Funções Lógicas Básicas

Prof. Gustavo Girão
girao@imd.ufrn.br

Baseado no material do Prof. Ricardo Weber (UFRGS)

Roteiro

- Álgebra Booleana
  - Função E ou AND
  - Função OU ou OR
  - Função NÃO ou NOT
- Representações algebricas e por portas lógicas

Algebra Booleana

- Em 1854 o matemático inglês George Boole apresentou um sistema matemático de análise lógica conhecido como **álgebra de Boole.**

- Somente em 1938, um engenheiro americano utilizou as teorias da álgebra de Boole para a solução de problemas de circuitos de telefonia com relés, tendo publicado um artigo que praticamente introduziu na área tecnológica o campo da eletrônica digital.

Álgebra Booleana

- Os sistemas digitais são formados por circuitos lógicos denominados de portas lógicas que, utilizados de forma conveniente, podem implementar todas as expressões geradas pela álgebra de Boole.

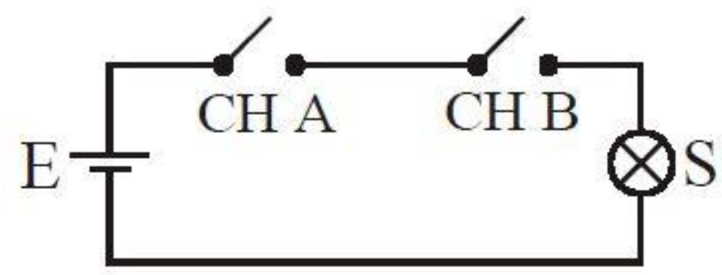
- Existem três portas básicas (**E, OU e NÃO) que podem ser conectadas de várias maneiras, formando sistemas que vão de simples relógios digitais aos computadores de grande porte.**

Função E ou AND

- A função **E é aquela que executa a multiplicação de duas ou mais variáveis booleanas. Sua representação algébrica para duas variáveis é S=A.B, onde se lê: S=A e B.**

- Para compreender a função **E da álgebra Booleana, podemos analisar um circuito com duas chaves e uma lâmpada em série, para o qual se adota as seguintes convenções:**
- **chave aberta=0, chave fechada=1,**
- **lâmpada apagada=0 e lâmpada acesa=1.**

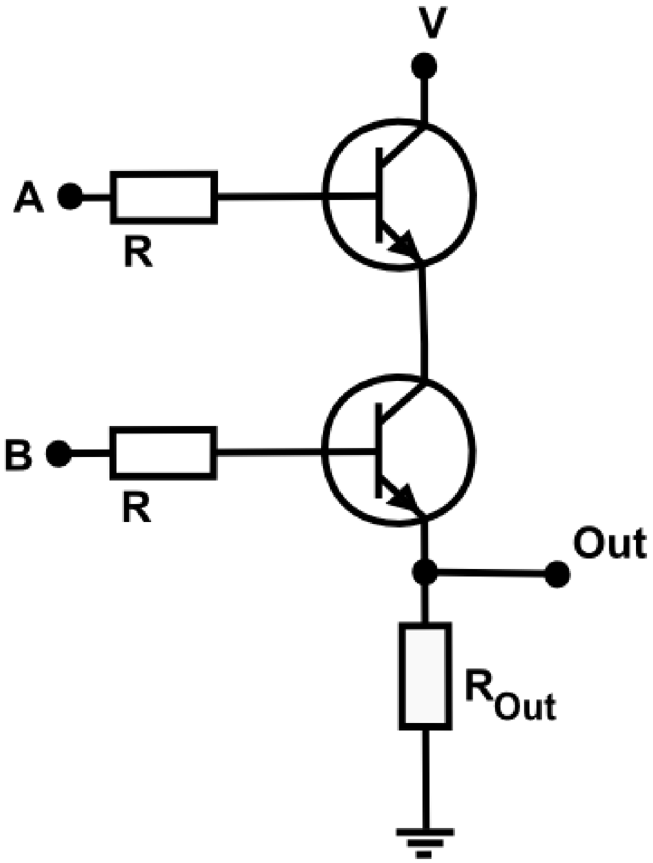
Função E ou AND



A análise da Figura revela que a lâmpada somente acenderá se ambas as chaves estiverem fechadas e, seguindo a convenção, tem-se: CH A=1, CH B=1, resulta em S=1.

Função E ou AND

- Porta lógica AND com transistores



Função E ou AND

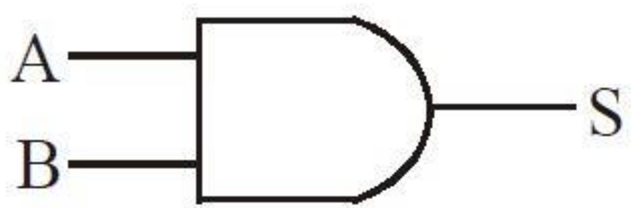
- Pode-se, desta forma, escrever todas as possíveis combinações de operação das chaves na chamada **Tabela da Verdade, que é definida como um mapa onde se depositam todas as possíveis situações com seus respectivos resultados. O número de combinações possíveis é igual a 2^N, onde N é o número de variáveis de entrada.**

Tabela da verdade da função ***E****.*

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

Função E ou AND

- A **porta lógica E é um circuito que executa a função E da álgebra de Boole, sendo representada, na prática, através do símbolo abaixo**



- **"A saída da porta E será 1, somente se todas as entradas forem 1".**

Função E ou AND

Exemplo: para a construção de uma parede é necessária a utilização de tijolos **e** cimento

A = utilização de tijolos

B = utilização de cimento

X = construção da parede

$$X = A.B = AB = A \wedge B \tag{2}$$

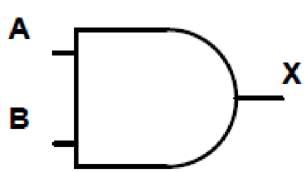


Tabela Verdade		
A	B	**X**
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

Função OU ou OR

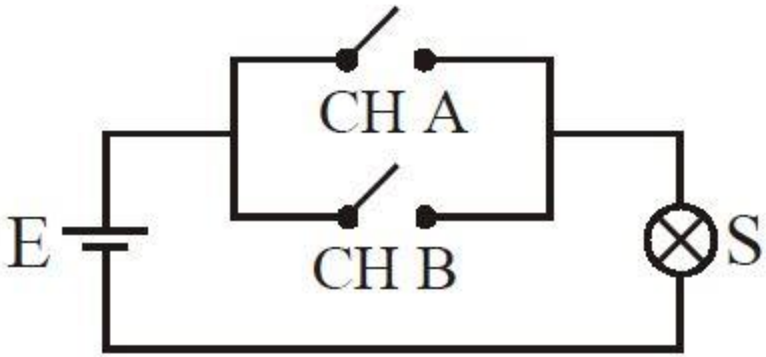
- A função **OU é aquela que assume valor 1 quando uma ou mais variáveis de entrada forem iguais a 1 e assume 0 se, e somente se, todas as variáveis de entrada forem iguais a zero. Sua representação algébrica para duas variáveis de entrada é S=A+B, onde se lê: S=A ou B.**

- Para entender melhor a função **OU da álgebra booleana, analisa-se todas as situações possíveis de operação das chaves de um circuito com chaves em paralelo.**

- **A convenção é a mesma adotada anteriormente:**
- **chave aberta=0, chave fechada=1,**
- **lâmpada apagada=0 e lâmpada acesa=1.**

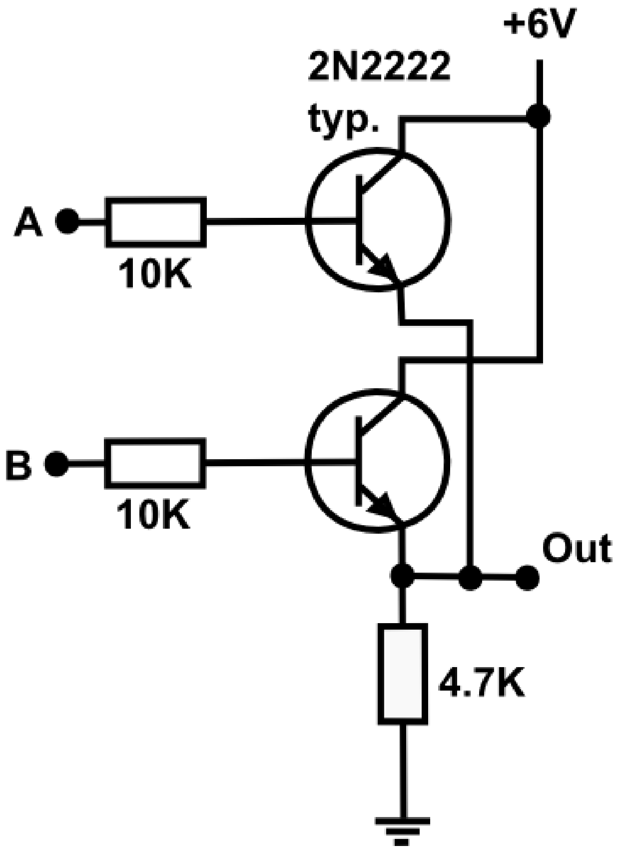
Função OU ou OR

- O circuito abaixo mostra que a lâmpada acende quando qualquer uma das chaves estiver fechada e permanece apagada se ambas estiverem abertas, ou seja, CH A=0, CH B=0, resulta em S=0.



Função OU ou OR

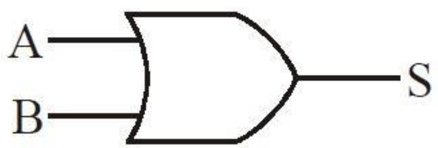
- Porta lógica OR com transistores



IMD0121

Função OU ou OR

- A Figura abaixo ilustra a porta lógica que executa a função **OU da álgebra de Boole, juntamente com a sua tabela da verdade.**



Porta lógica ***OU***

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

Tabela da verdade da função ***OU***

- **“A saída de uma porta OU será 1 se uma ou mais entradas forem 1”.**

Função OU ou OR

Exemplo: a compra será concluída se for realizado o pagamento com dinheiro **ou** cartão
A = pagamento com dinheiro
B = pagamento com cartão
X = conclusão da compra

$$X = A + B = A \vee B \qquad (1)$$

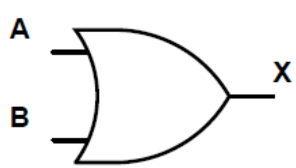


Tabela Verdade		
A	B	**X**
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

Função NÃO ou NOT

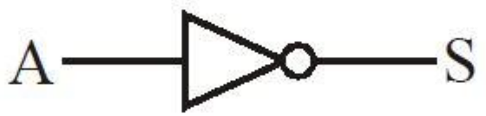
- A função **NÃO é aquela que inverte ou complementa o estado da variável de entrada, ou seja, se a variável estiver em 0, a saída vai para 1, e se estiver em 1 a saída vai para 0.**

- **É representada algebricamente da seguinte forma:, onde se lê: A barra, A ou NÃO A.**

$$\overline{A}$$

Função NÃO ou NOT

- O inversor é o bloco lógico que executa a função **NÃO. Sua representação simbólica é vista na Figura juntamente com sua tabela da verdade.**



*Porta lógica **NÃO** ou **inversora***

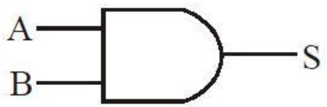
A	S
0	1
1	0

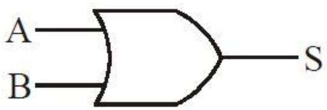
*Tabela da verdade da função **NÃO***

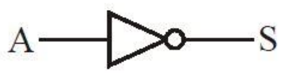
- **“A saída de uma porta NÃO assume o nível lógico 1**
- **somente quando sua entrada é 0 e vice-versa”.**

Blocos Lógicos Basicos

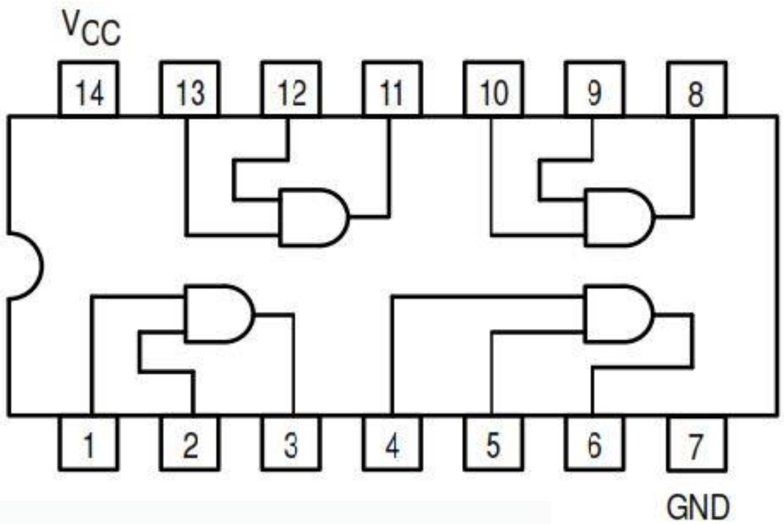
<table>
<tr><th colspan="5">BLOCOS LÓGICOS BÁSICOS</th></tr>
<tr><th>PORTA</th><th>Símbolo Usual</th><th>Tabela da Verdade</th><th>Função Lógica</th><th>Expressão</th></tr>
<tr><td>E

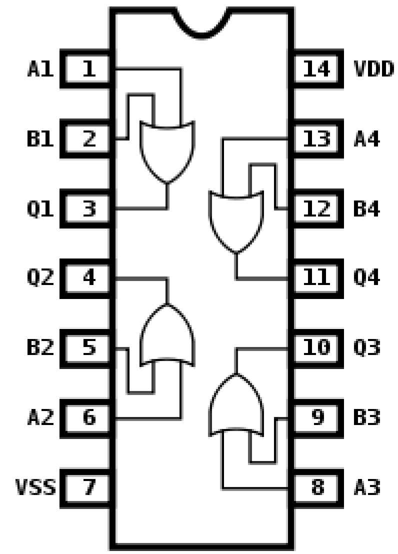
AND</td><td>
</td><td><table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td>Função E:
Assume 1 quando todas as variáveis forem 1 e 0 nos outros casos.</td><td>$S=A.B$</td></tr>
<tr><td>OU

OR</td><td>
</td><td><table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td>Função E:
Assume 0 quando todas as variáveis forem 0 e 1 nos outros casos.</td><td>$S=A+B$</td></tr>
<tr><td>NÃO

NOT</td><td>
</td><td><table><tr><th>A</th><th>S</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td><td>Função NÃO:
Inverte a variável aplicada à sua entrada.</td><td>$S=\overline{A}$</td></tr>
</table>

Como são comercializados





Funções booleanas

- O que é uma função booleana?
  - A função booleana pode ser completamente descrita por uma tabela com 2^n linhas
  - Cada linha da tabela é uma combinação diferente de valores de entrada
  - Esta tabela é denominada **tabela-verdade**

Funções booleanas

- Uma função booleana de 3 variáveis ***M = f(A,B,C)*** é representada pela tabela verdade mostrada abaixo:

A	B	C	M
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

- Essa é a função de lógica majoritária, ela é 0 se a maioria das entradas é 0 e é 1 quando a maioria das entradas é 1.

Funções booleanas

- Além da tabela verdade, há uma outra notação para representar as funções booleanas => ***a equação booleana***
- Qualquer função booleana pode ser especificada ao se dizer qual conjunto de variáveis de entrada dão um valor de saída igual a 1.
- Exemplo: $A \cdot \overline{B} \cdot C$ assume o valor 1 quando A = 1e B = 0 e C = 1

Funções booleanas

- Para a função do exemplo anterior, há 4 combinações de variáveis de entrada que fazem com que M seja 1.
- Portanto M será 1 se:
- A = 0 e B = 1 e C = 1 ou
- A = 1 e B = 0 e C = 1 ou
- A = 1 e B = 1 e C = 0 ou
- A = 1 e B = 1 e C = 1
- Assim podemos escrever a equação booleana de M:

$$\boldsymbol{M = \bar{A}\cdot B\cdot C + \bar{A}\cdot B\cdot C + A\cdot B\cdot \bar{C} + A\cdot B\cdot C}$$

Funções booleanas

- Exemplo: Expresse a equação booleana para a seguinte tabela verdade:

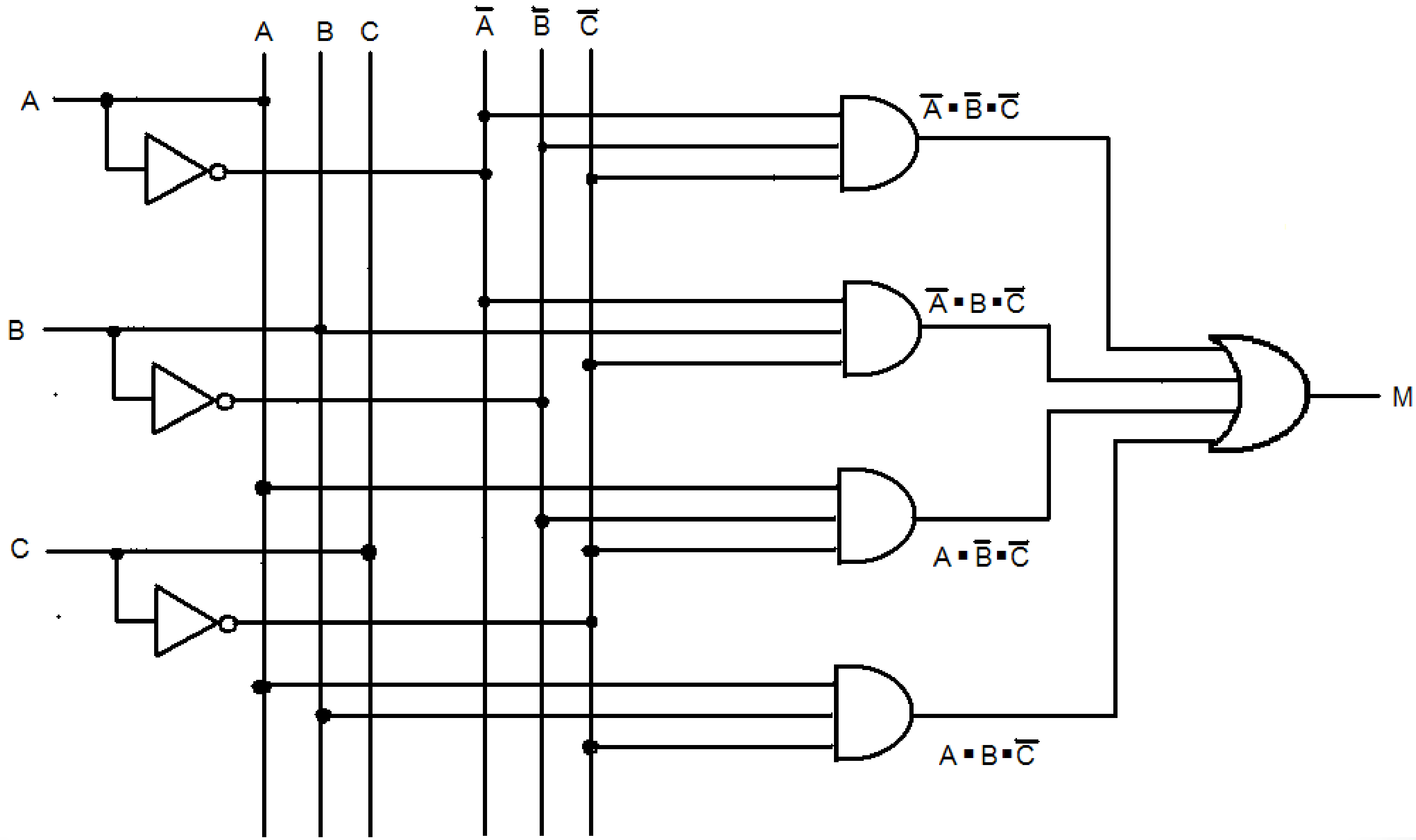
A	B	C	M	
0	0	0	1	$\overline{A}\cdot\overline{B}\cdot\overline{C}$
0	0	1	0	
0	1	0	1	$\overline{A}\cdot B\cdot\overline{C}$
0	1	1	0	
1	0	0	1	$A\cdot\overline{B}\cdot\overline{C}$
1	0	1	0	
1	1	0	1	$A\cdot B\cdot\overline{C}$
1	1	1	0	

$$M = \overline{A}\cdot\overline{B}\cdot\overline{C} + \overline{A}\cdot B\cdot\overline{C} + A\cdot\overline{B}\cdot\overline{C} + A\cdot B\cdot\overline{C}$$

Implementação de funções booleanas

- Uma vez que sabemos expressar equações booleanas, podemos implementar essas funções com as portas lógicas conhecidas.
- Para a equação: $M = \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot B \cdot \overline{C} + A \cdot \overline{B} \cdot \overline{C} + A \cdot B \cdot \overline{C}$ precisaremos de 4 portas lógicas AND de 3 entradas, uma porta lógica OU de 4 entradas e portas inversoras.
    - Por que apenas 4 portas AND??

Implementação de funções booleanas



Implementação de funções booleanas

- Regra para implementar um circuito:
  1. Escreva a tabela verdade para a função;
  2. Providencie inversores para gerar o complemento das entradas (ou seja, providencie portas not em cada entrada barrada);
  3. Desenhe uma porta AND para cada termo que tenha 1 na coluna de resultado;
  4. Ligue as portas AND às entradas adequadas;
  5. Alimente a saída de todas as portas AND a uma porta OR;

Precedência dos operadores

NOT: a primeira operação que deve ser feita é o NOT, exceto quando há outra operação dentro do NOT

AND: a segunda operação que deve ser feita é a AND, exceto quando há operações separadas por parênteses

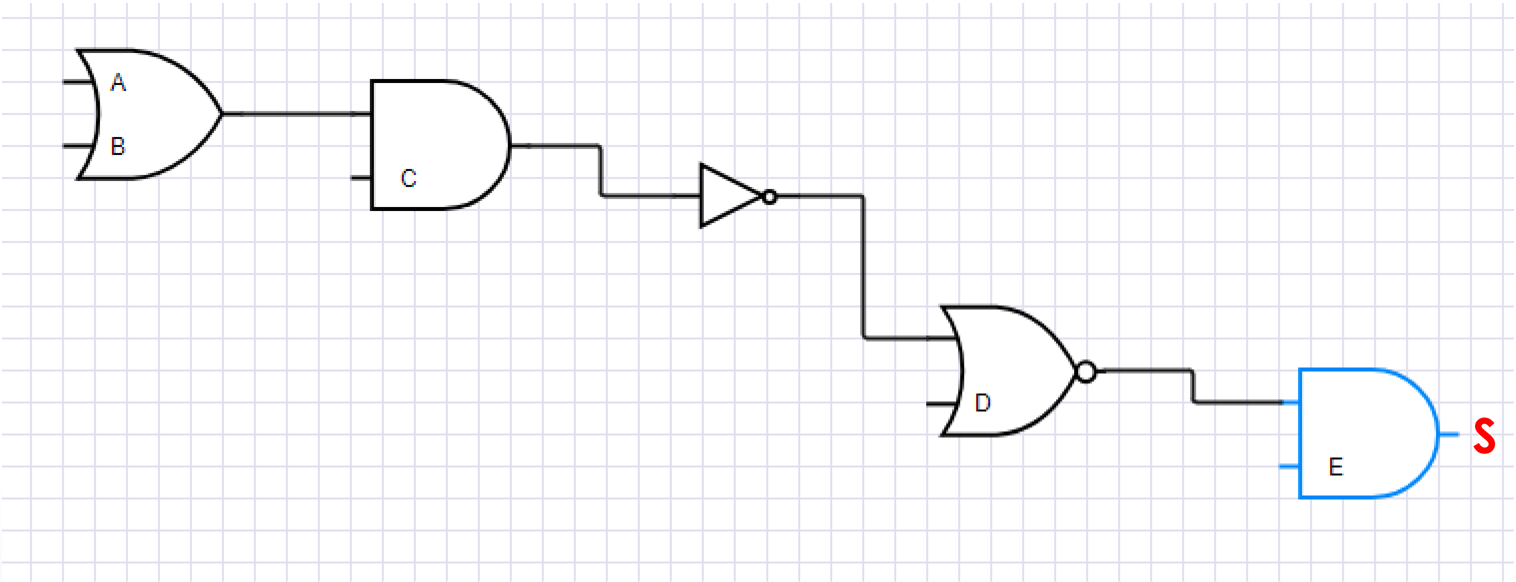
OR: a terceira operação que deve ser feita é a OR

Exemplos:

- $\bar{A} + B + C$: A operação NOT é a primeira a ser realizada.
  Exceção: $\overline{A.B} + B + C$: A operação AND é a primeira a ser realizada.
- $A.B + C$: A operação AND é a primeira a ser realizada.
  Exceção: $A.(B + C)$: A operação OR é a primeira a ser realizada.
- $A + B + C$: A operação OR é a primeira a ser realizada.

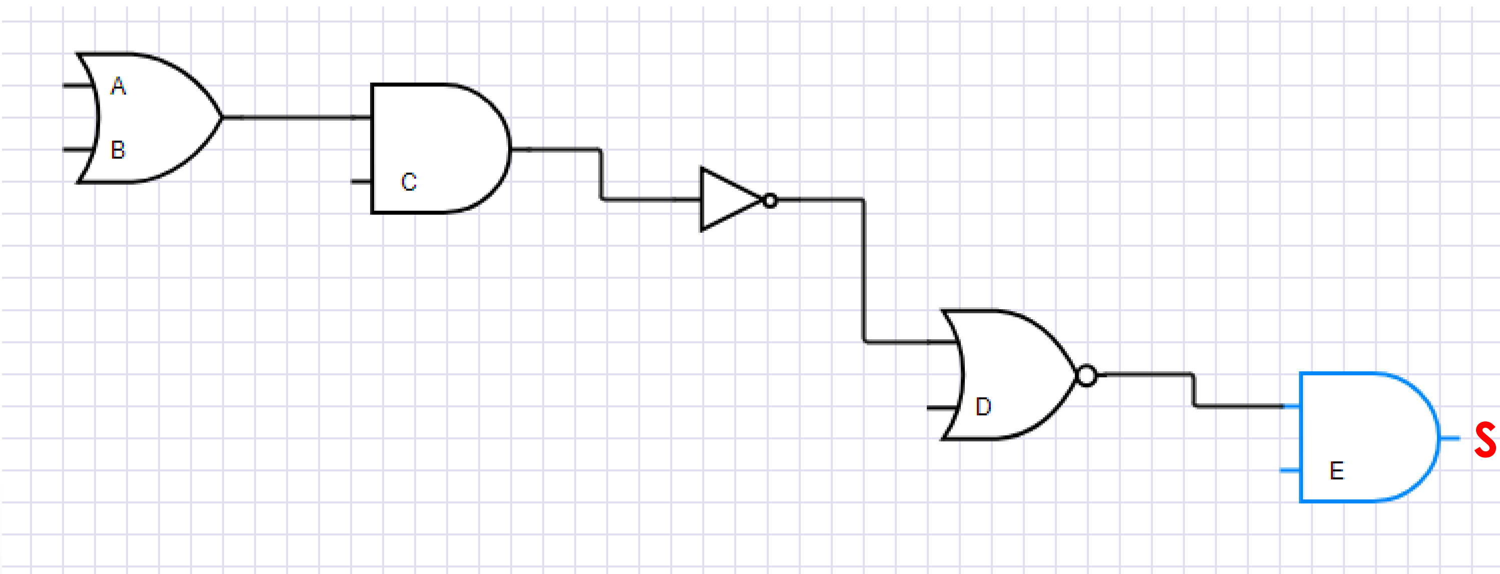
Exercício 1

- Qual a expressão booleana que representa o circuito abaixo?



Exercício 1

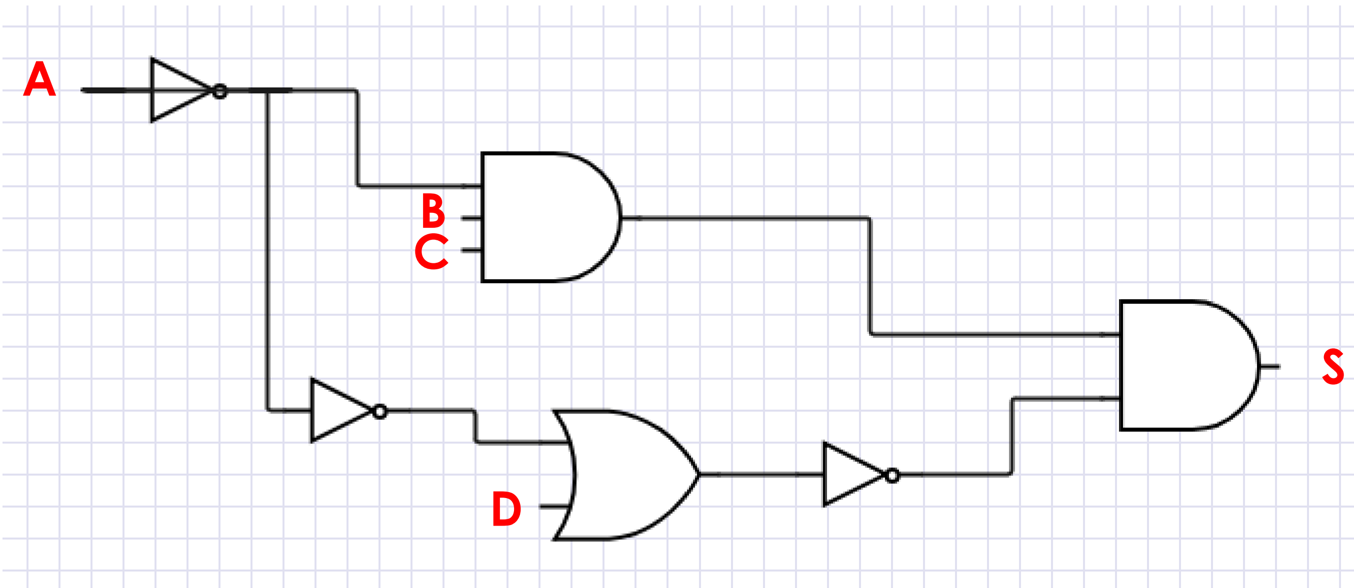
- Qual a expressão booleana que representa o circuito abaixo?



$$S=(\overline{(\overline{(A+B)\cdot C}) + D})\cdot E$$

Exercício 2

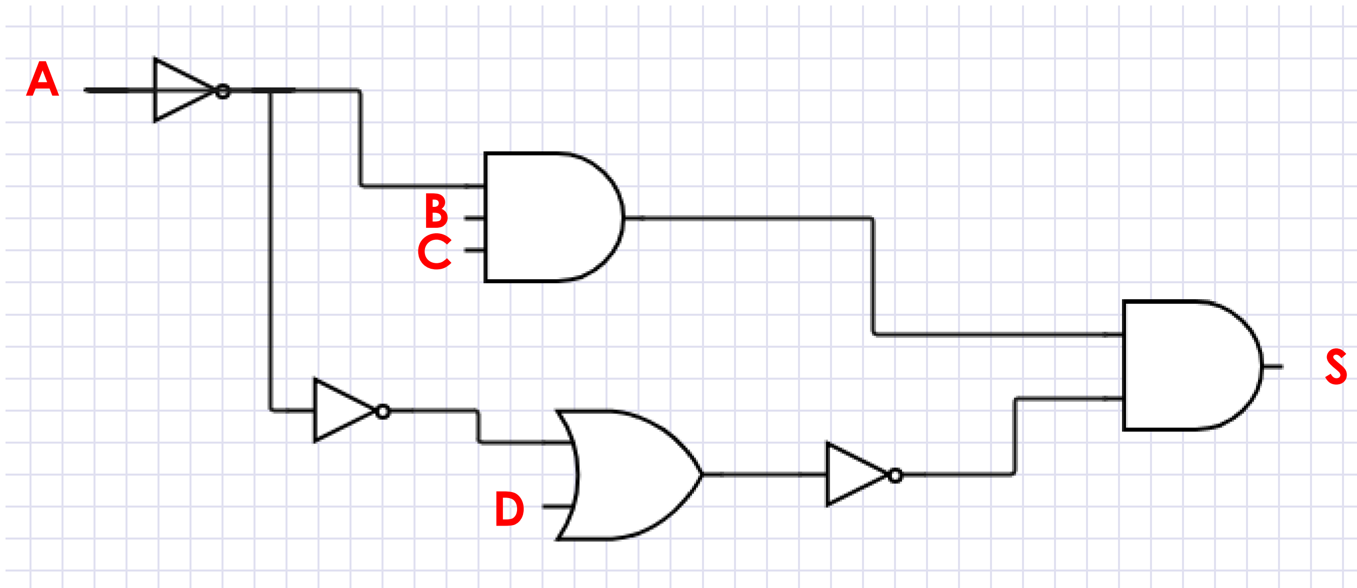
- Qual a expressão booleana que representa o circuito abaixo?



IMD0121

Exercício 2

- Qual a expressão booleana que representa o circuito abaixo?



$$S = (\bar{A}BC) \cdot \overline{(\bar{\bar{A}} + D)}$$

Exercício 3

- Qual o circuito que representa as expressões abaixo?

- **$S = A \cdot C + B \cdot \overline{C} + \overline{A} \cdot B \cdot C$**
- **$T = (\overline{A+B+\overline{C} \cdot D \cdot \overline{E}}) + \overline{B} \cdot C \cdot \overline{D}$**
- **$U = \overline{A \cdot B(C + D)}$**

Mintermos e Maxtermos

- Utilizando **Mintermos** e **Maxtermos**
- Mintermos
  - Conhecido como **soma de produtos**
  - É uma forma de escrever uma equação booleana na qual é utilizada uma porta OR com n entradas, e todas as entradas são saídas de portas AND

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

- Maxtermos
  - Conhecido como **produto de somas**
  - É uma forma de escrever expressões lógicas na qual é utilizada uma porta AND com n entradas, na qual todas as entradas são saídas de portas OR

$$X = (A + B + C).(\bar{B} + A).(C + \bar{D})$$

Conversão usando Mintermos

- Passos para conversão

1. Escreva o produto das variáveis, complementando-as, sempre que seu valor lógico seja ’0’ na linha correspondente (Exemplo: Linha 3 - 10 = $A.\overline{B}$)
2. Identifique as linhas da tabela que possuem saída igual a ’1’
3. Some os termos das linhas que possuem saída igual a ’1’

Tabela Verdade		
A	B	**X**
0	0	**1**
0	1	**0**
1	0	**1**
1	1	**1**

$$X = \bar{A}.\bar{B} + A.\bar{B} + A.B$$

Tabela Verdade			
A	B	**X**	Mintermos
0	0	**1**	$A.B$
0	1	**0**	$A.B$
1	0	**1**	$A.B$
1	1	**1**	$A.B$

Conversão usando Maxtermos

- Passos para conversão

1. Escreva a soma das variáveis, complementando-as, sempre que seu valor lógico seja ’1’ na linha correspondente (Exemplo: Linha 3 - 10 = A + $\overline{\text{B}}$)
2. Identifique as linhas da tabela que possuem saída igual a ’0’
3. Multiplique termos das linhas que possuem saída igual a ’0’

Tabela Verdade		
A	B	**X**
0	0	**1**
0	1	**0**
1	0	**1**
1	1	**1**

$X = A + \bar{B}$

Tabela Verdade			
A	B	**X**	Maxtermos
0	0	**1**	$A + B$
0	1	**0**	$A + B$
1	0	**1**	$A + B$
1	1	**1**	$A + B$

Exemplo

- Converta a tabela verdade em sua respectiva equação booleana utilizando mintermos e maxtermos, para descobrir a expressão lógica correspondente a ela.

Tabela Verdade			
A	B	C	**X**
0	0	0	**0**
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**

Exemplo - Resposta

Tabela Verdade					
A	B	C	**X**	Mintermos	Maxtermos
0	0	0	**0**		
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**		

Exemplo - Resposta

Tabela Verdade					
A	B	C	**X**	Mintermos	Maxtermos
0	0	0	**0**	$\bar{A}.\bar{B}.\bar{C}$	$A + B + C$
0	0	1	**1**	$\bar{A}.\bar{B}.C$	$A + B + \overline{C}$
0	1	0	**0**	$\bar{A}.B.\bar{C}$	$A + \overline{B} + C$
0	1	1	**0**	$\bar{A}.B.C$	$A + \overline{B} + \overline{C}$
1	0	0	**0**	$A.\bar{B}.\bar{C}$	$\overline{A} + B + C$
1	0	1	**1**	$A.\bar{B}.C$	$\overline{A} + B + \overline{C}$
1	1	0	**0**	$A.B.\bar{C}$	$\overline{A} + \overline{B} + C$
1	1	1	**1**	$A.B.C$	$\overline{A} + \overline{B} + \overline{C}$

Exemplo - Resposta

Tabela Verdade					
A	B	C	**X**	Mintermos	Maxtermos
0	0	0	**0**	$\bar{A}.\bar{B}.\bar{C}$	$A+B+C$
0	0	1	**1**	$\bar{A}.\bar{B}.C$	$A+B+\overline{C}$
0	1	0	**0**	$\bar{A}.B.\bar{C}$	$A+\overline{B}+C$
0	1	1	**0**	$\bar{A}.B.C$	$A+\overline{B}+\overline{C}$
1	0	0	**0**	$A.\bar{B}.\bar{C}$	$\overline{A}+B+C$
1	0	1	**1**	$A.\bar{B}.C$	$\overline{A}+B+\overline{C}$
1	1	0	**0**	$A.B.\bar{C}$	$\overline{A}+\overline{B}+C$
1	1	1	**1**	$A.B.C$	$\overline{A}+\overline{B}+\overline{C}$

- Mintermos: $X = \bar{A}.\bar{B}.C + A.\bar{B}.C + A.B.C$
- Maxtermos:
  $X = (A+B+C).(A+\bar{B}+C).(A+\bar{B}+\bar{C}).(\bar{A}+B+C).(\bar{A}+\bar{B}+C)$

Ferramentas de representação e simulação

- Logic Lab
  - http://www.neuroproductions.be/logic-lab/

- Circuits Cloud
  - http://circuits-cloud.com/

Referências

- **STALLINGS, William. Arquitetura e organização de computadores. 10. ed. São Paulo: Pearson, 2017. 814 p.**
  - **Capítulo 9**

- TOCCI, Ronald J; Widmer, Neal S. Sistemas Digitais: principios e Aplicações. 11. ed. São Paulo SP: Pearson, 2011, 817 p. ISBN 9788576050957
  - Capitulo 1

- PATTERSON, David A; HENNESSY, John L. Organização e projeto de computadores: A interface HARDWARE/SOFTWARE. Rio de Janeiro: Elsevier, 2005, 3ª edição.

Universidade Federal do Rio Grande do Norte
Instituto Metrópole Digital
IMD0121 –Arquitetura de Computadores

Funções Lógicas Básicas

Prof. Gustavo Girão
girao@imd.ufrn.br

Baseado no material do Prof. Ricardo Weber (UFRGS)