

# **MICROPROCESSORS A (17.383)**

**Fall 2010**

## **Lecture Outline**

Class # 02

September 14, 2010

Dohn Bowden

# Today's Lecture

- Syllabus review
- Microcontroller Hardware and/or Interface
- Programming/Software
- Understanding the LEDs of the PICkit 1
- Lab
- Homework
- Finish Lab #1

# Course Admin

# Administrative

- Admin for tonight ...
  - Syllabus Review
  - Email Distribution List

# Syllabus Review

| <i>Week</i>  | <i>Date</i>         | <i>Topics</i>                                                       | <i>Lab</i>   | <i>Lab Report Due</i> |
|--------------|---------------------|---------------------------------------------------------------------|--------------|-----------------------|
| <del>1</del> | <del>09/07/10</del> | <del>Intro, Course &amp; Lab Overview, Microcontroller Basics</del> | <del>1</del> |                       |
| 2            | 09/14/10            | PIC16F684 Overview and General Input/Output                         | 1 con't      |                       |
| 3            | 09/21/10            | Switches                                                            | 2            |                       |
| 4            | 09/28/10            | Seven Segment LEDs                                                  |              | 1                     |
| 5            | 10/05/10            | <b>Examination 1</b>                                                | 2 con't      |                       |
| <b>X</b>     | <b>10/12/10</b>     | <b>No Class – Monday Schedule</b>                                   |              |                       |
| 6            | 10/19/10            | Analog to Digital Conversion                                        | 3            | 2                     |
| 7            | 10/26/10            | Analog to Digital Conversion con't                                  | 3 con't      |                       |
| 8            | 11/02/10            | LCD Interface and Assembly Language                                 | 4            |                       |
| 9            | 11/09/10            | Comparators                                                         | 4 con't      | 3                     |
| 10           | 11/16/10            | Timers and Pulse Width Modulation (PWM)                             | 5            |                       |
| 11           | 11/23/10            | Mixed C & Assembly Programming/Course Project                       | Project      | 4                     |
| 12           | 11/30/10            | <b>Examination 2</b>                                                |              |                       |
| 13           | 12/07/10            | Course Project                                                      | Project      | 5                     |
| 14           | 12/14/10            | Final Exam/Course Project Brief and Demonstration                   | Demo         |                       |
|              |                     |                                                                     |              |                       |

## Email Distribution List

- For those who sent me an email as of 8:30 PM Monday night ...
  - Did you all receive my trial email on each email account?
- If not ... please send me another email
-  Email me at:

Dohn\_Bowden@uml.edu

# **Microcontroller Hardware** **and / or** **Interfaces**

# PIC16F684 Features

- 35 Instructions
- 8-level deep hardware stack
- 2048 Flash (words) Program Memory
- Interrupt capability
- 12 I/O pins with individual direction control
- 2 Comparators
- A/D Converter (10-bit resolution and 8 channels)
- 2 Timers



# PIC16F684 Features

| Device    | Program Memory | Data Memory  |                | I/O | 10-bit A/D (ch) | Comparators | Timers 8/16-bit |
|-----------|----------------|--------------|----------------|-----|-----------------|-------------|-----------------|
|           | Flash (words)  | SRAM (bytes) | EEPROM (bytes) |     |                 |             |                 |
| PIC16F684 | 2048           | 128          | 256            | 12  | 8               | 2           | 2/1             |

# PIC16F684 Pin Diagram

14-pin PDIP, SOIC, TSSOP

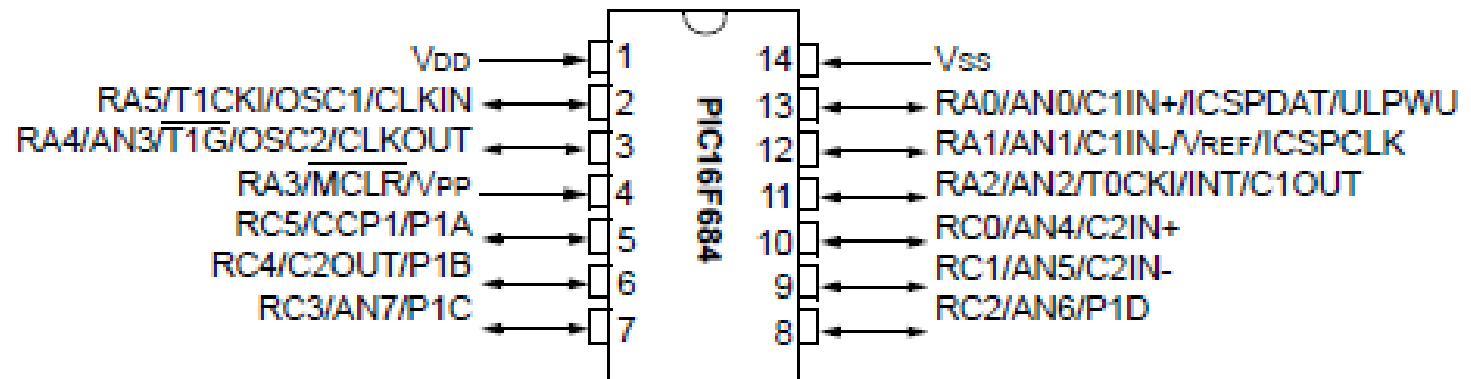
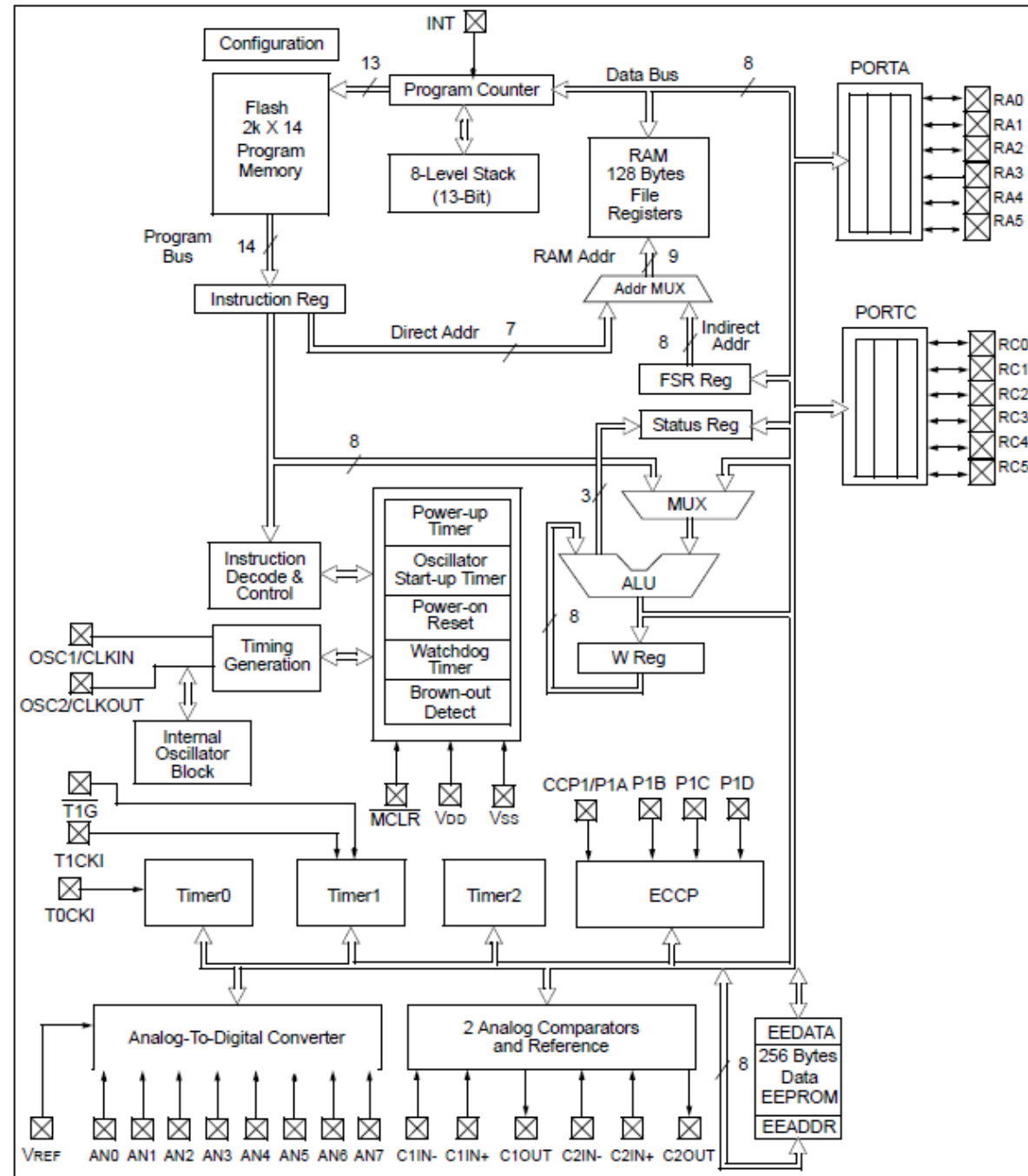


TABLE 1-1: PIC16F684 PINOUT DESCRIPTION

| Name                        | Function | Input Type | Output Type | Description                                              |
|-----------------------------|----------|------------|-------------|----------------------------------------------------------|
| RA0/AN0/C1IN+/ICSPDAT/ULPWU | RA0      | TTL        | CMOS        | PORTA I/O w/programmable pull-up and interrupt-on-change |
|                             | AN0      | AN         | —           | A/D Channel 0 input                                      |
|                             | C1IN+    | AN         | —           | Comparator 1 input                                       |
|                             | ICSPDAT  | TTL        | CMOS        | Serial Programming Data I/O                              |
|                             | ULPWU    | AN         | —           | Ultra Low-power Wake-up input                            |
| RA1/AN1/C1IN-/VREF/ICSPCLK  | RA1      | TTL        | CMOS        | PORTA I/O w/programmable pull-up and interrupt-on-change |
|                             | AN1      | AN         | —           | A/D Channel 1 input                                      |
|                             | C1IN-    | AN         | —           | Comparator 1 input                                       |
|                             | VREF     | AN         | —           | External Voltage Reference for A/D                       |
|                             | ICSPCLK  | ST         | —           | Serial Programming Clock                                 |
| RA2/AN2/T0CKI/INT/C1OUT     | RA2      | ST         | CMOS        | PORTA 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                                       |
|                             | C1OUT    | —          | CMOS        | Comparator 1 output                                      |
| RA3/MCLR/VPP                | RA3      | TTL        | —           | PORTA input with interrupt-on-change                     |
|                             | MCLR     | ST         | —           | Master Clear w/internal pull-up                          |
|                             | VPP      | HV         | —           | Programming voltage                                      |
| RA4/AN3/T1G/OSC2/CLKOUT     | RA4      | TTL        | CMOS        | PORTA I/O w/programmable pull-up and interrupt-on-change |
|                             | AN3      | AN         | —           | A/D Channel 3 input                                      |
|                             | T1G      | ST         | —           | Timer1 gate                                              |
|                             | OSC2     | —          | XTAL        | Crystal/Resonator                                        |
|                             | CLKOUT   | —          | CMOS        | Fosc/4 output                                            |
| RA5/T1CKI/OSC1/CLKIN        | RA5      | TTL        | CMOS        | PORTA 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            |
| RC0/AN4/C2IN+               | RC0      | TTL        | CMOS        | PORTC I/O                                                |
|                             | AN4      | AN         | —           | A/D Channel 4 input                                      |
|                             | C2IN+    | AN         | —           | Comparator 2 input                                       |
| RC1/AN5/C2IN-               | RC1      | TTL        | CMOS        | PORTC I/O                                                |
|                             | AN5      | AN         | —           | A/D Channel 5 input                                      |
|                             | C2IN-    | AN         | —           | Comparator 2 input                                       |
| RC2/AN6/P1D                 | RC2      | TTL        | CMOS        | PORTC I/O                                                |
|                             | AN6      | AN         | —           | A/D Channel 6 input                                      |
|                             | P1D      | —          | CMOS        | PWM output                                               |
| RC3/AN7/P1C                 | RC3      | TTL        | CMOS        | PORTC I/O                                                |
|                             | AN7      | AN         | —           | A/D Channel 7 input                                      |
|                             | P1C      | —          | CMOS        | PWM output                                               |
| RC4/C2OUT/P1B               | RC4      | TTL        | CMOS        | PORTC I/O                                                |
|                             | C2OUT    | —          | CMOS        | Comparator 2 output                                      |
|                             | P1B      | —          | CMOS        | PWM output                                               |
| RC5/CCP1/P1A                | RC5      | TTL        | CMOS        | PORTC I/O                                                |
|                             | CCP1     | ST         | CMOS        | Capture input/Compare output                             |
|                             | P1A      | —          | CMOS        | PWM output                                               |
| Vss                         | Vss      | Power      | —           | Ground reference                                         |
| VDD                         | VDD      | Power      | —           | Positive supply                                          |

Legend: TTL = TTL input buffer, ST = Schmitt Trigger input buffer, AN = Analog input

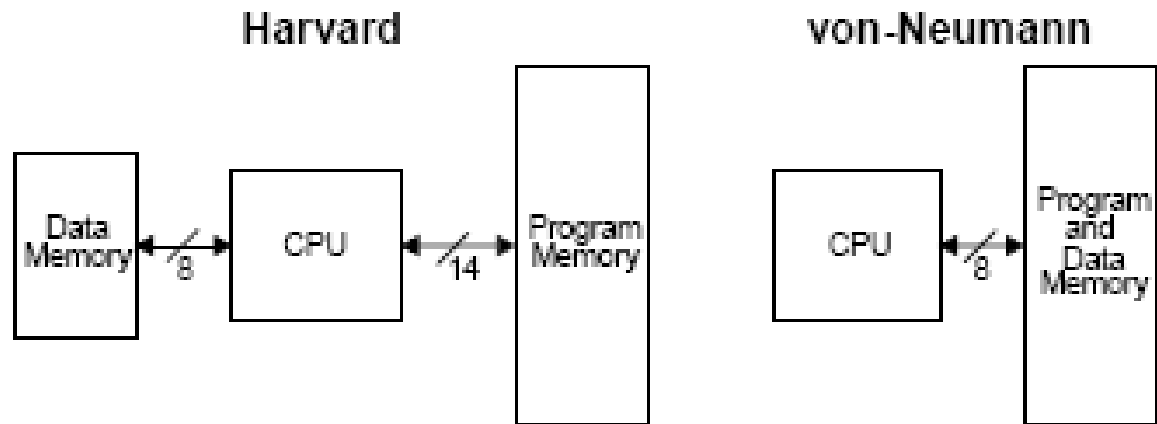
FIGURE 1-1: PIC16F684 BLOCK DIAGRAM



# Memory

- There are two different architecture regarding memory ...

# Program and Data Memory



# Harvard Architecture

- Physically separate memory for instructions and data
- Separate data buses to each memory

# von Neumann

- Architecture that uses a processing unit ... and ...
  - a *single* separate storage structure to hold *both* instructions and data



## **PIC16F684 Memory**

- The PIC16F684 uses the Harvard Architecture vice the von Neumann

# Program and Data Memory

- Separate Memory For ...
  - Program Memory
  - Variable Memory (data)/Register Spaces
  - Program Counter

# Program and Data Memory

- Advantages of separate memories ...
  - Ability of the processor to fetch new instructions ... while ...
    - Accessing the program memory/registers
  - However ... bad programs can still execute ... but ...
    - It will not try to execute data as instructions

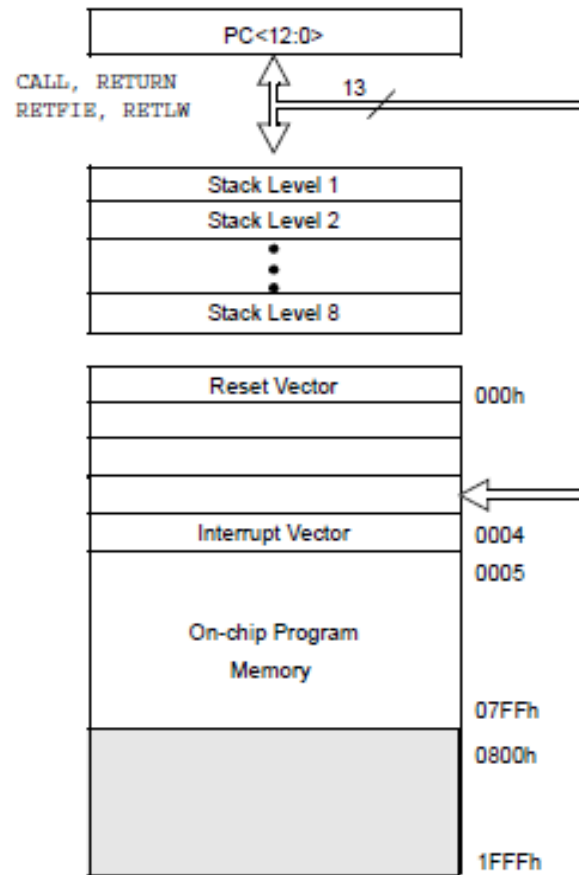
# Program and Data Memory

- Disadvantages of separate memories ...
  - Loss of flexibility in application organization ...
    - i.e. Changing data or stack segment sizes to accommodate different applications

# Program Memory Types

- Microchip offers three program memory types
- The memory type is designated in the part number by the first letter(s) after the family affiliation designators
  - C, as in PIC16CXXX.
    - These devices have *EPROM* type memory
  - CR, as in PIC16CRXXX
    - These devices have *ROM* type memory
  - F, as in PIC16FXXX
    - These devices have *Flash* type memory

# Program Memory Map and Stack for the PIC16F684



## **Registers/Variable Memory**

- Registers ... also known as Special Function Registers (SFR)
- Variable Memory ... also known as File Registers

# Special Function Registers (SFR)

- SFR functions/purpose:
  - Monitors the status of program execution
  - Provides an interface to hardware peripheral functions
- SFR can be classified into two sets:
  - Core and peripheral



**FIGURE 2-2: DATA MEMORY MAP OF THE PIC16F684**

| File Address                          |     | File Address                          |     |
|---------------------------------------|-----|---------------------------------------|-----|
| Indirect Addr. <sup>(1)</sup>         | 00h | Indirect Addr. <sup>(1)</sup>         | 80h |
| TMR0                                  | 01h | OPTION_REG                            | 81h |
| PCL                                   | 02h | PCL                                   | 82h |
| STATUS                                | 03h | STATUS                                | 83h |
| FSR                                   | 04h | FSR                                   | 84h |
| PORTA                                 | 05h | TRISA                                 | 85h |
|                                       | 06h |                                       | 86h |
| PORTC                                 | 07h | TRISC                                 | 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 | ANSEL                                 | 91h |
| T2CON                                 | 12h | PR2                                   | 92h |
| CCPR1L                                | 13h |                                       | 93h |
| CCPR1H                                | 14h |                                       | 94h |
| CCP1CON                               | 15h | WPUA                                  | 95h |
| PWM1CON                               | 16h | IOCA                                  | 96h |
| ECCPAS                                | 17h |                                       | 97h |
| WDTCN                                 | 18h |                                       | 98h |
| CMCON0                                | 19h | VRCON                                 | 99h |
| CMCON1                                | 1Ah | EEDAT                                 | 9Ah |
|                                       | 1Bh | EEADR                                 | 9Bh |
|                                       | 1Ch | EECON1                                | 9Ch |
|                                       | 1Dh | EECON2 <sup>(1)</sup>                 | 9Dh |
| ADRESH                                | 1Eh | ADRESL                                | 9Eh |
| ADCON0                                | 1Fh | ADCON1                                | 9Fh |
| General Purpose Registers<br>96 Bytes | 20h | General Purpose Registers<br>32 Bytes | A0h |
|                                       |     |                                       | BFh |
|                                       | 6Fh |                                       |     |
|                                       | 70  |                                       |     |
|                                       | 7Fh | Accesses 70h-7Fh                      | F0h |
|                                       |     |                                       | FFh |
| Bank 0                                |     | Bank 1                                |     |

 Unimplemented data memory locations, read as '0'.  
**Note 1:** Not a physical register.

**TABLE 2-1: PIC16F684 SPECIAL FUNCTION 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, BOR | Page     |
|--------|----------------------|------------------------------------------------------------------------------------------------|--------------------|---------|--------------------------------------------------|-----------------|---------------------|---------|---------|-------------------|----------|
| Bank 0 |                      |                                                                                                |                    |         |                                                  |                 |                     |         |         |                   |          |
| 00h    | INDF                 | Addressing this location uses contents of FSR to address data memory (not a physical register) |                    |         |                                                  |                 |                     |         |         | xxxx xxxx         | 19, 104  |
| 01h    | TMR0                 | Timer0 Module's Register                                                                       |                    |         |                                                  |                 |                     |         |         | xxxx xxxx         | 43, 104  |
| 02h    | PCL                  | Program Counter's (PC) Least Significant Byte                                                  |                    |         |                                                  |                 |                     |         |         | 0000 0000         | 19, 104  |
| 03h    | STATUS               | IRP <sup>(1)</sup>                                                                             | RP1 <sup>(1)</sup> | RP0     | $\overline{TO}$                                  | $\overline{PD}$ | Z                   | DC      | C       | 0001 1xxx         | 13, 104  |
| 04h    | FSR                  | Indirect Data Memory Address Pointer                                                           |                    |         |                                                  |                 |                     |         |         | xxxx xxxx         | 19, 104  |
| 05h    | PORTA <sup>(2)</sup> | —                                                                                              | —                  | RA5     | RA4                                              | RA3             | RA2                 | RA1     | RA0     | --x0 x000         | 31, 104  |
| 06h    | —                    | Unimplemented                                                                                  |                    |         |                                                  |                 |                     |         |         | —                 | —        |
| 07h    | PORTC <sup>(2)</sup> | —                                                                                              | —                  | RC5     | RC4                                              | RC3             | RC2                 | RC1     | RC0     | --xx 0000         | 40, 104  |
| 08h    | —                    | Unimplemented                                                                                  |                    |         |                                                  |                 |                     |         |         | —                 | —        |
| 09h    | —                    | Unimplemented                                                                                  |                    |         |                                                  |                 |                     |         |         | —                 | —        |
| 0Ah    | PCLATH               | —                                                                                              | —                  | —       | Write Buffer for upper 5 bits of Program Counter |                 |                     |         | ---     | 0 0000            | 19, 104  |
| 0Bh    | INTCON               | GIE                                                                                            | PEIE               | TOIE    | INTE                                             | RAIE            | TOIF                | INTF    | RAIF    | 0000 0000         | 15, 104  |
| 0Ch    | PIR1                 | EEIF                                                                                           | ADIF               | CCP1IF  | C2IF                                             | C1IF            | OSFIF               | TMR2IF  | TMR1IF  | 0000 0000         | 17, 104  |
| 0Dh    | —                    | Unimplemented                                                                                  |                    |         |                                                  |                 |                     |         |         | —                 | —        |
| 0Eh    | TMR1L                | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register                    |                    |         |                                                  |                 |                     |         |         | xxxx xxxx         | 47, 104  |
| 0Fh    | TMR1H                | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register                     |                    |         |                                                  |                 |                     |         |         | xxxx xxxx         | 47, 104  |
| 10h    | T1CON                | T1GINV                                                                                         | TMR1GE             | T1CKPS1 | T1CKPS0                                          | T1OSCEN         | $\overline{T1SYNC}$ | TMR1CS  | TMR1ON  | 0000 0000         | 50, 104  |
| 11h    | TMR2                 | Timer2 Module Register                                                                         |                    |         |                                                  |                 |                     |         |         | 0000 0000         | 53, 104  |
| 12h    | T2CON                | —                                                                                              | TOUTPS3            | TOUTPS2 | TOUTPS1                                          | TOUTPS0         | TMR2ON              | T2CKPS1 | T2CKPS0 | -000 0000         | 54, 104  |
| 13h    | CCPR1L               | Capture/Compare/PWM Register 1 Low Byte                                                        |                    |         |                                                  |                 |                     |         |         | XXXX XXXX         | 80, 104  |
| 14h    | CCPR1H               | Capture/Compare/PWM Register 1 High Byte                                                       |                    |         |                                                  |                 |                     |         |         | XXXX XXXX         | 80, 104  |
| 15h    | CCP1CON              | P1M1                                                                                           | P1M0               | DC1B1   | DC1B0                                            | CCP1M3          | CCP1M2              | CCP1M1  | CCP1M0  | 0000 0000         | 79, 104  |
| 16h    | PWM1CON              | PRSEN                                                                                          | PDC6               | PDC5    | PDC4                                             | PDC3            | PDC2                | PDC1    | PDC0    | 0000 0000         | 96, 104  |
| 17h    | ECCPAS               | ECCPASE                                                                                        | ECCPAS2            | ECCPAS1 | ECCPAS0                                          | PSSAC1          | PSSAC0              | PSSBD1  | PSSBD0  | 0000 0000         | 93, 104  |
| 18h    | WDTCON               | —                                                                                              | —                  | —       | WDTPS3                                           | WDTPS2          | WDTPS1              | WDTPS0  | SWDTEN  | ---0 1000         | 111, 104 |
| 19h    | CMCON0               | C2OUT                                                                                          | C1OUT              | C2INV   | C1INV                                            | CIS             | CM2                 | CM1     | CM0     | 0000 0000         | 61, 104  |
| 1Ah    | CMCON1               | —                                                                                              | —                  | —       | —                                                | —               | —                   | T1GSS   | C2SYNC  | ---- --10         | 62, 104  |
| 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         | 71, 104  |
| 1Fh    | ADCON0               | ADFM                                                                                           | VCFG               | —       | CHS2                                             | CHS1            | CHS0                | GO/DONE | ADON    | 00-0 0000         | 70, 104  |

**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.

**Note 2:** Port pins with analog functions controlled by the ANSEL register will read '0' immediately after a Reset even though the data latches are either undefined (POR) or unchanged (other Resets).

# Core

- The core pertains to the basic features that are required to make the device operate ... which are ...
  - Device Oscillator
  - Reset logic
  - CPU (Central Processing Unit) operation
  - ALU (Arithmetic Logical Unit) operation
  - Device memory map organization
  - Interrupt operation
  - Instruction set

# Peripherals

- Peripherals are the features that add a differentiation from a microprocessor/microcontroller
- These ease in interfacing to the external world ... such as ...
  - General purpose I/O
  - A/D inputs
  - PWM outputs
- And internal tasks ... such as ...
  - Keeping different time bases (such as timers)

# Peripherals

- General purpose I/O
- Timer0
- Timer1
- Capture, Compare, and PWM (CCP)
- Voltage References
- Comparators
- 10-bit Analog to Digital (A/D)

# Development

- We will explore the features of the PIC16F684 microcontroller as we encounter them
- Same will be true for introducing the languages
  - C
  - Assembly

# **Microcontroller Hardware Interface**

## **With LEDs ...**

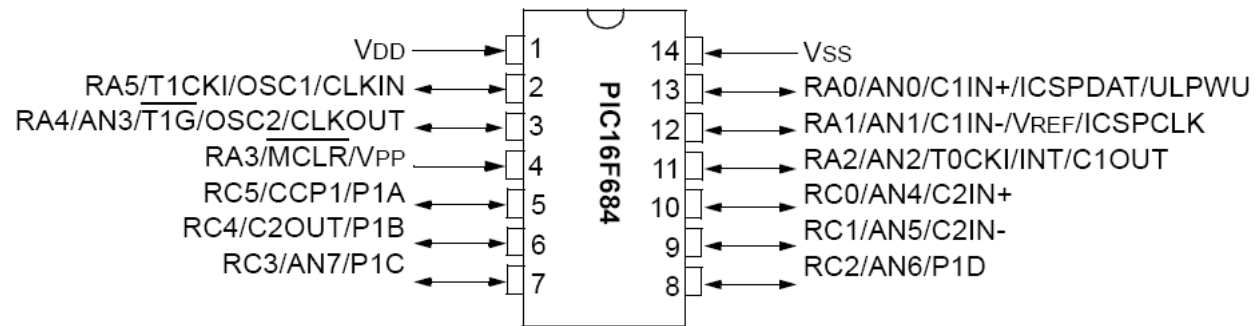
# How does the hardware work?

- Lets understand what we want to do ...
  - Before we try to write a program to make it work

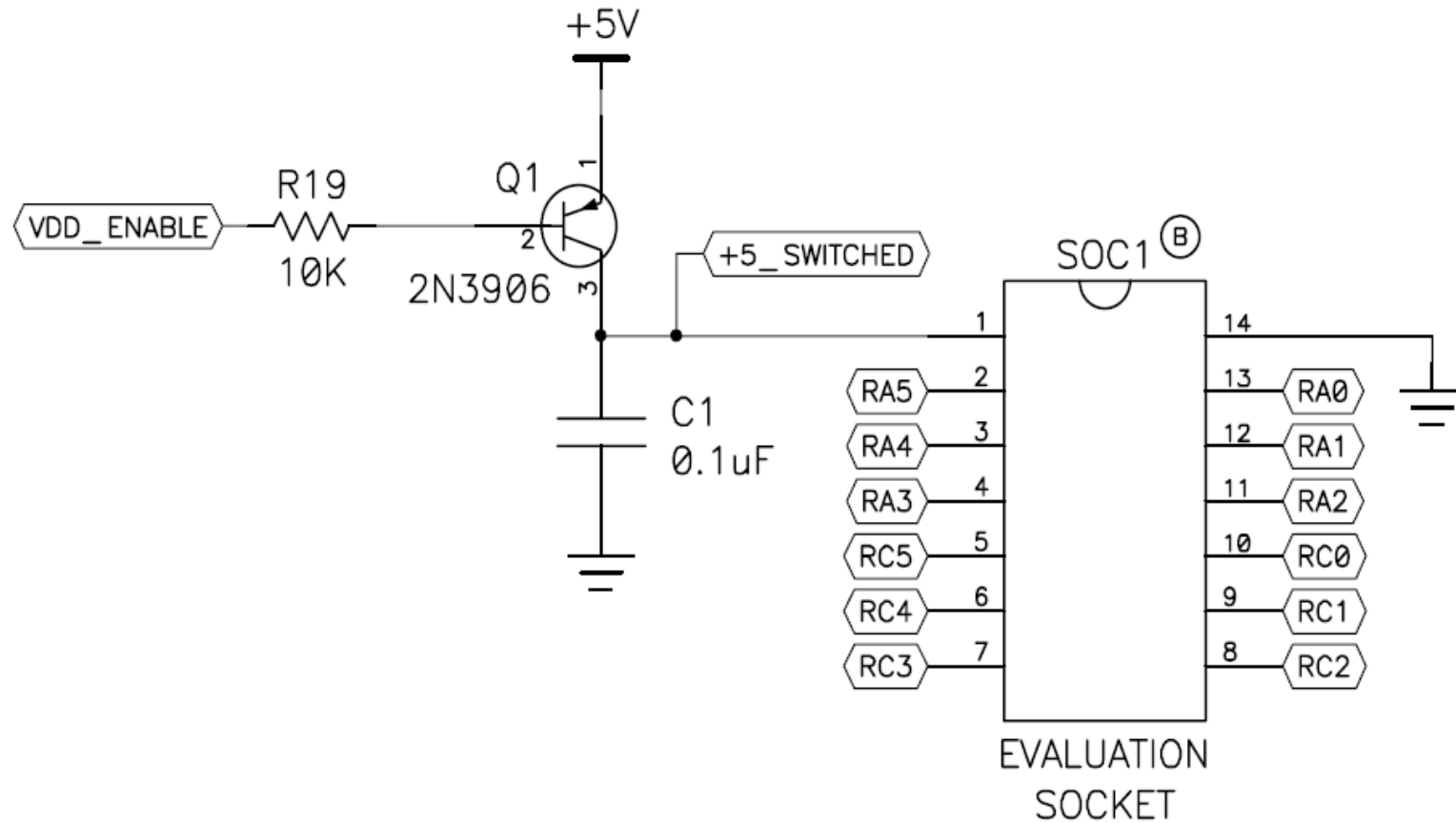


# PIC16F684 Pin Diagram

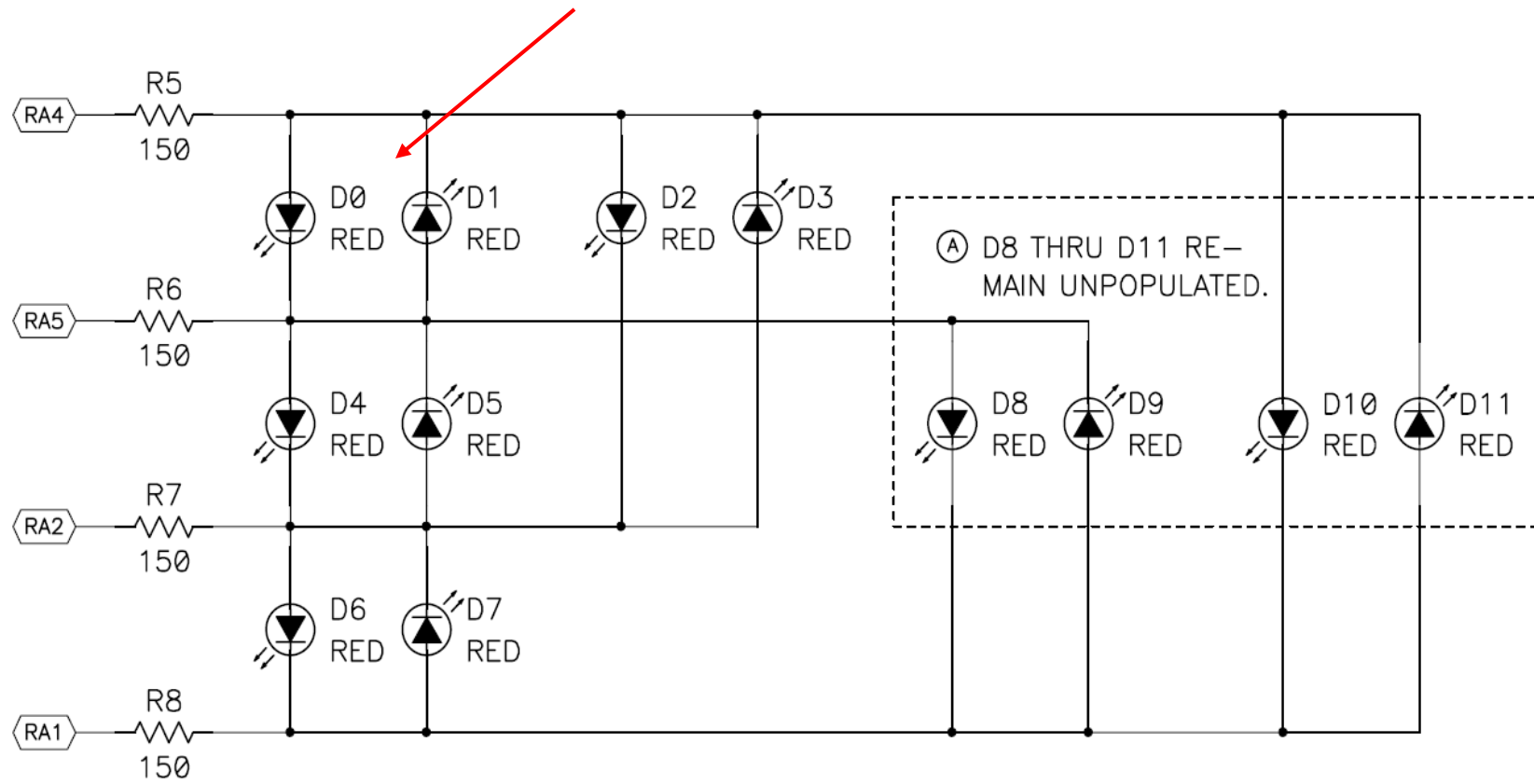
14-pin PDIP, SOIC, TSSOP



# PICkit 1 Evaluation Socket



# PICKit 1 LED Layout



**TABLE 1-1: PIC16F684 PINOUT DESCRIPTION**

| Name                        | Function | Input Type | Output Type | Description                                              |
|-----------------------------|----------|------------|-------------|----------------------------------------------------------|
| RA0/AN0/C1IN+/ICSPDAT/ULPWU | RA0      | TTL        | CMOS        | PORTA I/O w/programmable pull-up and interrupt-on-change |
|                             | AN0      | AN         | —           | A/D Channel 0 input                                      |
|                             | C1IN+    | AN         | —           | Comparator 1 input                                       |
|                             | ICSPDAT  | TTL        | CMOS        | Serial Programming Data I/O                              |
|                             | ULPWU    | AN         | —           | Ultra Low-power Wake-up input                            |
| RA1/AN1/C1IN-/VREF/ICSPCLK  | RA1      | TTL        | CMOS        | PORTA I/O w/programmable pull-up and interrupt-on-change |
|                             | AN1      | AN         | —           | A/D Channel 1 input                                      |
|                             | C1IN-    | AN         | —           | Comparator 1 input                                       |
|                             | VREF     | AN         | —           | External Voltage Reference for A/D                       |
|                             | ICSPCLK  | ST         | —           | Serial Programming Clock                                 |
| RA2/AN2/T0CKI/INT/C1OUT     | RA2      | ST         | CMOS        | PORTA 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                                       |
|                             | C1OUT    | —          | CMOS        | Comparator 1 output                                      |
| RA3/MCLR/VPP                | RA3      | TTL        | —           | PORTA input with interrupt-on-change                     |
|                             | MCLR     | ST         | —           | Master Clear w/internal pull-up                          |
|                             | VPP      | HV         | —           | Programming voltage                                      |
| RA4/AN3/T1G/OSC2/CLKOUT     | RA4      | TTL        | CMOS        | PORTA I/O w/programmable pull-up and interrupt-on-change |
|                             | AN3      | AN         | —           | A/D Channel 3 input                                      |
|                             | T1G      | ST         | —           | Timer1 gate                                              |
|                             | OSC2     | —          | XTAL        | Crystal/Resonator                                        |
|                             | CLKOUT   | —          | CMOS        | Fosc/4 output                                            |
| RA5/T1CKI/OSC1/CLKIN        | RA5      | TTL        | CMOS        | PORTA 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            |
| RC0/AN4/C2IN+               | RC0      | TTL        | CMOS        | PORTC I/O                                                |
|                             | AN4      | AN         | —           | A/D Channel 4 input                                      |
|                             | C2IN+    | AN         | —           | Comparator 2 input                                       |
| RC1/AN5/C2IN-               | RC1      | TTL        | CMOS        | PORTC I/O                                                |
|                             | AN5      | AN         | —           | A/D Channel 5 input                                      |
|                             | C2IN-    | AN         | —           | Comparator 2 input                                       |
| RC2/AN6/P1D                 | RC2      | TTL        | CMOS        | PORTC I/O                                                |
|                             | AN6      | AN         | —           | A/D Channel 6 input                                      |
|                             | P1D      | —          | CMOS        | PWM output                                               |
| RC3/AN7/P1C                 | RC3      | TTL        | CMOS        | PORTC I/O                                                |
|                             | AN7      | AN         | —           | A/D Channel 7 input                                      |
|                             | P1C      | —          | CMOS        | PWM output                                               |
| RC4/C2OUT/P1B               | RC4      | TTL        | CMOS        | PORTC I/O                                                |
|                             | C2OUT    | —          | CMOS        | Comparator 2 output                                      |
|                             | P1B      | —          | CMOS        | PWM output                                               |
| RC5/CCP1/P1A                | RC5      | TTL        | CMOS        | PORTC I/O                                                |
|                             | CCP1     | ST         | CMOS        | Capture input/Compare output                             |
|                             | P1A      | —          | CMOS        | PWM output                                               |
| Vss                         | Vss      | Power      | —           | Ground reference                                         |
| VDD                         | VDD      | Power      | —           | Positive supply                                          |

**Legend:** TTL = TTL input buffer, ST = Schmitt Trigger input buffer, AN = Analog input

## So ...

- In order to light D0 ...
  - We need a high on ...
    - RA4
  - And a low on ...
    - RA5

## **Now ... Lets examine the software ...**

- Let see what we need to do in order to light D0 via software
- First we will need to understand the basic structure of a ...
  - “C” program

# Programming / Software

# Programming

- Commands/instructions that we will encounter tonight
  - C commands
  - PIC16F684 control



# C commands

- C program structure
- Comments
- *# include*
- Integer variables
- *for*
- *while*
- *NOP()*
- functions

# The Structure of a Typical C Program

```
main()  
{  
  
}
```

- The first line .... **main()**
  - Informs the system that the name of the program is main
  - **()** specify that main is a *function*
  - **()** being empty indicates that the function main takes **no arguments**
- **{** ... start of program statements for the routine
- **}** ... end of program statements for the routine

# The “Main” Function

- Main is a special function
  - Contains the main program blocks
    - Within which all lower-level functions are contained

# Functions

- We have Functions other than the `main()` function within the program
- These functions perform certain tasks that can be called from within the `main()` function
- Up to now ... calling Functions always appear in the code following the functions they call
- They have been appearing prior to the `main()` Function

# Function Prototypes

- You can place other functions after the main() Function ... if ...
  - You place function prototypes in your program that describe a function's return and parameter types (to be discussed in a future lecture)

```
void function_name(int x, int y, int z);
```

- Then the Function will appear after the main() Function

# Function Prototypes

```

/*****
* Function: PORT_init
*
* Description: Initializes PORTA to a known condition
*
* Notes:  None
*
* Returns: None
*
*****/
void PORTA_init(void)
{
    PORTA = 0;           // All PORTA Pins are low
    CMCON0 = 7;          // Turn off Comparators
    ANSEL = 0;           // Turn off ADC
    TRISA = 0b001111;    // RA4 and 5 are outputs; RA0,1,2, and 3 are input

    return;

}
/***** END OF PORTA_init *****/

```

# Function Prototypes

- If the following command is placed prior to the main() function

```
void PORTA_init(void);
```

- Then the Function will can be placed after the main() Function
- Now the main() function is at the beginning of the listing vice at the end

# Comments (Within Your Programs)

- Comment statements in a program are used to ...
  - Document ... and ...
  - Enhance its readability
- The use of comments are extremely important ...
  - Enhances the routine or code
  - Can be used during the debugging phase
- Enter comments as you write the code
  - While the code is fresh in your mind



# Comments (Within Your Programs)

- There are two ways to insert comments ... FIRST ...
  - Initiate the comment by the two characters `/` and `*`
    - This marks the beginning of the comment
  - Terminate the comment by the two characters `*` and `/`
    - This marks the end of the comment
  - All characters included between the opening `/` `*` and the closing `*` `/` are ...
    - Treated as part of the comment ... and ...
    - Are ignored by the C compiler
  - This is used when comments span several lines of the program

## Comments (Within Your Programs)

- The second way to insert comments ... is by ...
  - Is by using two consecutive slash characters **//**
    - Any characters that follow these slashes up to the end of the line are ignored by the compiler

# Comments (Within Your Programs)

- Examples ...

```
// This entire line contains comments, next line not included
```

```
/* Comments go between the slashes and asterisk */
```

```
/* First line of a comment which runs multiple lines ...
```

```
    This is a comment, line two
```

```
    This is a comment, line three
```

```
    .
```

```
    .
```

```
    Etc
```

```
    The next line is the last line of a comment
```

```
*/
```

# The *#include* Statement

- The *#include* statement causes ...
  - ... the entire contents of a specified source file to be processed as if those contents had appeared in place of the *#include* command
  - Useful for items that are frequently used in most applications
    - Saves time retyping into the program
- These files normally end in *.h* and are referred to as header or include files

# The *# include* Statement

- Example ...

```
# include <pic.h>
```

- You can review the contents of the *pic.h* header file ...

C:\Program Files\HI-TECH Software\PICC\9.70\include\pic.h

- Take note of all the other header files available in the include directory

# The *pic.h* Header File

```
#ifndef _PIC_H_
#define _PIC_H_

#ifndef _HTC_H_
#include <htc.h>
#endif

#if defined(_10F200)    || defined(_10F202)    ||\
    defined(_10F204)    || defined(_10F206)    ||\
        #include <pic10f20x.h>
#endif
#if defined(_10F220)    || defined(_10F222)    ||\
        #include <pic10f22x.h>
#endif
#if defined(_12C508)    || defined(_12C509)    ||\
    defined(_12F508)    || defined(_12F509)    ||\
    defined(_12C508A)    || defined(_12C509A)    ||\
    defined(_12CE518)    || defined(_12CE519)    ||\
    defined(_12C509AG)    || defined(_12C509AF)    ||\
    defined(_12CR509A)    || defined(_RF509AG)    ||\
    defined(_RF509AF)
        #include <pic125xx.h>
#endif
#if defined(_12F510)
        #include <pic12f510.h>
```

**THE FILE CONTINUES ...**

# Working with Variables

- C programming allows you to assign symbolic names, known as variable names, for computations and results
- Variable declaration format:

```
data_type  Variable_Name = initial_value;
```

- More than one variable can be declared at once using the following format:

```
data_type  Name = initial_value, Name = initial_value, ... ;
```

## Variables --- data\_type

- Data types are summarized in Table 3.1 (PICC Pro Manual)

Table 3.1: Basic data types

| Type                | Size (bits) | Arithmetic Type            |
|---------------------|-------------|----------------------------|
| bit                 | 1           | unsigned integer           |
| char                | 8           | signed or unsigned integer |
| unsigned char       | 8           | unsigned integer           |
| short               | 16          | signed integer             |
| unsigned short      | 16          | unsigned integer           |
| int                 | 16          | signed integer             |
| unsigned int        | 16          | unsigned integer           |
| short long          | 24          | signed integer             |
| unsigned short long | 24          | unsigned integer           |
| long                | 32          | signed integer             |
| unsigned long       | 32          | unsigned integer           |
| float               | 24          | real                       |
| double              | 24 or 32    | real                       |



## Variables --- Variable\_Name

- Rules for forming variable names (Variable\_Name) ...
  - Can start with any upper or lower case letter or the underscore character
  - After the starting character ... any letter, number, or underscore character can be used. **NO BLANKS**

## Variables --- **initial\_value**

- Assignment of initial value is optional
- Signed integer (16 bits) range as follows ...

-32,768 to 32,767

# Working with Variables

- Examples ...

```
int i;  
int j;
```

```
int i, j;
```

```
int i = 1; j = 5;
```

# The *for* Statement

- The general format for declaring the *for* statement is as follows:

```
for ( expression_1; expression_2; expression_3 )  
    program_statement
```

- *expression\_1* is evaluated once the loop begins
- Next ... *expression\_2* is evaluated
  - If the value is nonzero, *program\_statement* is executed and then *expression\_3* is evaluated
- Execution of *program\_statement* continues as long as the value of *expression\_2* is nonzero

# The *for* Statement

- To summarize ... the *for* statement is as follows:

```
for (initialization; Loop test expression; Loop increment)  
    program_statement
```

Loop increment:

|                                      |                           |
|--------------------------------------|---------------------------|
| <code>variable = variable + 1</code> | Adds one to the variable  |
| <code>variable ++</code>             | Adds one to the variable  |
| <code>variable = variable -1</code>  | Subtracts one to variable |
| <code>variable --</code>             | Subtracts one to variable |
| <code>for(;;)</code>                 | Loop forever              |

# The *for* Statement

- Example ...

```
int i, j;  
  
    for (i = 0; i < 255; i++)  
        for (j = 0; j < 255; j++);
```

- The loop test expression is evaluated as follows:

$i < 255$  has the value 1 if  $i$  is less than **255**, and 0 otherwise

# The *while* Statement

- The *while* loop allows you to repeatedly execute a set of instructions while a test expression is true
- The general format for declaring the *while* statement is as follows:

```
while ( expression )  
    program_statement
```

- `program_statement` is executed as long as the value of expression is nonzero
- Note that, because `expression` is evaluated each time *before* the execution of `program_statement`, `program_statement` may never be executed

# The *while* Statement

- Example ...

```
while(1 == 1)           // Loop Forever
{
    delay_routine();

    RA4 = 1;             // D0 LED On by making RA4 high

    delay_routine();

    RA4 = 0;             // D0 LED Off by making RA4 low

}                        // *** END OF While (1 == 1) LOOP
```



## ***NOP()* Statement**

- The ***NOP();*** statement
  - No operation
- Can be used as a breakpoint without affecting the operation of the C program
- You cannot set a breakpoint on a line without code!

# General Purpose Input Output Associated Registers

- PORTA
- TRISA
- PORTC
- TRISC

## ***PORTA*** Register

- **PORTA** register make up six of the twelve general purpose **Input/Output (I/O)** pins available
- **PORTA** is a 6-bit wide, bidirectional port
  - Active **PORTA** Register bits are 0 through 5
    - Individual Register bits are called **RA0**, **RA1**, **RA2**, ... **RA5**
  - **PORTA** Register bits 6 and 7 are not used/unimplemented
- **PORTA** corresponding **data direction** register is the **TRISA** Register

## ***PORTA*** Register (continued)

REGISTER 4-1: PORTA – PORTA 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 |
| —     | —   | RA5   | RA4   | RA3   | RA2   | RA1   | RA0   |
| bit 7 |     | bit 0 |       |       |       |       |       |

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

bit 5-0: RA<5:0>: PORTA I/O Pin bit

1 = Port pin is > V<sub>IH</sub>

0 = Port pin is < V<sub>IL</sub>

### 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

## **TRISA** Register

- The **TRISA** Register controls the **data direction** of the six general purpose Input/Output pins of **PORTA**
- Meaning, the state of each **TRISA** bit will set the data direction for each corresponding **PORTA** pin
  - Setting an individual **TRISA** bit (= **1**)
    - will make the corresponding **PORTA** pin an **input**
  - Clearing an individual **TRISA** bit (= **0**)
    - will make the corresponding **PORTA** pin an **output**
- The exception is **RA3**, which is input only ... and ...
  - its **TRIS** bit will always read as '1', meaning input only

## *TRISA* Register

**REGISTER 4-2: TRISA – PORTA 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  |
| —     | —   | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 |
| bit 7 |     | bit 0  |        |        |        |        |        |

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

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

**Note 1:** TRISA<3> always reads '1'.

**2:** TRISA<5:4> always reads '1' in XT, HS and LP OSC 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

## ***TRISA* Register**

- When the chip is powered up ... TRISA bits ...
  - Default to “1”, meaning input
- Therefore, we do not need to initialize for input, only output
- Why?
  - If the pin is incorrectly wired, it could be easily damaged if it was set to an output ... why?
    - If the pin is accidentally grounded ... and ...
    - Driven to a high state
      - » The short circuit would likely damage the output circuit
      - » If set to an input, no damage would be done

## **TRISA** Register

- When used as analog inputs (will discuss in more detail in a future lecture) ...
  - The **TRISA** register controls the direction of the **PORTA** pins, even when they are being used as analog inputs
  - We must ensure the bits in the **TRISA** register are maintained set when using them as analog inputs
  - I/O pins configured as analog input always read '0'



# TRIS – Explained

- Three-State logic, also known as TRI-STATE logic
- The name is misleading ... it is not digital logic with 3 voltage levels
- It is just ordinary logic
  - High (1)
  - Low (0)
- With a third output state ...
  - Open circuit (disconnected)
- Therefore ... The three states (TRI-STATE) are ...
  - HIGH
  - LOW
  - Disconnected (open circuit)

## TRIS – Explained

- A separate “enable” input determines whether the output ...
  - Behaves like ordinary logic (high/low) ... or ...
  - The “third” (open) state
- In the “open” state ... which ...
  - Cuts off the logic states

# Software Control Of Registers

- The HI-TECH PICC compilers contains header files that equate *variables* to Special Function Registers
  - *pic16f684.h* is the header file used for the 684
  - Variable names are usually the same as the Register name
  - Variable names are also assigned to individual bits
- Access to each register is via reading or writing from/to a particular variable
  - Can either be the entire register or an individual bit

# PIC16F684 PICC STD Variables

- Today we shall explore a few variables used in Lab #1

PORTA

RA4

TRISA

CMCON0

ANSEL

\_\_config

## ***PORTA* Register PICC Pro C Variables**

- PORTA
- R&#
  - Where
    - & represents the port (either A or C)
    - # represents the port's pin (0 – 5)
- TRIS&#
  - Same convention as above

# Input/Output Commands on *PORTA*

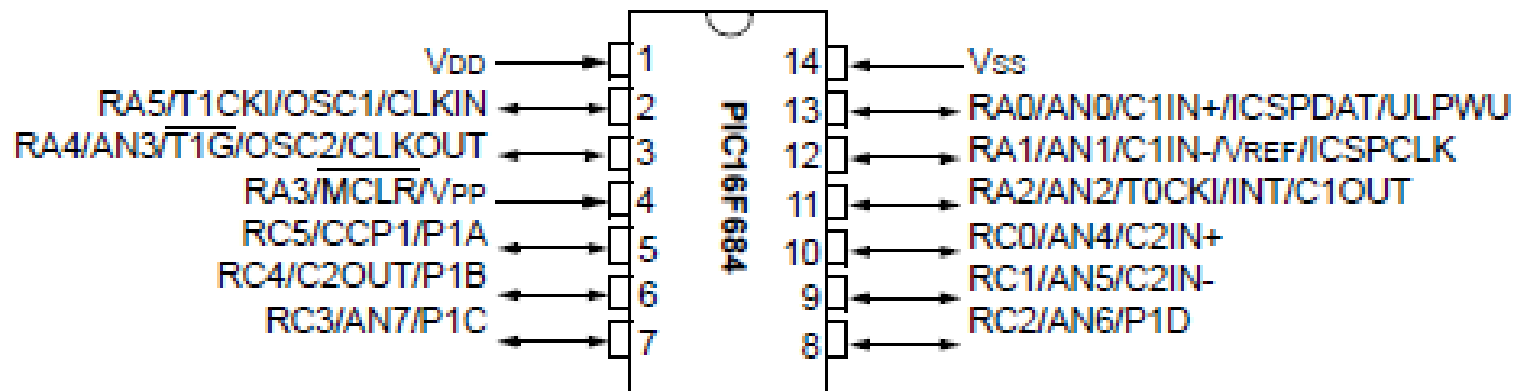
- Examples ...

```
PORTA = 0;      // Initializes PORTA. All pins are set to zero  
  
RA4 = 0;        // Makes RA4 low (0 volts)  
  
RA4 = 1;        // Makes RA4 high (5 volts)  
  
TRISA = 0;      // Makes all PORTA pins outputs  
  
TRISA4 = 0;     // Makes RA4 an Output  
  
TRISA5 = 1;     // Makes RA5 an Input  
  
RA5 = 0;        // Makes RA5 low (0 volts)  
  
RA5 = 1;        // Makes RA5 high (5 volts)
```

## ***PORTA*** Pins are Multipurpose

- From the Pin-out diagram ... pins 2, 3, 4, 11, 12, and 13

14-pin PDIP, SOIC, TSSOP



## ***PORTA* Pins are Multipurpose (con't)**

- We see that the pins are multipurpose
  - Input/Output
  - Analog to digital converter
  - Comparator
- To use PORTA for the I/O function
  - We need to turn off the other functions



## ***PORTA* Pins are Multipurpose (con't)**

- To turn off the other functions we use the following commands

```
CMCON0 = 7;           // Turns off the comparators
```

```
ANSEL = 0;            // Turns off the Analog to Digital Converter
```

- We will discuss the above commands in more details when we are using the associated peripheral features

# Configuration Bits

- Configuration bits can select various device configurations (shown in Register 12-1)
  - Bits are “programmed”
    - Read as ‘0’
  - Or ... “unprogrammed” ...
    - Read as ‘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

# Configuration Word

- Together ... all the configuration bits make up the configuration word
- Operating configuration parameters which are set on power up
- These parameters are specified by writing to a special word in memory
- This word is called the Configuration Word

\_\_CONFIG (xxx & yyy & etc);

*Note* ... \_\_ are two underscore characters

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

| —         | —                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | FCMEN | IESO | BODEN1 | BODEN0 | CPD | CP | MCLRE | PWRT | WDTE | FOSC2 | FOSC1 | FOSC0 |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------|--------|--------|-----|----|-------|------|------|-------|-------|-------|
| bit 13    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |       |      |        |        |     |    |       |      |      |       |       | bit 0 |
| bit 13-12 | Unimplemented: Read as '1'                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |      |        |        |     |    |       |      |      |       |       |       |
| bit 11    | <b>FCMEN:</b> Fail-Safe Clock Monitor Enabled bit<br>1 = Fail-Safe Clock Monitor is enabled<br>0 = Fail-Safe Clock Monitor is disabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |      |        |        |     |    |       |      |      |       |       |       |
| bit 10    | <b>IESO:</b> Internal External Switchover bit<br>1 = Internal External Switchover mode is enabled<br>0 = Internal External Switchover mode is disabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |      |        |        |     |    |       |      |      |       |       |       |
| bit 9-8   | <b>BODEN&lt;1:0&gt;:</b> Brown-out Detect Selection bits <sup>(1)</sup><br>11 = BOD enabled<br>10 = BOD enabled during operation and disabled in Sleep<br>01 = BOD controlled by SBODEN bit (PCON<4>)<br>00 = BOD disabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |      |        |        |     |    |       |      |      |       |       |       |
| bit 7     | <b>CPD:</b> Data Code Protection bit <sup>(2)</sup><br>1 = Data memory code protection is disabled<br>0 = Data memory code protection is enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |       |      |        |        |     |    |       |      |      |       |       |       |
| bit 6     | <b>CP:</b> Code Protection bit <sup>(3)</sup><br>1 = Program memory code protection is disabled<br>0 = Program memory code protection is enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |       |      |        |        |     |    |       |      |      |       |       |       |
| bit 5     | <b>MCLRE:</b> RA3/MCLR pin function select bit <sup>(4)</sup><br>1 = RA3/MCLR pin function is MCLR<br>0 = RA3/MCLR pin function is digital input, MCLR internally tied to V <sub>DD</sub>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |       |      |        |        |     |    |       |      |      |       |       |       |
| bit 4     | <b>PWRT:</b> Power-up Timer Enable bit<br>1 = PWRT disabled<br>0 = PWRT enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |       |      |        |        |     |    |       |      |      |       |       |       |
| bit 3     | <b>WDTE:</b> Watchdog Timer Enable bit<br>1 = WDT enabled<br>0 = WDT disabled and can be enabled by SWDTEN bit (WDTCON<0>)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |      |        |        |     |    |       |      |      |       |       |       |
| bit 2-0   | <b>FOSC&lt;2:0&gt;:</b> Oscillator Selection bits<br>111 = RC oscillator: CLKOUT function on RA4/OSC2/CLKOUT pin, RC on RA5/OSC1/CLKIN<br>110 = RCIO oscillator: I/O function on RA4/OSC2/CLKOUT pin, RC on RA5/OSC1/CLKIN<br>101 = INTOSC oscillator: CLKOUT function on RA4/OSC2/CLKOUT pin, I/O function on RA5/OSC1/CLKIN<br>100 = INTOSCIO oscillator: I/O function on RA4/OSC2/CLKOUT pin, I/O function on RA5/OSC1/CLKIN<br>011 = EC: I/O function on RA4/OSC2/CLKOUT pin, CLKIN on RA5/OSC1/CLKIN<br>010 = HS oscillator: High-speed crystal/resonator on RA4/OSC2/CLKOUT and RA5/OSC1/CLKIN<br>001 = XT oscillator: Crystal/resonator on RA4/OSC2/CLKOUT and RA5/OSC1/CLKIN<br>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.

**4:** When MCLR is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.

**Legend:**

R = Readable

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

## *pic16f684.h* Configuration Word Definitions

```
// Configuration Mask Definitions
#define CONFIG_ADDR                0x2007
// Fail-Safe clock monitor
#define FCMEN                      0x3FFF
#define FCMDIS                    0x37FF
// Internal External Switch Over
#define IESOEN                    0x3FFF
#define IESODIS                   0x3BFF
// Brown-out detect modes
#define BOREN                     0x3FFF
#define BOREN_XSLP                0x3EFF
#define SBOREN                    0x3DFF
#define BORDIS                    0x3CFF
// Protection of data block
#define UNPROTECT                 0x3FFF
#define CPD                      0x3F7F
// Protection of program code
#define UNPROTECT                 0x3FFF
#define PROTECT                   0x3FBF
// Master clear reset pin function
#define MCLREN                    0x3FFF
#define MCLRDIS                   0x3FDF
// Power up timer enable
#define PWRTDIS                   0x3FFF
#define PWRTEN                    0x3FEF
// Watchdog timer enable
#define WDTEN                     0x3FFF
#define WDTDIS                    0x3FF7
// Oscillator configurations
#define RCCLK                     0x3FFF
#define RCIO                      0x3FFE
#define INTCLK                    0x3FFD
#define INTIO                     0x3FFC
#define EC                       0x3FFB
#define HS                       0x3FFA
#define XT                       0x3FF9
#define LP                       0x3FF8
```

## Configuration Word (con't)

- We will talk more about the Configuration Word in future lectures
- In the meantime ... use the following

```
__CONFIG(INTIO & WDTDIS & PWRTEN & MCLRDIS & UNPROTECT & BORDIS & IESODIS & FCMDIS);
```

*Note* ... \_\_ are two underscore characters

- From Register 12-1 and the pic16f684.h file we can see that we have set the following conditions:

## Configuration Word (con't)

```
__CONFIG(INTIO & WDTDIS & PWRTEN & MCLRDIS & UNPROTECT & BORDIS & IESODIS & FCMDIS);
```

- PICC Labels as defined in ... `pic16f684.h`
  - Internal oscillator
  - Disable Watchdog timer
  - 70 ms Power Up Delay Timer Enabled
  - `_MCLR` Pin Function Inactive/Pin is Input RA3
  - Program Memory Protect Enabled
  - Brownout Detect/Reset Disabled
  - Internal/External Switchover Mode Disabled
  - Fail-Safe Clock Disabled

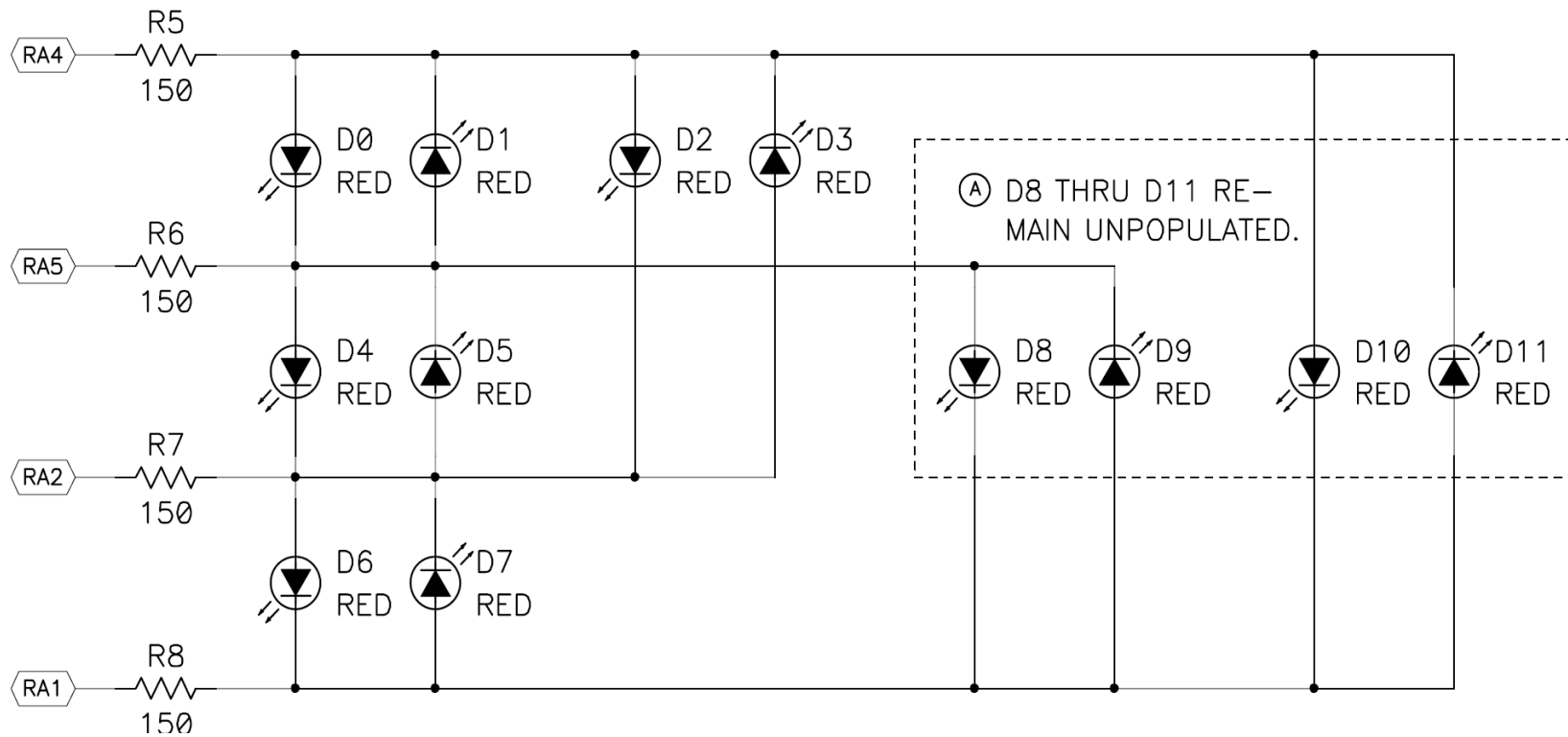
Lab



## Design Parameters

- Flash D0 of the PICkit 1 Starter Kit on and off
- Write the Code in C
- No other actions required

# Understand the Hardware



# PIC16F684 Pin Diagram

14-pin PDIP, SOIC, TSSOP

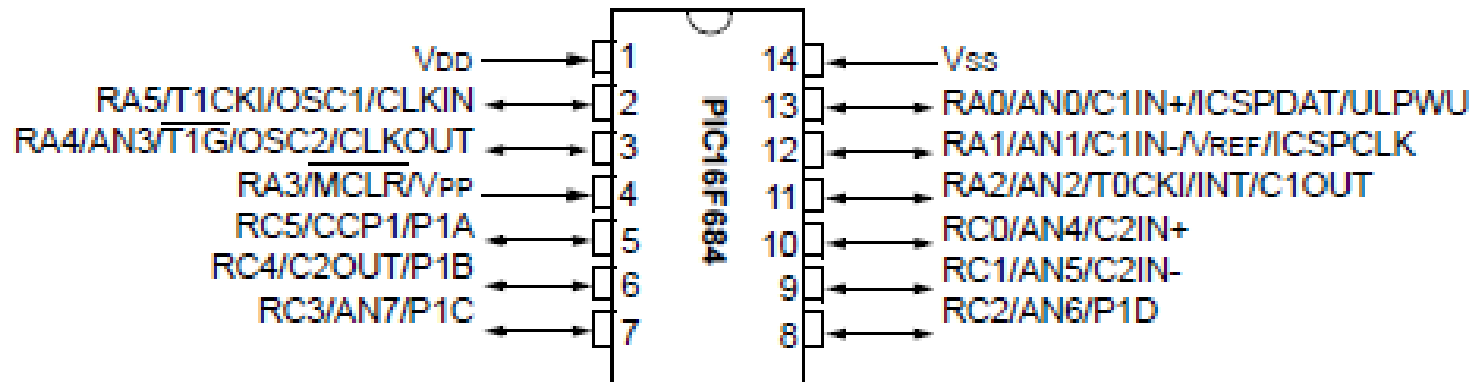


TABLE 1-1: PIC16F684 PINOUT DESCRIPTION

| Name                        | Function | Input Type | Output Type | Description                                              |
|-----------------------------|----------|------------|-------------|----------------------------------------------------------|
| RA0/AN0/C1IN+/ICSPDAT/ULPWU | RA0      | TTL        | CMOS        | PORTA I/O w/programmable pull-up and interrupt-on-change |
|                             | AN0      | AN         | —           | A/D Channel 0 input                                      |
|                             | C1IN+    | AN         | —           | Comparator 1 input                                       |
|                             | ICSPDAT  | TTL        | CMOS        | Serial Programming Data I/O                              |
|                             | ULPWU    | AN         | —           | Ultra Low-power Wake-up input                            |
| RA1/AN1/C1IN-/VREF/ICSPCLK  | RA1      | TTL        | CMOS        | PORTA I/O w/programmable pull-up and interrupt-on-change |
|                             | AN1      | AN         | —           | A/D Channel 1 input                                      |
|                             | C1IN-    | AN         | —           | Comparator 1 input                                       |
|                             | VREF     | AN         | —           | External Voltage Reference for A/D                       |
|                             | ICSPCLK  | ST         | —           | Serial Programming Clock                                 |
| RA2/AN2/T0CKI/INT/C1OUT     | RA2      | ST         | CMOS        | PORTA 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                                       |
|                             | C1OUT    | —          | CMOS        | Comparator 1 output                                      |
| RA3/MCLR/VPP                | RA3      | TTL        | —           | PORTA input with interrupt-on-change                     |
|                             | MCLR     | ST         | —           | Master Clear w/internal pull-up                          |
|                             | VPP      | HV         | —           | Programming voltage                                      |
| RA4/AN3/T1G/OSC2/CLKOUT     | RA4      | TTL        | CMOS        | PORTA I/O w/programmable pull-up and interrupt-on-change |
|                             | AN3      | AN         | —           | A/D Channel 3 input                                      |
|                             | T1G      | ST         | —           | Timer1 gate                                              |
|                             | OSC2     | —          | XTAL        | Crystal/Resonator                                        |
|                             | CLKOUT   | —          | CMOS        | Fosc/4 output                                            |
| RA5/T1CKI/OSC1/CLKIN        | RA5      | TTL        | CMOS        | PORTA 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            |
| RC0/AN4/C2IN+               | RC0      | TTL        | CMOS        | PORTC I/O                                                |
|                             | AN4      | AN         | —           | A/D Channel 4 input                                      |
|                             | C2IN+    | AN         | —           | Comparator 2 input                                       |
| RC1/AN5/C2IN-               | RC1      | TTL        | CMOS        | PORTC I/O                                                |
|                             | AN5      | AN         | —           | A/D Channel 5 input                                      |
|                             | C2IN-    | AN         | —           | Comparator 2 input                                       |
| RC2/AN6/P1D                 | RC2      | TTL        | CMOS        | PORTC I/O                                                |
|                             | AN6      | AN         | —           | A/D Channel 6 input                                      |
|                             | P1D      | —          | CMOS        | PWM output                                               |
| RC3/AN7/P1C                 | RC3      | TTL        | CMOS        | PORTC I/O                                                |
|                             | AN7      | AN         | —           | A/D Channel 7 input                                      |
|                             | P1C      | —          | CMOS        | PWM output                                               |
| RC4/C2OUT/P1B               | RC4      | TTL        | CMOS        | PORTC I/O                                                |
|                             | C2OUT    | —          | CMOS        | Comparator 2 output                                      |
|                             | P1B      | —          | CMOS        | PWM output                                               |
| RC5/CCP1/P1A                | RC5      | TTL        | CMOS        | PORTC I/O                                                |
|                             | CCP1     | ST         | CMOS        | Capture input/Compare output                             |
|                             | P1A      | —          | CMOS        | PWM output                                               |
| Vss                         | Vss      | Power      | —           | Ground reference                                         |
| VDD                         | VDD      | Power      | —           | Positive supply                                          |

Legend: TTL = TTL input buffer, ST = Schmitt Trigger input buffer, AN = Analog input

## Circuit Evaluation

- LED “DO” is connected between RA5 and RA4
- Pin 2 ... RA5
- Pin 3 ... RA4
- A high on RA4 and a low on RA5 will energize the LED
- We are using PORTA as our I/O Port ... in this case as an output

## ***Lab\_1A.c*** Evaluation (sheet 1 of 5)

```
#include <pic.h>

/*****
*
* Program Title:   Flash D0
*
* Program File Name: Lab_1A.c
*
* Microprocessors A 17.383
*
* xxxxxxxx - Put in Semester (i.e. Fall 2010) here
*
* xxxxxxxx - Put in your name here
*
* xx/xx/xx - Put date here
*
*****/
```

## ***Lab\_1A.c*** Evaluation (sheet 2 of 5)

```

/*****
* Function: PORT_init
*
* Description: Initializes PORTA to a known condition
*
* Notes:  None
*
* Returns: None
*
*****/
void PORTA_init(void)
{
    PORTA = 0;           // All PORTA Pins are low
    CMCON0 = 7;          // Turn off Comparators
    ANSEL = 0;           // Turn off ADC
    TRISA = 0b001111;    // RA4 & 5 are outputs; RA0,1,2, and 3 are input

    return;

}
/***** END OF PORTA_init *****/

```

## ***Lab\_1A.c*** Evaluation (sheet 3 of 5)

```

/*****
* Function: delay_routine
*
* Description: Causes a delay in program execution
*
* Notes:
*
* Delay was determined through trial and error
*
* Returns: None
*
*****/
void delay_routine(void)
{
    int i, j;

    for (i = 0; i < 255; i++)
        for (j = 0; j < 255; j++);

    return;
}
/***** END OF delay_routine *****/

```



## ***Lab\_1A.c*** Evaluation (sheet 4 of 5)

```
/******  
* Function: main  
*  
* Description: D0 on PICkit 1 will Flash on and off  
*  
* Notes:  
*  
* RA4 - Positive LED Connection for D0  
* RA5 - Negative LED Connection for D0  
*  
* Returns: This routine contains an infinite loop  
*  
*****/  
  
/* Configuration Word */  
  
__CONFIG(INTIO & WDTDIS & PWRTEN & MCLRDIS & UNPROTECT \  
& UNPROTECT & BORDIS & IESODIS & FCMDIS);
```

## ***Lab\_1A.c*** Evaluation (sheet 5 of 5)

```
main()
{
    PORTA_init();

    while(1 == 1)           // Loop Forever
    {
        delay_routine();

        RA4 = 1;           // D0 LED On by making RA4 high

        delay_routine();

        RA4 = 0;           // D0 LED Off by making RA4 low
    }                       // *** END OF While (1 == 1) LOOP

    return;

}

/***** END OF main ROUTINE *****/
```

# PICKit 1 LEDs

## PICKit 1 LEDs

To Light LED Dx, RAX values

|    | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 |
|----|-----|-----|-----|-----|-----|-----|
| D0 | 0   | 1   |     |     |     |     |
| D1 |     |     |     |     |     |     |
| D2 |     |     |     |     |     |     |
| D3 |     |     |     |     |     |     |
| D4 |     |     |     |     |     |     |
| D5 |     |     |     |     |     |     |
| D6 |     |     |     |     |     |     |
| D7 |     |     |     |     |     |     |

# PICKit 1 LEDs

PORT  
A

|    | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | PORTA (binary) | PORTA (HEX) |
|----|-----|-----|-----|-----|-----|-----|----------------|-------------|
| D0 | 0   | 1   |     |     |     |     | 0b010000       | 0x10        |
| D1 |     |     |     |     |     |     |                |             |
| D2 |     |     |     |     |     |     |                |             |
| D3 |     |     |     |     |     |     |                |             |
| D4 |     |     |     |     |     |     |                |             |
| D5 |     |     |     |     |     |     |                |             |
| D6 |     |     |     |     |     |     |                |             |
| D7 |     |     |     |     |     |     |                |             |

# PICKit 1 LEDs

TRISAx - 0 is an output; 1 is an input;

|    | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | TRISA (binary) | TRISA (HEX) |
|----|--------|--------|--------|--------|--------|--------|----------------|-------------|
| D0 | 0      | 0      | 1      | 1      | 1      | 1      | 0b001111       | 0x0f        |
| D1 |        |        |        |        |        |        |                |             |
| D2 |        |        |        |        |        |        |                |             |
| D3 |        |        |        |        |        |        |                |             |
| D4 |        |        |        |        |        |        |                |             |
| D5 |        |        |        |        |        |        |                |             |
| D6 |        |        |        |        |        |        |                |             |
| D7 |        |        |        |        |        |        |                |             |

**Next Week ...**

## Next Class Topics

- Switches
- C commands
- Number systems
- Start Lab #2

# HomeWork



# Homework

1. Send an email with your email address or addresses (for class distribution list), if not already sent
2. Optional - Read the C commands discussed in today's lecture found in any Programming in C text
3. Read the PIC16F684 data manual sections for the areas encountered tonight
4. Finish [Lab #1](#) and prepare for Lab #2, which will be on the web site
5. Work on Lab #1 Report ... due in two weeks (September 28, 2010)

**Time To ...**  
**Start the Lab ...**

# References

1. PIC16F684 Data Sheet 41202F
  - a) Pages 7 – 18 (memory organization)