

Exercise Q3.17

- Design an FSM to keep track of the mood of four students working in the digital design lab. Each Student is either:
 1. Happy (the circuit works)
 2. Sad (the circuit blew up)
 3. Busy (working on the circuit)
 4. Clueless (confused about the circuit)
 5. Asleep (face down on the circuit board)
- How many states does the FSM have ? What is the minimum number of bits necessary to represent these states ?

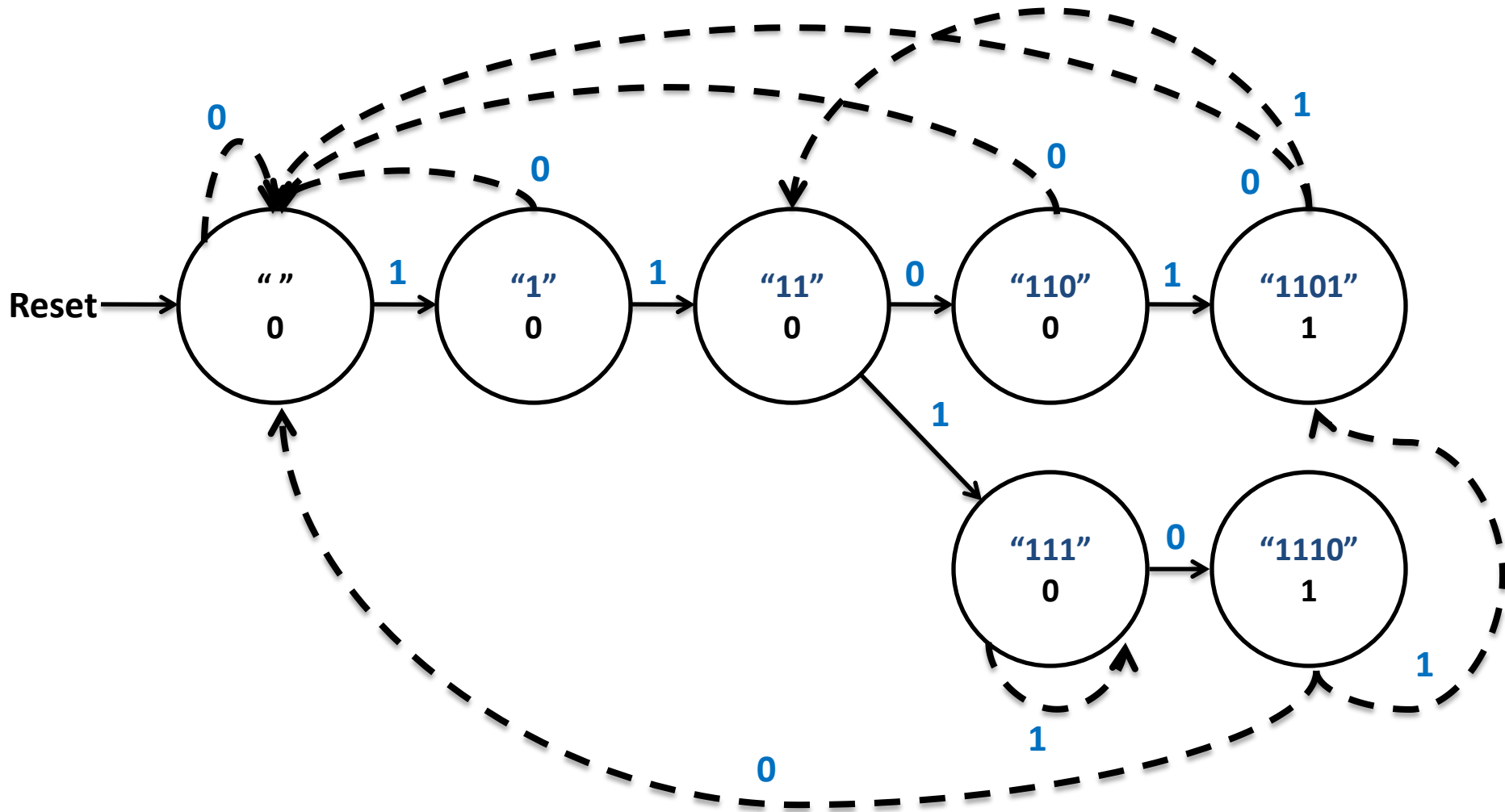
Exercise Q3.17 Solution

- Each Student can either be:
 1. Happy (the circuit works).
 2. Sad (the circuit blew up).
 3. Busy (working on the circuit).
 4. Clueless (confused about the circuit).
 5. Asleep (face down on the circuit board).
- So each student can be in five different states. Hence, we can say that overall we have 625 distinct states (5 possible states for 4 students, $5^4 = 625$).
- For 625 distinct states, we need a minimum of $\text{ceiling}(\log_2 625) = 10$ bits.

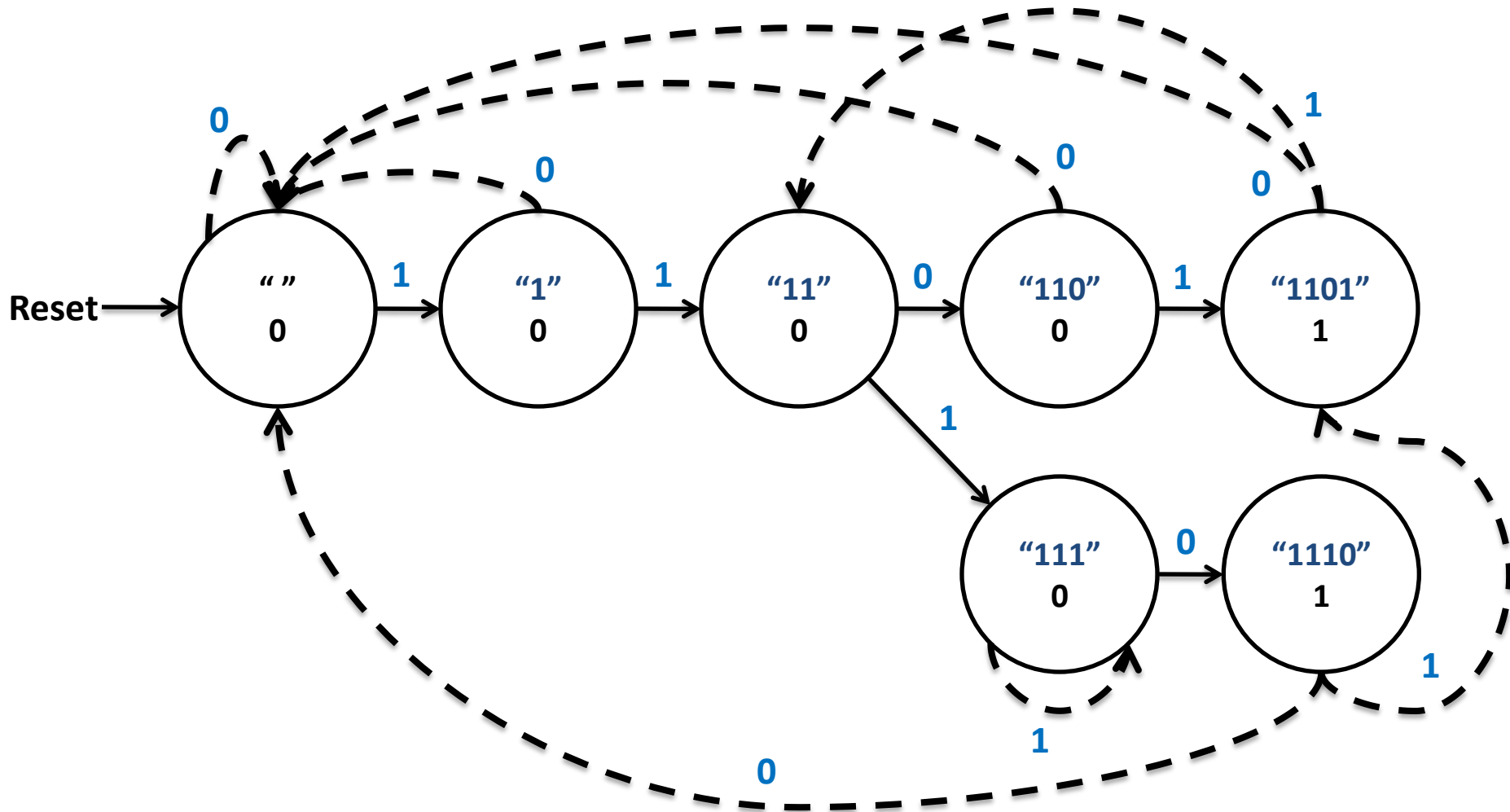
Exercise Q3.22

- Design an FSM that recognizes 1101 or 1110.
 - Draw state transition diagram (use as few states as possible).
 - Choose state encodings.
 - Write state transition and output table using the encodings.
 - Write next state equations and output equations.

Exercise Q3.22 Solution



State encoding:



A = 000 ; B = 001 ; C = 010 ; D = 011; E = 100 ; F = 101 ; G = 110

State transition and output table:

Present State S2 S1 S0	X = 0	X = 1	F
	NS S2+ S1+ S0+	NS S2+ S1+ S0+	
0 0 0	0 0 0	0 0 1	0
0 0 1	0 0 0	0 1 0	0
0 1 0	0 1 1	1 0 1	0
0 1 1	0 0 0	1 0 0	0
1 0 0	0 0 0	0 1 0	1
1 0 1	1 1 0	1 0 1	0
1 1 0	0 0 0	1 0 0	1

K-Maps:

S0+

		S0 X			
		00	01	11	10
S2 S1	00	0	1	0	0
	01	1	1	0	0
	11	0	0	X	X
	10	0	0	1	0

$$S0+ = S0'.S1.S2' + X.S0'.S2' + X.S0.S2$$

S1+

		S0 X			
		00	01	11	10
S2 S1	00	0	0	1	0
	01	1	0	0	0
	11	0	0	X	X
	10	0	1	0	1

$$S1+ = X.S0.S1'.S2' + X'.S0'.S1.S2' + X.S0'.S1'.S2 + X'.S0.S2$$

S2+

		S0 X			
		00	01	11	10
S2 S1	00	0	0	0	0
	01	0	1	1	0
	11	0	1	X	X
	10	0	0	1	1

$$S2+ = S0.S2 + X.S1$$

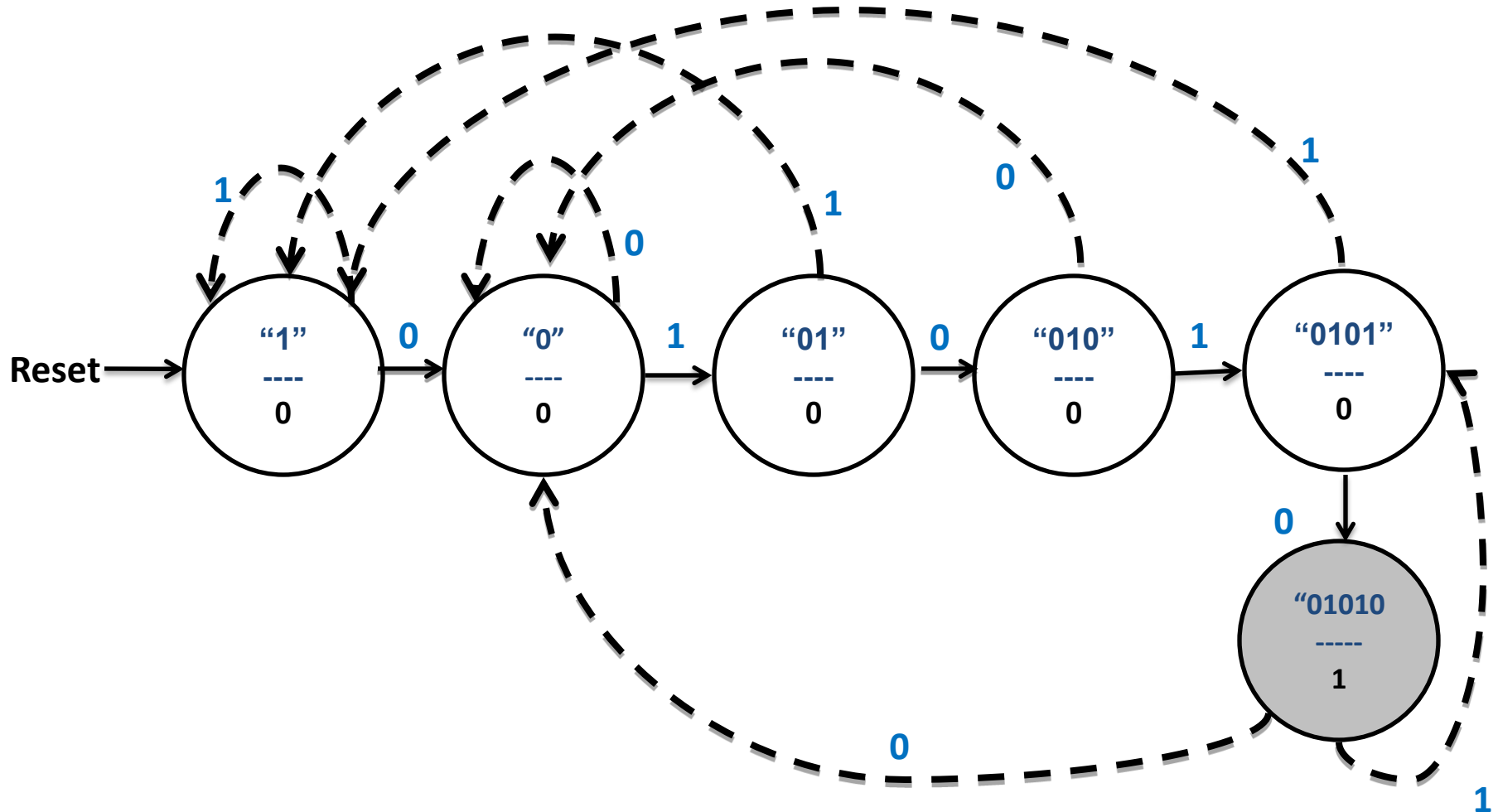
F

		S0 X	
		00	01
S2 S1	00	0	0
	01	0	0
	11	1	X
	10	1	0

$$F = S0'.S2$$

Interview Q 3.1 (textbook)

Design an FSM that recognizes 01010 when it is received serially.



Exercise Q3.27 (textbook)

- Design an FSM with one input, A , and two outputs, X and Y .
- X should be 1 if A has been 1 for at least three cycles altogether (not necessarily consecutively).
- Y should be 1 if A has been 1 for at least two consecutive cycles.
- Show your state transition diagram, encoded state transition table, next state and output equations, and schematic.

Important : Understand the problem correctly

- X should be 1 if A has been 1 for at least three cycles altogether (not necessarily consecutively).
- Y should be 1 if A has been 1 for at least two consecutive cycles.

Sample Pattern1 (assuming Moore Machine, output is 1 cycle delayed)

Input A	0	1	1	0	1	1	1
Output Y	0	0	0	1	1	1	1
Output X	0	0	0	0	0	1	1

Sample Pattern2

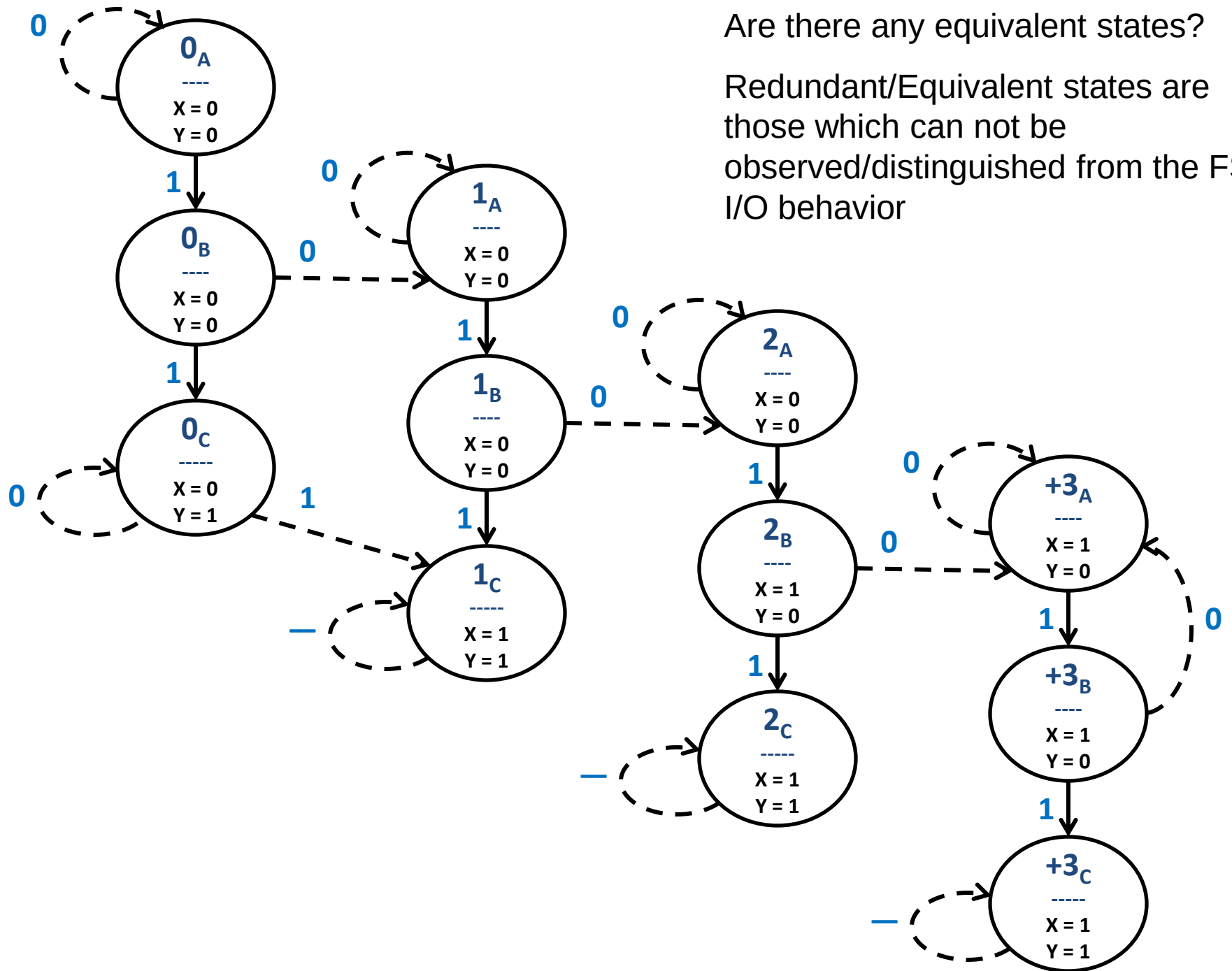
A	0	1	1	1	0	0	1
Y	0	0	0	1	1	1	1
X	0	0	0	0	1	1	1

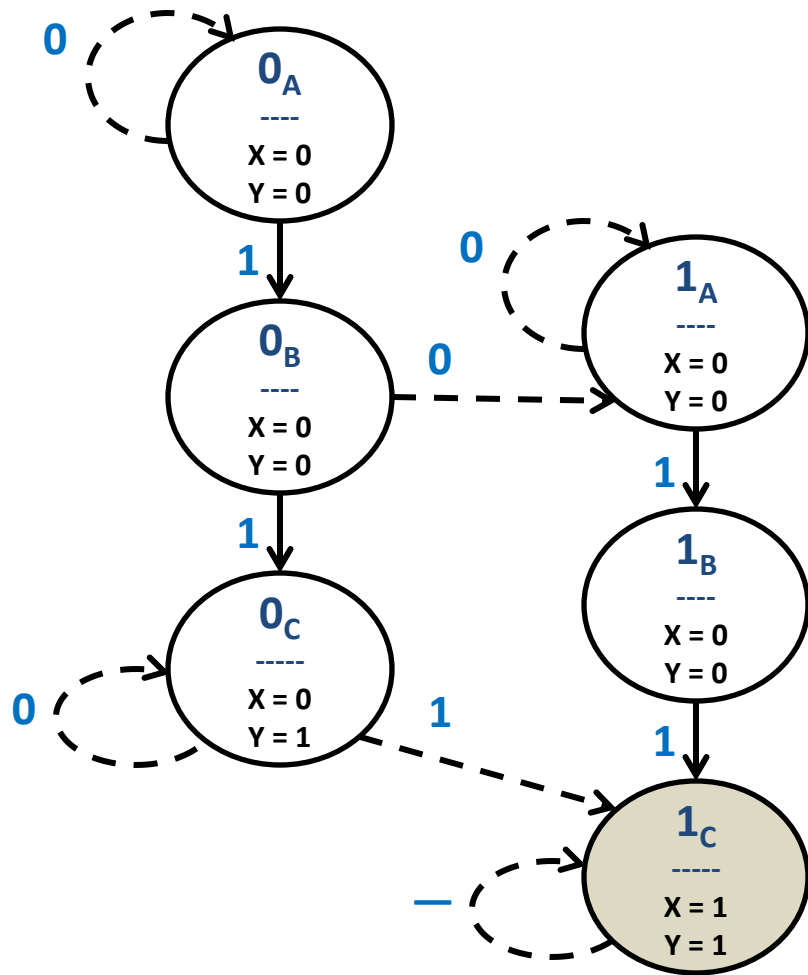
Sample Pattern3

A	1	0	1	0	1	1	0
Y	0	0	0	0	0	0	1
X	0	0	0	0	0	1	1

Sample Pattern4

A	1	0	1	1	1	1	0
Y	0	0	0	0	1	1	1
X	0	0	0	0	1	1	1





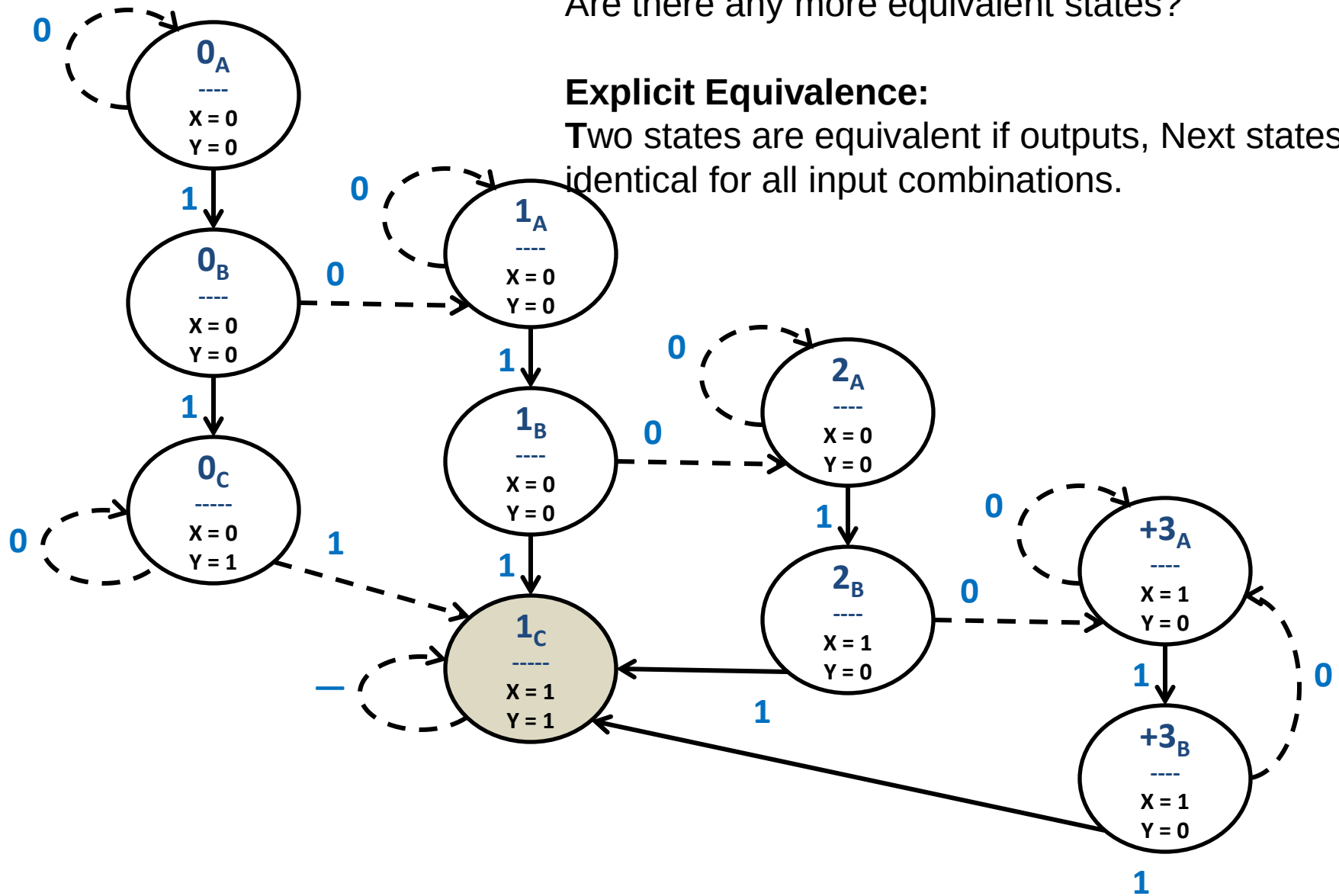
Combining equivalent states 1c, 2c, 3c

Reason : once these states are reached, output is always $X=1, Y=1$ for any input sequence.

Are there any more equivalent states?

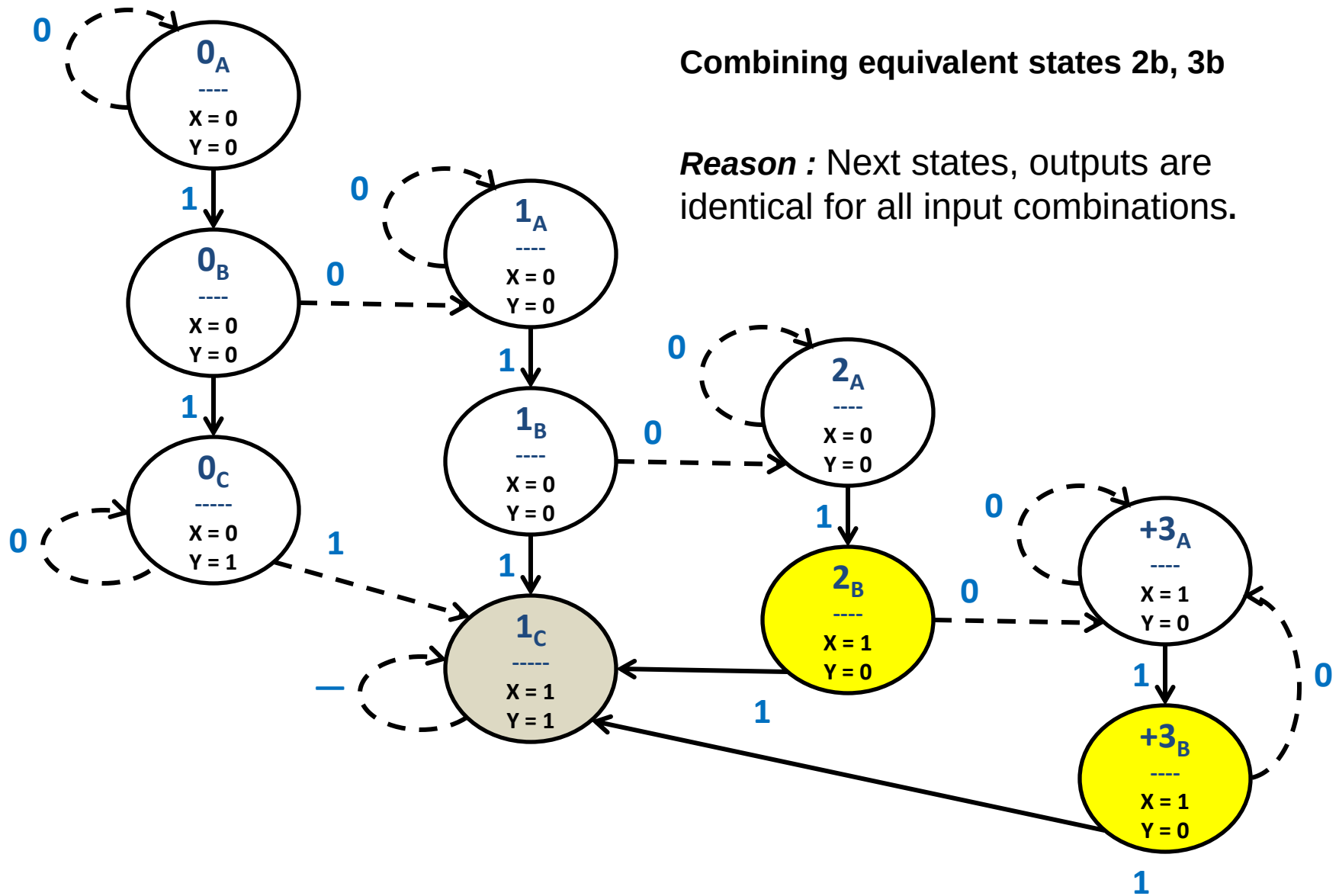
Explicit Equivalence:

Two states are equivalent if outputs, Next states are identical for all input combinations.



Combining equivalent states 2b, 3b

Reason : Next states, outputs are identical for all input combinations.

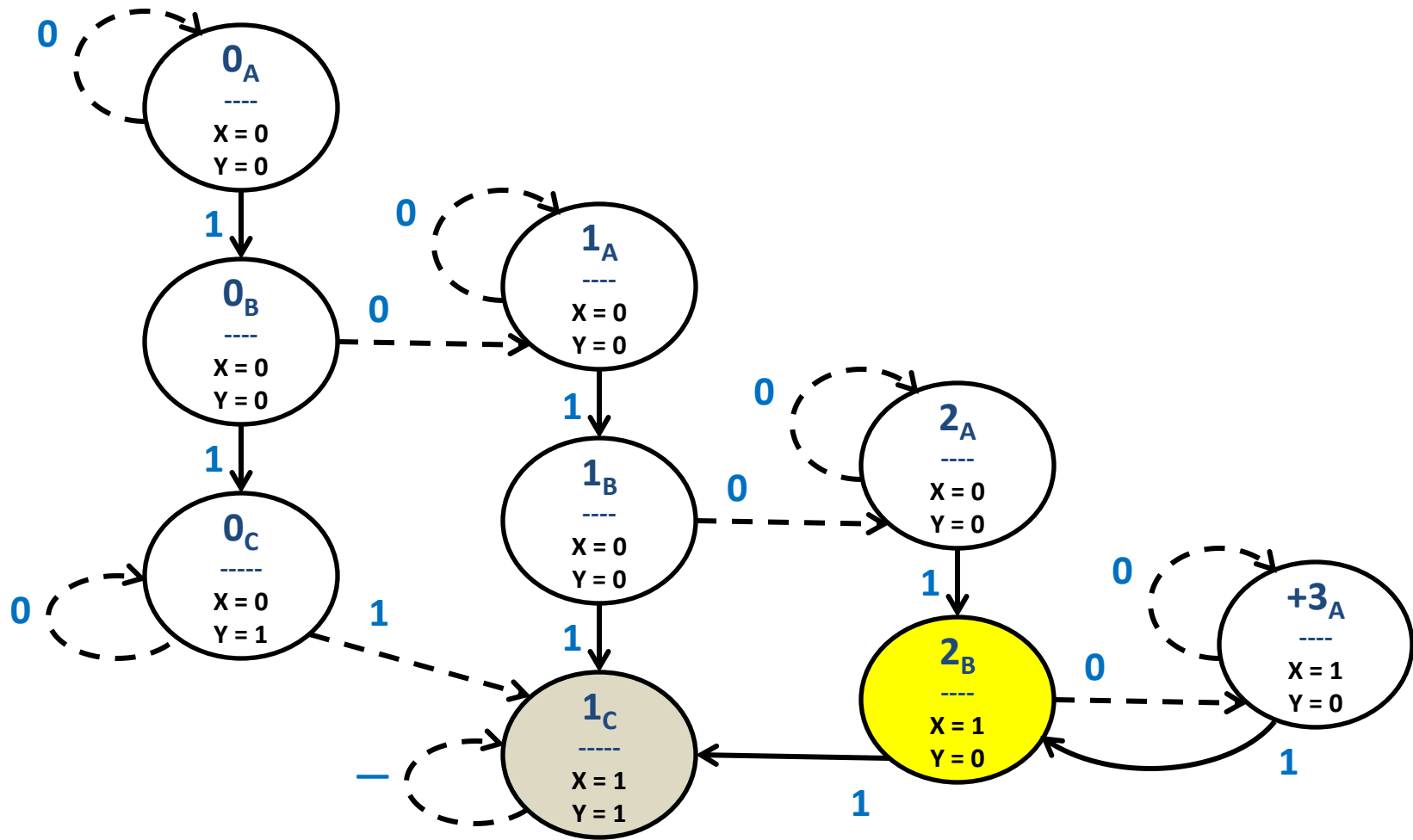


Common Mistake in Midterm2

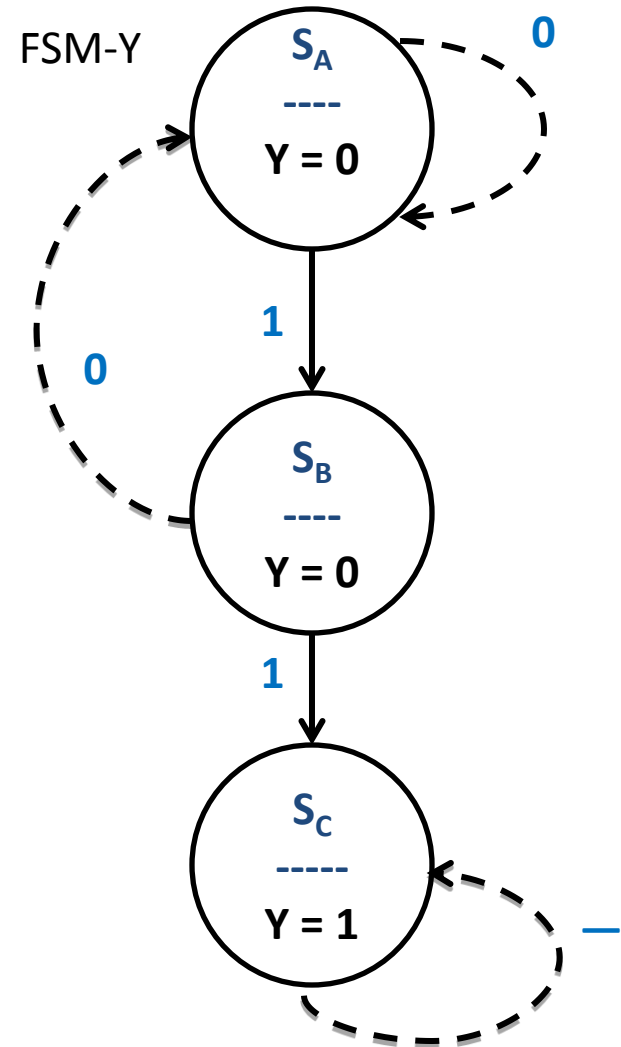
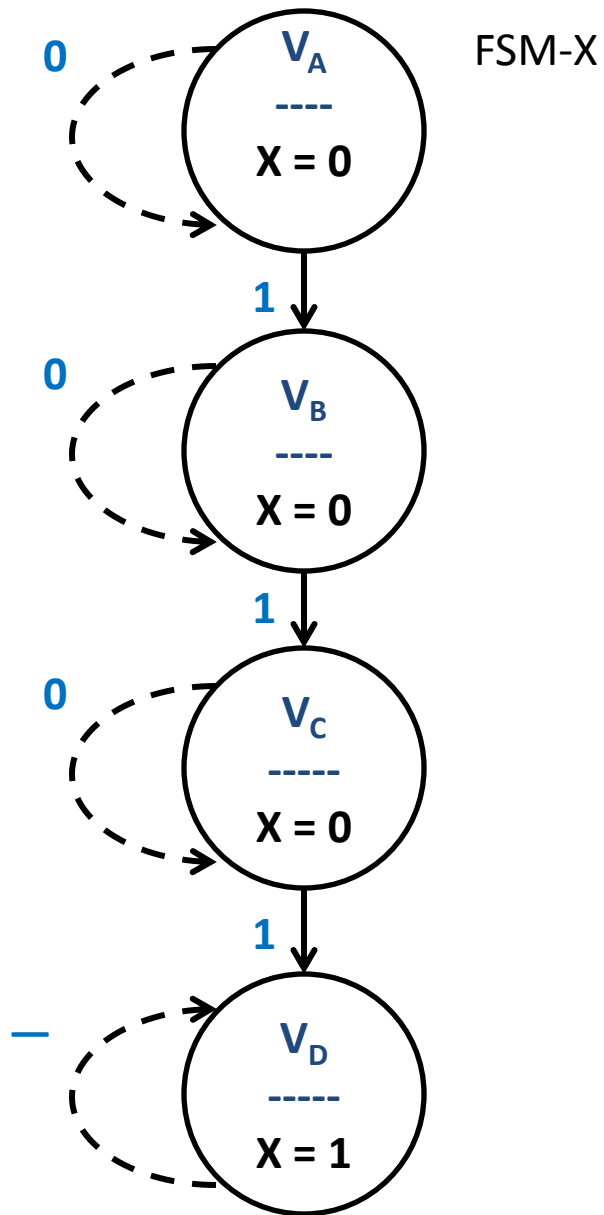
- Many got $B = C = F$, but didn't get $E = G$
- Their state table looked like this.

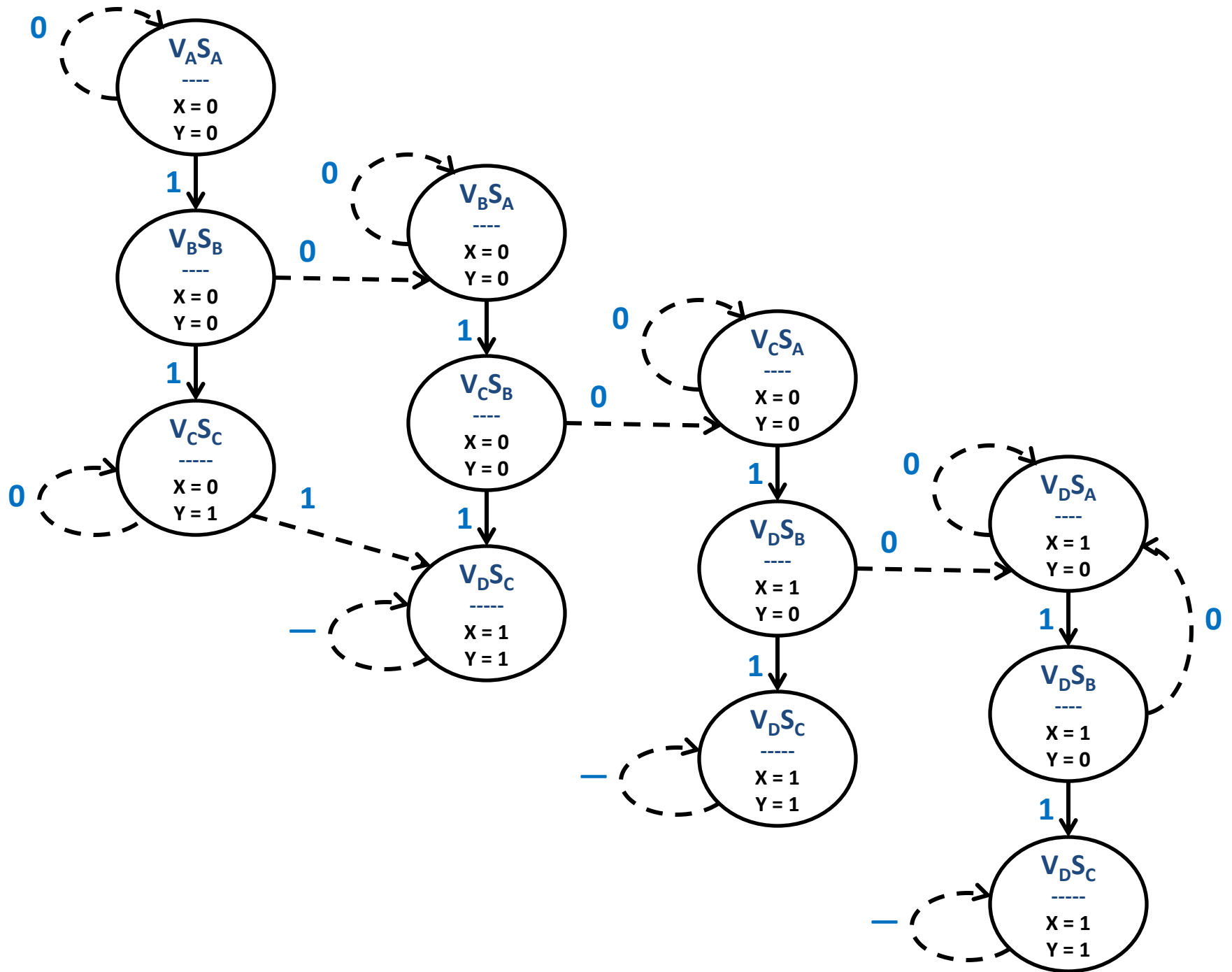
Explicit
Equivalence
 $E = G$.

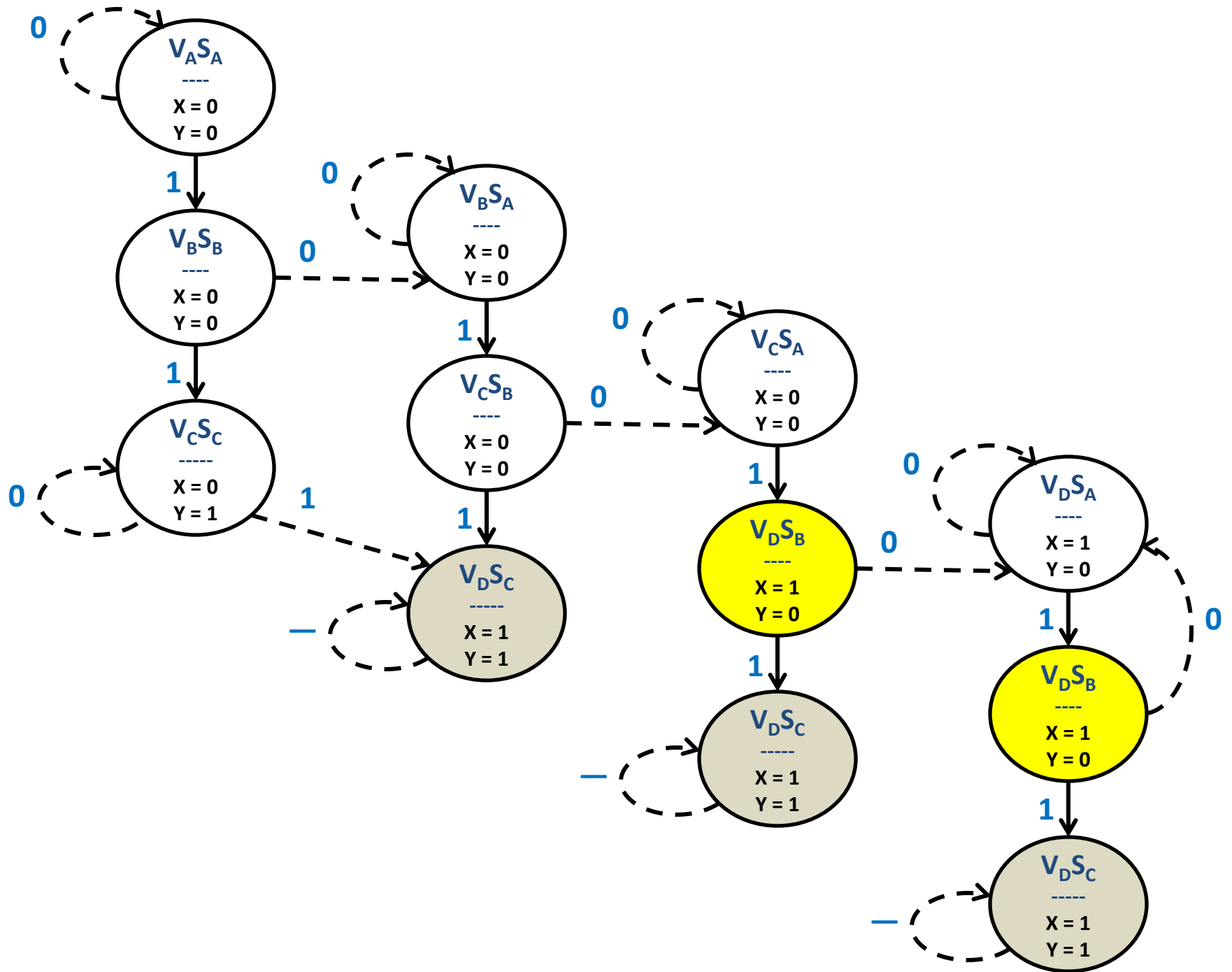
Present State	Next State		Output	
	$I=0$	$I=1$	$I = 0$	$I = 1$
A	A	B	0	0
B	B	E	1	1
D	B	A	0	0
E	B	D	1	0
G	B	D	1	0

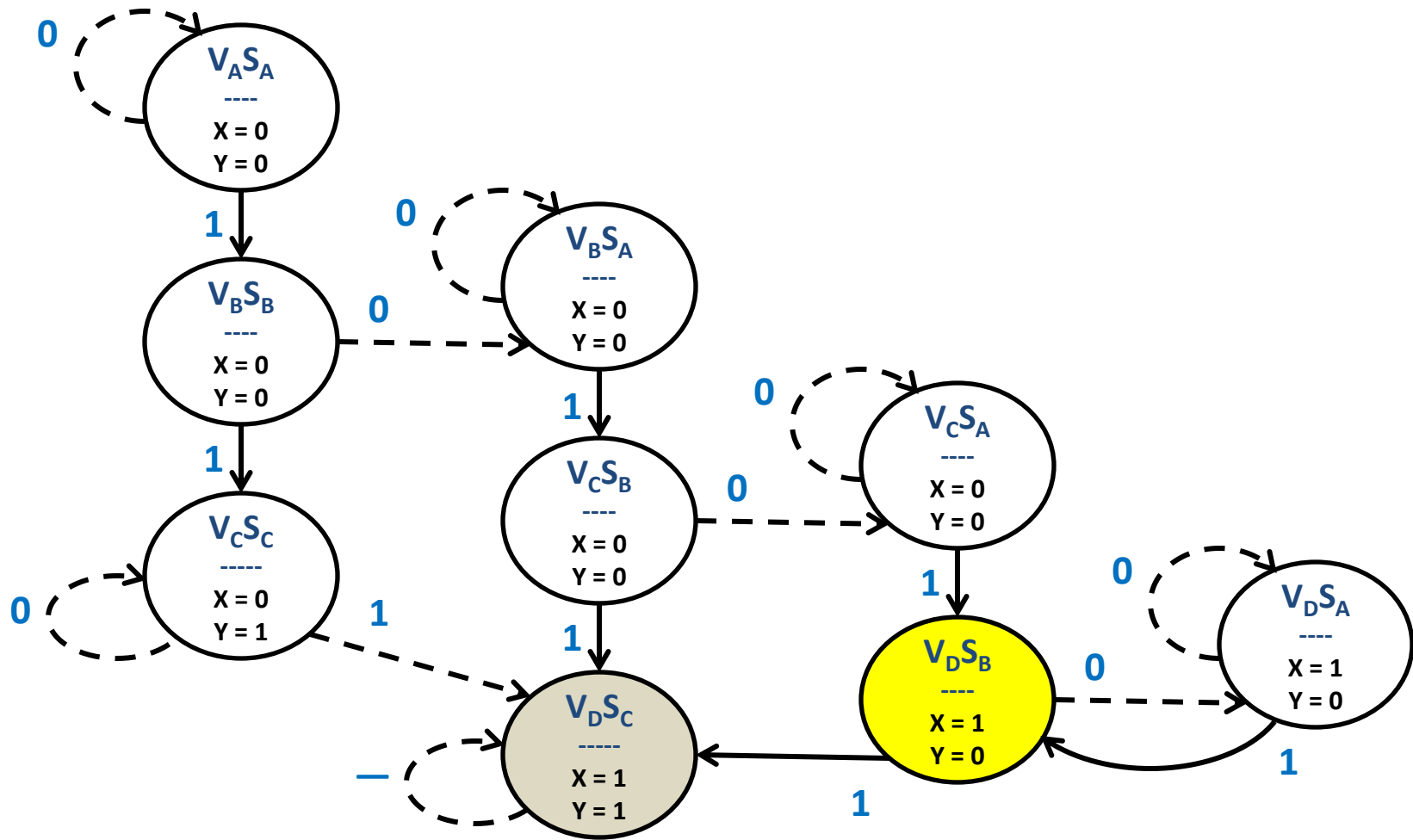


- Another approach is to design an FSM for X (FSM-X) and a separate FSM for Y (FSM-Y)
- Then “simulate” the execution from the “initial states”

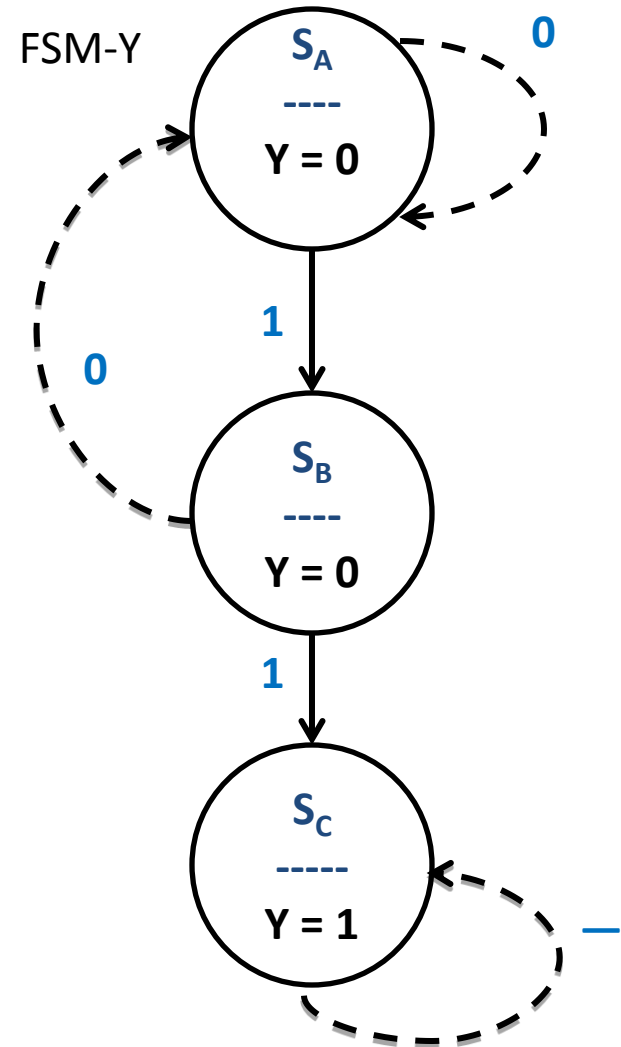
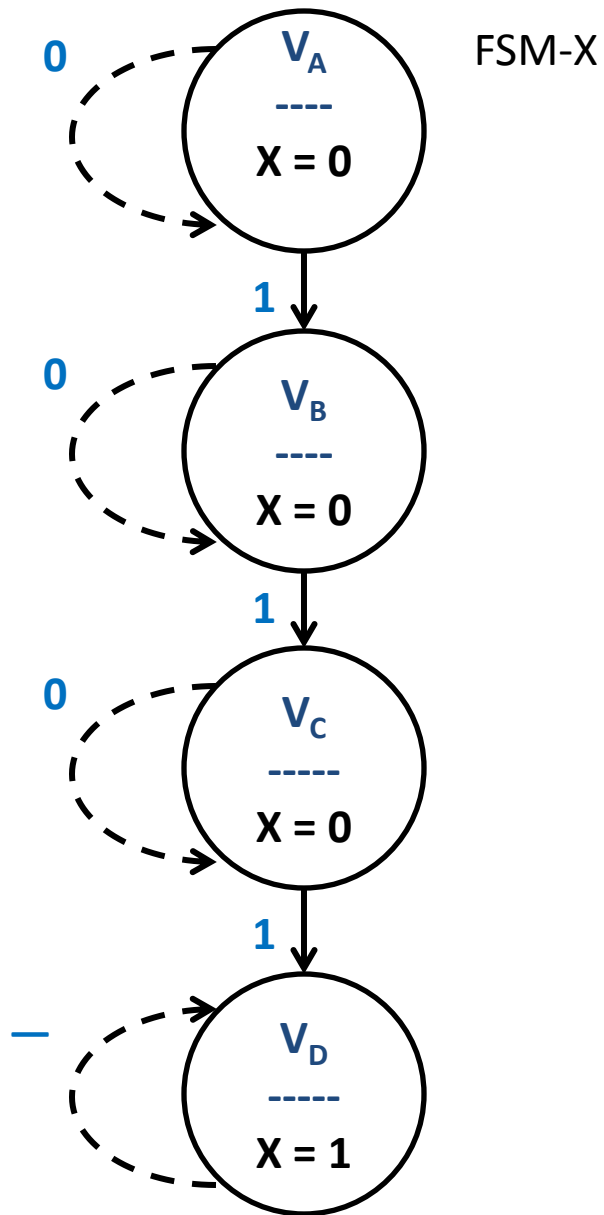








- Can also synthesize an FSM for X (FSM-X) and an FSM for Y (FSM-Y) separately (although this is different than what's asked in this question)



State transition and output table, **K-Maps for X, V1+, V0+** (for FSM-X)

Present State V1 V0	A = 0	A = 1	X
	NS V1+ V0+	NS V1+ V0+	
V _A = 0 0	0 0	0 1	0
V _B = 0 1	0 1	1 0	0
V _C = 1 0	1 0	1 1	0
V _D = 1 1	1 1	1 1	1

V1+ A

V1 V0 \ A	0	1
00	0	0
01	0	1
11	1	1
10	1	1

$V1+ = V1 + V0.A$

V0+ A

V1 V0 \ A	0	1
00	0	1
01	1	0
11	1	1
10	0	1

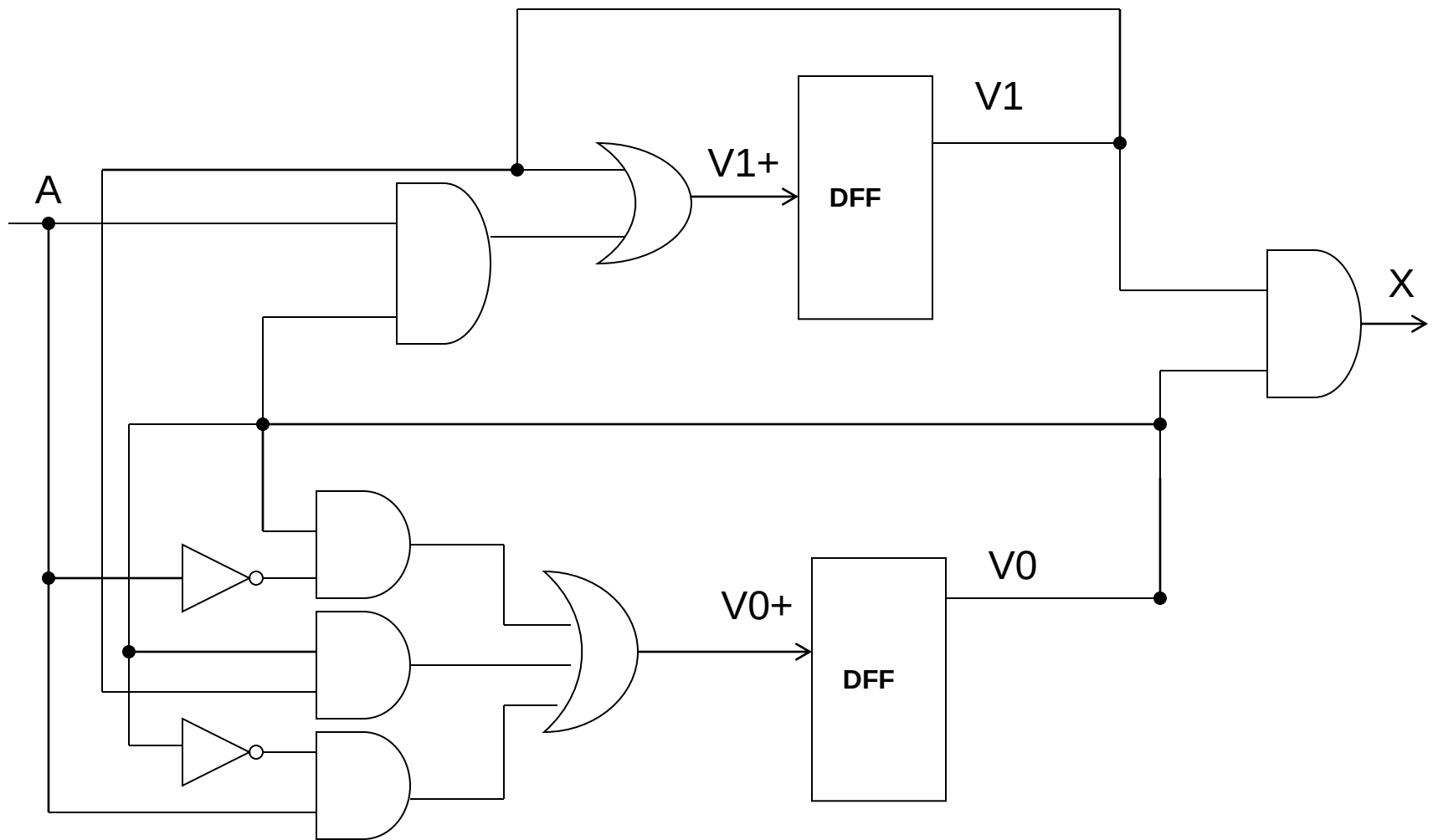
$V0+ = V0.A' + V1V0 + V0'.A$

Y V0

V1 \ V0	0	1
0	0	0
1	0	1

$Y = V1.V0$

Schematic for X, V1, V0



State transition and output table, **K-Maps for Y, S1+, S0+** (for FSM-Y)

Present State S1 S0	A = 0	A = 1	Y
	NS S1+ S0+	NS S1+ S0+	
S _A = 0 0	0 0	0 1	0
S _B = 0 1	0 0	1 0	0
S _C = 1 0	1 0	1 0	1

S1+ A

S1 S0 \ A	0	1
00	0	0
01	0	1
11	X	X
10	1	1

S1+ = S1 + S0.A

S0+ A

S1 S0 \ A	0	1
00	0	1
01	0	0
11	X	X
10	0	0

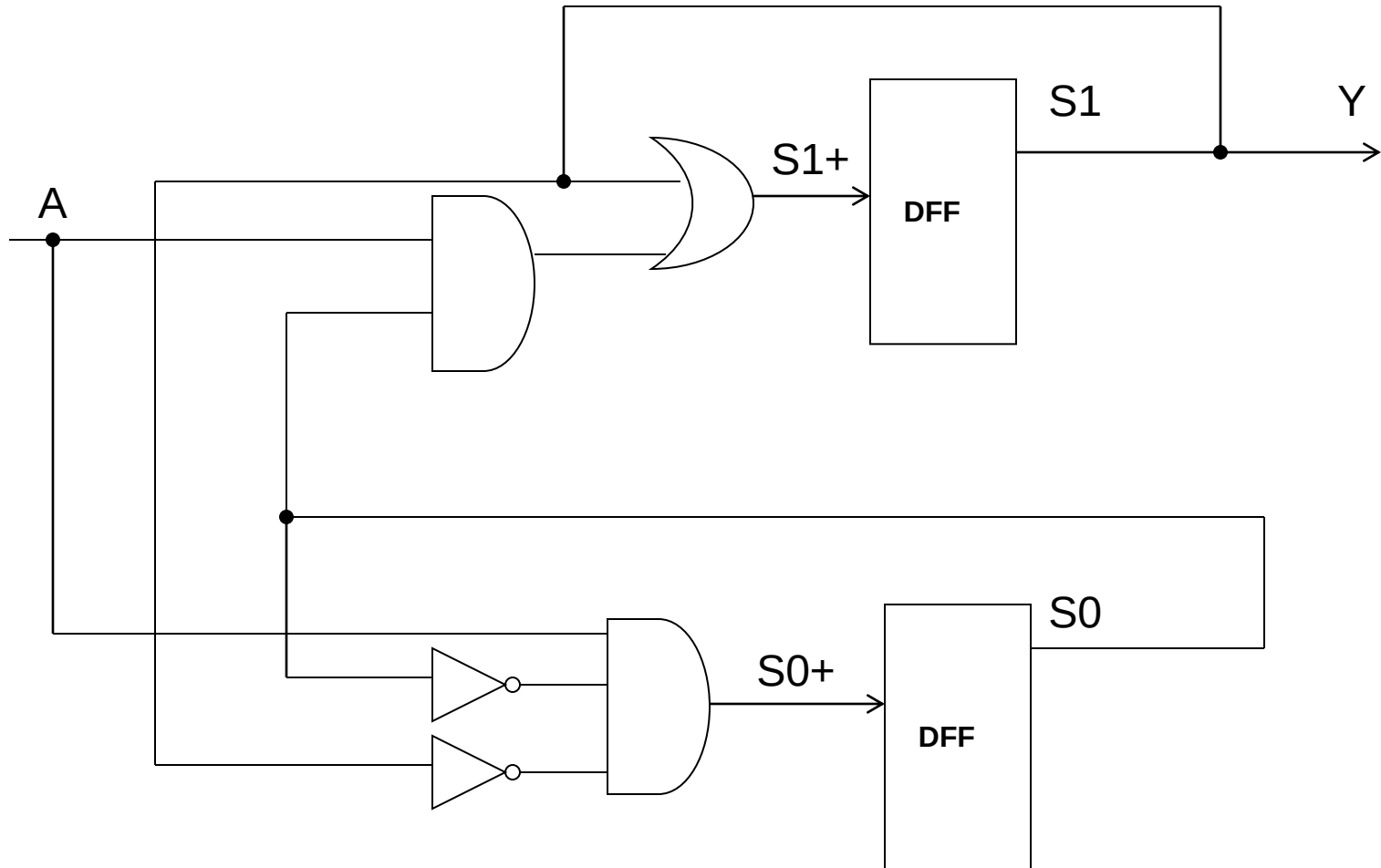
S0+ = A.S0'.S1'

Y S0

S1 \ S0	0	1
0	0	0
1	1	X

Y = S1

Schematic for Y, S1, S0



State transition and output table, **K-Maps for Y, S1+, S0+** (for FSM-Y) :
Using a different State Assignment for Sc. (Using 11 instead of 10)
This helps in reducing the number of literals required to compute S0+. (requires 2 literals instead of 3)
Efficient State Assignment problem is sometimes taken care of by EDA tools.

Present State S1 S0	A = 0	A = 1	Y
	NS S1+ S0+	NS S1+ S0+	
S _A = 0 0	0 0	0 1	0
S _B = 0 1	0 0	1 1	0
S _C = 1 1	1 1	1 1	1

S1+ A

S1 S0 \ A	0	1
00	0	0
01	0	1
11	1	1
10	X	X

S1+ = S1 + S0.A

S0+ A

S1 S0 \ A	0	1
00	0	1
01	0	1
11	1	1
10	X	X

S0+ = S1 + A

Y S0

S1 \ S0	0	1
0	0	0
1	X	1

Y = S1

