PIC12C67X

8-Pin, 8-Bit CMOS Microcontroller with A/D Converter and EEPROM Data Memory

Devices Included in this Data Sheet:

- PIC12C671
- PIC12C672
- PIC12CE673
- PIC12CE674

Note: Throughout this data sheet **PIC12C67X** refers to the PIC12C671, PIC12C672, PIC12CE673 and PIC12CE674. **PIC12CE67X** refers to PIC12CE673 and PIC12CE674.

High-Performance RISC CPU:

- Only 35 single word instructions to learn
- All instructions are single cycle (400 ns) except for program branches which are two-cycle
- Operating speed: DC - 10 MHz clock input
  DC - 400 ns instruction cycle

Device	Memory		
	Program	Data RAM	Data EEPROM
PIC12C671	1024 x 14	128 x 8	—
PIC12C672	2048 x 14	128 x 8	—
PIC12CE673	1024 x 14	128 x 8	16 x 8
PIC12CE674	2048 x 14	128 x 8	16 x 8

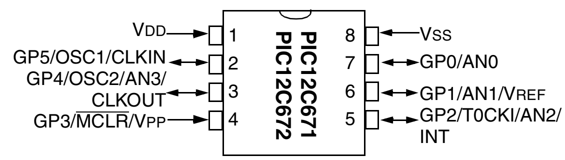
- 14-bit wide instructions
- 8-bit wide data path
- Interrupt capability
- Special function hardware registers
- 8-level deep hardware stack
- Direct, indirect and relative addressing modes for data and instructions

Peripheral Features:

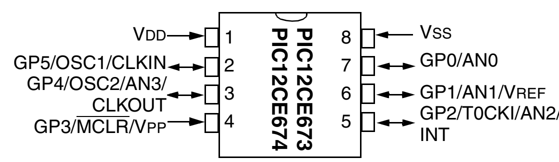
- Four-channel, 8-bit A/D converter
- 8-bit real time clock/counter (TMR0) with 8-bit programmable prescaler
- 1,000,000 erase/write cycle EEPROM data memory
- EEPROM data retention > 40 years

Pin Diagrams:

PDIP, SOIC, Windowed CERDIP



PDIP, Windowed CERDIP



Special Microcontroller Features:

- In-Circuit Serial Programming (ICSP™)
- Internal 4 MHz oscillator with programmable calibration
- Selectable clockout
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Interrupt-on-pin change (GP0, GP1, GP3)
- Internal pull-ups on I/O pins (GP0, GP1, GP3)
- Internal pull-up on $\overline{\text{MCLR}}$ pin
- Selectable oscillator options:
  - INTRC: Precision internal 4 MHz oscillator
  - EXTRC: External low-cost RC oscillator
  - XT: Standard crystal/resonator
  - HS: High speed crystal/resonator
  - LP: Power saving, low frequency crystal

CMOS Technology:

- Low-power, high-speed CMOS EPROM/EEPROM technology
- Fully static design
- Wide operating voltage range 2.5V to 5.5V
- Commercial, Industrial and Extended temperature ranges
- Low power consumption
  < 2 mA @ 5V, 4 MHz
  15 µA typical @ 3V, 32 kHz
  < 1 µA typical standby current

Table of Contents

To Our Valued Customers

Most Current Data Sheet

To automatically obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number. e.g., DS30000A is version A of document DS30000.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

Errata

An errata sheet may exist for current devices, describing minor operational differences (from the data sheet) and recommended workarounds. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 786-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Corrections to this Data Sheet

We constantly strive to improve the quality of all our products and documentation. We have spent a great deal of time to ensure that this document is correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error, please:

- Fill out and mail in the reader response form in the back of this data sheet.
- E-mail us at webmaster@microchip.com.

We appreciate your assistance in making this a better document.

1.0 GENERAL DESCRIPTION

The PIC12C67X devices are low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers with integrated analog-to-digital (A/D) converter and EEPROM data memory (EEPROM on PIC12CE67X versions only).

All PIC® microcontrollers employ an advanced RISC architecture. The PIC12C67X microcontrollers have enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with the separate 8-bit wide data. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches, which require two cycles. A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance.

PIC12C67X microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

The PIC12C67X devices have 128 bytes of RAM, 16 bytes of EEPROM data memory (PIC12CE67X only), 5 I/O pins and 1 input pin. In addition a timer/counter is available. Also a 4-channel, high-speed, 8-bit A/D is provided. The 8-bit resolution is ideally suited for applications requiring low-cost analog interface, (i.e., thermostat control, pressure sensing, etc.)

The PIC12C67X devices have special features to reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. The Power-On Reset (POR), Power-up Timer (PWRT), and Oscillator Start-up Timer (OST) eliminate the need for external reset circuitry. There are five oscillator configurations to choose from, including INTRC precision internal oscillator mode and the power-saving LP (Low Power) oscillator mode. Power-saving SLEEP mode, Watchdog Timer and code protection features improve system cost, power and reliability. The SLEEP (power-down) feature provides a power-saving mode. The user can wake-up the chip from SLEEP through several external and internal interrupts and resets.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lock-up.

A UV erasable windowed package version is ideal for code development, while the cost-effective One-Time-Programmable (OTP) version is suitable for production in any volume. The customer can take full advantage of Microchip's price leadership in OTP microcontrollers, while benefiting from the OTP's flexibility.

1.1 Applications

The PIC12C67X series fits perfectly in applications ranging from personal care appliances and security systems to low-power remote transmitters/receivers. The EPROM technology makes customizing application programs (transmitter codes, appliance settings, receiver frequencies, etc.) extremely fast and convenient, while the EEPROM data memory (PIC12CE67X only) technology allows for the changing of calibration factors and security codes. The small footprint packages, for through hole or surface mounting, make this microcontroller series perfect for applications with space limitations. Low-cost, low-power, high performance, ease of use and I/O flexibility make the PIC12C67X series very versatile even in areas where no microcontroller use has been considered before (i.e., timer functions, replacement of "glue" logic and PLD's in larger systems, coprocessor applications).

1.2 Family and Upward Compatibility

The PIC12C67X products are compatible with other members of the 14-bit PIC16CXXX families.

1.3 Development Support

The PIC12C67X devices are supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low-cost development programmer and a full-featured programmer. A "C" compiler and fuzzy logic support tools are also available.

TABLE 1-1: PIC12C67X & PIC12CE67X FAMILY OF DEVICES

		PIC12C671	PIC12LC671	PIC12C672	PIC12LC672	PIC12CE673	PIC12LCE673	PIC12CE674	PIC12LCE674
Clock	Maximum Frequency of Operation (MHz)	10	10	10	10	10	10	10	10
Memory	EPROM Program Memory	1024 x 14	1024 x 14	2048 x 14	2048 x 14	1024 x 14	1024 x 14	2048 x 14	2048 x 14
	RAM Data Memory (bytes)	128	128	128	128	128	128	128	128
Peripherals	EEPROM Data Memory (bytes)	——		—	—	16	16	16	16
	Timer Module(s)	TMR0	TMR0	TMR0	TMR0	TMR0	TMR0	TMR0	TMR0
	A/D Converter (8-bit) Channels	44		4	4	44		4	4
Features	Wake-up from SLEEP on pin change	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Interrupt Sources	44		4	4	44		4	4
	I/O Pins	5	5	5	5	5	5	5	5
	Input Pins	1	1	1	1	1	1	1	1
	Internal Pull-ups	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	In-Circuit Serial Programming	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Number of Instructions	35	35	35	35	35	35	35	35
	Voltage Range (Volts)	3.0V - 5.5V	2.5V - 5.5V	3.0V - 5.5V	2.5V - 5.5V	3.0V - 5.5V	2.5V - 5.5V	3.0V - 5.5V	2.5V - 5.5V
	Packages	8-pin DIP, JW, SOIC	8-pin DIP, JW, SOIC	8-pin DIP, JW, SOIC	8-pin DIP, JW, SOIC	8-pin DIP, JW	8-pin DIP, JW	8-pin DIP, JW	8-pin DIP, JW

All PIC12C67X devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.
All PIC12C67X devices use serial programming with data pin GP0 and clock pin GP1.

2.0 PIC12C67X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in the PIC12C67X Product Identification System section at the end of this data sheet. When placing orders, please use that page of the data sheet to specify the correct part number.

For example, the PIC12C67X device "type" is indicated in the device number:

1. **C**, as in PIC12**C**671. These devices have EPROM type memory and operate over the standard voltage range.
2. **LC**, as in PIC12**LC**671. These devices have EPROM type memory and operate over an extended voltage range.
3. **CE**, as in PIC12**CE**674. These devices have EPROM type memory, EEPROM data memory and operate over the standard voltage range.
4. **LCE**, as in PIC12**LCE**674. These devices have EPROM type memory, EEPROM data memory and operate over an extended voltage range.

2.1 UV Erasable Devices

The UV erasable version, offered in windowed package, is optimal for prototype development and pilot programs.

The UV erasable version can be erased and reprogrammed to any of the configuration modes. Microchip's PICSTART® Plus and PRO MATE® programmers both support the PIC12C67X. Third party programmers also are available; refer to the Microchip Third Party Guide for a list of sources.

Note:	Please note that erasing the device will also erase the pre-programmed internal calibration value for the internal oscillator. The calibration value must be saved prior to erasing the part.

2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers who need the flexibility for frequent code updates and small volume applications.

The OTP devices, packaged in plastic packages, permit the user to program them once. In addition to the program memory, the configuration bits must also be programmed.

2.3 Quick-Turn-Programming (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices, but with all EPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your local Microchip Technology sales office for more details.

2.4 Serialized Quick-Turn Programming (SQTP℠) Devices

Microchip offers a unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random, or sequential.

Serial programming allows each device to have a unique number which can serve as an entry-code, password, or ID number.

NOTES:

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC12C67X family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC12C67X uses a Harvard architecture, in which program and data are accessed from separate memories using separate buses. This improves bandwidth over traditional von Neumann architecture in which program and data are fetched from the same memory using the same bus. Separating program and data buses also allow instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 14-bits wide making it possible to have all single word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single instruction cycle. A two-stage pipeline overlaps fetch and execution of instructions (Example 3-1). Consequently, all instructions (35) execute in a single cycle (200 ns @ 20 MHz) except for program branches.

The table below lists program memory (EPROM), data memory (RAM), and non-volatile memory (EEPROM) for each PIC12C67X device.

Device	Program Memory	RAM Data Memory	EEPROM Data Memory
PIC12C671	1K x 14	128 x 8	—
PIC12C672	2K x 14	128 x 8	—
PIC12CE673	1K x 14	128 x 8	16x8
PIC12CE674	2K x 14	128 x 8	16x8

The PIC12C67X can directly or indirectly address its register files or data memory. All special function registers, including the program counter, are mapped in the data memory. The PIC12C67X has an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of ‘special optimal situations’ make programming with the PIC12C67X simple yet efficient. In addition, the learning curve is reduced significantly.

PIC12C67X devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between the data in the working register and any register file.

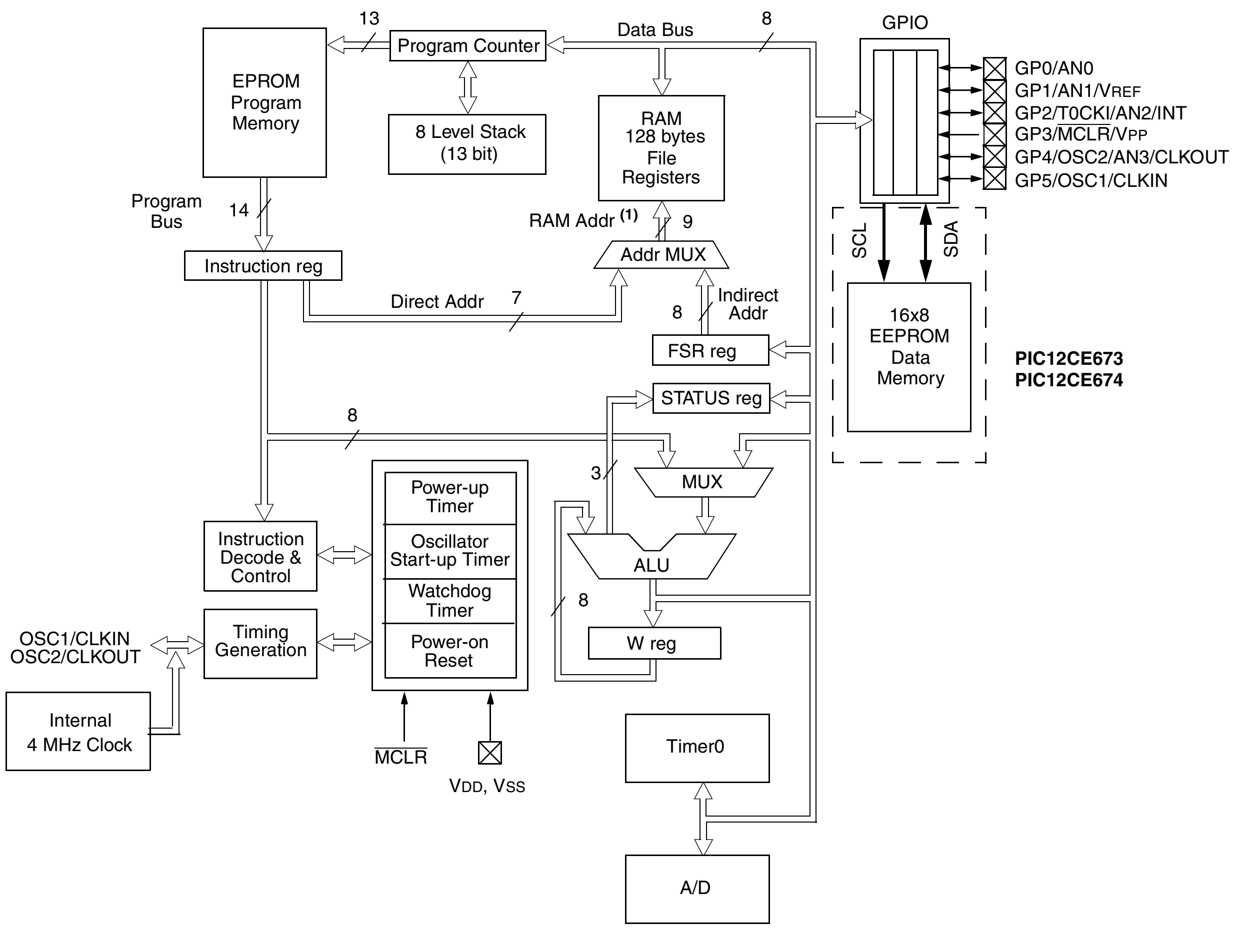
The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{borrow}}$ bit and a $\overline{\text{digit borrow}}$ out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

FIGURE 3-1: PIC12C67X BLOCK DIAGRAM

Device	Program Memory	Data Memory (RAM)	Non-Volatile Memory (EEPROM)
PIC12C671	1K x 14	128 x 8	—
PIC12C672	2K x 14	128 x 8	—
PIC12CE673	1K x 14	128 x 8	16 x 8
PIC12CE674	2K x 14	128 x 8	16 x 8



Note 1: Higher order bits are from the STATUS Register.

TABLE 3-1: PIC12C67X PINOUT DESCRIPTION

Name	DIP Pin #	I/O/P Type	Buffer Type	Description
GP0/AN0	7	I/O	TTL/ST	Bi-directional I/O port/serial programming data/analog input 0. Can be software programmed for internal weak pull-up and interrupt-on-pin change. This buffer is a Schmitt Trigger input when used in serial programming mode.
GP1/AN1/V_{REF}	6	I/O	TTL/ST	Bi-directional I/O port/serial programming clock/analog input 1/ voltage reference. Can be software programmed for internal weak pull-up and interrupt-on-pin change. This buffer is a Schmitt Trigger input when used in serial programming mode.
GP2/T0CKI/AN2/INT	5	I/O	ST	Bi-directional I/O port/analog input 2. Can be configured as T0CKI or external interrupt.
GP3/$\overline{MCLR}$/V_{PP}	4	I	TTL/ST	Input port/master clear (reset) input/programming voltage input. When configured as $\overline{MCLR}$, this pin is an active low reset to the device. Voltage on $\overline{MCLR}$/V_{PP} must not exceed V_{DD} during normal device operation. Can be software programmed for internal weak pull-up and interrupt-on-pin change. Weak pull-up always on if configured as $\overline{MCLR}$. This buffer is Schmitt Trigger when in $\overline{MCLR}$ mode.
GP4/OSC2/AN3/CLKOUT	3	I/O	TTL	Bi-directional I/O port/oscillator crystal output/analog input 3. Connections to crystal or resonator in crystal oscillator mode (HS, XT and LP modes only, GPIO in other modes). In EXTRC and INTRC modes, the pin output can be configured to CLK-OUT, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
GP5/OSC1/CLKIN	2	I/O	TTL/ST	Bi-directional IO port/oscillator crystal input/external clock source input (GPIO in INTRC mode only, OSC1 in all other oscillator modes). Schmitt trigger input for EXTRC oscillator mode.
V_{DD}	1	P	—	Positive supply for logic and I/O pins.
V_{SS}	8	P	—	Ground reference for logic and I/O pins.

Legend: I = input, O = output, I/O = input/output, P = power, — = not used, TTL = TTL input,
ST = Schmitt Trigger input.

3.1 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, and the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-2.

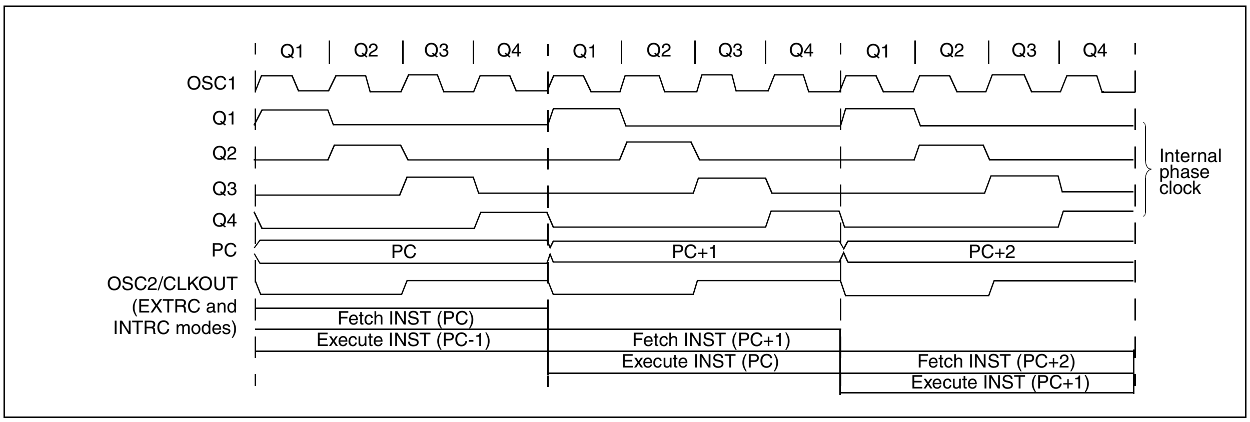
3.2 Instruction Flow/Pipelining

An “Instruction Cycle” consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle, while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (i.e., `GOTO`), then two cycles are required to complete the instruction (Example 3-1).

A fetch cycle begins with the program counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the “Instruction Register" (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW

	TCY0	TCY1	TCY2	TCY3	TCY4	TCY5
`1. MOVLW 55h`	Fetch 1	Execute 1				
`2. MOVWF GPIO`		Fetch 2	Execute 2			
`3. CALL SUB_1`			Fetch 3	Execute 3		
`4. BSF GPIO, BIT3 (Forced NOP)`				Fetch 4	Flush	
`5. Instruction @ address SUB_1`					Fetch SUB_1	Execute SUB_1

All instructions are single cycle, except for any program branches. These take two cycles since the fetched instruction is “flushed” from the pipeline while the new instruction is being fetched and then executed.

4.0 MEMORY ORGANIZATION

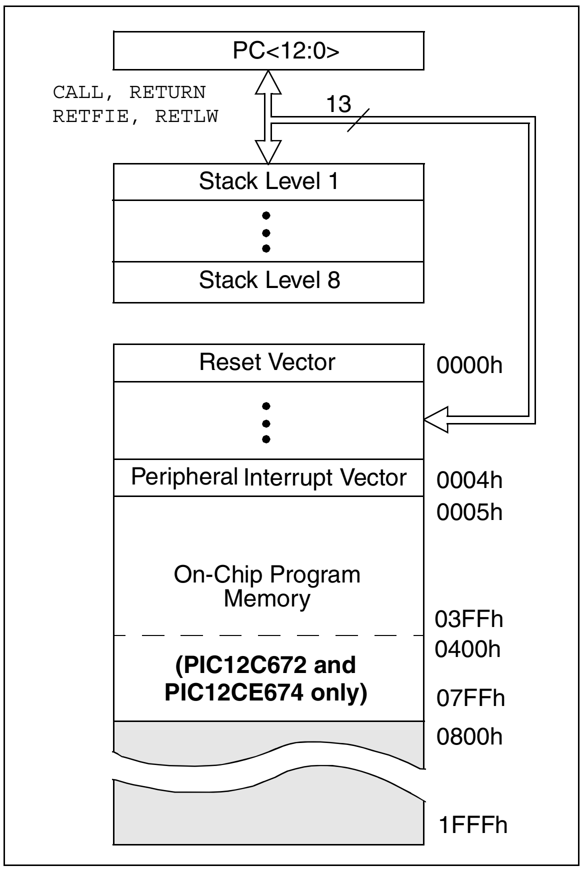
4.1 Program Memory Organization

The PIC12C67X has a 13-bit program counter capable of addressing an 8K x 14 program memory space.

For the PIC12C671 and the PIC12CE673, the first 1K x 14 (0000h-03FFh) is implemented.

For the PIC12C672 and the PIC12CE674, the first 2K x 14 (0000h-07FFh) is implemented. Accessing a location above the physically implemented address will cause a wraparound. The reset vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 4-1: PIC12C67X PROGRAM MEMORY MAP AND STACK



4.2 Data Memory Organization

The data memory is partitioned into two banks, which contain the General Purpose Registers and the Special Function Registers. Bit RP0 is the bank select bit.

RP0 (STATUS<5>) = 1 $\rightarrow$ Bank 1

RP0 (STATUS<5>) = 0 $\rightarrow$ Bank 0

Each Bank extends up to 7Fh (128 bytes). The lower locations of each Bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers implemented as static RAM. Both Bank 0 and Bank 1 contain Special Function Registers. Some "high use" Special Function Registers from Bank 0 are mirrored in Bank 1 for code reduction and quicker access.

Also note that F0h through FFh on the PIC12C67X is mapped into Bank 0 registers 70h-7Fh as common RAM.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly or indirectly through the File Select Register FSR (Section 4.5).

FIGURE 4-2: PIC12C67X REGISTER FILE MAP

File Address	Bank 0	Bank 1	File Address
00h	INDF(1)	INDF(1)	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	GPIO	TRIS	85h
06h			86h
07h			87h
08h			88h
09h			89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh			8Dh
0Eh		PCON	8Eh
0Fh		OSCCAL	8Fh
10h			90h
11h			91h
12h			92h
13h			93h
14h			94h
15h			95h
16h			96h
17h			97h
18h			98h
19h			99h
1Ah			9Ah
1Bh			9Bh
1Ch			9Ch
1Dh			9Dh
1Eh	ADRES		9Eh
1Fh	ADCON0	ADCON1	9Fh
20h	General Purpose Register	General Purpose Register	A0h–BFh
			C0h–EFh
70h–7Fh		Mapped in Bank 0	F0h–FFh

Unimplemented data memory locations, read as '0'. (shaded cells)

Note 1: Not a physical register.

4.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and Peripheral Modules for controlling the desired operation of the device. These registers are implemented as static RAM.

The Special Function Registers can be classified into two sets (core and peripheral). Those registers associated with the "core" functions are described in this section, and those related to the operation of the peripheral features are described in the section of that peripheral feature.

TABLE 4-1: PIC12C67X SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other Resets[3]
Bank 0											
00h[1]	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
01h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
02h[1]	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h[1]	STATUS	IRP[4]	RP1[4]	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h[1]	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05h	GPIO	SCL[5]	SDA[5]	GP5	GP4	GP3	GP2	GP1	GP0	11xx xxxx	11uu uuuu
06h	—	Unimplemented								—	—
07h	—	Unimplemented								—	—
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah[1,2]	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
0Bh[1]	INTCON	GIE PEI	E	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF	—	—	—	—	—	—	-0-- ----	-0-- ----
0Dh	—	Unimplemented								—	—
0Eh	—	Unimplemented								—	—
0Fh	—	Unimplemented								—	—
10h	—	Unimplemented								—	—
11h	—	Unimplemented								—	—
12h	—	Unimplemented								—	—
13h	—	Unimplemented								—	—
14h	—	Unimplemented								—	—
15h	—	Unimplemented								—	—
16h	—	Unimplemented								—	—
17h	—	Unimplemented								—	—
18h	—	Unimplemented								—	—
19h	—	Unimplemented								—	—
1Ah	—	Unimplemented								—	—
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	reserved	CHS1	CHS0	GO/$\overline{DONE}$	reserved	ADON	0000 0000	0000 0000

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

Note 1: These registers can be addressed from either bank.

2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

3: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.

4: The IRP and RP1 bits are reserved on the PIC12C67X; always maintain these bits clear.

5: The SCL (GP7) and SDA (GP6) bits are unimplemented on the PIC12C671/672 and read as '0'.

TABLE 4-1: PIC12C67X SPECIAL FUNCTION REGISTER SUMMARY (CONT.)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other Resets(3)
Bank 1											
80h(1)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
81h	OPTION	$\overline{\text{GPPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h(1)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
83h(1)	STATUS	IRP(4)	RP1(4)	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	000q quuu
84h(1)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85h	TRIS	—	—	GPIO Data Direction Register						--11 1111	--11 1111
86h	—	Unimplemented								—	—
87h	—	Unimplemented								—	—
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah(1,2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the PC					---0 0000	---0 0000
8Bh(1)	INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 000x	0000 000u
8Ch	PIE1	—	ADIE	—	—	—	—	—	—	-0-- ----	-0-- ----
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	—	—	—	—	$\overline{\text{POR}}$	—	---- --0-	---- --u-
8Fh	OSCCAL	CAL3	CAL2	CAL1	CAL0	CALFST	CALSLW	—	—	0111 00--	uuuu uu--
90h	—	Unimplemented								—	—
91h	—	Unimplemented								—	—
92h	—	Unimplemented								—	—
93h	—	Unimplemented								—	—
94h	—	Unimplemented								—	—
95h	—	Unimplemented								—	—
96h	—	Unimplemented								—	—
97h	—	Unimplemented								—	—
98h	—	Unimplemented								—	—
99h	—	Unimplemented								—	—
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	—	Unimplemented								—	—
9Dh	—	Unimplemented								—	—
9Eh	—	Unimplemented								—	—
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	---- -000

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

Note 1: These registers can be addressed from either bank.

2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

3: Other (non power-up) resets include external reset through $\overline{\text{MCLR}}$ and Watchdog Timer Reset.

4: The IRP and RP1 bits are reserved on the PIC12C67X; always maintain these bits clear.

5: The SCL (GP7) and SDA (GP6) bits are unimplemented on the PIC12C671/672 and read as '0'.

4.2.2.1 STATUS REGISTER

The STATUS Register, shown in Register 4-1, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The STATUS Register can be the destination for any instruction, as with any other register. If the STATUS Register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are not writable. Therefore, the result of an instruction with the STATUS Register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS Register as `000u u1uu` (where `u` = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS Register, because these instructions do not affect the Z, C or DC bits from the STATUS Register. For other instructions, not affecting any status bits, see the "Instruction Set Summary."

> **Note 1:** Bits IRP and RP1 (STATUS<7:6>) are not used by the PIC12C67X and should be maintained clear. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.
>
> **2:** The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

REGISTER 4-1: STATUS REGISTER (ADDRESS 03h, 83h)

Reserved	Reserved	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
- n = Value at POR reset

bit 7: **IRP:** Register Bank Select bit (used for indirect addressing)
`1` = Bank 2, 3 (100h - 1FFh)
`0` = Bank 0, 1 (00h - FFh)
The IRP bit is reserved; always maintain this bit clear.

bit 6-5: **RP<1:0>:** Register Bank Select bits (used for direct addressing)
`11` = Bank 3 (180h - 1FFh)
`10` = Bank 2 (100h - 17Fh)
`01` = Bank 1 (80h - FFh)
`00` = Bank 0 (00h - 7Fh)
Each bank is 128 bytes. The RP1 bit is reserved; always maintain this bit clear.

bit 4: **$\overline{\text{TO}}$:** Time-out bit
`1` = After power-up, `CLRWDT` instruction, or `SLEEP` instruction
`0` = A WDT time-out occurred

bit 3: **$\overline{\text{PD}}$:** Power-down bit
`1` = After power-up or by the `CLRWDT` instruction
`0` = By execution of the `SLEEP` instruction

bit 2: **Z:** Zero bit
`1` = The result of an arithmetic or logic operation is zero
`0` = The result of an arithmetic or logic operation is not zero

bit 1: **DC:** Digit Carry/$\overline{\text{borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions) (for $\overline{\text{borrow}}$ the polarity is reversed)
`1` = A carry-out from the 4th low order bit of the result occurred
`0` = No carry-out from the 4th low order bit of the result

bit 0: **C:** Carry/$\overline{\text{borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
`1` = A carry-out from the most significant bit of the result occurred
`0` = No carry-out from the most significant bit of the result occurred

Note: For $\overline{\text{borrow}}$ the polarity is reversed. A subtraction is executed by adding the two’s complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low order bit of the source register.

4.2.2.2 OPTION REGISTER

The OPTION Register is a readable and writable register, which contains various control bits to configure the TMR0/WDT prescaler, the External INT Interrupt, TMR0 and the weak pull-ups on GPIO.

Note:	To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer by setting bit PSA (OPTION<3>).

REGISTER 4-2: OPTION REGISTER (ADDRESS 81h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{GPPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
- n = Value at POR reset

bit 7: **$\overline{\text{GPPU}}$:** Weak Pull-up Enable
1 = Weak pull-ups disabled
0 = Weak pull-ups enabled (GP0, GP1, GP3)

bit 6: **INTEDG:** Interrupt Edge
1 = Interrupt on rising edge of GP2/T0CKI/AN2/INT pin
0 = Interrupt on falling edge of GP2/T0CKI/AN2/INT pin

bit 5: **T0CS:** TMR0 Clock Source Select bit
1 = Transition on GP2/T0CKI/AN2/INT pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4: **T0SE:** TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on GP2/T0CKI/AN2/INT pin
0 = Increment on low-to-high transition on GP2/T0CKI/AN2/INT pin

bit 3: **PSA:** Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0: **PS<2:0>**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

4.2.2.3 INTCON REGISTER

The INTCON Register is a readable and writable register, which contains various enable and flag bits for the TMR0 Register overflow, GPIO port change and external GP2/INT pin interrupts.

Note:	Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>).

REGISTER 4-3: INTCON REGISTER (ADDRESS 0Bh, 8Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
- n = Value at POR reset

bit 7: **GIE:** Global Interrupt Enable bit
1 = Enables all un-masked interrupts
0 = Disables all interrupts

bit 6: **PEIE:** Peripheral Interrupt Enable bit
1 = Enables all un-masked peripheral interrupts
0 = Disables all peripheral interrupts

bit 5: **T0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4: **INTE:** INT External Interrupt Enable bit
1 = Enables the external interrupt on GP2/INT/T0CKI/AN2 pin
0 = Disables the external interrupt on GP2/INT/T0CKI/AN2 pin

bit 3: **GPIE:** GPIO Interrupt on Change Enable bit
1 = Enables the GPIO Interrupt on Change
0 = Disables the GPIO Interrupt on Change

bit 2: **T0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1: **INTF:** INT External Interrupt Flag bit
1 = The external interrupt on GP2/INT/T0CKI/AN2 pin occurred (must be cleared in software)
0 = The external interrupt on GP2/INT/T0CKI/AN2 pin did not occur

bit 0: **GPIF:** GPIO Interrupt on Change Flag bit
1 = GP0, GP1 or GP3 pins changed state (must be cleared in software)
0 = Neither GP0, GP1 nor GP3 pins have changed state

4.2.2.4 PIE1 REGISTER

This register contains the individual enable bits for the Peripheral interrupts.

Note:	Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

REGISTER 4-4: PIE1 REGISTER (ADDRESS 8Ch)

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	ADIE	—	—	—	—	—	—
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
- n = Value at POR reset

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

bit 6: **ADIE:** A/D Converter Interrupt Enable bit
`1` = Enables the A/D interrupt
`0` = Disables the A/D interrupt

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

4.2.2.5 PIR1 REGISTER

This register contains the individual flag bits for the Peripheral interrupts.

Note:	Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 4-5: PIR1 REGISTER (ADDRESS 0Ch)

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	ADIF	—	—	—	—	—	—
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
- n = Value at POR reset

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

bit 6: **ADIF:** A/D Converter Interrupt Flag bit
1 = An A/D conversion completed (must be cleared in software)
0 = The A/D conversion is not complete

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

4.2.2.6 PCON REGISTER

The Power Control (PCON) Register contains a flag bit to allow differentiation between a Power-on Reset (POR), an external $\overline{\text{MCLR}}$ Reset and a WDT Reset.

REGISTER 4-6: PCON REGISTER (ADDRESS 8Eh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0
—	—	—	—	—	—	$\overline{\text{POR}}$	—
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
- n = Value at POR reset

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

bit 1: **$\overline{\text{POR}}$:** Power-on Reset Status bit
`1` = No Power-on Reset occurred
`0` = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

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

4.2.2.7 OSCCAL REGISTER

The Oscillator Calibration (OSCCAL) Register is used to calibrate the internal 4 MHz oscillator. It contains four bits for fine calibration and two other bits to either increase or decrease frequency.

REGISTER 4-7: OSCCAL REGISTER (ADDRESS 8Fh)

R/W-0	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0	U-0	U-0
CAL3	CAL2	CAL1	CAL0	CALFST	CALSLW	—	—
bit7							bit0

R =	Readable bit
W =	Writable bit
U =	Unimplemented bit, read as ‘0’
- n =	Value at POR reset

bit 7-4: **CAL<3:0>:** Fine Calibration

bit 3: **CALFST:** Calibration Fast
1 = Increase frequency
0 = No change

bit 2: **CALSLW:** Calibration Slow
1 = Decrease frequency
0 = No change

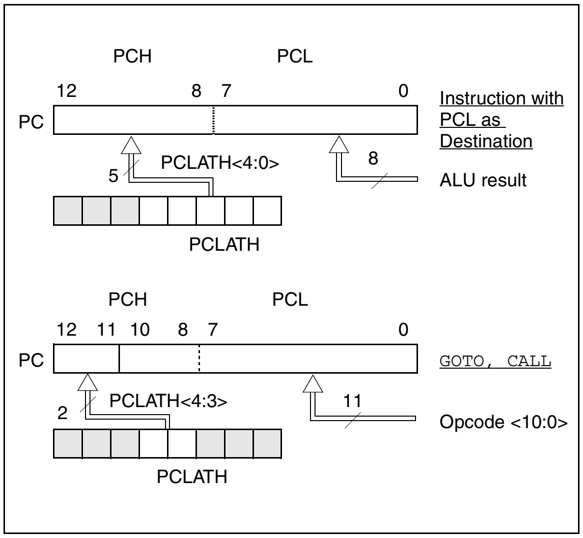
bit 1-0: **Unimplemented:** Read as ’0’

Note: If CALFST = 1 and CALSLW = 1, CALFST has precedence.

4.3 PCL and PCLATH

The Program Counter (PC) is 13-bits wide. The low byte comes from the PCL Register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any reset, the PC is cleared. Figure 4-3 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in the figure shows how the PC is loaded during a `CALL` or `GOTO` instruction (PCLATH<4:3> → PCH).

FIGURE 4-3: LOADING OF PC IN DIFFERENT SITUATIONS



4.3.1 COMPUTED GOTO

A Computed GOTO is accomplished by adding an offset to the program counter (`ADDWF PCL`). When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the application note *"Implementing a Table Read"* (AN556).

4.3.2 STACK

The PIC12C67X family has an 8-level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a `CALL` instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a `RETURN`, `RETLW` or a `RETFIE` instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1:	There are no status bits to indicate stack overflow or stack underflow conditions.
2:	There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the `CALL`, `RETURN`, `RETLW`, and `RETFIE` instructions, or the vectoring to an interrupt address.

4.4 Program Memory Paging

The PIC12C67X ignores both paging bits PCLATH<4:3>, which are used to access program memory when more than one page is available. The use of PCLATH<4:3> as general purpose read/write bits for the PIC12C67X is not recommended since this may affect upward compatibility with future products.

4.5 Indirect Addressing, INDF and FSR Registers

The INDF Register is not a physical register. Addressing the INDF Register will cause indirect addressing.

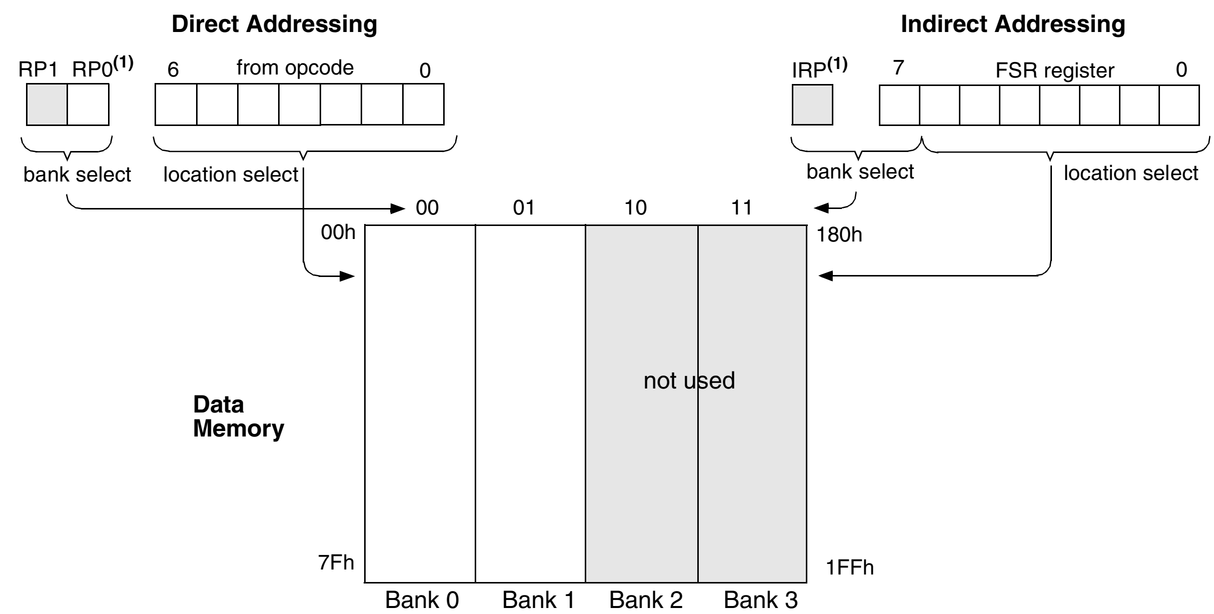
Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF Register itself indirectly (FSR = '0') will read 00h. Writing to the INDF Register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR Register and the IRP bit (STATUS<7>), as shown in Figure 4-4. However, IRP is not used in the PIC12C67X.

A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 4-1.

EXAMPLE 4-1: INDIRECT ADDRESSING

```
         movlw  0x20   ;initialize pointer
         movwf  FSR    ;to RAM
NEXT     clrf   INDF   ;clear INDF register
         incf   FSR,F  ;inc pointer
         btfss  FSR,4  ;all done?
         goto   NEXT   ;no clear next
CONTINUE
         :              ;yes continue
```

FIGURE 4-4: DIRECT/INDIRECT ADDRESSING



For register file map detail see Figure 4-2.

Note 1: The RP1 and IRP bits are reserved; always maintain these bits clear.

NOTES:

5.0 I/O PORT

As with any other register, the I/O register can be written and read under program control. However, read instructions (i.e., `MOVF GPIO,W`) always read the I/O pins independent of the pin's input/output modes. On RESET, all I/O ports are defined as input (inputs are at hi-impedance), since the I/O control registers are all set.

5.1 GPIO

GPIO is an 8-bit I/O register. Only the low order 6 bits are used (GP<5:0>). Bits 6 and 7 (SDA and SCL, respectively) are used by the EEPROM peripheral on the PIC 12CE673/674. Refer to Section 6.0 and Appendix B for use of SDA and SCL. Please note that GP3 is an input only pin. The configuration word can set several I/O's to alternate functions. When acting as alternate functions, the pins will read as '0' during port read. Pins GP0, GP1 and GP3 can be configured with weak pull-ups and also with interrupt-on-change. The interrupt on change and weak pull-up functions are not pin selectable. If pin 4, (GP3), is configured as $\overline{\text{MCLR}}$, a weak pull-up is always on. Interrupt-on-change for this pin is not set and GP3 will read as '0'. Interrupt-on-change is enabled by setting bit GPIE, INTCON<3>. Note that external oscillator use overrides the GPIO functions on GP4 and GP5.

5.2 TRIS Register

This register controls the data direction for GPIO. A '1' from a TRIS Register bit puts the corresponding output driver in a hi-impedance mode. A '0' puts the contents of the output data latch on the selected pins, enabling the output buffer. The exceptions are GP3, which is input only and its TRIS bit will always read as '1', while GP6 and GP7 TRIS bits will read as '0'.

Note:	A read of the ports reads the pins, not the output data latches. That is, if an output driver on a pin is enabled and driven high, but the external system is holding it low, a read of the port will indicate that the pin is low.

Upon reset, the TRIS Register is all '1's, making all pins inputs.

TRIS for pins GP4 and GP5 is forced to a '1' where appropriate. Writes to TRIS <5:4> will have an effect in EXTRC and INTRC oscillator modes only. When GP4 is configured as CLKOUT, changes to TRIS<4> will have no effect.

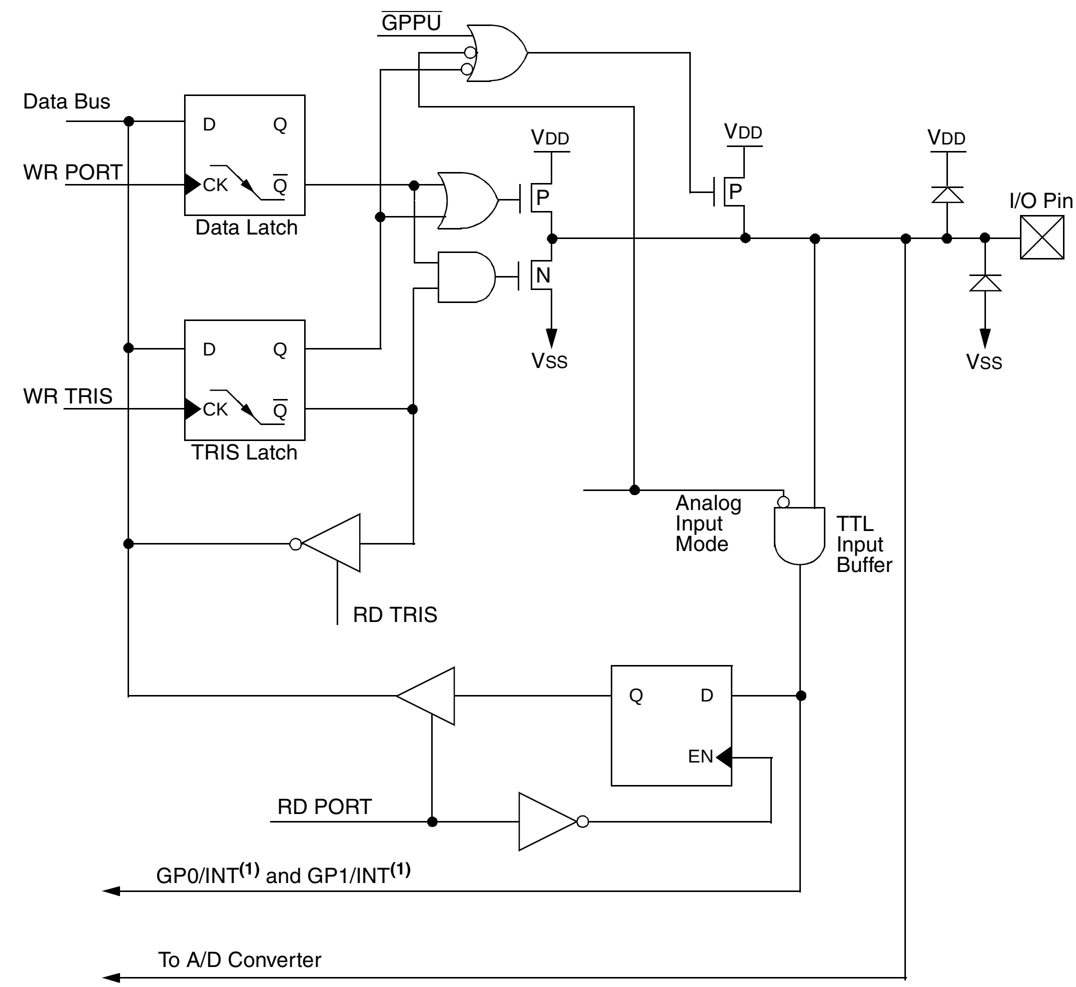
5.3 I/O Interfacing

The equivalent circuit for an I/O port pin is shown in Figure 5-1 through Figure 5-5. All port pins, except GP3, which is input only, may be used for both input and output operations. For input operations, these ports are non-latching. Any input must be present until read by an input instruction (i.e., `MOVF GPIO,W`). The outputs are latched and remain unchanged until the output latch is rewritten. To use a port pin as output, the corresponding direction control bit in TRIS must be cleared (= 0). For use as an input, the corresponding TRIS bit must be set. Any I/O pin (except GP3) can be programmed individually as input or output.

Port pins GP6 (SDA) and GP7 (SCL) are used for the serial EEPROM interface on the PIC12CE673/674. These port pins are not available externally on the package. Users should avoid writing to pins GP6 (SDA) and GP7 (SCL) when not communicating with the serial EEPROM memory. Please see Section 6.0, EEPROM Peripheral Operation, for information on serial EEPROM communication.

Note:	On a Power-on Reset, GP0, GP1, GP2 and GP4 are configured as analog inputs and read as '0'.

FIGURE 5-1: BLOCK DIAGRAM OF GP0/AN0 AND GP1/AN1/VREF PIN



Note 1: Wake-up on pin change interrupts for GP0 and GP1.

FIGURE 5-2: BLOCK DIAGRAM OF GP2/T0CKI/AN2/INT PIN

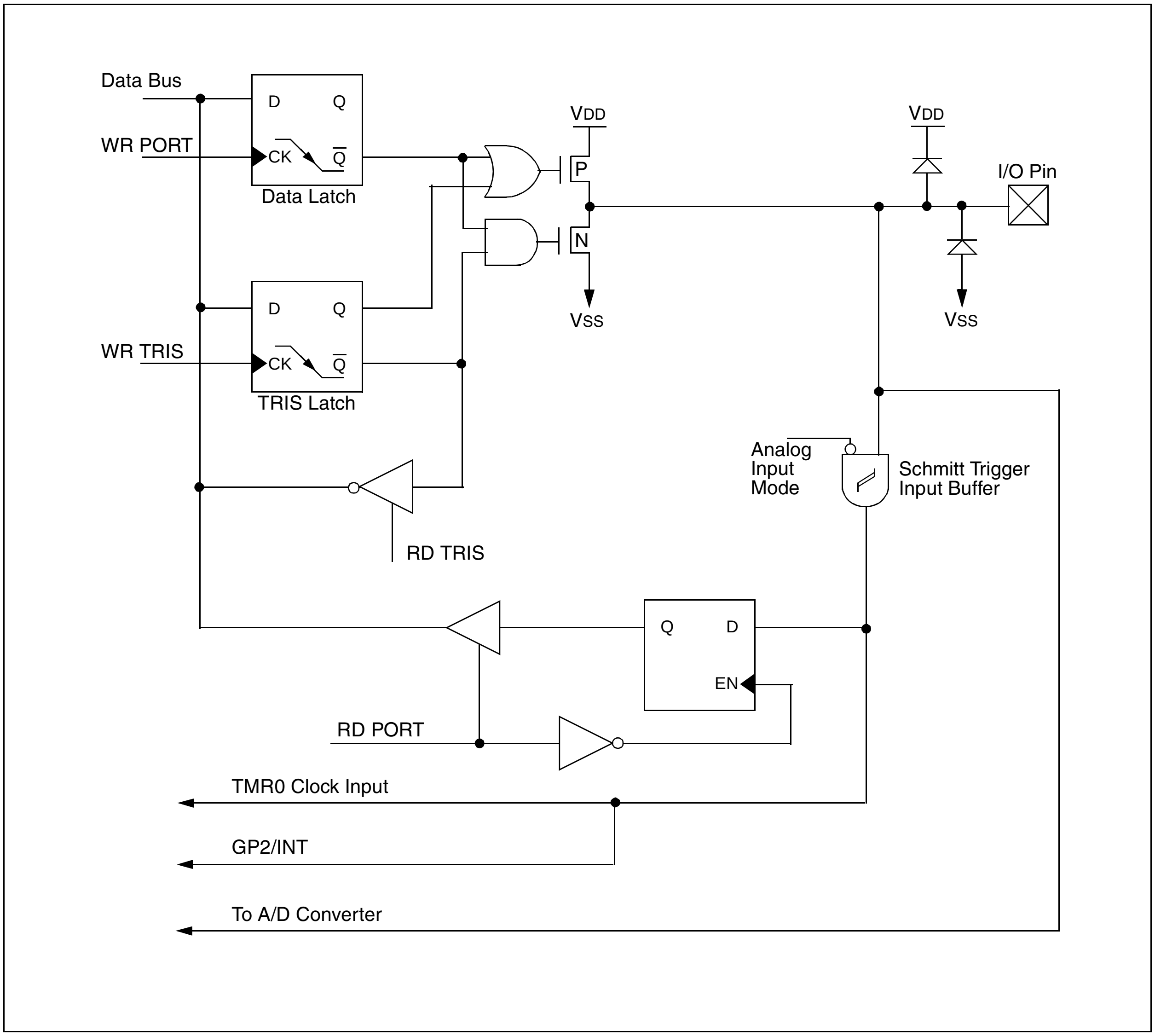
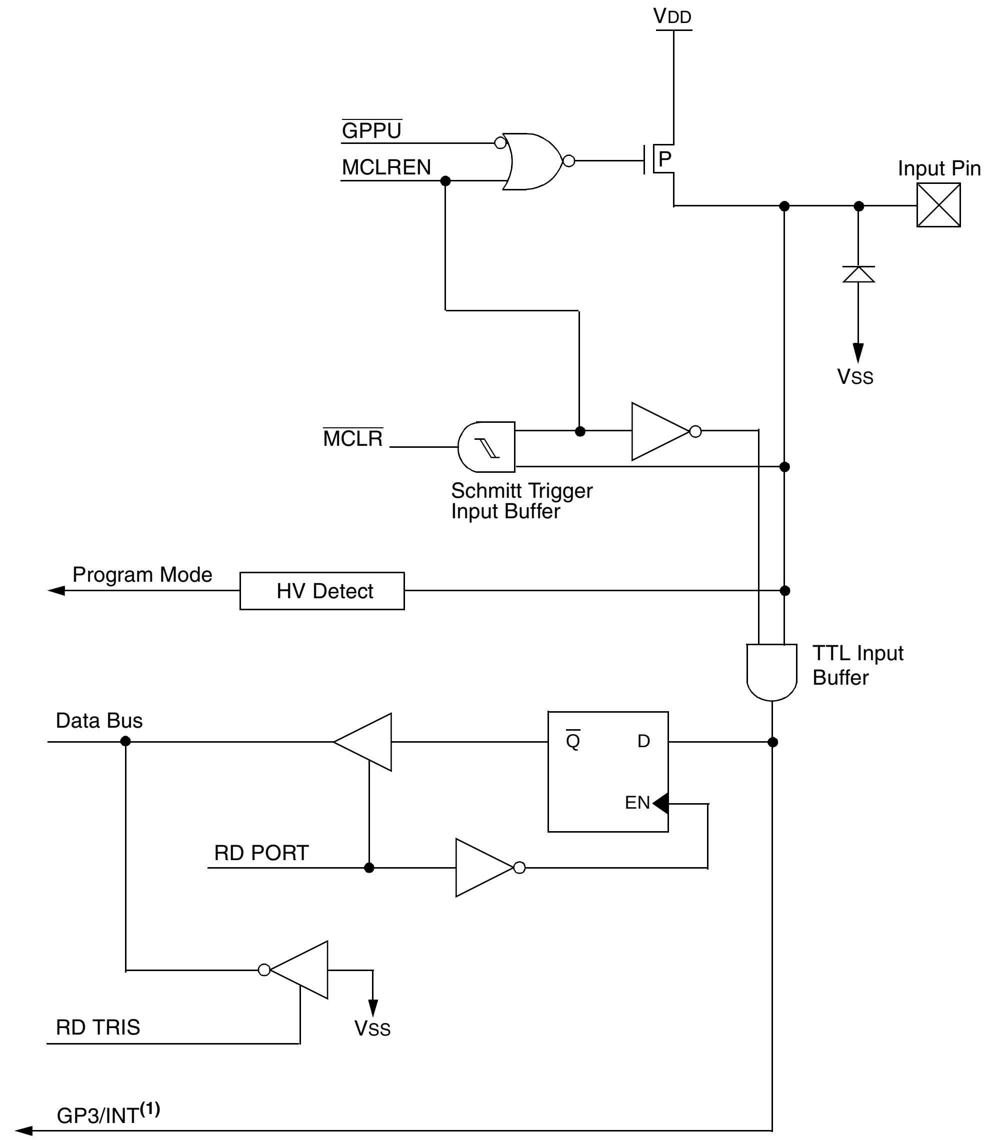


FIGURE 5-3: BLOCK DIAGRAM OF GP3/$\overline{\text{MCLR}}$/VPP PIN



Note 1: Wake-up on pin change interrupt for GP3.

FIGURE 5-4: BLOCK DIAGRAM OF GP4/OSC2/AN3/CLKOUT PIN

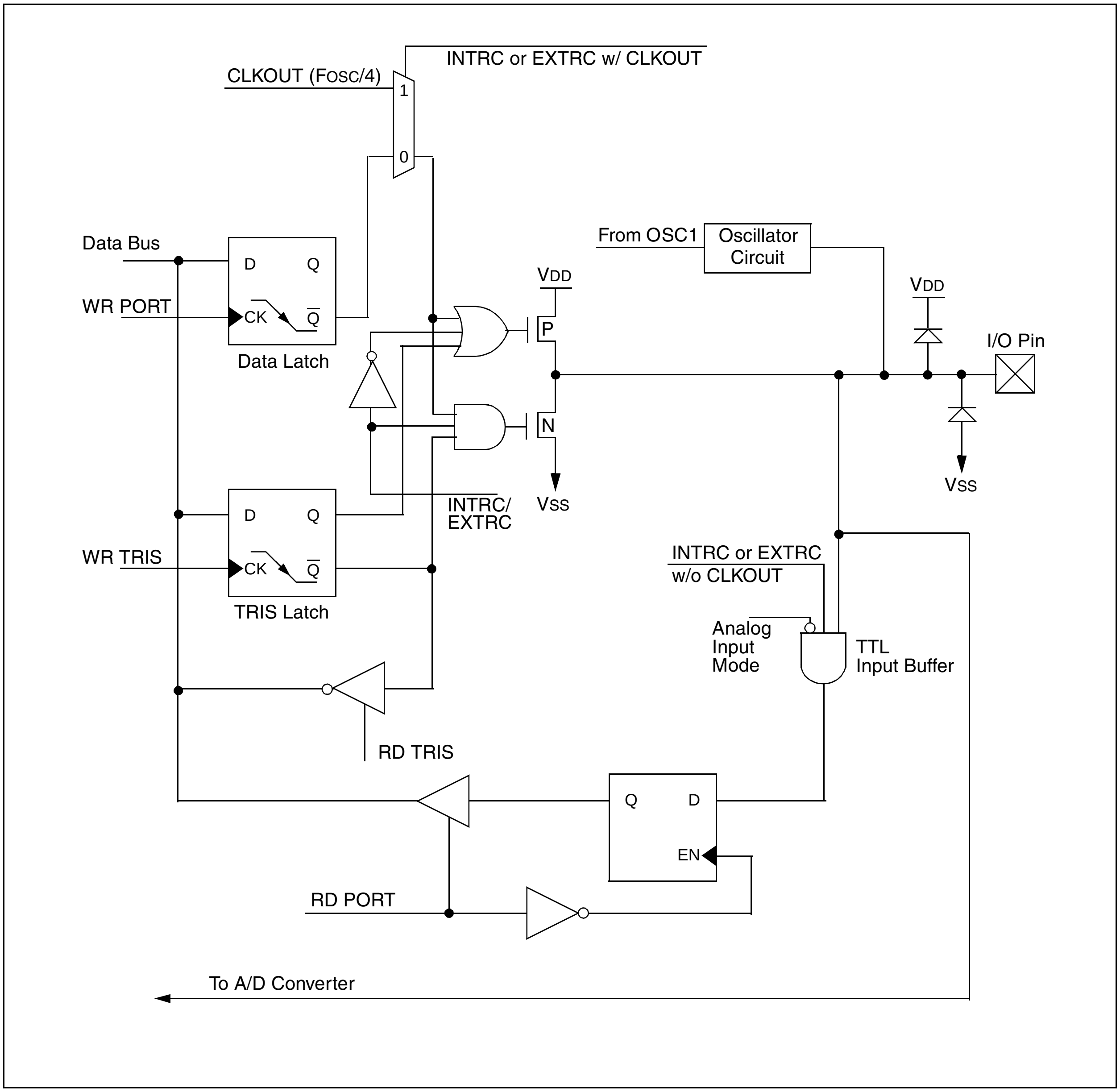


FIGURE 5-5: BLOCK DIAGRAM OF GP5/OSC1/CLKIN PIN

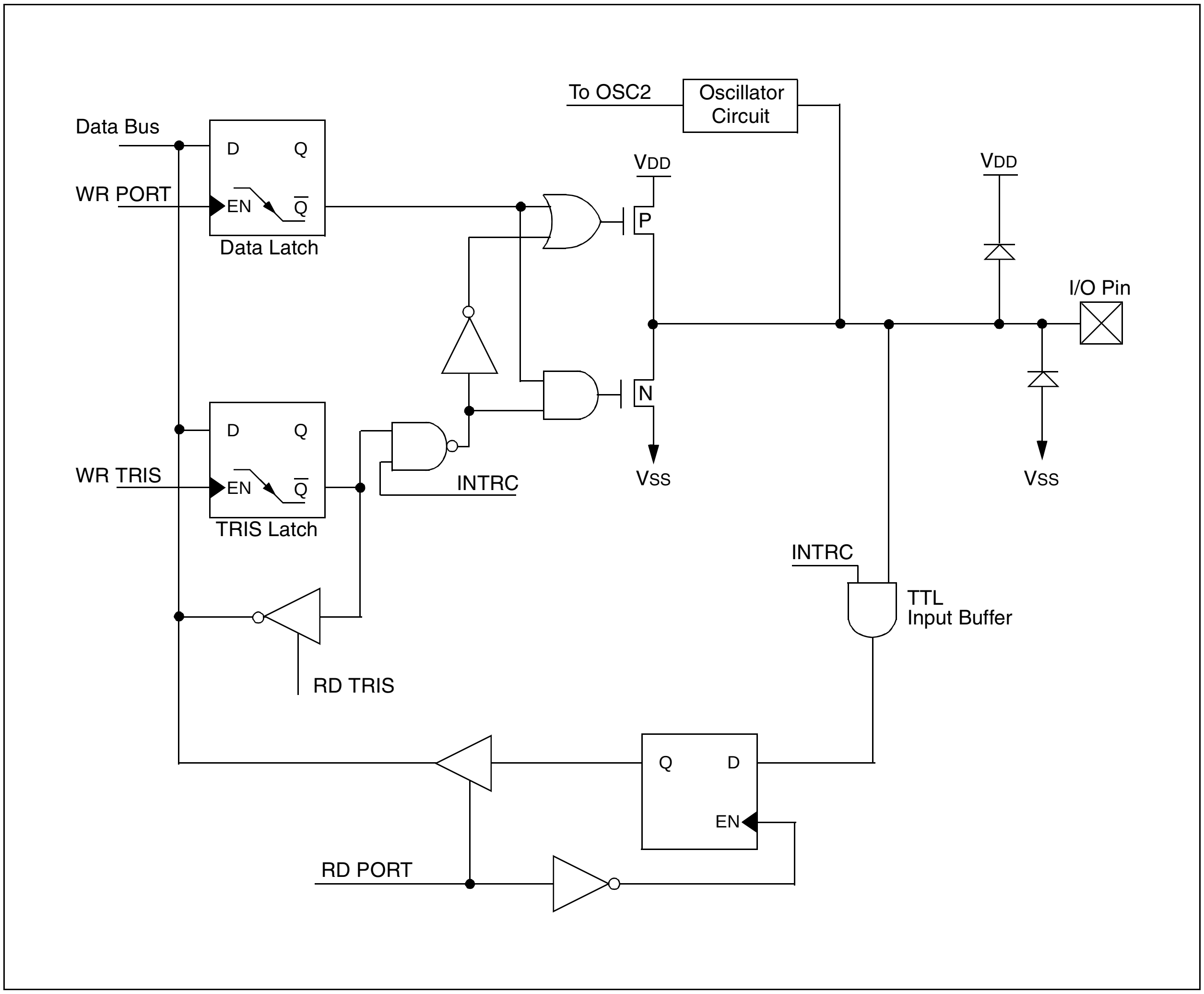


TABLE 5-1: SUMMARY OF PORT REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other Resets
85h	TRIS	—	—	GPIO Data Direction Register						--11 1111	--11 1111
81h	OPTION	$\overline{\text{GPPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
03h	STATUS	IRP(1)	RP1(1)	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	000q quuu
05h	GPIO	SCL(2)	SDA(2)	GP5	GP4	GP3	GP2	GP1	GP0	11xx xxxx	11uu uuuu

Legend: Shaded cells not used by Port Registers, read as '0', — = unimplemented, read as '0', x = unknown, u = unchanged, q = see tables in Section 9.4 for possible values.

Note 1: The IRP and RP1 bits are reserved on the PIC12C67X; always maintain these bits clear.

2: The SCL and SDA bits are unimplemented on the PIC12C671 and PIC12C672.

5.4 I/O Programming Considerations

5.4.1 BI-DIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. The BCF and BSF instructions, for example, read the register into the CPU, execute the bit operation and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a BSF operation on bit5 of GPIO will cause all eight bits of GPIO to be read into the CPU. Then the BSF operation takes place on bit5 and GPIO is written to the output latches. If another bit of GPIO is used as a bi-directional I/O pin (i.e., bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit0 is switched to an output, the content of the data latch may now be unknown.

Reading the port register reads the values of the port pins. Writing to the port register writes the value to the port latch. When using read-modify-write instructions (i.e., BCF, BSF, etc.) on a port, the value of the port pins is read, the desired operation is done to this value, and this value is then written to the port latch.

Example 5-1 shows the effect of two sequential read-modify-write instructions on an I/O port.

EXAMPLE 5-1: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
;Initial GPIO Settings
; GPIO<5:3> Inputs
; GPIO<2:0> Outputs
;
;                    GPIO latch  GPIO pins
;                    ----------  ----------
  BCF   GPIO, 5   ;--01 -ppp   --11 pppp
  BCF   GPIO, 4   ;--10 -ppp   --11 pppp
  MOVLW 007h       ;
  TRIS  GPIO      ;--10 -ppp   --10 pppp
;
;Note that the user may have expected the pin
;values to be --00 pppp. The 2nd BCF caused
;GP5 to be latched as the pin value (High).
```

A pin actively outputting a Low or High should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output currents may damage the chip.

NOTES:

6.0 EEPROM PERIPHERAL OPERATION

The PIC 12CE673 and PIC 12CE674 each have 16 bytes of EEPROM data memory. The EEPROM memory has an endurance of 1,000,000 erase/write cycles and a data retention of greater than 40 years. The EEPROM data memory supports a bi-directional 2-wire bus and data transmission protocol. These two-wires are serial data (SDA) and serial clock (SCL), that are mapped to bit6 and bit7, respectively, of the GPIO register (SFR 06h). Unlike the GP0-GP5 that are connected to the I/O pins, SDA and SCL are only connected to the internal EEPROM peripheral. For most applications, all that is required is calls to the following functions:

```
; Byte_Write: Byte write routine
;       Inputs: EEPROM Address     EEADDR
;               EEPROM Data        EEDATA
;       Outputs:    Return 01 in W if OK, else
                    return 00 in W
;
; Read_Current: Read EEPROM at address
currently held by EE device.
;       Inputs: NONE
;       Outputs:    EEPROM Data    EEDATA
;                   Return 01 in W if OK, else
                    return 00 in W
;
; Read_Random: Read EEPROM byte at supplied
address
;       Inputs: EEPROM Address     EEADDR
;       Outputs:    EEPROM Data    EEDATA
;                   Return 01 in W if OK,
                    else return 00 in W
```

The code for these functions is available on our web site (www.microchip.com). The code will be accessed by either including the source code FL67XINC.ASM or by linking FLASH67X.ASM. FLASH67X.INC provides external definition to the calling program.

6.0.1 SERIAL DATA

SDA is a bi-directional pin used to transfer addresses and data into and data out of the device.

For normal data transfer, SDA is allowed to change only during SCL low. Changes during SCL high are reserved for indicating the START and STOP conditions.

6.0.2 SERIAL CLOCK

This SCL signal is used to synchronize the data transfer from and to the EEPROM.

6.1 Bus Characteristics

The following **bus protocol** is to be used with the EEPROM data memory. In this section, the term "processor" is used to denote the portion of the PIC12C67X that interfaces to the EEPROM via software.

- Data transfer may be initiated only when the bus is not busy.

During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is HIGH will be interpreted as a START or STOP condition.

Accordingly, the following bus conditions have been defined (Figure 6-3).

6.1.1 BUS NOT BUSY (A)

Both data and clock lines remain HIGH.

6.1.2 START DATA TRANSFER (B)

A HIGH to LOW transition of the SDA line while the clock (SCL) is HIGH determines a START condition. All commands must be preceded by a START condition.

6.1.3 STOP DATA TRANSFER (C)

A LOW to HIGH transition of the SDA line while the clock (SCL) is HIGH determines a STOP condition. All operations must be ended with a STOP condition.

6.1.4 DATA VALID (D)

The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal.

The data on the line must be changed during the LOW period of the clock signal. There is one bit of data per clock pulse.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of the data bytes transferred between the START and STOP conditions is determined by the available data EEPROM space.

6.1.5 ACKNOWLEDGE

The EEPROM, when addressed, will generate an acknowledge after the reception of each byte. The processor must generate an extra clock pulse which is associated with this acknowledge bit.

Note:	Acknowledge bits are not generated if an internal programming cycle is in progress.

The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken in to account. The processor must signal an end of data to the EEPROM by not generating an acknowledge bit on the last byte that has been clocked out of the EEPROM. In this case, the EEPROM must leave the data line HIGH to enable the processor to generate the STOP condition (Figure 6-4).

FIGURE 6-1: BLOCK DIAGRAM OF GPIO6 (SDA LINE)

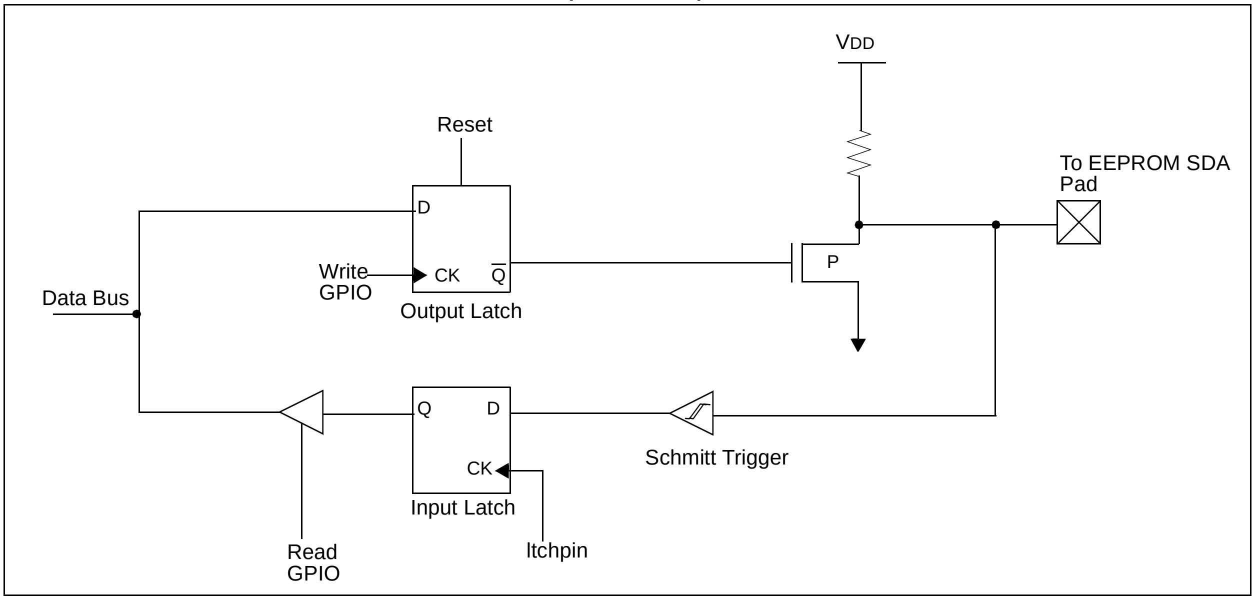


FIGURE 6-2: BLOCK DIAGRAM OF GPIO7 (SCL LINE)

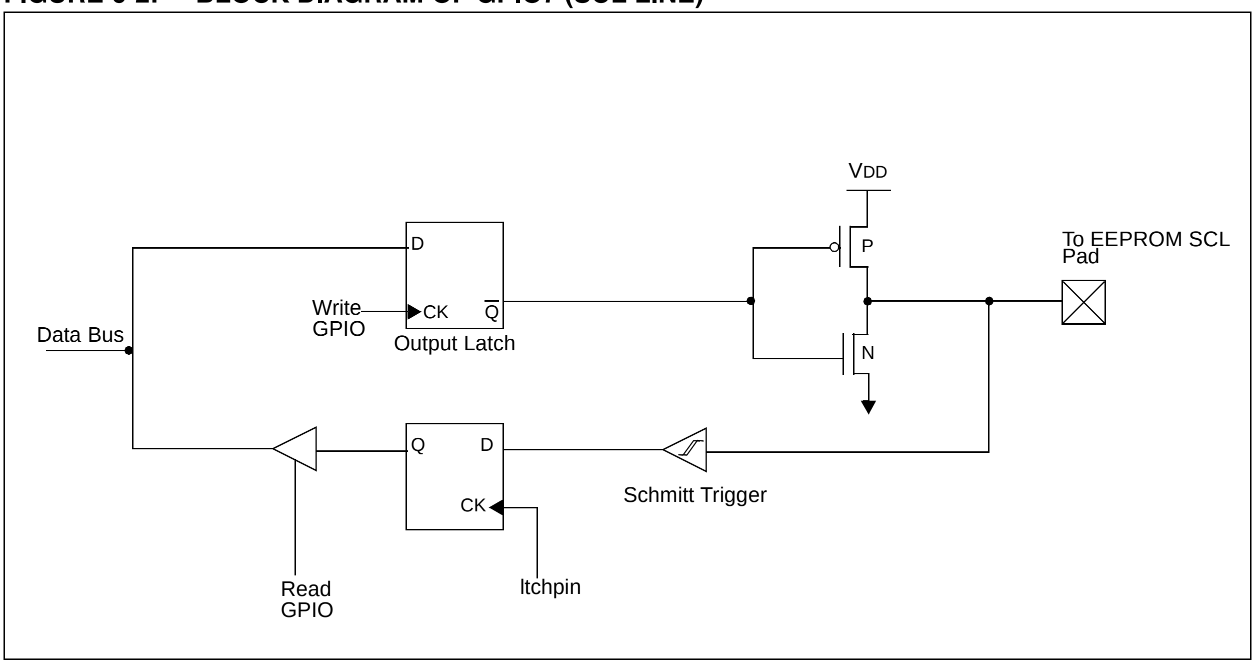


FIGURE 6-3: DATA TRANSFER SEQUENCE ON THE SERIAL BUS

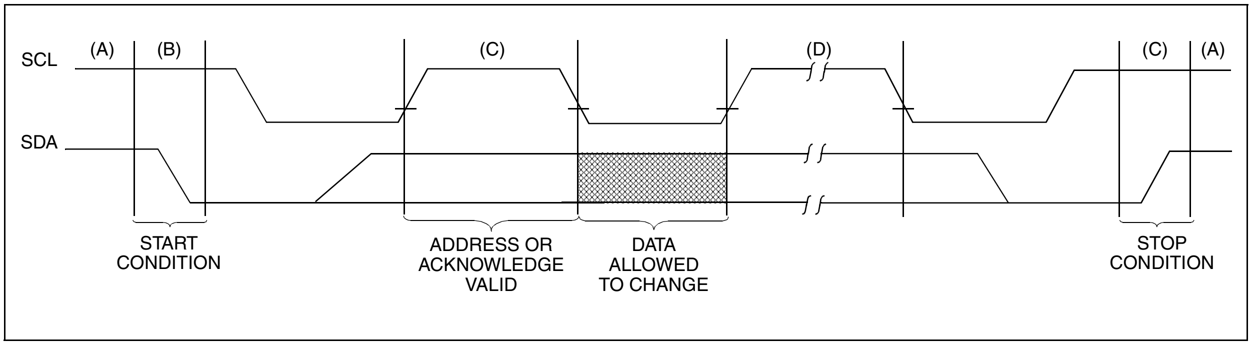
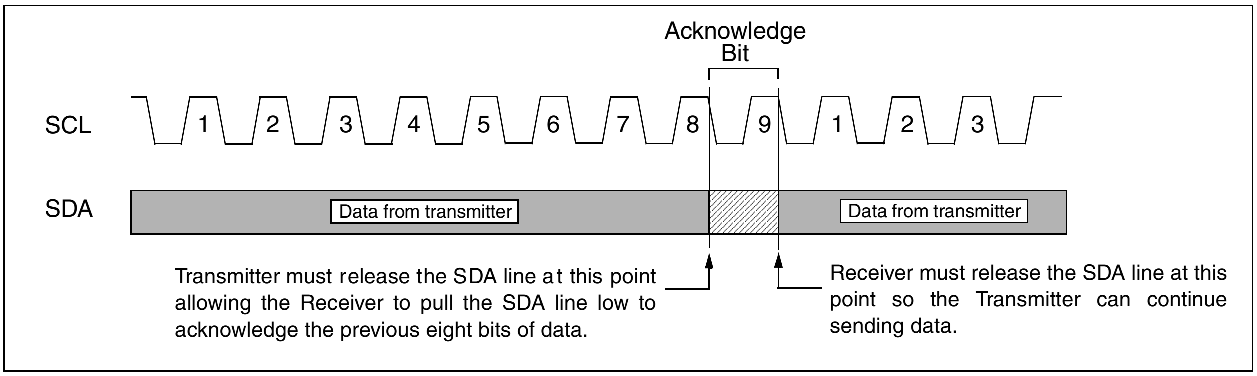


FIGURE 6-4: ACKNOWLEDGE TIMING

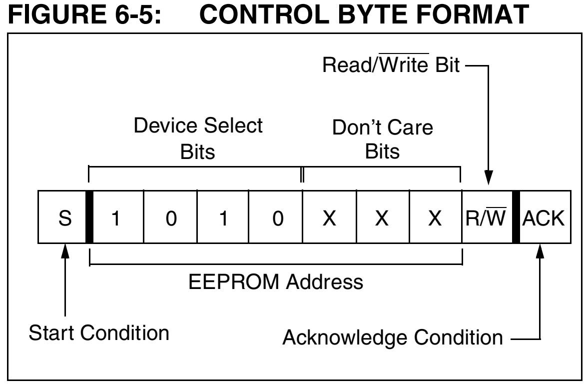


6.2 Device Addressing

After generating a START condition, the processor transmits a control byte consisting of a EEPROM address and a Read/$\overline{\text{Write}}$ bit that indicates what type of operation is to be performed. The EEPROM address consists of a 4-bit device code (1010) followed by three don't care bits.

The last bit of the control byte determines the operation to be performed. When set to a one, a read operation is selected, and when set to a zero, a write operation is selected (Figure 6-5). The bus is monitored for its corresponding EEPROM address all the time. It generates an acknowledge bit if the EEPROM address was true and it is not in a programming mode.

FIGURE 6-5: CONTROL BYTE FORMAT



6.3 Write Operations

6.3.1 BYTE WRITE

Following the start signal from the processor, the device code (4 bits), the don't care bits (3 bits), and the R/$\overline{W}$ bit (which is a logic low) are placed onto the bus by the processor. This indicates to the addressed EEPROM that a byte with a word address will follow after it has generated an acknowledge bit during the ninth clock cycle. Therefore, the next byte transmitted by the processor is the word address and will be written into the address pointer. Only the lower four address bits are used by the device, and the upper four bits are don't cares. If the address byte is acknowledged, the processor will then transmit the data word to be written into the addressed memory location. The memory acknowledges again and the processor generates a stop condition. This initiates the internal write cycle, and during this time will not generate acknowledge signals. After a byte write command, the internal address counter will not be incremented and will point to the same address location that was just written. If a stop bit sequence is transmitted to the device at any point in the write command sequence before the entire sequence is complete, then the command will abort and no data will be written. If more than 8 data bits are transmitted before the stop bit sequence is sent, then the device will clear the previously loaded byte and begin loading the data buffer again. If more than one data byte is transmitted to the device and a stop bit is sent before a full eight data bits have been transmitted, then the write command will abort and no data will be written. The EEPROM memory employs a VCC threshold detector circuit, which disables the internal erase/write logic if the VCC is below minimum VDD. Byte write operations must be preceded and immediately followed by a bus not busy bus cycle where both SDA and SCL are held high. (See Figure 6-7 for Byte Write operation.)

6.4 Acknowledge Polling

Since the EEPROM will not acknowledge during a write cycle, this can be used to determine when the cycle is complete (this feature can be used to maximize bus throughput). Once the stop condition for a write command has been issued from the processor, the device initiates the internally timed write cycle. ACK polling can be initiated immediately. This involves the processor sending a start condition followed by the control byte for a write command (R/$\overline{W}$ = 0). If the device is still busy with the write cycle, then no ACK will be returned. If no ACK is returned, then the start bit and control byte must be re-sent. If the cycle is complete, then the device will return the ACK and the processor can then proceed with the next read or write command. (See Figure 6-6 for flow diagram.)

FIGURE 6-6: ACKNOWLEDGE POLLING FLOW

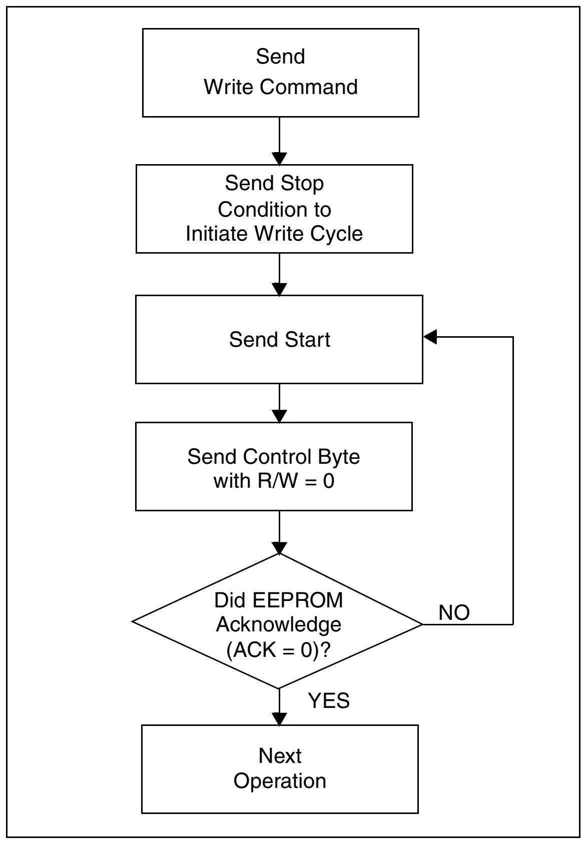
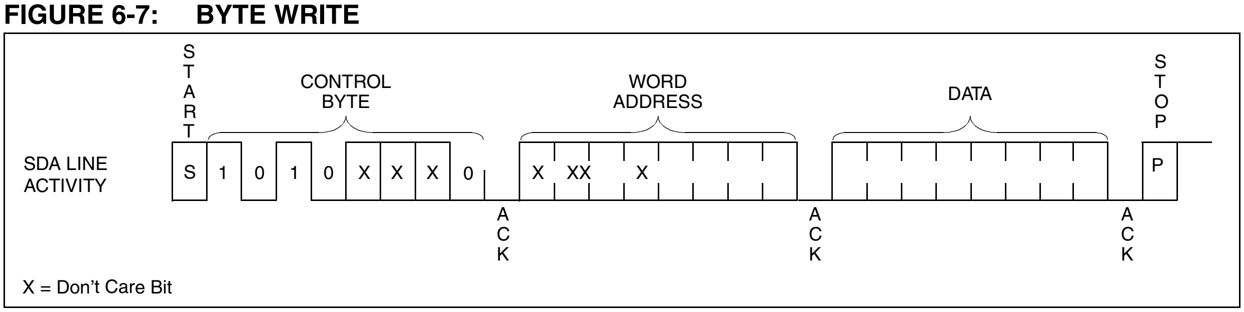


FIGURE 6-7: BYTE WRITE



6.5 Read Operations

Read operations are initiated in the same way as write operations with the exception that the R/$\overline{W}$ bit of the EEPROM address is set to one. There are three basic types of read operations; current address read, random read and sequential read.

6.5.1 CURRENT ADDRESS READ

The EEPROM contains an address counter that maintains the address of the last word accessed, internally incremented by one. Therefore, if the previous read access was to address n, the next current address read operation would access data from address n + 1. Upon receipt of the EEPROM address with the R/$\overline{W}$ bit set to one, the EEPROM issues an acknowledge and transmits the 8-bit data word. The processor will not acknowledge the transfer, but does generate a stop condition and the EEPROM discontinues transmission (Figure 6-8).

6.5.2 RANDOM READ

Random read operations allow the processor to access any memory location in a random manner. To perform this type of read operation, first the word address must be set. This is done by sending the word address to the EEPROM as part of a write operation. After the word address is sent, the processor generates a start condition following the acknowledge. This terminates the write operation, but not before the internal address pointer is set. Then the processor issues the control byte again, but with the R/$\overline{W}$ bit set to a one. The EEPROM will then issue an acknowledge and transmits the 8-bit data word. The processor will not acknowledge the transfer, but does generate a stop condition and the EEPROM discontinues transmission (Figure 6-9). After this command, the internal address counter will point to the address location following the one that was just read.

6.5.3 SEQUENTIAL READ

Sequential reads are initiated in the same way as a random read, except that after the device transmits the first data byte, the processor issues an acknowledge as opposed to a stop condition in a random read. This directs the EEPROM to transmit the next sequentially addressed 8-bit word (Figure 6-10).

To provide sequential reads, the EEPROM contains an internal address pointer, which is incremented by one at the completion of each read operation. This address pointer allows the entire memory contents to be serially read during one operation.

FIGURE 6-8: CURRENT ADDRESS READ

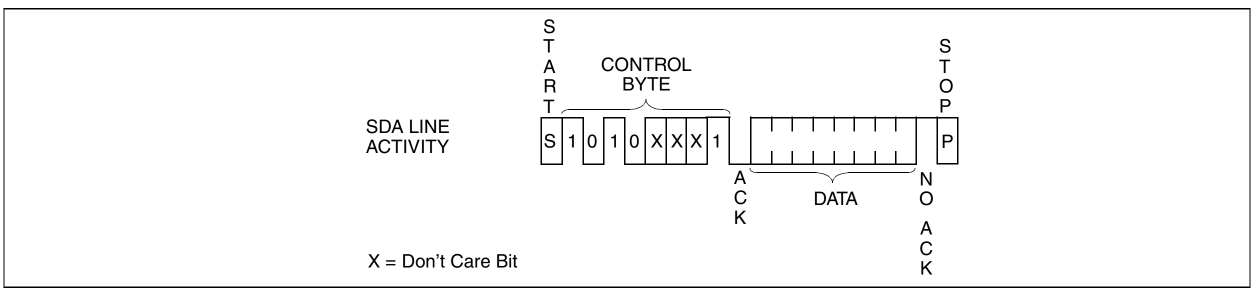


FIGURE 6-9: RANDOM READ

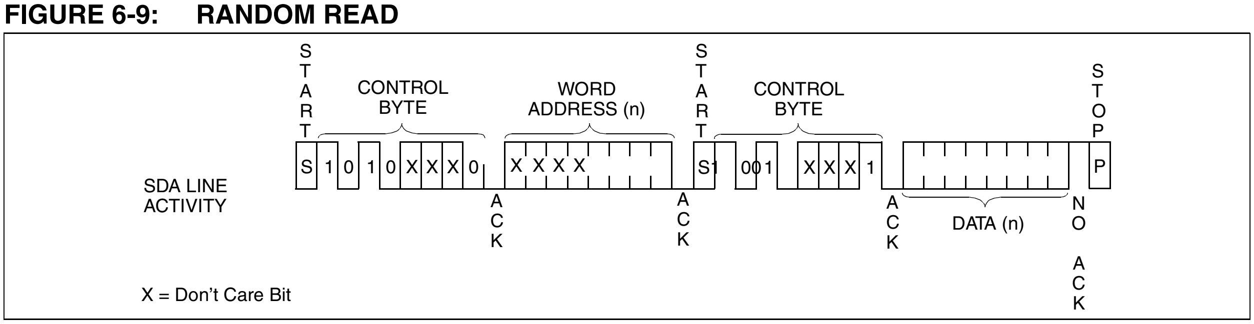
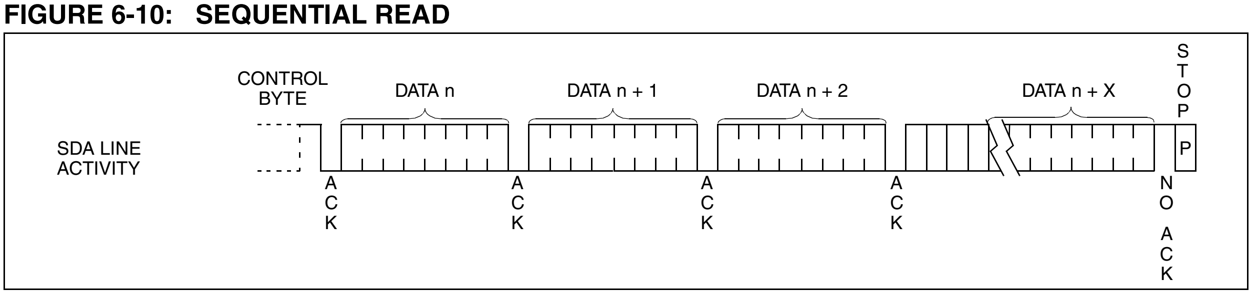


FIGURE 6-10: SEQUENTIAL READ



NOTES:

7.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 7-1 is a simplified block diagram of the Timer0 module.

Timer mode is selected by clearing bit T0CS (OPTION<5>). In timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If the TMR0 register is written, the increment is inhibited for the following two instruction cycles (Figure 7-2 and Figure 7-3). The user can work around this by writing an adjusted value to the TMR0 register.

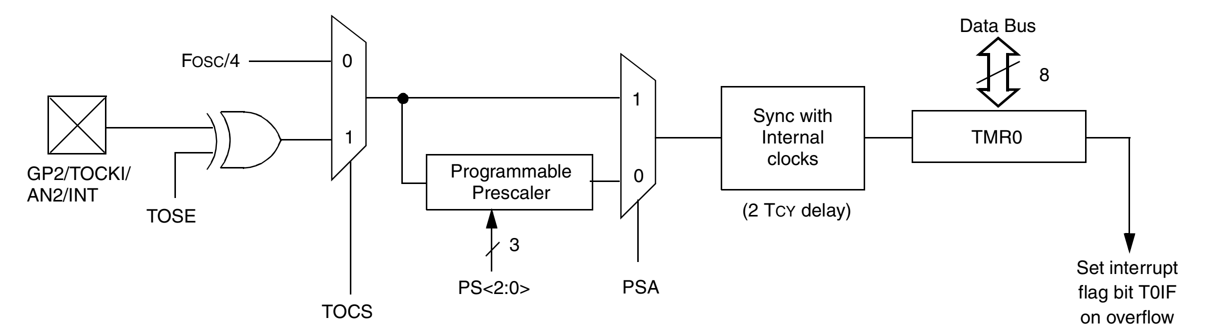
Counter mode is selected by setting bit T0CS (OPTION<5>). In counter mode, Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the bit T0SE (OPTION<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 7.2.

The prescaler is mutually exclusively shared between the Timer0 module and the Watchdog Timer. The prescaler assignment is controlled in software by control bit PSA (OPTION<3>). Clearing bit PSA will assign the prescaler to the Timer0 module. The prescaler is not readable or writable. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4, ..., 1:256 are selectable. Section 7.3 details the operation of the prescaler.

7.1 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit T0IF (INTCON<2>). The interrupt can be masked by clearing bit T0IE (INTCON<5>). Bit T0IF must be cleared in software by the Timer0 module interrupt service routine before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from SLEEP, since the timer is shut off during SLEEP. See Figure 7-4 for Timer0 interrupt timing.

FIGURE 7-1: TIMER0 BLOCK DIAGRAM



Note 1: TOCS, TOSE, PSA, PS<2:0> (OPTION<5:0>).
2: The prescaler is shared with Watchdog Timer (refer to Figure 7-6 for detailed block diagram).

FIGURE 7-2: TIMER0 TIMING: INTERNAL CLOCK/NO PRESCALE

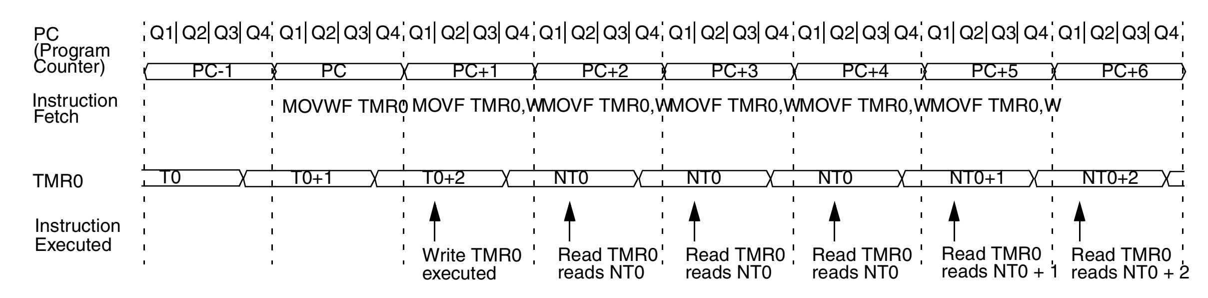


FIGURE 7-3: TIMER0 TIMING: INTERNAL CLOCK/PRESCALE 1:2

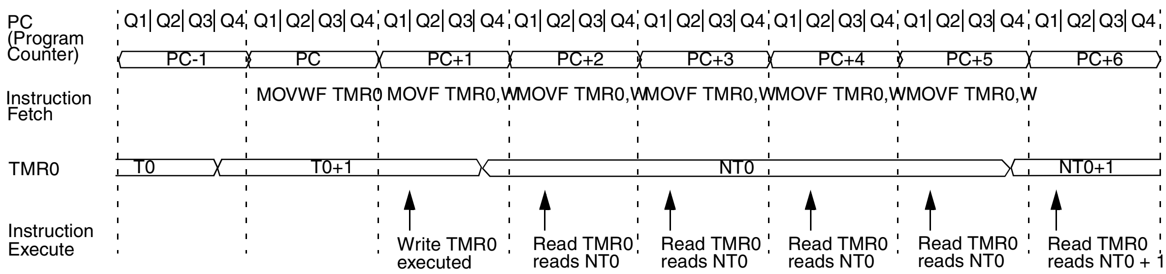
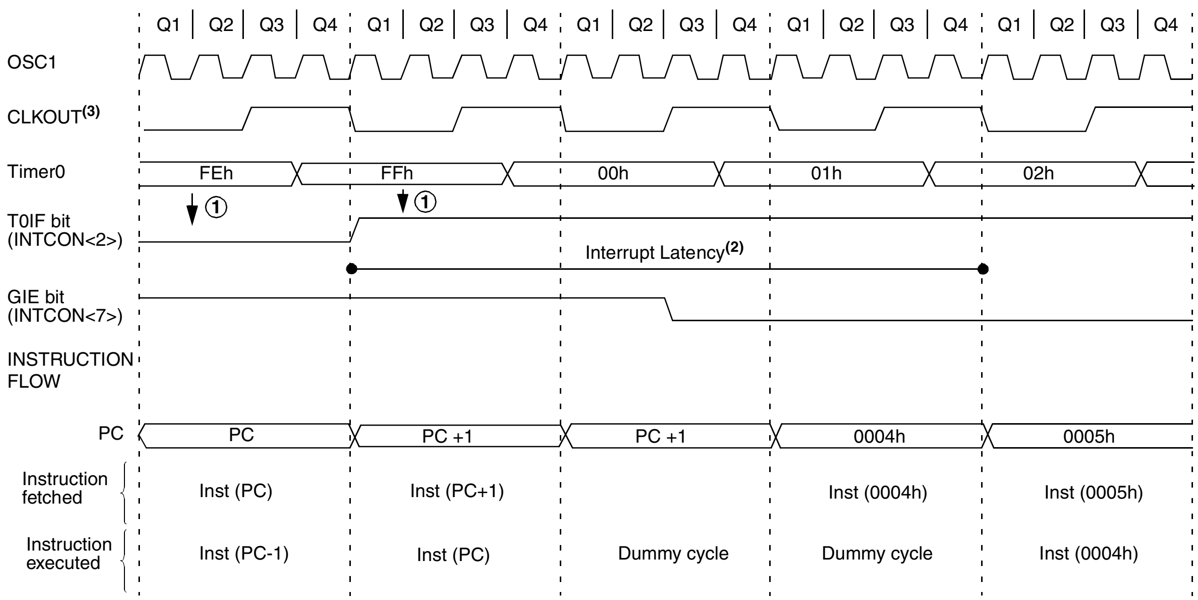


FIGURE 7-4: TIMER0 INTERRUPT TIMING



Note 1: Interrupt flag bit T0IF is sampled here (every Q1).

2: Interrupt latency = $3T_{CY}$ where T_{CY} = instruction cycle time.

3: CLKOUT is available only in the INTRC and EXTRC oscillator modes.

7.2 Using Timer0 with an External Clock

When an external clock input is used for Timer0, it must meet certain requirements. The requirements ensure the external clock can be synchronized with the internal phase clock (TOSC). Also, there is a delay in the actual incrementing of Timer0 after synchronization.

7.2.1 EXTERNAL CLOCK SYNCHRONIZATION

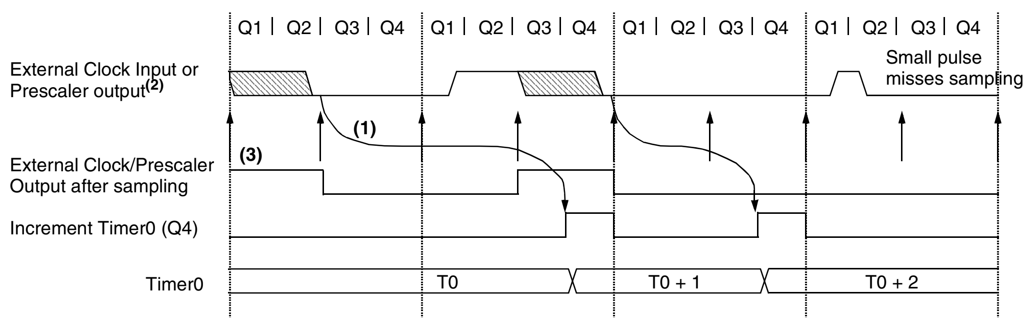
When no prescaler is used, the external clock input is used as the clock source. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 7-5). Therefore, it is necessary for T0CKI to be high for at least 2TOSC (and a small RC delay of 20 ns) and low for at least 2TOSC (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple-counter type prescaler, so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple-counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4TOSC (and a small RC delay of 40 ns) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of 10 ns. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.

7.2.2 TMR0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the Timer0 module is actually incremented. Figure 7-5 shows the delay from the external clock edge to the timer incrementing.

FIGURE 7-5: TIMER0 TIMING WITH EXTERNAL CLOCK



Note 1: Delay from clock input change to Timer0 increment is 3TOSC to 7TOSC. (Duration of Q = TOSC). Therefore, the error in measuring the interval between two edges on Timer0 input = ±4TOSC max.

2: External clock if no prescaler selected; prescaler output otherwise.

3: The arrows indicate the points in time where sampling occurs.

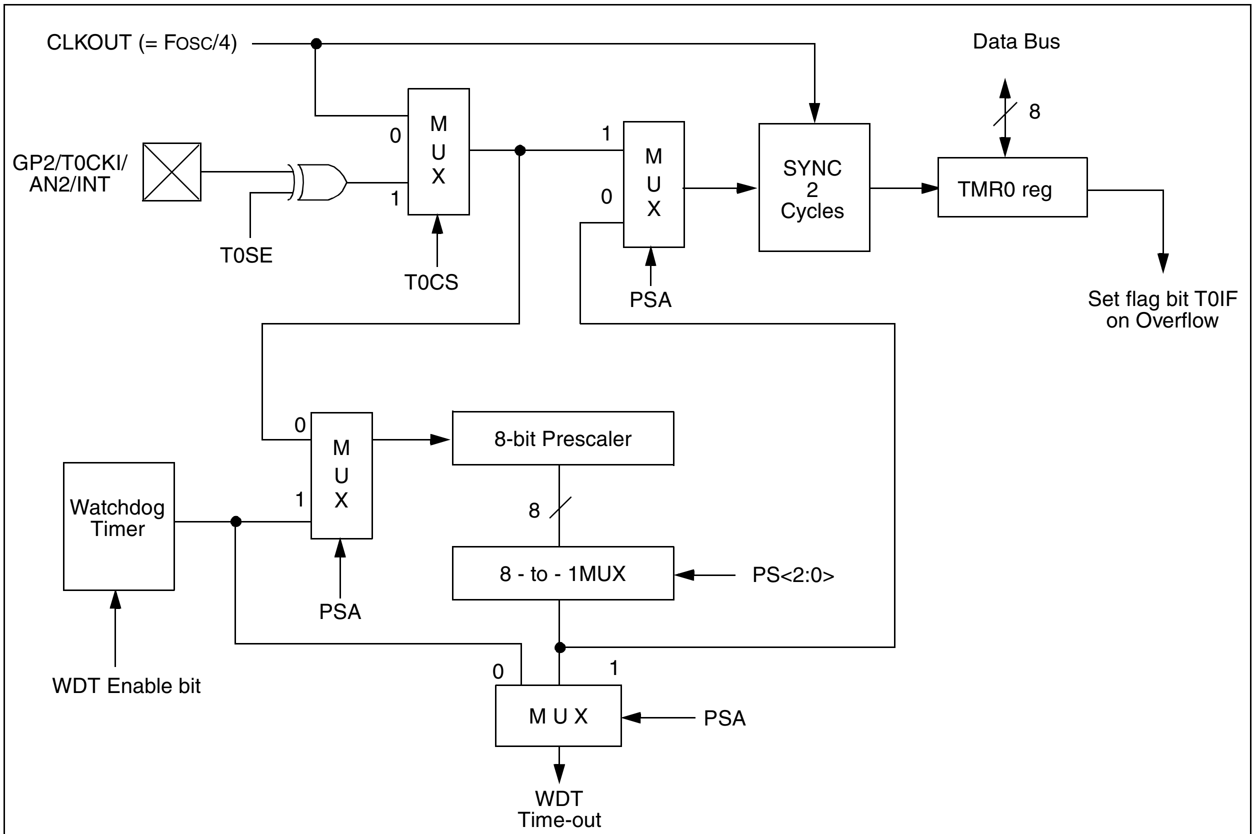
7.3 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer, respectively (Figure 7-6). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet. Note that there is only one prescaler available which is mutually exclusively shared between the Timer0 module and the Watchdog Timer. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer, and vice-versa.

The PSA and PS<2:0> bits (OPTION<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (i.e., `CLRF 1, MOVWF 1, BSF 1,x` ...., etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

FIGURE 7-6: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



Note: T0CS, T0SE, PSA, PS<2:0> are (OPTION<5:0>).

7.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control, (i.e., it can be changed “on-the-fly” during program execution).

Note:	To avoid an unintended device RESET, the following instruction sequence (shown in Example 7-1) must be executed when changing the prescaler assignment from Timer0 to the WDT. This sequence must be followed even if the WDT is disabled.

EXAMPLE 7-1: CHANGING PRESCALER (TIMER0→WDT)

```
BCF     STATUS, RP0   ;Bank 0
CLRF    TMR0          ;Clear TMR0 & Prescaler
BSF     STATUS, RP0   ;Bank 1
CLRWDT                ;Clears WDT
MOVLW   b'xxxx1xxx'   ;Select new prescale
MOVWF   OPTION_REG    ;value & WDT
BCF     STATUS, RP0   ;Bank 0
```

To change prescaler from the WDT to the Timer0 module, use the sequence shown in Example 7-2.

EXAMPLE 7-2: CHANGING PRESCALER (WDT→TIMER0)

```
CLRWDT                ;Clear WDT and
                      ;prescaler
BSF     STATUS, RP0   ;Bank 1
MOVLW   b'xxxx0xxx'   ;Select TMR0, new
                      ;prescale value and
MOVWF   OPTION_REG    ;clock source
BCF     STATUS, RP0   ;Bank 0
```

TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other Resets
01h	TMR0	Timer0 module’s register								xxxx xxxx	uuuu uuuu
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 000x	0000 000u
81h	OPTION	$\overline{\text{GPPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRIS	—	—	TRIS5	TRIS4	TRIS3	TRIS2	TRIS1	TRIS0	--11 1111	--11 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.

NOTES:

8.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-To-Digital (A/D) converter module has four analog inputs.

The A/D allows conversion of an analog input signal to a corresponding 8-bit digital number (refer to Application Note AN546 for use of A/D Converter). The output of the sample and hold is the input into the converter, which generates the result via successive approximation. The analog reference voltage is software selectable to either the device's positive supply voltage (VDD) or the voltage level on the GP1/AN1/VREF pin. The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode.

The A/D module has three registers. These registers are:

- A/D Result Register (ADRES)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)

The ADCON0 Register, shown in Figure 8-1, controls the operation of the A/D module. The ADCON1 Register, shown in Figure 8-2, configures the functions of the port pins. The port pins can be configured as analog inputs (GP1 can also be a voltage reference) or as digital I/O.

> **Note 1:** If the port pins are configured as analog inputs (reset condition), reading the port (MOVF GPIO,W) results in reading '0's.
>
> **2:** Changing ADCON1 Register can cause the GPIF and INTF flags to be set in the INTCON Register. These interrupts should be disabled prior to modifying ADCON1.

REGISTER 8-1: ADCON0 REGISTER (ADDRESS 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS1	ADCS0	reserved	CHS1	CHS0	GO/$\overline{\text{DONE}}$	reserved	ADON
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset

bit 7-6: **ADCS<1:0>:** A/D Conversion Clock Select bits
00 = FOSC/2
01 = FOSC/8
10 = FOSC/32
11 = FRC (clock derived from an RC oscillation)

bit 5: **Reserved**

bit 4-3: **CHS<1:0>:** Analog Channel Select bits
00 = channel 0, (GP0/AN0)
01 = channel 1, (GP1/AN1)
10 = channel 2, (GP2/AN2)
11 = channel 3, (GP4/AN3)

bit 2: **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
If ADON = 1
1 = A/D conversion in progress (setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)

bit 1: **Reserved**

bit 0: **ADON:** A/D on bit
1 = A/D converter module is operating
0 = A/D converter module is shut off and consumes no operating current

REGISTER 8-2: ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PCFG2	PCFG1	PCFG0
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
- n = Value at POR reset

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

bit 1-0: **PCFG<2:0>:** A/D Port Configuration Control bits

PCFG<2:0>	GP4	GP2	GP1	GP0	VREF
000[(1)]	AA		A	A	VDD
001	AA		VREF	AG	P1
010	DA		A	A	VDD
011	DA		VREF	AG	P1
100	DD		A	A	VDD
101	DD		VREF	AG	P1
110	DD		D	A	VDD
111	DD		D	D	VDD

A = Analog input

D = Digital I/O

Note 1: Value on reset.

2: Any instruction that reads a pin configured as an analog input will read a '0'.

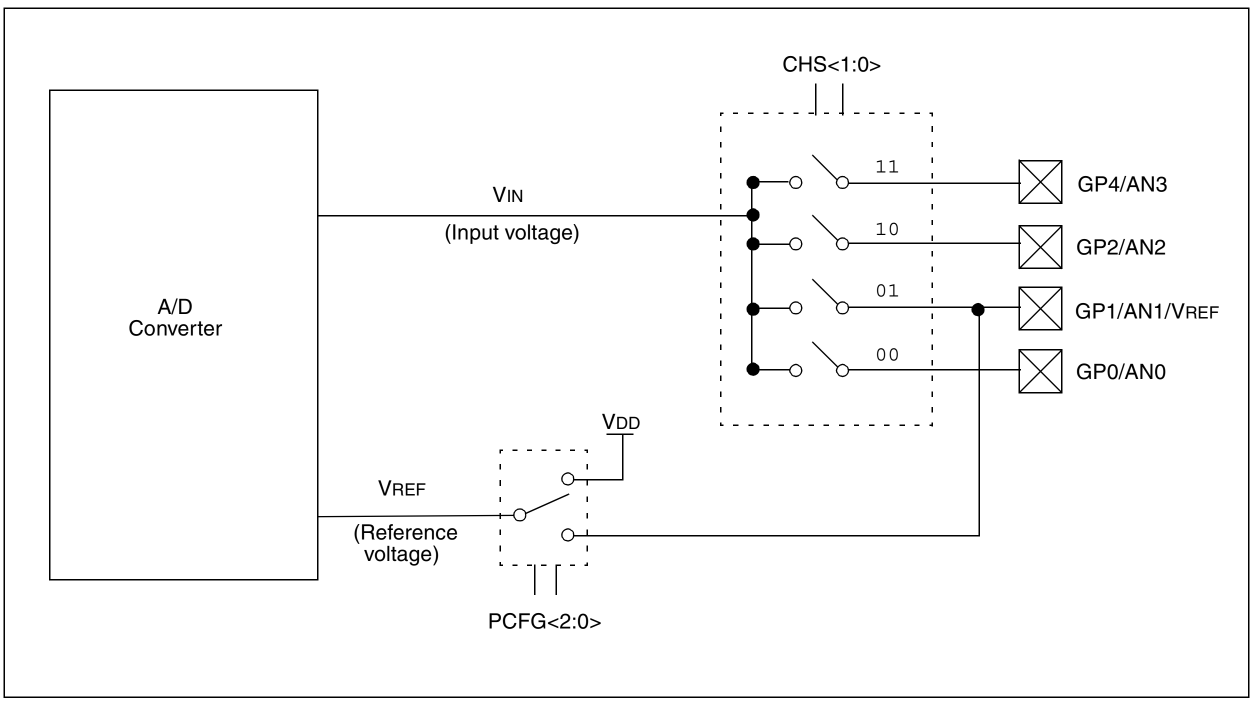
The ADRES Register contains the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRES register, the GO/$\overline{\text{DONE}}$ bit (ADCON0<2>) is cleared, and A/D interrupt flag bit ADIF (PIE1<6>) is set. The block diagrams of the A/D module are shown in Figure 8-1.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as an input. To determine sample time, see Section 8.1. After this acquisition time has elapsed, the A/D conversion can be started. The following steps should be followed for doing an A/D conversion:

1. Configure the A/D module:
   - Configure analog pins / voltage reference / and digital I/O (ADCON1 and TRIS)
   - Select A/D input channel (ADCON0)
   - Select A/D conversion clock (ADCON0)
   - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
   - Clear ADIF bit
   - Set ADIE bit
   - Set GIE bit
3. Wait the required acquisition time.
4. Start conversion:
   - Set GO/$\overline{\text{DONE}}$ bit (ADCON0)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared

   OR
   - Waiting for the A/D interrupt
6. Read A/D Result Register (ADRES), clear bit ADIF if required.
7. For the next conversion, go to step 1, step 2 or step 3 as required. The A/D conversion time per bit is defined as T_{AD}. A minimum wait of $2T_{AD}$ is required before next acquisition starts.

FIGURE 8-1: A/D BLOCK DIAGRAM



8.1 A/D Sampling Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (C_{HOLD}) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 8-2. The source impedance (R_S) and the internal sampling switch (R_{SS}) impedance directly affect the time required to charge the capacitor C_{HOLD}. The sampling switch (R_{SS}) impedance varies over the device voltage (V_{DD}), see Figure 8-2. **The maximum recommended impedance for analog sources is 10 kΩ.** After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 8-1 may be used. This equation assumes that 1/2 LSb error is used (512 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

EQUATION 8-1: A/D MINIMUM CHARGING TIME

$V_{HOLD} = (V_{REF} - (V_{REF}/512)) \bullet (1 - e^{(-Tc/C_{HOLD}(R_{IC} + R_{SS} + R_S))})$

or

$Tc = -(51.2\text{ pF})(1\text{ k}\Omega + R_{SS} + R_S) \ln(1/511)$

Example 8-1 shows the calculation of the minimum required acquisition time T_{ACQ}. This calculation is based on the following system assumptions.

$R_S = 10\text{ k}\Omega$

1/2 LSb error

$V_{DD} = 5V \rightarrow R_{SS} = 7\text{ k}\Omega$

Temp (system max.) = 50°C

$V_{HOLD} = 0$ @ $t = 0$

> **Note 1:** The reference voltage (V_{REF}) has no effect on the equation, since it cancels itself out.
>
> **2:** The charge holding capacitor (C_{HOLD}) is not discharged after each conversion.
>
> **3:** The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.
>
> **4:** After a conversion has completed, a 2.0 T_{AD} delay must complete before acquisition can begin again. During this time, the holding capacitor is not connected to the selected A/D input channel.

EXAMPLE 8-1: CALCULATING THE MINIMUM REQUIRED SAMPLE TIME

T_{ACQ} = Internal Amplifier Settling Time + Holding Capacitor Charging Time + Temperature Coefficient

$T_{ACQ} = 5\ \mu s + Tc + [(\text{Temp} - 25°C)(0.05\ \mu s/°C)]$

$T_C = -C_{HOLD}(R_{IC} + R_{SS} + R_S) \ln(1/512)$

$-51.2\text{ pF}(1\text{ k}\Omega + 7\text{ k}\Omega + 10\text{ k}\Omega) \ln(0.0020)$

$-51.2\text{ pF}(18\text{ k}\Omega) \ln(0.0020)$

$-0.921\ \mu s\ (-6.2146)$

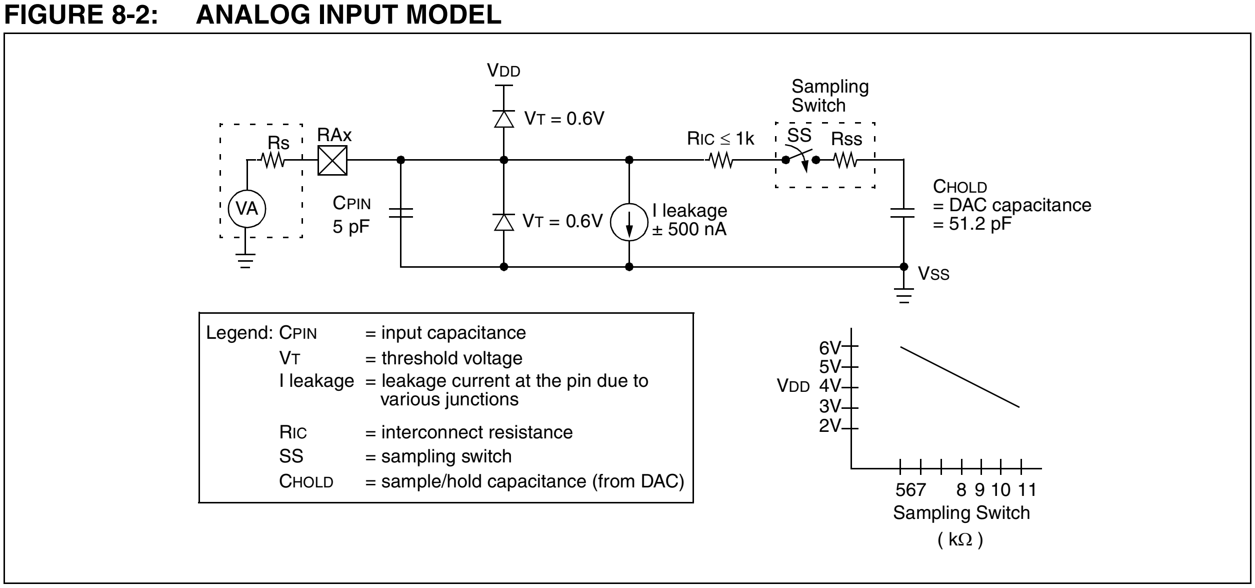
$5.724\ \mu s$

$T_{ACQ} = 5\ \mu s + 5.724\ \mu s + [(50°C - 25°C)(0.05\ \mu s/°C)]$

$10.724\ \mu s + 1.25\ \mu s$

$11.974\ \mu s$

FIGURE 8-2: ANALOG INPUT MODEL



8.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 9.5 TAD per 8-bit conversion. The source of the A/D conversion clock is software selected. The four possible options for TAD are:

- 2TOSC
- 8TOSC
- 32TOSC
- Internal ADC RC oscillator

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6 μs. If the minimum TAD time of 1.6 μs can not be obtained, TAD should be ≤8 μs for preferred operation.

Table 8-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

8.3 Configuring Analog Port Pins

The ADCON1 and TRIS Registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS<2:0> bits and the TRIS bits.

Note 1: When reading the port register, all pins configured as analog input channel will read as cleared (a low level). Pins configured as digital inputs, will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.

2: Analog levels on any pin that is defined as a digital input (including the AN<3:0> pins) may cause the input buffer to consume current that is out of the devices specification.

TABLE 8-1: TAD vs. DEVICE OPERATING FREQUENCIES

AD Clock Source (TAD)		Device Frequency		
Operation	**ADCS<1:0>**	**4 MHz**	**1.25 MHz**	**333.33 kHz**
2TOSC	`00`	500 ns(2)	1.6 μs	6 μs
8TOSC	`01`	2.0 μs	6.4 μs	24 μs(3)
32TOSC	`10`	8.0 μs	25.6 μs(3)	96 μs(3)
Internal ADC RC Oscillator(5)	`11`	2 - 6 μs(1,4)	2 - 6 μs(1,4)	2 - 6 μs(1)

Note 1: The RC source has a typical TAD time of 4 μs.
2: These values violate the minimum required TAD time.
3: For faster conversion times, the selection of another clock source is recommended.
4: While in RC mode, with device frequency above 1 MHz, conversion accuracy is out of specification.
5: For extended voltage devices (LC), please refer to Electrical Specifications section.

8.4 A/D Conversions

Example 8-2 shows how to perform an A/D conversion. The GPIO pins are configured as analog inputs. The analog reference (VREF) is the device VDD. The A/D interrupt is enabled and the A/D conversion clock is FRC. The conversion is performed on the GP0 channel.

Note:	The GO/$\overline{\text{DONE}}$ bit should **NOT** be set in the same instruction that turns on the A/D.

Clearing the GO/$\overline{\text{DONE}}$ bit during a conversion will abort the current conversion. The ADRES register will NOT be updated with the partially completed A/D conversion sample. That is, the ADRES register will continue to contain the value of the last completed conversion (or the last value written to the ADRES register). After the A/D conversion is aborted, a 2TAD wait is required before the next acquisition is started. After this 2TAD wait, an acquisition is automatically started on the selected channel.

EXAMPLE 8-2: DOING AN A/D CONVERSION

```
     BSF     STATUS, RP0         ; Select Page 1
     CLRF    ADCON1              ; Configure A/D inputs
     BSF     PIE1,   ADIE        ; Enable A/D interrupts
     BCF     STATUS, RP0         ; Select Page 0
     MOVLW   0xC1                ; RC Clock, A/D is on, Channel 0 is selected
     MOVWF   ADCON0              ;
     BCF     PIR1,   ADIF        ; Clear A/D interrupt flag bit
     BSF     INTCON, PEIE        ; Enable peripheral interrupts
     BSF     INTCON, GIE         ; Enable all interrupts
;
;  Ensure that the required sampling time for the selected input channel has elapsed.
;  Then the conversion may be started.
;
     BSF     ADCON0, GO          ; Start A/D Conversion
       :                         ; The ADIF bit will be set and the GO/DONE bit
       :                         ;  is cleared upon completion of the A/D Conversion.
```

8.5 A/D Operation During Sleep

The A/D module can operate during SLEEP mode. This requires that the A/D clock source be set to RC (ADCS<1:0> = `11`). When the RC clock source is selected, the A/D module waits one instruction cycle before starting the conversion. This allows the `SLEEP` instruction to be executed, which eliminates all digital switching noise from the conversion. When the conversion is completed, the GO/$\overline{\text{DONE}}$ bit will be cleared, and the result loaded into the ADRES Register. If the A/D interrupt is enabled, the device will wake-up from SLEEP. If the A/D interrupt is not enabled, the A/D module will then be turned off, although the ADON bit will remain set.

When the A/D clock source is another clock option (not RC), a `SLEEP` instruction will cause the present conversion to be aborted and the A/D module to be turned off, though the ADON bit will remain set.

Turning off the A/D places the A/D module in its lowest current consumption state.

Note:	For the A/D module to operate in SLEEP, the A/D clock source must be set to RC (ADCS<1:0> = `11`). To perform an A/D conversion in SLEEP, the GO/$\overline{\text{DONE}}$ bit must be set, followed by the `SLEEP` instruction.

8.6 A/D Accuracy/Error

The overall accuracy of the A/D is less than ± 1 LSb for V_{DD} = 5V ± 10% and the analog V_{REF} = V_{DD}. This overall accuracy includes offset error, full scale error, and integral error. The A/D converter is monotonic over the full V_{DD} range. The resolution and accuracy may be less when either the analog reference (V_{DD}) is less than 5.0V or when the analog reference (V_{REF}) is less than V_{DD}.

The maximum pin leakage current is specified in the Device Data Sheet electrical specification, parameter #D060.

In systems where the device frequency is low, use of the A/D RC clock is preferred. At moderate to high frequencies, T_{AD} should be derived from the device oscillator. T_{AD} must not violate the minimum and should be ≤8 μs for preferred operation. This is because T_{AD}, when derived from T_{OSC}, is kept away from on-chip phase clock transitions. This reduces, to a large extent, the effects of digital switching noise. This is not possible with the RC derived clock. The loss of accuracy due to digital switching noise can be significant if many I/O pins are active.

In systems where the device will enter SLEEP mode after the start of the A/D conversion, the RC clock source selection is required. In this mode, the digital noise from the modules in SLEEP are stopped. This method gives high accuracy.

8.7 Effects of a Reset

A device reset forces all registers to their reset state. This forces the A/D module to be turned off, and any conversion is aborted. The value that is in the ADRES register is not modified for a Reset. The ADRES register will contain unknown data after a Power-on Reset.

8.8 Connection Considerations

If the input voltage exceeds the rail values (V_{SS} or V_{DD}) by greater than 0.2V, then the accuracy of the conversion is out of specification.

Note:	For the PIC12C67X, care must be taken when using the GP4 pin in A/D conversions due to its proximity to the OSC1 pin.

An external RC filter is sometimes added for anti-aliasing of the input signal. The R component should be selected to ensure that the total source impedance is kept under the 10 kΩ recommended specification. Any external components connected (via hi-impedance) to an analog input pin (capacitor, zener diode, etc.) should have very little leakage current at the pin.

8.9 Transfer Function

The ideal transfer function of the A/D converter is as follows: the first transition occurs when the analog input voltage (V_{AIN}) is 1 LSb (or Analog V_{REF} / 256) (Figure 8-3).

FIGURE 8-3: A/D TRANSFER FUNCTION

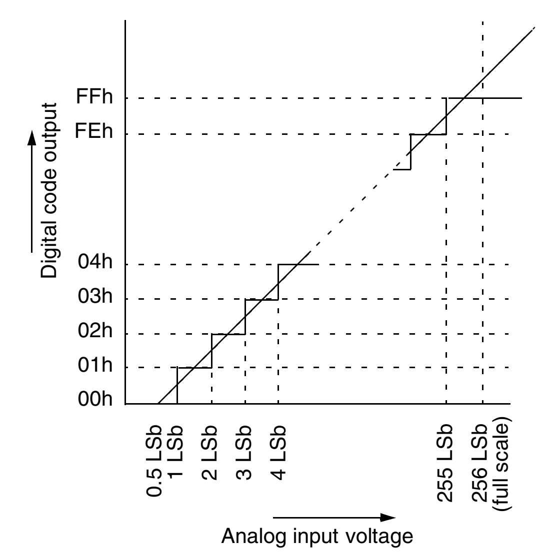


FIGURE 8-4: FLOWCHART OF A/D OPERATION

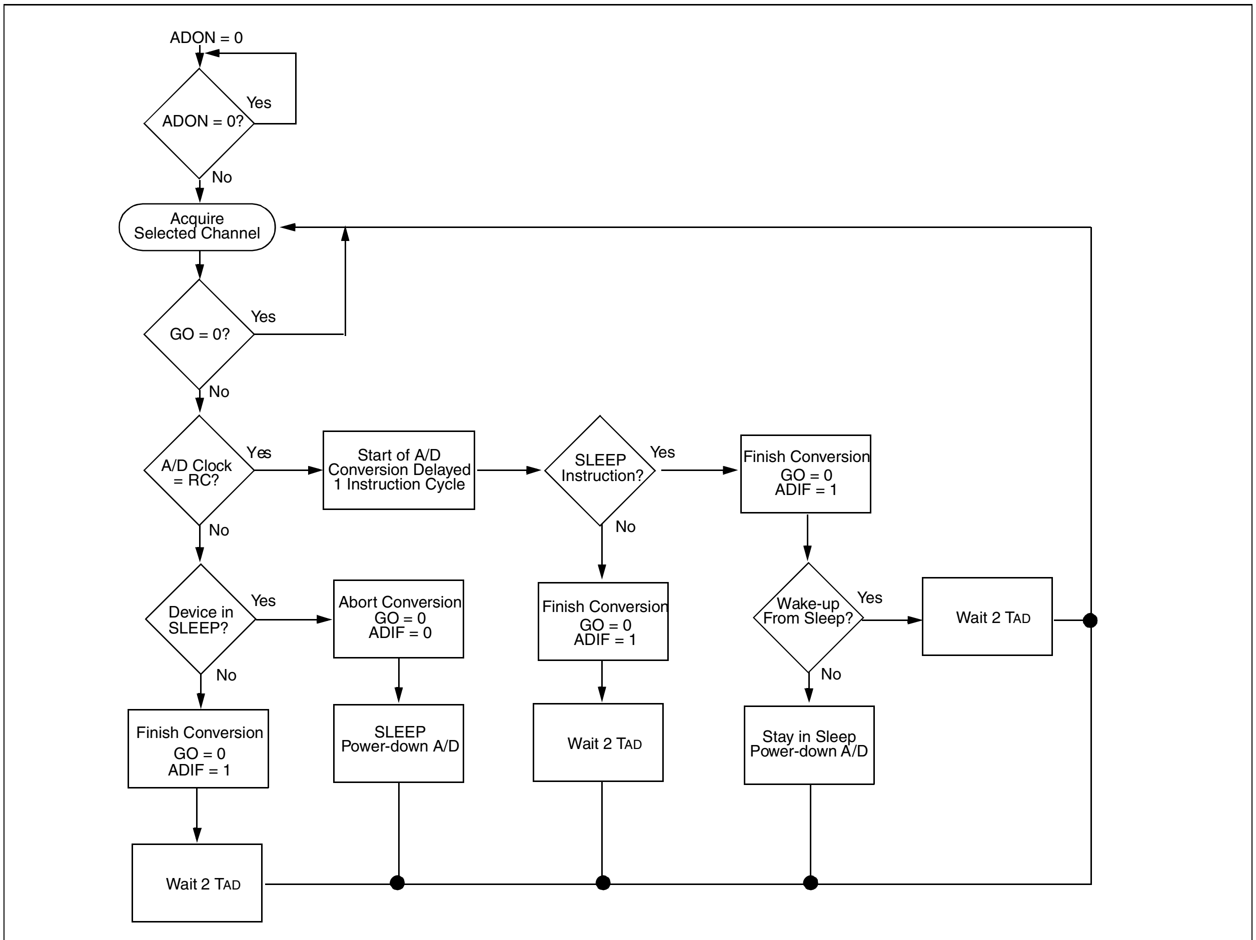


TABLE 8-2: SUMMARY OF A/D REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other Resets
0Bh/8Bh	INTCON(1)	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF	—	—	—	—	—	—	-0-- ----	-0-- ----
8Ch	PIE1	—	ADIE	—	—	—	—	—	—	-0-- ----	-0-- ----
1Eh	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	reserved	CHS1	CHS0	GO/$\overline{\text{DONE}}$	reserved	ADON	0000 0000	0000 0000
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	---- -000
05h	GPIO	SCL(2)	SDA(2)	GP5	GP4	GP3	GP2	GP1	GP0	11xx xxxx	11uu uuuu
85h	TRIS	—	—	TRIS5	TRIS4	TRIS3	TRIS2	TRIS1	TRIS0	--11 1111	--11 1111

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for A/D conversion.
Note 1: These registers can be addressed from either bank.
2: The SCL (GP7) and SDA (GP6) bits are unimplemented on the PIC12C671/672 and read as '0'.

9.0 SPECIAL FEATURES OF THE CPU

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real-time applications. The PIC12C67X family has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

- Oscillator selection
- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
- Interrupts
- Watchdog Timer (WDT)
- SLEEP
- Code protection
- ID locations
- In-circuit serial programming

The PIC12C67X has a Watchdog Timer, which can be shut off only through configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only, designed to keep the part in reset while the power supply stabilizes. With these two timers on-chip, most applications need no external reset circuitry.

SLEEP mode is designed to offer a very low current power-down mode. The user can wake-up from SLEEP through external reset, Watchdog Timer Wake-up, or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The INTRC/EXTRC oscillator option saves system cost, while the LP crystal option saves power. A set of configuration bits are used to select various options.

9.1 Configuration Bits

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory location 2007h.

The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special test/configuration memory space (2000h-3FFFh), which can be accessed only during programming.

REGISTER 9-1: CONFIGURATION WORD

CP1	CP0	CP1	CP0	CP1	CP0	MCLRE	CP1	CP0	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0
bit13													bit0

Register: CONFIG
Address 2007h

bit 13-8, 6-5: **CP<1:0>:** Code Protection bit pairs(1)
`11` = Code protection off
`10` = Locations 400h through 7FEh code protected (do not use for PIC12C671 and PIC12CE673)
`01` = Locations 200h through 7FEh code protected
`00` = All memory is code protected

bit 7: **MCLRE:** Master Clear Reset Enable bit
1 = Master Clear Enabled
0 = Master Clear Disabled

bit 4: **$\overline{\text{PWRTE}}$:** Power-up Timer Enable bit
1 = PWRT disabled
0 = PWRT enabled

bit 3: **WDTE:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 2-0: **FOSC<2:0>:** Oscillator Selection bits
`111` = EXTRC, Clockout on OSC2
`110` = EXTRC, OSC2 is I/O
`101` = INTRC, Clockout on OSC2
`100` = INTRC, OSC2 is I/O
`011` = Invalid Selection
`010` = HS Oscillator
`001` = XT Oscillator
`000` = LP Oscillator

Note 1: All of the CP<1:0> pairs have to be given the same value to enable the code protection scheme listed.

9.2 Oscillator Configurations

9.2.1 OSCILLATOR TYPES

The PIC12C67X can be operated in seven different oscillator modes. The user can program three configuration bits (FOSC<2:0>) to select one of these seven modes:

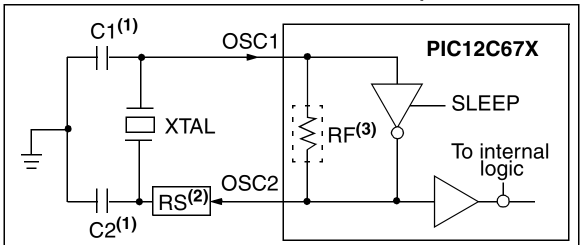
- LP: Low Power Crystal
- HS: High Speed Crystal/Resonator
- XT: Crystal/Resonator
- INTRC*: Internal 4 MHz Oscillator
- EXTRC*: External Resistor/Capacitor

*Can be configured to support CLKOUT

9.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, HS or LP modes, a crystal or ceramic resonator is connected to the GP5/OSC1/CLKIN and GP4/OSC2 pins to establish oscillation (Figure 9-1). The PIC12C67X oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, HS or LP modes, the device can have an external clock source drive the GP5/OSC1/CLKIN pin (Figure 9-2).

FIGURE 9-1: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (XT, HS OR LP OSC CONFIGURATION)



Note 1: See Capacitor Selection tables for recommended values of C1 and C2.
2: A series resistor (RS) may be required for AT strip cut crystals.
3: RF varies with the oscillator mode selected (approx. value = 10 MΩ).

FIGURE 9-2: EXTERNAL CLOCK INPUT OPERATION (XT, HS OR LP OSC CONFIGURATION)

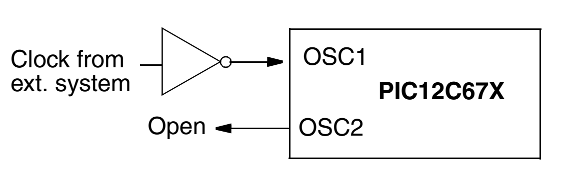


TABLE 9-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS - PIC12C67X

Osc Type	Resonator Freq	Cap. Range C1	Cap. Range C2
XT	455 kHz 2.0 MHz 4.0 MHz	22-100 pF 15-68 pF 15-68 pF	22-100 pF 15-68 pF 15-68 pF
HS	4.0 MHz 8.0 MHz 10.0 MHz	15-68 pF 10-68 pF 10-22 pF	15-68 pF 10-68 pF 10-22 pF

These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

TABLE 9-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR - PIC12C67X

Osc Type	Resonator Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz(1) 100 kHz 200 kHz	15 pF 15-30 pF 15-30 pF	15 pF 30-47 pF 15-82 pF
XT	100 kHz 200 kHz 455 kHz 1 MHz 2 MHz 4 MHz	15-30 pF 15-30 pF 15-30 pF 15-30 pF 15-30 pF 15-47 pF	200-300 pF 100-200 pF 15-100 pF 15-30 pF 15-30 pF 15-47 pF
HS	4 MHz 8 MHz 10 MHz	15-30 pF 15-30 pF 15-30 pF	15-30 pF 15-30 pF 15-30 pF

Note 1: For V_{DD} > 4.5V, C1 = C2 ≈ 30 pF is recommended.

These values are for design guidance only. Rs may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

9.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a pre-packaged oscillator or a simple oscillator circuit with TTL gates can be used as an external crystal oscillator circuit. Pre-packaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used; one with parallel resonance or one with series resonance.

Figure 9-3 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometers bias the 74AS04 in the linear region. This circuit could be used for external oscillator designs.

FIGURE 9-3: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

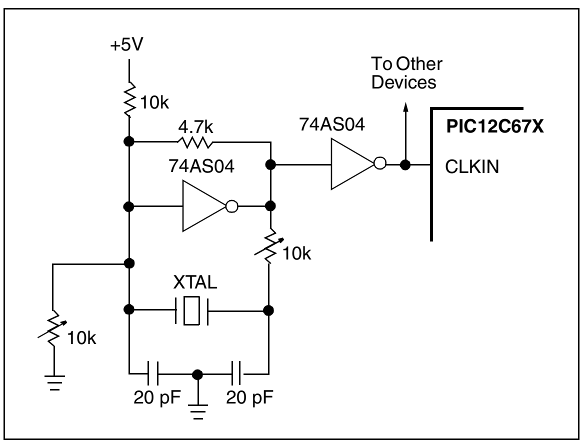
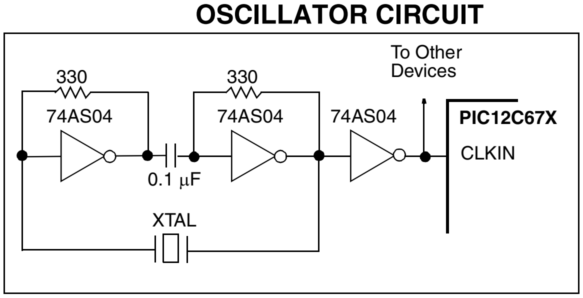


Figure 9-4 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 Ω resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 9-4: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



9.2.4 EXTERNAL RC OSCILLATOR

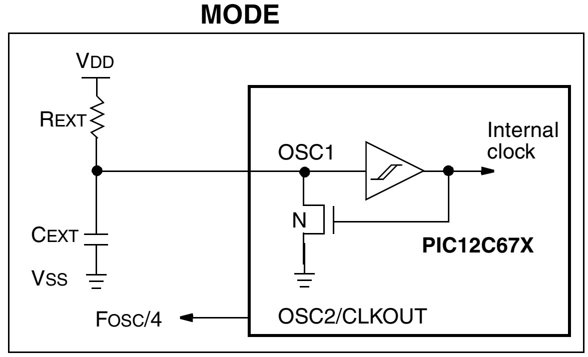
For timing insensitive applications, the RC device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (R_{EXT}) and capacitor (C_{EXT}) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low C_{EXT} values. The user also needs to take into account variation due to tolerance of external R and C components used.

Figure 9-5 shows how the R/C combination is connected to the PIC12C67X. For R_{EXT} values below 2.2 kΩ, the oscillator operation may become unstable or stop completely. For very high R_{EXT} values (i.e., 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend keeping R_{EXT} between 3 kΩ and 100 kΩ.

Although the oscillator will operate with no external capacitor (C_{EXT} = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

FIGURE 9-5: EXTERNAL RC OSCILLATOR MODE



9.2.5 INTERNAL 4 MHz RC OSCILLATOR

The internal RC oscillator provides a fixed 4 MHz (nominal) system clock at V_{DD} = 5V and 25 °C. See Section 13.0 for information on variation over voltage and temperature.

In addition, a calibration instruction is programmed into the last address of the program memory which contains the calibration value for the internal RC oscillator. This value is programmed as a `RETLW XX` instruction where XX is the calibration value. In order to retrieve the calibration value, issue a `CALL YY` instruction where YY is the last location in program memory (03FFh for the PIC12C671 and the PIC12CE673, 07FFh for the PIC12C672 and the PIC12CE674). Control will be returned to the user's program with the calibration value loaded into the W register. The program should then perform a `MOVWF OSCCAL` instruction to load the value into the internal RC oscillator trim register.

OSCCAL, when written to with the calibration value, will "trim" the internal oscillator to remove process variation from the oscillator frequency. Bits <7:4>, CAL<3:0> are used for fine calibration, while bit 3, CALFST, and bit 2, CALSLW, are used for more coarse adjustment. Adjusting CAL<3:0> from 0000 to 1111 yields a higher clock speed. Set CALFST = 1 for greater increase in frequency or set CALSLW = 1 for greater decrease in frequency. Note that bits 1 and 0 of OSCCAL are unimplemented and should be written as 0 when modifying OSCCAL for compatibility with future devices.

Note:	Please note that erasing the device will also erase the pre-programmed internal calibration value for the internal oscillator. The calibration value must be saved prior to erasing the part.

9.2.6 CLKOUT

The PIC12C67X can be configured to provide a clock out signal (CLKOUT) on pin 3 when the configuration word address (2007h) is programmed with FOSC2, FOSC1, and FOSC0, equal to 101 for INTRC or 111 for EXTRC. The oscillator frequency, divided by 4, can be used for test purposes or to synchronize other logic.

9.3 Reset

The PIC12C67X differentiates between various kinds of reset:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset during normal operation
- $\overline{\text{MCLR}}$ Reset during SLEEP
- WDT Reset (normal operation)

Some registers are not affected in any reset condition; their status is unknown on POR and unchanged in any other reset. Most other registers are reset to a "reset state" on Power-on Reset (POR), $\overline{\text{MCLR}}$ Reset, WDT Reset, and $\overline{\text{MCLR}}$ Reset during SLEEP. They are not affected by a WDT Wake-up, which is viewed as the resumption of normal operation. The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different reset situations, as indicated in Table 9-5. These bits are used in software to determine the nature of the reset. See Table 9-6 for a full description of reset states of all registers.

A simplified block diagram of the on-chip reset circuit is shown in Figure 9-6.

The PIC12C67X has a $\overline{\text{MCLR}}$ noise filter in the $\overline{\text{MCLR}}$ reset path. The filter will detect and ignore small pulses.

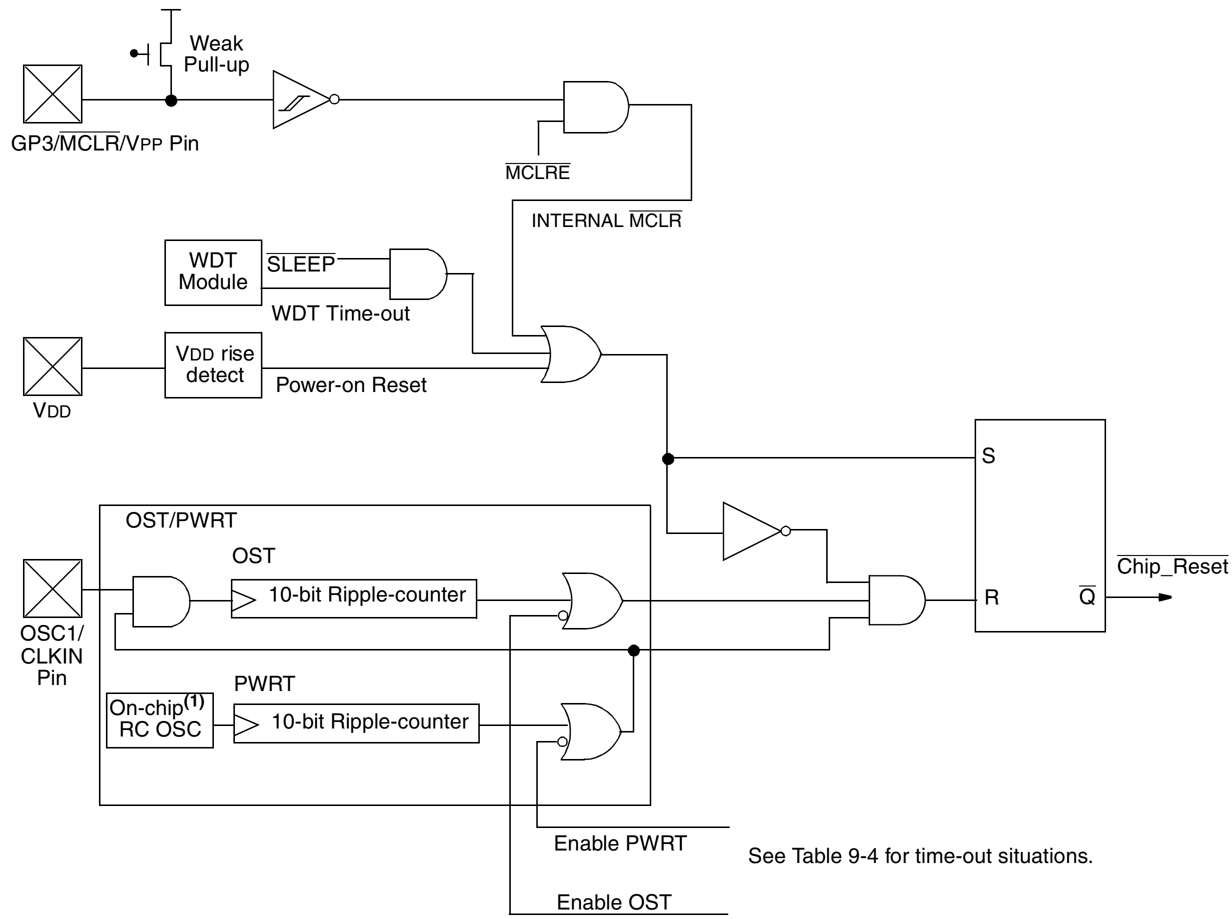
It should be noted that a WDT Reset does not drive $\overline{\text{MCLR}}$ pin low.

When MCLR is asserted, the state of the OSC1/CLKIN and CLKOUT/OSC2 pins are as follows:

TABLE 9-3: CLKIN/CLKOUT PIN STATES WHEN $\overline{\text{MCLR}}$ ASSERTED

Oscillator Mode	OSC1/CLKIN Pin	OSC2/CLKout Pin
EXTRC, CLKOUT on OSC2	OSC1 pin is tristated and driven by external circuit	OSC2 pin is driven low
EXTRC, OSC2 is I/O	OSC1 pin is tristated and driven by external circuit	OSC2 pin is tristate input
INTRC, CLKOUT on OSC2	OSC1 pin is tristate input	OSC2 pin is driven low
INTRC, OSC2 is I/O	OSC1 pin is tristate input	OSC2 pin is tristate input

FIGURE 9-6: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



Note 1: This is a separate oscillator from the RC oscillator of the CLKIN pin.

9.4 Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)

9.4.1 POWER-ON RESET (POR)

The on-chip POR circuit holds the chip in reset until V_{DD} has reached a high enough level for proper operation. To take advantage of the POR, just tie the $\overline{\text{MCLR}}$ pin through a resistor to V_{DD}. This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for V_{DD} is specified. See Electrical Specifications for details.

When the device starts normal operation (exits the reset condition), device operating parameters (voltage, frequency, temperature, . ..) must be met to ensure operation. If these conditions are not met, the device must be held in reset until the operating conditions are met.

For additional information, refer to Application Note AN607, "*Power-up Trouble Shooting*."

9.4.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 72 ms nominal time-out on power-up only, from the POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in reset as long as the PWRT is active. The PWRT's time delay allows V_{DD} to rise to an acceptable level. A configuration bit is provided to enable/disable the PWRT.

The power-up time delay will vary from chip to chip due to V_{DD}, temperature and process variation. See Table 11-4.

9.4.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP and HS modes and only on Power-on Reset or wake-up from SLEEP.

9.4.4 TIME-OUT SEQUENCE

On power-up, the Time-out Sequence is as follows: first, PWRT time-out is invoked after the POR time delay has expired; then, OST is activated. The total time-out will vary, based on oscillator configuration and the status of the PWRT. For example, in RC mode with the PWRT disabled, there will be no time-out at all. Figure 9-7, Figure 9-8, and Figure 9-9 depict time-out sequences on power-up.

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{\text{MCLR}}$ high will begin execution immediately (Figure 9-9). This is useful for testing purposes or to synchronize more than one PIC12C67X device operating in parallel.

9.4.5 POWER CONTROL (PCON)/STATUS REGISTER

The Power Control/Status Register, PCON (address 8Eh), has one bit. See Register 4-6 for register.

Bit1 is $\overline{\text{POR}}$ (Power-on Reset). It is cleared on a Power-on Reset and is unaffected otherwise. The user sets this bit following a Power-on Reset. On subsequent resets, if POR is '0', it will indicate that a Power-on Reset must have occurred.

TABLE 9-4: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up		Wake-up from SLEEP
	$\overline{\text{PWRTE}}$ = 0	$\overline{\text{PWRTE}}$ = 1	
XT, HS, LP	72 ms + 1024TOSC	1024TOSC	1024TOSC
INTRC, EXTRC	72 ms	—	—

TABLE 9-5: STATUS/PCON BITS AND THEIR SIGNIFICANCE

$\overline{\text{POR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	
0	1	1	Power-on Reset
0	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
1	0	u	WDT Reset
1	0	0	WDT Wake-up
1	u	u	$\overline{\text{MCLR}}$ Reset during normal operation
1	1	0	$\overline{\text{MCLR}}$ Reset during SLEEP or interrupt wake-up from SLEEP

Legend: u = unchanged, x = unknown.

TABLE 9-6: RESET CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	000h	0001 1xxx	---- --0-
$\overline{\text{MCLR}}$ Reset during normal operation	000h	000u uuuu	---- --u-
$\overline{\text{MCLR}}$ Reset during SLEEP	000h	0001 0uuu	---- --u-
WDT Reset during normal operation	000h	0000 uuuu	---- --u-
WDT Wake-up from SLEEP	PC + 1	uuu0 0uuu	---- --u-
Interrupt wake-up from SLEEP	PC + 1(1)	uuu1 0uuu	---- --u-

Legend: u = unchanged, x = unknown, - = unimplemented bit read as '0'.
Note 1: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

TABLE 9-7: INITIALIZATION CON\DITIONS FOR ALL REGISTERS

Register	Power-on Reset	$\overline{\text{MCLR}}$ Resets WDT Reset	Wake-up via WDT or Interrupt
W	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	0000 0000	0000 0000	0000 0000
TMR0	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	PC + 1(2)
STATUS	0001 1xxx	000q quuu(3)	uuuq quuu(3)
FSR	xxxx xxxx	uuuu uuuu	uuuu uuuu
GPIO PIC12CE67X	11xx xxxx	11uu uuuu	11uu uuuu
GPIO PIC12C67X	--xx xxxx	--uu uuuu	--uu uuuu
PCLATH	---0 0000	---0 0000	---u uuuu
INTCON	0000 000x	0000 000u	uuuu uqqq(1)
PIR1	-0-- ----	-0-- ----	-q-- ----(4)
ADCON0	0000 0000	0000 0000	uuuu uquu(5)
OPTION	1111 1111	1111 1111	uuuu uuuu
TRIS	--11 1111	--11 1111	--uu uuuu
PIE1	-0-- ----	-0-- ----	-u-- ----
PCON	---- --0-	---- --u-	---- --u-
OSCCAL	0111 00--	uuuu uu--	uuuu uu--
ADCON1	---- -000	---- -000	---- -uuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition.
Note 1: One or more bits in INTCON and PIR1 will be affected (to cause wake-up).
2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).
3: See Table 9-5 for reset value for specific condition.
4: If wake-up was due to A/D completing then bit 6 = 1, all other interrupts generating a wake-up will cause bit 6 = u.
5: If wake-up was due to A/D completing then bit 3 = 0, all other interrupts generating a wake-up will cause bit 3 = u.

FIGURE 9-7: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO VDD): CASE 1

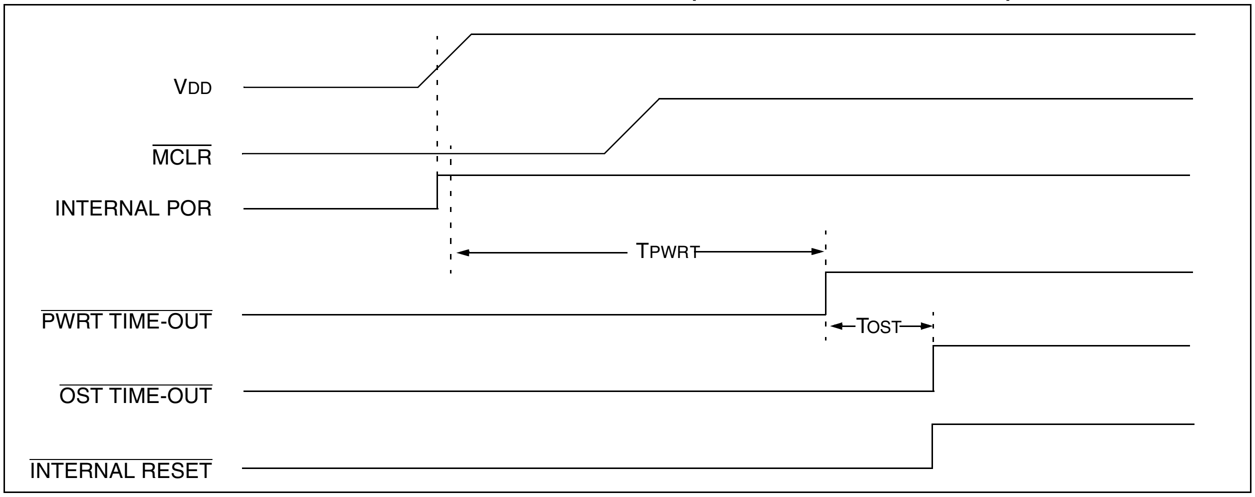


FIGURE 9-8: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO VDD): CASE 2

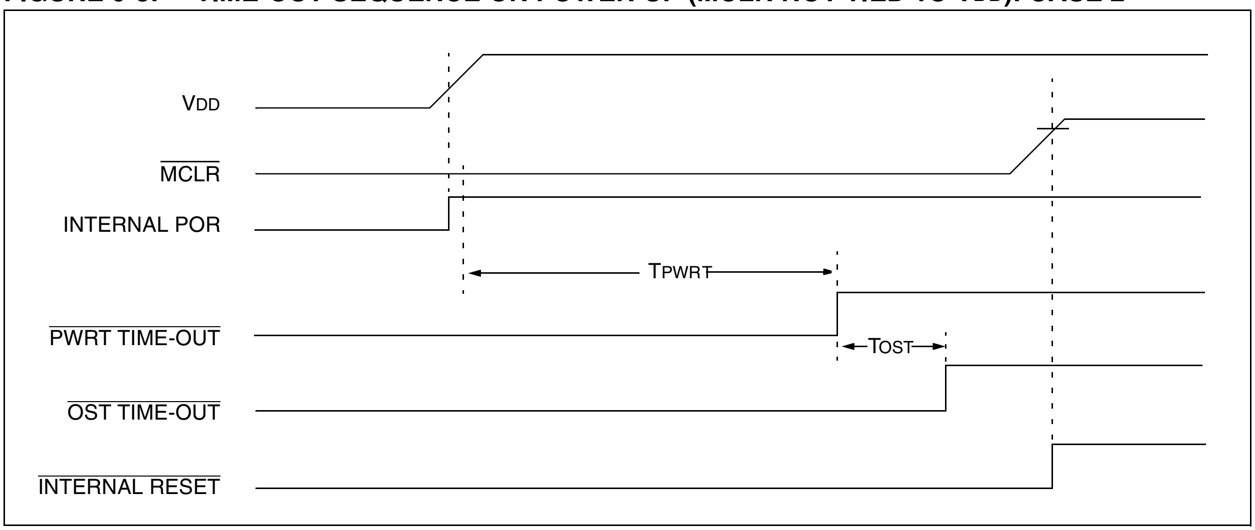


FIGURE 9-9: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ TIED TO VDD)

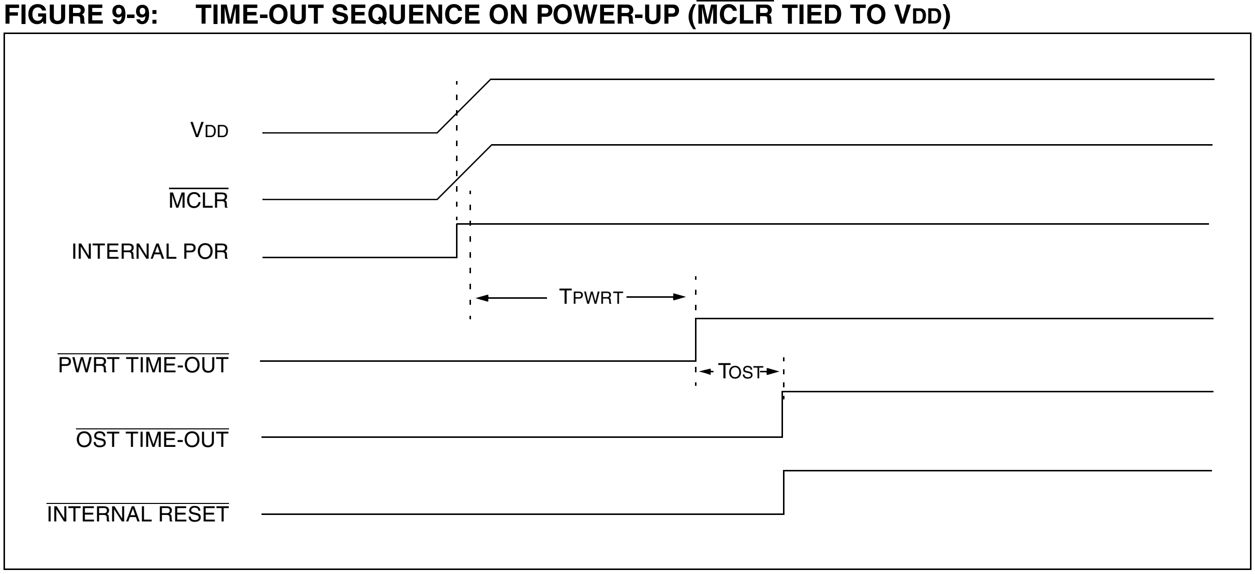
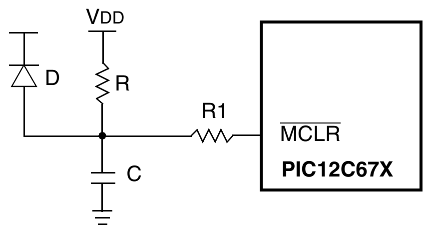


FIGURE 9-10: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)

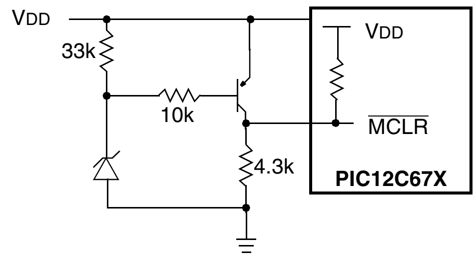


Note 1: External Power-on Reset circuit is required only if VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: R < 40 kΩ is recommended to make sure that voltage drop across R does not violate the device's electrical specification.

3: R1 = 100Ω to 1 kΩ will limit any current flowing into $\overline{\text{MCLR}}$ from external capacitor C, in the event of $\overline{\text{MCLR}}$/VPP pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

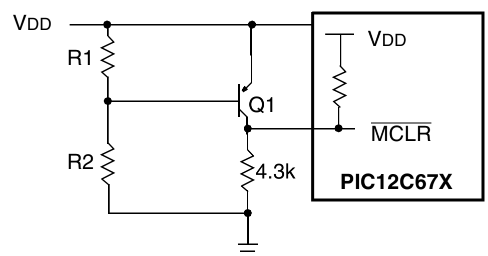
FIGURE 9-11: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 1



Note 1: This circuit will activate reset when VDD goes below (Vz + 0.7V), where Vz = Zener voltage.

2: Resistors should be adjusted for the characteristics of the transistor.

FIGURE 9-12: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 2



Note 1: This brown-out circuit is less expensive, albeit less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \bullet \frac{R1}{R1 + R2} = 0.7V$$

2: Resistors should be adjusted for the characteristics of the transistor.

9.5 Interrupts

There are four sources of interrupt:

Interrupt Sources
TMR0 Overflow Interrupt
External Interrupt GP2/INT pin
GPIO Port Change Interrupts (pins GP0, GP1, GP3)
A/D Interrupt

The Interrupt Control Register (INTCON) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

> **Note:** Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit.

A global interrupt enable bit, GIE (INTCON<7>), enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. When bit GIE is enabled and an interrupt's flag bit and mask bit are set, the interrupt will vector immediately. Individual interrupts can be disabled through their corresponding enable bits in various registers. Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit. The GIE bit is cleared on reset.

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

The GP2/INT, GPIO port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flag ADIF, is contained in the Special Function Register PIR1. The corresponding interrupt enable bit is contained in Special Function Register PIE1, and the peripheral interrupt enable bit is contained in Special Function Register INTCON.

When an interrupt is responded to, the GIE bit is cleared to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with 0004h. 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 repeated interrupts.

For external interrupt events, such as GPIO change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends on when the interrupt event occurs (Figure 9-14). The latency is the same for one or two cycle instructions. Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit.

FIGURE 9-13: INTERRUPT LOGIC

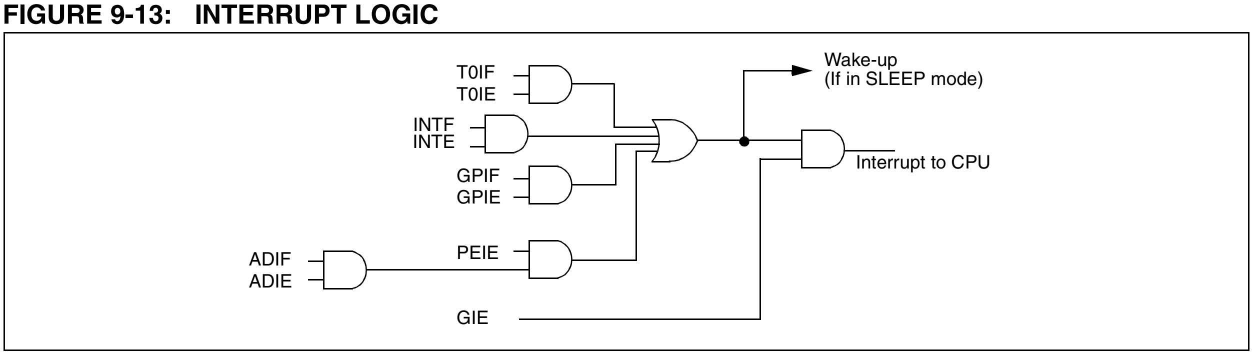
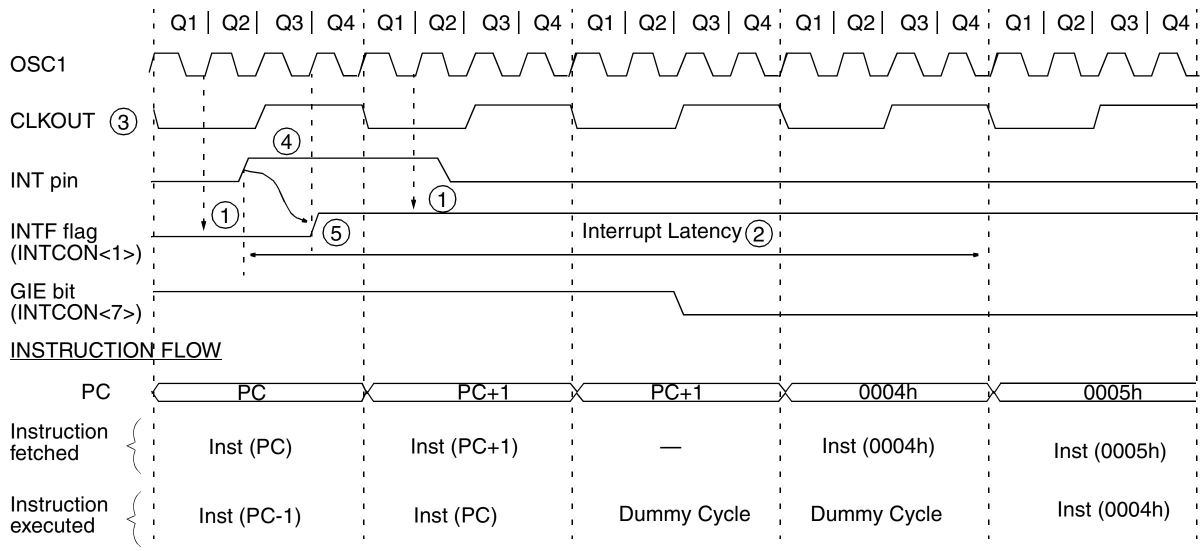


FIGURE 9-14: INT PIN INTERRUPT TIMING



Note
1: INTF flag is sampled here (every Q1).
2: Interrupt latency = 3-4 TCY where TCY = instruction cycle time.
Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.
3: CLKOUT is available only in INTRC and EXTRC oscillator modes.
4: For minimum width of INT pulse, refer to AC specs.
5: INTF is enabled to be set anytime during the Q4-Q1 cycles.

9.5.1 TMR0 INTERRUPT

An overflow (FFh $\rightarrow$ 00h) in the TMR0 register will set flag bit T0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit T0IE (INTCON<5>) (Section 7.0). The flag bit T0IF (INTCON<2>) will be set, regardless of the state of the enable bits. If used, this flag must be cleared in software.

9.5.2 INT INTERRUPT

External interrupt on GP2/INT pin is edge triggered; either rising if bit INTEDG (OPTION<6>) is set, or falling, if the INTEDG bit is clear. When a valid edge appears on the GP2/INT pin, flag bit INTF (INTCON<1>) is set. This interrupt can be disabled by clearing enable bit INTE (INTCON<4>). Flag bit INTF must be cleared in software in the interrupt service routine before re-enabling this interrupt. The INT interrupt can wake-up the processor from SLEEP, if bit INTE was set prior to going into SLEEP. The status of global interrupt enable bit GIE decides whether or not the processor branches to the interrupt vector following wake-up. See Section 9.8 for details on SLEEP mode.

9.5.3 GPIO INTCON CHANGE

An input change on GP3, GP1 or GP0 sets flag bit GPIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit GPIE (INTCON<3>) (Section 5.1). This flag bit GPIF (INTCON<0>) will be set, regardless of the state of the enable bits. If used, this flag must be cleared in software.

9.6 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (i.e., W register and STATUS register). This will have to be implemented in software.

Example 9-1 shows the storing and restoring of the STATUS and W registers. The register, W_TEMP, must be defined in both banks and must be defined at the same offset from the bank base address (i.e., if W_TEMP is defined at 0x20 in bank 0, it must also be defined at 0xA0 in bank 1).

Example 9-2 shows the saving and restoring of STATUS and W using RAM locations 0x70 - 0x7F. W_TEMP is defined at 0x70 and STATUS_TEMP is defined at 0x71.

The example:

a) Stores the W register.
b) Stores the STATUS register in bank 0.
c) Executes the ISR code.
d) Restores the STATUS register (and bank select bit).
e) Restores the W register.
f) Returns from interrupt.

EXAMPLE 9-1: SAVING STATUS AND W REGISTERS USING GENERAL PURPOSE RAM (0x20 - 0x6F)

```
MOVWF    W_TEMP           ;Copy W to TEMP register, could be bank one or zero
SWAPF    STATUS,W         ;Swap status to be saved into W
BCF      STATUS,RP0       ;Change to bank zero, regardless of current bank
MOVWF    STATUS_TEMP      ;Save status to bank zero STATUS_TEMP register
:
:(ISR)
:
SWAPF    STATUS_TEMP,W    ;Swap STATUS_TEMP register into W
                          ;(sets bank to original state)
MOVWF    STATUS           ;Move W into STATUS register
SWAPF    W_TEMP,F         ;Swap W_TEMP
SWAPF    W_TEMP,W         ;Swap W_TEMP into W
RETFIE                    ;Return from interrupt
```

EXAMPLE 9-2: SAVING STATUS AND W REGISTERS USING SHARED RAM (0x70 - 0x7F)

```
MOVWF    W_TEMP           ;Copy W to TEMP register (bank independent)
MOVF     STATUS,W         ;Move STATUS register into W
MOVWF    STATUS_TEMP      ;Save contents of STATUS register
:
:(ISR)
:
MOVF     STATUS_TEMP,W    ;Retrieve copy of STATUS register
MOVWF    STATUS           ;Restore pre-isr STATUS register contents
SWAPF    W_TEMP,F         ;
SWAPF    W_TEMP,W         ;Restore pre-isr W register contents
RETFIE                    ;Return from interrupt
```

9.7 Watchdog Timer (WDT)

The Watchdog Timer is a free running, on-chip RC oscillator, which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run, even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins of the device has been stopped, for example, by execution of a `SLEEP` instruction. During normal operation, a WDT time-out generates a device RESET (Watchdog Timer Reset). If the device is in SLEEP mode, a WDT time-out causes the device to wake-up and continue with normal operation (Watchdog Timer Wake-up). The WDT can be permanently disabled by clearing configuration bit WDTE (Section 9.1).

9.7.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms (with no prescaler). The time-out periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION register. Thus, time-out periods up to 2.3 seconds can be realized.

The `CLRWDT` and `SLEEP` instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out early and generating a premature device RESET condition.

The $\overline{TO}$ bit in the STATUS register will be cleared upon a Watchdog Timer time-out.

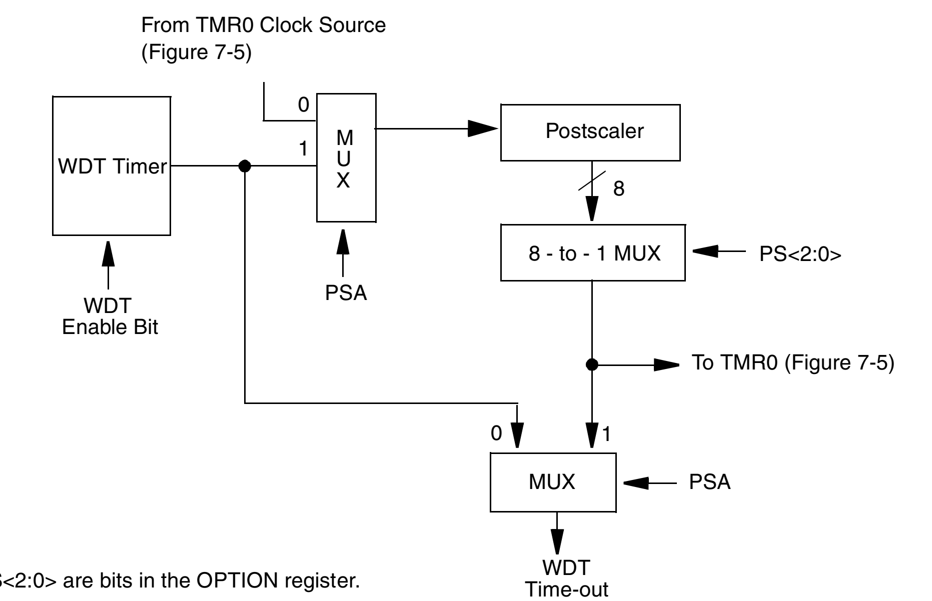
9.7.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken into account that under worst case conditions (VDD = Min., Temperature = Max., and max. WDT prescaler), it may take several seconds before a WDT time-out occurs.

Note:	When the prescaler is assigned to the WDT, always execute a `CLRWDT` instruction before changing the prescale value, otherwise a WDT reset may occur.

See Example 7-1 and Example 7-2 for changing prescaler between WDT and Timer0.

FIGURE 9-15: WATCHDOG TIMER BLOCK DIAGRAM



Note: PSA and PS<2:0> are bits in the OPTION register.

TABLE 9-8: SUMMARY OF WATCHDOG TIMER REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits(1)	MCLRE	CP1	CP0	$\overline{PWRTE}$	$\overline{WDTE}$	FOSC2	FOSC1	FOSC0
81h	OPTION	$\overline{GPPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

Legend: Shaded cells are not used by the Watchdog Timer.
Note 1: See Register 9-1 for operation of these bits. Not all CP0 and CP1 bits are shown.

9.8 Power-down Mode (SLEEP)

Power-down mode is entered by executing a `SLEEP` instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{PD}$ bit (STATUS<3>) is cleared, the $\overline{TO}$ (STATUS<4>) bit is set, and the oscillator driver is turned off. The I/O ports maintain the status they had, before the `SLEEP` instruction was executed (driving high, low or hi-impedance).

For lowest current consumption in this mode, place all I/O pins at either V_{DD} or V_{SS}, ensure no external circuitry is drawing current from the I/O pin, power-down the A/D, and disable external clocks. Pull all I/O pins that are hi-impedance inputs, high or low externally, to avoid switching currents caused by floating inputs. The T0CKI input, if enabled, should also be at V_{DD} or V_{SS} for lowest current consumption. The contribution from on-chip pull-ups on GPIO should be considered.

The $\overline{MCLR}$ pin, if enabled, must be at a logic high level (V_{IHMC}).

9.8.1 WAKE-UP FROM SLEEP

The device can wake-up from SLEEP through one of the following events:

1. External reset input on $\overline{MCLR}$ pin.
2. Watchdog Timer Wake-up (if WDT was enabled).
3. GP2/INT interrupt, interrupt GPIO port change or some Peripheral Interrupts.

External $\overline{MCLR}$ Reset will cause a device reset. All other events are considered a continuation of program execution and cause a "wake-up". The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register can be used to determine the cause of device reset. The $\overline{PD}$ bit, which is set on power-up, is cleared when `SLEEP` is invoked. The $\overline{TO}$ bit is cleared if a WDT time-out occurred (and caused wake-up).

The following peripheral interrupt can wake the device from SLEEP:

1. A/D conversion (when A/D clock source is RC).

Other peripherals can not generate interrupts since during SLEEP, no on-chip Q clocks are present.

When the `SLEEP` instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the `SLEEP` instruction. If the GIE bit is set (enabled), the device executes the instruction after the `SLEEP` instruction and then branches to the interrupt address (0004h). In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have a `NOP` after the `SLEEP` instruction.

9.8.2 WAKE-UP USING INTERRUPTS

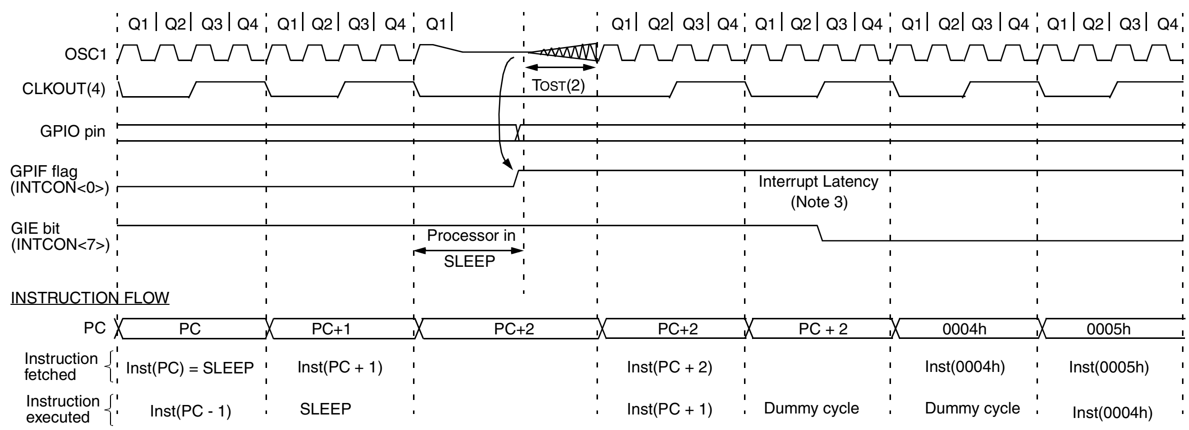
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the the execution of a `SLEEP` instruction, the `SLEEP` instruction will complete as a `NOP`. Therefore, the WDT and WDT postscaler will not be cleared, the $\overline{TO}$ bit will not be set and $\overline{PD}$ bits will not be cleared.
- If the interrupt occurs **during or after** the execution of a `SLEEP` instruction, the device will immediately wake-up from sleep . The `SLEEP` instruction will be completely executed before the wake-up. Therefore, the WDT and WDT postscaler will be cleared, the $\overline{TO}$ bit will be set and the $\overline{PD}$ bit will be cleared.

Even if the flag bits were checked before executing a `SLEEP` instruction, it may be possible for flag bits to become set before the `SLEEP` instruction completes. To determine whether a `SLEEP` instruction executed, test the $\overline{PD}$ bit. If the $\overline{PD}$ bit is set, the `SLEEP` instruction was executed as a `NOP`.

To ensure that the WDT is cleared, a `CLRWDT` instruction should be executed before a `SLEEP` instruction.

FIGURE 9-16: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: XT, HS or LP oscillator mode assumed.
2: TOST = 1024TOSC (drawing not to scale) This delay will not be there for INTRC and EXTRC osc mode.
3: GIE = '1' assumed. In this case after wake- up, the processor jumps to the interrupt routine. If GIE = '0', execution will continue in-line.
4: CLKOUT is not available in XT, HS or LP osc modes, but shown here for timing reference.

9.9 Program Verification/Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

Note:	Microchip does not recommend code protecting windowed devices.

9.10 ID Locations

Four memory locations (2000h - 2003h) are designated as ID locations, where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution, but are readable and writable during program/verify. It is recommended that only the 4 least significant bits of the ID location are used.

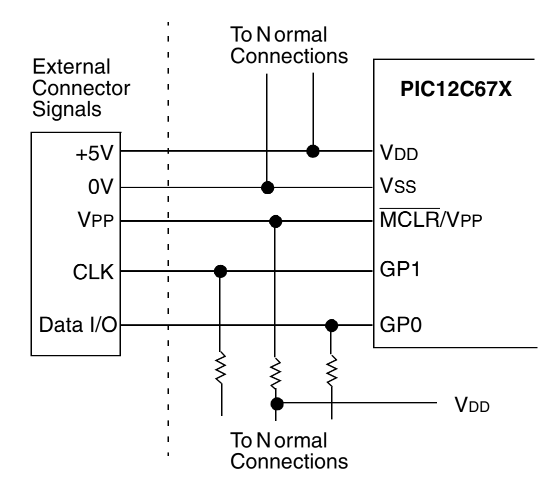
9.11 In-Circuit Serial Programming

PIC12C67X microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a program/verify mode by holding the GP1 and GP0 pins low, while raising the $\overline{MCLR}$ (V_{PP}) pin from V_{IL} to V_{IHH} (see programming specification). GP1 (clock) becomes the programming clock and GP0 (data) becomes the programming data. Both GP0 and GP1 are Schmitt Trigger inputs in this mode.

After reset, and if the device is placed into programming/verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14-bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the PIC12C67X Programming Specifications.

FIGURE 9-17: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION



NOTES:

10.0 INSTRUCTION SET SUMMARY

Each PIC12C67X instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC12C67X instruction set summary in Table 10-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 10-1 shows the opcode field descriptions.

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

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

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

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 10-1: OPCODE FIELD DESCRIPTIONS

Field	Description
`f`	Register file address (0x00 to 0x7F)
`W`	Working register (accumulator)
`b`	Bit address within an 8-bit file register
`k`	Literal field, constant data or label
`x`	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
`d`	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
`label`	Label name
`TOS`	Top of Stack
`PC`	Program Counter
`PCLATH`	Program Counter High Latch
`GIE`	Global Interrupt Enable bit
`WDT`	Watchdog Timer/Counter
$\overline{\texttt{TO}}$	Time-out bit
$\overline{\texttt{PD}}$	Power-down bit
`dest`	Destination either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

The instruction set is highly orthogonal and is grouped into three basic categories:

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

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs.

Table 10-2 lists the instructions recognized by the MPASM assembler.

Figure 10-1 shows the three general formats that the instructions can have.

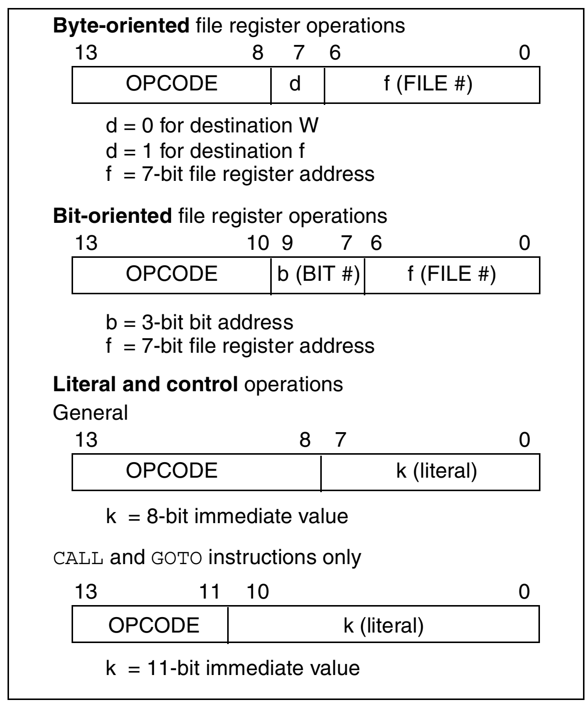
> **Note:** To maintain upward compatibility with future PIC12C67X products, do not use the `OPTION` and `TRIS` instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 10-1: GENERAL FORMAT FOR INSTRUCTIONS



10.1 Special Function Registers as Source/Destination

The PIC12C67X's orthogonal instruction set allows read and write of all file registers, including special function registers. There are some special situations the user should be aware of:

10.1.1 STATUS AS DESTINATION

If an instruction writes to STATUS, the Z, C and DC bits may be set or cleared as a result of the instruction and overwrite the original data bits written. For example, executing `CLRF STATUS` will clear register STATUS, and then set the Z bit leaving `0000 0100b` in the register.

10.1.2 TRIS AS DESTINATION

Bit 3 of the TRIS register always reads as a '1' since GP3 is an input only pin. This fact can affect some read-modify-write operations on the TRIS register.

10.1.3 PCL AS SOURCE OR DESTINATION

Read, write or read-modify-write on PCL may have the following results:

Read PC:	PCL $\rightarrow$ dest
Write PCL:	PCLATH $\rightarrow$ PCH; 8-bit destination value $\rightarrow$ PCL
Read-Modify-Write:	PCL$\rightarrow$ ALU operand PCLATH $\rightarrow$ PCH; 8-bit result $\rightarrow$ PCL

Where PCH = program counter high byte (not an addressable register), PCLATH = Program counter high holding latch, dest = destination, WREG or f.

10.1.4 BIT MANIPULATION

All bit manipulation instructions are done by first reading the entire register, operating on the selected bit and writing the result back (read-modify-write). The user should keep this in mind when operating on special function registers, such as ports.

TABLE 10-2: INSTRUCTION SET SUMMARY

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb			LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	**f, d**	Add W and f	1	`00`	`0111`	`dfff`	`ffff`	C,DC,Z	1,2
ANDWF	**f, d**	AND W with f	1	`00`	`0101`	`dfff`	`ffff`	Z	1,2
CLRF	**f**	Clear f	1	`00`	`0001`	`lfff`	`ffff`	Z	2
CLRW	**-**	Clear W	1	`00`	`0001`	`0000`	`0011`	Z	
COMF	**f, d**	Complement f	1	`00`	`1001`	`dfff`	`ffff`	Z	1,2
DECF	**f, d**	Decrement f	1	`00`	`0011`	`dfff`	`ffff`	Z	1,2
DECFSZ	**f, d**	Decrement f, Skip if 0	1(2)	`00`	`1011`	`dfff`	`ffff`		1,2,3
INCF	**f, d**	Increment f	1	`00`	`1010`	`dfff`	`ffff`	Z	1,2
INCFSZ	**f, d**	Increment f, Skip if 0	1(2)	`00`	`1111`	`dfff`	`ffff`		1,2,3
IORWF	**f, d**	Inclusive OR W with f	1	`00`	`0100`	`dfff`	`ffff`	Z	1,2
MOVF	**f, d**	Move f	1	`00`	`1000`	`dfff`	`ffff`	Z	1,2
MOVWF	**f**	Move W to f	1	`00`	`0000`	`lfff`	`ffff`		
NOP	**-**	No Operation	1	`00`	`0000`	`0xx0`	`0000`		
RLF	**f, d**	Rotate Left f through Carry	1	`00`	`1101`	`dfff`	`ffff`	C	1,2
RRF	**f, d**	Rotate Right f through Carry	1	`00`	`1100`	`dfff`	`ffff`	C	1,2
SUBWF	**f, d**	Subtract W from f	1	`00`	`0010`	`dfff`	`ffff`	C,DC,Z	1,2
SWAPF	**f, d**	Swap nibbles in f	1	`00`	`1110`	`dfff`	`ffff`		1,2
XORWF	**f, d**	Exclusive OR W with f	1	`00`	`0110`	`dfff`	`ffff`	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	**f, b**	Bit Clear f	1	`01`	`00bb`	`bfff`	`ffff`		1,2
BSF	**f, b**	Bit Set f	1	`01`	`01bb`	`bfff`	`ffff`		1,2
BTFSC	**f, b**	Bit Test f, Skip if Clear	1 (2)	`01`	`10bb`	`bfff`	`ffff`		3
BTFSS	**f, b**	Bit Test f, Skip if Set	1 (2)	`01`	`11bb`	`bfff`	`ffff`		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	**k**	Add literal and W	1	`11`	`111x`	`kkkk`	`kkkk`	C,DC,Z	
ANDLW	**k**	AND literal with W	1	`11`	`1001`	`kkkk`	`kkkk`	Z	
CALL	**k**	Call subroutine	2	`10`	`0kkk`	`kkkk`	`kkkk`		
CLRWDT	**-**	Clear Watchdog Timer	1	`00`	`0000`	`0110`	`0100`	$\overline{TO}$,$\overline{PD}$	
GOTO	**k**	Go to address	2	`10`	`1kkk`	`kkkk`	`kkkk`		
IORLW	**k**	Inclusive OR literal with W	1	`11`	`1000`	`kkkk`	`kkkk`	Z	
MOVLW	**k**	Move literal to W	1	`11`	`00xx`	`kkkk`	`kkkk`		
RETFIE	**-**	Return from interrupt	2	`00`	`0000`	`0000`	`1001`		
RETLW	**k**	Return with literal in W	2	`11`	`01xx`	`kkkk`	`kkkk`		
RETURN	**-**	Return from Subroutine	2	`00`	`0000`	`0000`	`1000`		
SLEEP	**-**	Go into standby mode	1	`00`	`0000`	`0110`	`0011`	$\overline{TO}$,$\overline{PD}$	
SUBLW	**k**	Subtract W from literal	1	`11`	`110x`	`kkkk`	`kkkk`	C,DC,Z	
XORLW	**k**	Exclusive OR literal with W	1	`11`	`1010`	`kkkk`	`kkkk`	Z	

Note 1: When an I/O register is modified as a function of itself (i.e., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

10.2 Instruction Descriptions

ADDLW — Add Literal and W

Syntax:	[*label*] ADDLW k
Operands:	$0 \le k \le 255$
Operation:	$(W) + k \rightarrow (W)$
Status Affected:	C, DC, Z
Encoding:	`11` | `111x` | `kkkk` | `kkkk`
Description:	The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register.
Words:	1
Cycles:	1
Example	`ADDLW 0x15` Before Instruction W = 0x10 After Instruction W = 0x25

ADDWF — Add W and f

Syntax:	[*label*] ADDWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	$(W) + (f) \rightarrow (dest)$
Status Affected:	C, DC, Z
Encoding:	`00` | `0111` | `dfff` | `ffff`
Description:	Add the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example	`ADDWF FSR, 0` Before Instruction W = 0x17 FSR = 0xC2 After Instruction W= 0xD9 FSR = 0xC2

ANDLW — And Literal with W

Syntax:	[*label*] ANDLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .AND. $(k) \rightarrow (W)$
Status Affected:	Z
Encoding:	`11` | `1001` | `kkkk` | `kkkk`
Description:	The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.
Words:	1
Cycles:	1
Example	`ANDLW 0x5F` Before Instruction W= 0xA3 After Instruction W = 0x03

ANDWF — AND W with f

Syntax:	[*label*] ANDWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .AND. $(f) \rightarrow (dest)$
Status Affected:	Z
Encoding:	`00` | `0101` | `dfff` | `ffff`
Description:	AND the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example	`ANDWF FSR, 1` Before Instruction W = 0x17 FSR = 0xC2 After Instruction W = 0x17 FSR = 0x02

BCF	Bit Clear f			
Syntax:	[*label*] BCF f,b			
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$			
Operation:	$0 \rightarrow$ (f<b>)			
Status Affected:	None			
Encoding:	`01`	`00bb`	`bfff`	`ffff`
Description:	Bit 'b' in register 'f' is cleared.			
Words:	1			
Cycles:	1			
Example	`BCF FLAG_REG, 7` Before Instruction FLAG_REG = 0xC7 After Instruction FLAG_REG = 0x47			

BSF	Bit Set f			
Syntax:	[*label*] BSF f,b			
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$			
Operation:	$1 \rightarrow$ (f<b>)			
Status Affected:	None			
Encoding:	`01`	`01bb`	`bfff`	`ffff`
Description:	Bit 'b' in register 'f' is set.			
Words:	1			
Cycles:	1			
Example	`BSF FLAG_REG, 7` Before Instruction FLAG_REG = 0x0A After Instruction FLAG_REG = 0x8A			

BTFSC	Bit Test, Skip if Clear			
Syntax:	[*label*] BTFSC f,b			
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$			
Operation:	skip if (f<b>) = 0			
Status Affected:	None			
Encoding:	`01`	`10bb`	`bfff`	`ffff`
Description:	If bit 'b' in register 'f' is '0', then the next instruction is skipped. If bit 'b' is '0', then the next instruction fetched during the current instruction execution is discarded, and a `NOP` is executed instead, making this a 2 cycle instruction.			
Words:	1			
Cycles:	1(2)			

Example	`HERE BTFSC FLAG,1`
	`FALSE GOTO PROCESS_CODE`
	`TRUE •`
	`•`
	`•`
	Before Instruction
	PC = address `HERE`
	After Instruction
	if FLAG<1> = 0,
	PC = address `TRUE`
	if FLAG<1>=1,
	PC = address `FALSE`			

BTFSS — Bit Test f, Skip if Set

Syntax: [*label*] BTFSS f,b

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

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

Status Affected: None

Encoding:

01	11bb	bfff	ffff

Description: If bit 'b' in register 'f' is '1', then the next instruction is skipped.
If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and a `NOP` is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example

```
HERE   BTFSS  FLAG,1
FALSE  GOTO   PROCESS_CODE
TRUE   •
       •
       •
```

Before Instruction
PC = address `HERE`
After Instruction
if FLAG<1> = 0,
PC = address `FALSE`
if FLAG<1> = 1,
PC = address `TRUE`

CALL — Call Subroutine

Syntax: [*label*] CALL k

Operands: $0 \le k \le 2047$

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

Status Affected: None

Encoding:

10	0kkk	kkkk	kkkk

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

Words: 1

Cycles: 2

Example

```
HERE   CALL   THERE
```

Before Instruction
PC = Address `HERE`
After Instruction
PC = Address `THERE`
TOS = Address `HERE+1`

CLRF — Clear f

Syntax: [*label*] CLRF f

Operands: $0 \le f \le 127$

Operation: 00h → (f)
1 → Z

Status Affected: Z

Encoding:

00	0001	1fff	ffff

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

Words: 1

Cycles: 1

Example

```
CLRF   FLAG_REG
```

Before Instruction
FLAG_REG = 0x5A
After Instruction
FLAG_REG = 0x00
Z= 1

CLRW — Clear W

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W)
1 → Z

Status Affected: Z

Encoding:

00	0001	0000	0011

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

Words: 1

Cycles: 1

Example

```
CLRW
```

Before Instruction
W = 0x5A
After Instruction
W = 0x00
Z= 1

CLRWDT — Clear Watchdog Timer

Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{\text{TO}}$ 1 → $\overline{\text{PD}}$
Status Affected:	$\overline{\text{TO}}$, $\overline{\text{PD}}$
Encoding:	`00` `0000` `0110` `0100`
Description:	`CLRWDT` instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{\text{TO}}$ and $\overline{\text{PD}}$ are set.
Words:	1
Cycles:	1
Example	`CLRWDT`

Before Instruction
WDT counter = ?
After Instruction
WDT counter = 0x00
WDT prescaler= 0
$\overline{\text{TO}}$ = 1
$\overline{\text{PD}}$ = 1

COMF — Complement f

Syntax:	[*label*] COMF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	($\bar{f}$) → (dest)
Status Affected:	Z
Encoding:	`00` `1001` `dfff` `ffff`
Description:	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example	`COMF REG1,0`

Before Instruction
REG1 = 0x13
After Instruction
REG1 = 0x13
W = 0xEC

DECF — Decrement f

Syntax:	[*label*] DECF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) - 1 → (dest)
Status Affected:	Z
Encoding:	`00` `0011` `dfff` `ffff`
Description:	Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example	`DECF CNT, 1`

Before Instruction
CNT = 0x01
Z= 0
After Instruction
CNT = 0x00
Z= 1

DECFSZ — Decrement f, Skip if 0

Syntax:	[*label*] DECFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) - 1 → (dest); skip if result = 0
Status Affected:	None
Encoding:	`00` `1011` `dfff` `ffff`
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 0, the next instruction, which is already fetched, is discarded. A `NOP` is executed instead making it a two cycle instruction.
Words:	1
Cycles:	1(2)
Example	

```
HERE      DECFSZ  CNT, 1
          GOTO    LOOP
CONTINUE  •
          •
          •
```

Before Instruction
PC = address `HERE`
After Instruction
CNT = CNT - 1
if CNT = 0,
PC = address `CONTINUE`
if CNT ≠ 0,
PC = address `HERE+1`

GOTO — Unconditional Branch

Syntax:	[*label*] GOTO k
Operands:	0 ≤ k ≤ 2047
Operation:	k → PC<10:0> PCLATH<4:3> → PC<12:11>
Status Affected:	None
Encoding:	`10` | `1kkk` | `kkkk` | `kkkk`
Description:	`GOTO` is an unconditional branch. The eleven bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. `GOTO` is a two cycle instruction.
Words:	1
Cycles:	2
Example	`GOTO THERE`

After Instruction
PC = Address `THERE`

INCFSZ — Increment f, Skip if 0

Syntax:	[*label*] INCFSZ f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (dest), skip if result = 0
Status Affected:	None
Encoding:	`00` | `1111` | `dfff` | `ffff`
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 0, the next instruction, which is already fetched, is discarded. A `NOP` is executed instead making it a two cycle instruction.
Words:	1
Cycles:	1(2)
Example	

```
HERE      INCFSZ    CNT, 1
          GOTO      LOOP
CONTINUE  •
          •
          •
```

Before Instruction
PC = address `HERE`
After Instruction
CNT = CNT + 1
if CNT= 0,
PC = address `CONTINUE`
if CNT≠ 0,
PC = address `HERE +1`

INCF — Increment f

Syntax:	[*label*] INCF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (dest)
Status Affected:	Z
Encoding:	`00` | `1010` | `dfff` | `ffff`
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'.
Words:	1
Cycles:	1
Example	`INCF CNT, 1`

Before Instruction
CNT = 0xFF
Z= 0
After Instruction
CNT = 0x00
Z= 1

IORLW — Inclusive OR Literal with W

Syntax:	[*label*] IORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Encoding:	`11` | `1000` | `kkkk` | `kkkk`
Description:	The contents of the W register are OR'ed with the eight bit literal 'k'. The result is placed in the W register.
Words:	1
Cycles:	1
Example	`IORLW 0x35`

Before Instruction
W = 0x9A
After Instruction
W= 0xBF
Z= 1

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) $\rightarrow$ (dest)

Status Affected: Z

Encoding:

`00`	`0100`	`dfff`	`ffff`

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'.

Words: 1

Cycles: 1

Example `IORWF` RESULT, 0

Before Instruction
RESULT = 0x13
W = 0x91
After Instruction
RESULT = 0x13
W = 0x93
Z= 1

MOVLW **Move Literal to W**

Syntax: [*label*] MOVLW k

Operands: $0 \le k \le 255$

Operation: k $\rightarrow$ (W)

Status Affected: None

Encoding:

`11`	`00xx`	`kkkk`	`kkkk`

Description: The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

Words: 1

Cycles: 1

Example `MOVLW 0x5A`

After Instruction
W = 0x5A

MOVF **Move f**

Syntax: [*label*] MOVF f,d

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

Operation: (f) $\rightarrow$ (dest)

Status Affected: Z

Encoding:

`00`	`1000`	`dfff`	`ffff`

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

Words: 1

Cycles: 1

Example `MOVF FSR,` 0

After Instruction
W = value in FSR register
Z= 1

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 127$

Operation: (W) $\rightarrow$ (f)

Status Affected: None

Encoding:

`00`	`0000`	`1fff`	`ffff`

Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Example `MOVWF OPTION`

Before Instruction
OPTION = 0xFF
W = 0x4F
After Instruction
OPTION = 0x4F
W = 0x4F

NOP — No Operation

Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Encoding:	`00` `0000` `0xx0` `0000`
Description:	No operation.
Words:	1
Cycles:	1
Example	`NOP`

OPTION — Load Option Register

Syntax:	[*label*] OPTION
Operands:	None
Operation:	(W) → OPTION
Status Affected:	None
Encoding:	`00` `0000` `0110` `0010`
Description:	The contents of the W register are loaded in the OPTION register. This instruction is supported for code compatibility with PIC16C5X products. Since OPTION is a readable/writable register, the user can directly address it.
Words:	1
Cycles:	1
Example	

To maintain upward compatibility with future PIC12C67X products, do not use this instruction.

RETFIE — Return from Interrupt

Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None
Encoding:	`00` `0000` `0000` `1001`
Description:	Return from Interrupt. Stack is POPed and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two cycle instruction.
Words:	1
Cycles:	2
Example	`RETFIE` After Interrupt PC = TOS GIE = 1

RETLW — Return with Literal in W

Syntax:	[*label*] RETLW k
Operands:	$0 \leq k \leq 255$
Operation:	k → (W); TOS → PC
Status Affected:	None
Encoding:	`11` `01xx` `kkkk` `kkkk`
Description:	The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two cycle instruction.
Words:	1
Cycles:	2
Example	`CALL TABLE;W contains table` ;offset value • ;W now has table value • •
`TABLE`	ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; • • • RETLW kn ; End of table Before Instruction W = 0x07 After Instruction W = value of k8

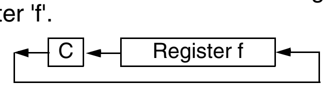
RETURN — Return from Subroutine

Syntax:	[*label*] RETURN
Operands:	None
Operation:	TOS → PC
Status Affected:	None
Encoding:	`00` `0000` `0000` `1000`
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two cycle instruction.
Words:	1
Cycles:	2
Example	`RETURN`

After Interrupt
PC = TOS

RLF — Rotate Left f through Carry

Syntax:	[*label*] RLF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C
Encoding:	`00` `1101` `dfff` `ffff`
Description:	The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.

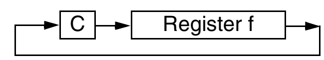


Words:	1
Cycles:	1
Example	`RLF REG1,0`

Before Instruction
REG1 = `1110 0110`
C = `0`
After Instruction
REG1 = `1110 0110`
W = `1100 1100`
C = `1`

RRF — Rotate Right f through Carry

Syntax:	[*label*] RRF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C
Encoding:	`00` `1100` `dfff` `ffff`
Description:	The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.



Words:	1
Cycles:	1
Example	`RRF` REG1, 0

Before Instruction
REG1 = `1110 0110`
C = `0`
After Instruction
REG1 = `1110 0110`
W = `0111 0011`
C = `0`

SLEEP

Syntax:	[*label*] SLEEP
Operands:	None
Operation:	00h → WDT, 0 → WDT prescaler, 1 → $\overline{TO}$, 0 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Encoding:	`00` `0000` `0110` `0011`
Description:	The power-down status bit, $\overline{PD}$ is cleared. Time-out status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared. The processor is put into SLEEP mode with the oscillator stopped.
Words:	1
Cycles:	1
Example:	`SLEEP`

SUBLW Subtract W from Literal

Syntax: [*label*] SUBLW k

Operands: $0 \leq k \leq 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Encoding:

`11`	`110x`	`kkkk`	`kkkk`

Description: The W register is subtracted (2’s complement method) from the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example 1: `SUBLW 0x02`

Before Instruction

W= 1
C= ?

After Instruction

W= 1
C = 1; result is positive

Example 2: Before Instruction

W= 2
C= ?

After Instruction

W= 0
C = 1; result is zero

Example 3: Before Instruction

W= 3
C= ?

After Instruction

W= 0xFF
C = 0; result is negative

SUBWF Subtract W from f

Syntax: [*label*] SUBWF f,d

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

Operation: (f) - (W) → (dest)

Status Affected: C, DC, Z

Encoding:

`00`	`0010`	`dfff`	`ffff`

Description: Subtract (2’s complement method) W register from register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example 1: `SUBWF REG1,1`

Before Instruction

REG1 = 3
W = 2
C = ?

After Instruction

REG1 = 1
W = 2
C = 1; result is positive

Example 2: Before Instruction

REG1 = 2
W= 2
C= ?

After Instruction

REG1 = 0
W= 2
C = 1; result is zero

Example 3: Before Instruction

REG1 = 1
W= 2
C= ?

After Instruction

REG1 = 0xFF
W= 2
C = 0; result is negative

SWAPF — Swap Nibbles in f

Syntax:	[*label*] SWAPF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f<3:0>) → (dest<7:4>), (f<7:4>) → (dest<3:0>)
Status Affected:	None
Encoding:	`00` `1110` `dfff` `ffff`
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in W register. If 'd' is 1, the result is placed in register 'f'.
Words:	1
Cycles:	1
Example	`SWAPF` REG, 0

Before Instruction

REG1 = 0xA5

After Instruction

REG1 = 0xA5
W = 0x5A

TRIS — Load TRIS Register

Syntax:	[*label*] TRIS f
Operands:	$5 \le f \le 7$
Operation:	(W) → TRIS register f;
Status Affected:	None
Encoding:	`00` `0000` `0110` `0fff`
Description:	The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them.
Words:	1
Cycles:	1
Example	

To maintain upward compatibility with future PIC12C67X products, do not use this instruction.

XORLW — Exclusive OR Literal with W

Syntax:	[*label*] XORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .XOR. k → (W)
Status Affected:	Z
Encoding:	`11` `1010` `kkkk` `kkkk`
Description:	The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.
Words:	1
Cycles:	1
Example:	`XORLW 0xAF`

Before Instruction

W= 0xB5

After Instruction

W = 0x1A

XORWF — Exclusive OR W with f

Syntax:	[*label*] XORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .XOR. (f) → (dest)
Status Affected:	Z
Encoding:	`00` `0110` `dfff` `ffff`
Description:	Exclusive OR the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example	`XORWF REG` 1

Before Instruction

REG = 0xAF
W= 0xB5

After Instruction

REG = 0x1A
W= 0xB5

NOTES:

11.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM Assembler
  - MPLAB-C17 and MPLAB-C18 C Compilers
  - MPLINK/MPLIB Linker/Librarian
- Simulators
  - MPLAB-SIM Software Simulator
- Emulators
  - MPLAB-ICE Real-Time In-Circuit Emulator
  - PICMASTER®/PICMASTER-CE In-Circuit Emulator
  - ICEPIC™
- In-Circuit Debugger
  - MPLAB-ICD for PIC16F877
- Device Programmers
  - PRO MATE® II Universal Programmer
  - PICSTART® Plus Entry-Level Prototype Programmer
- Low-Cost Demonstration Boards
  - SIMICE
  - PICDEM-1
  - PICDEM-2
  - PICDEM-3
  - PICDEM-17
  - SEEVAL®
  - KEELOQ®

11.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8-bit microcontroller market. MPLAB is a Windows®-based application which contains:

- Multiple functionality
  - editor
  - simulator
  - programmer (sold separately)
  - emulator (sold separately)
- A full featured editor
- A project manager
- Customizable tool bar and key mapping
- A status bar
- On-line help

MPLAB allows you to:

- Edit your source files (either assembly or 'C')
- One touch assemble (or compile) and download to PIC MCU tools (automatically updates all project information)
- Debug using:
  - source files
  - absolute listing file
  - object code

The ability to use MPLAB with Microchip's simulator, MPLAB-SIM, allows a consistent platform and the ability to easily switch from the cost-effective simulator to the full featured emulator with minimal retraining.

11.2 MPASM Assembler

MPASM is a full featured universal macro assembler for all PIC MCUs. It can produce absolute code directly in the form of HEX files for device programmers, or it can generate relocatable objects for MPLINK.

MPASM has a command line interface and a Windows shell and can be used as a standalone application on a Windows 3.x or greater system. MPASM generates relocatable object files, Intel standard HEX files, MAP files to detail memory usage and symbol reference, an absolute LST file which contains source lines and generated machine code, and a COD file for MPLAB debugging.

MPASM features include:

- MPASM and MPLINK are integrated into MPLAB projects.
- MPASM allows user defined macros to be created for streamlined assembly.
- MPASM allows conditional assembly for multi purpose source files.
- MPASM directives allow complete control over the assembly process.

11.3 MPLAB-C17 and MPLAB-C18 C Compilers

The MPLAB-C17 and MPLAB-C18 Code Development Systems are complete ANSI 'C' compilers and integrated development environments for Microchip's PIC17CXXX and PIC18CXXX family of microcontrollers, respectively. These compilers provide powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compilers provide symbol information that is compatible with the MPLAB IDE memory display.

11.4 MPLINK/MPLIB Linker/Librarian

MPLINK is a relocatable linker for MPASM and MPLAB-C17 and MPLAB-C18. It can link relocatable objects from assembly or C source files along with precompiled libraries using directives from a linker script.

MPLIB is a librarian for pre-compiled code to be used with MPLINK. When a routine from a library is called from another source file, only the modules that contains that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications. MPLIB manages the creation and modification of library files.

MPLINK features include:

- MPLINK works with MPASM and MPLAB-C17 and MPLAB-C18.
- MPLINK allows all memory areas to be defined as sections to provide link-time flexibility.

MPLIB features include:

- MPLIB makes linking easier because single libraries can be included instead of many smaller files.
- MPLIB helps keep code maintainable by grouping related modules together.
- MPLIB commands allow libraries to be created and modules to be added, listed, replaced, deleted, or extracted.

11.5 MPLAB-SIM Software Simulator

The MPLAB-SIM Software Simulator allows code development in a PC host environment by simulating the PIC series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file or user-defined key press to any of the pins. The execution can be performed in single step, execute until break, or trace mode.

MPLAB-SIM fully supports symbolic debugging using MPLAB-C17 and MPLAB-C18 and MPASM. The Software Simulator offers the flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

11.6 MPLAB-ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

The MPLAB-ICE Universal In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers (MCUs). Software control of MPLAB-ICE is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, “make” and download, and source debugging from a single environment.

Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB-ICE allows expansion to support new PIC microcontrollers.

The MPLAB-ICE Emulator System has been designed as a real-time emulation system with advanced features that are generally found on more expensive development tools. The PC platform and Microsoft® Windows 3.x/95/98 environment were chosen to best make these features available to you, the end user.

MPLAB-ICE 2000 is a full-featured emulator system with enhanced trace, trigger, and data monitoring features. Both systems use the same processor modules and will operate across the full operating speed range of the PIC MCU.

11.7 PICMASTER/PICMASTER CE

The PICMASTER system from Microchip Technology is a full-featured, professional quality emulator system. This flexible in-circuit emulator provides a high-quality, universal platform for emulating Microchip 8-bit PIC microcontrollers (MCUs). PICMASTER systems are sold worldwide, with a CE compliant model available for European Union (EU) countries.

11.8 ICEPIC

ICEPIC is a low-cost in-circuit emulation solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X, and PIC16CXXX families of 8-bit one-time-programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules or daughter boards. The emulator is capable of emulating without target application circuitry being present.

11.9 MPLAB-ICD In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB-ICD, is a powerful, low-cost run-time development tool. This tool is based on the flash PIC16F877 and can be used to develop for this and other PIC microcontrollers from the PIC16CXXX family. MPLAB-ICD utilizes the In-Circuit Debugging capability built into the PIC16F87X. This feature, along with Microchip's In-Circuit Serial Programming protocol, offers cost-effective in-circuit flash programming and debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by watching variables, single-stepping and setting break points. Running at full speed enables testing hardware in real-time. The MPLAB-ICD is also a programmer for the flash PIC16F87X family.

11.10 PRO MATE II Universal Programmer

The PRO MATE II Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode. PRO MATE II is CE compliant.

The PRO MATE II has programmable VDD and VPP supplies which allows it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for instructions and error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode the PRO MATE II can read, verify or program PIC devices. It can also set code-protect bits in this mode.

11.11 PICSTART Plus Entry Level Development System

The PICSTART programmer is an easy-to-use, low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. MPLAB Integrated Development Environment software makes using the programmer simple and efficient.

PICSTART Plus supports all PIC devices with up to 40 pins. Larger pin count devices such as the PIC16C92X, and PIC17C76X may be supported with an adapter socket. PICSTART Plus is CE compliant.

11.12 SIMICE Entry-Level Hardware Simulator

SIMICE is an entry-level hardware development system designed to operate in a PC-based environment with Microchip's simulator MPLAB-SIM. Both SIMICE and MPLAB-SIM run under Microchip Technology's MPLAB Integrated Development Environment (IDE) software. Specifically, SIMICE provides hardware simulation for Microchip's PIC12C5XX, PIC12CE5XX, and PIC16C5X families of PIC 8-bit microcontrollers. SIMICE works in conjunction with MPLAB-SIM to provide non-real-time I/O port emulation. SIMICE enables a developer to run simulator code for driving the target system. In addition, the target system can provide input to the simulator code. This capability allows for simple and interactive debugging without having to manually generate MPLAB-SIM stimulus files. SIMICE is a valuable debugging tool for entry-level system development.

11.13 PICDEM-1 Low-Cost PIC MCU Demonstration Board

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE II or PICSTART-Plus programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the MPLAB-ICE emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

11.14 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE II programmer or PICSTART-Plus, and easily test firmware. The MPLAB-ICE emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I^2C bus and separate headers for connection to an LCD module and a keypad.

11.15 PICDEM-3 Low-Cost PIC16CXXX Demonstration Board

The PICDEM-3 is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with a LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-3 board, on a PRO MATE II programmer or PICSTART Plus with an adapter socket, and easily test firmware. The MPLAB-ICE emulator may also be used with the PICDEM-3 board to test firmware. Additional prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include an RS-232 interface, push-button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM-3 board is an LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM-3 provides an additional RS-232 interface and Windows 3.1 software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals.

11.16 PICDEM-17

The PICDEM-17 is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756, PIC17C762, and PIC17C766. All necessary hardware is included to run basic demo programs, which are supplied on a 3.5-inch disk. A programmed sample is included, and the user may erase it and program it with the other sample programs using the PRO MATE II or PICSTART Plus device programmers and easily debug and test the sample code. In addition, PICDEM-17 supports down-loading of programs to and executing out of external FLASH memory on board. The PICDEM-17 is also usable with the MPLAB-ICE or PICMASTER emulator, and all of the sample programs can be run and modified using either emulator. Additionally, a generous prototype area is available for user hardware.

11.17 SEEVAL Evaluation and Programming System

The SEEVAL SEEPROM Designer's Kit supports all Microchip 2-wire and 3-wire Serial EEPROMs. The kit includes everything necessary to read, write, erase or program special features of any Microchip SEEPROM product including Smart Serials™ and secure serials. The Total Endurance™ Disk is included to aid in trade-off analysis and reliability calculations. The total kit can significantly reduce time-to-market and result in an optimized system.

11.18 KEELOQ Evaluation and Programming Tools

KEELOQ evaluation and programming tools support Microchips HCS Secure Data Products. The HCS evaluation kit includes an LCD display to show changing codes, a decoder to decode transmissions, and a programming interface to program test transmitters.

TABLE 11-1: DEVELOPMENT TOOLS FROM MICROCHIP

		PIC12CXXX	PIC14000	PIC16C5X	PIC16C6X	PIC16CXXX	PIC16F62X	PIC16C7X	PIC16C7XX	PIC16C8X	PIC16F8XX	PIC16C9XX	PIC17C4X	PIC17C7XX	PIC18CXX2	24CXX/ 25CXX/ 93CXX	HCSXXX	MCRFXXX	MCP2510
Software Tools	MPLAB® Integrated Development Environment	✓	✓✓✓✓✓✓✓✓✓✓✓✓✓																
	MPLAB® C17 Compiler												✓✓						
	MPLAB® C18 Compiler														✓				
	MPASM/MPLINK	✓	✓✓✓✓✓✓✓✓✓✓✓✓✓✓														✓		
Emulators	MPLAB®-ICE	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓				
	PICMASTER/PICMASTER-CE	✓	✓✓✓✓					✓✓✓				✓✓✓							
	ICEPIC™ Low-Cost In-Circuit Emulator	✓		✓✓✓				✓✓✓				✓							
Debugger	MPLAB®-ICD In-Circuit Debugger				✓*			✓*			✓								
Programmers	PICSTART®Plus Low-Cost Universal Dev. Kit	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓				
	PRO MATE® II Universal Programmer	✓	✓✓✓✓				✓**	✓✓✓✓✓✓✓✓									✓		
Demo Boards and Eval Kits	SIMICE	✓		✓															
	PICDEM-1			✓		✓		✓†		✓			✓						
	PICDEM-2				✓†			✓†							✓				
	PICDEM-3											✓							
	PICDEM-14A		✓																
	PICDEM-17													✓					
	KEELOQ® Evaluation Kit																✓		
	KEELOQ Transponder Kit																✓		
	microID™ Programmer's Kit																	✓	
	125 kHz microID Developer's Kit																	✓	
	125 kHz Anticollision microID Developer's Kit																	✓	
	13.56 MHz Anticollision microID Developer's Kit																	✓	
	MCP2510 CAN Developer's Kit																		✓

* Contact the Microchip Technology Inc. web site at www.microchip.com for information on how to use the MPLAB®-ICD In-Circuit Debugger (DV164001) with PIC16C62, 63, 64, 65, 72, 73, 74, 76, 77

** Contact Microchip Technology Inc. for availability date.

† Development tool is available on select devices.

NOTES:

12.0 ELECTRICAL SPECIFICATIONS FOR PIC12C67X

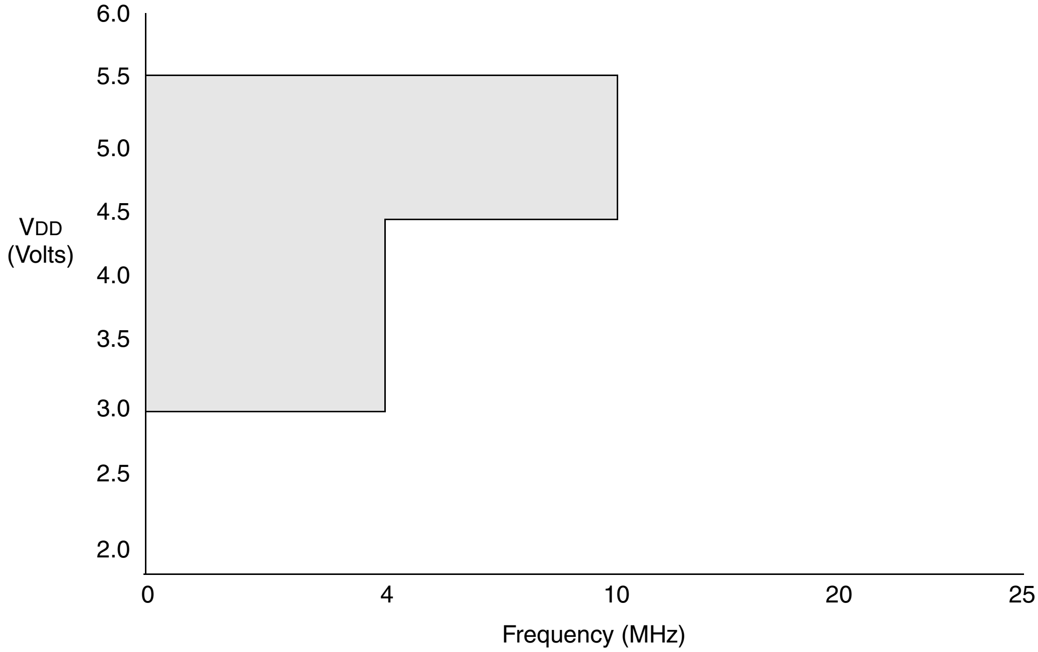
Absolute Maximum Ratings †

Ambient temperature under bias..........–40° to +125°C
Storage temperature .... –65°C to +150°C
Voltage on any pin with respect to VSS (except VDD and $\overline{MCLR}$).......... –0.3V to (VDD + 0.3V)
Voltage on VDD with respect to VSS .... 0 to +7.0V
Voltage on $\overline{MCLR}$ with respect to VSS (Note 2)..........0 to +14V
Total power dissipation (Note 1)..........700 mW
Maximum current out of VSS pin..........200 mA
Maximum current into VDD pin..........150 mA
Input clamp current, I_{IK} (V_I < 0 or V_I > VDD)..........± 20 mA
Output clamp current, I_{OK} (V_O < 0 or V_O > VDD)..........± 20 mA
Maximum output current sunk by any I/O pin..........25 mA
Maximum output current sourced by any I/O pin..........25 mA
Maximum current sunk by GPIO pins combined..........100 mA
Maximum current sourced by GPIO pins combined..........100 mA

Note 1: Power dissipation is calculated as follows: Pdis = $V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum(V_{OL} \times I_{OL})$.

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

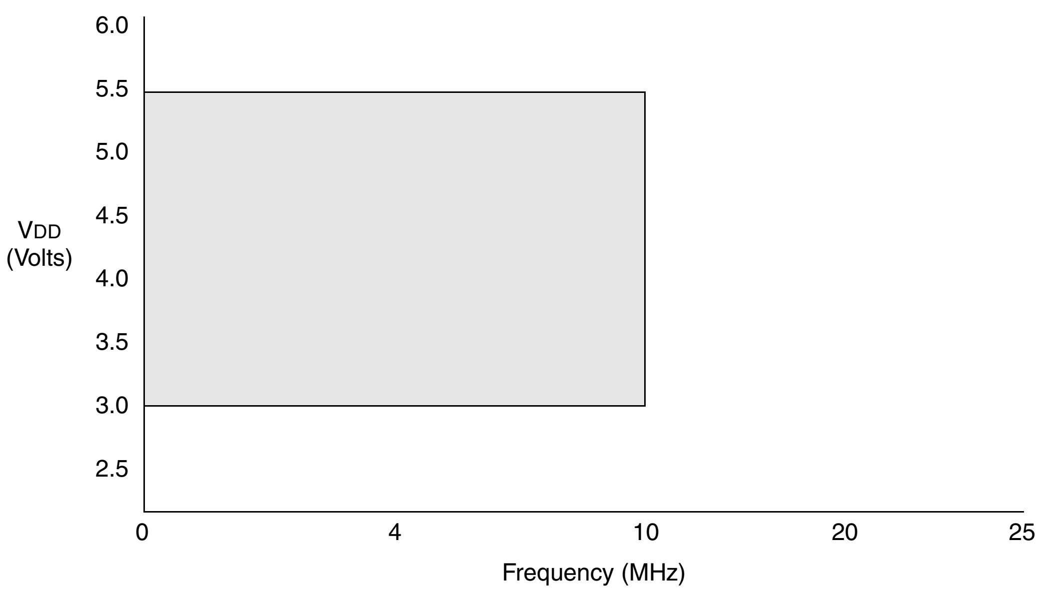
FIGURE 12-1: PIC12C67X VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA < 0°C, +70°C <TA ≤ +125°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

2: The maximum rated speed of the part limits the permissible combinations of voltage and frequency. Please reference the Product Identification System section for the maximum rated speed of the parts.

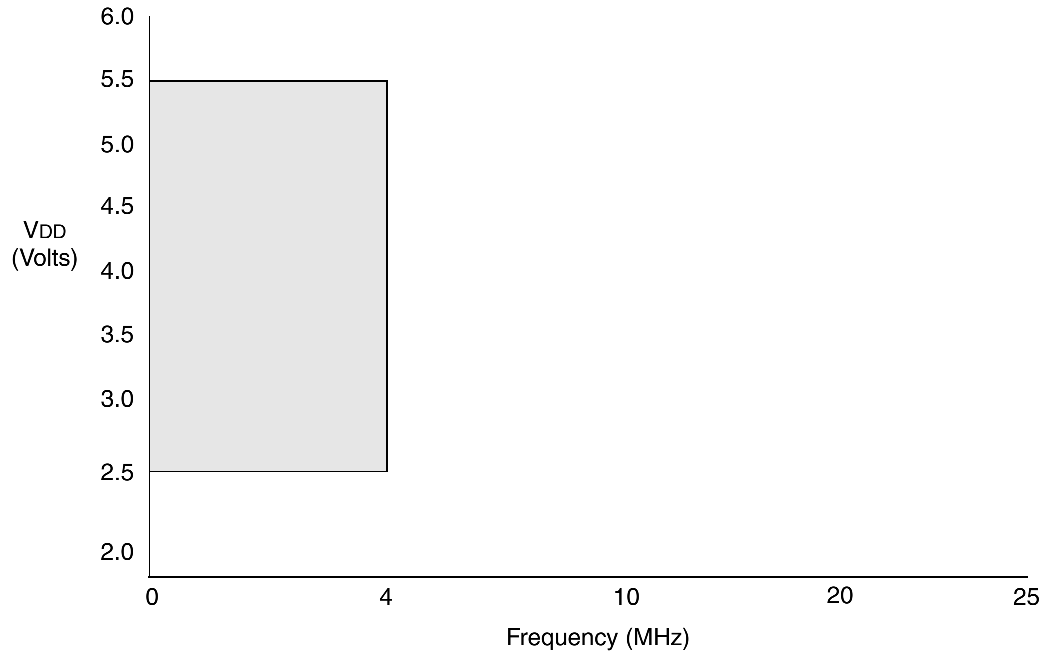
FIGURE 12-2: PIC12C67X VOLTAGE-FREQUENCY GRAPH, 0°C ≤ TA ≤ +70°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

2: The maximum rated speed of the part limits the permissible combinations of voltage and frequency. Please reference the Product Identification System section for the maximum rated speed of the parts.

FIGURE 12-3: PIC12LC67X VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ +85°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

2: The maximum rated speed of the part limits the permissible combinations of voltage and frequency. Please reference the Product Identification System section for the maximum rated speed of the parts.

12.1 DC Characteristics: PIC12C671/672 (Commercial, Industrial, Extended) PIC12CE673/674 (Commercial, Industrial, Extended)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise specified) Operating Temperature 0°C ≤ TA ≤ +70°C (commercial) –40°C ≤ TA ≤ +85°C (industrial) –40°C ≤ TA ≤ +125°C (extended)					
Parm No.	**Characteristic**	**Sym**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
D001	**Supply Voltage**	VDD	3.0		5.5	V	
D002	**RAM Data Retention Voltage(2)**	VDR		1.5*		V	Device in SLEEP mode
D003	**VDD Start Voltage to ensure Power-on Reset**	VPOR		VSS		V	See section on Power-on Reset for details
D004	**VDD Rise Rate to ensure Power-on Reset**	SVDD	0.05*			V/ms	See section on Power-on Reset for details
D010	**Supply Current(3)**	IDD	—	1.2	2.5	mA	FOSC = 4MHz, VDD = 3.0V XT and EXTRC mode (Note 4)
D010C			—	1.2	2.5	mA	FOSC = 4MHz, VDD = 3.0V INTRC mode (Note 6)
			—	2.2	8	mA	FOSC = 10MHz, VDD = 5.5V HS mode
D010A			—	19	29	µA	FOSC = 32kHz, VDD = 3.0V, WDT disabled LP mode, Commercial Temperature
			—	19	37	µA	FOSC = 32kHz, VDD = 3.0V, WDT disabled LP mode, Industrial Temperature
			—	32	60	µA	FOSC = 32kHz, VDD = 3.0V, WDT disabled LP mode, Extended Temperature
D020	**Power-down Current(5)**	IPD	—	0.25	6	µA	VDD = 3.0V, Commercial, WDT disabled
D021			—	0.25	7	µA	VDD = 3.0V, Industrial, WDT disabled
D021B			—	2	14	µA	VDD = 3.0V, Extended, WDT disabled
			—	0.5	8	µA	VDD = 5.5V, Commercial, WDT disabled
			—	0.8	9	µA	VDD = 5.5V, Industrial, WDT disabled
			—	3	16	µA	VDD = 5.5V, Extended, WDT disabled
D022	**Watchdog Timer Current**	ΔIWDT	—	2.2	5	µA	VDD = 3.0V, Commercial
			—	2.2	6	µA	VDD = 3.0V, Industrial
			—	4	11	µA	VDD = 3.0V, Extended
D028	**Supply Current(3)** During read/write to EEPROM peripheral	ΔIEE	—0	.1	0.2	mA	FOSC = 4MHz, VDD = 5.5V, SCL = 400kHz For PIC12CE673/674 only

* These parameters are characterized but not tested.

Note 1: Data in Typical ("Typ") column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

2: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

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

a) The test conditions for all IDD measurements in active operation mode are:
$\overline{\text{OSC1}}$ = external square wave, from rail-to-rail; all I/O pins tristated, pulled to VSS, T0CKI = VDD,
$\overline{\text{MCLR}}$ = VDD; WDT disabled.

b) For standby current measurements, the conditions are the same, except that the device is in SLEEP mode.

4: For EXTRC osc configuration, current through REXT is not included. The current through the resistor can be estimated by the formula:
Ir = VDD/2REXT (mA) with REXT in kOhm.

5: 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 hi-impedance state and tied to VDD or VSS.

6: INTRC calibration value is for 4MHz nominal at 5V, 25°C.

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise specified) Operating Temperature 0°C ≤ TA ≤ +70°C (commercial) –40°C ≤ TA ≤ +85°C (industrial) –40°C ≤ TA ≤ +125°C (extended)					
Parm No.	**Characteristic**	**Sym**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
	LP Oscillator Operating Frequency	FOSC	0		200	kHz	All temperatures
	INTRC/EXTRC Oscillator Operating Frequency		—		4(6)	MHz	All temperatures
	XT Oscillator Operating Frequency		0		4	MHz	All temperatures
	HS Oscillator Operating Frequency		0		10	MHz	All temperatures

* These parameters are characterized but not tested.

Note 1: Data in Typical ("Typ") column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

2: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

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

a) The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tristated, pulled to VSS, T0CKI = VDD, $\overline{\text{MCLR}}$ = VDD; WDT disabled.

b) For standby current measurements, the conditions are the same, except that the device is in SLEEP mode.

4: For EXTRC osc configuration, current through REXT is not included. The current through the resistor can be estimated by the formula:
Ir = VDD/2REXT (mA) with REXT in kOhm.

5: 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 hi-impedance state and tied to VDD or VSS.

6: INTRC calibration value is for 4MHz nominal at 5V, 25°C.

12.2 DC Characteristics: PIC12LC671/672 (Commercial, Industrial) PIC12LCE673/674 (Commercial, Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating temperature 0°C ≤ TA ≤ +70°C (commercial) –40°C ≤ TA ≤ +85°C (industrial)				
Param No.	**Characteristic**	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
D001	**Supply Voltage**	VDD	2.5		5.5	V	
D002	**RAM Data Retention Voltage(2)**	VDR		1.5*		V	Device in SLEEP mode
D003	**VDD Start Voltage to ensure Power-on Reset**	VPOR		VSS		V	See section on Power-on Reset for details
D004	**VDD Rise Rate to ensure Power-on Reset**	SVDD	0.05*			V/ms	See section on Power-on Reset for details
D010	**Supply Current(3)**	IDD	—	0.4	2.1	mA	FOSC = 4MHz, VDD = 2.5V XT and EXTRC mode (Note 4)
D010C			—	0.4	2.1	mA	FOSC = 4MHz, VDD = 2.5V INTRC mode (Note 6)
D010A			—	15	33	µA	FOSC = 32kHz, VDD = 2.5V, WDT disabled LP mode, Industrial Temperature
D020	**Power-down Current(5)**	IPD					
D021			—	0.2	5	µA	VDD = 2.5V, Commercial
D021B			—	0.2	6	µA	VDD = 2.5V, Industrial
	Watchdog Timer Current	ΔIWDT	—2	.0	4	µA	VDD = 2.5V, Commercial
				2.0	6	µA	VDD = 2.5V, Industrial
	LP Oscillator Operating Frequency	FOSC	0		200	kHz	All temperatures
	INTRC/EXTRC Oscillator Operating Frequency		—		4(6)	MHz	All temperatures
	XT Oscillator Operating Frequency		0		4	MHz	All temperatures
	HS Oscillator Operating Frequency		0		10	MHz	All temperatures

* These parameters are characterized but not tested.

Note 1: Data in Typical ("Typ") column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

2: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

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

a) The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tristated, pulled to VSS, T0CKI = VDD,
$\overline{MCLR}$ = VDD; WDT disabled.

b) For standby current measurements, the conditions are the same, except that the device is in SLEEP mode.

4: For EXTRC osc configuration, current through REXT is not included. The current through the resistor can be estimated by the formula:
Ir = VDD/2REXT (mA) with REXT in kOhm.

5: 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 hi-impedance state and tied to VDD or VSS.

6: INTRC calibration value is for 4MHz nominal at 5V, 25°C.

12.3 DC CHARACTERISTICS: PIC12C671/672 (Commercial, Industrial, Extended) PIC12CE673/674 (Commercial, Industrial, Extended)

DC CHARACTERISTICS							**Standard Operating Conditions (unless otherwise specified)** Operating temperature 0°C $\leq T_A \leq$ +70°C (commercial) –40°C $\leq T_A \leq$ +85°C (industrial) –40°C $\leq T_A \leq$ +125°C (extended) Operating voltage V_{DD} range as described in DC spec Section 12.1 and Section 12.2.
Param No.	**Characteristic**	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	Input Low Voltage						
	I/O ports	V_{IL}					
D030	with TTL buffer		V_{SS}	—	0.8V	V	For 4.5V $\leq V_{DD} \leq$ 5.5V
			V_{SS}	—	$0.15V_{DD}$	V	otherwise
D031	with Schmitt Trigger buffer		V_{SS}	—	$0.2V_{DD}$	V	
D032	$\overline{MCLR}$, GP2/T0CKI/AN2/INT (in EXTRC mode)		V_{SS}	—	$0.2V_{DD}$	V	
D033	OSC1 (in EXTRC mode)		V_{SS}	—	$0.2V_{DD}$		Note 1
D033	OSC1 (in XT, HS, and LP)		V_{SS}	—	$0.3V_{DD}$	V	Note 1
	Input High Voltage						
	I/O ports	V_{IH}		—			
D040	with TTL buffer		2.0V	—	V_{DD}	V4	.5V $\leq V_{DD} \leq$ 5.5V
D040A			$0.25V_{DD}$ + 0.8V	—	V_{DD}	V	otherwise
D041	with Schmitt Trigger buffer		$0.8V_{DD}$	—	V_{DD}	VF	or entire V_{DD} range
D042	$\overline{MCLR}$, GP2/T0CKI/AN2/INT		$0.8V_{DD}$	—	V_{DD}	V	
D042A	OSC1 (XT, HS, and LP)		$0.7V_{DD}$	—	V_{DD}	V	Note 1
D043	OSC1 (in EXTRC mode)		$0.9V_{DD}$	—	V_{DD}	V	
	Input Leakage Current (Notes 2, 3)						
D060	I/O ports	I_{IL}	—	—	±1	μA	$V_{SS} \leq V_{PIN} \leq V_{DD}$, Pin at hi-impedance
D061	GP3/$\overline{MCLR}$ (Note 5)				±30	μA	$V_{SS} \leq V_{PIN} \leq V_{DD}$
D061A	GP3 (Note 6)				±5	μA	$V_{SS} \leq V_{PIN} \leq V_{DD}$
D062	GP2/T0CKI		—	—	±5	μA	$V_{SS} \leq V_{PIN} \leq V_{DD}$
D063	OSC1		—	—	±5	μA	$V_{SS} \leq V_{PIN} \leq V_{DD}$, XT, HS, and LP osc configuration
D070	GPIO weak pull-up current (Note 4)	I_{PUR}	50	250	400	μA	V_{DD} = 5V, $V_{PIN} = V_{SS}$
	$\overline{MCLR}$ pull-up current	—	—	—	30	μA	V_{DD} = 5V, $V_{PIN} = V_{SS}$
	Output Low Voltage						
D080	I/O ports	V_{OL}	—	—	0.6	V	I_{OL} = 8.5 mA, V_{DD} = 4.5V, –40°C to +85°C
D080A			—	—	0.6	V	I_{OL} = 7.0 mA, V_{DD} = 4.5V, –40°C to +125°C
D083	OSC2/CLKOUT		—	—	0.6	V	I_{OL} = 1.6 mA, V_{DD} = 4.5V, –40°C to +85°C
D083A			—	—	0.6	V	I_{OL} = 1.2 mA, V_{DD} = 4.5V, –40°C to +125°C

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In EXTRC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC12C67X be driven with external clock in RC mode.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

4: Does not include GP3. For GP3 see parameters D061 and D061A.

5: This spec. applies to GP3/$\overline{MCLR}$ configured as external $\overline{MCLR}$ and GP3/$\overline{MCLR}$ configured as input with internal pull-up enabled.

6: This spec. applies when GP3/$\overline{MCLR}$ is configured as an input with pull-up disabled. The leakage current of the $\overline{MCLR}$ circuit is higher than the standard I/O logic.

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise specified) Operating temperature 0°C ≤ T_A ≤ +70°C (commercial) –40°C ≤ T_A ≤ +85°C (industrial) –40°C ≤ T_A ≤ +125°C (extended) Operating voltage V_{DD} range as described in DC spec Section 12.1 and Section 12.2.					
Param No.	**Characteristic**	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	Output High Voltage						
D090	I/O ports (Note 3)	V_{OH}	V_{DD} - 0.7	—	—	V	I_{OH} = -3.0 mA, V_{DD} = 4.5V, –40°C to +85°C
D090A			V_{DD} - 0.7	——		VI	OH = -2.5 mA, V_{DD} = 4.5V, –40°C to +125°C
D092	OSC2/CLKOUT		V_{DD} - 0.7	—	—	V	I_{OH} = 1.3 mA, V_{DD} = 4.5V, –40°C to +85°C
D092A			V_{DD} - 0.7	—	—	V	I_{OH} = 1.0 mA, V_{DD} = 4.5V, –40°C to +125°C
	Capacitive Loading Specs on Output Pins						
D100	OSC2 pin	C_{OSC2}	——		15	pF	In XT and LP modes when external clock is used to drive OSC1.
D101	All I/O pins	C_{IO}	——		50	pF	

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In EXTRC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC12C67X be driven with external clock in RC mode.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

4: Does not include GP3. For GP3 see parameters D061 and D061A.

5: This spec. applies to GP3/$\overline{MCLR}$ configured as external $\overline{MCLR}$ and GP3/$\overline{MCLR}$ configured as input with internal pull-up enabled.

6: This spec. applies when GP3/$\overline{MCLR}$ is configured as an input with pull-up disabled. The leakage current of the $\overline{MCLR}$ circuit is higher than the standard I/O logic.

12.4 DC CHARACTERISTICS: PIC12LC671/672 (Commercial, Industrial) PIC12LCE673/674 (Commercial, Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating temperature 0°C ≤ TA ≤ +70°C (commercial) –40°C ≤ TA ≤ +85°C (industrial) Operating voltage VDD range as described in DC spec Section 12.1 and Section 12.2.				
Param No.	**Characteristic**	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	Input Low Voltage						
	I/O ports	VIL					
D030	with TTL buffer		VSS	—	0.8V	V	For 4.5V ≤ VDD ≤ 5.5V
			VSS	—0	.15VDD	Vo	therwise
D031	with Schmitt Trigger buffer		VSS	—0	.2VDD	V	
D032	$\overline{\text{MCLR}}$, GP2/T0CKI/AN2/INT (in EXTRC mode)		VSS	—	0.2VDD	V	
D033	OSC1 (in EXTRC mode)		VSS	—	0.2VDD	V	Note 1
D033	OSC1 (in XT, HS, and LP)		VSS	—	0.3VDD	V	Note 1
	Input High Voltage						
	I/O ports	VIH		—			
D040	with TTL buffer		2.0V	—	VDD	V	4.5V ≤ VDD ≤ 5.5V
D040A			0.25VDD + 0.8V	—	VDD	V	otherwise
D041	with Schmitt Trigger buffer		0.8VDD	—	VDD	V	For entire VDD range
D042	$\overline{\text{MCLR}}$, GP2/T0CKI/AN2/INT		0.8VDD	—	VDD	V	
D042A	OSC1 (XT, HS, and LP)		0.7VDD	—	VDD	V	Note 1
D043	OSC1 (in EXTRC mode)		0.9VDD	—	VDD	V	
	Input Leakage Current (Notes 2, 3)						
D060	I/O ports	IIL	—	—	±1	μA	Vss ≤ VPIN ≤ VDD, Pin at hi-impedance
D061	GP3/$\overline{\text{MCLR}}$ (Note 5)				±30	μA	Vss ≤ VPIN ≤ VDD
D061A	GP3 (Note 6)				±5	μA	Vss ≤ VPIN ≤ VDD
D062	GP2/T0CKI		—	—	±5	μA	Vss ≤ VPIN ≤ VDD
D063	OSC1		—	—	±5	μA	Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
D070	GPIO weak pull-up current (Note 4)	IPUR	50	250	400	μA	VDD = 5V, VPIN = VSS
	$\overline{\text{MCLR}}$ pull-up current	—	—	—	30	μA	VDD = 5V, VPIN = VSS
	Output Low Voltage						
D080	I/O ports	VOL	—	—	0.6	V	IOL = 8.5 mA, VDD = 4.5V, –40°C to +85°C
D080A			—	—	0.6	V	IOL = 7.0 mA, VDD = 4.5V, –40°C to +125°C
D083	OSC2/CLKOUT		—	—	0.6	V	IOL = TBD, VDD = 4.5V, –40°C to +85°C
D083A			—	—	0.6	V	IOL = TBD, VDD = 4.5V, –40°C to +125°C

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In EXTRC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC12C67X be driven with external clock in RC mode.

2: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

4: Does not include GP3. For GP3 see parameters D061 and D061A.

5: This spec. applies to GP3/$\overline{\text{MCLR}}$ configured as external $\overline{\text{MCLR}}$ and GP3/$\overline{\text{MCLR}}$ configured as input with internal pull-up enabled.

6: This spec. applies when GP3/$\overline{\text{MCLR}}$ is configured as an input with pull-up disabled. The leakage current of the $\overline{\text{MCLR}}$ circuit is higher than the standard I/O logic.

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise specified) Operating temperature 0°C ≤ T_A ≤ +70°C (commercial) –40°C ≤ T_A ≤ +85°C (industrial) Operating voltage V_{DD} range as described in DC spec Section 12.1 and Section 12.2.					
Param No.	**Characteristic**	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	Output High Voltage						
D090	I/O ports (Note 3)	V_{OH}	V_{DD} - 0.7	—	—	V	I_{OH} = -3.0 mA, V_{DD} = 4.5V, –40°C to +85°C
D090A			V_{DD} - 0.7	—	—	V	I_{OH} = -2.5 mA, V_{DD} = 4.5V, –40°C to +125°C
D092	OSC2/CLKOUT		V_{DD} - 0.7	—	—	V	I_{OH} = TBD, V_{DD} = 4.5V, –40°C to +85°C
D092A			V_{DD} - 0.7	—	—	V	I_{OH} = TBD, V_{DD} = 4.5V, –40°C to +125°C
	Capacitive Loading Specs on Output Pins						
D100	OSC2 pin	C_{OSC2}	—	—	15	pF	In XT and LP modes when external clock is used to drive OSC1.
D101	All I/O pins	C_{IO}	—	—	50	pF	

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In EXTRC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC12C67X be driven with external clock in RC mode.

2: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

4: Does not include GP3. For GP3 see parameters D061 and D061A.

5: This spec. applies to GP3/$\overline{\text{MCLR}}$ configured as external $\overline{\text{MCLR}}$ and GP3/$\overline{\text{MCLR}}$ configured as input with internal pull-up enabled.

6: This spec. applies when GP3/$\overline{\text{MCLR}}$ is configured as an input with pull-up disabled. The leakage current of the $\overline{\text{MCLR}}$ circuit is higher than the standard I/O logic.

12.5 Timing Parameter Symbology

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS	3. TCC:ST (I ^{2}C specifications only)
2. TppS	4. Ts (I^{2}C specifications only)

T			
F	Frequency	T	Time

Lowercase letters (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{\text{RD}}$
cs	$\overline{\text{CS}}$	rw	$\overline{\text{RD}}$ or $\overline{\text{WR}}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{\text{SS}}$
dt	Data in	t0	T0CKI
io	I/O port	t1	T1CKI
mc	$\overline{\text{MCLR}}$	wr	$\overline{\text{WR}}$

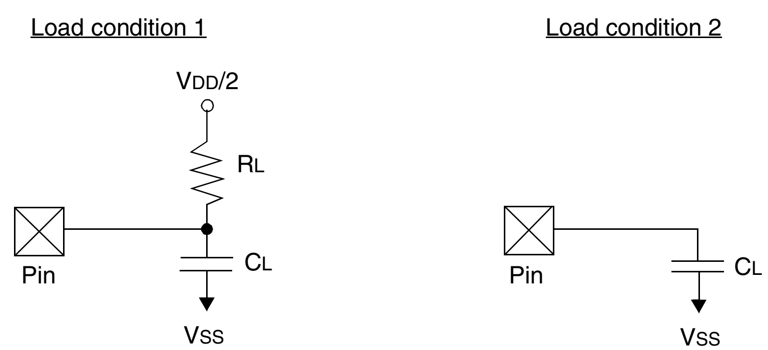
Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedance)	V	Valid
L	Low	Z	Hi-impedance
I^{2}C only			
AA	output access	High	High
BUF	Bus free	Low	Low

TCC:ST (I^{2}C specifications only)

CC			
HD	Hold	SU	Setup
ST			
DAT	DATA input hold	STO	STOP condition
STA	START condition		

FIGURE 12-4: LOAD CONDITIONS



R_L = 464Ω

C_L = 50 pF for all pins except OSC2

15 pF for OSC2 output

12.6 Timing Diagrams and Specifications

FIGURE 12-5: EXTERNAL CLOCK TIMING

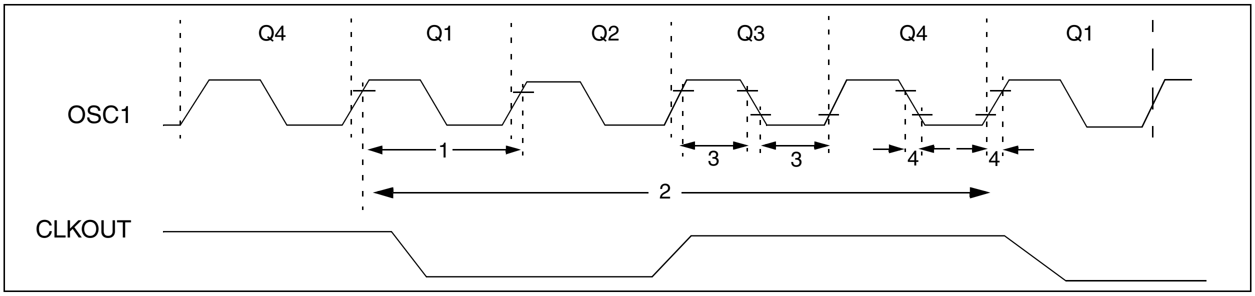


TABLE 12-1: CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	FOSC	**External CLKIN Frequency (Note 1)**	DC	—	4	MHz	XT and EXTRC osc mode
			DC	—	4	MHz	HS osc mode (PIC12CE67X-04)
			DC	—	10	MHz	HS osc mode (PIC12CE67X-10)
			DC	—	200	kHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	EXTRC osc mode
			.455	—	4	MHz	XT osc mode
			4	—	4	MHz	HS osc mode (PIC12CE67X-04)
			4	—	10	MHz	HS osc mode (PIC12CE67X-10)
			5	—	200	kHz	LP osc mode
1T	OSC	**External CLKIN Period (Note 1)**	250	—	—	ns	XT and EXTRC osc mode
			250	—	—	ns	HS osc mode (PIC12CE67X-04)
			100	—	—	ns	HS osc mode (PIC12CE67X-10)
			5—		—	µs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	EXTRC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	250	ns	HS osc mode (PIC12CE67X-04)
			100	—	250	ns	HS osc mode (PIC12CE67X-10)
			5—		—	µsL	P osc mode
2T	CY	**Instruction Cycle Time (Note 1)**	400	—	DC	ns	TCY = 4/FOSC
3	TosL, TosH	**External Clock in (OSC1) High or Low Time**	50 —		—	ns	XT oscillator
			2.5	—	—	µs	LP oscillator
			10	—	—	ns	HS oscillator
4T	osR, TosF	**External Clock in (OSC1) Rise or Fall Time**	— —		25	ns	XT oscillator
			— —		50	ns	LP oscillator
			—	—	15	ns	HS oscillator

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1/CLKIN pin.
When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices. OSC2 is disconnected (has no loading) for the PIC12C67X.

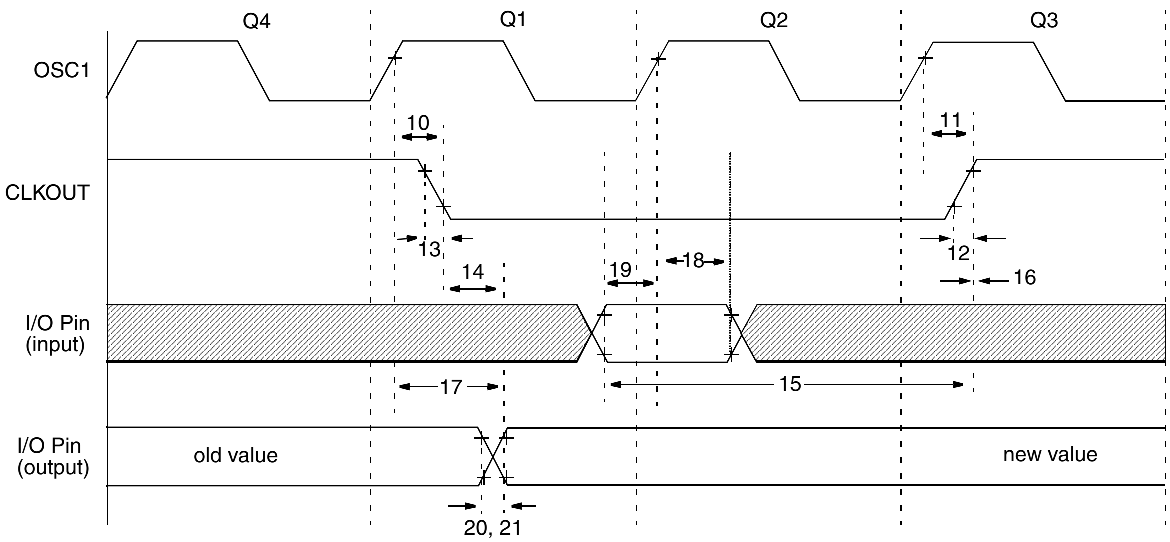
TABLE 12-2: CALIBRATED INTERNAL RC FREQUENCIES - PIC12C671, PIC12C672, PIC12CE673, PIC12CE674, PIC12LC671, PIC12LC672, PIC12LCE673, PIC12LCE674

AC Characteristics	Standard Operating Conditions (unless otherwise specified) Operating Temperature 0°C ≤ TA ≤ +70°C (commercial), –40°C ≤ TA ≤ +85°C (industrial), –40°C ≤ TA ≤ +125°C (extended) Operating Voltage VDD range is described in Section 10.1						
Parameter No.	**Sym**	**Characteristic**	**Min***	**Typ(1)**	**Max***	**Units**	**Conditions**
		Internal Calibrated RC Frequency	3.65	4.00	4.28	MHz	VDD = 5.0V
		Internal Calibrated RC Frequency	3.55	4.00	4.31	MHz	VDD = 2.5V

* These parameters are characterized but not tested.

Note 1: Data in the Typical ("Typ") column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 12-6: CLKOUT AND I/O TIMING



Note: Refer to Figure 12-4 for load conditions.

TABLE 12-3: CLKOUT AND I/O TIMING REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
10*	TosH2ckL	OSC1↑ to CLKOUT↓ —			75	200	ns	Note 1
11*	TosH2ckH	OSC1↑ to CLKOUT↑ —			75	200	ns	Note 1
12*	TckR	CLKOUT rise time		—	35	100	ns	Note 1
13*	TckF	CLKOUT fall time		—	35	100	ns	Note 1
14*	TckL2ioV	CLKOUT ↓ to Port out valid		—	—	0.5TCY + 20	ns	Note 1
15*	TioV2ckH	Port in valid before CLKOUT ↑ T		OSC + 200	—	—	ns	Note 1
16*	TckH2ioI	Port in hold after CLKOUT ↑ 0—				—	ns	Note 1
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid		—	50	150	ns	
18*	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	PIC12**C**67X	100	—	—	ns	
18A*			PIC12**LC**67X	200	—	—	ns	
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)		0—		—	ns	
20*	TioR	Port output rise time	PIC12**C**67X	—	10	40	ns	
20A*			PIC12**LC**67X	—	—	80	ns	
21*	TioF	Port output fall time	PIC12**C**67X	—	10	40	ns	
21A*			PIC12**LC**67X	—	—	80	ns	
22††*	Tinp	GP2/INT pin high or low time		TCY	—		ns	
23††*	Trbp	GP0/GP1/GP3 change INT high or low time		TCY	—		ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

†† These parameters are asynchronous events not related to any internal clock edge.

Note 1: Measurements are taken in EXTRC and INTRC modes where CLKOUT output is 4 x TOSC.

FIGURE 12-7: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, AND POWER-UP TIMER TIMING

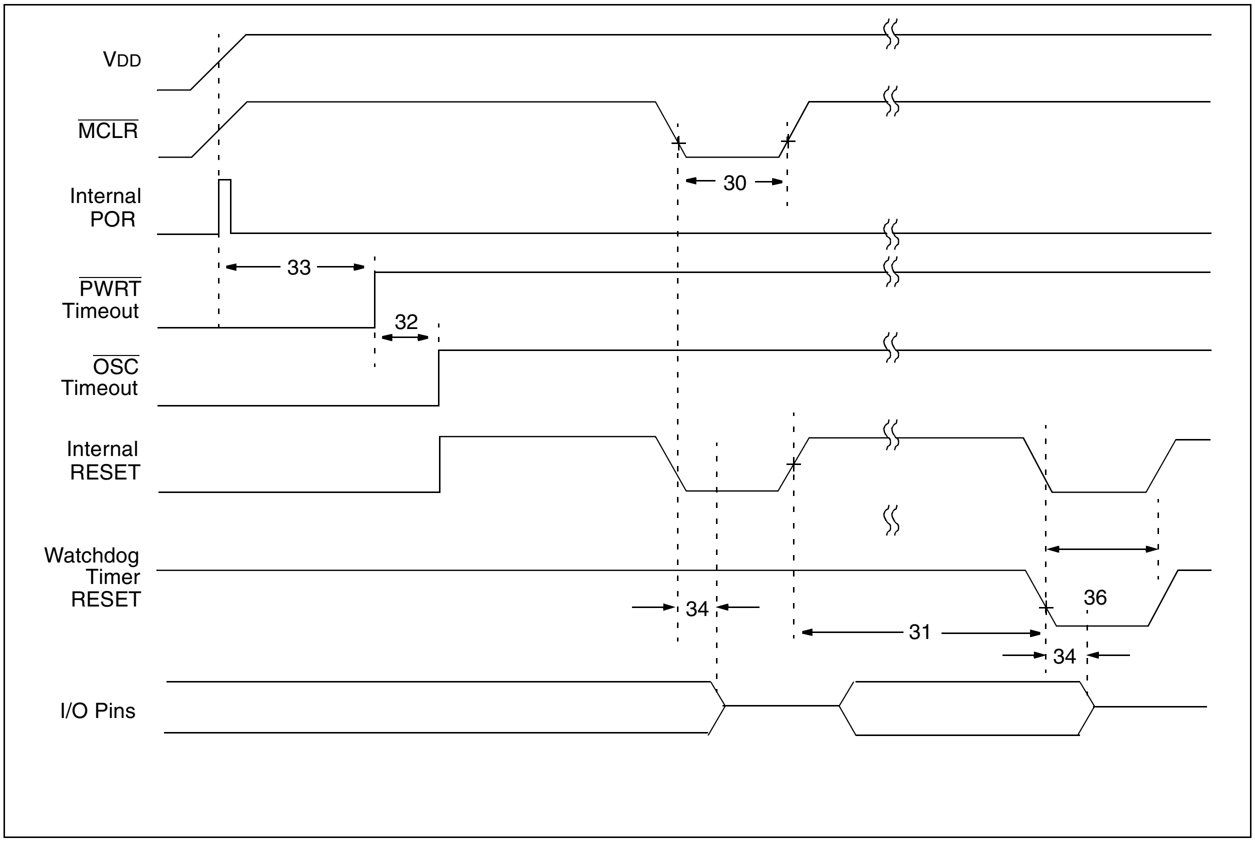


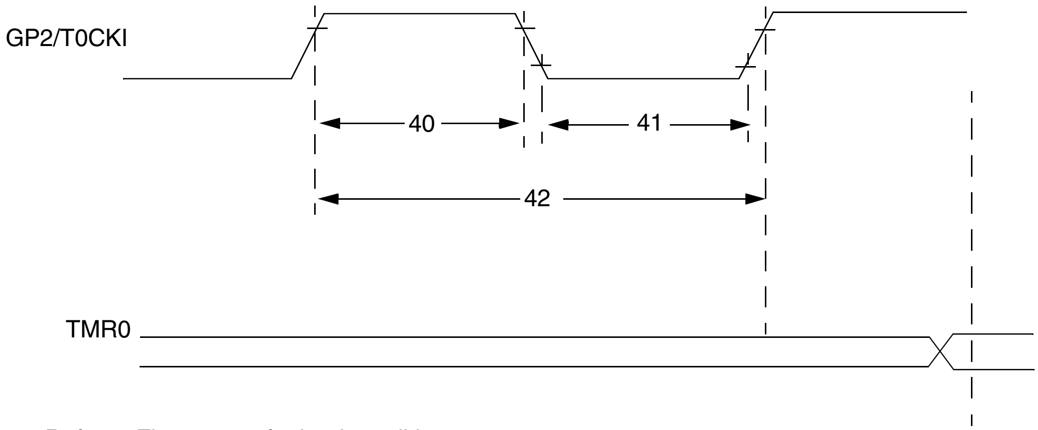
TABLE 12-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{\text{MCLR}}$ Pulse Width (low)	2	—	—	µsV	DD = 5V, –40°C to +125°C
31*	Twdt	Watchdog Timer Time-out Period (No Prescaler)	71	8	33	ms	V_{DD} = 5V, –40°C to +125°C
32	Tost	Oscillation Start-up Timer Period	—	1024T_{OSC}	——		T_{OSC} = OSC1 period
33*	Tpwrt	Power up Timer Period	28	72	132	ms	V_{DD} = 5V, –40°C to +125°C
34	TIOZ	I/O Hi-impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset	—	—	2.1	µs	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 12-8: TIMER0 CLOCK TIMINGS



Note: Refer to Figure 12-4 for load conditions.

TABLE 12-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5TCY + 20	—	—	ns	Must also meet parameter 42
			With Prescaler	10	—	—	ns	
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5TCY + 20	—	—	ns	Must also meet parameter 42
			With Prescaler	10	—	—	ns	
42*	Tt0P	T0CKI Period	No Prescaler	TCY + 40	—	—	ns	
			With Prescaler	Greater of: 20 or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (2, 4,..., 256)
48	TCKE2tmr1	Delay from external clock edge to timer increment		2TOSC	—7	Tosc	—	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 12-6: GPIO PULL-UP RESISTOR RANGES

VDD (Volts)	Temperature (°C)	Min	Typ	Max	Units
GP0/GP1					
2.5	–40	38K	42K	63K	Ω
	25	42K	48K	63K	Ω
	85	42K	49K	63K	Ω
	125	50K	55K	63K	Ω
5.5	–40	15K	17K	20K	Ω
	25	18K	20K	23K	Ω
	85	19K	22K	25K	Ω
	125	22K	24K	28K	Ω
GP3					
2.5	–40	285K	346K	417K	Ω
	25	343K	414K	532K	Ω
	85	368K	457K	532K	Ω
	125	431K	504K	593K	Ω
5.5	–40	247K	292K	360K	Ω
	25	288K	341K	437K	Ω
	85	306K	371K	448K	Ω
	125	351K	407K	500K	Ω

* These parameters are characterized but not tested.

TABLE 12-7: A/D CONVERTER CHARACTERISTICS: PIC12C671/672-04/PIC12CE673/674-04 (COMMERCIAL, INDUSTRIAL, EXTENDED) PIC12C671/672-10/PIC12CE673/674-10 (COMMERCIAL, INDUSTRIAL, EXTENDED) PIC12LC671/672-04/PIC12LCE673/674-04 (COMMERCIAL, INDUSTRIAL)

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
A01	NR	Resolution	—	—	8-bits	bit	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A02	EABS	Total absolute error	—	—	< ±1L	Sb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A03	EIL	Integral linearity error	—	—	< ±1L	Sb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A04	EDL	Differential linearity error	—	—	< ±1L	Sb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A05	EFS	Full scale error	—	—	< ±1L	Sb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A06	EOFF	Offset error	—	—	< ±1L	Sb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A10	—	Monotonicity	—	guaranteed (Note 3)	——		VSS ≤ VAIN ≤ VREF
A20	VREF	Reference voltage	2.5V	—	VDD + 0.3	V	
A25	VAIN	Analog input voltage	VSS - 0.3	—	VREF +0.3	V	
A30	ZAIN	Recommended impedance of analog voltage source	——		10.0	kΩ	
A40	IAD	A/D conversion current (VDD) PIC12**C**67X	—	180	—	μA	Average current consumption when A/D is on. (Note 1)
		PIC12**LC**67X	—	90	—	μA	
A50	IREF	VREF input current (Note 2)	10	—	1000	μA	During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see Section 8.1.
			—	—	10	μA	During A/D Conversion cycle

* These parameters are characterized but not tested.

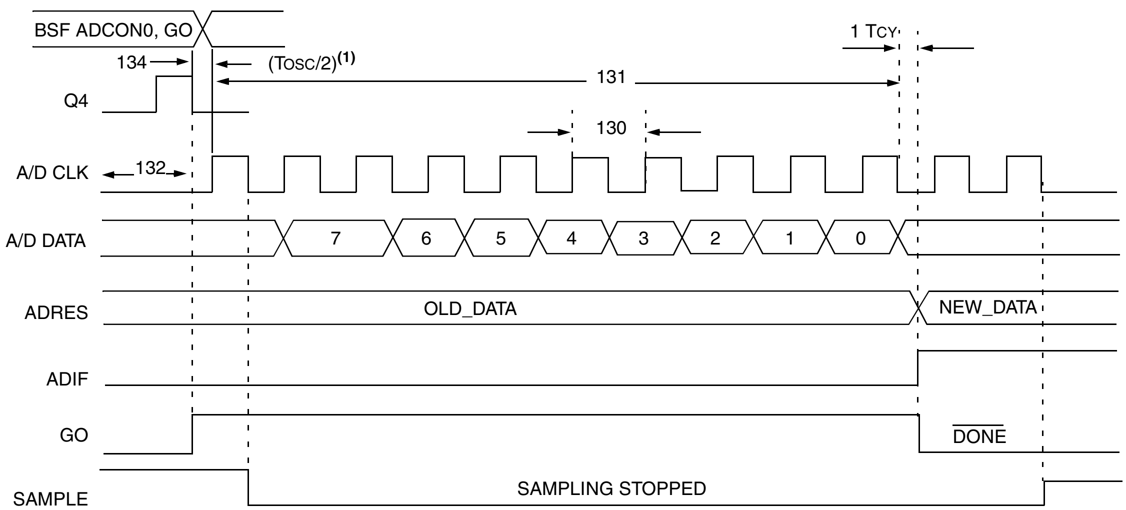
† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.

2: VREF current is from GP1 pin or VDD pin, whichever is selected as reference input.

3: The A/D conversion result never decreases with an increase in the Input Voltage, and has no missing codes.

FIGURE 12-9: A/D CONVERSION TIMING



Note 1: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.

TABLE 12-8: A/D CONVERSION REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	PIC12**C**67X	1.6	—	—	µs	TOSC based, VREF ≥ 3.0V
			PIC12**LC**67X	2.0	—	—	µsT	OSC based, VREF full range
			PIC12**C**67X	2.0	4.0	6.0	µsA	/D RC Mode
			PIC12**LC**67X	3.0	6.0	9.0	µsA	/D RC Mode
131	TCNV	Conversion time (not including S/H time) (Note 1)		11	—	11	TAD	
132	TACQ	Acquisition time		Note 2	20	—	µs	
				5*	—	—	µs	The minimum time is the amplifier setting time. This may be used if the "new" input voltage has not changed by more than 1 LSb (i.e., 20.0 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).
134	TGO	Q4 to A/D clock start		—	TOSC/2 §	—	—	If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.
135	TSWC	Switching from convert → sample time		1.5 §	—	—	TAD	

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification ensured by design.

Note 1: ADRES register may be read on the following TCY cycle.

2: See Section 8.1 for min. conditions.

TABLE 12-9: EEPROM MEMORY BUS TIMING REQUIREMENTS - PIC12CE673/674 ONLY.

AC Characteristics	Standard Operating Conditions (unless otherwise specified) Operating Temperature 0°C ≤ TA ≤ +70°C, Vcc = 3.0V to 5.5V (commercial) –40°C ≤ TA ≤ +85°C, Vcc = 3.0V to 5.5V (industrial) –40°C ≤ TA ≤ +125°C, Vcc = 4.5V to 5.5V (extended) Operating Voltage VDD range is described in Section 12.1				
Parameter	**Symbol**	**Min**	**Max**	**Units**	**Conditions**
Clock frequency	FCLK	— — —	100 100 400	kHz	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
Clock high time	THIGH	4000 4000 600	— — —	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
Clock low time	TLOW	4700 4700 1300	— — —	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
SDA and SCL rise time (Note 1)	TR	— — —	1000 1000 300	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
SDA and SCL fall time	TF	—	300	ns	(Note 1)
START condition hold time	THD:STA	4000 4000 600	— — —	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
START condition setup time	TSU:STA	4700 4700 600	— — —	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
Data input hold time	THD:DAT	0	—	ns	(Note 2)
Data input setup time	TSU:DAT	250 250 100	— — —	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
STOP condition setup time	TSU:STO	4000 4000 600	— — —	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
Output valid from clock (Note 2)	TAA	— — —	3500 3500 900	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
Bus free time: Time the bus must be free before a new transmission can start	TBUF	4700 4700 1300	— — —	ns	4.5V ≤ Vcc ≤ 5.5V (E Temp range) 3.0V ≤ Vcc ≤ 4.5V 4.5V ≤ Vcc ≤ 5.5V
Output fall time from VIH minimum to VIL maximum	TOF	20+0.1 CB	250	ns	(Note 1), CB ≤ 100 pF
Input filter spike suppression (SDA and SCL pins)	TSP	—	50	ns	(Notes 1, 3)
Write cycle time	TWC	—4		ms	
Endurance		1M	—	cycles	25°C, VCC = 5.0V, Block Mode (Note 4)

Note 1: Not 100% tested. CB = total capacitance of one bus line in pF.

2: As a transmitter, the device must provide an internal minimum delay time to bridge the undefined region (minimum 300 ns) of the falling edge of SCL and avoid unintended generation of START or STOP conditions.

3: The combined TSP and VHYS specifications are due to new Schmitt Trigger inputs which provide improved noise spike suppression. This eliminates the need for a TI specification for standard operation.

4: This parameter is not tested but ensured by characterization. For endurance estimates in a specific application, please consult the Total Endurance Model which can be obtained on Microchip's website.

NOTES:

13.0 DC AND AC CHARACTERISTICS - PIC12C671/PIC12C672/PIC12LC671/ PIC12LC672/PIC12CE673/PIC12CE674/PIC12LCE673/PIC12LCE674

The graphs and tables provided in this section are for design guidance and are not tested. In some graphs or tables the data presented are outside specified operating range (i.e., outside specified VDD range). This is for information only and devices will operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. “Typical” represents the mean of the distribution while “max” or “min” represents (mean + 3σ) and (mean – 3σ) respectively, where σ is standard deviation.

FIGURE 13-1: CALIBRATED INTERNAL RC FREQUENCY RANGE VS. TEMPERATURE (VDD = 5.0V) (INTERNAL RC IS CALIBRATED TO 25°C, 5.0V)

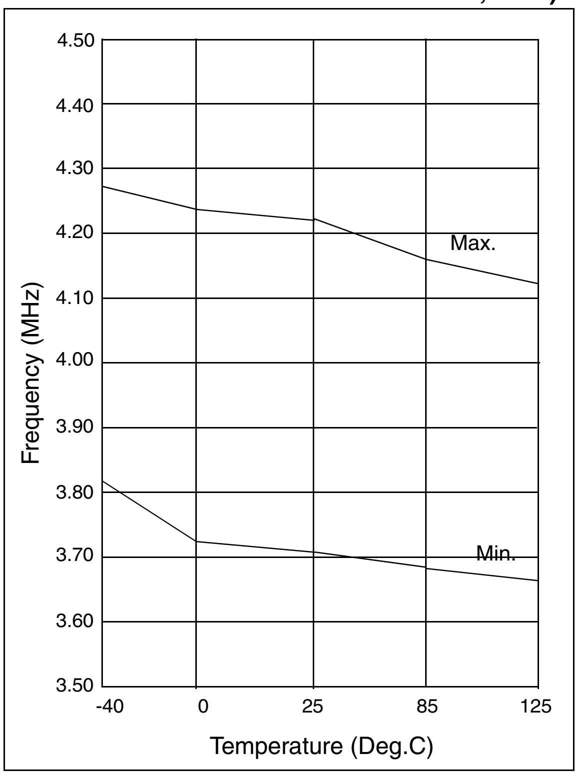


FIGURE 13-2: CALIBRATED INTERNAL RC FREQUENCY RANGE VS. TEMPERATURE (VDD = 2.5V) (INTERNAL RC IS CALIBRATED TO 25°C, 5.0V)

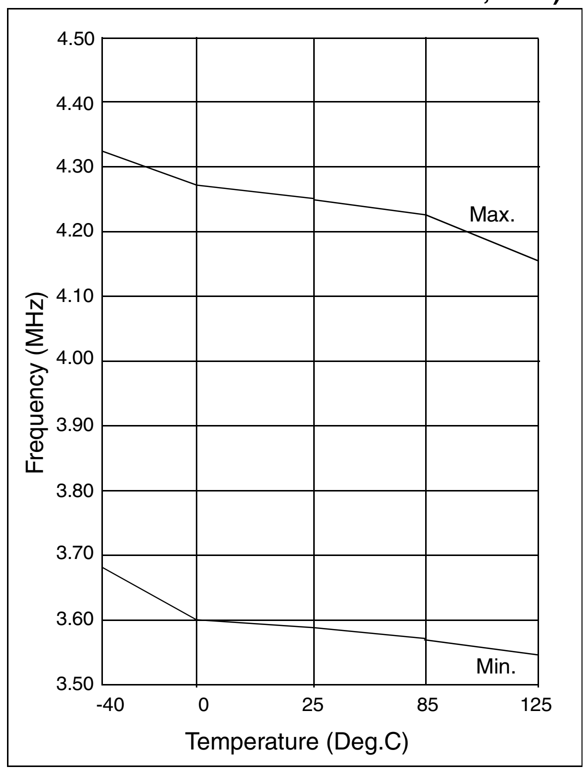


TABLE 13-1: DYNAMIC IDD (TYPICAL) - WDT ENABLED, 25°C

Oscillator	Frequency	VDD = 2.5V	VDD = 5.5V
External RC	4 MHz	400 µA*	900 µA*
Internal RC	4 MHz	400 µA	900 µA
XT	4 MHz	400 µA	900 µA
LP	32 kHz	15 µA	60 µA

*Does not include current through external R&C.

FIGURE 13-3: WDT TIMER TIME-OUT PERIOD vs. VDD

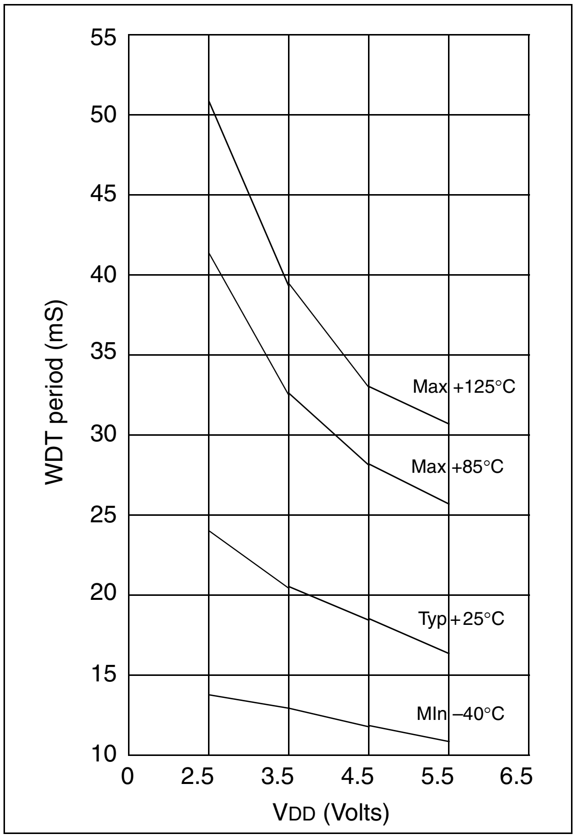


FIGURE 13-4: IOH vs. VOH, VDD = 2.5 V

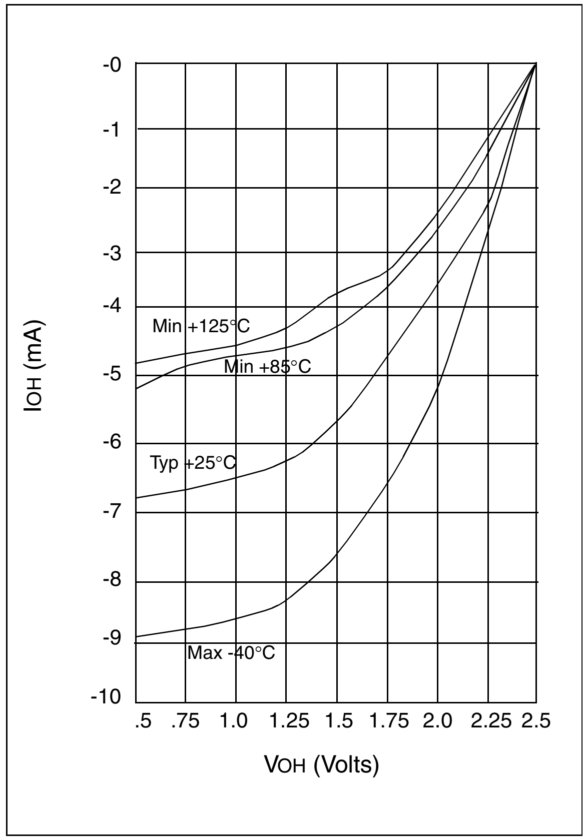


FIGURE 13-5: IOH vs. VOH, VDD = 3.5 V

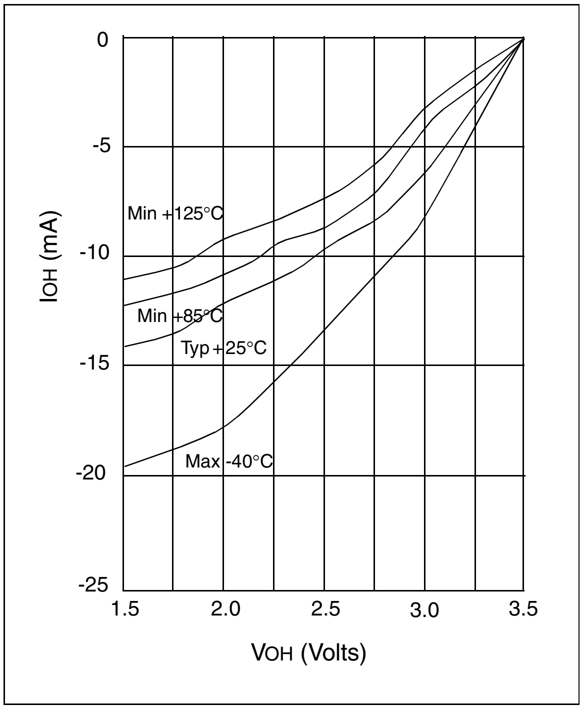


FIGURE 13-6: IOH vs. VOH, VDD = 5.5 V

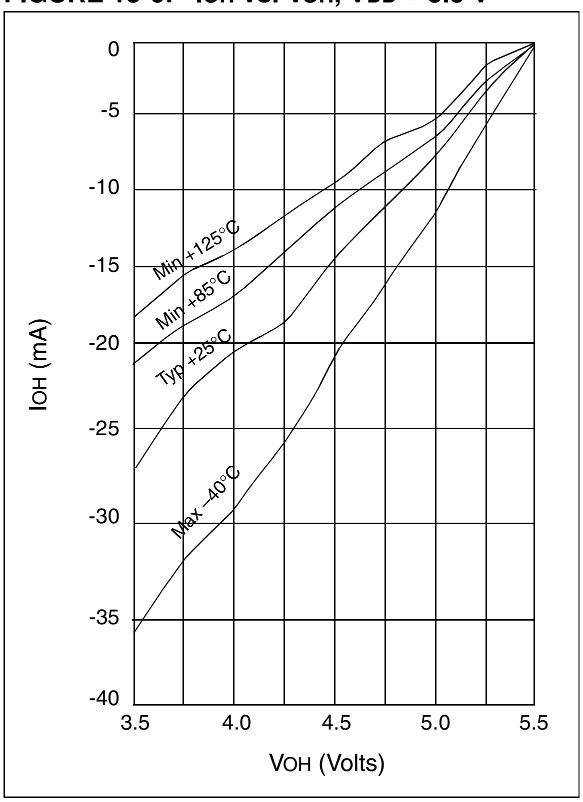


FIGURE 13-7: IOL vs. VOL, VDD = 2.5 V

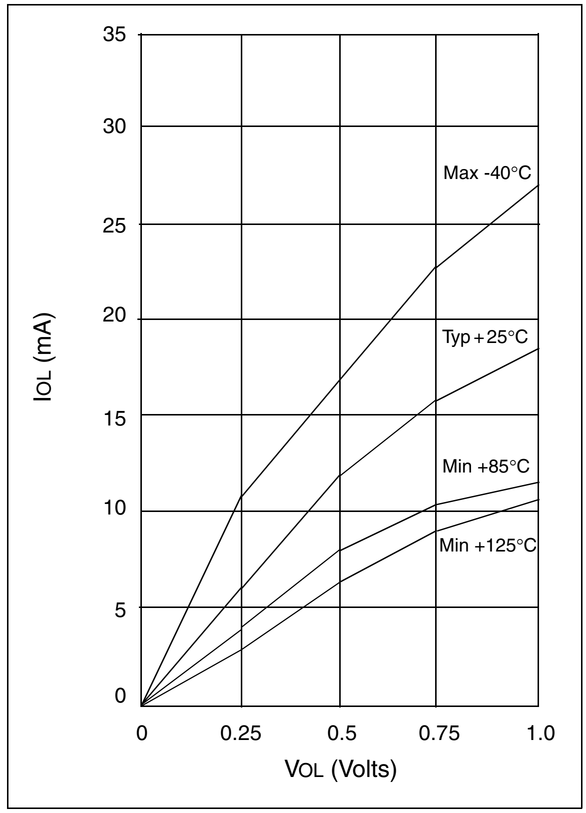


FIGURE 13-8: IOL vs. VOL, VDD = 3.5 V

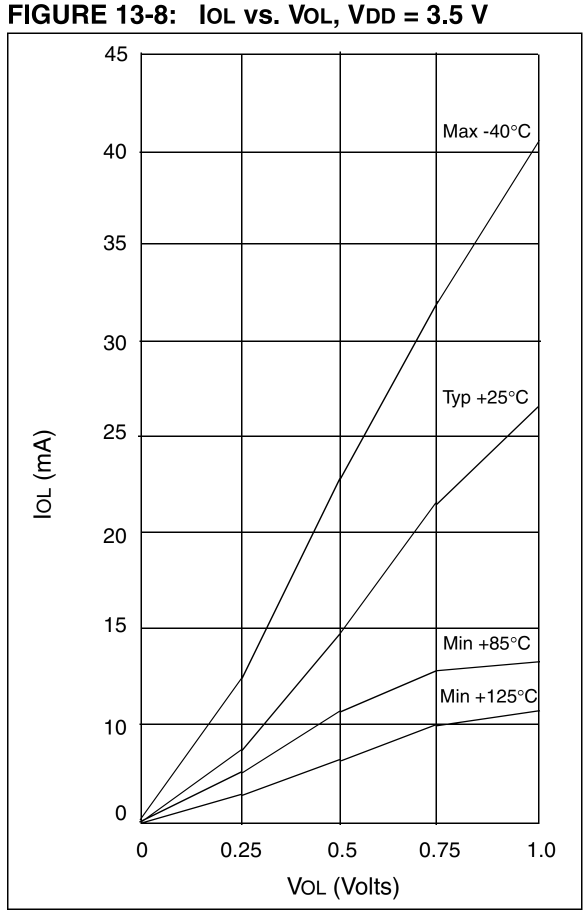


FIGURE 13-9: I_{OL} vs. V_{OL}, V_{DD} = 5.5 V

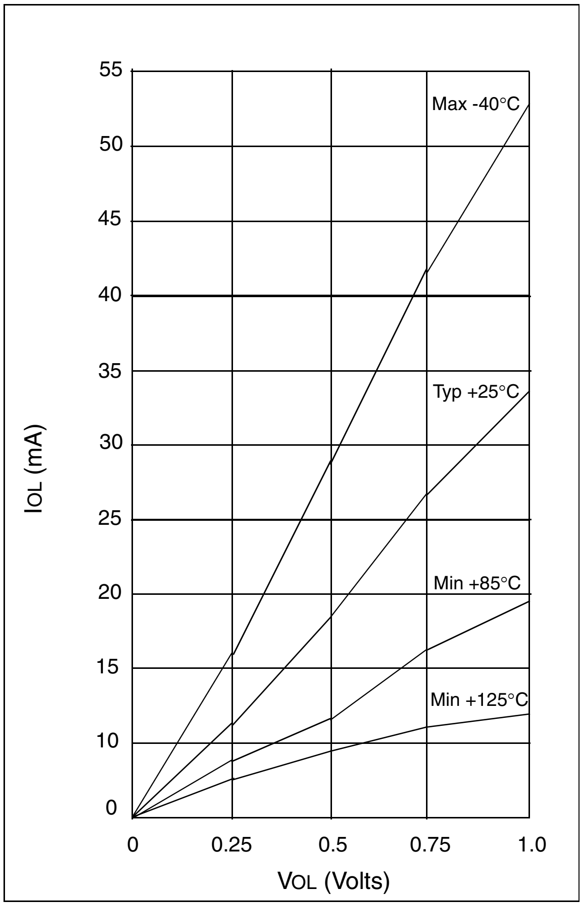


FIGURE 13-10: V_{TH} (INPUT THRESHOLD VOLTAGE) OF GPIO PINS vs. V_{DD}

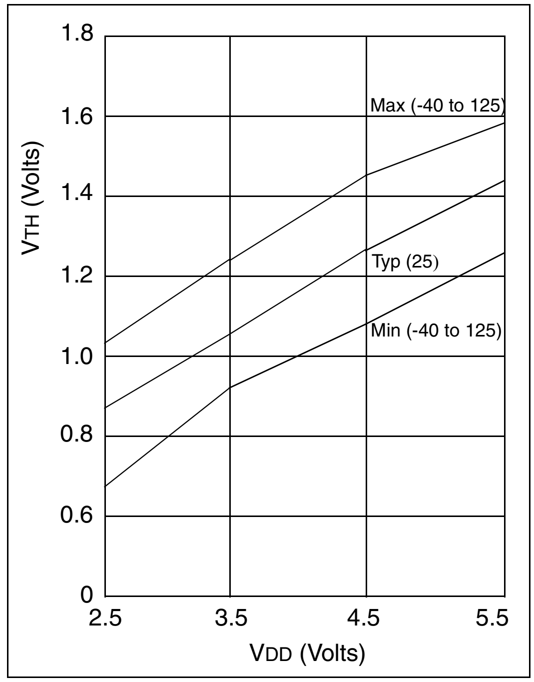
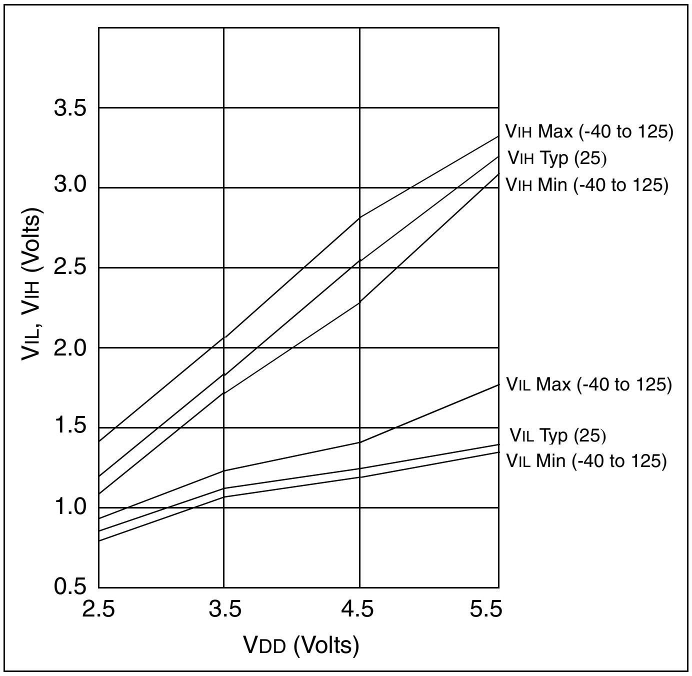


FIGURE 13-11: VIL, VIH OF NMCLR AND T0CKI vs. VDD

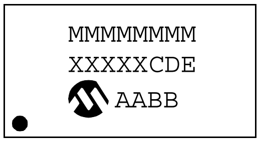


NOTES:

14.0 PACKAGING INFORMATION

14.1 Package Marking Information

8-Lead PDIP (300 mil)

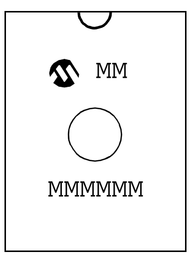


Example

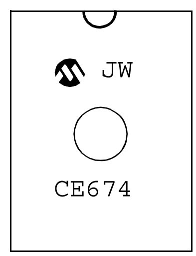
8-Lead SOIC (208 mil)

Example

8-Lead Windowed Ceramic Side Brazed (300 mil)



Example



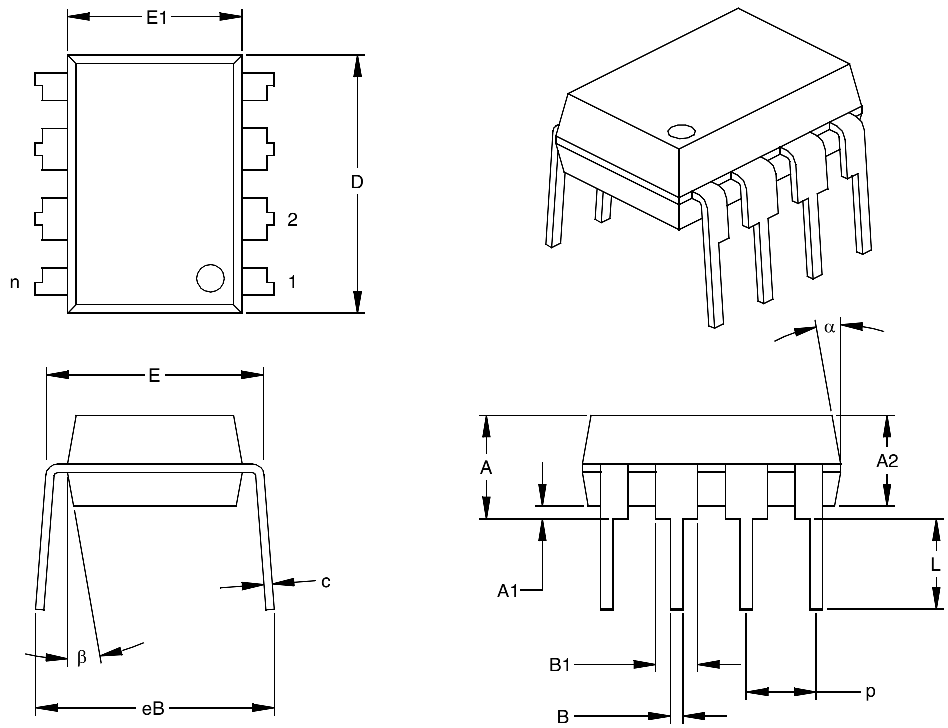
Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calendar year)
	BB	Week code (week of January 1 is week ‘01’)
	C	Facility code of the plant at which wafer is manufactured O = Outside Vendor C = 5” Line S = 6” Line H = 8” Line
	D	Mask revision number
	E	Assembly code of the plant or country of origin in which part was assembled

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

8-Lead Plastic Dual In-line (P) – 300 mil (PDIP)

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

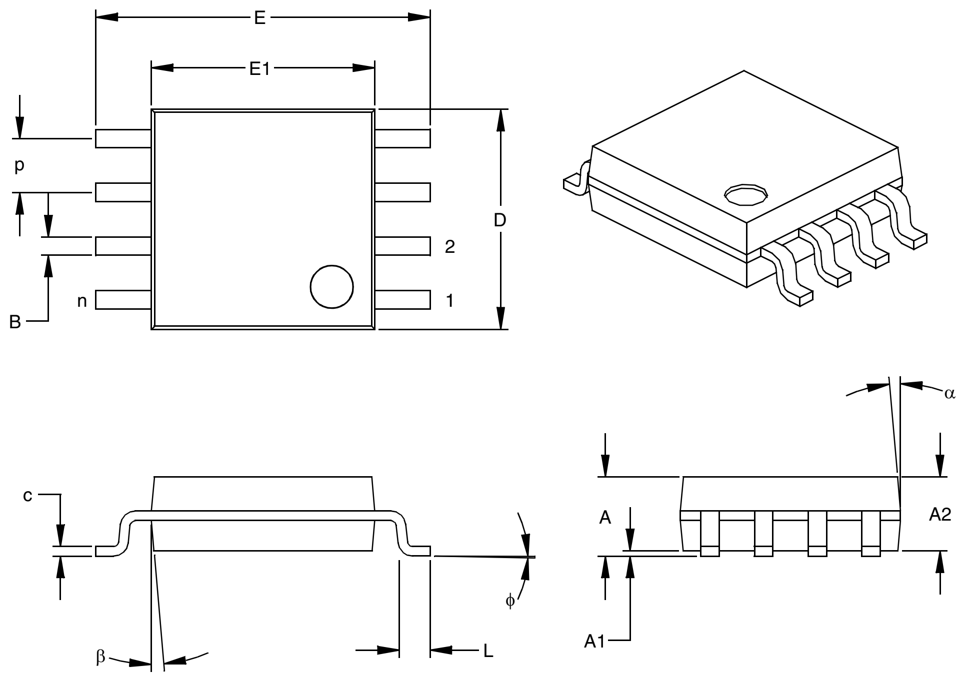


Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		88				
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.360	.373	.385	9.14	9.46	9.78
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing	eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	0	15	5	0	15
Mold Draft Angle Bottom	β	5	0	15	5	0	15

*Controlling Parameter
Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-001
Drawing No. C04-018

8-Lead Plastic Small Outline (SM) – Medium, 208 mil (SOIC)

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



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		8			8	
Pitch	p		.050			1.27	
Overall Height	A	.070	.075	.080	1.78	1.97	2.03
Molded Package Thickness	A2	.069	.074	.078	1.75	1.88	1.98
Standoff	A1	.002	.005	.010	0.05	0.13	0.25
Overall Width	E	.300	.313	.325	7.62	7.95	8.26
Molded Package Width	E1	.201	.208	.212	5.11	5.28	5.38
Overall Length	D	.202	.205	.210	5.13	5.21	5.33
Foot Length	L	.020	.025	.030	0.51	0.64	0.76
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	c	.008	.009	.010	0.20	0.23	0.25
Lead Width	B	.014	.017	.020	0.36	0.43	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

*Controlling Parameter

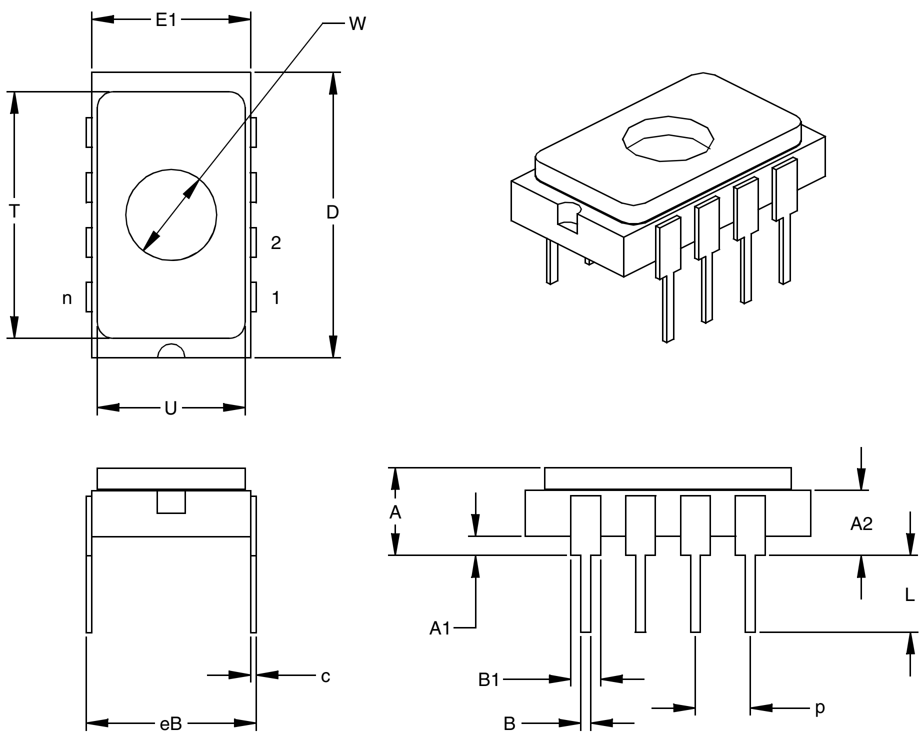
Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

Drawing No. C04-056

8-Lead Ceramic Side Brazed Dual In-line with Window (JW) – 300 mil

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



	Units	INCHES*			MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		8			8	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.145	.165	.185	3.68	4.19	4.70
Top of Body to Seating Plane	A2	.103	.123	.143	2.62	3.12	3.63
Standoff	A1	.025	.035	.045	0.64	0.89	1.14
Package Width	E1	.280	.290	.300	7.11	7.37	7.62
Overall Length	D	.510	.520	.530	12.95	13.21	13.46
Tip to Seating Plane	L	.130	.140	.150	3.30	3.56	3.81
Lead Thickness	c	.008	.010	.012	0.20	0.25	0.30
Upper Lead Width	B1	.050	.055	.060	1.27	1.40	1.52
Lower Lead Width	B	.016	.018	.020	0.41	0.46	0.51
Overall Row Spacing	eB	.296	.310	.324	7.52	7.87	8.23
Window Diameter	W	.161	.166	.171	4.09	4.22	4.34
Lid Length	T	.440	.450	.460	11.18	11.43	11.68
Lid Width	U	.260	.270	.280	6.60	6.86	7.11

*Controlling Parameter
JEDC Equivalent: MS-015
Drawing No. C04-083

APPENDIX A: COMPATIBILITY

To convert code written for PIC16C5X to PIC12C67X, the user should take the following steps:

1. Remove any program memory page select operations (PA2, PA1, PA0 bits) for `CALL`, `GOTO`.
2. Revisit any computed jump operations (write to PC or add to PC, etc.) to make sure page bits are set properly under the new scheme.
3. Eliminate any data memory page switching. Redefine data variables to reallocate them.
4. Verify all writes to STATUS, OPTION, and FSR registers since these have changed.
5. Change reset vector to 0000h.

APPENDIX B: CODE FOR ACCESSING EEPROM DATA MEMORY

Please refer to our web site at www.microchip.com for code availability.

APPENDIX C: REVISION HISTORY

Revision C (January 2013)

Added a note to each package outline drawing.

NOTES:

INDEX

NOTES:

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user’s guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip’s customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under “Support”, click on “Customer Change Notification” and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

TO: Technical Publications Manager

RE: Reader Response

Total Pages Sent ________

From: Name ______________________________

Company ______________________________

Address ______________________________

City / State / ZIP / Country ______________________________

Telephone: (_______) _________ - _________

FAX: (______) _________ - _________

Application (optional):

Would you like a reply? ___Y ___N

Device: LPIC16xxxxxx family

terature Number: DS30561C

Questions:

1. What are the best features of this document?

____

____

2. How does this document meet your hardware and software development needs?

____

____

3. Do you find the organization of this document easy to follow? If not, why?

____

____

4. What additions to the document do you think would enhance the structure and subject?

____

____

5. What deletions from the document could be made without affecting the overall usefulness?

____

____

6. Is there any incorrect or misleading information (what and where)?

____

____

7. How would you improve this document?

____

____

PIC12C67X PRODUCT IDENTIFICATION SYSTEM

PART NO. -XX X /XX XXX

Field	Code	Description
Pattern:		Special Requirements
Package:	P	= 300 mil PDIP
	JW	= 300 mil Windowed Ceramic Side Brazed
	SM	= 208 mil SOIC
Temperature Range:	-=	0°C to +70°C
	I=	-40°C to +85°C
	E=	-40°C to +125°C
Frequency Range:	04	= 4 MHz/200 kHz
	10	= 10 MHz
Device		PIC12CE673 PIC12CE674 PIC12LCE673 PIC12LCE674 PIC12C671 PIC12C672 PIC12C671T (Tape & reel for SOIC only) PIC12C672T (Tape & reel for SOIC only) PIC12LC671 PIC12LC672 PIC12LC671T (Tape & reel for SOIC only) PIC12LC672T (Tape & reel for SOIC only)

Examples

a) PIC12CE673-04/P Commercial Temp., PDIP Package, 4 MHz, normal VDD limits

b) PIC12CE673-04I/P Industrial Temp., PDIP package, 4 MHz, normal VDD limits

c) PIC12CE673-10I/P Industrial Temp., PDIP package, 10 MHz, normal VDD limits

d) PIC12C671-04/P Commercial Temp., PDIP Package, 4 MHz, normal VDD limits

e) PIC12C671-04I/SM Industrial Temp., SOIC package, 4 MHz, normal VDD limits

f) PIC12C671-04I/P Industrial Temp., PDIP package, 4 MHz, normal VDD limits

* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type (including LC devices).

Sales and Support

Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Worldwide Site (www.microchip.com)

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

ISO/TS 16949

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC^{32} logo, rfPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. & KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

Printed on recycled paper.

ISBN: 9781620769287

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

MICROCHIP

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

11/29/12