Eletrônica Digital

Notas de aula

Prof. Yago Pessanha Corrêa

Laboratório de Mecatrônica e Processamento de Sinais (MSP)
Instituto Federal de Educação, Ciência e Tecnologia Fluminense (IFFluminense)
Curso Técnico em Automação

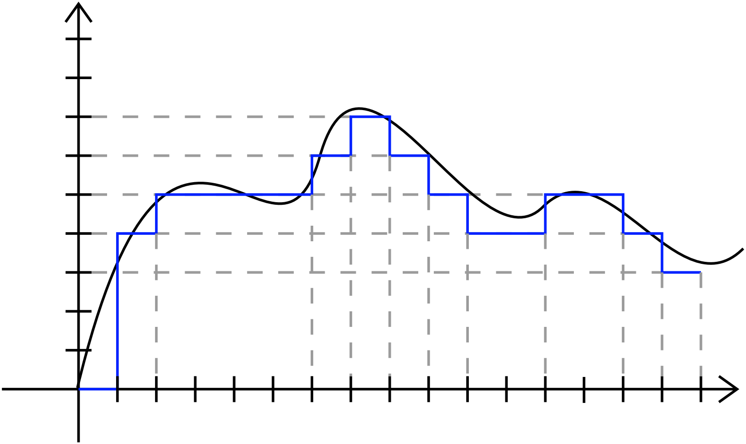
yago.correa@iff.edu.br

20 de maio de 2021

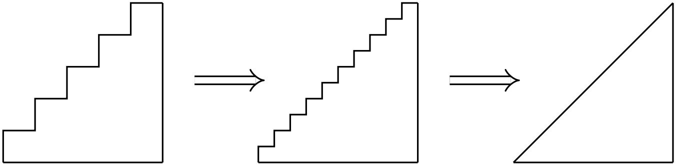
Sumário

Sumário

Analógico x digital



Analógico x digital



Grandezas digitais

Grandezas analógicas

Analógico x digital

Vantagens e desvantagens

Vantagens

- Sistemas digitais são mais simples de serem projetados.
- Maior facilidade em manter precisão e exatidão.
- Operações podem ser programadas.
- Menos afetados por ruídos.
- CI's (Circuitos Integrados) podem ser fabricados com mais dispositivos.

Vantagens e desvantagens

Desvantagens

- O mundo é quase todo analógico.
- Processar sinais digitais demanda tempo.

E no mundo real?

Importante

Como converter?

Sistemas de numeração

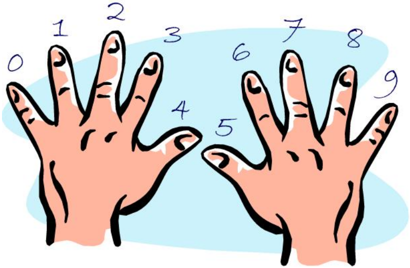
As quatro bases numéricas

- Decimal
- Binário
- Octal
- Hexadecimal

Sistemas de numeração

Sistema decimal

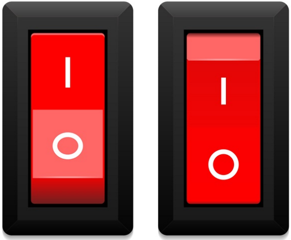
- Caracteres 0 a 9.
- Notação posicional – base 10.
- MSB x LSB.
- Exemplos de representação.



Sistemas de numeração

Sistema binário

- 2 dígitos.
- Notação posicional – base 2.
- MSB x LSB.
- Quantidade de dígitos para representar – como?.



Sistemas de numeração

Tipos de dados binários

Tipo de dado	Tamanho (em bits)
bit	1
nibble	4
byte	8
word	16
double word	32

Aritmética binária

Adição

$$
\begin{array}{r}
\,00\overset{1}{1}1 \\
+\,0101 \\
\hline
0110
\end{array}
$$

Aritmética binária

Subtração

$$
\begin{array}{r}
\,11 \\
\cancel{1}\,\cancel{1}011 \\
-\ 0\,1101 \\
\hline
0\,1110
\end{array}
$$

Aritmética binária

Multiplicação

$$
\begin{array}{rrl}
 & 101 & [\text{5 em decimal}] \\
 & \times\ 110 & [\text{6 em decimal}] \\
\hline
+ & 000 & [101 \times 0] \\
+ & 101 & [101 \times 1, \text{ movido para a esquerda}] \\
+ & 101 & [101 \times 1, \text{ movido para a esquerda duas posições}] \\
\hline
 & 11110 & [\text{30 em decimal}]
\end{array}
$$

Conversão binário-decimal

Procedimento

- Se dá pela soma das multiplicações de cada termo por sua potência de base 2 equivalente.

11001_2

$$
\begin{aligned}
1 \cdot 2^0 &= 1 \\
0 \cdot 2^1 &= 0 \\
0 \cdot 2^2 &= 0 \\
1 \cdot 2^3 &= 8 \\
1 \cdot 2^4 &= \underline{16} \\
& \ 25
\end{aligned}
$$

Conversão decimal-binário

Procedimento

- Método das divisões sucessivas

$$
\begin{array}{r|r|r|r}
12 & 2 & & \\
12 & 6 & 2 & \\
\hline
\color{red}{0} & 6 & 3 & 2 \\
 & \color{red}{0} & 2 & \color{red}{1} \\
 & & \color{red}{1} & \\
\end{array}
$$

$$\longleftarrow$$

$$12 = 1100_2$$

Conversão fração decimal-binário

Procedimento

- Método das multiplicações sucessivas

Número para converter para a base 2: 3,75.

$2 \times 3{,}75 = \boxed{7} + 0{,}5 \qquad \longrightarrow 7$

$2 \times 0{,}5 = \boxed{1} + 0 \qquad \longrightarrow 1$

$3{,}75 = [0{,}71]_2$

Exercícios

01. Converter os seguintes números binários para decimais:
- $11\,111_2$
- $1\,001\,100_2$
- $1011{,}11_2$
- $1100{,}0011_2$

02. Converter os seguintes números decimais para binários:
- 215
- 102
- 9,92
- 7,47

Referências e exercícios complementares

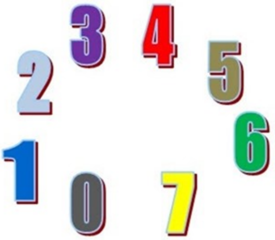
- IDOETA, Ivan V. e CAPUANO, Francisco G. Elementos de Eletrônica Digital. São Paulo: Editora Érica, ed. 40. 2008.

Página 36 - **1.6.1 até 1.6.5, 1.6.17 até 1.6.19**

Sistemas de numeração

Sistema octal

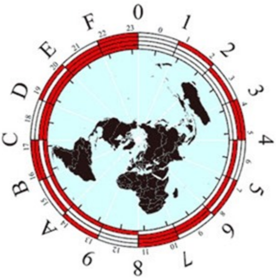
- Caracteres 0 a 7.
- CLP - gerenciamento de memórias.
- Novamente... notação posicional.
- Quantidade de dígitos para representar um número.



Sistemas de numeração

Sistema hexadecimal

- 16 caracteres... e agora, como representar?
- Novamente... notação posicional.
- Quantidade de dígitos para representar um número.
- Aplicações computacionais.



Conversão octal-decimal

Procedimento

- Se dá pela soma das multiplicações de cada termo por sua potência de base 8 equivalente.

331_8

$$\begin{aligned} 1 \cdot 8^0 &= 1 \\ 3 \cdot 8^1 &= 24 \\ 3 \cdot 8^2 &= 192 \\ \hline & 217 \end{aligned}$$

45_8

$$\begin{aligned} 5 \cdot 8^0 &= 5 \\ 4 \cdot 8^1 &= 32 \\ \hline & 37 \end{aligned}$$

Conversão decimal-octal

Procedimento

- Método das divisões sucessivas

$$
\begin{array}{r|c|c}
157 & 8 & \\
\cline{2-2}
152 & 19 & 8 \\
\cline{1-1}\cline{3-3}
\multicolumn{1}{c}{\color{red}5} & \multicolumn{1}{c}{16} & \multicolumn{1}{c}{\color{red}2} \\
\cline{2-2}
\multicolumn{1}{c}{} & \multicolumn{1}{c}{\color{red}3} & \multicolumn{1}{c}{}
\end{array}
$$

$$\longleftarrow$$

$$157 = 235_8$$

Conversão hexadecimal-decimal

Procedimento

- Se dá pela soma das multiplicações de cada termo por sua potência de base 16 equivalente.

$$A3_{16}$$

$$\begin{aligned} 3 \cdot 16^0 &= 3 \\ A \cdot 16^1 &= 160 \\ \hline &\ 163 \end{aligned}$$

Conversão decimal-hexadecimal

Procedimento

- Método das divisões sucessivas

$$
\begin{array}{r|c}
12412 & 16 \\ \cline{2-2}
12400 & \multicolumn{1}{c|}{775} \quad 16 \\ \hline
12 & 768 \quad \multicolumn{1}{c|}{48} \quad 16 \\ \cline{2-2}
 & 7 \quad 48 \quad 3 \\
 & 0
\end{array}
$$

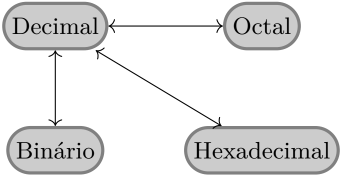
$\longleftarrow$

$$12412 = 30712_{16}$$

O que vimos até agora?

Até agora o ”centro das atenções” era o sistema **DECIMAL**

- decimal-binário-decimal
- decimal-octal-decimal
- decimal-hexadecimal-decimal



E como converter entre as outras bases?

Conversão binário-octal

Procedimento

- Basta separar em grupos de **3 bits** a partir da direita

$$100\,001\,111_8 = \underbrace{100} \quad \underbrace{001} \quad \underbrace{111}$$

$$4\ 1\ 7_2$$

Conversão binário-hexadecimal

Procedimento

- Basta separar em grupos de **4 bits** a partir da direita

$$11\,1010\,0110_2 = \underbrace{11}\quad\underbrace{1010}\quad\underbrace{0110}$$

$$3\;A\;6_{16}$$

Conversão octal-binário

Procedimento

- Observe cada dígito e transforme para binário

Octal	4	5	3	6
Binário	100	101	011	110
Final	100 101 011 110			

Conversão hexadecimal-binário

Procedimento

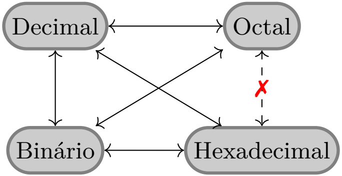
- Observe cada dígito e transforme para binário

Hexadecimal	1	A	6	0	
Binário	0001	1010	0110	0000	s
Final	0001 1010 0110 0000				

E as outras duas conversões?

Para completar só faltaram os métodos de conversão:

- Octal para hexadecimal
- Hexadecimal para octal



São métodos complicados, portanto, é melhor utilizar conversões intermediárias.

Exercícios

01. Complete a tabela de conversão abaixo:

Decimal	Binário	Octal	Hexadecimal
33			
	1110101		
		703	
			$1AF$

Referências e exercícios complementares

- IDOETA, Ivan V. e CAPUANO, Francisco G. Elementos de Eletrônica Digital. São Paulo: Editora Érica, ed. 40. 2008.

Página 37 - **1.6.6 até 1.6.16**

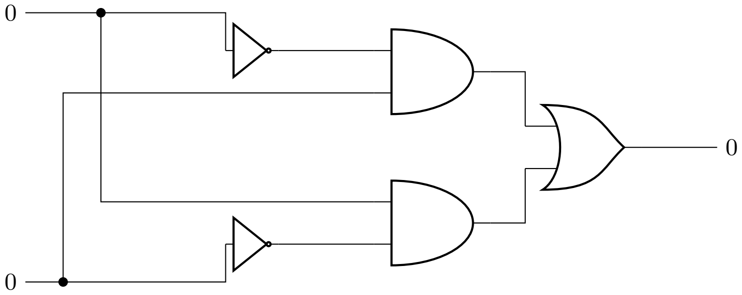
Sumário

Álgebra booleana

Notação

- "1" representa a classe de todos os objetos (o universo).
- "0" representa a classe a que nenhum objeto pertença (a classe vazia).

Portas lógicas - Introdução



Portas lógicas - Introdução

O que são portas lógicas?

- Blocos básicos.
- No mínimo: 1 entrada e 1 saída.
- Não possuem memória.
- Representação gráfica.
- Tabela verdade.

Nas funções lógicas há 2 valores:

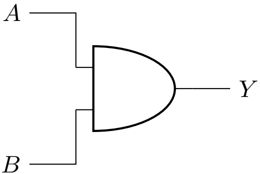
- 0 = Falso, sem tensão, baixo.
- 1 = Verdadeiro, com tensão, alto.

Portas lógicas - E (AND)

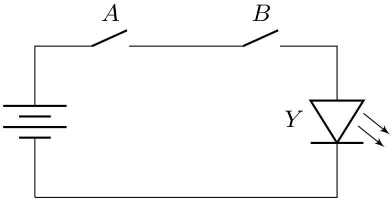
Notação

$$Y = A \cdot B$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



Portas lógicas - E (AND) - Circuito

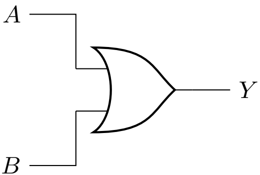


Portas lógicas - OU (OR)

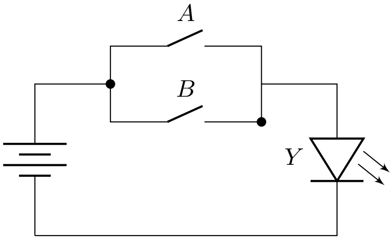
Notação

$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



Portas lógicas - OU (OR) - Circuito

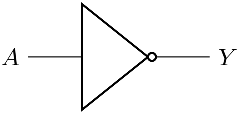


Portas lógicas - NÃO (NOT)

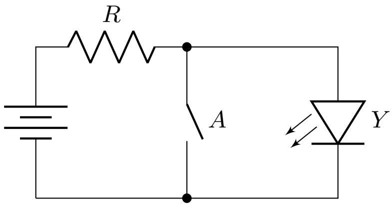
Notação

$$Y = \overline{A}$$

A	Y
1	0
0	1



Portas lógicas - NÃO (NOT) - Circuito

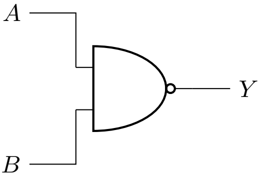


Portas lógicas - NÃO E (NAND)

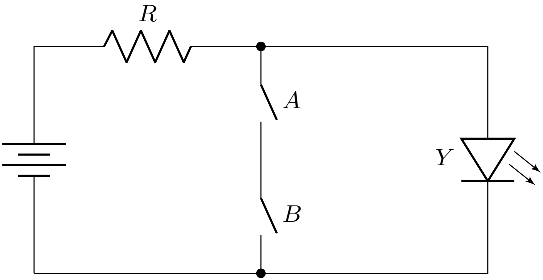
Notação

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

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



Portas lógicas - NÃO E (NAND) - Circuito

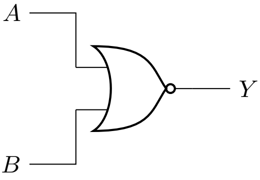


Portas lógicas - NÃO OU (NOR)

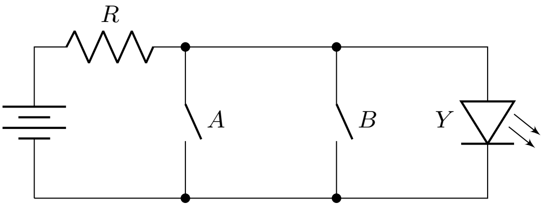
Notação

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

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



Portas lógicas - NÃO OU (NOR) - Circuito

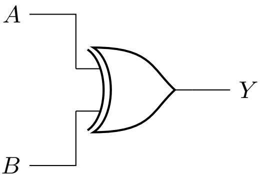


Portas lógicas - OU EXCLUSIVO (XOR)

Notação

$$Y = A \oplus B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

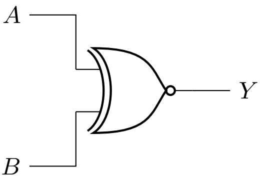


Portas lógicas - COINCIDÊNCIA (XNOR)

Notação

$$Y = \overline{A \oplus B} = A \odot B$$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1



Portas universais

NAND e NOR são portas universais

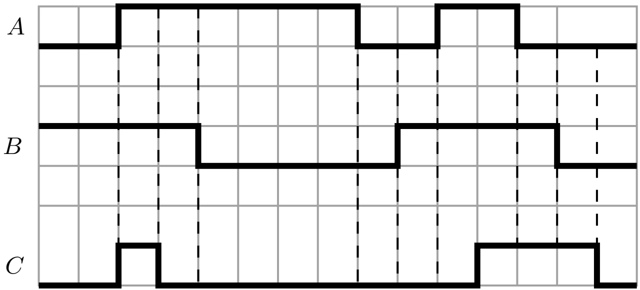
É possível obter a partir delas as portas:

- NOT
- AND
- OR

APOSTILA!

Exercícios

01. Desenhe a forma de onda de saída para uma porta NOR.
    - Repita para a entrada C mantida sempre em nível baixo.
    - Repita para a entrada C mantida sempre em nível alto.
02. Repita o problema para uma porta NAND.



Referências e exercícios complementares

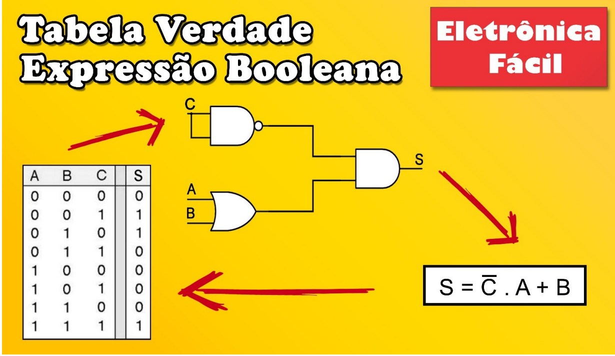
- IDOETA, Ivan V. e CAPUANO, Francisco G. Elementos de Eletrônica Digital. São Paulo: Editora Érica, ed. 40. 2008.

Página 82 - **2.9.1, 2.9.14, 2.9.18 até 2.9.22**

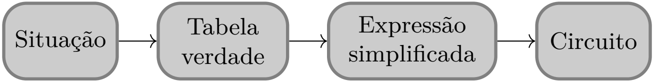
Sumário

O que vimos até agora?

- Circuitos Lógicos.
- Expressões booleanas.
- Tabela verdade.



Como projetar um circuito digital?

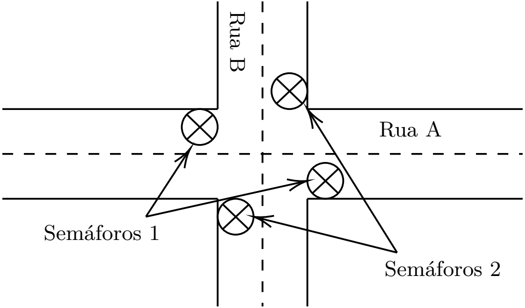


Resolução de projetos lógicos

Exemplo com duas variáveis

Caso tenha carro na(s)...

- Rua B
- Rua A
- Ruas A e B

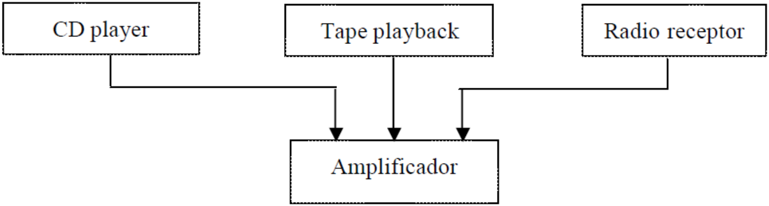


Resolução de projetos lógicos

Exemplo com três variáveis

Conexão de 3 aparelhos a um amplificador, obedecendo às prioridades:

- CD player
- Tape playback
- Radio receptor

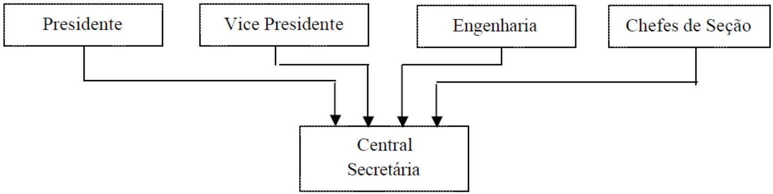


Resolução de projetos lógicos

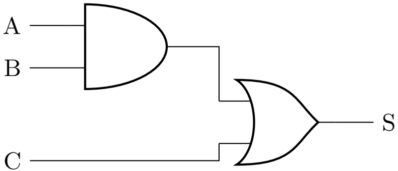
Exemplo com quatro variáveis

Conexão de 4 setores, via intercomunicadores, a central da Secretária, obedecendo às prioridades:

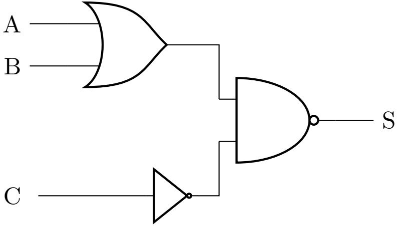
- Presidente
- Vice Presidente
- Engenharia
- Chefes de Seção



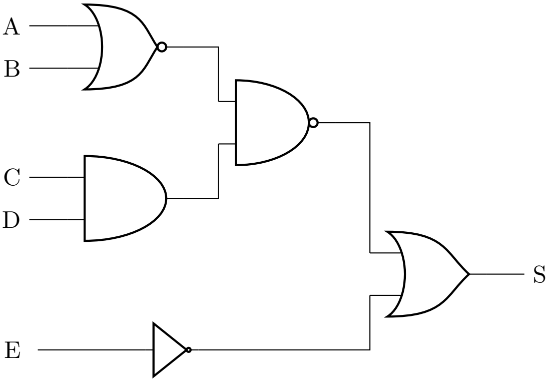
Obtendo a expressão a partir do circuito - Exemplo #01



Obtendo a expressão a partir do circuito - Exemplo #02



Obtendo a expressão a partir do circuito - Exemplo #03



Obtendo o circuito a partir da expressão

Hierarquia

- Parênteses.
- Bloco AND.
- Bloco OR.
- Negação.

Obtendo o circuito a partir da expressão

Exercícios

- $S = (A + B) \cdot C \cdot (B + D)$
- $Z = (\overline{A \cdot B} \oplus A) + (\overline{A} + B)$
- $Y = (A \cdot B) + \overline{C} \cdot D$
- $X = (A + B + C) \cdot \overline{(\overline{A} + C) \cdot (A + \overline{B})}$

Obtendo a tabela verdade a partir da expressão

Tabela verdade

A **tabela verdade** deve registrar **todas as possibilidades** de um dada expressão booleana, e pode ser obtida através dos seguintes passos:

1. Monte o quadro de possibilidades.
2. **n** colunas para as variáveis.
3. Use colunas auxiliares para os membros.
4. Monte uma coluna para o resultado final.

Obtendo a tabela verdade a partir da expressão

Exemplo #01

$$Z = (\overline{A \cdot B} \oplus A) + (\overline{A} + B)$$

A	B	$\overline{A \cdot B}$	$\overline{A \cdot B} \oplus A$	$\overline{A}$	$\overline{A} + B$	S
0	0	1	1	1	1	1
0	1	1	1	1	1	1
1	0	1	0	0	0	0
1	1	0	1	0	1	1

Obtendo a tabela verdade a partir da expressão

Exercícios

- $S = (A \cdot \overline{B} \cdot C) + (A \cdot \overline{D}) + (\overline{A} \cdot B \cdot D)$
- $Z = \overline{A} + B + A \cdot \overline{B} \cdot \overline{C}$
- $Y = C \oplus (\overline{A.B})$
- $X = A \cdot \overline{B} + \overline{A} \cdot B$

Obtendo a expressão a partir da tabela verdade

Situação mais utilizada

Lembre-se do problema inicial

- Situação.
- Tabela verdade.
- Expressão booleana.
- Circuito lógico.

MINTERMOS: saídas iguais a **1** - SoP

MAXTERMOS: saídas iguais a **0** - PoS

Obtendo a expressão a partir da tabela verdade

MINTERMOS

- Saídas iguais a **1** - SoP

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

Expressão final

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

Obtendo a expressão a partir da tabela verdade

MAXTERMOS

- Saídas iguais a **0** - PoS

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

Expressão final

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

Exercícios

01. A partir da tabela verdade abaixo encontre a expressão booleana por mintermos e maxtermos. Após isto, implemente ambos os circuitos digitais.

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

Referências e exercícios complementares

- IDOETA, Ivan V. e CAPUANO, Francisco G. Elementos de Eletrônica Digital. São Paulo: Editora Érica, ed. 40. 2008.

Página 82 - **2.9.2 até 2.9.13, 2.9.15 até 2.9.17**

Sumário

Por que simplificar?

Vantagens

- Menor número de portas lógicas.
- Mais simples, menor.
- Mais barato.

Postulados

P1.	Identidade	$A + 0 = A$	$A \cdot 1 = A$
P2.	Elemento nulo	$A + 1 = 1$	$A \cdot 0 = 0$
P3.	Equivalência	$A + A = A$	$A \cdot A = A$
P4.	Complemento	$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$
P5.	Involução	$\overline{\overline{A}} = A$	$\overline{\overline{A}} = A$

Propriedades

PR1.	Cumulativa	$A + B = B + A$
PR2.	Associativa	$(A + B) + C = A + (B + C)$
PR3.	Distributiva	$A + (B \cdot C) = (A + B) \cdot (A + C)$

PR1.	Cumulativa	$A \cdot B = B \cdot A$
PR2.	Associativa	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$
PR3.	Distributiva	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$

Teoremas

T1.	Absorção 1	$A + (A \cdot B) = A$
T2.	Absorção 2	$A + (\overline{A} \cdot B) = A + B$
T3.	De Morgan	$\overline{A + B} = \overline{A} \cdot \overline{B}$

T1.	Absorção 1	$A \cdot (A + B) = A$
T2.	Absorção 2	$A \cdot (\overline{A} + B) = A \cdot B$
T3.	De Morgan	$\overline{A.B} = \overline{A} + \overline{B}$

Exemplo #01

Resolução

$$
\begin{aligned}
S &= A \cdot B \cdot C + A \cdot \overline{C} + A \cdot \overline{B} \\
&\qquad\qquad\qquad\qquad \textit{Distributiva} \\
&= A \cdot (B \cdot C + \overline{C} + \overline{B}) \\
&\qquad\qquad\qquad\qquad \textit{Comutativa} \\
&= A \cdot (B \cdot C + \overline{B} + \overline{C}) \\
&\qquad\qquad\qquad\qquad \textit{De Morgan} \\
&= A \cdot (B \cdot C + \overline{B \cdot C}) \\
&\qquad\qquad\qquad\qquad \textit{Complemento} \\
&= A \cdot (1) \\
&\qquad\qquad\qquad\qquad \textit{Identidade} \\
&= A
\end{aligned}
$$

Exemplo #02

Resolução

$$
\begin{aligned}
S &= \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot B \cdot \overline{C} + A \cdot \overline{B} \cdot C \\
&\qquad\qquad\qquad\qquad \mathit{Distributiva} \\
&= \overline{A} \cdot \overline{C} \cdot (\overline{B} + B) + A \cdot \overline{B} \cdot C \\
&\qquad\qquad\qquad\qquad \mathit{Complemento} \\
&= \overline{A} \cdot \overline{C} \cdot (1) + A \cdot \overline{B} \cdot C \\
&\qquad\qquad\qquad\qquad \mathit{Identidade} \\
&= \overline{A} \cdot \overline{C} + A \cdot \overline{B} \cdot C
\end{aligned}
$$

Exemplo #03

Resolução #01 - Parte 1

$$
\begin{aligned}
S &= A \cdot B + A \cdot \overline{B} \cdot C \\
& \qquad\qquad\qquad\qquad \textit{Identidade} \\
&= A \cdot B \cdot 1 + A \cdot \overline{B} \cdot C \\
& \qquad\qquad\qquad\qquad \textit{Elemento Nulo} \\
&= A \cdot B \cdot (1 + C) + A \cdot \overline{B} \cdot C \\
& \qquad\qquad\qquad\qquad \textit{Distributiva} \\
&= A \cdot B \cdot 1 + A \cdot B \cdot C + A \cdot \overline{B} \cdot C \\
& \qquad\qquad\qquad\qquad \textit{Identidade} \\
&= A \cdot B + A \cdot B \cdot C + A \cdot \overline{B} \cdot C
\end{aligned}
$$

Exemplo #03

Resolução #01 - Parte 2

$$
\begin{aligned}
S &= A \cdot B + A \cdot B \cdot C + A \cdot \overline{B} \cdot C \\
& \qquad\qquad\qquad\qquad \textit{Distributiva} \\
&= A \cdot B + A \cdot C \cdot (B + \overline{B}) \\
& \qquad\qquad\qquad\qquad \textit{Complemento} \\
&= A \cdot B + A \cdot C \cdot (1) \\
& \qquad\qquad\qquad\qquad \textit{Identidade} \\
&= A \cdot B + A \cdot C
\end{aligned}
$$

Exemplo #03

Resolução #02

- Alternativamente, podemos resolver a mesma questão em menos etapas usando a *absorção*.

$$
\begin{aligned}
S &= A \cdot B + A \cdot \overline{B} \cdot C \\
&\quad\textit{Distributiva} \\
&= A \cdot (B + \overline{B} \cdot C) \\
&\quad\textit{Absorção 1} \\
&= A \cdot (B + C)
\end{aligned}
$$

Exercícios

01. Simplifique as expressões booleanas a seguir:

- $\overline{A + \overline{B} \cdot C}$
- $\overline{(\overline{A} + C) \cdot (B + \overline{D})}$
- $(\overline{A} + B) \cdot (A + B + D) \cdot \overline{D}$
- $A \cdot \overline{B} \cdot C + A \cdot \overline{B} \cdot \overline{C}$
- $(\overline{A} + B) \cdot (A + B)$

Referências e exercícios complementares

- IDOETA, Ivan V. e CAPUANO, Francisco G. Elementos de Eletrônica Digital. São Paulo: Editora Érica, ed. 40. 2008.

Página 148 - **3.10.1 até 3.10.7**

Sumário

Mapa de Karnaugh

Mapeamento da tabela verdade

- Circuito mínimo.
- Fácil utilização.

K-map - Duas variáveis

		A	
		0	1
B	0		
	1		

K-map - Três variáveis

		AB			
		00	01	11	10
C	0				
	1				

K-map - Quatro variáveis

$CD \backslash AB$	00	01	11	10
00				
01				
11				
10				

Mapa de Karnaugh x Tabela verdade

Atenção à ORDEM

- **Troca de 10 e 11**.
- Em células adjacentes apenas uma variável pode mudar de valor.

Como montar?

Montagem do K-map

Podemos utilizar mintermos ou maxtermos.

Passo-a-passo:

1. Preencher as células de acordo com a tabela verdade.
2. Agrupar os 1's (mintermos) ou 0's (maxtermos):
    1. Unir na horizontal ou vertical.
    2. **NUNCA** na diagonal.
    3. **PODE USAR** os cantos.
    4. Deve ser múltiplos da **BASE 2**.
    5. **MAIOR AGRUPAMENTO POSSÍVEL**.

Como agrupar? - Adjacências

		A	
		0	1
B	0	0	0
	1	1	1

Como agrupar? - Adjacências

		A	
		0	1
B	0	0	0
	1	1	1

Como agrupar? - Adjacências

		AB 00	01	11	10
C	0	0	0	1	1
	1	1	1	0	0

Como agrupar? - Adjacências

C \ AB	00	01	11	10
0	0	0	1	1
1	1	1	0	0

Como agrupar? - Linhas ou colunas

$CD \backslash AB$	00	01	11	10
00	1	1	1	1
01	1	0	0	0
11	1	0	0	0
10	1	0	0	0

Como agrupar? - Linhas ou colunas

CD \ AB	00	01	11	10
00	1	1	1	1
01	1	0	0	0
11	1	0	0	0
10	1	0	0	0

Como agrupar? - Quadra

CD \ AB	00	01	11	10
00	1	1	0	0
01	1	1	0	0
11	0	0	1	1
10	0	0	1	1

Como agrupar?

$CD \backslash AB$	00	01	11	10
00	1	1	0	0
01	1	1	0	0
11	0	0	1	1
10	0	0	1	1

Como agrupar? - Quadra em anel

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

Como agrupar? - Quadra em anel

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

Como agrupar? - Quadra em anel

$CD \backslash AB$	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

Como agrupar? - Quadra em anel

CD \ AB	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

Como agrupar? - Oitava

$CD \backslash AB$	00	01	11	10
00	0	1	1	0
01	1	1	1	1
11	1	1	1	1
10	0	1	1	0

Como agrupar? - Oitava

CD \ AB	00	01	11	10
00	0	1	1	0
01	1	1	1	1
11	1	1	1	1
10	0	1	1	0

Diagramas com condições irrelevantes

CD \ AB	00	01	11	10
00	X	0	X	1
01	1	0	1	1
11	0	X	X	0
10	0	1	0	X

Diagramas com condições irrelevantes

CD \ AB	00	01	11	10
00	X	0	X	1
01	1	0	1	1
11	0	X	X	0
10	0	1	0	X

Obtendo a expressão simplificada - Exemplo #01

- A variável que **muda de estado** é **retirada**.

$$S = A \cdot \overline{B} + A \cdot B + \overline{A} \cdot B$$

B \ A	0	1
0	0	1
1	1	1

Expressão simplificada

$$S = \color{red}{A} + \color{green}{B}$$

Obtendo a expressão simplificada - Exemplo #02

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

CD \ AB	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	0	0

Expressão simplificada

$$S = A \cdot B$$

Obtendo a expressão simplificada - Exemplo #03

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

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

Expressão simplificada

$$S = \overline{B} \cdot \overline{D}$$

Obtendo a expressão simplificada - Exemplo #03

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

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

Expressão simplificada

$$S = \overline{B} \cdot \overline{D}$$

Obtendo a expressão simplificada - Exemplo #04

A	B	C	D	S
0	0	0	0	X
0	0	0	1	0
0	0	1	0	1
0	0	1	1	X
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	X
1	0	1	1	0
1	1	0	0	0
1	1	0	1	X
1	1	1	0	0
1	1	1	1	X

Obtendo a expressão simplificada - Exemplo #04

CD \ AB	00	01	11	10
00	X	1	0	0
01	0	0	X	1
11	X	1	X	0
10	1	1	0	X

Expressão simplificada

$$S = \overline{A} \cdot C + A \cdot \overline{C} \cdot D + \overline{A} \cdot \overline{D}$$

Obtendo a expressão simplificada - Exemplo #05

$CD \backslash AB$	00	01	11	10
00	0	0	1	0
01	1	1	1	1
11	1	1	0	0
10	0	0	0	0

Expressão simplificada

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

Exercícios

01. Simplifique as expressões booleanas a seguir pelo Mapa de Karnaugh:

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

Referências e exercícios complementares

- IDOETA, Ivan V. e CAPUANO, Francisco G. Elementos de Eletrônica Digital. São Paulo: Editora Érica, ed. 40. 2008.

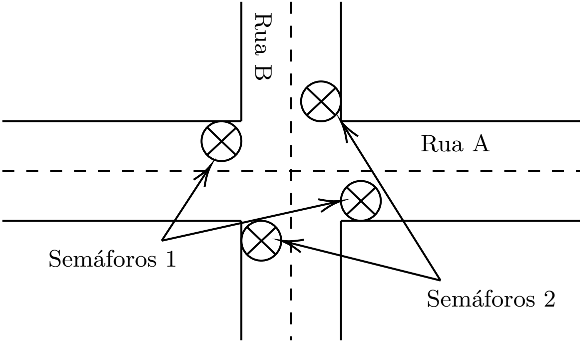
Página 149 - **3.10.9 até 3.10.15**

Sumário

Relembrando...

Situação $\rightarrow$ Tabela verdade $\rightarrow$ Expressão simplificada $\rightarrow$ Circuito

Circuitos combinacionais com 2 variáveis - Exemplo #01



Circuitos combinacionais com 2 variáveis - Exemplo #01

Deseja-se instalar, no cruzamento, um sistema automático de semáforos, com as seguintes características:

- Quando houver carros transitando somente na rua B, o semáforo 2 deverá permanecer verde para os carros trafegarem livremente.
- Igualmente, quando houver carros transitando somente na rua A, o semáforo 1 deverá permanecer verde.
- Quando houver carros transitando em ambas as ruas, o semáforo da rua A deve ficar verde, pois é a rua preferencial.

Circuitos combinacionais com 2 variáveis - Exemplo #01

A	B	G1	R1	G2	R2
0	0				
0	1				
1	0				
1	1				

Circuitos combinacionais com 2 variáveis - Exemplo #01

A	B	G1	R1	G2	R2
0	0				
0	1				
1	0				
1	1				

Circuitos combinacionais com 2 variáveis - Exemplo #01

A	B	G1	R1	G2	R2
0	0	X	X	X	X
0	1				
1	0				
1	1				

Circuitos combinacionais com 2 variáveis - Exemplo #01

A	B	G1	R1	G2	R2
0	0	X	X	X	X
0	1				
1	0				
1	1				

Circuitos combinacionais com 2 variáveis - Exemplo #01

A	B	G1	R1	G2	R2
0	0	X	X	X	X
0	1	0	1	1	0
1	0				
1	1				

Circuitos combinacionais com 2 variáveis - Exemplo #01

A	B	G1	R1	G2	R2
0	0	X	X	X	X
0	1	0	1	1	0
1	0				
1	1				

Circuitos combinacionais com 2 variáveis - Exemplo #01

A	B	G1	R1	G2	R2
0	0	X	X	X	X
0	1	0	1	1	0
1	0	1	0	0	1
1	1				

Circuitos combinacionais com 2 variáveis - Exemplo #01

A	B	G1	R1	G2	R2
0	0	X	X	X	X
0	1	0	1	1	0
1	0	1	0	0	1
1	1				

Circuitos combinacionais com 2 variáveis - Exemplo #01

A	B	G1	R1	G2	R2
0	0	X	X	X	X
0	1	0	1	1	0
1	0	1	0	0	1
1	1	1	0	0	1

Circuitos combinacionais com 2 variáveis - Exemplo #01

A	B	G1	R1	G2	R2
0	0	X	X	X	X
0	1	0	1	1	0
1	0	1	0	0	1
1	1	1	0	0	1

Circuitos combinacionais com 2 variáveis - Exemplo #01

B \ A	0	1
0	X	1
1	0	1

Expressão simplificada

$$G1 = A$$

Circuitos combinacionais com 2 variáveis - Exemplo #01

B \ A	0	1
0	X	0
1	1	0

Expressão simplificada

$$R1 = \overline{A}$$

Circuitos combinacionais com 2 variáveis - Exemplo #01

B \ A	0	1
0	X	0
1	1	0

Expressão simplificada

$$G2 = \overline{A}$$

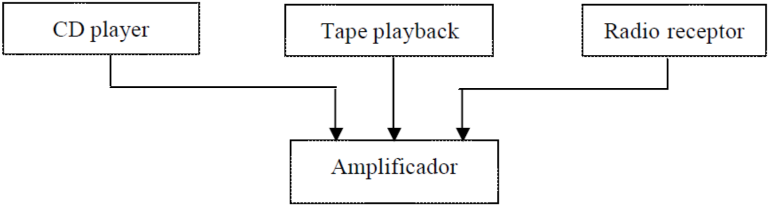
Circuitos combinacionais com 2 variáveis - Exemplo #01

		A	
		0	1
B	0	X	1
	1	0	1

Expressão simplificada

$$R2 = A$$

Circuitos combinacionais com 3 variáveis - Exemplo #02



Circuitos combinacionais com 3 variáveis - Exemplo #02

Deseja-se instalar um amplificador para ligar três aparelhos, com as seguintes prioridades:

- Prioridade 1: CD player
- Prioridade 2: Tape playback
- Prioridade 3: Rádio receptor

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1			
1	1	0			
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1			
1	1	0			
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0			
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0			
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1			

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	1	0	0

Circuitos combinacionais com 3 variáveis - Exemplo #02

A	B	C	SA	SB	SC
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	1	0	0

Circuitos combinacionais com 3 variáveis - Exemplo #02

C \ AB	00	01	11	10
0	X	0	1	1
1	0	0	1	1

Expressão simplificada

$$SA = A$$

Circuitos combinacionais com 3 variáveis - Exemplo #02

		AB			
		00	01	11	10
C	0	X	1	0	0
	1	0	1	0	0

Expressão simplificada

$$SB = \overline{A} \cdot B$$

Circuitos combinacionais com 3 variáveis - Exemplo #02

C \ AB	00	01	11	10
0	X	0	0	0
1	1	0	0	0

Expressão simplificada

$$SC = \overline{A} \cdot \overline{B}$$

Exercícios

01. Quatro juízes participam de um programa de calouros e cada um tem a sua disposição, uma chave liga/desliga correspondendo ao julgamento do candidato: aprovado ou reprovado. Na saída existem três lâmpadas, correspondentes a três resultados: aprovado (pela maioria), reprovado (pela maioria) ou empate.

02. Um motor deve funcionar quando uma ou mais das seguintes condições forem satisfeitas: **(1)** Regime de carga $\geqslant 80\%$ e Temperatura $> 25\,°\mathrm{C}$; **(2)** Regime de carga $< 80\%$, Umidade relativa $> 60\%$ e Temperatura $> 25\,°\mathrm{C}$; **(3)** Regime de carga $< 80\%$, no período de carga entre as 15 h e 16 h; **(4)** Temperatura $> 25\,°\mathrm{C}$, fora do período de carga entre as 15 h e 16 h.

03. Projete um circuito digital com 4 variáveis de entrada, que indique quando há um número primo presente na entrada.

Referências e exercícios complementares

- IDOETA, Ivan V. e CAPUANO, Francisco G. Elementos de Eletrônica Digital. São Paulo: Editora Érica, ed. 40. 2008.

Página 174 - **4.3.1 até 4.3.8**

Códigos binários - BCD 8421

BCD 8241

- BCD: *binary coded decimal.*
- Técnica mais simples e normal de ser utilizada – **natural**.
- Cada dígito binário é representado pelo seu correspondente binário de 4 dígitos.
- 8421 indica os ”pesos” de cada algarismo no sistema binário.
- A grande vantagem do sistema BCD 8421 é a facilidade entre homem-máquina.
- O código BCD, no entanto, é menos eficiente que o código binário puro, pois são usados **mais bits** para se representar um determinado valor.

Códigos binários - BCD 8421

Decimal	BCD 8421			
	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

- 10=0001 0000
- 11=0001 0001
- ...
- 15=0001 0101

Códigos binários - BCD 7421

Decimal	BCD 7421			
	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	1	0	0	0
8	1	0	0	1
9	1	0	1	0

Códigos binários - Excesso 3

- Transformação do número decimal no binário correspondente somando-se 3 unidades.
- É obtido adiantando o código BCD três vezes.
- Neste modelo há apenas de zero a nove decimal.
- Apresenta algumas vantagens nas operações matemáticas - utilizado em circuitos aritméticos.

Códigos binários - Excesso 3

Decimal	Excesso 3			
	A	B	C	D
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Códigos binários - Gray

- De um número para o outro apenas **um bit** varia.
- A cada linha o número binário é variado em um algarismo de forma que não se repita nenhum anterior.
- Utilizado no K-map.
- Sua estrutura facilita a detecção de erros.

Códigos binários - Gray

Decimal	Gray			
	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

Códigos binários - Johnson

- Possui cinco dígitos binários.
- Cada código possui apenas um bit diferente do seu sucessor.
- Sua vantagem é a facilidade de gerar palavras código.

Códigos binários - Johnson

Decimal	Johnson				
	A	B	C	D	E
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	1
3	0	0	1	1	1
4	0	1	1	1	1
5	1	1	1	1	1
6	1	1	1	1	0
7	1	1	1	0	0
8	1	1	0	0	0
9	1	0	0	0	0

Códigos binários - 9876543210

- Decodificação de “uma saída de 10”.
- Um único bit será 1, enquanto os demais serão 0
- O valor 1 assume a posição correspondente ao número decimal

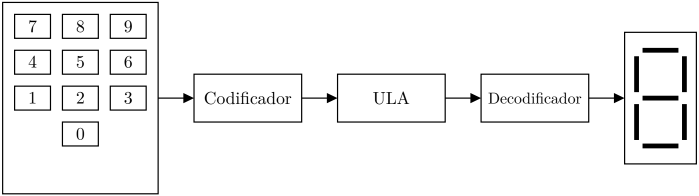
Códigos binários - 9876543210

Decimal	9876543210									
	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	1	0	0
3	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	1	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0

Codificadores e decodificadores

- **Codificador** é um circuito combinacional que torna possível a passagem de um código conhecido para um desconhecido.
- Por outro lado, **decodificador** faz a operação inversa: passa um código desconhecido para um conhecido.

Codificadores e decodificadores

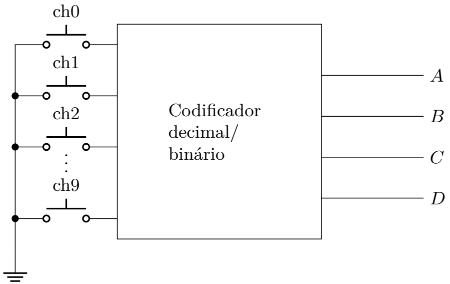


Atenção:

- Depende do referencial.
- Para o usuário:
  - Sistema de entrada: codificador.
  - Sistema de saída: decodificador.
- Para a máquina:
  - Sistema de entrada: decodificador.
  - Sistema de saída: codificador.

Codificadores e decodificadores - Projeto #01 - calculadora

- Primeira parte: codificador decimal-binário



Codificador decimal-binário

- Entrada: chaves de 0 a 9
- Saída: 4 bits em BCD 8421
- Chave fechada equivale a nível 0 - problema TTL.

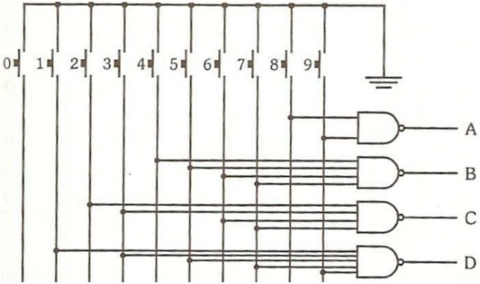
Codificadores e decodificadores - Projeto #01 - calculadora

Decimal	BCD 8421			
	A	B	C	D
Ch0	0	0	0	0
Ch1	0	0	0	1
Ch2	0	0	1	0
Ch3	0	0	1	1
Ch4	0	1	0	0
Ch5	0	1	0	1
Ch6	0	1	1	0
Ch7	0	1	1	1
Ch8	1	0	0	0
Ch9	1	0	0	1

Codificadores e decodificadores - Projeto #01 - calculadora

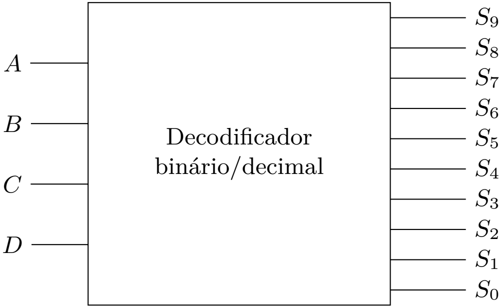
Lembre-se:

- Chave fechada equivale a nível 0 - problema TTL.
- Utilização de uma porta NAND em cada saída
- Fornece nível 1 quando qualquer entrada for 0 (chave fechada).



Codificadores e decodificadores - Projeto #01 - calculadora

- Segunda parte: decodificador binário-decimal



Decodificador decimal-binário

- Entrada: 4 bits em BCD 8421
- Saída: Código 9876543210

Codificadores e decodificadores - Projeto #01 - calculadora

BCD 8421				Código 9876543210									
A	B	C	D	S_9	S_8	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0

Codificadores e decodificadores - Projeto #01 - calculadora

Agora...

- Monta-se o Mapa de Karnaugh para cada uma das saídas da tabela.
- Codigo BCD 8421 só vai até o 9 - demais casos: condição irrelevante

Codificadores e decodificadores - Projeto #01 - calculadora

$AB \backslash CD$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	X	X	X	X
10	0	1	X	X

Expressão simplificada

$$S_9 = A \cdot D$$

Codificadores e decodificadores - Projeto #01 - calculadora

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	X	X	X	X
10	1	0	X	X

Expressão simplificada

$$S_8 = A \cdot \overline{D}$$

Codificadores e decodificadores - Projeto #01 - calculadora

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	0	0	X	X

Expressão simplificada

$$S_7 = B \cdot C \cdot D$$

Codificadores e decodificadores - Projeto #01 - calculadora

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	X	X	X	X
10	0	0	X	X

Expressão simplificada

$$S_6 = B \cdot C \cdot \overline{D}$$

Codificadores e decodificadores - Projeto #01 - calculadora

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	1	0	0
11	X	X	X	X
10	0	0	X	X

Expressão simplificada

$$S_5 = B \cdot \overline{C} \cdot D$$

Codificadores e decodificadores - Projeto #01 - calculadora

CD

AB	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	X	X	X	X
10	0	0	X	X

Expressão simplificada

$$S_4 = B \cdot \overline{C} \cdot \overline{D}$$

Codificadores e decodificadores - Projeto #01 - calculadora

$AB \backslash CD$	00	01	11	10
00	0	0	1	0
01	0	0	0	0
11	X	X	X	X
10	0	0	X	X

Expressão simplificada

$$S_3 = \overline{B} \cdot C \cdot D$$

Codificadores e decodificadores - Projeto #01 - calculadora

AB \ CD	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	X	X	X	X
10	0	0	X	X

Expressão simplificada

$$S_2 = \overline{B} \cdot C \cdot \overline{D}$$

Codificadores e decodificadores - Projeto #01 - calculadora

CD

AB \ CD	00	01	11	10
00	0	1	0	0
01	0	0	0	0
11	X	X	X	X
10	0	0	X	X

Expressão simplificada

$$S_1 = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D$$

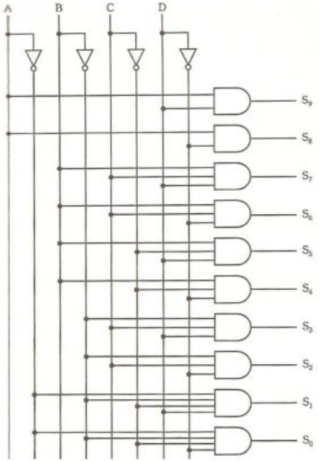
Codificadores e decodificadores - Projeto #01 - calculadora

$AB \backslash CD$	00	01	11	10
00	$\boxed{1}$	0	0	0
01	0	0	0	0
11	X	X	X	X
10	0	0	X	X

Expressão simplificada

$$S_0 = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}$$

Codificadores e decodificadores - Projeto #01 - calculadora



Codificadores e decodificadores - Projeto #02 - conversor BCD 8421 para Excesso 3

Situação

- 2 códigos.
- Construir a tabela verdade que os relaciona.
- Simplificar via K-Map.

Conversor BCD 8421 para Excesso 3

- Entrada: código BCD 8421.
- Saída: código Excesso 3.

Codificadores e decodificadores - Projeto #02 - conversor BCD 8421 para Excesso 3

BCD 8421				Excesso 3			
A	B	C	D	S_3	S_2	S_1	S_0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

Codificadores e decodificadores - Projeto #02 - conversor BCD 8421 para Excesso 3

$AB \backslash CD$	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

Expressão simplificada

$$S_3 = B \cdot D + B \cdot C + A$$

Codificadores e decodificadores - Projeto #02 - conversor BCD 8421 para Excesso 3

$AB \backslash CD$	00	01	11	10
00	0	1	1	1
01	1	0	0	0
11	X	X	X	X
10	0	1	X	X

Expressão simplificada

$$S_2 = B \cdot \overline{C} \cdot \overline{D} + \overline{B} \cdot C + \overline{B} \cdot D$$

Codificadores e decodificadores - Projeto #02 - conversor BCD 8421 para Excesso 3

AB \ CD	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	X	X	X	X
10	1	0	X	X

Expressão simplificada

$$S_1 = \overline{C} \cdot \overline{D} + C \cdot D$$

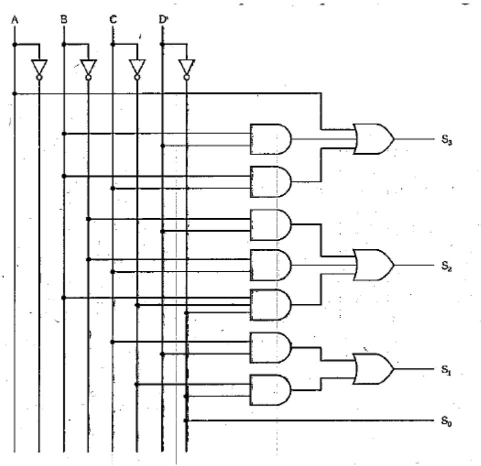
Codificadores e decodificadores - Projeto #02 - conversor BCD 8421 para Excesso 3

$AB \backslash CD$	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	X	X	X	X
10	1	0	X	X

Expressão simplificada

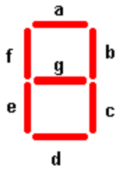
$$S_0 = \overline{D}$$

Codificadores e decodificadores - Projeto #02 - conversor BCD 8421 para Excesso 3



Exercícios

01. Um **Display de 7 segmentos** pode ser visto a seguir.



Este dispositivo possibilita a escrita de números decimais de 0 a 9, além de outros caracteres. Cada segmento possui um LED.

Projete um decodificador BCD 8421 para um display 7 segmentos. Como entrada considere os dígitos 0 a 9; e como saída os segmentos "a" até "g". *Dica:* Verifique em cada caractere os segmentos que devem ser acesos e atribua o nível 1 em função da respectiva entrada no código binário.

Referências e exercícios complementares

- IDOETA, Ivan V. e CAPUANO, Francisco G. Elementos de Eletrônica Digital. São Paulo: Editora Érica, ed. 40. 2008.

Página 229 - **5.6.1 até 5.6.9**

Sumário

Circuitos aritméticos

- Circuitos que realizam operações aritméticas com números binários.
- São utilizados, principalmente, para construir a **ULA** - Unidade Lógico Aritmética.
- Geralmente realizam operações de soma e subtração.

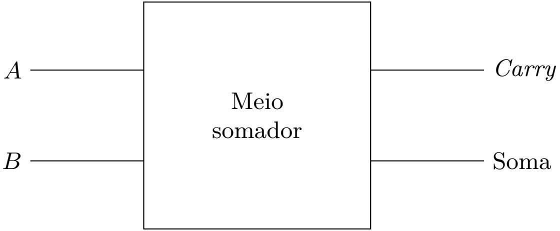
Circuitos somadores - Meio somador

Meio Somador

- Possibilita a soma de 2 números binários de 1 bit.
- Entradas: 2 bits.
- Saídas: Soma + *Carry*.

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

Circuitos somadores - Meio somador



Circuitos somadores - Meio somador

		A	
		0	1
B	0	0	1
	1	1	0

Expressão simplificada

$$S = A \oplus B$$

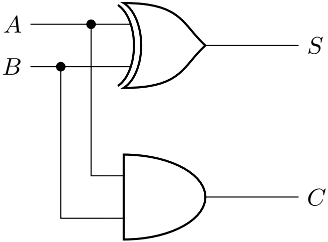
Circuitos somadores - Meio somador

		A	
		0	1
B	0	0	0
	1	0	$\boxed{1}$

Expressão simplificada

$$C = A \cdot B$$

Circuitos somadores - Meio somador



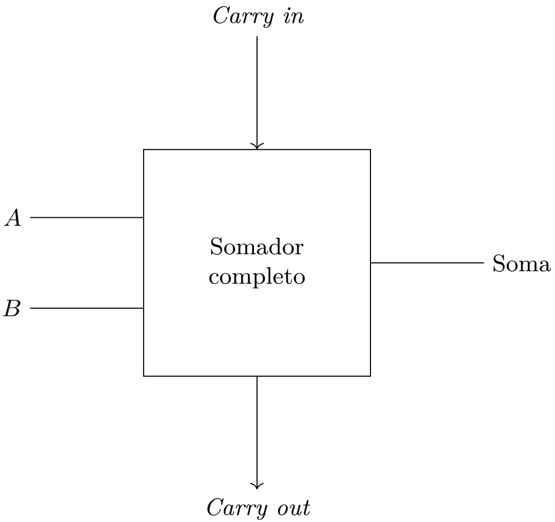
Circuitos somadores - Somador completo

Somador Completo

- Possibilita a soma de 2 números binários de 1 bit + o carry anterior.
- Entradas: 2 bits + *Carry in*.
- Saídas: Soma + *Carry out*.

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Circuitos somadores - Somador completo



Circuitos somadores - Somador completo

C_{in} \ AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Expressão simplificada

$$S = A \oplus B \oplus C_{in}$$

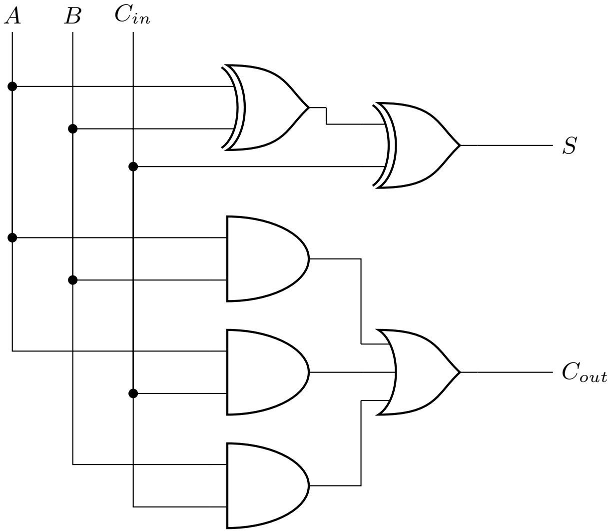
Circuitos somadores - Somador completo

		AB			
		00	01	11	10
C_{in}	0	0	0	1	0
	1	0	1	1	1

Expressão simplificada

$$C_{out} = A \cdot B + B \cdot C_{in} + A \cdot C_{in}$$

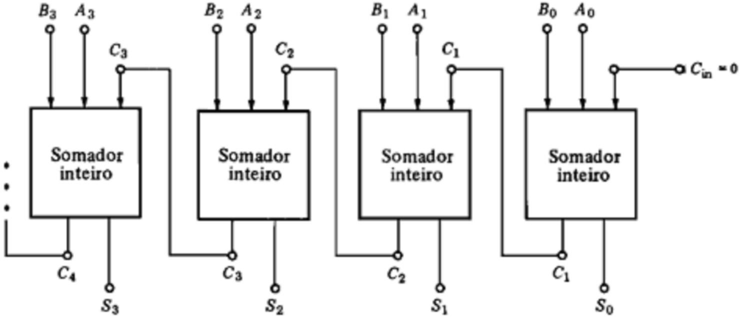
Circuitos somadores - Somador completo



Circuitos somadores - Somador de n bits

Ex.: Somador de 4 bits

- Utiliza-se 4 somadores completos, um para cada bit.
- Para o LSB pode ser utilizado um meio somador - **Por quê?**
- Conecta-se cada C_{out} no C_{in} do próximo bit.



Circuitos somadores - Somador completo a partir de meio somadores

- Meio somador:

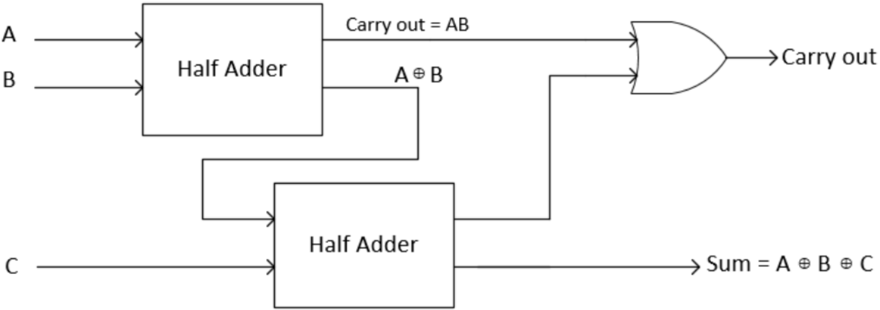
$$S = A \oplus B$$
$$C = A \cdot B$$

- Somador completo:

$$S = A \oplus B \oplus C_{in}$$

$$\begin{aligned} C_{out} &= \overline{A} \cdot B \cdot C_{in} + A \cdot \overline{B} \cdot C_{in} + A \cdot B \cdot \overline{C_{in}} + A \cdot B \cdot C_{in} \\ &= C_{in}(A \oplus B) + A \cdot B \end{aligned}$$

Circuitos Somadores - Somador completo a partir de meio somadores



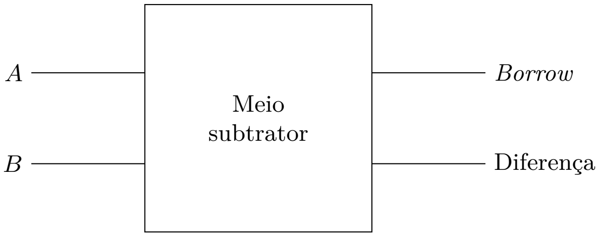
Circuitos subtratores - Meio subtrator

Meio Subtrator

- Possibilita a subtração de 2 números binários de 1 bit.
- Entradas: 2 bits.
- Saídas: Diferença + *Borrow*.

A	B	D	B_o
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Circuitos subtratores - Meio subtrator



Circuitos subtratores - Meio subtrator

		A	
		0	1
B	0	0	1
	1	1	0

Expressão simplificada

$$D = A \oplus B$$

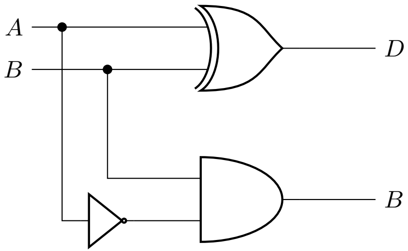
Circuitos subtratores - Meio subtrator

		A = 0	A = 1
B	0	0	0
	1	1	0

Expressão simplificada

$$B_o = \overline{A} \cdot B$$

Circuitos subtratores - Meio subtrator



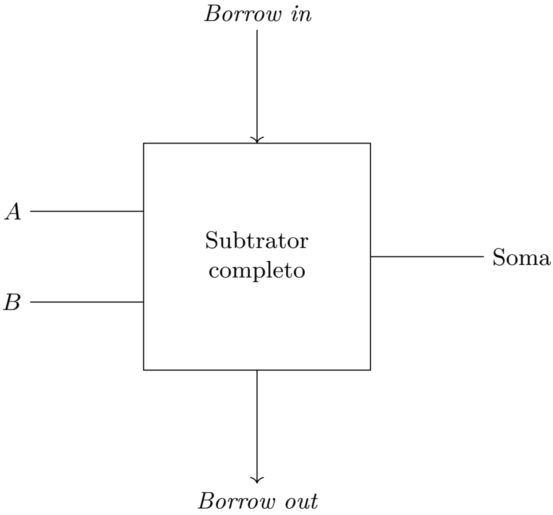
Circuitos subtratores - Subtrator completo

Subtrator Completo

- Possibilita a soma de 2 números binários de 1 bit + o borrow anterior.
- Entradas: 2 bits + *Borrow* anterior.
- Saídas: Diferença + *Borrow*.

A	B	B_{in}	D	B_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Circuitos subtratores - Subtrator completo



Circuitos subtratores - Subtrator completo

		AB			
		00	01	11	10
B_{in}	0	0	1	0	1
	1	1	0	1	0

Expressão simplificada

$$D = A \oplus B \oplus B_{in}$$

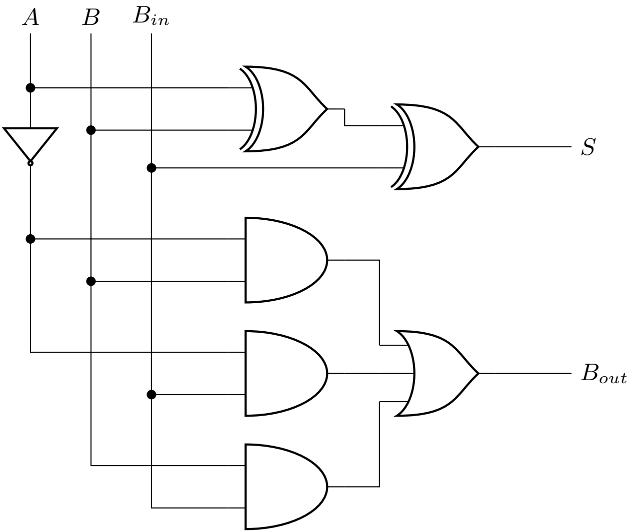
Circuitos subtratores - Subtrator completo

B_{in} \ AB	00	01	11	10
0	0	1	0	0
1	1	1	1	0

Expressão simplificada

$$B_{out} = \overline{A} \cdot B + \overline{A} \cdot B_{in} + B \cdot B_{in}$$

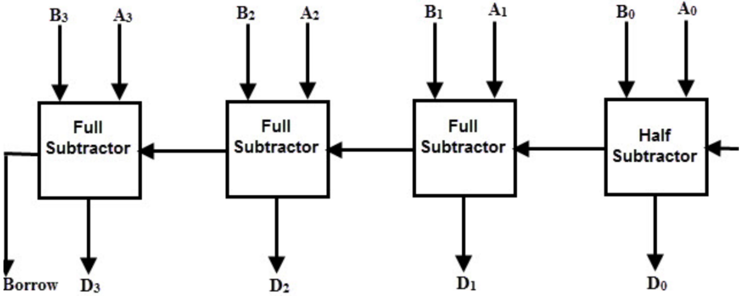
Circuitos subtratores - Subtrator completo



Circuitos subtratores - Subtrator de n bits

Ex.: Subtrator de 4 bits

- Utiliza-se 4 subtratores completos, um para cada bit.
- Conecta-se cada B_{out} no B_{in} do próximo bit.



Circuitos subtratores - Subtrator completo a partir de meio subtratores

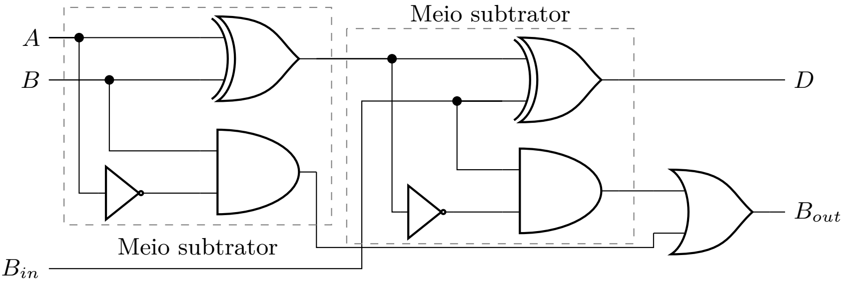
- Meio subtrator:

$$D = A \oplus B_{in}$$
$$B_{out} = \overline{A} \cdot B$$

- Subtrator completo:

$$\begin{aligned} D &= A \oplus B \oplus B_{in} \\ B_{out} &= \overline{A} \cdot B \cdot \overline{B_{in}} + \overline{A} \cdot \overline{B} \cdot B_{in} + \overline{A} \cdot B \cdot B_{in} + A \cdot B \cdot B_{in} \\ &= B_{in}\overline{(A \oplus B)} + \overline{A} \cdot B \end{aligned}$$

Circuitos subtratores - Subtrator completo a partir de meio subtratores



Exercícios

01. Projete um circuito combinacional que realize as duas operações - somador e subtrator. Para tal, acrescente uma variável de entrada **M** que faça este controle. Assuma que para $M = 0$, o circuito realize uma soma completa.

02. Desenhe um sistema somador para dois números de dois bits apenas com blocos de somadores completos.

Referências e exercícios complementares

- IDOETA, Ivan V. e CAPUANO, Francisco G. Elementos de Eletrônica Digital. São Paulo: Editora Érica, ed. 40. 2008.

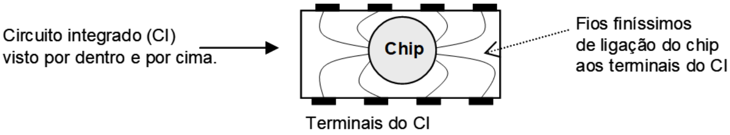
Página 229 - **5.6.10 até 5.6.17**

Sumário

Introdução

Definição

Os circuitos integrados são circuitos eletrônicos funcionais, constituídos por um conjunto de transistores, diodos, resistências e condensadores, fabricados num mesmo processo, sobre uma substância comum semicondutora de silício que se designa vulgarmente por chip.



Vantagens e limitações

Vantagens

- Redução de custos, peso e tamanho.
- Maior velocidade de trabalho.
- Menor consumo de energia.
- Redução dos erros de montagem.
- Melhoria das características técnicas do circuito.
- Simplifica ao máximo a produção industrial.

Vantagens e limitações

Limitações

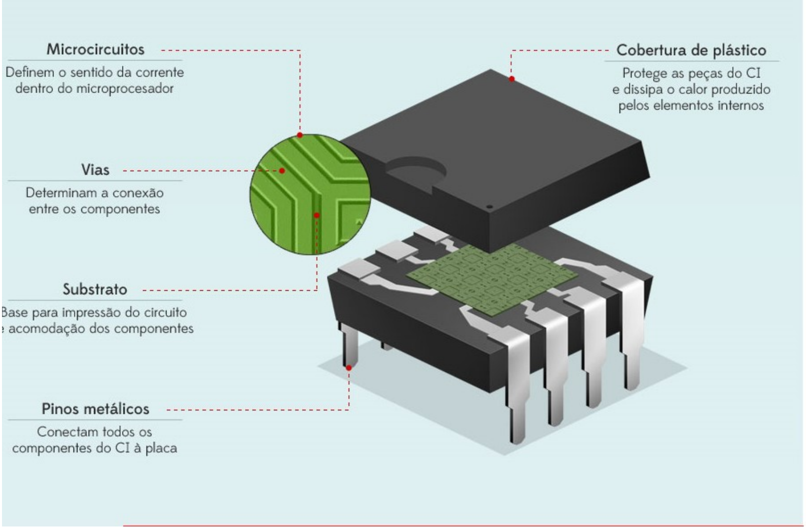
- Limitação nos valores das resistências e condensadores.
- Reduzida potência de dissipação.
- Limitações nas tensões de funcionamento.
- Impossibilidade de integrar num chip bobinas ou indutâncias.

Escala de integração e nanotecnologia

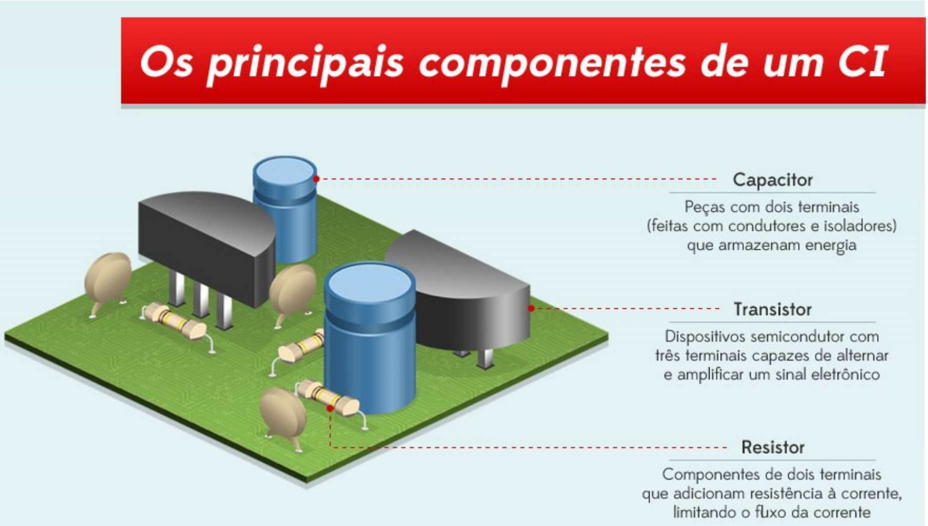
As portas não são vendidas individualmente, mas agrupadas em um circuito integrado (chip). Variam de acordo com seu tamanho:

- **SSI (Small Scale Integration)** - 1 a 12 portas.
- **MSI (Medium Scale Integration)** - 13 a 99 portas.
- **LSI (Large Scale Integration)** - 100 a 9999 portas.
- **VLSI (Very Large Scale Integration)** - 10 000 a 99 999 portas.
- **ULSI (Ultra Large Scale Integration)** - Acima de 100 000 portas.

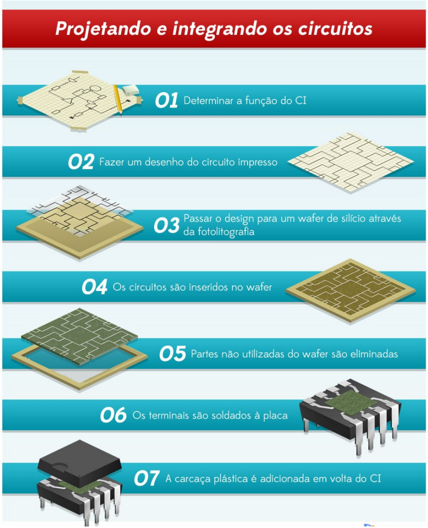
Como funciona?



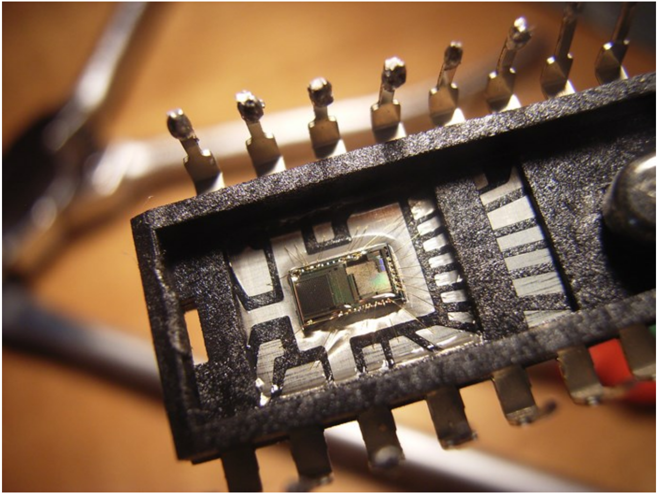
Como funciona?



Como funciona?



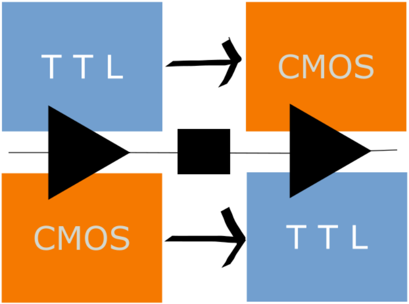
Como funciona?



Famílias: TTL x CMOS

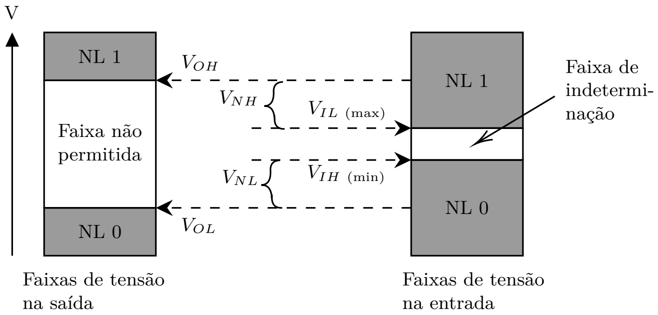
Introdução

- Tipo de estrutura interna.
- Cada família utiliza determinados componentes em seus blocos.
- Diferentes características.



Conceitos - Níveis de tensão

- Níveis 1 e 0 dentro de faixas.



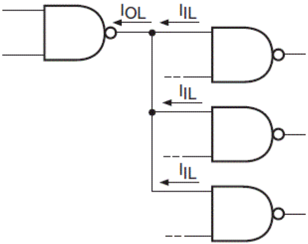
Conceitos - Níveis de corrente

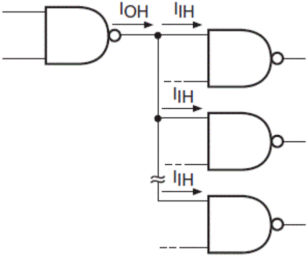
De modo semelhante ao nível de tensão:

- I_{IL}: valor de corrente máxima no terminal de entrada quando é aplicado nível 0
- I_{OL}: valor de corrente máxima que a saída pode receber quando em nível 0
- I_{IH}: valor de corrente máxima no terminal de entrada quando é aplicado nível 1
- I_{OH}: valor de corrente máxima que a saída pode receber quando em nível 1

Conceitos - *Fan out*

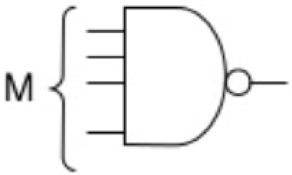
- A mesma saída pode ser usada para excitar diversas funções.
- A entrada de cada função precisa de certa corrente e a saída da função que vai excitar tem uma capacidade limitada de fornecimento de corrente.





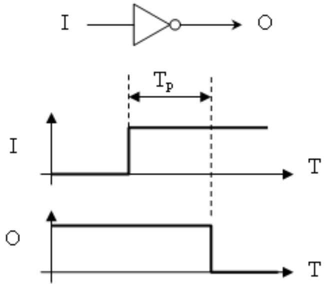
Conceitos - *Fan in*

- Definido como o número de entradas de determinada porta lógica
- Quanto maior este número, mais lenta será a porta.



Conceitos - Tempo de atraso de propagação

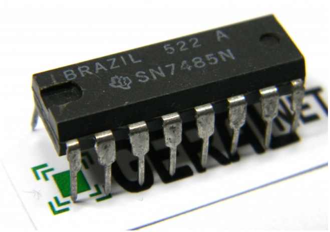
- Tempo que o bloco lógico leva para mudar de estado após uma mudança de nível.
- t_{PLH}: *low* to *high* - tempo para mudar de 0 para 1.
- t_{PHL}: *high* to *low* - tempo para mudar de 1 para 0.



A família TTL

Conceitos

- TTL significa Transistor-Transistor-Logic.
- A família TTL foi originalmente desenvolvida pela Texas Instruments.
- Série 74xx - comercial.
- Série 54xx - militar.



A família TTL - Voltagem

- Alimentado com uma tensão de 5 V.
- Nível lógico 0 é sempre a ausência de tensão ou 0 V.
- Nível lógico 1 é sempre uma tensão de 5 V.
- Os níveis lógicos para serem reconhecidos devem estar dentro de faixas bem definidas.

A família TTL - Corrente

- I_{IL}: 1,6 mA.
- I_{OL}: 16 mA.
- I_{IH}: 40 µA.
- I_{OH}: 400 µA.

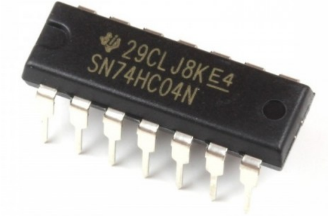
A família TTL - Tempo de atraso

- Valor médio de aproximadamente 10 ns, na versão *standard*.

A família CMOS

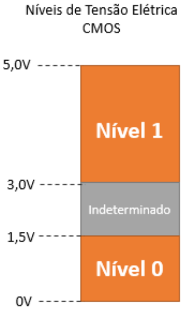
Conceitos

- CMOS significa Complementary MOS.
- Transistores MOSFET.
- Série 4000A - comercial.
- Série 4000B - comercial.
- Série 54/74C - comercial - similar ao TTL.
- Série 74HC/74HCT - comercial - alta velocidade.



A família CMOS - Voltagem

- Série 4000/74C - 3 a 15 V
- Série 74HC - 2 a 6 V
- Série 74HCT - 4,5 a 5,5 V



A família CMOS - Corrente

- I_{IL}: 1 µA.
- I_{OL}: 0,4 mA.
- I_{IH}: 1 µA.
- I_{OH}: 0,4 mA.

A família CMOS - Tempo de atraso

- Valor médio de aproximadamente 90 ns, nas séries mais comuns.

TTL x CMOS

Comparação entre as famílias

- A família CMOS possui maior imunidade a ruídos.
- A família CMOS possui um maior número de *fan out*.
- Ao contrário da família TTL, na CMOS não é aconselhável deixar terminais de entradas desconectados. Estes devem ser ligados na tensão positiva ou terra, dependendo do modelo.
- A família CMOS trabalha com uma faixa maior para alimentação.
- A família TTL não possui problemas com o manuseio dos CI - eletricidade estática.
- A família TTL possui um tempo de atraso menor, se comparado com a CMOS.

Exercícios

01. No último slide, têm-se a seguinte afirmação: *"A família CMOS possui um maior número de fan out".* Através dos conceitos e cálculos estudados, mostre que esta afirmação é verdadeira.

Referências e exercícios complementares

- IDOETA, Ivan V. e CAPUANO, Francisco G. Elementos de Eletrônica Digital. São Paulo: Editora Érica, ed. 40. 2008.

Página 469 - **9.8.9 até 9.8.11**

Encerramento

Tais notas de aula consistem em um compilado de diversas fontes, inclusive do livro texto indicado.

Estas não substituem a leitura do livro e a presença em sala de aula.

Para quaisquer dúvidas e/ou informações:

yago.correa@iff.edu.br