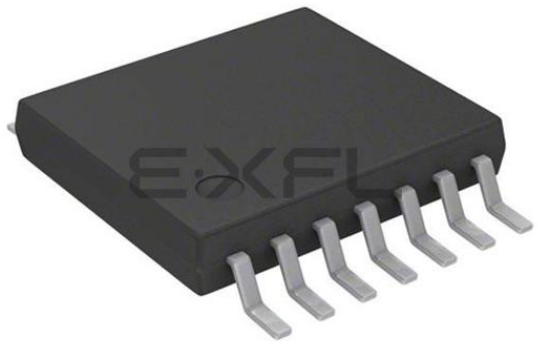


Welcome to E-XFL.COM

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	-
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	11
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5V
Data Converters	A/D 8x10b; D/A 1x9b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	14-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	14-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16hv753-e-st

TABLE 2-2: PIC16F753/HV753 SPECIAL REGISTERS SUMMARY BANK 1

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR	Values on all other Resets
Bank 1											
80h	INDF	INDF<7:0>								xxxx xxxx	uuuu uuuu
81h	OPTION_REG	$\overline{RAPU}$	INTEDG	T0CS	T0SE	PSA	PS<2:0>			1111 1111	1111 1111
82h	PCL	PCL<7:0>								0000 0000	0000 0000
83h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
84h	FSR	FSR								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
86h	—	Unimplemented								—	—
87h	TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
88h	IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	--00 0000	--00 0000
89h	IOCCP	—	—	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	--00 0000	--00 0000
8Ah	PCLATH	—	—	—	PCLATH<4:0>					---0 0000	---0 0000
8Bh	INTCON	GIE	PEIE	T0IE	INTE	IOCIE	T0IF	INTF	IOCIF	0000 0000	0000 0000
8Ch	PIE1	TMR1GIE	ADIE	—	—	HLTMR2IE	HLTMR1IE	TMR2IE	TMR1IE	00-- 0000	00-- 0000
8Dh	PIE2	—	—	C2IE	C1IE	—	COG1IE	—	CCP1IE	--00 -0-0	--00 -0-0
8Eh	—	Unimplemented								—	—
8Fh	OSCCON	—	—	IRCF<1:0>		—	HTS	LTS	—	--01 -00-	--uu -uu-
90h	FVR1CON0	FVREN	FVRRDY	FVROE	FVRBUFSS1	FVRBUFSS0	—	—	FVRBUFEN	0000 0--0	0000 0--0
91h	DAC1CON0	DACEN	DACFM	DACOE	—	DACPSS1	DACPSS0	—	—	000- 00--	000- 00--
92h	DAC1REFL	Least Significant bit of the left shifted result or eight bits of the right shifted DAC setting								0000 0000	0000 0000
93h	DAC1REFH	Most Significant eight bits of the left shifted DAC setting or first bit of the right shifted result								0000 0000	0000 0000
94h	—	Unimplemented								—	—
95h	—	Unimplemented								—	—
96h	OPA1CON	OPA1EN	—	—	OPA1UGM	OPA1NCH<1:0>		OPA1PCH<1:0>		0--0 0000	0--0 0000
97h	—	Unimplemented								—	—
98h	—	Unimplemented								—	—
99h	—	Unimplemented								—	—
9Ah	—	Unimplemented								—	—
9Bh	CM2CON0	C2ON	C2OUT	C2OE	C2POL	C2ZLF	C2SP	C2HYS	C2SYNC	0000 0100	0000 0100
9Ch	CM2CON1	C2INTP	C2INTN	C2PCH<2:0>			C2NCH<2:0>			0000 0000	0000 0000
9Dh	CM1CON0	C1ON	C1OUT	C1OE	C1POL	C1ZLF	C1SP	C1HYS	C1SYNC	0000 0100	0000 0100
9Eh	CM1CON1	C1INTP	C1INTN	C1PCH<2:0>			C1NCH<2:0>			0000 0000	0000 0000
9Fh	CMOUT	—	—	—	—	—	—	MCOUT2	MCOUT1	---- --00	---- --00

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition shaded = unimplemented.

2.3.4 PIE1 REGISTER

The PIE1 register contains the Peripheral Interrupt Enable bits, as shown in Register 2-4.

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

REGISTER 2-4: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
TMR1GIE	ADIE	—	—	HLTMR2IE	HLTMR1IE	TMR2IE	TMR1IE
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 **TMR1GIE:** ADC Interrupt Enable bit
1 = Enables the TMR1 gate interrupt
0 = Disables the TMR1 gate interrupt

bit 6 **ADIE:** ADC Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

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

bit 3 **HLTMR2IE:** HLT2 Interrupt Enable bit
1 = Enables the HLT2 interrupt
0 = Disables the HLT2 interrupt

bit 2 **HLTMR1IE:** HLT1 Interrupt Enable bit
1 = Enables the HLT1 interrupt
0 = Disables the HLT1 interrupt

bit 1 **TMR2IE:** Timer2 Interrupt Enable bit
1 = Enables the Timer2 interrupt
0 = Disables the Timer2 interrupt

bit 0 **TMR1IE:** Timer1 Interrupt Enable bit
1 = Enables the Timer1 interrupt
0 = Disables the Timer1 interrupt

3.5 Writing the Flash Program Memory

A word of the Flash program memory may only be written to if the word is in an unprotected segment of memory.

Flash program memory must be written in four-word blocks. See Figure 3-2 and Figure 3-3 for more details. A block consists of four words with sequential addresses, with a lower boundary defined by an address, where PMADRL<1:0> = 00. All block writes to program memory are done as 16-word erase by four-word write operations. The write operation is edge-aligned and cannot occur across boundaries.

To write program data, it must first be loaded into the buffer registers (see Figure 3-2). This is accomplished by first writing the destination address to PMADRL and PMADRH and then writing the data to PMDATL and PMDATH. After the address and data have been set up, then the following sequence of events must be executed:

1. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
2. Set the WR control bit of the PMCON1 register.

All four buffer register locations should be written to with correct data. If less than four words are being written to in the block of four words, then a read from the program memory location(s) not being written to must be performed. This takes the data from the program location(s) not being written and loads it into the PMDATL and PMDATH registers. Then the sequence of events to transfer data to the buffer registers must be executed.

To transfer data from the buffer registers to the program memory, the PMADRL and PMADRH must point to the last location in the four-word block (PMADRL<1:0> = 11). Then the following sequence of events must be executed:

1. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
2. Set control bit WR of the PMCON1 register to begin the write operation.

The user must follow the same specific sequence to initiate the write for each word in the program block, writing each program word in sequence (000, 001, 010, 011). When the write is performed on the last word (PMADRL<1:0> = 11), a block of sixteen words is automatically erased and the content of the four-word buffer registers are written into the program memory.

After the "BSF PMCON1,WR" instruction, the processor requires two cycles to set up the erase/write operation. The user must place two NOP instructions after the WR bit is set. Since data is being written to buffer registers, the writing of the first three words of the block appears to occur immediately. The processor will halt internal operations for the typical 4 ms, only during the cycle in which the erase takes place (i.e., the last word of the sixteen-word block erase). This is not Sleep mode as the clocks and peripherals will continue to run. After the four-word write cycle, the processor will resume operation with the third instruction after the PMCON1 write instruction. The above sequence must be repeated for the higher 12 words.

3.6 Protection Against Spurious Write

There are conditions when the device should not write to the program memory. To protect against spurious writes, various mechanisms have been built in. On power-up, WREN is cleared. Also, the Power-up Timer (64 ms duration) prevents program memory writes.

The write initiate sequence and the WREN bit help prevent an accidental write during brown-out, power glitch or software malfunction.

3.7 Operation During Code-Protect

When the device is code-protected, the CPU is able to read and write unscrambled data to the program memory.

3.8 Operation During Write Protect

When the program memory is write-protected, the CPU can read and execute from the program memory. The portions of program memory that are write-protected can be modified by the CPU using the PMCON registers, but the protected program memory cannot be modified using ICSP mode.

4.3 System Clock Output

The CLKOUT pin is available for general purpose I/O or system clock output. The $\overline{\text{CLKOUTEN}}$ bit of the Configuration Word controls the function of the CLKOUT pin.

When the $\overline{\text{CLKOUTEN}}$ bit is cleared, the CLKOUT pin is driven by the selected internal oscillator frequency divided by 4. The corresponding I/O pin always reads '0' in this configuration.

The CLKOUT signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements.

When the $\overline{\text{CLKOUTEN}}$ bit is set, the system clock out function is disabled and the CLKOUT pin is available for general purpose I/O.

4.4 Oscillator Delay upon Wake-Up, Power-Up, and Base Frequency Change

In applications where the OSCTUNE register is used to shift the HFINTOSC frequency, the application should not expect the frequency to stabilize immediately. In this case, the frequency may shift gradually toward the new value. The time for this frequency shift is less than eight cycles of the base frequency.

A short delay is invoked upon power-up and when waking from sleep to allow the memory bias circuitry to stabilize. Table 4-1 shows examples where the oscillator delay is invoked.

TABLE 4-1: OSCILLATOR DELAY EXAMPLES

Switch From	Switch To	Frequency	Oscillator Delay
Sleep/POR	INTOSC	31 kHz to 8 MHz	10 μs internal delay to allow memory bias to stabilize.
Sleep/POR	EC	DC – 20 MHz	

FIGURE 7-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE

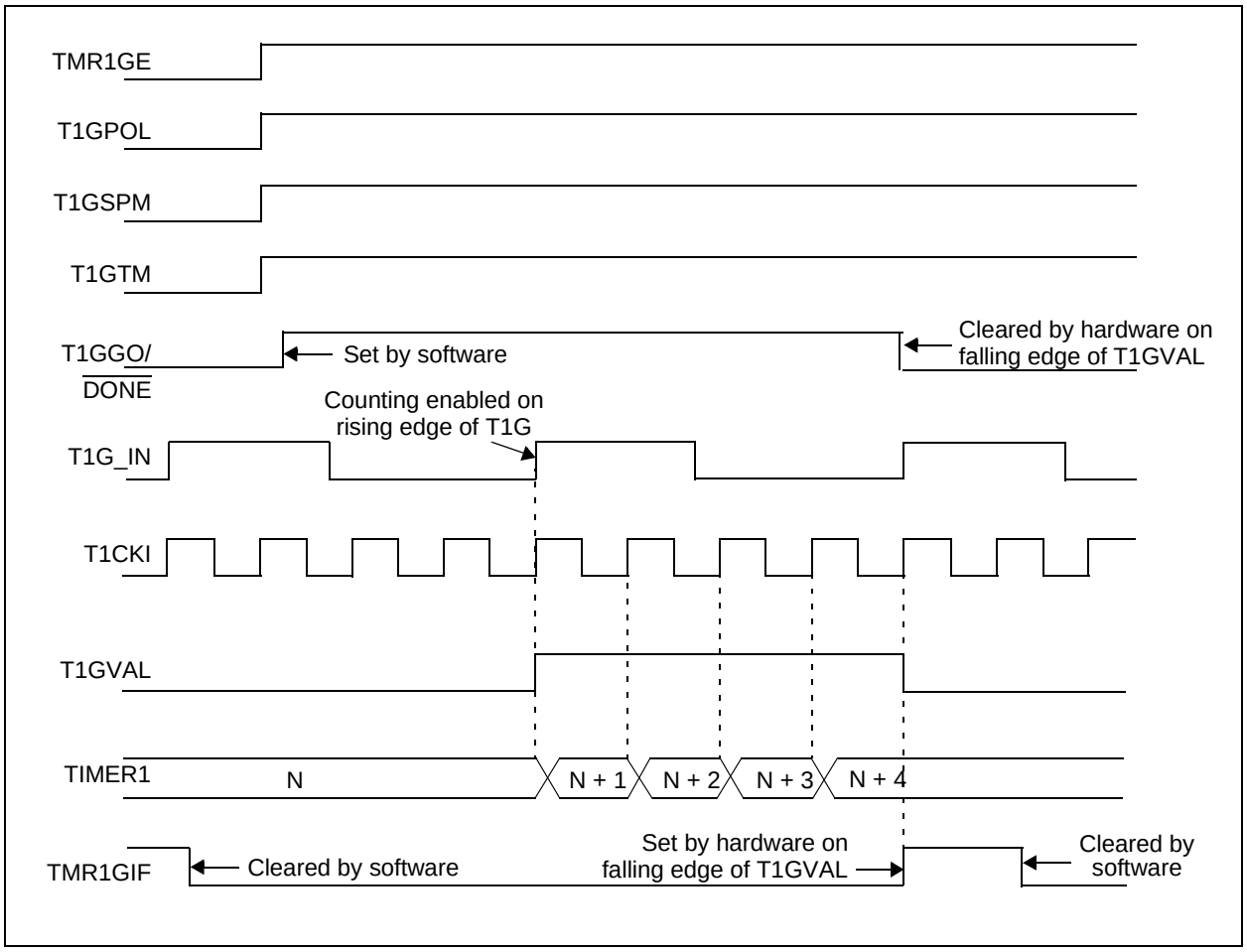
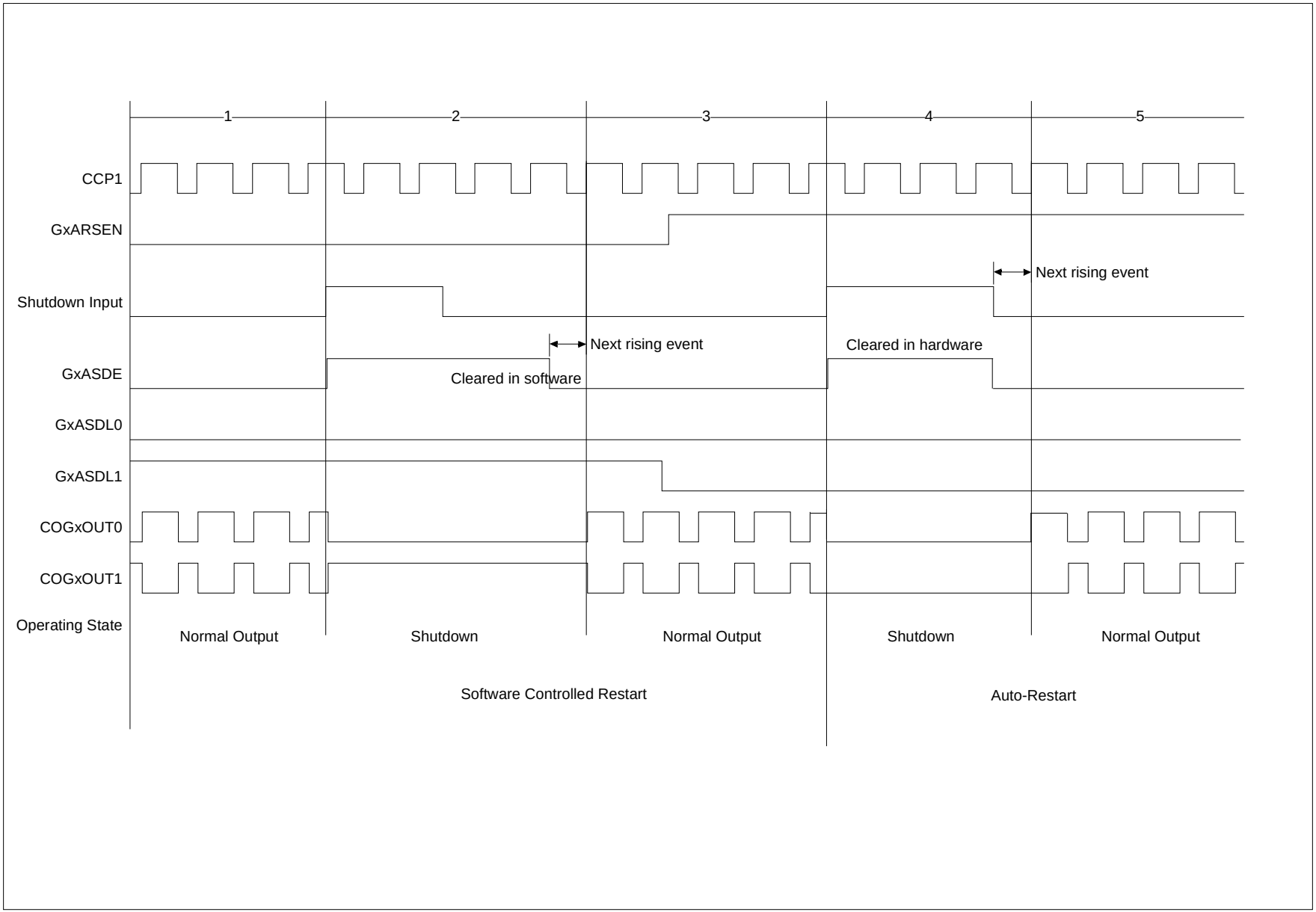


FIGURE 11-8: AUTO-SHUTDOWN WAVEFORM – CCP1 AS RISING AND FALLING EVENT INPUT SOURCE



11.9 Buffer Updates

Changes to the phase, dead-band, and blanking count registers need to occur simultaneously during COG operation to avoid unintended operation that may occur as a result of delays between each register write. This is accomplished with the GxLD bit of the COGxCON0 register and double buffering of the phase, blanking, and dead-band count registers.

Before the COG module is enabled, writing the count registers loads the count buffers without need of the GxLD bit. However, when the COG is enabled, the count buffers updates are suspended after writing the count registers until after the GxLD bit is set. When the GxLD bit is set, the phase, dead-band, and blanking register values are transferred to the corresponding buffers synchronous with COG operation. The GxLD bit is cleared by hardware when the transfer is complete.

11.10 Alternate Pin Selection

The COGxOUT0, COGxOUT1 and COGxFLT functions can be directed to alternate pins with control bits of the APFCON register. Refer to Register 5-1.

Note:	The default COG outputs have high drive strength capability, whereas the alternate outputs do not.

11.11 Operation During Sleep

The COG continues to operate in Sleep provided that the COG_clock, rising event, and falling event sources remain active.

The HFINTSOC remains active during Sleep when the COG is enabled and the HFINTOSC is selected as the COG_clock source.

11.12 Configuring the COG

The following steps illustrate how to properly configure the COG to ensure a synchronous start with the rising event input:

1. Configure the desired COGxFLT input, COGxOUT0 and COGxOUT1 pins with the corresponding bits in the APFCON register.
2. Clear all ANSELA register bits associated with pins that are used for COG functions.
3. Ensure that the TRIS control bits corresponding to COGxOUT0 and COGxOUT1 are set so that both are configured as inputs. These will be set as outputs later.
4. Clear the GxEN bit, if not already cleared.
5. Set desired dead-band times with the COGxDBR and COGxDBF registers.
6. Set desired blanking times with the COGxBKR and COGxBKF registers.
7. Set desired phase delay with the COGxPHR and COGxPHF registers.
8. Select the desired shutdown sources with the COGxASD1 register.
9. Set up the following controls in COGxASD0 auto-shutdown register:
   - Select both output overrides to the desired levels (this is necessary, even if not using auto-shutdown because start-up will be from a shutdown state).
   - Set the GxASDE bit and clear the GxARSEN bit.
10. Select the desired rising and falling event sources with the COGxRIS and COGxFIS registers.
11. Select the desired rising and falling event modes with the COGxRSIM and COGxFSIM registers.
12. Configure the following controls in the COGxCON1 register:
    - Select the desired clock source
    - Select the desired dead-band timing sources
13. Configure the following controls in the COGxCON0 register:
    - Select the desired output polarities.
    - Set the output enables of the outputs to be used.
14. Set the GxEN bit.
15. Clear TRIS control bits corresponding to COGxOUT0 and COGxOUT1 to be used, thereby configuring those pins as outputs.
16. If auto-restart is to be used, set the GxARSEN bit and the GxASDE will be cleared automatically. Otherwise, clear the GxASDE bit to start the COG.

11.13 Register Definitions: COG Control

REGISTER 11-1: COGxCON0: COG CONTROL REGISTER 0

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0
GxEN	GxOE1	GxOE0	GxPOL1	GxPOL0	GxLD	—	GxMD
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	q = Value depends on condition

bit 7 **GxEN:** COGx Enable bit
1 = Module is enabled
0 = Module is disabled

bit 6 **GxOE1:** COGxOUT1 Output Enable bit
1 = COGxOUT1 is available on associated I/O pin
0 = COGxOUT1 is not available on associated I/O pin

bit 5 **GxOE0:** COGxOUT0 Output Enable bit
1 = COGxOUT0 is available on associated I/O pin
0 = COGxOUT0 is not available on associated I/O pin

bit 4 **GxPOL1:** COGxOUT1 Output Polarity bit
1 = Output is inverted polarity
0 = Output is normal polarity

bit 3 **GxPOL0:** COGxOUT0 Output Polarity bit
1 = Output is inverted polarity
0 = Output is normal polarity

bit 2 **GxLD:** COGx Load Buffers bit
1 = Phase, blanking, and dead-band buffers to be loaded with register values on next input events
0 = Register to buffer transfer is complete

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **GxMD:** COGx Mode bit
1 = COG outputs operate in Push-Pull mode
0 = COG outputs operate in Synchronous mode

REGISTER 11-5: COGxFIS: COG FALLING EVENT INPUT SELECTION REGISTER

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	GxFIHLT2	GxFIHLT1	GxFIT2M	GxFIFLT	GxFICCP1	GxFIC2	GxFIC1
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	q = Value depends on condition

bit 7 **Unimplemented:** Read as ‘0’

bit 6 **GxFIHLT2:** COGx Falling Event Input Source 6 Enable bit
1 = HLTimer2 output is enabled as a falling event input
0 = HLTimer2 has no effect on the falling event

bit 5 **GxFIHLT1:** COGx Falling Event Input Source 5 Enable bit
1 = HLTimer1 output is enabled as a falling event input
0 = HLTimer1 has no effect on the falling event

bit 4 **GxFIT2M:** COGx Falling Event Input Source 4 Enable bit
1 = Timer2 match with PR2 is enabled as a falling event input
0 = Timer2 match with PR2 has no effect on the falling event

bit 3 **GxFIFLT:** COGx Falling Event Input Source 3 Enable bit
1 = COGxFLT pin is enabled as a falling event input
0 = COGxFLT pin has no effect on the falling event

bit 2 **GxFICCP1:** COGx Falling Event Input Source 2 Enable bit
1 = CCP1 output is enabled as a falling event input
0 = CCP1 has no effect on the falling event

bit 1 **GxFIC2:** COGx Falling Event Input Source 1 Enable bit
1 = Comparator 2 output is enabled as a falling event input
0 = Comparator 2 output has no effect on the falling event

bit 0 **GxFIC1:** COGx Falling Event Input Source 0 Enable bit
1 = Comparator 1 output is enabled as a falling event input
0 = Comparator 1 output has no effect on the falling event

REGISTER 11-6: COGxFSIM: COG FALLING EVENT SOURCE INPUT MODE REGISTER

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	GxFMHLT2	GxFMHLT1	GxFMT2M	GxFMFLT	GxFMCCP1	GxFMC2	GxFMC1
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	q = Value depends on condition

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

bit 6 **GxFMHLT2:** COGx Falling Event Input Source 6 Mode bit(1)
GxFIHLT2 = 1:
1 = HLTimer2 low-to-high transition will cause a falling event after falling event phase delay
0 = HLTimer2 high level will cause an immediate falling event
GxFIHLT2 = 0:
HLTimer2 has no effect on falling event

bit 5 **GxFMHLT1:** COGx Falling Event Input Source 5 Mode bit(1)
GxFIHLT1 = 1:
1 = HLTimer1 low-to-high transition will cause a falling event after falling event phase delay
0 = HLTimer1 high level will cause an immediate falling event
GxFIHLT1 = 0:
HLTimer1 has no effect on falling event

bit 4 **GxFMT2M:** COGx Falling Event Input Source 4 Mode bit(1)
GxFIT2M = 1:
1 = Timer2 match with PR2 low-to-high transition will cause a falling event after falling event phase delay
0 = Timer2 match with PR2 high level will cause an immediate falling event
GxFIT2M = 0:
Timer2 match with PR2 has no effect on falling event

bit 3 **GxFMFLT:** COGx Falling Event Input Source 3 Mode bit
GxFIFLT = 1:
1 = COGxFLT pin low-to-high transition will cause a falling event after falling event phase delay
0 = COGxFLT pin high level will cause an immediate falling event
GxFIFLT = 0:
COGxFLT pin has no effect on falling event

bit 2 **GxFMCCP1:** COGx Falling Event Input Source 2 Mode bit
GxFICCP1 = 1:
1 = CCP1 low-to-high transition will cause a falling event after falling event phase delay
0 = CCP1 high level will cause an immediate falling event
GxFICCP1 = 0:
CCP1 has no effect on falling event

bit 1 **GxFMC2:** COGx Falling Event Input Source 1 Mode bit
GxFIC2 = 1:
1 = Comparator 2 low-to-high transition will cause a falling event after falling event phase delay
0 = Comparator 2 high level will cause an immediate falling event
GxFIC2 = 0:
Comparator 2 has no effect on falling event

bit 0 **GxFMC1:** COGx Falling Event Input Source 0 Mode bit
GxFIC1 = 1:
1 = Comparator 1 low-to-high transition will cause a falling event after falling event phase delay
0 = Comparator 1 high level will cause an immediate falling event
GxFIC1 = 0:
Comparator 1 has no effect on falling event

Note 1: These sources are pulses and therefore the only benefit of Edge mode over Level mode is that they can be delayed by falling event phase delay.

REGISTER 11-13: COGxPHR: COG RISING EDGE PHASE DELAY COUNT REGISTER

U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	GxPHR<3:0>			
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	q = Value depends on condition

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **GxPHR<3:0>:** Rising Edge Phase Delay Count Value bits
= Number of COGx clock periods to delay rising edge event

REGISTER 11-14: COGxPHF: COG FALLING EDGE PHASE DELAY COUNT REGISTER

U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	GxPHF<3:0>			
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	q = Value depends on condition

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **GxPHF<3:0>:** Falling Edge Phase Delay Count Value bits
= Number of COGx clock periods to delay falling edge event

TABLE 12-2: SUMMARY OF ASSOCIATED ADC REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	ADFM	—	CHS<3:0>				GO/$\overline{\text{DONE}}$	ADON	109
ADCON1	—	ADCS<2:0>			—	—	—	ADPREF1	110
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	44
ADRESH(2)	Most Significant eight bits of the left shifted A/D result or two bits of the right shifted result								111*
ADRESL(2)	Least Significant two bits of the left shifted result or eight bits of the right shifted result								109*
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	43
INTCON	GIE	PEIE	T0IE	INTE	IOCIE	T0IF	INTF	IOCIF	17
PIE1	TMR1GIE	ADIE	—	—	HLTMR2IE	HLTMR1IE	TMR2IE	TMR1IE	18
PIR1	TMR1GIF	ADIF	—	—	HLTMR2IF	HLTMR1IF	TMR2IF	TMR1IF	20
TRISA	—	—	TRISA5	TRISA4	TRISA3(1)	TRISA2	TRISA1	TRISA0	43

Legend: x = unknown, u = unchanged, – = unimplemented read as '0'. Shaded cells are not used for ADC module.

* Page provides register information.

Note 1: TRISA3 always reads '1'.

2: Read-only register.

REGISTER 14-2: DACxREFH: DAC REFERENCE HIGH REGISTER (DACxFM = 0)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
DACR<8:1>							
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 **DACR<8:1>**: DAC Reference Selection bits
DACxOUT = (DACR<8:0> x (Vdac_ref)/512)

REGISTER 14-3: DACxREFL: DAC REFERENCE LOW REGISTER (DACxFM = 0)

R/W-0/0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
DACR0	—	—	—	—	—	—	—
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 **DACR0**: DAC Reference Selection bits
DACxOUT = (DACR<8:0> x (Vdac_ref)/512)

bit 6-0 **Unimplemented**: Read as ‘0’

15.4.1 COMPARATOR OUTPUT SYNCHRONIZATION

The output from either comparator, C1 or C2, can be synchronized with Timer1 by setting the CxSYNC bit of the CMxCON0 register.

Once enabled, the comparator output is latched on the falling edge of the Timer1 source clock. If a prescaler is used with Timer1, the comparator output is latched after the prescaling function. To prevent a race condition, the comparator output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. See the Comparator Block Diagram (Figure 15-2) and the Timer1 Block Diagram (Figure 7-1) for more information.

15.5 Comparator Interrupt

An interrupt can be generated upon a change in the output value of the comparator for each comparator, a rising edge detector and a falling edge detector are present.

When either edge detector is triggered and its associated enable bit is set (CxINTP and/or CxINTN bits of the CMxCON1 register), the Corresponding Interrupt Flag bit (CxIF bit of the PIR2 register) will be set.

To enable the interrupt, you must set the following bits:

- CxON, CxPOL and CxSP bits of the CMxCON0 register
- CxIE bit of the PIE2 register
- CxINTP bit of the CMxCON1 register (for a rising edge detection)
- CxINTN bit of the CMxCON1 register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

The associated interrupt flag bit, CxIF bit of the PIR2 register, must be cleared in software. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

Note:	Although a comparator is disabled, an interrupt can be generated by changing the output polarity with the CxPOL bit of the CMxCON0 register, or by switching the comparator on or off with the CxON bit of the CMxCON0 register.

15.6 Comparator Positive Input Selection

Configuring the CxPCH<1:0> bits of the CMxCON1 register directs an internal voltage reference or an analog pin to the non-inverting input of the comparator:

- CxIN0+ analog pin
- DAC Reference Voltage (DAC_REF)
- FVR Reference Voltage (FVR_REF)
- VSS (Ground)

See **Section 13.0 “Fixed Voltage Reference (FVR)”** for more information on the Fixed Voltage Reference module.

See **Section 14.0 “Digital-to-Analog Converter (DAC) Module”** for more information on the DAC input signal.

Any time the comparator is disabled (CxON = 0), all comparator inputs are disabled.

15.7 Comparator Negative Input Selection

The CxNCH0 bit of the CMxCON0 register selects the analog input pin to the comparator inverting input.

Note:	To use CxIN0+ and CxIN1x- pins as analog input, the appropriate bits must be set in the ANSEL register and the corresponding TRIS bits must also be set to disable the output drivers.

15.8 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in **Section 22.0 “Electrical Specifications”** for more details.

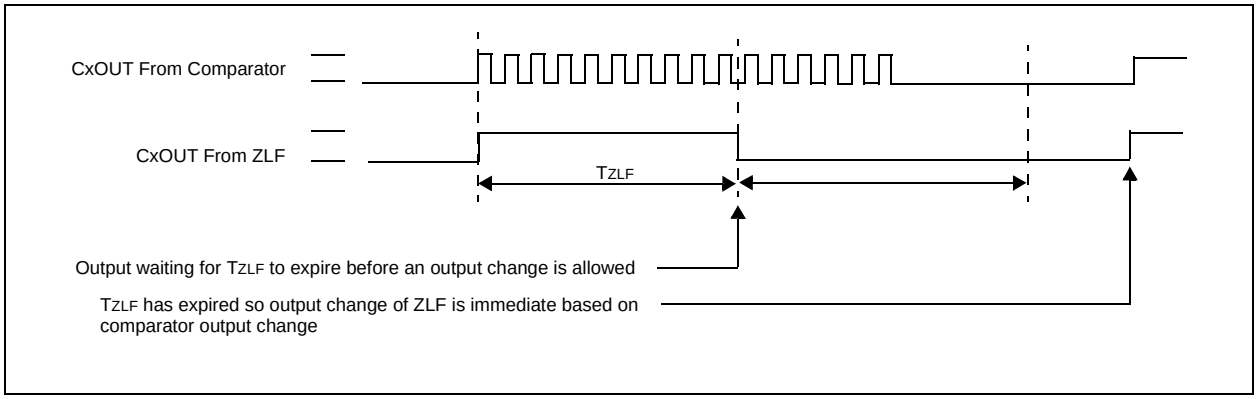
15.9 Interaction with the COG Module

The comparator outputs can be brought to the COG module in order to facilitate auto-shutdown. If auto-restart is also enabled, the comparators can be configured as a closed loop analog feedback to the COG, thereby creating an analog controlled PWM.

15.10 Zero Latency Filter

In high-speed operation, and under proper circuit conditions, it is possible for the comparator output to oscillate. This oscillation can have adverse effects on the hardware and software relying on this signal. Therefore, a digital filter has been added to the comparator output to suppress the comparator output oscillation. Once the comparator output changes, the output is prevented from reversing the change for a nominal time of 20 ns. This allows the comparator output to stabilize without affecting other dependent devices. Refer to Figure 15-3.

FIGURE 15-3: COMPARATOR ZERO LATENCY FILTER OPERATION



REGISTER 15-2: CMxCON1: COMPARATOR Cx CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CxINTP	CxINTN	CxPCH<1:0>			CxNCH<2:0>		
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 **CxINTP:** Comparator Interrupt on Positive Going Edge Enable bit
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 6 **CxINTN:** Comparator Interrupt on Negative Going Edge Enable bit
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

bit 5-3 **CxPCH<1:0>:** Comparator Positive Input Channel Select bits
000 = CxVP connects to CxIN+ pin
001 = CxVP connects to dac_out
010 = CxVP connects to FVR
011 = CxVP connects to Slope Compensator Output
1xx = CxVP connects to AGND

bit 2-0 **CxNCH<2:0>:** Comparator Negative Input Channel Select bits
000 = CxVN connects to CxIN0- pin
001 = CxVN connects to CxIN1- pin
010 = CxVN connects to CxIN2- pin
011 = CxVN connects to CxIN3- pin
1xx = CxVN connects to Slope Compensator Output

REGISTER 15-3: CMOUT: COMPARATOR OUTPUT REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
—	—	—	—	—	—	MCOUT2	MCOUT1
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 **MCOUT2:** Mirror Copy of C2OUT bit

bit 0 **MCOUT1:** Mirror Copy of C1OUT bit

TABLE 19-4: INITIALIZATION CONDITION FOR REGISTERS

Register	Address	Power-on Reset	$\overline{\text{MCLR}}$ Reset WDT Reset Brown-out Reset(1)	Wake-up from Sleep through Interrupt Wake-up from Sleep through WDT Time-out
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h/80h/ 100h/180h	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h/82h/ 102h/182h	0000 0000	0000 0000	PC + 1(3)
STATUS	03h/83h/ 103h/183h	0001 1xxx	000q quuu(4)	uuuq quuu(4)
FSR	04h/84h/ 104h/184h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	--xx xxxx	--uu uuuu	--uu uuuu
IOCAF	08h	--00 0000	--00 0000	--uu uuuu
PCLATH	0Ah/8Ah/ 10Ah/18Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh/8Bh/ 10Bh/18Bh	0000 0000	0000 0000	uuuu uuuu(2)
PIR1	0Ch	00-- -0-0	00-- -0-0	uu-- -u-u(2)
PIR2	0Dh	--00 -0-0	--00 -0-0	--uu -u-u(2)
TMR1L	0Fh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	10h	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	11h	0000 00-0	uuuu uu-u	uuuu uu-u
T1GCON	12h	0000 0x00	0000 0x00	uuuu uuuu
CCPR1L(1)	13h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H(1)	14h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON(1)	15h	--00 0000	--00 0000	--uu uuuu
ADRESL(1)	1Ch	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESH(1)	1Dh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0(1)	1Eh	0000 0000	0000 0000	uuuu uuuu
ADCON1(1)	1Fh	-000 ----	-000 ----	-uuu ----
OPTION_REG	81h/181h	1111 1111	1111 1111	uuuu uuuu
TRISA	85h	--11 1111	--11 1111	--uu uuuu
IOCAP	88h	--00 0000	--00 0000	--uu uuuu
PIE1	8Ch	00-- -000	00-- -000	uu-- -uuu
PIE2	8Dh	--00 -0-0	--00 -0-0	--uu -u-u
OSCCON	8Fh	--01 -00-	--uu -uu-	--uu -uu-
FVRCON	90h	0000 ----	0000 ----	uuuu ----
DACCON0	91h	000- -0--	000- -0--	uuu- -u--
DACCON1	92h	---0 0000	---0 0000	---u uuuu
CM2CON0	9Bh	0000 0100	0000 0100	uuuu uuuu
CM2CON1	9Ch	0000 ---0	0000 ---0	uuuu ---u

Legend: u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition.

Note 1: If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

2: One or more bits in INTCON and/or PIRx will be affected (to cause wake-up).

3: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

4: See Table 19-5 for Reset value for specific condition.

5: If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

TABLE 22-2: SUPPLY CURRENT (I_{DD})(1,2)

PIC16F753			Standard Operating Conditions (unless otherwise stated)					
PIC16HV753								
Param No.	Device Characteristics	Min.	Typ†	Max. 85°C	Max. 125°C	Units	Conditions	
							V_{DD}	Note
	Supply Current (I_{DD})(1, 2)							
D010		—	10	31	31	µA	2.0	FOSC = 31 kHz LFINTOSC mode
		—	15	36	36	µA	3.0	
		—	28	62	62	µA	5.0	
D010		—	75	158	158	µA	2.0	FOSC = 31 kHz LFINTOSC mode
		—	151	192	192	µA	3.0	
		—	201	385	385	µA	4.5	
D011		—	97	140	140	µA	2.0	FOSC = 1 MHz EC Oscillator mode
		—	155	235	235	µA	3.0	
		—	334	475	475	µA	5.0	
D011		—	135	225	225	µA	2.0	FOSC = 1 MHz EC Oscillator mode
		—	260	370	370	µA	3.0	
		—	395	595	595	µA	4.5	
D012		—	172	260	260	µA	2.0	FOSC = 1 MHz HFINTOSC mode
		—	220	360	360	µA	3.0	
		—	398	516	516	µA	5.0	
D012		—	210	338	338	µA	2.0	FOSC = 1 MHz HFINTOSC mode
		—	334	432	432	µA	3.0	
		—	461	680	680	µA	4.5	
D013		—	243	333	333	µA	2.0	FOSC = 4 MHz EC Oscillator mode
		—	365	485	485	µA	3.0	
		—	762	956	956	µA	5.0	
D013		—	261	385	385	µA	2.0	FOSC = 4 MHz EC Oscillator mode
		—	490	620	620	µA	3.0	
		—	710	1045	1045	µA	4.5	

* 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.

Note 1: The test conditions for all I_{DD} measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{SS}; $\overline{MCLR}$ = V_{DD}; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

FIGURE 23-5: IDD TYPICAL, EC OSCILLATOR, HIGH-POWER MODE, PIC16HV753 ONLY

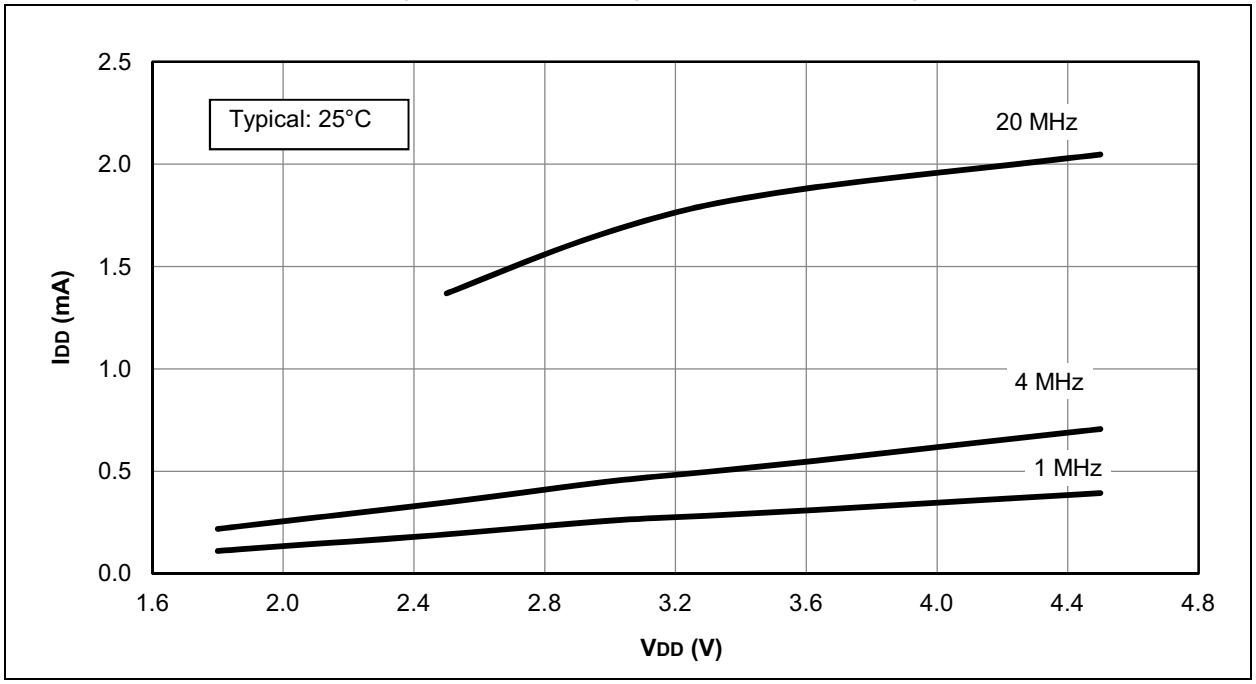


FIGURE 23-6: IDD MAXIMUM, EC OSCILLATOR, HIGH-POWER MODE, PIC16HV753 ONLY

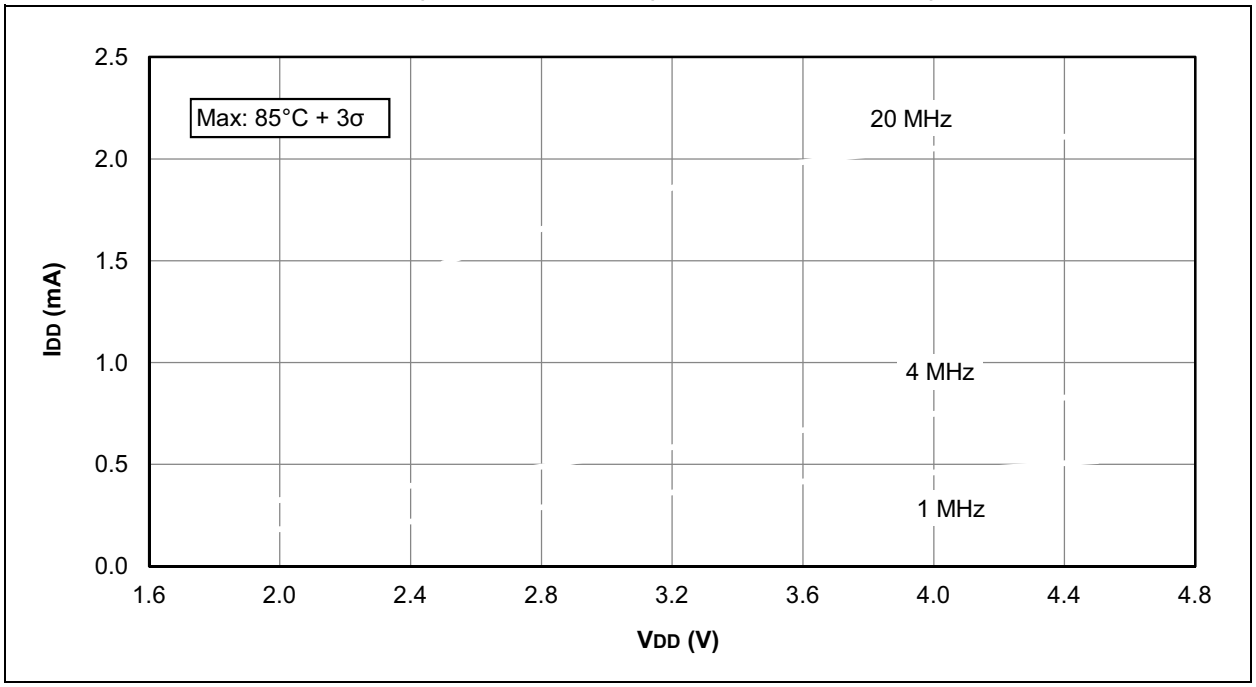


FIGURE 23-7: IDD TYPICAL, EC OSCILLATOR, HIGH-POWER MODE, PIC16F753 ONLY

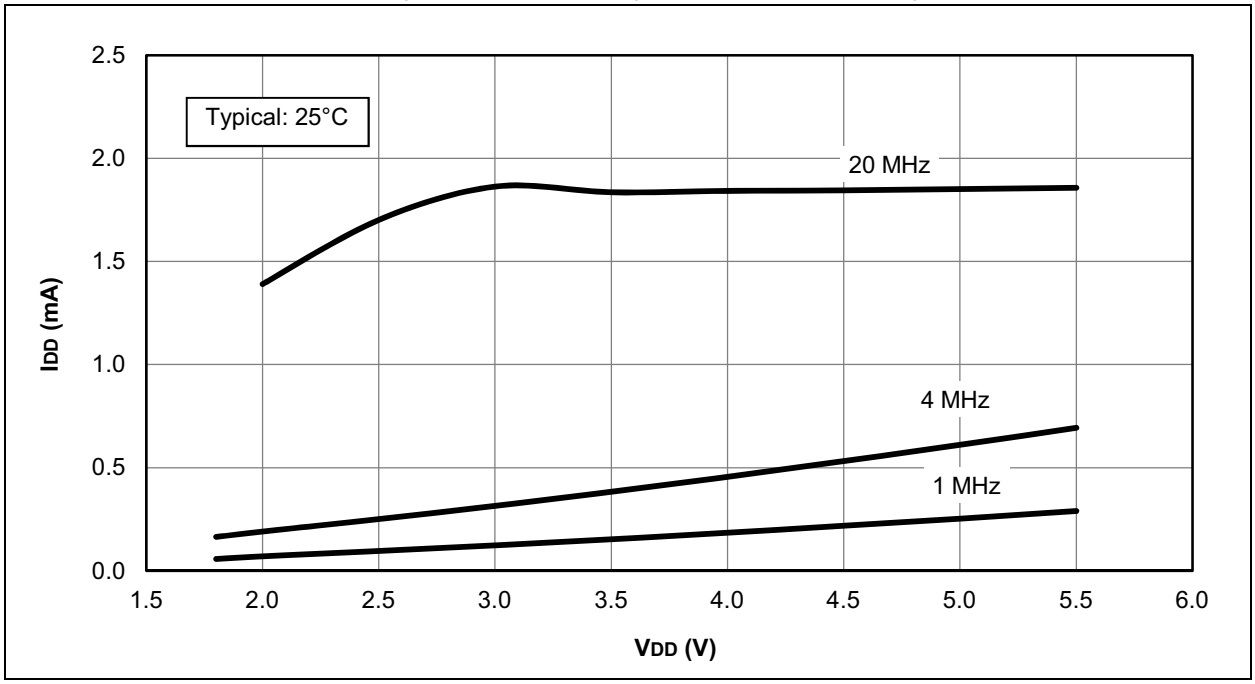


FIGURE 23-8: IDD MAXIMUM, EC OSCILLATOR, HIGH-POWER MODE, PIC16F753 ONLY