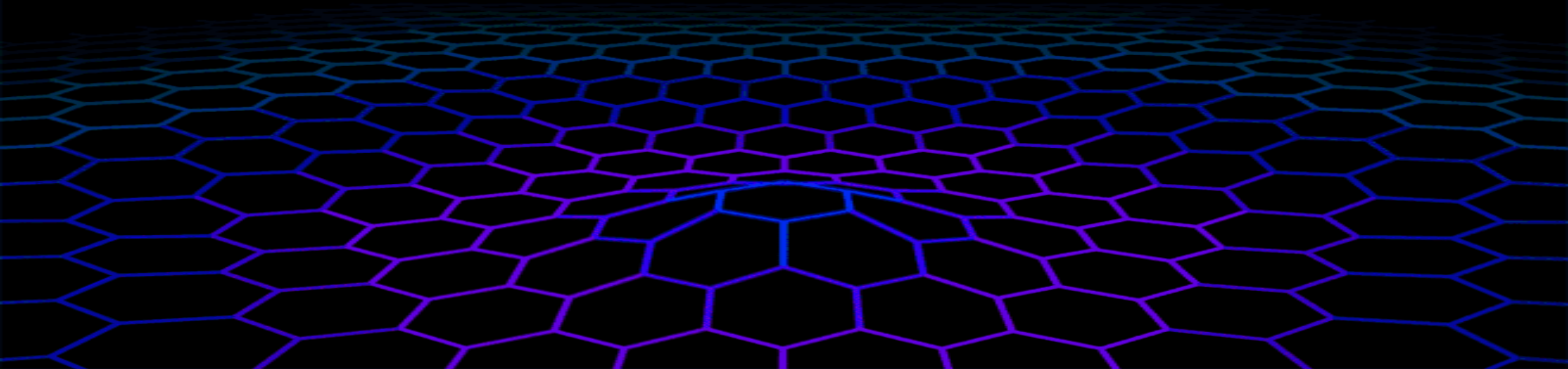


Sistemas Operacionais



Sistemas de Numeração

UNIDADES DE MEDIDA

Unidades de medida				
8 Bit	1	Byte	B	
1024 Bytes	1	kiloByte	KB	
1024 KiloBytes	1	MegaByte	MB	
1024 MegaBytes	1	GigaByte	GB	
1024 GigaBytes	1	TeraByte	TB	
1024 TeraBytes	1	PetaByte	PB	
1024 PetaBytes	1	ExaByte	EB	
1024 ExaBytes	1	ZettaByte	ZB	
1024 ZettaBytes	1	YottaByte	YB	

Obs.: Prefixos Kilo, Mega, Giga, Tera, Peta, Exa, Zeta e Yotta são utilizados para representar números de **base 10**

“Novidades”

Comercial x Computacional

Comercial

1KB comercial = 1000 Bytes

1MB comercial = 1000 KB

1GB comercial = 1000 MB

1GB comercial = 1000 x 1000 x 1000 Bytes = 1.000.000.000 Bytes

Computacional

1KB computacional = 1024 Bytes

1MB computacional = 1024 KB

1GB computacional = 1024 MB

1GB computacional = 1024 x 1024 x 1024 Bytes = 1.073.741.824 Bytes

$(1\text{GB computacional}) / (1\text{GB comercial}) = 1,073741824$

Kilo x Kibi



X



kilo, Mega, Giga, Tera, Peta, Exa, Zetta e Yotta

X

kibi, Mebi, Gibi, Tebi, Pebi, Exbi, Zebi e Yobi

kilo, Mega, Giga, Tera, Peta, Exa, Zetta e Yotta

kibi, Mebi, Gibi, Tebi, Pebi, Exbi, Zebi e Yobi

Os prefixos de unidade kilo, Mega, Giga, Tera, Peta, Exa, Zeta e Yotta são utilizados para representar números de base 10.

Em 1998 o **IEC** (International Electrotechnical Commission) introduziu os prefixos "kibi", "Mebi", "Gibi", "Tebi", "Pebi", "Exbi", "Zebi" e "Yobi" para que os prefixos kilo, Mega, Giga, Tera, Peta, Exa e Yotta parassem de ser utilizados na unidade de medida Byte e assim não gerar confusões.

Kibi, Mebi, Gibi, Tebi, Pebi, Exbi, Zebi e Yobi

Unidades de medida				
8	Bit	1	Byte	B
1024	Bytes	1	kibiByte	KiB
1024	KiloBytes	1	MebiByte	MiB
1024	MegaBytes	1	GibiByte	GiB
1024	GigaBytes	1	TebiByte	TiB
1024	TeraBytes	1	PebiByte	PiB
1024	PetaBytes	1	ExbiByte	EiB
1024	ExaBytes	1	ZebiByte	ZiB
1024	ZettaBytes	1	YobiByte	YiB

A letra "i" na abreviação e a sílaba "bi" do prefixo indicam que o padrão binário está sendo utilizado.

Kilo x Kibi

Base 10

8 Bit	1	Byte	B
1024 Bytes	1	kiloByte	KB
1024 KiloBytes	1	MegaByte	MB
1024 MegaBytes	1	GigaByte	GB
1024 GigaBytes	1	TeraByte	TB
1024 TeraBytes	1	PetaByte	PB
1024 PetaBytes	1	ExaByte	EB
1024 ExaBytes	1	ZettaByte	ZB
1024 ZettaBytes	1	YottaByte	YB

Base 2

8 Bit	1	Byte	B
1024 Bytes	1	kibiByte	KiB
1024 KiloBytes	1	MebiByte	MiB
1024 MegaBytes	1	GibiByte	GiB
1024 GigaBytes	1	TebiByte	TiB
1024 TeraBytes	1	PebiByte	PiB
1024 PetaBytes	1	ExbiByte	EiB
1024 ExaBytes	1	ZebiByte	ZiB
1024 ZettaBytes	1	YobiByte	YiB

Diferenças relativas entre múltiplos decimais e binários

Nome	Símbolo	Potência = valor (SI)	Nome	Símbolo	Potência binária	Diferença
quilo	k	$10^3 = 1000$	kibi	Ki	$2^{10} = 1024$	2,4%
mega	M	$10^6 = 1\,000\,000$	mebi	Mi	$2^{20} = 1\,048\,576$	4,9%
giga	G	$10^9 = 1\,000\,000\,000$	gibi	Gi	$2^{30} = 1\,073\,741\,824$	7,4%
tera	T	$10^{12} = 1\,000\,000\,000\,000$	tebi	Ti	$2^{40} = 1\,099\,511\,627\,776$	10,0%
peta	P	$10^{15} = 1\,000\,000\,000\,000\,000$	pebi	Pi	$2^{50} = 1\,125\,899\,906\,842\,624$	12,6%
exa	E	$10^{18} = 1\,000\,000\,000\,000\,000\,000$	exbi	Ei	$2^{60} = 1\,152\,921\,504\,606\,846\,976$	15,3%
zetta	Z	$10^{21} = 1\,000\,000\,000\,000\,000\,000\,000$	zebi	Zi	$2^{70} = 1\,180\,591\,620\,717\,411\,303\,424$	18,1%
yotta	Y	$10^{24} = 1\,000\,000\,000\,000\,000\,000\,000\,000$	yobi	Yi	$2^{80} = 1\,208\,925\,819\,614\,629\,174\,706\,176$	20,9%

IEC (International Electrotechnical Commission)

Comercial x Computacional

Exemplo

$$(1\text{GB computacional}) / (1\text{GB comercial}) = 1,073741824$$

Um HD de 500GB (comercial) possui 465,66GB

$$500\text{GB comercial} = 500 / 1,07374 = 465,66 \text{ GB computacional}$$

Resumo da encrenca

Os prefixos kilo, Mega, Giga, Tera, Peta, Exa, Zeta e Yotta são de base 10

1 kiloMetro = 1000 metros

1 kiloOhm = 1000 ohms

O sistema de medição de quantidade de dados tem base 2

1 kiloByte = 1024 Bytes.

O IEC criou os novos prefixos kibi, Mebi, Gibi, Tebi, Pebi, Exbi, Zebi e Yobi

Curiosidade

1Byte = 1 octeto

Metade de 1 Byte (4 bits) = nibble

bit => **B**inary Digit

16 bits = Word (palavra)

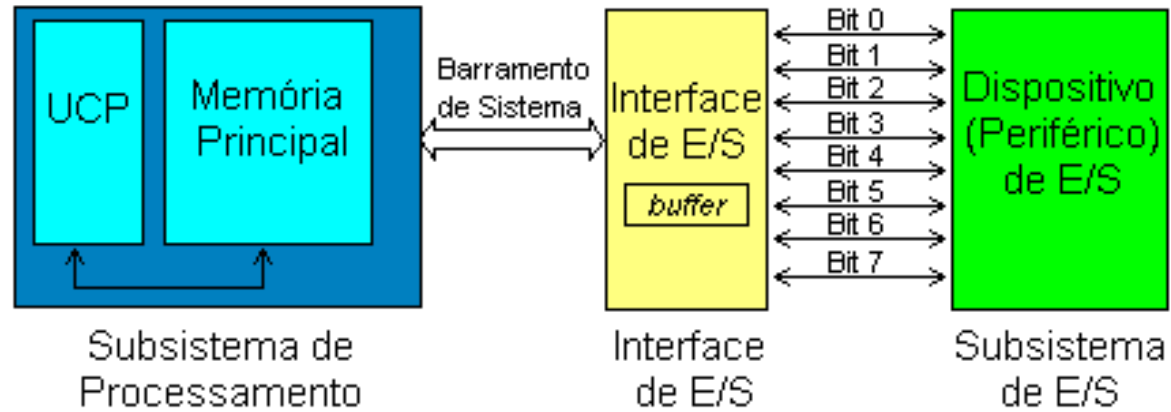
2 palavras = Dual Word

4 palavras = Quad Word

Nome	Nr de bits	Qtd de variações
Nible	4 bits	$2^4 = 16$
Byte (octeto)	8 bits	$2^8 = 256$
Word	16 bits	$2^{16} = 65.536$
Double Word	32 bits	$2^{32} = 4.294.967.296$
Quad Word	64 bits	$2^{64} = 18.446.744.073.709.551.616$

Comunicação

COMUNICAÇÃO EM PARALELO

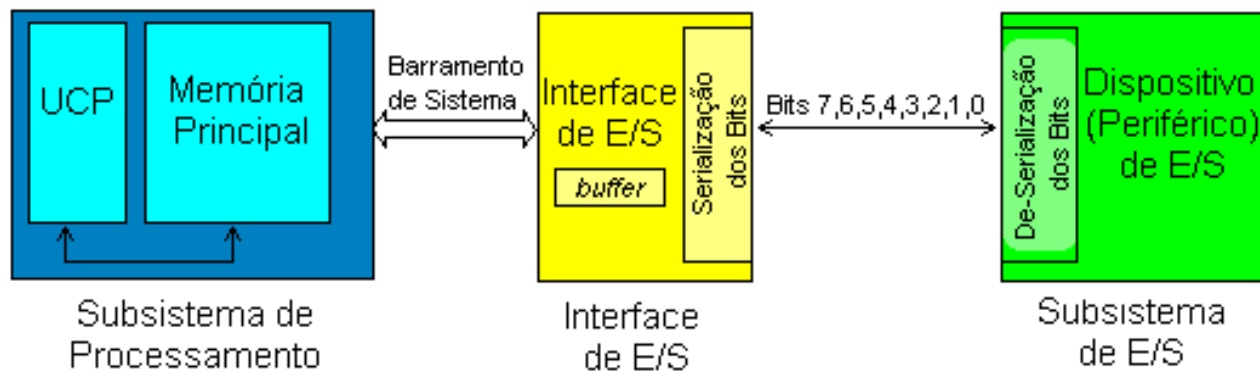


Vantagem:
Velocidade

Desvantagem:
Cabo longo / perder dados

COMUNICAÇÃO SERIAL

COMUNICAÇÃO SERIAL



Vantagem:

Menor possibilidade de perda de dados

Desvantagem:

Lentidão

COMUNICAÇÃO SERIAL

Aperfeiçoamentos (protocolo, *interface* e meio de transmissão)
Aumento da velocidade de transmissão por um único par de fios
Cabo coaxial ou de fibra ótica

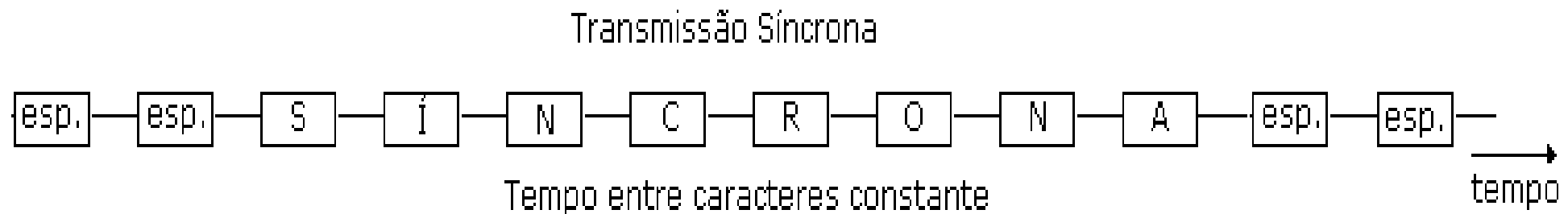
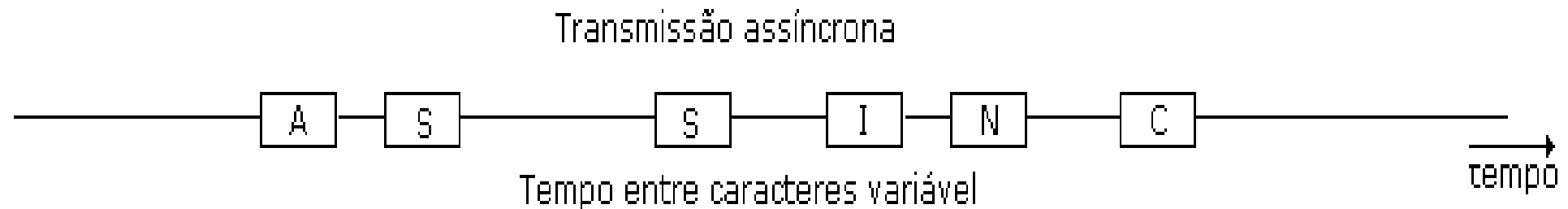
USB - Universal Serial Bus

Permite ligar até 128 dispositivos

Comunicação SERIAL x PARALELA

Característica	PARALELO	SERIAL
Custo	maior	menor
Distância	curta	sem limite
Throughput	alto	baixo

Transmissão Síncrona e Assíncrona



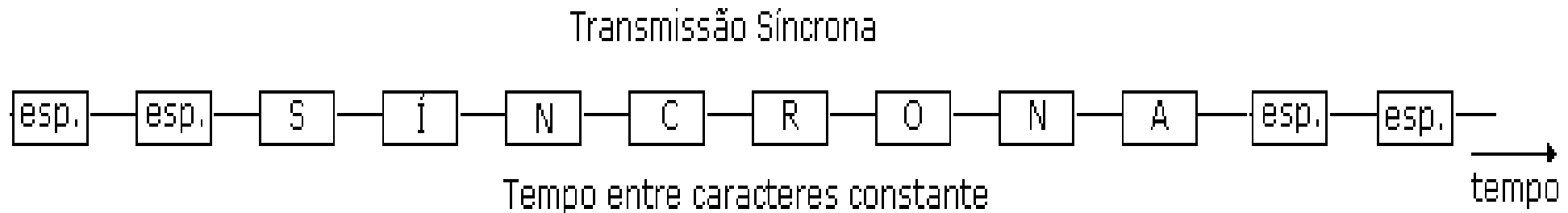
Transmissão Síncrona

Intervalo de tempo fixo.

Transmissor e receptor sincronizados

Relação direta entre tempo e os caracteres transferidos.

Transmissor continua enviando caracteres especiais



Transmissão Assíncrona

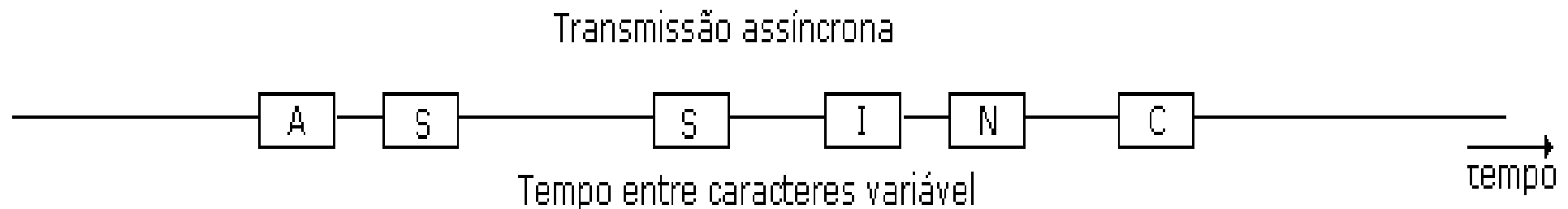
Intervalo de tempo não é fixo.

Na ausência de caracteres a serem transmitidos o transmissor mantém a linha sempre no estado 1

bit de partida (*start bit*) antes de cada caractere.

Linha livre (*idle*).

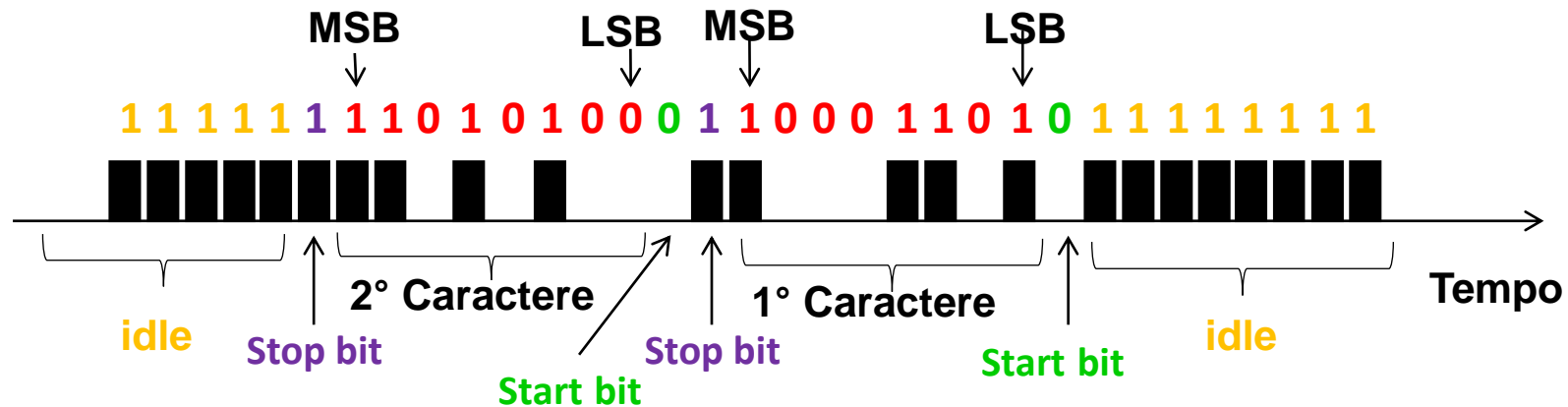
Ao final de cada caractere o transmissor insere bits de parada (*stop bits*).



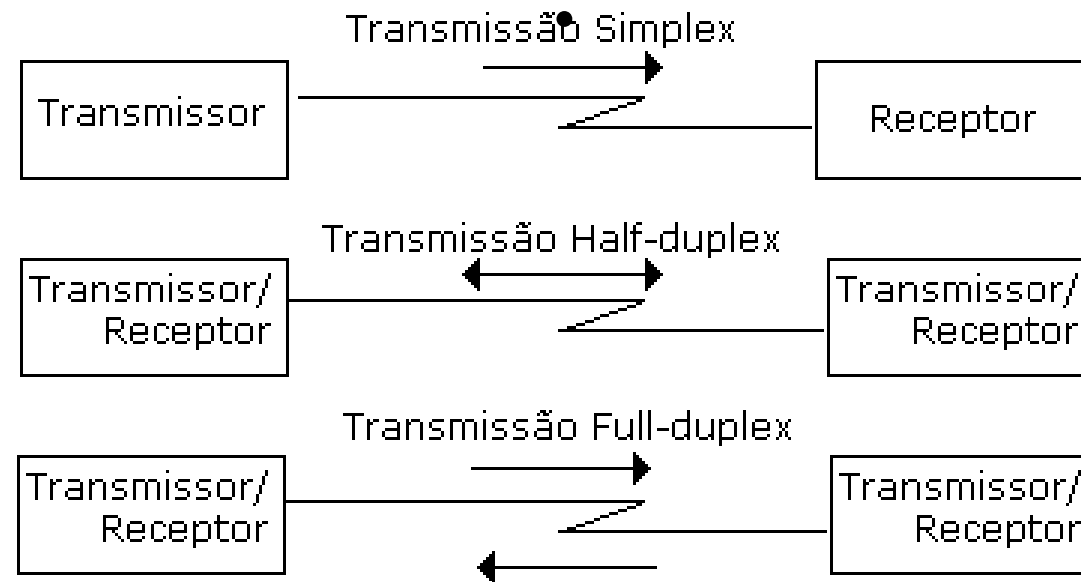
Transmissão Assíncrona

bits de informação são transmitidos em intervalos de tempo uniformes entre o *start bit* e o(s) *stop bit(s)*.

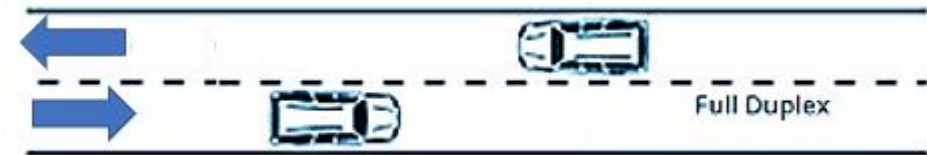
"start-stop".



TRANSMISSÃO SIMPLEX, HALF-DUPLEX E FULL-DUPLEX



TRANSMISSÃO SIMPLEX, HALF-DUPLEX E FULL-DUPLEX



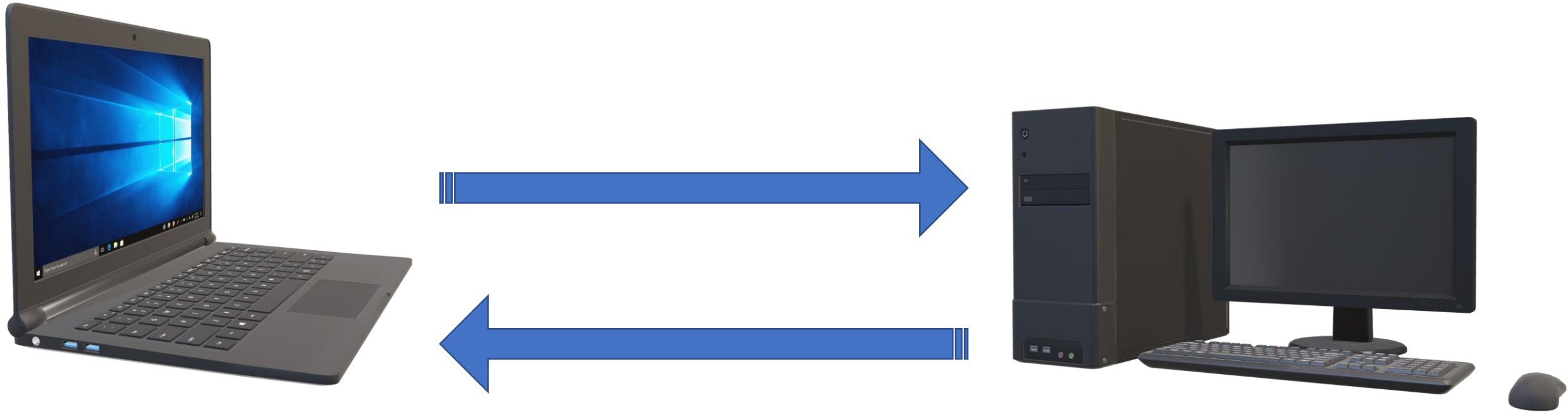
Comunicação SIMPLEX



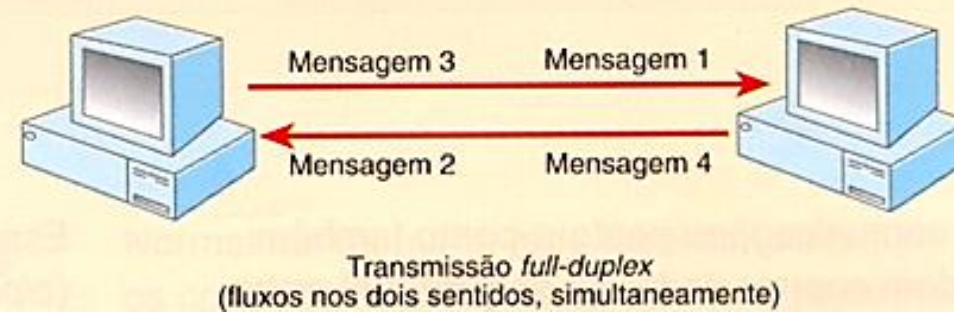
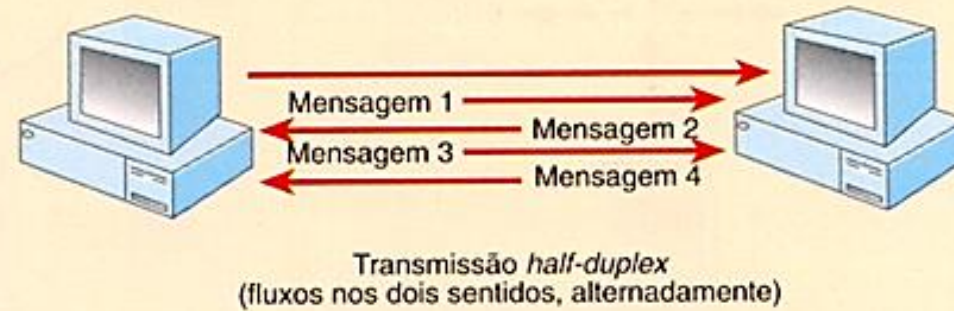
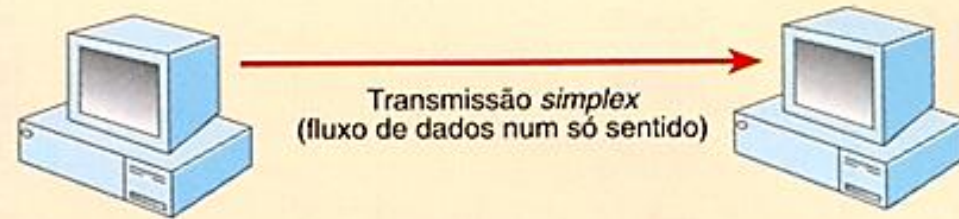
Transmissão HALF-DUPLEX



Transmissão FULL-DUPLEX



TRANSMISSÃO SIMPLEX, HALF-DUPLEX E FULL-DUPLEX



Portas Lógicas (Logic Gates)

Operações lógicas

As operações lógicas são estudadas pela álgebra de boole (George Boole)

A álgebra de Boole trabalha com apenas duas grandezas: **falso** ou **verdadeiro**.

As duas grandezas são representadas por **0** (falso) e **1** (verdadeiro).

Nos circuitos lógicos do computador, os sinais binários são representados por níveis de tensão.

Constantes e Variáveis Booleanas

A álgebra booleana permite apenas dois valores: 0 e 1.

Lógica 0 pode ser: *falso, desligado, baixo, não, interruptor aberto.*

Lógica 1 pode ser: *verdadeira, ligado, alto, sim, interruptor fechado.*

Três operações básicas:

- OR, AND e NOT.

Constantes e Variáveis Booleanas

0	1
Falso	Verdadeiro
Desligado	Ligado
Baixo	Alto
Não	Sim
Aberto	Fechado

Portas lógicas

As portas lógicas são os elementos básicos e elementares de um sistema de computação.

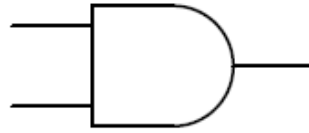
Elas são responsáveis por realizar as operações lógicas sobre os bits.

Os valores de entrada e saída são números binários.

Cada porta lógica realiza uma tarefa trivial.

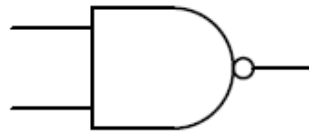
Portas lógicas

Portas ou circuitos **lógicos** são dispositivos que operam e trabalham com um ou mais sinais **lógicos** de entrada para produzir uma e somente uma saída, dependente da função implementada no circuito.



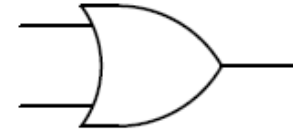
AND

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1



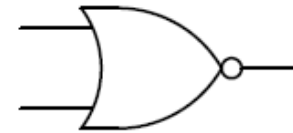
NAND

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0



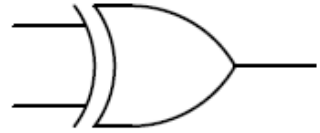
OR

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1



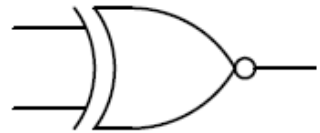
NOR

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0



XOR

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

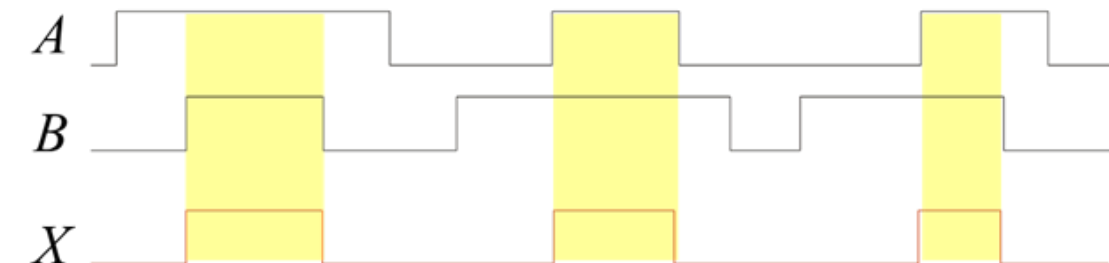
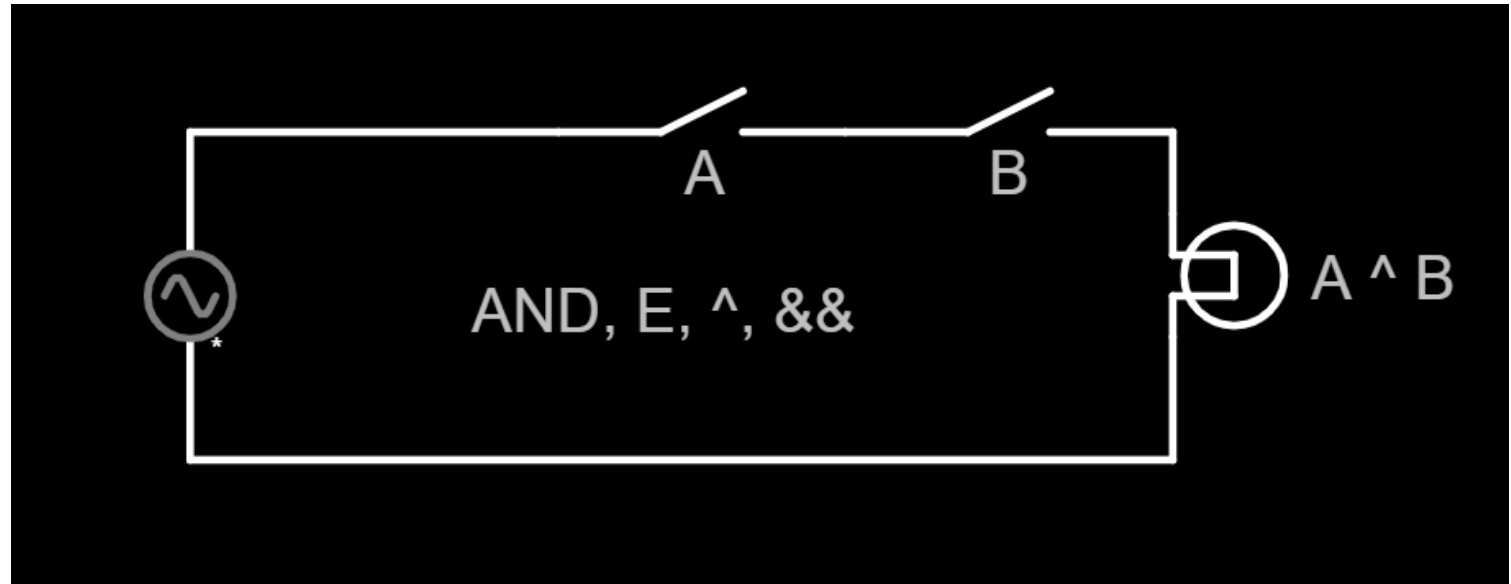
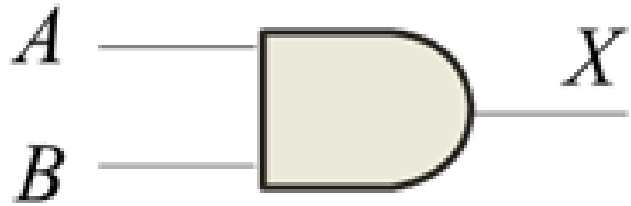


XNOR

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1

Porta AND (E) ^

A saída de uma porta AND é verdadeira se e somente se todas as entradas da porta forem verdadeiras.

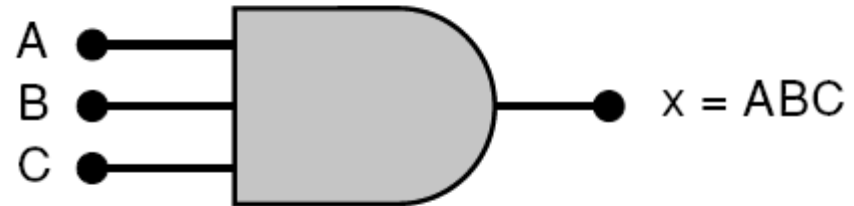


A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

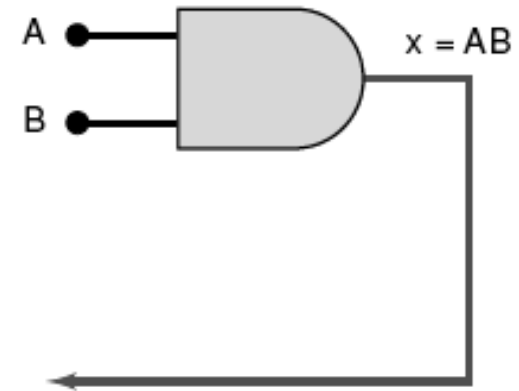
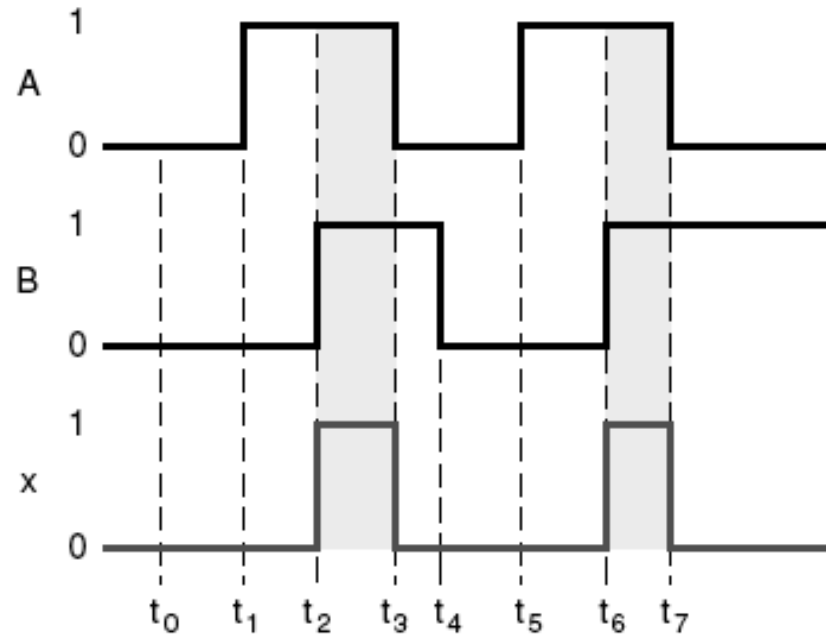
<https://tinyurl.com/289coyhb>

Porta AND (E)

A	B	C	$x = ABC$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

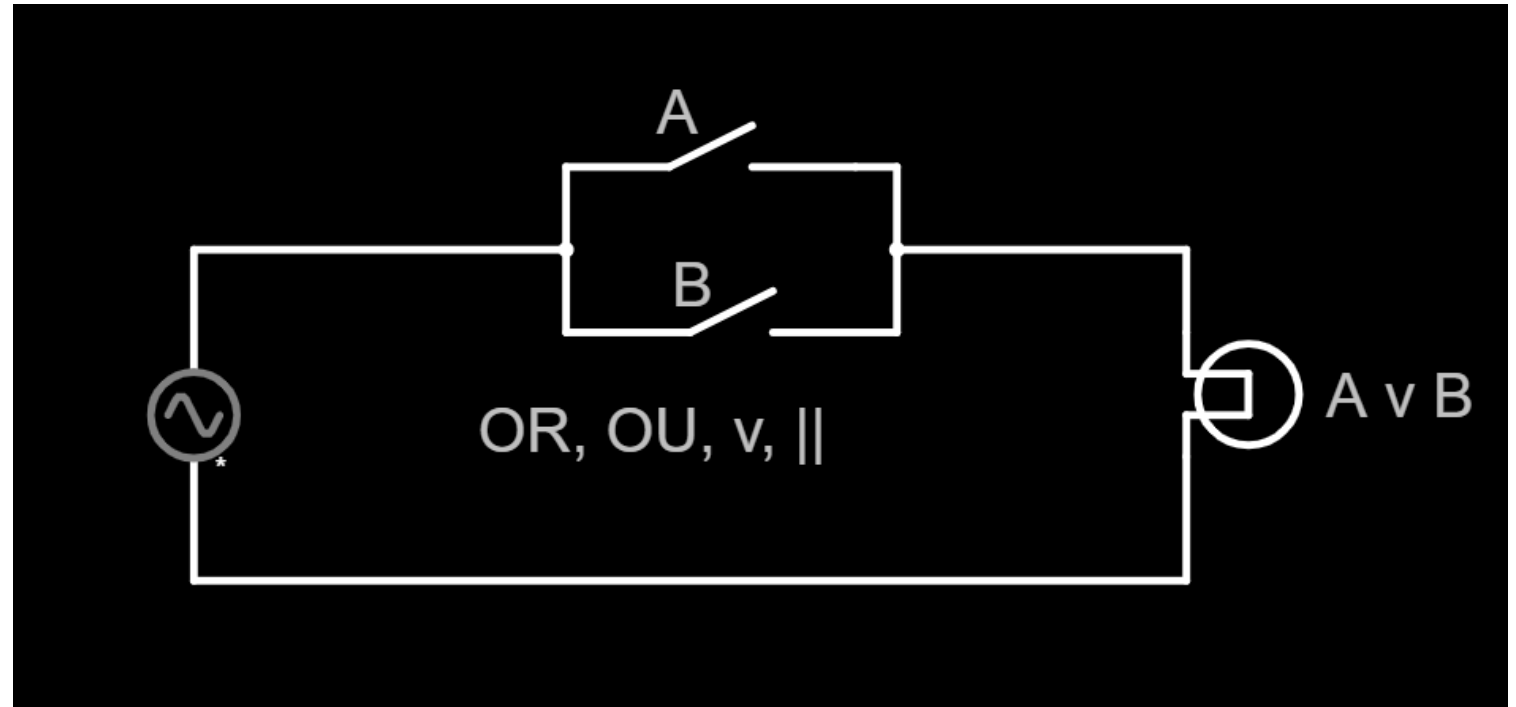
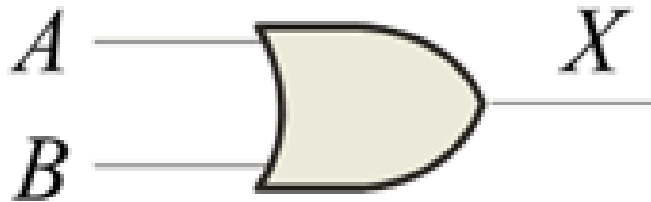


Porta AND (E)



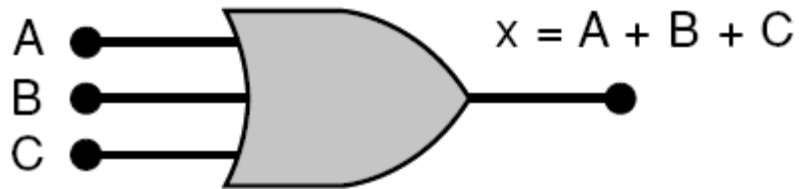
Portas OR (OU) v

A saída de uma porta OR é verdadeira se alguma ou todas as entradas da porta forem verdadeiras.



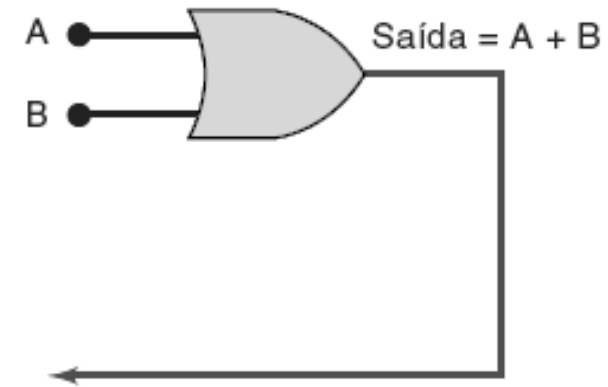
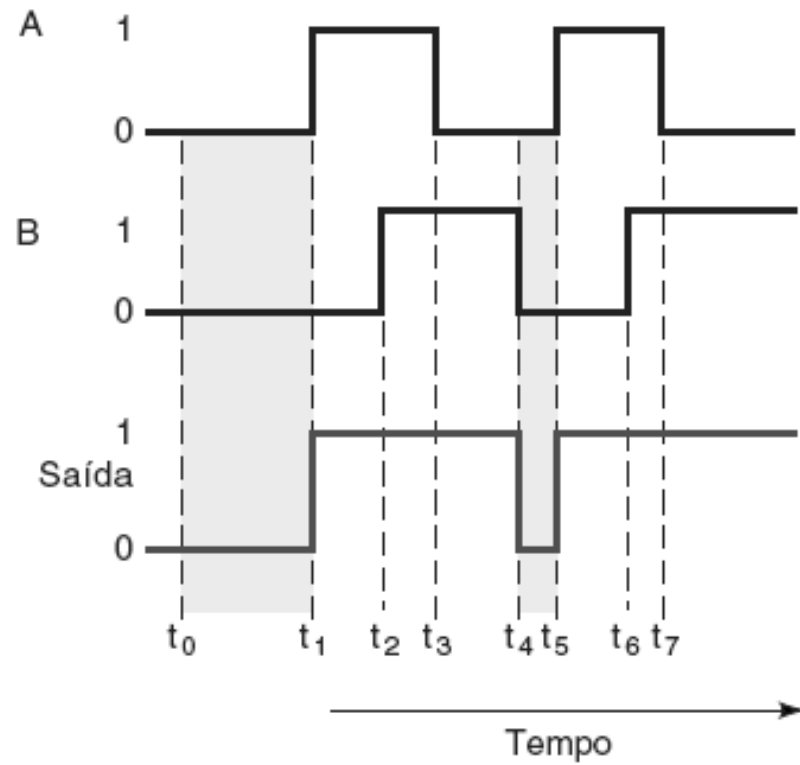
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Portas OR (OU)



A	B	C	$x = A + B + C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

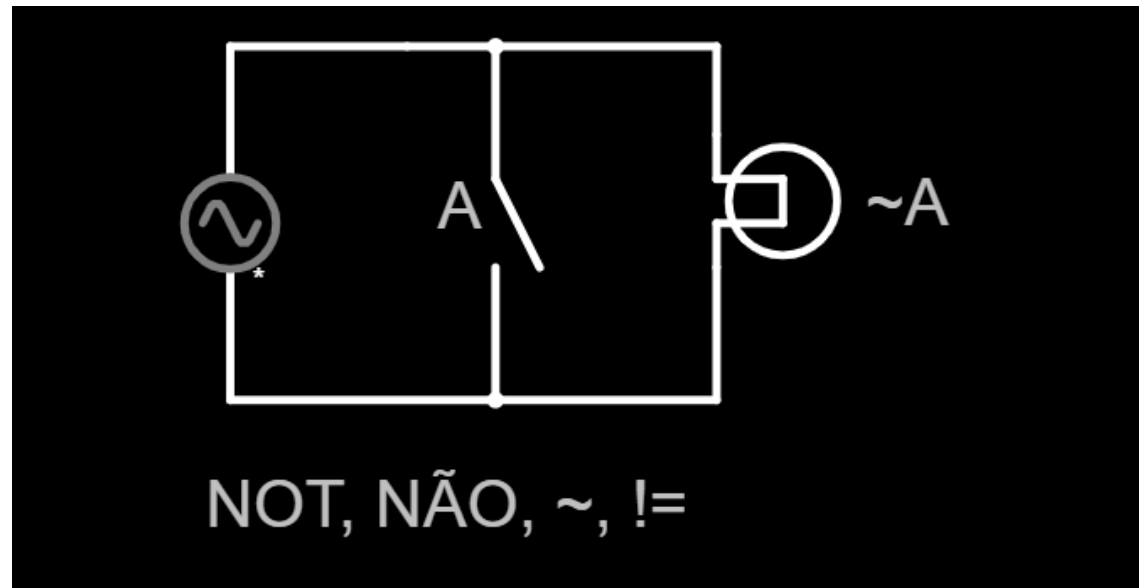
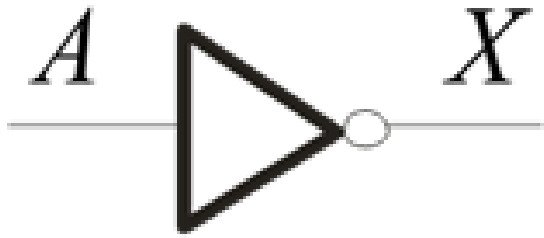
Portas OR (OU)



Porta NOT (NÃO) - Inversor

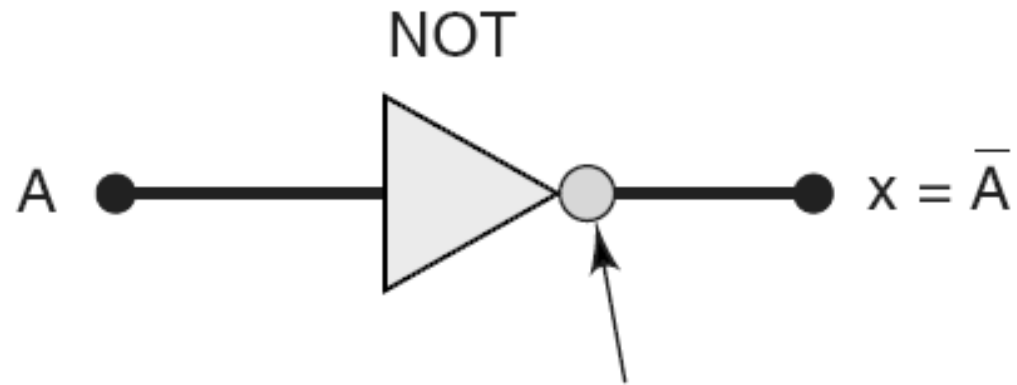
A saída de um inversor é o complemento (oposto) da entrada.

Quando a entrada para um inversor é alta (1), a saída é baixa (0); e quando a entrada é baixa, a saída é alta.



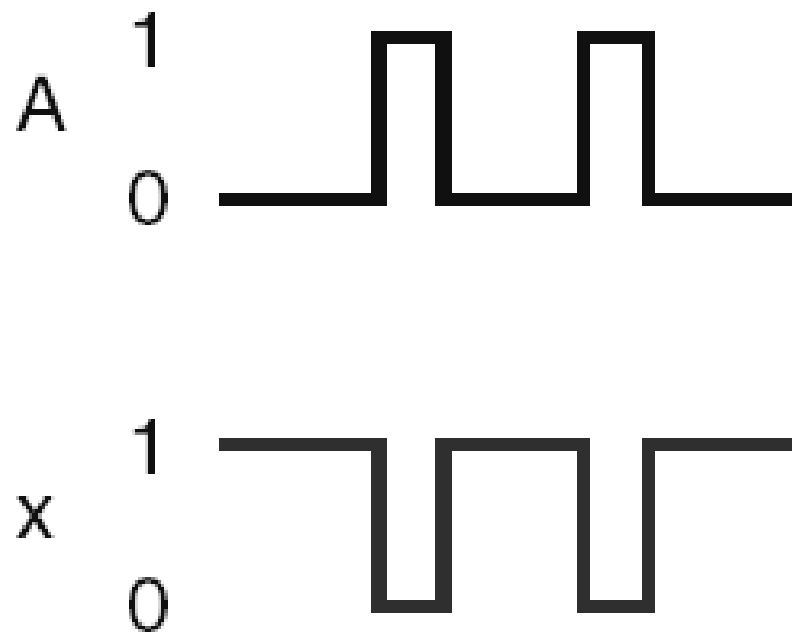
Input	Output
A	X
LOW (0)	HIGH (1)
HIGH (1)	LOW(0)

Porta NOT (NÃO) - Inversor



A presença de um
pequeno círculo
sempre denota inversão

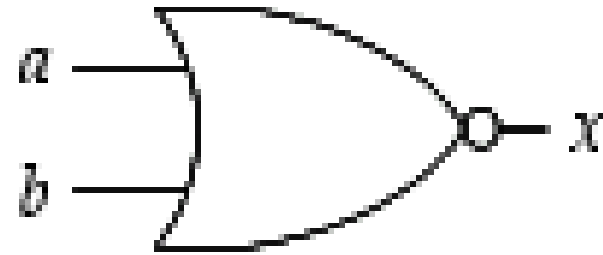
Porta NOT (NÃO) - Inversor



Input	Output
A	X
LOW (0)	HIGH (1)
HIGH (1)	LOW(0)

NOR

É uma porta OR e uma porta NOT combinadas.
O resultado é exatamente o inverso da porta OR.

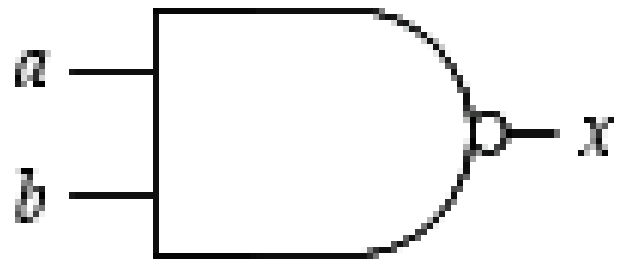


Expressão: $x = (a + b)'$

NOR		
a	b	x
0	0	1
1	0	0
0	1	0
1	1	0

NAND

É uma porta AND e uma porta NOT combinadas.
O resultado é exatamente o inverso da porta AND.

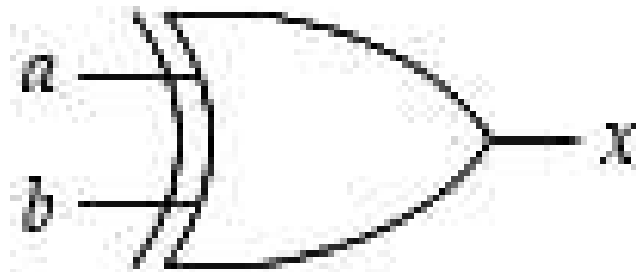


Expressão: $x = (a \times b)'$

NAND		
<i>a</i>	<i>b</i>	<i>x</i>
0	0	1
1	0	1
0	1	1
1	1	0

XOR – Exclusive OR

Retorna 1 somente se uma das entradas é 1.



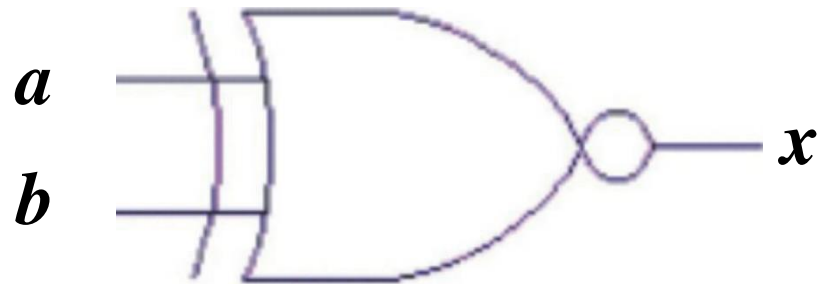
Expressão: $x = a \oplus b$

XOR		
a	b	x
0	0	0
1	0	1
0	1	1
1	1	0

XNOR

É uma porta XOR e uma porta NOT combinadas.

O resultado é exatamente o inverso da porta XOR.



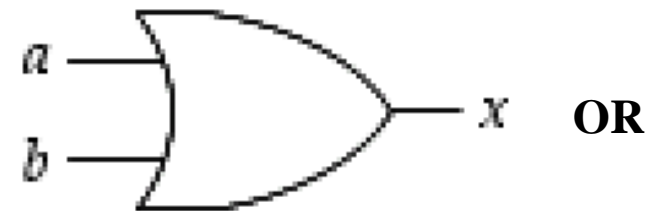
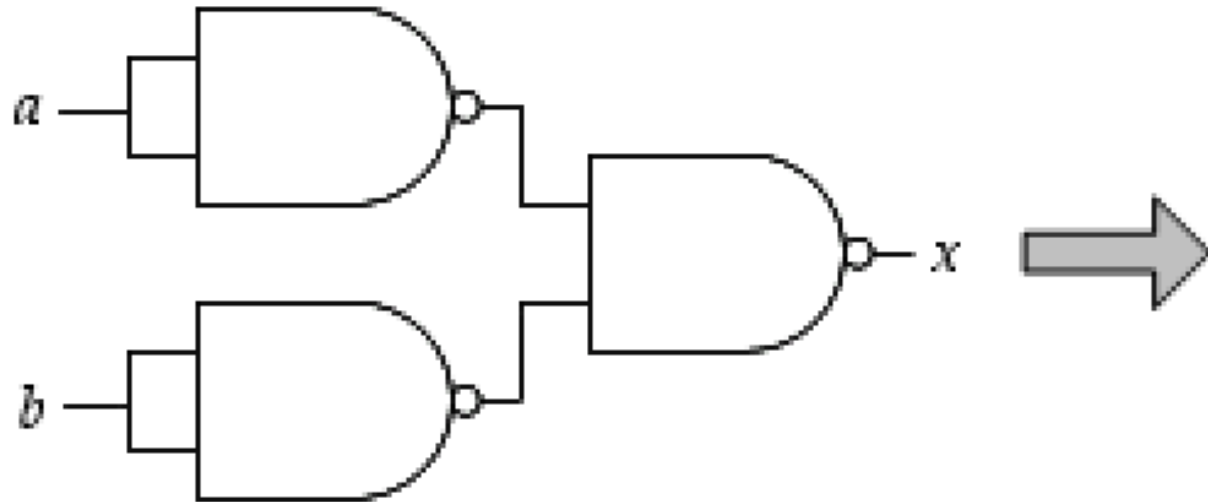
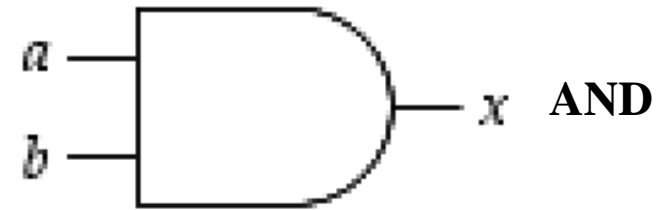
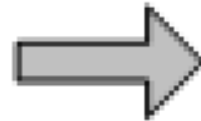
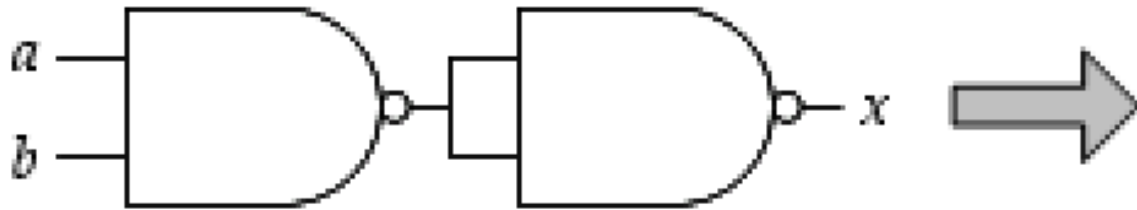
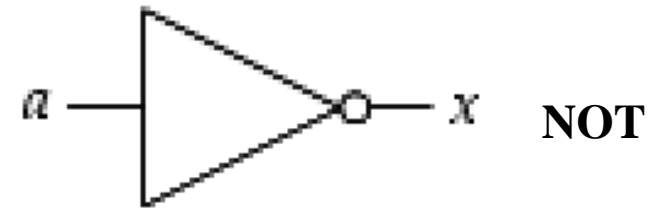
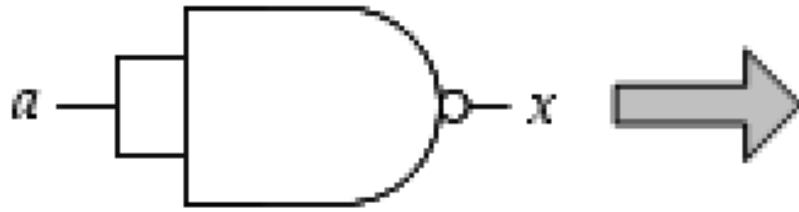
XNOR		
a	b	x
0	0	1
1	0	0
0	1	0
1	1	1

Expressão: $x = a \otimes b$

NAND – Porta universal

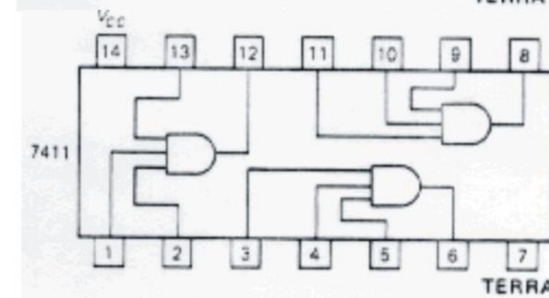
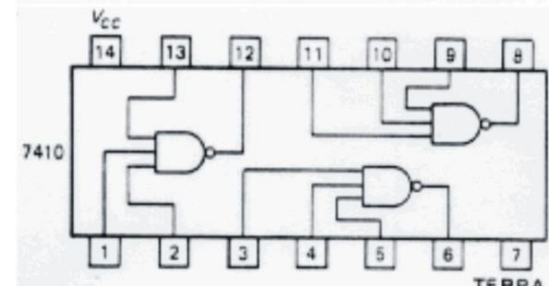
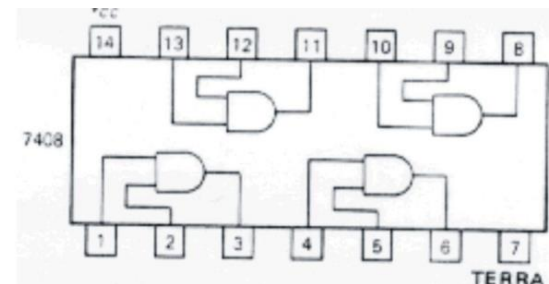
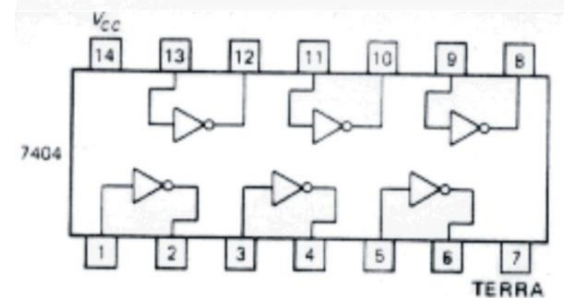
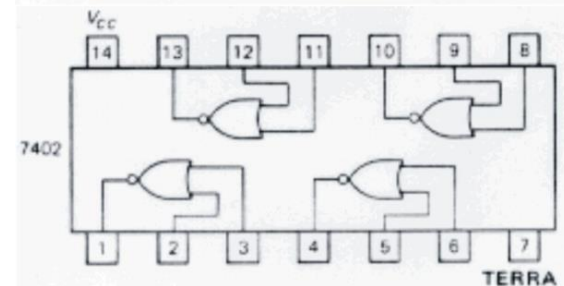
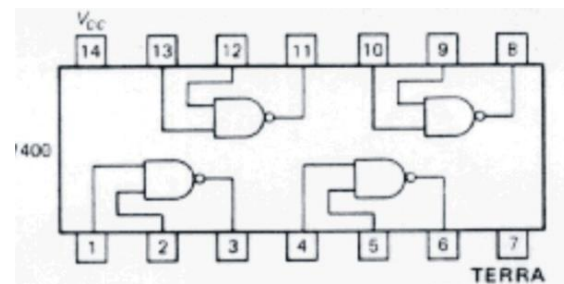
- Combinações de portas NAND podem ser usadas para simular todas as outras.
- Por este motivo, a porta NAND é considerada uma **porta universal**.
- Isso significa que qualquer circuito pode ser expresso pela combinação de portas NAND.

NAND – Porta universal



Circuitos

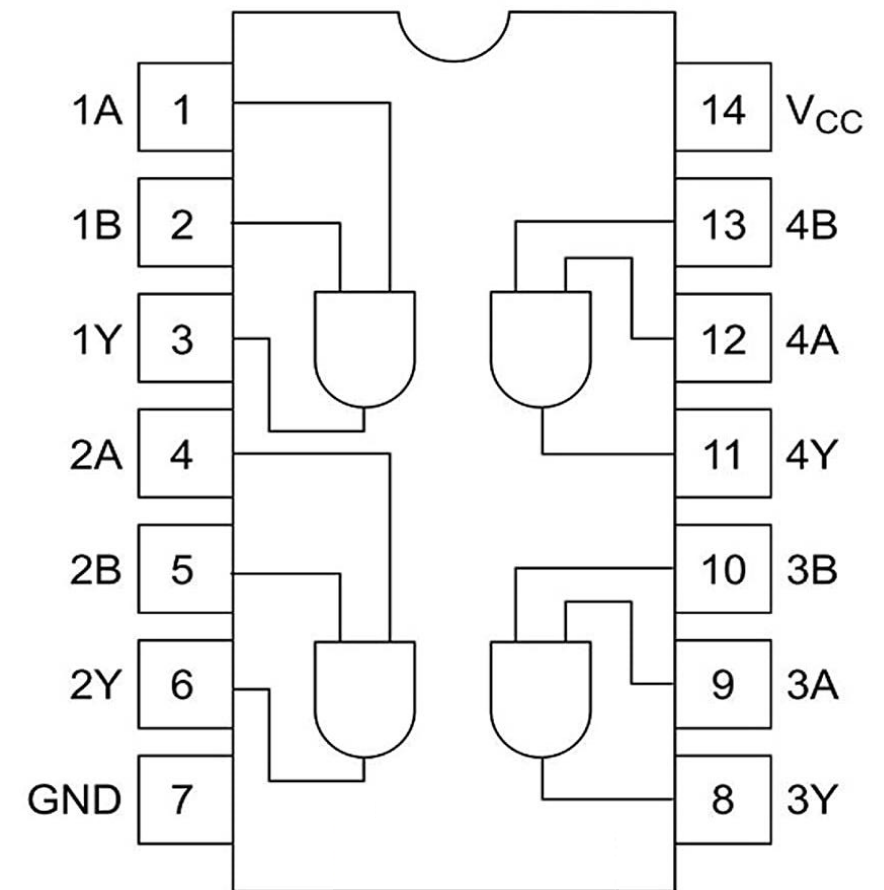
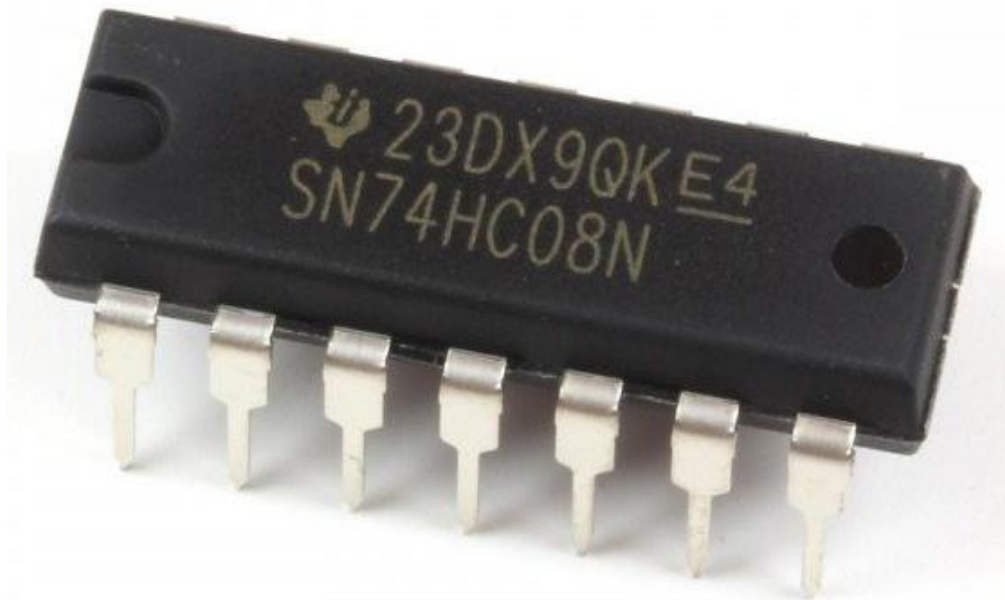
As portas lógicas são encontradas no mercado encapsuladas em chips de silício.



CI - 7408/74LS08/74HC08/74HCT08 (Porta AND)

O Circuito integrado **74HC08** fornece 4 portas **AND** independentes de 2 entradas com 1 saída cada. Faixa de operação: 2.0V ~ 6.0V.

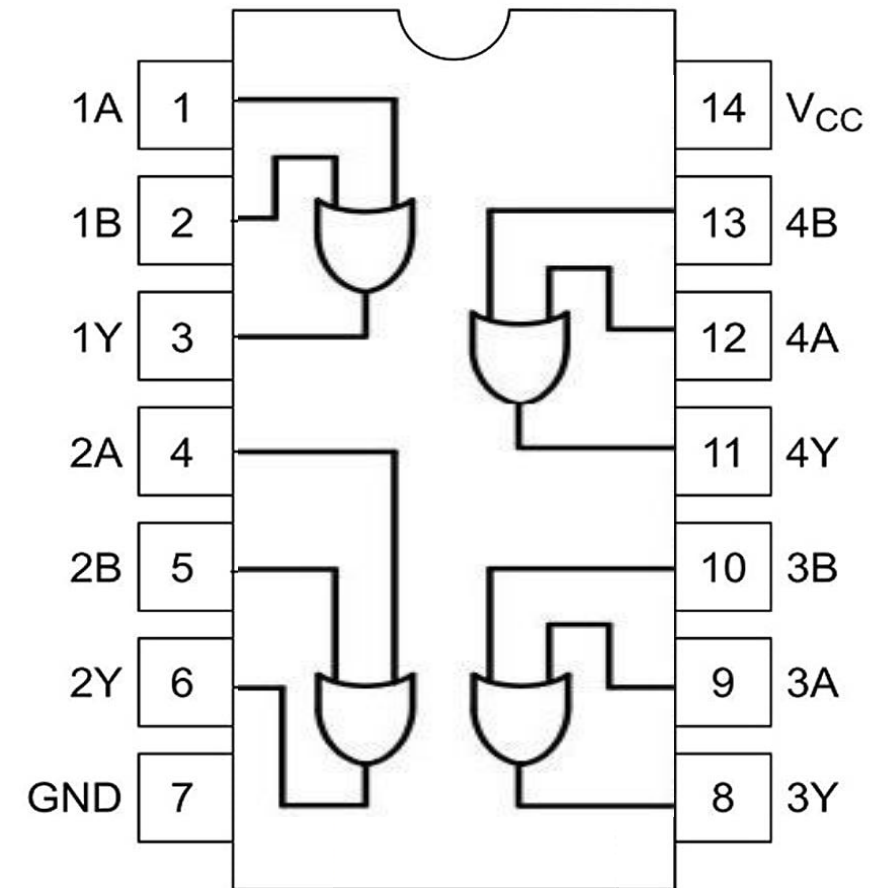
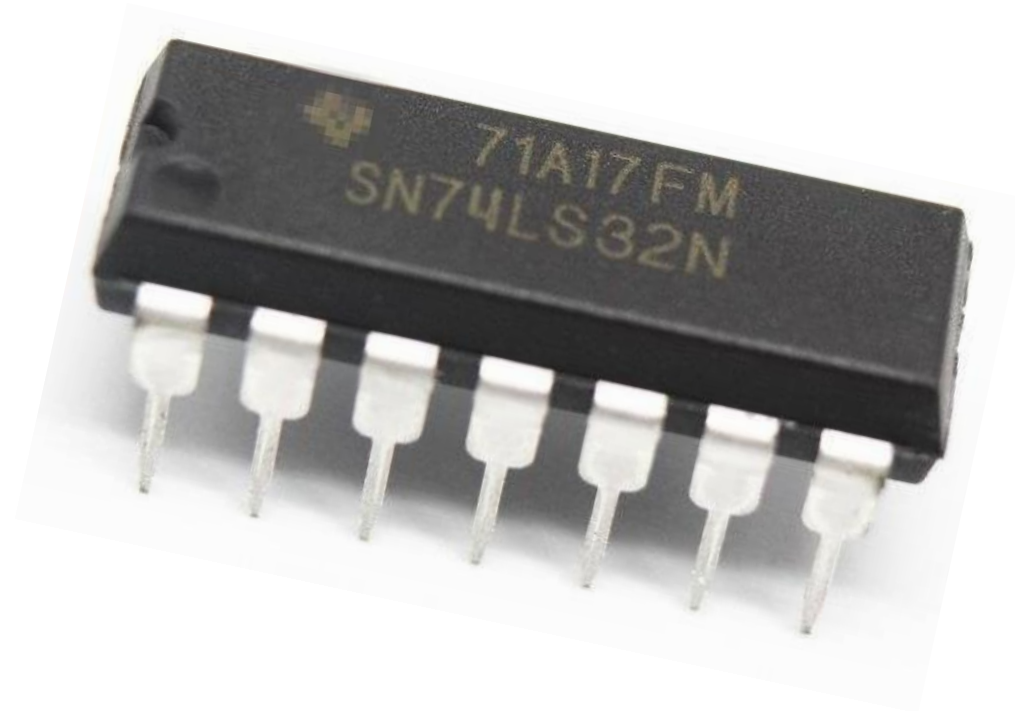
As portas executam a função booleana: $Y=A \cdot B$



CI - 7432/74LS32/74HC32/74HCT32 (Porta OR)

O Circuito integrado **74HC32** fornece 4 portas **OR** independentes de 2 entradas com 1 saída cada. Faixa de operação: 2.0V ~ 6.0V.

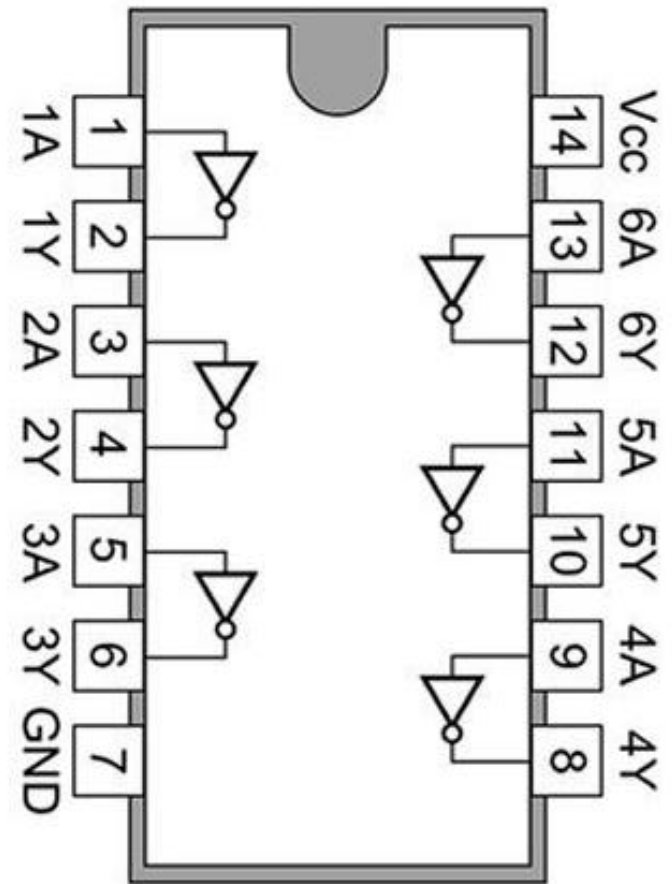
As portas executam a função booleana: $Y=A+B$



CI - 7404/74LS04/74HC04/74HCT04 (Porta NOT)

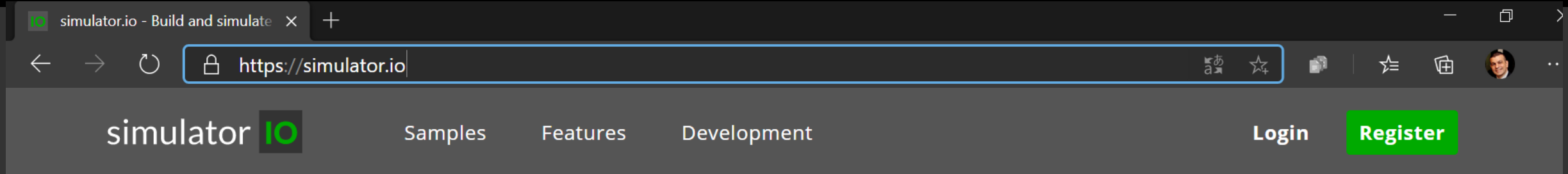
O Circuito integrado **74HC04** fornece 6 portas **NOT** independentes de 1 entrada com 1 saída cada. Faixa de operação: 2.0V ~ 6.0V.

As portas executam a função booleana: $Y = \overline{A}$

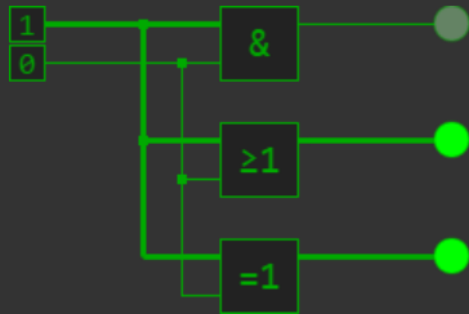


Sites para praticar

Simulator IO - <https://simulator.io>



Build and simulate logic circuits.



Create new logic circuit

Web-based logic circuit simulator for people who want to build a computer from scratch.

The image is a screenshot of a web browser displaying the Simulator IO website. The browser's address bar shows the URL "https://simulator.io/board". The website's header includes the text "simulator IO Anonymous board" and a notification "Unsaved changes. Click Link or Fork to save." with buttons for "Login" and "Register". A toolbar with various icons for running, undo, redo, and editing is visible. On the left, a sidebar titled "Elements" lists categories: "GATES" (containing AND, OR, NOT, and XOR), "BASIC" (containing constants 0 and 1, a square wave, and a clock), and "ADDER" (containing Half Adder and Full Adder). The main workspace is a grid where a logic circuit is being built. The circuit consists of three components: an AND gate with two inputs connected to constant '0' blocks, an OR gate with two inputs connected to constant '0' blocks, and a NOT gate with one input connected to a constant '0' block. Each component has a green output line ending in a small circle. The bottom of the sidebar shows an "Options" section with the text "No options available for this pool/element".

← → ↺ 🔒 <https://simulator.io/board>



simulator **IO** **Anonymous board**

Unsaved changes. Click **Link** or **Fork** to save.

Login

Register

RUN Aa

[Link](#) [Fork](#)

Elements



Options

No options available for this pool/element

Logic.ly - <https://logic.ly/demo/>

Untitled Circuit* - Logic.ly Online

<https://logic.ly/demo>

logic.ly File Edit View Tools Simulate Help Buy Logic.ly — \$59

Output Controls

- Light Bulb
- 4-Bit Digit

Logic Gates

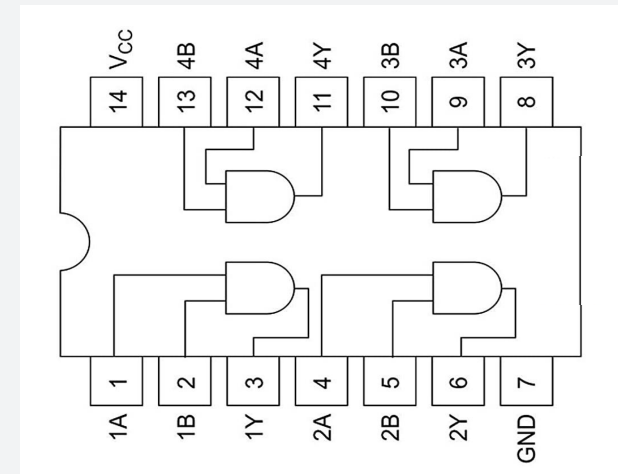
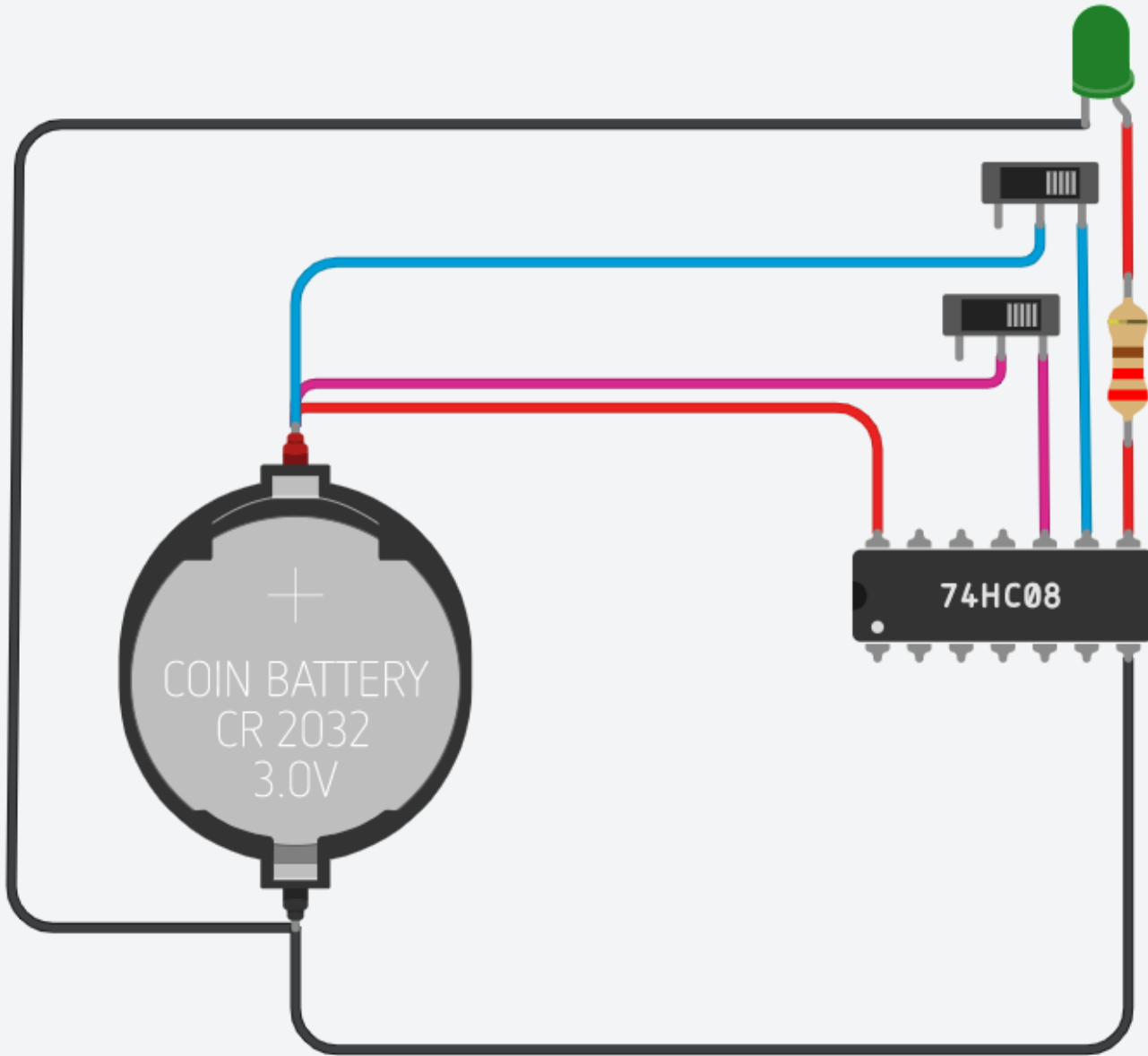
- Buffer
- NOT Gate
- AND Gate
- NAND Gate
- OR Gate
- NOR Gate
- XOR Gate
- XNOR Gate
- Tri-State

Flip-Flops

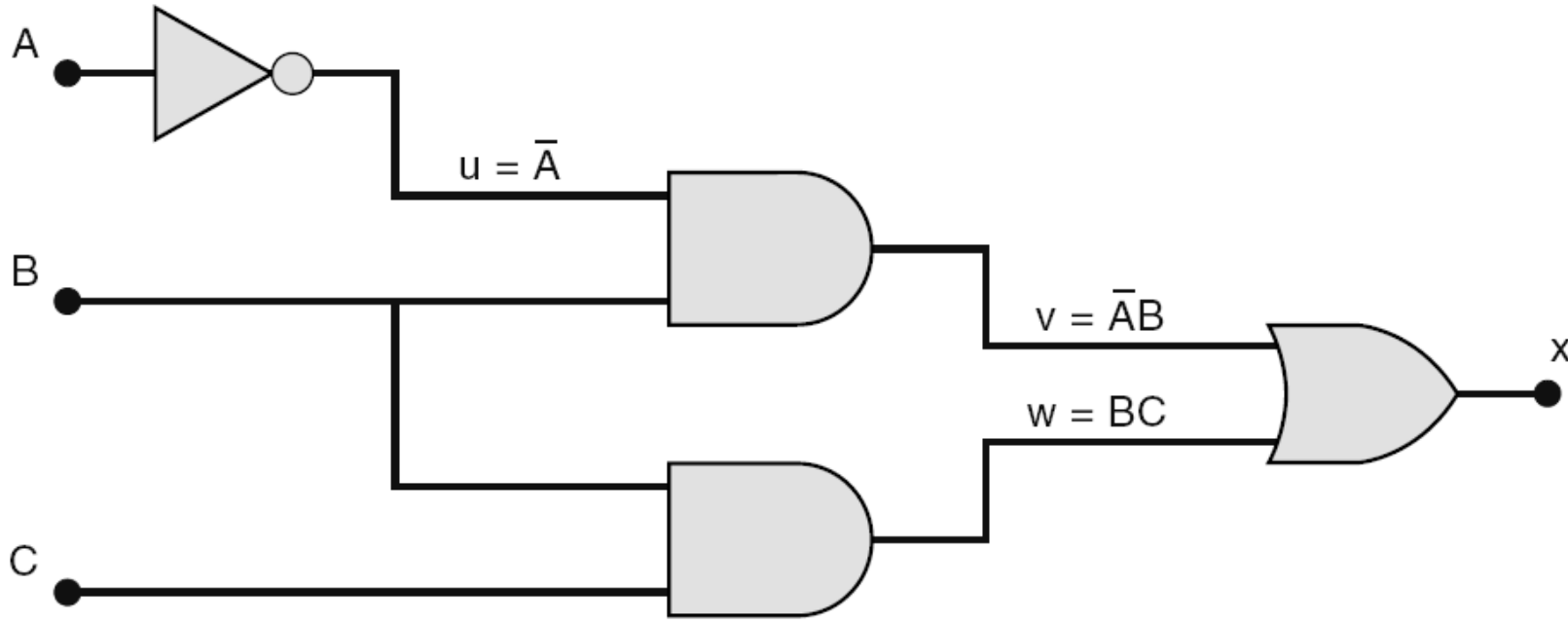
- D Flip-Flop
- J-K Flip-Flop

The image shows a screenshot of the Logic.ly online circuit simulator. The interface includes a browser window with the URL <https://logic.ly/demo/>, a menu bar with options like File, Edit, View, Tools, Simulate, and Help, and a toolbar with various icons. On the left, there are panels for Output Controls (Light Bulb, 4-Bit Digit), Logic Gates (Buffer, NOT Gate, AND Gate, NAND Gate, OR Gate, NOR Gate, XOR Gate, XNOR Gate, Tri-State), and Flip-Flops (D Flip-Flop, J-K Flip-Flop). The main workspace displays a logic circuit with two AND gates. The top AND gate has three inputs from switches, and the bottom AND gate has three inputs from switches. The top AND gate is currently active, indicated by a blue wire and a glowing light bulb.

TinkerCAD - <https://www.tinkercad.com/>

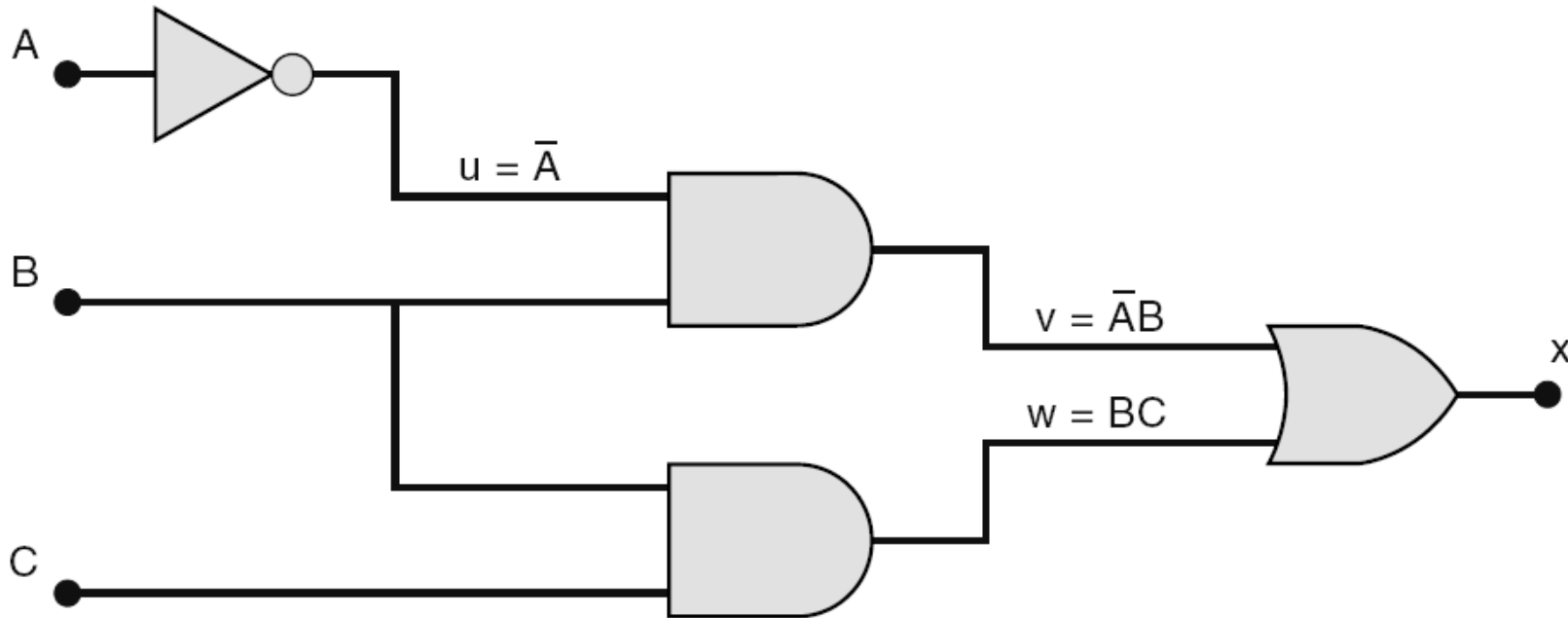


Resolva a tabela verdade



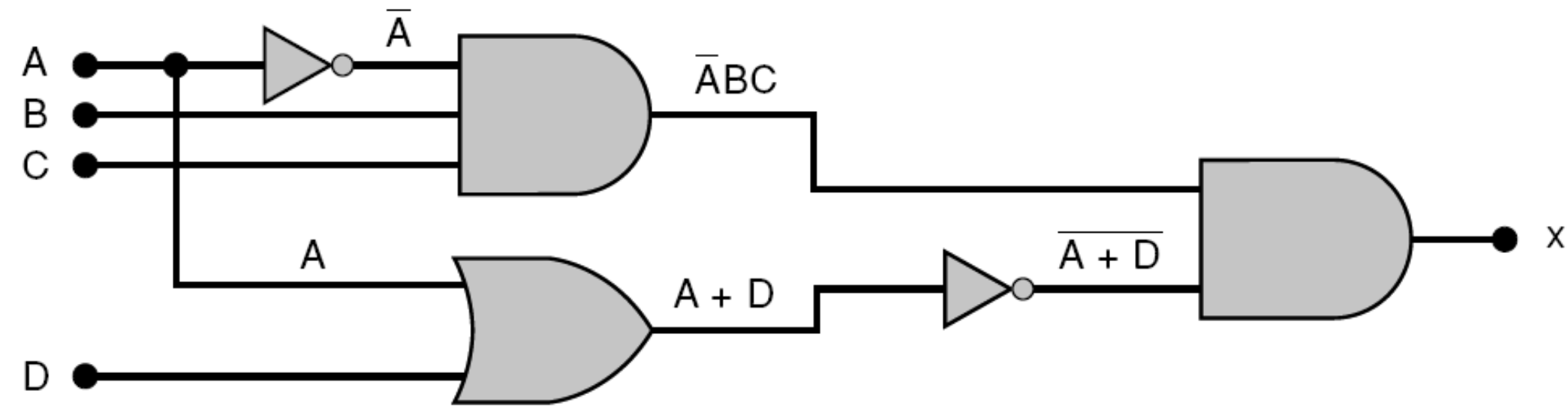
A	B	C	$\bar{A}B$	BC	X
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Resolva a tabela verdade



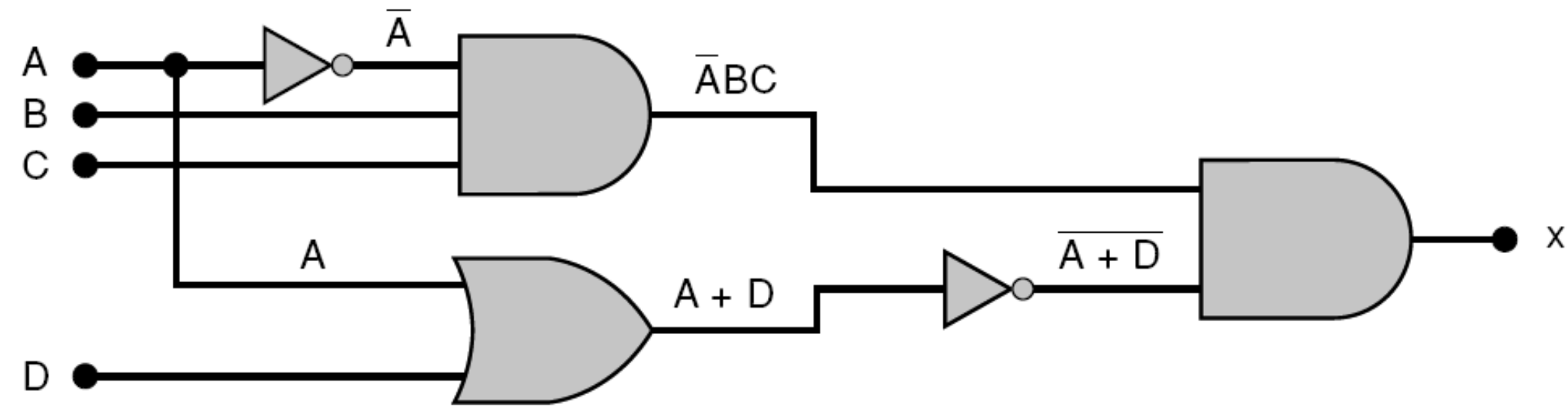
A	B	C	$\bar{A}B$	BC	X
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	1	1

Resolva a tabela verdade



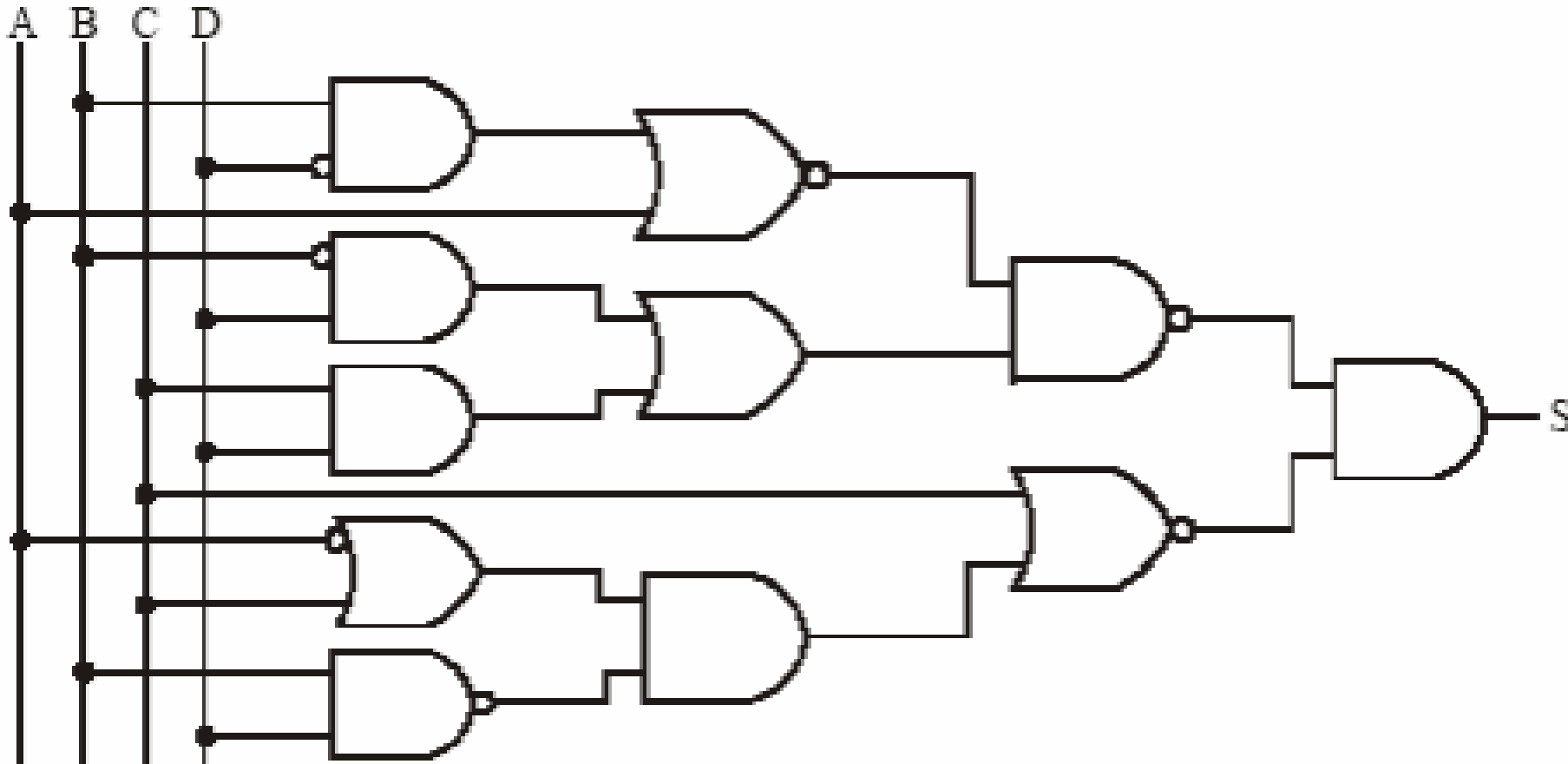
A	B	C	D	X
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Resolva a tabela verdade



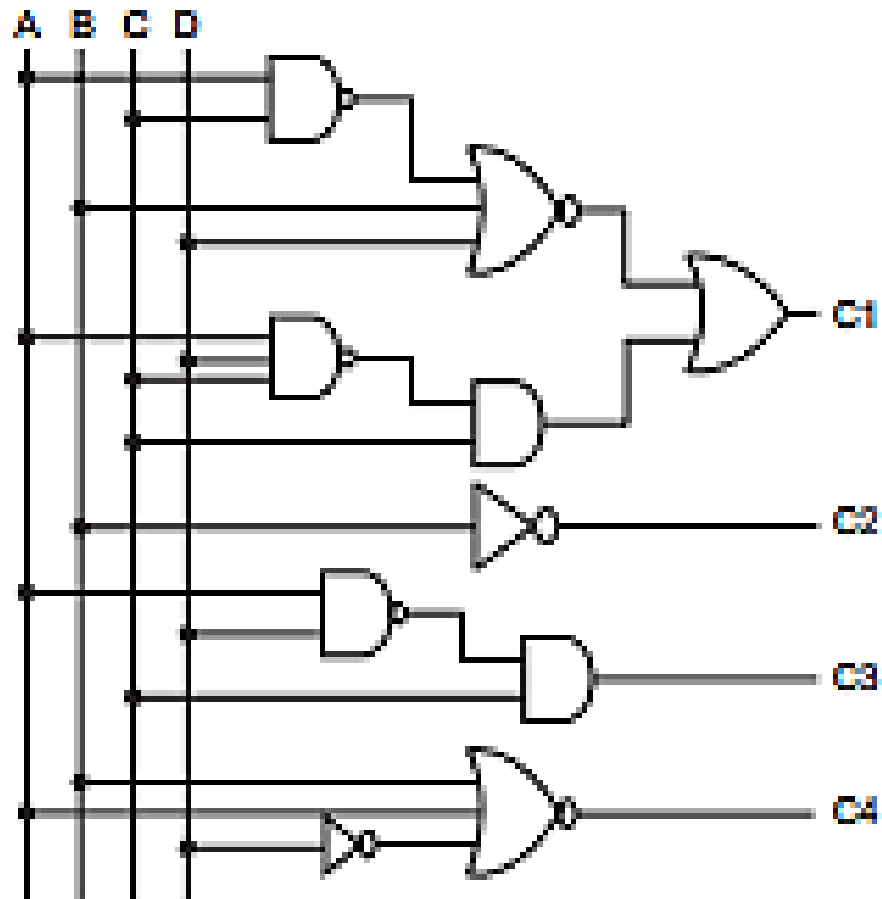
A	B	C	D	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Resolva a tabela verdade



A	B	C	D	S
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Crie as tabelas verdade



A	B	C	D	C1	C2	C3	C4
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

Não contavam
com minha
astúcia!



Atividade