



Fundamentos de los computadores

TEMA 3: Bloques combinacionales básicos

- Conocer el funcionamiento de los circuitos combinacionales siguientes:
 - Decodificadores
 - Codificadores
 - Multiplexores
 - Demultiplexores

1. Introducción
2. Decodificadores
 1. Decodificadores binarios
 2. Composición de decodificadores binarios
 3. Decodificadores no binarios
3. Codificadores
4. Multiplexores
 1. Composición de multiplexores
 2. Multiplexores para datos de n bits
5. Demultiplexores

- En el tema anterior se han estudiado los principios básicos necesarios para abordar la descripción e implementación de circuitos digitales a partir de las puertas lógicas elementales.
- En este tema se aplicarán dichos principios para comprender el funcionamiento e implementación de los circuitos combinatoriales básicos más utilizados.

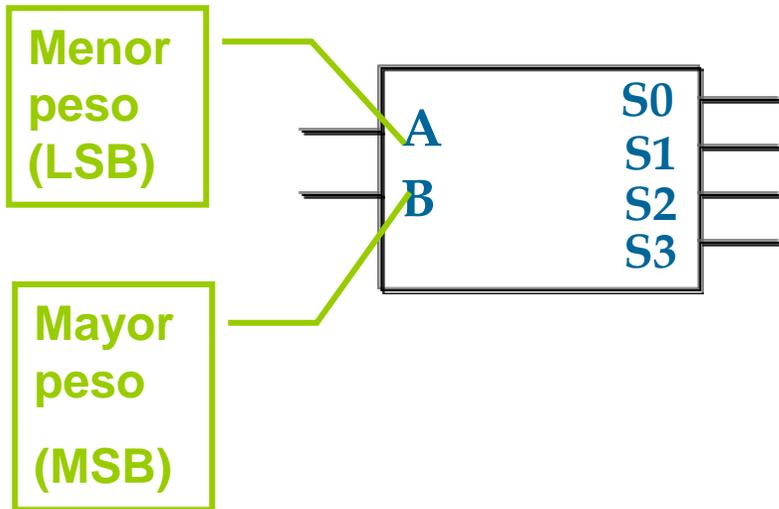
- Estos circuitos implementan funciones sencillas
 - Se pueden encontrar integrados en pastillas (chips)
- Se estudiará la importancia de dichos circuitos como elementos básicos en la implementación de las diferentes unidades funcionales del computador y en la transferencia de información entre ellas.

- En un circuito combinacional, la relación entre las entradas y las salidas puede expresarse mediante una función lógica
 - El valor de las salidas en un instante dado depende exclusivamente del valor de las entradas en ese instante
- Las puertas lógicas introducen un pequeño retardo entre la entrada y la salida (del orden de nanoseg.)
 - En un circuito combinacional real los cambios en las entradas se manifiestan en las salidas con un retardo
 - El retardo depende del tipo de puertas, su n^0 de entradas y el nivel del circuito



- Decodificadores binarios
 - m entradas y $n = 2^m$ salidas (2 a 4, 3 a 8, 4 a 16)
 - Son de gran utilidad para habilitar dispositivos
- Decodificadores de BCD a 7 segmentos
 - 4 entradas y 7 salidas
- Decodificadores de BCD a decimal
 - 4 entradas y 10 salidas

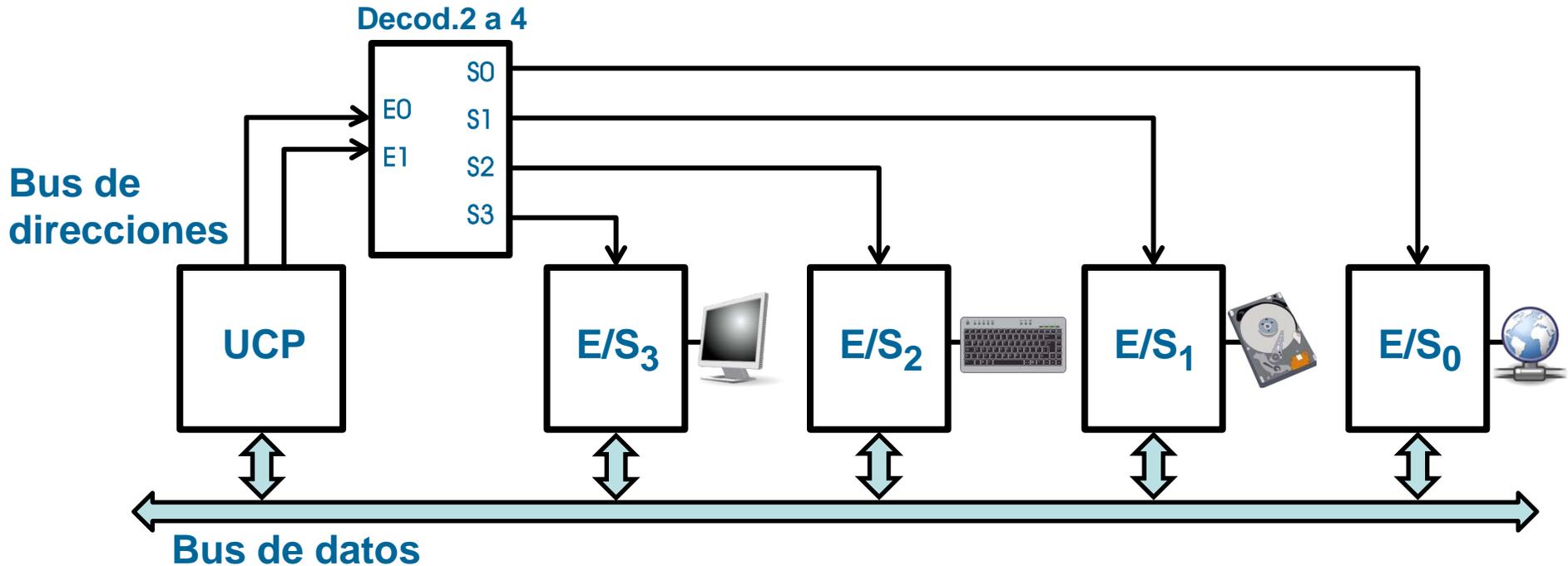
- Decodificador binario



| ENTRADAS | | SALIDAS | | | |
|----------|---|---------|----|----|----|
| B | A | S3 | S2 | S1 | S0 |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

Las salidas son mutuamente excluyentes

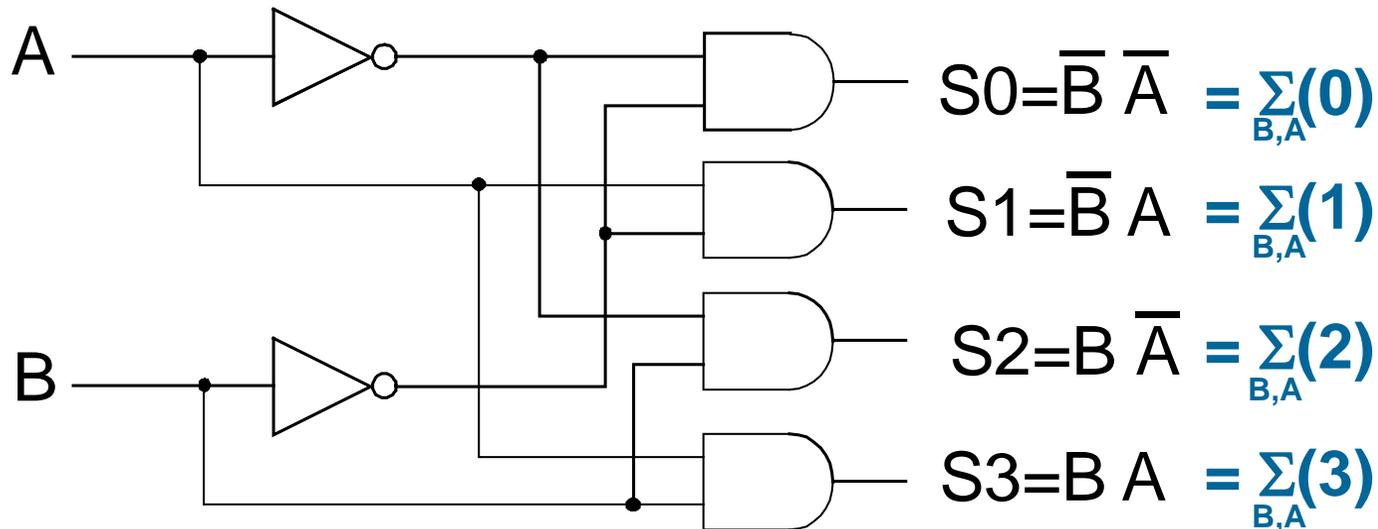
- Uso de un decodificador binario para habilitar dispositivos:



- Diseño de un decodificador binario

| ENTRADAS | | SALIDAS | | | |
|----------|---|---------|----|----|----|
| B | A | S3 | S2 | S1 | S0 |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

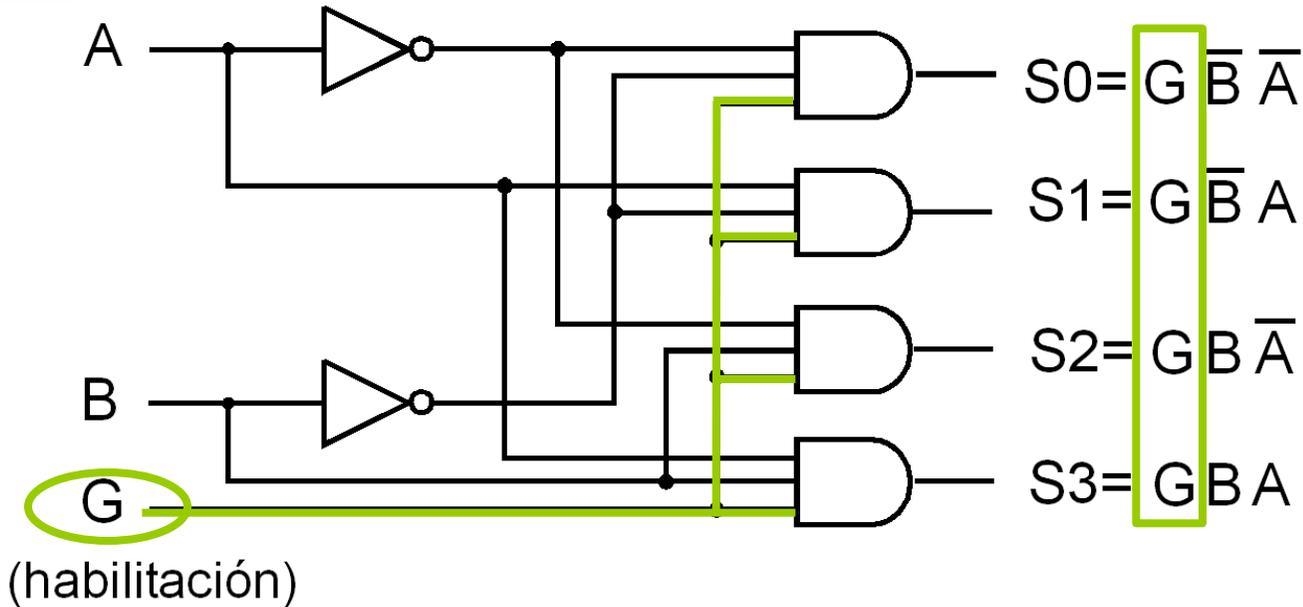
Las funciones de salida sólo valen 1 para una valoración → No se podrán formar grupos por Karnaugh → No hay simplificación posible, sirve la forma canónica para obtener la expresión algebraica.



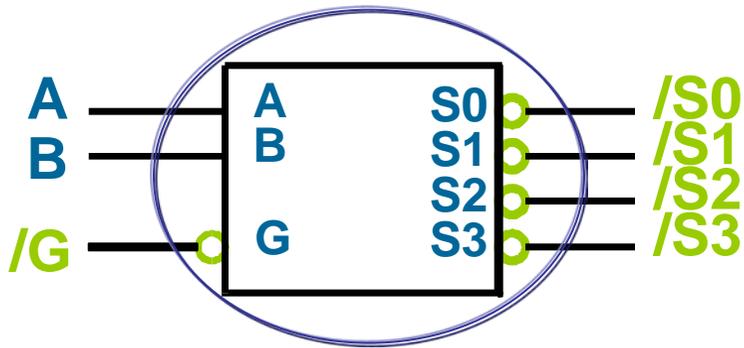
- Con entrada de habilitación
(Enable o strobe)



| ENTRADAS | | | SALIDAS | | | |
|----------|---|---|---------|----|----|----|
| G | B | A | S3 | S2 | S1 | S0 |
| 0 | X | X | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |



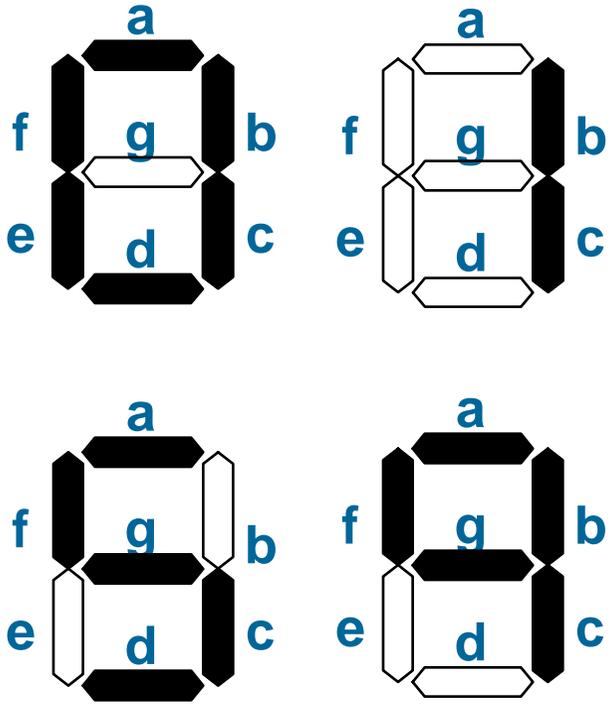
- Circuito integrado 74LS139



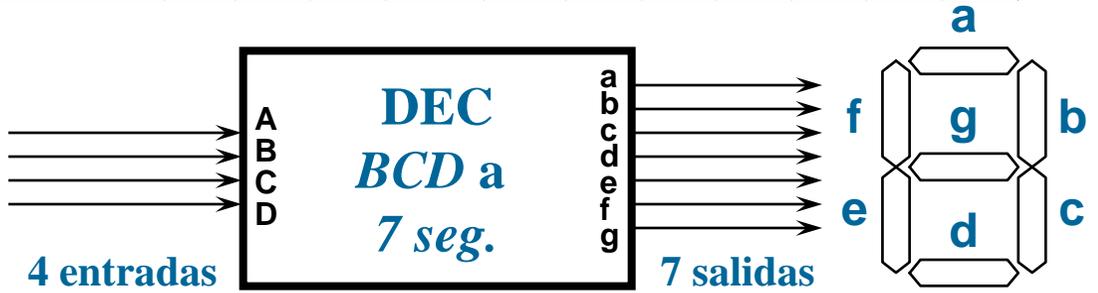
Entrada de habilitación y salidas activas a nivel bajo. Se indica con los círculos en el símbolo lógico y las barras en los nombres de las variables.

| ENTRADAS | | | SALIDAS | | | |
|----------|---|---|---------|-----|-----|-----|
| /G | B | A | /S3 | /S2 | /S1 | /S0 |
| 1 | X | X | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 |

- Decodificadores BCD a 7 segmentos (salidas no excluyentes)



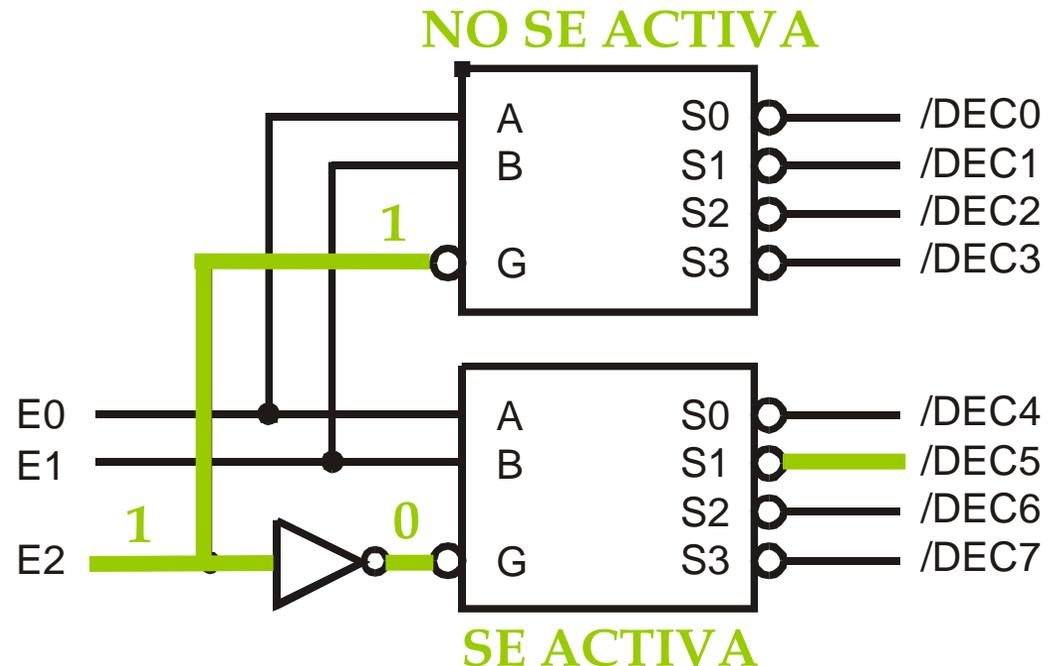
| DECIMAL | ENTRADAS | | | | SALIDAS | | | | | | |
|---------|----------|---|---|---|---------|---|---|---|---|---|---|
| | D | C | B | A | a | b | c | d | e | f | g |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |



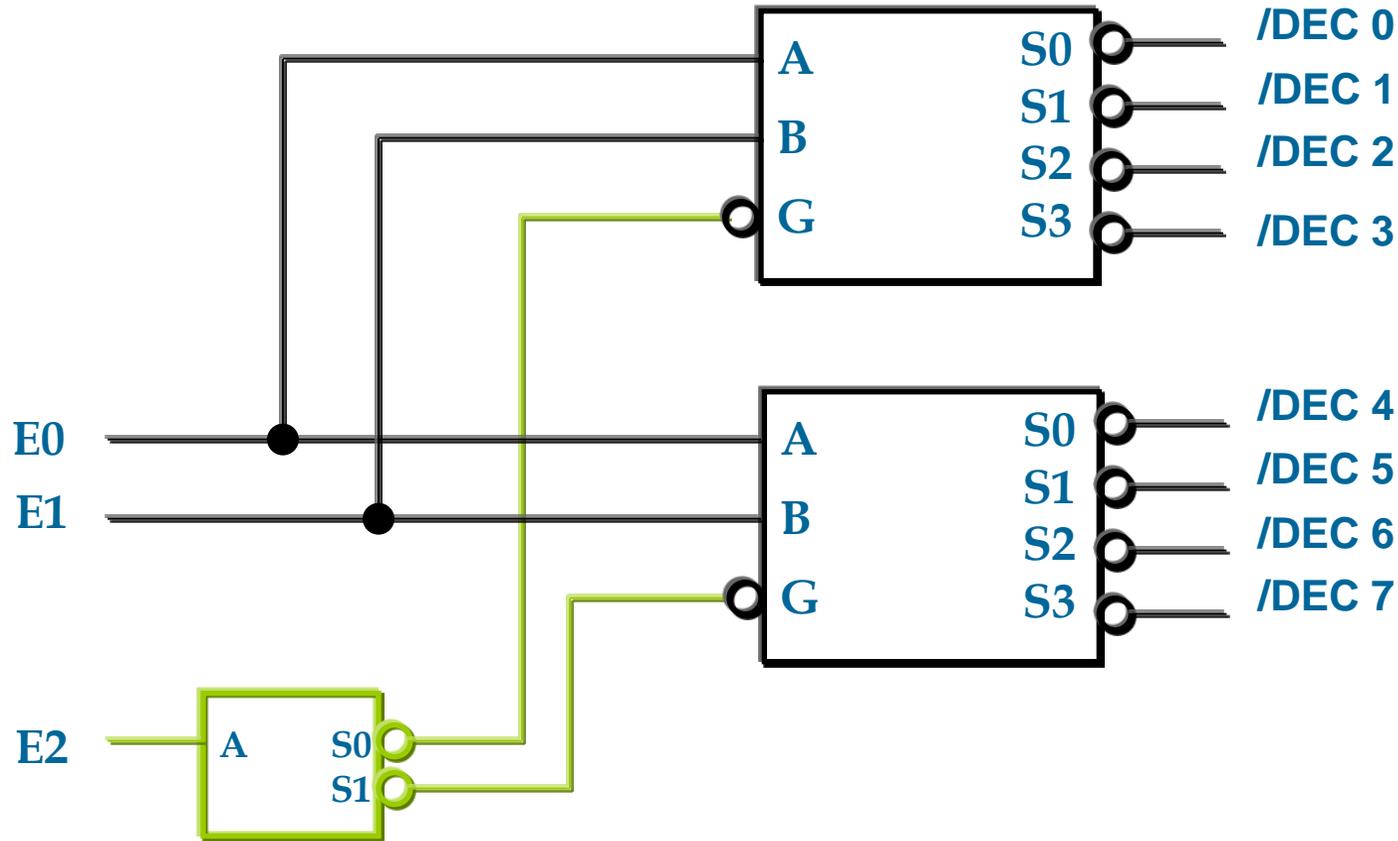
- Tamaño más grande existente en el mercado: 4 a 16
- Podemos implementar decodificadores mayores combinando o componiendo decodificadores más pequeños en paralelo

Ejemplo: Decod.de 3 a 8 (con decodificadores de 2 a 4)

| E2 | E1 | E0 | |
|----|----|----|-------|
| 0 | 0 | 0 | /DEC0 |
| 0 | 0 | 1 | /DEC1 |
| 0 | 1 | 0 | /DEC2 |
| 0 | 1 | 1 | /DEC3 |
| 1 | 0 | 0 | /DEC4 |
| 1 | 0 | 1 | /DEC5 |
| 1 | 1 | 0 | /DEC6 |
| 1 | 1 | 1 | /DEC7 |



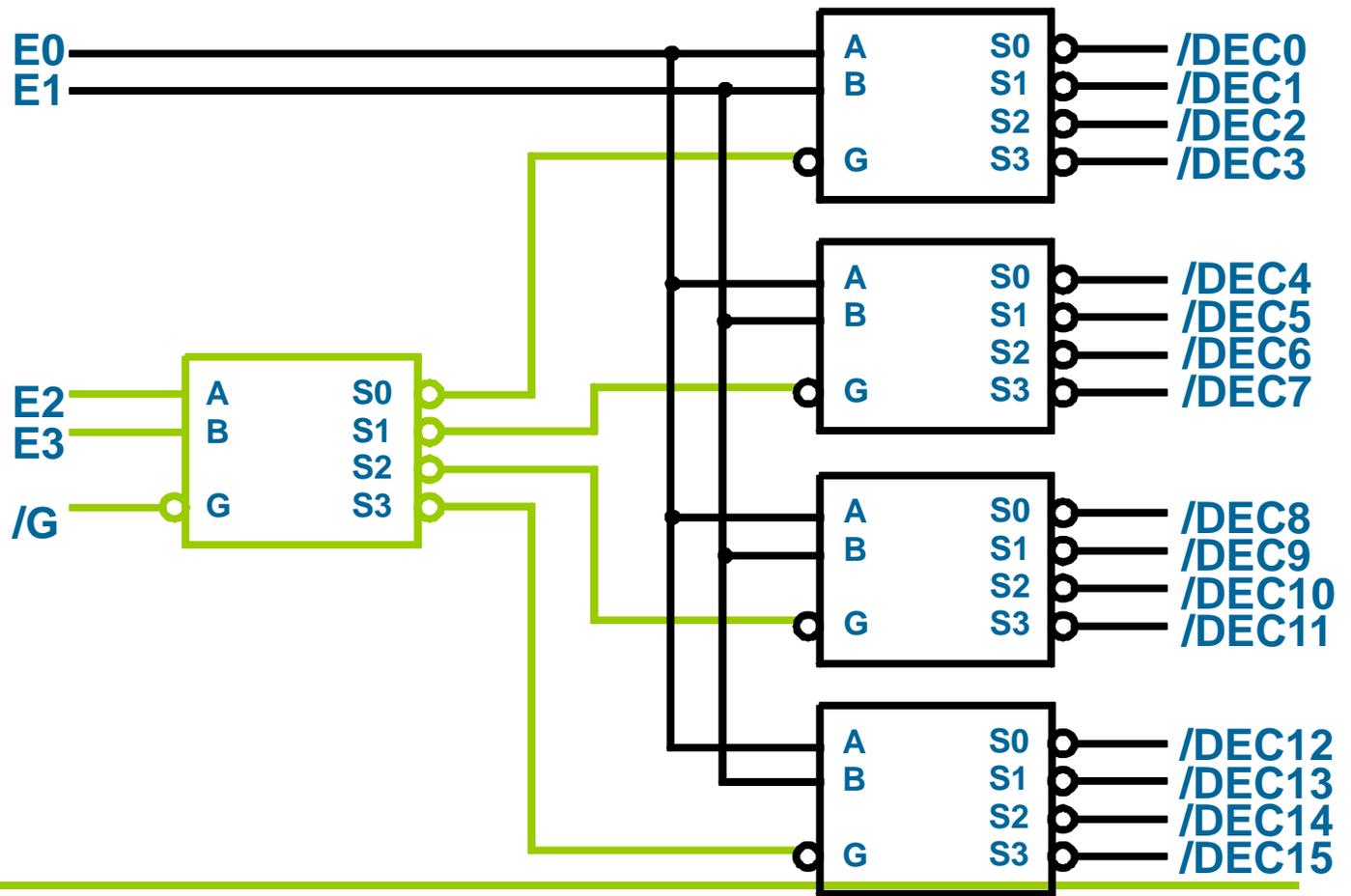
Decodificador de 3 a 8 utilizando dos decodificadores de 2 a 4 y otro de 1 a 2:



Composición de decodificadores (iii)

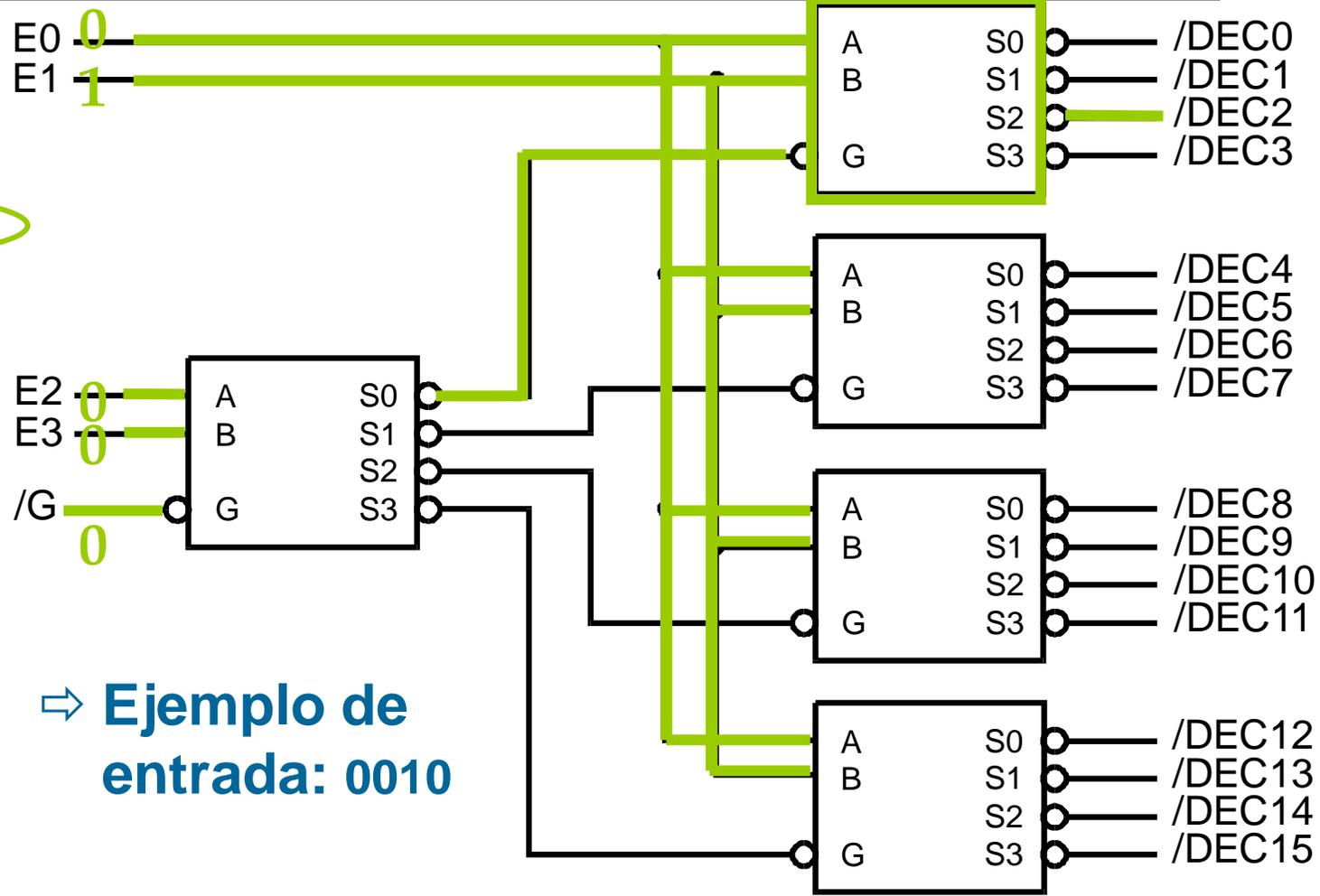
Decod. de 4 a 16 con decodificadores de 2 a 4 (74LS139)

| | E3 | E2 | E1 | E0 | |
|------|----|----|----|----|--------|
| (0) | 0 | 0 | 0 | 0 | /DEC0 |
| (1) | 0 | 0 | 0 | 1 | /DEC1 |
| (2) | 0 | 0 | 1 | 0 | /DEC2 |
| (3) | 0 | 0 | 1 | 1 | /DEC3 |
| (4) | 0 | 1 | 0 | 0 | /DEC4 |
| (5) | 0 | 1 | 0 | 1 | /DEC5 |
| (6) | 0 | 1 | 1 | 0 | /DEC6 |
| (7) | 0 | 1 | 1 | 1 | /DEC7 |
| (8) | 1 | 0 | 0 | 0 | /DEC8 |
| (9) | 1 | 0 | 0 | 1 | /DEC9 |
| (10) | 1 | 0 | 1 | 0 | /DEC10 |
| (11) | 1 | 0 | 1 | 1 | /DEC11 |
| (12) | 1 | 1 | 0 | 0 | /DEC12 |
| (13) | 1 | 1 | 0 | 1 | /DEC13 |
| (14) | 1 | 1 | 1 | 0 | /DEC14 |
| (15) | 1 | 1 | 1 | 1 | /DEC15 |



Composición de decodificadores (iv)

| | E3 | E2 | E1 | E0 | |
|------|----|----|----|----|--------|
| (0) | 0 | 0 | 0 | 0 | /DEC0 |
| (1) | 0 | 0 | 0 | 1 | /DEC1 |
| (2) | 0 | 0 | 1 | 0 | /DEC2 |
| (3) | 0 | 0 | 1 | 1 | /DEC3 |
| (4) | 0 | 1 | 0 | 0 | /DEC4 |
| (5) | 0 | 1 | 0 | 1 | /DEC5 |
| (6) | 0 | 1 | 1 | 0 | /DEC6 |
| (7) | 0 | 1 | 1 | 1 | /DEC7 |
| (8) | 1 | 0 | 0 | 0 | /DEC8 |
| (9) | 1 | 0 | 0 | 1 | /DEC9 |
| (10) | 1 | 0 | 1 | 0 | /DEC10 |
| (11) | 1 | 0 | 1 | 1 | /DEC11 |
| (12) | 1 | 1 | 0 | 0 | /DEC12 |
| (13) | 1 | 1 | 0 | 1 | /DEC13 |
| (14) | 1 | 1 | 1 | 0 | /DEC14 |
| (15) | 1 | 1 | 1 | 1 | /DEC15 |



⇒ Ejemplo de entrada: 0010

- Función opuesta al decodificador



– Codificador binario

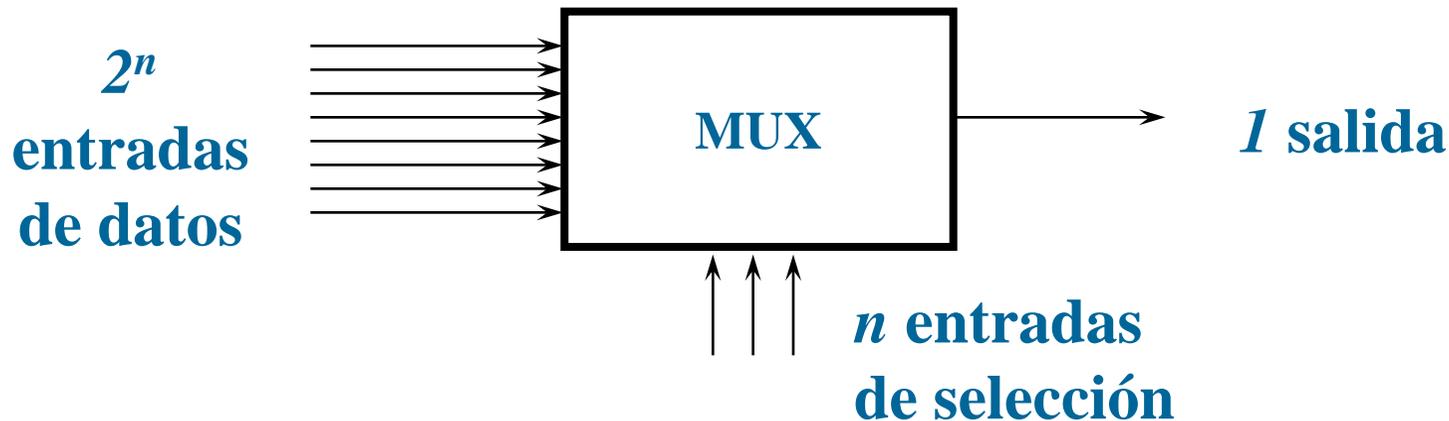
- $m = 2^n$ entradas y n salidas
- La salida codifica en binario el número de la entrada activa
- Se emplean en subsistemas de entrada/salida
 - Ejemplo: el código de salida identifica el dispositivo que realiza una petición al procesador
- Es necesario establecer una prioridad en las entradas si éstas pueden activarse a la vez

- Codificador binario (con prioridad)

| ENTRADAS | | | | SALIDAS | |
|----------|----|----|----|---------|----|
| E3 | E2 | E1 | E0 | S1 | S0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | X | 0 | 1 |
| 0 | 1 | X | X | 1 | 0 |
| 1 | X | X | X | 1 | 1 |

Las entradas de más peso tienen prioridad sobre las de menos peso

- Las líneas de selección indican qué entrada se obtendrá en la salida
- Muy utilizados en los caminos que sigue la información en los sistemas informáticos



| ENTRADAS DE SELECCION | | SALIDA |
|-----------------------|---|--------|
| B | A | S |
| 0 | 0 | E0 |
| 0 | 1 | E1 |
| 1 | 0 | E2 |
| 1 | 1 | E3 |

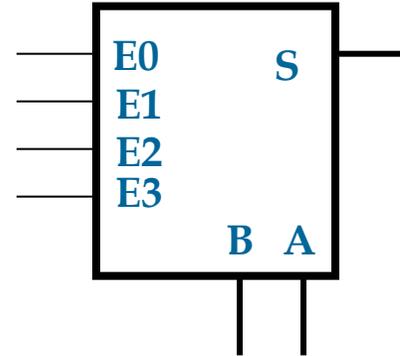


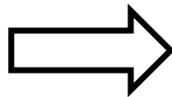
Tabla de verdad extendida

| ENTRADAS DE SELECCION | | ENTRADAS DE DATOS | | | | SALIDA |
|-----------------------|---|-------------------|----|----|----|--------|
| B | A | E3 | E2 | E1 | E0 | S |
| 0 | 0 | X | X | X | 0 | 0 |
| 0 | 0 | X | X | X | 1 | 1 |
| 0 | 1 | X | X | 0 | X | 0 |
| 0 | 1 | X | X | 1 | X | 1 |
| 1 | 0 | X | 0 | X | X | 0 |
| 1 | 0 | X | 1 | X | X | 1 |
| 1 | 1 | 0 | X | X | X | 0 |
| 1 | 1 | 1 | X | X | X | 1 |

– Ejemplo de diseño de un MUX de 2 entradas

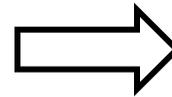
| A | E0 | E1 | S |
|---|----|----|---|
| 0 | 0 | X | 0 |
| 0 | 1 | X | 1 |
| 1 | X | 0 | 0 |
| 1 | X | 1 | 1 |

Tabla reducida



| A | E0 | E1 | S |
|---|----|----|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 |

Tabla extendida

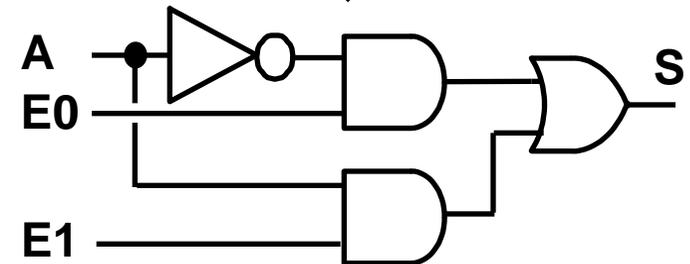


| | | A E0 | | | |
|----|---|----------------|----------------|----------------|----------------|
| | | 00 | 01 | 11 | 10 |
| E1 | 0 | 0 ⁰ | 1 ² | 0 ⁶ | 0 ⁴ |
| | 1 | 0 ¹ | 1 ³ | 1 ⁷ | 1 ⁵ |

Mapa de Karnaugh

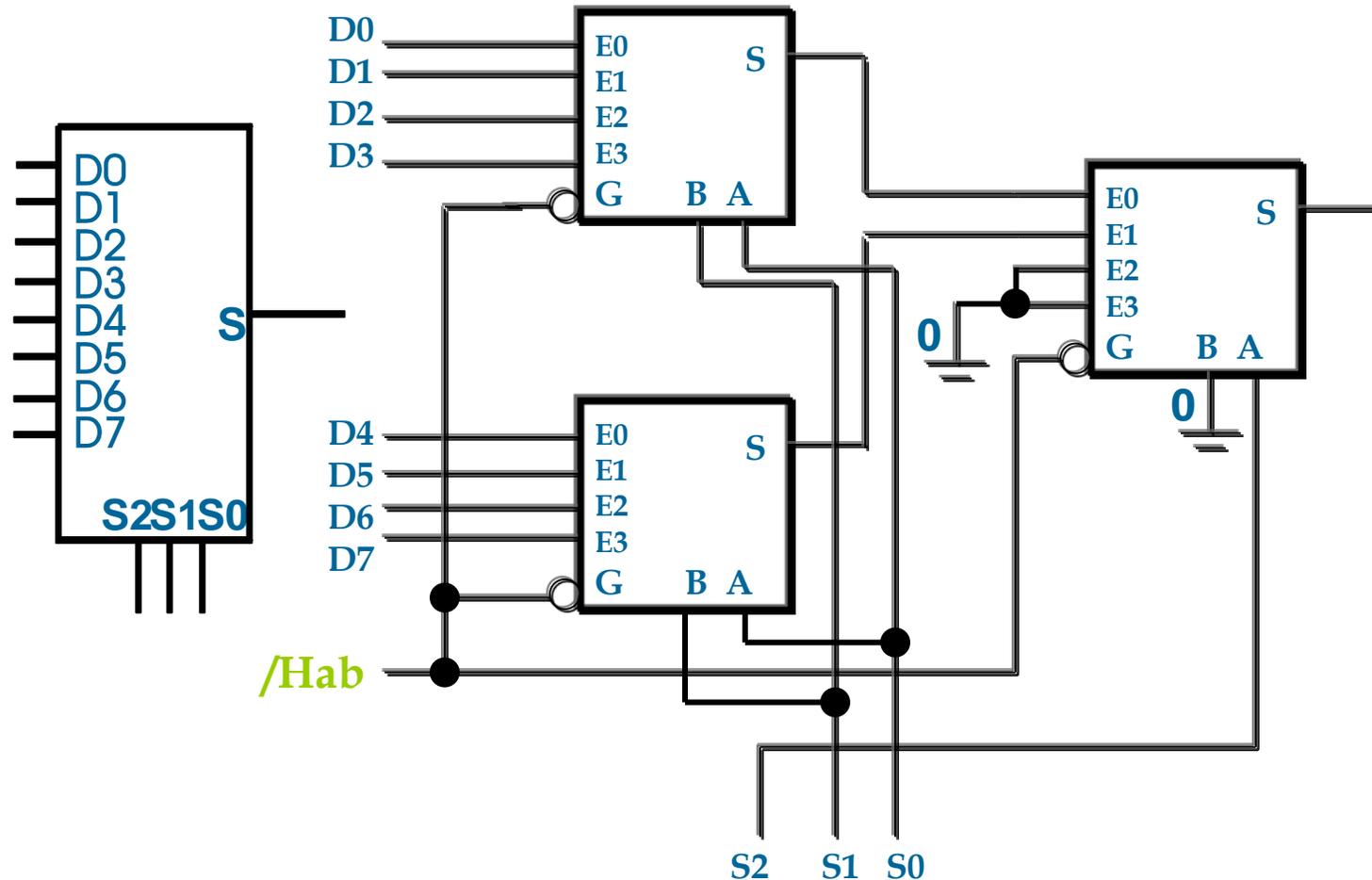


$$S = \bar{A} E0 + A E1$$



MUX de 8 entradas de datos con MUX's de 4 entradas de datos

| S2 | S1 | S0 | S |
|----|----|----|----|
| 0 | 0 | 0 | D0 |
| 0 | 0 | 1 | D1 |
| 0 | 1 | 0 | D2 |
| 0 | 1 | 1 | D3 |
| 1 | 0 | 0 | D4 |
| 1 | 0 | 1 | D5 |
| 1 | 1 | 0 | D6 |
| 1 | 1 | 1 | D7 |

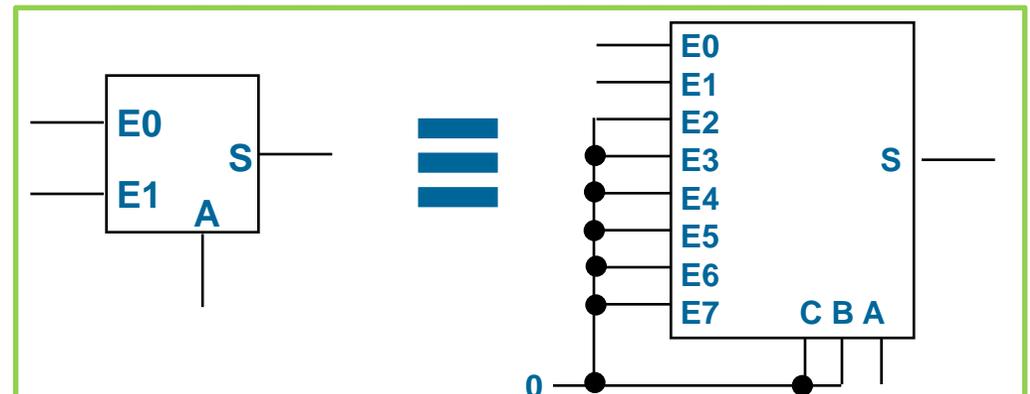


- Método general. Ejemplo:
 - Se desea implementar un multiplexor de 1024 a 1 a base de una composición con multiplexores de 3 entradas de selección. ¿Cuántos se necesitan y cómo se organizan?

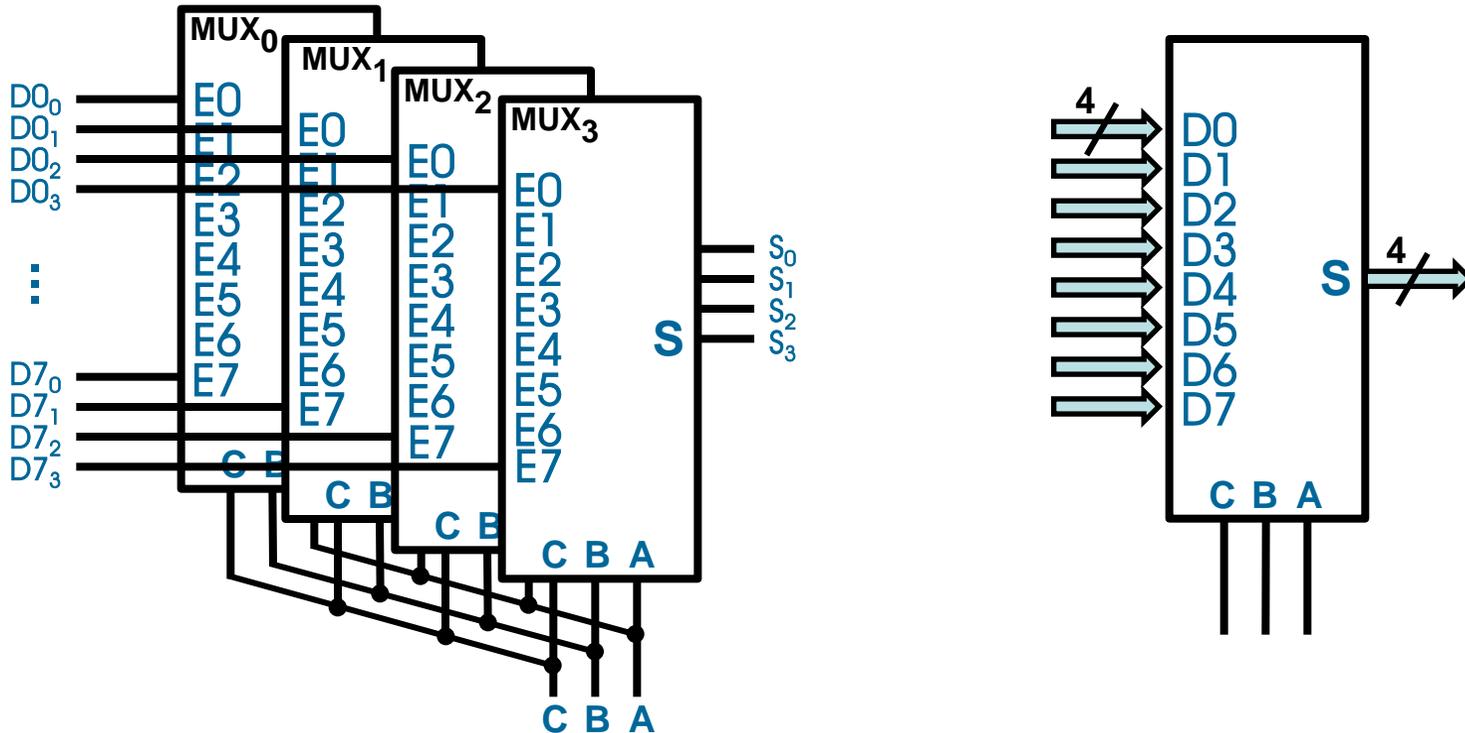


- Para cubrir 1024 entradas con mux. de 8 entradas necesitamos $1024 / 8 = 128$ mux. de 8 entradas en el nivel 1.
- Para cubrir las salidas de esos 128 multiplexores necesitamos $128 / 8 = 16$ mux. de 8 entradas en el nivel 2.
- Para cubrir las salidas de esos 16 multiplexores necesitamos $16 / 8 = 2$ mux. de 8 entradas en el nivel 3.
- Para cubrir las salidas de esos 2 multiplexores necesitamos: 1 mux. de 2 entradas en el nivel 4.
 - Este último se puede implementar también con 1 mux. de 8 entradas:

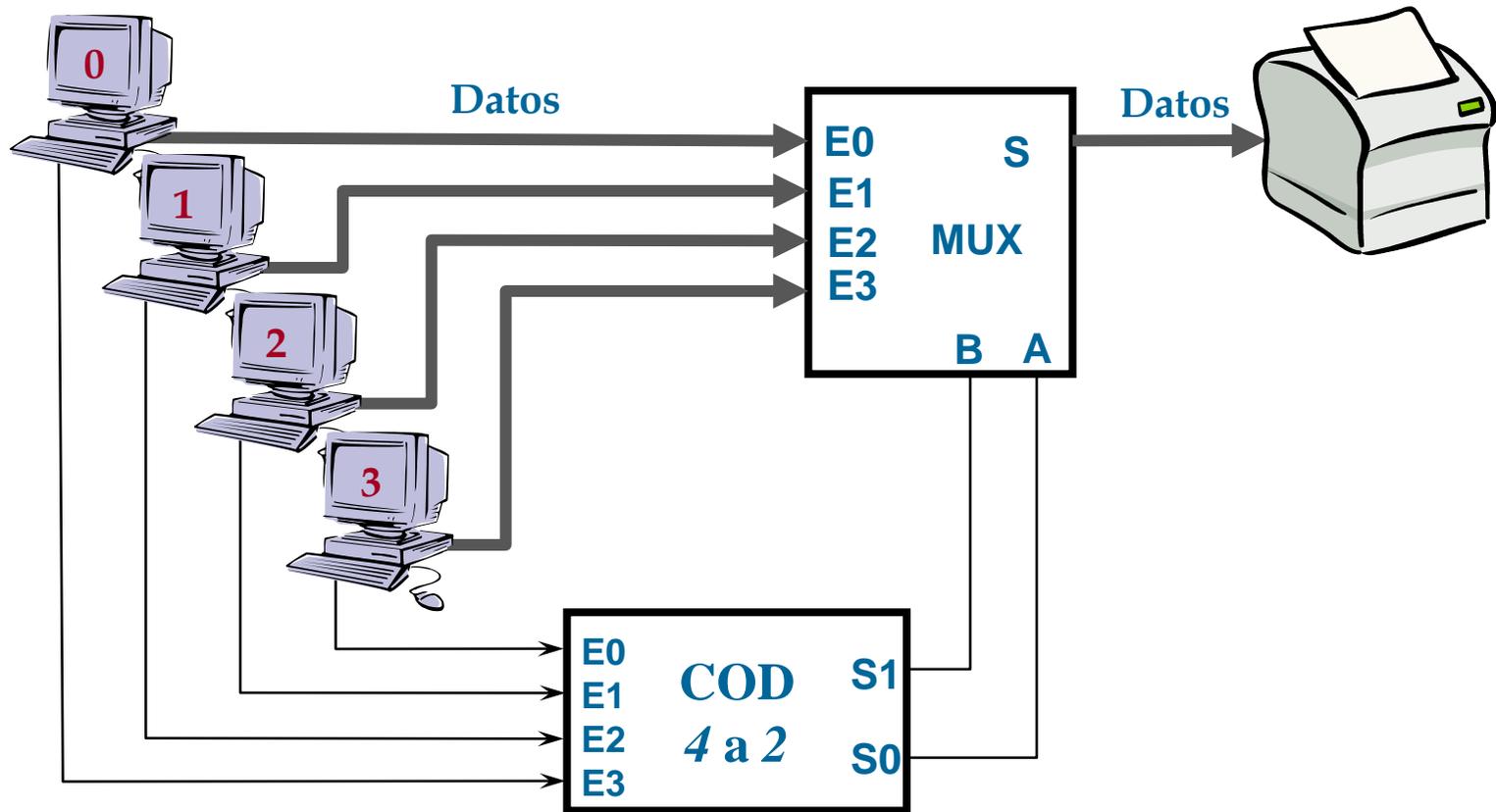
**$128 + 16 + 2 + 1 =$
147 multiplexores de 8
entradas**



- Construcción de multiplexores de datos de ancho mayor que 1 bit.
 - Ejemplo: MUX de 8 entradas de datos de 4 bits



- Ejemplo de uso de un MUX de 4 entradas



- Se pueden construir a partir de decodificadores
- Pueden ser utilizados para habilitar dispositivos

