End-term Exam (Solution)

1.

B2FA

15's complement : FFFF − B2FA = 4D05

16's complement : 4D05 + 1 = 4D06

Ans.

2.

	Present state			Next State			Flip-flop Inputs		
	A	B	C	A	B	C	T_A	T_B	T_C
0	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	1	0	1	0
3	0	1	1	1	1	1	1	0	0
7	1	1	1	1	1	0	0	0	1
6	1	1	0	1	0	0	0	1	0
4	1	0	0	0	0	0	1	0	0
2	0	1	0	x	x	x	x	x	x
5	1	0	1	x	x	x	x	x	x

A \ BC	00	01	11	10
0	0 (0)	0 (1)	1 (3)	X (2)
1	1 (4)	X (5)	0 (7)	0 (6)

$$T_A = A \oplus B$$

A \ BC	00	01	11	10
0	0 (0)	1 (1)	0 (3)	X (2)
1	0 (4)	X (5)	0 (7)	1 (6)

$$T_B = B \oplus C$$

A \ BC	00	01	11	10
0	1 (0)	0 (1)	0 (3)	X (2)
1	0 (4)	X (5)	1 (7)	0 (6)

$$T_C = (A \oplus C)'$$

If this counter somehow reaches the binary state 010, the next state would be 101 and then the state after that would be 010 again (so, it will be stuck in that loop of invalid states).

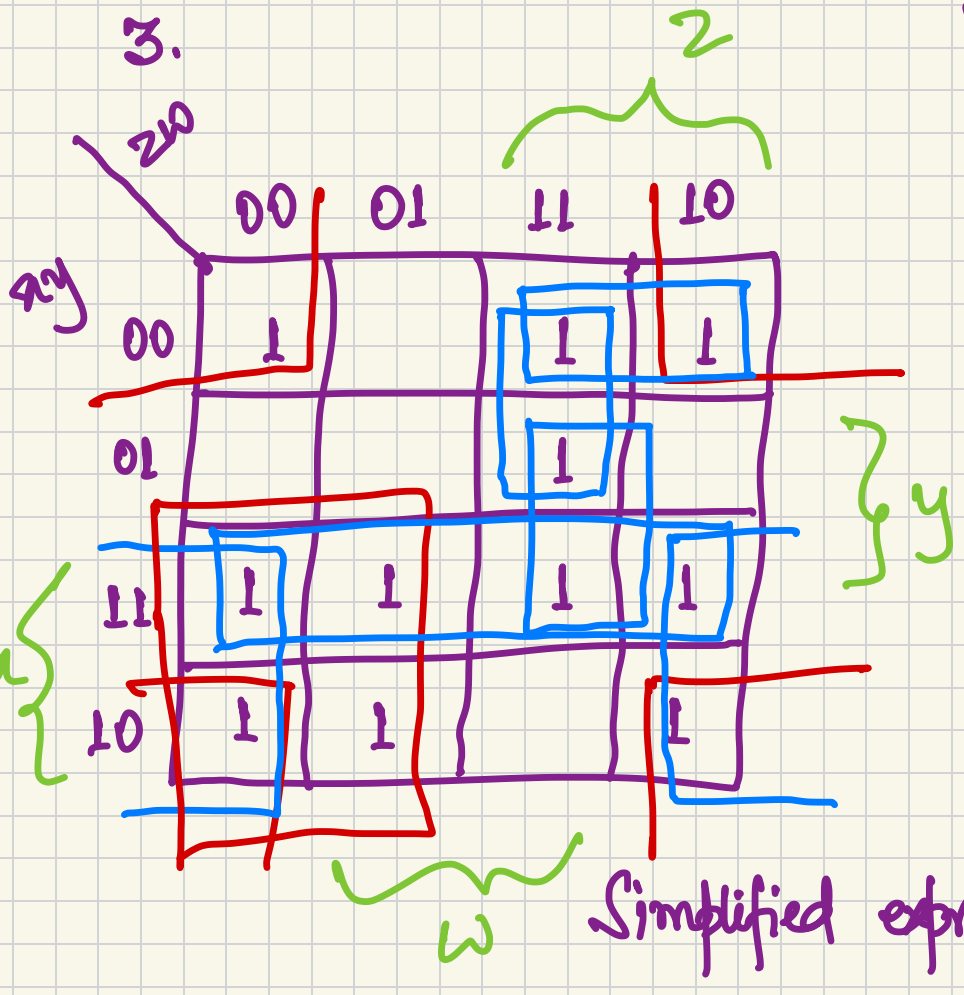
Why? Because all the X's in the flip-flop inputs have been taken as 1, so each input state bit will complement each time.

One way to correct the design could be to assign valid next-state values for the states 010 and 101. For instance, we may decide to have the last two rows of the state table as:

Present state			Next state			Flip-flop Inputs		
A	B	C	A	B	C	T_A	T_B	T_C
0	1	0	0	0	0	0	1	0
1	0	1	0	0	0	1	0	1

And then design the circuit. ∎

3.



Prime implicants:

$w'y'$, $w'x$,

xz' , xy ,

$x'y'z$, $x'zw$,

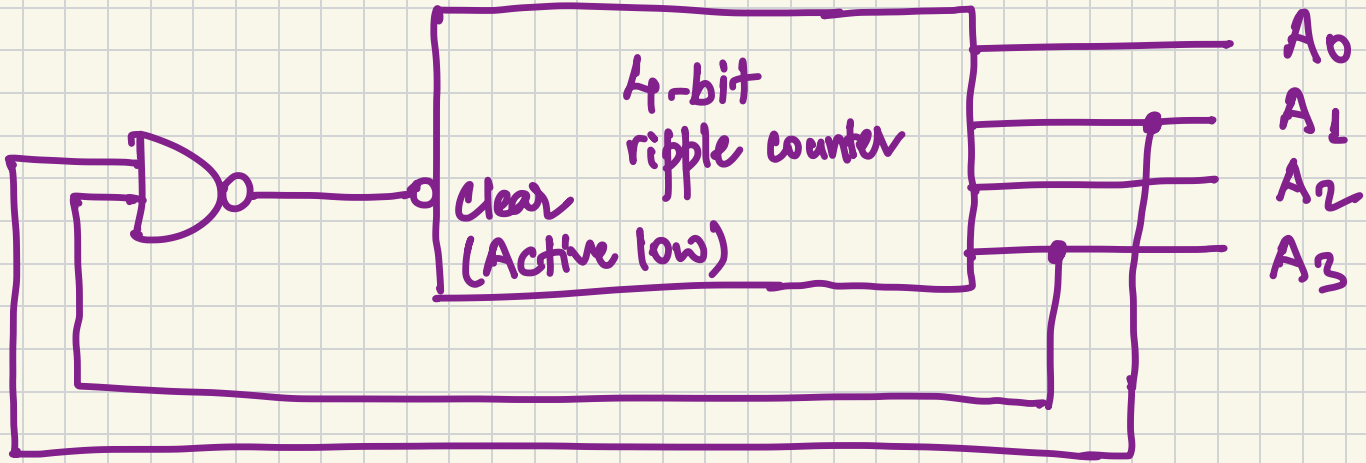
yzw

Essential prime implicants are shown in red box.

Simplified expression: $xz' + w'y' + xy + x'zw$

Ans.

4.



On 1010 there is an asynchronous reset and therefore it behaves like a BCD counter.

5. There are 13 common pins.

The same pins are used for row as well as column addresses.

No. of pins for column address : 12

(since row addresses have one more bit than column addresses)

No. of bits in the address : 13 + 12 = 25

No. of different addresses : 2^{25}

$\therefore$ Memory capacity = 2^{25} words.

Ans.

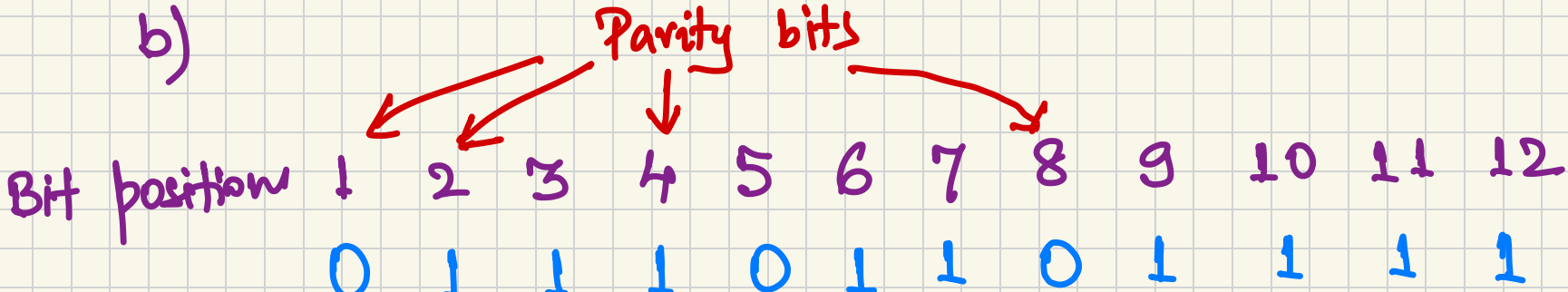
6. a) For single error detection and single error correction, we know that we need k parity bits where k is the smallest integer satisfying

$$2^k \geq 30 + k + 1$$

clearly, $k = 6$.

For double-error detection (and single error correction), we need another bit. So, total no. of check bits needed is 7. Ans.

b)



C_1 = XOR of bits (1, 3, 5, 7, 9, 11) = 0

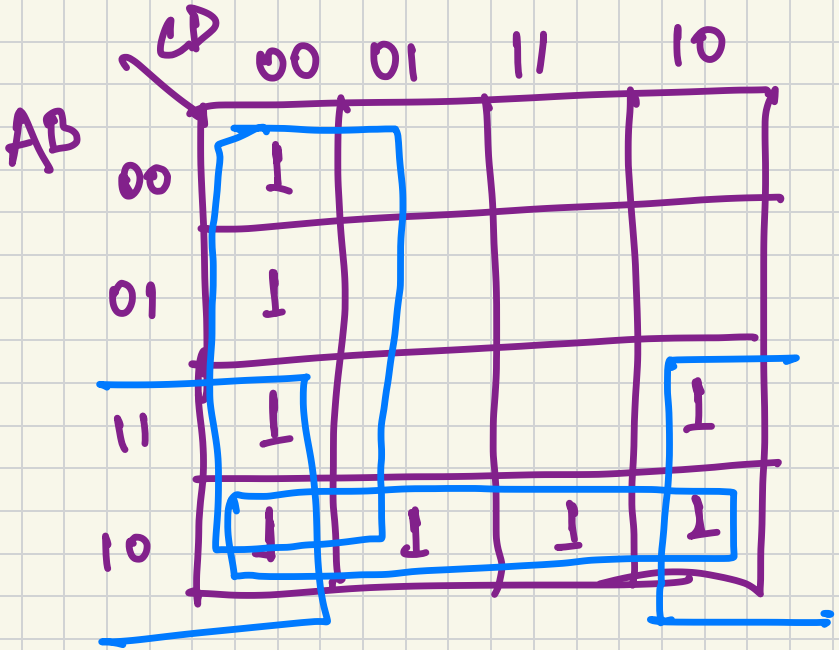
C_2 = XOR of bits (2, 3, 6, 7, 10, 11) = 0

C_4 = XOR of bits (4, 5, 6, 7, 12) = 0

C_8 = XOR of bits (8, 9, 10, 11, 12) = 0

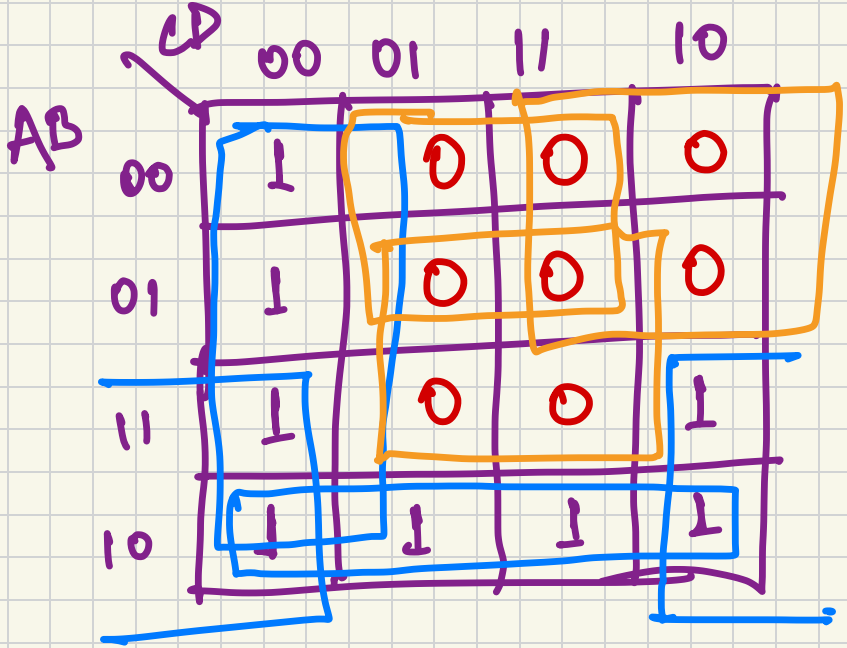
Clearly, there were no errors. So, the original 8-bit data word was 10111111 Ans.

7. $F(A,B,C,D) = \sum(0,4,8,9,10,11,12,14)$



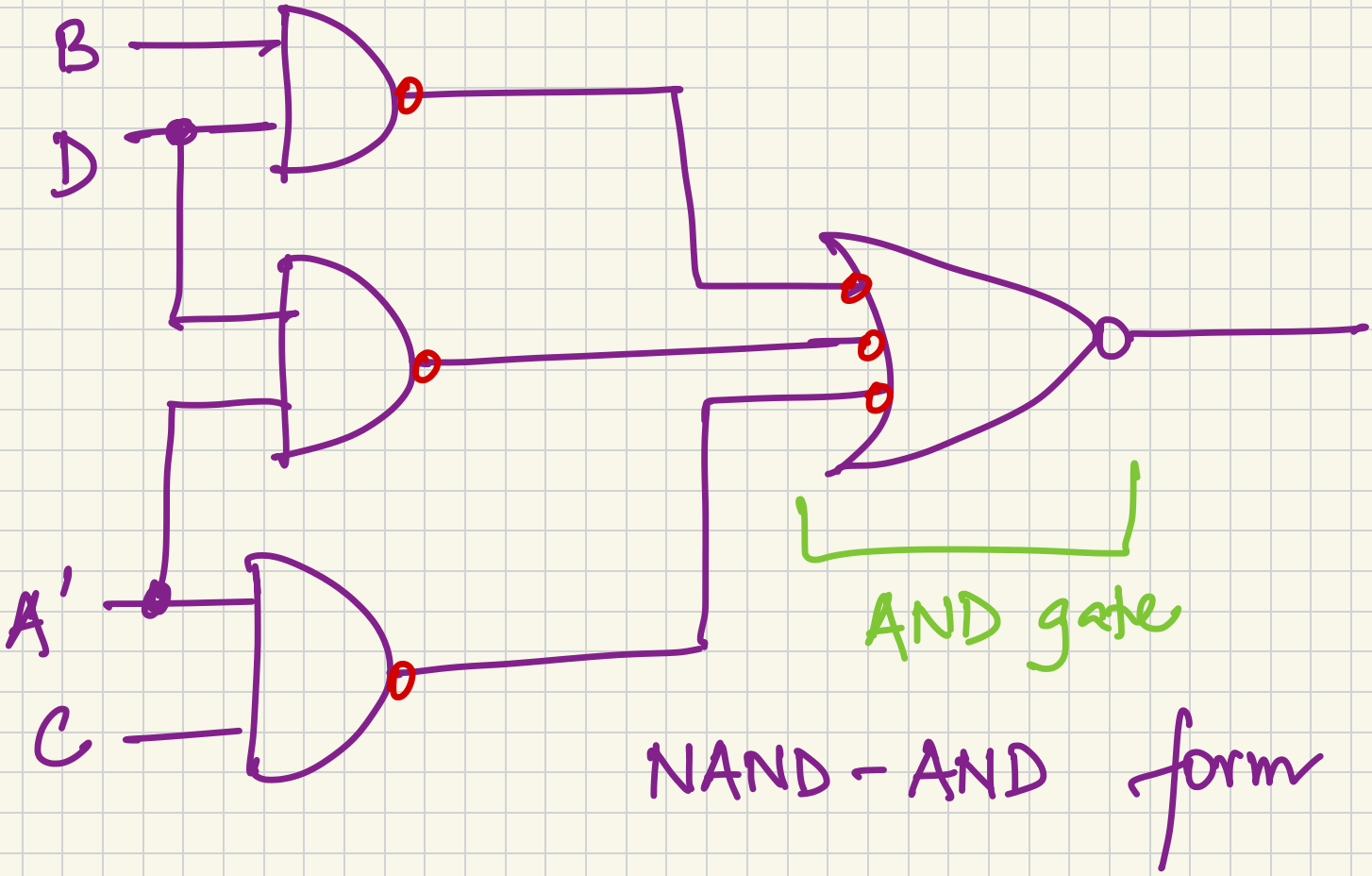
$F = C'D' + AB' + AD'$

i)



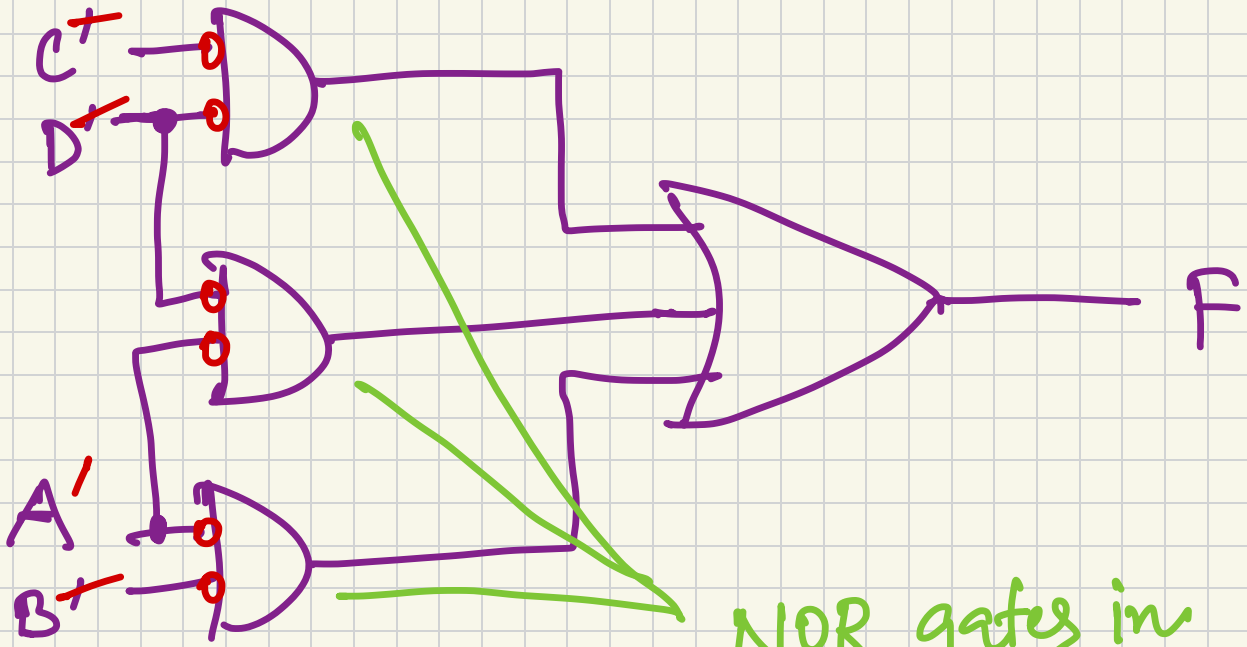
$$F' = BD + A'D + A'C$$

$$F = (BD + A'D + A'C)'$$



NAND - AND form

ii) NOR-OR



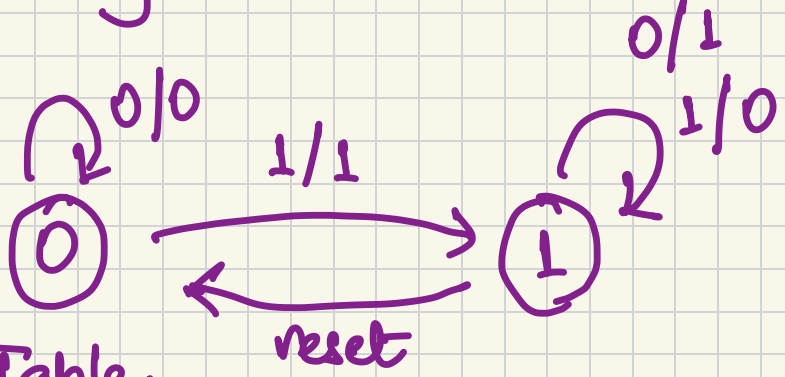
NOR-OR form

NOR gates in invert-AND representation

8. The 2's complement can be formed by leaving all least significant 0's and the first 1 unchanged, and replacing 1's with 0's and 0's with 1's in all other higher significant digits.

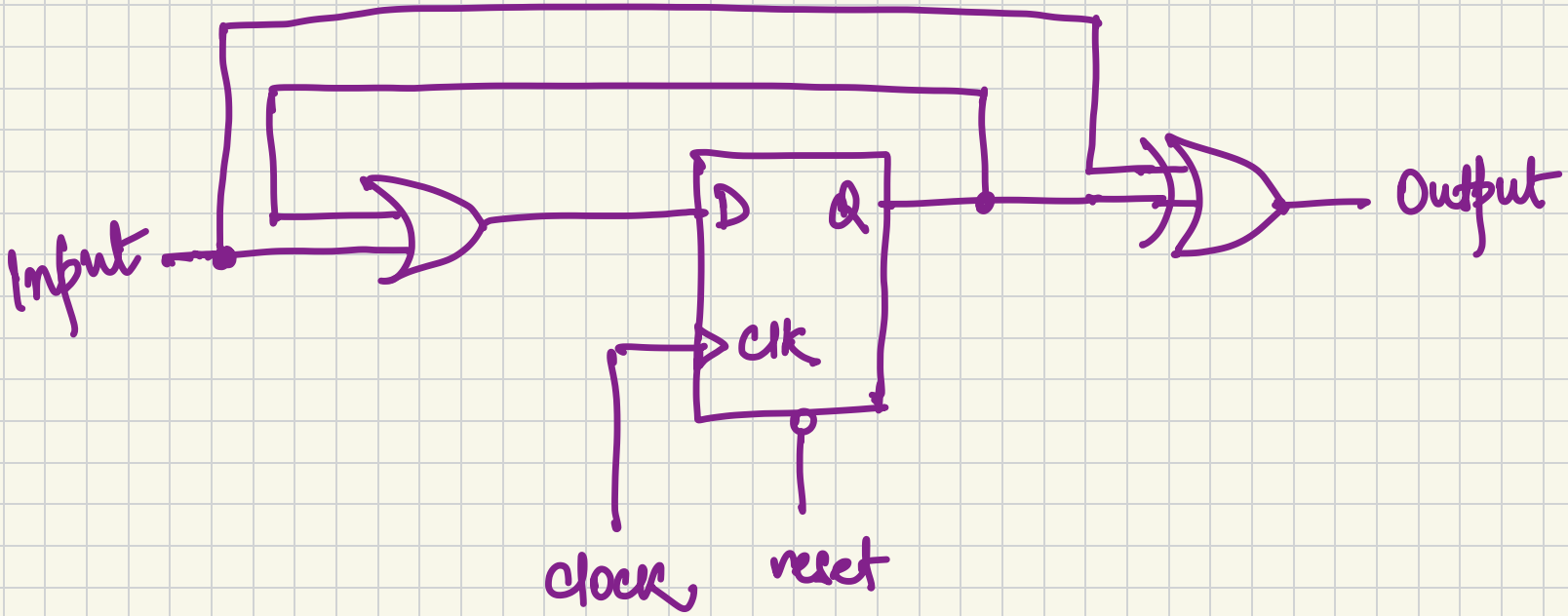
(It is easy to see that this is exactly what happens when you take 1's complement and then add 1.)

State Diagram



State Table

Present state	Input	Next state	Output
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0


Input
D
Q
clk
Output
clock
reset

9.

	Inputs			Outputs			
	x	y	z	A	B	C	D
0	0	0	0	1	1	0	1
1	0	0	1	0	1	1	1
2	0	1	0	0	0	0	0
3	0	1	1	1	1	0	0
4	1	0	0	1	0	0	1
5	1	0	1	0	0	1	1
6	1	1	0	1	0	0	0
7	1	1	1	0	1	0	1

Memory contents at address 2 and 5