CS100 #04

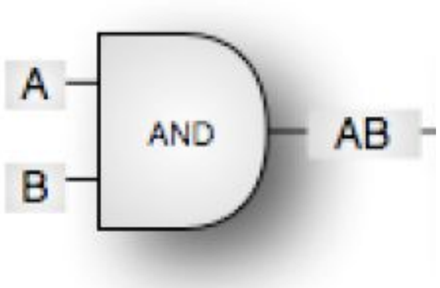
Logic Gates

Vadim Surov

Introduction

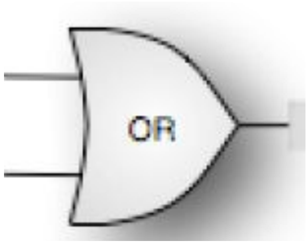
- We have looked at Boolean expressions in abstract terms.
- In this presentation, we see that Boolean functions are implemented in digital computer circuits called gates.
- A gate is an electronic device that produces a result based on two or more input values.
  - In reality, gates consist of one to six transistors, but digital designers think of them as a single unit.
  - Integrated circuits contain collections of gates suited to a particular purpose.

Logic Gates AND



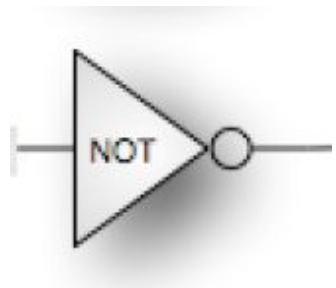
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Logic Gate OR



A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

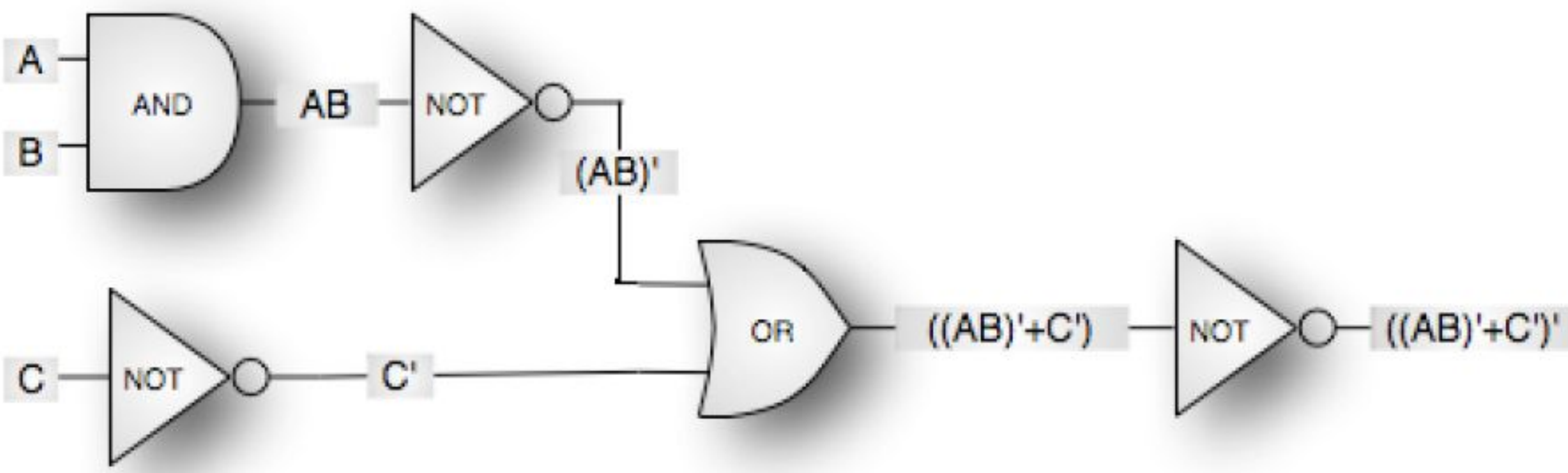
Logic Gate NOT



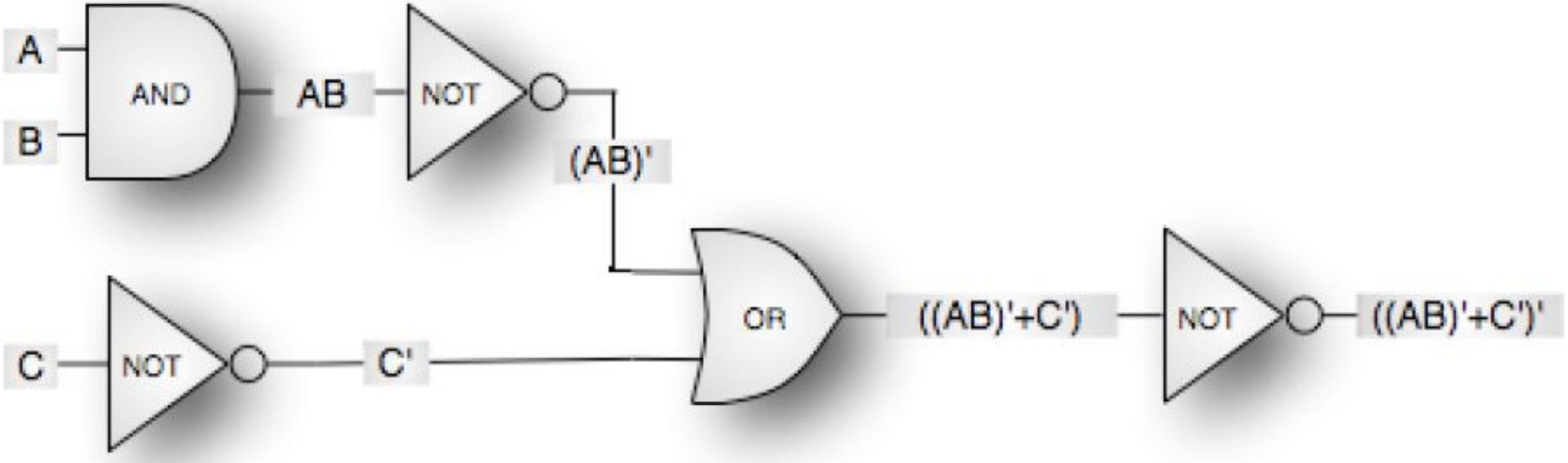
A	NOT A
0	1
1	0

Combinational Circuits

- The main thing to remember is that combinations of gates implement Boolean functions.
- The circuit below implements the Boolean expression



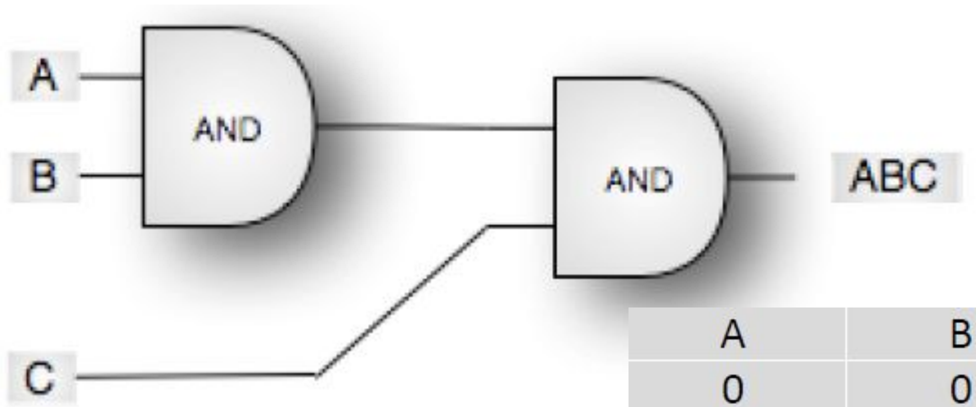
Combinational Circuits



A	B	C	AB	C’	AB’	((AB’)+c’)	((AB’)+c’)’
0	0	0	0	1	1	1	0
0	0	1	0	0	1	1	0
0	1	0	0	1	1	1	0
0	1	1	0	0	1	1	0
1	0	0	0	1	1	1	0
1	0	1	0	0	1	1	0
1	1	0	1	1	0	1	0
1	1	1	1	0	0	0	1

Combinational Circuits

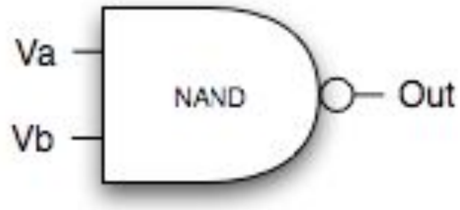
- Compare previous circuit to the following one, you will find that they are equivalent, they produce similar outputs:



A	B	C	AB	ABC
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1

Combining Logical Gate: NAND

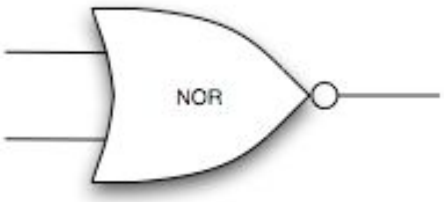
$(A \cdot B)'$



A	B	NAND
0	0	1
0	1	1
1	0	1
1	1	0

Combining Logical Gate: NOR

$$(A + B)'$$



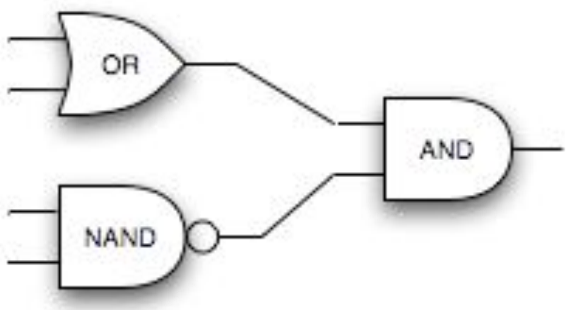
A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

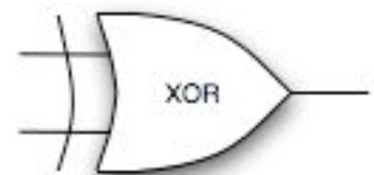
Combining Logical Gate: XOR

$$A \cdot B' + A' \cdot B$$

$$(A + B) \cdot (A' + B')$$

$$(A + B) \cdot (A \cdot B)'$$

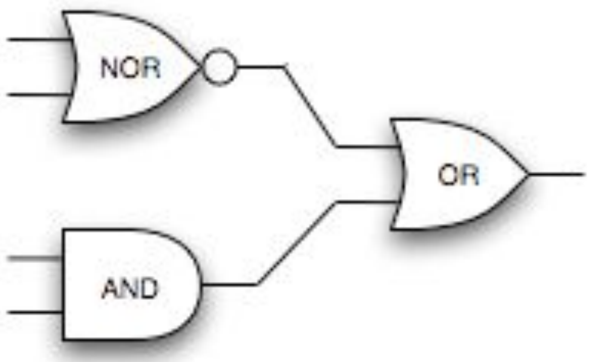


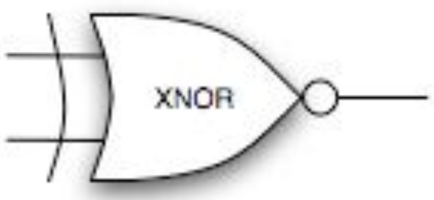


0	0	0
0	1	1
1	0	1
1	1	0

Combining Logical Gate: XNOR

$$(A + B)' + (A \cdot B)$$





A	*B*	*XNOR*
0	0	1
0	1	0
1	0	0
1	1	1