Circuitos Digitais

Aula 03 Circuito Combinatório

UNIP JUNDIAÍ

Conteúdo

- Introdução
- Portas Lógicas
- Álgebra Booleana
- Equivalência de Circuitos
- Circuitos Combinatórios

Digital x Analógico

- Nos Sistemas Digitais as variáveis são discretas.
- Já num Sistema Analógico, as variáveis são contínuas (sinais reais do mundo físico).

Digital x Analógico

- Um sinal analógico amplificado, convertido em digital usando um conversor analógico-digital (AD)

- Esse processo envolve amostragem e quantização.

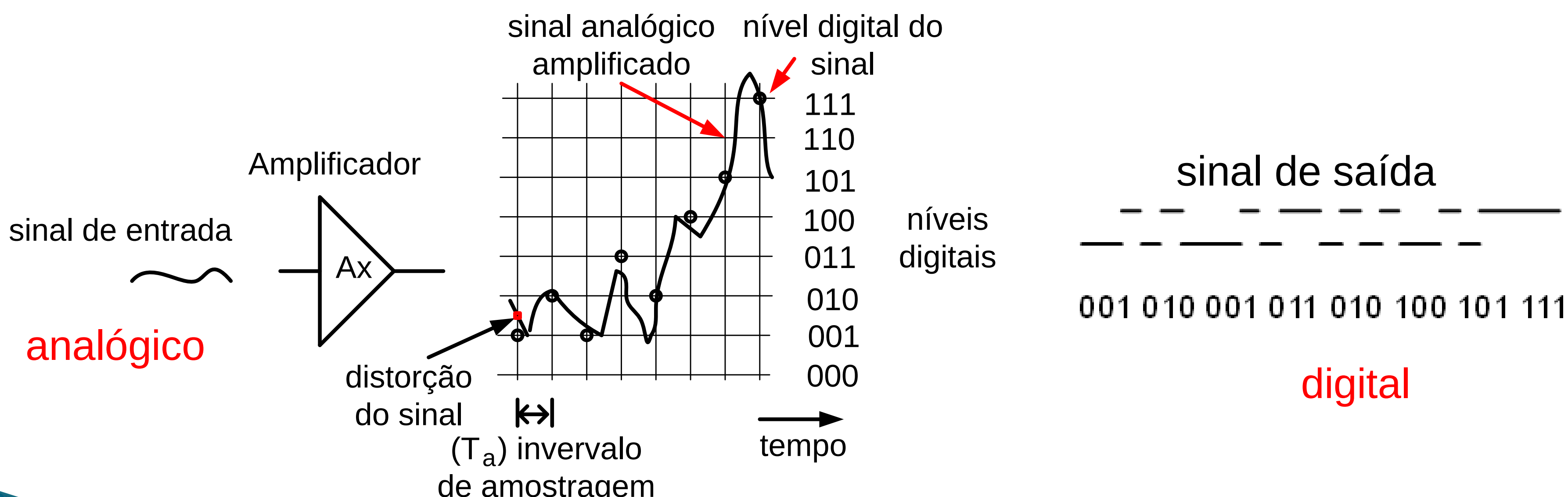


Ilustração do processo de amostragem e quantização.

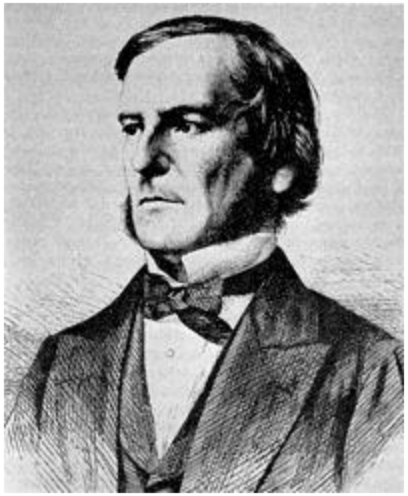
Os sinais discretos ou binários

- Os sinais discretos ou binários são compostos de elementos lógicos chamados bits, que podem assumir valores 0 e 1.
  - O valor 0 corresponde a *falso* em lógica; *desligado* em circuito de chaveamento; ou voltagem *baixa* em circuitos eletrônicos
  - O valor 1 corresponde a *verdadeiro, ligado* ou voltagem *alta,* respectivamente*.*

Álgebra Booleana

- A Álgebra Booleana é um tipo de álgebra usado para modelar operações lógicas, inventada por George Boole.

- A álgebra de chaveamento (ou comutação), *switching* em inglês, é uma classe de álgebras booleanas.
  - Portanto, os teoremas das álgebras booleanas são aplicáveis às álgebras de chaveamento.



George Boole
Matemático e
Filósofo britânico
(1815-1864)

Definição

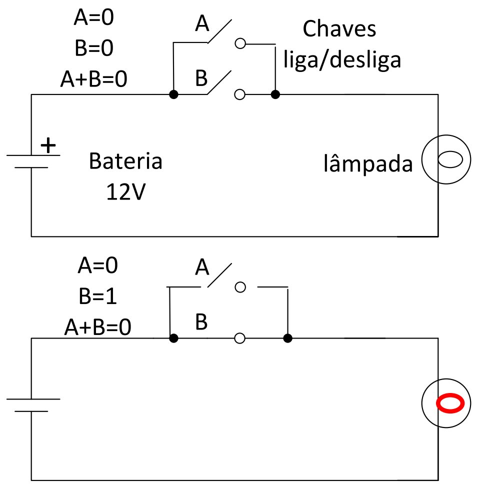
- A álgebra Booleana é composta por um conjunto de elementos, operações binárias OR e AND, e operação unária NOT, aplicadas sobre os elementos.
  - a operação OR é também representada pelo símbolo "+ " e a operação AND, pelo símbolo ". "
    - AND tem precedência sobre OR.

Operação OR

OR (+)	
0 + 0	0
0 + 1	1
1 + 0	1
1 + 1	1

Operação AND

AND (.)	
0 . 0	0
0 . 1	0
1 . 0	0
1 . 1	1

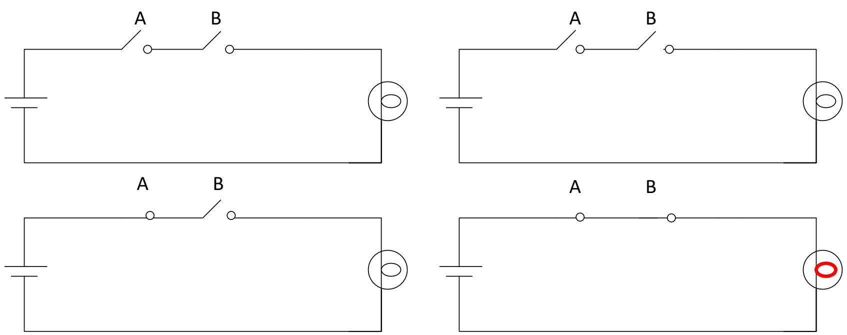


Operação
OR

OR (+)	
0 + 0	0
0 + 1	1
1 + 0	1
1 + 1	1

Chaves
0 – aberta
1 -fechada

Lâmpada
0 – apagada
1 - acesa



Operação
AND

AND (.)	
0 . 0	0
0 . 1	0
1 . 0	0
1 . 1	1

Operação de complemento e variáveis

- Além das operações AND e OR, tem a operação de complemento:

COMPLEMENTO	
0	1
1	0

Variáveis

- Os elementos na álgebra Booleana são representados por variáveis, ou seus complementos, que podem assumir os valores binários: 0 e 1.

- Exemplo de variáveis ou seus complementos: a, A, b, c, a, A (ou a’, A’)

- Conforme os valores das variáveis é computado o valor de uma expressão:
- Ex: $a + b = 0$, se $a = 0$ e $b = 0$, pois $0 + 0 = 0$.

Postulados

- P1: Comutatividade

$$a+b=b+a$$

$$a.b=b.a$$

- P2: Distributividade

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

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

- P3: Associatividade

$$a+(b+c)=(a+b)+c=a+b+c$$

$$a.(b.c)=(a.b).c=a.b.c$$

- P3: Existência de elemento de identidade em operações OR e AND

$$0+a=a+0=a$$

$$1.a=a.1=a$$

- P4: Existência de complemento

$$a+a'=1$$

$$a.a'=0$$

ou

$$a+\bar{a}=1$$

$$a.\bar{a}=0$$

Principais Propriedades

- Dualidade

  Numa equação, se trocarmos as operações AND e OR e os elementos de identidade 0 e 1, obtemos uma equação equivalente.

$$a + 0 = a \qquad a.1 = a$$

Isso significa que as operações AND e OR são duais; e os elementos 0 e 1 também são duais.

- Outras propriedades

$$a + 1 = 1$$

$$a.0 = 0$$

$$0' = 1 \quad \text{ou} \quad \overline{0} = 1$$

$$1' = 0 \qquad \overline{1} = 0$$

- Lei da idempotência

$$a + a = a$$

$$a.a = a$$

- Lei da involução
- $(a')' = a$ ou $\overline{\overline{a}} = a$
- Lei da absorção

$$a + a.b = a$$

$$a.(a + b) = a$$

- Simplificação

$$a + a'.b = a + b \qquad a + \overline{a}.b = a + b$$

$$a.(a' + b) = a.b \qquad a.(\overline{a} + b) = a.b$$

- Lei de DeMorgan
- para complementar uma expressão, complementa-se as variáveis e troca-se as operações AND por OR e vice-versa.

$$(a + b)' = a'.b' \qquad \overline{(a + b)} = \overline{a}.\overline{b}$$

$$(a.b)' = a' + b' \qquad \overline{(a.b)} = \overline{a} + \overline{b}$$

Resumo das principais propriedades

$$a + 0 = a \qquad a.1 = a$$

$$a + 1 = 1 \qquad a.0 = 0$$

$$a + a = a \qquad a.a = a$$

$$a + a' = 1 \qquad a.a' = 0$$

$$(a')' = a$$

$$a + a.b = a \qquad a.(a + b) = a$$

$$a + a'.b = a + b \qquad a.(a' + b) = a.b$$

$$(a + b)' = a'.b' \qquad (a.b)' = a' + b'$$

$$(a + b)(a + c) = a + bc$$

Exercício 1

- Aplicando as identidades ou propriedades da álgebra booleana simplificar a expressão:

$$\overline{x_2}\,\overline{x_1}\,x_0 + \overline{x_2}\,x_1\,x_0$$

Aplicando a distributividade, complemento e a identidade multiplicativa:

$$\overline{x_2}\,\overline{x_1}\,x_0 + \overline{x_2}\,x_1\,x_0 = \overline{x_2}(\overline{x_1}x_0 + x_1x_0) = \overline{x_2}((\overline{x_1} + x_1)x_0) = \overline{x_2}(1.x_0) = \overline{x_2}x_0$$

Ident. multiplicativa

$$0 + a = a + 0 = a$$

$$1.a = a.1 = a$$

distributividade

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

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

complemento

$$a + a' = 1$$

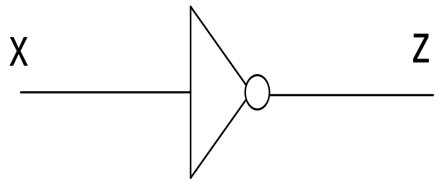
$$a.a' = 0$$

Portas Lógicas

São circuitos digitais que operam em conformidade com a álgebra Booleana.
Usadas para a construção de sistemas digitais.

Função NOT ou NÃO

- Símbolo:



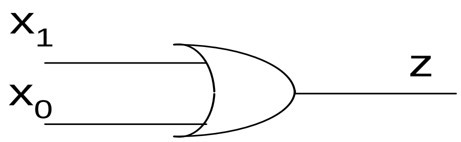
- Equação de chaveamento : $z = x'$ ou $z = \overline{x}$
- Tabela verdade:

entrada | saída

x	z
0	1
1	0

Função OR ou OU

- Símbolo:



- Equação de chaveamento :
- Tabela verdade:

$$z = x_1 + x_0$$

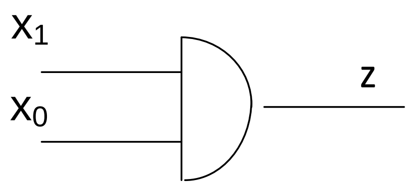
"x_1 ou x_0"

entrada | saída

x_1 x_0	z
0 0	0
0 1	1
1 0	1
1 1	1

Função AND

- Símbolo:



- Função de chaveamento :

$$z = x_1 . x_0$$

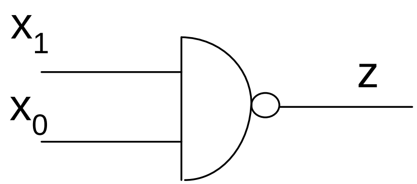
“x1 e x0”

- Tabela verdade:

entrada	saída
x_1 x_0	z
0 0	0
0 1	0
1 0	0
1 1	1

Função NAND

- Símbolo:



- Função de chaveamento :
- Tabela verdade:

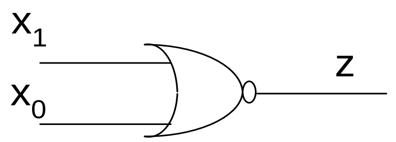
$$z = (x_1 . x_0)' \quad \text{ou} \quad z = \overline{\overline{(x_1 . x_0)}}$$

"x1 e x0, linha"

entrada	saída
x_1 x_0	z
0 0	1
0 1	1
1 0	1
1 1	0

Função NOR

- Símbolo:



- Função de chaveamento :
- Tabela verdade:

$$z = (x_1 + x_0)'$$

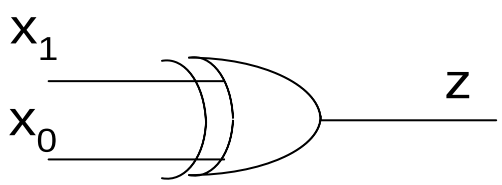
ou $$z = \overline{(x_1 + x_0)}$$

“x1 ou x0, linha”

entrada		saída
x_1	x_0	z
0	0	1
0	1	0
1	0	0
1	1	0

Função XOR (OU-exclusivo)

- Símbolo:



- Função de chaveamento :
- Tabela verdade:

$$z = x_1 . x_0' + x_1' . x_0$$

$$= x_1 \oplus x_0$$

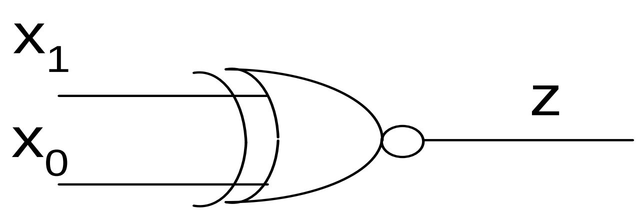
“x1 xor x0”

entrada | saída

x_1	x_0	z
0	0	0
0	1	1
1	0	1
1	1	0

Função XNOR

- Símbolo:



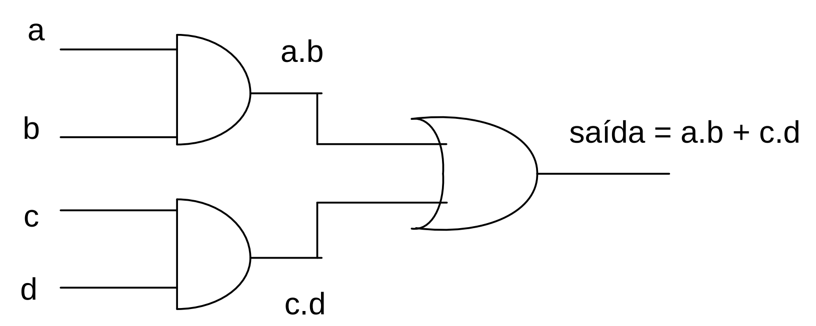
- Função de chaveamento : $z = x_1' . x_0' + x_1 . x_0$
- Tabela verdade:

entrada		saída
x_1	x_0	z
0	0	1
0	1	0
1	0	0
1	1	1

Composição de circuitos usando portas lógicas

- Compondo circuitos com portas lógicas, construímos os circuitos combinatórios.

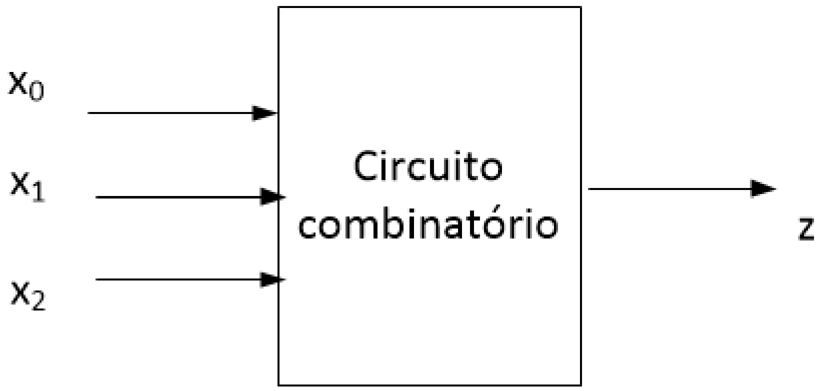
- Em cada saída de uma porta lógica é possível determinar a expressão Booleana correspondente em função das variáveis de entrada. Também, podemos obter a expressão da saída.
- Ex:



Aplicação das portas lógicas

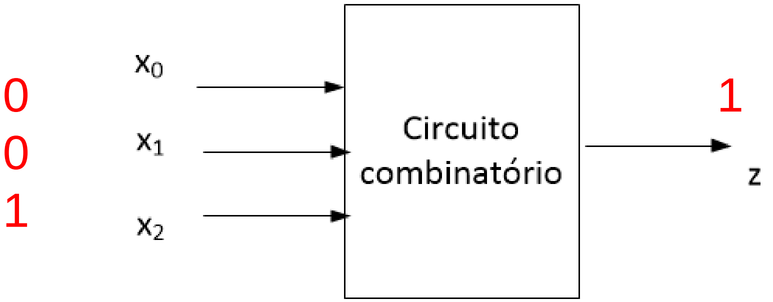
- As portas lógicas são usadas para construir circuitos combinatórios.
- Circuitos combinatórios podem ser representados por tabelas verdade, com um número finito de entradas e uma saída.

entrada	saída
x_2 x_1 x_0	z
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



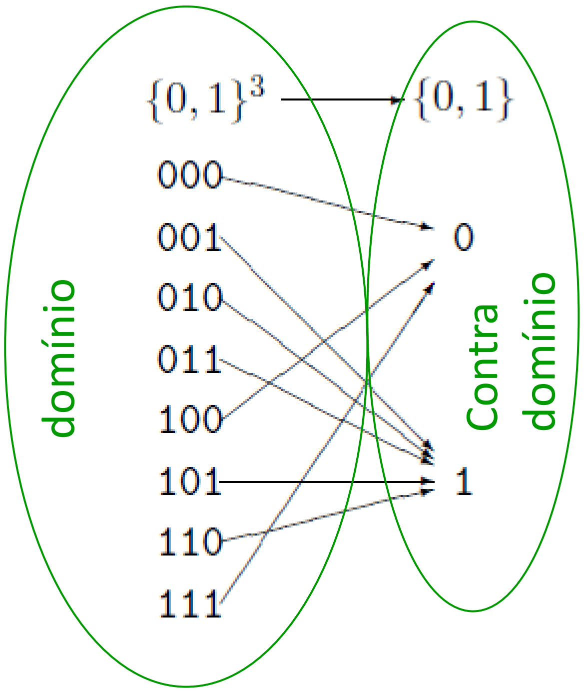
Funcionamento de um circuito combinatório

- O circuito combinatório funciona de acordo com a tabela verdade que o representa.
- Se na tabela verdade, com a entrada 001 a saída é 1, então, introduzindo a entrada 001 no circuito, a resposta deve ser 1.
- Portanto, a tabela verdade deve conter todas as possíveis combinações de entrada e as saídas correspondente.



Circuito Combinatório

- Um circuito combinatório de n variáveis $f\left(x_{n-1},\ldots,x_1,x_0\right)$ é uma função biunívoca do conjunto $\{0, 1\}^n$ para um conjunto $\{0, 1\}$.



Circuito combinatório com n = 3

Tabela verdade

$x_2\ x_1\ x_0$	z
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

Equação Booleana :

$z = x'_2x'_1x_0 + x'_2x_1x'_0 + x'_2x_1x_0 + x_2x'_1x'_0 + x_2x'_1x_0 + x_2x_1x'_0 + x_2x_1x_0$

Circuito combinatório

Equação : $z = x'_2x'_1x_0 + x'_2x_1x'_0 + x'_2x_1x_0 + x_2x'_1x'_0 + x_2x'_1x_0 + x_2x_1x'_0 + x_2x_1x_0$

A equação Booleana do circuito combinatório é obtida incluindo os termos correspondentes a todas as entradas cuja saída z resulta em 1 na tabela verdade.

No exemplo apenas a entrada $x'_2x'_1x'_0$ resulta em z = 0, portanto não consta na equação Booleana.

Essa forma da equação Booleana é conhecida como **forma canônica ou padrão**. E cada uma das entradas na equação é chamada **mintermo**.

Portanto cada mintermo representa uma entrada na tabela verdade.

Tabela verdade

$x_2\ x_1\ x_0$	z
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

Exercício 3

- Verificar se a tabela verdade abaixo é válida para a equação Booleana:

  $z = x_0 + x_1 + x_2$

Tabela verdade:

x_2 x_1 x_0	z
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

Equação Booleana de z expressa em forma canônica:

$$z = x'_2x'_1x_0 + x'_2x_1x'_0 + x'_2x_1x_0 + x_2x'_1x'_0 + x_2x'_1x_0 + x_2x_1x'_0 + x_2x_1x_0$$

Exercício 4

- Aplicando as identidades ou propriedades da álgebra booleana simplificar as expressões:

$x'_2x_1x'_0 + x'_2x_1x_0$

$x_2x'_1x'_0 + x_2x'_1x_0$

$x_2x_1x'_0 + x_2x_1x_0$

$x_2x'_1x_0 + x_2x_1x_0$

Respostas:

$x'_2x_1x'_0 + x'_2x_1x_0 = x'_2x_1$

$x_2x'_1x'_0 + x_2x'_1x_0 = x_2\,x'_1$

$x_2x_1x'_0 + x_2x_1x_0 = x_2\,x_1$

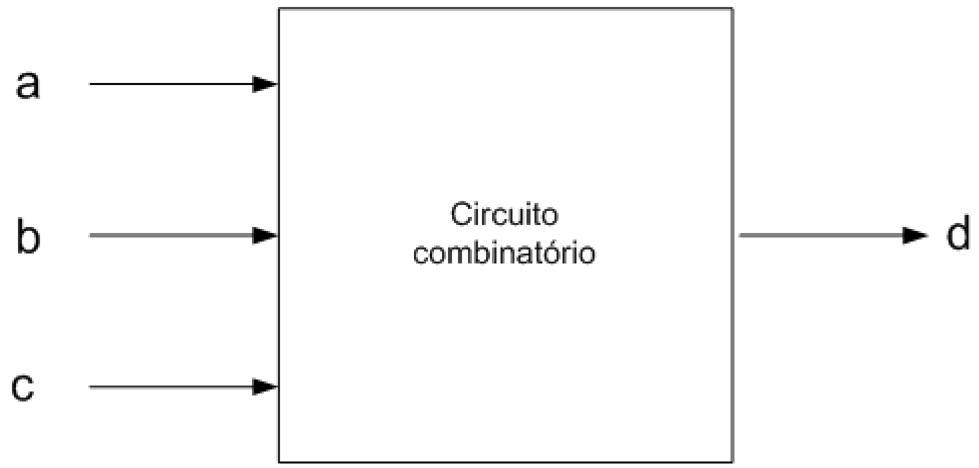
$x_2x'_1x_0 + x_2x_1x_0 = x_2\,x_0$

Exercício 5

- Dado um circuito combinatório de 3 entradas (a, b, c) e uma saída (d):
  1) mostrar todas as combinações de valores binários (0,1) de entradas possíveis;
  2) mostrar para cada uma das combinações, do item anterior, o mintermo correspondente.

Solução: a) 000, 001, 010, 011, 100, 101, 110, 111

b) a’b’c’, a’b’c, a’bc’, a’bc, ab’c’, ab’c, abc’, abc



Exercício 6

- Dada uma porta AND de 3 entradas (x_2, x_1, x_0) e saída (z), mostrar a sua tabela-verdade e a expressão Booleana.

- Solução:

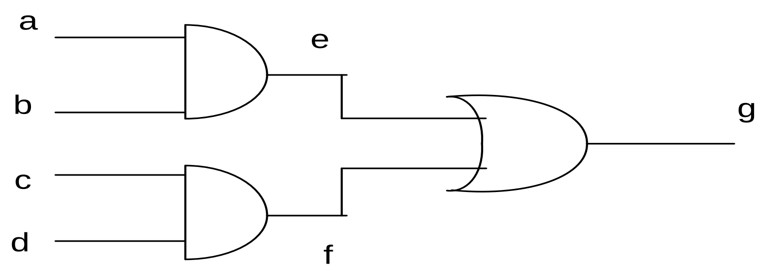
x_2 x_1 x_0	z
0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	0
1 0 0	0
1 0 1	0
1 1 0	0
1 1 1	1

- Expressão Booleana:

$$z = x_2 x_1 x_0$$

Exercício 7

- Dado o circuito com 2 portas AND de duas entradas e uma porta OR de duas entradas, conforme figura, obter a tabela verdade correspondente à saída g.



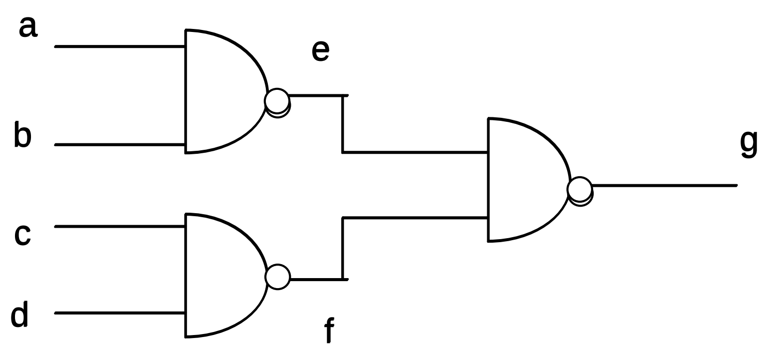
a b c d	e f	g
0 0 00	00	0
0 0 01	00	0
0 01 0	00	0
0 01 1	01	1
01 0 0	00	0
01 0 1	00	0
01 1 0	00	0
01 1 1	01	1
1 0 00	00	0
1 0 01	00	0
1 01 0	00	0
1 01 1	01	1
11 0 0	10	1
11 0 1	10	1
11 1 0	10	1
11 1 1	11	1

Solução:

Mostrando na tabela as variáveis intermediárias e e f e a saída g.

Exercício 8

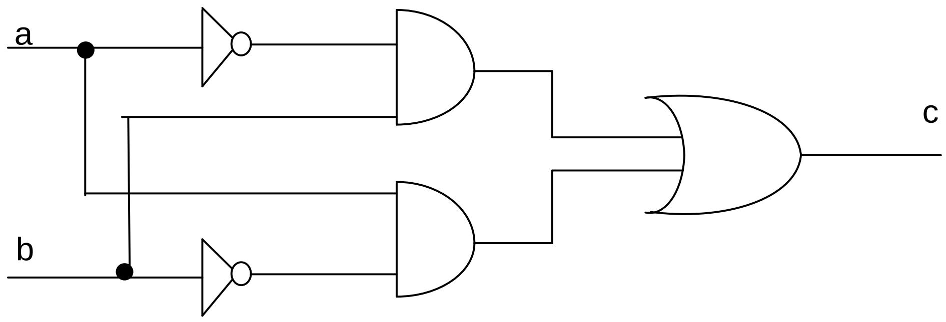
- Mostrar que o circuito de 3 portas NAND da figura é equivalente ao circuito do exercício anterior.



Solução: Da mesma forma que o exercício anterior, preenche-se a tabela obtendo primeiro os valores para as variáveis intermediárias *e* e *f* e posteriormente para a saída *g*.

Exercício 9

- Mostrar que o circuito da Figura é equivalente a uma porta XOR.



Uma porta XOR tem a equação Booleana:
c = a’b + ab’
que corresponde ao circuito da Figura.

Exercício 10

- Dado um circuito digital com
- 3 entradas e 1 saída,

cuja tabela-verdade é :

a b c_i	c_o
0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	1
1 1 1	1

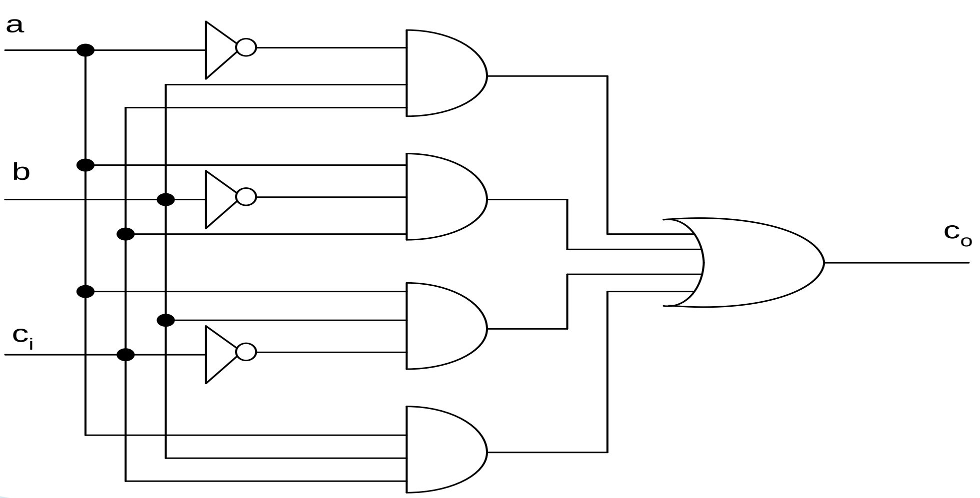
1) obter a expressão da função de chaveamento em forma canônica de c_o; e

2) desenhar o circuito digital resultante usando portas lógicas.

Solução:

Equação Booleana:

$c_o = a'bc_i + ab'c_i + abc_i' + abc_i$



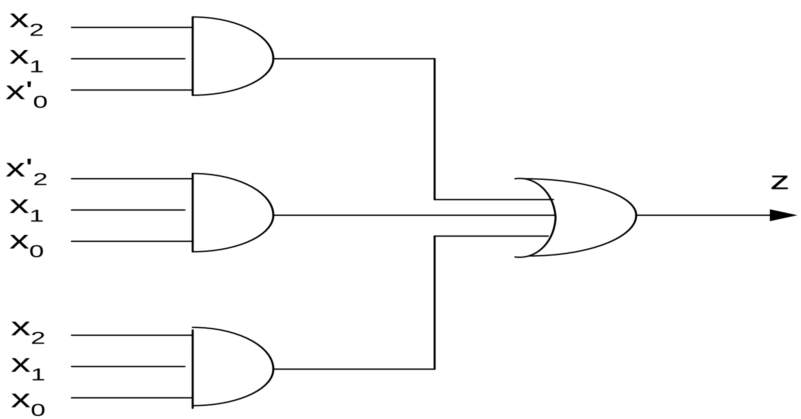
Equivalência de circuitos

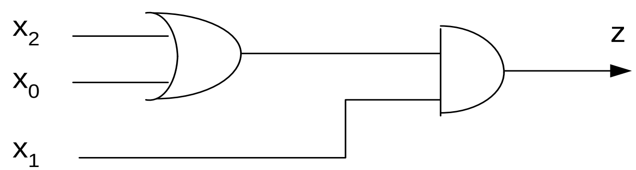
- Dois circuitos combinatórios são equivalentes quando as suas tabelas verdades são iguais.

- Utilizando as identidades da álgebra booleana é possível encontrar uma expressão Booleana comum para os dois circuitos equivalentes.

Exemplo de equivalência de circuitos

- Abaixo temos dois circuitos combinatórios equivalentes :



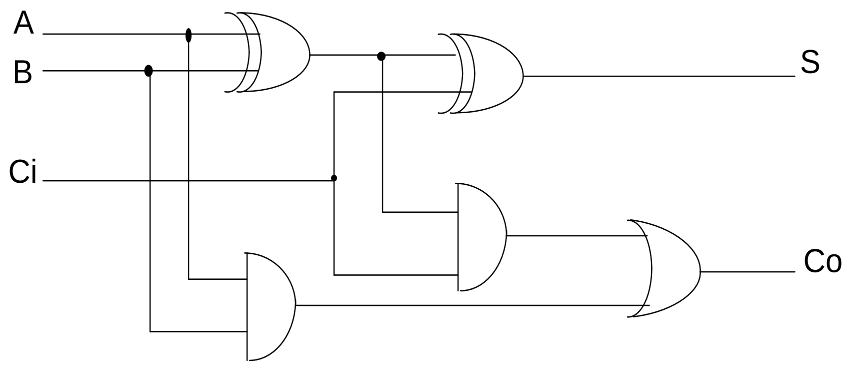


$$z = x_1.(x_2 + x_0)$$

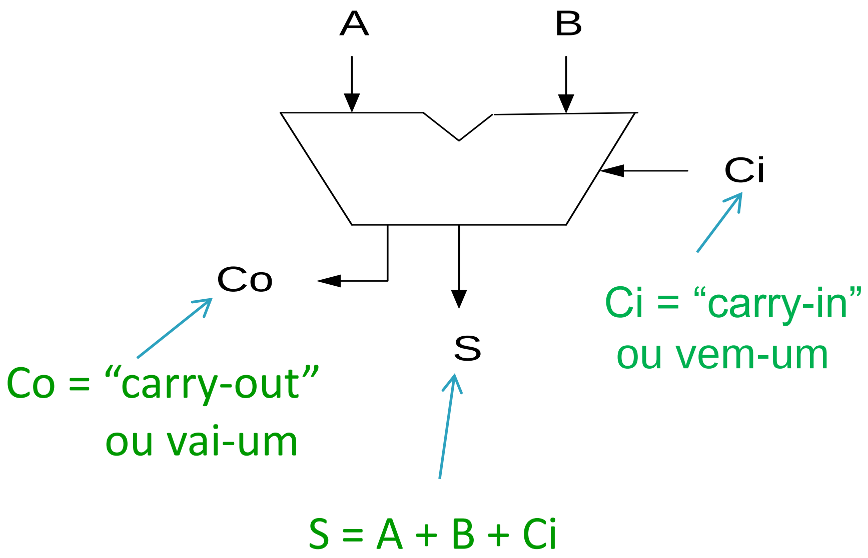
$$z = \left(x_2 x_1 x_0{}' \right) + \left(x_2{}' x_1 x_0 \right) + \left(x_2 x_1 x_0 \right)$$

Somador Completo

- Um circuito combinatório capaz de realizar operações de soma de dois bits, com entrada de vem-um, resultando em um bit de soma e vai-um, é chamado de somador completo.

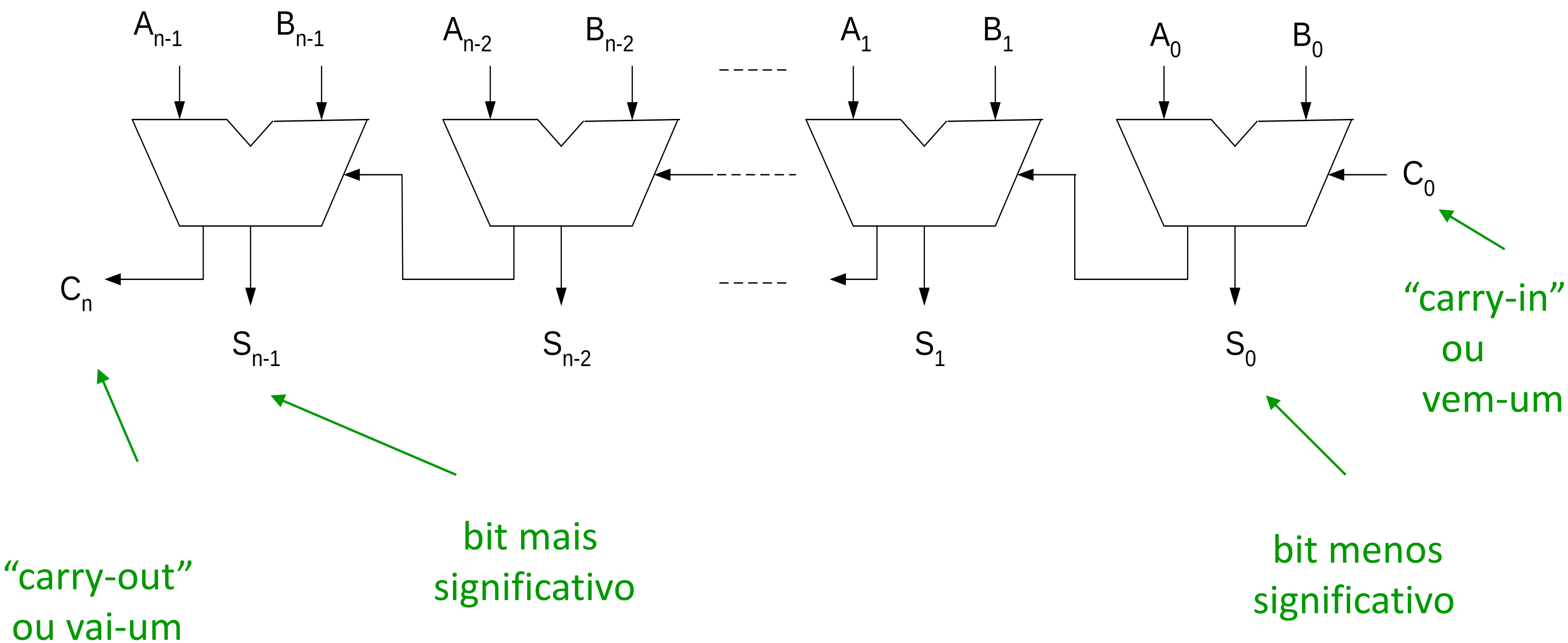


Somador completo de 1 bit representado por portas lógicas

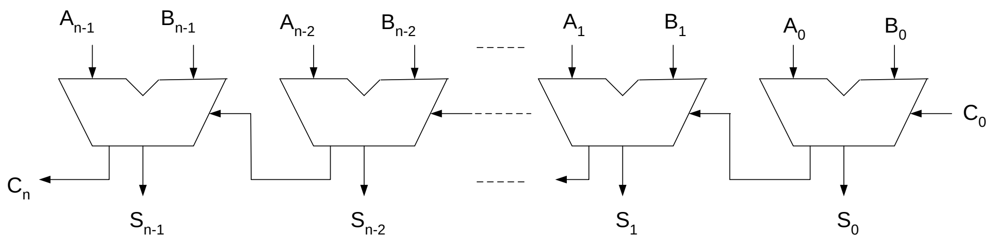


Símbolo do Somador completo

Somador de n bits



Considerações sobre os somadores



O somador de n bits acima, *ripple carry adder*, é lento devido à necessidade de espera do vem-um pelos módulos mais à esquerda (bits mais significativos).
O somador mais rápido, *carry look-ahead adder*, é provido de um circuito de cálculo de vai-um diretamente para os bits mais significativos.

Multiplexadores

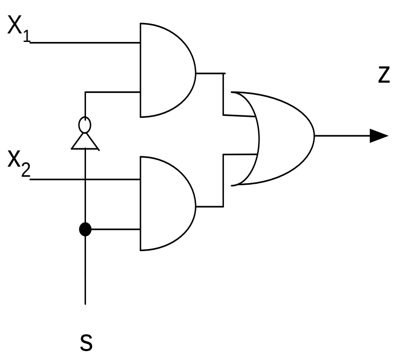
- Um multiplexador (MUX) consiste de um módulo que implementa um seletor ou uma estrutura de seleção.

- Considerando um caso de duas entradas de dados binários (x_1, x_0), um MUX possui ainda uma entrada de controle binária s e uma saída binária z e pode ser representado pela seguinte expressão:

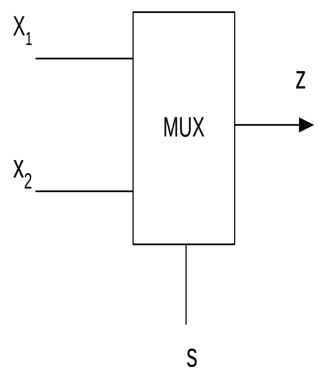
$$z = MUX(x_1, x_0, s) = x_1 s + x_0 s'$$

Multiplexadores

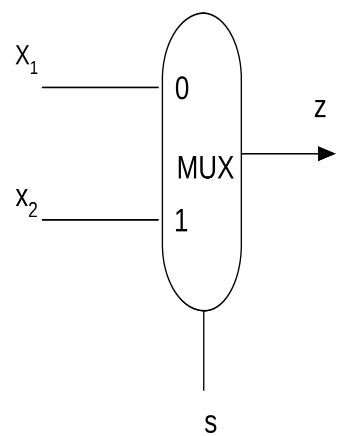
- O circuito e o símbolo lógico de um multiplexador 2-MUX (de 2 entradas) são ilustrados abaixo.



circuito



Símbolo

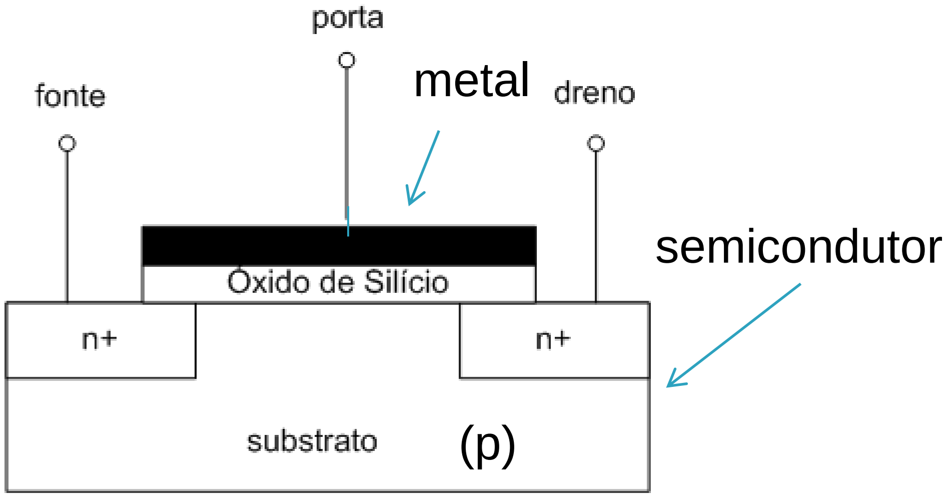


símboloalternativo

Transistor MOS (Metal-Óxido-Semicondutor)

Tipo de transistor usado para construção de circuitos integrados com baixo consumo de energia.

Corte transversal de um transistor MOS – canal N.

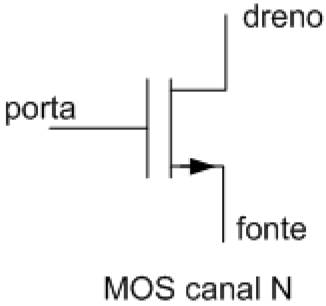


Substrato – base de silício onde se constrói o transistor.
O substrato (p) é constituído de cargas positivas.

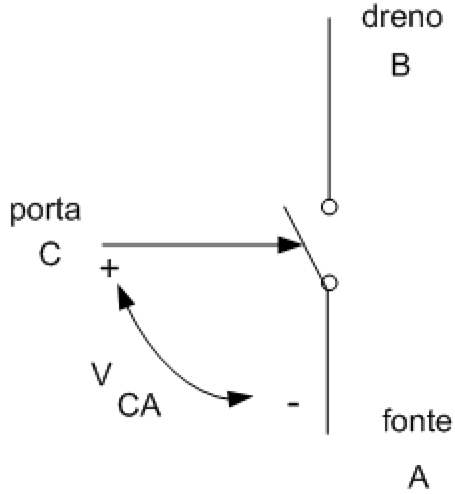
A fonte e o dreno são regiões com alta quantidade de cargas negativas (n+).

A porta controla a conexão entre a fonte e o dreno, fazendo funcionar a região (canal) entre a fonte e o dreno, como capacitor.

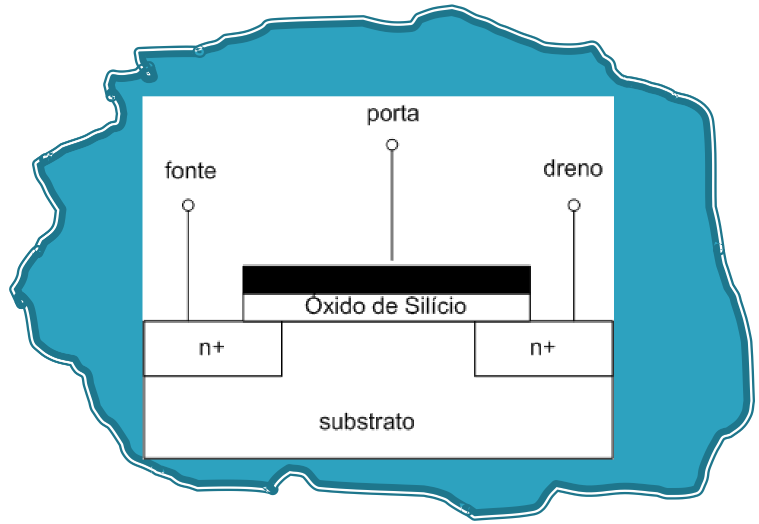
Transistor MOS – Canal N

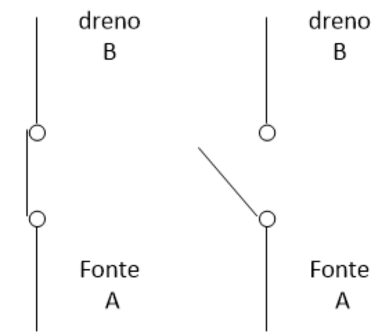


símbolo



função



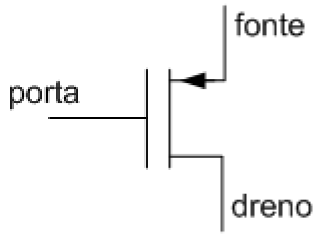


fechado aberto

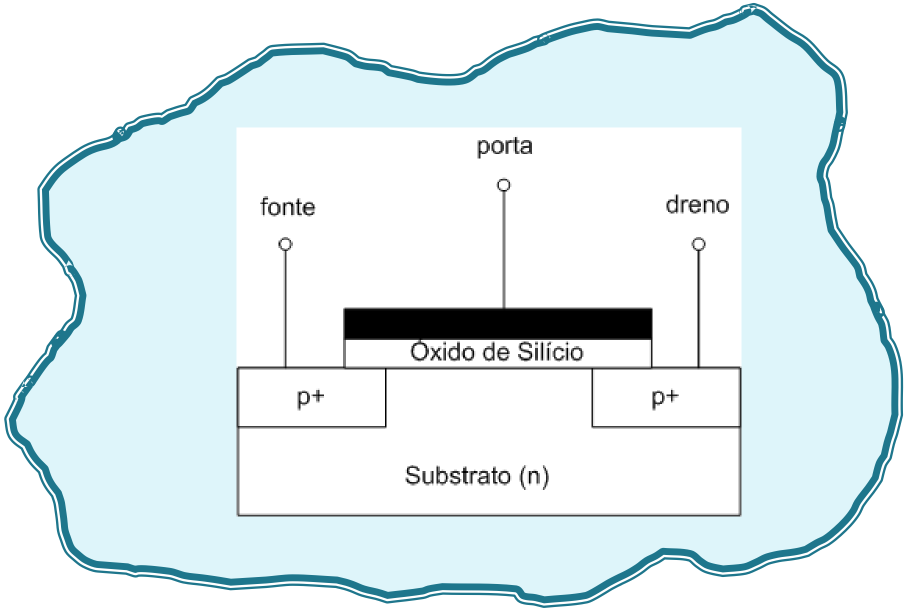
Quando V_{CA} for maior que um limiar (threshold) fecha-se o contato entre o dreno (B) e fonte (A), caso contrário, o contato é aberto.

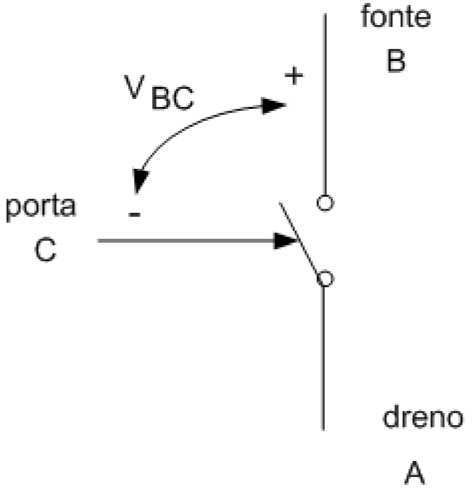
Transistor MOS – canal P

símbolo



MOS canal P



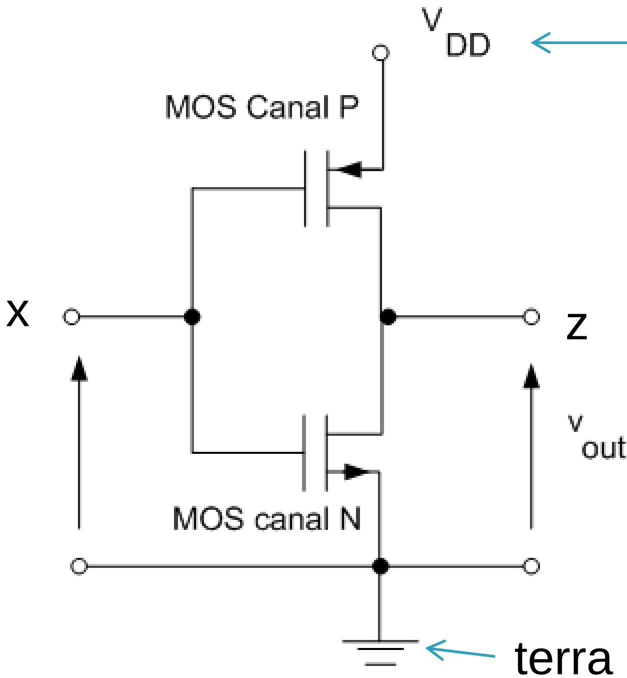


função

Quando a tensão V_{BC} for maior que um limiar, fecha-se a conexão entre a fonte e o dreno, abrindo caso contrário.

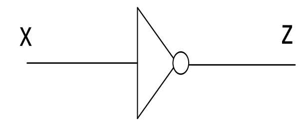
Circuito inversor com transistores MOS, N e P

(voltagem de alimentação positiva)



terra

porta not



Símbolo do inversor

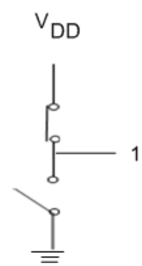
$$z = x' \quad ou \quad z = \overline{x}$$

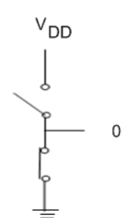
Lê-se 'x linha' ou 'x barra'

entrada

saída

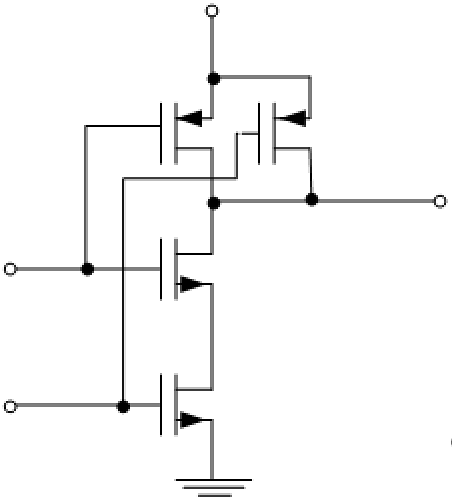
x	z
0	1
1	0

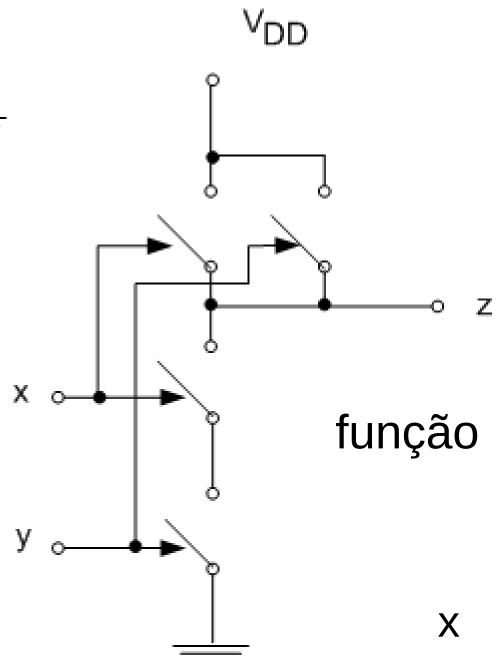




0 = terra
1 = voltagem alta

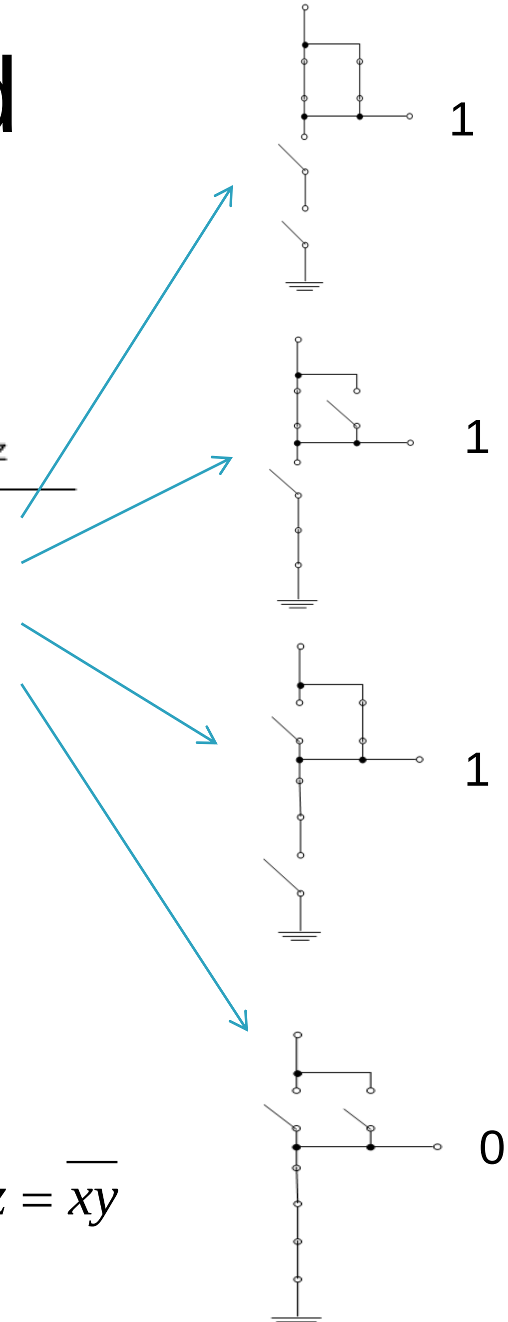
Circuito nand

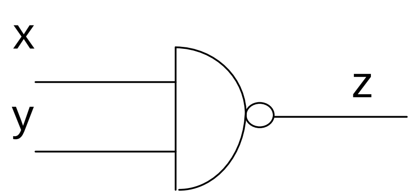




função

x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

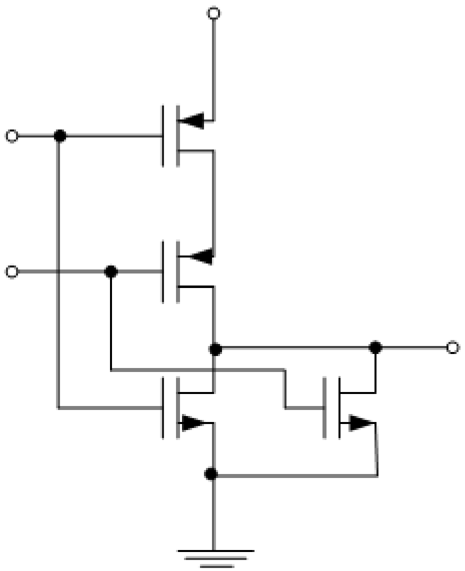




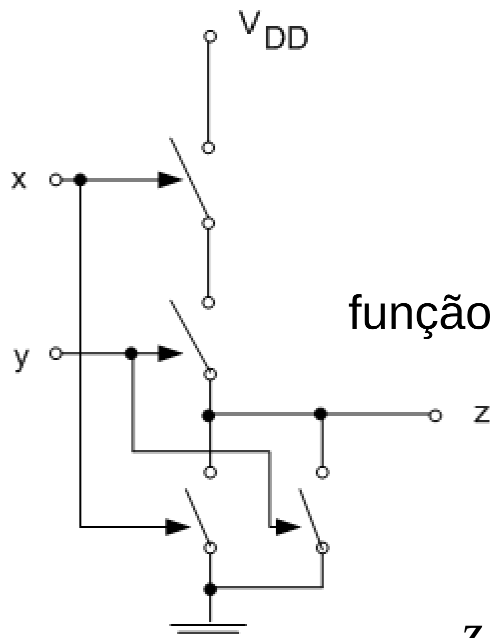
porta nand

$$z = (xy)' \quad ou \quad z = \overline{xy}$$

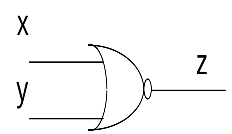
Circuito nor



x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

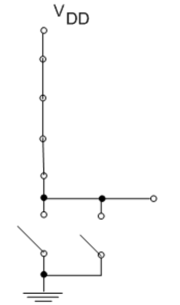


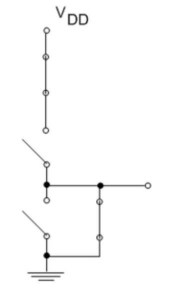
função

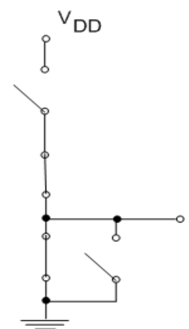


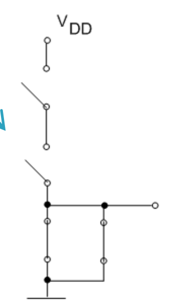
porta nor

$$z = (x + y)' \quad ou \quad z = \overline{x + y}$$

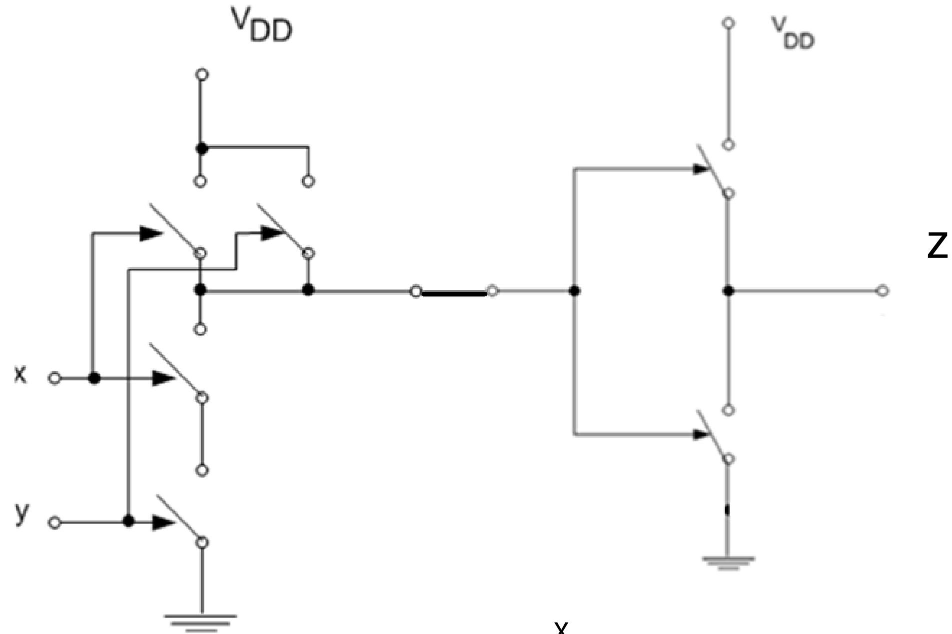




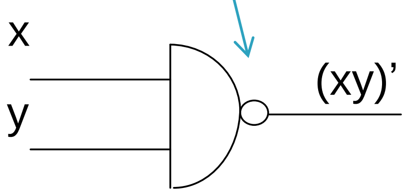




and = not (nand)



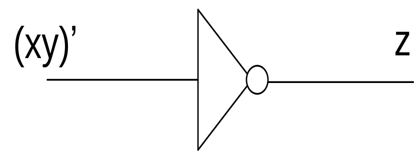
essa bola
significa inversão



$$z = \overline{xy}$$

x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

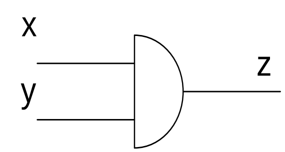
nand



$$z = \overline{x}$$

x	z
0	1
1	0

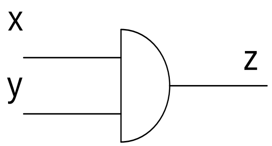
not



$$z = xy$$

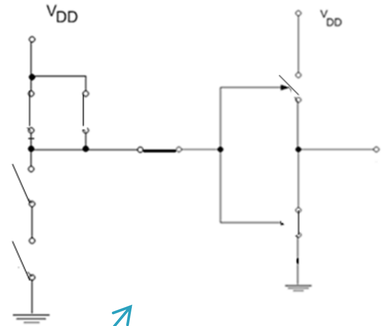
x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

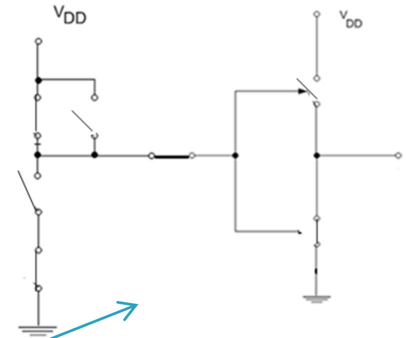
and

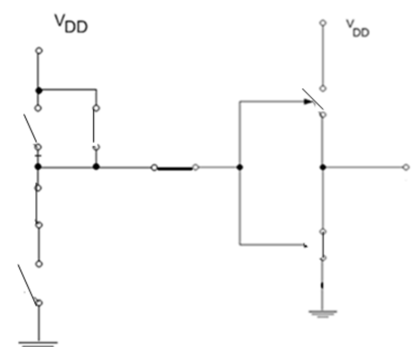


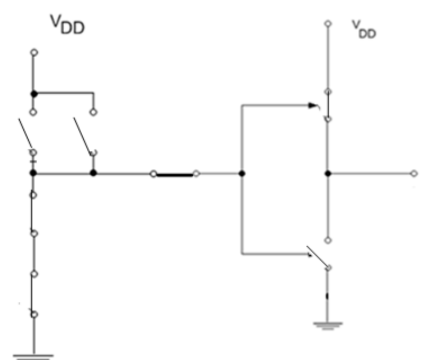
and

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

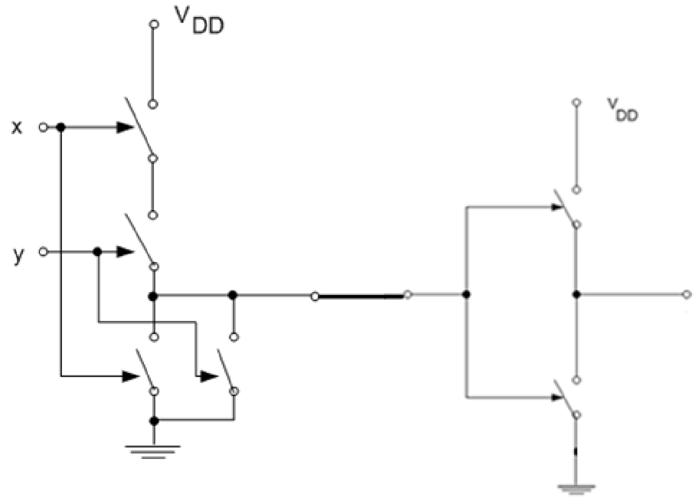


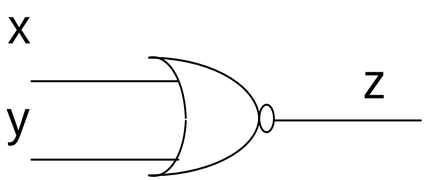


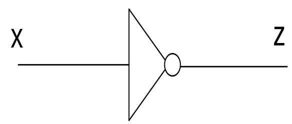


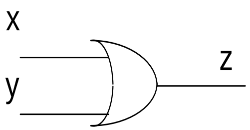


or = not (nor)









x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

nor

x	z
0	1
1	0

not

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

or

Referências

- [1] ERCEGOVAC, M.; LANG, T.; MORENO, J. Tradução: SANTOS, J. C. B. Introdução aos Sistemas Digitais. Porto Alegre: Bookman, 2000.

- [2] TANENBAUM, A.; AUSTIN, T. Tradução: VIEIRA, D. Organização estruturada de computadores, 6ª edição, Pearson Prentice Hall, 2013.

Fim!!