CPET-233 Final Exam

Monday, December 16, 2019 4:15-6:45 PM

The exam will be cumulative and a mix of multiple choice, short answer and coding questions.

You may bring a double-sided handwritten note sheet and a calculator

The following syntax guide will be provided:

VHDL Syntax Sheet

Copyright: 2007 Bryan J. Mealy

Concurrent Statements		**Sequential Statements**
Concurrent Signal Assignment (dataflow model)		**Signal Assignment**
`target <= expression;`		`target <= expression;`
`A <= B AND C;` `DAT <= (D AND E) OR (F AND G);`		`A <= B AND C;` `DAT <= (D AND E) OR (F AND G);`
Conditional Signal Assignment (dataflow model)		***if* statements**
`target <= expressn when condition else` `expressn when condition else` `expressn;`		`if (condition) then` `{ sequence of statements }` `elsif (condition) then` `{ sequence of statements }` `else --(the else is optional)` `{ sequence of statements }` `end if;`
Selective Signal Assignment (dataflow model)		***case* statements**
`with chooser_expression select` `target <= expression when choices,` `expression when choices;`		`case (expression) is` `when choices =>` `{sequential statements}` `when choices =>` `{sequential statements}` `when others => -- (optional)` `{sequential statements}` `end case;`

- Know binary and hexadecimal numbering systems. Be able to convert between decimal and the other two. There is a powerpoint presentation on number systems posted with this study guide

- Lecture 1 - Be familiar with the digital review. Be sure to know all of the gate and how they function.

- Lecture 2 – Know how to turn on specific segments in a common anode 7-segment display. Know which bit in the HEXn output vector correspond to the segments a-g. Understand waveform simulation and how to read a waveform to verify if an output is correct. Know the purpose of simulation. Understand what synthesis is.

- Lecture 3 - Know what VHDL is, what it is used for, and the main parts of a module. Know the purpose for the entity and the architecture and the information contained in each. You do not have to memorize libraries. Know what signals are and how and where they are declared.

- Lecture 4 - Understand the data types STD_LOGIC and STD_LOGIC_VECTOR and what they represent. Know how to write:
    - Simple assignment statement
    - Selected signal assignment
    - Conditional signal assignment

  Be sure that you understand the concept of concurrency with regards to VHDL.

- Lecture 5 – Know what a constant is and how to declare a constant. How does a constant differ from a signal (in use and assignment)? Understand what a process is and why it is used. How are statements evaluated inside a process? When are signal inside a process assigned their value? Know what a sensitivity list is and which signals belong in the sensitivity list. Be able to write:
    - Case statement
    - If/then/else statement

- Lecture 6 - You will not have to write or use a test bench. However, you should be familiar with what a test bench is, what it is used for and how it is different from synthesizable VHDL. What statements can you use in a testbench that you cannot use in synthesizable VHDL.

- Lecture 7 – Understand the difference between std_logic_vector and integer in terms of synthesis. Understand how negative numbers are represented internally. Be able to convert a signed binary number to its decimal equivalence and how to negate a number. Know the range of possible values for an n-bit signed number and n-bit unsigned number. Understand what casting is and how to cast between std_logic_vector, signed and unsigned. Know the rules of VHDL addition, subtraction, multiplication and division. Be able to sign extend to create operands that are the correct length for the given operation. Know what an ALU is and where one can be found.

- Lecture 8 – this lecture is very important. It's a summary of some very important topics. Know which signals go in a sensitivity list
    - Understand when signals are assigned in a process and why the following code will not work as expected:

```
PROCESS(a_unsign, b_unsign)
   BEGIN
         a_unsign <= unsigned(a);
         b_unsign <= unsigned(b);
   if (a_unsign < b_unsign) then
         altb <= '1';
  else
         altb <= '0';
  end if;
end process;
```

    - Understand how to write case and if/then/else statements so that a latch is not inferred and what is meant by one output per process. Know what it means that an output port cannot be read and how to get around that restriction. Know how to create a vector with concatenation and how to access individual bits of a vector. Be comfortable with the parts of a test bench and how a test bench relates to the module being tested

- Lecture 9 and 9.5 - Understand what is meant by hierarchical design and advantages of using it. Know what a component is and how to declare it in an architecture. Know how to write a structural architecture with components. Know what a port map is and the syntax for one.

- Lecture 10 - Be able to identify what a comparator, decoder, encoder, multiplexer and code converter all do. Understand the relationship between number of inputs, outputs and selects for decoders, encoders and multiplexers. Understand how the VHDL is written to implement each of the above. Understand what a priority encoder is and how priority is determined
- Lecture 10.5 - Be sure you can answer all of these questions

- Lecture 11 - Understand how to write VHDL for synchronous systems. Be able to create a synchronous VHDL with a synchronous or asynchronous reset. Be sure you know what it means. Be able to write the VHDL for a flip flop. Understand what a latch is, how it is inferred and how it differs from a flip-flop

- Lecture 12 - Understand what a register is and how to write the VHDL for a register of any size. Know how to create a counter of any size that counts either up or down. Know how to create a counter that stops at any value and how to use a counter to create a delay. Understand how an enable works for any synchronous component and be able to create a synchronous component with an enable. Be familiar with BCD counters and how they differ from binary counters - think about why you would use one over the other. Know the relationship between the number of flip-flops in a counter and its max count value.

- Lecture 13 - Understand the difference between a regular register and a shift register in terms of operation and design. Know how shift registers work and how to create one in VHDL. Be sure you know how to load and how to shift in either direction. Be sure to understand the difference between serial and parallel data and how a shift register is used to convert between the two formats. Pay careful attention to the feedback on HW 9.

- Lecture 14 - Know how to declare an array type for a one-dimensional and a two-dimensional array. Understand what arrays are and how to assign values to a complete array and individual elements in an array. Know how to create a lookup table with an array. You will not have to create RAM, but you need to understand how a two-dimensional array is used to create one. Understand the relationship between the size of a RAM's data bus, address bus and storage capacity.

- Lecture 15 – 17 -Review carefully the lecture notes on state machines. Understand the difference between Mealy and Moore state machines. You will not have to design a Mealy state machine. Understand the relationship between the state machine VHDL code and the synthesized hardware, although you will not have to design a state machine with gates (only VHDL). Understand what one-hot encoding means and the relationship between number of states and number of flip-flops. For the VHDL, understand enumerated types and also the purpose of the 3(+) processes used in a state machine. Be able to write all 3(+) processes. Be able to create a state transition diagram and write VHDL code from a given state transition diagram

- Lecture 18 - Understand what generic components are and be able to write the code for a generic component. Be able to instantiate a generic component for a specific size. Review feedback on HW #12. Know the syntax for loops and be able to write a simple loop structure. Understand how nested loops work and how to exit a loop or skip to the next iteration.

- Lecture 19 – Understand what variables are and how they differ from signals. Know where in the code variables are declared and the range of their visibility. Understand what is meant by one output per process. Be familiar with all of the coding situations that can cause latches. Understand what synthesis and RTL are.

- Lecture 20 – Understand the difference in how variables and signals are updated in a process. Be familiar with attributes and know what they are used for. Know the Array (A') attributes.

- Lecture 21 - Understand what metastability is, what causes it and how it can be mitigated. Be familiar with the synchronization circuit and other uses for it

- Labs 1-11
- HW 1-12
- Book Sections:

Ch. 1, Ch. 2 sections 1-5
Ch. 3 section 6, Ch. 5 sections 1-4
Ch. 6 sections 3-4, 7-9
Ch. 10 sections 1-2, 5-8
Ch. 3 sections 7 & 18, Ch. 5 section 7
Ch. 8 section 3
Ch. 6 section 2
Ch. 13, sections 1-2, 4-5
Ch. 11 Sections 1-4
Ch. 2 sect 6, Ch. 8 sect 4, Ch. 6 sect 6
Ch. 2 section 8, Ch. 7 sections 1-4
Ch. 12 section 4