Another example of truth table

- Logic function:
  - **3 inputs**: A, B, C
  - **3 outputs**: D, E, F
  - D = T if at least 1 input is T
  - E = T if 2 inputs are T
  - F = T if 3 inputs are T

Inputs			Outputs		
A	B	C	D	E	F
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	0	1

Truth tables

- Completely describe any function
- can get big quickly
- difficult to interpret the function

Boolean algebra

- Alternative to truth table
- Variables have 0 or 1 values
- 3 operators: OR, AND, NOT
- OR (+) : A + B
  - 1 if at least one input is 1
  - *logical sum*
- AND (•) : $A \bullet B$
  - 1 if both inputs are 1
  - *logical product*
- NOT (¯) : $\bar{A}$
  - 1 if input is 0
  - *inversion*
- Gates implement these functions

Laws of boolean algebra

Identity law: $A + 0 = A$ and $A \cdot 1 = A$.

Zero and One laws: $A + 1 = 1$ and $A \cdot 0 = 0$.

Inverse laws: $A + \overline{A} = 1$ and $A \cdot \overline{A} = 0$.

Commutative laws: $A + B = B + A$ and $A \cdot B = B \cdot A$.

Associative laws: $A + (B + C) = (A + B) + C$ and $A \cdot (B \cdot C) = (A \cdot B) \cdot C$.

Distributive laws: $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ and
$A + (B \cdot C) = (A + B) \cdot (A + C)$.

Example

- Logic function
  - 3 inputs: A, B, C
  - 3 outputs: D, E, F
  - D = T if at least 1 input is T
  - E = T if 2 inputs are T
  - F = T if 3 inputs are T

$$D = A + B + C$$

$$E = ((A \cdot B) + (A \cdot C) + (B \cdot C)) \cdot (\overline{A \cdot B \cdot C})$$ (what can be true *and* what cannot be)

$$E = (A \cdot B \cdot \overline{C}) + (A \cdot C \cdot \overline{B}) + (B \cdot C \cdot \overline{A})$$ (exactly two inputs are true)

$$F = A \cdot B \cdot C$$

Gates

- Gates implement circuits for logic functions

FIGURE B.2.1 Standard drawing for an AND gate, OR gate, and an inverter, shown from left to right. The signals to the left of each symbol are the inputs, while the output appears on the right. The AND and OR gates both have two inputs. Inverters have a single input.

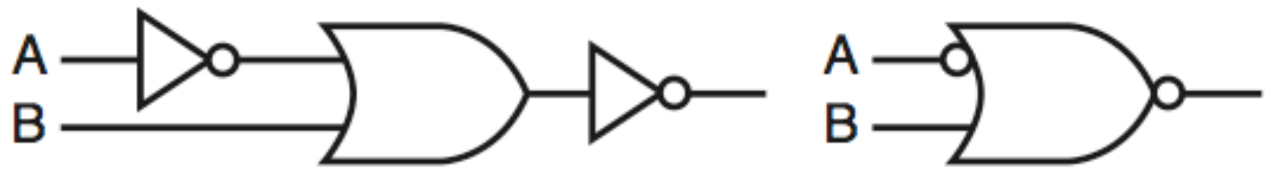
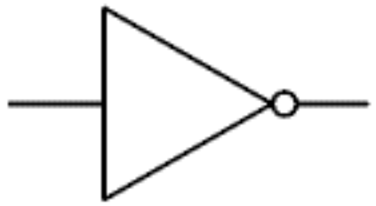


FIGURE B.2.2 Logic gate implementation of $\overline{\bar{A} + B}$ using explicit inverts on the left and using bubbled inputs and output on the right. This logic function can be simplified to $A \cdot \bar{B}$ or in Verilog, `A & ~ B`.

Any logic function can be implemented by using AND, OR gates and inversions

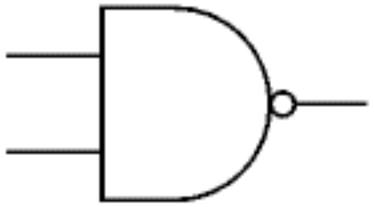
More gates

- NOT gate



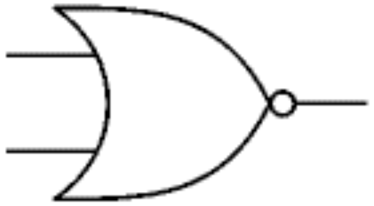
Input A	Output Q
0	1
1	0

- NAND gate: inverse of AND gate



Input A	Input B	Output Q
0	0	1
0	1	1
1	0	1
1	1	0

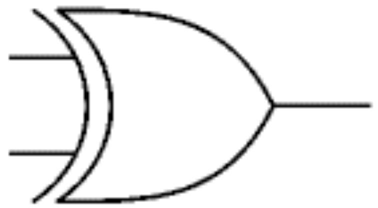
- NOR gate: inverse of OR gate



Input A	Input B	Output Q
0	0	1
0	1	0
1	0	0
1	1	0

More gates

- XOR gates: different inputs = positive output



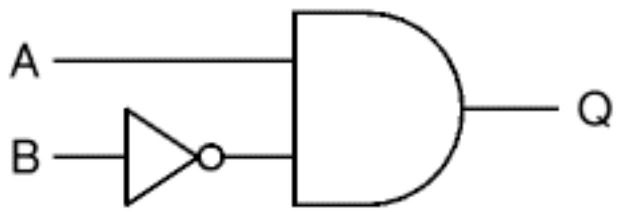
Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	0

- Summary truth tables

Summary for all 2-input gates							
Inputs		Output of each gate					
A	B	AND	NAND	OR	NOR	EX-OR	EX-NOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

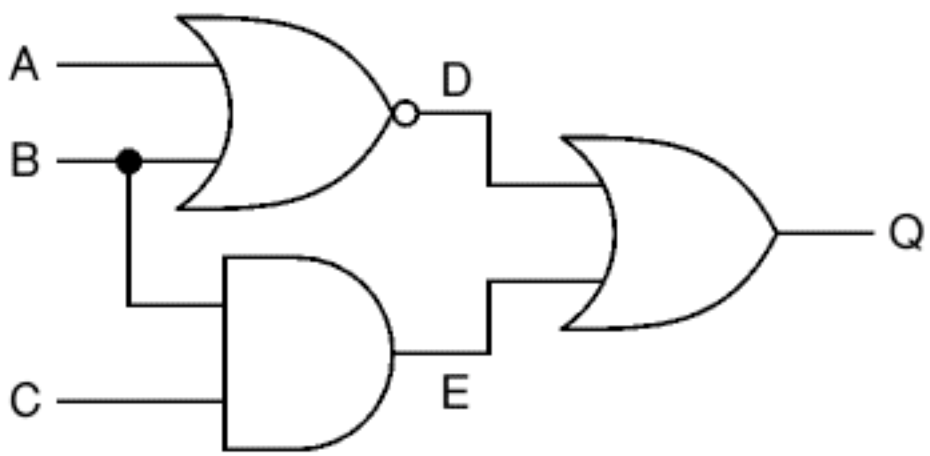
Summary for all 3-input gates						
Inputs			Output of each gate			
A	B	C	AND	NAND	OR	NOR
0	0	0	0	1	0	1
0	0	1	0	1	1	0
0	1	0	0	1	1	0
0	1	1	0	1	1	0
1	0	0	0	1	1	0
1	0	1	0	1	1	0
1	1	0	0	1	1	0
1	1	1	1	0	1	0

Combination of logic gates



Q = A AND NOT B

Input A	Input B	Output Q
0	0	0
0	1	0
1	0	1
1	1	0



D= NOT (A OR B)
E = B AND C
Q = D OR E

Inputs			Outputs		
A	B	C	D	E	Q
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	0	0	0
0	1	1	0	1	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	1	1

What will be the output?