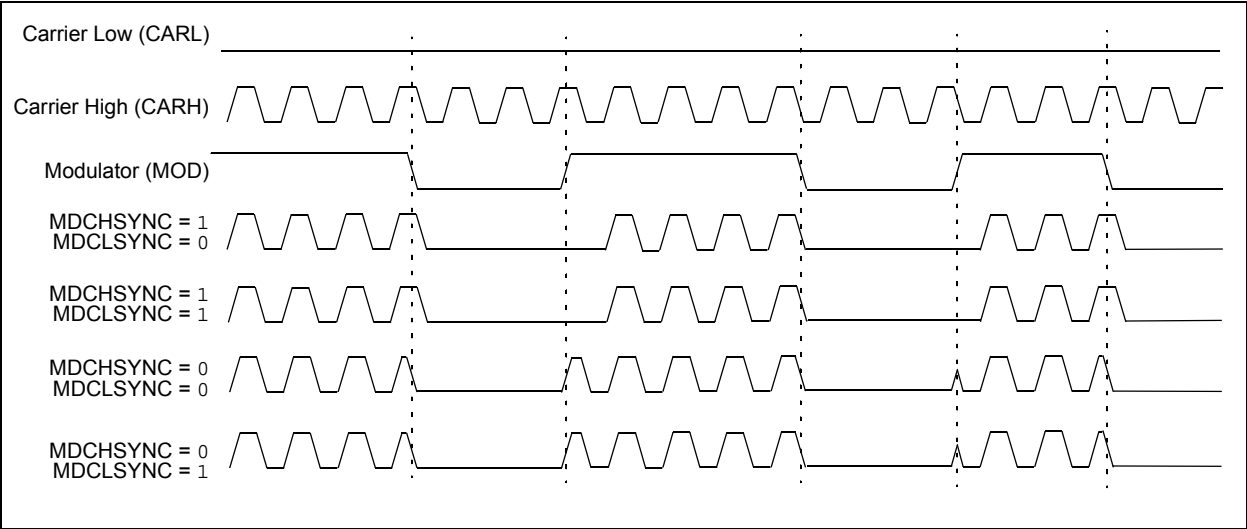


FIGURE 23-2: ON OFF KEYING (OOK) SYNCHRONIZATION



EXAMPLE 23-1: NO SYNCHRONIZATION (MDSHSYNC = 0, MDCLSYNC = 0)

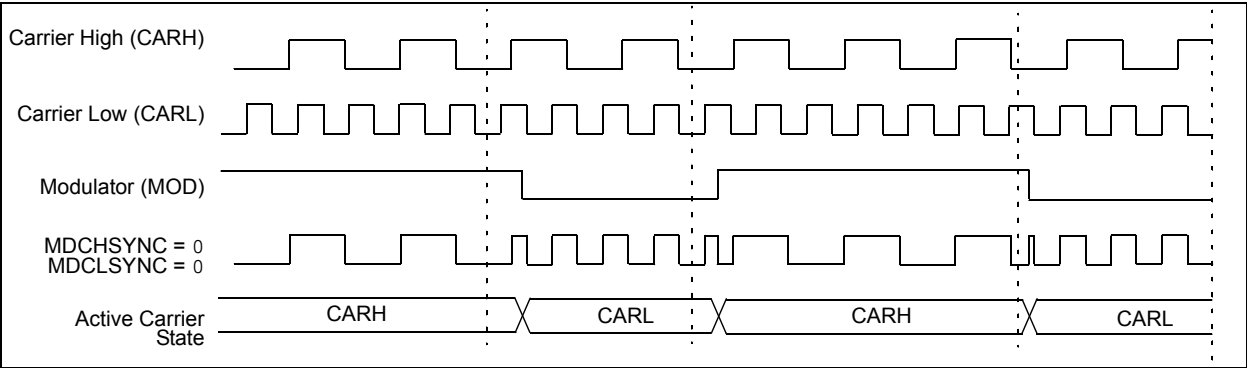
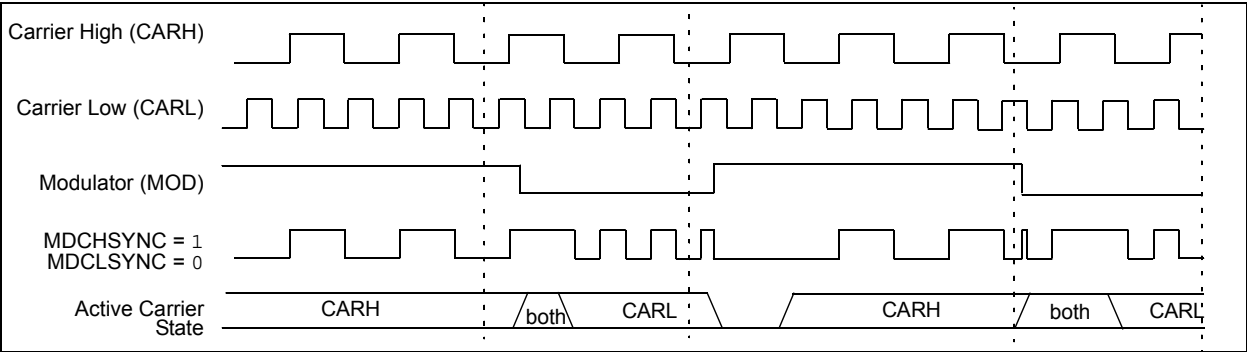


FIGURE 23-3: CARRIER HIGH SYNCHRONIZATION (MDSHSYNC = 1, MDCLSYNC = 0)



PIC12(L)F1840

REGISTER 23-4: MDCARL: MODULATION LOW CARRIER CONTROL REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
MDCLODIS	MDCLPOL	MDCLSYNC	—	MDCL<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 **MDCLODIS:** Modulator Low Carrier Output Disable bit
1 = Output signal driving the peripheral output pin (selected by MDCL<3:0> of the MDCARL register) is disabled
0 = Output signal driving the peripheral output pin (selected by MDCL<3:0> of the MDCARL register) is enabled
- 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 = Reserved. No channel connected.
•
•
•
0101 = Reserved. No channel connected.
0100 = CCP1 output (PWM Output mode only)
0011 = Reference Clock module signal
0010 = Reserved. No channel connected.
0001 = MDCIN1 port pin
0000 = Vss

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

TABLE 23-1: 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
MDCARH	MDCHODIS	MDCHPOL	MDCHSYNC	—	MDCH<3:0>				169
MDCARL	MDCLODIS	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				170
MDCON	MDEN	MDOE	MDSLRL	MDOPOL	MDOUT	—	—	MDBIT	167
MDSRC	MDMSODIS	—	—	—	MDMS<3:0>				168

Legend: — = unimplemented, read as '0'. Shaded cells are not used in the Data Signal Modulator mode.

PIC12(L)F1840

24.4 PWM (Enhanced Mode)

The enhanced PWM mode generates a Pulse-Width Modulation (PWM) signal on up to two different output pins with up to ten bits of resolution. The period, duty cycle, and resolution are controlled by the following registers:

- PR2 registers
- T2CON registers
- CCPR1L registers
- CCP1CON registers

The ECCP modules have the following additional PWM registers which control Auto-shutdown, Auto-restart, Dead-band Delay and PWM Steering modes:

- CCP1AS registers
- PSTR1CON registers
- PWM1CON registers

The enhanced PWM module can generate the following three PWM Output modes:

- Single PWM
- Half-Bridge PWM
- Single PWM with PWM Steering Mode

To select an Enhanced PWM Output mode, the P1M bits of the CCP1CON register must be configured appropriately.

The PWM outputs are multiplexed with I/O pins and are designated P1A and P1B. The polarity of the PWM pins is configurable and is selected by setting the bits CCP1M<3:0> in the CCP1CON register appropriately.

Figure 24-5 shows an example of a simplified block diagram of the Enhanced PWM module.

Table 24-8 shows the pin assignments for various Enhanced PWM modes.

- Note 1:** The corresponding TRIS bit must be cleared to enable the PWM output on the CCP1 pin.
- 2:** Clearing the CCP1CON register will relinquish control of the CCP1 pin.
- 3:** Any pin not used in the enhanced PWM mode is available for alternate pin functions, if applicable.
- 4:** To prevent the generation of an incomplete waveform when the PWM is first enabled, the ECCP module waits until the start of a new PWM period before generating a PWM signal.

FIGURE 24-5: EXAMPLE SIMPLIFIED BLOCK DIAGRAM OF THE ENHANCED PWM MODE

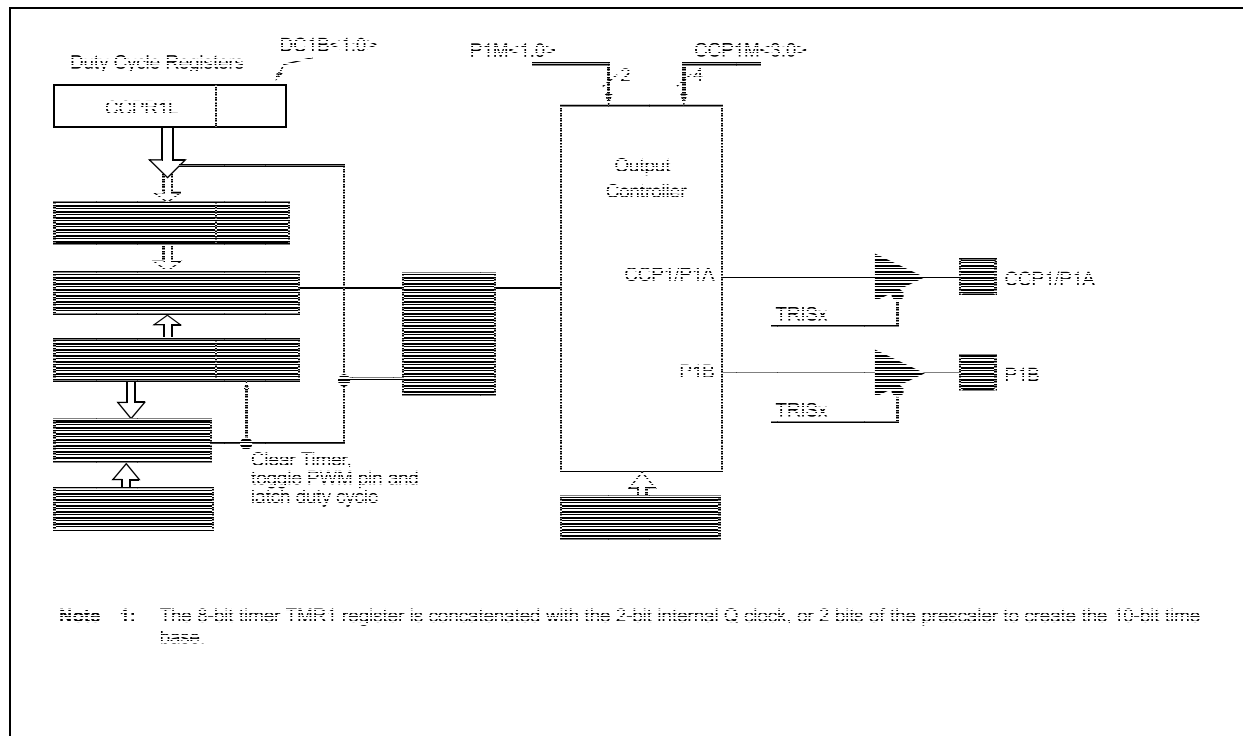
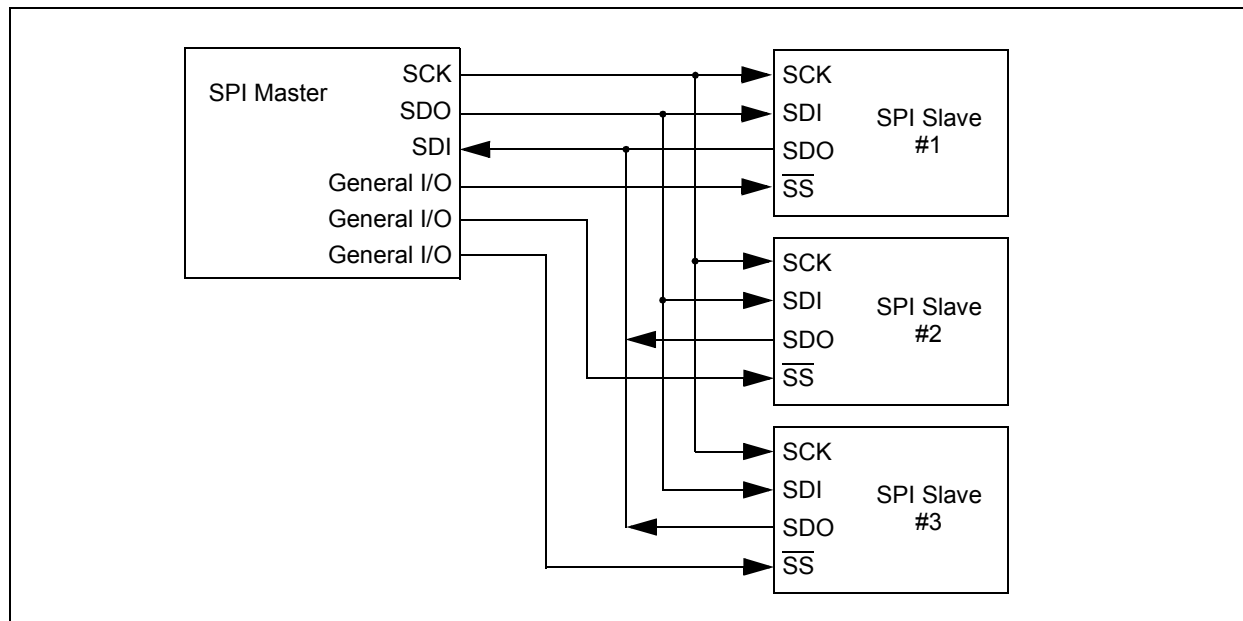


FIGURE 25-4: SPI MASTER AND MULTIPLE SLAVE CONNECTION



25.2.1 SPI MODE REGISTERS

The MSSP1 module has five registers for SPI mode operation. These are:

- MSSP1 STATUS register (SSP1STAT)
- MSSP1 Control register 1 (SSP1CON1)
- MSSP1 Control register 3 (SSP1CON3)
- MSSP1 Data Buffer register (SSP1BUF)
- MSSP1 Address register (SSP1ADD)
- MSSP1 Shift register (SSP1SR)
(Not directly accessible)

SSP1CON1 and SSP1STAT are the control and STATUS registers in SPI mode operation. The SSP1CON1 register is readable and writable. The lower 6 bits of the SSP1STAT are read-only. The upper two bits of the SSP1STAT are read/write.

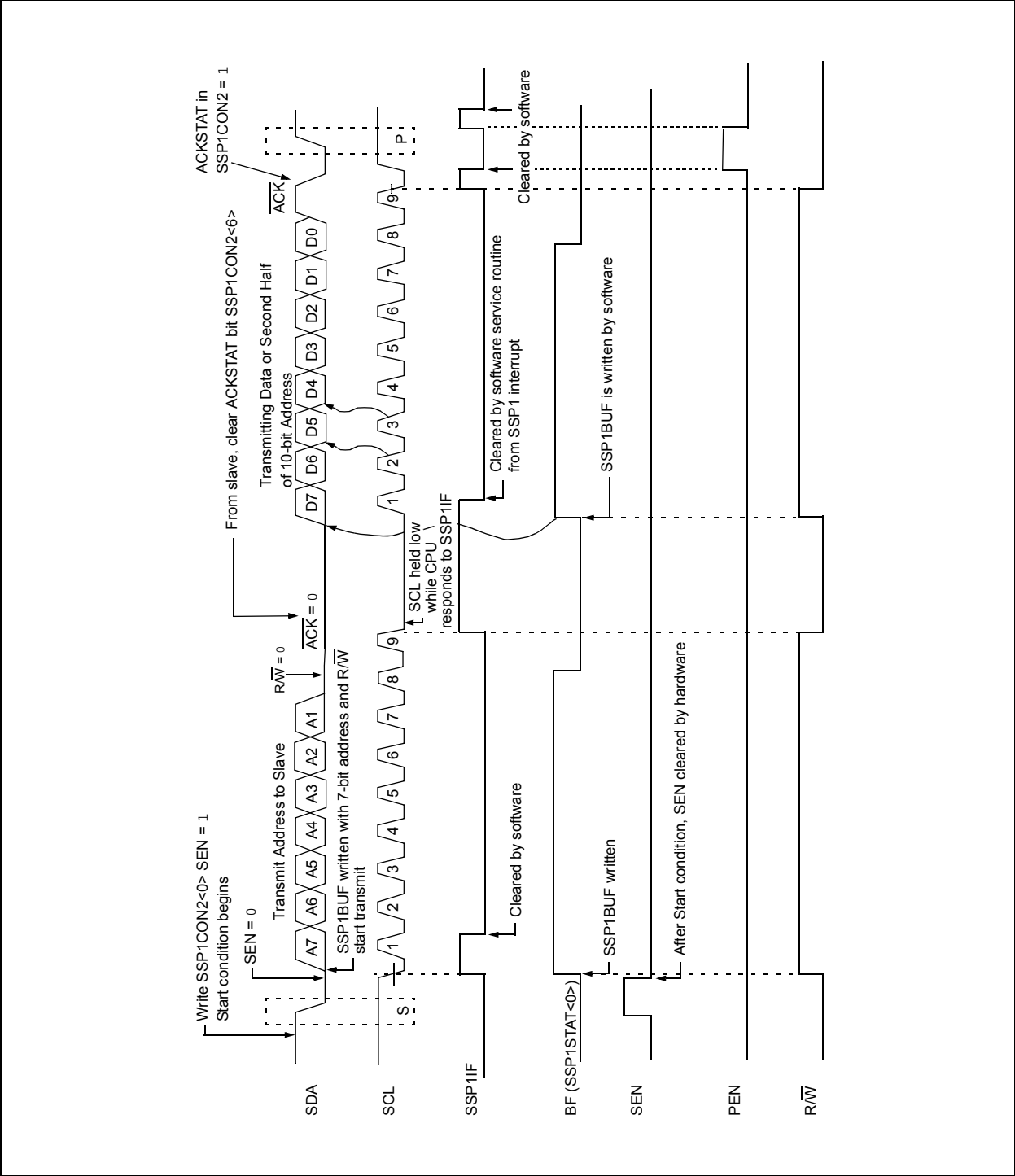
In one SPI master mode, SSP1ADD can be loaded with a value used in the Baud Rate Generator. More information on the Baud Rate Generator is available in **Section 25.7 “Baud Rate Generator”**.

SSP1SR is the shift register used for shifting data in and out. SSP1BUF provides indirect access to the SSP1SR register. SSP1BUF is the buffer register to which data bytes are written, and from which data bytes are read.

In receive operations, SSP1SR and SSP1BUF together create a buffered receiver. When SSP1SR receives a complete byte, it is transferred to SSP1BUF and the SSP1IF interrupt is set.

During transmission, the SSP1BUF is not buffered. A write to SSP1BUF will write to both SSP1BUF and SSP1SR.

FIGURE 25-28: I²C MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)



25.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN bit of the SSP1CON2 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 MSSP1 module then goes into Idle mode (Figure 25-30).

25.6.8.1 WCOL Status Flag

If the user writes the SSP1BUF 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).

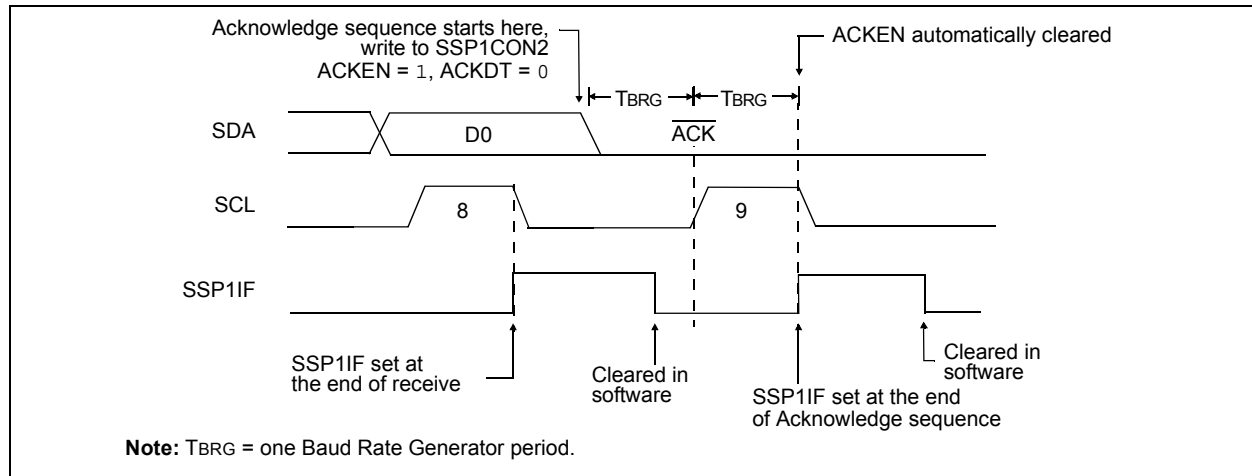
25.6.9 STOP CONDITION TIMING

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 SSP1CON2 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 SSP1STAT register is set. A TBRG later, the PEN bit is cleared and the SSP1IF bit is set (Figure 25-31).

25.6.9.1 WCOL Status Flag

If the user writes the SSP1BUF 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 25-30: ACKNOWLEDGE SEQUENCE WAVEFORM



PIC12(L)F1840

26.1.1.5 TSR Status

The TRMT bit of the TXSTA 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 TXREG. 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.

26.1.1.6 Transmitting 9-Bit Characters

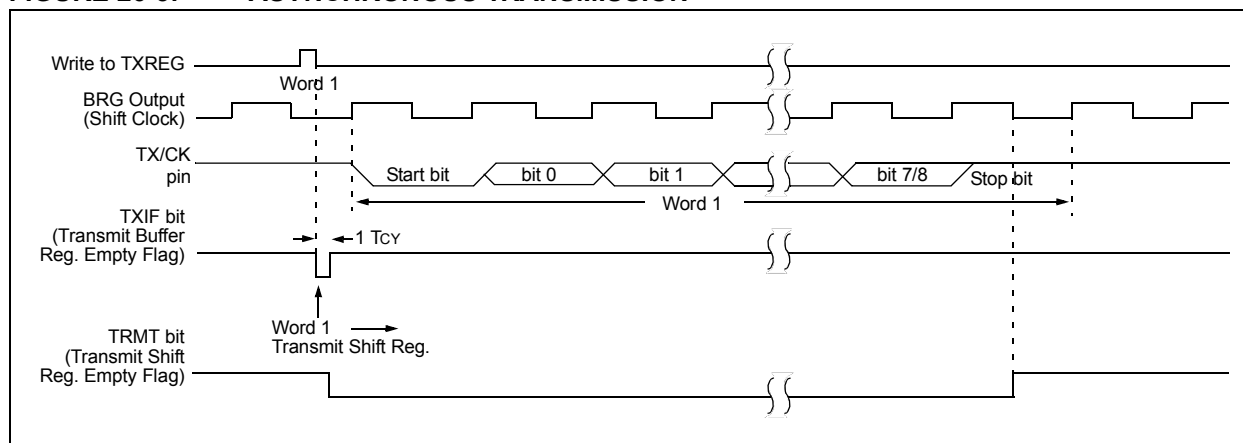
The EUSART supports 9-bit character transmissions. When the TX9 bit of the TXSTA register is set, the EUSART will shift nine bits out for each character transmitted. The TX9D bit of the TXSTA register is the ninth, and Most Significant, data bit. When transmitting 9-bit data, the TX9D data bit must be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

A special 9-bit Address mode is available for use with multiple receivers. See **Section 26.1.2.7 “Address Detection”** for more information on the address mode.

26.1.1.7 Asynchronous Transmission Set-up:

1. Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 26.4 “EUSART Baud Rate Generator (BRG)”**).
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If 9-bit transmission is desired, set the TX9 control bit. A set ninth data bit will indicate that the eight Least Significant data bits are an address when the receiver is set for address detection.
4. Set SCKP bit if inverted transmit is desired.
5. Enable the transmission by setting the TXEN control bit. This will cause the TXIF interrupt bit to be set.
6. If interrupts are desired, set the TXIE interrupt enable bit of the PIE1 register. An interrupt will occur immediately provided that the GIE and PEIE bits of the INTCON register are also set.
7. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D data bit.
8. Load 8-bit data into the TXREG register. This will start the transmission.

FIGURE 26-3: ASYNCHRONOUS TRANSMISSION



PIC12(L)F1840

REGISTER 26-2: RCSTA: 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	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 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)
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:
Don't care
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave
Don't care
- bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables receiver
0 = Disables receiver
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 the receive buffer when RSR<8> 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):
Don't care
- bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG 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.

PIC12(L)F1840

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: $W \rightarrow \text{INDFn}$
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: $(W) \rightarrow \text{OPTION_REG}$

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 $\overline{\text{RI}}$ flag of the PCON register.

Status Affected: None

Description: This instruction provides a way to execute a hardware Reset by software.

FIGURE 31-19: I_{DD} , LFINTOSC, $F_{osc} = 31$ kHz, PIC12LF1840 ONLY

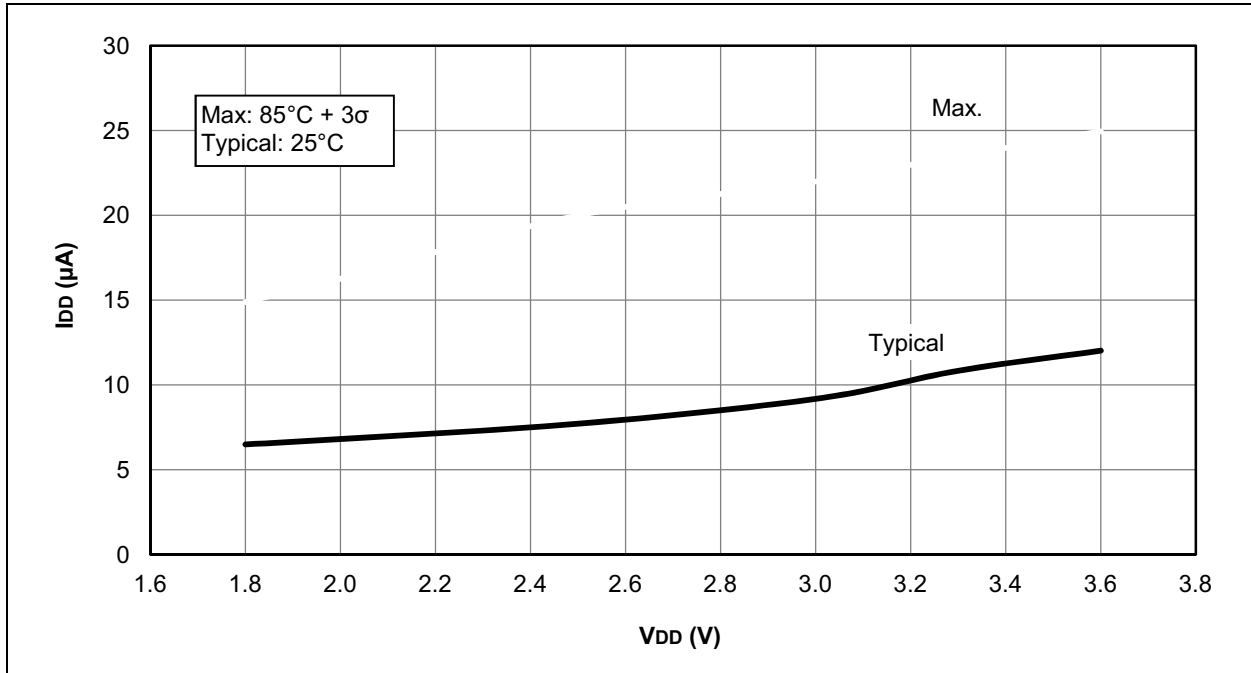


FIGURE 31-20: I_{DD} , LFINTOSC, $F_{osc} = 31$ kHz, PIC12F1840 ONLY

