Name: _____

Lab
Section**: Circle One** **M**(A01) **T**(A02) **W**(A03) **R**(A04)

UNIVERSITY OF CALIFORNIA, DAVIS
Department of Electrical and Computer Engineering

EEC 180A **Digital Systems I** **Spring 2003**

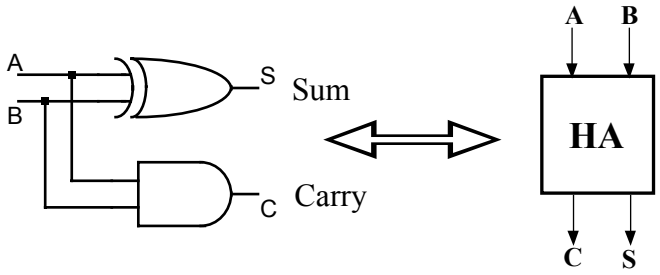
FINAL EXAM

Open-Book, Open-Notes

Number	Topic	Weight	Score
1	Ripple Adder from Half-Adders	15	_____
2	Hazards in 2-Level NAND Realizations	15	_____
3	Essential Prime Implicants	10	_____
4	Minimum Equivalent States	20	_____
5	RS Flip-Flops & ROM-Based Design	25	_____
6	Partitioning State Machine for Implementation	15	_____

Total _____

1) (15 points) Realize a four-bit ripple adder using only half-adder modules, depicted below, and possibly some inverters. The operands are denoted as (A_3, A_2, A_1, A_0) and (B_3, B_2, B_1, B_0) where A_0 and B_0 are the least significant bits. There is no carry-in bit to the ripple adder. Note that in terms of usual propagate and generate terms, **S=P** and **C=G**. Remember for a full adder, $C_{out} = G + P \cdot C_{in}$ where C_{in} is the carry in to a FA and C_{out} is the respective carry out.



a) (12 points) Draw a schematic for the ripple-adder using only HA modules and possibly inverters.

Problem 1 (continued)

b) (3 points) Assuming the delay through both parts of half-adder is 5 ns. and the delay through an inverter is 2 ns., indicate the maximum delay **path** through your implementation. What is the total maximum delay in nanoseconds?

2) (15 points) A five-variable K-map was used to design a realization of the function g(A,B,C,D,E) using two levels of NAND gates employing the prime implicants $\overline{A}\,\overline{B}\,\overline{D}\,\overline{E}$; $\overline{A}\,B\,E$; $B\,\overline{D}\,E$; $A\,\overline{B}\,\overline{C}\,\overline{D}$.

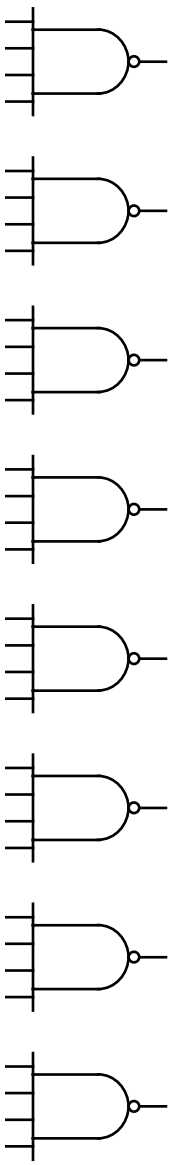
	BC	**00**	**01**	**11**	**10**
DE					
A=0	**00**	0 **1**	4 **1**	12	8
	01	1	5	13 **1**	9 **1**
	11	3	7	15 **1**	11 **1**
	10	2	6	14	10
A=1	**00**	16 **1**	20	28	24
	01	17 **1**	21	29 **1**	25 **1**
	11	19	23	31	27
	10	18	22	30	26

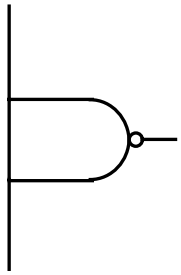
g(A,B,C,D,E)

a) (5 points) What kind of static hazards can be present? Explain.

Problem 2 (continued)

b) (10 points) Redesign the 2-level NAND gate realization to eliminate all static hazards using a minimum number of gates. The output NAND gate can have as many inputs as needed, but the first level gates are restricted to 4 inputs each. Use the gate array below.





g(A,B,C,D,E)

3) (10 points) A prime implicant chart given on the next page contains the listing of all prime implicants and minterms associated with a function f(U,V,W,X,Y,Z). However, the entries in the rows and columns have not been completed yet. (The format considers Z as least for decimal minterm listing purposes, the usual convention.)

a) (3 points) Complete the construction of the prime implicant chart after describing a typical entry.

b) (3 points) Determine all essential prime implicants using the chart, fully explaining your criteria for their selection.

PRIME IMPLICANTS:

c) (4 points) Give a minimum realization of the function expressed in a sum-of-products form.

Minterm Numbers

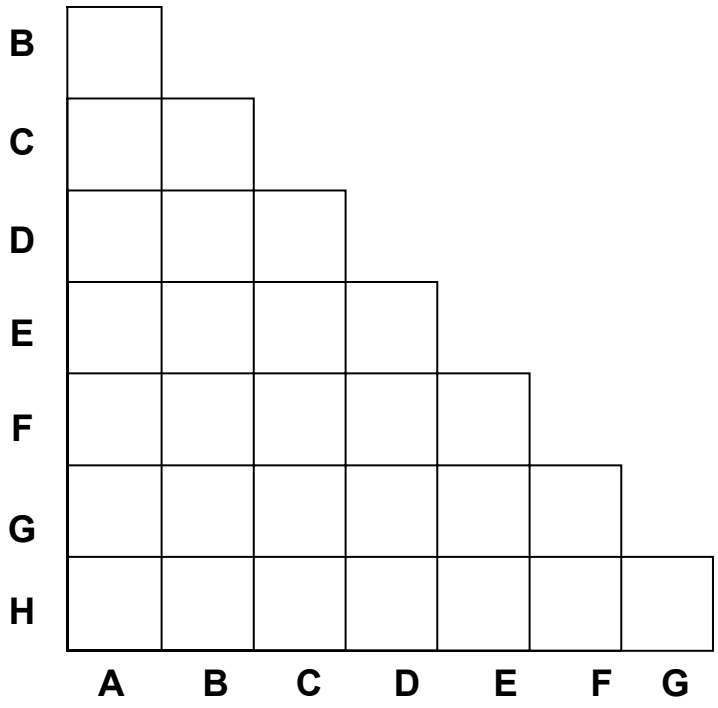
Prime Implicants	1	5	7	13	14	15	17	21	22	23	24	29	30	31	32	39	40	41	44	45	56	57	60	61
0--1-1 (5, 7, 13, 15, 21, 23, 29, 31)																								
10-000 (32, 40)																								
01-11- (22, 23, 30, 31)																								
1-1-0- (40, 41, 44, 45, 56, 57, 60, 61)																								
-00111 (7, 39)																								
-11000 (24, 56)																								
0-111- (14, 15, 30, 31)																								
0-0-01 (1, 5, 17, 21)																								

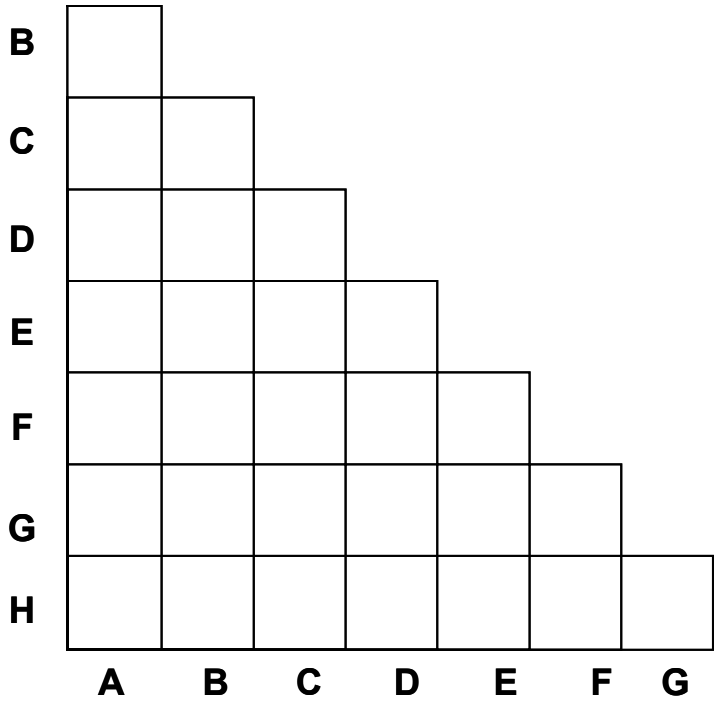
4) (20 points) Using implication charts (you may use the ones on next page), determine the minimum state realization for the machine described in the state table below. It has states A through H and input P and output T.
Give the reduced state table for the equivalent states in the table at end of problem.

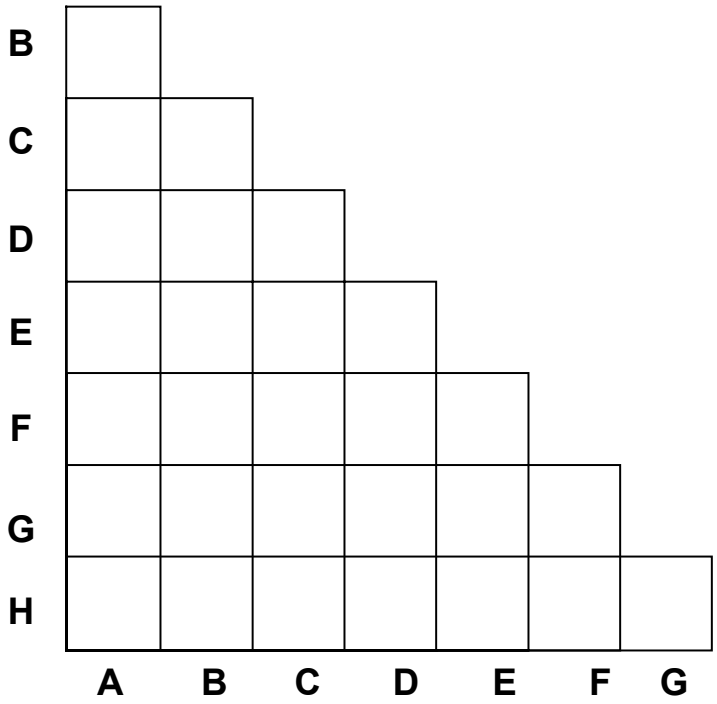
	NEXT STATE		OUTPUT T	
PRESENT STATE	P = 0	P = 1	P = 0	P = 1
A	D	G	1	0
B	C	H	1	0
C	H	B	1	1
D	C	G	1	0
E	G	D	1	1
F	A	B	1	1
G	B	G	1	0
H	D	A	1	0

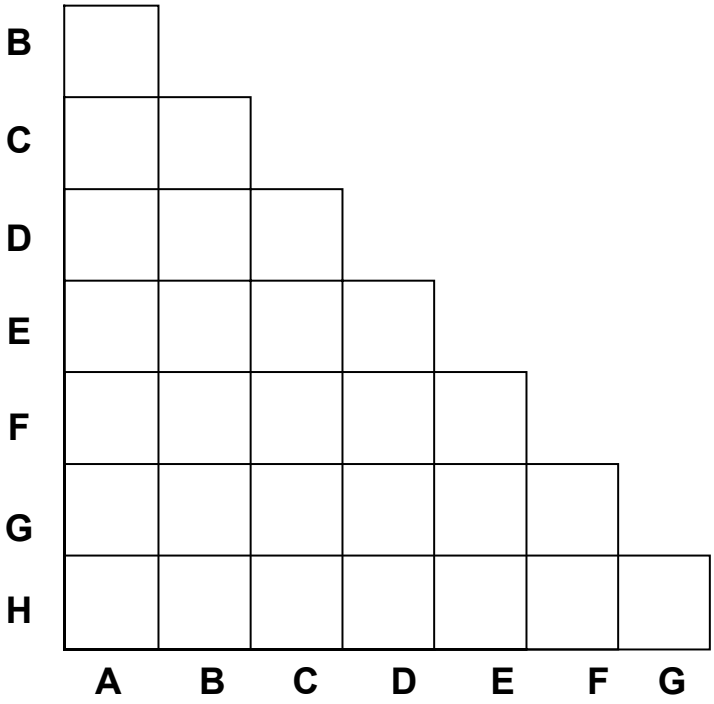
ORIGINAL STATE TABLE

Extra pages for Implication Charts






B
C
D
E
F
G
H
A B C D E F G

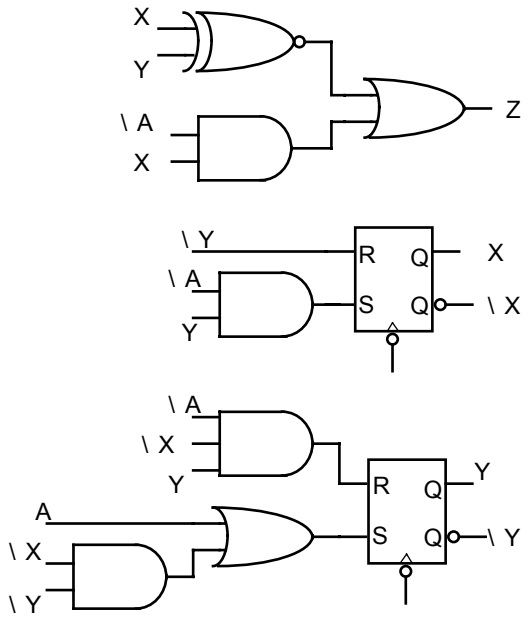

B
C
D
E
F
G
H
A B C D E F G

4. continued
List the equivalent states and show their behavior on the blank table below using only a necessary number of rows

	NEXT STATE		OUTPUT T	
PRESENT STATE	P = 0	P = 1	P = 0	P = 1

EQUIVALENT STATE TABLE

5) (25 points) A finite-state machine with state variables X and Y, input A and output Z is realized with RS flip-flops and standard gates as shown below, where the \ symbol denotes the complement of the Boolean variable as usual.



a) (5 points) Determine the next-state equations and output equation for this machine.

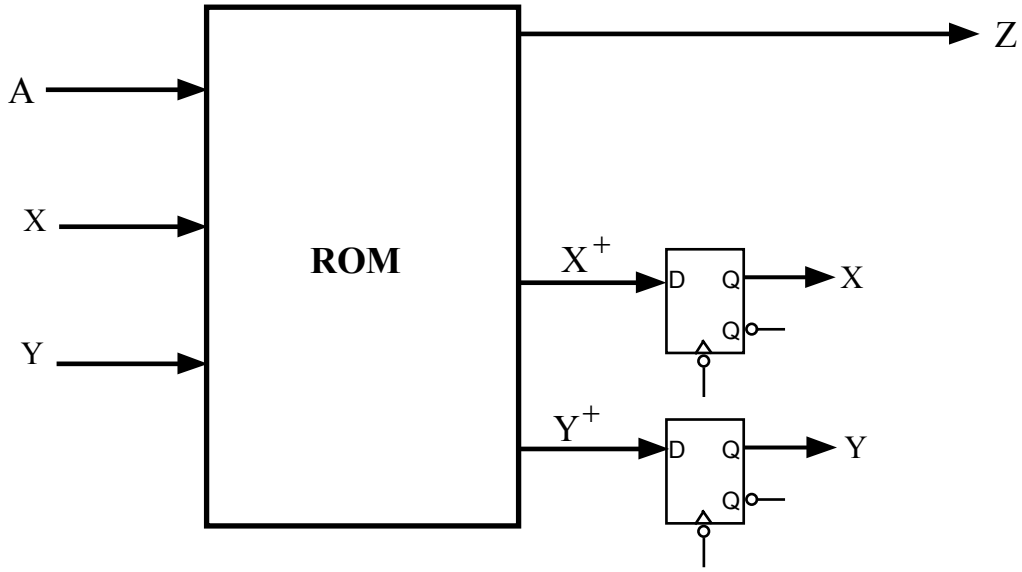
$X^+ =$

$Y^+ =$

$Z =$

5) (continued)

b) (10 points) The same machine is to be implemented with a ROM and two D flip-flops as shown below. Design the machine. Be specific about the size of the ROM, its addressing bits and output bits. Explicitly give the contents of the ROM in a table you complete below.



Address (give labels)	**ROM Contents** (give labels)

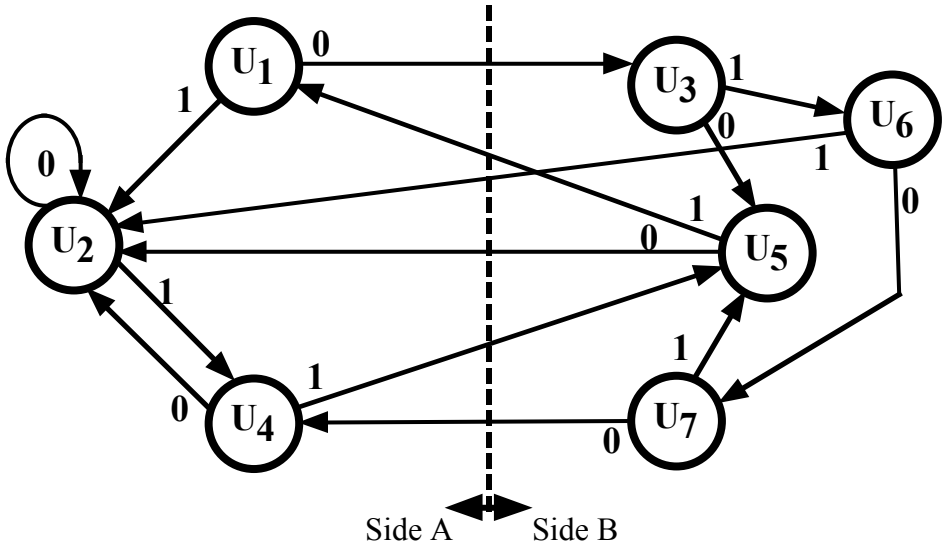
5) (continued)

c) (10 points) Assume the ROM in part b) has access time T_{acc} = 35 ns but cycle time T_{cyc} = 60 ns. The flip-flops require set-up time T_{su} = 10 ns and hold time T_h = 7 ns, and have propagation delays of T_{pHL} = 12 ns and T_{pLH} = 14 ns.

i) Give the general formula for determining the **minimum** clock period T_{min} for which such a state machine configuration will function properly. Use the symbols defined above.

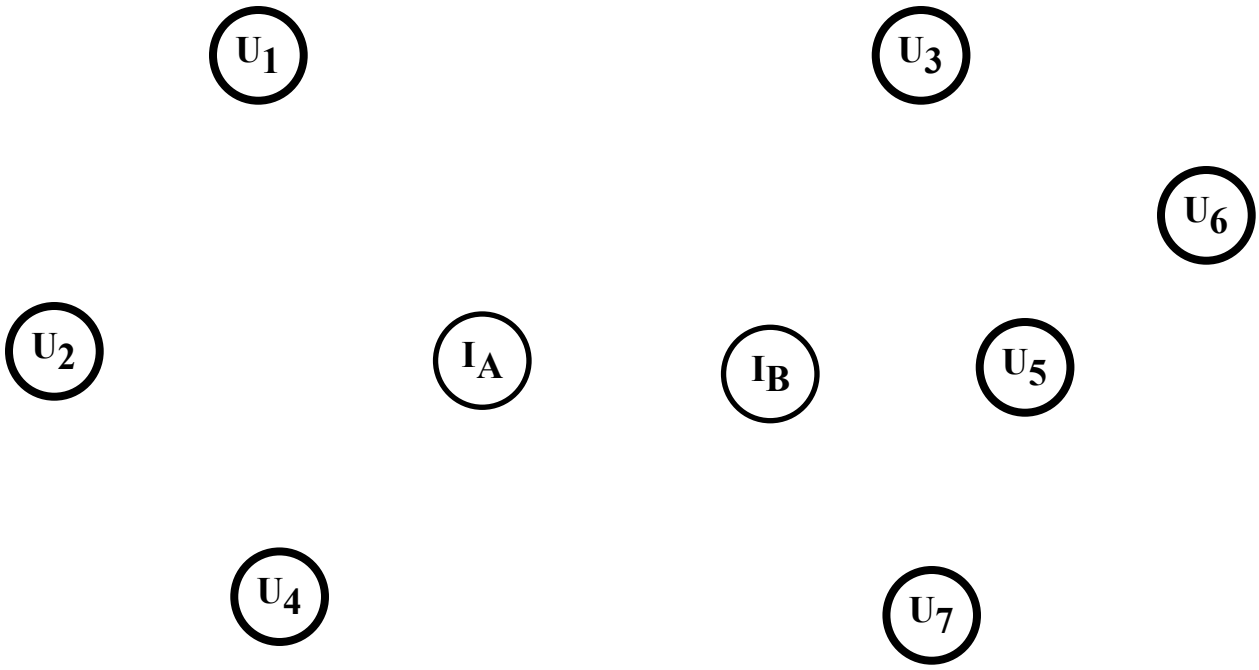
ii) For the parameters given, determine the **maximum** clock frequency for which the machine will work correctly.

6) (15 points) A state machine with input Y (a Moore type machine with the output values inside each state node omitted for simplicity) is shown below. It is to be partitioned into two machines across the dotted boundary indicated. The A side will contain idle state I_A while side B involves idle state I_B. Complete the separated state diagram begun on the next page. Label all arcs employing state occupancy variables, e.g., Boolean variable U_i denotes that the machine is currently in state U_i.



Original State Diagram (outputs omitted)

6) (continued)



Partitioned State Machine