Final exam

ECE 303: Advanced Digital Logic Design
10 December 2003

You may not refer to your book or notes during this exam.

For all problems, show your work.

Put your name on the exam as well as the answer sheets.

Skip 20 points worth of problems by putting an **X** in the problem's "Score" column on this page.
You may not skip problems labeled **required**.

Hand in the exam as well as your answer sheet.

Problem	Score	Maximum score	Notes
1		5	
2		5	
3		5	
4		5	
5		5	
6		5	
7		5	
8		5	
9		5	required
10		10	
11		10	required
12		10	
13		10	required
14		10	
15		10	
16		15	required
Total		100	

1. **(5 pts)** Name two NP-complete problems encountered in digital logic design. For each problem, describe a practical way of completing a design in practice when the problem is encountered. Use no more than six sentences in total.

2. **(5 pts)** Give an example of a function of three variables, $f(a,b,c)$, for which the minimal POS formula contains fewer literals than the minimal SOP formula.

3. **(5 pts)** Draw a circuit within which a sub-circuit contains an observability don't-care. Indicate the sub-circuit and don't-care input value(s).

4. **(5 pts)** Indicate one advantage and disadvantage, each, for VHDL, Verilog, and SystemC. Use no more than six sentences.

5. **(5 pts)** Use at most one sentence to answer each of the following questions.

   (a) What type of PLD is most often used in systems requiring dynamic hardware configuration?

   (b) Name a company that makes such PLDs.

   (c) What are the basic building blocks for this company's PLDs called?

   (d) What is the approximate complexity of a combinational function that can be implemented in this type of building block?

   (e) Are these building blocks connected with a heterogeneous or homogeneous communication network?

   (f) Are these building blocks connected with a fixed or reconfigurable communication network?

6. **(5 pts)** Consider the following set of functions:

$$f = a\overline{b} + a\overline{c} + bd$$
$$g = a\overline{b}c + bd$$
$$h = a\overline{c} + bd$$

   Show a diagram of a PLA implementation using a minimal number of product terms. You may use any manipulations you like, as long as the final PLA implements the original functions.

7. **(5 pts)** How many half-adders and full-adders are required to build a two's-complement 14-bit ripple carry adder? You may use a small diagram or up to three sentences to justify your answer.

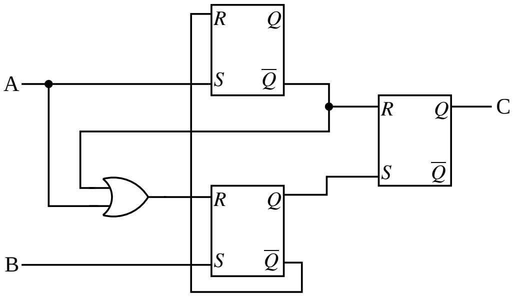
8. **(5 pts)** Name or describe (in one or two sentences) the following device. Identify *I* and *O*.

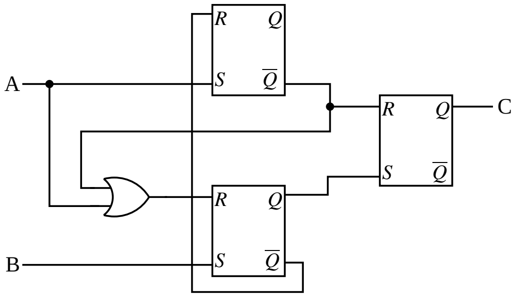
```
entity ANON_A is
port(
    I: in std_logic_vector(1 downto 0);
    O: out std_logic_vector(3 downto 0)
);
end ANON_A;

architecture behv of ANON_A is
begin
    process (I)
    begin
    case I is
        when "00" => O <= "0001";
        when "01" => O <= "0010";
        when "10" => O <= "0100";
        when "11" => O <= "1000";
        when others => O <= "XXXX";
   end case;
   end process;
end behv;
```

9. **(5 pts) required** Name or describe (in one or two sentences) the following device. Identify A,B,C,D,E, and F.

```
entity ANON_B is
port (
    A:    in std_logic;
    B, C: in std_logic;
    D:    in std_logic;
    E, F: out std_logic
);
end ANON_B;

architecture behv of ANON_B is
    signal state: std_logic;
    signal input: std_logic_vector(1 downto 0);

begin
    input <= B & C;

    p: process(A, D) is
    begin

    if (D=’1’) then
        state <= ’0’;
    elsif (rising_edge(A)) then
        case (input) is
        when "11" => state <= not state;
        when "10" => state <= ’1’;
        when "01" => state <= ’0’;
        when others => null;
        end case;
    end if;

    end process;

    E <= state;
    F <= not state;
end behv;
```

10. **(10 pts)** Name the following device as well as the A, B, and C signals. If you are unable to name the device, describe it. Use no more than six sentences. I have included two copies of the diagram so you can use it for experimentation. Note that the RS latches are each implemented with two NOR gates.





11. **(10 pts) required** Use the Quine-McCluskey method to find a minimal SOP expression for

$$f(a,b,c,d) = \sum(0,7,10,12) + d(1,2,3,5,13)$$

12. **(10 pts)** Consider the following set of cubes: 1X0X, 0X01, X1X1, X11X, and 1XX1.

(a) Use fast tautology checking to determine whether X1X1 is relatively essential.

(b) Use fast tautology checking to determine whether 1XX1 is relatively essential.

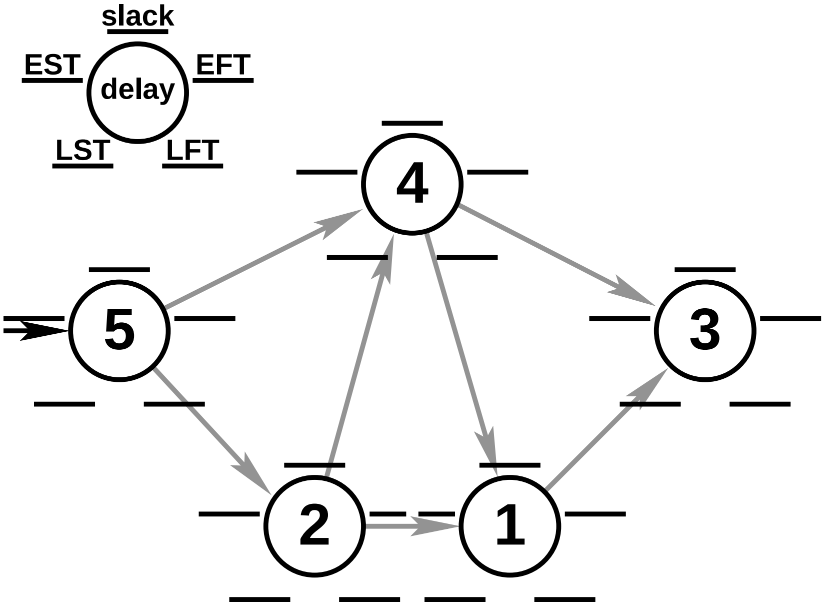
13. **(10 pts) required** Consider the following expression:

$$f(a,b,c,d) = \sum(6,10,15) + d(0,5,7,11)$$

(a) Use a Karnaugh map to derive a minimal SOP expression.

(b) Use kernel extraction to derive a multi-level implementation with a minimal number of literals. Don't forget to recursively apply kernel extraction to the kernels produced in previous steps.

14. **(10 pts)** In the following DAG, each node represents a logic gate and is labeled with the gate's delay.



(a) Label each node's EST, LST, EFT, LFT, and slack, as indicated by the key node. You may assume that the final node's LFT is equal to its EFT. You may find that thinking about the graph allows you to get an answer more quickly than blindly applying the methods given in class.

(b) Circle the nodes which may have their speeds decreased without changing the delay of the circuit.

15. **(10 pts)** Design an asynchronous finite state machine with two-inputs, L and M, and one output, N. L acts as a reset. If L is high, N should be low. N should be high if and only if, after L went low, M was high at some point in time and low at some point in time.

(a) Show the state diagram.

(b) Do AFSM state assignment.

(c) Use three or fewer sentences to describe any special techniques and considerations to use when implementing state variable functions.

16. **(15 pts) required** Design a synchronous finite state machine with one input, I, and two outputs, L and M. L should be high if and only if the machine has observed two or more zeros on I. M should be high if and only if the machine has observed exactly one zero on I, combined with any number of ones. You may use regular expressions and an NFA to help you derive a Moore machine.

   (a) Show the state diagram.

   (b) Is a binate covering formulation necessary to do state minimization for this machine? Why or why not? Use three or fewer sentences.

   (c) Do state minimization.

   (d) State the prioritized rules for heuristic state assignment.

   (e) Do heuristic state assignment.

   (f) Derive minimal and correct state variable and output equations.

   (g) Draw a diagram of your circuit.