

Arquitetura de computadores e sistemas operacionais

Circuitos combinacionais

Gabriel V C Candido
gabriel.candido@ifpr.edu.br

Instituto Federal do Paraná - Pinhais

Sumário

Circuitos combinacionais básicos

Memória

Sumário

Circuitos combinacionais básicos

Memória

Multiplexador

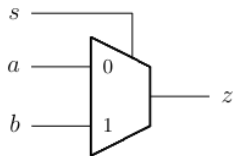
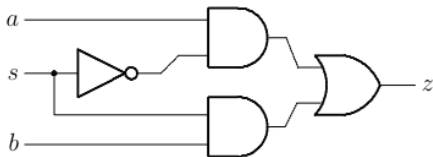


Figura: Multiplexador de 2 entradas. Fonte: RH

Multiplexador

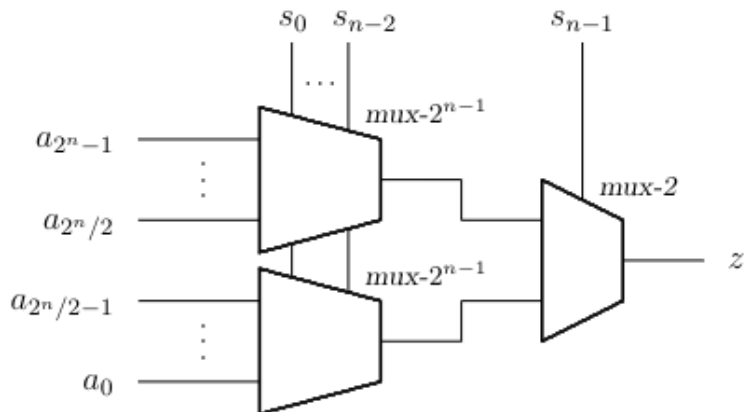


Figura: Multiplexador de 2^n : um $\text{mux-}2$ e dois $\text{mux-}2^{n-1}$.

Fonte: RH

Multiplexador

índice	t	p	q	r
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	1	0	1	1
4	0	1	0	0
5	1	1	0	1
6	1	1	1	0
7	1	1	1	1

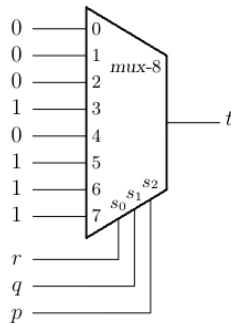


Figura: Implementação de função de 3 variáveis com mux-8.

Fonte: RH

Multiplexador

índice	t	p	q	r	t'
0	0	0	0	0	0
1	0	0	0	1	
2	0	0	1	0	r
3	1	0	1	1	
4	0	1	0	0	r
5	1	1	0	1	
6	1	1	1	0	1
7	1	1	1	1	

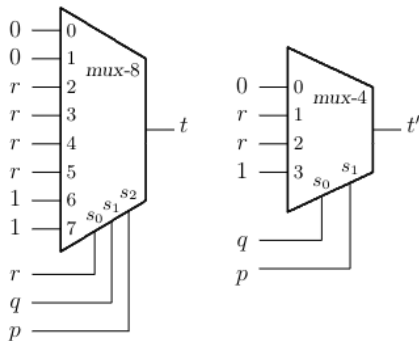


Figura: Implementação de função de 3 variáveis com mux-4.

Fonte: RH

Decodificador ou seletor

s	y_1	y_0
0	0	1
1	1	0

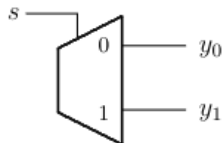
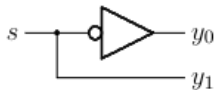


Figura: Decodificador de 2 saídas. Fonte: RH

Decodificador ou seletor

s	y_1	y_0
0	0	1
1	1	0

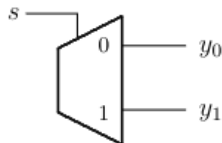
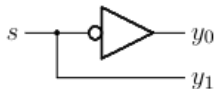


Figura: Decodificador de 2 saídas. Fonte: RH

Pode ser usado para selecionar qual dispositivo será habilitado para falar no barramento (exemplo: PCI)

Decodificador ou seletor

s_1	s_0	y_3	y_2	y_1	y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

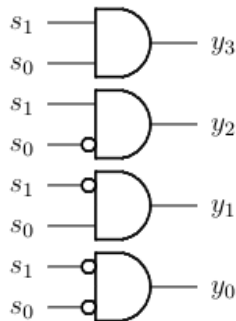


Figura: Decodificador de 4 saídas. Fonte: RH

Decodificador ou seletor

índice	t	p	q	r
0	0	0	0	0
1	1	0	0	1
2	0	0	1	0
3	1	0	1	1
4	0	1	0	0
5	1	1	0	1
6	1	1	1	0
7	0	1	1	1

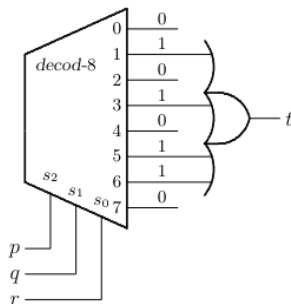


Figura: Implementação de função com decodificador de 8 saídas. Fonte: RH

Demultiplexador

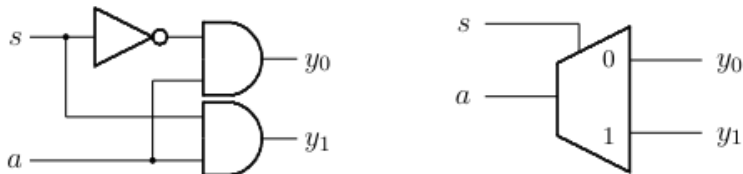
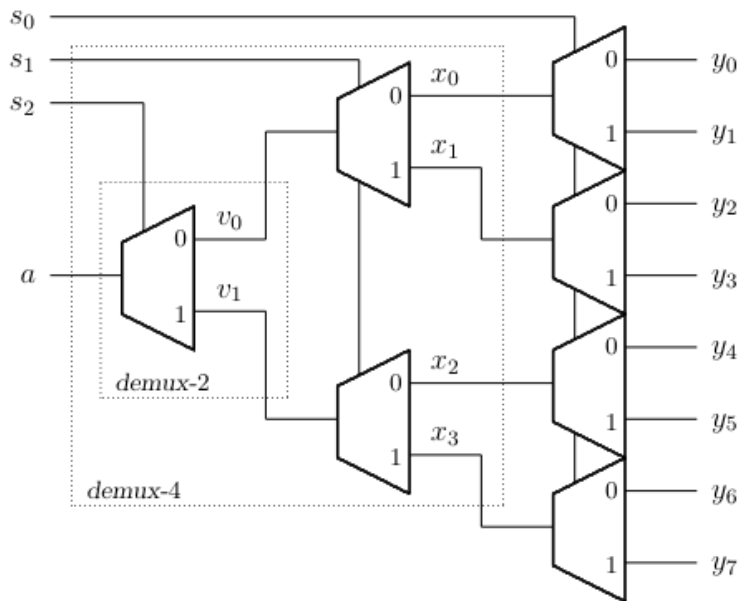


Figura: Demultiplexador de duas saídas. Fonte: RH

Demultiplexador



Sumário

Circuitos combinacionais básicos

Memória

Memória

Memória ROM: *Read-Only Memory* (somente leitura)

Memória RAM: *Random Access Memory* (leitura e escrita)

Memória

Memória ROM: *Read-Only Memory* (somente leitura)

Memória RAM: *Random Access Memory* (leitura e escrita)

Memórias são vetores de bits, acessados através de *endereços*

Memória

Memória ROM: *Read-Only Memory* (somente leitura)

Memória RAM: *Random Access Memory* (leitura e escrita)

Memórias são vetores de bits, acessados através de *endereços*

Do ponto de vista de fabricação, podem ser matrizes, para aproveitar melhor o espaço

Memória ROM

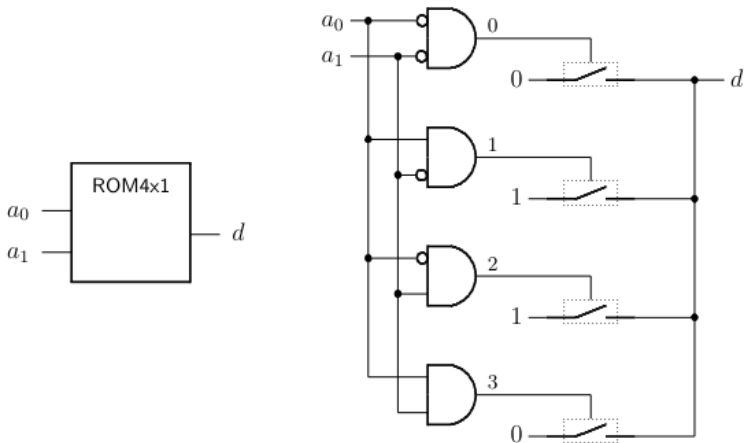


Figura: Memória ROM 4×1 . Fonte: RH

Memória ROM

Essa memória ROM 4×1 é puramente combinacional: ao passar o tempo de propagação das portas lógicas internas, a saída d estabiliza com o bit selecionado pelos sinais nas entradas de endereçamento.

Memória ROM

Essa memória ROM 4×1 é puramente combinacional: ao passar o tempo de propagação das portas lógicas internas, a saída d estabiliza com o bit selecionado pelos sinais nas entradas de endereçamento.

Para fazer uma ROM 8×1 basta duplicar as portas *and* e adicionar uma linha de endereçamento no decodificador.

Memória ROM

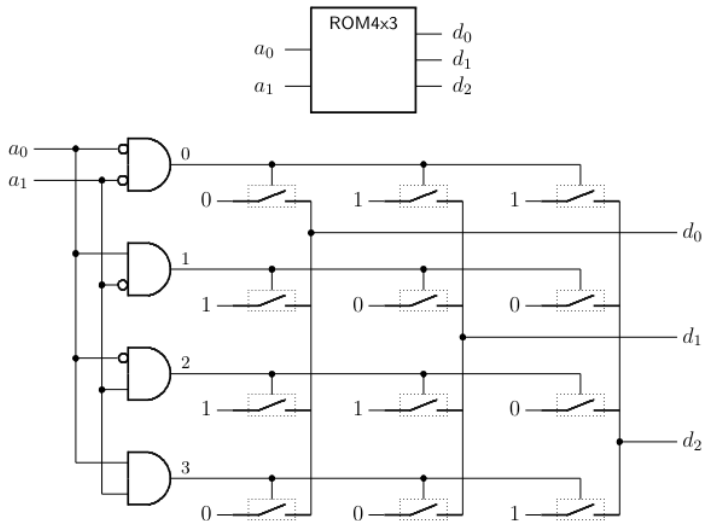


Figura: Memória ROM 4×3 . Fonte: RH

Memória ROM

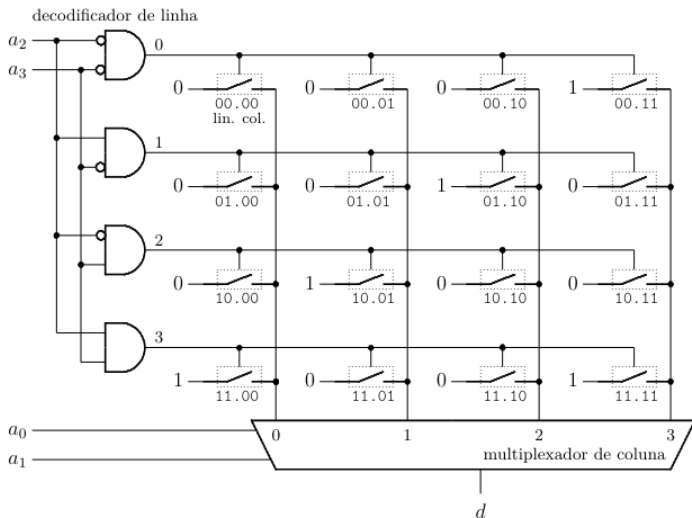


Figura: Memória ROM 16×1 . Fonte: RH

Memória RAM

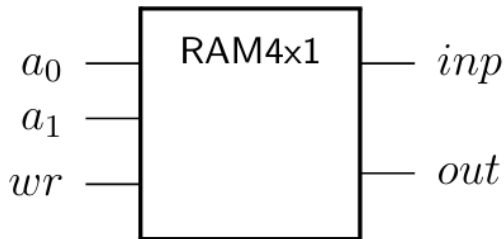


Figura: Memória RAM 4×1 . Fonte: RH

Memória RAM

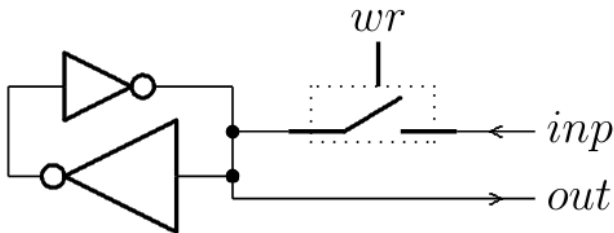


Figura: Circuito que implementa a RAM 4×1 . Fonte: RH

Memória RAM

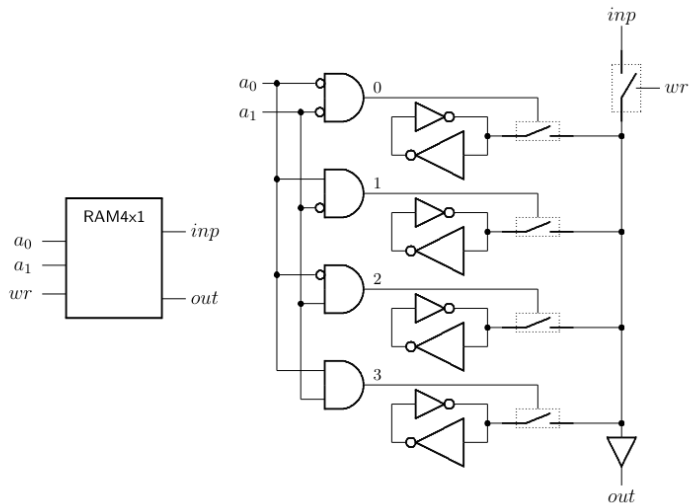


Figura: RAM 4×1 . Fonte: RH