

# Sistemas Digitais

1º Ano de Engenharia Informática



Trabalho Prático n.º 6

*Codificadores e Descodificadores*

Grupo

Diogo António Costa Medeiros n.º 70633

☐

\_\_\_\_\_ n.º \_\_\_\_\_

☐

\_\_\_\_\_ n.º \_\_\_\_\_

☐

Turma 5

## Objectivos

- Verificar o funcionamento de um decodificador  $n$ -para- $2^n$
- Verificar o funcionamento de um codificador  $2^n$ -para- $n$
- Compreender a noção de prioridade
- Compreender as possibilidades de ligação em cascata de circuitos codificadores e decodificadores.
- Utilizar o decodificador como gerador de mintermos (ou maxtermos).

## Referências

- TAUB, Herbert, “Circuitos Digitais e Microprocessadores”, McGraw-Hill
- Texas Instruments online [<http://www.ti.com/>]

## Material

- Placa RH21
- 74LS138 — 3-TO-8 LINE DECODERS/DEMULTIPLEXERS
- 74LS139A — 2-TO-4 LINE DECODERS/DEMULTIPLEXERS
- $2 \times 74LS148$  — 8-TO-3 LINE PRIORITY ENCODER
- 74LS08 — AND 2 entradas
- 74LS11 — AND 3 entradas
- 74LS32 — OR
- 74LS04 — NOT

# 1. Decodificador $n$ -para- $2^n$

Um decodificador lógico é um dispositivo MSI de entradas múltiplas com capacidade de seleccionar uma única saída com base nos valores de entrada (figura 1).

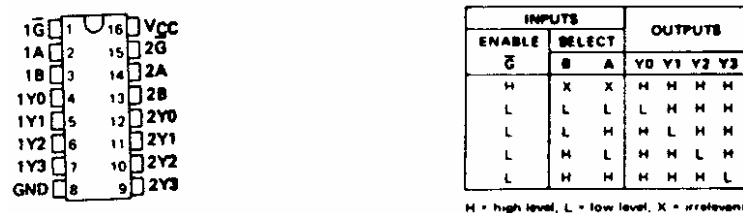


Figura 1. Pin-out e tabela de verdade do 74LS139A

O 74LS139A é um decodificador dual, isto é, existem dois decodificadores independentes em cada integrado de 16 pinos. Cada decodificador tem três entradas ( $A$ ,  $B$  e  $\overline{G}$ ) e quatro saídas ( $\overline{Y0}$ ,  $\overline{Y1}$ ,  $\overline{Y2}$  e  $\overline{Y3}$ ).<sup>1</sup>

<sup>1</sup> As saídas são **activas baixas**, pelo que a notação correcta deverá incluir uma barra, coisa que o fabricante, erradamente, não faz. Neste protocolo usaremos sempre a notação correcta, explicitando as entradas e as saídas activas baixas.

A entrada  $\overline{G}$  é a linha de **enable** do chip. Quando  $\overline{G}$  está no nível lógico 1, o chip está *disabled* (desabilitado, desactivado), e todas as saídas estão no nível lógico 1, independentemente das entradas 1A e 1B. Quando  $\overline{G}$  está no nível lógico 0, o chip está *enabled* (habilitado, activado) e executa a função para a qual foi destinado.

**1.1** Ligue  $1\overline{G}$  a  $V_{CC}$  e meça os valores lógicos em todos os pinos de saída.

$1\overline{Y3} = \underline{\quad 1 \quad}$      $1\overline{Y2} = \underline{\quad 1 \quad}$      $1\overline{Y1} = \underline{\quad 1 \quad}$      $1\overline{Y0} = \underline{\quad 1 \quad}$

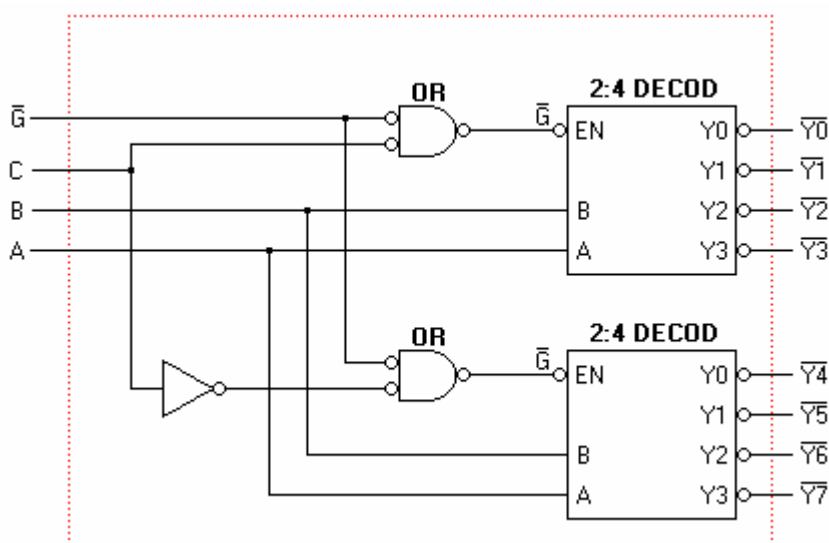
**1.2** Verifique que, com  $1\overline{G}$  ainda ligado a  $V_{CC}$ , as saídas não se alteram ainda que mude os níveis lógicos aplicados a 1A e 1B.

**1.3** Ligue  $1\overline{G}$  a GND e preencha a tabela seguinte. Compare-a com a tabela da figura 1.

1B	1A	$1\overline{Y3}$	$1\overline{Y2}$	$1\overline{Y1}$	$1\overline{Y0}$
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

Por vezes há a necessidade de implementar decodificadores usando como bloco fundamental circuitos decodificadores menores (i. e, com menos bits de entrada e, consequentemente, menos linhas de saída).

**1.4** Monte o circuito da figura ao lado, e preencha a tabela apresentada a seguir.



**Figura 2.** Um decodificador 3:8 implementado com decodificadores 2:4.

Verifique que o circuito resultante se comporta como um decodificador 3:8.

$\overline{G}$	C	B	A	$\overline{Y7}$	$\overline{Y6}$	$\overline{Y5}$	$\overline{Y4}$	$\overline{Y3}$	$\overline{Y2}$	$\overline{Y1}$	$\overline{Y0}$
1	1	0	0	1	1	1	1	1	1	1	1
0	1	0	1	1	1	0	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	0	1	1	1
0	1	0	0	1	1	1	0	1	1	1	1
0	1	1	1	0	1	1	1	1	1	1	1

## 2. Codificador $2^n$ -para- $n$ com prioridades

Um codificador com prioridades é um dispositivo MSI de múltiplas entradas que tem a capacidade de **colocar à saída o valor binário correspondente à entrada activa de maior prioridade**. Um codificador de oito entradas tem assim três saídas, sendo um exemplo disto o 74LS148, cujo circuito lógico, *pin-out* e tabela de verdade se encontra na figura 3.

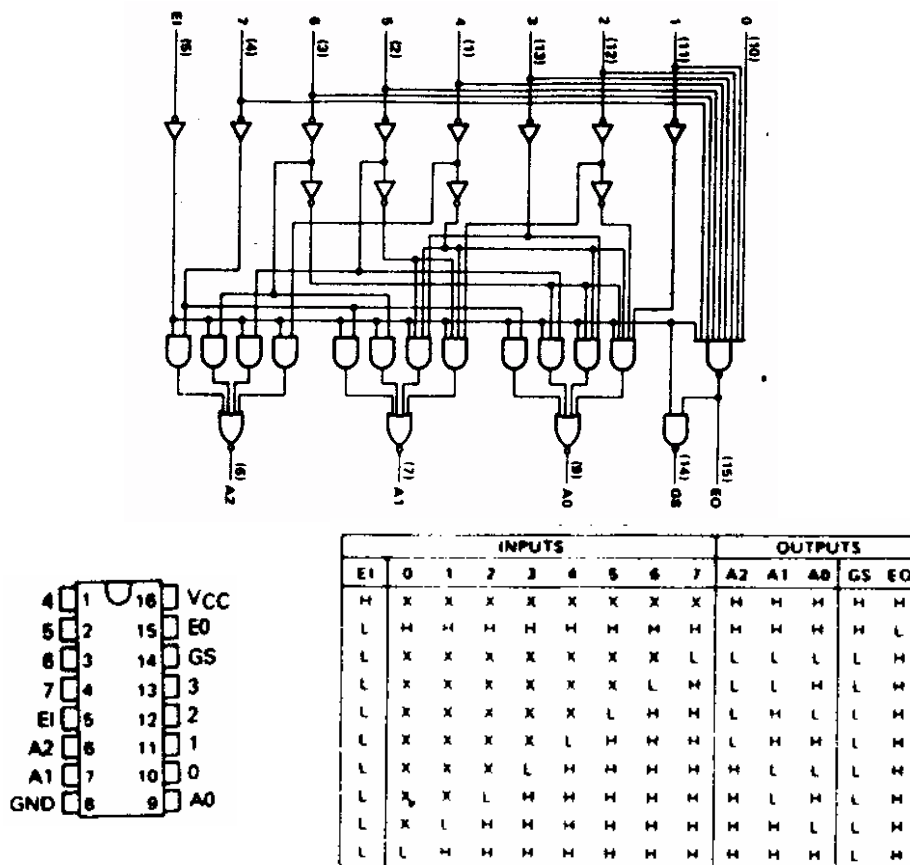


Figura 3. Circuito lógico, *pin-out* e tabela de verdade do 74LS148.

As entradas do chip são  $\bar{0}$ ,  $\bar{1}$ ,  $\bar{2}$ , ...,  $\bar{6}$ ,  $\bar{7}$ , com **prioridade** mais alta para a entrada  $\bar{7}$  e mais baixa para a entrada  $\bar{0}$ . A saída  $\overline{GS}$  é o **sinal de grupo**, isto é,  $\overline{GS}$  fica ao nível baixo se em qualquer instante uma ou mais entradas são activadas, estando o  $\overline{EI}$  (**enable de entrada**) igualmente activado. A saída  $\overline{EO}$  é o **enable de saída** e fica baixo sempre que todas as entradas estejam altas (isto é, desactivadas), estando o  $\overline{EI}$  activado.

### 2.1 “O endereço de saída reflecte a entrada activa com maior prioridade.”

Verifique esta afirmação registando as saídas obtidas quando as entradas são as indicadas na página seguinte (faça sempre  $\overline{EI} = 0$ ).

Entradas								Saídas			Corresponde à entrada:
$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$	$\bar{6}$	$\bar{7}$	$\bar{A2}$	$\bar{A1}$	$\bar{A0}$	
1	1	1	1	0	1	0	0	0	0	0	7
1	1	1	1	0	1	1	0	0	0	0	7
1	1	1	1	0	1	1	1	0	1	1	4
1	1	1	1	0	1	0	1	0	0	1	6

**2.2 Monte o circuito em cascata** como mostrado na figura 4 para formar um **codificador 16-para-4**. Preencha a tabela 1 com as leituras efectuadas.

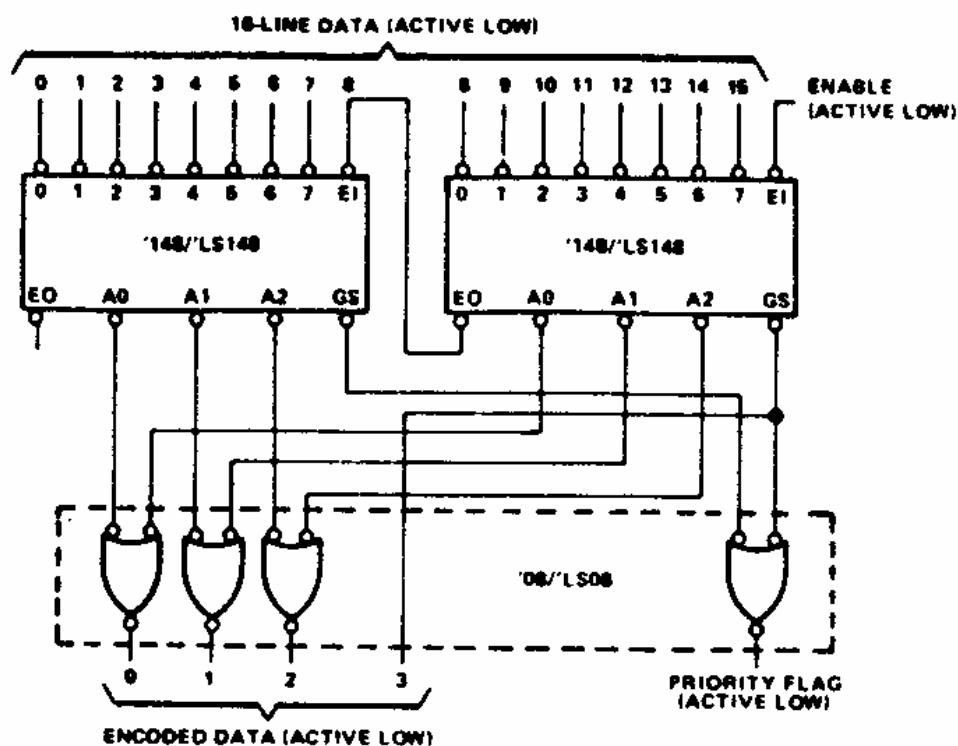


Figura 4. Codificador 16-para-4 com prioridades (dois codificadores 8-para-3 em cascata).

Entradas																	Saídas				
$\overline{En}$	$\overline{15}$	$\overline{14}$	$\overline{13}$	$\overline{12}$	$\overline{11}$	$\overline{10}$	$\overline{9}$	$\overline{8}$	$\overline{7}$	$\overline{6}$	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	$\overline{0}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	$\overline{0}$	$\overline{PF}$
<b>1</b>	1	<b>0</b>	1	1	<b>0</b>	1	1	1	<b>0</b>	1	1	1	1	1	1	<b>0</b>	1	1	1	1	1
<b>0</b>	1	<b>0</b>	1	1	<b>0</b>	1	1	1	<b>0</b>	1	1	1	1	1	1	<b>0</b>	0	0	0	1	0
<b>0</b>	1	1	1	1	<b>0</b>	1	1	1	<b>0</b>	1	1	1	1	1	1	<b>0</b>	0	1	0	0	0
<b>0</b>	1	1	1	1	1	1	1	1	<b>0</b>	1	1	1	1	1	1	<b>0</b>	1	0	0	0	0
<b>0</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	<b>0</b>	1	1	1	1	0
<b>0</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

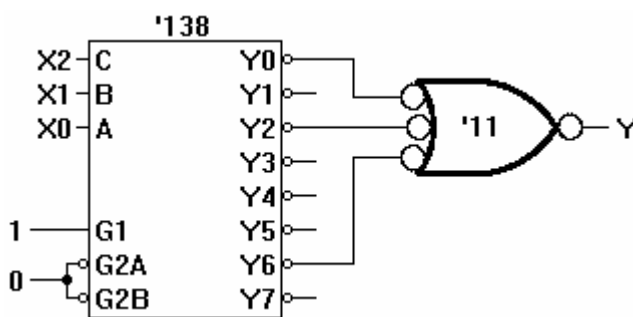
### 3. Decodificador: gerador de mintermos (ou de maxtermos)

Um decodificador binário  $n:2^n$ , ao activar apenas uma das suas linhas de saída de acordo com o código de entrada, pode ser interpretado como um gerador de mintermos ou de maxtermos. Para o primeiro caso, as saídas terão de ser activas-altas, pois nesta situação apenas uma linha de cada vez estará a '1' (o mintermo); a geração de maxtermos é feita com recurso a decodificadores com saídas activas-baixas.

Como qualquer função lógica pode ser expressa em termos de uma soma de mintermos ou um produto de maxtermos, é fácil de concluir que se pode implementar qualquer função recorrendo a um decodificador de dimensão apropriada e a alguma lógica discreta adicional.

**3.1** Implemente o circuito do lado e preencha a seguinte tabela com as leituras efectuadas:

X2	X1	X0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1



**3.2** Escreva a função obtida em termos de soma de mintermos e de produto de maxtermos:

$$Y = \sum m ( \underline{1, 3, 4, 5, 7} )$$

$$Y = \prod M ( \underline{0, 2, 6} )$$

**3.3** Compare as respostas do ponto anterior com o circuito lógico apresentado em 3.1 e conclua.

Nesta situação, o decodificador funciona como um gerador de maxtermos para a função Y, dado que as saídas são ativas baixas e estão ligadas a uma porta AND, ou seja,

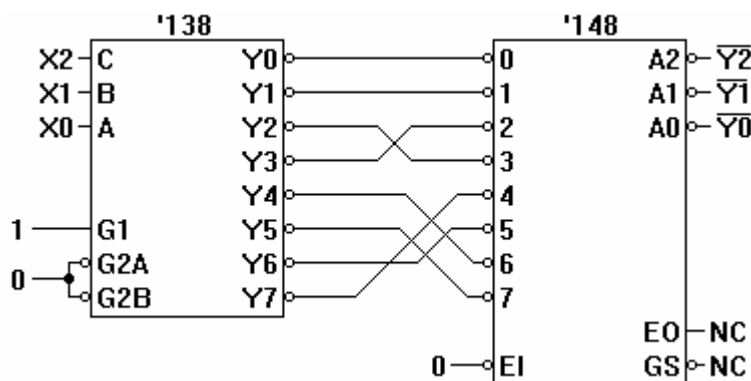
Y será igual ao produto dos M(0,2,6).

## 4. Conversor de códigos binários

Um decodificador binário  $n:2^n$  e um codificador correspondente ( $2^n:n$ ) podem ser conjugados para implementarem um conversor entre códigos binários de  $n$  bits.

**4.1** Implemente o circuito do lado.

**4.2** Preencha a tabela apresentada a seguir, tendo em atenção que as saídas do circuito são activas-baixas e as saídas indicadas na tabela são activas-altas.



X2	X1	X0	Y2	Y1	Y0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

**4.3** Qual a conversão de códigos efectuada por este circuito?

Este circuito converte código binário para  
código Gray. No entanto, dado que as saídas são  
ativas-baixas, o código Gray é apresentado com  
valores complementares.

## Questionário:

1. Construa a **tabela de verdade completa** do circuito implementado no ponto 2.2 do Procedimento e **explique o funcionamento** de tal circuito.
2. Apresente o **diagrama de blocos** de um **decodificador 4:16** implementado usando como bloco fundamental o **decodificador 2:4**.
3. Apresente o circuito lógico que implementa a função  $Y = \sum m(1,2,4,7)$ , usando um **decodificador 3:8** com **saídas activas altas** e a lógica adicional indispensável. Indique uma possível utilização desta função.

## Questionário

### Exercício 1:

Entradas																Saídas					
EN	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	3	2	1	0	PF
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1	1
0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0
0	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	1	0
0	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	1	0	0
0	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	0	0	1	1	0
0	1	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	0	1	0	0	0
0	1	1	1	1	1	0	X	X	X	X	X	X	X	X	X	X	0	1	0	1	0
0	1	1	1	1	1	1	0	X	X	X	X	X	X	X	X	X	0	1	1	0	0
0	1	1	1	1	1	1	1	0	X	X	X	X	X	X	X	X	0	1	1	1	0
0	1	1	1	1	1	1	1	1	0	X	X	X	X	X	X	X	1	0	0	0	0
0	1	1	1	1	1	1	1	1	1	0	X	X	X	X	X	X	1	0	0	1	0
0	1	1	1	1	1	1	1	1	1	1	0	X	X	X	X	X	1	0	1	0	0
0	1	1	1	1	1	1	1	1	1	1	1	0	X	X	X	X	1	0	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0	X	X	X	1	1	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	X	X	1	1	0	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	X	1	1	1	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

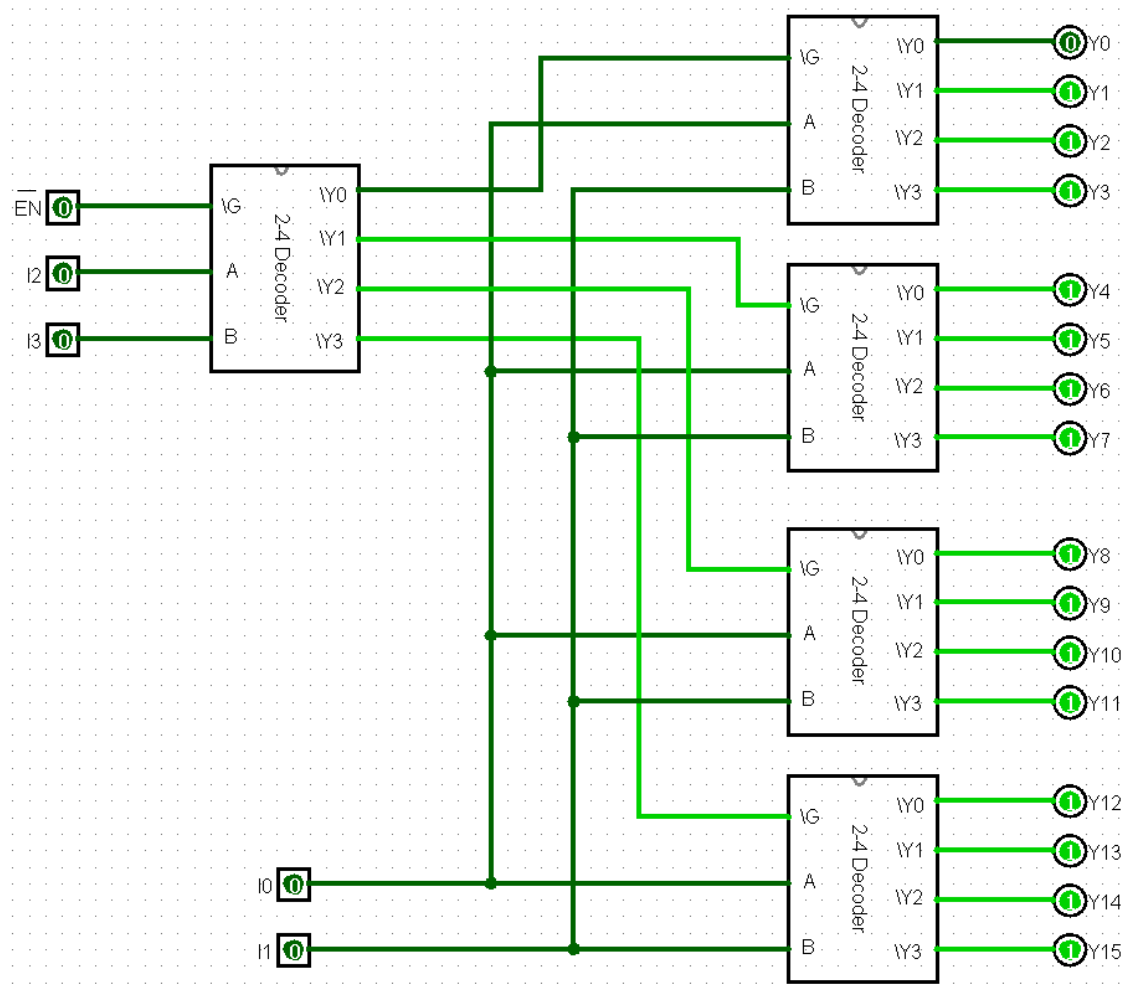
Este circuito é constituído por dois codificadores de prioridades de 8 para 3 (74LS148) com o intuito de funcionarem como um codificador de prioridades de 16 para 4. Caso o  $\overline{EN}$  ( $\overline{EI}$  do segundo codificador) esteja inativo-alto ( $\overline{EN} = 1$ ), quer as saídas  $\overline{3}$  a  $\overline{0}$ , quer a Priority Flag, ficam inativas-altas, independentemente das entradas  $\overline{15}$  a  $\overline{0}$ .

O circuito está em funcionamento quando  $\overline{EN}$  está ativo-baixo ( $\overline{EN} = 0$ ). Nesta situação, o circuito codifica a entrada mais prioritária (n) que se apresente como ativa-baixa, apresentando o correspondente código binário de n à saída. Tendo em conta que as saídas  $\overline{3}$  a  $\overline{0}$  são ativas-baixas, o codificador apresenta nas mesmas os valores complementares. Por exemplo, caso a entrada mais prioritária que se encontre ativa-baixa seja a entrada  $\overline{5}$ , o codificador deverá apresentar à saída o código 1010 (e não 0101).

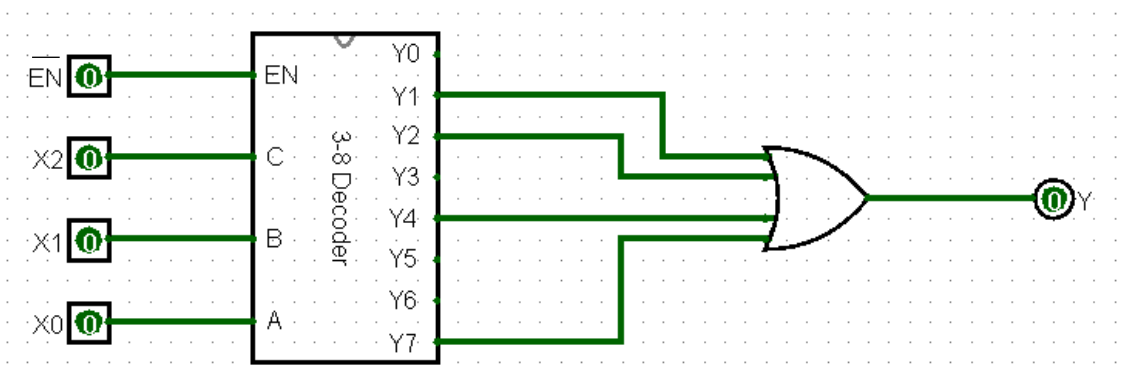
Quanto à Priority Flag, ela apresenta-se como inativa-alta ( $\overline{PF} = 1$ ) em apenas duas situações: o circuito não está em funcionamento ( $\overline{EN} = 1$ ); o circuito está em funcionamento ( $\overline{EN} = 0$ ), no entanto todas as entradas apresentam-se inativas-altas, logo as saídas encontram-se com o mesmo estado lógico. Neste último caso, a  $\overline{PF}$  permite-nos distinguir a situação em que o circuito codifica a entrada  $\overline{0}$  da situação ( $\overline{PF} = 0$ ) em que não codifica nenhuma entrada.



## Exercício 2:



## Exercício 3:



Este circuito encontra-se ativo quando a palavra binária tem um número ímpar de 1s (001, 010, 100, 111), ou seja, tem paridade ímpar. Por esta razão, a função lógica  $Y$  pode ser utilizada para determinar a paridade de uma palavra binária de 3 bits.