# Final Exam Review

*This is a list of the topic covered in the course.*

The final exam is a 3-hour open-book exam. Material from before the mid-term exam will be included. Any of the material covered in the lectures (except the final lecture on the 8088 architecture) may be included.

The format of the questions will be similar to those given in the assignments and may include design and analysis questions (e.g. C programming and state machine design) as well as short numerical problems (e.g. power electronics), short-answer (e.g. reliability) and multiple-choice (e.g. motors, sensors).

On the final exam you should be able to:

- give examples of computer control systems and identify their inputs and outputs.
- 
  - explain the following terms: variable, statement, operator, precedence
  - evaluate expressions involving integer variables, constants, and the operators described in this lecture
  - write a simple C program including a main() function declaration, integer variable declarations, and the following statements: expression, if/else, while, and for
  - predict the result of executing such a program
  - 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
  - declare a function including function argument types and return values
  - give the values of function parameters for a given function call
  - define the terms array, index, function, argument, and return value
- convert a decimal value to and from binary, convert a binary value to and from hexadecimal, evaluate expressions that use C's bitwise logical and shift operators, and convert between bus signal levels and a numeric value.
- design a state machine from a description of its behaviour.
- describe the state machine using state transition diagrams and tables.
- design a combinational or sequential logic circuit from a description of its behaviour.
- (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.
- (1) design simple input, output and bidirectional I/O ports using registers, tri-state buffers and and open-collector buffers; (2) describe the operation of a parallel printer interface; and (3) write C programs to read and write the individual bits of an I/O port.
- describe the operation and format of data and handshaking signals on an RS-232 interface.
- specify the required number of bits of accuracy for a given voltage range and maximum allowed error.
- state the purposes of the following circuits: sample-and-hold, multiplexer, differential amplifier, and low-pass filter.

- design a transistor power switch including determining minimum $V_{CE}$ or $V_{DS}$, minimum current gain or $V_{GS}$, power dissipation, and maximum heat sink thermal resistance.
- analyze the operation of, and draw schematics for power control circuits using SCRs, diacs and triacs.
- use the CUPL logic design language to design a simple PAL-based circuit.
- compute the important specifications and select the appropriate type(s) of motor for a given application (the motors covered were: synchronous, squirrel-cage, universal, shunt DC, series DC, permanent-magnet DC and stepper)
- select an appropriate type of transducer for a given application.
- differentiate between availability and reliability.
- read a description of a system failure and identify and classify the fault, identify the resulting error and describe strategies for reducing the likelihood of the faults.