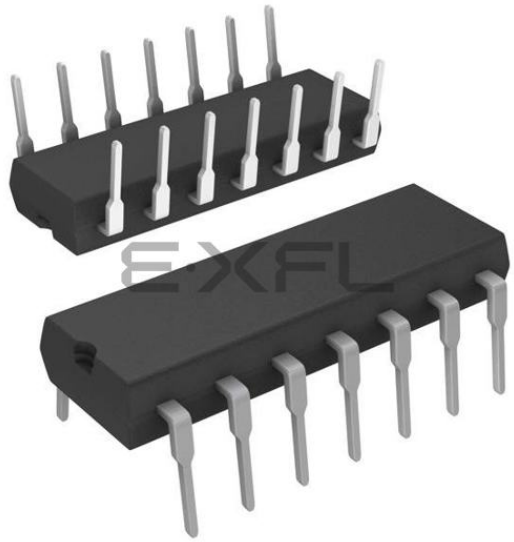


Welcome to E-XFL.COM

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	Through Hole
Package / Case	14-DIP (0.300", 7.62mm)
Supplier Device Package	14-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16hv753-e-p

TABLE 2-1: PIC16F753/HV753 SPECIAL REGISTERS SUMMARY BANK 0

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR	Value on all other Resets
Bank 0											
00h	INDF	INDF<7:0>								xxxx xxxx	xxxx xxxx
01h	TMR0	TMR0<7:0>								xxxx xxxx	uuuu uuuu
02h	PCL	PCL<7:0>								0000 0000	0000 0000
03h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h	FSR	FSR<7:0>								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--uu uuuu
06h	—	Unimplemented								—	—
07h	PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx xxxx	--uu uuuu
08h	IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	--00 0000	--00 0000
09h	IOCCF	—	—	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	--00 0000	--00 0000
0Ah	PCLATH	—	—	—	PCLATH<4:0>					---0 0000	---0 0000
0Bh	INTCON	GIE	PEIE	T0IE	INTE	IOCIE	T0IF	INTF	IOCIF	0000 0000	0000 0000
0Ch	PIR1	TMR1GIF	ADIF	—	—	HLTMR2IF	HLTMR1IF	TMR2IF	TMR1IF	00--0000	00--0000
0Dh	PIR2	—	—	C2IF	C1IF	—	COG1IF	—	CCP1IF	--00 -0-0	--00 -0-0
0Eh	—	Unimplemented								—	—
0Fh	TMR1L	TMR1L<7:0>								xxxx xxxx	uuuu uuuu
10h	TMR1H	TMR1H<7:0>								xxxx xxxx	uuuu uuuu
11h	T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	$\overline{T1SYNC}$	—	TMR1ON	0000 00-0	0000 00-0
12h	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{DONE}$	T1GVAL	T1GSS<1:0>		0000 0x00	0000 0x00
13h	CCPR1L	CCPR1L<7:0>								xxxx xxxx	uuuu uuuu
14h	CCPR1H	CCPR1H<7:0>								xxxx xxxx	uuuu uuuu
15h	CCP1CON	—	—	DC1B<1:0>		CCP1M<3:0>				--00 0000	--00 0000
16h	—	Unimplemented								—	—
17h	—	Unimplemented								—	—
18h	—	Unimplemented								—	—
19h	—	Unimplemented								—	—
1Ah	—	Unimplemented								—	—
1Bh	—	Unimplemented								—	—
1Ch	ADRESL	Least Significant two bits of the left shifted result or eight bits of the right shifted result								xxxx xxxx	uuuu uuuu
1Dh	ADRESH	Most Significant eight bits of the left shifted A/D result or two bits of the right shifted result								xxxx xxxx	uuuu uuuu
1Eh	ADCON0	ADFM	—	CHS<3:0>				GO/$\overline{DONE}$	ADON	0-00 0000	0-00 0000
1Fh	ADCON1	—	ADCS<2:0>			—	—	—	ADPREF1	-000 ---0	-000 ---0

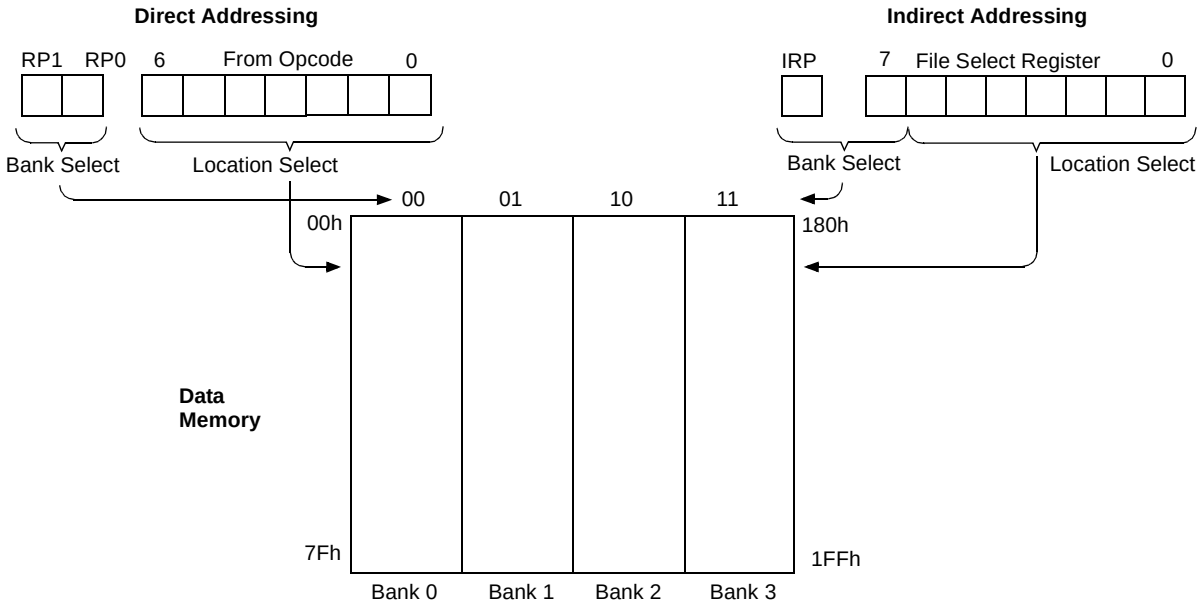
Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition shaded = unimplemented.

TABLE 2-4: PIC16F753/HV753 SPECIAL FUNCTION REGISTERS SUMMARY BANK 3

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 3											
180h	INDF	INDF<7:0>								xxxx xxxx	uuuu uuuu
181h	OPTION_REG	$\overline{RAPU}$	INTEDG	T0CS	T0SE	PSA	PS<2:0>			1111 1111	1111 1111
182h	PCL	PCL<7:0>								0000 0000	0000 0000
183h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
184h	FSR	FSR<7:0>								xxxx xxxx	uuuu uuuu
185h	ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	---1 -111	---1 -111
186h	—	Unimplemented								—	—
187h	ANSELC	—	—	—	—	ANSC3	ANSC2	ANSC1	ANSC0	---- 0000	---- 0000
188h	APFCON	—	—	—	T1GSEL	—	—	—	—	---0 ----	---0 ----
189h	OSCTUNE	—	—	—	TUN<4:0>					---0 0000	---0 0000
18Ah	PCLATH	—	—	—	PCLATH<4:0>					---0 0000	---0 0000
18Bh	INTCON	GIE	PEIE	T0IE	INTE	IOCIE	T0IF	INTF	IOCIF	0000 0000	0000 0000
18Ch	PMCON1	—	—	—	—	—	WREN	WR	RD	---- -000	---- -000
18Dh	PMCON2	Program Memory Control Register 2								---- ----	---- ----
18Eh	PMADRL	PMADRL<7:0>								0000 0000	0000 0000
18Fh	PMADRH	—	—	—	—	—	—	PMADRH<1:0>		---- --00	---- --00
190h	PMDATL	PMDATL<7:0>								0000 0000	0000 0000
191h	PMDATH	—	—	PMDATH<5:0>						--00 0000	--00 0000
192h	COG1PHR	—	—	—	—	G1PHR<3:0>				---- xxxx	---- uuuu
193h	COG1PHF	—	—	—	—	G1PHF<3:0>				---- xxxx	---- uuuu
194h	COG1BKR	—	—	—	—	G1BKR<3:0>				---- xxxx	---- uuuu
195h	COG1BKF	—	—	—	—	G1BKF<3:0>				---- xxxx	---- uuuu
196h	COG1DBR	—	—	—	—	G1DBR<3:0>				---- xxxx	---- uuuu
197h	COG1DBF	—	—	—	—	G1DBF<3:0>				---- xxxx	---- uuuu
198h	COG1CON0	G1EN	G1OE1	G1OE0	G1POL1	G1POL0	G1LD	—	G1MD	0000 00-0	0000 00-0
199h	COG1CON1	G1RDBTS	G1FDBTS	—	—	—	—	G1CS<1:0>		00-- --00	00-- --00
19Ah	COG1RIS	—	G1RIHLT2	G1RIHLT1	G1RIT2M	G1RIFLT	G1RICCP1	G1RIC2	G1RIC1	0000 0000	0000 0000
19Bh	COG1RSIM	—	G1RMHLT2	G1RMHLT1	G1RMT2M	G1RMFLT	G1RMCCP1	G1RMC2	G1RMC1	0000 0000	0000 0000
19Ch	COG1FIS	—	G1FIHLT2	G1FIHLT1	G1FIT2M	G1FIFLT	G1FICCP1	G1FIC2	G1FIC1	0000 0000	0000 0000
19Dh	COG1FSIM	—	G1FMHLT2	G1FMHLT1	G1FMT2M	G1FMFLT	G1FMCCP1	G1FMC2	G1FMC1	0000 0000	0000 0000
19Eh	COG1ASD0	C1ASDE	C1ARSEN	G1ASD1L<1:0>		G1ASD0L<1:0>		—	—	0000 00--	0000 00--
19Fh	COG1ASD1	—	—	—	G1ASDSHLT2	G1ASDSHLT1	G1ASDSC2	G1ASDSC1	G1ASDSFLT	0000 0000	0000 0000

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

FIGURE 2-4: DIRECT/INDIRECT ADDRESSING PIC16F753/HV753



For memory map detail, see Figure 2-2.

3.4 Reading the Flash Program Memory

To read a program memory location, the user must write two bytes of the address to the PMADRL and PMADRH registers, and then set control bit RD (PMCON1<0>). Once the read control bit is set, the program memory Flash controller will use the second instruction cycle after to read the data. This causes the second instruction immediately following the "`BSF PMCON1,RD`" instruction to be ignored. The data is available in the very next cycle in the PMDATL and PMDATH registers; it can be read as two bytes in the following instructions. PMDATL and PMDATH registers will hold this value until another read or until it is written to by the user (during a write operation).

EXAMPLE 3-1: FLASH PROGRAM READ

```
    BANKSEL  PM_ADR          ; Change STATUS bits RP1:0 to select bank with PMADRL
    MOVLW    MS_PROG_PM_ADDR ;
    MOVWF    PMADRH          ; MS Byte of Program Address to read
    MOVLW    LS_PROG_PM_ADDR ;
    MOVWF    PMADRL          ; LS Byte of Program Address to read
    BANKSEL  PMCON1          ; Bank to containing PMCON1
    BSF      PMCON1, RD      ; PM Read

    NOP                      ; First instruction after BSF PMCON1,RD executes normally

    NOP                      ; Any instructions here are ignored as program
                             ; memory is read in second cycle after BSF PMCON1,RD
                             ;
    BANKSEL  PMDATL          ; Bank to containing PMADRL
    MOVF     PMDATL, W       ; W = LS Byte of Program PMDATL
    MOVF     PMDATH, W       ; W = MS Byte of Program PMDATL
```

5.6 PORTC Registers

PORTC is a 6-bit wide port with five bidirectional and one input-only pin. The corresponding data direction register is TRISC (Register 5-2). Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., disable the output driver). Clearing a TRISC bit (= 0) will make the corresponding PORTC 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 PORTC.

Reading the PORTC 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. RC3 reads '0' when MCLRE = 1.

The TRISC register controls the direction of the PORTC pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISC 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.

5.6.1 PORTC FUNCTIONS AND OUTPUT PRIORITIES

Each PORTC 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 ANSELC 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 5-3: PORTC OUTPUT PRIORITY

Pin Name	Function Priority
RC0	OPA1IN+ C2IN0+ RC0
RC1	OPA1IN- C1IN1- C2IN1- RC1
RC2	SLPCIN OPA1OUT C1IN2- C2IN2- RC2
RC3	C1IN3- C2IN3- RC3
RC4	COG1OUT1 C2OUT RC4
RC5	COG1OUT0 CCP1 RC5

7.5.2.1 T1G Pin Gate Operation

The T1G pin is one source for Timer1 gate control. It can be used to supply an external source to the Timer1 gate circuitry.

7.5.2.2 Timer0 Overflow Gate Operation

When Timer0 increments from FFh to 00h, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 gate circuitry.

7.5.2.3 C1OUT/C2OUT Gate Operation

The outputs from the Comparator C1 and C2 modules can be used as gate sources for the Timer1 module.

7.5.3 TIMER1 GATE TOGGLE MODE

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

The Timer1 gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. See Figure 7-4 for timing details.

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

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

7.5.4 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 T1GSPM bit in the T1GCON register. Next, the T1GGO/$\overline{\text{DONE}}$ bit in the T1GCON register must be set. The Timer1 will be fully enabled on the next incrementing edge. On the next trailing edge of the pulse, the T1GGO/$\overline{\text{DONE}}$ bit will automatically be cleared. No other gate events will be allowed to increment Timer1 until the T1GGO/$\overline{\text{DONE}}$ bit is once again set in software.

Clearing the T1GSPM bit of the T1GCON register will also clear the T1GGO/$\overline{\text{DONE}}$ bit. See Figure 7-5 for timing details.

Enabling the Toggle mode and the Single-Pulse mode simultaneously will permit both sections to work together. This allows the cycle times on the Timer1 gate source to be measured. See Figure 7-6 for timing details.

7.5.5 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 T1GVAL bit in the T1GCON register. The T1GVAL bit is valid even when the Timer1 gate is not enabled (TMR1GE bit is cleared).

7.5.6 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 T1GVAL occurs, the TMR1GIF flag bit in the PIR1 register will be set. If the TMR1GIE bit in the PIE1 register is set, then an interrupt will be recognized.

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

TABLE 11-1: SUMMARY OF REGISTERS ASSOCIATED WITH COG

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
COG1PHR	—	—	—	—	G1PHR<3:0>				102
COG1PHF	—	—	—	—	G1PHF<3:0>				102
COG1BKR	—	—	—	—	G1BKR<3:0>				101
COG1BKF	—	—	—	—	G1BKF<3:0>				101
COG1DBR	—	—	—	—	G1DBR<3:0>				100
COG1DBF	—	—	—	—	G1DBF<3:0>				100
COG1RIS	—	G1RIHLT2	G1RIHLT1	G1RIT2M	G1RIFLT	G1RICCP1	G1RIC2	G1RIC1	94
COG1RSIM	—	G1RMHLT2	G1RMHLT1	G1RMT2M	G1RMFLT	G1RMCCP1	G1RMC2	G1RMC1	95
COG1FIS	—	G1FIHLT2	G1FIHLT1	G1FIT2M	G1FIFLT	G1FICCP1	G1FIC2	G1FIC1	96
COG1FSIM	—	G1FMHLT2	G1FMHLT1	G1FMT2M	G1FMFLT	G1FMCCP1	G1FMC2	G1FMC1	97
COG1CON0	G1EN	G1OE1	G1OE0	G1POL1	G1POL0	G1LD	—	G1MD	92
COG1CON1	G1RDBTS	G1FDBTS	—	—	—	—	G1CS<1:0>		93
COG1ASD0	G1ASDE	G1ARSEN	G1ASD1L<1:0>		G1ASD0L<1:0>		—	—	98
COG1ASD1	—	—	—	G1ASDSHLT2	G1ASDSHLT1	G1ASDSC2	G1ASDSC1	G1ASDSFLT	99
INTCON	GIE	PEIE	T0IE	INTE	IOCIE	T0IF	INTF	IOCIF	17
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	43
PIE2	—	—	C2IE	C1IE	—	COG1IE	—	CCP1IE	19
PIR2	—	—	C2IF	C1IF	—	COG1IF	—	CCP1IF	21
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	43

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by COG.

12.3 Register Definitions: ADC Control

REGISTER 12-1: ADCON0: A/D CONTROL REGISTER 0

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	CHS<3:0>				GO/$\overline{\text{DONE}}$	ADON
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 **ADFM:** A/D Conversion Result Format Select bit
1 = Right justified
0 = Left justified

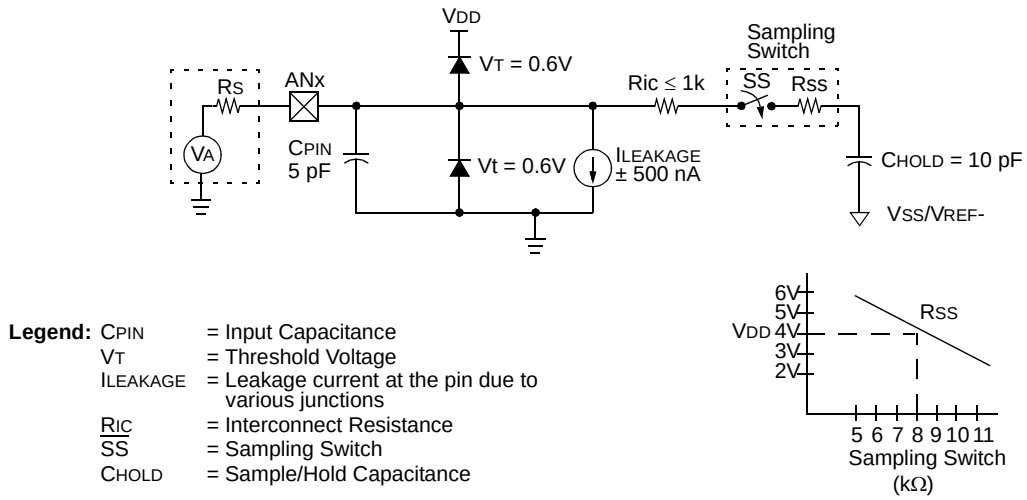
bit 6 **Unimplemented:** Read as ‘0’

bit 5-2 **CHS<3:0>:** Analog Channel Select bits
0000 = AN0
0001 = AN1
0010 = AN2
0011 = AN3
0100 = AN4
0101 = AN5
0110 = AN6
0111 = AN7
1110 = DAC output
1111 = Fixed Voltage Reference

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

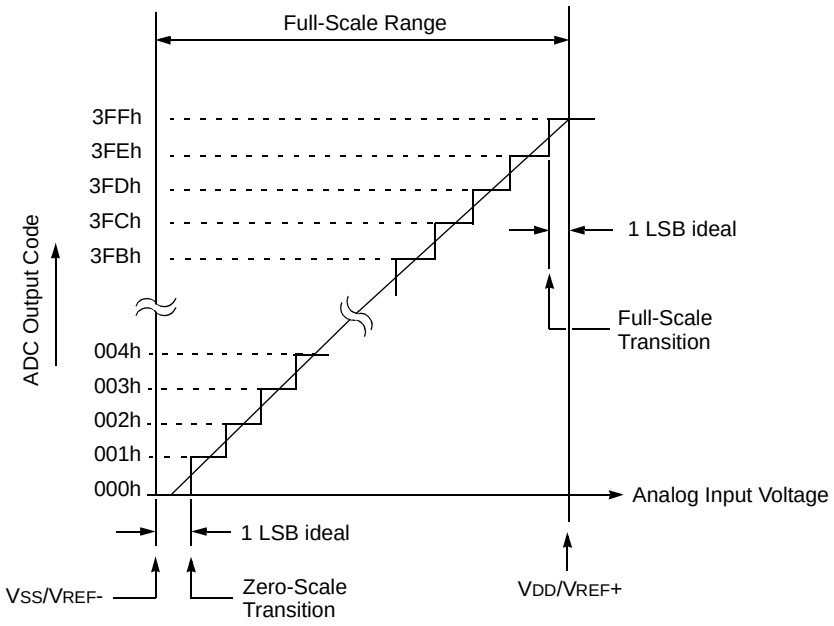
bit 0 **ADON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

FIGURE 12-4: ANALOG INPUT MODEL



Legend: CPIN = Input Capacitance
VT = Threshold Voltage
ILEAKAGE = Leakage current at the pin due to various junctions
RIC = Interconnect Resistance
SS = Sampling Switch
CHOLD = Sample/Hold Capacitance

FIGURE 12-5: ADC TRANSFER FUNCTION



15.11 Analog Input Connection Considerations

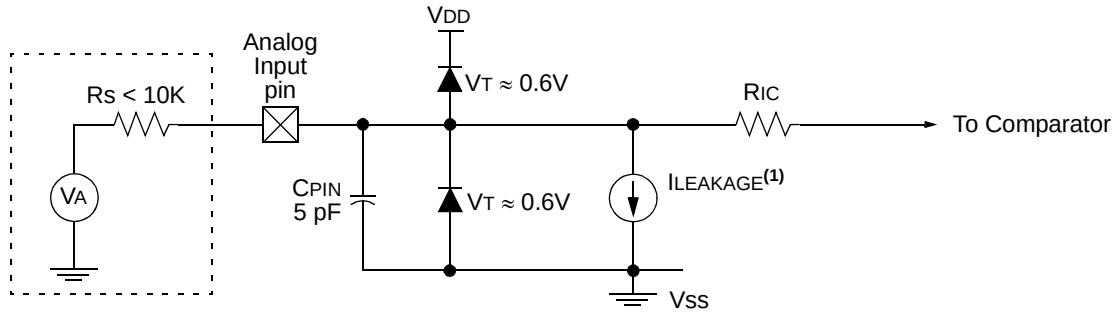
A simplified circuit for an analog input is shown in Figure 15-4. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range 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"**.

16.0 OPERATIONAL AMPLIFIER (OPA) MODULE

The Operational Amplifier (OPA) is a standard three-terminal device requiring external feedback to operate. The OPA module has the following features:

- External Connections to I/O Ports
- Selectable Unity Gain Bandwidth Product Option
- Low-Leakage Inputs
- Factory-Calibrated Input Offset Voltage

16.1 OPAxCON0 Register

The OPAxCON0 register, shown in Register 16-1, controls the OPA module.

The OPA module is enabled by setting the OPAxEN bit of the OPAxCON register. When enabled, the OPA forces the output driver of the OPAxOUT pin into tri-state to prevent contention between the driver and the OPA output.

The OPAxUGM bit of the OPAxCON register enables the Unity Gain Bandwidth mode (voltage follower) of the amplifier. In Unity Gain mode, the OPAxNCH<1:0> inputs are disabled. The default mode is normal three-terminal operation.

Note:	When the OPA module is enabled, the OPAxOUT pin is driven by the op amp output, not by the PORT digital driver. Refer to **Section 22.0 "Electrical Specifications"** for the op amp output drive capability.

16.2 Effects of a Reset

A device Reset forces all registers to their Reset state. This disables the OPA module.

16.3 OPA Module Performance

Common AC and DC performance specifications for the OPA module:

- Common Mode Voltage Range
- Leakage Current
- Input Offset Voltage
- Open Loop Gain
- Gain Bandwidth Product

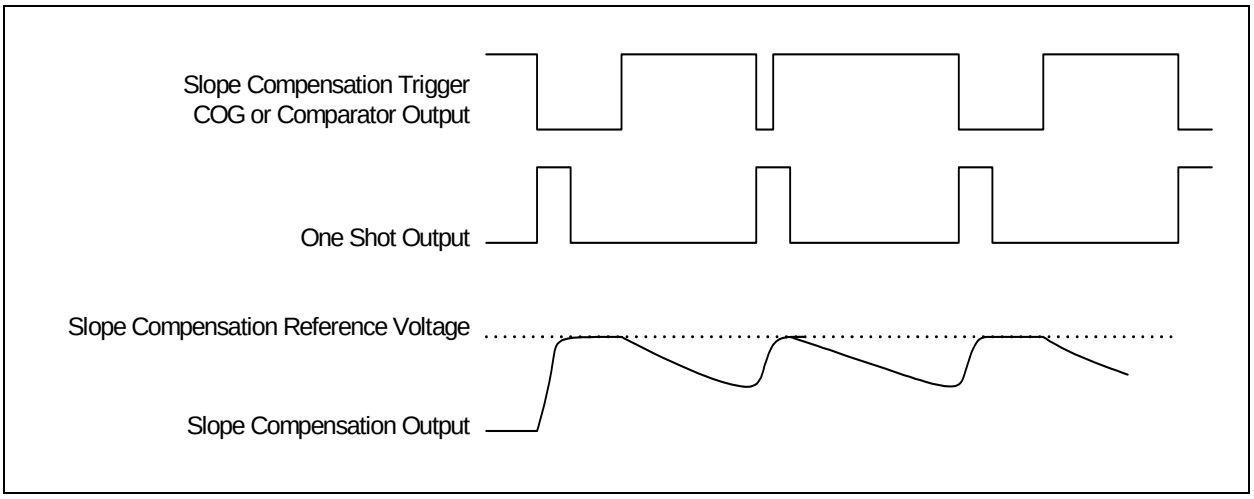
Common mode voltage range is the specified voltage range for the OPAx+ and OPAx- inputs, for which the OPA module will perform within its specifications. The OPA module is designed to operate with input voltages between V_{SS} and V_{DD}. Behavior for Common mode voltages greater than V_{DD} or below V_{SS} is not guaranteed.

Leakage current is a measure of the small source or sink currents on the OPAx+ and OPAx- inputs. To minimize the effect of leakage currents, the effective impedances connected to the OPAx+ and OPAx- inputs should be kept as small as possible and equal. Input offset voltage is a measure of the voltage difference between the OPAx+ and OPAx- inputs in a closed loop circuit with the OPA in its linear region. The offset voltage will appear as a DC offset in the output equal to the input offset voltage, multiplied by the gain of the circuit.

The input offset voltage is also affected by the Common mode voltage. The OPA is factory-calibrated to minimize the input offset voltage of the module. Open loop gain is the ratio of the output voltage to the differential input voltage (OPAx+) - (OPAx-). The gain is greatest at DC and falls off with frequency.

Gain Bandwidth Product or GBWP is the frequency at which the open loop gain falls off to 0 dB. The lower GBWP is optimized for systems requiring low-frequency response and low-power consumption.

FIGURE 17-2: SLOPE COMPENSATION TIMING DIAGRAM



17.2 Using the SC Module

The slope compensator input reference voltage should be set to the target circuit peak current sense voltage. The slope compensator output voltage starts at the input reference voltage and should fall at a rate less than half the target circuit current sense voltage rate of rise. Therefore, the compensator slope expressed as volts per µs can be computed as shown in Equation 17-2.

EQUATION 17-1: SC MODULE

$$\frac{V}{\mu s} \geq \frac{\frac{V_{REF}}{2}}{PWM\ Period\ (\mu s)}$$

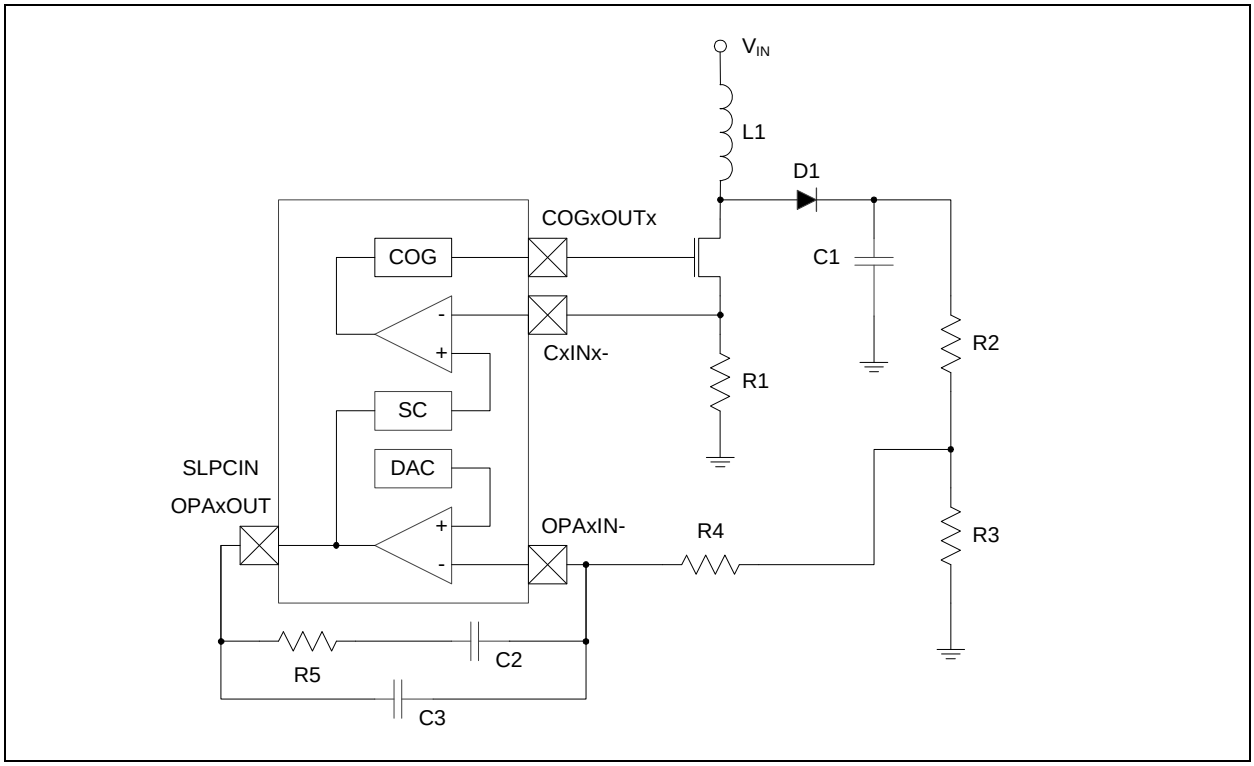
For example, when the circuit is using a 1Ω current sense resistor and the peak current is 1A, then the peak current expressed as a voltage (V_{REF}) is 1V. If your power supply is running at 1 MHz, then the period is 1 µs. Therefore, the desired slope is:

EQUATION 17-2: SLOPE COMPENSATION VOLTAGE

$$\frac{\frac{V_{REF}}{2}}{PWM\ Period\ (\mu s)} = \frac{\frac{1}{2}}{1\mu s} = 0.5V/\mu s$$

Note: The setting for 0.5V/µs is SCxISET<3:0> = 6 and SCxRNG = 0.

FIGURE 17-3: EXAMPLE SLOPE COMPENSATION CIRCUIT



17.3 Inputs

The SC module connects to the following inputs:

- COG1
- COG2
- Comparator C1
- Comparator C2

17.4 Outputs

The SC module connects to the following outputs:

- Comparator C1
- Comparator C2
- Op amp

17.5 Operation During Sleep

The SC module is unaffected by Sleep.

17.6 Effects of a Reset

The SC module resets to a disabled condition.

17.7 Register Definitions: Slope Compensation Control

REGISTER 17-1: SLPCCON0: SLOPE COMPENSATION CONTROL 0 REGISTER

R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0
SCxEN	—	—	SCxPOL	SCxTSS<1:0>		—	SCxINS
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 configuration bits

bit 7 **SCxEN:** Slope Compensation Enable bit
1 = Slope compensation is enabled
0 = Slope compensation is disabled

bit 6-5 **Unimplemented:** Read as ‘0’

bit 4 **SCxPOL:** Slope Compensation Input Polarity bit
1 = Signal is inverted polarity (active-low)
0 = Signal is normal polarity (active-high)

bit 3-2 **SCxTSS<1:0>:** Slope Compensation Timing Select bits
11 = C2OUT_sync
10 = C1OUT_sync
01 = COG1_output1
00 = COG1_output0

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **SCxINS:** Slope Compensation Input Select bit
1 = FVR_buffer1 is selected
0 = SLPC1IN pin is selected

REGISTER 17-2: SLPCCON1: SLOPE COMPENSATION CONTROL 1 REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	SCxRNG	SCxISET<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 configuration bits

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4 **SCxRNG:** Slope Compensator Range bit
1 = Range setting is SCxISET +1.0V/μs
0 = Range setting is SCxISET * 0.75/15 +0.2V/μs

bit 3-0 **SCxISET<3:0>:** Slope Compensator Current Sink Set bits
xxxxx = SC module Slope Selection

DECFSZ	Decrement f, Skip if 0
Syntax:	[*label*] DECFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) - 1 → (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2-cycle instruction.

GOTO	Unconditional Branch
Syntax:	[*label*] GOTO k
Operands:	$0 \le k \le 2047$
Operation:	k → PC<10:0> PCLATH<4:3> → PC<12:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 11-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a 2-cycle instruction.

INCF	Increment f
Syntax:	[*label*] INCF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

INCFSZ	Increment f, Skip if 0
Syntax:	[*label*] INCFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 → (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2-cycle instruction.

IORLW	Inclusive OR literal with W
Syntax:	[*label*] IORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are OR'ed with the 8-bit literal 'k'. The result is placed in the W register.

IORWF	Inclusive OR W with f
Syntax:	[*label*] IORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .OR. (f) → (destination)
Status Affected:	Z
Description:	Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

21.0 DEVELOPMENT SUPPORT

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

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

21.1 MPLAB X Integrated Development Environment Software

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

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

Feature-Rich Editor:

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

User-Friendly, Customizable Interface:

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

Project-Based Workspaces:

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

File History and Bug Tracking:

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

21.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

21.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

21.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

21.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

21.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

FIGURE 23-1: IDD TYPICAL, EC OSCILLATOR, MEDIUM-POWER MODE, PIC16HV753 ONLY

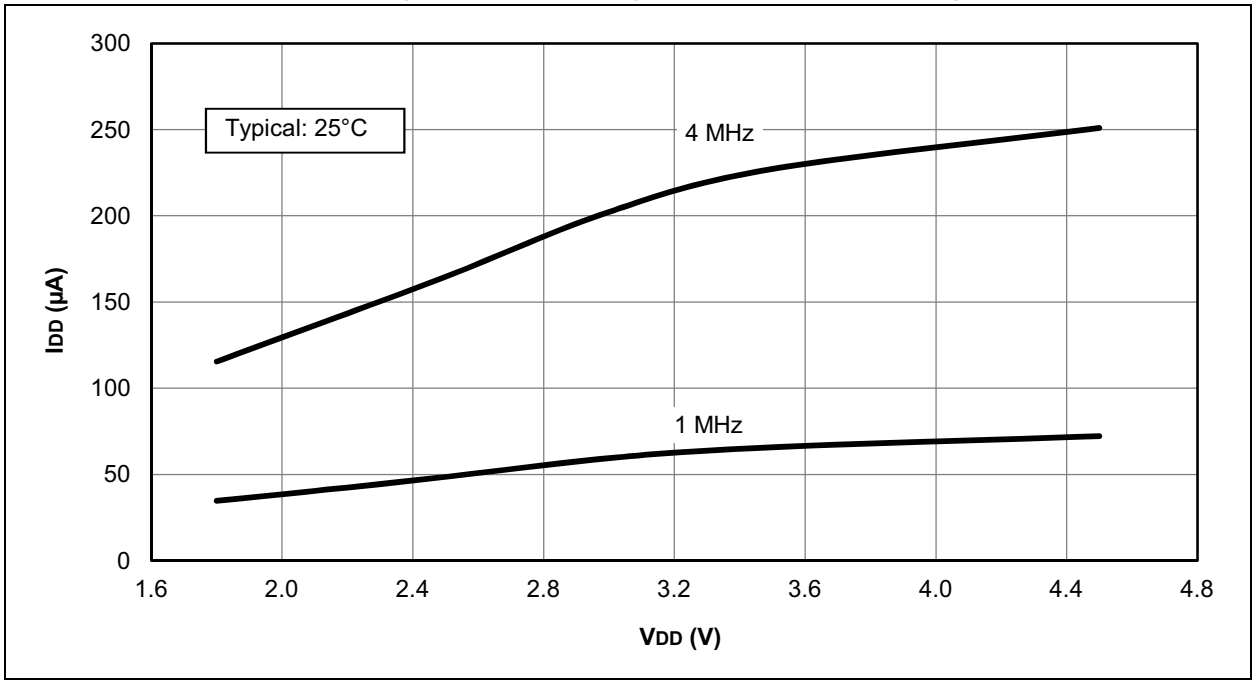


FIGURE 23-2: IDD MAXIMUM, EC OSCILLATOR, MEDIUM-POWER MODE, PIC16HV753 ONLY

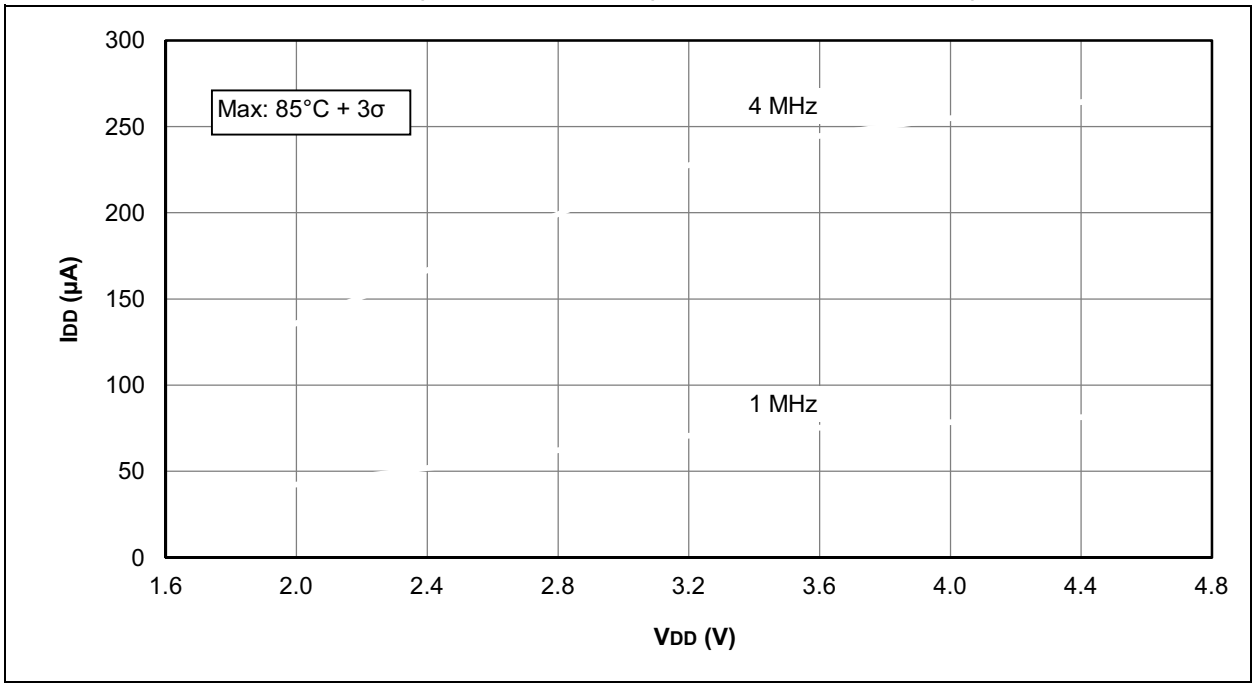


FIGURE 23-26: IPD, OP AMP, PIC16HV753 ONLY

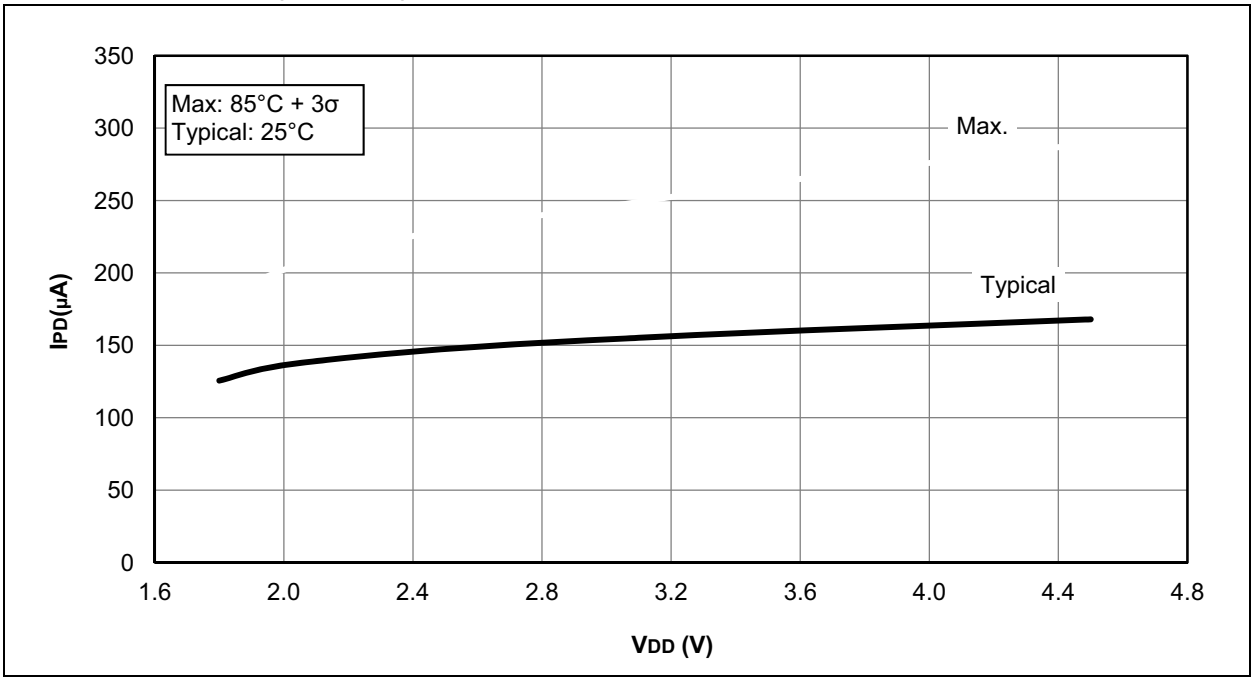
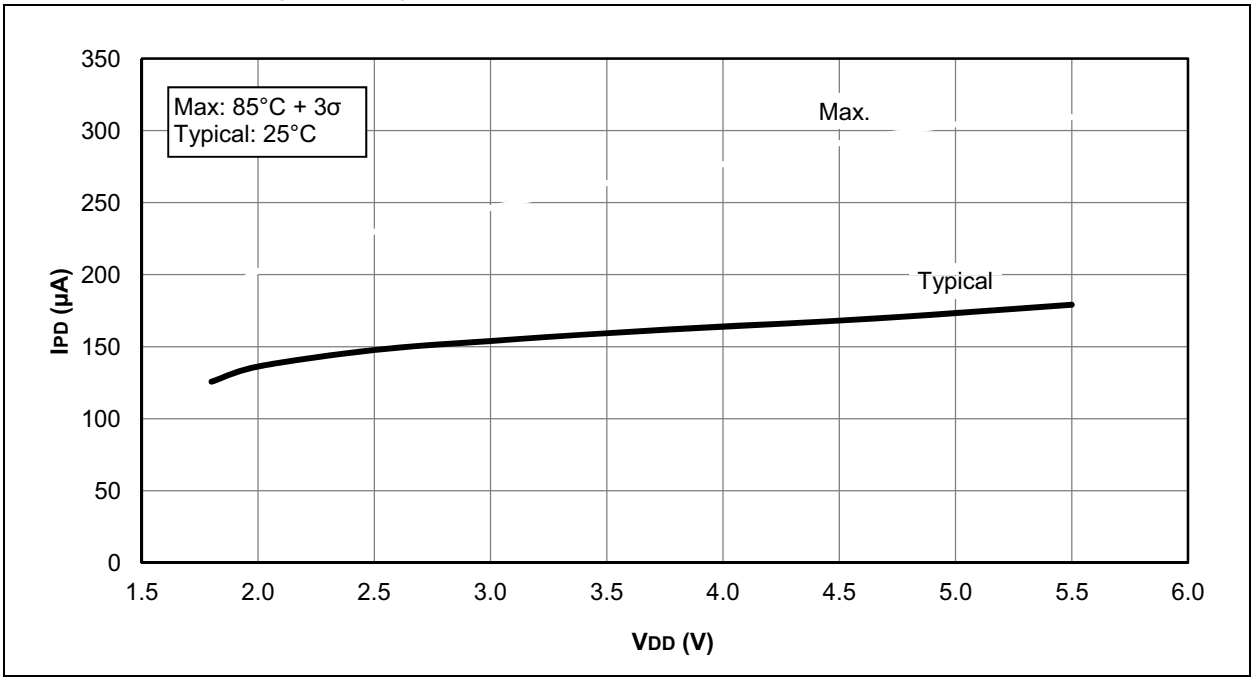
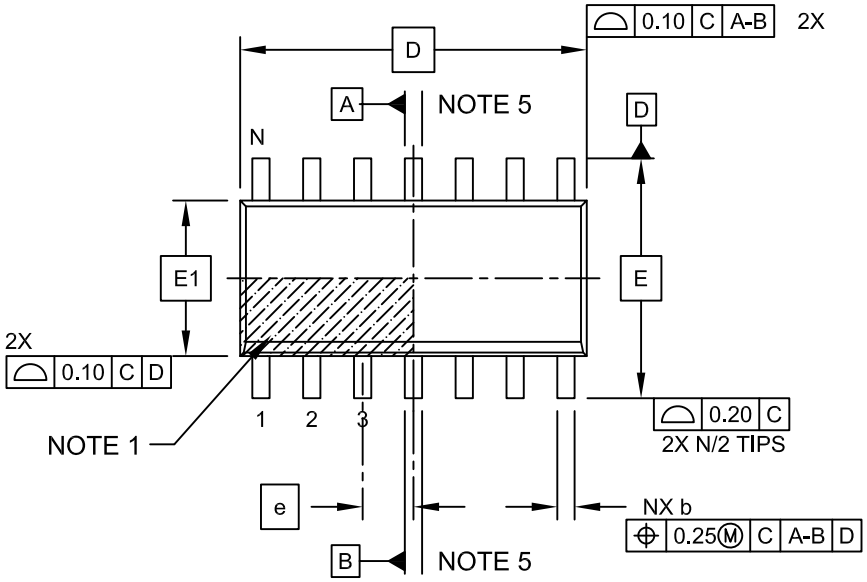


FIGURE 23-27: IPD, OP AMP, PIC16F753 ONLY

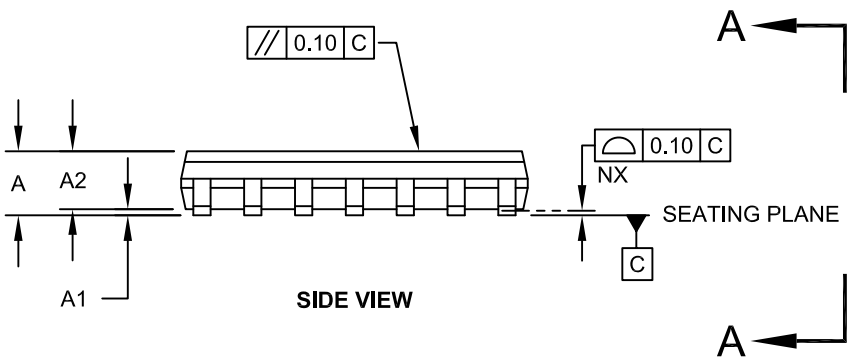


14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

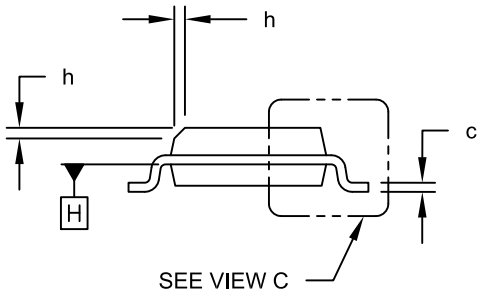
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW



VIEW A-A

Microchip Technology Drawing No. C04-065C Sheet 1 of 2