

Arquitetura e Organização de Computadores

Nesta disciplina, você estudará como a arquitetura e a organização dos computadores foi pensada. Será possível ter uma visão mais clara das diversas versões dos computadores, como funcionam seus componentes além de entender que, por mais diferentes que sejam sua tecnologia é baseada em padrões que há muito tempo são utilizados.

Fundamentos de Sistemas Computacionais - Primeira Unidade

Sistema Computacional: *A combinação entre o hardware e o software, ou seja, entre a parte física e a parte lógica dos computadores formam um sistema computacional.*

Conceitos Básicos

Os diversos componentes de um computador podem ser classificados de acordo com a função básica que possuem. Essa classificação possui a seguinte forma:

Unidade de Entrada: Teclados e Mouses

Unidade de Saída: Telas e Impressoras

Unidade de Armazenamento: Memórias

Unidade de Processamento: Processadores

Desenvolvimento Histórico

As gerações de computadores foram classificadas de acordo com a tecnologia empregada e com seu desempenho. Essa classificação possui a seguinte forma:

Geração Zero (Antes de 1946): Computadores Analógicos

1º Geração (Entre 1946 e 1954): Válvulas

2º Geração (Entre 1954 e 1964): Transistores

3º Geração (Entre 1964 e 1977): Circuitos Integrados

4º Geração (Entre 1977 e 1991): Microprocessadores

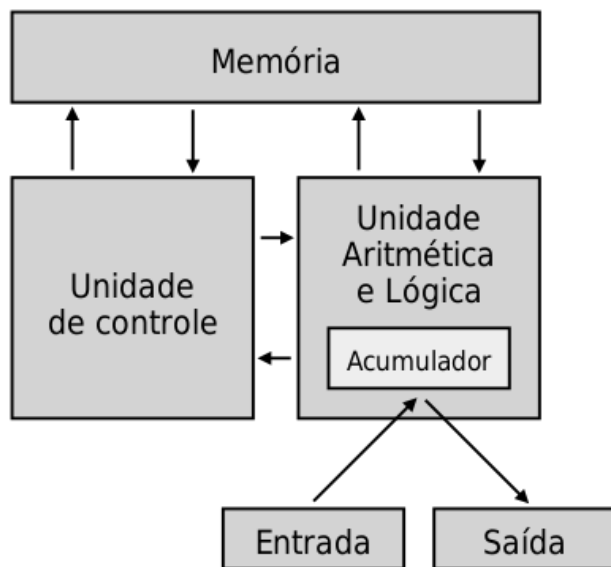
5º Geração (Depois de 1991): Internet

Lei de Moore: *Em 1965, Gordon Moore, afirmou que a densidade de transistores em um circuito integrado, iria dobrar a cada doze meses. Na realidade, essa densidade dobrou a cada 18 meses em média.*

Estrutura Básica de um Computador

Após o final da Segunda Guerra Mundial, o pesquisador e cientista John Von Neumann introduziu a ideia de um computador com uma Unidade Central de Processamento ou CPU. A arquitetura utilizada na CPU deste computador permitia o armazenamento de dados e programas na mesma unidade de memória por meio de seus endereçamentos. Isso deu origem aos primeiros processadores, que evoluíram até os dias atuais.

Essa arquitetura ficou conhecida como **Arquitetura de Von Neumann**:



Gargalo de Von Neumann: A via de transmissão de dados entre a CPU e a memória principal limita de certa forma a velocidade do processamento de um computador. Os Barramentos têm essa função e a troca de dados entre o processador e memória fica limitada pela taxa de transferência de dados que eles são capazes de proporcionar.

Unidade Central de Processamento - Segunda Unidade

Unidade Central de Processamento

A Unidade Lógica e Aritmética (ULA) é responsável por executar os cálculos matemáticos utilizados para processar os dados dentro do computador. Os dados usados para estes cálculos são armazenados na memória do computador, e o caminho para que esta informação seja conduzida entre a ULA e a memória são os Barramentos.

Tipos de Barramentos

Barramento de Dados: Interliga a CPU e a memória, fazendo a transferência das informações que serão processadas. Ele determina diretamente o desempenho do sistema, pois quanto maior o número de vias de comunicação, maior rapidez com que estes dados são processados.

Barramento de Endereços: Interliga a CPU e a memória fazendo seus endereçamentos e tem o número de vias correspondente à tecnologia do processador.

Barramento de Controle: Interliga a Unidade de Controle aos componentes e dispositivos de um computador.

A Unidade de Controle tem a função de coordenar e direcionar as principais funções de um computador. É responsável por coordenar a ULA, os registradores que controlam as memórias, os barramentos internos e todo o funcionamento da Motherboard. Os processadores têm dois tipos de arquiteturas empregadas pelos seus fabricantes:

Processadores CISC: Sistema com um conjunto de instruções complexo.

Processadores RISC: Os processadores com tecnologia RISC tem um conjunto reduzido de instruções e diferentemente da tecnologia CISC, estas instruções são consideradas complexas, pois cada uma executa várias tarefas conjuntas. Isso permite uma vantagem dos processadores RISC em relação aos processadores CISC, que por ter um número menor de instruções tem menos circuitos internos e assim podem trabalhar com frequências muito maiores sem ter problemas de superaquecimento.

Memórias

Existem vários tipos de memórias e elas podem ser classificadas em Memória Principal e Memória Secundária. Além destes dois tipos, ainda temos a Memória Cache e os Registradores. As memórias podem ser voláteis, que se apagam quando o computador é desligado ou não voláteis sendo previamente gravadas e não se apagam quando se desliga o computador, o que permite que os dados gravados sejam lidos posteriormente.

Registradores

O processador recebe os dados e os deposita temporariamente na memória e os registradores são os locais de memória onde estes dados ficam armazenados para que o processamento aconteça. É um tipo de memória volátil que se apaga quando o computador é desligado. Por estar dentro do processador, proporciona uma velocidade de transferência alta e capacidade de armazenamento geralmente muito baixa.

Memória Cache

Quando entramos com dados em um computador, o processador busca estes dados em uma memória externa e como a velocidade dos processadores é em geral muito maior do que as velocidades das memórias gera um congestionamento entre os dados encontrados na memória e o processador. Para poder solucionar este problema foi desenvolvida a Memória Cache, que tem a função de criar condições que aumentem a velocidade de comunicação entre esses componentes, aumentando consideravelmente a velocidade final do processamento. Este tipo de memória também é do tipo volátil.

Memória Principal (RAM)

A memória principal de um computador é chamada de memória RAM. Essa memória permite ao processador ter acesso às memórias secundárias, disponibilizando os dados gravados nelas para serem processados. Esse tipo de memória é do tipo volátil.

Memória ROM

A memória ROM também é uma memória principal do computador, mas com função apenas de leitura, onde seu conteúdo é gravado apenas uma vez e não é alterado.

Memórias Secundárias

As memórias secundárias são responsáveis por armazenar dados para o uso posterior, pois elas não se apagam quando o computador é desligado, são do tipo não volátil e podem ser alteradas e regravadas quantas vezes for necessário. Outra característica das memórias secundárias é que elas não são endereçadas diretamente pelo processador, por este motivo os dados armazenados, nestas memórias, precisam ser carregados na memória principal para serem processados. A memória secundária por possuir uma capacidade de armazenamento muito superior a das outras memórias. A principal memória secundária que temos hoje em dia são os famosos discos rígidos.

Dispositivos de Entrada e Saída

Há diversos dispositivos de entrada e saída de dados que também são chamados de periféricos e a cada dia surgem novos equipamentos que fazem essa função. Existem também muitas diferenças de características entre esses diversos periféricos, por exemplo, a velocidade de transferência de um teclado ou de um mouse é muito menor do que a velocidade de um HD. Por este motivo foram criados novos tipos de barramentos, com taxas de transferência de bits diferentes. Onde podemos citar:

Barramento Local: Funciona na mesma velocidade do processador e Interliga o mesmo aos dispositivos com maior velocidade.

Barramento de Expansão: Também chamado de barramento de entrada e de saída, é responsável por interligar os dispositivos de entrada e saída aos demais componentes.

Sistemas Numéricos - Terceira Unidade

Principais Sistemas Numéricos

Sistema Decimal

O sistema de numeração decimal é composto por 10 símbolos com os quais podemos formar dígitos. Os números iniciam em zero e vão até nove. Daqui para frente à sequência de repetição começa em ordem crescente e seguindo os símbolos da base. Com isso podemos representar as casas decimais de um numero e podemos afirmar que qualquer numero é simplesmente uma soma de produtos do valor posicional ou de seu peso. Os pesos são representados pela sequência padrão como potencia de base 10. Esta potencia é inserida da direita para esquerda sobre as bases do numero. Por exemplo:

$$387_{10} = (3 \times 10^2) + (8 \times 10^1) + (7 \times 10^0) = 300 + 80 + 7 = 387_{10}$$

Sistema Binário

Esse sistema é baseado apenas nos símbolos 0 e 1 e é amplamente usado pelos computadores. O zero representa a ausência de tensão, enquanto o um representa a presença de tensão. Números binários que são chamados de BITS (**Oito Bits = Um Byte**) (**Um Byte = Um Caractere**) e são representados sempre com a base dois.

Sistema Octal

No sistema octal, os números são representados por oito símbolos, os números iniciam em zero e vão até sete e a escrita de um número octal fica representada com a base 8. O sistema octal foi usado como alternativa ao binário como uma forma mais enxuta e compacta na utilização das antigas linguagens de máquina. Hoje se utiliza mais o sistema hexadecimal como um meio viável ao sistema binário.

Sistema Hexadecimal

16 Símbolos - Base 16

Tabela de Correspondência Entre os Sistemas

0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Conversão Entre Bases Numéricas

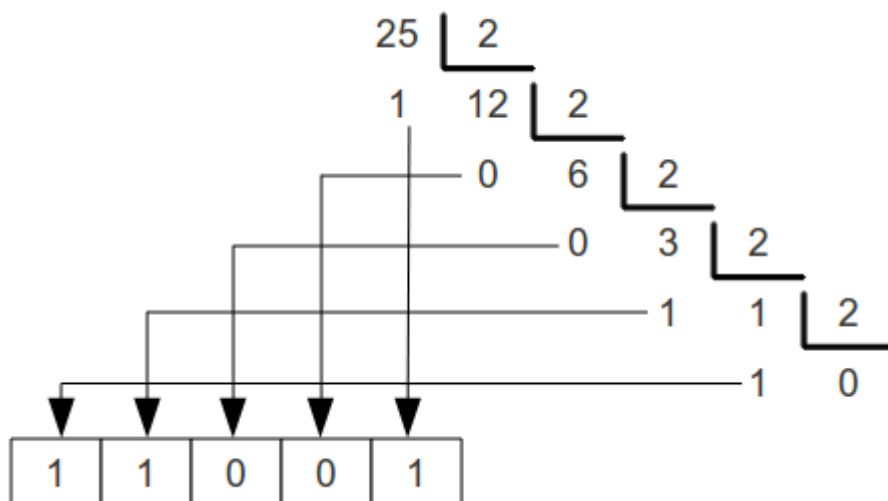
Conversão de Decimal para Binário

Fazer a divisão sucessiva por dois até que o quociente chegue à zero.

Restos sempre serão zero ou um.

O Quociente só recebe o numero inteiro, caso o resultado der uma fração.

Quando o quociente chegar à zero, se pegam os restos de baixo para cima.



Conversão de Binário para Decimal

Separe os números binários.

Acima de cada um deles coloque a potência, numerando da direita para a esquerda.
Escreva o número binário correspondente multiplicado pela base 2 na potência.
Some os valores para chegar ao resultado decimal.

11001_2

$$(1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 16 + 8 + 0 + 0 + 1 = 25_{10}$$

Conversão de Decimal para Hexadecimal

Fazer a divisão sucessiva por 16.

O Quociente só recebe o número inteiro, caso o resultado der uma fração.

Restos sempre serão números menores que 16.

Quando o quociente for menor que 16, utilizam-se ele e os restos de baixo para cima.

Substitua os valores entre 10 e 15 pelo seu valor correspondente em hexadecimal.

$$\begin{array}{r|l}
 2002 & 16 \\
 \hline
 40 & 125 \\
 & \underline{13} \\
 82 & \\
 & \underline{2}
 \end{array}$$

$7D2_{16}$

Conversão de Hexadecimal para Decimal

Separe os números e acima de cada um deles coloque a potência.

Escreva o número hexadecimal correspondente multiplicado pela base 16 na potência.

Realize a troca dos símbolos pelo valor decimal correspondente.

Some os valores para chegar ao resultado decimal.

$E9317_{16} = E \ 9 \ 3 \ 1 \ 7$

$$(E \times 16^4) + (9 \times 16^3) + (3 \times 16^2) + (1 \times 16^1) + (7 \times 16^0)$$

$$(14 \times 65536) + 36864 + 768 + 16 + 7$$

$$917504 + 36864 + 768 + 16 + 7 = 955.159_{10}$$

Conversão de Decimal para Octal

Fazer a divisão sucessiva por oito até que o quociente chegue à zero.

O Quociente só recebe o número inteiro, caso o resultado der uma fração.

Restos sempre estarão ente 0 e 7, pois agora trabalhamos com base 8.

Quando o quociente chegar à zero, se pega os restos de baixo para cima.

$$29 \div 8 = 3 \text{ (Resto 5)}$$

$$3 \div 8 = \emptyset \text{ (Resto 3)}$$

$$29_{10} = 35_8$$

Conversão de Octal para Decimal

Separe os números octais e acima de cada um deles coloque a potência.

Escreva o número octal correspondente multiplicado pela base oito na potência.

Some os valores para chegar ao resultado decimal.

$$167_8 = (1 \times 8^2) + (6 \times 8^1) + (7 \times 8^0) = 64 + 48 + 7 = 119_{10}$$

Conversão de Binário para Hexadecimal

Convertendo entre as bases

Primeiro convertemos do binário para o decimal.

Depois pegamos decimal e convertemos para hexadecimal.

$$110100111100_2$$

$$(1 \times 2^{11}) + (1 \times 2^{10}) + (0 \times 2^9) + (1 \times 2^8) + (0 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = 2048 + 1024 + 256 + 32 + 16 + 8 + 4 = 3388_{10}$$

$$3388 \div 16 = 211 \text{ (Resto 12)} / 211 \div 16 = 13 \text{ (Resto 3)} / 13 \div 16 = \emptyset \text{ (Resto 13)}$$

$$110100111100_2 = D3C_{16}$$

Conversão Direta

Separe o número binário a ser convertido em grupos de quatro.

Caso o último grupo não tenha quatro dígitos, complete com zeros a esquerda.

Cada grupo corresponde na tabela a um valor no sistema hexadecimal.

Procure cada grupo de quatro dígitos na tabela de valores.

$$110100111100_2 = (1101) (0011) (1100)$$

$$1101 = D / 0011 = 3 / 1100 = C$$

$$110100111100_2 = D3C_{16}$$

Conversão de Hexadecimal para Binário

Convertendo entre as bases

Converter o número hexadecimal para decimal.

Pegar o resultado do valor decimal e convertê-lo para binário.

$$CA5_{16} = C A 5$$

$$(C \times 16^2) + (A \times 16^1) + (5 \times 16^0)$$

$$(12 \times 256) + (10 \times 16) + 5 = 3072 + 160 + 5 = 3237_{10}$$

$$3237_{10} \div 2 \text{ Sucessivamente} = 110010100101_2$$

$$CA5_{16} = 110010100101_2$$

Conversão Direta

Consulte a tabela de valores para realizar a conversão.

$$CA5_{16} = C A 5$$

$$(C = 1100) (A = 1010) (5 = 0101)$$

$$CA5_{16} = 110010100101_2$$

Convertendo de Binário para Octal

Separe o número binário em grupos de três dígitos, da direita para a esquerda.

Caso na esquerda não formou um grupo de três dígitos, complete com zeros à esquerda.

Fazer a correspondência de cada grupo binário com a tabela e achar seu valor em octal.

$$1011010_2 = (001) (011) (010) = 1 3 2 = 1011010_2 = 132_8$$

Sistema Binário	Sistema Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Convertendo de Binário para Octal

Separe os números octais.

Ache para cada dígito octal seu correspondente em binário através da tabela.

$$3716_8 = 3\ 7\ 1\ 6$$

$$011\ 111\ 001\ 110$$

$$3716_8 = 011111001110_2$$

Convertendo de Octal para Hexadecimal

Pegamos os dígitos do número octal e os separamos.

Achamos seu correspondente em binário.

Reescrevemos o número em binário separando agora em grupos de quatro bits.

Se no grupo da esquerda faltar dígitos, complete com zeros à esquerda.

Achamos na tabela de valores entre as bases seus correspondentes em hexadecimal.

$$1657_8 = 1\ 6\ 5\ 7$$

$$001\ 110\ 101\ 111 = 001110101111_2$$

$$0011\ 1010\ 1111 = 3\ A\ F$$

$$001110101111_2 = 3AF_{16}$$

Convertendo de Hexadecimal para Octal

Converter o número hexadecimal para binário.

Depois converter de binário para o sistema de numeração octal.

$$3AF_{16} = 3\ A\ F$$

$$(3 \times 16^2) + (A \times 16^1) + (F \times 16^0) = 768 + (10 \times 16) + (15 \times 1)$$

$$768 + 160 + 15 = 943_{10}$$

$$943_{10} \div 2 \text{ Sucessivamente} = 001110101111_2$$

$$001\ 110\ 101\ 111 = 1\ 6\ 5\ 7$$

$$3AF_{16} = 1657_8$$

Álgebra Booleana - Quarta Unidade

Introdução Álgebra de Booleana

A álgebra booleana usa **Variáveis** e **Operações Lógicas**. Essas variáveis podem ter o valor lógico **Verdadeiro** ou **Falso** e cada variável pode assumir apenas um valor.

Operações Lógicas Básicas	AND	OR	NOT
Simbologia Matemática	•	+	
Simbologia Computacional	\wedge	\vee	!

$$A \text{ AND } B = A \cdot B = A \wedge B$$

$$A \text{ OR } B = A + B = A \vee B$$

$$\text{NOT } A = \neg A = !A$$

AND → Produto Lógico

OR → Soma Lógica

NOT → Negação

A = Ana Dança

B = Ana Brinca

$A \wedge B$ = Ana Dança E Ana Brinca

$A \vee B$ = Ana Dança Ou Ana Brinca

$\neg A$ = Ana Não Dança

Tabela-Verdade AND

P	Q	$P \wedge Q$
0	0	0
0	1	0
1	0	0
1	1	1

Nessa **Tabela-Verdade** o valor de $P \wedge Q$ será verdadeiro sempre que P e Q também forem verdadeiros e negativo sempre que o valor de uma das variáveis for negativo.

Tabela-Verdade OR

P	Q	$P \vee Q$
0	0	0
0	1	1
1	0	1
1	1	1

Nessa **Tabela-Verdade** também usamos a mesma proposição. Mas nesse caso, basta que apenas uma das variáveis seja verdadeira para que **P V Q** seja verdadeiro e caso as duas variáveis forem negativas, o resultado sempre será negativo.

Tabela-Verdade NOT

P	!P
0	1
1	0

Nessa **Tabela-Verdade** o valor de **!P** se dá pela negação da variável. A negação sempre será o inverso do valor em questão. Como no seguinte exemplo:

P = 8

!P = Qualquer Número \neq 8.

Expressões Lógicas

O símbolo para **AND** e **OR** será alterado, usaremos **(•)** e **(+)** respectivamente. Na álgebra booleana, temos alguns termos que são usados com frequência, antes de vermos às operações matemáticas booleanas aplicadas no desenvolvimento de sistemas digitais, vamos conhecer e entender esses termos.

Variável: *Valor qualquer a ser utilizado*

Complemento: *Algo que agregará valor a uma variável*

Literal: *Representação de um valor fixo para uma variável*

Adição Booleana

Quando falamos em adição booleana, estamos falando da porta **OR**. Dentro da álgebra booleana, temos o chamado **Termo-Soma**, o que significa que é uma soma. O termo se chama **Termo-Soma**, porém, o símbolo de **(+)** indica a porta lógica **OR** e não o símbolo de adição que estamos acostumados. Logo Abaixo estão representados os dados da Tabela-Verdade OR, representados pelo seu símbolo lógico.

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1$$

Multiplicação Booleana

Quando falamos de multiplicação booleana, estamos falando da porta **AND**. Pela álgebra booleana, o **Termo-Produto** é o produto. O símbolo **(•)** representa a porta lógica **AND** e não o símbolo de multiplicação. Abaixo estão representados os dados da Tabela-Verdade AND, representados pelo seu símbolo lógico.

$$0 \bullet 0 = 0$$

$$0 \bullet 1 = 0$$

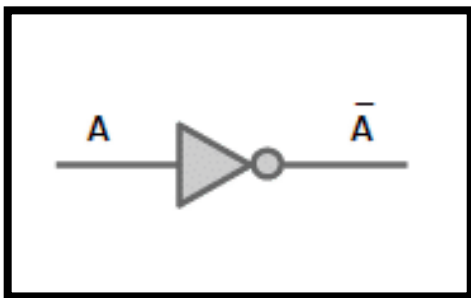
$$1 \bullet 0 = 0$$

$$1 \bullet 1 = 1$$

Portas Lógicas

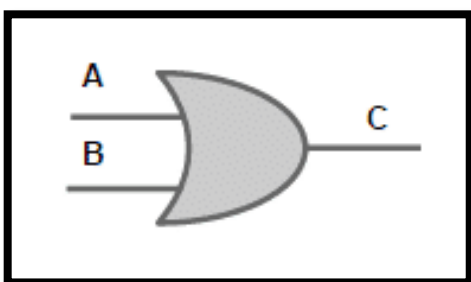
As portas lógicas são consideradas os elementos básicos da eletrônica digital. Os símbolos das portas lógicas são sempre representados por entradas lógicas e saídas lógicas. As entradas e saídas lógicas sempre assumirão os valores lógicos positivos ou negativos e sempre terão relação com as **Tabelas-Verdade**.

Inversor - Porta NOT



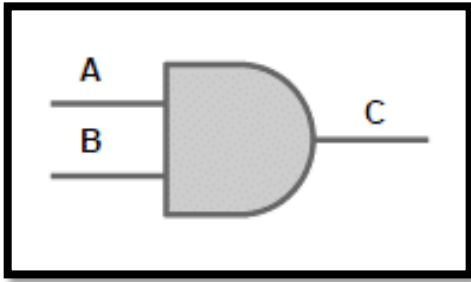
A	C
0	1
1	0

Porta OR



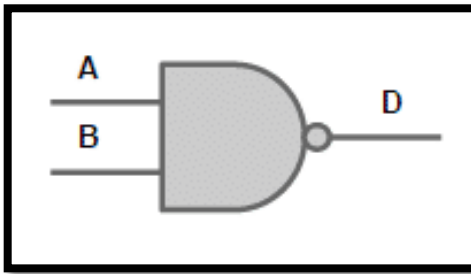
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

Porta AND



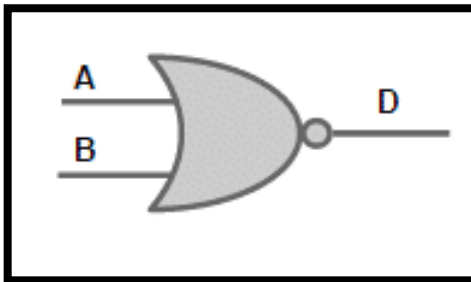
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

Porta NAND



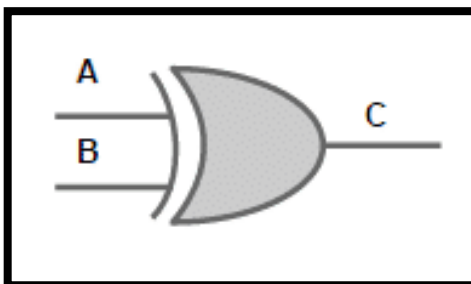
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

Porta NOR



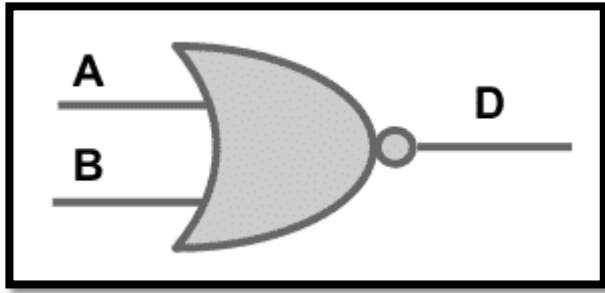
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

Porta XOR



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

Porta XNOR



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