



# Instituto Tecnológico de Aeronáutica - ITA

## Divisão de Engenharia Eletrônica

Departamento de Eletrônica Aplicada

### Laboratório de EEA-21

#### 2ª Experiência

### Análise e síntese de funções combinacionais de múltiplas saídas.

## 1. Objetivos

- a. Familiarização e utilização das funções básicas: Codificador, Decodificador e Demultiplexador;
- b. Minimização de funções combinacionais de múltiplas saídas; e
- c. Compreensão das funções: Multiplexador e Codificador de prioridade.

## 2. Instruções gerais

Para cada uma das tarefas propostas, observe o que se pede em negrito. Quando a solicitação é da forma:

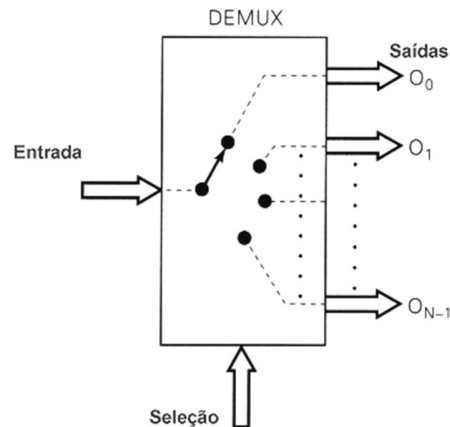
- “**Projete**”, espera-se que os alunos projetem os circuitos seguindo algum tipo de especificação **dada**. Assim, os cálculos combinacionais, os procedimentos de minimização (ou outros mecanismos utilizados para o projeto) e o diagrama esquemático final do circuito devem ser mostrados (constar do relatório).
- “**Simule**”, espera-se que seja utilizado um software de captura esquemática para a obtenção dos resultados. Assim, o diagrama esquemático (no caso de captura esquemática) deve ser apresentado, bem como o diagrama de temporização contendo as entradas e as saídas.
- “**Analise**”, espera-se que sejam obtidas as expressões lógicas, tabelas verdade, a partir de um diagrama esquemático, diagrama de temporização ou outra informação sobre o circuito lógico. Dessa forma, os procedimentos de análise devem ser mostrados no relatório.

## 3. Informações úteis

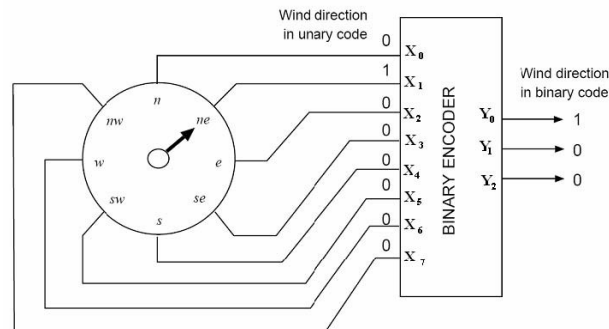
### 3.1 Demultiplexador

Este bloco (DEMUX) opera de forma inversa ao MUX, visto no 1º laboratório. Possui uma entrada de 1 bit e  $n$  saídas de 1 bit. O sinal de entrada é apresentado em uma das  $n$  saídas a partir da seleção feita nos bits de seleção.

O DEMUX apresenta ainda uma entrada de habilitação,  $En$ : se  $En = 1$ , a operação é conforme descrito; se  $En = 0$ , todas as saídas apresentam valor lógico 0.



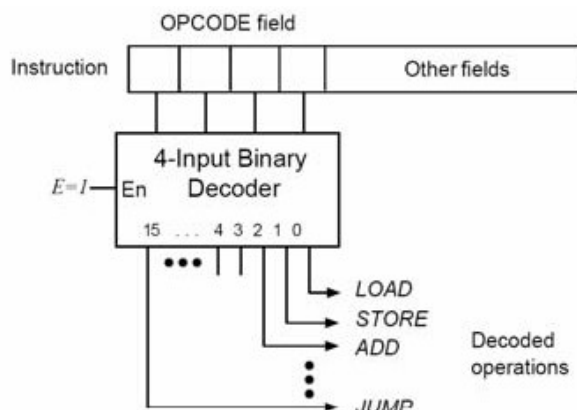
**3.2 Codificador** - Exemplo de utilização de Codificador binário: Codificador da direção do vento. Neste ponto deve-se ressaltar que por convenção o bit com índice zero (ex.  $x_0$ ) representa o bit menos significativo ou bit LSB (*“Least Significant Bit”*). Já o bit com maior índice (ex.  $x_m$ ) representa o bit mais significativo ou bit MSB (*“Most Significant Bit”*).



O codificador binário (*encoder*) apresenta como entrada  $2^n$  bits, e saída de  $n$  bits. No exemplo apresentado tem-se 8 bits de entrada e 3 bits de saída. Os bits da saída representam o índice da entrada que apresenta valor lógico 1. É necessário garantir que apenas uma das entradas apresente valor lógico 1 em cada instante. A seguir apresenta-se a tabela verdade para este bloco:

$x_7$	$x_6$	$x_5$	$x_4$	$x_3$	$x_2$	$x_1$	$x_0$	$y_2$	$y_1$	$y_0$
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

### 3.3 Decodificador - Exemplo de utilização de Decodificadores: Decodificador de instruções.



O decodificador (*decoder*) possui operação inversa ao codificador. Para  $n$  bits de entrada, ele apresenta  $2^n$  bits de saída. No exemplo acima temos um decodificador 4x16 (4 entradas e 16 saídas). O número binário formado pelos bits de entrada indica qual o índice daquela saída que apresentará valor lógico 1. Todas as outras saídas apresentarão valor lógico 0. O decodificador apresenta ainda um sinal de ativação ( $En - Enable$ ): se  $En = 1$  o decodificador está ativado e opera conforme descrito; se  $En = 0$ , todas as  $2^n$  saídas apresentam valor lógico 0.

Para um decodificador 2x4 apresenta-se a tabela verdade:

$x_1$	$x_0$	En	$y_3$	$y_2$	$y_1$	$y_0$
0	0	1	0	0	0	1
0	1	1	0	0	1	0
1	0	1	0	1	0	0
1	1	1	1	0	0	0

**3.4 Codificador de Prioridade - Codificador de Prioridade (4x2):** É um dispositivo combinacional com 4 entradas ( $x_3, x_2, x_1, x_0$ ) e três saídas ( $A, y_1, y_0$ ). A saída  $A$  indica que uma das 4 entradas foi acionada (qualquer uma delas). As saídas  $y_1$  e  $y_0$ , na forma de número binário, indicam qual foi a saída de maior prioridade acionada. Este circuito atribui à entrada  $x_0$  a menor prioridade e à entrada  $x_3$  a maior prioridade.

Tabela verdade de um Codificador de Prioridade 4x2.

$x_3$	$x_2$	$x_1$	$x_0$	$y_1$	$y_0$	$A$
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	-	0	1	1
0	1	-	-	1	0	1
1	-	-	-	1	1	1

Equações Lógicas:

$$y_1 = x_2 + x_3$$

$$y_0 = \overline{x_2}x_1 + x_3$$

$$A = x_0 + x_1 + x_2 + x_3$$

#### 4. Tarefas:

##### 4.1 Demultiplexador - Projete e simule a função demultiplexadora (DEMUX) $1 \times 4$ , utilizando apenas portas NOR.

Obtenha:

- A tabela de verdade completa e as expressões booleanas simplificadas das quatro saídas;
- O diagrama esquemático (mostre os passos seguidos para obtenção do diagrama esquemático a partir das expressões booleanas); e
- O diagrama de temporização obtido através da simulação funcional do circuito.

##### 4.2 Codificador - Projete e simule a função conversora de códigos *de 3 bits (código de Gray $\rightarrow$ binário puro)*. Sugestão: utilize portas XOR e XNOR para minimizar a implementação.

O código Gray é uma codificação em que há variação de apenas um bit de um número para um adjacente, diferente do que ocorre na codificação binária pura, onde há variação de mais bits (por exemplo, 3(011)  $\rightarrow$  4(100) tem-se a variação dos 3 bits. A tabela a seguir apresenta a correspondência para números de 3 bits. Para construir a tabela basta observar que o bit mais significativo é idêntico ao da codificação binária pura, e os demais bits são “espelhados” (observe as cores na tabela abaixo).

Decimal	Binário Puro	Código Gray
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Obtenha:

- A tabela verdade completa;
- O diagrama esquemático; e
- O diagrama de temporização obtido através da simulação funcional do circuito.

### 4.3 Decodificador - Projete e simule a função decodificadora (DECOD) 3x8 (entrada binário puro).

Obtenha:

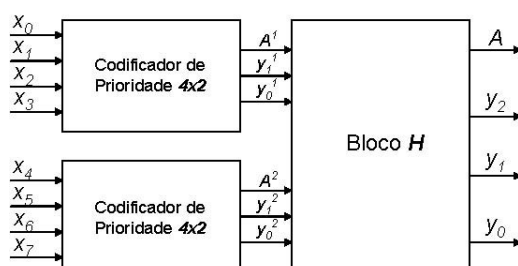
- A tabela de verdade completa e as expressões booleanas simplificadas das saídas;
- O diagrama esquemático; e
- O diagrama de temporização obtido através da simulação funcional do circuito.

### 4.4 Codificador de Prioridade (1) - Projete e simule a função codificadora (COD) 4x2 (saída binário puro) de prioridade, minimizada na forma de produto de somas.

Obtenha:

- A tabela verdade completa e as expressões booleanas simplificadas das saídas;
- O diagrama esquemático; e
- O diagrama de temporização obtido através da simulação funcional do circuito.

### 4.5 Codificador de Prioridade (2) - Projete e simule um codificador de prioridade 8x3 utilizando como célula básica o codificador de prioridade 4x2 projetado na questão 4.3. Decomponha o projeto da seguinte forma:



O codificador de prioridade 8x3 atende as especificações abaixo.

Inputs:  $\underline{x} = (x_{2^n-1}, \dots, x_0)$ ,  $x_i \in \{0, 1\}$

Outputs:  $\underline{y} = (y_{n-1}, \dots, y_0)$ ,  $y_j \in \{0, 1\}$

Function:

$$y = \begin{cases} i & \text{if } (x_i = 1) \text{ and } (x_k = 0, k > i) \\ 0 & \text{otherwise} \end{cases}$$

$$A = \begin{cases} 1 & \text{if (some } x_i = 1) \\ 0 & \text{otherwise} \end{cases}$$

$$y = \sum_{j=0}^{n-1} y_j 2^j$$

and

$$i, k \in \{0, 1, \dots, 2^n - 1\}$$

onde  $n = 3$ .

$x_7$	$x_6$	$x_5$	$x_4$	$x_3$	$x_2$	$x_1$	$x_0$	$y_2$	$y_1$	$y_0$	$A$
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	-	0	0	1	1
0	0	0	0	0	1	-	-	0	1	0	1
0	0	0	0	1	-	-	-	0	1	1	1
0	0	0	1	-	-	-	-	1	0	0	1
0	0	1	-	-	-	-	-	1	0	1	1
0	1	-	-	-	-	-	-	1	1	0	1
1	-	-	-	-	-	-	-	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
-	-	-	-	-	-	-	-	0	0	0	0

Já a tabela verdade de um Codificador de Prioridade **4x2** é dada por:

$x_3$	$x_2$	$x_1$	$x_0$	$y_1$	$y_0$	$A$
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	-	0	1	1
0	1	-	-	1	0	1
1	-	-	-	1	1	1

Obtenha:

- As expressões booleanas e o diagrama esquemático do Bloco **H**;
- O diagrama de temporização do Bloco H, obtido pela simulação funcional; e
- O diagrama de temporização do dispositivo completo, obtido pela simulação funcional.