

Codificadores e decodificadores

Sequências de **bits** podem ser usadas para codificar valores numéricos. Dois códigos binários têm aplicações especiais: o BCD (**Binary-Coded Decimal**) e o de Gray.

Código BCD

O código BCD é basicamente uma forma de codificar valores numéricos na base 10 em seus equivalentes binários, dígito por dígito, individualmente.

Decimal	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

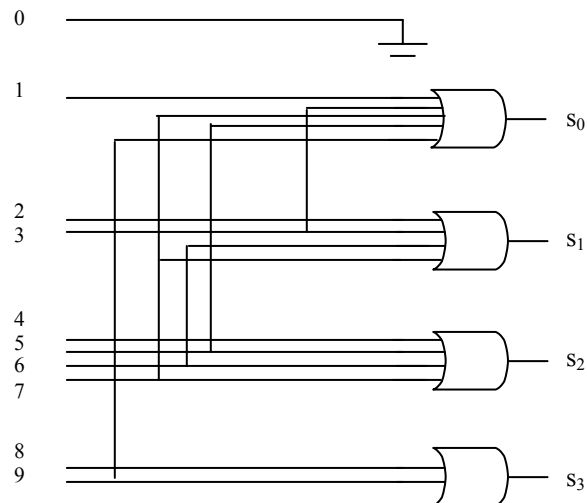
Exemplos:

Codificar o valor decimal 369 em código BCD: $369 = (0011) (0110) (1001)$

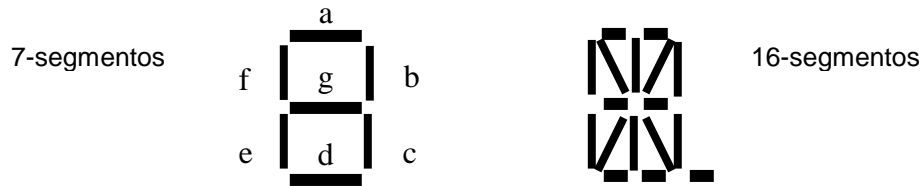
Em operações com valores em BCD, os resultados devem ser reajustados:

$$9 + 8 = (1001)_{(2\text{-BCD})} + (1000)_{(2\text{-BCD})} = 10001_{(2)} = [(0001) (0111)]_{(2\text{-BCD})} = (1)(7) = 17$$

Um circuito codificador “*decimal-para-BCD*” é aquele capaz de mapear um conjunto de entradas (0-9) em outro conjunto de quatro valores binários ($s_3s_2s_1s_0$), se apenas uma das entradas for acionada.



Displays de 7-segmentos e 16-segmentos podem ser usados para mostrar letras e valores numéricos codificados.



Por exemplo, um decodificador de “BCD-para-7-segmentos” pode ser usado para exibir valores de 0 a 9. O esquema a seguir identifica os sinais necessários (**abcdefg**). A tabela abaixo indica as funções para mapear os sinais de entrada em BCD e as saídas no **display**.

Decimal	BCD	a	b	c	d	e	f	g
0	0000	1	1	1	1	1	1	0
1	0001	0	1	1	0	0	0	0
2	0010	1	1	0	1	1	0	1
3	0011	1	1	1	1	0	0	1
4	0100	0	1	1	0	0	1	1
5	0101	1	0	1	1	0	1	1
6	0110	0	0	0	1	1	1	1
7	0111	1	1	1	0	0	0	0
8	1000	1	1	1	1	1	1	1
9	1001	1	1	1	1	0	1	1

Código de Gray

O código de Gray serve para expressar sequências binárias nas quais dois valores sucessivos tenham apenas um **bit** de diferença (distância de Hamming = 1).

Decimal	Binário	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

Esse código também é conhecido com “*código binário refletido*”, por causa da característica abaixo:

eixo de reflexão		000
		001
	00	011
	0	010
	1	110
	10	111
		101
		100

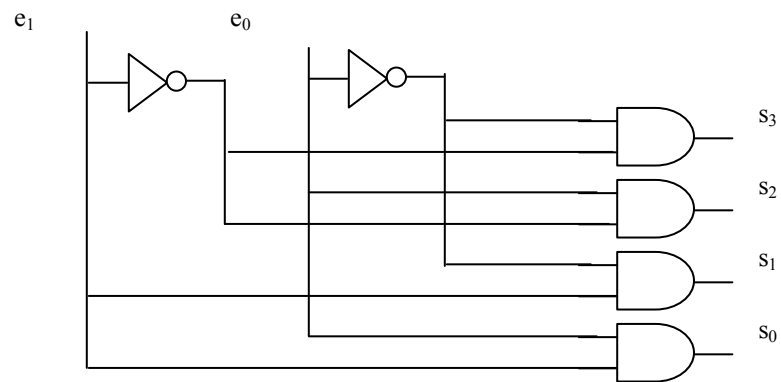
Decodificadores em geral são circuitos lógicos capazes de ativar uma saída de acordo com uma seleção de sinais de entrada.

Um decodificador de nível alto ativa uma saída quando uma das entradas estiver em nível 1 e as outras em nível 0.

Exemplo:

Montar um decodificador em nível alto para a tabela abaixo:

e_1	e_0	$s_3 s_2 s_1 s_0$
0	0	1 0 0 0
0	1	0 1 0 0
1	0	0 0 1 0
1	1	0 0 0 1

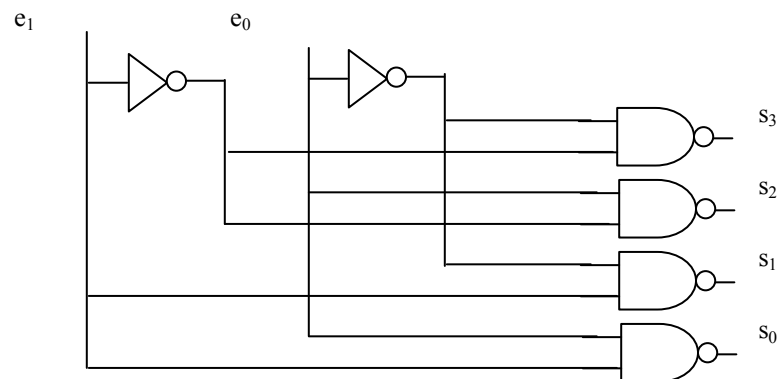


Um decodificador de nível baixo ativa uma saída quando uma das entradas estiver em nível 0 e as outras em nível 1.

Exemplo:

Montar um decodificador em nível baixo para a tabela abaixo:

e_1	e_0	$s_3 s_2 s_1 s_0$
0	0	0 1 1 1
0	1	1 0 1 1
1	0	1 1 0 1
1	1	1 1 1 0

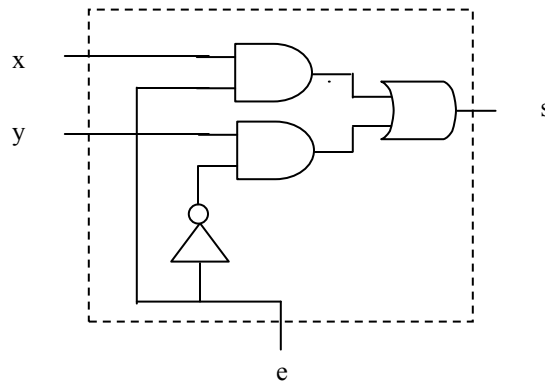


Multiplexadores (MUX) e demultiplexadores (DEMUX)

Multiplexadores (ou seletor de dados) são circuitos lógicos capazes de atuar como chaves digitais: recebem várias entradas e selecionam uma delas, em certo instante, e realizam sua transferência para a saída, mediante um código de seleção. Podem ser usados para rotear dados, sequenciar operações, realizar conversões do tipo paralelo-série e gerar formas de ondas.

Exemplo:

Dados os sinais de entrada (x) e (y), escolher a saída mediante um sinal de seleção.

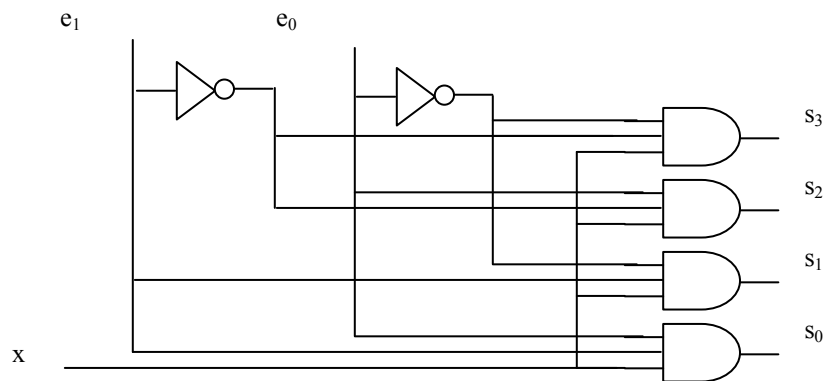


e	$s = x \cdot e + y \cdot e'$
0	y
1	x

Demultiplexadores (ou distribuidores de dados) são circuitos capazes de receber um sinal de entrada e distribuí-lo em várias saídas. Podem ser usados para distribuir um mesmo sinal de ativação ou sequenciamento (**clock**) para vários circuitos.

Exemplo:

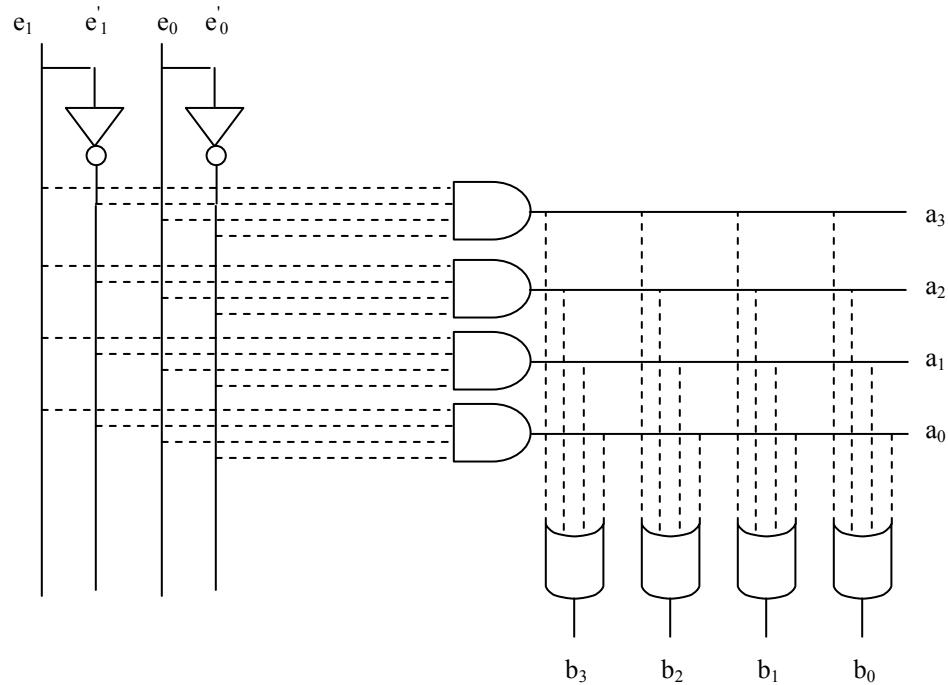
Dados um sinal de entrada (x) e dois sinais de ativação e_0 e e_1 , distribuí-lo à saída.



e_1	e_0	$e_1 \ e_0 \ x$	$s_3 \ s_2 \ s_1 \ s_0$
0	0	$s_3 = 0 \ 0 \ X$	$X \ 0 \ 0 \ 0$
0	1	$s_2 = 0 \ 1 \ X$	$0 \ X \ 0 \ 0$
1	0	$s_1 = 1 \ 0 \ X$	$0 \ 0 \ X \ 0$
1	1	$s_0 = 1 \ 1 \ X$	$0 \ 0 \ 0 \ X$

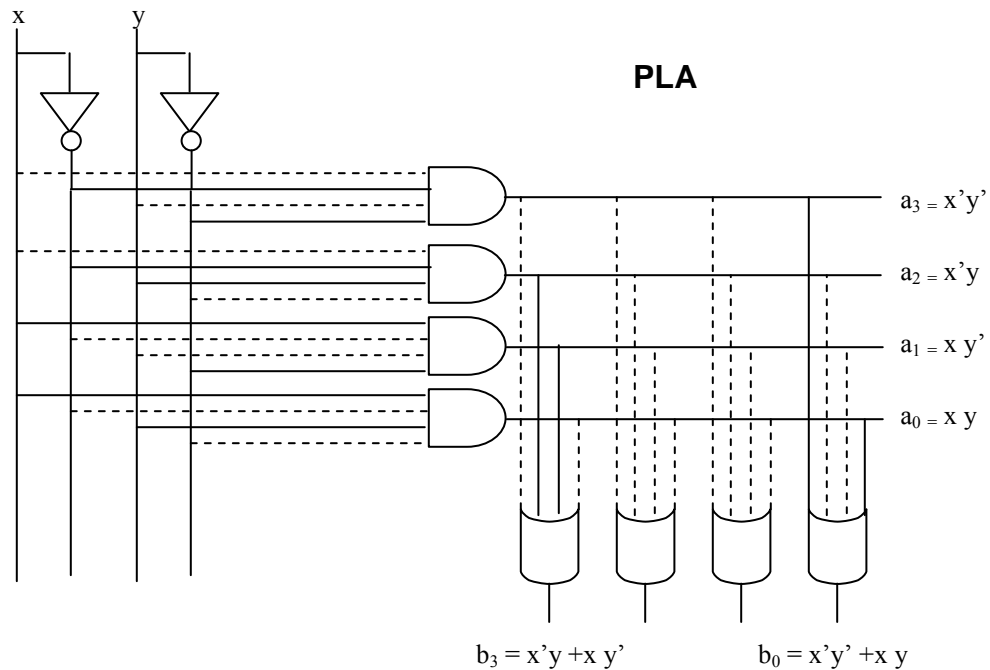
Dispositivos lógicos programáveis

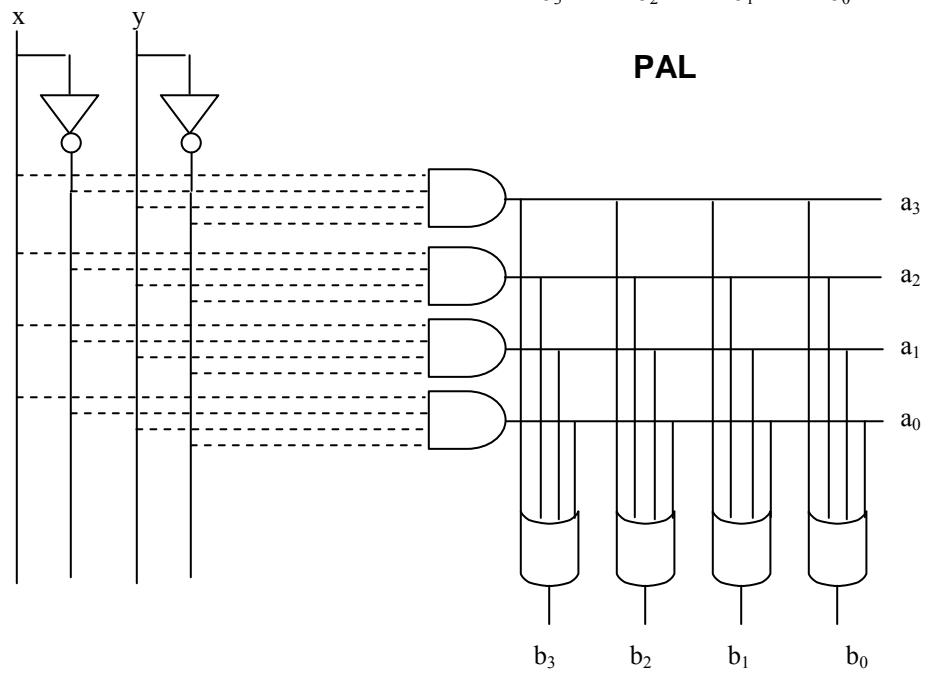
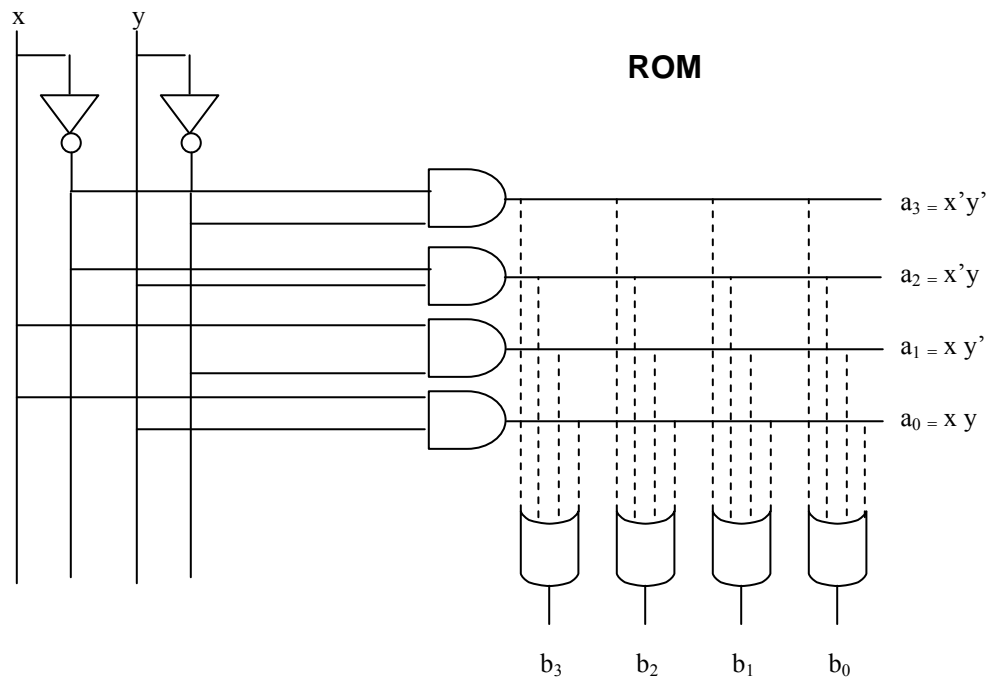
PLDs (**Programmable Logic Devices** ou FPGAs – **Field Programmable Gate Arrays**) são arranjos de portas lógicas que servem para uma rápida implementação de circuitos razoavelmente complexos. A figura abaixo ilustra uma organização de um arranjo de portas AND e OR.



Há três tipos comuns de arranjos lógicos combinacionais:

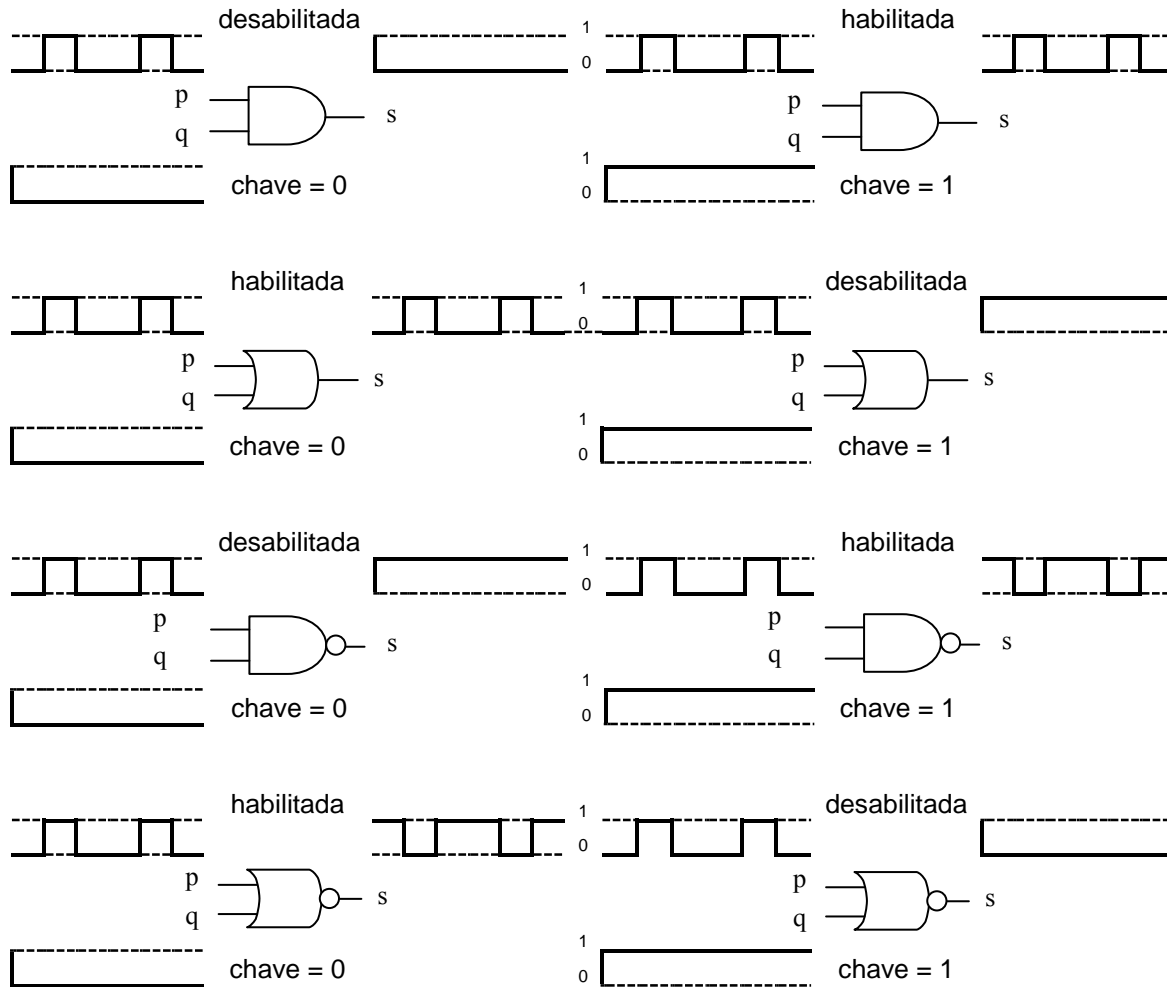
- PLA (**Programmable Logic Array**) - AND's e OR's programáveis
- ROM (**Read-Only Memory**) - AND's fixas e OR's programáveis
- PAL (**Programmable Array Logic**) - AND's programáveis e OR's fixas





Circuitos com chaveamento

As portas lógicas básicas podem ser usadas para controlar a passagem (chavear) de um sinal de entrada (p) para a saída, se a outra entrada (q) for usada como o controle (chave).



Exemplo:

Montar um *circuito direcionador de pulsos*: envia um pulso de entrada (p) para uma saída ou outra, dependendo de uma chave de controle (q).

