Written Exam
IE1205 Digital Design
2022-01-17, 14:00-18:00

General Information

Examiner: Ahmed Hemani

Responsible teacher for the exam: Ahmed Hemani

Exam text must be turned in when you hand in your solutions. No aids allowed except ruler and calculator.

Start each problem on a new piece of paper. Do not write on the backside of the papers.

Write your name and personal number on every paper that you hand in.

Organization

The exam consists of three parts and a total of 40 points. The exercises are NOT in order of difficulty.

Part 1 (10p): Digital Design Fundamentals

Part 2 (14p): Binary Arithmetic and Combinational Building Blocks

Part 3 (16p): Sequential Circuit and State Machines

Grade

To pass the exam requires **at least 20 points in total** AND **a minimum of 2 points of each part reached.**

Grades are given as follows:

0 – 19	20 – 23	24 – 27	28 – 31	32 – 34	35 – 40
F	E	D	C	B	A

Fx if (19 points AND minimal required points reached for all parts) OR (20 points or more AND minimal required points reached except for 1 part)

Cheat Sheet

Boolean Identities

Axiomer			
(1a)	$0 \cdot 0 = 0$	(1b)	$1 + 1 = 1$
(2a)	$1 \cdot 1 = 1$	(2b)	$0 + 0 = 0$
(3a)	$0 \cdot 1 = 1 \cdot 0 = 0$	(3b)	$1 + 0 = 0 + 1 = 1$
(4a)	If $x = 0$, then $\overline{x} = 1$	(4b)	If $x = 1$, then $\overline{x} = 0$

Räknelagar			
(5a)	$x \cdot 0 = 0$	(5b)	$x + 1 = 1$
(6a)	$x \cdot 1 = x$	(6b)	$x + 0 = x$
(7a)	$x \cdot x = x$	(7b)	$x + x = x$
(8a)	$x \cdot \overline{x} = 0$	(8b)	$x + \overline{x} = 1$
(9a)	$\overline{\overline{x}} = x$		

Räknelagar				
(10a)	$x \cdot y = y \cdot x$	(10b)	$x + y = y + x$	*kommutativ*
(11a)	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$	(11b)	$x + (y + z) = (x + y) + z$	*associativ*
(12a)	$x \cdot (y + z) = x \cdot y + x \cdot z$	(12b)	$x + y \cdot z = (x + y) \cdot (x + z)$	*distributiv*
(13a)	$x + x \cdot y = x$	(13b)	$x \cdot (x + y) = x$	*absorption*
(14a)	$x \cdot y + x \cdot \overline{y} = x$	(14b)	$(x + y) \cdot (x + \overline{y}) = x$	
(15a)	$\overline{x \cdot y} = \overline{x} + \overline{y}$	(15b)	$\overline{x + y} = \overline{x} \cdot \overline{y}$	*DeMorgan*
(16a)	$x + \overline{x} \cdot y = x + y$	(16b)	$x \cdot (\overline{x} + y) = x \cdot y$	
(17a)	$x \cdot y + y \cdot z + \overline{x} \cdot z$ $= x \cdot y + \overline{x} \cdot z$	(17b)	$(x + y) \cdot (y + z) \cdot (\overline{x} + z)$ $= (x + y) \cdot (\overline{x} + z)$	*consensus*

Area and Delay Costs of 2, 3 and 4 input NAND Gates

1. It is allowed to use 2, 3, 4-input NAND/NOR gate unless stated otherwise in the question.
2. Properties of NAND gates:
    a. 2-input NAND: delay=1.00T, area=1.0A
    b. 3-input NAND: delay=1.25T, area=1.5A
    c. 4-input NAND: delay=1.50T, area=2.0A
3. Delay of N input NAND gates in terms of
    a. 2 input NAND gates: 2.00 T$\cdot \lceil \log_2 (N) \rceil$ - T
    b. 3 input NAND gates: 2.25 T$\cdot \lceil \log_3 (N) \rceil$ - T
    c. 4 input NAND gates: 2.50 T$\cdot \lceil \log_4 (N) \rceil$ - T
    d. $\lceil \ \rceil$ is the round-up function e.g. $\lceil 2.3 \rceil = 3$ and $\lceil 2.7 \rceil = 3$

e. To compute "log" to the base 3 or 4 in terms of the natural logarithm, you can use these formulas: i) $\log_3(N) = \log(N)/\log(3)$ ii) $\log_4(N) = \log(N)/\log(4)$

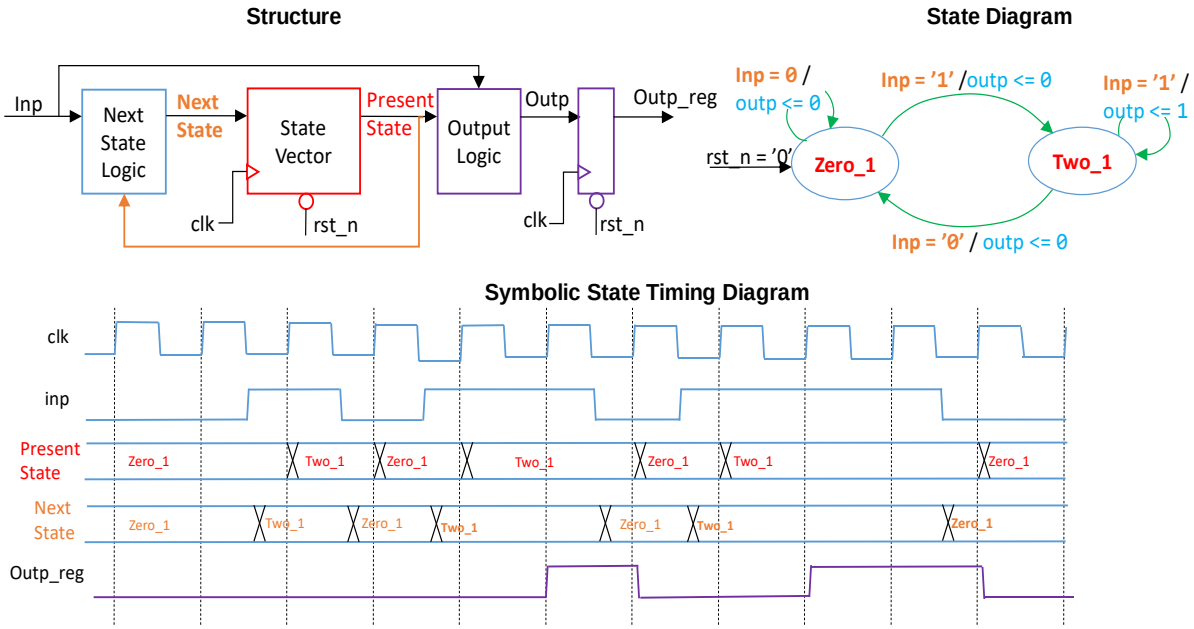
Excitation Tables of Latches and Flops

Inputs			Present State		Next State	
Clocked S R Latch (Positive Level Sensitive)						
Clk = 1	S = 0	R = 1	-	-	$Q^+ = 0$	$QB^+ = 1$
Clk = 1	S = 1	R = 0	-	-	$Q^+ = 1$	$QB^+ = 0$
Clk = 1	S = 0	R = 0	Q	QB	$Q^+ = Q$	$QB^+ = QB$
Clk = 1	S = 1	R = 1	-	-	$Q^+ = 1$	$QB^+ = 1$
Clk = 0	S = -	R = -	Q	QB	$Q^+ = Q$	$QB^+ = QB$
Clocked D Latch (Positive Clock Level Sensitive)						
Clk = 1		D = 1	-	-	$Q^+ = 1$	$QB^+ = 0$
Clk = 1		D = 0	-	-	$Q^+ = 0$	$QB^+ = 1$
Clk = 0		D = -	Q	QB	$Q^+ = Q$	$QB^+ = QB$
D Flip-Flop (Rising Edge Sensitive;)						
Rising edge of the Clk		D = 1	-	-	$Q^+ = 1$	$QB^+ = 0$
		D = 0	-	-	$Q^+ = 0$	$QB^+ = 1$
Rest of the Clk Period		D = -	Q	QB	$Q^+ = Q$	$QB^+ = QB$
S R Flip-Flop (Rising Edge Sensitive)						
Rising Edge of the Clk	S = 0	R = 1	-	-	$Q^+ = 0$	$QB^+ = 1$
	S = 1	R = 0	-	-	$Q^+ = 1$	$QB^+ = 0$
	S = 0	R = 0	-	-	$Q^+ = Q$	$QB^+ = QB$
	S = 1	R = 1	Q	QB	$Q^+ = 1$	$QB^+ = 1$
Rest of the Clk Period	S = -	R = -	Q	QB	$Q^+ = Q$	$QB^+ = QB$
J K Flip-Flop (Rising Edge Sensitive)						
Rising Edge of the Clk	J = 0	K = 1	-	-	$Q^+ = 0$	$QB^+ = 1$
	J = 1	K = 0	-	-	$Q^+ = 1$	$QB^+ = 0$
	J = 0	K = 0	Q	QB	$Q^+ = Q$	$QB^+ = QB$
	J = 1	K = 1	Q	QB	$Q^+ = QB$	$QB^+ = Q$
Rest of the Clk Period	J = -	K = -	Q	QB	$Q^+ = Q$	$QB^+ = QB$
T Flip-Flop (Rising Edge Sensitive)						
Rising edge of the Clk		T = 1	Q	QB	$Q^+ = QB$	$QB^+ = Q$
		T = 0	Q	QB	$Q^+ = Q$	$QB^+ = QB$
Rest of the Clk Period		T = -	Q	QB	$Q^+ = Q$	$QB^+ = QB$

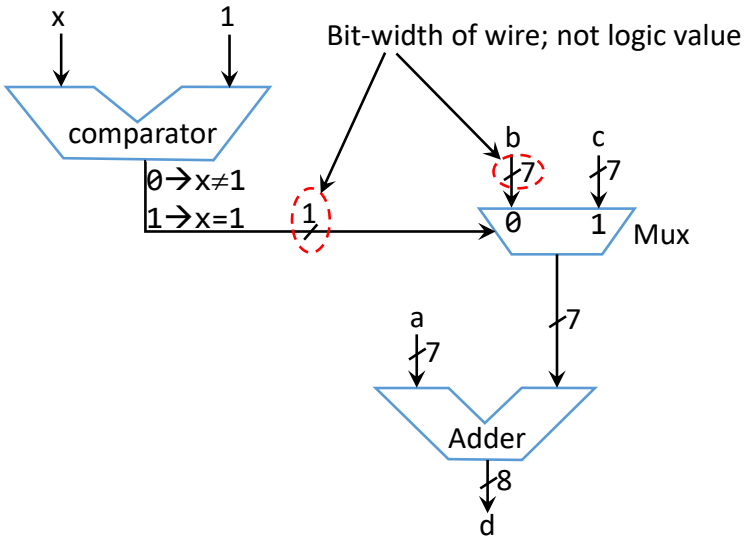
Style Guide of Symbolic Timing Diagram for FSM

As an example, we show how to draw a symbolic timing diagram for a registered Mealy FSM to detect two 1s in a row. The state diagram is shown in the figure below in which state *Zero_1* represents there is no 1 detected, *Two_1* represents there is at least one 1 detected, *Inp* is the input and *Outp* is the output.

When asked to draw a symbolic state diagram, you should draw *present states* and *next states* like the timing diagram shown below. Pay attention to the "present state" and "next state", they are both marked by symbolic states *Zero_0* or *Two_1*. Their transition points are marked by a cross-like divider.




```
If x = 1 Then
   d = a + c
Else
   d = a + b
End If
```



Part 1 (10p)

1. (2p)

Five friends are planning a trip in a group chat. Their message history is shown below:

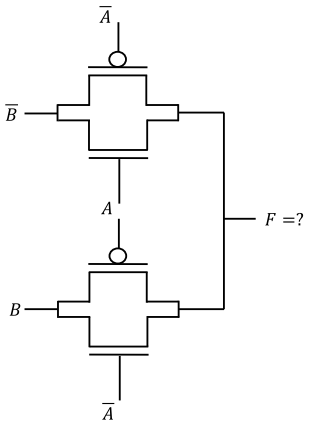
Alice: Hi everyone, do you want to go to the beach in the weekend?
Ben: Yes! But if both Elena and Carlos cannot be there, I'll skip the trip.
Carlos: I don't like to hangout with small group of friends, I'll go if at least 2 people besides me also go.
David: On the contrary, I don't like big groups. I'll go only if less than 4 people including me go. But if Ben is there, I'll definitely be there.
Elena: I'm not sure, I guess I'll join only if everybody else goes.

Write down the boolean expression for the requirements expressed by **Ben**, **Carlos**, **David** and **Elena**. You should use A, B, C, D, E to represent each person. For example: A means Alice will go, A' means Alice will not go. (0.5p for each expression.)

2. (2p)

What is the function of the following transmission gate circuit?

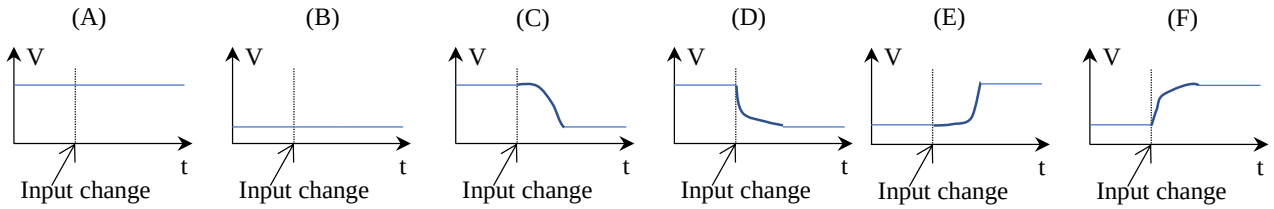
(A) AND
(B) OR
(C) XOR
(D) XNOR



3. (2p)

(1) (1p) Implement the function $z = c(a' + b)$ using only PMOS. Draw the schematic of the PMOS implementation, assuming that the inverted version of signals *a*, *b*, *c* is also available. Other electrical components except for NMOS (such as wires, resistors, capacitors and power sources) are also available.

(2) (1p) If you have a load capacitor attached to the output of the above circuit, how will the voltage of the load capacitor change when the inputs change from a=1, b=1, c=0 to a=0, b=1, c=1? Choose from the following options:



4. (4p)

A four-variable logic function that is equal to 1 if any three or all four of its variables are equal to 1 is called a majority function. Design a SOP circuit that implements this majority function. Assuming the inputs are “a, b, c and d”, output is “f”. You need to:

(1) (1p) Write the truth table.
(2) (1p) Find the minimum SOP form of output “f” using Karnaugh map. Expression is sufficient, no need to draw the logic schematic.
(3) (1p) Find the minimum POS form of output “f” using Karnaugh map. Expression is sufficient, no need to draw the logic schematic.
(4) (1p) Implement the majority function using only NAND gates based on the SOP form. You can use 2-input, 3-input and 4-input NAND gate if needed.

Part 2 (14p)

5. (2p)

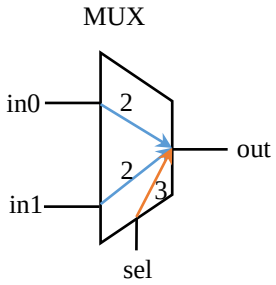
(1) (1p) Divide 6-bit unsigned integer `36` by 3-bit integer `3`. Show each step of the division.

(2) (1p) Multiply multiplicand `-0.25` with multiplier `-0.5` as two 2's complement fixed-point binary numbers in Q1.2 format. The result format should be Q2.4.

Note: In both problems above, you are not allowed to convert the negative number to a positive number. Nor can you change the role of multiplicand and multiplier.

6. (4p)

Giving that the propagation delay (from any operand to carry-out signal) of an N-bit ripple carry adder (RCA) is 4+2N, and the propagation delay of MUX is as follows,



(1) (2p) Draw the schematic diagram of an M*N-bit carry-select adder that uses M sections of N-bit RCA. You do not have to draw the internal details of N-bit RCA. A rectangle labeled as RCA with right inputs, outputs with their appropriate subscripts to indicate the slice indices, would be sufficient. The carry-in of the first, lowest significant, section M`0` is zero. The carry-in of the other, more significant sections, is connected to the carry-out of the previous section.

(2) (2p) Derive the symbolic formula for the critical path from inputs to the carry-out signal of such an M*N-bit carry-select adder.

7. (2p)

Use minimal number of 4-to-1 MUXes to implement the following function. HINT: You should not use the generic MUX structure to ensure you use the minimal number of Muxes. You can only use 4-to-1 MUXes:

$$f = ab + ac' + bc'd + c$$

8. (3p)

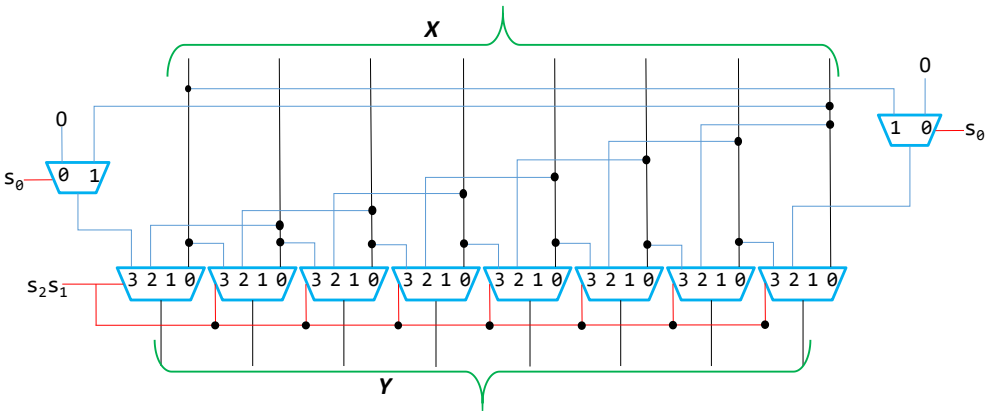
Draw the schematic of digital design using comparator (to check if two inputs are equal or not), adder, multiplexors and wires.
Assume A, B, C and D are 8-bit unsigned integers
Clearly mark the width of the wires and draw the schematic shown in the style guide

```
if A = B Then // "=" means "equal to"
  if A = C Then
    D <= A // "<=" means assigned to, i.e. A is assigned to D
  else
    D <= A + C
  end if
else
  D <= 0
end if
```

9. (3p)

The logic schematic shown below performs shifts and rotations for different combinations of s2, s1, s0. Each combination represents an operation.

s_2	s_1	s_0	Operation
0	0	-	OP_0
0	1	0	OP_1
0	1	1	OP_2
1	0	0	OP_3
1	0	1	OP_4
1	1	0	OP_5
1	1	1	OP_6

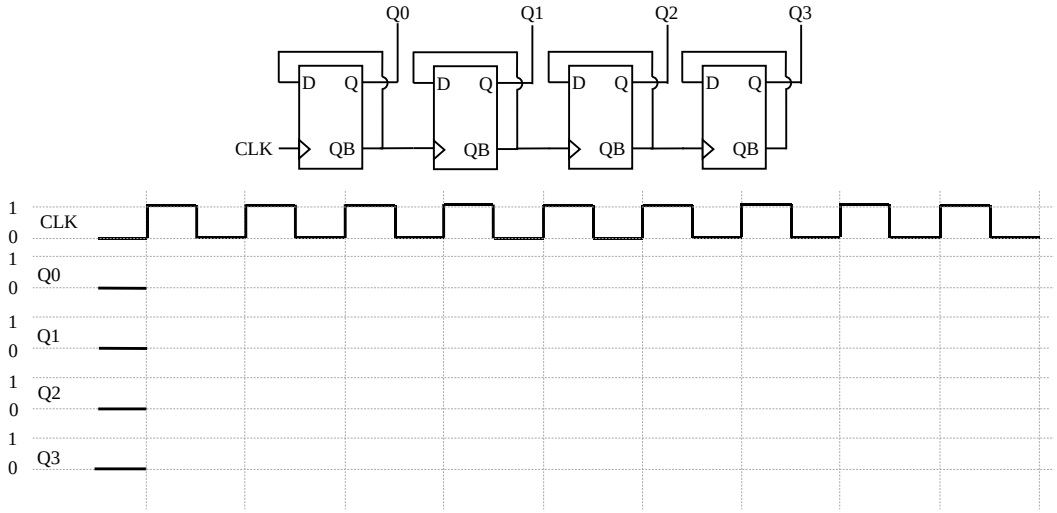


(1) (1p) Identify OP_0 to OP_6 as one of the following: No Operation, i.e. $Y = X$, Not used, i.e., invalid operation, Left Shift, Right Shift, Rotate Left or Rotate Right.
(2) (1p) Are the shifts logical or arithmetic?
(3) (1p) What will be the value of output Y in Hexadecimal notation for the following inputs:
   i. $s_2s_1s_0$ = 100 and X = H'AB
   ii. $s_2s_1s_0$ = 111 and X = H'AB

Part 3 (16p)

10. (2p)

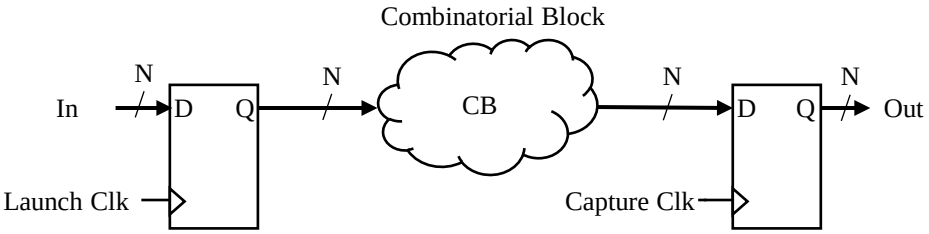
Complete the timing diagram for the following circuit. **Answer this question on the answer sheet!**



11. (4p)

For the following Flop based circuit, the clocks – clk_launch and clk_capture - are 100 MHz (10 ns clock period).

a. T_{setup} is 1 ns,
b. $T_{clk\text{-}2\text{-}Q}$ is 2 ns
c. T_{hold} (hold time) is 0.1 ns
d. T_{sp} (shortest path of the CB – Combinatorial Block) is 1 ns
e. T_{cp} (the longest - critical path) of the CB is 10 ns



Answer the questions:

(1) (1p) What should be the minimum skew between Capture Clk and Launch Clk to avoid setup violation?
(2) (2p) Will this skew induce hold violation? By how much. Show the calculation
(3) (1p) How will you avoid the hold violation?

Note: You need to answer question (1) and (2) quantitatively and show the calculation that justifies your answers.

12. (4p)

A **Moore** sequential circuit has one input (X) and one output (Z). The output goes to 1 when the input sequence 111 has occurred. At all other times, the output return to 0. For example:

X	0	1	0	1	1	1	0	1	1	0	0	1	1	1	1	0	1	0	0
Z	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0
Curr_state	S0																		
Next_state																			

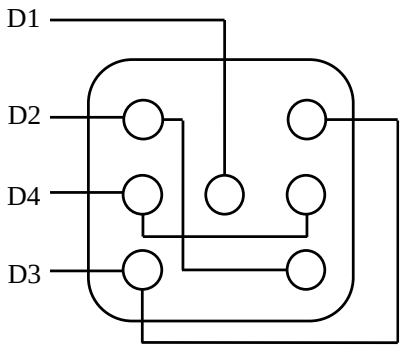
Each column in above table represents a clock cycle.
Answer the following question:

(1) (2p) Derive the state diagram. Using s0, s1, s2, … to mark your states
(2) (2p) Complete the above table by filling the missing curr_state and next_state row. Assuming, both X and Z initially are 0 and initially curr_state is in state s0.

13. (4p)

Design a counter that counts through six states and displays the result using the Dice display below. A "1" will light up the connected LED(s).

Dice	D1	D2	D3	D4
"1"	1	0	0	0
"2"	0	1	0	0
"3"	1	1	0	0
"4"	0	1	1	0
"5"	1	1	1	0
"6"	0	1	1	1

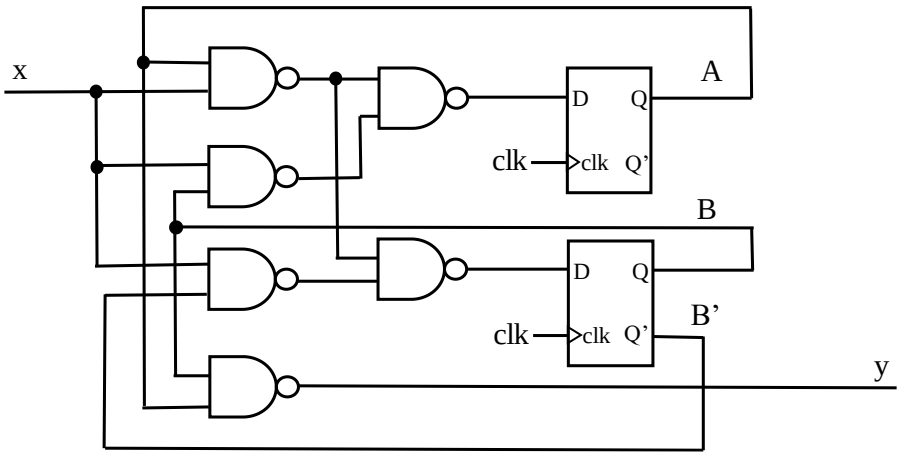


Design the counter using T flip-flops so that the states can be directly connected to the inputs of the Dice display without extra logic (D1 = Q1, D2 = Q2, D3 = Q3 and D4 = Q4). N.B: D1, D2 etc. are not inputs to the D flops; this design does not use D-Flops. They are terminals of the Dice. Design the state table to cycle through the states (Q1 Q2 Q3 Q4) in the order of the table above ("1, 2, 3, 4, 5, 6, 1, 2, ...").

(1) (1p) Derive the symbolic state transition table and then do state assignment such that state code represents the desired output (D1, D2, D3 and D4).
(2) (2p) Implement the next state logic using **T flip-flops**. Only equations needed. No need for logic schematics.
(3) (1p) Identify illegal states and what would be the next states from illegal states based on the expression derived in (2).

14. (2p)

An FSM implementation is shown below.



Answer the following questions:

(1) (1p) What is the next state logic expression?

(2) (1p) Write the state transition table of the FSM.