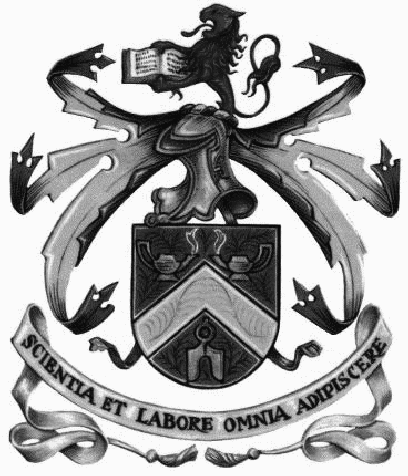
**Universidade de Trás-os-Montes e Alto Douro Departamento de Engenharias**

Sistemas Digitais

1º Ano de Engenharia Informática



Trabalho Prático n.º 2

*Álgebra de Boole*

Grupo



n.º

n.º

n.º

Turma

FMGG / 2008

### Objectivos

* Verificar com medições algumas leis e teoremas da **Álgebra Booleana**
* Usar as leis e teoremas para simplificar expressões booleanas

### Referências

* TAUB, Herbert, “Circuitos Digitais e Microprocessadores”, McGraw–Hill
* Texas Instruments online [<http://www.ti.com/>]

### Material

* Placa RH21
* Circuito Integrado (CI) 74LS04 — inversor (NOT)
* Circuito Integrado (CI) 74LS08 — AND, duas entradas
* Circuito Integrado (CI) 74LS11 — AND, três entradas
* Circuito Integrado (CI) 74LS32 — OR, duas entradas

# Leis fundamentais da Álgebra Booleana

As leis fundamentais da Álgebra Booleana são importantes como meio para a avaliação e sim- plificação de expressões booleanas, e formam a base a partir da qual surgem outros teoremas.

## Leis comutativas

As duas leis comutativas (ao lado) afirmam simplesmente que a

**ordem** da **adição ou multiplicação lógicas** não é importante.

***A* + *B* = *B* + *A A* . *B* = *B* . *A***

## Leis associativas

As duas leis associativas (ao lado, exemplo para três vari- áveis) oferecem a possibilidade de **agrupar termos** da adi- ção ou da multiplicação lógicas de qualquer forma.

## Lei distributiva

***A* + (*B* + *C*) = (*A* + *B*) + *C A* . (*B* . *C*) = (*A* . *B*) . *C***

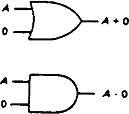
A lei distributiva possibilita a **expansão por multiplicação termo-a-termo**. De modo inver- so, também permite a **factorização de uma expressão** (vulgo. “pôr em evidência”). Ao contrário da Álgebra tradicional, na Álgebra Booleana também existe a distributividade da adição em relação à multiplicação (página seguinte).

***A* . (*B* + *C*) = (*A* . *B*) + (*A* . *C*)**

***A* + (*B* . *C*) = (*A* + *B*) . (*A* + *C*)**

As oito leis que se seguem, agrupadas em quatro pares, são diferentes da Álgebra usual e formam a coluna vertebral da Álgebra Booleana. Cada par tem as operações **AND** (multiplicação lógica) e **OR** (adição lógica), e cada lei pode ser facilmente visualizada em termos de **portas lógicas**.

## Operações com 0

Preencha as tabelas de verdade apresentadas em baixo com os **valores lógicos** obtidos experimentalmente e conclua quanto à **regra geral** (à esquerda).

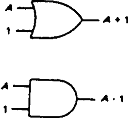
|  |  |
| --- | --- |
| *A* | *A* + 0 |
| 0 | 0 |
| 1 | 1 |

*A* + 0 = A

*A* . 0 = 0

|  |  |
| --- | --- |
| *A* | *A* . 0 |
| 0 | 0 |
| 1 | 0 |

## Operações com 1

Preencha as tabelas de verdade apresentadas em baixo com os **valores lógicos** obtidos experimentalmente e conclua quanto à **regra geral** (à esquerda).

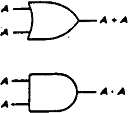
|  |  |
| --- | --- |
| *A* | *A* + 1 |
| 0 | 1 |
| 1 | 1 |

*A* + 1 = 1

*A* . 1 = A

|  |  |
| --- | --- |
| *A* | *A* . 1 |
| 0 | 0 |
| 1 | 1 |

## Operações duma variável com ela própria

Preencha as tabelas de verdade apresentadas em baixo com os **valores lógicos** obtidos experimentalmente e conclua quanto à **regra geral** (à esquerda).

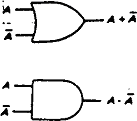
|  |  |
| --- | --- |
| *A* | *A* + *A* |
| 0 | 0 |
| 1 | 1 |

*A* + *A* = A

*A* . *A* = A

|  |  |
| --- | --- |
| *A* | *A* . *A* |
| 0 | 0 |
| 1 | 1 |

## Operações duma variável com o seu complemento

Preencha as tabelas de verdade apresentadas em baixo com os **valores lógicos** obtidos experimentalmente e conclua quanto à **regra geral** (à esquerda).

|  |  |
| --- | --- |
| *A* | *A*  *A* |
| 0 | 1 |
| 1 | 1 |

|  |  |
| --- | --- |
| *A* | *A*.*A* |
| 0 | 0 |
| 1 | 0 |

*A*  *A*  1

* 1.  0

## Elementos neutros e elementos absorventes

Preencha a tabela seguinte com as conclusões que se podem tirar dos pontos anteriores.

|  |  |  |
| --- | --- | --- |
|  | **Adição lógica (OR)** | **Multiplicação lógica (AND)** |
| **Elemento Neutro:** | 0 | 1 |
| **Elemento Absorvente:** | 1 | 0 |

# Teoremas Booleanos

As leis básicas da Álgebra Booleana que se seguem podem ser usadas para obter um sem-nú- mero de teoremas úteis:

*A*  *A*

*A* + *AB* = *A A*.(*A* + *B*) = *A*

*A*  *AB*  *A*  *B A*.( *A*  *B*)  *AB*

*AB*  *AB*  *A*

*AB*  *AC*  ( *A*  *C*)( *A*  *B*) ( *A*  *B*)( *A*  *C*)  *AC*  *AB*

*AB*  *AC*  *BC*  *AB*  *AC*

( *A*  *B*)( *A*  *C*)(*B*  *C*)  ( *A*  *B*)( *A*  *C* )

* 1.  *A*  *B*

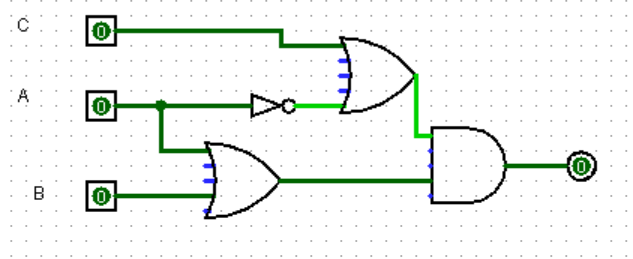
( *A*  *B*).( *A*  *B* )  *A*

*A*  *B* 

*A*.*B*

O último par de teoremas (coluna da direita), devido a **DeMorgan**, são expressões paras as portas **NAND** ( *A*.*B* ) e **NOR** ( *A*  *B* ). Estes dois teoremas devem ser memorizados, pois são muito úteis na simplificação de expressões contendo complementos.

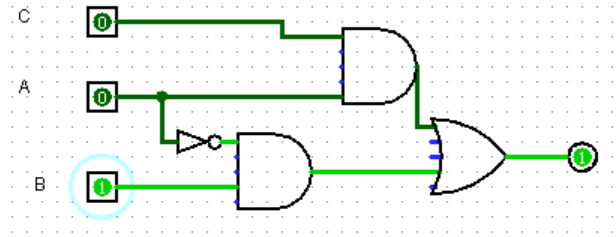
* 1. Desenhe o circuito lógico para ( *A*  *B*)( *A*  *C*) :



(fiz uma troca entre o A, B e o C, para ser mais percetível o esquema)

* 1. Desenhe o circuito lógico para

*AC*  *AB* :



* 1. Verifique, **por indução** (**prática**), o teorema ( *A*  *B*)( *A*  *C*)  *AC*  *AB* . Para tal **imple- mente o circuito lógico** correspondente ao primeiro membro, aplique nas entradas os va- lores lógicos apresentados na tabela de verdade em baixo e registe os valores de saída na mesma tabela. Proceda de igual modo para o segundo membro da igualdade.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *A* | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| *B* | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| *C* | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| ( *A*  *B*)( *A*  *C*) | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| *AC*  *AB* | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |

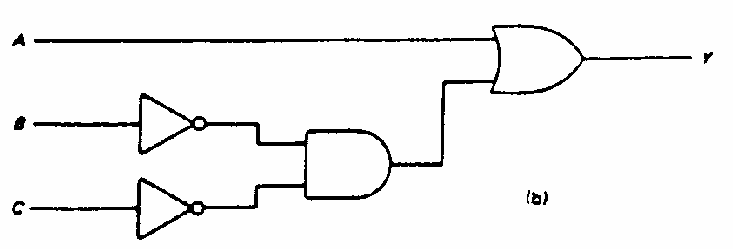
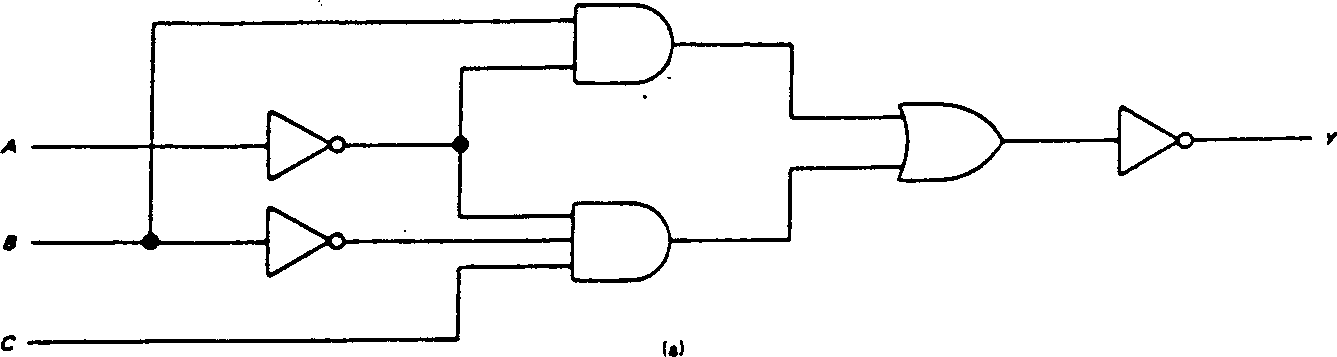
* 1. Use a **indução** (**teórica**) para provar o teorema circuitos — sirva-se apenas da seguinte **tabela**:

*AB*  *A*( *A*  *B*) . **Não implemente** os

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *A* | *B* | *A* | *AB* | *A*  *B* | *A*( *A*  *B*) |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |

 

* 1. Use a **indução** (**prática**), **implementando os circuitos**, para verificar a equivalência dos dois circuitos lógicos apresentados a seguir:



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *A* | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| *B* | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| *C* | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| a) | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| b) | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |