

Welcome to E-XFL.COM

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

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

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

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	6
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 5x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	8-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f15313-e-p

PIC16(L)F15313/23

TABLE 4-9: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 (ALL BANKS)

Bank Offset Bank 0-Bank 63	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR	
All Banks												
x00h or x80h	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
x01h or x81h	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
x02h or x82h	PCL	PCL								0000 0000	0000 0000	
x03h or x83h	STATUS	—	—	—	\overline{TO}	\overline{PD}	Z	DC	C	---1 1000	---q quuu	
x04h or x84h	FSR0L	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
x05h or x85h	FSR0H	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
x06h or x86h	FSR1L	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
x07h or x87h	FSR1H	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
x08h or x88h	BSR	—	—	BSR<5:0>						--00 0000	--00 0000	
x09h or x89h	WREG	Working Register								0000 0000	uuuu uuuu	
x0Ah or x8Ah	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
x0Bh or x8Bh	INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	00-- ---1	00-- ---1	

Legend: x = unknown, u = unchanged, q = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.

Note 1: These Registers can be accessed from any bank.

REGISTER 10-3: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
OSFIE	CSWIE	—	—	—	—	—	ADIE
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

- bit 7 **OSFIE:** Oscillator Fail Interrupt Enable bit
 1 = Enables the Oscillator Fail Interrupt
 0 = Disables the Oscillator Fail Interrupt
- bit 6 **CSWIE:** Clock Switch Complete Interrupt Enable bit
 1 = The clock switch module interrupt is enabled
 0 = The clock switch module interrupt is disabled
- bit 5-1 **Unimplemented:** Read as '0'
- bit 0 **ADIE:** Analog-to-Digital Converter (ADC) Interrupt Enable bit
 1 = Enables the ADC interrupt
 0 = Disables the ADC interrupt

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE7

REGISTER 10-6: PIE4: PERIPHERAL INTERRUPT ENABLE REGISTER 4

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	TMR2IE	TMR1IE
bit 7						bit 0	

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

HS = Hardware set

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit

1 = Enables the Timer2 to PR2 match interrupt

0 = Disables the Timer2 to PR2 match interrupt

bit 0 **TMR1IE:** Timer1 Overflow Interrupt Enable bit

1 = Enables the Timer1 overflow interrupt

0 = Enables the Timer1 overflow interrupt

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE7.

REGISTER 10-13: PIR3: PERIPHERAL INTERRUPT REQUEST REGISTER 3

U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	U-0	U-0	R/W/HS-0/0	R/W/HS-0/0
—	—	RC1IF	TX1IF	—	—	BCL1IF	SSP1IF
bit 7						bit 0	

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HS = Hardware clearable

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

bit 5 **RC1IF:** EUSART1 Receive Interrupt Flag (read-only) bit ⁽¹⁾

1 = The EUSART1 receive buffer is not empty (contains at least one byte)

0 = The EUSART1 receive buffer is empty

bit 4 **TX1IF:** EUSART1 Transmit Interrupt Flag (read-only) bit ⁽²⁾

1 = The EUSART1 transmit buffer contains at least one unoccupied space

0 = The EUSART1 transmit buffer is currently full. The application firmware should not write to TXxREG again, until more room becomes available in the transmit buffer.

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **BCL1IF:** MSSP1 Bus Collision Interrupt Flag bit

1 = A bus collision was detected (must be cleared in software)

0 = No bus collision was detected

bit 0 **SSP1IF:** Synchronous Serial Port (MSSP1) Interrupt Flag bit

1 = The Transmission/Reception/Bus Condition is complete (must be cleared in software)

0 = Waiting for the Transmission/Reception/Bus Condition in progress

Note 1: The RCxIF flag is a read-only bit. To clear the RCxIF flag, the firmware must read from RCxREG enough times to remove all bytes from the receive buffer.

2: The TXxIF flag is a read-only bit, indicating if there is room in the transmit buffer. To clear the TX1IF flag, the firmware must write enough data to TXxREG to completely fill all available bytes in the buffer. The TXxIF flag does not indicate transmit completion (use TRMT for this purpose instead).

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

13.0 NONVOLATILE MEMORY (NVM) CONTROL

NVM consists of the Program Flash Memory (PFM).

NVM is accessible by using both the FSR and INDF registers, or through the NVMREG register interface.

The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump rated to operate over the operating voltage range of the device.

NVM can be protected in two ways; by either code protection or write protection.

Code protection (\overline{CP} bit in Configuration Word 5) disables access, reading and writing, to the PFM via external device programmers. Code protection does not affect the self-write and erase functionality. Code protection can only be Reset by a device programmer performing a Bulk Erase to the device, clearing all nonvolatile memory, Configuration bits, and User IDs.

Write protection prohibits self-write and erase to a portion or all of the PFM, as defined by the WRT<1:0> bits of Configuration Word 4. Write protection does not affect a device programmer's ability to read, write, or erase the device.

13.1 Program Flash Memory (PFM)

PFM consists of an array of 14-bit words as user memory, with additional words for User ID information, Configuration words, and interrupt vectors. PFM provides storage locations for:

- User program instructions
- User defined data

PFM data can be read and/or written to through:

- CPU instruction fetch (read-only)
- FSR/INDF indirect access (read-only)
(Section 13.2 “FSR and INDF Access”)
- NVMREG access (Section 13.3 “NVMREG Access”)
- In-Circuit Serial Programming™ (ICSP™)

Read operations return a single word of memory. When write and erase operations are done on a row basis, the row size is defined in Table 13-1. PFM will erase to a logic '1' and program to a logic '0'.

TABLE 13-1: FLASH MEMORY ORGANIZATION BY DEVICE

Device	Row Erase (words)	Write Latches (words)	Total Program Flash (words)
PIC16(L)F15313	32	32	2048
PIC16(L)F15323			2048

It is important to understand the PFM memory structure for erase and programming operations. PFM is arranged in rows. A row consists of 32 14-bit program memory words. A row is the minimum size that can be erased by user software.

All or a portion of a row can be programmed. Data to be written into the program memory row is written to 14-bit wide data write latches. These latches are not directly accessible, but may be loaded via sequential writes to the NVMDATH:NVMDATL register pair.

Note: To modify only a portion of a previously programmed row, the contents of the entire row must be read. Then, the new data and retained data can be written into the write latches to reprogram the row of PFM. However, any unprogrammed locations can be written without first erasing the row. In this case, it is not necessary to save and rewrite the other previously programmed locations

13.1.1 PROGRAM MEMORY VOLTAGES

The PFM is readable and writable during normal operation over the full VDD range.

13.1.1.1 Programming Externally

The program memory cell and control logic support write and Bulk Erase operations down to the minimum device operating voltage. Special BOR operation is enabled during Bulk Erase (Section 8.2.4 “BOR is always OFF”).

13.1.1.2 Self-programming

The program memory cell and control logic will support write and row erase operations across the entire VDD range. Bulk Erase is not available when self-programming.

EXAMPLE 13-1: PROGRAM MEMORY READ

```
* This code block will read 1 word of program
* memory at the memory address:
  PROG_ADDR_HI : PROG_ADDR_LO
* data will be returned in the variables:
* PROG_DATA_HI, PROG_DATA_LO

BANKSEL  NVMADRL          ; Select Bank for NVMCON registers
MOVLW   PROG_ADDR_LO     ;
MOVWF   NVMADRL          ; Store LSB of address
MOVLW   PROG_ADDR_HI     ;
MOVWF   NVMADRH          ; Store MSB of address

BCF     NVMCON1,NVMREGS  ; Do not select Configuration Space
BSF     NVMCON1,RD       ; Initiate read

MOVF    NVMDATL,W        ; Get LSB of word
MOVWF   PROG_DATA_LO    ; Store in user location
MOVF    NVMDATH,W        ; Get MSB of word
MOVWF   PROG_DATA_HI    ; Store in user location
```

TABLE 15-1: PPS INPUT SIGNAL ROUTING OPTIONS (PIC16(L)F15313)

INPUT SIGNAL NAME	Input Register Name	Default Location at POR	Reset Value (xxxPPS<4:0>)	Remappable to Pins of PORTx
				PIC16(L)F15313
				PORTA
INT	INTPPS	RA2	00010	•
T0CKI	T0CKIPPS	RA2	00010	•
T1CKI	T1CKIPSS	RA5	00101	•
T1G	T1GPPS	RA4	00100	•
T2IN	T2INPPS	RA5	00101	•
CCP1	CCP1PPS	RA5	00101	•
CCP2	CCP2PPS	RA5	00101	•
CWG1IN	CWG1INPPS	RA2	00010	•
CLCIN0	CLCIN0PPS	RA3	00011	•
CLCIN1	CLCIN1PPS	RA5	00101	•
CLCIN2	CLCIN2PPS	RA1	00001	•
CLCIN3	CLCIN3PPS	RA0	00000	•
ADACT	ADACTPPS	RA5	00101	•
SCK1/SCL1	SSP1CLKPPS	RA1	00001	•
SDI1/SDA1	SSP1DATPPS	RA2	00010	•
SS1	SSP1SS1PPS	RA3	00011	•
RX1/DT1	RX1PPS	RA1	00001	•
CK1	TX1PPS	RA0	00000	•

23.0 COMPARATOR MODULE

Comparators are used to interface analog circuits to a digital circuit by comparing two analog voltages and providing a digital indication of their relative magnitudes. Comparators are very useful mixed signal building blocks because they provide analog functionality independent of program execution. The analog comparator module includes the following features:

- Programmable input selection
- Selectable voltage reference
- Programmable output polarity
- Rising/falling output edge interrupts
- CWG1 Auto-shutdown source

23.1 Comparator Overview

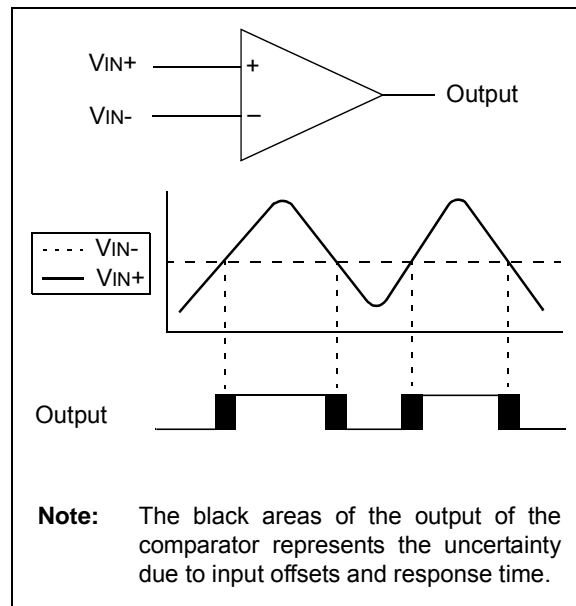
A single comparator is shown in Figure 23-1 along with the relationship between the analog input levels and the digital output. When the analog voltage at V_{IN+} is less than the analog voltage at V_{IN-} , the output of the comparator is a digital low level. When the analog voltage at V_{IN+} is greater than the analog voltage at V_{IN-} , the output of the comparator is a digital high level.

The comparators available are shown in Table 23-1.

TABLE 23-1: AVAILABLE COMPARATORS

Device	C1	C2
PIC16(L)F15313	•	
PIC16(L)F15323	•	•

FIGURE 23-1: SINGLE COMPARATOR



REGISTER 23-2: CMxCON1: COMPARATOR Cx CONTROL REGISTER 1

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	INTP	INTN
bit 7						bit 0	

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **INTP:** Comparator Interrupt on Positive-Going Edge Enable bits

1 = The CxIF interrupt flag will be set upon a positive-going edge of the CxOUT bit

0 = No interrupt flag will be set on a positive-going edge of the CxOUT bit

bit 0 **INTN:** Comparator Interrupt on Negative-Going Edge Enable bits

1 = The CxIF interrupt flag will be set upon a negative-going edge of the CxOUT bit

0 = No interrupt flag will be set on a negative-going edge of the CxOUT bit

26.5.2 TIMER GATE SOURCE SELECTION

One of the several different external or internal signal sources may be chosen to gate the timer and allow the timer to increment. The gate input signal source can be selected based on the T1GATE register setting. See the T1GATE register (Register 26-4) description for a complete list of the available gate sources. The polarity for each available source is also selectable. Polarity selection is controlled by the GPOL bit of the T1GCON register.

26.5.2.1 T1G Pin Gate Operation

The T1G pin is one source for the timer gate control. It can be used to supply an external source to the time gate circuitry.

26.5.2.2 Timer0 Overflow Gate Operation

When Timer0 overflows, or a period register match condition occurs (in 8-bit mode), a low-to-high pulse will automatically be generated and internally supplied to the Timer1 gate circuitry.

26.5.2.3 Comparator C1 Gate Operation

The output resulting from a Comparator 1 operation can be selected as a source for the timer gate control. The Comparator 1 output can be synchronized to the timer clock or left asynchronous. For more information see **Section 23.4.1 “Comparator Output Synchronization”**.

26.5.2.4 Comparator C2 Gate Operation

The output resulting from a Comparator 2 operation can be selected as a source for the timer gate control. The Comparator 2 output can be synchronized to the timer clock or left asynchronous. For more information see **Section 23.4.1 “Comparator Output Synchronization”**.

26.5.3 TIMER1 GATE TOGGLE MODE

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

The timer gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. See Figure 26-4 for timing details.

Timer1 Gate Toggle mode is enabled by setting the GTM bit of the T1GCON register. When the GTM 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.

26.5.4 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 GSPM bit in the T1GCON register. Next, the GGO/DONE bit in the T1GCON register must be set. The timer will be fully enabled on the next incrementing edge. On the next trailing edge of the pulse, the GGO/DONE bit will automatically be cleared. No other gate events will be allowed to increment the timer until the GGO/DONE bit is once again set in software. See Figure 26-5 for timing details.

If the Single-Pulse Gate mode is disabled by clearing the GSPM bit in the T1GCON register, the GGO/DONE bit should also be cleared.

Enabling the Toggle mode and the Single-Pulse mode simultaneously will permit both sections to work together. This allows the cycle times on the timer gate source to be measured. See Figure 26-6 for timing details.

26.5.5 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 GVAL bit in the T1GCON register. The GVAL bit is valid even when the timer gate is not enabled (GE bit is cleared).

26.5.6 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 GVAL occurs, the TMR1GIF flag bit in the PIR5 register will be set. If the TMR1GIE bit in the PIE5 register is set, then an interrupt will be recognized.

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

30.3 Selectable Input Sources

The CWG generates the output waveforms from the input sources in Table 30-2.

TABLE 30-2: SELECTABLE INPUT SOURCES

Source Peripheral	Signal Name
CWG input PPS pin	CWG1IN_PPS
CCP1	CCP1_out
CCP2	CCP2_out
PWM3	PWM3_out
PWM4	PWM4_out
PWM5	PWM5_out
PWM6	PWM6_out
NCO	NCO1_out
Comparator C1	C1OUT_sync
Comparator C2	C2OUT_sync
CLC1	LC1_out
CLC2	LC2_out
CLC3	LC3_out
CLC4	LC4_out

The input sources are selected using the CWG1ISM register.

30.4 Output Control

30.4.1 POLARITY CONTROL

The polarity of each CWG output can be selected independently. When the output polarity bit is set, the corresponding output is active-high. Clearing the output polarity bit configures the corresponding output as active-low. However, polarity does not affect the override levels. Output polarity is selected with the POLx bits of the CWG1CON1. Auto-shutdown and steering options are unaffected by polarity.

30.9 CWG Steering Mode

In Steering mode ($MODE = 00x$), the CWG allows any combination of the CWG1x pins to be the modulated signal. The same signal can be simultaneously available on multiple pins, or a fixed-value output can be presented.

When the respective STRx bit of CWG1OCON0 is '0', the corresponding pin is held at the level defined. When the respective STRx bit of CWG1OCON0 is '1', the pin is driven by the input data signal. The user can assign the input data signal to one, two, three, or all four output pins.

The POLx bits of the CWG1CON1 register control the signal polarity only when STRx = 1.

The CWG auto-shutdown operation also applies in Steering modes as described in **Section 30.10 "Auto-Shutdown"**. An auto-shutdown event will only affect pins that have STRx = 1.

30.9.1 STEERING SYNCHRONIZATION

Changing the MODE bits allows for two modes of steering, synchronous and asynchronous.

When $MODE = 000$, the steering event is asynchronous and will happen at the end of the instruction that writes to STRx (that is, immediately). In this case, the output signal at the output pin may be an incomplete waveform. This can be useful for immediately removing a signal from the pin.

When $MODE = 001$, the steering update is synchronous and occurs at the beginning of the next rising edge of the input data signal. In this case, steering the output on/off will always produce a complete waveform.

Figure 30-10 and Figure 30-11 illustrate the timing of asynchronous and synchronous steering, respectively.

FIGURE 30-10: EXAMPLE OF ASYNCHRONOUS STEERING EVENT ($MODE<2:0> = 000$)

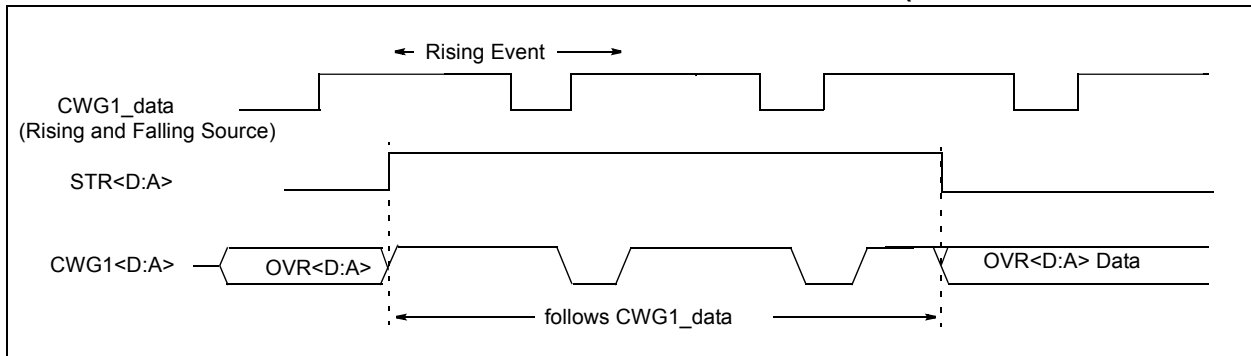
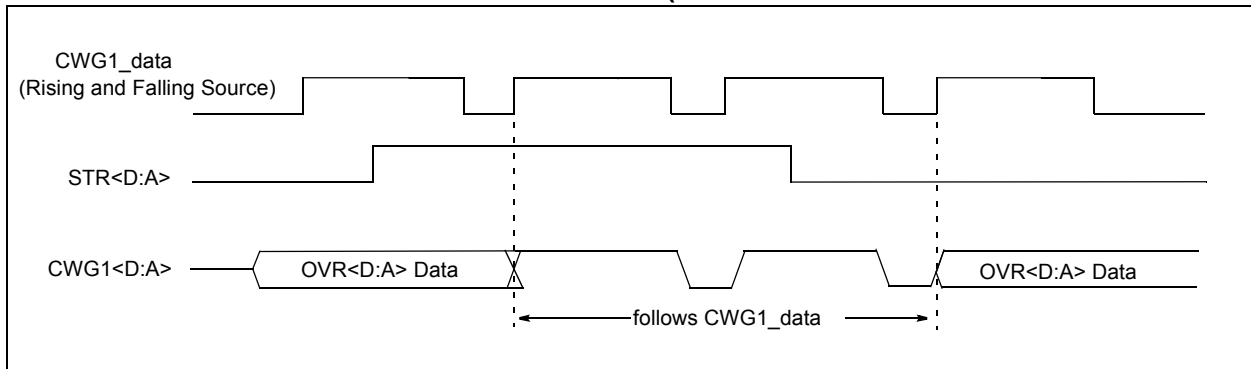


FIGURE 30-11: EXAMPLE OF STEERING EVENT ($MODE<2:0> = 001$)



REGISTER 32-7: SSP1BUF: MSSP1 BUFFER REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SSP1BUF<7:0>							
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **SSP1BUF<7:0>**: MSSP Buffer bits

TABLE 32-3: SUMMARY OF REGISTERS ASSOCIATED WITH MSSP1

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page	
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	121	
PIR1	OSFIF	CSWIF	—	—	—	—	—	ADIF	131	
PIE1	OSFIE	CSWIE	—	—	—	—	—	ADIE	123	
SSP1STAT	SMP	CKE	D \bar{A}	P	S	R \bar{W}	UA	BF	405	
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				406	
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	407	
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	405	
SSP1MSK	SSPMSK<7:0>								409	
SSP1ADD	SSPADD<7:0>								409	
SSP1BUF	SSPBUF<7:0>								410	
SSP1CLKPPS	—	—	SSP1CLKPPS<5:0>						191	
SSP1DATPPS	—	—	SSP1DATPPS<5:0>						191	
SSP1SSPPS	—	—	SSP1SSPPS<5:0>						191	
RxyPPS	—	—	—	RxyPPS<4:0>						192

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by the MSSP module

Note 1: When using designated I²C pins, the associated pin values in INLVLx will be ignored.

LSLF **Logical Left Shift**

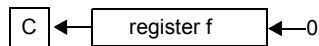
Syntax: `[label] LSLF f {,d}`

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

Operation: $(f<7>) \rightarrow C$
 $(f<6:0>) \rightarrow \text{dest}<7:1>$
 $0 \rightarrow \text{dest}<0>$

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the left through the Carry flag. A '0' is shifted into the LSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.



LSRF **Logical Right Shift**

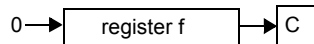
Syntax: `[label] LSRF f {,d}`

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

Operation: $0 \rightarrow \text{dest}<7>$
 $(f<7:1>) \rightarrow \text{dest}<6:0>$,
 $(f<0>) \rightarrow C$,

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the right through the Carry flag. A '0' is shifted into the MSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.



MOVF **Move f**

Syntax: `[label] MOVF f,d`

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

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

Status Affected: Z

Description: The contents of register f is moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example: `MOVF FSR, 0`

After Instruction

`W = value in FSR register`

`Z = 1`

37.0 ELECTRICAL SPECIFICATIONS

37.1 Absolute Maximum Ratings^(†)

Ambient temperature under bias	-40°C to +125°C
Storage temperature	-65°C to +150°C
Voltage on pins with respect to V _{SS}	
on V _{DD} pin	
PIC16F15313/23	-0.3V to +6.5V
PIC16LF15313/23	-0.3V to +4.0V
on MCLR pin	-0.3V to +9.0V
on all other pins	-0.3V to (V _{DD} + 0.3V)
Maximum current	
on V _{SS} pin ⁽¹⁾	
-40°C ≤ T _A ≤ +85°C	250 mA
85°C < T _A ≤ +125°C	85 mA
on V _{DD} pin for 28-Pin devices ⁽¹⁾	
-40°C ≤ T _A ≤ +85°C	250 mA
85°C < T _A ≤ +125°C	85 mA
on V _{DD} pin for 40-Pin devices ⁽¹⁾	
-40°C ≤ T _A ≤ +85°C	350 mA
85°C < T _A ≤ +125°C	120 mA
on any standard I/O pin	±50 mA
Clamp current, I _K (V _{PIN} < 0 or V _{PIN} > V _{DD})	±20 mA
Total power dissipation ⁽²⁾	800 mW

Note 1: Maximum current rating requires even load distribution across I/O pins. Maximum current rating may be limited by the device package power dissipation characterizations, see Table 37-6 to calculate device specifications.

2: Power dissipation is calculated as follows:

$$P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OI} \times I_{OL})$$

† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure above maximum rating conditions for extended periods may affect device reliability.

TABLE 37-8: INTERNAL OSCILLATOR PARAMETERS⁽¹⁾

Standard Operating Conditions (unless otherwise stated)							
Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
OS50	FHFOSC	Precision Calibrated HFINTOSC Frequency	—	4 8 12 16 32	—	MHz	(Note 2)
OS51	FHFOSCLP	Low-Power Optimized HFINTOSC Frequency	—	1 2	—	MHz MHz	
OS52	FMFOSC	Internal Calibrated MFINTOSC Frequency	—	500	—	kHz	
OS53	LFOSC	Internal LFINTOSC Frequency	—	31	—	kHz	
OS54	THFOSCST	HFINTOSC Wake-up from Sleep Start-up Time	—	11 50	20 —	μs μs	VREGPM = 0 VREGPM = 1
OS56	TLFOSCST	LFINTOSC Wake-up from Sleep Start-up Time	—	0.2	—	ms	

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

- Note 1:** To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.
- 2:** See Figure 37-6: Precision Calibrated HFINTOSC Frequency Accuracy Over Device VDD and Temperature.

FIGURE 37-6: PRECISION CALIBRATED HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE

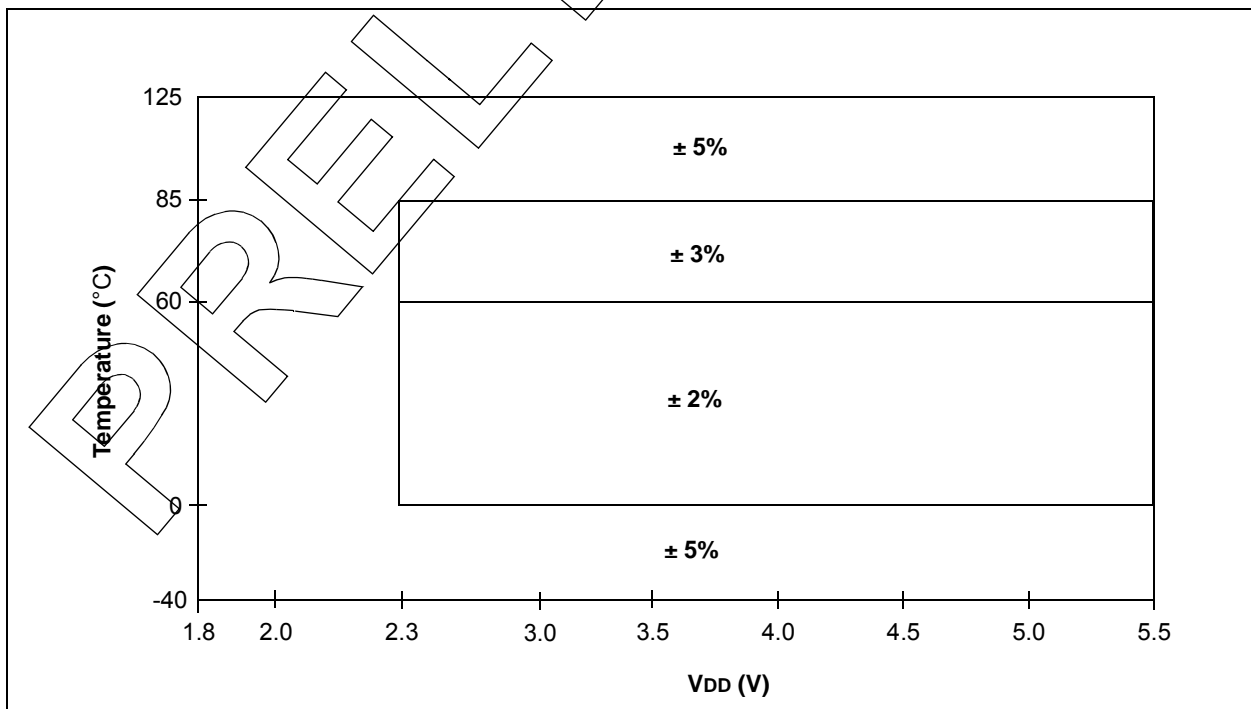


FIGURE 37-7: CLKOUT AND I/O TIMING

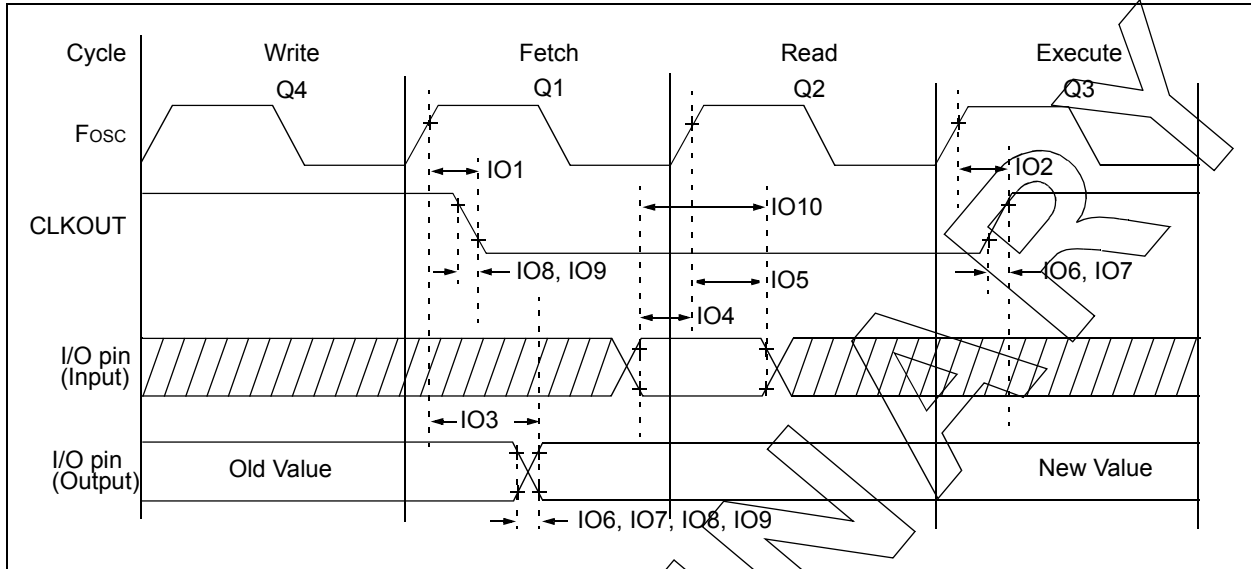


TABLE 37-10: I/O AND CLKOUT TIMING SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
IO1*	$T_{CLKOUTH}$	CLKOUT rising edge delay (rising edge Fosc (Q1 cycle) to falling edge CLKOUT)	—	—	70	ns	
IO2*	$T_{CLKOUTL}$	CLKOUT falling edge delay (rising edge Fosc (Q3 cycle) to rising edge CLKOUT)	—	—	72	ns	
IO3*	T_{IO_VALID}	Port output valid time (rising edge Fosc (Q1 cycle) to port valid)	—	50	70	ns	
IO4*	T_{IO_SETUP}	Port input setup time (Setup time before rising edge Fosc – Q2 cycle)	20	—	—	ns	
IO5*	T_{IO_HOLD}	Port input hold time (Hold time after rising edge Fosc – Q2 cycle)	50	—	—	ns	
IO6*	T_{IOR_SLREN}	Port I/O rise time, slew rate enabled	—	25	—	ns	VDD = 3.0V
IO7*	T_{IOR_SLRDIS}	Port I/O rise time, slew rate disabled	—	5	—	ns	VDD = 3.0V
IO8*	T_{IOF_SLREN}	Port I/O fall time, slew rate enabled	—	25	—	ns	VDD = 3.0V
IO9*	T_{IOF_SLRDIS}	Port I/O fall time, slew rate disabled	—	5	—	ns	VDD = 3.0V
IO10*	T_{INT}	INT pin high or low time to trigger an interrupt	25	—	—	ns	
IO11*	T_{IOC}	Interrupt-on-Change minimum high or low time to trigger interrupt	25	—	—	ns	

*These parameters are characterized but not tested.

FIGURE 37-8: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

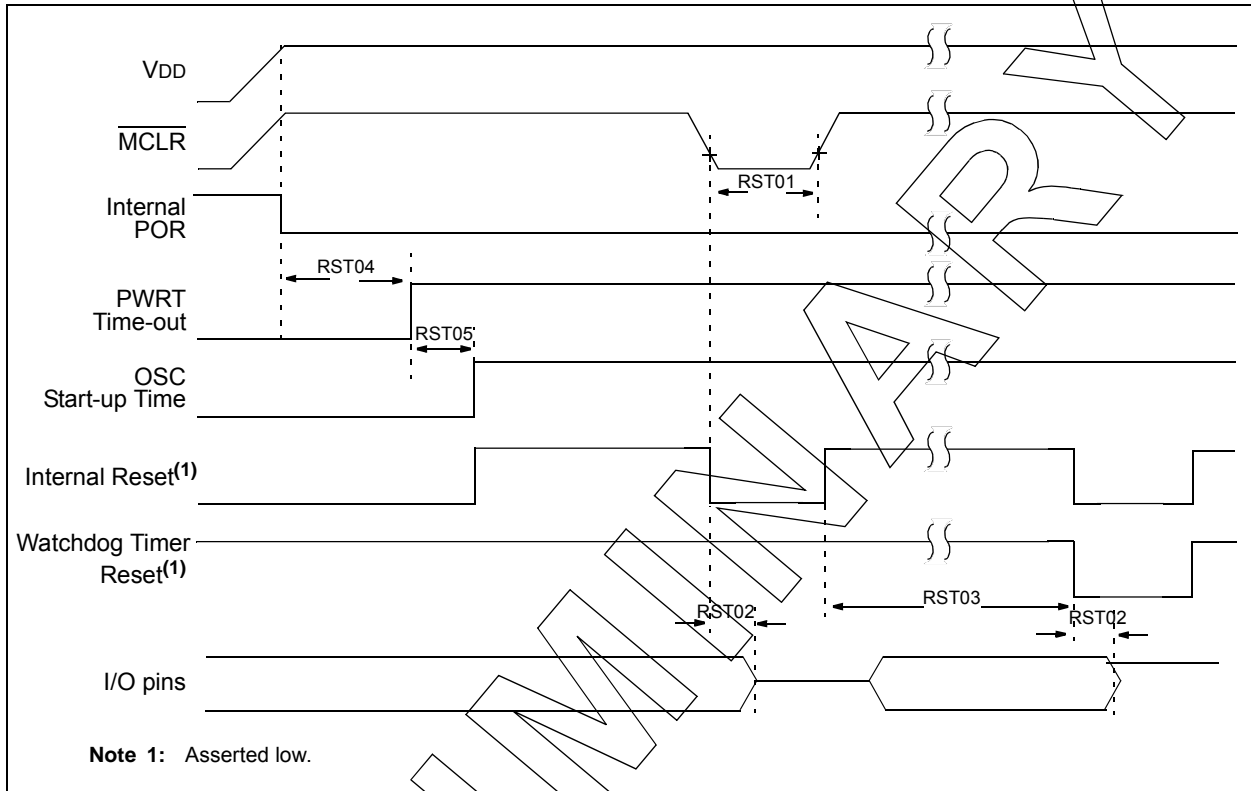
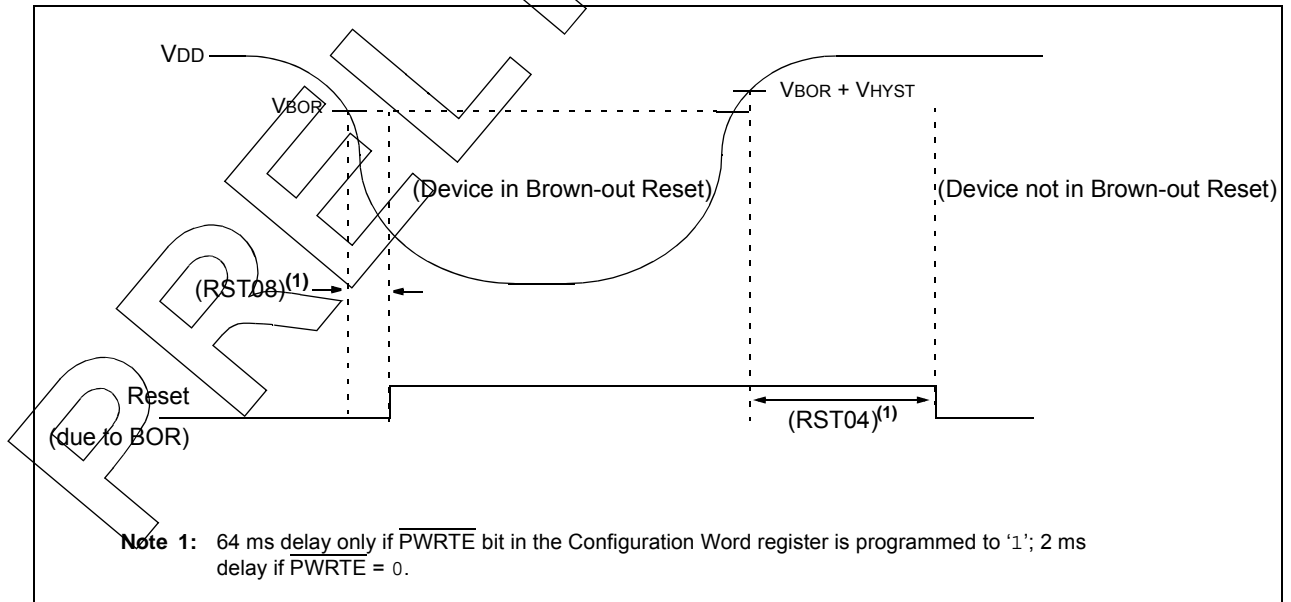


FIGURE 37-9: BROWN-OUT RESET TIMING AND CHARACTERISTICS



39.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

39.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

39.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/librarian features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

39.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

