Problems in Old Midterm 1 Exam with Solution

To receive credits show all your work clearly. Write only answer gets 0.

1. **Number system and conversion**

1a.1) Convert 155 to

a) ***Binary*** number and

b) ***Hexadecimal*** number

	128	64	32	16	8	4	2	1
155	1			1	1		1	1

=> binary = 10011011

=> Hexadecimal = 9B

1a.2) Convert 764 in octal-based number (base-8) to

a) ***Binary*** number and

b) ***Hexadecimal*** number

Each octal digit is 3-bit binary

764 111 110 100 (binary)

1F4 (hexadecimal)

1b) Find a ***binary number*** of a given *decimal* number **456**

1 1 1 0 0 1 0 0 0

256 128 64 32 16 8 4 2 1

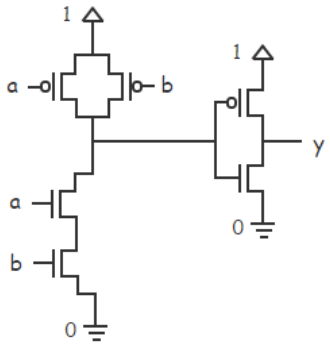
1c) Find ***hexadecimal number*** of a given *octal (based-8)* number $\mathbf{6471}_8$

6471 = 110100111001 in binary

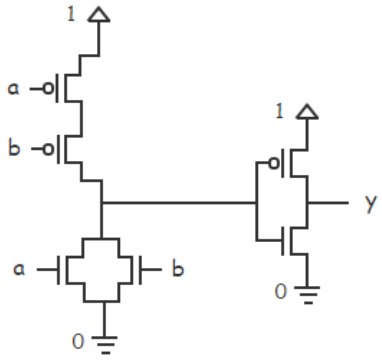
=> 1101 0011 1001 = D39 in hexadecimal

2) Given the circuit implemented with transistors, choose (circle) the one that implements the logic 2-input **NOR gate**.

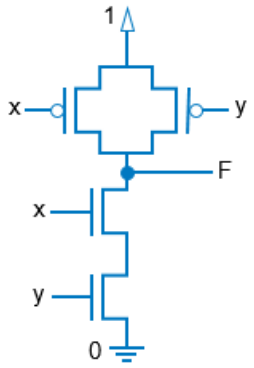
(a)



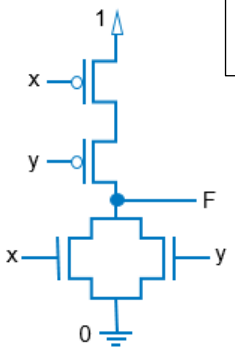
(b)



(c)



(d)



When both x and y are 0, F = 1

x	y	F = (x+y)′
0	0	1
0	1	0
1	0	0
1	1	0

3) Which of the following is the correct Boolean equation for **output d12 of a 4x16 decoder**?
Note: inputs are in the order i3 i2 i1 i0 and outputs are in the order of d15 d14 … d1 d0

(a) d12 = i3 + i2 + i1’ + i0

(b) d12 = i3 + i2 + i1’ + i0’

(c) d12 = i3 i2 i1’ i0

(d) d12 = i3’ i2’ i1 i0’

(e) d12 = i3’ i2’ i1 i0

(f) d12 = i3 i2 i1’ i0’

4) Given a 4x2 priority encoder (where d3 has the highest priority and inputs are in the order d3 d2 d1 d0), what is the output of e1 and e0 given that d0 = 1, d1 = 0, d2 = 1, d3 = 0.

(a) e1 = 0, e0 = 0

(b) e1 = 0, e0 = 1

(c) e1 = 1, e0 = 0

(d) e1 = 1, e0 = 1

(e) The output is unknown since the input combination is invalid for a priority encoder.

5) Given F(a, b, c) = ac + b’c, which of the following is the **inverse of F**?

(a) F’ = c’

(b) F’ = a’c’ + b

(c) F’= a’b + c’

(d) F’ = a’c’ + bc’

(e) F’ = a’b + c

(f) None of the above

F’ = (ac + b’c)’ = (ac)’ . (b’c)’
= (a’ + c’)(b + c’)
= a’b + a’c’ + c’b + c’
= a’b + c’(a’ + b + 1)
= a’b + c’

6) Given the following logic equation

F = A'B' + A'BC' + B'C + A(B+C)'

Minimize/Simplify using *only* *Boolean algebra*.

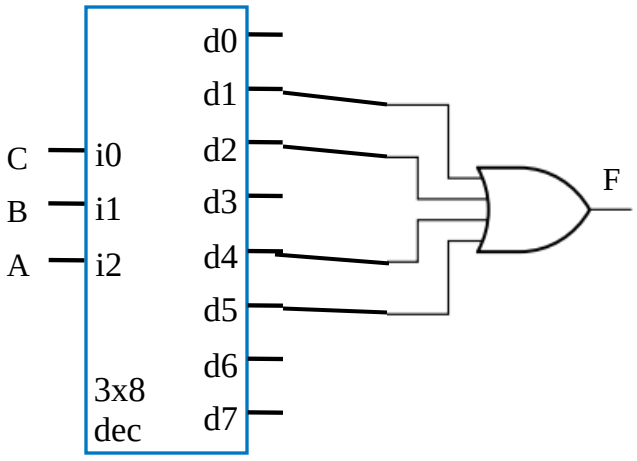
F = A'B' + A'BC' + B'C + AB'C'
= A'(B' + BC') + B'(C + AC')
= A'(B' + C') + B'(C + A)
= A'B' + A'C' + B'C + B'A
= A'B' + AB'+ A'C' + B'C
= B'+ B'C+ A'C'
= B'(1+ C)+ A'C'
= B'+ A'C'

Double check using K-map

A \ BC	00	01	11	10
0	1	1		1
1	1	1		

=> F = B' + A'C'

7) Given the following circuit using ***3x8 decoder and 4-input OR gate***, write the Boolean equation/expression in a **sum-of-minterm** form.



F = d1 + d2 + d4 + d5

i2	i1	i0		
A	B	C	from 3x8 decoder	F
0	0	0	d0 = 1	0
0	0	1	d1 = 1	1
0	1	0	d2 = 1	1
0	1	1	d3 = 1	0
1	0	0	d4 = 1	1
1	0	1	d5 = 1	1
1	1	0	d6 = 1	0
1	1	1	d7 = 1	0

F = d1 + d2 + d4 + d5
= A'B'C + A'BC' + AB'C' + AB'C

8) Given the following Boolean equation:

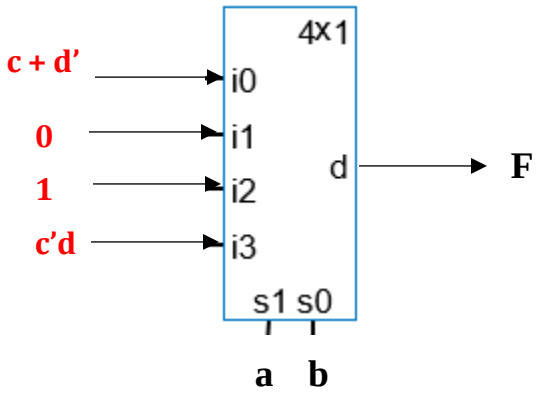
$$F(c, d, a, b) = \Sigma(0, 2, 6, 7, 8, 10, 12, 14)$$

8a) Use Karnaugh-map to find its ***minimal*** sum-of-product form.

cd\ab	00	01	11	10
00	1			1
01			1	1
11	**1**			**1**
10	**1**			**1**

F = c'da + **cb'** + d'b'

8b) Implement the given equation F(c, d, a, b) using 4x1 MUX with signals a and b connecting to select lines S1 and S0, respectively.



From K-map above, look at one column at a time. Why?

Look at **1st column**, when ab = 00, use ab = 00 in 4x1 mux above, F = i0 (from mux)
From first coln, c \d 0 1
0 1 0 => c + d'
1 1 1
Therefore, the circuit at i0 is c + d'

Look at **2nd column**, when ab = 01, use ab = 01 in 4x1 mux above, F = i1 (from mux)
Since value from the 2nd column are all logic 0, the circuit (or logic) at i1 is 0

Look at **4th column**, when ab = 10, use ab = 10 in 4x1 mux above, F = i2 (from mux)
Since value from the 4th column are all logic 1, the circuit (or logic) at i2 is 1

Look at **3rd column**, when ab = 11, use ab = 11 in 4x1 mux above, F = i3 (from mux)
the circuit (or logic) at i3 is c'd

9) Write the ***simplified/minimized sum-of-products*** form for F using **K-map**

F = (x + y + w')(x' + y' + z)(x' + w + z')

From F, the term (x + y + w') = 0 when x = 0, y =0 and w = 1

the term (x' + y' + z) = 0 when x = 1, y =1 and z = 0

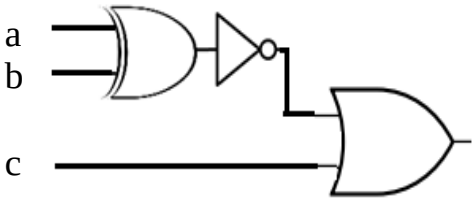
the term (x' + w + z') = 0 when x = 1, w =0 and z = 1

wz / xy	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	0
10	1	0	1	1

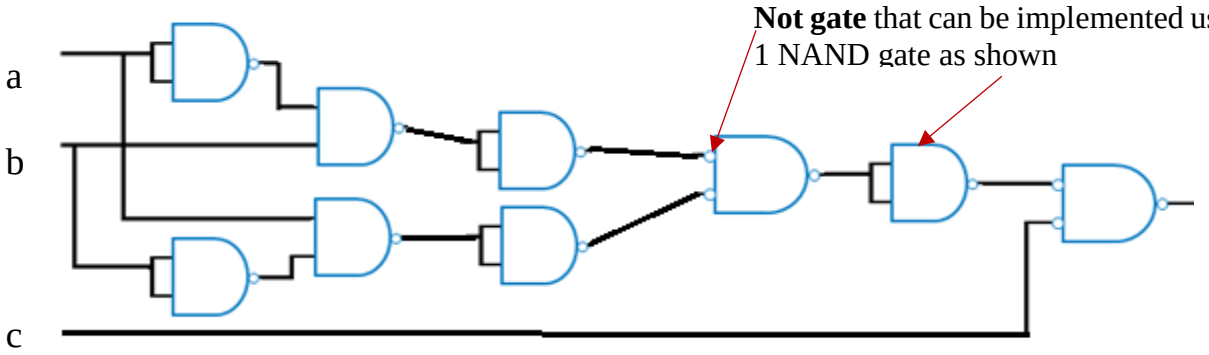
Group the terms with 1, then

F = x'y + x'w' + xwz + xy'z'

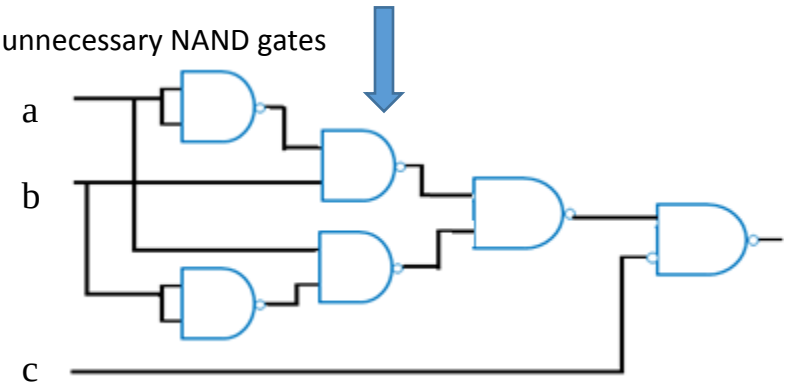
10) Given a digital circuit below, implement the circuit above using **only** ***2-input NAND gates***. Eliminate unnecessary NAND gates.



One way of implementations (there are several ways)



Eliminate unnecessary NAND gates



11) Using the combinational logic design process, design a circuit with two 2-bit inputs (A and B) and one 2-bit output (M). Your circuit should output the ***minimum*** of the two input values. For example,

- A =a1a0 = 10 and B = b1b0 = 11, then M = m1m0 = 10

  since 10 (= 2 in decimal) < 11 (= 3 in decimal)
- A =a1a0 = 11 and B = b1b0 = 01, then M = m1m0 = 01
- A = 10 and B = 10, then M = 10 since both inputs are the same value

Your answers must include
a) **Truth table**
b) **minimized** sum-of-products form for the outputs

a1	a0	b1	b0	m1	m0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	1	1

For m1,

a1a0 \ b1b0	00	01	11	10
00				
01				
11			1	1
10			1	1

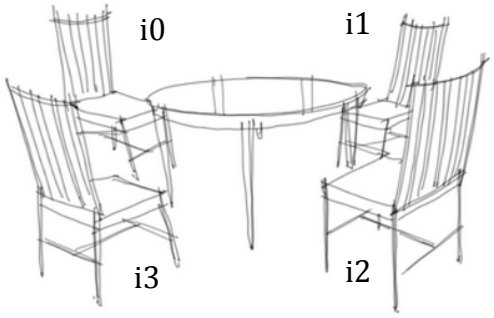
Group the terms with 1, then
m1 = a1b1

For m0,

a1a0 \ b1b0	00	01	11	10
00				
01		1	1	1
11		1	1	
10		1		

Group the terms with 1, then
m0 = a0b0 + a1'a0b1 + a1b1'b0

12) Consider four seats arranged in a circle and described by **Boolean variables i0** to **i3**.



Boolean variable i0 is true (=1) if seat 0 is occupied and i0 is false if the seat is not occupied (no one is sitting in the seat), likewise for i1, i2, and i3.

a) Write a **truth table that is true (F = 1) if <u>at least</u> two people are sitting next to each other and <u>at least</u> one seat is not occupied.**

b) Simplify your design using K-map. Write the expression/equation in minimal sum-of-product form.

Inputs				Output
i0	i1	i2	i3	F
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	1
0	1	0	0	
0	1	0	1	
0	1	1	0	1
0	1	1	1	1
1	0	0	0	
1	0	0	1	1
1	0	1	0	
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	

K-map

i0i1\i2i3	00	01	11	10
00			**1**	
01			**1**	1
11	**1**	**1**		1
10		1	1	

F = i0i1i2' + i0i1'i3 + **i0'i2i3** + i1i2i3'

13) A "2-bit" comparator circuit receives **two 2-bit numbers**, A = A_1A_0, and B = B_1B_0. Design this comparator that produces **an output F = 1 if A ≥ B.** ***Write its <u>minimum</u> sum-of-product form.***

$A_1A_0 \backslash B_1B_0$	00	01	11	10
00	**1**			
01	**1**	1		
11	**1**	1	1	**1**
10	**1**	1		**1**

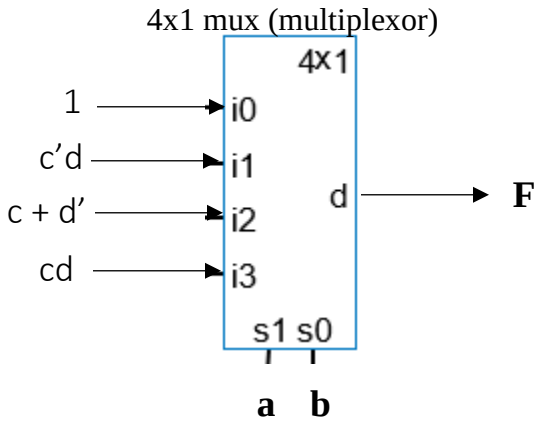
F = **B1'B0'** + A0B1' + A1B1' + A1A0 + **A1B0'**

14) Given the following circuit,

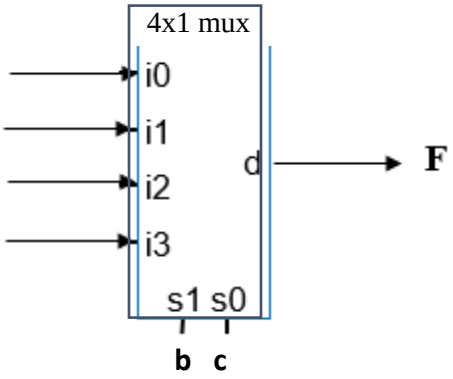
a) write its *minimized sum-of-products form* for F

cd / ab	00	01	11	10
00	1	1	1	1
01		1		
11			1	
10	1		1	1

Group the terms with 1, then
F = b'd' + a'c'd + acd + b'c
Or
F = b'd' + a'c'd + acd + a'b'

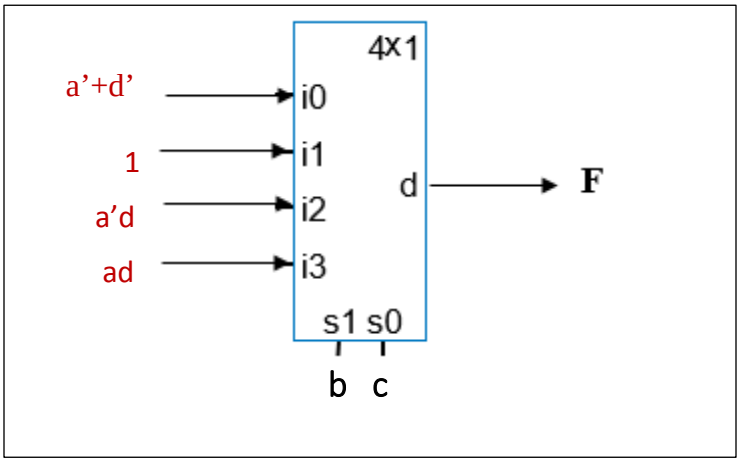


b) Implement the simplified equation of F from 14a) using 4x1 MUX with signals b and c connecting to select lines S1 and S0, respectively, as shown below.

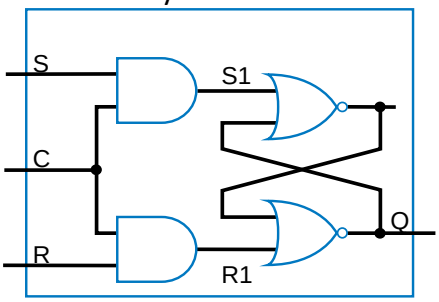


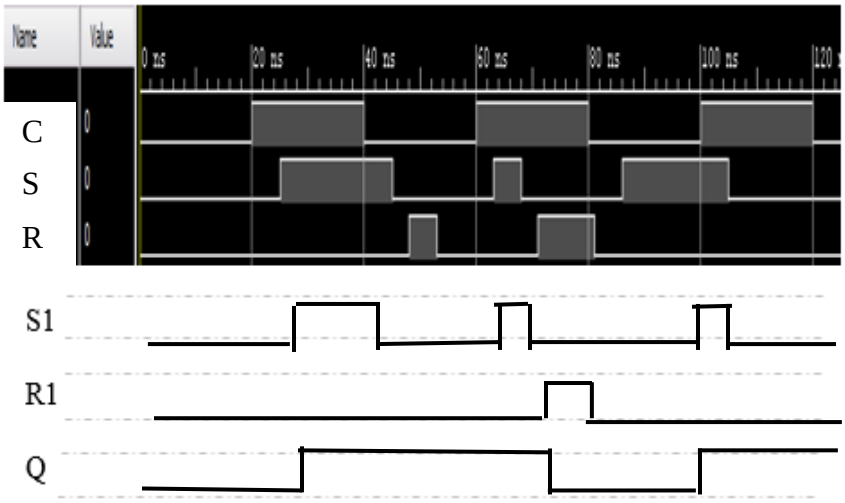
From the equation in 14a), we can write the following K-map

bc \ ad	00	01	11	10
00	1	1		1
01	1	1	1	1
11			1	
10		1		

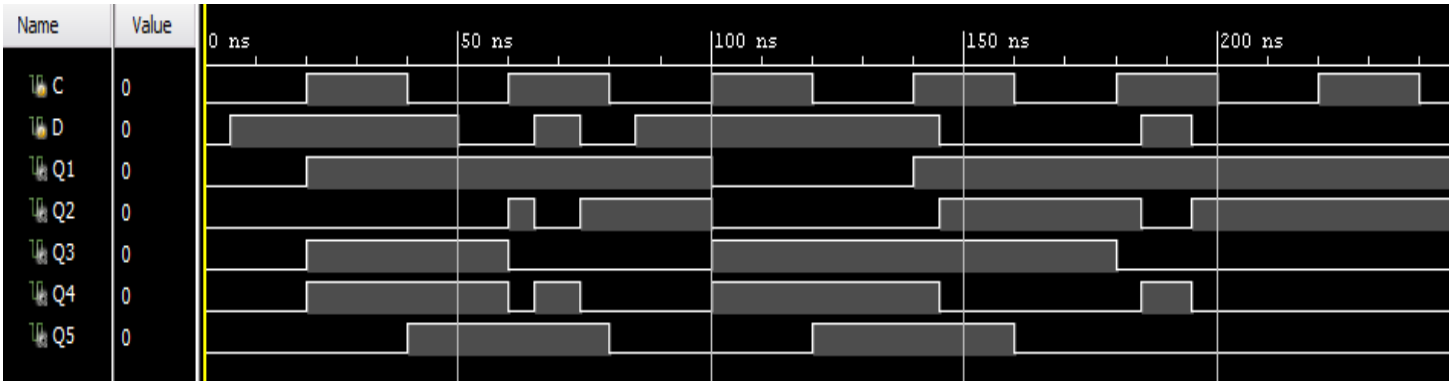


15) Given a level-sensitive SR Latch with the input pattern below, complete the timing diagram below (draw S1, R1, and Q). Assuming that S1, R1, and Q are initially 0 and the logic gates have a *very small* nonzero delay.



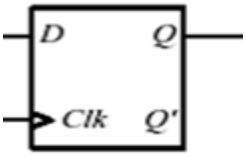


16) Given the following waveform where C and D are inputs and the rest are outputs, which output waveform (Q1, Q2, Q3, Q4, or Q5) belongs to



D-FF using rising edge of Clk _____Q3_____

Choose one of Q1, Q2, Q3, Q4, Q5

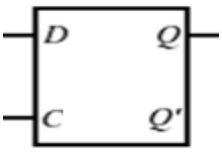


Clk	Q
↑	D

Note: ***C*** and **D** in the ***above waveform*** are connected to ***Clk*** and **D**, respectively of this flip-flop.

D-latch with the function table below _____ Q4_____

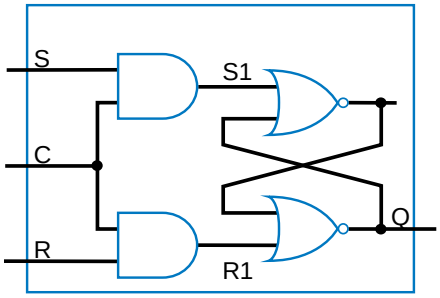
Choose one of Q1, Q2, Q3, Q4, Q5

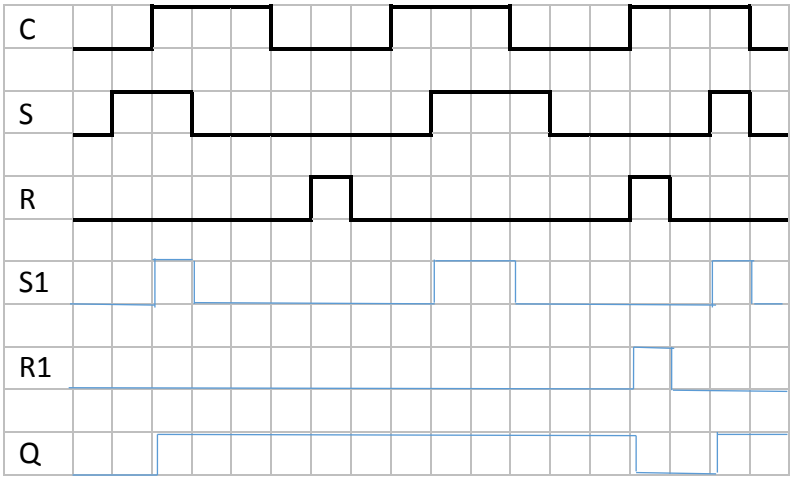


C	Q
1	D
0	Q (stay the same)

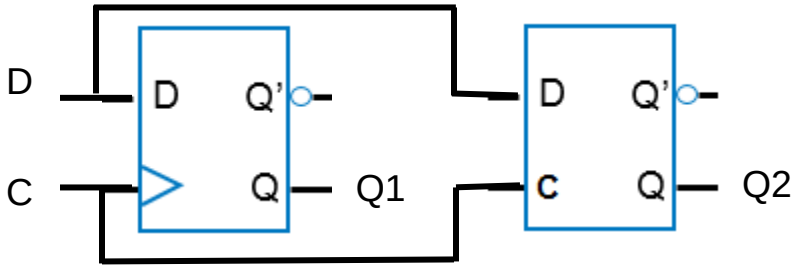
Note: ***C*** and **D** in the ***above waveform*** are connected to ***C*** and **D**, respectively of this latch.

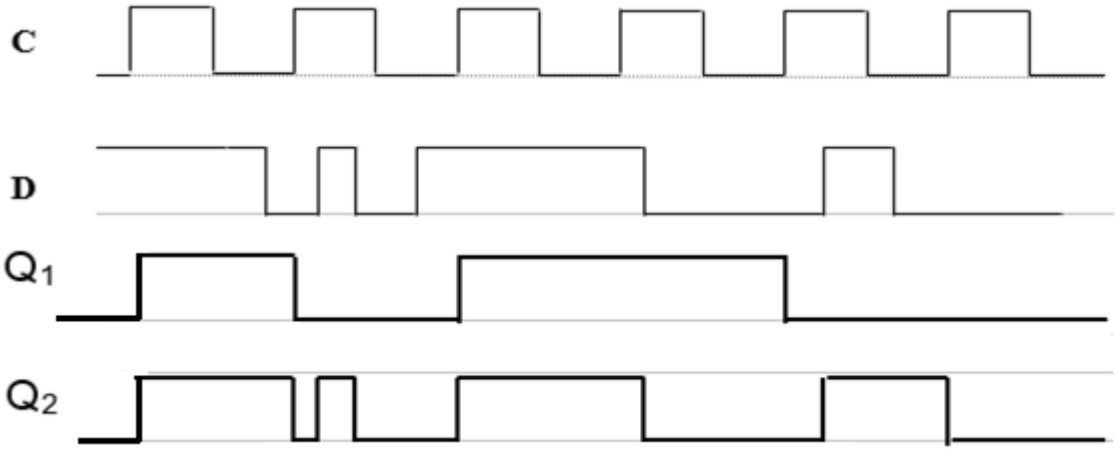
17) Given a level-sensitive SR Latch with the input pattern below, complete the timing diagram below (draw S1, R1, and Q). Assuming that S1, R1, and Q are initially 0



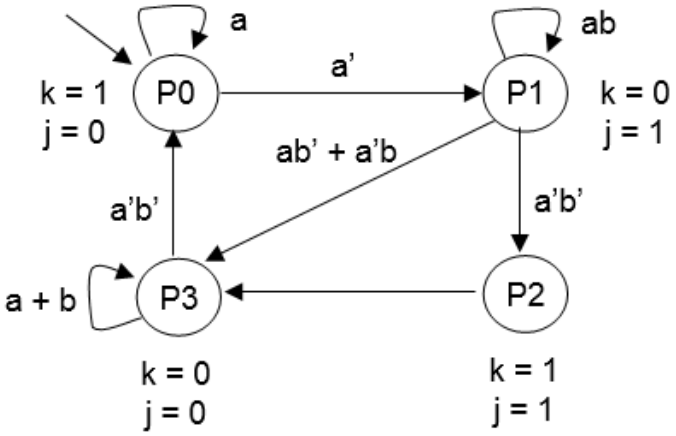


18) Trace the behavior of **D-Latch** and **D flip-flop** for the following input patterns. Assume that Q1 and Q2 is initially 0 and logic gates have a very small nonzero delay.





19) Using the **sequential logic design process**, **convert the provided state diagram (FSM) to a circuit**. Be sure to label and show each step.
Note: **Write the Boolean equations (in minimum sum-of-products form) is sufficient.** *NO need to draw a circuit.*



Inputs: a, b
Outputs: k, j

4 states -> 2 D-FFs (2-bit state) are used. P0 = 00, P1 = 01, P2 = 10 and P3 = 11

S1	S0	a	b	n1	n0
0	0	0	0	0	1
0	0	0	1	0	1
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	1	1	1
0	1	1	0	1	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

Next state n1 and n0

n1	ab			
S1S0	00	01	11	10
00				
01	1	1		1
11		1	1	1
10	1	1	1	1

n1 = S1S0' +S1b + S0ab' + S1'S0a'
or
n1 = S1S0' +S1a + S0a'b + S1'S0a'

n0	ab			
S1S0	00	01	11	10
00	1	1		
01		1	1	1
11		1	1	1
10	1	1	1	1

n0 = S0'a' + a'b + S0a + S1a
or
n0 = S1S0' + S0a + a'b +S0'a'

Outputs k and j

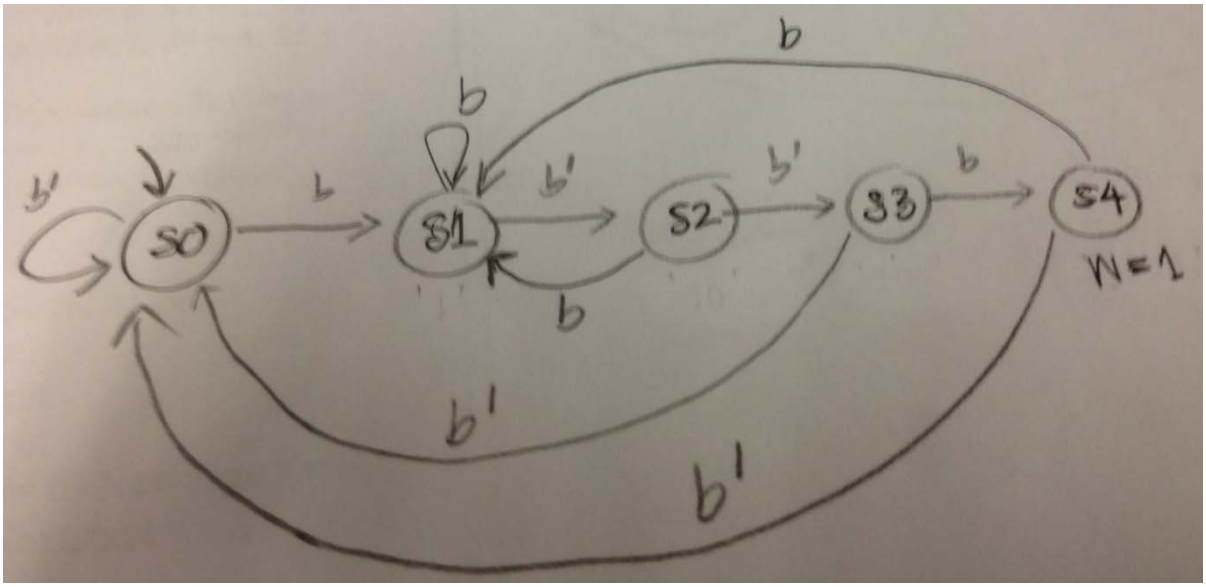
S1	S0	k	j
0	0	1	0
0	1	0	1
1	0	1	1
1	1	0	0

k = s0'
j = s1's0 + s1s0'

20) Design a synchronous sequential circuit of a sequence detector, which has one input b and one output w. It accepts a sequence of bits (one bit (0 or 1) at a time) and outputs w = 1 when the **target sequence 1001** is detected. ***Non-Overlapping*** *sequence is detected.*

a) Draw its **state diagram**

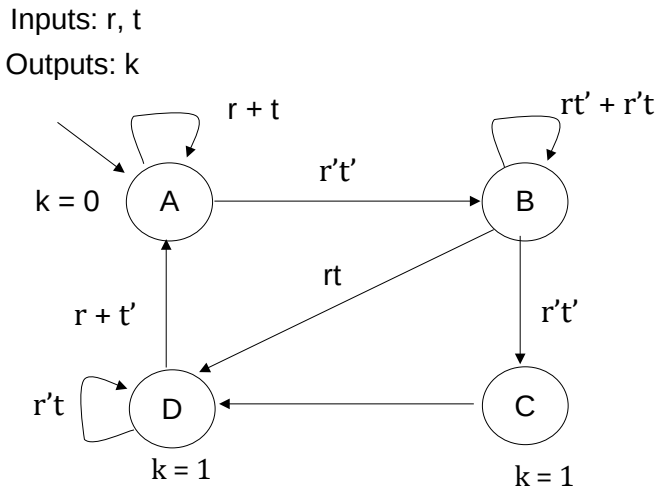
b) ***Minimum of bits*** to represent/encode states in your designed state diagram are _____



Minimum of bits to represent/encode states in your designed state diagram are ___3_____

21) Using the sequential logic design process, convert the provided state diagram (FSM) to a circuit. Be sure to label and show each step.

Note: Write the Boolean equations (in minimum sum-of-products form) is sufficient. You <u>DO NOT</u> need to provide a gate-level implementation (no drawing a circuit).



Let A – 00, B – 01, C – 10, D – 11

s1	s0	r	t	k	n1	n0
0	0	0	0	0	0	1
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	1	0
0	1	0	1	0	0	1
0	1	1	0	0	0	1
0	1	1	1	0	1	1
1	0	0	0	1	1	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	0	0
1	1	0	1	1	1	1
1	1	1	0	1	0	0
1	1	1	1	1	0	0

k = s1

n1 = s1s0' + s1r't + s1's0r't' + s1's0rt

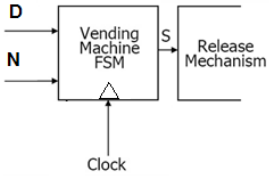
s1s0 \ rt	00	01	11	10
00	0	0	0	0
01	1	0	1	0
11	0	1	0	0
10	1	1	1	1

n0 = s1s0' + s0r't + s1's0r + s0'r't'

s1s0 \ rt	00	01	11	10
00	1	0	0	0
01	0	1	1	1
11	0	1	0	0
10	1	1	1	1

22) Design a vending machine that **delivers coffee after depositing 35 cents**.

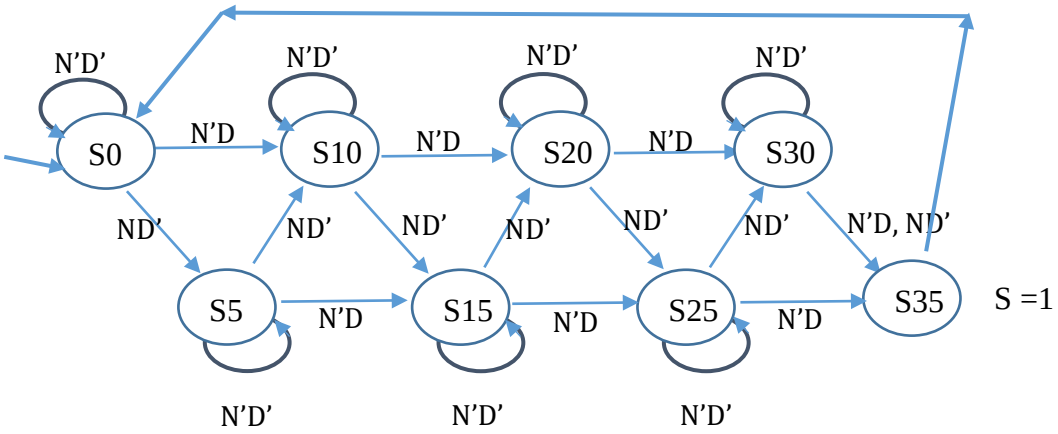
There is a single coin slot for two types of coins, N and D.
D is for dimes (10 cents) and N is for nickels (5 cents).
It does not take other coin types. Single coin slot means that you can insert one coin at a time!



If a dime is deposited, D is 1 and N is 0.
If a nickel is deposited N is 1, D is 0.
N and D cannot be 1 at the same time.
The vending machine does ***not*** return changes if you deposit more than 35 cents.

When 35 cents is reached, coffee is served by outputting S as 1 that enables the coffee dripping system. ***In the next cycle, the vending machine goes back to the initial state for the next coin input for another coffee***.

a) Draw a state diagram (FSM).
b) From your state diagram, what are the minimum of bits that your design needs to use for a state register?



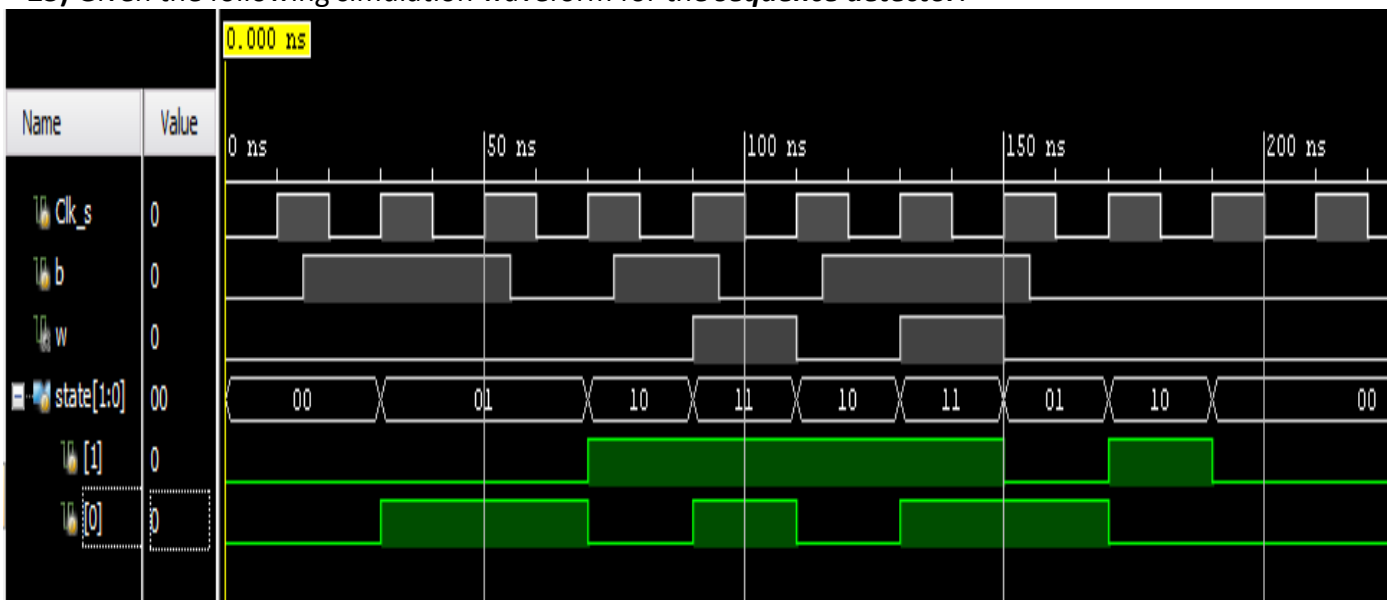
b) since there are 8 states, minimum number of bits used for the state register = 3

Note: since ND never occurs, we do not need to draw ND on the state diagram. If you want to include ND in each state, it can be either

1) when ND occurs, make it go back to S0. This way, the machine takes the money (all the coins that a buyer has entered so far) if a buyer tries to insert two coins at one time.

2) when ND occurs, make it stay at the same state (similar to N'D' case). This way, the machine will still take a buyer money (but only for 15 cents) if a buyer tries to insert two coins at one time.

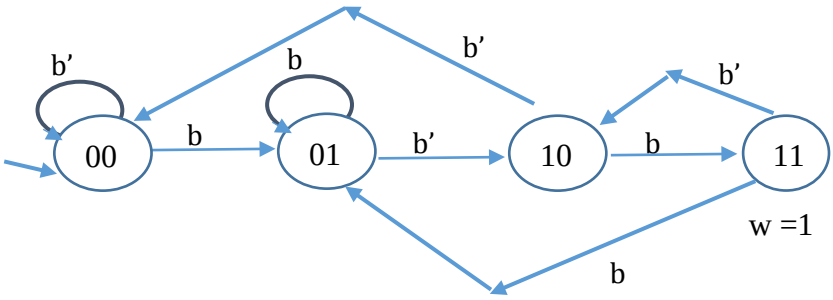
23) Given the following simulation waveform for the ***sequence detector***.



It has one input b and one output w and its state register uses 2 bits (state[1:0]) as shown in the waveform.

a) Draw its state diagram (FSM diagram).

b) What is the target sequence that this detector can detect? (Hint: starting from the initial state (00), find the input sequence such that this detector gives output w = 1.)



b) target sequence is 101 (overlapping sequence can be detected)