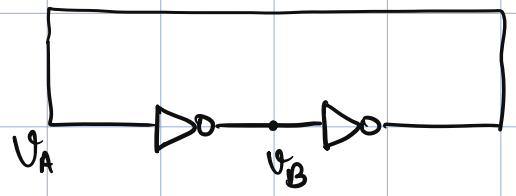
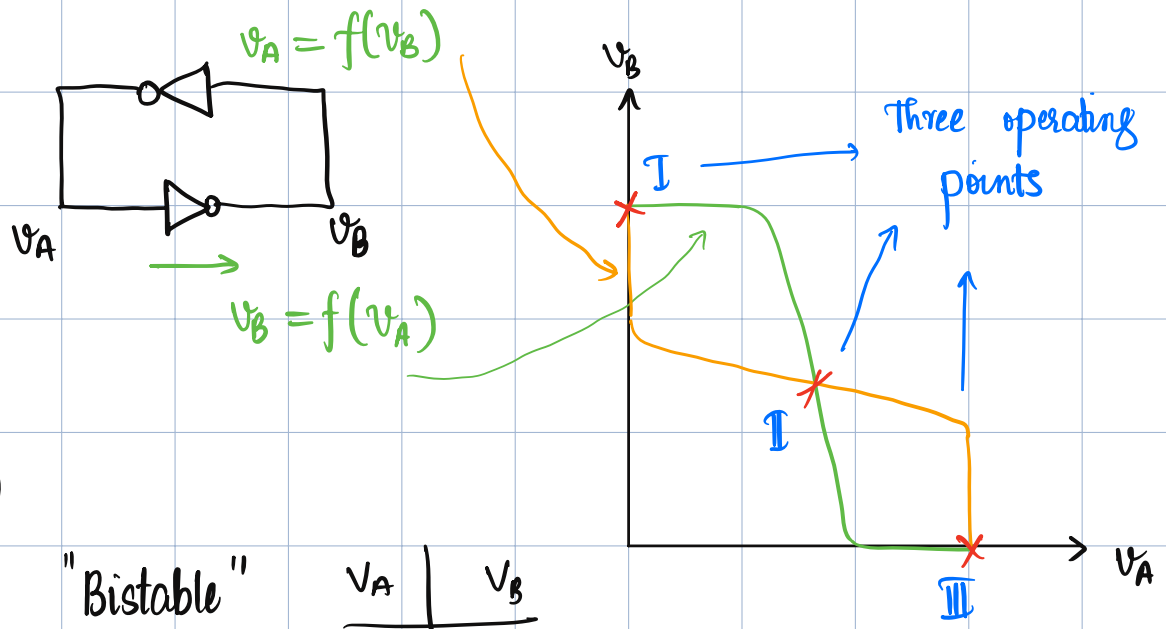
2024/10/21 - Digital Circuits - Week 12



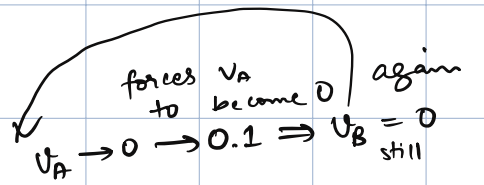
|||



Latch

"Bistable"

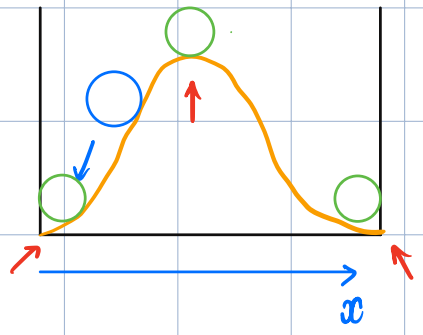
V_A	V_B
0	1
1	0



Quality of solutions:
How good are the operating points?

electrical interference may cause V_A or V_B to drift

At these three points signals will remain consistent
(V_B will not force V_A to change and vice versa)



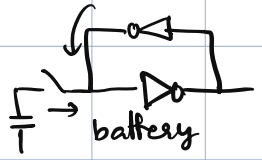
Three solutions are possible

(II) → unstable operating point

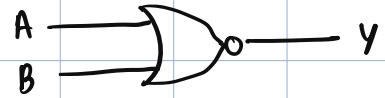
- can be used as storage element.
- how to make the storage useful / dependent on user input?

ideal capacitor
↳ can store ∞ charge

battery ≡ ideal capacitor

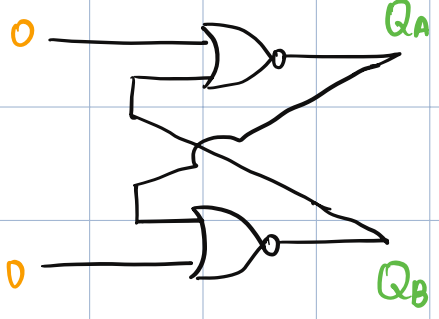


stronger battery

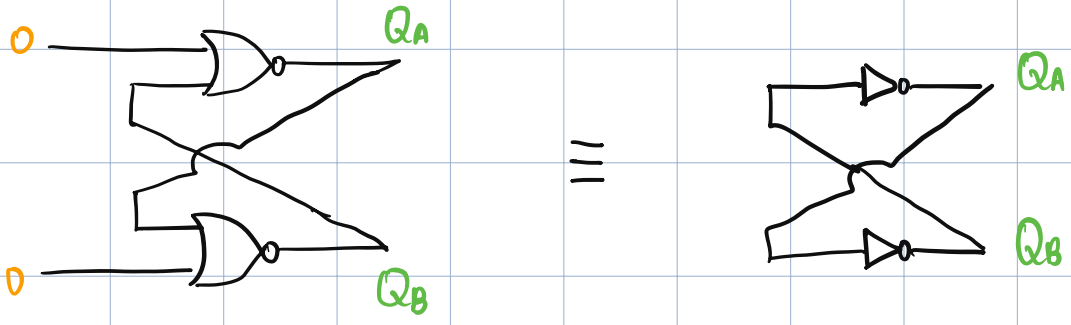


A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

$Y = \bar{B}$



2024|10|23



S	R	Q_A	Q_B	
0	0	Q_A^*	Q_B^*	last state
0	1	1	0	set state
1	0	0	1	reset state
1	1	0	0	undefined state

↳ invalid input (may result in meta-stability)

01

$Q_B = \overline{1+Q_A} = 0$

$Q_A = \overline{0+0} = 1$

$Q_B = \overline{1+Q_A} = 0$

$Q_A = \overline{1+Q_B} = 0$

→ NOR gate:

controlled inverter

$S = 0 = R$

1 0 → stays the same after applying $S=R=0$

or 0 1

Using first three rows, you can use the circuit as a controlled memory

→ state : set of variables in a digital circuit whose value at present instant is sufficient to predict the current O/P based on current input.

→ A digital circuit with n state variables : finite no. of states.

Q_A^* Q_B^* (1 0 or 0 1)

4^{th} row → 1^{st} row

→ unknown state (1 0 or 0 1)

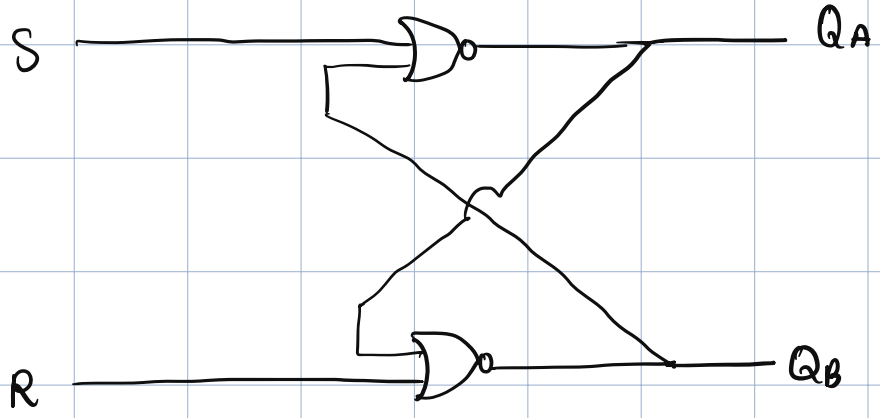
→ Avoid 4^{th} input

→ Assume output is not defined (it is, but assume)

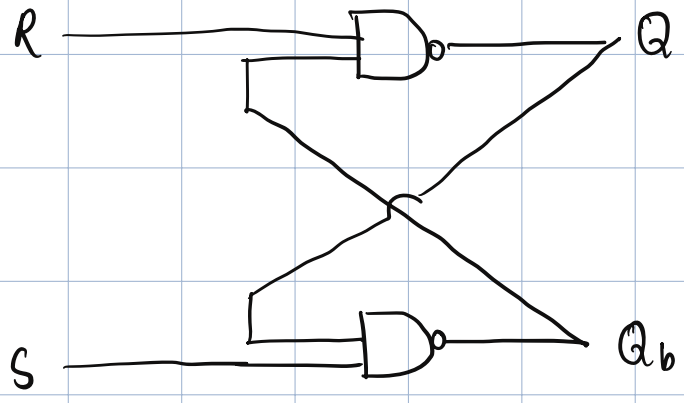
↓

fluctuation after changing R & S from (1, 1) to (0, 1) or (1, 0)

2024|10|24

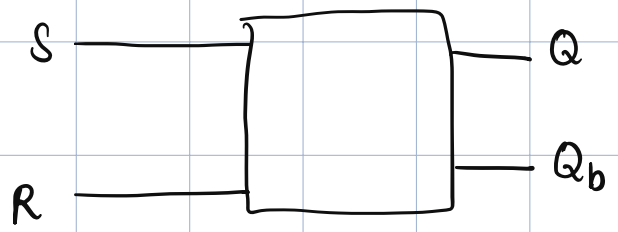


S	R	Q_A	Q_B	
0	0	Q_A^*	Q_B^*	valid inputs for SR latch
0	1	0	1	
1	0	1	0	
1	1	0	0	← Not desired

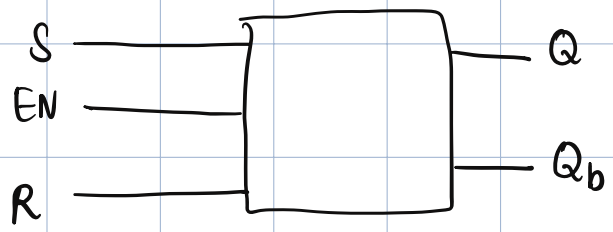


$\bar{S}$-$\bar{R}$ latch

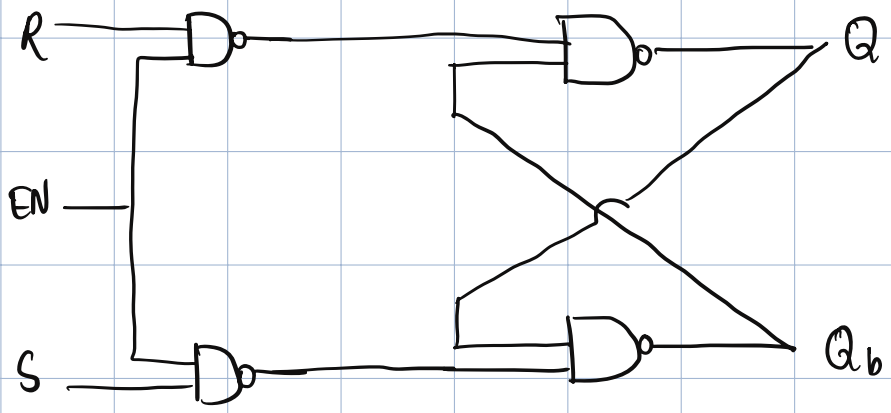
S	R	Q	Q_B
0	0	1	1
0	1	0	1
1	0	1	0
1	1	Q_A^*	Q_B^*



SR Latch

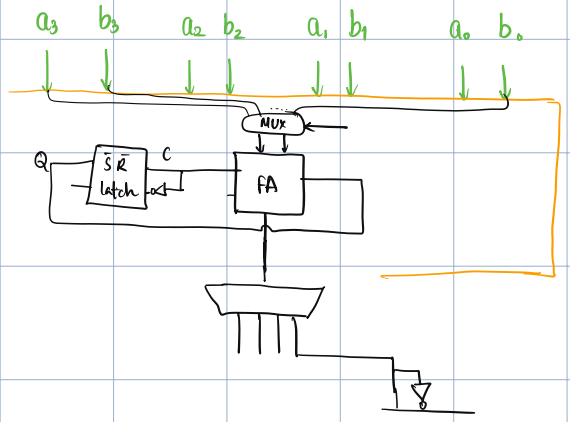


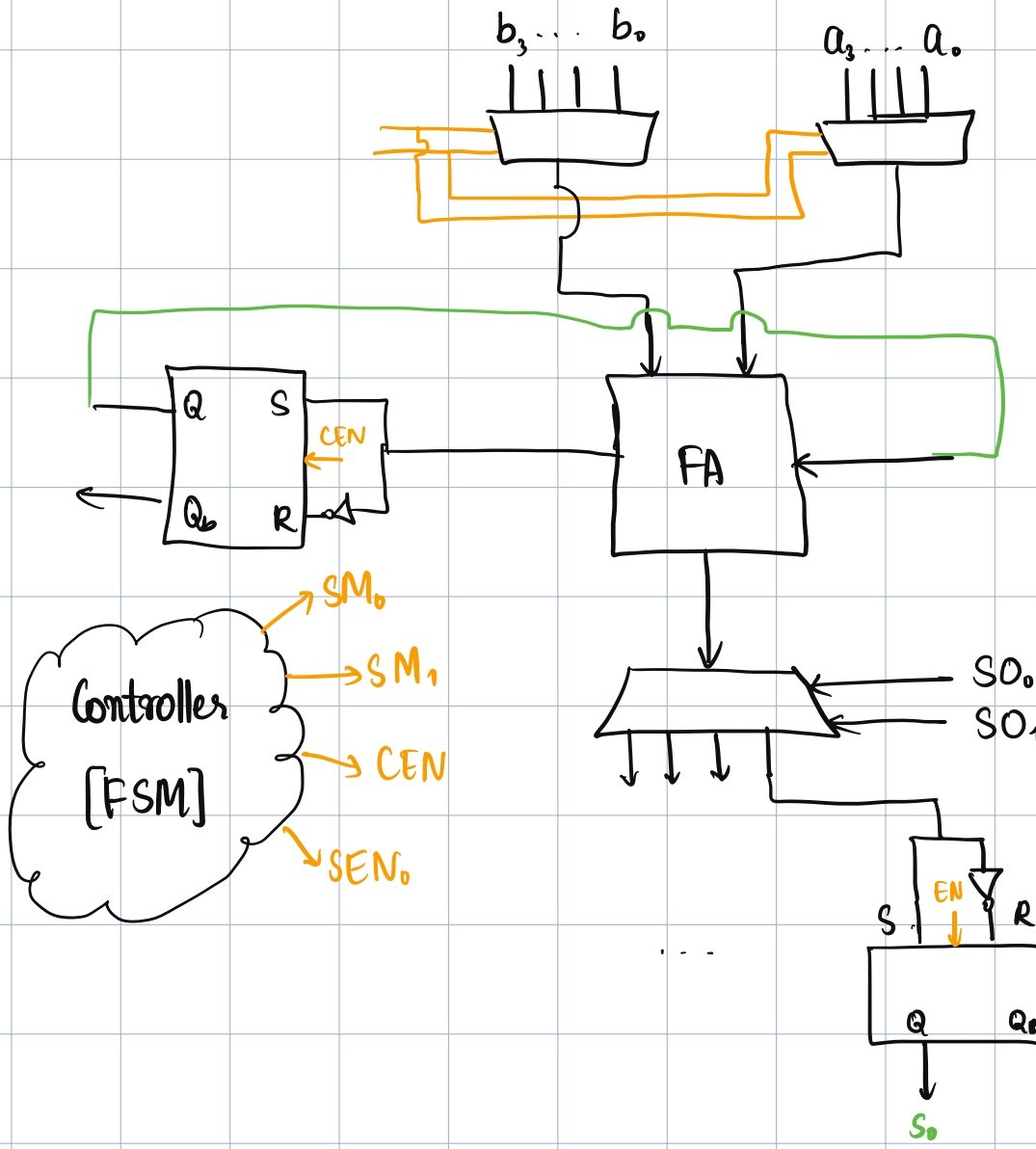
SR Latch with enable



→ used in caches

Ex:- 4 bit adder using only one instance of FA and latches, MUX, decoder





← Finite state machine