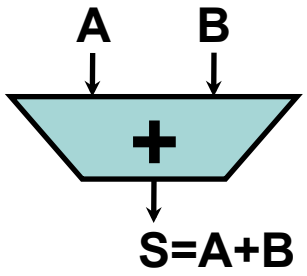
Circuitos combinacionales

© Luis Entrena, Celia López,
Mario García, Enrique San Millán

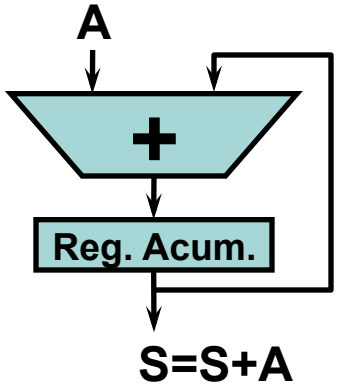
Universidad Carlos III de Madrid

Circuitos combinacionales y secuenciales

- Combinacionales:
  - Salida depende sólo de la entrada
  - Ejemplo: sumador de dos operandos



- Secuenciales:
  - Salida depende de las entradas y del estado
  - Ejemplo: sumador acumulador



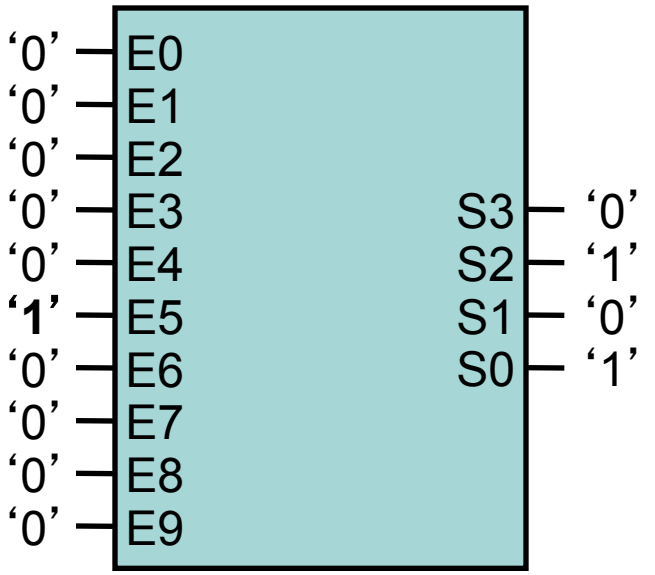
Índice

- Codificadores
- Decodificadores
- Multiplexores
- Demultiplexores
- Comparadores

- Funcionalidad
- Implementación
- Asociación
- Uso para implementación de funciones
- Utilidad

1. Codificadores

- Definición:
  - Circuito combinacional que permite transformar un nivel activo en una de sus entradas en un valor codificado
- Ejemplo: teclado numérico
  - Entradas: dígitos 0-9
  - Salidas: codificación binaria (4 bits)



Activar E5 => S="0101" (=5)

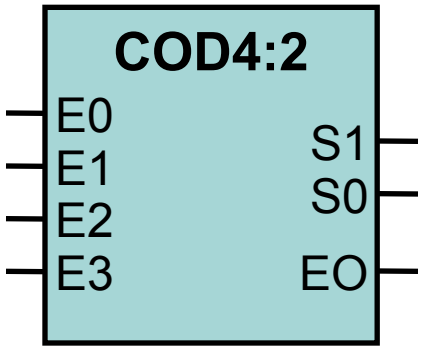
Codificadores sin prioridad

- Características
  - Suponen que sólo una entrada puede estar activa
  - Si se activan varias entradas a la vez, la salida puede ser errónea.
- Funciones lógicas
  - $S_3=E_8+E_9$
  - $S_2=E_4+E_5+E_6+E_7$
  - $S_1=E_2+E_3+E_6+E_7$
  - $S_0=E_1+E_3+E_5+E_7+E_9$
- Problemas:
  - E1 y E4 activas dan resultado 5
  - Ninguna entrada activa da resultado 0

Entrada activa	$S_3S_2S_1S_0$
E_0	0000
E_1	0001
E_2	0010
E_3	0011
E_4	0100
E_5	0101
E_6	0110
E_7	0111
E_8	1000
E_9	1001

Ejemplo: codificador 4:2 sin prioridad

- M:N ⇒ 'M' entradas, 'N' salidas
- EO: "Enable Output"
  - Sirve para diferenciar el caso de activarse E_0 y el de que no haya nada activo
  - También sirve para asociar varios codificadores
- Casos no contemplados
  - Cualquier combinación de activación múltiple
  - Las salidas son indiferentes



E_3	E_2	E_1	E_0	S_1	S_0	EO
0	0	0	1	0	0	0
0	0	1	0	0	1	0
0	1	0	0	1	0	0
1	0	0	0	1	1	0
0	0	0	0	0	0	1
Resto de casos				X	X	X

Ejemplo: codificador 4:2 sin prioridad

E_3	E_2	E_1	E_0	S_1	S_0	EO
0	0	0	1	0	0	0
0	0	1	0	0	1	0
0	1	0	0	1	0	0
1	0	0	0	1	1	0
0	0	0	0	0	0	1
Resto de casos				X	X	X

E_1E_0 / E_3E_2	00	01	11	10
00	0	0	X	0
01	1	X	X	X
11	X	X	X	X
10	1	X	X	X

$$S_1 = E_2 + E_3$$

E_1E_0 / E_3E_2	00	01	11	10
00	1	0	X	0
01	0	X	X	X
11	X	X	X	X
10	0	X	X	X

$$EO = \overline{E_3}\,\overline{E_2}\,\overline{E_1}\,\overline{E_0}$$

E_1E_0 / E_3E_2	00	01	11	10
00	0	0	X	1
01	0	X	X	X
11	X	X	X	X
10	1	X	X	X

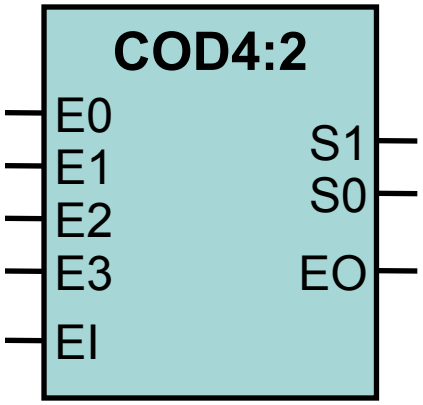
$$S_0 = E_1 + E_3$$

Codificadores con prioridad

- Características
  - Si se activan varias entradas a la vez, dan prioridad a una de ellas
  - Prioridad:
    - Al bit más significativo: se da prioridad a la entrada mayor
      Si se activan E1 y E5, el resultado es 5
    - Al bit menos significativo: se da prioridad a la entrada menor
      Si se activan E1 y E5, el resultado es 1

Ejemplo: codificador 4:2 con prioridad al más significativo

- M:N ⇒ 'M' entradas, 'N' salidas
- EO: "Enable Output"
- EI ó E: "Enable Input" o "Enable". Habilitación
  - Sirve para habilitar:
    - '0' (deshabilitado) implica que las salidas valen '0'
    - '1' (habilitado) indica funcionamiento normal
  - Junto con EO también sirve para asociar varios codificadores



EI	E_3	E_2	E_1	E_0	S_1	S_0	EO
0	X	X	X	X	0	0	0
1	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	X	0	1	0
1	0	1	X	X	1	0	0
1	1	X	X	X	1	1	0

Ejemplo: codificador 4:2 con prioridad al más significativo

EI	E3	E2	E1	E0	S1	S0	EO
0	X	X	X	X	0	0	0
1	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	X	0	1	0
1	0	1	X	X	1	0	0
1	1	X	X	X	1	1	0

- Recordatorio
  - 'X' en las salidas ⇒ 'X' en el diagrama
  - 'X' en las entradas ⇒ múltiples casos

E3E2 \ E1E0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$S_1 = EI(E_2 + E_3)$$

E3E2 \ E1E0	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

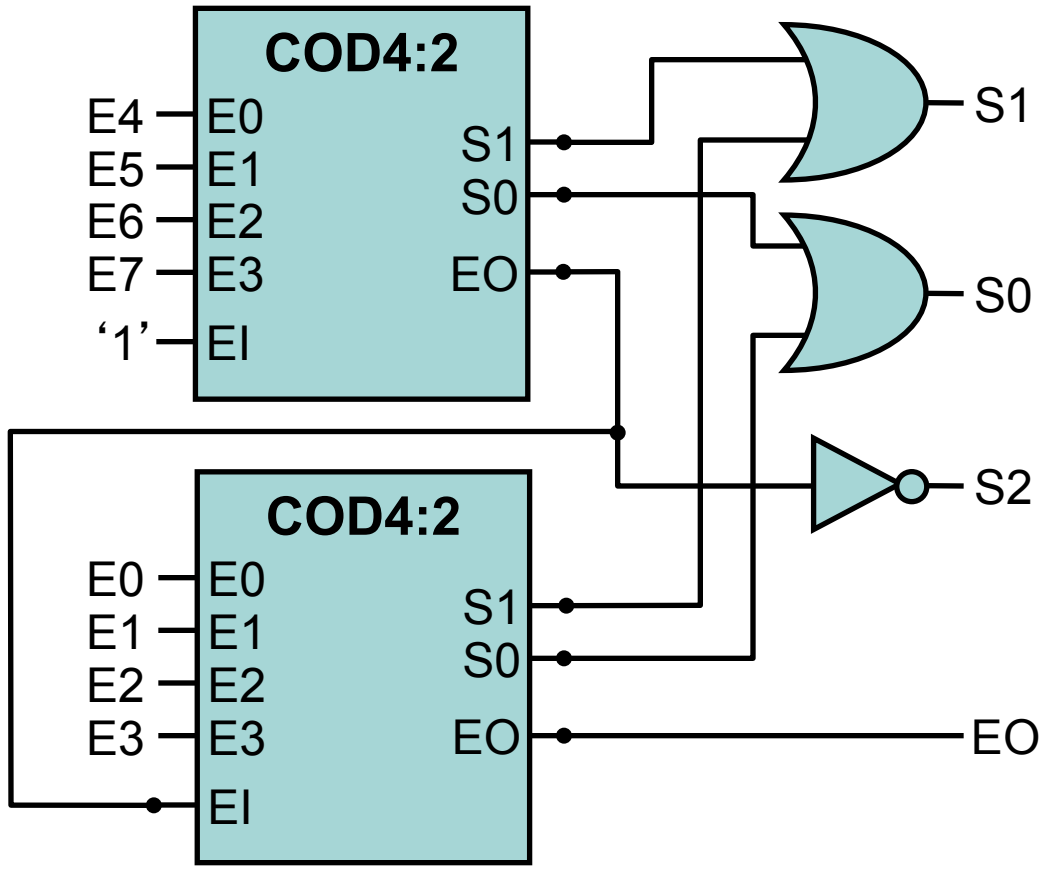
$$S_0 = EI(E_1\overline{E_2} + E_3)$$

E3E2 \ E1E0	00	01	11	10
00	1	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

$$EO = EI(\overline{E_3}\,\overline{E_2}\,\overline{E_1}\,\overline{E_0})$$

Asociación de codificadores: COD8:3 con dos COD 4:2

- Se encadenan los EI y EO
- Cuando un COD está activo (EI= ‘1’) y no tiene ninguna entrada activa, activa al siguiente COD (EO= ‘1’).

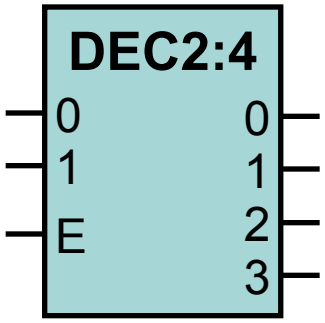


Utilidad de los codificadores

- Sensores de piso de un ascensor
  - Codifican cada sensor al número de piso
  - No necesita prioridad, ya que el ascensor sólo puede estar en un piso
- Botonera
  - Codifica el valor de la tecla pulsada
  - Necesita prioridad, ya que se pueden pulsar varios botones a la vez

2. Decodificadores

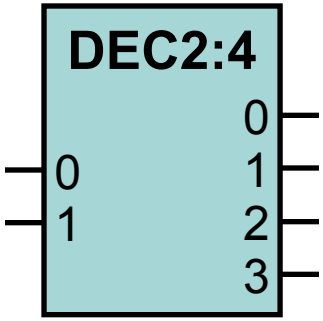
- Definición:
  - Circuito combinacional que transforma un valor codificado en la activación de la salida correspondiente al dicho valor.
  - Realizan la función inversa a los codificadores



E	E_1	E_0	S_3	S_2	S_1	S_0
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

Decodificadores

- Funciones lógicas:
  - Cada salida del decodificador es un mintérmino

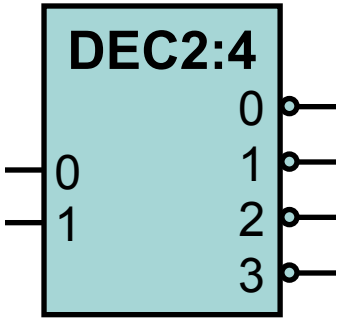


E_1	E_0	S_3	S_2	S_1	S_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$S_0 = \overline{E_1}\,\overline{E_0}$$
$$S_1 = \overline{E_1} E_0$$
$$S_2 = E_1 \overline{E_0}$$
$$S_3 = E_1 E_0$$

Decodificadores

- Decodificador con salidas activas por nivel bajo:
  - Cada salida del decodificador es un maxtérmino

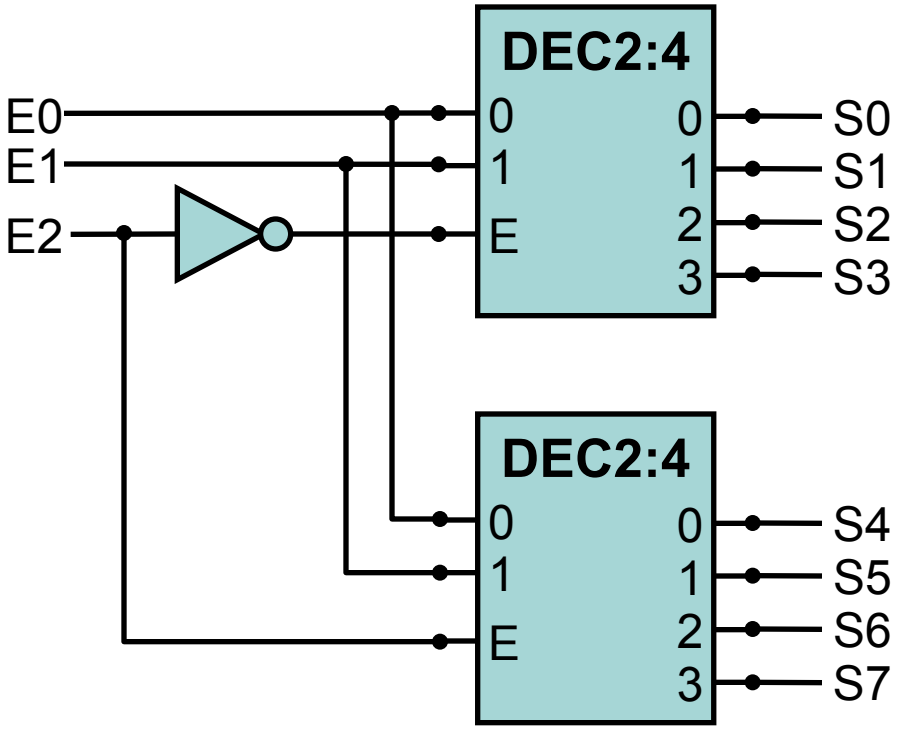


E_1	E_0	S_3	S_2	S_1	S_0
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

$$S_0 = E_1 + E_0$$
$$S_1 = E_1 + \overline{E_0}$$
$$S_2 = \overline{E_1} + E_0$$
$$S_3 = \overline{E_1} + \overline{E_0}$$

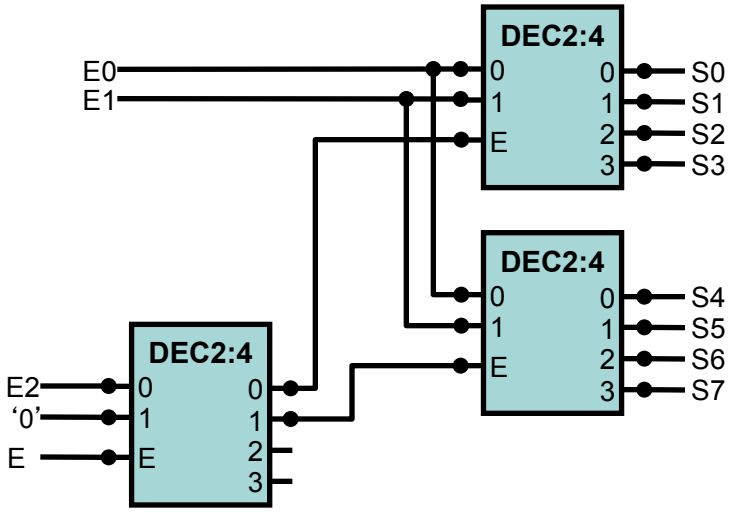
Asociación de decodificadores

- DEC3:8 con DEC2:4
  - Sólo uno de los decodificadores está activo, dependiendo del valor de E2
  - El inversor hace la función de un DEC1:2
  - No tiene Enable global



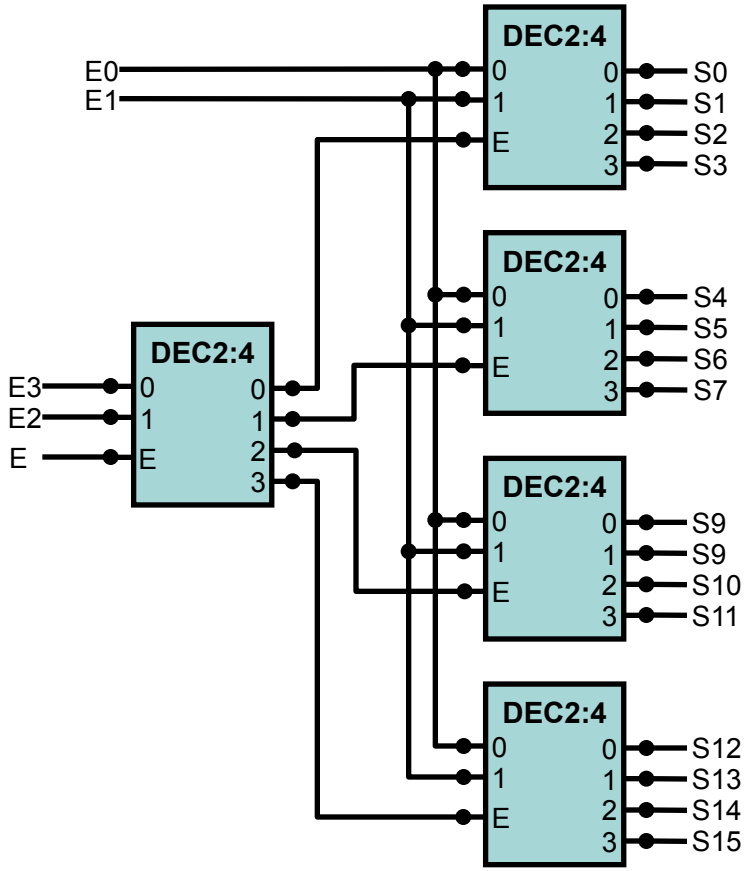
Asociación de decodificadores

- DEC4:16 con DEC2:4
  - Sólo uno de los decodificadores está activo, dependiendo del valor de E2
  - El decodificador de la izquierda se comporta como un DEC1:2
  - Tiene Enable Global. Si E= ‘0’ , ningún decodificador se activa y las salidas valen ‘0’



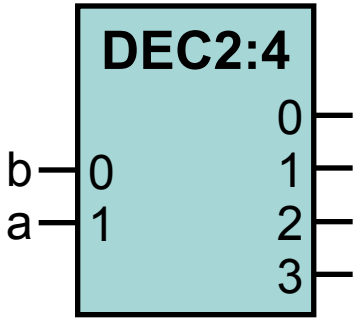
Asociación de decodificadores

- DEC4:16 con DEC2:4
  - Sólo uno de los decodificadores está activo, dependiendo del valor de E3 y E2



Implementación de funciones lógicas con decodificadores

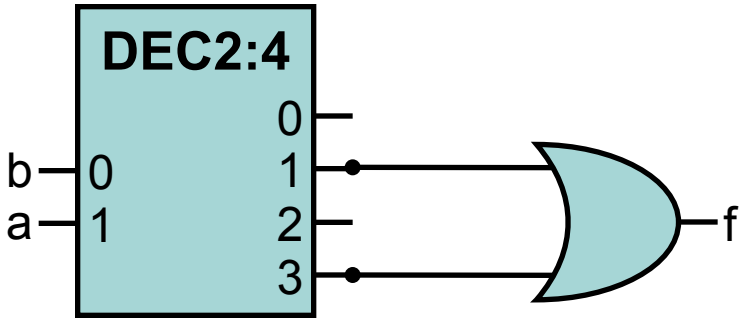
- Se pueden implementar funciones lógicas con un DEC y una puerta OR
- Las salidas del DEC son los mintérminos. Se suman las que valgan ‘1’ en la tabla de verdad
- El dual se hace con DEC de salidas a nivel bajo y una puerta AND.



$$S_0 = \bar{a}\bar{b}$$
$$S_1 = \bar{a}b$$
$$S_2 = a\bar{b}$$
$$S_3 = ab$$

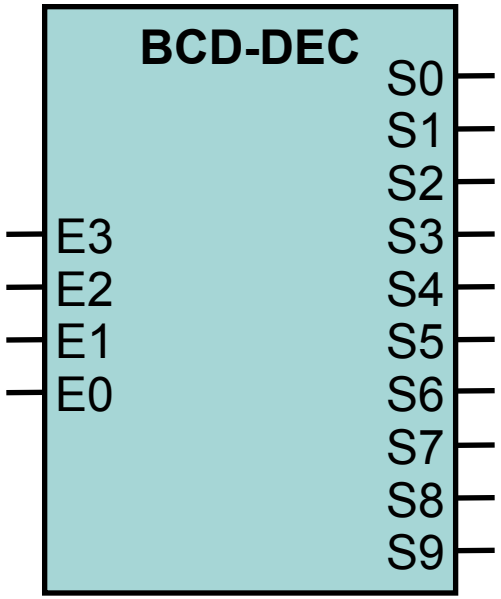
m	a	b	f
0	0	0	0
1	0	1	1
2	1	0	0
3	1	1	1

$$f = \bar{a}b + ab = S_1 + S_3$$



Decodificador BCD-decimal

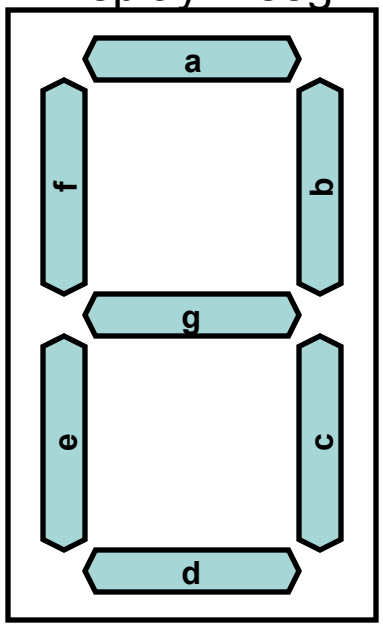
- Decodifica un dígito decimal codificado en BCD (natural) a 10 salidas que representan 0-9
- El comportamiento no está definido si la entrada no es un dígito decimal



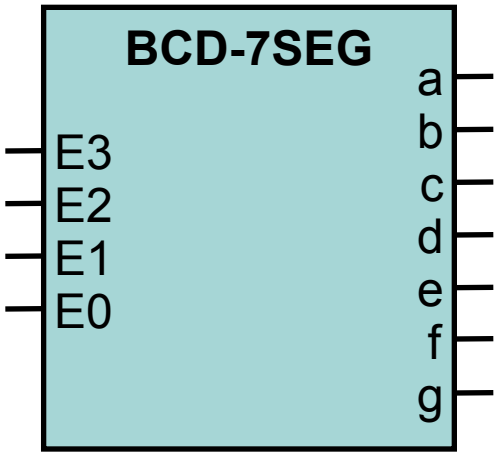
Decodificador BCD-7 segmentos

- Decodifica un dígito decimal codificado en BCD (natural) a los LEDs de un “display 7-segmentos”

Display 7-seg



E_3	E_2	E_1	E_0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
Resto				X	X	X	X	X	X	X

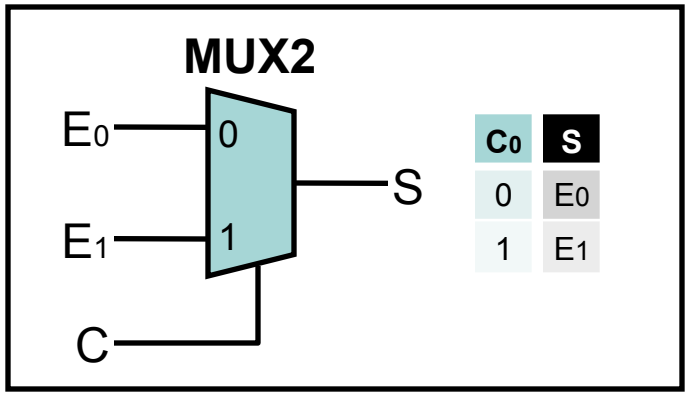


Utilidad de los decodificadores

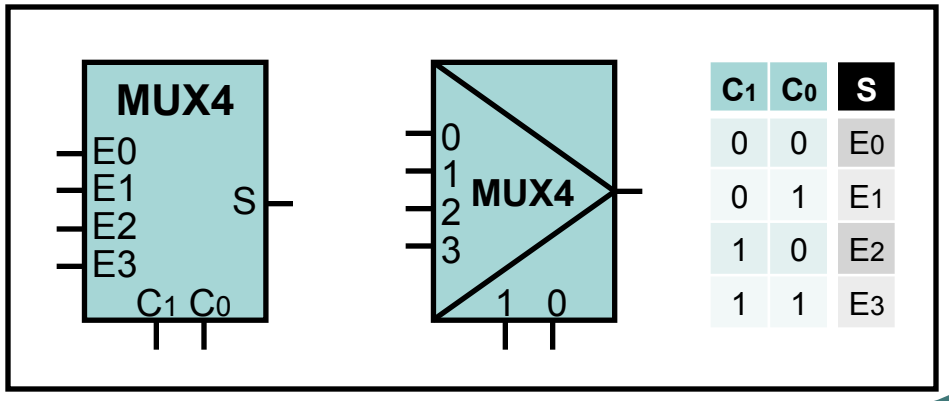
- Microprocesadores:
  - Decodificación de instrucciones
  - Puertos de E/S, direcciones de memoria, etc.

3. Multiplexores

- Definición:
  - Circuito que permite seleccionar una de las entradas y copiar su valor a la salida. La entrada seleccionada depende del valor que se dé a las entradas de control.
  - Se denominan por el número de entradas de dato: MUX2, MUX4, …
  - N=entradas de datos, n=entradas de control $\Rightarrow$ $2^n = N$



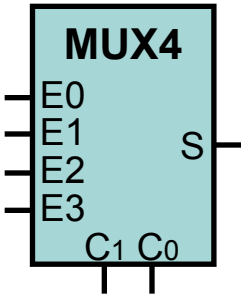
C0	S
0	E0
1	E1



C1	C0	S
0	0	E0
0	1	E1
1	0	E2
1	1	E3

Multiplexores

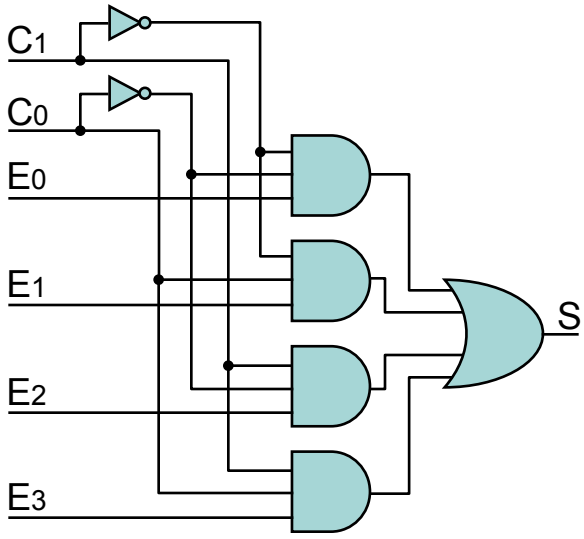
- Función lógica



C_1	C_0	S
0	0	E_0
0	1	E_1
1	0	E_2
1	1	E_3

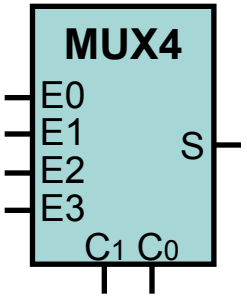
$$S = \overline{C_1}\,\overline{C_0}E_0 + \overline{C_1}C_0E_1 + C_1\overline{C_0}E_2 + C_1C_0E3_0$$

- Implementación con puertas



Multiplexores

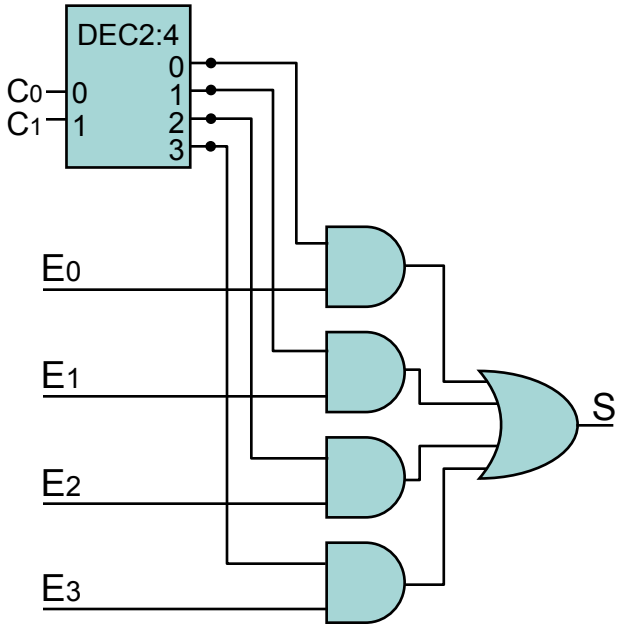
- Función lógica



C_1	C_0	S
0	0	E_0
0	1	E_1
1	0	E_2
1	1	E_3

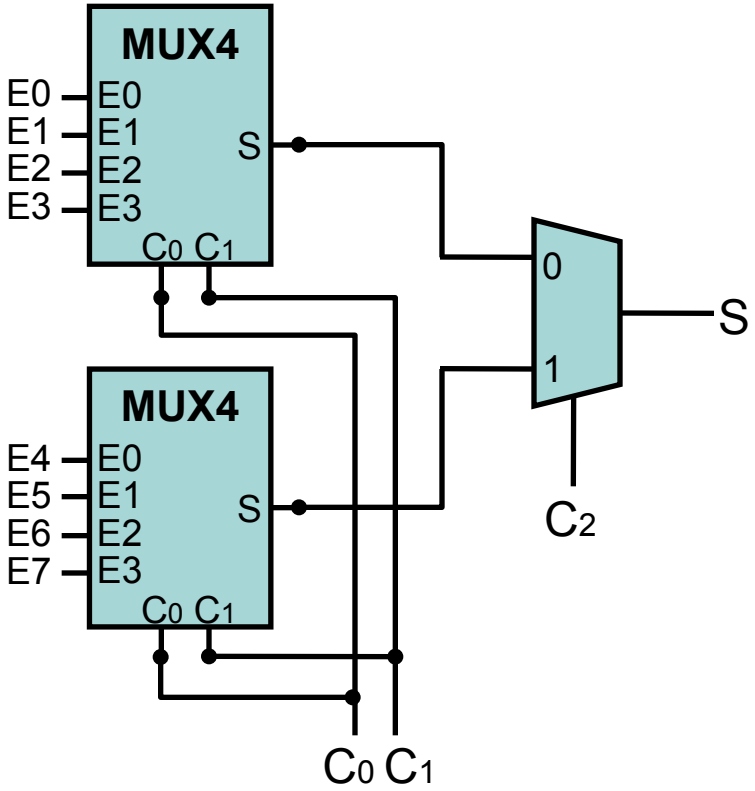
$$S = \overline{C_1}\,\overline{C_0}E_0 + \overline{C_1}C_0E_1 +$$
$$+ C_1\overline{C_0}E_2 + C_1C_0E3_0$$

- Implementación con decodificador



Asociación de multiplexores

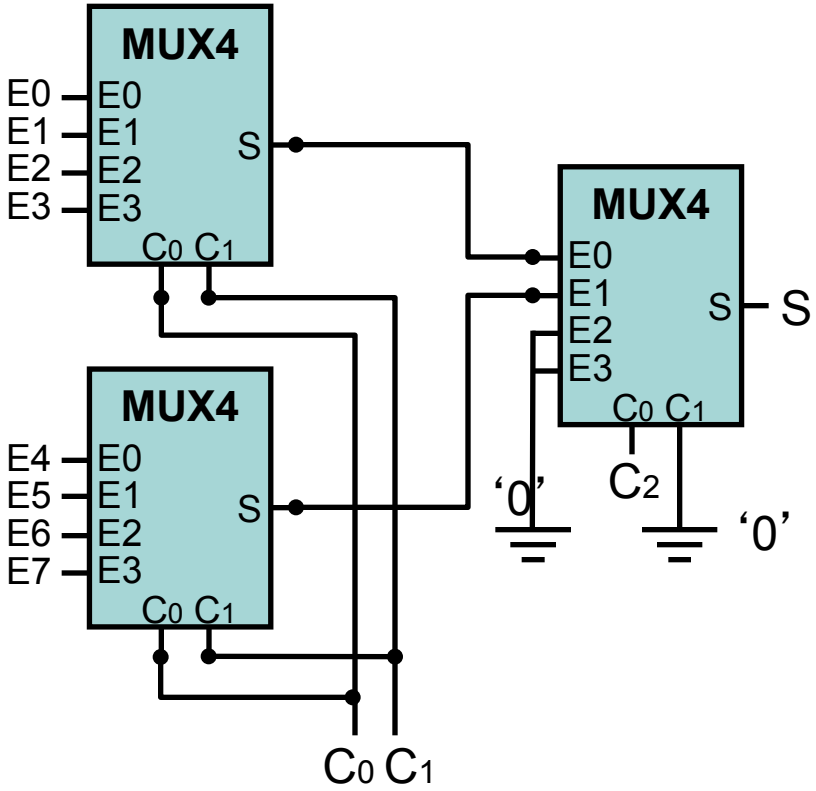
- MUX8 a partir de MUX4 y MUX2



- MUX2 selecciona entre los MUX4, dependiendo del valor del bit de control más significativo (C_2)
- Los bits de C y E deben asignarse según su peso

Asociación de multiplexores

- MUX8 a partir de MUX4

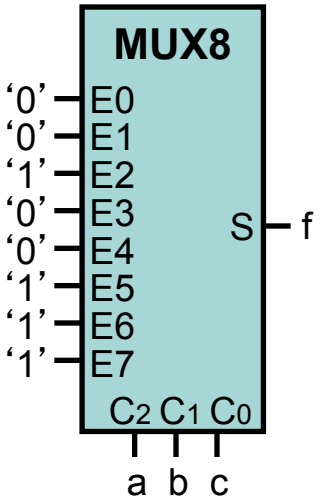


- El MUX4 de la derecha se comporta como un MUX2
- Recordatorio: las entradas de un circuito DEBEN estar conectadas; las salidas pueden quedar desconectadas

Implementación de funciones lógicas con multiplexores

- Con un MUX de tantas entradas de control como variables tiene la función
  - Las variables de la función van al control del MUX, ordenadas por peso
  - Los valores de la función en la tabla de verdad son las entradas de datos del MUX

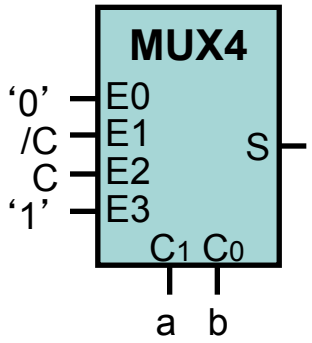
a	b	c	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Implementación de funciones lógicas con multiplexores

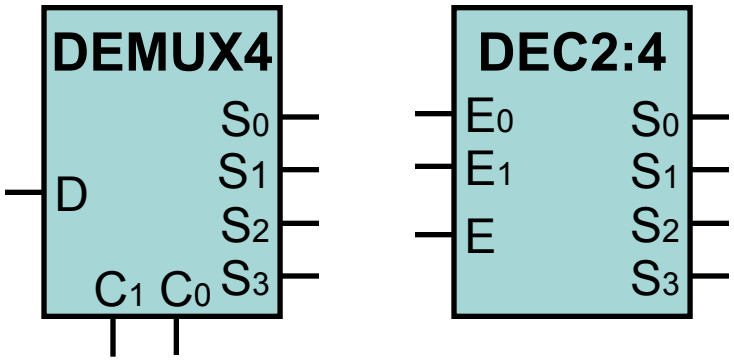
- Con un MUX de menos entradas de control que variables tiene la función
  - Agrupar la tabla de verdad según las variables menos significativas
  - Las variables de la función de mayor peso van al control del MUX, ordenadas por peso
  - Los valores de la función en la tabla de verdad son las entradas de datos del MUX

a	b	c	f	f(c)
0	0	0	0	0
0	0	1	0	
0	1	0	1	/C
0	1	1	0	
1	0	0	0	C
1	0	1	1	
1	1	0	1	1
1	1	1	1	



4. Demultiplexores

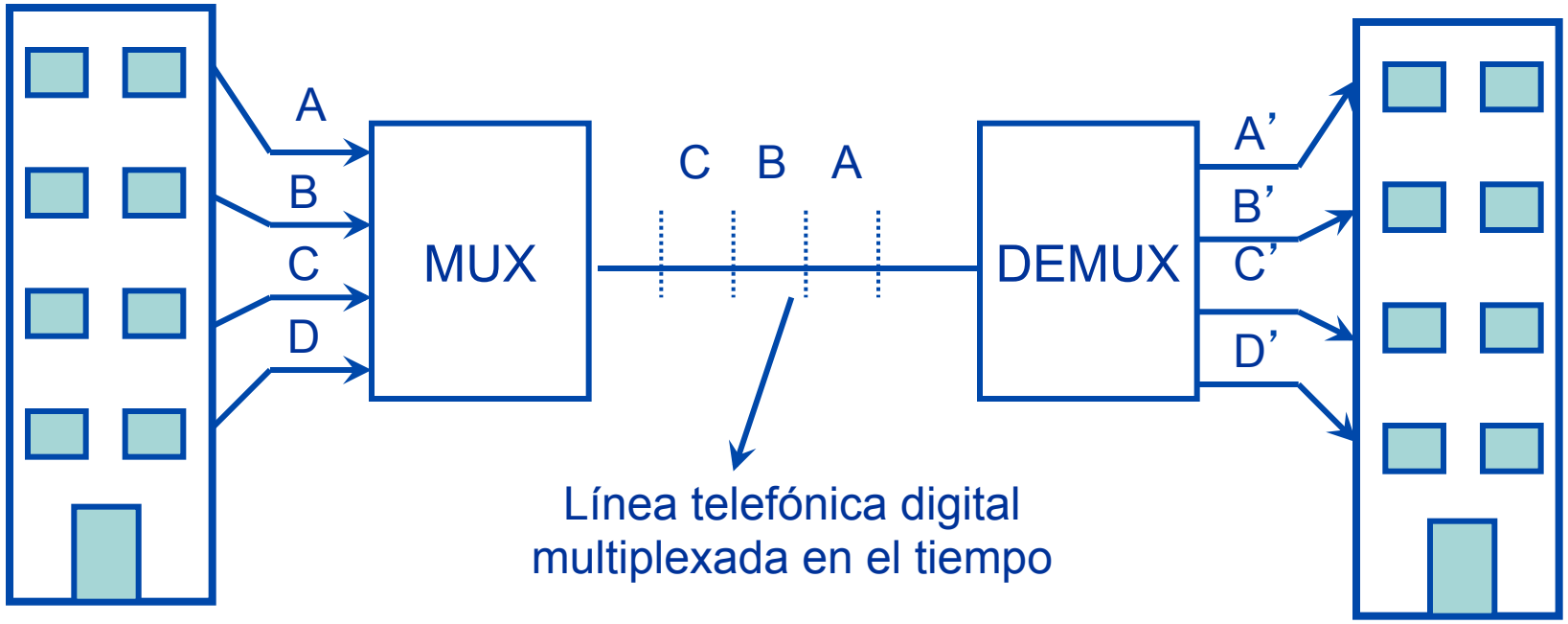
- Definición: circuito que copia el valor de la entrada de datos en la salida indicada por el valor de las señales de control.
- Son el circuito opuesto a los multiplexores
- Son equivalentes a decodificadores, si equiparamos las entradas de control (C_i) del DEMUX a las de datos (E_i) del DEC, y la señal de dato del DEMUX (D) al Enable del DEC (E)



D	C_1	C_0				
E	E_1	E_0	S_3	S_2	S_1	S_0
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

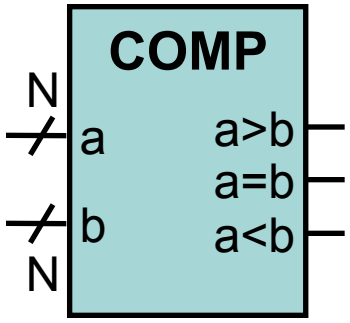
Utilidad de multiplexores y demultiplexores

- Transmisión serie multiplexada



5. Comparadores

- Definición: circuito que permite determinar si dos datos son iguales, o si uno es mayor que otro.
- N es el número de bits de los datos



Comparador 1-bit

a	b	a=b	a>b	a<b
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

$$f_{a=b} = \overline{a \oplus b}$$

$$f_{a>b} = a\bar{b}$$

$$f_{a<b} = \bar{a}b$$

Comparadores

- Comparador 3-bit

$$f_{a=b} = \overline{(a_2 \oplus b_2)} \cdot \overline{(a_1 \oplus b_1)} \cdot \overline{(a_0 \oplus b_0)}$$

$a_2=b_2$ $a_1=b_1$ $a_0=b_0$

$$f_{a>b} = a_2 \overline{b_2} +$$ $a_2>b_2$

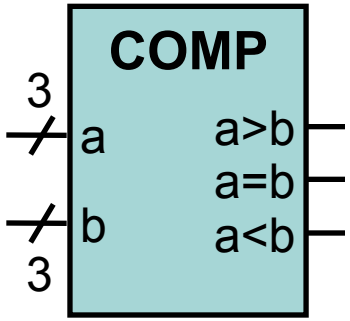
$$+ \overline{(a_2 \oplus b_2)} \cdot a_1 \overline{b_1} +$$ $a_2=b_2$ y $a_1>b_1$

$$+ \overline{(a_2 \oplus b_2)} \cdot \overline{(a_1 \oplus b_1)} \cdot a_0 \overline{b_0}$$ $a_2=b_2$, $a_1=b_1$ y $a_0>b_0$

$$f_{a>b} = \overline{a_2} b_2 +$$ $a_2<b_2$

$$+ \overline{(a_2 \oplus b_2)} \cdot \overline{a_1} b_1 +$$ $a_2=b_2$ y $a_1<b_1$

$$+ \overline{(a_2 \oplus b_2)} \cdot \overline{(a_1 \oplus b_1)} \cdot \overline{a_0} b_0$$ $a_2=b_2$, $a_1=b_1$ y $a_0<b_0$



- Se puede generalizar
- De este modo se reutilizan muchas puertas (XOR)

Bibliografía

- “Circuitos y Sistemas Digitales”. J. E. García Sánchez, D. G. Tomás, M. Martínez Iniesta. Ed. Tebar-Flores
- “Electrónica Digital”, L. Cuesta, E. Gil, F. Remiro, McGraw-Hill
- “Fundamentos de Sistemas Digitales”, T.L Floyd, Prentice-Hall