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	20MHz
Connectivity	I²C, SPI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	15
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 6x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lc771-i-so

FIGURE 2-3: REGISTER FILE MAP

Bank 0	File Address	Bank 1	File Address	Bank 2	File Address	Bank 3	File Address
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
	07h		87h		107h		187h
	08h		88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	PMDATL	10Ch	PMCON1	18Ch
PIR2	0Dh	PIE2	8Dh	PMADRL	10Dh		18Dh
TMR1L	0Eh	PCON	8Eh	PMDATH	10Eh		18Eh
TMR1H	0Fh		8Fh	PMADRH	10Fh		18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h	WPUB	95h		115h		195h
CCPR1H	16h	IOCB	96h		116h		196h
CCP1CON	17h	P1DEL	97h		117h		197h
	18h		98h		118h		198h
	19h		99h		119h		199h
	1Ah		9Ah		11Ah		19Ah
	1Bh	REFCON	9Bh		11Bh		19Bh
	1Ch	LVDCON	9Ch		11Ch		19Ch
	1Dh	ANSEL	9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Register 96 Bytes	20h	General Purpose Register 80 Bytes	A0h	General Purpose Register 80 Bytes	120h		1A0h
			EFh		16Fh		1EFh
		accesses 70h-7Fh	F0h	accesses 70h - 7Fh	170h	accesses 70h - 7Fh	1F0h
	7Fh		FFh		17Fh		1FFh

Unimplemented data memory locations, read as '0'. (shaded)

* Not a physical register.

2.2.2.4 PIE1 REGISTER

This register contains the individual enable bits for the peripheral interrupts.

Note:	Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

REGISTER 2-4: PERIPHERAL INTERRUPT ENABLE REGISTER 1 (PIE1: 8Ch)

U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

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

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

bit 5-4 **Unimplemented:** Read as '0'

bit 3 **SSPIE:** Synchronous Serial Port Interrupt Enable bit
1 = Enables the SSP interrupt
0 = Disables the SSP interrupt

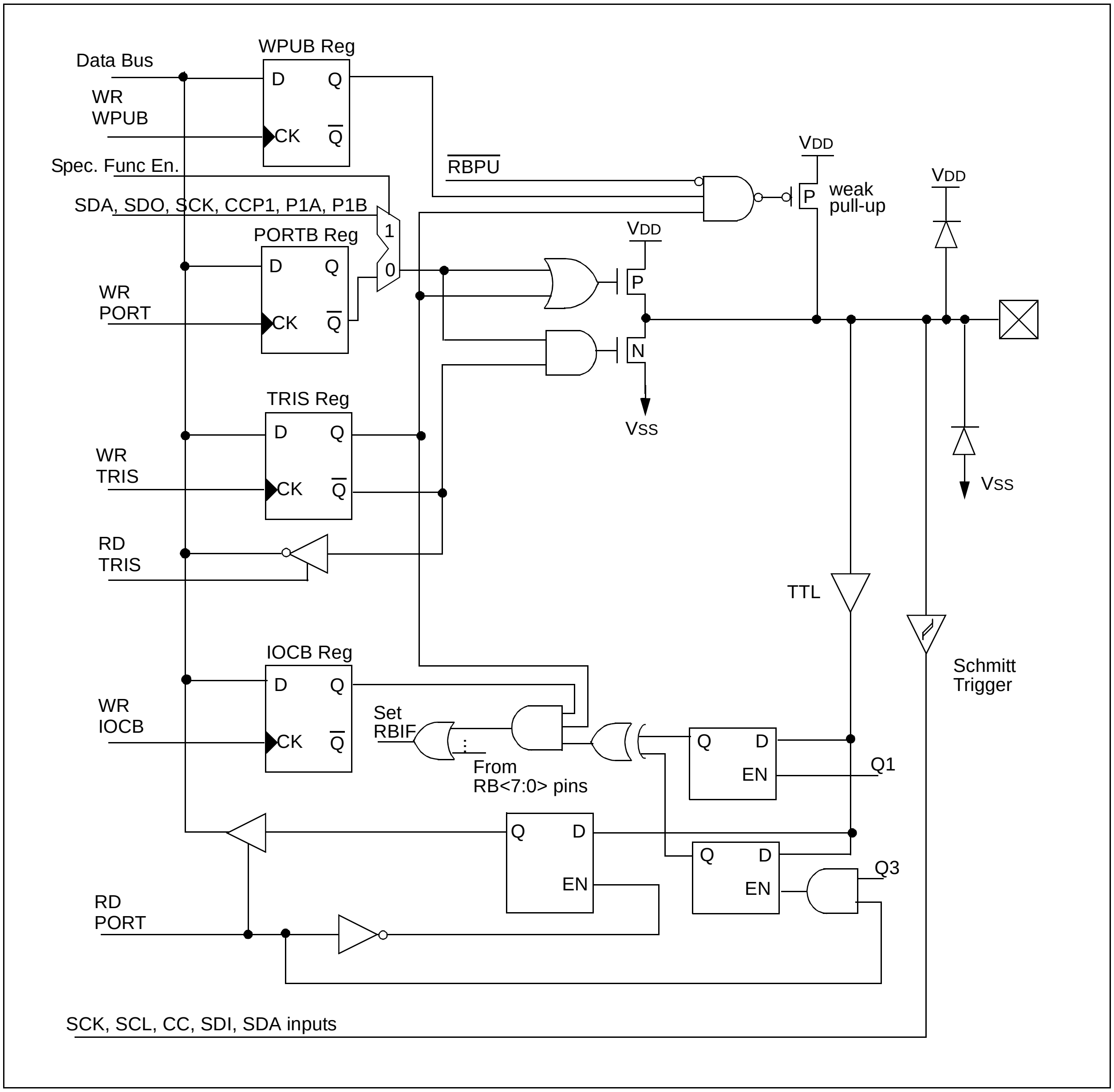
bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

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

bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

FIGURE 3-8: BLOCK DIAGRAM OF RB2/SCK/SCL, RB3/CCP1/P1A, RB4/SDI/SDA, RB5/SDO/P1B



7.0 TIMER2 MODULE

The Timer2 module timer has the following features:

- 8-bit timer (TMR2 register)
- 8-bit period register (PR2)
- Readable and writable (Both registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR2 match of PR2
- SSP module optional use of TMR2 output to generate clock shift

Timer2 has a control register, shown in Register 7-1. Timer2 can be shut off by clearing control bit TMR2ON (T2CON<2>) to minimize power consumption.

Figure 7-1 is a simplified block diagram of the Timer2 module.

Additional information on timer modules is available in the PIC Mid-Range MCU Family Reference Manual, (DS33023).

7.1 Timer2 Operation

Timer2 can be used as the PWM time-base for PWM mode of the ECCP module.

The TMR2 register is readable and writable, and is cleared on any device RESET.

The input clock ($F_{OSC}/4$) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits T2CKPS<1:0> (T2CON<1:0>).

The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit TMR2IF, (PIR1<1>)).

The prescaler and postscaler counters are cleared when any of the following occurs:

- a write to the TMR2 register
- a write to the T2CON register
- any device RESET (Power-on Reset, $\overline{MCLR}$ Reset, Watchdog Timer Reset, or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

REGISTER 7-1: TIMER2 CONTROL REGISTER (T2CON1: 12h)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

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

bit 6-3 **TOUTPS<3:0>:** Timer2 Output Postscale Select bits
`0000` = 1:1 Postscale
`0001` = 1:2 Postscale
•
•
•
`1111` = 1:16 Postscale

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

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

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
- n = Value at POR	’1’ = Bit is set	’0’ = Bit is cleared	x = Bit is unknown

9.1.4 MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2, Figure 9-2) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI module is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a “line activity monitor”.

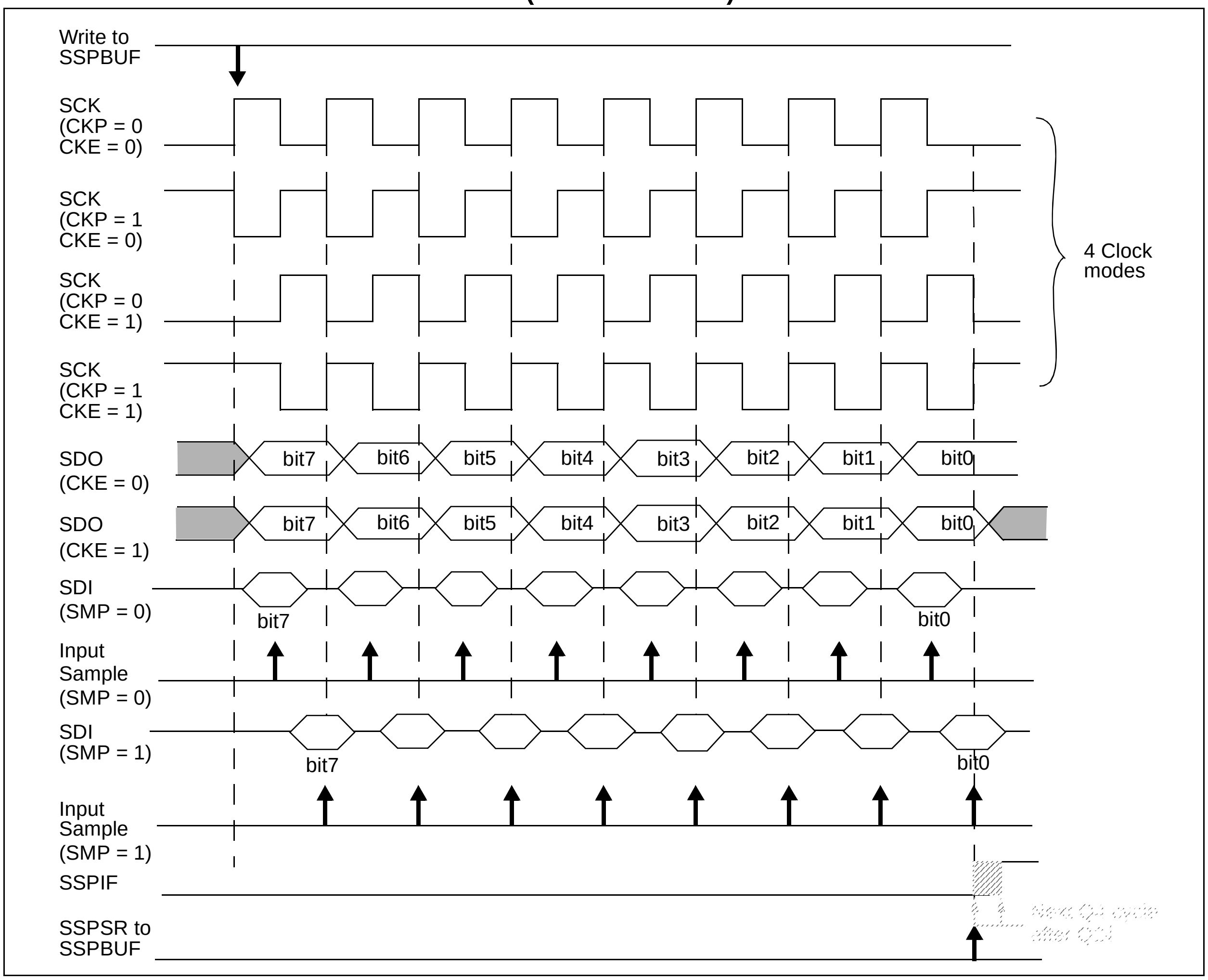
The clock polarity is selected by appropriately programming bit CKP (SSPCON<4>). This then would give waveforms for SPI communication as shown in Figure 9-3, Figure 9-5 and Figure 9-6, where the MSb is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- FOSC/4 (or TCY)
- FOSC/16 (or 4 • TCY)
- FOSC/64 (or 16 • TCY)
- Timer2 output/2

This allows a maximum bit clock frequency (at 20 MHz) of 8.25 MHz.

Figure 9-3 shows the waveforms for Master mode. When CKE = 1, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

FIGURE 9-3: SPI MODE WAVEFORM (MASTER MODE)



 Advance Information

9.1.5 SLAVE MODE

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched the interrupt flag bit SSPIF (PIR1<3>) is set.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in SLEEP mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from SLEEP.

9.1.6 SLAVE SELECT SYNCHRONIZATION

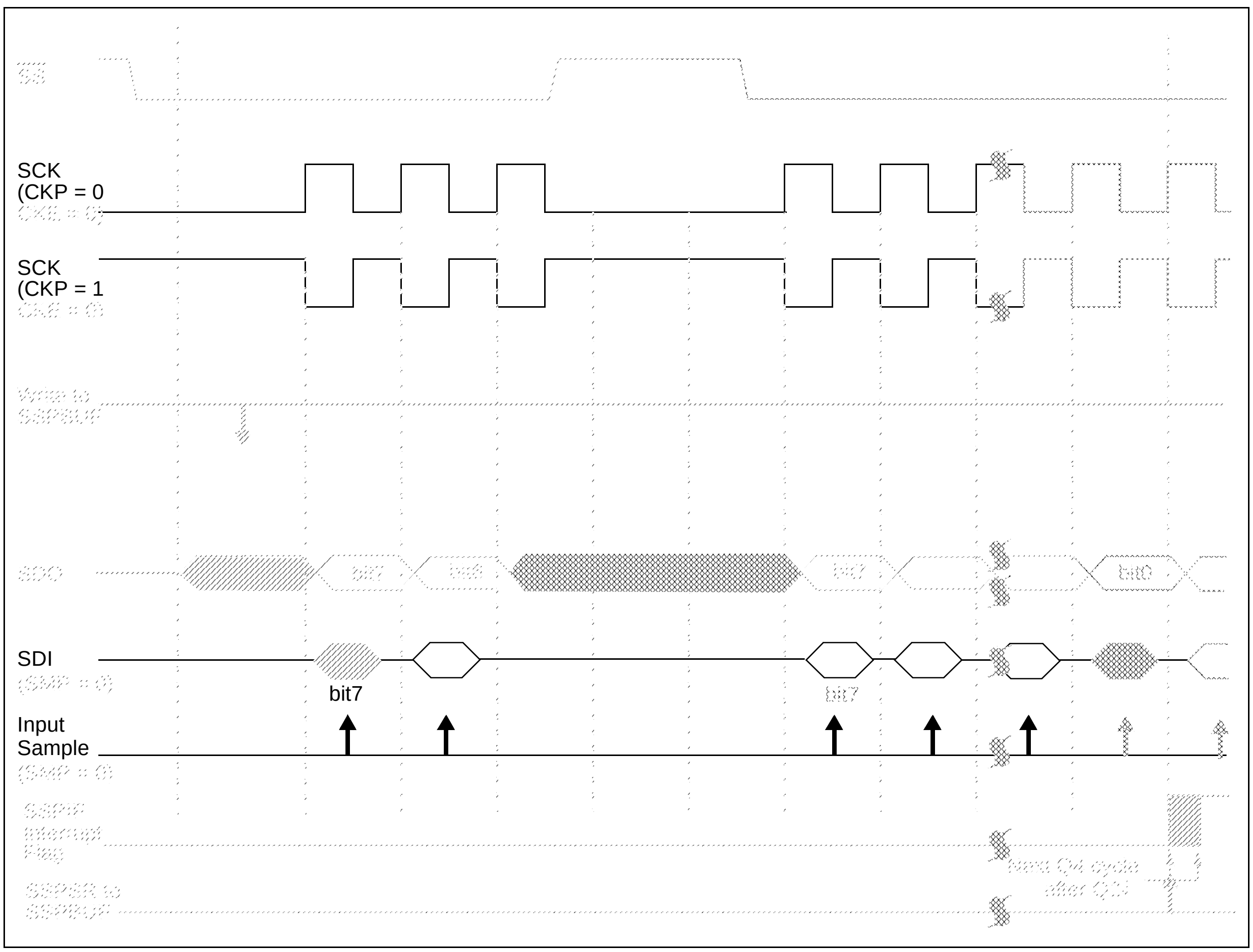
The $\overline{\text{SS}}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{\text{SS}}$ pin control enabled (SSPCON<3:0> = `0100`). The pin must not be driven low for the $\overline{\text{SS}}$ pin to function as an input. TRISB<1> must be set. When the $\overline{\text{SS}}$ pin is low, transmission and reception are enabled and the SDO pin is driven. When the $\overline{\text{SS}}$ pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte, and becomes a floating output. External pull-up/ pull-down resistors may be desirable, depending on the application.

> **Note 1:** When the SPI module is in Slave mode with $\overline{\text{SS}}$ pin control enabled, (SSPCON<3:0> = `0100`) the SPI module will RESET if the $\overline{\text{SS}}$ pin is set to V_{DD}.
>
> **2:** If the SPI is used in Slave Mode with CKE = '1', then $\overline{\text{SS}}$ pin control must be enabled.

When the SPI module RESETS, the bit counter is forced to 0. This can be done by either forcing the $\overline{\text{SS}}$ pin to a high level or clearing the SSPEN bit.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver, the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.

FIGURE 9-4: SLAVE SYNCHRONIZATION WAVEFORM



9.2 MSSP I^{2}C Operation

The MSSP module in I^{2}C mode fully implements all master and slave functions (including general call support) and provides interrupts on START and STOP bits in hardware to determine when the bus is free (multi-master function). The MSSP module implements the Standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used to transfer data. They are the SCL pin (clock) and the SDA pin (data). The MSSP module functions are enabled by setting SSP Enable bit SSPEN (SSPCON<5>). The SCL and SDA pins are "glitch" filtered when operating as inputs. This filter functions in both the 100 kHz and 400 kHz modes. When these pins operate as outputs in the 100 kHz mode, there is a slew rate control of the pin that is independent of device frequency.

Before selecting any I^{2}C mode, the SCL and SDA pins must be programmed as inputs by setting the appropriate TRIS bits. This allows the MSSP module to configure and drive the I/O pins as required by the I^{2}C protocol.

The MSSP module has six registers for I^{2}C operation. They are listed below.

- SSP Control Register (SSPCON)
- SSP Control Register2 (SSPCON2)
- SSP STATUS Register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- SSP Shift Register (SSPSR) - Not directly accessible
- SSP Address Register (SSPADD)

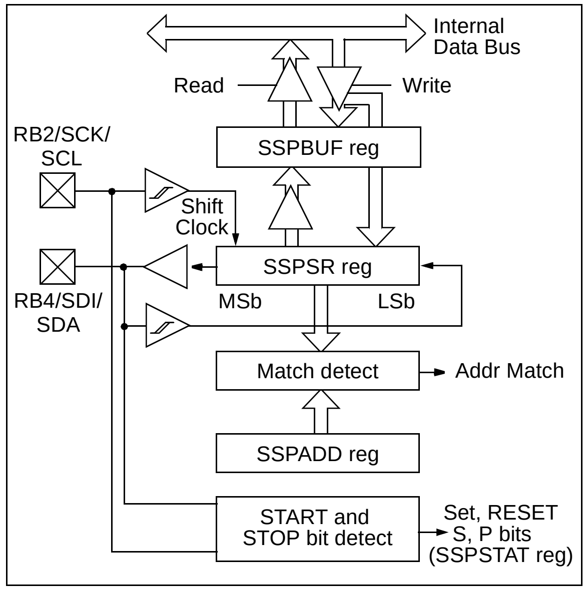
The SSPCON register allows for control of the I^{2}C operation. Four mode selection bits (SSPCON<3:0>) configure the MSSP as any one of the following I^{2}C modes:

- I^{2}C Slave mode (7-bit address)
- I^{2}C Slave mode (10-bit address)
- I^{2}C Master mode
  SCL Freq = FOSC / [4 • (SSPADD + 1)]
- I^{2}C Slave mode with START and STOP interrupts (7-bit address)
- I^{2}C Slave mode with START and STOP interrupts (10-bit address)
- Firmware Controlled Master mode

The SSPSTAT register gives the status of the data transfer. This information includes detection of a START (S) or STOP (P) bit. It specifies whether the received byte was data or address, if the next byte is the completion of 10-bit address, and if this will be a read or write data transfer.

SSPBUF is the register to which the transfer data is written, and from which the transfer data is read. The SSPSR register shifts the data in or out of the device. In receive operations, the SSPBUF and SSPSR create a doubled, buffered receiver. This allows reception of the next byte to begin before reading the last byte of received data. When the complete byte is received, it is transferred from the SSPSR register to the SSPBUF register and flag bit SSPIF is set. If another complete byte is received before the SSPBUF register is read a receiver overflow occurs, in which case, the SSPOV bit (SSPCON<6>) is set and the byte in the SSPSR is lost.

FIGURE 9-7: I^{2}C SLAVE MODE BLOCK DIAGRAM



9.2.1 UPWARD COMPATIBILITY WITH SSP MODULE

The MSSP module includes three SSP modes of operation to maintain upward compatibility with the SSP module. These modes are:

- Firmware controlled Master mode (slave idle)
- 7-bit Slave mode with START and STOP condition interrupts.
- 10-bit Slave mode with START and STOP condition interrupts.

The firmware controlled Master mode enables the START and STOP condition interrupts but all other I^{2}C functions are generated through firmware including:

- Generating the START and STOP conditions
- Generating the SCL clock
- Supplying the SDA bits in the proper time and phase relationship to the SCL signal.

In firmware controlled Master mode, the SCL and SDA lines are manipulated by clearing and setting the corresponding TRIS bits. The output level is always low irrespective of the value(s) in the PORT register. A ‘1’ is output by setting the TRIS bit and a ‘0’ is output by clearing the TRIS bit

The 7-bit and 10-bit Slave modes with START and STOP condition interrupts operate identically to the MSSP Slave modes except that START and STOP conditions generate SSPIF interrupts.

9.2.2.4 SLAVE TRANSMISSION

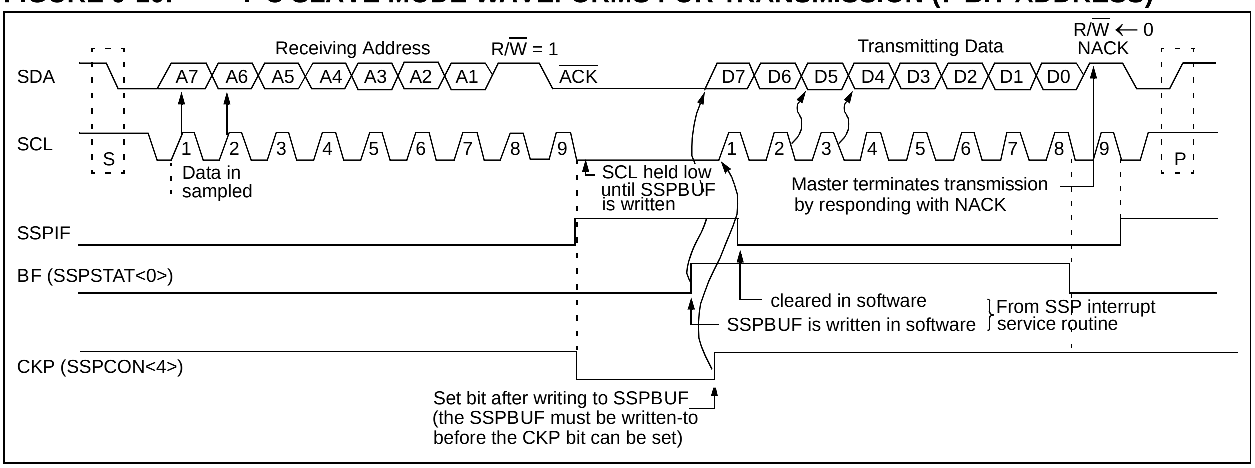
When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register on the falling edge of the eighth SCL pulse. The $\overline{ACK}$ pulse will be sent on the ninth bit, and the SCL pin is held low. The slave module automatically stretches the clock by holding the SCL line low so that the master will be unable to assert another clock pulse until the slave is finished preparing the transmit data. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. The CKP bit (SSPCON<4>) must then be set to release the SCL pin from the forced low condition. The eight data bits are shifted out on the falling edges of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 9-10).

The $\overline{ACK}$ or NACK signal from the master-receiver is latched on the rising edge of the ninth SCL input pulse. The master-receiver terminates slave transmission by sending a NACK. If the SDA line is high (NACK), then the data transfer is complete. When the NACK is latched by the slave, the slave logic is RESET which also resets the R/$\overline{W}$ bit to '0'. The slave module then monitors for another occurrence of the START bit. The slave firmware knows not to load another byte into the SSPBUF register by sensing that the buffer is empty (BF = 0) and the R/$\overline{W}$ bit has gone low. If the SDA line is low ($\overline{ACK}$), the R/$\overline{W}$ bit remains high indicating that the next transmit data must be loaded into the SSPBUF register.

An MSSP interrupt (SSPIF flag) is generated for each data transfer byte on the falling edge of the ninth clock pulse. The SSPIF flag bit must be cleared in software. The SSPSTAT register is used to determine the status of the byte transfer.

For more information about the I^{2}C Slave mode, refer to Application Note AN734, "*Using the PIC® SSP for Slave I^{2}C™ Communication*".

FIGURE 9-10: I^{2}C SLAVE MODE WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)



9.2.4 SLEEP OPERATION

While in SLEEP mode, the I^2C slave module can receive addresses or data. When an address match or complete byte transfer occurs, it wakes the processor from SLEEP (if the SSP interrupt bit is enabled).

9.2.5 EFFECTS OF A RESET

A RESET disables the MSSP module and terminates the current transfer.

9.2.6 MASTER MODE

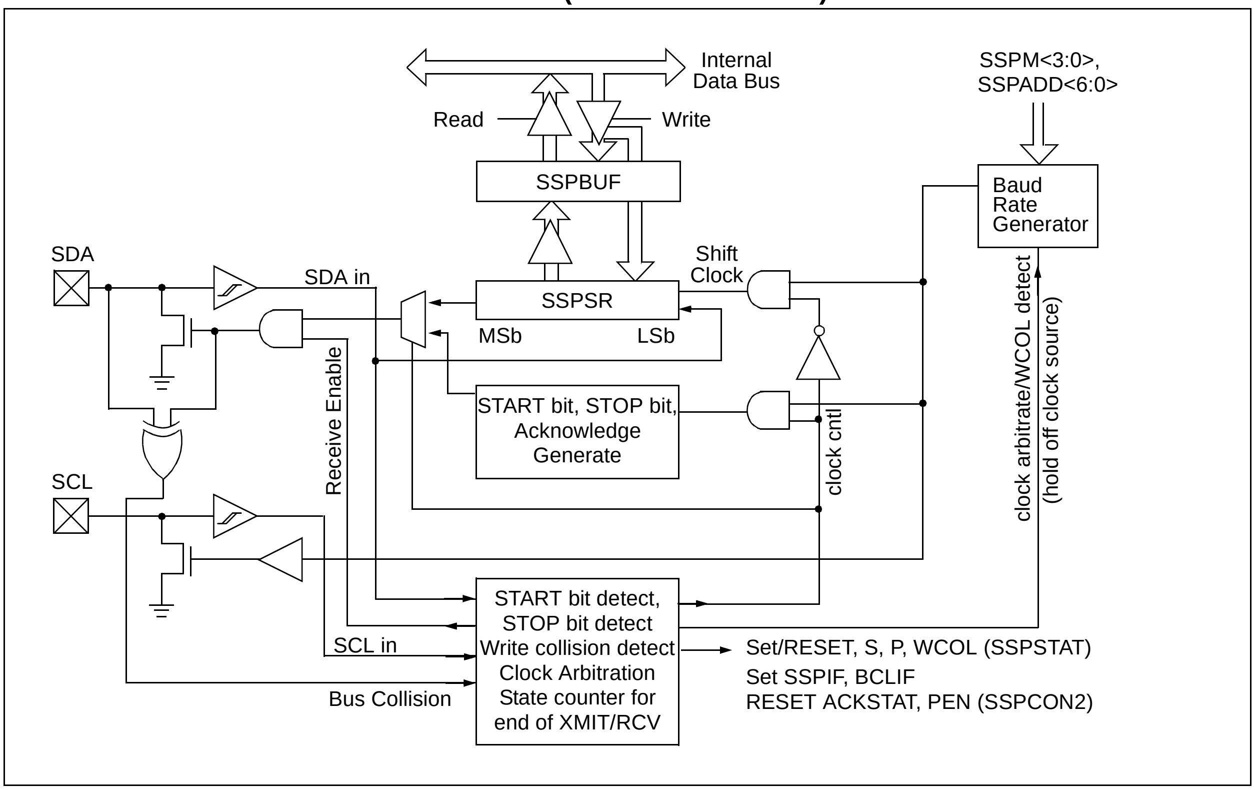
Master mode operation supports interrupt generation on the detection of the START and STOP conditions. The STOP (P) and START (S) bits are cleared from a RESET or when the MSSP module is disabled. Control of the I^2C bus may be taken when the P bit is set or the bus is idle with both the S and P bits clear.

In Master mode, the SCL and SDA lines are manipulated by the MSSP hardware.

The following events will cause SSP Interrupt Flag bit (SSPIF) to be set (SSP Interrupt, if enabled):

- START condition
- STOP condition
- Data transfer byte transmitted/received
- Acknowledge transmit
- Repeated START

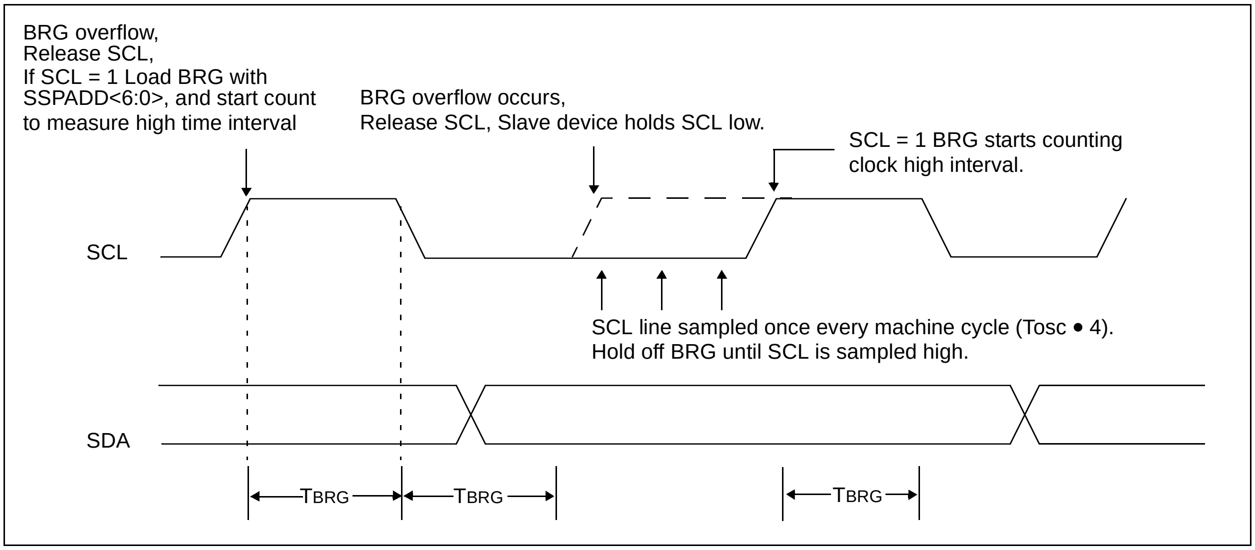
FIGURE 9-13: MSSP BLOCK DIAGRAM (I^2C MASTER MODE)



9.2.16 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit or repeated START/STOP condition, de-asserts the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the baud rate generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the baud rate generator is reloaded with the contents of SSPADD<6:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 9-22).

FIGURE 9-22: CLOCK ARBITRATION TIMING IN MASTER TRANSMIT MODE



9.2.17.2 BUS COLLISION DURING A REPEATED START CONDITION

During a Repeated START condition, a bus collision occurs if:

a) A low level is sampled on SDA when SCL goes from low level to high level.
b) SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data ’1’.

When the master module de-asserts SDA and the pin is allowed to float high, the BRG is loaded with SSPADD<6:0>, and counts down to ‘0’. The SCL pin is then de-asserted, and when sampled high, the SDA pin is sampled. If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data ’0’). If however SDA is sampled high, then the BRG is reloaded and begins counting. If SDA goes from high to low before the BRG times out, no bus collision occurs, because no two masters can assert SDA at exactly the same time.

If, however, SCL goes from high to low before the BRG times out and SDA has not already been asserted, then a bus collision occurs. In this case, another master is attempting to transmit a data ’1’ during the Repeated START condition.

If at the end of the BRG time-out both SCL and SDA are still high, the SDA pin is driven low, the BRG is reloaded, and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated START condition is complete (Figure 9-27).

FIGURE 9-27: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

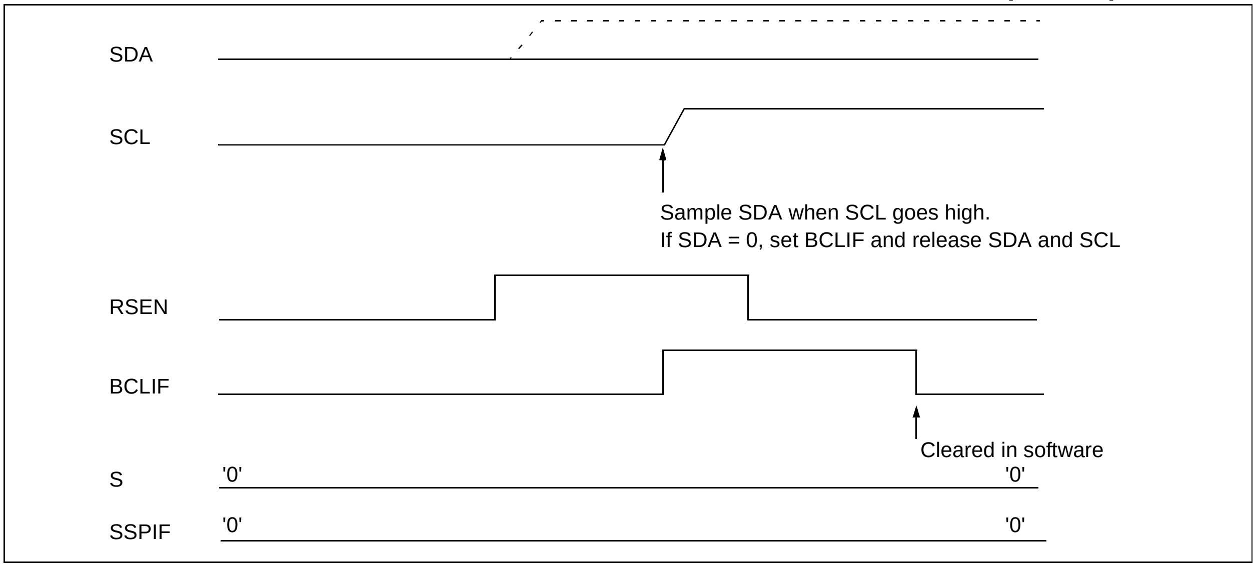
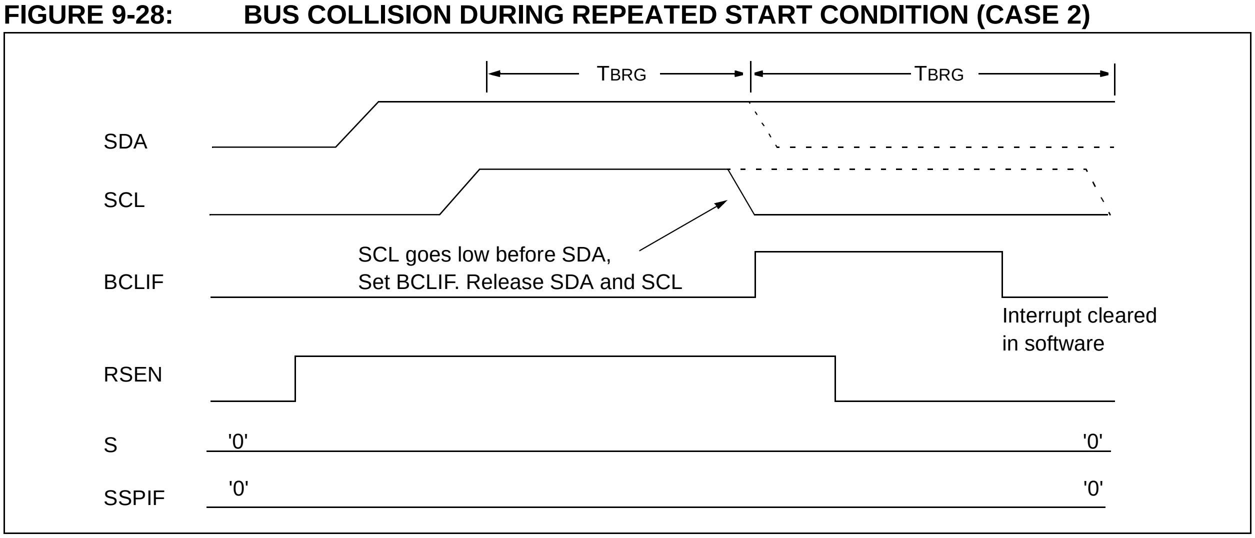


FIGURE 9-28: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



12.2 Oscillator Configurations

12.2.1 OSCILLATOR TYPES

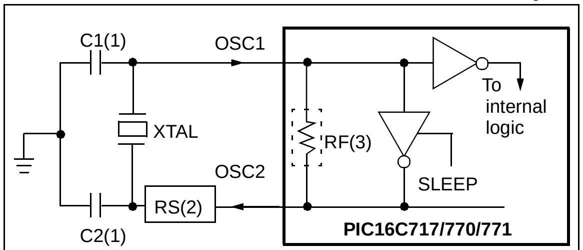
The PIC16C717/770/771 can be operated in eight different Oscillator modes. The user can program three configuration bits (FOSC<2:0>) to select one of these eight modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- ER External Resistor (with and without CLKOUT)
- INTRC Internal 4 MHz (with and without CLKOUT)
- EC External Clock

12.2.2 LP, XT AND HS MODES

In LP, XT or HS modes, a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 12-1). The PIC16C717/770/771 oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications.

FIGURE 12-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)



Note1: See Table 12-1 and Table 12-2 for recommended values of C1 and C2.
2: A series resistor (RS) may be required for AT strip cut crystals.
3: RF varies with the Crystal mode chosen.

TABLE 12-1: CERAMIC RESONATORS

Ranges Tested:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	455 kHz 2.0 MHz 4.0 MHz	68 - 100 pF 15 - 68 pF 15 - 68 pF	68 - 100 pF 15 - 68 pF 15 - 68 pF
HS	8.0 MHz 16.0 MHz	10 - 68 pF 10 - 22 pF	10 - 68 pF 10 - 22 pF
These values are for design guidance only. See notes at bottom of page.			
All resonators used did not have built-in capacitors.			

TABLE 12-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

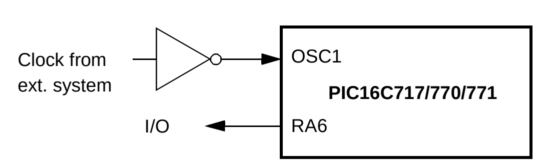
Osc Type	Crystal Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF
These values are for design guidance only. See notes at bottom of page.			

Note 1: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.
2: Higher capacitance increases the stability of oscillator but also increases the start-up time.

12.2.3 EC MODE

In applications where the clock source is external, the PIC16C717/770/771 should be programmed to select the EC (External Clock) mode. In this mode, the RA6/OSC2/CLKOUT pin is available as an I/O pin. See Figure 12-2 for illustration.

FIGURE 12-2: EXTERNAL CLOCK INPUT OPERATION (EC OSC CONFIGURATION)



RETFIE — **Return from Interrupt**

Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS $\rightarrow$ PC, 1 $\rightarrow$ GIE
Status Affected:	None

RETLW — **Return with Literal in W**

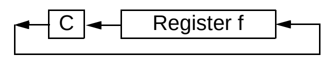
Syntax:	[*label*] RETLW k
Operands:	$0 \leq k \leq 255$
Operation:	k $\rightarrow$ (W); TOS $\rightarrow$ PC
Status Affected:	None
Description:	The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two cycle instruction.

RETURN — **Return from Subroutine**

Syntax:	[*label*] RETURN
Operands:	None
Operation:	TOS $\rightarrow$ PC
Status Affected:	None
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two cycle instruction.

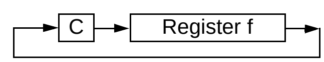
RLF — **Rotate Left f through Carry**

Syntax:	[*label*] RLF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C
Description:	The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.



RRF — **Rotate Right f through Carry**

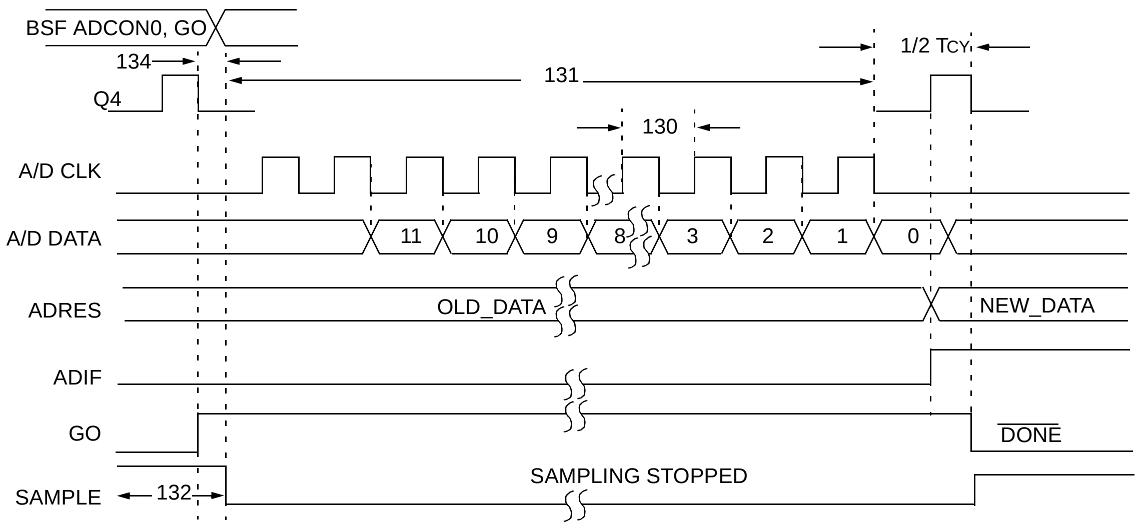
Syntax:	[*label*] RRF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C
Description:	The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.



SLEEP

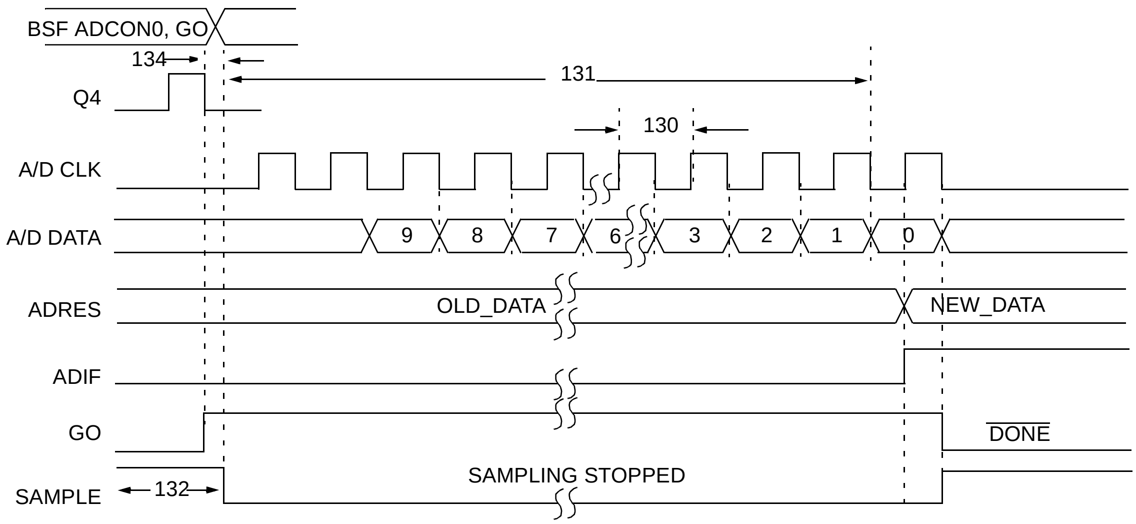
Syntax:	[*label*] SLEEP
Operands:	None
Operation:	00h $\rightarrow$ WDT, 0 $\rightarrow$ WDT prescaler, 1 $\rightarrow$ $\overline{TO}$, 0 $\rightarrow$ $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	The power-down status bit, $\overline{PD}$ is cleared. Time-out status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared. The processor is put into SLEEP mode with the oscillator stopped. See Section 12.8 for more details.

FIGURE 15-14: PIC16C770/771 AND PIC16LC770/771 A/D CONVERSION TIMING (NORMAL MODE)



Note 1: If the A/D RC clock source is selected, a time of TCY is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.

FIGURE 15-17: PIC16C717 A/D CONVERSION TIMING (SLEEP MODE)



Note 1: If the A/D RC clock source is selected, a time of T_{CY} is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.

TABLE 15-16: PIC16C717 AND PIC16LC717 A/D CONVERSION REQUIREMENT (SLEEP MODE)

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
130*(3)	T_{AD}	A/D clock period	3.0	6.0	9.0	μs	ADCS<1:0> = 11 (A/D RC mode) At V_{DD} = 3.0V
			2.0	4.0	6.0	μs	At V_{DD} = 5.0V
131*	T_{CNV}	Conversion time (not including acquisition time) **(Note 1)**	—	11T_{AD}	—	—	
132*	T_{ACQ}	Acquisition Time	**(Note 2)**	11.5	—	μs	
			5*	—	—	μs	The minimum time is the amplifier settling time. This may be used if the “new” input voltage has not changed by more than 1LSb (i.e., 1mV @ 4.096V) from the last sampled voltage (as stated on C_{HOLD}).
134*	T_{GO}	Q4 to A/D clock start	—	$T_{OSC}/2 + T_{CY}$	—	—	If the A/D RC clock source is selected, a time of T_{CY} is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: ADRES register may be read on the following T_{CY} cycle.

2: See Section 11.6 for minimum conditions.

3: These numbers multiplied by 8 if V_{RH} or V_{RL} is selected as A/D reference.

FIGURE 16-16: TYPICAL IDD VS. VDD (INTRC 4 MHz MODE)

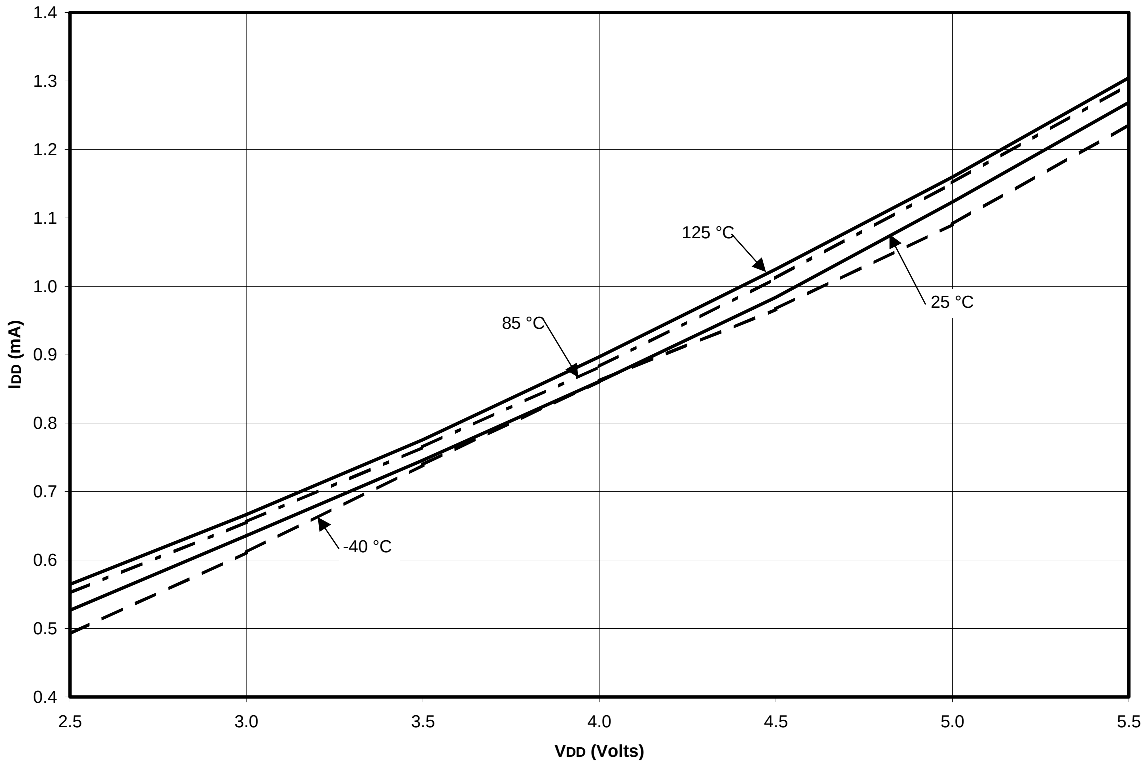


FIGURE 16-17: INTERNAL RC FOSC VS. VDD OVER TEMPERATURE (4 MHz)

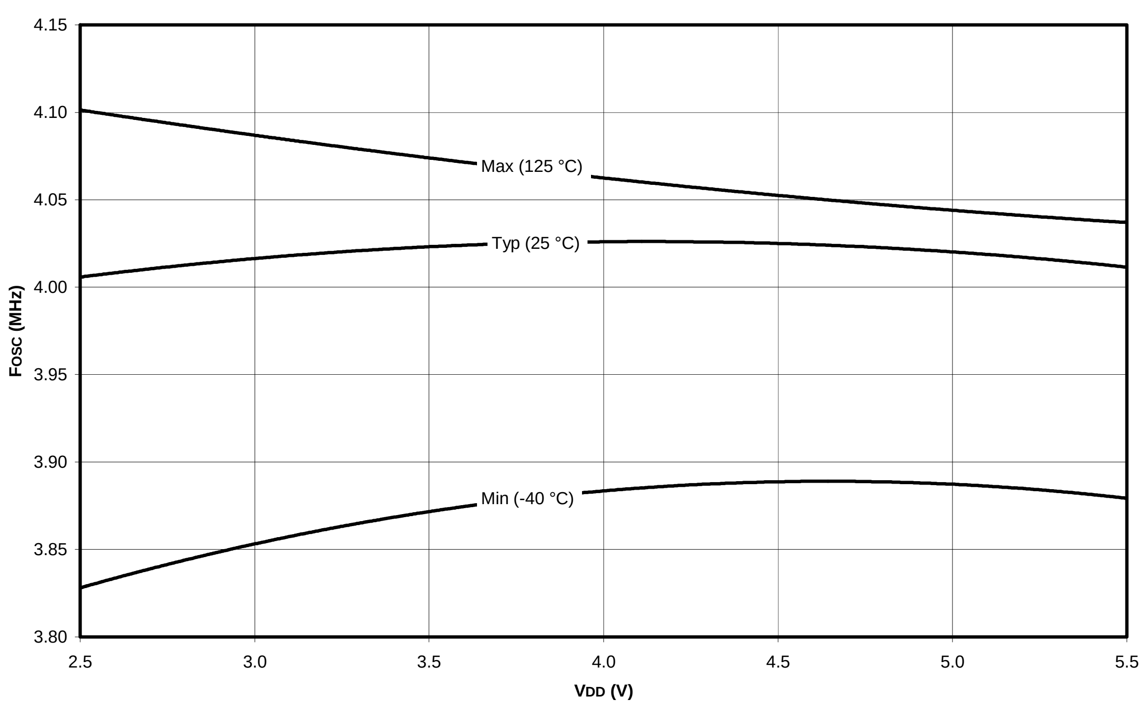


FIGURE 16-24: TYPICAL AND MAXIMUM ΔILVD VS. VDD (-40°C TO +125°C) (LVD TRIP = 4.5V)

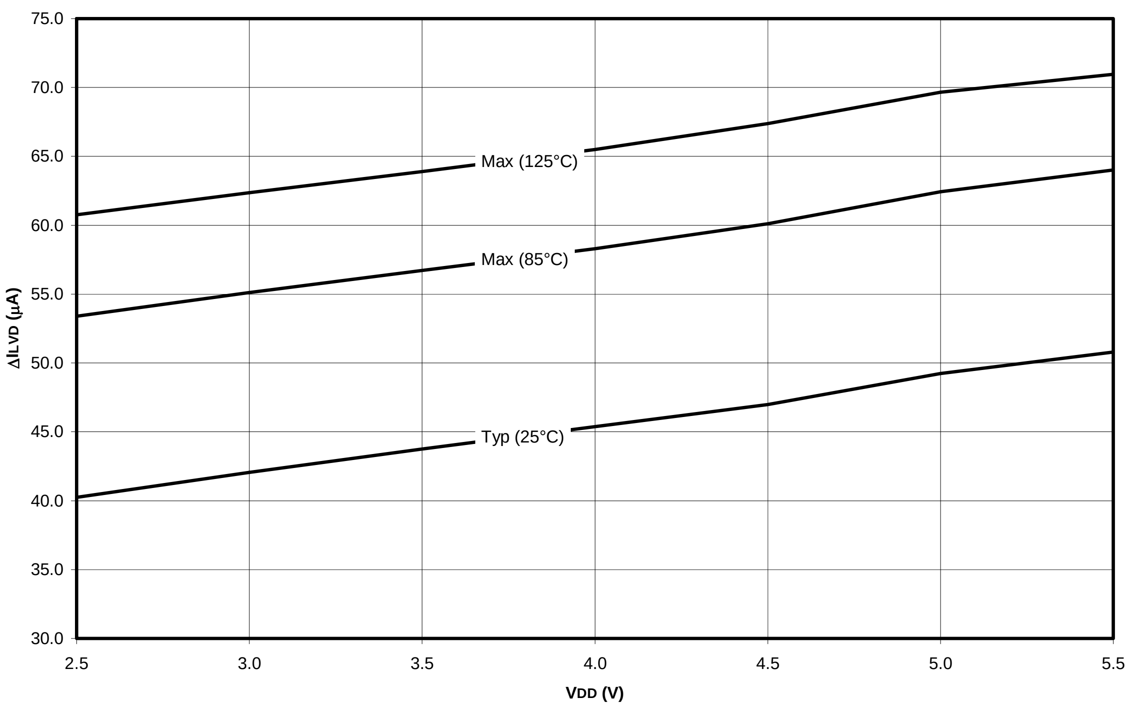


FIGURE 16-25: TYPICAL AND MAXIMUM ΔIBOR VS. VDD (-40°C TO +125°C) (VBOR = 2.5V)

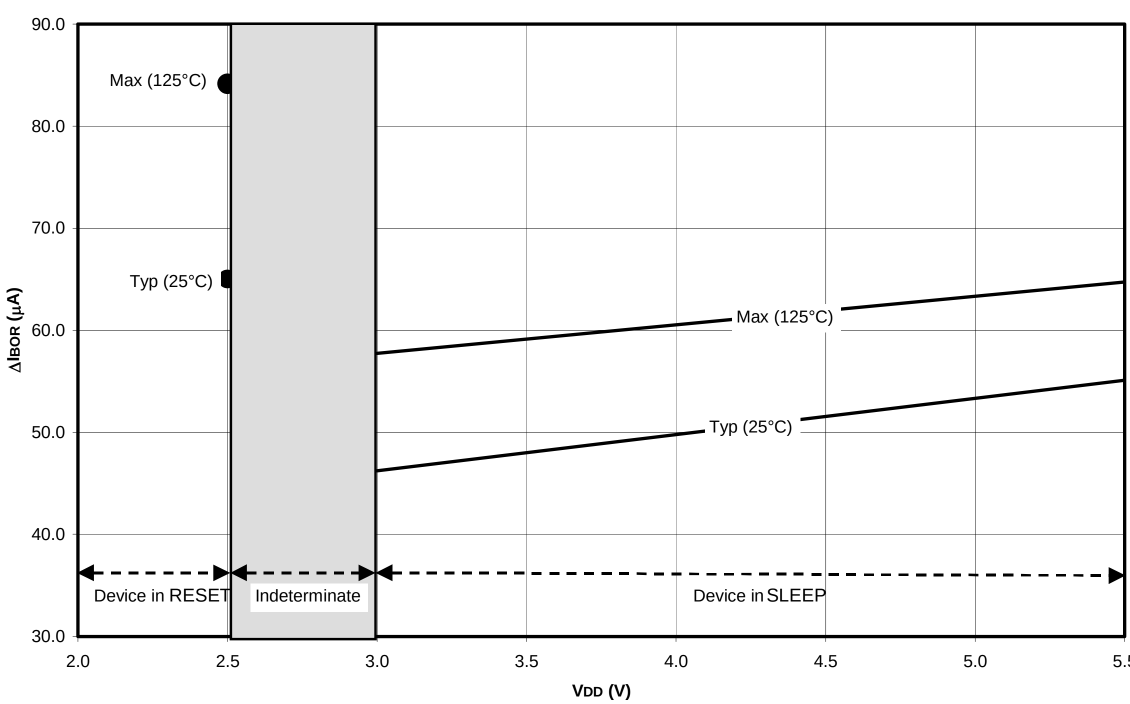


FIGURE 16-32: MINIMUM AND MAXIMUM VIH/VIL VS. VDD (ST INPUT,-40°C TO +125°C)

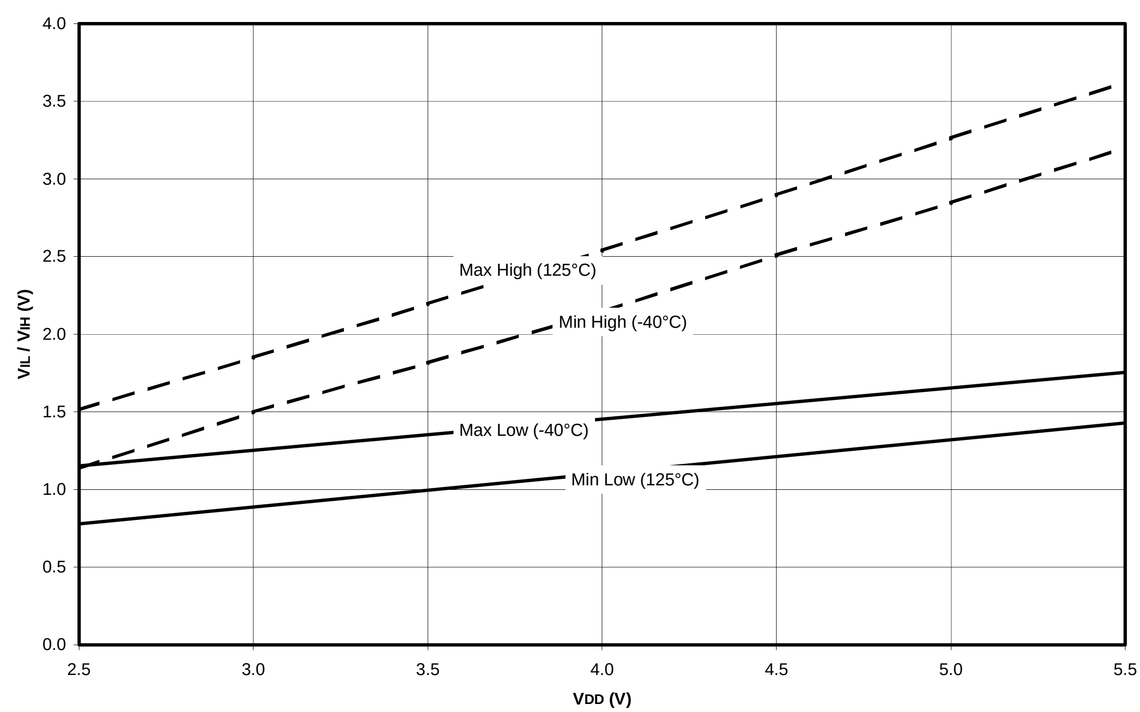
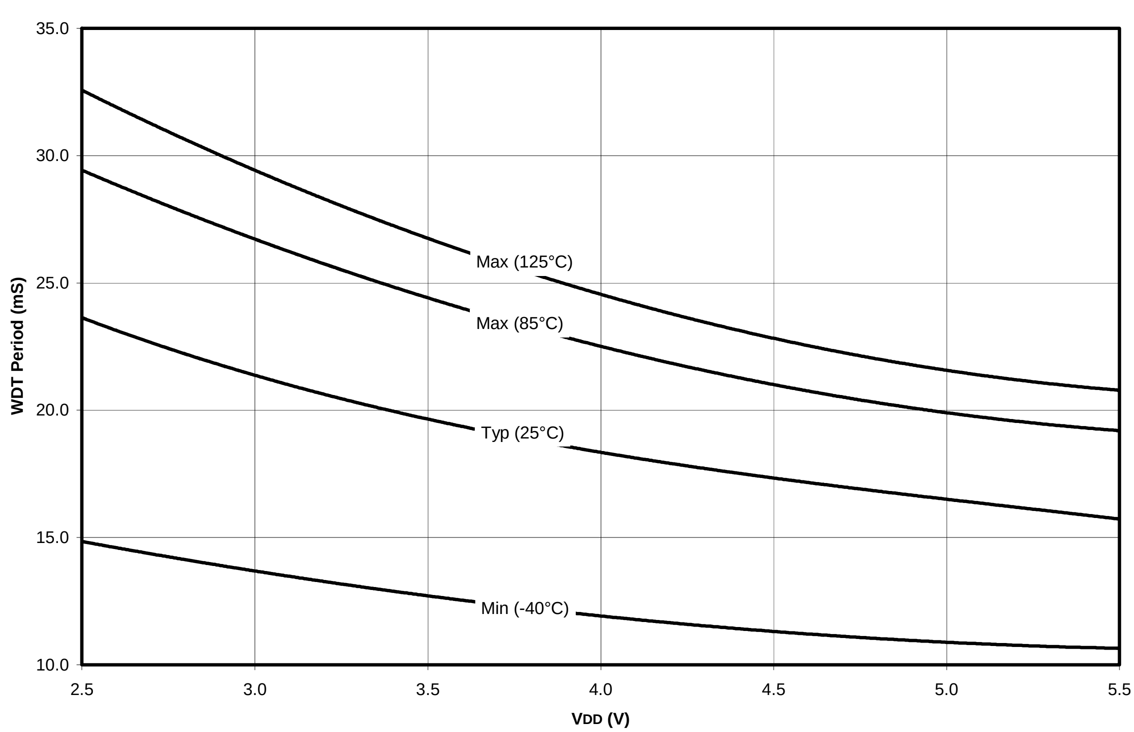


FIGURE 16-33: TYPICAL, MINIMUM AND MAXIMUM WDT PERIOD VS. VDD (-40°C TO +125°C)



17.6 20-Lead Ceramic Dual In-line with Window (JW) – 300 mil (CERDIP)

DRAWING NOT AVAILABLE