

EXP. 5: CODIFICADOR/DECODIFICADOR/CONVERSÃO DE CÓDIGOS

1. Introdução:

Os **codificadores** são circuitos combinacionais, cuja função é fornecer na sua saída determinados códigos em função de um sinal ativado em uma de suas entradas. Através da fig. 1 se pode melhor entender o funcionamento de um codificador:

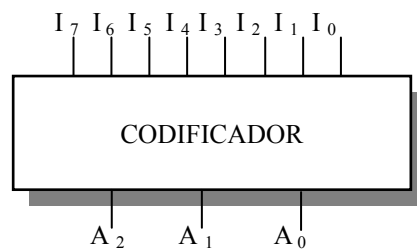


Fig. 1 - Diagrama típico de um codificador.

O codificador mostrado acima apresenta oito entradas (I_7 a I_0) e três saídas (A_2 , A_1 , A_0). A cada entrada “I” ativada na entrada vai ser liberado um código de saída, dado pela combinação dos três bits (A_2 , A_1 , A_0), neste caso supondo que uma única entrada é ativada em um determinado instante.

No caso de se ter vários sinais de entrada ativados simultaneamente, pode-se resolver qual código será liberado através da implementação de um mecanismo de prioridades.

A fig.2 mostra o diagrama do CI 74148, onde se pode verificar oito entradas (I_7 a I_0) e três saídas (A_2 , A_1 , A_0).

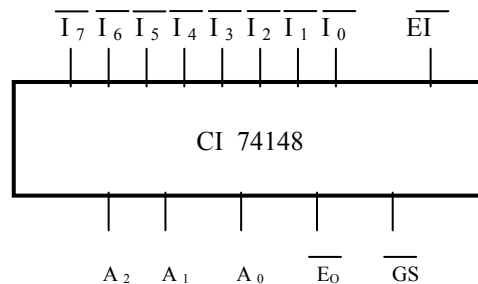


Fig. 2 - Diagrama do CI 74148 (Codificador com prioridade).

A entrada I_7 é a entrada mais prioritária e I_0 a de menor prioridade. O código relativo a uma entrada “I” é liberado somente quando o sinal nesta entrada for ativado e todos os outros sinais ativados (se houverem) forem de entradas de prioridade inferior. Por exemplo, se o pino I_6 estiver ativo (baixo) a saída será 001 (complemento de 6). GS (baixo) identifica a existência de uma solicitação de serviço. E_0 (baixo) indica a inexistência de qualquer solicitação. EI habilita a operação do codificador.

Os **DECODIFICADORES** executam a operação contrária à de um codificador, isto é, selecionam uma única linha de saída a partir do código presente na entrada (ver a figura 3).

A_2 A_1 A_0 E_1 $\overline{E_2}$ $\overline{E_3}$ (HABILITADORES)

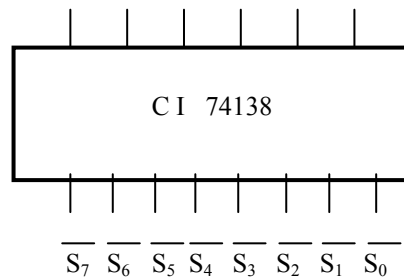


Fig. 3 - Decodificador 3/8 - CI 74138

O **DECODIFICADOR** permite selecionar uma das oito linhas de saída (S_7 a S_0) dependendo do código presente nas entradas A_2 A_1 A_0 . Algumas das aplicações mais comuns dos decodificadores são: Endereçamento de dispositivos de memória em computadores; Acionamento de dispositivos através de um endereço; Implementação de circuitos combinacionais, etc, etc...

Os **conversores de códigos** permitem converter informação binária de um código para outro. Um conversor de código muito utilizado é o conversor BCD/7 segmentos que permite mostrar via display de 7 segmentos o dígito decimal correspondente à entrada em binário (figura 4). Estes conversores são extremamente utilizados na maioria dos instrumentos digitais uma vez que realizam a interface com o homem. Exemplos: multímetros digitais, frequencímetros, relógios etc...

display 7 SEG.

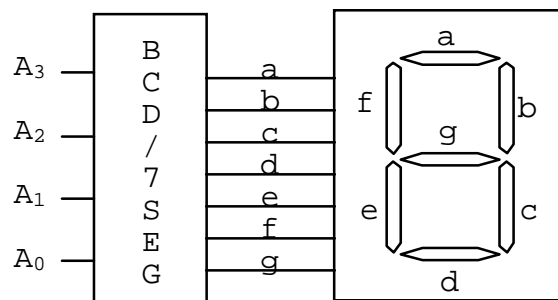
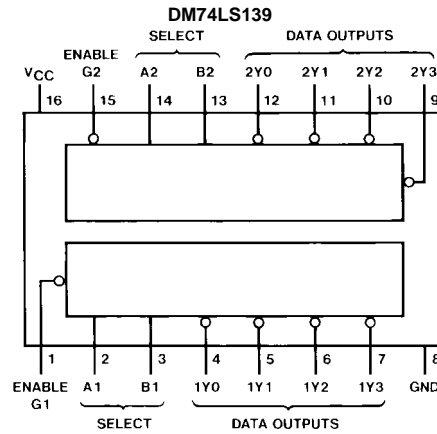
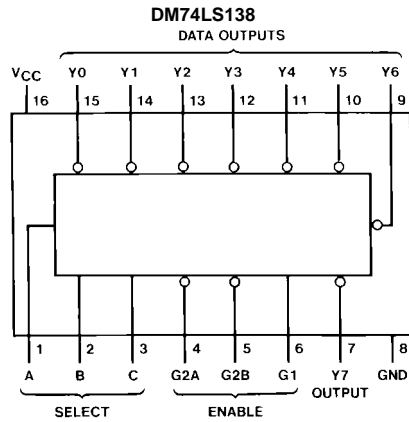


Fig. 4 - Esquema de ligação de um conversor para BCD/7 segmentos

2. Parte Experimental

- 2.1 Analise o funcionamento do circuito decodificador (CI 74138). Observe a tabela de função no manual de dados dos **CI**s. Implemente um simulador de casa habitada que comanda o acionamento de forma programada de 5 dispositivos num ciclo de operação de 8 horas.
- 2.2 Analise a tabela de função do conversor de código (LS7447 / 9368). Implemente um circuito conversor BCD/7segmentos de dois (2) dígitos que poderá ser utilizado na saída de qualquer instrumento digital.

Connection Diagrams



Function Tables

DM74LS138

Inputs			Outputs									
Enable		Select										
G1	G2 (Note 1)	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	H	H	H	H	H	H	H	H
H	L	L	L	H	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H
H	L	L	L	H	H	H	H	H	H	L	H	H
H	L	L	L	H	H	H	H	H	H	H	L	H
H	L	L	L	H	H	H	H	H	H	H	H	L

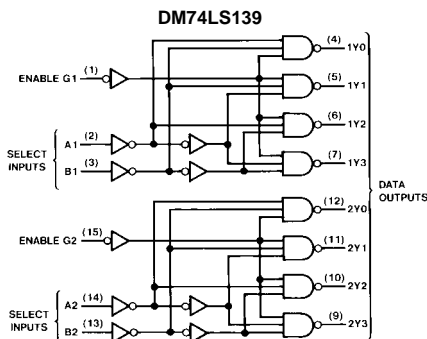
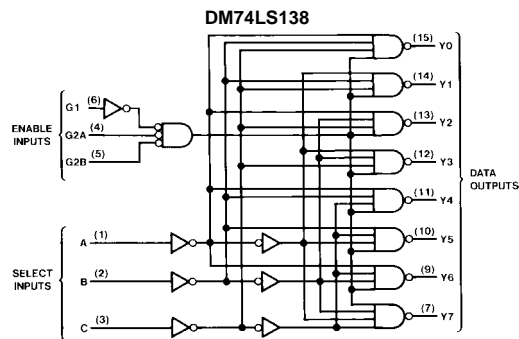
DM74LS139

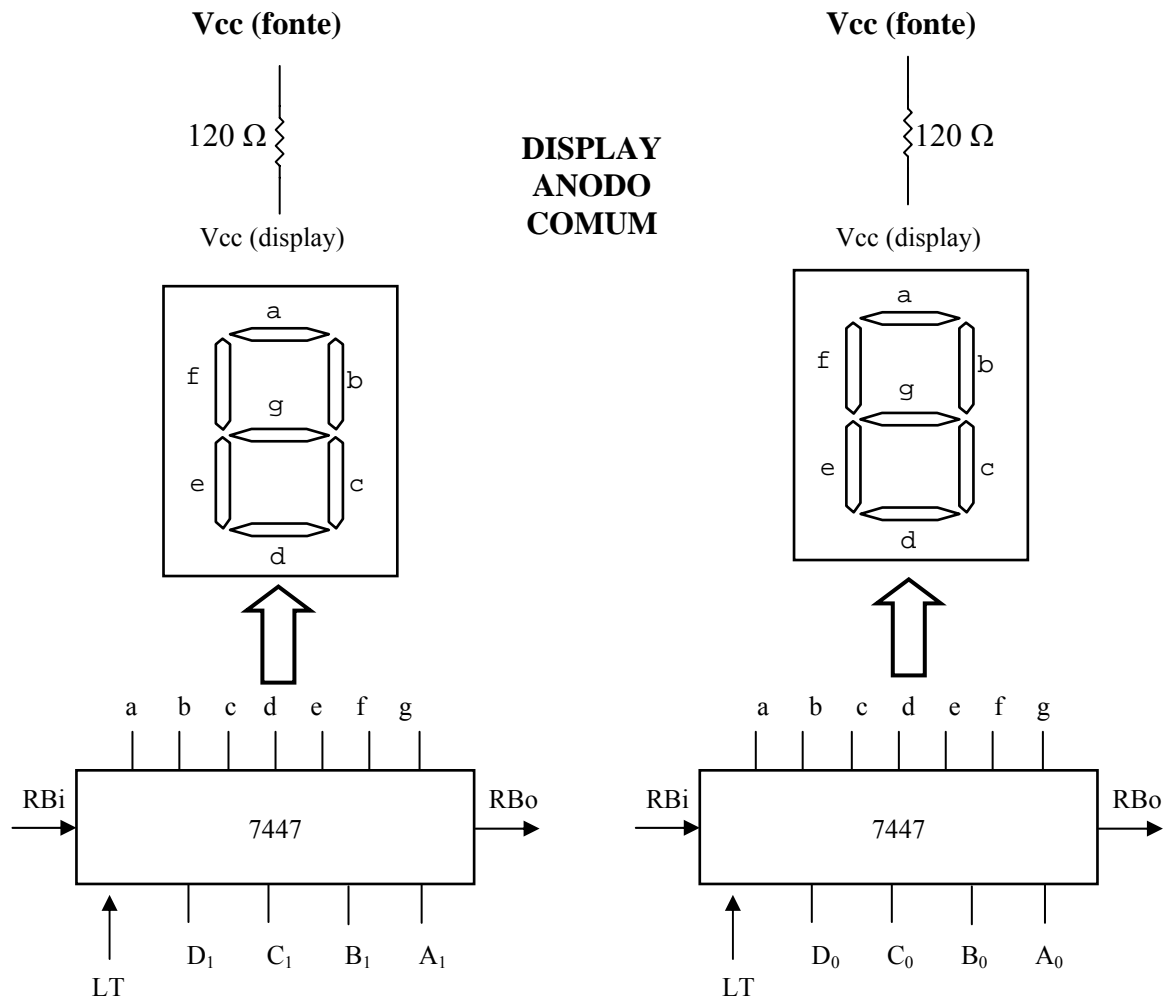
Inputs			Outputs			
Enable	Select					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

H = HIGH Level
L = LOW Level
X = Don't Care

Note 1: G2 = G2A + G2B

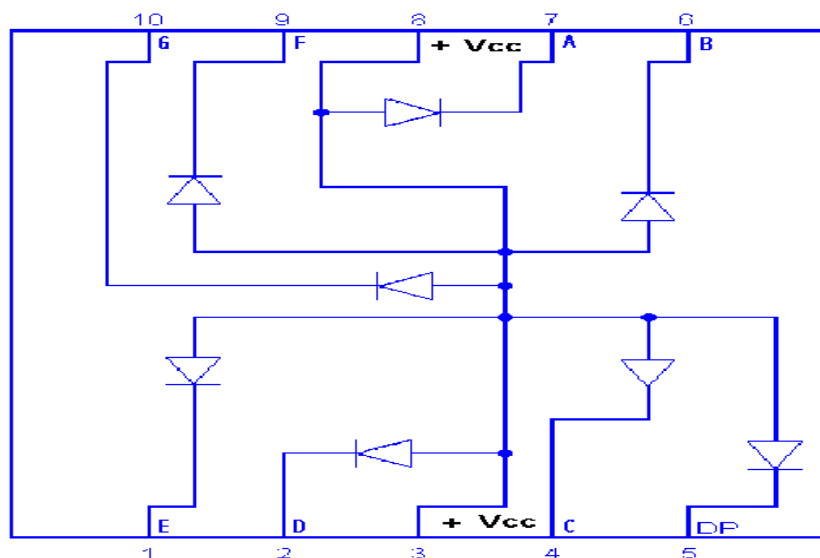
Logic Diagrams





A eliminação dos zeros (0) à esquerda numa conversão de “n” dígitos é realizada através dos pinos RBi e Rbo. Se RBi=0 e a entrada do conversor for 0000 então o display será apagado e a saída Rbo irá à zero (0) para testar o dígito imediatamente inferior.

DISPLAY ANODO COMUM



DM7446A, DM7447A BCD to 7-Segment Decoders/Drivers

General Description

The 46A and 47A feature active-low outputs designed for driving common-anode LEDs or incandescent indicators directly. All of the circuits have full ripple-blanking input/output controls and a lamp test input. Segment identification and resultant displays are shown on a following page. Display patterns for BCD input counts above nine are unique symbols to authenticate input conditions.

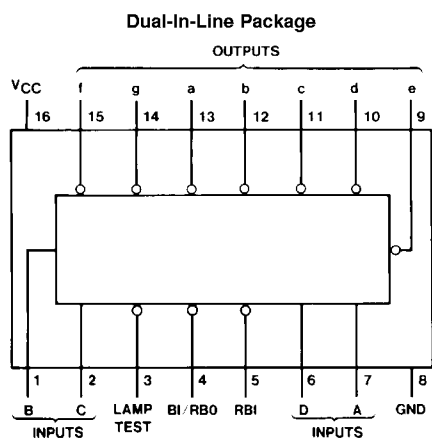
All of the circuits incorporate automatic leading and/or trailing-edge, zero-blanking control (RBI and RBO). Lamp test (LT) of these devices may be performed at any time

when the BI/RBO node is at a high logic level. All types contain an overriding blanking input (BI) which can be used to control the lamp intensity (by pulsing) or to inhibit the outputs.

Features

- All circuit types feature lamp intensity modulation capability
- Open-collector outputs drive indicators directly
- Lamp-test provision
- Leading/trailing zero suppression

Connection Diagram



Order Number DM5447AJ, DM7446AN or DM7447AN
See Package Number J16A or N16E

DM7446A, DM7447A BCD to 7-Segment Decoders/Drivers

'47A Switching Characteristics

at $V_{CC} = 5V$ and $T_A = 25^\circ C$ (See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	Conditions	Min	Max	Units
t_{PLH}	Propagation Delay Time Low to High Level Output	$C_L = 15\text{ pF}$ $R_L = 120\Omega$		100	ns
t_{PHL}	Propagation Delay Time High to Low Level Output			100	ns

Function Table

46A, 47A

Decimal or Function	Inputs						BI/RBO (Note 6)	Outputs								Note
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g		
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H	(Note 7)	
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H		
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L		
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L		
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L		
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L		
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L		
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H		
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L		
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L		
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L		
11	H	X	H	L	H	H	H	H	H	L	L	H	H	L		
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L		
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L		
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L		
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H		
BI	X	X	X	X	X	X	L	H	H	H	H	H	H	H	(Note 8)	
RBI	H	L	L	L	L	L	L	H	H	H	H	H	H	H	(Note 9)	
LT	L	X	X	X	X	X	H	L	L	L	L	L	L	L	(Note 10)	

H = High level, L = Low level, X = Don't Care

Note 6: BI/RBO is a wire-AND logic serving as blanking input (BI) and/or ripple-blanking output (RBO).

Note 7: The blanking input (BI) must be open or held at a high logic level when output functions 0 through 15 are desired. The ripple-blanking input (RBI) must be open or high if blanking of a decimal zero is not desired.

Note 8: When a low logic level is applied directly to the blanking input (BI), all segment outputs are high regardless of the level of any other input.

Note 9: When ripple-blanking input (RBI) and inputs A, B, C, and D are at a low level with the lamp test input high, all segment outputs go H and the ripple-blanking output (RBO) goes to a low level (response condition).

Note 10: When the blanking input/ripple-blanking output (BI/RBO) is open or held high and a low is applied to the lamp-test input, all segment outputs are L.