

Decodificadores e Multiplexadores

Versão 2022

INTRODUÇÃO

Esta experiência visa apresentar ao aluno circuitos decodificadores e multiplexadores que são usados em muitos circuitos combinatórios de circuitos digitais, implementando funções lógicas mais elaboradas como seleção e decisão.

OBJETIVO

O objetivo dessa experiência é estudar e implementar funções lógicas usando circuitos integrados (CIs) mais elaborados como decodificadores e multiplexadores.

Ao final da experiência, espera-se que o aluno tenha aprendido:

- Familiarização com as características dos decodificadores e multiplexadores;
- Implementação de circuitos com funções lógicas mais elaboradas.

1. PARTE EXPERIMENTAL

Os circuitos digitais típicos recebem dados em binário e realizam operações continuamente até se obter um resultado desejado. Dentre as principais operações pode-se citar a codificação, decodificação, multiplexação, demultiplexação, comparação e conversão de código. Para a sua implementação, existem numerosos CIs do tipo MSI (*Medium Scale Integration*).

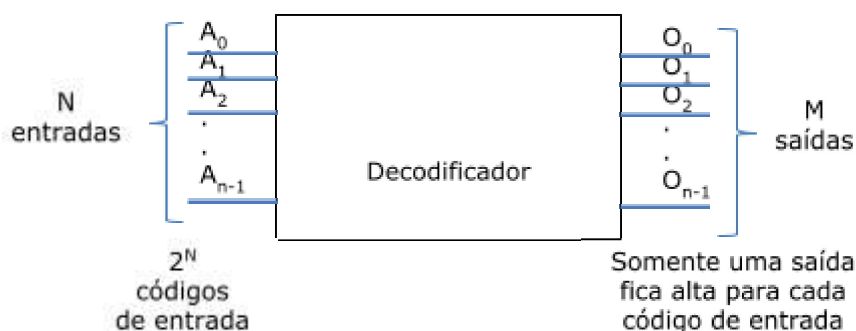
É uma alternativa interessante se comparado aos CIs do tipo SSI (*Small Scale Integration*) como os 7400, 7404, 7402, 7474, etc., pois se pode reduzir o número de componentes resultando-se em um projeto mais simples e econômico e com menor consumo de energia.

Nesta experiência são estudados e implementados circuitos digitais usando decodificadores e multiplexadores.

a) Decodificadores

O decodificador é um circuito lógico que recebe N bits de entrada e ativa 1 única saída entre os M bits de saída disponíveis, correspondendo ao número em binário representado pelos N bits de entrada. Observe que $M = 2^N$. A Figura 1 apresenta o diagrama de blocos de um decodificador de N bits.

Figura 1 – Diagrama de blocos de um decodificador de N bits.



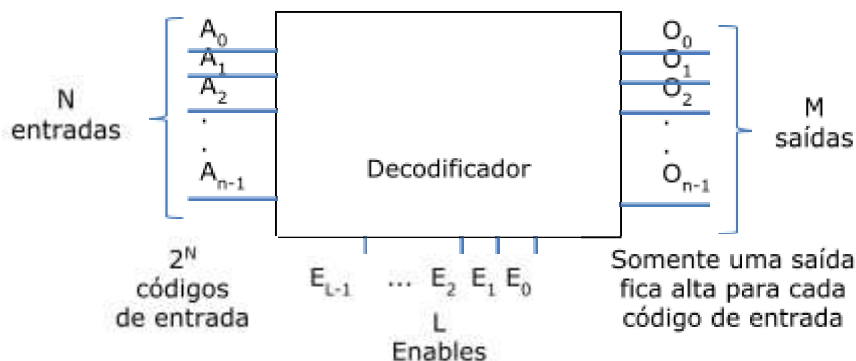
A Figura 2 apresenta a Tabela Verdade do decodificador para 3 bits de entrada e 8 bits de saída ou 3x8 bits. Para cada combinação das entradas A_0 -2, existe somente um bit acionado O_i entre as saídas O_0 -7.

Figura 2 – Tabela Verdade do decodificador 3x8 bits.

Entradas			Saídas							
A_2	A_1	A_0	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_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

Os decodificadores, em geral, possuem entradas que permitem controlar a sua operação, ou seja, L entradas de Enable que permitem fazer os decodificadores funcionarem ou não, conforme apresentadas na Figura 3. O decodificador funciona normalmente caso as L entradas de Enable estejam ativadas.

Na Figura 3 tem-se um $L > 1$ e na Figura 5, o $L = 1$.

Figura 3 – Diagrama de blocos do decodificador com entradas de *Enable*.

A Figura 4 apresenta a Tabela Verdade do decodificador 3x8 bits com *Enable* E de 1 bit.

Os circuitos digitais combinatórios que implementam funções lógicas podem ser obtidos por blocos lógicos, conforme foi visto na Experiência 1. Outra forma de implementar funções lógicas é usando decodificadores. Cada saída "1", nos exemplos até aqui apresentados, pode ser interpretada como uma combinação adequada das entradas e ela ocorre somente em uma situação. Por exemplo, O_0 é "1" somente quando A_2 , A_1 e A_0 são iguais a "0". Neste caso, O_0 implementa a função $\bar{A}_2\bar{A}_1\bar{A}_0$. De forma análoga, pode-se encontrar as funções implementadas pelas demais saídas O_i .

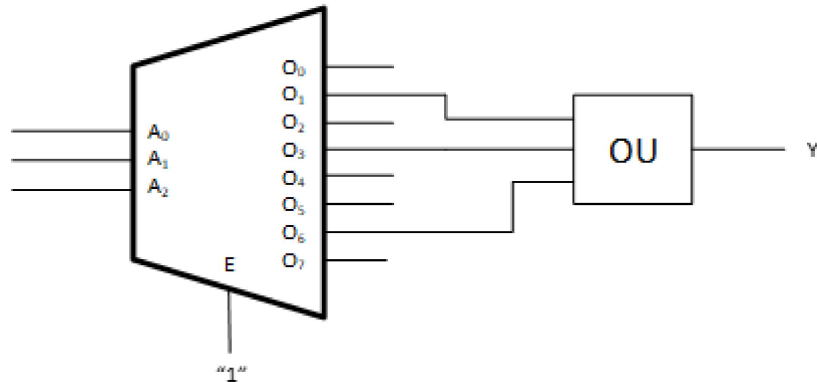
Assim, para implementar uma função qualquer de 3 bits basta integrar todas as parcelas que contribuem para o resultado desejado. Por exemplo, uma saída Y deve ser "1" quando A_2 , A_1 e A_0 forem 001, 011 ou 110, respectivamente.

$$Y = \bar{A}_2\bar{A}_1A_0 + \bar{A}_2A_1A_0 + A_2A_1\bar{A}_0$$

Figura 4 – Tabela Verdade do decodificador 3x8 com *Enable* E de 1 bit.

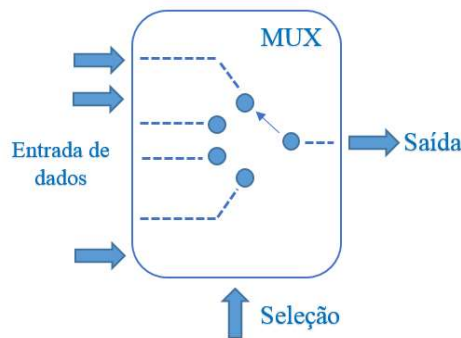
Entrada s				Saídas							
Enable E	A ₂	A ₁	A ₀	O ₇	sO ₆	O ₅	O ₄	O ₃	O ₂	O ₁	O ₀
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0
0	X	X	X	0	0	0	0	0	0	0	0

O diagrama lógico que implementa a saída Y com decodificador 3x8 bits e Enable E é apresentado na Figura 5.

Figura 5 – Diagrama lógico do decodificador 3x8 bits com Enable E com saída Y.

b) Multiplexadores

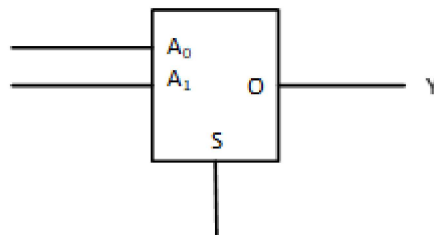
O multiplexador (MUX) ou seletor de dados é um circuito lógico que recebe diversas entradas de dados digitais, seleciona uma delas e, em seguida, transfere-a para a saída. A seleção é feita por uma entrada chamada Seleção. A Figura 6 apresenta o funcionamento do MUX com entradas de dados, de seleção e a saída gerada.

Figura 6 – Diagrama representando o MUX como seletor de dados.

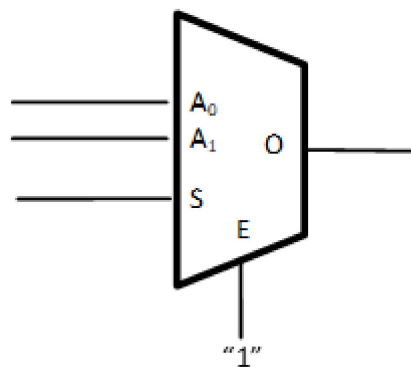
Um MUX simples de duas entradas de dados A_1 e A_0 , uma entrada de seleção S e uma saída Y pode ser expresso da seguinte forma.

$$Y = \bar{S}A_0 + SA_1$$

A Figura 7 apresenta o diagrama lógico de um MUX 2x1 e Seleção S .

Figura 7 – Diagrama lógico de um MUX 2x1 e Seleção S .

Analogamente ao decodificador, um MUX pode ter entradas Enable para controlar a sua operação e permitir o seu funcionamento. A Figura 8 apresenta o diagrama lógico do MUX 2x1 com Enable E .

Figura 8 – Diagrama lógico do MUX 2x1 com *Enable* E .

A Tabela Verdade do MUX 2x1 com *Enable* E pode ser vista na Figura 9.

Figura 9 – Tabela Verdade do MUX 2x1 com Enable E.

Enable	Seleção	Saída
E	S	Y
1	0	A0
1	1	A1
0	X	0

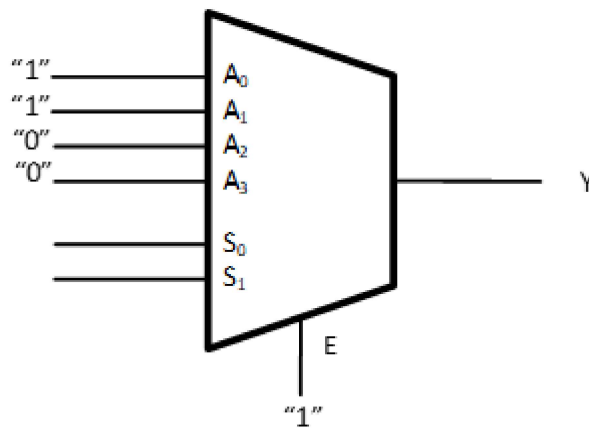
Analogamente ao apresentado com o decodificador, outra forma de implementar funções lógicas é usando multiplexadores. Cada entrada combinada com a Seleção pode expressar uma função lógica. Por exemplo, sendo um MUX de 4 bits de dados de entrada A_i e Seleção de 2 bits S_i , pode-se implementar a função lógica completa a seguir.

$$Y = \bar{S}_1 \bar{S}_0 \bar{A}_0 + \bar{S}_1 \bar{S}_0 A_0 + \bar{S}_1 S_0 \bar{A}_1 + \bar{S}_1 S_0 A_1 + S_1 \bar{S}_0 \bar{A}_2 + S_1 \bar{S}_0 A_2 + S_1 S_0 \bar{A}_3 + S_1 S_0 A_3$$

Para implementar uma função qualquer de 3 bits basta integrar todas as parcelas de acordo com o resultado desejado. Por exemplo, a saída Y deve ser "1" quando S_1 , S_0 e A_i forem 001, 011 ou 110, respectivamente.

$$Y = \bar{S}_1 \bar{S}_0 A_0 + \bar{S}_1 S_0 A_1 + S_1 S_0 \bar{A}_3$$

O diagrama lógico que implementa a saída Y com multiplexador 4x1 com dois bits de Seleção S é apresentado na Figura 10.

Figura 10 – Diagrama lógico do multiplexador 4x1 com dois bits de Seleção S.

Os CIs que são usados no Laboratório Digital são da família TTL 74138/74LS138 (decodificador) e 74151/74LS151 (multiplexador).

1.1. Projeto de dois circuitos lógicos

- Estude o funcionamento do decodificador e do multiplexador apresentado neste enunciado. Procure no *datasheet* os componentes correspondentes.
- Elabore um Diagrama de Blocos (em alto nível) do circuito completo.

- c) Elabore um Diagrama Lógico de um circuito digital usando um **decodificador** que defina uma função Y cuja saída seja "1" quando as entradas forem 0011, 0110, 0111, 1001 e 1011. Insira o Diagrama Lógico no planejamento.
- d) Simule os circuitos na ferramenta Quartus. Insira a Carta de Tempos no planejamento.
- e) Elabore uma Tabela de Testes do circuito. Insira a Tabela de Testes no planejamento.
- f) Projete um circuito digital usando um **multiplexador** na ferramenta Quartus, elabore um Diagrama Lógico que implemente uma função Y cuja saída seja "1" quando as entradas forem 0011, 0110, 0111, 1001 e 1011. Insira o Diagrama Lógico no Planejamento.
- g) Simule os circuitos na ferramenta Quartus. Insira a Carta de Tempos no planejamento.
- h) Elabore uma Tabela de Testes do circuito. Insira a Tabela de Testes no planejamento.

1.2. Implementação dos dois circuitos lógicos

- a) Monte os circuitos no Pannel de Montagens.

Dica: Integre cada componente testado isoladamente aos outros já testados.

- b) Realize os testes dos circuitos usando as Tabelas de Testes planejadas e anote os resultados. Caso um dos testes não seja verificado, proceda a depuração do circuito. Repita os testes após a correção do problema.

1.3. Desafio (Opcional)

O professor irá propor um desafio sobre esta experiência.

2. BIBLIOGRAFIA

- FREGNI, E.; SARAIVA, A. M. **Engenharia do Projeto Lógico Digital: Conceitos e Prática**. Editora Edgard Blücher, 1995.
- TEXAS INSTRUMENTS. **The TTL Logic Data Book**, 1994.
- TOCCI, R. J.; WIDMER, N.S.; MOSS, G.L. **Sistemas Digitais: Princípios e Aplicações**. Prentice-Hall, 11ª ed., 2011.
- WAKERLY, John F. **Digital Design Principles & Practices**. 4th edition, Prentice Hall, 2006.
- ALMEIDA, F. V.; SATO, L.; MIDORIKAWA, E. Tutorial Esquemático Quartus Prime 16.1, versão 1.0, 2017.

3. RECURSOS NECESSÁRIOS

- ✓ 1 Computador pessoal
- ✓ 1 Pannel de montagens de circuitos digitais
- ✓ 1 Fonte de 5 volts
- ✓ 1 Multímetro
- ✓ 1 Ferramenta Quartus Prime 16.1
- ✓ 1 Link de Internet.
- ✓ *Portas básicas TTL*
- ✓ *74138/74LS138*
- ✓ *74151/74LS151*
- ✓ *Cabos, fios e jumpers*

Histórico de Revisões

Prof. Edson T. Midorikawa – versão 2019

Profs. Kechi Hiramã, Jorge Rady de Almeida, Sérgio Roberto de Mello Canovas – versão 2020

Profs. Kechi Hiramã, Jorge Kinoshita, Pedro L C Pizzigatti – versão 2021

Profs. Kechi Hiramã, Madeleine L B Puente de la Vega, – versão 2022