

Circuitos Sequenciais

Autor: Prof. João Ricardo dos S. Rosa
UNEMAT, Campus de Sinop

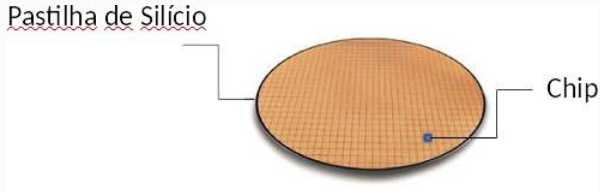
- 1 **Circuitos Integrados**
- 2 **Classificação de CI's**
- 3 **Portas Lógicas**
- 4 **Circuitos Combinacionais**

Os CI's são circuitos eletrônicos funcionais, constituídos por um conjunto de dispositivos eletrônicos (transístores, resistências, ...)

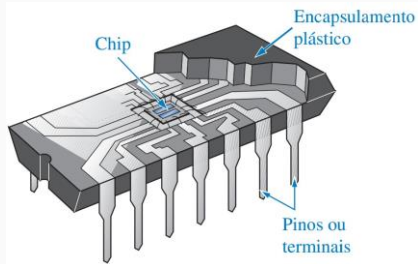
- ▶ Portas lógicas são fabricadas e vendidas como CI's ou chips
 - ▶ CI's são fabricados num mesmo processo, sobre uma substância comum semicondutora (silício)
 - ▶ O CI é um quadrado de silício montados em invólucros retangulares
- ▶ plástico ou de cerâmica
 - ▶ (5 a 15 x 20 a 50mm)



Pastilha de Silício



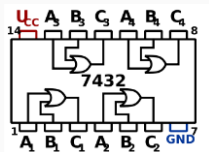
Placa de Circuito Impresso (PCB)



Classificação de CI's

► SSI (Small Scale Integration)

- Integração em pequena escala, com ≈ 12 portas lógicas em uma pastilha
- Portas Individuais e Flip-Flops



► MSI (Medium Scale Integration)

- Integração em média escala de 13 a 99 portas ou funções lógicas
- Somadores, contadores e registradores

- ▶ **LSI (Large Scale Integration)**
 - ▶ Larga escala de 100 a 999 portas ou funções lógicas
 - ▶ ROMs, PLA e memórias pequenas
- ▶ **VLSI (Very Large Scale Integration)**
 - ▶ Muito larga escala de 1.000 a 99.999 portas ou funções lógicas
 - ▶ Memórias grandes, microprocessadores e sistemas complexos
- ▶ **ULSI (Ultra Large Scale Integration)**
 - ▶ Ultra larga escala de 100.000 portas ou funções lógicas
 - ▶ Memórias grandes, microprocessadores e sistemas complexos
- ▶ **UVLSI (Ultra Very Large Scale Integration)**
 - ▶ Atualmente não é medido por portas ou funções lógicas (Ex.: Memórias e processadores atuais)

► Tipos de invólucros

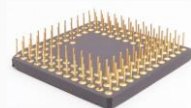
- DIP ou DIL (Dual In Line Packages) contém duas linhas de pinos para fora e com um CI dentro
- LCC (Leaded Chip Carrier) os pinos são encaixados na placa
 - Terminais nos 4 lados para encaixar em soquetes apropriados
 - PLCC (Plastic Leaded Chip Carrier)
- QFP (Quad Flat Pack)
 - Parecido com o LCC, a diferença é que são soldados na PCB



► Tipos de invólucros

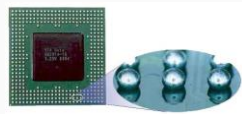
► PGA (Pin Grid Array)

- Contém pinos para a conexão com a placa em baixo do circuito



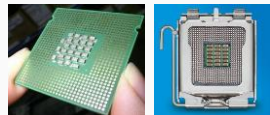
► BGA (Ball Grid Array)

- Bastante utilizado em chipsets
- Terminais presentes em baixo do circuito, sendo um PGA em miniatura



► LGA (Land Grid Array)

- Significa que os pinos em um socket LGA são incorporadas ao socket - não para a CPU



Portas Lógicas

Álgebra Booleana

- ▶ Na álgebra de boole, há somente dois **estados** (**valores** ou **símbolos**) permitidos
 - ▶ Estado **0** (zero)
 - ▶ Estado **1** (um)
- ▶ Em geral
 - ▶ O estado zero representa **não, falso**, aparelho desligado, ausência de tensão, chave elétrica desligada, etc
 - ▶ O estado um representa **sim, verdadeiro**, aparelho ligado, presença de tensão, chave ligada, etc

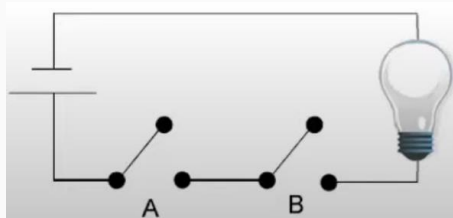
Tipos de portas lógicas

- ▶ AND (E)
- ▶ OR (OU)
- ▶ NOT(Inversora)
- ▶ NAND (Não E)
- ▶ NOR (Não OU)
- ▶ XOR (Exclusiva)

Função E (AND)

Funcionam da mesma forma que a conjunção E em nossa Língua

- Por exemplo, assuma a convecção no circuito
 - Chave aberta = 0; Chave fechada = 1
 - Lâmpada apagada = 0; Lâmpada acessa = 1



Função E (AND)

► Situações possíveis

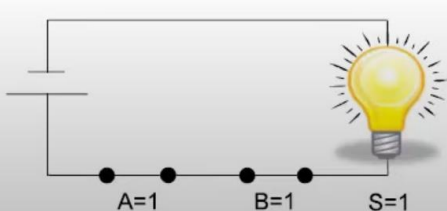
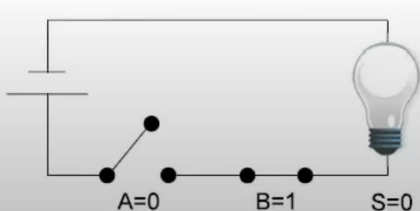
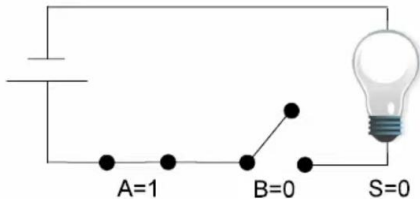
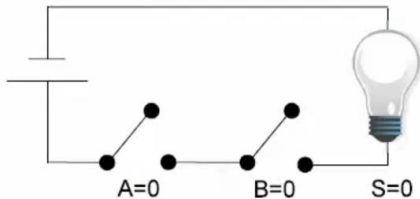


Tabela Verdade

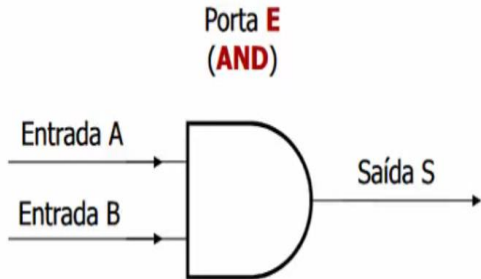
- ▶ A tabela verdade é um mapa onde são colocadas todas as possíveis interpretações (situações), com seus respectivos resultados para uma expressão
- ▶ Como visto no exemplo anterior, para 2 variáveis booleanas (**A** e **B**) existe 4 interpretações possíveis

Tabela Verdade da Função E (AND)

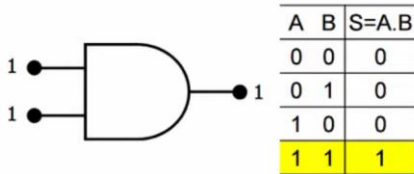
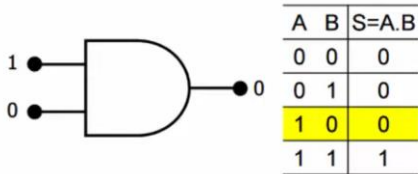
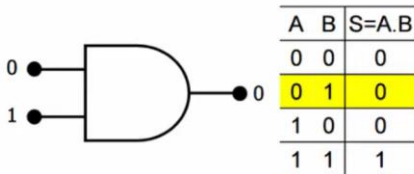
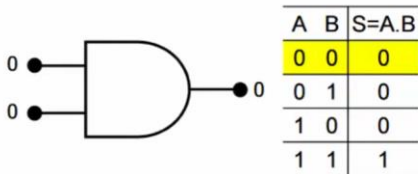
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

Porta lógica E (AND)

- ▶ A porta E é um circuito que executa a função E
- ▶ A porta E executa a tabela verdade da função E
 - ▶ Portanto, a saída será 1 somente se ambas as entradas forem iguais a 1 (**verdadeira**); nos demais casos a saída será 0
- ▶ Representação

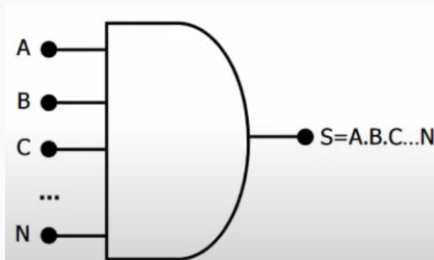


Porta lógica E (AND) $S = A.B$



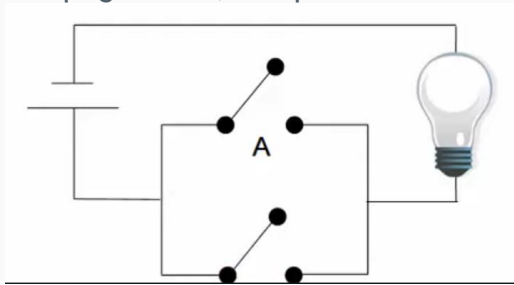
Porta lógica E (AND)

- ▶ É possível entender o conceito de uma porta E para um número qualquer de variáveis de entrada
- ▶ Nesse caso, temos uma porta E com N entradas e somente uma saída
- ▶ A saída será 1 se somente as N entradas forem iguais a 1; nos demais casos, a saída será 0

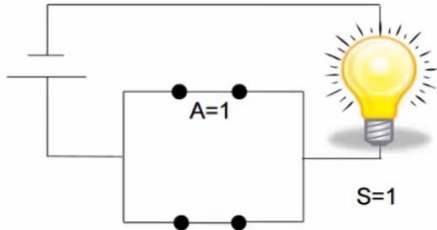
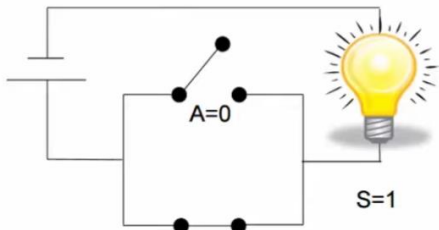
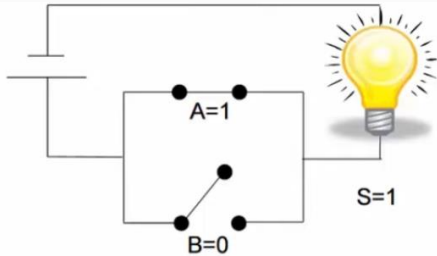
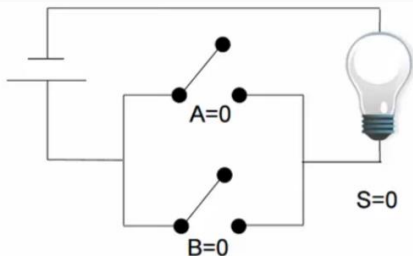


Função OU (OR)

- ▶ Executa a soma (disjunção) booleana de duas ou mais variáveis binárias
- ▶ Por exemplo, assumamos a conexão no circuito
 - ▶ Chave aberta = 0; Chave fechada = 1
 - ▶ Lâmpada apagada = 0; Lâmpada acesa = 1



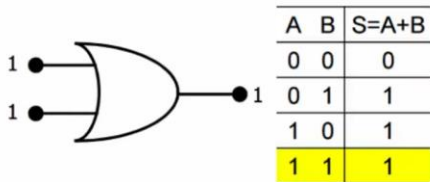
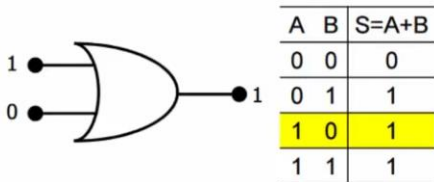
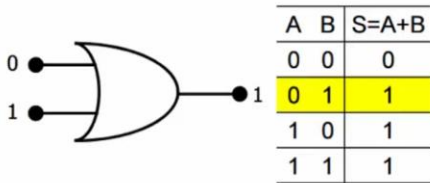
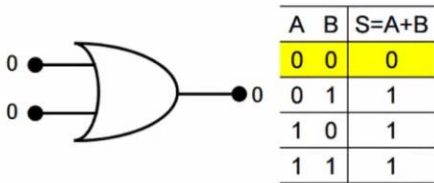
Função OU (OR)



Função **OU** (**OR**)

- <https://www.tinkercad.com/things/hoOi...>

Porta Lógica OU (OR) $S = A + B$



Função **NÃO** (NOT)

- ▶ Executa o complemento (negação de uma variável binária)
 - ▶ Se a variável estiver em 0, o resultado da função é 1
 - ▶ Se a variável estiver em 1, o resultado da função é 0
- ▶ Essa função também é chamada de inversora

Função **NÃO** (NOT)

- Usando as mesmas conecções dos circuitos anteriores, tem se que:
 - Quando a chave é A esta aberta ($A=0$), passara corrente pela lâmpada e ela acendera ($S=1$)
 - Quando a chave A esta fechada ($A=1$) a lâmpada estará em curto circuito e não passara corrente por ela ficando apagada ($S=0$)



Função NÃO (NOT)

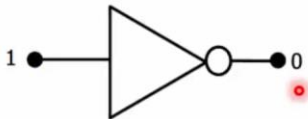
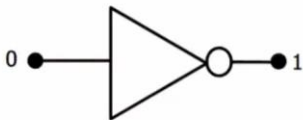
- ▶ Para representar a expressão
 - ▶ $S = \text{não } A$
- ▶ Adotaremos a representação
 - ▶ $S = \text{não } \bar{A}$ onde se lê $S = \text{não } A$
- ▶ Notações alternativas
 - ▶ $S = A'$
 - ▶ $S = \neg A$
 - ▶ $S = \tilde{A}$

Tabela Verdade NÃO (NOT)

- A tabela verdade da função NÃO (NOT)

A	\bar{A}
0	1
1	0

Porta lógica **NÃO** (NOT) $S = \bar{A}$



A	$S = \bar{A}$
0	1
1	0

A	$S = \bar{A}$
0	1
1	0

Função **NÃO E (NAND)**

- Composição da função **E** com a função **NÃO**, ou seja, a saída da função **E** é invertida
- Notações alternativas
 - ▶ $S = \overline{(A.B)}$
 - ▶ $S = (A.B)'$
 - ▶ $S = \vdash(A.B)$

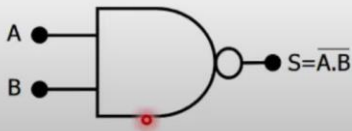
Tabela Verdade **NÃO E (NAND)**

- A tabela verdade da função **NÃO E (NAND)**

A	B	A.B
0	0	1
0	1	1
1	0	1
1	1	0

Porta lógica **NÃO E (NAND)** $S = \overline{A \cdot B}$

- ▶ A porta **NÃO E (NE)** é o bloco logico que executa a função **NÃO E**, ou seja, sua tabela verdade
- ▶ Representação



Função **NÃO OU (NOR)**

- Composição da função OU com a função NÃO, ou seja, a saída da função E é invertida
- Notações alternativas
 - ▶ $S = \overline{(A + B)} = \overline{(A + B)}$
 - ▶ $S = (A+B)'$
 - ▶ $S = \downarrow(A+B)$

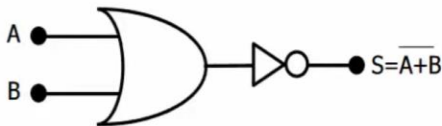
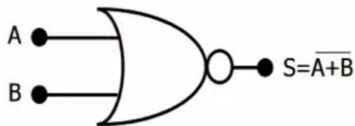
Tabela Verdade **NÃO OU (D)**

- A tabela verdade da função **NÃO E (NAND)**

A	B	$A+B$
0	0	1
0	1	0
1	0	0
1	1	0

Porta lógica **NÃO OU (NOR)** $S = \overline{A + B}$

- ▶ A porta **NÃO OU (NOR)** é o bloco logico que executa a função **NÃO OU**, ou seja, sua tabela verdade
- ▶ Representação



Função OU Exclusivo (XOR)

- ▶ A função OU Exclusivo fornece
 - ▶ 1 na saída quando as entradas forem diferentes entre si e
 - ▶ 0 caso o contrario
- ▶ Notações alternativas
 - ▶ $S = A \oplus B$
 - ▶ $S = \bar{A} \cdot B + A \cdot \bar{B}$

Tabela Verdade **OU Exclusivo (XOR)**

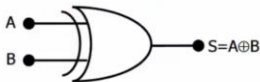
- A tabela verdade da função **OU Exclusivo (XOR)**

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

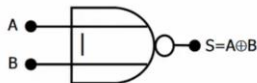
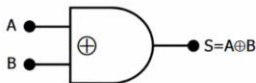
Porta lógica **OU Exclusivo (XOR)** $S = A \oplus B$

- ▶ A porta **NÃO OU (NOR)** é o bloco logico que executa a função **NÃO OU**, ou seja, sua tabela verdade
- ▶ Representação

Simbologia adotada





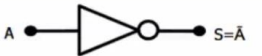



Outros símbolos utilizados



Porta lógica OU Exclusivo (XOR)

<https://www.tinkercad.com/things/0Sjz8KFSK11>

Resumo dos Blocos lógicos Básicos

Nome	Símbolo Gráfico	Função Algébrica	Tabela Verdade															
E (AND)		$S=A.B$ $S=AB$	<table><tr><th>A</th><th>B</th><th>S=A.B</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>	A	B	S=A.B	0	0	0	0	1	0	1	0	0	1	1	1
A	B	S=A.B																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OU (OR)		$S=A+B$	<table><tr><th>A</th><th>B</th><th>S=A+B</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>	A	B	S=A+B	0	0	0	0	1	1	1	0	1	1	1	1
A	B	S=A+B																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NÃO (NOT) Inversor		$S=\bar{A}$ $S=A'$ $S=\neg A$	<table><tr><th>A</th><th>S=\bar{A}</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	S= \bar{A}	0	1	1	0									
A	S= \bar{A}																	
0	1																	
1	0																	
NE (NAND)		$S=\overline{A.B}$ $S=(A.B)'$ $S=\neg(A.B)$	<table><tr><th>A</th><th>B</th><th>S=$\overline{A.B}$</th></tr><tr><td>0</td><td>0</td><td>1</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>0</td></tr></table>	A	B	S= $\overline{A.B}$	0	0	1	0	1	1	1	0	1	1	1	0
A	B	S= $\overline{A.B}$																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOU (NOR)		$S=\overline{A+B}$ $S=(A+B)'$ $S=\neg(A+B)$	<table><tr><th>A</th><th>B</th><th>S=$\overline{A+B}$</th></tr><tr><td>0</td><td>0</td><td>1</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>0</td></tr></table>	A	B	S= $\overline{A+B}$	0	0	1	0	1	0	1	0	0	1	1	0
A	B	S= $\overline{A+B}$																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR		$S=A\oplus B$	<table><tr><th>A</th><th>B</th><th>S=A⊕B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr></table>	A	B	S=A⊕B	0	0	0									
A	B	S=A⊕B																
0	0	0																

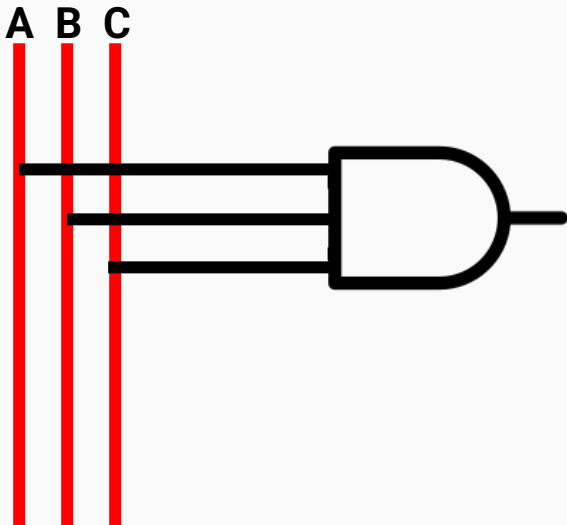
- Abra site Logic.ly e ascenda uma lâmpada utilizando um interruptor para as seguintes expressões – AND, OR, NOR, NAND

<https://logic.ly/demo/>

Exemplo 1

- Dada a expressão $S = A.B.C$ realize sua simbologia gráfica e sua tabela verdade

- Simbologia gráfica de $S = A.B.C$



- ▶ Em Sistemas digitais para a definimos o tamanho da tabela verdade de acordo com a expressão lógica utilizamos a expressão 2^x . Ou seja, neste caso temos a expressão composta por A.B.C.
- ▶ Sendo assim, nossa tabela será $2^3 = 8$. Portanto nossa tabela terá o total de 8 linhas (Possibilidades)

- Tabela verdade expressão $S = A.B.C$

A	B	C	A.B.C
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

Atividade

- Realize a simbologia lógica assim como a tabela verdade e indique a saída das seguintes expressões lógicas

► $S = A.B.C.D$

► $S = A+B+C$

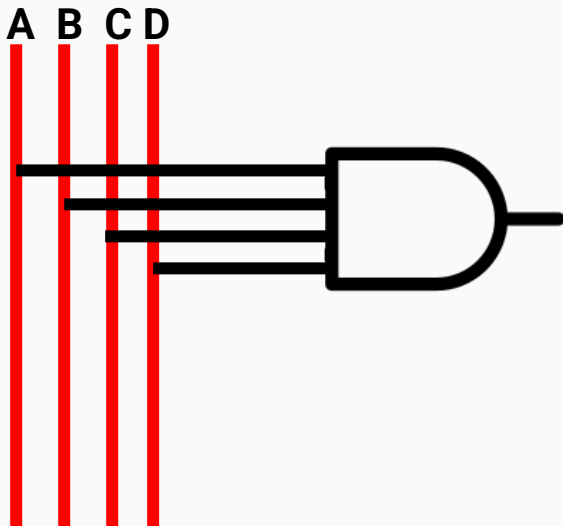
► $S = A \oplus B \oplus C \oplus D$

► $S = \overline{(A.B.C)}$

► $S = \bar{A} + B + \bar{C} + D$

► $S = A \oplus \bar{B} \oplus \bar{C}$

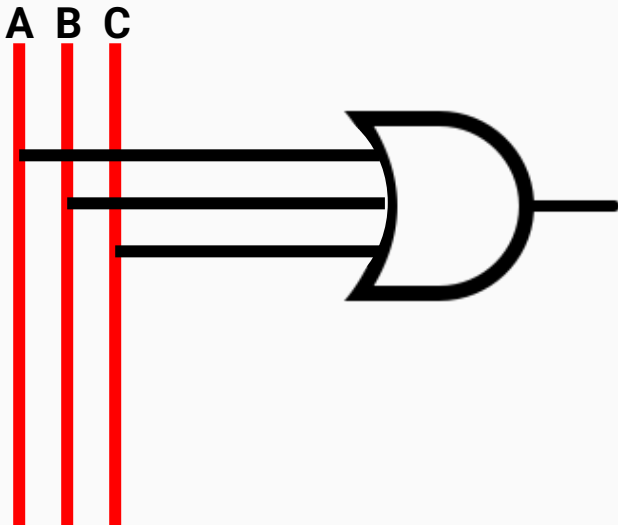
- A) Simbologia gráfica AND $S = A.B.C.D$



- A) Tabela verdade expressão AND $S = A.B.C.D$

A	B	C	D	A.B.C.D
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

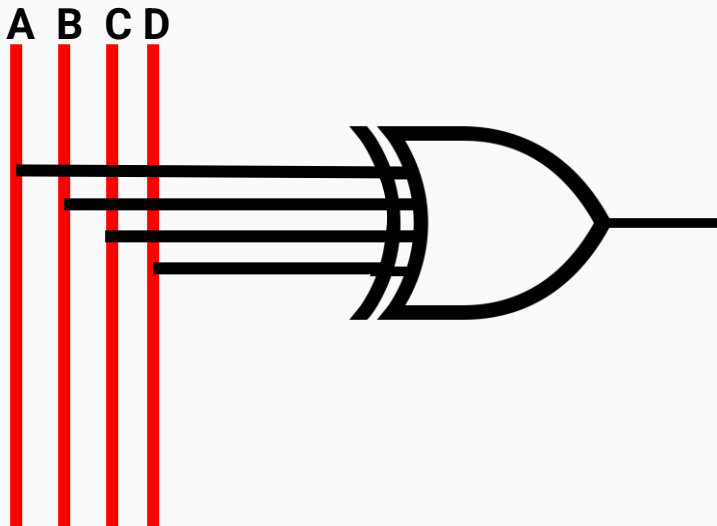
- B) Simbologia gráfica OR $S = A+B+C$



- B) Tabela verdade expressão OR $S = A+B+C$

A	B	C	$A+B+C$
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

- C) Simbologia gráfica XOR $S = A \oplus B \oplus C \oplus D$

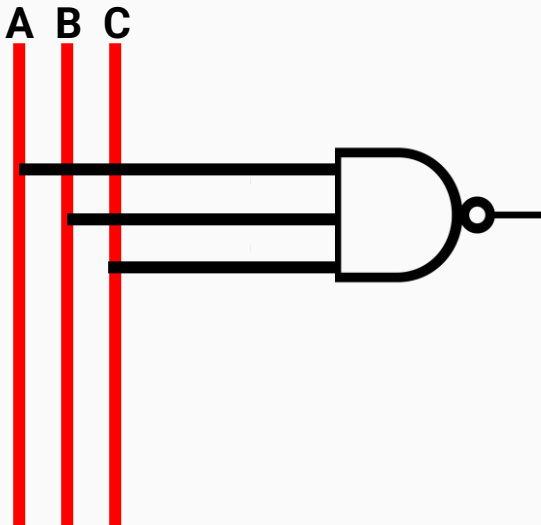


- C) Tabela verdade expressão XOR $S = A \oplus B \oplus C \oplus D$

A	B	C	D	$A \oplus B \oplus C \oplus D$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

- Simbologia gráfica NAND

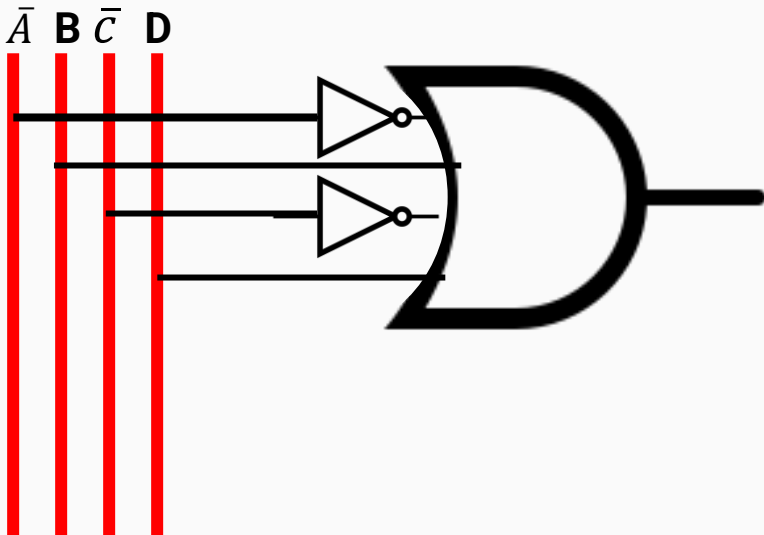
$$S = \overline{(A \cdot B \cdot C)}$$



- A) Tabela verdade expressão NAND $S = A.B.C$

A	B	C	D	A.B.C.D
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

- A) Simbologia gráfica XOR $\bar{A} + B + \bar{C} + D$



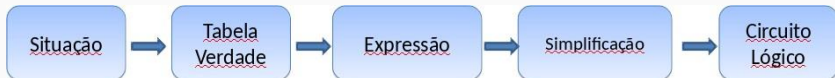
► A) Tabela verdade expressão NAND $S = \bar{A} + B + \bar{C} + D$

A	B	C	D	$\bar{A} + B + \bar{C} + D$
1	0	1	0	1
1	0	1	1	1
1	0	0	0	1
1	0	0	1	1
1	1	1	0	1
1	1	1	1	1
1	1	0	0	1
1	1	0	1	1
0	0	1	0	1
0	0	1	1	1
0	0	0	0	1
0	0	0	1	1
0	1	1	0	1
0	1	1	1	1
0	1	0	0	1
0	1	0	1	1

Introdução

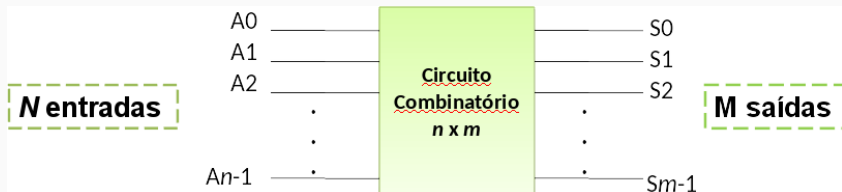
Soluciona problemas em que necessitamos de uma resposta, quando acontecerem determinadas situações (entradas)

Para um projeto de circuito combinacional, segue-se alguns passos básicos



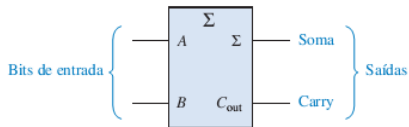
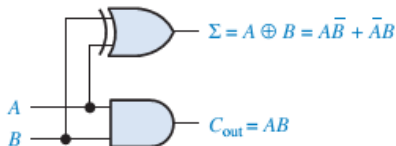
Introdução

São aqueles em que a saída depende única e exclusivamente das variáveis de entrada



Introdução

O meio-somador aceita dois dígitos binários em suas entradas e produz dois dígitos binários em suas saídas, um bit de soma e um bit de carry.



$0 + 0 = 0$
 $0 + 1 = 1$
 $1 + 0 = 1$
 $1 + 1 = 10$

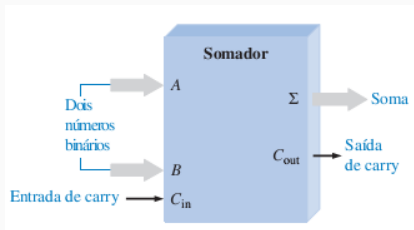
A	B	C_{out}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Σ = soma

C_{out} = carry de saída

A e B = variáveis de entrada (operandos)

- ▶ Contém dois bits de entrada e um carry de entrada
- ▶ Gera uma saída de soma e um carry de saída
- ▶ Diferença para o meio-somador é o carry de entrada

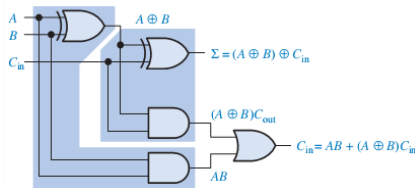
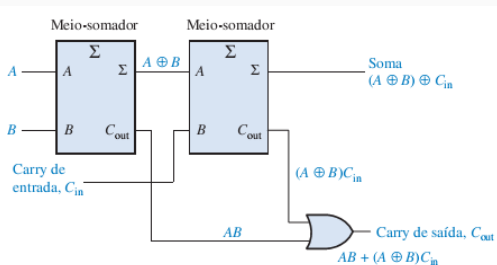


A	B	C_{in}	C_{out}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

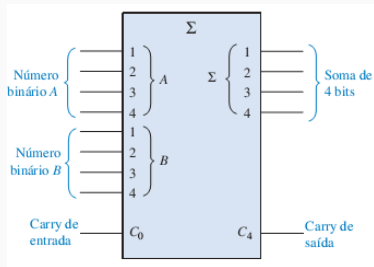
C_{in} = carry de entrada, algumas vezes indicado como CI
 C_{out} = carry de saída, algumas vezes indicado como CO
 Σ = soma
 A e B = variáveis de entrada (operandos)

Introdução

O somador completo é utilizado como circuito base para operações com maior quantidade de bits.

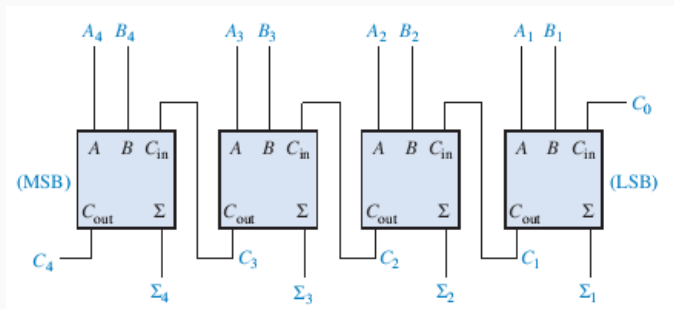


- Tabela-verdade para cada estágio de um somador paralelo de 4 bits



C_{n-1}	A_n	B_n	Σ_n	C_n
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

- Implementado com 4 estágios de somadores completos
- Ordem de execução é do LSB p/ o MSB
- A saída de carry de cada somador é conectada à entrada de carry do próximo somador



Introdução

O meio-subtrator aceita dois dígitos binários em suas entradas e produz dois dígitos binários em suas saídas, um bit de saída e um bit de borrow.

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

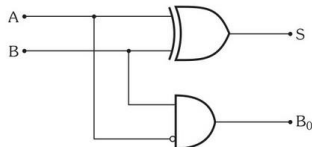
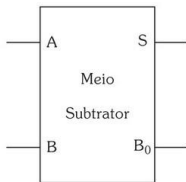


Figura 4.44 - Tabela-verdade e diagrama de blocos do meio subtrator.

Figura 4.45 - Circuito lógico do meio subtrator.

- ▶ Contém dois bits de entrada e um borrow de entrada
- ▶ Gera uma saída e um borrow de saída
- ▶ Diferença para o meio-subtrator é o borrow de entrada

A	B _i	B	S	B ₀
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

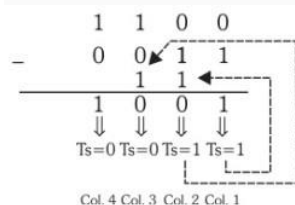
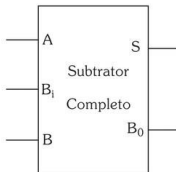
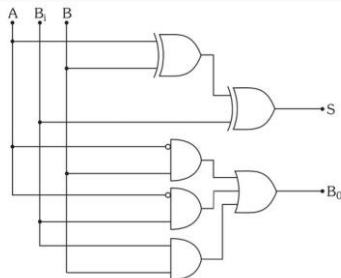
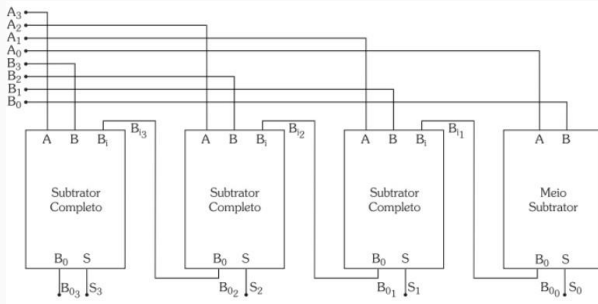
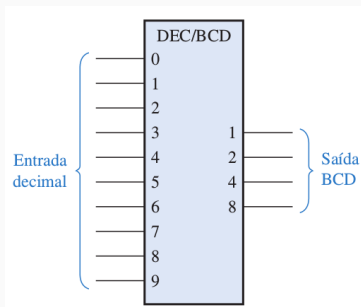


Figura 4.46 - Tabela-verdade e diagrama de blocos do subtrator completo.



- ▶ é um circuito lógico que possibilita a passagem de um código conhecido para um desconhecido
- ▶ Aceita um nível ativo em uma de suas entradas representando um dígito (decimal) e o converte em uma saída codificada (binário ou BCD - binary coded decimal).



DÍGITO DECIMAL	CÓDIGO BCD			
	A ₃	A ₂	A ₁	A ₀
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

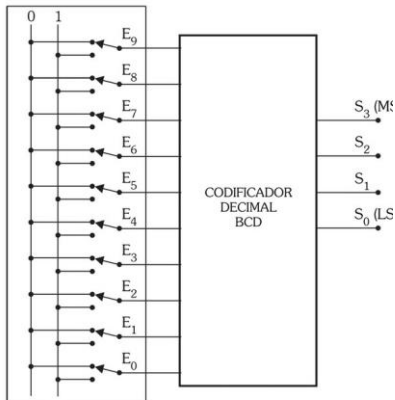


Figura 4.8 - Sistema teclado - codificador BCD.

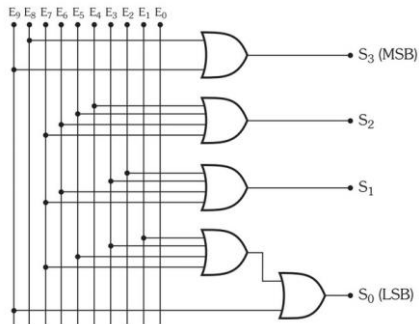
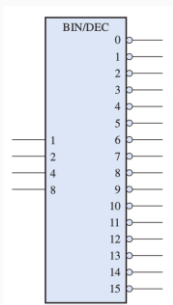
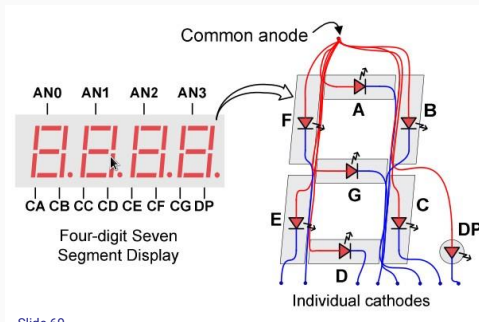
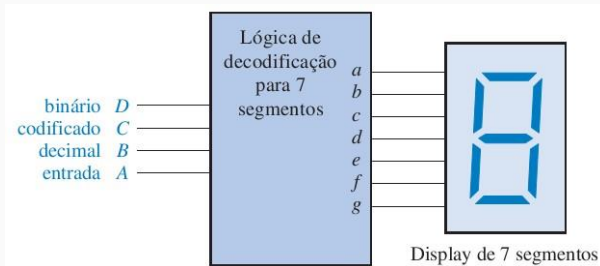


Figura 4.10 - Circuito lógico do codificador decimal - BCD.

- ▶ É um circuito lógico que faz o inverso do codificador
- ▶ Detecta a presença de uma combinação específica de bits (código) em suas entradas indicando a presença desse código através de um nível de saída especificado.



DÍGITO DECIMAL	CÓDIGO BCD				FUNÇÃO DE DECODIFICAÇÃO
	A ₃	A ₂	A ₁	A ₀	
0	0	0	0	0	$\bar{A}_3\bar{A}_2\bar{A}_1\bar{A}_0$
1	0	0	0	1	$\bar{A}_3\bar{A}_2\bar{A}_1A_0$
2	0	0	1	0	$\bar{A}_3\bar{A}_2A_1\bar{A}_0$
3	0	0	1	1	$\bar{A}_3\bar{A}_2A_1A_0$
4	0	1	0	0	$\bar{A}_3A_2\bar{A}_1\bar{A}_0$
5	0	1	0	1	$\bar{A}_3A_2\bar{A}_1A_0$
6	0	1	1	0	$\bar{A}_3A_2A_1\bar{A}_0$
7	0	1	1	1	$\bar{A}_3A_2A_1A_0$
8	1	0	0	0	$A_3\bar{A}_2\bar{A}_1\bar{A}_0$
9	1	0	0	1	$A_3\bar{A}_2\bar{A}_1A_0$



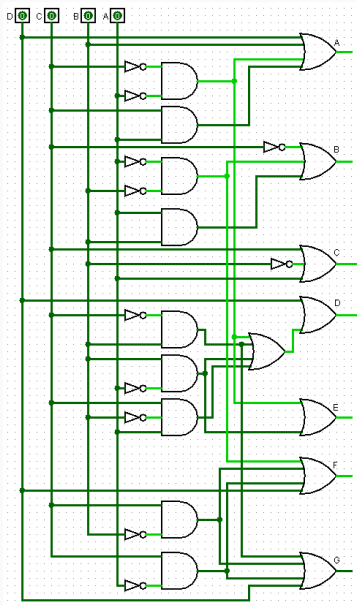




Figura 4.11 - Diagrama genérico de um decodificador.

D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X

Figura 4.14 - Tabela-verdade do decodificador BCD - 7 segmentos para display catodo comum.

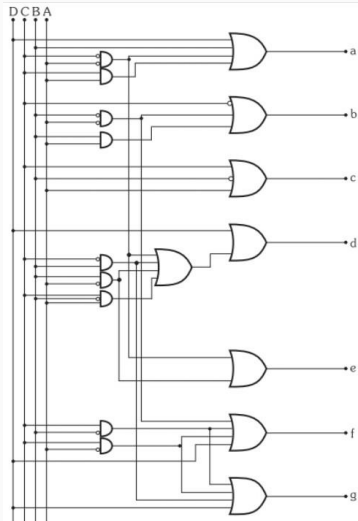
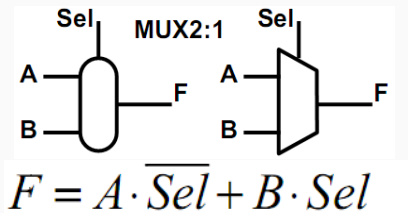


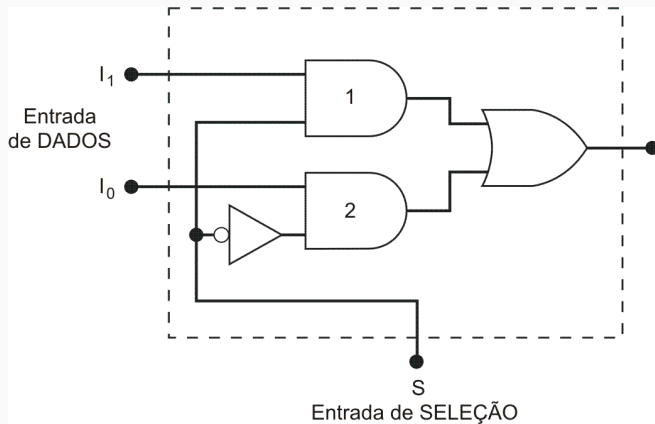
Figura 4.15 - Circuito lógico do decodificador BCD - 7 segmentos para display catodo comum.

- ▶ É um circuito com 2^n entradas de dados, uma saída e n entradas de controle (seleção)
- ▶ é um dispositivo que permite que informações digitais de diversas fontes sejam encaminhadas para uma única linha para serem transmitidas nessa linha para um destino comum



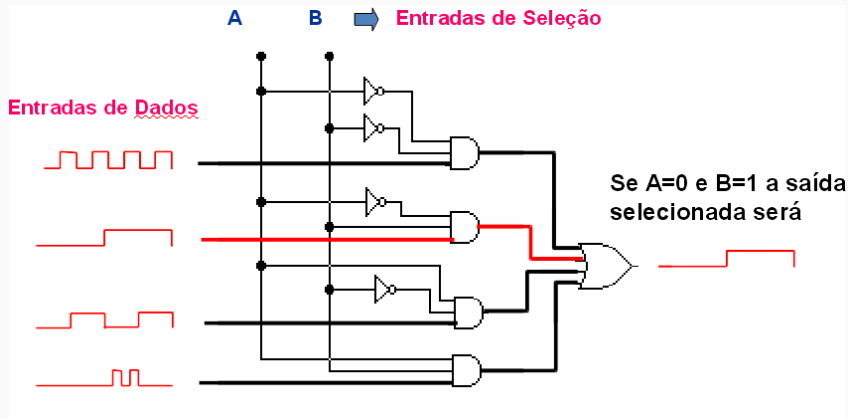
A	B	Sel	F
0	X	0	0
1	X	0	1
X	0	1	0
X	1	1	1

Multiplexador (Mux) ou Seletor



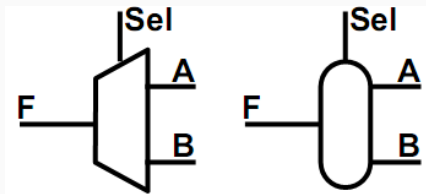
$$Z = I_0 \cdot \bar{S} + I_1 \cdot S$$

S	Saída
0	$Z = I_0$
1	$Z = I_1$



Demultiplexador (DeMux)

- Realiza a função inversa dos multiplexadores
- Uma entrada direcionada a uma de várias saídas



$$A = F \cdot \overline{Sel}$$

$$B = F \cdot Sel$$

F	Sel	A	B
0	0	0	-
1	0	1	-
0	1	-	0
1	1	-	1

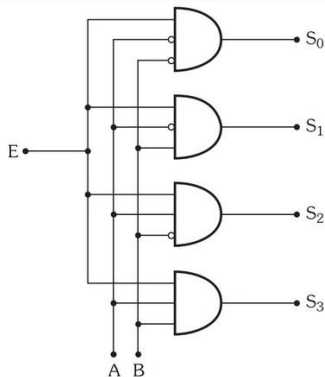
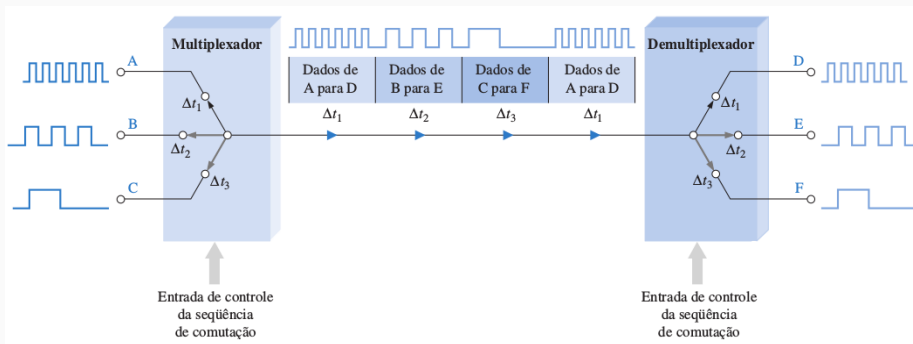
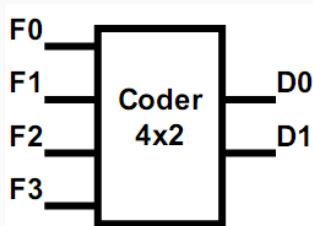
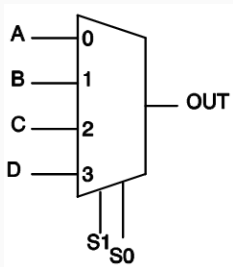


Figura 4.31 - Circuito lógico do DEMUX de quatro canais.



- ▶ Existe diferença?
- ▶ Qual(is) a(s) diferença(s)?



Comparador de 2 e 4 bits

- Comparam dois valores binários, bit a bit
 - Geram uma saída indicando se eles são iguais ou não
 - Pode indicar nas saídas se $A > B$ (GT), $A = B$ (EQ) ou $A < B$ (LT)
 - As saídas são mutuamente exclusivas, apenas uma delas pode estar ativa

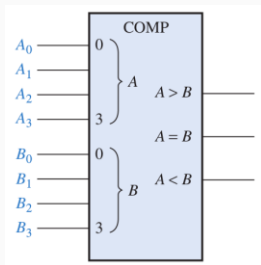


Tabela funcional

				GT ($A > B$)	EQ ($A = B$)	LT ($A < B$)
A_1	B_1	A_0	B_0			
$A_1 > B_1$		X	X	1	0	0
$A_1 < B_1$		X	X	0	0	1
$A_1 = B_1$		$A_0 > B_0$		1	0	0
$A_1 = B_1$		$A_0 < B_0$		0	0	1
$A_1 = B_1$		$A_0 = B_0$		0	1	0

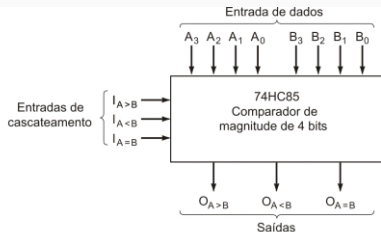


TABELA-VERDADE

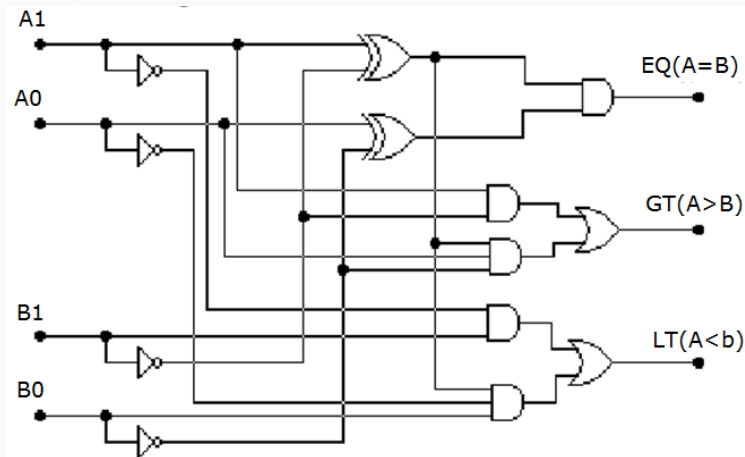
ENTRADAS DE COMPARAÇÃO				ENTRADAS DE CASCADEAMENTO			SAÍDAS		
A ₃ , B ₃	A ₂ , B ₂	A ₁ , B ₁	A ₀ , B ₀	I _{A>B}	I _{A<B}	I _{A=B}	O _{A>B}	O _{A<B}	O _{A=B}
A ₃ >B ₃	X	X	X	X	X	X	H	L	L
A ₃ <B ₃	X	X	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ >B ₂	X	X	X	X	X	H	L	L
A ₃ =B ₃	A ₂ <B ₂	X	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ >B ₁	X	X	X	X	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ <B ₁	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ >B ₀	X	X	X	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ <B ₀	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	H	L	L	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	L	H	L	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	X	X	H	L	L	H
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	L	L	L	H	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	H	H	L	L	L	L

H = Nível de tensão ALTO

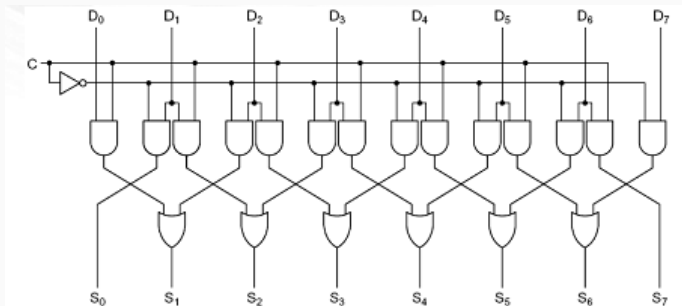
L = Nível de tensão BAIXO

X = Irr relevante

Comparador 2 bits

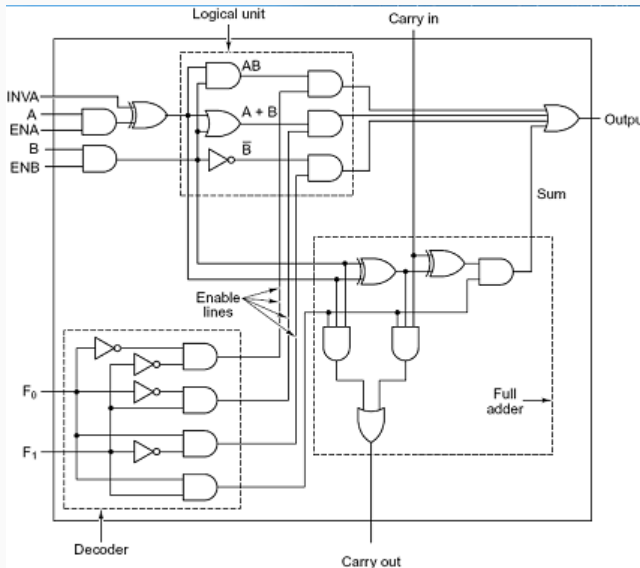
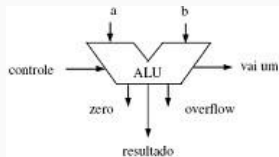


- ▶ Um deslocador de 8 bits contém 8 entradas (D_0, \dots, D_7) e 8 saídas (S_0, \dots, S_7)
- ▶ A saída é a entrada deslocada de 1 bit
- ▶ A linha de controle C determina a direção do deslocamento, 0 para a esquerda e 1 para a direita
- ▶ Repare nas portas AND que ficam dependentes de C



Unidade Lógica e Aritmética (ULA)

- ▶ é uma das 3 principais partes de um processador
- ▶ Executa operações lógicas e aritméticas



Perguntas ???

► [Sumário](#)

- 1 FLOYD, Thomas L. "Sistemas Digitais – Fundamentos e Aplicações". Bookman. 2007.
- 2 TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. Sistemas digitais: princípios e aplicações. São Paulo: Pearson Prentice Hall, 2007.
- 3 FERNANDES, E. M. L. "Álgebra de Boole e Circuitos Lógicos Básicos". Curso de AOC – UDESC, 2007. Cap III.
- 4 CASTRO, de F. C. C. "Eletrônica Digital". PUCRS – Faculdade de Engenharia Elétrica, 2007. Cap.IV.
- 5 CALVI, C. "Nível da Lógica Digital: Circuitos Lógicos Digitais Básicos". UFES – Departamento de Informática, 2006.
- 6 MONTEIRO, Mario A. Introdução à organização de computadores. Rio de Janeiro: LTC, 2010.
- 7 LOURENÇO, A. C.; CRUZ, E. C. A; FERREIRA, S. R, CHOVERI, J.