

EL-611

Multiplexores y Decodificadores



Multiplexores (Mux) o Selector

Un Mux de orden p ; $M(p)$, puede realizar una función:

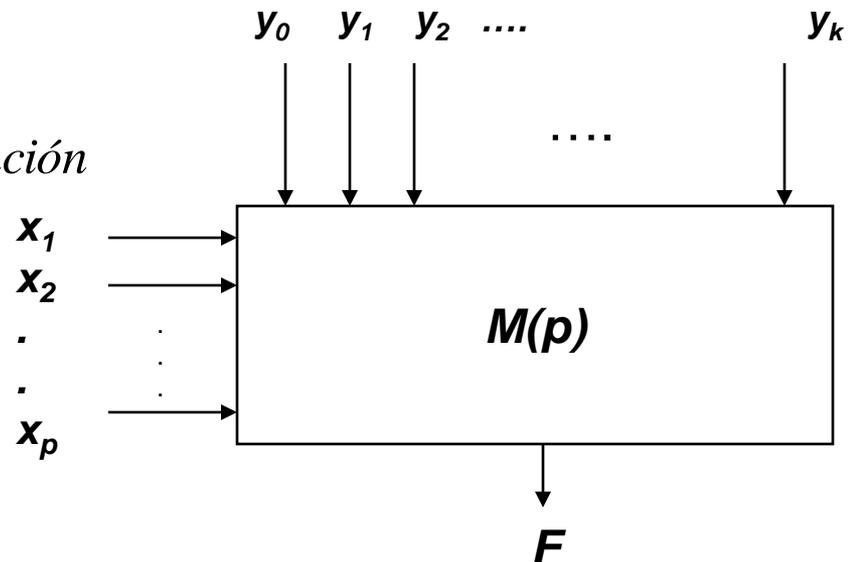
$$F(x_1, x_2, \dots, x_p, y_0, y_1, \dots, y_k) = \sum_{i=0}^{2^p-1} (\dot{x}_1 \dot{x}_2 \dot{x}_3 \dots \dot{x}_p)_i y_i$$

donde:

$$k = 2^p - 1;$$

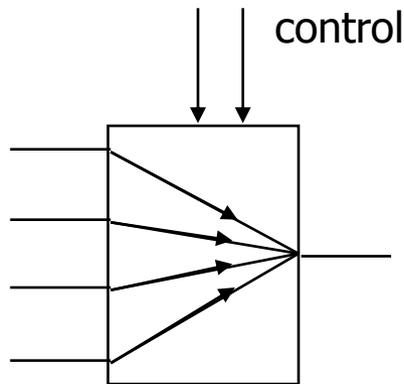
$$\dot{x}_j \in \{x_j, \bar{x}_j\};$$

$(\dot{x}_1 \dot{x}_2 \dot{x}_3 \dots \dot{x}_p)_i \rightarrow$ *minitérmino i de la función*

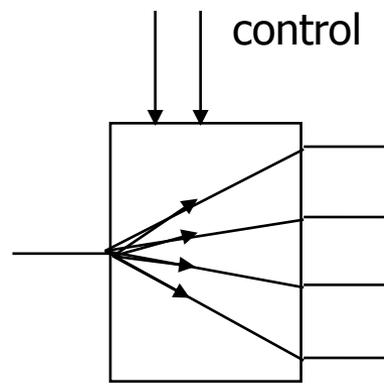


Hacer conexiones

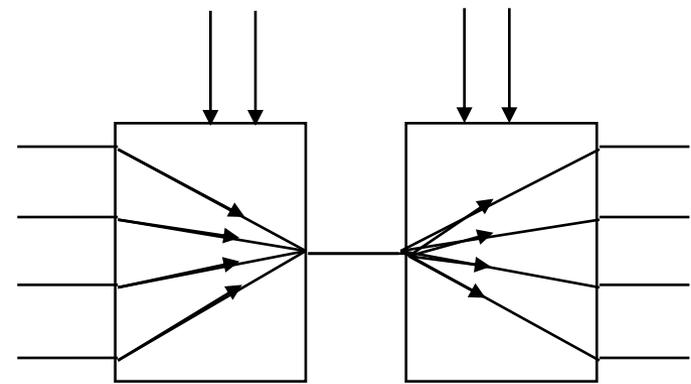
- Conexiones directas punto-a-punto entre compuertas
- Enrutar una o muchas entradas a una sola salida ---
Multiplexor
- Enrutar una sola entrada a una o muchas salidas ---
Demultiplexor



multiplexor



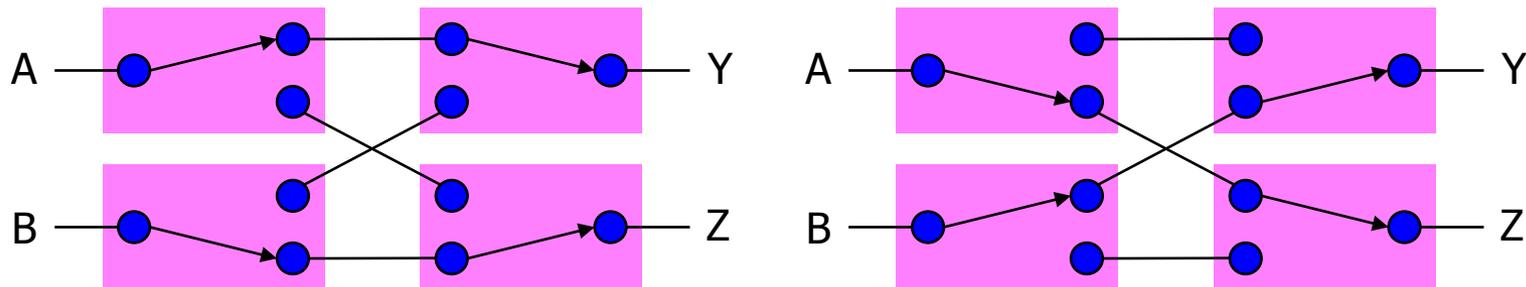
demultiplexor



4x4 switch

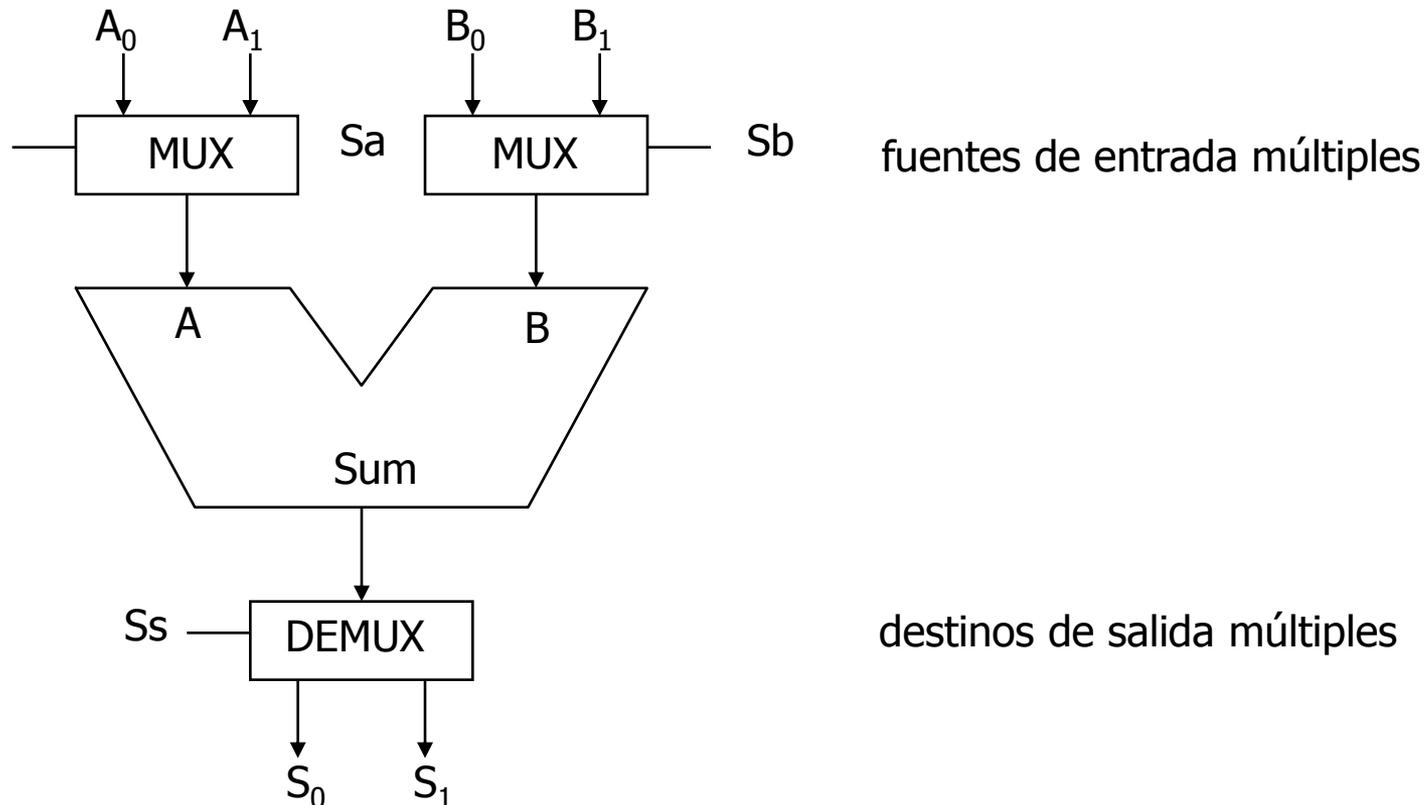
Mux y Demux

- Implementación en Switch de multiplexores y demultiplexores
 - pueden ser conformados para realizar redes de switching de tamaño arbitrario
 - Utilizado para implementar interconexiones múltiples fuentes/múltiples destinos



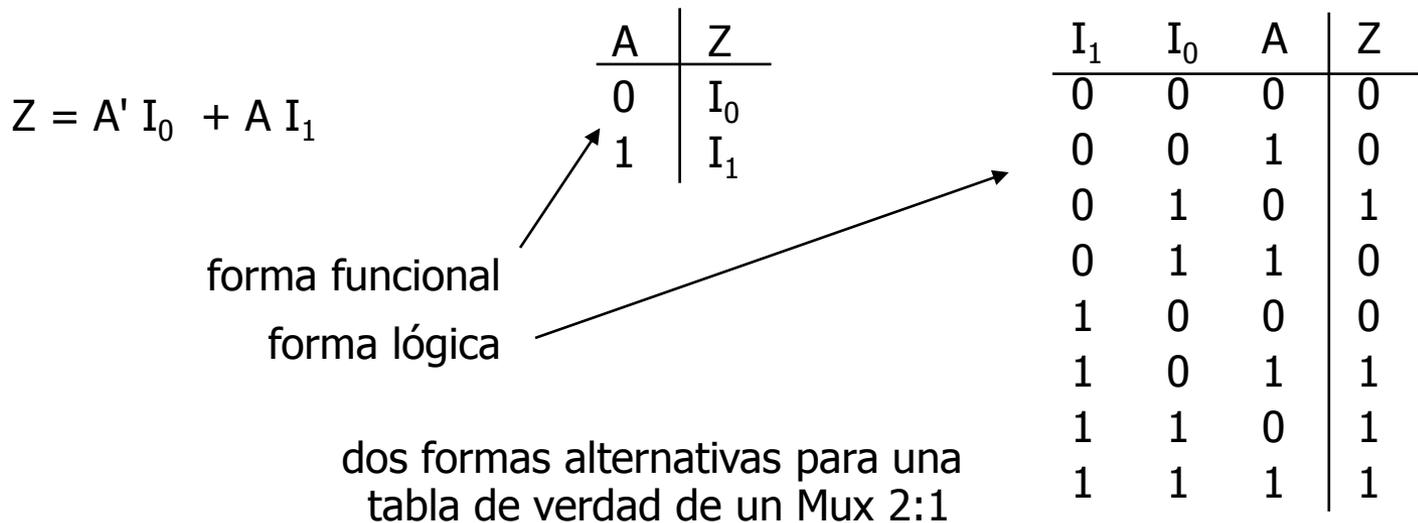
Mux and Demux (cont.)

- Uso de multiplexores/demultiplexores en conexiones multi-punto



Multiplexores/selectores

- Multiplexores/selectores: conceptos generales
 - 2^n entrada de datos, n entradas de control (llamados "selectores"), 1 salida
 - utilizado para conectar 2^n puntos a un sólo punto
 - patrones de señales de control forman índices binarios de entrada conectados a la salida

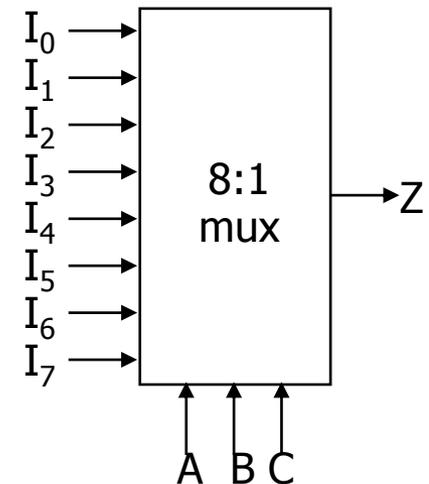
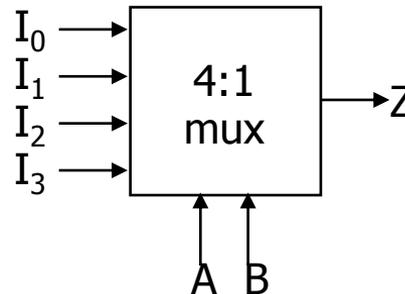
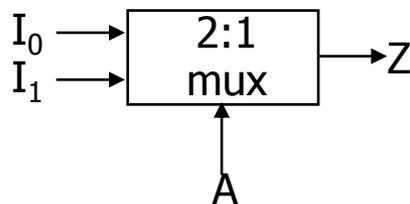


Multiplexores/selectores (cont.)

- 2:1 mux: $Z = \bar{A}I_0 + AI_1$
- 4:1 mux: $Z = \bar{A}\bar{B}I_0 + \bar{A}BI_1 + A\bar{B}I_2 + ABI_3$
- 8:1 mux: $Z = \bar{A}\bar{B}\bar{C}I_0 + \bar{A}\bar{B}CI_1 + \bar{A}BCI_2 + A\bar{B}\bar{C}I_3 + A\bar{B}CI_4 + AB\bar{C}I_5 + ABCI_6 + ABCI_7$

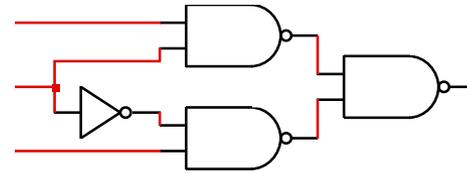
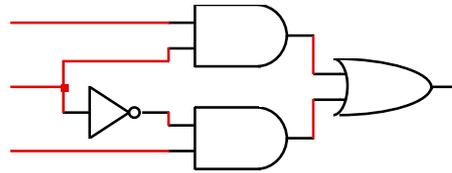
➤ En general:
$$Z = \sum_{k=0}^{2^n-1} (m_k I_k)$$

– en forma de minterminos
para un Mux $2^n:1$

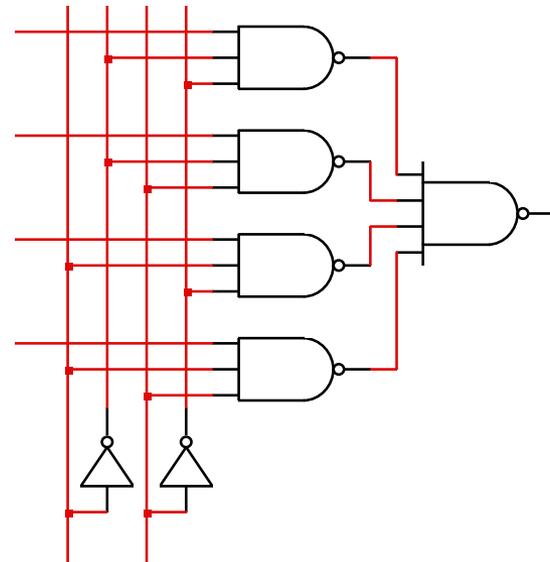
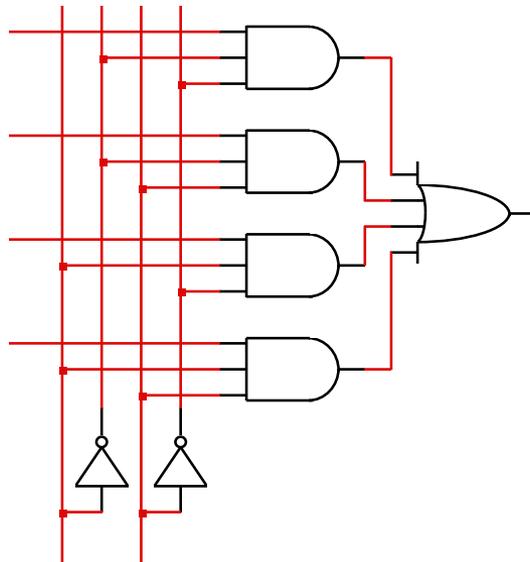


Implementación de Muxes a nivel de Compuertas

➤ 2:1 mux

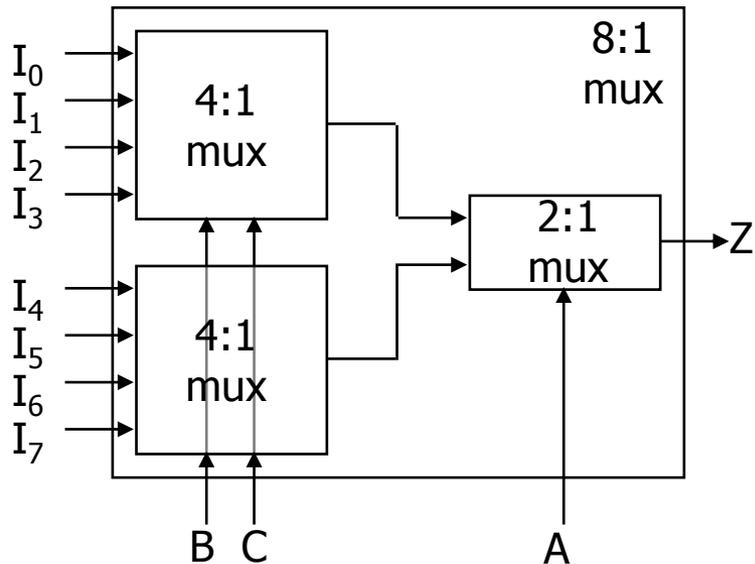


➤ 4:1 mux



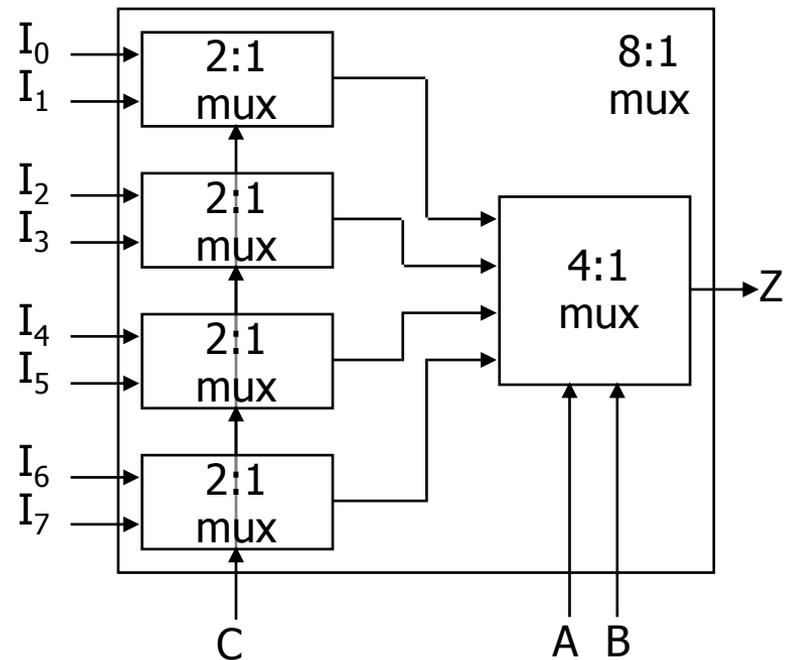
Multiplexores en Cascada

- Multiplexores más grandes pueden realizarse poniendo en cascada Mux's más pequeños



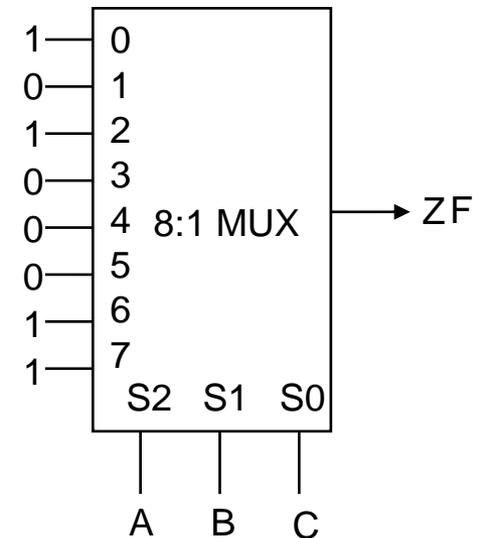
líneas de control B y C seleccionan simultáneamente una de entradas I_0, I_1, I_2, I_3 y una de I_4, I_5, I_6, I_7
línea de control A selecciona cual de los mux's sale como Z

implementación alternativa



Multiplexores como lógica de propósitos generales

- Un multiplexor $2^n:1$ puede implementar cualquier función de n variables
 - con las variables utilizadas como entradas de control, y
 - las entradas de datos conectadas a 0 ó 1
 - lo anterior porque $\{M(n), 0, 1\}$ es “funcionalmente completo”
 - en esencia, un “lookup table”



- Ejemplo: $F(A, B, C) = m_0 + m_2 + m_6 + m_7$

$$= \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$= \bar{A}\bar{B}\bar{C}(1) + \bar{A}\bar{B}\bar{C}(0) + \bar{A}B\bar{C}(1) + \bar{A}B\bar{C}(0) + A\bar{B}\bar{C}(0) + A\bar{B}\bar{C}(0) + ABC(1) + ABC(1)$$

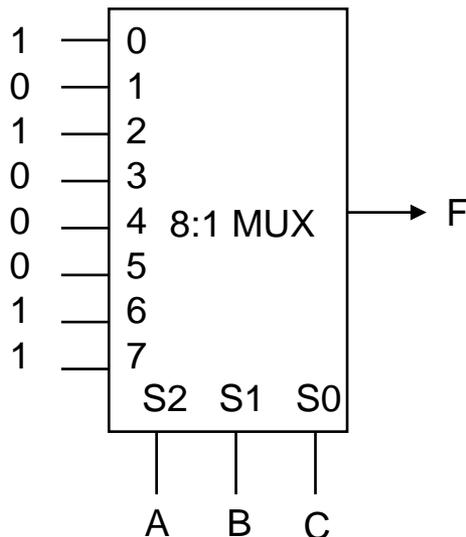
$$Z = \bar{A}\bar{B}\bar{C}I_0 + \bar{A}\bar{B}CI_1 + \bar{A}B\bar{C}I_2 + \bar{A}BCI_3 + A\bar{B}\bar{C}I_4 + A\bar{B}CI_5 + ABCI_6 + ABCI_7$$

Multiplexores como lógica de propósitos generales (cont.)

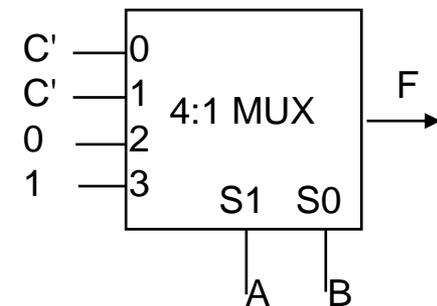
- Un multiplexor $2^{n-1}:1$ puede implementar cualquier función de n variables
 - con $n-1$ variables utilizadas como entradas de control, y
 - entradas de datos conectadas a la última variable o su complemento
- Ejemplo: $F(A, B, C) = m_0 + m_2 + m_6 + m_7$

$$= \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + AB\bar{C} + ABC$$

$$= \bar{A}\bar{B}(\bar{C}) + \bar{A}B(\bar{C}) + AB(0) + AB(1)$$



A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

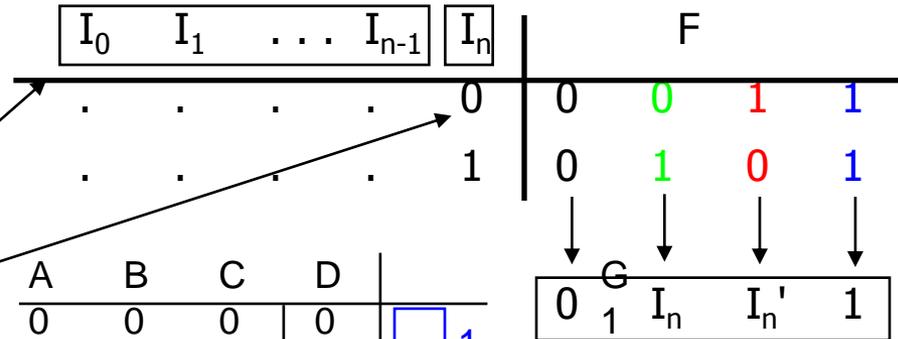


Multiplexores como lógica de propósitos generales (cont.)

➤ Generalización

n-1 variables de control mux

variable mux única de dato



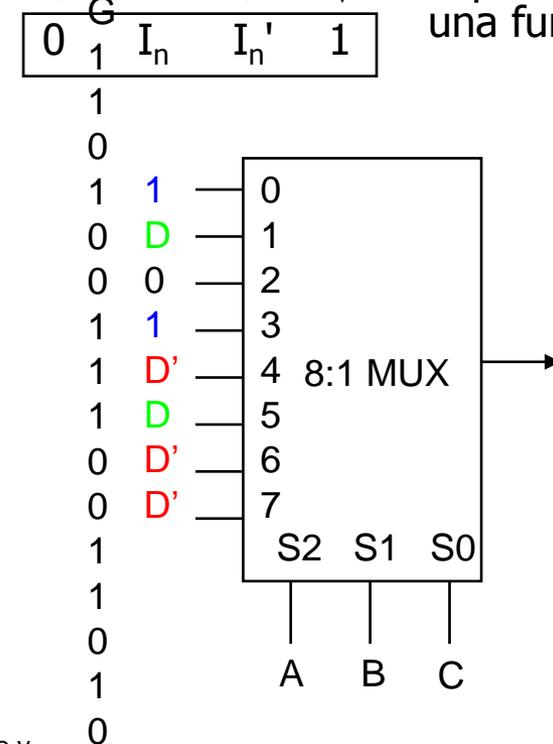
cuatro posibles configuraciones de la tabla de verdad
filas pueden ser expresadas como una función de I_n

➤ Ejemplo:

$G(A,B,C,D)$

puede ser realizada con un MUX 8:1

seleccionar A,B,C como variables de control



APLICACIONES DEL MEV EN EL DISEÑO CON MUX'S

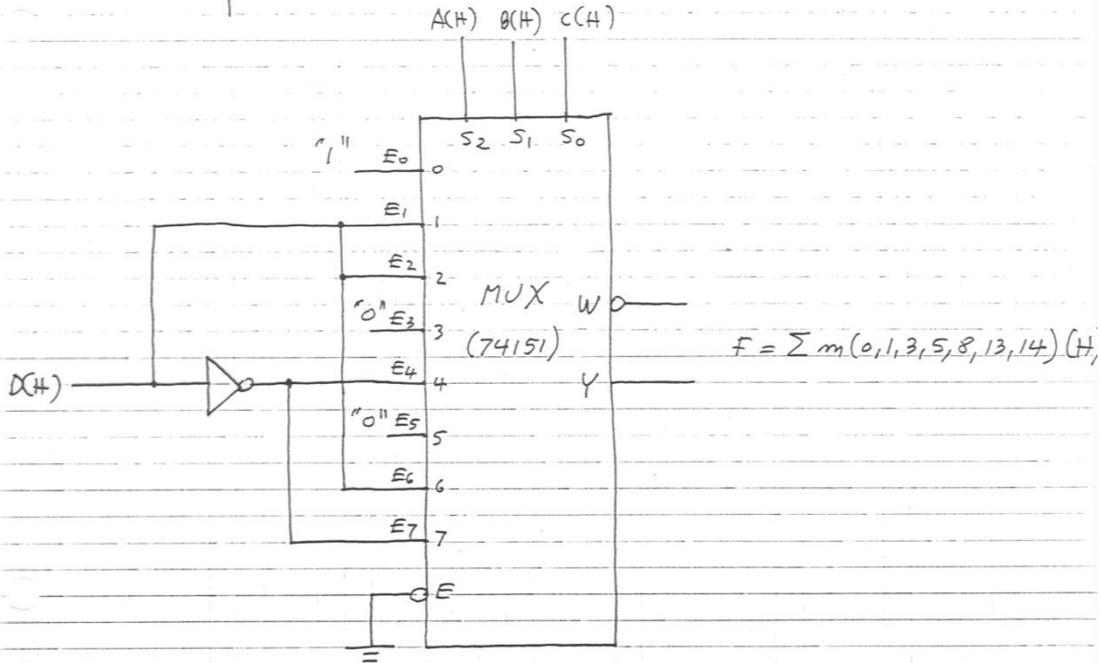
EJEMPLO :

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

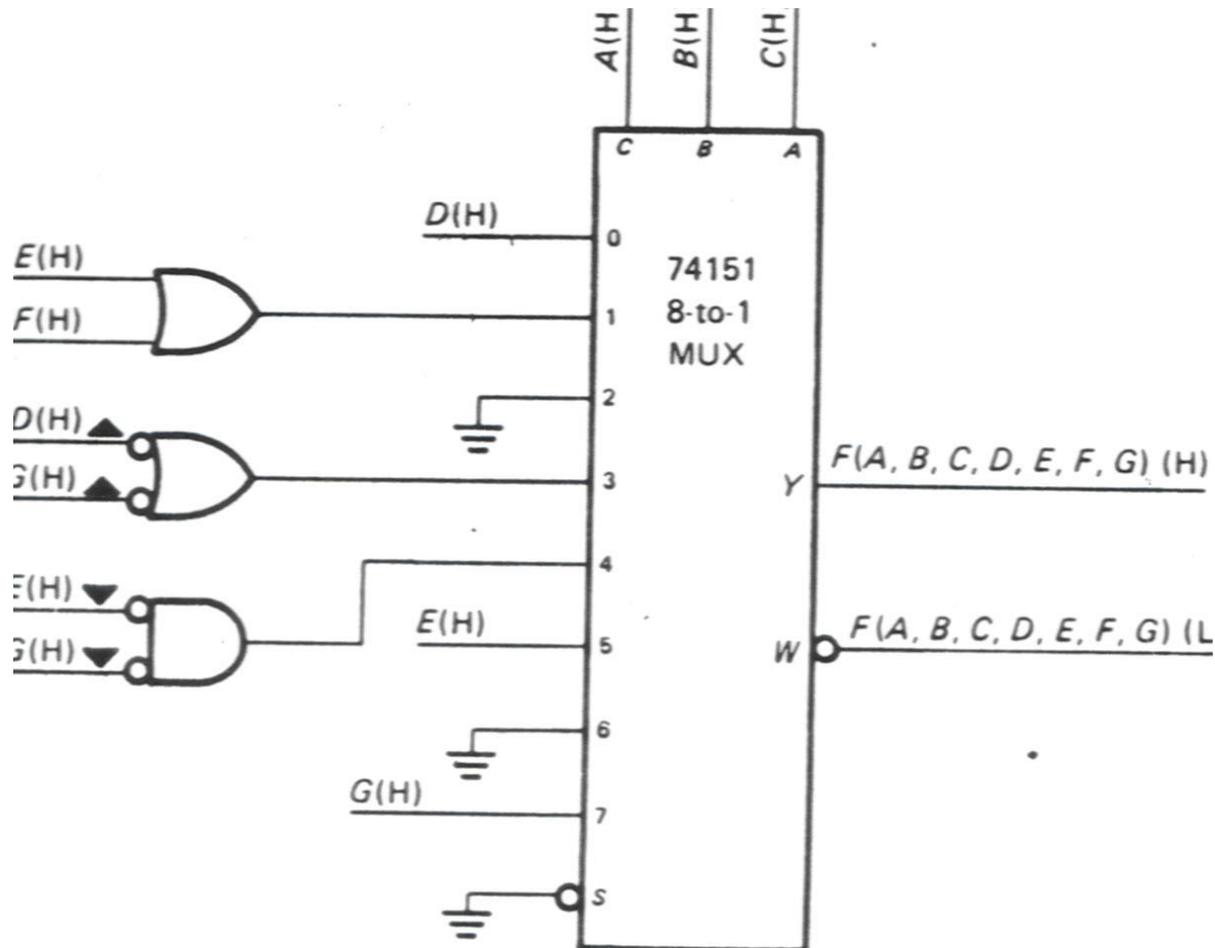
$A = S_2$
 $B = S_1$
 $C = S_0$

S_2	S_1	S_0	00	01	11	10
0	0	0	F_0	F_2	F_6	F_4
0	1	0	F_1	F_3	F_7	F_5

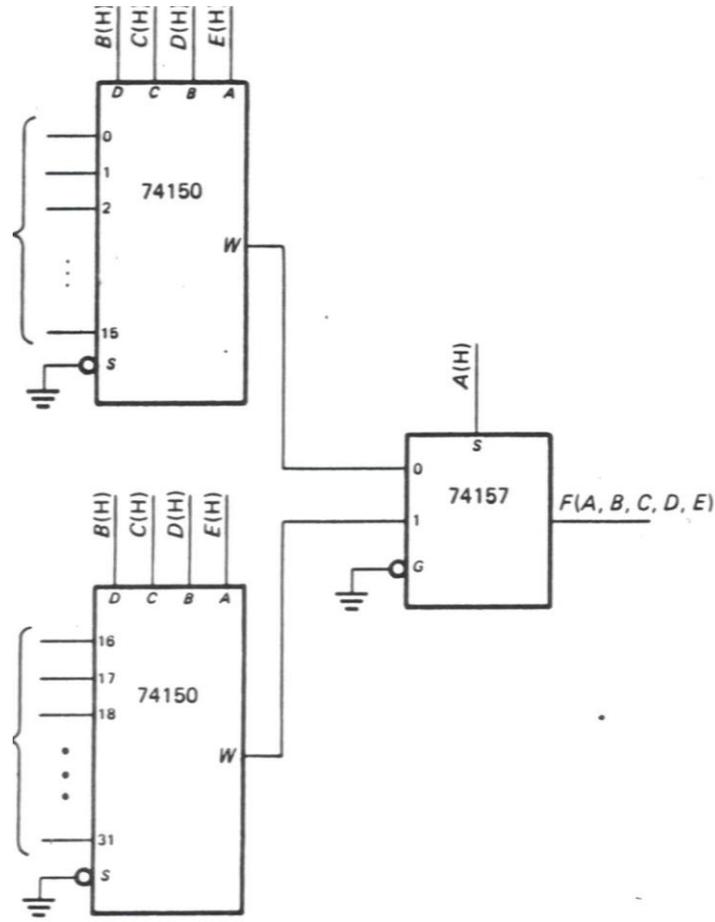
S_1	S_2	S_0	00	01	11	10
0	0	0	1	D	D	\bar{D}
0	1	0	D	0	\bar{D}	0



Uso de un Multiplexor



Estructuras con Multiplexores (otro ejemplo)



Demultiplexores/decodificadores (“decoder”)

- Decodificadores/demultiplexores: concepto general
 - una entrada de dato, n entradas de control, 2^n salidas
 - entradas de control (llamadas “selectores” (S)) representan índices binarios de salidas a las cuales las entradas son conectadas
 - entrada de dato normalmente llamada “enable” (G)

1:2 Decoder:

$$O0 = G \cdot S'$$

$$O1 = G \cdot S$$

2:4 Decoder:

$$O0 = G \cdot S1' \cdot S0'$$

$$O1 = G \cdot S1' \cdot S0$$

$$O2 = G \cdot S1 \cdot S0'$$

$$O3 = G \cdot S1 \cdot S0$$

3:8 Decoder:

$$O0 = G \cdot S2' \cdot S1' \cdot S0'$$

$$O1 = G \cdot S2' \cdot S1' \cdot S0$$

$$O2 = G \cdot S2' \cdot S1 \cdot S0'$$

$$O3 = G \cdot S2' \cdot S1 \cdot S0$$

$$O4 = G \cdot S2 \cdot S1' \cdot S0'$$

$$O5 = G \cdot S2 \cdot S1' \cdot S0$$

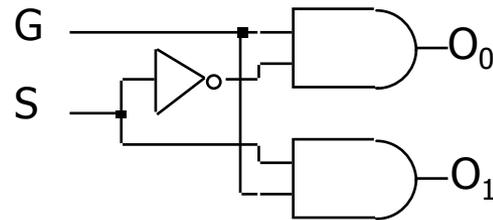
$$O6 = G \cdot S2 \cdot S1 \cdot S0'$$

$$O7 = G \cdot S2 \cdot S1 \cdot S0$$

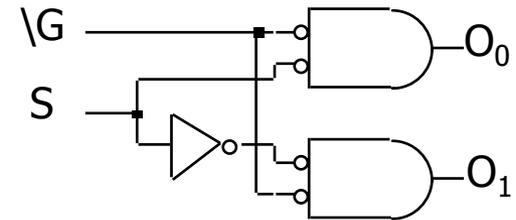
Implementación de Demultiplexores a nivel de Compuerta

➤ 1:2 decoders

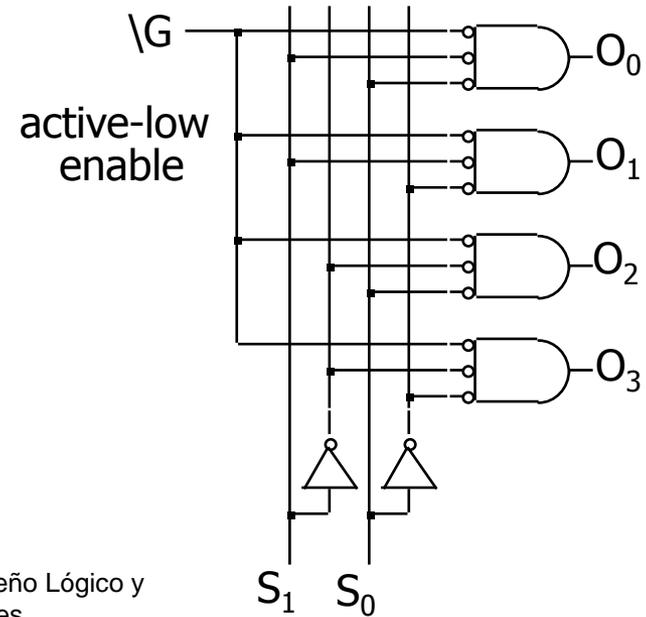
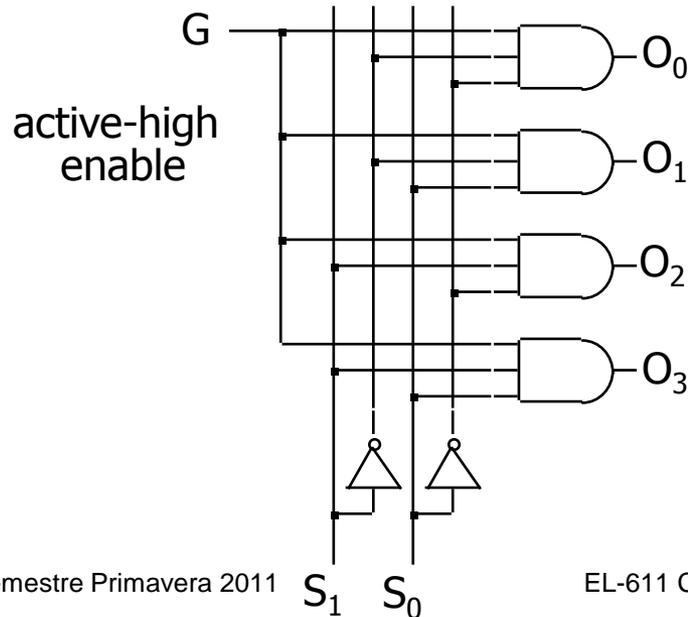
active-high enable



active-low enable

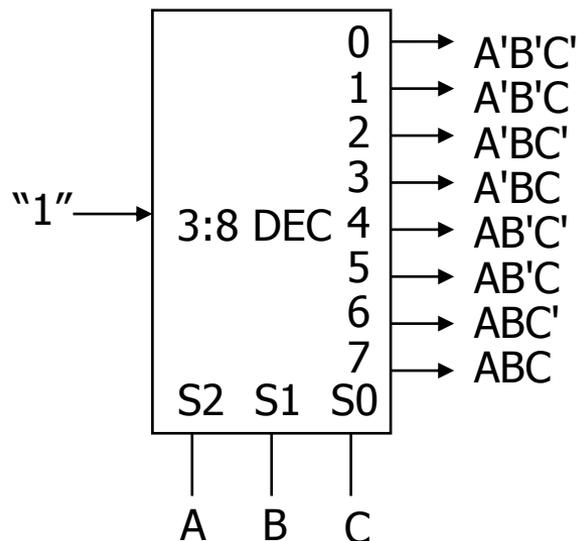


➤ 2:4 decoders



Demultiplexores como lógica de propósitos generales

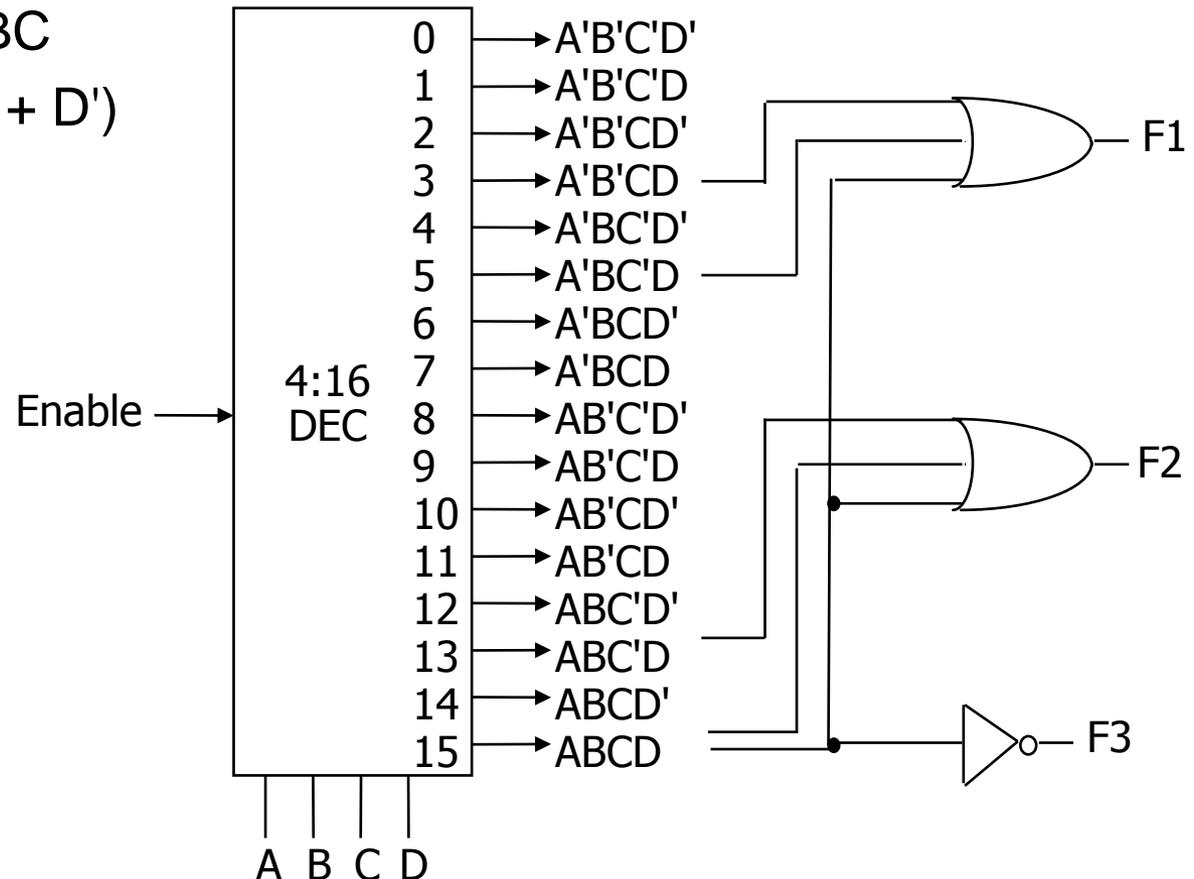
- Un decoder $n:2^n$ puede implementar cualquier función de n variables
 - con las variables utilizadas como entradas de control
 - las entradas “enable” conectadas a un 1, y
 - los minitérminos correspondientes “sumados” para formar la función



el demultiplexor genera los correspondiente minitérminos en base a las líneas de control (decodifica las líneas de control)

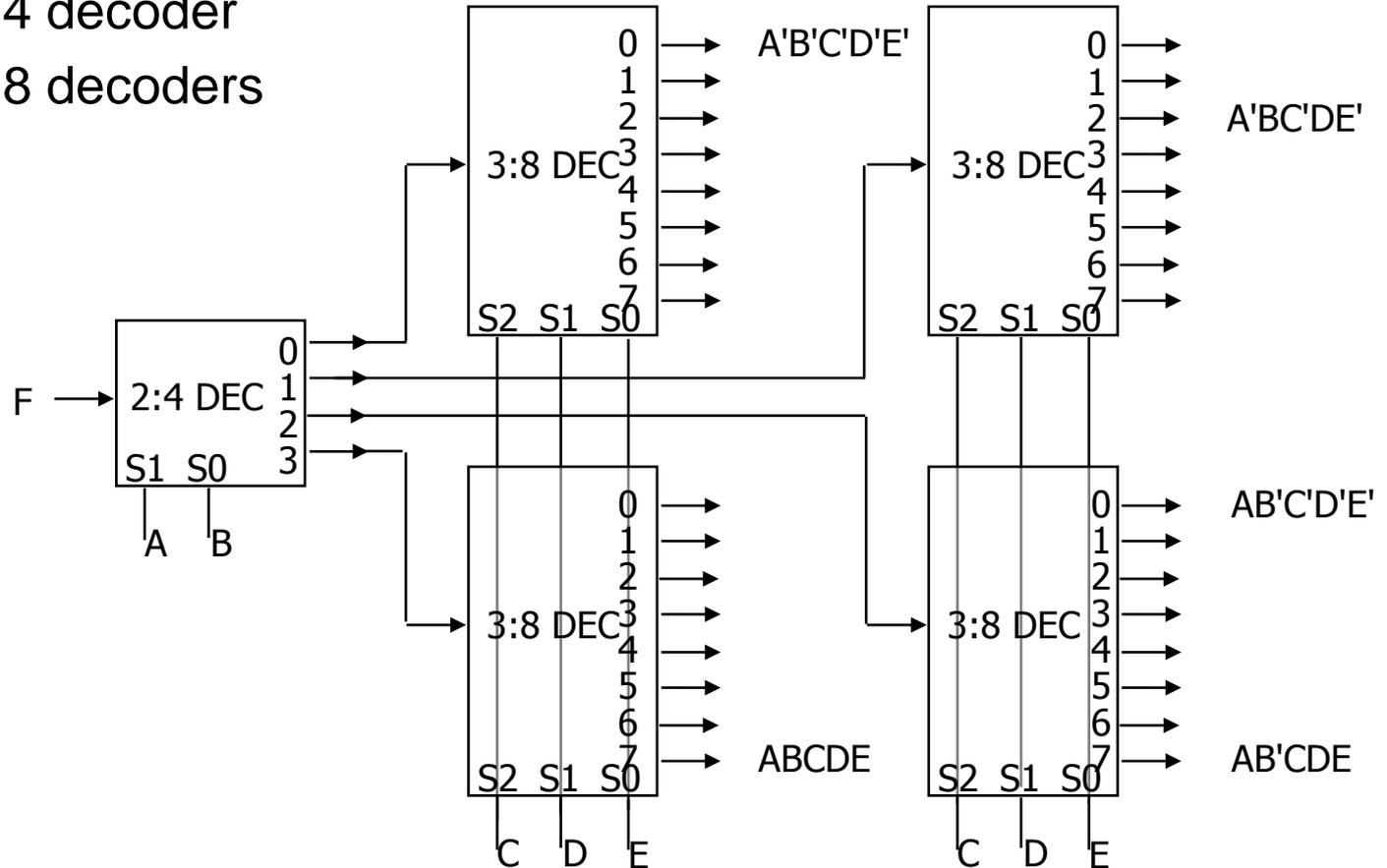
Demultiplexores como lógica de propósitos generales (cont.)

- $F1 = A'BC'D + A'B'CD + ABCD$
- $F2 = ABC'D' + ABC$
- $F3 = (A' + B' + C' + D')$

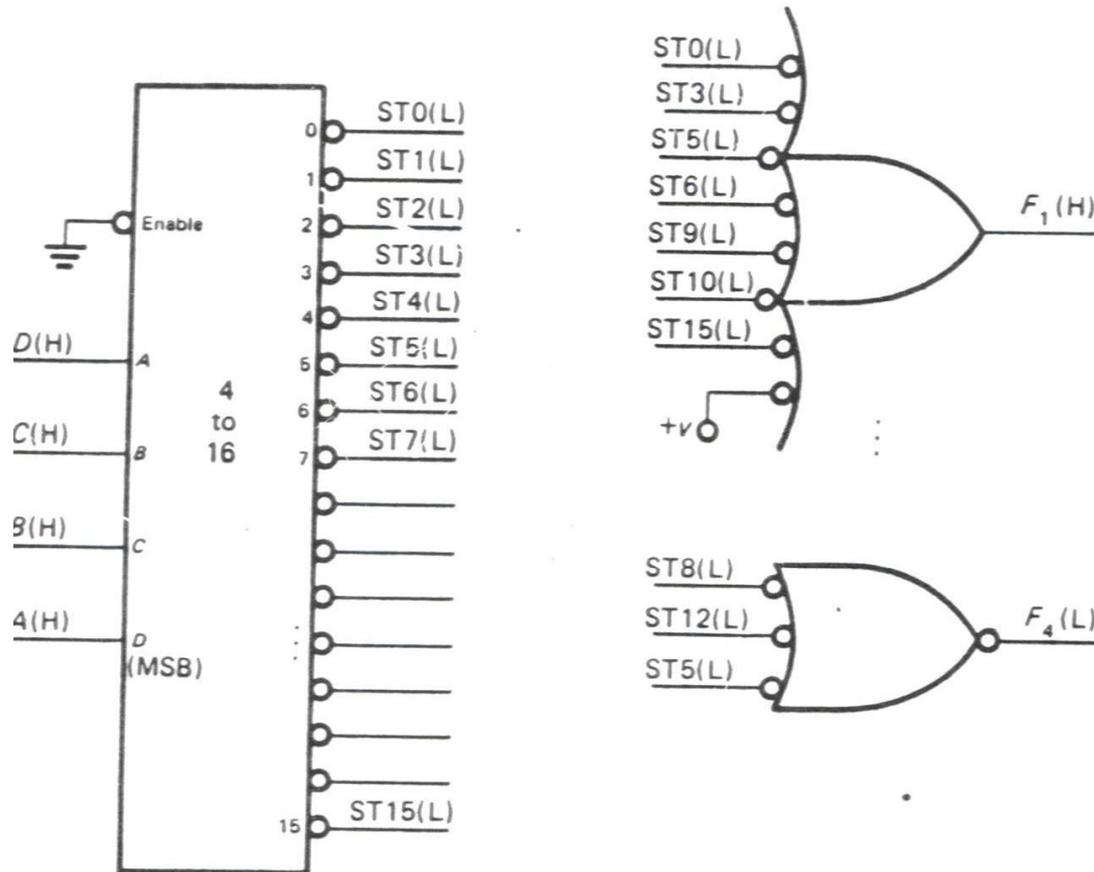


Decoders en Cascada

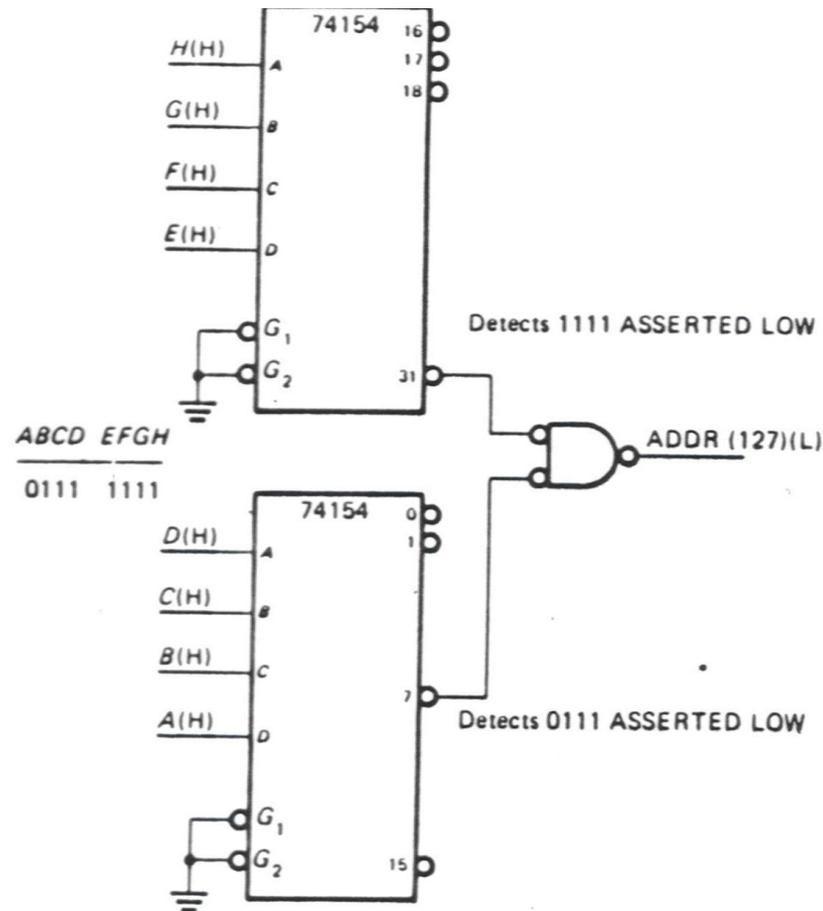
- 5:32 decoder
 - 1x2:4 decoder
 - 4x3:8 decoders



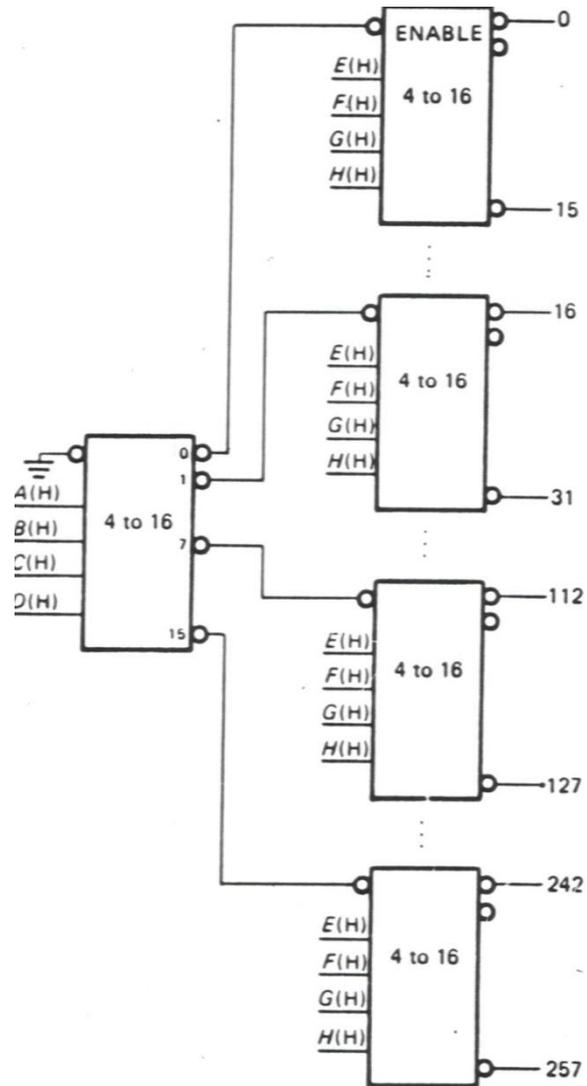
Decodificador



Estructuras con Decodificadores



Estructuras con Decodificadores



Bibliografía

- “ULM Implicants for Minimization of Universal Logic Module Circuits”, Raymond P. Voith, IEEE Transaction on Computers, Vol. C-26, N° 5, May 1977
- “An Algorithm of Optimal Logic Design Using Multiplexer”, Ajit Pal, IEEE Transaction on Computers, Vol. C-35, August 1986