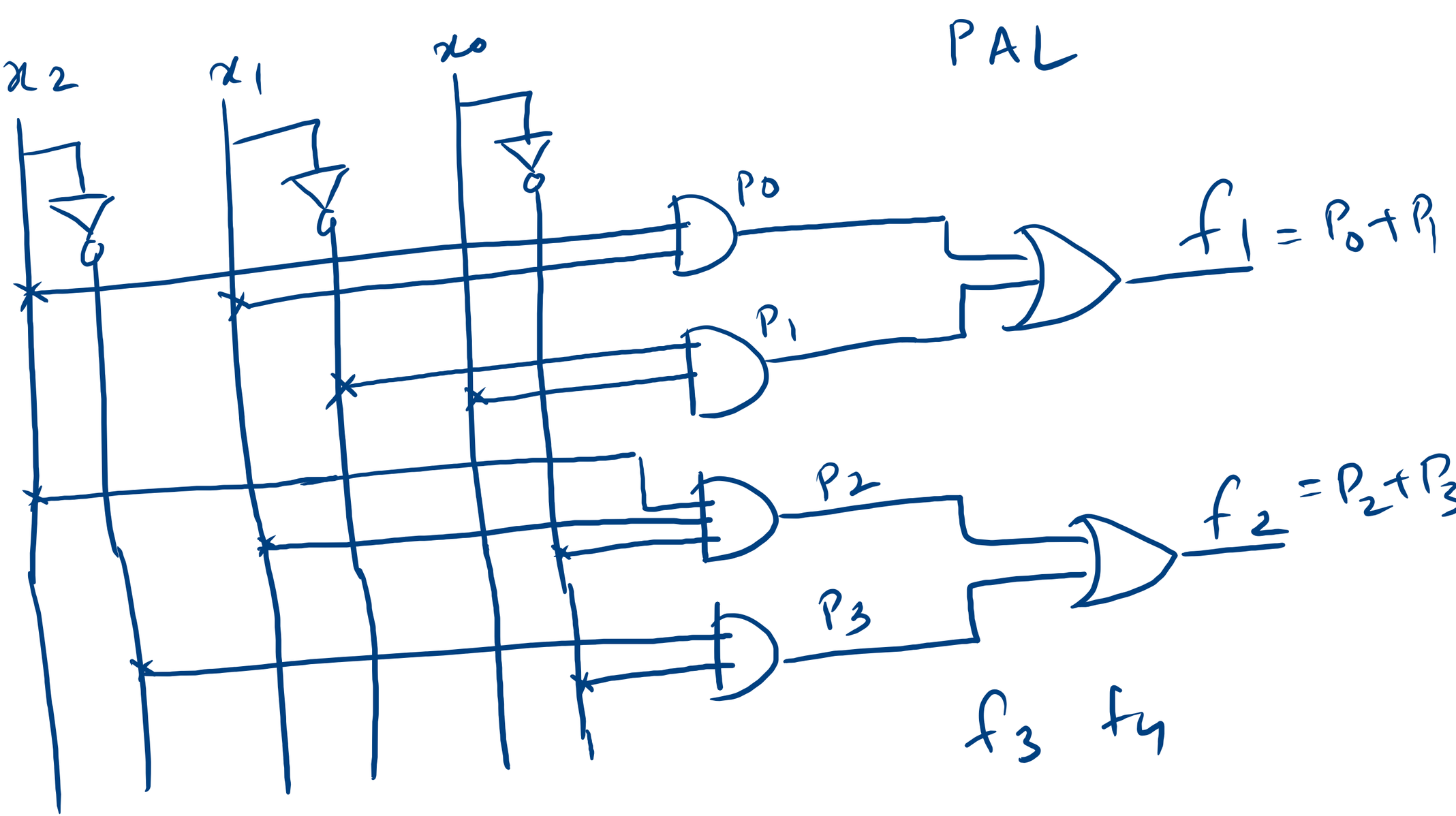
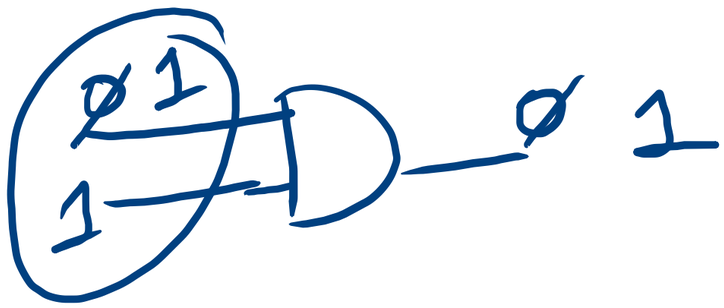
Programmable Array Logic

1. Programmable AND array
2. Fixed OR array

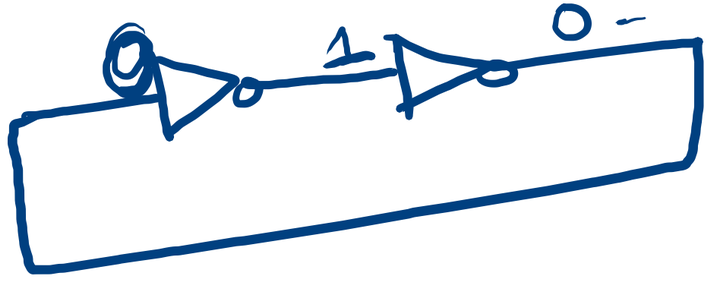
PAL



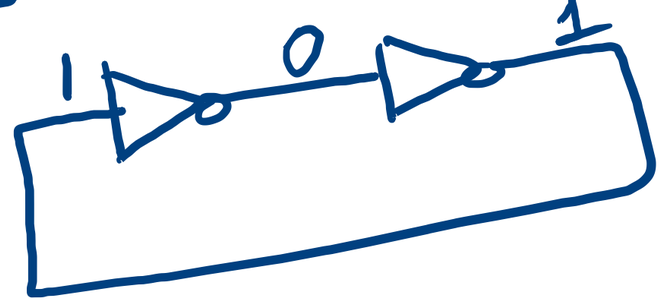
Sequential circuits



Memory → how to store bits electronically



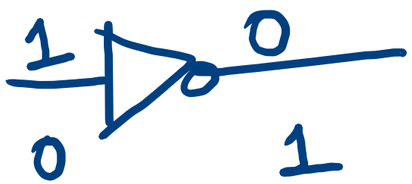
store a 0



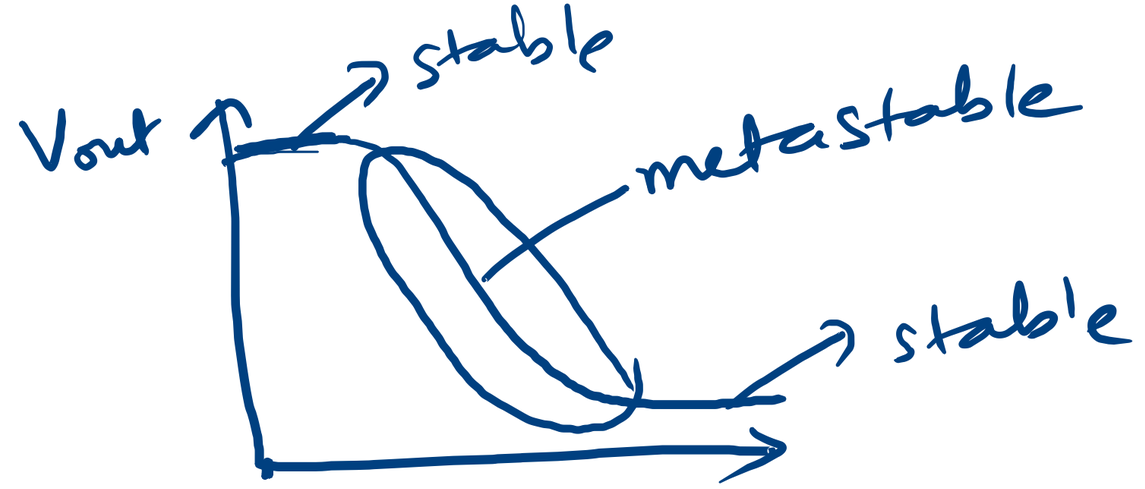
store a 1

Latches and flipflops

Memory elements used to store informat°



Unstable



Bistable
device
multivibrator

Latch

Both 0 & 1

2 i/ps & 2 O/ps

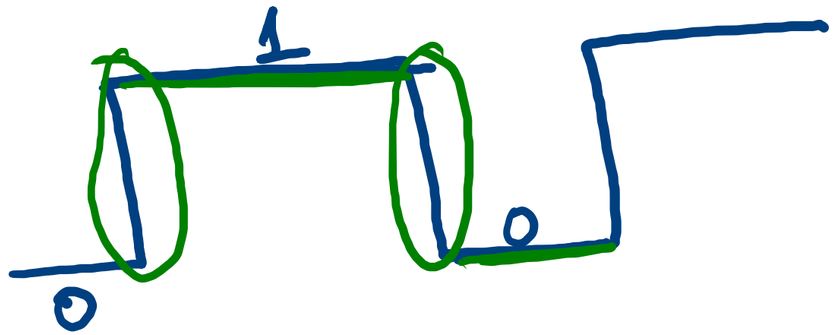
① SR (Set - Reset)

② D type (Delay)

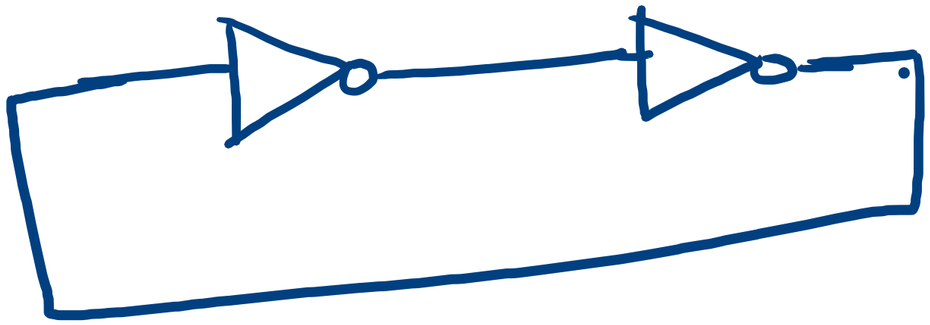
③ JK

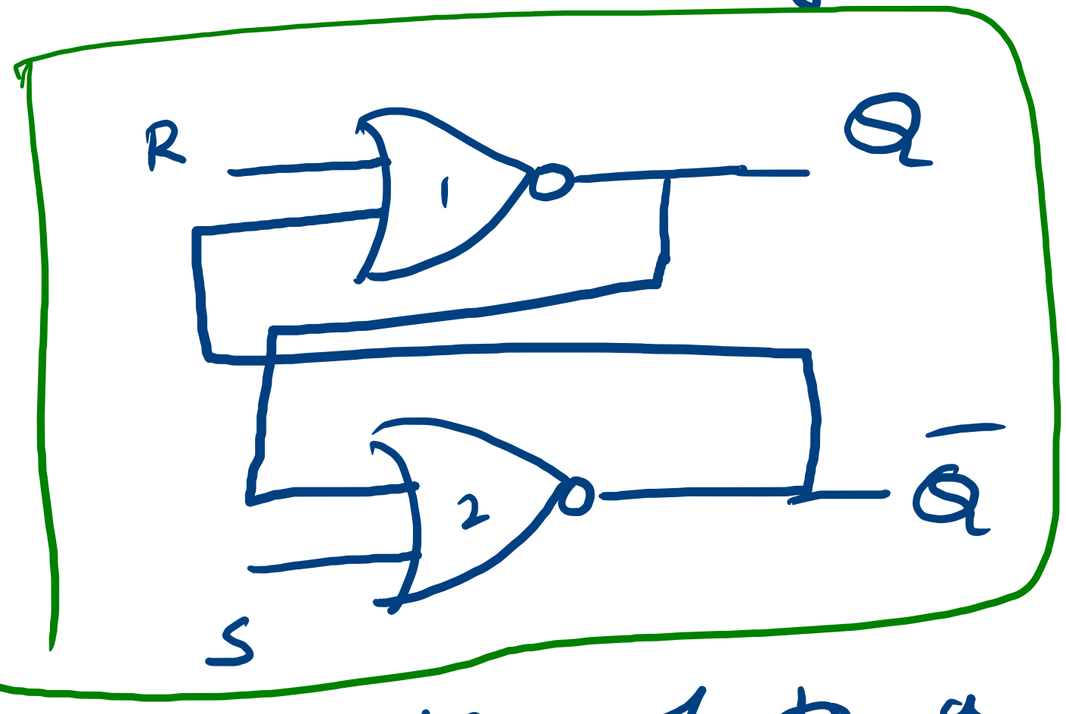
④ Toggle/ T type.

Level sensitive

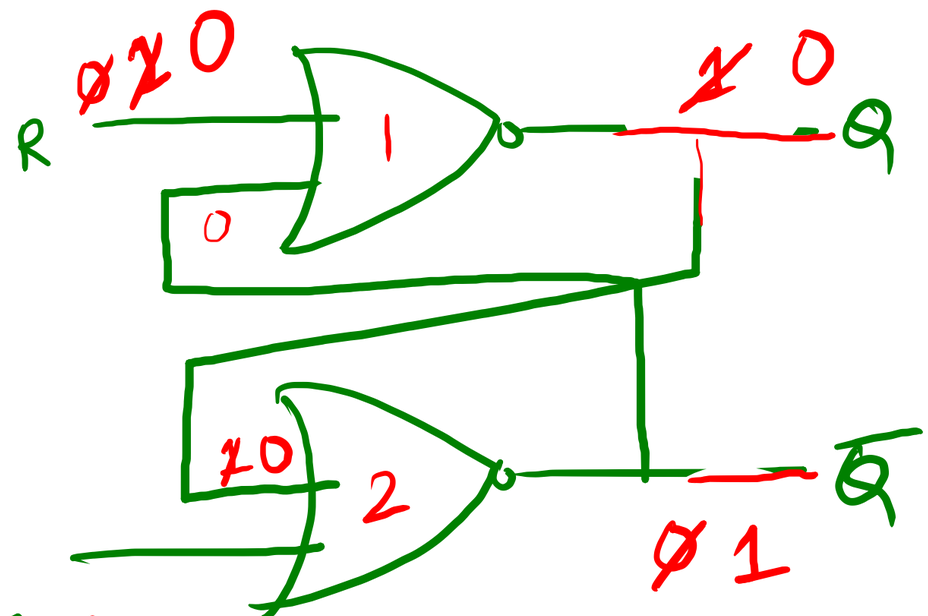


Latch is a pair of cross coupled NOR or NAND gates





force the o/p to a specific value by applying appropriate i/ps at R & S.



TT for NOR

A	B	F
X	X	
0	0	1
X	X	
0	1	0
1	0	0
1	1	0

$S = 1, R = 0, Q = 1, \bar{Q} = 0 \rightarrow$ set o/p to 1

$S = 0, R = 0, Q = 1, \bar{Q} = 0 \rightarrow$ no change (store)

$S = 0, R = 1, Q = 0, \bar{Q} = 1 \rightarrow$ set o/p to 0

$S = 0, R = 0, Q = 0, \bar{Q} = 1 \rightarrow$ NC (store)

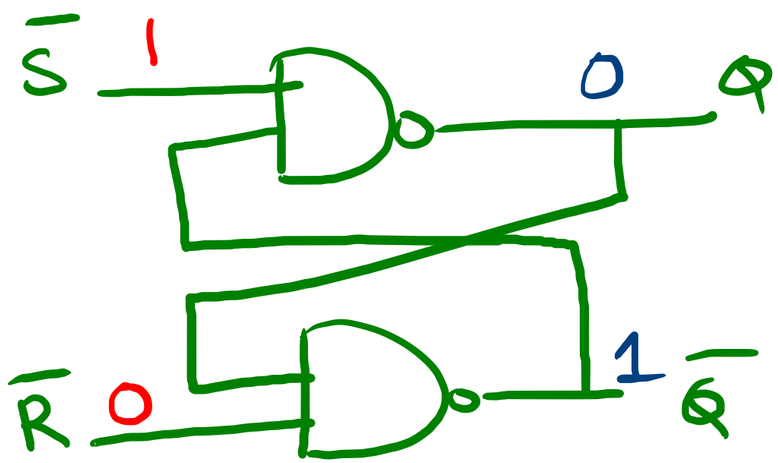
Truth table for SR latch

S	R	Q	$\bar{Q}$
0	0	NC	NC
0	1	0	1
1	0	1	0
1	1	invalid	invalid

NAND TT

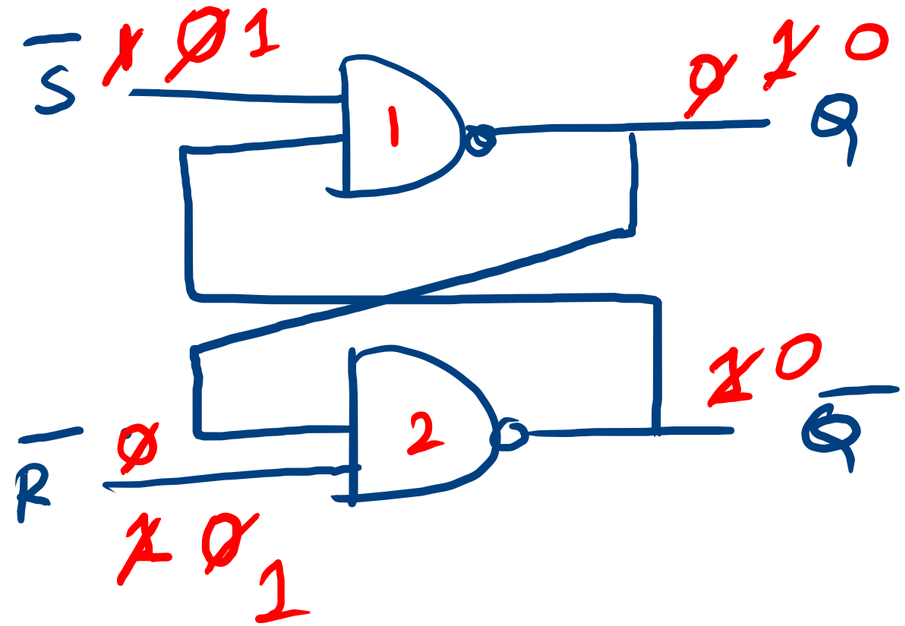
0	0	1 →
0	1	1
1	0	1
1	1	0

NAND implementation



Case → S = 0, R = 1

$\bar{S} = 1$, $\bar{R} = 0$, Q = 0, $\bar{Q} = 1$



$S = 0, R = 1$

$S = 1, R = 1$

$\bar{S} = 0, \bar{R} = 0$

$Q = 1, \bar{Q} = 1$

$\bar{S} = 1, \bar{R} = 1, Q = 0, \bar{Q} = 0$

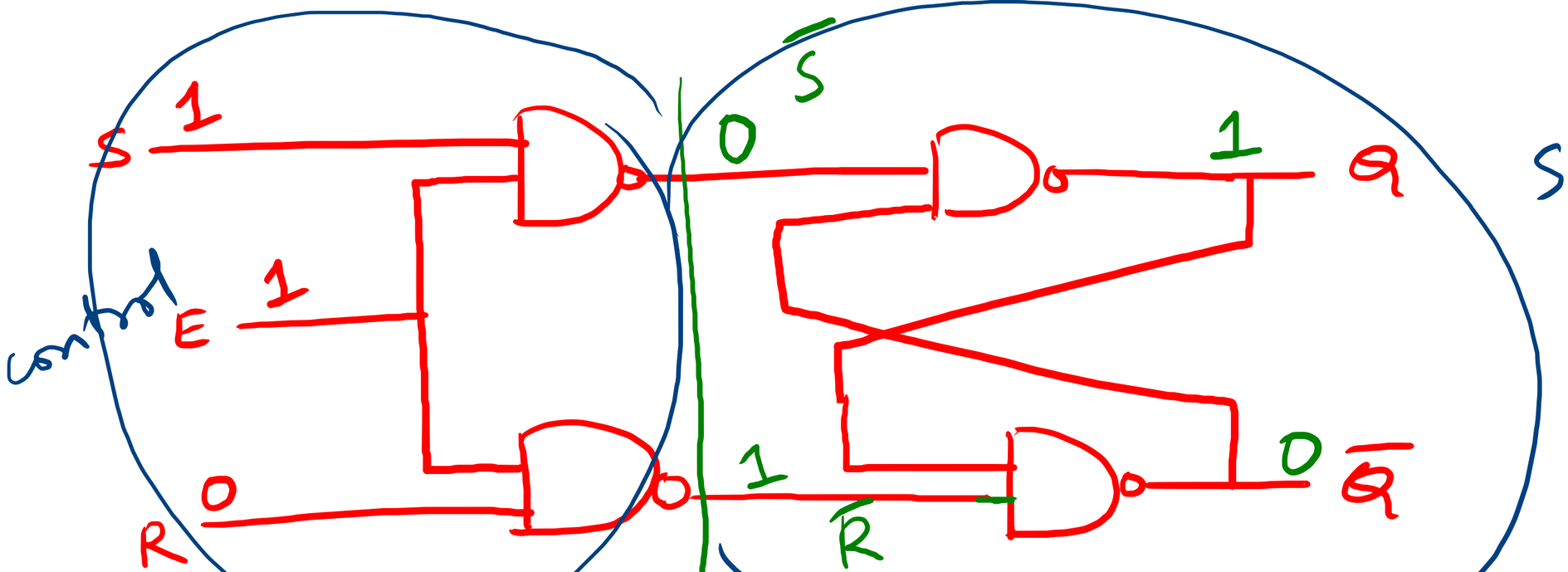
$S = 0, R = 0 \rightarrow$ store

Gated SR latch

Enable i/p E

When E = 1, the latch is active

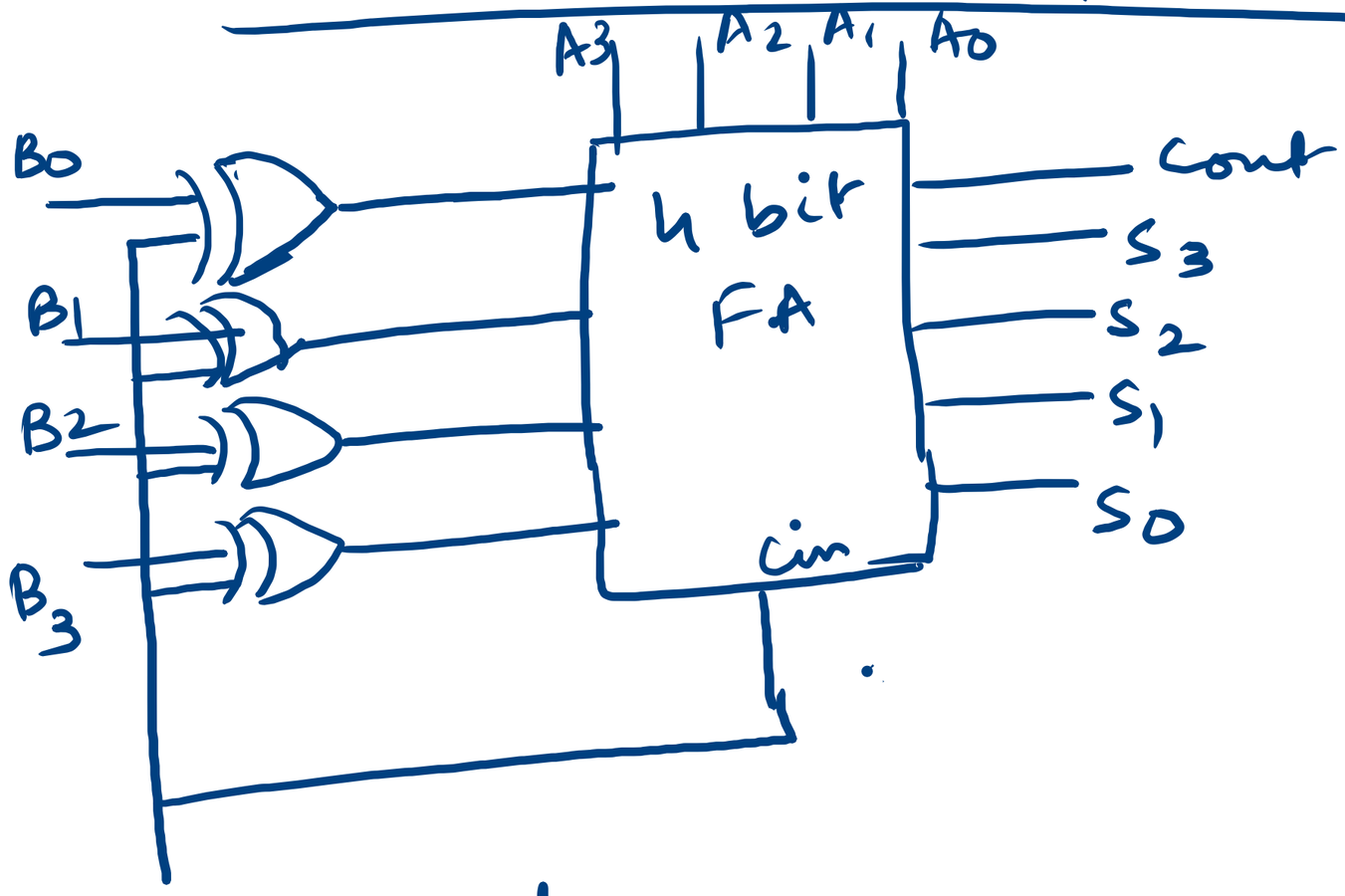
When E = 0, the latch is deactivated & o/p doesnot change.



SR NAND latch

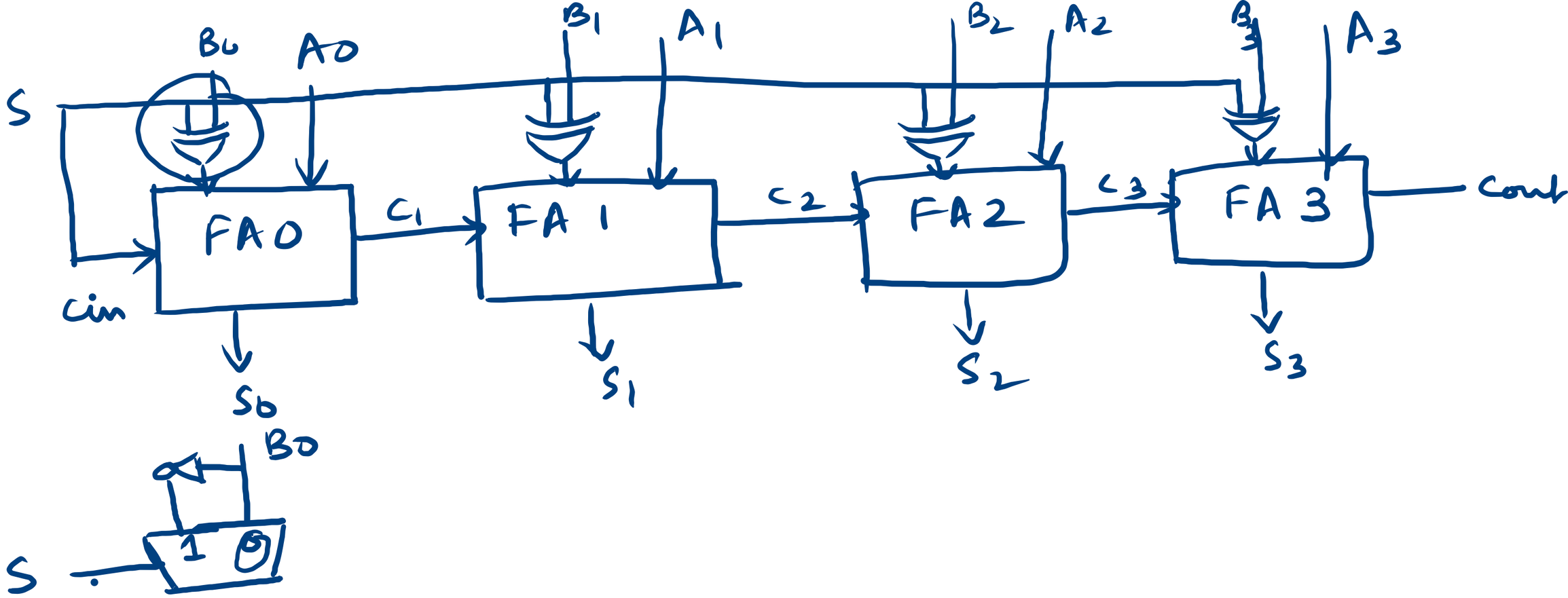
E	S	R	Q	$\bar{Q}$
0	x	x	NC	NC
1	0	0	NC	NC
1	0	1	0	1
1	1	0	1	0
1	1	1	invalid	invalid

Adder/ subtractor using FAs & MUX



S = 1 , sub

S = 0 , add



$S = 0, \quad B_0$

$S = 1 \quad - \overline{B_0}$

Alternate design for SR latch