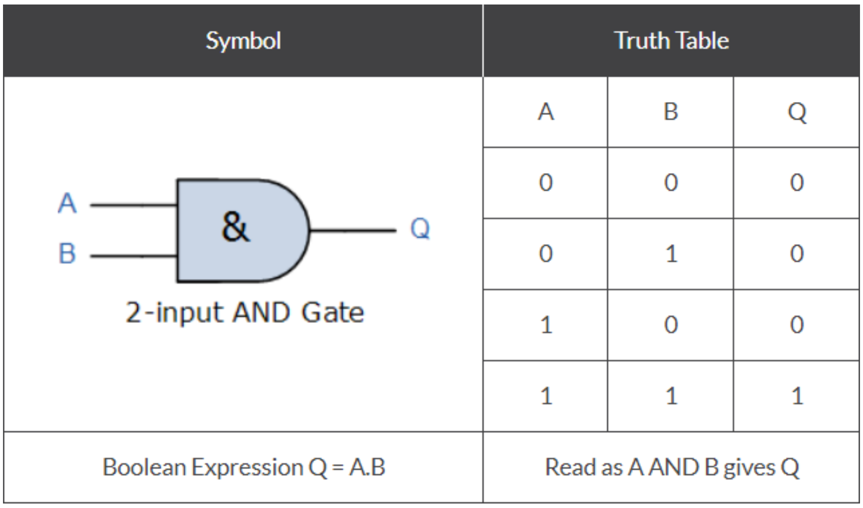
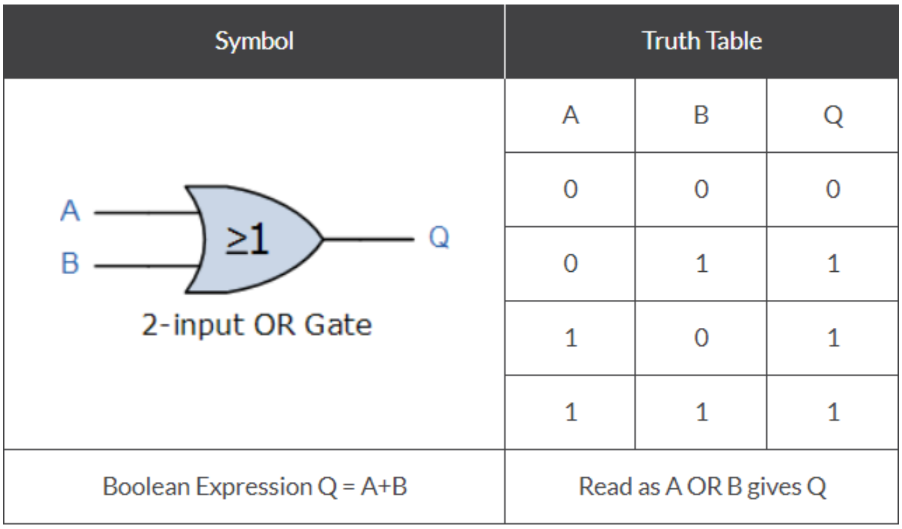
Assignment1

Provide answers to the following questions:

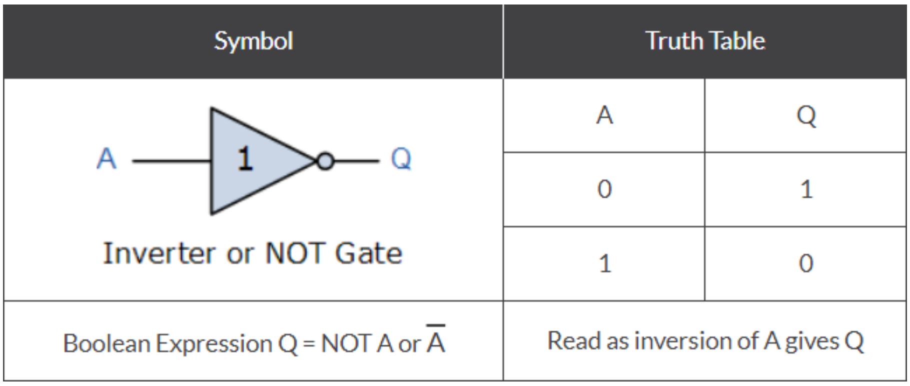
1. List the octal and hexadecimal numbers from 12 to 28. Using A, B, C and D for the last four digits, list the numbers from 12 to 28 in base 14.
   14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 26, 27, 30, 31, 32, 33, 34;
   C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C;
   C, D, 10, 11, 12, 13, 14, 15,16, 17, 18, 19, 1A, 1B, 1C, 1D, 20
2. What is the largest signed and unsigned binary number that can be expressed with 14 bits? What are the equivalent decimal and hexadecimal numbers?
   0111 1111 1111 11; +8191; 1FFF;
   1111 1111 1111 11; 16383; 3FFF
3. Convert the decimal number 240 to binary in two ways: (a) convert directly to binary; (b) convert first to hexadecimal and then from hexadecimal to binary. Which method is faster?
   (a) 1111 0000
   (b) F0; 1111 0000
   b should be faster
4. Describe the three basic types of logic calculation (write the truth table).
   AND, OR, NOT


A
B
&
Q
2-input AND Gate

Symbol	Truth Table		
	A	B	Q
	0	0	0
	0	1	0
	1	0	0
	1	1	1
Boolean Expression Q = A.B	Read as A AND B gives Q		



Symbol	Truth Table		
2-input OR Gate	A	B	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	1
Boolean Expression Q = A+B	Read as A OR B gives Q		



Symbol	Truth Table	
Inverter or NOT Gate	A	Q
	0	1
	1	0
Boolean Expression Q = NOT A or $\overline{A}$	Read as inversion of A gives Q	

5. Give the 1s’ and 2s’ complement representation of the following decimal numbers, using 32 bits.

   a. 14274836

   b. -27854048

   a. 0000 0000 1101 1001 1101 0001 0001 0100;

   0000 0000 1101 1001 1101 0001 0001 0100

   b. 1111 1110 0101 0110 1111 1011 0001 1111;

   1111 1110 0101 0110 1111 1011 0010 0000

   (In exam, we’ll clearly state it’s the code or representation. Typically it will be representation)

ASSIGNMENT2

Provide answers to the following questions:

1. Convert the following binary numbers to hexadecimal and to decimal:
   (a) 1.10101,
   (b) 1101.01
   How many times the decimal answer in (b) is larger than that in (a)? Explain why.

(a) 1.A8; 1.65625

(b) D.4; 13.25

8 times larger. (or 7, or some students may say a*8=b, also OK), Because (b) is shifted right for 3 bits in respect to (a), so its 2^3=8 times larger.

2. Do the following conversion problems:

   (a) Convert decimal 37.875 to binary.

   (b) Calculate the binary equivalent of 1/7 out to eight binary places. Then convert from binary to decimal. How much difference ((difference between 2 numbers)/(value of 1/7), in percentage, 2 decimal places) is the result to 1/7?

   (c) Convert the binary result in (b) into hexadecimal. Then convert the result to decimal. Is the answer the same?

(a) 100101.111

(b) 0.00100100; 0.140625; 1.56% or 0.00223214285 (at least 2 effective digits)

(c) 0.24; 0.001001; Yes

3. Represent the decimal number 7814 in (a) BCD, (b) excess-3 code, (c) 2421 code, and (d) 6311 code.

(a) 0111 1000 0001 0100

(b) 1010 1011 0100 0111

(c) 1101 1110 0001 0100

(d) 1001 1011 0001 0101

4. We can perform logical operations on strings of bits by considering each pair of corresponding bits separately (called bitwise operation). Given two eight-bit strings A = 10110101 and B = 00011100, evaluate the eight-bit result after the following logical operations: (a) AND (b) OR (c) XOR (d) NOT A (e) NOT B (f) NAND (g) NOR (represent the results in hexadecimal)

(a) 0x14 (OK without 0x)

(b) 0xBD

(c) 0xA9

(d) 0x4A

(e) 0xE3

(f) 0xEB

(g) 0x42

5. Simplify the following Boolean expressions to a minimum or the indicated number of literals:

   a. (a + b + c’)(a’b’+ c)

   b. a’b’c + ab’c + abc + a’bc

   c. (a + c)(a’+ b + c)(a’+ b’+ c)

   d. A’BD’ + ABC’D’+ ABCD’ to two literals

Need to show their process. E.g. Karnaugh map etc.

a. a’b’c’+bc+ac

b. c

c. c

d. BD’

6. Given the Boolean functions F1 and F2 , show that
   a. The Boolean function E = F1 + F2 contains the sum of the minterms of F1 and F2
   b. The Boolean function G = F1 · F2 contains only the minterms that are common to F1 and F2 .

a. $\mathrm{F1}+\mathrm{F2}=\Sigma m1_i+\Sigma m2_i=\Sigma(m1_i+\Sigma m2_i)$

b. $\mathrm{F1F2}=\Sigma m_i \cdot \Sigma m_j=\begin{cases}0\ (i \neq j)\\ 1\ (i=j)\end{cases}$

Or, if the explanation is reasonable, it’s also ok.

7. Convert each of the following to the other canonical form.
   a. F(x, y, z) = ∑(1, 3, 5,7)
   b. F(A, B, C, D) = ∏(3, 5, 8, 11, 13, 15)

a. F(x, y, z) = ∏ (0,2,4,6)

b. F(A, B, C, D) = ∑ (0,1,2,3,6,7,9,10,12,14)

8. Write the following Boolean expressions in:
   a. (b + d)(a’+ b’+ c)(a + c) SOP form
   b. a’b + a’c’+ bc POS form

a. ab’d+bc+cd

b. (a’+b’+c)(a+b+c’)

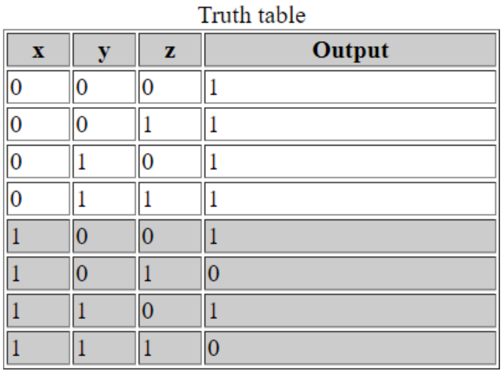
9. Determine whether the following Boolean equation is true or false.
   a. y’z’ + yz’ + x’z = x’+ xz
   b. x’y’+ xz’+ yz = y’z’ + xy + x’z

Truth table or by simplifying / converting, all OK.

a. false

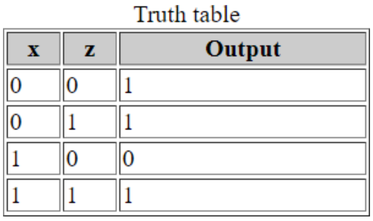
y’z’ + yz’ + x’z=(y+y’)z’+x’z=z’+x’z=x’+z’

x’+xz=x’+z



Truth table

x	y	z	Output
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



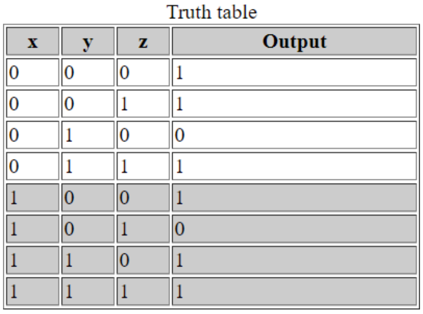
Truth table

x	z	Output
0	0	1
0	1	1
1	0	0
1	1	1

b. True

x’y’+ xz’+ yz = (x+y’+z)(x’+y+z’)

y’z’ + xy + x’z = (x+y’+z)(x’+y+z’)

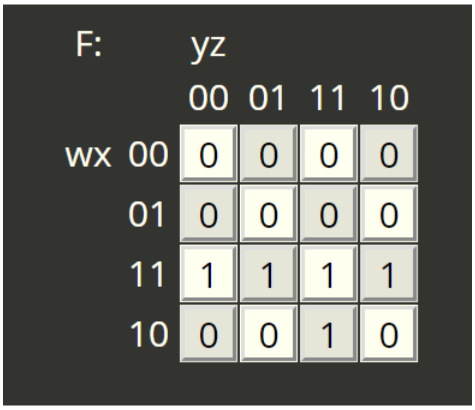


Truth table

x	y	z	Output
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

10. Simplify the following Boolean functions, using Karnaugh maps:
    a. F (w, x, y, z) = ∑(11, 12, 13, 14, 15)
    b. F (w, x, y, z) = ∑(8, 10, 12, 13, 14)

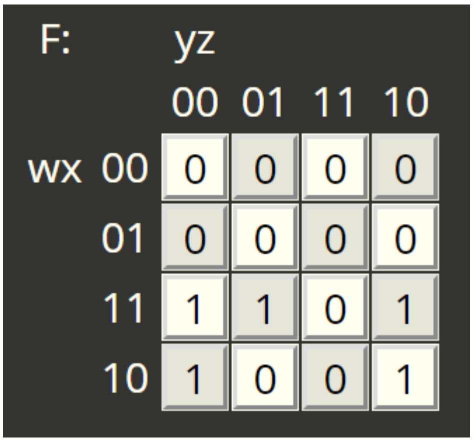
a. F=wyz+wx



F: yz

	00	01	11	10
wx 00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	0

b. F=wxy’+wz’

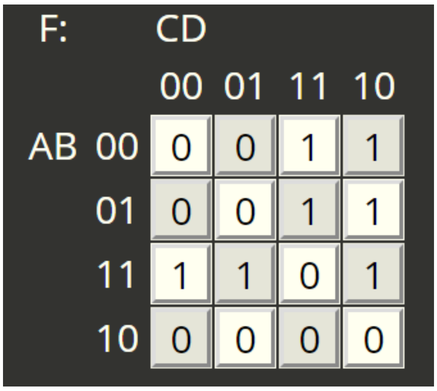


F: yz

	00	01	11	10
wx 00	0	0	0	0
01	0	0	0	0
11	1	1	0	1
10	1	0	0	1

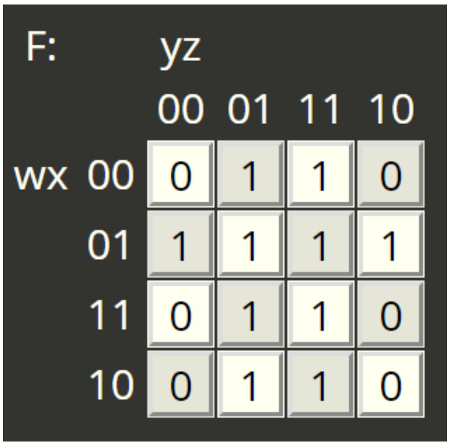
11. Simplify the following Boolean functions and expressions, using four-variable maps:
    a. F (A, B, C, D) = ∑ (2, 3, 6, 7, 12, 13, 14)
    b. F (w, x, y, z) = ∑ (1, 3, 4, 5, 6, 7, 9, 11, 13, 15)
    c. A’BCD + ABC + CD + B’D
    d. A’B’C’D’ + BC’D + A’C’D + A’BCD + ACD

a. F = ABC' + BCD' + A'C



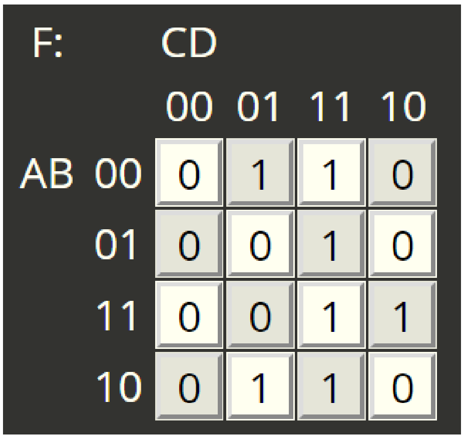
F:		CD			
		00	01	11	10
AB	00	0	0	1	1
	01	0	0	1	1
	11	1	1	0	1
	10	0	0	0	0

b. F = w'x + z



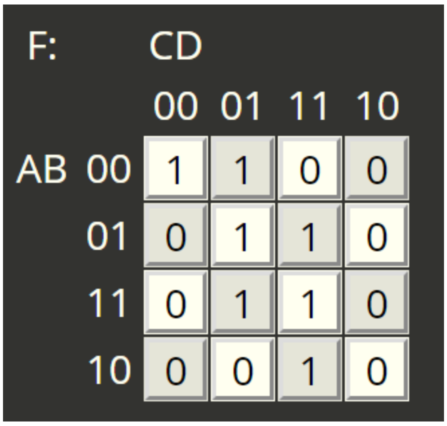
F:		yz			
		00	01	11	10
wx	00	0	1	1	0
	01	1	1	1	1
	11	0	1	1	0
	10	0	1	1	0

c. F=ABC+B’D+CD



F:		CD			
		00	01	11	10
AB	00	0	1	1	0
	01	0	0	1	0
	11	0	0	1	1
	10	0	1	1	0

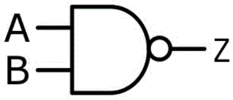
d. F=A’B’C’+ACD+BD



F:	CD				
		00	01	11	10
AB	00	1	1	0	0
	01	0	1	1	0
	11	0	1	1	0
	10	0	0	1	0

12. Implement the following logical functions with two-level NAND gate circuits. Write down the simplification process.
    a. F(A, B, C, D) = AD + BC’D+ ABC + A’BC’D
    b. F(A, B, C, D) = A’B’C’D + CD’ + AC’D
    c. F(A, B, C, D)= (A’+ C’+ D’)(A’+ C’)(C’+ D’)
    d. F(A, B, C, D) = A’+ AB + B’C + ACD

We don’t care the simplification process, if the figure is correct then get full mark. (⊼ means NAND)



NAND gate should be like this. NOT gate is also acceptable. All other gates are forbidden.

Some may use the inversed-or, also OK.

a. (A ⊼ B ⊼ C) ⊼ (A ⊼ D) ⊼ (B ⊼ C’ ⊼ D)

b. (A ⊼ C’ ⊼ D) ⊼ (A’ ⊼ B’ ⊼ D’) ⊼ (C ⊼ D’)

c. (A’ ⊼ D’) ⊼ C

d. A ⊼ B’ ⊼ C’

The answer is not unique.

ASSIGNMENT3

Provide answers to the following questions:

1. Draw the multiple-level nor circuit for the following expression: F = AB(D + C) + (BC’ + DE’) + BD’

   2-Level:

   Convert to POS form first: (students need to write internal transformation process)

F = (A + C’ + D’ + E’)(B + D)(B + E’)

Multi-level:

Or just use the direct way (represent and/or in NOR)

2. Implement the following Boolean function F, using the two-level forms of logic

(a) AND-OR, (b) OR-NAND, (c) NOR-OR, (d) NAND-NAND, (e) OR-AND, (f) NOR-NOR, and (g) NAND-AND:

F(A, B, C, D) = ∑ (1, 4, 5, 8, 9, 10, 11, 13, 15)

Please note that you only need to write the logic equation, no drawing needed.

F = A'BC' + AB' + C'D + AD

(a) AND-OR: SOP form. F(A, B, C, D) = ∑ (1, 4, 5, 8, 9, 10, 11, 13, 15)

Or

AB' + AD + A'BC' + C'D

(b) OR-NAND: Using POS of F’ and OR-AND-INVERT. F = (F’)’ = [(A'+B)(A'+D')(A+B'+C)(C+D')]’

F = (A'+B) nand (A'+D') nand (A+B'+C) nand (C+D')

(c) NOR-OR: Same as above.

F = (A' nor B)+(A' nor D')+(A nor B' nor C)+(C nor D')

(d) NAND-NAND: (A nand B') nand (A nand D) nand (A' nand B nand C') nand (C' nand D)

(e) OR-AND: POS form. F(A, B, C, D) = Π (0, 2, 3, 6, 7, 12, 14)

Or

(A' + B' + D)(A + B + D)(A + C')

(f) NOR-NOR: (A' nor B' nor D) nor (A nor B nor D) nor (A nor C')

(g) NAND-AND: Using SOP form of F’ and AND-OR-INVERT.

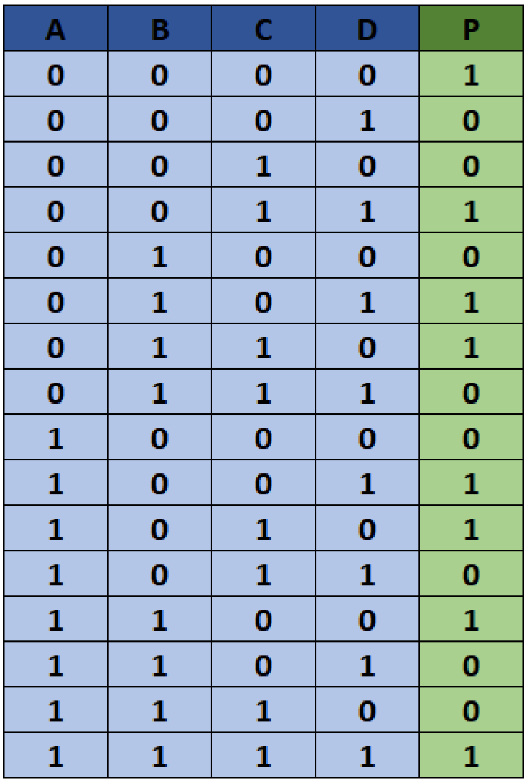
F = (F’)’ = (A'B'D' + ABD' + A'C)’

F = (A' nand B' nand D')(A nand B nand D')(A' nand C)

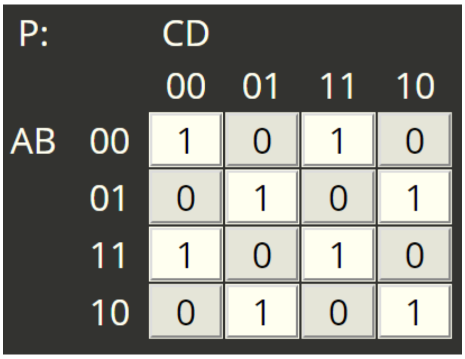
3. Derive the circuits for a four-bit parity generator and three-bit parity checker using an odd parity bit (3bit data+1bit parity). Write the truth table, simplify your logic equation and draw circuit diagram.

Generator:

Write truth table first:



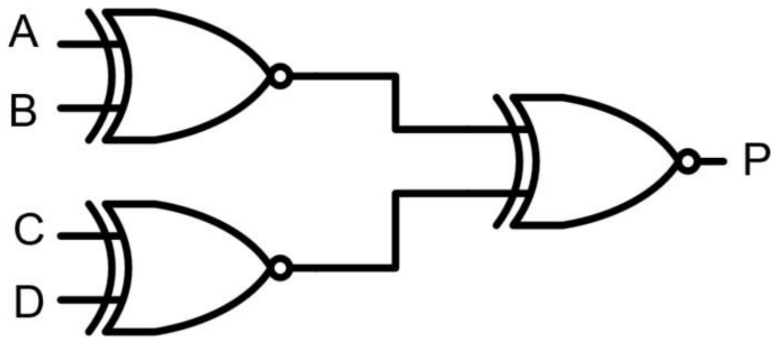
A	B	C	D	P
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1



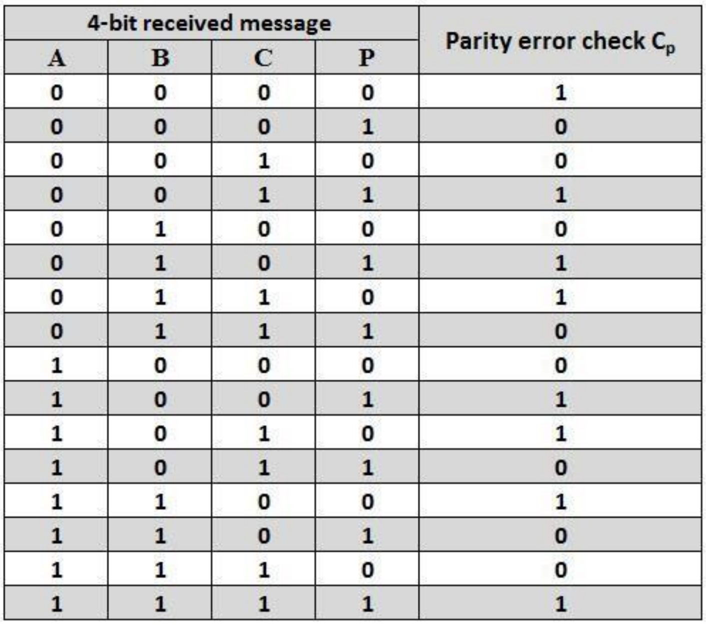
P:		CD			
		00	01	11	10
AB	00	1	0	1	0
	01	0	1	0	1
	11	1	0	1	0
	10	0	1	0	1

Then write K-map (not required)

P = (A xnor B) xnor (C xnor D)

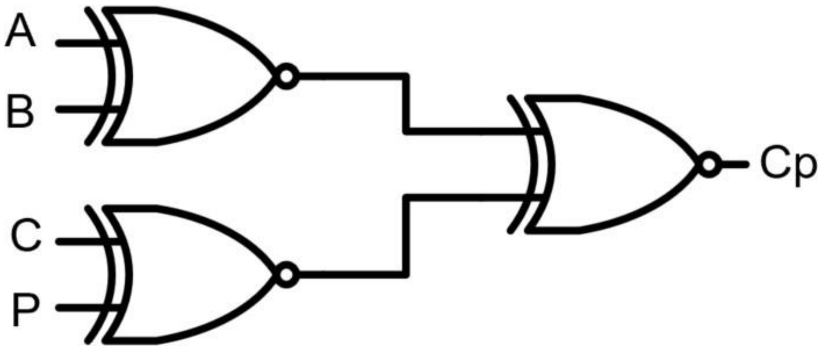


Checker:



4-bit received message				Parity error check C_p
A	B	C	P	
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Cp = (A xnor B) xnor (C xnor P)



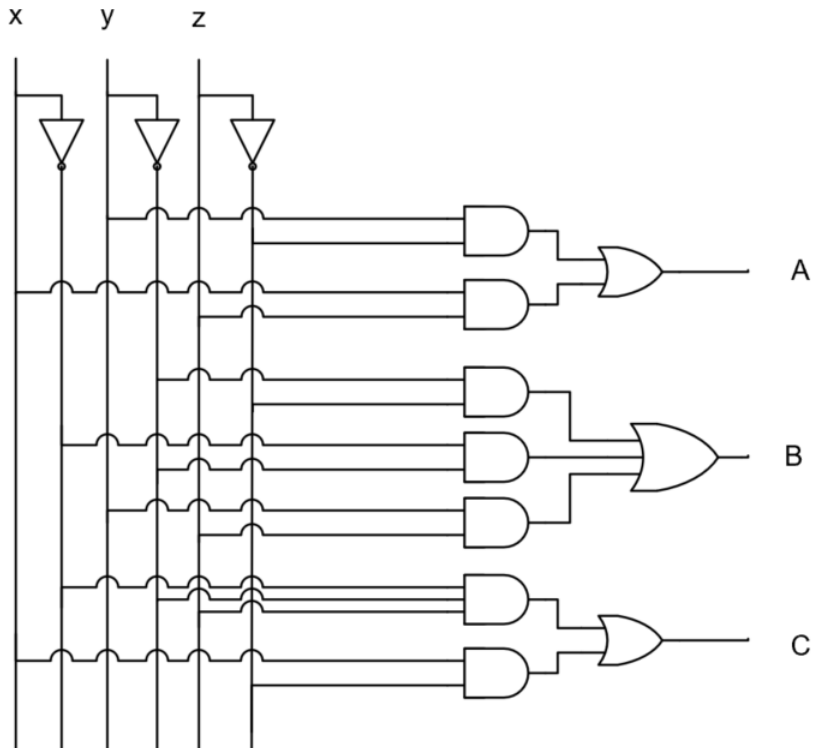
4. Design a combinational circuit with three inputs, x , y , and z , and three outputs, A, B , and C. When the binary input is 3, 4, 5, 6, or 7, the binary output is one less than the input. When the binary input is 0, 1, or 2, the binary output is two greater than the input. Write the logic formula and draw circuit diagram.

x	y	z	A	B	C
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0

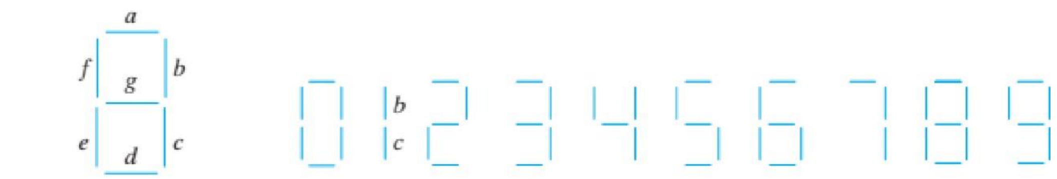
A = yz' + xz

B = y'z' + x'y' + yz

C = x'y'z + xz'



5. An ABCD-to-seven-segment decoder is a combinational circuit that converts a decimal digit in BCD to an appropriate code for the selection of segments in an indicator used to display the decimal digit in a familiar form. The seven outputs of the decoder (a, b, c, d, e, f, g) select the corresponding segments in the display, as shown in figure (a). The numeric display chosen to represent the decimal digit is shown in (b). Using a truth table and/or Karnaugh maps, design the BCD-to-seven-segment decoder using a minimum number of gates. All the invalid input should display nothing.



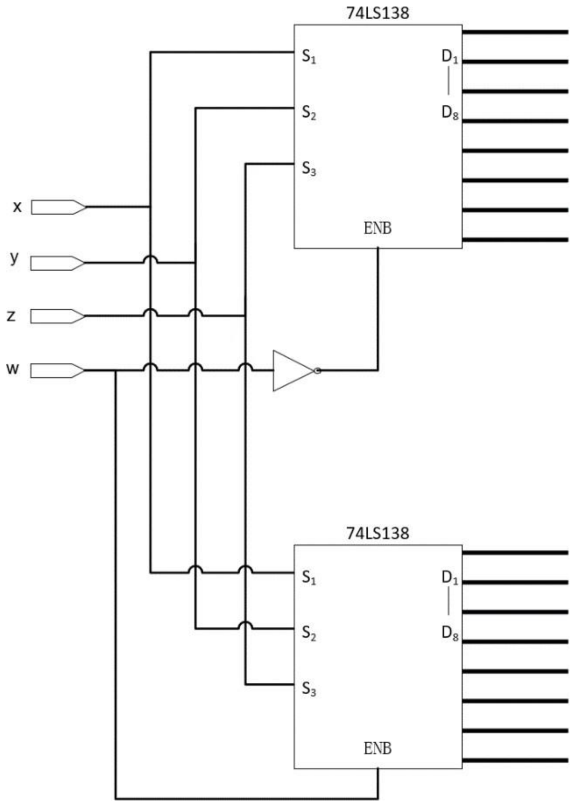
(a) Segment designation

(b) Numerical designation for display

Hint: You may use 74LS138 decoder.

Truth table:

w	x	y	z	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	1
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	X	0	0	0	0	0	0	0
1	1	X	X	0	0	0	0	0	0	0



$a(w,x,y,z)=\Sigma(0,2,3,5,6,7,8,9)$

$b(w,x,y,z)=\Sigma(0,1,2,3,4,7,8,9)$

$c(w,x,y,z)=\Sigma(0,1,3,4,5,6,7,8,9)$

$d(w,x,y,z)=\Sigma(0,2,3,5,6,8,9)$

$e(w,x,y,z)=\Sigma(0,2,6,8)$

$f(w,x,y,z)=\Sigma(0,4,5,6,8,9)$

$g(w,x,y,z)=\Sigma(2,3,4,5,6,7,8,9)$

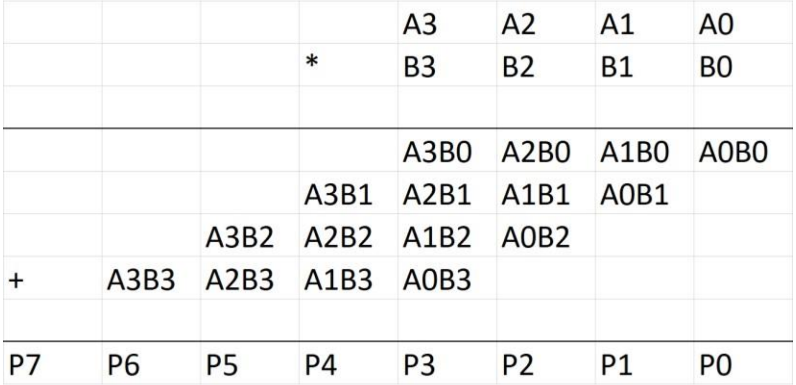
Use 2 74LS138 to build a 4-16 decoder

Then connect based on the minterm sum formula above.

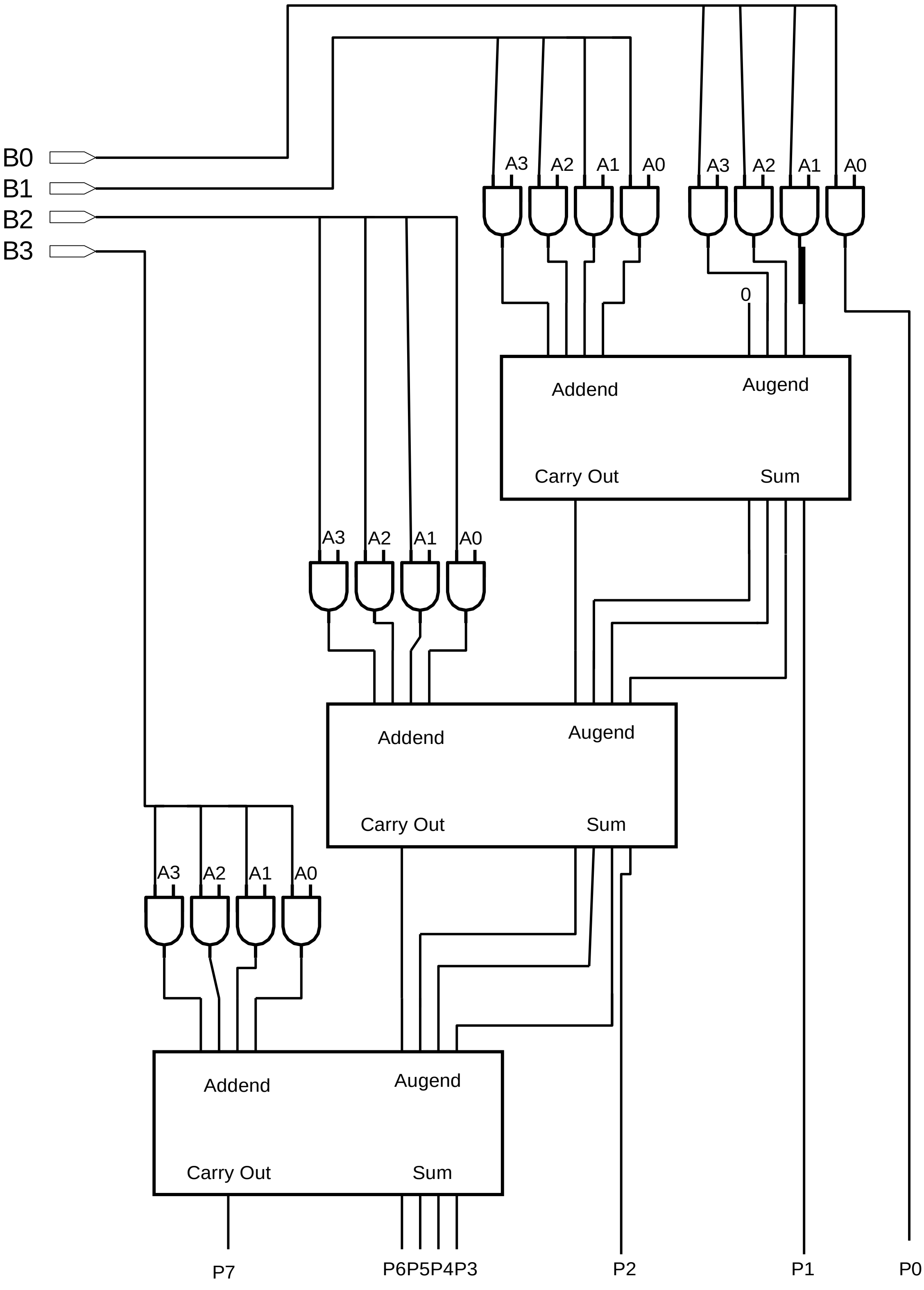
6. For a binary multiplier that multiplies two unsigned four-bit numbers

a. Using AND gates and binary adders, design the circuit. Briefly describe your design and list the logic formulas. Draw the circuit diagram. For adders, you can just use a block diagram with ports to represent them.

$A_3A_2A_1A_0*B_3B_2B_1B_0 = P_7P_6P_5P_4P_3P_2P_1P_0$



				A3	A2	A1	A0
			*	B3	B2	B1	B0
				A3B0	A2B0	A1B0	A0B0
			A3B1	A2B1	A1B1	A0B1	
		A3B2	A2B2	A1B2	A0B2		
+	A3B3	A2B3	A1B3	A0B3			
P7	P6	P5	P4	P3	P2	P1	P0



7. Implement the following Boolean function with a multiplexer

   a. F(A, B, C, D) =∑ (0, 2, 5, 8, 10, 14)

   b. F(A, B, C, D) = Π (2, 6, 11)

a. Just set D0, 2, 5, 8, 10, 14 to 1, others to 0.

b. F(A, B, C, D) = Π (2, 6, 11) =$\sum$ (0, 1, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,15)

Set D0, 1, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,15 to 1, others to 0.

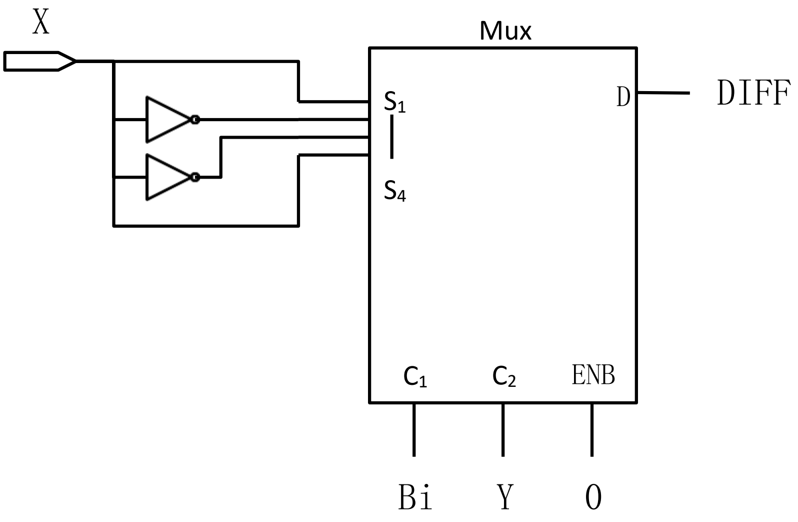
8. Implement a full subtractor with two 4x1 multiplexers. Write down your process.

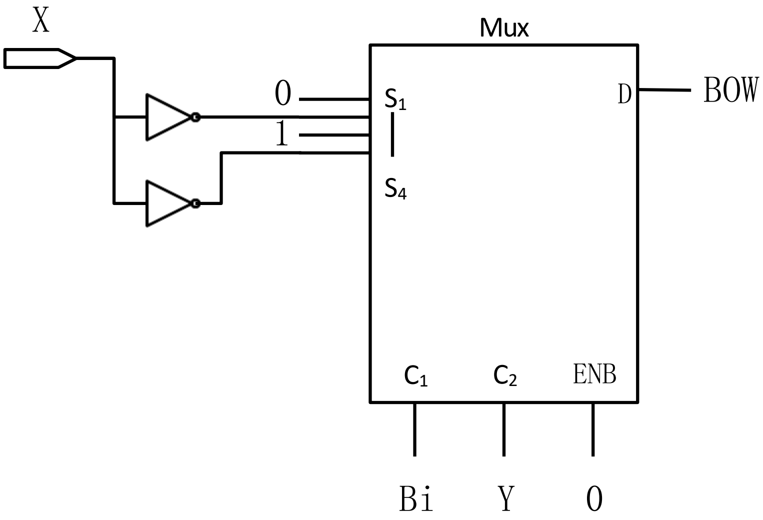
Truth table:

X	Y	Bi	BOW	DIFF
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

BOW =$\sum$ (1, 2, 3, 7)

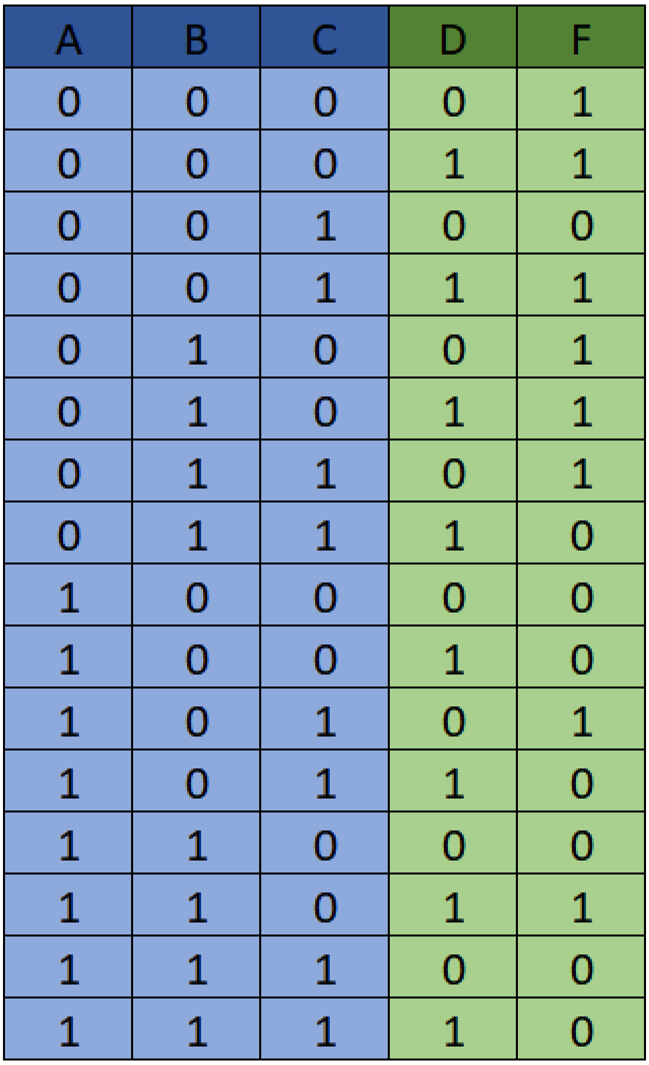
DIFF =$\sum$ (1, 2, 4, 7)





9. An 8x1 multiplexer has inputs A, B, and C connected to the selection inputs
   S0, S1, and S2, respectively. The data inputs I0 through I7 are as follows:
   I0 = I2 = 1; I4 = I7 = 0; I3 = I5 = D'; I1 = I6 = D. Determine the Boolean function that the multiplexer implements. Write your process.

   Write down the truth table:

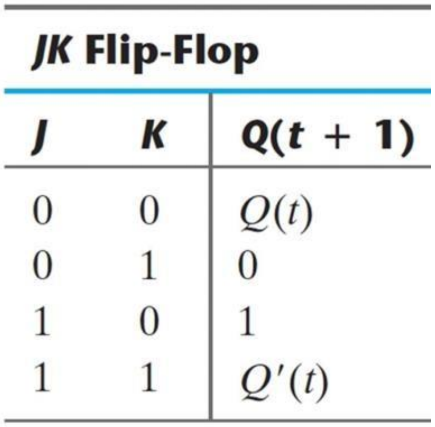


A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

$F(A, B, C, D) = \sum (0,1,3,4,5,6,10,13)$

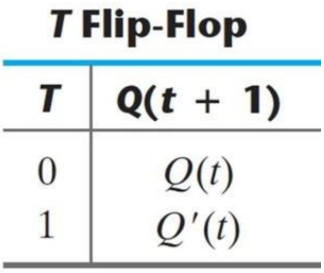
ASSIGNMENT4

1. Write down the characteristic tables of both JK flip-flop and T flip-flop. Based on the truth table, write down their characteristic equations. (Use character T, J, K, Q, Q_{t+1})



JK Flip-Flop

J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$Q'(t)$



T Flip-Flop

T	$Q(t+1)$
0	$Q(t)$
1	$Q'(t)$

JK: $Q_{t+1} = JQ' + K'Q$

T: $Q_{t+1} = T'Q' + TQ$

2. A sequential circuit has two JK flip-flops A and B, two inputs x and y, and one output z. The input and output equations are:

$$J_A = Bx + B'y' \qquad K_A = B'xy'$$

$$J_B = A'x \qquad K_B = A + xy'$$

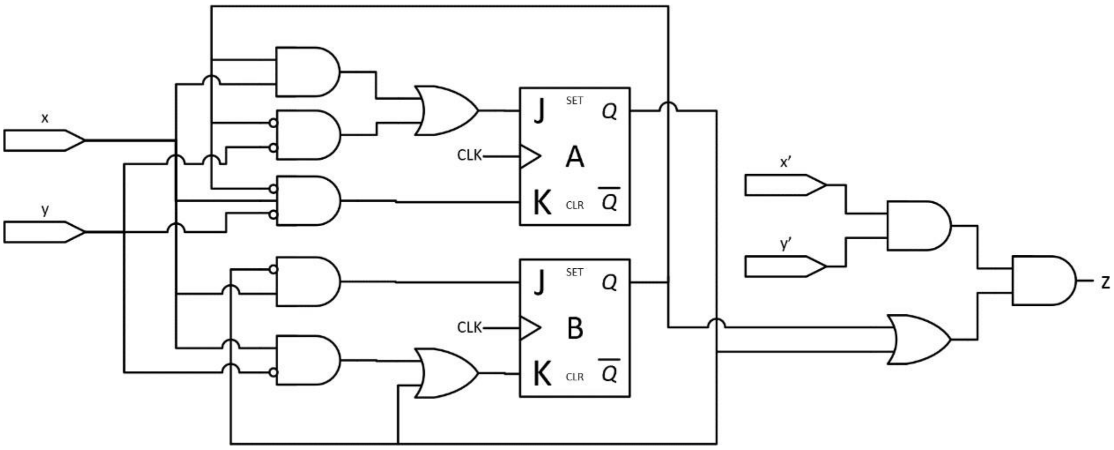
$$z = Ax'y' + Bx'y'$$

(a) Draw the circuit diagram.

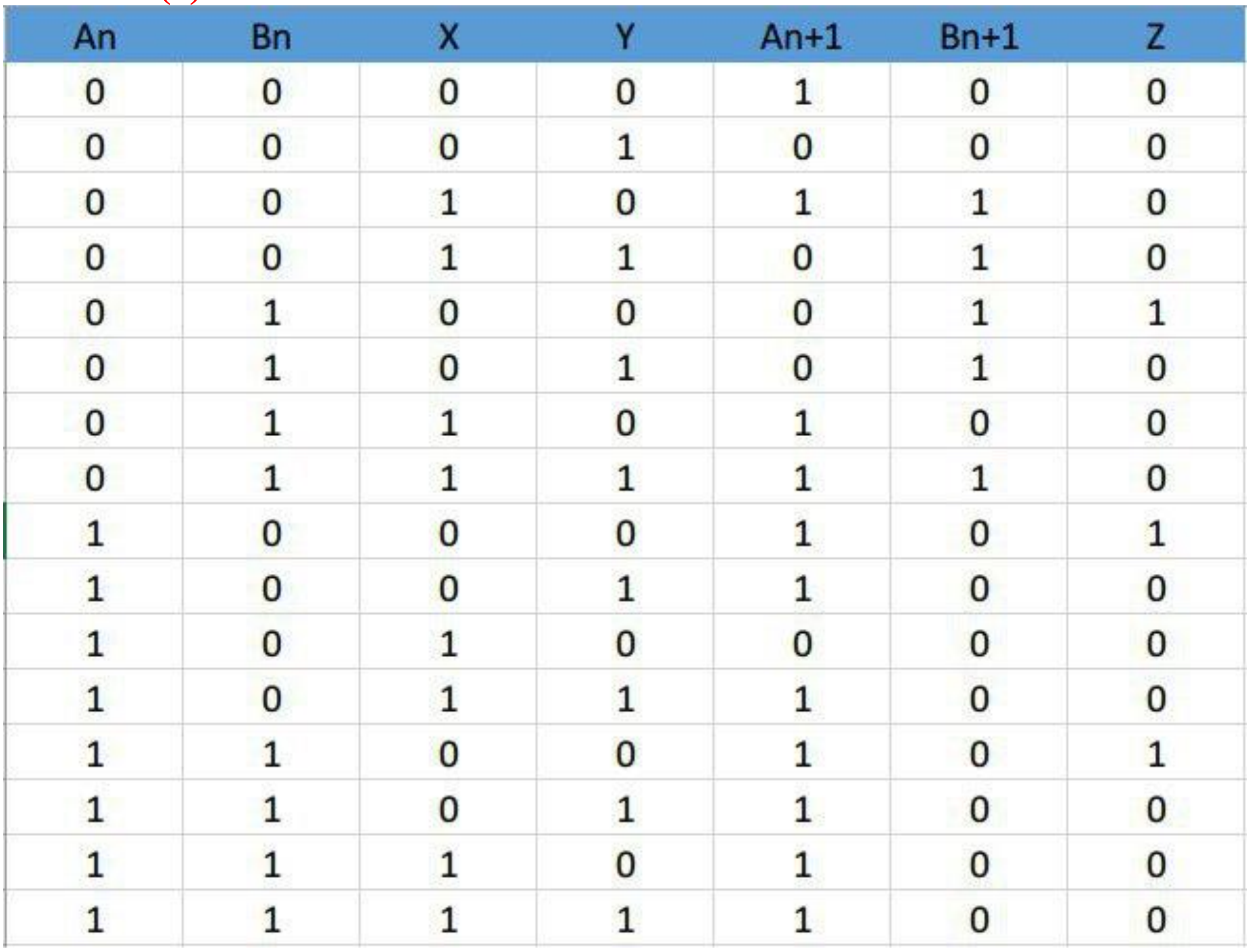
(b) Write down the state table.

(c) Derive the state equations for A and B.

(a)



(b)



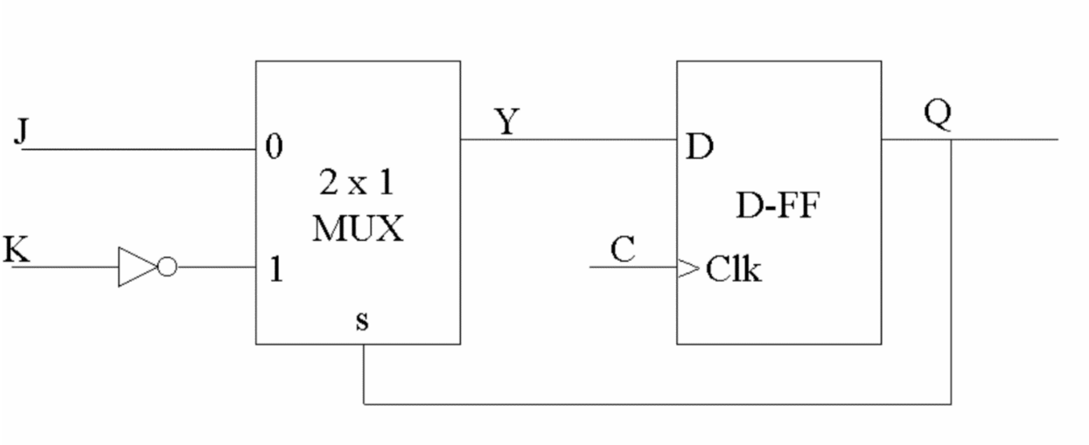
An	Bn	X	Y	An+1	Bn+1	Z
0	0	0	0	1	0	0
0	0	0	1	0	0	0
0	0	1	0	1	1	0
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	1	0
1	0	0	0	1	0	1
1	0	0	1	1	0	0
1	0	1	0	0	0	0
1	0	1	1	1	0	0
1	1	0	0	1	0	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	0	0

(c) JK: $Q_{t+1} = JQ' + K'Q$

$$A(t+1) = A'B'y'+Bx+Ax'+Ay$$

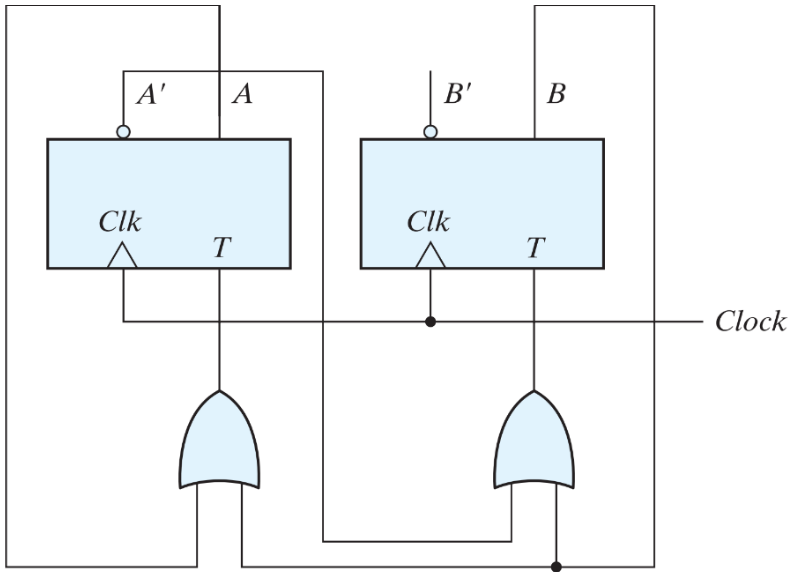
$$B(t+1) = A'B'x+A'Bx'+A'By$$

3. Use a D flip-flop, a 2x1 multiplexer, and a NOT gate to build a JK flip-flop. Draw the circuit diagram. Explain your design.

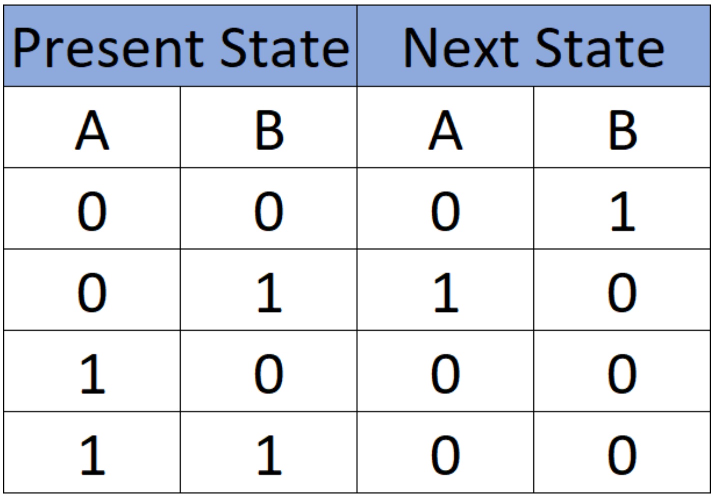


Any reasonable explanations are fine.

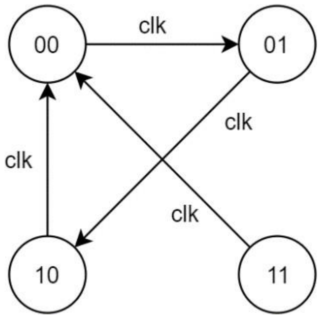
4. Derive the state table and the state diagram of the sequential circuit shown in the figure below. Explain the function that the circuit performs.



(a) $T_A = A + B$ $T_B = A' + B$



Present State		Next State	
A	B	A	B
0	0	0	1
0	1	1	0
1	0	0	0
1	1	0	0



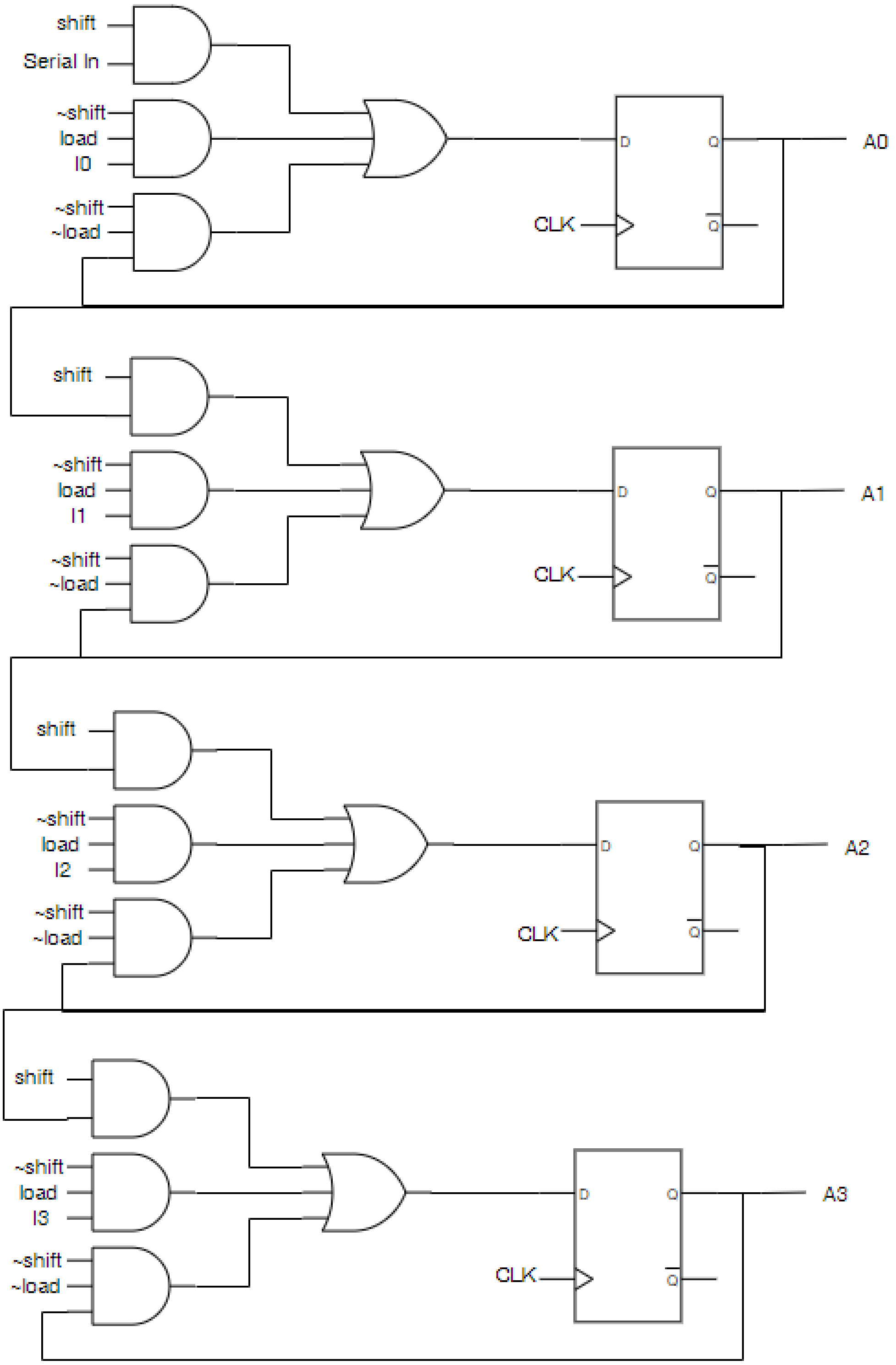
(b)

(c) A mod-3 counter. (Any other reasonable explainations are OK)

ASSIGNMENT5

Provide answers to the following questions:

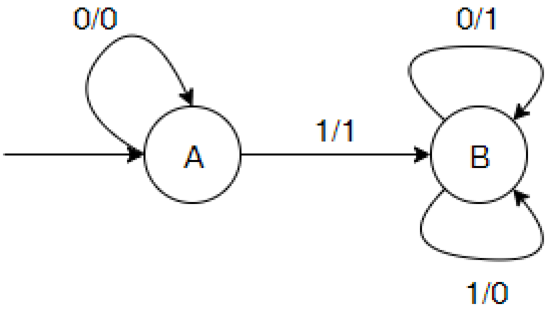
1. What is the difference between serial and parallel transfer? Explain how to convert serial data to parallel and parallel data to serial.
   Serial: **One bit** of data is transmitted at each clock cycle; All data get transmitted using same data line.
   Parallel: **Multiple bits** of data (or byte, character) are transmitted at each clock cycle. Each bit is transmitted using different parallel data lines.
   Conversion: Using a **shift register**. To convert serial to parallel, take one bit at a time and shift all other bits to the other direction. Repeat until all data were received and output in parallel at once. To convert parallel to serial, take multiple bits into shift register at once, and output one bit at a time, with shifting remaining bits to the direction of output. (Any reasonable explanations are OK, refer to the bolded text)
2. Design a four-bit shift left register with parallel load using D flip-flops and gates. There are two control inputs: *shift* and *load*. When *shift* = 1, the content of the register is shifted by one position. New data are transferred into the register when *load* = 1 and *shift* = 0. If both control inputs are equal to 0, the content of the register does not change. Briefly describe your design and draw the circuit diagram.
   Explanation: Any reasonable explanations are OK.

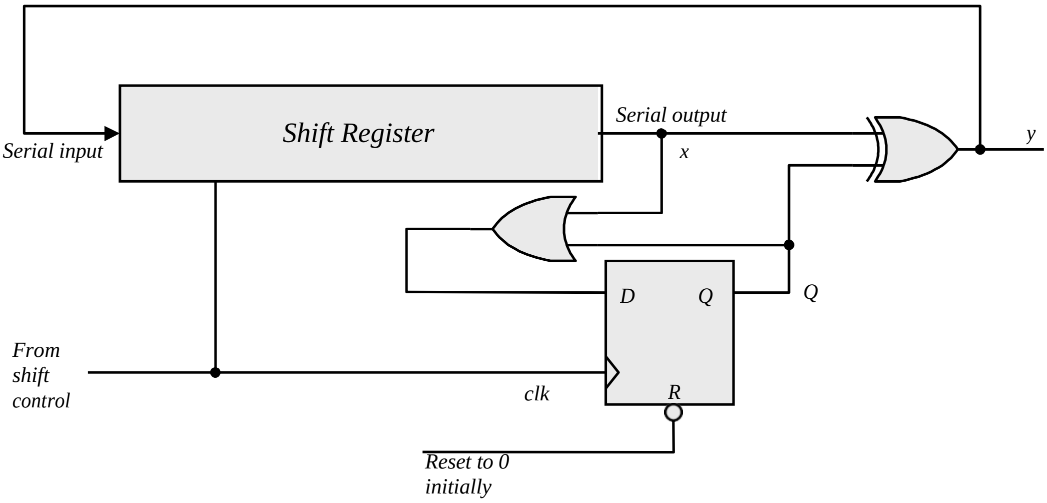


3. Design a serial 2's complementer with a shift register, a flip-flop and gates. The binary number is shifted out from one side and it's 2's complement shifted into the other side of the shift register. Provide the state diagram and draw the circuit diagram.
Hint: - 2's complement of a number can be obtained by keeping the least significant bits as such until the first 1, and then complementing all bits

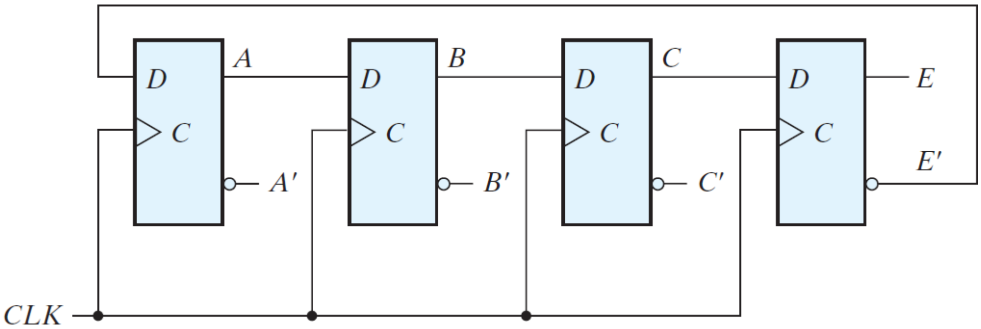
eg: 001010 → 110110

Must have the RST input in figure.





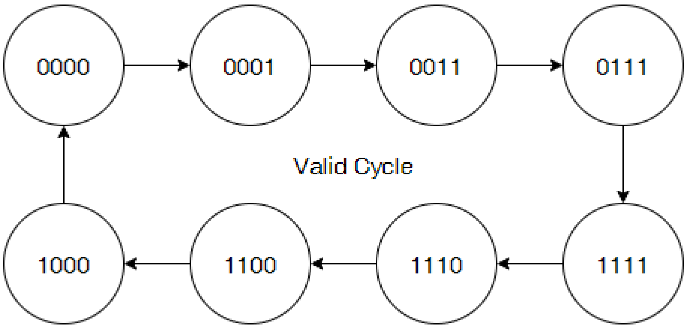
4. List the all used and unused states in a 4-bit switch-tail ring counter show in the figure below. Determine the next state for each of unused states and show that, if the counter finds itself in an invalid state, it does not return to a valid state. Modify the circuit to avoid this. Show that your modified counter produces the same sequence of states and that the circuit reaches a valid state from any one of the unused states.

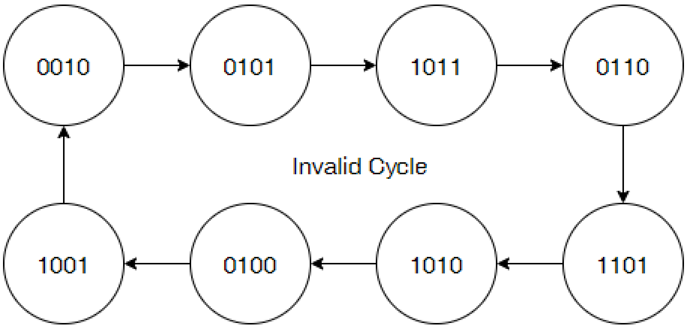


Used: 0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001

Unused: 0010, 0100, 0101, 0110, 1001, 1010, 1011, 1101

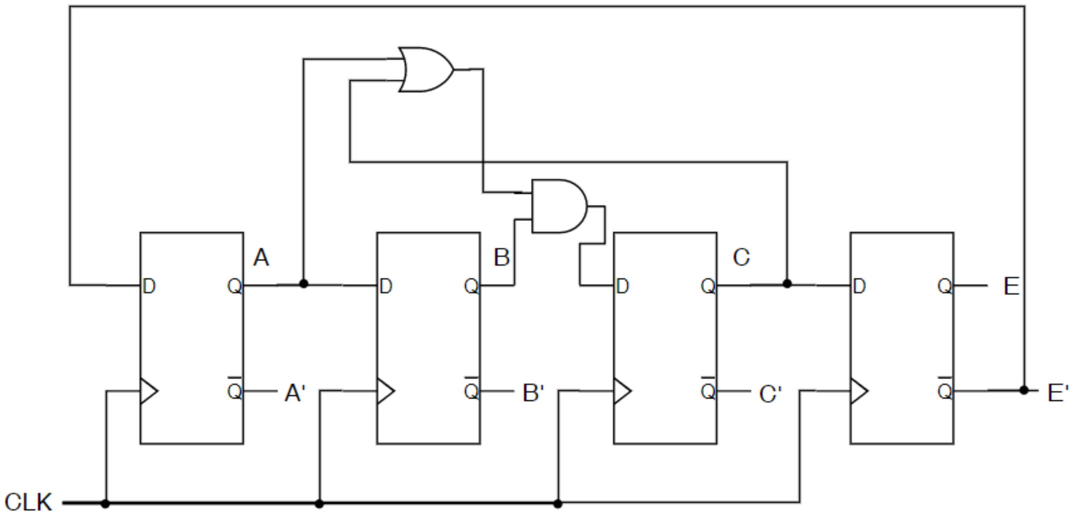
Proof of not returning o valid states, using state diagram:



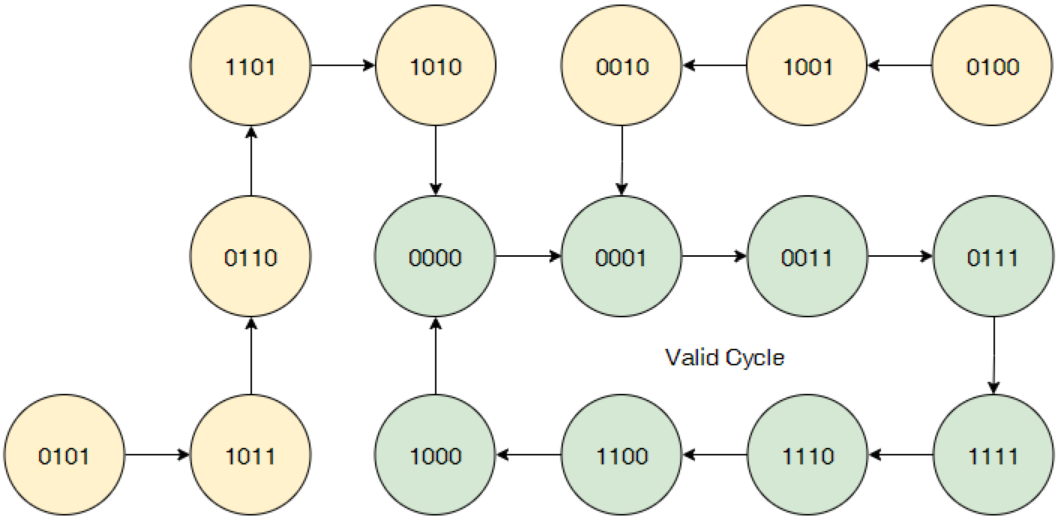


(Or reversed, both OK)

Modification: Let $D_C=(A+C)B$



Proof using state diagram:



So, it would always return to the valid cycle in the new design.

(The answer is not unique. Judge by state diagram.)