

Fundamentos de Sistemas Digitais

Funções de Lógica Combinacional

prof. Dr. Alexandre M. Amory
prof. Dr. Edson I. Moreno

Referências

- Sugiro estudarem nesta ordem de preferência:
- Floyd, Cap 6.
 - ppt segue a ordem deste capítulo
- D'AMORE, Roberto. VHDL: Descrição e Síntese de Circuitos Digitais. Rio de Janeiro: LTC, 2005. 259 p.
 - <http://www.ele.ita.br/~damore/vhdl/>
- Digital Design and Computer Architecture
 - Cap 5, Sec 2.8
- Free Range VHDL
- Vahid, Cap 4 – Ignorar parte com flip-flop e registradores
- PEDRONI, Volnei A. Eletrônica Digital Moderna e VHDL. Elsevier Ltda. Editora, Rio de Janeiro, RJ: 2010. 619 p.
 - Teoria: Cap 11 e 12
 - VHDL: Cap 20 e 21

ATENÇÃO

- Não se aprende alguma linguagem :
 - Escutando alguém explicar
 - Somente lendo livros
- Aprende praticando
 - Exercício EXTRA-CLASSE é **FUNDAMENTAL** !
 - Façam exercícios sugeridos e procurem outros nos livros indicados

Circuitos combinacionais básicos e representação em VHDL

• Exemplos de circuitos combinacionais

- Decodificador / Codificador
- Comparadores
- Geradores de paridade
- Multiplexador / Demultiplexador
- Somador / Subtrator
- ULA
- Multiplicadores / Divisores
- PLAs

• Exemplos de circuitos seqüenciais

- Multiplicadores / Divisores
- Registradores (deslocamento, carga paralela, acumulador, serial-paralelo)
- Contadores (binário, BCD, Johnson, Gray / up, down, up-down)
- Máquina de Estados
- Geradores de clock
- Seqüenciadores

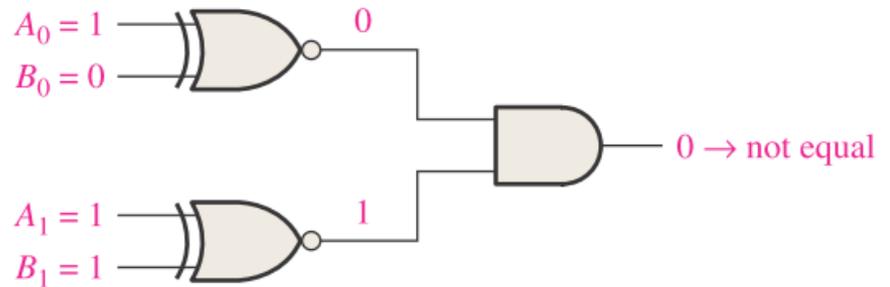
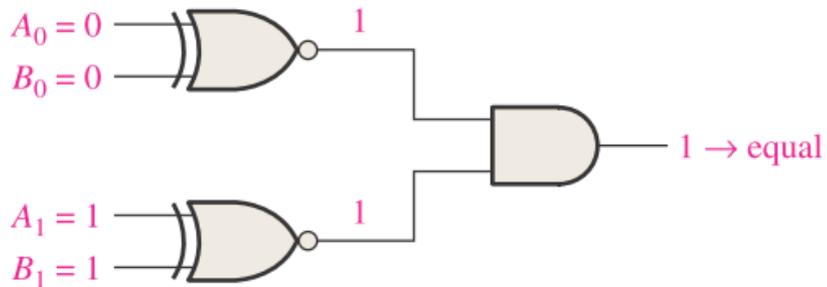
Comparador de Igualdade

Construção estrutural



| A | B | S |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

Como comparar $a[0:1]$ e $b[0:1]$?



Comparador de desigualdade

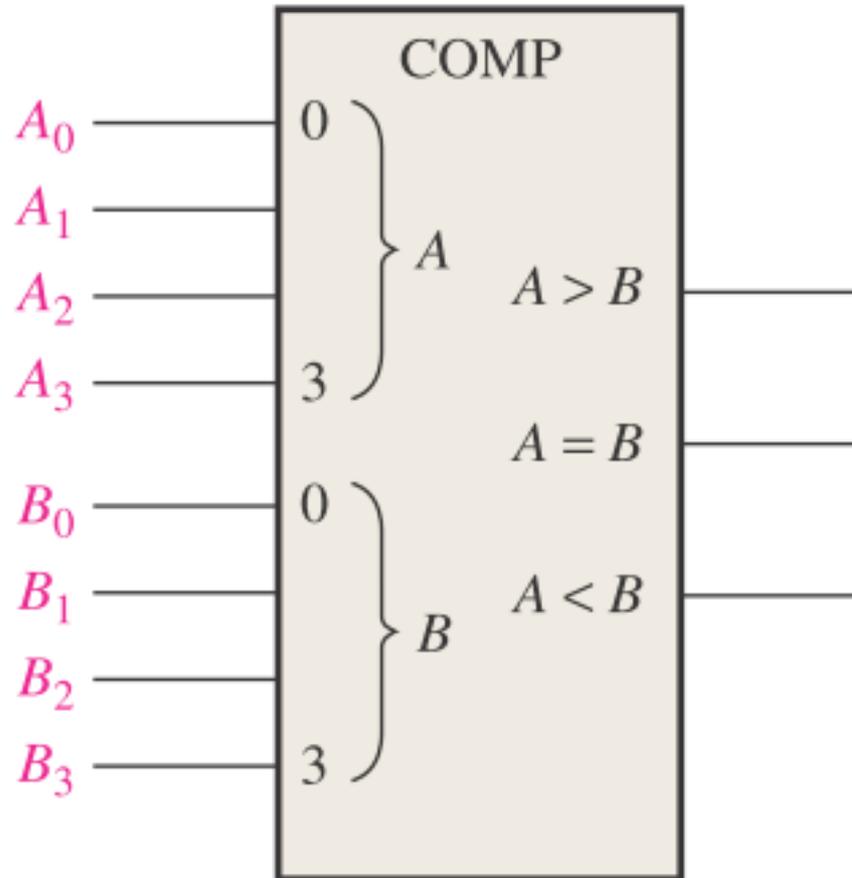


FIGURE 6-21 Logic symbol for a 4-bit comparator with inequality indication.

Comparação em VHDL

construção comportamental

```
entity comp is
```

```
...
```

```
end comp;
```

```
architecture beh of comp is
```

```
    signal a,b: std_logic_vector(1 downto 0);
```

```
    signal igual, maior, menor: std_logic;
```

```
begin
```

```
    igual <= '1' when a = b else '0';
```

```
    maior <= '1' when a > b else '0';
```

```
    menor <= '1' when a < b else '0';
```

```
end;
```

Decodificador BIN/DEC

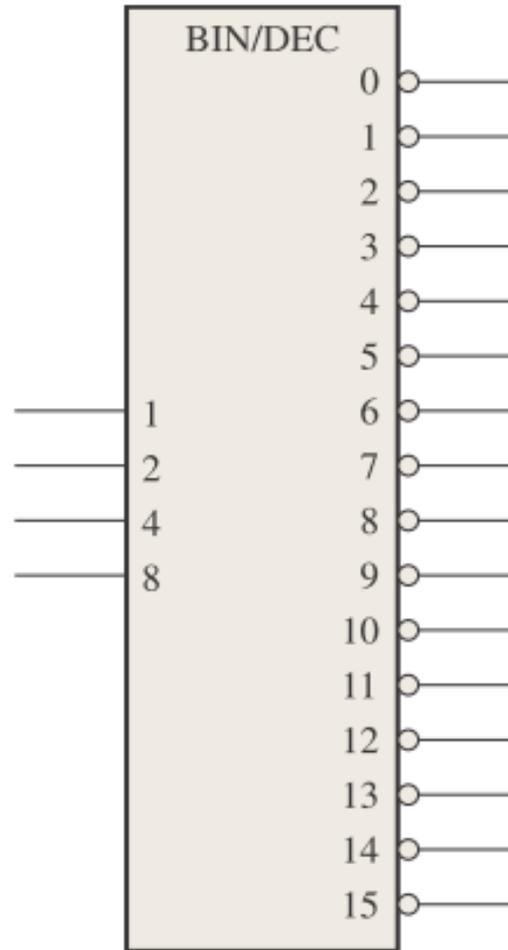


FIGURE 6-28 Logic symbol for a 4-line-to-16-line (1-of-16) decoder. Open file F06-28 to verify operation.

Decodificador BIN/DEC

- em Portas Lógicas:

- em VHDL (entrada de 2 bits):

```
decod <= "1110" when a = "00" else  
        "1101" when a = "01" else  
        "1011" when a = "10" else  
        "0111" when a = "11";
```

Decodificador com Prioridade

- Codificador com prioridade é um circuito que determina entradas preferenciais
- Conforme a ordem de preferência das entradas, no caso de haver mais de uma entrada ativa (por exemplo em 1), a entrada de maior prioridade é que determinará a codificação
- Exemplo:
 - **A prioridade máxima é de S(2) e a mínima de S(0)**

```
Y <= "11" when S(2) = '1' else  
     "10" when S(1) = '1' else  
     "01" when S(0) = '1' else  
     "00";
```



Importante haver condição *default* em atribuições e estruturas similares:
NÃO HAVENDO ESTA CONDIÇÃO IMPLICA EM HAVER MEMORIZAÇÃO DO SINAL - diferente de software! (warning *latch infered*)

Decodificador BIN/DEC

- Outras formas de descrever VHDL

```

decod <= (0=>'0', others => '1') when a = "00" else
        (1=>'0', others => '1') when a = "01" else
        (2=>'0', others => '1') when a = "10" else
        (3=>'0', others => '1') when a = "11";

```

```

-----

process (a)
begin
    case a is
        when "00" => decod <= "1110";
        when "01" => decod <= "1101";
        when "10" => decod <= "1011";
        when "11" => decod <= "0111";
        when others => report "unreachable" severity failure;
    end case;
end process;

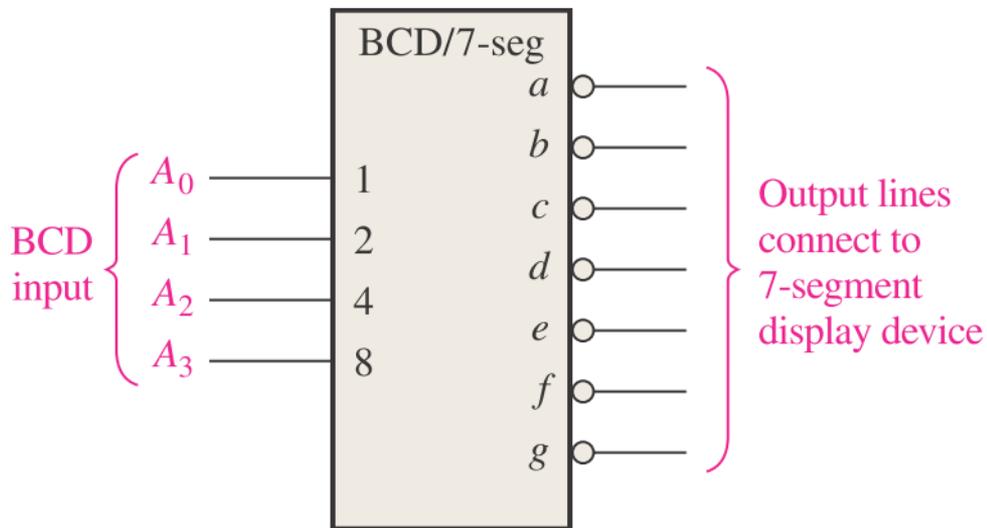
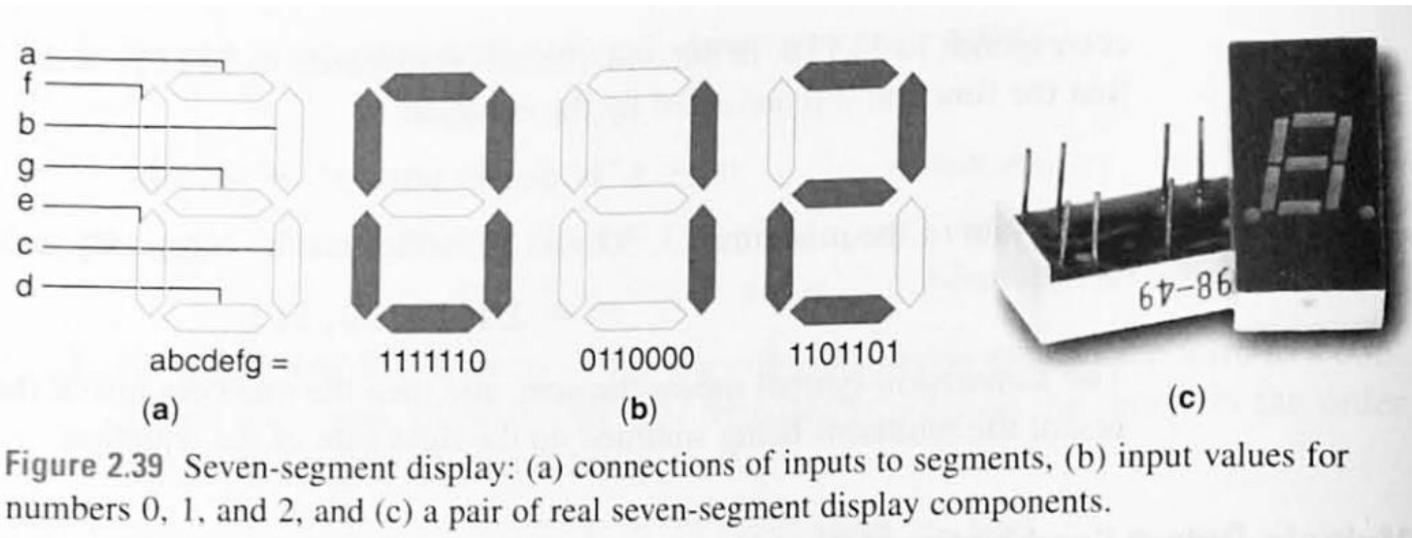
```

Decodificador BIN/DEC

- Outras formas de descrever VHDL

```
with a select decod <=  
  "1110" when "00",  
  "1101" when "01",  
  "1011" when "10",  
  "0111" when "11";
```

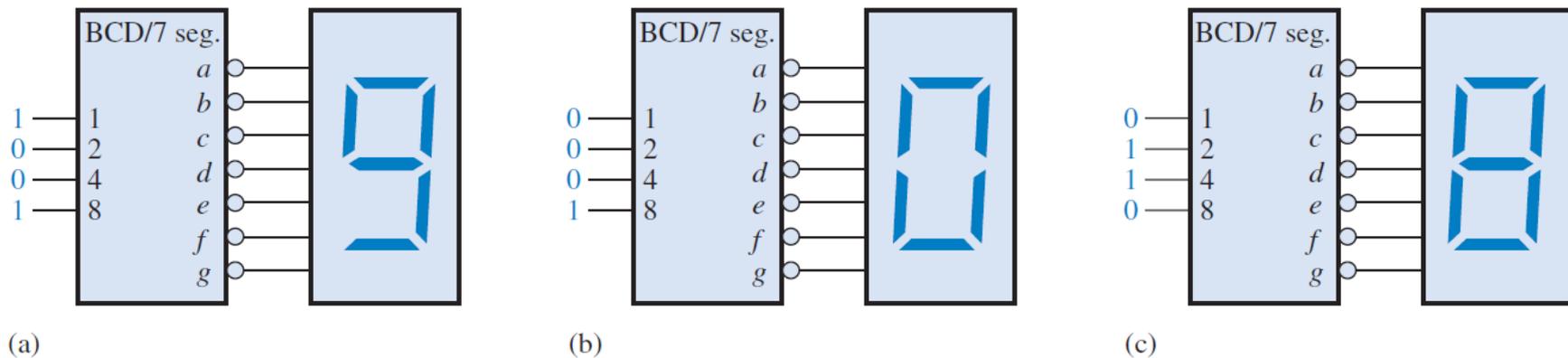
Decoder 7seg



Atividade para entregar próxima aula:

Otimizar o circuito com Karnaugh e escrever VHDL (portas/rtl) deste módulo

Fazer Descrição Comportamental



▲ FIGURA 6-91

Codificador

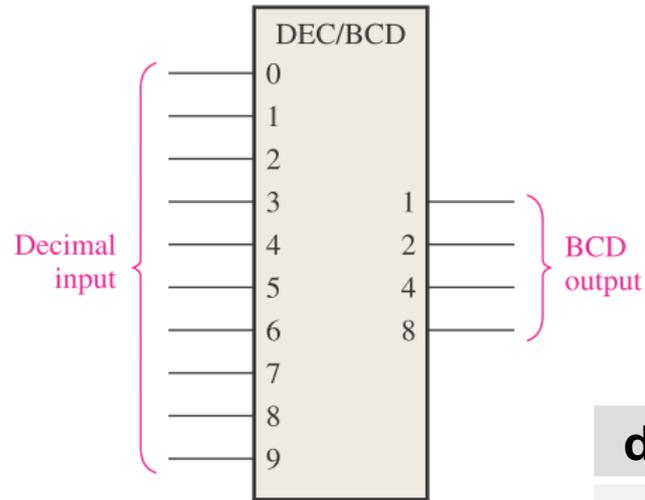
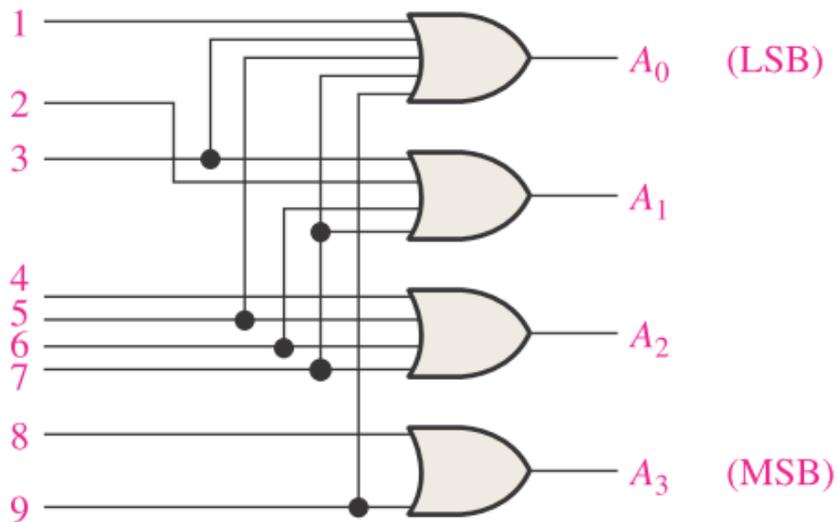


FIGURE 6-36 Logic symbol for a decimal-to-BCD encoder.

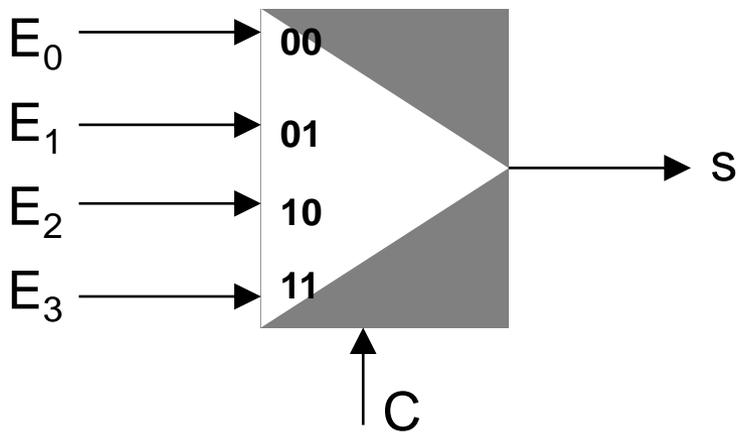


| dec | A_3 | A_2 | A_1 | A_0 |
|-----|-------|-------|-------|-------|
| 0 | | | | |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |

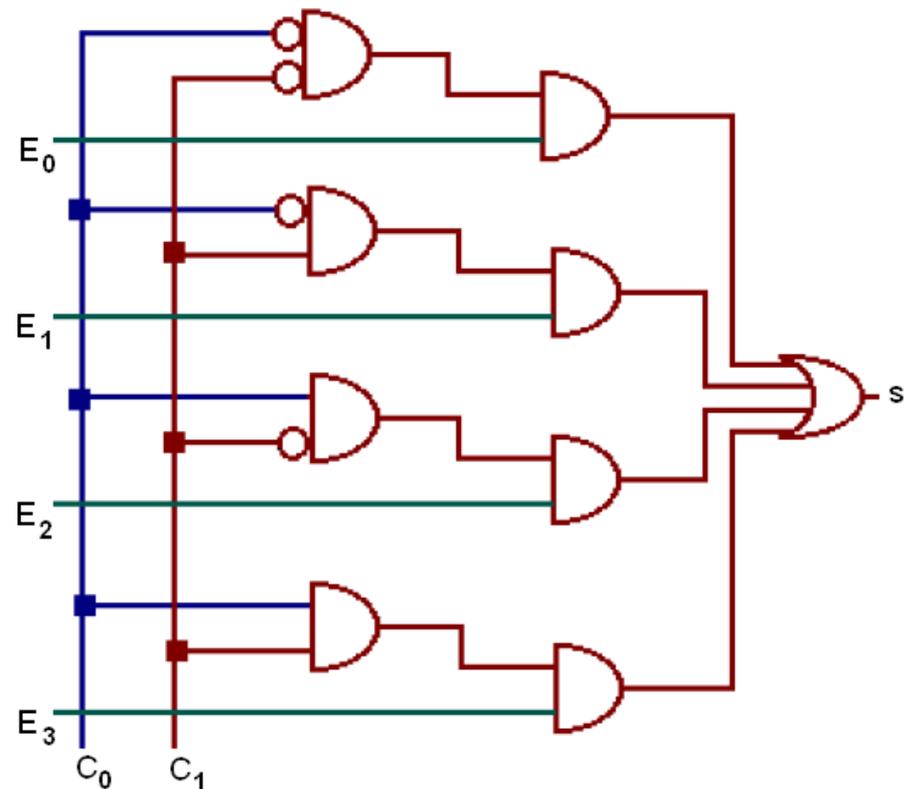
Multiplexador

- É um circuito que permite selecionar mais de uma entrada em uma mesma saída, conforme um sinal de seleção
- Exemplo de um multiplexador 4x1

Representação



Implementação com portas lógicas



Multiplexador 2x1

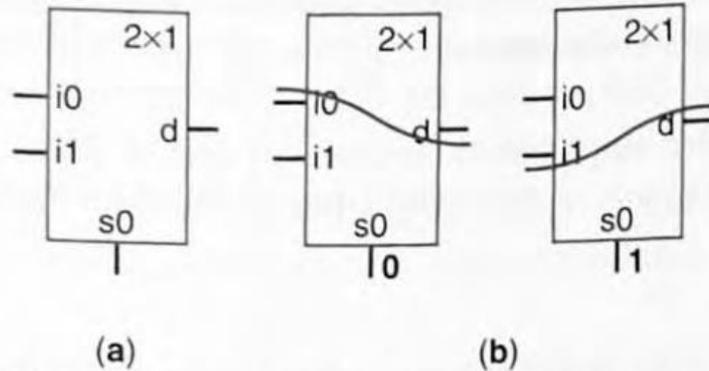
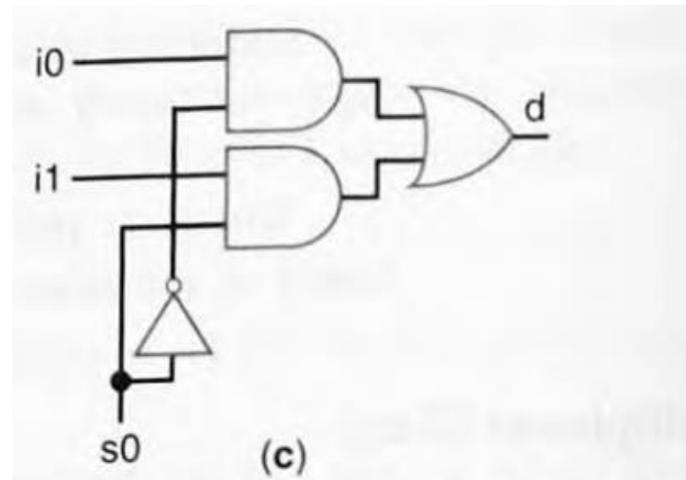
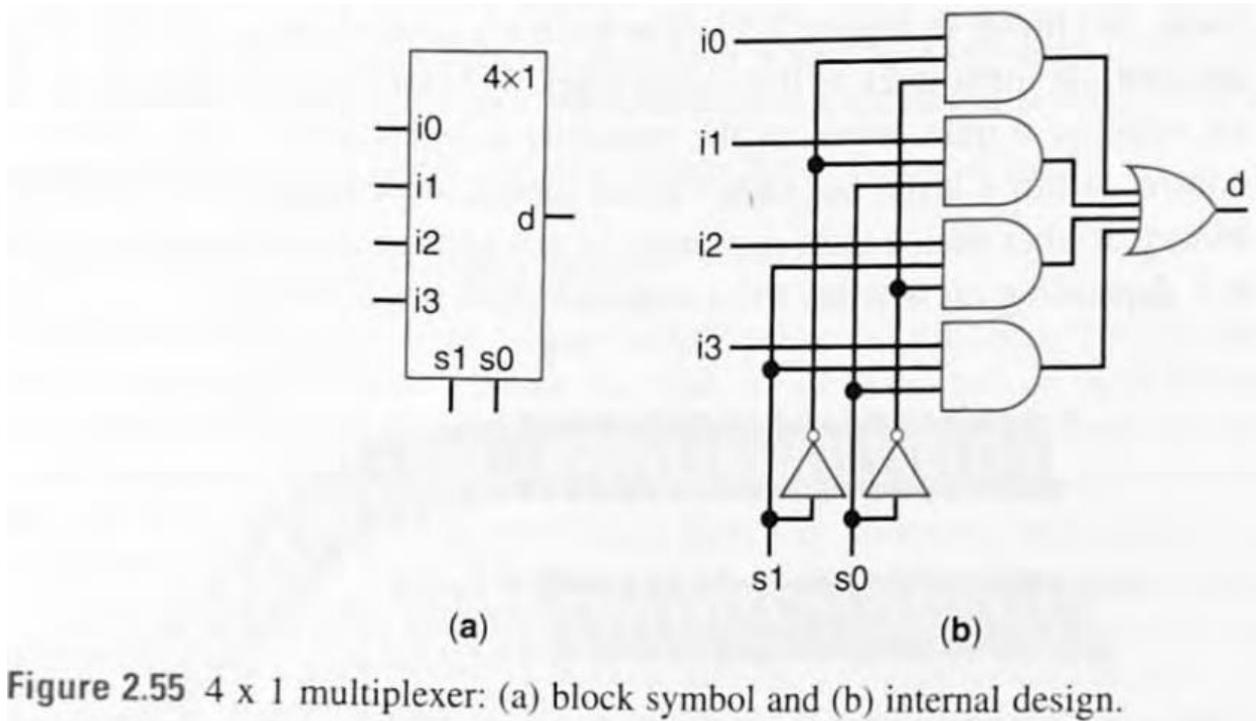


Figure 2.54 2 x 1 multiplexer: (a) block symbol, (b) connection internal design.

| I_0 | I_1 | S_0 | D |
|-------|-------|-------|-----|
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |



Multiplexador 4x1



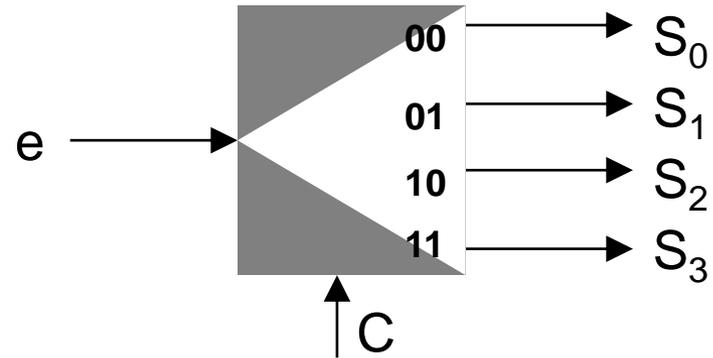
MUX em VHDL

- Usando :
 - when / else
 - with / select
 - process / case

Demultiplexador

– É um circuito que opera de forma inversa ao multiplexador. Ou seja, recebe uma entrada e distribui esta em uma de várias saídas conforme um sinal de seleção

– Exemplo de um multiplexador 1x4



– Igualmente ao multiplexador, um demultiplexador pode ser implementado com diversos comandos. O mais comum, é o que segue:

S(0) <= e when C = "00" else 'Z';

S(1) <= e when C = "01" else 'Z';

S(2) <= e when C = "10" else 'Z';

S(3) <= e when C = "11" else 'Z';

Demultiplexador 2x1 (demux)

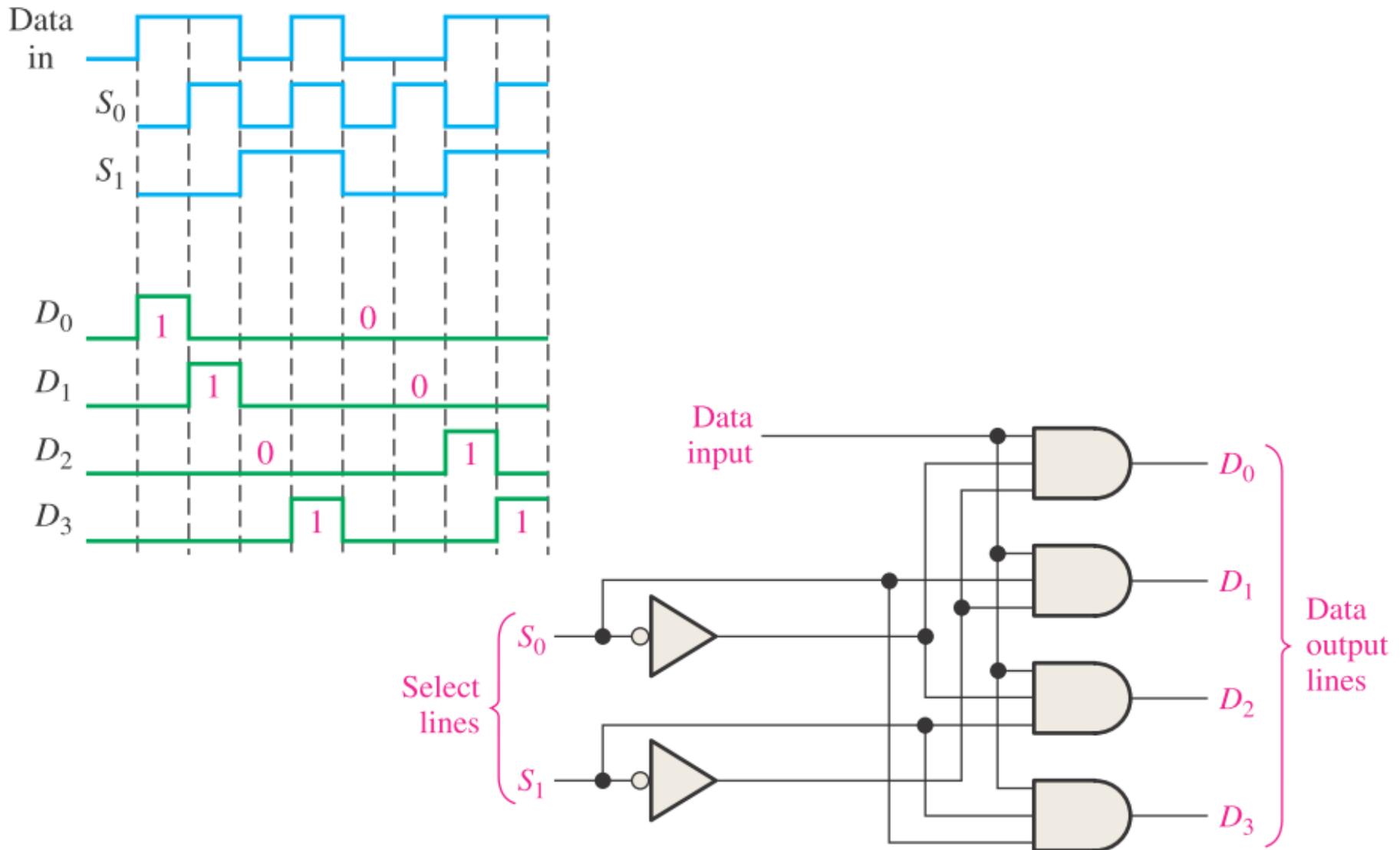


FIGURE 6-52 A 1-line-to-4-line demultiplexer.

Atividade da Próxima Aula

- **Atividade p Próxima Aula**
 - Entregar no início da aula
 - Descrever em VHDL circuitos vistos em aula
 - Escrito à mão !!!! Explicar a resolução do problema !!!
 - Pelo menos 5 módulos de complexidades diferentes
 - Versão estrutural
 - Versão comportamental
 - Mais exemplos VHDL de decodificadores, multiplex no cap 12 do livro Free Range