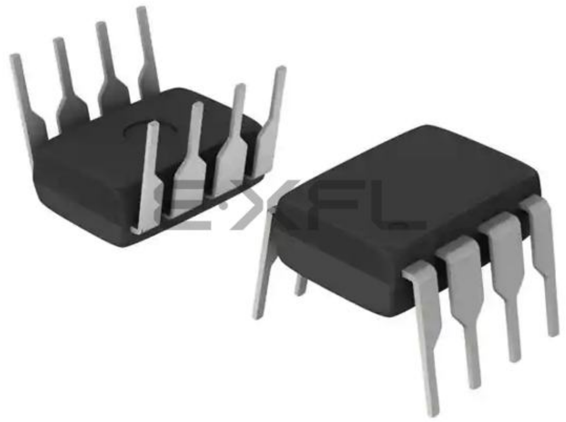


Welcome to E-XFL.COM

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	1.75KB (1K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 4x8b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°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/pic12c671-04-p

NOTES:

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		Mapped in Bank 0	F0h
7Fh			FFh
	Bank 0	Bank 1	

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

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.

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
$\overline{GPPU}$	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: **$\overline{GPPU}$:** 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

4.2.2.4 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

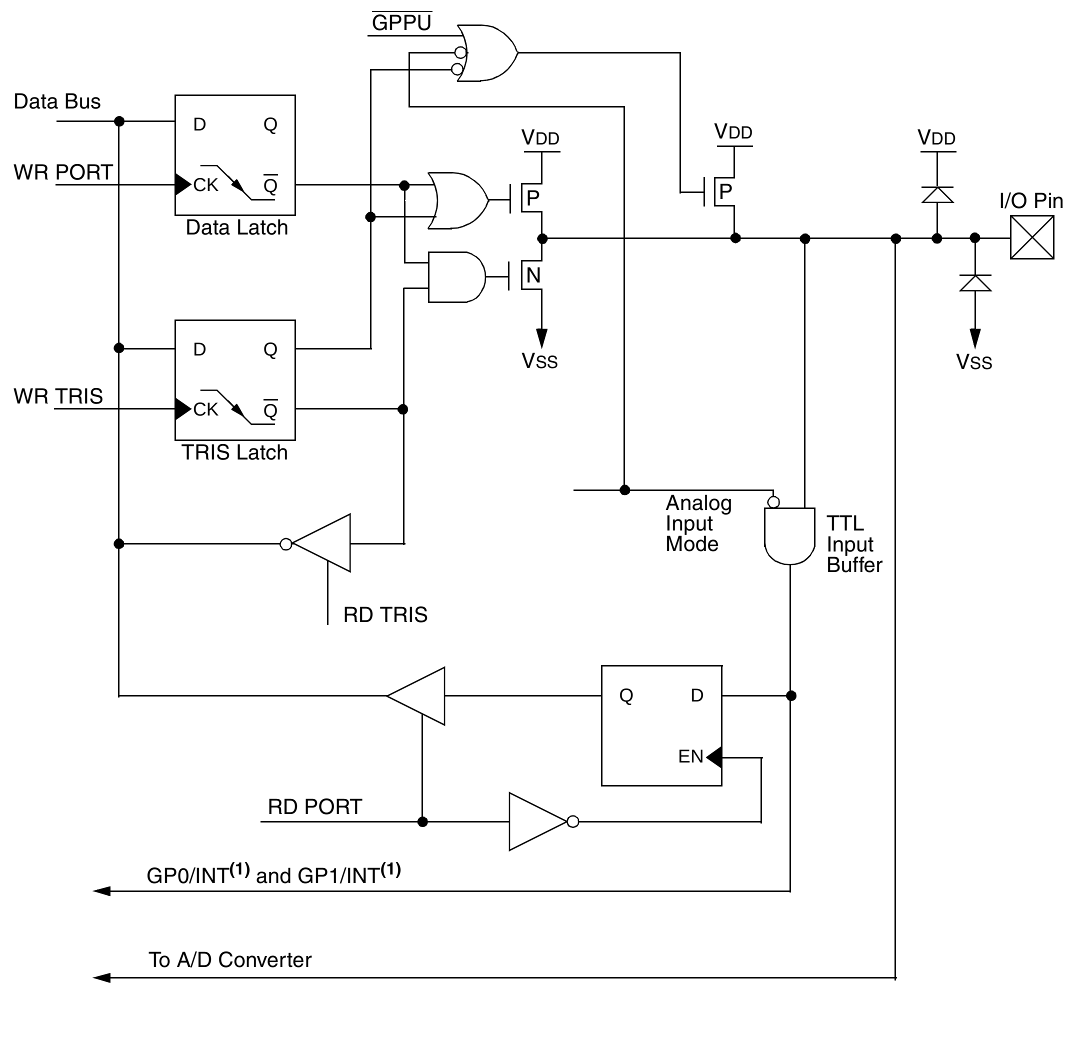
bit 7: **Unimplemented:** Read as '0'

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'

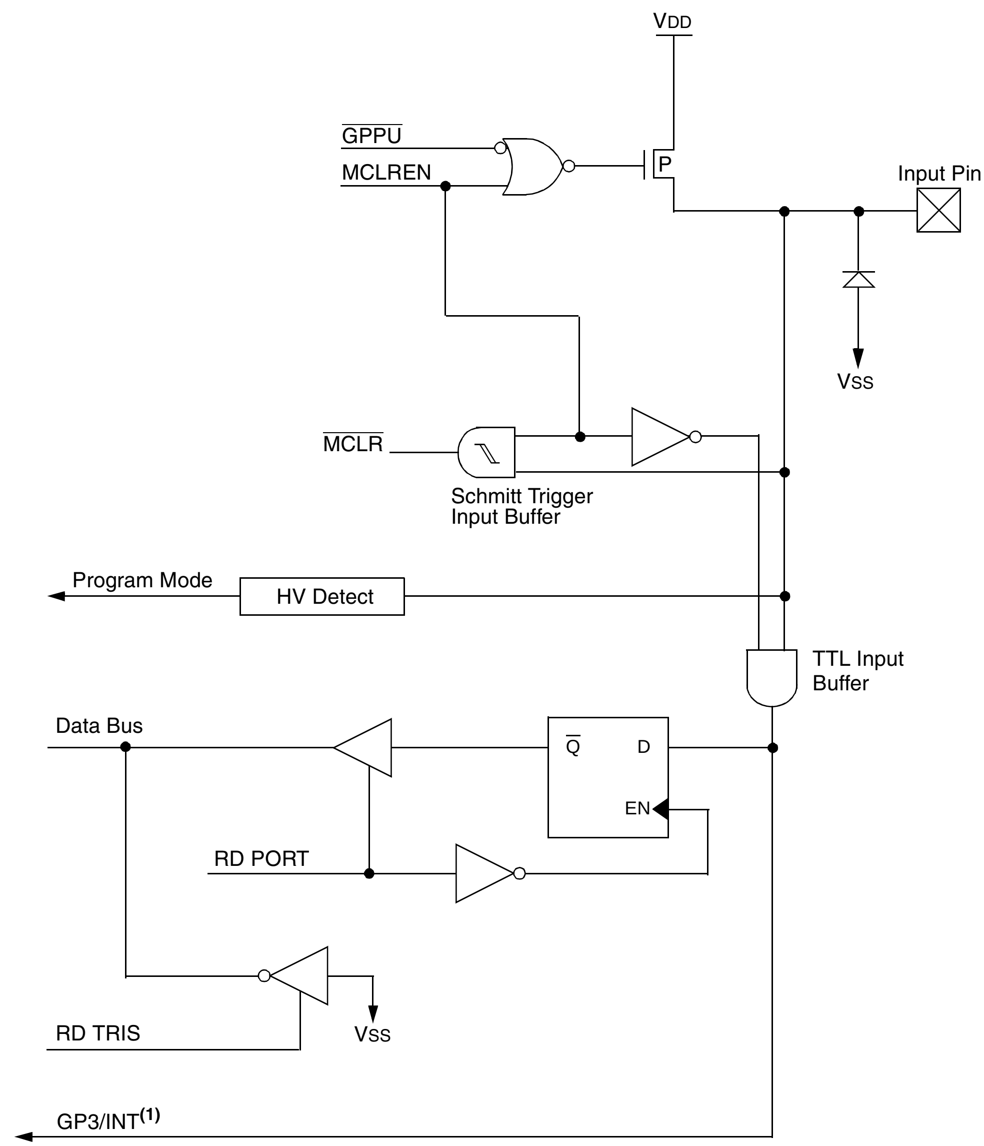
NOTES:

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



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

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

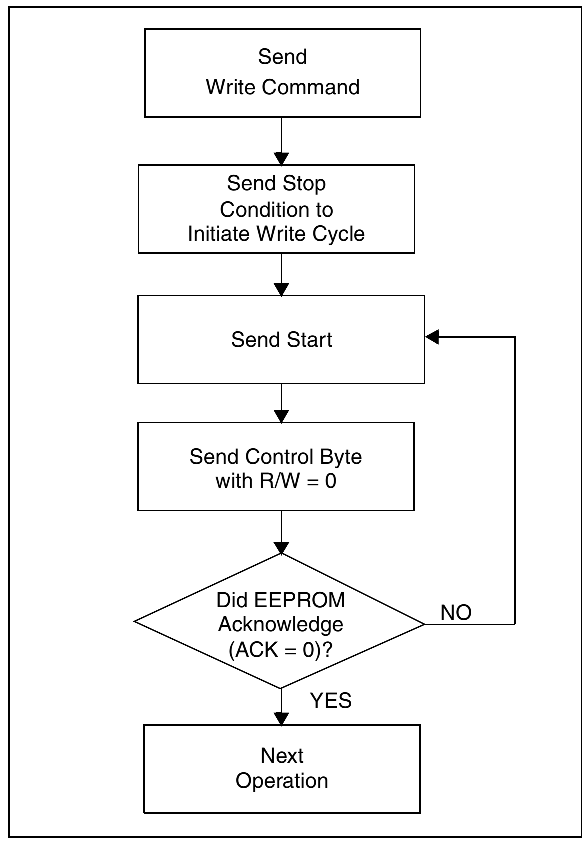
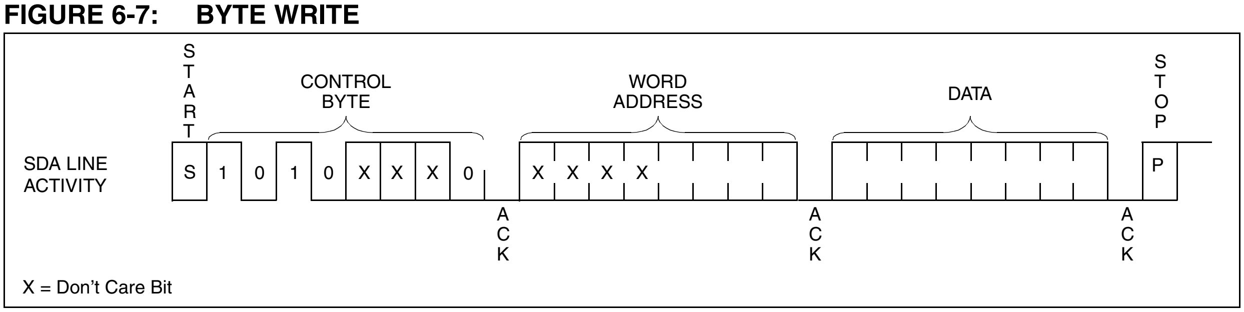


FIGURE 6-7: BYTE WRITE



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

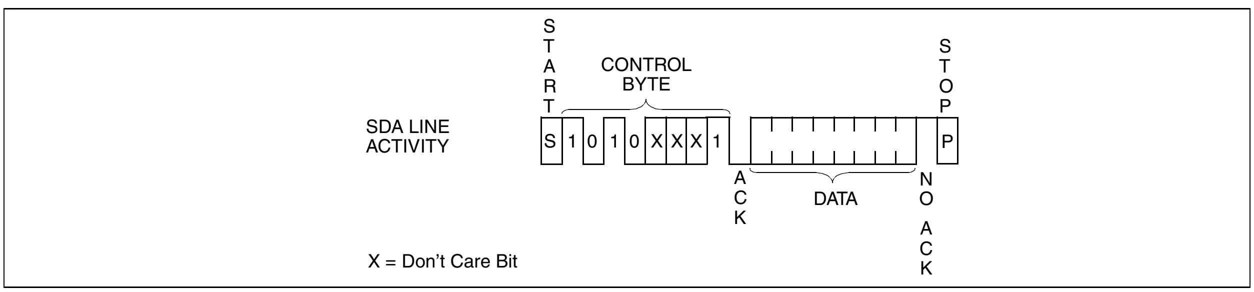


FIGURE 6-9: RANDOM READ

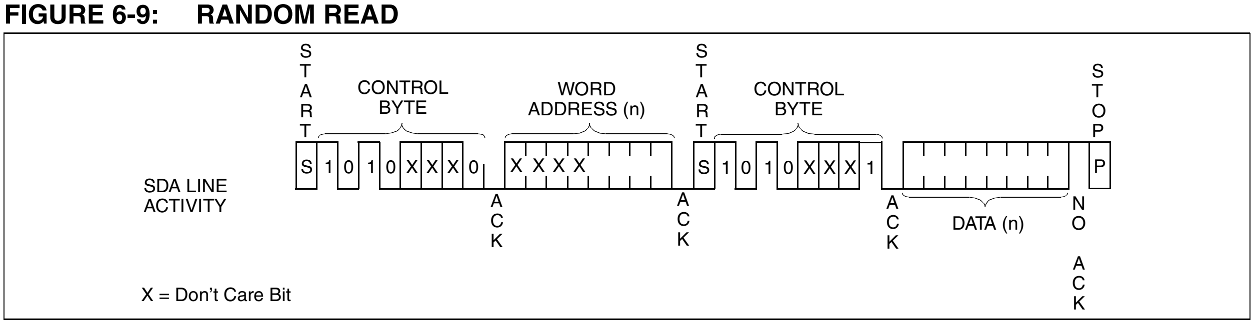
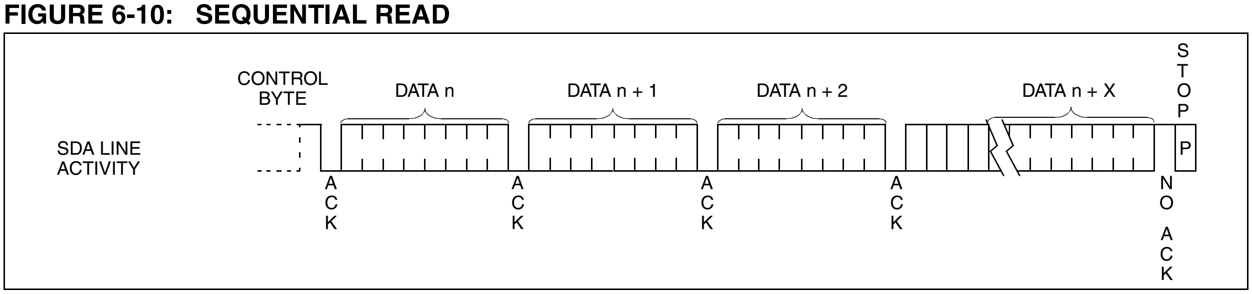


FIGURE 6-10: SEQUENTIAL READ



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.

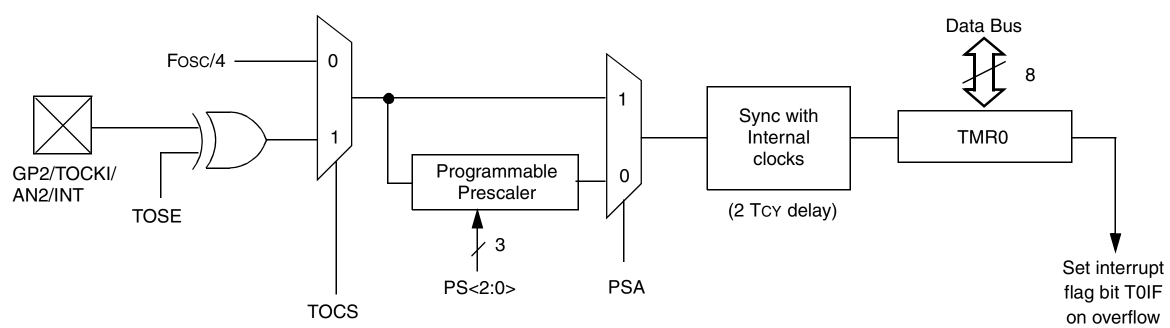
Counter mode is selected by setting bit T0CS (OPTION<5>). In counter mode, Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the bit T0SE (OPTION<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input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

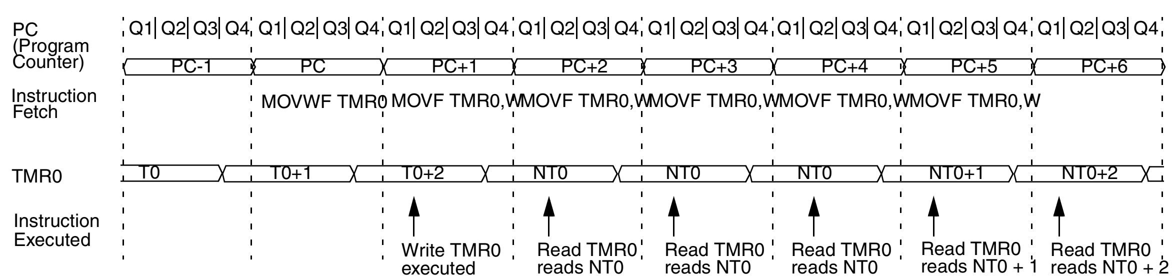
The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit T0IF (INTCON<2>). The interrupt can be masked by clearing bit T0IE (INTCON<5>). Bit T0IF must be cleared in software by the Timer0 module interrupt service routine before re-enabling this interrupt. 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



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 $\overline{MCLR}$ 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 $\overline{MCLR}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{MCLR}$ 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 $\overline{POR}$ (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
	$\overline{PWRTE}$ = 0	$\overline{PWRTE}$ = 1	
XT, HS, LP	72 ms + 1024T_{OSC}	1024T_{OSC}	1024T_{OSC}
INTRC, EXTRC	72 ms	—	—

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

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

Legend: u = unchanged, x = unknown.

10.2 Instruction Descriptions

ADDLW — Add Literal and W

Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (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 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) → (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 ≤ k ≤ 255
Operation:	(W) .AND. (k) → (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 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .AND. (f) → (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

GOTO **Unconditional Branch**

Syntax:	[*label*] GOTO k
Operands:	$0 \le k \le 2047$
Operation:	k → PC<10:0> PCLATH<4:3> → PC<12:11>
Status Affected:	None
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`

INCF **Increment f**

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

INCFSZ **Increment f, Skip if 0**

Syntax:	[*label*] INCFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
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	see below

```
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`

IORLW **Inclusive OR Literal with W**

Syntax:	[*label*] IORLW k
Operands:	$0 \le k \le 255$
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

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
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

12.3 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}	V	4.5V $\leq V_{DD} \leq$ 5.5V
D040A			$0.25V_{DD}$ + 0.8V	—	V_{DD}	V	otherwise
D041	with Schmitt Trigger buffer		$0.8V_{DD}$	—	V_{DD}	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	$V_{SS} \leq V_{PIN} \leq V_{DD}$, 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.

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	V	otherwise
D031	with Schmitt Trigger buffer		VSS	—	0.2VDD	V	
D032	$\overline{MCLR}$, 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	$\overline{MCLR}$, 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/$\overline{MCLR}$ (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
	$\overline{MCLR}$ 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 $\overline{MCLR}$ 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.

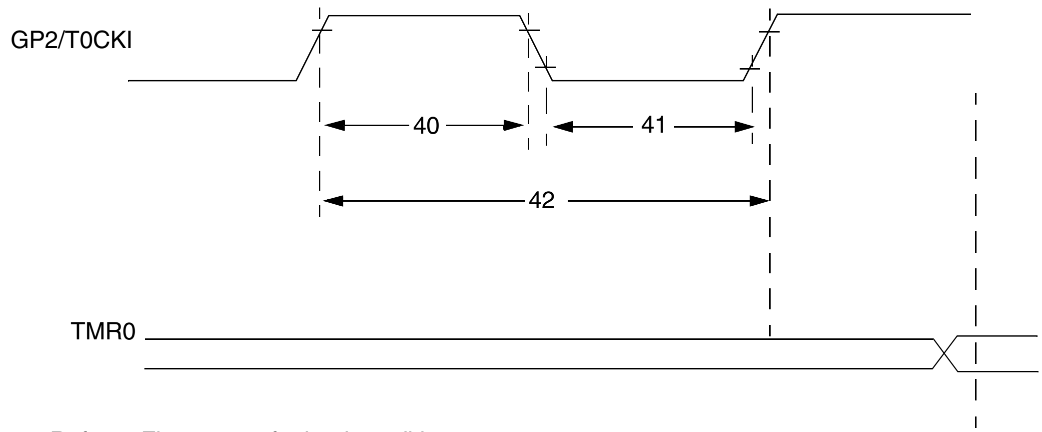
3: Negative current is defined as coming out of the pin.

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.

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	0.5T_{CY} + 20	—	—	ns	Must also meet parameter 42
			With Prescaler	10	—	—	ns	Must also meet parameter 42
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5T_{CY} + 20	—	—	ns	Must also meet parameter 42
			With Prescaler	10	—	—	ns	Must also meet parameter 42
42*	Tt0P	T0CKI Period	No Prescaler	T_{CY} + 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	—	7Tosc	—	

* 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.

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)

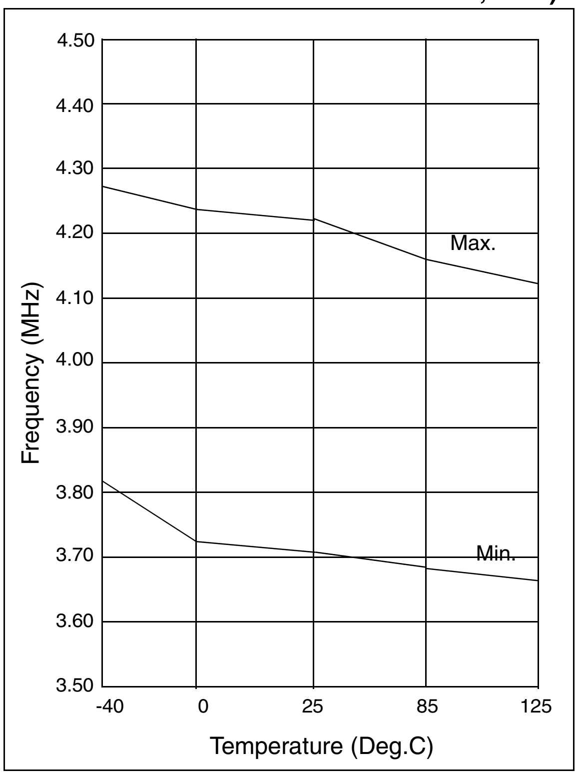


FIGURE 13-2: CALIBRATED INTERNAL RC FREQUENCY RANGE VS. TEMPERATURE (VDD = 2.5V) (INTERNAL RC IS CALIBRATED TO 25°C, 5.0V)

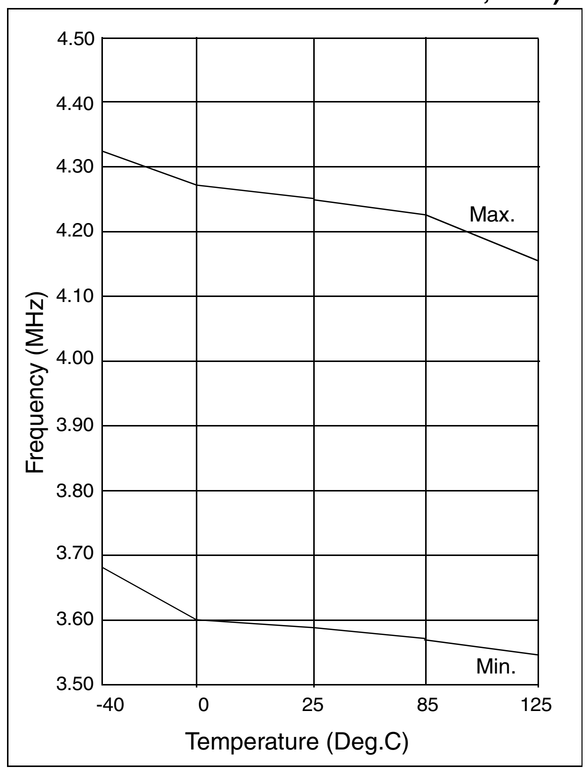
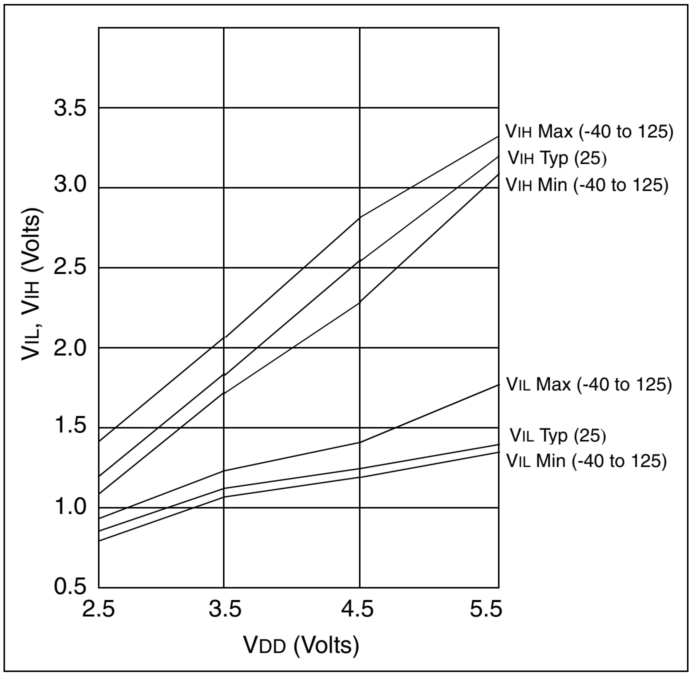


FIGURE 13-11: VIL, VIH OF NMCLR AND T0CKI vs. VDD



NOTES: