

EE361 MICROCOMPUTER-BASED  
DIGITAL DESIGN  
LESSON OBJECTIVES (VERSION 2)  
FALL 2009

CAPT Charles B. Cameron

September 11, 2009

1. Introduction and Overview

- Understand the course policy, including grading.
- Understand the purpose of the course.

2. Digital Logic Review

- Be able to use and explain binary number systems.
- Be able to use and explain ASCII code.
- Be able to create Gray codes.
- Be able to use and explain Boolean algebra.
- Be able to use Karnaugh maps to simplify Boolean algebraic expressions.
- Be able to read and write Boolean algebraic expressions in Sum of Products form using conventional, typographic, and summation formats.
- Be able to use and explain AND, OR, NOT, NAND, NOR, XOR, NXOR logic gates; SR flip-flops; JK flip-flops; D flip-flops; serial registers; parallel registers; multiplexers; demultiplexers; and counters.

### 3. PIC16F884 Architecture

- Be able to explain how data flows inside the PIC16F884.
- Be able to explain the purpose of the W register, flash program memory, RAM, EEPROM, timers, A/D converters, stack, and I/O ports of the PIC16F884.
- Be able to explain how memory is organized in the PIC16F884.
- Be able to explain how to refer to and configure the special function registers in the PIC16F884.
- Be able to explain the use of the program counter in the PIC16F884.
- Be able to explain direct and indirect addressing in the PIC16F884.

### 4. Machine Language and Assembly Language

- Be able to read and interpret the instruction descriptions in the manual.
- Be able to combine instructions to achieve a goal.
- Be able to construct a program source file.
- Be able to explain how symbolic addresses relate to numerical addresses.

### 5. Arithmetic Instructions and Computer Arithmetic

- Be able to use and explain ASCII, BCD, and two's-complement data formats.
- Know how to convert from one form of data to another.
- Know the arithmetic instructions of the PIC16F884 microcontroller.

### 6. Logic Instructions

- Know the logic instructions of the PIC16F884 microcontroller.

### 7. Testing for Conditions

- Know the bit-testing instructions of the PIC16F884 microcontroller.

- Know how to set, reset, and test collections of bits with the PIC16F884 microcontroller.

#### 8. PIC16F884 I/O Ports

- Be able to explain the block diagrams of the I/O ports in the PIC16F884.
- Be able to initialize I/O ports in the PIC16F884.
- Be able to use tri-state and open-collector connections.

#### 9. Subroutines, Functions, and Branching

- Know the distinction between subroutines, functions, and branching.
- Know the rôle of the hardware stack in subroutines, functions, and interrupts with the PIC16F884 microcontroller.
- Know the instruction-flow-control instructions of the PIC16F884 microcontroller.

#### 10. Light-Emitting Diodes (LEDs) and 7-Segment Displays

- Know how to connect LEDs and 7-segment displays to a microprocessor.
- Know how to operate LEDs and 7-segment displays under program control.
- Be able to predict and control the amount of current flowing through an LED or a 7-segment display.

#### 11. Debugging with the Debugger and the Oscilloscope

- Be able to connect the oscilloscope to a circuit in order to analyze its behavior.
- Be able to modify a program so that it generates a suitable oscilloscope trigger so that the oscilloscope display is stable and useful.

#### 12. Structured Programming

- Know how to combine instructions to implement high-level language control structures.

### 13. Macros

- Be able to write assembly-language macros for common sequences of instructions.

### 14. Handling Interrupts

- Understand the interrupt scheme in the PIC16F884.
- Be able to configure interrupts and write interrupt service routines for them.

### 15. Stepper Motors

- Understand how stepper motors work.
- Be able to take a stepper motor without a circuit diagram, create one, and connect it for digital control.
- Be able to generate signals to drive a stepper motor in either direction for an arbitrary number of steps or half-steps at an arbitrary rate.

### 16. PIC16F884 Timing Programs

- Be able to write programs to delay for an arbitrary amount of time by repeated execution of `nop` instructions.

### 17. PIC16F884 Timers 0 and 1

- Be able to initialize and operate Timer 0 and Timer 1 in the PIC16F884 using two methods: polling and interrupts.

### 18. Input from Switches, Knobs, Sliders, & Sensors

- Be able to input data from switches and potentiometers using the PIC16F884.
- Be able to measure switch bounce time.
- Be able to design switch debouncing circuits.

### 19. C: Review of Types, Operators, and Expressions

- Be able to explain the sizes of C characters, integers, floats, and doubles on the PIC processor.

- Be able to use C constants, expressing them in hexadecimal as well as conventional notation.
- Be able to explain how strings are stored in the PIC processor's memory.
- Know and be able to use C's arithmetic, relational, logical, increment, decrement, and bitwise logical operators.
- Understand and be able to explain why converting one data type into another may result in erroneous results.
- Know how to determine the order of precedence of operators in C.
- Be able to use the C Preprocessor for defining constants and macros.

#### 20. C: Review of Control Flow (Conditions, Switches, and Loops)

- Be able to use and explain the meaning of statements blocks.
- Be able to use `if-then-else` statements.
- Be able to use `switch` statements.
- Be able to use `while`, `do-until`, and `for` loops.
- Be able to use `break` statements.
- Be able to explain when a `goto` statement should be used and how to avoid it when it should not be used.

#### 21. C: Review of Functions and Programs

- Know and be able to explain the difference between C functions and subroutines.
- Know and be able to explain the use of storage-class specifiers `auto`, `static`, `extern`, `register`, and `typedef`.
- Know and be able to explain the scope rules for variables.
- Know how to initialize variables.
- Be able to explain while recursion is a risky strategy to use in the PIC processor.

#### 22. C: Review of Pointers and Structures

- Know and be able to explain the relationship between pointers, addresses, and arrays.
  - Know how to declare multidimensional arrays.
  - Know how to define and use a pointer to a function.
23. Using C with the PIC Processor
- Know how to use the header file that defines the special function registers of the PIC to the C compiler.
  - Know how to use C to initialize the PIC's registers.
  - Know how to use the debugger to step through the execution of a C program executing on the PIC.
24. Combining Assembly Language with C
- Know how to use assembly language for an interrupt service routine with a C program for the main loop.
25. A/D & D/A Conversion on the PIC16F884
- Be able to explain quantization, quantization error, and the relationship between input and output of analog-to-digital and digital-to-analog converters.
  - Be able to configure and use the A/D converter in the PIC16F884 using two methods: polling and interrupts.
27. PIC16F884 Timer 2 and Capture/Compare
- Be able to initialize and operate Timer 2 and the Capture/Compare module in the PIC16F884 using two methods: polling and interrupts.
28. Comparator Module on the PIC16F884
- Be able to use the PIC16F884 Comparator Module to respond to an incoming analog signal of sufficiently high voltage.
29. Using PWM for D/A Conversion
- Be able to use the PIC16F884 to transmit a PWM signal.

- Be able to use a PWM signal to convert a digital value to an analog signal level.
30. Asynchronous Serial I/O on the PIC16F884
- Be able to explain how asynchronous serial I/O works.
  - Be able to configure the USART to make asynchronous serial I/O work in the PIC16F884.
32. Synchronous Serial I/O on the PIC16F884
- Be able to explain how synchronous serial I/O works.
  - Be able to configure the USART to make synchronous serial I/O work in the PIC16F884.
  - Be able to use the SPI interface standard.
  - Be able to use the I<sup>2</sup>C interface standard.
34. Memory Types
- Be able to explain the operation, characteristics, and general design of ROM, PROM, EPROM, EEPROM, flash memory, RAM, DRAM, magnetic tape, magnetic disk, and CD-ROM.
35. Microprocessor-Based System
- Understand how to design a microprocessor-based system.
38. Using EEPROM in the PIC16F884
- Be able to initialize and use the EEPROM device in the PIC16F884.
  - Be able to list three uses of an EEPROM.
39. Memory: Error detection & correction
- Know how to create even and odd parity bits.
  - Know how to calculate a CRC code for any CRC polynomial and how to determine whether errors have occurred or not.
  - Know how to create Hamming error detection and correction codes.
40. Watchdog Timers

- Be able to explain the purpose of a watchdog timer.
- Be able to use the watchdog timer on the PIC16F884.