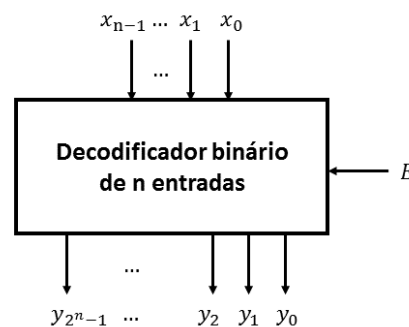


Notas da Aula 13 – Decodificadores e Codificadores

- Sistemas digitais complexos são implementados a partir de redes modulares. Essas redes correspondem a funções que foram identificadas como úteis para uma grande variedade de aplicações e, por isso mesmo, foram colocadas à disposição como componentes de biblioteca de dispositivos.
- Os módulos combinacionais mais usados são os decodificadores, os codificadores, os multiplexadores, os demultiplexadores e os deslocadores. Adiante, para os **módulos decodificador binário e codificador binário**, serão fornecidas a descrição de alto nível, a descrição binária, as expressões de chaveamento, a representação das portas lógicas e algumas formas de aplicações.

- Decodificador binário: é um sistema combinacional que apresenta n entradas binárias $\underline{x} = (x_{n-1}, \dots, x_0)$ e 2^n saídas binárias $\underline{y} = (y_{2^n-1}, \dots, y_0)$. O vetor de entradas \underline{x} admite os números inteiros de 0 a $2^n - 1$ na representação de base dois e, em qualquer instante, o vetor de saídas \underline{y} admite no máximo que uma delas, y_i por exemplo, seja igual a 1 (todas as outras são iguais a 0), sendo que $y_i = 1$ indica que \underline{x} representa o número inteiro i . Uma entrada adicional E (*Enable*, do inglês habilita) é usada para facilitar a implementação de redes decodificadoras (isso ficará claro logo); quando $E = 0$, todas as saídas do decodificador se tornam zero. A figura abaixo ilustra esse módulo.



- Uma descrição de alto nível de um decodificador binário de n entradas é:

Entradas: $\underline{x} = (x_{n-1}, \dots, x_0), x_j \in \{0, 1\}$

$E \in \{0, 1\}$

Saídas: $\underline{y} = (y_{2^n-1}, \dots, y_0), y_i \in \{0, 1\}$

Função: $y_i = \begin{cases} 1, & \text{se } x = i \text{ e } E = 1 \\ 0, & \text{caso contrário} \end{cases}$

em que $x = \sum_{j=0}^{n-1} x_j 2^j$ e $i = 0, \dots, 2^n - 1$

- Uma descrição binária de um decodificador binário de n entradas é:

Entradas: $\underline{x} = (x_{n-1}, \dots, x_0), x_j \in \{0, 1\}$

$E \in \{0, 1\}$

Saídas: $\underline{y} = (y_{2^n-1}, \dots, y_0), y_i \in \{0, 1\}$

Função: $y_i = E \cdot m_i(\underline{x}), i = 0, \dots, 2^n - 1$

em que $m_i(\underline{x})$ é o i -ésimo mintermo do vetor de entradas.

- Decodificador binário de duas entradas:

Tabela verdade

E	x_1	x_0	x	y_3	y_2	y_1	y_0
1	0	0	0	0	0	0	1
1	0	1	1	0	0	1	0
1	1	0	2	0	1	0	0
1	1	1	3	1	0	0	0
0	-	-	-	0	0	0	0

Expressões de chaveamento

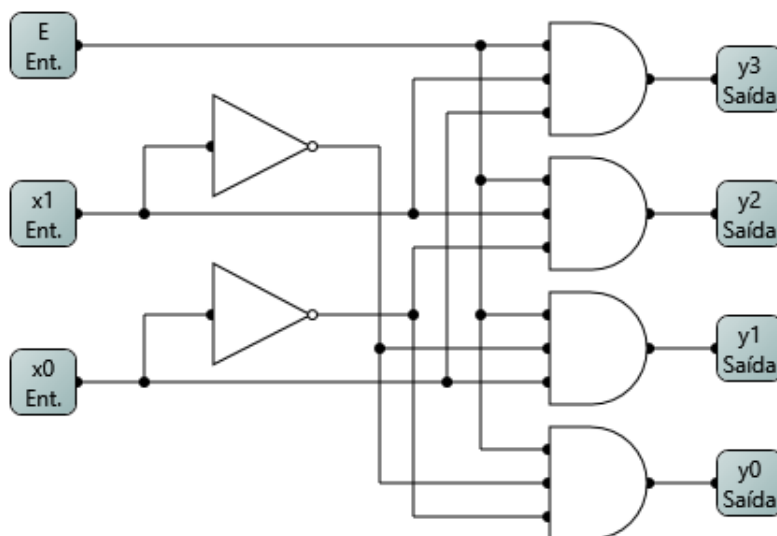
$$y_0 = E \cdot m_0 = E \cdot x'_1 x'_0$$

$$y_1 = E \cdot m_1 = E \cdot x'_1 x_0$$

$$y_2 = E \cdot m_2 = E \cdot x_1 x'_0$$

$$y_3 = E \cdot m_3 = E \cdot x_1 x_0$$

Rede de portas lógicas



- Decodificador binário de três entradas:

Tabela verdade

E	x_2	x_1	x_0	x	y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0
1	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	1	0	0	0	0	0	0	1	0
1	0	1	0	2	0	0	0	0	0	1	0	0
1	0	1	1	3	0	0	0	0	1	0	0	0
1	1	0	0	4	0	0	0	1	0	0	0	0
1	1	0	1	5	0	0	1	0	0	0	0	0
1	1	1	0	6	0	1	0	0	0	0	0	0
1	1	1	1	7	1	0	0	0	0	0	0	0
0	-	-	-	-	0	0	0	0	0	0	0	0

Expressões de chaveamento

$$y_0 = E \cdot m_0 = E \cdot x'_2 x'_1 x'_0$$

$$y_1 = E \cdot m_1 = E \cdot x'_2 x'_1 x_0$$

$$y_2 = E \cdot m_2 = E \cdot x'_2 x_1 x'_0$$

$$y_3 = E \cdot m_3 = E \cdot x'_2 x_1 x_0$$

$$y_4 = E \cdot m_4 = E \cdot x_2 x'_1 x'_0$$

$$y_5 = E \cdot m_5 = E \cdot x_2 x'_1 x_0$$

$$y_6 = E \cdot m_6 = E \cdot x_2 x_1 x'_0$$

$$y_7 = E \cdot m_7 = E \cdot x_2 x_1 x_0$$

Rede de portas lógicas (na próxima página)

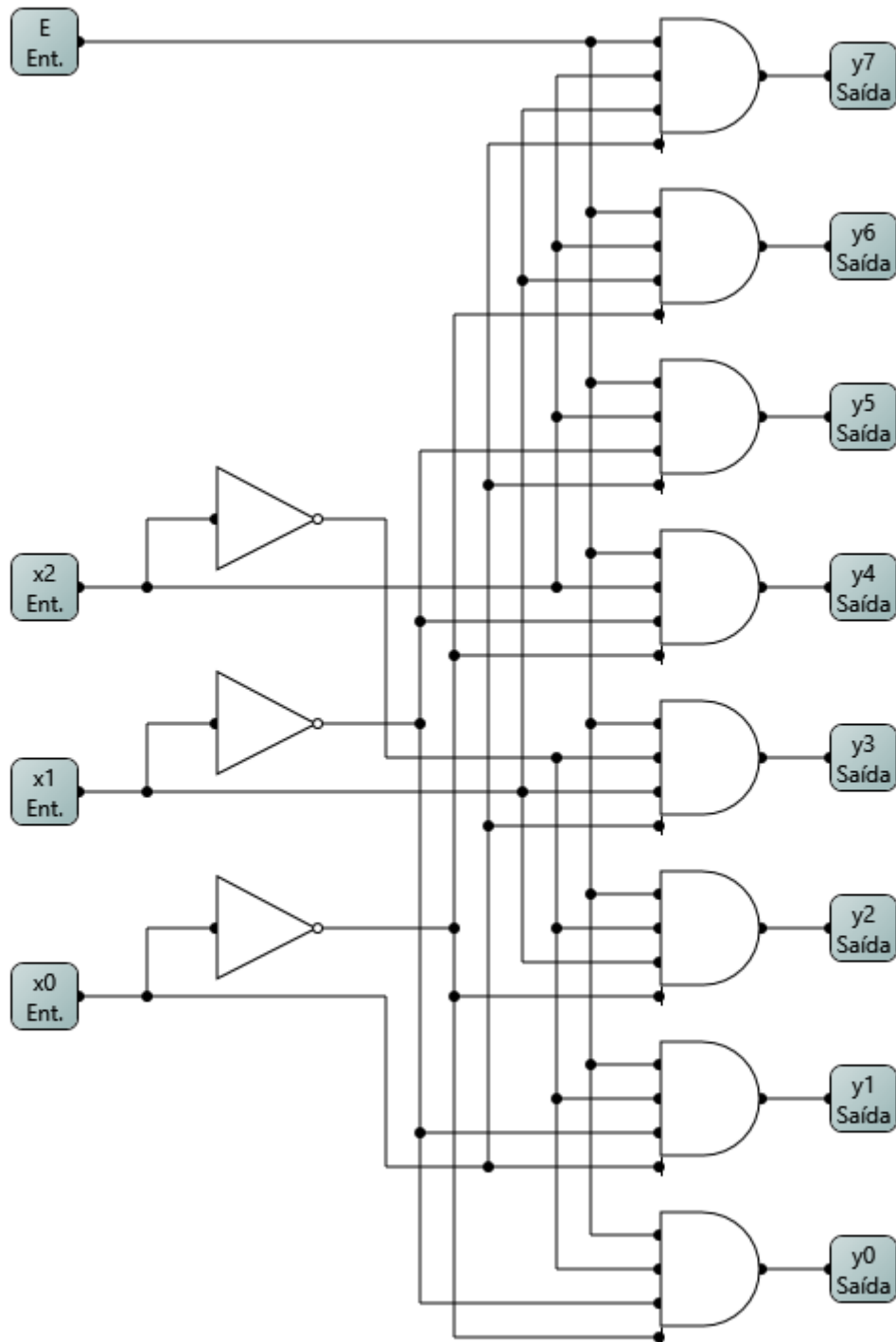
- Notação abreviada: um módulo decodificador é denotado por:

$$\underline{y} = DEC(\underline{x}, E)$$

Por exemplo, para um decodificador de três entradas, tem-se que:

$$\underline{y} = DEC((1, 1, 0), 1) = (0, 1, 0, 0, 0, 0, 0, 0)$$

- Aplicações: um conjunto de 2^n elementos pode ser codificado com n variáveis binárias. Dessa forma, um **decodificador de n bits** pode ser usado para identificar qual elemento do conjunto foi codificado. Exemplos típicos são: decodificar o código de operação de uma instrução de máquina ou decodificar o endereço referenciado quando se acessa uma posição de memória.



- Observação importante: evidentemente que existem decodificadores para outras representações de entrada. Por exemplo, os decodificadores decimais têm dez saídas para códigos como o BCD e Excesso-3; a definição e implementação deles são semelhantes às do decodificador binário, de modo que esse assunto será cobrado como forma de exercício. Implemente!

- Decodificador binário e porta OR como conjunto universal: um decodificador binário de n entradas e uma porta OR podem expressar qualquer função de chaveamento de n variáveis, de forma que esses dois componentes correspondem a um conjunto universal de módulos combinacionais. Por exemplo, considere o sistema combinacional adiante.

Tabela verdade

x_2	x_1	x_0	x	z_2	z_1	z_0
0	0	0	0	0	1	0
0	0	1	1	1	0	0
0	1	0	2	0	0	1
0	1	1	3	0	1	0
1	0	0	4	0	0	1
1	0	1	5	1	0	1
1	1	0	6	0	0	0
1	1	1	7	1	0	0

Esse sistema pode ser implementado com um decodificador binário de três entradas e três portas OR, como mostra as expressões de chaveamento e a figura abaixo.

$$(y_7, \dots, y_0) = DEC((x_2, x_1, x_0), 1)$$

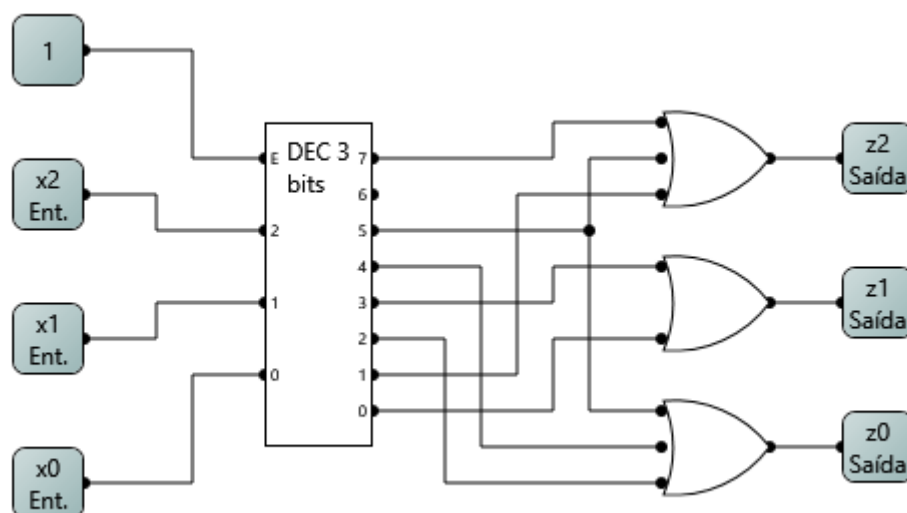
Expressões de chaveamento

$$z_2(x_2, x_1, x_0) = y_1 + y_5 + y_7$$

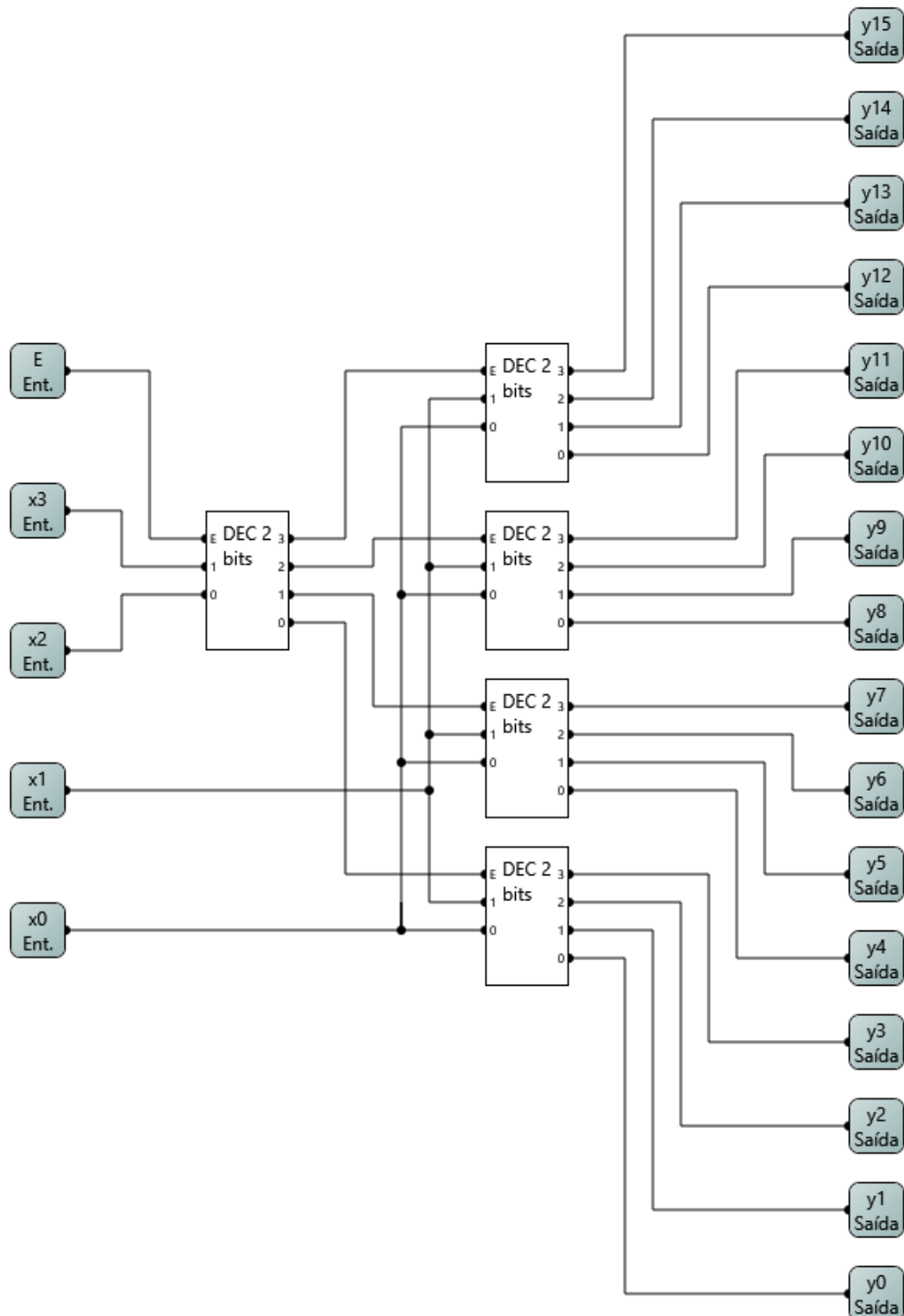
$$z_1(x_2, x_1, x_0) = y_0 + y_3$$

$$z_0(x_2, x_1, x_0) = y_2 + y_4 + y_6$$

Rede de portas lógicas



- Redes decodificadoras: devido às limitações quanto ao número máximo de conexões e de portas em uma implementação, geralmente os decodificadores são disponibilizados em módulos-padrão com apenas quatro entradas, o que os limita a dezesseis saídas. Quando decodificadores maiores forem necessários, uma rede de decodificadores é construída a partir dos módulos-padrão, como mostra figura abaixo.

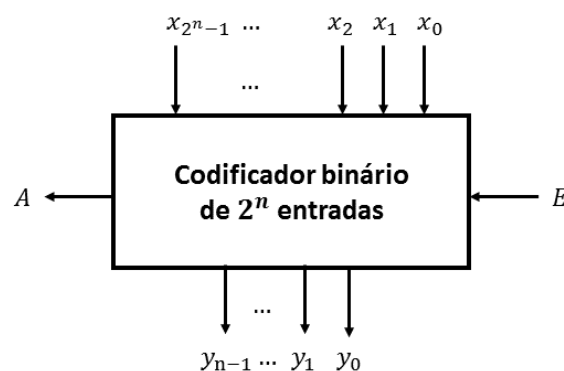


- Decodificação em árvore: como implementar um decodificador binário de quatro entradas (e, portanto, dezesseis saídas) com o uso de decodificadores-padrão de duas entradas? Primeiramente, divide-se o vetor de entradas de quatro bits em dois vetores de dois bits, ou seja, $\underline{x} = (\underline{x}_{\text{esquerdo}}, \underline{x}_{\text{direito}})$, no que $\underline{x}_{\text{esquerdo}} = (x_3, x_2)$ e $\underline{x}_{\text{direito}} = (x_1, x_0)$. Como mostra a figura acima, a partir dessa decomposição, é possível construir uma rede em árvore de dois níveis que apresenta um decodificador no primeiro nível, que decodifica $\underline{x}_{\text{esquerdo}}$, e quatro decodificadores no segundo nível, que decodificam $\underline{x}_{\text{direito}}$. As dezesseis saídas são divididas em quatro grupos de quatro saídas, em que cada grupo é formado por um módulo decodificador-padrão.

- Observações importantes: nota-se que $\underline{x}_{\text{esquerdo}}$ ativa apenas um dos decodificadores do segundo nível e que $\underline{x}_{\text{direito}}$ produz a saída correspondente ao decodificador ativado. Além disso, nota-se o papel fundamental do *enable* para a operação da rede. Por fim, convém lembrar que esse esquema de decodificação em árvore pode ser facilmente generalizado para o caso de múltiplos níveis, o que forneceria cada vez mais saídas.

- Para os curiosos: há outras estratégias de criar redes decodificadoras: a **decodificação coincidente**, por exemplo. Pesquise sobre o assunto!

- Codificador binário: é um sistema combinacional que apresenta 2^n entradas binárias $\underline{x} = (x_{2^n-1}, \dots, x_0)$ e n saídas binárias $\underline{y} = (y_{n-1}, \dots, y_0)$. Esse módulo realiza a função inversa de um decodificador, ou seja, converte de um código 1 entre 2^n para um código binário. Nesse caso, em qualquer instante, no máximo uma das entradas pode ser igual a um e o vetor de saída representa o índice de entrada com o valor 1. Uma entrada adicional E (*Enable*, do inglês habilita) e uma saída adicional A (*Active*, do inglês ativo) são usadas para facilitar a implementação de redes codificadoras; quando $E = 0$, todas as saídas do codificador se tornam zero e, quando $A = 1$, o codificador está ativo (uma das entradas está com valor 1). A figura abaixo ilustra esse módulo.



- Uma descrição de alto nível de um codificador binário de 2^n entradas é:

Entradas: $\underline{x} = (x_{2^n-1}, \dots, x_0)$, $x_i \in \{0, 1\}$, com no máximo um $x_i = 1$
 $E \in \{0, 1\}$

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

Função: $y = \begin{cases} i, & \text{se } x_i = 1 \text{ e } E = 1 \\ 0, & \text{caso contrário} \end{cases}$
 $A = \begin{cases} 1, & \text{se algum } x_i = 1 \text{ e } E = 1 \\ 0, & \text{caso contrário} \end{cases}$
 em que $y = \sum_{j=0}^{n-1} y_j 2^j$ e $i = 0, \dots, 2^n - 1$

- Uma descrição binária de um codificador binário de 2^n entradas é:

Entradas: $\underline{x} = (x_{2^n-1}, \dots, x_0)$, $x_i \in \{0, 1\}$, com no máximo um $x_i = 1$
 $E \in \{0, 1\}$

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

Função: $y_j = E \cdot \sum(x_k)$
 $A = E \cdot \sum(x_i)$
 em que x_k é incluído na expressão para y_j se o j -ésimo bit da representação binária de k for 1.

- Codificador binário de quatro entradas:

Tabela verdade

E	x_3	x_2	x_1	x_0	y	y_1	y_0	A
1	0	0	0	1	0	0	0	1
1	0	0	1	0	1	0	1	1
1	0	1	0	0	2	1	0	1
1	1	0	0	0	3	1	1	1
1	0	0	0	0	0	0	0	0
0	-	-	-	-	0	0	0	0

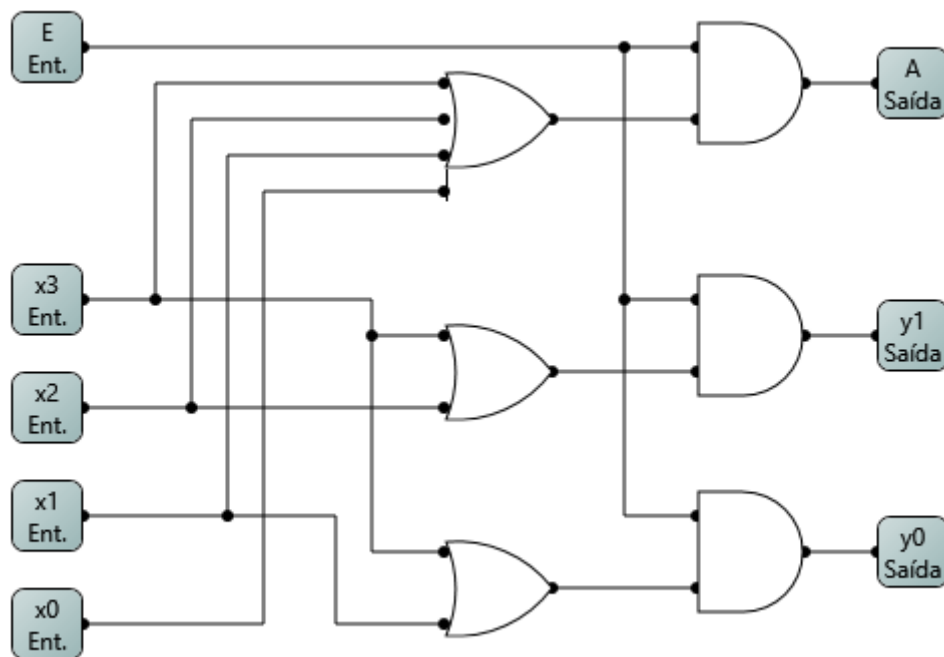
Expressões de chaveamento

$$y_0 = E \cdot (x_1 + x_3)$$

$$y_1 = E \cdot (x_2 + x_3)$$

$$A = E \cdot (x_0 + x_1 + x_2 + x_3)$$

Rede de portas lógicas



- Codificador binário de oito entradas:

Tabela verdade

E	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	y	y_2	y_1	y_0	A
1	0	0	0	0	0	0	0	1	0	0	0	0	1
1	0	0	0	0	0	0	1	0	1	0	0	1	1
1	0	0	0	0	0	1	0	0	2	0	1	0	1
1	0	0	0	0	1	0	0	0	3	0	1	1	1
1	0	0	0	1	0	0	0	0	4	1	0	0	1
1	0	0	1	0	0	0	0	0	5	1	0	1	1
1	0	1	0	0	0	0	0	0	6	1	1	0	1
1	1	0	0	0	0	0	0	0	7	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-	-	-	-	-	-	-	-	0	0	0	0	0

Expressões de chaveamento

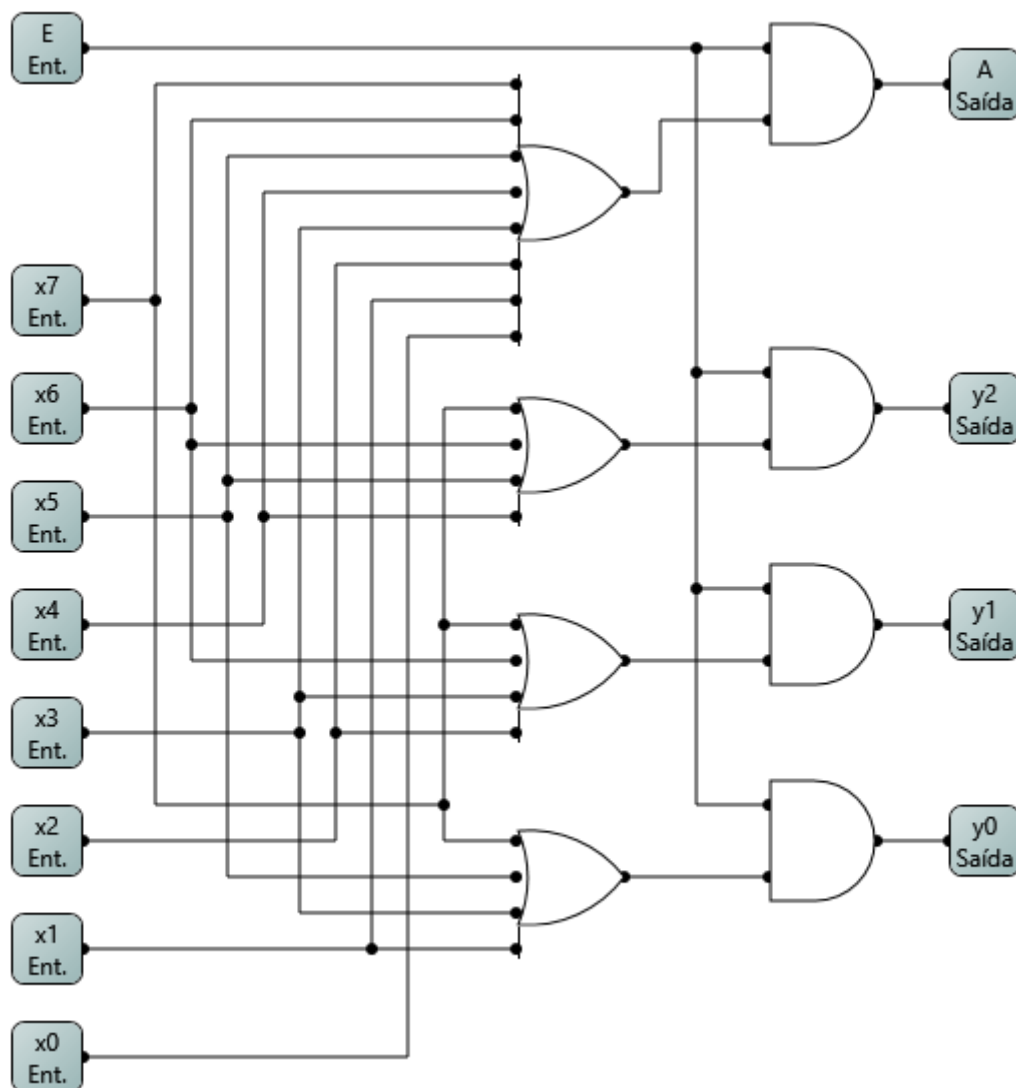
$$y_0 = E \cdot (x_1 + x_3 + x_5 + x_7)$$

$$y_1 = E \cdot (x_2 + x_3 + x_6 + x_7)$$

$$y_2 = E \cdot (x_4 + x_5 + x_6 + x_7)$$

$$A = E \cdot (x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7)$$

Rede de portas lógicas



- Aplicações: toda vez que a ocorrência de um dentre diversos eventos disjuntos precisa ser representada por um número inteiro, um codificador binário pode ser usado. Por exemplo, um dispositivo que determina a direção do vento (N, NE, L, SE, S, SO, O, NO) pode ser implementado por um codificador que recebe a direção, por meio de um código 1-entre-8, e fornece essa mesma direção em um código binário de três bits, ou seja, em uma forma mais adequada para lidar com circuitos digitais.

- Observação importante: evidentemente que existem codificadores para outras representações de saídas. Por exemplo, codificadores decimais têm quatro saídas para códigos como o BCD e Excesso-3; a definição e implementação deles são semelhantes às do codificador binário, de modo que esse assunto será cobrado como forma de exercício. Implemente!

- Outra observação importante: como no caso dos decodificadores, bem como para todos os módulos-padrão que serão discutidos adiante, sempre há a limitação quanto ao número máximo de conexões e portas lógicas em uma implementação de um módulo. Por conta disso, codificadores grandes são geralmente construídos como redes codificadoras a partir de codificadores-padrão que apresentam no máximo 16 entradas.
- Para os curiosos: o codificador binário tem a limitação que, em qualquer instante, no máximo uma entrada pode ter valor um. Contudo, há diversas aplicações em que mais de uma entrada seja um e, ainda assim, seja necessário codificá-la segundo algum **critério de prioridade**. Por exemplo, imagine que o processador de um computador receba pedidos de atenção por meio de sinais de interrupção. Diversos desses sinais podem estar ativos ao mesmo tempo, sendo necessário selecionar um deles de acordo com uma escala de prioridade pré-definida. Para fazer isso, é necessário um **codificador de prioridade**, que nada mais é do que um codificador com um módulo adicional para determinar dentre os sinais de entrada qual deles representa o bit mais significativo e dar prioridade para ele durante a codificação. Pesquise mais sobre esse assunto!

- Exercício proposto: especifique e implemente decodificadores decimais (dez saídas) para os seguintes códigos: BCD, código de Gray de 4 bits e código 2-4-2-1. Em cada um deles, forneça a tabela verdade, as expressões de chaveamento e o circuito.
- Exercício proposto: especifique e implemente codificadores decimais (dez entradas) para os seguintes códigos: BCD, código de Gray de 4 bits e código 2-4-2-1. Em cada um deles, forneça a tabela verdade, as expressões de chaveamento e o circuito.
- Exercício proposto: utilize **decodificadores e codificadores binários** (e portas lógicas quando conveniente) para implementar conversores de código entre: BCD para código de sete segmentos (para acionar um display de sete segmentos), binário de 4bits para código de Gray de 4 bits, binário de 4bits para código Excesso-3.