1. Equivalência entre portas
2. Derivação de expressões booleanas
3. Simplificação de expressões booleanas
4. Circuitos combinacionais

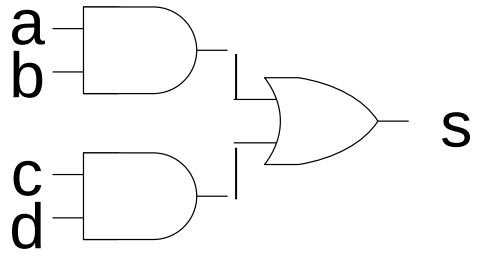
Outras equivalências

Expressão ou porta	Expressões equivalentes
a’	(a.a)’ a NAND a (a + a)’ a NOR a a XOR 1 a XNOR 0
a . b	(a’ + b’)’ a’ NOR b’
a + b	(a’ . b’)’ a’ NAND b’
a XOR b	a.b’ + a’.b
a XNOR b	a.b + a’.b’

Derivação de expressões booleanas a partir de tabelas de entradas e saídas

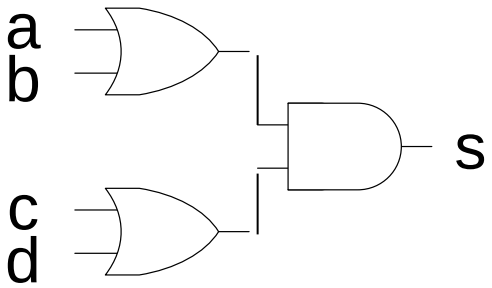
- soma de produtos

$s = a.b + c.d$



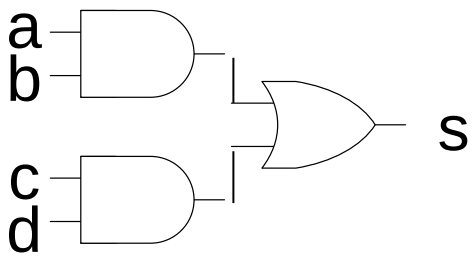
- produto de somas

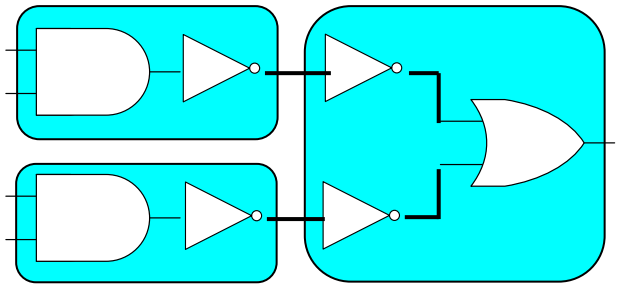
$s = (a+b).(c+d)$

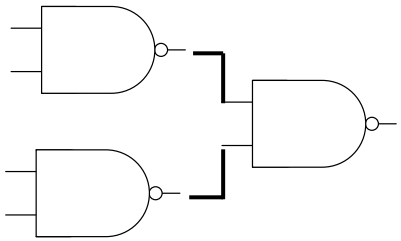


Redes equivalentes

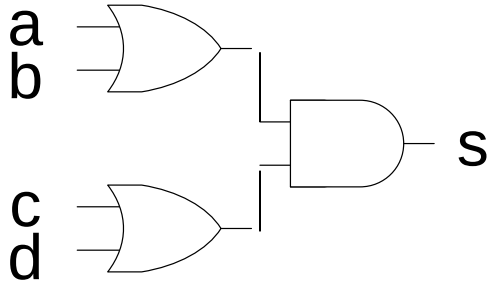
Soma de Produtos

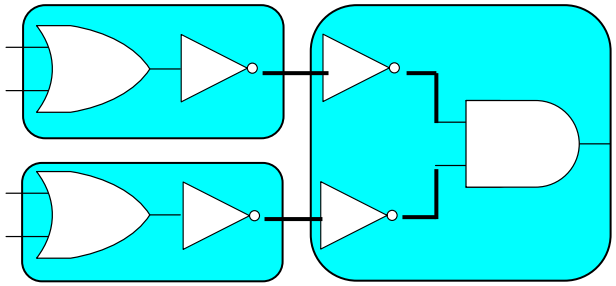


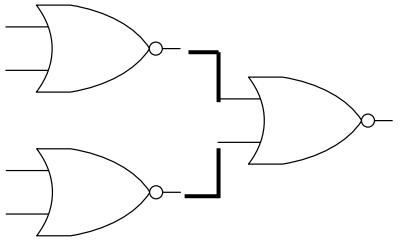




Produto de Somas







Derivação de soma de produtos

1. Construir a tabela-verdade com as entradas e saídas do circuito.

a	b	c	s
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Derivação de soma de produtos

2. Acrescentar uma coluna que contenha, para cada uma das linhas das possíveis combinações de entrada, um termo-produto formado pelo ‘e’ lógico de todas as variáveis de entrada. Se o valor da variável de entrada for igual a zero (naquela linha da tabela), ela aparece complementada no termo-produto. Se o valor da variável de entrada for igual a um, ela aparece na forma normal (sem ser complementada) no termo-produto.

a	b	c	s	termos-produto
0	0	0	1	a’.b’.c’
0	0	1	0	a’.b’.c
0	1	0	1	a’.b.c’
0	1	1	0	a’.b.c
1	0	0	1	a.b’.c’
1	0	1	0	a.b’.c
1	1	0	1	a.b.c’
1	1	1	0	a.b.c

Derivação de soma de produtos

3. Construir uma soma de produtos, na qual aparecem todos os termos-produto correspondentes a valores de saída iguais a 1.

a	b	c	s	termos-produto
0	0	0	1	a’.b’.c’
0	0	1	0	a’.b’.c
0	1	0	1	a’.b.c’
0	1	1	0	a’.b.c
1	0	0	1	a.b’.c’
1	0	1	0	a.b’.c
1	1	0	1	a.b.c’
1	1	1	0	a.b.c

$$s = a' \cdot b' \cdot c' + a' \cdot b \cdot c' + a \cdot b' \cdot c' + a \cdot b \cdot c'$$

4. Simplificar a expressão obtida, aplicando as propriedades da álgebra booleana.

$$s = c'$$

Derivação de produto de somas

1. Construir a tabela-verdade com as entradas e saídas do circuito.

a	b	c	s
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Derivação de produto de somas

2. Acrescentar uma coluna que contenha, para cada uma das linhas das possíveis combinações de entrada, um termo-soma formado pelo ‘ou’ lógico de todas as variáveis de entrada. Se o valor da variável de entrada for igual a um (naquela linha da tabela), ela aparece complementada no termo-soma. Se o valor da variável de entrada for igual a zero, ela aparece na forma normal (sem ser complementada) no termo-soma.

a	b	c	s	termos-soma
0	0	0	1	a+b+c
0	0	1	0	a+b+c’
0	1	0	1	a+b’+c
0	1	1	0	a+b’+c’
1	0	0	1	a’+b+c
1	0	1	0	a’+b+c’
1	1	0	1	a’+b’+c
1	1	1	0	a’+b’+c’

Derivação de produto de somas

3. Construir um produto de somas, no qual aparecem todos os termos-soma correspondentes a valores de saída iguais a 0.

a	b	c	s	termos-soma
0	0	0	1	a+b+c
0	0	1	0	a+b+c’
0	1	0	1	a+b’+c
0	1	1	0	a+b’+c’
1	0	0	1	a’+b+c
1	0	1	0	a’+b+c’
1	1	0	1	a’+b’+c
1	1	1	0	a’+b’+c’

$$s = (a + b + c') \cdot (a + b' + c') \cdot (a' + b + c') \cdot (a' + b' + c')$$

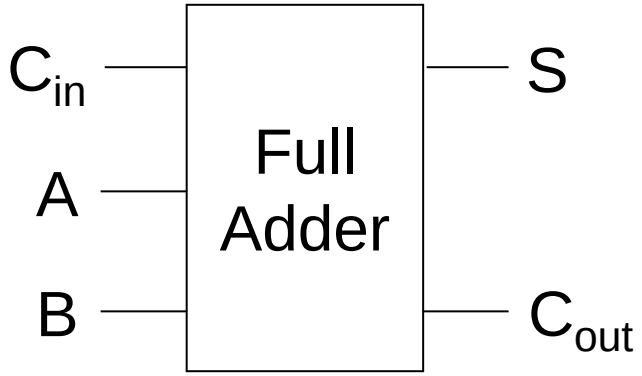
4. Simplificar a expressão obtida, aplicando as propriedades da álgebra booleana.

$$s = c'$$

Exercício

Derivar as expressões booleanas que representem o funcionamento das duas saídas (soma e vai um) de um somador binário completo (*full adder*) através de somas de produtos.

Não é necessário simplificar as expressões.

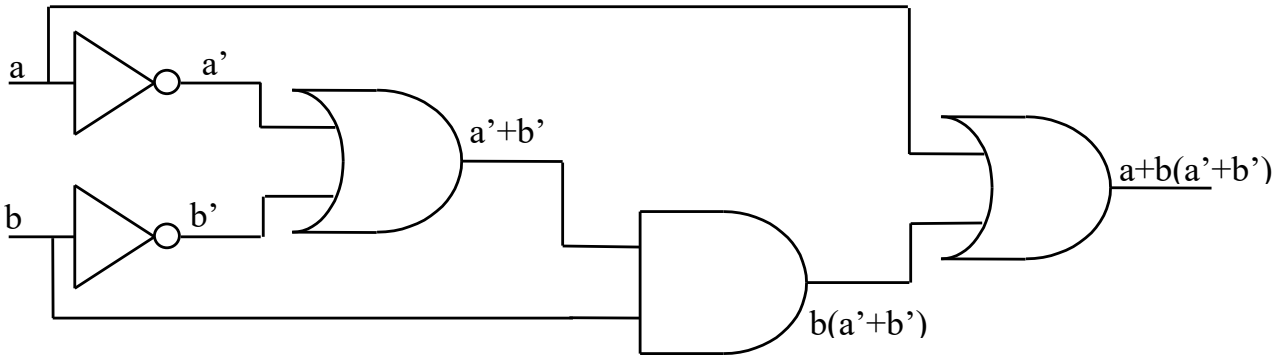


A	B	Cin	S	Cout	t. produto
0	0	0	0	0	A'.B'.Cin'
0	0	1	1	0	A'.B'.Cin
0	1	0	1	0	A'.B.Cin'
0	1	1	0	1	A'.B.Cin
1	0	0	1	0	A.B'.Cin'
1	0	1	0	1	A.B'.Cin
1	1	0	0	1	A.B.Cin'
1	1	1	1	1	A.B.Cin

S = A'.B'.Cin + A'.B.Cin +
+ A.B'.Cin + A.B.Cin

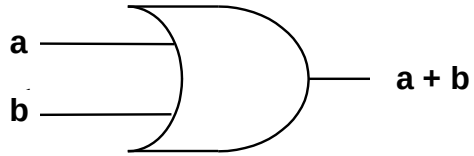
Cout = A'.B.Cin + A.B'.Cin +
+ A.B.Cin' + A.B.Cin

Simplificação de expressões booleanas



Como chegar à conclusão de que este circuito pode ser implementado com menos portas, ou com menos entradas, ou com menos conexões?

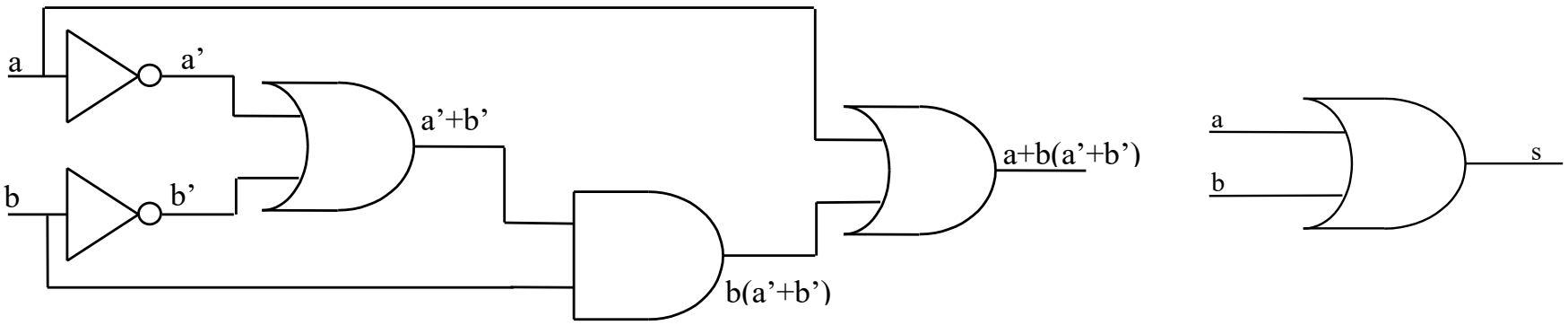
➲ custo (área), velocidade, consumo de potência ➲



Simplificação de expressões booleanas

- aplicando leis, propriedades e teoremas da álgebra booleana
- mapas de Karnaugh
- método de Quine-McCluskey
- outros métodos
- ferramentas (software) - exemplo: Karma (http://www.inf.ufrgs.br/lagarto/)

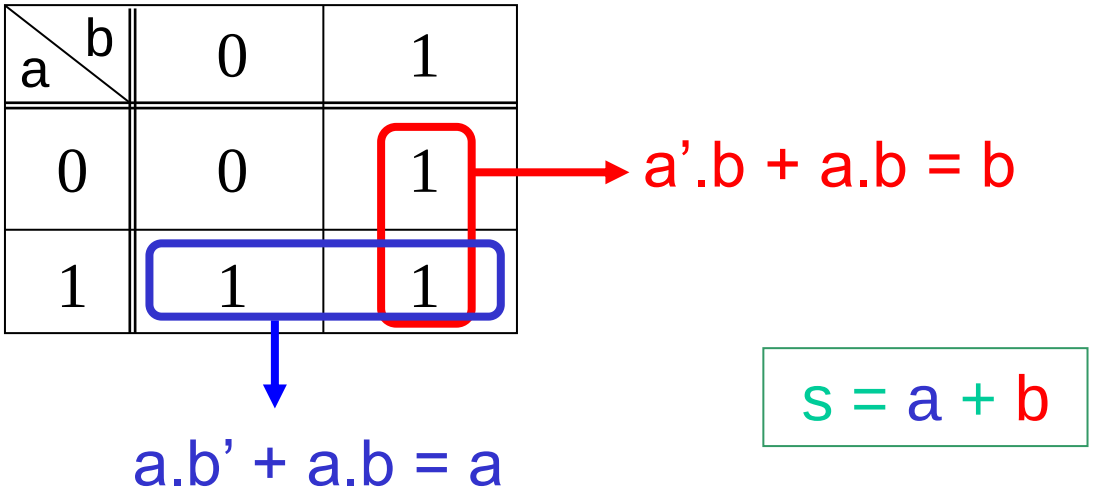
Aplicando as leis, propriedades e teoremas da álgebra booleana



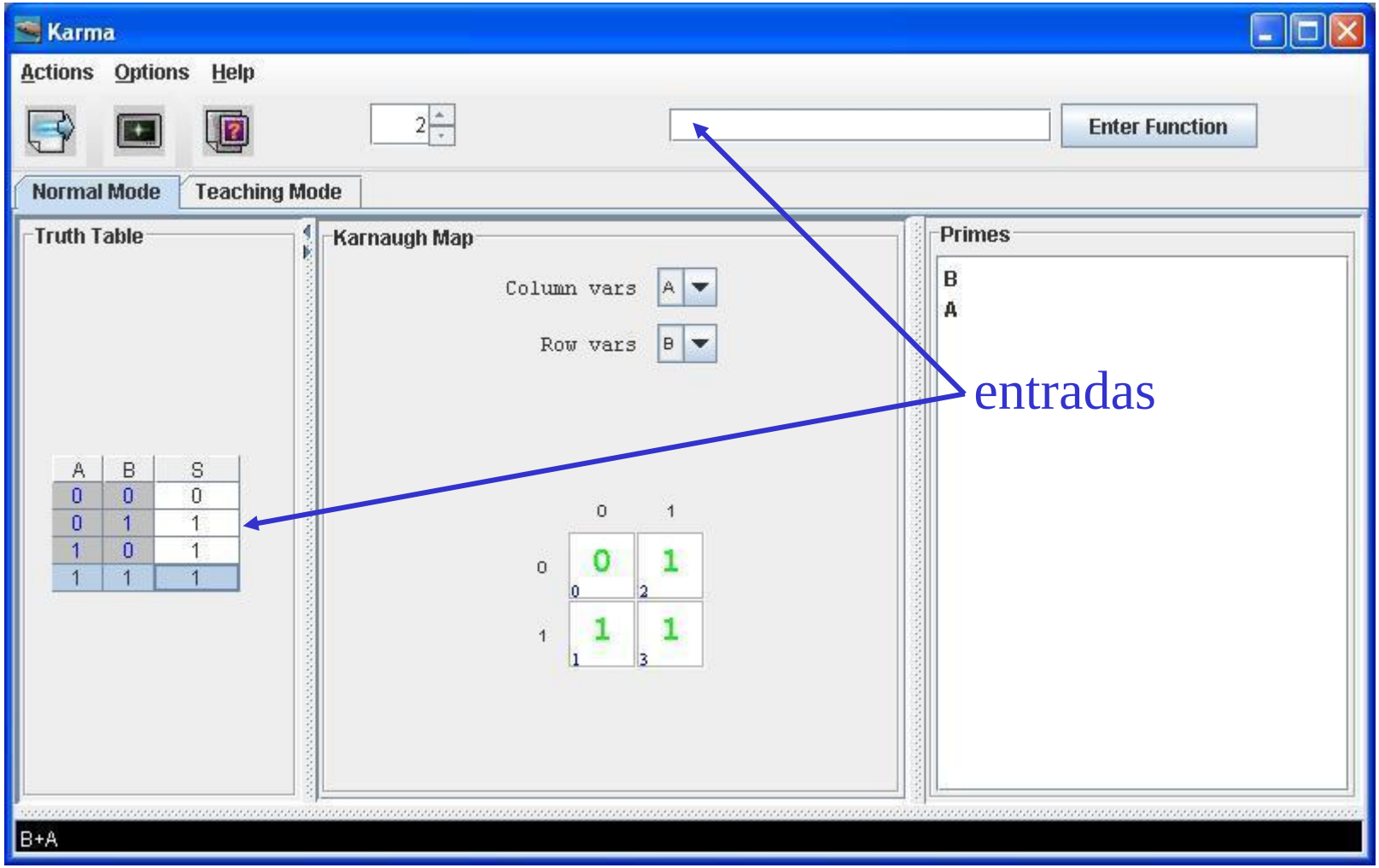
a + b.(a' + b')	=	(distributiva)
	= a + b.a'+ b.b'	(x . x' = 0)
	= a + b.a' + 0	(x + 0 = x)
	= a + b.a'	(distributiva)
	= (a + b).(a + a')	(x + x' = 1)
	= (a + b).1	(x . 1 = x)
	= a + b	

Usando mapas de Karnaugh

a	b	a’	b’	a+b	a’+b’	b.(a’+b’)	a+b.(a’+b’)
0	0	1	1	0	1	0	0
0	1	1	0	1	1	1	1
1	0	0	1	1	1	0	1
1	1	0	0	1	0	0	1



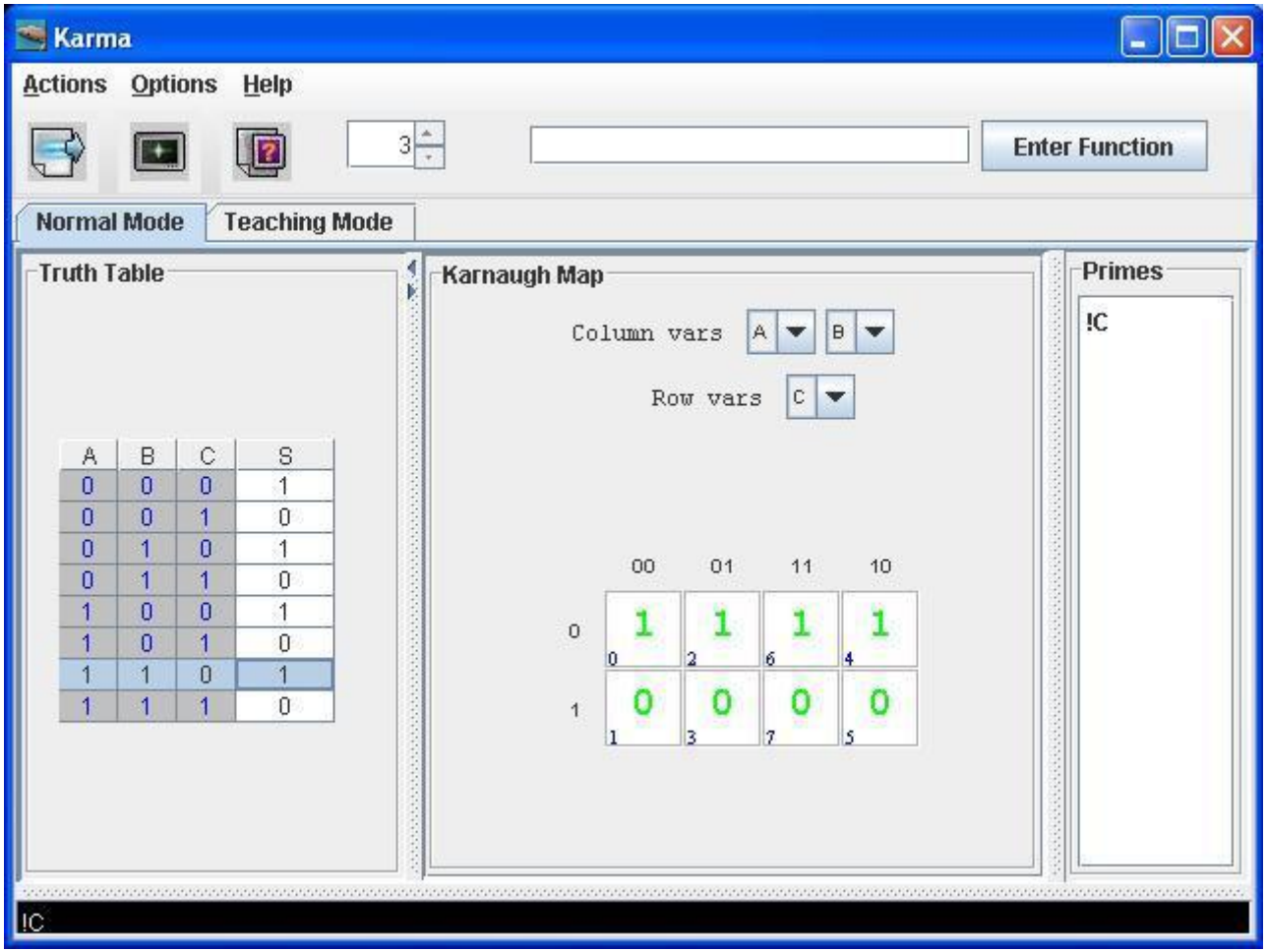
Usando uma ferramenta (software)



Obs: esta ferramenta (Karma) usa o método de Quine-McCluskey para chegar à expressão mais simples, mas também desenha o mapa de Karnaugh correspondente à tabela de entradas e saídas do circuito

Usando uma ferramenta (software)

s = a’ . b’ . c’ + a’ . b . c’ + a . b’ . c’ + a . b . c’ = c’



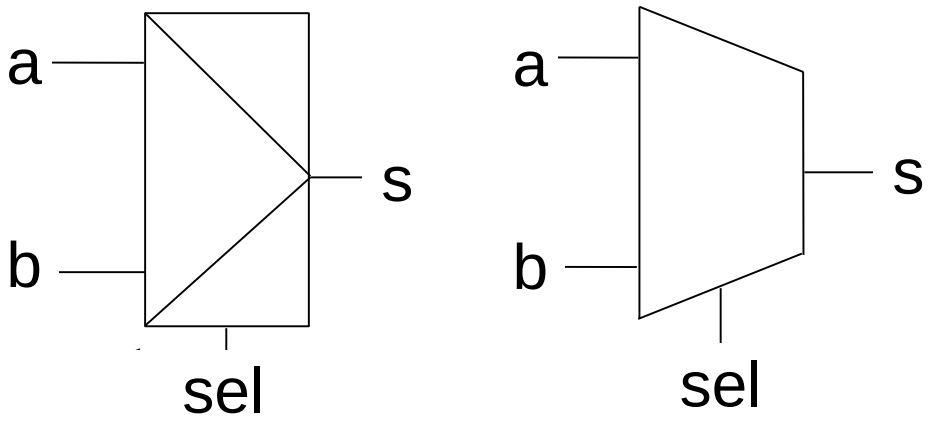
Circuitos Combinacionais

- saídas são função apenas das entradas
- são construídos apenas com portas lógicas sem realimentação
- não possuem elementos de armazenamento (memórias)
- exemplos:
  - multiplexador
  - decodificador
  - unidade aritmética e lógica

Multiplexador (ou Seletor)

- duas ou mais entradas (normalmente 2^n)
- somente uma saída
- um sinal de “seleção” define qual das entradas é copiada na saída
- para 2^n entradas são usados n bits de seleção

Símbolos usados para representar multiplexadores 2-para-1



Multiplexador (ou Seletor)

a	b	sel	saída	t-produto
0	0	0	0	a’.b’.sel’
0	0	1	0	a’.b’.sel
0	1	0	0	a’.b.sel’
0	1	1	1	a’.b.sel
1	0	0	1	a.b’.sel’
1	0	1	0	a.b’.sel
1	1	0	1	a.b.sel’
1	1	1	1	a.b.sel

saída

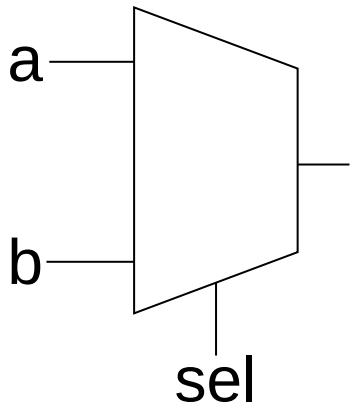
= a’.b.sel + a.b’.sel’ + a.b.sel’ + a.b.sel

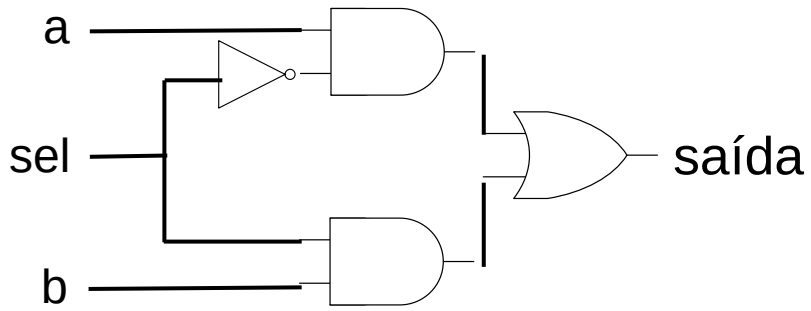
= (a’.b + a.b).sel + (a.b’ + a.b).sel’

= (b.(a’+a)).sel + (a.(b’+b)).sel’

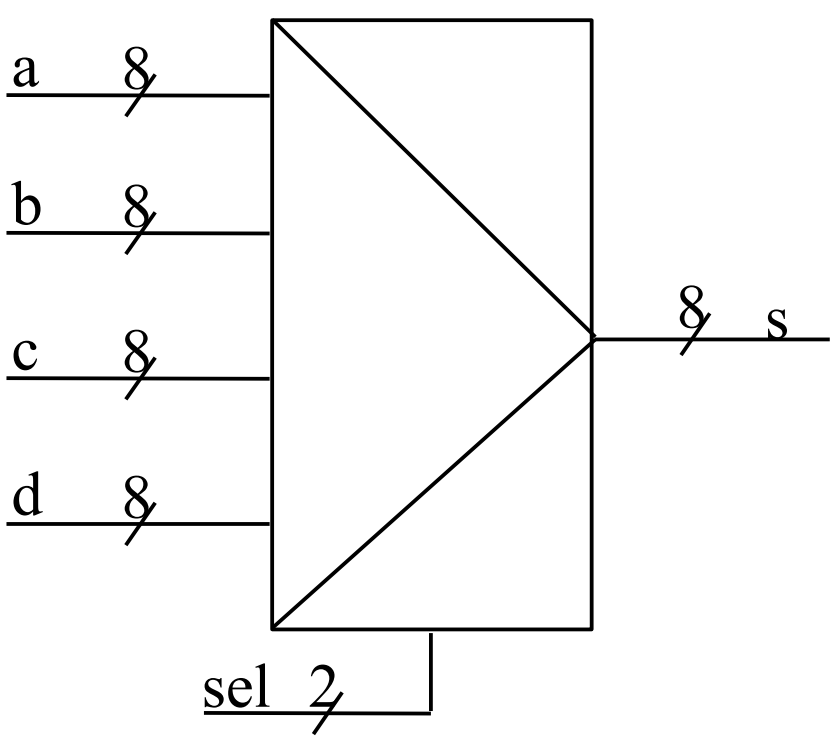
= (b.1).sel + (a.1).sel’

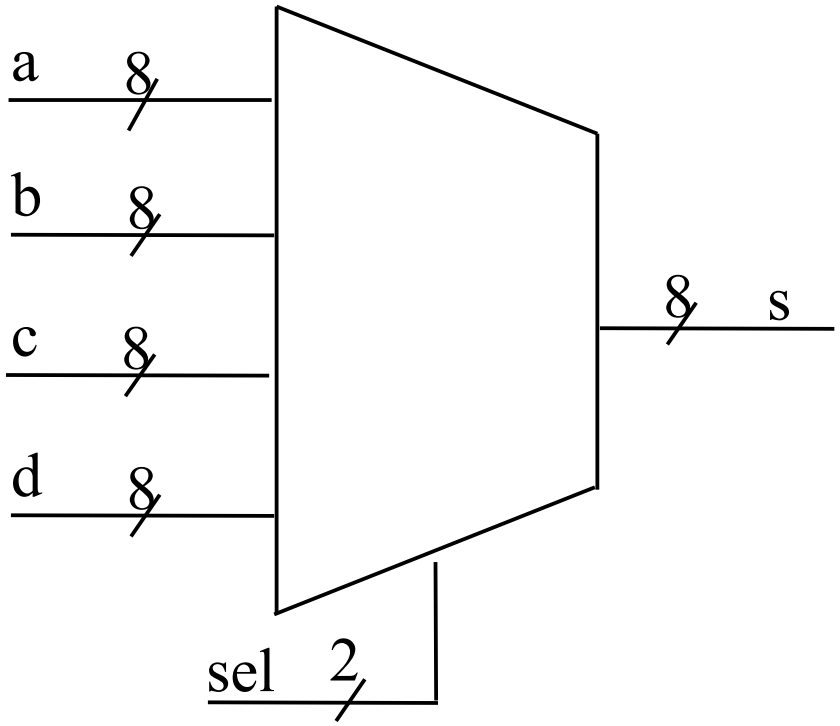
= a.sel’ + b.sel





Multiplexador 4-para-1 de 8 bits

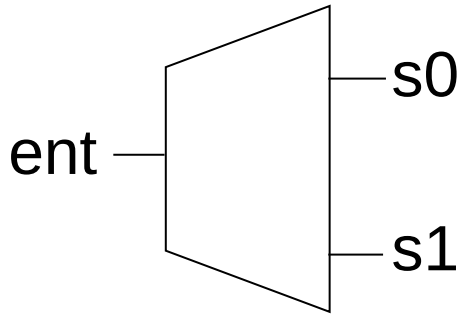


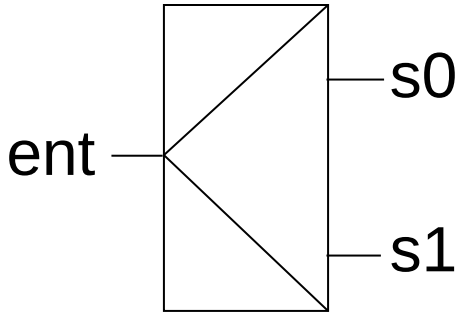


Decodificador

- entrada com n bits
- 2^n saídas (correspondem a valores de 0 a 2^{n-1} da entrada)
- somente a saída de índice igual ao valor binário representado pelas entrada fica “ativa” (igual a 1, por exemplo)
- todas as demais saídas ficam “desativadas” (iguais a zero, por exemplo)

Símbolos usados para representar decodificadores com entrada de 1 bit





Decodificador

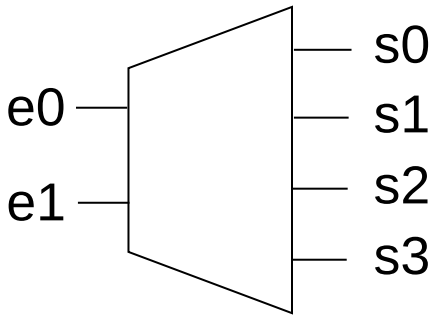
e0	e1	s0	s1	s2	s3	t-produto
0	0	1	0	0	0	e0’.e1’
0	1	0	1	0	0	e0’.e1
1	0	0	0	1	0	e0.e1’
1	1	0	0	0	1	e0.e1

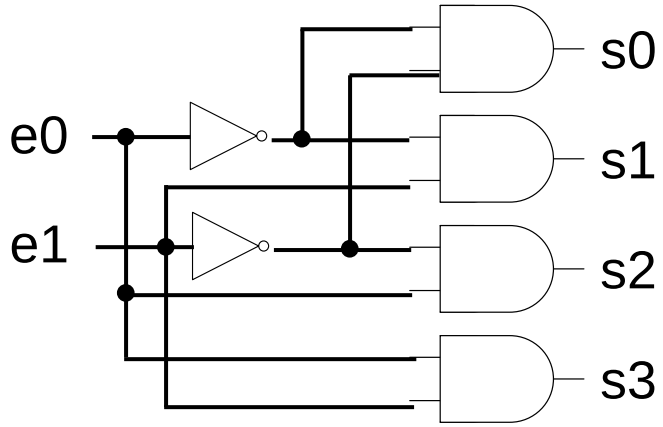
s0 = e0’.e1’

s1 = e0’.e1

s2 = e0.e1’

s3 = e0.e1





Circuitos Seqüenciais

- saídas são função tanto das entradas quanto dos valores de saída (estado atual)
- são construídos com portas lógicas com realimentação
- possuem elementos de armazenamento (memórias)
- exemplos:
  - flip-flop
  - registrador
  - contador