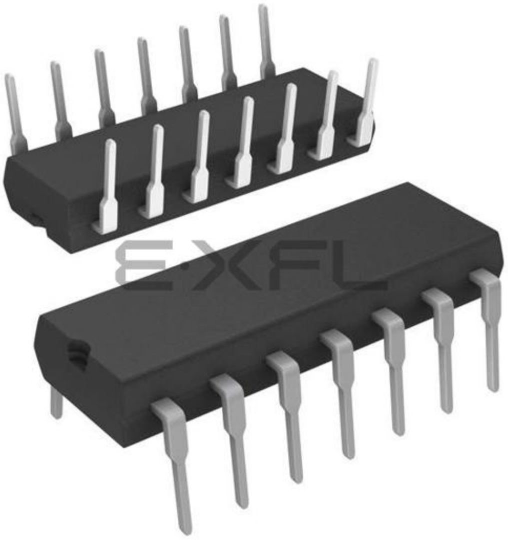


Welcome to E-XFL.COM

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	11
Program Memory Size	1.5KB (1K x 12)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	72 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	14-DIP (0.300", 7.62mm)
Supplier Device Package	14-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c505-04i-p

TABLE 1-1: PIC16C505 DEVICE

		PIC16C505
Clock	Maximum Frequency of Operation (MHz)	20
Memory	EPROM Program Memory	1024
	Data Memory (bytes)	72
Peripherals	Timer Module(s)	TMR0
	Wake-up from SLEEP on pin change	Yes
Features	I/O Pins	11
	Input Pins	1
	Internal Pull-ups	Yes
	In-Circuit Serial Programming	Yes
	Number of Instructions	33
	Packages	14-pin DIP, SOIC, TSSOP

The PIC16C505 device has Power-on Reset, selectable Watchdog Timer, selectable code protect, high I/O current capability and precision internal oscillator.
The PIC16C505 device uses serial programming with data pin RB0 and clock pin RB1.

3.1 Clocking Scheme/Instruction Cycle

The clock input (OSC1/CLKIN pin) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter is incremented every Q1, and the instruction is fetched from program memory and latched into the instruction register in Q4. It is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-2 and Example 3-1.

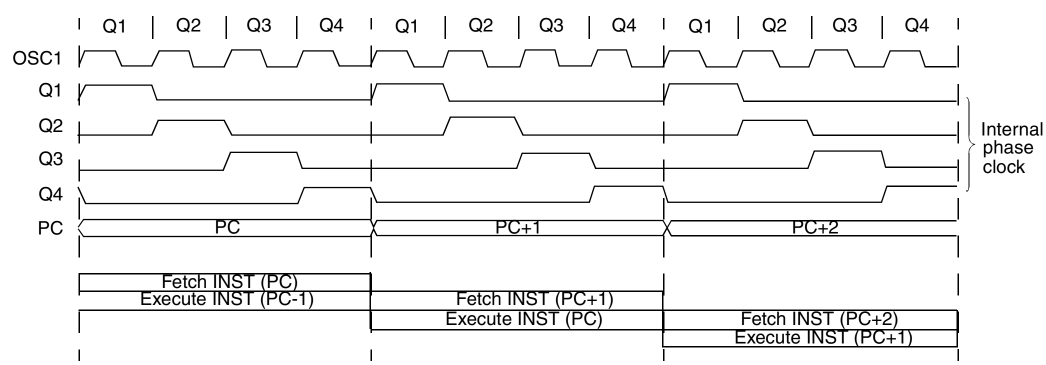
3.2 Instruction Flow/Pipelining

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 (e.g., `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

`1. MOVLW 03H`	Fetch 1	Execute 1				
`2. MOVWF PORTB`		Fetch 2	Execute 2			
`3. CALL SUB_1`			Fetch 3	Execute 3		
`4. BSF PORTB, BIT1`				Fetch 4	Flush	
					Fetch SUB_1	Execute SUB_1

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

4.0 MEMORY ORGANIZATION

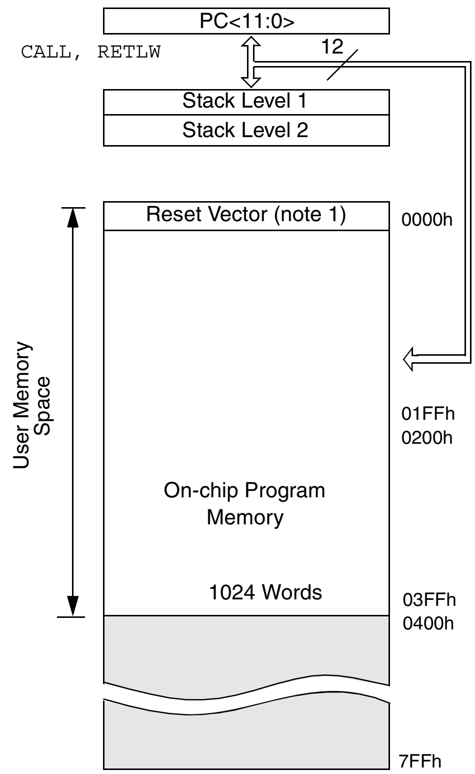
PIC16C505 memory is organized into program memory and data memory. For the PIC16C505, a paging scheme is used. Program memory pages are accessed using one STATUS register bit. Data memory banks are accessed using the File Select Register (FSR).

4.1 Program Memory Organization

The PIC16C505 devices have a 12-bit Program Counter (PC).

The 1K x 12 (0000h-03FFh) for the PIC16C505 are physically implemented. Refer to Figure 4-1. Accessing a location above this boundary will cause a wrap-around within the first 1K x 12 space. The effective reset vector is at 0000h, (see Figure 4-1). Location 03FFh contains the internal clock oscillator calibration value. This value should never be overwritten.

FIGURE 4-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16C505



Note 1: Address 0000h becomes the effective reset vector. Location 03FFh contains the `MOVLW XX` INTERNAL RC oscillator calibration value.

4.2 Data Memory Organization

Data memory is composed of registers or bytes of RAM. Therefore, data memory for a device is specified by its register file. The register file is divided into two functional groups: Special Function Registers and General Purpose Registers.

The Special Function Registers include the TMR0 register, the Program Counter (PCL), the Status Register, the I/O registers (ports) and the File Select Register (FSR). In addition, Special Function Registers are used to control the I/O port configuration and prescaler options.

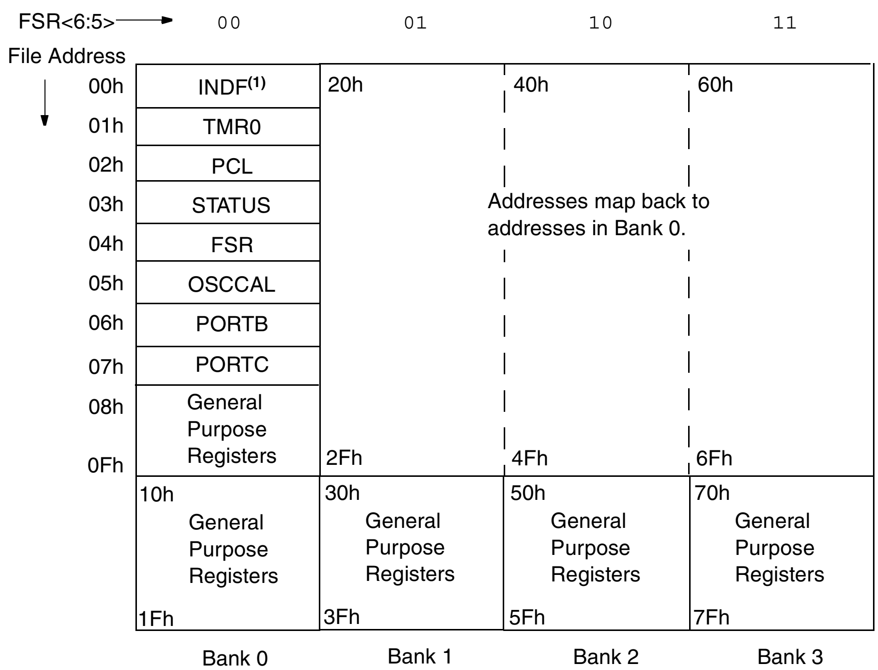
The General Purpose Registers are used for data and control information under command of the instructions.

For the PIC16C505, the register file is composed of 8 Special Function Registers, 24 General Purpose Registers and 48 General Purpose Registers that may be addressed using a banking scheme (Figure 4-2).

4.2.1 GENERAL PURPOSE REGISTER FILE

The General Purpose Register file is accessed, either directly or indirectly, through the File Select Register FSR (Section 4.8).

FIGURE 4-2: PIC16C505 REGISTER FILE MAP



Note 1: Not a physical register.

4.8 Indirect Data Addressing; INDF and FSR Registers

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a *pointer*). This is indirect addressing.

EXAMPLE 4-1: INDIRECT ADDRESSING

- Register file 07 contains the value 10h
- Register file 08 contains the value 0Ah
- Load the value 07 into the FSR register
- A read of the INDF register will return the value of 10h
- Increment the value of the FSR register by one (FSR = 08)
- A read of the INDR register now will return the value of 0Ah.

Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no-operation (although STATUS bits may be affected).

A simple program to clear RAM locations 10h-1Fh using indirect addressing is shown in Example 4-2.

EXAMPLE 4-2: HOW TO CLEAR RAM USING INDIRECT ADDRESSING

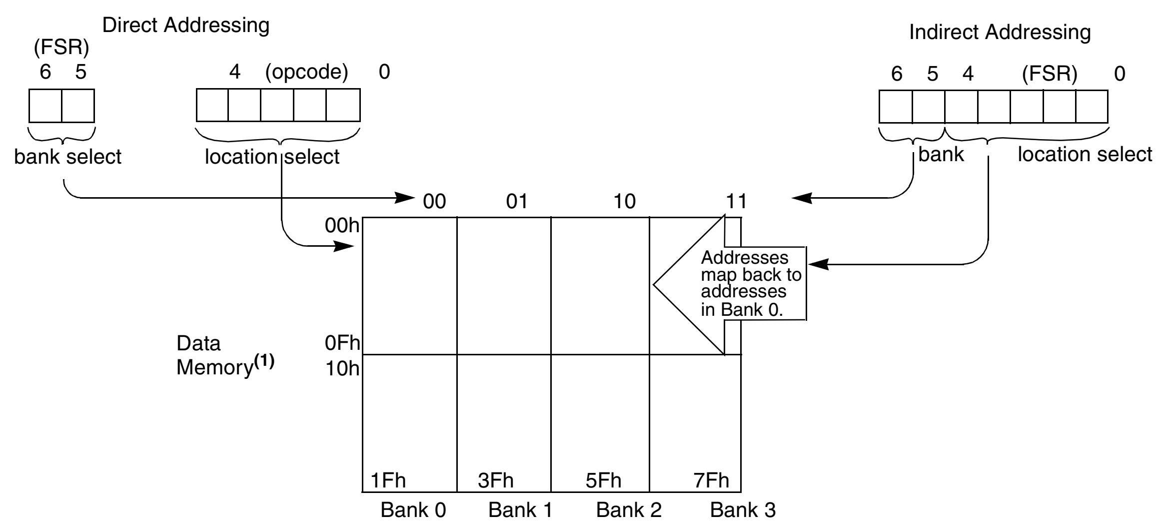
```
            movlw   0x10    ;initialize pointer
            movwf   FSR     ; to RAM
NEXT        clrf    INDF    ;clear INDF register
            incf    FSR,F   ;inc pointer
            btfsc   FSR,4   ;all done?
            goto    NEXT    ;NO, clear next
CONTINUE
            :               ;YES, continue
            :
```

The FSR is a 5-bit wide register. It is used in conjunction with the INDF register to indirectly address the data memory area.

The FSR<4:0> bits are used to select data memory addresses 00h to 1Fh.

The device uses FSR<6:5> to select between banks 0:3.

FIGURE 4-4: DIRECT/INDIRECT ADDRESSING



Note 1: For register map detail see Section 4.2.

5.0 I/O PORT

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

5.1 PORTB

PORTB is an 8-bit I/O register. Only the low order 6 bits are used (RB<5:0>). Bits 7 and 6 are unimplemented and read as '0's. Please note that RB3 is an input only pin. The configuration word can set several I/O's to alternate functions. When acting as alternate functions, the pins will read as '0' during port read. Pins RB0, RB1, RB3 and RB4 can be configured with weak pull-ups and also with wake-up on change. The wake-up on change and weak pull-up functions are not pin selectable. If pin 4 is configured as $\overline{\text{MCLR}}$, weak pull-up is always off and wake-up on change for this pin is not enabled.

5.2 PORTC

PORTC is an 8-bit I/O register. Only the low order 6 bits are used (RC<5:0>). Bits 7 and 6 are unimplemented and read as '0's.

5.3 TRIS Registers

The output driver control register is loaded with the contents of the W register by executing the `TRIS f` instruction. A '1' from a TRIS register bit puts the corresponding output driver in a hi-impedance mode. A '0' puts the contents of the output data latch on the selected pins, enabling the output buffer. The exceptions are RB3, which is input only, and RC5, which may be controlled by the option register. See Register 4-2.

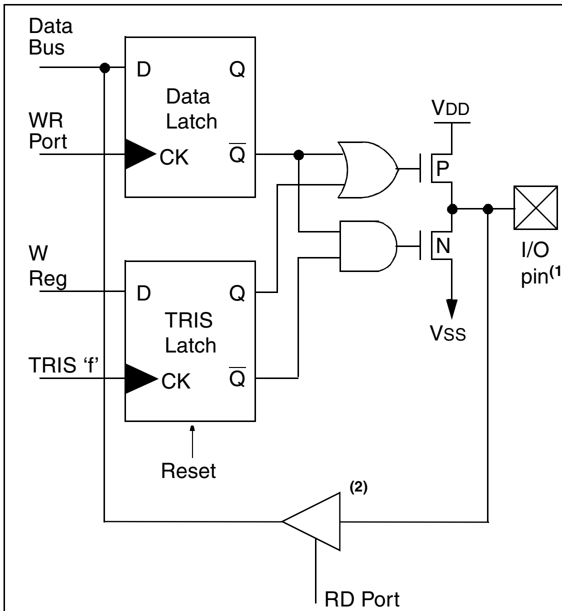
Note:	A read of the ports reads the pins, not the output data latches. That is, if an output driver on a pin is enabled and driven high, but the external system is holding it low, a read of the port will indicate that the pin is low.

The TRIS registers are "write-only" and are set (output drivers disabled) upon RESET.

5.4 I/O Interfacing

The equivalent circuit for an I/O port pin is shown in Figure 5-1. All port pins except RB3, which is input only, may be used for both input and output operations. For input operations, these ports are non-latching. Any input must be present until read by an input instruction (e.g., `MOVF PORTB,W`). The outputs are latched and remain unchanged until the output latch is rewritten. To use a port pin as output, the corresponding direction control bit in TRIS must be cleared (= 0). For use as an input, the corresponding TRIS bit must be set. Any I/O pin (except RB3) can be programmed individually as input or output.

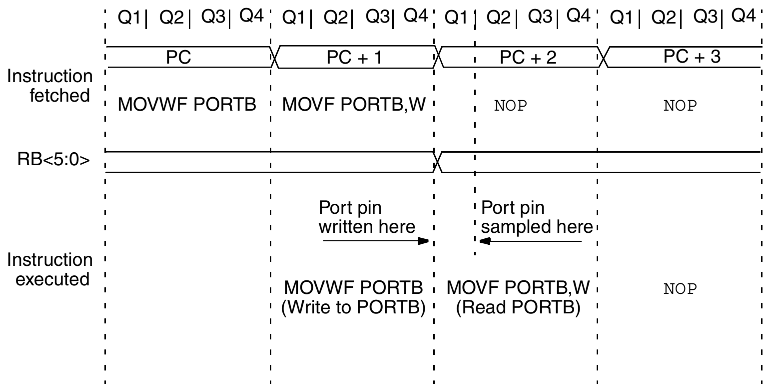
FIGURE 5-1: EQUIVALENT CIRCUIT FOR A SINGLE I/O PIN



Note 1: I/O pins have protection diodes to VDD and VSS.

Note 2: See Table 3-1 for buffer type.

FIGURE 5-2: SUCCESSIVE I/O OPERATION



This example shows a write to PORTB followed by a read from PORTB.

Data setup time = (0.25 T_{CY} – T_{PD})

where: T_{CY} = instruction cycle.

T_{PD} = propagation delay

Therefore, at higher clock frequencies, a write followed by a read may be problematic.

TABLE 7-3: RESET CONDITIONS FOR REGISTERS

Register	Address	Power-on Reset	$\overline{MCLR}$ Reset WDT time-out Wake-up on Pin Change
W	—	qqqq qqqq(1)	qqqq qqqq(1)
INDF	00h	xxxx xxxx	uuuu uuuu
TMR0	01h	xxxx xxxx	uuuu uuuu
PC	02h	1111 1111	1111 1111
STATUS	03h	0001 1xxx	q00q quuu(2,3)
FSR	04h	110x xxxx	11uu uuuu
OSCCAL	05h	1000 00--	uuuu uu--
PORTB	06h	--xx xxxxx	--uu uuuu
PORTC	07h	--xx xxxxx	--uu uuuu
OPTION	—	1111 1111	1111 1111
TRISB	—	--11 1111	--11 1111
TRISC	—	--11 1111	--11 1111

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as ‘0’, q = value depends on condition.

Note 1: Bits <7:2> of W register contain oscillator calibration values due to MOVLW XX instruction at top of memory.

Note 2: See Table 7-7 for reset value for specific conditions.

Note 3: If reset was due to wake-up on pin change, then bit 7 = 1. All other resets will cause bit 7 = 0.

TABLE 7-4: RESET CONDITION FOR SPECIAL REGISTERS

	STATUS Addr: 03h	PCL Addr: 02h
Power on reset	0001 1xxx	1111 1111
$\overline{MCLR}$ reset during normal operation	000u uuuu	1111 1111
$\overline{MCLR}$ reset during SLEEP	0001 0uuu	1111 1111
WDT reset during SLEEP	0000 0uuu	1111 1111
WDT reset normal operation	0000 uuuu	1111 1111
Wake-up from SLEEP on pin change	1001 0uuu	1111 1111

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as ‘0’.

FIGURE 7-7: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT

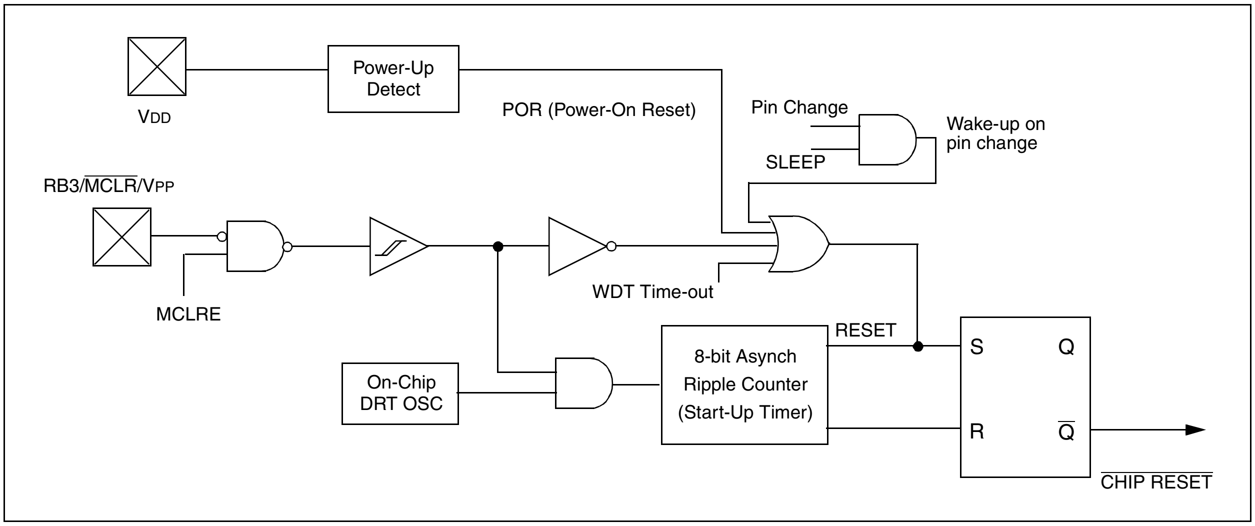


FIGURE 7-8: TIME-OUT SEQUENCE ON POWER-UP (MCLR PULLED LOW)

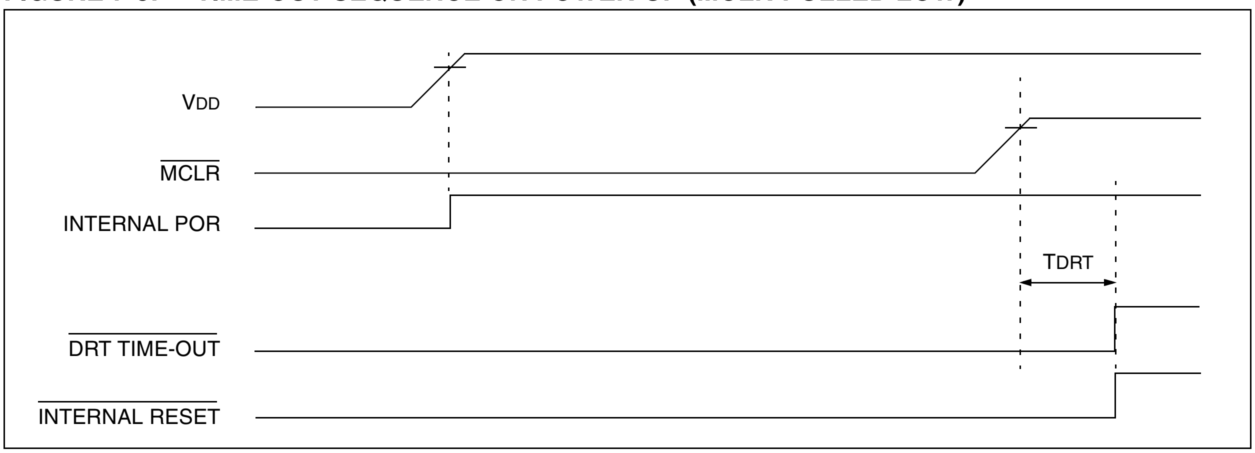


FIGURE 7-9: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD): FAST VDD RISE TIME

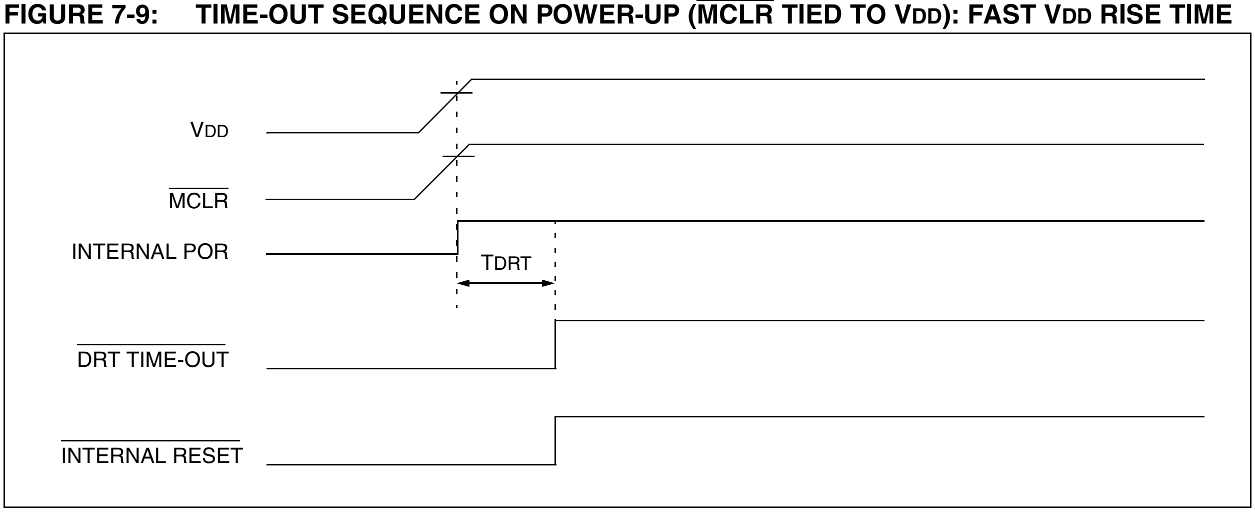
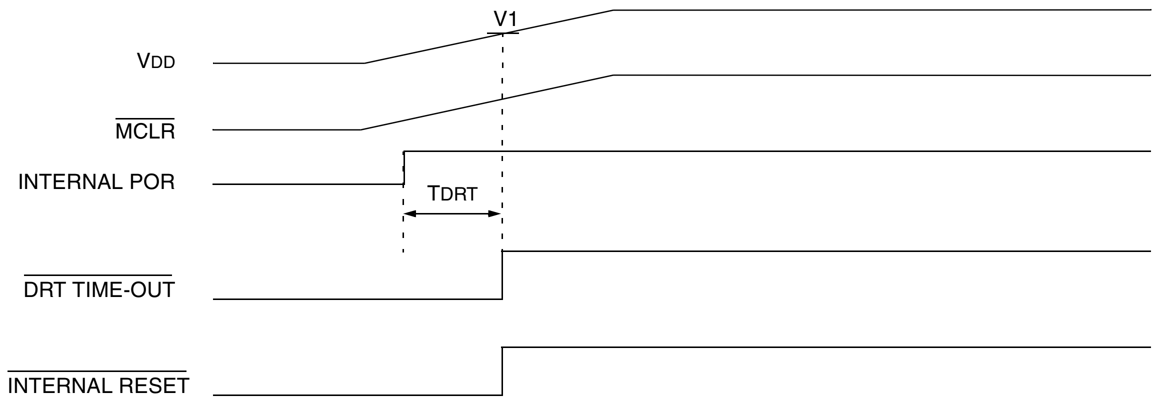


FIGURE 7-10: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO V_{DD}): SLOW V_{DD} RISE TIME



Note: When V_{DD} rises slowly, the T_{DRT} time-out expires long before V_{DD} has reached its final value. In this example, the chip will reset properly if, and only if, $V1 \geq V_{DD}$ min.

7.5 Device Reset Timer (DRT)

In the PIC16C505, the DRT runs any time the device is powered up. DRT runs from RESET and varies based on oscillator selection and reset type (see Table 7-5).

The DRT operates on an internal RC oscillator. The processor is kept in RESET as long as the DRT is active. The DRT delay allows V_{DD} to rise above V_{DD} min. and for the oscillator to stabilize.

Oscillator circuits based on crystals or ceramic resonators require a certain time after power-up to establish a stable oscillation. The on-chip DRT keeps the device in a RESET condition for approximately 18 ms after $\overline{MCLR}$ has reached a logic high ($V_{IH}\overline{MCLR}$) level. Thus, programming RB3/$\overline{MCLR}$/V_{PP} as $\overline{MCLR}$ and using an external RC network connected to the $\overline{MCLR}$ input is not required in most cases, allowing for savings in cost-sensitive and/or space restricted applications, as well as allowing the use of the RB3/$\overline{MCLR}$/V_{PP} pin as a general purpose input.

The Device Reset time delay will vary from chip to chip due to V_{DD}, temperature and process variation. See AC parameters for details.

The DRT will also be triggered upon a Watchdog Timer time-out. This is particularly important for applications using the WDT to wake from SLEEP mode automatically.

Reset sources are POR, $\overline{MCLR}$, WDT time-out and Wake-up on pin change. (See Section 7.9.2, Notes 1, 2, and 3, page 37.)

7.6 Watchdog Timer (WDT)

The Watchdog Timer (WDT) is a free running on-chip RC oscillator, which does not require any external components. This RC oscillator is separate from the external RC oscillator of the RB5/OSC1/CLKIN pin and the internal 4 MHz oscillator. That means that the WDT will run even if the main processor clock has been stopped, for example, by execution of a `SLEEP` instruction. During normal operation or SLEEP, a WDT reset or wake-up reset generates a device RESET.

The $\overline{TO}$ bit (STATUS<4>) will be cleared upon a Watchdog Timer reset.

The WDT can be permanently disabled by programming the configuration bit WDTE as a '0' (Section 7.1). Refer to the PIC16C505 Programming Specifications to determine how to access the configuration word.

TABLE 7-5: DRT (DEVICE RESET TIMER PERIOD)

Oscillator Configuration	POR Reset	Subsequent Resets
IntRC & ExtRC	18 ms (typical)	300 μs (typical)
HS, XT & LP	18 ms (typical)	18 ms (typical)

8.0 INSTRUCTION SET SUMMARY

Each PIC16C505 instruction is a 12-bit word divided into an OPCODE, which specifies the instruction type, and one or more operands which further specify the operation of the instruction. The PIC16C505 instruction set summary in Table 8-2 groups the instructions into byte-oriented, bit-oriented, and literal and control operations. Table 8-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator is used to specify which one of the 32 file registers is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in the file register specified in the instruction.

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

For **literal and control** operations, 'k' represents an 8 or 9-bit constant or literal value.

TABLE 8-1: OPCODE FIELD DESCRIPTIONS

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

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

Figure 8-1 shows the three general formats that the instructions can have. All examples in the figure use the following format to represent a hexadecimal number:

0xhhh

where 'h' signifies a hexadecimal digit.

FIGURE 8-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations

11 — 6	5	4 — 0
OPCODE	d	f (FILE #)

d = 0 for destination W
d = 1 for destination f
f = 5-bit file register address

Bit-oriented file register operations

11 — 8	7 — 5	4 — 0
OPCODE	b (BIT #)	f (FILE #)

b = 3-bit bit address
f = 5-bit file register address

Literal and control operations (except `GOTO`)

11 — 8	7 — 0
OPCODE	k (literal)

k = 8-bit immediate value

Literal and control operations - `GOTO` instruction

11 — 9	8 — 0
OPCODE	k (literal)

k = 9-bit immediate value

CALL — Subroutine Call

Syntax: [*label*] CALL k

Operands: $0 \leq k \leq 255$

Operation: (PC) + 1→ Top of Stack;
k → PC<7:0>;
(STATUS<6:5>) → PC<10:9>;
0 → PC<8>

Status Affected: None

Encoding:

1001	kkkk	kkkk

Description: Subroutine call. First, return address (PC+1) is pushed onto the stack. The eight bit immediate address is loaded into PC bits <7:0>. The upper bits PC<10:9> are loaded from STATUS<6:5>, PC<8> is cleared. `CALL` is a two cycle instruction.

Words: 1

Cycles: 2

Example: `HERE CALL THERE`

Before Instruction
PC = address (HERE)

After Instruction
PC = address (THERE)
TOS = address (HERE + 1)

CLRF — Clear f

Syntax: [*label*] CLRF f

Operands: $0 \leq f \leq 31$

Operation: 00h → (f);
1 → Z

Status Affected: Z

Encoding:

0000	011f	ffff

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

Words: 1

Cycles: 1

Example: `CLRF FLAG_REG`

Before Instruction
FLAG_REG = 0x5A

After Instruction
FLAG_REG = 0x00
Z = 1

CLRW — Clear W

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W);
1 → Z

Status Affected: Z

Encoding:

0000	0100	0000

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

Words: 1

Cycles: 1

Example: `CLRW`

Before Instruction
W = 0x5A

After Instruction
W = 0x00
Z = 1

CLRWDT — Clear Watchdog Timer

Syntax: [*label*] CLRWDT

Operands: None

Operation: 00h → WDT;
0 → WDT prescaler (if assigned);
1 → $\overline{TO}$;
1 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

0000	0000	0100

Description: The `CLRWDT` instruction resets the WDT. It also resets the prescaler, if the prescaler is assigned to the WDT and not Timer0. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

Words: 1

Cycles: 1

Example: `CLRWDT`

Before Instruction
WDT counter = ?

After Instruction
WDT counter = 0x00
WDT prescale = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

SWAPF **Swap Nibbles in f**

Syntax:	[*label*] SWAPF f,d
Operands:	0 ≤ f ≤ 31 d ∈ [0,1]
Operation:	(f<3:0>) → (dest<7:4>); (f<7:4>) → (dest<3:0>)
Status Affected:	None
Encoding:	`0011` `10df` `ffff`
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in W register. If 'd' is 1, the result is placed in register 'f'.
Words:	1
Cycles:	1
Example	`SWAPF REG1, 0`

Before Instruction
REG1 = 0xA5

After Instruction
REG1 = 0xA5
W = 0X5A

TRIS **Load TRIS Register**

Syntax:	[*label*] TRIS f
Operands:	f = 6
Operation:	(W) → TRIS register f
Status Affected:	None
Encoding:	`0000` `0000` `0fff`
Description:	TRIS register 'f' (f = 6 or 7) is loaded with the contents of the W register
Words:	1
Cycles:	1
Example	`TRIS PORTB`

Before Instruction
W = 0XA5

After Instruction
TRIS = 0XA5

XORLW **Exclusive OR literal with W**

Syntax:	[*label*] XORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .XOR. k → (W)
Status Affected:	Z
Encoding:	`1111` `kkkk` `kkkk`
Description:	The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.
Words:	1
Cycles:	1
Example:	`XORLW 0xAF`

Before Instruction
W = 0xB5

After Instruction
W = 0x1A

XORWF **Exclusive OR W with f**

Syntax:	[*label*] XORWF f,d
Operands:	0 ≤ f ≤ 31 d ∈ [0,1]
Operation:	(W) .XOR. (f) → (dest)
Status Affected:	Z
Encoding:	`0001` `10df` `ffff`
Description:	Exclusive OR the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example	`XORWF REG,1`

Before Instruction
REG = 0xAF
W = 0xB5

After Instruction
REG = 0x1A
W = 0xB5

9.0 DEVELOPMENT SUPPORT

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

- Integrated Development Environment
  - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB C Compiler for Various Device Families
  - HI-TECH C® for Various Device Families
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- Device Programmers
  - PICkit™ 2 Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

9.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

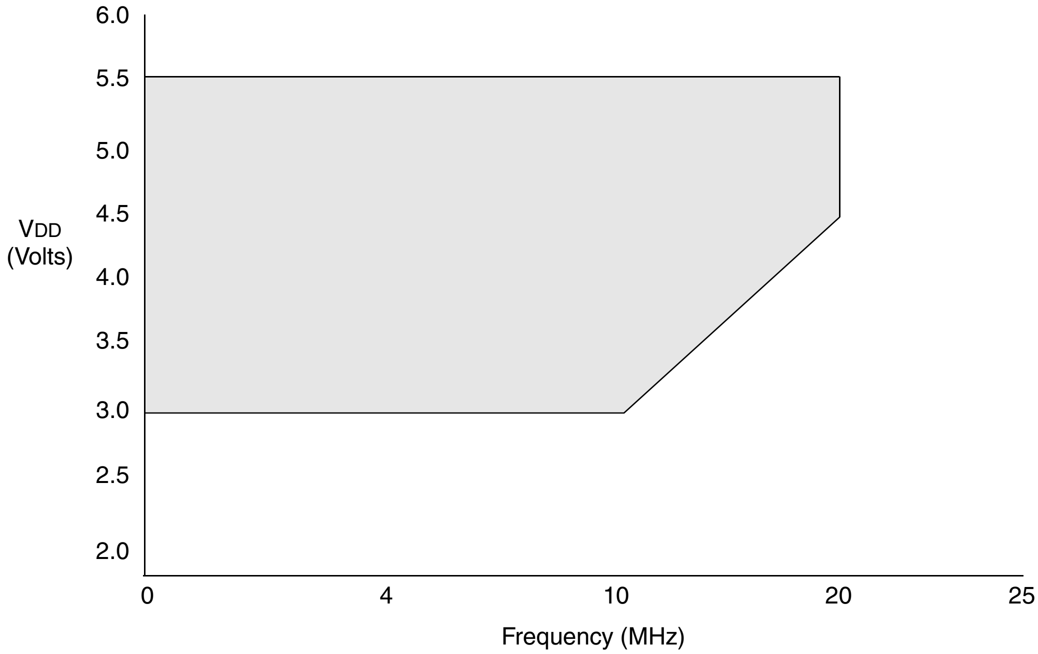
- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - In-Circuit Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (C or assembly)
  - Mixed C and assembly
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

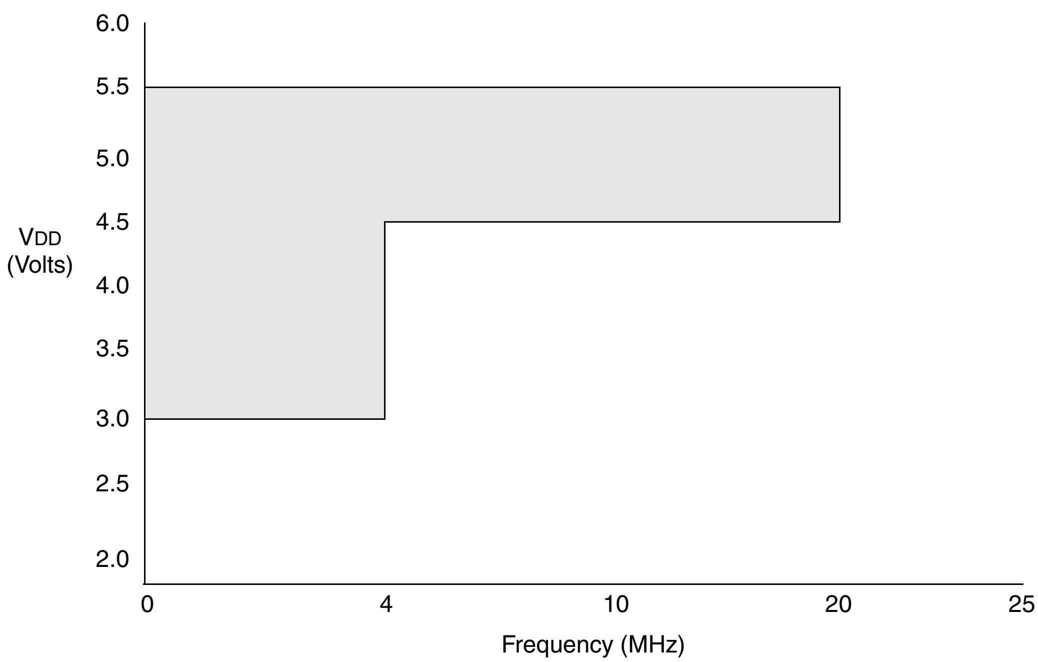
FIGURE 10-1: PIC16C505 VOLTAGE-FREQUENCY GRAPH, 0°C ≤ TA ≤ +70°C



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

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

FIGURE 10-2: PIC16C505 VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ 0°C, +70°C ≤ TA ≤ +125°C



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

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

10.1 DC CHARACTERISTICS: PIC16C505-04 (Commercial, Industrial, Extended) PIC16C505-20(Commercial, Industrial, Extended)

DC Characteristics Power Supply Pins							Standard Operating Conditions (unless otherwise specified) Operating Temperature $0°C \le T_A \le +70°C$ (commercial) $-40°C \le T_A \le +85°C$ (industrial) $-40°C \le T_A \le +125°C$ (extended)
Parm. No.	Characteristic	Sym	Min	Typ(1)	Max	Units	Conditions
D001	Supply Voltage	VDD	3.0		5.5	V	See Figure 10-1 through Figure 10-3
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	— — — — — —	0.8 0.6 3 4 4.5 19	1.4 1.0 7 12 16 27	mA mA mA mA mA μA	FOSC = 4MHz, VDD = 5.5V, WDT disabled (Note 4)* FOSC = 4MHz, VDD = 3.0V, WDT disabled (Note 4) FOSC = 10MHz, VDD = 3.0V, WDT disabled (Note 6) FOSC = 20MHz, VDD = 4.5V, WDT disabled FOSC = 20MHz, VDD = 5.5V, WDT disabled* FOSC = 32kHz, VDD = 3.0V, WDT disabled (Note 6)
D020	Power-Down Current (5)	IPD	— — — —	0.25 0.4 3 5	4 5.5 8 14	μA μA μA μA	VDD = 3.0V (Note 6) VDD = 4.5V* (Note 6) VDD = 5.5V, Industrial VDD = 5.5V, Extended Temp.
D022	WDT Current(5)	ΔIWDT	—	2.2	5	μA	VDD = 3.0V (Note 6)
1A	LP Oscillator Operating Frequency RC Oscillator Operating Frequency XT Oscillator Operating Frequency HS Oscillator Operating Frequency	Fosc	0 0 0 0	— — — —	200 4 4 20	kHz MHz MHz MHz	All temperatures All temperatures All temperatures All temperatures

* These parameters are characterized but not tested.

Note 1: Data in the Typical ("Typ") column is based on characterization results at 25°C. This data is for design guidance only and is not tested.
2: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.
3: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern and temperature also have an impact on the current consumption.
a) The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tristated, pulled to VSS, T0CKI = VDD, $\overline{\text{MCLR}}$ = VDD;
WDT enabled/disabled as specified.
b) For standby current measurements, the conditions are the same, except that the device is in SLEEP mode.
4: Does not include current through Rext. 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: Commercial temperature range only.

FIGURE 10-8: TIMER0 CLOCK TIMINGS - PIC16C505

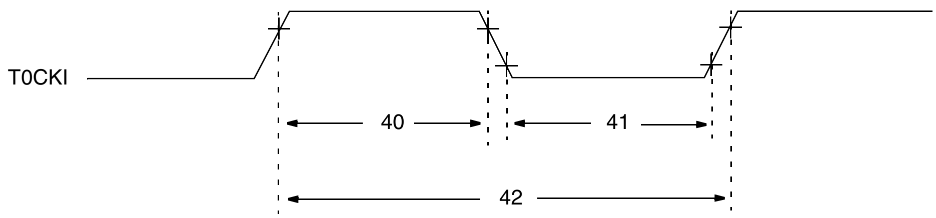


TABLE 10-7: TIMER0 CLOCK REQUIREMENTS - PIC16C505

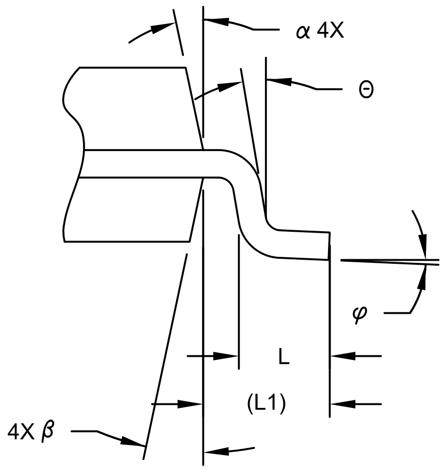
AC Characteristics				Standard Operating Conditions (unless otherwise specified) Operating Temperature $0°C \leq T_A \leq +70°C$ (commercial) $-40°C \leq T_A \leq +85°C$ (industrial) $-40°C \leq T_A \leq +125°C$ (extended) Operating Voltage V_{DD} range is described in Section 10.1.				
Parm No.	Sym	Characteristic		Min	Typ(1)	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 T_{CY} + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 T_{CY} + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		20 or T_{CY} + 40* N	—	—	ns	Whichever is greater. N = Prescale Value (1, 2, 4,..., 256)

* These parameters are characterized but not tested.

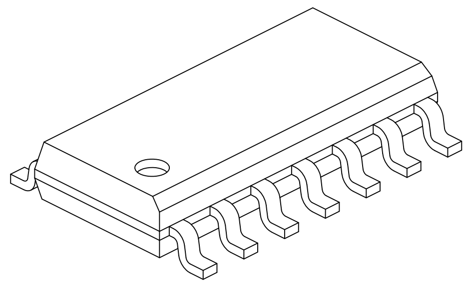
Note 1: Data in the Typical ("Typ") column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

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

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



VIEW C



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	1.75
Molded Package Thickness	A2	1.25	-	-
Standoff §	A1	0.10	-	0.25
Overall Width	E	6.00 BSC		
Molded Package Width	E1	3.90 BSC		
Overall Length	D	8.65 BSC		
Chamfer (Optional)	h	0.25	-	0.50
Foot Length	L	0.40	-	1.27
Footprint	L1	1.04 REF		
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.10	-	0.25
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

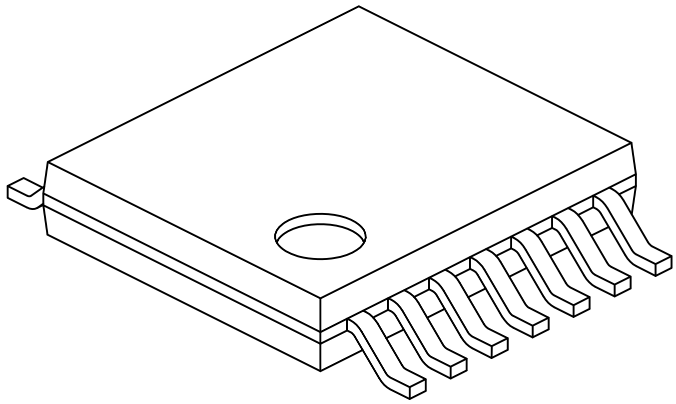
Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

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

14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

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



	Units	MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	0.65 BSC		
Overall Height	A	-	-	1.20
Molded Package Thickness	A2	0.80	1.00	1.05
Standoff	A1	0.05	-	0.15
Overall Width	E	6.40 BSC		
Molded Package Width	E1	4.30	4.40	4.50
Molded Package Length	D	4.90	5.00	5.10
Foot Length	L	0.45	0.60	0.75
Footprint	(L1)	1.00 REF		
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.09	-	0.20
Lead Width	b	0.19	-	0.30

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing No. C04-087C Sheet 2 of 2

W

Wake-up from SLEEP 37
Watchdog Timer (WDT) 27, 34
Period 35
Programming Considerations 35
WWW Address 85
WWW, On-Line Support 2

Z

Zero bit 7