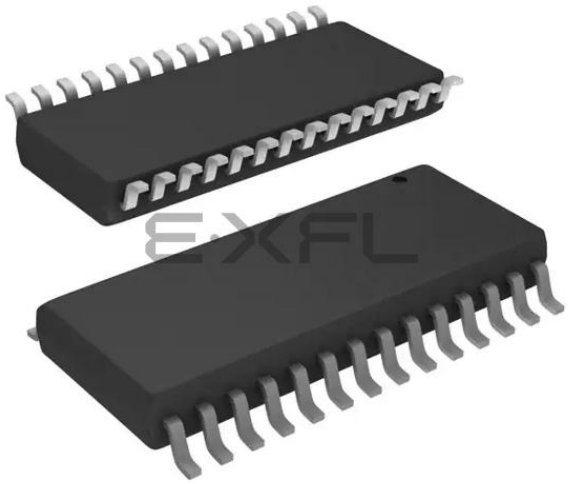


Welcome to E-XFL.COM

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.

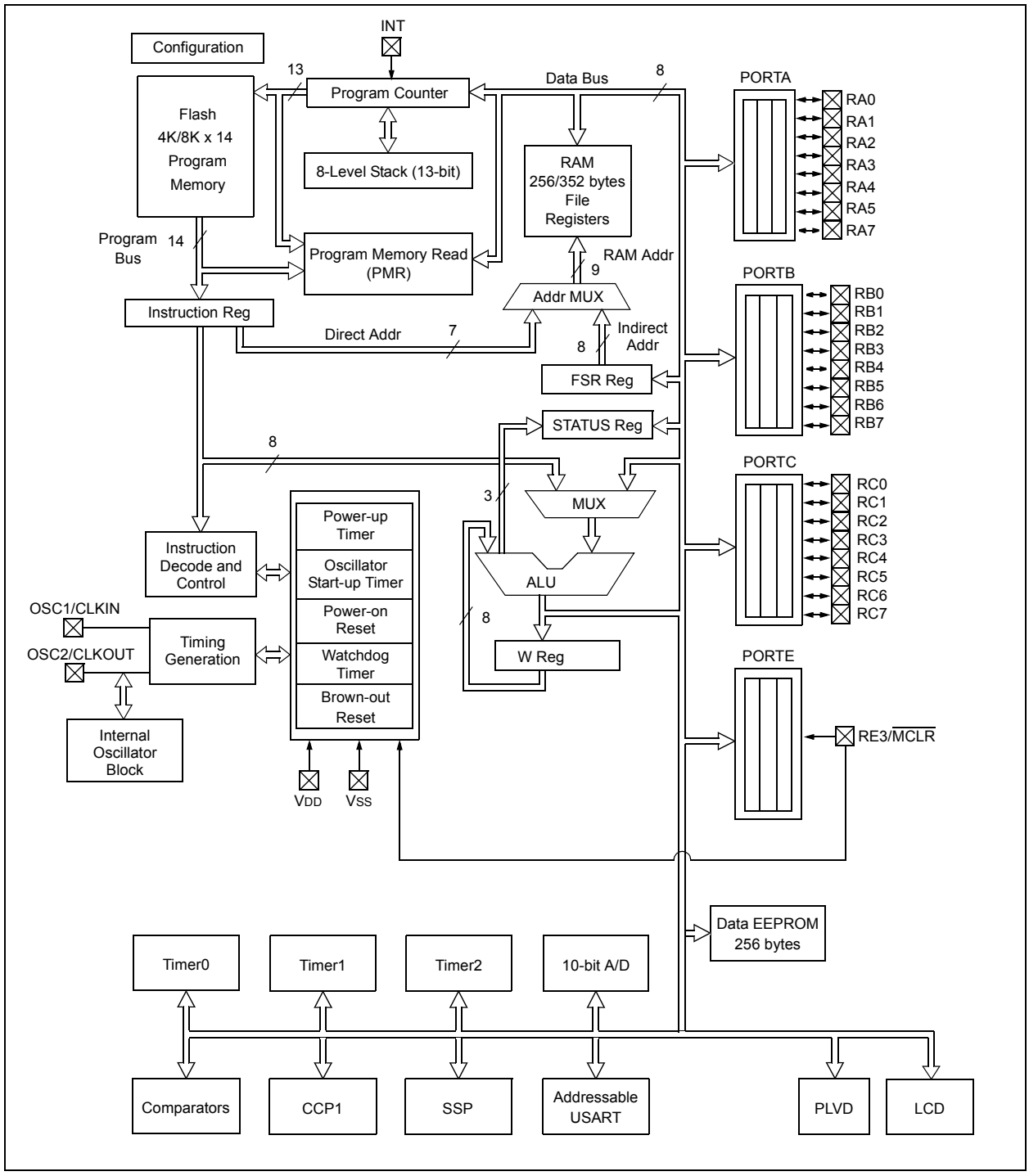
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	352 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f916-e-so

1.0 DEVICE OVERVIEW

The PIC16F91X/946 devices are covered by this data sheet. They are available in 28/40/44/64-pin packages. Figure 1-1 shows a block diagram of the PIC16F913/916 device, Figure 1-2 shows a block diagram of the PIC16F914/917 device, and Figure 1-3 shows a block diagram of the PIC16F946 device. Table 1-1 shows the pinout descriptions.

FIGURE 1-1: PIC16F913/916 BLOCK DIAGRAM



REGISTER 3-7: WPUB: WEAK PULL-UP 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
WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0
bit 7							bit 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

bit 7-0 **WPUB<7:0>:** Weak Pull-up Register bits
1 = Pull-up enabled
0 = Pull-up disabled

Note 1: Global $\overline{\text{RBPU}}$ must be enabled for individual pull-ups to be enabled.
2: The weak pull-up device is automatically disabled if the pin is in Output mode (TRISx<7:0> = 0).

3.4.3 PIN DESCRIPTIONS AND DIAGRAMS

Each PORTB pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the LCD or interrupts, refer to the appropriate section in this data sheet.

3.4.3.1 RB0/INT/SEG0

Figure 3-9 shows the diagram for this pin. The RB0 pin is configurable to function as one of the following:

- a general purpose I/O
- an external edge triggered interrupt
- an analog output for the LCD

3.4.3.2 RB1/SEG1

Figure 3-9 shows the diagram for this pin. The RB1 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog output for the LCD

3.4.3.3 RB2/SEG2

Figure 3-9 shows the diagram for this pin. The RB2 pin is configurable to function as one of the following:

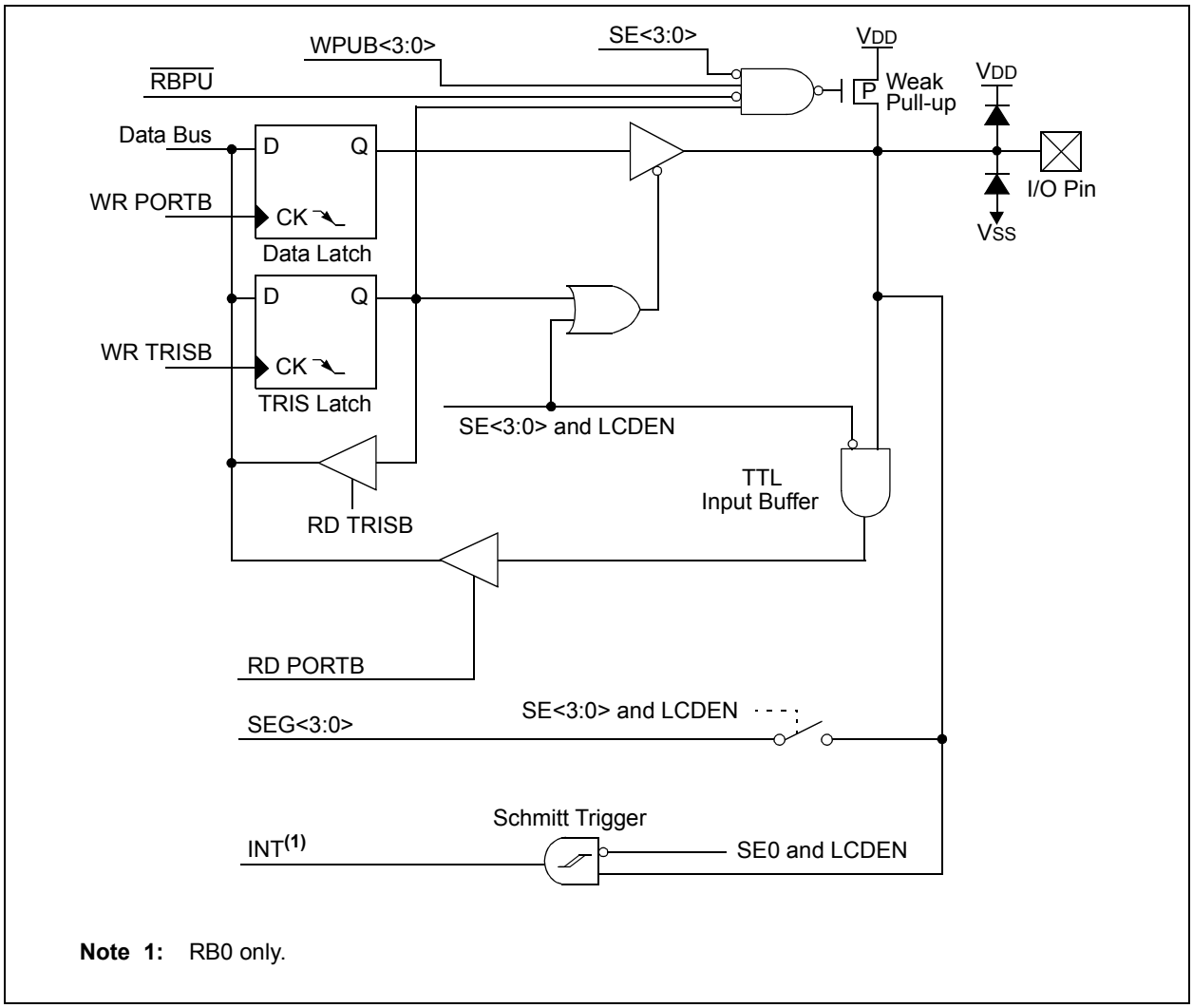
- a general purpose I/O
- an analog output for the LCD

3.4.3.4 RB3/SEG3

Figure 3-9 shows the diagram for this pin. The RB3 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog output for the LCD

FIGURE 3-9: BLOCK DIAGRAM OF RB<3:0>



Note 1: RB0 only.

3.8 PORTF and TRISF Registers

PORTF is an 8-bit port with Schmitt Trigger input buffers. RF<7:0> are individually configured as inputs or outputs, depending on the state of the port direction. The port bits are also multiplexed with LCD segment functions. PORTF is available on the PIC16F946 only.

EXAMPLE 3-6: INITIALIZING PORTF

```
BANKSEL PORTF       ;
CLRF    PORTF       ;Init PORTF
BANKSEL TRISF       ;
MOVLW   0FFh        ;Set RF<7:0> as inputs
MOVWF   TRISF       ;
```

REGISTER 3-14: PORTF: PORTF REGISTER(1)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0
bit 7							bit 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

bit 7-0 **RF<7:0>**: PORTF I/O Pin bits
1 = Port pin is >VIH min.
0 = Port pin is <VIL max.

Note 1: PIC16F946 only.

REGISTER 3-15: TRISF: PORTF TRI-STATE REGISTER(1)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0
bit 7							bit 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

bit 7-0 **TRISF<7:0>:** PORTF Tri-State Control bits
1 = PORTF pin configured as an input (tri-stated)
0 = PORTF pin configured as an output

Note 1: PIC16F946 only.

7.0 TIMER2 MODULE

The Timer2 module is an 8-bit timer with the following features:

- 8-bit timer register (TMR2)
- 8-bit period register (PR2)
- Interrupt on TMR2 match with PR2
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)

See Figure 7-1 for a block diagram of Timer2.

7.1 Timer2 Operation

The clock input to the Timer2 module is the system instruction clock (FOSC/4). The clock is fed into the Timer2 prescaler, which has prescale options of 1:1, 1:4 or 1:16. The output of the prescaler is then used to increment the TMR2 register.

The values of TMR2 and PR2 are constantly compared to determine when they match. TMR2 will increment from 00h until it matches the value in PR2. When a match occurs, two things happen:

- TMR2 is reset to 00h on the next increment cycle.
- The Timer2 postscaler is incremented.

The match output of the Timer2/PR2 comparator is then fed into the Timer2 postscaler. The postscaler has postscale options of 1:1 to 1:16 inclusive. The output of the Timer2 postscaler is used to set the TMR2IF interrupt flag bit in the PIR1 register.

The TMR2 and PR2 registers are both fully readable and writable. On any Reset, the TMR2 register is set to 00h and the PR2 register is set to FFh.

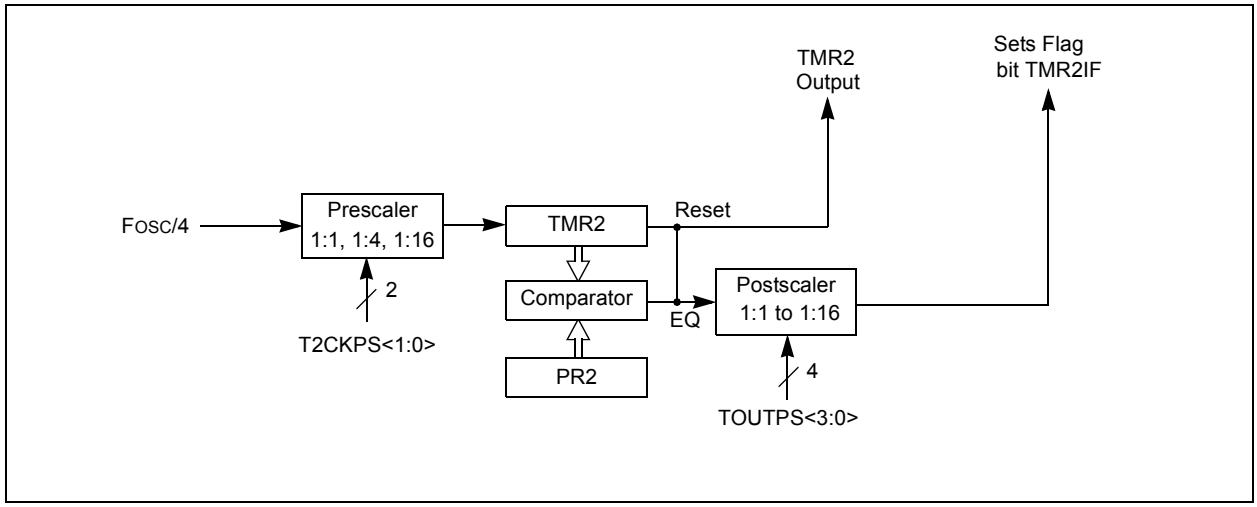
Timer2 is turned on by setting the TMR2ON bit in the T2CON register to a '1'. Timer2 is turned off by clearing the TMR2ON bit to a '0'.

The Timer2 prescaler is controlled by the T2CKPS bits in the T2CON register. The Timer2 postscaler is controlled by the TOUTPS bits in the T2CON register. The prescaler and postscaler counters are cleared when:

- A write to TMR2 occurs.
- A write to T2CON occurs.
- Any device Reset occurs (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset, or Brown-out Reset).

Note:	TMR2 is not cleared when T2CON is written.

FIGURE 7-1: TIMER2 BLOCK DIAGRAM



NOTES:

12.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

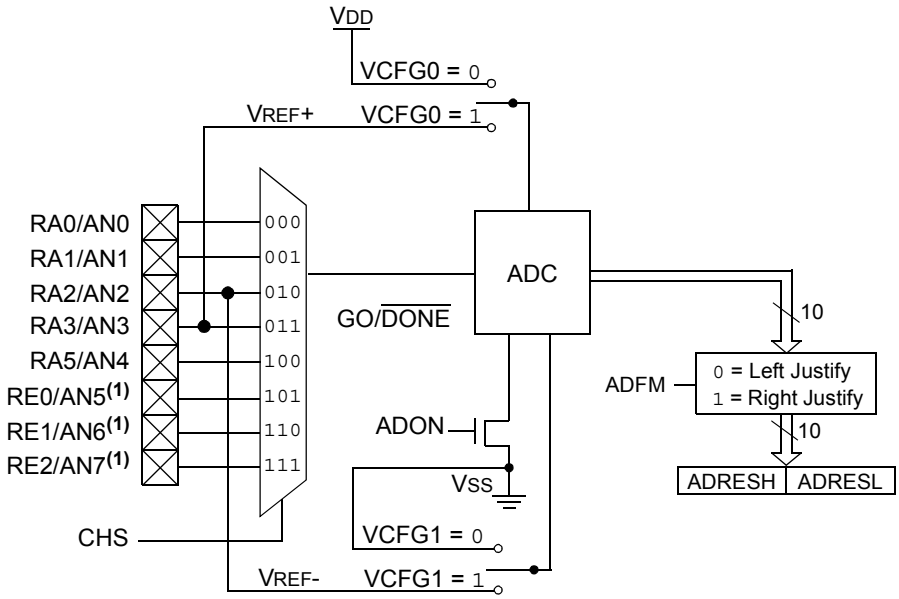
The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADRESL and ADRESH).

The ADC voltage reference is software selectable to be either internally generated or externally supplied.

The ADC can generate an interrupt upon completion of a conversion. This interrupt can be used to wake-up the device from Sleep.

Figure 12-1 shows the block diagram of the ADC.

FIGURE 12-1: ADC BLOCK DIAGRAM



Note 1: These channels are only available on PIC16F914/917 and PIC16F946 devices.

12.2.6 A/D CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

1. Configure Port:
   - Disable pin output driver (See TRIS register)
   - Configure pin as analog
2. Configure the ADC module:
   - Select ADC conversion clock
   - Configure voltage reference
   - Select ADC input channel
   - Select result format
   - Turn on ADC module
3. Configure ADC interrupt (optional):
   - Clear ADC interrupt flag
   - Enable ADC interrupt
   - Enable peripheral interrupt
   - Enable global interrupt(1)
4. Wait the required acquisition time(2).
5. Start conversion by setting the GO/$\overline{\text{DONE}}$ bit.
6. Wait for ADC conversion to complete by one of the following:
   - Polling the GO/$\overline{\text{DONE}}$ bit
   - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result
8. Clear the ADC interrupt flag (required if interrupt is enabled).

Note 1: The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.

2: See **Section 12.3 "A/D Acquisition Requirements"**.

EXAMPLE 12-1: A/D CONVERSION

```
;This code block configures the ADC
;for polling, Vdd reference, Frc clock
;and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL   ADCON1       ;
MOVLW     B'01110000'  ;ADC Frc clock
MOVWF     ADCON1       ;
BANKSEL   TRISA        ;
BSF       TRISA,0      ;Set RA0 to input
BANKSEL   ANSEL        ;
BSF       ANSEL,0      ;Set RA0 to analog
BANKSEL   ADCON0       ;
MOVLW     B'10000001'  ;Right justify,
MOVWF     ADCON0       ;Vdd Vref, AN0, On
CALL      SampleTime   ;Acquisiton delay
BSF       ADCON0,GO    ;Start conversion
BTFSC     ADCON0,GO    ;Is conversion done?
GOTO      $-1          ;No, test again
BANKSEL   ADRESH       ;
MOVF      ADRESH,W     ;Read upper 2 bits
MOVWF     RESULTHI     ;store in GPR space
BANKSEL   ADRESL       ;
MOVF      ADRESL,W     ;Read lower 8 bits
MOVWF     RESULTLO     ;Store in GPR space
```

12.2.7 ADC REGISTER DEFINITIONS

The following registers are used to control the operation of the ADC.

REGISTER 13-1: EEDATL: EEPROM/PROGRAM MEMORY DATA LOW BYTE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EEDATL7	EEDATL6	EEDATL5	EEDATL4	EEDATL3	EEDATL2	EEDATL1	EEDATL0
bit 7							bit 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

bit 7-0 **EEDATL<7:0>**: Byte value to Write to or Read from data EEPROM bits or to Read from program memory

REGISTER 13-2: EEADRL: EEPROM/PROGRAM MEMORY ADDRESS LOW BYTE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EEADRL7	EEADRL6	EEADRL5	EEADRL4	EEADRL3	EEADRL2	EEADRL1	EEADRL0
bit 7							bit 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

bit 7-0 **EEADRL<7:0>**: Specifies one of 256 locations for EEPROM Read/Write operation bits or low address byte for program memory reads

REGISTER 13-3: EEDATH: PROGRAM MEMORY DATA HIGH BYTE REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0
bit 7							bit 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

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **EEDATH<5:0>**: Byte value to Read from program memory

REGISTER 13-4: EEADRH: PROGRAM MEMORY ADDRESS HIGH BYTE REGISTER

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0
bit 7							bit 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

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **EEADRH<4:0>**: Specifies the high address byte for program memory reads

FIGURE 13-1: FLASH PROGRAM MEMORY READ CYCLE EXECUTION

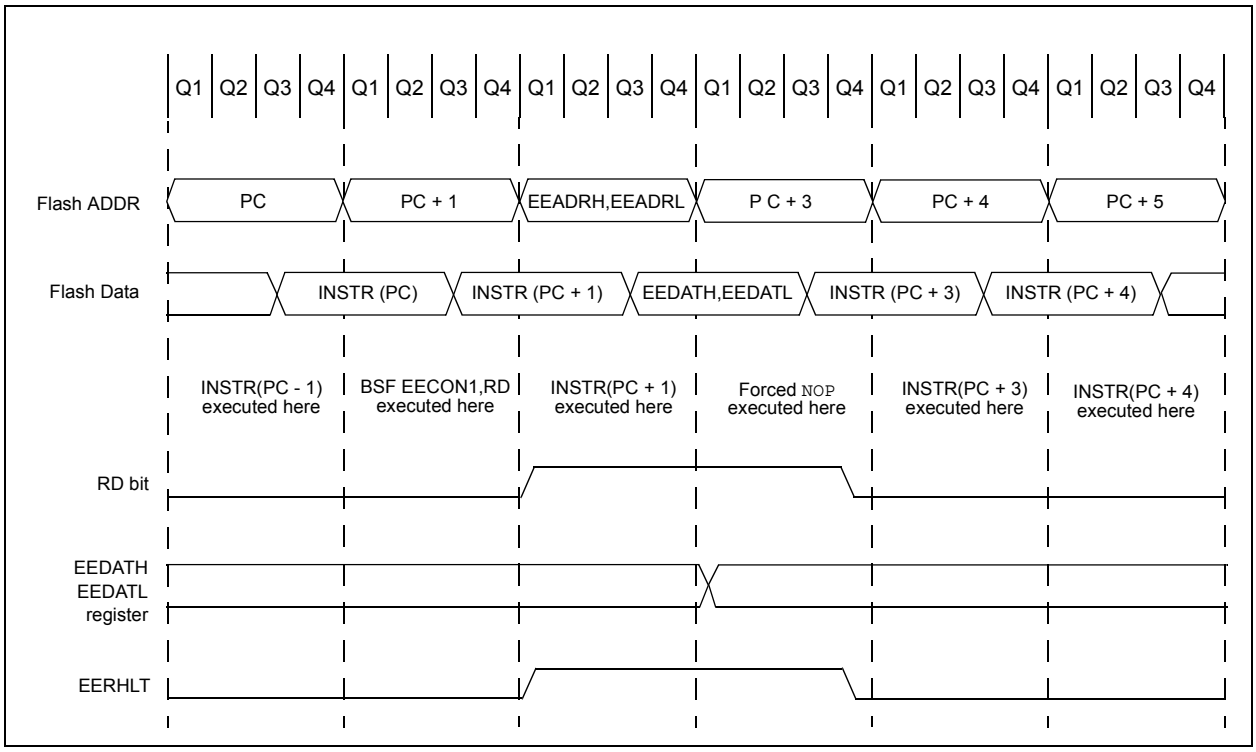


TABLE 13-1: SUMMARY OF ASSOCIATED REGISTERS WITH DATA EEPROM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
EEADRH	—	—	—	EEADRH4	EEADRH3	EEADRH2	EEADRH1	EEADRH0	---0 0000	---0 0000
EEADRL	EEADRL7	EEADRL6	EEADRL5	EEADRL4	EEADRL3	EEADRL2	EEADRL1	EEADRL0	0000 0000	0000 0000
EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	0--- x000	---- q000
EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	---- ----
EEDATH	—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0	--00 0000	--00 0000
EEDATL	EEDATL7	EEDATL6	EEDATL5	EEDATL4	EEDATL3	EEDATL2	EEDATL1	EEDATL0	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', q = value depends upon condition.
Shaded cells are not used by data EEPROM module.

NOTES:

15.3.1 PWM PERIOD

The PWM period is specified by the PR2 register of Timer2. The PWM period can be calculated using the formula of Equation 15-1.

EQUATION 15-1: PWM PERIOD

$$PWM\ Period = [(PR2) + 1] \bullet 4 \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

Note: TOSC = 1/FOSC

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from CCPRxL into CCPRxH.

Note: The Timer2 postscaler (see **Section 7.1 "Timer2 Operation"**) is not used in the determination of the PWM frequency.

15.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to multiple registers: CCPRxL register and CCPx<1:0> bits of the CCPxCON register. The CCPRxL contains the eight MSbs and the CCPx<1:0> bits of the CCPxCON register contain the two LSbs. CCPRxL and CCPx<1:0> bits of the CCPxCON register can be written to at any time. The duty cycle value is not latched into CCPRxH until after the period completes (i.e., a match between PR2 and TMR2 registers occurs). While using the PWM, the CCPRxH register is read-only.

Equation 15-2 is used to calculate the PWM pulse width.

Equation 15-3 is used to calculate the PWM duty cycle ratio.

EQUATION 15-2: PULSE WIDTH

$$Pulse\ Width = (CCPRxL:CCPxCON<5:4>) \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

EQUATION 15-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(CCPRxL:CCPxCON<5:4>)}{4(PR2 + 1)}$$

The CCPRxH register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

The 8-bit timer TMR2 register is concatenated with either the 2-bit internal system clock (FOSC), or 2 bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

When the 10-bit time base matches the CCPRxH and 2-bit latch, then the CCPx pin is cleared (see Figure 15-3).

16.9 In-Circuit Debugger

When the debug bit in the Configuration Word register is programmed to a '0', the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB® ICD 2. When the microcontroller has this feature enabled, some of the resources are not available for general use. See Table 16-9 for more detail.

Note: The user's application must have the circuitry required to support ICD functionality. Once the ICD circuitry is enabled, normal device pin functions on RB6/ICSPCLK/ICDCK/SEG14 and RB7/ICSPDAT/ICDDAT/SEG13 will not be usable. The ICD circuitry uses these pins for communication with the ICD2 external debugger.

For more information, see "*Using MPLAB® ICD 2*" (DS51265), available on Microchip's web site (www.microchip.com).

16.9.1 ICD PINOUT

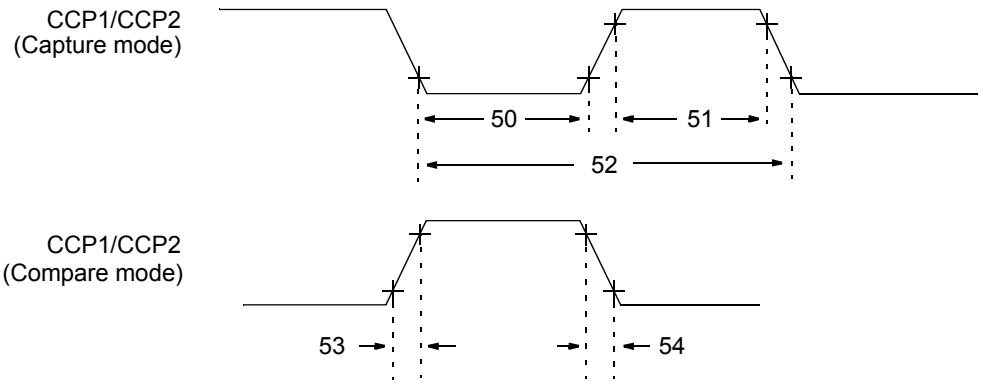
The devices in the PIC16F91X/946 family carry the circuitry for the In-Circuit Debugger on-chip and on existing device pins. This eliminates the need for a separate die or package for the ICD device. The pinout for the ICD device is the same as the devices (see **Section 1.0 "Device Overview"** for complete pinout and pin descriptions). Table 16-9 shows the location and function of the ICD related pins on the 28 and 40-pin devices.

TABLE 16-9: PIC16F91X/946-ICD PIN DESCRIPTIONS

Pin Numbers			Name	Type	Pull-up	Description
PDIP		TQFP				
PIC16F914/917	PIC16F913/916	PIC16F946				
40	28	24	ICDDATA	TTL	—	In Circuit Debugger Bidirectional data
39	27	23	ICDCLK	ST	—	In Circuit Debugger Bidirectional clock
1	1	36	$\overline{MCLR}$/VPP	HV	—	Programming voltage
11,32	20	10, 19, 38, 51	VDD	P	—	Power
12,31	8,19	9, 20, 41, 56	VSS	P	—	Ground
—	—	26	AVDD	P	—	Analog power
—	—	25	AVSS	P	—	Analog ground

Legend: TTL = TTL input buffer, ST = Schmitt Trigger input buffer, P = Power, HV = High Voltage

FIGURE 19-13: CAPTURE/COMPARE/PWM TIMINGS



Note: Refer to Figure 19-3 for load conditions.

TABLE 19-12: CAPTURE/COMPARE/PWM (CCP) REQUIREMENTS

Param. No.	Sym.	Characteristic			Min.	Typ†	Max.	Units	Conditions
50*	TccL	CCPx input low time	No Prescaler		0.5TCY + 5	—	—	ns	
			With Prescaler	3.0-5.5V	10	—	—	ns	
				2.0-5.5V	20	—	—	ns	
51*	TccH	CCPx input high time	No Prescaler		0.5TCY + 5	—	—	ns	
			With Prescaler	3.0-5.5V	10	—	—	ns	
				2.0-5.5V	20	—	—	ns	
52*	TccP	CCPx input period			$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1,4 or 16)
53*	TccR	CCPx output fall time		3.0-5.5V	—	10	25	ns	
				2.0-5.5V	—	25	50	ns	
54*	TccF	CCPx output fall time		3.0-5.5V	—	10	25	ns	
				2.0-5.5V	—	25	45	ns	

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

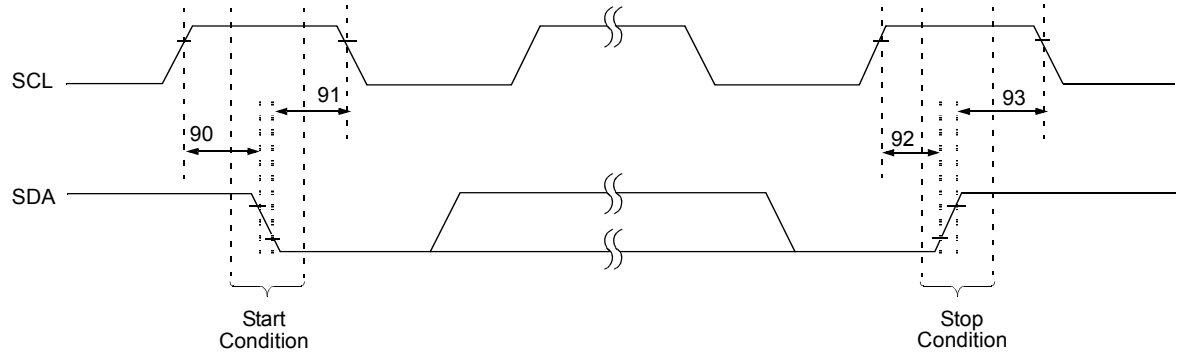
TABLE 19-14: SPI MODE REQUIREMENTS

Param No.	Symbol	Characteristic		Min.	Typ†	Max.	Units	Conditions
70*	TssL2scH, TssL2scL	$\overline{SS}\downarrow$ to SCK$\downarrow$ or SCK$\uparrow$ input		Tcy	—	—	ns	
71*	TscH	SCK input high time (Slave mode)		Tcy + 20	—	—	ns	
72*	TscL	SCK input low time (Slave mode)		Tcy + 20	—	—	ns	
73*	TdiV2scH, TdiV2scL	Setup time of SDI data input to SCK edge		100	—	—	ns	
74*	TscH2diL, TscL2diL	Hold time of SDI data input to SCK edge		100	—	—	ns	
75*	TdoR	SDO data output rise time	3.0-5.5V	—	10	25	ns	
			2.0-5.5V	—	25	50	ns	
76*	TdoF	SDO data output fall time		—	10	25	ns	
77*	TssH2doZ	$\overline{SS}\uparrow$ to SDO output high-impedance		10	—	50	ns	
78*	TscR	SCK output rise time (Master mode)	3.0-5.5V	—	10	25	ns	
			2.0-5.5V	—	25	50	ns	
79*	TscF	SCK output fall time (Master mode)		—	10	25	ns	
80*	TscH2doV, TscL2doV	SDO data output valid after SCK edge	3.0-5.5V	—	—	50	ns	
			2.0-5.5V	—	—	145	ns	
81*	TdoV2scH, TdoV2scL	SDO data output setup to SCK edge		Tcy	—	—	ns	
82*	TssL2doV	SDO data output valid after $\overline{SS}\downarrow$ edge		—	—	50	ns	
83*	TscH2ssH, TscL2ssH	$\overline{SS}\uparrow$ after SCK edge		1.5Tcy + 40	—	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 19-18: I^2C™ BUS START/STOP BITS TIMING



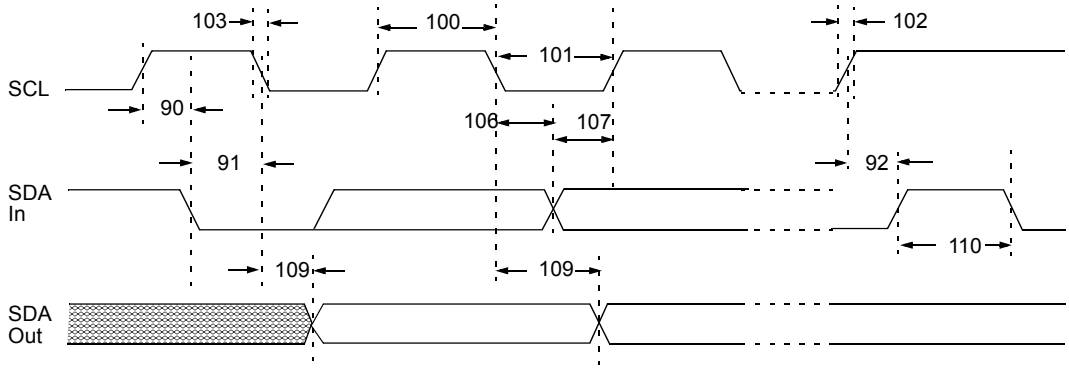
Note: Refer to Figure 19-3 for load conditions.

TABLE 19-15: I²C™ BUS START/STOP BITS REQUIREMENTS

Param No.	Symbol	Characteristic		Min.	Typ.	Max.	Units	Conditions
90*	TSU:STA	Start condition Setup time	400 kHz mode	600	—	—	ns	Only relevant for Repeated Start condition
91*	THD:STA	Start condition Hold time	400 kHz mode	600	—	—	ns	After this period, the first clock pulse is generated
92*	TSU:STO	Stop condition Setup time	400 kHz mode	600	—	—	ns	
93	THD:STO	Stop condition Hold time	400 kHz mode	600	—	—	ns	

* These parameters are characterized but not tested.

FIGURE 19-19: I²C™ BUS DATA TIMING



Note: Refer to Figure 19-3 for load conditions.

FIGURE 20-12: MAXIMUM IDD vs. FOSC OVER VDD (HFINTOSC MODE)

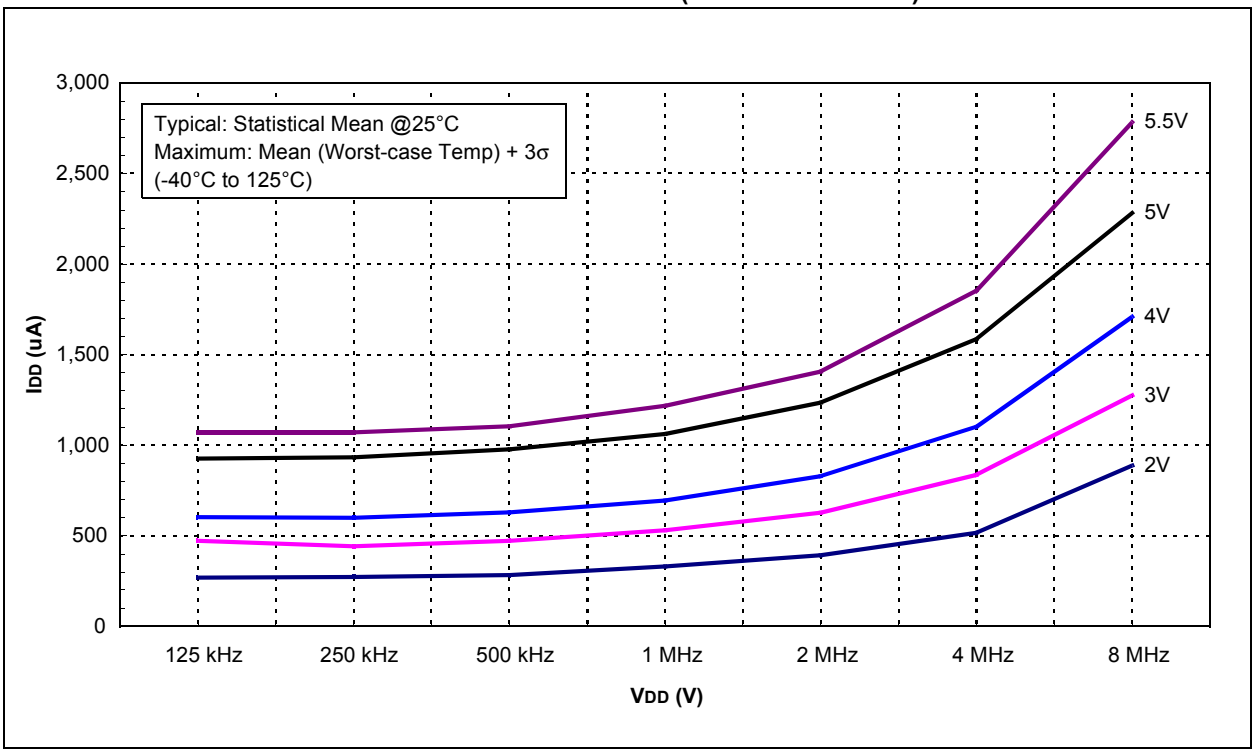


FIGURE 20-13: TYPICAL IPD vs. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)

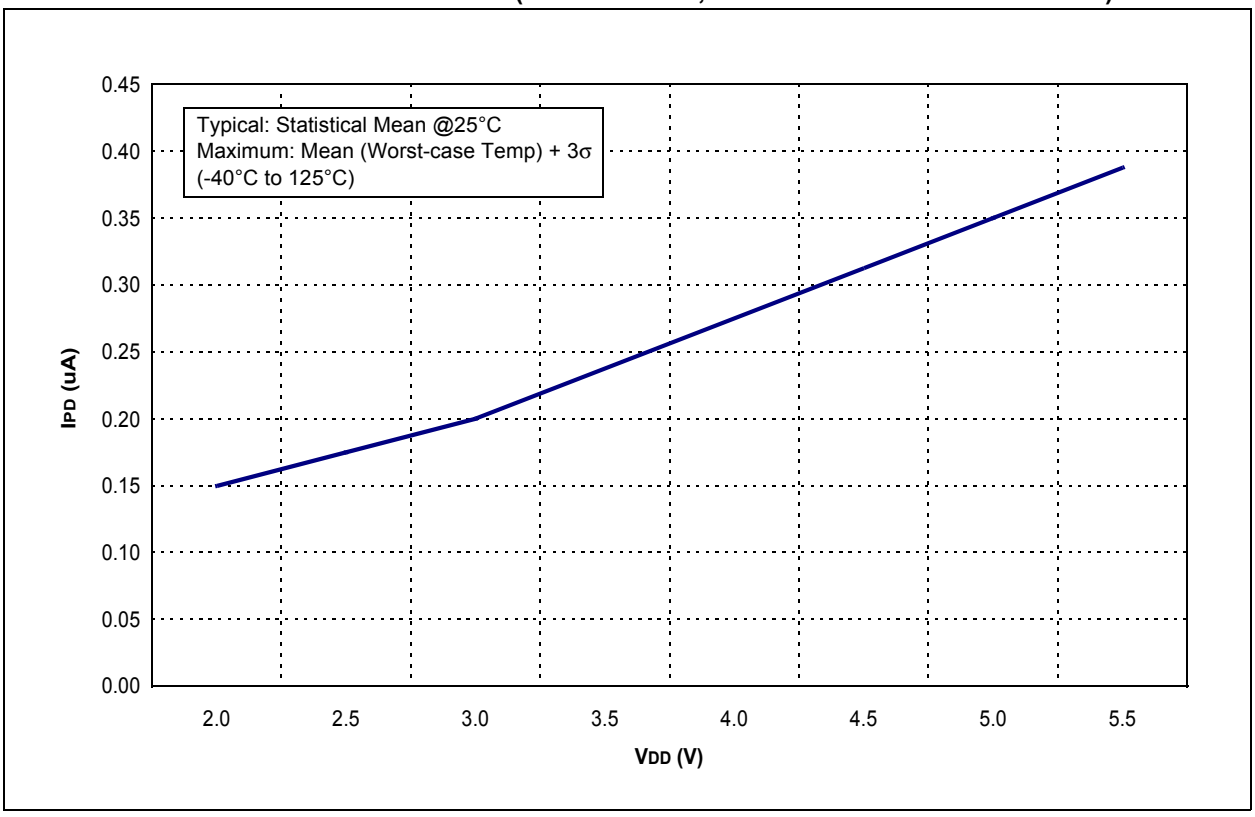


FIGURE 20-18: MAXIMUM WDT IPD vs. VDD OVER TEMPERATURE

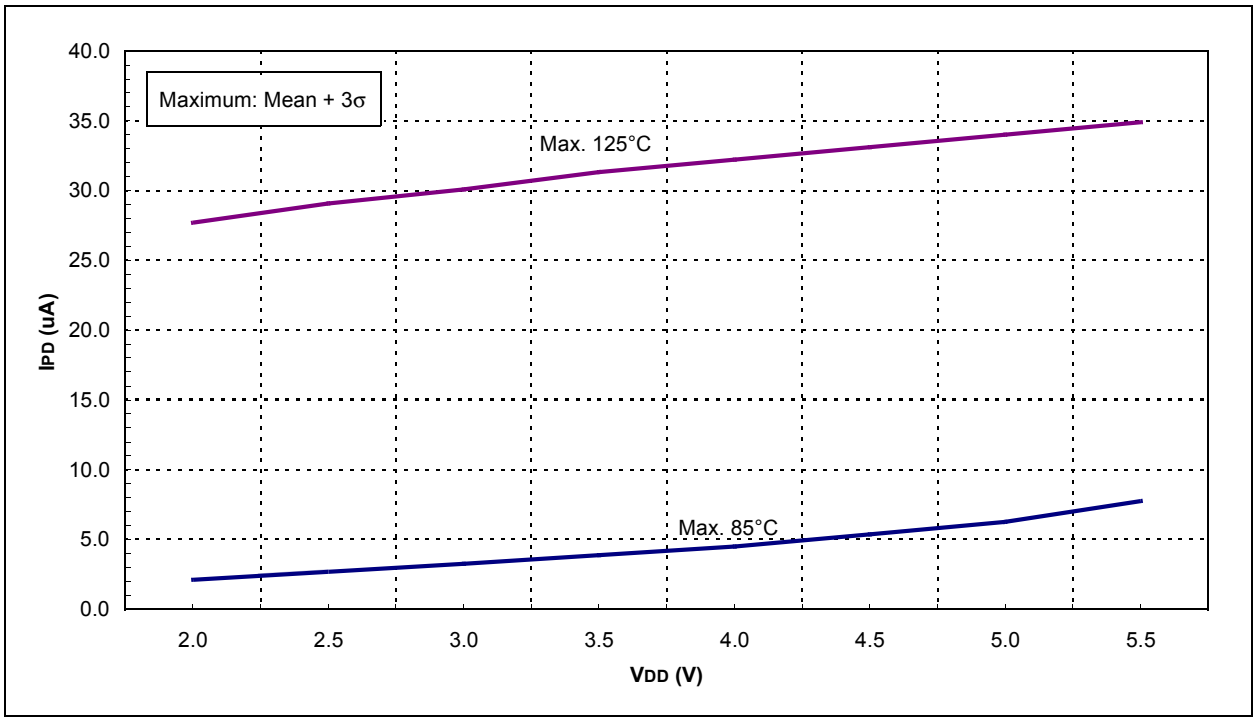


FIGURE 20-19: WDT PERIOD vs. VDD OVER TEMPERATURE

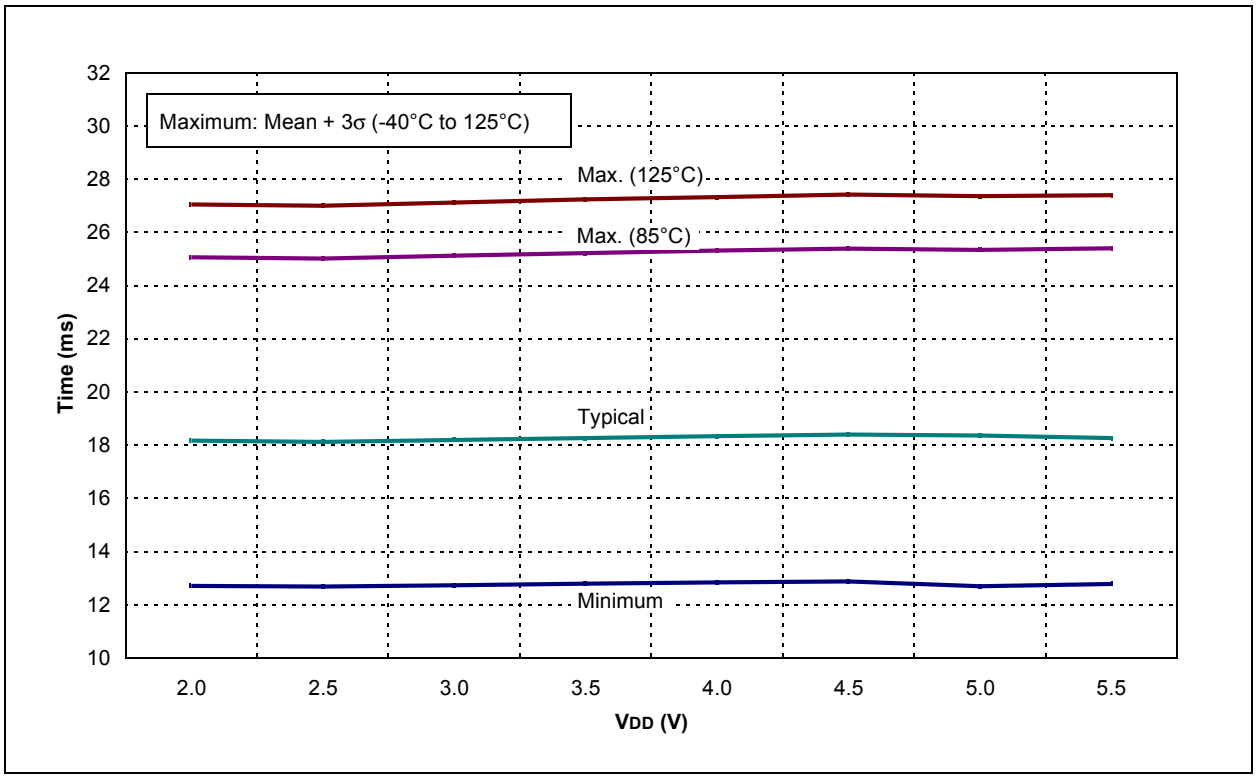


FIGURE 20-24: T1OSC IPD vs. VDD OVER TEMPERATURE (32 kHz)

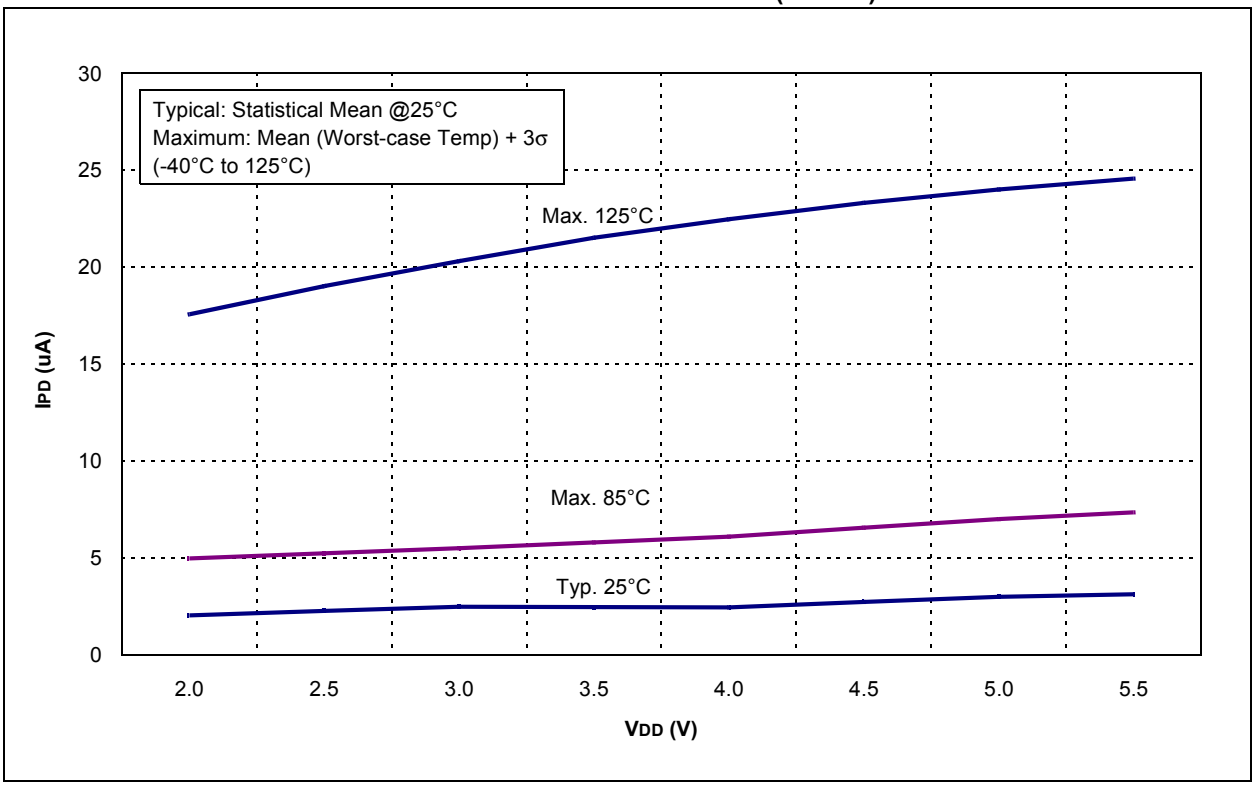


FIGURE 20-25: VOL vs. IOL OVER TEMPERATURE (VDD = 3.0V)

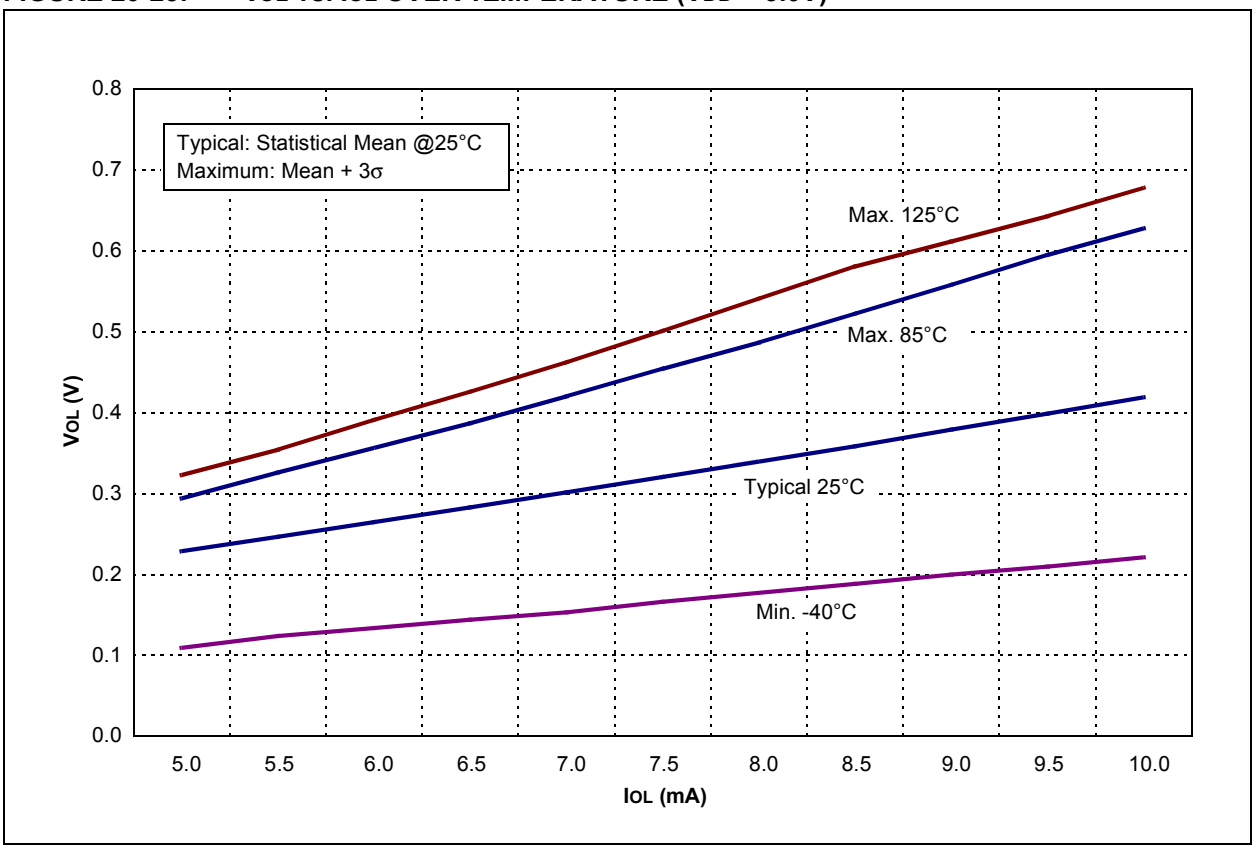


FIGURE 20-32: COMPARATOR RESPONSE TIME (FALLING EDGE)

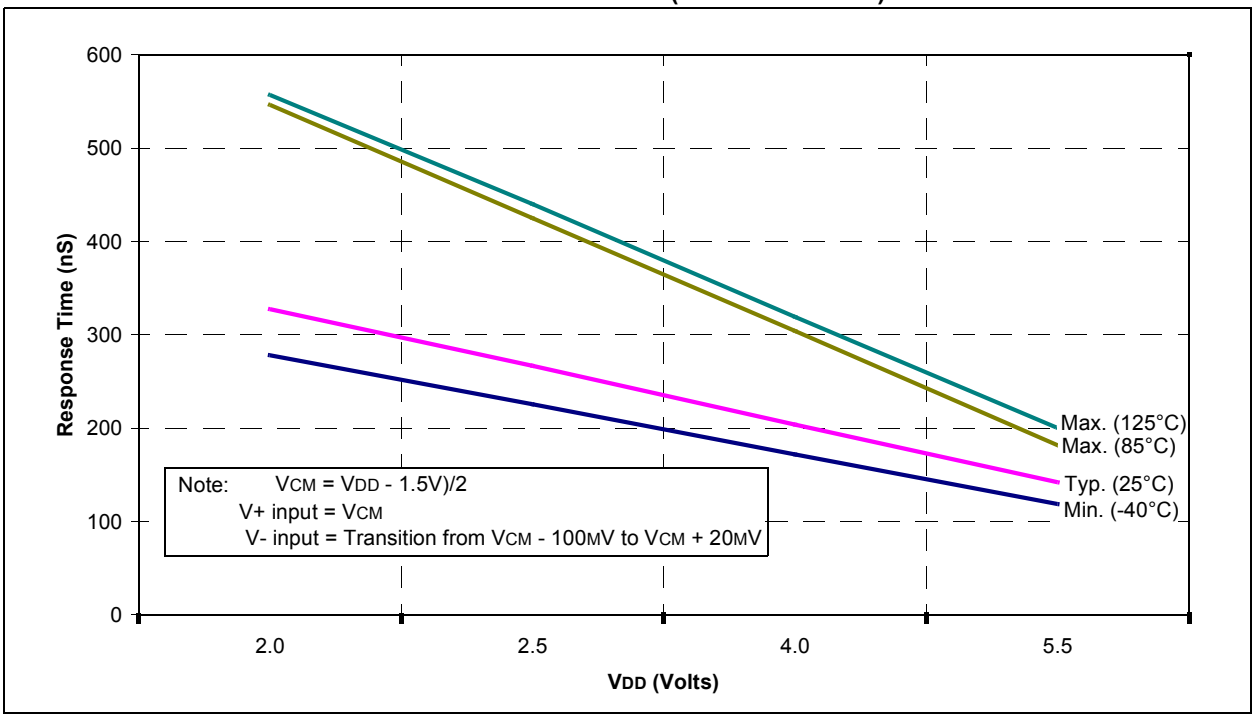


FIGURE 20-33: LFINTOSC FREQUENCY vs. VDD OVER TEMPERATURE (31 kHz)