

Circuitos Combinacionais

GEN 253 – Circuitos Digitais

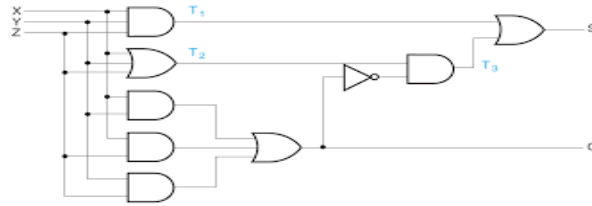
Prof. Luciano L. Caimi
lcaimi@uffs.edu.br

Circuitos Combinacionais: em que as saídas do circuito dependem exclusivamente do valor presente na entrada. Exemplos: multiplexadores, somadores, codificadores, etc

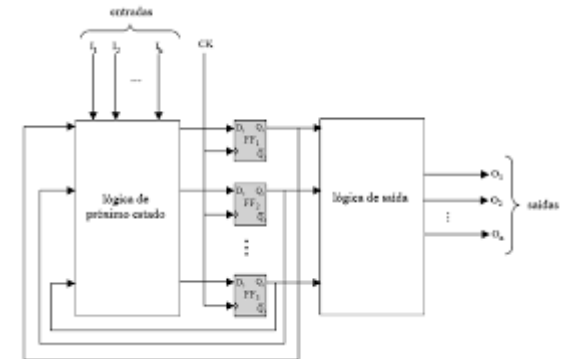
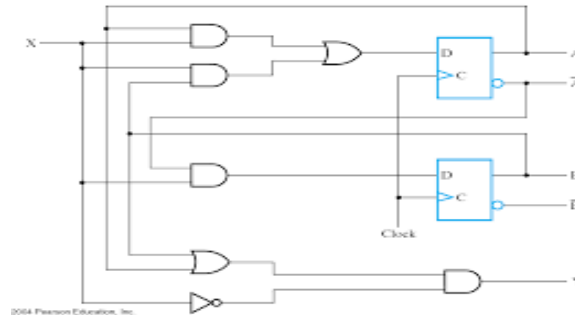
Circuitos Sequenciais: onde as saídas do circuito dependem dos valores presentes nas entradas e do estado anterior em que o circuito se encontra. Exemplo: registradores, contadores, máquinas de estado finito (FSM), memórias, etc

Introdução

Circuitos Combinacionais: as saídas do circuito dependem exclusivamente do valor presente na entrada



Circuitos Sequenciais: as saídas do circuito dependem dos valores presentes nas entradas e do estado anterior em que o circuito se encontra



Tipos de Circuitos Combinacionais

Circuitos de Interconexão: seletores (conhecidos como multiplexadores), codificadores e decodificadores

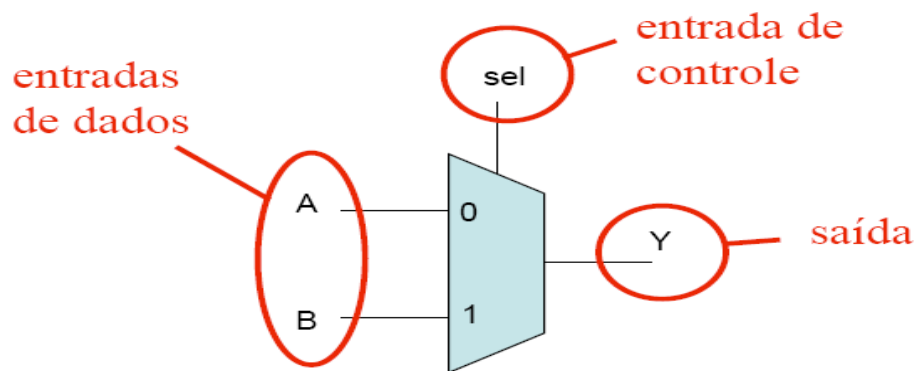
Circuitos Aritméticos: somadores, subtratores, somadores/subtratores, multiplicadores, deslocadores, comparadores e ULAS (circuitos que combinam mais de duas operações aritméticas e/ou lógicas)

Circuitos Combinacionais

Multiplexadores (ou seletores)

Multiplexador 2x1:

Sua função é selecionar uma dentre as duas entradas de dados, fazendo a entrada selecionada aparecer na saída

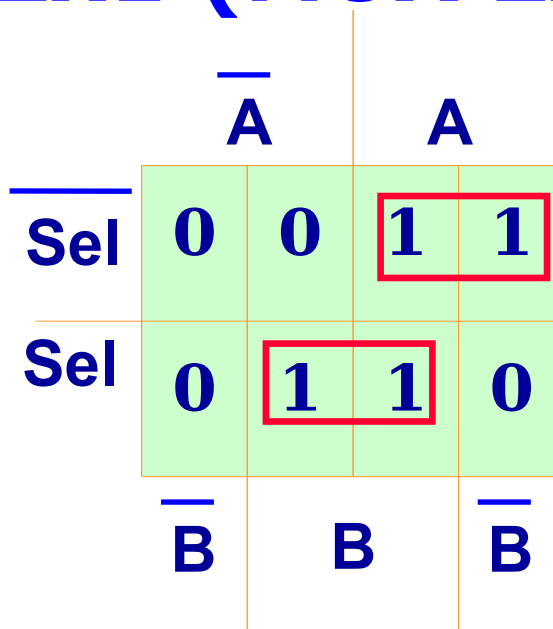


Sel	A	B	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

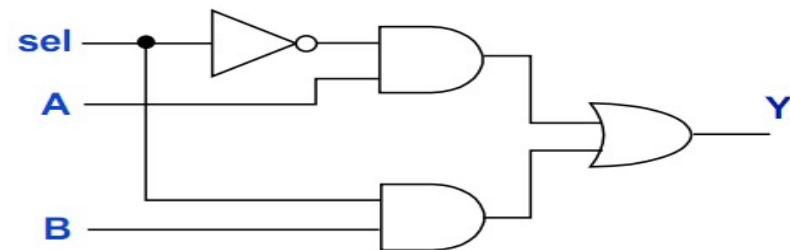
Circuitos Combinacionais

Multiplexador 2x1 (MUX 2x1)

Sel	A	B	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



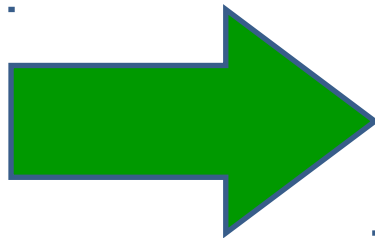
$$Y = \overline{Sel} \cdot A + Sel \cdot B$$



Multiplexador 2x1 (MUX 2x1)

Outra forma de ver a tabela

Sel	A	B	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



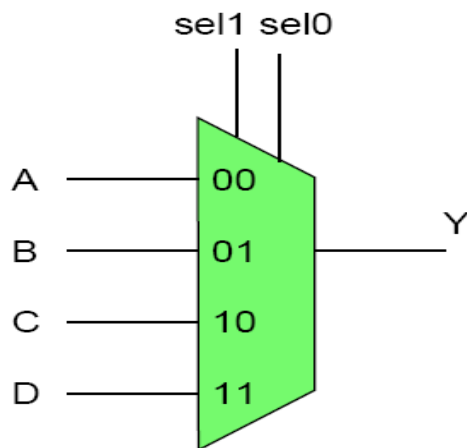
Sel	Y
0	A
1	B

$$Y = \overline{\text{Sel}} \cdot A + \text{Sel} \cdot B$$

Circuitos Combinacionais

Multiplexador 4x1 (MUX 4x1)

Sua função é selecionar uma dentre as quatro entradas de dados, fazendo a entrada selecionada aparecer na saída



símbolo

Sel1	Sel0	Y
0	0	A
0	1	B
1	0	C
1	1	D

$$Y = \overline{\text{Sel1}} \cdot \overline{\text{Sel0}} \cdot A + \overline{\text{Sel1}} \cdot \text{Sel0} \cdot B + \text{Sel1} \cdot \overline{\text{Sel0}} \cdot C + \text{Sel1} \cdot \text{Sel0} \cdot D$$

Implementação de Funções

Um multiplexador pode ser utilizado na implementação de uma função combinacional diretamente a partir da tabela-verdade

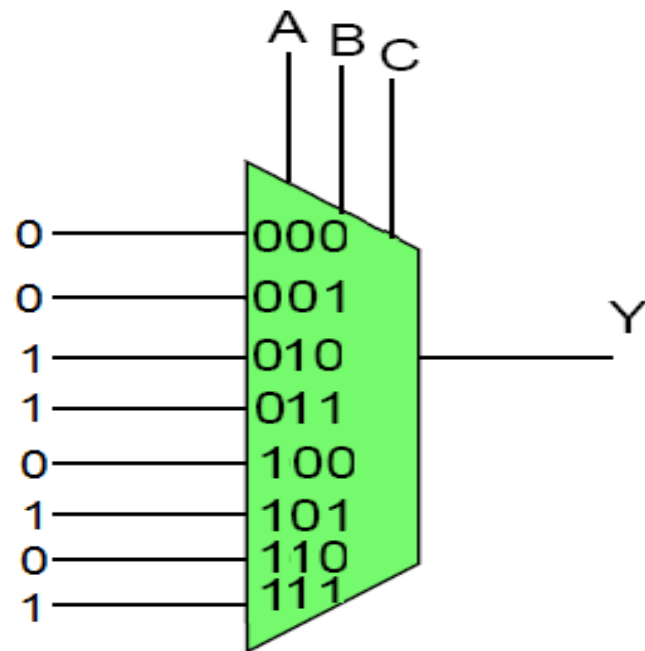
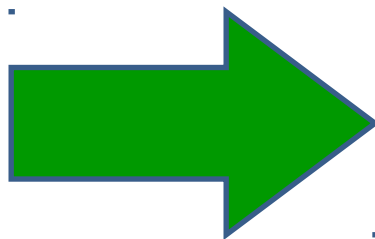
Os mintermos de uma função são gerados por um multiplexador através das entradas de seleção, restando apenas ligar as entradas de dados a 0 ou a 1 em conformidade com a respectiva tabela-verdade.

Uma função de N variáveis pode ser implementada com um MUX $2^N:1$ (N entradas de seleção e 2^N entradas de dados).

Circuitos Combinacionais

Implementação de Funções

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1



Exercícios:

- a) Implemente um MUX 8x1 (tabela-verdade, equação, circuito)
- b) Implemente um MUX 4x1 utilizando somente MUX 2x1
- c) Implemente um MUX 8x1 utilizando somente MUX 4x1

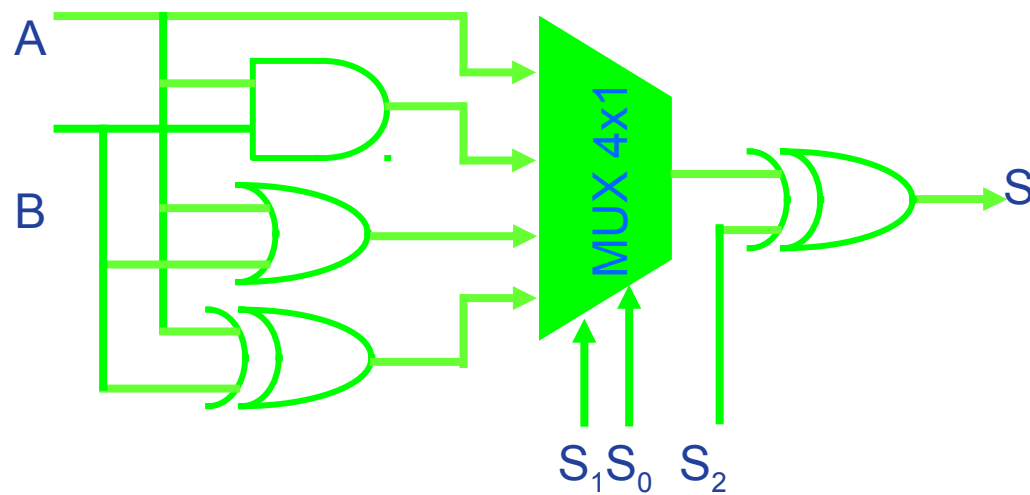
Circuitos Combinacionais



Aplicações

MUX são utilizados para seleção de caminhos de dados em Unidades Lógicas e Aritméticas (ULAs). Observe sua utilização em uma ULA que possui a seguinte tabela verdade

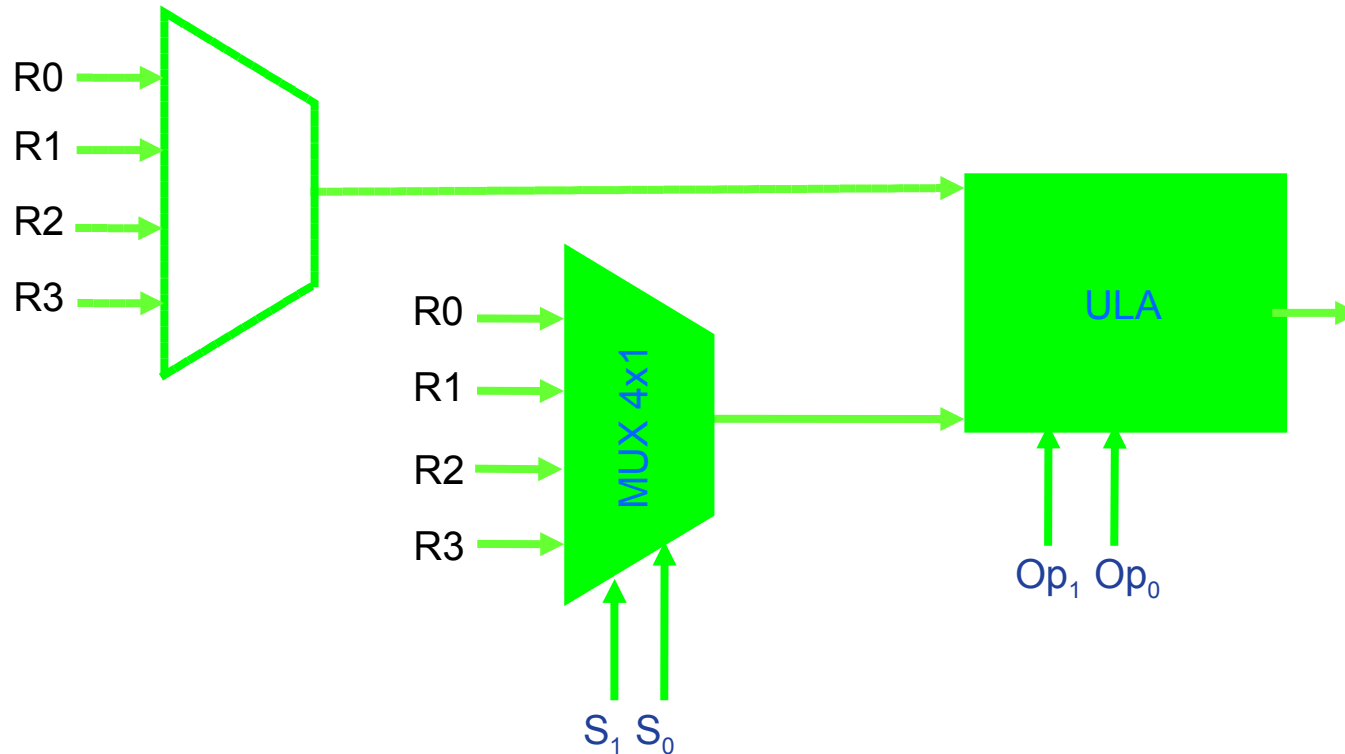
S2	S1	S0	Saída
0	0	0	A
0	0	1	A AND B
0	1	0	A OR B
0	1	1	A XOR B
1	0	0	NOT A
1	0	1	A NAND B
1	1	0	A NOR B
1	1	1	A XNOR B



Circuitos Combinacionais

Aplicações

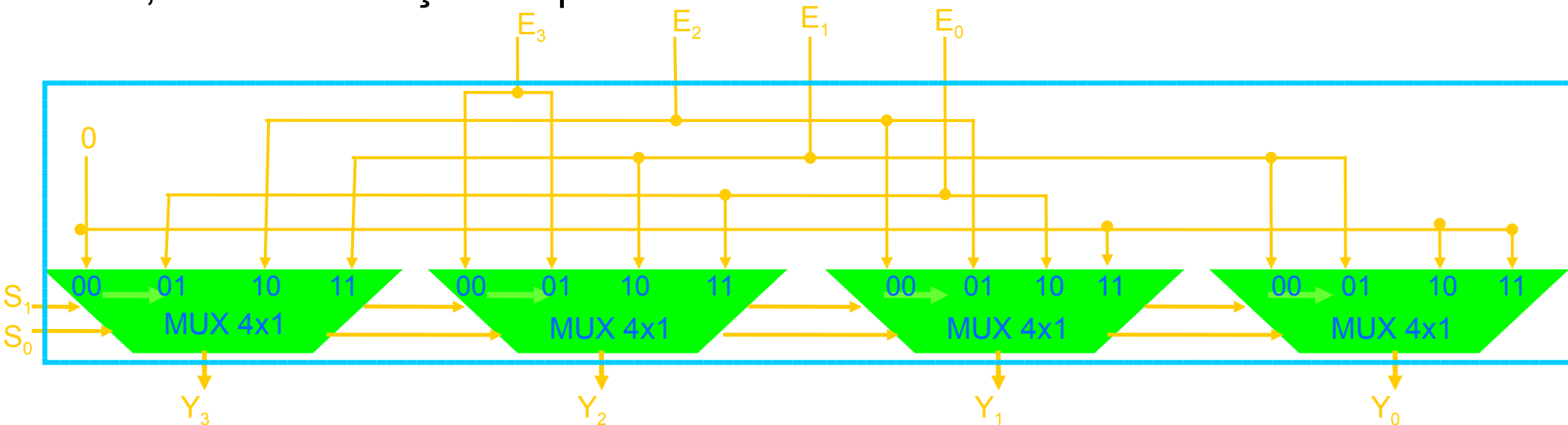
Ainda no que diz respeito a seleção de caminhos de dados



Circuitos Combinacionais

Aplicações

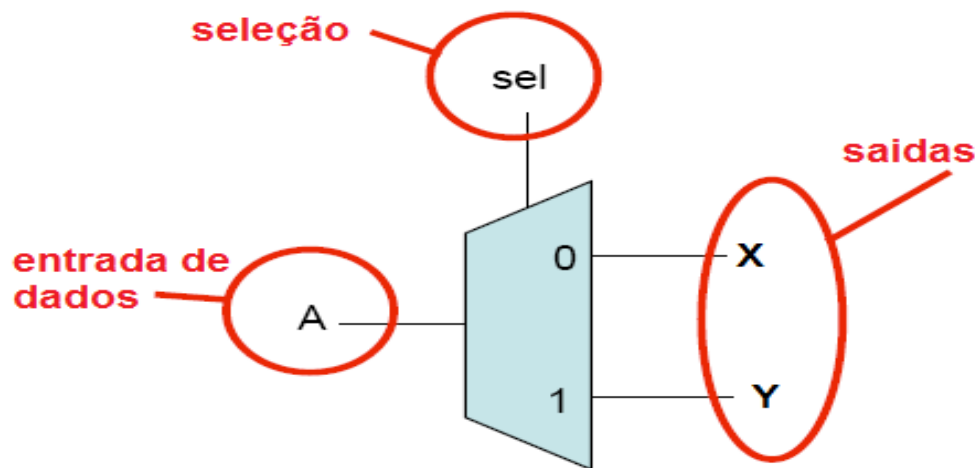
O MUX pode ser usado para construção de deslocadores (shifters), que recebem uma palavra binária e fornecem o valor deslocado a direita ou a esquerda, com a inclusão de “0” ou “1”, ou até a rotação da palavra



Demultiplexadores

Demultiplexador 1x2:

Realiza a função inversa dos multiplexadores, ou seja direciona a entrada para uma de 2 possíveis saídas



Sel	A	X	Y
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

Circuito???

Demultiplexadores

Demultiplexador 1x4:

Direciona a entrada para uma de 4 possíveis saídas

Sel0	Sel1	A	S0	S1	S2	S3
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	0	0
1	1	1	0	0	0	1

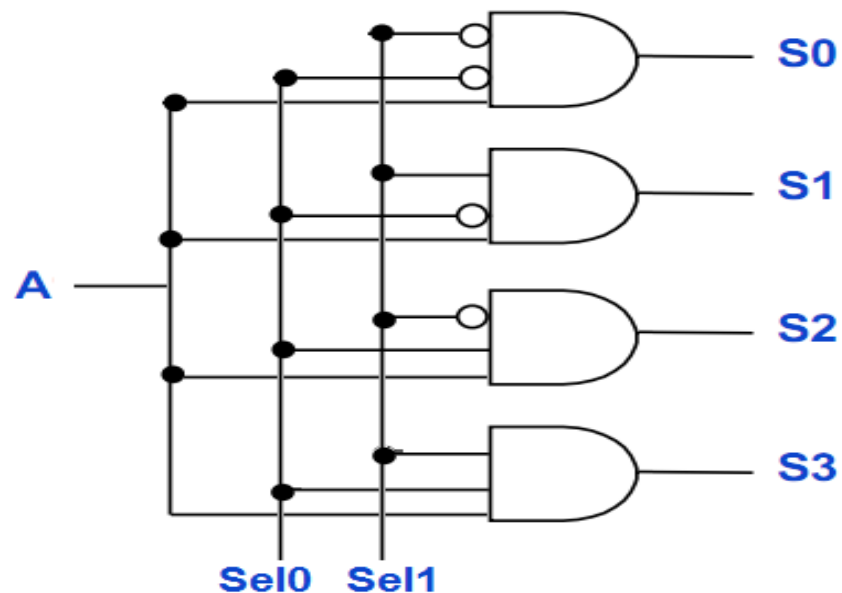
Sel0	Sel1	S0	S1	S2	S3
0	0	A	0	0	0
0	1	0	A	0	0
1	0	0	0	A	0
1	1	0	0	0	A

Circuito???

Demultiplexadores

Demultiplexador 1x4:

Sel0	Sel1	S0	S1	S2	S3
0	0	A	0	0	0
0	1	0	A	0	0
1	0	0	0	A	0
1	1	0	0	0	A



Exercícios:

- a) Implemente um DEMUX 1x8 (tabela-verdade, equação, circuito)
- b) Implemente um DEMUX 1x4 utilizando somente DEMUX 1x2
- c) Implemente um DEMUX 1x8 utilizando somente DEMUX 1x4

Decodificadores

Decodificador 2:4

Sua função é **ativar** uma e somente uma dentre as 4 saídas, de acordo com a combinação de valores das entradas

Ativar, neste caso, quer dizer diferenciar, destacar

Existe uma relação entre o número de saídas (ns) e o número de entradas (ne):

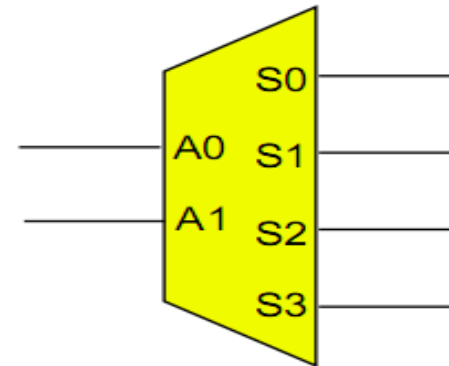
$$ns = 2^{ne}$$

Circuitos Combinacionais

Decodificadores

Decodificador 2:4

entradas		saídas			
A0	A1	S0	S1	S2	S3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



símbolo

Decodificadores

Decodificador 2:4

Cada combinação de entrada pode ser vista como o endereço de uma saída específica

entradas		saídas			
A0	A1	S0	S1	S2	S3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

endereço da saída
(=2 em decimal)

saída S2
ativada

Circuitos Combinacionais

Decodificadores

Decodificador 2:4

Cada uma das 4 saídas corresponde a um mintermo diferente

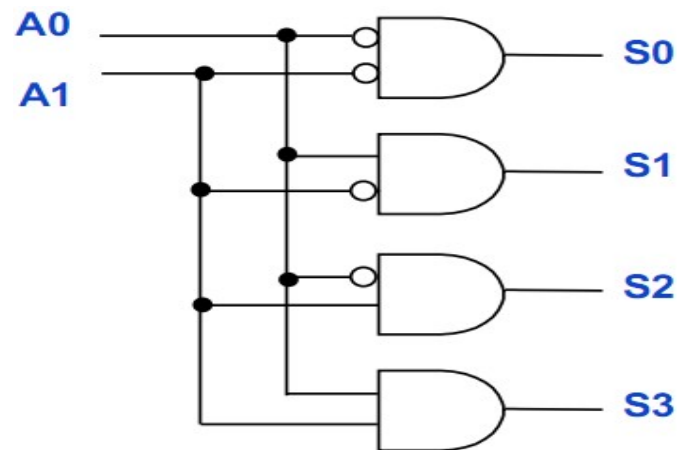
A0	A1	S0	S1	S2	S3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$S0 = \overline{A0} \cdot \overline{A1}$

$S1 = \overline{A0} \cdot A1$

$S2 = A0 \cdot \overline{A1}$

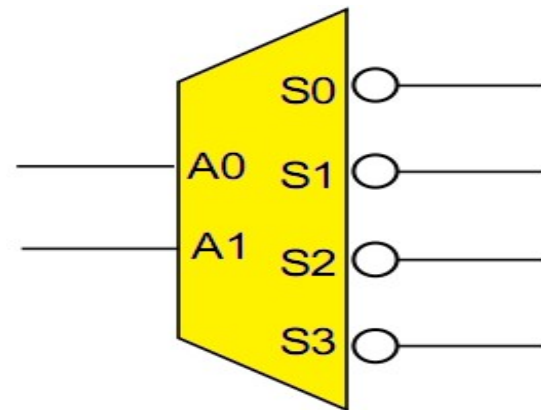
$S3 = A0 \cdot A1$



Decodificadores

Decodificador 2:4 - com saídas em lógica invertida (ou complementar)

entradas		saídas			
A0	A1	S0	S1	S2	S3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0



símbolo

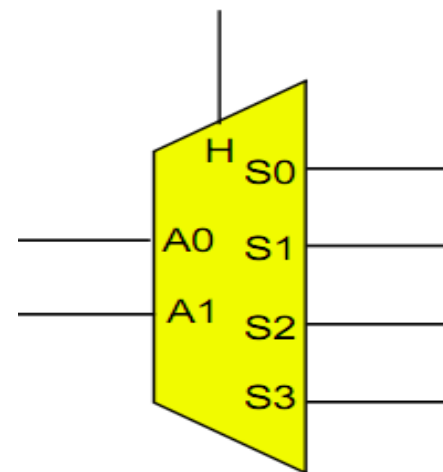
► Decodificadores

Decodificador 2:4 – acrescentando uma entrada de habilitação

entradas saídas (enable)

A1	S0	S1	S2	S3
X	0	0	0	0
0	1	0	0	0
1	0	1	0	0
0	0	0	1	0
1	0	0	0	1

todas saídas
desativadas



símbolo

Decodificadores

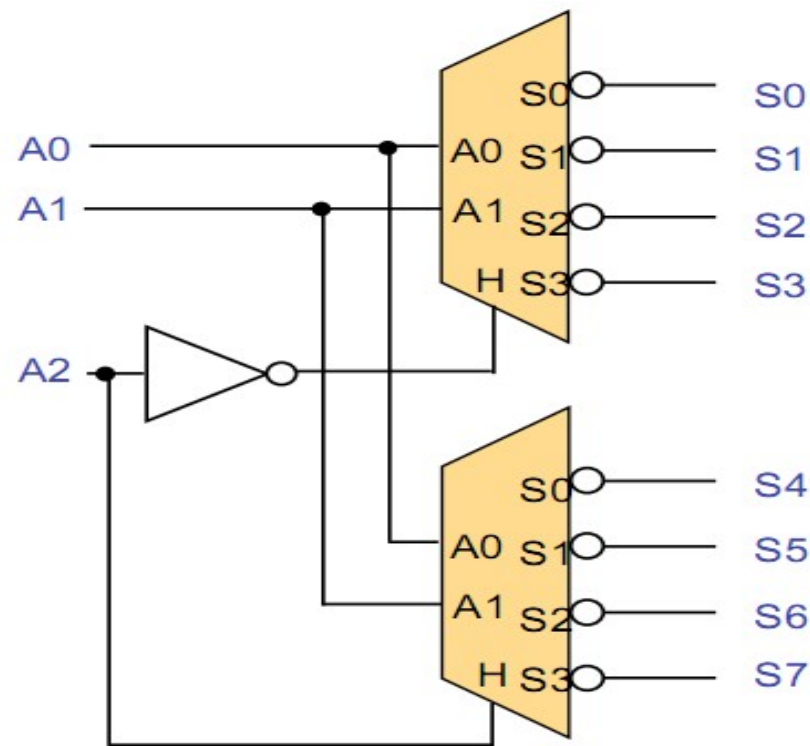
Decodificador 3:8, 4:16, 5:32, etc

- ✓ Seguem o mesmo princípio dos decodificadores vistos, sempre observando a relação $n:2^n$ (número de entradas: número de saídas)
- ✓ Também pode-se construir um decodificador a partir de decodificadores menores, que possuam entrada de habilitação

Circuitos Combinacionais

Decodificadores

Decodificador 3:8 - formado de 2:4, complementar, sem entrada de habilitação



Circuitos Combinacionais

Exemplo Decodificador BCD

BCD				O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

Circuito???

Alguns Códigos

Numéricos

- Código BCD (0000; 0001; 0010 ...)
- Código Excesso de 3 (0011; 0100; 0101 ...)
- Código Gray (000; 001; 011; 010; 110 ...)
- 7 segmentos

Alfanuméricos

- Código ASCII (7 bits; estendida: 8bits)
- Código Unicode (16 bits e 32 bits)
- Código UTF
- Código ISO 8859

Codificadores

Conceito: grosso modo, codificadores realizam a função oposta dos decodificadores

Codificadores servem para reduzir o número de bits necessários para a representação de alguma informação (facilitando sua manipulação e seu armazenamento)

Os principais tipos de codificadores são: binários, de prioridade

Circuitos Combinacionais



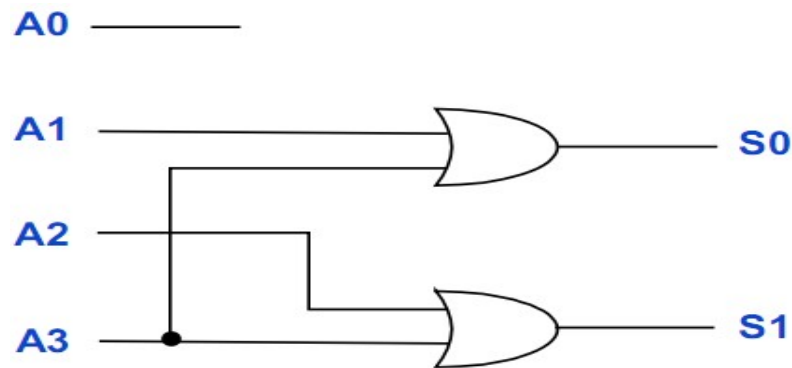
Codificadores

Codificador binário 4:2

Apenas as situações de entrada contendo somente uma posição valendo 1 são consideradas

As demais situações são tratadas como *don't cares* (usar Karnaugh)

entradas				saídas	
A3	A2	A1	A0	S1	S0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1



Exercícios

- Implementar um codificador BCD
- Implementar um codificador Gray
- Implementar um decodificador Gray
- Implementar um decodificador Gray/7 segmentos

