

7.2-)

ABC	A ⁺	B ⁺	C ⁺
000	0	1	0
001	0	0	0
010	1	1	1
011	0	0	1
100	1	1	0
101	X	X	X
110	0	1	1
111	1	0	0

A⁺

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

B⁺

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

C⁺

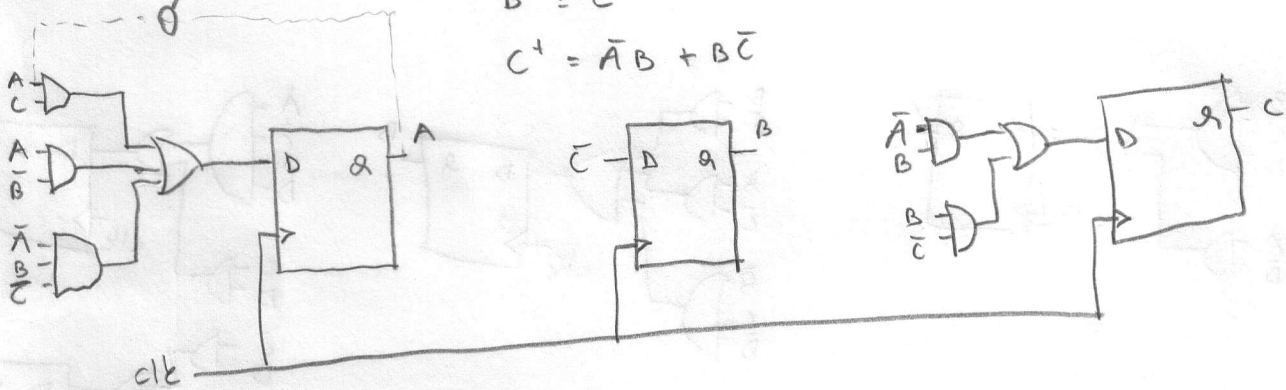
A\BC	00	01	10	11
0	0	0	1	1
1	0	X	0	1

these corrections implicitly indicated by labels

$$A^+ = A\bar{B} + AC + \bar{A}B\bar{C}$$

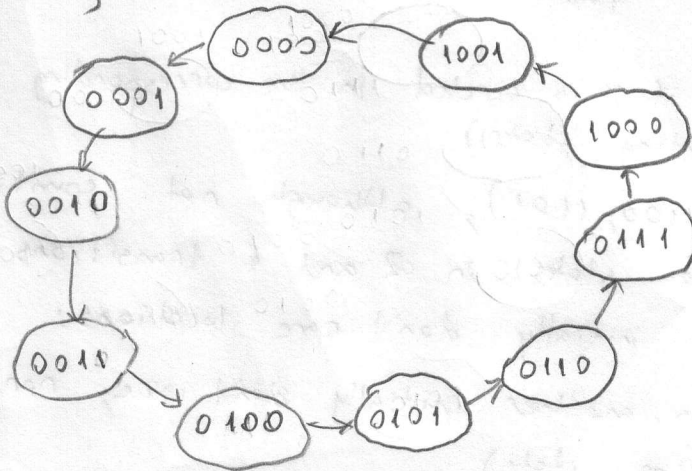
$$B^+ = \bar{C}$$

$$C^+ = \bar{A}B + B\bar{C}$$



Notice that 101 state transitions to 100, which is a counting state; hence the counter is self-starting.

7.3-)



ABCD	A ⁺	B ⁺	C ⁺	D ⁺
0000	0	0	0	1
0001	0	0	1	0
0010	0	0	1	1
0011	0	1	0	0
0100	0	1	0	1
0101	0	1	1	0
0110	0	1	1	1
0111	1	0	0	0
1000	1	0	0	1
1001	0	0	0	0
1010	X	X	X	X
1011	X	X	X	X
1100	X	X	X	X
1101	X	X	X	X
1110	X	X	X	X
1111	X	X	X	X

(A⁺)

	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	1	0	X	X

$$A^+ = BCD + A\bar{D}$$

(B⁺)

	00	01	10	11
00	0	0	1	0
01	1	1	0	1
11	X	X	X	X
10	0	0	X	X

$$B^+ = B\bar{D} + B\bar{C} + \bar{B}C\bar{D}$$

(C⁺)

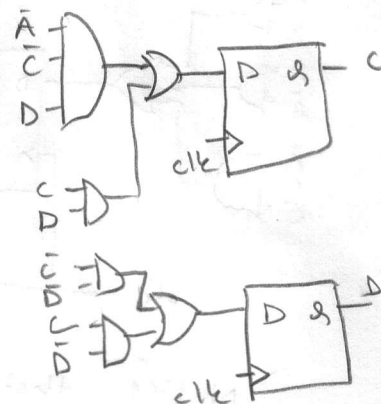
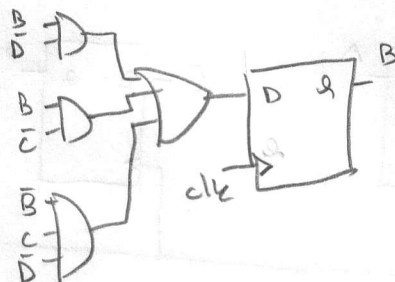
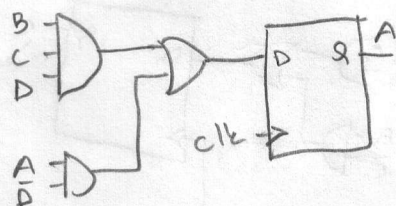
	00	01	10	11
00	0	1	0	1
01	0	1	0	1
11	X	X	X	X
10	0	0	X	X

$$C^+ = \bar{A}\bar{C}D + CD$$

(D⁺)

	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	X	X	X	X
10	1	0	X	X

$$D^+ = \bar{C}\bar{D} + C\bar{D}$$



Using tables above, observe that

1100 → 1101 (put 1 if X is circled in the corresponding K-map.)

1101 → 0101 (a counting state)

Hence these two states (1100, 1101), although not counting states, converge into counting states in 2 and 1 transitions respectively. Check other initially don't care locations:

1110 → 1111 (1111 is another initially don't care, non-counting, state)

1111 → 1000 (a counting state)

Only two non-counting states left to check:

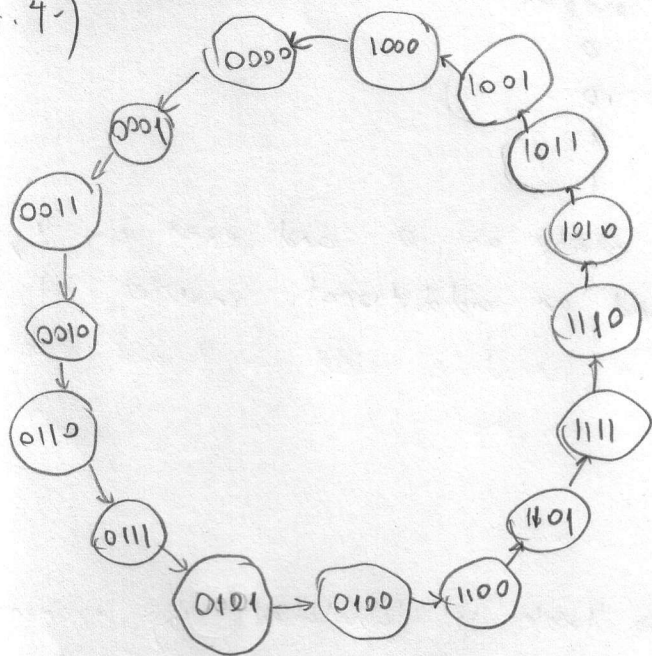
1010 → 1011

1011 → 0100 (this converged to a counting state too)

Hence, this implementation is self-starting. If it was not, one could re-adjust circling of X's. This can systematically be done by changing only the output don't cares to a counting state for

the non-counting states which are identified as diverging or cycling.

7.4-)



AB CD	A ⁺ B ⁺ C ⁺ D ⁺
0000	0 0 0 1
0001	0 0 1 1
0010	0 1 1 0
0011	0 0 1 0
0100	1 1 0 0
0101	0 1 0 0
0110	0 1 1 1
0111	0 1 0 1
1000	0 0 0 0
1001	1 0 0 0
1010	1 0 1 1
1011	1 0 0 1
1100	1 1 0 1
1101	1 1 1 1
1110	1 0 1 0
1111	1 1 1 0

A⁺

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

$$A^+ = B\bar{C}\bar{D} + AD + AC$$

B⁺

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

$$B^+ = B\bar{C} + BD$$

C⁺

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

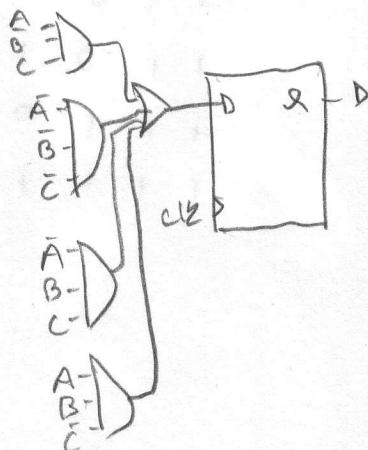
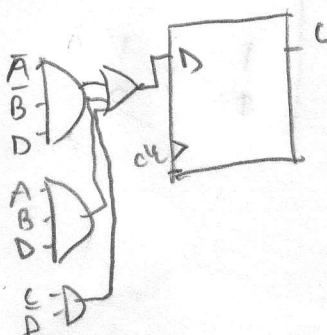
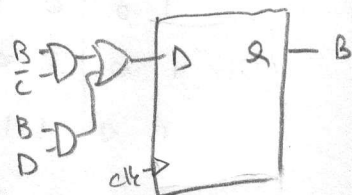
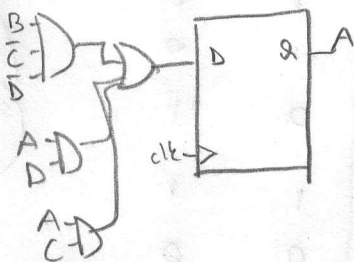
$$C^+ = \bar{A}\bar{B}D + ABD + C\bar{D}$$

D⁺

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

$$D^+ = \bar{A}\bar{B}\bar{C} + \bar{A}BC + AB\bar{C} + A\bar{B}C$$

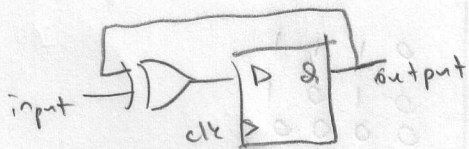
Since all possible counting states are used, self-starting is guaranteed.



7.16-)

Present state	input	Next state	output
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

This time, we encoded the odd state as 0 and even as 1, so we have saved an additional inverter at the output.



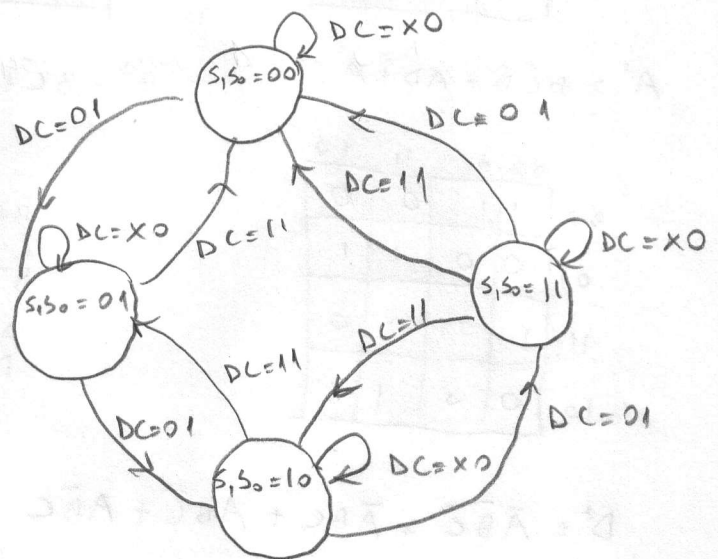
7.20-)

$$N_1 = ((S_1 \oplus S_0) \oplus D)C + S_1 \bar{C}$$

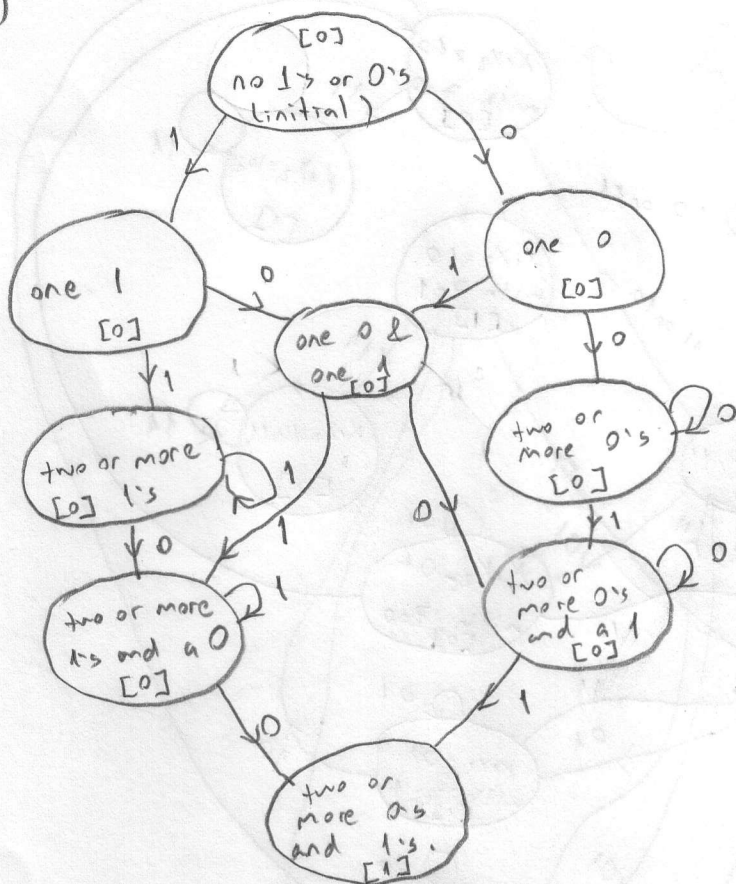
$$N_0 = S_0 \oplus C$$

S_1	S_0	D	C	N_1	N_0
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	1
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	0	0
1	1	1	0	1	1
1	1	1	1	1	0

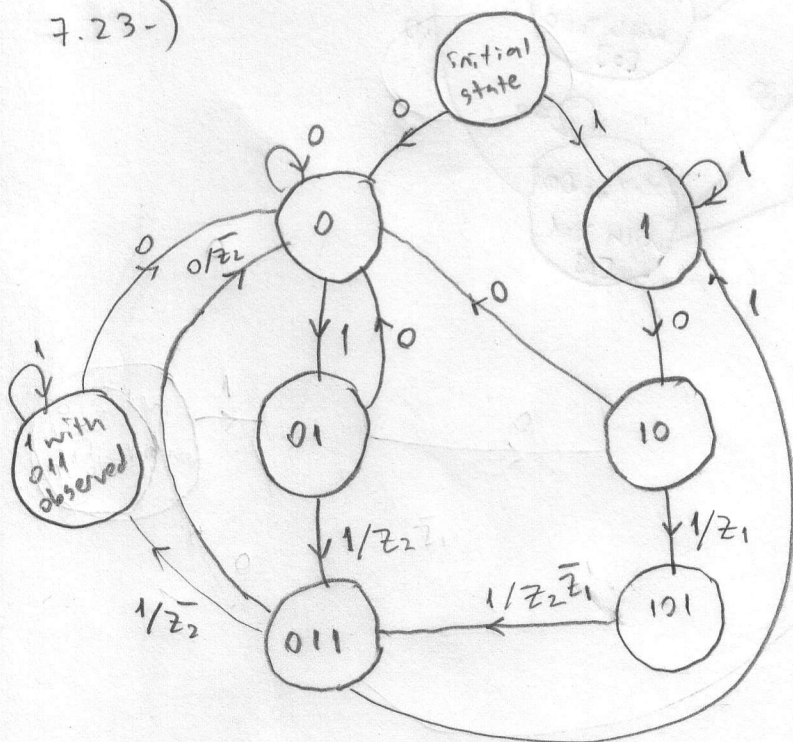
State table is tabulated by examining the circuit and keeping in mind that an XOR toggles an input when its other input is a 1 and passes it directly when its other input is 0. This way, you do not have to exhaustively try all $S_1 S_0 D C$ pairs on the circuit.



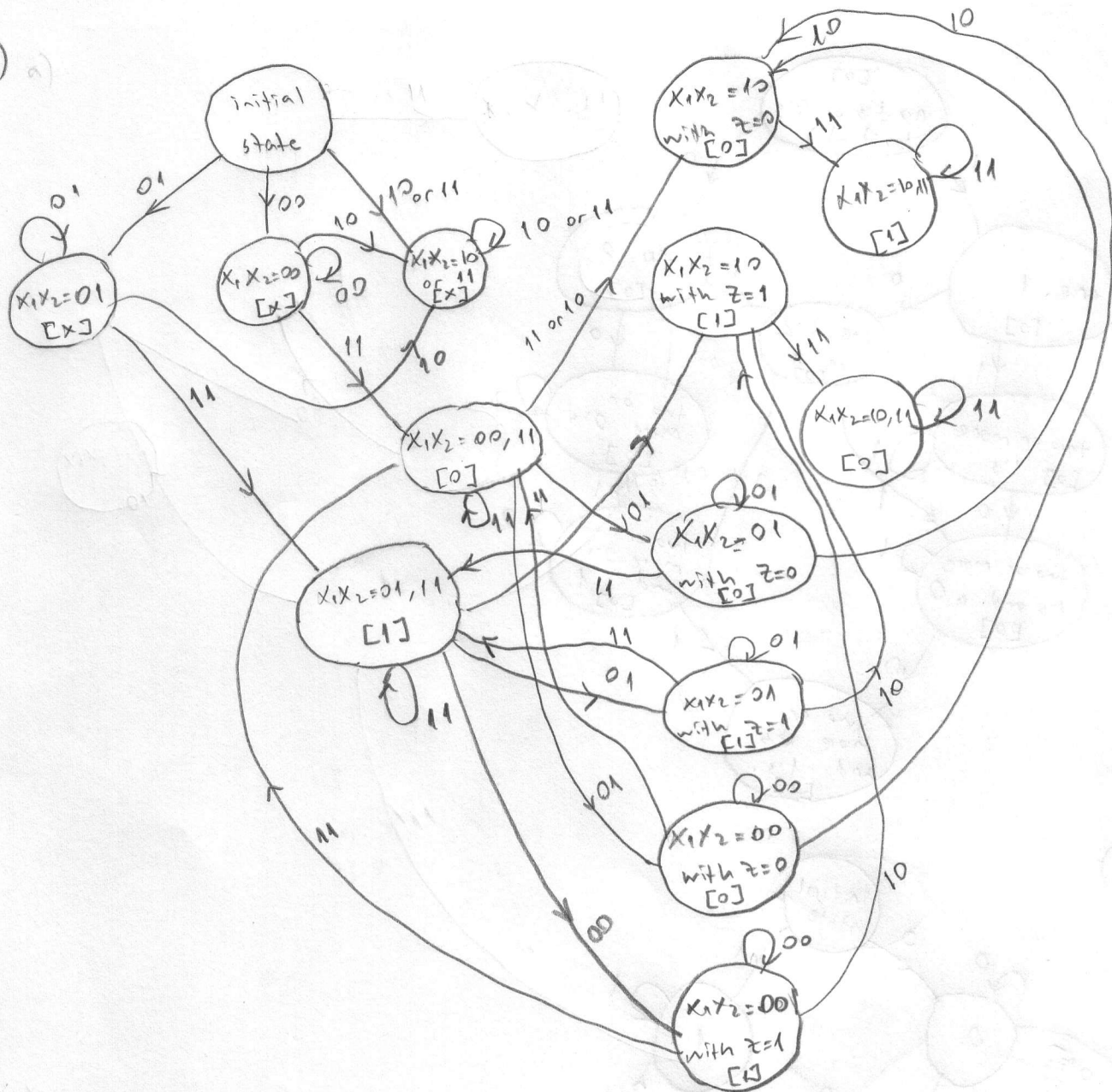
7.22-)



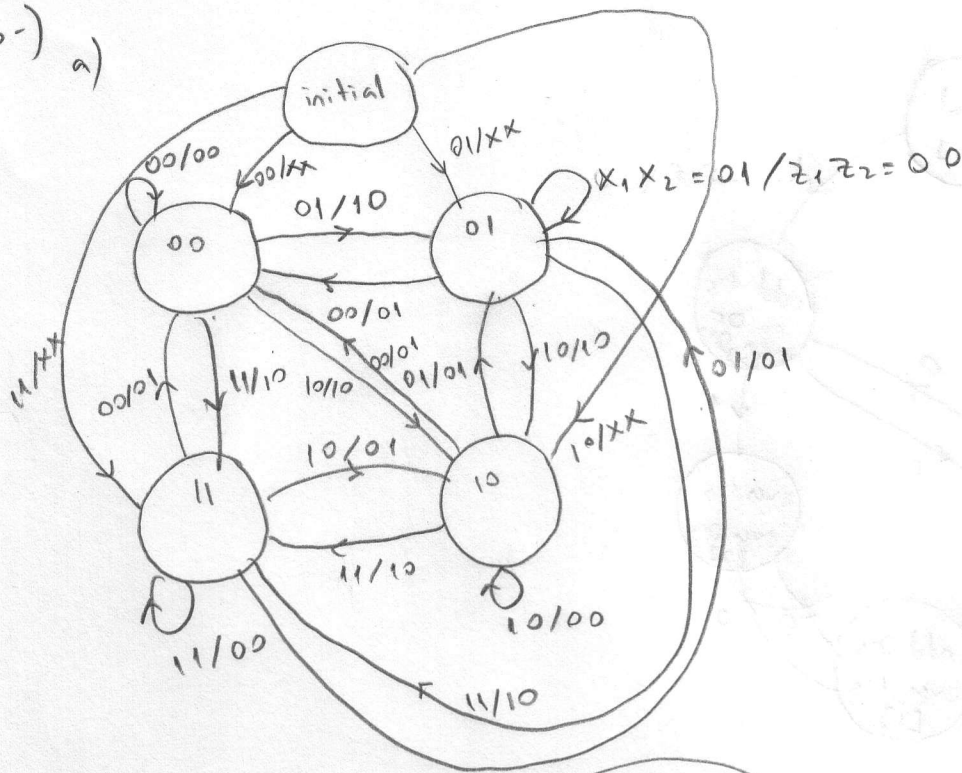
7.23-)



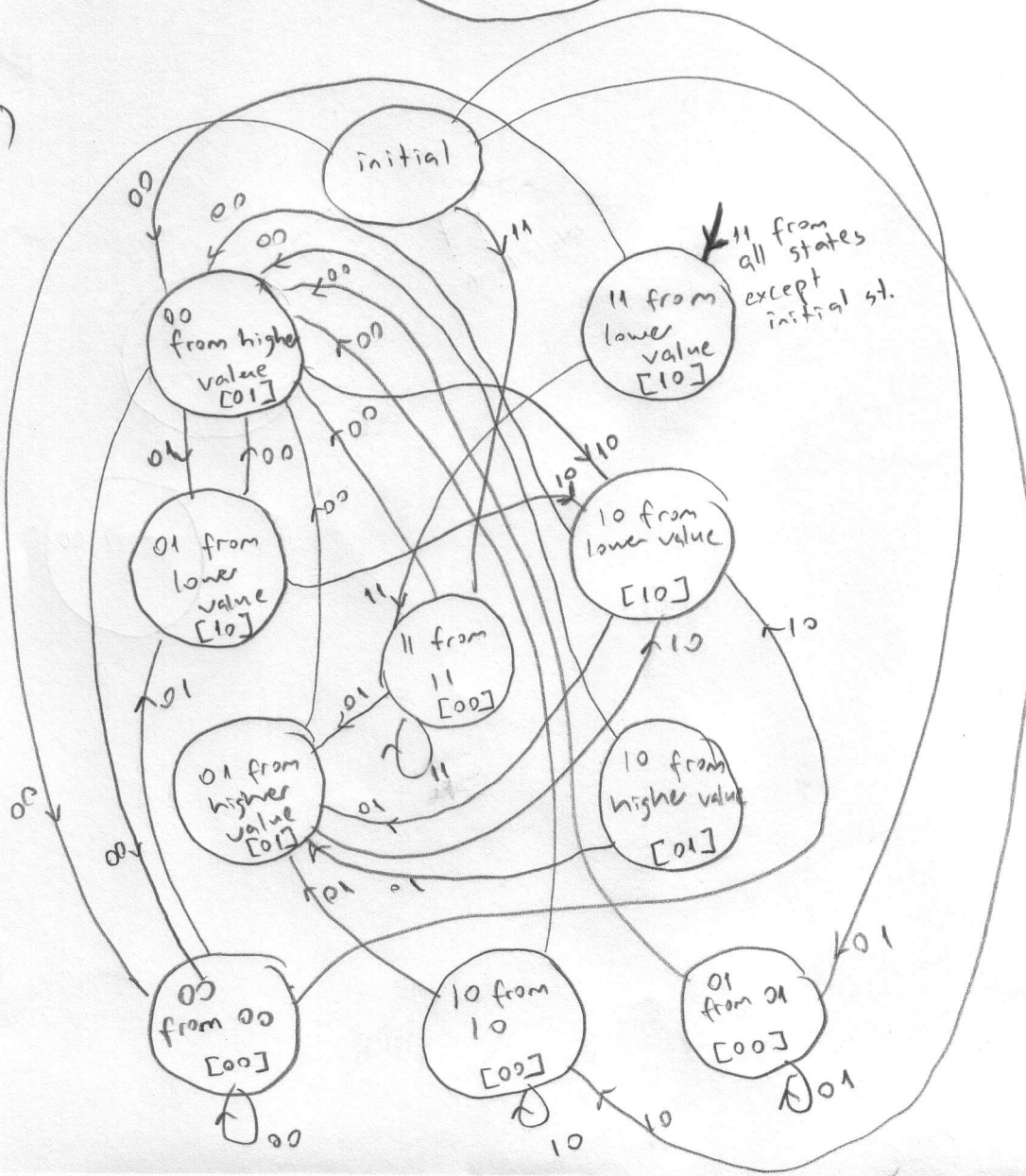
7.24-)



7.26-) a)



b)



7-27-

