PIC12F529T48A
Data Sheet

14-Pin, 8-Bit Flash Microcontrollers

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

ISBN: 9781620762424

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

PIC12F529T48A

14-Pin, 8-Bit Flash Microcontroller

High-Performance RISC CPU:

- Only 34 Single-Word Instructions
- All Single-Cycle Instructions Except for Program Branches which are Two-Cycle
- Four-Level Deep Hardware Stack
- Direct, Indirect and Relative Addressing modes for Data and Instructions
- Operating Speed:
  - DC – 8 MHz internal clock
  - DC – 500 ns instruction cycle

Special Microcontroller Features:

- 8 MHz Precision Internal Oscillator
  - Factory calibrated to ±1%
- In-Circuit Serial Programming™ (ICSP™)
- Power-on Reset (POR)
- Device Reset Timer (DRT)
- Watchdog Timer (WDT) with Dedicated On-Chip RC Oscillator for Reliable Operation
- Programmable Code Protection
- Multiplexed $\overline{\text{MCLR}}$ Input Pin
- Internal Weak Pull-Ups on I/O Pins
- Power-Saving Sleep mode
- Wake-up from Sleep on Pin Change
- Selectable Oscillator Options:
  - INTRC: 4 MHz or 8 MHz precision internal RC oscillator
  - EXTRC: External low-cost RC oscillator
  - XT: Standard crystal/resonator
  - LP: Power-saving, low-frequency crystal

Low-Power Features/CMOS Technology:

- Standby Current:
  - 250 nA @ 2.0V, RF Sleep, typical
- Operating Current:
  - 170 µA @ 4 MHz, 2.0V, RF Sleep, typical
  - 9.17 mA @ 4 MHz, 2.0V, RF on at +0 dBm, typical
  - 16.67 mA @ 4 MHz, 2.0V, RF on at +10 dBm, typical
- Watchdog Timer Current:
  - 1 µA @ 2.0V, typical
- High Endurance Program and Flash Data Memory Cells
  - 100,000 write Flash program memory endurance
  - 1,000,000 write Flash data memory endurance
  - Program and Flash data retention: >40 years
- Fully Static Design
- Operating Voltage Range: 2.0V to 3.7V
  - Industrial temperature range: -40°C to +85°C

RF Transmitter:

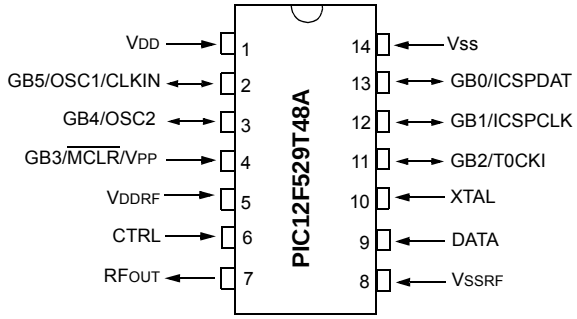
- Fully Integrated Transmitter
- FSK Operation up to 100 kbps
- OOK Operation up to 10 kbps
- Operation in 418, 434 and 868 MHz Bands:
  - 8 selectable center frequencies
- Configurable Output Power: +10 dBm, 0 dBm

Peripheral Features:

- 6 I/O Pins:
  - 5 I/O pins with individual direction control
  - 1 input-only pin
  - High-current sink/source for direct LED drive
- 8-Bit Real-Time Clock/Counter (TMR0) with 8-Bit Programmable Prescaler

FIGURE 1: 14-PIN DIAGRAM, PIC12F529T48A

TSSOP



Device	Program Memory	Data Memory		I/O	RF Transmitter	Comparators	Timers (8-bit)	8-bit A/D Channels
	Flash (words)	SRAM (bytes)	Flash (bytes)					
PIC12F529T48A	1536	201	64	6	1	0	1	0

Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; **http://www.microchip.com**
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

NOTES:

1.0 GENERAL DESCRIPTION

The PIC12F529T48A device from Microchip Technology is a low-cost, high-performance, 8-bit, fully-static, Flash-based CMOS microcontroller. It employs a RISC architecture with only 34 single-word/single-cycle instructions. All instructions are single cycle except for program branches, which take two cycles. The PIC12F529T48A device delivers performance an order of magnitude higher than their competitors in the same price category. The 12-bit wide instructions are highly symmetrical, resulting in a typical 2:1 code compression over other 8-bit microcontrollers in its class. The easy-to-use and easy to remember instruction set reduces development time significantly.

The PIC12F529T48A product is equipped with special features that reduce system cost and power requirements. The Power-on Reset (POR) and Device Reset Timer (DRT) eliminate the need for external Reset circuitry. There are four oscillator configurations to choose from including INTRC Internal Oscillator mode and the power-saving LP (Low-power) Oscillator mode. Power-Saving Sleep mode, Watchdog Timer and code protection features improve system cost, power and reliability.

The PIC12F529T48A device is available in the cost-effective Flash programmable version, which is suitable for production in any volume. The customer can take full advantage of Microchip's price leadership in Flash programmable microcontrollers, while benefiting from the Flash programmable flexibility.

The PIC12F529T48A product is supported by a full-featured macro assembler, a software simulator, a low-cost development programmer and a full-featured programmer. All the tools are supported on PC and compatible machines.

1.1 Applications

The PIC12F529T48A device fits in applications ranging from personal care appliances and security systems to low-power remote transmitters/receivers. The Flash technology makes customizing application programs (transmitter codes, appliance settings, receiver frequencies, etc.) extremely fast and convenient. The small footprint packages, for through hole or surface mounting, make these microcontrollers perfect for applications with space limitations. Low cost, low power, high performance, ease of use and I/O flexibility make the PIC12F529T48A device very versatile even in areas where no microcontroller use has been considered before (e.g., timer functions, logic and programmable logic devices (PLDs) in larger systems and coprocessor applications).

TABLE 1-1: FEATURES AND MEMORY OF PIC12F529T48A

		PIC12F529T48A
Clock	Maximum Frequency of Operation (MHz)	8
Memory	Flash Program Memory	1536
	SRAM Data Memory (bytes)	201
	Flash Data Memory (bytes)	64
Peripherals	Timer Module(s)	TMR0
	Wake-up from Sleep on Pin Change	Yes
Features	I/O Pins	5
	Input Pins	1
	Internal Pull-ups	Yes
	In-Circuit Serial Programming™	Yes
	Number of Instructions	34
	Packages	14-pin TSSOP

Note 1: The PIC12F529T48A device has Power-on Reset, selectable Watchdog Timer, selectable code-protect, high I/O current capability and precision internal oscillator.

2: The PIC12F529T48A device uses serial programming with data pin GP0 and clock pin GP1.

NOTES:

2.0 PIC12F529T48A DEVICE VARIETIES

When placing orders, please use the PIC12F529T48A Product Identification System at the back of this data sheet to specify the correct part number. Depending on application and production requirements, the proper device option can be selected using the information in this section.

2.1 Quick Turn Programming (QTP) Devices

Microchip offers a QTP programming service for factory production orders. This service is made available for users who choose not to program medium-to-high quantity units and whose code patterns have stabilized. The devices are identical to the Flash devices but with all Flash locations and fuse options already programmed by the factory. Certain code and prototype verification procedures do apply before production shipments are available. Please contact your local Microchip Technology sales office for more details.

2.2 Serialized Quick Turn Programming^{SM} (SQTP^{SM}) Devices

Microchip offers a unique programming service, where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number, which can serve as an entry code, password or ID number.

NOTES:

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC12F529T48A device can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC12F529T48A device uses a Harvard architecture in which program and data are accessed on separate buses. This improves bandwidth over traditional von Neumann architectures where program and data are fetched on the same bus. Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 12 bits wide, making it possible to have all single-word instructions. A 12-bit wide program memory access bus fetches a 12-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (34) execute in a single cycle (500 ns @ 8 MHz, 1 μs @ 4 MHz) except for program branches.

Table 3-1 below lists memory supported by the PIC12F529T48A device.

TABLE 3-1: PIC12F529T48A MEMORY

Device	Program Memory	Data Memory	
	Flash (words)	SRAM (bytes)	Flash Data (bytes)
PIC12F529T48A	1536	201	64

The PIC12F529T48A device can directly or indirectly address its register files and data memory. All Special Function Registers (SFR), including the PC, are mapped in the data memory. The PIC12F529T48A device has a highly orthogonal (symmetrical) instruction set that makes it possible to carry out any operation, on any register, using any addressing mode. This symmetrical nature and lack of “special optimal situations” make programming with the PIC12F529T48A device simple, yet efficient. In addition, the learning curve is reduced significantly.

The PIC12F529T48A device contains an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

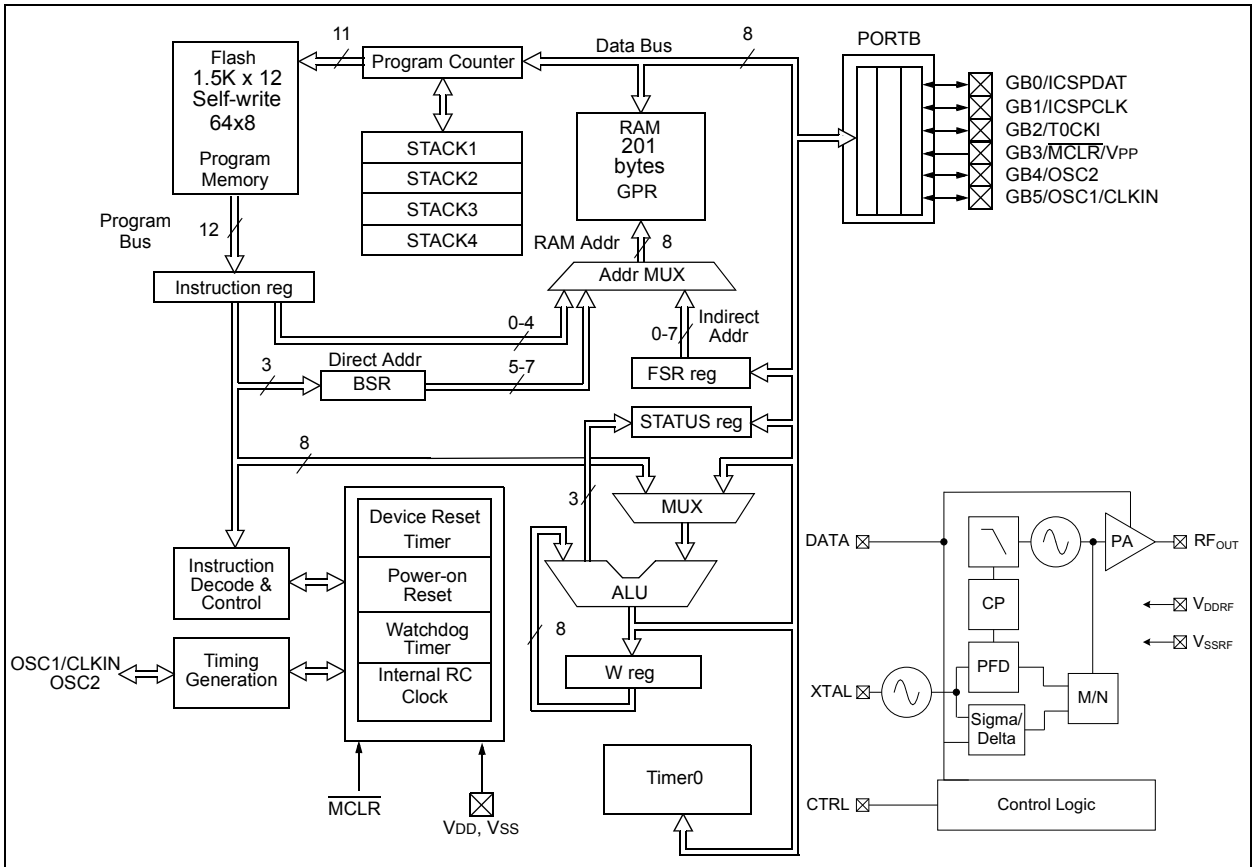
The ALU is 8 bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two’s complement in nature. In two-operand instructions, one operand is typically the W (working) register. The other operand is either a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC) and Zero (Z) bits in the STATUS register. The C and DC bits operate as a borrow and digit borrow out bit, respectively, in subtraction. See the SUBWF and ADDWF instructions for examples.

A simplified block diagram is shown in Figure 3-1, with the corresponding device pins described in Table 3-2.

FIGURE 3-1: PIC12F529T48A ARCHITECTURAL BLOCK DIAGRAM



Note 1: 201-byte GPR in PIC12F529T48A, including linear RAM.
2: FSR and direct addressing differs from standard baseline parts.

TABLE 3-2: PIC12F529T48A PINOUT DESCRIPTION

Name	Function	Type	Input Type	Output Type	Description
GP0/ICSPDAT	GP0	I/O	TTL	CMOS	Bidirectional I/O port with weak pull-up.
	ICSPDAT	I/O	ST	CMOS	ICSP™ mode Schmitt Trigger.
GP1/ICSPCLK	GP1	I/O	TTL	CMOS	Bidirectional I/O port with weak pull-up.
	ICSPCLK	I	ST	—	ICSP™ mode Schmitt Trigger.
GP2/T0CKI	GP2	I/O	TTL	CMOS	Bidirectional I/O port.
	T0CKI	I	ST	—	Timer0 clock input.
GP3/$\overline{MCLR}$/V_{PP}	GP3	I	TTL	—	Standard TTL input with weak pull-up.
	$\overline{MCLR}$	I	ST	—	$\overline{MCLR}$ input (weak pull-up always enabled in this mode).
	V_{PP}	I	High Voltage	—	Test mode high voltage pin.
GP4/OSC2	GP4	I/O	TTL	CMOS	Bidirectional I/O port.
	OSC2	O	—	XTAL	XTAL oscillator output pin.
GP5/OSC1/ CLKIN	GP5	I/O	TTL	CMOS	Bidirectional I/O port.
	OSC1	I	XTAL	—	XTAL oscillator input pin.
	CLKIN	I	ST	—	EXTRC Schmitt Trigger input.
V_{DD}	V_{DD}	P	—	—	Positive supply for logic and I/O pins.
V_{SS}	V_{SS}	P	—	—	Ground reference for logic and I/O pins.
V_{DDRF}	V_{DDRF}	P	Power	—	RF Power Supply.
CTRL	CTRL	I	CMOS	—	Configuration Selection and Configuration Clock.
RF_{OUT}	RF_{OUT}	—	—	RF	Transmitter RF output.
V_{SSRF}	V_{SSRF}	P	Power	—	RF Power Supply.
DATA	DATA	I/O	CMOS	CMOS	Configuration Data and Transmit Data.
XTAL	XTAL	—	XTAL	—	Crystal Oscillator.

Legend: I = Input, O = Output, I/O = Input/Output, P = Power, — = Not Used, TTL = TTL input, ST = Schmitt Trigger input, AN = Analog Voltage

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 PC 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.

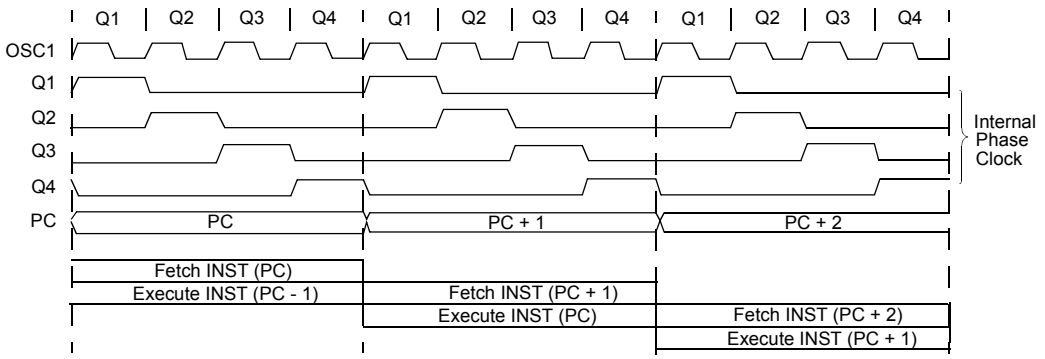
3.2 Instruction Flow/Pipelining

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 take another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the PC to change (e.g., GOTO), then two cycles are required to complete the instruction (Example 3-1).

A fetch cycle begins with the PC incrementing in Q1.

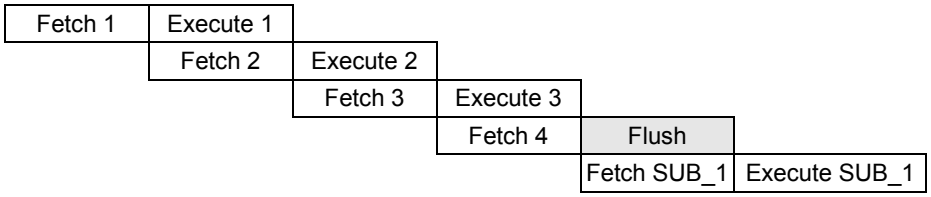
In the execution cycle, the fetched instruction is latched into the Instruction Register (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW

```
1. MOVLW 03H
2. MOVWF GPIO
3. CALL  SUB_1
4. BSF   GPIO, 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

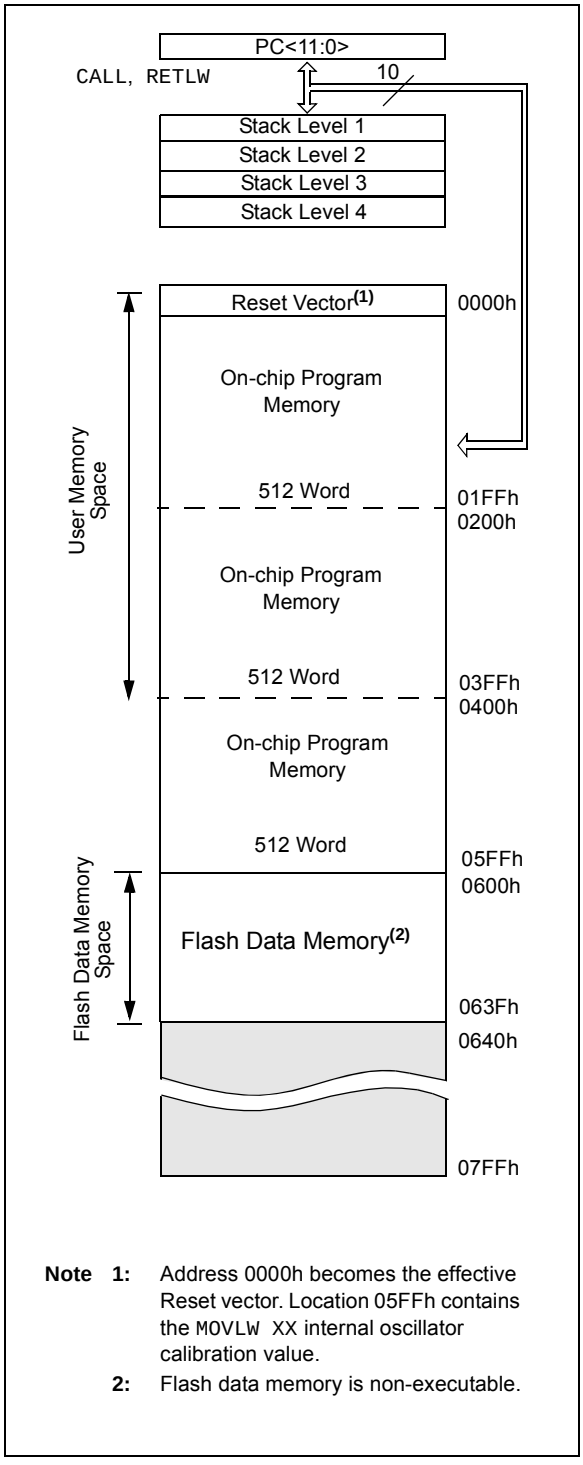
The PIC12F529T48A memory is organized into program memory and data memory (SRAM). The self-writable portion of the program memory called Flash data memory, is located at addresses 600h-63Fh. As the device has more than 512 bytes of program memory, a paging scheme is used. Program memory pages are accessed using STATUS register bit, PA0. For the PIC12F529T48A, with data memory register files of more than 32 registers, a banking scheme is used. Data memory banks are accessed using the File Select Register (FSR).

4.1 Program Memory Organization for the PIC12F529T48A

The PIC12F529T48A device has an 11-bit Program Counter (PC) capable of addressing a 2K x 12 program memory space.

Only the first 1.5K x 12 (0000h-05FFh) are physically implemented (see Figure 4-1). Accessing a location above these boundaries will cause a wrap-around within the 1.5K x 12 space. The effective Reset vector is a 0000h (see Figure 4-1). Location 05FFh contains the internal clock oscillator calibration value. This value should never be overwritten.

FIGURE 4-1: MEMORY MAP



Note 1: Address 0000h becomes the effective Reset vector. Location 05FFh contains the MOVLW XX internal oscillator calibration value.

2: Flash data memory is non-executable.

4.2 Data Memory (SRAM and FSRs)

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

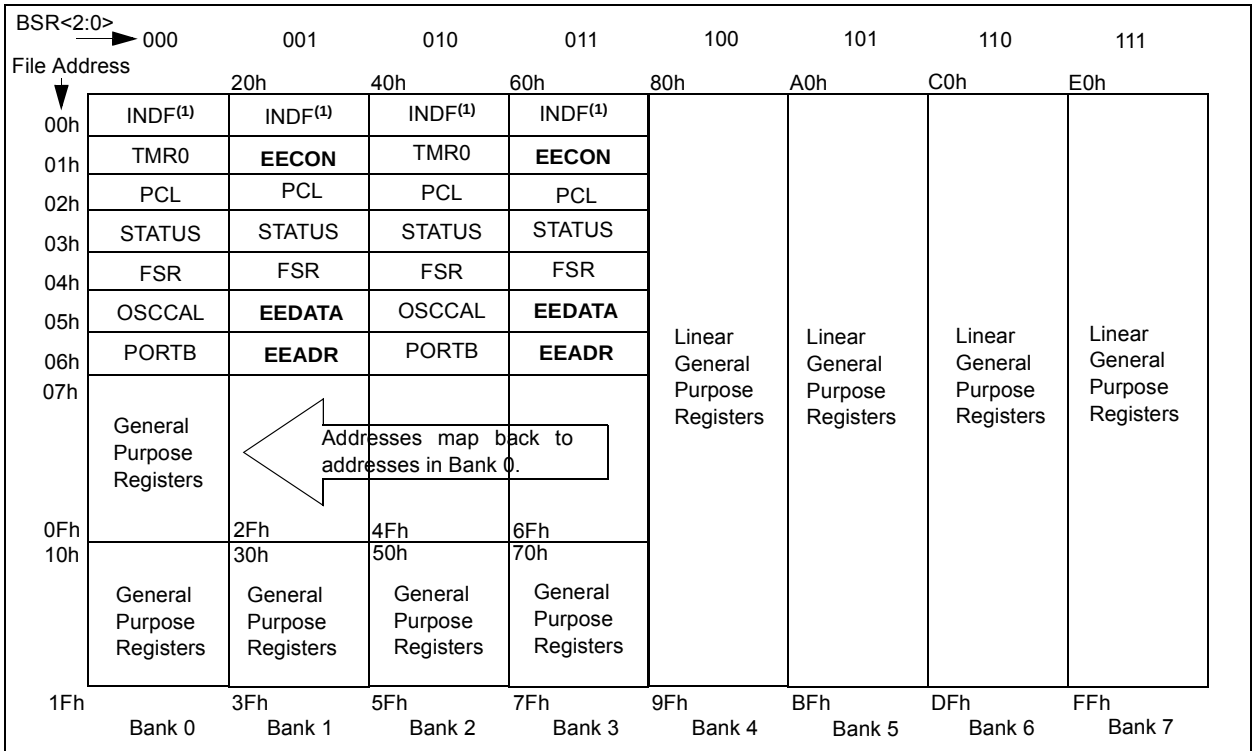
The Special Function Registers include the TMR0 register, the Program Counter Low (PCL), the STATUS register, the I/O register (port) and the File Select Register (FSR). In addition, the EECON, EEDATA and EEADR registers provide for interface with the Flash data memory.

The PIC12F529T48A register file is composed of 10 Special Function Registers and 201 General Purpose Registers.

4.2.1 GENERAL PURPOSE REGISTER FILE

The General Purpose Register file is accessed, either directly or indirectly, through the File Select Register (FSR). See **Section 4.8 "Indirect Data Addressing: INDF and FSR Registers"**.

FIGURE 4-2: REGISTER FILE MAP



Note 1: Not a physical register.

4.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) are registers used by the CPU and peripheral functions to control the operation of the device (Table 4-1).

The Special Function Registers can be classified into two sets. The Special Function Registers associated with the “core” functions are described in this section. Those related to the operation of the peripheral features are described in the section for each peripheral feature.

4.2.3 LINEAR RAM

The last four banks, addresses 0x80 to 0xFF, are general purpose RAM registers, unbroken by SFRs. This region is ideal for indirect access using the FSR and INDF registers.

Note:	Unlike other baseline devices, the FSR register does not contain bank bits and, therefore, does not affect direct addressing schemes. The FSR/INDF registers have full access to RAM.

TABLE 4-1: SPECIAL FUNCTION REGISTER SUMMARY

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset
N/A	TRISGPIO	—	—	TRISGPIO5	TRISGPIO4	TRISGPIO3	TRISGPIO2	TRISGPIO1	TRISGPIO0	--11 1111
N/A	OPTION	Contains Control Bits to Configure Timer0 and Timer0/WDT Prescaler								1111 1111
N/A	BSR	—	—	—	—	—	BSR<2:0>			---- -000
00h	INDF	Uses Contents of FSR to Address Data Memory (not a physical register)								xxxx xxxx
01h	TMR0	Timer0 Module Register								xxxx xxxx
02h(1)	PCL	Low Order 8 bits of PC								1111 1111
03h	STATUS	GPWUF	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx
04h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx
05h	OSCCAL	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	—	1111 111-
06h	GPIO	—	—	GP5	GP4	GP3	GP2	GP1	GP0	--xx xxxx
21h	EECON	—	—	—	FREE	WRERR	WREN	WR	RD	---0 x000
25h	EEDATA	EEDATA7	EEDATA6	EEDATA5	EEDATA4	EEDATA3	EEDATA2	EEDATA1	EEDATA0	xxxx xxxx
26h	EEADR	—	—	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	--xx xxxx

Legend: x = unknown, u = unchanged, – = unimplemented, read as ‘0’ (if applicable). Shaded cells = unimplemented or unused

Note 1: The upper byte of the Program Counter is not directly accessible. See **Section 4.6 “Program Counter”** for an explanation of how to access these bits.

4.3 STATUS register

This register contains the arithmetic status of the ALU, the Reset status and the page preselect bit.

The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, CLRF STATUS, will clear the upper three bits and set the Z bit. This leaves the STATUS register as '000u u1uu' (where u = unchanged).

Therefore, it is recommended that only BCF, BSF and MOVWF instructions be used to alter the STATUS register. These instructions do not affect the Z, DC or C bits from the STATUS register. For other instructions which do affect Status bits, see **Section 10.0 "Instruction Set Summary"**.

REGISTER 4-1: STATUS: STATUS REGISTER

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
GPWUF	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **GPWUF**: Wake-up From Sleep on Pin Change bit
1 = Reset due to wake-up from Sleep on pin change
0 = After power-up or other Reset

bit 6-5 **PA<1:0>:** Program Page Preselect bits(1)
00 = Page 0 (000h-1FFh)
01 = Page 1 (200h-3FFh)
10 = Page 2 (400h-5FFh)
11 = Reserved. Do not use.

bit 4 **$\overline{TO}$**: Time-Out bit
1 = After power-up, CLRWDT instruction, or SLEEP instruction
0 = A WDT time-out occurred

bit 3 **$\overline{PD}$**: Power-Down bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

bit 2 **Z**: Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit Carry/$\overline{Borrow}$ bit (for ADDWF and SUBWF instructions)
ADDWF:
1 = A carry from the 4th low-order bit of the result occurred
0 = A carry from the 4th low-order bit of the result did not occur
SUBWF:
1 = A borrow from the 4th low-order bit of the result did not occur
0 = A borrow from the 4th low-order bit of the result occurred

bit 0 **C**: Carry/$\overline{Borrow}$ bit (for ADDWF, SUBWF and RRF, RLF instructions)

ADDWF:	SUBWF:	RRF or RLF:
1 = A carry occurred	1 = A borrow did not occur	Load bit with LSb or MSb, respectively
0 = A carry did not occur	0 = A borrow occurred	

Note 1: Do not set both PA0 and PA1.

4.4 OPTION Register

The OPTION register is a 8-bit wide, write-only register, which contains various control bits to configure the Timer0/WDT prescaler and Timer0.

By executing the OPTION instruction, the contents of the W register will be transferred to the OPTION register. A Reset sets the OPTION<7:0> bits.

Note:	If the T0SC bit is set to '1', it will override the TRIS function on the T0CKI pin.

REGISTER 4-2: OPTION: OPTION REGISTER

W-1	W-1	W-1	W-1	W-1	W-1	W-1	W-1
$\overline{\text{GPWU}}$	$\overline{\text{GPPU}}$	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	x = Bit is unknown
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared

bit 7 **$\overline{\text{GPWU}}$:** Enable Wake-up On Pin Change bit
1 = Disabled
0 = Enabled

bit 6 **$\overline{\text{GPPU}}$:** Enable Weak Pull-Ups bit
1 = Disabled
0 = Enabled

bit 5 **T0CS:** Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE:** Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler assigned to the WDT
0 = Prescaler assigned to Timer0

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

Bit Value	Timer0 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

4.5 OSCCAL Register

The Oscillator Calibration (OSCCAL) register is used to calibrate the 8 MHz internal oscillator macro. It contains seven bits of calibration that uses a two's complement scheme for controlling the oscillator speed. See Register 4-3 for details.

REGISTER 4-3: OSCCAL: OSCILLATOR CALIBRATION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	U-0
CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-1 **CAL<6:0>**: Oscillator Calibration bits

0111111 = Maximum frequency
•
•
•
0000001
0000000 = Center frequency
1111111
•
•
•
1000000 = Minimum frequency

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

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.

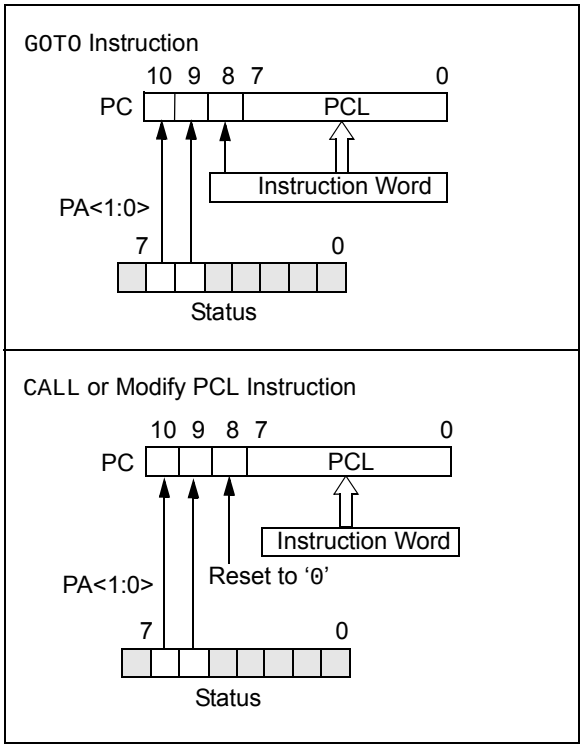
For a GOTO instruction, bits <8:0> of the PC are provided by the GOTO instruction word. The Program Counter (PCL) is mapped to PC<7:0>. Bits 5 and 6 of the STATUS register provide page information to bits 9 and 10 of the PC. (Figure 4-3).

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 PCL, ADDWF PCL and BSF PCL,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



4.6.1 EFFECTS OF RESET

The PC is set upon a Reset, which means that the PC addresses the last location in the last page (i.e., the oscillator calibration instruction). After executing MOVLW XX, the PC will roll over to location 00h and begin executing user code.

The STATUS register page preselect bits are cleared upon a Reset, which means that page 0 is pre-selected.

Therefore, upon a Reset, a GOTO instruction will automatically cause the program to jump to page 0 until the value of the page bits is altered.

4.7 Stack

The PIC12F529T48A device has a four-deep, 12-bit wide hardware PUSH/POP stack.

A CALL instruction will PUSH the current value of Stack 1 into Stack 2 and then PUSH the current PC value, incremented by one, into Stack Level 1. If more than four sequential CALLs are executed, only the most recent four return addresses are stored.

A RETLW instruction will POP the contents of Stack Level 1 into the PC and then copy Stack Level 2 contents into Stack Level 1. If more than four sequential RETLWs are executed, the stack will be filled with the address previously stored in Stack Level 2. 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.

Note 1:	There are no Status bits to indicate Stack Overflow or Stack Underflow conditions.
2:	There are no instruction 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.

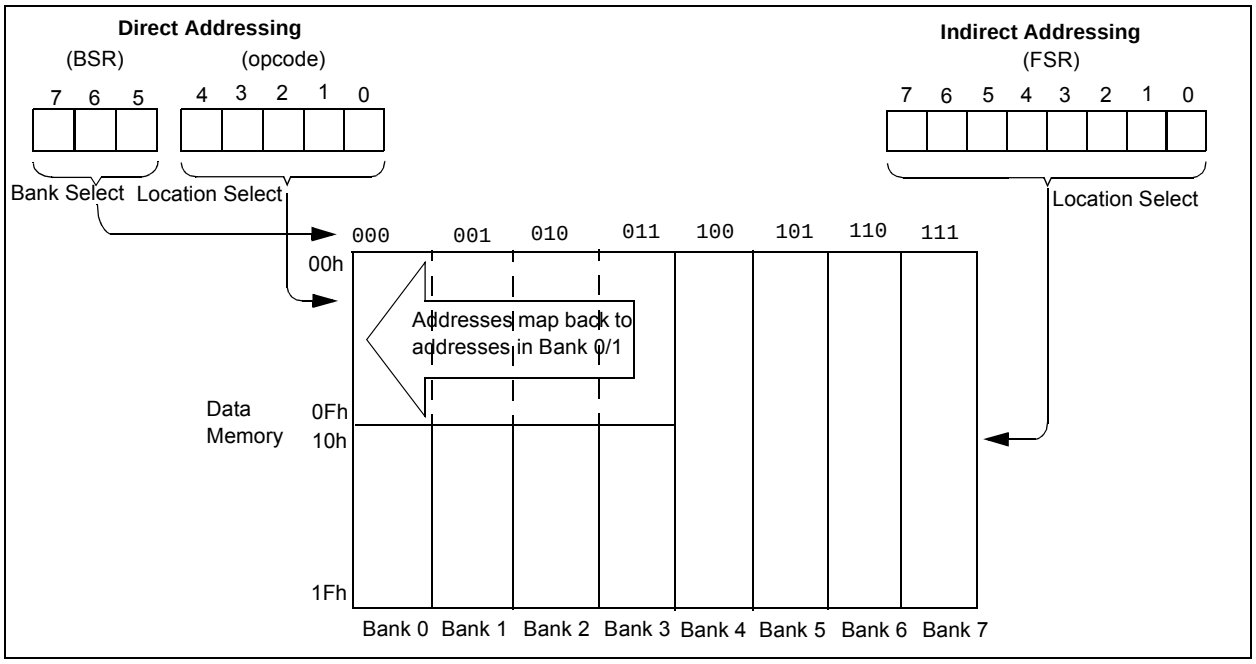
Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no-operation (although Status bits may be affected).

The FSR is an 8-bit wide register. It is used in conjunction with the INDF Register to indirectly address the data memory area.

EXAMPLE 4-1: 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
            :
```

FIGURE 4-4: DIRECT/INDIRECT ADDRESSING



4.9 Direct Data Addressing

Banking when using direct addressing methods is accomplished using the MOVLB instruction to write to the BSR. The BSR, like the OPTION register, is not mapped to user-accessable memory. The value in BSR has no effect on indirect addressed operations.

5.0 FLASH DATA MEMORY CONTROL

The Flash data memory is readable and writable during normal operation (full VDD range). This memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers (SFRs).

5.1 Reading Flash Data Memory

To read a Flash data memory location the user must:

- Write the EEADR register
- Set the RD bit of the EECON register

The value written to the EEADR register determines which Flash data memory location is read. Setting the RD bit of the EECON register initiates the read. Data from the Flash data memory read is available in the EEDATA register immediately. The EEDATA register will hold this value until another read is initiated or it is modified by a write operation. Program execution is suspended while the read cycle is in progress. Execution will continue with the instruction following the one that sets the WR bit. See Example 5-1 for sample code.

EXAMPLE 5-1: READING FROM FLASH DATA MEMORY

```
BANKSEL EEADR            ;
MOVF DATA_EE_ADDR, W     ;
MOVWF EEADR              ;Data Memory
                         ;Address to read
BANKSEL EECON1           ;

BSF EECON, RD            ;EE Read
MOVF EEDATA, W           ;W = EEDATA
```

Note: Only a BSF command will work to enable the Flash data memory read documented in Example 5-1. No other sequence of commands will work, no exceptions.

5.2 Writing and Erasing Flash Data Memory

Flash data memory is erased one row at a time and written one byte at a time. The 64-byte array is made up of eight rows. A row contains eight sequential bytes. Row boundaries exist every eight bytes.

Generally, the procedure to write a byte of data to Flash data memory is:

1. Identify the row containing the address where the byte will be written.
2. If there is other information in that row that must be saved, copy those bytes from Flash data memory to RAM.
3. Perform a row erase of the row of interest.
4. Write the new byte of data and any saved bytes back to the appropriate addresses in Flash data memory.

To prevent accidental corruption of the Flash data memory, an unlock sequence is required to initiate a write or erase cycle. This sequence requires that the bit set instructions used to configure the EECON register happen exactly as shown in Example 5-2 and Example 5-3, depending on the operation requested.

5.2.1 ERASING FLASH DATA MEMORY

A row must be manually erased before writing new data. The following sequence must be performed for a single row erase.

1. Load EEADR with an address in the row to be erased.
2. Set the FREE bit to enable the erase.
3. Set the WREN bit to enable write access to the array.
4. Set the WR bit to initiate the erase cycle.

If the WREN bit is not set in the instruction cycle after the FREE bit is set, the FREE bit will be cleared in hardware.

If the WR bit is not set in the instruction cycle after the WREN bit is set, the WREN bit will be cleared in hardware.

Sample code that follows this procedure is included in Example 5-2.

Program execution is suspended while the erase cycle is in progress. Execution will continue with the instruction following the one that sets the WR bit.

EXAMPLE 5-2: ERASING A FLASH DATA MEMORY ROW

```
BANKSEL   EEADR
MOVLW     EE_ADR_ERASE   ; LOAD ADDRESS OF ROW TO
                         ; ERASE
MOVWF     EEADR          ;
BSF       EECON,FREE     ; SELECT ERASE
BSF       EECON,WREN     ; ENABLE WRITES
BSF       EECON,WR       ; INITITATE ERASE
```

Note 1: The FREE bit may be set by any command normally used by the core. However, the WREN and WR bits can only be set using a series of BSF commands, as documented in Example 5-1. No other sequence of commands will work, no exceptions.

2: Bits <5:3> of the EEADR register indicate which row is to be erased.

5.2.2 WRITING TO FLASH DATA MEMORY

Once a cell is erased, new data can be written. Program execution is suspended during the write cycle. The following sequence must be performed for a single byte write.

1. Load EEADR with the address.
2. Load EEDATA with the data to write.
3. Set the WREN bit to enable write access to the array.
4. Set the WR bit to initiate the erase cycle.

If the WR bit is not set in the instruction cycle after the WREN bit is set, the WREN bit will be cleared in hardware.

Sample code that follows this procedure is included in Example 5-3.

EXAMPLE 5-3: WRITING A FLASH DATA MEMORY ROW

```
BANKSEL   EEADR
MOVLW     EE_ADR_WRITE     ; LOAD ADDRESS
MOVWF     EEADR            ;
MOVLW     EE_DATA_TO_WRITE ; LOAD DATA
MOVWF     EEDATA           ; INTO EEDATA REGISTER
BSF       EECON,WREN       ; ENABLE WRITES
BSF       EECON,WR         ; INITITATE ERASE
```

Note 1: Only a series of BSF commands will work to enable the memory write sequence documented in Example 5-2. No other sequence of commands will work, no exceptions.

2: For reads, erases and writes to the Flash data memory, there is no need to insert a NOP into the user code as is done on mid-range devices. The instruction immediately following the "BSF EECON, WR/RD" will be fetched and executed properly.

5.3 Write Verify

Depending on the application, good programming practice may dictate that data written to the Flash data memory be verified. Example 5-4 is an example of a write verify.

EXAMPLE 5-4: WRITE VERIFY OF DATA EEPROM

```
MOVF    EEDATA, W    ;EEDATA has not changed
                     ;from previous write
BSF     EECON, RD    ;Read the value written
XORWF   EEDATA, W    ;
BTFSS   STATUS, Z    ;Is data the same
GOTO    WRITE_ERR    ;No, handle error
                     ;Yes, continue
```

5.4 Code Protection

Code protection does not prevent the CPU from performing read or write operations on the Flash data memory. Refer to the code protection chapter for more information.

6.0 I/O PORT

As with any other register, the I/O register(s) 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 high-impedance) since the I/O control registers are all set.

6.1 GPIO

GPIO is an 8-bit I/O register. Only the low-order 6 bits are used (GP<5:0>). Bits 7 and 6 are unimplemented and read as '0's. Please note that GP3 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 a port read. Pins GP0, GP1, and GP3 can be configured with weak pull-ups and also for wake-up on change. The wake-up on change and weak pull-up functions are not pin select-able. If GP3/MCLR is configured as MCLR, weak pull-up is always on and wake-up on change for this pin is not enabled.

6.2 TRIS Registers

The Output Driver Control registers are loaded with the contents of the W register by executing the TRIS f instruction. A '1' from a TRISGPIO register bit puts the corresponding output driver in a high-impedance (Input) mode. A '0' puts the contents of the output data latch on the selected pins, enabling the output buffer.

The TRISGPIO register is "write-only". Bits <5:0> are set (output drivers disabled) upon Reset.

Note:	If the T0CS bit is set to '1', it will override the TRISGPIO function on the T0CKI pin.

TABLE 6-1: WEAK PULL-UP ENABLED PINS

Pin	WPU	WU
GP0	Y	Y
GP1	Y	Y
GP2	N	N
GP3	Y(1)	Y
GP4	N	N
GP5	N	N

Note 1: When MCLRE = 1, the weak pull-up on GP3/MCLR is always enabled.

2: WPU = Weak pull-up; WU = Wake-up.

REGISTER 6-1: GPIO: GPIO REGISTER

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	GP5	GP4	GP3	GP2	GP1	GP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **GP<5:0>**: GPIO I/O Pin bits
1 = GPIO pin is >VIH min.
0 = GPIO pin is <VIL max.

REGISTER 6-2: TRISGPIO: TRI-STATE GPIO REGISTER

U-0	U-0	W-1	W-1	W-1	W-1	W-1	W-1
—	—	TRISGPIO5	TRISGPIO4	TRISGPIO3	TRISGPIO2	TRISGPIO1	TRISGPIO0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

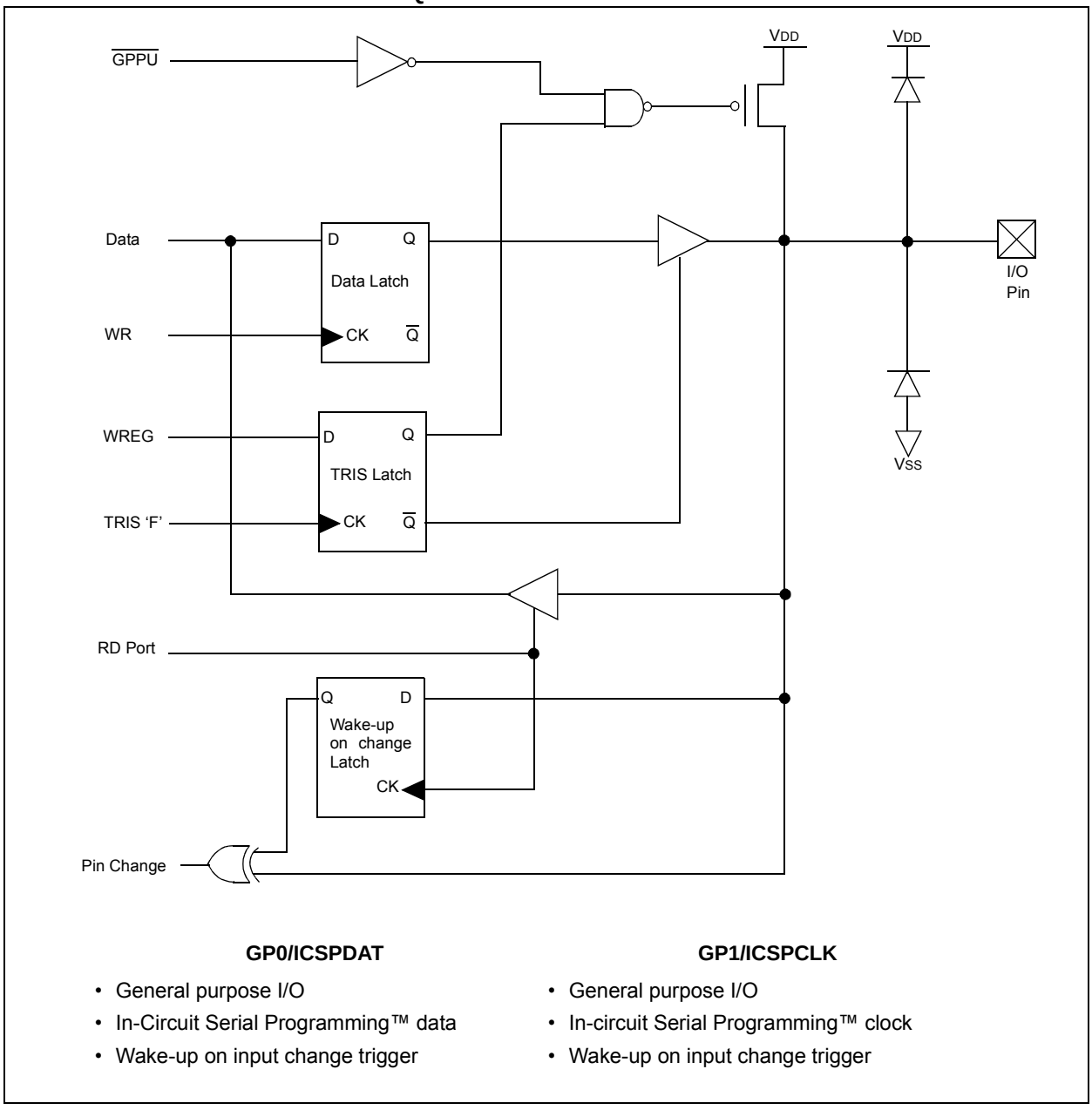
bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **TRISGPIO<5:0>:** GPIO Tri-State Control bits
1 = GPIO pin configured as an input (tri-stated)
0 = GPIO pin configured as an output

6.3 I/O Interfacing

The equivalent circuit for an I/O port pin is shown in Figure 6-1. All port pins, except GP3 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 GPIO, 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 TRISGPIO must be cleared (= `0`). For use as an input, the corresponding TRISGPIO bit must be set. Any I/O pin (except GP3) can be programmed individually as input or output.

FIGURE 6-1: PIC12F529T48A EQUIVALENT CIRCUIT FOR I/O PINS – GP0/GP1



GP0/ICSPDAT

- General purpose I/O
- In-Circuit Serial Programming™ data
- Wake-up on input change trigger

GP1/ICSPCLK

- General purpose I/O
- In-circuit Serial Programming™ clock
- Wake-up on input change trigger

FIGURE 6-2: GP2/TOCK1

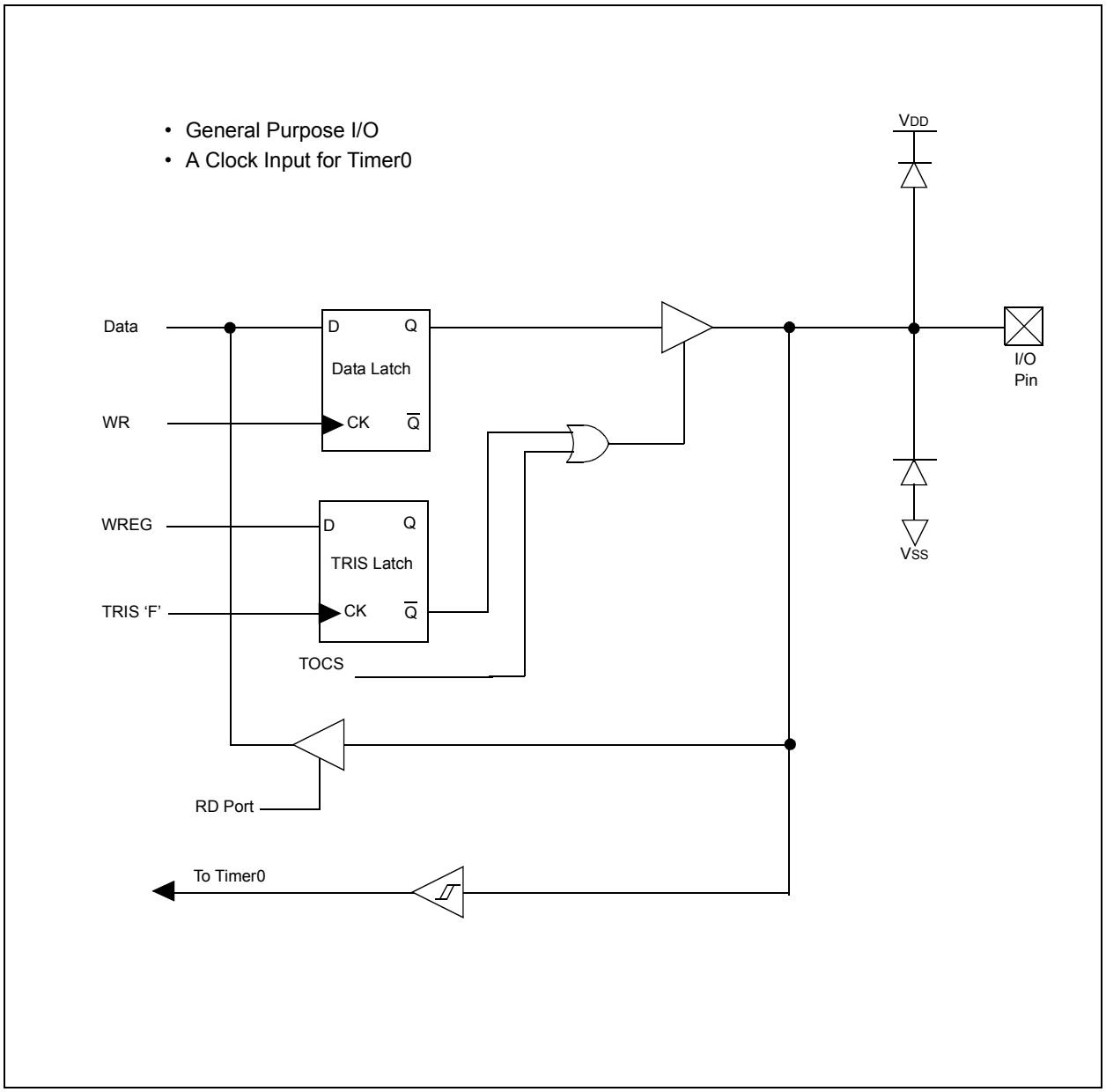


FIGURE 6-3: GP4/OSC2

- General Purpose I/O
- A crystal resonator connection

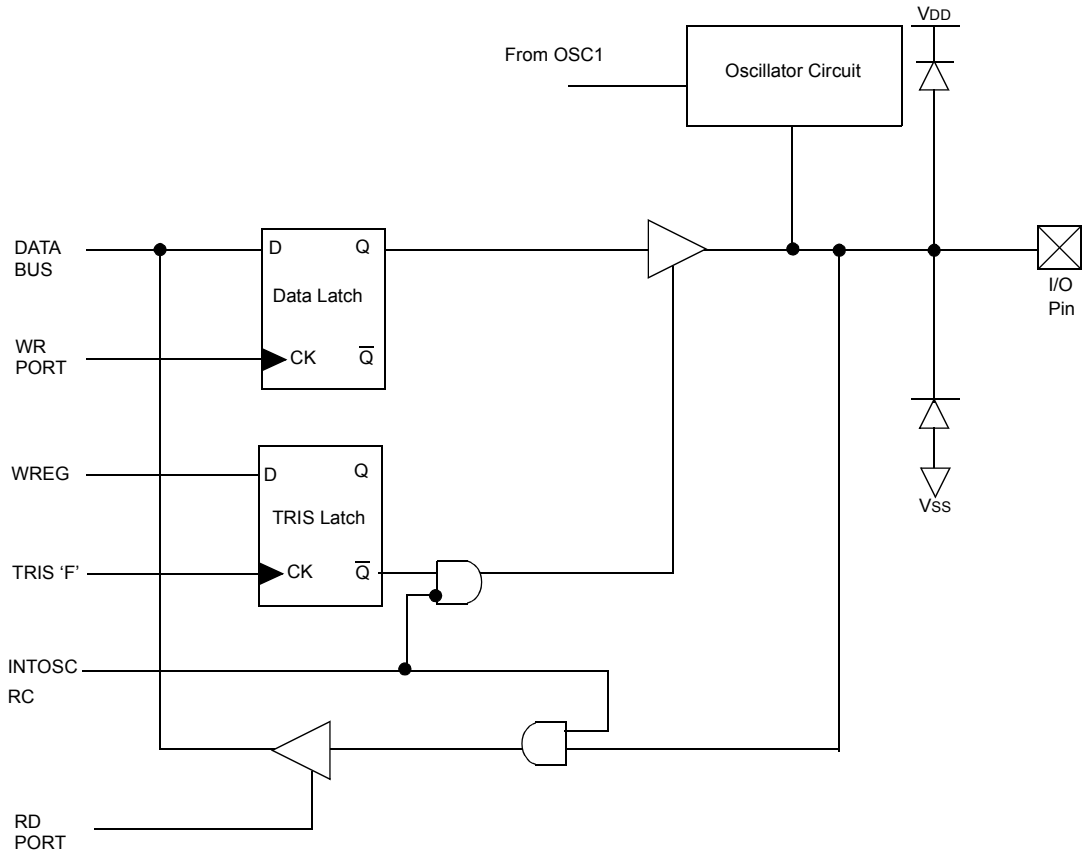


FIGURE 6-4: GP5/OSC1/CLKIN

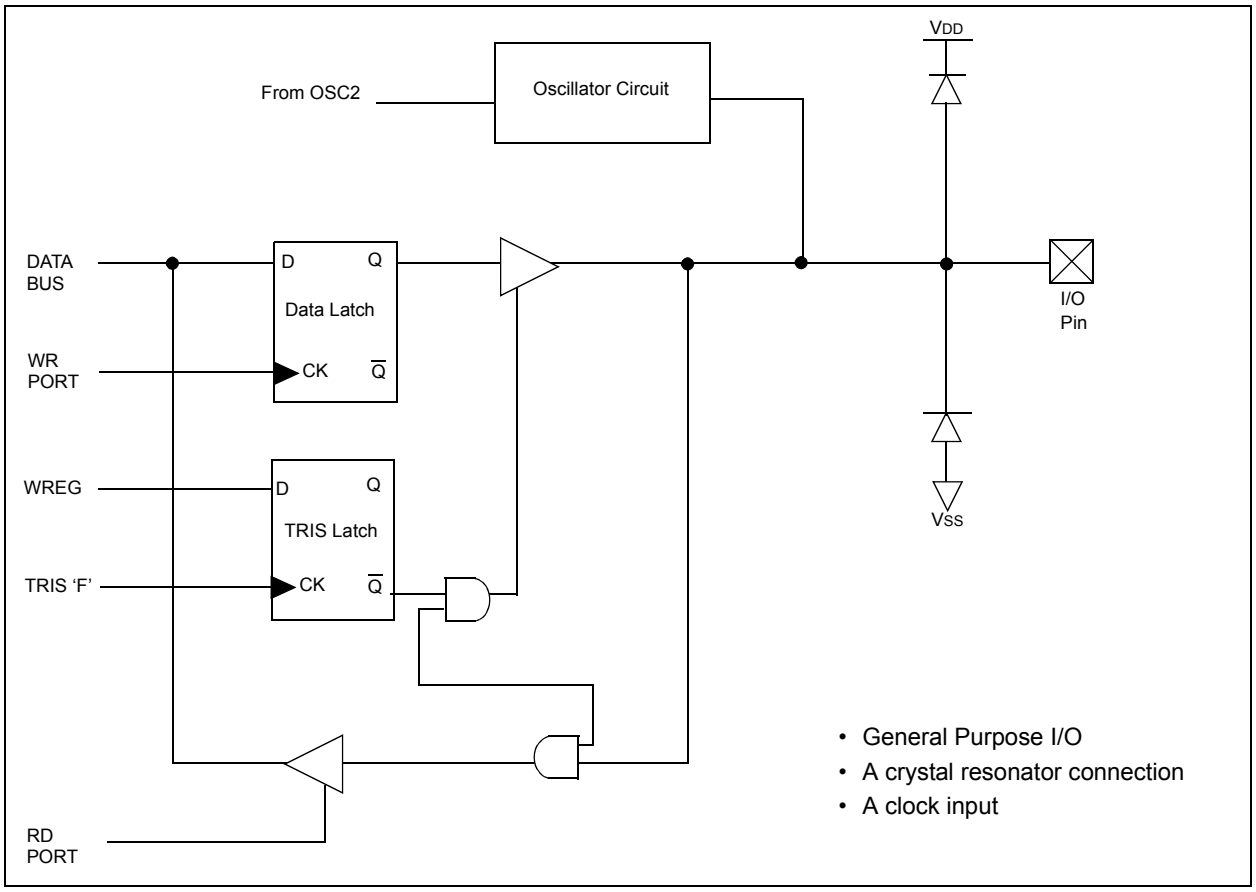
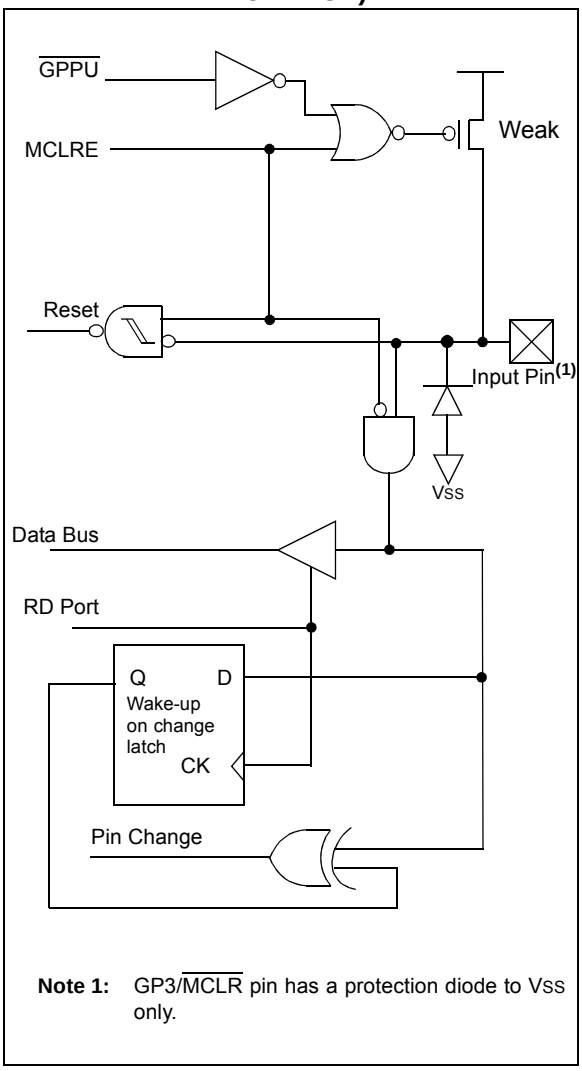


FIGURE 6-5: GP3 (WITH WEAK PULL-UP AND WAKE-UP ON CHANGE)



Note 1: GP3/$\overline{MCLR}$ pin has a protection diode to VSS only.

TABLE 6-2: SUMMARY OF PORT REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
GPIO	—	—	GP5	GP4	GP3	GP2	GP1	GP0	26
TRISGPIO	—	—	TRISGPIO5	TRISGPIO4	TRISGPIO3	TRISGPIO2	TRISGPIO1	TRISGPIO0	26
STATUS	GPWUF	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	18
OPTION	$\overline{GPWU}$	$\overline{GPPU}$	T0CS	T0SE	PSA	PS2	PS1	PS0	19

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0', Shaded cells = unimplemented, read as '0', q = depends on the condition

6.4 I/O Programming Considerations

6.4.1 BIDIRECTIONAL I/O PORTS

Some instructions operate internally as read followed by write operations. The BCF and BSF instructions, for example, read the entire port into the CPU, execute the bit operation and re-write the result. Caution must be used when these instructions are applied to a port where one or more pins are used as input/outputs. For example, a BSF operation on bit 5 of GPIO will cause all eight bits of GPIO to be read into the CPU, bit 5 to be set and the GPIO value to be written to the output latches. If another bit of GPIO is used as a bidirectional I/O pin (say bit 0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the Input mode, no problem occurs. However, if bit 0 is switched into Output mode later on, the content of the data latch may now be unknown.

Example 6-1 shows the effect of two sequential Read-Modify-Write instructions (e.g., BCF, BSF, etc.) on an I/O port.

A pin actively outputting a high or a low should not be driven from external devices at the same time in order to change the level on this pin ("wired OR", "wired AND"). The resulting high output currents may damage the chip.

EXAMPLE 6-1: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

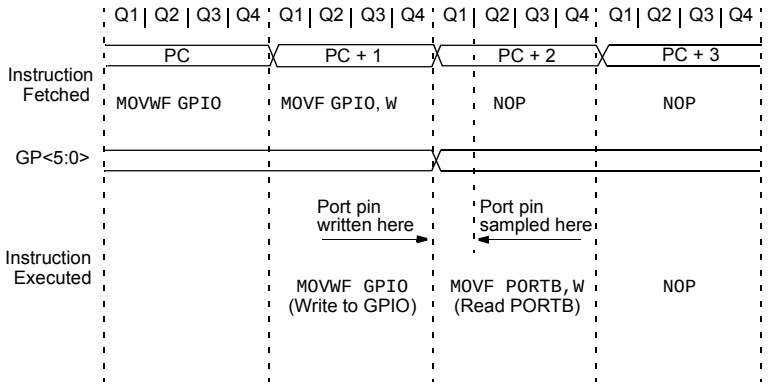
```
;Initial GPIO Settings
;GPIO<5:3> Inputs
;GPIO<2:0> Outputs
;
;                 GPIO latch   GPIO pins
;                 ----------   ----------
  BCF    GPIO, 5  ;--01 -ppp    --11 pppp
  BCF    GPIO, 4  ;--10 -ppp    --11 pppp
  MOVLW  007h;
  TRIS   GPIO     ;--10 -ppp    --11 pppp
;
```

Note 1: The user may have expected the pin values to be '--00 pppp'. The 2nd BCF caused GP5 to be latched as the pin value (High).

6.4.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 6-6). Therefore, care must be exercised if a write, followed by a read operation, is carried out on the same I/O port. The sequence of instructions should allow the pin voltage to stabilize (load dependent) before the next instruction causes that file to be read into the CPU. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a NOP or another instruction not accessing this I/O port.

FIGURE 6-6: SUCCESSIVE I/O OPERATION



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

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.

7.0 TIMER0 MODULE AND TMR0 REGISTER

The Timer0 module has the following features:

- 8-bit timer/counter register, TMR0
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select:
  - Edge select for external clock

Figure 7-1 is a simplified block diagram of the Timer0 module.

Timer mode is selected by clearing the T0CS bit (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 cycles (Figure 7-2 and Figure 7-3). The user can work around this by writing an adjusted value to the TMR0 register.

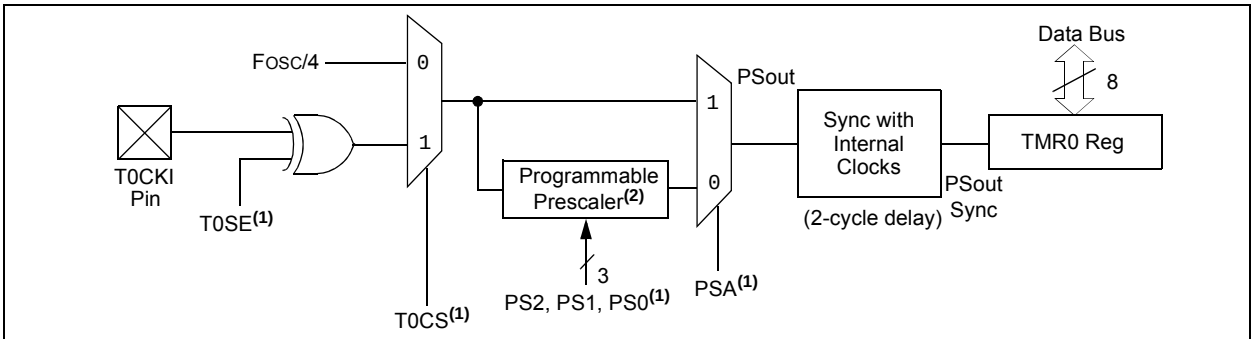
Counter mode is selected by setting the T0CS bit (OPTION<5>). In this mode, Timer0 will increment either on every rising or falling edge of pin T0CKI. The T0SE bit (OPTION<4>) determines the source edge. Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed in detail in **Section 7.1 "Using Timer0 with an External Clock"**.

The prescaler may be used by either the Timer0 module or the Watchdog Timer, but not both. The prescaler assignment is controlled in software by the control bit, PSA (OPTION<3>). Clearing the PSA bit will assign the prescaler to Timer0. 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 7.2 "Prescaler"** details the operation of the prescaler.

A summary of registers associated with the Timer0 module is found in Table 7-1.

The Timer0 contained in the CPU core follows the standard baseline definition.

FIGURE 7-1: TIMER0 BLOCK DIAGRAM



Note 1: Bits T0CS, T0SE, PSA, PS2, PS1 and PS0 are located in the OPTION register.
2: The prescaler is shared with the Watchdog Timer.

FIGURE 7-2: TIMER0 TIMING: INTERNAL CLOCK/NO PRESCALE

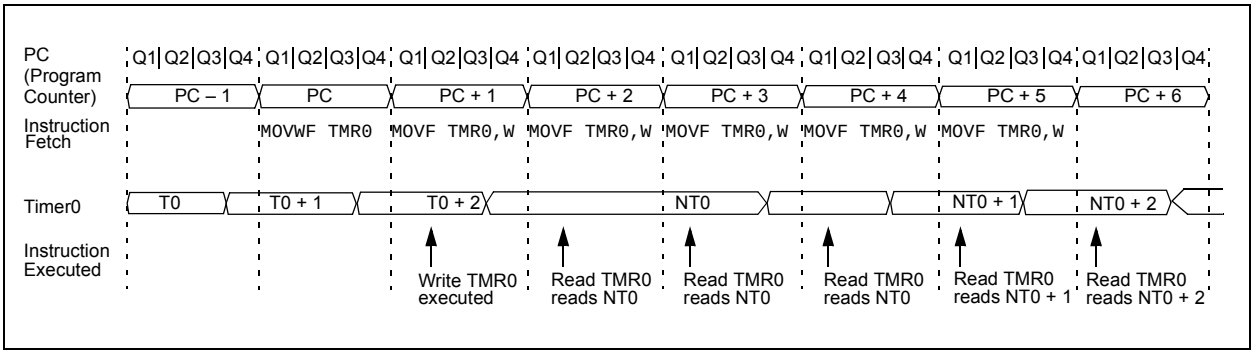


FIGURE 7-3: TIMER0 TIMING: INTERNAL CLOCK/PRESCALE 1:2

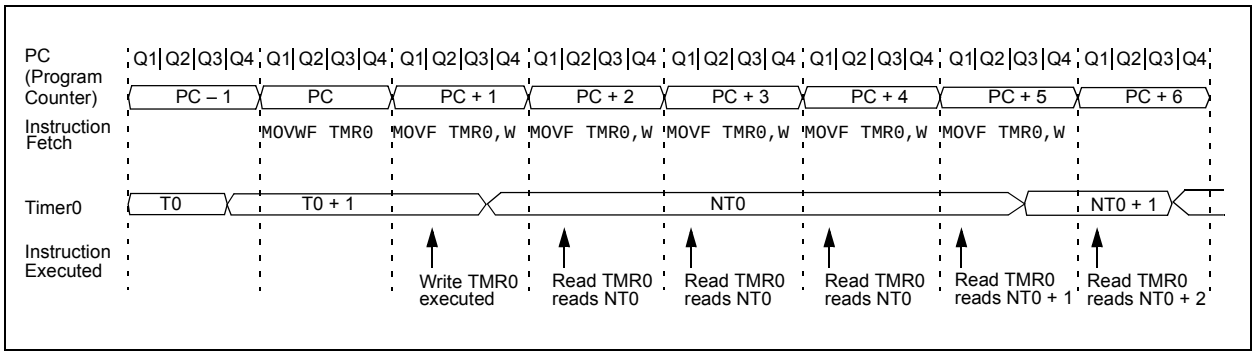


TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
01h	TMR0	Timer0 – 8-bit Real-Time Clock/Counter								33*
N/A	OPTION	$\overline{\text{GPWU}}$	$\overline{\text{GPPU}}$	T0CS	T0SE	PSA	PS2	PS1	PS0	19
N/A	TRISGPIO	—	—	TRISGPIO5	TRISGPIO4	TRISGPIO3	TRISGPIO2	TRISGPIO1	TRISGPIO0	26

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0', Shaded cells = unimplemented, read as '0'

* Page provides register information.

7.1 Using Timer0 with an External Clock

When an external clock input is used for Timer0, it must meet certain requirements. The external clock requirement is due to internal phase clock (TOSC) synchronization. Also, there is a delay in the actual incrementing of Timer0 after synchronization.

7.1.1 EXTERNAL CLOCK SYNCHRONIZATION

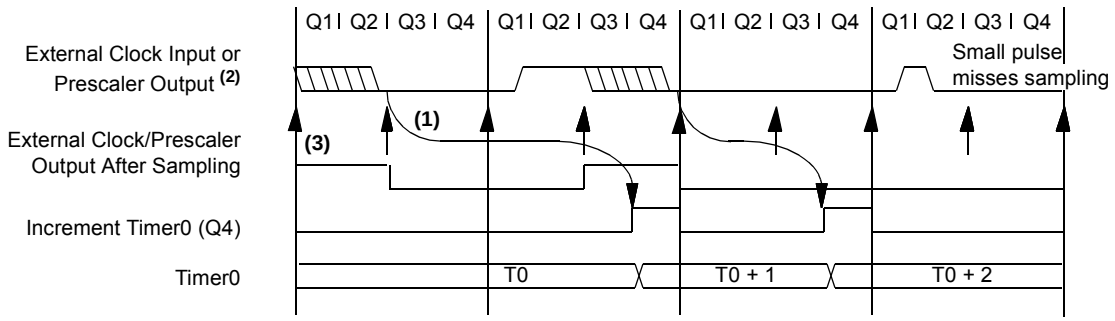
When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 7-4). Therefore, it is necessary for T0CKI to be high for at least two TOSC (and a small RC delay of two Tt0H) and low for at least two TOSC (and a small RC delay of two Tt0H). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple counter-type prescaler, so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least four TOSC (and a small RC delay of four Tt0H) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of Tt0H. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.

7.1.2 TIMER0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the Timer0 module is actually incremented. Figure 7-4 shows the delay from the external clock edge to the timer incrementing.

FIGURE 7-4: TIMER0 TIMING WITH EXTERNAL CLOCK



Note 1: Delay from clock input change to Timer0 increment is three TOSC to seven TOSC. (Duration of Q = TOSC). Therefore, the error in measuring the interval between two edges on Timer0 input = ±4 TOSC max.

2: External clock if no prescaler selected; prescaler output otherwise.

3: The arrows indicate the times at which sampling occurs.

7.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module or as a postscaler for the Watchdog Timer (WDT), respectively (see **Section 8.6 "Watchdog Timer (WDT)"**). For simplicity, this counter is being referred to as "prescaler" throughout this data sheet.

Note:	The prescaler may be used by either the Timer0 module or the WDT, but not both. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the WDT and vice versa.

The PSA and PS<2:0> bits (OPTION<3:0>) determine prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF TMR0`, `MOVWF TMR0`, etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the WDT. The prescaler is neither readable nor writable. On a Reset, the prescaler contains all '0's.

7.2.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed "on-the-fly" during program execution). To avoid an unintended device Reset, the following instruction sequence (Example 7-1) must be executed when changing the prescaler assignment from Timer0 to the WDT.

EXAMPLE 7-1: CHANGING PRESCALER (TIMER0 → WDT)

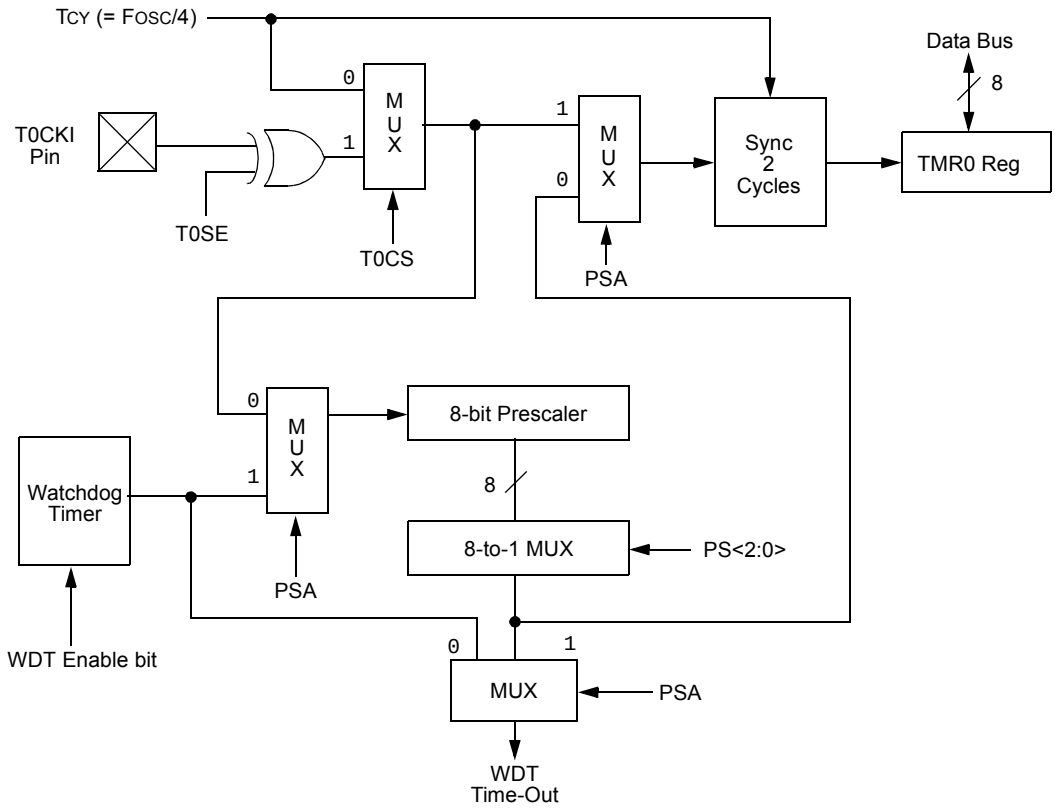
```
CLRWDT              ;Clear WDT
CLRF   TMR0         ;Clear TMR0 and Prescaler
MOVLW  b'00xx1111'
OPTION

CLRWDT              ;PS<2:0> are 000 or 001
MOVLW  b'00xx1xxx'  ;Set Postscaler to
OPTION              ;desired WDT rate
```

To change the prescaler from the WDT to the Timer0 module, use the sequence shown in Example 7-2. This sequence must be used even if the WDT is disabled. A `CLRWDT` instruction should be executed before switching the prescaler.

EXAMPLE 7-2: CHANGING PRESCALER (WDT → TIMER0)

```
CLRWDT              ;Clear WDT and
                    ;prescaler
MOVLW  b'xxxx0xxx'  ;Select TMR0, new
                    ;prescale value and
                    ;clock source
OPTION
```


FIGURE 7-5: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER(1)



Note 1: T0CS, T0SE, PSA, PS<2:0> are bits in the OPTION register.

NOTES:

8.0 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 PIC12F529T48A microcontroller 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)
  - Device Reset Timer (DRT)
  - Wake-up from Sleep on Pin Change
- Watchdog Timer (WDT)
- Sleep
- Code Protection
- ID Locations
- In-Circuit Serial Programming™

The PIC12F529T48A device has a Watchdog Timer, which can be shut off only through Configuration bit WDTE. It runs off of its own RC oscillator for added reliability. If using XT or LP selectable oscillator options, there is always an 18 ms (nominal) delay provided by the Device Reset Timer (DRT), intended to keep the chip in Reset until the crystal oscillator is stable. If using INTRC or EXTRC, the DRT provides a 1 ms (nominal) delay.

The Sleep mode is designed to offer a very low-current Power-Down mode. The user can wake-up the device from Sleep through a change-on-input-pin or through a Watchdog Timer time-out. Several oscillator options are also made available to allow the part to fit the application, including an internal 4 MHz or 8 MHz oscillator. The EXTRC oscillator option saves system cost while the LP crystal option saves power. A set of Configuration bits are used to select various options.

8.1 Configuration Bits

The PIC12F529T48A Configuration Words consist of 12 bits. Configuration bits can be programmed to select various device configurations. Two bits are for the selection of the oscillator type; one bit is the Watchdog Timer enable bit, one bit is the $\overline{\text{MCLR}}$ enable bit and six bits are for code protection (Register 8-1).

REGISTER 8-1: CONFIG: CONFIGURATION WORD REGISTER(1)

U-1	P-1	P-1	P-1	P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	CP3	CP2	CP1	CP0	$\overline{\text{CPDF}}$	IOSCFS	MCLRE	CP	WDTE	FOSC1	FOSC0
bit 11											bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 11 **Unimplemented**: Read as '1'

bit 10-7 **CP<3:0>:** Enhanced Code Protect bits
1011 = Code protect disabled
0010 = Code protect enabled
All others = Memory access disabled(3)

bit 6 **$\overline{\text{CPDF}}$:** Code Protection bit – Flash Data Memory
1 = Code protection off
0 = Code protection on

bit 5 **IOSCFS:** Internal Oscillator Frequency Select bit
1 = 8 MHz INTOSC speed
0 = 4 MHz INTOSC speed

bit 4 **MCLRE:** Master Clear Enable bit
1 = GP3/$\overline{\text{MCLR}}$ pin functions as $\overline{\text{MCLR}}$
0 = GP3/$\overline{\text{MCLR}}$ pin functions as GP3, $\overline{\text{MCLR}}$ internally tied to VDD

bit 3 **CP:** Configuration Word Parity bit(4)
1 = Parity bit set
0 = Parity bit clear

bit 2 **WDTE:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 1-0 **FOSC<1:0>:** Oscillator Selection bits
00 = LP oscillator with 18 ms DRT(2)
01 = XT oscillator with 18 ms DRT(2)
10 = INTRC with 1 ms DRT(2)
11 = EXTRC with 1 ms DRT(2)

Note 1: Refer to the "PIC12F529T48A *Memory Programming Specification*", DS41619 to determine how to program/erase the Configuration Word.

2: DRT length (18 ms or 1 ms) is a function of clock mode selection. It is the responsibility of the application designer to ensure the use of either 18 ms (nominal) DRT or the 1 ms (nominal) DRT will result in acceptable operation. Refer to Figure 12-1 for VDD rise time and stability requirements for this mode of operation.

3: See **Section 8.9 "Program Verification/Code Protection"**.

4: Set or clear to create odd parity with Configuration Word excluding CP<3:0>.

8.2 Oscillator Configurations

8.2.1 OSCILLATOR TYPES

The PIC12F529T48A device can be operated in up to four different oscillator modes. The user can program using the Configuration bits (FOSC<1:0>), to select one of these modes:

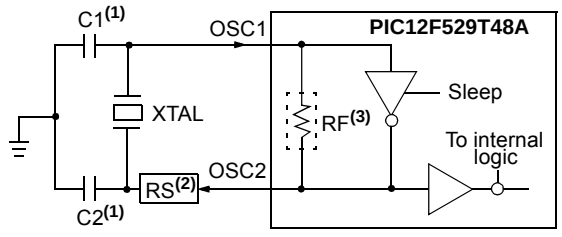
- LP: Low-Power Crystal
- XT: Crystal/Resonator
- INTRC: Internal 4 MHz or 8 MHz Oscillator
- EXTRC: External Resistor/Capacitor

8.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT or LP modes, a crystal or ceramic resonator is connected to the (GP5)/OSC1/(CLKIN) and (GP4)/OSC2 pins to establish oscillation (Figure 8-1). The PIC12F529T48A oscillator designs require 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 or LP modes, the device can have an external clock source drive the (GP5)/OSC1/CLKIN pin (Figure 8-2). When the part is used in this fashion, the output drive levels on the OSC2 pin are very weak. This pin should be left open and unloaded. Also when using this mode, the external clock should observe the frequency limits for the clock mode chosen (XT or LP).

Note 1: The user should verify that the device oscillator starts and performs as expected. Adjusting the loading capacitor values and/or the Oscillator mode may be required.

FIGURE 8-1: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (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 approx. value = 10 MΩ.

FIGURE 8-2: EXTERNAL CLOCK INPUT OPERATION (XT OR LP OSC CONFIGURATION)

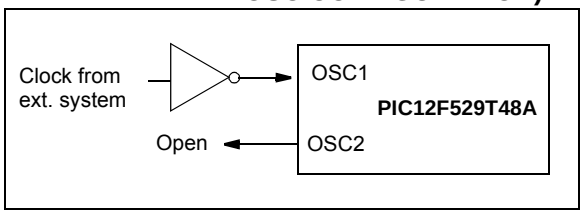


TABLE 8-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS (1)

Osc. Type	Resonator Freq.	Cap. Range C1	Cap. Range C2
XT	4.0 MHz	30 pF	30 pF

Note 1: Component values shown 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 8-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR – PIC12F529T48A(2)

Osc. Type	Resonator Freq.	Cap.Range C1	Cap. Range C2
LP	32 kHz(1)	15 pF	15 pF
XT	200 kHz 1 MHz 4 MHz	47-68 pF 15 pF 15 pF	47-68 pF 15 pF 15 pF

Note 1: For V_{DD} > 4.5V, C1 = C2 ≈ 30 pF is recommended.
2: Component values shown are for design guidance only. Rs may be required 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.

8.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator or a simple oscillator circuit with TTL gates can be used as an external crystal oscillator circuit. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used: one with parallel resonance, or one with series resonance.

Figure 8-3 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometers bias the 74AS04 in the linear region. This circuit could be used for external oscillator designs.

FIGURE 8-3: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

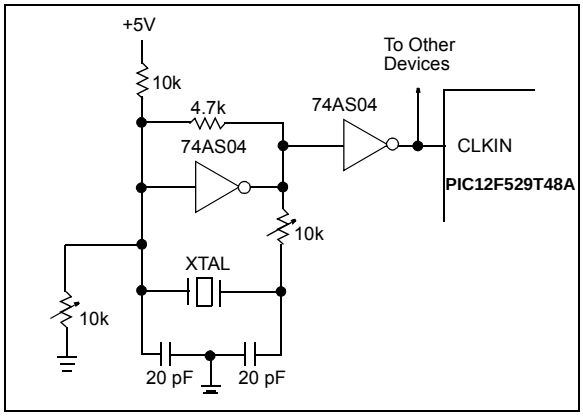
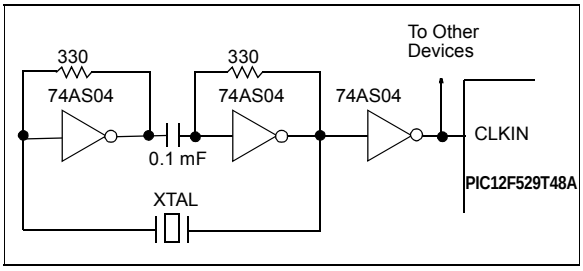


Figure 8-4 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330Ω resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 8-4: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



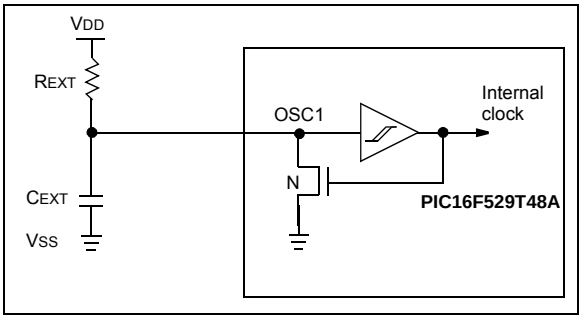
8.2.4 EXTERNAL RC OSCILLATOR

For timing insensitive applications, the RC circuit option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit-to-unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low CEXT values. The user also needs to take into account variation due to tolerance of external R and C components used.

Figure 8-5 shows how the R/C combination is connected to the PIC12F529T48A device. For REXT values below 3.0 kΩ, the oscillator operation may become unstable, or stop completely. For very high REXT values (e.g., 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. It is recommended keeping REXT between 5.0 kΩ and 100 kΩ.

Although the oscillator will operate with no external capacitor (CEXT = 0 pF), it is recommended using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance. See Figure 12-1 and Figure 12-2.

FIGURE 8-5: EXTERNAL RC OSCILLATOR MODE



8.2.5 INTERNAL 4/8 MHz RC OSCILLATOR

The internal RC oscillator provides a fixed 4/8 MHz (nominal) system clock at VDD = 3.5V and 25°C, (see **Section 12.0 "Electrical Characteristics"** for information on variation over voltage and temperature).

In addition, a calibration instruction is programmed into the last address of memory, which contains the calibration value for the internal RC oscillator. This location is always non-code-protected, regardless of the code-protect settings. This value is programmed as a MOVLW XX instruction where XX is the calibration value, and is placed at the Reset vector. This will load the W register with the calibration value upon Reset and the PC will then roll over to the users program at address 0x000. The user then has the option of writing the value to the OSCCAL Register (05h) or ignoring it.

OSCCAL, when written to with the calibration value, will "trim" the internal oscillator to remove process variation from the oscillator frequency.

Note:	Erasing the device will also erase the pre-programmed internal calibration value for the internal oscillator. The calibration value must be read prior to erasing the part so it can be reprogrammed correctly later.

For the PIC12F529T48A device, only bits <7:1> of OSCCAL are used for calibration. See Register 4-3 for more information.

Note:	The bit 0 of the OSCCAL register is unimplemented and should be written as '0' when modifying OSCCAL for compatibility with future devices.

8.3 Reset

The device differentiates between various kinds of Reset:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset during normal operation
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Time-out Reset during normal operation
- WDT Time-out Reset during Sleep
- Wake-up from Sleep on pin change

Some registers are not reset in any way, and they are unknown on Power-on Reset (POR) and unchanged in any other Reset. Most other registers are reset to "Reset state" on Power-on Reset (POR), $\overline{\text{MCLR}}$, WDT or Wake-up on pin change Reset during normal operation. They are not affected by a WDT Reset during Sleep or $\overline{\text{MCLR}}$ Reset during Sleep, since these Resets are viewed as resumption of normal operation.

TABLE 8-3: RESET CONDITIONS FOR REGISTERS

Register	Address	Power-on Reset	$\overline{\text{MCLR}}$ Reset, WDT Time-out, Wake-up On Pin Change
W	—	qqqq qqq0(1)	qqqq qqq0(1)
INDF	00h	xxxx xxxx	uuuu uuuu
TMR0	01h	xxxx xxxx	uuuu uuuu
PCL	02h	1111 1111	1111 1111
STATUS	03h	0001 1xxx	q00q quuu(2), (3)
FSR	04h	110x xxxx	11uu uuuu
OSCCAL	05h	1111 111-	uuuu uuu-
PORTB	06h	--xx xxxx	--uu uuuu
OPTION	—	1111 1111	1111 1111
TRIS	—	--11 1111	--11 1111
BSR	—	---- -000	---- -000
EECON	21h	---0 x000	---0 q000
EEDATA	25h	xxxx xxxx	uuuu uuuu
EEADR	26h	--xx xxxx	--uu uuuu

Legend: u = unchanged, x = unknown, – = unimplemented bit, read as '0', q = value depends on condition.

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

2: See Table 8-4 for Reset value for specific conditions.

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

TABLE 8-4: RESET CONDITION FOR SPECIAL REGISTERS

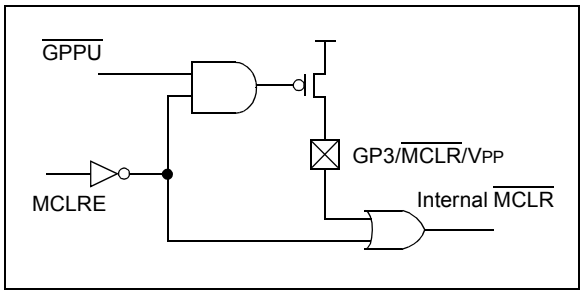
	STATUS Addr: 03h
Power-on Reset	0-01 1xxx
$\overline{MCLR}$ Reset during normal operation	0-0u uuuu
$\overline{MCLR}$ Reset during Sleep	0-01 0uuu
WDT Reset during Sleep	0-00 0uuu
WDT Reset normal operation	0-00 uuuu
Wake-up from Sleep on pin change	1-01 0uuu

Legend: u = unchanged, x = unknown

8.3.1 $\overline{MCLR}$ ENABLE

This Configuration bit, when unprogrammed (left in the '1' state), enables the external $\overline{MCLR}$ function. When programmed, the $\overline{MCLR}$ function is tied to the internal VDD and the pin is assigned to be a I/O. See Figure 8-6.

FIGURE 8-6: $\overline{MCLR}$ SELECT



8.4 Power-on Reset (POR)

The PIC12F529T48A device incorporates an on-chip Power-on Reset (POR) circuitry, which provides an internal chip Reset for most power-up situations.

The on-chip POR circuit holds the chip in Reset until VDD has reached a high enough level for proper operation. To take advantage of the internal POR, program the GP3/$\overline{MCLR}$/VPP pin as $\overline{MCLR}$ and tie through a resistor to VDD, or program the pin as GP3, in which case, an internal weak pull-up resistor is implemented using a transistor (refer to Table 12-3 for the pull-up resistor ranges). This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for VDD is specified. See **Section 12.0 "Electrical Characteristics"** for details.

When the devices start normal operation (exit the Reset condition), device operating parameters (voltage, frequency, temperature,...) must be met to ensure operation. If these conditions are not met, the devices must be held in Reset until the operating parameters are met.

A simplified block diagram of the on-chip Power-on Reset circuit is shown in Figure 8-7.

The Power-on Reset circuit and the Device Reset Timer (see **Section 8.5 "Device Reset Timer (DRT)"**) circuit are closely related. On power-up, the Reset latch is set and the DRT is reset. The DRT timer begins counting once it detects $\overline{MCLR}$ to be high. After the time-out period, which is typically 18 ms or 1 ms, it will reset the Reset latch and thus end the on-chip Reset signal.

A power-up example where $\overline{MCLR}$ is held low is shown in Figure 8-8. VDD is allowed to rise and stabilize before bringing $\overline{MCLR}$ high. The chip will actually come out of Reset TDRT after $\overline{MCLR}$ goes high.

In Figure 8-9, the on-chip Power-on Reset feature is being used ($\overline{MCLR}$ and VDD are tied together or the pin is programmed to be GP3). The VDD is stable before the Start-up timer times out and there is no problem in getting a proper Reset. However, Figure 8-10 depicts a problem situation where VDD rises too slowly. The time between when the DRT senses that $\overline{MCLR}$ is high and when $\overline{MCLR}$ and VDD actually reach their full value, is too long. In this situation, when the start-up timer times out, VDD has not reached the VDD (min) value and the chip may not function correctly. For such situations, we recommend that external RC circuits be used to achieve longer POR delay times (Figure 8-9).

Note:	When the devices start normal operation (exit the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) 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 AN522, "*Power-Up Considerations*" (DS00522).

FIGURE 8-7: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT

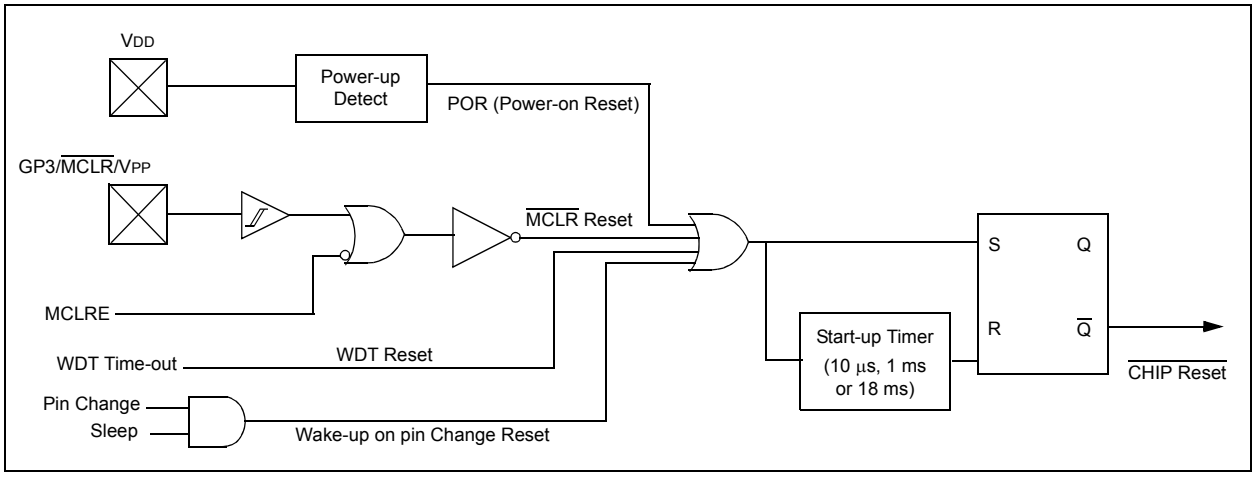


FIGURE 8-8: TIME-OUT SEQUENCE ON POWER-UP (MCLR PULLED LOW)

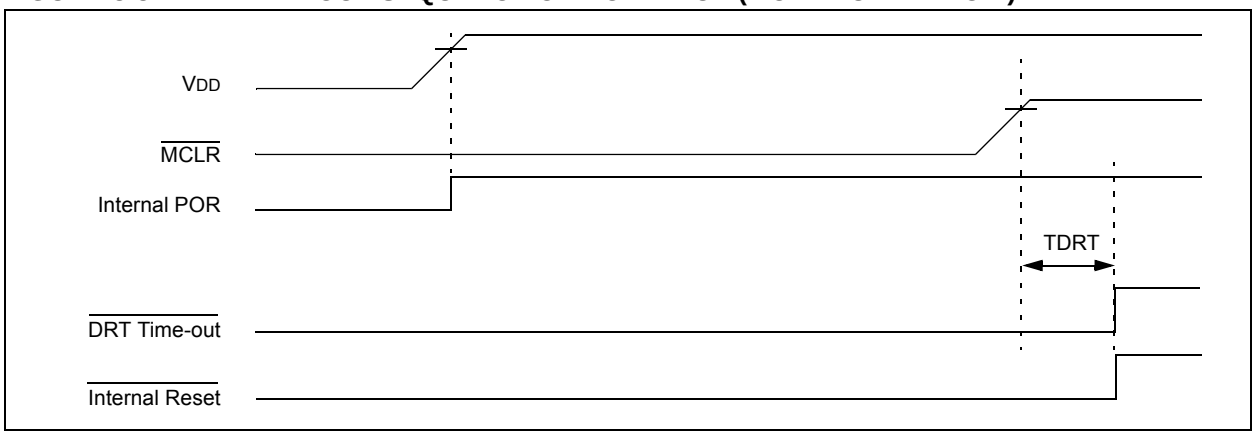


FIGURE 8-9: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD): FAST VDD RISE TIME

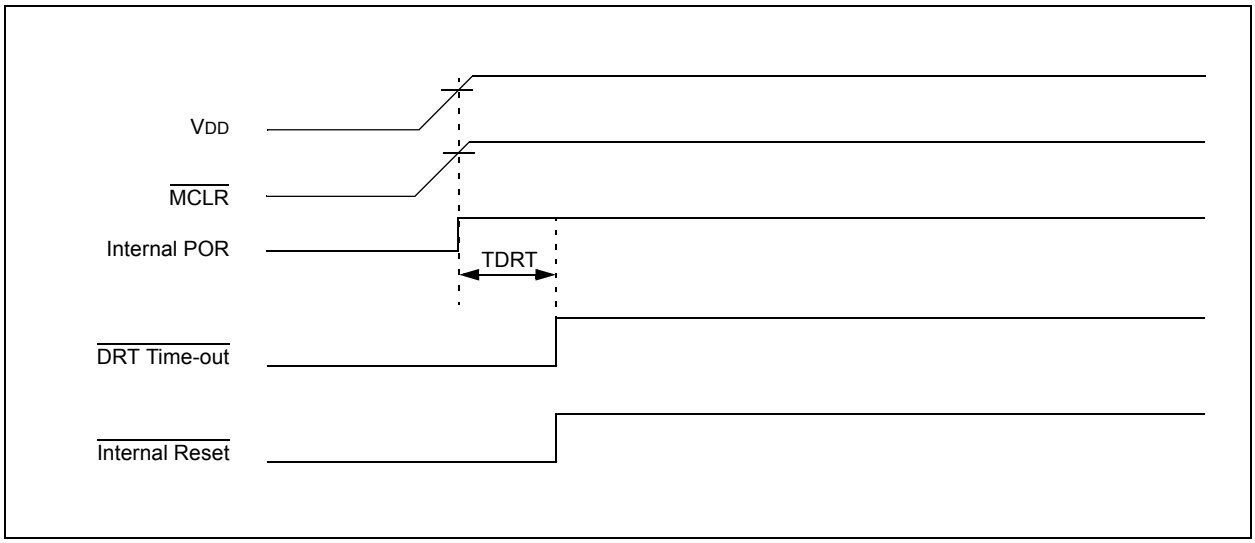
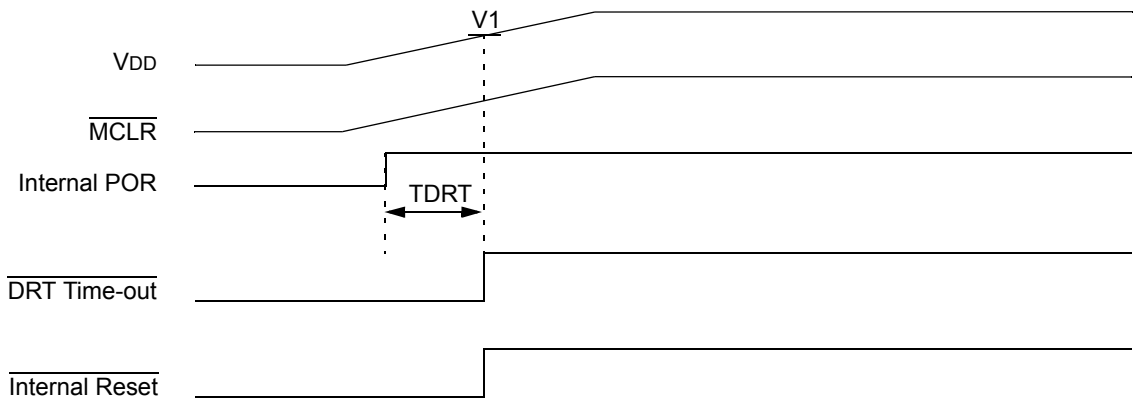


FIGURE 8-10: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO V_{DD}): SLOW V_{DD} RISE TIME



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 ≥ V_{DD} min.

8.5 Device Reset Timer (DRT)

On the PIC12F529T48A device, 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 8-5).

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 VDD to rise above VDD 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 devices in a Reset condition after MCLR has reached a logic high (VIH MCLR) level. Programming GP3/MCLR/VPP as MCLR and using an external RC network connected to the MCLR input is not required in most cases. This allows savings in cost-sensitive and/or space restricted applications, as well as allowing the use of the GP3/MCLR/VPP pin as a general purpose input.

The Device Reset Time delays will vary from chip-to-chip due to VDD, temperature and process variation. See AC parameters for details.

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

Reset sources are POR, MCLR, WDT time-out and wake-up on pin change. See **Section 8.8.2 "Wake-up from Sleep"**, Notes 1, 2 and 3.

TABLE 8-5: DRT (DEVICE RESET TIMER PERIOD)

Oscillator Configuration	POR Reset	Subsequent Resets
INTOSC, EXTRC	1 ms (typical)	10 μs (typical)
LP, XT	18 ms (typical)	18 ms (typical)

8.6 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 (GP5)/OSC1/CLKIN pin and the internal 4 or 8 MHz oscillator. This 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 TO bit (STATUS<4>) will be cleared upon a Watchdog Timer Reset.

The WDT can be permanently disabled by programming the configuration WDTE as a '0' (see **Section 8.1 "Configuration Bits"**). Refer to the PIC12F529T48A Programming Specification (DS41316) to determine how to access the Configuration Word.

8.6.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). If a longer time-out period is desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT (under software control) by writing to the OPTION register. Thus, a time-out period of a nominal 2.3 seconds can be realized. These periods vary with temperature, VDD and part-to-part process variations (see DC specs).

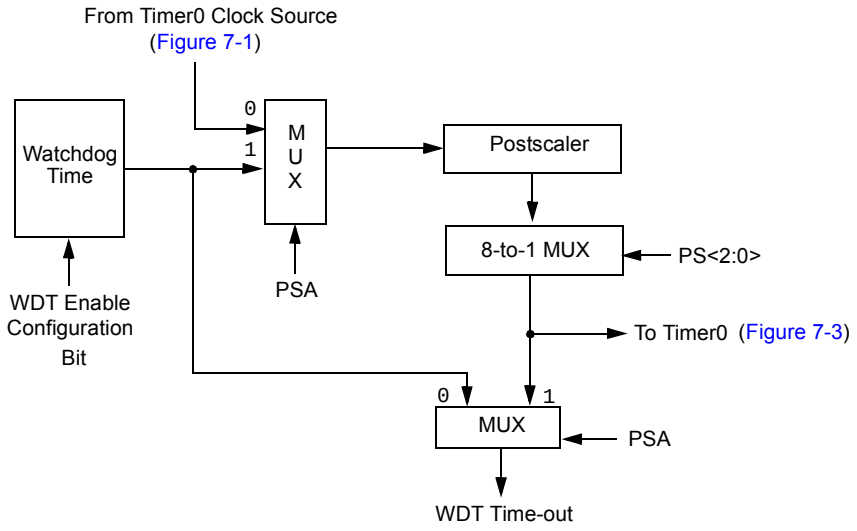
Under worst-case conditions (VDD = Min., Temperature = Max., max. WDT prescaler), it may take several seconds before a WDT time-out occurs.

8.6.2 WDT PROGRAMMING CONSIDERATIONS

The CLRWDT instruction clears the WDT and the postscaler, if assigned to the WDT, and prevents it from timing out and generating a device Reset.

The SLEEP instruction resets the WDT and the postscaler, if assigned to the WDT. This gives the maximum Sleep time before a WDT wake-up Reset.

FIGURE 8-11: WATCHDOG TIMER BLOCK DIAGRAM



Note 1: PSA, PS<2:0> are bits in the OPTION register.

TABLE 8-6: SUMMARY OF REGISTER ASSOCIATED WITH THE WATCHDOG TIMER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OPTION	$\overline{\text{GPWU}}$	$\overline{\text{GPPU}}$	T0CS	T0SE	PSA	PS2	PS1	PS0	19

Legend: Shaded boxes = Not used by Watchdog Timer.

8.7 Time-out Sequence, Power-down and Wake-up from Sleep Status Bits (TO, PD, GPWUF)

The $\overline{TO}$, $\overline{PD}$ and (GPWUF) bits in the STATUS register can be tested to determine if a Reset condition has been caused by a power-up condition, a $\overline{MCLR}$ or Watchdog Timer (WDT) Reset.

TABLE 8-7: $\overline{TO}$/$\overline{PD}$/(GPWUF) STATUS AFTER RESET(1)

GPWUF	$\overline{TO}$	$\overline{PD}$	Reset Caused By
0	0	0	WDT wake-up from Sleep
0	0	u	WDT time-out (not from Sleep)
0	1	0	$\overline{MCLR}$ wake-up from Sleep
0	1	1	Power-up
0	u	u	$\overline{MCLR}$ not during Sleep
1	1	0	Wake-up from Sleep on pin change

Legend: u = unchanged

Note 1: The $\overline{TO}$, $\overline{PD}$ and GPWUF bits maintain their status (u) until a Reset occurs. A low-pulse on the $\overline{MCLR}$ input does not change the $\overline{TO}$, $\overline{PD}$ and GPWUF Status bits.

8.8 Power-down Mode (Sleep)

A device may be powered down (Sleep) and later powered up (wake-up from Sleep).

8.8.1 SLEEP

The Power-Down mode is entered by executing a SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{TO}$ bit (STATUS<4>) is set, the $\overline{PD}$ bit (STATUS<3>) is cleared and the oscillator driver is turned off. The I/O ports maintain the status they had before the SLEEP instruction was executed (driving high, driving low or high-impedance).

Note: A Reset generated by a WDT time-out does not drive the $\overline{MCLR}$ pin low.

For lowest current consumption while powered down, the T0CKI input should be at V_{DD} or V_{SS} and the GP3/$\overline{MCLR}$/V_{PP} pin must be at a logic high level if $\overline{MCLR}$ is enabled.

8.8.2 WAKE-UP FROM SLEEP

The device can wake-up from Sleep through one of the following events:

1. An external Reset input on GP3/$\overline{MCLR}$/V_{PP} pin, when configured as $\overline{MCLR}$.
2. A Watchdog Timer Time-out Reset (if WDT was enabled).
3. A change on input pin GP0, GP1 and GP3 when wake-up on change is enabled.

These events cause a device Reset. The $\overline{TO}$, $\overline{PD}$ and GPWUF bits can be used to determine the cause of a device Reset. The $\overline{TO}$ bit is cleared if a WDT time-out occurred (and caused wake-up). The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked. The GPWUF bit indicates a change in state while in Sleep at pins GP0, GP1 and GP3 (since the last file or bit operation on GPIO port).

CAUTION

Right before entering Sleep, read the input pins. When in Sleep, wake-up occurs when the values at the pins change from the state they were in at the last reading. If a wake-up on change occurs and the pins are not read before re-entering Sleep, a wake-up will occur immediately even if no pins change while in Sleep mode.

The WDT is cleared when the device wakes from Sleep, regardless of the wake-up source.

8.9 Program Verification/Code Protection

Code protection is enabled or disabled by writing the correct value to the CP<3:0> bits of the Configuration register. These bits must be written every time the device is erased.

If the code protection bits have not been enabled, the on-chip program and data memory can be read out for verification purposes.

The last location (the oscillator calibration value) can be read, regardless of the setting of the program memory's code protection bit. If the code protect bit specific to the Flash data memory is programmed, then none of the contents of this memory region can be verified externally.

Refer to *PIC12F529T48A/T39A Memory Programming Specification* (DS41619) for more information on programming the Configuration Word.

Note:	The device code protection must be disabled before attempting to program Flash memory.

8.10 ID Locations

Four memory locations are designated as ID locations where users 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. The upper bits should be programmed as 0s.

8.11 In-Circuit Serial Programming™

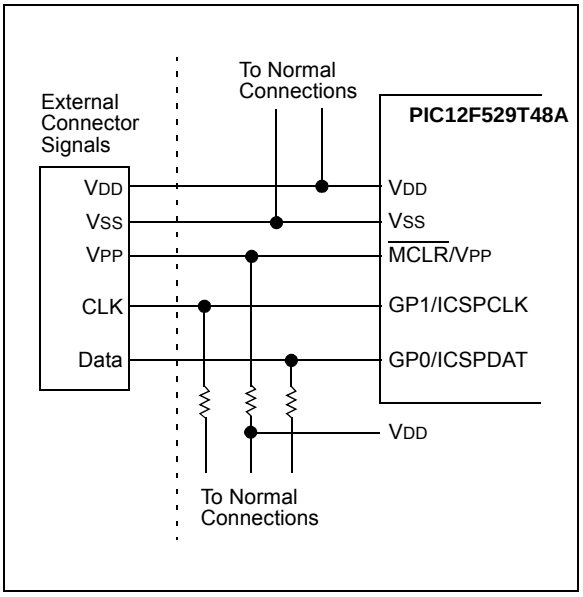
The PIC12F529T48A device can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground and the programming voltage. This allows users to manufacture boards with unprogrammed PIC12F529T48A device and then program the PIC12F529T48A device just before shipping the product. This also allows the most recent firmware, or a custom firmware, to be programmed.

The PIC12F529T48A device is placed into a Program/Verify mode by holding the GP1 and GP0 pins low while raising the $\overline{MCLR}$ (V_{PP}) pin from V_{IL} to V_{IHH} (see programming specification). The GP1 pin becomes the programming clock, and the GP0 pin becomes the programming data. Both GP1 and GP0 pins are Schmitt Trigger inputs in this mode.

After Reset, a 6-bit command is then supplied to the device. Depending on the command, 14 bits of program data are then supplied to or from the device, depending if the command was a Load or a Read. For complete details of serial programming, please refer to the "PIC12F529T48A/T39A *Memory Programming Specification*," (DS41619).

A typical In-Circuit Serial Programming connection is shown in Figure 8-12.

FIGURE 8-12: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION



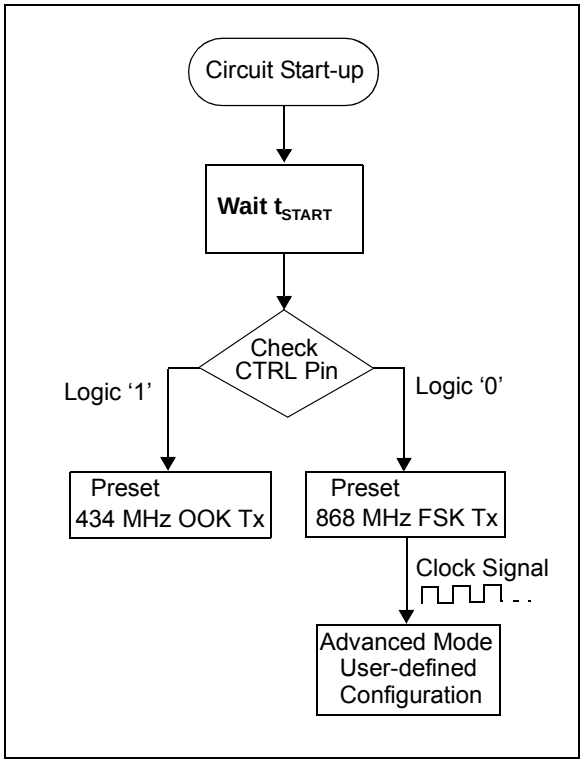
9.0 RF TRANSMITTER

The RF transmitter is a fully integrated transmitter, capable of Frequency-Shift Keying (FSK) and On-Off Keying (OOK) modulation of an input data stream. The transmitter is capable of operating in Preset or Advanced mode. The Preset mode configures the RF transmitter in one or two fixed configurations.

9.1 Mode Selection

Two modes of operation are supported: Preset and Advanced. Preset mode allows operation in one of two preconfigured modes. Advanced mode allows full configuration of all features of the transmitter. Mode selection is made by applying a logical '1' or '0' on the CTRL pin during power-up, followed by a clock signal on the DATA pin if Advanced mode is required.

FIGURE 9-1: MODE SELECTION



9.1.1 PRESET MODE

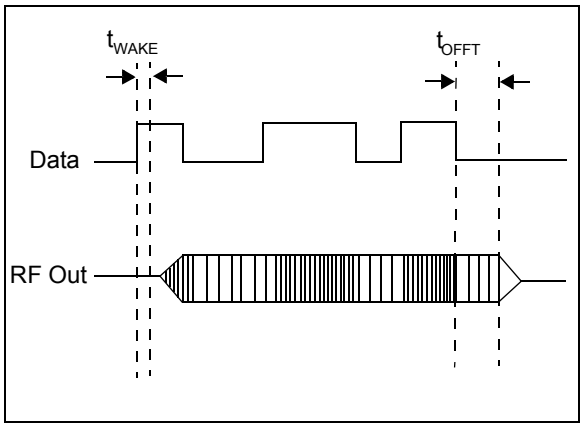
In Preset mode, the transmitter is configured according to Table 9-1. One of the two configuration modes can be selected by changing the logical state of the CTRL pin at power-up. The timing of a typical transmit operation in Preset mode is shown in Figure 9-2. A rising edge on the DATA pin activates the transmitter. DATA must be held high for the start-up time, TWAKE, while the transmitter transitions out of Sleep mode. Signals on the DATA pin, after the start-up time has elapsed, are then transmitted.

The transition back to Sleep mode is managed automatically. In 868 MHz FSK mode, the transmitter returns to Sleep mode after 2 ms of inactivity on the DATA line. In 434 MHz OOK mode, the transmitter returns to Sleep after 20 ms of inactivity.

TABLE 9-1: PRESET MODE CONFIGURATIONS

CTRL	Configuration
1	OOK 433.92 MHz, 10 dBm
0	FSK 868.3 MHz, FDEV = 20 kHz, 10 dBm

FIGURE 9-2: TRANSMITTER TIMING IN PRESET MODE



Note 1: While the logic level of the CTRL pin during TSTART does not have any effect on the device operation, the pin should not be connected to VDD through an impedance lower than 20 kΩ or higher than 1 MΩ.

9.1.2 ADVANCED MODE

Advanced mode allows full configuration of the transmitter by writing to the Configuration register. Writing and reading from this register is performed via a two-wire interface formed by the CTRL and DATA pins.

Advanced mode is enabled by applying a rising signal on the CTRL pin while driving DATA low. Upon detection of this rising edge, the data applied to the DATA pin is accepted as register configuration information. Data bits are clocked on subsequent rising edges of the clock signal. The first bit of serial data selects register read or write operation. The timing for module Configuration register 'write' is shown in Figure 9-3. When writing, all 13 data bits must be written to the register. Similarly, the Configuration register may be read using the timing of Figure 9-4. When reading, all 13 bits of configuration and all 43 bits of test data must be read.

The contents of the Configuration register are described in Table 9-2. During the register 'write' or 'read' phases, the transmitter remains in Sleep mode.

The CTRL pin is sampled after t_{START} has elapsed. The CTRL pin should not be allowed to float.

FIGURE 9-3: CONFIGURATION REGISTER 'WRITE'

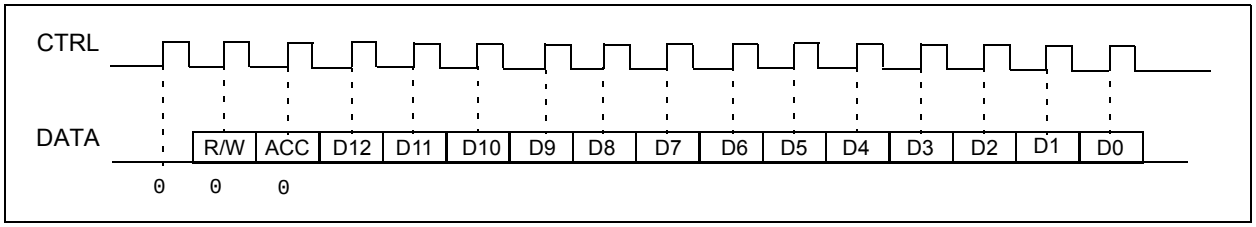
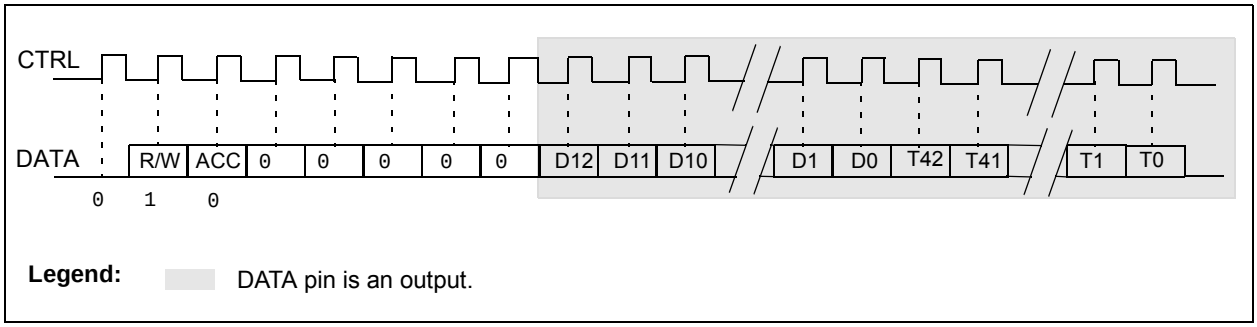
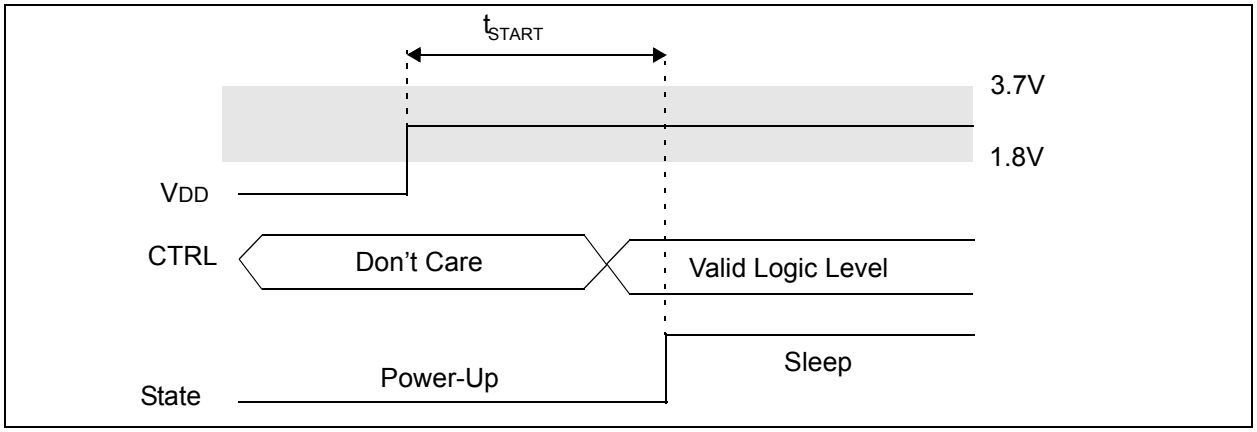


FIGURE 9-4: CONFIGURATION REGISTER 'READ'



Note 1: When reading the Configuration register, 64 clock cycles on the CTRL pin must be issued, shifting out on the DATA pin the 13 Configuration bits, plus 43 internal test bits. Reading only the 13 Configuration bits is not allowed.

FIGURE 9-5: POWER-UP TIMING



When operating in Advanced mode, two possibilities exist for operation of the transmitter, these are dependent upon the state of the TX mode bit (D12 of the register description in Table 9-2).

When set to logical '0', operation is identical to that of the Preset mode. Transmit operation will be in accordance with that of Figure 9-2 with the time TOFFT corresponding to that set in the Configuration register.

Note that prior to programming the default, logical '0', configuration is loaded. Note also that subsequent programming iterations can only be performed once the transmit cycle is finished – including the time required for switching off the PA (TOFFT).

With TX mode (D12) set, the transmitter is placed directly in Transmit mode. It will remain in Transmit mode until a second register write operation clears the TX mode bit. Refer to Figure 9-6.

Note 1:	Once in Sleep mode, activity on the DATA pin (without clocking of the CTRL line) will trigger transmission according to the current configuration settings. Care must be taken to avoid inadvertent transmissions.

FIGURE 9-6: MANUAL CONTROL OF TRANSMITTER

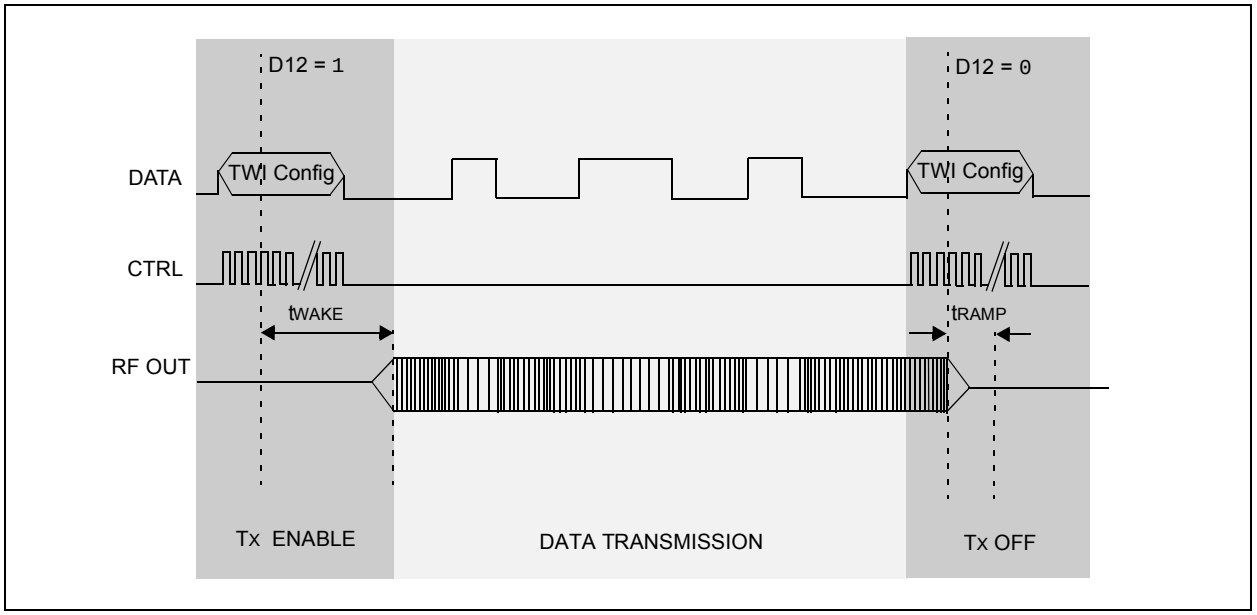


TABLE 9-2: CONFIGURATION REGISTER

Bit	Name	Value	Setting	Default		Notes
				CTRL = 0	CTRL = 1	
D12	Tx Mode	0	Preset	0	0	When set to '1', the transmitter will continuously transmit
		1	Forced Transmit			
D(11:9)	Frequency	000	418.00 MHz	100	010	RF operating center frequency
		001	433.42 MHz			
		010	433.92 MHz			
		011	864.00 MHz			
		100	868.30 MHz			
		101	868.65 MHz			
		110	868.95 MHz			
		111	869.85 MHz			
D8	Modulation	0	FSK	0	1	Modulation format
		1	OOK			
D(7:5)	Freq. Deviation	000	10 kHz	010	010	FSK frequency deviation (not used in OOK mode)
		001	12.5 kHz			
		010	20 kHz			
		011	25 kHz			
		100	40 kHz			
		101	50 kHz			
		110	80 kHz			
		111	100 kHz			
D4	RF Power	0	0 dBm	1	1	Programmed RF output power
		1	10 dBm			
D3	Tx Timer	0	2 ms	0	1	Transmit power-off timer, T_{OFFT}
		1	20 ms			
D(2:0)	Fine Tuning	011	fc + 6 * PLL Step	000	000	Fine tuning from programmed center frequency
		010	fc + 4 * PLL Step			
		001	fc + 2 * PLL Step			
		000	fc + 0 * PLL Step			
		111	fc - 2 * PLL Step			
		110	fc - 4 * PLL Step			
		101	fc - 6 * PLL Step			
		100	fc - 8 * PLL Step			

10.0 INSTRUCTION SET SUMMARY

The PIC12F529T48A instruction set is highly orthogonal and is comprised of three basic categories.

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC12F529T48A 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 formats for each of the categories is presented in Figure 10-1, while the various opcode fields are summarized in Table 10-1.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register 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 10-1: 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
$\overline{TO}$	Time-out bit
$\overline{PD}$	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 10-1 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 10-1: GENERAL FORMAT FOR INSTRUCTIONS

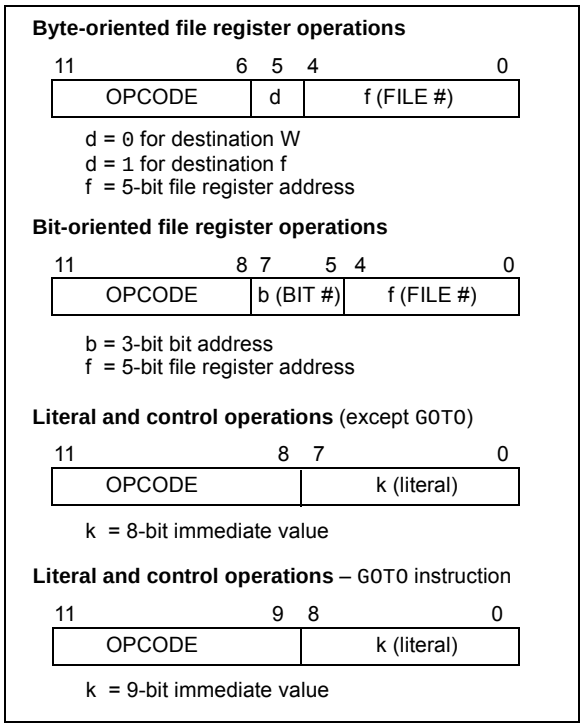


TABLE 10-2: INSTRUCTION SET SUMMARY

Mnemonic, Operands		Description	Cycles	12-Bit Opcode MSb	12-Bit Opcode 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	–	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	
MOVLB	k	Move literal to BSR	1	0000 0001 0kkk		None	
OPTION	–	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		$\overline{TO}$, $\overline{PD}$	**3**
TRISGPIO	f	Load TRISGPIO register	1	0000 0000 0fff		None	
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 **Section 4.6 "Program Counter"**.

2: When an I/O register is modified as a function of itself (e.g. MOVF GPIO, 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 = 6, causes the contents of the W register to be written to the tri-state latches of GPIO. A '1' forces the pin to a high-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).

ADDWF **Add W and f**

Syntax:	[*label*] ADDWF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(W) + (f) → (dest)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register and 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'.

ANDLW **AND literal with W**

Syntax:	[*label*] ANDLW k
Operands:	$0 \le k \le 255$
Operation:	(W).AND. (k) → (W)
Status Affected:	Z
Description:	The contents of the W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register.

ANDWF **AND W with f**

Syntax:	[*label*] ANDWF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(W) .AND. (f) → (dest)
Status Affected:	Z
Description:	The contents of the W register are AND'ed 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'.

BCF **Bit Clear f**

Syntax:	[*label*] BCF f,b
Operands:	$0 \le f \le 31$ $0 \le b \le 7$
Operation:	0 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

BSF **Bit Set f**

Syntax:	[*label*] BSF f,b
Operands:	$0 \le f \le 31$ $0 \le b \le 7$
Operation:	1 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

BTFSC **Bit Test f, Skip if Clear**

Syntax:	[*label*] BTFSC f,b
Operands:	$0 \le f \le 31$ $0 \le b \le 7$
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', then the next instruction is skipped. If bit 'b' is '0', then the next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead, making this a two-cycle instruction.

BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	$0 \le f \le 31$ $0 \le b < 7$
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and a NOP is executed instead, making this a two-cycle instruction.

CALL	Subroutine Call
Syntax:	[*label*] CALL k
Operands:	$0 \le k \le 255$
Operation:	(PC) + 1→ Top-of-Stack; k → PC<7:0>; (STATUS<6:5>) → PC<10:9>; 0 → PC<8>
Status Affected:	None
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.

CLRF	Clear f
Syntax:	[*label*] CLRF f
Operands:	$0 \le f \le 31$
Operation:	00h → (f); 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

CLRW	Clear W
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W); 1 → Z
Status Affected:	Z
Description:	The W register is cleared. Zero bit (Z) is set.

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:	$\overline{TO}$, $\overline{PD}$
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.

COMF	Complement f
Syntax:	[*label*] COMF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	$(\bar{f}) \rightarrow$ (dest)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

DECF	Decrement f
Syntax:	[*label*] DECF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(f) – 1 → (dest)
Status Affected:	Z
Description:	Decrement 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’.

DECFSZ	Decrement f, Skip if 0
Syntax:	[*label*] DECFSZ f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(f) – 1 → d; skip if result = 0
Status Affected:	None
Description:	The contents of register ‘f’ are decremented. 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 and a NOP is executed instead making it a two-cycle instruction.

GOTO	Unconditional Branch
Syntax:	[*label*] GOTO k
Operands:	$0 \le k \le 511$
Operation:	k → PC<8:0>; STATUS<6:5> → PC<10:9>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 9-bit immediate value is loaded into PC bits <8:0>. The upper bits of PC are loaded from STATUS<6:5>. GOTO is a two-cycle instruction.

INCF	Increment f
Syntax:	[*label*] INCF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(f) + 1 → (dest)
Status Affected:	Z
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’.

INCFSZ	Increment f, Skip if 0
Syntax:	[*label*] INCFSZ f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(f) + 1 → (dest), skip if result = 0
Status Affected:	None
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’, then the next instruction, which is already fetched, is discarded and a NOP is executed instead making it a two-cycle instruction.

IORLW	Inclusive OR literal with W
Syntax:	[*label*] IORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .OR. (k) → (W)
Status Affected:	Z
Description:	The contents of the W register are OR’ed with the eight-bit literal ‘k’. The result is placed in the W register.

IORWF	Inclusive OR W with f
Syntax:	[*label*] IORWF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(W).OR. (f) → (dest)
Status Affected:	Z
Description:	Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

MOVF	Move f
Syntax:	[*label*] MOVF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(f) → (dest)
Status Affected:	Z
Description:	The contents of register 'f' are moved to destination 'd'. If 'd' is '0', destination is the W register. If 'd' is '1', the destination is file register 'f'. 'd' = 1 is useful as a test of a file register, since status flag Z is affected.

MOVLB	Move literal to BSR
Syntax:	[*label*] MOVLB k
Operands:	$0 \le k \le 7$
Operation:	k → BSR
Status Affected:	None
Description:	The three-bit literal 'k' is loaded into the Bank Select Register (BSR). The "don't cares" will be assembled at '0'.

MOVLW	Move Literal to W
Syntax:	[*label*] MOVLW k
Operands:	$0 \le k \le 255$
Operation:	k → (W)
Status Affected:	None
Description:	The eight-bit literal 'k' is loaded into the W register. The "don't cares" will assembled as '0's.

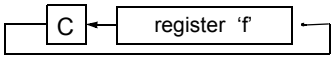
MOVWF	Move W to f
Syntax:	[*label*] MOVWF f
Operands:	$0 \le f \le 31$
Operation:	(W) → (f)
Status Affected:	None
Description:	Move data from the W register to register 'f'.

NOP	No Operation
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.

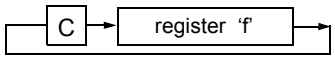
OPTION	Load OPTION Register
Syntax:	[*label*] Option
Operands:	None
Operation:	(W) → Option
Status Affected:	None
Description:	The content of the W register is loaded into the OPTION register.

RETLW	Return with Literal in W
Syntax:	[*label*] RETLW k
Operands:	$0 \le k \le 255$
Operation:	$k \rightarrow (W)$; $TOS \rightarrow PC$
Status Affected:	None
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.

RLF	Rotate Left f through Carry
Syntax:	[*label*] RLF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C
Description:	The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is stored back in register 'f'.



RRF	Rotate Right f through Carry
Syntax:	[*label*] RRF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C
Description:	The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.



SLEEP	Enter SLEEP Mode
Syntax:	[*label*] SLEEP
Operands:	None
Operation:	00h → WDT; 0 → WDT prescaler; 1 → $\overline{TO}$; 0 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$, GPWUF
Description:	Time-out Status bit ($\overline{TO}$) is set. The Power-down Status bit ($\overline{PD}$) is cleared. GPWUF is unaffected. The WDT and its prescaler are cleared. The processor is put into Sleep mode with the oscillator stopped. See **Section 8.8 "Power-down Mode (Sleep)"** on Sleep for more details.

SUBWF	Subtract W from f
Syntax:	[*label*] SUBWF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(f) – (W) → (dest)
Status Affected:	C, DC, Z
Description:	Subtract (two's complement method) the W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

SWAPF	Swap Nibbles in f
Syntax:	[*label*] SWAPF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(f<3:0>) → (dest<7:4>); (f<7:4>) → (dest<3:0>)
Status Affected:	None
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'.

TRIS	Load TRIS Register
Syntax:	[*label*] TRIS f
Operands:	f = 6
Operation:	(W) → TRIS register f
Status Affected:	None
Description:	TRIS register 'f' (f = 6 or 7) is loaded with the contents of the W register.

XORLW	Exclusive OR literal with W
Syntax:	[*label*] XORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .XOR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.

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
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'.

11.0 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

11.1 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.

11.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

11.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

The compilers include a macro assembler, linker, preprocessor, and one-step driver, and can run on multiple platforms.

11.4 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

11.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

11.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

11.7 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

11.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

11.9 MPLAB ICD 3 In-Circuit Debugger System

MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost effective high-speed hardware debugger/programmer for Microchip Flash Digital Signal Controller (DSC) and microcontroller (MCU) devices. It debugs and programs PIC® Flash microcontrollers and dsPIC® DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (IDE).

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

11.10 PICkit 3 In-Circuit Debugger/ Programmer and PICkit 3 Debug Express

The MPLAB PICkit 3 allows debugging and programming of PIC® and dsPIC® Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE). The MPLAB PICkit 3 is connected to the design engineer's PC using a full speed USB interface and can be connected to the target via an Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the reset line to implement in-circuit debugging and In-Circuit Serial Programming™.

The PICkit 3 Debug Express include the PICkit 3, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

11.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

11.12 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an MMC card for file storage and data applications.

11.13 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

12.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings(†)

Ambient temperature under bias ..... -40°C to +85°C
Storage temperature ..... -55°C to +150°C
Voltage on V_{DD} with respect to V_{SS} ..... 0 to +6.5V
Voltage on V_{DDRF} with respect to V_{SSRF} ..... 0 to +3.9V
Voltage on $\overline{MCLR}$ with respect to V_{SS} ..... 0 to +13.5V
Voltage on all other pins with respect to V_{SS} ..... -0.3V to (V_{DD} + 0.3V)
Total power dissipation(1) ..... 700 mW
Max. current out of V_{SS} pin ..... 200 mA
Max. current into V_{DD} pin ..... 150 mA
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 ..... 25 mA
Max. output current sourced by I/O port ..... 75 mA
Max. output current sunk by I/O port ..... 75 mA

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

†NOTICE: Stresses above those listed under “Absolute 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.

FIGURE 12-1: PIC12F529T48A VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ +85°C

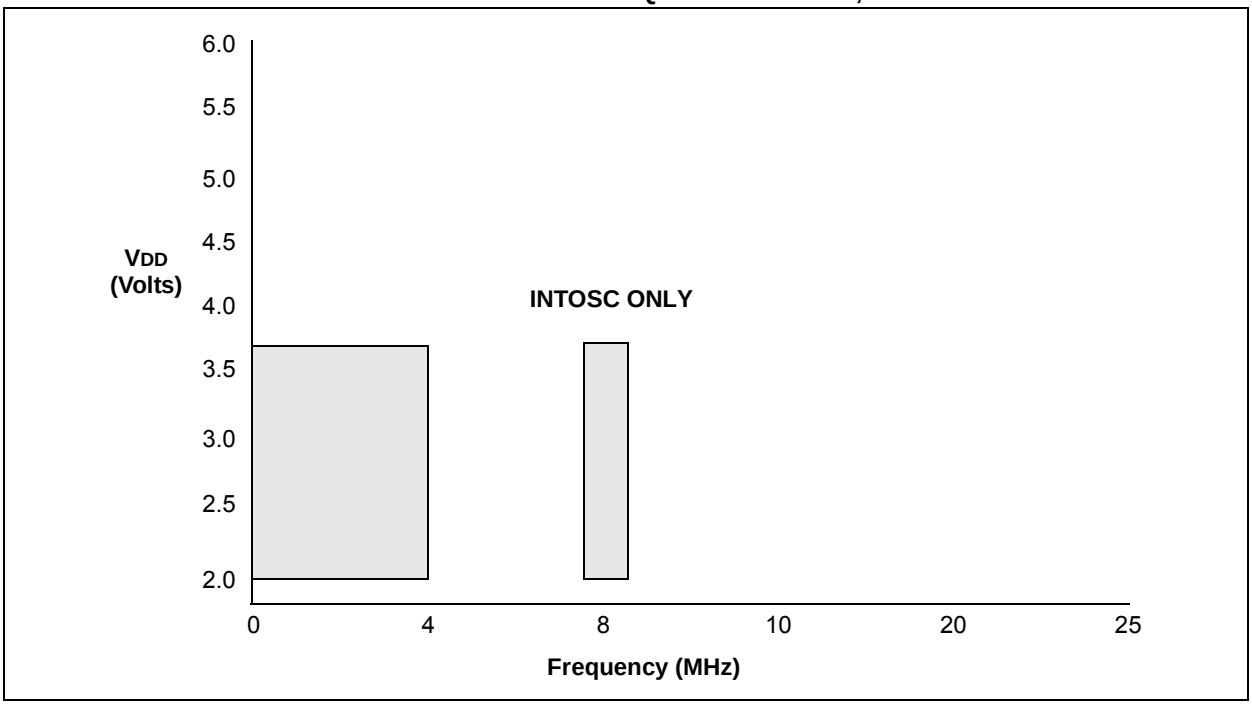
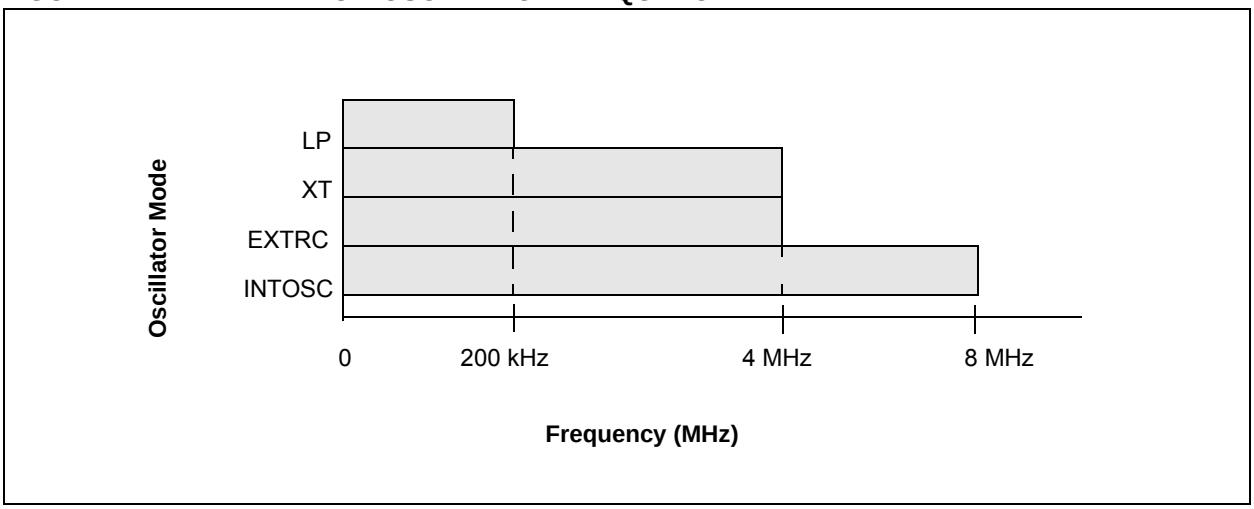


FIGURE 12-2: MAXIMUM OSCILLATOR FREQUENCY TABLE



12.1 DC Characteristics

TABLE 12-1: DC CHARACTERISTICS: PIC12F529T48A (INDUSTRIAL)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C (industrial)				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ(1)**	**Max.**	**Units**	**Conditions**
D001	VDD	**Supply Voltage**	2.0		3.7	V	See **Figure 12-1**
D002	VDR	**RAM Data Retention Voltage(2)**	—	1.5*	—	V	Device in Sleep mode
D003	VPOR	**VDD Start Voltage** to ensure Power-on Reset	—	Vss	—	V	See **Section 8.4 "Power-on Reset (POR)"** for details
D004	SVDD	**VDD Rise Rate** to ensure Power-on Reset	0.05*	—	—	V/ms	See **Section 8.4 "Power-on Reset (POR)"** for details
D005	IDDP	**Supply Current** During Prog/ Erase.	—	250*	—	µA	
D010	IDD	**Supply Current(3,4)**	—	175	250	µA	FOSC = 4 MHz, VDD = 2.0V
			—	250	400	µA	FOSC = 8 MHz, VDD = 2.0V
			—	11	20	µA	FOSC = 32 kHz, VDD = 2.0V
D020	IPD	**Power-down Current(5)**	—	0.1	1.2	µA	VDD = 2.0V
D022	IWDT	**WDT Current**	—	1.0	3.0	µA	VDD = 2.0V

* 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.

4: The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail for external clock modes; all I/O pins tri-stated, pulled to VSS, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

5: For standby current measurements, the conditions are the same as IDD, except that the device is in Sleep mode. If a module current is listed, the current is for that specific module enabled and the device in Sleep.

12.2 RF Transmitter Electrical Specifications

Symbol	Description	Conditions	Min	Typ	Max	Unit
Current Consumption						
IDDSL	Supply current in Sleep mode	—	—	0.5	1	µs
IDDT	Supply current in Transmit mode with appropriate external matching	RF Power o/p = +10 dBm	—	16.5	—	mA
		RF Power o/p = 0 dBm	—	9	—	mA
RF and Baseband Specifications						
FDA_D	Frequency deviation, FSK	Number of programmable values	—	8	—	—
FDA	Frequency deviation, FSK*	—	10	—	100	kHz
BRF	Bit rate, FSK	Permissible Range	0.5	—	100	kbps
BRO	Bit rate, OOK	Permissible Range	0.5	—	10	kbps
OOK_B	OOK Modulation Depth	—	—	50	—	dB
RFOP	RF output power in 50 ohms	High Power Setting	7	10	—	dBm
		Low Power Setting**	-3	0	—	dBm
DRFOPV	Variation in RF output power with supply voltage	2.5V to 3.3V	—	—	3	dB
		1.8V to 3.7V	—	—	7	dB
PHN	Transmitter phase noise at 868.3 MHz	Offset from center frequency: 100 kHz	—	—	-76	dBc/Hz
		350 kHz	—	—	-81	dBc/Hz
		550 kHz	—	—	-91	dBc/Hz
		1.15 MHz	—	—	-101	dBc/Hz
FR	Number of selectable frequencies	—	—	8	—	—
FXOSC	Crystal Oscillator Frequency	—	26	26	26	MHz
STEP	RF Frequency Step	868 MHz	—	3.174	—	kHz
		434 MHz	—	1.587	—	kHz
DFXOSC	Frequency variation of the Oscillator Circuit	No crystal contribution	—	—	+/-25	ppm

Timing Specifications						
t_{WAKE}	Time from Sleep to Tx mode	—	—	—	2	ms
t_{OFFT}	Timer from Tx data activity to Sleep	Programmable	—	2	—	ms
		—	—	20	—	ms
t_{RAMP}	PA Ramp up and down time	—	—	20	—	µs
t_{START}	Time before CTRL Pin mode selection	Time from power on to sampling of CTRL	—	200 µs + TS_OSC	—	ms
Serial Interface Timing Specifications						
f_{ctrl}	CTRL Clock Frequency	—	—	—	10	MHz
f_{ch}	CTRL Clock High time	—	45	—	—	ns
f_{cl}	CTRL Clock Low time	—	45	—	—	ns
t_{rise}	CTRL Clock Rise time	—	—	—	5	ns
t_{fall}	CTRL Clock Fall time	—	—	—	5	ns
t_{setup}	DATA Setup time	From DATA transition to CTRL rising edge	45	—	—	ns
t_{hold}	DATA Hold time	From CTRL rising edge to DATA transition	45	—	—	ns

TABLE 12-2: DC CHARACTERISTICS: PIC12F529T48A (Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating temperature -40°C ≤ TA ≤ +85°C (industrial) Operating voltage VDD range as described in DC specification.				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
	VIL	**Input Low Voltage**					
		I/O ports					
D030A			Vss	—	0.15 VDD	V	Otherwise
D031		with Schmitt Trigger buffer	Vss	—	0.15 VDD	V	
D032		$\overline{MCLR}$, T0CKI	Vss	—	0.15 VDD	V	
D033		OSC1 (EXTRC mode)	Vss	—	0.15 VDD	V	**(Note 1)**
D033A		OSC1 (XT and LP modes)	Vss	—	0.3	V	
	VIH	**Input High Voltage**					
		I/O ports		—			
D040A			0.25 VDD + 0.8V	—	VDD	V	Otherwise
D041		with Schmitt Trigger buffer	0.85 VDD	—	VDD	V	For entire VDD range
D042		MCLR, T0CKI	0.85 VDD	—	VDD	V	
D042A		OSC1 (EXTRC mode)	0.85 VDD	—	VDD	V	**(Note 1)**
D043		OSC1 (XT and LP modes)	1.6	—	VDD	V	
D070	IPUR	**I/O PORT weak pull-up current(5)**	50	250	400	µA	VDD = 3.7V, VPIN = VSS
	IIL	**Input Leakage Current(2), (3)**					
D060		I/O ports	—	—	±1	µA	Vss ≤ VPIN ≤ VDD, Pin at high-impedance
D061		GP3/$\overline{MCLR}$(4)	—	±0.7	±5	µA	Vss ≤ VPIN ≤ VDD
D063		OSC1	—	—	±5	µA	Vss ≤ VPIN ≤ VDD, XT and LP osc configuration
		Output Low Voltage					
D080		I/O ports	—	—	0.6	V	IOL = 8.5 mA, VDD = 4.5V, –40°C to +85°C
		Output High Voltage					
D090		I/O ports(3)	VDD – 0.7	—	—	V	IOH = -3.0 mA, VDD = 4.5V, –40°C to +85°C
		Capacitive Loading Specs on Output Pins					
D101		All I/O pins	—	—	50	pF	
		Flash Data Memory					
D120	ED	Byte endurance	100K	1M	—	E/W	–40°C ≤ TA ≤ +85°C
D121	VDRW	VDD for read/write	VMIN	—	3.7	V	

† 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 PIC12F529T48A 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: This specification applies to GP3/$\overline{MCLR}$ configured as GP3 with internal pull-up disabled.

5: This specification applies to all weak pull-up devices, including the weak pull-up found on GP3/$\overline{MCLR}$. The current value listed will be the same whether or not the pin is configured as GP3 with pull-up enabled or $\overline{MCLR}$.

TABLE 12-3: PULL-UP RESISTOR RANGES

VDD (Volts)	Temperature (°C)	Min.	Typ.	Max.	Units
GP0/GP1					
2.0	–40	73K	105K	186K	Ω
	25	73K	113K	187K	Ω
	85	82K	123K	190K	Ω
GP3					
2.0	–40	63K	81K	96K	Ω
	25	77K	93K	116K	Ω
	85	82K	96K	116K	Ω

12.3 Timing Parameter Symbology and Load Conditions – PIC12F529T48A

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS

T	
F Frequency	T Time

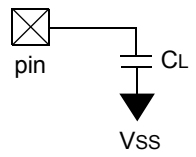
Lowercase subscripts (pp) and their meanings:

pp			
2	to	mc	$\overline{\text{MCLR}}$
ck	CLKOUT	osc	Oscillator
cy	Cycle time	os	OSC1
drt	Device Reset Timer	t0	T0CKI
io	I/O port	wdt	Watchdog Timer

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (high-impedance)	V	Valid
L	Low	Z	High-impedance

FIGURE 12-3: LOAD CONDITIONS – PIC12F529T48A

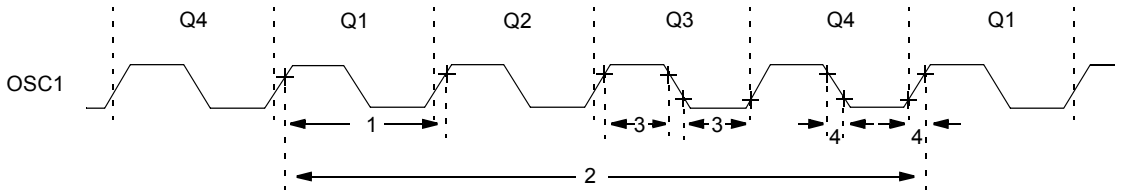


Legend:

C_L = 50 pF for all pins except OSC2

15 pF for OSC2 in XT or LP modes when external clock is used to drive OSC1

FIGURE 12-4: EXTERNAL CLOCK TIMING – PIC12F529T48A



12.4 AC Characteristics

TABLE 12-4: EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C (industrial), Operating Voltage VDD range is described in Section 12.0 "Electrical Characteristics".				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ(1)**	**Max.**	**Units**	**Conditions**
1A	FOSC	External CLKIN Frequency(2)	DC	—	4	MHz	XT Oscillator mode
			DC	—	200	kHz	LP Oscillator mode
		Oscillator Frequency(2)	DC	—	4	MHz	EXTRC Oscillator mode
			0.1	—	4	MHz	XT Oscillator mode
			DC	—	200	kHz	LP Oscillator mode
1	TOSC	External CLKIN Period(2)	250	—	—	ns	XT Oscillator mode
			5	—	—	µs	LP Oscillator mode
		Oscillator Period(2)	250	—	—	ns	EXTRC Oscillator mode
			250	—	10,000	ns	XT Oscillator mode
			5	—	—	µs	LP Oscillator mode
2	TCY	Instruction Cycle Time	200	4/FOSC	DC	ns	
3	TosL, TosH	Clock in (OSC1) Low or High Time	50*	—	—	ns	XT Oscillator
			2*	—	—	µs	LP Oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	—	—	25*	ns	XT Oscillator
			—	—	50*	ns	LP Oscillator

* These parameters are characterized but not tested.

Note 1: All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

TABLE 12-5: CALIBRATED INTERNAL RC FREQUENCIES

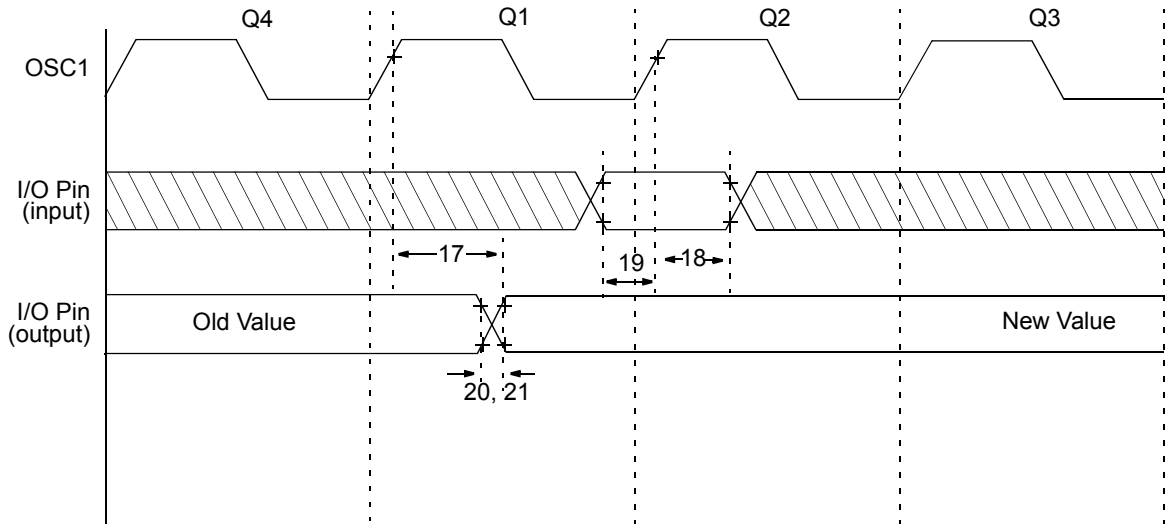
AC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C (industrial), Operating Voltage VDD range is described in Section 12.0 "Electrical Characteristics".					
Param No.	**Sym.**	**Characteristic**	**Freq. Tolerance**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
F10	FOSC	Internal Calibrated INTOSC Frequency(1)	±1%	7.92	8.00	8.08	MHz	3.5V, 25C
			±2%	7.84	8.00	8.16	MHz	2.5V ≤ VDD ≤ 3.7V 0°C ≤ TA ≤ +85°C
			±5%	7.60	8.00	8.40	MHz	2.0V ≤ VDD ≤ 3.7V -40°C ≤ TA ≤ +85°C (Ind.)

* These parameters are characterized but not tested.

† Data in the Typical ("Typ") column is at 3.7V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 uF and 0.01 uF values in parallel are recommended.

FIGURE 12-5: I/O TIMING



Note: All tests must be done with specified capacitive loads (see data sheet) 50 pF on I/O pins and CLKOUT.

TABLE 12-6: TIMING REQUIREMENTS

AC CHARACTERISTICS		**Standard Operating Conditions (unless otherwise specified)** Operating Temperature -40°C ≤ TA ≤ +85°C (industrial) Operating Voltage VDD range is described in **Section 12.0 "Electrical Characteristics"**.				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ(1)**	**Max.**	**Units**
17	TOSH2IOV	OSC1↑ (Q1 cycle) to Port Out Valid(2), (3)	—	—	100*	ns
18	TOSH2IOI	OSC1↑ (Q2 cycle) to Port Input Invalid (I/O in hold time)(2)	50	—	—	ns
19	TIOV2OSH	Port Input Valid to OSC1↑ (I/O in setup time)	20	—	—	ns
20	TIOR	Port Output Rise Time(3)	—	10	50**	ns
21	TIOF	Port Output Fall Time(3)	—	10	50**	ns

TBD = To be determined.

* These parameters are characterized but not tested.

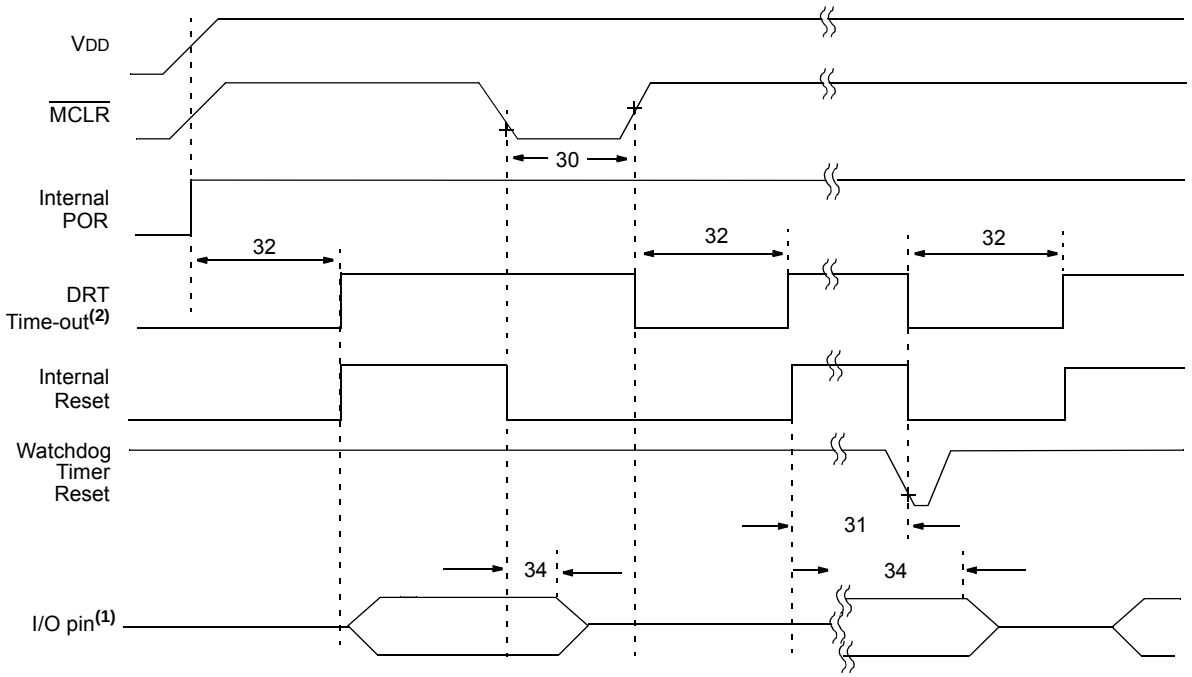
** These parameters are design targets and are not tested.

Note 1: Data in the Typical ("Typ") column is at 3.7V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

2: Measurements are taken in EXTRC mode.

3: See Figure 12-3 for loading conditions.

FIGURE 12-6: RESET, WATCHDOG TIMER AND DEVICE RESET TIMER TIMING



Note 1: I/O pins must be taken out of High-Impedance mode by enabling the output drivers in software.
2: Runs in MCLR or WDT Reset only in XT and LP.

TABLE 12-7: RESET, WATCHDOG TIMER AND DEVICE RESET TIMER – PIC12F529T48A

AC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C (industrial) Operating Voltage VDD range is described in Table 12-2.				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ(1)**	**Max.**	**Units**	**Conditions**
30	TMCL	MCLR Pulse Width (low)	2000*	—	—	ns	VDD = 3.0V
31	TWDT	Watchdog Timer Time-out Period (no prescaler)	9*	20*	35*	ms	VDD = 3.0V (Industrial)
32	TDRT	Device Reset Timer Period					
		Standard	9*	20*	35*	ms	VDD = 3.0V (Industrial)
		Short	0.5*	1.125*	2*	ms	VDD = 3.0V (Industrial)
34	TIOZ	I/O High-impedance from MCLR low	—	—	2000*	ns	

* These parameters are characterized but not tested.

Note 1: Data in the Typical ("Typ") column is at 3.7V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 12-8: DRT (DEVICE RESET TIMER PERIOD)

Oscillator Configuration	POR Reset	Subsequent Resets
IntRC and ExtRC	1 ms (typical)	10 μs (typical)
XT and LP	18 ms (typical)	18 ms (typical)

FIGURE 12-7: TIMER0 CLOCK TIMINGS

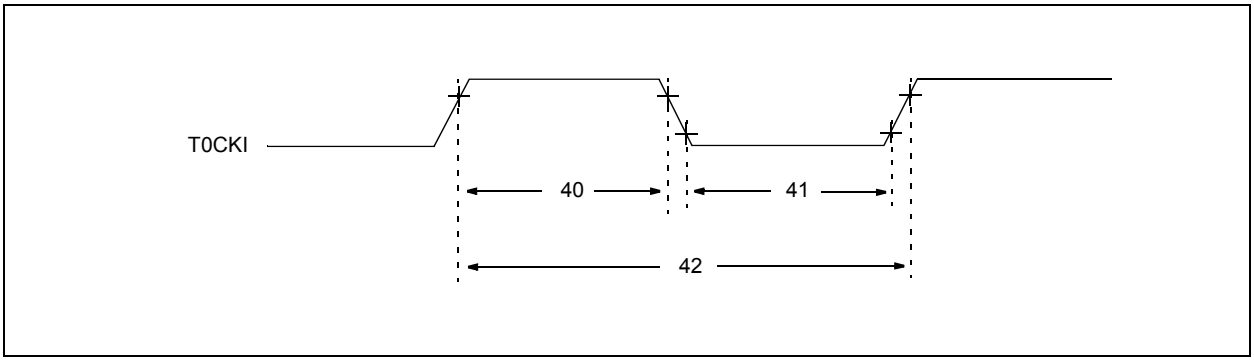


TABLE 12-9: TIMER0 CLOCK REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C (industrial) Operating Voltage VDD range is described in Table 12-2.				
Param No.	**Sym.**	**Characteristic**		**Min.**	**Typ(1)**	**Max.**	**Units**	**Conditions**
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		20 or TCY + 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 3.7V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 12-10: FLASH DATA MEMORY WRITE/ERASE REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C (industrial) Operating Voltage VDD range is described in Table 12-2.				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ(1)**	**Max.**	**Units**	**Conditions**
43	TDW	Flash Data Memory Write Cycle Time	2	3.5	5	ms	
44	TDE	Flash Data Memory Erase Cycle Time	2	3	4	ms	

* These parameters are characterized but not tested.

Note 1: Data in the Typical ("Typ") column is at 3.7V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

NOTES:

13.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

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 are ensured to operate properly only within the specified range.

Note:	The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

"Typical" represents the mean of the distribution at 25°C. "Maximum" or "minimum" represents (mean + 3σ) or (mean - 3σ) respectively, where σ is a standard deviation, over each temperature range.

FIGURE 13-1: TYPICAL IDD vs. FOSC OVER VDD (XT, EXTRC mode)

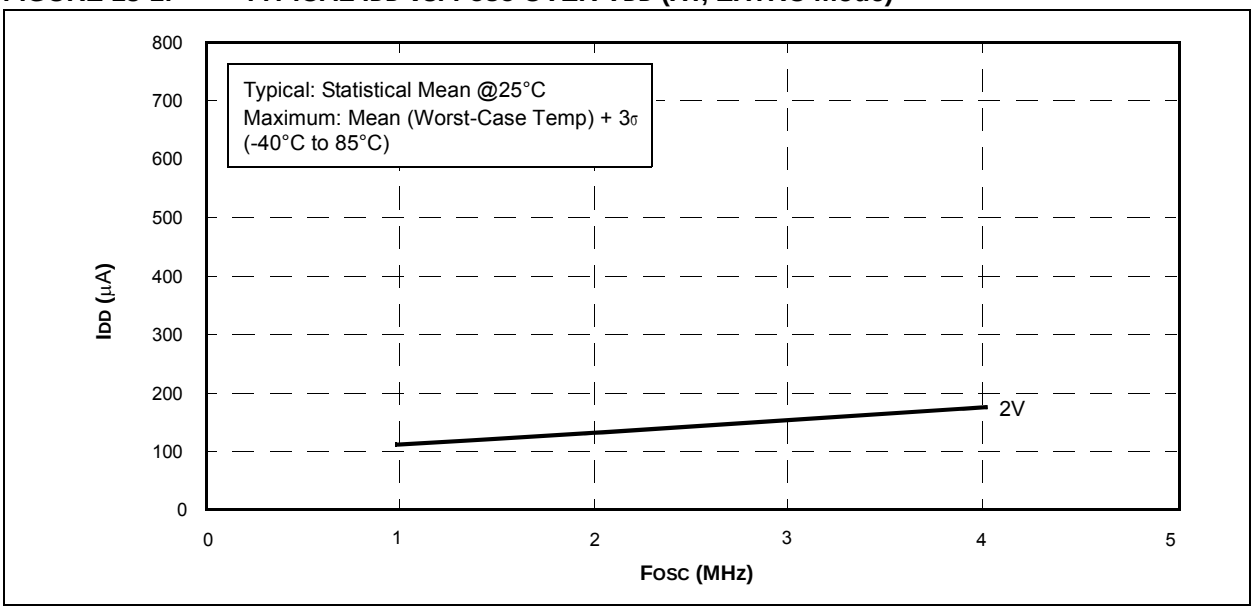


FIGURE 13-2: MAXIMUM IDD vs. FOSC OVER VDD (XT, EXTRC mode)

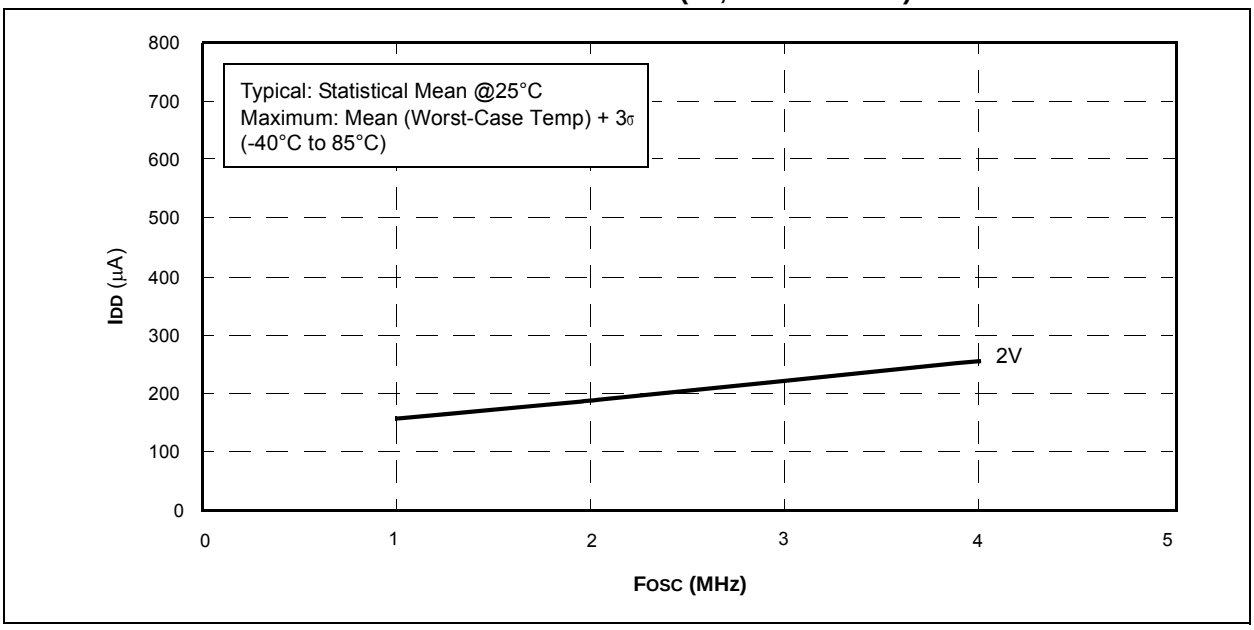


FIGURE 13-3: IDD VS. VDD OVER FOSC (LP MODE)

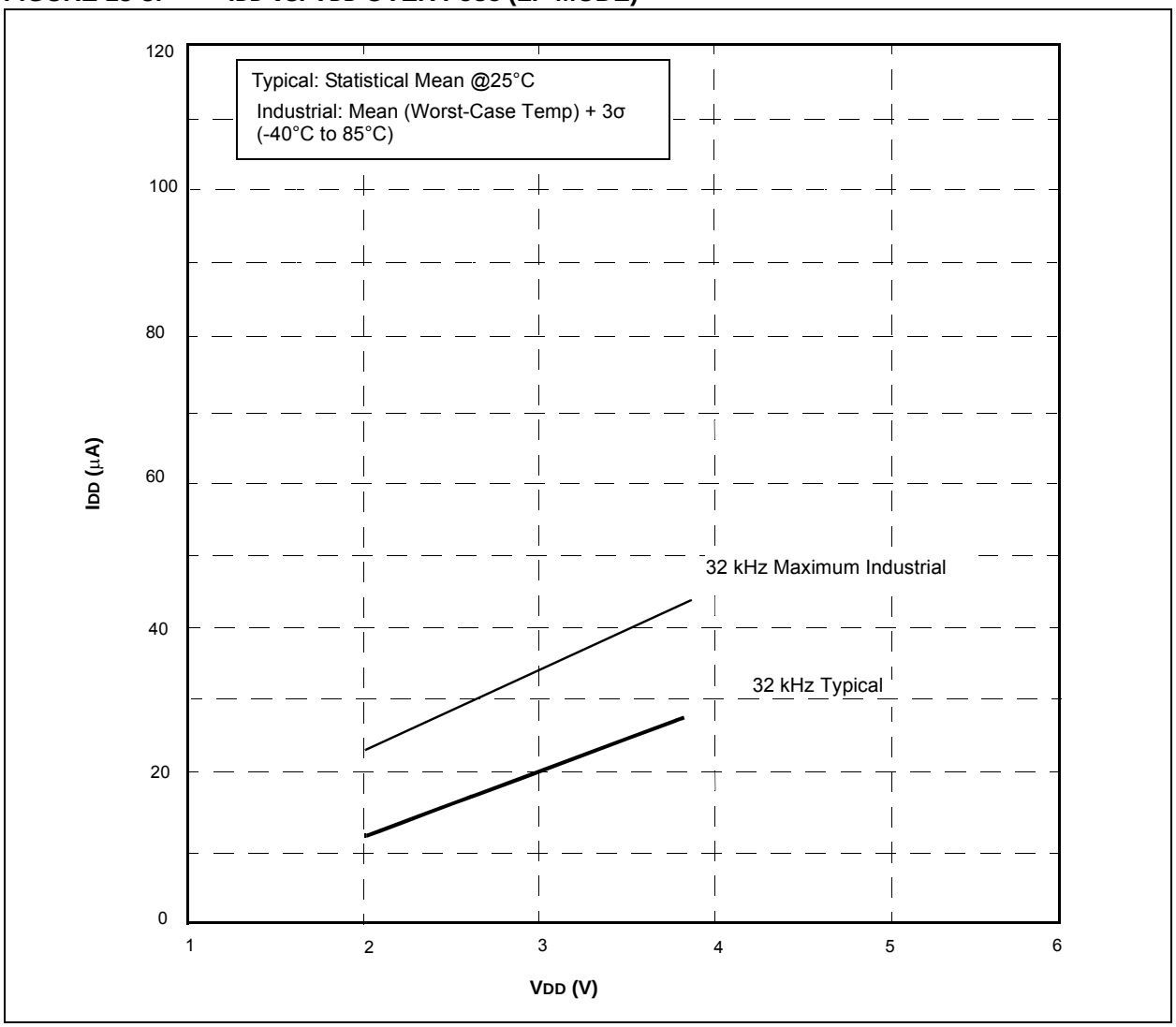


FIGURE 13-4: TYPICAL IPD VS. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)

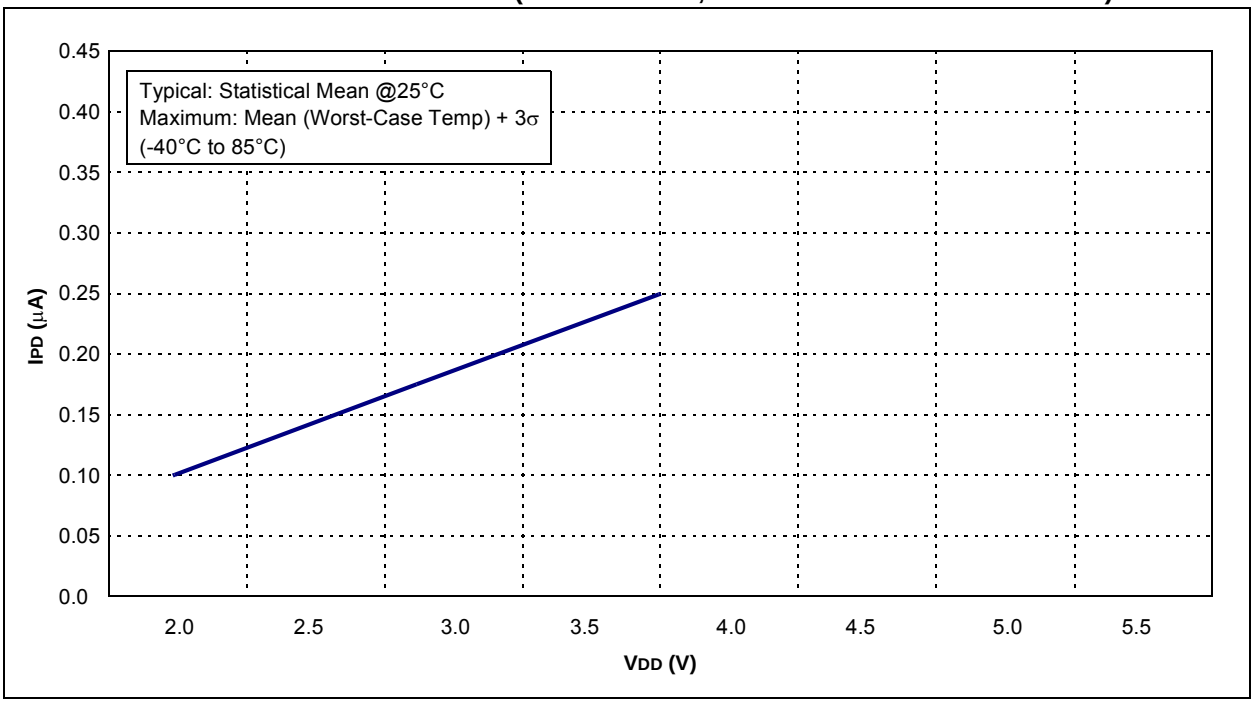


FIGURE 13-5: MAXIMUM IPD VS. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)

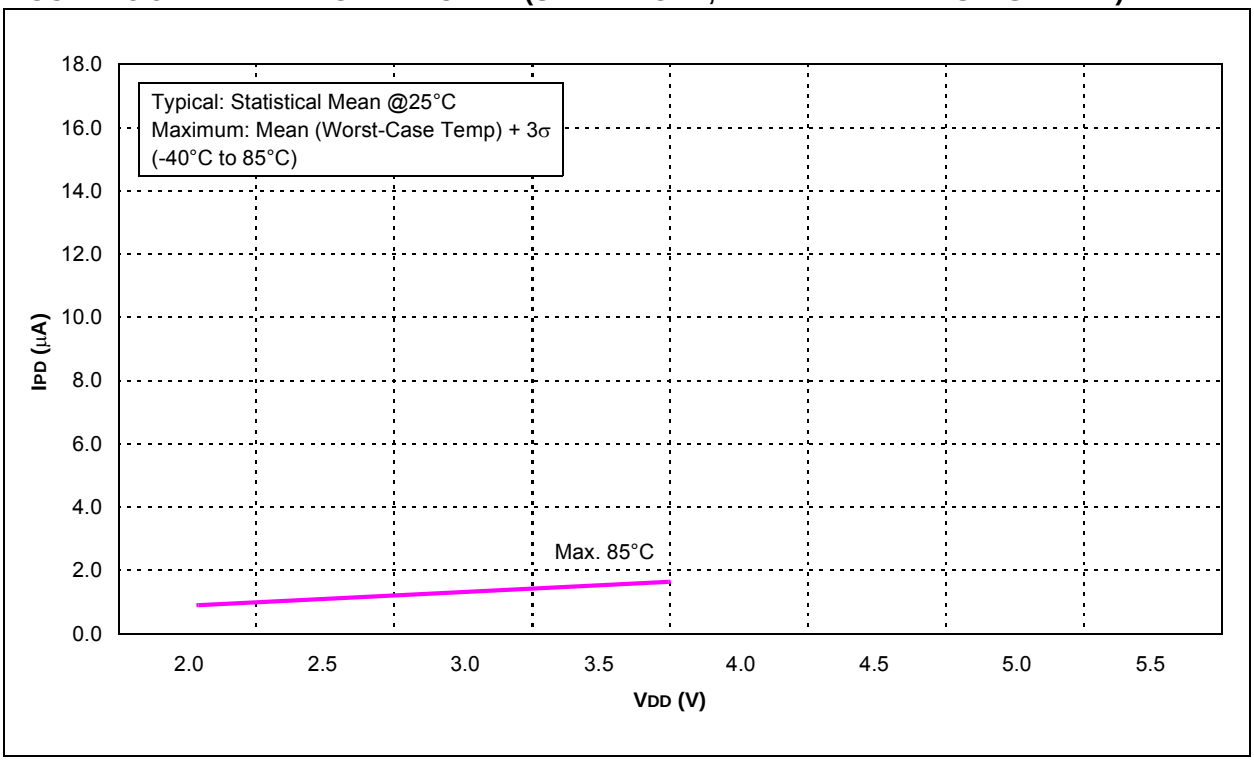


FIGURE 13-6: TYPICAL WDT IPD vs. VDD

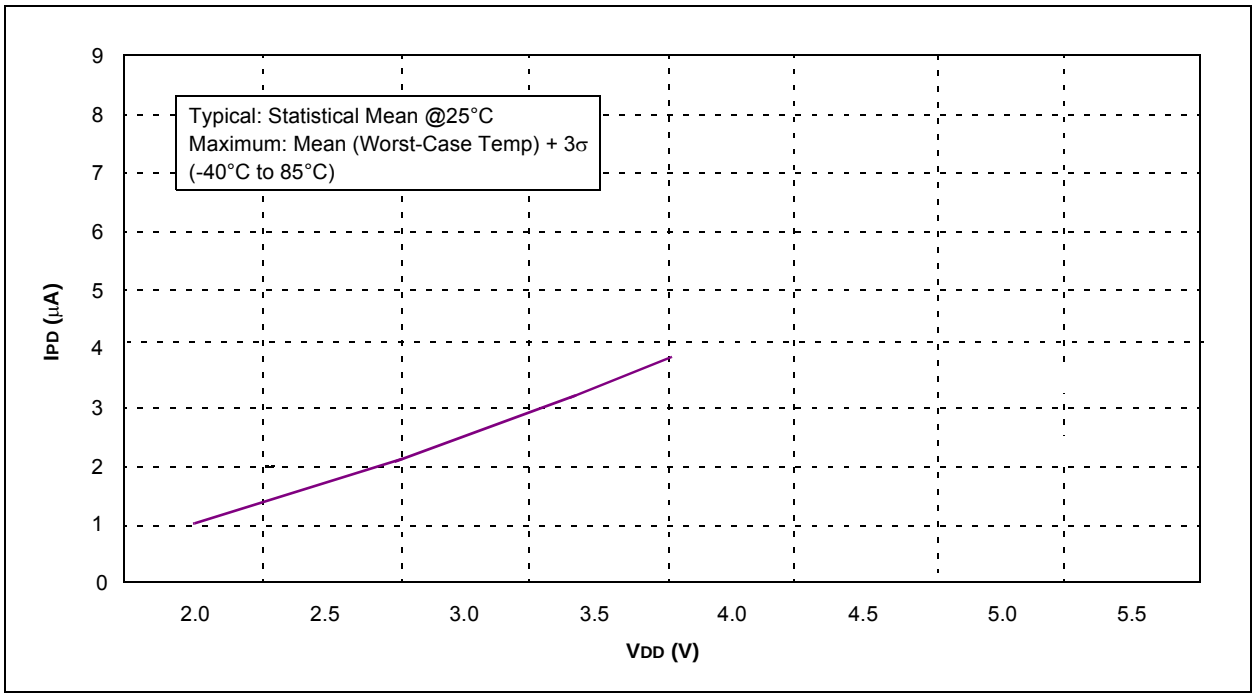


FIGURE 13-7: MAXIMUM WDT IPD vs. VDD OVER TEMPERATURE

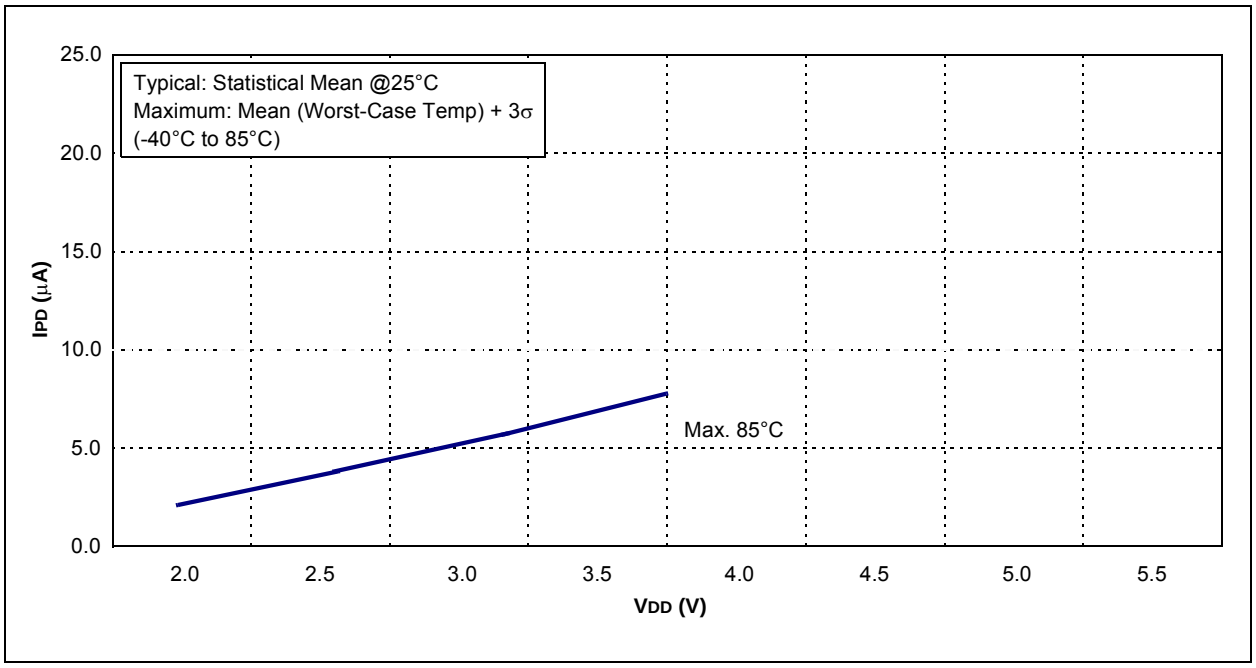


FIGURE 13-8: WDT TIME-OUT vs. VDD OVER TEMPERATURE (NO PRESCALER)

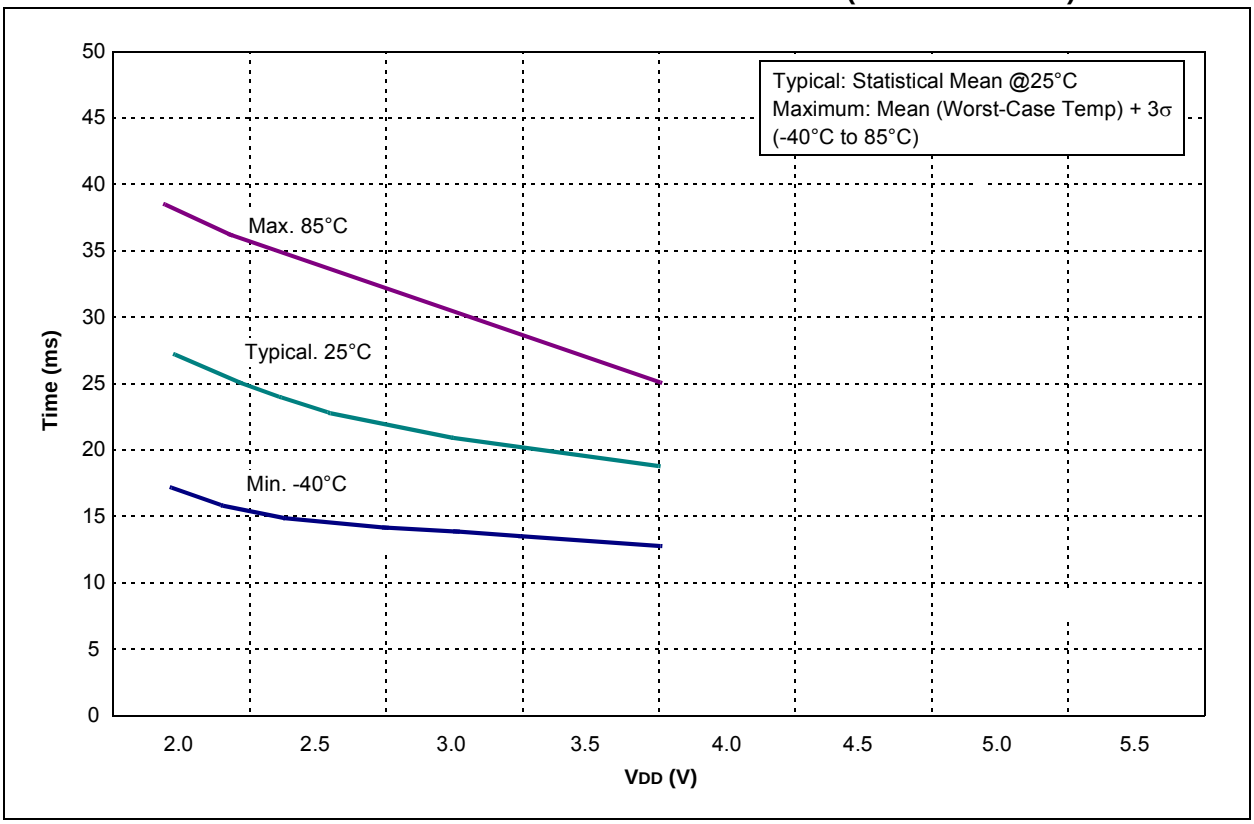


FIGURE 13-9: VOL vs. IOL OVER TEMPERATURE (VDD = 3.0V)

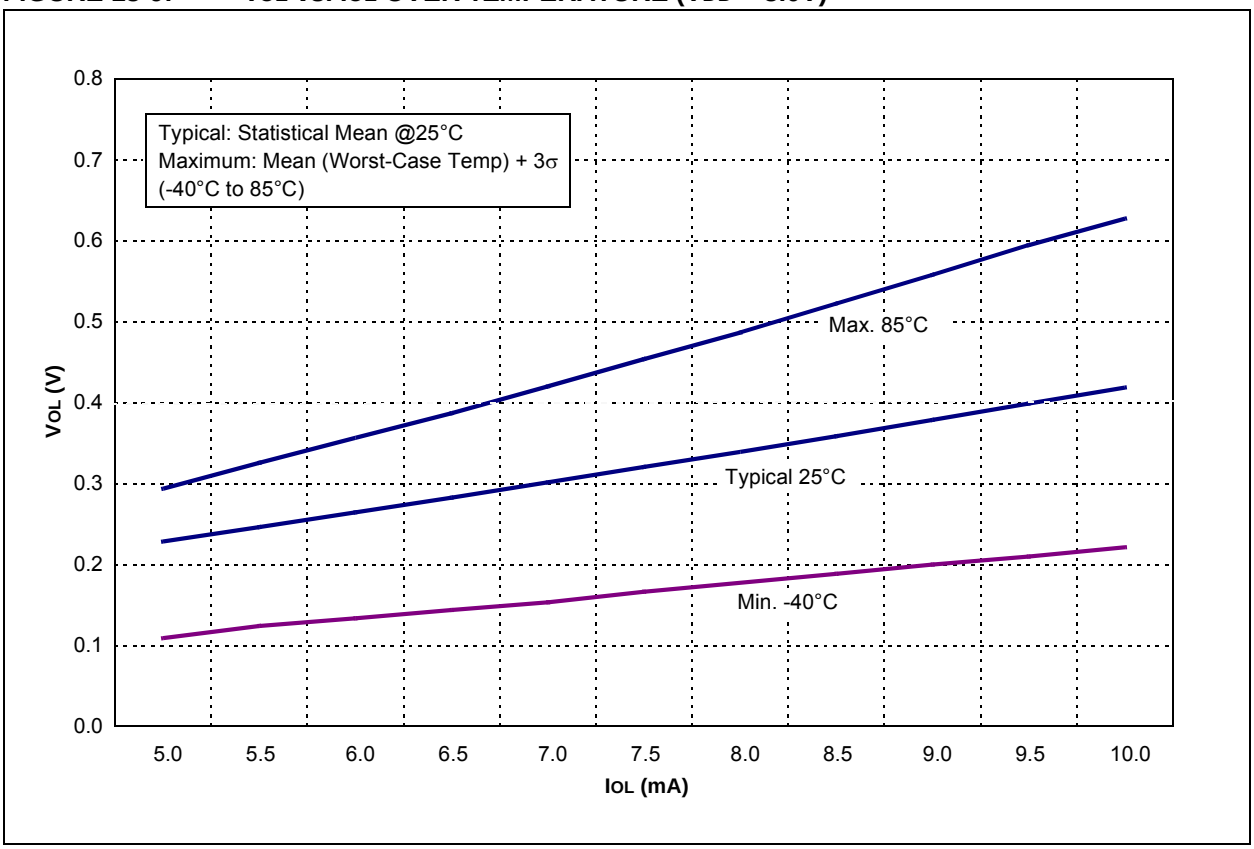


FIGURE 13-10: VOH vs. IOH OVER TEMPERATURE (VDD = 3.0V)

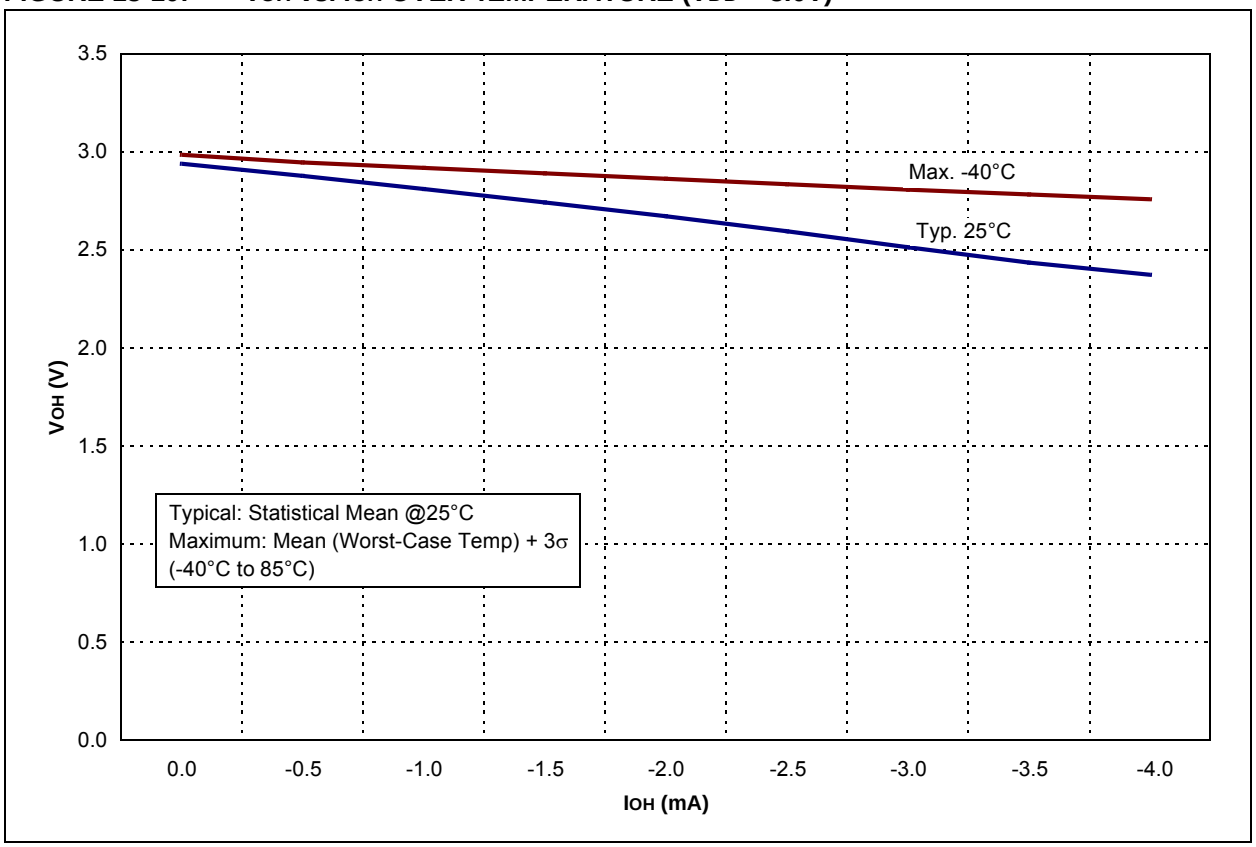


FIGURE 13-11: TTL INPUT THRESHOLD VIN vs. VDD

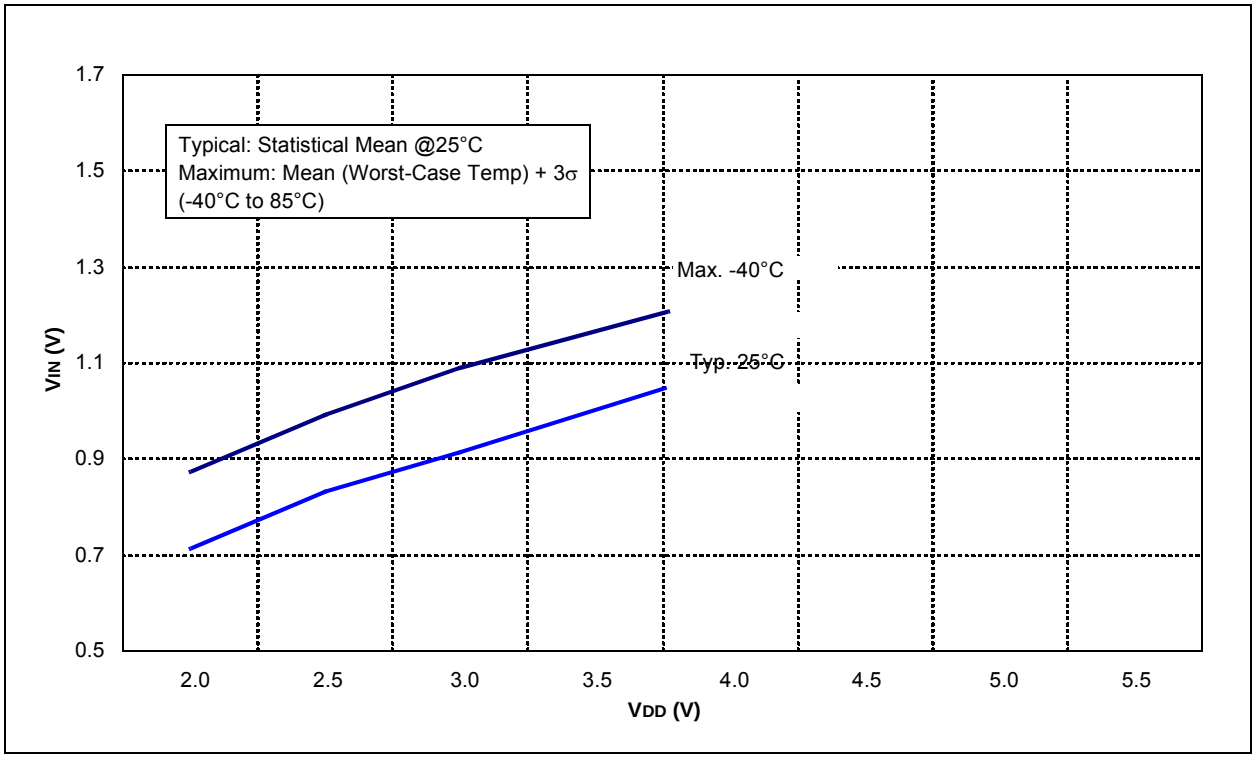


FIGURE 13-12: SCHMITT TRIGGER INPUT THRESHOLD VIN VS. VDD

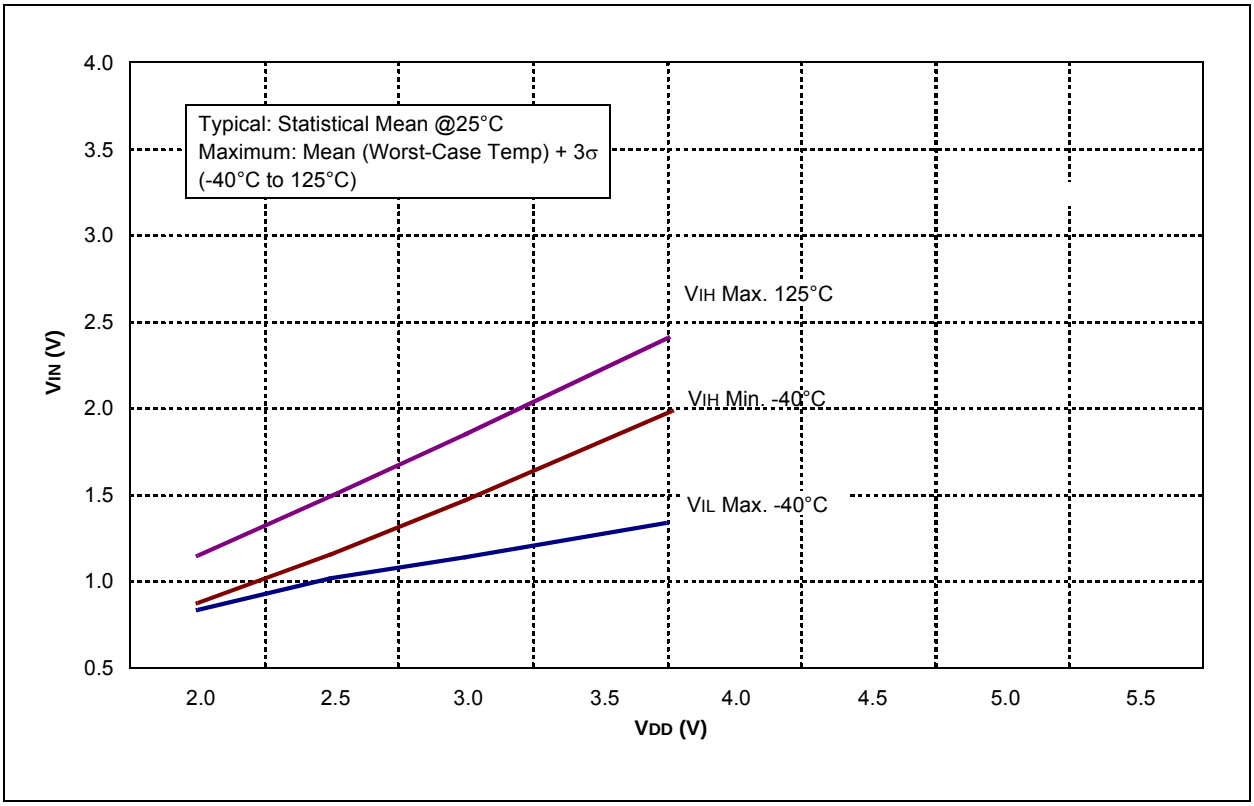
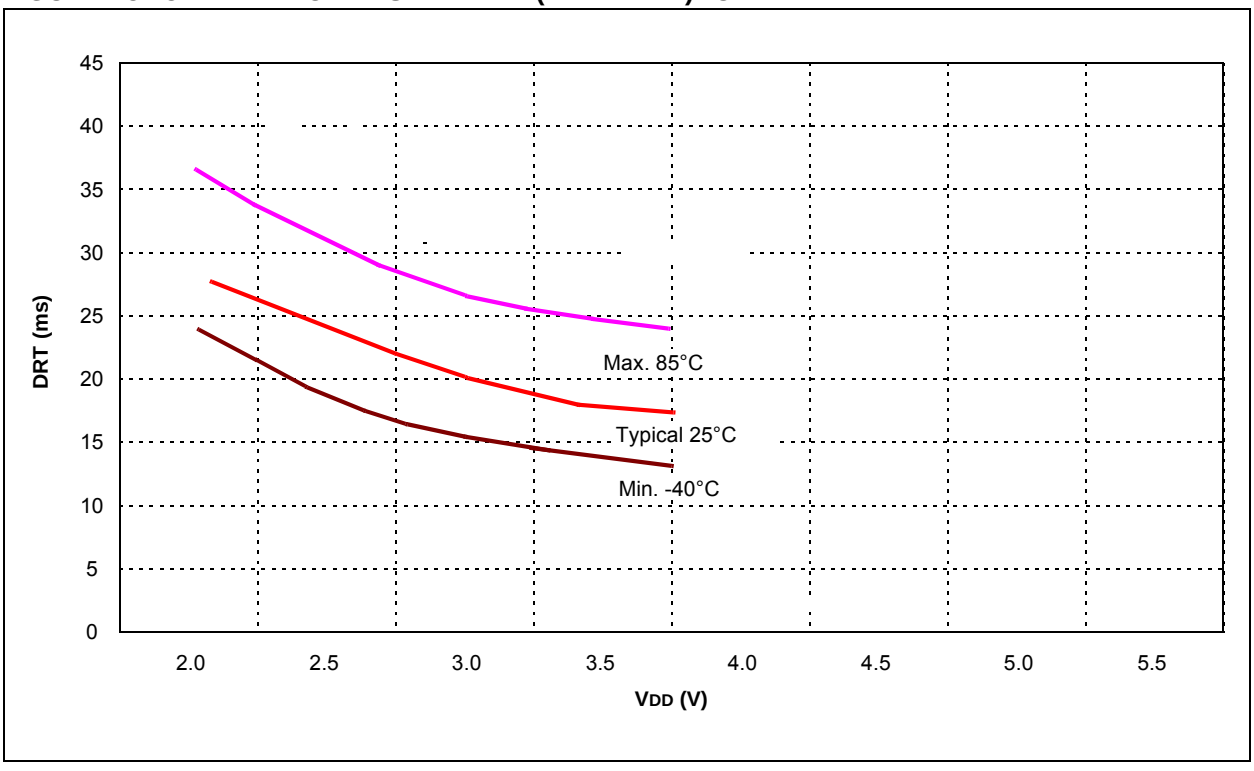


FIGURE 13-13: DEVICE RESET TIMER (XT AND LP) VS. VDD

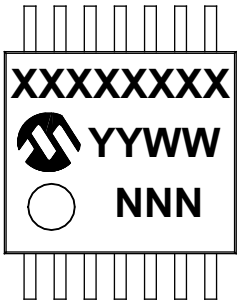


NOTES:

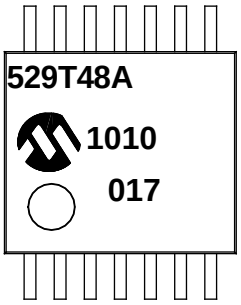
14.0 PACKAGING INFORMATION

14.1 Package Marking Information

14-Lead TSSOP (4.4 mm)



Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

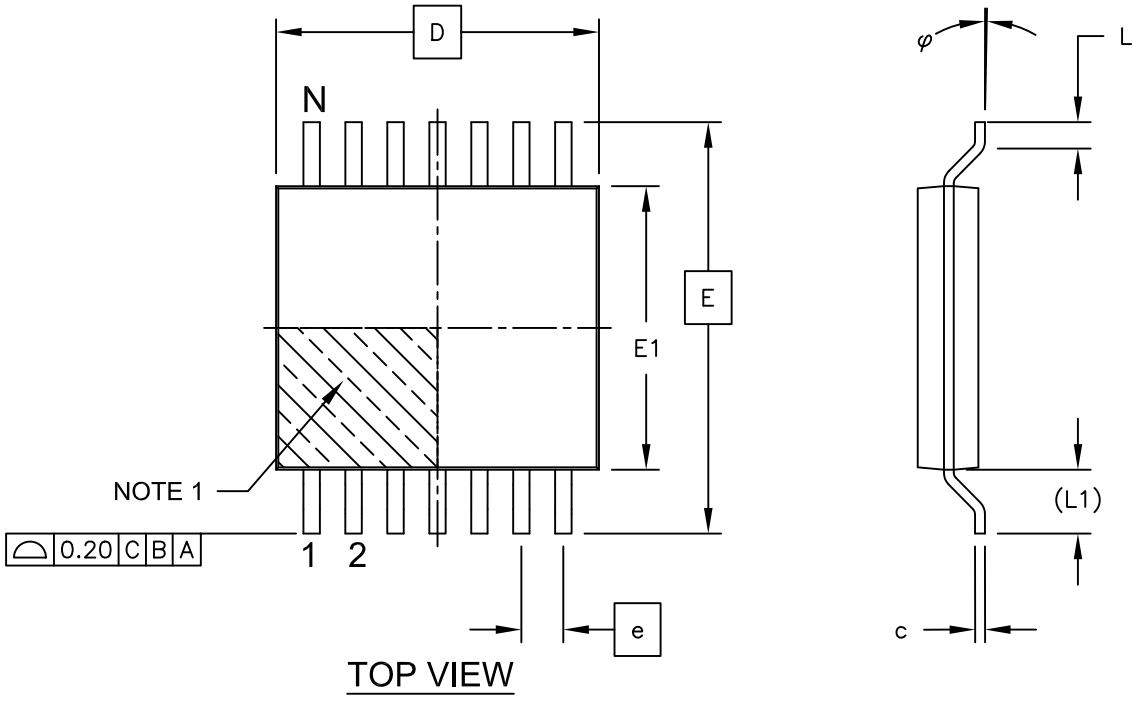
***** Standard PIC® device marking consists of Microchip part number, year code, week code, and traceability code. For PIC device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

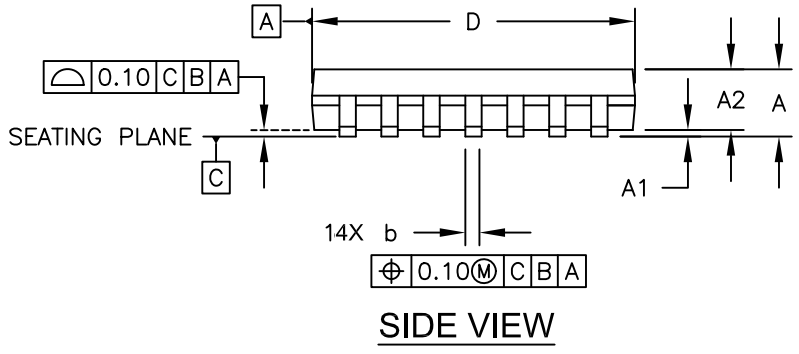
14.2 Package Details

The following sections give the technical details of the packages.

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

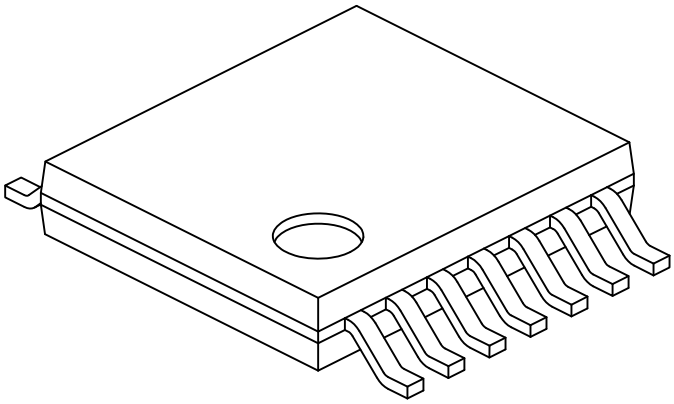




Microchip Technology Drawing C04-087C Sheet 1 of 2

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:

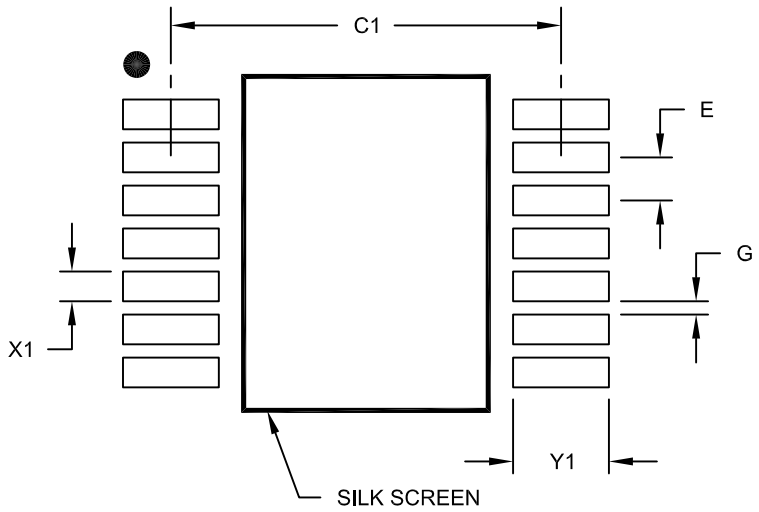
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M

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

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C1		5.90	
Contact Pad Width (X14)	X1			0.45
Contact Pad Length (X14)	Y1			1.45
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2087A

APPENDIX A: DATA SHEET REVISION HISTORY

Revision A (04/2012)

Initial release of this data sheet.

NOTES:

INDEX

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

TO: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC12F529T48A Literature Number: DS41634A

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X](1)	X	/XX	XXX
Device	Tape and Reel Option	Temperature Range	Package	Pattern

Device:	PIC12F529T48A	
Tape and Reel Option:	Blank T	= Standard packaging (tube or tray) = Tape and Reel(1)
Temperature Range:	I	= -40°C to+85°C (Industrial)
Package:	ST	= 14-pin TSSOP
Pattern:	Special Requirements	

Note: Tape and Reel available for only the following packages: TSSOP.

Examples:

a) PIC12F529T48A-I/P = Industrial temp., TSSOP package (Pb-free)

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Osaka
Tel: 81-66-152-7160
Fax: 81-66-152-9310

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

11/29/11

