Review Lecture

This lecture summarizes the topics covered in the course.

C Programming

Most of the course dealt with programming in C.

You should be able to do the following:

- explain, identify and give examples of the following terms: file, file name, file type, compiler, source, executable, variable, statement, operator, precedence
- write a simple C program including a `main()` and other function declarations, integer and array declarations, and statements.
- predict the result of executing such a program
- indent your code according to K&R conventions
- use symbolic constants in C programs
- write a C program to implement a state machine.

Exercise: Write a complete C program that prints the integers from 1 to 10. Use a 'for' loop.

Expressions

You should be able to evaluate expressions involving variables; character, hexadecimal and decimal constants; and all of the operators covered in the course.

Exercise: What is the value of the expression `y = ( 6 3̂ ) * 2`?

Exercise: Write an expression that is zero if the second least-significant bit of the variable `x` is non-zero.

Flow of Control

You should be able to use `if/else` statements, `while` loops and `for` loops.

Exercise: How many times will the statements in the body of the following 'for' loop be executed: `for(i=7 ; i ; i=i/2)`?

Arrays

Arrays allow us to use a numerical "index" value to access individual elements in a table of values.

You should be able to:

- declare an array
- write expressions that reference an array element
- write C code that iterates over all of the elements of an array using either an element count of a terminator array value
- define the terms array and index

Exercise: Write C code that counts the number of elements in an array `x` that are between 10 and 20. You do not know the length of the array. The last element of the array has a value -1.

Functions

Functions allow us to divide a program into smaller parts. By using parameters and arguments a function can also be used to perform the same operation on different data.

You should be able to:

- declare a function including function argument types and return values
- give the values of function arguments for a given function call
- define the terms function, argument, parameter and return value

Exercise: Write the function declaration for a function named `f` that takes two integer array arguments `a` and `b` and returns a character value.

Binary, Hexadecimal and Decimal Notation

Binary and hexadecimal notation is often more convenient than decimal.

You should be able to convert an integer value between decimal, binary and hexadecimal representations.

Exercise: Convert the value 0x83 to binary and decimal. Convert 67 (decimal) to hexadecimal. What are the the binary, hex and decimal values of the C character constant 'X'?

Programming Methods

Stepwise decomposition and flowcharts are two techniques that can be used to help write computer programs.

You should be able to: (1) reduce a familiar but complex procedure into a sequence of simpler steps, (2) describe an algorithm using a flowchart, and (3) convert between a flowchart and C code.

Microcomputer Architecture

Microcomputers are built from CPU, memory and I/O chips.

You should be able to: (1) show how the following buses and signals are connected in a microcomputer system: power and ground, address and data buses, read and write strobes, and chip enables; (2) give the sequence of signals that must appear on the address, data, and control lines of a memory or I/O chip in order to read or write a particular data value to/from a particular address; (3) explain the purpose of these lines; and (4) compute the number of address lines required for a given memory size or vice-versa.

You should also be able to explain how a grouping of logic-level signals (a bus) can be used to represent or transmit a number or a character.

Exercise: At what point in a read cycle does a CPU's address bus act as an input?

A/D and D/A Converters

A/D and D/A converters are used to convert between analog voltages and the binary numbers used by computers.

You should be able to specify the required number of bits of accuracy for a given voltage range and required accuracy.

Exercise: What is the smallest possible output voltage step for a 16-bit D/A converter that has a range of $\pm$ 3 volts?

You should also be able to state the purposes of the following circuits: sample-and-hold, multiplexer, differential amplifier, and low-pass filter.

Serial and Parallel Interfaces

Serial interfaces are typically used to connect computer systems and low-speed external peripherals such as modems and printers.

You should be able to describe the operation and format of data and handshaking signals on an RS-232 interface.

Parallel I/O ports transfer one word at a time between the CPU and a peripheral. A parallel interface usually involves additional "handshaking" lines and a well-defined protocol to control the transfer of data. Parallel interfaces are used to transfer data with higher-speed peripherals such as printers. We will study one simple example of a parallel interface, the parallel printer interface.

You should be able to: (1) describe the operation of a parallel printer interface; and (2) write C code to read and write the individual bits of an I/O port.

Exercise: Give C expressions that will set and clear bit 3 of a variable x.

State Machines

State machines allow us to unambiguously describe the behaviour of a controller in a simple way.

You should be able to design a state machine from a description of its behaviour. You should be able to describe the state machine using state transition diagrams and tables.