Mid-Term Exam Review

This lecture lists the topics that might be included on the mid-term exam.
It is assumed you will bring (and are familiar with the contents of) the lecture notes (lec1 to lec9), the two two-page VHDL summaries, and a summary of the 8086 instruction set. These are available on the course web page if you don't already have them.

For the mid-term exam you should be prepared to:

- convert an informal description of the behaviour of a combinational logic circuit into a truth table, a sum of products boolean equation and a schematic.
- convert an informal description of a combinational logic circuit into a VHDL entity and architecture.
- design a state machine from an informal description of its operation.
- write a VHDL description of a state machine.
- draw timing diagrams for the processor bus signals described in Lecture 5 and state the values that would appear on the data and address buses. You should be able to do this for memory and I/O read and write cycles and for interrupt acknowledge cycles.
- for the 8086 processor:
  - write a simple program in 8086 assembly language including: (1) transfer of 8 and 16-bit data between registers and memory using register, immediate, direct, and register indirect addressing, (2) some essential arithmetic and logic instructions on byte and 16-bit values, (3) stack push/pop, (4) input/output, (5) conditional and unconditional branches, (6) call/return, (7) interrupt/return, (8) essential pseudo-ops (org, db, dw).
  - compute a physical address from segment and offset values,
  - describe the response of the 8086 CPU to NMI, software (INT) and external (IRQ) interrupts and return from interrupts.
- use some advanced features of VHDL including:
  - make library packages visible
  - declare components and save them in packages
  - instantiate components into an architecture
  - declare std_logic, std_logic_vector, signed and unsigned signals
  - declare enumerated types and subtypes of array types and save them in packages
  - use conditional signal assignments
  - instantiate tri-state outputs
- design complex circuits using an RTL approach, including:
  - classify a VHDL description as a behavioral, structural, or dataflow (RTL) description
  - identify the registers and logic/arithmetic functions required to implement a particular algorithm
  - partition this algorithm into a sequence of these operations and register transfers
  - write synthesizeable VHDL RTL code to implement the algorithm
- select the appropriate type of memory device for different applications, combine memory ICs to form memory arrays, and design address decoders.