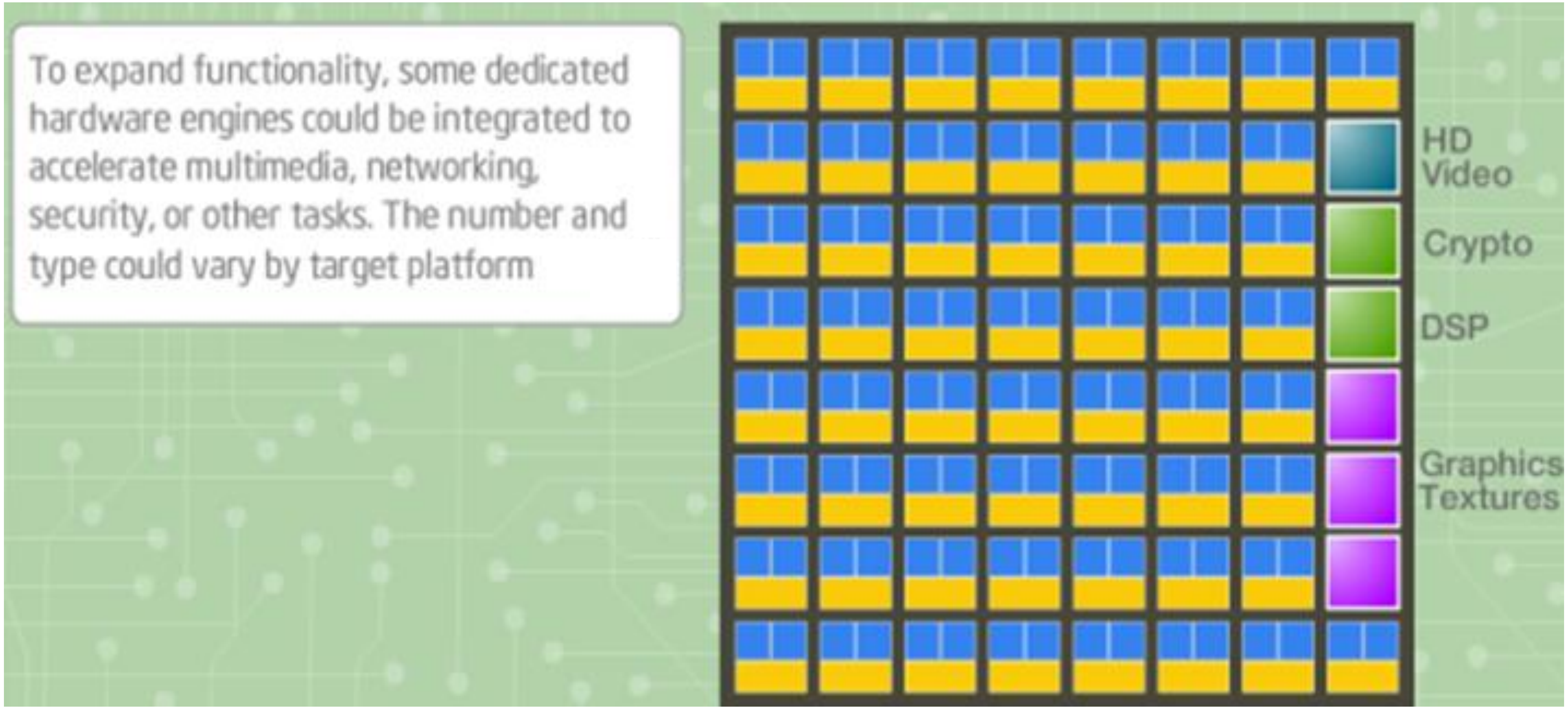
Arquitetura de Computadores I

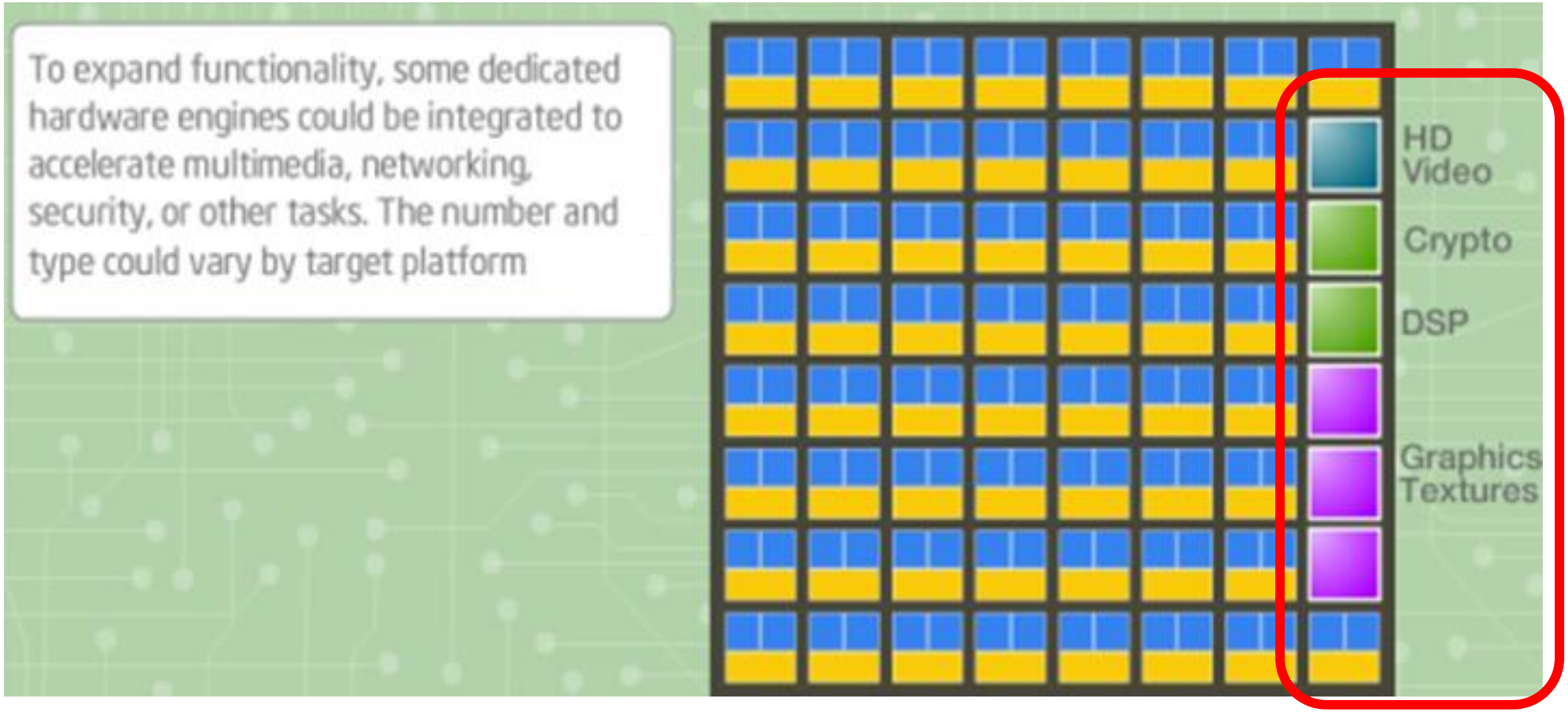
Aula 5

Onde usar um conjunto de instruções de propósito específico?



https://www.youtube.com/watch?v=TAKG0UvtzpE
https://www.youtube.com/watch?v=We_PRtRfiNs

Onde usar um conjunto de instruções de propósito específico?

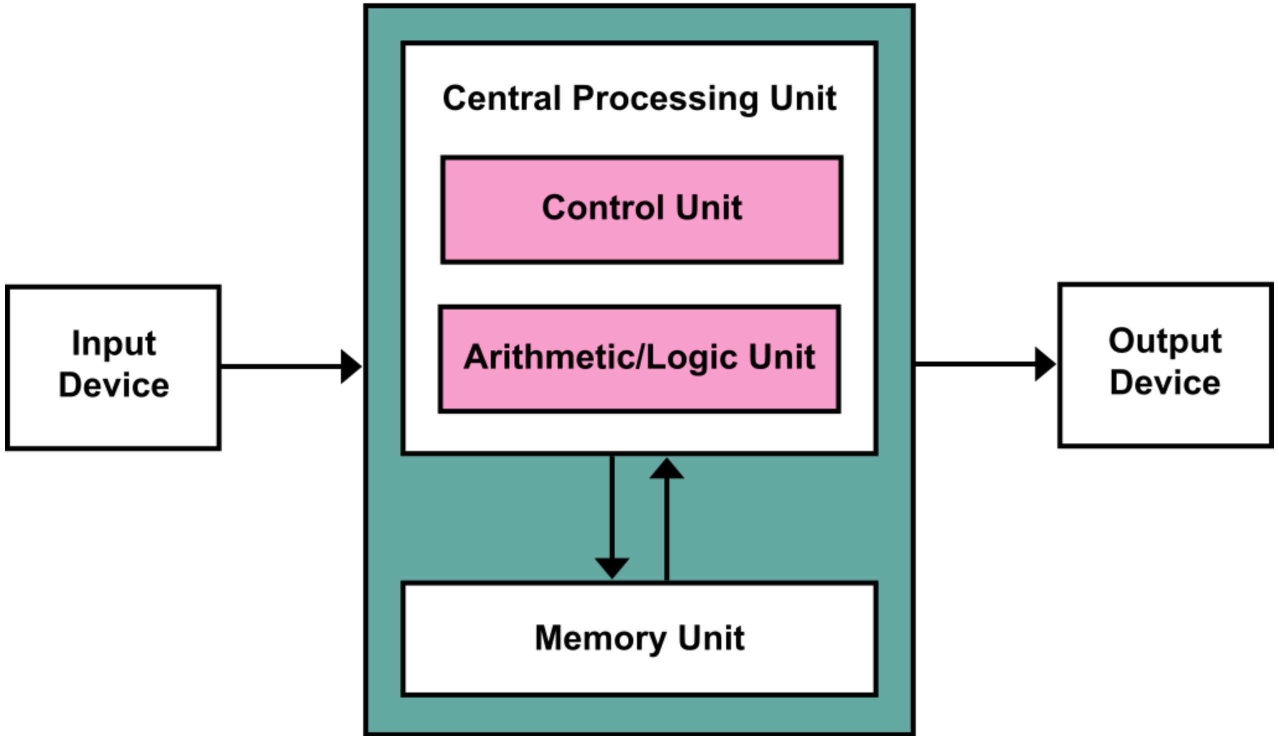


https://www.youtube.com/watch?v=TAKG0UvtzpE
https://www.youtube.com/watch?v=We_PRtRfiNs

Um fluxo de instruções em execução

Fluxo 1

	LOAD R1, 0(R2)
	BEQ R1, R2, Salto
	ADD R3, R1, R2
Salto:	STORE R1, 0(R2)
	⋮

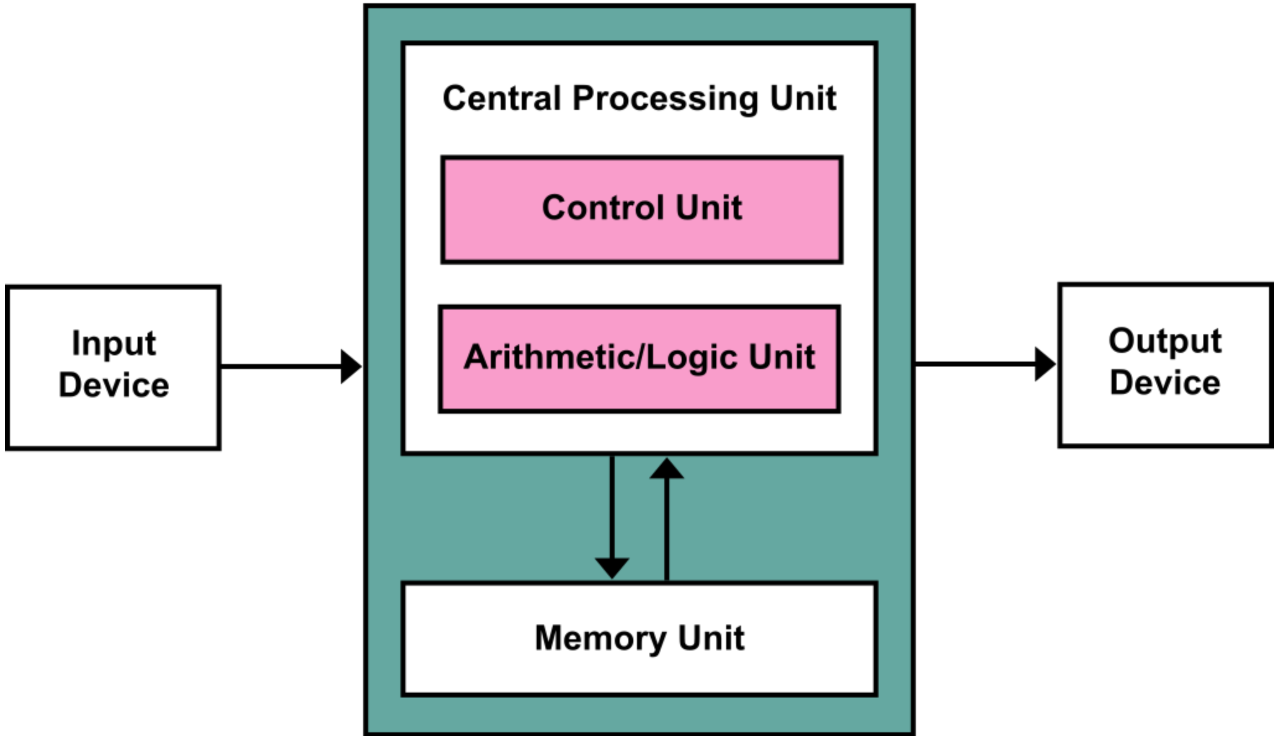


https://en.wikipedia.org/wiki/Von_Neumann_architecture

Fluxos de instruções em execução

Fluxo 1		Fluxo 2	
	LOAD R1, 0(R2)		LOAD R1, 0(R2)
	BEQ R1, R2, Salto		BEQ R1, R2, Salto
	ADD R3, R1, R2		SEND R1, 0(R2)
Salto:	STORE R1, 0(R2)	Salto:	RECV R1, 0(R2)
	⋮		⋮

Fluxos de instruções em execução

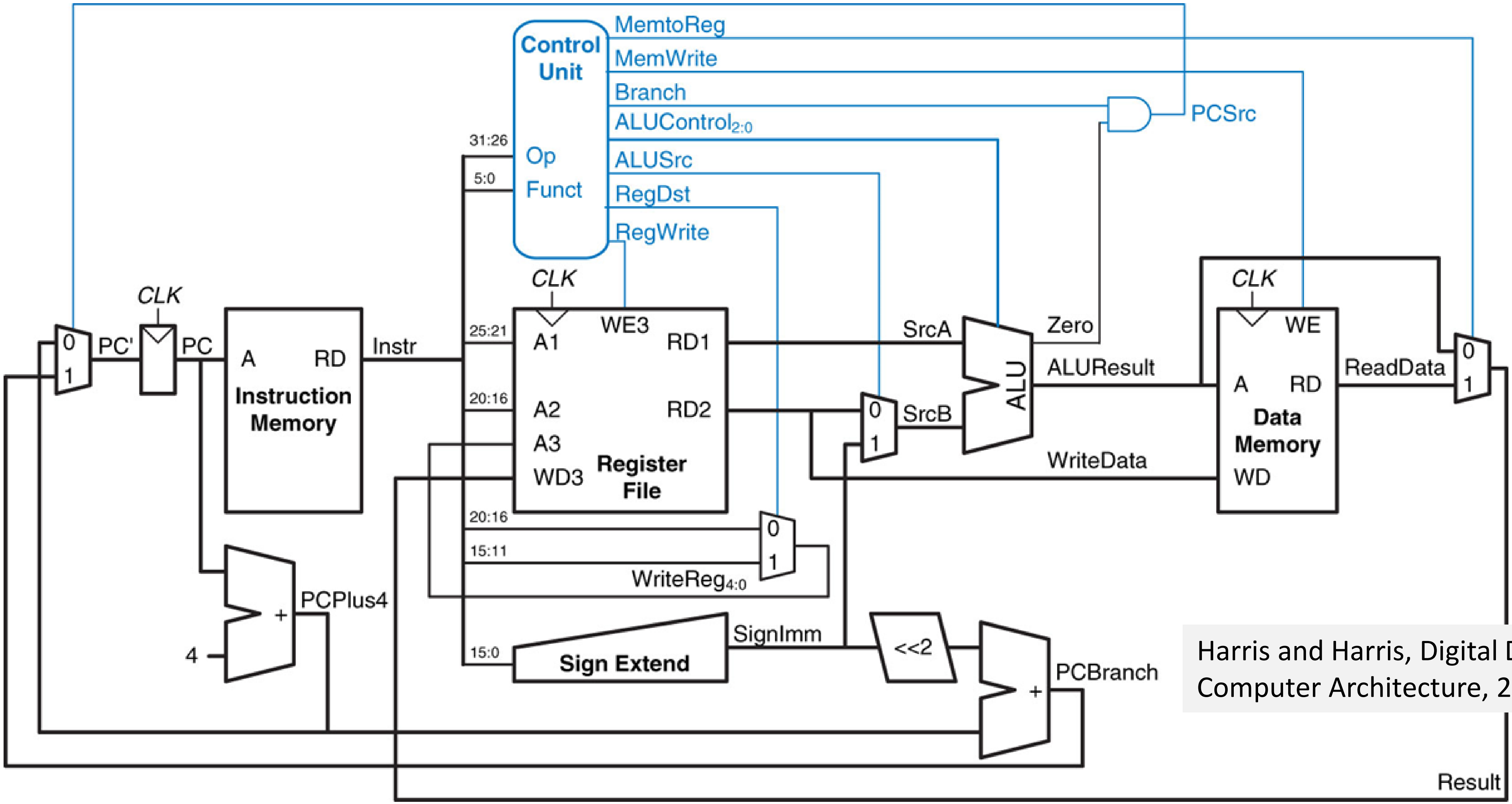


https://en.wikipedia.org/wiki/Von_Neumann_architecture

Fluxo 2

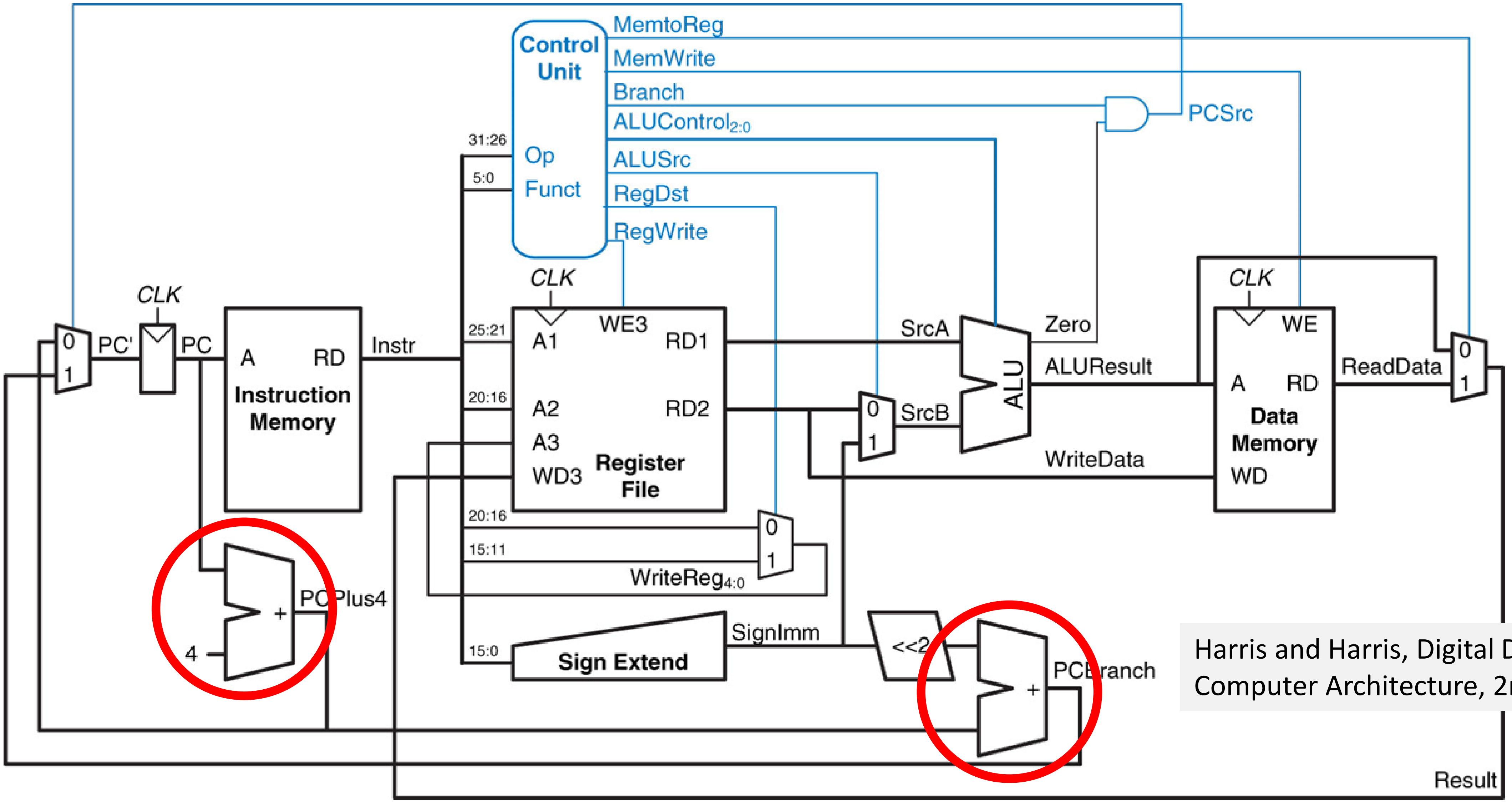
```
        LOAD R1, 0(R2)
        BEQ R1, R2, Salto
        SEND R1, 0(R2)
Salto:  RECV R1, 0(R2)
        ⋮
```

Caminho das instruções (dados e controle)



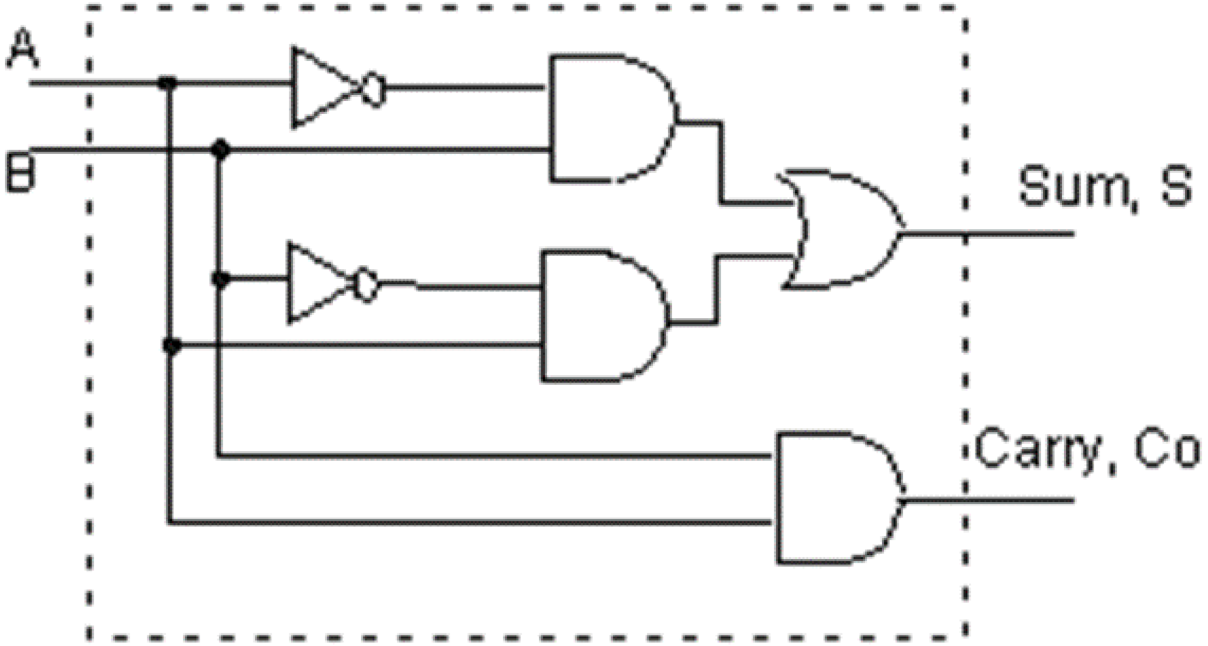
Harris and Harris, Digital Design and Computer Architecture, 2nd Edition

Caminho das instruções (dados e controle)



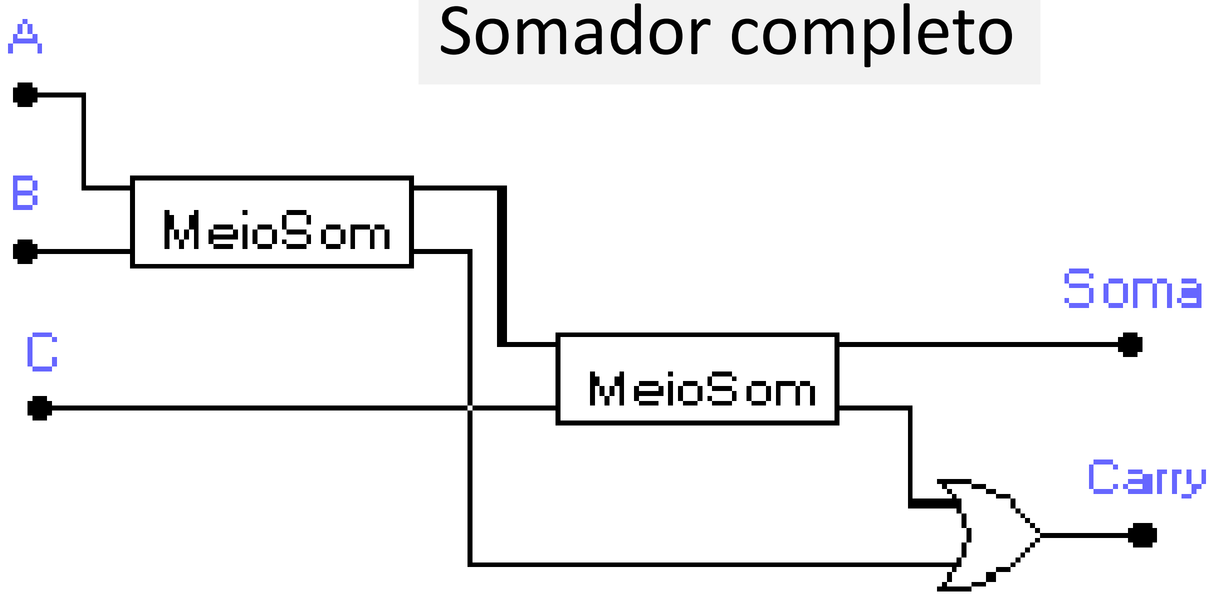
Harris and Harris, Digital Design and Computer Architecture, 2nd Edition

Somador de 1 bit

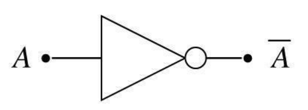
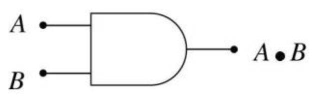
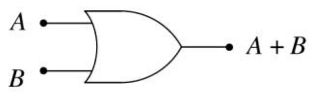
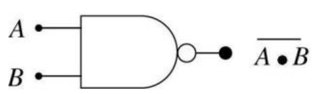
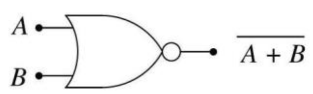
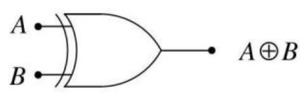
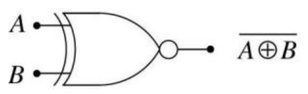


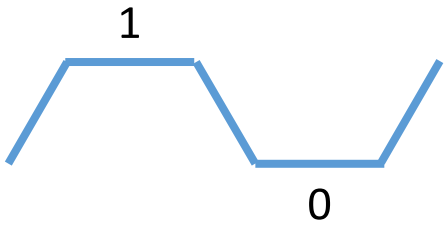
Meio somador

Somador completo



Portas Lógicas

	NOT: Operação que inverte o sinal de entrada
	AND: Saída igual a 1, se todas as entradas forem iguais a 1.
	OR: Saída igual a 1, se apenas uma entrada for igual a 1.
	NAND: Saída igual a 1, se apenas uma entrada for igual a 0.
	NOR: Saída igual a 1, se todas as entradas forem iguais a 0.
	XOR: Saída igual a 1, quando apenas uma das entradas for igual a 1.
	XNOR: Saída igual 1, quando todas as entradas forem iguais a 1 ou iguais a 0.



1: nível lógico alto, ex. 5V

0: nível lógico baixo, ex. 0V

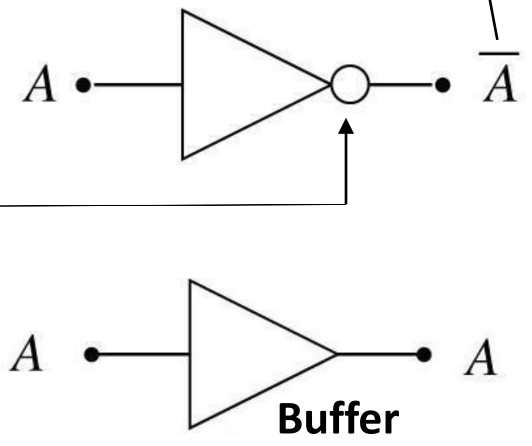
Portas lógicas (NOT)

- Operação **NOT**: Altera o valor de uma variável de 0 para 1 ou de 1 para 0.
- Tabela Verdade: Combinações dos bits de entrada e respectivos bits de saída após a operação lógica.
- O símbolo lógico (porta lógica) que representa a operação **NOT** é o seguinte:
- O segundo símbolo não é de um inversor, mas de um **buffer**. Repare que um buffer não possui um círculo na saída. Este círculo representa a inversão do bit de entrada e pode ser encontrado nas entradas e saídas de qualquer porta lógica, conforme veremos ainda nesta unidade.

Tabela Verdade

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

Esta barra Significa inversão



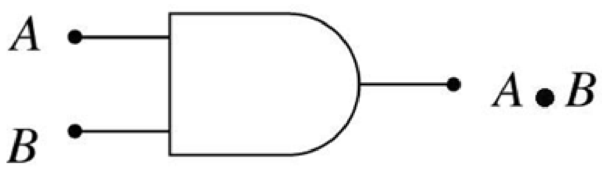
Portas lógicas (AND)

Porta AND

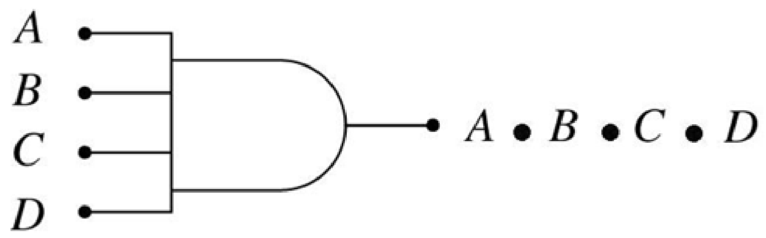
Tabela Verdade

A	B	A • B
0	0	0
0	1	0
1	0	0
1	1	1

Símbolos



Duas entradas



Quatro entradas. Uma porta lógica pode ter N entradas.

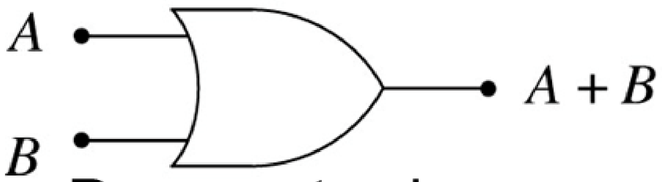
Portas lógicas (OR)

Porta OR

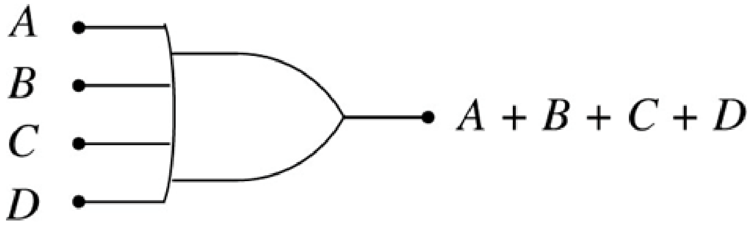
Tabela Verdade

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

Símbolos



Duas entradas



Quatro entradas. Uma porta lógica pode ter N entradas.

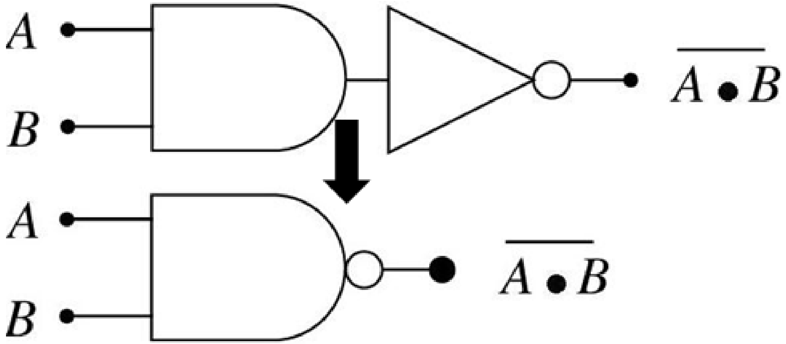
Portas lógicas (NAND)

Porta NAND

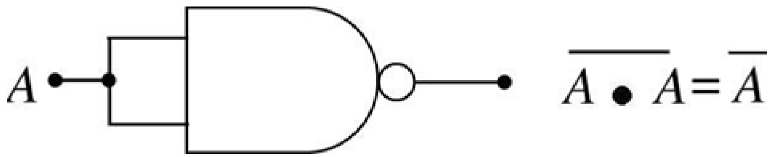
Tabela Verdade

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

Símbolos



Símbolo resultante.



Transformando uma NAND em uma NOT.

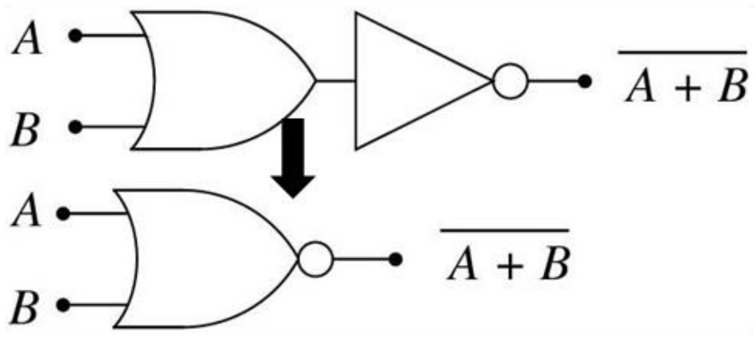
Portas lógicas (NOR)

Porta NOR

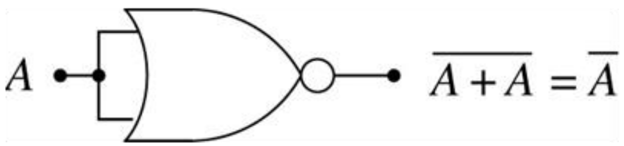
Tabela Verdade

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

Símbolos



Símbolo resultante.



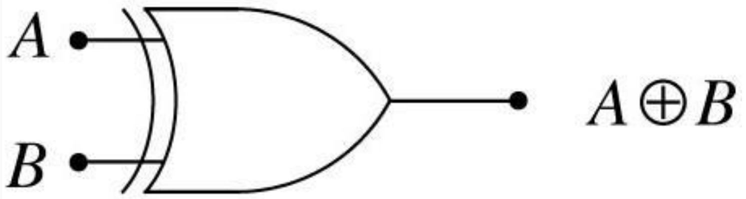
Transformando uma NOR em uma NOT.

Portas lógicas (XOR)

A operação **XOR** significa Exclusive-OR (OU Exclusivo). É muito parecida com a operação OR, mas a saída é igual a 1 quando somente **uma** das entradas for igual a 1.

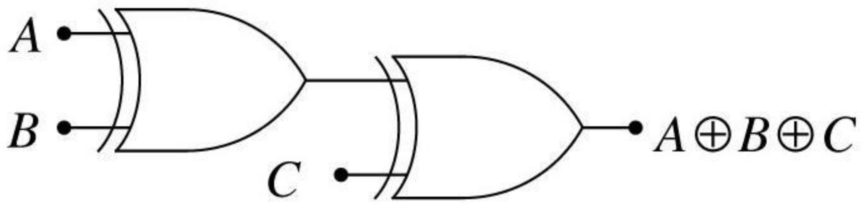
XOR

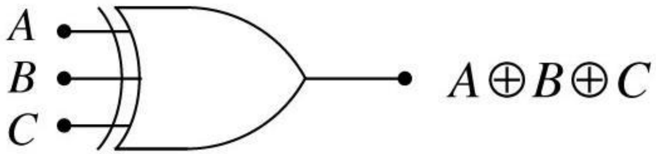
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



XOR 3 entradas

	A	B	C	$A \oplus B \oplus C$
	0	0	0	0
	0	0	1	1
	0	1	0	1
	0	1	1	0
	1	0	0	1
	1	0	1	0
par	1	1	0	0
ímpar	1	1	1	1



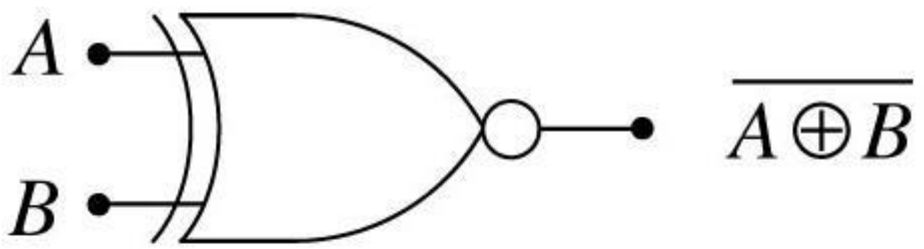


Portas lógicas (XNOR)

A operação **XNOR** significa Exclusive-NOR. A saída é zero quando **uma** das entradas for igual a 1.

XNOR

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



Propriedades básicas

- **Função NOT:** Se aplicarmos a operação **NOT** duas vezes sobre uma variável, esta terá seu valor original retornado.

$$\overline{(\overline{A})} = A$$

- **Função AND:** Sempre que uma das entradas for igual a 0 a saída será também igual a 0. Ou seja, 0 operação **AND** com qualquer outra coisa é igual a 0.

$$A \bullet 1 = A$$

$$A \bullet 0 = 0$$

$A \bullet A = A$ O teorema da idempotência diz que uma operação **AND** executada na mesma variável, resulta nela mesma

$A \bullet \overline{A} = 0$ A propriedade complementar:

$$1 \bullet \overline{1} = 0$$

$$0 \bullet \overline{0} = 0$$

Propriedades básicas

- **Função OR:** Sempre que uma das entradas for igual a 1 a saída será também igual a 1. Ou seja, 1 operação **OR** com qualquer outra coisa é igual a 1.

$$A + 1 = 1$$

$$A + 0 = A$$

O teorema da idempotência diz que uma operação **OR** executada na mesma variável, resulta nela mesma:

$$A + A = A$$

$$1 + 1 = 1$$

$$0 + 0 = 0$$

Baseado que sempre 1 + 0 = 1, chegamos na propriedade complementar:

$$A + \overline{A} = 1$$

Leis da Álgebra

Leis Comutativas: é possível trocar a ordem das variáveis sem alterar o resultado.

$$A + B = B + A$$

$$A \bullet B = B \bullet A$$

Leis Associativas: utilizadas para definir uma ordem para Execução das operações.

$$A + B + C = (A + B) + C = A + (B + C)$$

$$A \bullet B \bullet C = (A \bullet B) \bullet C = A \bullet (B \bullet C)$$

Quais os termos nas expressões abaixo são executados primeiro?

$D = A \bullet (B + C)$ Termo entre parênteses. No caso, operação **OR**.

$D = A \bullet B + C$ Se não houver parênteses a operação **AND** tem prioridade.

Leis Distributivas: Seguem os exemplos abaixo:

$$A \bullet (B + C) = (A \bullet B) + (A \bullet C)$$

$$A + (B \bullet C) = (A + B) \bullet (A + C)$$

Leis da Álgebra

- É importante ressaltar que estas leis já são conhecidas por nós. Estamos aplicando a matemática para entender as diversas expressões lógicas que estamos estudando.

- Durante o projeto de um circuito lógico algumas expressões complexas são obtidas e é necessário que sejam simplificadas, para se obter um circuito mais simples. Uma vez que o circuito lógico pode ser resultado da expressão e vice-e-versa.

- Simplificações algébricas e extração de circuitos lógicos de expressões booleanas fazem parte do nosso estudo.

Teorema de DeMorgan

- Utilizando as portas lógicas NOT, OR, AND, NOR e NAND podemos construir circuitos alternativos, diferentes, mas com o mesmo resultado de saída.

- O teorema de **DeMorgan** trabalha basicamente sobre operações NOR e NAND. Através deste teorema é possível mostrar que uma operação NOR é equivalente à operação AND dos complementos das variáveis e uma operação NAND é equivalente à operação OR dos complementos das variáveis, conforme é mostrado a seguir:

$$\overline{(A+B)} = \overline{A} \bullet \overline{B}$$

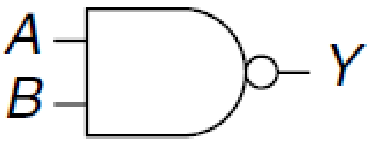
$$\overline{(A+B+C)} = \overline{A} \bullet \overline{B} \bullet \overline{C}$$

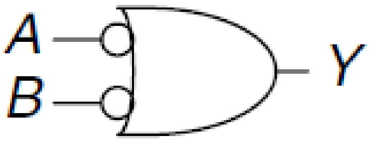
$$\overline{(A \bullet B)} = \overline{A} + \overline{B}$$

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

Teorema de DeMorgan

NAND

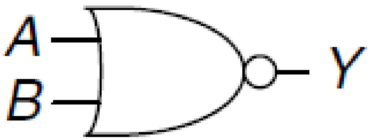


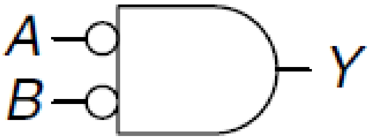


$$Y = \overline{AB} = \overline{A} + \overline{B}$$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NOR



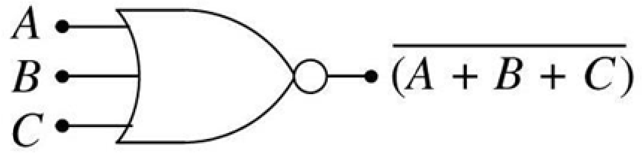


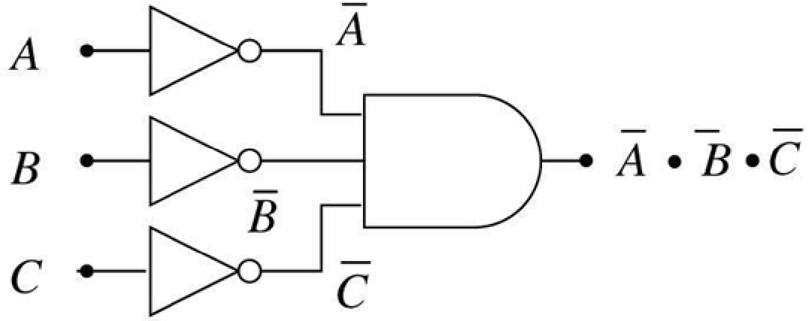
$$Y = \overline{A + B} = \overline{A}\,\overline{B}$$

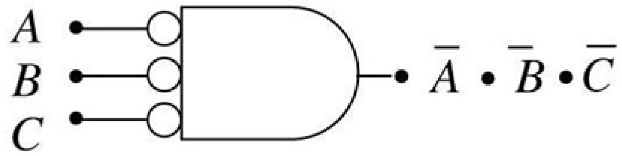
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Teorema de DeMorgan (circuitos resultantes)

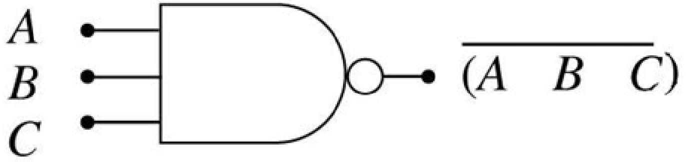
$$\overline{(A+B+C)} = \overline{A}\bullet\overline{B}\bullet\overline{C}$$

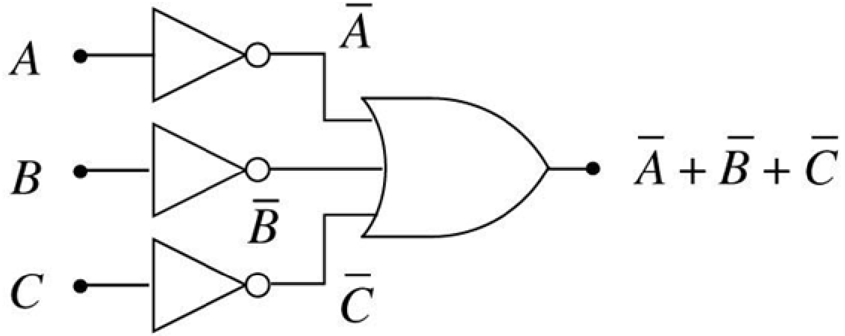


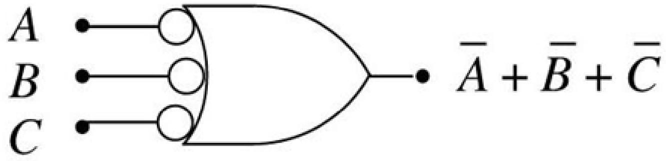




$$\overline{(A\bullet B\bullet C)} = \overline{A}+\overline{B}+\overline{C}$$







Identidades Booleanas úteis

$$A + A \bullet B = A$$

A	B	$A \bullet B$	$A + A \bullet B$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

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

A	B	$\bar{A}$	$\bar{A} \bullet B$	$A + \bar{A} \bullet B$	$A+B$
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

- As identidades booleanas apresentadas acima são confirmadas através de suas respectivas tabelas verdades.

- Podemos também verificar a primeira identidade através de atribuição de valores às variáveis:

$$A + A \bullet 0 = A + 0 = A$$

$$A + A \bullet 1 = A + A = A$$

- Ou através da lei distributiva:

$$A + A \bullet B = A \bullet (1 + B) \quad = A \bullet (1) = A$$

Identidades Booleanas úteis

OR Identities	AND Identities
$A + 0 = A$	$A \bullet 0 = 0$
$A + 1 = 1$	$A \bullet 1 = A$
$A + A = A$	$A \bullet A = A$
$A + \overline{A} = 1$	$A \bullet \overline{A} = 0$
$\overline{\overline{A}} = A$	
$A + B = B + A$	$A \bullet B = B \bullet A$
$A + (B + C) = (A + B) + C$	$A \bullet (B \bullet C) = (A \bullet B) \bullet C$
$A \bullet (B + C) = A \bullet B + A \; C$	$A + (B \bullet C) = (A + B) \; (A + C)$
$\overline{(A + B)} = \overline{A} \bullet \overline{B}$	$\overline{(A \bullet B)} = \overline{A} + \overline{B}$
$A + A \bullet B = A$	$A + \overline{A} \bullet B = A + B$

Uyemura, Sistemas Digitais, 2002

Arquitetura de Computadores I

Aula 6

Identidades Booleanas úteis

	Theorem		Dual	Name
T6	$B \bullet C = C \bullet B$	T6′	$B + C = C + B$	Commutativity
T7	$(B \bullet C) \bullet D = B \bullet (C \bullet D)$	T7′	$(B + C) + D = B + (C + D)$	Associativity
T8	$(B \bullet C) + (B \bullet D) = B \bullet (C + D)$	T8′	$(B + C) \bullet (B + D) = B + (C \bullet D)$	Distributivity
T9	$B \bullet (B + C) = B$	T9′	$B + (B \bullet C) = B$	Covering
T10	$(B \bullet C) + (B \bullet \overline{C}) = B$	T10′	$(B + C) \bullet (B + \overline{C}) = B$	Combining
T11	$(B \bullet C) + (\overline{B} \bullet D) + (C \bullet D)$ $= B \bullet C + \overline{B} \bullet D$	T11′	$(B + C) \bullet (\overline{B} + D) \bullet (C + D)$ $= (B + C) \bullet (\overline{B} + D)$	Consensus
T12	$\overline{B_0 \bullet B_1 \bullet B_2 \ldots}$ $= (\overline{B_0} + \overline{B_1} + \overline{B_2} \ldots)$	T12′	$\overline{B_0 + B_1 + B_2 \ldots}$ $= (\overline{B_0} \bullet \overline{B_1} \bullet \overline{B_2})$	De Morgan's Theorem

Harris and Harris, Digital Design and Computer Architecture, 2nd Edition

Simplificações algébricas

Exemplo:

$$f = A \bullet B + A \bullet \overline{B}$$

$$f = A \bullet \left(B + \overline{B}\right)$$

$$f = A \bullet 1 = A$$

O exemplo ao lado apresenta uma situação onde há duas entradas no circuito: A e B.

No entanto, ao simplificarmos a equação verificamos que o circuito precisa apenas de uma das entradas, somente a entrada A.

A simplificação algébrica é importante durante o projeto de um determinado circuito lógico.

Através da simplificação de uma equação é possível reduzir o tamanho e o custo de um determinado circuito e por consequência de um *hardware* final.

Simplificações algébricas (exercícios)

$$(1)\, f = A \bullet B \bullet C + B \bullet C$$

$$(2)\, f = \overline{\left(A + \overline{B} + C\right) + \left(B + \overline{C}\right)}$$

$$(3)\, f = \left(A + B + C\right) \bullet (A + B)$$

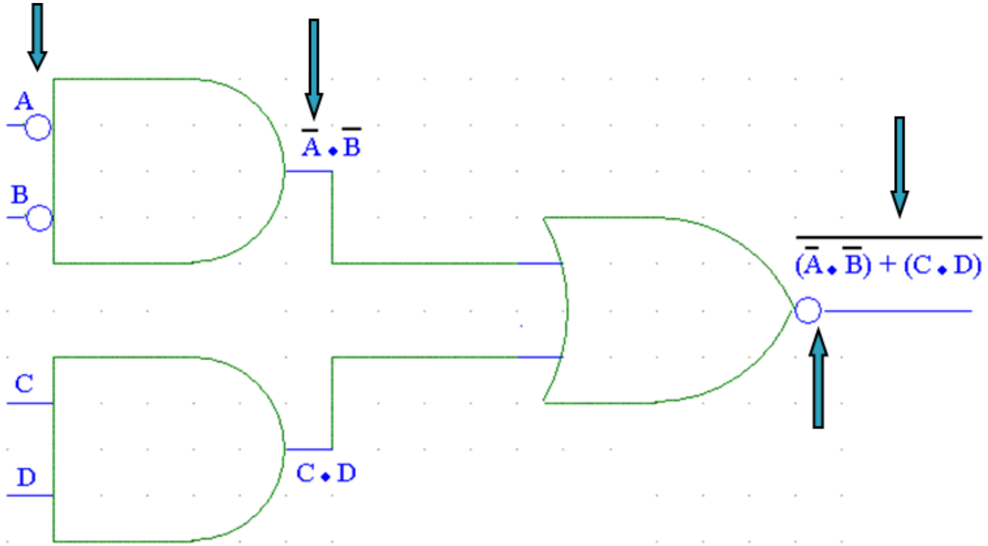
$$(4)\, f = A \bullet B + A \bullet B \bullet C + A \bullet B \bullet \overline{C}$$

$$(5)\, f = 1 + A \bullet \left(A + \overline{B} + C + \overline{D} + \overline{E}\right) \bullet \left(\overline{B + A}\right) \bullet C \bullet D$$

$$(6)\, f = \left(\overline{\left(\overline{A + B}\right) \bullet \left(\overline{B + C}\right)}\right) \bullet \left(A + B + C\right) \bullet 0$$

Expressões booleanas de circuitos lógicos

Exemplo



- O primeiro passo é escrever a expressão de saída de cada bloco básico (porta lógica).
- Se a entrada de uma porta lógica possui o círculo que representa a inversão, somente esta entrada (letra no desenho) receberá a barra que identifica sinal invertido:

$$A \Rightarrow \overline{A} \quad \text{ou} \quad B \Rightarrow \overline{B}$$

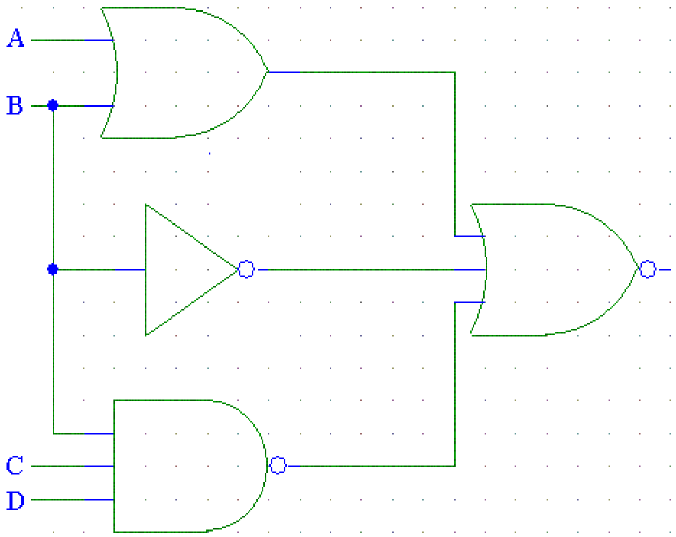
- Se a saída de uma porta lógica possui o círculo, símbolo que representa a inversão, toda a expressão de saída desta porta lógica receberá a barra que identifica sinal invertido:

$$\overline{\left(\overline{A} \bullet \overline{B}\right) + \left(C \bullet D\right)}$$

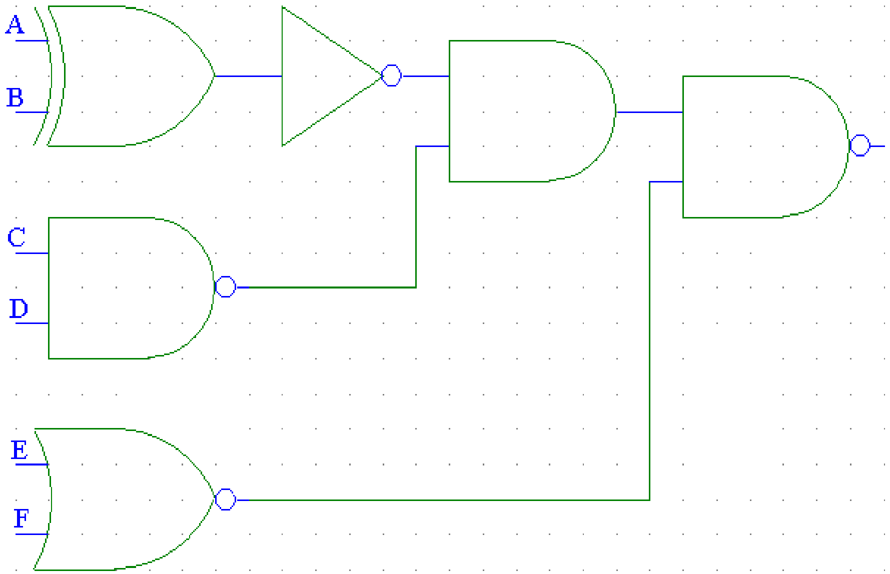
Extrair expressões booleanas de circuitos lógicos

exercícios

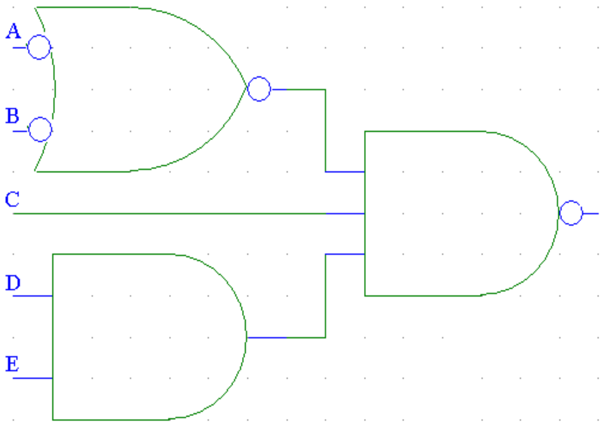
(7)



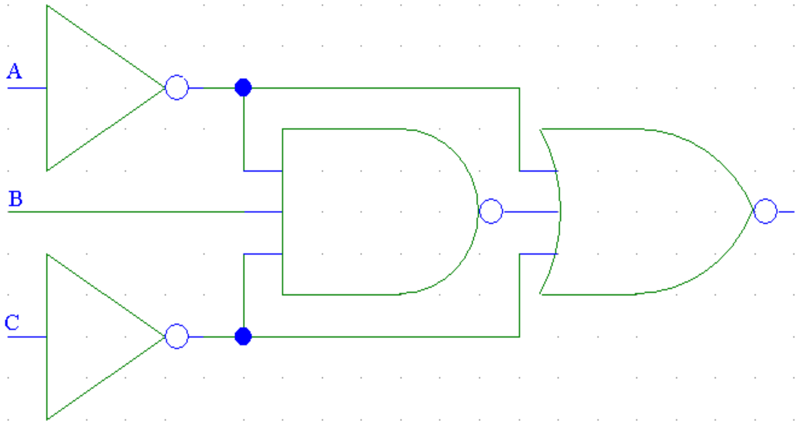
(9)



(8)



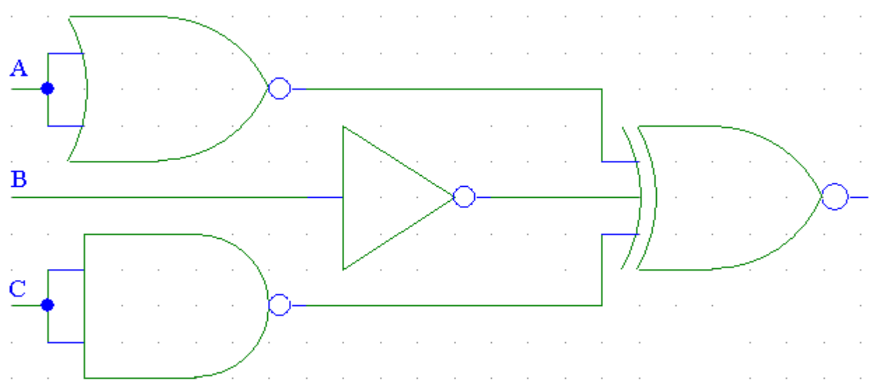
(10)



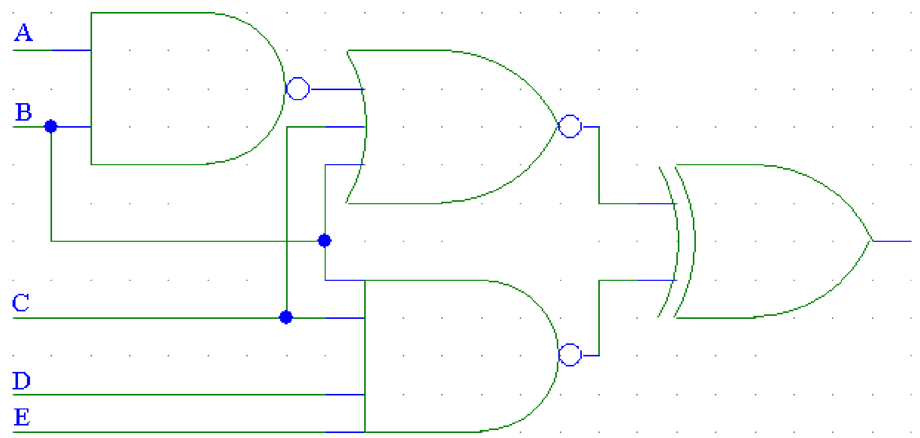
Extrair expressões booleanas de circuitos lógicos

exercícios

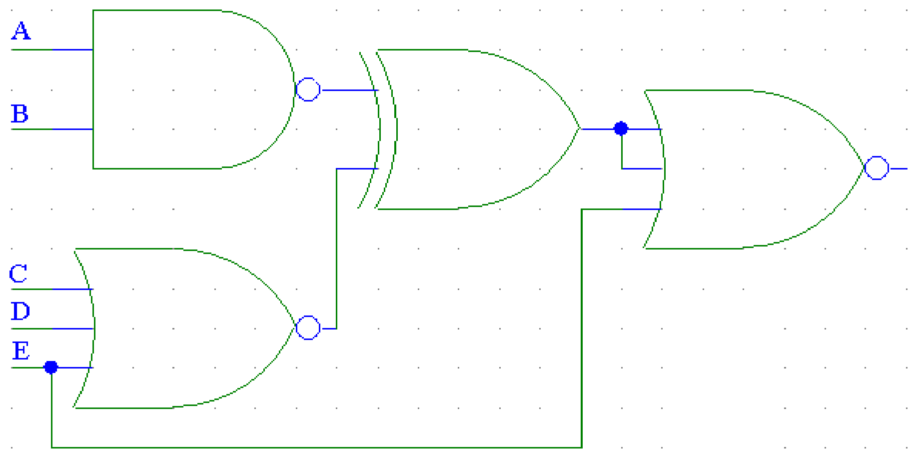
(11)



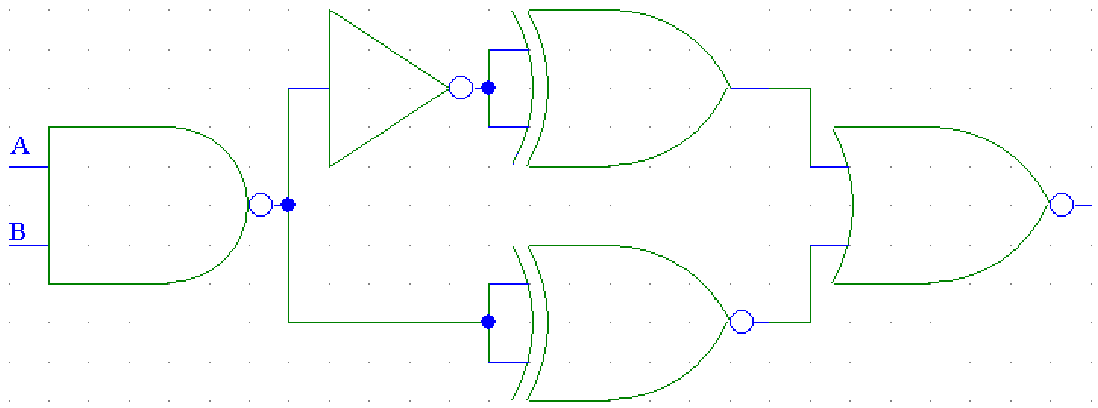
(13)



(12)



(14)



Circuitos lógicos obtidos de expressões booleanas

Exemplo

Entrada C invertida. Operação **NOT**

$$f = \overline{(A+B)\bullet \overline{C} \bullet \overline{(B+D)}}$$

Operação **OR**

Operação **NOR**

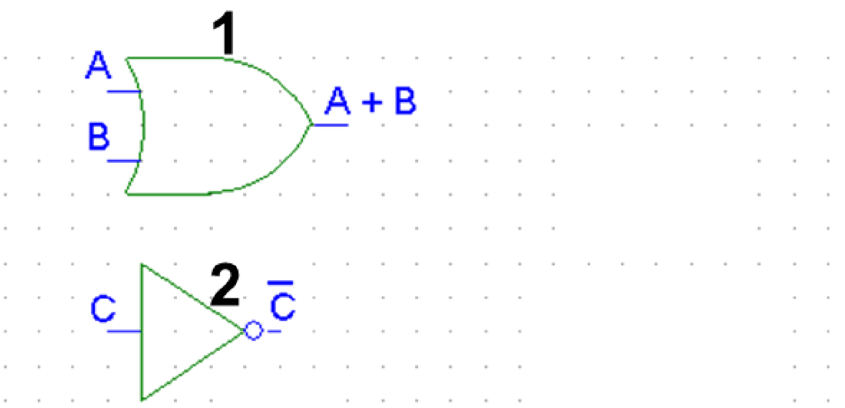
Operação **AND**

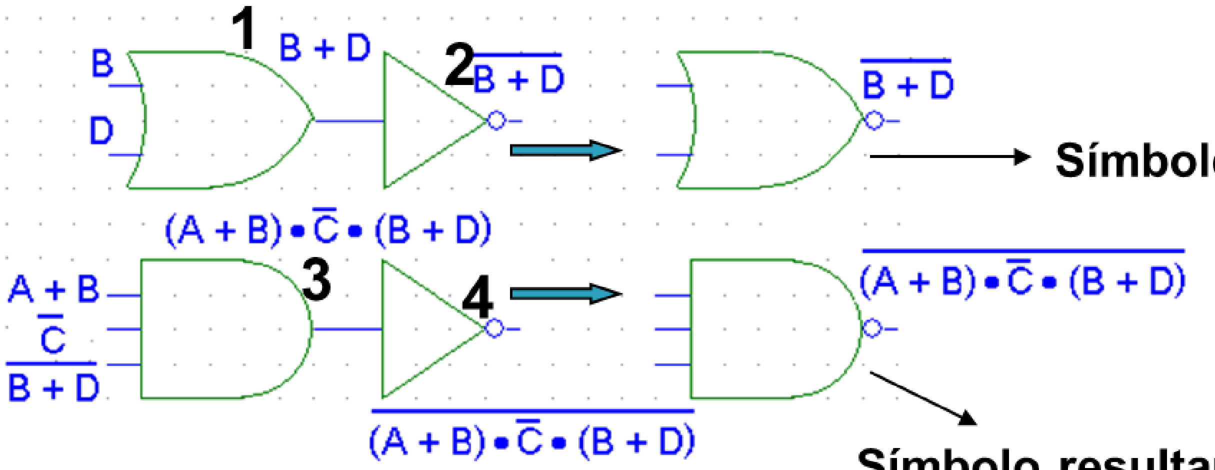
- Primeiro passo: extrair o circuito representado pelas operações entre parênteses.
- Segundo passo: extrair o circuito invertendo os sinais de entrada ou saída onde houver uma barra na expressão.
- Terceiro passo: extrair o circuito representado pelas operações que estão fora dos parênteses.
- Quarto passo: extrair o circuito invertendo os sinais de entrada ou saída onde houver uma barra na expressão.
- **Observação:** Quando não houver parênteses deve ser respeitado a prioridade de operação entre os sinais. Por exemplo: Primeiro AND depois OR. Os passos acima **não** são uma regra! É necessário identificar os termos prioritários em cada expressão lógica.

Circuitos lógicos obtidos de expressões booleanas

Exemplo $$f = \overline{(A+B)\bullet\overline{C}\bullet\overline{(B+D)}}$$

1	Primeiro passo
2	Segundo passo
3	Terceiro passo
4	Quarto passo



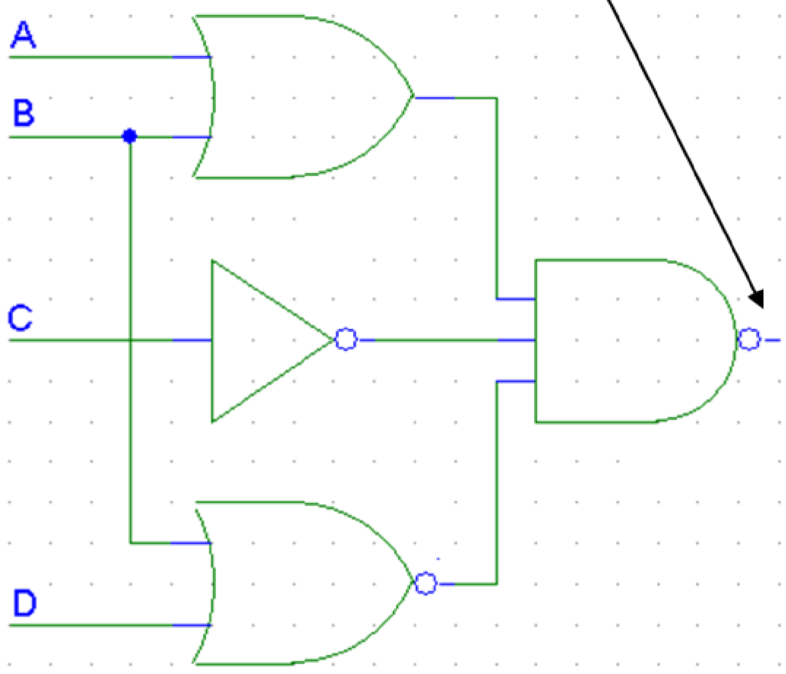


Símbolo resultante = NOR (2 entradas)

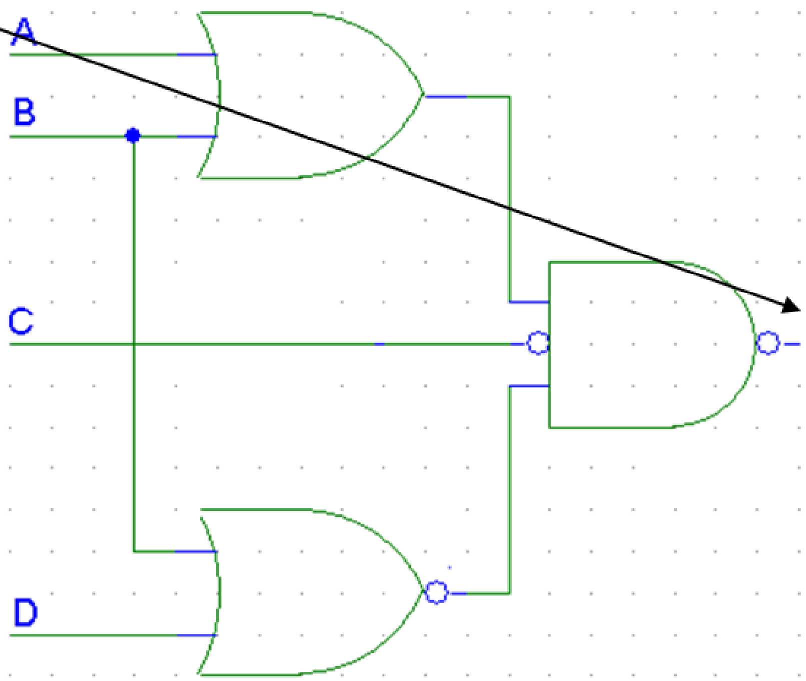
Símbolo resultante = NAND (3 entradas)

Circuitos lógicos obtidos de expressões booleanas

Exemplo $f = \overline{\left(A+B\right)\bullet\overline{C}\bullet\left(\overline{B+D}\right)}$



Ou



Circuitos resultantes

Circuitos lógicos obtidos de expressões booleanas

exercícios

$$(15)\, f = A\bullet B\bullet C + (A+B)\bullet C$$

$$(16)\, f = \overline{\left(\overline{A\bullet B}+\overline{C\bullet D}\right)}$$

$$(17)\, f = \left(\overline{\left(\overline{A}+B\right)}+\overline{\left(\overline{C}+D\right)}\right)\bullet\overline{D}$$

$$(18)\, f = \overline{\left(\overline{\left(\overline{A}\bullet B\right)}+\overline{\left(C\bullet\overline{D}\right)}\right)}\bullet E + \left(\left(A\bullet\overline{D}\bullet\overline{E}\right)+\left(C\bullet D\bullet E\right)\right)\bullet\overline{A}$$

$$(19)\, f = \overline{(A\oplus B)}+C+D\bullet E+A+(C\oplus D)$$

$$(20)\, f = \overline{\left(\overline{A}+\left(\overline{B}\oplus\overline{C}\right)\right)\bullet(A+B)\bullet\overline{(A+C)}}$$

Lembrar da observação anterior! Quem tem prioridade?

Arquitetura de Computadores I

Aula 7

Tabela Verdade obtida de uma expressão booleana

$$f = \underline{A \bullet B \bullet C} + \underline{A \bullet D} + \underline{A \bullet B \bullet D}$$

Exemplo

				Primeiro membro	Segundo membro	Terceiro membro	Resultado
A	B	C	D	A . B . C	A . D	A . B . D	Saída
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	1
1	0	1	0	0	0	0	0
1	0	1	1	0	1	0	1
1	1	0	0	0	0	0	0
1	1	0	1	0	1	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	1	1	1

Tabela Verdade obtida de uma expressão booleana

- A tabela anterior foi construída da seguinte forma:
  - Na coluna do primeiro membro é colocado o resultado da operação **AND** entre A, B e C.
  - Na coluna do segundo membro é colocado o resultado da operação **AND** entre A e D.
  - Na coluna do terceiro membro é colocado o resultado da operação **AND** entre A, B e D.
  - Na coluna Saída é colocado o resultado da operação **OR** entre o primeiro, segundo e terceiro membros.

Tabela Verdade obtida de uma expressão booleana

exercícios

$$(21)f = \overline{A} + B + A \bullet B \bullet \overline{C}$$

$$(22)f = (A \oplus B) + \overline{C \bullet D} + \overline{B}$$

$$(23)f = C \bullet B + B \bullet A + (\overline{A} + \overline{C})$$

$$(24)f = A + B \bullet \overline{C} \bullet A + B$$

Expressão booleana obtida de uma tabela verdade

Existem duas formas para se obter a expressão booleana através de tabelas verdades. Elas são conhecidas como **soma de produtos** e produto de somas. A forma mais intuitiva e, portanto, mais usual é a soma de produtos.

A	B	C	mintermo
0	0	0	$\bar{A} . \bar{B} . \bar{C}$
0	0	1	$\bar{A} . \bar{B} . C$
0	1	0	$\bar{A} . B . \bar{C}$
0	1	1	$\bar{A} . B . C$
1	0	0	$A . \bar{B} . \bar{C}$
1	0	1	$A . \bar{B} . C$
1	1	0	$A . B . \bar{C}$
1	1	1	$A . B . C$

Saída = 1

Expressão booleana obtida de uma tabela verdade

Existem duas formas para se obter a expressão booleana através de tabelas verdades. Elas são conhecidas como soma de produtos e **produto de somas**. A forma mais intuitiva e, portanto, mais usual é a soma de produtos.

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

Saída = 0

Expressão booleana obtida de uma tabela verdade

Soma de Produtos (SdP)

Cada termo produto é denominado **mintermo**.

A	B	C	Saída
0	0	0	0
0	0	1	0
0	1	0	**1**
0	1	1	**1**
1	0	0	0
1	0	1	**1**
1	1	0	**1**
1	1	1	0

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

Forma canônica: $F = m2 + m3 + m5 + m6$

Expressão booleana obtida de uma tabela verdade

Produto de Somas (PdS)

Cada termo soma é denominado **maxtermo**.

A	B	C	Saída
0	0	0	**0**
0	0	1	**0**
0	1	0	1
0	1	1	1
1	0	0	**0**
1	0	1	1
1	1	0	1
1	1	1	**0**

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

Forma canônica: F = M0 . M1 . M4 . M7

Expressão booleana obtida de uma tabela verdade

exercícios

(25)

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

(26)

A	B	C	S1	S2	S3
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	1	0	0

Expressão booleana obtida de uma tabela verdade

exercícios

(27)

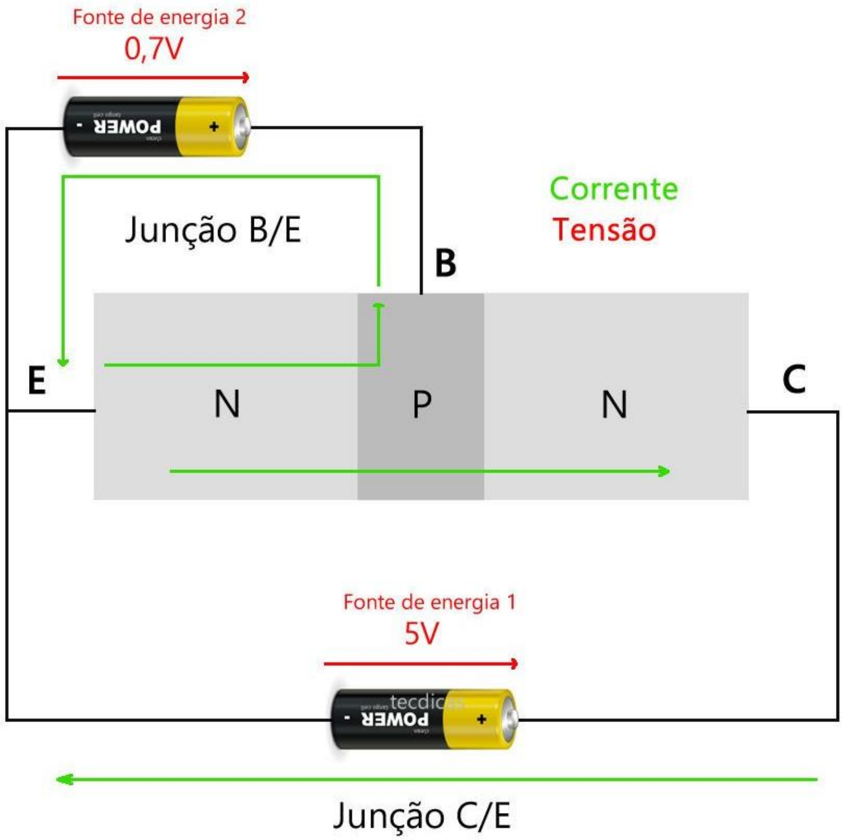
A	B	C	S1	S2	S3
0	0	0	0	1	1
0	0	1	1	0	0
0	1	0	1	1	1
0	1	1	1	1	0
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	0	1	1
1	1	1	1	1	0

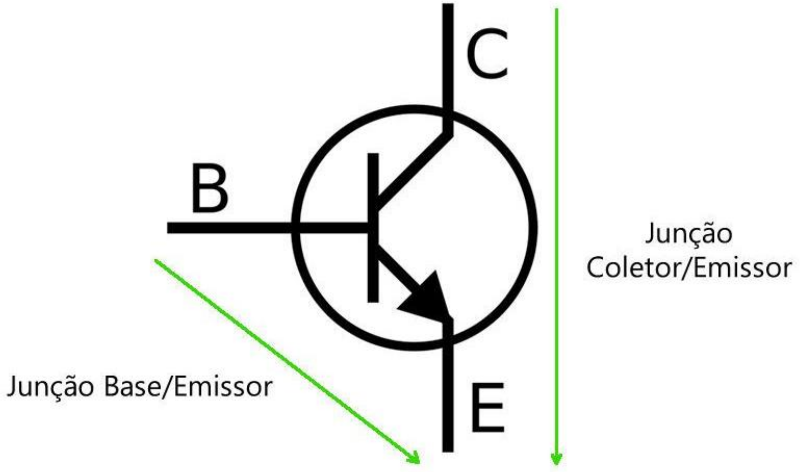
(28)

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

Como construir portas lógicas?

Transistor NPN

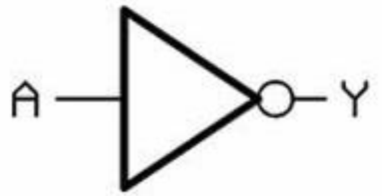




<u>Saber mais em</u>:
Dispositivos Eletrônicos e Teoria dos Circuitos, Pearson
Robert L. Boylestad, Louis Nashelsky

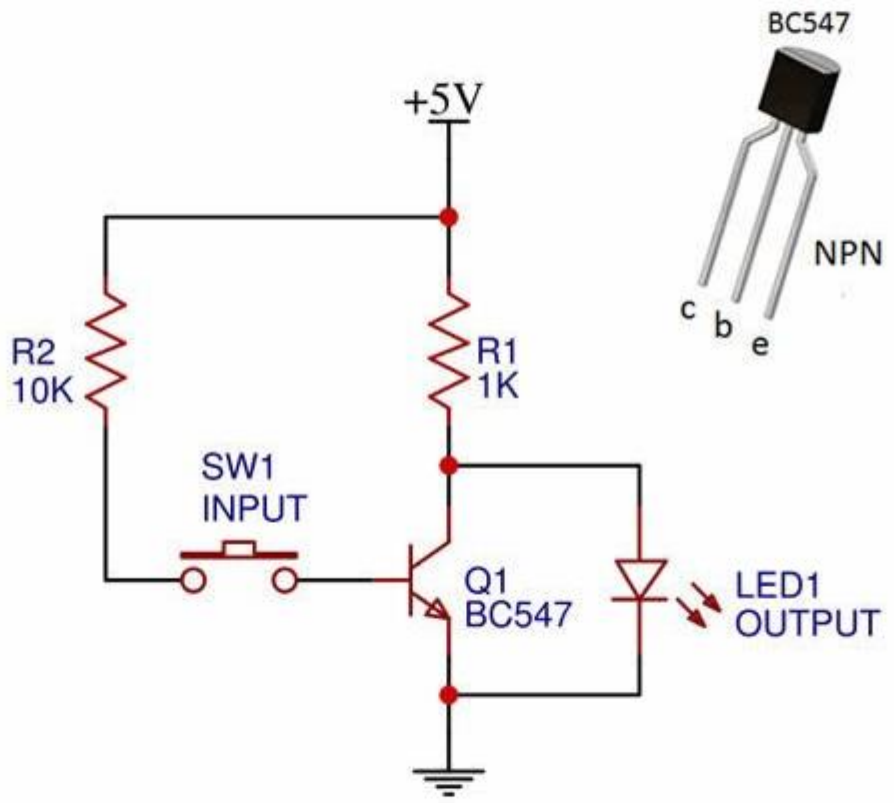
Figuras retiradas de: https://tecdicas.com/criando-portas-logicas-com-transistores/

Como construir portas lógicas?



$$Y = \overline{A}$$

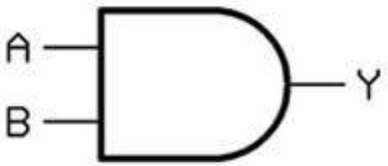
Porta NOT	
Input	Output
A	Y
0	1
1	0



<u>Saber mais em</u>:
Dispositivos Eletrônicos e Teoria dos Circuitos, Pearson
Robert L. Boylestad, Louis Nashelsky

Figuras retiradas de: https://tecdicas.com/criando-portas-logicas-com-transistores/

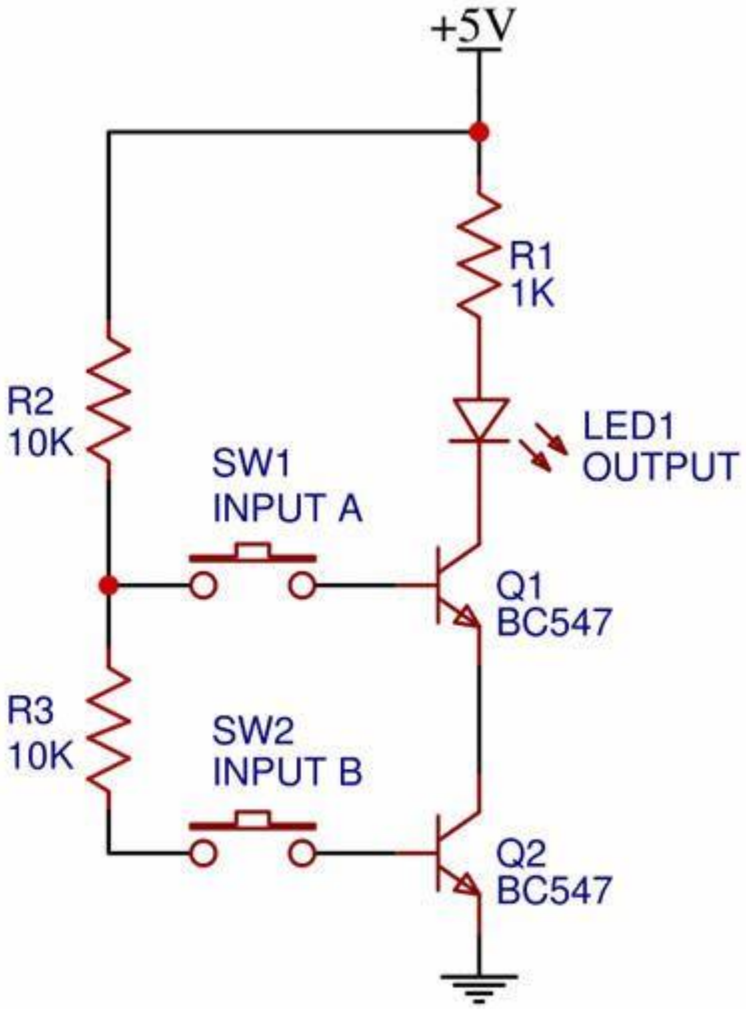
Como construir portas lógicas?



$$Y = AB$$

PORTA AND		
Input		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

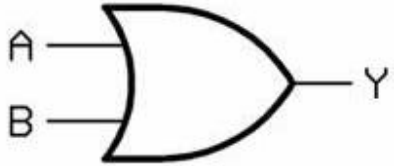
tecdicas



Saber mais em:
Dispositivos Eletrônicos e Teoria dos Circuitos, Pearson
Robert L. Boylestad, Louis Nashelsky

Figuras retiradas de: https://tecdicas.com/criando-portas-logicas-com-transistores/

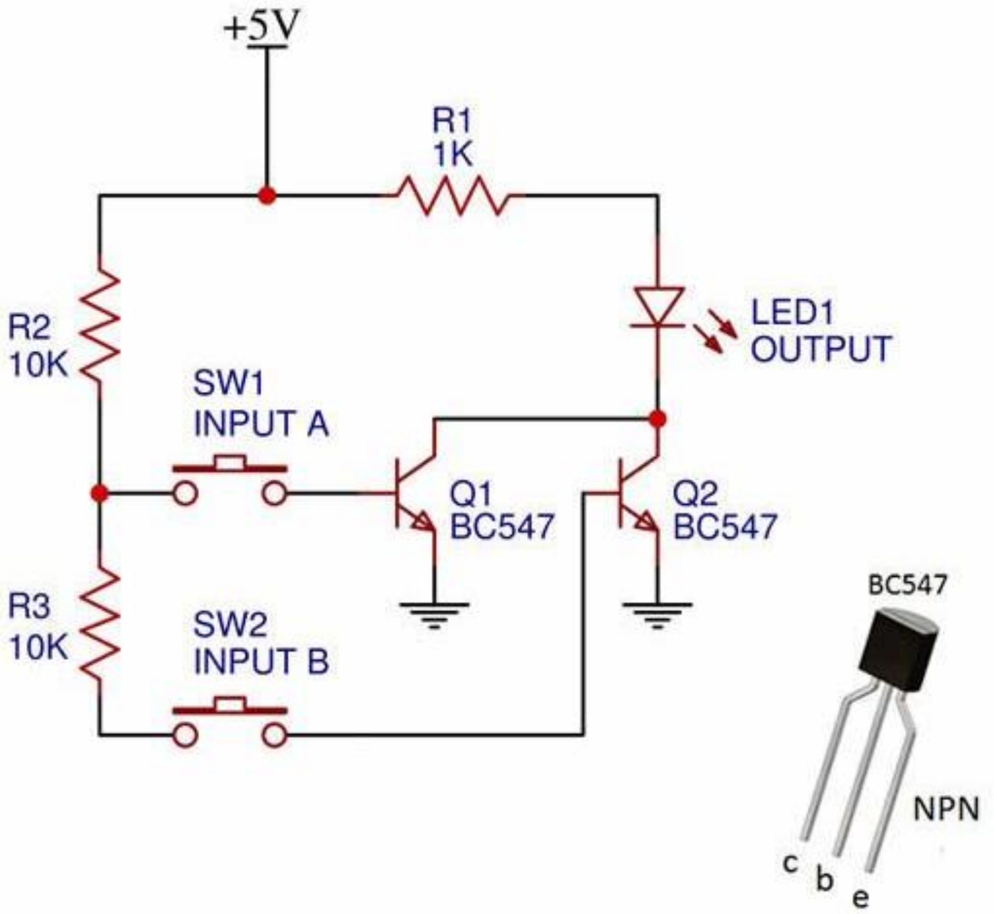
Como construir portas lógicas?



$$Y = A + B$$

PORTA OR		
Input		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

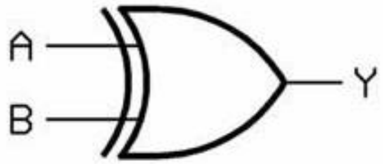
tecdicas



<u>Saber mais em</u>:
Dispositivos Eletrônicos e Teoria dos Circuitos, Pearson
Robert L. Boylestad, Louis Nashelsky

Figuras retiradas de: https://tecdicas.com/criando-portas-logicas-com-transistores/

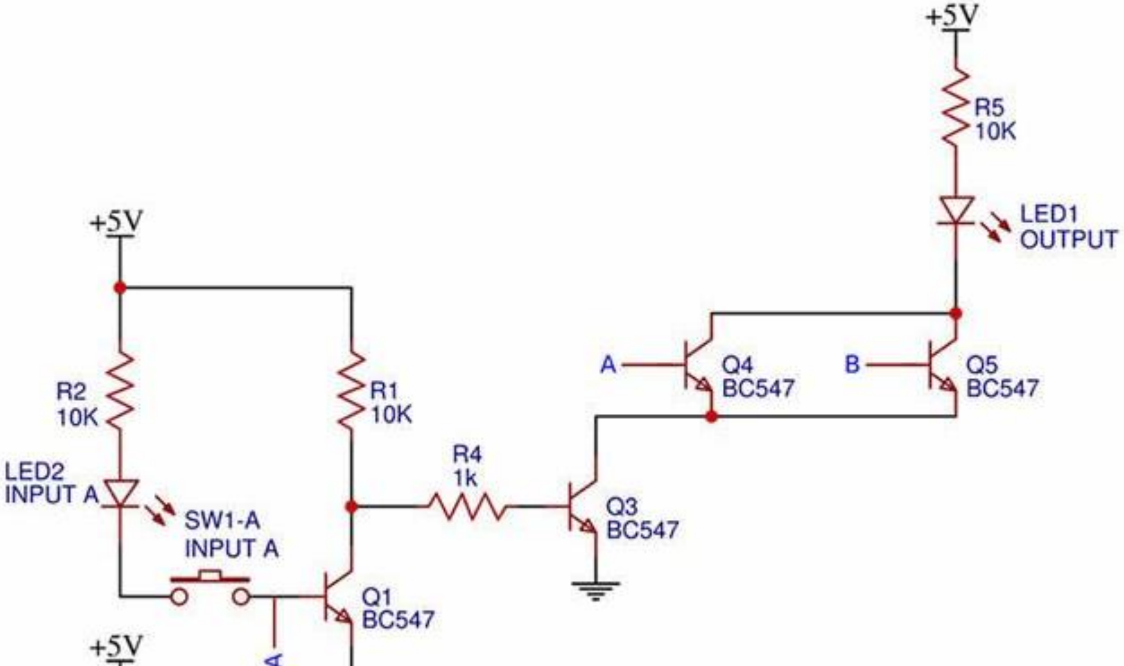
Como construir portas lógicas?



$$Y = A \oplus B$$

PORTA XOR		
Input		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

tecdicas



<u>Saber mais em</u>:
Dispositivos Eletrônicos e Teoria dos Circuitos, Pearson
Robert L. Boylestad, Louis Nashelsky

Figuras retiradas de: https://tecdicas.com/criando-portas-logicas-com-transistores/

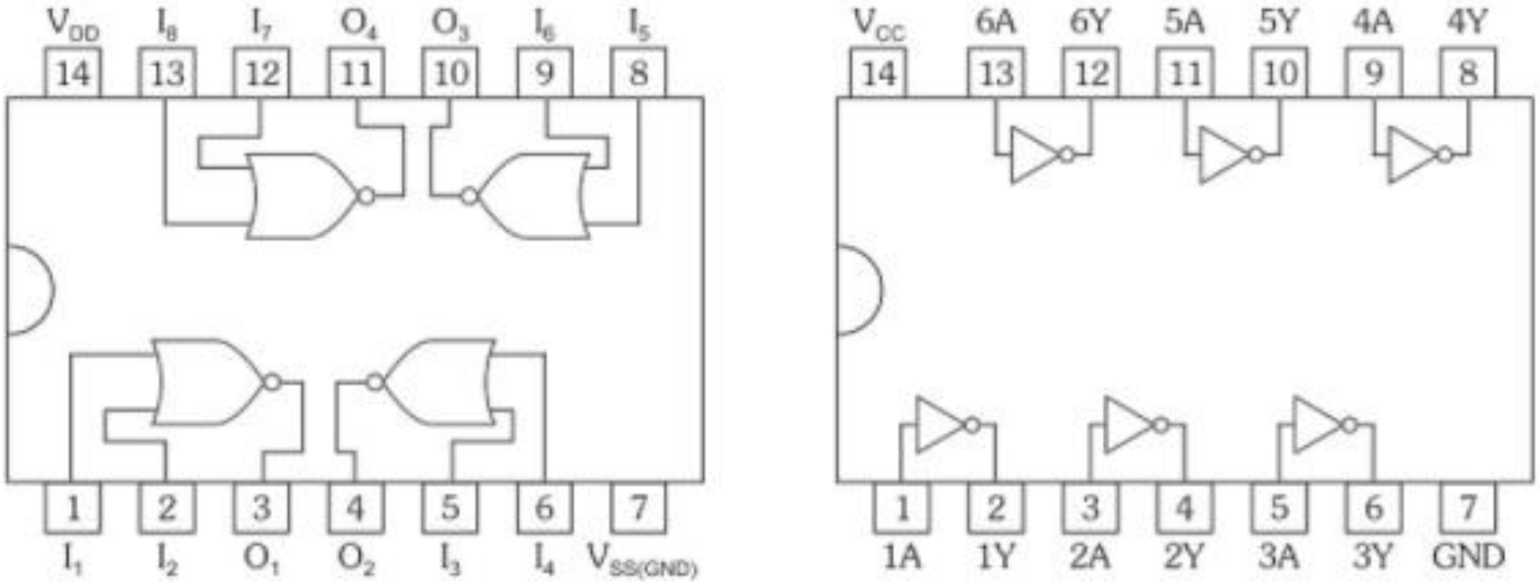
Circuitos integrados (CIs ou Chips)

Portas lógicas em circuitos integrados

Amplie seus conhecimentos

Sistemas Digitais: Circuitos Combinacionais e Sequenciais - Francisco Gabriel Capuano

Na prática as portas lógicas são encontradas dentro de circuitos integrados comerciais específicos, pertencentes a uma determinada família de circuitos lógicos, como ilustra a Figura 1.13.



(a) 4001B (b) 74HC04/74HCT04

Figura 1.13 - Pinagem e configuração interna de circuitos integrados da família CMOS: (a) 4001B: 4 portas NOU de 2 entradas e (b) 74HCT04: 6 inversores CMOS.

Arquitetura de Computadores I

Aula 8

Mapa de Karnaugh (Simplificação de Expressão Booleana)

- Método gráfico usado para simplificar uma expressão booleana (produzir circuitos lógicos com um número menor de portas e maior desempenho

- Precisa da tabela verdade completa de um circuito ou da expressão booleana completa (todos os termos possuem todas as entradas) do circuito

A	B	X	
0	0	1	$\longrightarrow \overline{A}\,\overline{B}$
0	1	0	
1	0	0	
1	1	1	$\longrightarrow AB$

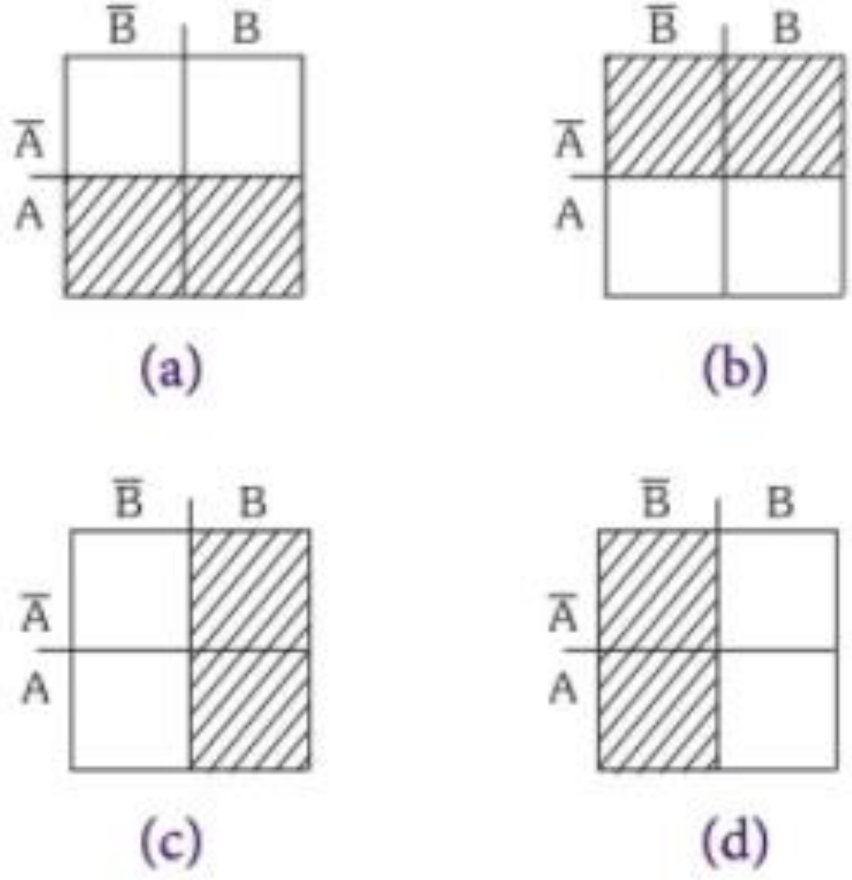
	$\overline{B}$	B
$\overline{A}$	1	0
A	0	1

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

Mapa de Karnaugh (Simplificação de Expressão Booleana)

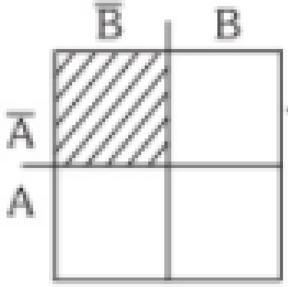
A	B
0	0
0	1
1	0
1	1

→ caso 0
→ caso 1
→ caso 2
→ caso 3

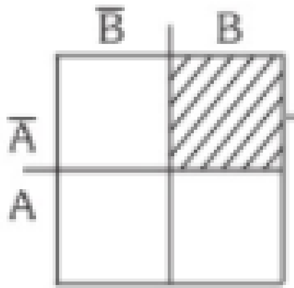


(a) região onde A = 1. (b) região onde A = 0 ($\overline{A}$ = 1).
(c) região onde B = 1. (d) região onde B = 0 ($\overline{B}$ = 1).

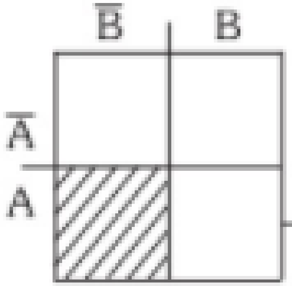
Mapa de Karnaugh (Simplificação de Expressão Booleana)



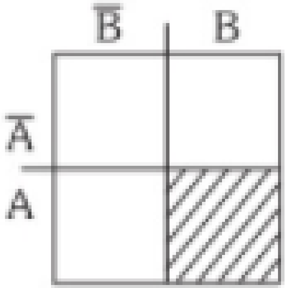
Esta região também pode ser chamada de região $\overline{A}\overline{B}$.



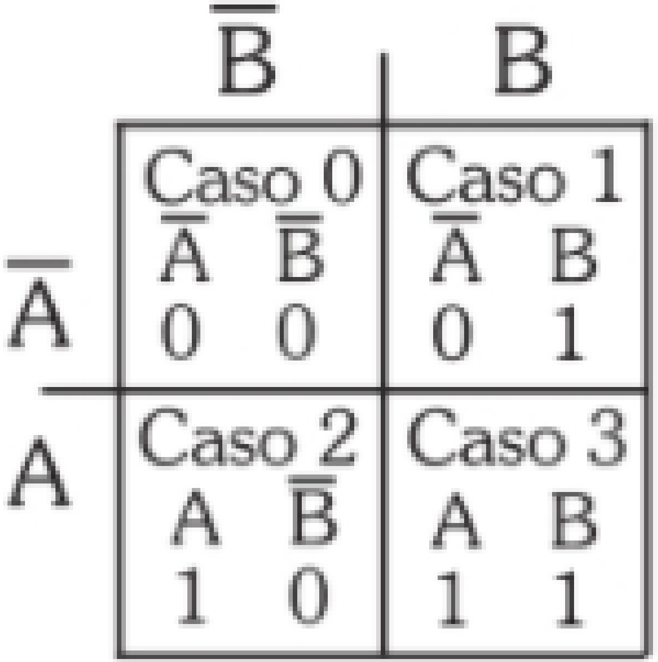
Esta região também pode ser chamada de região $\overline{A}B$.



Esta região também pode ser chamada de região $A\overline{B}$.



Esta região também pode ser chamada de região AB.



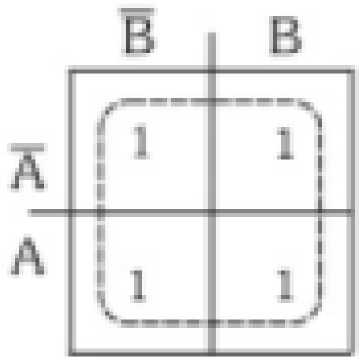
Mapa de Karnaugh (Simplificação de Expressão Booleana)

A	B	S	
0	0	0	caso 0
0	1	1	caso 1
1	0	1	caso 2
1	1	1	caso 3

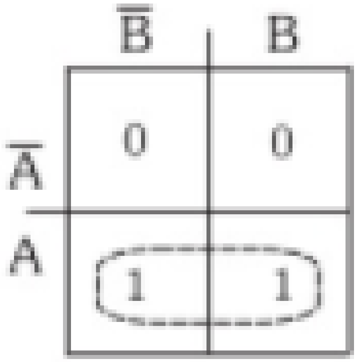
		0	1
		$\overline{B}$	B
0	$\overline{A}$	0	1
1	A	1	1

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

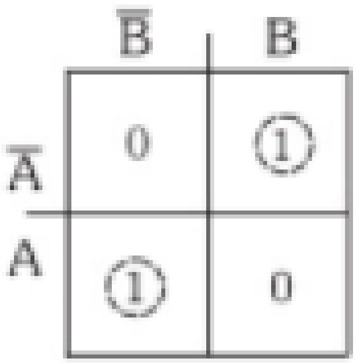
Mapa de Karnaugh (Simplificação de Expressão Booleana)



⇐ Quadra : S=1

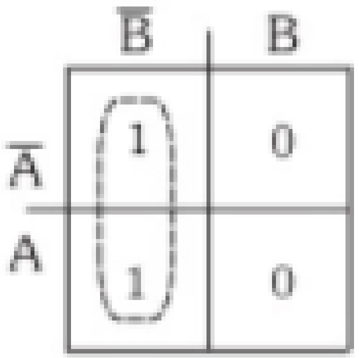


⇐ Par A (está exclusivamente na região A)



⇐ Termo $\overline{A}B$

⇐ Termo $A\overline{B}$



⇐ Par $\overline{B}$ (está exclusivamente na região $\overline{B}$)

Mapa de Karnaugh (Simplificação de Expressão Booleana)

	$\bar{B}$	B
$\bar{A}$	1	1
A	1	1

$\Leftarrow$ Quadra : S=1

	$\bar{B}$	B
$\bar{A}$	0	0
A	1	1

$\Leftarrow$ Par A (está exclusivamente na região A)

	$\bar{B}$	B
$\bar{A}$	0	1
A	1	0

$\Leftarrow$ Termo $\bar{A}B$

$\Leftarrow$ Termo $A\bar{B}$

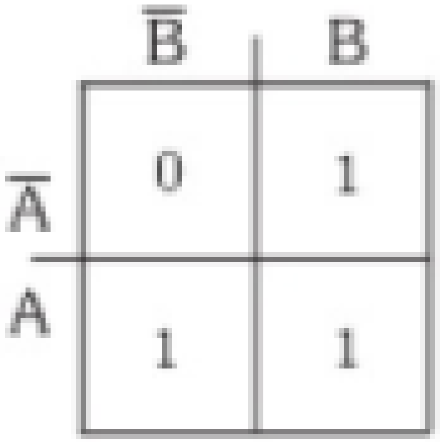
	$\bar{B}$	B
$\bar{A}$	1	0
A	1	0

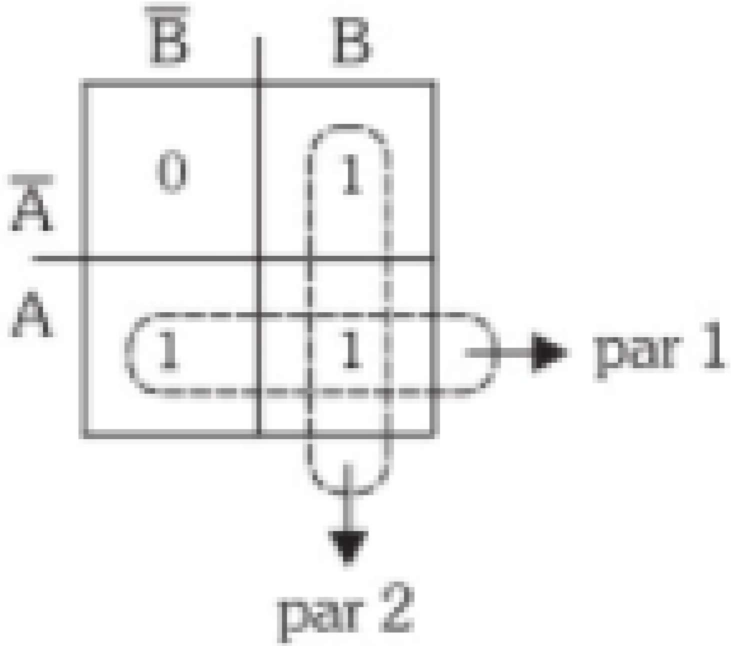
$\Leftarrow$ Par $\bar{B}$ (está exclusivamente na região $\bar{B}$)

Mapa de Karnaugh (Simplificação de Expressão Booleana)

A	B	S	
0	0	0	caso 0
0	1	1	caso 1
1	0	1	caso 2
1	1	1	caso 3

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





S = par 1 + par 2

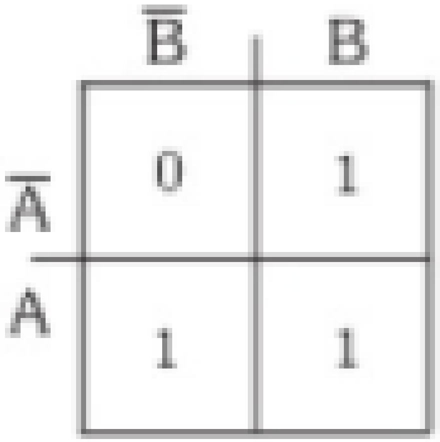
Porta OU

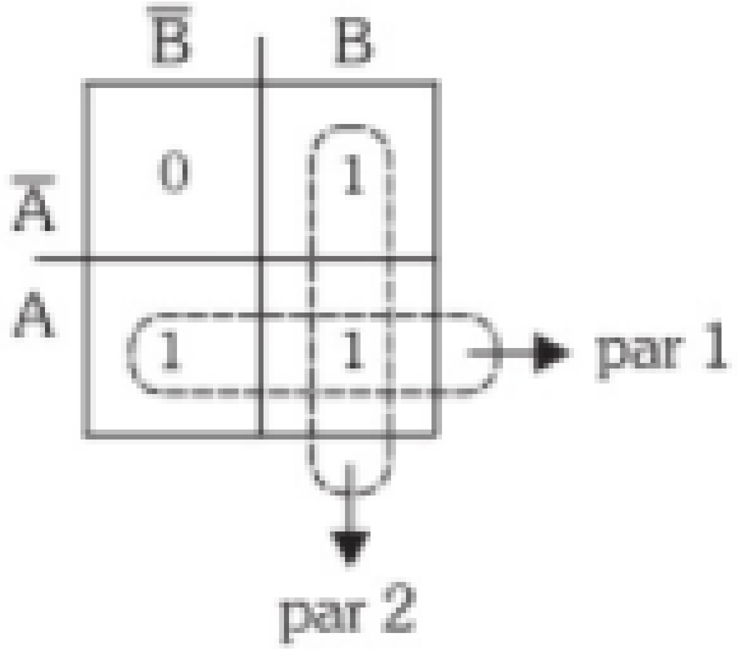
Mapa de Karnaugh (Simplificação de Expressão Booleana)

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

→ caso 0
→ caso 1
→ caso 2
→ caso 3

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





$$S = (A.\overline{B} + A.B) + (B.\overline{A} + B.A)$$

↓

Porta OU

Mapa de Karnaugh (Simplificação de Expressão Booleana)

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

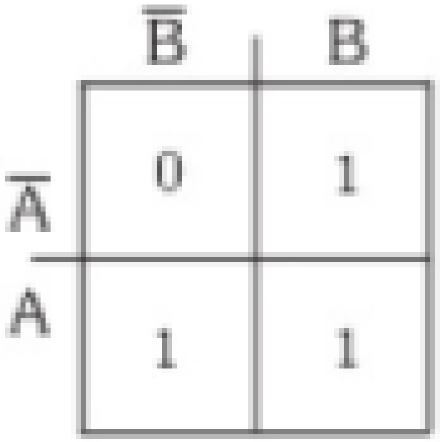
caso 0

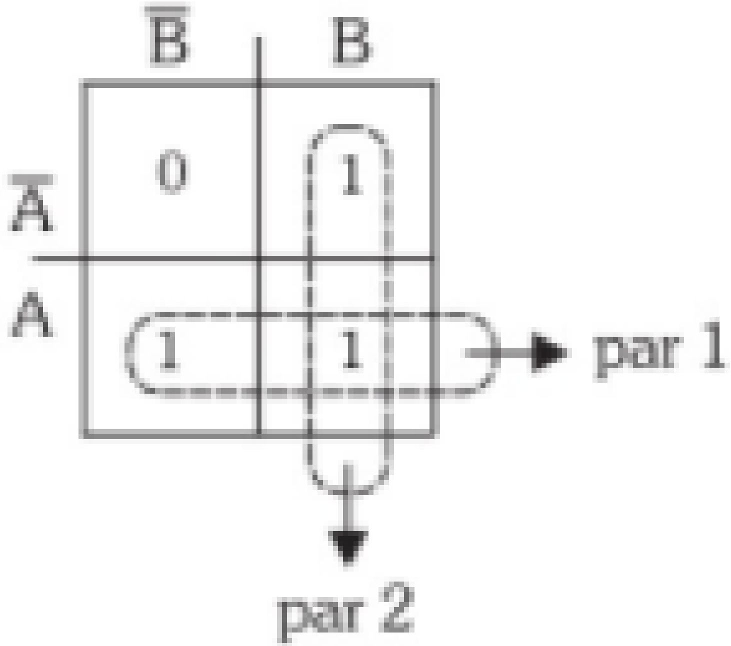
caso 1

caso 2

caso 3

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





$$S = (A.\overline{B} + A.B) + (B.\overline{A} + B.A)$$

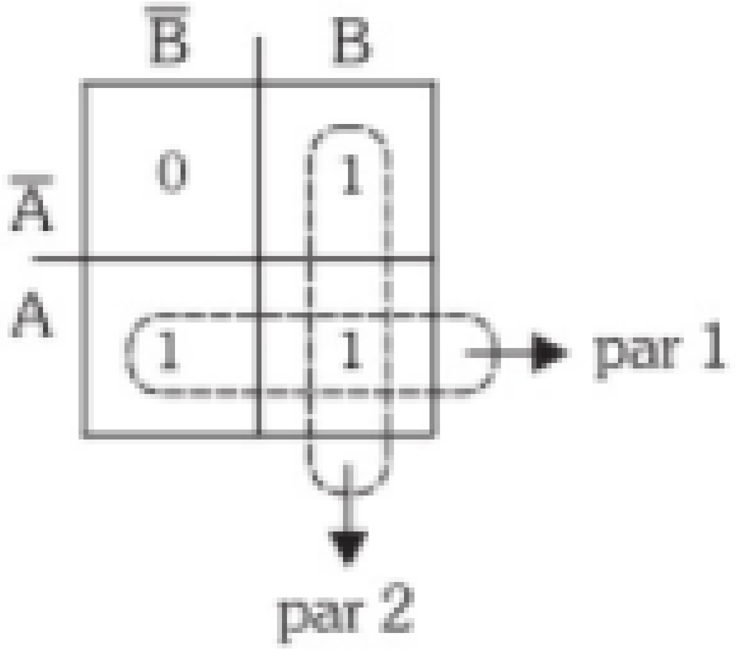
Porta OU

Mapa de Karnaugh (Simplificação de Expressão Booleana)

A	B	S	
0	0	0	→ caso 0
0	1	1	→ caso 1
1	0	1	→ caso 2
1	1	1	→ caso 3

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

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



$S = \mathbf{A} + \mathbf{B}$

↓

Porta OU

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Extrair a expressão booleana da tabela verdade e simplificar via Mapa de Karnaugh

Tabela verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

	$\bar{B}$	B
$\bar{A}$		
A		

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Extrair a expressão booleana da tabela verdade e simplificar via Mapa de Karnaugh

Tabela verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

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

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Extrair a expressão booleana da tabela verdade e simplificar via Mapa de Karnaugh

Tabela verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

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

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Extrair a expressão booleana da tabela verdade e simplificar via Mapa de Karnaugh

Tabela verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

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

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Extrair a expressão booleana da tabela verdade e simplificar via Mapa de Karnaugh

Tabela verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

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

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Extrair a expressão booleana da tabela verdade e simplificar via Mapa de Karnaugh

Tabela verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

	$\bar{B}$	B
$\bar{A}$	1	1
A	1	0

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Extrair a expressão booleana da tabela verdade e simplificar via Mapa de Karnaugh

Tabela verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

	$\bar{B}$	B
$\bar{A}$	1	1
A	1	0

Par 1 = $\bar{A} . \bar{B} + \bar{A} . B$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Extrair a expressão booleana da tabela verdade e simplificar via Mapa de Karnaugh

Tabela verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

	$\bar{B}$	B
$\bar{A}$	1	1
A	1	0

$$\text{Par 1} = \bar{A} . \bar{B} + \bar{A} . B$$

$$\text{Par 1} = \bar{A} . (\bar{B} + B)$$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Extrair a expressão booleana da tabela verdade e simplificar via Mapa de Karnaugh

Tabela verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

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

$$\text{Par } 1 = \overline{A} \,.\, \overline{B} + \overline{A} \,.\, B$$

$$\text{Par } 1 = \overline{A} \,.\, (\overline{B} + B)$$

$$\text{Par } 1 = \overline{A}$$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Extrair a expressão booleana da tabela verdade e simplificar via Mapa de Karnaugh

Tabela verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

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

$$\text{Par 1} = \overline{A} \cdot \overline{B} + \overline{A} \cdot B$$

$$\text{Par 1} = \overline{A} \cdot (\overline{B} + B)$$

$$\text{Par 1} = \overline{A}$$

Se o B varia, então, fica só o $\overline{A}$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Extrair a expressão booleana da tabela verdade e simplificar via Mapa de Karnaugh

Tabela verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

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

Par 1 = $\overline{A}$

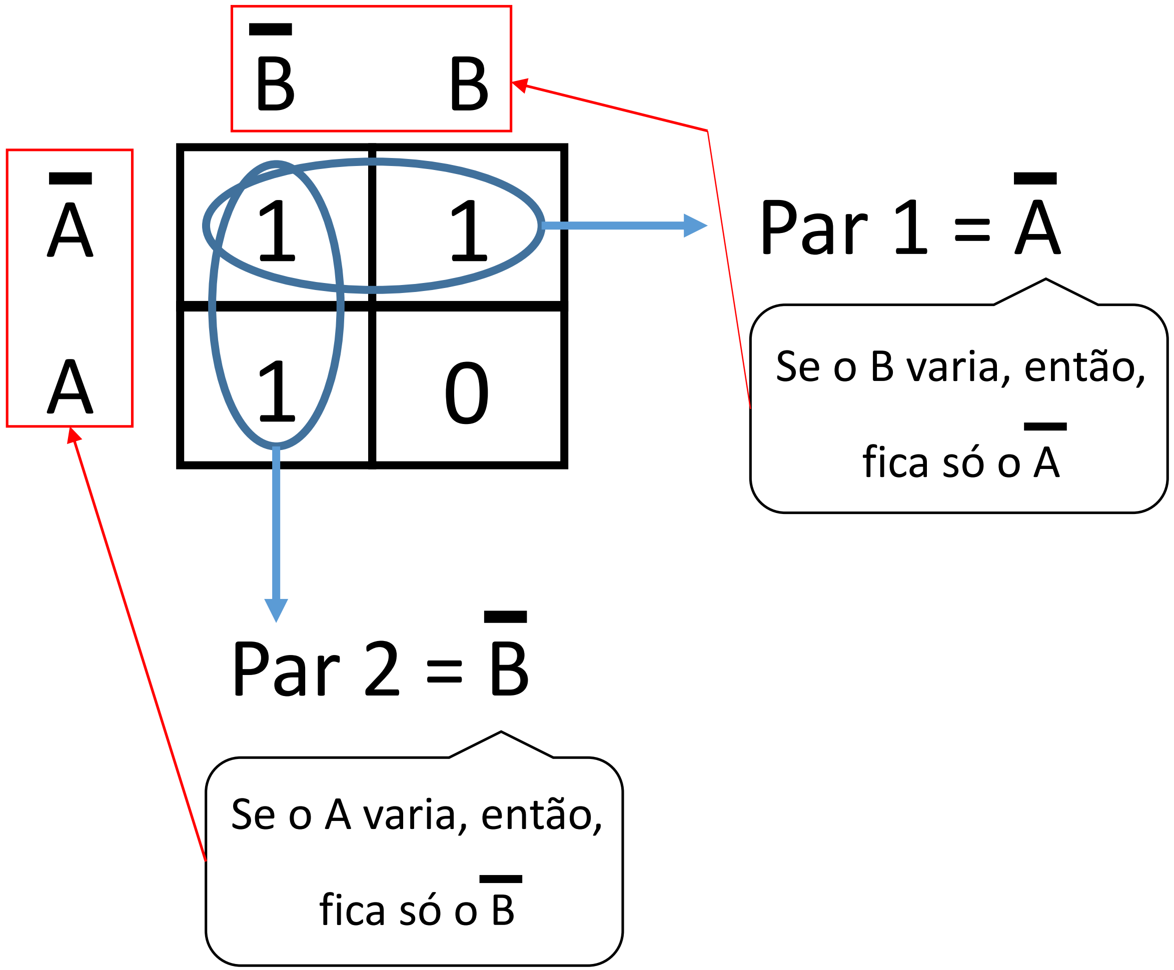
Se o B varia, então, fica só o $\overline{A}$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Extrair a expressão booleana da tabela verdade e simplificar via Mapa de Karnaugh

Tabela verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



Mapa de Karnaugh (Simplificação de Expressão Booleana)

Extrair a expressão booleana da tabela verdade e simplificar via Mapa de Karnaugh

Tabela verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

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

Par 1 = $\overline{A}$

Se o B varia, então, fica só o $\overline{A}$

Par 2 = $\overline{B}$

Se o A varia, então, fica só o $\overline{B}$

$$S = \text{Par } 1 + \text{Par } 2$$

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

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Simplificar expressão

$$X = \overline{A}B\overline{C} + AB\overline{C}$$
$$= B\overline{C}(\overline{A} + A)$$
$$= B\overline{C}(1) = B\overline{C}$$

Montar Mapa de Karnaugh

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Simplificar expressão

$$X = \overline{A}B\overline{C} + AB\overline{C}$$
$$= B\overline{C}(\overline{A} + A)$$
$$= B\overline{C}(1) = B\overline{C}$$

Montar Mapa de Karnaugh

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

A é o único que possui variação. Pode assumir 0 ou 1.
$\overline{A} + A = 1$

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

Mapa de Karnaugh (Tabela Verdade p/ Expressão Booleana)

A	B	C	X	
0	0	0	1	$\rightarrow \overline{A}\,\overline{B}\,\overline{C}$
0	0	1	1	$\rightarrow \overline{A}\,\overline{B}C$
0	1	0	1	$\rightarrow \overline{A}B\overline{C}$
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	$\rightarrow AB\overline{C}$
1	1	1	0	

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

$$\left\{X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + AB\overline{C}\right\}$$

Mapa de Karnaugh (Tabela Verdade p/ Expressão Booleana)

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

$\rightarrow \overline{A}\,\overline{B}\,\overline{C}$

$\rightarrow \overline{A}\,\overline{B}C$

$\rightarrow \overline{A}B\overline{C}$

$\rightarrow AB\overline{C}$

$$\{X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + AB\overline{C}\}$$

Isola o $\overline{\mathbf{A}}\overline{\mathbf{B}}$

Corta o **C**

Fica o $\overline{\mathbf{A}}\overline{\mathbf{B}}$

+

Isola o $\overline{\mathbf{C}}$ ou $\mathbf{B}\overline{\mathbf{C}}$

$\overline{C}$ ($\overline{\mathbf{A}}\mathbf{B}$ **+ AB)**

Corta o **A**, Fica **B**

Fica o $\mathbf{B}\overline{\mathbf{C}}$

Mapa de Karnaugh (Tabela Verdade p/ Expressão Booleana)

A	B	C	X	
0	0	0	1	→ $\bar{A}\bar{B}\bar{C}$
0	0	1	1	→ $\bar{A}\bar{B}C$
0	1	0	1	→ $\bar{A}B\bar{C}$
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	→ $AB\bar{C}$
1	1	1	0	

$$\{X = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + AB\bar{C}\}$$

Isola o $\bar{A}\bar{B}$

Corta o **C**

Fica o $\bar{A}\bar{B}$

+

Isola o $\bar{C}$ ou $B\bar{C}$

$\bar{C}(\bar{A}B + AB)$

Corta o **A**, Fica **B**

Fica o $B\bar{C}$

		0	1
		$\bar{C}$	C
0 0	$\bar{A}\bar{B}$	1	1
0 1	$\bar{A}B$	1	0
1 1	AB	1	0
1 0	$A\bar{B}$	0	0

Par 1 = ~A~B = $\bar{A} \cdot \bar{B}$

Par 2 = B~C = $B \cdot \bar{C}$

$$X = \bar{A} \cdot \bar{B} + B \cdot \bar{C}$$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	0	1	1
$\overline{A}B$	0	0	0	0
AB	0	0	0	0
$A\overline{B}$	1	0	0	1

$\longrightarrow \overline{A}\,\overline{B}C$

$\longrightarrow A\overline{B}\,\overline{D}$

$$X = \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}C\overline{D}$$
$$= \overline{A}\,\overline{B}C + A\overline{B}\,\overline{D}$$

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	0	1	0
$\overline{A}B$	1	1	1	1
AB	1	1	0	0
$A\overline{B}$	0	0	0	0

$\longrightarrow \overline{A}CD$

$\longrightarrow \overline{A}B$

$\longrightarrow B\overline{C}$

$$X = \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}\,\overline{D} + \overline{A}B\overline{C}D + \overline{A}BCD + \overline{A}BC\overline{D} + AB\overline{C}\,\overline{D} + AB\overline{C}D$$
$$= \overline{A}CD + \overline{A}B + B\overline{C}$$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Exemplos com 4 termos

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	1	0	0
$\overline{A}B$	0	1	1	1
AB	0	0	0	1
$A\overline{B}$	1	1	0	1

$\rightarrow \overline{B}\,\overline{C}D$

$\rightarrow BC\overline{D}$

$\rightarrow \overline{A}BD$

$\rightarrow A\overline{B}\,\overline{D}$

$$X = \overline{B}\,\overline{C}D + BC\overline{D} + \overline{A}BD + A\overline{B}\,\overline{D}$$

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	0	0	0
$\overline{A}B$	0	0	0	0
AB	1	1	1	1
$A\overline{B}$	0	0	0	0

$$X = AB$$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Exemplos com 5 termos

Posições equidistantes a linha central são vizinhas

	$\overline{C}\,\overline{D}\,\overline{E}$	$\overline{C}\,\overline{D}E$	$\overline{C}DE$	$\overline{C}D\overline{E}$	$CD\overline{E}$	CDE	$C\overline{D}E$	$C\overline{D}\,\overline{E}$
$\overline{A}\,\overline{B}$	1	1	0	0	1	1	1	X
$\overline{A}B$	0	X	X	1	1	X	X	X
AB	0	0	0	X	X	0	0	X
$A\overline{B}$	1	0	1	0	0	1	1	X

$$X = A\overline{B}DE + \overline{A}\,\overline{B}\,\overline{D} + \overline{B}CE + \overline{B}\,\overline{D}\,\overline{E} + \overline{A}C + \overline{A}BD$$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Exemplos com 5 termos

Posições equidistantes a linha central são vizinhas

	$\overline{C}\,\overline{D}\,\overline{E}$	$\overline{C}\,\overline{D}E$	$\overline{C}DE$	$\overline{C}D\overline{E}$	$CD\overline{E}$	CDE	$C\overline{D}E$	$C\overline{D}\,\overline{E}$
$\overline{A}\,\overline{B}$	1	1	0	0	1	1	1	X
$\overline{A}B$	0	X	X	1	1	X	X	X
AB	0	0	0	X	X	0	0	X
$A\overline{B}$	1	0	1	0	0	1	1	X

$$X = A\overline{B}DE + \overline{A}\,\overline{B}\,\overline{D} + \overline{B}CE + \overline{B}\,\overline{D}\,\overline{E} + \overline{A}C + \boxed{\overline{A}BD}$$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Exemplos com 5 termos

Posições equidistantes a linha central são vizinhas

	$\overline{C}\,\overline{D}\,\overline{E}$	$\overline{C}\,\overline{D}E$	$\overline{C}DE$	$\overline{C}D\overline{E}$	$CD\overline{E}$	CDE	$C\overline{D}E$	$C\overline{D}\,\overline{E}$
$\overline{A}\,\overline{B}$	1	1	0	0	1	1	1	X
$\overline{A}B$	0	X	X	1	1	X	X	X
AB	0	0	0	X	X	0	0	X
$A\overline{B}$	1	0	1	0	0	1	1	X

$$X = A\overline{B}DE + \overline{A}\,\overline{B}\,\overline{D} + \overline{B}CE + \overline{B}\,\overline{D}\,\overline{E} + \boxed{\overline{A}C} + \boxed{\overline{A}BD}$$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Exemplos com 5 termos

Posições equidistantes a linha central são vizinhas

	$\bar{C}\bar{D}\bar{E}$	$\bar{C}\bar{D}E$	$\bar{C}DE$	$\bar{C}D\bar{E}$	$CD\bar{E}$	CDE	$C\bar{D}E$	$C\bar{D}\bar{E}$
$\bar{A}\bar{B}$	1	1	0	0	1	1	1	X
$\bar{A}B$	0	X	X	1	1	X	X	X
AB	0	0	0	X	X	0	0	X
$A\bar{B}$	1	0	1	0	0	1	1	X

$$X = A\bar{B}DE + \bar{A}\bar{B}\bar{D} + \boxed{\bar{B}CE} + \bar{B}\bar{D}\bar{E} + \boxed{\bar{A}C} + \boxed{\bar{A}BD}$$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Exemplos com 5 termos

Posições equidistantes a linha central são vizinhas

	$\overline{C}\,\overline{D}\,\overline{E}$	$\overline{C}\,\overline{D}E$	$\overline{C}DE$	$\overline{C}D\overline{E}$	$CD\overline{E}$	CDE	$C\overline{D}E$	$C\overline{D}\,\overline{E}$
$\overline{A}\,\overline{B}$	1	1	0	0	1	1	1	X
$\overline{A}B$	0	X	X	1	1	X	X	X
AB	0	0	0	X	X	0	0	X
$A\overline{B}$	1	0	1	0	0	1	1	X

$$X = A\overline{B}DE + \boxed{\overline{A}\,\overline{B}\,\overline{D}} + \boxed{\overline{B}CE} + \overline{B}\,\overline{D}\,\overline{E} + \boxed{\overline{A}C} + \boxed{\overline{A}BD}$$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Exemplos com 5 termos

Posições equidistantes a linha central são vizinhas

	$\overline{C}\,\overline{D}\,\overline{E}$	$\overline{C}\,\overline{D}E$	$\overline{C}DE$	$\overline{C}D\overline{E}$	$CD\overline{E}$	CDE	$C\overline{D}E$	$C\overline{D}\,\overline{E}$
$\overline{A}\,\overline{B}$	1	1	0	0	1	1	1	X
$\overline{A}B$	0	X	X	1	1	X	X	X
AB	0	0	0	X	X	0	0	X
$A\overline{B}$	1	0	1	0	0	1	1	X

$$X = \boxed{A\overline{B}DE} + \boxed{\overline{A}\,\overline{B}\,\overline{D}} + \boxed{\overline{B}CE} + \overline{B}\,\overline{D}\,\overline{E} + \boxed{\overline{A}C} + \boxed{\overline{A}BD}$$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Exemplos com 5 termos

Posições equidistantes a linha central são vizinhas

	$\bar{C}\bar{D}\bar{E}$	$\bar{C}\bar{D}E$	$\bar{C}DE$	$\bar{C}D\bar{E}$	$CD\bar{E}$	CDE	$C\bar{D}E$	$C\bar{D}\bar{E}$
$\bar{A}\bar{B}$	1	1	0	0	1	1	1	X
$\bar{A}B$	0	X	X	1	1	X	X	X
AB	0	0	0	X	X	0	0	X
$A\bar{B}$	1	0	1	0	0	1	1	X

$$X = A\bar{B}DE + \bar{A}\bar{B}\bar{D} + \bar{B}CE + \bar{B}\bar{D}\bar{E} + \bar{A}C + \bar{A}BD$$

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Exercícios

29)

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	0	0	0
$\overline{A}B$	1	1	1	1
AB	1	1	1	1
$A\overline{B}$	0	0	0	0

30)

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	1	1	1	0
$\overline{A}B$	1	1	1	1
AB	1	1	1	1
$A\overline{B}$	1	1	1	1

Mapa de Karnaugh (Simplificação de Expressão Booleana)

Exercícios

31)

	$\overline{C}\,\overline{D}\,\overline{E}$	$\overline{C}\,\overline{D}E$	$\overline{C}DE$	$\overline{C}D\overline{E}$	$CD\overline{E}$	CDE	$C\overline{D}E$	$C\overline{D}\,\overline{E}$
$\overline{A}\,\overline{B}$	1	0	0	1	X	X	0	X
$\overline{A}B$	1	0	1	0	0	1	0	X
AB	1	0	1	0	0	1	0	X
$A\overline{B}$	1	0	1	X	X	1	0	1