Álgebra Booleana – Resumo (parte 2)

Operadores secundários

São operadores criados a partir dos operadores primitivos. Aqui serão mostrados três operadores secundários:

- NÃO E (também chamado de NAND)
- NÃO OU (também chamado de NOR)
- OU EXCLUSIVO (também chamado de EXNOR)

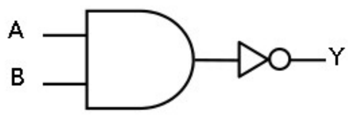
Operador NÃO E (NAND)

É um operador criado com os operadores E e NÃO.

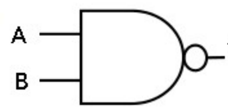
Expressão

$$Y=\overline{A.B}$$

Circuito lógico equivalente



Símbolo



Antes de iniciar a apresentação dos operadores secundários faremos uma explicação sobre o operador de negação quando utilizado sobre uma expressão booleana. Neste caso, assume-se, por convenção, que a expressão que está abaixo do sinal de negação está entre parênteses. Por exemplo, a expressão $\overline{A.B}$

deve ser interpretada como

$$\overline{(A.B)}$$

Para ficar mais clara essa interpretação, observaremos que nas linguagens de programação C, Java, PHP, dentre outras, essa operação seria implementada pela expressão abaixo, onde nota-se obrigatoriamente o uso do parênteses:

```
! ( A && B )
```

Nesse caso, seguindo-se as regras das precedências entre operadores, deve-se primeiro executar a expressão entre parênteses para depois executar o operador de negação. Portanto, vamos calcular a tabela verdade para esse operador:

Calculando a tabela verdade da expressão $Y = \overline{A.B}$

Calculando para A=0 e B=0

$Y = \overline{0.0}$ primeiro resolver 0.0

$Y = \overline{0}$ agora sim, resolver a negação

$Y = 1$

Calculando para A=0 e B=1

$Y = \overline{0.1}$ primeiro resolver 0.1

$Y = \overline{0}$ agora sim, resolver a negação

$Y = 1$

Calculando para A=1 e B=0

$Y = \overline{1.0}$ primeiro resolver 1.0

$Y = \overline{0}$ agora sim, resolver a negação

$Y = 1$

Calculando para A=1 e B=1

$Y = \overline{1.1}$ primeiro resolver 1.1

$Y = \overline{1}$ agora sim, resolver a negação

$Y = 0$

Ou, usando o formato de tabela e calculando as parciais...

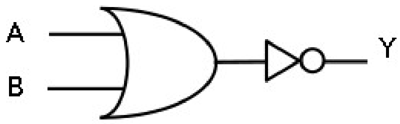
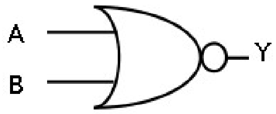
A	B	A.B	$\overline{A.B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Portanto, a tabela verdade da expressão $\overline{A.B}$ é:

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

Operador NÃO OU (NOR)

É um operador criado com os operadores OU e NÃO.

Expressão	Circuito lógico equivalente	Símbolo
$Y=\overline{A+B}$		

Calculando a Tabela Verdade do Operador NÃO OU (NOR)

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

Tabela Verdade do Operador NÃO OU (NOR)

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

Operador OU EXCLUSIVO (XOR)

É um operador que resulta verdadeiro somente se apenas uma das entradas for verdadeira.

$$S = A \oplus B$$

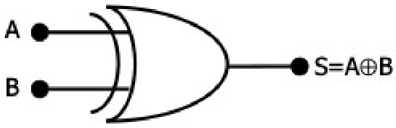


Tabela Verdade do Operador OU EXCLUSIVO

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

É um operador bem complexo se pensarmos na sua implementação a partir de operadores primitivos, conforme mostra a expressão $Y = \overline{A}.B + A.\overline{B}$

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