

Final exam

ECE 303: Advanced Digital Logic Design
Spring 2003

9 June

Permitted time: two hours

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

You must do problem one. Do only seven of the remaining eight problems.

1. Short questions

- In four or fewer sentences, explain the advantages of two-level representations for hazard-free implementation. Consider both static and dynamic hazards.
- Define the term “Satisfiability Don’t-Care”.
- In two sentences or fewer, describe the difference between PLAs and PALs.
- When is the topological sort algorithm used?
- Name three problems that can be solved by a binate covering algorithm.

2. Combinational minimization

- Use a Karnaugh map to simplify the following function. Your answer should be in *POS* form.

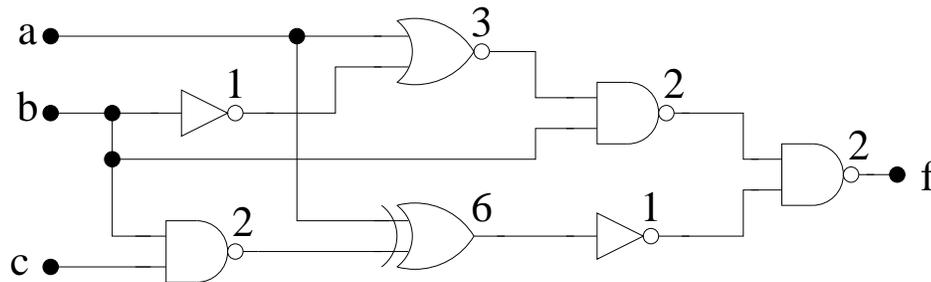
$$f(a,b,c,d) = \sum m(1,4,5,9,12) + d(0,2,3,6,14)$$

- Do SOP Quine-McCluskey minimization of the following function

$$f(a,b,c,d) = \sum m(0,1,4,7,9,10) + d(3,5,11)$$

3. Timing analysis and optimization

In the following circuit diagram, the delay of each gate is given. The output has a deadline of 18 time units.



- Find the EST, LST, and slack at each gate.

- (b) State which gates are on the critical path and which are not.
- (c) Assuming access to complemented literals, design a minimal-delay circuit implementing the same function. Use only those gate types in the diagram.

4. Sequential elements

- (a) Use an rising edge triggered D flip-flop and a minimal number of additional logic gates to build a rising edge triggered T flip-flop. Draw a schematic for your design. You need not show the gate-level D flip-flop.
- (b) Draw the gate-level schematic for a gated (has an enable input) active-high RS latch. You may use only NAND and NOR gates and should use as few gates as possible.
- (c) Draw the gate-level schematic for a master-slave D flip-flop for which the output changes on a low to high clock transition. You can use a gate-level RS latch as a component. If you understand master-slave JK flip-flops, designing a master-slave D flip-flop is straight-forward.

5. Synchronous finite state machine design

Design a one-input (I), two-output (L and M) synchronous Moore FSM. L should be 1 if and only if the most recent two inputs were 01. M should be 1 if and only if L was 1 at some time in the past and the most recent input was 0.

- (a) Draw the state diagram. You may use intermediate representations if this helps you.
- (b) Give the state table.
- (c) Write the three guidelines to use during state assignment to reduce implementation complexity.
- (d) Use these guidelines to do state assignment for the FSM.
- (e) Draw a block diagram for the FSM. You need not show your circuit at the gate level. However, you do need to use the correct number of inputs, outputs, and state variables.

6. State minimization

Do state minimization of the following Moore machine. You may use n -equivalence partitions or an implication chart.

current state	next state		output
	$i=0$	$i=1$	
A	E	B	0
B	A	C	1
C	D	C	1
D	E	D	0
E	A	F	0
F	E	C	1

7. AFSM design

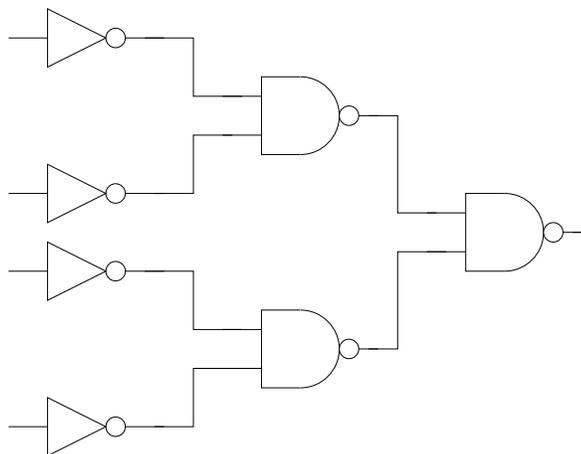
You may assume that only one input changes at a time.

- (a) Design a two-input (a and b), one output (c) asynchronous Moore finite-state machine that has an output of 0 when $a = 0$ and $b = 1$, an output of 1 when $a = 1$ and $b = 0$. The machine should maintain its previous output when both inputs are 0. You need not consider a and b simultaneously being 1.
- Draw the state diagram for the machine.
 - Do state assignment.
 - Write the state table.
 - Find minimized state variable and output functions.
 - Draw a gate-level diagram of the circuit.
 - Name this AFSM.
- (b) Design a four-state (S_1, S_2, S_3, S_4) asynchronous FSM. The machine has two inputs, c (clear) and a (advance). The machine starts in S_1 . If a is 0, the machine remains in S_1 . If a is 1, it advances to, and remains in S_2 . Each time a changes, the machine advances, then remains in, the next state. The machine advances from S_4 to S_1 . If, at any time, c is 1, the machine immediately transitions to, and remains in S_1 . The machine has the following two-bit outputs, depending only on the current state: $S_1 = 00, S_2 = 01, S_3 = 11, S_4 = 10$.
- Is this a Mealy or Moore machine?
 - Draw the state diagram for the machine.
 - Do state assignment.
 - Write the state table, giving state names and assigned values.

8. Technology mapping

Given the following technology library, find a minimal-area technology mapping for the given decomposed circuit. You may use either of the approaches shown in class. However, you need to show the steps you follow.

Gate	Area
NAND2	2
OR2	3
OAI4	9



9. VHDL

- (a) Write descriptive names for *dsgn*, *a*, *b*, *c*, and *d*.

```
ENTITY dsgn IS
    PORT(a, b, c: in bit;
         d: out bit
        );
END dsgn;

ARCHITECTURE behavioral OF dsgn IS
BEGIN
    WITH c SELECT
        d <= a WHEN '0',
        b WHEN OTHERS;
END behavioral;
```

- (b) Write descriptive names for *dsgn*, *a*, *b*, *c*, *d*, and *e*.

```
ENTITY dsgn IS
    PORT(a, b, c, d: in bit;
         e: out bit
        );
END dsgn;

ARCHITECTURE arch1 OF dsgn IS
BEGIN
    PROCESS (a, b, c)
        VARIABLE temp: bit;
    BEGIN
        IF (a = '1') THEN
            temp := '1';
        ELSIF (b = '1') THEN
            temp := '0';
        ELSIF (c'event and c = '1') THEN
            IF (d = '1') THEN
                temp := NOT(temp);
            END IF;
        END IF;

        e <= temp;
    END PROCESS;
END;
```