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 ~ 5.5V
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-SOIC (0.154", 3.90mm Width)
Supplier Device Package	14-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f753-e-sl

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.

An example of the complete four-word write sequence is shown in Example 3-2. The initial address is loaded into the PMADRH and PMADRL register pair; the four words of data are loaded using indirect addressing.

EXAMPLE 3-2: WRITING TO FLASH PROGRAM MEMORY

```
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ; This write routine assumes the following:
    ;        A valid starting address (the least significant bits = '00')
    ;        is loaded in ADDRH:ADDRL
    ;        ADDRH, ADDRL and DATADDR are all located in data memory
    ;
   BANKSEL         PMADRH
   MOVF    ADDRH,W      ;Load initial address
   MOVWF   PMADRH       ;
   MOVF    ADDRL,W      ;
   MOVWF   PMADRL       ;
   MOVF    DATAADDR,W   ;Load initial data address
   MOVWF   FSR          ;
LOOP MOVF  INDF,W       ;Load first data byte into lower
   MOVWF   PMDATL       ;
   INCF    FSR,F        ;Next byte
   MOVF    INDF,W       ;Load second data byte into upper
   MOVWF   PMDATH       ;
   INCF    FSR,F        ;
   BANKSEL PMCON1
   BSF     PMCON1,WREN  ;Enable writes
   BCF     INTCON,GIE   ;Disable interrupts (if using)
   BTFSC   INTCON,GIE   ;See AN576
   GOTO    $-2
   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
   ;       Required Sequence
   MOVLW   55h          ;Start of required write sequence:
   MOVWF   PMCON2       ;Write 55h
   MOVLW   0AAh         ;
   MOVWF   PMCON2       ;Write 0AAh
   BSF     PMCON1,WR    ;Set WR bit to begin write
   NOP                  ;Required to transfer data to the buffer
   NOP                  ;registers
   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
   BCF     PMCON1,WREN  ;Disable writes
   BSF     INTCON,GIE   ;Enable interrupts (comment out if not using interrupts)
   BANKSEL PMADRL
   MOVF    PMADRL, W
   INCF    PMADRL,F     ;Increment address
   ANDLW   0x03         ;Indicates when sixteen words have been programmed
   SUBLW   0x03         ;Change value for different size write blocks
                        ;0x0F = 16 words
                        ;0x0B = 12 words
                        ;0x07 = 8 words
                        ;0x03 = 4 words
   BTFSS   STATUS,Z     ;Exit on a match,
   GOTO    LOOP         ;Continue if more data needs to be written
```

5.3 PORTA and TRISA Registers

PORTA is a 6-bit wide port with five bidirectional and one input-only pin. The corresponding data direction register is TRISA (Register 5-2). Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., disable the output driver). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., enables output driver and puts the contents of the output latch on the selected pin). The exception is RA3, which is input-only and its TRIS bit will always read as '1'. Example 5-1 shows how to initialize PORTA.

Reading the PORTA register (Register 5-2) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch. RA3 reads '0' when MCLRE = 1.

The TRISA register controls the direction of the PORTA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

Note:	The ANSEL register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0' and cannot generate an interrupt.

5.3.1 PORTA FUNCTIONS AND OUTPUT PRIORITIES

Each PORTA pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 5-1.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Analog input functions, such as comparator inputs, are not shown in the priority lists. These inputs are active when the peripheral is enabled and the input multiplexer for the pin is selected. The Analog mode, set with the ANSELA register, disables the digital input buffer thereby preventing excessive input current when the analog input voltage is between logic states. Digital output functions may control the pin when it is in Analog mode with the priority shown in Table 5-1.

TABLE 5-1: PORTA OUTPUT PRIORITY

Pin Name	Function Priority
RA0	ICSPDAT FVROUT DACOUT C1IN0+ RA0
RA1	FVRIN ICSPCLK V_{REF+} C1IN0- C2IN0- RA1
RA2	COG1FLT T0CKI C1OUT INT RA2
RA3	$\overline{MCLR}$ V_{PP} T1G RA3
RA4	CLKOUT T1G RA4
RA5	CLKIN T1CKI RA5

REGISTER 5-7: IOCAP: INTERRUPT-ON-CHANGE POSITIVE EDGE REGISTER

U-0	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
—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0
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 7-6 **Unimplemented**: Read as ‘0’

bit 5-0 **IOCAP<5:0>:** Interrupt-on-Change Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 5-8: IOCAN: INTERRUPT-ON-CHANGE NEGATIVE EDGE REGISTER

U-0	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
—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0
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 7-6 **Unimplemented**: Read as ‘0’

bit 5-0 **IOCAN<5:0>:** Interrupt-on-Change Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 5-9: IOCAF: INTERRUPT-ON-CHANGE FLAG REGISTER

U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0
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 - Bit is set in hardware

bit 7-6 **Unimplemented**: Read as ‘0’

bit 5-0 **IOCAF<5:0>:** Interrupt-on-Change Flag bits
1 = An enabled change was detected on the associated pin.
Set when IOCAPx = 1 and a rising edge was detected on RBx, or when IOCANx = 1 and a falling edge was detected on RAx.
0 = No change was detected, or the user cleared the detected change.

REGISTER 5-13: SLRCONC: SLEW RATE CONTROL REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
—	—	SLRC5	SLRC4	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **SLRC<5:4>:** Slew Rate Control Register bit
1 = Slew rate control enabled
0 = Slew rate control disabled

bit 3-0 **Unimplemented:** Read as ‘0’

REGISTER 5-14: ANSELC: PORTC ANALOG SELECT REGISTER

U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	ANSC3	ANSC2	ANSC1	ANSC0
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 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **ANSC<3:0>:** Analog Select Between Analog or Digital Function on Pin RC<3:0> bits
1 = Analog input. Pin is assigned as analog input.(1)
0 = Digital I/O. Pin is assigned to port or special function.

Note 1: Setting a pin to an analog input automatically disables the digital input circuitry, weak pull-ups, and interrupt-on-change if available. The corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

7.3 Timer1 Prescaler

Timer1 has four prescaler options allowing one, two, four or eight divisions of the clock input. The T1CKPS bits of the T1CON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

7.4 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 7.4.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

Note:	When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

7.4.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMR1H:TMR1L register pair.

7.5 Timer1 Gate

Timer1 can be configured to count freely or the count can be enabled and disabled using Timer1 gate circuitry. This is also referred to as Timer1 gate count enable.

Timer1 gate can also be driven by multiple selectable sources.

7.5.1 TIMER1 GATE COUNT ENABLE

The Timer1 gate is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 gate is configured using the T1GPOL bit of the T1GCON register.

When Timer1 Gate (T1G) input is active, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 gate input is inactive, no incrementing will occur and Timer1 will hold the current count. See Figure 7-3 for timing details.

TABLE 7-3: TIMER1 GATE ENABLE SELECTIONS

T1CLK	T1GPOL	T1G	Timer1 Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
↑	1	1	Counts

7.5.2 TIMER1 GATE SOURCE SELECTION

The Timer1 gate source can be selected from one of four different sources. Source selection is controlled by the T1GSS bits of the T1GCON register. The polarity for each available source is also selectable. Polarity selection is controlled by the T1GPOL bit of the T1GCON register.

TABLE 7-4: TIMER1 GATE SOURCES

T1GSS	Timer1 Gate Source
11	SYNCC2OUT
10	SYNCC1OUT
01	Overflow of Timer0 (TMR0 increments from FFh to 00h)
00	Timer1 Gate Pin

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

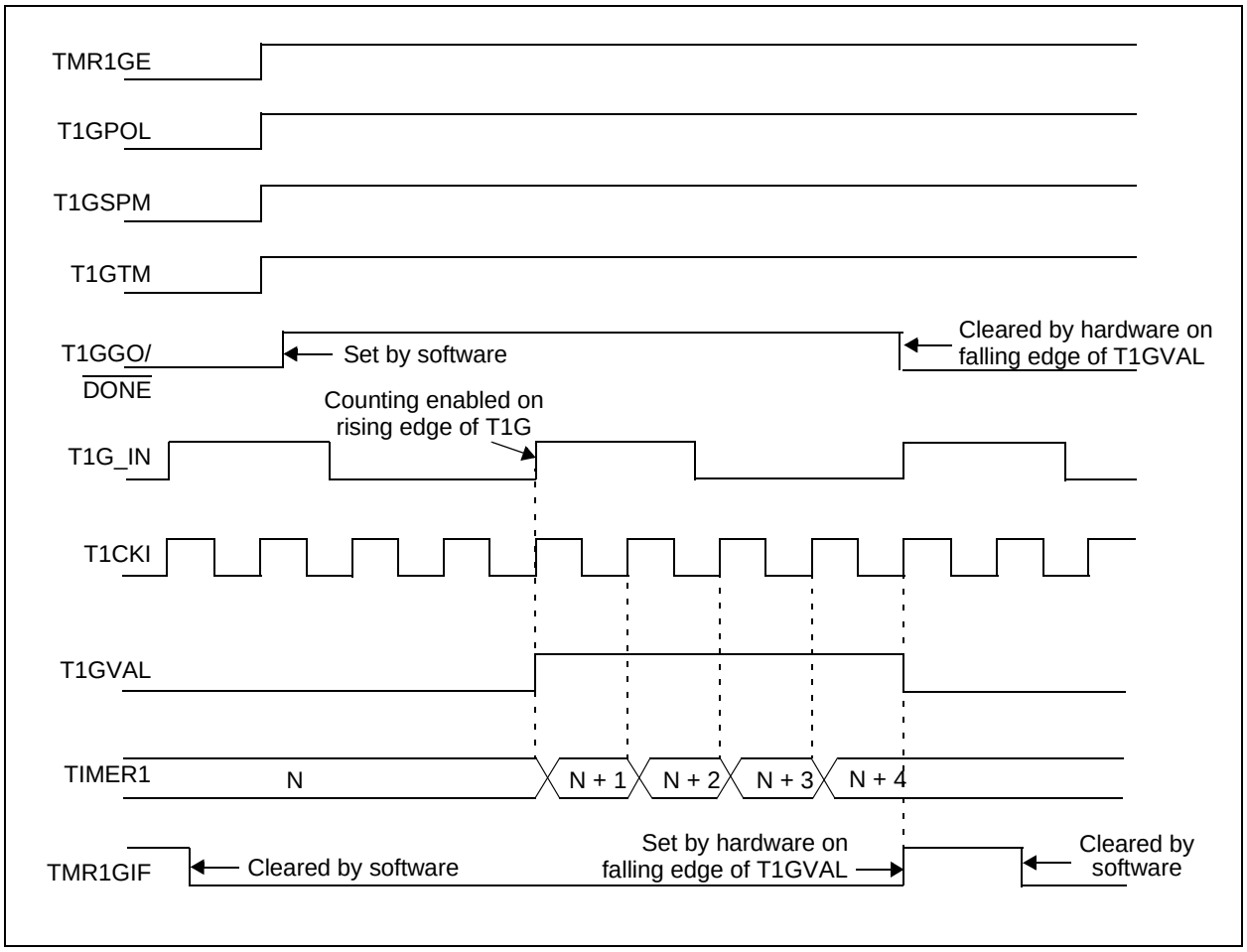


TABLE 7-5: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	44
APFCON	—	—	—	T1GSEL	—	—	—	—	40
CCP1CON	—	—	DC1B<1:0>		CCP1M<3:0>				80
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
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	43
TMR1H	TMR1H<7:0>								57*
TMR1L	TMR1L<7:0>								57*
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	43
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	65
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		66

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.
* Page provides register information.

9.1 HLT Operation

The clock input to the HLT module is the system instruction clock (FOSC/4). HLTMRx increments on each rising clock edge.

A 4-bit counter/prescaler on the clock input provides the following prescale options:

- Direct input
- Divide-by-4
- Divide-by-16
- Divide-by-64

The prescale options are selected by the prescaler control bits, HxCKPS<1:0> of the HLTxCON0 register.

The value of HLTMRx is compared to that of the Period register, HLTPRx, on each clock cycle. When the two values match, then the comparator generates a match signal as the HLTimerx output. This signal also resets the value of HLTMRx to 00h on the next clock rising edge and drives the output counter/postscaler (see **Section 9.2 "HLT Interrupt"**).

The HLTMRx and HLTPRx registers are both directly readable and writable. The HLTMRx register is cleared on any device Reset, whereas the HLTPRx register initializes to FFh. Both the prescaler and postscaler counters are cleared on any of the following events:

- A write to the HLTMRx register
- A write to the HLTxCON0 register
- Power-on Reset (POR)
- Brown-out Reset (BOR)
- $\overline{\text{MCLR}}$ Reset
- Watchdog Timer (WDT) Reset
- Stack Overflow Reset
- Stack Underflow Reset
- RESET Instruction

Note:	HLTMRx is not cleared when HLTxCON0 is written.

9.2 HLT Interrupt

The HLT can also generate an optional device interrupt. The HLTMRx output signal (HLTMRx-to-HLTPRx match) provides the input for the 4-bit counter/postscaler. The overflow output of the postscaler sets the HLTMRxIF bit of the PIR1 register. The interrupt is enabled by setting the HLTMRx Match Interrupt Enable bit, HLTMRxIE of the PIE1 register.

A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, HxOUTPS<3:0>, of the HLTxCON0 register.

9.3 Peripheral Resets

Resets driven from the selected peripheral output prevents the HLTMRx from matching the HLTPRx register and generating an output. In this manner, the HLT can be used as a hardware time limit to other peripherals.

In this device, the primary purpose of the HLT is to limit the COG PWM duty cycle. Normally, the COG operation uses analog feedback to determine the PWM duty cycle. The same feedback signal is used as an HLT Reset input. The HLTPRx register is set to occur at the maximum allowed duty cycle. If the analog feedback to the COG exceeds the maximum time, then an HLTMRx-to-HLTPRx match will occur and generate the output needed to limit the COG drive output.

The HLTMRx can be reset by one of several selectable peripheral sources. Reset inputs include:

- CCP1 output
- Comparator 1 output
- Comparator 2 output
- COGxFLT pin
- COG1OUT0
- COG1OUT1

The external Reset input is selected with the HxERS<2:0> bits of the HLTxCON1 register. High and low Reset enables are selected with the HxREREN and HxFEREN bits, respectively. Setting the HxRES and HxFES bits makes the respective rising and falling Reset events edge sensitive. Reset inputs that are not edge sensitive are level sensitive.

HLTMRx Resets are synchronous with the HLT clock. In other words, HLTMRx is cleared on the rising edge of the HLT clock after the enabled Reset event occurs.

If an enabled external Reset occurs at the same time a write occurs to the TMR4A register, the write to the timer takes precedence and pending Resets are cleared.

9.4 HLTimerx Output

The unscaled output of HLTMRx is available only to the COG module, where it is used as a selectable limit to the maximum COG period.

9.5 HLT Operation During Sleep

The HLT cannot be operated while the processor is in Sleep mode. The contents of the HLTMRx register will remain unchanged while the processor is in Sleep mode.

11.0 COMPLEMENTARY OUTPUT GENERATOR (COG) MODULE

The primary purpose of the Complementary Output Generator (COG) is to convert a single output PWM signal into a two output complementary PWM signal. The COG can also convert two separate input events into a single or complementary PWM output.

The COG PWM frequency and duty cycle are determined by a rising event input and a falling event input. The rising event and falling event may be the same source. Sources may be synchronous or asynchronous to the COG_clock.

The rate at which the rising event occurs determines the PWM frequency. The time from the rising event input to the falling event input determines the duty cycle.

A selectable clock input is used to generate the phase delay, blanking and dead-band times.

A simplified block diagram of the COG is shown in Figure 11-1.

The COG module has the following features:

- Two modes of operation:
  - Synchronous PWM
  - Push-pull
- Selectable clock source
- Independently selectable rising event sources
- Independently selectable falling event sources
- Independently selectable edge or level event sensitivity
- Independent output enables
- Independent output polarity selection
- Phase delay with independent rising and falling delay times
- Dead-band control with:
  - Independent rising and falling event dead-band times
  - Synchronous and asynchronous timing
- Blanking control with independent rising and falling event blanking times
- Auto-shutdown control with:
  - Independently selectable shutdown sources
  - Auto-restart enable
  - Auto-shutdown pin override control (high, low, off, and High-Z)

11.1 Fundamental Operation

11.1.1 SYNCHRONOUS PWM MODE

In synchronous PWM mode, the COG generates a two output complementary PWM waveform from rising and falling event sources. In the simplest configuration, the rising and falling event sources have the same signal, which is a PWM signal with the desired period and duty cycle. The COG converts this single PWM input into a dual complementary PWM output. The frequency and duty cycle of the dual PWM output match those of the single input PWM signal. The off-to-on transition of each output can be delayed from the on-to-off transition of the other output, thereby creating a time immediately after the PWM transition where neither output is driven. This is referred to as dead time and is covered in **Section 11.5 “Dead-Band Control”**.

A typical operating waveform, with dead band, generated from a single CCP1 input is shown in Figure 11-4.

11.1.2 PUSH-PULL MODE

In Push-Pull mode, the COG generates a single PWM output that alternates every PWM period, between the two COG output pins. The output drive activates with the rising input event and terminates with the falling event input. Each rising event starts a new period and causes the output to switch to the COG pin not used in the previous period.

A typical push-pull waveform generated from a single CCP1 input is shown in Figure 11-6.

Push-Pull mode is selected by setting the GxMD bit of the COGxCON0 register.

11.1.3 ALL MODES

In addition to generating a complementary output from a single PWM input, the COG can also generate PWM waveforms from a periodic rising event and a separate falling event. In this case, the falling event is usually derived from analog feedback within the external PWM driver circuit. In this configuration, high-power switching transients may trigger a false falling event that needs to be blanked out. The COG can be configured to blank falling (and rising) event inputs for a period of time immediately following the rising (and falling) event drive output. This is referred to as input blanking and is described in **Section 11.6 “Blanking Control”**.

It may be necessary to guard against the possibility of circuit faults. In this case, the active drive must be terminated before the Fault condition causes damage. This is referred to as auto-shutdown and is described in **Section 11.8 “Auto-shutdown Control”**.

A feedback falling event arriving too late or not at all can be terminated with auto-shutdown or by enabling one of the Hardware Limit Timer (HLT) event inputs. See **Section 9.0 “Hardware Limit Timer (HLT) Module”** for more information about the HLT.

The COG can be configured to operate in phase delayed conjunction with another PWM. The active drive cycle is delayed from the rising event by a phase delay timer. Phase delay is covered in more detail in **Section 11.7 “Phase Delay”**. A typical operating waveform, with phase delay and dead band, generated from a single CCP1 input, is shown in Figure 11-5.

When the phase delay count value is zero, phase delay is disabled and the phase delay counter output is true, thereby allowing the event signal to pass straight through to complementary output driver flop.

11.7.1 CUMULATIVE UNCERTAINTY

It is not possible to create more than one COG_clock of uncertainty by successive stages. Consider that the phase delay stage comes after the blanking stage, the dead-band stage comes after either the blanking or phase delay stages, and the blanking stage comes after the dead-band stage. When the preceding stage is enabled, the output of that stage is necessarily synchronous with the COG_clock, which removes any possibility of uncertainty in the succeeding stage.

EQUATION 11-1: PHASE, DEAD-BAND AND BLANKING TIME CALCULATION

$$T_{\text{min}} = \frac{\text{Count}}{F_{COG\_clock}}$$

$$T_{\text{max}} = \frac{\text{Count} + 1}{F_{COG\_\text{clock}}}$$

$$T_{\text{uncertainty}} = T_{\text{max}} - T_{\text{min}}$$

Also:

$$T_{\text{uncertainty}} = \frac{1}{F_{COG\_\text{clock}}}$$

Where:

T	Count
Rising Phase Delay	COGxPHR
Falling Phase Delay	COGxPHF
Rising Dead Band	COGxDBR
Falling Dead Band	COGxDBF
Rising Event Blanking	COGxBKR
Falling Event Blanking	COGxBKF

EQUATION 11-2: TIMER UNCERTAINTY

Given:

$$Count = Ah = 10d$$

$$F_{COG\_\text{Clock}} = 8\text{MHz}$$

Therefore:

$$T_{\text{uncertainty}} = \frac{1}{F_{COG\_\text{clock}}}$$

$$= \frac{1}{8MHz} = 125ns$$

Proof:

$$T_{\text{min}} = \frac{Count}{F_{COG\_\text{clock}}}$$

$$= 125ns \bullet 10d = 1.25\mu s$$

$$T_{\text{max}} = \frac{Count + 1}{F_{COG\_\text{clock}}}$$

$$= 125ns \bullet (10d + 1)$$

$$= 1.375\mu s$$

Therefore:

$$T_{\text{uncertainty}} = T_{\text{max}} - T_{\text{min}}$$

$$= 1.375\mu s - 1.25\mu s$$

$$= 125ns$$

REGISTER 12-3: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0 (READ-ONLY)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
ADRESH<9:2>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **ADRESH<9:2>**: ADC Result Register bits
Upper eight bits of 10-bit conversion result

REGISTER 12-4: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0 (READ-ONLY)

R-x	R-x	U-0	U-0	U-0	U-0	U-0	U-0
ADRESL<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **ADRESL<7:0>**: ADC Result Register bits
Lower two bits of 10-bit conversion result

REGISTER 12-5: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1 (READ-ONLY)

U-0	U-0	U-0	U-0	U-0	U-0	R-x	R-x
—	—	—	—	—	—	ADRESH<9:8>	
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 1-0 **ADRESH<9:8>**: ADC Result Register bits
Upper two bits of 10-bit conversion result

REGISTER 12-6: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 1 (READ-ONLY)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
ADRESL<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **ADRESL<7:0>**: ADC Result Register bits
Lower eight bits of 10-bit conversion result

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’

15.11 Analog Input Connection Considerations

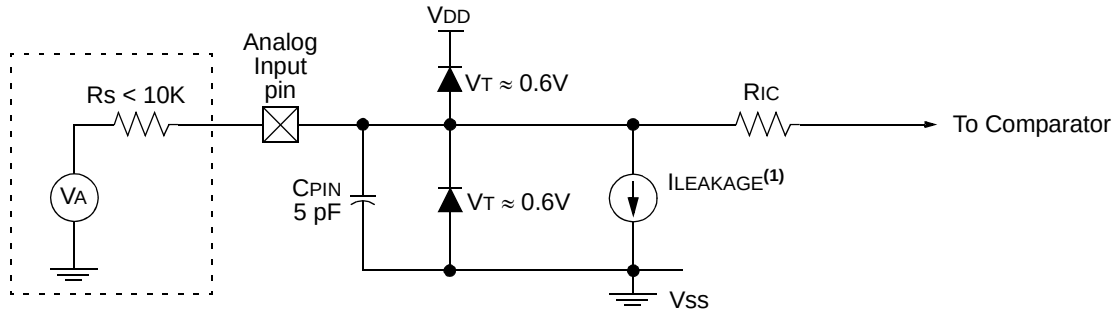
A simplified circuit for an analog input is shown in Figure 15-4. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

Note 1: When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.

2: Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

FIGURE 15-4: ANALOG INPUT MODEL



Legend: CPIN = Input Capacitance
ILEAKAGE = Leakage Current at the pin due to various junctions
RIC = Interconnect Resistance
RS = Source Impedance
VA = Analog Voltage
VT = Threshold Voltage

Note 1: See **Section 22.0 "Electrical Specifications"**.

TABLE 17-1: SLOPE COMPENSATOR CURRENT SETTINGS

SC1ISET Value	Current Setting (uA)	Slope Value (V/us)	SC1ISET Value	Current Setting (uA)	Slope Value (V/us)
0h	2	0.2	10h	10	1.0
1h	2.5	0.25	11h	11	1.1
2h	3	0.3	12h	12	1.2
3h	3.5	0.35	13h	13	1.3
4h	4	0.4	14h	14	1.4
5h	4.5	0.45	15h	15	1.5
6h	5	0.5	16h	16	1.6
7h	5.5	0.55	17h	17	1.7
8h	6	0.6	18h	18	1.8
9h	6.5	0.65	19h	19	1.9
Ah	7	0.7	1Ah	20	2.0
Bh	7.5	0.75	1Bh	21	2.1
Ch	8	0.8	1Ch	22	2.2
Dh	8.5	0.85	1Dh	23	2.3
Eh	9	0.9	1Eh	24	2.4
Fh	9.5	0.95	1Fh	25	2.5

TABLE 17-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE SC MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
SLPCCON0	SC1EN	—	—	SC1POL	SC1TSS<1:0>		—	SC1INS	138
SLPCCON1	—	—	—	SC1RNG	SC1ISET<3:0>				138
PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	49
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	49
ANSELC	—	—	—	—	ANSC3	ANSC2	ANSC1	ANSC0	50
WPUC	—	—	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	51

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the slope compensation module.

19.4 Interrupts

The PIC16F753/HV753 has multiple sources of interrupt:

- External Interrupt (INT pin)
- Interrupt-On-Change (IOC) Interrupts
- Timer0 Overflow Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt
- Hardware Limit Timer (HLT) Interrupt
- Comparator Interrupt (C1/C2)
- ADC Interrupt
- Complementary Output Generator (COG)
- CCP1 Interrupt
- Flash Memory Self-Write

The Interrupt Control register (INTCON) and Peripheral Interrupt Request Registers (PIRx) record individual interrupt requests in flag bits. The INTCON register also has individual and global interrupt enable bits.

The Global Interrupt Enable bit, GIE of the INTCON register, enables (if set) all unmasked interrupts, or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in the INTCON register and PIEx registers. GIE is cleared on Reset.

When an interrupt is serviced, the following actions occur automatically:

- The GIE is cleared to disable any further interrupt.
- The return address is pushed onto the stack.
- The PC is loaded with 0004h.

The Return from Interrupt instruction, `RETFIE`, exits the interrupt routine, as well as sets the GIE bit, which re-enables unmasked interrupts.

The following interrupt flags are contained in the INTCON register:

- INT Pin Interrupt
- Interrupt-On-Change (IOC) Interrupts
- Timer0 Overflow Interrupt

The peripheral interrupt flags are contained in the PIR1 and PIR2 registers. The corresponding interrupt enable bit is contained in the PIE1 and PIE2 registers.

The following interrupt flags are contained in the PIR1 register:

- A/D Interrupt
- Comparator Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt
- Enhanced CCP Interrupt

For external interrupt events, such as the INT pin or PORTA change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends upon when the interrupt event occurs (see Figure 19-8). The latency is the same for one or two-cycle instructions. Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid multiple interrupt requests.

Note 1:	Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit.
2:	When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The interrupts, which were ignored, are still pending to be serviced when the GIE bit is set again.

For additional information on Timer1, Timer2, comparators, ADC, Enhanced CCP modules, refer to the respective peripheral section.

19.4.1 RA2/INT INTERRUPT

The external interrupt on the RA2/INT pin is edge-triggered; either on the rising edge if the INTEDG bit of the OPTION register is set, or the falling edge, if the INTEDG bit is clear. When a valid edge appears on the RA2/INT pin, the INTF bit of the INTCON register is set. This interrupt can be disabled by clearing the INTE control bit of the INTCON register. The INTF bit must be cleared by software in the Interrupt Service Routine before re-enabling this interrupt. The RA2/INT interrupt can wake-up the processor from Sleep, if the INTE bit was set prior to going into Sleep. See **Section 19.7 "Power-Down Mode (Sleep)"** for details on Sleep and Figure 19-10 for timing of wake-up from Sleep through RA2/INT interrupt.

Note:	The ANSEL register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0' and cannot generate an interrupt.

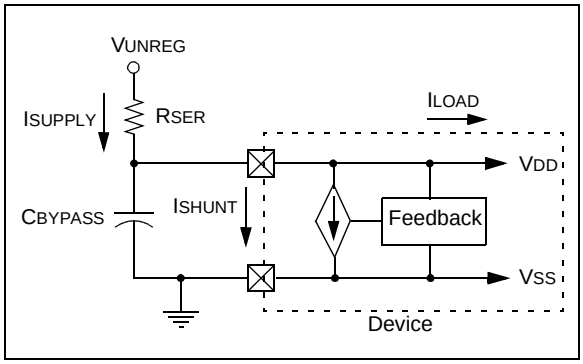
20.0 SHUNT REGULATOR (PIC16HV753 ONLY)

The PIC16HV753 devices include a permanent internal 5 volt (nominal) shunt regulator in parallel with the VDD pin. This eliminates the need for an external voltage regulator in systems sourced by an unregulated supply. All external devices connected directly to the VDD pin will share the regulated supply voltage and contribute to the total VDD supply current (ILOAD).

20.1 Regulator Operation

A shunt regulator generates a specific supply voltage by creating a voltage drop across a pass resistor RSER. The voltage at the VDD pin of the microcontroller is monitored and compared to an internal voltage reference. The current through the resistor is then adjusted, based on the result of the comparison, to produce a voltage drop equal to the difference between the supply voltage VUNREG and the VDD of the microcontroller. See Figure 20-1 for voltage regulator schematic.

FIGURE 20-1: SHUNT REGULATOR



An external current limiting resistor, RSER, located between the unregulated supply, VUNREG, and the VDD pin, drops the difference in voltage between VUNREG and VDD. RSER must be between RMAX and RMIN as defined by Equation 20-1.

EQUATION 20-1: RSER LIMITING RESISTOR

$$R_{MAX} = \frac{(VU_{MIN} - 5V)}{1.05 \bullet (1\ mA + I_{LOAD})}$$

$$R_{MIN} = \frac{(VU_{MAX} - 5V)}{0.95 \bullet (50\ mA)}$$

Where:

RMAX = maximum value of RSER (ohms)

RMIN = minimum value of RSER (ohms)

VUMIN = minimum value of VUNREG

VUMAX = maximum value of VUNREG

VDD = regulated voltage (5V nominal)

ILOAD = maximum expected load current in mA including I/O pin currents and external circuits connected to VDD.

1.05 = compensation for +5% tolerance of RSER

0.95 = compensation for -5% tolerance of RSER

20.2 Regulator Considerations

The supply voltage VUNREG and load current are not constant. Therefore, the current range of the regulator is limited. Selecting a value for RSER must take these three factors into consideration.

Since the regulator uses the band gap voltage as the regulated voltage reference, this voltage reference is permanently enabled in the PIC16HV753 devices.

The shunt regulator will still consume current when below operating voltage range for the shunt regulator.

20.3 Design Considerations

For more information on using the shunt regulator and managing current load, see Application Note *AN1035*, *Designing with HV Microcontrollers* (DS01035).

TABLE 22-3: POWER-DOWN CURRENTS (I_{PD}) (CONTINUED)(1,2)

PIC16F753		Standard Operating Conditions (unless otherwise stated) Sleep mode						
PIC16HV753								
Param No.	Device Characteristics	Min.	Typ†	Max. 85°C	Max. 125°C	Units	Conditions	
							V_{DD}	Note
	Power-down Base Current (I_{PD})(2, 3)							
D025		—	0.10	0.41	3.51	µA	3.0	A/D Current(1), no conversion in progress
		—	0.12	0.55	4.41	µA	5.0	
D025		—	145	171	175	µA	3.0	
		—	185	226	231	µA	4.5	
D026		—	20	37	37	µA	2.0	DAC Current(1)
		—	30	46	46	µA	3.0	
		—	50	76	76	µA	5.0	
D026		—	85	155	155	µA	2.0	
		—	165	213	213	µA	3.0	
		—	215	284	284	µA	4.5	
D027		—	115	185	203	µA	2.0	FVR Current(1), FVRBUFEN = 1, FVROUT buffer enabled
		—	120	193	219	µA	3.0	
		—	125	196	224	µA	5.0	
D027		—	65	126	145	µA	2.0	
		—	136	171	182	µA	3.0	
		—	175	226	231	µA	4.5	
D028		—	1	2	4	µA	2.0	T1OSC Current, TMR1CS <1:0> = 11
		—	2	3	5	µA	3.0	
		—	9	20	21	µA	5.0	
D028		—	65	126	140	µA	2.0	
		—	136	172	180	µA	3.0	
		—	175	228	235	µA	4.5	
D029		—	140	258	265	µA	2.0	Op-Amp Current(1)
		—	155	326	340	µA	3.0	
		—	165	421	422	µA	5.0	
D029		—	140	260	265	µA	2.0	
		—	155	325	340	µA	3.0	
		—	165	400	410	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 peripheral Δ current can be determined by subtracting the base I_{PD} current from this limit. Max values should be used when calculating total current consumption.

2: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to V_{SS}.

3: Shunt regulator is always ON and always draws operating current.

FIGURE 23-3: IDD TYPICAL, EC OSCILLATOR, MEDIUM-POWER MODE, PIC16F753 ONLY

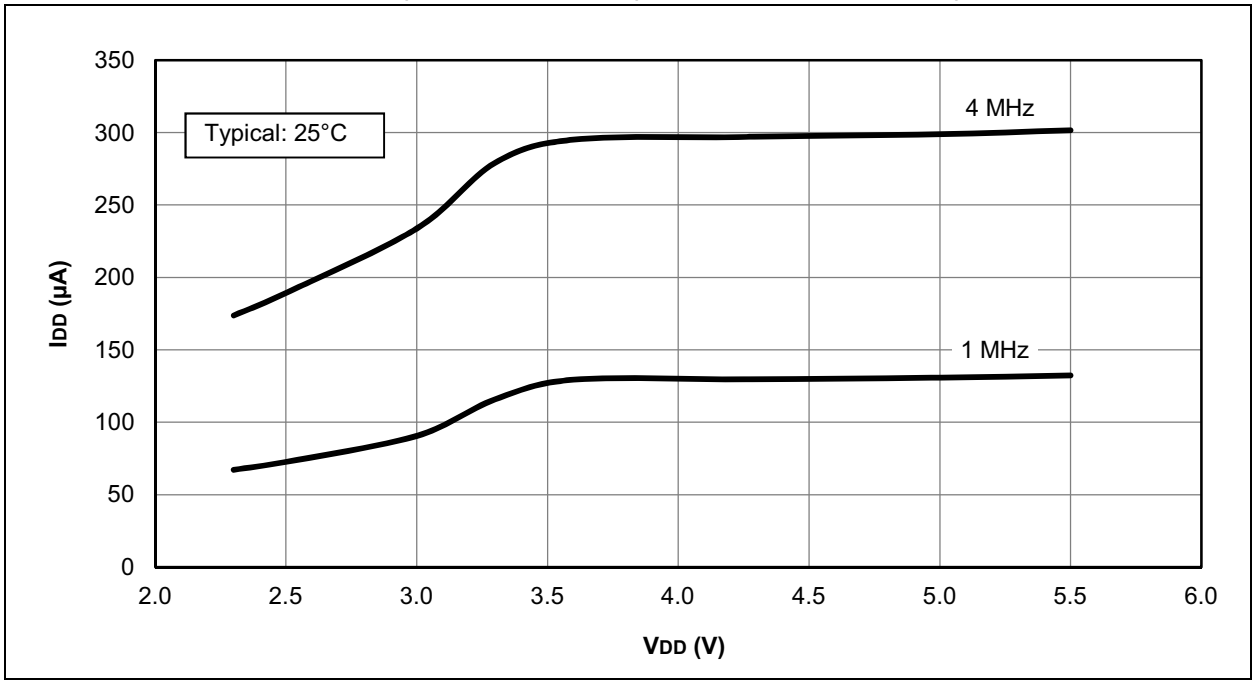


FIGURE 23-4: IDD MAXIMUM, EC OSCILLATOR, MEDIUM-POWER MODE, PIC16F753 ONLY