CARLETON UNIVERSITY

333Final
EXAMINATION
April, 16, 2008; 9:00 hrs

Name:
Number:
Signature:

DURATION: **3** HOURS   No. of Students **218**

Department Name & Course Number: **Electronics ELEC 2607 A and B**

Course Instructor(s) T.G.Ray and J. Knight

AUTHORIZED MEMORANDA TURN OFF cell phones and personal communications equipment and LEAVE THEM AT THE FRONT. Notes, books, and **non-communicating** calculator are allowed.

Students MUST count the number of pages in this examination question paper before beginning to write, and report any discrepancy immediately to a proctor. This question paper has 10 pages.

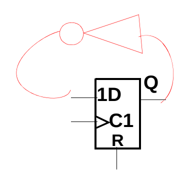
This examination pape May Not be taken from the examination room.

In addition to this question paper, students require: an examination booklet yes ☐ no ☒ *may request*
a Scantron sheet yes ☐ no ☒

Please answer on the examination paper. If your answer does not fit check you are not doing a long hard way. You may ask for a booklet if you need it.

1 Boolean

For ALL questions: If you use a map, indicate for which function it is.
If you use algebra, indicate the rules used at the right side of each line.



2% a) Using a D flip-flop and gates, produce a circuit which toggles from "0" to "1" on one active clock edge, and back to "0" again on the next active clock edge. Show only your final answer.

AB \ C	0	1
00		
01		
11		
10		

map of ___

2% b) Factor g = A+BCD into a Product of Sums (Π of Σ).

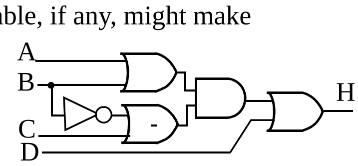
g = (A + B)(A + CD) {D2}
g = (A + B)(A + C)(A + D) {D2}

Alternately, looping zeros
$\overline{g} = \overline{A}\cdot\overline{B} + \overline{A}\cdot\overline{C} + \overline{A}\cdot\overline{D}$
then use DeMorgan
g = (A + B)(A + C)(A + D)

AB \ CD	00	01	11	10
00				
01			1	
11	1	1	1	1
10	1	1	1	1

map of _g_
with $\overline{g}$ looped

2% c) Given a circuit represented by H = (B+A)($\overline{B}$+C) +D
i) Changing the value of which input variable, if any, might make a glitch appear in H?__*B*_
ii) What must the values of the other variables be to get the glitch?
A, C, D are all 0.



2% d) Find the dual of K = AD + ($\overline{A}$ + B)($\overline{B}$ + $\overline{C}$D)

K = [*AD*] + [($\overline{A}$ + *B*)($\overline{B}$ + {$\overline{C}$*D*})]
K_{dual} = [A+D][($\overline{A}$·*B*)+($\overline{B}$·{$\overline{C}$ + *D*})]
= (*A + D*)[$\overline{A}$·*B* + $\overline{B}$·($\overline{C}$ + *D*)]

Alternately using Swap
K= AD + ($\overline{A}$ + *B*)($\overline{B}$ + $\overline{C}$*D*)
= *AD* + $\overline{A}\cdot\overline{B}$ + *B*$\overline{C}$*D*
K_{dual} =
(*A+D*)($\overline{A}$+$\overline{B}$)(*B*+$\overline{C}$+*D*)

2% e) i) Loop the map shown, with no extra loops, to give the minimum gate equation for L. Each extra loop halves your mark.
ii) Write the equation.
L = $\overline{D}$

AB \ CD	00	01	11	10
00	d	d	d	d
01	1	d	0	d
11	d	d	d	1
10	1	d	d	1

Final map of _L_

AB \ CD	00	01	11	10
00	d	d	d	d
01	1	d	0	d
11	d	d	d	1
10	1	d	d	1

Rough map of _L_

2% f) i) Draw the Karnaugh map for a circuit with output M, and inputs A, B and C:
M = A if C is 1; M = B if C is 0.
ii) Write the equation.
M = AC + B$\overline{C}$

Alternate K map accepted if equation was right

AB \ C	0	1
00	B	A
01	B	A
11	B	A
10	B	A

AB \ C	0	1
00		
01	1	
11	1	1
10		1

map of ___

page 1	2	3	4	5	6	7	8	9	10	

2 Boolean ____

3%

a) Given $q = \overline{(\bar c\cdot\bar b + \bar a)} + a + c\cdot b + e\cdot a$ find $\bar q$ without using any long overbars. A three letter answer will be rewarded.

$\overline{(\bar c\cdot\bar b + \bar a)} = (\overline{[\bar c\cdot\bar b] + \bar a}) = ([c+b]\cdot a)$ Gen DeMorgan

$q = ([c+b]\cdot a) + a + c\cdot b + e\cdot a$ Subst into q

$q = a + c\cdot b$ Use Simp twice

$\bar q = \bar a\cdot(\bar c + \bar b)$

$\bar q = (\bar c\cdot\bar b + \bar a)\cdot a\cdot(\bar c + \bar b).(\bar e + \bar a)$ desperation ans

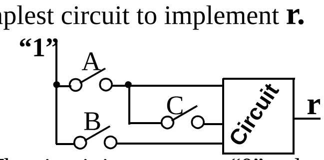
CD → AB ↓	00	01	11	10
00				
01				
11				
10				

map of ___

4%

b) The inputs to a circuit are connected as shown so they have the restriction that C will never be 1 when A is 0.
Insert don't cares on the map to satisfy these restrictions.
Given this restriction, find the simplest circuit to implement **r.**

$r = \bar A\cdot\bar B\cdot\bar C + AB + A\cdot\bar B C$



The circuit inputs go to "0" unless connected to "1".

On the map find where A=0 and C=1. These inputs can never happen, so one can let the outputs be anything. Then from the map

$r = \bar A\cdot\bar B + AB + C$

C → AB ↓	0	1
00	1	d
01		d
11	1	1
10		1

map of _r__

3%

c) Simplify $s = X\bar Y$ where

$X = \bar A\cdot B + \bar B\cdot C + \bar C\cdot A$

$Y = \bar A\cdot C + \bar B\cdot A + \bar C\cdot B$

From the maps, Y=X
Therefore
$s = XY = X\bar X = 0$

If you love algebra try:
$\bar Y = \bar A\cdot\bar B\cdot\bar C + ABC$ (loop zeros on Y map)
$XY = (\bar A\cdot\bar B\cdot\bar C + ABC)(\bar A\cdot B + \bar B\cdot C + \bar C\cdot A)$
$= (0+0+0) + (0+0+0) + (0+0+0)$
XY =0

C → AB ↓	0	1
00		1
01	1	1
11	1	
10	1	1

map of X

C → AB ↓	0	1
00	0	1
01	1	1
11	1	0
10	1	1

map of Y

3%

d) (i) Factor **u** into a minimal **Product-of-Sums** expression.
(ii) Check you did not find the ~~sum of products.~~

$u = \bar A\cdot\bar B + B\cdot\bar C + AB + A\cdot\bar C$

$\bar u = \bar A BC + A\bar B C$ from maps

$u = (A + \bar B + \bar C)(\bar A + B + \bar C)$

CD → AB ↓	00	01	11	10
00	1	1	1	1
01	1	1	0	0
11	1	1	1	1
10	1	1	0	0

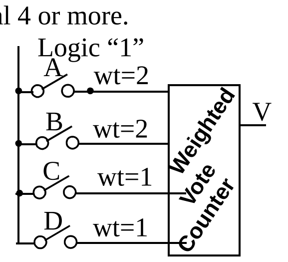
map of _u_

Zeros show $\bar u$

4%

e) Four people, A, B, C and D, need a machine to tally their votes. A and B's votes each have a weight of 2; C and D's votes each have a weight of 1. Design the logic equation for a vote tallying circuit which gives V=1 if the weighted votes total 4 or more.

from maps
$V = AB + CD(A + B)$



The circuit inputs go to "0" unless connected to "1".

CD → AB ↓	00	01	11	10
00		1	2	1
01	2	3	4	3
11	4	5	6	5
10	2	3	4	3

map of _weights of votes

CD → AB ↓	00	01	11	10
00				
01			1	
11	1	1	1	1
10			1	

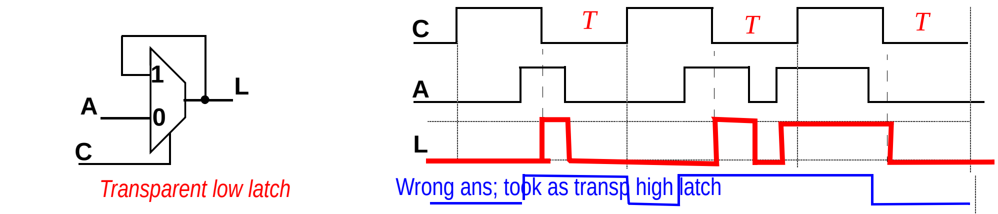
map of _V__

3 Machines with Storage

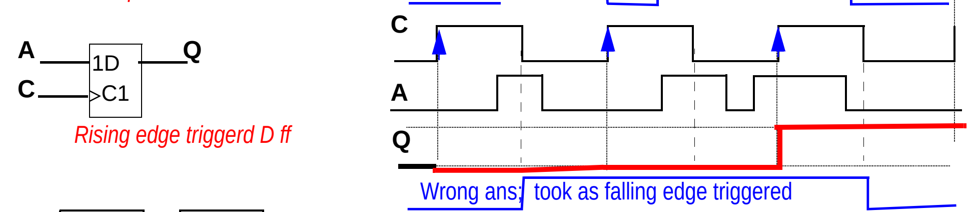
6%

a) Sketch the output waveform(s) for the following:

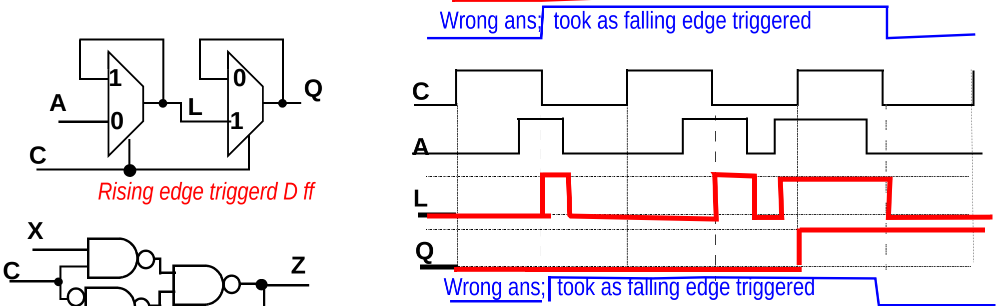
i)



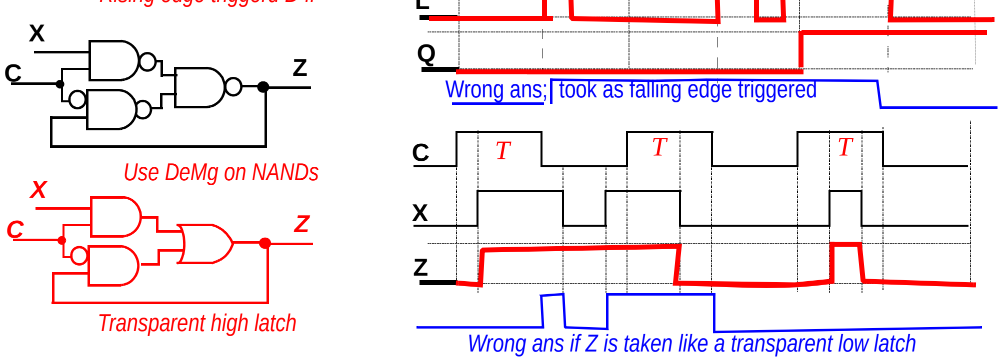
ii)



iii)

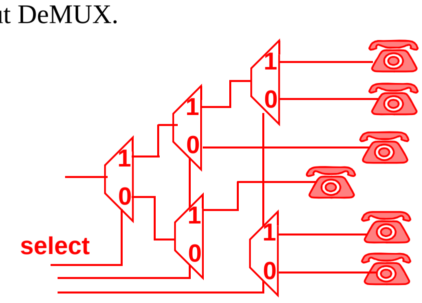


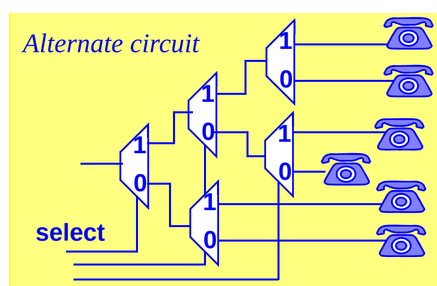
iv)



4%

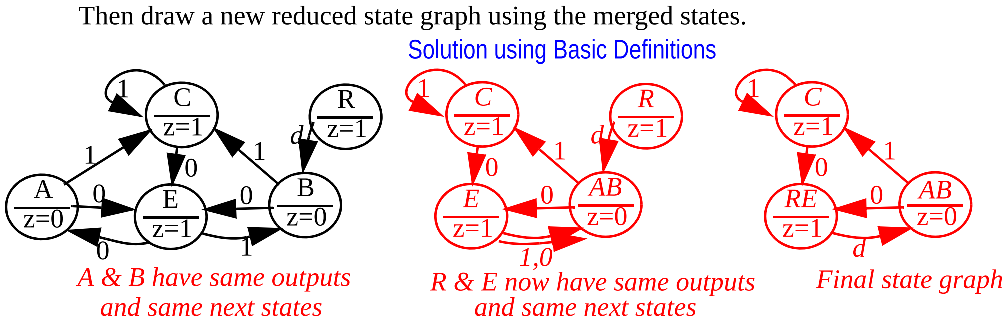
b) Using several two-output DeMUXs, and nothing else, except possibly NOT gates, design a six-output DeMUX.

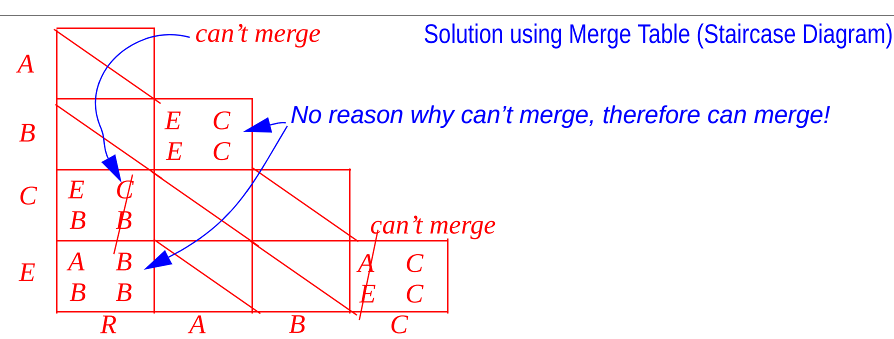




3%

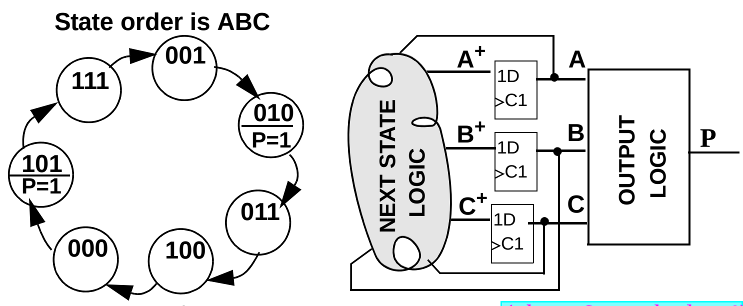
c) If any of the states below can be merged, merge them.
Then draw a new reduced state graph using the merged states.

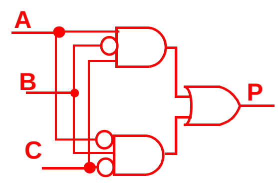




3%

d) The logic in the circuit below makes it count according to the state graph shown. The output P is 1 when the machine is in state ABC=010 or in state 101. Otherwise P is 0. Design the **output** logic.





Hint: Think before you dive into state tables. *(about 6 people dove!)*

$P = A\overline{B}C + \overline{A}B\overline{C}$

AB \ C	0	1
00		
01	1	
11	d	
10		1

map of P

State ABC=110 does not appear

$P = A\overline{B}C + B\overline{C}$

Using the don't care state
Bonus mark

Two clever students pointed out that

$P = (A{=}{=}C)(A \neq B) \Rightarrow \overline{(A \oplus C)}(A \oplus B)$

4%

4 Product-of-Sums Map

(a) Loop the map to find the minimum **Product-of-Sums** logic.
(b) Write the equation.
(c) Check you did not find the ~~sum of products.~~

AB \ CD	00	01	11	10
00	1	1	0	1
01	0	0	1	1
11	*d*	1	1	0
10	0	1	0	1

map of F

AB \ CD	00	01	11	10
00			*1*	
01	*1*	*1*		
11	*d*			*1*
10	*1*		*1*	

map of $\overline{F}$

The d can be made 1 or 0 in $\overline{F}$, just as it can in F.

$\overline{F} = \overline{C}\,\overline{A}B + D\overline{B}C + AB\overline{D} + \overline{C}A\overline{D}$ *(DeMorg)*

$F = (C+A+\overline{B})(\overline{D}+B+\overline{C})(A+B+\overline{D})(\overline{C}+A+\overline{D})$

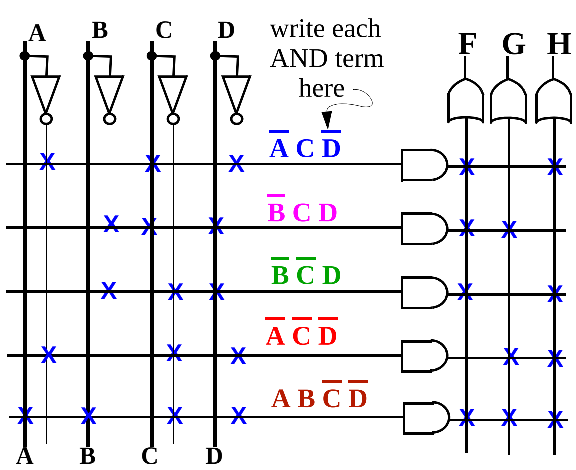
Sum of products, not the answer asked for:

$F = \overline{A}\cdot\overline{B}\cdot\overline{C} + \overline{B}C\overline{D} + \overline{A}BC + A\overline{C}D + ABD$

4%

5 Programmable Logic

Program the PLA to implement the logic defined by the K-maps shown. The inverted lines are made thinner to help avoid using the wrong line.



(This is the only size of PLA available to you.)

Map of F

AB \ CD	00	01	11	10
00		1	1	1
01				1
11	1			
10		1	1	

Map of G

AB \ CD	00	01	11	10
00	1		1	
01	1			
11	1			
10			1	

Map of H

AB \ CD	00	01	11	10
00	1	1		1
01	1			1
11	1			
10		1		

If the AND term were written, a wrong X counted as a misprint mistake.
If there were no AND term written, a wrong X counted as an equation error

6 State Reduction

7%

(i) Find the equivalent states in the table below.
(ii) Make a new state table with the minimum number of states.

State	Next State X=0	X=1	Output Z X=0	X=1
A	G	B	0	0
B	E	D	0	0
C	G	A	0	0
D	E	C	0	0
E	H	D	0	0
G	H	A	0	0
H	E	J	1	0
J	A	E	1	1

. Revised State Table

State	Next State X=0	X=1	Output X=0	X=1
ABCD	*GE*	*ABCD*	*0*	*0*
GE	*H*	*ABCD*	*0*	*0*
H	*GE*	*J*	*1*	*0*
J	*ABCD*	*GE*	*1*	*1*

B	*E D* *G B*						
C	*G A* *G B*	G A E D					
D	*E C* *G B*	*E C* *E D*	*E C* *G A*				
E	*H D* *G B*	H D E D	H D G A	H D E C			
G	*H A* *G B*	H A E D	H A G A	H A E C	H A H D		
H							
J							
	A	*B*	*C*	*D*	*E*	*G*	*H*

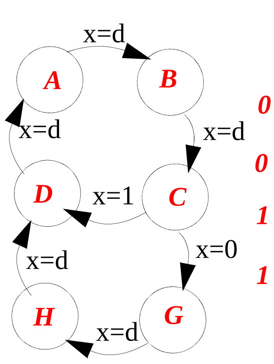
4-WAY MERGE

2-WAY MERGE

OUTPUTS DIFFERENT

7 Asynchronous State Assignment

3%



	0	1
0 0	*A*	*B*
0 1	*D*	*C*
1 1	*H*	*G*
1 0		

The **asynchronous** state diagram on the left shows all of the transitions for a particular asynchronous machine.
(i) Do a good state assignment for this machine and put the states inside the circles on diagram.
(ii) Check that you did **not** do an assignment only good for a synchronous machine.

One must have only one state bit change at a time during any allowed transition..
Put the states in K-map order.

8 Synchronous State Graph

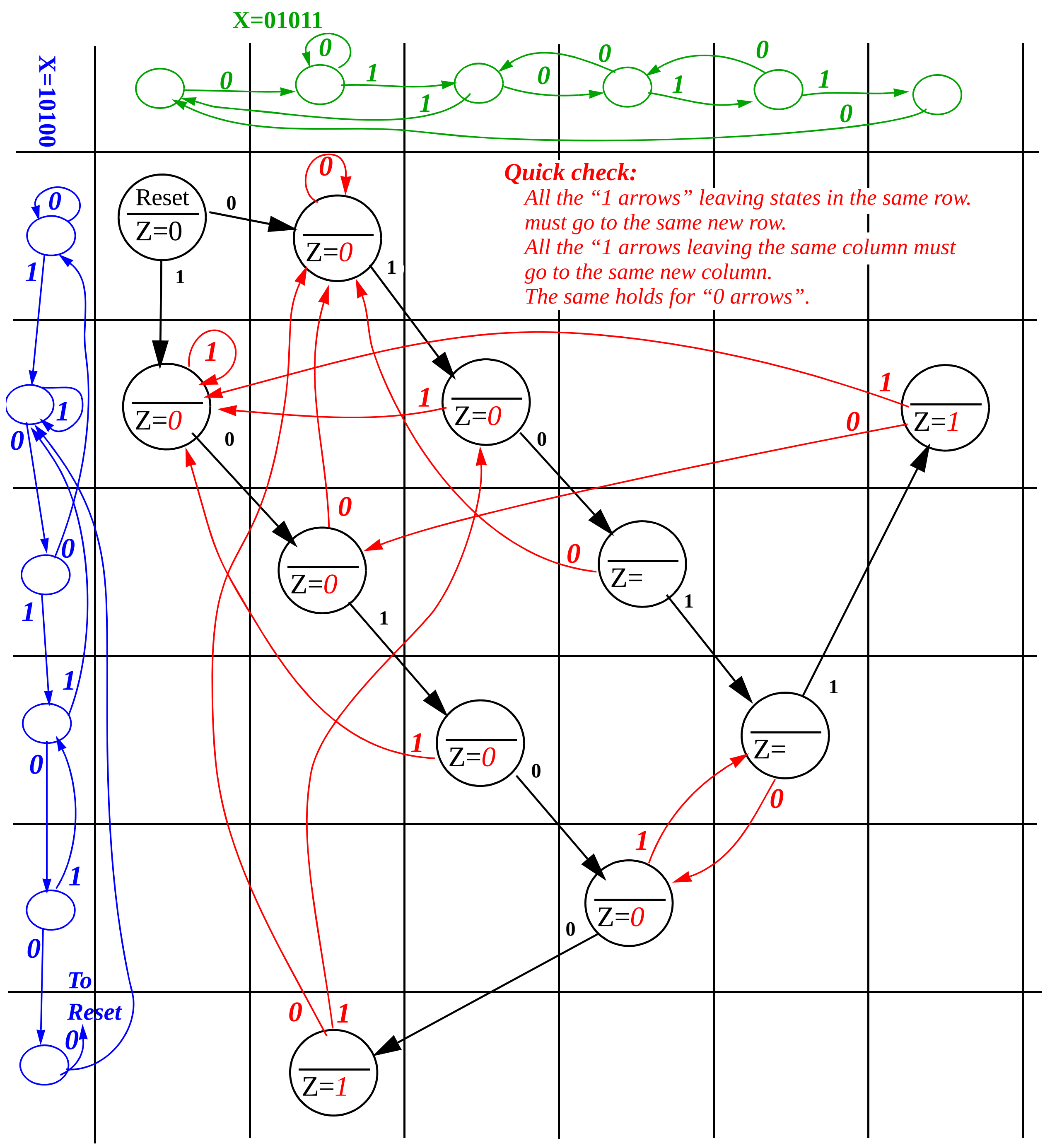
9%

Draw the state graph for a machine to meet all the following requirements:

- Has one input X, and one output Z.
- Z=1 during the cycle **AFTER** the machine receives the complete sequence **X=10100**.
- Z=1 during the cycle **AFTER** the machine receives the complete sequence **X=01011**.
- Otherwise Z is 0.
- The leftmost bit of X is received first.
- Overlapped sequences are to be detected, including a sequence overlapping itself.
- The machine starts at the RESET state.

Draw only the state graph.

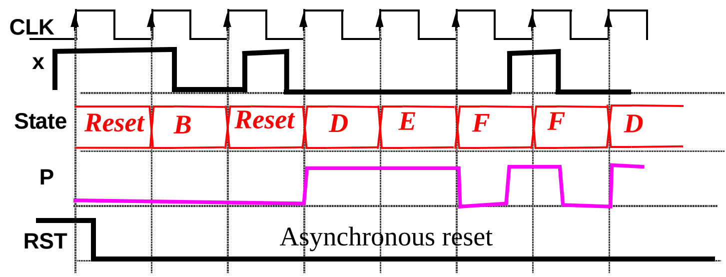
Use ONLY the states provided below and **do NOT change anything** already written.

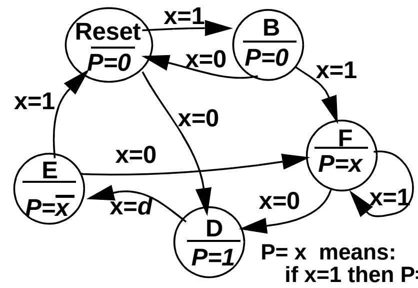


9 Synchronous Waveforms

5%

Fill in the state letter and the value of the output P on the timing diagram.





10 Asynchronous Circuits

6%

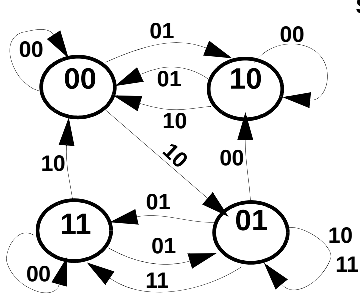
The three state graphs below are identical.

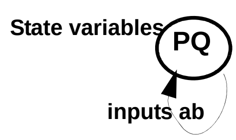
a) Using the state graph, construct the asynchronous state-tables. The state variables are P and Q, the inputs are a and b.

b) Circle the stable states.

c) Find a critical race either on the state graph or the state table. Then on the lower left graph, darken the copy of the transition that has the critical race.

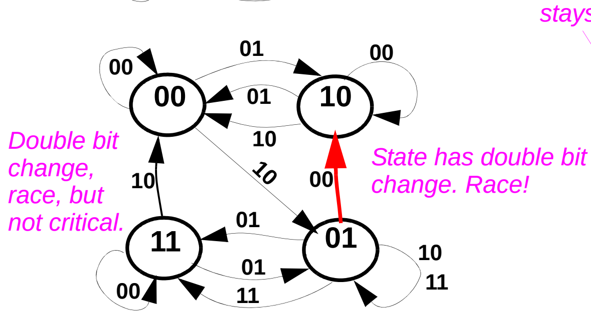
d) Find any cycle(s) and darken their arrows on the lower right state graph.



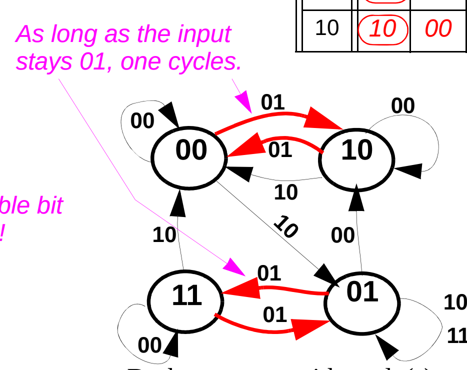


State table

state PQ	p+ q+ ab= 00	ab= 01	ab= 11	ab= 10
00	(00)	10		01
01	10	11	(01)	(01)
11	(11)	01		00
10	(10)	00		00



Darken arrows with races



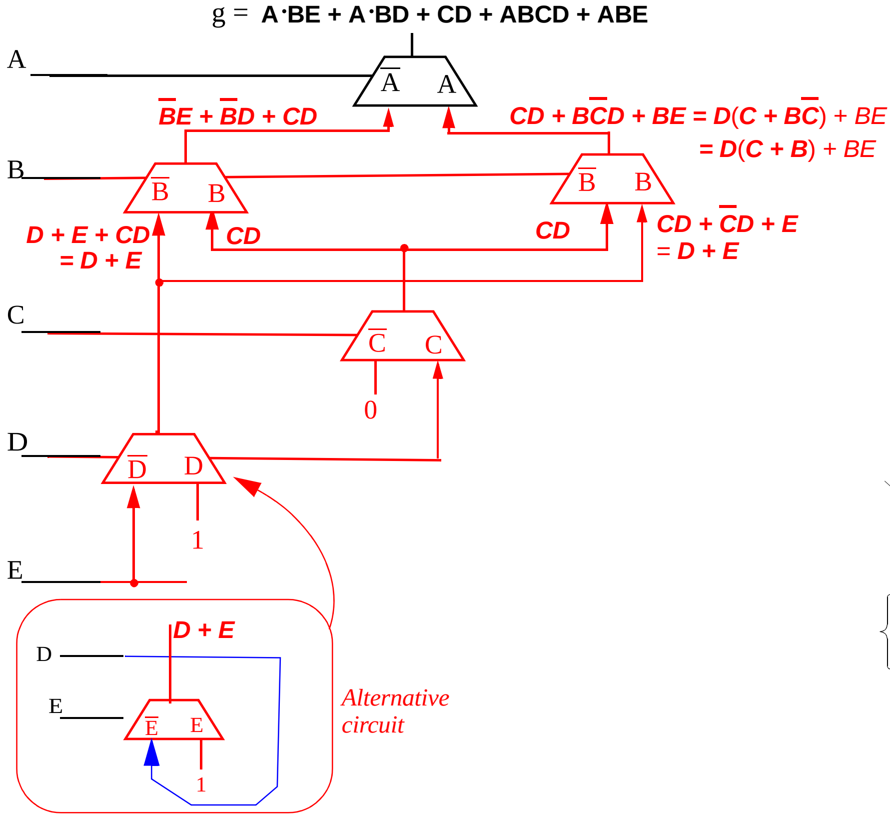
Darken arrows with cycle(s)

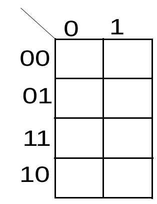
11 Mux Logic

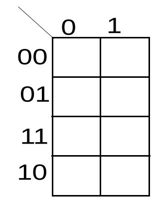
6%

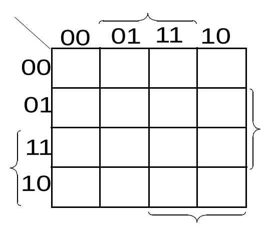
Implement the function g using nothing but 2-input MUXs and possibly inverters.
Write the subexpressions at the output of each mux.
Expand in the order A, B, C, D, E as indicated.
Marks are removed for unnecessary MUXs..

$g = \bar{A}\cdot\bar{B}E + \bar{A}\cdot\bar{B}D + CD + AB\bar{C}D + ABE$









12 Hazards

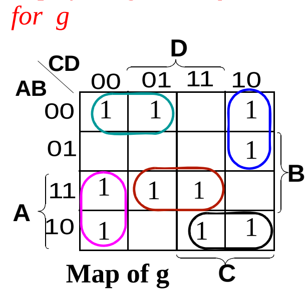
3%

One way to implement the function **g** shown on the map, is using

g = $\bar{A}\cdot\bar{B}\cdot\bar{C} + \bar{A}\cdot C\cdot\bar{D} + A\cdot\bar{B}C + A\cdot\bar{C}\cdot\bar{D} + ABD$

Modify the equation so that the function g, implemented according to your modified equation, has no single-variable-change hazards.

Loops from given equation for g



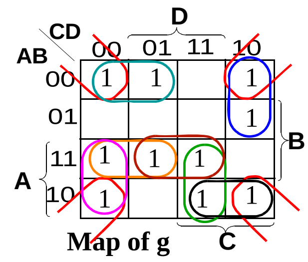
blanks are zeros

g = $\bar{A}\cdot\bar{B}\cdot\bar{C} + \bar{A}\cdot C\cdot\bar{D} + A\cdot\bar{B}C + A\cdot\bar{C}\cdot\bar{D} + ABD$

$+ \bar{B}\cdot\bar{D} + ACD + AB\bar{C}$ *added to mask hazards*

Note that without $\bar{B}\cdot\bar{D}$ *one would get hazards in the wrap around.*

Masks added



13 Do either a) or b) but not both.

a) Asynchronous State Assignment.

9%

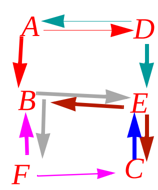
For the following state table, revise the table and select a critical-race-free state assignment.
Do not add any states or change the stable states.
Keep A as state 000.
Do not increase the number of rows in the table.

-- means "don't care"

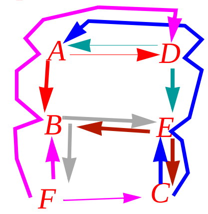
State	Next State Input XY=00	Input XY=01	Input XY=11	Input XY=10
A	B	A	D	--
B	B	F	--	E
C	C	A	C	E
D	--	A	D	E
E	B	--	C	E
F	B	F	D	--

Circle the stable states and put all the allowed transitions on the state table.

Set up an initial pseudo graph trying to put all transitions adjacent.



Can't do C->A or F->D in one jump. but can in three hops.



Write your revised state table below.

Fill in the "Next State" columns using letters **only**.You need only enter your changes.

Write your state assignment in the table/map on the right.

State	Next State Input XY=00	Input XY=01	Input XY=11	Input XY=10
A	B	A	D	--
B	B	F	-- ***A***	E
C	C	A ***E***	C	E
D		A	D	E
E	B	-- ***D***	C	E
F	B	F	D ***B***	--

Only put letters here

	0	1
00	A	*D*
01	*B*	*E*
11	*F*	*C*
10		

Write your state assignment here

One of many correct assignments.

If you cannot get a race free answer:
present your best answer above, and indicate here the number of races in your solution.

Number of races______________

13 b) Hazardous Design (If you did 13a, skip this)

9%

a) The type of circuit where glitches do not matter is called ____*asynchronous*____

The type of circuit where glitches can change the machine operation are called __*synchronous*__

b) **Hazards location**

Find any single-variable change hazards in the circuit represented by the equation below.

(i) Indicate what you did about the long overbar. __*Used Demorgan's theorem*

(ii) State any variables that obviously have no hazards. ___*B and E*__

(iii) Find the hazards, and state in what variable, and of what type.