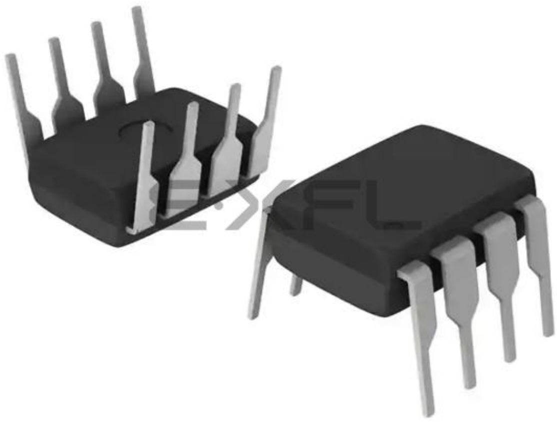


Welcome to E-XFL.COM

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	4MHz
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/pic12ce674-04i-p

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.

4.0 MEMORY ORGANIZATION

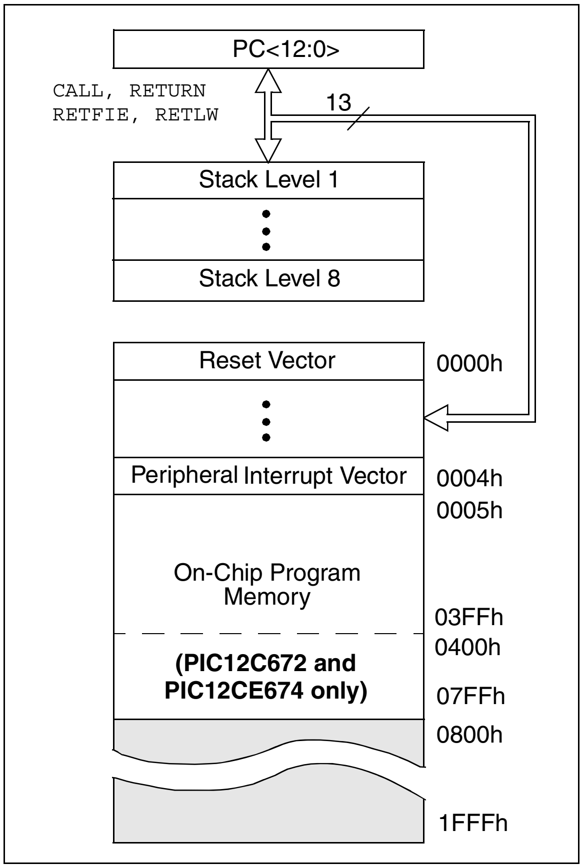
4.1 Program Memory Organization

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 → Bank 1

RP0 (STATUS<5>) = 0 → 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).

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	PEIE	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'.

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'

4.2.2.7 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

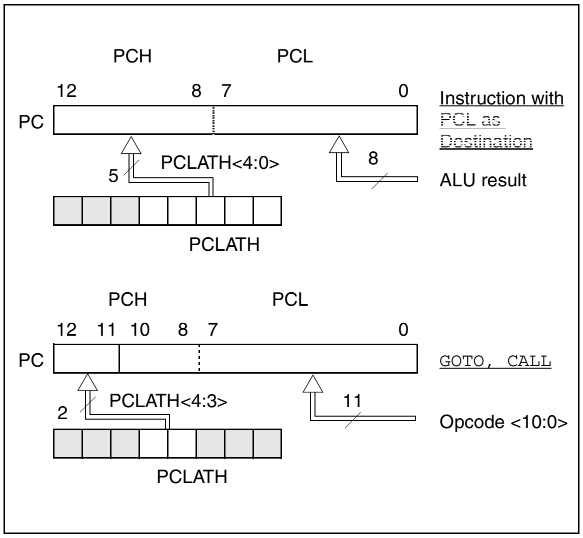
bit 1-0: **Unimplemented:** Read as ’0’

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.

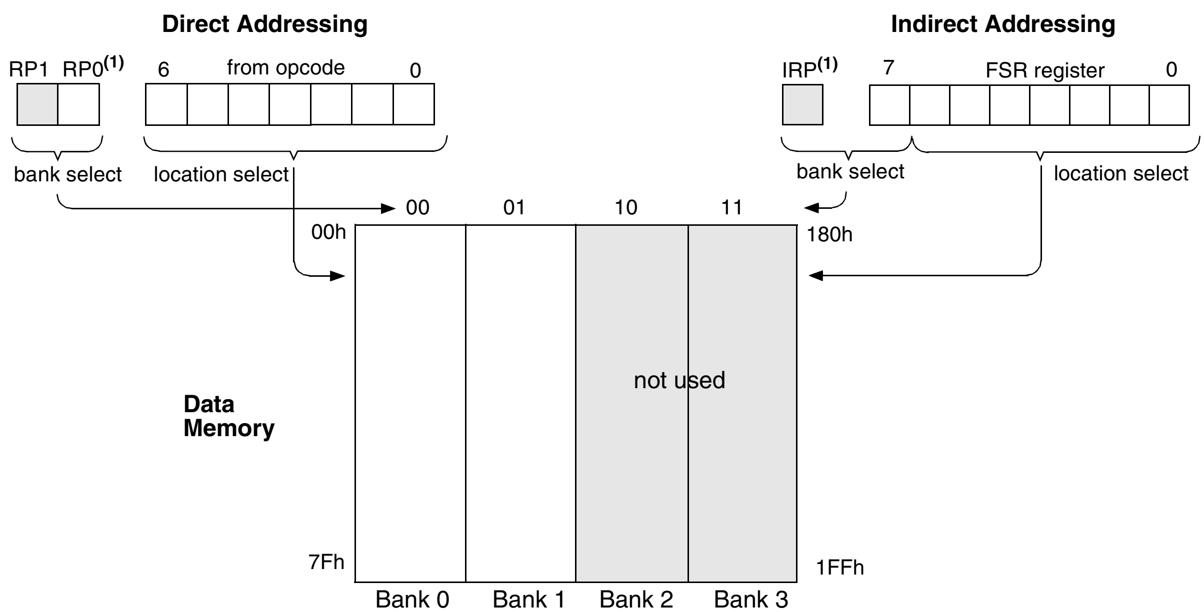
Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF Register itself indirectly (FSR = '0') will read 00h. Writing to the INDF Register indirectly results in a no-operation (although status bits may be affected). An effective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

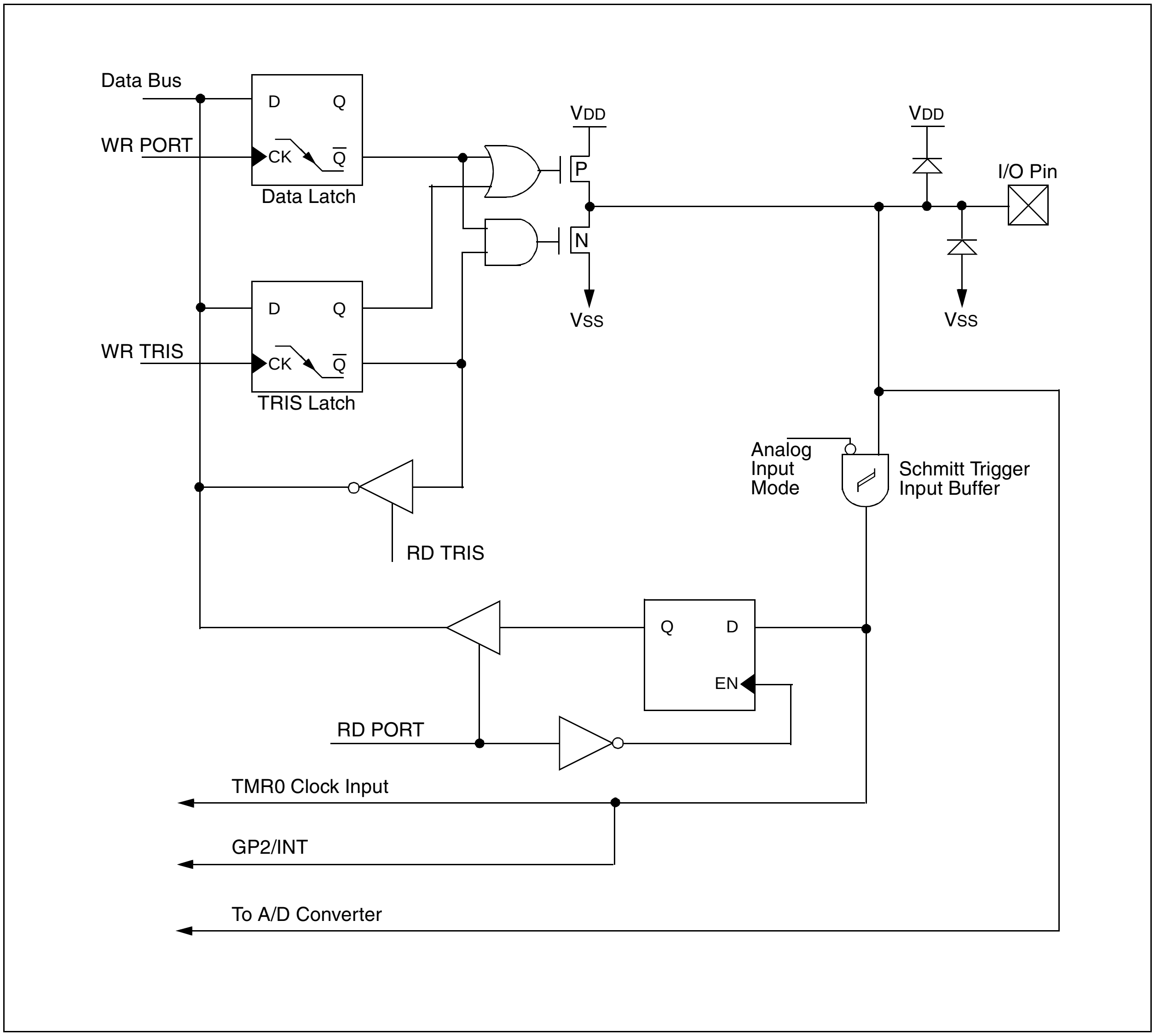
FIGURE 4-4: DIRECT/INDIRECT ADDRESSING



For register file map detail see Figure 4-2.

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

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



NOTES:

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.

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, V_{DD} 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.

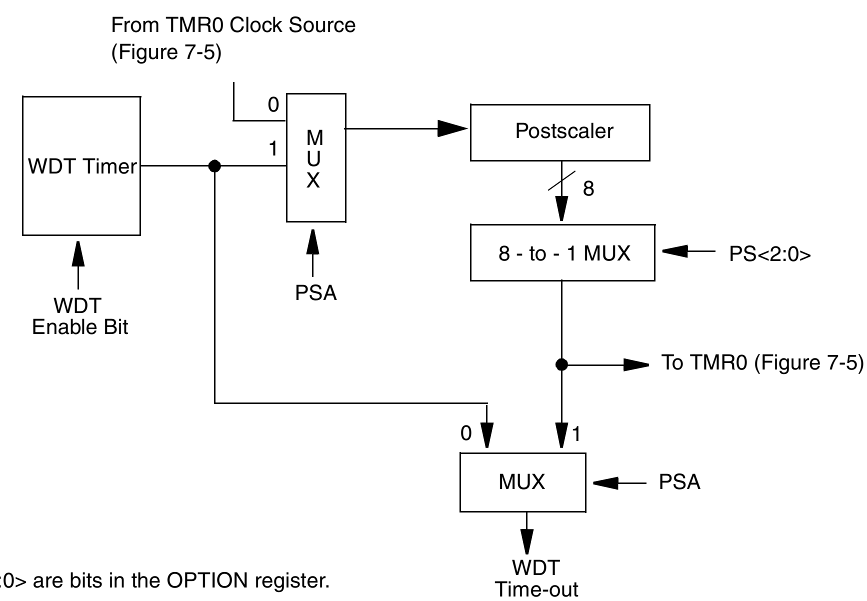
9.7.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken into account that under worst case conditions (V_{DD} = 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.

BCF — Bit Clear f

Syntax: [*label*] BCF f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b ≤ 7

Operation: 0 → (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 ≤ f ≤ 127
0 ≤ b ≤ 7

Operation: 1 → (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 ≤ f ≤ 127
0 ≤ b ≤ 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`

SUBLW Subtract W from Literal

Syntax:	[*label*] SUBLW k
Operands:	$0 \le k \le 255$
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:	$0 \le f \le 127$ $d \in [0,1]$
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

NOTES:

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.

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 ≤ 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**
	Input Low Voltage						
	I/O ports	V_{IL}					
D030	with TTL buffer		V_{SS}	—	0.8V	V	For 4.5V ≤ V_{DD} ≤ 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}	V	4.5V ≤ V_{DD} ≤ 5.5V
D040A			$0.25V_{DD}$ + 0.8V	—	V_{DD}	V	otherwise
D041	with Schmitt Trigger buffer		$0.8V_{DD}$	—	V_{DD}	V	For 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} ≤ V_{PIN} ≤ V_{DD}, Pin at hi-impedance
D061	GP3/$\overline{MCLR}$ (Note 5)				±30	μA	V_{SS} ≤ V_{PIN} ≤ V_{DD}
D061A	GP3 (Note 6)				±5	μA	V_{SS} ≤ V_{PIN} ≤ V_{DD}
D062	GP2/T0CKI		—	—	±5	μA	V_{SS} ≤ V_{PIN} ≤ V_{DD}
D063	OSC1		—	—	±5	μA	V_{SS} ≤ V_{PIN} ≤ 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 ≤ TA ≤ +70°C (commercial) –40°C ≤ TA ≤ +85°C (industrial) –40°C ≤ TA ≤ +125°C (extended) 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**
	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} = 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.

NOTES:

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: PIC16xxxxxx family

Literature 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?

____

____