Notes about the ENGR-433 exam 2

(Note: in the paragraph below the phrase "there will be" is used repeatedly. Interpret that to mean some probability of what follows will appear on the exam remembering that while the probability is high the possibility of it not appearing does exist).

This exam will emphasize digital design using VHDL.

There will be a problem(s) on this exam where a description of desired function is given and you will design the circuit, possibly including a state machine for which a state diagram will need to be created, and describe it with VHDL. There will be a problem(s) for which a circuit is given and you will write a VHDL description. There will be a problem(s) for which a VHDL description is given and you need to figure out what it does. There will be a problem(s) for which a circuit diagram or functional description (such as a state diagram) is given along with a timing diagram of incoming signals and you will extend the timing diagram to detail timing of other signals in the circuit. Knowledge regarding basic VHDL data types, as defined in the IEEE standard, such as std_logic, std_logic_vector, unsigned, and signed is assumed. You should understand structural and behavioral VHDL methods of description.

When using If Then or Case constructs in a process it is possible (even easy) to create implied latches that were not intended. Know how to correctly use If Then and Case to avoid creating memory when you don't intend to.

Having done these things in homework and labs, you should know how basic digital logic blocks function and be able to do the following using VHDL:

- Write an entity and architecture to create a component or overall design
- Instantiate a VHDL component in an architecture. pg31-32 & 475-
- Create combinational logic using concurrent signal assignment statements
- Know the conditional signal assignment statement. pg 72
- Know the selected signal assignment statement. pg 85
- Know the syntax for common sequential statements such as
  - If Then pg.103
  - Case pg. 113

  and the fact that these must be in the context of a process
- Know how to create a synchronous state machine. pg. 330-332 is good example
- Know how to initialize a synchronous state machine to a known state (typically zero)
- Be able to create the following logic circuits:
  - Edge triggered D F/F pg. 222
  - Edge triggered D F/F with enable pg 227. Note how enable is implemented.
  - Edge triggered T F/F pg 228-229
  - Counters, binary or decade, with enable and asynchronous reset. Listing 9.4 pg261
  - How to create a square wave output from a counter (we do this in our labs)
  - Multiplexer pg 72-73, 85
  - Binary decoder pg 73-74, 86
  - Shift register pg. 231-232