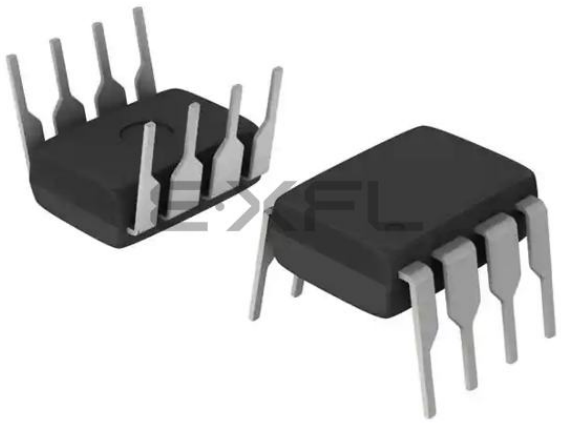


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	10MHz
Connectivity	-
Peripherals	POR, WDT
Number of I/O	5
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	OTP
EEPROM Size	16 x 8
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 4x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	8-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic12ce674-10e-p

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.

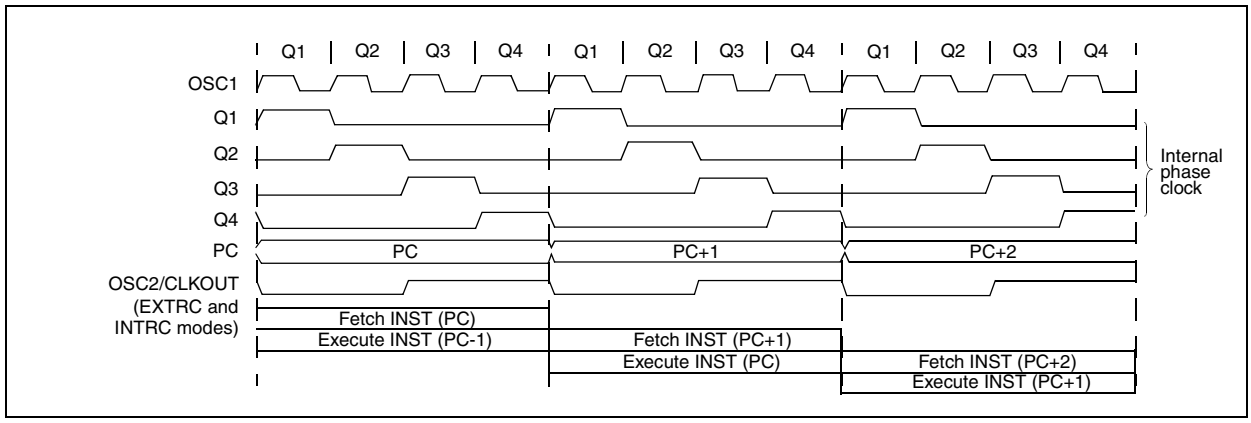
3.2 Instruction Flow/Pipelining

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).

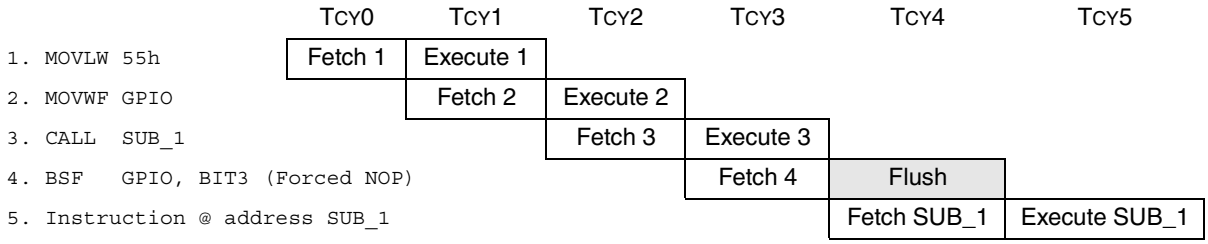
A fetch cycle begins with the program counter (PC) incrementing in Q1.

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



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.

4.2.2.4 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'

4.5 Indirect Addressing, INDF and FSR Registers

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

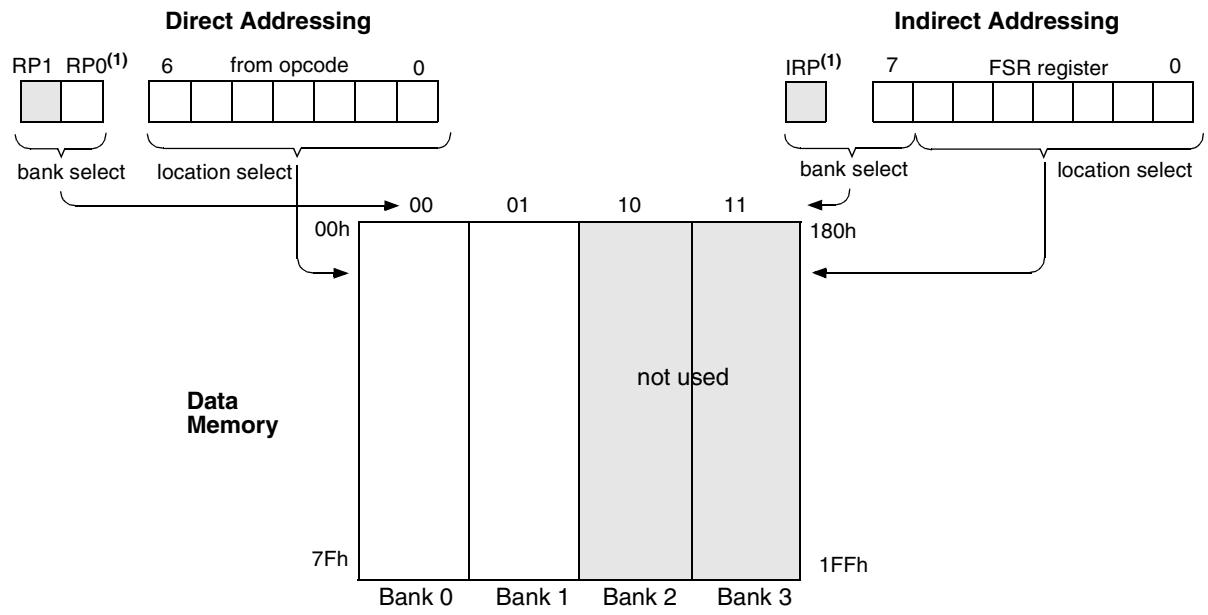
Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF Register itself indirectly (FSR = '0') will read 00h. Writing to the INDF Register indirectly results in a no-operation (although status bits may be affected). An effective 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
```

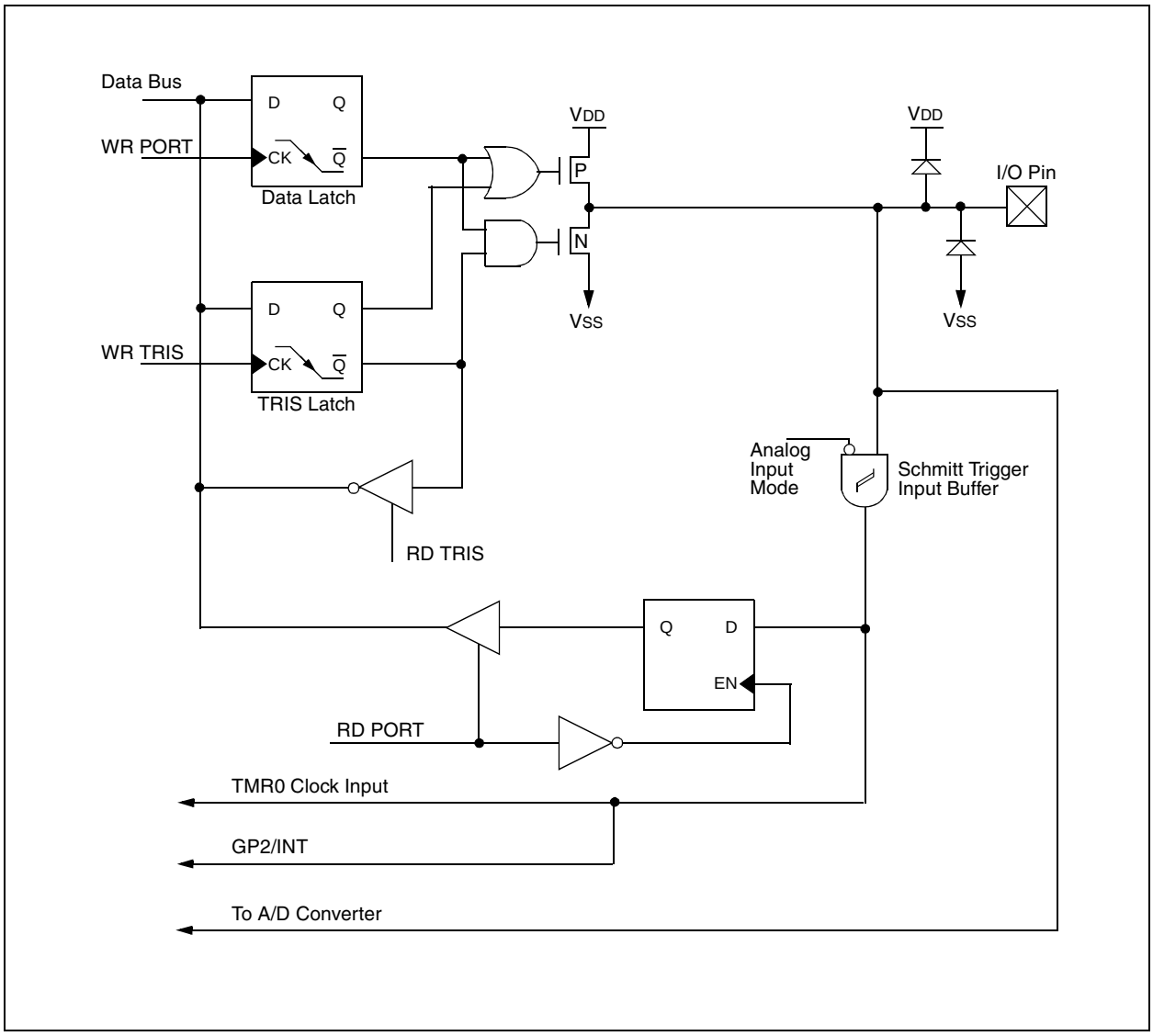
FIGURE 4-4: DIRECT/INDIRECT ADDRESSING



For register file map detail see Figure 4-2.

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

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



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 V_{CC} threshold detector circuit, which disables the internal erase/write logic if the V_{CC} is below minimum V_{DD}. 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

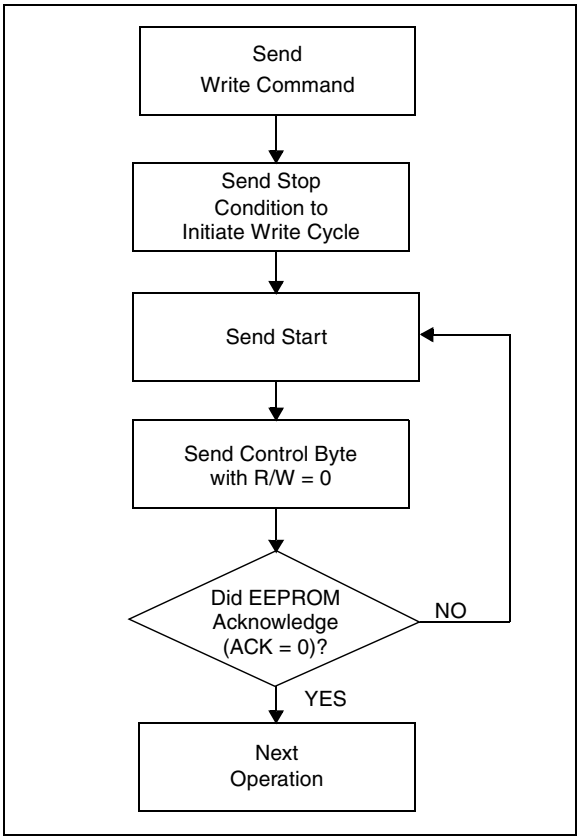
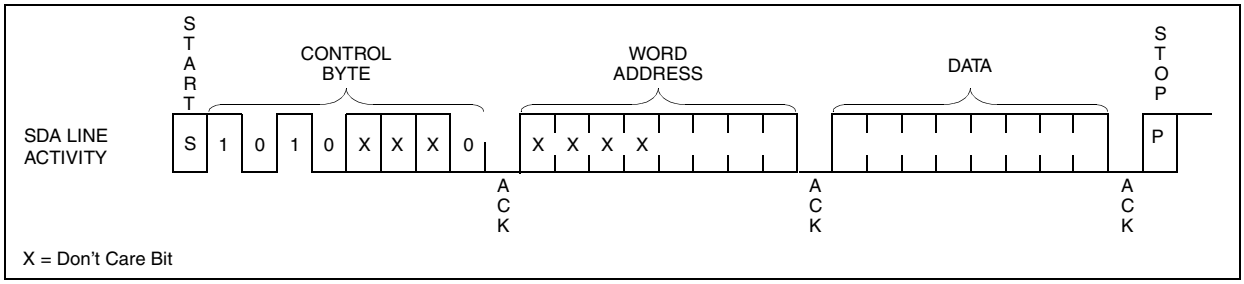


FIGURE 6-7: BYTE WRITE



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 (V_{DD}) or the voltage level on the GP1/AN1/V_{REF} 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 = F_{OSC}/2
01 = F_{OSC}/8
10 = F_{OSC}/32
11 = F_{RC} (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

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

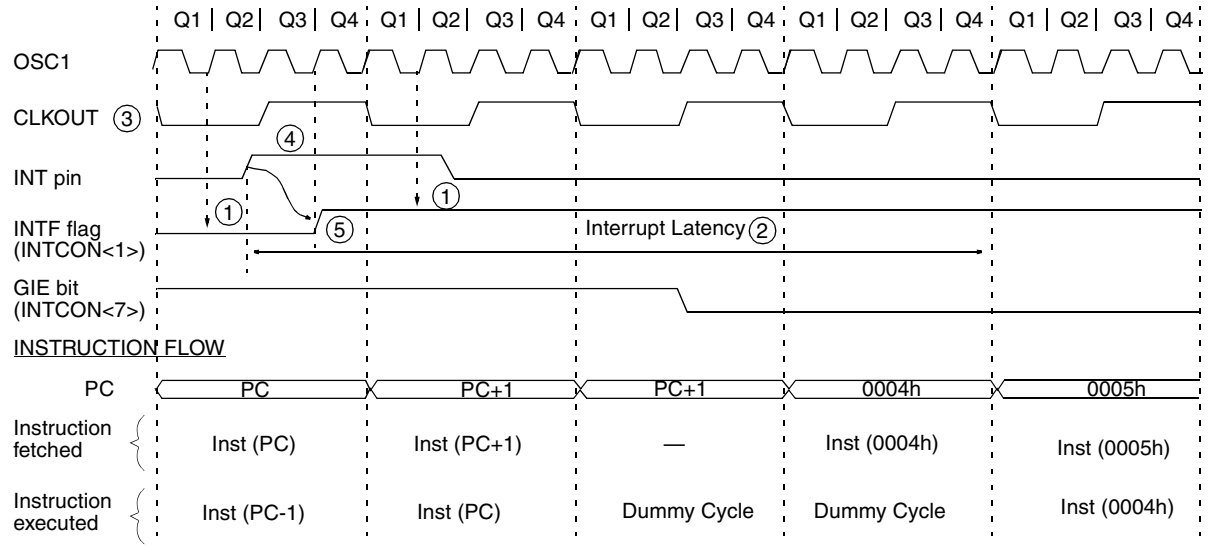
bit 4: **$\overline{\text{PWRTE}}$:** Power-up Timer Enable bit
- 1 = PWRT disabled
- 0 = PWRT enabled

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.

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 → 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
```

BCF **Bit Clear f**

Syntax: [*label*] BCF f,b

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

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

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

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_CO
TRUE    •       DE
        •
        •
```

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

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) → (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 → (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) → (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) → (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 ≤ k ≤ 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 TABLE • ;W now has table value • • 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

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:
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.

12.2 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.

12.6 Timing Diagrams and Specifications

FIGURE 12-5: EXTERNAL CLOCK TIMING

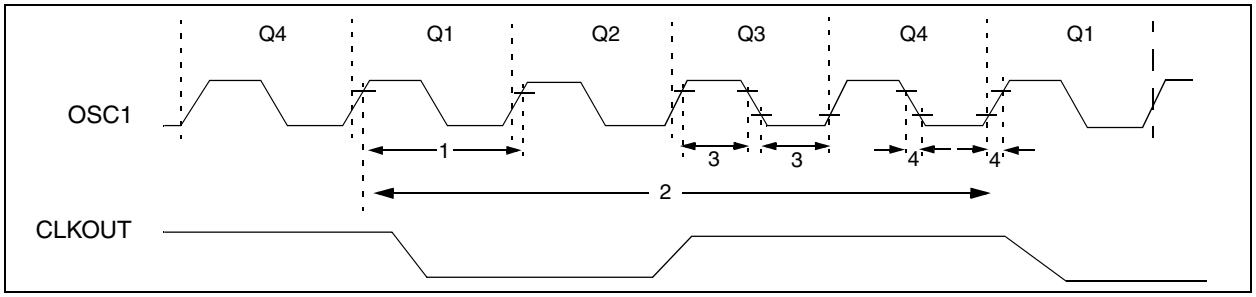


TABLE 12-1: CLOCK TIMING REQUIREMENTS

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
1	TOSC	**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	—	—	µs	LP osc mode
2	TCY	**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
4	TosR, 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 12-7: 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		—	—	< ±1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A03	EIL	Integral linearity error		—	—	< ±1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A04	EDL	Differential linearity error		—	—	< ±1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A05	EFS	Full scale error		—	—	< ±1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A06	EOFF	Offset error		—	—	< ±1	LSb	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.

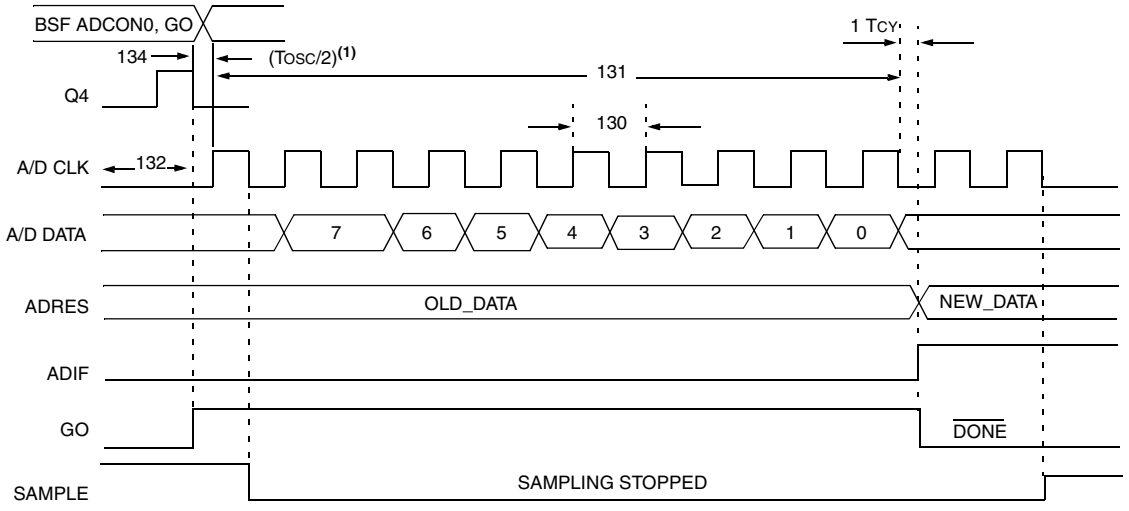
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

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	—	—	µs	TOSC based, VREF full range
			PIC12**C**67X	2.0	4.0	6.0	µs	A/D RC Mode
			PIC12**LC**67X	3.0	6.0	9.0	µs	A/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.

FIGURE 13-5: I_{OH} vs. V_{OH}, V_{DD} = 3.5 V

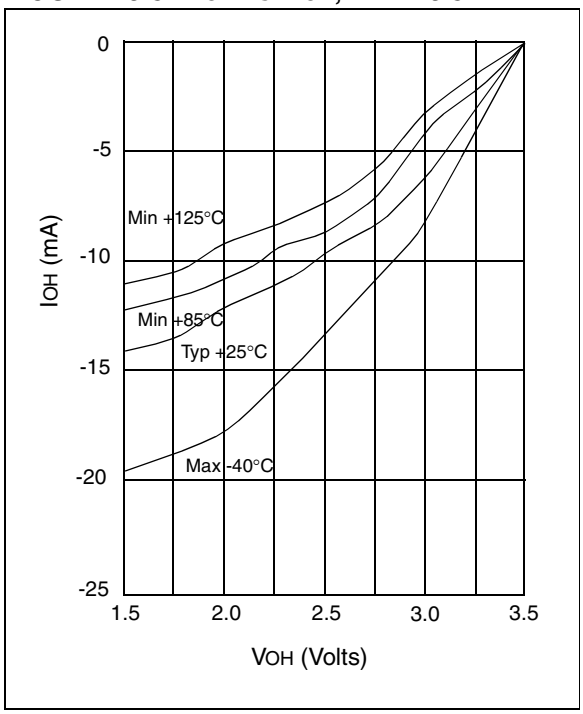


FIGURE 13-6: I_{OH} vs. V_{OH}, V_{DD} = 5.5 V

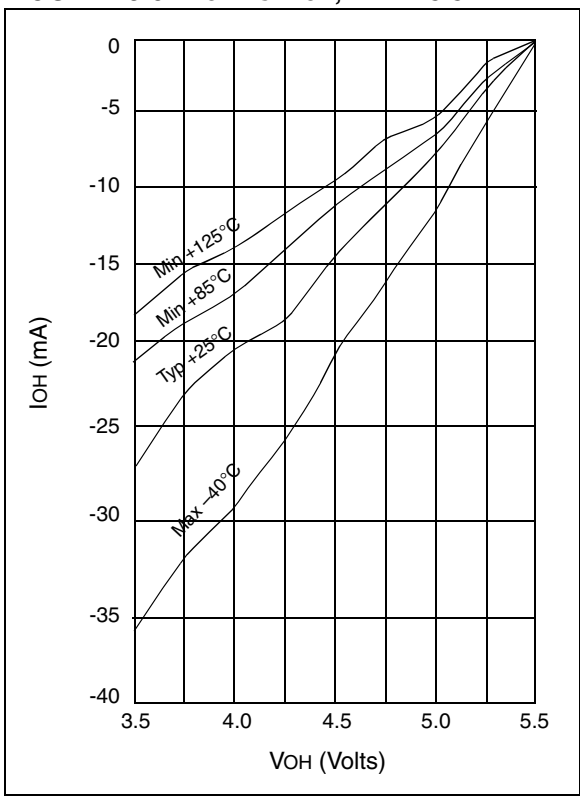


FIGURE 13-7: I_{OL} vs. V_{OL}, V_{DD} = 2.5 V

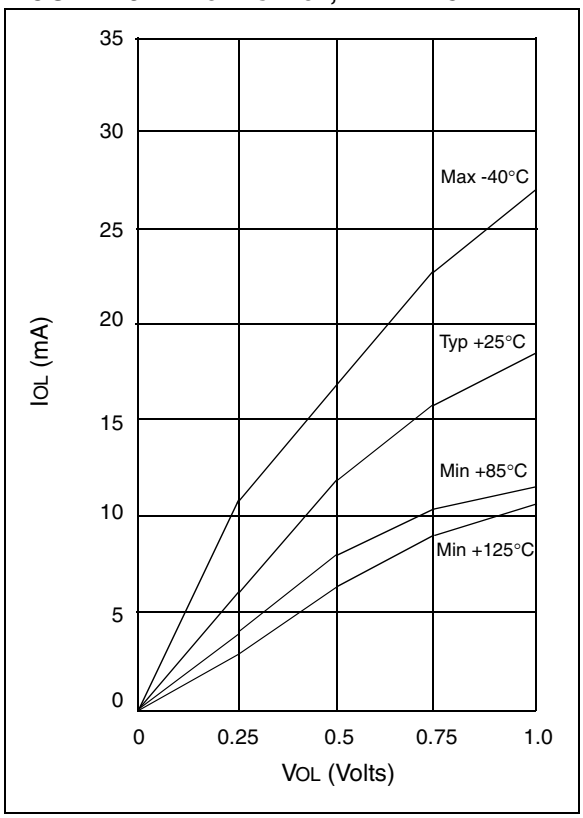


FIGURE 13-8: I_{OL} vs. V_{OL}, V_{DD} = 3.5 V

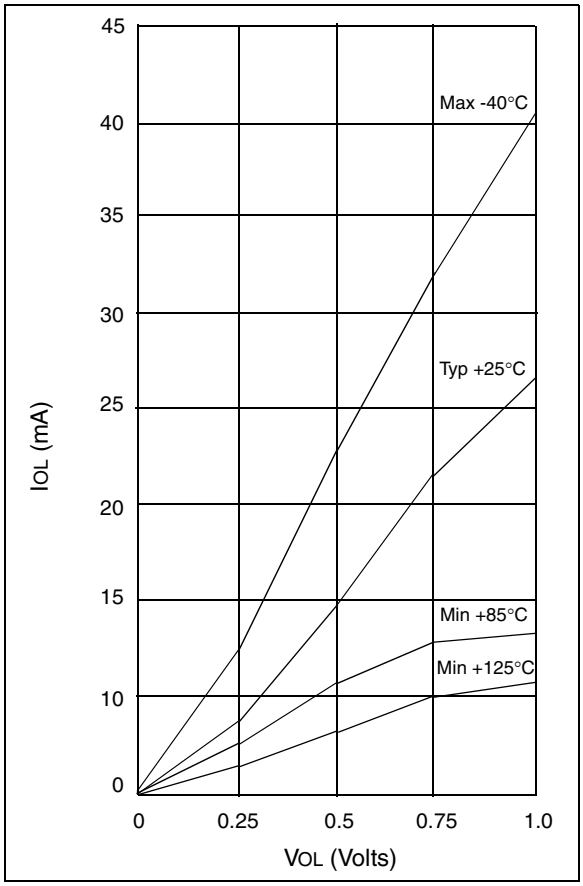


FIGURE 13-11: VIL, VIH OF NMCLR AND T0CKI vs. VDD