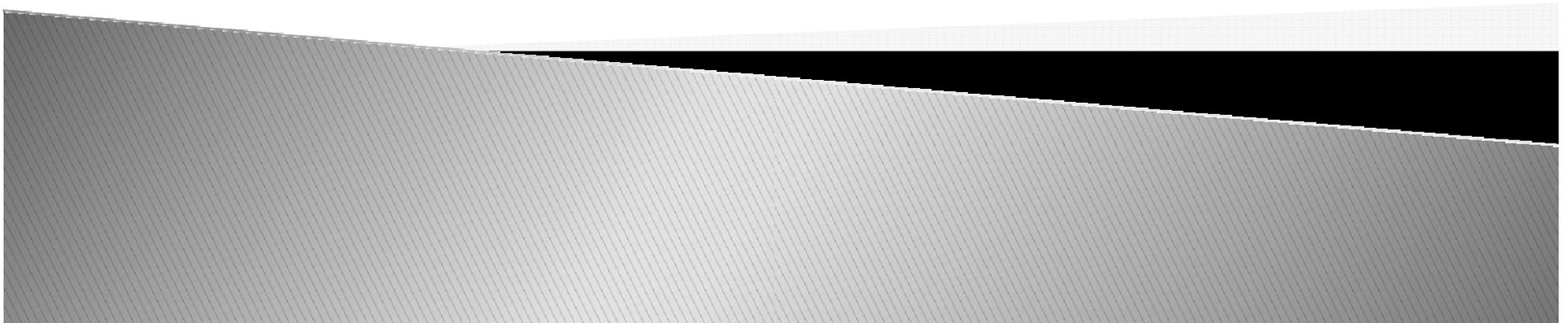


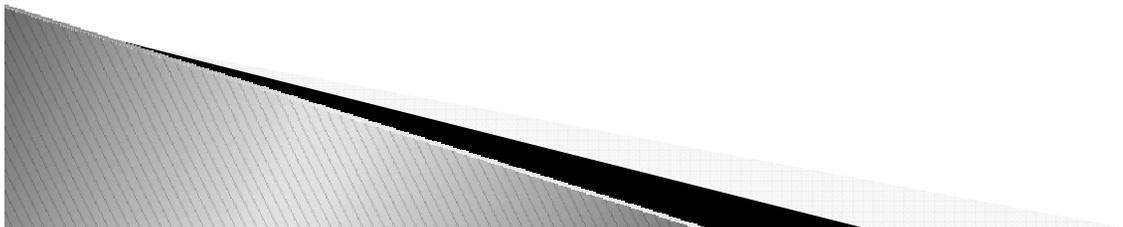
# Codificadores con prioridad

Rubén Ruiz Torres  
Raúl Sánchez San Bartolomé  
Gaspar Fernández Barreras



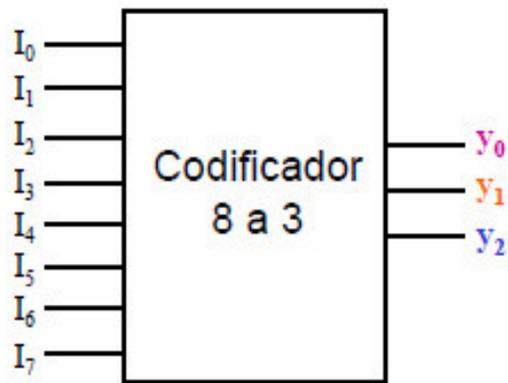
# Introducción a los codificadores

- ▶ Un codificador es un circuito combinacional con  $2^N$  entradas y N salidas, cuya misión es presentar en la salida el código binario correspondiente a la entrada activa.
- ▶ Existen dos tipos fundamentales de codificadores: codificadores sin prioridad y codificadores con prioridad.
- ▶ En el caso de codificadores sin prioridad, puede darse el caso de salidas cuya entrada no pueda ser conocida: por ejemplo, la salida 0 podría indicar que no hay ninguna entrada activada o que se ha activado la entrada número 0.
- ▶ Además, ciertas entradas pueden hacer que en la salida se presente la suma lógica de dichas entradas, ocasionando mayor confusión. Por ello, este tipo de codificadores es usado únicamente cuando el rango de datos de entrada está correctamente acotado y su funcionamiento garantizado.

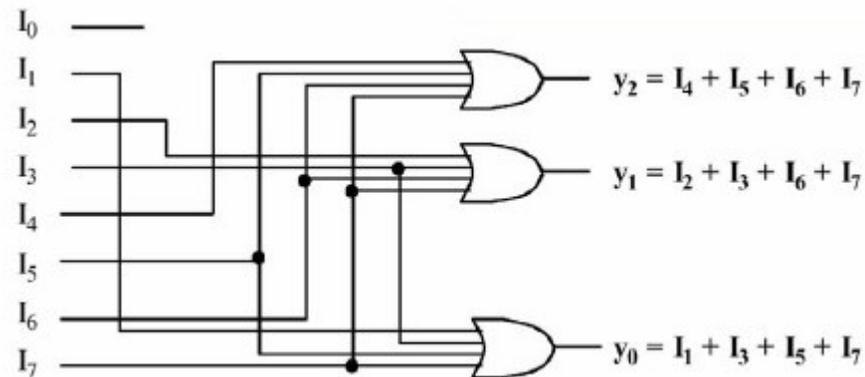


# Codificadores

- ▶ Ejemplo; Codificador 8 a 3 sin prioridad



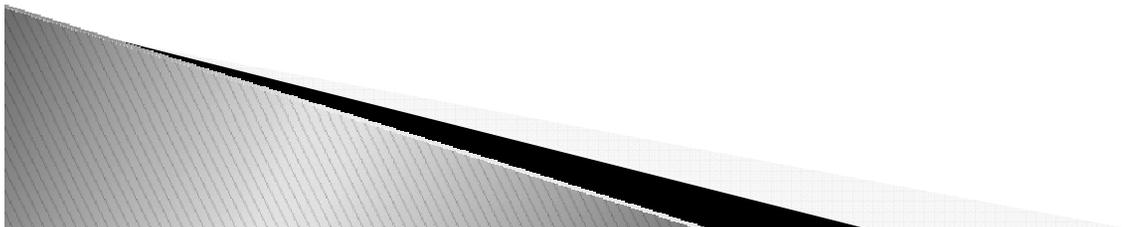
Entradas								Salidas		
$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$	$y_2$	$y_1$	$y_0$
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



• En la tabla de verdad vemos que solo se activa una única entrada simultáneamente

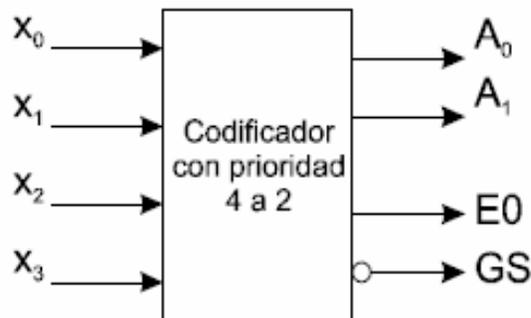
# Codificadores con prioridad

- ▶ Para evitar los problemas anteriormente comentados, se diseñan los codificadores con prioridad. En estos sistemas, cuando existe más de una señal activa, la salida codifica la de mayor prioridad (generalmente correspondiente al valor decimal más alto). Normalmente se establece prioridad alta HPRI o prioridad baja LPRI



# Codificadores con prioridad

- ▶ Para el caso de un codificador 4 a 2 si ninguna línea de entrada está activa, el codificador con prioridad produce  $(A_1A_0) = (00)$ . Si sólo está activa una línea, el codificador produce el valor binario del subíndice de la línea activa. Si está activa más de una entrada, el codificador envía el valor binario del máximo subíndice de las líneas activas.

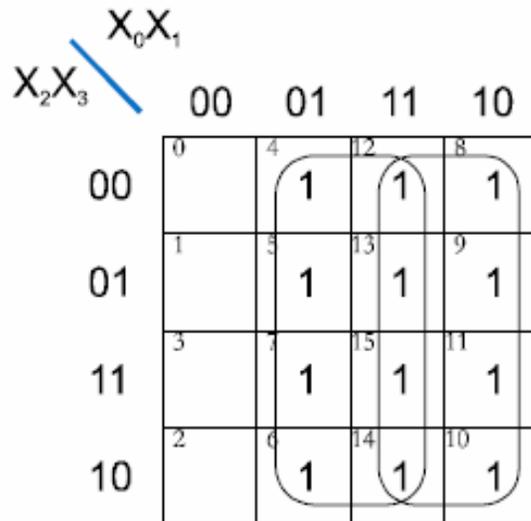


$X_0$	$X_1$	$X_2$	$X_3$	$A_1$	$A_0$	$Z$
X	X	X	1	1	1	1
X	X	1	0	1	0	1
X	1	0	0	0	1	1
1	0	0	0	0	0	1
0	0	0	0	0	0	0

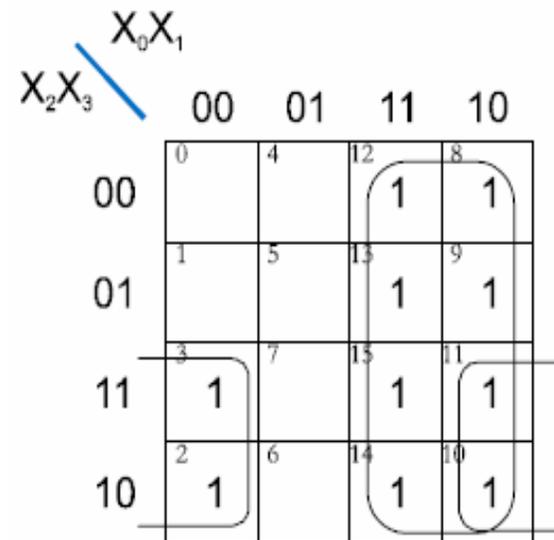
- Tabla de verdad reducida para un codificador con prioridad alta con  $Z=GS$

# Codificadores con prioridad

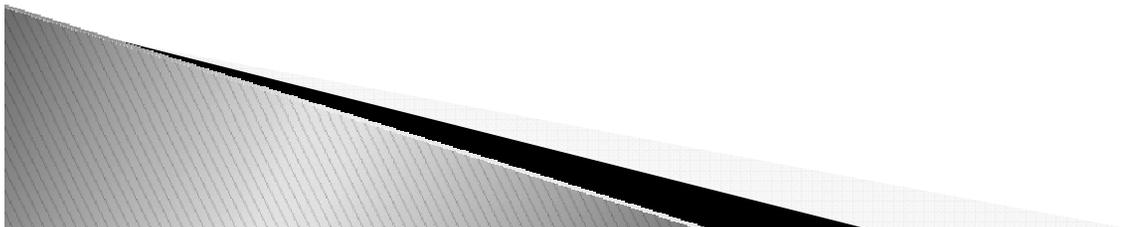
- A partir de la tabla de verdad anterior podemos obtener los mapas de karnaugh para las dos salidas A1 y A0;



A1



A0



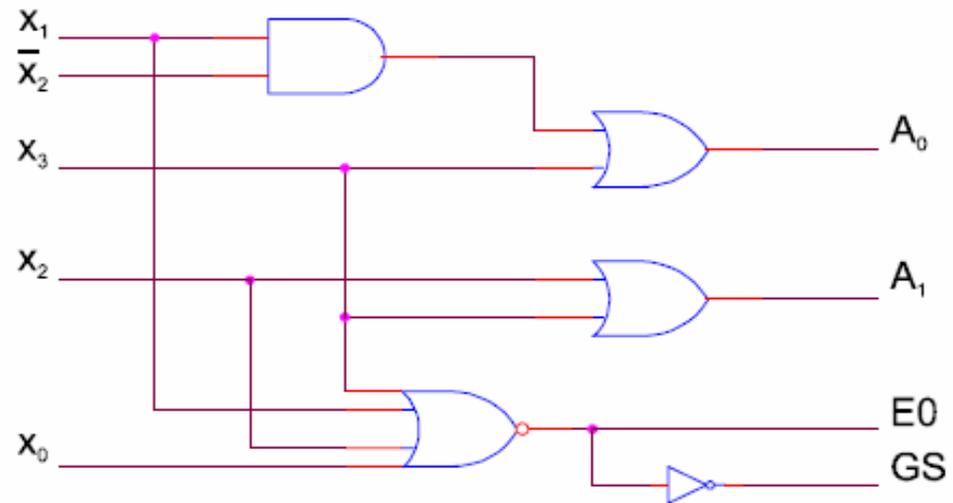
# Codificadores con prioridad

- Las ecuaciones lógicas obtenidas de los mapas de karnaugh y el circuito final implementado con puertas logicas queda finalmente;

$$A_1 = X_2 + X_3$$

$$A_0 = X_3 + X_1 \overline{X_2}$$

$$E0 = \overline{GS} = \overline{X_3 + X_2 + X_1 + X_0}$$



- Las dos líneas de salida adicionales indican que ninguna línea está activa ( $E0 = 1$ ) y que una o más líneas de entrada están activas ( $GS = 1$ ).

# Codificadores con prioridad(VHDL)

- Modelo VHDL de un codificador 4 a 2 con prioridad alta

```
library ieee;
```

```
use ieee.std_logic_1164.all;
```

```
entity hpri4to2 is
```

```
port (I: in std_logic_vector(3 downto 0);
```

```
      A: out std_logic_vector(1 downto 0);
```

```
      Z: in std_logic);
```

```
end hpri4to2;
```

```
architecture comportamiento of hpri4to2 is
```

```
begin
```

```
  A <= "11" when I(3) = '1' else  
      "10" when I(2) = '1' else  
      "01" when I(1) = '1' else  
      "00";
```

```
  Z <= '0' when I = "0000" else '1';
```

```
end comportamiento;
```

- Para definir un codificador con prioridad baja bastaria con cambiar el orden de la sentencia.



# Codificadores con prioridad

## ► La familia 74

**FAIRCHILD**  
SEMICONDUCTOR™

April 1988  
Revised September 1988

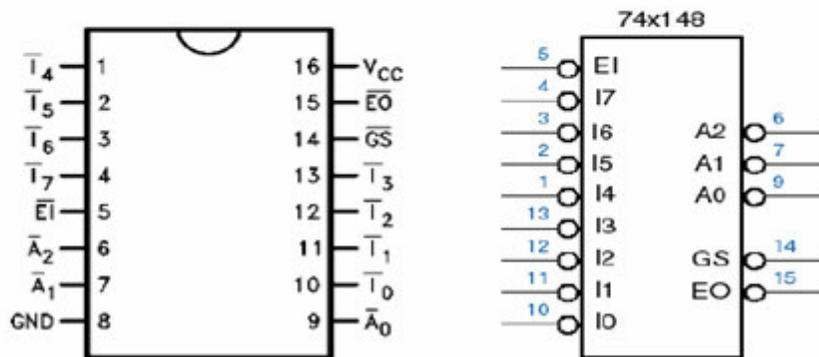
### 74F148

#### 8-Line to 3-Line Priority Encoder

**General Description**  
The F148 provides three bits of binary coded output representing the position of the highest order active input, along with an output indicating the presence of any active input. It is easily expanded via input and output enables to provide priority encoding over many bits.

**Features**

- Encodes eight data lines in priority
- Provides 3-bit binary priority code
- Input enable capability
- Signals when data is present on any input
- Cascadable for priority encoding of n bits

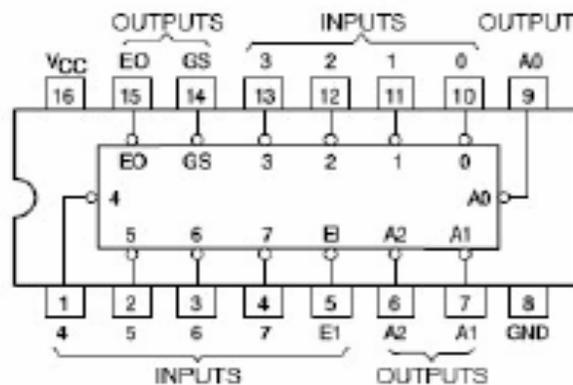


• Hoja de características y tabla de verdad del 74F148

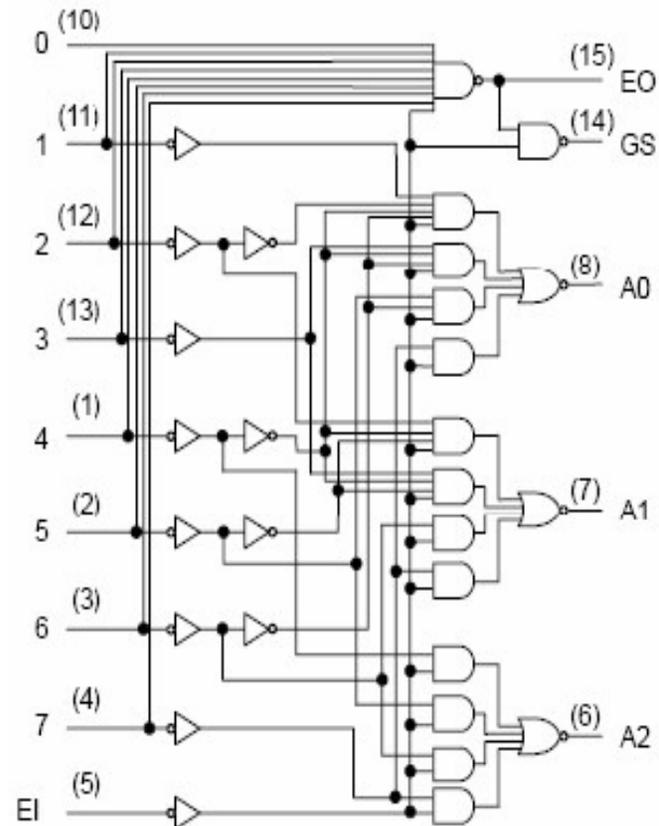
Inputs								Outputs					
$\overline{EI}$	$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$	$\overline{GS}$	$A_0$	$A_1$	$A_2$	$\overline{EO}$
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	L	L	L	H	H
L	X	X	L	H	H	H	H	H	L	H	L	H	H
L	X	L	H	H	H	H	H	H	L	L	H	H	H
L	L	H	H	H	H	H	H	H	L	H	H	H	H

H = HIGH Voltage Level  
L = LOW Voltage Level  
X = Immaterial

# Codificadores con prioridad

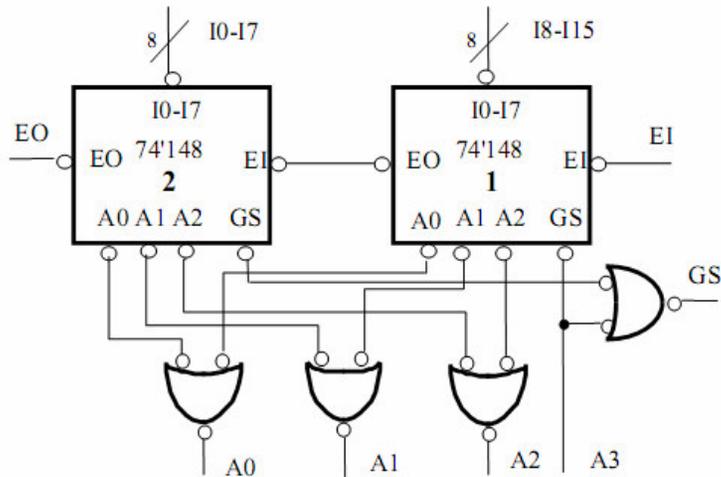


- Patillaje y diagrama circuital del 74F148

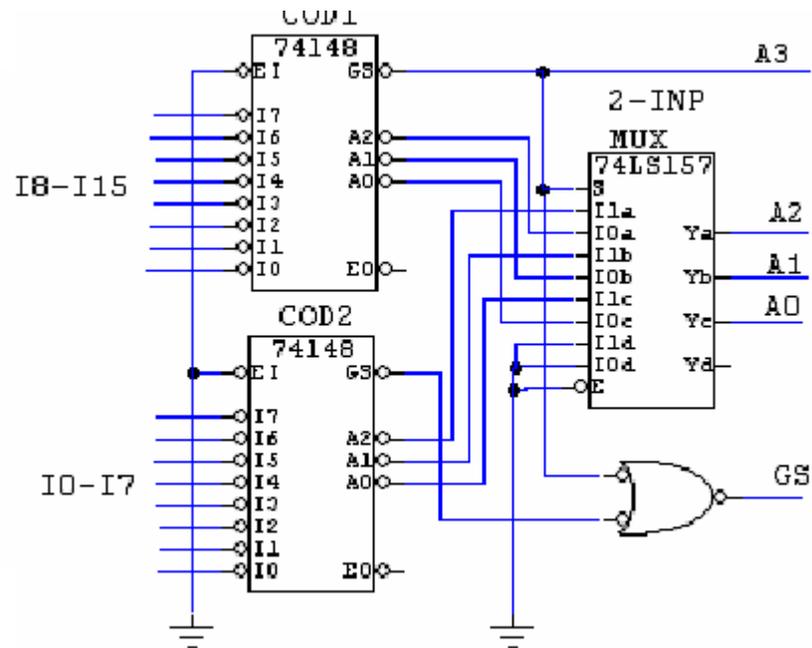


# Codificadores más grandes en base a codificadores más pequeños

- Codificadores 16 a 4 en base codificadores 4 a 3



Cascada "Ripple"



Paralelo

# Problemas Propuestos

- ▶ **1)** Encontrar las ecuaciones lógicas que permiten definir un circuito codificador con prioridad baja de 8 bits de entrada ( $I_7-I_0$ ) y salidas en código gray (de más a menos significativas:  $A B C$ ). Factorizar en lo posible las ecuaciones lógicas suponiendo todas las entradas y salidas asertadas bajas.
- ▶ **2)** Construir un circuito codificador binario de 8 a 3 con prioridad baja tomando como base el circuito codificador 74LS148, y el menor número posible de puertas lógicas que sean necesarias. Se permite definir como mejor convenga la polaridad de las entradas y de las salidas.
- ▶ **3)** Se quiere realizar un circuito de 8 entradas ( $I_8-I_1$ ) y 8 salidas ( $O_8-O_1$ ), tal que la salida muestra la entrada pero eliminando todos los unos menos el más significativo. Por ejemplo, si  $I = "01101101"$ ,  $O = "01000000"$ ; si  $I = "00010110"$ ,  $O = "00010000"$ , etc. Si todos los bits de la entrada son 0, los de la salida también:  $I = "00000000"$ ,  $O = "00000000"$ .  
Implementar el circuito con un circuito codificador 8 a 3 74LS148 y un circuito decodificador 3 a 8 74LS138. Suponer las entradas y salidas  $I_{8.L}, \dots I_{1.L}$ ;  $O_{8.L}, \dots O_{1.L}$  en polaridad negativa.

