The University of Alabama in Huntsville
ECE Department
CPE 426 01
Midterm Exam
February 25, 2009

Name: _____

1. (15 points) (a) (4 points) Create a VHDL entity named `32_bit_adder`.(b) (11 points) Create a VHDL architecture representing a structural model of the 32 bit adder using as many `8_bit_adder` components as are needed. You do not need to write an entity or an architecture for `8_bit_adder`. You may also assume that a component has already been declared and that no configuration statement is required.

2. (1 point) _____ (True/False) There is no difference between CLK'event and not CLK'stable.

3. (1 point) _____ (True/False) Operators may be overloaded in VHDL.

4. (1 point) _____ delay is the delay which represents wire delay in VHDL.

5. (1 point) _____ (True/False) Functions are primary design units.

6. (20 points) (a) (12 points) Write a VHDL function that takes two std_logic_vectors. The function searches the first argument to see whether the second argument appears as a subvector of the first. If the second argument is found, the function returns the bit position of the first match, otherwise it returns -1. You may assume that both arguments have the form std_logic_vector(x'length-1 down to 0) where x is the name of the argument. Output an error if the length of the second argument exceeds that of the first. (b)(8 points) Show an architecture that includes three calls to the function with the following properties. 1 - returns a value, 2 – triggers an error message, 3 – returns -1.

7. (3 points) (a) (2 points) Specify a CLASSIFICATION enumeration data type that spells out the various classifications for undergraduate students.(b) (1 point) Write a signal declaration MY_CLASS that has a value equal to the rightmost element of the type.

8. (1 point) _____Multiple Choice: Which of the following cannot occur outside a process?
(a) Signal Assignment (b) Variable Declaration (c) Signal Declaration

9. (4 points) (a) (3 points) Write a declaration of an array that can be used to hold the email addresses of the students in this class. (b) (1 point) Initialize the first element of this array with your email address.

10. (18 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 : out 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 (D, C)
  begin
    B <= D after 2 ns;
    E <= C after 7 ns;
  end process P1;
  C <= transport A or E
         after 6 ns;
  P2: process (C, E)
  begin
    F <= (C and E) after 4 ns;
  end process P2;
  D <= A xor B xor C 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. (15 points) Design a new type of positive-edge-triggered flip-flop called the LH flip-flop. It has a clock C, a data input D, and a load input L. If, at the positive edge of C, L equals 1, then the data on D is stored in the flip-flop. If, at the positive edge of C, L equals 0, then the current stored value in the flip-flop is held. (a) (3 points) Write a VHDL entity. (b) (6 points) Use concurrent signal assignments to implement the architecture. (c) (6 points) Use sequential statements to implement the architecture. Include any necessary library references.

12. (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, rst : 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 <= c;
      CASE current_state
      WHEN a =>
         IF sig_in = '0' THEN
            next_state <= c;
            sig_out <= '1';
         ELSE
            next_state <= d;
         END IF;
      WHEN b =>
         IF sig_in = '0' THEN
            next_state <= b;
         ELSE
            next_state <= c;
         END IF;
         sig_out <= '1';
     WHEN c =>
       IF sig_in = '1' THEN
           sig_out <= '1';
           next_state <= a;
       ELSE
          next_state <= b;
       END IF;
       sig_out <= '1';
     WHEN d =>
       IF sig_in = '0' THEN
           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 (rst = '0') then
         current_state <= a;
      ELSIF (clk'EVENT AND clk = '1') THEN
         current_state <= next_state;
      END IF;
   END PROCESS;
END state_machine;
```