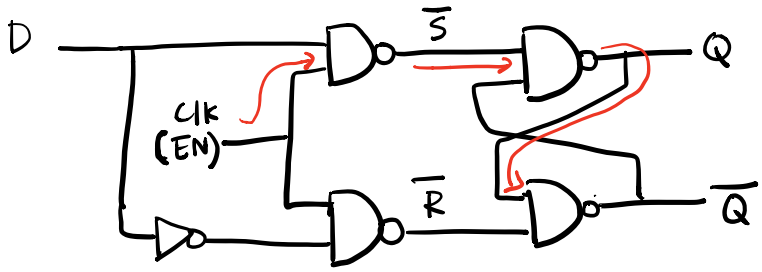
#1. [D-latch timing]

D-latch implemented w/ NAND gates:

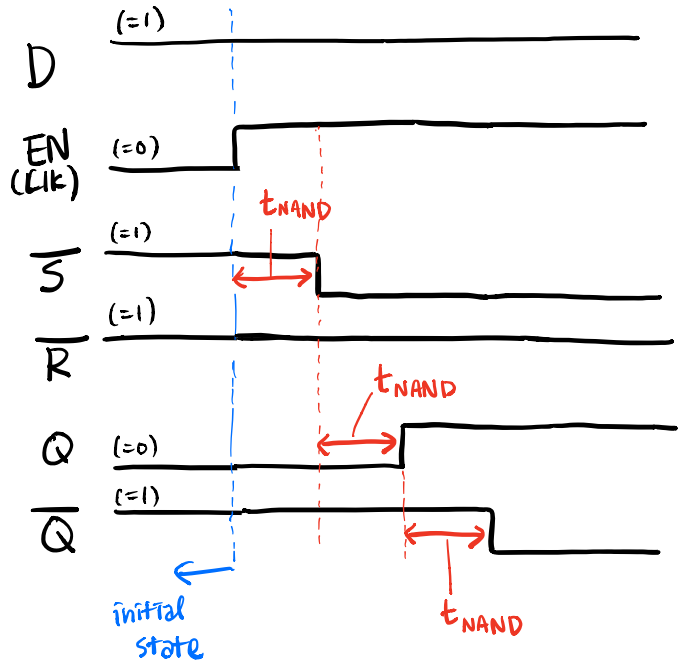


Note:

$(\overline{A}+\overline{B}) = \overline{AB}$

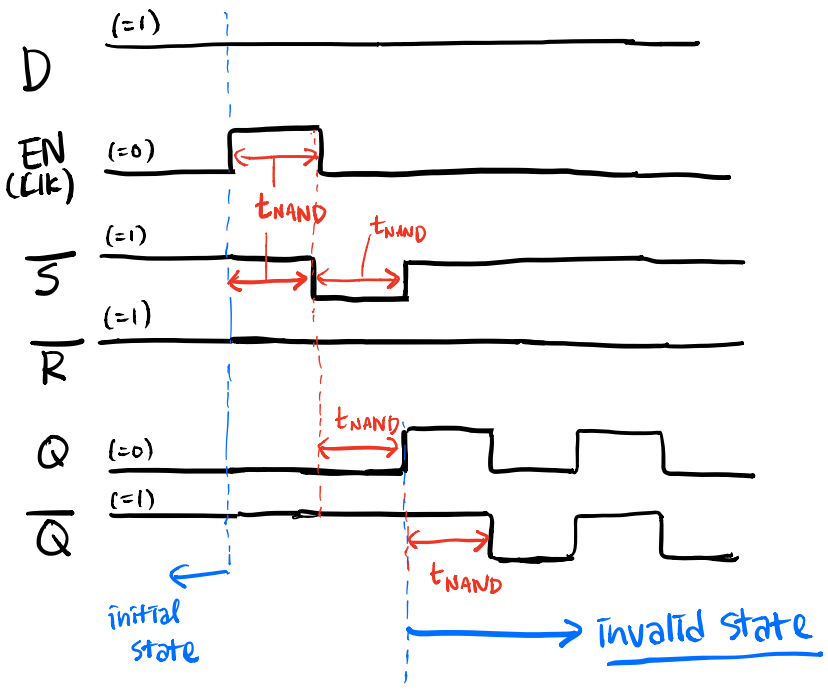
(De Morgan's law)

(a)



∴ takes $3t_{NAND}$ for all outputs to stabilize.

(b)

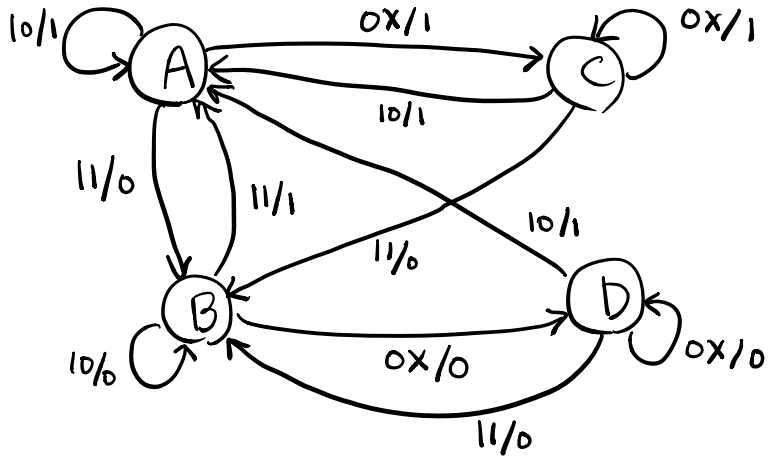


∴ the D-latch would end up in an invalid state

#2. [Finite State Machine]

(a) This is a Moore machine, because the output only depends on the state that it is in. (current output does not depend on current input.)

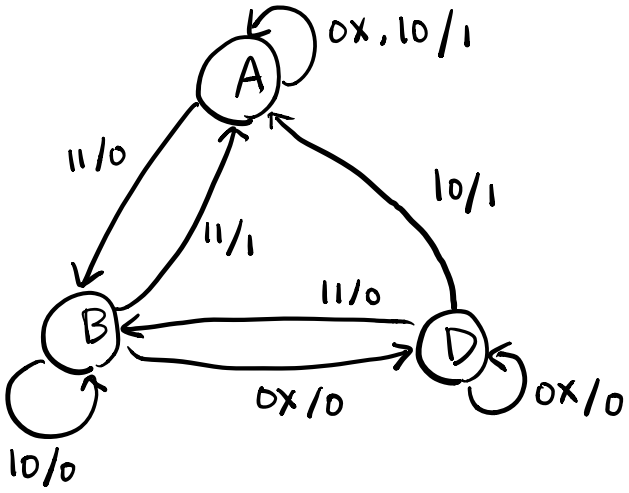
(b) input is noted as SD (SYNC. DATA)



state	(input) SYNC	DATA	next state	out
A	0	X	C	1
A	1	0	A	1
A	1	1	B	0
B	0	X	D	0
B	1	0	B	0
B	1	1	A	1
C	0	X	C	1
C	1	0	A	1
C	1	1	B	0
D	0	X	D	0
D	1	0	A	1
D	1	1	B	0

A, C are same states

reduced state diagram



reduced state table

State		Input SYNC	Input DATA	next state	out
00	A	0	X	A	1
00	A	1	0	A	1
00	A	1	1	B	0
01	B	0	X	D	0
01	B	1	0	B	0
01	B	1	1	A	1
11	D	0	X	D	0
11	D	1	0	A	1
11	D	1	1	B	0

(c) ① encode the states : A = 00 , B = 01 , D = 11

(there are many other ways to encode states)

(state $T = T_1 T_0$)

② draw K-map for each output (T_1, T_0, out)

T_1 :

T_1T_0 \ SD	00	01	11	10
00				
01	1	1		
11	1	1		
10				

$$T_{1(next)} = \overline{S}T_0$$

T_0 :

T_1T_0 \ SD	00	01	11	10
00			1	
01	1	1		1
11	1	1	1	
10				

$$T_{0(next)} = \overline{S}T_0 + T_1T_0D + \overline{T_1}T_0\overline{D} + SD\overline{T_1}\overline{T_0}$$

out :

T_0 \ T_1	0	1
0	1	X
1	0	0

$$out = \overline{T_0}$$