

## Laboratório 04 - Decodificadores e Codificadores

### Objetivos:

1. Experimentar a descrição em VHDL de circuitos na forma comportamental;
2. Reforçar os conceitos de decodificadores.
3. Pôr em prática conceitos aprendidos na disciplina Circuitos Digitais - Teoria.

### Introdução:

Na aula de hoje iremos ver dois conceitos importantes de circuitos digitais: Decodificadores e codificadores.

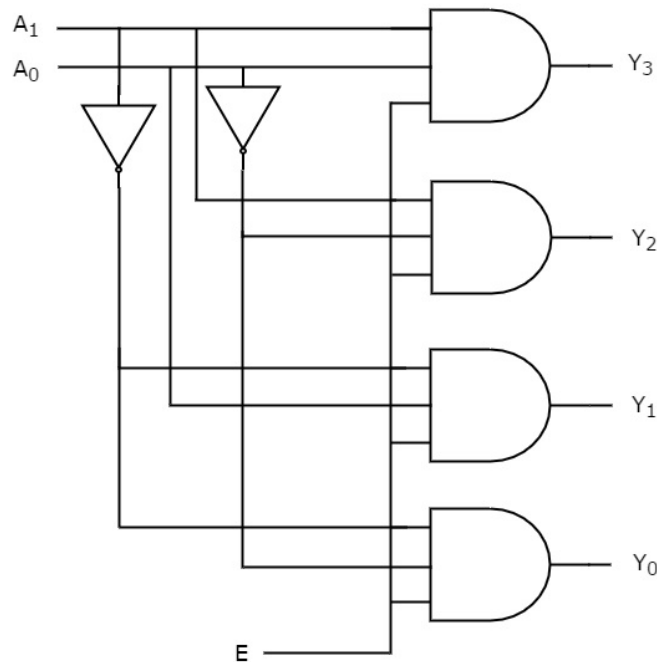
### Decodificadores:

Um decodificador é um bloco construtivo de nível mais elevado, comumente usados em circuitos digitais. Um **decodificador**, como o nome já diz, decodifica um número binário de  $N$  bits de entrada colocando exatamente uma das  $2^n$  saídas do decodificador em 1. Como, por exemplo, o circuito representado pela Tabela 1.

**Tabela 1:** Decodificador de 2 bits.

Entradas		Saídas			
A <sub>1</sub>	A <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

O circuito que descreve um decodificador 2x4 está representado na imagem 1



**Figura 1:** Circuito de um decodificador 2x4.

Note que na Figura 1 existe uma entrada adicional, identificada como **E**. Essa entrada é conhecida como **enable** (traduzida para português como *habilitar*) e é comum em decodificadores. Quando enable é 0, o decodificador coloca todas as saídas em 0, quando o enable é 1, o decodificador funciona conforme mostrado na Tabela 1.

Esse sinal de enable é especialmente útil quando se deseja que nenhuma das saídas sejam ativadas (enable = 0). Por outro lado, se a entrada de habilitação for ativada (enable = 1), pelo menos uma das saídas será 1.

Em VHDL pode-se criar decodificadores utilizando diferentes estratégias, como a descrição de circuitos por portas lógicas ou a descrição comportamental. Os quadros 1 e 2 mostram exemplos destes dois tipos de descrição VHDL. A Figura 2 apresenta formas de onda obtidas com a simulação de ambos os códigos.

**Quadro 1** - Estratégia de descrição via portas lógicas

```
ENTITY Decode IS
PORT(A1,A0,E: IN BIT;
      Y0,Y1,Y2,Y3 : OUT BIT);
END;
ARCHITECTURE behav OF Decode IS
BEGIN
Y0 <= E AND NOT(A1) AND NOT(A0);
```

```
Y1 <= E AND NOT(A1) AND A0;
Y2 <= E AND A1 AND NOT(A0);
Y3 <= E AND A1 AND A0;
END;
```

#### Quadro 2 - Descrição Comportamental do decodificador

```
ENTITY Decode_Comp IS
PORT(A1,A0,E: IN BIT;
      Y0,Y1,Y2,Y3 : OUT BIT);
END;

ARCHITECTURE behav OF Decode_Comp IS
SIGNAL out_aux: BIT_VECTOR(3 DOWNTO 0);
SIGNAL in_aux : BIT_VECTOR(2 DOWNTO 0);
BEGIN
in_aux <= E & A1 & A0 ; -- Usado para concatenar os sinais de entrada
WITH in_aux SELECT
out_aux <= "0001" WHEN "100",
           "0010" WHEN "101",
           "0100" WHEN "110",
           "1000" WHEN "111",
           "0000" WHEN OTHERS;
Y0 <= out_aux(0);
Y1 <= out_aux(1);
Y2 <= out_aux(2);
Y3 <= out_aux(3);
END;
```

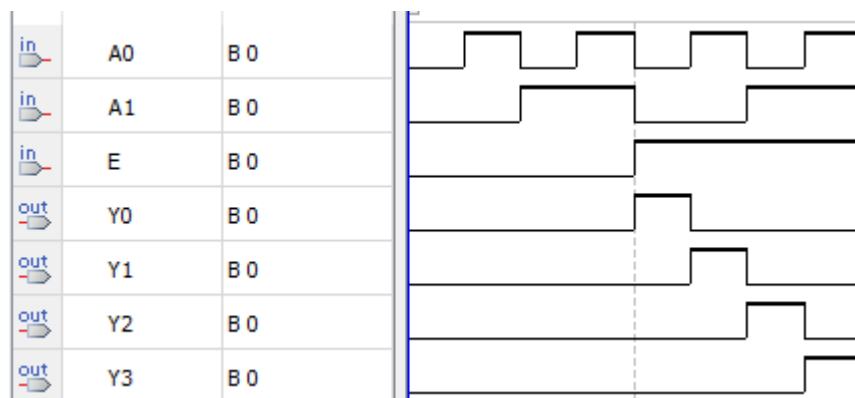


Figura 2 - Simulação do decodificador 2x4.

Note na Figura 2 que quando enable é 0, as saídas do Y0, Y1, Y2 e Y3 são 0. Quando o pino de enable (E) está ativo, as saídas se comportam conforme descrito na Tabela 1.

Circuitos digitais compostos por decodificadores são empregados em diversas aplicações, por exemplo: display de sete segmentos, contadores regressivos, sistemas de irrigação etc.

## **Codificadores:**

Um codificador tem o comportamento oposto ao um decodificador, ou seja, dadas N entradas haverá  $\log_2(N)$  saídas. Na prática, um codificador de quatro entradas, terá duas saídas. A tabela 2 mostra a tabela verdade para um codificador 4x2.

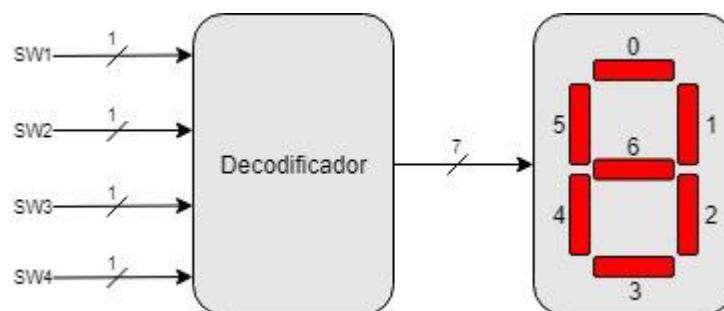
**Tabela 2:** Codificador 4x2.

Entradas				Saídas	
A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

## Atividade:

### Display de 7 segmentos

Muitos eletrodomésticos exibem números para serem lidos por nós, humanos. Exemplos desses eletrodomésticos incluem relógios, fornos micro-ondas, geladeiras etc. Um dispositivo muito popular e simples, capaz de exibir um número de um único dígito é o display de sete segmentos, mostrado na Figura 3.



**Figura 3:** Display de sete segmentos.

Um display de sete segmentos funciona da seguinte forma:

- Quando deseja-se mostrar o número 1, apaga-se os pinos: 0, 3, 4, 5 e 6, deixando ligados apenas os pinos 1 e 2;
- Quando deseja-se mostrar o número 2, apaga-se os pinos: 2 e 5, deixando os restantes ligados.

Pode-se projetar um decodificador BCD (Binary-coded decimal) que é responsável por receber comandos externos (representação binária) e mostrar o número em decimal no display de sete segmentos.

Tarefa:

1. Monte a tabela verdade para um decodificador de 4 para 7 bits. Utilize nível lógico '1' para indicar que o pino está apagado e nível lógico '0' para indicar que o pino está aceso.

Exemplo:

Entradas				Saídas							
A4	A3	A2	A1	S6	S5	S4	S3	S2	S1	S0	Valor Visualizado
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0	1	1
0	0	1	0								2
0	0	1	1								3
0	1	0	0								4
0	1	0	1								5
0	1	1	0								6
0	1	1	1								7
1	0	0	0								8
1	0	0	1								9

- Projete este decodificador BCD em VHDL utilizando a descrição por portas lógicas.
- Projete este decodificador BCD em VHDL utilizando a descrição comportamental.
- Considere, em seu projeto, que a saída deve ter todos os pinos apagados para os casos em que as entradas não correspondam a uma saída que mostre um algarismo entre 0 e 9.
- Elabore e entregue um relatório contendo a descrição, códigos e resultados de simulação deste experimento.