The University of Alabama in Huntsville
ECE Department
CPE 426 01
Midterm Exam
March 11, 2010

Name: _____

1. (10 points) (a) (3 points) Create a VHDL entity named `mux_4_to_1` that represents a 4 to 1 multiplexor. (b) (7 points) Create a VHDL architecture representing a structural model of the 4 to1 mux using as many `mux_2_to_1` muxes as are needed. You do not need to write an entity or an architecture for `mux_2_to_1`. You may also assume that a component has already been declared and that no configuration statement is required.

2. (1 point) All statements inside of a block are _____.

3. (1 point) _____ is an example of a VHDL attribute.

4. (1 point) A process is triggered whenever a signal in its _____ has an event on it.

5. (1 point) _____ (True/False) Multiple assignments to a signal within a process can cause that signal to have multiple drivers.

.

6. (20 points) (a) (12 points) Write a VHDL function that takes a std_logic_vector and two integers as input. The function extracts another std_logic_vector from the first that starts at the first integer index and is the second integer characters long. You may assume that both arguments have the form std_logic_vector(0 to x'length -1) where x is the name of the argument. Output an error if the first integer is greater than x'length -1. If there are not enough characters, only output the ones that you can. Output (b)(8 points) Show an architecture that includes three calls to the function with the following properties. 1 - returns a std_logic_vector of the proper length, 2 – triggers an error message, 3 – returns a std_logic_vector that is shorter than expected.

7. (2 points) Create a `DAY_OF_WEEK` enumeration data type.

8. (1 point) _____ is an example of an unconstrained array.
9. (3 points) Write a declaration of a three-dimensional table, `TABLE_3D`, with index values and table entries all of type `std_logic` (which has been declared elsewhere and is visible). Initialize all elements of the array to 'Z';

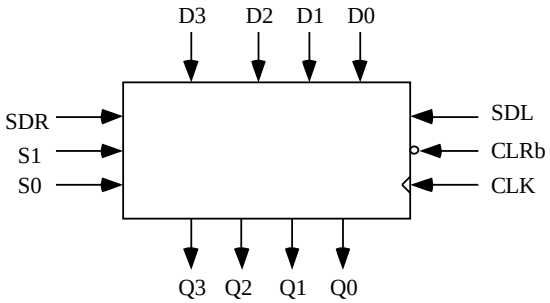
10. (20 points) Given the following VHDL, indicate all transactions and events. Give the values of A, B, C, D, E, and F each time a change occurs. Carry this out until no further change occurs.

```
entity prob is
  port (D : inout bit);
end prob;

architecture PROB of PROB is
  signal A, B, C, E, F : bit;
begin
  process
    A <= '1' after 5 ns;
    wait;
  end process;

  P1: process (A, C)
  begin
    B <= A after 3 ns;
    E <= C after 5 ns;
  end process P1;
  C <= A after 10 ns;
  P2: process (C, E)
  begin
    F <= C and E after 4 ns;
  end process P2;
  D <= A or B or C or F after 1 ns;
end PROB;
```

Time	A	B	C	D	E	F
0 ns	0	0	0	0	0	0
5 ns	1	0	0	0	0	0

Time Event Processes Triggered Scheduled Transactions Event?

11. (20 points) A description of a 74194 4-bit bi-directional shift register follows: The CLRb input is asynchronous and active low and overrides all the other control inputs. All other state changes occur following the rising edge of the clock. If the control inputs S1 = S0 = 1, the register is loaded in parallel. If S1 = 1 and S0 = 0, the register is shifted right and SDR (serial data right) is shifted into Q3. IF S1 = 0 and S0 = 1, the register is shifted left and SDL is shifted into Q0. If S1 = S0 = 0, no action occurs. (a) (4 points) Write a VHDL entity. (b) (8 points) Use concurrent signal assignments to implement the architecture. (c) (8 points) Use sequential statements to implement the architecture. Include any necessary library references.



12. (10 points) Modify the following VHDL model to use process(es) instead of blocks.

```
library ieee;
use ieee.std_logic_1164.all;

entity BUFF_REG is
 generic (STRB_DEL, EN_DEL, ODEL: TIME);
 port (DI: in std_logic_vector (1 to 8);
       DS1, NDS2, STRB : in std_logic;
       DO: out std_logic_vector (1 to 8));
end BUFF_REG;

architecture THREE_BLOCK of BUFF_REG is
  signal REG : std_logic_vector (1 to 8);
  signal ENBLD : std_logic;
begin
  PREG: block (STRB = '1')
  REG <= guarded DI after STRB_DEL;
  end PREG;

  ENABLE: block
    ENBLD <= DS1 and not NDS2 after
    EN_DEL;
  end ENABLE;

  OUTPUT: block
    DO <= REG after ODEL when ENBLD =
          '1' else "ZZZZZZZZ' after ODEL;
  end OUTPUT;
end THREE_BLOCK;
```

13. (10 points) Draw the state diagram for the following state machine. Is it a Moore machine or a Mealy machine?

```
ENTITY state_machine IS
   PORT (Clock, Resetn ; IN BIT;
         r             : IN BIT_VECTOR(1 to 3);
         g             : OUT BIT_VECTOR(1 to 3));
END state_machine;

ARCHITECTURE behavior OF state_machine IS
   TYPE state_type IS (Idle, gnt1, gnt2, gnt3);
   SIGNAL y : state_type;
BEGIN
   PROCESS (Resetn, Clock)
   BEGIN
      IF (Resetn = '0') THEN
         y <= Idle;
      ELSIF (Clock'EVENT AND Clock = '1') THEN
         CASE y IS
            WHEN Idle =>
               IF (r(1) = '1') THEN
                  y <= gnt1;
               ELSIF (r(2) = '1') THEN
                  y <= gnt2;
               ELSIF (r(3) = '1') THEN
                  y <= gnt3;
               ELSE
                  y <= Idle;
               END IF;
            WHEN gnt1 =>
               IF (r(1) = '1') THEN
                  y <= gnt1;
               ELSE
                  y <= Idle;
               END IF;
            WHEN gnt2 =>
               IF (r(2) = '1') THEN
                  y <= gnt2;
               ELSE
                  y <= Idle;
               END IF;
            WHEN gnt3 =>
               IF (r(3) = '1') THEN
                  y <= gnt3;
               ELSE
                  y <= Idle;
               END IF;
         END CASE;
      END IF;
   END PROCESS;

   g(1) <= '1' WHEN y = gnt1 ELSE '0';
   g(2) <= '1' WHEN y = gnt2 ELSE '0';
   g(3) <= '1' WHEN y = gnt3 ELSE '0';

END behavior;
```