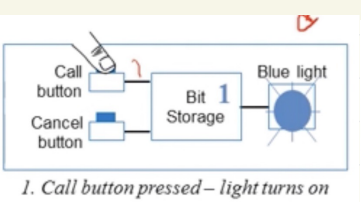
Combinational Circuits

- o/p is a fn of present inputs
- Δ input ⇒ Δ output after a short propagation delay
- knowing current input ≡ predicting an output

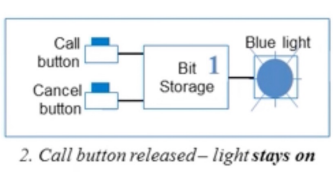
Sequential Circuits

- introduce the idea of "state"
- state of a circuit basically contains info generated in the past, and that info is used to determine what the output will be for the current input
- hence, o/p depends on present and past inputs.
- ∴ they have memory

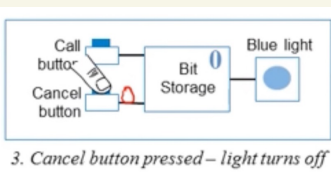
eg. pressing and then releasing a button, but wanting its effect to last until another button is pressed



1. Call button pressed – light turns on

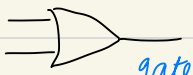


2. Call button released – light **stays on**

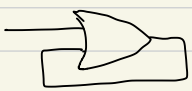


3. Cancel button pressed – light turns off

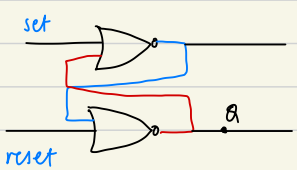
- sequential logic

i) 

can't store a value

ii) 

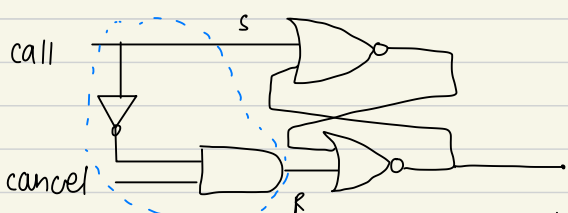
can store a value, but it can't be replaced

iii) 

SR (set-reset) latch

S	R	Q
1	0	1
0	1	0
1	1	→ flickering on & off
0	0	store

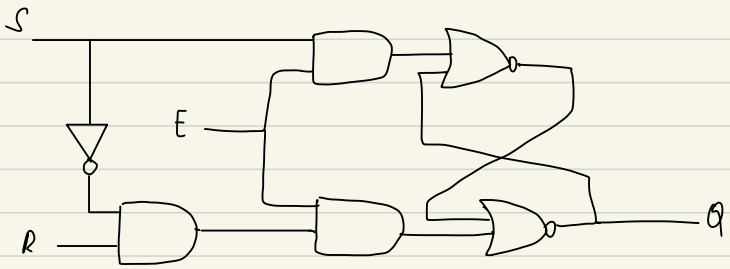
so when S & R are asserted / de-asserted at the same time, the output will oscillate, irl it will stabilize because one propagation is faster than the other, but it is impossible to know which one

iv) 

fixes above problem

but now there is a time delay b/w inverter & and gate and so for a brief period the output does oscillate "call to R is longer than call to S"

v)



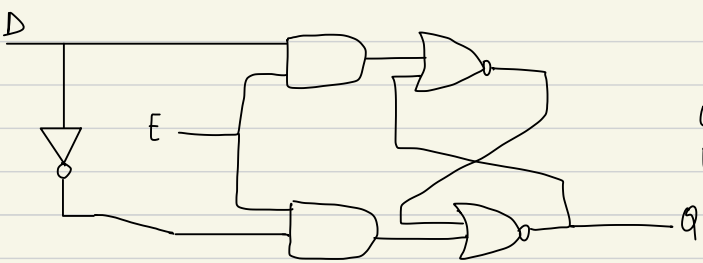
gate/enabled SR latch

E = 0 → retains prev value
E = 1 → changes

Q → o/p at current state
Q+ → o/p at next state

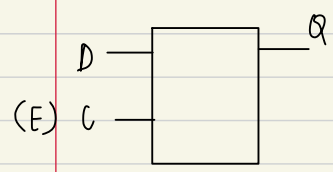
E	S	R	Q+	Fn
0	X	X	Q	store
1	0	0	Q	store
1	0	1	0	reset
1	1	0	1	set
1	1	1	?	undef.

vi)



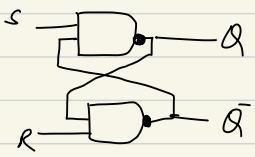
removes chance of invalid inputs

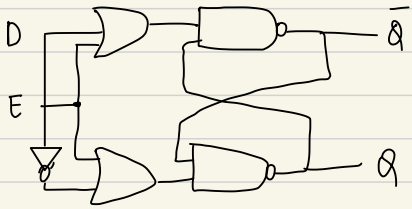
transparent D latch



E	D	Q+	fn
0	X	Q	store
1	0	0	reset
1	1	1	set

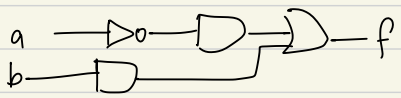
vii) nor ↔ nand, or ↔ and





E.	D	Q+	fn
1	X	Q	store
0	0	0	trans-
0	1	1	parent

propagation delay (again)



$t_a \equiv a \to f$

$t_b \equiv b \to f$

- give both values
- if asked for one, then max value

the clock

- a signal that toggles between 0 & 1 at a fixed rate
- sequential components are connected to the clock
- oscillation of the clock gives MHz / GHz numbers for the chips
- period : $0 \to 1$ (ns)
  freq $= \frac{1}{\text{period}}$ (GHz)
- we use only 1 clock in synchronously designed sequential circuits