Binary Blah Blah!

- The Binary system comprises only two numbers viz. 0 and 1.
- Numbers could be of the form:
- 00, 01, 10, 11

(meaning 0, 1, 2, 3 in decimal)

- Thus,

$$(0101)_2 = 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0$$

Similarly,

$$(110.01)_2 = 1x2^2 + 1x2^1 + 0x2^0 + 0x2^{-1} + 1x2^{-2}$$

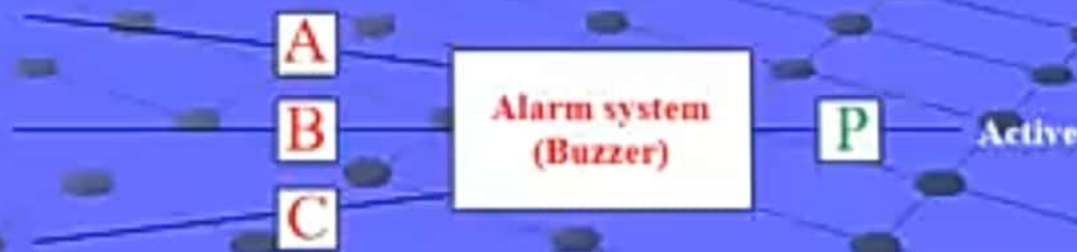
Bits, Nibbles and Bytes!

- **Bit**: Binary digiT
- 4 bits make a Nibble e.g.1011
- 8 bits make a Byte e.g. 11001101
- Using 2 bits we can generate 4 combinations viz. 00, 01, 10, 11 standing for 0, 1, 2 and 3 in the decimal system
- Thus, using *n* bits we can generate 2^n combinations

Thinking in terms of *0*s and

The Truth Table

A	B	C	P
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	1
1	1	1	0



The buzzer P is activated only under this condition -

A=1, B=1, C=0

A is TRUE **AND** B is TRUE **AND** C is FALSE

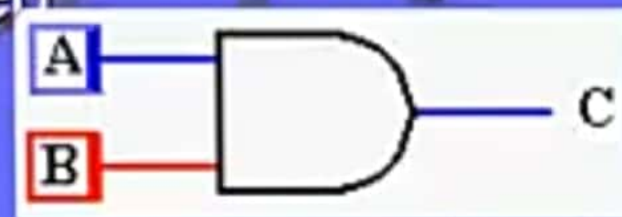
Observe that there is some logic that drives P

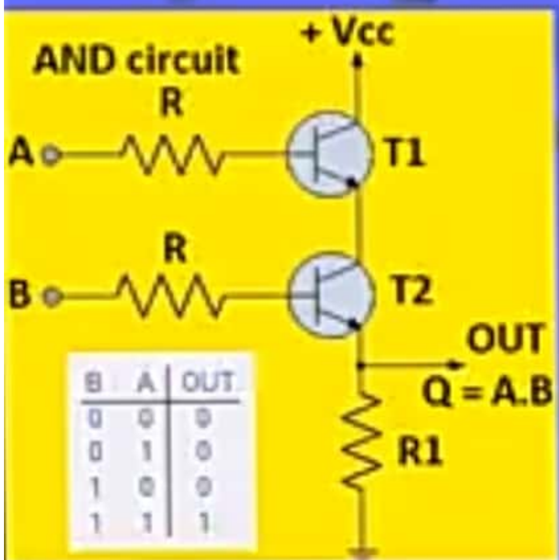
LOGIC GATES

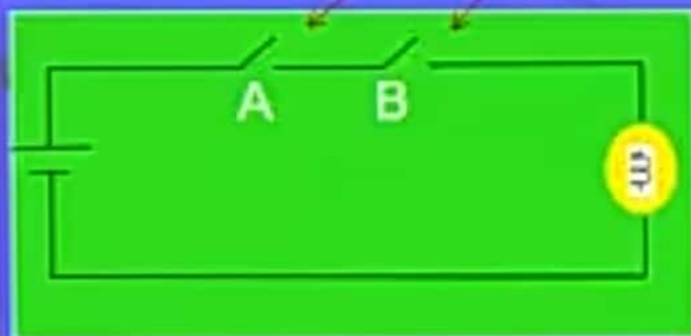
- A **logic gate** is an electronic component whose output is computed based on a function of the inputs.
- A gate can have one or more inputs.
- Inputs may be given directly as ***0*** (**LOW** VOLTAGE*) or ***1*** (**HIGH** VOLTAGE*) or they could be derived from the output of other logic gates.
- * LOW generally means 0V and HIGH generally means 3V or 5V.
- Computers use a very large number of such interconnected gates.

LOGIC GATES: *AND*

- In order for current to flow & the lamp light up, both switches must be closed
  - Logic notation A•B = C



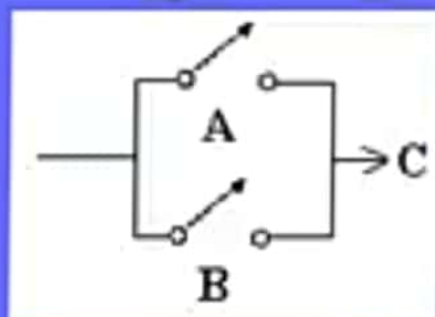


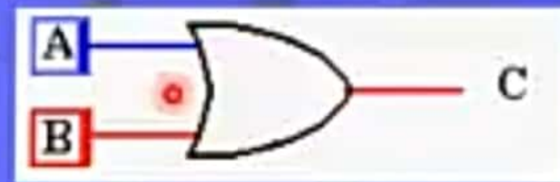


A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

LOGIC GATES: *OR*

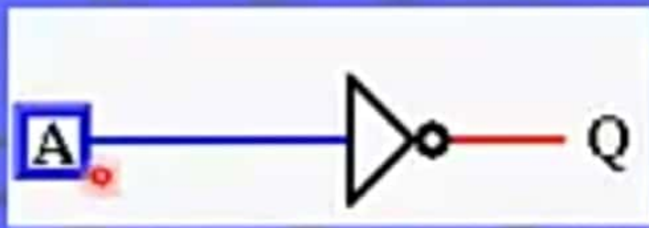
- Current flows if either switch is closed
  - Logic notation A + B = C





A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

GATES: Inversion (NOT)

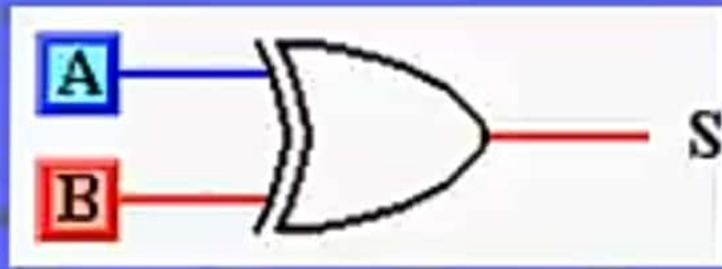


A	Q
0	1
1	0

Logic: Q = A'

Q is said to be the complement of A and is denoted as either $\overline{A}$ or A'

GATES: Exclusive OR (XOR



<u>Either</u> A or B, but not both, should be TRUE

This is sometimes called the **inequality detector**, because the result will be 0 when the inputs are the same and 1 when they are different.

A	B	S
0	0	0
1	0	1
0	1	1
1	1	0

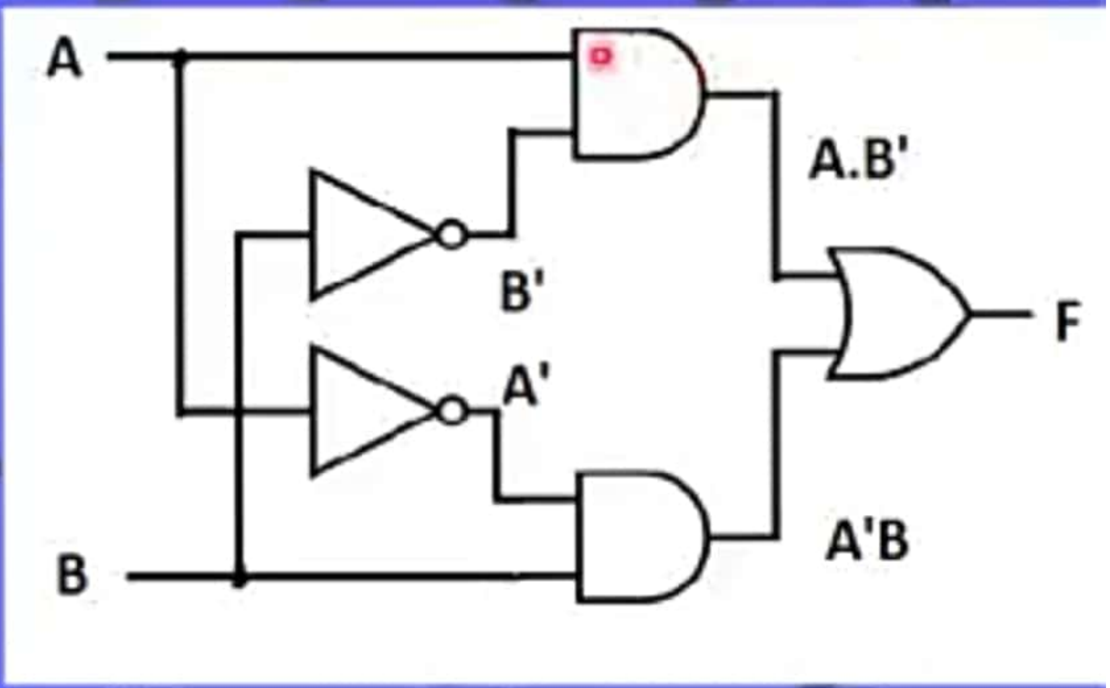
Using Logic Gates

How can we realize a function say,
F = A.B' + A'B using such logic gates?

AND OR AND

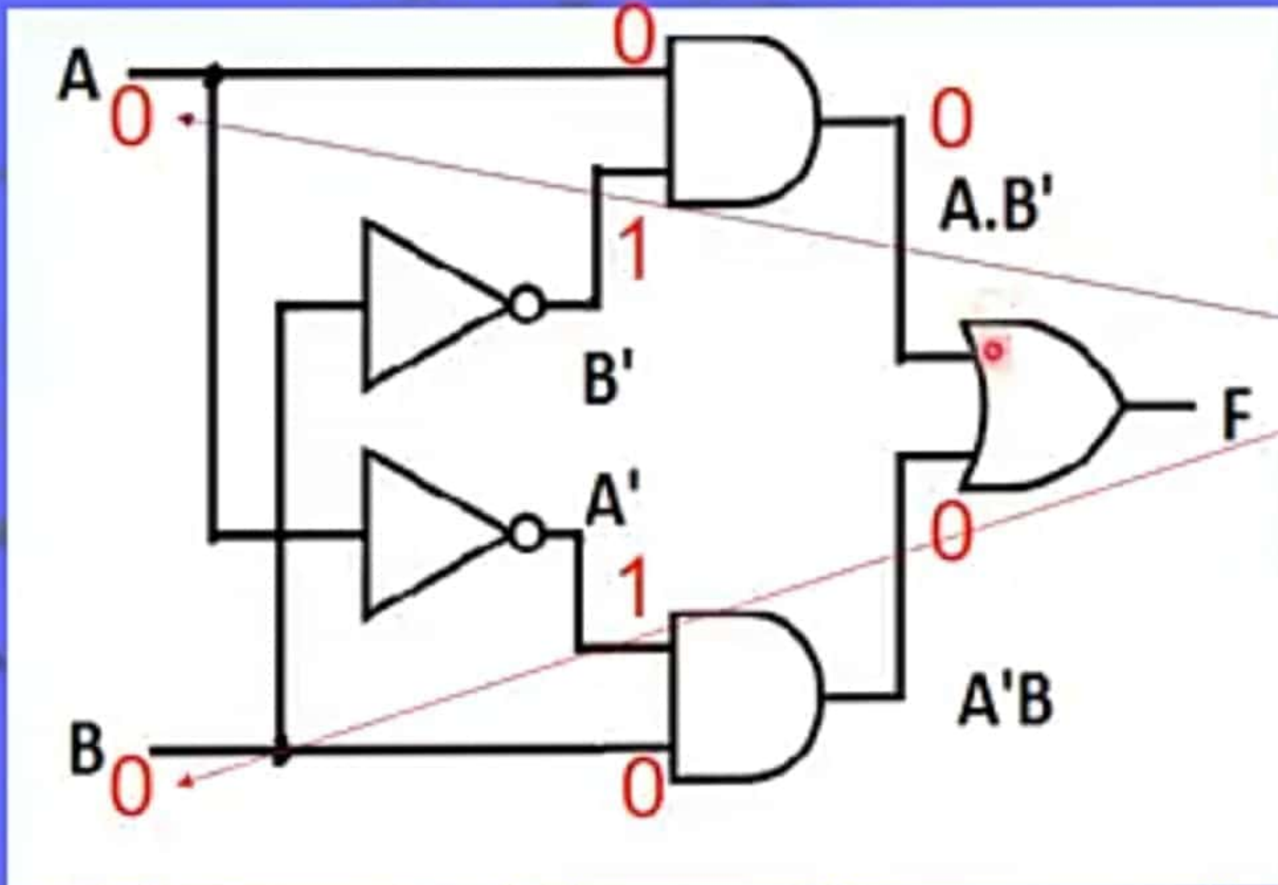
NOT NOT

Requirement:
2 AND
2 NOT
1 OR



Tuesday, March 9, 2021

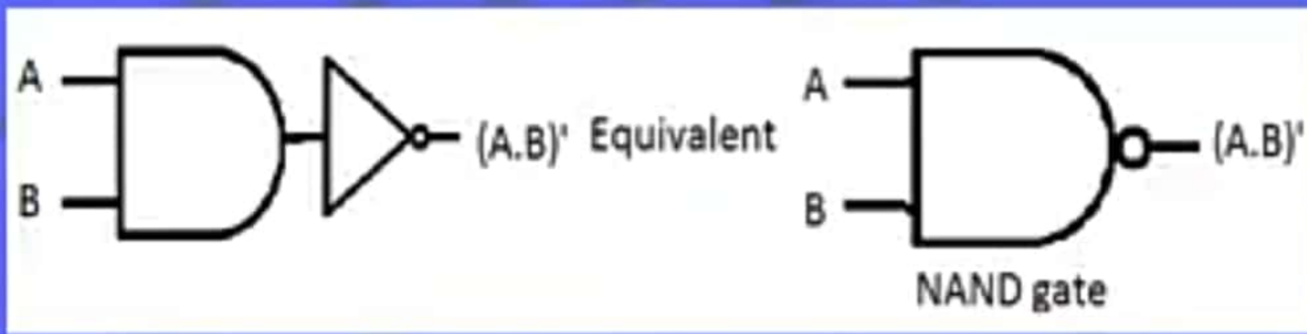
Guess which truth table thi circuit satisfies?



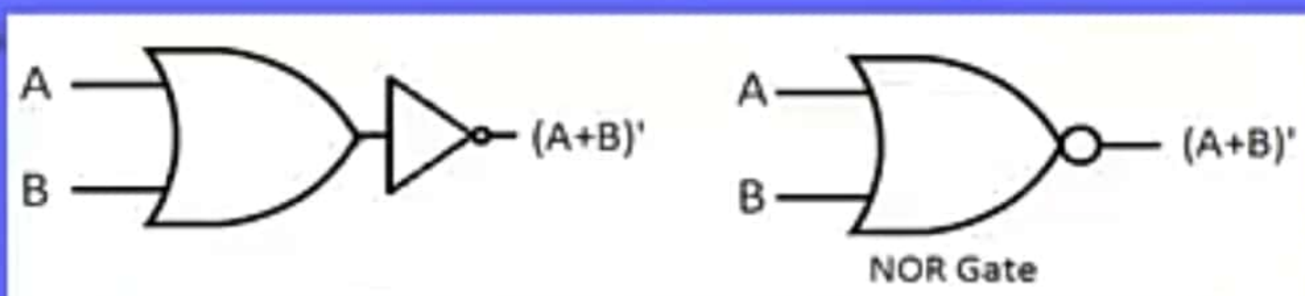
A	B	F
0	0	0
1	0	1
0	1	1
1	1	0

Fill the rest of the entries accordingly to find what it satisfies.

A quick peek into other logi gates



A	B	(A.B)'
0	0	1
0	1	1
1	0	1
1	1	0



A	B	(A+B)'
0	0	1
0	1	0
1	0	0
1	1	0