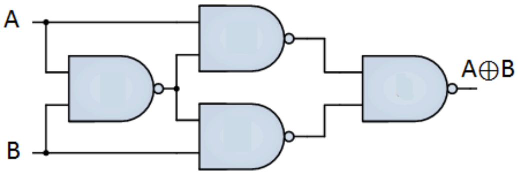
Question 2: A universal gate is a gate which can implement any Boolean function without need to use any other gate type. The NAND and NOR gates are universal gates. Design a two-input XOR gate using two-input NAND gates.



Question 3: Reduce each of the following expressions:

(a) $\text{Out} = A + \overline{(B\,.\,A)}$

$= A + (\overline{B} + \overline{A})$ (DeMorgan's)

$= A + (\overline{A} + \overline{B})$ (Commutative)

$= (A + \overline{A}) + \overline{B}$ (Associative)

$= 1 + \overline{B}$ (Identity)

$= 1$ (Identity)

(b) $\text{Out} = \overline{(A\,.\,B)}\,.\,(\overline{A} + B)\,.\,(\overline{B} + B)$

$= \overline{(A\,.\,B)}\,.\,(\overline{A} + B)\,.\,1$ (Identity)

$= (\overline{A} + \overline{B})\,.\,(\overline{A} + B)$ (Identity & DeMorgan's)

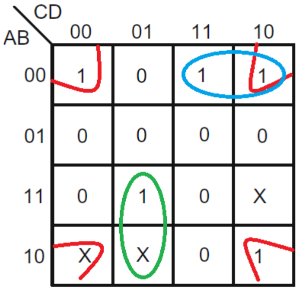
$= \overline{A} + (\overline{B}\,.\,B)$ (Inverse Distributive)

$= \overline{A} + 0$ (Identity)

$= \overline{A}$ (Identity)

Question 4: Determine the minimized SOP and minimized POS forms of the output function from the following K-map. Note that the "X" sign corresponds to a "don't care" condition that represents either logic 0 or logic 1.

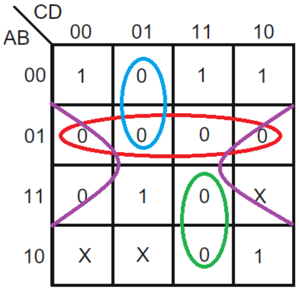
Minimized SOP:



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

$\text{Out} = (\overline{B}\,.\,\overline{D}) + (A\,.\,\overline{C}\,.\,D) + (\overline{A}\,.\,\overline{B}\,.\,C)$

Minimized POS:



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

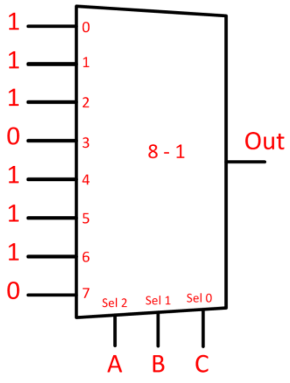
$\text{Out} = (A + \overline{B})\,.\,(\overline{B} + D)\,.\,(A + C + \overline{D})\,.\,(\overline{A} + \overline{C} + \overline{D})$

Question 5: Implement the function $\text{Out} = (\overline{\text{A}}.\text{B}.\overline{\text{C}}) + \overline{(\text{B}.\text{C})}$ using:

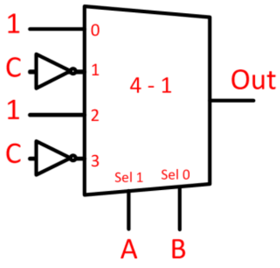
(a) One 8-1 MUX.

(b) One 4-1 MUX (and Inverters if you need.)

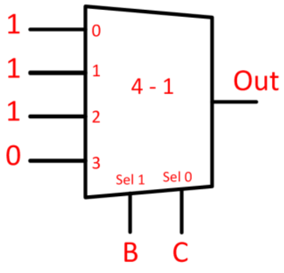
			8:1	4:1
A	B	C	Out	Out
0	0	0	1	1
0	0	1	1	
0	1	0	1	$\bar{C}$
0	1	1	0	
1	0	0	1	1
1	0	1	1	
1	1	0	1	$\bar{C}$
1	1	1	0	



(a)



(b) Solution 1

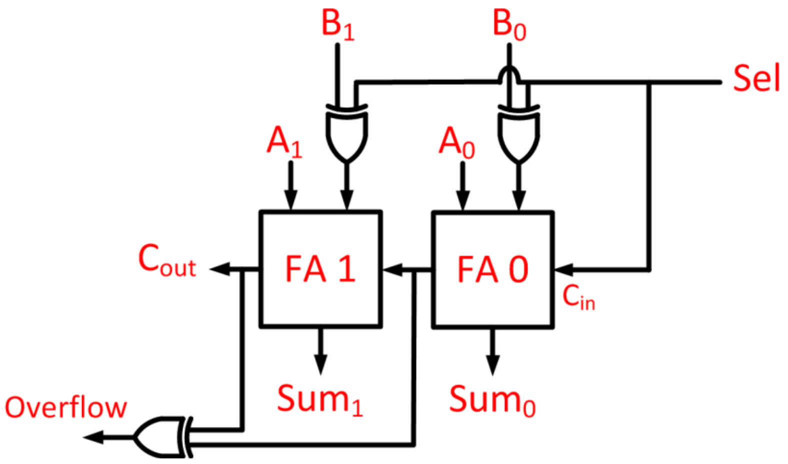


(b) Solution 2

Question 6: Design a two-bit ripple-carry adder & subtractor using one-bit full-adders and logic gates. The design should have a selection bit to choose between addition & subtraction. The design should also preserve the overflow flag.

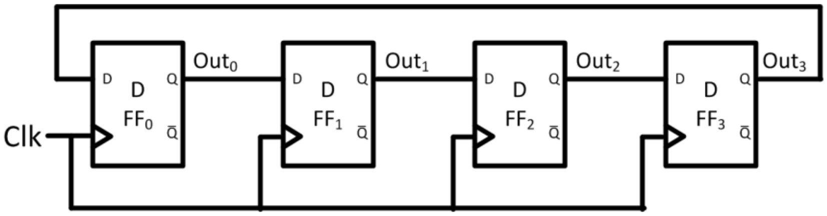
When Sel = 0, the system functions as a 2-bit adder.

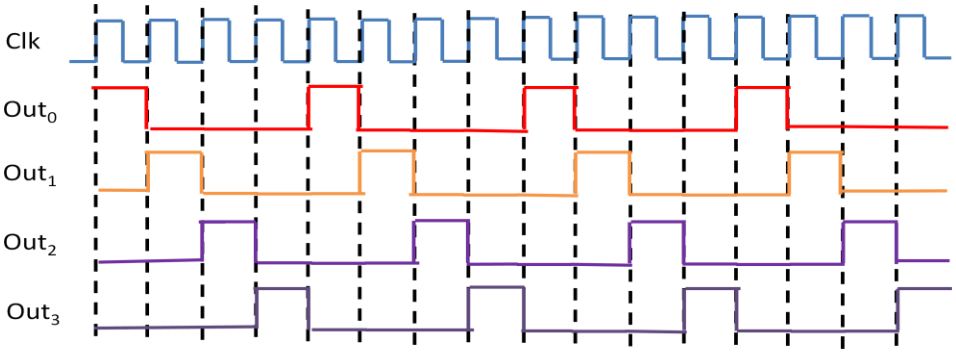
When Sel = 1, the system functions as a 2-bit subtractor.



Question 2: A ring counter is a type of synchronous counter composed of cascaded D FFs with the output of the last FF fed to the input of the first. Draw the waveforms for the following four-bit ring counter. Suppose FF_0 is initialized to 1 and all the other FFs are initialized to 0. How many numbers can be distinguished by four-bit ring counter?

After four clock cycles the wave forms are repeated. So, it can count up to 4 distinct numbers.



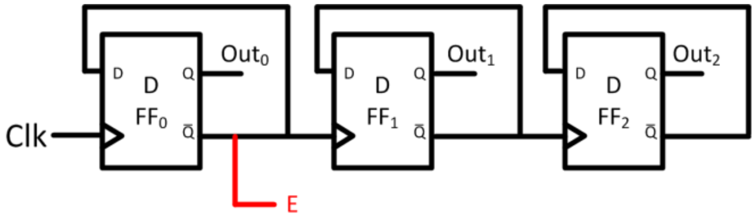


Question 3: Using D FFs (and logic gates if necessary), design a three-bit asynchronous up binary counter that detects even numbers.

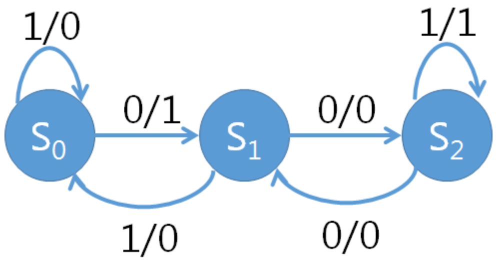
Out2	Out1	Out0	E
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

	Out_1Out_0=00	Out_1Out_0=01	Out_1Out_0=11	Out_1Out_0=10
Out_2=0	1	0	0	1
Out_2=1	1	0	0	1

$$E = \overline{Out_0}$$



Question 4: Convert the given Mealy machine to Moore machine.



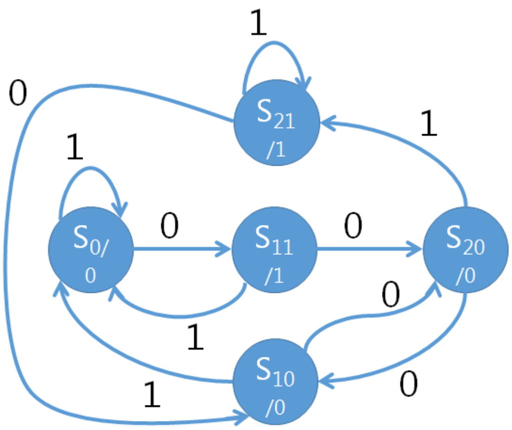
Draw the transition table of Mealy machine.

Current State	Input = 0		Input = 1	
	Next State	Output	Next State	Output
S_0	S_1	1	S_0	0
S_1	S_2	0	S_0	0
S_2	S_1	0	S_2	1

Draw the transition table of Moore machine.

Current State	Input = 0	Input = 1	Output
	Next State	Next State	
S_0	S_{11}	S_0	0
S_{10}	S_{20}	S_0	0
S_{11}	S_{20}	S_0	1
S_{20}	S_{10}	S_{21}	0
S_{21}	S_{10}	S_{21}	1

Draw Moore state diagram.



Question 5: Design the Mealy machine in Question 4 using D FFs and logic gates.

Convert the state diagram to a truth table.

Q_1	Q_0	I	$Q'_1=D_1$	$Q'_0=D_0$	O
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	1	0	0
0	1	1	0	0	0
1	0	0	0	1	0
1	0	1	1	0	1
1	1	0	X	X	X
1	1	1	X	X	X

Find simplified equations.

	$Q_0I=00$	$Q_0I=01$	$Q_0I=11$	$Q_0I=10$
$Q_1=0$	0	0	0	1
$Q_1=1$	0	1	X	X

$$D_1 = Q_1 I + Q_0\bar{I}$$

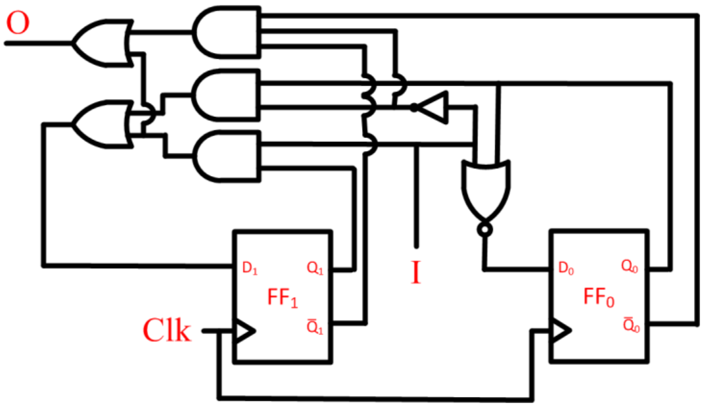
	$Q_0I=00$	$Q_0I=01$	$Q_0I=11$	$Q_0I=10$
$Q_1=0$	1	0	0	0
$Q_1=1$	1	0	X	X

$$D_0 = \bar{Q}_0\bar{I} = \overline{Q_0 + I}$$

	$Q_0I=00$	$Q_0I=01$	$Q_0I=11$	$Q_0I=10$
$Q_1=0$	1	0	0	0
$Q_1=1$	0	1	X	X

$$O = Q_1 I + \bar{Q}_1\bar{Q}_0\bar{I}$$

Build the circuit.

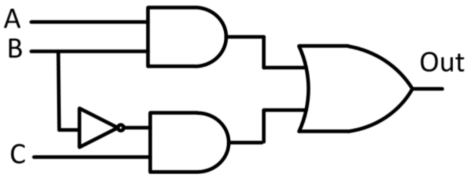


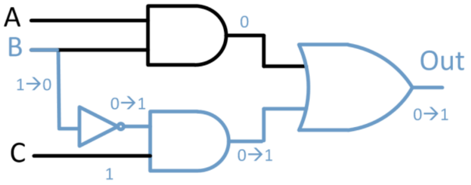
Question 2: Suppose each gate has propagation delay of 100ps and contamination delay of 60ps. What is the propagation delay and the contamination delay for the simplified combinational circuit implemented based on the given truth table.

A	B	C	Out
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

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

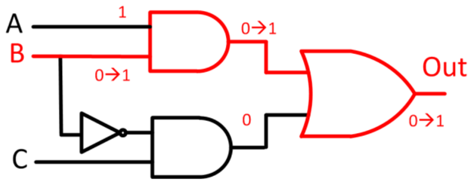
Out = $\overline{B}C + AB$





$$t_{pd} = t_{pd_{INV}} + t_{pd_{AND}} + t_{pd_{OR}}$$

$$t_{pd} = 3 \times 100 = \mathbf{300ps}$$



$$t_{cd} = t_{cd_{AND}} + t_{cd_{OR}}$$

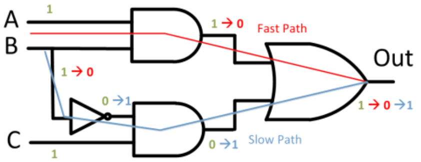
$$t_{cd} = 2 \times 60 = \mathbf{120ps}$$

Question 3: Does your result circuit in Question 2 have glitches? If yes, avoid all the glitches by modifying the circuit. If no, describe how did you achieve the glitch-free circuit?

The result circuit in Question 2 has glitch in the following transition:

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

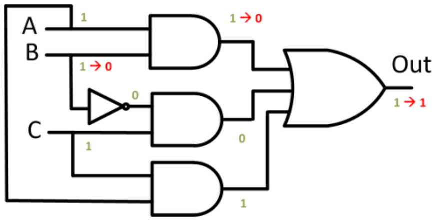
Out = $\overline{B}C + AB$



In order to avoid this glitch, we may add an extra group to the final formula:

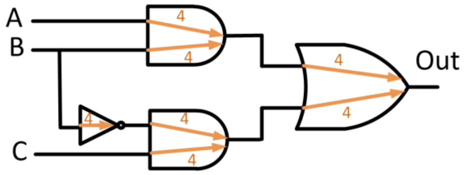
	BC=00	BC=01	BC=11	BC=10
A=0	0	1	0	0
A=1	0	1	1	1

Out = $\overline{B}C + AB + AC$



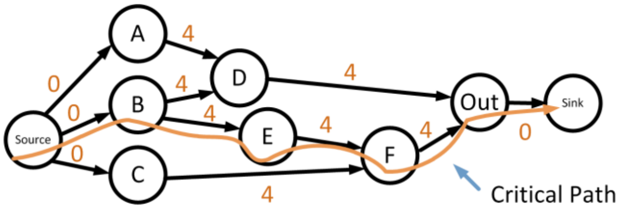
Question 4: Suppose the corresponding delay between two vertices is D=4, construct timing graph for the result circuit in Question 2. Then, suppose the clock cycle is T_c=10, compute Actual Arrival Time (AAT), Required Arrival Time (RAT), and Slack for each vertex of the timing graph. What is the topological critical path? Does the circuit meet timing constraints?

Vertex	AAT	RAT	Slack
A	0	2	2
B	0	-2	-2
C	0	2	2
D	4	6	2
E	4	2	-2
F	8	6	-2
Out	12	10	-2



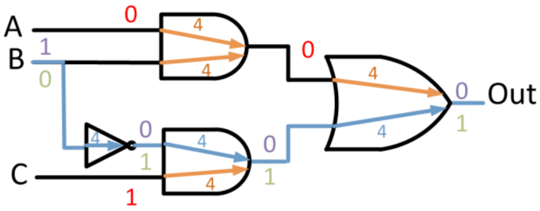
Critical path: B, E, F, Out

The design does not meet timing constraints.



Question 5: Does the result critical path in Question 4 meet the sensitization requirements? If yes, show the corresponding input vector for it to be statically sensitizable. If no, find the true critical path.

Yes. This is a statically sensitizable critical path by input vector AC=01.

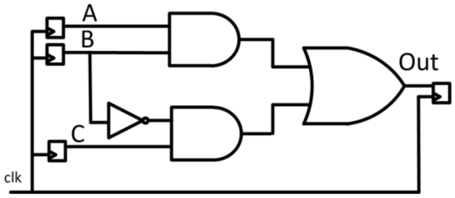


Question 6: Add one flip flop before each input and after the output of the result circuit in Question 2. Suppose all the added flip flops are synchronized by a single clock signal. Find the minimum clock period based on the following timing information. Does the design have any timing violation? If yes, modify the circuit to pass the timing constraints.

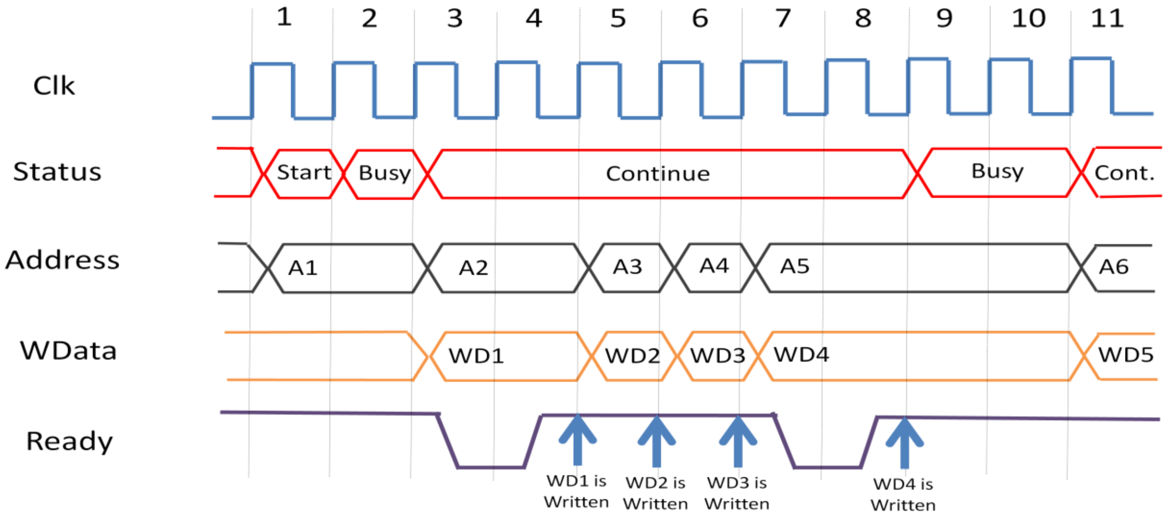
Contamination delay clock-to-q	$t_{cq} = 10ps$
Propagation delay clock-to-q	$t_{pq} = 30ps$
Setup time	$t_s = 40ps$
Hold time	$t_h = 60ps$
Clock skew	$t_{sk} = 5ps$

$t_{pd} < T_c - t_{pq} - t_s - t_{sk}$
$T_c > 300 + 30 + 40 + 5 = \mathbf{375ps}$

$t_{cd} > t_h - t_{cq} + t_{sk}$
$120 >? 60 - 10 + 5 = 55$ => **Pass**



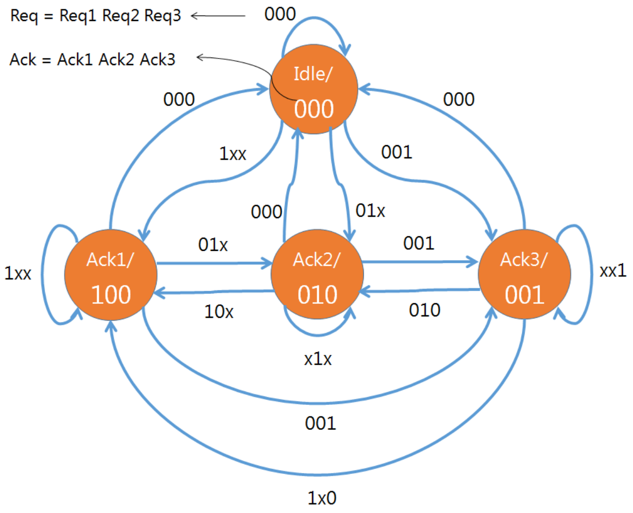
Question 2: When the bus master is busy to carry out its own internal tasks, the bus protocol requires the bus master to hold the address, control, and data values as long as it is busy. The following figure illustrates an example where the bus master becomes busy in clock cycles 2, 9 and 10 while writing data into a slave. Draw corresponding Address and Write Data (WData) rows. At which clock cycle each WData is written?



WD1, WD2, WD3, and WD4 are written at the clock cycles 5, 6, 7, and 9 respectively. The diagram does not tell us about the written time of WD5 and WD6.

Question 3: Draw bus arbitration table and state machine of an arbiter with three bus masters where bus master 1 has the highest priority followed by bus masters 2 and 3.

Req1	Req2	Req3	Ack1	Ack2	Ack3
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	1	0	0

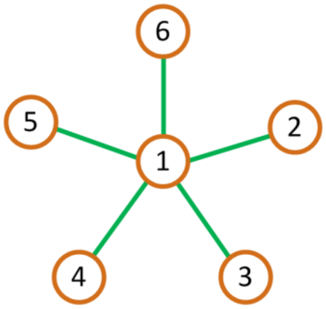


Question 4: Considering the following information, complete the given topology comparison table.

Reliability: Interconnection networks should be able to send messages through alternative paths when some faults are detected. Reliability of a topology can be seen as high, medium, and low based on the amount of faults that it can tolerate.

Planarity: If a topology can be implemented in two-dimensional plane, we call it planar; otherwise, it is not planar.

Star Topology: In a star topology, all the nodes are connected to a single management node called hub.



Star Topology

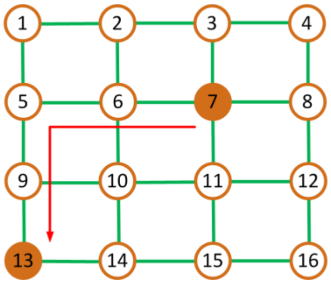
Topology	Cost	Speed	Contention	Reliability	Planarity
Bus	Lowest	Low	Highest	Low	Planar
P2P	Highest	Highest	Lowest	Highest	Not Planar
Ring	Low	Low	Medium	Low	Planar
Mesh	Low	Medium	Low	High	Planar
Torus	Low	Medium to High	Low	High	Not Planar
H-Tree	Low	Medium to High	Low	Medium	Planar
Star	Medium	High	Medium	Low	Planar

Note: In star topology, hub is costly for large systems. Also, hub is the hot-spot and may become saturated. In addition, hub is a single point of failure.

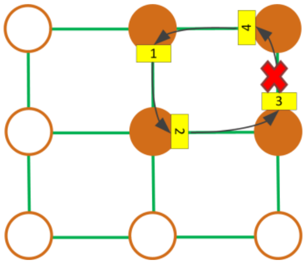
Question 5: A suggested DOR algorithm for mesh topology restricts all turns to the west direction. This means west direction should be taken first if needed in the proposed route.

(a) For the given topology, highlight the path that a message takes from source node 7 to destination node 13.

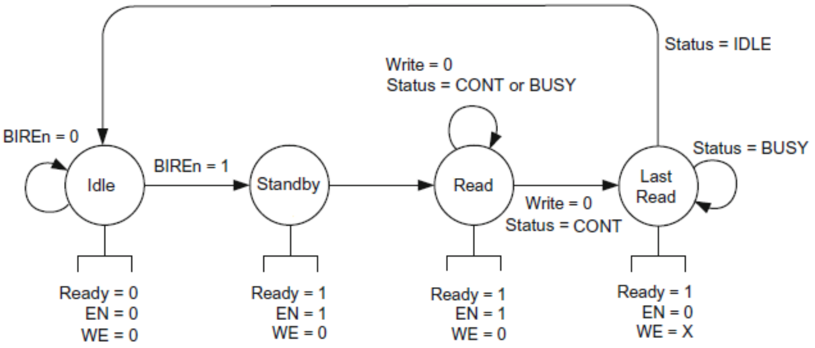
(b) Can the given deadlock situation happen in this system?

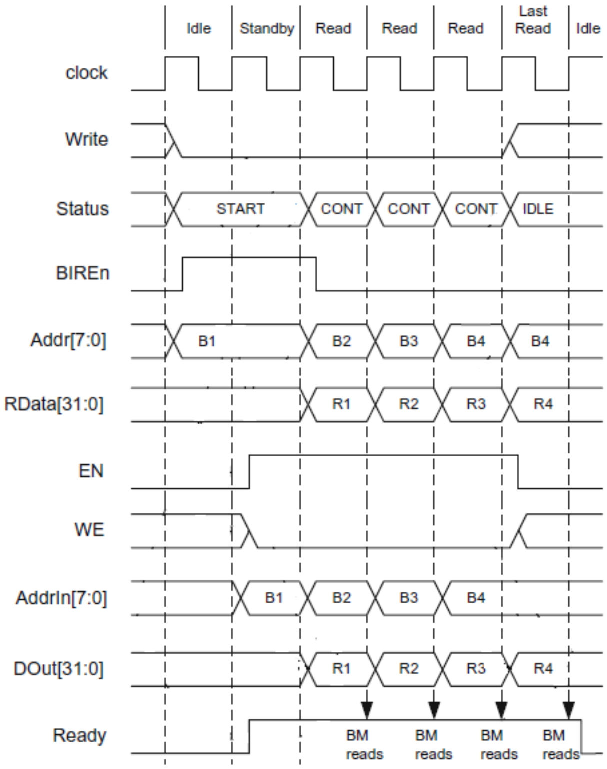


No. Since there should not be any turn to the west, message 3 cannot be waited for message 4 to proceed.



Question 2: Consider the given SRAM Bus interface in the lecture, draw SRAM bus interface state machine and timing diagram for **reading data** from four consecutive SRAM addresses.





Question 3: Suppose t_{PRE} = 4 cycles, t_{CAS} = 2 cycles, and t_{RAS} = 8 cycles, compute the total number of clock cycles, a bus master requires to write in a DRAM in each of the following scenarios:

(a) The bus master writes 4 messages in consecutive addresses on Bank 0, and 4 messages in another consecutive addresses of the same bank.

First Write = Precharge Code + t_{PRE} + Row Address + t_{CAS} + Write Period 1 = 1 + 4 + 1 + 2 + 4 = 12
Wait Time = t_{RAS} - t_{CAS} - Write Period 1 = 8 - 2 - 4 = 2

Second Write = Precharge Code + t_{PRE} + Row Address + t_{CAS} + Write Period 2 = 1 + 4 + 1 + 2 + 4 = 12
Total = First Write + Wait Time + Second Time = 12 + 2 + 12 = 26

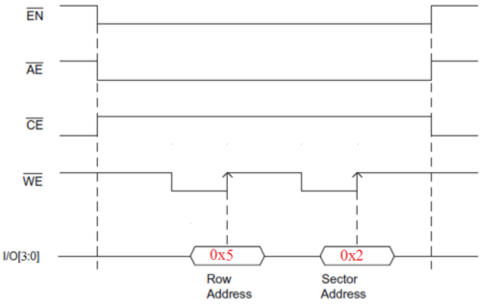
(b) The bus master writes 4 messages in consecutive addresses on Bank 0, another 4 messages in consecutive addresses on Bank1, and the last 4 messages in consecutive addresses on Bank 2.

Pre. Code	t_{PRE}	Row Ad.	t_{CAS}		Write 1								
1	4	1	2		4								
1	4	1	1	Pre. Code	t_{PRE}	Row Ad.	t_{CAS}		Write 2				
1	4	1	1	1	4	1	2		4				
1	4	1	1	1	4	1	1	Pre. Code	t_{PRE}	Row Ad.	t_{CAS}	Write 3	
1	4	1	1	1	4	1	1	1	4	1	2	4	=26

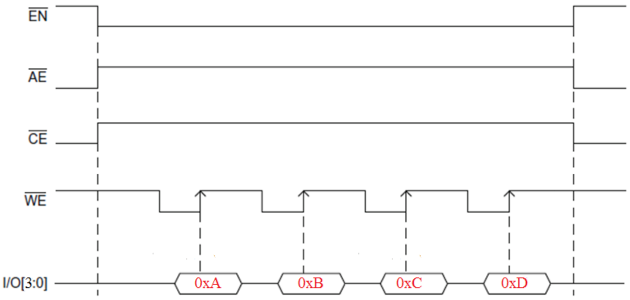
Question 4: An E^{2}PROM memory is organized in four sectors. There are eight rows in each sector but no pages. The existing data in this memory is shown below.

	Sector 0	Sector 1	Sector 2	Sector 3
	3 0	3 0	3 0	3 0
7	0x7	0xF	0x0	0x8
6	0x6	0xE	0x1	0x9
5	0x5	0xD	0x2	0xA
4	0x4	0xC	0x3	0xB
3	0x3	0xB	0x4	0xC
2	0x2	0xA	0x5	0xD
1	0x1	0x9	0x6	0xE
0	0x0	0x8	0x7	0xF

Draw a timing diagram to write 0xA, 0xB, 0xC, 0xD starting from the row address = 5 and the sector address = 2 in the following manner: the first data, 0xA, to the row address 5; the second data, 0xB, to the row address 6 and so forth. Draw the contents of the memory after the write sequence is complete.



Address Input Timing Diagram



Data Write Timing Diagram

	Sector 0	Sector 1	Sector 2	Sector 3
	3 0	3 0	3 0	3 0
7	0x7	0xF	0xC	0x8
6	0x6	0xE	0xB	0x9
5	0x5	0xD	0xA	0xA
4	0x4	0xC	0x3	0xB
3	0x3	0xB	0x4	0xC
2	0x2	0xA	0x5	0xD
1	0x1	0x9	0x6	0xE
0	0x0	0x8	0x7	0xD

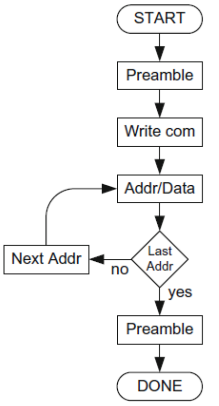
Contents of Memory after the Write Sequence

Question 5: A Flash memory block has an eight-bit address, and executes all reads and writes on an eight-bit bidirectional data bus. The Flash memory write sequence contains a preamble, a write command, and an address/data combination as shown in the flow chart below. Once the write command is issued, the address/data combination is generated continuously until the last write takes place. The sequence ends with the same preamble that starts the write.

START and DONE do not have any significance in timing diagrams other than that they indicate the start and the end of the sequence, respectively. The preamble and write commands are issued with the hexadecimal values shown in the truth table below.

The state of the Flash memory before any operation is shown below. The leftmost column in this figure shows the Flash memory address in hexadecimal.

Draw a timing diagram to write four data messages to the Flash memory as shown bellow. Also, draw the contents of the memory after the write sequence is complete.



Flowchart

COMMANDS	Address	Data
Preamble	FF	00
Write com	AA	FF

Commands

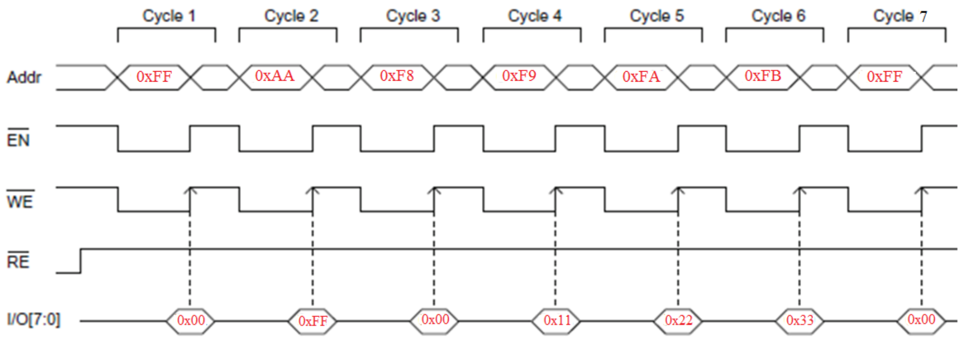
	7 0
	. .
F8	FF
F9	EE
FA	DD
FB	CC
FC	BB
FD	AA
FE	99
FF	88
	. .

State of Memory

Message #	Address	Data
1	F8	00
2	F9	11
3	FA	22
4	FB	33

Messages

Command	First Cycle		Second Cycle		Write Cycles		Last Cycle	
	Data	Address	Data	Address	Data	Address	Data	Address
Wrtie	00	FF	FF	AA	Write Data	Write Address	00	FF



Write Timing Diagram

	7 0
	. .
F8	00
F9	11
FA	22
FB	33
FC	BB
FD	AA
FE	99
FF	88
	. .

Contents of Memory after the Write Sequence