Spina

PCS3515 – Sistemas Digitais

Blocos Básicos

- Decodificadores e Codificadores -

Seções 6.4 e 6.5 – livro texto

Com apoio do material dos demais professores

2016/1

Spina

Decodificadores

- Decodificador ou *Decoder*
  - É um Bloco Lógico Funcional (ou Bloco Combinatório Lógico) que possui *n* entradas e (até) 2^n saídas
  - Para cada combinação de valores das *n* entradas apenas uma saída é ativada

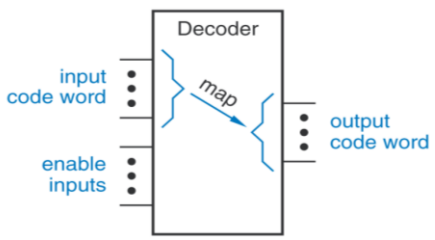


Figure 6-31
Decoder circuit structure.

Spina

Decodificadores 2

- As entradas formam uma palavra binária de n bits
- A palavra pode assumir os valores de 0 a 2^n -1
- As 2^n saídas são numeradas de 0 a 2^n -1
- A saída ativada corresponde àquela cujo índice corresponde ao valor da palavra binária de entrada
- Exemplo
  - Entradas = 011 (3_{10}); ativa-se a saída S3
  - Entradas = 101 (5_{10}); ativa-se a saída S5

Spina

Decodificadores 3

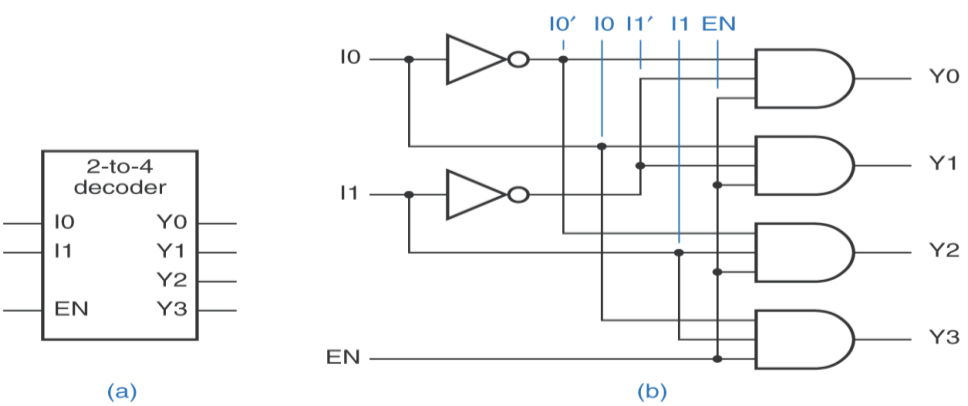


Figure 6-32
A 2-to-4 decoder: (a) inputs and outputs; (b) logic diagram.

Decodificadores 4

```
library IEEE;
use IEEE.std_logic_1164.all;

entity V2to4dec is
  port (I0, I1, EN: in STD_LOGIC;
        Y0, Y1, Y2, Y3: out STD_LOGIC );
end V2to4dec;

architecture V2to4dec_s of V2to4dec is
  signal NOTI0, NOTI1: STD_LOGIC;
  component inv port (I: in STD_LOGIC; O: out STD_LOGIC ); end component;
  component and3 port (I0, I1, I2: in STD_LOGIC; O: out STD_LOGIC ); end component;
begin
  U1: inv port map (I0,NOTI0);
  U2: inv port map (I1,NOTI1);
  U3: and3 port map (NOTI0,NOTI1,EN,Y0);
  U4: and3 port map (   I0,NOTI1,EN,Y1);
  U5: and3 port map (NOTI0,   I1,EN,Y2);
  U6: and3 port map (   I0,   I1,EN,Y3);
end V2to4dec_s;
```

Table 6-13

VHDL structural program for the decoder in Figure 6-32.

Spina

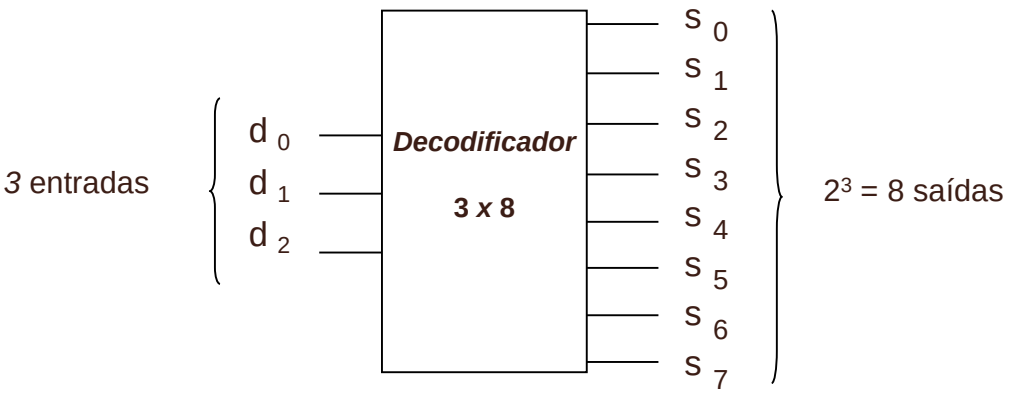
Notação

- Símbolo funcional (convenção):
- Entradas à esquerda
- Saídas à direita
- Índice menor indica bit menos significativo na palavra binária
- Denominação
  - Decodificador 3 por 8
  - Decodificador 3 x 8

Spina

Spina

Representação

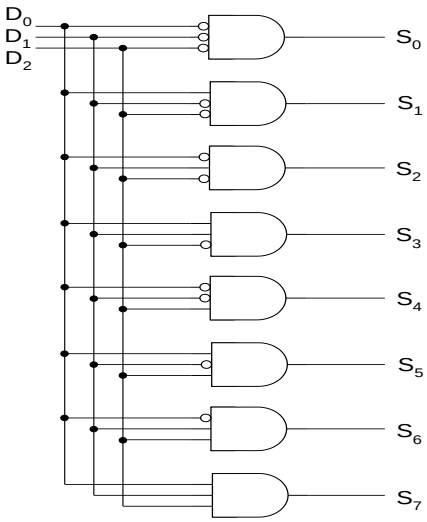


Spina

Dec. – Tabela da Verdade

ENTRADAS			SAÍDAS							
d_2	d_1	d_0	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	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Dec – Circuito interno



$S_0 = D_2' .D_1' .D_0'$

$S_1 = D_2' .D_1' .D_0$

$S_2 = D_2' .D_1 .D_0'$

$S_3 = D_2' .D_1 .D_0$

$S_4 = D_2 .D_1' .D_0'$

$S_5 = D_2 .D_1' .D_0$

$S_6 = D_2 .D_1 .D_0'$

$S_7 = D_2 .D_1 .D_0$

Spina

Dec – Entradas adicionais

- *Enable*
  - Permite habilitar/desabilitar o bloco todo
    - SE *Enable* ativo
      - Funcionamento normal: apenas uma saída ativa
    - SE *Enable* inativo
      - Nenhuma saída ativa, independente do código nas entradas de endereço
  - Atua em todos os *mintermos*

Spina

Dec Tabela Verdade

Decodificador 3 por 8, com ***ENABLE*** (EN)

ENTRADAS				SAÍDAS							
EN	d_2	d_1	d_0	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0
0	X	X	X	0	0	0	0	0	0	0	0

Spina

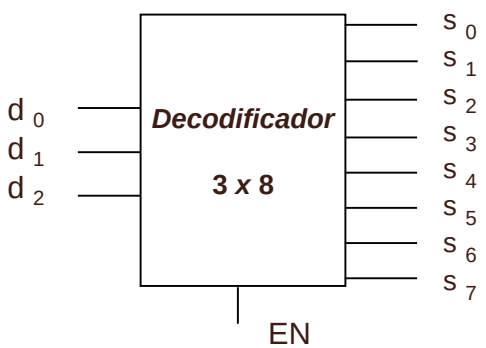
Decodificadores – Com ***Enable***

$S_0 = EN \cdot D_2' \cdot D_1' \cdot D_0'$

.

..

$S_7 = EN \cdot D_2 \cdot D_1 \cdot D_0$



Spina

Dec – Saídas invertidas

- Decodificadores podem ter as saídas invertidas ou complementadas
  - Diz-se saídas *active-low* – ativas em ZERO
    - Saídas inativas em 1, a única saída ativa = 0
  - Implicações
    - na tabela verdade: inversão nas saídas
    - no circuito: uso de NAND
    - Nomenclatura: S_i' ou
    - Cada saída é um maxtermo: $S_i' = M_i$

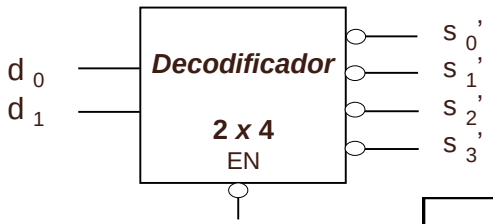
Spina

Dec – Entradas *active low*

- Entrada ENABLE também pode ser *active-low*
  - SE Enable'=0, circuito habilitado
  - SE Enable'=1, saídas todas desabilitadas
  - Implicações
    - na tabela verdade: inversão na entrada
    - no circuito: uso de inversor
    - Nomenclatura: EN' ou $\overline{EN}$

Spina

Dec – Saídas active low



Entradas			Saídas			
EN'	d_1	d_0	S'_3	S'_2	S'_1	S'_0
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1
1	X	X	1	1	1	1

Spina

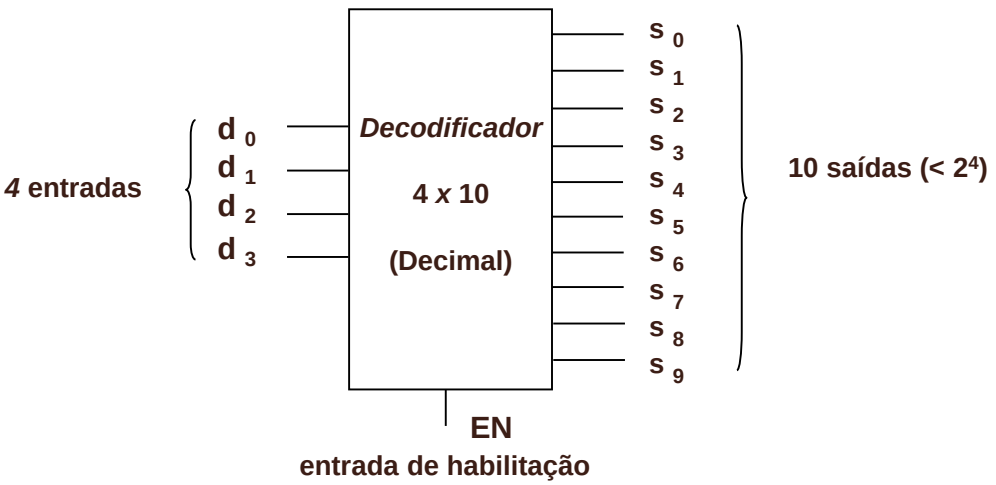
Dec – Decimal

- Decodificador BCD para decimal
  - Entrada: 1 dígito BCD (4 bits)
    - Palavras válidas 0000 → 1001
  - Saída: 10 saídas ($<2^4$)
    - Saída ativa: corresponde à palavra de entrada
    - Saídas inativas (todas):

      SE *Enable* = Falso **OU SE** palavra de entrada >1001

Spina

Dec – Decimal



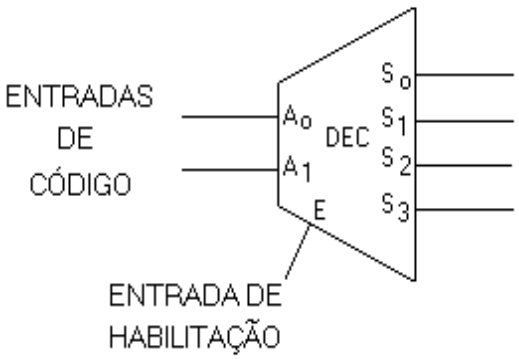
Spina

Dec – Decimal

EN	d_3	d_2	d_1	d_0	S_9	S_8	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	1	0	0
1	0	0	1	1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	0	0	1	0	0	0	0
1	0	1	0	1	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	1	0	0	0	0	0	0
1	0	1	1	1	0	0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	1	0	0	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
	.	.	.	.										
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	X	X	X	X	0	0	0	0	0	0	0	0	0	0

Spina

Dec – Símbolo alternativo



ENTRADAS			SAÍDAS			
E	A_1	A_0	S_0	S_1	S_2	S_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Spina

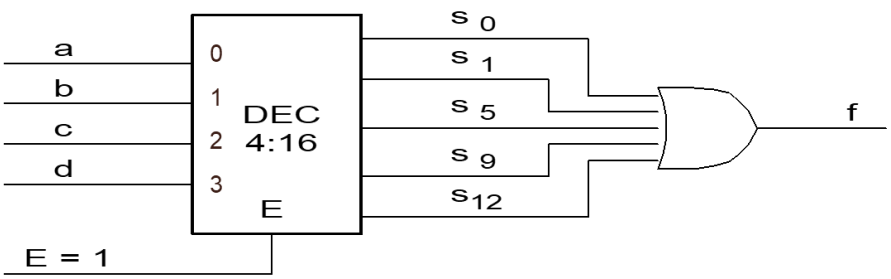
Dec – Uso

- Síntese de funções de chaveamento
  - Saídas dos *Decoders* são os *mintermos* para as variáveis de entrada
  - 1ª.forma canônica: Função = $\Sigma_{\text{mintermos}}$
- Exemplo:
  - F(d,c,b,a) = Σ(0,1,5,9,12)

Spina

Dec – Síntese de funções *

- F(d,c,b,a) = Σ(0,1,5,9,12)



Spina

Dec – 74x138

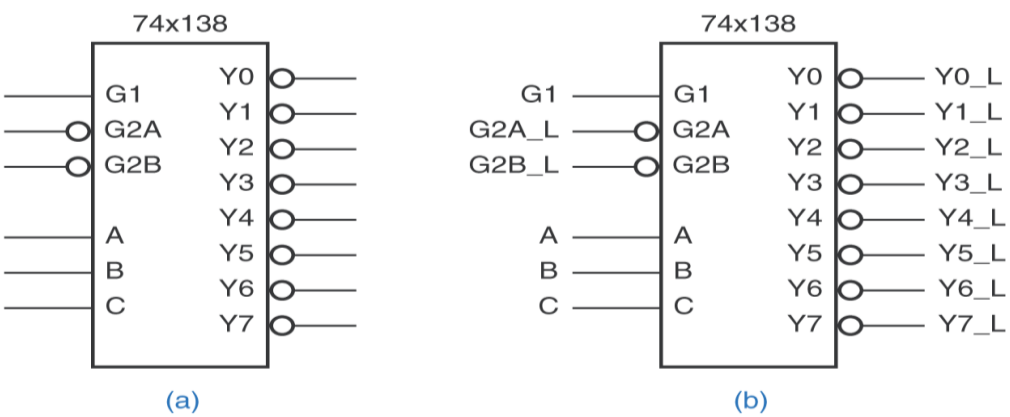


Figure 6-34

Logic symbol for the 74×138 3-to-8 decoder: (a) conventional symbol; (b) default signal names associated with external pins.

Spina

Dec – 74x138

```
library IEEE;
use IEEE.std_logic_1164.all;

entity V74x138 is
    port (G1, G2A_L, G2B_L: in STD_LOGIC;       -- enable inputs
          A: in STD_LOGIC_VECTOR (2 downto 0);   -- select inputs
          Y_L: out STD_LOGIC_VECTOR (0 to 7) );  -- decoded outputs
 end V74x138;

architecture V74x138_a of V74x138 is
  signal Y_L_i: STD_LOGIC_VECTOR (0 to 7);
begin
  with A select Y_L_i <=
    "01111111" when "000",
    "10111111" when "001",
    "11011111" when "010",
    "11101111" when "011",
    "11110111" when "100",
    "11111011" when "101",
    "11111101" when "110",
    "11111110" when "111",
    "11111111" when others;
  Y_L <= Y_L_i when (G1 and not G2A_L and not G2B_L)='1' else "11111111";
end V74x138_a;
```

Table 6-14

Dataflow-style VHDL program for a 74×138-like 3-to-8 binary decoder.

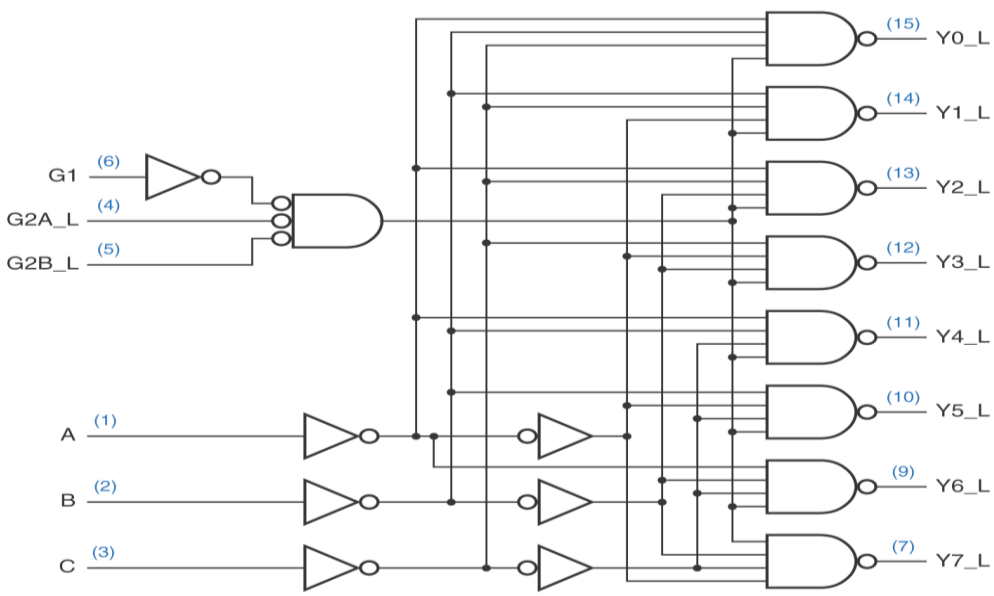


Figure 6-35

Logic diagram for the 74×138 3-to-8 decoder

Inputs						Outputs							
G1	G2A_L	G2B_L	C	B	A	Y7_L	Y6_L	Y5_L	Y4_L	Y3_L	Y2_L	Y1_L	Y0_L
0	x	x	x	x	x	1	1	1	1	1	1	1	1
x	1	x	x	x	x	1	1	1	1	1	1	1	1
x	x	1	x	x	x	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1

Table 6-6

Truth table for a 74×138 3-to-8 decoder.

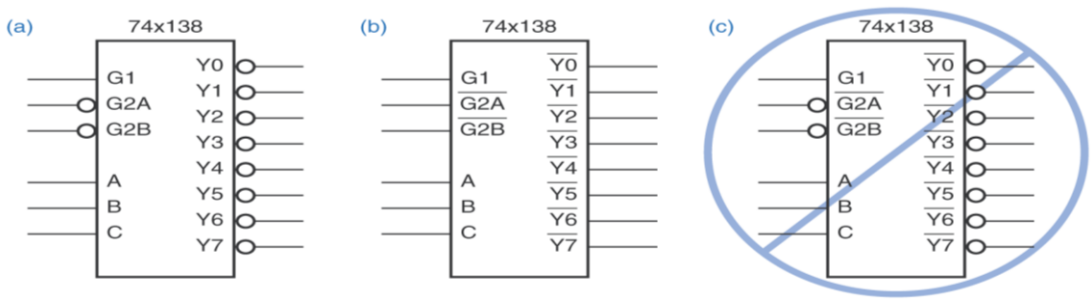
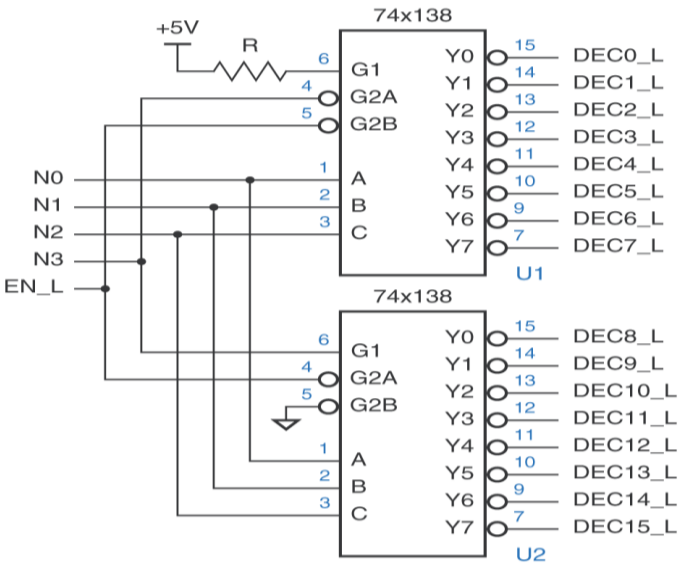


Figure 6-36

Logic symbols for the 74×138: (a) preferred symbol; (b) correct but to be avoided; (b) incorrect because of double negations.

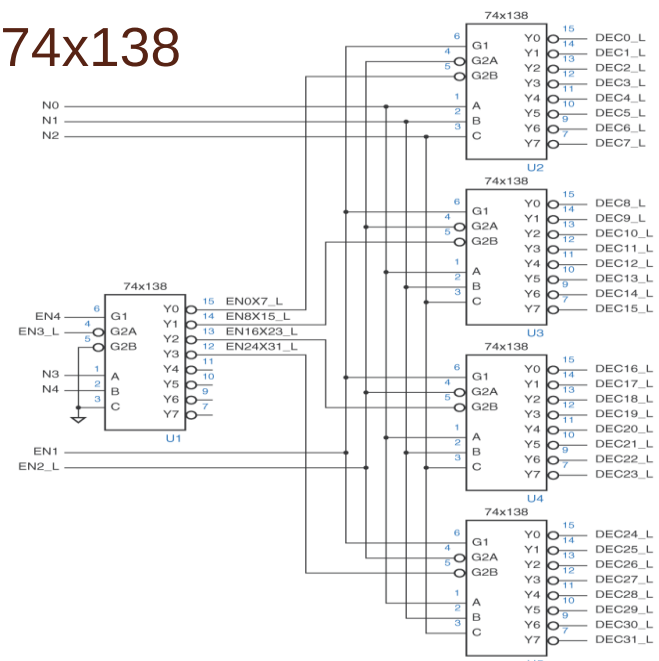
Spina

4-para-16 com 74x138



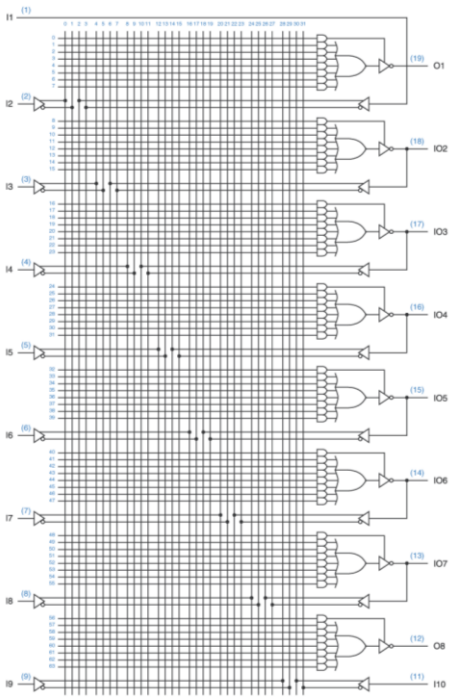
Spina

5-para-32 com 74x138



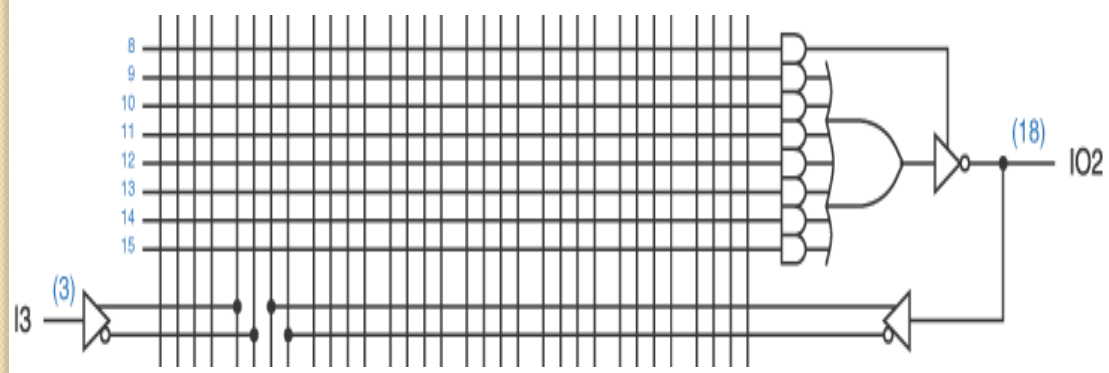
Spina

PAL16L8



Spina

PAL16L8



Codificadores

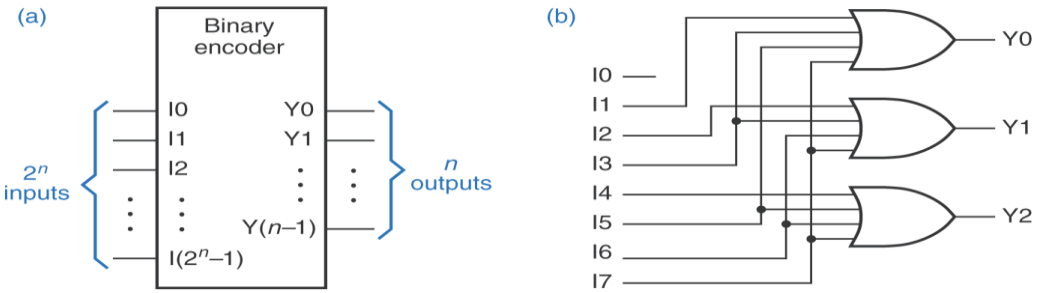


Figure 6-45

Binary encoder: (a) general structure; (b) 8-to-3 encoder.

Spina

Codificadores

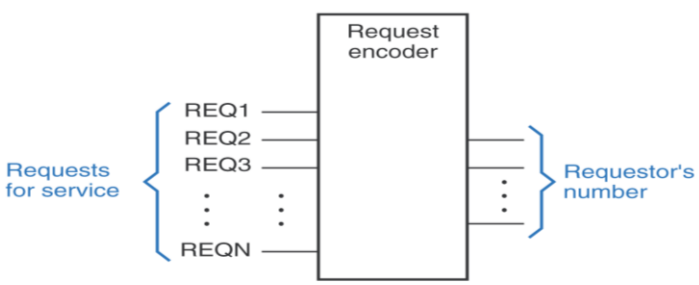


Figure 6-46

Spina

Spina

Codificador de Prioridade

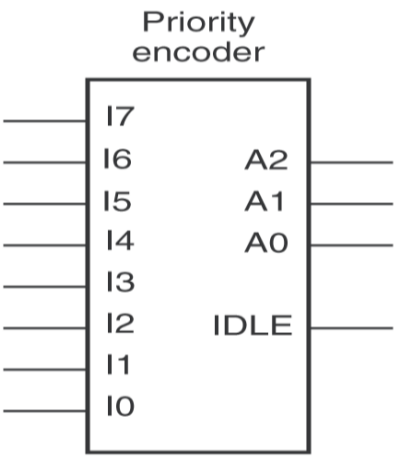
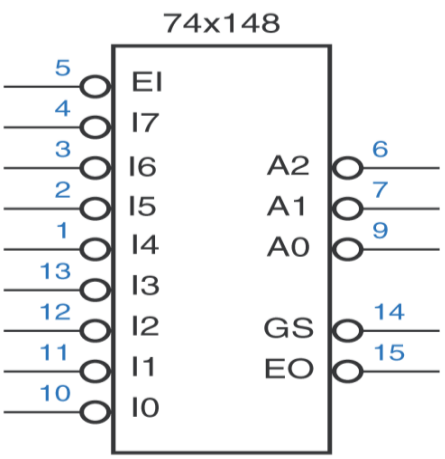


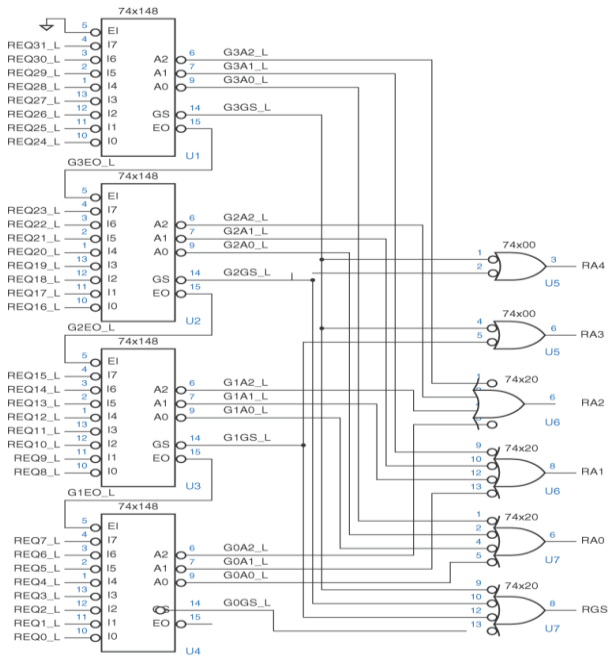
Figure 6-47

Spina

Codificador de Prioridade 74x148



Spina



Spina

```
library IEEE;
use IEEE.std_logic_1164.all;

entity V74x148 is
    port (
        EI_L: in STD_LOGIC;
        I_L: in STD_LOGIC_VECTOR (7 downto 0);
        A_L: out STD_LOGIC_VECTOR (2 downto 0);
        EO_L, GS_L: out STD_LOGIC
    );
end V74x148;

architecture V74x148p of V74x148 is
  signal EI: STD_LOGIC;                       -- active-high version of input
  signal I: STD_LOGIC_VECTOR (7 downto 0); -- active-high version of inputs
  signal EO, GS: STD_LOGIC;                   -- active-high version of outputs
  signal A: STD_LOGIC_VECTOR (2 downto 0); -- active-high version of outputs
begin
  process (EI_L, I_L, EI, EO, GS, I, A)
  variable j: INTEGER range 7 downto 0;
  begin
    EI <= not EI_L; -- convert input
    I <= not I_L;   -- convert inputs
    EO <= '1'; GS <= '0'; A <= "000";
    if (EI)='0' then EO <= '0';
    else for j in 7 downto 0 loop
        if I(j)='1' then
          GS <= '1'; EO <= '0'; A <= CONV_STD_LOGIC_VECTOR(j,3);
          exit;
        end if;
      end loop;
    end if;
    EO_L <= not EO; -- convert output
    GS_L <= not GS; -- convert output
    A_L <= not A;   -- convert outputs
  end process;
end V74x148p;
```

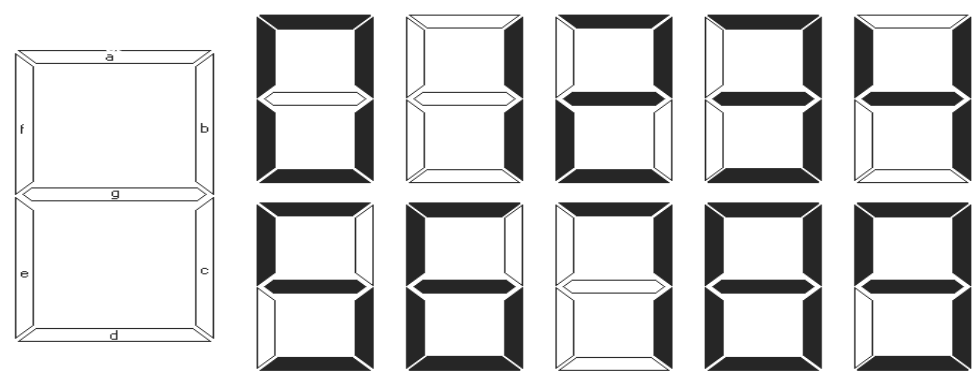
Spina

Codificadores – 7 segmentos

- Codificadores ou Transcodificadores
  - de BCD para 7 segmentos
  - de binário para 7 segmentos
- Entradas: 4 bits
- Saídas: 7 bits, código 7 segmentos
- Uso: acionamento de *displays* de 7 segmentos

Spina

Codificadores – 7 segmentos

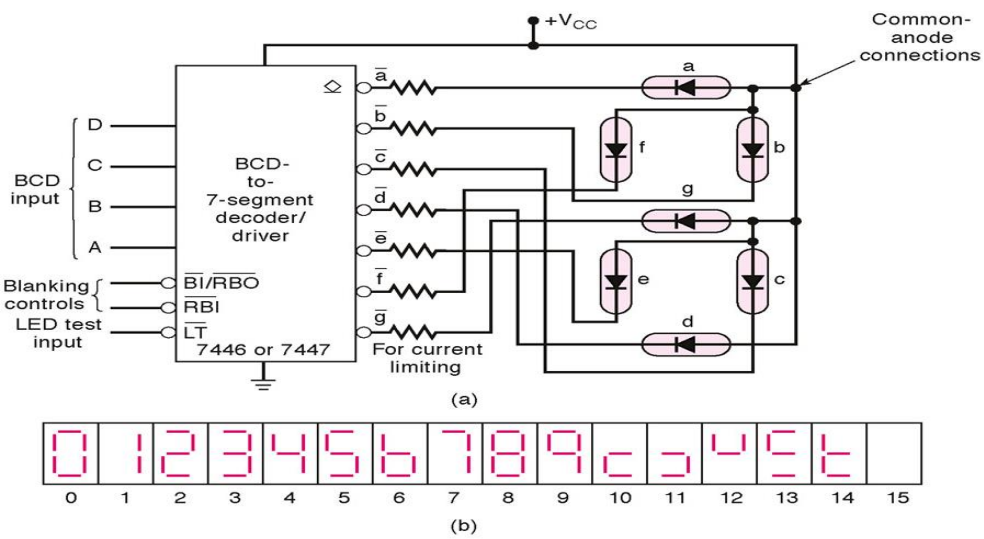


Codificadores: BCD – 7 segmentos

Decimal	EN	d_3	d_2	d_1	d_0	a	b	c	d	e	f	g
0	1	0	0	0	0	1	1	1	1	1	1	
1	1	0	0	0	1		1	1				
2	1	0	0	1	0	1	1		1	1		1
3	1	0	0	1	1	1	1	1	1			1
4	1	0	1	0	0		1	1			1	1
5	1	0	1	0	1	1		1	1		1	1
6	1	0	1	1	0	1		1	1	1	1	1
7	1	0	1	1	1	1	1	1				
8	1	1	0	0	0	1	1	1	1	1	1	1
9	1	1	0	0	1	1	1	1	1		1	1
10	1	1	0	1	0	0	0	0	0	0	0	0
		.	.	.	.							
15	1	1	1	1	1	0	0	0	0	0	0	0
	0	X	X	X	X	0	0	0	0	0	0	0

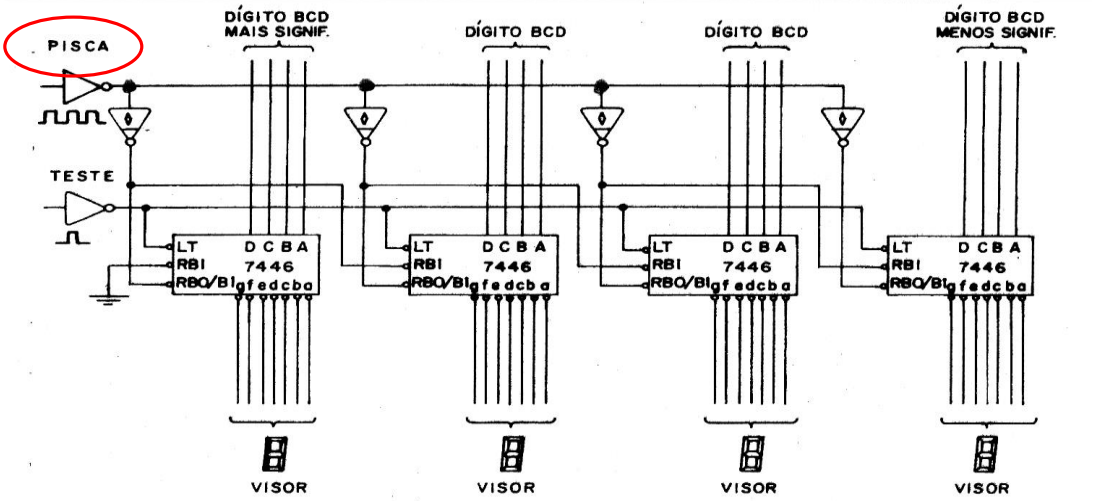
Spina

Codificadores: BCD – 7 segmentos



Spina

Codificadores: BCD – 7 segmentos



Extraído de [Fregni/Saraiva-1.995]