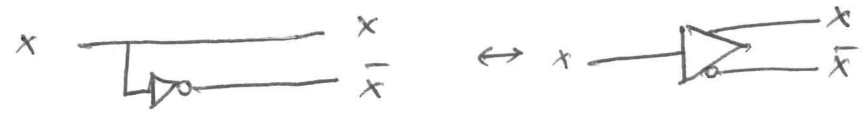
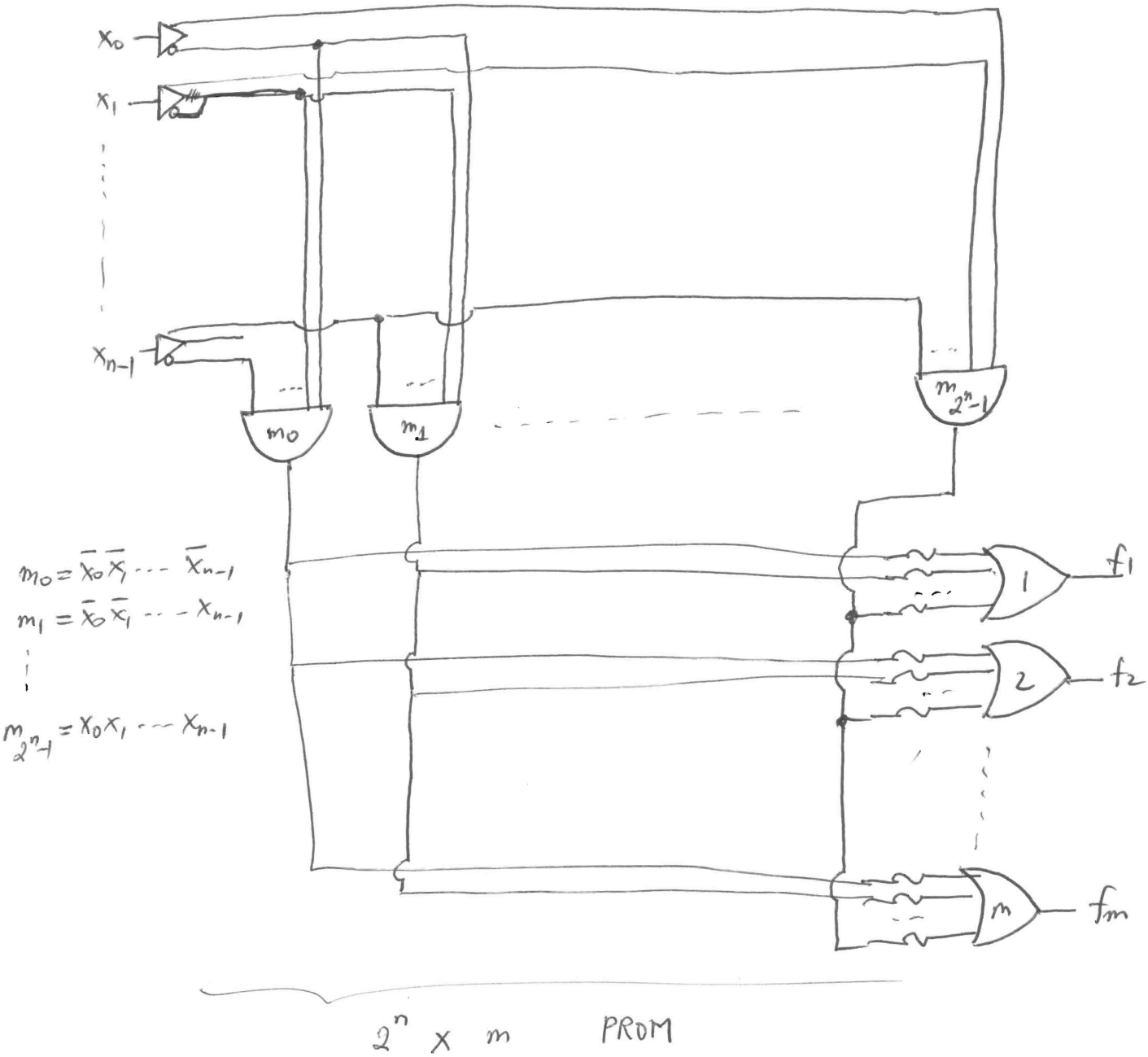
PROM (AND's are fixed ; OR's are programmable)

↓

AND gate version of decoders

Inputs : buffer (direct) & inverter (complemented)



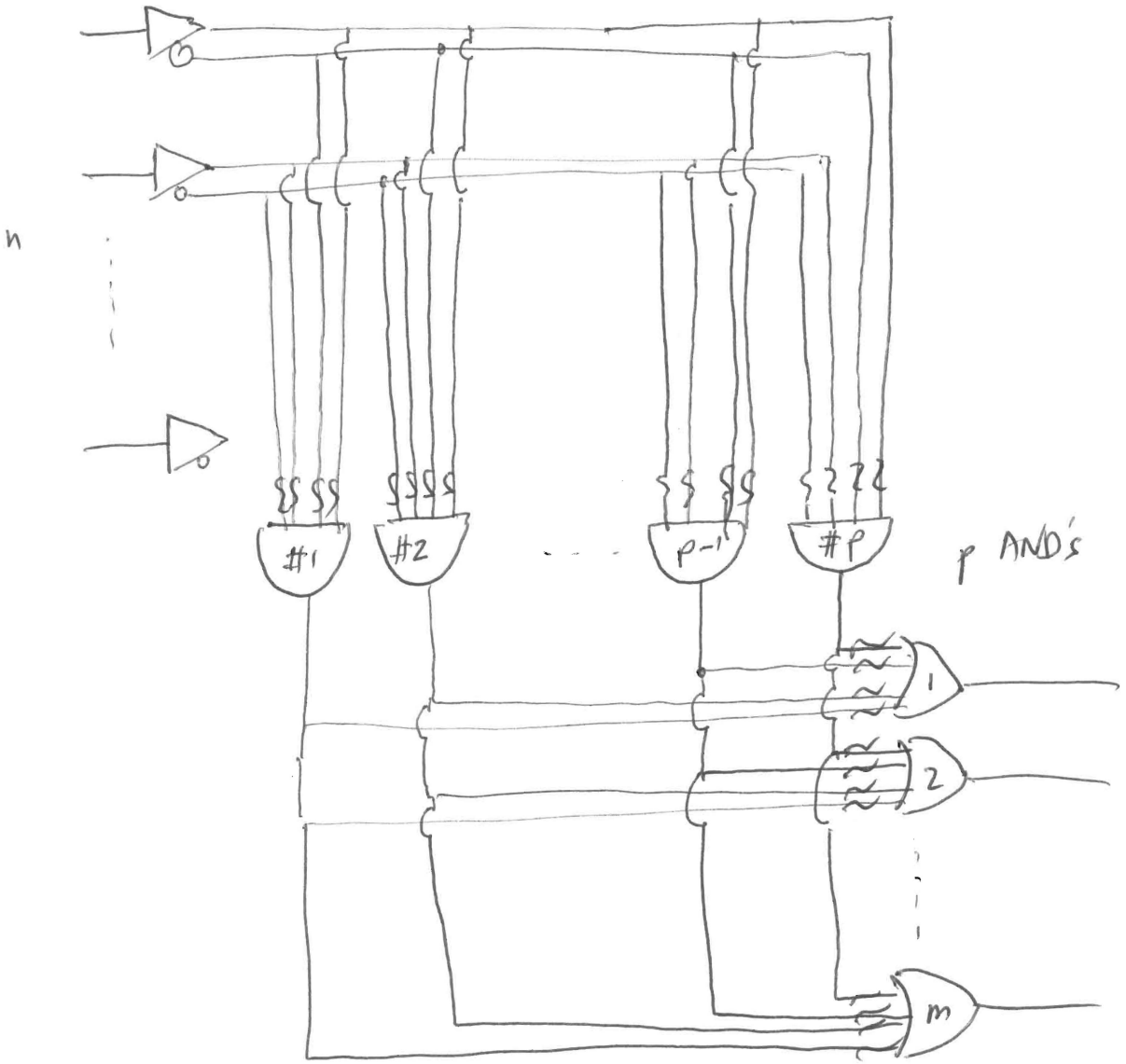


$2^n \times m$ PROM

PLA (AND's & OR's are programmable)

Example: $n \times p \times m$

- n: inputs
- p: AND's
- m: OR's



PAL (AND's are programable & OR's are fixed)

PLA's for combinational Logic Design:

$f_1(x,y,z) = \sum m(0,1,3,4)$

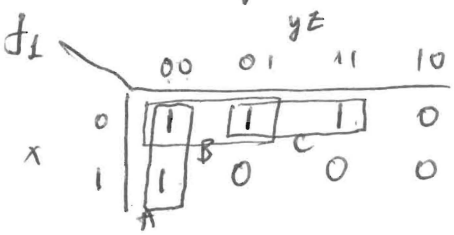
$f_2(x,y,z) = \sum m(1,2,3,4,5)$

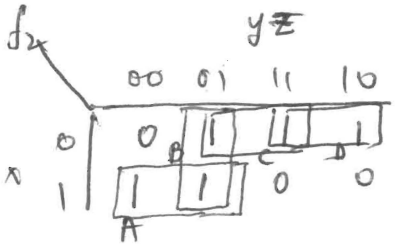
- 3 inputs (x,y,z)
- 2 outputs (f_1 & f_2) → 20 Pins
- Depends on simplifications. Let's pick 4.

PLA: $3 \times 4 \times 2$

m	x	y	z	f_1	f_2
0	0	0	0	1	0
1	0	0	1	1	1
2	0	1	0	0	1
3	0	1	1	1	1
4	1	0	0	1	1
5	1	0	1	0	0
6	1	1	0	0	0
7	1	1	1	0	0

K-Maps:





3 groups of two ones

PI's: $\bar{y}\bar{z}$ (A), $\bar{x}\bar{y}$ (B), $\bar{x}z$ (C)

Essential: A & C to cover all four min terms.

$f_1(x,y,z) = \bar{y}\bar{z} + \bar{x}z$

4 groups of two ones

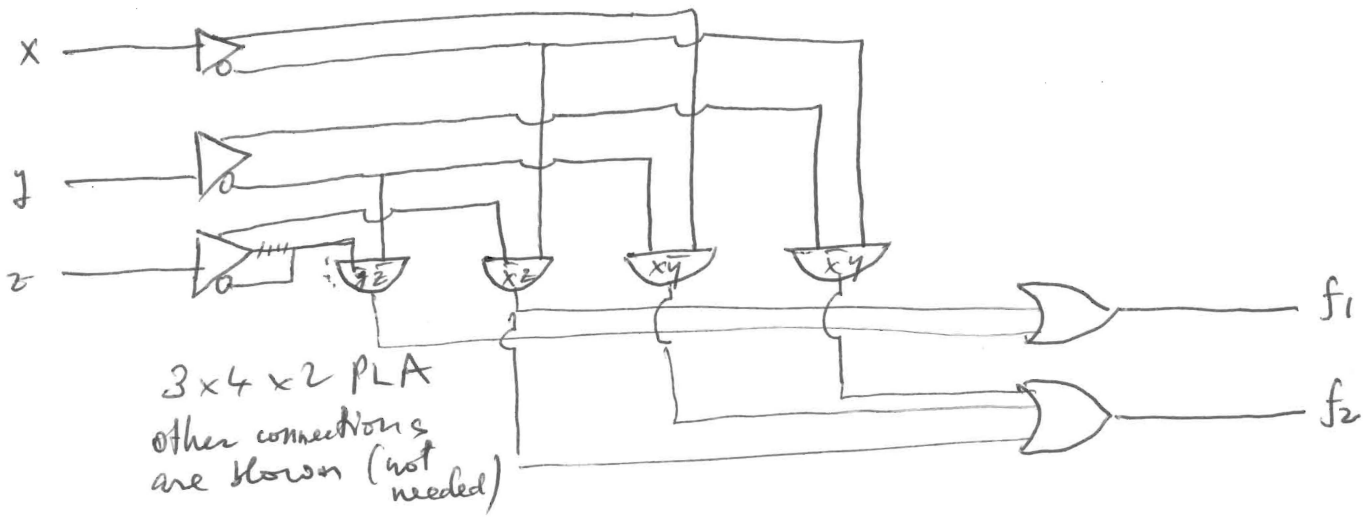
PI's: $x\bar{y}$ (A), $\bar{y}z$ (B), $\bar{x}\bar{z}$ (C), $\bar{x}y$ (D)

Essential: A, C, D or A, B, D

$f_2(x,y,z) = x\bar{y} + \bar{x}\bar{z} + \bar{x}y$

AND's needed (products): y & z; x & z; x & y; $x\bar{y}$; $\bar{x}y$

✓ Good with 4 AND's ✓ PLA.



3×4×2 PLA

other connections are blown (not needed)

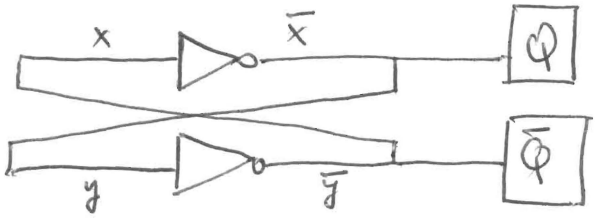
Ch 6 Flip-Flops & Applications

Networks or Circuits.

- **Non-sequential:** output of a function depends on a set of inputs (x, y, z, etc.) at a same time instant
- **Sequential:** output of a function depends not only on inputs at the same time instant, but on inputs at previous time instants. A sequential network has memory.

Sequential Networks
- Synchronous: internal clock sets updates @ discrete time instants.
- Asynchronous: no set time for update.

Bistable element : two state: Q & $\bar{Q}$

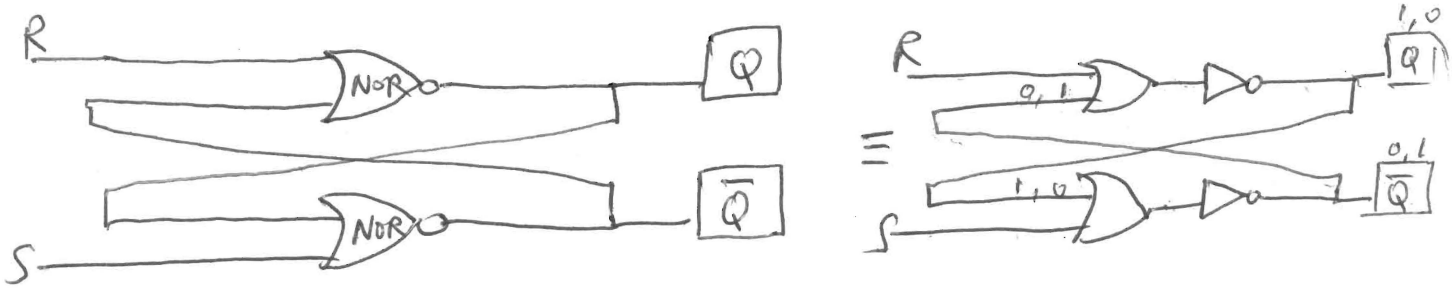


$\bar{Q}$	Q	
1	0	} Two states
0	1	

(0.5 0.5 : Meta-stable state.
↓ (Should be avoided)

Note: if $x=0 \longrightarrow \begin{cases} Q=1 \\ \bar{Q}=0 \end{cases}$ and the element will stay in this state unless voltage is introduced to clear setting / clear memory / reset

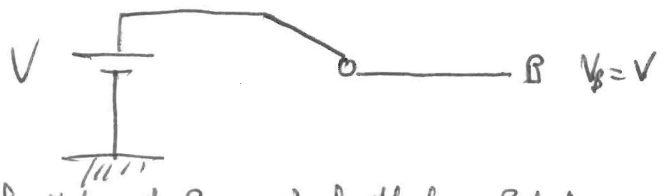
SR Latch (Set & Reset Latch)

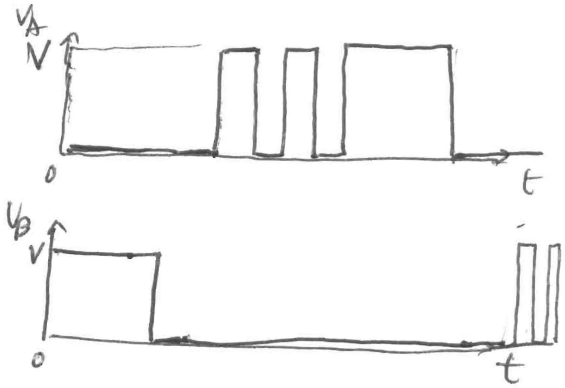


R S	Q $\bar{Q}$	
0 0	1 0 0 1	} Bistable element.
0 1	1 0 → 1 0 0 1 → 1 0	
1 0	1 0 → 0 1 0 1 → 0 1	
1 1	unpredictable (to be avoided)	

Application of a SR Latch: Switch debouncer

Contact bounce (very common):

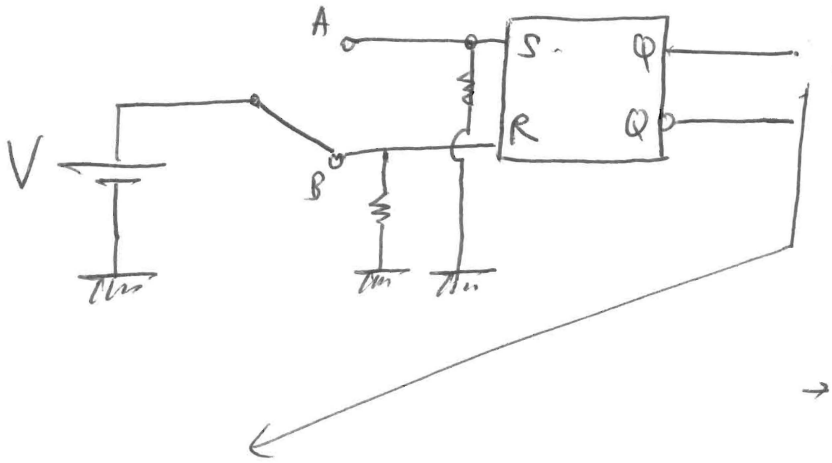




1) Switch at B ; 2) Switch from B to A : it takes some time to go from B to A : B drops to 0 while A still @ 0
3) Switch arrives @ A : V_A is up to V 4) Mechanical switch bounces away from A and back a few times → CONTACT BOUNCE. Happens every time switch goes into operation.

Contact bounce is a big issue in electronic keyboards when a key is pressed once it may act like it is pressed multiple times.

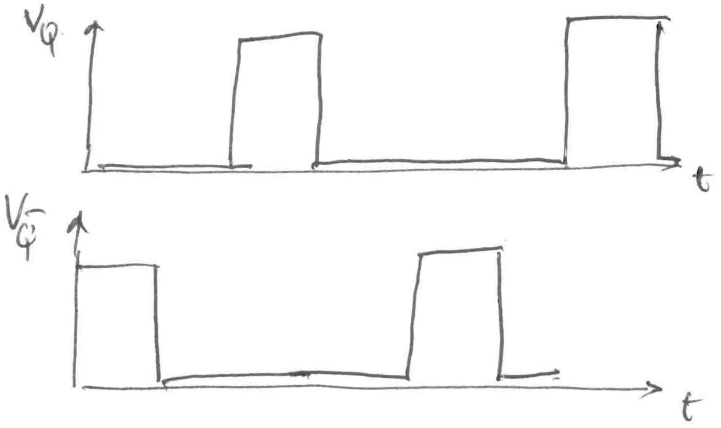
Solution: use an SR Latch after the switch:



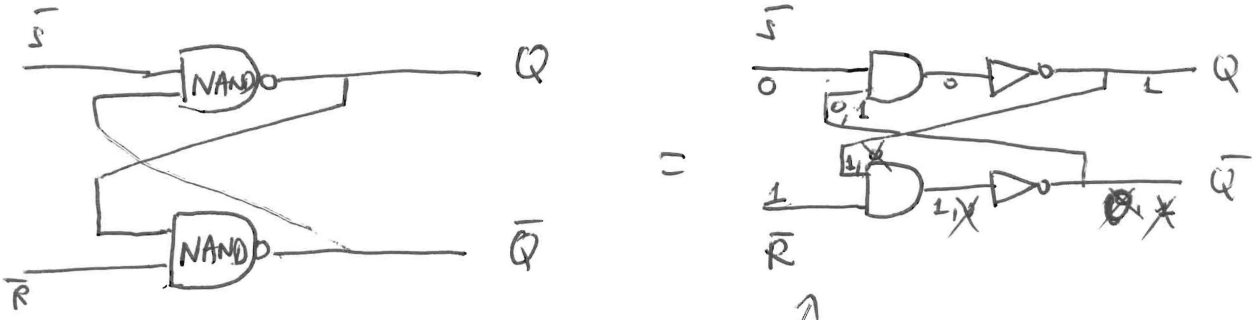
→ When not connected, A or B will stay @ 0
(i.e. one example: $V_A = 0$, $V_B = V$ ($S=0$; $R=1$)

→ When we switch from B to A: $S = 0 \to 1$; $R = 1 \to 0$

→ If switch bounce away from A: $S=0$; $R=0$ → Bistable Element behavior
→ $Q\bar{Q}$ will stay the way it was → bouncing effect is eliminated!



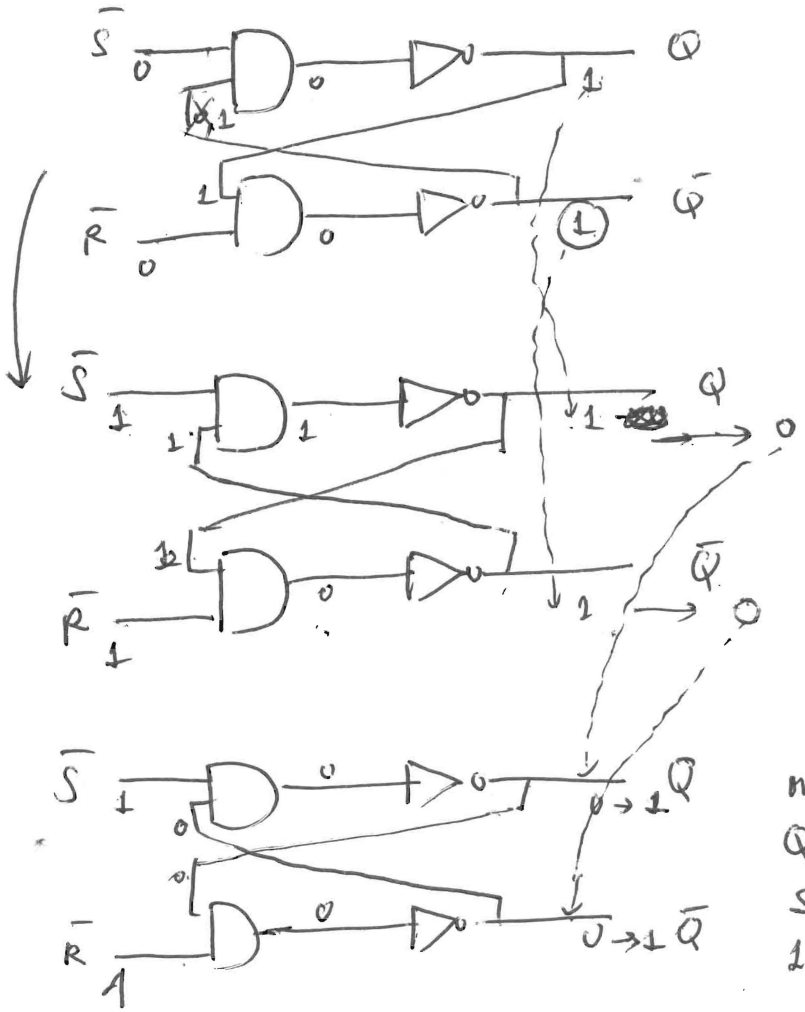
$\bar{S}\bar{R}$ Latch : using two NAND's



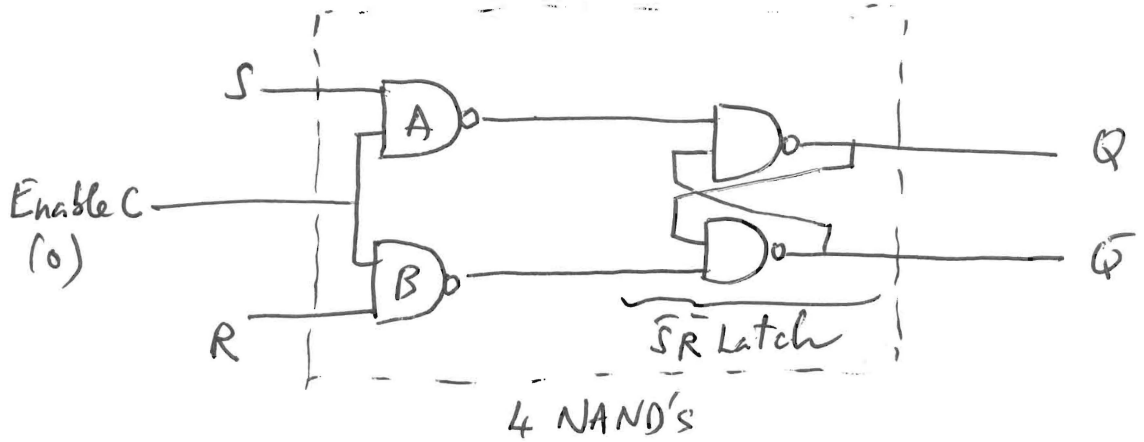
RS	$Q\bar{Q}$	
0 0	1 0	} Bistable element
	0 1	
0 1	1 0	
1 0	0 1	
1 1	unpredictable	

$\bar{R}\bar{S}$	$Q\bar{Q}$	
1 1	0 1	} Bistable element.
	1 0	
1 0	1 0	
0 1	0 1	
0 0	1 1	* This is OK but unstable/unpredictable if both $\bar{R}\bar{S}$ turns from 0→1. See below

X Y	AND
0 0	0
0 1	0
1 0	0
1 1	1



Gated SR Latch: (adding enable/disable)



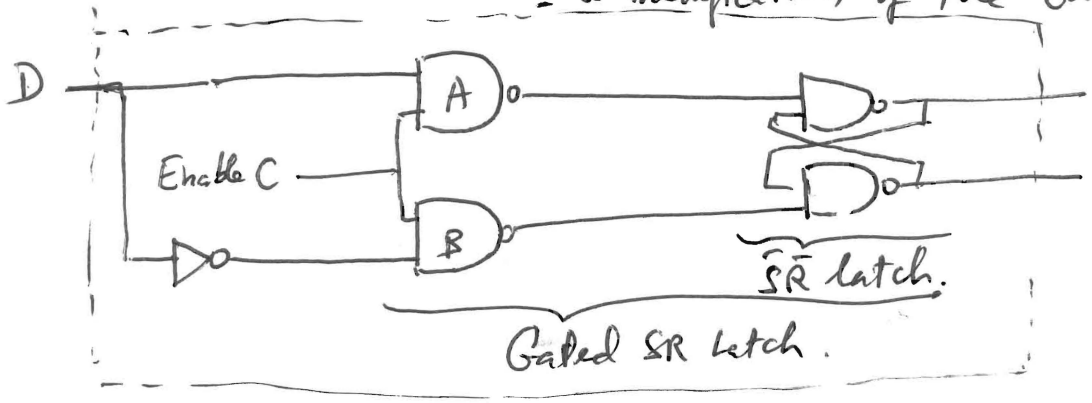
X y	AND	NAND
0 0	0	1
0 1	0	1
1 0	0	1
1 1	1	0

S R	C	A B	Q $\bar{Q}$	
– –	0	1 1	$\begin{matrix}0&1\\1&0\end{matrix}$	Bistable element
0 1	1	1 0	0 1	
1 0	1	0 1	1 0	
1 1	1	0 0	1 1	(unpredictable)
0 0	1	1 1	$\begin{matrix}0&1\\1&0\end{matrix}$	Bistable element.

If C=0, no matter what changes we apply to S & R, Q $\bar{Q}$ is bistable (it keeps the same state it was in before the changes)

Gated D latch:

- eliminates the inputs that lead to unpredictable outputs in the previous 3 latches → Eliminate S=1=R by making $R=\bar{S}$
- a modification of the Gated SR latch.



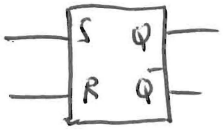
D	C	A B	Q $\bar{Q}$	
–	0	1 1	$\begin{matrix}0&1\\1&0\end{matrix}$	Bistable
0	1	1 0	0 1	
1	1	0 1	1 0	

Propagation Delay in flip-flops:

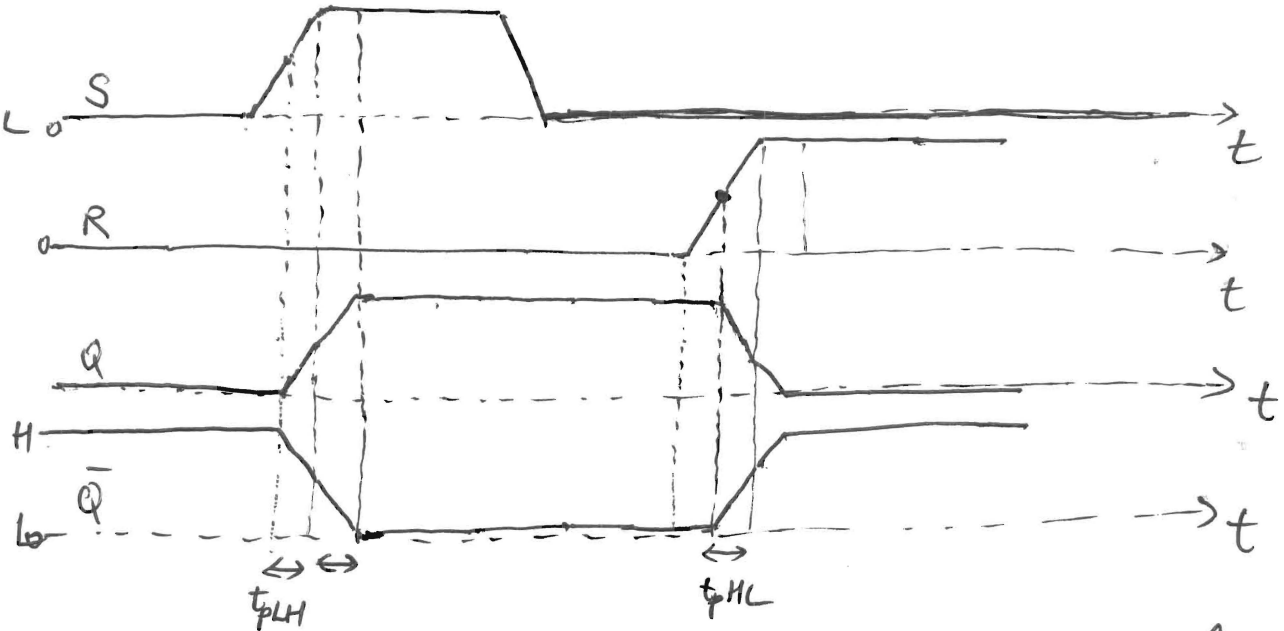
R S	Q $\bar{Q}$
0 0	Q $\bar{Q}$
0 1	1 0
1 0	0 1
1 1	unpredictable.

Voltages can be { Low (0) / High (1) }

SR Latch



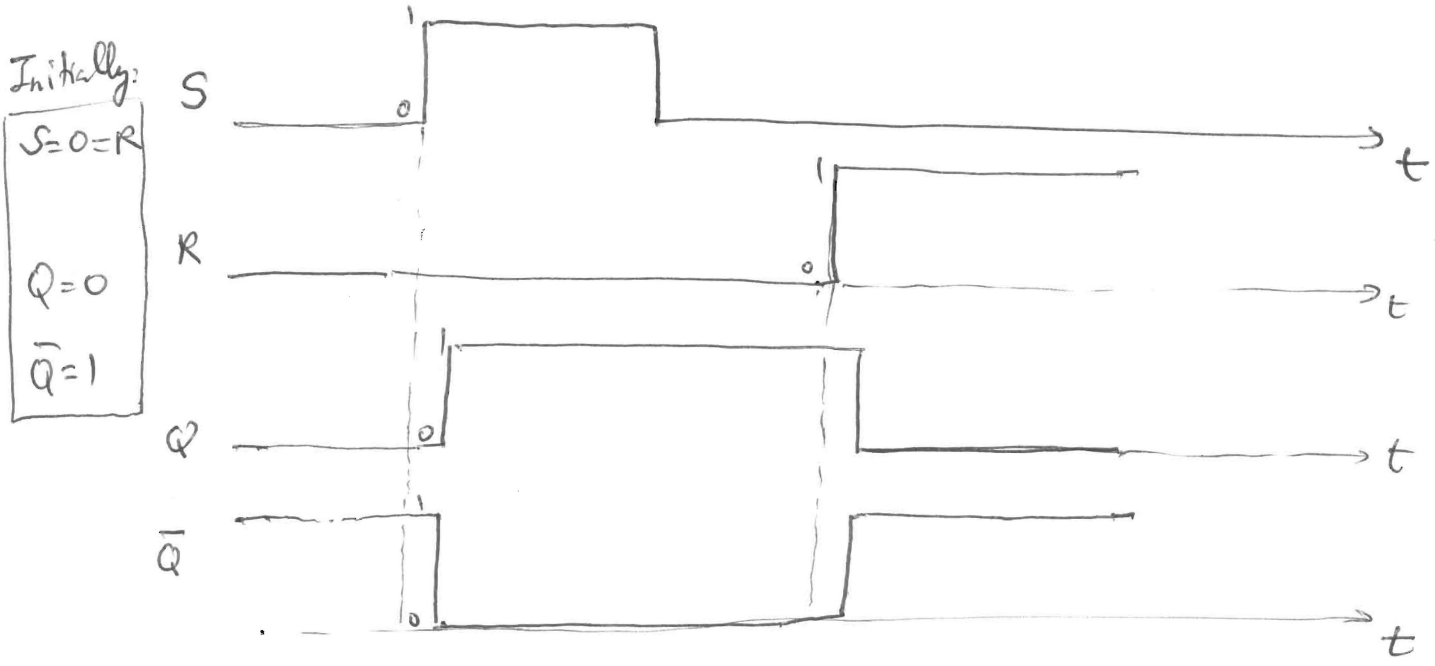
Assuming: initially
$S=0=R$
$Q=0=\cancel{\bar{Q}}$
$\bar{Q}=1$



* When S changes from 0 ✓ to some finite nonzero value, Q $\bar{Q}$ are changed instantaneously !
* $S_0=0$, $R_0=0$ | when S starts changing to 1 there is a delay for Q to change from 0 to 1 & $\bar{Q}$ to change from 1 to 0
* When S goes back to 0 (R still at 0) : since output in this case is bistable → Q $\bar{Q}$ is unchanged at 10
* Then S will stay 0 until the end
* At some point R goes to 1

R S	Q $\bar{Q}$	
0 0	1 0	
↓ ↓	↓ ↓	→ with delay.
1 0	0 1	

* Most of the times, we omit finite slopes, assuming all delay times are equal : <u>there are still delay times !</u>



Master - Slave Flip Flops (pulse - trigged flip flops)

SR Flip - flops :

RS	$Q\bar{Q}$	
00	10	} Bistable state
	01	
01	10	
10	01	
11	unpredictable.	

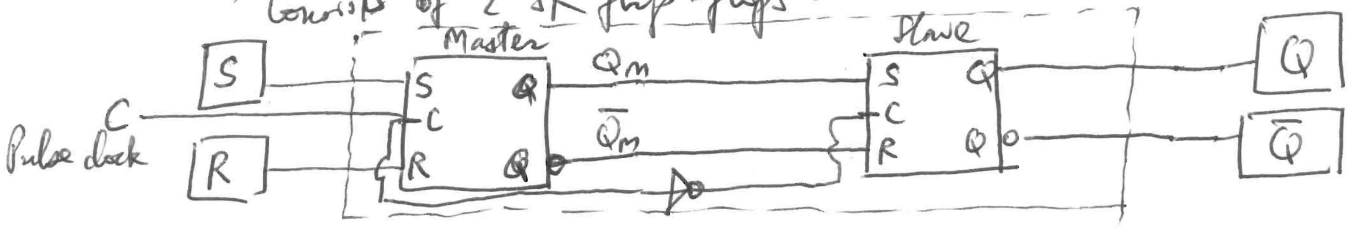
} When inputs are reset, the information propagates to the output right away (except for delays).

* input → output : immediate (except delays) or "transparent"

M-S Flip flops : the information is not propagated (input to output) immediately, but controlled by a pulse from Master's clock

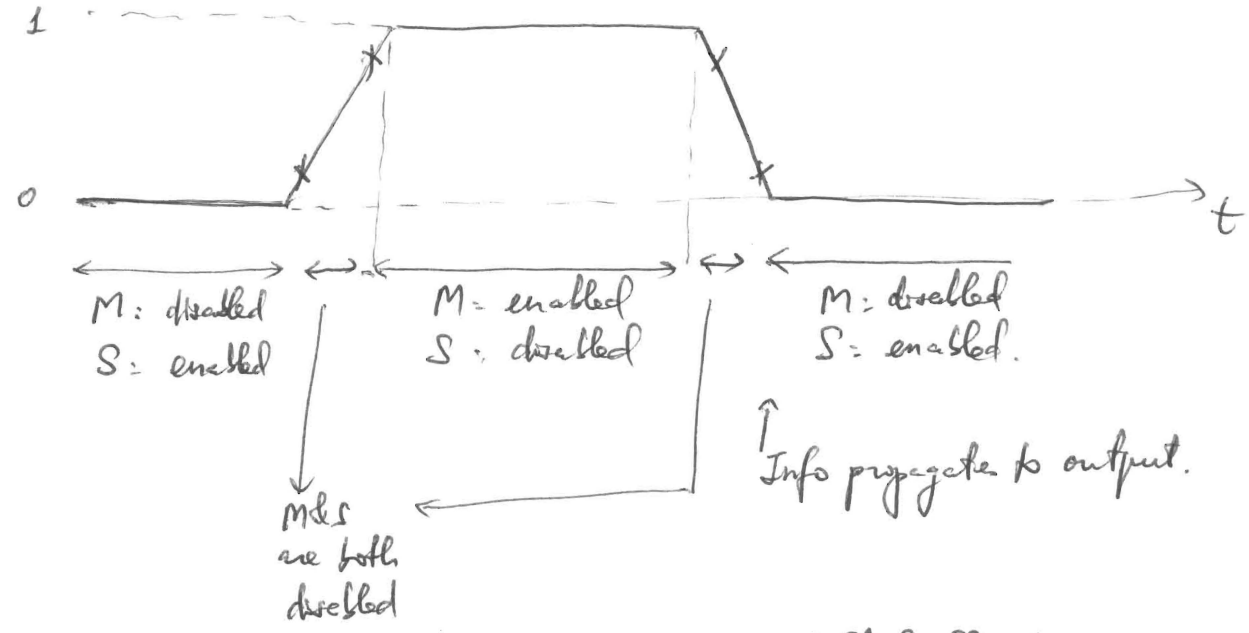
* input → output : not immediate or "non-transparent"

→ Consists of 2 SR flip-flops : a master & a slave :



C: Enable input : via an inverter when the Master is enabled the Slave is disabled. Using a clock pulse on C of the Master we can control when the Slave is enabled (information propagation input → output)

Clock pulse



There different types of M-S Flip-Flops
- M-S SR ✓
- M-S JK
- M-S D
- M-S T

MS SR Flip Flop

S	R	C	Q	$\bar{Q}$	
–	–	0	Q	$\bar{Q}$	(stays at its initial state)
0	0	1	Q	$\bar{Q}$	
0	1	1	0	1	
1	0	1	1	0	
1	1	1	–	–	(unpredictable)

(Bistable state)

$S_m = S_s$	$\bar{Q}_m = R_s$	Q	$\bar{Q}$
Q	$\bar{Q}$	Q	$\bar{Q}$
0	1	0	1
1	0	1	0
–	–	–	–