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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	12
Program Memory Size	768B (512 x 12)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	25 x 8
Voltage - Supply (Vcc/Vdd)	3.5V ~ 15V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16hv540t-04i-ss

4.6 Program Counter

As a program instruction is executed, the Program Counter (PC) will contain the address of the next program instruction to be executed. The PC value is increased by one every instruction cycle, unless an instruction changes the PC.

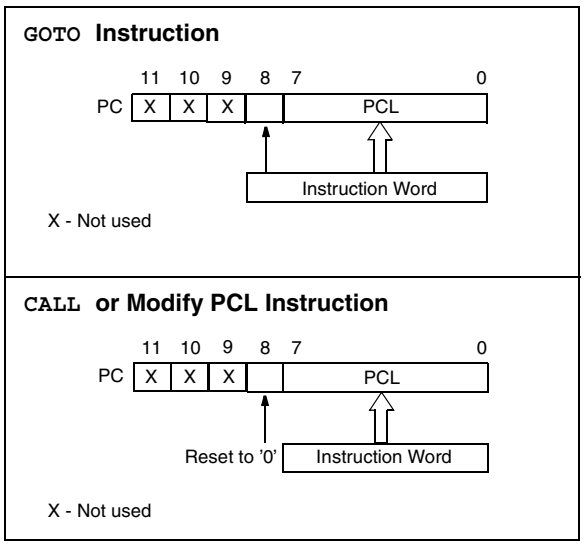
For a GOTO instruction, bits 8:0 of the PC are provided by the GOTO instruction word. (Figure 4-3).

For a CALL instruction, or any instruction where the PCL is the destination, bits 7:0 of the PC again are provided by the instruction word. However, PC<8> does not come from the instruction word, but is always cleared (Figure 4-3).

Instructions where the PCL is the destination, or Modify PCL instructions, include MOVWF PC, ADDWF PC, and BSF PC, 5. .

Note:	Because PC<8> is cleared in the CALL instruction, or any Modify PCL instruction, all subroutine calls or computed jumps are limited to the first 256 locations of any program memory page (512 words long).

FIGURE 4-3: LOADING OF PC BRANCH INSTRUCTIONS - PIC16HV540



4.6.1 EFFECTS OF RESET

The Program Counter is set upon a RESET, which means that the PC addresses the last location in the last page i.e., the reset vector.

The STATUS register page preselect bits are cleared upon a RESET, which means that page 0 is preselected.

Therefore, upon a RESET, a GOTO instruction at the reset vector location will automatically cause the program to jump to page 0.

4.7 Stack

PIC16HV540 device has a 12-bit wide L.I.F.O. (last in, first out) hardware 4 level stack.

A CALL instruction will *push* the current value of stack 1 into stack 2 and then push the current program counter value, incremented by one, into stack level 1. If more than four sequential CALL's are executed, only the most recent four return addresses are stored.

A RETLW instruction will *pop* the contents of stack level 1 into the program counter and then copy stack level 2 contents into level 1. If more than four sequential RETLW's are executed, the stack will be filled with the address previously stored in level 4. Note that the W register will be loaded with the literal value specified in the instruction. This is particularly useful for the implementation of data look-up tables within the program memory.

Upon any reset, the contents of the stack remain unchanged, however the program counter (PCL) will also be reset to 0.

Note 1:	There are no STATUS bits to indicate stack overflows or stack underflow conditions.
Note 2:	There are no instructions mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL and RETLW instructions.

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 4-3: INDIRECT ADDRESSING

- Register file 05 contains the value 10h
- Register file 06 contains the value 0Ah
- Load the value 05 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 = 06)
- 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 4-4.

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

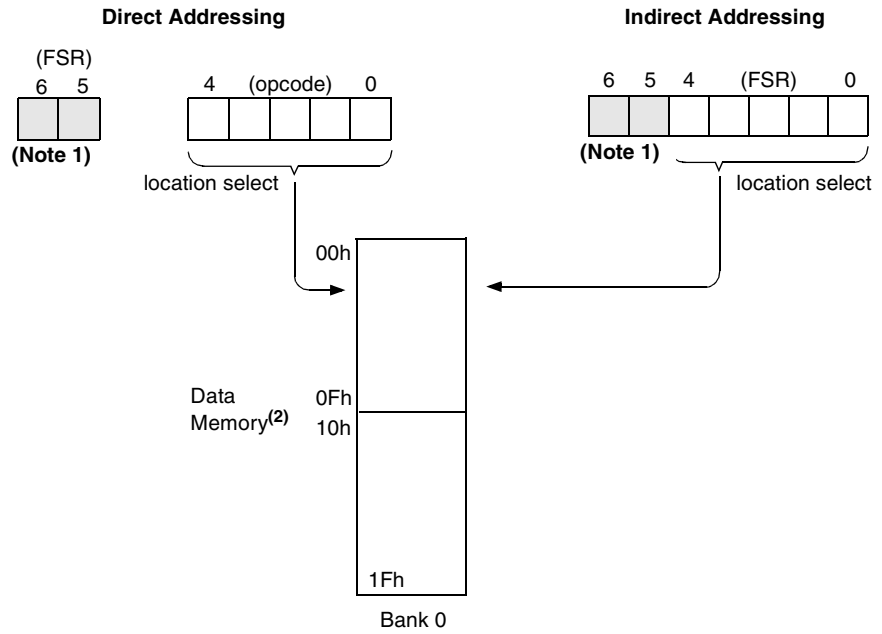
```
            movlw   0x10    ;initialize pointer
            movwf   FSR     ; to RAM
NEXT        clrf    INDF    ;clear INDF register
            incf    FSR,F   ;inc pointer
            btfsc   FSR,4   ;all done?
            goto    NEXT    ;NO, clear next
CONTINUE
            :               ;YES,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

The FSR is a 5-bit (PIC16HV540) 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.

PIC16HV540: Do not use banking. FSR<6:5> are unimplemented and read as '1's.

FIGURE 4-4: DIRECT/INDIRECT ADDRESSING



Note 1: Bits 5 and 6 are unimplemented and read as 1's.

2: For register map detail, see Section 4.2.

NOTES:

FIGURE 5-3: BLOCK DIAGRAM OF PORTB<4:6> PINS

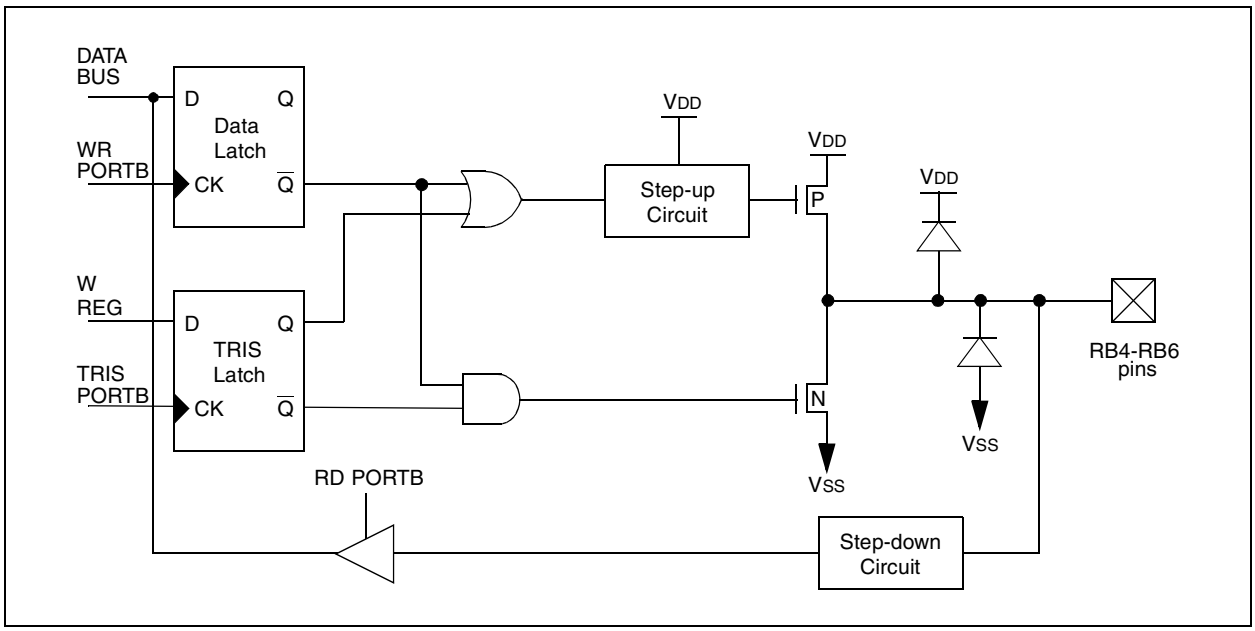


FIGURE 5-4: BLOCK DIAGRAM OF PORTB<7> PIN

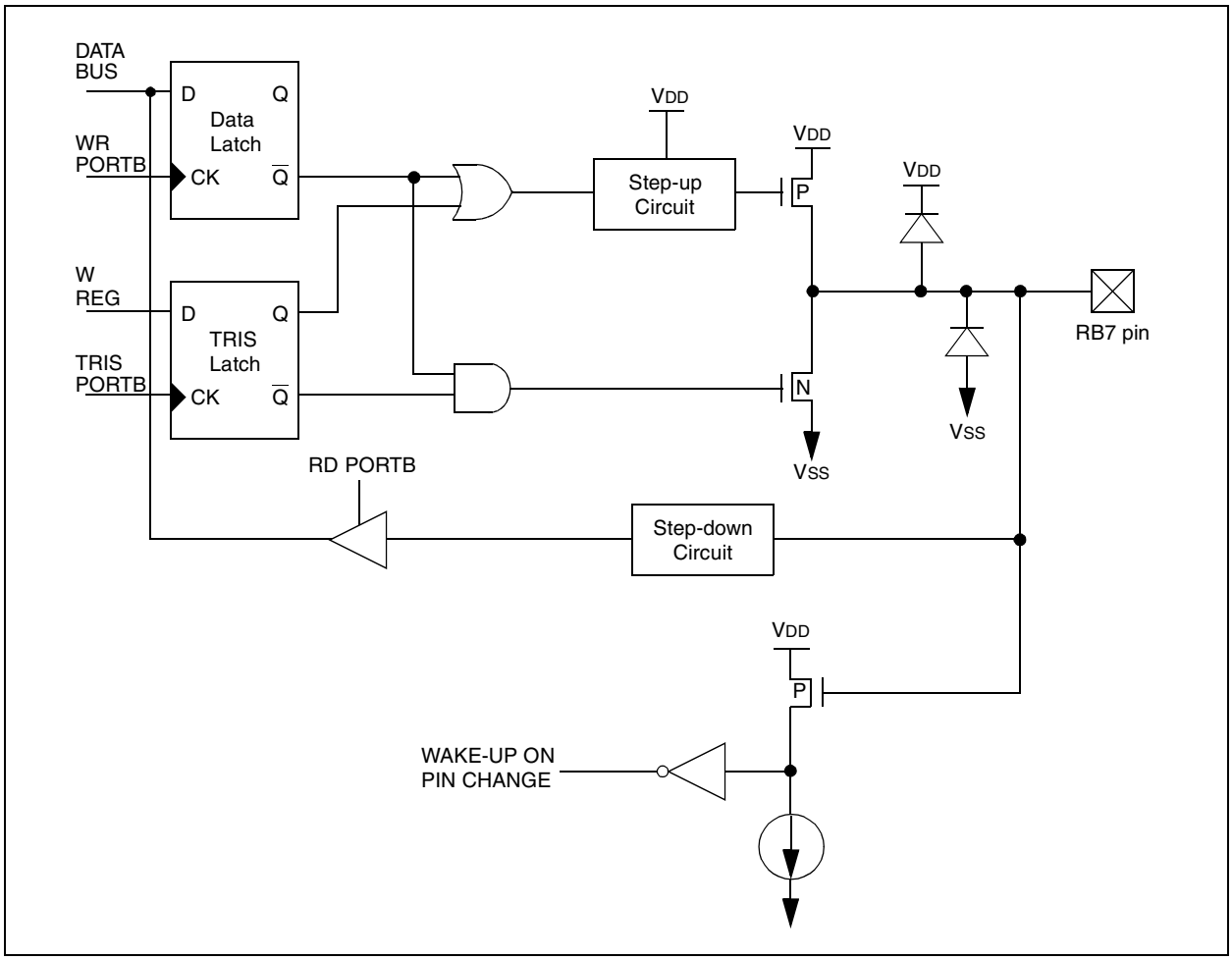
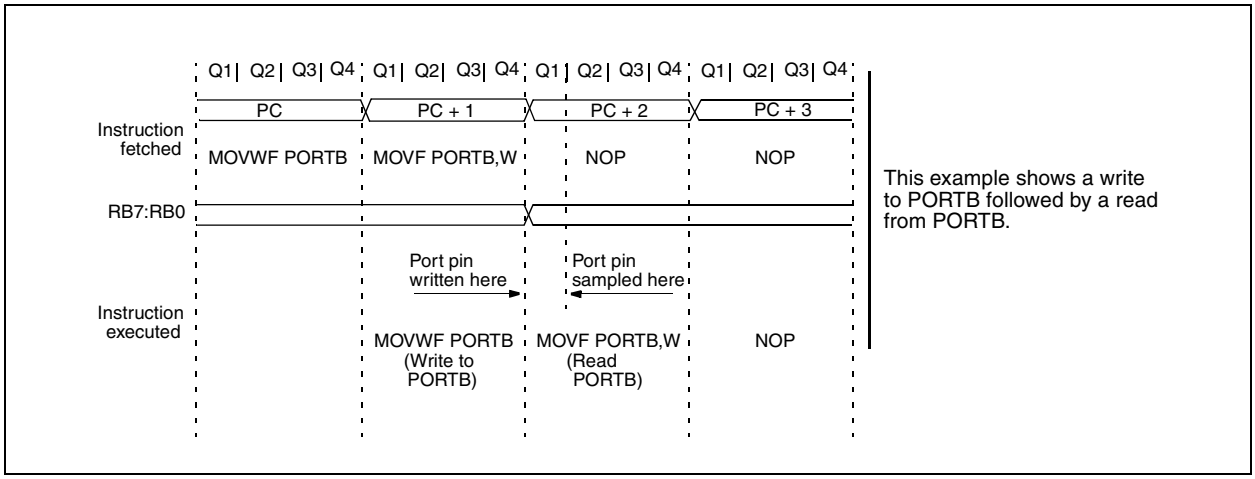


FIGURE 5-5: SUCCESSIVE I/O OPERATION



NOTES:

7.0 SPECIAL FEATURES OF THE CPU

What sets a microcontroller apart from other processors are special circuits that deal with the needs of real-time applications. The PIC16HV540 family of microcontrollers has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These features are:

- Oscillator selection
- Reset
- Power-On Reset (POR)
- Brown-out Detect
- Device Reset Timer (DRT)
- Wake-up from SLEEP on Pin Change
- Enhanced Watchdog Timer (WDT)
- SLEEP
- Code protection

The PIC16HV540 Family has a Watchdog Timer which can be shut off only through configuration bit WDTEN. It runs off of its own RC oscillator for added reliability. There is an 18 ms delay provided by the Device Reset Timer (DRT), intended to keep the chip in reset until the crystal oscillator is stable. With this timer on-chip, most applications need no external reset circuitry.

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

7.1 Configuration Bits

Configuration bits can be programmed to select various device configurations. Two bits are for the selection of the oscillator type and one bit is the Watchdog Timer enable bit. Nine bits are code protection bits (Figure 7-1) for the PIC16HV540 devices.

REGISTER 7-1: CONFIGURATION WORD FOR PIC16HV540

CP	CP	CP	CP	CP	CP	CP	CP	CP	WDTEN	FOSC1	FOSC0
bit11											bit0

Register:CONFIG
Address(1):0FFFh

bit 11-3: **CP:** Code Protection bits
1 = Code protection off
0 = Code protection on

bit 2: **WDTEN:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled (control is placed on the $\overline{\text{SWDTEN}}$ bit)

bit 1-0: **Fosc<1:0>:** Oscillator Selection bits
11 = RC oscillator
10 = HS oscillator
01 = XT oscillator
00 = LP oscillator

Note 1: Refer to the PIC16C5X Programming Specification (Literature number DS30190) to determine how to access the configuration word.

7.2 Oscillator Configurations

7.2.1 OSCILLATOR TYPES

The PIC16HV540 can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1:FOSC0) to select one of these four modes:

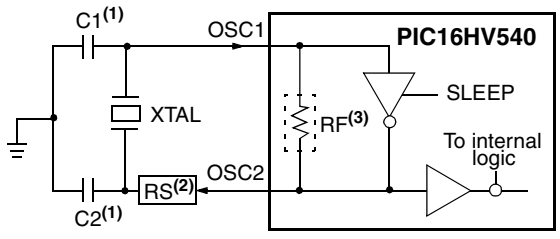
- LP: Low Power Crystal
- XT: Crystal/Resonator
- HS: High Speed Crystal/Resonator
- RC: Resistor/Capacitor

Note:	Not all oscillator selections available for all parts. See Section 7.1.

7.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 7-1). The PIC16HV540 oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source drive the OSC1/CLKIN pin (Figure 7-2).

FIGURE 7-1: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)



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

FIGURE 7-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

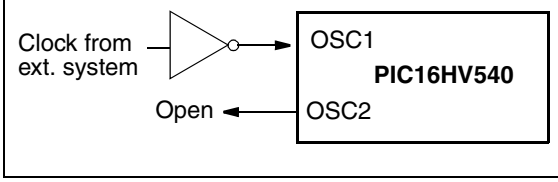


TABLE 7-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS - PIC16HV540

Osc Type	Resonator Freq	Cap. Range C1	Cap. Range C2
XT	455 kHz 2.0 MHz 4.0 MHz	68-100 pF 15-33 pF 10-22 pF	68-100 pF 15-33 pF 10-22 pF
HS	8.0 MHz 16.0 MHz	10-22 pF 10 pF	10-22 pF 10 pF

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

TABLE 7-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR - PIC16HV540

Osc Type	Resonator Freq	Cap.Range C1	Cap. Range C2
LP	32 kHz(1)	15 pF	15 pF
XT	100 kHz 200 kHz 455 kHz 1 MHz 2 MHz 4 MHz	15-30 pF 15-30 pF 15-30 pF 15-30 pF 15 pF 15 pF	200-300 pF 100-200 pF 15-100 pF 15-30 pF 15 pF 15 pF
HS	4 MHz 8 MHz 20 MHz	15 pF 15 pF 15 pF	15 pF 15 pF 15 pF

Note 1: For V_{DD} > 4.5V, C1 = C2 ≈ 30 pF is recommended.
2: These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

Note:	If you change from this device to another device, please verify oscillator characteristics in your application.

7.11 Program Verification/Code Protection

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

Note:	Microchip does not recommend code protecting windowed devices.

7.12 ID Locations

Four memory locations are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify.

Use only the lower 4 bits of the ID locations and always program the upper 8 bits as '1's.

Note:	Microchip will assign a unique pattern number for QTP and SQTP requests and for ROM devices. This pattern number will be unique and traceable to the submitted code.

TABLE 8-2: INSTRUCTION SET SUMMARY

Mnemonic, Operands		Description	Cycles	12-Bit Opcode MSb LSb	Status Affected	Notes
ADDWF	f,d	Add W and f	1	0001 11df ffff	C,DC,Z	1,2,4
ANDWF	f,d	AND W with f	1	0001 01df ffff	Z	2,4
CLRF	f	Clear f	1	0000 011f ffff	Z	4
CLRW	–	Clear W	1	0000 0100 0000	Z	
COMF	f, d	Complement f	1	0010 01df ffff	Z	
DECF	f, d	Decrement f	1	0000 11df ffff	Z	2,4
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	0010 11df ffff	None	2,4
INCF	f, d	Increment f	1	0010 10df ffff	Z	2,4
INCFSZ	f, d	Increment f, Skip if 0	1(2)	0011 11df ffff	None	2,4
IORWF	f, d	Inclusive OR W with f	1	0001 00df ffff	Z	2,4
MOVF	f, d	Move f	1	0010 00df ffff	Z	2,4
MOVWF	f	Move W to f	1	0000 001f ffff	None	1,4
NOP	–	No Operation	1	0000 0000 0000	None	
RLF	f, d	Rotate left f through Carry	1	0011 01df ffff	C	2,4
RRF	f, d	Rotate right f through Carry	1	0011 00df ffff	C	2,4
SUBWF	f, d	Subtract W from f	1	0000 10df ffff	C,DC,Z	1,2,4
SWAPF	f, d	Swap f	1	0011 10df ffff	None	2,4
XORWF	f, d	Exclusive OR W with f	1	0001 10df ffff	Z	2,4
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b	Bit Clear f	1	0100 bbbf ffff	None	2,4
BSF	f, b	Bit Set f	1	0101 bbbf ffff	None	2,4
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	0110 bbbf ffff	None	
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	0111 bbbf ffff	None	
LITERAL AND CONTROL OPERATIONS						
ANDLW	k	AND literal with W	1	1110 kkkk kkkk	Z	
CALL	k	Call subroutine	2	1001 kkkk kkkk	None	1
CLRWDT	k	Clear Watchdog Timer	1	0000 0000 0100	$\overline{TO}$, $\overline{PD}$	
GOTO	k	Unconditional branch	2	101k kkkk kkkk	None	
IORLW	k	Inclusive OR Literal with W	1	1101 kkkk kkkk	Z	
MOVLW	k	Move Literal to W	1	1100 kkkk kkkk	None	
OPTION	k	Load OPTION register	1	0000 0000 0010	None	
RETLW	k	Return, place Literal in W	2	1000 kkkk kkkk	None	
SLEEP	–	Go into standby mode	1	0000 0000 0011	$\overline{TO}$, $\overline{PD}$,$\overline{PCWUF}$	
TRIS	f	Load TRIS register	1	0000 0000 0fff	None	3
XORLW	k	Exclusive OR Literal to W	1	1111 kkkk kkkk	Z	

Note 1: The 9th bit of the program counter will be forced to a '0' by any instruction that writes to the PC except for GOTO. (See individual device data sheets, Memory Section/Indirect Data Addressing, INDF and FSR Registers)

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

3: The instruction TRIS f, where f = 5 or 6 causes the contents of the W register to be written to the tristate latches of PORTA or B respectively. A '1' forces the pin to a hi-impedance state and disables the output buffers.

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

CALL **Subroutine Call**

Syntax: [*label*] CALL k

Operands: 0 ≤ k ≤ 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 ≤ f ≤ 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

MOVF **Move f**

Syntax: [*label*] MOVF f,d

Operands: $0 \le f \le 31$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Encoding:

0010	00df	ffff

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

Words: 1

Cycles: 1

Example:
```
MOVF    FSR,    0
```
After Instruction
W = value in FSR register

MOVLW **Move Literal to W**

Syntax: [*label*] MOVLW k

Operands: $0 \le k \le 255$

Operation: k → (W)

Status Affected: None

Encoding:

1100	kkkk	kkkk

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

Words: 1

Cycles: 1

Example:
```
MOVLW   0x5A
```
After Instruction
W = 0x5A

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 31$

Operation: (W) → (f)

Status Affected: None

Encoding:

0000	001f	ffff

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

Words: 1

Cycles: 1

Example:
```
MOVWF   TEMP_REG
```
Before Instruction
TEMP_REG = 0xFF
W = 0x4F

After Instruction
TEMP_REG = 0x4F
W = 0x4F

NOP **No Operation**

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

0000	0000	0000

Description: No operation.

Words: 1

Cycles: 1

Example:
```
NOP
```


9.0 DEVELOPMENT SUPPORT

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

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

9.1 MPLAB Integrated Development Environment Software

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

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

MPLAB allows you to:

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

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

9.2 MPASM Assembler

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

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

MPASM features include:

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

9.3 MPLAB-C17 and MPLAB-C18 C Compilers

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

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

9.10 PRO MATE II Universal Programmer

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

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

9.11 PICSTART Plus Entry Level Development System

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

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

9.12 SIMICE Entry-Level Hardware Simulator

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

9.13 PICDEM-1 Low-Cost PICmicro Demonstration Board

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

9.14 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

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

9.15 PICDEM-3 Low-Cost PIC16CXXX Demonstration Board

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

9.16 PICDEM-17

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

9.17 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.

9.18 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.

NOTES:

10.0 ELECTRICAL CHARACTERISTICS - PIC16HV540

Absolute Maximum Ratings†

Ambient temperature under bias ..... –20°C to +85°C

Storage temperature ..... –65°C to +150°C

Voltage on V_{DD} with respect to V_{SS} ..... 0 to +16V

Voltage on $\overline{MCLR}$ with respect to V_{SS} ..... 0 to +14V

Voltage on all other pins with respect to V_{SS} ..... –0.6V to (V_{DD} + 0.6V)

Total power dissipation(1) ..... 800 mW

Max. current out of V_{SS} pin ..... 150 mA

Max. current into V_{DD} pin ..... 100 mA

Max. current into an input pin (T0CKI only) ..... ±500 µA

Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD}) ..... ±20 mA

Output clamp current, I_{OK} (V_O < 0 or V_O > V_{DD}) ..... ±20 mA

Max. output current sunk by any I/O pin ..... 25 mA

Max. output current sourced by any I/O pin ..... 10 mA

Max. output current sourced by a single I/O port A or B ..... 40 mA

Max. output current sourced by a single I/O port A or B ..... 50 mA

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

2: Voltage spikes below V_{SS} at the $\overline{MCLR}$ pin, inducing currents greater than 80mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to V_{SS}.

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

10.2 DC Characteristics: PIC16HV540-04, 20 (Commercial) PIC16HV540-04I, 20I (Industrial)

DC Characteristics All Pins Except Power Supply Pins		Standard Operating Conditions (unless otherwise specified) Operating Temperature 0°C ≤ TA ≤ +70°C (commercial) –40°C ≤ TA ≤ +85°C (industrial)				
Characteristic	**Sym.**	**Min.**	**Typ.(1)**	**Max.**	**Units**	**Conditions**
Input Low Voltage	VIL					
I/O Ports PORTA		VSS	—	0.10 VREG	V	Pin at Hi-impedance
$\overline{MCLR}$ (Schmitt Trigger)		VSS	—	0.10 VREG	V	
T0CKI (Schmitt Trigger)		VSS	—	0.10 VREG	V	
OSC1 (Schmitt Trigger)		VSS	—	0.10 VREG	V	RC option only(4)
OSC1		VSS	—	0.3 VREG	V	HS, XT, and LP options
I/O Ports PORTB		VSS	—	0.10 VREG	V	
Input High Voltage	VIH					
I/O Ports PORTA		0.25 VREG+0.8V	—	VREG	V	For all VREG
$\overline{MCLR}$ (Schmitt Trigger)		0.85 VREG	—	VDD	V	
T0CKI (Schmitt Trigger)		0.85 VREG	—	VDD	V	
OSC1 (Schmitt Trigger)		4.5V	—	VDD	V	RC option only (VDD = 15V)(4)
OSC1		4.5V	—	VDD	V	HS, XT, and LP options (VDD = 15V)
I/O Ports PORTB		0.25 VREG+0.8V	—	VDD	V	
Hysteresis of Schmitt Trigger inputs	VHYS	0.15 VREG*	—	—	V	
Input Leakage Current(3)	IIL					
I/O Ports PORTA		-1.0	0.5	+1.0	µA	VSS ≤ VPIN ≤ VIO, Pin at Hi-impedance
I/O Ports PORTB		-1.0	0.5	+1.0	µA	VSS ≤ VPIN ≤ VDD
$\overline{MCLR}$		-5.0		+5.0	µA	VPIN = VSS +0.25V(2)
			0.5	+3.0	µA	VPIN = VDD(2)
T0CKI		-3.0	0.5	+3.0	µA	VSS ≤ VPIN ≤ VDD
OSC1		-3.0	0.5	+3.0	µA	VSS ≤ VPIN ≤ VDD, HS, XT, and LP options
Output Low Voltage	VOL					
I/O Ports PORTA		—	—	0.6	V	VDD = 15V, VREG = 5V, IOL = 8.7 mA
						VDD = 15V, VREG = 3V, IOL = 5.0 mA
OSC2/CLKOUT		—	—	0.6	V	VDD = 15V, VREG = 5V, IOL = 1.2 mA, (RC option only)
						VDD = 15V, VREG = 3V, IOL = 1.0 mA, (RC option only)
I/O Ports PORTB		—	—	0.6	V	VDD = 15V, VREG = 5V, IOL = 3.0 mA
						VDD = 10V, VREG = 3V, IOL = 3.0 mA
Output High Voltage	VOH					
I/O ports(3) PORTA		VREG-0.7	—	—	V	VDD = 15V, VIO = 3V, IOH = -2.0 mA
						VDD = 15V, VIO = 5V, IOH = -3.0 mA
OSC2/CLKOUT		VREG-0.7	—	—	V	VDD = 15V, VIO = 3V, IOH = -0.5 mA (RC option only)
						VDD = 15V, VIO = 5V, IOH = -1.0 mA (RC option only)
I/O Ports PORTB		VDD-0.7	—	—	V	VDD = 15V, VIO = 5V, IOH = -5.4 mA
Threshold Voltage						
I/O Ports PORTB [7]	VLEV	VDD-1.5	VDD-1.0	VDD-0.5	V	VDD = 15V

* 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: The leakage current on the $\overline{MCLR}$/VPP 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 voltage.

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

4: For the RC option, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16HV540 be driven with external clock in RC mode.

11.0 DC AND AC CHARACTERISTICS - PIC16HV540

The graphs and tables provided in this section are for design guidance and are not tested or guaranteed. In some graphs or tables the data presented are outside specified operating range (e.g., 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 11-1: TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE

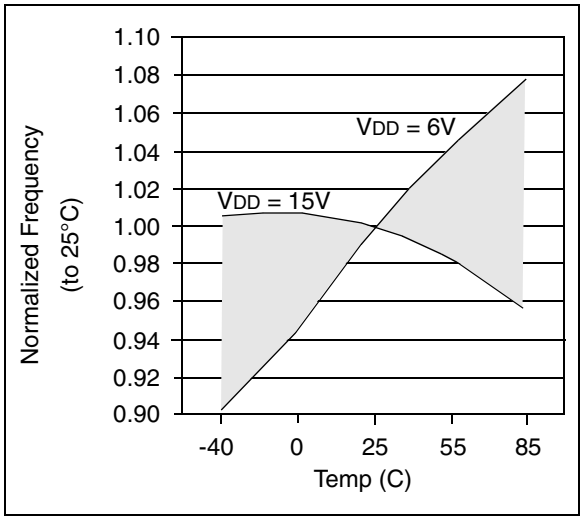


TABLE 11-1: RC OSCILLATOR FREQUENCIES

CEXT	REXT	Average FOSC, VIO = 5V	
		25°C, VDD = 6V	25°C, VDD = 15V
20 pF	3.3k	4986.7 kHz	**(1)**
	5k	4233.3 kHz	**(1)**
	10k	2656.7 kHz	5150.0 kHz
	24k	1223.3 kHz	3286.7 kHz
	100k	325.7 kHz	955.7 kHz
	390k	79.0 kHz	250.7 kHz
100 pF	3.3k	1916.7 kHz	**(1)**
	5k	1593.3 kHz	**(1)**
	10k	995.7 kHz	2086.7 kHz
	24k	448.3 kHz	1210.0 kHz
	100k	116.0 kHz	355.7 kHz
	390k	28.3 kHz	89.7 kHz
300 pF	3.3k	744 kHz	**(1)**
	5k	620.3 kHz	**(1)**
	10k	382.0 kHz	817.3 kHz
	24k	169.7 kHz	483.0 kHz
	100k	44.1 kHz	135.7 kHz
	390k	10.6 kHz	34.4 kHz

Note 1: This combination of R, C and VDD draws too much current and prohibits oscillator operation.

FIGURE 11-2: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD (CEXT = 20pF)

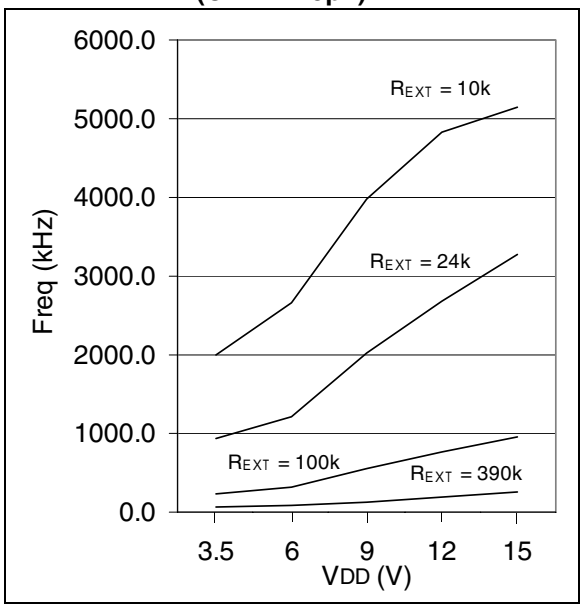


FIGURE 11-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD (CEXT = 100pF)

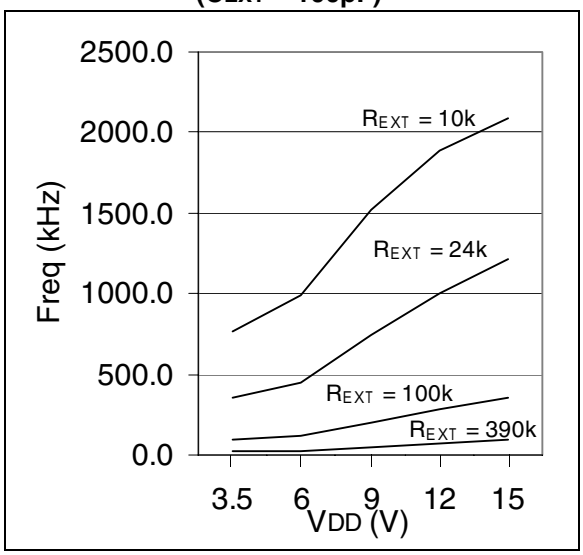


FIGURE 11-10: MAXIMUM IPD vs. VDD, WATCHDOG TIMER DISABLED (VIO = 3V)

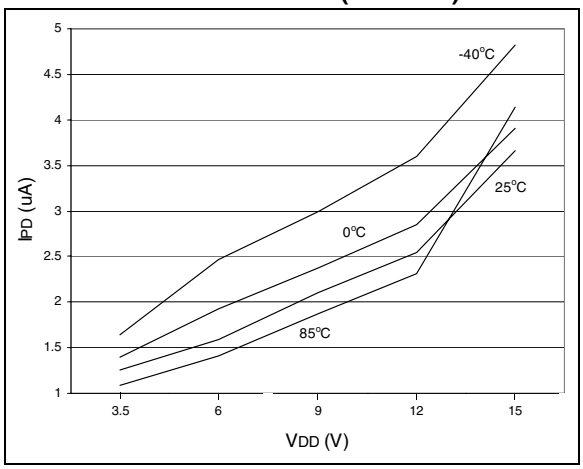


FIGURE 11-11: TYPICAL IPD vs. VDD, WATCHDOG TIMER ENABLED (VIO = 3V)

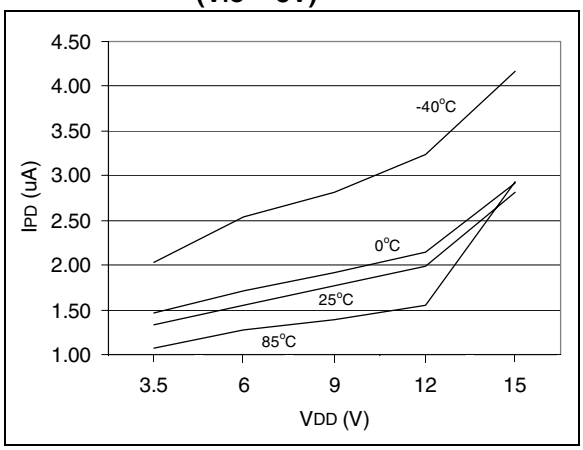


FIGURE 11-12: MAXIMUM IPD vs. VDD, WATCHDOG TIMER ENABLED (VIO = 3V)

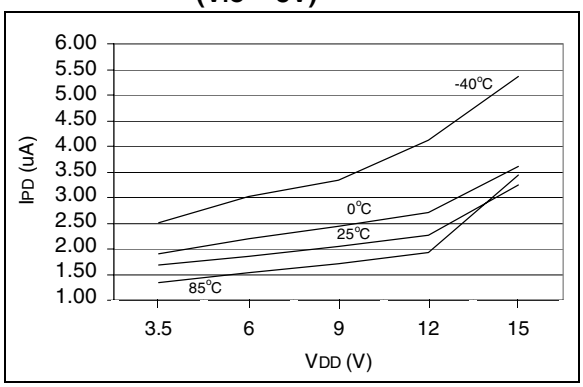


FIGURE 11-13: MAXIMUM IDD vs. FREQUENCY, WATCHDOG TIMER DISABLED, RC MODE (VDD = 15V, VIO = 5V, -40°C TO +85°C)

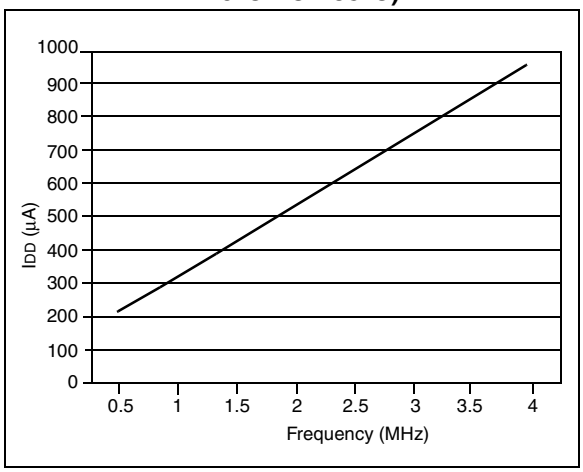


FIGURE 11-14: MAXIMUM IDD vs. FREQUENCY, WATCHDOG TIMER ENABLED, RC MODE (VDD = 15V, VIO = 5V)