



Sistemas Digitais

Blocos Lógicos Combinatórios

Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro

Algumas figuras provenientes de
J. Wakerly, "Digital Design Principles & practices"



Resumo

- Documentação
- Descodificadores
- Codificadores
- Multiplexers
- Demultiplexers
- Multiplexagem em BUS



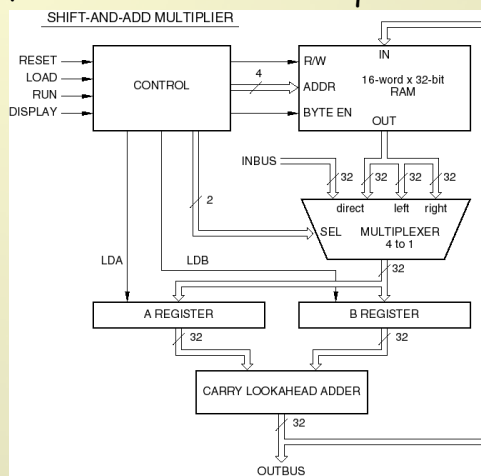
Documentação

- Essencial em todo o ciclo de vida do projecto
- Diagramas de Blocos
- Esquemas lógicos (Logigramas)
- Esquemas eléctricos
- HDL (ABEL, Verilog, VHDL)
- Diagramas Temporais
- Especificações dos componentes



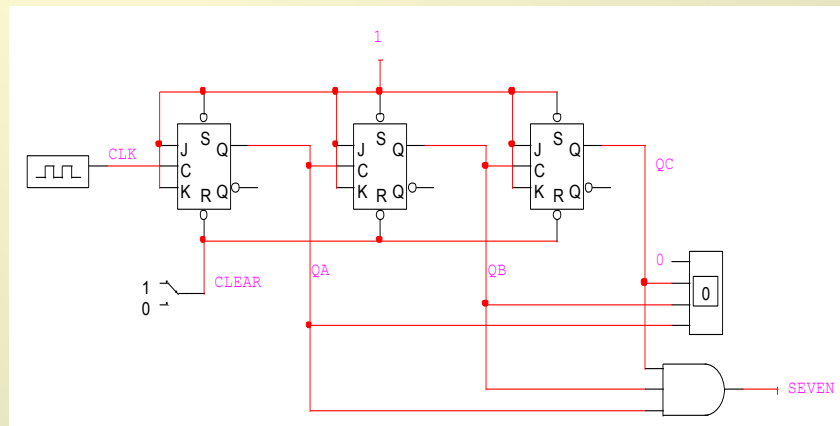
Diagrama de Blocos

- Descrição funcional ao nível da arquitectura



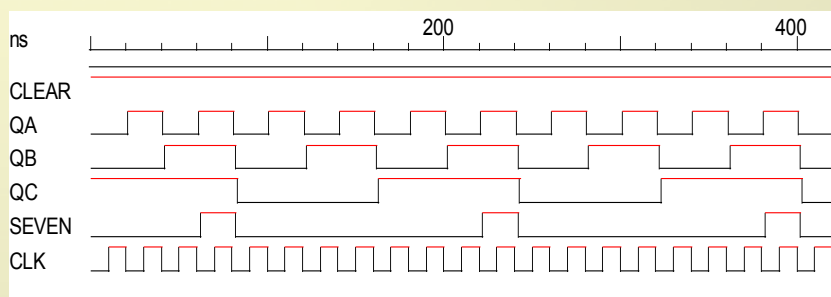


Esquemas Lógicos



Diagramas Temporais

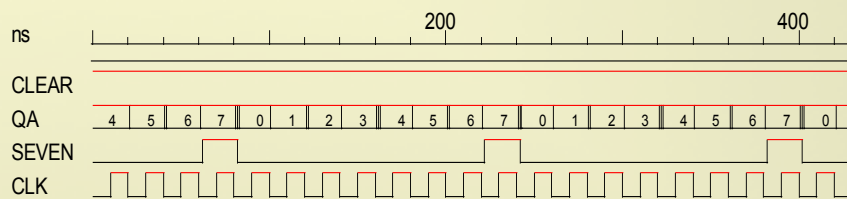
- Devem representar-se sinais estratégicos para o estudo do comportamento do circuito



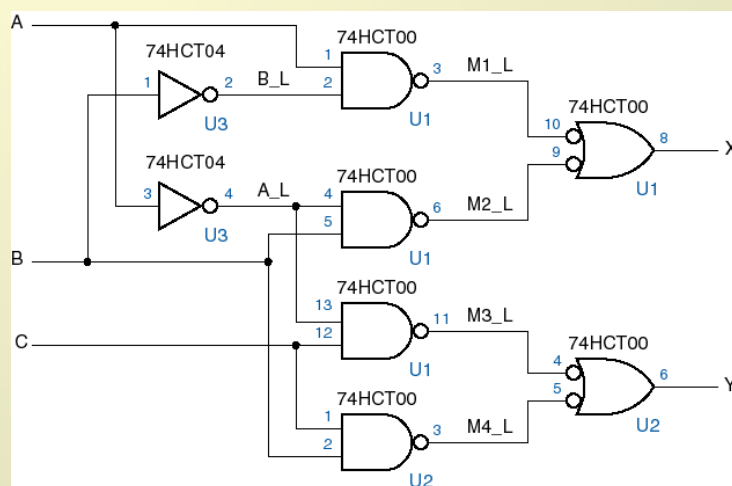


Diagramas Temporais

- Visualização alternativa
 - (QA,QB,QC) colapsados e representados pelo nº decimal correspondente ao respectivo código binário



Esquemas Eléctricos



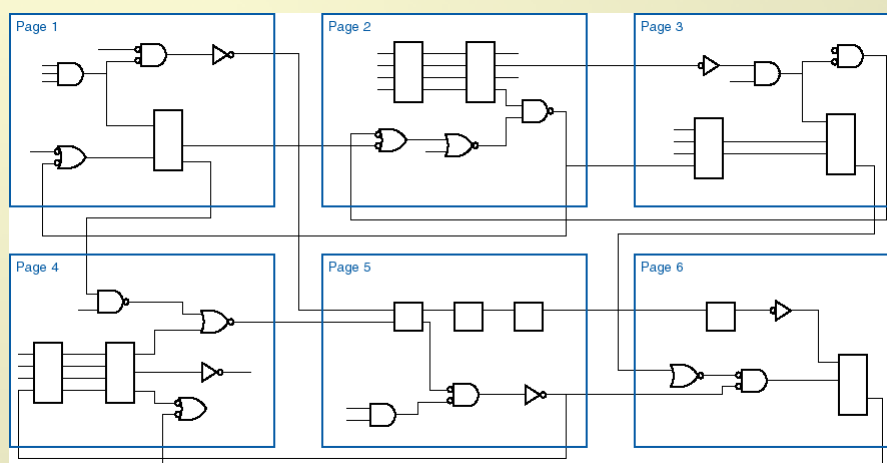


Esquemas Eléctricos

- Referência do componente
- Detalhes sobre entradas/saídas de cada componente
- Números dos pinos
- Títulos de blocos constituintes em desenho hierárquico
- Designação dos sinais
- Conectores entre páginas

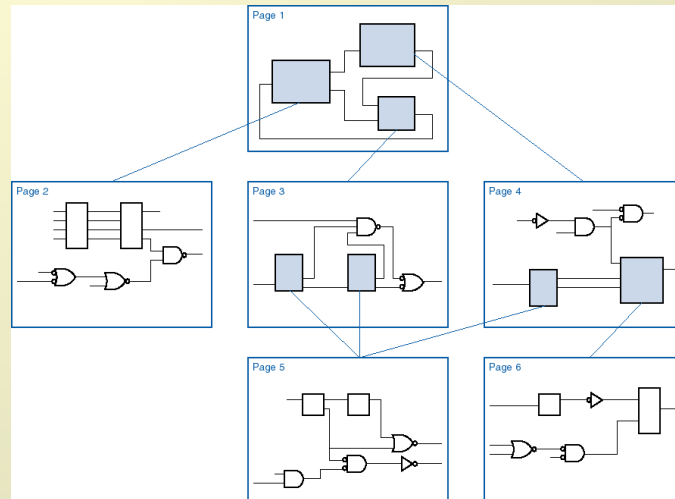


Organização "Flat"





Organização Hierárquica



VHDL

- Assunto do próximo semestre em Labs. Sistemas Digitais.

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
entity COUNT8 is
  port(
    DIN   : in  std_logic_vector(7 downto 0);
    CLK   : in  std_logic;
    LOAD  : in  std_logic;
    DOUT  : out std_logic_vector(7 downto 0)
  );
end COUNT8;
...
```



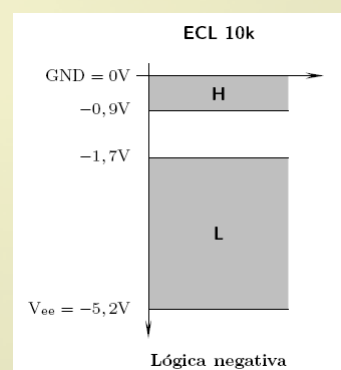
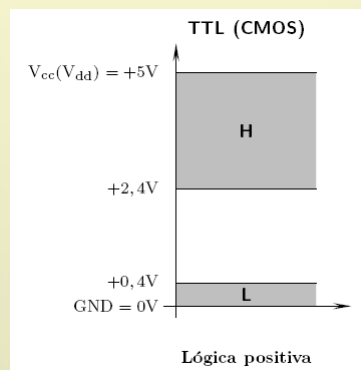
VHDL

```
...  
  
architecture arch1 of COUNT8 is  
  
begin  
  clk_proc : process(CLK)  
    variable COUNT : unsigned(7 downto 0) := "00000000";  
  begin  
    if CLK'EVENT AND CLK = '1' then  
      if LOAD = '1' then  
        COUNT := DIN;  
      else COUNT := COUNT + 1;  
      end if;  
    end if;  
    DOUT <= COUNT after 500ps;  
  end process clk_proc;  
  
end arch1;
```



Lógica de Polaridade

- Será que ao "1" corresponde sempre ao nível eléctrico mais positivo e o "0" ao mais negativo?
- R: Depende da realização electrónica.





Tabelas de verdade físicas

- Descrição tabular do comportamento do circuito recorrendo aos níveis H e L.
- Em contextos não algébricos a descrição funcional deve confinar-se aos símbolos H e L

Tabela de verdade física			Tabela de verdade em lógica positiva		
A	B	S	A	B	S
L	L	H	0	0	1
L	H	H	0	1	1
H	L	H	1	0	1
H	H	L	1	1	0

Lógica positiva
 $\xrightarrow{H=1, L=0}$

Tabela de verdade física			Tabela de verdade em lógica negativa		
A	B	S	A	B	S
L	L	H	1	1	0
L	H	H	1	0	0
H	L	H	0	1	0
H	H	L	0	0	1

Lógica negativa
 $\xrightarrow{H=0, L=1}$



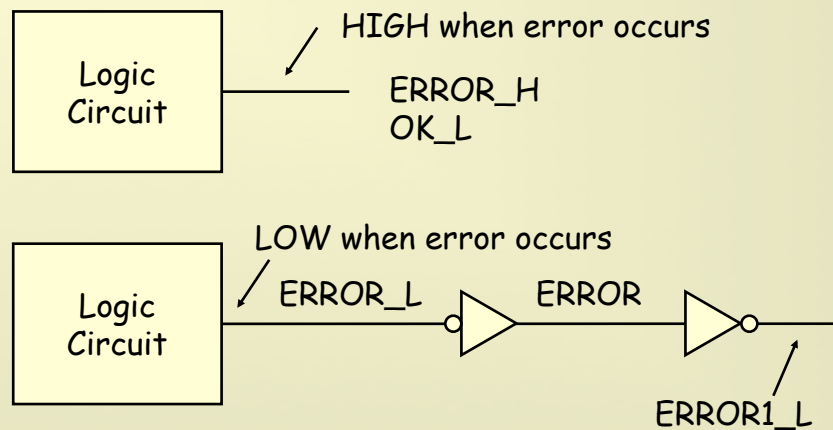
Nomes e níveis activos

- Por razões do contexto físico interessa associar a um sinal uma determinada acção no circuito e o nível eléctrico que o desencadeia
 - Pode ser H ou "muitas vezes" pode ser L
 - O nível assim definido designa-se por nível activo
- Exemplos

Active Low	Active High
READY-	READY+
ERROR.L	ERROR.H
ADDR15(L)	ADDR15(H)
RESET*	RESET
ENABLE~	ENABLE
~GO	GO
/RECEIVE	RECEIVE
TRANSMIT_L	TRANSMIT

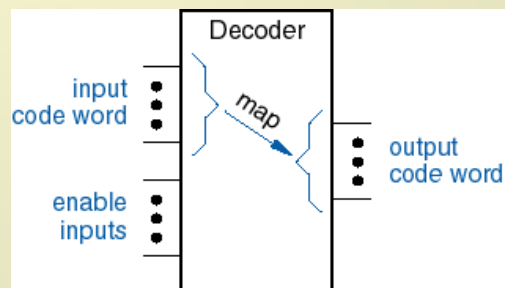


Exemplo



Descodificadores

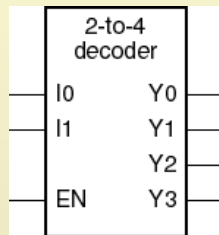
- Um descodificador é um circuito combinatório que permite, para uma palavra em código à entrada activar uma e só uma saída da função de descodificação
- Mais comum: descodificadores para o código binário natural
- Estrutura genérica:





Descodificador 2:4

- Tipicamente temos n entradas (código) e 2^n saídas
 - 2-to-4, 3-to-8, 4-to-16, etc.



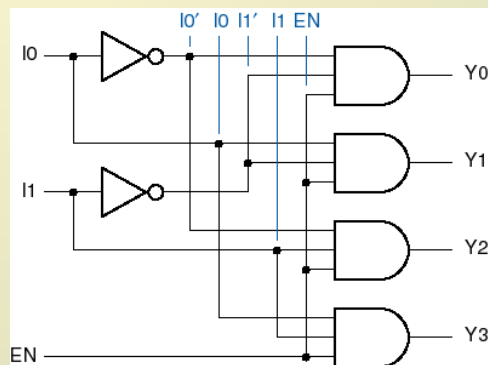
Inputs			Outputs			
EN	I1	I0	Y3	Y2	Y1	Y0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

"x" (don't care)



Descodificador 2:4

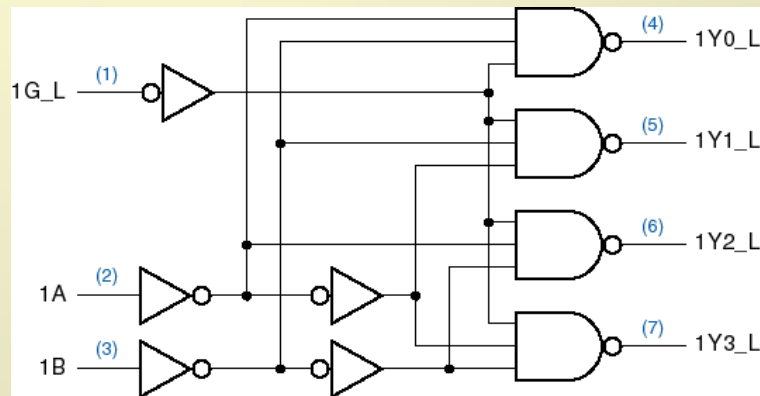
- Diagrama Lógico
- Qual o papel da entrada EN?
- Exercício: Determinar as equações das saídas



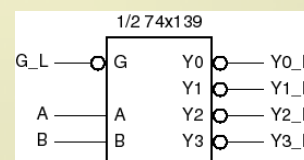
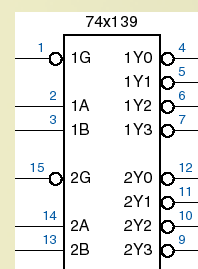
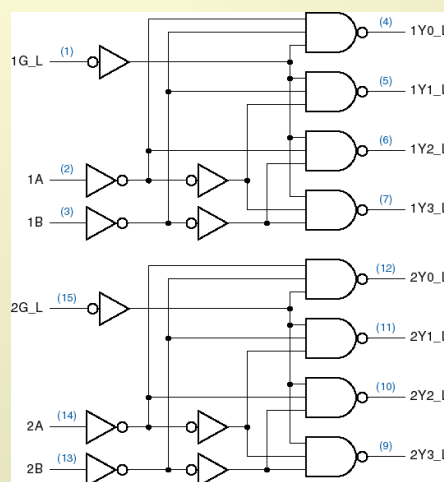


Descodificador 2:4

- Diagrama Lógico
- Saídas "active low"

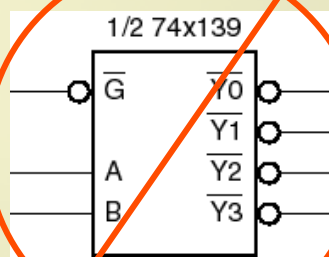
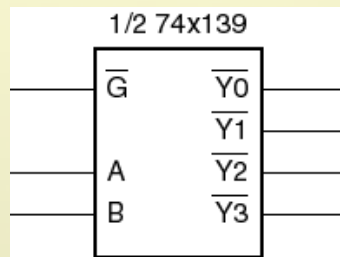


Símbolos

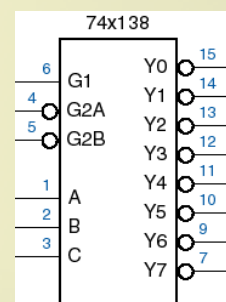
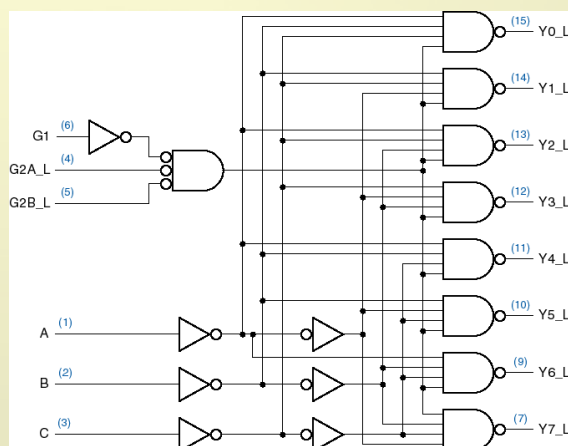




Inconsistência gráfica



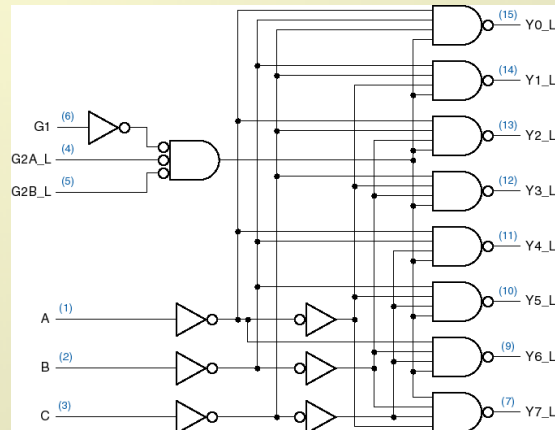
Descodificador 3:8





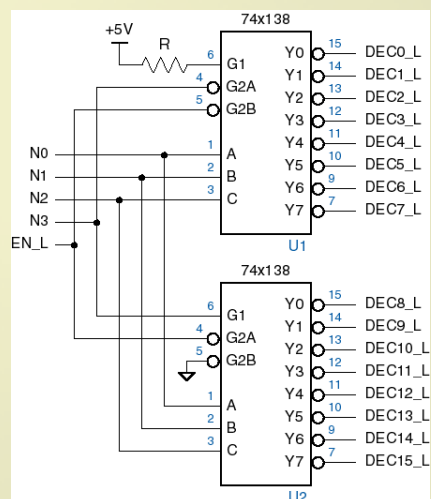
Descodificador 3:8

- Exercício: Determine a tabela de verdade física (funcional)



Cascata de decodificadores

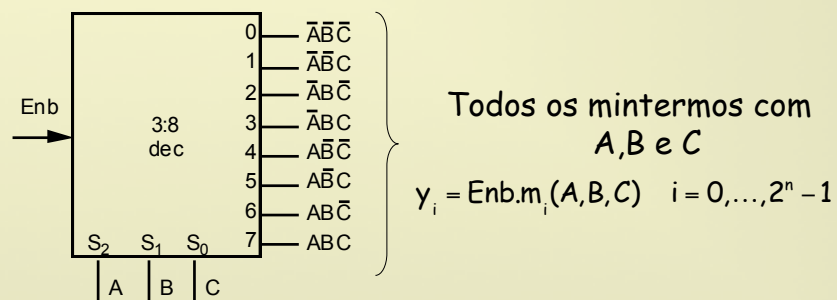
- 4:16 a partir de 2x(3:8)
- Exercício:
 - Construa um dec. 4:16 a partir de blocos 2:4
 - Construa um dec. 5:32 com blocos 2:4 e 3:8



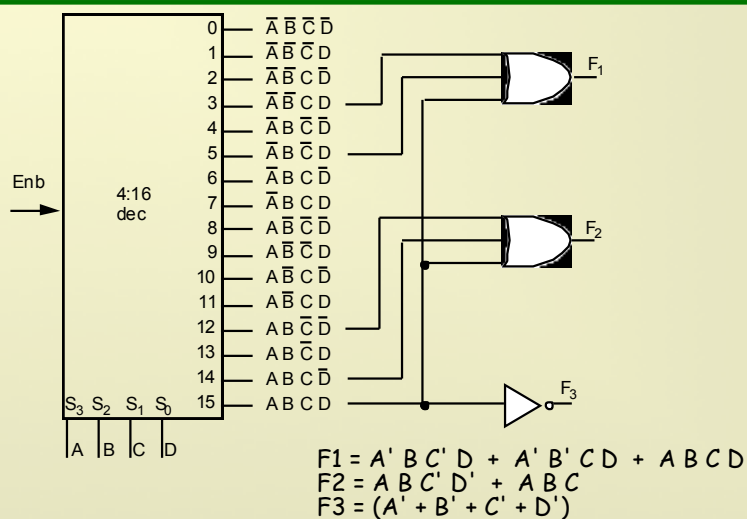


Descodificador e Mintermos

- Podemos recorrer a um bloco de descodificação para obter os mintermos necessários à implementação duma função booleana genérica



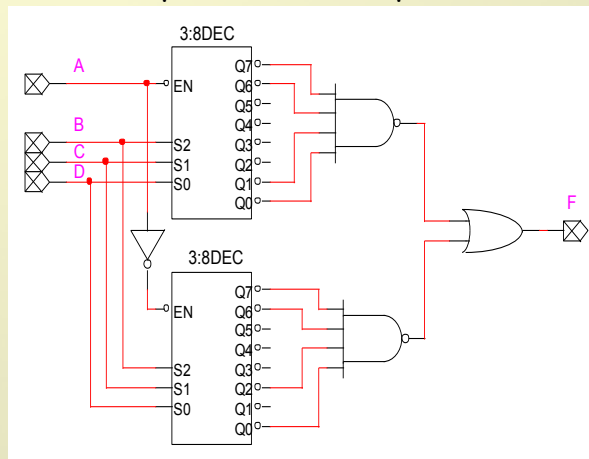
Exemplo





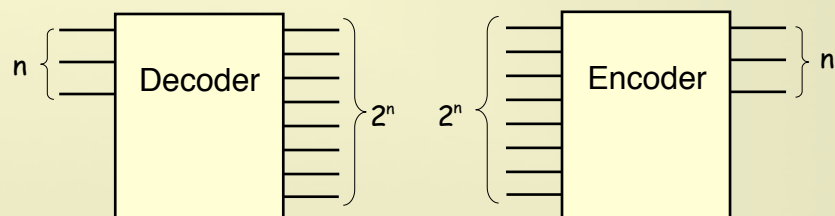
Exercício

- Determine a expressão mínima para $f(A,B,C,D)$



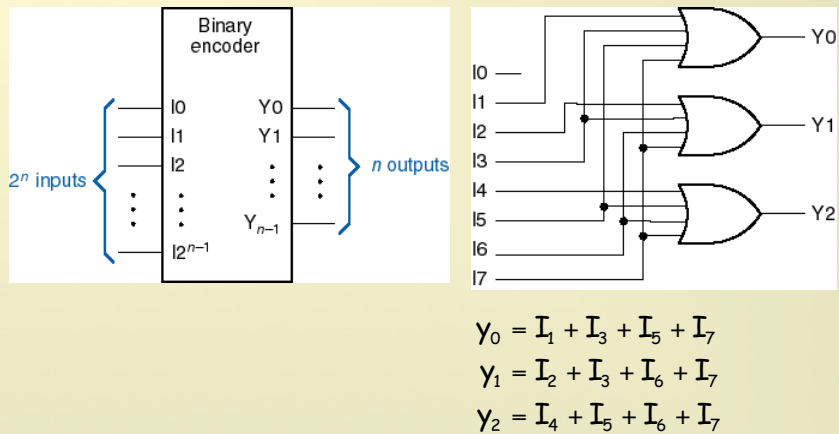
Codificadores

Funcionalmente inversos dos decodificadores



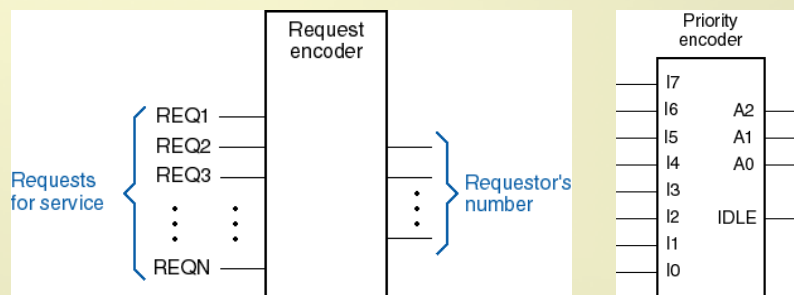


Codificadores binários



O problema da Prioridade

- Exercício: veja o que acontece no exemplo anterior quando por exemplo I_3 e I_5 estão activos.

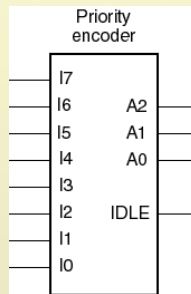


- Situações de conflito "arbitram-se" mediante uma estratégia de prioridade de atendimento



Codificador de Prioridade

- Redefinição da lógica interna através das variáveis intermédias H_n



$$H_7 = I_7$$

$$H_6 = I_6 \cdot \bar{I}_7$$

$$H_5 = I_5 \cdot \bar{I}_6 \cdot \bar{I}_7$$

⋮

$$H_0 = I_0 \cdot \bar{I}_1 \cdot \bar{I}_2 \cdots \bar{I}_6 \cdot \bar{I}_7$$

$$IDLE = \sum_{n=0}^{2^n-1} I_n = \prod_{n=0}^{2^n-1} \bar{I}_n$$

$$A_0 = H_1 + H_3 + H_5 + H_7$$

$$A_1 = H_2 + H_3 + H_6 + H_7$$

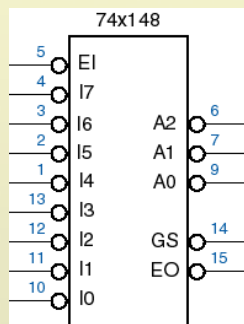
$$A_2 = H_4 + H_5 + H_6 + H_7$$

- Qual o significado da saída IDLE?
- Qual o código de saída quando I3 e I5 estão activos?



O Codificador de Prioridade "74148"

- Active-low I/O
- Enable Input
- "Got Something"
- Enable Output

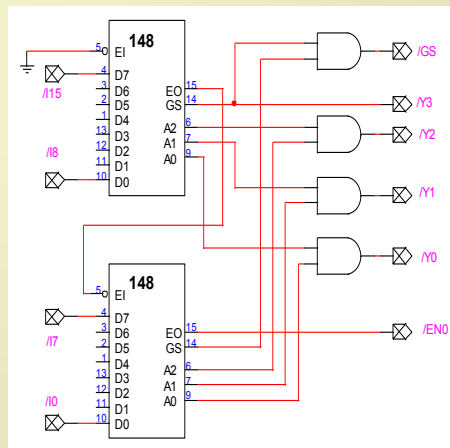


Inputs										Outputs				
EI_L	I0_L	I1_L	I2_L	I3_L	I4_L	I5_L	I6_L	I7_L		A2_L	A1_L	A0_L	GS_L	EO_L
1	x	x	x	x	x	x	x	x		1	1	1	1	1
0	x	x	x	x	x	x	x	0		0	0	0	0	1
0	x	x	x	x	x	x	0	1		0	0	1	0	1
0	x	x	x	x	x	0	1	1		0	1	0	0	1
0	x	x	x	x	0	1	1	1		0	1	1	0	1
0	x	x	x	0	1	1	1	1		1	0	0	0	1
0	x	x	0	1	1	1	1	1		1	0	1	0	1
0	x	0	1	1	1	1	1	1		1	1	0	0	1
0	0	1	1	1	1	1	1	1		1	1	1	0	1
0	1	1	1	1	1	1	1	1		1	1	1	1	0



Problema

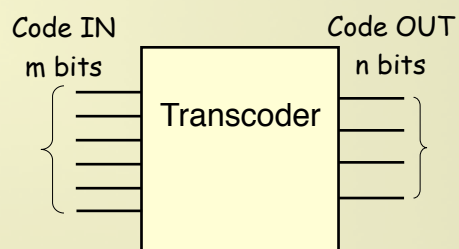
- Construir um Codificador de Prioridade 16:4 com base em codificadores do tipo 74148 e lógica adicional.



Transcodificação

- Conversão de código
- Exemplos

Code IN	Code OUT
BCD8421	D+3
BCD8421	7-Segs
BCD8421	Gray
Outros...	

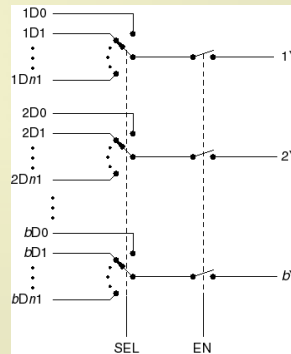
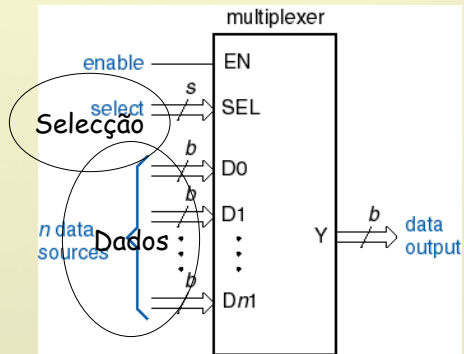


- Exercício: Projecte um transcodificador BCD-7segs



Multiplexers/Demultiplexers

- Selecção de informação
- Encaminhamento



Modelo Genérico

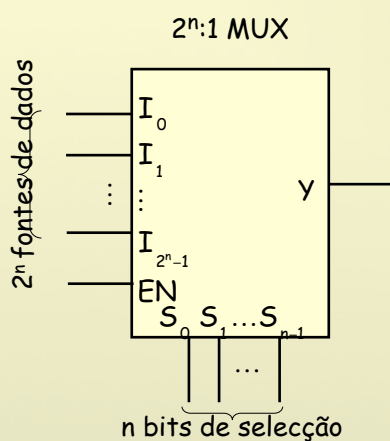


Tabela de Verdade Funcional (4:1)

EN	S ₁	S ₀	Y
0	x	x	0
1	0	0	I ₀
1	0	1	I ₁
1	1	0	I ₂
1	1	1	I ₃

Função booleana genérica

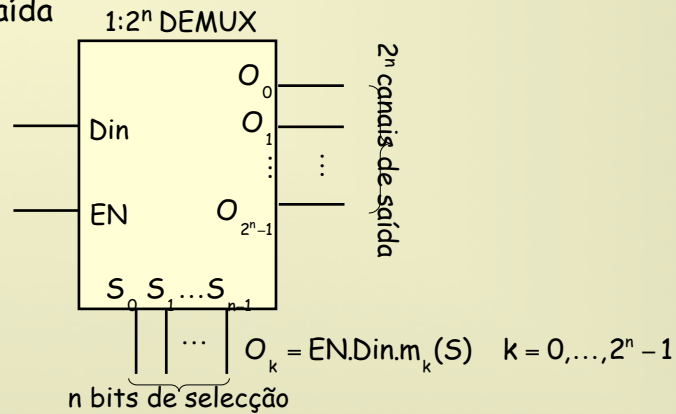
$$Y = EN \cdot \left(\sum_k m_k(S) I_k \right) \quad k = 0, \dots, 2^n - 1$$

$m_k(S)$ é o k -ésimo mintermo nas variáveis de selecção S_0, S_1, \dots, S_n



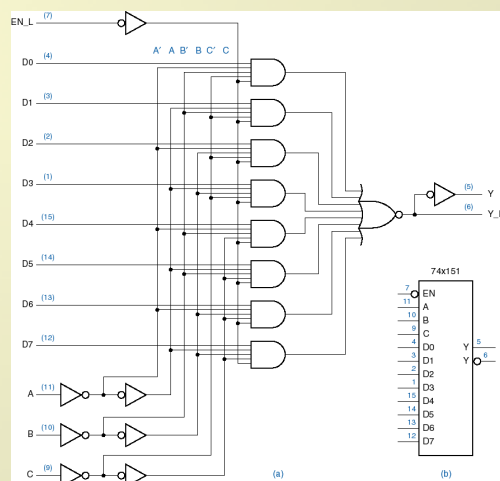
Demultiplexer

- Funcionalmente inverso do multiplexer
- 1 entrada de dados é encaminhada para um de 2^n canais de saída



Exemplo

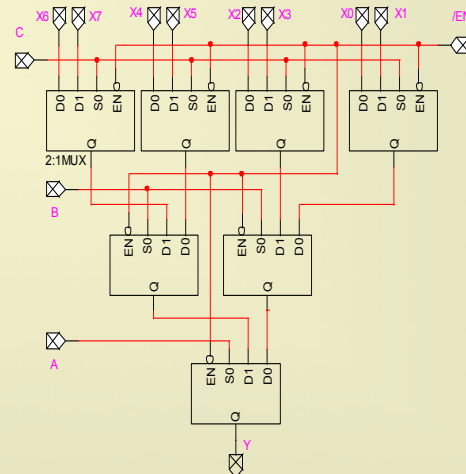
- 74151
- 8:1 MUX
- Exercício:
 - Deduza a Função booleana genérica para Y





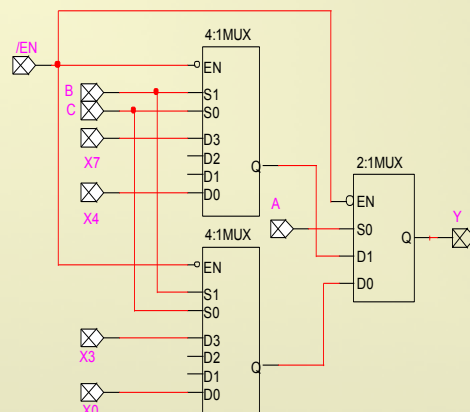
Hierarquia de multiplexagem

- Exemplo: 8:1 MUX a partir de 7x(2:1 MUX)



Hierarquia de multiplexagem

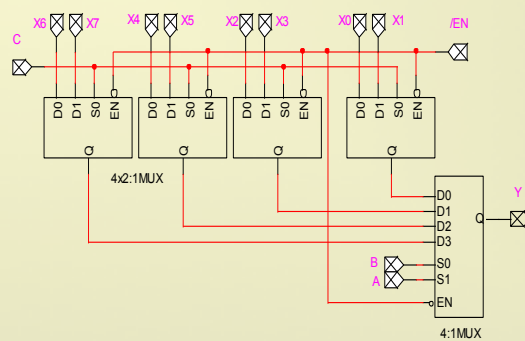
- Exemplo: 8:1 MUX a partir de 2x(4:1 MUX) + 1x(2:1 MUX)





Hierarquia de multiplexagem

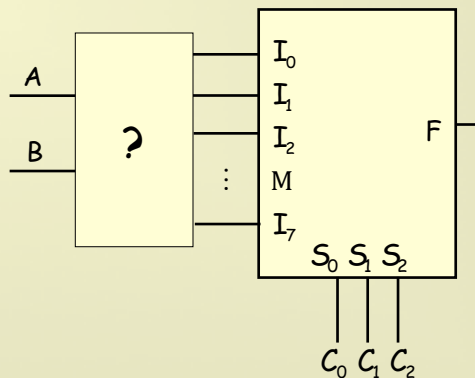
- Exemplo: 8:1 MUX a partir de 4x(2:1 MUX) + 1x(4:1 MUX)



Unidade Funcional Lógica

- Exemplo de selecção de operações lógicas a partir dum código de operação

C_2	C_1	C_0	F
0	0	0	1
0	0	1	$A+B$
0	1	0	AB
0	1	1	$A \oplus B$
1	0	0	$A \odot B$
1	0	1	\overline{AB}
1	1	0	$A+B$
1	1	1	0

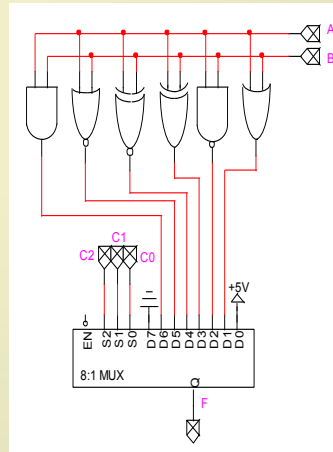




Unidade Funcional Lógica

- Exemplo de selecção de operações lógicas a partir dum código de operação

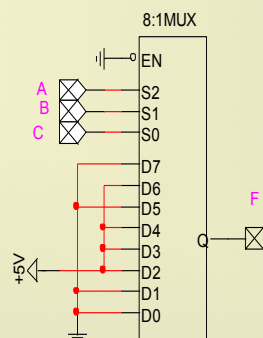
C_2	C_1	C_0	F
0	0	0	1
0	0	1	$A + B$
0	1	0	$\overline{A \cdot B}$
0	1	1	$A \oplus B$
1	0	0	$A \text{ e } B$
1	0	1	$A \cdot B$
1	1	0	$\overline{A + B}$
1	1	1	0



Funções genéricas com MUX's

- Mapeamento directo da tabela de verdade
 - $D_k = F_k$
 - Selecção = Entradas

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0





Funções genéricas com MUX's

- Caso geral
 - $D_k = F_k +$ lógica adicional a partir das entradas
 - Selecção = subconjunto das entradas
- Exemplo mapeando $n-1$ variáveis independentes nas entradas de selecção

$n-1$ variáveis independentes nas entradas de selecção

I1	I2	..	In	F			
...	0	0	0	1	1
...	1	0	1	0	1
				0	In	\bar{In}	1

Possíveis valores para F em função de In



Exemplo

- Implementar a função F a partir dum MUX 8:1

$$F(A,B,C,D) = \sum m(1,3,6,7,8,11,12,14)$$

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

D

B

1. Mapear variáveis de selecção. Neste caso A, B e C. (Outras escolhas seriam possíveis)

2. Determinar a lógica associada a cada entrada do mux em função das restantes variáveis independentes (neste caso D)

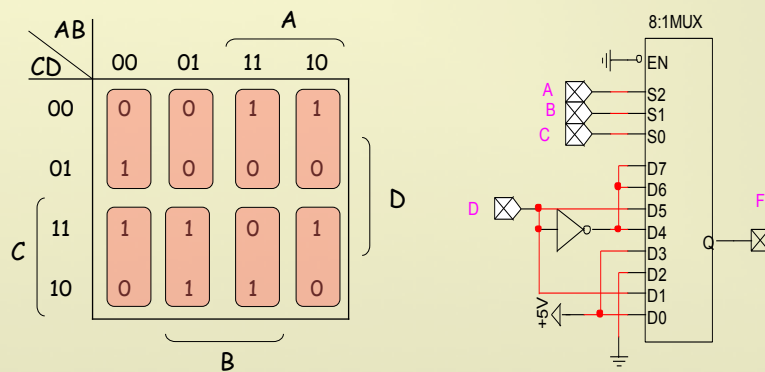
Regiões da tabela de verdade onde A, B e C são constantes. Não confundir com os implicantes no Mapa de Karnaugh



Exemplo

- Implementar a função F a partir dum MUX 8:1

$$F(A,B,C,D) = \sum m(1,3,6,7,8,11,12,14)$$



Exercício

- Implementar a função F a partir dum MUX 4:1 e lógica adicional

$$F(A,B,C,D) = \sum m(1,3,6,7,8,11,12,14)$$

- Escolha várias combinações das variáveis independentes para entradas de selecção e verifique que há escolhas preferenciais



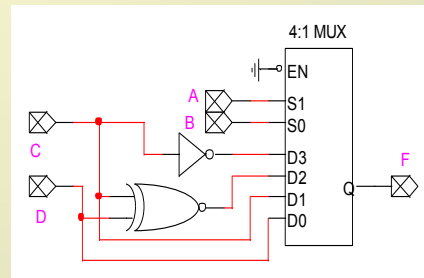
Exercício

- Escolhendo AB para entradas de selecção

$$F(A,B,C,D) = \sum m(1,3,6,7,8,11,12,14)$$

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

B



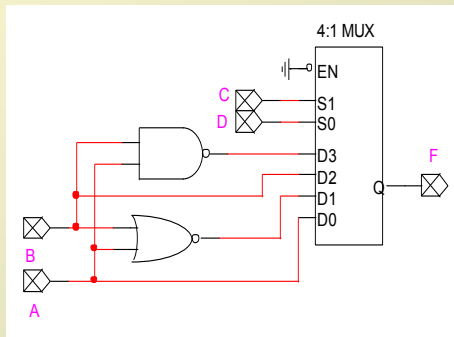
Exercício

- Escolhendo CD para entradas de selecção

$$F(A,B,C,D) = \sum m(1,3,6,7,8,11,12,14)$$

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

B





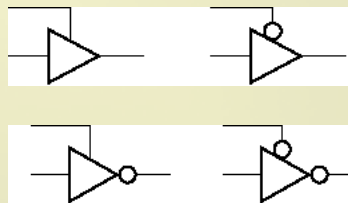
Buffers 3'state

- Saídas possíveis: LOW, HIGH e Hi-Z
- Hi-Z ou Z significa "alta impedância" vista da saída do dispositivo



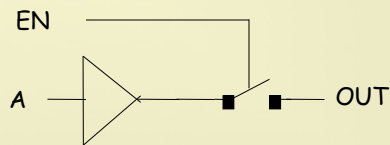
EN	A	OUT
L	L	Hi-Z
L	H	Hi-Z
H	L	L
H	H	H

Alternativas



Alta-Impedância

- Alta-Impedância significa resistência praticamente infinita, sem ligação.
- Modelo do interruptor

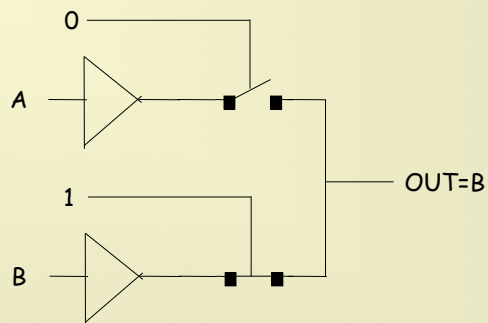


- Quando EN=0 o interruptor estão aberto determinando uma resistência eléctrica infinita vista do lado da saída



Conexão física das saídas

- Partilha de saídas

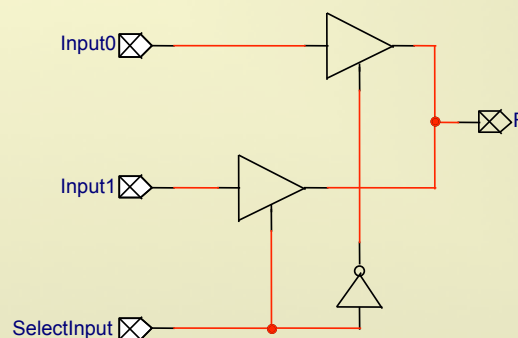


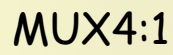
- Fisicamente podemos ligar as saídas dos buffers desde que o controlo dos "enables" seja devidamente temporizado



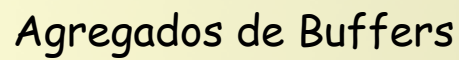
Multiplexagem revisitada

- Estratégias eficientes de multiplexagem
- Exemplo 2:1 MUX





-

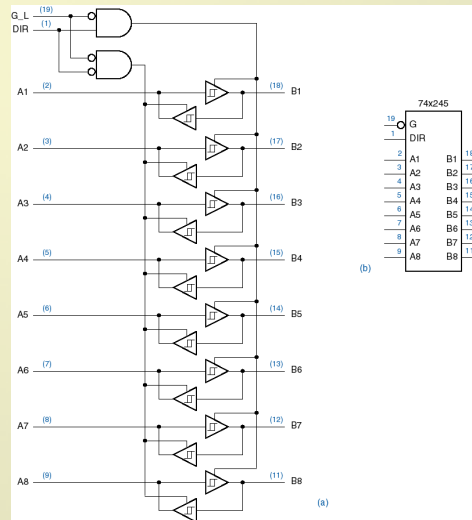


-
- Figure 1 shows the logic diagram of a 74x541 8-to-3 priority encoder. The diagram includes the 74x541 chip and a logic circuit. The 74x541 chip has inputs A1 through A8 and outputs Y1 through Y3. The logic circuit uses an AND gate and seven inverters. The inputs to the AND gate are G1_L (1) and G2_L (19). The outputs of the inverters are connected to the inputs of the AND gate. The outputs of the AND gate are connected to the inputs of the priority encoder. The outputs of the priority encoder are Y1 (18), Y2 (17), and Y3 (16).



Agregados de Buffers

- "Transceiver"



Multiplexagem em "BUS"

- Generalização da estrutura anterior com blocos de buffers. Exemplo MUX4:1 para palavras de 4 bits

