Digital Logic Structures

Joannah Nanjekye

July 10, 2024

Outline

Boolean Algebra

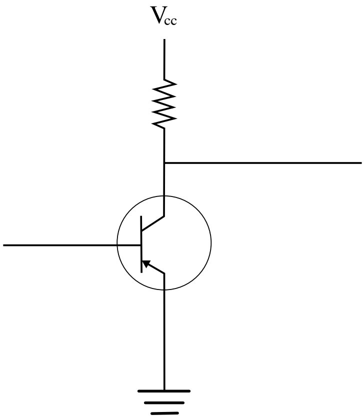
- A mathematical discipline used in the analysis and design of digital circuits
- It uses logical variables and operations
- Variables:
  - 1 (TRUE)
  - 0 (FALSE)
- Operations:
  - A AND B = $A \cdot B$
  - A OR B = $A + B$
  - NOT A = $\overline{A}$

Identities of Boolean Algebra

Basic Postulates		
$A \cdot B = B \cdot A$	$A + B = B + A$	Commutative Laws
$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$	$A + (B \cdot C) = (A + B) \cdot (A + C)$	Distributive Laws
$1 \cdot A = A$	$0 + A = A$	Identity Elements
$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$	Inverse Elements
Other Identities		
$0 \cdot A = 0$	$1 + A = 1$	
$A \cdot A = A$	$A + A = A$	
$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$	Associative Laws
$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$	DeMorgan's Theorem

The Transistor

A transistor is a semiconductor device used to amplify and switch electronic signals



Logic Gates

Electronic circuits that produce output signals that are Boolean operations on their input signals

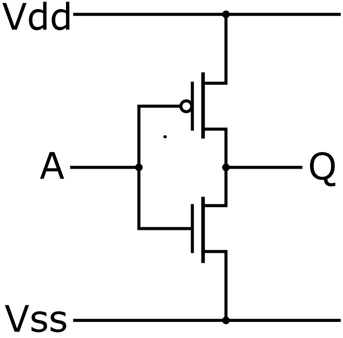
When the gate input changes, the output is instant unless a gate delay occurs as a result of signal propagation time

Digital logic uses the following gates:

- AND
- OR
- NOT
- NAND
- NOR
- XOR

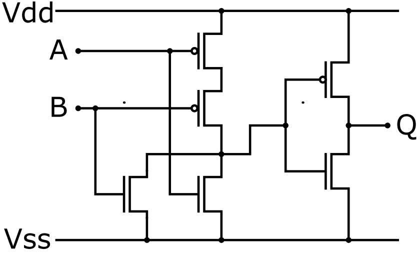
The NOT Gate (inverter)

A	$\overline{A}$
0	1
1	0



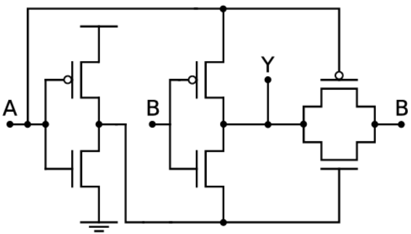
The OR Gate

A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1



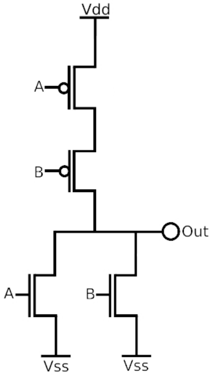
The XOR Gate

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



The NOR Gate

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0



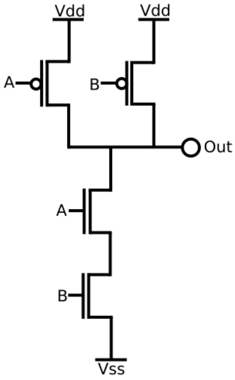
The AND Gate

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

How would you construct an AND gate?

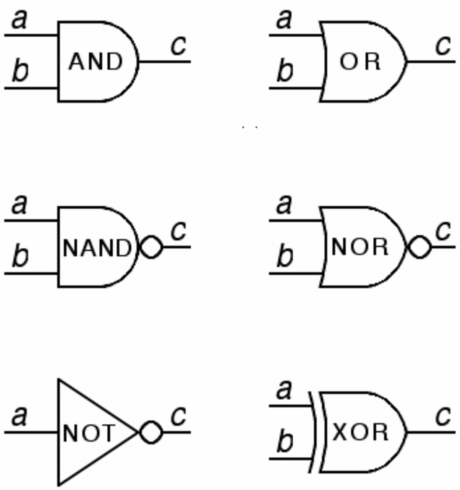
The NAND gate

A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0



Logic Gate Symbols

Assert a signal is to cause a signal line transition from its logically false (0) state to its logically true (1) state



DeMorgan's Law

$$\overline{AB} = \overline{A} + \overline{B}$$

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

Show the relationship between AND and OR gates.
AND, OR, NAND, and NOR gates may have more than two inputs.

Complete Sets of gates

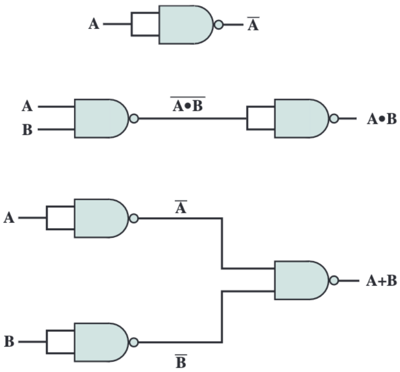
Any Boolean function can be implemented using a subset of gates

We need to identify a functionally complete set of gates

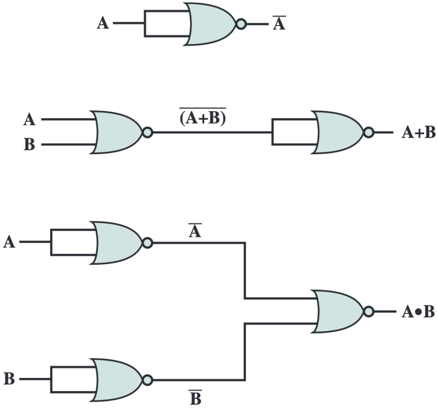
The following sets of gates are functionally complete

- AND, OR, NOT
- AND, NOT
- OR, NOT
- NAND
- NOR

Using NAND Gates



Using NOR Gates



Combinational Circuits

- An interconnected set of gates
- The output at any time is a function only of the input at that time
- The output is determined by the input
- Examples
  - Adder
  - Decoder
  - MUX

Design of Combinational Circuits

- Problem statement
- Formal specification
  - Truth table
  - Boolean equation
  - Graphical symbols: hardware description language (eg VHDL)
- Minimize
- Implementation

Design of Combinational Circuits

Example

A car key alert should be turned on under the following circumstances:

- The key is in and the door is open;
- The key is not in and the light is on.

The system has three inputs:

- L - the lights are on
- K - the key is in
- D - the door is open

The system has one output:

- B - the buzzer will be activated

Description to Truth Table

From the description we can obtain the truth table

L	K	D	B
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Sum of Products

The Sum of Products (SOP) has one term for each 1 in the output

L	K	D	B
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$$B = \overline{L}KD + L\overline{K}\,\overline{D} + L\overline{K}D + LKD$$

Product of Sums

The Product of Sums (POS) has one term for each 0 in the output

L	K	D	B
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$$(\overline{L} + \overline{K} + \overline{D}) \cdot (\overline{L} + \overline{K} + D) \cdot (\overline{L} + K + \overline{D}) \cdot (L + K + \overline{D})$$

Definitions

Definition (Literal)

A variable X has two **literals** X and $\overline{X}$.

Definition (Product term)

A logical product where each variable is represented by at most one literal is a **product** or a **product term** or a **term.** A term can be a single literal.

Definition (Sum-of-products)

A logical sum of product terms forms a sum-of-products expression (SOP).

Definitions

Definition (Minterm)

A **minterm** is a logical product of n literals where each variable occurs as exactly one literal

Definition (Canonical SOP)

A **canonical SOP** is a logical sum of minterms, where all minterms are different. Also called canonical disjunctive form or minterm expansion.

Problem

- List the minterms;
  $B(L, K, D) = \Sigma m(3, 4, 5, 7)$
- Write the canonical SOP for the car buzzer problem;
  $B(L, K, D) = \overline{L}KD + L\overline{K}\overline{D} + L\overline{K}D + LKD$
- draw the circuit;
- how can it be minimized?
- show that $L\overline{K} + KD + LD = L\overline{K} + KD$

Simplification of a Boolean Expression

A Boolean function can be realized in either SOP or POS form

However we need a simpler Boolean expression that uses fewer gates:

- Algebraic simplification
- Karnaugh maps
- Quine-McCluskey tables

Algebraic Simplification

- Involves the use of boolean algebra identities
- To reduce the Boolean expression to one with fewer elements
- Or even simple observation, for example:

$$F = \overline{A}B + B\overline{C}$$

- Can be simplified as:

$$F = B(\overline{A} + \overline{C})$$