Universidad de Costa Rica
Facultad de Ingeniería

Escuela de Ingeniería Eléctrica
IE-0624 Laboratorio de Microcontroladores
Grupo 01 - I Ciclo 2025

# Laboratorio 1
*Introducción a microcontroladores y manejo de GPIOS*

Profesor: MSc. Marco Villalta Fallas

Estudiante:
Ángeles Ulate Jarquín - C07908

Martes 25 de Marzo

# Índice

# 1. Introducción

En un mundo donde la microelectrónica se vuelve cada vez más esencial para la automatización de procesos, comprender y aplicar los principios fundamentales de los microcontroladores representa una base indispensable para la ingeniería moderna. Este proyecto nace del deseo de aplicar esos conocimientos en una solución creativa, funcional y sencilla: un dado electrónico.

El objetivo principal fue implementar un sistema digital capaz de simular el funcionamiento de un dado tradicional, utilizando el microcontrolador PIC12F683. El dispositivo genera números aleatorios del 1 al 6 y los representa mediante el encendido momentáneo de LEDs, dispuestos según la configuración visual de un dado físico. La activación del sistema se realiza mediante un interruptor momentáneo, lo que ofrece una interfaz simple pero efectiva para el usuario.

Este trabajo permitió integrar y reforzar habilidades en programación en lenguaje C, configuración de entradas y salidas digitales (GPIO), diseño lógico, y validación funcional a través de simulaciones realizadas en SimulIDE. Además, el proyecto fue desarrollado de forma estructurada, documentado con estándares técnicos y versionado mediante Git.

Toda la implementación, incluyendo el código en C y la simulación, que puede consultarse públicamente en el siguiente repositorio en línea:

- https://github.com/AngelesUJ/IE-0624-Laboratorio-de-Microcontroladores-IC25.git

# 2. Nota teórica

## 2.1. Microcontroladores: visión general

Los microcontroladores son dispositivos integrados capaces de ejecutar instrucciones programadas, controlar periféricos y gestionar entradas/salidas, todo dentro de un solo circuito integrado. A diferencia de los microprocesadores, que requieren memoria y periféricos externos, los microcontroladores contienen CPU, memoria y módulos de E/S integrados, permitiendo soluciones compactas y eficientes para sistemas embebidos [4].

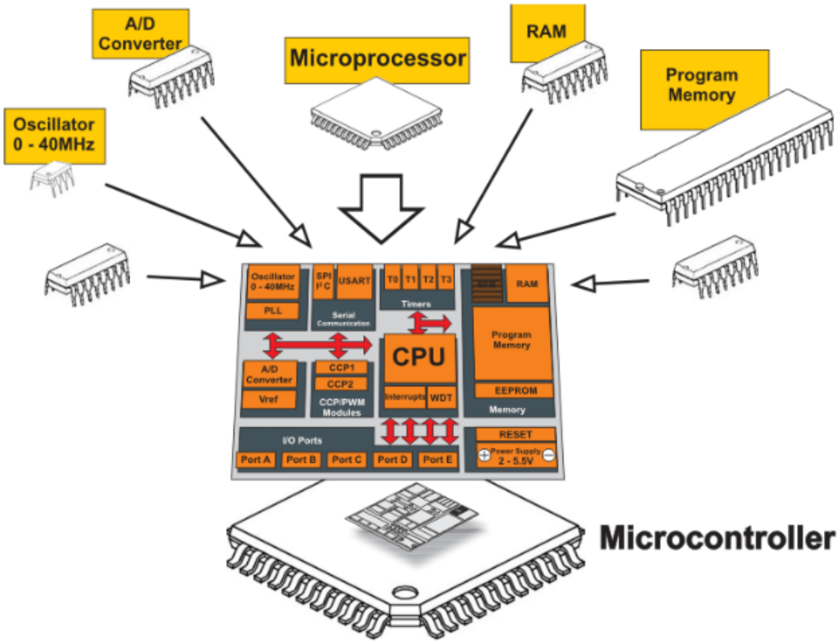


Figura 1: Microcontrolador versus Microprocesador [4].

Este diseño todo en uno reduce costos, consumo de energía y complejidad, lo cual los convierte en una elección ideal para aplicaciones como electrodomésticos, sensores inteligentes, instrumentación, y sistemas didácticos como el dado electrónico implementado en este laboratorio.

## 2.2. PIC12F683: arquitectura y capacidades

El PIC12F683 es un microcontrolador de 8 bits fabricado por Microchip Technology, perteneciente a la familia PIC12. Presenta una arquitectura RISC con 35 instrucciones, 6 pines GPIO configurables, memoria flash de 2 KB y un oscilador interno de 8 MHz. También incluye módulos como comparadores, temporizadores, y soporte para interrupciones [2].

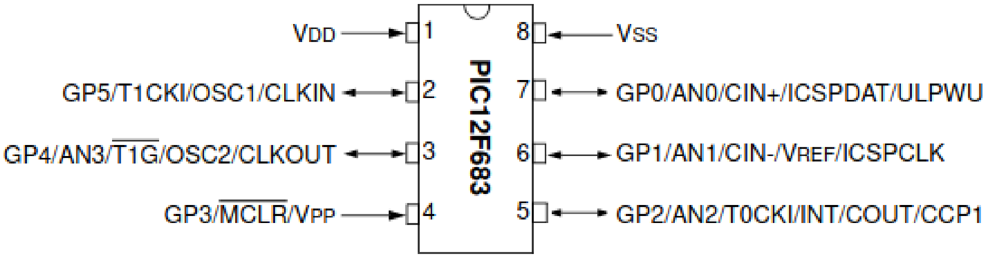

VDD
GP5/T1CKI/OSC1/CLKIN
GP4/AN3/$\overline{\text{T1G}}$/OSC2/CLKOUT
GP3/$\overline{\text{MCLR}}$/VPP
1
2
3
4
PIC12F683
8
7
6
5
VSS
GP0/AN0/CIN+/ICSPDAT/ULPWU
GP1/AN1/CIN-/VREF/ICSPCLK
GP2/AN2/T0CKI/INT/COUT/CCP1

Figura 2: Diagrama de pines del PIC12F683 [2].

Este dispositivo es particularmente útil en aplicaciones pequeñas donde el número de pines es limitado, pero se requiere control preciso de señales digitales. En este proyecto, el PIC12F683 controla LEDs conectados a sus salidas digitales para simular la visualización de las caras de un dado.

### 2.3. GPIOs: entradas y salidas digitales

Los pines GPIO (General Purpose Input/Output) permiten que el microcontrolador lea el estado lógico de sensores o botones (entradas), o controle dispositivos como LEDs o relevadores (salidas). La dirección de cada pin se configura mediante el registro TRISIO, y el valor lógico se lee o escribe en GPIO.

Además, el diseño del dado requiere resistencias limitadoras de corriente en los pines de salida conectados a los LEDs, típicamente de 220 Ω. Esto protege tanto al LED como al microcontrolador, evitando sobrecorriente [3].

### 2.4. Generación pseudoaleatoria en sistemas embebidos

Dado que el PIC12F683 no cuenta con un generador de números aleatorios por hardware, se utilizó una técnica basada en la lectura de un contador que incrementa continuamente dentro del bucle principal. Al presionar el interruptor, se toma el valor actual del contador y se aplica una operación módulo 6 para obtener un valor en el rango 1–6.

Este método aprovecha la aleatoriedad inherente al instante humano de la pulsación y ha sido documentado como una solución válida para sistemas embebidos de bajo costo [5].

### 2.5. Simulación con SimulIDE y programación con SDCC

El entorno de desarrollo se basó en el compilador libre SDCC y el simulador SimulIDE. SimulIDE permite validar circuitos en tiempo real, facilitando la depuración del código y la observación del comportamiento del hardware virtual. La integración de estos entornos proporciona una plataforma educativa eficaz y accesible para prácticas de laboratorio [1].

### 2.6. Lista de componentes con precios

| Componente | Cantidad | Precio Unitario (CRC) | Precio Total (CRC) |
|---|---|---|---|
| Microcontrolador PIC12F683 | 1 | 1114 | 1114 |
| Resistencias 25 Ω | 6 | 50 | 300 |
| Resistencia 50 Ω | 1 | 50 | 50 |
| Resistencia 10 kΩ | 1 | 50 | 50 |
| LEDs | 7 | 100 | 700 |
| Botón momentáneo | 1 | 500 | 500 |
| **Total** | | | **2,714** |

Cuadro 1: Lista de componentes y costos asociados

Notas:

- Los precios unitarios son estimaciones basadas en costos comunes de componentes en tiendas locales de electrónica.
- Es crucial verificar los precios con proveedores específicos para obtener datos precisos que reflejen el mercado actual en Costa Rica.
- El costo total es la suma de los productos de las cantidades por los precios unitarios de cada componente.

# 3. Desarrollo

## 3.1. Explicación del código

El código completo es el siguiente.

```c
#include <pic14/pic12f683.h>

// Configuracion de bits: desactivar MCLR y WDT
__code unsigned int __at (_CONFIG) configword = _INTOSCIO & _WDTE_OFF & _PWRTE_OFF & _MCLRE_OFF & _CP_ON & _CPD_ON;

// Prototipos
void pausa(unsigned int milis);
void mostrar_dado(unsigned char valor);
void main(void) {
    TRISIO = 0b00100000; // GP5 como entrada, demas como salidas
    GPIO = 0x00;

    unsigned int contador = 0;
    unsigned char valor_dado = 0;

    while (1) {
        contador++;

        if (GP5) {
            pausa(2); // Antirebote
            while (GP5); // Espera a que se suelte

            // "Aleatoriedad" basica basada en contador
            valor_dado = (contador % 6) + 1;

            mostrar_dado(valor_dado);

            pausa(500); // Tiempo visible
            GPIO = 0x00;
        }
    }
}
// Muestra el numero en LEDs segun distribucion personalizada
void mostrar_dado(unsigned char valor) {
    switch (valor) {
        case 1:
            GPIO = 0b00000100; // GP2 (P2) = centro
            break;
        case 2:
            GPIO = 0b00010000; // GP4 (P4)
            break;
        case 3:
            GPIO = 0b00010100; // GP4 + GP2
            break;
        case 4:
            GPIO = 0b00010001; // GP0 + GP4
            break;
        case 5:
            GPIO = 0b00010101; // GP0 + GP4 + GP2
            break;
        case 6:
            GPIO = 0b00010011; // GP0 + GP1 + GP4
            break;
        default:
            GPIO = 0x00;
            break;
    }
}
// Delay aproximado en ms (no preciso)
void pausa(unsigned int milis) {
    unsigned int i, j;
    for (i = 0; i < milis; i++) {
        for (j = 0; j < 1275; j++);
    }
}
```

Código 1: Código para simulación del DADO electrónico

Ahora se desglosará y explicara parte por parte el código implementado.

### 3.1.1. Configuración y operación del microcontrolador PIC12F683

```
#include <pic14/pic12f683.h>

__code unsigned int __at (_CONFIG) configword = _INTOSCIO & _WDTE_OFF & _PWRTE_OFF &
    _MCLRE_OFF & _CP_ON & _CPD_ON;
```

Código 2: Inclusión de biblioteca y configuración del microcontrolador

Esta sección del código es fundamental para la correcta configuración y operación del microcontrolador PIC12F683. La inclusión de la biblioteca pic14/pic12f683.h es crucial ya que proporciona todas las definiciones de hardware específicas, tales como registros y funciones especiales que permiten un manejo más efectivo y directo del microcontrolador.

La línea de configuración define varios parámetros críticos para el comportamiento del microcontrolador:

- _INTOSCIO: Configura el oscilador interno del chip para su funcionamiento y para usar los pines del microcontrolador para entrada/salida, lo que simplifica el diseño al no requerir un oscilador externo.
- _WDTE_OFF: Deshabilita el watchdog timer, un componente crítico para aplicaciones en tiempo real pero no necesario en este contexto, lo que evita reinicios inesperados del microcontrolador si el software entra en un bucle infinito.
- _PWRTE_OFF: Desactiva el timer de encendido, que normalmente añade un retraso al inicio del microcontrolador para estabilizar el suministro de energía.
- _MCLRE_OFF: Elimina la funcionalidad de reset externo del pin MCLR, permitiendo su uso como entrada/salida digital, maximizando así los recursos del dispositivo.
- _CP_ON, _CPD_ON: Activa la protección de código y datos para prevenir manipulaciones y lecturas no autorizadas, protegiendo el software embarcado.

### 3.1.2. Función principal y control de flujo

```
void main(void) {
    TRISIO = 0b00100000;  // Configura GP5 como entrada y el resto como salidas
    GPIO = 0x00;          // Inicializa todos los pines de salida a bajo

    unsigned int contador = 0;
    unsigned char valor_dado = 0;

    while (1) {
        contador++;
        if (GP5) {
            pausa(2);         // Implementa un antirebote simple
            while (GP5);      // Espera hasta que el boton se libere
            valor_dado = (contador % 6) + 1;  // Genera un numero pseudoaleatorio
            mostrar_dado(valor_dado);        // Muestra el valor en los LEDs
            pausa(500);       // Mantiene los LEDs encendidos por un tiempo visible
            GPIO = 0x00;      // Apaga todos los LEDs antes del proximo ciclo
        }
    }
}
```

Código 3: Función principal y manejo de entrada/salida

La función main define el comportamiento principal del programa. Utiliza TRISIO para configurar los pines del microcontrolador, especificando GP5 como entrada para el botón y el resto como salidas para controlar los LEDs. Esto permite interactuar con el hardware de manera efectiva y directa.

El bucle infinito while (1) es típico en programación de microcontroladores, destinado a mantener el programa en ejecución continua. Dentro de este bucle, se incrementa un contador que sirve como base para la generación de un número pseudoaleatorio, aprovechando la naturaleza indeterminada del tiempo de respuesta del usuario al presionar el botón.

La condición if (GP5) detecta si el botón está presionado. En caso afirmativo, se llama a la función ‘pausa‘ para mitigar el efecto de rebote del botón, una simplificación común en proyectos didácticos. Posteriormente, se genera un número aleatorio basado en el contador, que luego se muestra utilizando la función mostrar_dado. Los LEDs se apagan tras una breve pausa para preparar el sistema para el siguiente uso.

### 3.1.3. Funciones de soporte

```
void mostrar_dado(unsigned char valor) {...}
void pausa(unsigned int milis) {...}
```

Código 4: Funciones de visualización y pausa

Las funciones mostrar_dado y pausa ofrecen funcionalidades clave para el sistema. mostrar_dado utiliza un switch para controlar qué LEDs se encienden en función del número generado, simulando las caras de un dado. pausa ofrece una forma simple de introducir retrasos, crucial para la interacción humana y el control del antirebote del botón.

### 3.1.4. Descripción de la función mostrar_dado

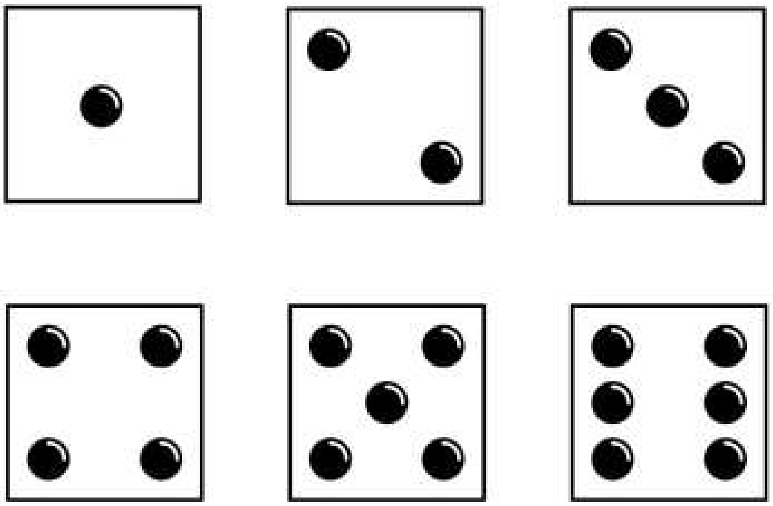


Figura 3: Caras del diseño de DADO [ref].

La función mostrar_dado utiliza la variable valor para determinar cuáles LEDs deben encenderse, según el valor del dado. Cada LED está conectado a un pin específico del PIC12F683, y se controla escribiendo un valor en el registro GPIO. El registro TRISIO configura qué pines son de entrada y cuáles de salida, con 1 indicando entrada y 0 salida.

Para entender la configuración, primero se identifica cada pin del PIC12F683 que está involucrado:

- GP0
- GP1
- GP2
- GP4

Ahora, se establece qué patrón de bits (0 para apagado y 1 para encendido) corresponde a cada número del dado:

| Número del Dado | GP0 | GP1 | GP2 | GP4 | Descripción LED |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | Solo el LED central está encendido (GP2). |
| 2 | 0 | 0 | 0 | 1 | Solo un LED en GP4 está encendido. |
| 3 | 0 | 0 | 1 | 1 | LEDs en GP2 y GP4 están encendidos. |
| 4 | 1 | 0 | 0 | 1 | LEDs en las esquinas GP0 y GP4 están encendidos. |
| 5 | 1 | 0 | 1 | 1 | LEDs en GP0, GP2, y GP4 están encendidos. |
| 6 | 1 | 1 | 0 | 1 | LEDs en GP0, GP1, y GP4 están encendidos. |

Cuadro 2: Estado de los pines GPIO para diferentes valores del dado

## 3.2. Justificación de cada decisión del diseño

### 3.2.1. Diagrama del circuito

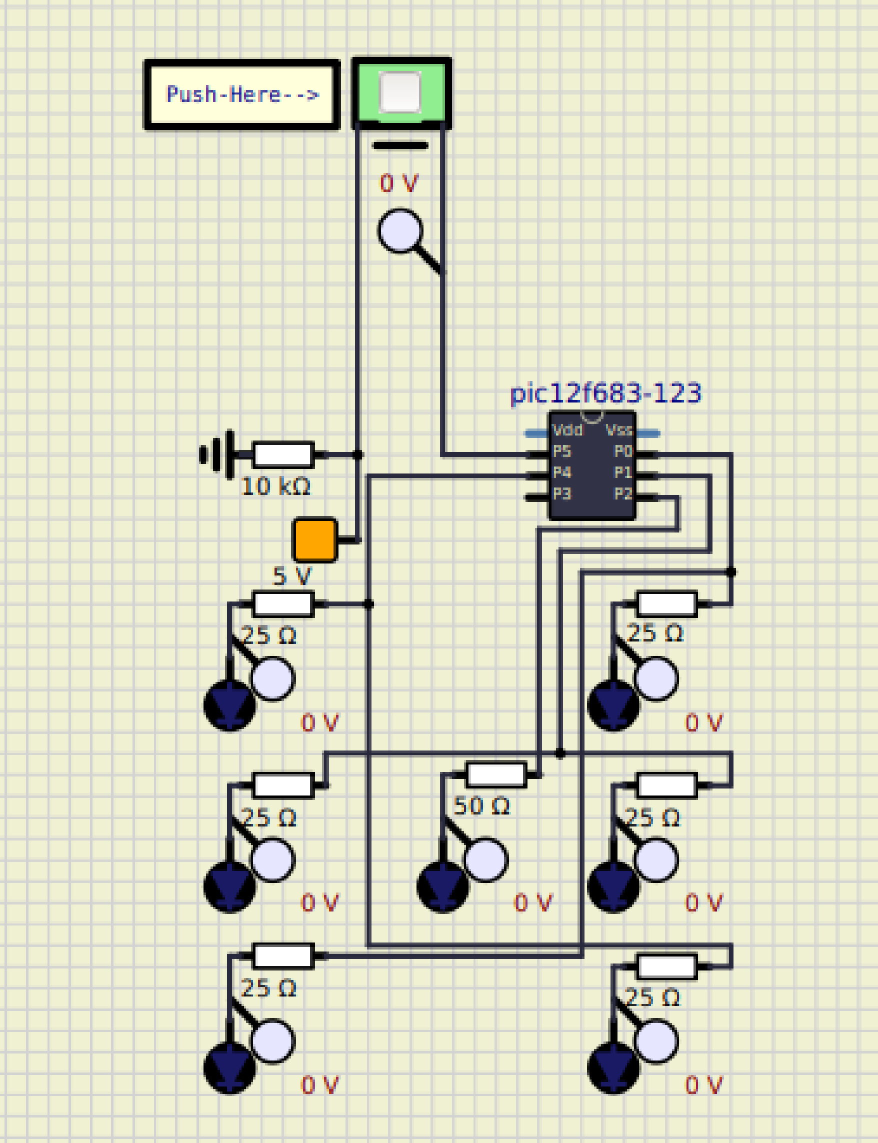


Figura 4: Diagrama del circuito del simulador de DADO electrónico

#### 3.2.2. Resistencias

La elección de las resistencias en el circuito está determinada por la necesidad de proteger los LEDs y asegurar un funcionamiento adecuado del botón:

| Componente | Valor | Propósito |
|---|---|---|
| Resistencia Pull-Down | 10 kΩ | Garantizar que voltaje sea reconocido como bajo sí el botón no está presionado |
| Resistencias para LEDs | 25 Ω | Limitar la corriente a través de los LEDs para evitar daños |
| Resistencia central para LED | 50 Ω | Balancear la corriente en el LED central para igualar el brillo |

Cuadro 3: Valores y propósitos de las resistencias utilizadas en el circuito

#### 3.2.3. Botón de entrada

El botón momentáneo inicia la generación de números. Utiliza una resistencia de pull-down de 10 kΩ para asegurar que el pin de entrada del microcontrolador lea un nivel bajo claro cuando el botón no está presionado, evitando así entradas falsas debido a ruido eléctrico o rebotes.

#### 3.2.4. LEDs y configuración de resistencias

Cada LED está conectado a una resistencia limitadora para proteger contra la sobrecorriente. Las resistencias de 25 Ω son adecuadas para limitar la corriente manteniendo un brillo óptimo bajo una alimentación de 5V. El LED central tiene una resistencia ligeramente mayor (50 Ω) para ajustar el brillo y mantener la uniformidad visual entre todos los LEDs.

#### 3.2.5. Microcontrolador y configuración general

El microcontrolador PIC12F683 es elegido por su simplicidad, bajo costo y adecuación para controlar pequeños dispositivos como los LEDs en este proyecto. Funciona eficientemente bajo un voltaje de alimentación de 5V, común en circuitos embebidos de baja potencia.

#### 3.2.6. Video de muestra del funcionamiento

- https://youtu.be/yAgO76Tneac

# 4. Análisis de resultados

## 4.1. Figuras de resultados

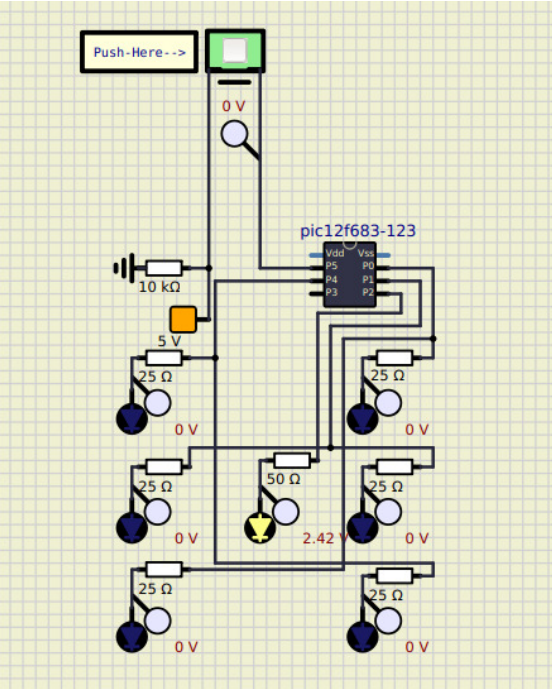


Figura 5: Cara de DADO: Número 1 [SimulIDE].

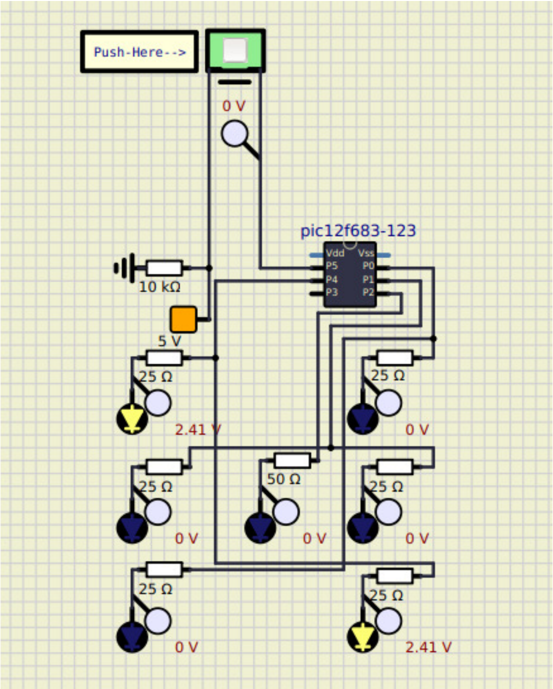


Figura 6: Cara de DADO: Número 2 [SimulIDE].

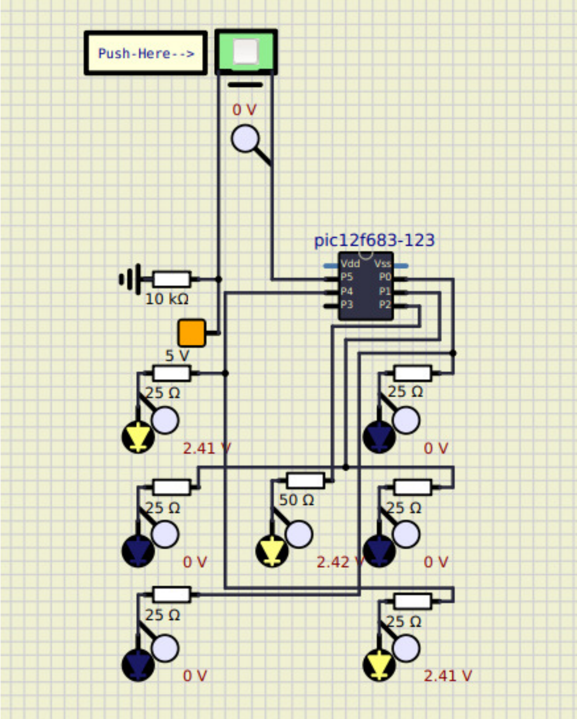


Figura 7: Cara de DADO: Número 3 [SimulIDE].

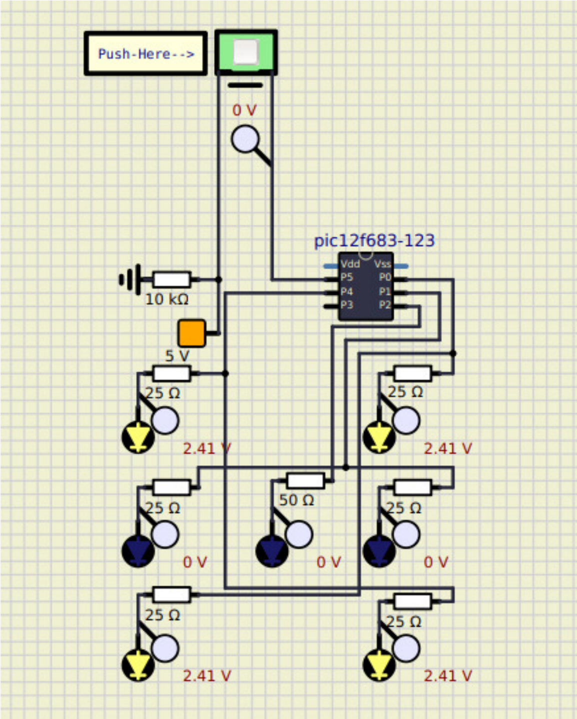


Figura 8: Cara de DADO: Número 4 [SimulIDE].

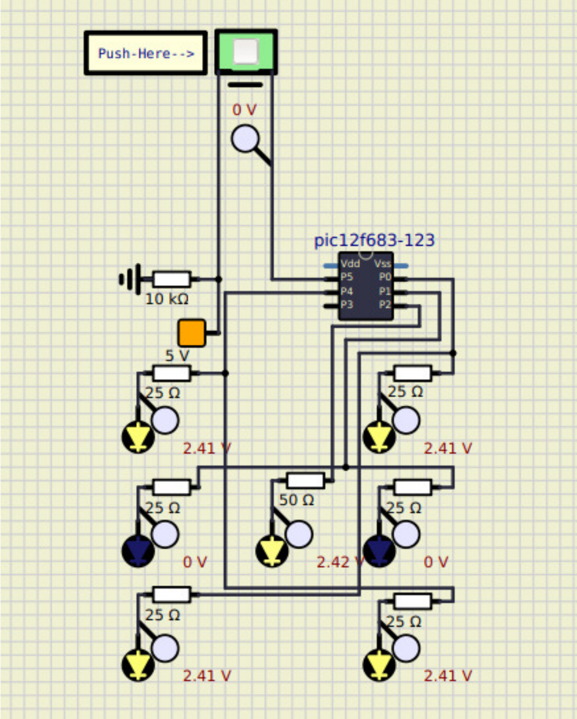


Figura 9: Cara de DADO: Número 5 [SimulIDE].

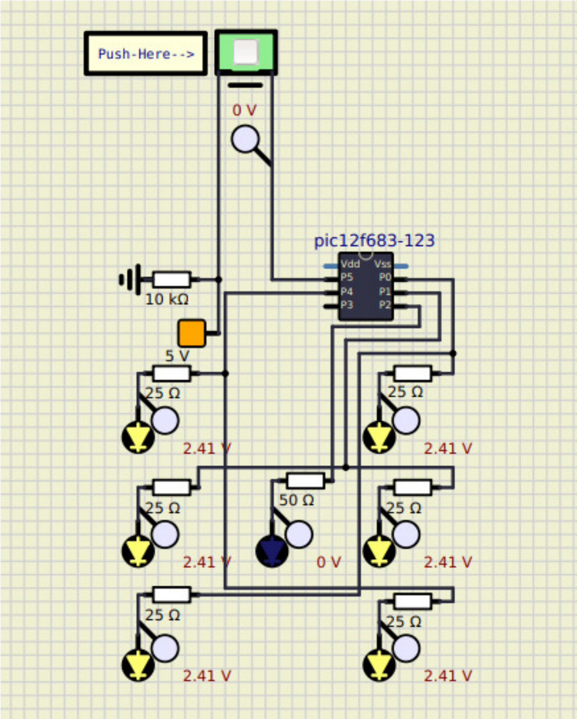


Figura 10: Cara de DADO: Número 6 [SimulIDE].

## 4.2. Funcionamiento del circuito y mediciones de voltaje

Durante las pruebas del circuito simulador de dado, se observaron los siguientes voltajes en los LEDs cuando el botón fue presionado, indicando la activación de los diferentes LEDs según el número generado aleatoriamente por el microcontrolador:

| Número del Dado | LEDs Activados | Voltaje Medido (V) |
|---|---|---|
| 1 | Centro (GP2) | 2.42 |
| 2 | Esquina (GP4) | 2.41 |
| 3 | Centro, Esquina (GP2, GP4) | 2.42, 2.41, 2.41 |
| 4 | Lados (GP0, GP4) | 2.41, 2.41 |
| 5 | Centro, Lados (GP0, GP2, GP4) | 2.42, 2.41, 2.41, 2.41 |
| 6 | Tres esquinas (GP0, GP1, GP4) | 2.41, 2.41, 2.41 |

Cuadro 4: Mediciones de voltaje en los LEDs para diferentes números generados

## 4.3. Análisis de voltajes medidos

Los voltajes medidos en cada uno de los LEDs cuando se generaron diferentes números en el dado fueron consistentemente alrededor de 2.41V. Esta pequeña variación puede atribuirse a la resistencia interna del circuito y los componentes utilizados, lo cual es normal en la mayoría de los circuitos electrónicos prácticos.

## 4.4. Evaluación de la funcionalidad del circuito

El funcionamiento del circuito fue conforme a lo esperado. Cada vez que se presionó el botón, los LEDs correspondientes al número generado se iluminaron correctamente, demostrando que el microcontrolador y el código están funcionando como se diseñaron. La pequeña diferencia en el voltaje no afectó la funcionalidad general del dispositivo.

# 5. Conclusiones

En el Laboratorio 1, se logró desarrollar un simulador de dado simplificado que incorporó el manejo eficiente de GPIOs y la generación de números aleatorios utilizando el microcontrolador PIC12F683. Se pudo demostrar que el sistema responde adecuadamente al input del usuario, generando una salida visual coherente en los LEDs que representa las caras del dado. A través del desarrollo y la simulación, se validaron las funciones de generación de números aleatorios y el control de los LEDs, destacando la capacidad del PIC12F683 para manejar tareas de entrada/salida de propósito general.

El análisis de la funcionalidad del programa mostró que el código es robusto y maneja correctamente las interrupciones y estados del microcontrolador. Las medidas realizadas con multímetros y osciloscopios confirmaron que los componentes electrónicos operan dentro de los parámetros esperados, asegurando así la integridad del circuito y la seguridad del dispositivo. Además, el uso de Git para el control de versiones demostró un desarrollo estructurado y documentado del código, permitiendo una fácil trazabilidad de los cambios efectiva.

# 6. Recomendaciones

1. Revisión de componentes: Sería prudente revisar periódicamente la disponibilidad y los costos de los componentes utilizados, para garantizar que el proyecto pueda replicarse o modificarse económicamente en futuras iteraciones.

2. Optimización del código: Aunque el código actual es funcional, siempre existe la posibilidad de optimizar. Se podrían explorar métodos para reducir el uso de memoria o aumentar la eficiencia del procesador, lo que podría ser particularmente útil en aplicaciones más complejas que utilicen este microcontrolador.

3. Pruebas adicionales: Realizar pruebas adicionales en diferentes condiciones ambientales para asegurar la robustez del dispositivo ante variaciones de temperatura y humedad, que podrían afectar el rendimiento de los componentes electrónicos. Pero en este laboratorio todo se manejó por medio del simulador.

4. Expandir la funcionalidad: Considerar la expansión de la funcionalidad del dado para incluir características como memoria de tiradas anteriores, o interfaces de usuario más complejas que podrían incluir pantallas o sonidos.

5. Documentación y capacitación: Mantener la documentación actualizada y realizar sesiones de capacitación para nuevos estudiantes o usuarios del laboratorio, asegurando que puedan replicar o modificar el proyecto sin dificultades.

# Referencias

[1] L. Alley. *Microcontroller Interfacing Revolution*. 2018.

[2] Microchip Technology Inc. *PIC12F683 Data Sheet, 8-Pin Flash-Based 8-Bit CMOS Microcontroller*. 2009.

[3] STMicroelectronics. *Microcontrollers Made Easy*. 2020.

[4] M. Verle. *PIC Microcontrollers. MikroElektronika*. 2007.

[5] Pete Warden y Daniel Situnayake. *TinyML, O'Reilly*. 2019.

# 7. APENDICE

# PIC12F683 Data Sheet

## 8-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

* 8-bit, 8-pin Devices Protected by Microchip's Low Pin Count Patent: U.S. Patent No. 5,847,450. Additional U.S. and foreign patents and applications may be issued or pending.

**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.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip’s products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

# PIC12F683

# 8-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

## High-Performance RISC CPU

- Only 35 instructions to learn:
  - All single-cycle instructions except branches
- Operating speed:
  - DC – 20 MHz oscillator/clock input
  - DC – 200 ns instruction cycle
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

## Special Microcontroller Features

- Precision Internal Oscillator:
  - Factory calibrated to ±1%
  - Software selectable frequency range of 8 MHz to 31 kHz
  - Two-speed Start-up mode
  - Crystal fail detect for critical applications
  - Clock mode switching during operation for power savings
- Power-saving Sleep mode
- Wide operating voltage range. (2.0V-5.5V)
- Industrial and Extended temperature range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Multiplexed Master Clear with pull-up/input pin
- Programmable code protection
- High Endurance Flash/EEPROM cell:
  - 100,000 write Flash endurance
  - 1,000,000 write EEPROM endurance
  - Flash/Data EEPROM Retention: > 40 years

## Low-Power Features

- Standby Current:
  - 1 nA @ 2.0V, typical
- Operating Current:
  - 8.5 μA @ 32 kHz, 2.0V, typical
  - 100 μA @ 1 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 1 μA @ 2.0V, typical

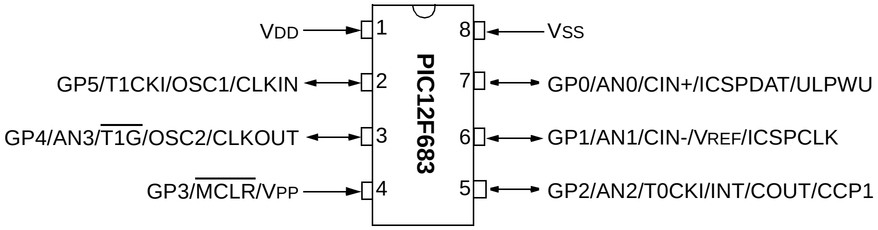
## Peripheral Features

- 6 I/O pins with individual direction control:
  - High current source/sink for direct LED drive
  - Interrupt-on-pin change
  - Individually programmable weak pull-ups
  - Ultra Low-Power Wake-up on GP0
- Analog comparator module with:
  - One analog comparator
  - Programmable on-chip voltage reference (CVREF) module (% of VDD)
  - Comparator inputs and output externally accessible
- A/D Converter:
  - 10-bit resolution and 4 channels
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Enhanced Timer1:
  - 16-bit timer/counter with prescaler
  - External Gate Input mode
  - Option to use OSC1 and OSC2 in LP mode as Timer1 oscillator if INTOSC mode selected
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Capture, Compare, PWM module:
  - 16-bit Capture, max resolution 12.5 ns
  - Compare, max resolution 200 ns
  - 10-bit PWM, max frequency 20 kHz
- In-Circuit Serial Programming™ (ICSP™) via two pins

| Device | Program Memory | Data Memory | | I/O | 10-bit A/D (ch) | Comparators | Timers 8/16-bit |
|---|---|---|---|---|---|---|---|
| | Flash (words) | SRAM (bytes) | EEPROM (bytes) | | | | |
| PIC12F683 | 2048 | 128 | 256 | 6 | 4 | 1 | 2/1 |

## Pin Diagram

8-pin PDIP, SOIC, DFN-S


VDD
GP5/T1CKI/OSC1/CLKIN
GP4/AN3/$\overline{\text{T1G}}$/OSC2/CLKOUT
GP3/$\overline{\text{MCLR}}$/VPP
1
2
3
4
PIC12F683
8
7
6
5
VSS
GP0/AN0/CIN+/ICSPDAT/ULPWU
GP1/AN1/CIN-/VREF/ICSPCLK
GP2/AN2/T0CKI/INT/COUT/CCP1

## 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@mail.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)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, 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/cn** to receive the most current information on all of our products.

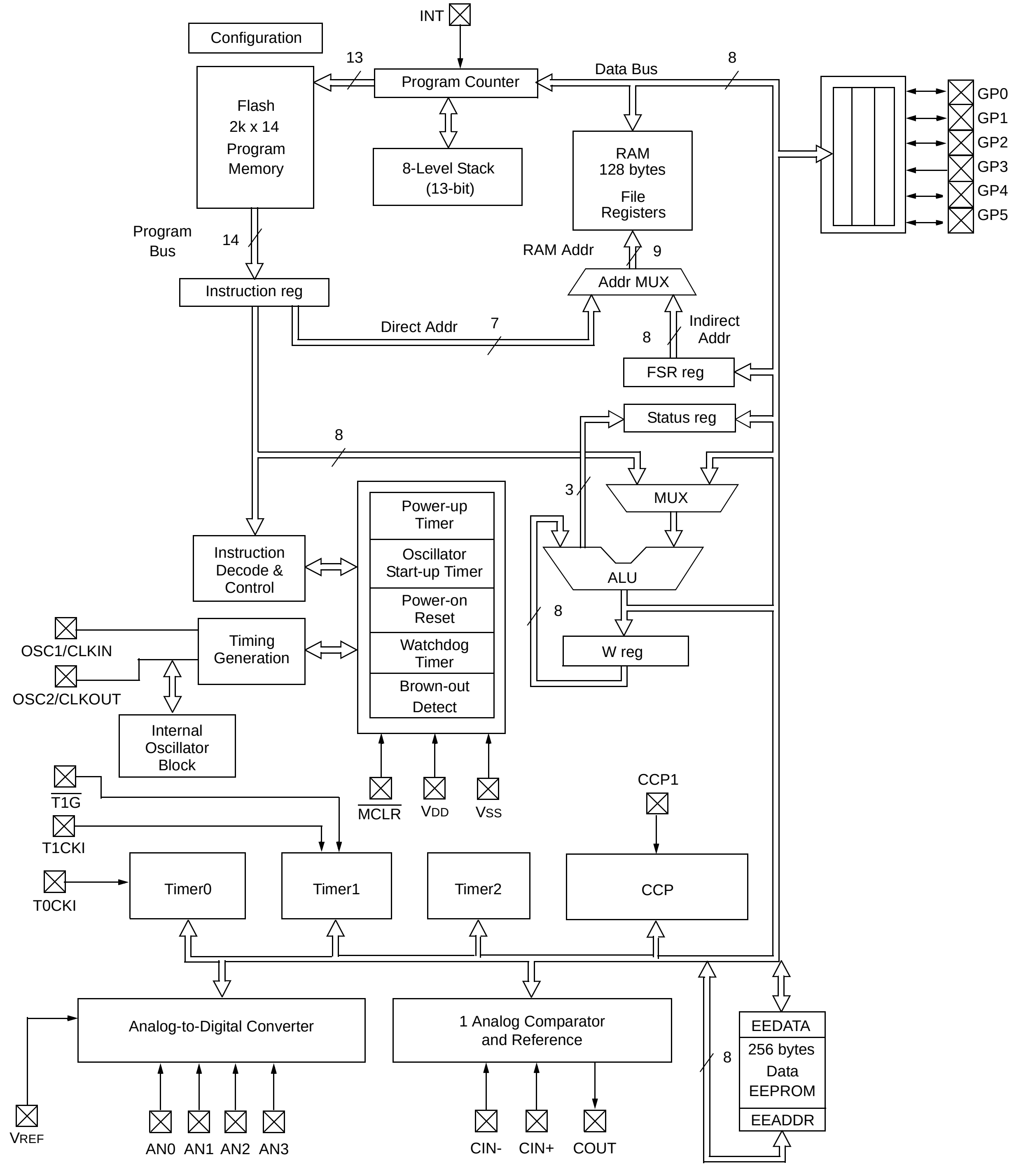
NOTES:

## 1.0 DEVICE OVERVIEW

This document contains device specific information for the PIC12F683. Additional information may be found in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The reference manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

The PIC12F683 is covered by this data sheet. It is available in 8-pin PDIP, SOIC and DFN-S packages. Figure 1-1 shows a block diagram of the PIC12F683 device. Table 1-1 shows the pinout description.

**FIGURE 1-1: PIC12F683 BLOCK DIAGRAM**



### TABLE 1-1: PIC12F683 PINOUT DESCRIPTION

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| VDD | VDD | Power | — | Positive supply |
| GP5/T1CKI/OSC1/CLKIN | GP5 | TTL | CMOS | GPIO I/O w/programmable pull-up and interrupt-on-change |
| | T1CKI | ST | — | Timer1 clock |
| | OSC1 | XTAL | — | Crystal/Resonator |
| | CLKIN | ST | — | External clock input/RC oscillator connection |
| GP4/AN3/$\overline{\mathrm{T1G}}$/OSC2/CLKOUT | GP4 | TTL | CMOS | GPIO I/O w/programmable pull-up and interrupt-on-change |
| | AN3 | AN | — | A/D Channel 3 input |
| | $\overline{\mathrm{T1G}}$ | ST | — | Timer1 gate |
| | OSC2 | — | XTAL | Crystal/Resonator |
| | CLKOUT | — | CMOS | FOSC/4 output |
| GP3/$\overline{\mathrm{MCLR}}$/VPP | GP3 | TTL | — | GPIO input with interrupt-on-change |
| | $\overline{\mathrm{MCLR}}$ | ST | — | Master Clear w/internal pull-up |
| | VPP | HV | — | Programming voltage |
| GP2/AN2/T0CKI/INT/COUT/CCP1 | GP2 | ST | CMOS | GPIO I/O w/programmable pull-up and interrupt-on-change |
| | AN2 | AN | — | A/D Channel 2 input |
| | T0CKI | ST | — | Timer0 clock input |
| | INT | ST | — | External Interrupt |
| | COUT | — | CMOS | Comparator 1 output |
| | CCP1 | ST | CMOS | Capture input/Compare output/PWM output |
| GP1/AN1/CIN-/VREF/ICSPCLK | GP1 | TTL | CMOS | GPIO I/O w/programmable pull-up and interrupt-on-change |
| | AN1 | AN | — | A/D Channel 1 input |
| | CIN- | AN | — | Comparator 1 input |
| | VREF | AN | — | External Voltage Reference for A/D |
| | ICSPCLK | ST | — | Serial Programming Clock |
| GP0/AN0/CIN+/ICSPDAT/ULPWU | GP0 | TTL | CMOS | GPIO I/O w/programmable pull-up and interrupt-on-change |
| | AN0 | AN | — | A/D Channel 0 input |
| | CIN+ | AN | — | Comparator 1 input |
| | ICSPDAT | ST | CMOS | Serial Programming Data I/O |
| | ULPWU | AN | — | Ultra Low-power Wake-up input |
| VSS | VSS | Power | — | Ground reference |

**Legend:**

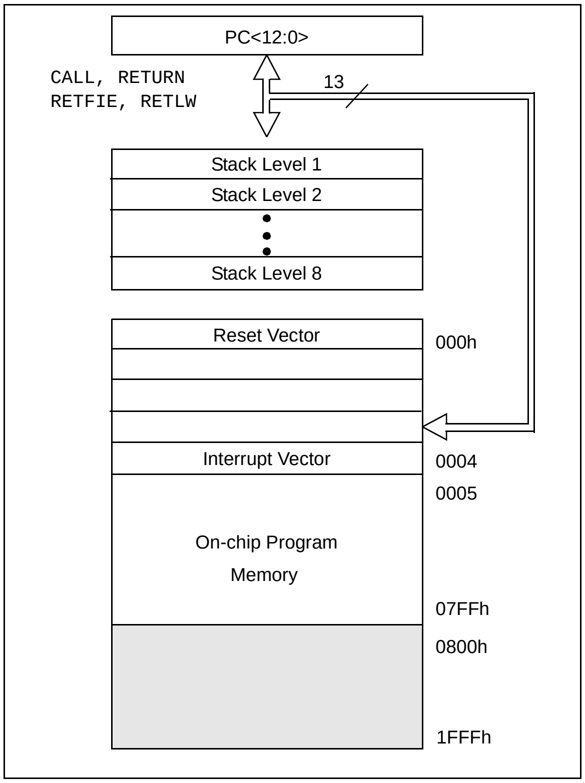
| | | |
|---|---|---|
| AN = Analog input or output | CMOS = CMOS compatible input or output |
| TTL = TTL compatible input | ST = Schmitt Trigger input with CMOS levels |
| HV = High Voltage | XTAL = Crystal |

# 2.0 MEMORY ORGANIZATION

## 2.1 Program Memory Organization

The PIC12F683 has a 13-bit program counter capable of addressing an 8k x 14 program memory space. Only the first 2k x 14 (0000h-07FFh) for the PIC12F683 is physically implemented. Accessing a location above these boundaries will cause a wrap around within the first 2k x 14 space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figure 2-1).

**FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC12F683**



## 2.2 Data Memory Organization

The data memory (see Figure 2-2) is partitioned into two banks, which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). The Special Function Registers are located in the first 32 locations of each bank. Register locations 20h-7Fh in Bank 0 and A0h-BFh in Bank 1 are general purpose registers, implemented as static RAM. Register locations F0h-FFh in Bank 1 point to addresses 70h-7Fh in Bank 0. All other RAM is unimplemented and returns '0' when read. RP0 (Status<5>) is the bank select bit.

- RP0 = 0: Bank 0 is selected
- RP0 = 1: Bank 1 is selected

**Note:** The IRP and RP1 bits (Status<7:6>) are reserved and should always be maintained as '0's.

### 2.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 128 x 8 in the PIC12F683. Each register is accessed, either directly or indirectly, through the File Select Register FSR (see **Section 2.4 "Indirect Addressing, INDF and FSR Registers"**).

### 2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (see Table 2-1). These registers are static RAM.

The special registers can be classified into two sets: core and peripheral. The Special Function Registers associated with the "core" are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

**FIGURE 2-2: DATA MEMORY MAP OF THE PIC12F683**

| | File Address | | File Address |
|---|---|---|---|
| Indirect addr.[(1)] | 00h | Indirect addr.[(1)] | 80h |
| TMR0 | 01h | OPTION_REG | 81h |
| PCL | 02h | PCL | 82h |
| STATUS | 03h | STATUS | 83h |
| FSR | 04h | FSR | 84h |
| GPIO | 05h | TRISIO | 85h |
| ▨ | 06h | ▨ | 86h |
| ▨ | 07h | ▨ | 87h |
| ▨ | 08h | ▨ | 88h |
| ▨ | 09h | ▨ | 89h |
| PCLATH | 0Ah | PCLATH | 8Ah |
| INTCON | 0Bh | INTCON | 8Bh |
| PIR1 | 0Ch | PIE1 | 8Ch |
| ▨ | 0Dh | ▨ | 8Dh |
| TMR1L | 0Eh | PCON | 8Eh |
| TMR1H | 0Fh | OSCCON | 8Fh |
| T1CON | 10h | OSCTUNE | 90h |
| TMR2 | 11h | ▨ | 91h |
| T2CON | 12h | PR2 | 92h |
| CCPR1L | 13h | ▨ | 93h |
| CCPR1H | 14h | ▨ | 94h |
| CCP1CON | 15h | WPU | 95h |
| ▨ | 16h | IOC | 96h |
| ▨ | 17h | ▨ | 97h |
| WDTCON | 18h | ▨ | 98h |
| CMCON0 | 19h | VRCON | 99h |
| CMCON1 | 1Ah | EEDAT | 9Ah |
| ▨ | 1Bh | EEADR | 9Bh |
| ▨ | 1Ch | EECON1 | 9Ch |
| ▨ | 1Dh | EECON2[(1)] | 9Dh |
| ADRESH | 1Eh | ADRESL | 9Eh |
| ADCON0 | 1Fh | ANSEL | 9Fh |
| General Purpose Registers<br>96 Bytes | 20h | General Purpose Registers<br>32 Bytes | A0h |
| | | | BFh |
| | | ▨ | |
| | | Accesses 70h-7Fh | F0h |
| | 7Fh | | FFh |
| BANK 0 | | BANK 1 | |

▨ Unimplemented data memory locations, read as '0'.

**Note 1:** Not a physical register.

### TABLE 2-1: PIC12F683 SPECIAL REGISTERS SUMMARY BANK 0

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Bank 0 | | | | | | | | | | | |
| 00h | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | `xxxx xxxx` | 17, 83 |
| 01h | TMR0 | Timer0 Module's Register | | | | | | | | `xxxx xxxx` | 39, 83 |
| 02h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | `0000 0000` | 17, 83 |
| 03h | STATUS | IRP(1) | RP1(1) | RP0 | $\overline{\text{TO}}$ | PD | Z | DC | C | `0001 1xxx` | 11, 83 |
| 04h | FSR | Indirect Data Memory Address Pointer | | | | | | | | `xxxx xxxx` | 17, 83 |
| 05h | GPIO | — | — | GP5 | GP4 | GP3 | GP2 | GP1 | GP0 | `--xx xxxx` | 31, 83 |
| 06h | — | Unimplemented | | | | | | | | — | — |
| 07h | — | Unimplemented | | | | | | | | — | — |
| 08h | — | Unimplemented | | | | | | | | — | — |
| 09h | — | Unimplemented | | | | | | | | — | — |
| 0Ah | PCLATH | — | — | — | Write Buffer for upper 5 bits of Program Counter | | | | | `---0 0000` | 17, 83 |
| 0Bh | INTCON | GIE | PEIE | T0IE | INTE | GPIE | T0IF | INTF | GPIF | `0000 0000` | 13, 83 |
| 0Ch | PIR1 | EEIF | ADIF | CCP1IF | — | CMIF | OSFIF | TMR2IF | TMR1IF | `000- 0000` | 15, 83 |
| 0Dh | — | Unimplemented | | | | | | | | — | — |
| 0Eh | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 | | | | | | | | `xxxx xxxx` | 41, 83 |
| 0Fh | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 | | | | | | | | `xxxx xxxx` | 41, 83 |
| 10h | T1CON | T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | `0000 0000` | 43, 83 |
| 11h | TMR2 | Timer2 Module Register | | | | | | | | `0000 0000` | 45, 83 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | `-000 0000` | 45, 83 |
| 13h | CCPR1L | Capture/Compare/PWM Register 1 Low Byte | | | | | | | | `xxxx xxxx` | 70, 83 |
| 14h | CCPR1H | Capture/Compare/PWM Register 1 High Byte | | | | | | | | `xxxx xxxx` | 70, 83 |
| 15h | CCP1CON | — | — | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | `--00 0000` | 69, 83 |
| 16h | — | Unimplemented | | | | | | | | — | — |
| 17h | — | Unimplemented | | | | | | | | — | — |
| 18h | WDTCON | — | — | — | WDTPS3 | WDTPS2 | WDTPS1 | WDTPS0 | SWDTEN | `---0 1000` | 90, 83 |
| 19h | CMCON0 | — | COUT | — | CINV | CIS | CM2 | CM1 | CM0 | `-0-0 0000` | 47, 83 |
| 1Ah | CMCON1 | — | — | — | — | — | — | T1GSS | CMSYNC | `---- --10` | 50, 83 |
| 1Bh | — | Unimplemented | | | | | | | | — | — |
| 1Ch | — | Unimplemented | | | | | | | | — | — |
| 1Dh | — | Unimplemented | | | | | | | | — | — |
| 1Eh | ADRESH | Most Significant 8 bits of the left shifted A/D result or 2 bits of right shifted result | | | | | | | | `xxxx xxxx` | 57,83 |
| 1Fh | ADCON0 | ADFM | VCFG | — | — | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON | `00-- 0000` | 58,83 |

**Legend:** — = unimplemented locations read as '`0`', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

**Note 1:** IRP and RP1 bits are reserved, always maintain these bits clear.

### TABLE 2-2: PIC12F683 SPECIAL FUNCTION REGISTERS SUMMARY BANK 1

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Bank 1 | | | | | | | | | | | |
| 80h | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | 17, 83 |
| 81h | OPTION_REG | $\overline{\text{GPPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 12, 83 |
| 82h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 17, 83 |
| 83h | STATUS | IRP(1) | RP1(1) | RP0 | $\overline{\text{TO}}$ | PD | Z | DC | C | 0001 1xxx | 11, 83 |
| 84h | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | 17, 83 |
| 85h | TRISIO | — | — | TRISIO5 | TRISIO4 | TRISIO3 | TRISIO2 | TRISIO1 | TRISIO0 | --11 1111 | 32, 83 |
| 86h | — | Unimplemented | | | | | | | | — | — |
| 87h | — | Unimplemented | | | | | | | | — | — |
| 88h | — | Unimplemented | | | | | | | | — | — |
| 89h | — | Unimplemented | | | | | | | | — | — |
| 8Ah | PCLATH | — | — | — | Write Buffer for upper 5 bits of Program Counter | | | | | ---0 0000 | 17, 83 |
| 8Bh | INTCON | GIE | PEIE | T0IE | INTE | GPIE | T0IF | INTF | GPIF | 0000 0000 | 13, 83 |
| 8Ch | PIE1 | EEIE | ADIE | CCP1IE | — | CMIE | OSFIE | TMR2IE | TMR1IE | 000- 0000 | 14, 83 |
| 8Dh | — | Unimplemented | | | | | | | | — | — |
| 8Eh | PCON | — | — | ULPWUE | SBODEN | — | — | $\overline{\text{POR}}$ | $\overline{\text{BOD}}$ | --01 --qq | 16, 83 |
| 8Fh | OSCCON | — | IRCF2 | IRCF1 | IRCF0 | OSTS(2) | HTS | LTS | SCS | -110 x000 | 28, 83 |
| 90h | OSCTUNE | — | — | — | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 | ---0 0000 | 23, 83 |
| 91h | — | Unimplemented | | | | | | | | — | — |
| 92h | PR2 | Timer2 Module Period Register | | | | | | | | 1111 1111 | 45, 83 |
| 93h | — | Unimplemented | | | | | | | | — | — |
| 94h | — | Unimplemented | | | | | | | | — | — |
| 95h | WPU(3) | — | — | WPU5 | WPU4 | — | WPU2 | WPU1 | WPU0 | --11 -111 | 32, 83 |
| 96h | IOC | — | — | IOC5 | IOC4 | IOC3 | IOC2 | IOC1 | IOC0 | --00 0000 | 33, 83 |
| 97h | — | Unimplemented | | | | | | | | — | — |
| 98h | — | Unimplemented | | | | | | | | — | — |
| 99h | VRCON | VREN | — | VRR | — | VR3 | VR2 | VR1 | VR0 | 0-0- 0000 | 53, 83 |
| 9Ah | EEDAT | EEDAT7 | EEDAT6 | EEDAT5 | EEDAT4 | EEDAT3 | EEDAT2 | EEDAT1 | EEDAT0 | 0000 0000 | 65, 83 |
| 9Bh | EEADR | EEADR7 | EEADR6 | EEADR5 | EEADR4 | EEADR3 | EEADR2 | EEADR1 | EEADR0 | 0000 0000 | 65, 83 |
| 9Ch | EECON1 | — | — | — | — | WRERR | WREN | WR | RD | ---- x000 | 66, 84 |
| 9Dh | EECON2 | EEPROM Control Register 2 (not a physical register) | | | | | | | | ---- ---- | 66, 84 |
| 9Eh | ADRESL | Least Significant 2 bits of the left shifted result or 8 bits of the right shifted result | | | | | | | | xxxx xxxx | 57, 84 |
| 9Fh | ANSEL | — | ADCS2 | ADCS1 | ADCS0 | ANS3 | ANS2 | ANS1 | ANS0 | -000 1111 | 59, 84 |

**Legend:** — = unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

**Note 1:** IRP and RP1 bits are reserved, always maintain these bits clear.

**2:** OSCCON<OSTS> bit reset to '0' with Dual Speed Start-up and LP, HS or XT selected as the oscillator.

**3:** GP3 pull-up is enabled when MCLRE is '1' in the Configuration Word register.

### 2.2.2.1 Status Register

The Status register, shown in Register 2-1, contains:

- Arithmetic status of the ALU
- Reset status
- Bank select bits for data memory (SRAM)

The Status register can be the destination for any instruction, like 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 $\overline{\text{TO}}$ and 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).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the Status register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits, see the “Instruction Set Summary”.

**Note 1:** Bits IRP and RP1 (Status<7:6>) are not used by the PIC12F683 and should be maintained as clear. Use of these bits is not recommended, since this may affect upward compatibility with future products.

**2:** The C and DC bits operate as a $\overline{\text{Borrow}}$ and Digit $\overline{\text{Borrow}}$ out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

**REGISTER 2-1: STATUS – STATUS REGISTER (ADDRESS: 03h OR 83h)**

| Reserved | Reserved | R/W-0 | R-1 | R-1 | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| IRP | RP1 | RP0 | $\overline{\text{TO}}$ | PD | Z | DC | C |
| bit 7 | | | | | | | bit 0 |

bit 7 **IRP:** This bit is reserved and should be maintained as ‘`0`’

bit 6 **RP1:** This bit is reserved and should be maintained as ‘`0`’

bit 5 **RP0:** Register Bank Select bit (used for direct addressing)
`1` = Bank 1 (80h–FFh)
`0` = Bank 0 (00h–7Fh)

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

bit 3 **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/$\overline{\text{borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
For $\overline{\text{borrow}}$, the polarity is reversed.
`1` = A carry-out from the 4th low-order bit of the result occurred
`0` = No carry-out from the 4th low-order bit of the result

bit 0 **C**: Carry/$\overline{\text{borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
`1` = A carry-out from the Most Significant bit of the result occurred
`0` = No carry-out from the Most Significant bit of the result occurred

**Note 1:** For $\overline{\text{borrow}}$, the polarity is reversed. A subtraction is executed by adding the two’s complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low-order bit of the source register.

| **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 |

### 2.2.2.2 Option Register

The Option register is a readable and writable register, which contains various control bits to configure:

- TMR0/WDT prescaler
- External GP2/INT interrupt
- TMR0
- Weak pull-ups on GPIO

**Note:** To achieve a 1:1 prescaler assignment for TMR0, assign the prescaler to the WDT by setting PSA bit to ‘1’ (Option<3>). See **Section 5.4 “Prescaler”**.

**REGISTER 2-2: OPTION_REG – OPTION REGISTER (ADDRESS: 81h)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{GPPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **$\overline{\text{GPPU}}$:** GPIO Pull-up Enable bit
1 = GPIO pull-ups are disabled
0 = GPIO pull-ups are enabled by individual port latch values in WPU register

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of GP2/INT pin
0 = Interrupt on falling edge of GP2/INT pin

bit 5 **T0CS:** TMR0 Clock Source Select bit
1 = Transition on GP2/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE:** TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on GP2/T0CKI pin
0 = Increment on low-to-high transition on GP2/T0CKI pin

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

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

| Bit Value | TMR0 Rate | WDT Rate[(1)] |
|---|---|---|
| 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 |

**Note 1:** A dedicated 16-bit WDT postscaler is available for the PIC12F683. See **Section 12.6 “Watchdog Timer (WDT)”** for more information.

| **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 |

### 2.2.2.3 INTCON Register

The INTCON register is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, GPIO change and external GP2/INT pin interrupts.

**Note:** Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**REGISTER 2-3: INTCON – INTERRUPT CONTROL REGISTER (ADDRESS: 0Bh OR 8Bh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| GIE | PEIE | T0IE | INTE | GPIE | T0IF | INTF | GPIF |
| bit 7 | | | | | | | bit 0 |

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6 **PEIE:** Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

bit 5 **T0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4 **INTE:** GP2/INT External Interrupt Enable bit
1 = Enables the GP2/INT external interrupt
0 = Disables the GP2/INT external interrupt

bit 3 **GPIE:** GPIO Change Interrupt Enable bit(1)
1 = Enables the GPIO change interrupt
0 = Disables the GPIO change interrupt

bit 2 **T0IF:** TMR0 Overflow Interrupt Flag bit(2)
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INTF:** GP2/INT External Interrupt Flag bit
1 = The GP2/INT external interrupt occurred (must be cleared in software)
0 = The GP2/INT external interrupt did not occur

bit 0 **GPIF:** GPIO Change Interrupt Flag bit
1 = When at least one of the GPIO<5:0> pins changed state (must be cleared in software)
0 = None of the GPIO<5:0> pins have changed state

**Note 1:** IOC register must also be enabled.

**2:** T0IF bit is set when Timer0 rolls over. Timer0 is unchanged on Reset and should be initialized before clearing T0IF bit.

**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 |

### 2.2.2.4 PIE1 Register

The PIE1 register contains the interrupt enable bits, as shown in Register 2-4.

> **Note:** Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

**REGISTER 2-4: PIE1 – PERIPHERAL INTERRUPT ENABLE REGISTER 1 (ADDRESS: 8Ch)**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| EEIE | ADIE | CCP1IE | — | CMIE | OSFIE | TMR2IE | TMR1IE |
| bit 7 | | | | | | | bit 0 |

bit 7 **EEIE:** EE Write Complete Interrupt Enable bit
1 = Enables the EE write complete interrupt
0 = Disables the EE write complete interrupt

bit 6 **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables the A/D converter interrupt
0 = Disables the A/D converter interrupt

bit 5 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 4 **Unimplemented:** Read as '0'

bit 3 **CMIE:** Comparator Interrupt Enable bit
1 = Enables the Comparator 1 interrupt
0 = Disables the Comparator 1 interrupt

bit 2 **OSFIE:** Oscillator Fail Interrupt Enable bit
1 = Enables the oscillator fail interrupt
0 = disables the oscillator fail interrupt

bit 1 **TMR2IE:** Timer 2 to PR2 Match Interrupt Enable bit
1 = Enables the Timer 2 to PR2 match interrupt
0 = Disables the Timer 2 to PR2 match interrupt

bit 0 **TMR1IE:** Timer 1 Overflow Interrupt Enable bit
1 = Enables the Timer 1 overflow interrupt
0 = Disables the Timer 1 overflow interrupt

| **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 |

### 2.2.2.5 PIR1 Register

The PIR1 register contains the interrupt flag bits, as shown in Register 2-5.

| **Note:** | Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. |
|---|---|

**REGISTER 2-5: PIR1 – PERIPHERAL INTERRUPT REQUEST REGISTER 1 (ADDRESS: 0Ch)**

| R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| EEIF | ADIF | CCP1IF | — | CMIF | OSFIF | TMR2IF | TMR1IF |
| bit 7 | | | | | | | bit 0 |

bit 7 **EEIF**: EEPROM Write Operation Interrupt Flag bit

1 = The write operation completed (must be cleared in software)
0 = The write operation has not completed or has not been started

bit 6 **ADIF**: A/D Interrupt Flag bit

1 = A/D conversion complete
0 = A/D conversion has not completed or has not been started

bit 5 **CCP1IF:** CCP1 Interrupt Flag bit

Capture mode:
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred

Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred

PWM mode:
Unused in this mode.

bit 4 **Unimplemented:** Read as '0'

bit 3 **CMIF**: Comparator Interrupt Flag bit

1 = Comparator 1 output has changed (must be cleared in software)
0 = Comparator 1 output has not changed

bit 2 **OSFIF**: Oscillator Fail Interrupt Flag bit

1 = System oscillator failed, clock input has changed to INTOSC (must be cleared in software)
0 = System clock operating

bit 1 **TMR2IF**: Timer 2 to PR2 Match Interrupt Flag bit

1 = Timer 2 to PR2 match occurred (must be cleared in software)
0 = Timer 2 to PR2 match has not occurred

bit 0 **TMR1IF**: Timer 1 Overflow Interrupt Flag bit

1 = Timer 1 register overflowed (must be cleared in software)
0 = Timer 1 has not overflowed

| **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 |

### 2.2.2.6 PCON Register

The Power Control (PCON) register contains flag bits (see Table 12-2) to differentiate between a:

- Power-on Reset ($\overline{\text{POR}}$)
- Brown-out Detect ($\overline{\text{BOD}}$)
- Watchdog Timer Reset (WDT)
- External $\overline{\text{MCLR}}$ Reset

The PCON register also controls the Ultra Low-Power Wake-up and software enable of the $\overline{\text{BOD}}$.

The PCON register bits are shown in Register 2-6.

**REGISTER 2-6: PCON – POWER CONTROL REGISTER (ADDRESS: 8Eh)**

| U-0 | U-0 | R/W-0 | R/W-1 | U-0 | U-0 | R/W-0 | R/W-x |
|---|---|---|---|---|---|---|---|
| — | — | ULPWUE | SBODEN | — | — | $\overline{\text{POR}}$ | $\overline{\text{BOD}}$ |
| bit 7 | | | | | | | bit 0 |

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5 **ULPWUE**: Ultra Low-Power Wake-up Enable bit
1 = Ultra Low-Power Wake-up enabled
0 = Ultra Low-Power Wake-up disabled

bit 4 **SBODEN**: Software BOD Enable bit[(1)]
1 = BOD enabled
0 = BOD disabled

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{\text{BOD}}$:** Brown-out Detect Status bit
1 = No Brown-out Detect occurred
0 = A Brown-out Detect occurred (must be set in software after a Brown-out Detect occurs)

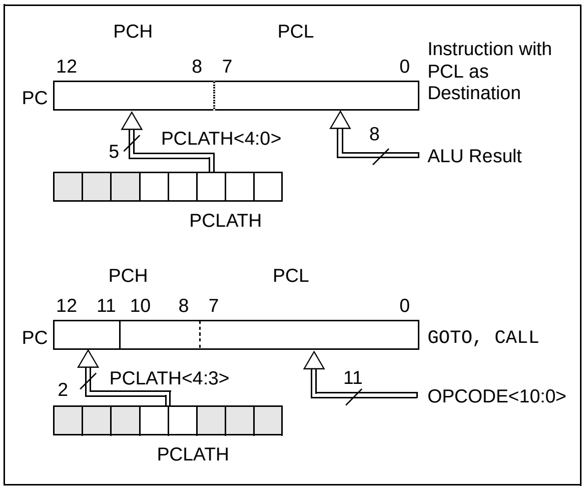
**Note 1:** BODEN<1:0> = 01 in the Configuration Word register for this bit to control the BOD.

| **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 |

## 2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-3 shows the two situations for the loading of the PC. The upper example in Figure 2-3 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 2-3 shows how the PC is loaded during a `CALL` or `GOTO` instruction (PCLATH<4:3> → PCH).

**FIGURE 2-3: LOADING OF PC IN DIFFERENT SITUATIONS**



### 2.3.1 COMPUTED `GOTO`

A computed `GOTO` is accomplished by adding an offset to the program counter (`ADDWF PCL`). When performing a table read using a computed `GOTO` method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to the Application Note *AN556, “Implementing a Table Read”* (DS00556).

### 2.3.2 STACK

The PIC12F683 family has an 8-level x 13-bit wide hardware stack (see Figure 2-1). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a `CALL` instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a `RETURN, RETLW` or a `RETFIE` instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

**Note 1:** There are no Status bits to indicate stack overflow or stack underflow conditions.

**2:** There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the `CALL, RETURN, RETLW` and `RETFIE` instructions or the vectoring to an interrupt address.

## 2.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

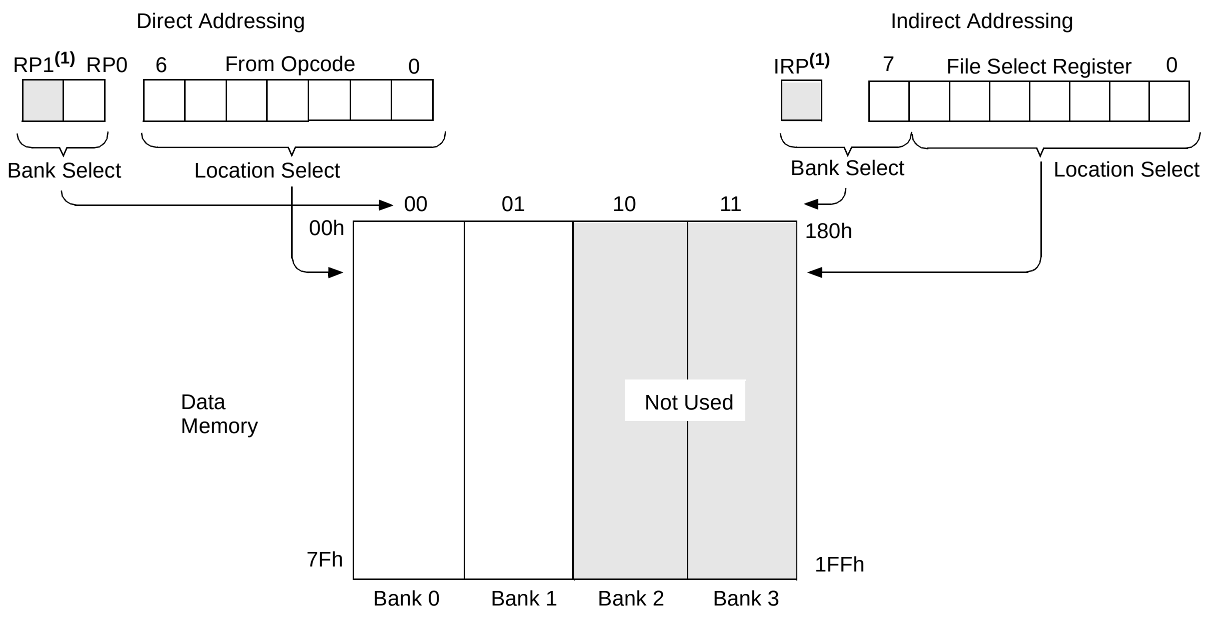
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (Status<7>), as shown in Figure 2-4.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 2-1.

**EXAMPLE 2-1: INDIRECT ADDRESSING**

```
        MOVLW   0x20    ;initialize pointer
        MOVWF   FSR     ;to RAM
NEXT    CLRF    INDF    ;clear INDF register
        INCF    FSR     ;inc pointer
        BTFSS   FSR,4   ;all done?
        GOTO    NEXT    ;no clear next
CONTINUE                ;yes continue
```

**FIGURE 2-4: DIRECT/INDIRECT ADDRESSING PIC12F683**



For memory map detail, see Figure 2-2.

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

# 3.0 CLOCK SOURCES

## 3.1 Overview

The PIC12F683 has a wide variety of clock sources and selection features to allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 3-1 illustrates a block diagram of the PIC12F683 clock sources.

Clock sources can be configured from external oscillators, quartz crystal resonators, ceramic resonators and Resistor-Capacitor (RC) circuits. In addition, the system clock source can be configured from one of two internal oscillators, with a choice of speeds selectable via software. Additional clock features include:

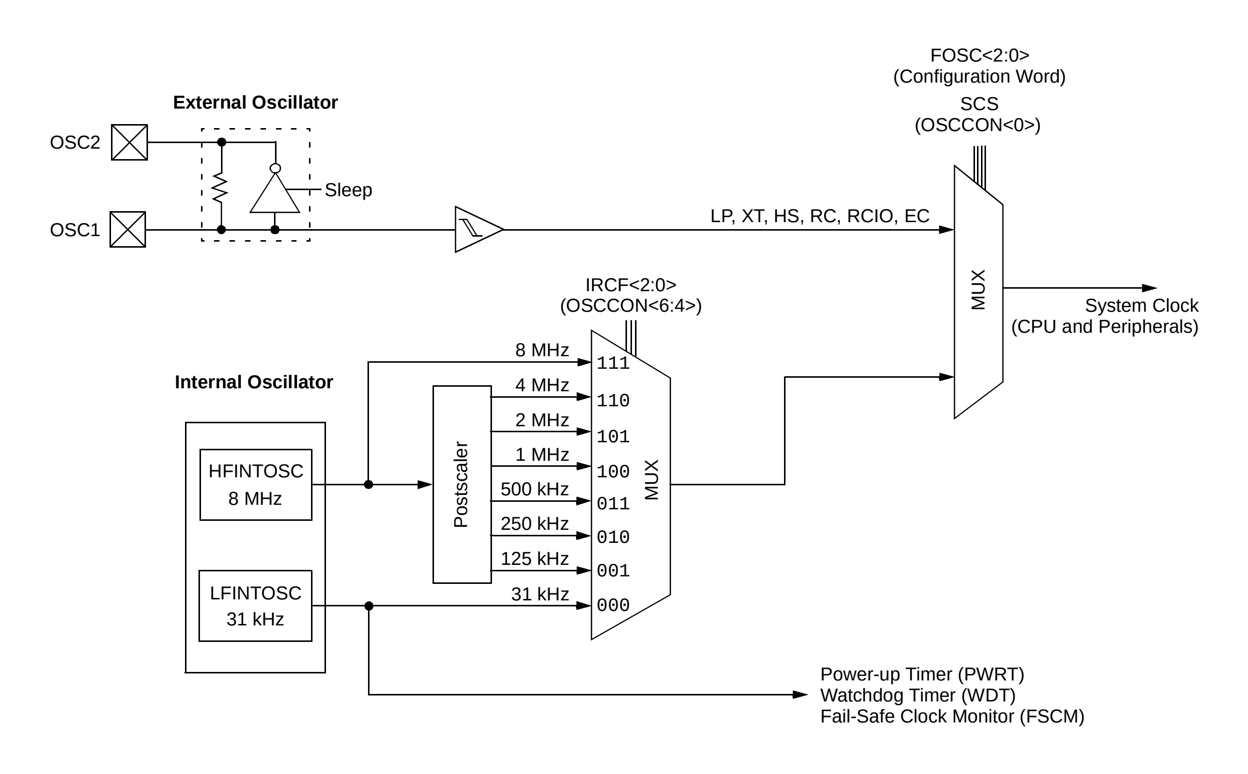
- Selectable system clock source between external or internal via software.
- Two-Speed Clock Start-up mode, which minimizes latency between external oscillator start-up and code execution.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, EC or RC modes) and switch to the internal oscillator.

The PIC12F683 can be configured in one of eight clock modes.

1. EC – External clock with I/O on GP4.
2. LP – Low gain crystal or Ceramic Resonator Oscillator mode.
3. XT – Medium gain crystal or Ceramic Resonator Oscillator mode.
4. HS – High gain crystal or Ceramic Resonator mode.
5. RC – External Resistor-Capacitor (RC) with FOSC/4 output on GP4
6. RCIO – External Resistor-Capacitor with I/O on GP4.
7. INTRC – Internal oscillator with FOSC/4 output on GP4 and I/O on GP5.
8. INTRCIO – Internal oscillator with I/O on GP4 and GP5.

Clock source modes are configured by the FOSC<2:0> bits in the Configuration Word register (see **Section 12.0 “Special Features of the CPU”**). The internal clock can be generated by two oscillators. The HFINTOSC is a high-frequency calibrated oscillator. The LFINTOSC is a low-frequency uncalibrated oscillator.

**FIGURE 3-1: PIC12F683 CLOCK SOURCE BLOCK DIAGRAM**



## 3.2 Clock Source Modes

Clock source modes can be classified as external or internal.

- External clock modes rely on external circuitry for the clock source. Examples are oscillator modules (EC mode), quartz crystal resonators or ceramic resonators (LP, XT and HS modes) and Resistor-Capacitor (RC mode) circuits.
- Internal clock sources are contained internally within the PIC12F683. The PIC12F683 has two internal oscillators: the 8 MHz High-Frequency Internal Oscillator (HFINTOSC) and 31 kHz Low-Frequency Internal Oscillator (LFINTOSC).

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS) bit (see **Section 3.5 "Clock Switching"**).

## 3.3 External Clock Modes

### 3.3.1 OSCILLATOR START-UP TIMER (OST)

If the PIC12F683 is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations from the OSC1 pin, following a Power-on Reset (POR) and the Power-up Timer (PWRT) has expired (if configured), or a wake-up from Sleep. During this time, the program counter does not increment and program execution is suspended. The OST ensures that the oscillator circuit, using a quartz crystal resonator or ceramic resonator, has started and is providing a stable system clock to the PIC12F683. When switching between clock sources a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 3-1.

In order to minimize latency between external oscillator start-up and code execution, the Two-Speed Clock Start-up mode can be selected (see **Section 3.6 "Two-Speed Clock Start-up Mode"**).

**TABLE 3-1: OSCILLATOR DELAY EXAMPLES**

| Switch From | Switch To | Frequency | Oscillator Delay |
|---|---|---|---|
| Sleep/POR | LFINTOSC<br>HFINTOSC | 31 kHz<br>125 kHz-8 MHz | 5 µs–10 µs (approx.) CPU Start-up[(1)] |
| Sleep/POR | EC, RC | DC – 20 MHz | |
| LFINTOSC (31 kHz) | EC, RC | DC – 20 MHz | |
| Sleep/POR | LP, XT, HS | 31 kHz-20 MHz | 1024 Clock Cycles (OST) |
| LFINTOSC (31 kHz) | HFINTOSC | 125 kHz-8 MHz | 1 µs (approx.) |

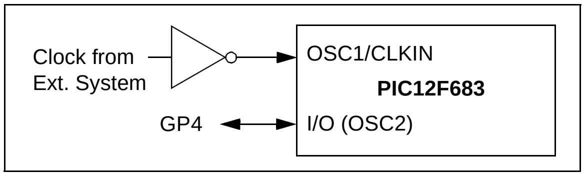
**Note 1:** The 5 µs–10 µs start-up delay is based on a 1 MHz system clock.

### 3.3.2 EC MODE

The External Clock (EC) mode allows an externally generated logic level as the system clock source. When operating in this mode, an external clock source is connected to the OSC1 pin and the GP5 pin is available for general purpose I/O. Figure 3-2 shows the pin connections for EC mode.

The Oscillator Start-up Timer (OST) is disabled when EC mode is selected. Therefore, there is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the PIC12F683 design is fully static, stopping the external clock input will have the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.

**FIGURE 3-2: EXTERNAL CLOCK (EC) MODE OPERATION**



### 3.3.3 LP, XT, HS MODES

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to the OSC1 and OSC2 pins (Figure 3-1). The mode selects a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

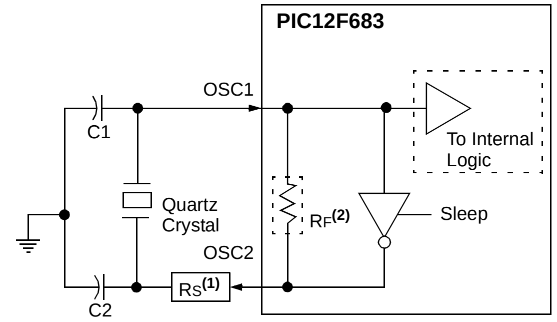
**LP** Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is best suited to drive resonators with a low drive level specification, for example, tuning fork type crystals.

**XT** Oscillator mode selects the intermediate gain setting of the internal inverter-amplifier. XT mode current consumption is the medium of the three modes. This mode is best suited to drive resonators with a medium drive level specification, for example, AT-cut quartz crystal resonators.

**HS** Oscillator mode selects the highest gain setting of the internal inverter-amplifier. HS mode current consumption is the highest of the three modes. This mode is best suited for resonators that require a high drive setting, for example, AT-cut quartz crystal resonators or ceramic resonators.

Figure 3-3 and Figure 3-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

**FIGURE 3-3: QUARTZ CRYSTAL OPERATION (LP, XT OR HS MODE)**



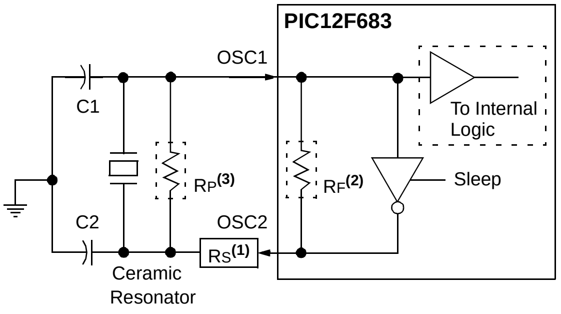
**Note 1:** A series resistor (RS) may be required for quartz crystals with low drive level.

**2:** The value of RF varies with the oscillator mode selected (typically between 2 MΩ to 10 MΩ).

**Note 1:** Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

**2:** Always verify oscillator performance over the VDD and temperature range that is expected for the application.

**FIGURE 3-4: CERAMIC RESONATOR OPERATION (XT OR HS MODE)**



**Note 1:** A series resistor (RS) may be required for ceramic resonators with low drive level.

**2:** The value of RF varies with the oscillator mode selected (typically between 2 MΩ to 10 MΩ).

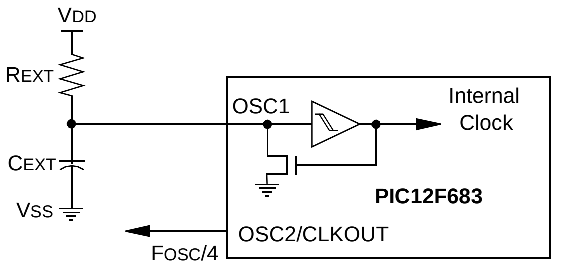
**3:** An additional parallel feedback resistor (RP) may be required for proper ceramic resonator operation (typical value 1 MΩ).

### 3.3.4 EXTERNAL RC MODES

The External Resistor-Capacitor (RC) modes support the use of an external RC circuit. This allows the designer maximum flexibility in frequency choice while keeping costs to a minimum when clock accuracy is not required. There are two modes, RC and RCIO.

In RC mode, the RC circuit connects to the OSC1 pin. The OSC2/CLKOUT pin outputs the RC oscillator frequency divided by 4. This signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements. Figure 3-5 shows the RC mode connections.

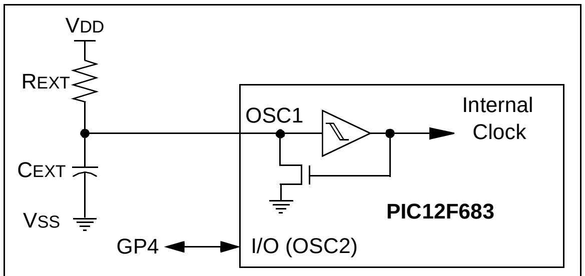
**FIGURE 3-5: RC MODE**



Recommended values: 3 kΩ ≤ REXT ≤ 100 kΩ
CEXT > 20 pF

In RCIO mode, the RC circuit is connected to the OSC1 pin. The OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 4 of GPIO (GP4). Figure 3-6 shows the RCIO mode connections.

**FIGURE 3-6: RCIO MODE**



Recommended values: 3 kΩ ≤ REXT ≤ 100 kΩ
CEXT > 20 pF

The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values and the operating temperature. Other factors affecting the oscillator frequency are:

- threshold voltage variation
- component tolerances
- packaging variations in capacitances

## 3.4 Internal Clock Modes

The PIC12F683 has two independent, internal oscillators that can be configured or selected as the system clock source.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 8 MHz. The frequency of the HFINTOSC can be user adjusted ±12% via software using the OSCTUNE register (Register 3-1).
2. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at approximately 31 kHz.

The system clock speed can be selected via software using the Internal Oscillator Frequency Select (IRCF) bits.

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS) bit (see **Section 3.5 “Clock Switching”**).

### 3.4.1 INTRC AND INTRCIO MODES

The INTRC and INTRCIO modes configure the internal oscillators as the system clock source when the device is programmed using the Oscillator Selection (FOSC) bits in the Configuration Word register (Register 12-1).

In **INTRC** mode, the OSC1 pin is available for general purpose I/O. The OSC2/CLKOUT pin outputs the selected internal oscillator frequency divided by 4. The CLKOUT signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements.

In **INTRCIO** mode, the OSC1 and OSC2 pins are available for general purpose I/O.

### 3.4.2 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 8 MHz internal clock source. The frequency of the HFINTOSC can be altered approximately ±12% via software using the OSCTUNE register (Register 3-1).

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). One of seven frequencies can be selected via software using the IRCF bits (see **Section 3.4.4 “Frequency Select Bits (IRCF)”**).

The HFINTOSC is enabled by selecting any frequency between 8 MHz and 125 kHz (IRCF ≠ 000) as the system clock source (SCS = 1), or when Two-Speed Start-up is enabled (IESO = 1 and IRCF ≠ 000).

The HF Internal Oscillator (HTS) bit (OSCCON<2>) indicates whether the HFINTOSC is stable or not.

### 3.4.2.1 OSCTUNE Register

The HFINTOSC is factory calibrated but can be adjusted in software by writing to the OSCTUNE register (Register 3-1).

The OSCTUNE register has a tuning range of ±12%. The default value of the OSCTUNE register is '0'. The value is a 5-bit two's complement number. Due to process variation, the monotonicity and frequency step cannot be specified.

When the OSCTUNE register is modified, the HFINTOSC frequency will begin shifting to the new frequency. The HFINTOSC clock will stabilize within 1 ms. Code execution continues during this shift. There is no indication that the shift has occurred.

OSCTUNE does not affect the LFINTOSC frequency. Operation of features that depend on the LFINTOSC clock source frequency, such as the Power-up Timer (PWRT), Watchdog Timer (WDT), Fail-Safe Clock Monitor (FSCM) and peripherals, are *not* affected by the change in frequency.

**REGISTER 3-1: OSCTUNE – OSCILLATOR TUNING RESISTOR (ADDRESS: 90h)**

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 |
| bit 7 | | | | | | | bit 0 |

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **TUN<4:0>:** Frequency Tuning bits

01111 = Maximum frequency
01110 =
•
•
•
00001 =
00000 = Oscillator module is running at the calibrated frequency.
11111 =
•
•
•
10000 = Minimum frequency

| **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 |

### 3.4.3 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is an uncalibrated (approximate) 31 kHz internal clock source.

The output of the LFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). 31 kHz can be selected via software using the IRCF bits (see **Section 3.4.4 "Frequency Select Bits (IRCF)"**). The LFINTOSC is also the frequency for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

The LFINTOSC is enabled by selecting 31 kHz (IRCF = `000`) as the system clock source (SCS = `1`), or when any of the following are enabled:

- Two-Speed Start-up (IESO = `1` and IRCF = `000`)
- Power-up Timer (PWRT)
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor (FSCM)

The LF Internal Oscillator (LTS) bit (OSCCON<1>) indicates whether the LFINTOSC is stable or not.

### 3.4.4 FREQUENCY SELECT BITS (IRCF)

The output of the 8 MHz HFINTOSC and 31 kHz LFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). The Internal Oscillator Frequency select bits, IRCF<2:0> (OSCCON<6:4>), select the frequency output of the internal oscillators. One of eight frequencies can be selected via software:

- 8 MHz
- 4 MHz (Default after Reset)
- 2 MHz
- 1 MHz
- 500 kHz
- 250 kHz
- 125 kHz
- 31 kHz

| **Note:** | Following any Reset, the IRCF bits are set to '`110`' and the frequency selection is set to 4 MHz. The user can modify the IRCF bits to select a different frequency. |
|---|---|

### 3.4.5 HF AND LF INTOSC CLOCK SWITCH TIMING

When switching between the LFINTOSC and the HFINTOSC, the new oscillator may already be shut down to save power. If this is the case, there is a 10 μs delay after the IRCF bits are modified before the frequency selection takes place. The LTS/HTS bits will reflect the current active status of the LFINTOSC and the HFINTOSC oscillators. The timing of a frequency selection is as follows:

1. IRCF bits are modified.
2. If the new clock is shut down, a 10 μs clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. CLKOUT is held low and the clock switch circuitry waits for a rising edge in the new clock.
5. CLKOUT is now connected with the new clock. HTS/LTS bits are updated as required.
6. Clock switch is complete.

If the internal oscillator speed selected is between 8 MHz and 125 kHz, there is no start-up delay before the new frequency is selected. This is because the old and the new frequencies are derived from the HFINTOSC via the postscaler and multiplexer.

## 3.5 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS) bit.

### 3.5.1 SYSTEM CLOCK SELECT (SCS) BIT

The System Clock Select (SCS) bit (OSCCON<0>) selects the system clock source that is used for the CPU and peripherals.

- When SCS = `0`, the system clock source is determined by configuration of the FOSC<2:0> bits in the Configuration Word register (CONFIG).
- When SCS = `1`, the system clock source is chosen by the internal oscillator frequency selected by the IRCF bits. After a Reset, SCS is always cleared.

| **Note:** | Any automatic clock switch, which may occur from Two-Speed Start-up or Fail-Safe Clock Monitor, does not update the SCS bit. The user can monitor the OSTS (OSCCON<3>) to determine the current system clock source. |
|---|---|

### 3.5.2 OSCILLATOR START-UP TIME-OUT STATUS BIT

The Oscillator Start-up Time-out Status (OSTS) bit (OSCCON<3>) indicates whether the system clock is running from the external clock source, as defined by the FOSC bits, or from internal clock source. In particular, OSTS indicates that the Oscillator Start-up Timer (OST) has timed out for LP, XT or HS modes.

## 3.6 Two-Speed Clock Start-up Mode

Two-Speed Start-up mode provides additional power savings by minimizing the latency between external oscillator start-up and code execution. In applications that make heavy use of the Sleep mode, Two-Speed Start-up will remove the external oscillator start-up time from the time spent awake and can reduce the overall power consumption of the device.

This mode allows the application to wake-up from Sleep, perform a few instructions using the INTOSC as the clock source and go back to Sleep without waiting for the primary oscillator to become stable.

| **Note:** | Executing a `SLEEP` instruction will abort the oscillator start-up time and will cause the OSTS bit (OSCCON<3>) to remain clear. |
|---|---|

When the PIC12F683 is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) is enabled (see **Section 3.3.1 "Oscillator Start-up Timer (OST)"**). The OST timer will suspend program execution until 1024 oscillations are counted. Two-Speed Start-up mode minimizes the delay in code execution by operating from the internal oscillator as the OST is counting. When the OST count reaches 1024 and the OSTS bit (OSCCON<3>) is set, program execution switches to the external oscillator.

### 3.6.1 TWO-SPEED START-UP MODE CONFIGURATION

Two-Speed Start-up mode is configured by the following settings:

- IESO = `1` (CONFIG<10>) Internal/External Switch Over bit.
- SCS = `0`.
- FOSC configured for LP, XT or HS mode.

Two-Speed Start-up mode is entered after:

- Power-on Reset (POR) and, if enabled, after PWRT has expired, or
- Wake-up from Sleep.

If the external clock oscillator is configured to be anything other than LP, XT or HS mode, then Two-Speed Start-up is disabled. This is because the external clock oscillator does not require any stabilization time after POR or an exit from Sleep.

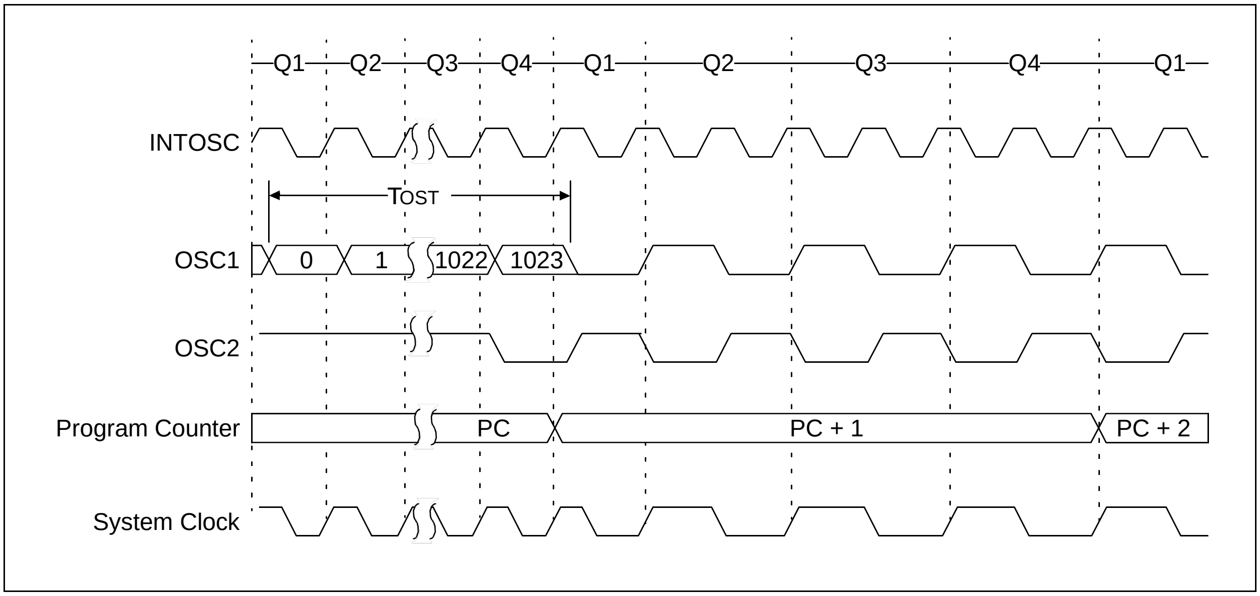
### 3.6.2 TWO-SPEED START-UP SEQUENCE

1. Wake-up from Power-on Reset or Sleep.
2. Instructions begin execution by the internal oscillator at the frequency set in the IRCF bits (OSCCON<6:4>).
3. OST enabled to count 1024 clock cycles.
4. OST timed out, wait for falling edge of the internal oscillator.
5. OSTS is set.
6. System clock held low until the next falling edge of new clock (LP, XT or HS mode).
7. System clock is switched to external clock source.

### 3.6.3 CHECKING EXTERNAL/INTERNAL CLOCK STATUS

Checking the state of the OSTS bit (OSCCON<3>) will confirm if the PIC12F683 is running from the external clock source as defined by the FOSC bits in the Configuration Word register (CONFIG) or the internal oscillator.

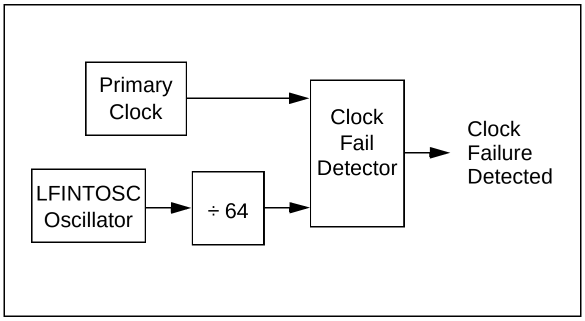
**FIGURE 3-7: TWO-SPEED START-UP**



## 3.7 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) is designed to allow the device to continue to operate in the event of an oscillator failure. The FSCM can detect oscillator failure at any point after the device has exited a Reset or Sleep condition and the Oscillator Start-up Timer (OST) has expired.

**FIGURE 3-8: FSCM BLOCK DIAGRAM**



The FSCM function is enabled by setting the FCMEN bit in the Configuration Word register (CONFIG). It is applicable to all external clock options (LP, XT, HS, EC, RC or IO modes).

In the event of an external clock failure, the FSCM will set the OSFIF bit (PIR1<2>) and generate an oscillator fail interrupt if the OSFIE bit (PIE1<2>) is set. The device will then switch the system clock to the internal oscillator. The system clock will continue to come from the internal oscillator unless the external clock recovers and the Fail-Safe condition is exited.

The frequency of the internal oscillator will depend upon the value contained in the IRCF bits (OSCCON<6:4>). Upon entering the Fail-Safe condition, the OSTS bit (OSCCON<3>) is automatically cleared to reflect that the internal oscillator is active and the WDT is cleared. The SCS bit (OSCCON<0>) is not updated. Enabling FSCM does not affect the LTS bit.

The FSCM sample clock is generated by dividing the INTRC clock by 64. This will allow enough time between FSCM sample clocks for a system clock edge to occur. Figure 3-8 shows the FSCM block diagram.

On the rising edge of the sample clock, the monitoring latch (CM = 0) will be cleared. On a falling edge of the primary system clock, the monitoring latch will be set (CM = 1). In the event that a falling edge of the sample clock occurs and the monitoring latch is not set, a clock failure has been detected. The assigned internal oscillator is enabled when FSCM is enabled, as reflected by the IRCF.

| **Note:** | Two-Speed Start-up is automatically enabled when the Fail-Safe Clock Monitor mode is enabled. |
|---|---|

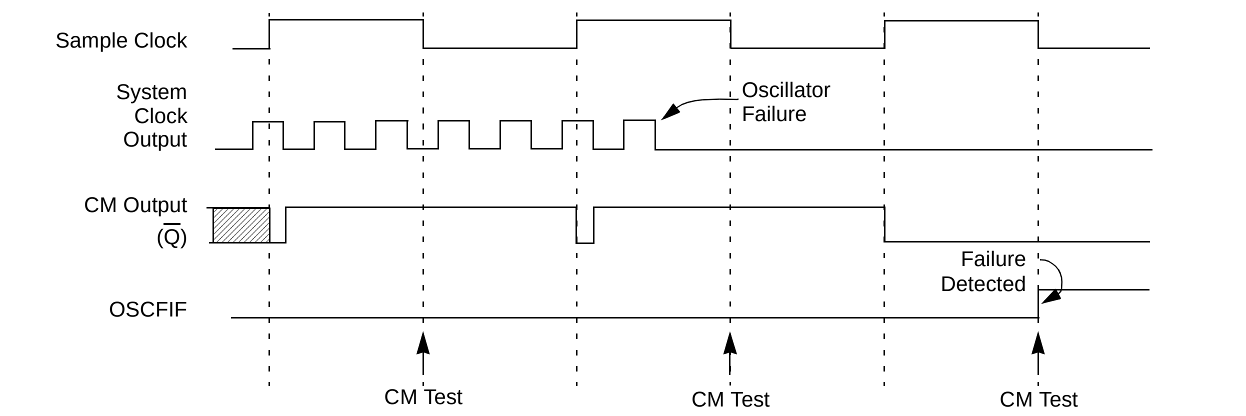
| **Note:** | Primary clocks with a frequency ≤ ~488 Hz will be considered failed by the FSCM. A slow starting oscillator can cause an FSCM interrupt. |
|---|---|

### 3.7.1 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared after a Reset, the execution of a SLEEP instruction, or a modification of the SCS bit. While in Fail-Safe condition, the PIC12F683 uses the internal oscillator as the system clock source. The IRCF bits (OSCCON<6:4>) can be modified to adjust the internal oscillator frequency without exiting the Fail-Safe condition.

The Fail-Safe condition must be cleared before the OSFIF flag can be cleared.

**FIGURE 3-9: FSCM TIMING DIAGRAM**



**Note:** The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

### 3.7.2 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect oscillator failure at any point after the device has exited a Reset or Sleep condition and the Oscillator Start-up Timer (OST) has expired. If the external clock is EC or RC mode, monitoring will begin immediately following these events.

For LP, XT or HS mode the external oscillator may require a start-up time considerably longer than the FSCM sample clock time or a false clock failure may be detected (see Figure 3-9). To prevent this, the internal oscillator is automatically configured as the system clock and functions until the external clock is stable (the OST has timed out). This is identical to Two-Speed Start-up mode. Once the external oscillator is stable, the LFINTOSC returns to its role as the FSCM source.

| | |
|---|---|
| **Note:** | Due to the wide range of oscillator start-up times, the Fail-Safe circuit is not active during oscillator start-up (i.e., after exiting Reset or Sleep). After an appropriate amount of time, the user should check the OSTS bit (OSCCON<3>) to verify the oscillator start-up and system clock switchover has successfully completed. |

**REGISTER 3-2: OSCCON – OSCILLATOR CONTROL REGISTER (ADDRESS: 8Fh)**

| U-0 | R/W-1 | R/W-1 | R/W-0 | R-1 | R-0 | R-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | IRCF2 | IRCF1 | IRCF0 | OSTS[1] | HTS | LTS | SCS |
| bit 7 | | | | | | | bit 0 |

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **IRCF<2:0>:** Internal Oscillator Frequency Select bits
000 = 31 kHz
001 = 125 kHz
010 = 250 kHz
011 = 500 kHz
100 = 1 MHz
101 = 2 MHz
110 = 4 MHz
111 = 8 MHz

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit
1 = Device is running from the external system clock defined by FOSC<2:0>
0 = Device is running from the internal system clock (HFINTOSC or LFINTOSC)

bit 2 **HTS:** HFINTOSC (High Frequency – 8 MHz to 125 kHz) Status bit
1 = HFINTOSC is stable
0 = HFINTOSC is not stable

bit 1 **LTS:** LFINTOSC (Low Frequency – 31 kHz) Stable bit
1 = LFINTOSC is stable
0 = LFINTOSC is not stable

bit 0 **SCS:** System Clock Select bit
1 = Internal oscillator is used for system clock
0 = Clock source defined by FOSC<2:0>

**Note 1:** Bit resets to ‘0’ with Two-Speed Start-up and LP, XT or HS selected as the oscillator mode or Fail-Safe mode is enabled.

| **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 |

### TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | EEIF | ADIF | CCP1IF | — | CMIF | OSFIF | TMR2IF | TMR1IF | `000- 0000` | `0000 0000` |
| 8Ch | PIE1 | EEIE | ADIE | CCP1IE | — | CMIE | OSFIE | TMR2IE | TMR1IE | `000- 0000` | `0000 0000` |
| 8Fh | OSCCON | — | IRCF2 | IRCF1 | IRCF0 | OSTS(2) | HTS | LTS | SCS | `-110 x000` | `-110 x000` |
| 90h | OSCTUNE | — | — | — | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 | `---0 0000` | `---u uuuu` |
| 2007h(1) | CONFIG | $\overline{\text{CPD}}$ | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | FOSC2 | FOSC1 | FOSC0 | — | — |

**Legend:** x = unknown, u = unchanged, — = unimplemented locations read as '0'. Shaded cells are not used by oscillators.

**Note 1:** See Register 12-1 for operation of all Configuration Word register bits.

**2:** See Register 3-2 for details.

NOTES:

## 4.0 GPIO PORT

There are as many as six general purpose I/O pins available. Depending on which peripherals are enabled, some or all of the pins may not be available as general purpose I/O. In general, when a peripheral is enabled, the associated pin may not be used as a general purpose I/O pin.

**Note:** Additional information on I/O ports may be found in the "*PICmicro® Mid-Range MCU Family Reference Manual"* (DS33023).

### 4.1 GPIO and the TRISIO Registers

GPIO is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISIO. Setting a TRISIO bit (= 1) will make the corresponding GPIO pin an input (i.e., put the corresponding output driver in a High-impedance mode). Clearing a TRISIO bit (= 0) will make the corresponding GPIO pin an output (i.e., put the contents of the output latch on the selected pin). The exception is GP3, which is input only and its TRISIO bit will always read as '1'. Example 4-1 shows how to initialize GPIO.

Reading the GPIO register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the port data latch. GP3 reads '0' when MCLRE = 1.

The TRISIO register controls the direction of the GPIO pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISIO register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

**Note:** The ANSEL (9Fh) and CMCON0 (19h) registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

**EXAMPLE 4-1: INITIALIZING GPIO**

```
BCF    STATUS,RP0     ;Bank 0
CLRF   GPIO           ;Init GPIO
MOVLW  07h            ;Set GP<2:0> to
MOVWF  CMCON0         ;digital I/O
BSF    STATUS,RP0     ;Bank 1
CLRF   ANSEL          ;digital I/O
MOVLW  0Ch            ;Set GP<3:2> as inputs
MOVWF  TRISIO         ;and set GP<5:4,1:0>
                      ;as outputs
BCF    STATUS,RP0     ;Bank 0
```

### 4.2 Additional Pin Functions

Every GPIO pin on the PIC12F683 has an interrupt-on-change option and a weak pull-up option. GP0 has an Ultra Low-Power Wake-up option. The next three sections describe these functions.

#### 4.2.1 WEAK PULL-UPS

Each of the GPIO pins, except GP3, has an individually configurable weak internal pull-up. Control bits WPUx enable or disable each pull-up. Refer to Register 4-3. Each weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset by the $\overline{\text{GPPU}}$ bit (OPTION<7>). A weak pull-up is automatically enabled for GP3 when configured as $\overline{\text{MCLR}}$ and disabled when GP3 is an I/O. There is no software control of the $\overline{\text{MCLR}}$ pull-up.

**REGISTER 4-1: GPIO – GENERAL PURPOSE I/O REGISTER (ADDRESS: 05h)**

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

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

bit 5-0: **GPIO<5:0>**: GPIO I/O pin
1 = Port pin is > $V_{IH}$
0 = Port pin is < $V_{IL}$

**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 |

### REGISTER 4-2: TRISIO – GPIO TRI-STATE REGISTER (ADDRESS: 85h)

| U-0 | U-0 | R/W-1 | R/W-1 | R-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | TRISIO5 | TRISIO4 | TRISIO3 | TRISIO2 | TRISIO1 | TRISIO0 |
| bit 7 | | | | | | | bit 0 |

bit 7-6: **Unimplemented**: Read as ‘0’

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

**Note 1:** TRISIO<3> always reads ‘1’.

**2:** TRISIO<5:4> reads ‘1’ in XT, LP and HS modes.

| **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 |

### REGISTER 4-3: WPU – WEAK PULL-UP REGISTER (ADDRESS: 95h)

| U-0 | U-0 | R/W-1 | R/W-1 | U-0 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | WPU5 | WPU4 | — | WPU2 | WPU1 | WPU0 |
| bit 7 | | | | | | | bit 0 |

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **WPU<5:4>:** Weak Pull-up register bit
1 = Pull-up enabled
0 = Pull-up disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **WPU<2:0>:** Weak Pull-up register bit
1 = Pull-up enabled
0 = Pull-up disabled

**Note 1:** Global $\overline{\text{GPPU}}$ must be enabled for individual pull-ups to be enabled.

**2:** The weak pull-up device is automatically disabled if the pin is in output mode (TRISIO = 0).

**3:** The GP3 pull-up is enabled when configured as $\overline{\text{MCLR}}$ and disabled as an I/O in the Configuration Word.

**4:** WPU<5:4> reads ‘1’ in XT, LP and HS modes.

| **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 |

### 4.2.2 INTERRUPT-ON-CHANGE

Each of the GPIO pins is individually configurable as an interrupt-on-change pin. Control bits IOCx enable or disable the interrupt function for each pin. Refer to Register 4-4. The interrupt-on-change is disabled on a Power-on Reset.

For enabled interrupt-on-change pins, the values are compared with the old value latched on the last read of GPIO. The 'mismatch' outputs of the last read are OR'd together to set the GPIO Change Interrupt Flag bit (GPIF) in the INTCON register.

This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, clears the interrupt by:

a) Any read or write of GPIO. This will end the mismatch condition, then

b) Clear the flag bit GPIF.

A mismatch condition will continue to set flag bit GPIF. Reading GPIO will end the mismatch condition and allow flag bit GPIF to be cleared. The latch holding the last read value is not affected by a $\overline{\text{MCLR}}$ nor BOD Reset. After these resets, the GPIF flag will continue to be set if a mismatch is present.

| **Note:** | If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the GPIF interrupt flag may not get set. |
|---|---|

**REGISTER 4-4: IOC – INTERRUPT-ON-CHANGE GPIO REGISTER (ADDRESS: 96h)**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | IOC5 | IOC4 | IOC3 | IOC2 | IOC1 | IOC0 |
| bit 7 | | | | | | | bit 0 |

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

bit 5-0 **IOC<5:0>:** Interrupt-on-change GPIO Control bit
1 = Interrupt-on-change enabled
0 = Interrupt-on-change disabled

**Note 1:** Global Interrupt Enable (GIE) must be enabled for individual interrupts to be recognized.

**2:** IOC<5:4> reads '1' in XT, LP and HS modes.

| **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 |

### 4.2.3 ULTRA LOW-POWER WAKE-UP

The Ultra Low-Power Wake-up (ULPWU) on GP0 allows a slow falling voltage to generate an interrupt-on-change on GP0 without excess current consumption. The mode is selected by setting the ULPWUE bit (PCON<5>). This enables a small current sink which can be used to discharge a capacitor on GP0.

To use this feature, the GP0 pin is configured to output '1' to charge the capacitor, interrupt-on-change for GP0 is enabled and GP0 is configured as an input. The ULPWUE bit is set to begin the discharge and a SLEEP instruction is performed. When the voltage on GP0 drops below $V_{IL}$, an interrupt will be generated which will cause the device to wake-up. Depending on the state of the GIE bit (INTCON<7>), the device will either jump to the interrupt vector (0004h) or execute the next instruction when the interrupt event occurs. See **Section 4.2.2 "Interrupt-on-change"** and **Section 12.4.3 "GPIO Interrupt"** for more information.

This feature provides a low-power technique for periodically waking up the device from Sleep. The time-out is dependent on the discharge time of the RC circuit on GP0. See Example 4-2 for initializing the Ultra Low-Power Wake-up module.

The series resistor provides overcurrent protection for the GP0 pin and can allow for software calibration of the time-out (see Figure 4-1). A timer can be used to measure the charge time and discharge time of the capacitor. The charge time can then be adjusted to provide the desired interrupt delay. This technique will compensate for the affects of temperature, voltage and component accuracy. The Ultra Low-Power Wake-up peripheral can also be configured as a simple Programmable Low-Voltage Detect or temperature sensor.

| **Note:** | For more information, refer to the Application Note *AN879, "Using the Microchip Ultra Low-Power Wake-up Module"* (DS00879). |
|---|---|

**EXAMPLE 4-2: ULTRA LOW-POWER WAKE-UP INITIALIZATION**

```
BCF     STATUS,RP0      ;Bank 0
BSF     GPIO,0          ;Set GP0 data latch
MOVLW   H’7’            ;Turn off
MOVWF   CMCON0          ; comparator
BSF     STATUS,RP0      ;Bank 1
BCF     ANSEL,0         ;GP0 to digital I/O
BCF     TRISIO,0        ;Output high to
CALL    CapDelay        ; charge capacitor
BSF     PCON,ULPWUE     ;Enable ULP Wake-up
BSF     IOC,0           ;Select GP0 IOC
BSF     TRISIO,0        ;GP0 to input
MOVLW   B’10001000’     ;Enable interrupt
MOVWF   INTCON          ; and clear flag
SLEEP                   ;Wait for IOC
```

### 4.2.4 PIN DESCRIPTIONS AND DIAGRAMS

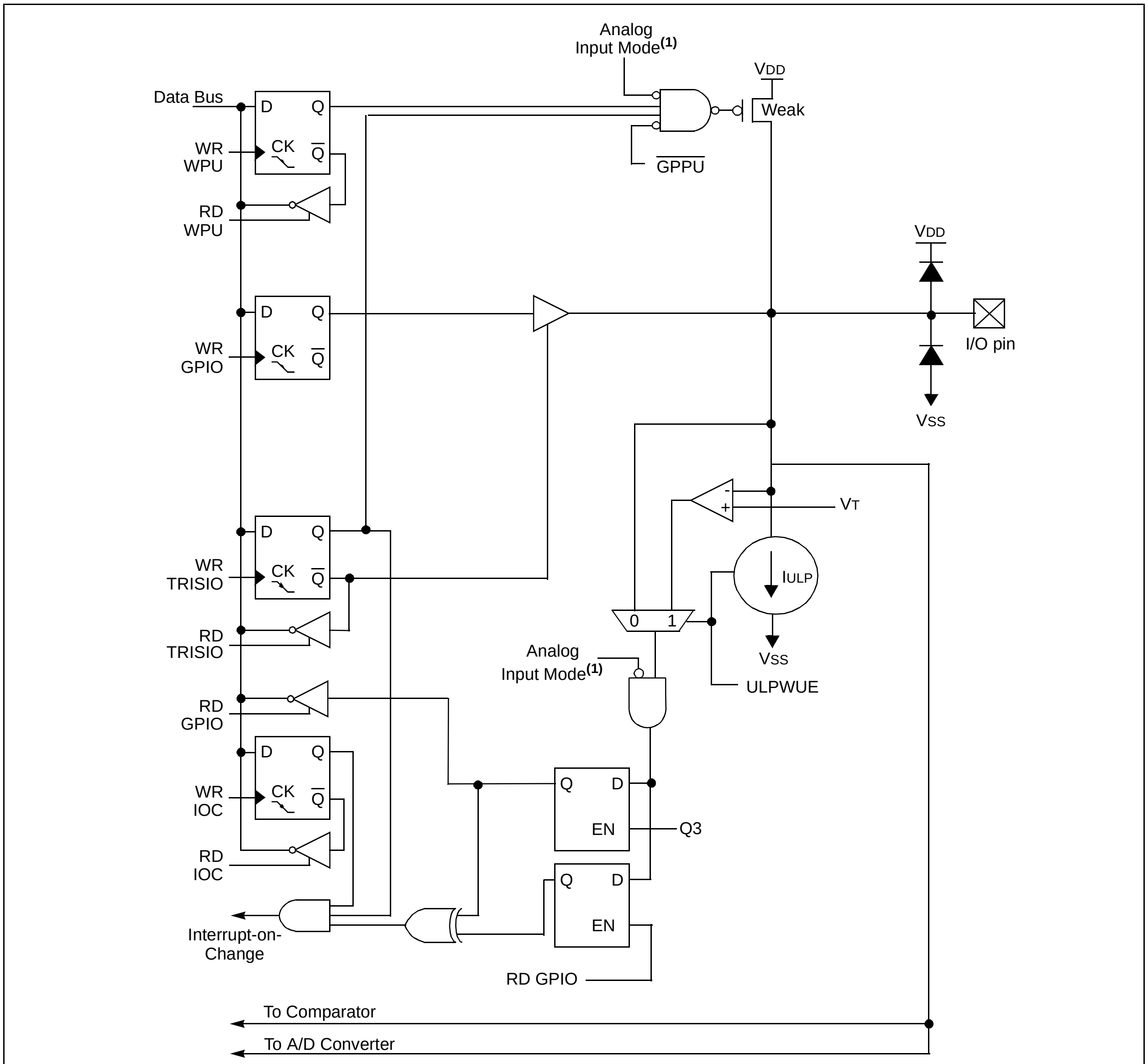
Each GPIO pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the comparator or the A/D, refer to the appropriate section in this data sheet.

#### 4.2.4.1 GP0/AN0/CIN+/ICSPDAT/ULPWU

Figure 4-1 shows the diagram for this pin. The GP0 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- an analog input to the comparator
- an analog input to the Ultra Low-Power Wake-up
- In-Circuit Serial Programming data

**FIGURE 4-1: BLOCK DIAGRAM OF GP0**



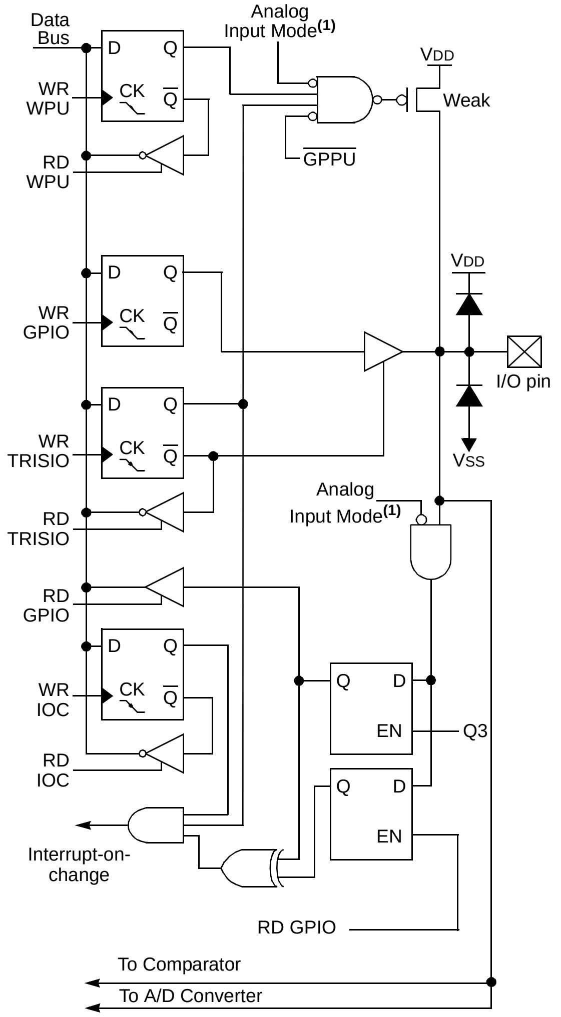
**Note 1:** Comparator mode and ANSEL determines Analog Input mode.

### 4.2.4.2 GP1/AN1/CIN-/VREF/ICSPCLK

Figure 4-1 shows the diagram for this pin. The GP1 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- a analog input to the comparator
- a voltage reference input for the A/D
- In-Circuit Serial Programming clock

**FIGURE 4-2: BLOCK DIAGRAM OF GP1**



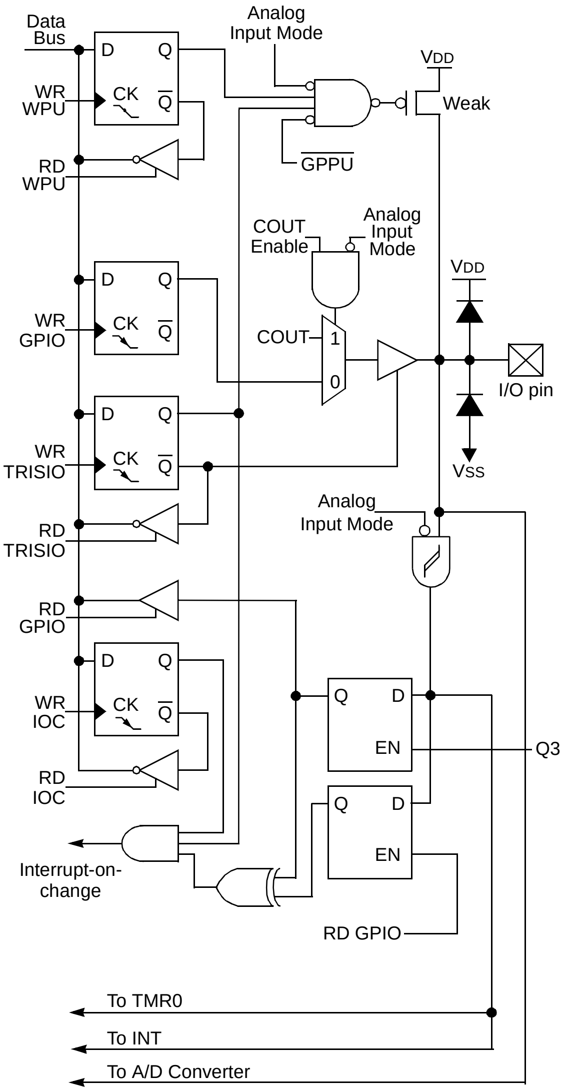
**Note 1:** Comparator mode and ANSEL determines Analog Input mode.

### 4.2.4.3 GP2/AN2/T0CKI/INT/COUT/CCP1

Figure 4-3 shows the diagram for this pin. The GP2 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- the clock input for TMR0
- an external edge triggered interrupt
- a digital output from the comparator
- a digital input/output for the CCP (refer to **Section 11.0 “Capture/Compare/PWM (CCP) Module”**).

**FIGURE 4-3: BLOCK DIAGRAM OF GP2**



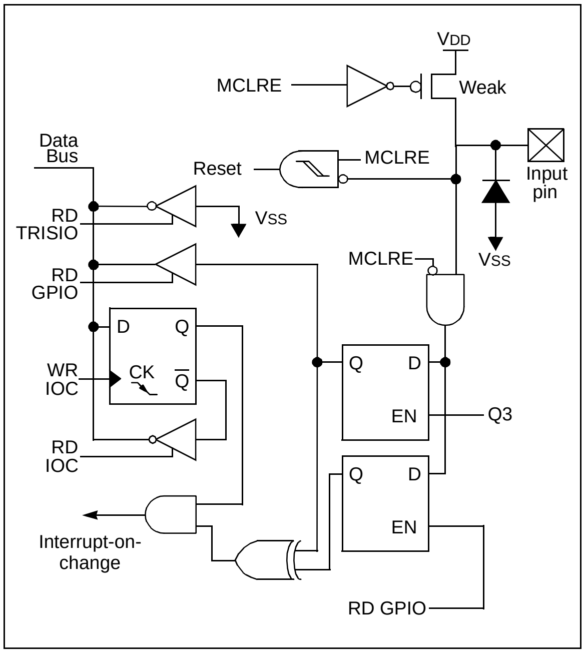
**Note 1:** Comparator mode and ANSEL determines Analog Input mode.

#### 4.2.4.4 GP3/$\overline{\text{MCLR}}$/VPP

Figure 4-4 shows the diagram for this pin. The GP3 pin is configurable to function as one of the following:

- a general purpose input
- as Master Clear Reset with weak pull-up

**FIGURE 4-4: BLOCK DIAGRAM OF GP3**

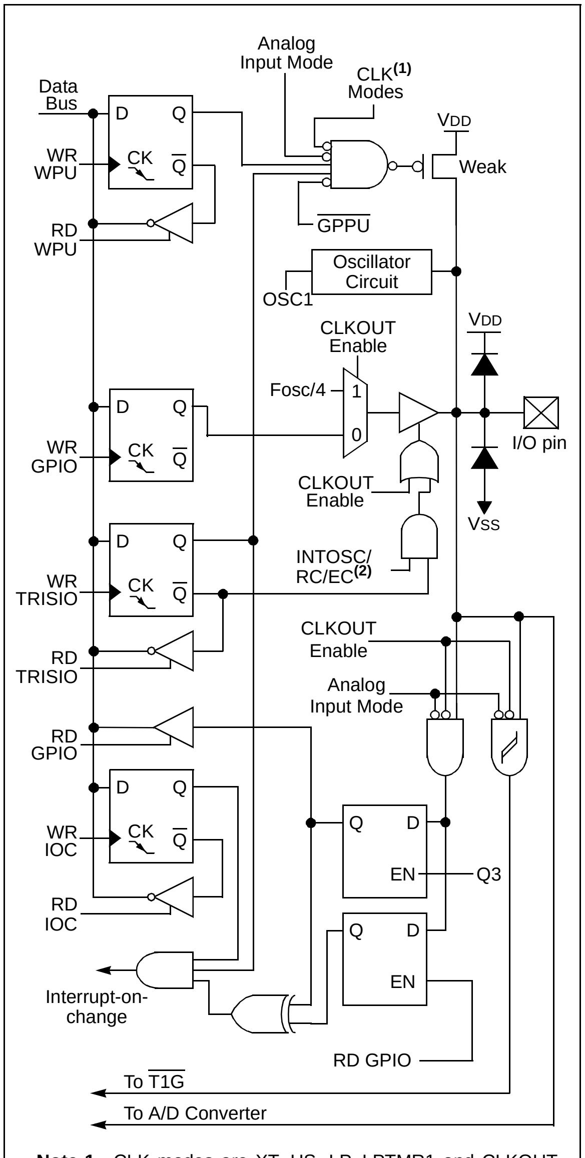


#### 4.2.4.5 GP4/AN3/$\overline{\text{T1G}}$/OSC2/CLKOUT

Figure 4-5 shows the diagram for this pin. The GP4 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the A/D
- a TMR1 gate input
- a crystal/resonator connection
- a clock output

**FIGURE 4-5: BLOCK DIAGRAM OF GP4**



**Note 1:** CLK modes are XT, HS, LP, LPTMR1 and CLKOUT Enable.

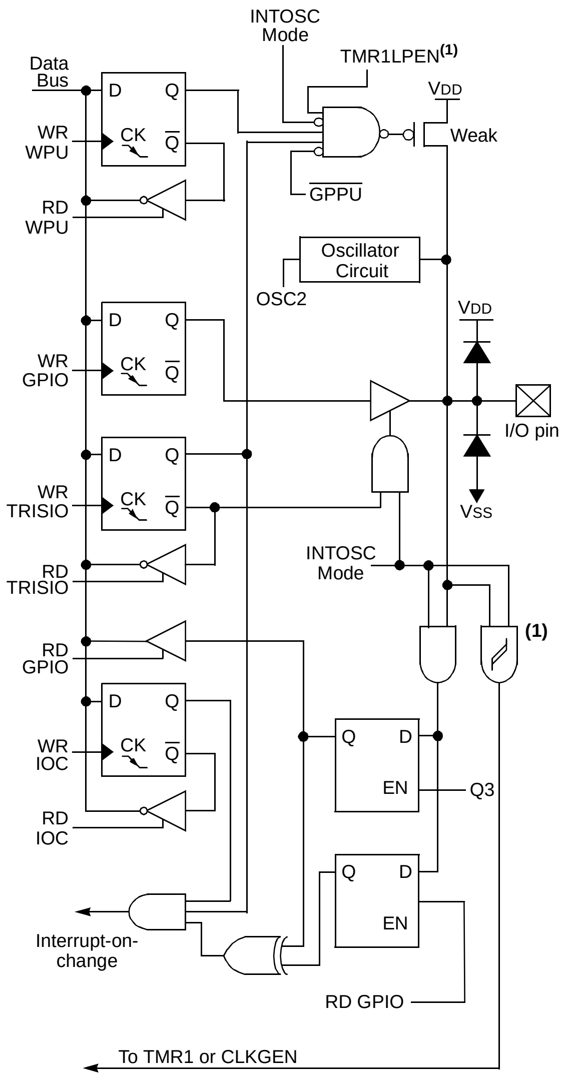
**2:** With CLKOUT option.

#### 4.2.4.6 GP5/T1CKI/OSC1/CLKIN

Figure 4-6 shows the diagram for this pin. The GP5 pin is configurable to function as one of the following:

- a general purpose I/O
- a TMR1 clock input
- a crystal/resonator connection
- a clock input

**FIGURE 4-6: BLOCK DIAGRAM OF GP5**



**Note 1:** Timer1 LP oscillator enabled.

**2:** When using Timer1 with LP oscillator, the Schmitt Trigger is bypassed.

**TABLE 4-1: SUMMARY OF REGISTERS ASSOCIATED WITH GPIO**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 05h | GPIO | — | — | GP5 | GP4 | GP3 | GP2 | GP1 | GP0 | `--xx xx00` | `--uu uu00` |
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | GPIE | T0IF | INTF | GPIF | `0000 0000` | `0000 0000` |
| 19h | CMCON0 | — | COUT | — | CINV | CIS | CM2 | CM1 | CM0 | `-0-0 0000` | `-0-0 0000` |
| 81h | OPTION_REG | $\overline{\text{GPPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | `1111 1111` | `1111 1111` |
| 85h | TRISIO | — | — | TRISIO5 | TRISIO4 | TRISIO3 | TRISIO2 | TRISIO1 | TRISIO0 | `--11 1111` | `--11 1111` |
| 95h | WPU | — | — | WPU5 | WPU4 | — | WPU2 | WPU1 | WPU0 | `--11 -111` | `--11 -111` |
| 96h | IOC | — | — | IOC5 | IOC4 | IOC3 | IOC2 | IOC1 | IOC0 | `--00 0000` | `--00 0000` |
| 9Fh | ANSEL | — | ADCS2 | ADCS1 | ADCS0 | ANS3 | ANS2 | ANS1 | ANS0 | `-000 1111` | `-000 1111` |

**Legend:** x = unknown, u = unchanged, — = unimplemented locations read as ‘`0`’. Shaded cells are not used by GPIO.

## 5.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 5-1 is a block diagram of the Timer0 module and the prescaler shared with the WDT.

| **Note:** | Additional information on the Timer0 module is available in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023). |
|---|---|

### 5.1 Timer0 Operation

Timer mode is selected by clearing the T0CS bit (OPTION_REG<5>). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If TMR0 is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register.

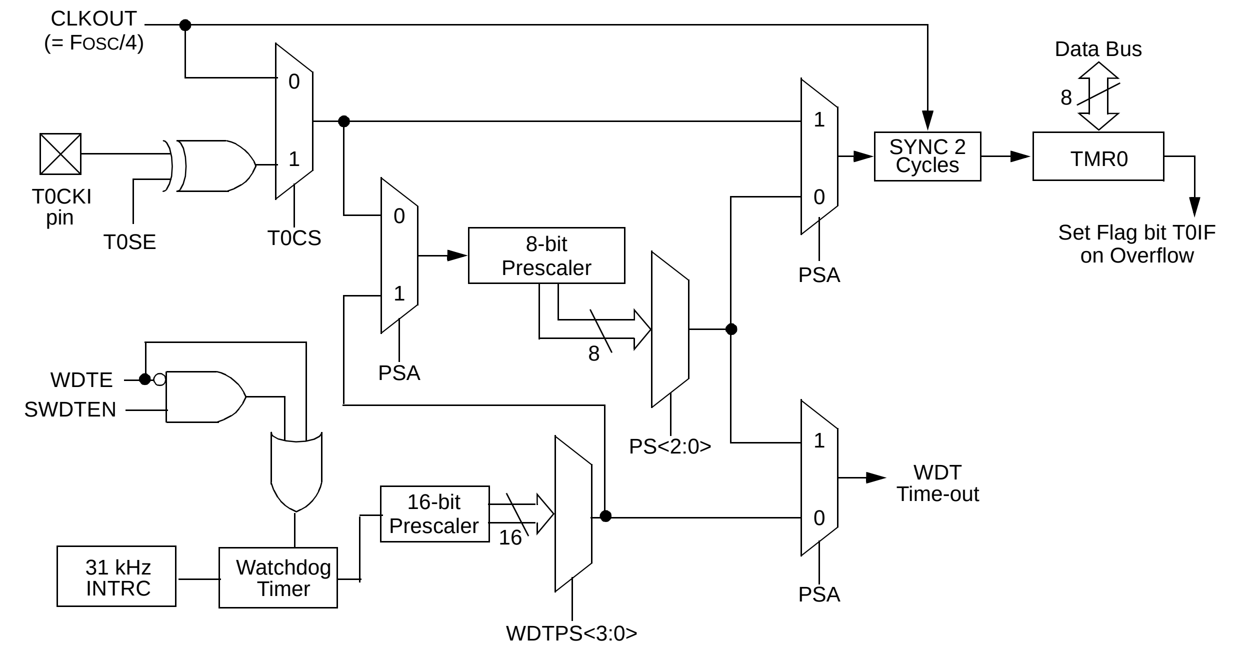
Counter mode is selected by setting the T0CS bit (OPTION_REG<5>). In this mode, the Timer0 module will increment either on every rising or falling edge of pin GP2/T0CKI. The incrementing edge is determined by the source edge (T0SE) control bit (OPTION_REG<4>). Clearing the T0SE bit selects the rising edge.

| **Note:** | Counter mode has specific external clock requirements. Additional information on these requirements is available in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023). |
|---|---|

### 5.2 Timer0 Interrupt

A Timer0 interrupt is generated when the TMR0 register timer/counter overflows from FFh to 00h. This overflow sets the T0IF bit (INTCON<2>). The interrupt can be masked by clearing the T0IE bit (INTCON<5>). The T0IF bit must be cleared in software by the Timer0 module Interrupt Service Routine before re-enabling this interrupt. The Timer0 interrupt cannot wake the processor from Sleep since the timer is shut off during Sleep.

**FIGURE 5-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**



**Note 1:** T0SE, T0CS, PSA and PS<2:0> are bits in the Option register, WDTPS<3:0> are bits in the WDTCON register.

## 5.3 Using Timer0 with an External Clock

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. Therefore, it is necessary for T0CKI to be high for at least 2 TOSC (and a small RC delay of 20 ns) and low for at least 2 TOSC (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

**Note:** The ANSEL (9Fh) and CMCON0 (19h) registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read ‘0’.

## 5.4 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer. For simplicity, this counter will be referred to as “prescaler” throughout this data sheet. The prescaler assignment is controlled in software by the control bit PSA (OPTION_REG<3>). Clearing the PSA bit will assign the prescaler to Timer0. Prescale values are selectable via the PS<2:0> bits (OPTION_REG<2:0>).

The prescaler is not readable or writable. When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF 1, MOVWF 1, BSF 1, x....etc.`) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer.

### 5.4.1 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 5-1 and Example 5-2) must be executed when changing the prescaler assignment from Timer0 to WDT.

**EXAMPLE 5-1: CHANGING PRESCALER (TIMER0 → WDT)**

```
BCF     STATUS,RP0      ;Bank 0
CLRWDT                  ;Clear WDT
CLRF    TMR0            ;Clear TMR0 and
                        ; prescaler
BSF     STATUS,RP0      ;Bank 1

MOVLW   b’00101111’     ;Required if desired
MOVWF   OPTION_REG      ; PS2:PS0 is
CLRWDT                  ; 000 or 001
                        ;
MOVLW   b’00101xxx’     ;Set postscaler to
MOVWF   OPTION_REG      ; desired WDT rate
BCF     STATUS,RP0      ;Bank 0
```

To change prescaler from the WDT to the TMR0 module, use the sequence shown in Example 5-2. This precaution must be taken even if the WDT is disabled.

**EXAMPLE 5-2: CHANGING PRESCALER (WDT → TIMER0)**

```
CLRWDT                  ;Clear WDT and
                        ; prescaler
BSF     STATUS,RP0      ;Bank 1

MOVLW   b’xxxx0xxx’     ;Select TMR0,
                        ; prescale, and
                        ; clock source
MOVWF   OPTION_REG      ;
BCF     STATUS,RP0      ;Bank 0
```

**TABLE 5-1: REGISTERS ASSOCIATED WITH TIMER0**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h | TMR0 | Timer0 Module Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | GPIE | T0IF | INTF | GPIF | 0000 0000 | 0000 0000 |
| 81h | OPTION_REG | $\overline{\text{GPPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 85h | TRISIO | — | — | TRISIO5 | TRISIO4 | TRISIO3 | TRISIO2 | TRISIO1 | TRISIO0 | --11 1111 | --11 1111 |

**Legend:** — = Unimplemented locations, read as ‘0’, u = unchanged, x = unknown. Shaded cells are not used by the Timer0 module.

## 6.0 TIMER1 MODULE WITH GATE CONTROL

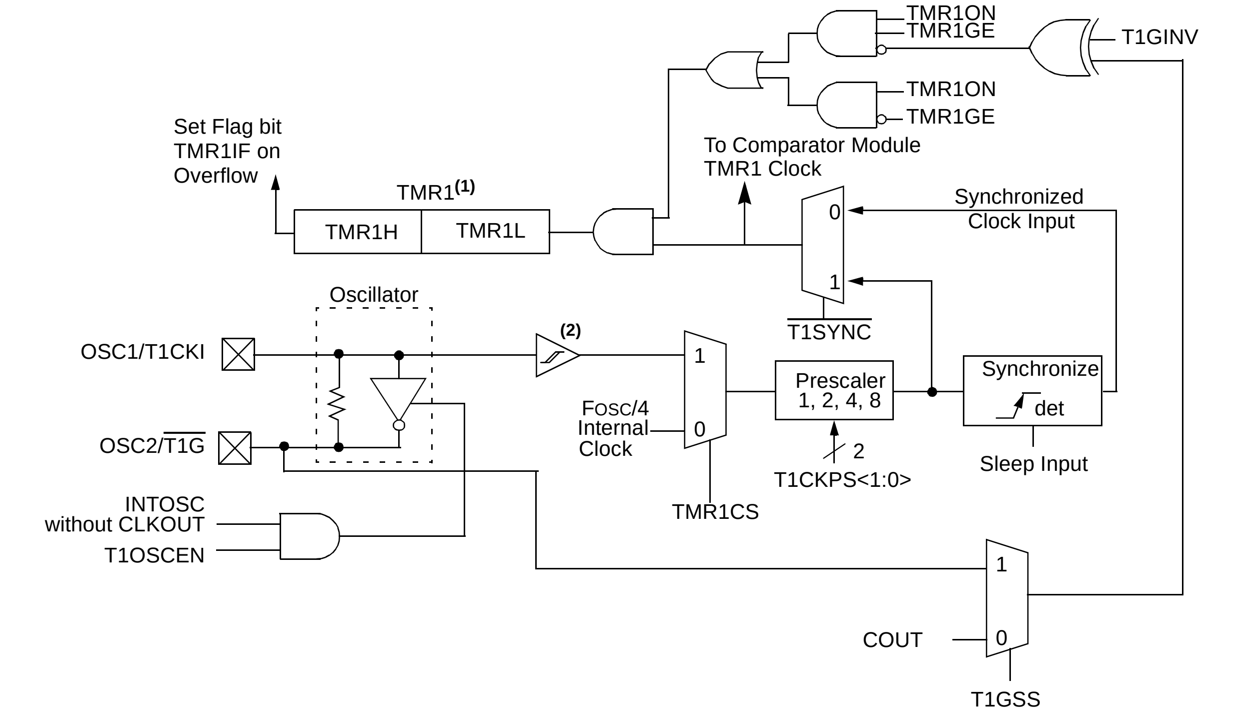
The PIC12F683 has a 16-bit timer. Figure 6-1 shows the basic block diagram of the Timer1 module. Timer1 has the following features:

- 16-bit timer/counter (TMR1H:TMR1L)
- Readable and writable
- Internal or external clock selection
- Synchronous or asynchronous operation
- Interrupt on overflow from FFFFh to 0000h
- Wake-up upon overflow (Asynchronous mode)
- Optional external enable input
  - Selectable gate source: $\overline{\text{T1G}}$ or COUT (T1GSS)
  - Selectable gate polarity (T1GINV)
- Optional LP oscillator

The Timer1 Control register (T1CON), shown in Register 6-1, is used to enable/disable Timer1 and select the various features of the Timer1 module.

| **Note:** | Additional information on timer modules is available in the *PICmicro® Mid-Range MCU Family Reference Manual* (DS33023). |
|---|---|

**FIGURE 6-1: TIMER1 ON THE PIC12F683 BLOCK DIAGRAM**



**Note 1:** Timer1 increments on the rising edge.

**2:** ST Buffer is low-power type when using LP oscillator or high-speed type when using T1CKI.

## 6.1 Timer1 Modes of Operation

Timer1 can operate in one of three modes:

- 16-bit timer with prescaler
- 16-bit synchronous counter
- 16-bit asynchronous counter

In Timer mode, Timer1 is incremented on every instruction cycle. In Counter mode, Timer1 is incremented on the rising edge of the external clock input T1CKI. In addition, the Counter mode clock can be synchronized to the microcontroller system clock or run asynchronously.

In Counter and Timer modules, the counter/timer clock can be gated by the Timer 1 gate, which can be selected as either the $\overline{\text{T1G}}$ pin or the comparator output.

If an external clock oscillator is needed (and the microcontroller is using the INTOSC w/o CLKOUT), Timer1 can use the LP oscillator as a clock source.

**Note:** In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge.

## 6.2 Timer1 Interrupt

The Timer1 register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When Timer1 rolls over, the Timer1 interrupt flag bit (PIR1<0>) is set. To enable the interrupt on rollover, you must set these bits:

- Timer1 interrupt enable bit (PIE1<0>)
- PEIE bit (INTCON<6>)
- GIE bit (INTCON<7>).

The interrupt is cleared by clearing the TMR1IF bit in the Interrupt Service Routine.

**Note:** The TMR1H:TTMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts.

## 6.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits (T1CON<5:4>) control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

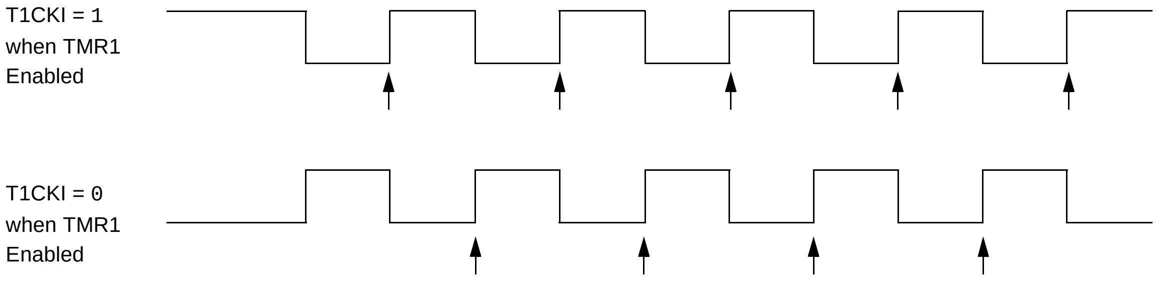
## 6.4 Timer1 Gate

Timer1 gate source is software configurable to be the $\overline{\text{T1G}}$ pin or the output of the comparator. This allows the device to directly time external events using $\overline{\text{T1G}}$ or analog events using the comparator. See CMCON1 (Register 8-2) for selecting the Timer1 gate source. This feature can simplify the software for a Delta-Sigma A/D converter and many other applications. For more information on Delta-Sigma A/D Converters, see the Microchip web site (www.microchip.com).

**Note:** TMR1GE bit (T1CON<6>) must be set to use either $\overline{\text{T1G}}$ or COUT as the Timer1 gate source. See Register 8-2 for more information on selecting the Timer1 gate source.

Timer1 gate can be inverted by using the T1GINV bit (T1CON<7>), whether it originates from the $\overline{\text{T1G}}$ pin or the comparator output. This configures Timer1 to measure either the active-high or active-low time between events.

**FIGURE 6-2: TIMER1 INCREMENTING EDGE**



**Note 1:** Arrows indicate counter increments.

**2:** In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge of the clock.

### REGISTER 6-1: T1CON – TIMER1 CONTROL REGISTER (ADDRESS: 10h)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON |
| bit 7 | | | | | | | bit 0 |

bit 7 **T1GINV:** Timer1 Gate Invert bit[1]

1 = Timer1 gate is inverted
0 = Timer1 gate is not inverted

bit 6 **TMR1GE:** Timer1 Gate Enable bit[2]

If TMR1ON = 0:
This bit is ignored.

If TMR1ON = 1:
1 = Timer1 is on if Timer1 gate is not active
0 = Timer1 is on

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits

11 = 1:8 Prescale Value
10 = 1:4 Prescale Value
01 = 1:2 Prescale Value
00 = 1:1 Prescale Value

bit 3 **T1OSCEN:** LP Oscillator Enable Control bit

If INTOSC without CLKOUT oscillator is active:
1 = LP oscillator is enabled for Timer1 clock
0 = LP oscillator is off

Else:
This bit is ignored.

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Control bit

TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input

TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit

1 = External clock from T1CKI pin (on the rising edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR1ON:** Timer1 On bit

1 = Enables Timer1
0 = Stops Timer1

**Note 1:** T1GINV bit inverts the Timer1 gate logic, regardless of source.

**2:** TMR1GE bit must be set to use either $\overline{\text{T1G}}$ pin or COUT, as selected by the T1GSS bit (CMCON1<1>), as a Timer1 gate source.

| **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 |

## 6.5 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ (T1CON<2>) is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 6.5.1 “Reading and Writing Timer1 in Asynchronous Counter Mode”**).

| **Note:** | The ANSEL (9Fh) and CMCON0 (19h) registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read ‘0’. |
|---|---|

### 6.5.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the timer register.

Reading the 16-bit value requires some care. Examples in the “*PICmicro® Mid-Range MCU Family Reference Manual* (DS33023) show how to read and write Timer1 when it is running in Asynchronous mode.

## 6.6 Timer1 Oscillator

A crystal oscillator circuit is built-in between pins OSC1 (input) and OSC2 (amplifier output). It is enabled by setting control bit, T1OSCEN (T1CON<3>). The oscillator is a low-power oscillator rated up to 32 kHz. It will continue to run during Sleep. It is primarily intended for a 32 kHz crystal. Table 3-1 shows the capacitor selection for the Timer1 oscillator.

The Timer1 oscillator is shared with the system LP oscillator. Thus, Timer1 can use this mode only when the primary system clock is derived from the internal oscillator. As with the system LP oscillator, the user must provide a software time delay to ensure proper oscillator start-up.

TRISIO5 and TRISIO4 bits are set when the Timer1 oscillator is enabled. GP5 and GP4 read as ‘0’ and TRISIO5 and TRISIO4 bits read as ‘1’.

| **Note:** | The oscillator requires a start-up and stabilization time before use. Thus, T1OSCEN should be set and a suitable delay observed prior to enabling Timer1. |
|---|---|

## 6.7 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when setup in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- Timer1 must be on (T1CON<0>)
- TMR1IE bit (PIE1<0>) must be set
- PEIE bit (INTCON<6>) must be set

The device will wake-up on an overflow. If the GIE bit (INTCON<7>) is set, the device will wake-up and jump to the Interrupt Service Routine (0004h) on an overflow. If the GIE bit is clear, execution will continue with the next instruction.

**TABLE 6-1: REGISTERS ASSOCIATED WITH TIMER1**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh/ 8Bh | INTCON | GIE | PEIE | T0IE | INTE | GPIE | T0IF | INTF | GPIF | 0000 0000 | 0000 0000 |
| 0Ch | PIR1 | EEIF | ADIF | CCP1IF | — | CMIF | OSFIF | TMR2IF | TMR1IF | 000- 0000 | 000- 0000 |
| 0Eh | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | 0000 0000 | uuuu uuuu |
| 1Ah | CMCON1 | — | — | — | — | — | — | T1GSS | CMSYNC | ---- --10 | ---- --10 |
| 8Ch | PIE1 | EEIE | ADIE | CCP1IE | — | CMIE | OSFIE | TMR2IE | TMR1IE | 000- 0000 | 000- 0000 |

**Legend:** x = unknown, u = unchanged, — = unimplemented, read as ‘0’. Shaded cells are not used by the Timer1 module.

## 7.0 TIMER2 MODULE

The Timer2 module timer has the following features:

- 8-bit timer (TMR2 register)
- 8-bit period register (PR2)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR2 match with PR2

Timer2 has a control register shown in Register 7-1. TMR2 can be shut off by clearing control bit, TMR2ON (T2CON<2>), to minimize power consumption. Figure 7-1 is a simplified block diagram of the Timer2 module. The prescaler and postscaler selection of Timer2 are controlled by this register.

## 7.1 Timer2 Operation

Timer2 can be used as the PWM time base for the PWM mode of the CCP module. The TMR2 register is readable and writable and is cleared on any device Reset. The input clock (FOSC/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits, T2CKPS<1:0> (T2CON<1:0>). The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit, TMR2IF (PIR1<1>)).

The prescaler and postscaler counters are cleared when any of the following occurs:

- A write to the TMR2 register
- A write to the T2CON register
- Any device Reset (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

### REGISTER 7-1: T2CON – TIMER2 CONTROL REGISTER (ADDRESS: 12h)

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **Unimplemented:** Read as ‘0’

bit 6-3 **TOUTPS<3:0>:** Timer2 Output Postscale Select bits
0000 = 1:1 postscale
0001 = 1:2 postscale
•
•
•
1111 = 1:16 postscale

bit 2 **TMR2ON:** Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

| **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 |

## 7.2 Timer2 Interrupt

The Timer2 module has an 8-bit period register, PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon Reset.

**FIGURE 7-1: TIMER2 BLOCK DIAGRAM**

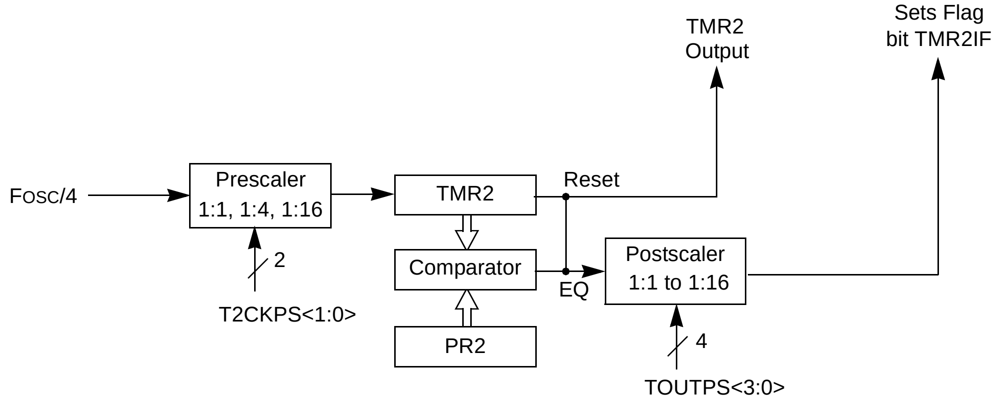


**TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER2**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh/<br>8Bh | INTCON | GIE | PEIE | T0IE | INTE | GPIE | T0IF | INTF | GPIF | 0000 0000 | 0000 0000 |
| 0Ch | PIR1 | EEIF | ADIF | CCP1IF | — | CMIF | OSFIF | TMR2IF | TMR1IF | 000- 0000 | 000- 0000 |
| 11h | TMR2 | Holding Register for the 8-bit TMR2 Register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 8Ch | PIE1 | EEIE | ADIE | CCP1IE | — | CMIE | OSFIE | TMR2IE | TMR1IE | 000- 0000 | 000- 0000 |
| 92h | PR2 | Timer2 Module Period Register | | | | | | | | 1111 1111 | 1111 1111 |

**Legend:** x = unknown, u = unchanged, — = unimplemented, read as ‘0’. Shaded cells are not used by the Timer2 module.

## 8.0 COMPARATOR MODULE

The comparator module contains one analog comparator. The inputs to the comparator are multiplexed with I/O port pins, GP0 and GP1, while the outputs are multiplexed to GP2. An on-chip Comparator Voltage Reference (CVREF) can also be applied to the inputs of the comparator.

The CMCON0 register (Register 8-1) controls the comparator input and output multiplexers. A block diagram of the various comparator configurations is shown in Figure 8-3.

**REGISTER 8-1: CMCON0 – COMPARATOR CONTROL REGISTER 0 (ADDRESS: 19h)**

| U-0 | R-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | COUT | — | CINV | CIS | CM2 | CM1 | CM0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **Unimplemented**: Read as ‘0’

bit 6 **COUT**: Comparator Output bit

When CINV = 0:
1 = VIN+ > VIN-
0 = VIN+ < VIN-

When CINV = 1:
1 = VIN+ < VIN-
0 = VIN+ > VIN-

bit 5 **Unimplemented**: Read as ‘0’

bit 4 **CINV**: Comparator Output Inversion bit
1 = Output inverted
0 = Output not inverted

bit 3 **CIS**: Comparator Input Switch bit

When CM<2:0> = 110 or 101:
1 = VIN- connects to CIN+
0 = VIN- connects to CIN-

bit 2 **CM<2:0>**: Comparator Mode bits

Figure 8-3 shows the Comparator modes and CM<2:0> bit settings.

| **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 |

## 8.1 Comparator Operation

A single comparator is shown in Figure 8-1 along with the relationship between the analog input levels and the digital output. When the analog input at VIN+ is less than the analog input VIN-, the output of the comparator is a digital low level. When the analog input at VIN+ is greater than the analog input VIN-, the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 8-1 represent the uncertainty due to input offsets and response time.

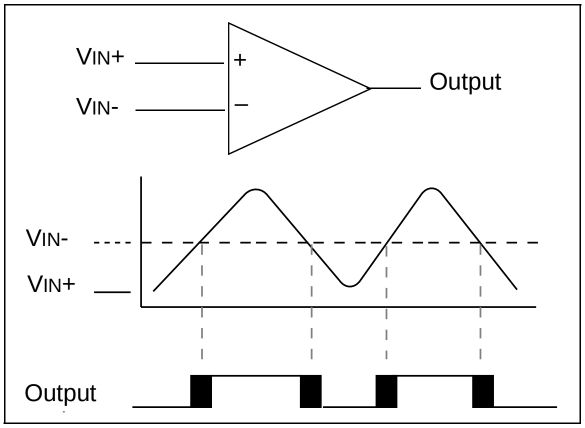
> **Note:** To use CIN+ and CIN- pins as analog inputs, the appropriate bits must be programmed in the CMCON0 (19h) register.

The polarity of the comparator output can be inverted by setting the CINV bit (CMCON0<4>). Clearing CINV results in a non-inverted output. A complete table showing the output state versus input conditions and the polarity bit is shown in Table 8-1.

**TABLE 8-1: OUTPUT STATE VS. INPUT CONDITIONS**

| Input Conditions | CINV | COUT |
|---|---|---|
| VIN- > VIN+ | 0 | 0 |
| VIN- < VIN+ | 0 | 1 |
| VIN- > VIN+ | 1 | 1 |
| VIN- < VIN+ | 1 | 0 |

**FIGURE 8-1: SINGLE COMPARATOR**

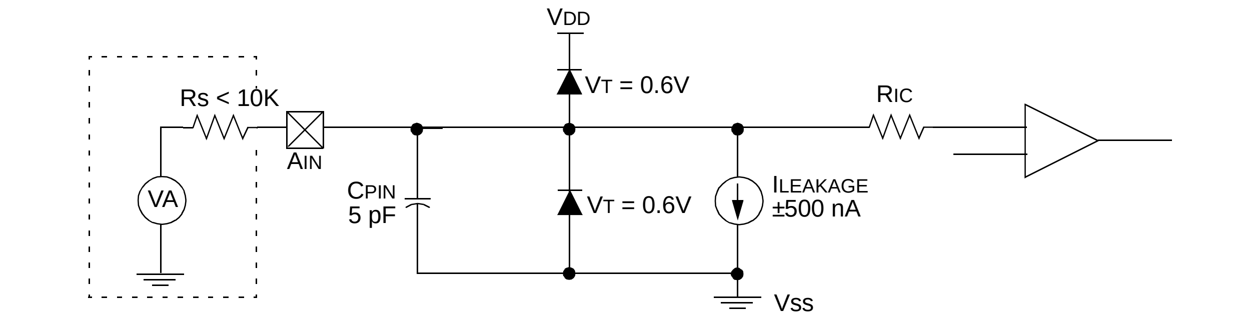


## 8.2 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 8-2. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur. A maximum source impedance of 10 kΩ is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

> **Note 1:** When reading the GPIO register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as analog inputs according to the input specification.
>
> **2:** Analog levels on any pin defined as a digital input may cause the input buffer to consume more current than is specified.

**FIGURE 8-2: ANALOG INPUT MODEL**



**Legend**:

| | |
|---|---|
| CPIN | = Input Capacitance |
| VT | = Threshold Voltage |
| ILEAKAGE | = Leakage Current at the pin due to various junctions |
| RIC | = Interconnect Resistance |
| RS | = Source Impedance |
| VA | = Analog Voltage |

## 8.3 Comparator Configuration

There are eight modes of operation for the comparator. The CMCON0 register is used to select these modes. Figure 8-3 shows the eight possible modes.

If the Comparator mode is changed, the comparator output level may not be valid for the specified mode change delay shown in **Section 15.0 “Electrical Specifications”**.

| **Note:** | Comparator interrupts should be disabled during a Comparator mode change. Otherwise, a false interrupt may occur. |
|---|---|

**FIGURE 8-3: COMPARATOR I/O OPERATING MODES**

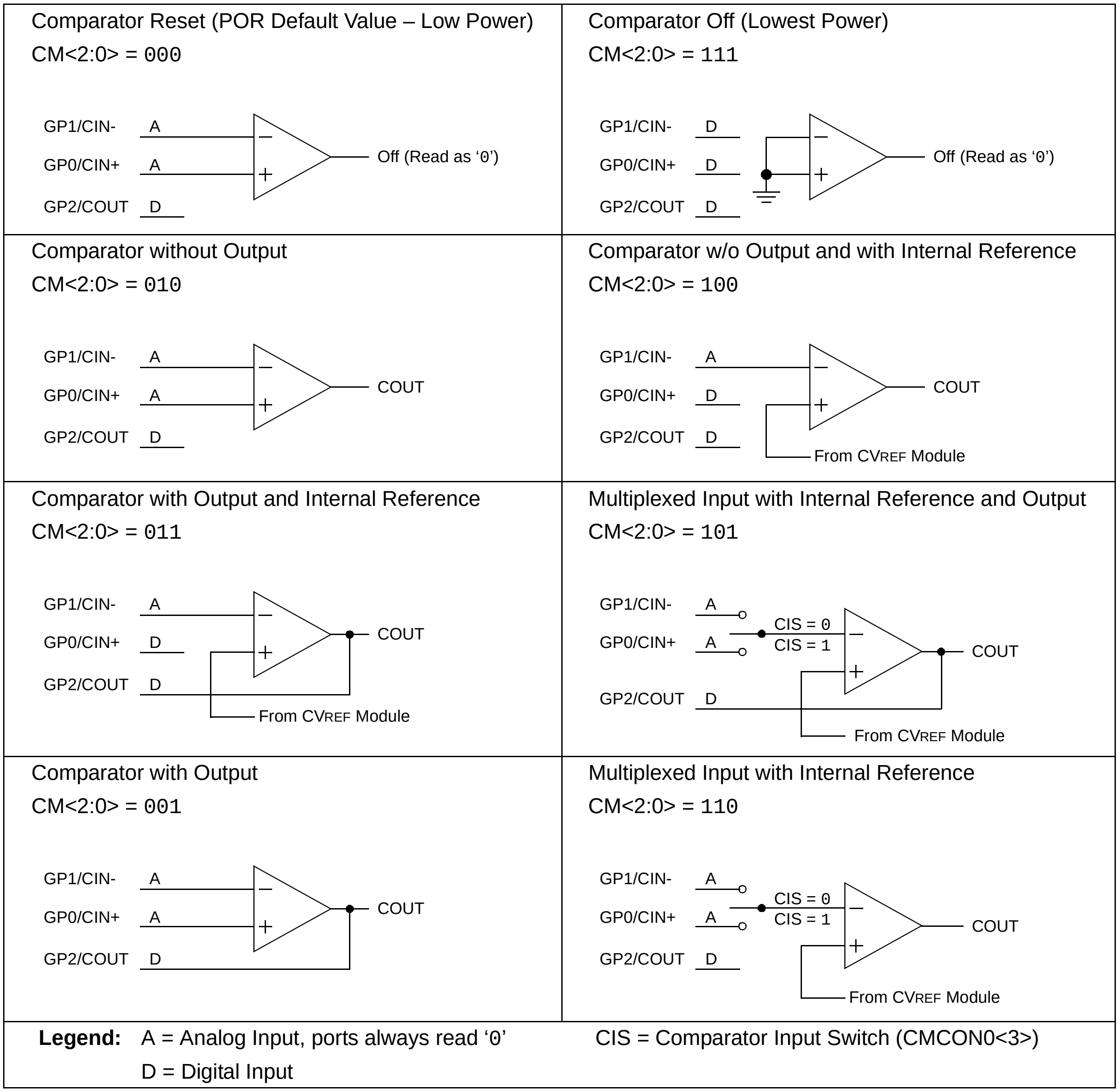
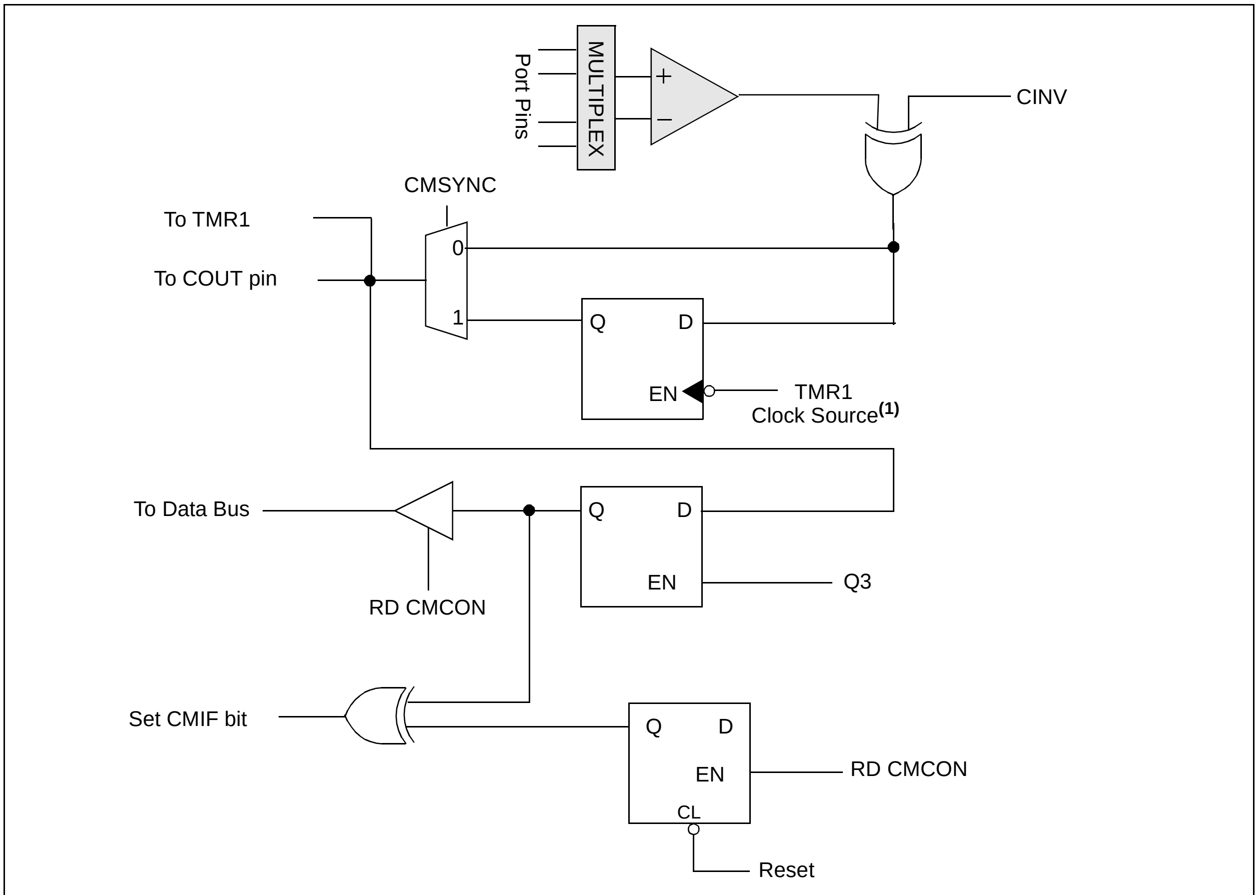


**FIGURE 8-4: COMPARATOR OUTPUT BLOCK DIAGRAM**



**Note 1:** Comparator output is latched on falling edge of T1 clock source.

### REGISTER 8-2: CMCON1 – COMPARATOR CONTROL REGISTER 1 (ADDRESS: 1Ah)

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-1 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | T1GSS | CMSYNC |
| bit 7 | | | | | | | bit 0 |

bit 7-2: **Unimplemented**: Read as ‘0’

bit 1 **T1GSS:** Timer1 Gate Source Select bit
1 = Timer1 gate source is $\overline{\text{T1G}}$ pin (GP4 must be configured as digital input)
0 = Timer1 gate source is comparator output

bit 0 **CMSYNC:** Comparator Synchronize bit
1 = COUT output synchronized with falling edge of Timer1 clock
0 = COUT output not synchronized with Timer1 clock

| **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 |

## 8.4 Comparator Output

The comparator output is read through the CMCON0 register. This bit is read-only. The comparator output may also be directly output to the GP2 pin. When enabled, multiplexors in the output path of the GP2 pin will switch and the output will be the unsynchronized output of the comparator. The uncertainty of the comparator is related to the input offset voltage and the response time given in the specifications. Figure 8-4 shows the output block diagram for the comparator.

The TRISIO bit will still function as an output enable/ disable for the GP2 pin while in this mode.

The polarity of the comparator outputs can be changed using the CINV bit (CMCON0<4>).

Timer1 gate source can be configured to use the $\overline{\text{T1G}}$ pin or the comparator output as selected by the T1GSS bit (CMCON1<1>). This feature can be used to time the duration or interval of analog events. The output of the comparator can also be synchronized with Timer1 by setting the CMSYNC bit (CMCON1<0>). When enabled, the output of the comparator is latched on the falling edge of the Timer1 clock source. If a prescaler is used with Timer1, the comparator is latched after the prescaler. To prevent a race condition, the comparator output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. See Figure 8-4, Comparator Output Block Diagram and Figure 6-1, Timer1 on the PIC12F683 Block Diagram for more information.

It is recommended to synchronize the comparator with Timer1 by setting the CMSYNC bit when the comparator is used as the Timer1 gate source. This ensures Timer1 does not miss an increment if the comparator changes during an increment.

## 8.5 Comparator Interrupt

The comparator interrupt flag is set whenever there is a change in the output value of the comparator. Software will need to maintain information about the status of the output bit, as read from CMCON0<6>, to determine the actual change that has occurred. The CMIF bit (PIR1<3>) is the Comparator Interrupt Flag. This bit must be reset in software by clearing it to ‘0’. Since it is also possible to write a ‘1’ to this register, a simulated interrupt may be initiated.

The CMIE bit (PIE1<3>) and the PEIE bit (INTCON<6>) must be set to enable the interrupts. In addition, the GIE bit must also be set. If any of these bits are cleared, the interrupt is not enabled, though the CMIF bit will still be set if an interrupt condition occurs.

The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

a) Any read or write of CMCON0. This will end the mismatch condition.

b) Clear flag bit CMIF.

A mismatch condition will continue to set flag bit CMIF. Reading CMCON0 will end the mismatch condition and allow flag bit CMIF to be cleared.

**Note:** If a change in the CMCON0 register (COUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF (PIR1<3>) interrupt flag may not get set.

## 8.6 Comparator Reference

The comparator module also allows the selection of an internally generated voltage reference for one of the comparator inputs. The VRCON register, Register 8-3, controls the voltage reference module shown in Figure 8-5.

### 8.6.1 CONFIGURING THE VOLTAGE REFERENCE

The voltage reference can output 32 distinct voltage levels, 16 in a high range and 16 in a low range.

The following equation determines the output voltages:

**EQUATION 8-1:**

*VRR = 1 (Low Range): CVREF = (VR3:VR0/24) x VDD*

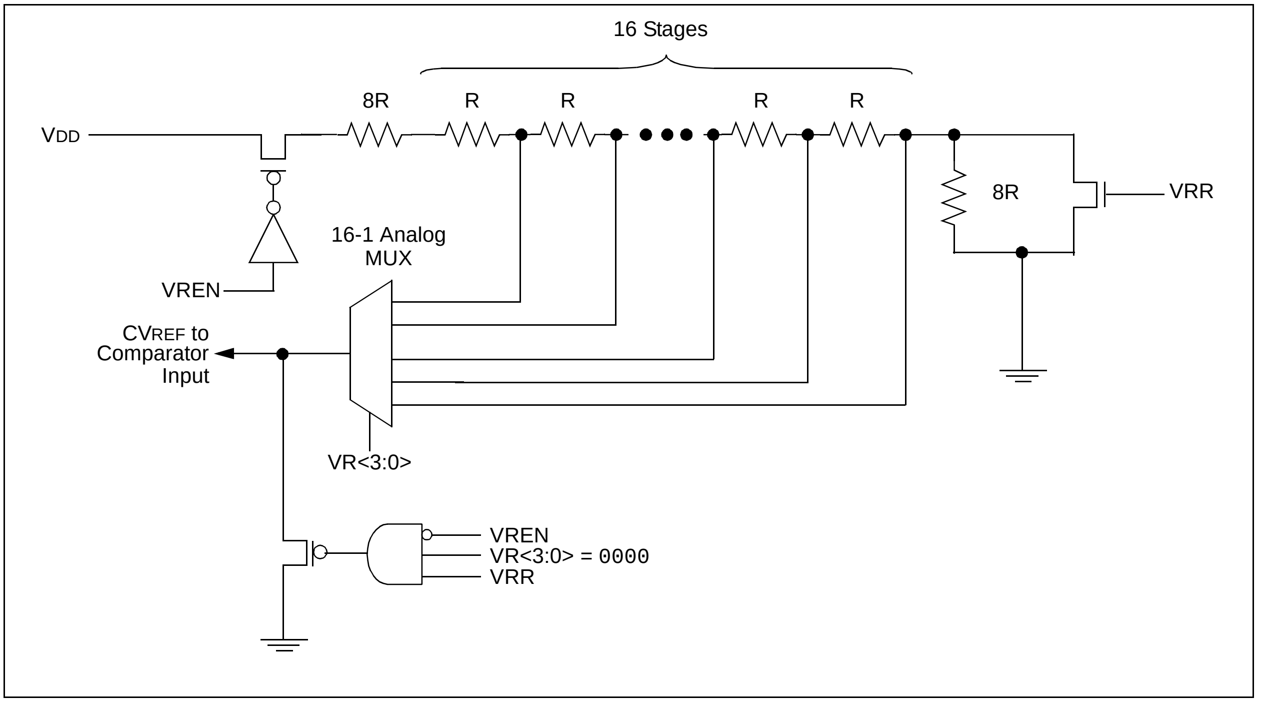
*VRR = 0 (High Range):*
*CVREF = (VDD/4) + (VR3:VR0 X VDD/32)*

### 8.6.2 VOLTAGE REFERENCE ACCURACY/ERROR

The full range of VSS to VDD cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network (Figure 8-5) keep CVREF from approaching VSS or VDD. The exception is when the module is disabled by clearing the VREN bit (VRCON<7>). When disabled, the reference voltage is VSS when VR<3:0> is '0000' and the VRR (VRCON<5>) bit is set. This allows the comparator to detect a zero-crossing and not consume CVREF module current.

The voltage reference is VDD derived and therefore, the CVREF output changes with fluctuations in VDD. The tested absolute accuracy of the comparator voltage reference can be found in **Section 15.0 "Electrical Specifications"**.

**FIGURE 8-5: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM**



## 8.7 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output is ensured to have a valid level. If the internal reference is changed, the maximum delay of the internal voltage reference must be considered when using the comparator output. Otherwise, the maximum delay of the comparator should be used (Table 15-8).

## 8.8 Operation During Sleep

The comparator and voltage reference, if enabled before entering Sleep mode, remain active during Sleep. This results in higher Sleep currents than shown in the power-down specifications. The additional current consumed by the comparator and the voltage reference is shown separately in the specifications. To minimize power consumption while in Sleep mode, turn off the comparator, CM<2:0> = 111 and voltage reference, VRCON<7> = 0.

While the comparator is enabled during Sleep, an interrupt will wake-up the device. If the GIE bit (INTCON<7>) is set, the device will jump to the interrupt vector (0004h) and if clear, continues execution with the next instruction. If the device wakes up from Sleep, the contents of the CMCON0, CMCON1 and VRCON registers are not affected.

## 8.9 Effects of a Reset

A device Reset forces the CMCON0, CMCON1 and VRCON registers to their Reset states. This forces the comparator module to be in the Comparator Reset mode, CM<2:0> = 000 and the voltage reference to its off state. Thus, all potential inputs are analog inputs with the comparator and voltage reference disabled to consume the smallest current possible.

**REGISTER 8-3: VRCON – VOLTAGE REFERENCE CONTROL REGISTER (ADDRESS: 99h)**

| R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| VREN | — | VRR | — | VR3 | VR2 | VR1 | VR0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **VREN:** CVREF Enable bit
1 = CVREF circuit powered on
0 = CVREF circuit powered down, no IDD drain and CVREF = VSS

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **VRR:** CVREF Range Selection bit
1 = Low range
0 = High range

bit 4 **Unimplemented:** Read as ‘0’

bit 3-0 **VR<3:0>:** CVREF Value Selection 0 ≤ VR <3:0> ≤ 15
When VRR = 1: CVREF = (VR<3:0>/24) * VDD
When VRR = 0: CVREF = VDD/4 + (VR<3:0>/32) * VDD

| **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 |

### TABLE 8-2: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

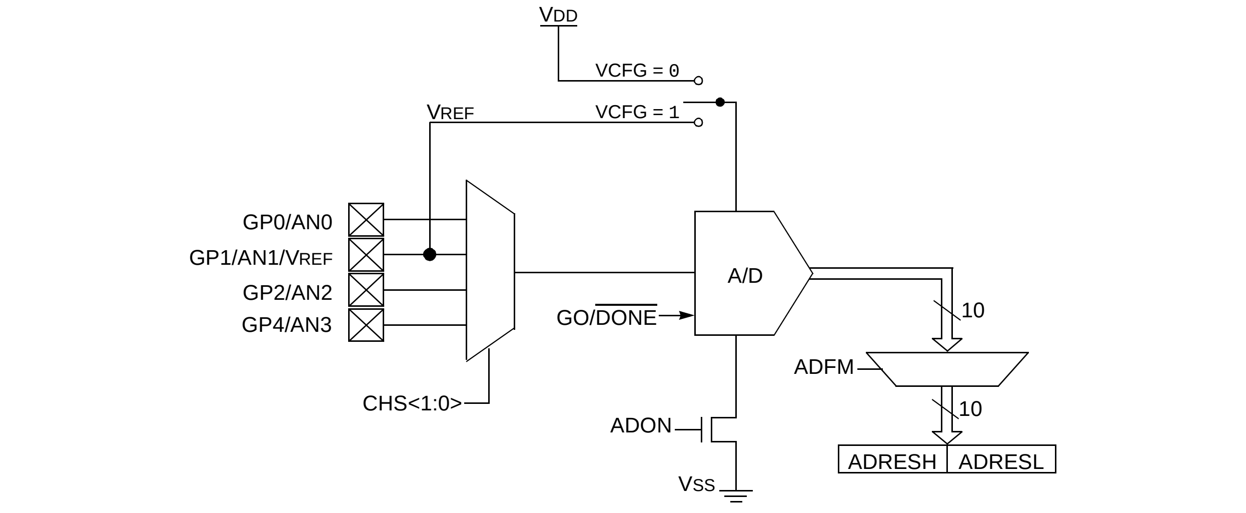
| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | GPIE | T0IF | INTF | GPIF | `0000 0000` | `0000 0000` |
| 0Ch | PIR1 | EEIF | ADIF | CCP1IF | — | CMIF | OSFIF | TMR2IF | TMR1IF | `000- 0000` | `000- 0000` |
| 19h | CMCON0 | — | COUT | — | CINV | CIS | CM2 | CM1 | CM0 | `-0-0 0000` | `-0-0 0000` |
| 1Ah | CMCON1 | — | — | — | — | — | — | T1GSS | CMSYNC | `---- --10` | `---- --10` |
| 85h | TRISIO | — | — | TRISIO5 | TRISIO4 | TRISIO3 | TRISIO2 | TRISIO1 | TRISIO0 | `--11 1111` | `--11 1111` |
| 8Ch | PIE1 | EEIE | ADIE | CCPIE | — | CMIE | OSFIE | TMR2IE | TMR1IE | `000- 0000` | `000- 0000` |
| 99h | VRCON | VREN | — | VRR | — | VR3 | VR2 | VR1 | VR0 | `0-0- 0000` | `0-0- 0000` |

**Legend:** x = unknown, u = unchanged, — = unimplemented, read as ‘0’. Shaded cells are not used by the comparator or comparator voltage reference module.

## 9.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital converter (A/D) allows conversion of an analog input signal to a 10-bit binary representation of that signal. The PIC12F683 has four analog inputs, multiplexed into one sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a binary result via successive approximation and stores the result in a 10-bit register. The voltage reference used in the conversion is software selectable to either VDD or a voltage applied by the VREF pin. Figure 9-1 shows the block diagram of the A/D on the PIC12F683.

**FIGURE 9-1: A/D BLOCK DIAGRAM**



## 9.1 A/D Configuration and Operation

There are two registers available to control the functionality of the A/D module:

1. ADCON0 (Register 9-1)
2. ANSEL (Register 9-2)

### 9.1.1 ANALOG PORT PINS

The ANS<3:0> bits (ANSEL<3:0>) and the TRISIO bits control the operation of the A/D port pins. Set the corresponding TRISIO bits to set the pin output driver to its high-impedance state. Likewise, set the corresponding ANSEL bit to disable the digital input buffer.

| **Note:** | Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current. |
|---|---|

### 9.1.2 CHANNEL SELECTION

There are four analog channels on the PIC12F683, AN0 through AN3. The CHS bits (ADCON0<3:2>) control which channel is connected to the sample and hold circuit.

### 9.1.3 VOLTAGE REFERENCE

There are two options for the voltage reference to the A/D converter: either VDD is used, or an analog voltage applied to VREF is used. The VCFG bit (ADCON0<6>) controls the voltage reference selection. If VCFG is set, then the voltage on the VREF pin is the reference; otherwise, VDD is the reference.

### 9.1.4 CONVERSION CLOCK

The A/D conversion cycle requires 11 TAD. The source of the conversion clock is software selectable via the ADCS bits (ANSEL<6:4>). There are seven possible clock options:

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC (dedicated internal oscillator)

For correct conversion, the A/D conversion clock (1/TAD) must be selected to ensure a minimum TAD of 1.6 µs. Table 9-1 shows a few TAD calculations for selected frequencies.

### TABLE 9-1: TAD vs. DEVICE OPERATING FREQUENCIES

| A/D Clock Source (TAD) | | Device Frequency | | | |
|---|---|---|---|---|---|
| **Operation** | **ADCS<2:0>** | **20 MHz** | **5 MHz** | **4 MHz** | **1.25 MHz** |
| 2 TOSC | `000` | 100 ns(2) | 400 ns(2) | 500 ns(2) | 1.6 µs |
| 4 TOSC | `100` | 200 ns(2) | 800 ns(2) | 1.0 µs(2) | 3.2 µs |
| 8 TOSC | `001` | 400 ns(2) | 1.6 µs | 2.0 µs | 6.4 µs |
| 16 TOSC | `101` | 800 ns(2) | 3.2 µs | 4.0 µs | 12.8 µs(3) |
| 32 TOSC | `010` | 1.6 µs | 6.4 µs | 8.0 µs(3) | 25.6 µs(3) |
| 64 TOSC | `110` | 3.2 µs | 12.8 µs(3) | 16.0 µs(3) | 51.2 µs(3) |
| A/D RC | `x11` | 2-6 µs(1,4) | 2-6 µs(1,4) | 2-6 µs(1,4) | 2-6 µs(1,4) |

**Legend:** Shaded cells are outside of recommended range.

**Note 1:** The A/D RC source has a typical TAD time of 4 µs for VDD > 3.0V.

**2:** These values violate the minimum required TAD time.

**3:** For faster conversion times, the selection of another clock source is recommended.

**4:** When the device frequency is greater than 1 MHz, the A/D RC clock source is only recommended if the conversion will be performed during Sleep.

### 9.1.5 STARTING A CONVERSION

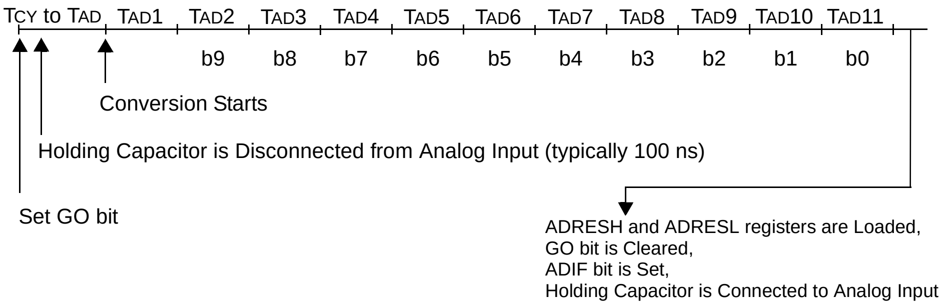
The A/D conversion is initiated by setting the GO/$\overline{\text{DONE}}$ bit (ADCON0<1>). When the conversion is complete, the A/D module:

- Clears the GO/$\overline{\text{DONE}}$ bit
- Sets the ADIF flag (PIR1<6>)
- Generates an interrupt (if enabled)

If the conversion must be aborted, the GO/$\overline{\text{DONE}}$ bit can be cleared in software. The ADRESH:ADRESL registers will not be updated with the partially complete A/D conversion sample. Instead, the ADRESH:ADRESL registers will retain the value of the previous conversion. After an aborted conversion, a 2 TAD delay is required before another acquisition can be initiated. Following the delay, an input acquisition is automatically started on the selected channel.

**Note:** The GO/$\overline{\text{DONE}}$ bit should not be set in the same instruction that turns on the A/D.

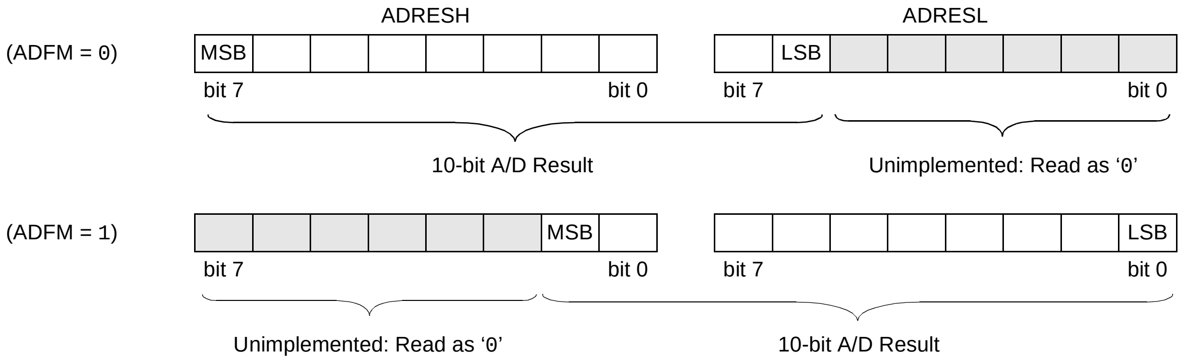
**FIGURE 9-2: A/D CONVERSION TAD CYCLES**


TCY to TAD
TAD1
TAD2
TAD3
TAD4
TAD5
TAD6
TAD7
TAD8
TAD9
TAD10
TAD11
b9
b8
b7
b6
b5
b4
b3
b2
b1
b0
Conversion Starts
Holding Capacitor is Disconnected from Analog Input (typically 100 ns)
Set GO bit
ADRESH and ADRESL registers are Loaded,
GO bit is Cleared,
ADIF bit is Set,
Holding Capacitor is Connected to Analog Input

### 9.1.6 CONVERSION OUTPUT

The A/D conversion can be supplied in two formats: left or right shifted. The ADFM bit (ADCON0<7>) controls the output format. Figure 9-3 shows the output formats.

**FIGURE 9-3: 10-BIT A/D RESULT FORMAT**



### REGISTER 9-1: ADCON0 – A/D CONTROL REGISTER (ADDRESS: 1Fh)

| R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADFM | VCFG | — | — | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON |
| bit 7 | | | | | | | bit 0 |

bit 7 **ADFM:** A/D Result Formed Select bit
1 = Right justified
0 = Left justified

bit 6 **VCFG:** Voltage Reference bit
1 = $V_{REF}$ pin
0 = $V_{DD}$

bit 5-4 **Unimplemented:** Read as ‘0’

bit 3-2 **CHS<1:0>:** Analog Channel Select bits
00 = Channel 00 (AN0)
01 = Channel 01 (AN1)
10 = Channel 02 (AN2)
11 = Channel 03 (AN3)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

bit 0 **ADON:** A/D Conversion Status bit
1 = A/D converter module is operating
0 = A/D converter is shut off and consumes no operating current

**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 |

### REGISTER 9-2: ANSEL – ANALOG SELECT REGISTER (ADDRESS: 9Fh)

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | ADCS2 | ADCS1 | ADCS0 | ANS3 | ANS2 | ANS1 | ANS0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **ADCS<2:0>:** A/D Conversion Clock Select bits
000 = FOSC/2
001 = FOSC/8
010 = FOSC/32
x11 = FRC (clock derived from a dedicated internal oscillator = 500 kHz max)
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64

bit 3-0 **ANS<3:0>**: Analog Select bits
Analog select between analog or digital function on pins ANS<3:0>, respectively.
1 = Analog input. Pin is assigned as analog input[(1)].
0 = Digital I/O. Pin is assigned to port or special function.

**Note 1:** Setting a pin to an analog input automatically disables the digital input circuitry, weak pull-ups and interrupt-on-change if available. The corresponding TRISIO bit must be set to input mode in order to allow external control of the voltage on the pin.

| **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 |

### 9.1.7 CONFIGURING THE A/D

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRISIO bits selected as inputs.

To determine sample time, see **Section 15.0 “Electrical Specifications”**. After this sample time has elapsed, the A/D conversion can be started.

These steps should be followed for an A/D conversion:

1. Configure the A/D module:
   - Configure analog/digital I/O (ANSEL)
   - Configure voltage reference (ADCON0)
   - Select A/D input channel (ADCON0)
   - Select A/D conversion clock (ANSEL)
   - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
   - Clear ADIF bit (PIR1<6>)
   - Set ADIE bit (PIE1<6>)
   - Set PEIE and GIE bits (INTCON<7:6>)
3. Wait the required acquisition time.
4. Start conversion:
   - Set GO/$\overline{\text{DONE}}$ bit (ADCON0<1>)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared (with interrupts disabled); OR
   - Waiting for the A/D interrupt
6. Read A/D Result register pair (ADRESH:ADRESL), clear bit ADIF if required.
7. For next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2 TAD is required before the next acquisition starts.

**EXAMPLE 9-1: A/D CONVERSION**

```
;This code block configures the A/D
;for polling, Vdd reference, R/C clock
;and GP0 input.
;
;Conversion start & wait for complete
;polling code included.
;
 BSF     STATUS,RP0     ;Bank 1
 MOVLW   B’01110001’    ;A/D RC clock
 MOVWF   ANSEL          ;Set GP0 to analog
 BSF     TRISIO,0       ;Set GP0 to input
 BCF     STATUS,RP0     ;Bank 0
 MOVLW   B’10000001’    ;Right, Vdd Vref, AN0
 MOVWF   ADCON0
 CALL    SampleTime     ;Wait min sample time
 BSF     ADCON0,GO      ;Start conversion
 BTFSC   ADCON0,GO      ;Is conversion done?
 GOTO    $-1            ;No, test again
 MOVF    ADRESH,W       ;Read upper 2 bits
 MOVWF   RESULTHI
 BSF     STATUS,RP0     ;Bank 1
 MOVF    ADRESL,W       ;Read lower 8 bits
 MOVWF   RESULTLO
```

## 9.2 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 9-4. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), see Figure 9-4. **The maximum recommended impedance for analog sources is 10 kΩ**. As the impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 9-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

To calculate the minimum acquisition time, TACQ, see the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

**EQUATION 9-1: ACQUISITION TIME**

$$
\begin{aligned}
T_{ACQ} &= \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \\
&= T_{AMP} + T_C = T_{COFF} \\
&= 2\ \mu s + T_C + [(\text{Temperature} - 25^\circ C)(0.05\ \mu s/^\circ C)]
\end{aligned}
$$

$$
\begin{aligned}
T_C &= C_{HOLD}\,(R_{IC} + R_{SS} + R_S)\ In(1/2047) \\
&= -120\ pF\ (1\ k\Omega + 7\ k\Omega + 10\ k\Omega)\ In(0.0004885) \\
&= 16.47\ \mu s
\end{aligned}
$$

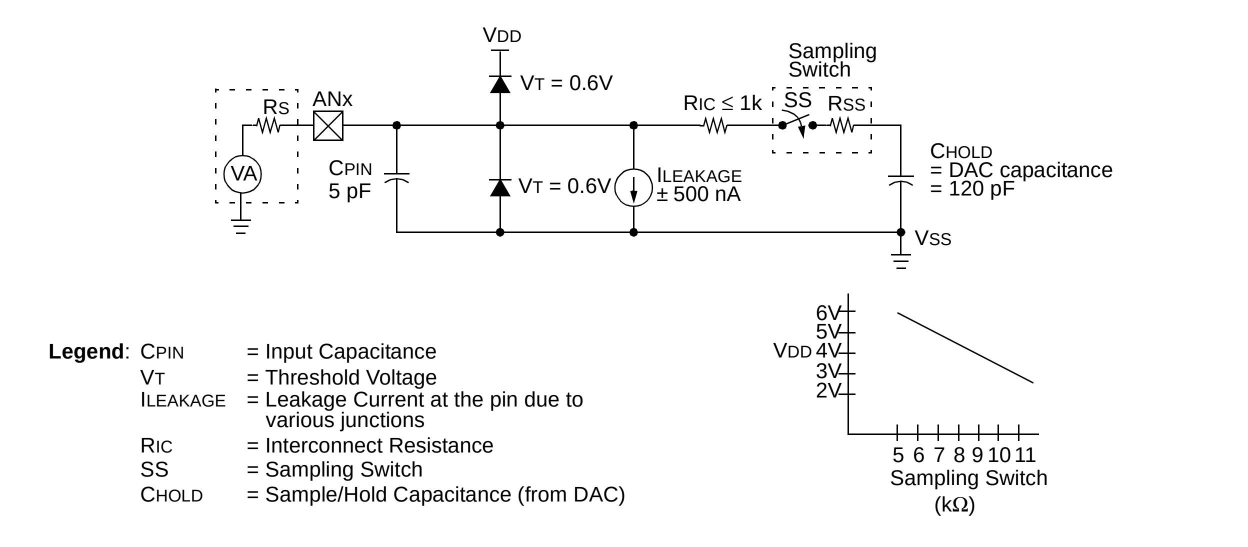
$$
\begin{aligned}
T_{ACQ} &= 2\ \mu s + 16.47\ \mu s + [(50^\circ C - 25^\circ C)(0.05\ \mu s/^\circ C)] \\
&= 19.72\ \mu s
\end{aligned}
$$

**Note 1:** The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

**2:** The charge holding capacitor (CHOLD) is not discharged after each conversion.

**3:** The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

**FIGURE 9-4: ANALOG INPUT MODEL**



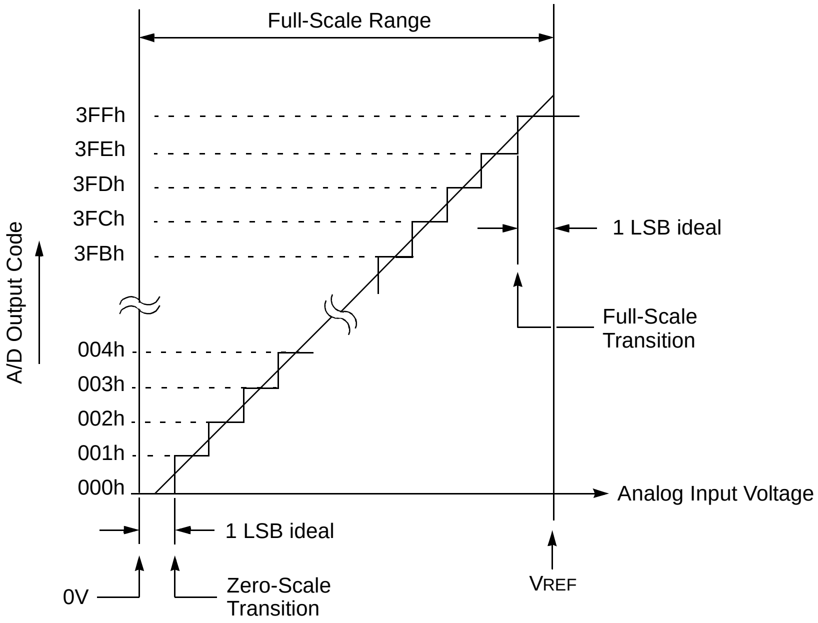
## 9.3 A/D Operation During Sleep

The A/D converter module can operate during Sleep. This requires the A/D clock source to be set to the internal oscillator. When the RC clock source is selected, the A/D waits one instruction before starting the conversion. This allows the SLEEP instruction to be executed, thus eliminating much of the switching noise from the conversion. When the conversion is complete, the GO/$\overline{\text{DONE}}$ bit is cleared and the result is loaded into the ADRESH:ADRESL registers.

If the A/D interrupt is enabled, the device awakens from Sleep. If the GIE bit (INTCON<7>) is set, the program counter is set to the interrupt vector (0004h); if GIE is clear, the next instruction is executed. If the A/D interrupt is not enabled, the A/D module is turned off, although the ADON bit remains set.

When the A/D clock source is something other than RC, a SLEEP instruction causes the present conversion to be aborted and the A/D module is turned off. The ADON bit remains set.

**FIGURE 9-5: PIC12F683 A/D TRANSFER FUNCTION**



## 9.4 Effects of Reset

A device Reset forces all registers to their Reset state. Thus, the A/D module is turned off and any pending conversion is aborted. The ADRESH:ADRESL registers are unchanged.

### TABLE 9-2: SUMMARY OF A/D REGISTERS

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 05h | GPIO | — | — | GP5 | GP4 | GP3 | GP2 | GP1 | GP0 | `--xx xxxx` | `--uu uuuu` |
| 0Bh/<br>8Bh | INTCON | GIE | PEIE | T0IE | INTE | GPIE | T0IF | INTF | GPIF | `0000 0000` | `0000 0000` |
| 0Ch | PIR1 | EEIF | ADIF | CCP1IF | — | CMIF | OSFIF | TMR2IF | TMR1IF | `000- 0000` | `000- 0000` |
| 1Eh | ADRESH | Most Significant 8 bits of the left shifted A/D result or 2 bits of the right shifted result | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| 1Fh | ADCON0 | ADFM | VCFG | — | — | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON | `00-- 0000` | `00-- 0000` |
| 85h | TRISIO | — | — | TRISIO5 | TRISIO4 | TRISIO3 | TRISIO2 | TRISIO1 | TRISIO0 | `--11 1111` | `--11 1111` |
| 8Ch | PIE1 | EEIE | ADIE | CCPIE | — | CMIE | OSFIE | TMR2IE | TMR1IE | `000- 0000` | `000- 0000` |
| 9Eh | ADRESL | Least Significant 2 bits of the left shifted A/D result or 8 bits of the right shifted result | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| 9Fh | ANSEL | — | ADCS2 | ADCS1 | ADCS0 | ANS3 | ANS2 | ANS1 | ANS0 | `-000 1111` | `-000 1111` |

**Legend:** x = unknown, u = unchanged, — = unimplemented read as ‘0’. Shaded cells are not used for A/D module.

NOTES:

## 10.0 DATA EEPROM MEMORY

The EEPROM 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. There are four SFRs used to read and write this memory:

- EECON1
- EECON2 (not a physically implemented register)
- EEDAT
- EEADR

EEDAT holds the 8-bit data for read/write and EEADR holds the address of the EEPROM location being accessed. PIC12F683 has 256 bytes of data EEPROM with an address range from 0h to FFh.

The EEPROM data memory allows byte read and write. A byte write automatically erases the location and writes the new data (erase before write). The EEPROM data memory is rated for high erase/write cycles. The write time is controlled by an on-chip timer. The write time will vary with voltage and temperature as well as from chip-to-chip. Please refer to AC Specifications in **Section 15.0 "Electrical Specifications"** for exact limits.

When the data memory is code protected, the CPU may continue to read and write the data EEPROM memory. The device programmer can no longer access the data EEPROM data and will read zeroes.

Additional information on the data EEPROM is available in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

**REGISTER 10-1: EEDAT – EEPROM DATA REGISTER (ADDRESS: 9Ah)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| EEDAT7 | EEDAT6 | EEDAT5 | EEDAT4 | EEDAT3 | EEDAT2 | EEDAT1 | EEDAT0 |
| bit 7 | | | | | | | bit 0 |

bit 7-0 **EEDATn**: Byte Value to Write to or Read From Data EEPROM bits

| **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 |

**REGISTER 10-2: EEADR – EEPROM ADDRESS REGISTER (ADDRESS: 9Bh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| EEADR7 | EEADR6 | EEADR5 | EEADR4 | EEADR3 | EEADR2 | EEADR1 | EEADR0 |
| bit 7 | | | | | | | bit 0 |

bit 7-0 **EEADR**: Specifies One of 256 Locations for EEPROM Read/Write Operation bits

| **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 |

## 10.1 EECON1 and EECON2 Registers

EECON1 is the control register with four low-order bits physically implemented. The upper four bits are non-implemented and read as '0'.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set in software. They are cleared in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a $\overline{\text{MCLR}}$ Reset, or a WDT Time-out Reset during normal operation.

In these situations, following Reset, the user can check the WRERR bit, clear it and rewrite the location. The data and address will be cleared. Therefore, the EEDAT and EEADR registers will need to be re-initialized.

Interrupt flag, EEIF bit (PIR1<7>), is set when write is complete. This bit must be cleared in software.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the data EEPROM write sequence.

**Note:** The EECON1, EEDAT and EEADR registers should not be modified during a data EEPROM write (WR bit = 1).

**REGISTER 10-3: EECON1 – EEPROM CONTROL REGISTER (ADDRESS: 9Ch)**

| U-0 | U-0 | U-0 | U-0 | R/W-x | R/W-0 | R/S-0 | R/S-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | WRERR | WREN | WR | RD |
| bit 7 | | | | | | | bit 0 |

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **WRERR:** EEPROM Error Flag bit
1 = A write operation is prematurely terminated (any $\overline{\text{MCLR}}$ Reset, any WDT Reset during normal operation or BOD detect)
0 = The write operation completed

bit 2 **WREN:** EEPROM Write Enable bit
1 = Allows write cycles
0 = Inhibits write to the data EEPROM

bit 1 **WR**: Write Control bit
1 = Initiates a write cycle (The bit is cleared by hardware once write is complete. The WR bit can only be set, not cleared, in software.)
0 = Write cycle to the data EEPROM is complete

bit 0 **RD**: Read Control bit
1 = Initiates an EEPROM read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set, not cleared, in software.)
0 = Does not initiate an EEPROM read

| **Legend:** | | | |
|---|---|---|---|
| S = Bit can only be set | | | |
| 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 |

## 10.2 Reading the EEPROM Data Memory

To read a data memory location, the user must write the address to the EEADR register and then set control bit RD (EECON1<0>), as shown in Example 10-1. The data is available, in the very next cycle, in the EEDAT register. Therefore, it can be read in the next instruction. EEDAT holds this value until another read, or until it is written to by the user (during a write operation).

**EXAMPLE 10-1: DATA EEPROM READ**

```
    BSF      STATUS,RP0    ;Bank 1
    MOVLW    CONFIG_ADDR   ;
    MOVWF    EEADR         ;Address to read
    BSF      EECON1,RD     ;EE Read
    MOVF     EEDAT,W       ;Move data to W
```

## 10.3 Writing to the EEPROM Data Memory

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDAT register. Then the user must follow a specific sequence to initiate the write for each byte, as shown in Example 10-2.

**EXAMPLE 10-2: DATA EEPROM WRITE**

```
     BSF      STATUS,RP0   ;Bank 1
     BSF      EECON1,WREN  ;Enable write
     BCF      INTCON,GIE   ;Disable INTs
     MOVLW    55h          ;Unlock write
     MOVWF    EECON2       ;
     MOVLW    AAh          ;
     MOVWF    EECON2       ;
     BSF      EECON1,WR    ;Start the write
     BSF      INTCON,GIE   ;Enable INTS
```

Required Sequence

The write will not initiate if the above sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. We strongly recommend that interrupts be disabled during this code segment. A cycle count is executed during the required sequence. Any number that is not equal to the required cycles to execute the required sequence will prevent the data from being written into the EEPROM.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. The EEIF bit (PIR1<7>) must be cleared by software.

## 10.4 Write Verify

Depending on the application, good programming practice may dictate that the value written to the data EEPROM should be verified (see Example 10-3) to the desired value to be written.

**EXAMPLE 10-3: WRITE VERIFY**

```
    BSF     STATUS,RP0   ;Bank 1
    MOVF    EEDAT,W      ;EEDAT not changed
                         ;from previous write
    BSF     EECON1,RD    ;YES, Read the
                         ;value written
    XORWF    EEDAT,W
    BTFSS    STATUS,Z    ;Is data the same
    GOTO     WRITE_ERR   ;No, handle error
    :                    ;Yes, continue
```

### 10.4.1 USING THE DATA EEPROM

The data EEPROM is a high-endurance, byte addressable array that has been optimized for the storage of frequently changing information. The maximum endurance for any EEPROM cell is specified as Dxxx. D120 or D120A specify a maximum number of writes to any EEPROM location before a refresh is required of infrequently changing memory locations.

#### 10.4.1.1 EEPROM Endurance

A hypothetical data EEPROM is 64 bytes long and has an endurance of 1M writes. It also has a refresh parameter of 10M writes. If every memory location in the cell were written the maximum number of times, the data EEPROM would fail after 64M write cycles. If every memory location, save one, were written the maximum number of times, the data EEPROM would fail after 63M write cycles but the one remaining location could fail after 10M cycles. If proper refreshes occurred, then the lone memory location would have to be refreshed six times for the data to remain correct.

## 10.5 Protection Against Spurious Write

There are conditions when the user may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built in. On power-up, WREN is cleared. Also, the Power-up Timer (64 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during:

- brown-out
- power glitch
- software malfunction

## 10.6 Data EEPROM Operation During Code-Protect

Data memory can be code-protected by programming the CPD bit in the Configuration Word (Register 12-1) to ‘0’.

When the data memory is code-protected, the CPU is able to read and write data to the data EEPROM. It is recommended to code-protect the program memory when code-protecting data memory. This prevents anyone from programming zeroes over the existing code (which will execute as NOPs) to reach an added routine, programmed in unused program memory, which outputs the contents of data memory. Programming unused locations in program memory to ‘0’ will also help prevent data memory code protection from becoming breached.

### TABLE 10-1: REGISTERS/BITS ASSOCIATED WITH DATA EEPROM

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | GPIE | T0IF | INTF | GPIF | 0000 0000 | 0000 0000 |
| 0Ch | PIR1 | EEIF | ADIF | CCP1IF | — | CMIF | OSFIF | TMR2IF | TMR1IF | 000- 0000 | 000- 0000 |
| 8Ch | PIE1 | EEIE | ADIE | CCP1IE | — | CMIE | OSFIE | TMR2IE | TMR1IE | 000- 0000 | 000- 0000 |
| 9Ah | EEDAT | EEDAT7 | EEDAT6 | EEDAT5 | EEDAT4 | EEDAT3 | EEDAT2 | EEDAT1 | EEDAT0 | 0000 0000 | 0000 0000 |
| 9Bh | EEADR | EEADR7 | EEADR6 | EEADR5 | EEADR4 | EEADR3 | EEADR2 | EEADR1 | EEADR0 | 0000 0000 | 0000 0000 |
| 9Ch | EECON1 | — | — | — | — | WRERR | WREN | WR | RD | ---- x000 | ---- q000 |
| 9Dh | EECON2[1] | EEPROM Control Register 2 | | | | | | | | ---- ---- | ---- ---- |

**Legend:** x = unknown, u = unchanged, — = unimplemented read as ‘0’, q = value depends upon condition. Shaded cells are not used by data EEPROM module.

**Note 1:** EECON2 is not a physical register.

## 11.0 CAPTURE/COMPARE/PWM (CCP) MODULE

The Capture/Compare/PWM (CCP) module contains a 16-bit register which can operate as a:

- 16-bit Capture register
- 16-bit Compare register
- PWM Master/Slave Duty Cycle register

Capture/Compare/PWM Register 1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP. The special event trigger is generated by a compare match and will clear both TMR1H and TMR1L registers.

**TABLE 11-1: CCP MODE – TIMER RESOURCES REQUIRED**

| CCP Mode | Timer Resource |
|---|---|
| Capture | Timer1 |
| Compare | Timer1 |
| PWM | Timer2 |

**REGISTER 11-1: CCP1CON – CCP CONTROL REGISTER 1 (ADDRESS: 15h)**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 |
| bit 7 | | | | | | | bit 0 |

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

bit 5-4 **DC1B<1:0>**: PWM Least Significant bits

Capture mode:
Unused.

Compare mode:
Unused.

PWM mode:
These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPR1L.

bit 3-0 **CCP1M<3:0>**: CCP1 Mode Select bits

0000 = Capture/Compare/PWM disabled (resets CCP1 module)
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, set output on match (CCP1IF bit is set)
1001 = Compare mode, clear output on match (CCP1IF bit is set)
1010 = Compare mode, generate software interrupt on match (CCP1IF bit is set, CCP1 pin is unaffected)
1011 = Compare mode, trigger special event (CCP1IF bit is set, CCP1 pin is unaffected); CCP1 resets TMR1 and starts an A/D conversion (if A/D module is enabled)
11xx = PWM mode

| **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 |

## 11.1 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 register when an event occurs on pin GP2/AN2/T0CKI/INT/COUT/CCP1. An event is defined as one of the following and is configured by CCP1CON<3:0>:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

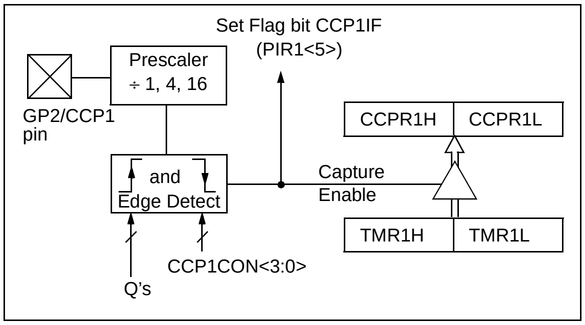
When a capture is made, the interrupt request flag bit, CCP1IF (PIR1<5>), is set. The interrupt flag must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value is overwritten by the new captured value.

### 11.1.1 CCP1 PIN CONFIGURATION

In Capture mode, the GP2/AN2/T0CKI/INT/COUT/CCP1 pin should be configured as an input by setting the TRISIO<2> bit.

**Note:** If the GP2/AN2/T0CKI/INT/COUT/CCP1 pin is configured as an output, a write to the port can cause a capture condition.

**FIGURE 11-1: CAPTURE MODE OPERATION BLOCK DIAGRAM**



### 11.1.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

### 11.1.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep bit CCP1IE (PIE1<5>) clear to avoid false interrupts and should clear the flag bit CCP1IF following any such change in operating mode.

### 11.1.4 CCP PRESCALER

There are four prescaler settings specified by bits CCP1M<3:0> (CCP1CON<3:0>). Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. Any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared; therefore, the first capture may be from a non-zero prescaler. Example 11-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

**EXAMPLE 11-1: CHANGING BETWEEN CAPTURE PRESCALERS**

```
CLRF    CCP1CON     ;Turn CCP module off
MOVLW   NEW_CAPT_PS ;Load the W reg with
                    ;the new prescaler
                    ;move value and CCP ON
MOVWF   CCP1CON     ;Load CCP1CON with this
                    ;value
```

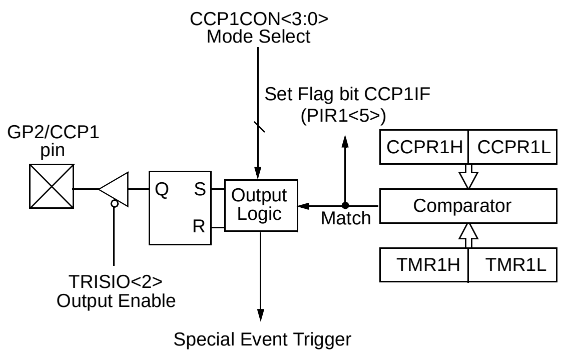
## 11.2 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the GP2/AN2/T0CKI/INT/COUT/CCP1 pin is:

- Driven high
- Driven low
- Remains unchanged

The action on the pin is based on the value of control bits, CCP1M<3:0> (CCP1CON<3:0>). At the same time, interrupt flag bit, CCP1IF (PIR1<5>), is set.

**FIGURE 11-2: COMPARE MODE OPERATION BLOCK DIAGRAM**



Special Event Trigger will:

- Clear TMR1H and TMR1L registers
- NOT set interrupt flag bit TMR1F (PIR1<0>)
- Set the GO/$\overline{\text{DONE}}$ bit (ADCON0<1>)

### 11.2.1 CCP1 PIN CONFIGURATION

The user must configure the GP2/AN2/T0CKI/INT/COUT/CCP1 pin as an output by clearing the TRISIO<2> bit.

> **Note:** Clearing the CCP1CON register will force the GP2/AN2/T0CKI/INT/COUT/CCP1 compare output latch to the default low level. This is not the GPIO data latch.

### 11.2.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

### 11.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen (CCP1M<3:0> = `1010`), the CCP1 pin is not affected. The CCP1IF (PIR1<5>) bit is set, causing a CCP interrupt (if enabled). See Register 11-1.

### 11.2.4 SPECIAL EVENT TRIGGER

In this mode, an internal hardware trigger is generated, which may be used to initiate an action.

The special event trigger output of CCP1 resets the TMR1 register pair and starts A/D conversion, if enabled. This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

> **Note:** The special event trigger from the CCP1 modules will not set interrupt flag bit TMR1IF (PIR1<0>).

**TABLE 11-2: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE AND TIMER1**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh/ 8Bh | INTCON | GIE | PEIE | T0IE | INTE | GPIE | T0IF | INTF | GPIF | `0000 0000` | `0000 0000` |
| 0Ch | PIR1 | EEIF | ADIF | CCP1IF | — | CMIF | OSFIF | TMR2IF | TMR1IF | `000- 0000` | `000- 0000` |
| 0Eh | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| 0Fh | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| 10h | T1CON | T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | `0000 0000` | `uuuu uuuu` |
| 1Ah | CMCON1 | — | — | — | — | — | — | T1GSS | CMSYNC | `---- --10` | `---- --10` |
| 13h | CCPR1L | Capture/Compare/PWM Register 1 Low Byte | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| 14h | CCPR1H | Capture/Compare/PWM Register 1 High Byte | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| 15h | CCP1CON | — | — | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | `0000 0000` | `0000 0000` |
| 8Ch | PIE1 | EEIE | ADIE | CCP1IE | — | CMIE | OSFIE | TMR2IE | TMR1IE | `000- 0000` | `000- 0000` |

**Legend:** — = unimplemented locations, read as '`0`', u = unchanged, x = unknown.
Shaded cells are not used by the Capture, Compare or Timer1 module.

## 11.3 PWM Mode (PWM)

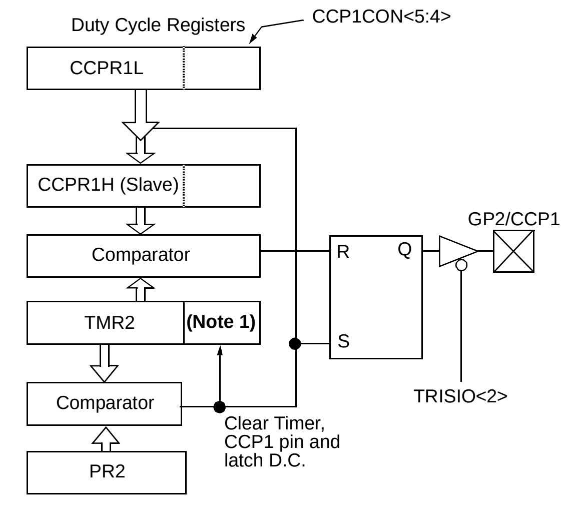
In Pulse Width Modulation mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the GPIO data latch, the TRISIO<2> bit must be cleared to make the CCP1 pin an output.

| **Note:** | Clearing the CCP1CON register will force the CCP1 PWM output latch to the default low level. This is not the GPIO data latch. |
|---|---|

Figure 11-3 shows a simplified block diagram of the CCP module in PWM mode.

For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 11.3.3 "Setup for PWM Operation"**.

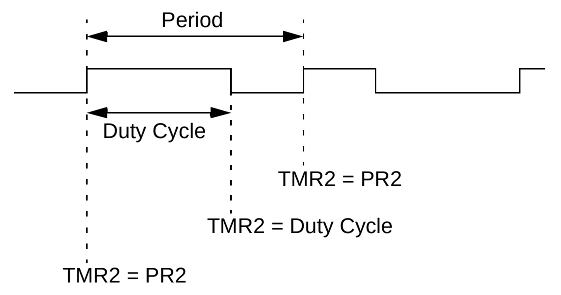
**FIGURE 11-3: SIMPLIFIED PWM BLOCK DIAGRAM**



**Note 1:** The 8-bit timer is concatenated with 2-bit internal Q clock, or 2 bits of the prescaler, to create 10-bit time base.

A PWM output (Figure 11-4) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

**FIGURE 11-4: PWM OUTPUT**



### 11.3.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula.

**EQUATION 11-1:**

$$PWM\ Period = [(PR2) + 1] \cdot 4 \cdot Tosc \cdot TMR2\ Prescale\ Value$$

PWM frequency is defined as 1/[PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared.
- The CCP1 pin is set (Exception: If PWM duty cycle = 0%, the CCP1 pin will not be set).
- The PWM duty cycle is latched from CCPR1L into CCPR1H.

| **Note:** | The Timer2 postscaler (see **Section 7.1 "Timer2 Operation"**) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output. |
|---|---|

### 11.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available. The CCPR1L contains the eight MSbs and the CCP1CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The following equation is used to calculate the PWM duty cycle in time.

**EQUATION 11-2:**

$$PWM\ Duty\ Cycle = (CCPR1L:CCP1CON<5:4> \bullet T_{OSC} \bullet TMR2\ Prescale\ Value$$

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR1H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read-only register.

The CCPR1H register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitch-free PWM operation.

When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.

The maximum PWM resolution (bits) for a given PWM frequency is given by the following formula.

**EQUATION 11-3:**

$$Resolution = \frac{\log\left(\frac{F_{OSC}}{F_{PWM} \bullet TMR2\ Prescale\ Value}\right)}{\log(2)}\ bits$$

**Note:** If the PWM duty cycle value is longer than the PWM period, the CCP1 pin will not be cleared.

### 11.3.3 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP1 module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
3. Make the CCP1 pin an output by clearing the TRISIO<2> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

**Note:** The PWM module may generate a premature pulse when changing the duty cycle. For sensitive applications, disable the PWM module prior to modifying the duty cycle.

**TABLE 11-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz**

| PWM Frequency | 1.22 kHz | 4.88 kHz | 19.53 kHz | 78.12 kHz | 156.3 kHz | 208.3 kHz |
|---|---|---|---|---|---|---|
| Timer Prescaler (1, 4, 16) | 16 | 4 | 1 | 1 | 1 | 1 |
| PR2 Value | 0xFFh | 0xFFh | 0xFFh | 0x3Fh | 0x1Fh | 0x17h |
| Maximum Resolution (bits) | 10 | 10 | 10 | 8 | 7 | 6.6 |

### TABLE 11-4: REGISTERS ASSOCIATED WITH PWM AND TIMER2

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh/<br>8Bh | INTCON | GIE | PEIE | T0IE | INTE | GPIE | T0IF | INTF | GPIF | `0000 0000` | `0000 0000` |
| 0Ch | PIR1 | EEIF | ADIF | CCP1IF | — | CMIF | OSFIF | TMR2IF | TMR1IF | `000- 0000` | `000- 0000` |
| 11h | TMR2 | Timer2 Module Register | | | | | | | | `0000 0000` | `0000 0000` |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | `-000 0000` | `-000 0000` |
| 13h | CCPR1L | Capture/Compare/PWM Register 1 Low Byte | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| 14h | CCPR1H | Capture/Compare/PWM Register 1 High Byte | | | | | | | | `xxxx xxxx` | `uuuu uuuu` |
| 15h | CCP1CON | — | — | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | `--00 0000` | `--00 0000` |
| 8Ch | PIE1 | EEIE | ADIE | CCP1IE | — | CMIE | OSFIE | TMR2IE | TMR1IE | `000- 0000` | `000- 0000` |
| 92h | PR2 | Timer2 Module Period Register | | | | | | | | `1111 1111` | `1111 1111` |

**Legend:** — = unimplemented locations, read as ‘0’, u = unchanged, x = unknown.
Shaded cells are not used by the PWM or Timer2 module.

## 12.0 SPECIAL FEATURES OF THE CPU

The PIC12F683 has a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving features and offer code protection.

These features are:

- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Detect (BOD)
- Interrupts
- Watchdog Timer (WDT)
- Oscillator Selection
- Sleep
- Code Protection
- ID Locations
- In-Circuit Serial Programming

The PIC12F683 has two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 64 ms (nominal) on power-up only, designed to keep the part in Reset while the power supply stabilizes. There is also circuitry to reset the device if a brown-out occurs, which can use the Power-up Timer to provide at least a 64 ms Reset. With these three functions on-chip, most applications need no external Reset circuitry.

The Sleep mode is designed to offer a very low-current Power-down mode. The user can wake-up from Sleep through:

- External Reset
- Watchdog Timer Wake-up
- An interrupt

Several oscillator options are also made available to allow the part to fit the application. The INTOSC option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select various options (see Register 12-1).

## 12.1 Configuration Bits

The configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1') to select various device configurations as shown in Register 12-1. These bits are mapped in program memory location 2007h.

**Note:** Address 2007h is beyond the user program memory space. It belongs to the special configuration memory space (2000h-3FFFh), which can be accessed only during programming. See "*PIC12F6XX/16F6XX Memory Programming Specification*" (DS41204) for more information.

### REGISTER 12-1: CONFIG – CONFIGURATION WORD (ADDRESS: 2007h)

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| — | — | FCMEN | IESO | BODEN1 | BODEN0 | $\overline{\text{CPD}}$ | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | FOSC2 | FOSC1 | FOSC0 |

bit 13 bit 0

bit 13-12 **Unimplemented**: Read as '1'

bit 11 **FCMEN:** Fail-Safe Clock Monitor Enabled bit
1 = Fail-Safe Clock Monitor is enabled
0 = Fail-Safe Clock Monitor is disabled

bit 10 **IESO:** Internal External Switchover bit
1 = Internal External Switchover mode is enabled
0 = Internal External Switchover mode is disabled

bit 9-8 **BODEN<1:0>:** Brown-out Detect Selection bits(1)
11 = BOD enabled
10 = BOD enabled during operation and disabled in Sleep
01 = BOD controlled by SBODEN bit (PCON<4>)
00 = BOD disabled

bit 7 **$\overline{\text{CPD}}$:** Data Code Protection bit(2)
1 = Data memory code protection is disabled
0 = Data memory code protection is enabled

bit 6 **$\overline{\text{CP}}$:** Code Protection bit(3)
1 = Program memory code protection is disabled
0 = Program memory code protection is enabled

bit 5 **MCLRE:** GP3/$\overline{\text{MCLR}}$ pin function select bit(4)
1 = GP3/$\overline{\text{MCLR}}$ pin function is $\overline{\text{MCLR}}$
0 = GP3/$\overline{\text{MCLR}}$ pin function is digital input, $\overline{\text{MCLR}}$ internally tied to $V_{DD}$

bit 4 **$\overline{\text{PWRTE}}$:** Power-up Timer Enable bit
1 = PWRT disabled
0 = PWRT enabled

bit 3 **WDTE:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled and can be enabled by SWDTEN bit (WDTCON<0>)

bit 2-0 **FOSC<2:0>:** Oscillator Selection bits
111 = RC oscillator: CLKOUT function on RA4/OSC2/CLKOUT pin, RC on RA5/OSC1/CLKIN
110 = RCIO oscillator: I/O function on RA4/OSC2/CLKOUT pin, RC on RA5/OSC1/CLKIN
101 = INTOSC oscillator: CLKOUT function on RA4/OSC2/CLKOUT pin, I/O function on RA5/OSC1/CLKIN
100 = INTOSCIO oscillator: I/O function on RA4/OSC2/CLKOUT pin, I/O function on RA5/OSC1/CLKIN
011 = EC: I/O function on RA4/OSC2/CLKOUT pin, CLKIN on RA5/OSC1/CLKIN
010 = HS oscillator: High-speed crystal/resonator on RA4/OSC2/CLKOUT and RA5/OSC1/CLKIN
001 = XT oscillator: Crystal/resonator on RA4/OSC2/CLKOUT and RA5/OSC1/CLKIN
000 = LP oscillator: Low-power crystal on RA4/OSC2/CLKOUT and RA5/OSC1/CLKIN

**Note 1:** Enabling Brown-out Detect does not automatically enable Power-up Timer.

**2:** The entire data EEPROM will be erased when the code protection is turned off.

**3:** The entire program memory will be erased when the code protection is turned off. When $\overline{\text{MCLR}}$ is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.

**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 |

## 12.2 Calibration Bits

The Brown-out Detect (BOD), Power-on Reset (POR) and 8 MHz internal oscillator (HFINTOSC) are factory calibrated. These calibration values are stored in the Calibration Word register, as shown in Register 12-2 and are mapped in program memory location 2008h.

The Calibration Word register is not erased when the device is erased when using the procedure described in the "*PIC12F6XX/16F6XX Memory Programming Specification"* (DS41204). Therefore, it is not necessary to store and reprogram these values when the device is erased.

| **Note:** | Address 2008h is beyond the user program memory space. It belongs to the special configuration memory space (2000h-3FFFh), which can be accessed only during programming. See "*PIC12F6XX/16F6XX Memory Programming Specification"* (DS41204) for more information. |
|---|---|

**REGISTER 12-2: CALIB – CALIBRATION WORD (ADDRESS: 2008h)**

| — | FCAL6 | FCAL5 | FCAL4 | FCAL3 | FCAL2 | FCAL1 | FCAL0 | — | POR1 | POR0 | BOD2 | BOD1 | BOD0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| bit 13 | | | | | | | | | | | | | bit 0 |

| | |
|---|---|
| bit 13 | **Unimplemented:** Read as '0' |
| bit 12-6 | **FCAL<6:0>:** Internal Oscillator Calibration bits<br>0111111 = Maximum frequency<br>.<br>.<br>0000001<br>0000000 = Center frequency<br>1111111<br>.<br>.<br>1000000 = Minimum frequency |
| bit 5 | **Unimplemented:** Read as '0' |
| bit 4-3 | **POR<1:0>:** POR Calibration bits<br>00 = Lowest POR voltage<br>11 = Highest POR voltage |
| bit 2-0 | **BOD<2:0>:** BOD Calibration bits<br>000 = Reserved<br>001 = Lowest BOD voltage<br>111 = Highest BOD voltage |

| **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 |

## 12.3 Reset

The PIC12F683 differentiates between various kinds of Reset:

a) Power-on Reset (POR)
b) WDT Reset during normal operation
c) WDT Reset during Sleep
d) $\overline{\text{MCLR}}$ Reset during normal operation
e) $\overline{\text{MCLR}}$ Reset during Sleep
f) Brown-out Detect (BOD)

Some registers are not affected in any Reset condition; their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a “Reset state” on:

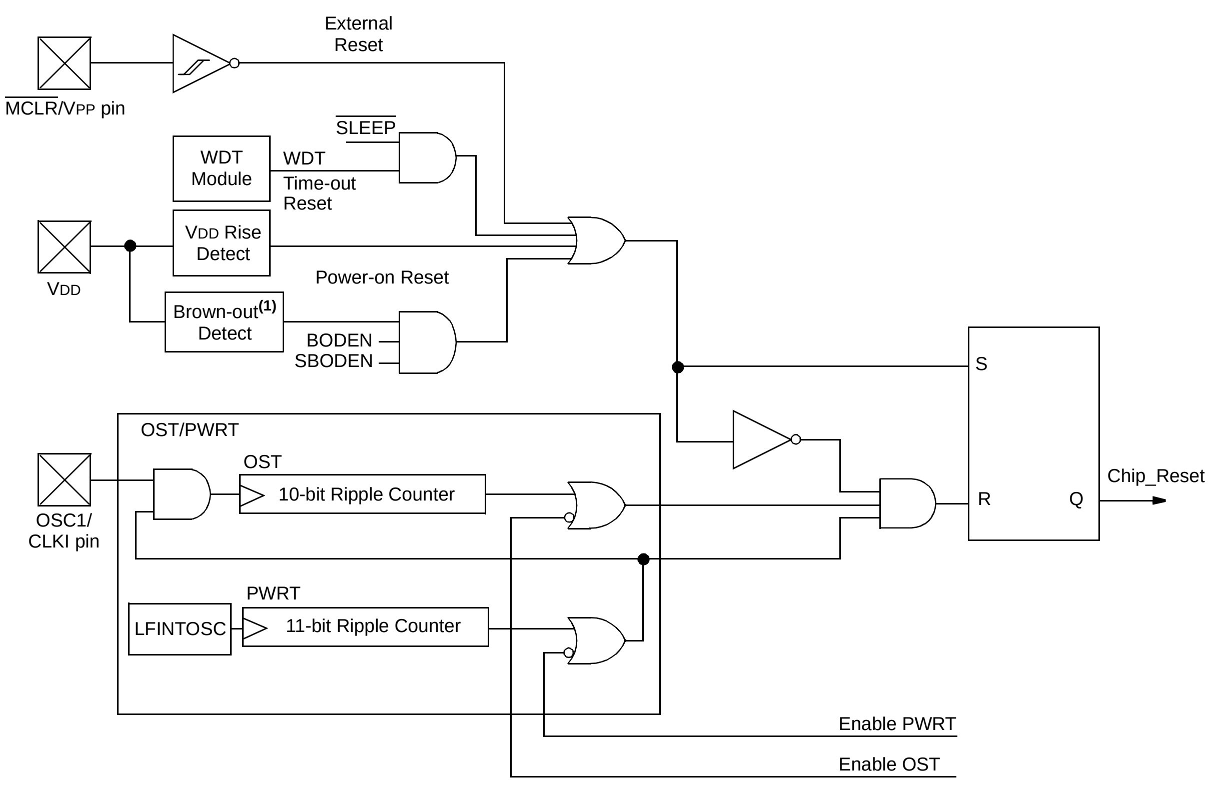
- Power-on Reset
- $\overline{\text{MCLR}}$ Reset
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Reset
- Brown-out Detect (BOD)

They are not affected by a WDT wake-up since this is viewed as the resumption of normal operation. $\overline{\text{TO}}$ and PD bits are set or cleared differently in different Reset situations, as indicated in Table 12-2. These bits are used in software to determine the nature of the Reset. See Table 12-4 for a full description of Reset states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 12-1.

The $\overline{\text{MCLR}}$ Reset path has a noise filter to detect and ignore small pulses. See **Section 15.0 “Electrical Specifications”** for pulse width specifications.

**FIGURE 12-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**



**Note 1:** Refer to the Configuration Word register (Register 12-1).

### 12.3.1 POWER-ON RESET

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 POR, simply connect the MCLR pin through a resistor to VDD. This will eliminate external RC components usually needed to create Power-on Reset. A maximum rise time for VDD is required. See **Section 15.0 "Electrical Specifications"** for details. If the BOD is enabled, the maximum rise time specification does not apply. The BOD circuitry will keep the device in Reset until VDD reaches VBOD (see **Section 12.3.4 "Brown-out Detect (BOD)"**).

> **Note:** The POR circuit does not produce an internal Reset when VDD declines. To re-enable the POR, VDD must reach Vss for a minimum of 100 μs.

When the device starts normal operation (exits the Reset condition), device operating parameters (i.e., 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 the Application Note *AN607, "Power-up Trouble Shooting"* (DS00607).

### 12.3.2 $\overline{\text{MCLR}}$

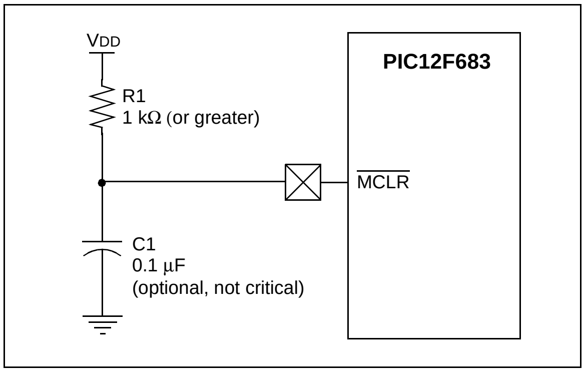
PIC12F683 has a noise filter in the MCLR Reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive MCLR pin low.

The behavior of the ESD protection on the MCLR pin has been altered from early devices of this family. Voltages applied to the pin that exceed its specification can result in both MCLR Resets and excessive current beyond the device specification during the ESD event. For this reason, Microchip recommends that the MCLR pin no longer be tied directly to VDD. The use of an RC network, as shown in Figure 12-2, is suggested.

An internal MCLR option is enabled by clearing the MCLRE bit in the Configuration Word register. When cleared, MCLR is internally tied to VDD and an internal weak pull-up is enabled for the MCLR pin. In-Circuit Serial Programming is not affected by selecting the internal MCLR option.

**FIGURE 12-2: RECOMMENDED $\overline{\text{MCLR}}$ CIRCUIT**



### 12.3.3 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 64 ms (nominal) time-out on power-up only, from POR or Brown-out Detect. The Power-up Timer operates from the 31 kHz LFINTOSC oscillator. For more information, see **Section 3.4 "Internal Clock Modes"**. The chip is kept in Reset as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A configuration bit, PWRTE, can disable (if set) or enable (if cleared or programmed) the Power-up Timer. The Power-up Timer should be enabled when Brown-out Detect is enabled, although it is not required.

The Power-up Timer delay will vary from chip-to-chip and vary due to:

- VDD variation
- Temperature variation
- Process variation

See DC parameters for details (**Section 15.0 "Electrical Specifications"**).

### 12.3.4 BROWN-OUT DETECT (BOD)

The BODEN0 and BODEN1 bits in the Configuration Word register select one of four BOD modes. Two modes have been added to allow software or hardware control of the BOD enable. When BODEN<1:0> = `01`, the SBODEN bit (PCON<4>) enables/disables the BOD allowing it to be controlled in software. By selecting BODEN<1:0>, the BOD is automatically disabled in Sleep to conserve power and enabled on wake-up. In this mode, the SBODEN bit is disabled. See Register 12-1 for the Configuration Word register definition.

If VDD falls below VBOD for greater than parameter TBOD (see **Section 15.0 "Electrical Specifications"**), the Brown-out situation will reset the device. This will occur regardless of VDD slew rate. A Reset is not ensured to occur if VDD falls below VBOD for less than parameter (TBOD).

On any Reset (Power-on, Brown-out Detect, Watchdog Timer, etc.), the chip will remain in Reset until VDD rises above VBOD (see Figure 12-3). The Power-up Timer will now be invoked, if enabled and will keep the chip in Reset an additional 64 ms.

**Note:** The Power-up Timer is enabled by the $\overline{\text{PWRTE}}$ bit in the Configuration Word register.

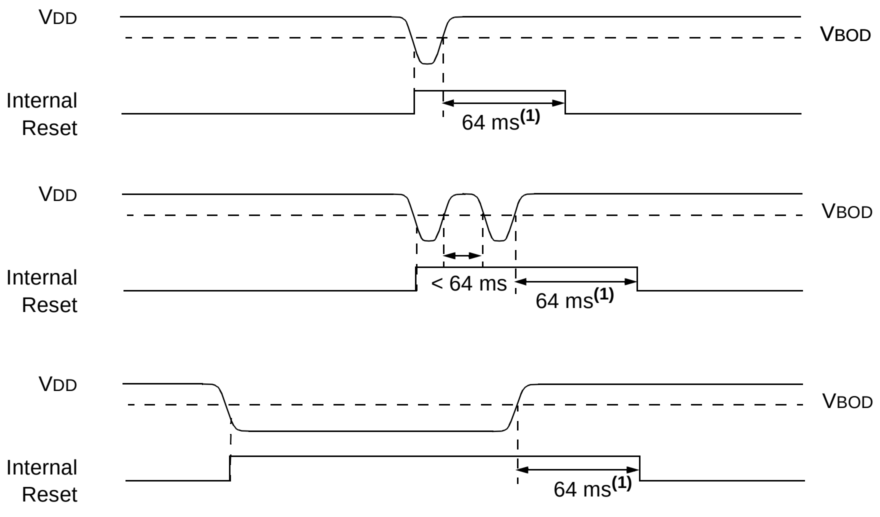
If VDD drops below VBOD while the Power-up Timer is running, the chip will go back into a Brown-out Detect and the Power-up Timer will be re-initialized. Once VDD rises above VBOD, the Power-up Timer will execute a 64 ms Reset.

#### 12.3.4.1 BOD Calibration

The PIC12F683 stores the BOD calibration values in fuses located in the Calibration Word register (2008h). The Calibration Word register is not erased when using the specified bulk erase sequence in the *"PIC12F6XX/16F6XX Memory Programming Specification"* (DS41204) and thus, does not require reprogramming.

**Note:** Address 2008h is beyond the user program memory space. It belongs to the special configuration memory space (2000h-3FFFh), which can be accessed only during programming. See *"PIC12F6XX/16F6XX Memory Programming Specification"* (DS41204) for more information.

**FIGURE 12-3: BROWN-OUT SITUATIONS**



**Note 1:** 64 ms delay only if $\overline{\text{PWRTE}}$ bit is programmed to '`0`'.

### 12.3.5 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows: first, PWRT time-out is invoked after POR has expired, then OST is activated after the PWRT time-out has expired. The total time-out will vary based on oscillator configuration and $\overline{\text{PWRTE}}$ bit status. For example, in EC mode with $\overline{\text{PWRTE}}$ bit erased (PWRT disabled), there will be no time-out at all. Figure 12-4, Figure 12-5 and Figure 12-6 depict time-out sequences. The device can execute code from the INTOSC while OST is active by enabling Two-Speed Start-up or Fail-Safe Monitor (see **Section 3.6 "Two-Speed Clock Start-up Mode"** and **Section 3.7 "Fail-Safe Clock Monitor"**).

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Then, bringing $\overline{\text{MCLR}}$ high will begin execution immediately (see Figure 12-5). This is useful for testing purposes or to synchronize more than one PIC12F683 device operating in parallel.

Table 12-5 shows the Reset conditions for some special registers, while Table 12-4 shows the Reset conditions for all the registers.

### 12.3.6 POWER CONTROL (PCON) REGISTER

The Power Control register PCON (address 8Eh) has two status bits to indicate what type of Reset that last occurred.

Bit 0 is $\overline{\text{BOD}}$ (Brown-out). $\overline{\text{BOD}}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if $\overline{\text{BOD}}$ = `0`, indicating that a Brown-out has occurred. The $\overline{\text{BOD}}$ status bit is a "don't care" and is not necessarily predictable if the brown-out circuit is disabled (BODEN<1:0> = `00` in the Configuration Word register).

Bit 1 is $\overline{\text{POR}}$ (Power-on Reset). It is a '`0`' on Power-on Reset and unaffected otherwise. The user must write a '`1`' to this bit following a Power-on Reset. On a subsequent Reset, if $\overline{\text{POR}}$ is '`0`', it will indicate that a Power-on Reset has occurred (i.e., VDD may have gone too low).

For more information, see **Section 4.2.3 "Ultra Low-Power Wake-up"** and **Section 12.3.4 "Brown-out Detect (BOD)"**.

**TABLE 12-1: TIME-OUT IN VARIOUS SITUATIONS**

| Oscillator Configuration | Power-up | | Brown-out Detect | | Wake-up from Sleep |
|---|---|---|---|---|---|
| | $\overline{\text{PWRTE}}$ = `0` | $\overline{\text{PWRTE}}$ = `1` | $\overline{\text{PWRTE}}$ = `0` | $\overline{\text{PWRTE}}$ = `1` | |
| XT, HS, LP | TPWRT + 1024 • TOSC | 1024 • TOSC | TPWRT + 1024 • TOSC | 1024 • TOSC | 1024 • TOSC |
| RC, EC, INTOSC | TPWRT | — | TPWRT | — | — |

**TABLE 12-2: PCON BITS AND THEIR SIGNIFICANCE**

| $\overline{\text{POR}}$ | $\overline{\text{BOD}}$ | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Condition |
|---|---|---|---|---|
| 0 | u | 1 | 1 | Power-on Reset |
| 1 | 0 | 1 | 1 | Brown-out Detect |
| u | u | 0 | u | WDT Reset |
| u | u | 0 | 0 | WDT Wake-up |
| u | u | u | u | $\overline{\text{MCLR}}$ Reset during normal operation |
| u | u | 1 | 0 | $\overline{\text{MCLR}}$ Reset during Sleep |

**Legend:** u = unchanged, x = unknown

**TABLE 12-3: SUMMARY OF REGISTERS ASSOCIATED WITH BROWN-OUT**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets[1] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 03h | STATUS | IRP | RP1 | RPO | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Z | DC | C | `0001 1xxx` | `000q quuu` |
| 8Eh | PCON | — | — | ULPWUE | SBODEN | — | — | $\overline{\text{POR}}$ | $\overline{\text{BOD}}$ | `--01 --qq` | `--0u --uu` |

**Legend:** u = unchanged, x = unknown, — = unimplemented bit, reads as '`0`', q = value depends on condition. Shaded cells are not used by BOD.

**Note 1:** Other (non Power-up) Resets include $\overline{\text{MCLR}}$ Reset and Watchdog Timer Reset during normal operation.

**FIGURE 12-4: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{\text{MCLR}}$)**

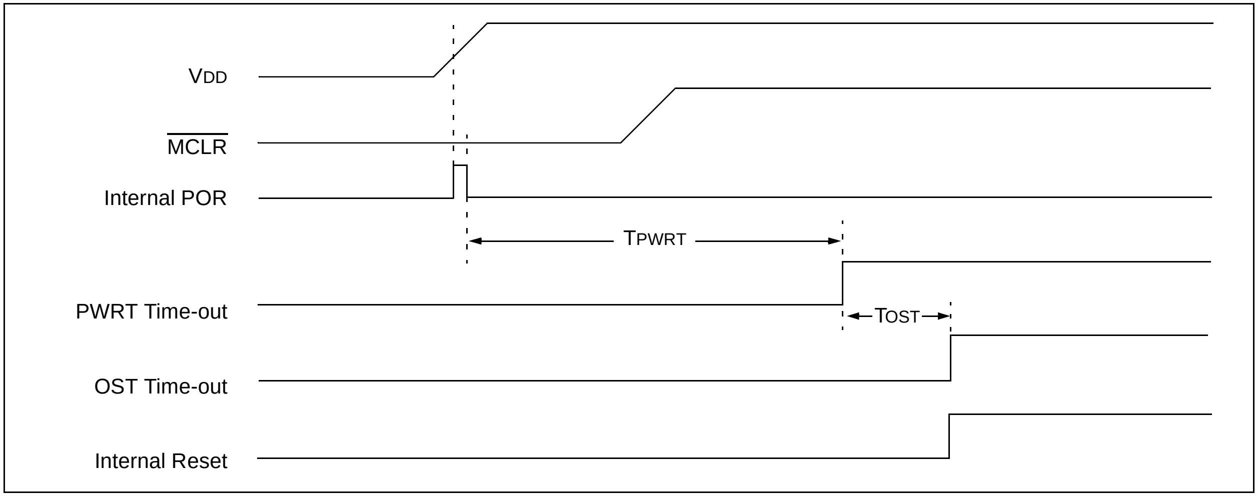


**FIGURE 12-5: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{\text{MCLR}}$)**

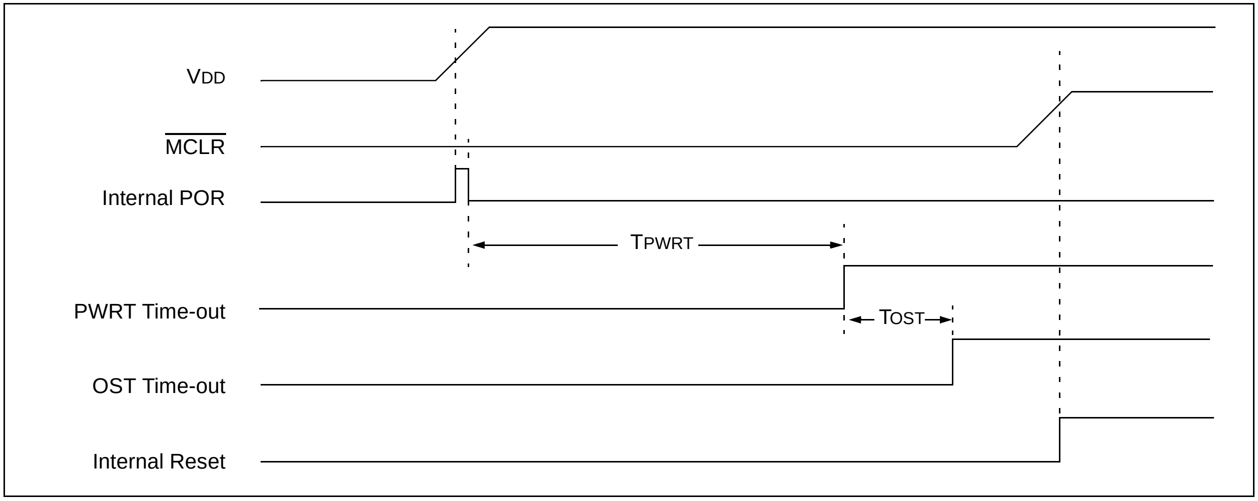


**FIGURE 12-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ WITH VDD)**

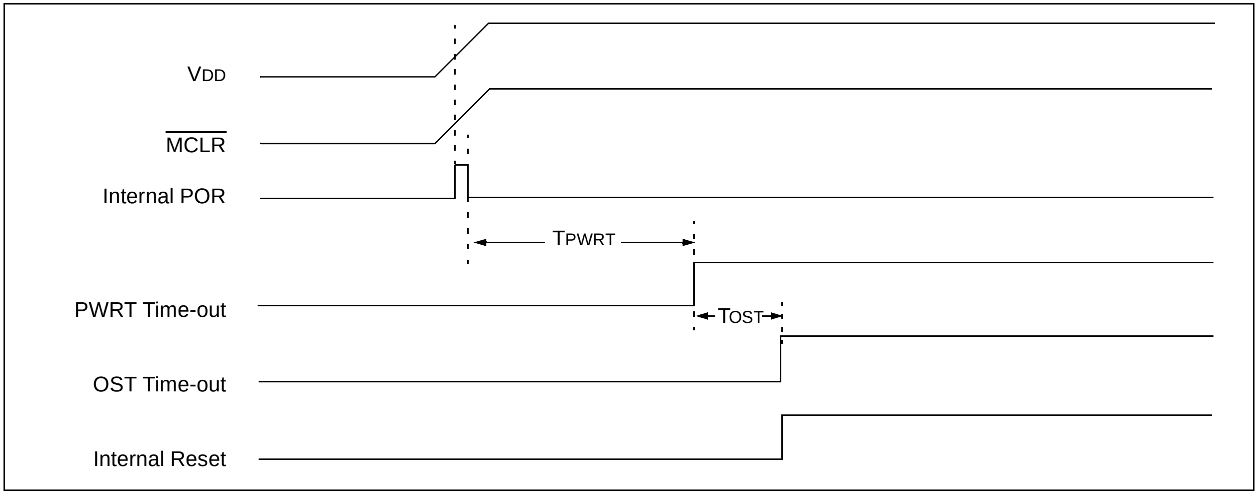


**TABLE 12-4: INITIALIZATION CONDITION FOR REGISTERS**

| Register | Address | Power-on Reset | $\overline{\text{MCLR}}$ Reset<br>WDT Reset<br>Brown-out Detect(1) | Wake-up from Sleep through Interrupt<br>Wake-up from Sleep through WDT Time-out |
|---|---|---|---|---|
| W | — | `xxxx xxxx` | `uuuu uuuu` | `uuuu uuuu` |
| INDF | 00h/80h | `xxxx xxxx` | `xxxx xxxx` | `uuuu uuuu` |
| TMR0 | 01h | `xxxx xxxx` | `uuuu uuuu` | `uuuu uuuu` |
| PCL | 02h/82h | `0000 0000` | `0000 0000` | PC + 1(3) |
| STATUS | 03h/83h | `0001 1xxx` | `000q quuu`(4) | `uuuq quuu`(4) |
| FSR | 04h/84h | `xxxx xxxx` | `uuuu uuuu` | `uuuu uuuu` |
| GPIO | 05h | `--xx xx00` | `--00 0000` | `--uu uuuu` |
| PCLATH | 0Ah/8Ah | `---0 0000` | `---0 0000` | `---u uuuu` |
| INTCON | 0Bh/8Bh | `0000 0000` | `0000 0000` | `uuuu uuuu`(2) |
| PIR1 | 0Ch | `0000 0000` | `0000 0000` | `uuuu uuuu`(2) |
| TMR1L | 0Eh | `xxxx xxxx` | `uuuu uuuu` | `uuuu uuuu` |
| TMR1H | 0Fh | `xxxx xxxx` | `uuuu uuuu` | `uuuu uuuu` |
| T1CON | 10h | `0000 0000` | `uuuu uuuu` | `-uuu uuuu` |
| TMR2 | 11h | `0000 0000` | `0000 0000` | `uuuu uuuu` |
| T2CON | 12h | `-000 0000` | `-000 0000` | `-uuu uuuu` |
| CCPR1L | 13h | `xxxx xxxx` | `uuuu uuuu` | `uuuu uuuu` |
| CCPR1H | 14h | `xxxx xxxx` | `uuuu uuuu` | `uuuu uuuu` |
| CCP1CON | 15h | `0000 0000` | `0000 0000` | `uuuu uuuu` |
| WDTCON | 18h | `---0 1000` | `---0 1000` | `---u uuuu` |
| CMCON0 | 19h | `0000 0000` | `0000 0000` | `uuuu uuuu` |
| CMCON1 | 20h | `---- --10` | `---- --10` | `---- --uu` |
| ADRESH | 1Eh | `xxxx xxxx` | `uuuu uuuu` | `uuuu uuuu` |
| ADCON0 | 1Fh | `00-- 0000` | `00-- 0000` | `uu-- uuuu` |
| OPTION_REG | 81h | `1111 1111` | `1111 1111` | `uuuu uuuu` |
| TRISIO | 85h | `--11 1111` | `--11 1111` | `--uu uuuu` |
| PIE1 | 8Ch | `0000 0000` | `0000 0000` | `uuuu uuuu` |
| PCON | 8Eh | `--01 --0x` | `--0u --uu`(1,5) | `--uu --uu` |
| OSCCON | 8Fh | `-110 x000` | `-110 x000` | `-uuu uuuu` |
| OSCTUNE | 90h | `---0 0000` | `---u uuuu` | `---u uuuu` |
| PR2 | 92h | `1111 1111` | `1111 1111` | `1111 1111` |
| WPU | 95h | `--11 -111` | `--11 -111` | `uuuu uuuu` |
| IOC | 96h | `--00 0000` | `--00 0000` | `--uu uuuu` |
| VRCON | 99h | `0-0- 0000` | `0-0- 0000` | `u-u- uuuu` |
| EEDAT | 9Ah | `0000 0000` | `0000 0000` | `uuuu uuuu` |
| EEADR | 9Bh | `0000 0000` | `0000 0000` | `uuuu uuuu` |

**Legend:** u = unchanged, x = unknown, — = unimplemented bit, reads as '`0`', q = value depends on condition.

**Note 1:** If $V_{DD}$ goes too low, Power-on Reset will be activated and registers will be affected differently.

**2:** One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

**3:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**4:** See Table 12-5 for Reset value for specific condition.

**5:** If Reset was due to brown-out, then bit 0 = `0`. All other Resets will cause bit 0 = u.

### TABLE 12-4: INITIALIZATION CONDITION FOR REGISTERS (CONTINUED)

| Register | Address | Power-on Reset | $\overline{\text{MCLR}}$ Reset<br>WDT Reset<br>Brown-out Detect[1] | Wake-up from Sleep through Interrupt<br>Wake-up from Sleep through WDT Time-out |
|---|---|---|---|---|
| EECON1 | 9Ch | `---- x000` | `---- q000` | `---- uuuu` |
| EECON2 | 9Dh | `---- ----` | `---- ----` | `---- ----` |
| ADRESL | 9Eh | `xxxx xxxx` | `uuuu uuuu` | `uuuu uuuu` |
| ANSEL | 9Fh | `1111 1111` | `1111 1111` | `uuuu uuuu` |

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

**Note 1:** If $V_{DD}$ goes too low, Power-on Reset will be activated and registers will be affected differently.

**2:** One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

**3:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**4:** See Table 12-5 for Reset value for specific condition.

**5:** If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

### TABLE 12-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS

| Condition | Program Counter | Status Register | PCON Register |
|---|---|---|---|
| Power-on Reset | 000h | `0001 1xxx` | `--01 --0x` |
| $\overline{\text{MCLR}}$ Reset during Normal Operation | 000h | `000u uuuu` | `--0u --uu` |
| $\overline{\text{MCLR}}$ Reset during Sleep | 000h | `0001 0uuu` | `--0u --uu` |
| WDT Reset | 000h | `0000 uuuu` | `--0u --uu` |
| WDT Wake-up | PC + 1 | `uuu0 0uuu` | `--uu --uu` |
| Brown-out Detect | 000h | `0001 1uuu` | `--01 --10` |
| Interrupt Wake-up from Sleep | PC + 1[1] | `uuu1 0uuu` | `--uu --uu` |

**Legend:** u = unchanged, x = unknown, — = unimplemented bit, reads as ‘0’.

**Note 1:** When the wake-up is due to an interrupt and Global Interrupt Enable bit, GIE, is set, the PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

## 12.4 Interrupts

The PIC12F683 has 11 sources of interrupt:

- External Interrupt GP2/INT
- TMR0 Overflow Interrupt
- GPIO Change Interrupts
- 2 Comparator Interrupts
- A/D Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt
- EEPROM Data Write Interrupt
- Fail-Safe Clock Monitor Interrupt
- CCP Interrupt

The Interrupt Control register (INTCON) and Peripheral Interrupt Request Register 1 (PIR1) record individual interrupt requests in flag bits. The INTCON register also has individual and global interrupt enable bits.

A Global Interrupt Enable bit, GIE (INTCON<7>), enables (if set) all unmasked interrupts, or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in the INTCON register and PIE1 register. GIE is cleared on Reset.

The Return from Interrupt instruction, `RETFIE`, exits the interrupt routine, as well as sets the GIE bit, which re-enables unmasked interrupts.

The following interrupt flags are contained in the INTCON register:

- INT Pin Interrupt
- GPIO Change Interrupt
- TMR0 Overflow Interrupt

The peripheral interrupt flags are contained in the special register, PIR1. The corresponding interrupt enable bit is contained in special register, PIE1.

The following interrupt flags are contained in the PIR1 register:

- EEPROM Data Write Interrupt
- A/D Interrupt
- 2 Comparator Interrupts
- Timer1 Overflow Interrupt
- Timer 2 Match Interrupt
- Fail-Safe Clock Monitor Interrupt
- CCP Interrupt

When an interrupt is serviced:

- The GIE is cleared to disable any further interrupt.
- The return address is pushed onto the stack.
- The PC is loaded with 0004h.

For external interrupt events, such as the INT pin or GPIO change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends upon when the interrupt event occurs (see Figure 12-8). The latency is the same for one or two-cycle instructions. Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid multiple interrupt requests.

| | |
|---|---|
| **Note 1:** | Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit. |
| **2:** | When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The interrupts, which were ignored, are still pending to be serviced when the GIE bit is set again. |

For additional information on Timer1, Timer2, comparators, A/D, data EEPROM or CCP modules, refer to the respective peripheral section.

### 12.4.1 GP2/INT INTERRUPT

External interrupt on GP2/INT pin is edge-triggered; either rising if the INTEDG bit (OPTION<6>) is set, or falling if the INTEDG bit is clear. When a valid edge appears on the GP2/INT pin, the INTF bit (INTCON<1>) is set. This interrupt can be disabled by clearing the INTE control bit (INTCON<4>). The INTF bit must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The GP2/INT interrupt can wake-up the processor from Sleep if the INTE bit was set prior to going into Sleep. The status of the GIE bit decides whether or not the processor branches to the interrupt vector following wake-up (0004h). See **Section 12.7 "Power-Down Mode (Sleep)"** for details on Sleep and Figure 12-10 for timing of wake-up from Sleep through GP2/INT interrupt.

| | |
|---|---|
| **Note:** | The ANSEL (91h) and CMCON0 (19h) registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'. |

### 12.4.2 TMR0 INTERRUPT

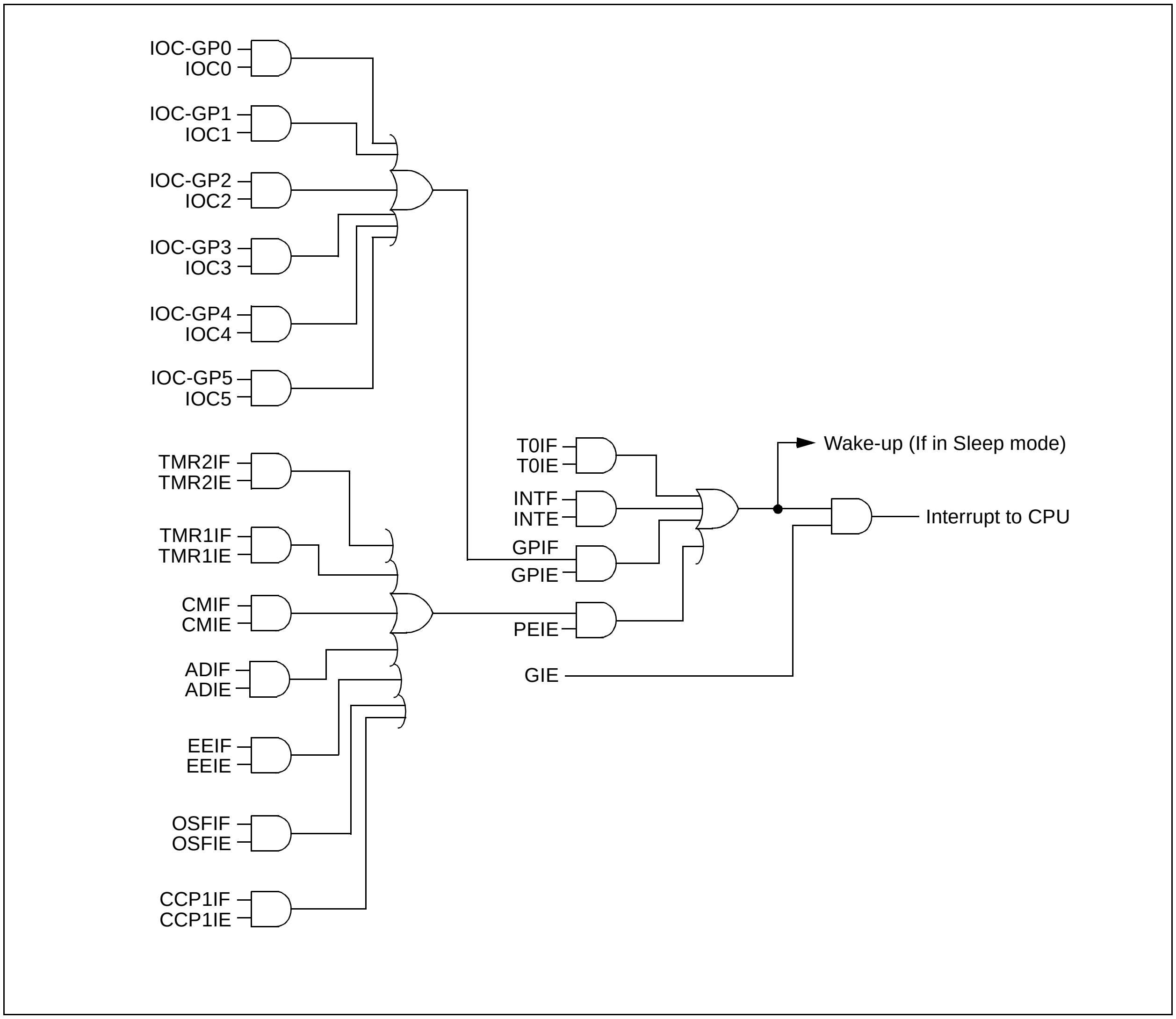
An overflow (FFh → 00h) in the TMR0 register will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing T0IE (INTCON<5>) bit. See **Section 5.0 “Timer0 Module”** for operation of the Timer0 module.

### 12.4.3 GPIO INTERRUPT

An input change on GPIO change sets the GPIF (INTCON<0>) bit. The interrupt can be enabled/ disabled by setting/clearing the GPIE (INTCON<3>) bit. Plus, individual pins can be configured through the IOC register.

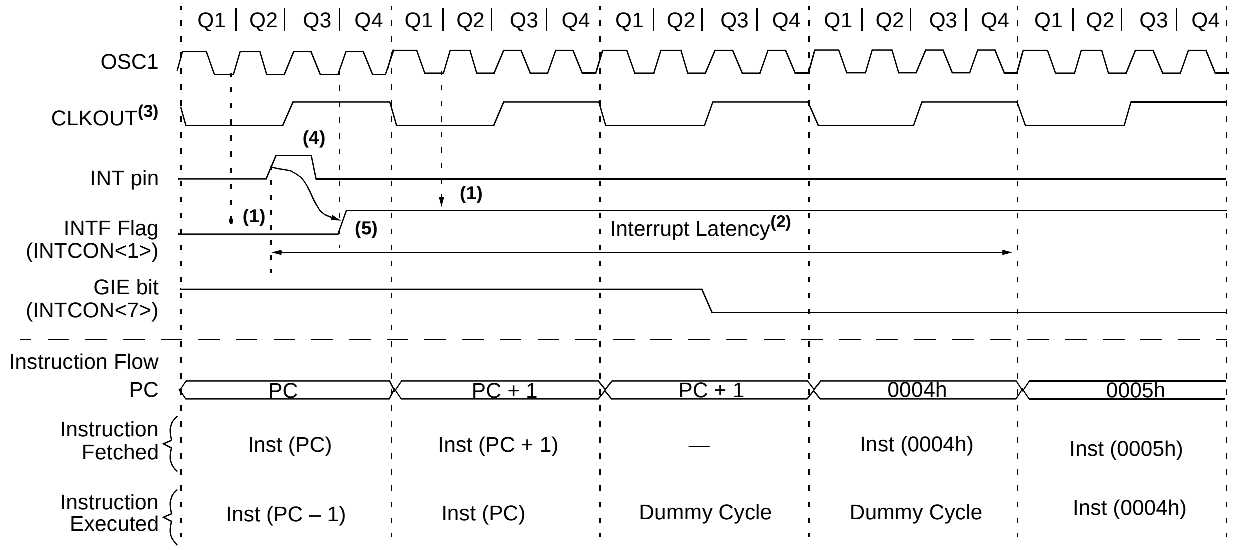
| **Note:** | If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the GPIF interrupt flag may not get set. |
|---|---|

**FIGURE 12-7: INTERRUPT LOGIC**



Preliminary

### FIGURE 12-8: INT PIN INTERRUPT TIMING



**Note 1:** INTF flag is sampled here (every Q1).

**2:** Asynchronous interrupt latency = 3-4 TCY. Synchronous latency = 3 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

**3:** CLKOUT is available only in INTOSC and RC Oscillator modes.

**4:** For minimum width of INT pulse, refer to AC specifications in **Section 15.0 “Electrical Specifications”**.

**5:** INTF is enabled to be set any time during the Q4-Q1 cycles.

### TABLE 12-6: SUMMARY OF INTERRUPT REGISTERS

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh, 8Bh | INTCON | GIE | PEIE | T0IE | INTE | GPIE | T0IF | INTF | GPIF | 0000 0000 | 0000 0000 |
| 0Ch | PIR1 | EEIF | ADIF | CCP1IF | — | CMIF | OSFIF | TMR2IF | TMR1IF | 000- 0000 | 000- 0000 |
| 8Ch | PIE1 | EEIE | ADIE | CCP1IE | — | CMIE | OSFIE | TMR2IE | TMR1IE | 000- 0000 | 000- 0000 |

**Legend:** x = unknown, u = unchanged, — = unimplemented read as ‘0’, q = value depends upon condition.
Shaded cells are not used by the interrupt module.

## 12.5 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (e.g., W and Status registers). This must be implemented in software.

Since the lower 16 bytes of all banks are common in the PIC12F683 (see Figure 2-2), temporary holding registers, W_TEMP and STATUS_TEMP, should be placed in here. These 16 locations do not require banking and therefore, makes it easier to context save and restore. The same code shown in Example 12-1 can be used to:

- Store the W register.
- Store the Status register.
- Execute the ISR code.
- Restore the Status (and Bank Select Bit register).
- Restore the W register.

**Note:** The PIC12F683 normally does not require saving the PCLATH. However, if computed `GOTO`s are used in the ISR and the main code, the PCLATH must be saved and restored in the ISR.

### EXAMPLE 12-1: SAVING STATUS AND W REGISTERS IN RAM

```
MOVWF   W_TEMP             ;Copy W to TEMP register
SWAPF   STATUS,W           ;Swap status to be saved into W
CLRF    STATUS             ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
MOVWF   STATUS_TEMP        ;Save status to bank zero STATUS_TEMP register
:
:(ISR)                     ;Insert user code here
:
SWAPF   STATUS_TEMP,W      ;Swap STATUS_TEMP register into W
                           ;(sets bank to original state)
MOVWF   STATUS             ;Move W into Status register
SWAPF   W_TEMP,F           ;Swap W_TEMP
SWAPF   W_TEMP,W           ;Swap W_TEMP into W
```

## 12.6 Watchdog Timer (WDT)

For PIC12F683, the WDT has been modified from previous PIC12F683 devices. The new WDT is code and functionally compatible with previous PIC12F683 WDT modules and adds a 16-bit prescaler to the WDT. This allows the user to have a scaler value for the WDT and TMR0 at the same time. In addition, the WDT time-out value can be extended to 268 seconds. WDT is cleared under certain conditions described in Table 12-7.

### 12.6.1 WDT OSCILLATOR

The WDT derives its time base from the 31 kHz LFINTOSC. The LTS bit does not reflect that the LFINTOSC is enabled.

The value of WDTCON is '`---0 1000`' on all Resets. This gives a nominal time base of 16 ms, which is compatible with the time base generated with previous PIC12F683 microcontroller versions.

| **Note:** | When the Oscillator Start-up Timer (OST) is invoked, the WDT is held in Reset, because the WDT Ripple Counter is used by the OST to perform the oscillator delay count. When the OST count has expired, the WDT will begin counting (if enabled). |
|---|---|

A new prescaler has been added to the path between the INTRC and the multiplexers used to select the path for the WDT. This prescaler is 16 bits and can be programmed to divide the INTRC by 128 to 65536, giving the time base used for the WDT a nominal range of 1 ms to 268s.

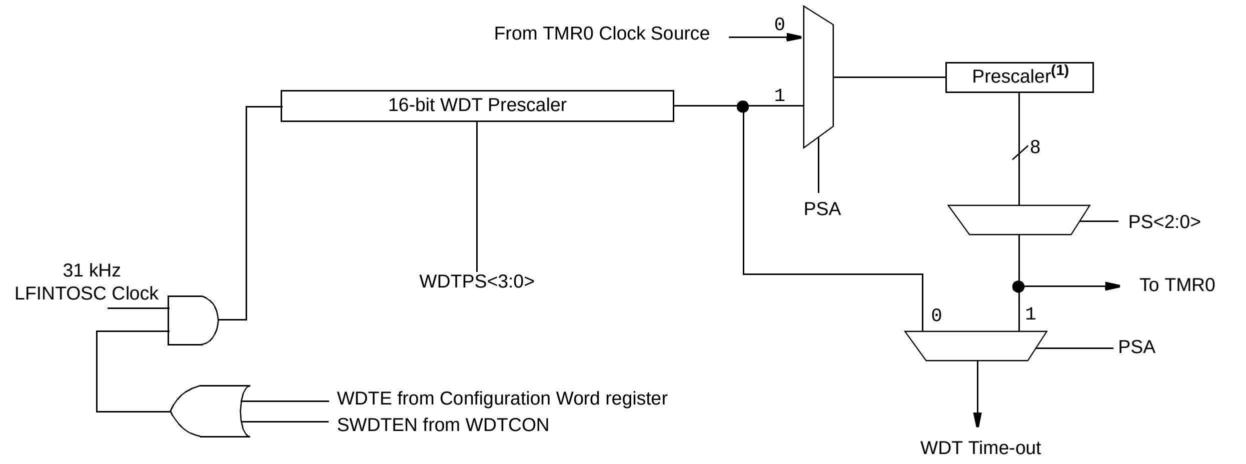
### 12.6.2 WDT CONTROL

The WDTE bit is located in the Configuration Word register. When set, the WDT runs continuously.

When the WDTE bit in the Configuration Word register is set, the SWDTEN bit (WDTCON<0>) has no effect. If WDTE is clear, then the SWDTEN bit can be used to enable and disable the WDT. Setting the bit will enable it and clearing the bit will disable it.

The PSA and PS<2:0> bits (OPTION_REG) have the same function as in previous versions of the PIC12F683 family of microcontrollers. See **Section 5.0 "Timer0 Module"** for more information.

**FIGURE 12-9: WATCHDOG TIMER BLOCK DIAGRAM**



**Note 1**: This is the shared Timer0/WDT prescaler. See **Section 5.4 "Prescaler"** for more information.

**TABLE 12-7: WDT STATUS**

| Conditions | WDT |
|---|---|
| WDTE = `0` | Cleared |
| `CLRWDT` Command | |
| Oscillator Fail Detected | |
| Exit Sleep + System Clock = T1OSC, EXTRC, INTRC, EXTCLK | |
| Exit Sleep + System Clock = XT, HS, LP | Cleared until the end of OST |

### REGISTER 12-3: WDTCON – WATCHDOG TIMER CONTROL REGISTER (ADDRESS: 18h)

| U-0 | U-0 | U-0 | R/W-0 | R/W-1 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | WDTPS3 | WDTPS2 | WDTPS1 | WDTPS0 | SWDTEN |
| bit 7 | | | | | | | bit 0 |

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-1 **WDTPS<3:0>:** Watchdog Timer Period Select bits

Bit Value = Prescale Rate

0000 = 1:32
0001 = 1:64
0010 = 1:128
0011 = 1:256
0100 = 1:512 (Reset value)
0101 = 1:1024
0110 = 1:2048
0111 = 1:4096
1000 = 1:8192
1001 = 1:16384
1010 = 1:32768
1011 = 1:65536
1100 = Reserved
1101 = Reserved
1110 = Reserved
1111 = Reserved

bit 0 **SWDTEN:** Software Enable or Disable the Watchdog Timer[(1)]

1 = WDT is turned on
0 = WDT is turned off (Reset value)

**Note 1:** If WDTE configuration bit = 1, then WDT is always enabled, irrespective of this control bit. If WDTE configuration bit = 0, then it is possible to turn WDT on/off with this control bit.

| **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 |

### TABLE 12-8: SUMMARY OF WATCHDOG TIMER REGISTERS

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 18h | WDTCON | — | — | — | WDTPS3 | WDTPS2 | WSTPS1 | WDTPS0 | SWDTEN |
| 81h | OPTION_REG | $\overline{\text{GPPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
| 2007h[(1)] | CONFIG | $\overline{\text{CPD}}$ | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | FOSC2 | FOSC1 | FOSC0 |

**Legend:** Shaded cells are not used by the Watchdog Timer.

**Note 1:** See Register 12-1 for operation of all Configuration Word register bits.

## 12.7 Power-Down Mode (Sleep)

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

If the Watchdog Timer is enabled:

- WDT will be cleared but keeps running.
- PD bit in the Status register is cleared.
- $\overline{\text{TO}}$ bit is set.
- Oscillator driver is turned off.
- I/O ports maintain the status they had before SLEEP was executed (driving high, low or high-impedance).

For lowest current consumption in this mode, all I/O pins should be either at VDD or VSS, with no external circuitry drawing current from the I/O pin and the comparators and CVREF should be disabled. I/O pins that are high-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS for lowest current consumption. The contribution from on-chip pull-ups on GPIO should be considered.

The $\overline{\text{MCLR}}$ pin must be at a logic high level.

**Note:** It should be noted that a Reset generated by a WDT time-out does not drive $\overline{\text{MCLR}}$ pin low.

### 12.7.1 WAKE-UP FROM SLEEP

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

1. External Reset input on $\overline{\text{MCLR}}$ pin.
2. Watchdog Timer wake-up (if WDT was enabled).
3. Interrupt from GP2/INT pin, GPIO change or a peripheral interrupt.

The first event will cause a device Reset. The two latter events are considered a continuation of program execution. The $\overline{\text{TO}}$ and PD bits in the Status register can be used to determine the cause of device Reset. The PD bit, which is set on power-up, is cleared when Sleep is invoked. $\overline{\text{TO}}$ bit is cleared if WDT wake-up occurred.

The following peripheral interrupts can wake the device from Sleep:

1. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
2. ECCP Capture mode interrupt.
3. Special event trigger (Timer1 in Asynchronous mode using an external clock).
4. A/D conversion (when A/D clock source is RC).
5. EEPROM write operation completion.
6. Comparator output changes state.
7. Interrupt-on-change.
8. External Interrupt from INT pin.

Other peripherals cannot generate interrupts, since during Sleep, no on-chip clocks are present.

When the SLEEP instruction is being executed, the next instruction (PC + 1) is prefetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction, then branches to the interrupt address (0004h). In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

**Note:** If the global interrupts are disabled (GIE is cleared), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake-up from Sleep. The SLEEP instruction is completely executed.

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

### 12.7.2 WAKE-UP USING INTERRUPTS

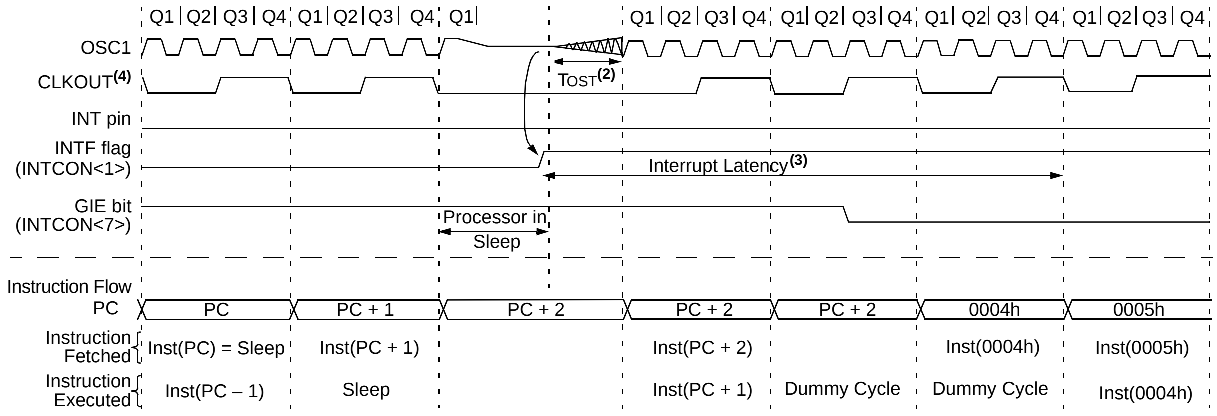
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a SLEEP instruction, the SLEEP instruction will complete as a NOP. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will not be cleared, the $\overline{\text{TO}}$ bit will not be set and the PD bit will not be cleared.
- If the interrupt occurs **during or after** the execution of a SLEEP instruction, the device will immediately wake-up from Sleep. The SLEEP instruction will be completely executed before the wake-up. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will be cleared, the $\overline{\text{TO}}$ bit will be set and the PD bit will be cleared.

Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the PD bit. If the PD bit is set, the SLEEP instruction was executed as a NOP.

To ensure that the WDT is cleared, a CLRWDT instruction should be executed before a SLEEP instruction.

**FIGURE 12-10: WAKE-UP FROM SLEEP THROUGH INTERRUPT**



**Note 1:** XT, HS or LP Oscillator mode assumed.
**2:** TOST = 1024 TOSC (drawing not to scale). This delay does not apply to EC and RCIO Oscillator modes.
**3:** GIE = 1 assumed. In this case after wake-up, the processor jumps to 0004h. If GIE = 0, execution will continue in-line.
**4:** CLKOUT is not available in XT, HS, LP or EC Oscillator modes, but shown here for timing reference.

## 12.8 Code Protection

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

**Note:** The entire data EEPROM and Flash program memory will be erased when the code protection is turned off. See the PIC12F6XX/16F6XX *Memory Programming Specification* (DS41204) for more information.

## 12.9 ID Locations

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution but are readable and writable during Program/Verify mode. Only the Least Significant 7 bits of the ID locations are used.

## 12.10 In-Circuit Serial Programming

The PIC12F683 microcontrollers 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
- Programming Voltage

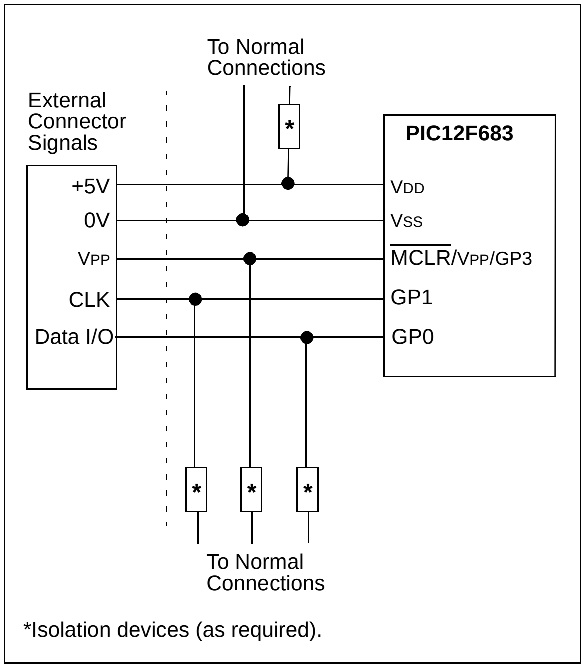
This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a Program/Verify mode by holding the GP0 and GP1 pins low, while raising the MCLR (VPP) pin from VIL to VIHH. See the "*PIC12F6XX/16F6XX Memory Programming Specification"* (DS41204) for more information. GP0 becomes the programming data and GP1 becomes the programming clock. Both GP0 and GP1 are Schmitt Trigger inputs in this mode.

After Reset, to place the device into Program/Verify mode, the Program Counter (PC) is at location 00h. 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 on whether the command was a load or a read. For complete details of serial programming, please refer to the "*PIC12F6XX/16F6XX Memory Programming Specification"* (DS41204).

A typical In-Circuit Serial Programming connection is shown in Figure 12-11.

**FIGURE 12-11: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**



## 12.11 In-Circuit Debugger

Since in-circuit debugging requires the loss of clock, data and MCLR pins, MPLAB® ICD 2 development with an 8-pin device is not practical. A special 14-pin PIC12F683 ICD device is used with MPLAB ICD 2 to provide separate clock, data and MCLR pins and frees all normally available pins to the user.

A special debugging adapter allows the ICD device to be used in place of a PIC12F683 device. The debugging adapter is the only source of the ICD device.

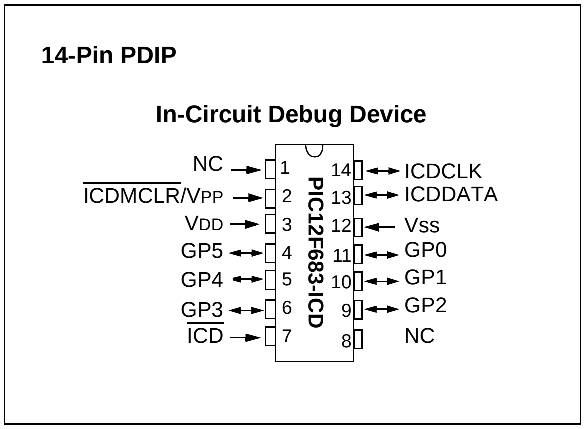
When the ICD pin on the PIC12F683 ICD device is held low, the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB ICD 2. When the microcontroller has this feature enabled, some of the resources are not available for general use. Table 12-9 shows which features are consumed by the background debugger:

**TABLE 12-9: DEBUGGER RESOURCES**

| Resource | Description |
|---|---|
| I/O pins | ICDCLK, ICDDATA |
| Stack | 1 level |
| Program Memory | Address 0h must be NOP<br>700h-7FFh |

For more information, see "*MPLAB ICD 2 In-Circuit Debugger User's Guide*" (DS51331), available on Microchip's web site (www.microchip.com).

**FIGURE 12-12: 14-PIN ICD PINOUT**



NOTES:

## 13.0 INSTRUCTION SET SUMMARY

The PIC12F683 instruction set is highly orthogonal and is comprised of three basic categories:

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

Each PIC16 instruction is a 14-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 13-1, while the various opcode fields are summarized in Table 13-1.

Table 13-2 lists the instructions recognized by the MPASM™ Assembler. A complete description of each instruction is also available in the "*PICmicro® Mid-Range MCU Family Reference Manual*" (DS33023).

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 zero, the result is placed in the W register. If 'd' is one, 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 bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a normal instruction execution time of 1 μs. 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. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

| **Note:** | To maintain upward compatibility with future products, do not use the OPTION and TRIS instructions. |
|---|---|

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

### 13.1 READ-MODIFY-WRITE OPERATIONS

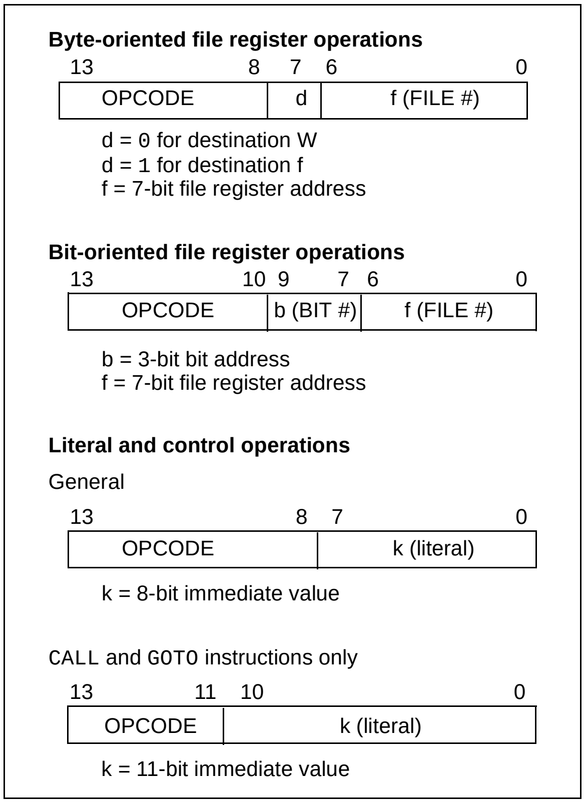
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified and the result is stored according to either the instruction or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a CLRF GPIO instruction will read GPIO, clear all the data bits, then write the result back to GPIO. This example would have the unintended result of clearing the condition that set the GPIF flag.

**TABLE 13-1: 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).<br>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,<br>d = 1: store result in file register f.<br>Default is d = 1. |
| PC | Program Counter |
| $\overline{\text{TO}}$ | Time-out bit |
| PD | Power-down bit |

**FIGURE 13-1: GENERAL FORMAT FOR INSTRUCTIONS**


Byte-oriented file register operations
13
8
7
6
0
OPCODE
d
f (FILE #)
d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address
Bit-oriented file register operations
13
10
9
7
6
0
OPCODE
b (BIT #)
f (FILE #)
b = 3-bit bit address
f = 7-bit file register address
Literal and control operations
General
13
8
7
0
OPCODE
k (literal)
k = 8-bit immediate value
CALL and GOTO instructions only
13
11
10
0
OPCODE
k (literal)
k = 11-bit immediate value

### TABLE 13-2: PIC12F683 INSTRUCTION SET

| Mnemonic, Operands | | Description | Cycles | 14-Bit Opcode | | | | Status Affected | Notes |
|---|---|---|---|---|---|---|---|---|---|
| | | | | MSb | | | LSb | | |
| **BYTE-ORIENTED FILE REGISTER OPERATIONS** | | | | | | | | | |
| `ADDWF` | f, d | Add W and f | 1 | `00` | `0111` | `dfff` | `ffff` | C, DC, Z | **1,2** |
| `ANDWF` | f, d | AND W with f | 1 | `00` | `0101` | `dfff` | `ffff` | Z | **1,2** |
| `CLRF` | f | Clear f | 1 | `00` | `0001` | `lfff` | `ffff` | Z | **2** |
| `CLRW` | - | Clear W | 1 | `00` | `0001` | `0xxx` | `xxxx` | Z | |
| `COMF` | f, d | Complement f | 1 | `00` | `1001` | `dfff` | `ffff` | Z | **1,2** |
| `DECF` | f, d | Decrement f | 1 | `00` | `0011` | `dfff` | `ffff` | Z | **1,2** |
| `DECFSZ` | f, d | Decrement f, Skip if 0 | 1 (2) | `00` | `1011` | `dfff` | `ffff` | | **1,2,3** |
| `INCF` | f, d | Increment f | 1 | `00` | `1010` | `dfff` | `ffff` | Z | **1,2** |
| `INCFSZ` | f, d | Increment f, Skip if 0 | 1 (2) | `00` | `1111` | `dfff` | `ffff` | | **1,2,3** |
| `IORWF` | f, d | Inclusive OR W with f | 1 | `00` | `0100` | `dfff` | `ffff` | Z | **1,2** |
| `MOVF` | f, d | Move f | 1 | `00` | `1000` | `dfff` | `ffff` | Z | **1,2** |
| `MOVWF` | f | Move W to f | 1 | `00` | `0000` | `lfff` | `ffff` | | |
| `NOP` | - | No Operation | 1 | `00` | `0000` | `0xx0` | `0000` | | |
| `RLF` | f, d | Rotate Left f through Carry | 1 | `00` | `1101` | `dfff` | `ffff` | C | **1,2** |
| `RRF` | f, d | Rotate Right f through Carry | 1 | `00` | `1100` | `dfff` | `ffff` | C | **1,2** |
| `SUBWF` | f, d | Subtract W from f | 1 | `00` | `0010` | `dfff` | `ffff` | C, DC, Z | **1,2** |
| `SWAPF` | f, d | Swap nibbles in f | 1 | `00` | `1110` | `dfff` | `ffff` | | **1,2** |
| `XORWF` | f, d | Exclusive OR W with f | 1 | `00` | `0110` | `dfff` | `ffff` | Z | **1,2** |
| **BIT-ORIENTED FILE REGISTER OPERATIONS** | | | | | | | | | |
| `BCF` | f, b | Bit Clear f | 1 | `01` | `00bb` | `bfff` | `ffff` | | **1,2** |
| `BSF` | f, b | Bit Set f | 1 | `01` | `01bb` | `bfff` | `ffff` | | **1,2** |
| `BTFSC` | f, b | Bit Test f, Skip if Clear | 1 (2) | `01` | `10bb` | `bfff` | `ffff` | | **3** |
| `BTFSS` | f, b | Bit Test f, Skip if Set | 1 (2) | `01` | `11bb` | `bfff` | `ffff` | | **3** |
| **LITERAL AND CONTROL OPERATIONS** | | | | | | | | | |
| `ADDLW` | k | Add literal and W | 1 | `11` | `111x` | `kkkk` | `kkkk` | C, DC, Z | |
| `ANDLW` | k | AND literal with W | 1 | `11` | `1001` | `kkkk` | `kkkk` | Z | |
| `CALL` | k | Call subroutine | 2 | `10` | `0kkk` | `kkkk` | `kkkk` | | |
| `CLRWDT` | - | Clear Watchdog Timer | 1 | `00` | `0000` | `0110` | `0100` | $\overline{\text{TO}}$, PD | |
| `GOTO` | k | Go to address | 2 | `10` | `1kkk` | `kkkk` | `kkkk` | | |
| `IORLW` | k | Inclusive OR literal with W | 1 | `11` | `1000` | `kkkk` | `kkkk` | Z | |
| `MOVLW` | k | Move literal to W | 1 | `11` | `00xx` | `kkkk` | `kkkk` | | |
| `RETFIE` | - | Return from interrupt | 2 | `00` | `0000` | `0000` | `1001` | | |
| `RETLW` | k | Return with literal in W | 2 | `11` | `01xx` | `kkkk` | `kkkk` | | |
| `RETURN` | - | Return from Subroutine | 2 | `00` | `0000` | `0000` | `1000` | | |
| `SLEEP` | - | Go into Standby mode | 1 | `00` | `0000` | `0110` | `0011` | $\overline{\text{TO}}$, PD | |
| `SUBLW` | k | Subtract W from literal | 1 | `11` | `110x` | `kkkk` | `kkkk` | C, DC, Z | |
| `XORLW` | k | Exclusive OR literal with W | 1 | `11` | `1010` | `kkkk` | `kkkk` | Z | |

**Note 1:** 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`’.

**2:** If this instruction is executed on the TMR0 register (and where applicable, d = `1`), the prescaler will be cleared if assigned to the Timer0 module.

**3:** If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

**Note:** Additional information on the mid-range instruction set is available in the “*PICmicro® Mid-Range MCU Family Reference Manual”* (DS33023).

## 13.2 Instruction Descriptions

| ADDLW | Add Literal and W |
|---|---|
| Syntax: | [ *label* ] ADDLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W) + k → (W) |
| Status Affected: | C, DC, Z |
| Description: | The contents of the W register are added to the eight-bit literal ‘k’ and the result is placed in the W register. |

| ADDWF | Add W and f |
|---|---|
| Syntax: | [ *label* ] ADDWF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) + (f) → (destination) |
| Status Affected: | C, DC, Z |
| Description: | Add the contents of the W register with register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’. |

| ANDLW | AND Literal with W |
|---|---|
| Syntax: | [ *label* ] ANDLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W) .AND. (k) → (W) |
| Status Affected: | Z |
| Description: | The contents of 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 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) .AND. (f) → (destination) |
| Status Affected: | Z |
| Description: | AND the W register with register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’. |

| BCF | Bit Clear f |
|---|---|
| Syntax: | [ *label* ] BCF f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 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 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | 1 → (f<b>) |
| Status Affected: | None |
| Description: | Bit ‘b’ in register ‘f’ is set. |

| BTFSC | Bit Test, Skip if Clear |
|---|---|
| Syntax: | [ *label* ] BTFSC f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | skip if (f<b>) = 0 |
| Status Affected: | None |
| Description: | If bit ‘b’ in register ‘f’ is ‘1’, the next instruction is executed.<br>If bit ‘b’ in register ‘f’ is ‘0’, the next instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction. |

| BTFSS | Bit Test f, Skip if Set |
|---|---|
| Syntax: | [ *label* ] BTFSS f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b < 7 |
| Operation: | skip if (f<b>) = 1 |
| Status Affected: | None |
| Description: | If bit ‘b’ in register ‘f’ is ‘0’, the next instruction is executed.<br>If bit ‘b’ is ‘1’, then the next instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction. |

| CALL | Call Subroutine |
|---|---|
| Syntax: | [ *label* ] CALL k |
| Operands: | $0 \le k \le 2047$ |
| Operation: | (PC) + 1 → TOS,<br>k → PC<10:0>,<br>(PCLATH<4:3>) → PC<12:11> |
| Status Affected: | None |
| Description: | Call subroutine. First, return address (PC + 1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction. |

| CLRF | Clear f |
|---|---|
| Syntax: | [ *label* ] CLRF f |
| Operands: | $0 \le f \le 127$ |
| Operation: | 00h → (f)<br>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)<br>1 → Z |
| Status Affected: | Z |
| Description: | W register is cleared. Zero bit (Z) is set. |

| CLRWDT | Clear Watchdog Timer |
|---|---|
| Syntax: | [ *label* ] CLRWDT |
| Operands: | None |
| Operation: | 00h → WDT<br>0 → WDT prescaler,<br>1 → $\overline{TO}$<br>1 → PD |
| Status Affected: | $\overline{TO}$, PD |
| Description: | CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT.<br>Status bits $\overline{TO}$ and PD are set. |

| COMF | Complement f |
|---|---|
| Syntax: | [ *label* ] COMF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | $(\overline{f})$ → (destination) |
| Status Affected: | Z |
| Description: | The contents of register ‘f’ are complemented. If ‘d’ is ‘0’, the result is stored in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’. |

| DECF | Decrement f |
|---|---|
| Syntax: | [ *label* ] DECF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f) – 1 → (destination) |
| 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 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) – 1 → (destination);<br>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’.<br>If the result is ‘1’, the next instruction is executed. If the result is ‘0’, then a NOP is executed instead, making it a 2-cycle instruction. |

| GOTO | Unconditional Branch |
|---|---|
| Syntax: | [ *label* ] GOTO k |
| Operands: | 0 ≤ k ≤ 2047 |
| Operation: | k → PC<10:0><br>PCLATH<4:3> → PC<12:11> |
| Status Affected: | None |
| Description: | GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction. |

| INCF | Increment f |
|---|---|
| Syntax: | [ *label* ] INCF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) + 1 → (destination) |
| 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 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) + 1 → (destination),<br>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’.<br>If the result is ‘1’, the next instruction is executed. If the result is ‘0’, a NOP is executed instead, making it a 2-cycle instruction. |

| MOVF | Move f |
|---|---|
| Syntax: | [ *label* ] MOVF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) → (dest) |
| Status Affected: | Z |
| Encoding: | `00` · `1000` · `dfff` · `ffff` |
| Description: | The contents of register ‘f’ are moved to a destination dependent upon the status of ‘d’. If ‘d’ = 0, destination is W register. If ‘d’ = 1, the destination is file register ‘f’ itself. ‘d’ = 1 is useful to test a file register since status flag Z is affected. |
| Words: | 1 |
| Cycles: | 1 |
| Example: | `MOVF FSR, 0`<br>After Instruction<br>W = value in FSR register<br>Z = 1 |

### MOVLW Move Literal to W

Syntax: [ *label* ] MOVLW k

Operands: $0 \le k \le 255$

Operation: k → (W)

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| `11` | `00xx` | `kkkk` | `kkkk` |

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

Words: 1

Cycles: 1

Example:

```
MOVLW  0x5A
```

After Instruction
W = 0x5A

### MOVWF Move W to f

Syntax: [ *label* ] MOVWF f

Operands: $0 \le f \le 127$

Operation: (W) → (f)

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| `00` | `0000` | `1fff` | `ffff` |

Description: Move data from W register to register ‘f’.

Words: 1

Cycles: 1

Example:

```
MOVWF  OPTION
```

Before Instruction
OPTION = 0xFF
W = 0x4F
After Instruction
OPTION = 0x4F
W = 0x4F

### NOP No Operation

Syntax: [ *label* ] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| `00` | `0000` | `0xx0` | `0000` |

Description: No operation.

Words: 1

Cycles: 1

Example:

```
NOP
```

### 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: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .OR. (f) → (destination)

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’.

### RETFIE Return from Interrupt

Syntax: [ *label* ] RETFIE

Operands: None

Operation: TOS → PC, 1 → GIE

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| `00` | `0000` | `0000` | `1001` |

Description: Return from interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction.

Words: 1

Cycles: 2

Example:

```
RETFIE
```

After Interrupt
PC = TOS
GIE = 1

| RETLW | Return with Literal in W |
|---|---|
| Syntax: | [ *label* ] RETLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | k → (W);<br>TOS → PC |
| Status Affected: | None |
| Encoding: | `11` \| `01xx` \| `kkkk` \| `kkkk` |
| Description: | The W register is loaded with the eight-bit literal ‘k’. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction. |
| Words: | 1 |
| Cycles: | 2 |

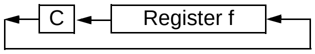
<u>Example:</u>

```
CALL TABLE;W contains table
          ;offset value
•         ;W now has table value
•
•
ADDWF PC   ;W = offset
RETLW k1   ;Begin table
RETLW k2   ;
•
•
•
RETLW kn   ; End of table
```

Before Instruction
W = 0x07
After Instruction
W = value of k8

| RETURN | Return from Subroutine |
|---|---|
| Syntax: | [ *label* ] RETURN |
| Operands: | None |
| Operation: | TOS → PC |
| Status Affected: | None |
| Description: | Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction. |

| RLF | Rotate Left f through Carry |
|---|---|
| Syntax: | [ *label* ] RLF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | See description below |
| Status Affected: | C |
| Encoding: | `00` \| `1101` \| `dfff` \| `ffff` |
| 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’. |



| | |
|---|---|
| Words: | 1 |
| Cycles: | 1 |

<u>Example:</u>

```
RLF     REG1,0
```

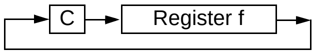
Before Instruction

| | | |
|---|---|---|
| REG1 | = | 1110 0110 |
| C | = | 0 |

After Instruction

| | | |
|---|---|---|
| REG1 | = | 1110 0110 |
| W | = | 1100 1100 |
| C | = | 1 |

| RRF | Rotate Right f through Carry |
|---|---|
| Syntax: | [ *label* ] RRF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [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 | |
|---|---|
| Syntax: | [ *label* ] SLEEP |
| Operands: | None |
| Operation: | 00h → WDT,<br>0 → WDT prescaler,<br>1 → $\overline{TO}$,<br>0 → PD |
| Status Affected: | $\overline{TO}$, PD |
| Description: | The Power-down status bit, PD, is cleared. Time-out status bit, $\overline{TO}$, is set. Watchdog Timer and its prescaler are cleared.<br>The processor is put into Sleep mode with the oscillator stopped. |

| SUBLW | Subtract W from Literal |
|---|---|
| Syntax: | [ *label* ] SUBLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | k - (W) → (W) |
| Status Affected: | C, DC, Z |
| Description: | The W register is subtracted (2’s complement method) from the eight-bit literal ‘k’. The result is placed in the W register. |

| SUBWF | Subtract W from f |
|---|---|
| Syntax: | [ *label* ] SUBWF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) – (W) → (destination) |
| Status Affected: | C, DC, Z |
| Description: | Subtract (2’s complement method) W register from register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’. |

| SWAPF | Swap Nibbles in f |
|---|---|
| Syntax: | [ *label* ] SWAPF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f<3:0>) → (destination<7:4>),<br>(f<7:4>) → (destination<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 the W register. If ‘d’ is ‘1’, the result is placed in register ‘f’. |

| 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 ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) .XOR. (f) → (destination) |
| 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’. |

## 14.0 DEVELOPMENT SUPPORT

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

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C17 and MPLAB C18 C Compilers
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB C30 C Compiler
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
  - MPLAB dsPIC30 Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB ICE 4000 In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PRO MATE® II Universal Device Programmer
  - PICSTART® Plus Development Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration Boards
  - PICDEM™ 1 Demonstration Board
  - PICDEM.net™ Demonstration Board
  - PICDEM 2 Plus Demonstration Board
  - PICDEM 3 Demonstration Board
  - PICDEM 4 Demonstration Board
  - PICDEM 17 Demonstration Board
  - PICDEM 18R Demonstration Board
  - PICDEM LIN Demonstration Board
  - PICDEM USB Demonstration Board
- Evaluation Kits
  - KEELOQ®
  - PICDEM MSC
  - microID®
  - CAN
  - PowerSmart®
  - Analog

## 14.1 MPLAB Integrated Development Environment Software

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

- An interface to debugging tools
  - simulator
  - programmer (sold separately)
  - 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
- Extensive on-line help

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PICmicro emulator and simulator tools (automatically updates all project information)
- Debug using:
  - source files (assembly or C)
  - mixed assembly and C
  - 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 increasing flexibility and power.

## 14.2 MPASM Assembler

The MPASM assembler is a full-featured, universal macro assembler for all PICmicro 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

## 14.3 MPLAB C17 and MPLAB C18 C Compilers

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

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

## 14.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. 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

## 14.5 MPLAB C30 C Compiler

The MPLAB C30 C compiler is a full-featured, ANSI compliant, optimizing compiler that translates standard ANSI C programs into dsPIC30F assembly language source. The compiler also supports many command line options and language extensions to take full advantage of the dsPIC30F device hardware capabilities and afford fine control of the compiler code generator.

MPLAB C30 is distributed with a complete ANSI C standard library. All library functions have been validated and conform to the ANSI C library standard. The library includes functions for string manipulation, dynamic memory allocation, data conversion, timekeeping and math functions (trigonometric, exponential and hyperbolic). The compiler provides symbolic information for high-level source debugging with the MPLAB IDE.

## 14.6 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 compiler uses the assembler to produce it's 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 dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

## 14.7 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any pin. The execution can be performed in Single-Step, Execute Until Break or Trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and MPLAB C18 C Compilers, as well as the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent, economical software development tool.

## 14.8 MPLAB SIM30 Software Simulator

The MPLAB SIM30 software simulator allows code development in a PC hosted environment by simulating the dsPIC30F series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any of the pins.

The MPLAB SIM30 simulator fully supports symbolic debugging using the MPLAB C30 C Compiler and MPLAB ASM30 assembler. The simulator runs in either a Command Line mode for automated tasks, or from MPLAB IDE. This high-speed simulator is designed to debug, analyze and optimize time intensive DSP routines.

## 14.9 MPLAB ICE 2000 High-Performance Universal In-Circuit Emulator

The MPLAB ICE 2000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PICmicro microcontrollers. Software control of the MPLAB ICE 2000 in-circuit emulator is advanced by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PICmicro microcontrollers.

The MPLAB ICE 2000 in-circuit emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft® Windows 32-bit operating system were chosen to best make these features available in a simple, unified application.

## 14.10 MPLAB ICE 4000 High-Performance Universal In-Circuit Emulator

The MPLAB ICE 4000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for high-end PICmicro microcontrollers. Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICD 4000 is a premium emulator system, providing the features of MPLAB ICE 2000, but with increased emulation memory and high-speed performance for dsPIC30F and PIC18XXXX devices. Its advanced emulator features include complex triggering and timing, up to 2 Mb of emulation memory and the ability to view variables in real-time.

The MPLAB ICE 4000 in-circuit emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft Windows 32-bit operating system were chosen to best make these features available in a simple, unified application.

## 14.11 MPLAB ICD 2 In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low-cost, run-time development tool, connecting to the host PC via an RS-232 or high-speed USB interface. This tool is based on the Flash PICmicro MCUs and can be used to develop for these and other PICmicro microcontrollers. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the Flash devices. This feature, along with Microchip's In-Circuit Serial Programming™ (ICSP™) protocol, offers cost effective in-circuit Flash debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single-stepping and watching variables, CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real-time. MPLAB ICD 2 also serves as a development programmer for selected PICmicro devices.

## 14.12 PRO MATE II Universal Device Programmer

The PRO MATE II is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features an LCD display for instructions and error messages and a modular detachable socket assembly to support various package types. In Stand-Alone mode, the PRO MATE II device programmer can read, verify and program PICmicro devices without a PC connection. It can also set code protection in this mode.

## 14.13 MPLAB PM3 Device Programmer

The MPLAB PM3 is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX 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 PICmicro devices without a PC connection. It can also set code protection in this mode. MPLAB PM3 connects to the host PC via an RS-232 or USB cable. MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an SD/MMC card for file storage and secure data applications.

## 14.14 PICSTART Plus Development Programmer

The PICSTART Plus development programmer is an easy-to-use, low-cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus development programmer supports most PICmicro devices up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus development programmer is CE compliant.

## 14.15 PICDEM 1 PICmicro Demonstration Board

The PICDEM 1 demonstration board demonstrates the capabilities of the PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The sample microcontrollers provided with the PICDEM 1 demonstration board can be programmed with a PRO MATE II device programmer or a PICSTART Plus development programmer. The PICDEM 1 demonstration board can be connected to the MPLAB ICE in-circuit emulator for testing. A prototype area extends the circuitry for additional application components. Features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs.

## 14.16 PICDEM.net Internet/Ethernet Demonstration Board

The PICDEM.net demonstration board is an Internet/ Ethernet demonstration board using the PIC18F452 microcontroller and TCP/IP firmware. The board supports any 40-pin DIP device that conforms to the standard pinout used by the PIC16F877 or PIC18C452. This kit features a user friendly TCP/IP stack, web server with HTML, a 24L256 Serial EEPROM for Xmodem download to web pages into Serial EEPROM, ICSP/MPLAB ICD 2 interface connector, an Ethernet interface, RS-232 interface and a 16 x 2 LCD display. Also included is the book and CD-ROM *“TCP/IP Lean, Web Servers for Embedded Systems,”* by Jeremy Bentham

## 14.17 PICDEM 2 Plus Demonstration Board

The PICDEM 2 Plus demonstration board supports many 18, 28 and 40-pin microcontrollers, including PIC16F87X and PIC18FXX2 devices. All the necessary hardware and software is included to run the demonstration programs. The sample microcontrollers provided with the PICDEM 2 demonstration board can be programmed with a PRO MATE II device programmer, PICSTART Plus development programmer, or MPLAB ICD 2 with a Universal Programmer Adapter. The MPLAB ICD 2 and MPLAB ICE in-circuit emulators may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area extends the circuitry for additional application components. Some of the features include an RS-232 interface, a 2 x 16 LCD display, a piezo speaker, an on-board temperature sensor, four LEDs and sample PIC18F452 and PIC16F877 Flash microcontrollers.

## 14.18 PICDEM 3 PIC16C92X Demonstration Board

The PICDEM 3 demonstration board supports the PIC16C923 and PIC16C924 in the PLCC package. All the necessary hardware and software is included to run the demonstration programs.

## 14.19 PICDEM 4 8/14/18-Pin Demonstration Board

The PICDEM 4 can be used to demonstrate the capabilities of the 8, 14 and 18-pin PIC16XXXX and PIC18XXXX MCUs, including the PIC16F818/819, PIC16F87/88, PIC16F62XA and the PIC18F1320 family of microcontrollers. PICDEM 4 is intended to showcase the many features of these low pin count parts, including LIN and Motor Control using ECCP. Special provisions are made for low-power operation with the supercapacitor circuit and jumpers allow on-board hardware to be disabled to eliminate current draw in this mode. Included on the demo board are provisions for Crystal, RC or Canned Oscillator modes, a five volt regulator for use with a nine volt wall adapter or battery, DB-9 RS-232 interface, ICD connector for programming via ICSP and development with MPLAB ICD 2, 2 x 16 liquid crystal display, PCB footprints for H-Bridge motor driver, LIN transceiver and EEPROM. Also included are: header for expansion, eight LEDs, four potentiometers, three push buttons and a prototyping area. Included with the kit is a PIC16F627A and a PIC18F1320. Tutorial firmware is included along with the User’s Guide.

## 14.20 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, PIC17C762 and PIC17C766. A programmed sample is included. The PRO MATE II device programmer, or the PICSTART Plus development programmer, can be used to reprogram the device for user tailored application development. The PICDEM 17 demonstration board supports program download and execution from external on-board Flash memory. A generous prototype area is available for user hardware expansion.

## 14.21 PICDEM 18R PIC18C601/801 Demonstration Board

The PICDEM 18R demonstration board serves to assist development of the PIC18C601/801 family of Microchip microcontrollers. It provides hardware implementation of both 8-bit Multiplexed/Demultiplexed and 16-bit Memory modes. The board includes 2 Mb external Flash memory and 128 Kb SRAM memory, as well as serial EEPROM, allowing access to the wide range of memory types supported by the PIC18C601/801.

## 14.22 PICDEM LIN PIC16C43X Demonstration Board

The powerful LIN hardware and software kit includes a series of boards and three PICmicro microcontrollers. The small footprint PIC16C432 and PIC16C433 are used as slaves in the LIN communication and feature on-board LIN transceivers. A PIC16F874 Flash microcontroller serves as the master. All three microcontrollers are programmed with firmware to provide LIN bus communication.

## 14.23 PICkit$^{TM}$ 1 Flash Starter Kit

A complete “development system in a box”, the PICkit Flash Starter Kit includes a convenient multi-section board for programming, evaluation and development of 8/14-pin Flash PIC® microcontrollers. Powered via USB, the board operates under a simple Windows GUI. The PICkit 1 Starter Kit includes the User’s Guide (on CD ROM), PICkit 1 tutorial software and code for various applications. Also included are MPLAB® IDE (Integrated Development Environment) software, software and hardware “Tips 'n Tricks for 8-pin Flash PIC® Microcontrollers” Handbook and a USB interface cable. Supports all current 8/14-pin Flash PIC microcontrollers, as well as many future planned devices.

## 14.24 PICDEM USB PIC16C7X5 Demonstration Board

The PICDEM USB Demonstration Board shows off the capabilities of the PIC16C745 and PIC16C765 USB microcontrollers. This board provides the basis for future USB products.

## 14.25 Evaluation and Programming Tools

In addition to the PICDEM series of circuits, Microchip has a line of evaluation kits and demonstration software for these products.

- KEELOQ evaluation and programming tools for Microchip’s HCS Secure Data Products
- CAN developers kit for automotive network applications
- Analog design boards and filter design software
- PowerSmart battery charging evaluation/ calibration kits
- IrDA® development kit
- microID development and rfLab™ development software
- SEEVAL® designer kit for memory evaluation and endurance calculations
- PICDEM MSC demo boards for Switching mode power supply, high-power IR driver, delta sigma ADC and flow rate sensor

Check the Microchip web page and the latest Product Selector Guide for the complete list of demonstration and evaluation kits.

NOTES:

## 15.0 ELECTRICAL SPECIFICATIONS

### Absolute Maximum Ratings(†)

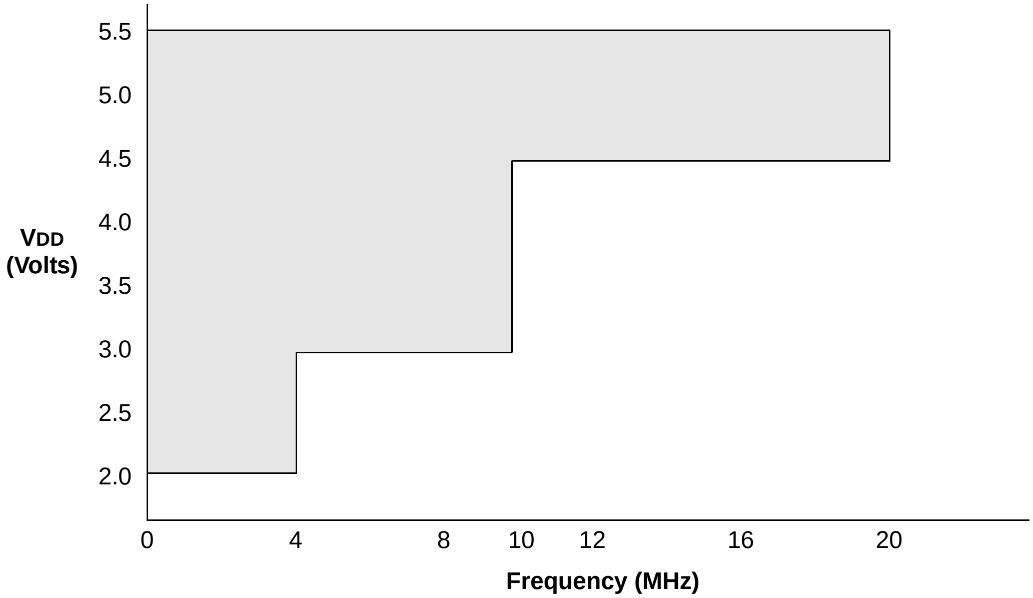
| | |
|---|---|
| Ambient temperature under bias | -40°C to +125°C |
| Storage temperature | -65°C to +150°C |
| Voltage on $V_{DD}$ with respect to $V_{SS}$ | -0.3V to +6.5V |
| Voltage on $\overline{MCLR}$ with respect to Vss | -0.3V to +13.5V |
| Voltage on all other pins with respect to $V_{SS}$ | -0.3V to ($V_{DD}$ + 0.3V) |
| Total power dissipation(1) | 800 mW |
| Maximum current out of $V_{SS}$ pin | 300 mA |
| Maximum current into $V_{DD}$ pin | 250 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 |
| Maximum output current sunk by any I/O pin | 25 mA |
| Maximum output current sourced by any I/O pin | 25 mA |
| Maximum current sunk by GPIO | 200 mA |
| Maximum current sourced GPIO | 200 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.

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

FIGURE 15-1: PIC12F683 VOLTAGE-FREQUENCY GRAPH, -40°C ≤ $T_A$ ≤ +125°C



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

### 15.1 DC Characteristics: PIC12F683-I (Industrial) PIC12F683-E (Extended)

| **DC CHARACTERISTICS** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ $T_A$ ≤ +85°C for industrial<br>-40°C ≤ $T_A$ ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | $V_{DD}$ | **Supply Voltage** | | | | | |
| D001 | | | 2.0 | — | 5.5 | V | $F_{OSC}$ < = 4 MHz |
| D001C | | | 3.0 | — | 5.5 | V | $F_{OSC}$ < = 10 MHz |
| D001D | | | 4.5 | — | 5.5 | V | $F_{OSC}$ < = 20 MHz |
| D002 | $V_{DR}$ | **RAM Data Retention Voltage**[1] | 1.5* | — | — | V | Device in Sleep mode |
| D003 | $V_{POR}$ | **$V_{DD}$ Start Voltage** to ensure internal Power-on Reset signal | — | $V_{SS}$ | — | V | See **Section 12.3.1 "Power-on Reset"** for details |
| D004 | $S_{VDD}$ | **$V_{DD}$ Rise Rate** to ensure internal Power-on Reset signal | 0.05* | — | — | V/ms | See **Section 12.3.1 "Power-on Reset"** for details |
| D005 | $V_{BOD}$ | **Brown-out Detect** | — | 2.1 | — | V | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** This is the limit to which $V_{DD}$ can be lowered in Sleep mode without losing RAM data.

## 15.2 DC Characteristics: PIC12F683-I (Industrial)

| **DC CHARACTERISTICS** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Device Characteristics** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** | |
| | | | | | | | **VDD** | **Note** |
| D010 | IDD | **Supply Current**[(1,2)] | — | 9 | TBD | µA | 2.0 | FOSC = 32 kHz<br>LP Oscillator mode |
| | | | — | 18 | TBD | µA | 3.0 | |
| | | | — | 35 | TBD | µA | 5.0 | |
| D011 | | | — | 110 | TBD | µA | 2.0 | FOSC = 1 MHz<br>XT Oscillator mode |
| | | | — | 190 | TBD | µA | 3.0 | |
| | | | — | 330 | TBD | µA | 5.0 | |
| D012 | | | — | 220 | TBD | µA | 2.0 | FOSC = 4 MHz<br>XT Oscillator mode |
| | | | — | 370 | TBD | µA | 3.0 | |
| | | | — | 0.6 | TBD | µA | 5.0 | |
| D013 | | | — | 70 | TBD | µA | 2.0 | FOSC = 1 MHz<br>EC Oscillator mode |
| | | | — | 140 | TBD | µA | 3.0 | |
| | | | — | 260 | TBD | µA | 5.0 | |
| D014 | | | — | 180 | TBD | µA | 2.0 | FOSC = 4 MHz<br>EC Oscillator mode |
| | | | — | 320 | TBD | µA | 3.0 | |
| | | | — | 580 | TBD | µA | 5.0 | |
| D015 | | | — | 10 | TBD | µA | 2.0 | FOSC = 31 kHz<br>INTRC mode |
| | | | — | 25 | TBD | µA | 3.0 | |
| | | | — | 40 | TBD | µA | 5.0 | |
| D016 | | | — | 340 | TBD | µA | 2.0 | FOSC = 4 MHz<br>INTOSC mode |
| | | | — | 500 | TBD | µA | 3.0 | |
| | | | — | 0.8 | TBD | mA | 5.0 | |
| D017 | | | — | 250 | TBD | µA | 2.0 | FOSC = 4 MHz<br>EXTRC mode |
| | | | — | 375 | TBD | µA | 3.0 | |
| | | | — | 750 | TBD | µA | 5.0 | |
| D018 | | | — | 3.0 | TBD | mA | 4.5 | FOSC = 20 MHz<br>HS Oscillator mode |
| | | | — | 3.7 | TBD | mA | 5.0 | |

**Legend:** TBD = To Be Determined

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail to rail; all I/O pins tri-stated, pulled to VDD; $\overline{\text{MCLR}}$ = VDD; WDT disabled.

**2:** The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

**3:** The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral ∆ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

**4:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

## 15.2 DC Characteristics: PIC12F683-I (Industrial) (Continued)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|---|---|---|---|---|---|---|---|---|
| Param No. | Sym | Device Characteristics | Min | Typ† | Max | Units | Conditions | |
| | | | | | | | VDD | Note |
| D020 | IPD | **Power-down Base Current**[(4)] | — | 0.00099 | TBD | N/A | 2.0 | WDT, BOD, Comparator, VREF and T1OSC disabled |
| | | | — | 0.0012 | TBD | N/A | 3.0 | |
| | | | — | 0.0029 | TBD | N/A | 5.0 | |
| D021 | | | — | 1.8 | TBD | µA | 2.0 | WDT Current**[(3)]** |
| | | | — | 2.7 | TBD | µA | 3.0 | |
| | | | — | 8.4 | TBD | µA | 5.0 | |
| D022 | | | — | 58 | TBD | µA | 3.0 | BOD Current**[(3)]** |
| | | | — | 109 | TBD | µA | 5.0 | |
| D023 | | | — | 18 | TBD | µA | 2.0 | Comparator Current**[(3)]** |
| | | | — | 28 | TBD | µA | 3.0 | |
| | | | — | 60 | TBD | µA | 5.0 | |
| D024 | | | — | 58 | TBD | µA | 2.0 | CVREF Current**[(3)]** |
| | | | — | 85 | TBD | µA | 3.0 | |
| | | | — | 138 | TBD | µA | 5.0 | |
| D025 | | | — | 7.0 | TBD | µA | 2.0 | T1OSC Current**[(3)]** |
| | | | — | 8.6 | TBD | µA | 3.0 | |
| | | | — | 10 | TBD | µA | 5.0 | |
| D026 | | | — | 1.2 | TBD | nA | 3.0 | A/D Current**[(3)]** |
| | | | — | 0.0029 | TBD | µA | 5.0 | |

**Legend:** TBD = To Be Determined

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail to rail; all I/O pins tri-stated, pulled to VDD; $\overline{\text{MCLR}}$ = VDD; WDT disabled.

**2:** The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

**3:** The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral ∆ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

**4:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

## 15.3 DC Characteristics: PIC12F683-E (Extended)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +125°C for extended | | | | | |
|---|---|---|---|---|---|---|---|---|
| Param No. | Sym | Device Characteristics | Min | Typ† | Max | Units | Conditions | |
| | | | | | | | VDD | Note |
| D010E | IDD | **Supply Current**(1,2) | — | 9 | TBD | µA | 2.0 | FOSC = 32 kHz<br>LP Oscillator mode |
| | | | — | 18 | TBD | µA | 3.0 | |
| | | | — | 35 | TBD | µA | 5.0 | |
| D011E | | | — | 110 | TBD | µA | 2.0 | FOSC = 1 MHz<br>XT Oscillator mode |
| | | | — | 190 | TBD | µA | 3.0 | |
| | | | — | 330 | TBD | µA | 5.0 | |
| D012E | | | — | 220 | TBD | µA | 2.0 | FOSC = 4 MHz<br>XT Oscillator mode |
| | | | — | 370 | TBD | µA | 3.0 | |
| | | | — | 0.6 | TBD | mA | 5.0 | |
| D013E | | | — | 70 | TBD | µA | 2.0 | FOSC = 1 MHz<br>EC Oscillator mode |
| | | | — | 140 | TBD | µA | 3.0 | |
| | | | — | 260 | TBD | µA | 5.0 | |
| D014E | | | — | 180 | TBD | µA | 2.0 | FOSC = 4 MHz<br>EC Oscillator mode |
| | | | — | 320 | TBD | µA | 3.0 | |
| | | | — | 580 | TBD | µA | 5.0 | |
| D015E | | | — | 10 | TBD | µA | 2.0 | FOSC = 31 kHz<br>INTRC mode |
| | | | — | 25 | TBD | µA | 3.0 | |
| | | | — | 40 | TBD | µA | 5.0 | |
| D016E | | | — | 340 | TBD | µA | 2.0 | FOSC = 4 MHz<br>INTOSC mode |
| | | | — | 500 | TBD | µA | 3.0 | |
| | | | — | 0.8 | TBD | mA | 5.0 | |
| D017E | | | — | 250 | TBD | µA | 2.0 | FOSC = 4 MHz<br>EXTRC mode |
| | | | — | 375 | TBD | µA | 3.0 | |
| | | | — | 750 | TBD | µA | 5.0 | |
| D018E | | | — | 3.0 | TBD | mA | 4.5 | FOSC = 20 MHz<br>HS Oscillator mode |
| | | | — | 3.7 | TBD | mA | 5.0 | |

**Legend:** TBD = To Be Determined

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail to rail; all I/O pins tri-stated, pulled to VDD; $\overline{\text{MCLR}}$ = VDD; WDT disabled.

**2:** The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

**3:** The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

**4:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

## 15.3 DC Characteristics: PIC12F683-E (Extended) (Continued)

| **DC CHARACTERISTICS** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +125°C for extended | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Device Characteristics** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** | |
| | | | | | | | **VDD** | **Note** |
| D020E | IPD | **Power-down Base Current**[(4)] | — | 0.99 | TBD | nA | 2.0 | WDT, BOD, Comparator, VREF and T1OSC disabled |
| | | | — | 1.2 | TBD | nA | 3.0 | |
| | | | — | 2.9 | TBD | nA | 5.0 | |
| D021E | | | — | 1.8 | TBD | μA | 2.0 | WDT Current[(3)] |
| | | | — | 2.7 | TBD | μA | 3.0 | |
| | | | — | 8.4 | TBD | μA | 5.0 | |
| D022E | | | — | 58 | TBD | μA | 3.0 | BOD Current[(3)] |
| | | | — | 109 | TBD | μA | 5.0 | |
| D023E | | | — | 18 | TBD | μA | 2.0 | Comparator Current[(3)] |
| | | | — | 28 | TBD | μA | 3.0 | |
| | | | — | 60 | TBD | μA | 5.0 | |
| D024E | | | — | 58 | TBD | μA | 2.0 | CVREF Current[(3)] |
| | | | — | 85 | TBD | μA | 3.0 | |
| | | | — | 138 | TBD | μA | 5.0 | |
| D025E | | | — | 7.0 | TBD | μA | 2.0 | T1OSC Current[(3)] |
| | | | — | 8.6 | TBD | μA | 3.0 | |
| | | | — | 10 | TBD | μA | 5.0 | |
| D026E | | | — | 1.2 | TBD | μA | 3.0 | A/D Current[(3)] |
| | | | — | 0.0029 | TBD | μA | 5.0 | |

**Legend:** TBD = To Be Determined

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail to rail; all I/O pins tri-stated, pulled to VDD; $\overline{\text{MCLR}}$ = VDD; WDT disabled.

**2:** The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

**3:** The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

**4:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

## 15.4 DC Characteristics: PIC12F683-I (Industrial) PIC12F683-E (Extended)

| DC CHARACTERISTICS | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | VIL | **Input Low Voltage** | | | | | |
| | | I/O port: | | | | | |
| D030 | | with TTL buffer | Vss | — | 0.8 | V | 4.5V ≤ VDD ≤ 5.5V |
| D030A | | | Vss | — | 0.15 VDD | V | Otherwise |
| D031 | | with Schmitt Trigger buffer | Vss | — | 0.2 VDD | V | Entire range |
| **TBD | | Ultra Low-Power | — | — | — | — | |
| D032 | | $\overline{\text{MCLR}}$, OSC1 (RC mode) | VSS | — | 0.2 VDD | V | |
| D033 | | OSC1 (XT and LP modes)(1) | VSS | — | 0.3 | V | |
| D033A | | OSC1 (HS mode)(1) | VSS | — | 0.3 VDD | V | |
| | VIH | **Input High Voltage** | | | | | |
| | | I/O port: | | — | | | |
| D040 | | with TTL buffer | 2.0 | — | VDD | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | (0.25 VDD + 0.8) | — | VDD | V | Otherwise |
| D041 | | with Schmitt Trigger buffer | 0.8 VDD | — | VDD | V | Entire range |
| TBD | | Ultra Low-Power | — | — | — | — | |
| D042 | | $\overline{\text{MCLR}}$ | 0.8 VDD | — | VDD | V | |
| D043 | | OSC1 (XT and LP modes) | 1.6 | — | VDD | V | **(Note 1)** |
| D043A | | OSC1 (HS mode) | 0.7 VDD | — | VDD | V | **(Note 1)** |
| D043B | | OSC1 (RC mode) | 0.9 VDD | — | VDD | V | |
| D070 | IPUR | **GPIO Weak Pull-up Current** | 50* | 250 | 400* | μA | VDD = 5.0V, VPIN = VSS |
| | IIL | **Input Leakage Current(2)** | | | | | |
| D060 | | I/O port | — | ± 0.1 | ±1 | μA | VSS ≤ VPIN ≤ VDD,<br>Pin at hi-impedance |
| D061 | | $\overline{\text{MCLR}}$(3) | — | ± 0.1 | ±5 | μA | VSS ≤ VPIN ≤ VDD |
| D063 | | OSC1 | — | ± 0.1 | ±5 | μA | VSS ≤ VPIN ≤ VDD, XT, HS and LP oscillator configuration |
| | VOL | **Output Low Voltage** | | | | | |
| D080 | | I/O port | — | — | 0.6 | V | IOL = 8.5 mA, VDD = 4.5V (Ind.) |
| D083 | | OSC2/CLKOUT (RC mode) | — | — | 0.6 | V | IOL = 1.6 mA, VDD = 4.5V (Ind.)<br>IOL = 1.2 mA, VDD = 4.5V (Ext.) |
| | VOH | **Output High Voltage** | | | | | |
| D090 | | I/O port | VDD – 0.7 | — | — | V | IOH = -3.0 mA, VDD = 4.5V (Ind.) |
| D092 | | OSC2/CLKOUT (RC mode) | VDD – 0.7 | — | — | V | IOH = -1.3 mA, VDD = 4.5V (Ind.)<br>IOH = -1.0 mA, VDD = 4.5V (Ext.) |

**Legend:** TBD = To Be Determined

* These parameters are characterized but not tested.

† Data in ‘Typ’ column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

**2:** Negative current is defined as current sourced by the pin.

**3:** The leakage current on the $\overline{\text{MCLR}}$ 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.

**4:** See **Section 10.4.1 “Using the Data EEPROM”** for additional information.

## 15.4 DC Characteristics: PIC12F683-I (Industrial) PIC12F683-E (Extended) (Continued)

| **DC CHARACTERISTICS** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| D100 | IULP | **Ultra Low-Power Wake-up Current** | — | 200 | — | nA | |
| | | **Capacitive Loading Specs on Output Pins** | | | | | |
| D100 | COSC2 | OSC2 pin | — | — | 15* | pF | In XT, HS and LP modes when external clock is used to drive OSC1 |
| D101 | CIO | All I/O pins | — | — | 50* | pF | |
| | | **Data EEPROM Memory** | | | | | |
| D120 | ED | Byte Endurance | 100K | 1M | — | E/W | -40°C ≤ TA ≤ +85°C |
| D120A | ED | Byte Endurance | 10K | 100K | — | E/W | +85°C ≤ TA ≤ +125°C |
| D121 | VDRW | VDD for Read/Write | VMIN | — | 5.5 | V | Using EECON1 to read/write<br>VMIN = Minimum operating voltage |
| D122 | TDEW | Erase/Write Cycle Time | — | 5 | 6 | ms | |
| D123 | TRETD | Characteristic Retention | 40 | — | — | Year | Provided no other specifications are violated |
| D124 | TREF | Number of Total Erase/Write Cycles before Refresh(4) | 1M | 10M | — | E/W | -40°C ≤ TA ≤ +85°C |
| | | **Program Flash Memory** | | | | | |
| D130 | EP | Cell Endurance | 10K | 100K | — | E/W | -40°C ≤ TA ≤ +85°C |
| D130A | ED | Cell Endurance | 1K | 10K | — | E/W | +85°C ≤ TA ≤ +125°C |
| D131 | VPR | VDD for Read | VMIN | — | 5.5 | V | VMIN = Minimum operating voltage |
| D132 | VPEW | VDD for Erase/Write | 4.5 | — | 5.5 | V | |
| D133 | TPEW | Erase/Write cycle time | — | 2 | 2.5 | ms | |
| D134 | TRETD | Characteristic Retention | 40 | — | — | Year | Provided no other specifications are violated |

**Legend:** TBD = To Be Determined

* These parameters are characterized but not tested.

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

**2:** Negative current is defined as current sourced by the pin.

**3:** The leakage current on the $\overline{\text{MCLR}}$ 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.

**4:** See **Section 10.4.1 "Using the Data EEPROM"** for additional information.

## 15.5 Timing Parameter Symbology

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

1. TppS2ppS
2. TppS

| | | | |
|---|---|---|---|
| **T** | | | |
| F | Frequency | T | Time |

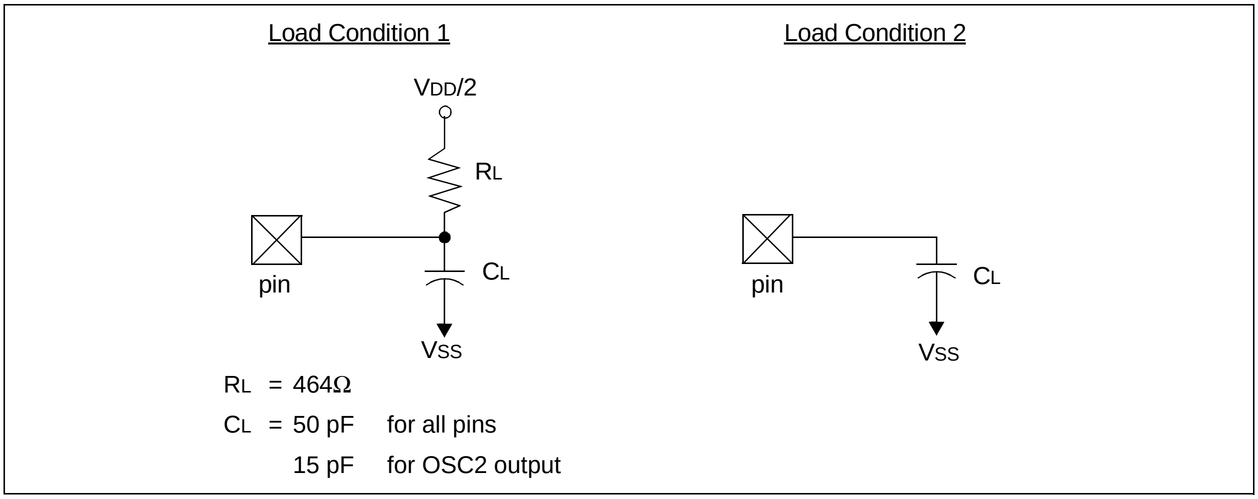
Lowercase letters (pp) and their meanings:

| | | | |
|---|---|---|---|
| **pp** | | | |
| cc | CCP1 | osc | OSC1 |
| ck | CLKOUT | rd | $\overline{\text{RD}}$ |
| cs | $\overline{\text{CS}}$ | rw | $\overline{\text{RD}}$ or $\overline{\text{WR}}$ |
| di | SDI | sc | SCK |
| do | SDO | ss | $\overline{\text{SS}}$ |
| dt | Data in | t0 | T0CKI |
| io | I/O port | t1 | T1CKI |
| mc | $\overline{\text{MCLR}}$ | wr | $\overline{\text{WR}}$ |

Uppercase letters and their meanings:

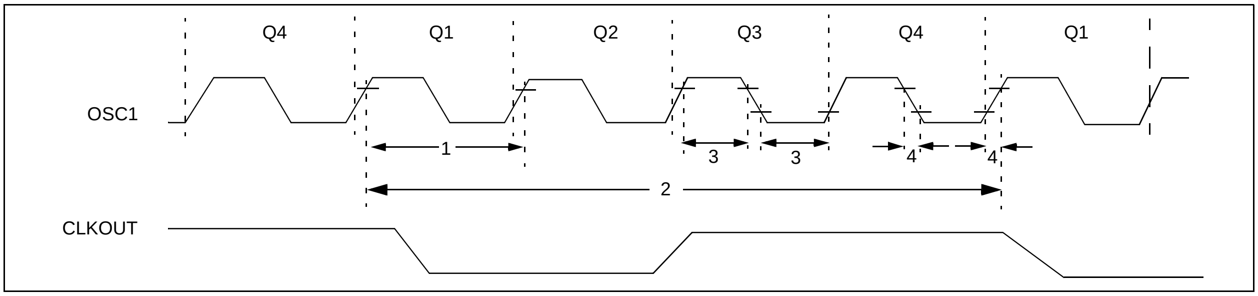
| | | | |
|---|---|---|---|
| **S** | | | |
| F | Fall | P | Period |
| H | High | R | Rise |
| I | Invalid (High-impedance) | V | Valid |
| L | Low | Z | High-impedance |

**FIGURE 15-2: LOAD CONDITIONS**



## 15.6 AC Characteristics: PIC12F683 (Industrial, Extended)

### FIGURE 15-3: EXTERNAL CLOCK TIMING



### TABLE 15-1: EXTERNAL CLOCK TIMING REQUIREMENTS

**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40°C ≤ TA ≤ +125°C

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | FOSC | External CLKIN Frequency[1] | DC | — | 37 | kHz | LP Oscillator mode |
| | | | DC | — | 4 | MHz | XT Oscillator mode |
| | | | DC | — | 20 | MHz | HS Oscillator mode |
| | | | DC | — | 20 | MHz | EC Oscillator mode |
| | | Oscillator Frequency[1] | 5 | — | 37 | kHz | LP Oscillator mode |
| | | | — | 4 | — | MHz | INTOSC mode |
| | | | DC | — | 4 | MHz | RC Oscillator mode |
| | | | 0.1 | — | 4 | MHz | XT Oscillator mode |
| | | | 1 | — | 20 | MHz | HS Oscillator mode |
| 1 | TOSC | External CLKIN Period[1] | 27 | — | — | µs | LP Oscillator mode |
| | | | 50 | — | — | ns | HS Oscillator mode |
| | | | 50 | — | — | ns | EC Oscillator mode |
| | | | 250 | — | — | ns | XT Oscillator mode |
| | | Oscillator Period[1] | 27 | — | 200 | µs | LP Oscillator mode |
| | | | — | 250 | — | ns | INTOSC mode |
| | | | 250 | — | — | ns | RC Oscillator mode |
| | | | 250 | — | 10,000 | ns | XT Oscillator mode |
| | | | 50 | — | 1,000 | ns | HS Oscillator mode |
| 2 | TCY | Instruction Cycle Time[1] | 200 | TCY | DC | ns | TCY = 4/FOSC |
| 3 | TosL,<br>TosH | External CLKIN (OSC1) High<br>External CLKIN Low | 2* | — | — | µs | LP oscillator, TOSC L/H duty cycle |
| | | | 20* | — | — | ns | HS oscillator, TOSC L/H duty cycle |
| | | | 100 * | — | — | ns | XT oscillator, TOSC L/H duty cycle |
| 4 | TosR,<br>TosF | External CLKIN Rise<br>External CLKIN Fall | — | — | 50* | ns | LP oscillator |
| | | | — | — | 25* | ns | XT oscillator |
| | | | — | — | 15* | ns | HS oscillator |

* These parameters are characterized but not tested.

† Data in ‘Typ’ column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Instruction cycle period (TCY) equals four times the input oscillator time base period. 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. All devices are tested to operate at ‘min’ values with an external clock applied to OSC1 pin. When an external clock input is used, the ‘max’ cycle time limit is ‘DC’ (no clock) for all devices.

### TABLE 15-2: PRECISION INTERNAL OSCILLATOR PARAMETERS

**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40°C ≤ TA ≤ +125°C

| Param No. | Sym | Characteristic | Freq Tolerance | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| F10 | FOSC | Internal Calibrated INTOSC Frequency(1) | ±1% | — | 8.00 | — | MHz | VDD and Temperature (TBD) |
| | | | ±2% | — | 8.00 | — | MHz | 2.5V ≤ VDD ≤ 5.5V<br>0°C ≤ TA ≤ +85°C |
| | | | ±5% | — | 8.00 | — | MHz | 2.0V ≤ VDD ≤ 5.5V<br>-40°C ≤ TA ≤ +85°C (Ind.)<br>-40°C ≤ TA ≤ +125°C (Ext.) |
| F14 | TIOSCST | Oscillator Wake-up from Sleep Start-up Time* | — | — | TBD | TBD | µs | VDD = 2.0V, -40°C to +85°C |
| | | | — | — | TBD | TBD | µs | VDD = 3.0V, -40°C to +85°C |
| | | | — | — | TBD | TBD | µs | VDD = 5.0V, -40°C to +85°C |

**Legend:** TBD = To Be Determined

* These parameters are characterized but not tested.

† Data in 'Typ' column is at 5.0V, 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 µF and 0.01 µF values in parallel are recommended.

FIGURE 15-4: CLKOUT AND I/O TIMING

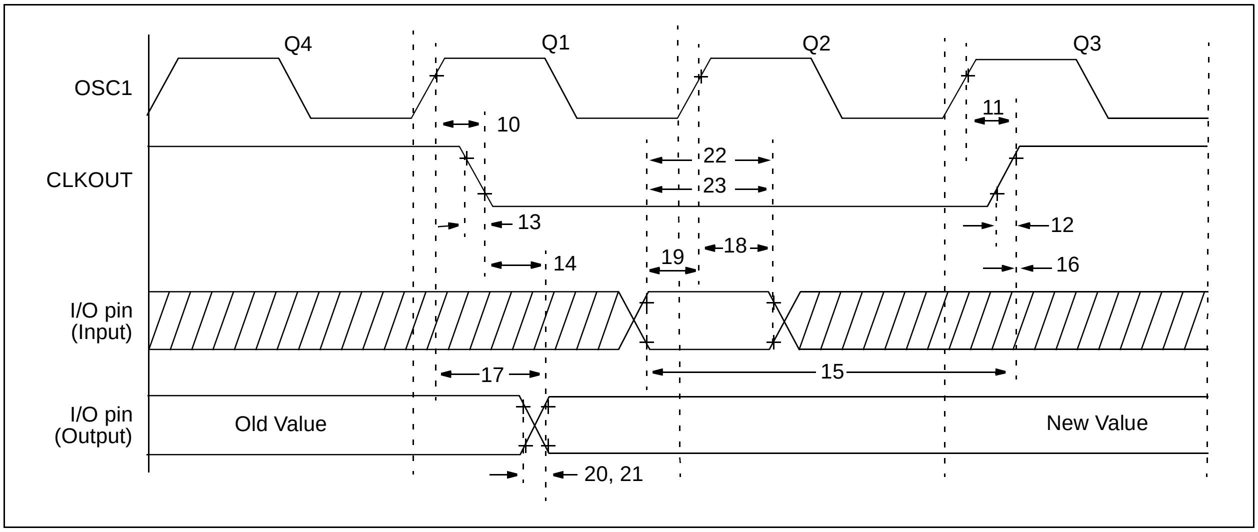


TABLE 15-3: CLKOUT AND I/O TIMING REQUIREMENTS

**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40°C ≤ TA ≤ +125°C

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 10 | TosH2ckL | OSC1↑ to CLOUT↓ | — | 75 | 200 | ns | **(Note 1)** |
| 11 | TosH2ckH | OSC1↑ to CLOUT↑ | — | 75 | 200 | ns | **(Note 1)** |
| 12 | TckR | CLKOUT Rise Time | — | 35 | 100 | ns | **(Note 1)** |
| 13 | TckF | CLKOUT Fall Time | — | 35 | 100 | ns | **(Note 1)** |
| 14 | TckL2ioV | CLKOUT↓ to Port Out Valid | — | — | 20 | ns | **(Note 1)** |
| 15 | TioV2ckH | Port In Valid before CLKOUT↑ | TOSC + 200 ns | — | — | ns | **(Note 1)** |
| 16 | TckH2ioI | Port In Hold after CLKOUT↑ | 0 | — | — | ns | **(Note 1)** |
| 17 | TosH2ioV | OSC1↑ (Q1 cycle) to Port Out Valid | — | 50 | 150* | ns | |
| | | | — | — | 300 | ns | |
| 18 | TosH2ioI | OSC1↑ (Q2 cycle) to Port Input Invalid (I/O in hold time) | 100 | — | — | ns | |
| 19 | TioV2osH | Port Input Valid to OSC1↑ (I/O in setup time) | 0 | — | — | ns | |
| 20 | TioR | Port Output Rise Time | — | 10 | 40 | ns | |
| 21 | TioF | Port Output Fall Time | — | 10 | 40 | ns | |
| 22 | Tinp | INT pin High or Low Time | 25 | — | — | ns | |
| 23 | Trbp | GPIO Change INT High or Low Time | TCY | — | — | ns | |

* These parameters are characterized but not tested.

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated.

**Note 1:** Measurements are taken in RC mode where CLKOUT output is 4 x TOSC.

**FIGURE 15-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**

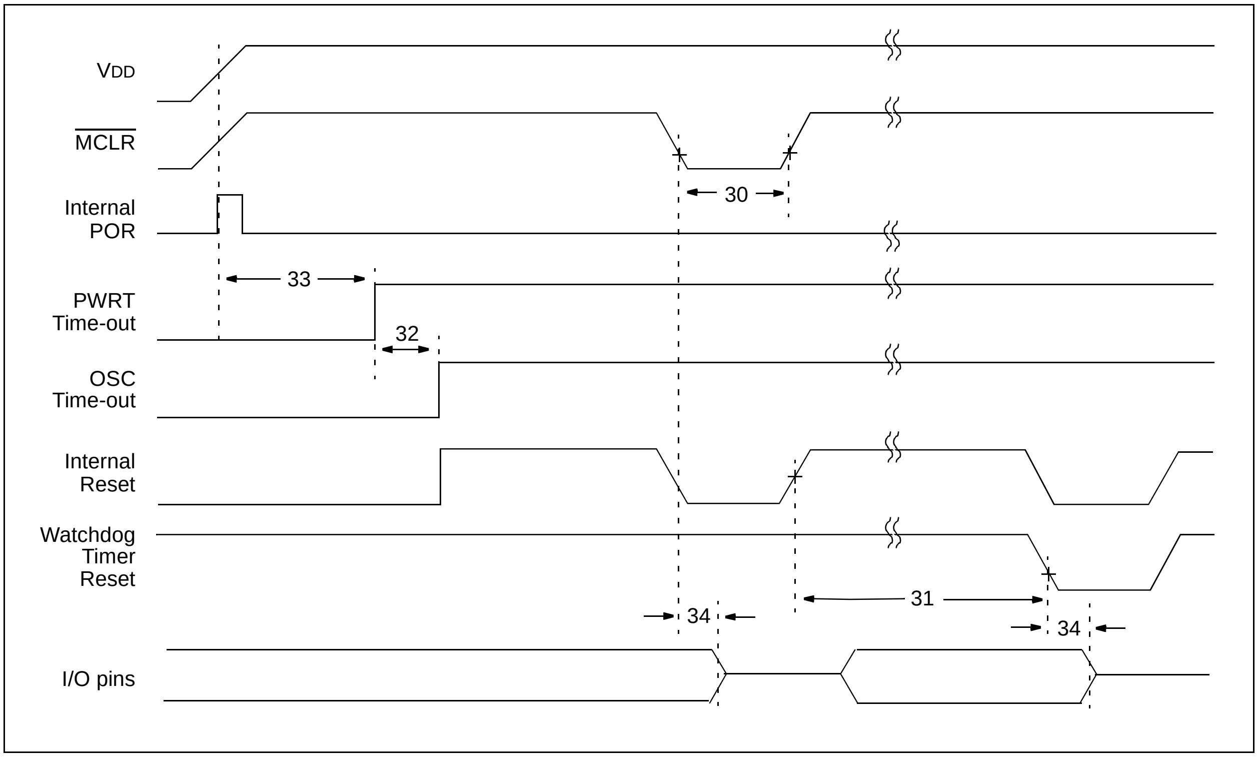
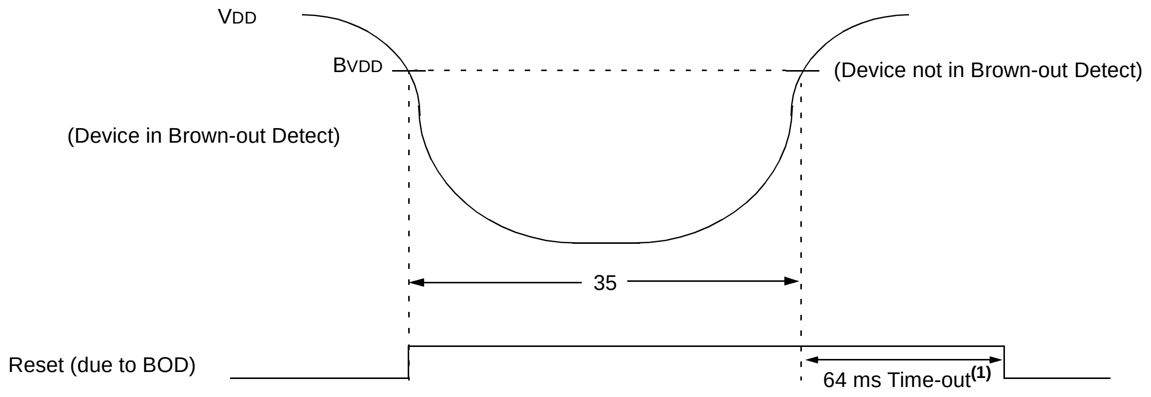


**FIGURE 15-6: BROWN-OUT DETECT TIMING AND CHARACTERISTICS**


VDD
BVDD
(Device not in Brown-out Detect)
(Device in Brown-out Detect)
35
Reset (due to BOD)
64 ms Time-out(1)

**Note 1:** 64 ms delay only if $\overline{\text{PWRTE}}$ bit in Configuration Word register is programmed to '0'.

### TABLE 15-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT DETECT REQUIREMENTS

**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40°C ≤ TA ≤ +125°C

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 30 | TMCL | $\overline{\text{MCLR}}$ Pulse Width (low) | 2<br>11 | —<br>18 | —<br>24 | µs<br>ms | VDD = 5V, -40°C to +85°C<br>Extended temperature |
| 31 | TWDT | Watchdog Timer Time-out Period (no prescaler) | 10<br>10 | 17<br>17 | 25<br>30 | ms<br>ms | VDD = 5V, -40°C to +85°C<br>Extended temperature |
| 32 | TOST | Oscillation Start-up Timer Period | — | 1024TOSC | — | — | TOSC = OSC1 period |
| 33* | TPWRT | Power-up Timer Period | 28*<br>TBD | 64<br>TBD | 132*<br>TBD | ms<br>ms | VDD = 5V, -40°C to +85°C<br>Extended Temperature |
| 34 | TIOZ | I/O High-impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset | — | — | 2.0 | µs | |
| | BVDD | Brown-out Detect Voltage | 2.025 | — | 2.175 | V | |
| 35 | TBOD | Brown-out Detect Pulse Width | 100* | — | — | µs | VDD ≤ BVDD (D005) |
| 36 | TR | Brown-out Detect Response Time | — | — | 1 | µs | |
| 37 | TRD | Brown-out Detect Retriggerable Delay Time | 5 | 10 | 15 | µs | |

**Legend:** TBD = To Be Determined

* These parameters are characterized but not tested.

† Data in 'Typ' column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 15-7: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

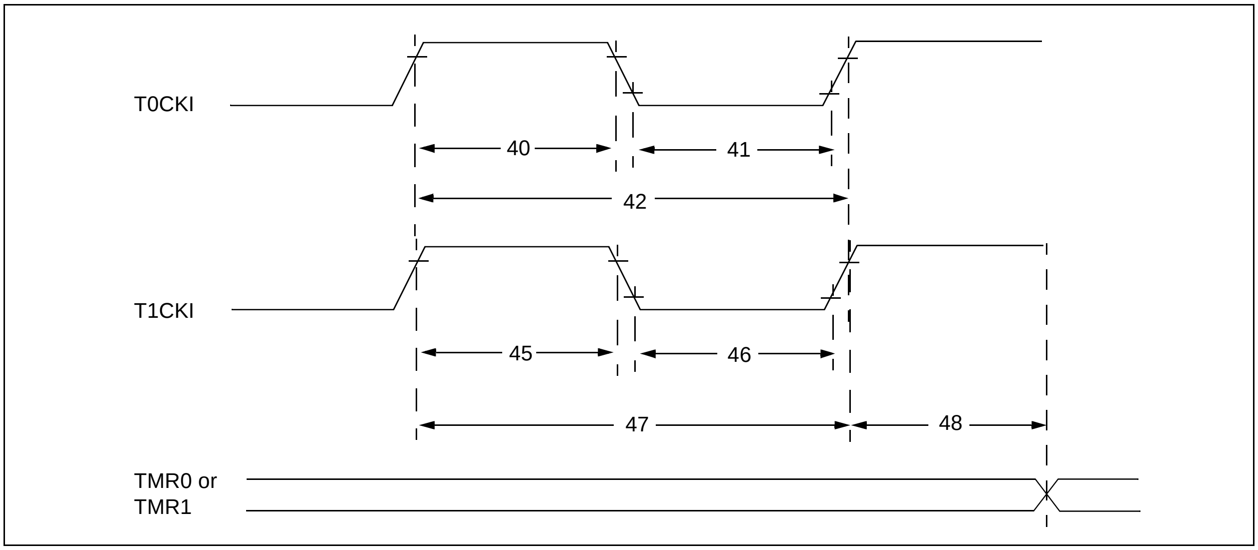


TABLE 15-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

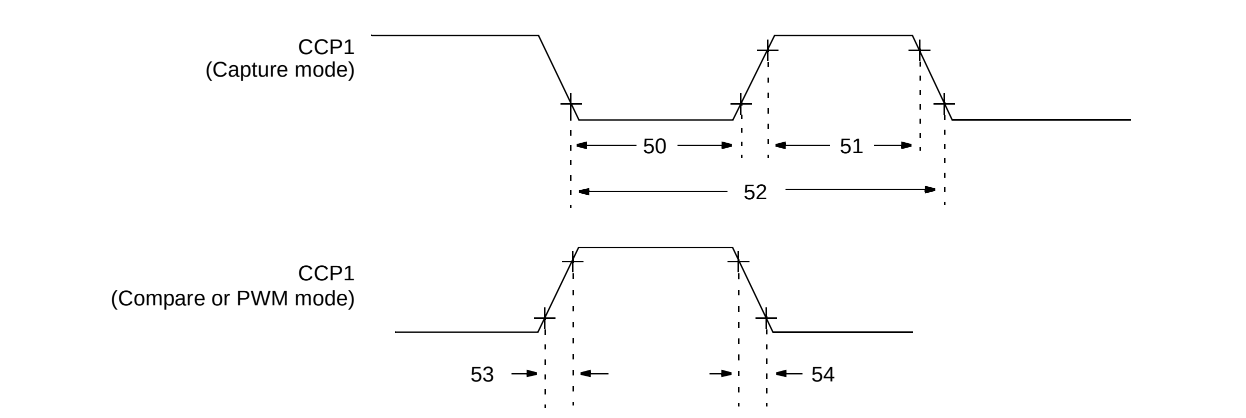
**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40°C ≤ TA ≤ +125°C

| Param No. | Sym | Characteristic | | Min | Typ† | 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 | | Greater of:<br>20 or $\frac{T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (2, 4, ..., 256) |
| 45* | Tt1H | T1CKI High Time | Synchronous, No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | Synchronous, with Prescaler | 15 | — | — | ns | |
| | | | Asynchronous | 30 | — | — | ns | |
| 46* | Tt1L | T1CKI Low Time | Synchronous, No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | Synchronous, with Prescaler | 15 | — | — | ns | |
| | | | Asynchronous | 30 | — | — | ns | |
| 47* | Tt1P | T1CKI Input Period | Synchronous | Greater of:<br>30 or $\frac{T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | 60 | — | — | ns | |
| | Ft1 | Timer1 Oscillator Input Frequency Range (oscillator enabled by setting bit T1OSCEN) | | DC | — | 200* | kHz | |
| 48 | TCKEZtmr1 | Delay from External Clock Edge to Timer Increment | | 2 TOSC* | — | 7 TOSC* | — | |

* These parameters are characterized but not tested.

† Data in 'Typ' column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

### FIGURE 15-8: CAPTURE/COMPARE/PWM TIMINGS (CCP)



**Note:** Refer to Figure 15-2 for load conditions.

### TABLE 15-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP)

**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40°C ≤ TA ≤ +125°C

| Param No. | Symbol | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 50* | TccL | CCP1 Input Low Time | No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | With Prescaler | 20 | — | — | ns | |
| 51* | TccH | CCP1 Input High Time | No Prescaler | 0.5 TCY + 20 | — | — | ns | |
| | | | With Prescaler | 20 | — | — | ns | |
| 52* | TccP | CCP1 Input Period | | $\frac{3\ T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1, 4 or 16) |
| 53* | TccR | CCP1 Output Rise Time | | — | 25 | 50 | ns | |
| 54* | TccF | CCP1 Output Fall Time | | — | 25 | 45 | ns | |

* These parameters are characterized but not tested.

† Data in ‘Typ’ column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

### TABLE 15-7: COMPARATOR SPECIFICATIONS

| Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | | | |
|---|---|---|---|---|---|---|
| **Sym** | **Characteristics** | **Min** | **Typ** | **Max** | **Units** | **Comments** |
| VOS | Input Offset Voltage | — | ± 5.0 | ± 10 | mV | |
| VCM | Input Common Mode Voltage | 0 | — | VDD - 1.5 | V | |
| CMRR | Common Mode Rejection Ratio | +55* | — | — | db | |
| TRT | Response Time[(1)] | — | 150 | 400* | ns | |
| TMC2COV | Comparator Mode Change to Output Valid | — | — | 10* | μs | |

* These parameters are characterized but not tested.

**Note 1:** Response time measured with one comparator input at (VDD - 1.5)/2 while the other input transitions from VSS to VDD – 1.5V.

### TABLE 15-8: COMPARATOR VOLTAGE REFERENCE SPECIFICATIONS

| Voltage Reference Specifications | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | |
|---|---|---|---|---|---|---|
| **Sym.** | **Characteristics** | **Min** | **Typ** | **Max** | **Units** | **Comments** |
| | Resolution | —<br>— | VDD/24*<br>VDD/32 | —<br>— | LSb<br>LSb | Low Range (VRR = 1)<br>High Range (VRR = 0) |
| | Absolute Accuracy | —<br>— | —<br>— | ± 1/4*<br>± 1/2* | LSb<br>LSb | Low Range (VRR = 1)<br>High Range (VRR = 0) |
| | Unit Resistor Value (R) | — | 2k* | — | Ω | |
| | Settling Time[(1)] | — | — | 10* | μs | |

* These parameters are characterized but not tested.

**Note 1:** Settling time measured while VRR = 1 and VR<3:0> transitions from '0000' to '1111'.

### TABLE 15-9: PIC12F683 A/D CONVERTER CHARACTERISTICS

**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40°C ≤ TA ≤ +125°C

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| A01 | NR | Resolution | — | — | 10 | bit | |
| A02 | EABS | Total Absolute Error*(1) | — | — | ±1 | LSb | VREF = 5.0V |
| A03 | EIL | Integral Error | — | — | ±1 | LSb | VREF = 5.0V |
| A04 | EDL | Differential Error | — | — | ±1 | LSb | No missing codes to 10 bits<br>VREF = 5.0V |
| A05 | EFS | Full-scale Range | 2.2* | — | 5.5* | V | |
| A06 | EOFF | Offset Error | — | — | ±1 | LSb | VREF = 5.0V |
| A07 | EGN | Gain Error | — | — | ±1 | LSb | VREF = 5.0V |
| A10 | — | Monotonicity | — | guaranteed(2) | — | — | VSS ≤ VAIN ≤ VREF+ |
| A20<br>A20A | VREF | Reference Voltage | 2.2<br>2.5 | — | VDD + 0.3<br>VDD + 0.3 | V | 0°C ≤ TA ≤ +125°C<br>Absolute limits to ensure 10-bit accuracy |
| A25 | VAIN | Analog Input Voltage | VSS | — | VREF | V | |
| A30 | ZAIN | Recommended Impedance of Analog Voltage Source | — | — | 10 | KΩ | |
| A50 | IREF | VREF Input Current*(3) | 10<br>— | —<br>— | 1000<br>10 | µA<br>µA | During VAIN acquisition. Based on differential of VHOLD to VAIN.<br>During A/D conversion cycle. |

* These parameters are characterized but not tested.

† Data in 'Typ' column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

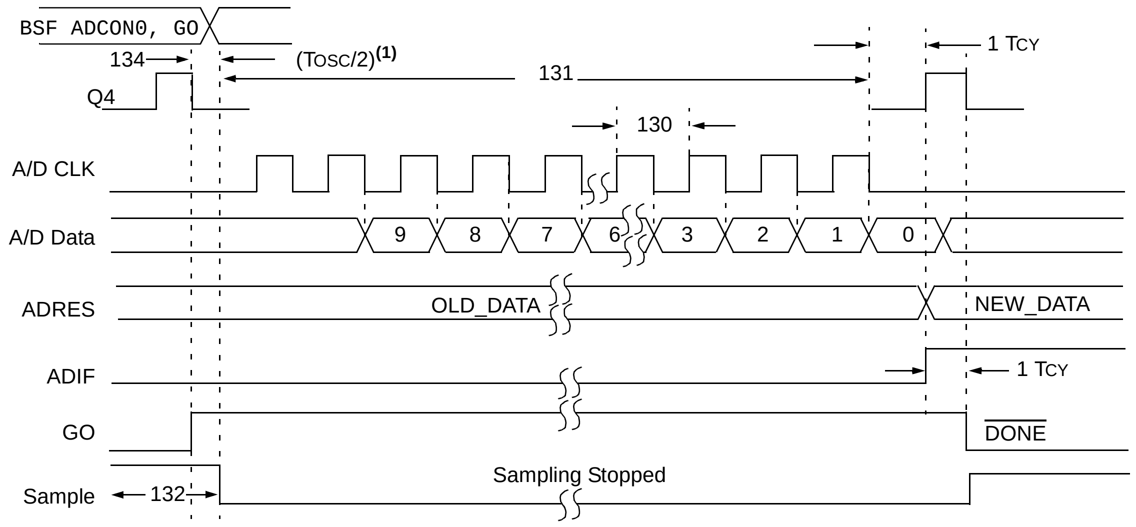
**Note 1:** Total Absolute Error includes integral, differential, offset and gain errors.

**2:** The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

**3:** VREF current is from external VREF or VDD pin, whichever is selected as reference input.

**4:** When A/D is off, it will not consume any current other than leakage current. The power-down current specification includes any such leakage from the A/D module.

**FIGURE 15-9: PIC12F683 A/D CONVERSION TIMING (NORMAL MODE)**



**Note 1:** If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

## TABLE 15-10: PIC12F683 A/D CONVERSION REQUIREMENTS

**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40°C ≤ TA ≤ +125°C

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 130 | TAD | A/D Clock Period | 1.6<br>3.0* | —<br>— | —<br>— | µs<br>µs | TOSC based, VREF ≥ 3.0V<br>TOSC based, VREF full range |
| 130 | TAD | A/D Internal RC Oscillator Period | 3.0*<br>2.0* | 6.0<br>4.0 | 9.0*<br>6.0* | µs<br>µs | ADCS<1:0> = 11 (RC mode)<br>At VDD = 2.5V<br>At VDD = 5.0V |
| 131 | TCNV | Conversion Time (not including Acquisition Time)(1) | — | 11 | — | TAD | Set GO bit to new data in A/D Result register |
| 132 | TACQ | Acquisition Time | | 11.5 | — | µs | |
| | | | 5* | — | — | µs | The minimum time is the amplifier settling time. This may be used if the “new” input voltage has not changed by more than 1 LSb (i.e., 4.1 mV @ 4.096V) from the last sampled voltage (as stored on CHOLD). |
| 134 | TGO | Q4 to A/D Clock Start | — | TOSC/2 | — | — | If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed. |

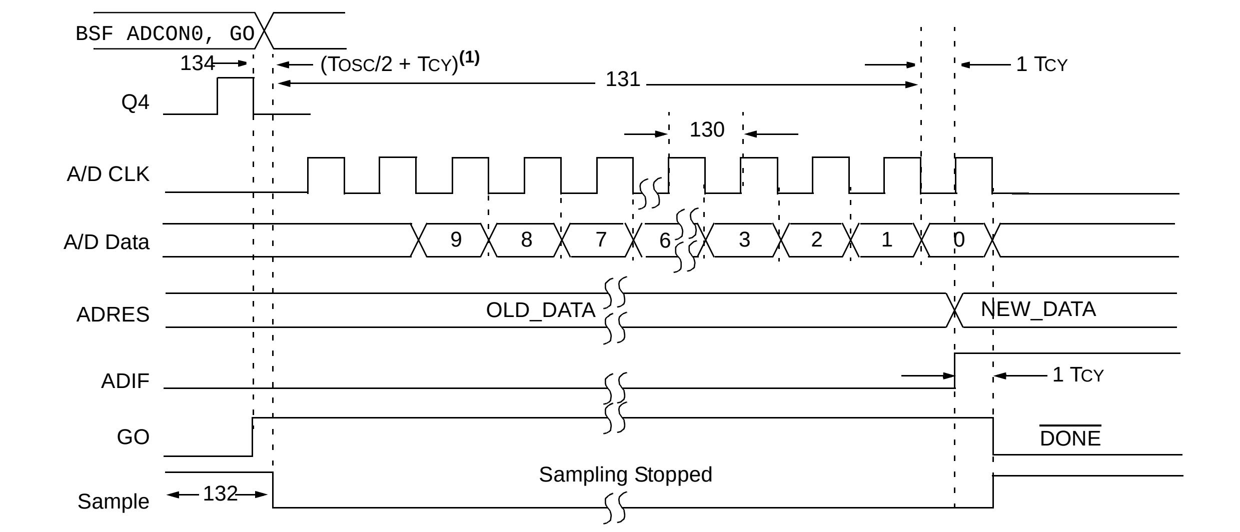
* These parameters are characterized but not tested.

† Data in ‘Typ’ column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** ADRESH and ADRESL registers may be read on the following TCY cycle.

**2:** See Table 9-1 for minimum conditions.

**FIGURE 15-10: PIC12F683 A/D CONVERSION TIMING (SLEEP MODE)**



**Note 1:** If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

**TABLE 15-11: PIC12F683 A/D CONVERSION REQUIREMENTS (SLEEP MODE)**

**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40°C ≤ TA ≤ +125°C

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 130 | TAD | A/D Internal RC Oscillator Period | <br>3.0*<br>2.0* | <br>6.0<br>4.0 | <br>9.0*<br>6.0* | <br>μs<br>μs | ADCS<1:0> = 11 (RC mode)<br>At VDD = 2.5V<br>At VDD = 5.0V |
| 131 | Tcnv | Conversion Time (not including Acquisition Time)(1) | — | 11 | — | TAD | |
| 132 | TACQ | Acquisition Time | **(2)**<br>5* | 11.5<br>— | —<br>— | μs<br>μs | <br>The minimum time is the amplifier settling time. This may be used if the “new” input voltage has not changed by more than 1 LSb (i.e., 4.1 mV @ 4.096V) from the last sampled voltage (as stored on CHOLD). |
| 134 | TGO | Q4 to A/D Clock Start | — | TOSC/2 + TCY | — | — | If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed. |

* These parameters are characterized but not tested.

† Data in ‘Typ’ column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** ADRES register may be read on the following TCY cycle.

**2:** See Table 9-1 for minimum conditions.

NOTES:

## 16.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

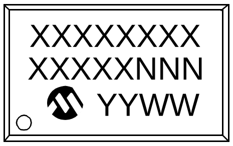
Graphs and Tables are not available at this time.

NOTES:

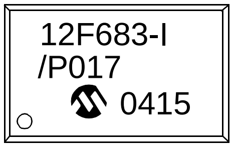
## 17.0 PACKAGING INFORMATION

### 17.1 Package Marking Information

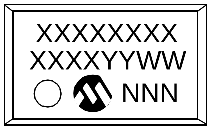
8-Lead PDIP (Skinny DIP)



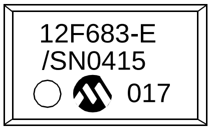
Example



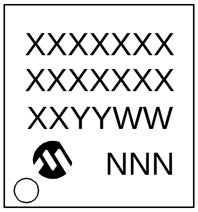
8-Lead SOIC



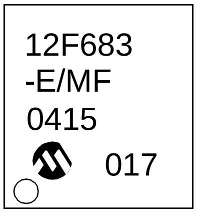
Example



8-Lead DFN-S



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 |

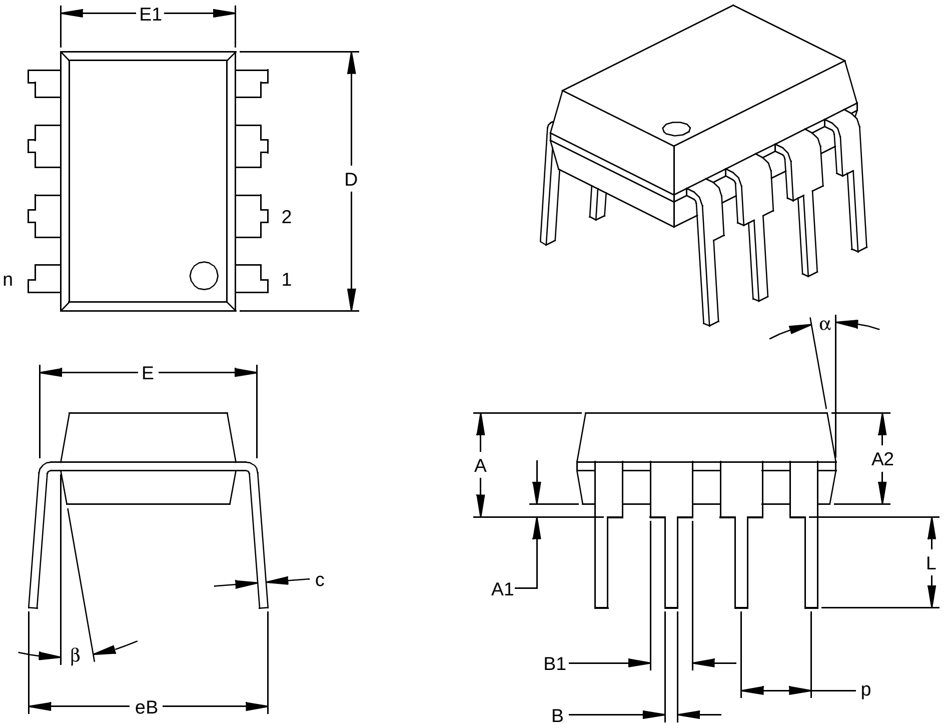
**Note**: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

* Standard PICmicro device marking consists of Microchip part number, year code, week code and traceability code. For PICmicro 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.

## 17.2 Package Details

The following sections give the technical details of the packages.

### 8-Lead Plastic Dual In-line (P) – 300 mil Body (PDIP)

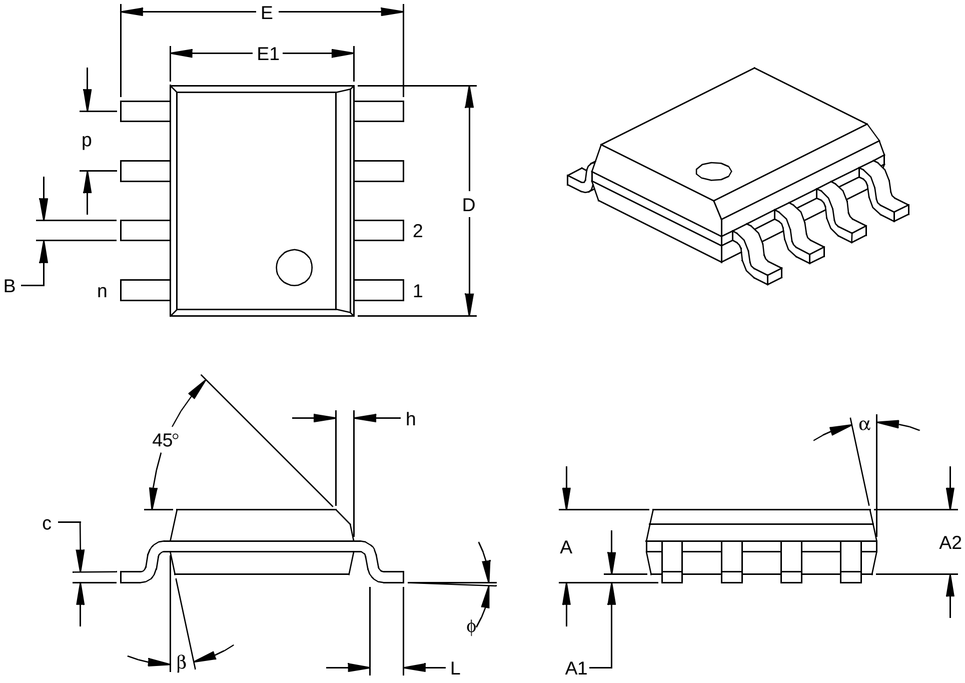


| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 8 | | | 8 | |
| Pitch | p | | .100 | | | 2.54 | |
| Top to Seating Plane | A | .140 | .155 | .170 | 3.56 | 3.94 | 4.32 |
| Molded Package Thickness | A2 | .115 | .130 | .145 | 2.92 | 3.30 | 3.68 |
| Base to Seating Plane | A1 | .015 | | | 0.38 | | |
| Shoulder to Shoulder Width | E | .300 | .313 | .325 | 7.62 | 7.94 | 8.26 |
| Molded Package Width | E1 | .240 | .250 | .260 | 6.10 | 6.35 | 6.60 |
| Overall Length | D | .360 | .373 | .385 | 9.14 | 9.46 | 9.78 |
| Tip to Seating Plane | L | .125 | .130 | .135 | 3.18 | 3.30 | 3.43 |
| Lead Thickness | c | .008 | .012 | .015 | 0.20 | 0.29 | 0.38 |
| Upper Lead Width | B1 | .045 | .058 | .070 | 1.14 | 1.46 | 1.78 |
| Lower Lead Width | B | .014 | .018 | .022 | 0.36 | 0.46 | 0.56 |
| Overall Row Spacing § | eB | .310 | .370 | .430 | 7.87 | 9.40 | 10.92 |
| Mold Draft Angle Top | α | 5 | 10 | 15 | 5 | 10 | 15 |
| Mold Draft Angle Bottom | β | 5 | 10 | 15 | 5 | 10 | 15 |

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-001
Drawing No. C04-018

## 8-Lead Plastic Small Outline (SN) – Narrow, 150 mil Body (SOIC)

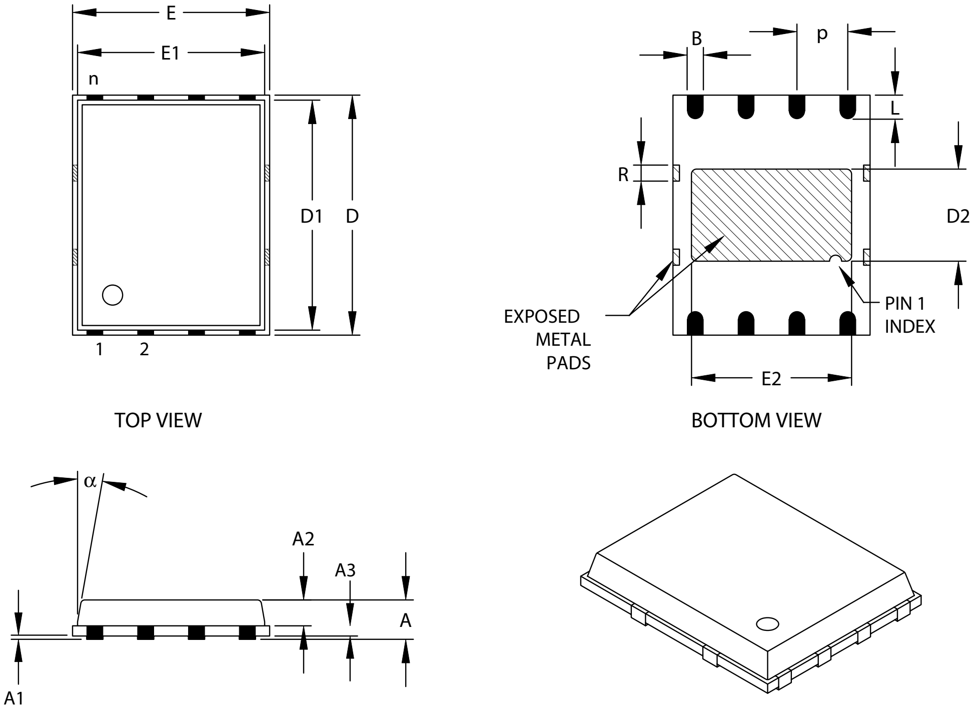


| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 8 | | | 8 | |
| Pitch | p | | .050 | | | 1.27 | |
| Overall Height | A | .053 | .061 | .069 | 1.35 | 1.55 | 1.75 |
| Molded Package Thickness | A2 | .052 | .056 | .061 | 1.32 | 1.42 | 1.55 |
| Standoff § | A1 | .004 | .007 | .010 | 0.10 | 0.18 | 0.25 |
| Overall Width | E | .228 | .237 | .244 | 5.79 | 6.02 | 6.20 |
| Molded Package Width | E1 | .146 | .154 | .157 | 3.71 | 3.91 | 3.99 |
| Overall Length | D | .189 | .193 | .197 | 4.80 | 4.90 | 5.00 |
| Chamfer Distance | h | .010 | .015 | .020 | 0.25 | 0.38 | 0.51 |
| Foot Length | L | .019 | .025 | .030 | 0.48 | 0.62 | 0.76 |
| Foot Angle | φ | 0 | 4 | 8 | 0 | 4 | 8 |
| Lead Thickness | c | .008 | .009 | .010 | 0.20 | 0.23 | 0.25 |
| Lead Width | B | .013 | .017 | .020 | 0.33 | 0.42 | 0.51 |
| Mold Draft Angle Top | α | 0 | 12 | 15 | 0 | 12 | 15 |
| Mold Draft Angle Bottom | β | 0 | 12 | 15 | 0 | 12 | 15 |

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-012
Drawing No. C04-057

## 8-Lead Plastic Dual Flat No Lead Package (MF) 6x5 mm Body (DFN-S) – Punch Singulated



| Units | | INCHES | | | MILLIMETERS* | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 8 | | | 8 | |
| Pitch | p | | .050 BSC | | | 1.27 BSC | |
| Overall Height | A | | .033 | .039 | | 0.85 | 1.00 |
| Molded Package Thickness | A2 | | .026 | .031 | | 0.65 | 0.80 |
| Standoff | A1 | .000 | .0004 | .002 | 0.00 | 0.01 | 0.05 |
| Base Thickness | A3 | | .008 REF. | | | 0.20 REF. | |
| Overall Length | E | | .194 BSC | | | 4.92 BSC | |
| Molded Package Length | E1 | | .184 BSC | | | 4.67 BSC | |
| Exposed Pad Length | E2 | .152 | .158 | .163 | 3.85 | 4.00 | 4.15 |
| Overall Width | D | | .236 BSC | | | 5.99 BSC | |
| Molded Package Width | D1 | | .226 BSC | | | 5.74 BSC | |
| Exposed Pad Width | D2 | .085 | .091 | .097 | 2.16 | 2.31 | 2.46 |
| Lead Width | B | .014 | .016 | .019 | 0.35 | 0.40 | 0.47 |
| Lead Length | L | .020 | .024 | .030 | 0.50 | 0.60 | 0.75 |
| Tie Bar Width | R | | .014 | | | .356 | |
| Mold Draft Angle Top | α | | | 12° | | | 12° |

*Controlling Parameter

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.
JEDEC equivalent: Pending

Drawing No. C04-113

## APPENDIX A: DATA SHEET REVISION HISTORY

### Revision A

This is a new data sheet.

### Revision B

Rewrites of the Oscillator and Special Features of the CPU sections. General corrections to Figures and formatting.

## APPENDIX B: MIGRATING FROM OTHER PICmicro® DEVICES

This discusses some of the issues in migrating from other PICmicro devices to the PIC12F6XX family of devices.

### B.1 PIC12F675 to PIC12F683

**TABLE B-1: FEATURE COMPARISON**

| Feature | PIC12F675 | PIC12F683 |
|---|---|---|
| Max Operating Speed | 20 MHz | 20 MHz |
| Max Program Memory (Words) | 1024 | 2048 |
| SRAM (Bytes) | 64 | 128 |
| A/D Resolution | 10-bit | 10-bit |
| Data EEPROM (Bytes) | 128 | 256 |
| Timers (8/16-bit) | 1/1 | 2/1 |
| Oscillator Modes | 8 | 8 |
| Brown-out Detect | Y | Y |
| Internal Pull-ups | GP0/1/2/4/5 | GP0/1/2/4/5, $\overline{\text{MCLR}}$ |
| Interrupt-on-change | GP0/1/2/3/4/5 | GP0/1/2/3/4/5 |
| Comparators | 1 | 1 |
| CCP | N | Y |
| Ultra Low-Power Wake-up | N | Y |
| Extended WDT | N | Y |
| Software Control Option of WDT/BOD | N | Y |
| INTOSC Frequencies | 4 MHz | 32 kHz-8 MHz |
| Clock Switching | N | Y |

**Note:** This device has been designed to perform to the parameters of its data sheet. It has been tested to an electrical specification designed to determine its conformance with these parameters. Due to process differences in the manufacture of this device, this device may have different performance characteristics than its earlier version. These differences may cause this device to perform differently in your application than the earlier version of this device.

NOTES:

## INDEX

## ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape® or Microsoft® Internet Explorer. Files are also available for FTP download from our FTP site.

### Connecting to the Microchip Internet Web Site

The Microchip web site is available at the following URL:

**www.microchip.com**

The file transfer site is available by using an FTP service to connect to:

**ftp://ftp.microchip.com**

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events

## SYSTEMS INFORMATION AND UPGRADE HOT LINE

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive the most current upgrade kits. The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada and

1-480-792-7302 for the rest of the world.

042003

## 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: PIC12F683 Literature Number: DS41211B

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 | /XX | XXX |
|---|---|---|---|
| Device | Temperature Range | Package | Pattern |

| | | | |
|---|---|---|---|
| Device | PIC12F683: | Standard V$_{DD}$ range | |
| | PIC12F683T: | (Tape and Reel) | |
| Temperature Range | I | = | -40°C to +85°C |
| | E | = | -40°C to +125°C |
| Package | P | = | PDIP |
| | SN | = | SOIC (Gull wing, 150 mil body) |
| | MF | = | DFN-S |
| Pattern | 3-Digit Pattern Code for QTP (blank otherwise) | | |

**Examples:**

a) PIC12F683-E/P 301 = Extended Temp., PDIP package, 20 MHz, QTP pattern #301

b) PIC12F683-I/SO = Industrial Temp., SOIC package, 20 MHz

* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type.

# 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: 480-792-7627
Web Address: http://www.microchip.com

### Atlanta
3780 Mansell Road, Suite 130
Alpharetta, GA 30022
Tel: 770-640-0034
Fax: 770-640-0307

### Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848
Fax: 978-692-3821

### Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071
Fax: 630-285-0075

### Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423
Fax: 972-818-2924

### Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250
Fax: 248-538-2260

### Kokomo
2767 S. Albright Road
Kokomo, IN 46902
Tel: 765-864-8360
Fax: 765-864-8387

### Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888
Fax: 949-263-1338

### San Jose
1300 Terra Bella Avenue
Mountain View, CA 94043
Tel: 650-215-1444
Fax: 650-961-0286

### Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699
Fax: 905-673-6509

## ASIA/PACIFIC

### Australia
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

### China - Beijing
Unit 706B
Wan Tai Bei Hai Bldg.
No. 6 Chaoyangmen Bei Str.
Beijing, 100027, China
Tel: 86-10-85282100
Fax: 86-10-85282104

### China - Chengdu
Rm. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200
Fax: 86-28-86766599

### China - Fuzhou
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506
Fax: 86-591-7503521

### China - Hong Kong SAR
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

### China - Shanghai
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700
Fax: 86-21-6275-5060

### China - Shenzhen
Rm. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380
Fax: 86-755-8295-1393

### China - Shunde
Room 401, Hongjian Building, No. 2
Fengxiangnan Road, Ronggui Town, Shunde
District, Foshan City, Guangdong 528303, China
Tel: 86-757-28395507 Fax: 86-757-28395571

### China - Qingdao
Rm. B505A, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

### India
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-22290061 Fax: 91-80-22290062

### Japan
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

### Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5932 or
82-2-558-5934

### Singapore
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

### Taiwan
Kaohsiung Branch
30F - 1 No. 8
Min Chuan 2nd Road
Kaohsiung 806, Taiwan
Tel: 886-7-536-4818
Fax: 886-7-536-4803

### Taiwan
Taiwan Branch
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

## EUROPE

### Austria
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

### Denmark
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45-4420-9895 Fax: 45-4420-9910

### France
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

### Germany
Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

### Italy
Via Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611
Fax: 39-0331-466781

### Netherlands
P. A. De Biesbosch 14
NL-5152 SC Drunen, Netherlands
Tel: 31-416-690399
Fax: 31-416-690340

### United Kingdom
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-118-921-5869
Fax: 44-118-921-5820

02/17/04