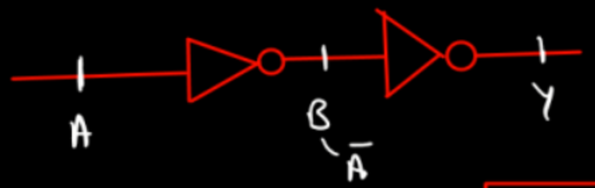
Cascading of Inverter:

जोड़ना (एक के बाद एक)



$B = \bar{A}$

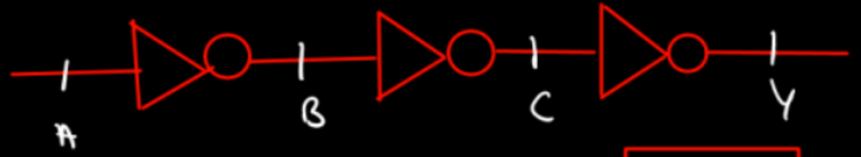
$Y = \bar{B} = \overline{\bar{A}} = A$

$\boxed{Y = A}$

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

input = output : Buffer gate

* If we Cascade 2 Inverters in a series then it will Act like a Buffer



$B = \bar{A}$

$C = \bar{B}$

$Y = \bar{C}$

$\Big\} \; Y = \bar{A}$

A = 0	B = 1	C = 0	Y = 1
A = 1	B = 0	C = 1	Y = 0

Inverter

output is Complement of Input $\Rightarrow \boxed{Y = \bar{A}}$

* If we Cascade 3 inverters in a series then it will act like an Inverter

Observation :

inverter

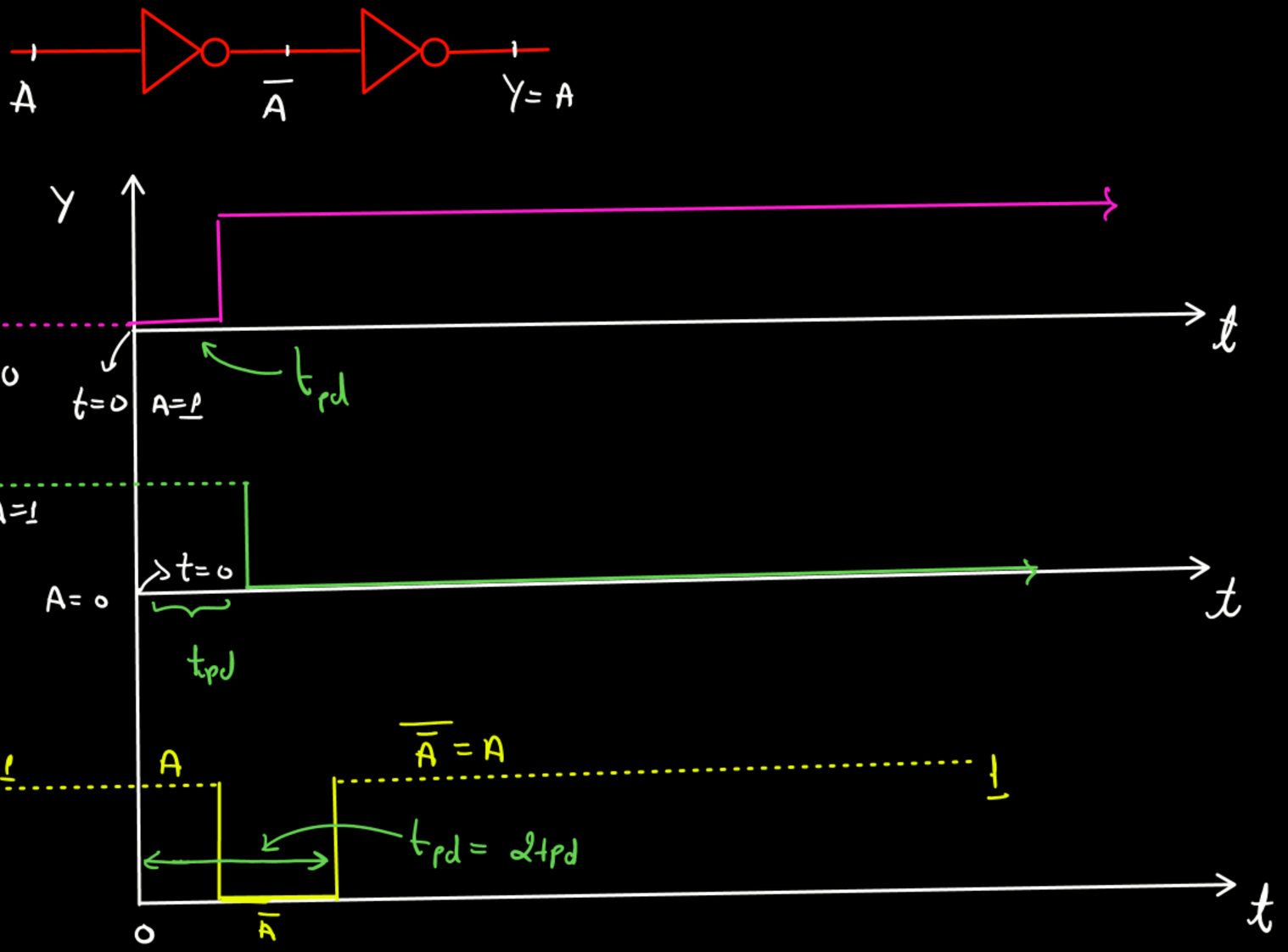
↳ if even number of NOT gates are connected in cascade then it behaves like a buffer.

→ But it generates a delay of $2n$ t_{pd}

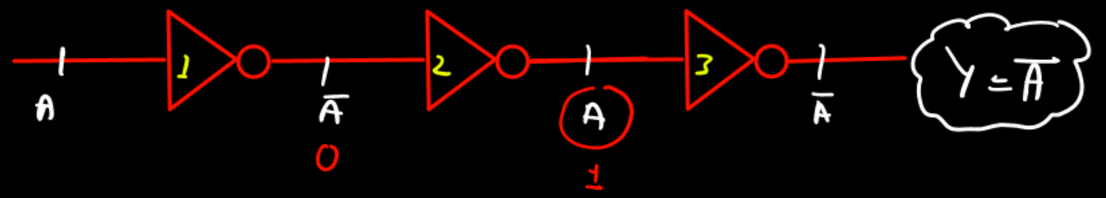
↳ if odd number of NOT gates are connected in cascade then it behave like an inverter

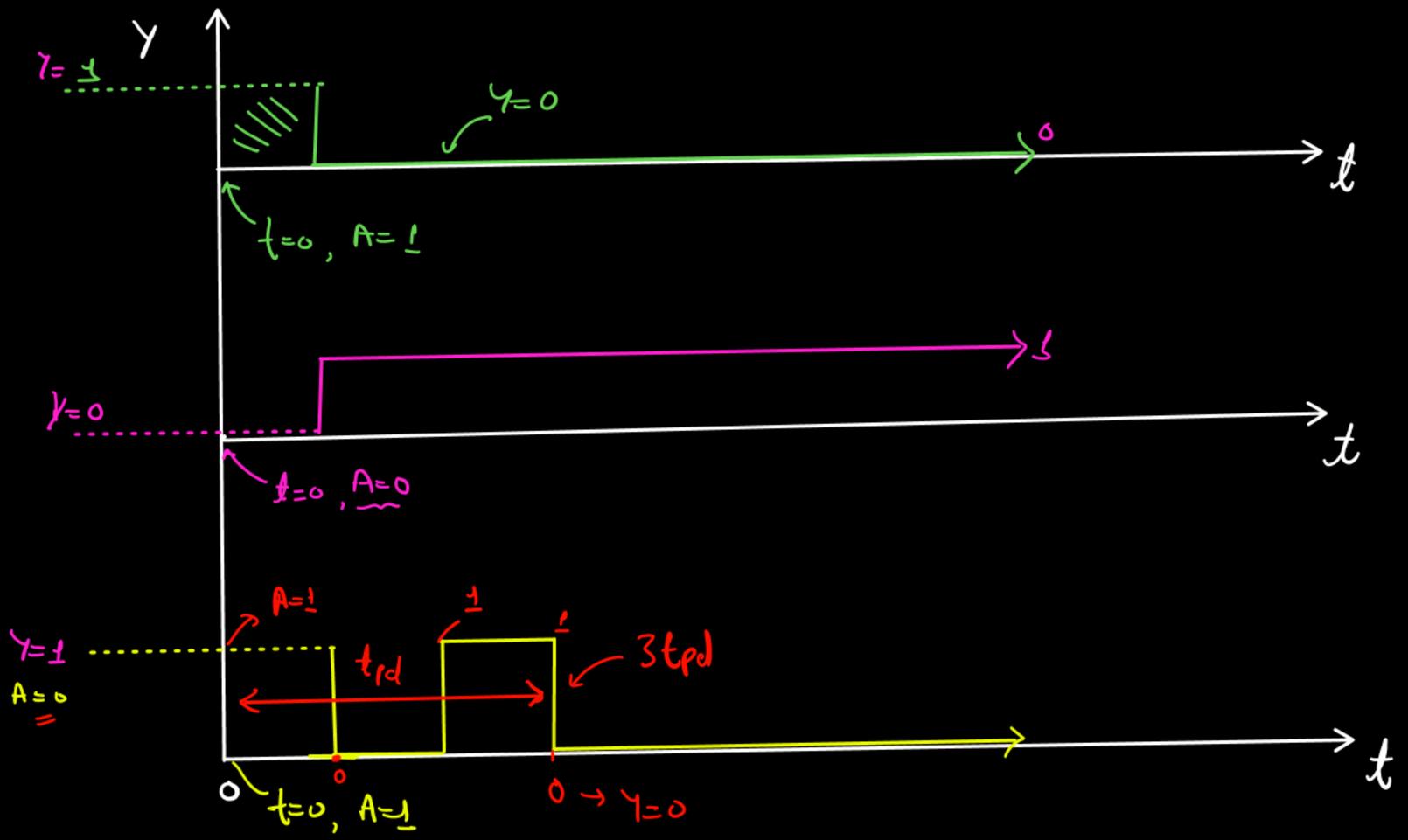
→ But it generates a delay of $2n+1$ t_{pd}

Graph of two inverters in Cascade

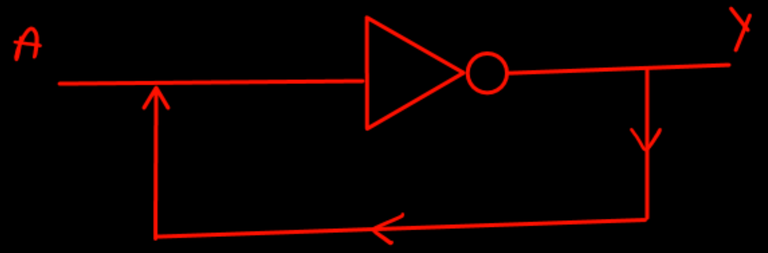


Graph of three inverters in Cascade



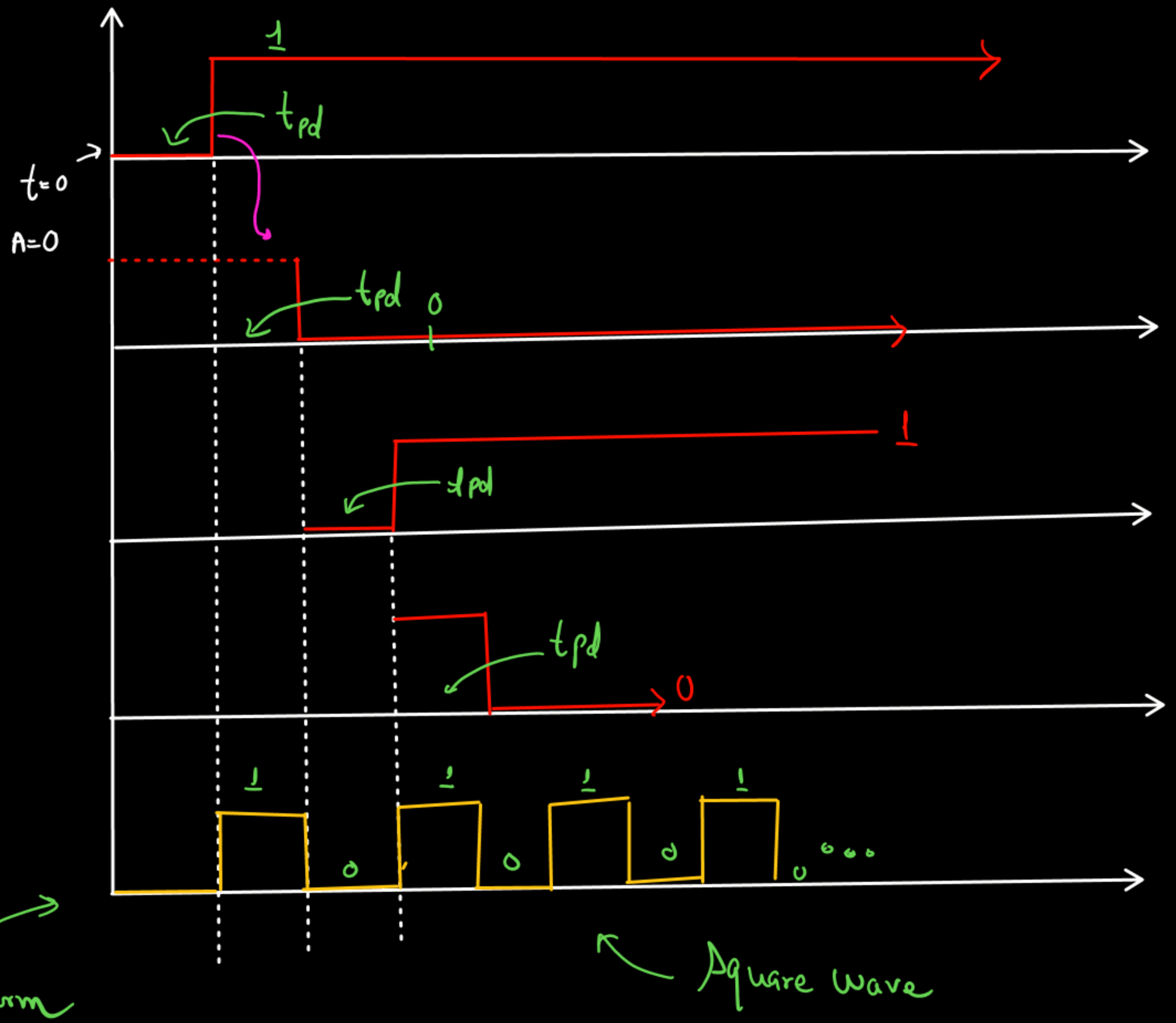


NOT gate with feedback



* The output generated by an inverter (by triggred input) will become the next input to the same inverter.

* Triggred input → input for a short duration



AND Gate:

अंग

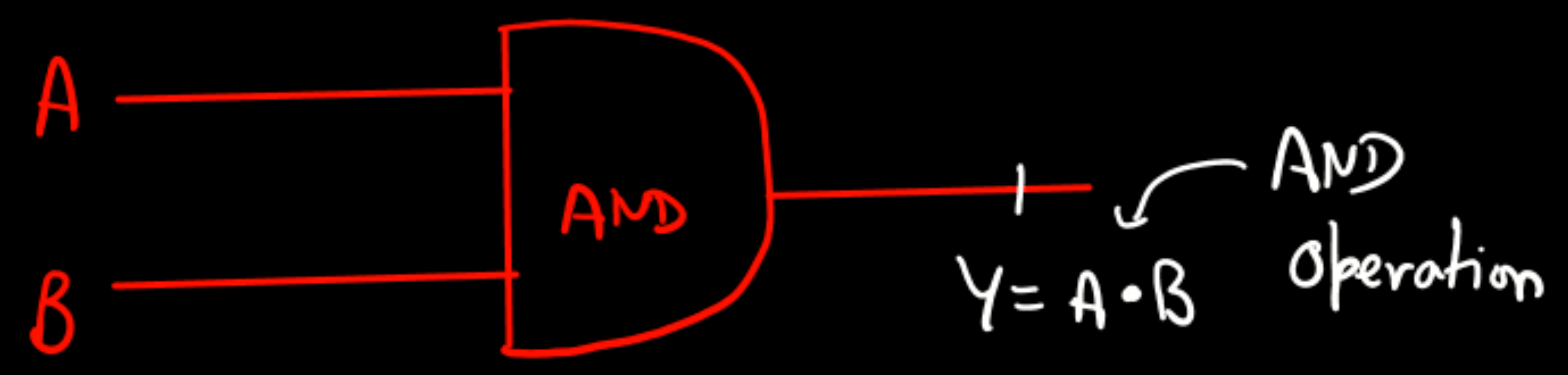
True = 1

False = 0

⇒ All input Conditions must be true in order for output to be True or high

If A = True & B = True then Y = True

Symbolic Representation



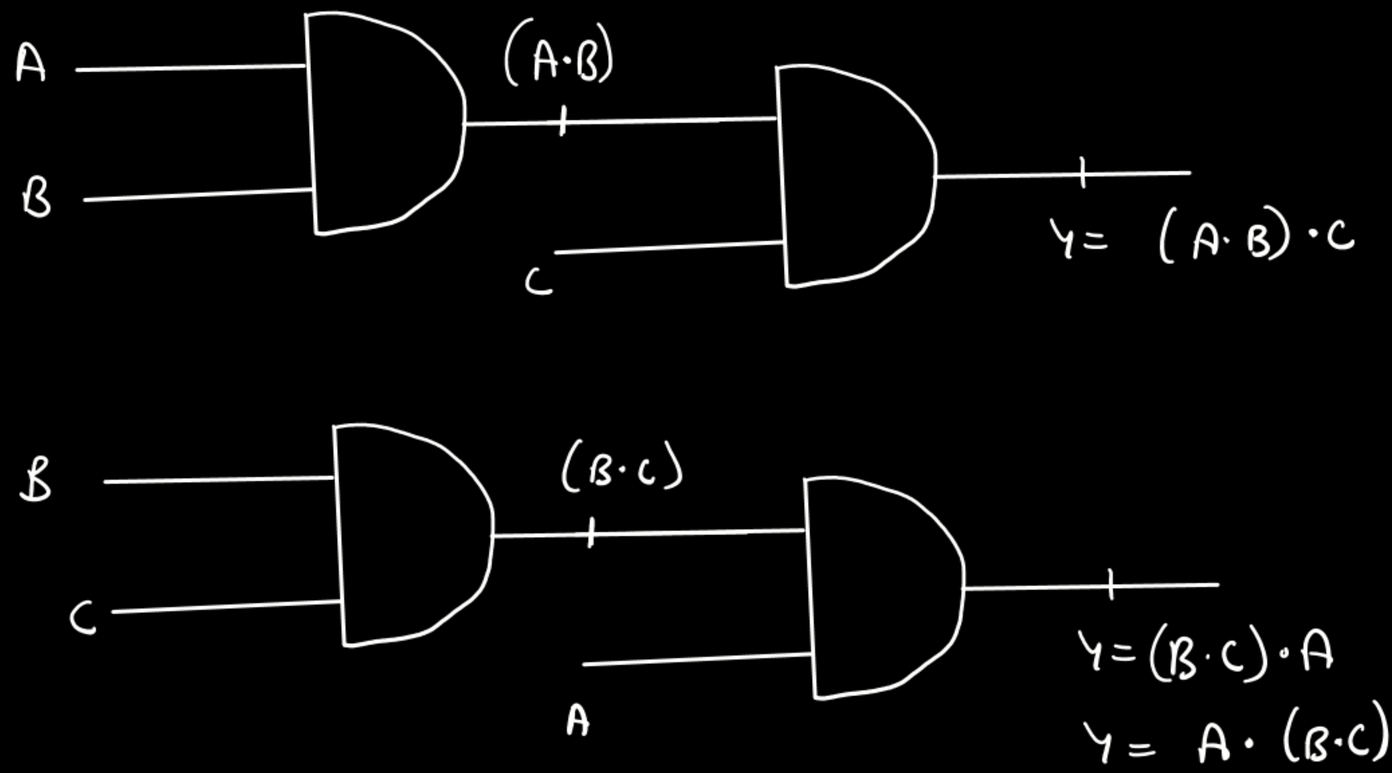
Truth Table

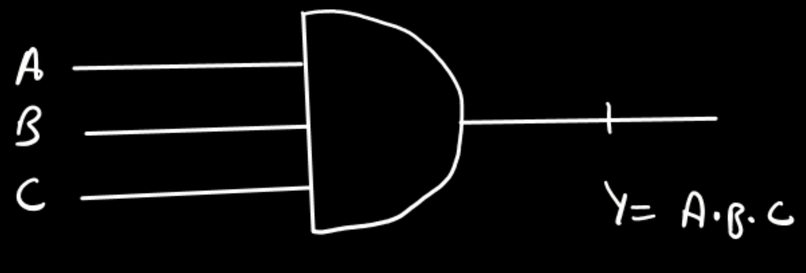
2 bit $= 2^2$ Combination $= 4$

A	B	Y = A·B
0	0	0
0	1	0
1	0	0
1	1	1

(a) Commutative law : $A \cdot B = B \cdot A$ (Possible)

(b) Associative law : $(A \cdot B) \cdot C = A \cdot (B \cdot C)$ (Possible)





Three Input AND gate

'Two Input AND gate in Cascade'

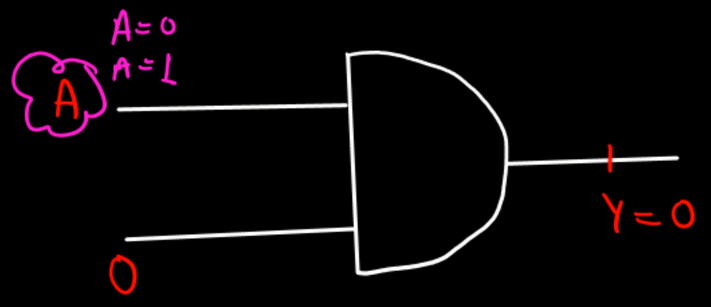
A	B	C	$Y = A \cdot B \cdot C$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

3 inputs → 2^3 Combinations = 8

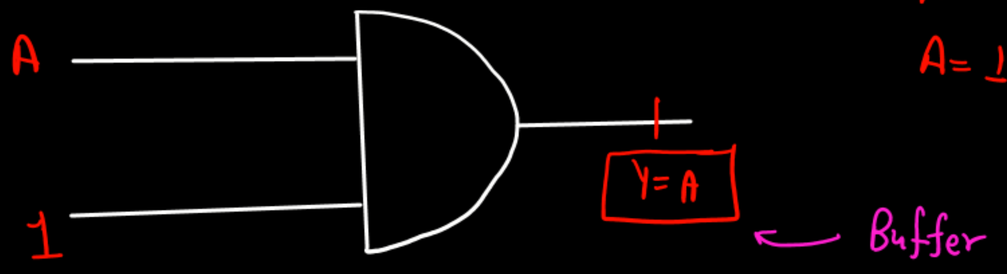
* Output is high if and only if all inputs are high
* It is similar to 'Dot Product'

$Y = A \times B \times C$

Enabled & Disabled AND gate



The input A not passed to the output So we can say that this in Disabled AND Gate.



A=0, B=1 ⇒ Y=0

A=1, B=1 ⇒ 1

The input is directly passes to the output

→ Enabled AND Gate

→ Work like a buffer

floating I/P :

↳ One Input is Unknown

↳ Ether 0, or 1, it's equals to other input