



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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	33
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.59x16.59)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f448t-i-l

8.0 INTERRUPTS

The PIC18FXX8 devices have multiple interrupt sources and an interrupt priority feature that allows each interrupt source to be assigned a high priority level or a low priority level. The high priority interrupt vector is at 000008h and the low priority interrupt vector is at 000018h. High priority interrupt events will override any low priority interrupts that may be in progress.

There are 13 registers that are used to control interrupt operation. These registers are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2, PIR3
- PIE1, PIE2, PIE3
- IPR1, IPR2, IPR3

It is recommended that the Microchip header files, supplied with MPLAB® IDE, be used for the symbolic bit names in these registers. This allows the assembler/compiler to automatically take care of the placement of these bits within the specified register.

Each interrupt source has three bits to control its operation. The functions of these bits are:

- Flag bit to indicate that an interrupt event occurred
- Enable bit that allows program execution to branch to the interrupt vector address when the flag bit is set
- Priority bit to select high priority or low priority

The interrupt priority feature is enabled by setting the IPEN bit (RCON register). When interrupt priority is enabled, there are two bits that enable interrupts globally. Setting the GIEH bit (INTCON<7>) enables all interrupts. Setting the GIEL bit (INTCON register) enables all interrupts that have the priority bit cleared. When the interrupt flag, enable bit and appropriate global interrupt enable bit are set, the interrupt will vector immediately to address 000008h or 000018h, depending on the priority level. Individual interrupts can be disabled through their corresponding enable bits.

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PICmicro® mid-range devices. In Compatibility mode, the interrupt priority bits for each source have no effect. The PEIE bit (INTCON register) enables/disables all peripheral interrupt sources. The GIE bit (INTCON register) enables/disables all interrupt sources. All interrupts branch to address 000008h in Compatibility mode.

When an interrupt is responded to, the global interrupt enable bit is cleared to disable further interrupts. If the IPEN bit is cleared, this is the GIE bit. If interrupt priority levels are used, this will be either the GIEH or GIEL bit. High priority interrupt sources can interrupt a low priority interrupt.

The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (000008h or 000018h). Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bits must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

The “return from interrupt” instruction, RETFIE, exits the interrupt routine and sets the GIE bit (GIEH or GIEL if priority levels are used), which re-enables interrupts.

For external interrupt events, such as the INT pins or the PORTB input change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency is the same for one or two-cycle instructions. Individual interrupt flag bits are set regardless of the status of their corresponding enable bit or the GIE bit.

Note: Do not use the MOVFF instruction to modify any of the interrupt control registers while **any** interrupt is enabled. Doing so may cause erratic microcontroller behavior.

PIC18FXX8

REGISTER 8-2: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-1	R/W-1	R/W-1	U-0	U-0	R/W-1	U-0	R/W-1
RBPU	INTEDG0	INTEDG1	—	—	TMR0IP	—	RBIP
bit 7							bit 0

- bit 7 **RBPU**: PORTB Pull-up Enable bit
 1 = All PORTB pull-ups are disabled
 0 = PORTB pull-ups are enabled by individual port latch values
- bit 6 **INTEDG0**: External Interrupt 0 Edge Select bit
 1 = Interrupt on rising edge
 0 = Interrupt on falling edge
- bit 5 **INTEDG1**: External Interrupt 1 Edge Select bit
 1 = Interrupt on rising edge
 0 = Interrupt on falling edge
- bit 4-3 **Unimplemented**: Read as '0'
- bit 2 **TMR0IP**: TMR0 Overflow Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 1 **Unimplemented**: Read as '0'
- bit 0 **RBIP**: RB Port Change Interrupt Priority bit
 1 = High priority
 0 = Low priority

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

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global interrupt enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows software polling.

REGISTER 8-5: PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CMIF ⁽¹⁾	—	EEIF	BCLIF	LVDIF	TMR3IF	ECCP1IF ⁽¹⁾
bit 7							bit 0

- bit 7 **Unimplemented:** Read as '0'
- bit 6 **CMIF:** Comparator Interrupt Flag bit⁽¹⁾
 1 = Comparator input has changed
 0 = Comparator input has not changed
- bit 5 **Unimplemented:** Read as '0'
- bit 4 **EEIF:** EEPROM Write Operation Interrupt Flag bit
 1 = Write operation is complete (must be cleared in software)
 0 = Write operation is not complete
- bit 3 **BCLIF:** Bus Collision Interrupt Flag bit
 1 = A bus collision occurred (must be cleared in software)
 0 = No bus collision occurred
- bit 2 **LVDIF:** Low-Voltage Detect Interrupt Flag bit
 1 = A low-voltage condition occurred (must be cleared in software)
 0 = The device voltage is above the Low-Voltage Detect trip point
- bit 1 **TMR3IF:** TMR3 Overflow Interrupt Flag bit
 1 = TMR3 register overflowed (must be cleared in software)
 0 = TMR3 register did not overflow
- bit 0 **ECCP1IF:** ECCP1 Interrupt Flag bit⁽¹⁾
Capture mode:
 1 = A TMR1 (TMR3) register capture occurred (must be cleared in software)
 0 = No TMR1 (TMR3) register capture occurred
Compare mode:
 1 = A TMR1 register compare match occurred (must be cleared in software)
 0 = No TMR1 register compare match occurred
PWM mode:
 Unused in this mode.

Note 1: This bit is only available on PIC18F4X8 devices. For PIC18F2X8 devices, this bit is unimplemented and reads as '0'.

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

PIC18FXX8

9.5 PORTE, TRISE and LATE Registers

Note: This port is only available on the PIC18F448 and PIC18F458.

PORTE is a 3-bit wide, bidirectional port. PORTE has three pins (RE0/AN5/RD, RE1/AN6/WR/C1OUT and RE2/AN7/CS/C2OUT) which are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers.

Read-modify-write operations on the LATE register, read and write the latched output value for PORTE.

The corresponding Data Direction register for the port is TRISE. Setting a TRISE bit (= 1) will make the corresponding PORTE pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISE bit (= 0) will make the corresponding PORTE pin an output (i.e., put the contents of the output latch on the selected pin).

The TRISE register also controls the operation of the Parallel Slave Port through the control bits in the upper half of the register. These are shown in Register 9-1.

When the Parallel Slave Port is active, the PORTE pins function as its control inputs. For additional details, refer to **Section 10.0 “Parallel Slave Port”**.

PORTE pins are also multiplexed with inputs for the A/D converter and outputs for the analog comparators. When selected as an analog input, these pins will read as ‘0’s. Direction bits TRISE<2:0> control the direction of the RE pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

EXAMPLE 9-5: INITIALIZING PORTE

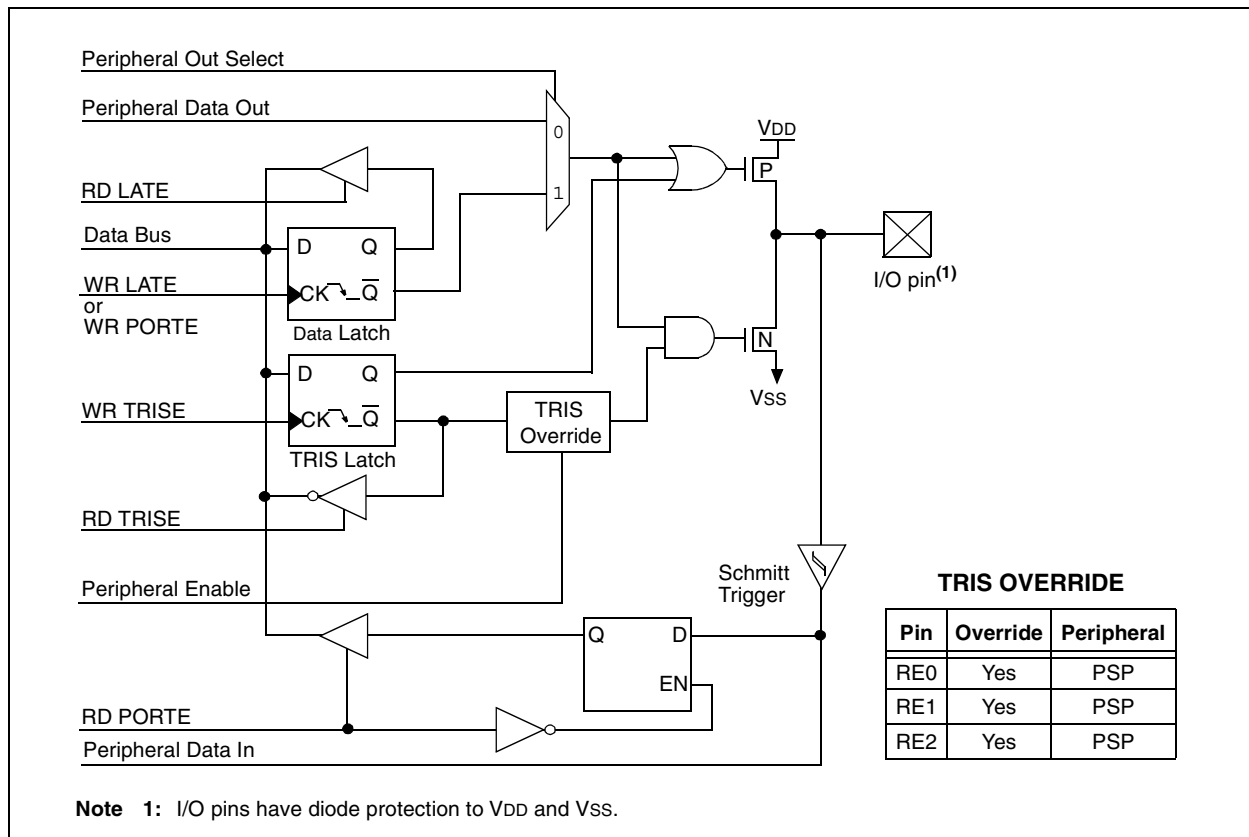
```
CLRF    PORTE      ; Initialize PORTE by
                   ; clearing output
                   ; data latches

CLRF    LATE        ; Alternate method
                   ; to clear output
                   ; data latches

MOVLW   03h        ; Value used to
                   ; initialize data
                   ; direction

MOVWF   TRISE       ; Set RE1:RE0 as inputs
                   ; RE2 as an output
                   ; (RE4=0 - PSPMODE Off)
```

FIGURE 9-10: PORTE BLOCK DIAGRAM



11.0 TIMER0 MODULE

The Timer0 module has the following features:

- Software selectable as an 8-bit or 16-bit timer/counter
- Readable and writable
- Dedicated 8-bit software programmable prescaler
- Clock source selectable to be external or internal
- Interrupt-on-overflow from FFh to 00h in 8-bit mode and FFFFh to 0000h in 16-bit mode
- Edge select for external clock

Register 11-1 shows the Timer0 Control register (T0CON).

Figure 11-1 shows a simplified block diagram of the Timer0 module in 8-bit mode and Figure 11-2 shows a simplified block diagram of the Timer0 module in 16-bit mode.

The T0CON register is a readable and writable register that controls all the aspects of Timer0, including the prescale selection.

Note: Timer0 is enabled on POR.

REGISTER 11-1: T0CON: TIMER0 CONTROL REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

- bit 7 **TMR0ON:** Timer0 On/Off Control bit
 1 = Enables Timer0
 0 = Stops Timer0
- bit 6 **T08BIT:** Timer0 8-bit/16-bit Control bit
 1 = Timer0 is configured as an 8-bit timer/counter
 0 = Timer0 is configured as a 16-bit timer/counter
- bit 5 **T0CS:** Timer0 Clock Source Select bit
 1 = Transition on T0CKI pin
 0 = Internal instruction cycle clock (CLKO)
- bit 4 **T0SE:** Timer0 Source Edge Select bit
 1 = Increment on high-to-low transition on T0CKI pin
 0 = Increment on low-to-high transition on T0CKI pin
- bit 3 **PSA:** Timer0 Prescaler Assignment bit
 1 = Timer0 prescaler is not assigned. Timer0 clock input bypasses prescaler.
 0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.
- bit 2-0 **T0PS2:T0PS0:** Timer0 Prescaler Select bits
 111 = 1:256 Prescale value
 110 = 1:128 Prescale value
 101 = 1:64 Prescale value
 100 = 1:32 Prescale value
 011 = 1:16 Prescale value
 010 = 1:8 Prescale value
 001 = 1:4 Prescale value
 000 = 1:2 Prescale value

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

PIC18FXX8

NOTES:

13.0 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

Register 13-1 shows the Timer2 Control register. Timer2 can be shut-off by clearing control bit TMR2ON (T2CON register) to minimize power consumption. Figure 13-1 is a simplified block diagram of the Timer2 module. The prescaler and postscaler selection of Timer2 are controlled by this register.

13.1 Timer2 Operation

Timer2 can be used as the PWM time base for the PWM mode of the CCP 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 T2CKPS1:T2CKPS0 (T2CON register). 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, PIR registers).

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.

Note: Timer2 is disabled on POR.

REGISTER 13-1: T2CON: TIMER2 CONTROL REGISTER

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 **TOUTPS3:TOUTPS0:** 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 **T2CKPS1:T2CKPS0:** 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'

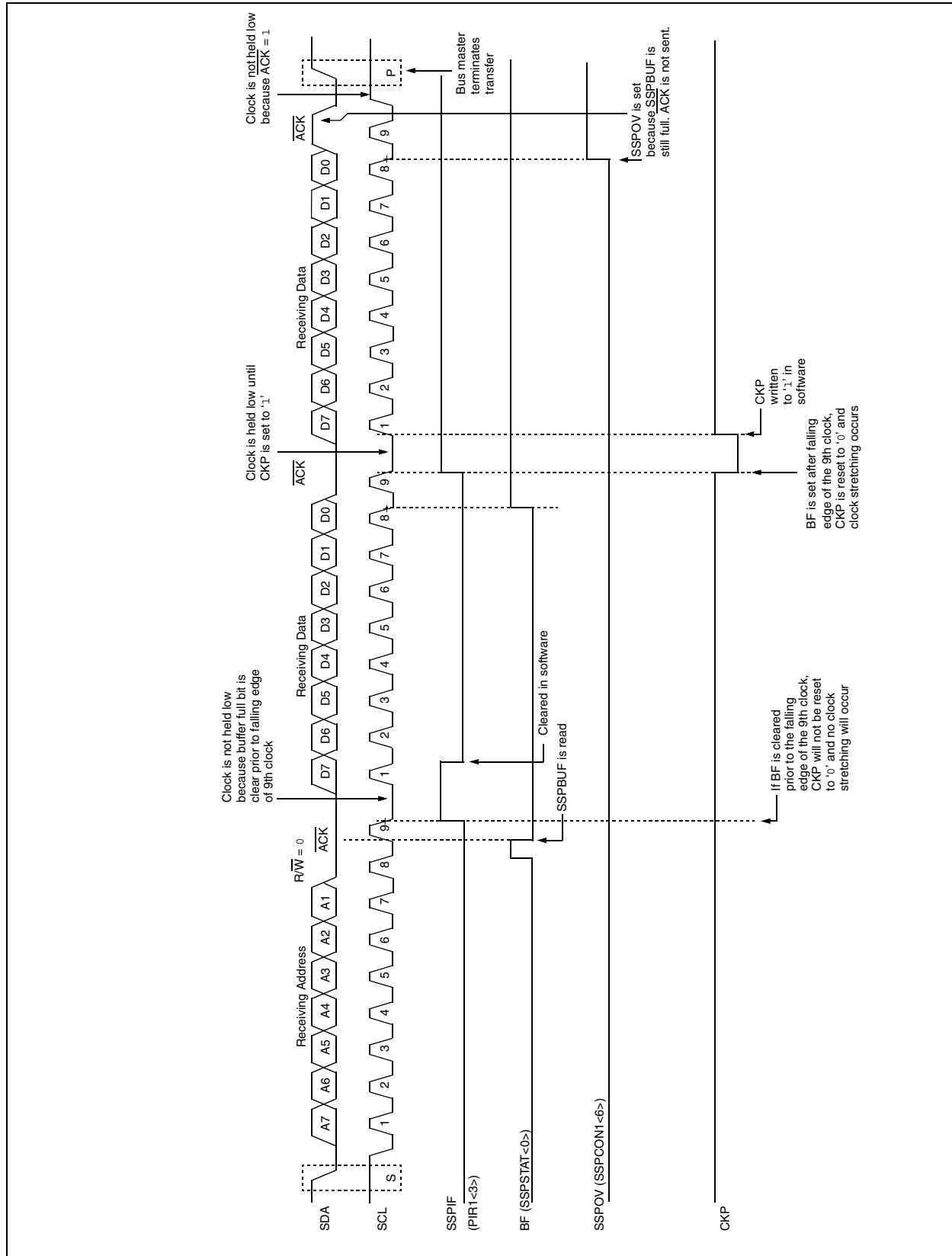
-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

FIGURE 17-13: I²C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 7-BIT ADDRESS)



17.4.8 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition, the user sets the Start condition enable bit, SEN (SSPCON2<0>). If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low, while SCL is high, is the Start condition and causes the S bit (SSPSTAT<3>) to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit (SSPCON2<0>) will be automatically cleared by hardware, the Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

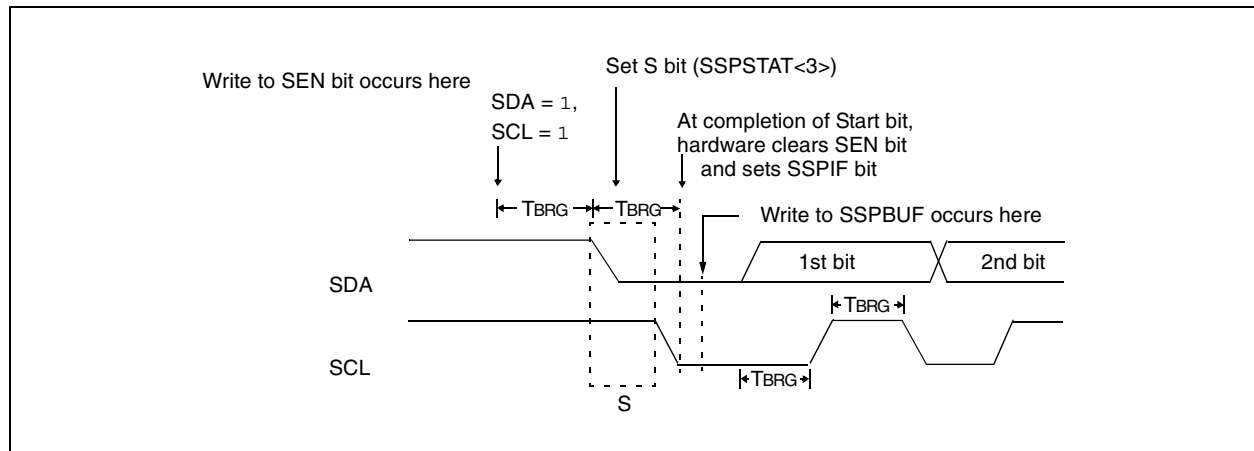
Note: If, at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs; the Bus Collision Interrupt Flag, BCLIF, is set, the Start condition is aborted and the I²C module is reset into its Idle state.

17.4.8.1 WCOL Status Flag

If the user writes the SSPBUF when a Start sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

Note: Because queueing of events is not allowed, writing to the lower 5 bits of SSPCON2 is disabled until the Start condition is complete.

FIGURE 17-19: FIRST START BIT TIMING



PIC18FXX8

NOTES:

PIC18FXX8

REGISTER 19-2: CANSTAT: CAN STATUS REGISTER

R-1	R-0	R-0	U-0	R-0	R-0	R-0	U-0
OPMODE2	OPMODE1	OPMODE0	—	ICODE2	ICODE1	ICODE0	—
bit 7							bit 0

bit 7-5 **OPMODE2:OPMODE0:** Operation Mode Status bits

111 = Reserved
110 = Reserved
101 = Reserved
100 = Configuration mode
011 = Listen Only mode
010 = Loopback mode
001 = Disable mode
000 = Normal mode

Note: Before the device goes into Sleep mode, select Disable mode.

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

bit 3-1 **ICODE2:ICODE0:** Interrupt Code bits

When an interrupt occurs, a prioritized coded interrupt value will be present in the ICODE2:ICODE0 bits. These codes indicate the source of the interrupt. The ICODE2:ICODE0 bits can be copied to the WIN2:WIN0 bits to select the correct buffer to map into the Access Bank area. See Example 19-1 for code example.

111 = Wake-up on interrupt
110 = RXB0 interrupt
101 = RXB1 interrupt
100 = TXB0 interrupt
011 = TXB1 interrupt
010 = TXB2 interrupt
001 = Error interrupt
000 = No interrupt

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

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

PIC18FXX8

NOTES:

CLRF		Clear f						
Syntax:	[<i>label</i>] CLRF f [,a]							
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]							
Operation:	000h → f 1 → Z							
Status Affected:	Z							
Encoding:	<table border="1"><tr><td>0110</td><td>101a</td><td>ffff</td><td>ffff</td></tr></table>				0110	101a	ffff	ffff
0110	101a	ffff	ffff					
Description:	Clears the contents of the specified register. If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).							
Words:	1							
Cycles:	1							
Q Cycle Activity:								
	Q1	Q2	Q3	Q4				
	Decode	Read register 'f'	Process Data	Write register 'f'				

Example: CLRF FLAG_REG

Before Instruction
FLAG_REG = 0x5A

After Instruction
FLAG_REG = 0x00

CLRWDT		Clear Watchdog Timer							
Syntax:	[<i>label</i>] CLRWDT								
Operands:	None								
Operation:	000h → WDT, 000h → WDT postscaler, 1 → \overline{TO} , 1 → \overline{PD}								
Status Affected:	\overline{TO} , \overline{PD}								
Encoding:	<table><tr><td>0000</td><td>0000</td><td>0000</td><td>0100</td></tr></table>				0000	0000	0000	0100	
0000	0000	0000	0100						
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the postscaler of the WDT. Status bits \overline{TO} and \overline{PD} are set.								
Words:	1								
Cycles:	1								
Q Cycle Activity:									
	Q1	Q2	Q3	Q4					
	Decode	No operation	Process Data	No operation					

Example: CLRWDT

Before Instruction
WDT Counter = ?

After Instruction
WDT Counter = 0x00
WDT Postscaler = 0
 \overline{TO} = 1
 \overline{PD} = 1

PIC18FXX8

GOTO Unconditional Branch

Syntax: [*label*] GOTO k

Operands: $0 \leq k \leq 1048575$

Operation: $k \rightarrow PC<20:1>$

Status Affected: None

Encoding:

1110	1111	k_7 kkk	kkkk k_0
1111	k_{19} kkk	kkkk	kkkk k_8

Description: GOTO allows an unconditional branch anywhere within entire 2-Mbyte memory range. The 20-bit value 'k' is loaded into PC<20:1>. GOTO is always a two-cycle instruction.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>	No operation	Read literal 'k'<19:8>, Write to PC
No operation	No operation	No operation	No operation

Example: GOTO THERE

After Instruction

PC = Address (THERE)

INCF Increment f

Syntax: [*label*] INCF f [,d [,a]]

Operands: $0 \leq f \leq 255$

$d \in [0,1]$

$a \in [0,1]$

Operation: $(f) + 1 \rightarrow \text{dest}$

Status Affected: C, DC, N, OV, Z

Encoding:

0010	10da	ffff	ffff
------	------	------	------

Description: The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: INCF CNT,

Before Instruction

CNT = 0xFF
Z = 0
C = ?
DC = ?

After Instruction

CNT = 0x00
Z = 1
C = 1
DC = 1

PIC18FXX8

FIGURE 27-12: PARALLEL SLAVE PORT TIMING (PIC18F248 AND PIC18F458)

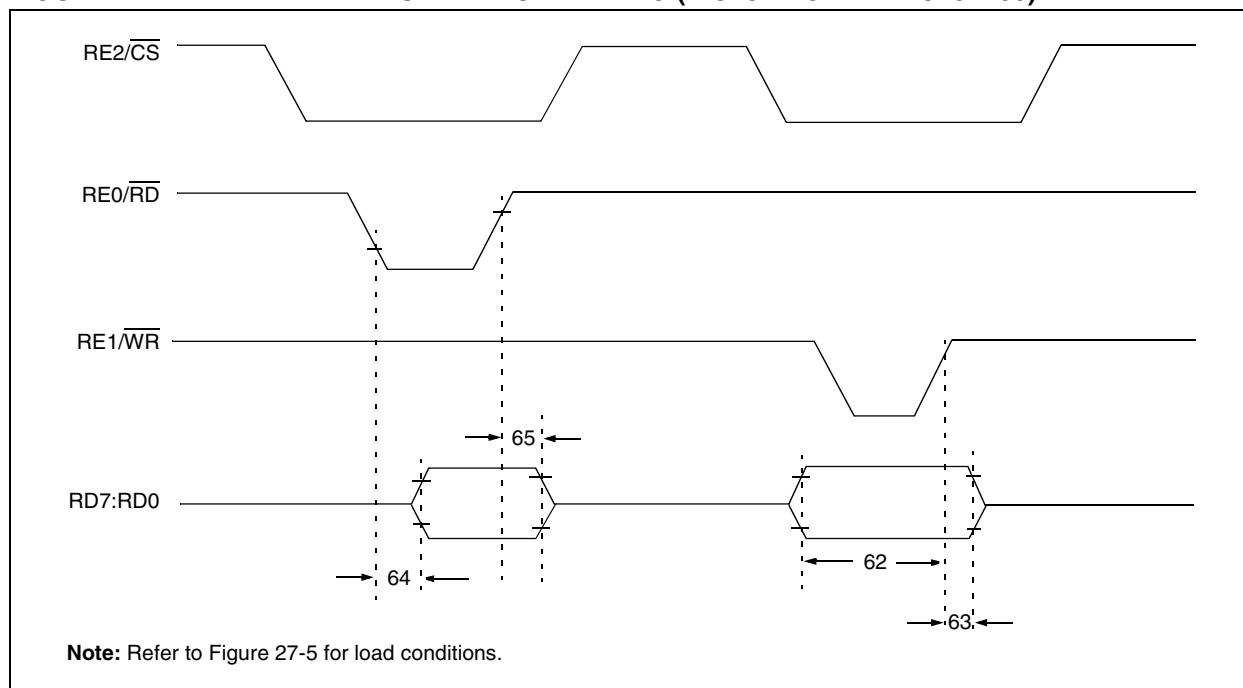


TABLE 27-12: PARALLEL SLAVE PORT REQUIREMENTS (PIC18F248 AND PIC18F458)

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
62	TdtV2wrH	Data-In Valid before $\overline{WR} \uparrow$ or $\overline{CS} \uparrow$ (setup time)	20 25	— —	ns ns	Extended Temp. range
63	TwrH2dtI	$\overline{WR} \uparrow$ or $\overline{CS} \uparrow$ to Data-In Invalid (hold time)	PIC18FXX8 20	— —	ns	
			PIC18LFX8 35	—	ns	
64	TrdL2dtV	$\overline{RD} \downarrow$ and $\overline{CS} \downarrow$ to Data-Out Valid	—	80	ns	Extended Temp. range
			—	90	ns	
65	TrdH2dtI	$\overline{RD} \uparrow$ or $\overline{CS} \downarrow$ to Data-Out Invalid	10	30	ns	
66	TibfINH	Inhibit the IBF flag bit being cleared from $\overline{WR} \uparrow$ or $\overline{CS} \uparrow$	—	3 Tcy	ns	

FIGURE 27-13: EXAMPLE SPI™ MASTER MODE TIMING (CKE = 0)

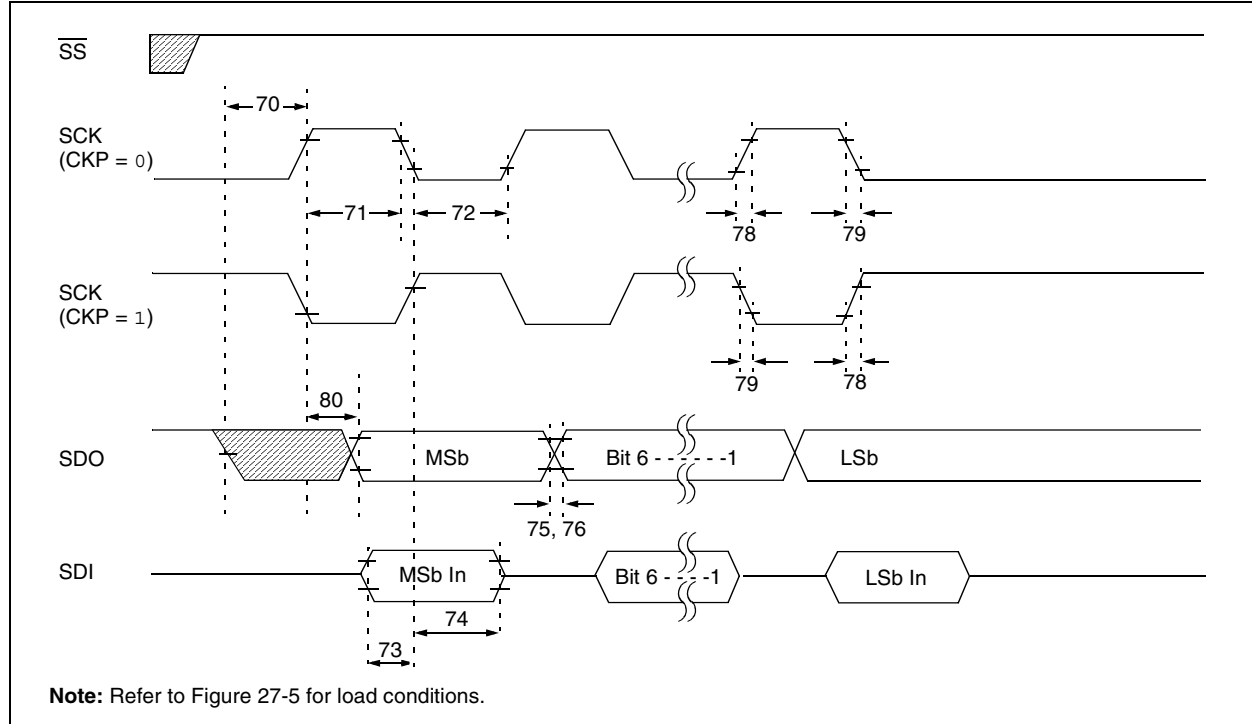


TABLE 27-13: EXAMPLE SPI™ MODE REQUIREMENTS (MASTER MODE, CKE = 0)

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
70	TssL2scH, TssL2scL	SS ↓ to SCK ↓ or SCK ↑ Input		T _{CY}	—	ns	
71	TscH	SCK Input High Time (Slave mode)	Continuous	1.25 T _{CY} + 30	—	ns	
71A			Single Byte	40	—	ns	(Note 1)
72	TscL	SCK Input Low Time (Slave mode)	Continuous	1.25 T _{CY} + 30	—	ns	
72A			Single Byte	40	—	ns	(Note 1)
73	TdiV2scH, TdiV2scL	Setup Time of SDI Data Input to SCK Edge		100	—	ns	
73A	Tb2B	Last Clock Edge of Byte 1 to the 1st Clock Edge of Byte 2		1.5 T _{CY} + 40	—	ns	(Note 2)
74	TscH2diL, TscL2diL	Hold Time of SDI Data Input to SCK Edge		100	—	ns	
75	TdoR	SDO Data Output Rise Time	PIC18FXX8	—	25	ns	
			PIC18LFX8	—	45	ns	
76	TdoF	SDO Data Output Fall Time		—	25	ns	
78	TscR	SCK Output Rise Time (Master mode)	PIC18FXX8	—	25	ns	
			PIC18LFX8	—	45	ns	
79	TscF	SCK Output Fall Time (Master mode)		—	25	ns	
80	TscH2doV, TscL2doV	SDO Data Output Valid after SCK Edge	PIC18FXX8	—	50	ns	
			PIC18LFX8	—	100	ns	

Note 1: Requires the use of parameter #73A.

Note 2: Only if parameter #71A and #72A are used.

PIC18FXX8

FIGURE 28-15: I_{PD} vs. V_{DD} , -40°C TO +125°C (SLEEP MODE, ALL PERIPHERALS DISABLED)

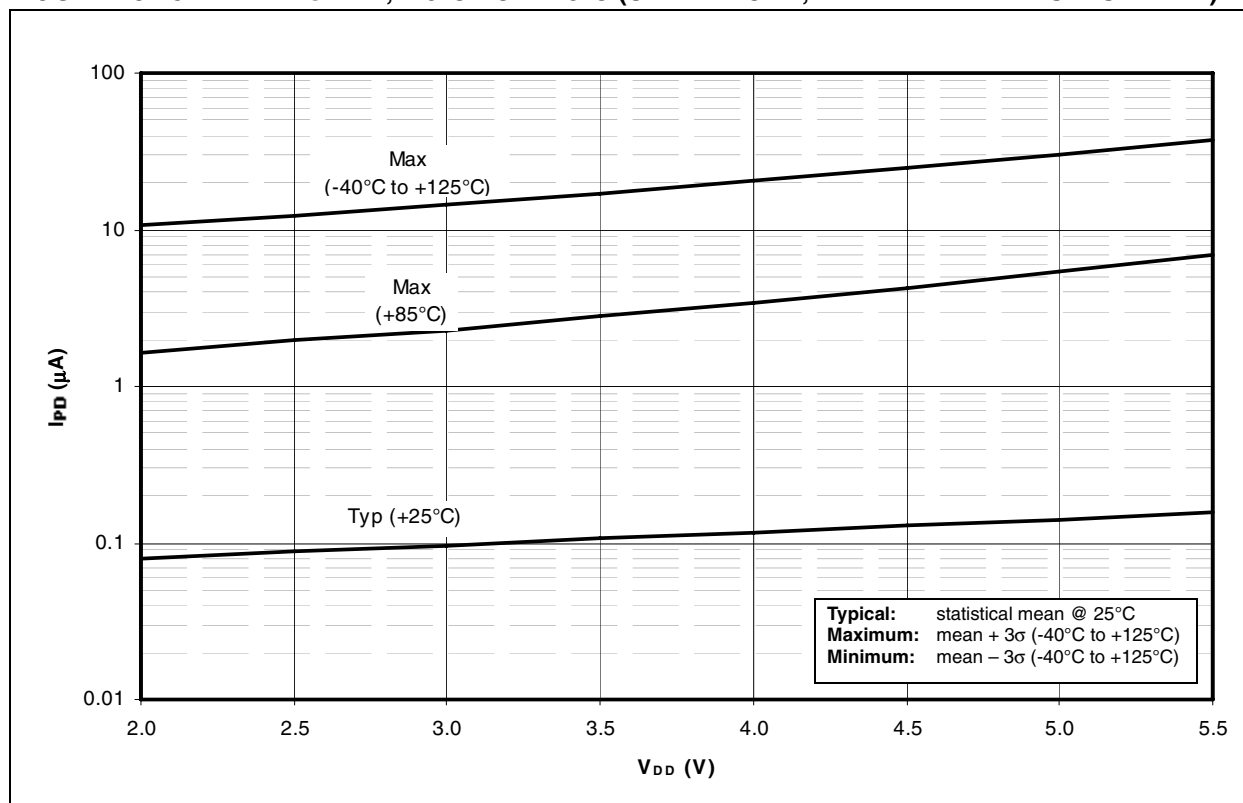


FIGURE 28-16: ΔI_{BOR} vs. V_{DD} OVER TEMPERATURE (BOR ENABLED, $V_{BOR} = 2.00-2.16V$)

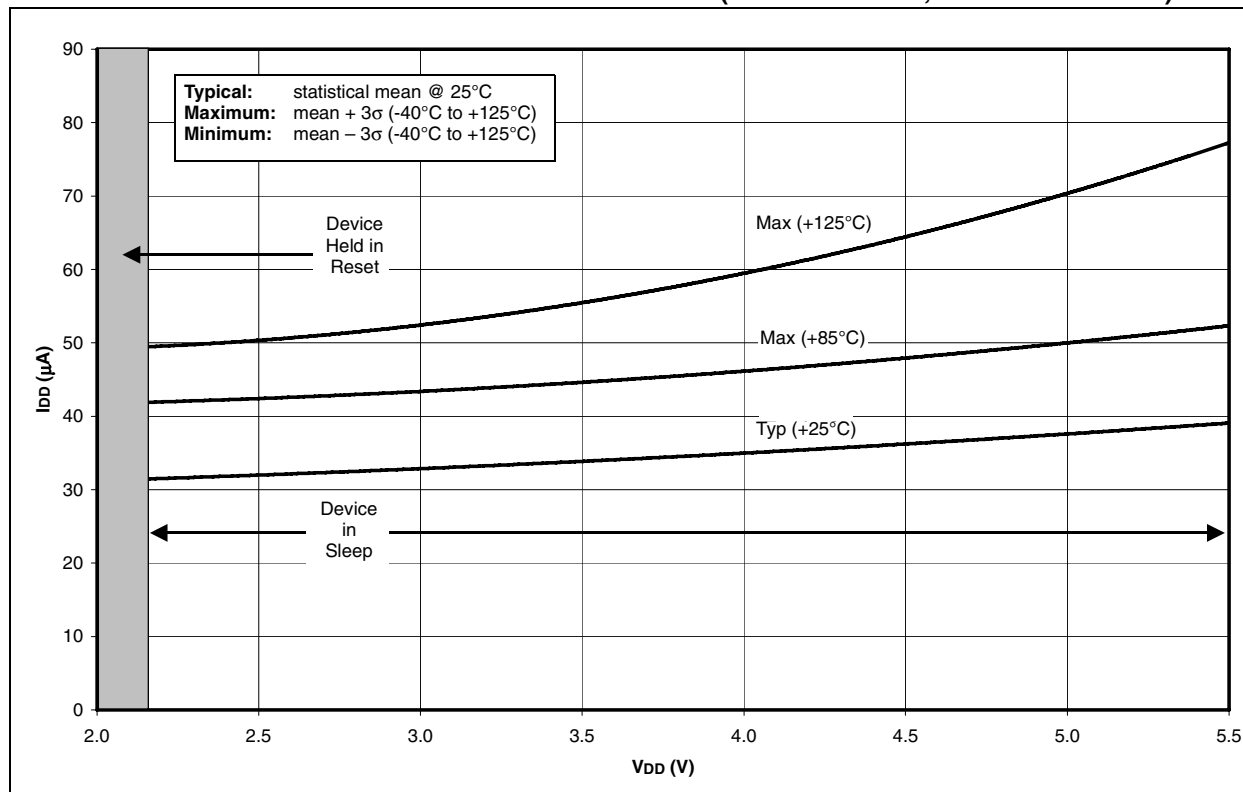


FIGURE 28-25: MINIMUM AND MAXIMUM V_{IN} vs. V_{DD} (ST INPUT, -40°C TO $+125^{\circ}\text{C}$)

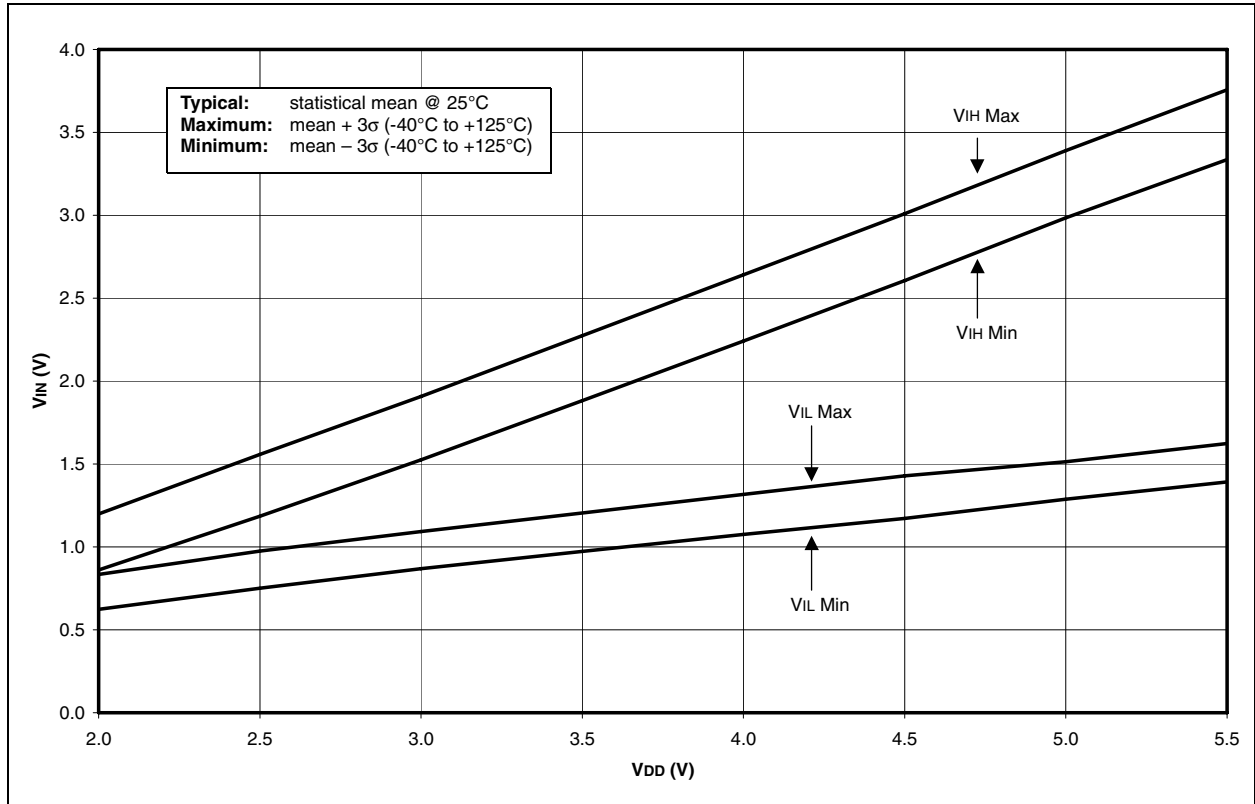
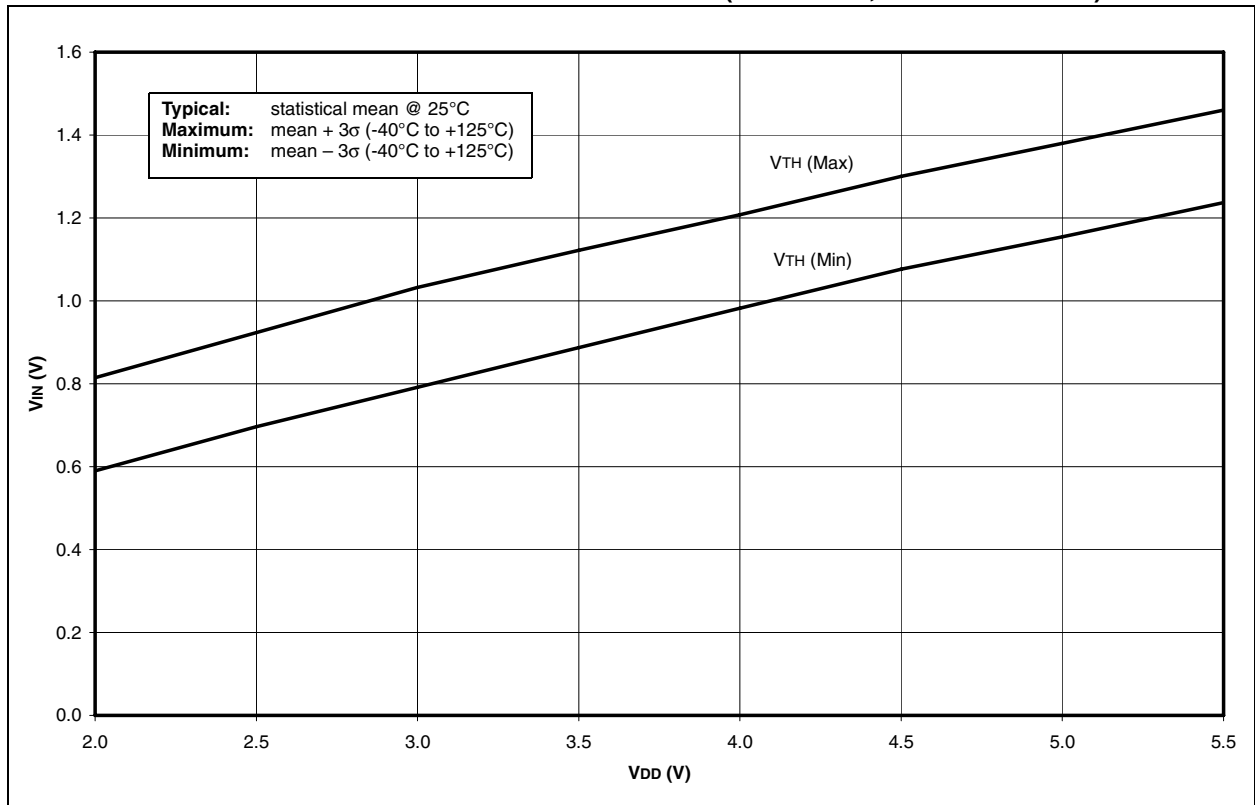


FIGURE 28-26: MINIMUM AND MAXIMUM V_{IN} vs. V_{DD} (TTL INPUT, -40°C TO $+125^{\circ}\text{C}$)



PIC18FXX8

BSF	293	Listen Only Mode	226
BTFSF	294	Loopback Mode	227
BTFSF	294	Message Acceptance Filters and Masks	215, 232
BTG	295	Message Acceptance Mask and Filter Operation (diagram)	232
BZ	296	Message Reception	230
C		Message Time-Stamping	230
C Compilers		Message Transmission	227
MPLAB C17	324	Modes of Operation	226
MPLAB C18	324	Normal Mode	226
MPLAB C30	324	Oscillator Tolerance	236
CALL	296	Overview	199
CAN Module		Phase Buffer Segments	234
Aborting Transmission	228	Programming Time Segments	236
Acknowledge Error	237	Propagation Segment	234
Baud Rate Registers	218	Receive Buffer Registers	210
Baud Rate Setting	233	Receive Buffers	230
Bit Error	237	Receive Message Buffering	230
Bit Time Partitioning (diagram)	233	Receive Priority	230
Bit Timing Configuration Registers	236	Registers	201
BRGCON1	236	Resynchronization	235
BRGCON2	236	Sample Point	234
BRGCON3	236	Shortening a Bit Period (diagram)	236
Calculating T _Q , Nominal Bit Rate and Nominal Bit Time	234	Stuff Bit Error	237
Configuration Mode	226	Synchronization	235
Control and Status Registers	201	Synchronization Rules	235
Controller Register Map	225	Synchronization Segment	234
CRC Error	237	Time Quanta	234
Disable Mode	226	Transmit Buffer Registers	206
Error Detection	237	Transmit Buffers	227
Error Modes and Error Counters	237	Transmit Priority	227
Error Modes State (diagram)	238	Transmit/Receive Buffers	199
Error Recognition Mode	227	Values for ICODE (table)	239
Error States	237	Capture (CCP Module)	124
Filter/Mask Truth (table)	232	CAN Message Time-Stamp	125
Form Error	237	CCP Pin Configuration	124
Hard Synchronization	235	CCP1 Prescaler	125
I/O Control Register	221	CCPR1H:CCPR1L Registers	124
Information Processing Time	234	Software Interrupt	125
Initiating Transmission	228	Timer1/Timer3 Mode Selection	124
Internal Message Reception Flowchart	231	Capture (ECCP Module)	133
Internal Transmit Message Flowchart	229	CAN Message Time-Stamp	133
Interrupt Acknowledge	239	Capture/Compare/PWM (CCP)	123
Interrupt Registers	222	Capture Mode. <i>See</i> Capture (CCP Module).	
Interrupts	238	CCP1 Module	124
Bus Activity Wake-up	239	Timer Resources	124
Bus-Off	239	CCPR1H Register	124
Code Bits	238	CCPR1L Register	124
Error	239	Compare Mode. <i>See</i> Compare (CCP Module).	
Message Error	239	Interaction of CCP1 and ECCP1 Modules	124
Receive	238	PWM Mode. <i>See</i> PWM (CCP Module).	
Receiver Bus Passive	239	Ceramic Resonators	
Receiver Overflow	239	Ranges Tested	17
Receiver Warning	239	Clocking Scheme	41
Transmit	238	CLRF	297
Transmitter Bus Passive	239	CLRWDT	297
Transmitter Warning	239		
Lengthening a Bit Period (diagram)	235		

PIC18FXX8

Enhanced CCP Auto-Shutdown	142
Enhanced PWM Mode. See PWM (ECCP Module).	
Errata	5
Error Recognition Mode (CAN Module)	226
Evaluation and Programming Tools	327
External Clock Input	19

F

Firmware Instructions	281
Flash Program Memory	65
Associated Registers	74
Control Registers	66
Erase Sequence	70
Erasing	70
Operation During Code-Protect	73
Reading	69
TABLAT (Table Latch) Register	68
Table Pointer Boundaries Based on Operation	68
Table Pointer Boundaries	68
Table Reads and Table Writes	65
TBLPTR (Table Pointer) Register	68
Write Sequence	71
Writing to	71
Protection Against Spurious Writes	73
Unexpected Termination	73
Write Verify	73

G

GOTO	302
------------	-----

H

Hardware Multiplier	75
Operation	75
Performance Comparison (table)	75
HS4 (PLL)	19

I

I/O Ports	93
I ² C Mode	152
ACK Pulse	156, 157
Acknowledge Sequence Timing	176
Baud Rate Generator	169
Bus Collision During a Repeated Start Condition	180
Bus Collision During a Start Condition	178
Bus Collision During a Stop Condition	181
Clock Arbitration	170
Clock Stretching	162
Effect of a Reset	177
General Call Address Support	166
Master Mode	167
Operation	168
Reception	173
Repeated Start Condition Timing	172
Start Condition Timing	171
Transmission	173
Multi-Master Mode	177
Communication, Bus Collision and Bus Arbitration	177
Operation	156
Read/Write Bit Information (R/W Bit)	156, 157
Registers	152
Serial Clock (RC3/SCK/SCL)	157

Slave Mode	156
Addressing	156
Reception	157
Transmission	157
Sleep Operation	177
Stop Condition Timing	176
ID Locations	265, 279
INCF	302
INCFSZ	303
In-Circuit Debugger	279
In-Circuit Serial Programming (ICSP)	265, 279
Indirect Addressing	56
FSR Register	55
INDF Register	55
Operation	55
INFSNZ	303
Initialization Conditions for All Registers	30
Instruction Cycle	41
Instruction Flow/Pipelining	41
Instruction Format	283
Instruction Set	281
ADDLW	287
ADDWF	287
ADDWFC	288
ANDLW	288
ANDWF	289
BC	289
BCF	290
BN	290
BNC	291
BNN	291
BNOV	292
BNZ	292
BOV	295
BRA	293
BSF	293
BTFSC	294
BTFSS	294
BTG	295
BZ	296
CALL	296
CLRF	297
CLRWDT	297
COMF	298
CPFSEQ	298
CPFSGT	299
CPFSLT	299
DAW	300
DCFSNZ	301
DECF	300
DECFSZ	301
GOTO	302
INCF	302
INCFSZ	303
INFSNZ	303
IORLW	304
IORWF	304
LFSR	305
MOVF	305
MOVFF	306
MOVLB	306
MOVLW	307
MOVWF	307
MULLW	308
MULWF	308