ELG 5195 SAMPLE EXAMINATION

Question 1

Find the multiple-output prime implicants of the following set of functions by applying the modified Quine-McCluskey method:

$f(a,b,c) = \Sigma m(1, 2, 5) + dc(6, 7)$
$g(a,b,c) = \Sigma(3, 5, 6, 7) + dc(1,4)$
$h(a,b,c) = \Sigma(1, 4, 6) + dc(0)$

Group	PI	a	b	c	f	g	h	
0	0	0	0	0	0	0	x	√
1	1	0	0	1	-	**x**	-	**A**
	2	0	1	0	-	0	0	√
	4	1	0	0	0	x	-	√
2	3	0	1	1	0	-	0	√
	5	1	0	1	-	-	0	√
	6	1	1	0	x	-	-	√
3	7	1	1	1	x	-	0	√

Group	PI	a	b	c	f	g	h	
0	0,1	0	0	-	**0**	**0**	-	**B**
	~~0,2~~	~~0~~	~~-~~	~~0~~	~~0~~	~~0~~	~~0~~	~~-~~
	0,4	-	0	0	**0**	**0**	-	**C**
1	1,3	0	-	1	0	-	0	√
	1,5	-	0	1	-	-	**0**	**D**
	~~2,3~~	~~0~~	~~1~~	~~-~~	~~0~~	~~0~~	~~0~~	~~-~~
	2,6	-	1	0	-	**0**	**0**	**E**
	4,5	1	0	-	0	-	0	√
	4,6	1	-	0	**0**	-	-	**F**
2	3,7	-	1	1	0	-	0	√
	5,7	1	-	1	-	-	**0**	**G**
	6,7	1	1	-	**x**	-	**0**	**H**

Group	PI	a	b	c	f	g	h	
0	~~0,1,4,5~~	~~-~~	~~0~~	~~-~~	~~0~~	~~0~~	~~0~~	~~-~~
	~~0,2,4,6~~	~~-~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~-~~
1	1,3,5,7	-	-	1	**0**	-	**0**	**I**
	~~2,3,6,7~~	~~-~~	~~1~~	~~-~~	~~0~~	~~0~~	~~0~~	~~-~~
	4,5,6,7	1	-	-	**0**	-	**0**	**J**

Acceptable if terms with fgh = 000 are missing!!!

PI					f			g				h			
		a	b	c	1	2	5	3	5	6	7	1	4	6	
D	b'c	-	0	1	1,5	1		1		1						fg
G	ac	1	-	1	5,7			1		1		1				
A	a'b'c	0	0	1	1	1							1			fh
F	ac'	1	-	0	4,6						1			1	1	gh
E	bc'	-	1	0	2,6											f
H	ab	1	1	-	6,7					1		1				
I	c	-	-	1	1,3,5,7				1	1		1				g
J	a	1	-	-	4,5,6,7					1	1	1				
B	a'b'	0	0	-	0,1								1			h
C	**b'c'**	-	0	0	0,4									1		

OR:

PI_{fg} = b'c, ac
PI_{fh} = a'b'c
PI_{gh} = ac'
PI_{f} = bc'
PI_{g} = ab, c, a
PI_{h} = a'b', b'c'

Acceptable :

PI_f ={ b'c, ac, a'b'c, bc'}
PI_g = { b'c, ac, ac', ab, c, a}
PI_h = { a'b'c, ac', a'b', b'c'}

Question 2

Digital engineer B. I. Nary has just completed the design of a sequential circuit which has the following state table:

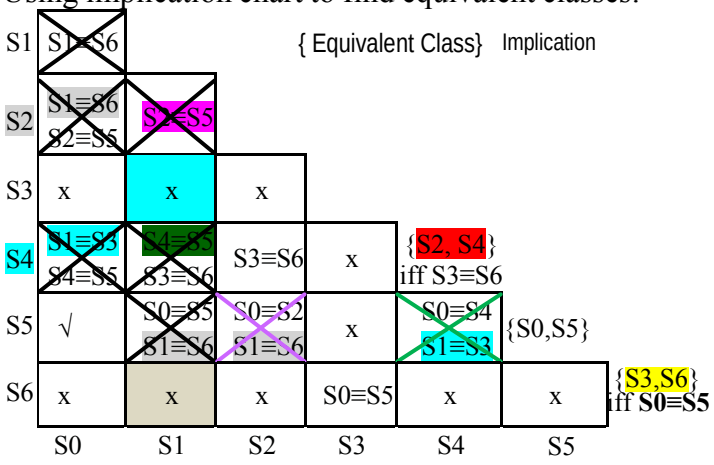
Present State	Next State		Output	
	x = 0	x = 1	x = 0	x = 1
S0	S5	S1	0	0
S1	S5	S6	0	0
S2	S2	S6	0	0
S3	S0	S1	1	0
S4	S4	S3	0	0
S5	S0	S1	0	0
S6	S5	S1	1	0

His assistant, F. L. Ipflop, who has just completed this course, claims that his design is fully equivalent to Mr. Nary's circuit. Mr. Ipflop's design has the following state table:

Present State	Next State / Output	
	x = 0	x = 1
a	a/0	b/0
b	a/0	c/0
c	a/1	b/0

(a) Is Mr. Ipflop correct? (Prove your answer.)

Using implication chart to find equivalent classes:



➔ Equivalence classes:
{S0, S5}, {S2,S4}, {S3,S6}, {S1}

Reorganized original B. I. Nary's state table

	Present State	Next State / Output	
		x = 0	x = 1
u	S0	S5/0	S1/0
	S5	S0/0	S1/0
v	S1	S5/0	S6/0
w	S2	S2/0	S6/0
	S4	S4/0	S3/0
z	S3	S0/1	S1/0
	S6	S5/1	S1/0

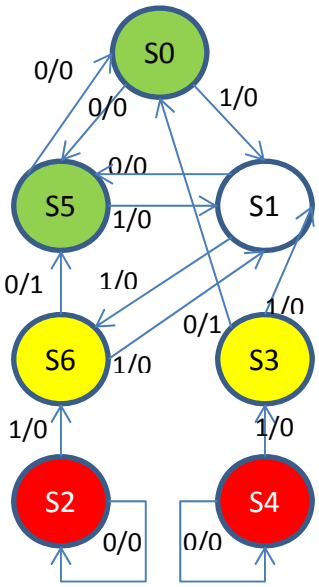
B. I. Nary's reduced table to be compared with F. L. Ipflop's

Present State	Next State	
	x =	x = 1
u	u/0	v/0
v	u/0	z/0
w	/0	z/0
z	u/1	v/0

F. L. Ipflop's FSM is not totally equivalent to B. I. Nary's FSM, since the F. L. Ipflop's FSM is missing state w (S2,S4).

So (a) **Mr. Ipflop is not correct**!

b) If Mr. Nary's circuit is always started in state S0, can be used Mr. Ipflop's design to replace Mr. Nary's circuit? (Prove your answer by showing equivalent states, state diagrams, etc)

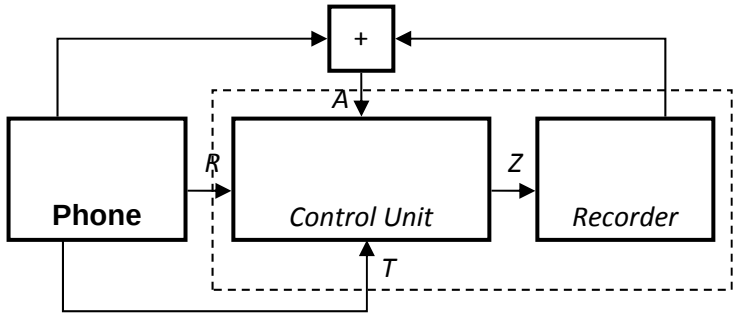


Yes, it can, since the missing state in Ipflop's design w (S2,S4) will be never reached (as shown in the next state diagram). The FSM will have transitions only between states u, v, and z, which are equivalent to a, b, c; hence the absence of w (S2,S4) will not affect the operation of the FSM.

Question 3

A *Control Unit* for a phone answering machine has to be designed. The circuit has three inputs, *R*, *A*, and *T*, and one output, *Z*.

- $R = 1$ for one clock cycle at the end of each phone ring.
- $A = 1$ when a call is answered (by a person or by the answering machine).
- T selects whether the machine should answer the phone after two rings ($T = 0$) or four rings ($T = 1$).

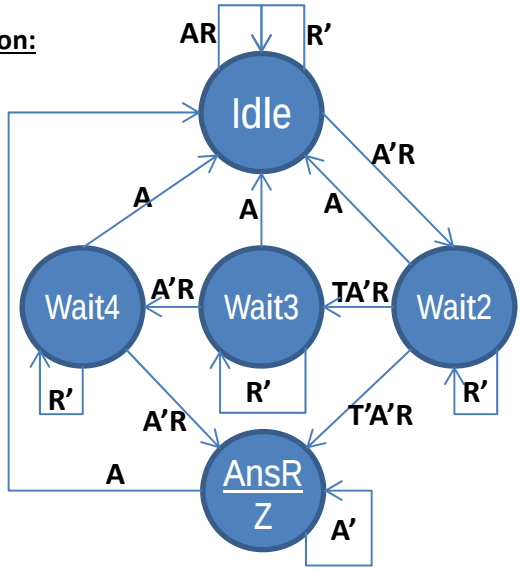


To cause the *Recorder* to answer the phone, the *Control Unit* should set the output $Z = 1$ after the end of the second ($T = 0$) or fourth ($T = 1$) ring, and hold $Z = 1$ until the *Recorder* circuit answers the phone (i.e., when *A* goes to 1), and after that the *Control Unit* gets back to the initial state.
If a person answers the phone at any time, *A* will become 1, and the *Control Unit* should go to the initial state, as well.

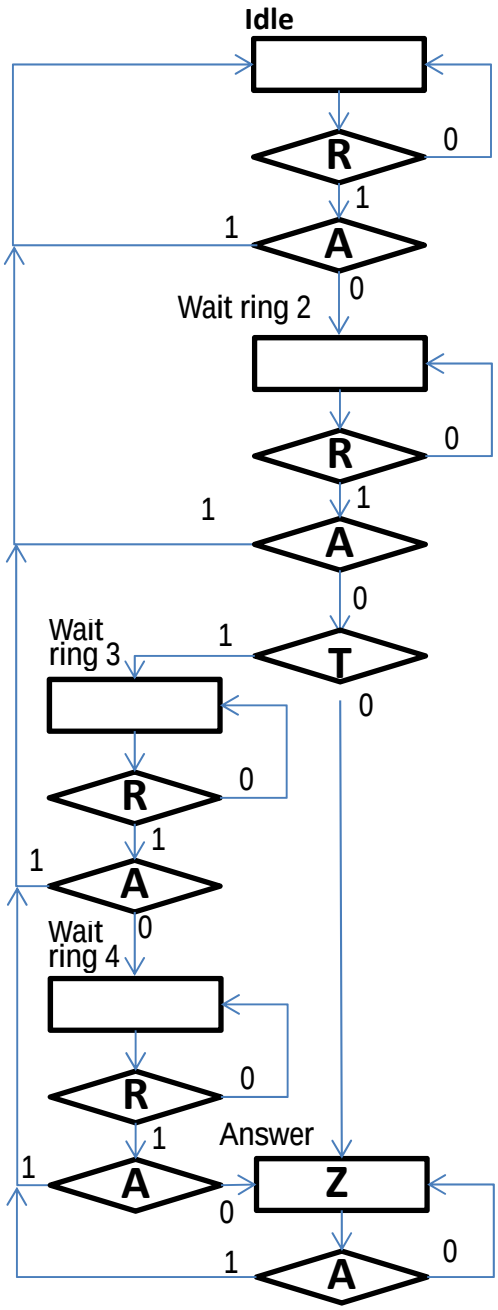
Give a **MOORE** FSM or ASM for this *Control Unit*.

Note: Assume that *T* is not changed while the phone is counting rings.

Solution:

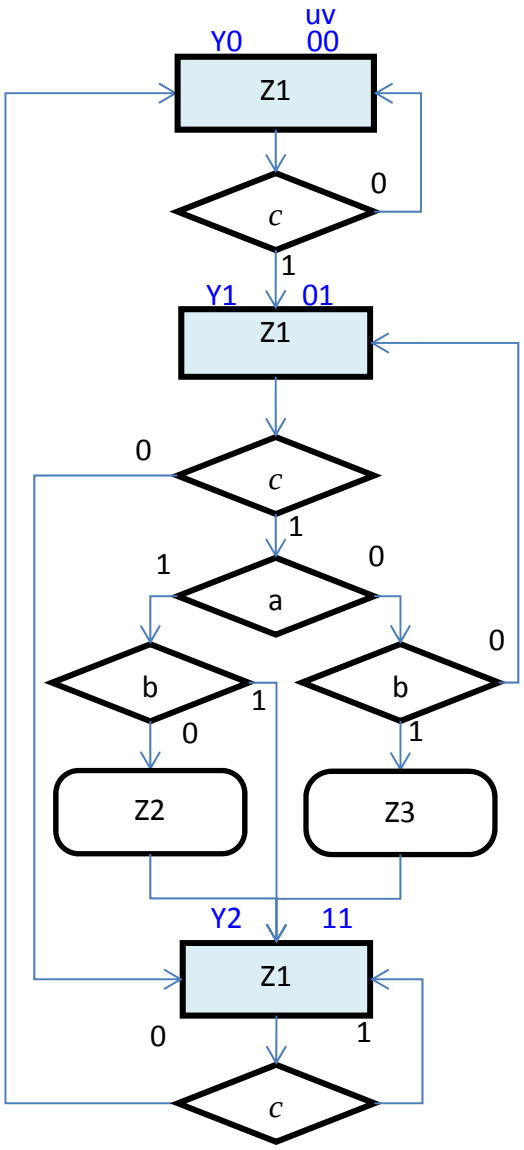


Solution:



Question 4

a) Two bits ("u" and "v", with "u" being m.s.b.) are employed to encode the states of the ASM presented in the next figure.



1) Provide the list of the input variables (***X***) and output variables (***Z***), while specifying if the later are Mealy or Moore.

X **{a,b,c};**

Moore: Z1, Mealy: Z2, Z3

2) What type is this ASM?

Type 3

3) Represent on a K-map the states of the circuit.

u \ v	0	1
0	Y0	Y1
1	-	Y2

4) Tracing the ASM link paths and using MEV (map-entered variables), draw the K-maps of the next-state variables $\{u^+ = \delta_u(u, v, \boldsymbol{X})$ and $v^+ = \delta_v(u, v, \boldsymbol{X})\}$ and the output functions $\{Z_i = \lambda(u, v, X), i=1,2,3\}$.

$u^+ = \delta_u(u, v, X)$

u^+ = u'v (a'b'c)' + uc

u \ v	0	1
0	0	(a'b'c)'
1	x	c

$v^+ = \delta_v(u, v, X)$

v^+ = c + u'v

u \ v	0	1
0	c	1
1	x	c

$Z_1 = \lambda(u, v, X)$

Z_1 = 1

u \ v	0	1
0	1	1
1	x	1

$Z_2 = \lambda(u, v, X)$

Z_2 = a b'c u'v

u \ v	0	1
0	0	cab'
1	x	0

$Z_3 = \lambda(u, v, X)$

Z_3 = a'b c u'v

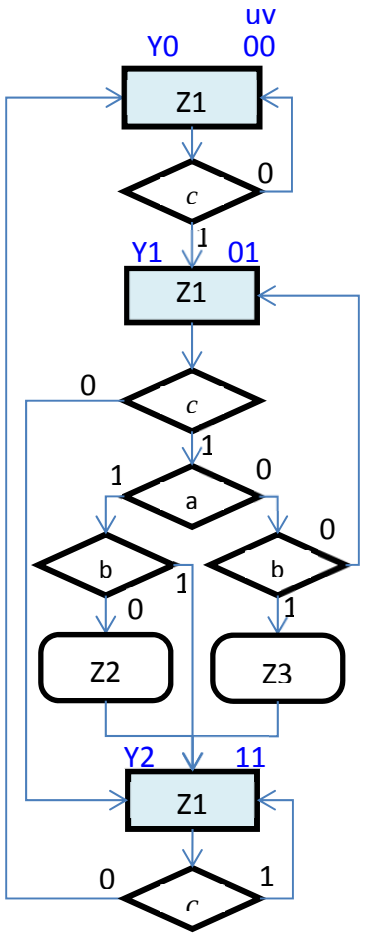
u \ v	0	1
0	0	ca'b
1	x	0

5) D flip-flops are used to implement the state register. Find the equations of the inputs of these flip-flops

D_u = u'v (a'b'c)'+ uc

D_v = c + u'v

b) Convert this ASM into *Type*-1 *Moore* machine to being implemented with PROM, parallel loading register (not counter) and multiplexers.



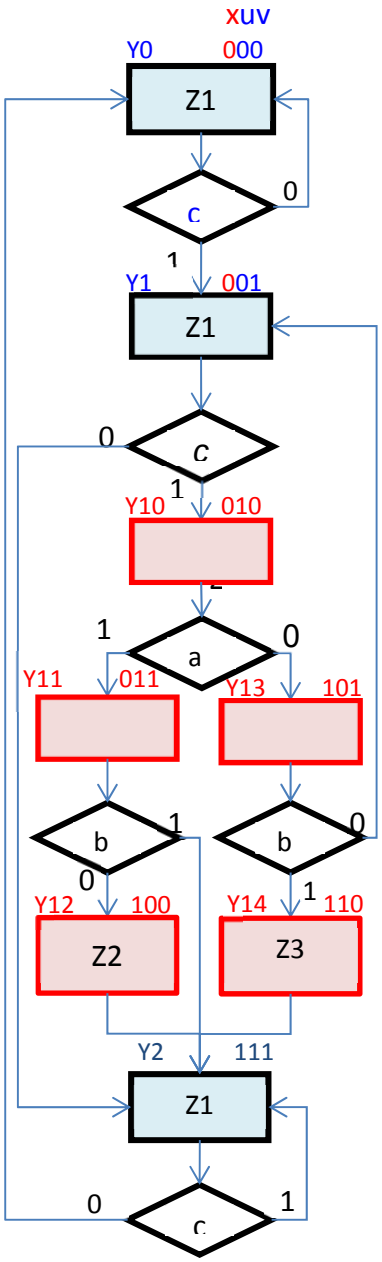
2) Draw the logic block diagram of the PROM based implementation.

3) What size of PROM is required?
8 x 11 bits

4) Provide the content of the PROM table.

	Address				Data										
	Present state			Inputs			Next State if FALSE			Next State if TRUE			Outputs		
	A2	A1	A0		D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	x	u	v		i1	i0	F_x	F_u	F_v	T_x	T_u	T_v	Z3	Z2	Z1
Y0	0	0	0	c	1	0	0	0	0	0	0	1	0	0	1
Y1	0	0	1	c	1	0	0	1	1	0	1	0	0	0	1
Y10	0	1	0	a	0	0	1	0	1	0	1	1	0	0	0
Y11	0	1	1	b	0	1	1	0	0	1	1	1	0	0	0
Y12	1	0	0	-	d	d	1	1	1	1	1	1	0	1	0
Y13	1	0	1	b	0	1	0	0	1	1	1	0	0	0	0
Y14	1	1	0	-	d	d	1	1	1	1	1	1	1	0	0
Y2	1	1	1	c	1	0	0	0	0	1	1	1	0	0	1

1) Draw the ASM chart of the converted machine.



2) **Next state δ if tested variable is FALSE or TRUE**