



Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	192 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 11x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf723at-i-ss

2.0 MEMORY ORGANIZATION

2.1 Program Memory Organization

The PIC16(L)F722A/723A has a 13-bit program counter capable of addressing a 2K x 14 program memory space for the PIC16(L)F722A (0000h-07FFh) and a 4K x 14 program memory space for the PIC16(L)F723A (0000h-0FFFh). Accessing a location above the memory boundaries for the PIC16(L)F722A will cause a wrap-around within the first 2K x 14 program memory space. Accessing a location above the memory boundaries for the PIC16(L)F723A will cause a wrap-around within the first 4K x 14 program memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16(L)F722A

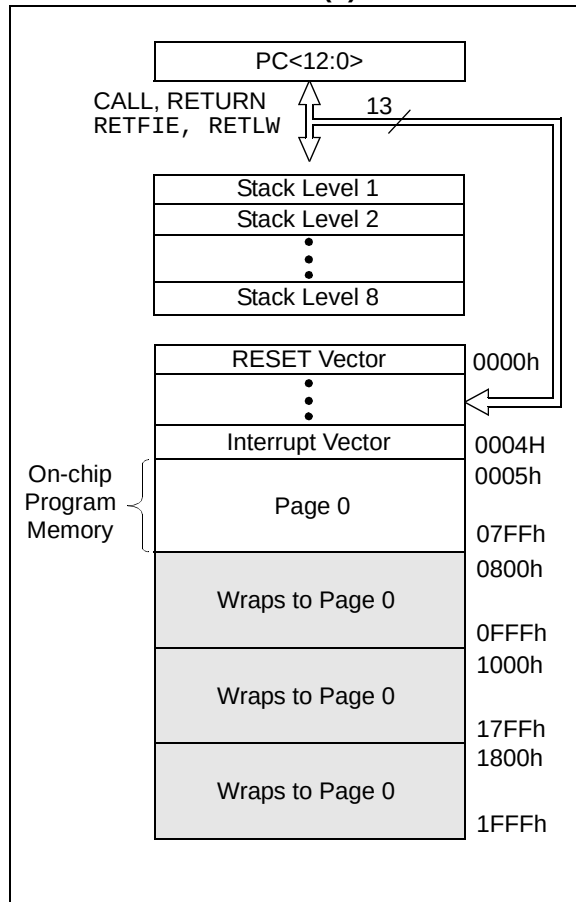
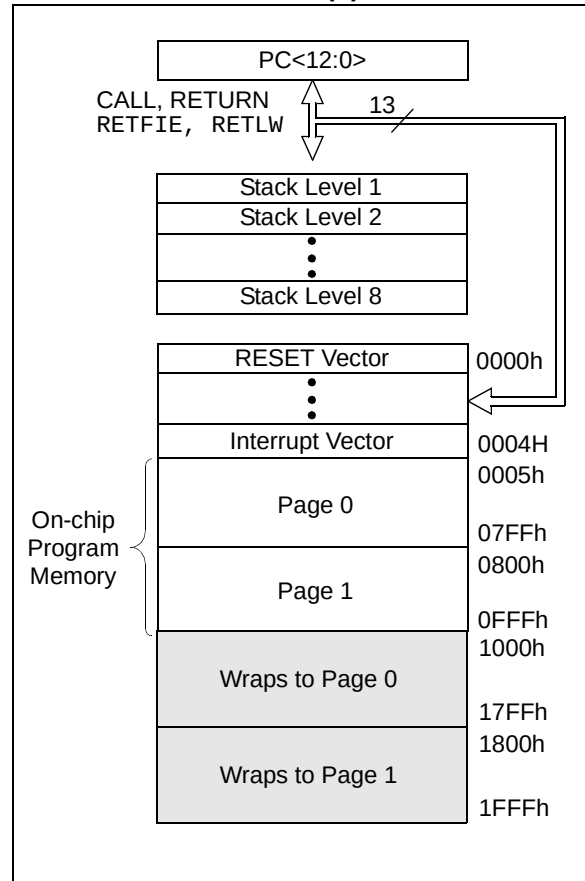


FIGURE 2-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16(L)F723A



PIC16(L)F722A/723A

FIGURE 2-4: PIC16(L)F723A SPECIAL FUNCTION REGISTERS

				File Address			
Indirect addr. (*)	00h	Indirect addr. (*)	80h	Indirect addr. (*)	100h	Indirect addr. (*)	180h
TMR0	01h	OPTION	81h	TMR0	101h	OPTION	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	ANSELA	185h
PORTB	06h	TRISB	86h		106h	ANSELB	186h
PORTC	07h	TRISC	87h		107h		187h
	08h		88h	CPSCON0	108h		188h
PORTE	09h	TRISE	89h	CPSCON1	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	Reserved	18Dh
TMR1L	0Eh	PCON	8Eh	PMDATH	10Eh	Reserved	18Eh
TMR1H	0Fh	T1GCON	8Fh	PMADRH	10Fh	Reserved	18Fh
T1CON	10h	OSCCON	90h		110h		190h
TMR2	11h	OSCTUNE	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD/SSPMSK	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h	WPUB	95h		115h		195h
CCPR1H	16h	IOCB	96h		116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch	APFCON	9Ch		11Ch		19Ch
CCP2CON	1Dh	FVRCON	9Dh		11Dh		19Dh
ADRES	1Eh		9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
	20h		A0h	General Purpose Register 16 Bytes	120h		1A0h
General Purpose Register 96 Bytes		General Purpose Register 80 Bytes			12Fh		
					130h		
			EFh		16Fh		1EFh
			F0h	Accesses 70h-7Fh	170h	Accesses 70h-7Fh	1F0h
	7Fh		FFh		17Fh		1FFh
Bank 0		Bank 1		Bank 2		Bank 3	

Legend: = Unimplemented data memory locations, read as '0'.
 * = Not a physical register.

2.2.2.2 OPTION register

The OPTION register, shown in Register 2-2, is a readable and writable register, which contains various control bits to configure:

- Timer0/WDT prescaler
- External RB0/INT interrupt
- Timer0
- Weak pull ups on PORTB

Note: To achieve a 1:1 prescaler assignment for Timer0, assign the prescaler to the WDT by setting PSA bit of the OPTION register to '1'. Refer to **Section 12.3 “Timer1 Prescaler”**.

REGISTER 2-2: OPTION_REG: OPTION 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
RBPUP	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
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 **RBPUP**: PORTB Pull-up Enable bit
 1 = PORTB pull ups are disabled
 0 = PORTB pull ups are enabled by individual bits in the WPUB register
- bit 6 **INTEDG**: Interrupt Edge Select bit
 1 = Interrupt on rising edge of RB0/INT pin
 0 = Interrupt on falling edge of RB0/INT pin
- bit 5 **T0CS**: Timer0 Clock Source Select bit
 1 = Transition on RA4/T0CKI pin
 0 = Internal instruction cycle clock (Fosc/4)
- bit 4 **T0SE**: Timer0 Source Edge Select bit
 1 = Increment on high-to-low transition on RA4/T0CKI pin
 0 = Increment on low-to-high transition on RA4/T0CKI pin
- bit 3 **PSA**: Prescaler Assignment bit
 1 = Prescaler is assigned to the WDT
 0 = Prescaler is assigned to the Timer0 module
- bit 2-0 **PS<2:0>**: Prescaler Rate Select bits

Bit Value	Timer0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

FIGURE 3-4: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{\text{MCLR}}$): CASE 1

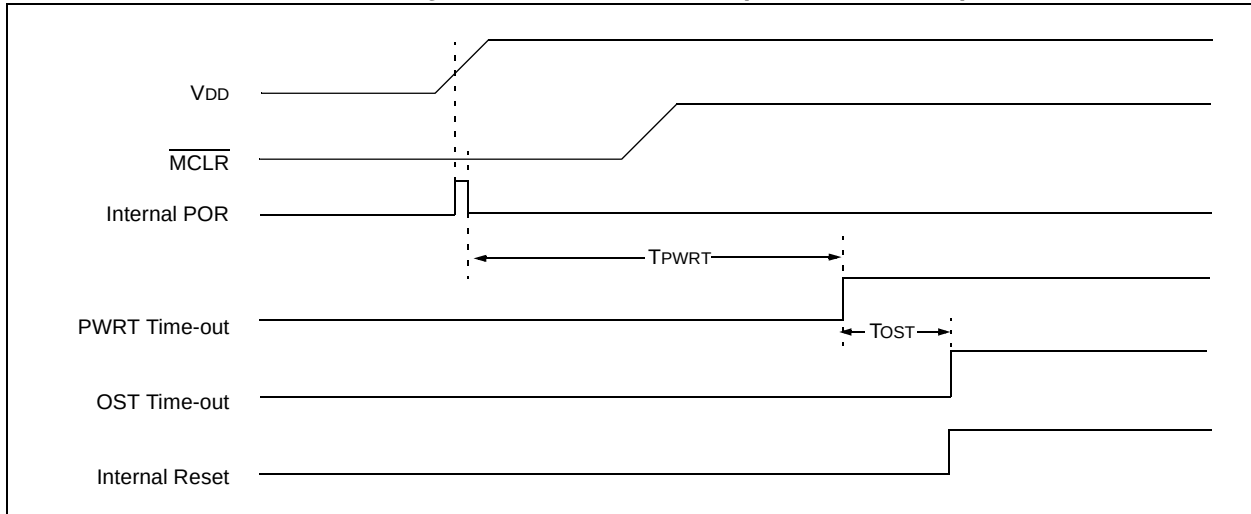


FIGURE 3-5: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{\text{MCLR}}$): CASE 2

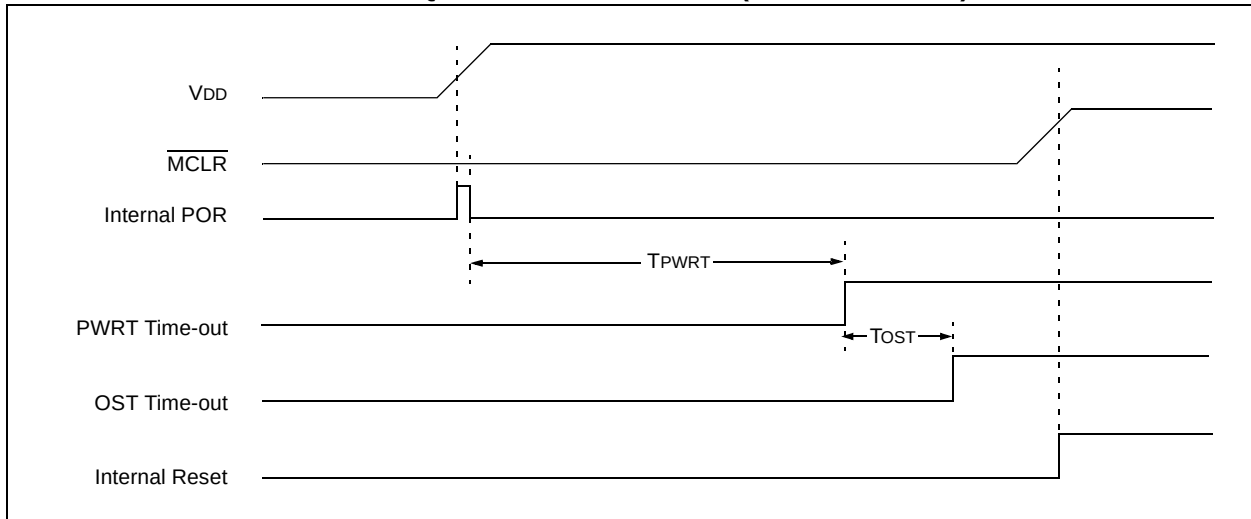
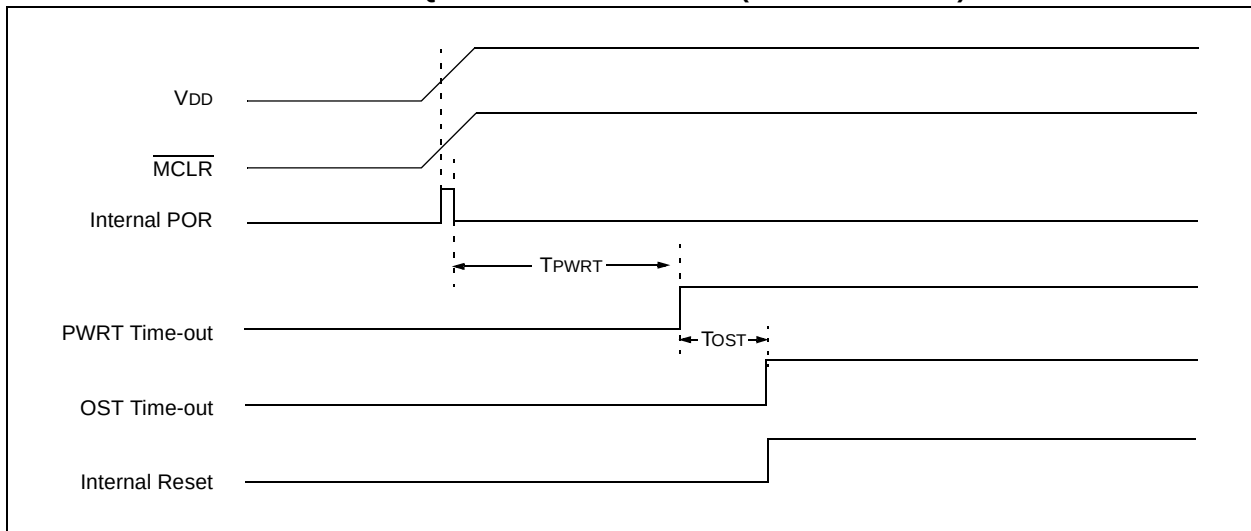


FIGURE 3-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ WITH VDD): CASE 3



PIC16(L)F722A/723A

6.0 I/O PORTS

There are as many as thirty-five general purpose I/O pins available. Depending on which peripherals are enabled, some or all of the pins may not be available as general purpose I/O. In general, when a peripheral is enabled, the associated pin may not be used as a general purpose I/O pin.

6.1 Alternate Pin Function

The Alternate Pin Function Control (APFCON) register is used to steer specific peripheral input and output functions between different pins. The APFCON register is shown in Register 6-1. For this device family, the following functions can be moved between different pins:

- $\overline{SS}$ (Slave Select)
- CCP2

REGISTER 6-1: APFCON: ALTERNATE PIN FUNCTION CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SSSEL	CCP2SEL
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 7-2 **Unimplemented:** Read as '0'.

bit 1 **SSSEL:** $\overline{SS}$ Input Pin Selection bit

0 = $\overline{SS}$ function is on RA5/AN4/CPS7/ $\overline{SS}$ /VCAP

1 = $\overline{SS}$ function is on RA0/AN0/ $\overline{SS}$ /VCAP

bit 0 **CCP2SEL:** CCP2 Input/Output Pin Selection bit

0 = CCP2 function is on RC1/T1OSI/CCP2

1 = CCP2 function is on RB3/CCP2

9.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Results formatting

For correct conversion, the appropriate TAD specification must be met. Refer to the A/D conversion requirements in **Section 23.0 “Electrical Specifications”** for more information. Table 9-1 gives examples of appropriate ADC clock selections.

Note: Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.

9.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 6.0 “I/O Ports”** for more information.

Note: Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

9.1.2 CHANNEL SELECTION

The CHS bits of the ADCON0 register determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 9.2 “ADC Operation”** for more information.

9.1.3 ADC VOLTAGE REFERENCE

The ADREF bits of the ADCON1 register provides control of the positive voltage reference. The positive voltage reference can be either VDD, an external voltage source or the internal Fixed Voltage Reference. The negative voltage reference is always connected to the ground reference. See **Section 10.0 “Fixed Voltage Reference”** for more details.

9.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS bits of the ADCON1 register. There are seven possible clock options:

- Fosc/2
- Fosc/4
- Fosc/8
- Fosc/16
- Fosc/32
- Fosc/64
- FRC (dedicated internal oscillator)

The time to complete one bit conversion is defined as TAD. One full 8-bit conversion requires 10 TAD periods as shown in Figure 9-2.

PIC16(L)F722A/723A

10.0 FIXED VOLTAGE REFERENCE

This device contains an internal voltage regulator. To provide a reference for the regulator, a band gap reference is provided. This band gap is also user accessible via an A/D converter channel.

User level band gap functions are controlled by the FVRCON register, which is shown in Register 10-1.

REGISTER 10-1: FVRCON: FIXED VOLTAGE REFERENCE REGISTER

R-q	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
FVRRDY	FVREN	—	—	—	—	ADFVR1	ADFVR0
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

q = Value depends on condition

bit 7 **FVRRDY:** Fixed Voltage Reference Ready Flag bit
0 = Fixed Voltage Reference output is not active or stable
1 = Fixed Voltage Reference output is ready for use

bit 6 **FVREN⁽¹⁾:** Fixed Voltage Reference Enable bit
0 = Fixed Voltage Reference is disabled
1 = Fixed Voltage Reference is enabled

bit 5-2 **Unimplemented:** Read as '0'

bit 1-0 **ADFVR<1:0>:** A/D Converter Fixed Voltage Reference Selection bits
00 = A/D Converter Fixed Voltage Reference Peripheral output is off.
01 = A/D Converter Fixed Voltage Reference Peripheral output is 1x (1.024V)
10 = A/D Converter Fixed Voltage Reference Peripheral output is 2x (2.048V)⁽¹⁾
11 = A/D Converter Fixed Voltage Reference Peripheral output is 4x (4.096V)⁽¹⁾

Note 1: Fixed Voltage Reference output cannot exceed V_{DD}.

PIC16(L)F722A/723A

12.0 TIMER1 MODULE WITH GATE CONTROL

The Timer1 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 3-bit prescaler
- Dedicated LP oscillator circuit
- Synchronous or asynchronous operation
- Multiple Timer1 gate (count enable) sources
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- Time base for the Capture/Compare function
- Special Event Trigger (with CCP)

- Selectable Gate Source Polarity
- Gate Toggle Mode
- Gate Single-pulse Mode
- Gate Value Status
- Gate Event Interrupt

Figure 12-1 is a block diagram of the Timer1 module.

FIGURE 12-1: TIMER1 BLOCK DIAGRAM

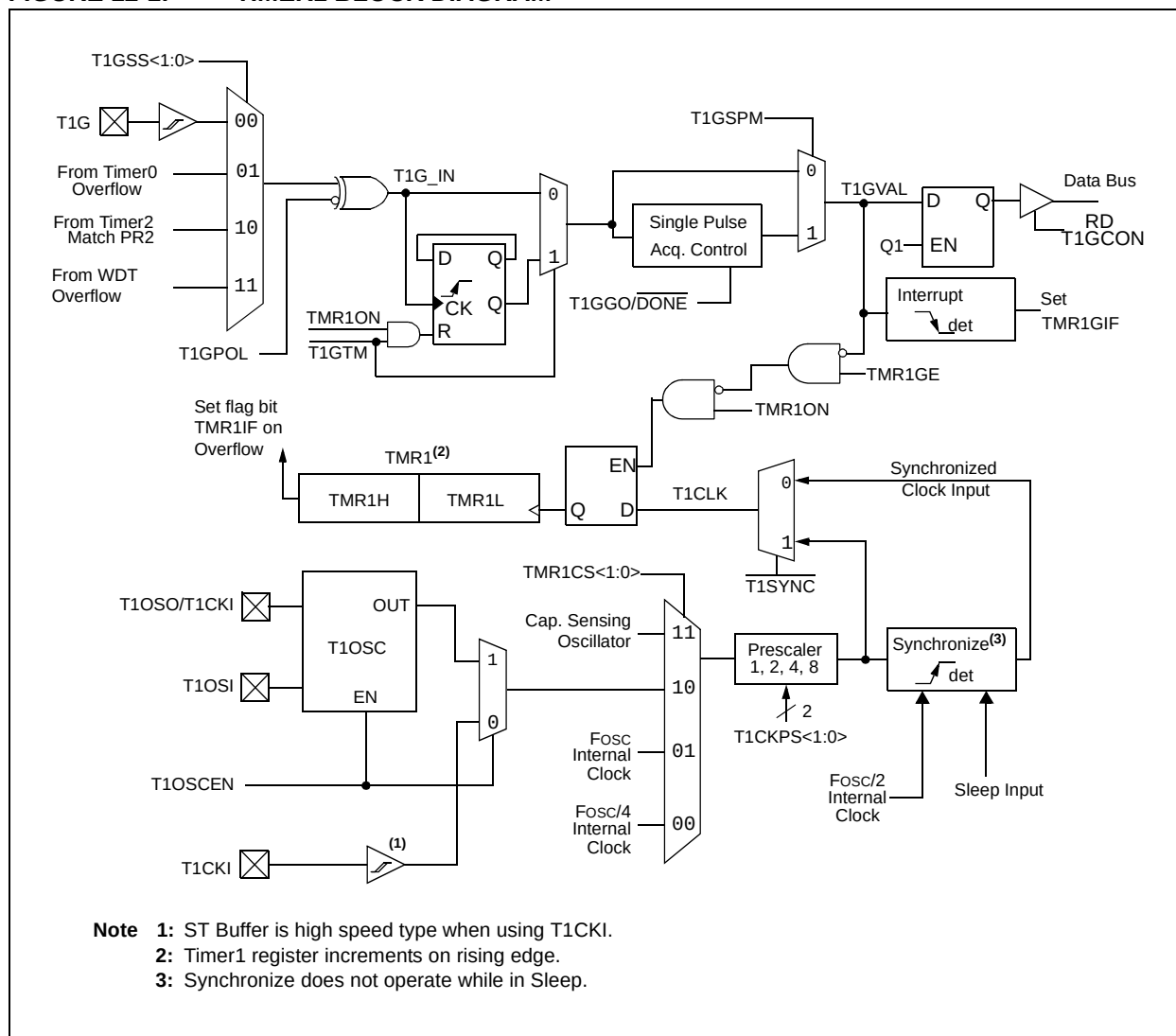


TABLE 12-2: WDT/TIMER1 GATE INTERACTION

WDTE	TMR1GE = 1 and T1GSS = 11	WDT Oscillator Enable	WDT Reset	Wake-up	WDT Available for T1G Source
1	N	Y	Y	Y	N
1	Y	Y	Y	Y	Y
0	Y	Y	N	N	Y
0	N	N	N	N	N

12.6.3 TIMER1 GATE TOGGLE MODE

When Timer1 Gate Toggle mode is enabled, it is possible to measure the full-cycle length of a Timer1 gate signal, as opposed to the duration of a single level pulse.

The Timer1 gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. See Figure 12-5 for timing details.

Timer1 Gate Toggle mode is enabled by setting the T1GTM bit of the T1GCON register. When the T1GTM bit is cleared, the flip-flop is cleared and held clear. This is necessary in order to control which edge is measured.

Note: Enabling Toggle mode at the same time as changing the gate polarity may result in indeterminate operation.

12.6.4 TIMER1 GATE SINGLE-PULSE MODE

When Timer1 Gate Single-Pulse mode is enabled, it is possible to capture a single pulse gate event. Timer1 Gate Single-Pulse mode is first enabled by setting the T1GSPM bit in the T1GCON register. Next, the T1GGO/DONE bit in the T1GCON register must be set. The Timer1 will be fully enabled on the next incrementing edge. On the next trailing edge of the pulse, the T1GGO/DONE bit will automatically be cleared. No other gate events will be allowed to increment Timer1 until the T1GGO/DONE bit is once again set in software.

Clearing the T1GSPM bit of the T1GCON register will also clear the T1GGO/DONE bit. See Figure 12-6 for timing details.

Enabling the Toggle mode and the Single-Pulse mode simultaneously will permit both sections to work together. This allows the cycle times on the Timer1 gate source to be measured. See Figure 12-7 for timing details.

12.6.5 TIMER1 GATE VALUE STATUS

When Timer1 gate value status is utilized, it is possible to read the most current level of the gate control value. The value is stored in the T1GVAL bit in the T1GCON register. The T1GVAL bit is valid even when the Timer1 gate is not enabled (TMR1GE bit is cleared).

12.6.6 TIMER1 GATE EVENT INTERRUPT

When Timer1 gate event interrupt is enabled, it is possible to generate an interrupt upon the completion of a gate event. When the falling edge of T1GVAL occurs, the TMR1GIF flag bit in the PIR1 register will be set. If the TMR1GIE bit in the PIE1 register is set, then an interrupt will be recognized.

The TMR1GIF flag bit operates even when the Timer1 gate is not enabled (TMR1GE bit is cleared).

PIC16(L)F722A/723A

TABLE 15-3: SUMMARY OF REGISTERS ASSOCIATED WITH CAPTURE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	53
APFCON	—	—	—	—	—	—	SSSEL	CCP2SEL	42
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	115
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	115
CCPRxL	Capture/Compare/PWM Register X Low Byte								116
CCPRxH	Capture/Compare/PWM Register X High Byte								116
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	36
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	37
PIE2	—	—	—	—	—	—	—	CCP2IE	38
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	39
PIR2	—	—	—	—	—	—	—	CCP2IF	40
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	—	TMR1ON	103
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS1	T1GSS0	104
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								99
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								99
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	52
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	62

Legend: - = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Capture.

16.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error Status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RCSTA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RCREG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = 1) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RCSTA register which resets the AUSART. Clearing the CREN bit of the RCSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

Note: If all receive characters in the receive FIFO have framing errors, repeated reads of the RCREG will not clear the FERR bit.
--

16.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RCSTA register is set. The characters already in the FIFO buffer can be read but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RCSTA register or by setting the AUSART by clearing the SPEN bit of the RCSTA register.

16.1.2.6 Receiving 9-bit Characters

The AUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set the AUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCSTA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCREG.

16.1.2.7 Address Detection

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems. Address detection is enabled by setting the ADDEN bit of the RCSTA register.

Address detection requires 9-bit character reception. When address detection is enabled, only characters with the ninth data bit set will be transferred to the receive FIFO buffer, thereby setting the RCIF interrupt bit of the PIR1 register. All other characters will be ignored.

Upon receiving an address character, user software determines if the address matches its own. Upon address match, user software must disable address detection by clearing the ADDEN bit before the next Stop bit occurs. When user software detects the end of the message, determined by the message protocol used, software places the receiver back into the Address Detection mode by setting the ADDEN bit.

PIC16(L)F722A/723A

17.2.2 START AND STOP CONDITIONS

During times of no data transfer (Idle time), both the clock line (SCL) and the data line (SDA) are pulled high through external pull-up resistors. The Start and Stop conditions determine the start and stop of data transmission. The Start condition is defined as a high-to-low transition of the SDA line while SCL is high. The Stop condition is defined as a low-to-high transition of the SDA line while SCL is high.

Figure 17-9 shows the Start and Stop conditions. A master device generates these conditions for starting and terminating data transfer. Due to the definition of the Start and Stop conditions, when data is being transmitted, the SDA line can only change state when the SCL line is low.

17.2.3 ACKNOWLEDGE

After the valid reception of an address or data byte, the hardware automatically will generate the Acknowledge (ACK) pulse and load the SSPBUF register with the received value currently in the SSPSR register. There are certain conditions that will cause the SSP module not to generate this ACK pulse. They include any or all of the following:

- The Buffer Full bit, BF of the SSPSTAT register, was set before the transfer was received.
- The SSP Overflow bit, SSPOV of the SSPCON register, was set before the transfer was received.
- The SSP module is being operated in Firmware Master mode.

In such a case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF of the PIR1 register is set. Table 17-2 shows the results of when a data transfer byte is received, given the status of bits BF and SSPOV. Flag bit BF is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

FIGURE 17-9: START AND STOP CONDITIONS

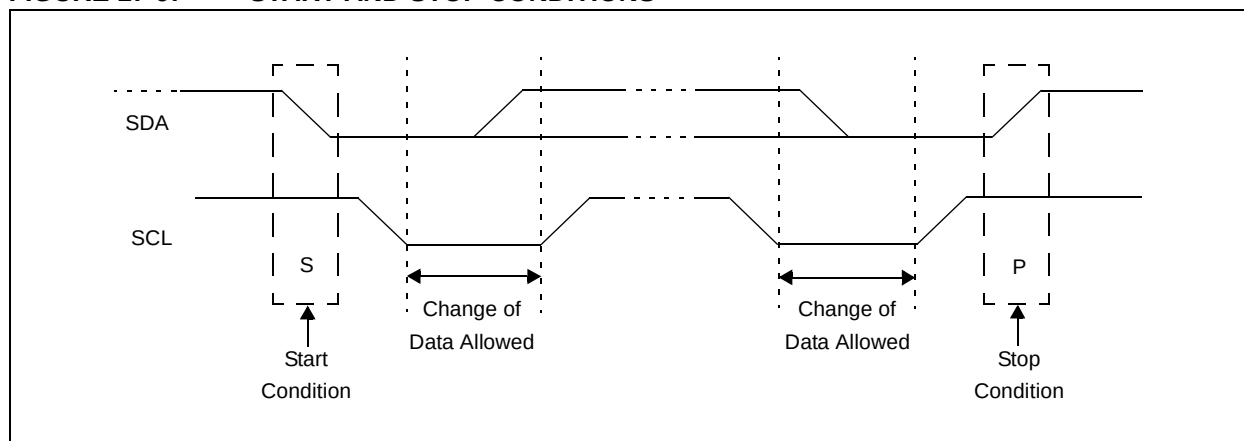
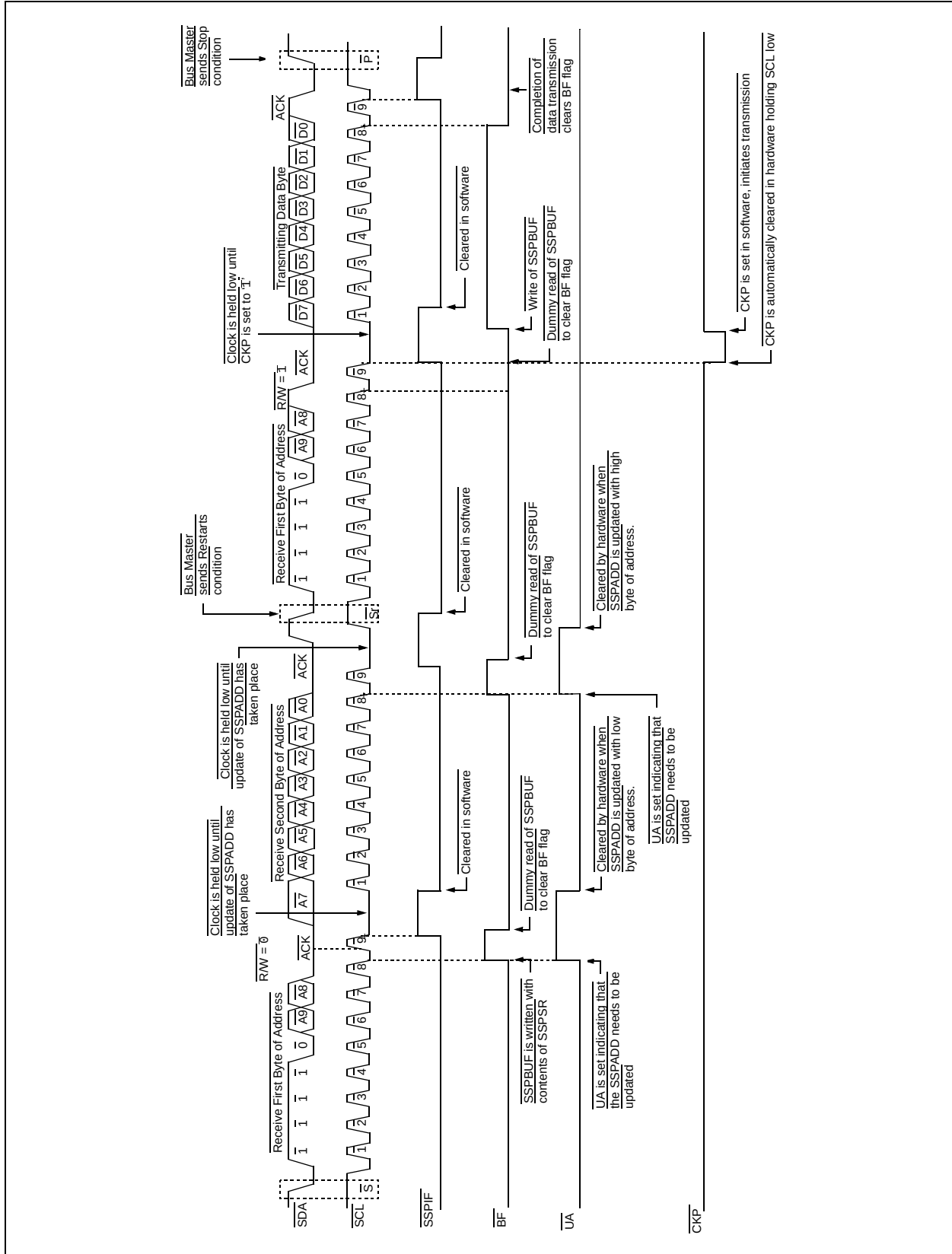


TABLE 17-2: DATA TRANSFER RECEIVED BYTE ACTIONS

Status Bits as Data Transfer is Received		SSPSR → SSPBUF	Generate $\overline{\text{ACK}}$ Pulse	Set bit SSPIF (SSP Interrupt occurs if enabled)
BF	SSPOV			
0	0	Yes	Yes	Yes
1	0	No	No	Yes
1	1	No	No	Yes
0	1	No	No	Yes

Note 1: Shaded cells show the conditions where the user software did not properly clear the overflow condition.

FIGURE 17-13: I²C SLAVE MODE TIMING (TRANSMISSION 10-BIT ADDRESS)



21.0 INSTRUCTION SET SUMMARY

The PIC16(L)F722A/723A instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 instruction is a 14-bit word divided into an **opcode**, which specifies the instruction type and one or more **operands**, which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 21-1, while the various opcode fields are summarized in Table 21-1.

Table 21-2 lists the instructions recognized by the MPASM™ assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution time of 1 μs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

21.1 Read-Modify-Write Operations

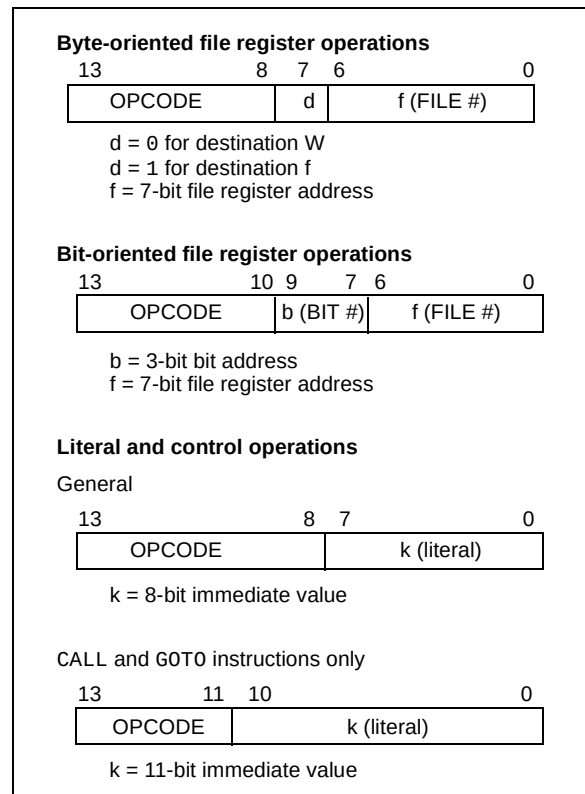
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a CLRF PORTB instruction will read PORTB, clear all the data bits, then write the result back to PORTB. This example would have the unintended consequence of clearing the condition that set the RBIF flag.

TABLE 21-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
TO	Time-out bit
C	Carry bit
DC	Digit carry bit
Z	Zero bit
PD	Power-down bit

FIGURE 21-1: GENERAL FORMAT FOR INSTRUCTIONS



PIC16(L)F722A/723A

BTFSS **Bit Test f, Skip if Set**

Syntax: [*label*] BTFSS *f*,*b*

Operands: $0 \leq f \leq 127$
 $0 \leq b < 7$

Operation: skip if (*f*<*b*>) = 1

Status Affected: None

Description: If bit 'b' in register 'f' is '0', the next instruction is executed.
 If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.

CLRWDT **Clear Watchdog Timer**

Syntax: [*label*] CLRWDT

Operands: None

Operation: 00h → WDT
 0 → WDT prescaler,
 1 → $\overline{TO}$
 1 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Description: CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT.
 Status bits $\overline{TO}$ and $\overline{PD}$ are set.

CALL **Call Subroutine**

Syntax: [*label*] CALL *k*

Operands: $0 \leq k \leq 2047$

Operation: (PC)+1 → TOS,
 k → PC<10:0>,
 (PCLATH<4:3>) → PC<12:11>

Status Affected: None

Description: Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The 11-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a 2-cycle instruction.

COMF **Complement f**

Syntax: [*label*] COMF *f*,*d*

Operands: $0 \leq f \leq 127$
 d ∈ [0,1]

Operation: ($\bar{f}$) → (destination)

Status Affected: Z

Description: The contents of register 'f' are complemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

CLRF **Clear f**

Syntax: [*label*] CLRF *f*

Operands: $0 \leq f \leq 127$

Operation: 00h → (*f*)
 1 → Z

Status Affected: Z

Description: The contents of register 'f' are cleared and the Z bit is set.

DECF **Decrement f**

Syntax: [*label*] DECF *f*,*d*

Operands: $0 \leq f \leq 127$
 d ∈ [0,1]

Operation: (*f*) - 1 → (destination)

Status Affected: Z

Description: Decrement register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

CLRW **Clear W**

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W)
 1 → Z

Status Affected: Z

Description: W register is cleared. Zero bit (Z) is set.

22.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
 - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
 - MPLAB XC Compiler
 - MPASM™ Assembler
 - MPLINK™ Object Linker/
MPLIB™ Object Librarian
 - MPLAB Assembler/Linker/Librarian for
Various Device Families
- Simulators
 - MPLAB X SIM Software Simulator
- Emulators
 - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
 - MPLAB ICD 3
 - PICKit™ 3
- Device Programmers
 - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards,
Evaluation Kits and Starter Kits
- Third-party development tools

22.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

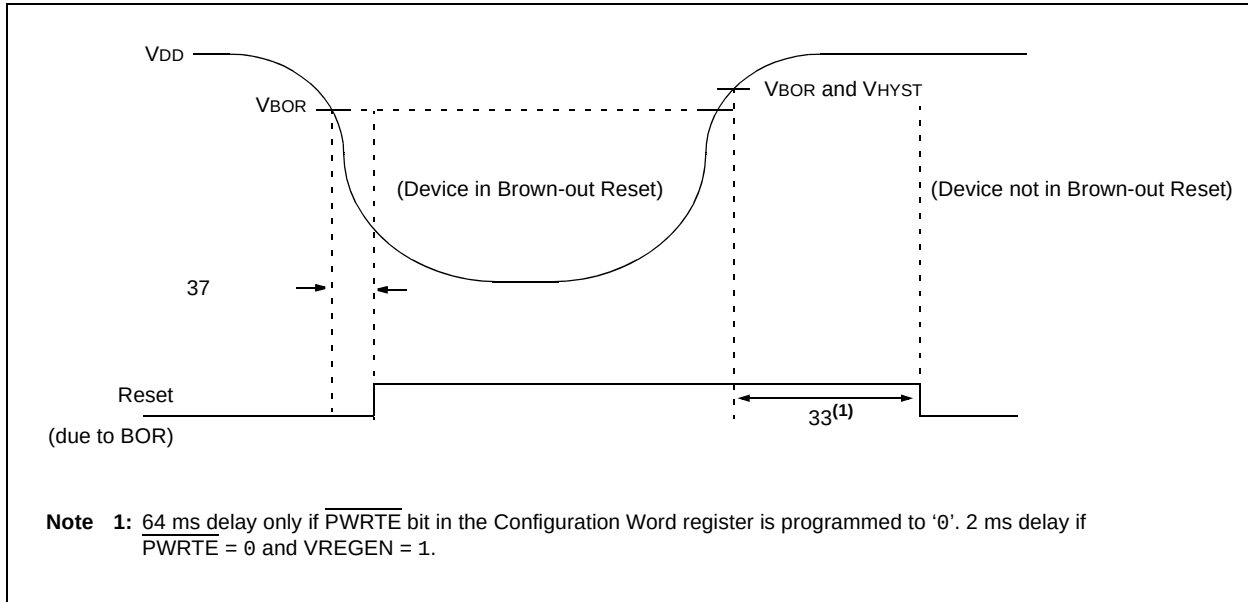
- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

PIC16(L)F722A/723A

FIGURE 23-9: BROWN-OUT RESET TIMING AND CHARACTERISTICS



PIC16(L)F722A/723A

FIGURE 24-8: PIC16LF722A/723A TYPICAL I_{DD} vs. V_{DD} OVER F_{OSC} , EXTRC MODE

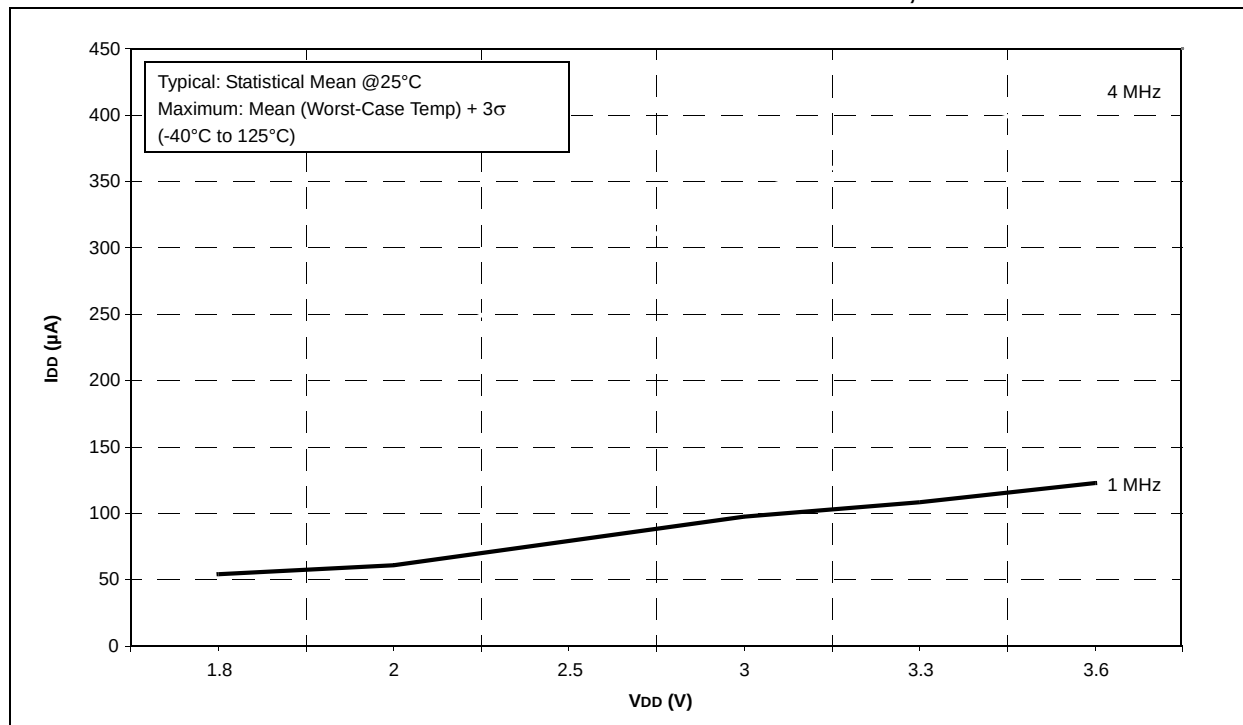
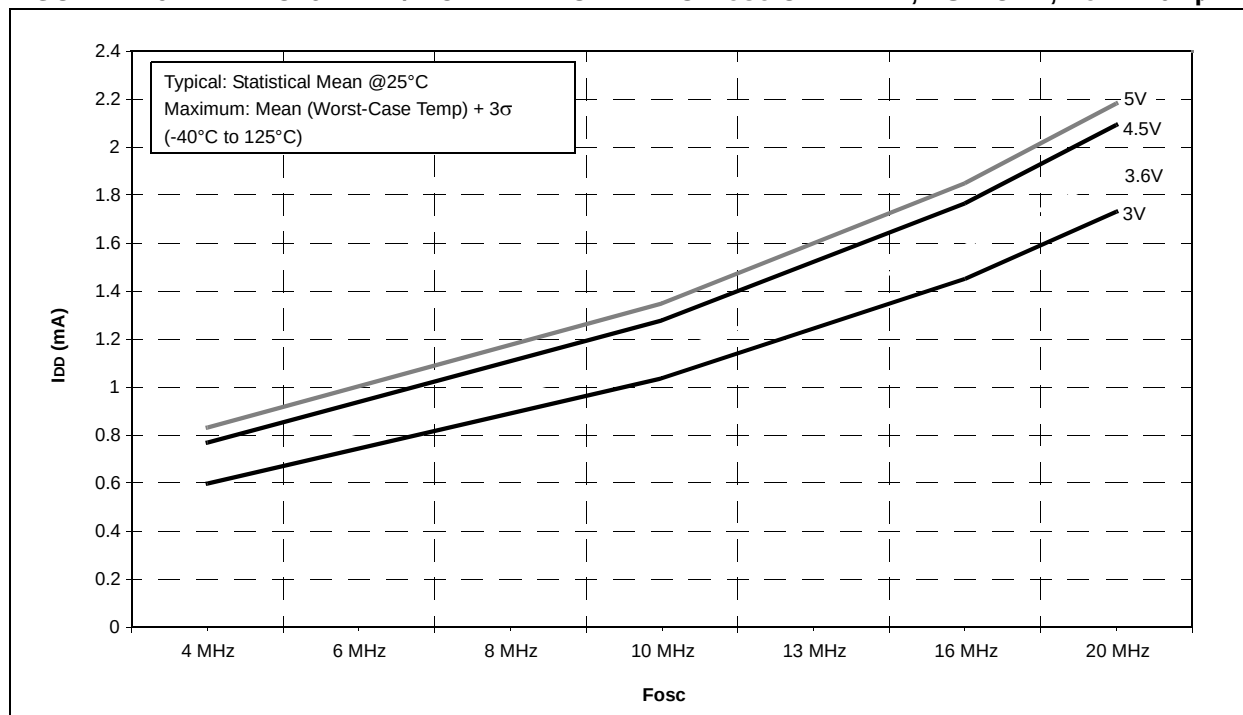


FIGURE 24-9: PIC16F722A/723A MAXIMUM I_{DD} vs. F_{OSC} OVER V_{DD} , HS MODE, $V_{CAP} = 0.1\mu F$



PIC16(L)F722A/723A

FIGURE 24-52: V_{OH} vs. I_{OH} OVER TEMPERATURE, $V_{DD} = 5.5V$

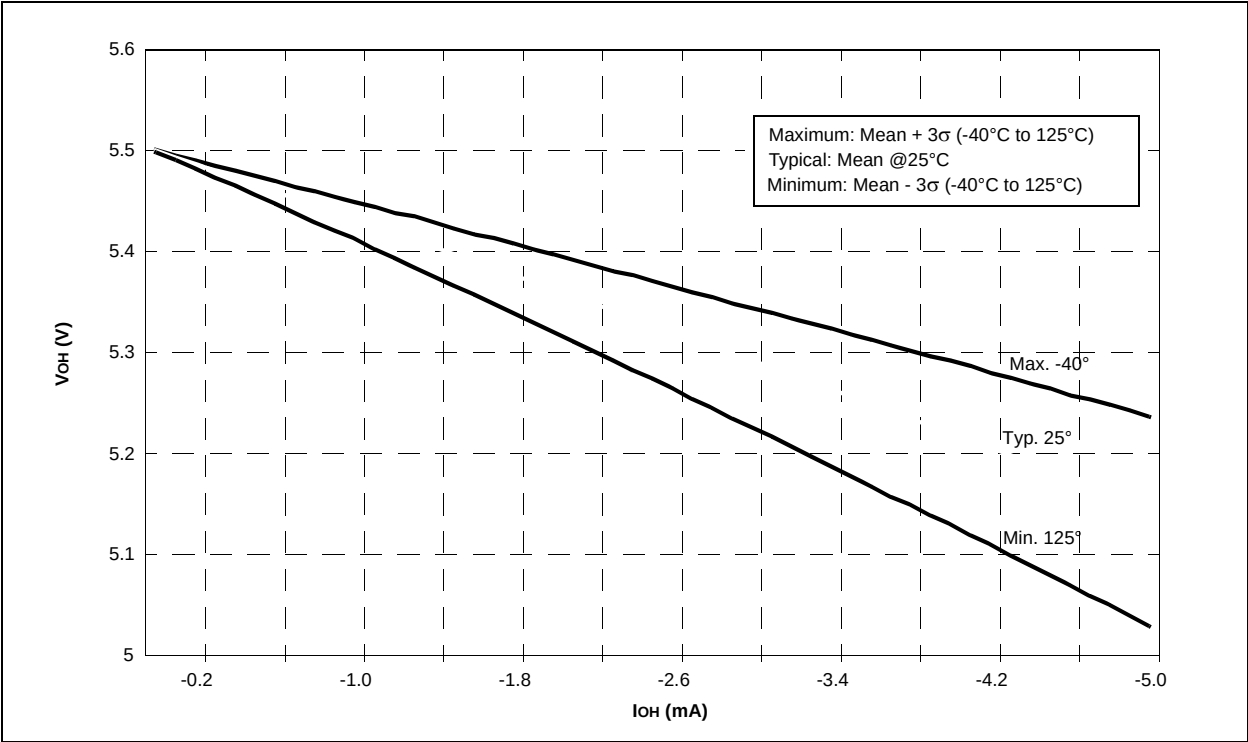
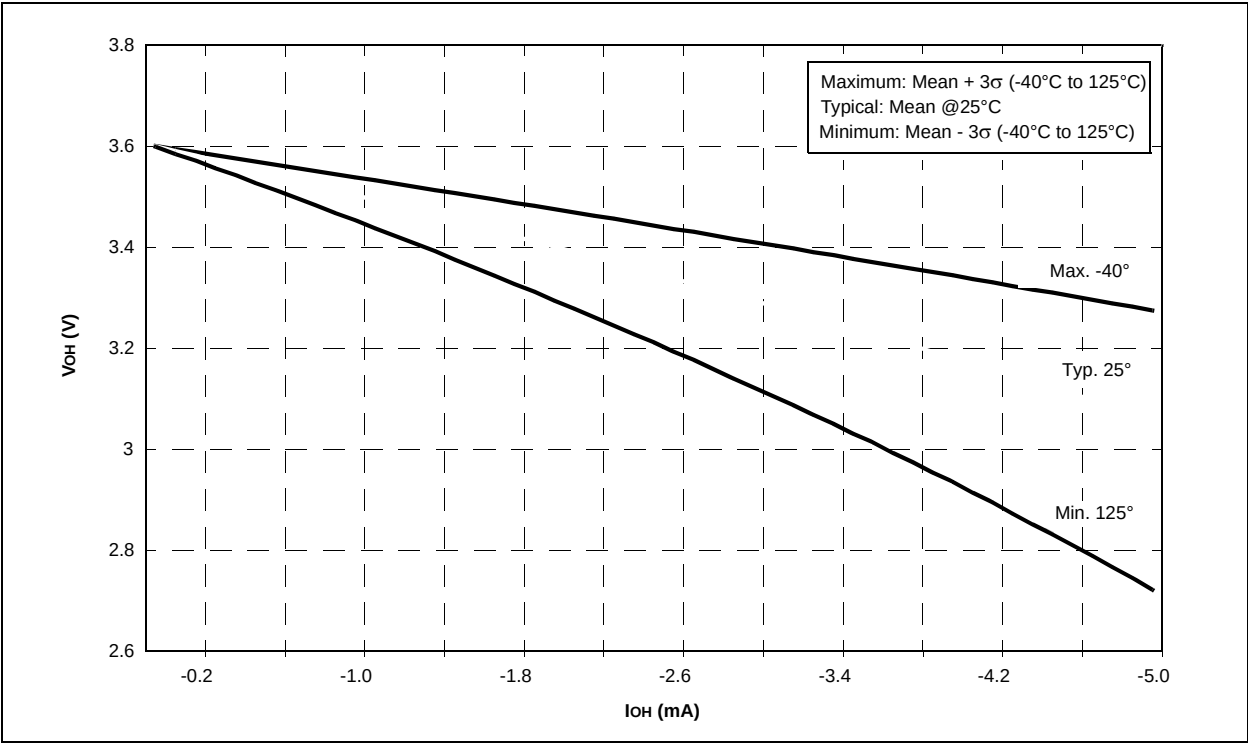


FIGURE 24-53: V_{OH} vs. I_{OH} OVER TEMPERATURE, $V_{DD} = 3.6V$



PIC16(L)F722A/723A

FIGURE 24-68: TYPICAL FVR (X1 AND X2) VS. SUPPLY VOLTAGE (V) NORMALIZED AT 3.0V

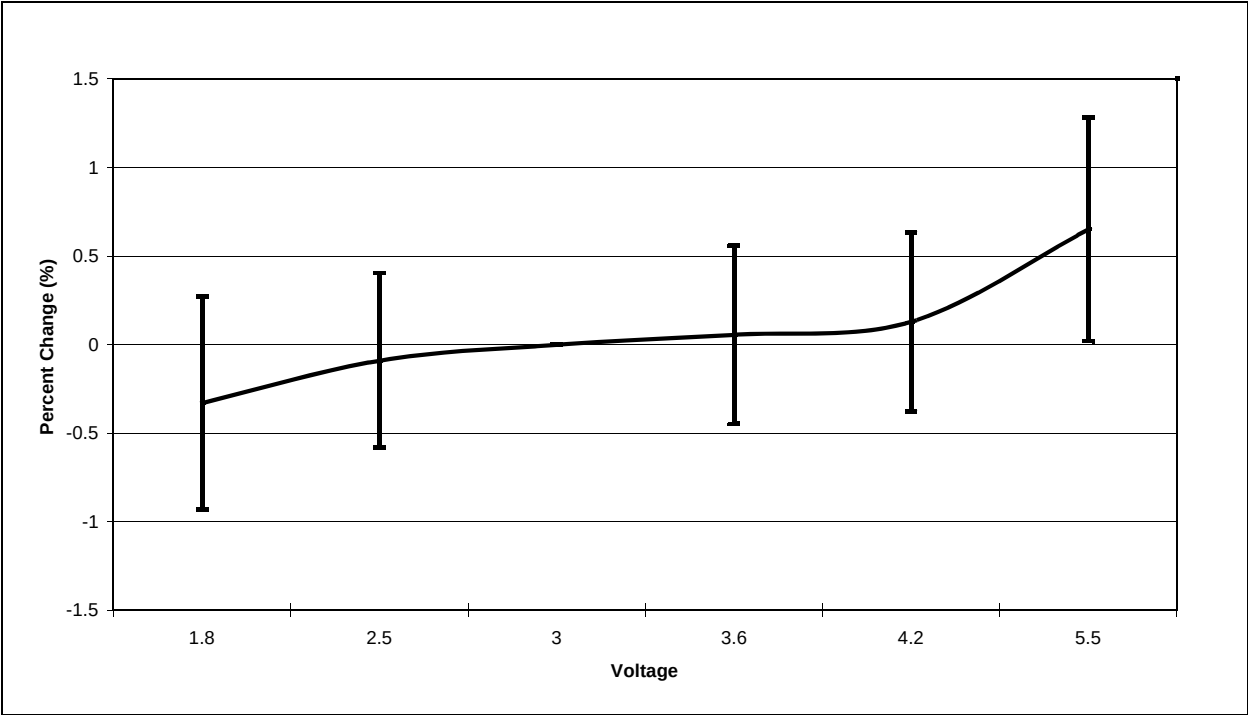


FIGURE 24-69: TYPICAL FVR CHANGE VS. TEMPERATURE NORMALIZED AT 25°C

