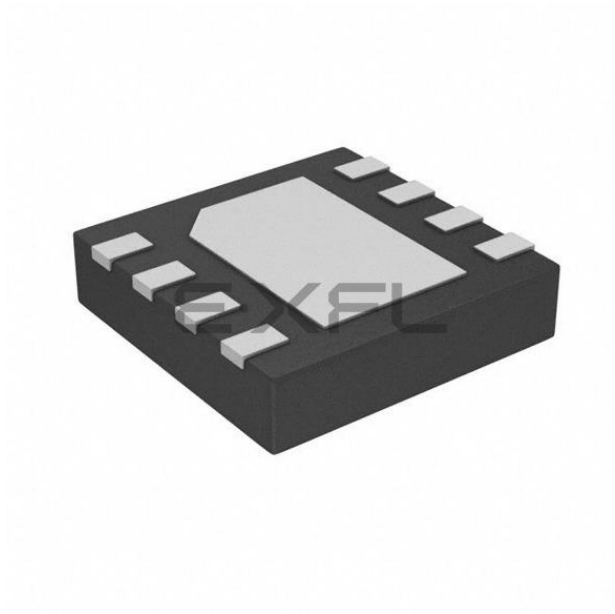


Welcome to E-XFL.COM

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	I²C, 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)	1.8V ~ 3.6V
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/pic12lf1840-i-mf

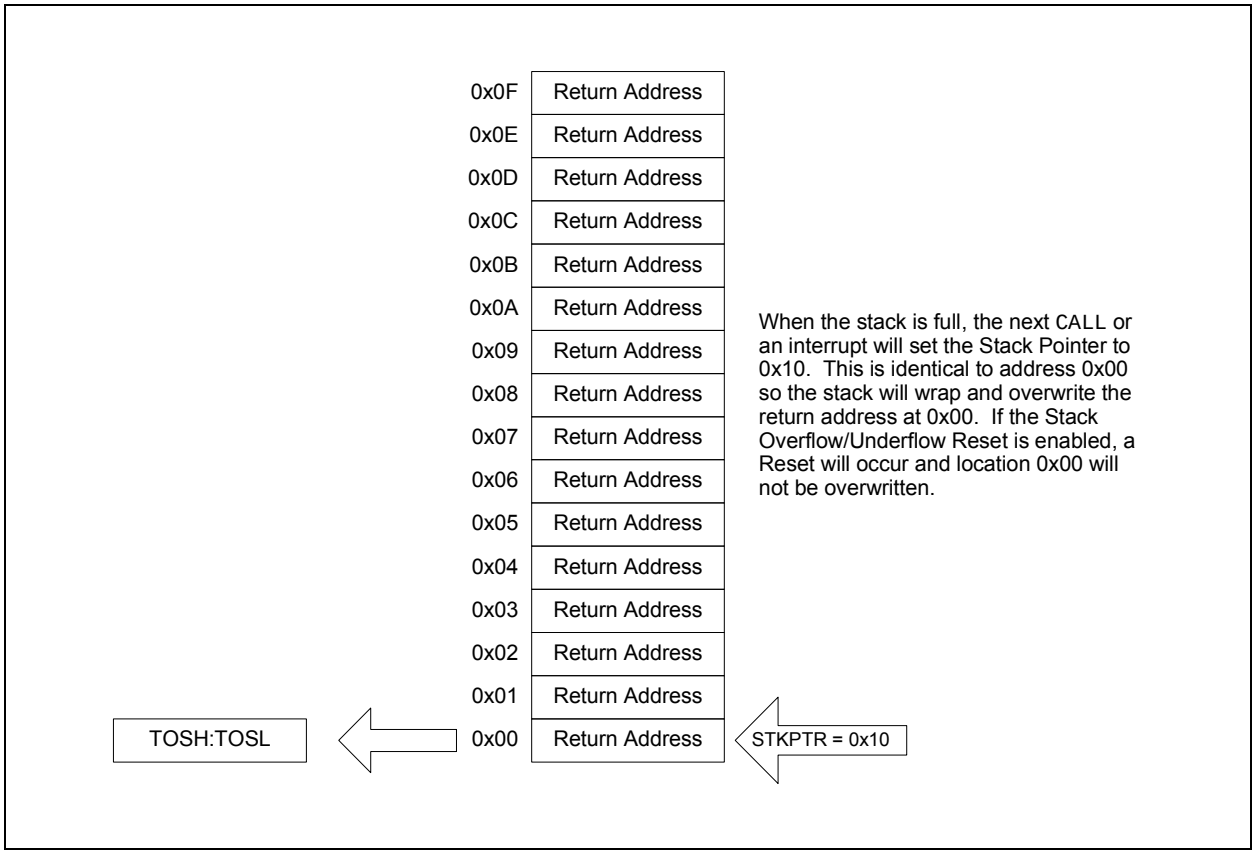
TABLE 1-2: PIC12(L)F1840 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/AN0/CPS0/C1IN+/ DACOUT/TX/CK/SDO/ $\overline{SS}$(1)/P1B/MDOUT/ICSPDAT/ ICDDAT	RA0	TTL	CMOS	General purpose I/O.
	AN0	AN	—	ADC Channel 0 input.
	CPS0	AN	—	Capacitive sensing input 0.
	C1IN+	AN	—	Comparator C1 positive input.
	DACOUT	—	AN	Digital-to-Analog Converter output.
	TX	—	CMOS	USART asynchronous transmit.
	CK	ST	CMOS	USART synchronous clock.
	SDO	—	CMOS	SPI data output.
	$\overline{SS}$	ST	—	Slave Select input.
	P1B	—	CMOS	PWM output.
	MDOUT	—	CMOS	Modulator output.
	ICSPDAT	ST	CMOS	ICSP™ Data I/O.
RA1/AN1/CPS1/VREF/C1IN0-/ SRI/RX/DT/SCL/SCK/ MDMIN/ICSPCLK/ICDCLK	RA1	TTL	CMOS	General purpose I/O.
	AN1	AN	—	ADC Channel 1 input.
	CPS1	AN	—	Capacitive sensing input 1.
	VREF	AN	—	ADC and DAC Positive Voltage Reference input.
	C1IN0-	AN	—	Comparator C1 negative input.
	SRI	ST	—	SR Latch input.
	RX	ST	—	USART asynchronous input.
	DT	ST	CMOS	USART synchronous data.
	SCL	I^2C™	OD	I^2C™ clock.
	SCK	ST	CMOS	SPI clock.
	MDMIN	ST	—	Modulator source input.
	ICSPCLK	ST	—	Serial Programming Clock.
RA2/AN2/CPS2/C1OUT/SRQ/ T0CKI/CCP1/P1A/FLT0/ SDA/SDI/INT/MDCIN1	RA2	ST	CMOS	General purpose I/O.
	AN2	AN	—	ADC Channel 2 input.
	CPS2	AN	—	Capacitive sensing input 2.
	C1OUT	—	CMOS	Comparator C1 output.
	SRQ	—	CMOS	SR Latch non-inverting output.
	T0CKI	ST	—	Timer0 clock input.
	CCP1	ST	CMOS	Capture/Compare/PWM 1.
	P1A	—	CMOS	PWM output.
	FLT0	ST	—	ECCP Auto-Shutdown Fault input.
	SDA	I^2C™	OD	I^2C™ data input/output.
	SDI	CMOS	—	SPI data input.
	INT	ST	—	External interrupt.
	MDCIN1	ST	—	Modulator Carrier Input 1.
RA3/$\overline{SS}$/T1G(1)/VPP/$\overline{MCLR}$	RA3	TTL	—	General purpose input.
	$\overline{SS}$	ST	—	Slave Select input.
	T1G	ST	—	Timer1 Gate input.
	VPP	HV	—	Programming voltage.
	$\overline{MCLR}$	ST	—	Master Clear with internal pull-up.

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 levels
HV = High Voltage XTAL = Crystal

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

FIGURE 3-7: ACCESSING THE STACK EXAMPLE 4



3.5.2 OVERFLOW/UNDERFLOW RESET

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

3.6 Indirect Addressing

The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the File Select Registers (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 three memory regions:

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

REGISTER 4-1: CONFIG1: CONFIGURATION WORD 1 (CONTINUED)

bit 4-3 **WDTE<1:0>:** Watchdog Timer Enable bit
11 = WDT enabled
10 = WDT enabled while running and disabled in Sleep
01 = WDT controlled by the SWDTEN bit in the WDTCON register
00 = WDT disabled

bit 2-0 **FOSC<2:0>:** Oscillator Selection bits
111 = ECH: External Clock, High-Power mode (4-20 MHz): device clock supplied to CLKIN pin
110 = ECM: External Clock, Medium-Power mode (0.5-4 MHz): device clock supplied to CLKIN pin
101 = ECL: External Clock, Low-Power mode (0-0.5 MHz): device clock supplied to CLKIN pin
100 = INTOSC oscillator: I/O function on CLKIN pin
011 = EXTRC oscillator: External RC circuit connected to CLKIN pin
010 = HS oscillator: High-speed crystal/resonator connected between OSC1 and OSC2 pins
001 = XT oscillator: Crystal/resonator connected between OSC1 and OSC2 pins
000 = LP oscillator: Low-power crystal connected between OSC1 and OSC2 pins

Note 1: Enabling Brown-out Reset does not automatically enable Power-up Timer.
2: The entire data EEPROM will be erased when the code protection is turned off during an erase.
3: The entire program memory will be erased when the code protection is turned off.

11.3.2 ERASING FLASH PROGRAM MEMORY

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

1. Load the EEADRH:EEADRL register pair with the address of new row to be erased.
2. Clear the CFGS bit of the EECON1 register.
3. Set the EEPGD, FREE and WREN bits of the EECON1 register.
4. Write 55h, then AAh, to EECON2 (Flash programming unlock sequence).
5. Set control bit WR of the EECON1 register to begin the erase operation.
6. Poll the FREE bit in the EECON1 register to determine when the row erase has completed.

See Example 11-4.

After the "`BSF EECON1,WR`" instruction, the processor requires two cycles to set up the erase operation. The user must place two `NOP` instructions after the WR bit is set. The processor will halt internal operations for the typical 2 ms erase time. This is not Sleep mode as the clocks and peripherals will continue to run. After the erase cycle, the processor will resume operation with the third instruction after the EECON1 write instruction.

11.3.3 WRITING TO FLASH PROGRAM MEMORY

Program memory is programmed using the following steps:

1. Load the starting address of the word(s) to be programmed.
2. Load the write latches 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 11-2 (block writes to program memory with 32 write latches) for more details. The write latches are aligned to the address boundary defined by EEADRL as shown in Table 11-1. Write operations do not cross these boundaries. At the completion of a program memory write operation, the write latches are reset to contain 0x3FFF.

The following steps should be completed to load the write latches and program a block of program memory. These steps are divided into two parts. First, all write latches are loaded with data except for the last program memory location. Then, the last write latch is loaded and the programming sequence is initiated. A special unlock sequence is required to load a write latch with data or initiate a Flash programming operation. This unlock sequence should not be interrupted.

1. Set the EEPGD and WREN bits of the EECON1 register.
2. Clear the CFGS bit of the EECON1 register.
3. Set the LWLO bit of the EECON1 register. When the LWLO bit of the EECON1 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 EEADRH:EEADRL register pair with the address of the location to be written.
5. Load the EEDATH:EEDATL register pair with the program memory data to be written.
6. Write 55h, then AAh, to EECON2, then set the WR bit of the EECON1 register (Flash programming unlock sequence). The write latch is now loaded.
7. Increment the EEADRH:EEADRL 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 EECON1 register. When the LWLO bit of the EECON1 register is '`0`', the write sequence will initiate the write to Flash program memory.
10. Load the EEDATH:EEDATL register pair with the program memory data to be written.
11. Write 55h, then AAh, to EECON2, then set the WR bit of the EECON1 register (Flash programming unlock sequence). The entire latch block is now written to Flash program memory.

It is not necessary to load the entire write latch block with user program data. However, the entire write latch block will be written to program memory.

An example of the complete write sequence for 32 words is shown in Example 11-5. The initial address is loaded into the EEADRH:EEADRL register pair; the 32 words of data are loaded using indirect addressing.

16.3 Register Definitions: ADC Control

REGISTER 16-1: ADCON0: ADC CONTROL REGISTER 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	R/W-0/0
—	CHS<4:0>					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 **Unimplemented:** Read as '0'

bit 6-2 **CHS<4:0>:** Analog Channel Select bits

11111 = FVR (Fixed Voltage Reference) Buffer 1 Output(2)
11110 = DAC_output(1)
11101 = Temperature Indicator(3).
11100 = Reserved. No channel connected.
•
•
•
00100 = Reserved. No channel connected.
00011 = AN3
00010 = AN2
00001 = AN1
00000 = AN0

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 17.0 "Digital-to-Analog Converter (DAC) Module"** for more information.
2: See **Section 14.0 "Fixed Voltage Reference (FVR)"** for more information.
3: See **Section 15.0 "Temperature Indicator Module"** for more information.

16.4 ADC Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (C_{HOLD}) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 16-4. The source impedance (R_S) and the internal sampling switch (R_{SS}) impedance directly affect the time required to charge the capacitor C_{HOLD}. The sampling switch (R_{SS}) impedance varies over the device voltage (V_{DD}), refer to Figure 16-4. **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 16-1 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 16-1: 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*:*

$$T_C = -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ ln(1/511)$$
$$= -12.5pF(1k\Omega + 7k\Omega + 10k\Omega)\ ln(0.001957)$$
$$1.72\mu s$$

Therefore:

$$T_{ACQ} = 2\mu s + 1.72\mu s + [(50°C - 25°C)(0.05\mu s/°C)]$$
$$= 4.97\mu s$$

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

2: The charge holding capacitor (C_{HOLD}) 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 17-1: DIGITAL-TO-ANALOG CONVERTER BLOCK DIAGRAM

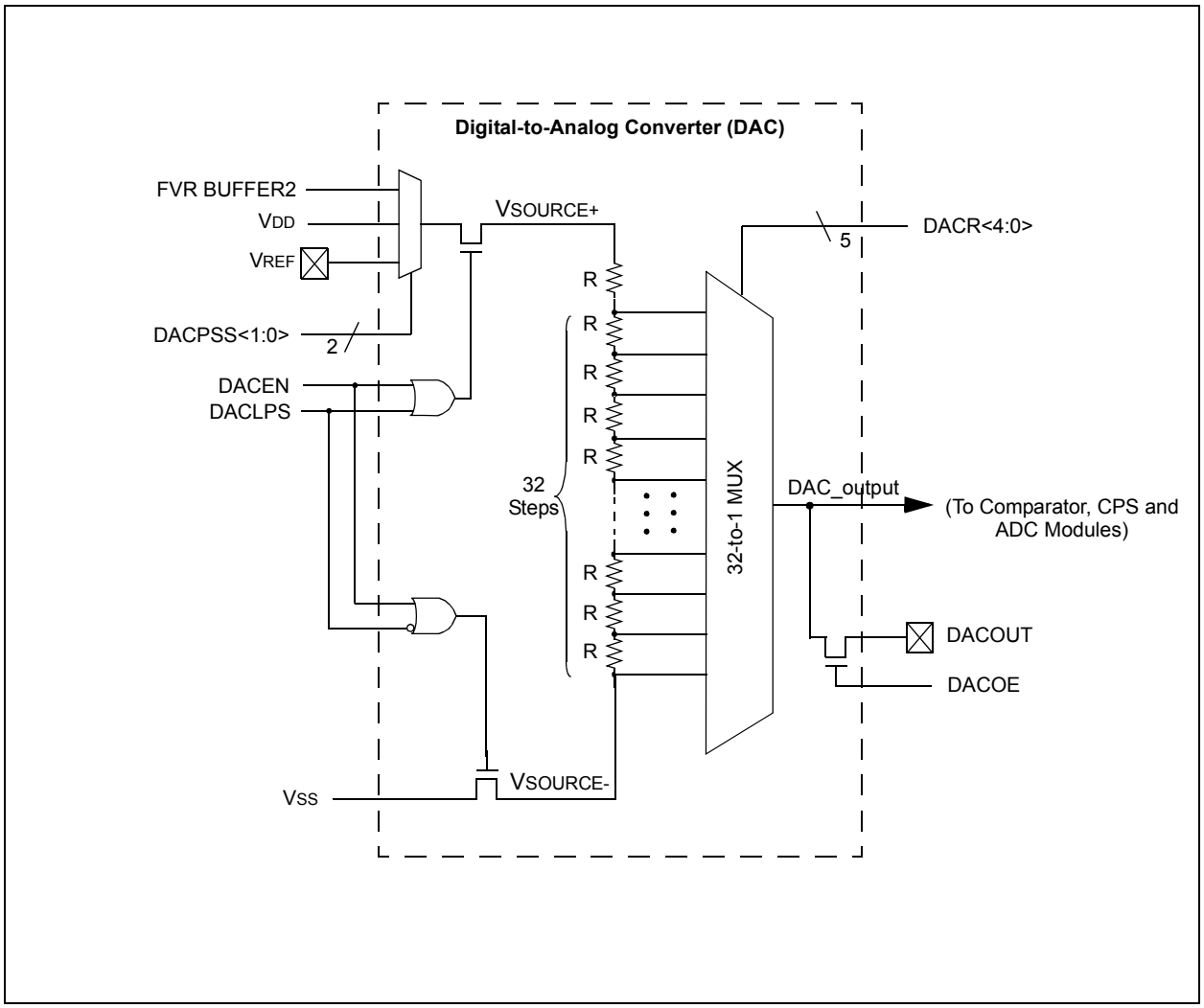


FIGURE 17-2: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE

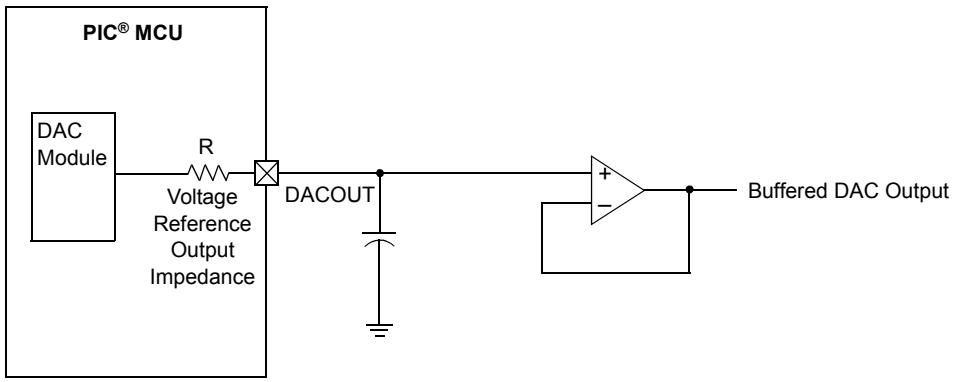
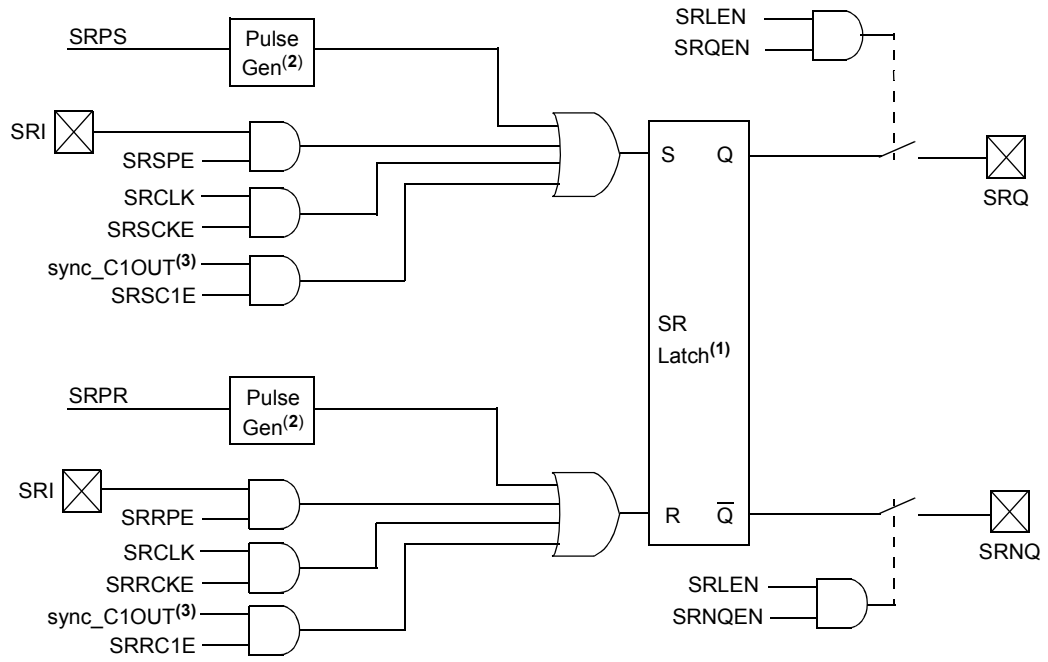


FIGURE 18-1: SR LATCH SIMPLIFIED BLOCK DIAGRAM



Note 1: If R = 1 and S = 1 simultaneously, Q = 0, $\overline{Q}$ = 1
2: Pulse generator causes a 1 Q-state pulse width.
3: Name denotes the connection point at the comparator output.

TABLE 18-1: SRCLK FREQUENCY TABLE

SRCLK	Divider	Fosc = 32 MHz	Fosc = 20 MHz	Fosc = 16 MHz	Fosc = 4 MHz	Fosc = 1 MHz
111	512	62.5 kHz	39.0 kHz	31.3 kHz	7.81 kHz	1.95 kHz
110	256	125 kHz	78.1 kHz	62.5 kHz	15.6 kHz	3.90 kHz
101	128	250 kHz	156 kHz	125 kHz	31.25 kHz	7.81 kHz
100	64	500 kHz	313 kHz	250 kHz	62.5 kHz	15.6 kHz
011	32	1 MHz	625 kHz	500 kHz	125 kHz	31.3 kHz
010	16	2 MHz	1.25 MHz	1 MHz	250 kHz	62.5 kHz
001	8	4 MHz	2.5 MHz	2 MHz	500 kHz	125 kHz
000	4	8 MHz	5 MHz	4 MHz	1 MHz	250 kHz

TABLE 21-5: 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
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	103
CCP1CON	P1M<1:0>		DC1B<1:0>		CCP1M<3:0>				189
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	72
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	73
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	75
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								150*
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								150*
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	102
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	154
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		155

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.

* Page provides register information.

TABLE 22-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCP1CON	P1M<1:0>		DC1B<1:0>		CCP1M<3:0>				189
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	72
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	73
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	75
PR2	Timer2 Module Period Register								157*
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		159
TMR2	Holding Register for the 8-bit TMR2 Register								157*

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for Timer2 module.

* Page provides register information.

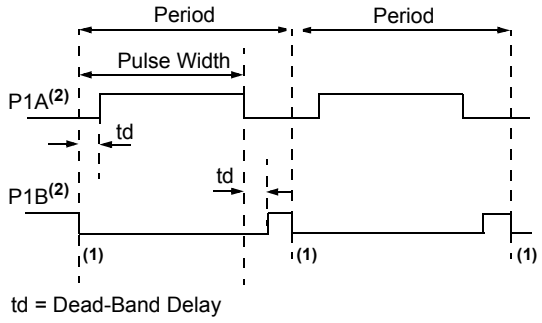
24.4.1 HALF-BRIDGE MODE

In Half-Bridge mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the CCP1/P1A pin, while the complementary PWM output signal is output on the P1B pin (see Figure 24-9). This mode can be used for Half-Bridge applications, as shown in Figure 24-9, or for Full-Bridge applications, where four power switches are being modulated with two PWM signals.

In Half-Bridge mode, the programmable dead-band delay can be used to prevent shoot-through current in Half-Bridge power devices. The value of the P1DC<6:0> bits of the PWM1CON register sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding output remains inactive during the entire cycle. See **Section 24.4.4 "Programmable Dead-Band Delay Mode"** for more details of the dead-band delay operations.

Since the P1A and P1B outputs are multiplexed with the PORT data latches, the associated TRIS bits must be cleared to configure P1A and P1B as outputs.

FIGURE 24-8: EXAMPLE OF HALF-BRIDGE PWM OUTPUT

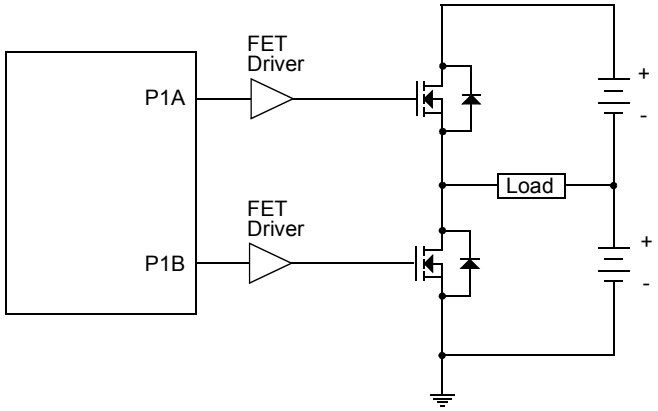


Note 1: At this time, the TMR2 register is equal to the PR2 register.

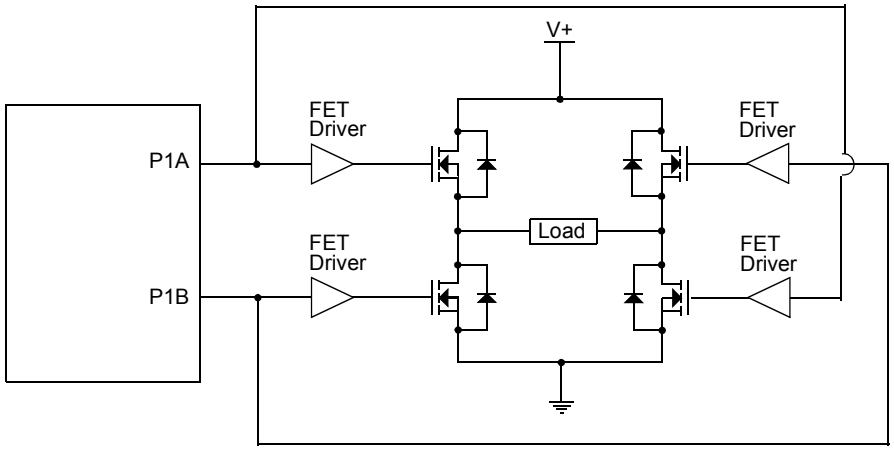
2: Output signals are shown as active-high.

FIGURE 24-9: EXAMPLE OF HALF-BRIDGE APPLICATIONS

Standard Half-Bridge Circuit ("Push-Pull")



Half-Bridge Output Driving a Full-Bridge Circuit



24.4.5 PWM STEERING MODE

In Single Output mode, PWM steering allows any of the PWM pins to be the modulated signal. Additionally, the same PWM signal can be simultaneously available on multiple pins.

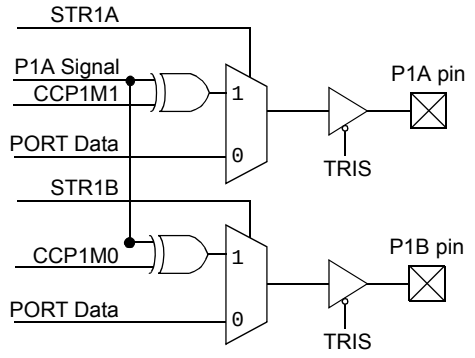
Once the Single Output mode is selected (CCP1M<3:2> = 11 and P1M<1:0> = 00 of the CCP1CON register), the user firmware can bring out the same PWM signal to one or two output pins by setting the appropriate STR1 bits of the PSTR1CON register, as shown in Table 24-8.

Note:	The associated TRIS bits must be set to output ('0') to enable the pin output driver in order to see the PWM signal on the pin.

While the PWM Steering mode is active, the CCP1M<1:0> bits of the CCP1CON register determine the polarity of the output pins.

The PWM auto-shutdown operation also applies to PWM Steering mode as described in **Section 24.4.2 "Enhanced PWM Auto-Shutdown Mode"**. An auto-shutdown event will only affect pins that have PWM outputs enabled.

FIGURE 24-14: SIMPLIFIED STEERING BLOCK DIAGRAM



Note 1: Port outputs are configured as shown when the CCP1CON register bits P1M<1:0> = 00 and CCP1M<3:2> = 11.

2: Single PWM output requires setting at least one of the STR1 bits.

24.4.5.1 Steering Synchronization

The STR1SYNC bit of the PSTR1CON register gives the user two selections of when the steering event will happen. When the STR1SYNC bit is '0', the steering event will happen at the end of the instruction that writes to the PSTR1CON register. In this case, the output signal at the output pins may be an incomplete PWM waveform. This operation is useful when the user firmware needs to immediately remove a PWM signal from the pin.

When the STR1SYNC bit is '1', the effective steering update will happen at the beginning of the next PWM period. In this case, steering on/off the PWM output will always produce a complete PWM waveform.

Figures 24-15 and 24-16 illustrate the timing diagrams of the PWM steering depending on the STR1SYNC setting.

24.4.6 START-UP CONSIDERATIONS

When any PWM mode is used, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins.

The CCP1M<1:0> bits of the CCP1CON register allow the user to choose whether the PWM output signals are active-high or active-low for each of the PWM output pins (P1A and P1B). The PWM output polarities must be selected before the PWM pin output drivers are enabled. Changing the polarity configuration while the PWM pin output drivers are enable is not recommended since it may result in damage to the application circuits.

The P1A and P1B output latches may not be in the proper states when the PWM module is initialized. Enabling the PWM pin output drivers at the same time as the Enhanced PWM modes may cause damage to the application circuit. The Enhanced PWM modes must be enabled in the proper Output mode and complete a full PWM cycle before enabling the PWM pin output drivers. The completion of a full PWM cycle is indicated by the TMR2IF bit of the PIR1 register being set as the second PWM period begins.

Note:	When the microcontroller is released from Reset, all of the I/O pins are in the high-impedance state. The external circuits must keep the power switch devices in the Off state until the microcontroller drives the I/O pins with the proper signal levels or activates the PWM output(s).

25.0 MASTER SYNCHRONOUS SERIAL PORT MODULE

25.1 Master SSP (MSSP1) Module Overview

The Master Synchronous Serial Port (MSSP1) 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 MSSP1 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 25-1 is a block diagram of the SPI interface module.

FIGURE 25-1: MSSP1 BLOCK DIAGRAM (SPI MODE)

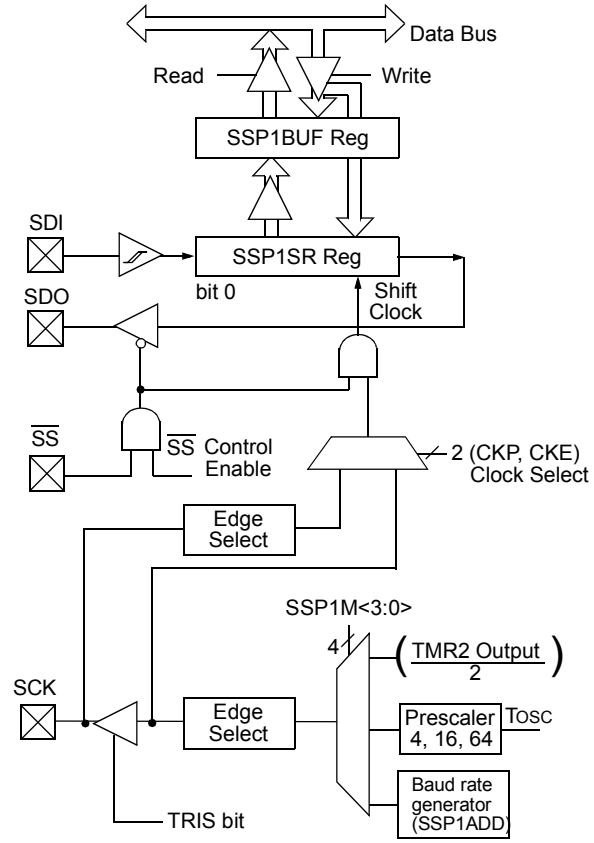


FIGURE 25-7: SPI DAISY-CHAIN CONNECTION

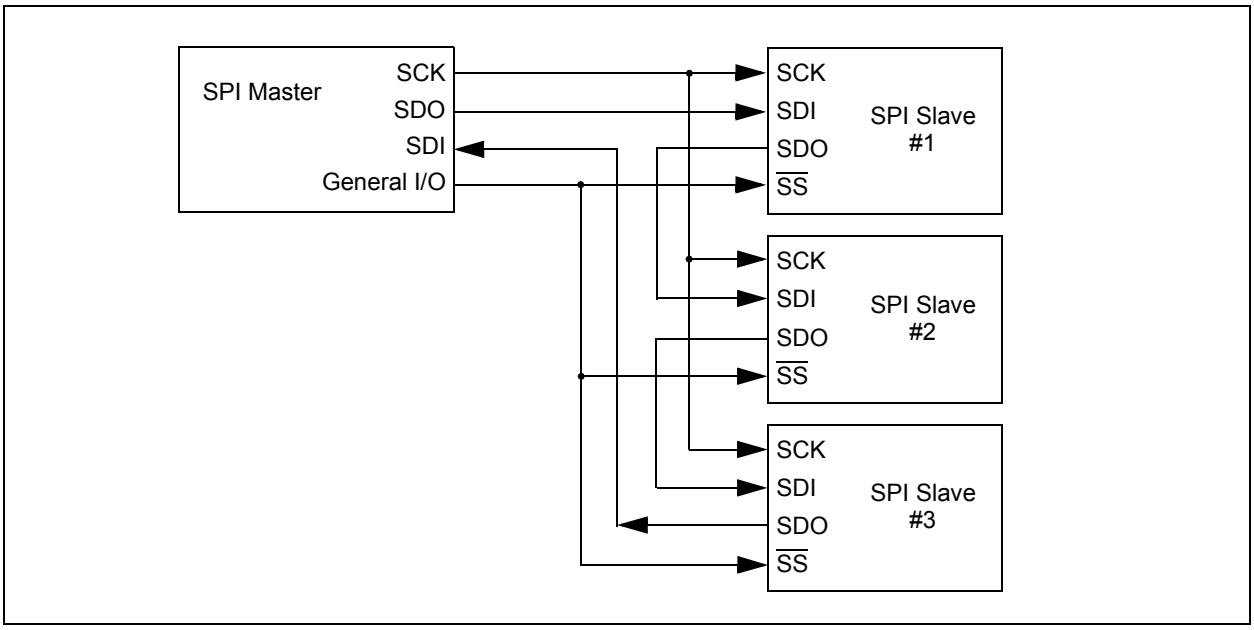
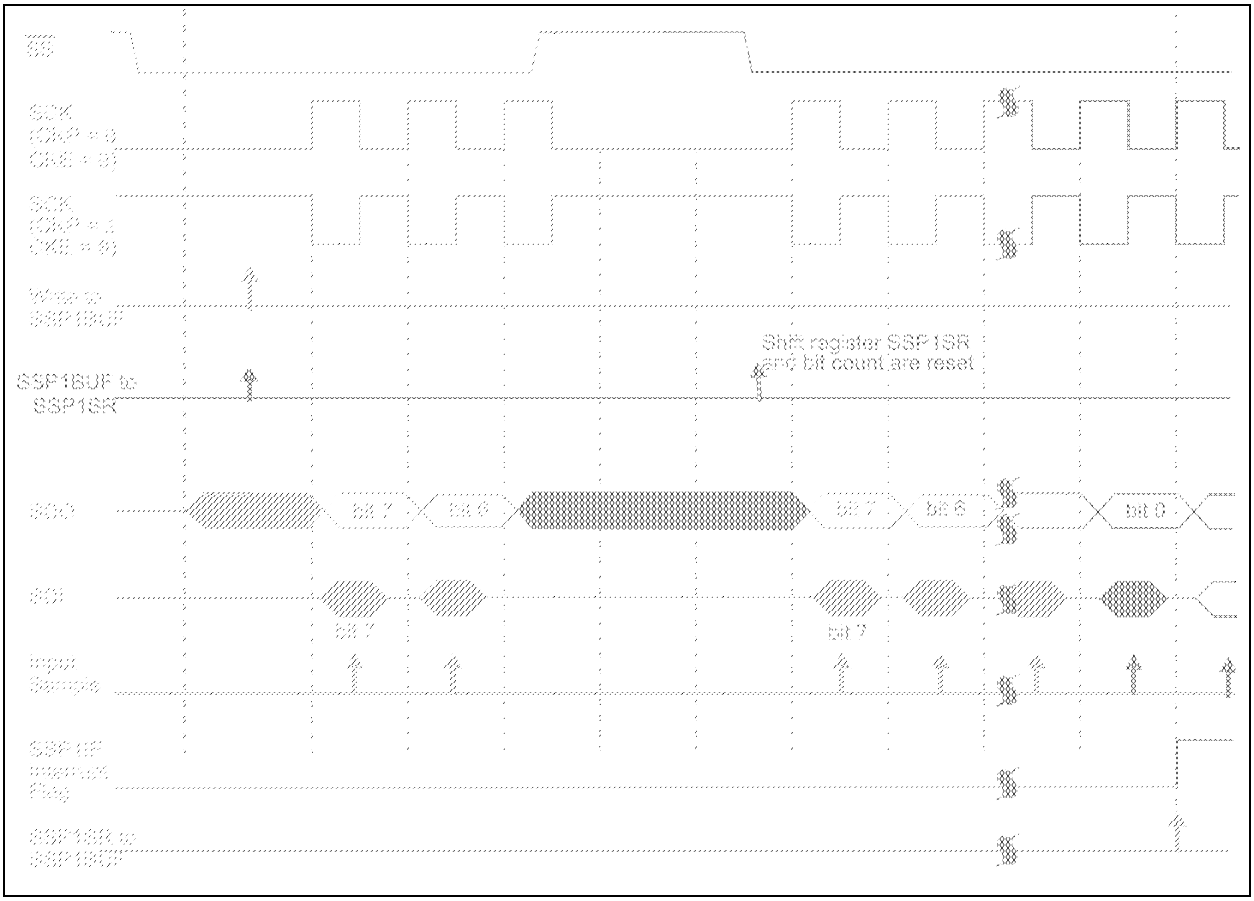


FIGURE 25-8: SLAVE SELECT SYNCHRONOUS WAVEFORM



25.5.2 SLAVE RECEPTION

When the R/$\overline{W}$ bit of a matching received address byte is clear, the R/$\overline{W}$ bit of the SSP1STAT register is cleared. The received address is loaded into the SSP1BUF 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 SSP1STAT register is set, or bit SSP1OV of the SSP1CON1 register is set. The BOEN bit of the SSP1CON3 register modifies this operation. For more information see Register 25-4.

An MSSP1 interrupt is generated for each transferred data byte. Flag bit, SSP1IF, must be cleared by software.

When the SEN bit of the SSP1CON2 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 SSP1CON1 register, except sometimes in 10-bit mode. See **Section 25.2.3 "SPI Master Mode"** for more detail.

25.5.2.1 7-bit Addressing Reception

This section describes a standard sequence of events for the MSSP1 module configured as an I^2C Slave in 7-bit Addressing mode. All decisions made by hardware or software and their effect on reception. Figure 25-14 and Figure 25-15 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 SSP1STAT is set; SSP1IF 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 SSP1IF bit.
5. Software clears the SSP1IF bit.
6. Software reads received address from SSP1BUF 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 SSP1IF bit.
10. Software clears SSP1IF.
11. Software reads the received byte from SSP1BUF clearing BF.
12. Steps 8-12 are repeated for all received bytes from the master.
13. Master sends Stop condition, setting P bit of SSP1STAT, and the bus goes idle.

25.5.2.2 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 8th 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.

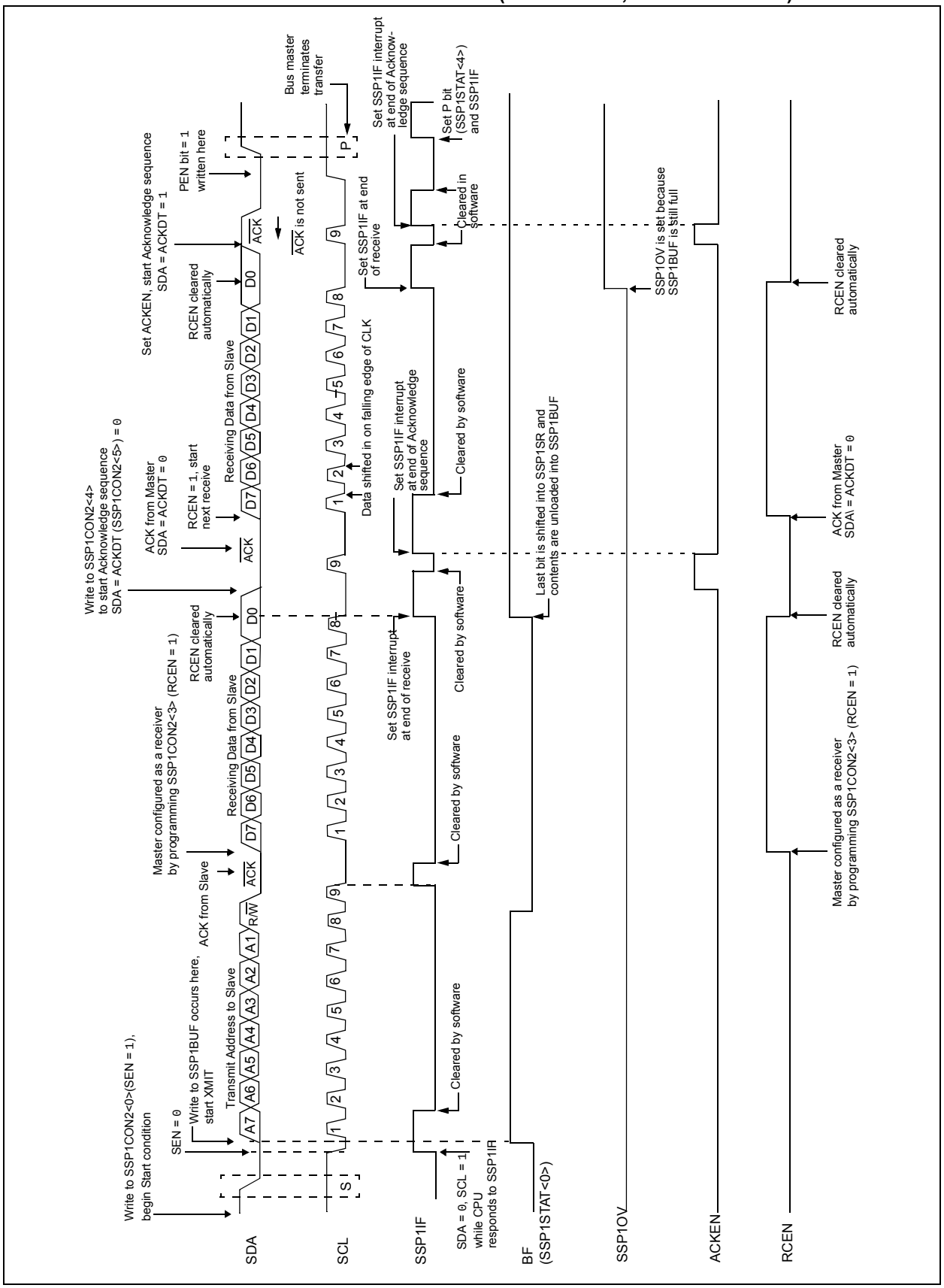
This list describes the steps that need to be taken by slave software to use these options for I^2C communication. Figure 25-16 displays a module using both address and data holding. Figure 25-17 includes the operation with the SEN bit of the SSP1CON2 register set.

1. S bit of SSP1STAT is set; SSP1IF is set if interrupt on Start detect is enabled.
2. Matching address with R/$\overline{W}$ bit clear is clocked in. SSP1IF is set and CKP cleared after the 8th falling edge of SCL.
3. Slave clears the SSP1IF.
4. Slave can look at the ACKTIM bit of the SSP1CON3 register to determine if the SSP1IF was after or before the $\overline{ACK}$.
5. Slave reads the address value from SSP1BUF, 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. SSP1IF 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 SSP1IF.

Note: SSP1IF is still set after the 9th falling edge of SCL even if there is no clock stretching and BF has been cleared. Only if NACK is sent to Master is SSP1IF not set

11. SSP1IF set and CKP cleared after 8th falling edge of SCL for a received data byte.
12. Slave looks at ACKTIM bit of SSP1CON3 to determine the source of the interrupt.
13. Slave reads the received data from SSP1BUF 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 SSTSTAT register.

FIGURE 25-29: I²C MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)



REGISTER 25-2: SSP1CON1: SSP1 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	SSP1OV	SSP1EN	CKP	SSP1M<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
Master mode:
1 = A write to the SSP1BUF register was attempted while the I^2C conditions were not valid for a transmission to be started
0 = No collision
Slave mode:
1 = The SSP1BUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6 **SSP1OV:** Receive Overflow Indicator bit(1)
In SPI mode:
1 = A new byte is received while the SSP1BUF register is still holding the previous data. In case of overflow, the data in SSP1SR is lost. Overflow can only occur in Slave mode. In Slave mode, the user must read the SSP1BUF, 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 SSP1BUF register (must be cleared in software).
0 = No overflow
In I^2C mode:
1 = A byte is received while the SSP1BUF register is still holding the previous byte. SSP1OV is a "don't care" in Transmit mode (must be cleared in software).
0 = No overflow

bit 5 **SSP1EN:** Synchronous Serial Port Enable bit
In both modes, when enabled, these 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

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 = TMR2 output/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 SSP1BUF register.
2: When enabled, these pins must be properly configured as input or output.
3: When enabled, the SDA and SCL pins must be configured as inputs.
4: SSP1ADD values of 0, 1 or 2 are not supported for I^2C mode.
5: SSP1ADD value of '0' is not supported. Use SSP1M = 0000 instead.

FIGURE 31-13: IDD TYPICAL, EXTERNAL CLOCK (ECM), MEDIUM-POWER MODE, PIC12F1840 ONLY

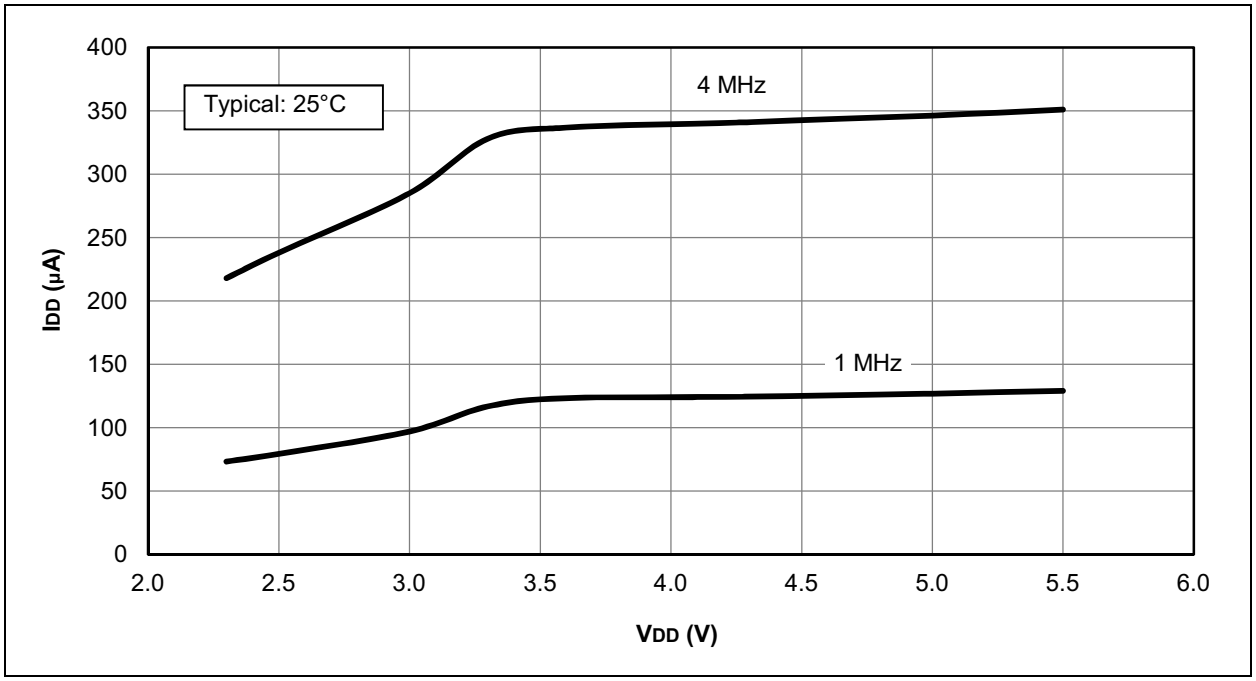


FIGURE 31-14: IDD MAXIMUM, EXTERNAL CLOCK (ECM), MEDIUM-POWER MODE, PIC12F1840 ONLY

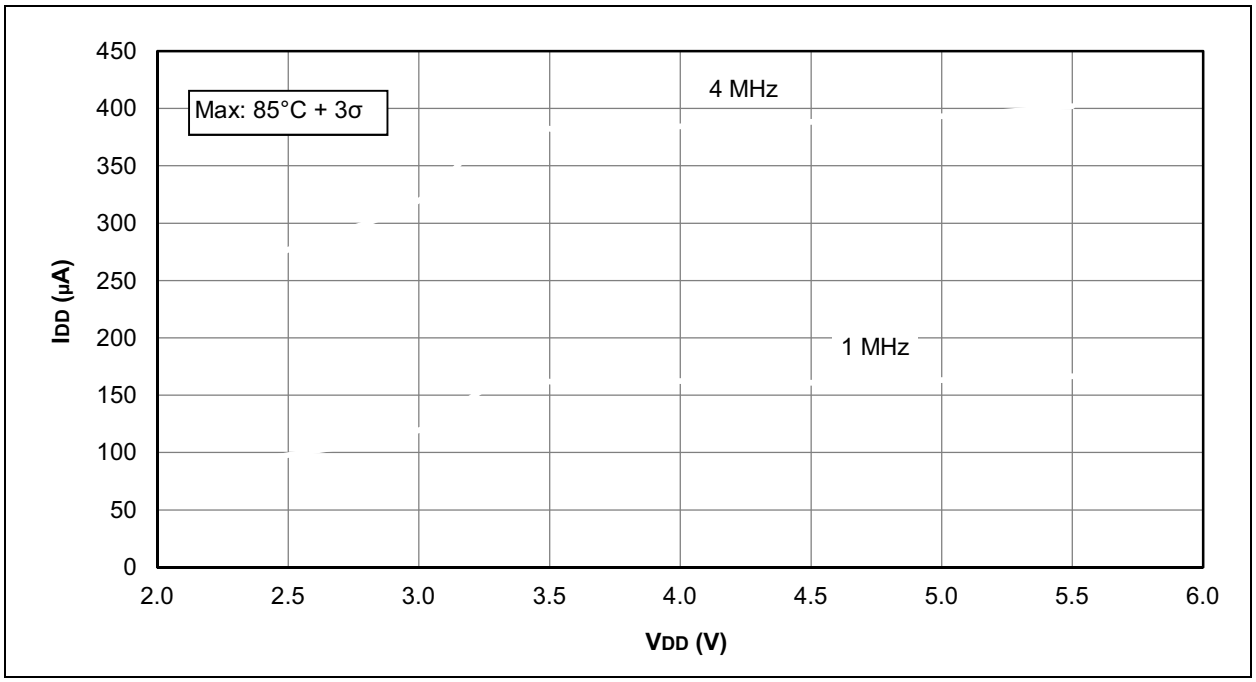


FIGURE 31-27: IDD TYPICAL, HS OSCILLATOR, PIC12LF1840 ONLY

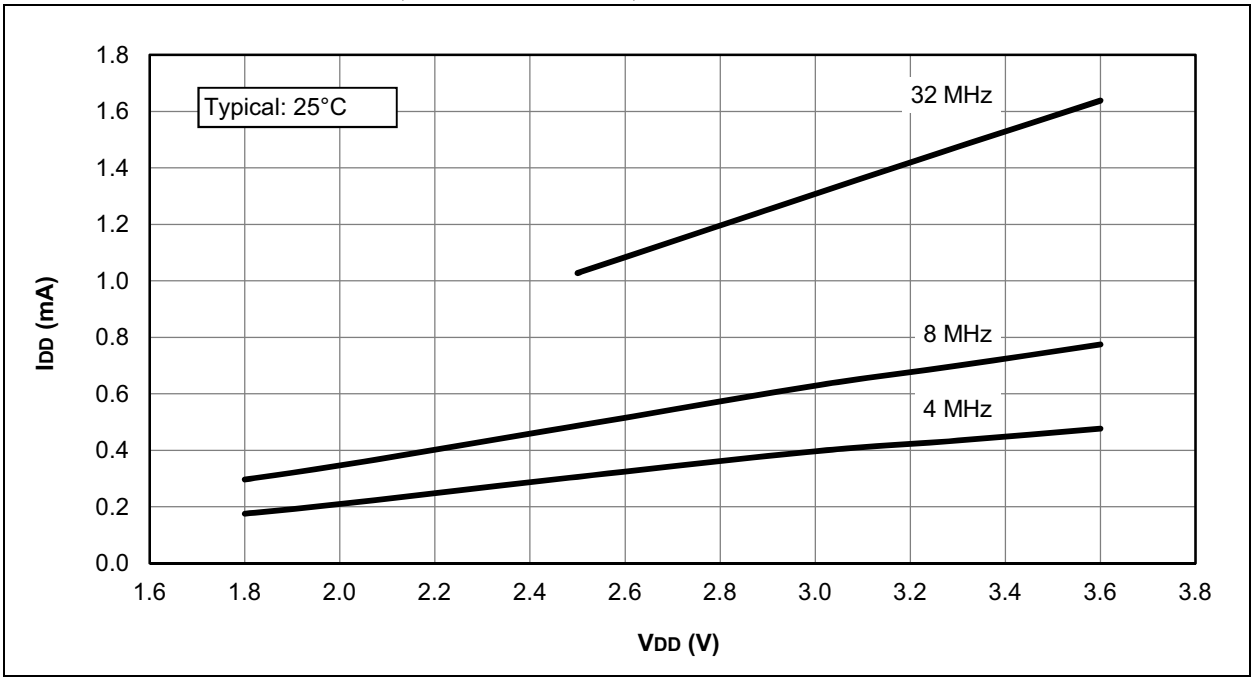


FIGURE 31-28: IDD MAXIMUM, HS OSCILLATOR, PIC12LF1840 ONLY

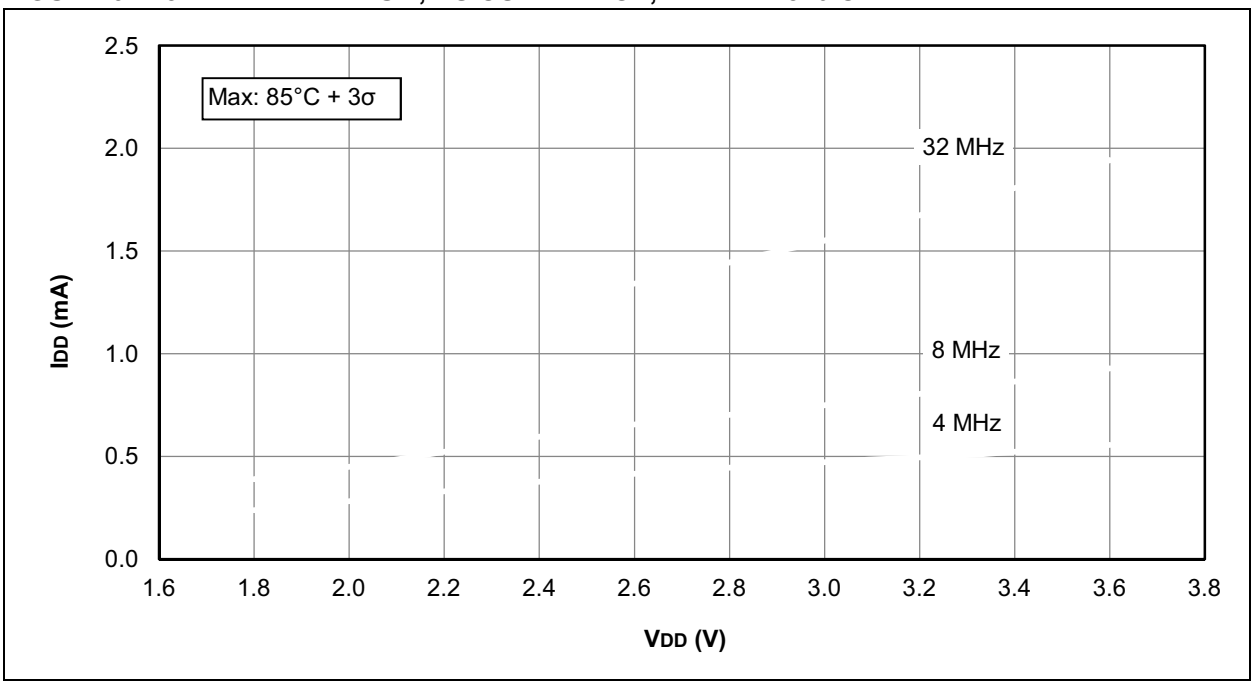


FIGURE 31-41: V_{OH} vs. I_{OH} OVER TEMPERATURE, V_{DD} = 5.5V, PIC12F1840 ONLY

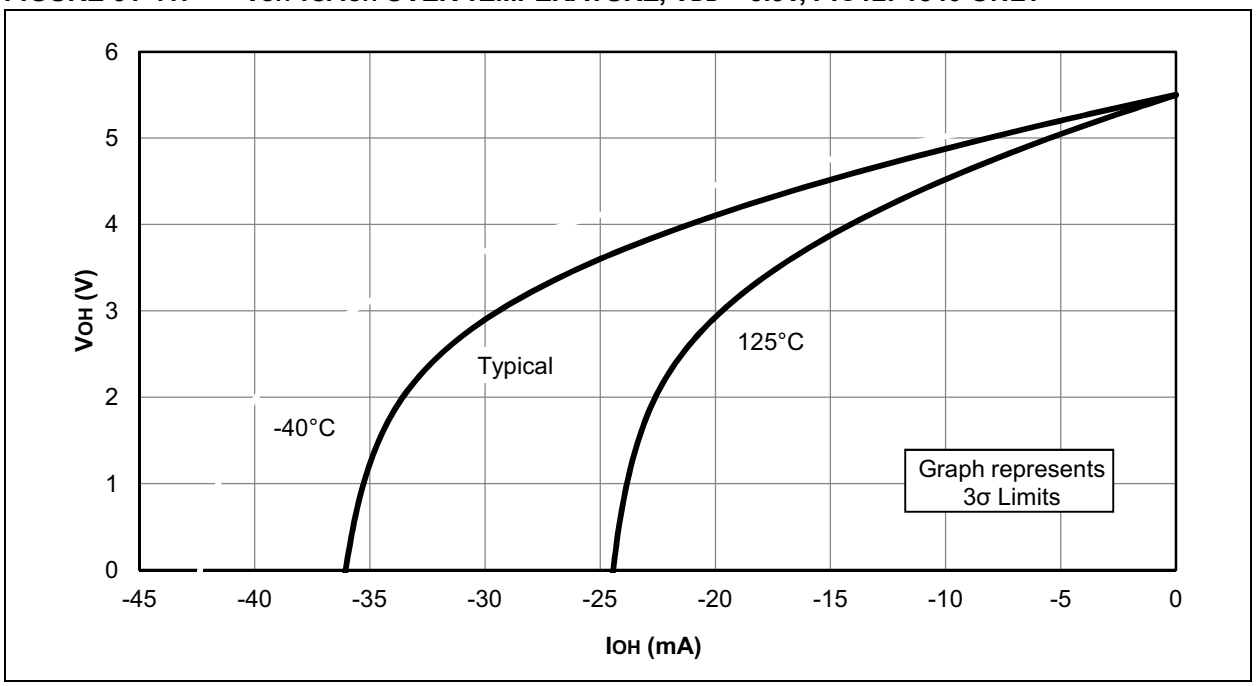


FIGURE 31-42: V_{OL} vs. I_{OL} OVER TEMPERATURE, V_{DD} = 5.5V, PIC12F1840 ONLY