

## Trabalho Prático N° 4

### Conversor Analógico/Digital tipo “flash” - 1ª Parte

#### 1-Introdução

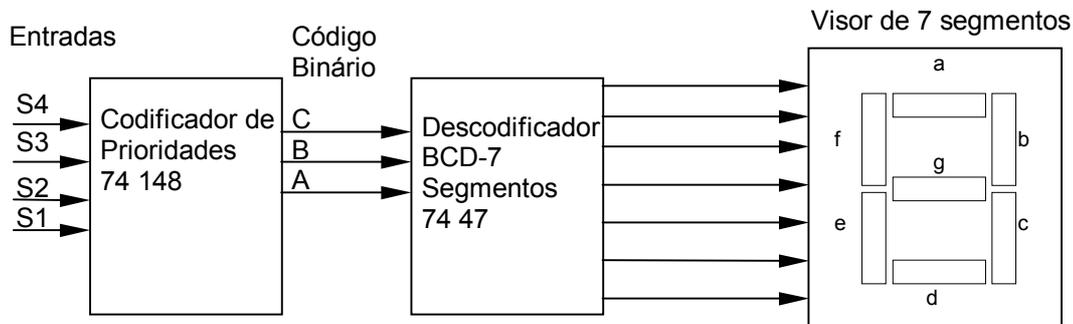
Pretende-se construir um sistema digital que implemente o sistema de conversão de códigos binários em conformidade com a tabela funcional e o diagrama de blocos a seguir apresentados.

#### Tabela Funcional

Entradas				Código Binário	Código 7 Segmentos Visor ânodo comum	
S4	S3	S2	S1	CBA	abcdefg	
0	0	0	0	000	0000001	0
0	0	0	1	001	1001111	1
0	0	1	1	010	0010010	2
0	1	1	1	011	0000110	3
1	1	1	1	100	1001100	4

As restantes combinações possíveis para as entradas S4, S3, S2 e S1 não devem ser consideradas neste sistema.

#### Diagrama de blocos



#### IC 74 148

A primeira fase de codificação deve ser implementada com base no IC 74148. Este IC disponibiliza um codificador de prioridades de 8:3, com entradas e saídas activas a zero (L). Numa primeira análise, este IC não se adapta à implementação da tabela funcional do sistema descrito anteriormente, onde as entradas (S4,S3,S2 e S1) e as saídas (C,B e A) são activas a um. No entanto, um exame detalhado da tabela de verdade do 74 148 mostra que uma nova interpretação da função associada a cada uma das entradas permite a sua aplicação imediata neste sistema.

#### IC 74 47

A descodificação do código binário (CBA) em código de 7 segmentos será implementada com o IC 74 47. Este IC foi especialmente concebido para alimentar directamente visores de 7 segmentos numa configuração de ânodo comum, apresentando as saídas activas a zero e em colector aberto.

#### 2-Execução do trabalho

1)- Analise as especificações técnicas dos IC 74 47 e IC 74 148 e desenhe o diagrama lógico. Deve ter em atenção as entradas e saídas adicionais destes circuitos integrados.

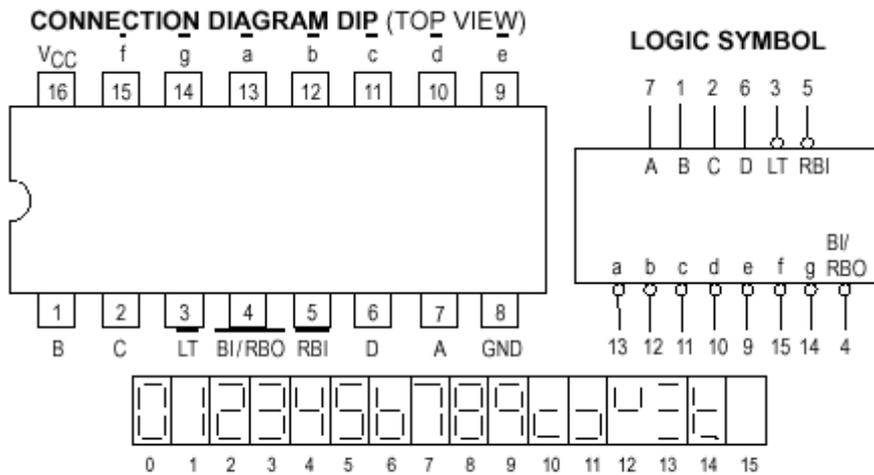
2)- Montagem e teste

Utilize os comutadores para fornecer as entradas (S4,S3,S2 e S1) e visualize o resultado no visor de 7 segmentos de ânodo comum.



7447

Open-collector BCD to 7-segment decoder/common-anode LED driver with ripple blank input and output.

**NUMERICAL DESIGNATIONS — RESULTANT DISPLAYS****TRUTH TABLE**

DECIMAL OR FUNCTION	INPUTS						OUTPUTS							NOTE	
	LT	RBI	D	C	B	A	BI/RBO	a	b	c	d	e	f		g
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H	A
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H	A
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	B
RBI	H	L	L	L	L	L	L	H	H	H	H	H	H	H	C
LT	L	X	X	X	X	X	H	L	L	L	L	L	L	L	D

**NOTES:**

- (A) BI/RBO is wire-AND logic serving as blanking Input (BI) and/or ripple-blanking output (RBO). The blanking out (BI) must be open or held at a HIGH level when output functions 0 through 15 are desired, and ripple-blanking input (RBI) must be open or at a HIGH level if blanking of a decimal 0 is not desired. X = input may be HIGH or LOW.
- (B) When a LOW level is applied to the blanking input (forced condition) all segment outputs go to a LOW level regardless of the state of any other input condition.
- (C) When ripple-blanking input (RBI) and inputs A, B, C, and D are at LOW level, with the lamp test input at HIGH level, all segment outputs go to a HIGH level and the ripple-blanking output (RBO) goes to a LOW level (response condition).
- (D) When the blanking input/ripple-blanking output (BI/RBO) is open or held at a HIGH level, and a LOW level is applied to lamp test input, all segment outputs go to a LOW level.