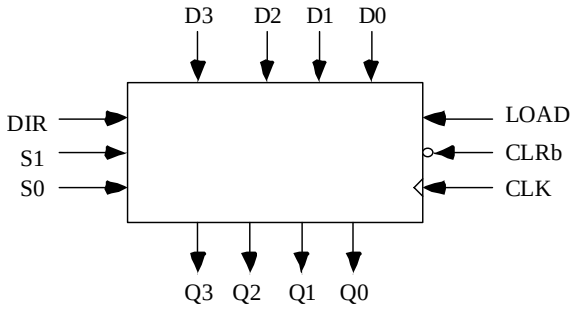
**The University of Alabama in Huntsville**
**ECE Department**
**CPE 526 01**
**Midterm Exam**
**March 2, 2006**

Name: _____

1. (15 points) A barrel shifter is a shift register in which the data can be shifted either by one bit position, as in a normal shift register, or by multiple positions. Design a four-bit barrel shifter that can shift to the right or left by 0, 1, 2, or 3 bits and has clear and parallel load capabilities. If DIR = '0', shift left, else shift right. The binary value of S1 and S0 dictate the amount of shift, for example S1S0 = 10 means shift by 2 bit positions.




(a) (4 points) Write an entity for the barrel shifter. (c) (11 points) Write an architecture for the barrel shifter

2. (1 point) A function is a primary design unit. (True/False) _____

3. (20 points).(a) (12 points) Write a VHDL function to compare two IEEE std_logic_vectors to see whether they are equal. Report an error if any bit in either vector is not ‘0’, ‘1’, or ‘-‘ (don’t care), or if the lengths of the vectors are not the same. The function call should pass only the vectors. The function should return TRUE if the vectors are equal, else FALSE. When comparing the vectors, consider that ‘0’ = ‘-‘ and ‘1’ = ‘-‘. Make no assumptions about the index range of the two vectors. (b) (8 points) Show an architecture that includes three calls to the function with the following properties. 1 - returns TRUE, 2 - returns FALSE and 3 - triggers an error message

4. (1 point) A process may have both a sensitivity list and wait statements (True/False) _____

5. (1 point) In order to specify edge behavior the _____ attribute is used in concurrent statements.

6. (1 point) A _____ binds an instantiated component to a library model

7. (8 points) Design a two-out-of-five code detector. The device receives as input a 5-bit parallel word. The detector output logic is a logic 1 for any code word that has exactly two 1's in it and is a logic 0, otherwise. (a) (2 points) Write a VHDL entity for the detector. (b) (6 points) Write a VHDL architecture for the detector.

8. (7 points) A clocked T flip-flop with synchronous CLEAR and PRESET operates in the following manner: At the falling edge of CLK, Q = ‘0’ and QB = ‘1’if CLEAR = ‘1’, Q = ‘0’ and QB = ‘1’ if PRESET = ‘1’ and Q = not Q and QB = not QB if T = ‘1’ and Q and QB remain unchanged if T = ‘0’. The delay associated with CLEAR and PRESET is TPCPQ, the delay associated with T is TPTQ. (a) (2 points) Write an entity for this flip-flop. (b) (5 points) Write an architecture for this flip-flop.

9. (5 points) Consider the following structural VHDL model.

```
entity SMODEL is
   port
     (P1 : in BIT;
      P2 : out BIT;
      P3 : inout BIT);
end SMODEL;

architecture STRUCTURE of SMODEL is
   component UNIT
      port (C1, C2, : in BIT; C3 : out BIT);
   end component;

begin
  U1 : UNIT port map (C1 => ?, C2 => ?, C3 => ?);
end STRUCTURE;
```

(a) (3 points) Complete the structural description by giving a legal set of port-to-port connections for entity ports P1, P2, and P3 and component ports C1, C2, and C3.

(b) (2 points) Is there more than one possible set of legal port-to-port connections?

10. (10 points) For the following VHDL, assume that A changes to ‘1’ at 5 ns and back to ‘0’ at 12 ns. 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
  P1: process (A, C)
  begin
    B <= A after 2 ns;
    E <= C after 7 ns;
  end process P1;
  C <= A and B after 6 ns;
  P2: process (C, E)
  begin
    F <= C and E after 4 ns;
  end process P2;
  D <= A or B or C or F;
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 |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

11. (4 points) (a) (2 points) Specify a DAY_OF_WEEK enumeration data type.

(b) (2 points) Write a variable declaration CURRENT_DAY that has a value equal to the current day of the week.

12. (10 points) Design a 2 to 4 decoder with enable. All outputs are tristated when the enable input = ‘0’. When the enable input = ‘1’, one of the four output D0, D1, D2, D3 is selected based on the binary value of the two select inputs S1 and S0. (a) (2 points) Write a VHDL entity. (b) (4 points) Use concurrent signal assignments to implement the architecture. (c) (4 points) Use sequential statements to implement the architecture. Include any necessary library references.

13. (1 points) _____ is the hardest problem.

14. (6 points) (a) (4 points) Write a declaration of a record data type, PERSONNEL, with fields for last name (LAST) (up to twenty characters); first name (FIRST) (up to twenty characters); middle initial (MID); and social security number (SOC_SEC).

(b) (2 points) Write a declaration for a constant MY_PERSONNEL_RECORD that defines your own data, consistent with type PERSONNEL.

15. (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 (sig_in ; IN BIT; clk : IN BIT;
         sig_out : OUT BIT);
END state_machine;

ARCHITECTURE state_machine OF state_machine IS
   TYPE state_type IS (a, b, c, d, e);
   SIGNAL current_state, next_state : state_type;
BEGIN
   PROCESS (sig_in, current_state)
   BEGIN
      sig_out <= '0';
      next_state <= e;
      CASE current_state
      WHEN a =>
         IF sig_in = '0' THEN
            next_state <= a;
          sig_out <= '1';
         ELSE
            next_state <= d;
         END IF;
      WHEN b =>
         IF sig_in = '0' THEN
            next_state <= b;
         ELSE
            next_state <= c;
          sig_out <= '1';
         END IF;
     WHEN c =>
       IF sig_in = '1' THEN
           sig_out <= '1';
           next_state <= a;
       ELSE
          next_state <= e;
       END IF;
     WHEN d =>
       IF sig_in = '0' THEN
           sig_out <= '1';
           next_state <= e;
       END IF;
     WHEN e =>
       IF sig_in = '1' THEN
          next_state <= c;
       END IF;
      END CASE;
   END PROCESS;
   PROCESS (clk)
   BEGIN
      IF (clk'EVENT AND clk = '1') THEN
      current_state <= next_state;
      END IF;
   END PROCESS;
END state_machine;
```