

ELEC 311 Test 2 Review

TOPICS

Combinational Logic Circuits

- Multiplexer
- Tri-State Buffer
- Decoder, (Priority) Encoder
- Read Only Memory (ROM)
- Programmable Logic Devices (CPLD, FPGA)

VHDL

- Entity, Architecture
- Types, Operators
- Libraries and Packages
- Structural Model
 - Component, Port Map
- Dataflow Model
 - Concurrent Signal Assignment Statements

Latches and Flip-Flops

- Set-Reset Latch, D Latch
- D Flip-Flop, J-K Flip-Flop, T Flip-Flop
- Timing Diagrams

Registers and Counters

- Register, Shift Register
- Tri-State Bus
- Counters
- Sequential Design

Sequential Analysis

- State Tables
- State Graphs
- Timing Diagrams
- Next State (FF) Equations
- Transition Tables

Sequential Design

- State Graphs
- State Tables
- Transition Tables
- Next State (FF) Equations

Practice Questions

1. Given a 1-to-2 binary decoder with input I, active low enable E, and active low outputs Y:
 - a. Draw the logic symbol and label all signals.
 - b. Fill in a truth table (using don't cares where appropriate).
 - c. Write a VHDL entity (using std_logic_vectors where appropriate).
2. Implement the following function using a 4-to-1 multiplexer:
$$F(w,x,y,z) = \Sigma m(3,4,5,7,10,14) + \Sigma d(1,6,8,15)$$
3. Determine the ROM size and contents to implement the function for the square of the integers from 0 to 8.
4. Given the following VHDL model, draw the schematic diagram (label completely).

```
entity MUX4 is
  port (D : in std_logic_vector(3 downto 0);
        S : in std_logic_vector(1 downto 0);
        Y : out std_logic);
end MUX4;
architecture STRUCTURE of MUX4 is
  signal YD : std_logic_vector(1 downto 0);
  component MUX2
    port (D : in std_logic_vector(1 downto 0);
          S : in std_logic;
          Y : out std_logic);
  end component;
begin
  M1 : MUX2 port map (D(3), D(2), S(0), YD(1));
  M2 : MUX2 port map (D(1), D(0), S(0), YD(0));
  M3 : MUX2 port map (YD(1), YD(0), S(1), Y);
end STRUCTURE;
```

5. Design a synchronous circuit that implements a 3-bit modulo-6 counter (000, 001, ..., 101, 000, ...):
 - a. Fill in a transition table (use don't cares for undefined next states).
 - b. Minimize the next state (D FF) equations using Karnaugh maps.