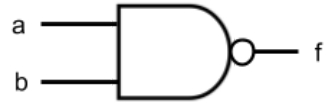
3.4 NAND / NOR (universal gates)

NAND

A ***NAND*** gate is the opposite (the NOT, hence the "N") of an AND gate, outputting 0 if all inputs are 1s; else the output is 1.

Figure 3.4.1: NAND truth table and gate.

a	b	f
0	0	1
0	1	1
1	0	1
1	1	0



PARTICIPATION ACTIVITY 3.4.1: NAND gates.

1) 0 NAND 1 = ?
   - 1
   - 0

2) 1 NAND 1 = ?
   - 1
   - 0

3) 0 NAND 0 = ?
   - 1

○ 0

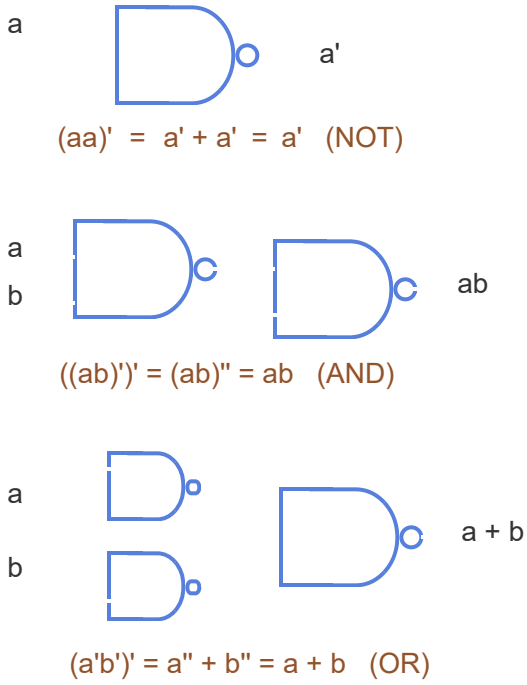
A NAND gate is a universal gate

NAND gates are popular due to having a simpler CMOS transistor circuit implementation than AND gates: Recall that an AN from a NAND transistor circuit followed by a NOT circuit.

Furthermore, NAND gates are popular due to being a universal gate. A ***universal gate*** is a single gate type that can impleme combinational circuit. NAND can implement NOT, AND, and OR, as shown below, and is thus universal.

PARTICIPATION ACTIVITY 3.4.2: NAND is a universal gate.

Start ☐ 2x speed



PARTICIPATION ACTIVITY 3.4.3: Universal gates.

1) A NAND gate is a universal gate.
   - ○ True
   - ○ False

2) A NAND gate cannot implement a NOT gate.
   - ○ True
   - ○ False

3) Inverting the output of a NAND gate produces an AND gate.
   - ○ True
   - ○ False

4) Inverting the inputs of a NAND gate produces an OR gate.
   - ○ True
   - ○ False

Converting to NAND gates

NAND being a universal gate enables chip makers to pre-fabricate a chip consisting of millions of NAND gates. Any circuit can be implemented simply by adding wires. Pre-fabricating the chip with AND, OR, and NOT gates would involve complexi how many of each gate to pre-fabricate, and where to place each gate type. Using NAND is much simpler.

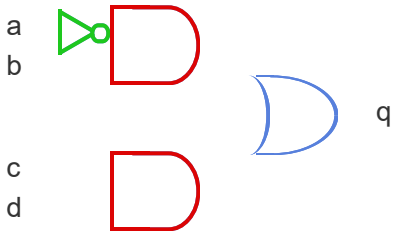
A chip with pre-fabricated gates is sometimes called a ***gate-array ASIC***. ***ASIC*** is short for Application-Specific Integrated Ci

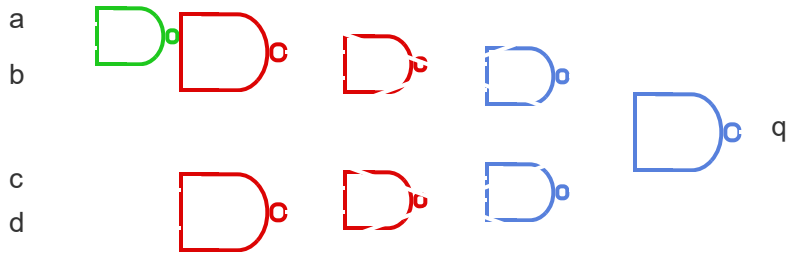
Converting an AND/OR/NOT circuit to a NAND-only circuit enables implementation using fewer transistors as well as enab implementation on a gate-array ASIC. The conversion can be done simply by replacing each AND, OR, and NOT gate by the structure of NAND gates, then removing double-inversions.

PARTICIPATION ACTIVITY 3.4.4: Converting to a NAND-only implementation.

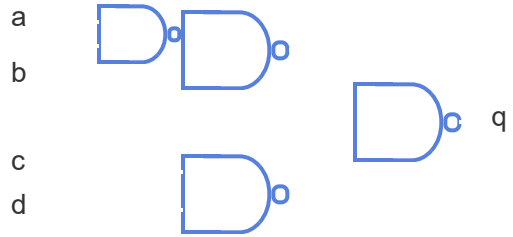
Start 2x speed

q = a'b + cd



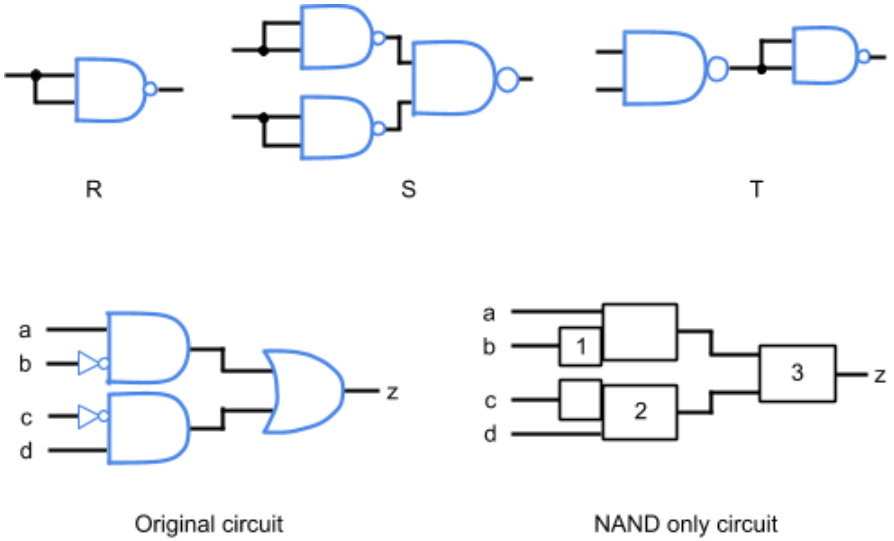


Direct replacement



Simplified circuit

PARTICIPATION ACTIVITY 3.4.5: Converting to NAND gates.



Consider the figure above. Which NAND pattern goes into which box?

1) Box 1

Check Show answer

2) Box 2

Check Show answer

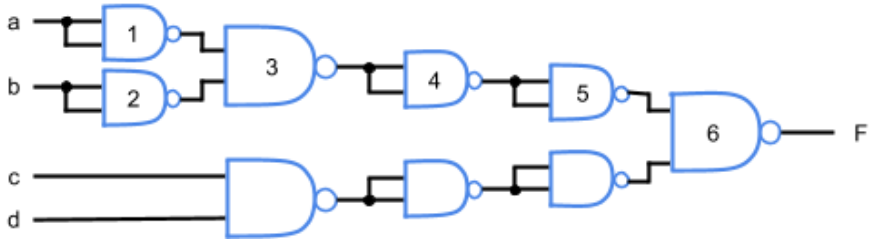
3) Box 3

Check Show answer

PARTICIPATION

3.4.6: Simplifying NAND circuits.

ACTIVITY



1) Which pair of NAND gates can be eliminated?

- 1, 2
- 3, 4
- 4, 5

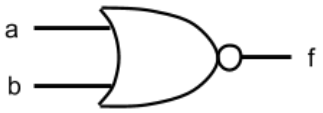
NOR

A ***NOR*** gate is the opposite of an OR gate, outputting 0 if any of the inputs are 1s; else the output is 1.

A discussion analogous to the above NAND discussion exists for NOR. Such discussion is omitted here. Briefly, NOR's trans simpler than OR's. NOR is also a universal gate. NOT: (a + a)' = a'a' = a' (NOR with inputs tied together). OR: ((a + b)')' = (a + b followed by NOT). AND: (a' + b')' = a"b" = ab (NOR with each input NOTed).

Figure 3.4.2: NOR truth table and gate.

a	b	f
0	0	1
0	1	0
1	0	0
1	1	0



PARTICIPATION ACTIVITY 3.4.7: NOR gates.

1) 0 NOR 0 = ?
   - 1
   - 0

2) 1 NOR 1 = ?
   - 1
   - 0

3) 0 NOR 1 NOR 1= ?
   - 1
   - 0

4) A NOR gate is a universal gate.
   - True
   - False

5) An AND gate is a universal gate.
   - True
   - False

6) A NOT gate is a universal gate.
   - True
   - False

Provide feedback on this section