

Instruções para entrega dos exercícios

- Você **deve** representar claramente o desenvolvimento das questões.
- Plágio não será tolerado.
- Envie um único arquivo pdf com as repostas de todos os exercícios que exigem escrita, devidamente numerados.
- Para os exercícios de implementação/projeto de circuitos, utilize o simulador de circuitos Digital e entregue um arquivo para cada exercício.
- No simulador, use componentes *FET tipo N* para realizar a função de uma chave.

1. [RH] Considere os tempos de propagação exibidos na tabela abaixo. Calcule os tempos de propagação do demultiplexador de duas saídas e do demultiplexador de oito saídas vistos em sala de aula.

Tabela 1: Tempo de propagação de portas lógicas em nanosegundos

porta	T_p	porta	T_p
<i>not</i>	2	<i>xor-2</i>	6
<i>and-2</i>	4	<i>or-2</i>	4
<i>and-3</i>	6	<i>or-3</i>	6
<i>and-4</i>	8	<i>or-4</i>	8
<i>and-5</i>	10	<i>or-5</i>	10
<i>and-6</i>	12	<i>or-6</i>	12

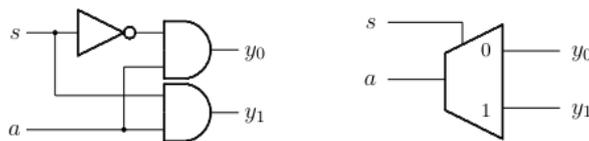


Figura 1: Demultiplexador de 2 saídas. Fonte: RH

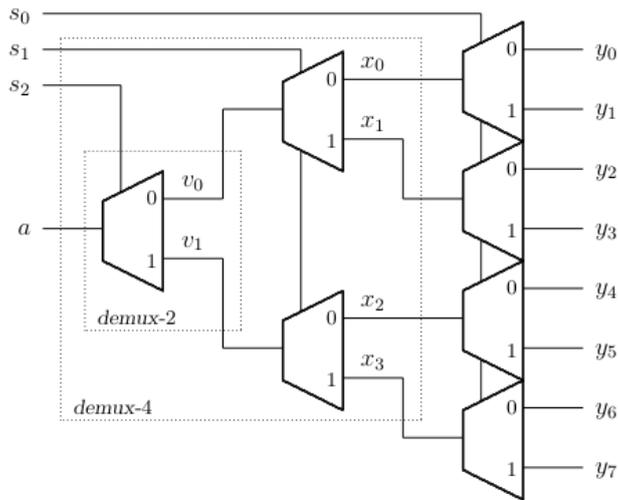


Figura 2: Demultiplexador de 8 saídas. Fonte: RH

2. [RH] Para somar dois números representados em notação científica, ou em ponto flutuante, deve-se deslocar o expoente do menor número para ‘emparelhar’ as vírgulas e então efetuar a soma.

Considere um circuito que tem como entrada um vetor de quatro bits $B = \langle b_3, b_2, b_1, b_0 \rangle$, e duas saídas: a saída V é 1 sempre que os quatro bits de B são zero, e a saída $N = \langle n_1, n_0 \rangle$ é o número (índice) da posição do bit mais à esquerda (mais significativo) que *não é zero* – o número N indica qual é a posição do ‘primeiro’ bit em 1.

A tabela abaixo mostra as combinações de entradas e saídas para a detecção do bit mais significativo de um quarteto de bits. Quando $B = \langle 0, 0, 0, 0 \rangle$, então $V = 1$ e N pode ser ignorado pois não há nenhum bit em 1. Quando $V = 0$, N deve indicar qual é o bit mais significativo que não é zero. Se $b_k = 1$, então o valor dos $b_i, i < k$ é irrelevante, e isso é denotado com ‘X’ na tabela verdade.

Você deve construir o circuito combinacional que implementa esse comportamento.

b_3	b_2	b_1	b_0	n_1	n_0	V
0	0	0	0	0	0	1
0	0	0	1	0	0	0
0	0	1	X	0	1	0
0	1	X	X	1	0	0
1	X	X	X	1	1	0

3. [RH] Implemente uma ROM 4×3 que tenha duas entradas (a_1, a_0) e implemente as 3 funções a seguir nos bits de saída (d_2, d_1, d_0). Indique qual é cada uma das três funções implementadas (o que elas significam).

Tabela 2: Três funções de duas variáveis implementadas com uma ROM 4×3

a_1	a_0	d_2	d_1	d_0
0	0	1	1	0
0	1	0	0	1
1	0	0	1	1
1	1	1	0	0

4. [RH] Mostre como implementar um circuito demux-2 com 4 portas *nor*.
5. [RH] Implemente um comparador de magnitude ($m = A > B$) para números de dois bits. Escreva a tabela verdade, simplifique usando mapa de *Karnaugh* e depois implemente o circuito.

6. [RH] Implemente o circuito de uma ROM 16×4 : 16 palavras de 4 bits.
7. [RH] Mostre como implementar os operadores de conjunção, disjunção e negação usando um multiplexador (*mux*)