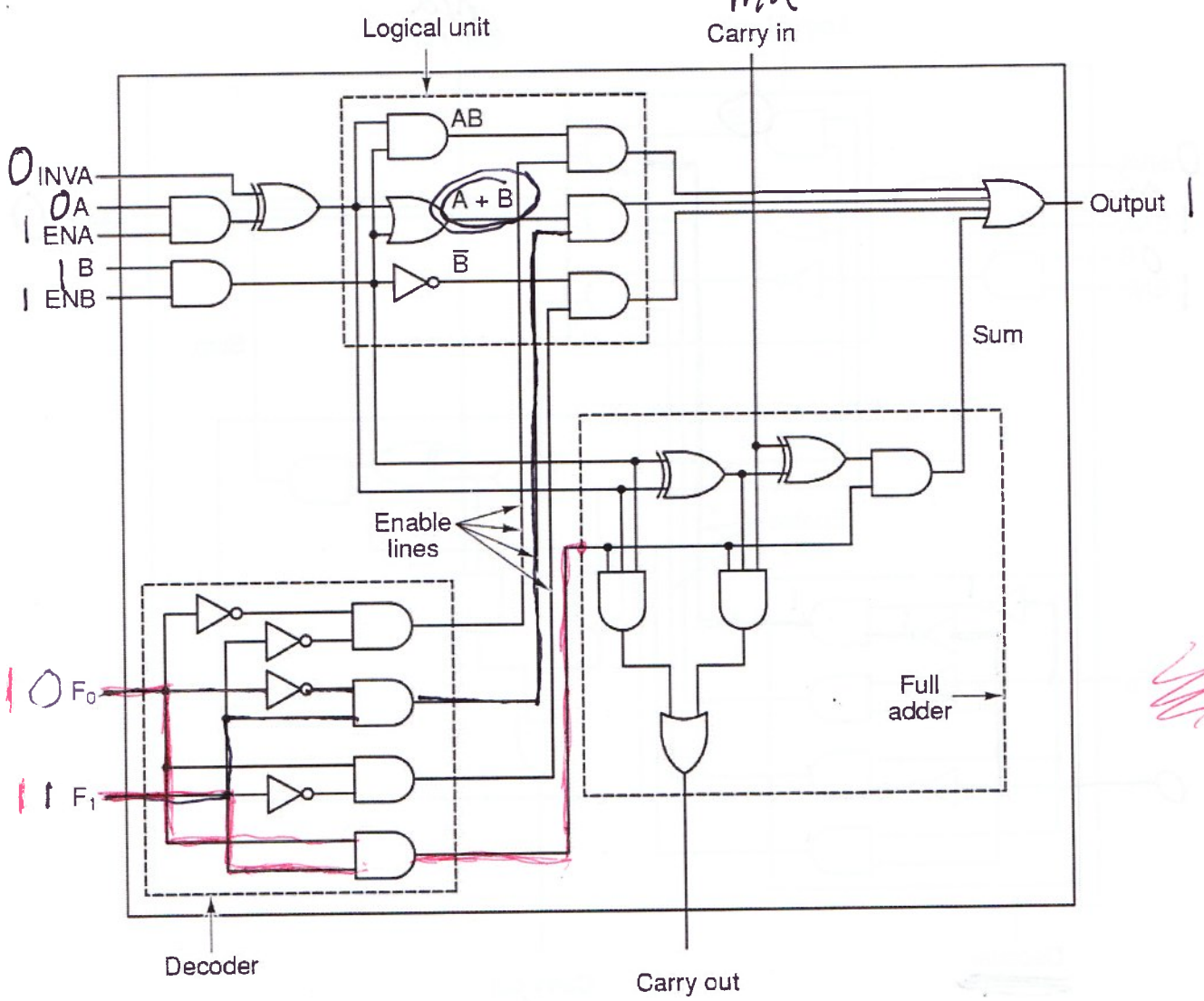
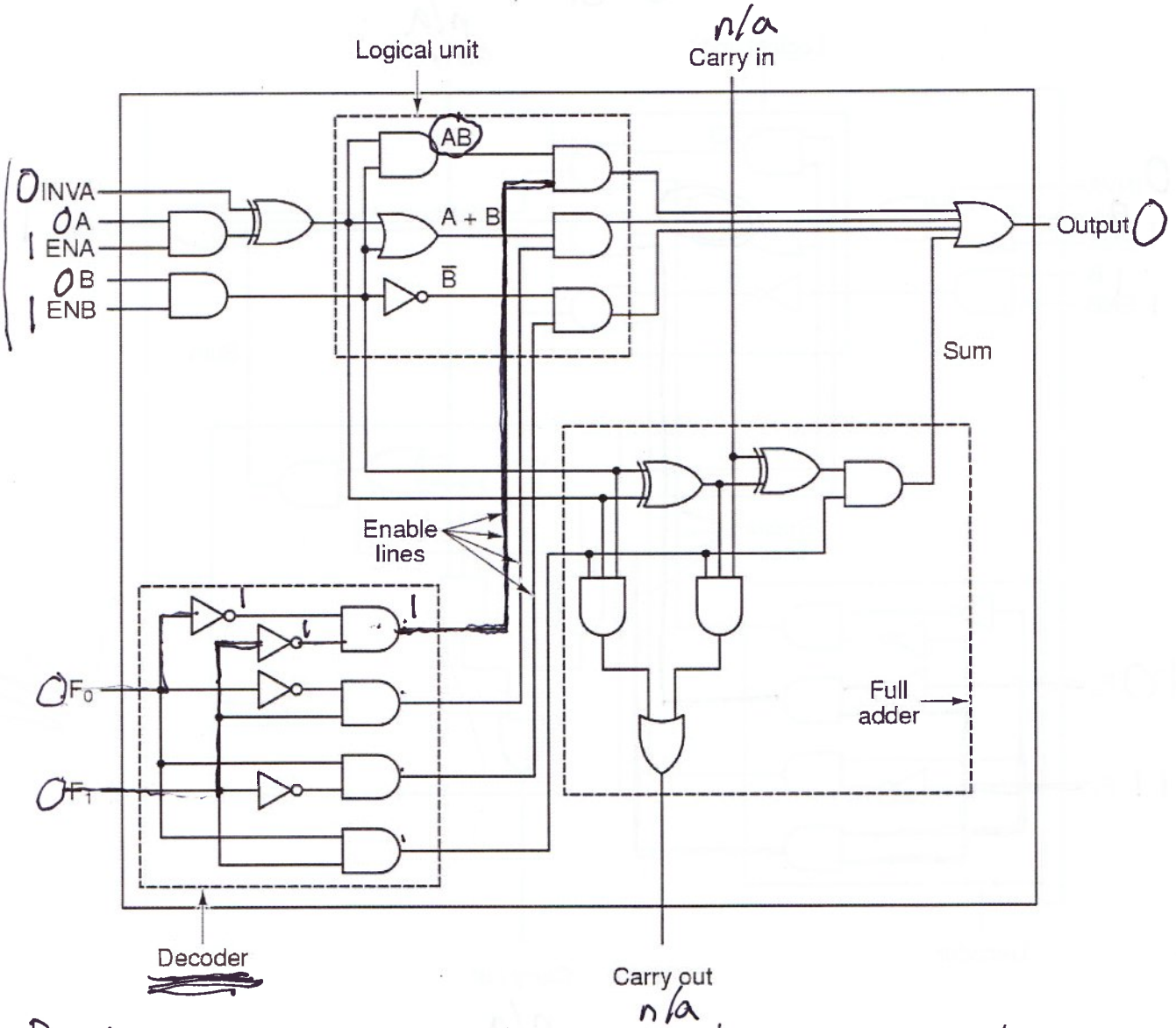
0 or 1



since we are not using the adder, Cin / Cout does not get used

morgan ariss & mac craig

0 and Q # 0



Decoder: ~~logic devic~~ Logic device with n inputs, and 2^n outputs

Game show analogy

ctrl lines = judges (n)

contestants (2^n)

output: winner (1)

Latches:

1-bit register

nor gates

bonus mark:
log base 10
2 decimal points

a	b	a nor b
0	0	1
0	1	0
1	0	0
1	1	0

Goal: to remember/store 1 bit: Q

Build a S-R latch: Set Q to 1
Reset Q to 0

} - SR is broken
- know why

S	R	Q
0	0	Q ← Memory
0	1	0
1	0	1
1	1	wtf??

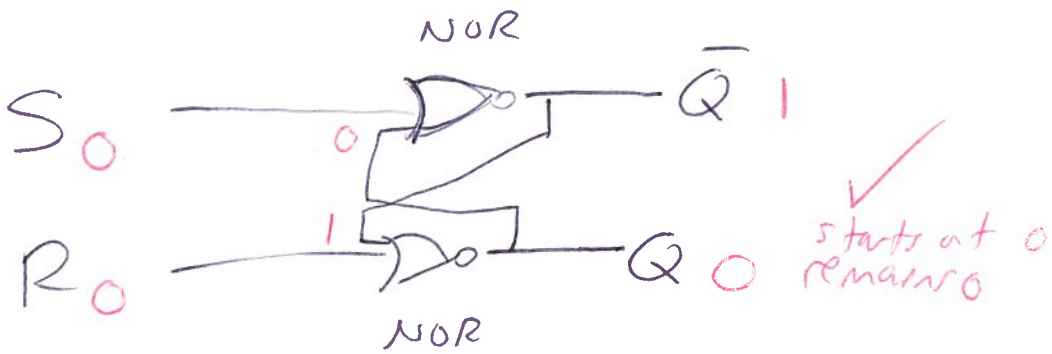
Row 0 of TT:

two cases:

(a) Q started at 0

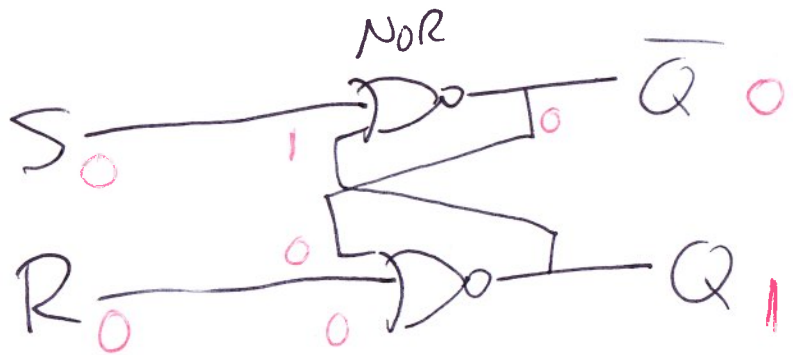
(b) Q started at 1

(a)



starts at 0 remains 0

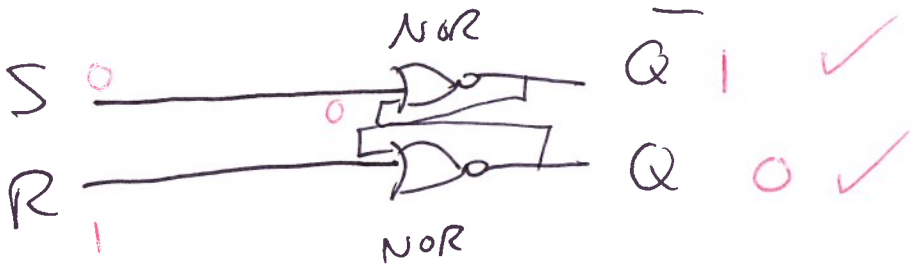
(b)



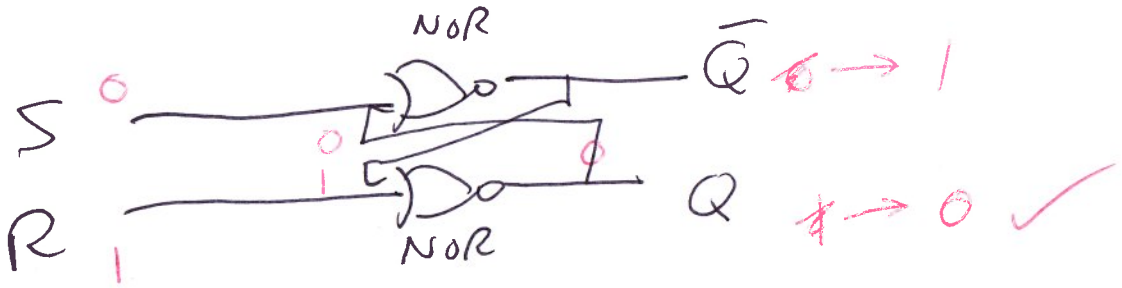
Starts at 1 stays at 1

find row of the TT

∴ An SR latch can function as a memory.



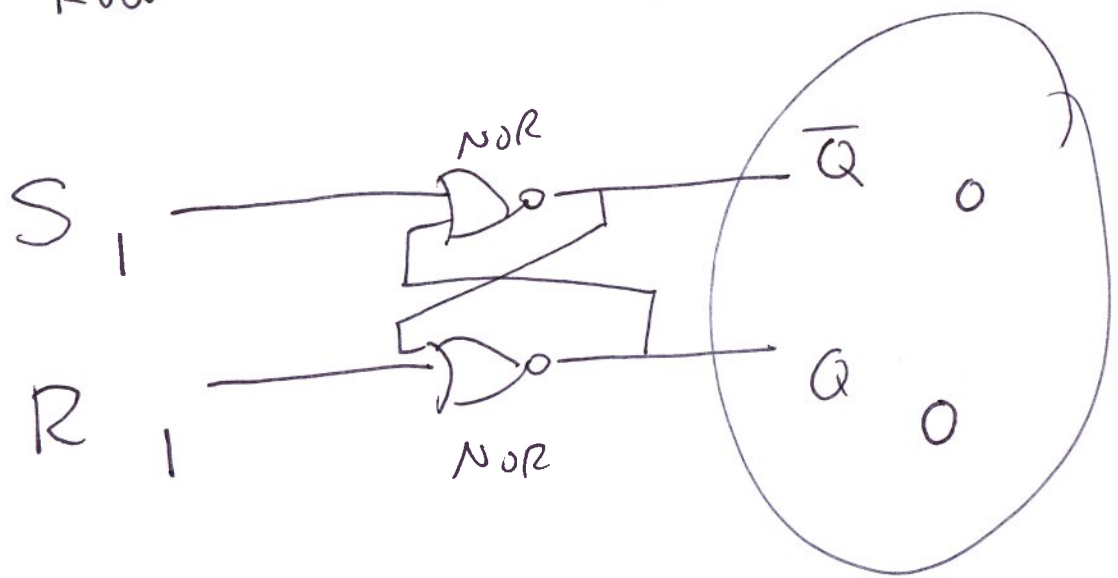
Row 1:
Reset the latch
ie
reset Q to 0



whether Q started at 0 or 1 it ends as 0

An SR can reset and set

Last Row: Row 3: $S = R = 1$



inconsistent in its output

nonsense for its input

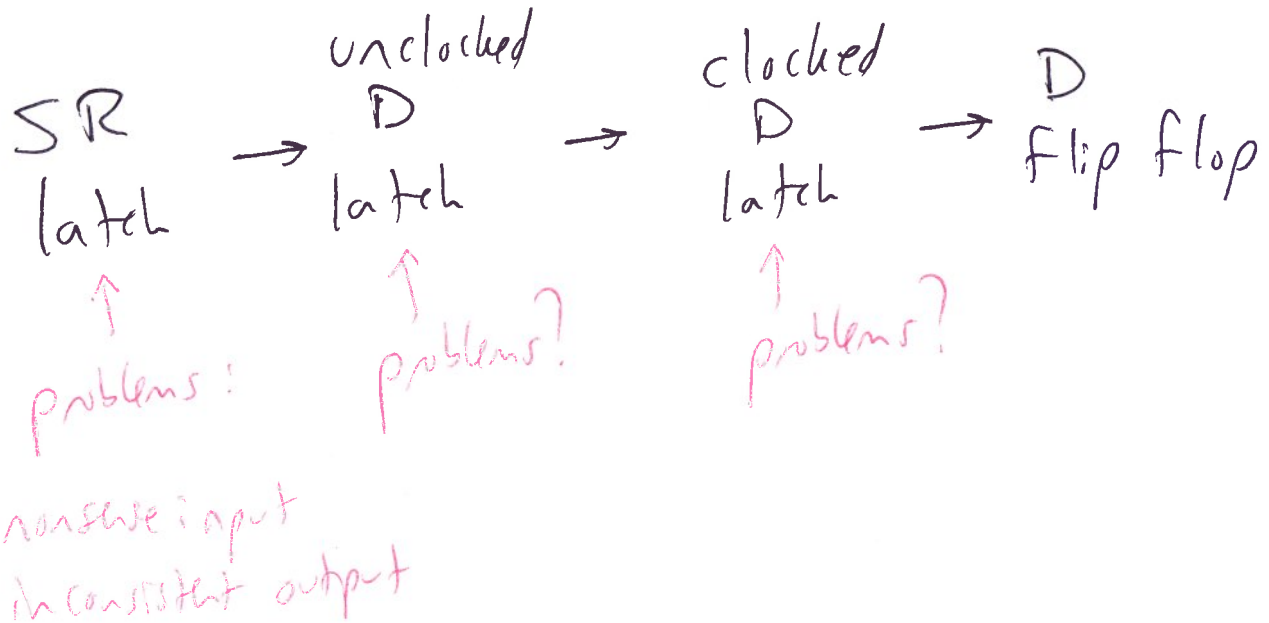
We need to change the SR latch to make it better:

✓ - memory is ok. it can store bits @ Q (S=0, R=0)

✓ - resetting works: (S=0, R=1, Q ends up at 0)

✓ - setting " (S=1, R=0, " " " " 1)

✗ - simultaneously setting and resetting need to be prevented.



quiz:

- presentation material
- SR latch
- $\log_{10}$ of decimal #s to 2 decimal places: bonus