

Circuitos Combinacionales

Fundamentos de Computadores
Escuela Politécnica Superior. U.A.M



Índice de la Unidad 2

U2. Circuitos combinacionales

U2.1. Implementación de la lógica combinacional. Funciones lógicas.

U2.2. Circuitos combinacionales básicos

U2.2.1. Decodificador.

U2.2.2. Multiplexor y Demultiplexor.

U2.2.3. Codificador de prioridad.

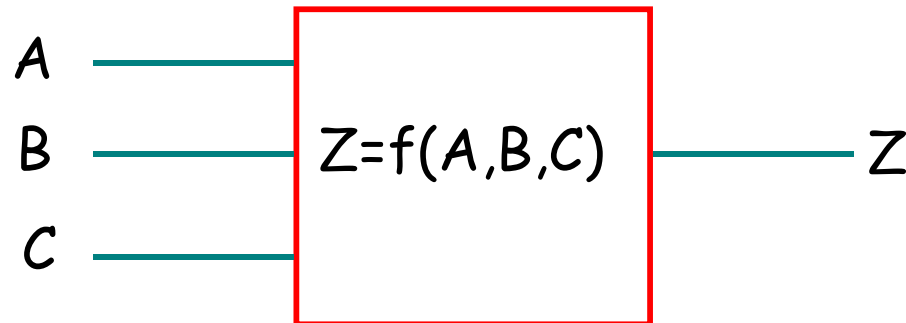
U2.2.4. Conversor de código.

U2.2.5. Comparador de bits.

U2.3. Uso de decodificadores y multiplexores como generadores de funciones

Circuito combinacional

- Circuito sin memoria, las salidas son función de una o varias variables de entrada
- Para una misma combinación de variables a la entrada siempre se obtiene (tras un cierto retraso) la misma combinación de variables a la salida



Pasos para resolver cualquier función lógica

- Representar la tabla de verdad de la función
- Simplificar la función a su expresión más sencilla (Diagrama de Karnaugh).
- Construir el circuito combinacional con el número menor de puertas.

Ejemplo 1: Sea un circuito que tiene 4 entradas que representan un número binario (ABCD). El circuito dispone de una salida Z_1 que debe activarse ($Z_1=1$) cuando el número binario sea múltiplo de 3, y una salida Z_2 que debe activarse cuando el número binario sea múltiplo de 2.

Pasos para resolver cualquier función lógica

- Ejemplo 1:

Nº	A	B	C	D	Z ₁	Z ₂
0	0	0	0	0	1	1
1	0	0	0	1	0	0
2	0	0	1	0	0	1
3	0	0	1	1	1	0
4	0	1	0	0	0	1
5	0	1	0	1	0	0
6	0	1	1	0	1	1
7	0	1	1	1	0	0
8	1	0	0	0	0	1
9	1	0	0	1	1	0
10	1	0	1	0	0	1
11	1	0	1	1	0	0
12	1	1	0	0	1	1
13	1	1	0	1	0	0
14	1	1	1	0	0	1
15	1	1	1	1	1	0

AB \ CD	00	01	11	10
00	1		1	
01				1
11	1		1	
10		1		

$$Z_1 = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD$$

AB \ CD	00	01	11	10
00	1			1
01	1			1
11	1			1
10	1			1

$$Z_2 = \overline{D}$$

Pasos para resolver cualquier función lógica

- Funciones incompletamente especificadas
 - Funciones que no tienen definidos todos sus estados.
 - Los estados no definidos se representan como "X" a la salida.
 - A la hora de resolver la tabla de Karnaugh las "X" se toman como 0 o 1 según convenga.

Pasos para resolver cualquier función lógica

- Funciones incompletamente especificadas
 - Ejemplo2: Ejemplo1 pero definiendo sólo de 0 a 9

N°	A	B	C	D	Z ₁	Z ₂
0	0	0	0	0	1	1
1	0	0	0	1	0	0
2	0	0	1	0	0	1
3	0	0	1	1	1	0
4	0	1	0	0	0	1
5	0	1	0	1	0	0
6	0	1	1	0	1	1
7	0	1	1	1	0	0
8	1	0	0	0	0	1
9	1	0	0	1	1	0
10	1	0	1	0	X	X
11	1	0	1	1	X	X
12	1	1	0	0	X	X
13	1	1	0	1	X	X
14	1	1	1	0	X	X
15	1	1	1	1	X	X

AB \ CD	00	01	11	10
00	1		1	
01				1
11	X	X	X	X
10		1	X	X

$$Z_1 = \overline{A}BC\overline{D} + \overline{B}CD + B\overline{C}\overline{D} + AD$$

AB \ CD	00	01	11	10
00	1			1
01	1			1
11	X	X	X	X
10	1		X	X

$$Z_2 = \overline{D}$$

Circuitos combinatoriales básicos

- **Decodificador ($n-2^n$):** n entradas y 2^n salidas (sólo una activa).
- **Codificador (2^n-n):** n entradas (una o varias activas) y $\lg_2 n$ salidas.
- **Multiplexor ($n-1$):** n entradas, 1 salida y $\lg_2 n$ señales de control.
- **Demultiplexor ($1-n$):** 1 entrada, n salidas y $\lg_2 n$ señales de control.
- **Convertor de código ($n-m$):** n entradas y m salidas, sin relación entre ellas.
- **Otros C. combinatoriales:**
 - Comparadores
 - Semisumadores y Sumadores

Decodificador

- Circuito combinatorial con n entradas y 2^n salidas
- Activa una única línea de salida para cada combinación de las líneas de entrada
- Decodificador 2-4. 2 entradas y $2^2 = 4$ salidas
 - Tabla de la verdad y ecuaciones:

A_1	A_0	O_3	O_2	O_1	O_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



$$O_0 = \overline{A_1} \cdot \overline{A_0}$$

$$O_1 = \overline{A_1} \cdot A_0$$

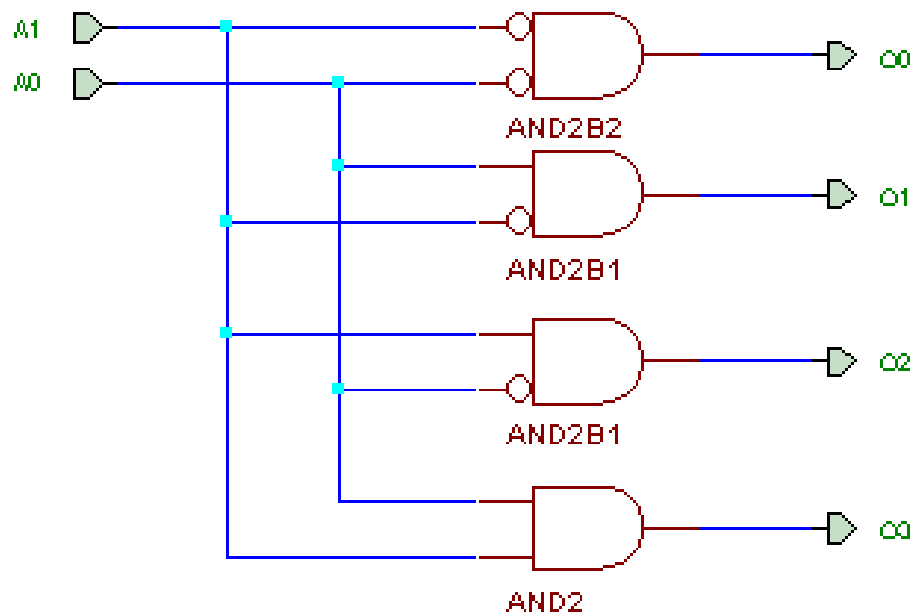
$$O_2 = A_1 \cdot \overline{A_0}$$

$$O_3 = A_1 \cdot A_0$$

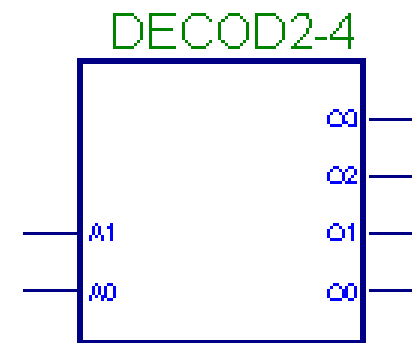
Decodificador

- Decodificador 2-4. Circuito lógico:

Circuito esquemático



Esquema de bloque



Decodificador

- Decodificador 2-4 con entrada de habilitación (Enable)
 - **Enable activado:** El decodificador funciona normalmente
 - **Enable desactivado:** Ninguna salida activa. Circuito “inhabilitado”
 - **Tabla de la verdad y ecuaciones:**

E	A ₁	A ₀	O ₃	O ₂	O ₁	O ₀
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	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

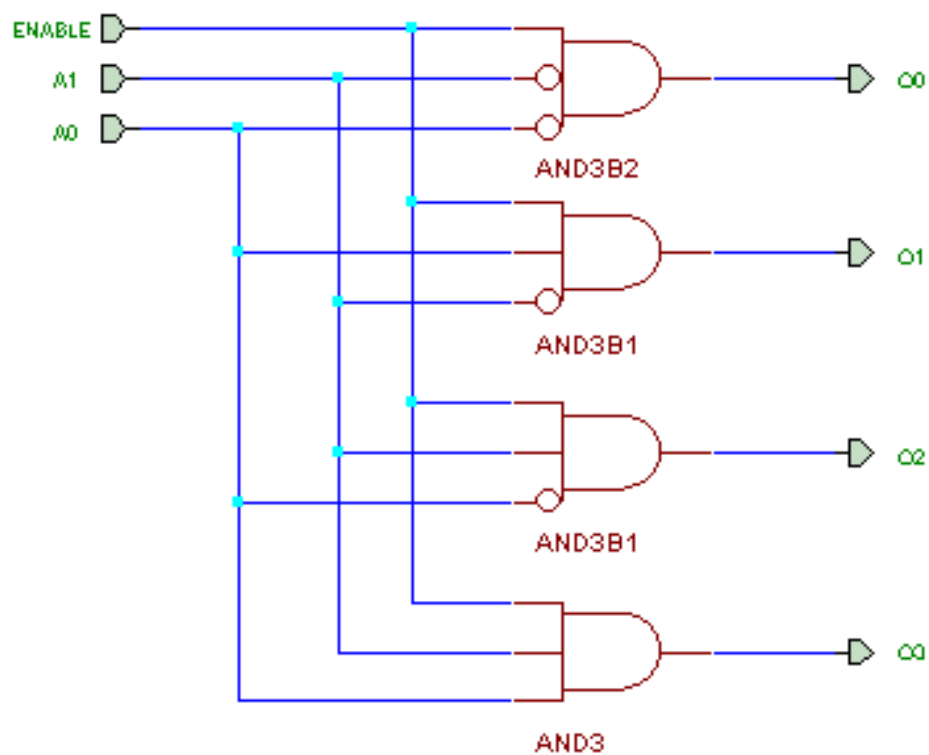


$$\begin{aligned}O_0 &= E \cdot \overline{A_1} \cdot \overline{A_0} \\O_1 &= E \cdot \overline{A_1} \cdot A_0 \\O_2 &= E \cdot A_1 \cdot \overline{A_0} \\O_3 &= E \cdot A_1 \cdot A_0\end{aligned}$$

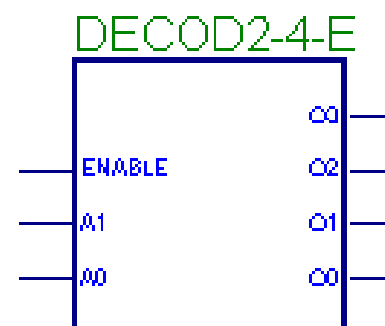
Decodificador

- Decodificador 2-4 con habilitación. **Circuito lógico:**

Circuito esquemático



Esquema de bloque



Decodificador

- Decodificador 3-8. 3 entradas y $2^3 = 8$ salidas

– Tabla de la verdad y ecuaciones

A_2	A_1	A_0	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

$$O_0 = m_0 = \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$$

$$O_1 = m_1 = \overline{A_2} \cdot \overline{A_1} \cdot A_0$$

$$O_2 = m_2 = \overline{A_2} \cdot A_1 \cdot \overline{A_0}$$

$$O_3 = m_3 = \overline{A_2} \cdot A_1 \cdot A_0$$

⋮

$$O_7 = m_7 = A_2 \cdot A_1 \cdot A_0$$

Codificador

- Circuito combinacional con n entradas y $\log_2 n$ salidas
 - **Codificador Elemental:** Para una única línea de entrada activa, codifica a la salida un código que la identifica (número de la entrada).
 - **Codificador de Prioridad:** Dadas varias líneas de entrada activas, codifica a la salida un código (número de la entrada), que identifica a la más prioritaria.

Codificador

- Codificador de prioridad 8-3 con Enable
 - Entre varias entradas activas, se asigna la mayor prioridad a la entrada con índice más alto.

Tabla de la verdad

E	I ₇	I ₆	I ₅	I ₄	I ₃	I ₂	I ₁	I ₀	A ₂	A ₁	A ₀
0	X	X	X	X	X	X	X	X	0	0	0
1	1	X	X	X	X	X	X	X	1	1	1
1	0	1	X	X	X	X	X	X	1	1	0
1	0	0	1	X	X	X	X	X	1	0	1
1	0	0	0	1	X	X	X	X	1	0	0
1	0	0	0	0	1	X	X	X	0	1	1
1	0	0	0	0	0	1	X	X	0	1	0
1	0	0	0	0	0	0	1	X	0	0	1
1	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0

Codificador

- Codificador de prioridad 8-3 con Enable

Ecuaciones:

$$A_0 = (I_7 + \overline{I_7} \overline{I_6} I_5 + \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} I_3 + \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} \overline{I_3} I_2 I_1) E$$

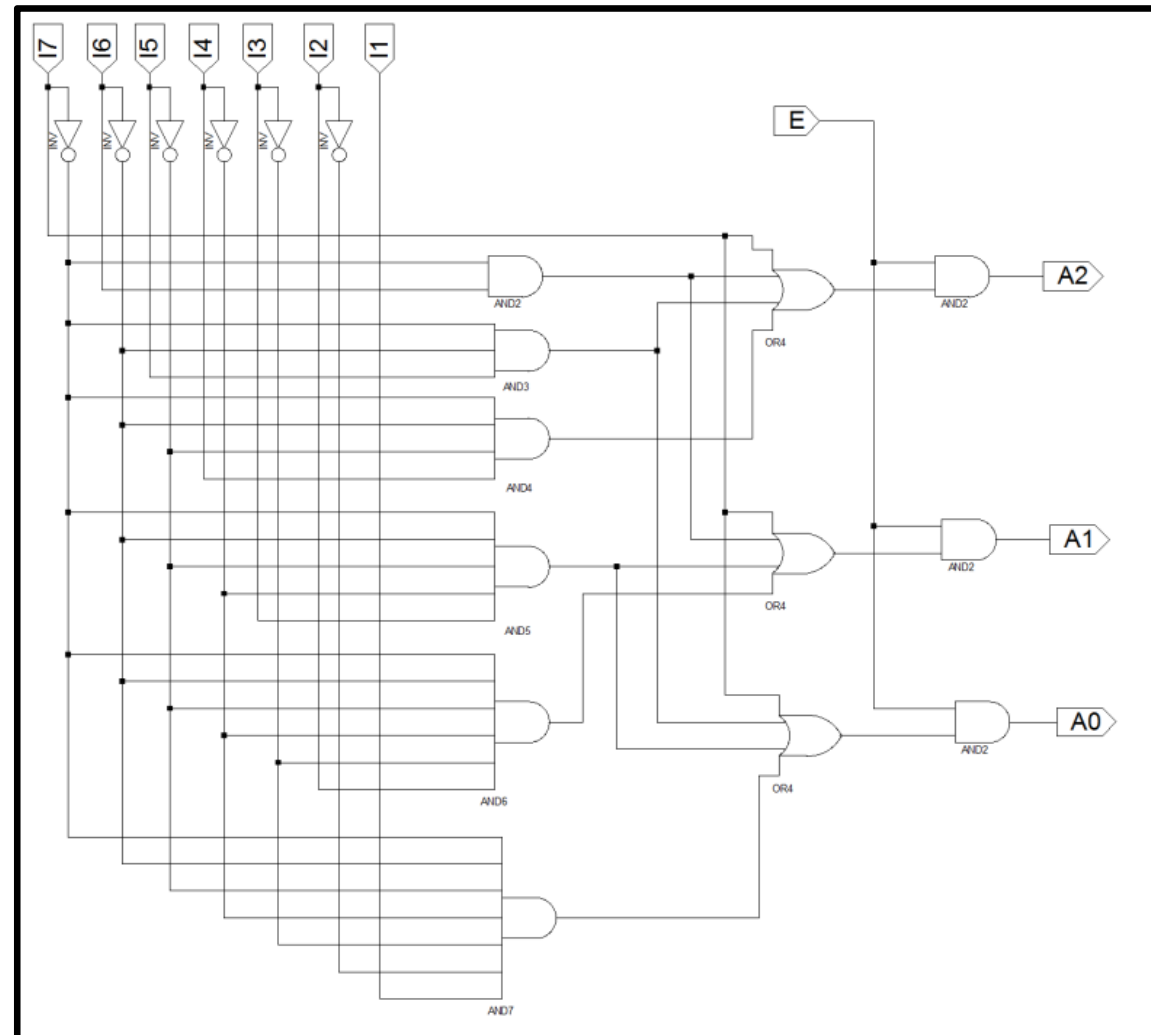
$$A_1 = (I_7 + \overline{I_7} I_6 + \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} I_3 + \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} \overline{I_3} I_2) E$$

$$A_2 = (I_7 + \overline{I_7} I_6 + \overline{I_7} \overline{I_6} I_5 + \overline{I_7} \overline{I_6} \overline{I_5} I_4) E$$

Codificador

- Codificador de prioridad 8-3 con Enable

Circuito:



Multiplexor

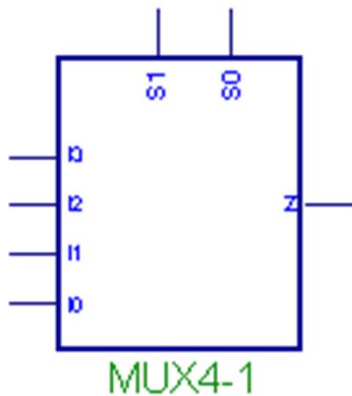
- Realiza la transmisión de datos desde una entrada seleccionable hacia una salida única
- Se caracteriza por tener n líneas de control que seleccionan (multiplexan) una de las 2^n líneas de entrada y la transmiten a la salida
- Cada combinación de las líneas de control activa una puerta
- Hay dos tipos de entradas:
 - Entradas de datos
 - Entradas de control

Multiplexor

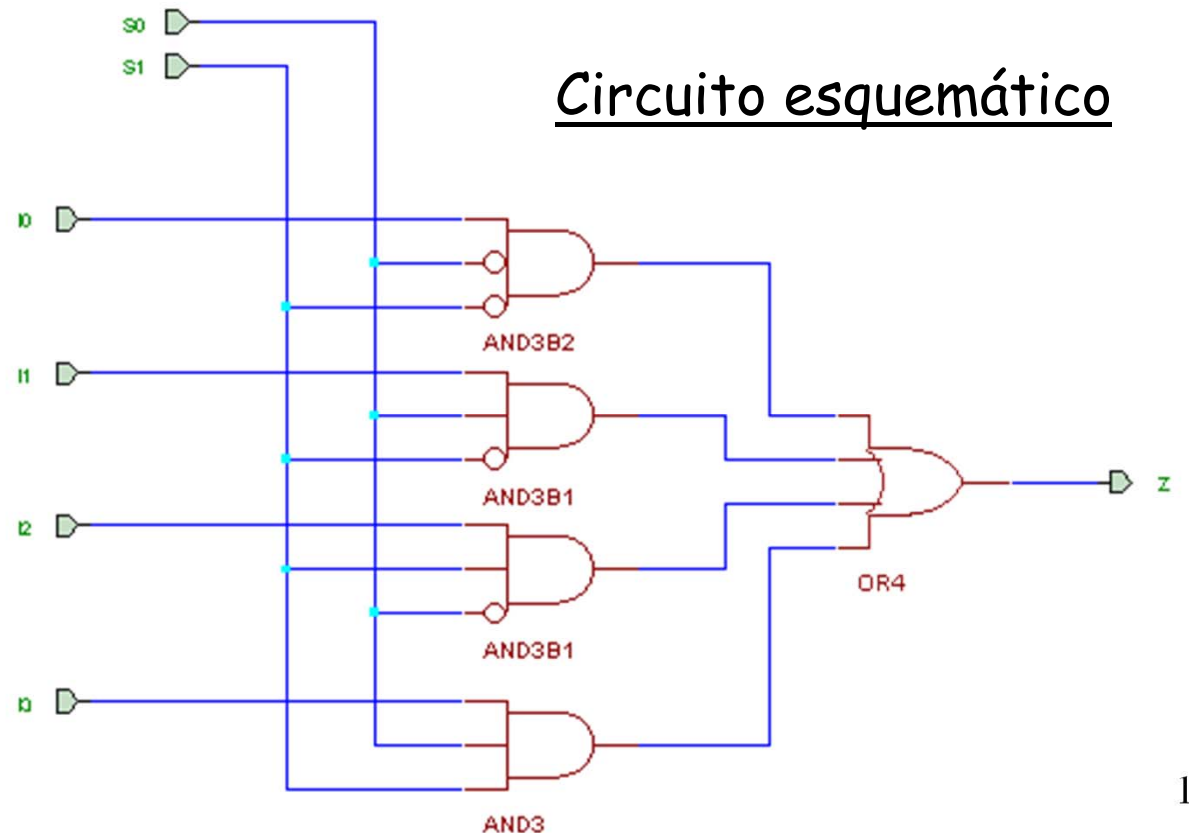
- Multiplexor 4-1. 4 entradas (datos), 2 entradas (control) y 1 salida
 - Tabla de la verdad, ecuación y circuito:

S_1	S_0	Salida (Z)
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$$Z = S_1 S_0 I_3 + S_1 \overline{S_0} I_2 + \overline{S_1} S_0 I_1 + \overline{S_1} \overline{S_0} I_0$$



Esquema de bloque



Multiplexor

- Multiplexor 8-1 con entrada de habilitación. 8 entradas (datos), 4 entradas (control) y 1 salida

Tabla de la verdad y ecuación

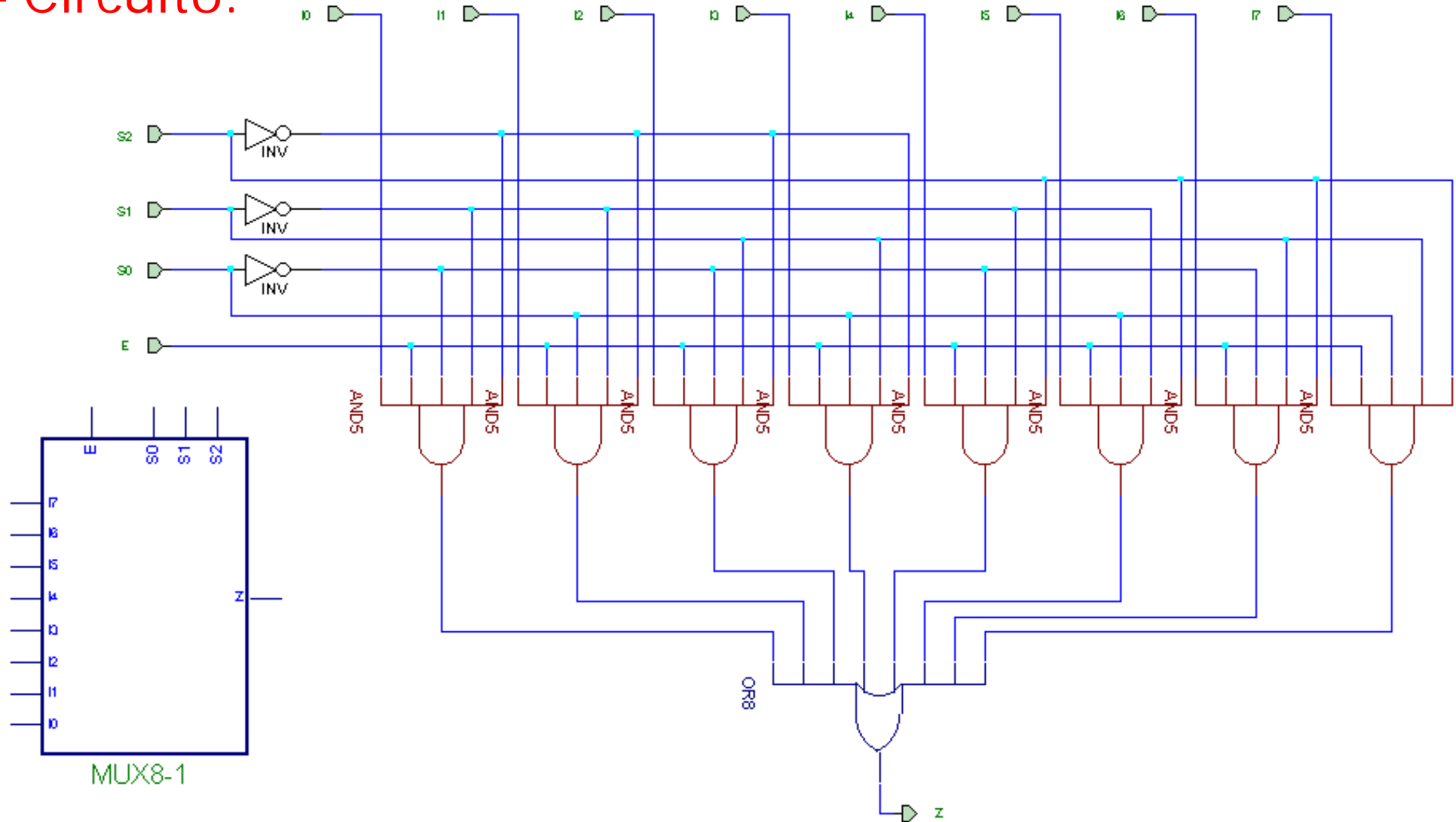
E	S ₂	S ₁	S ₀	Z
0	X	X	X	0
1	0	0	0	I ₀
1	0	0	1	I ₁
1	0	1	0	I ₂
1	0	1	1	I ₃
1	1	0	0	I ₄
1	1	0	1	I ₅
1	1	1	0	I ₆
1	1	1	1	I ₇

$$Z = (S_2 S_1 S_0 I_7 + S_2 S_1 \overline{S_0} I_6 + S_2 \overline{S_1} S_0 I_5 + S_2 \overline{S_1} \overline{S_0} I_4 + \overline{S_2} S_1 S_0 I_3 + \overline{S_2} S_1 \overline{S_0} I_2 + \overline{S_2} \overline{S_1} S_0 I_1 + \overline{S_2} \overline{S_1} \overline{S_0} I_0) E$$

Multiplexor

- Multiplexor 8-1 con entrada de habilitación

– Circuito:



Demultiplexor

- Realizan la función inversa del multiplexor
- Puede seleccionarse la transmisión de un dato desde una única línea de entrada hacia una de las posibles líneas de salida
- Utilizando n líneas de control, transmite (demultiplexa) la información de su única línea de entrada por cualquiera de sus 2^n líneas de salida
- Es lo mismo que un decodificador con enable, siendo la línea de datos equivalente al enable

Demultiplexor

- Demultiplexor 1-4. 1 entrada (dato), 2 entradas (control) y 4 salidas
 - Tabla de la verdad y ecuaciones:

S_1	S_0	O_3	O_2	O_1	O_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0

$$O_0 = \overline{S_1} \cdot \overline{S_0} \cdot I$$

$$O_1 = \overline{S_1} \cdot S_0 \cdot I$$

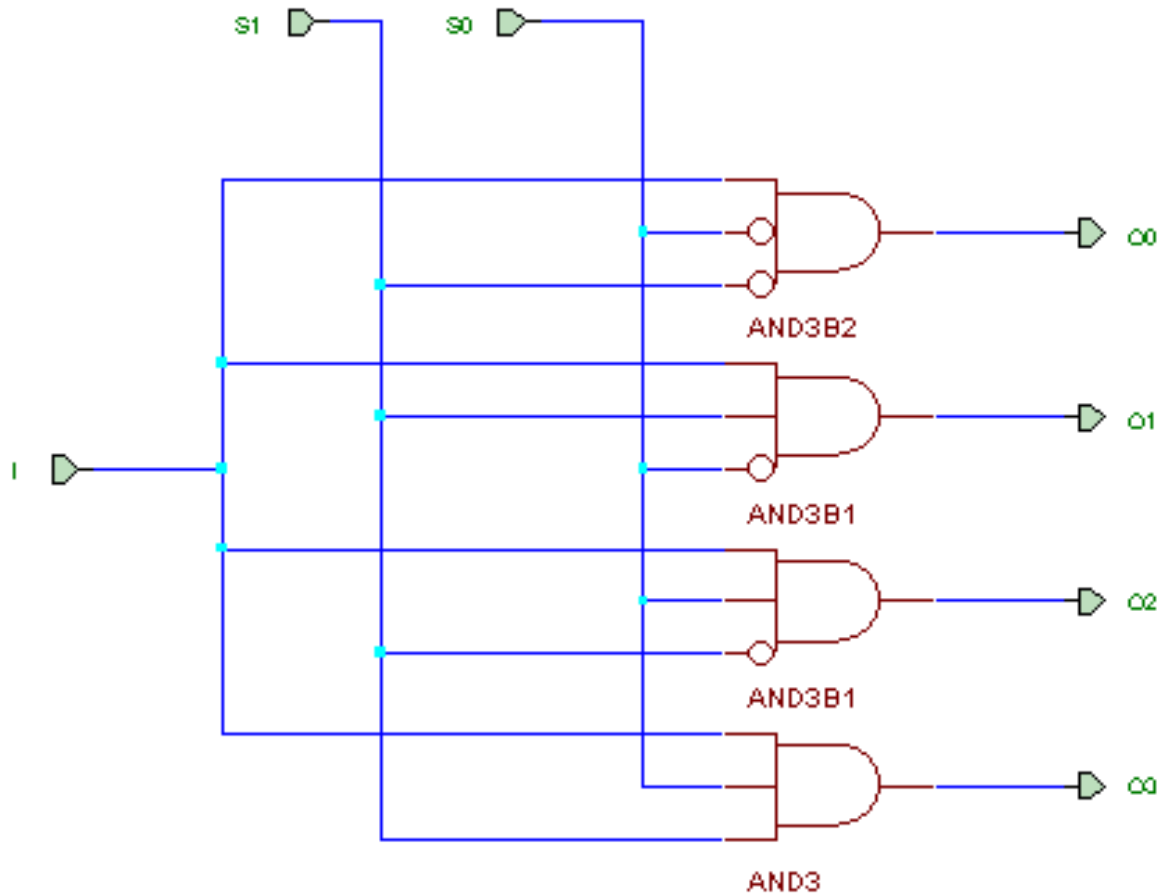
$$O_2 = S_1 \cdot \overline{S_0} \cdot I$$

$$O_3 = S_1 \cdot S_0 \cdot I$$

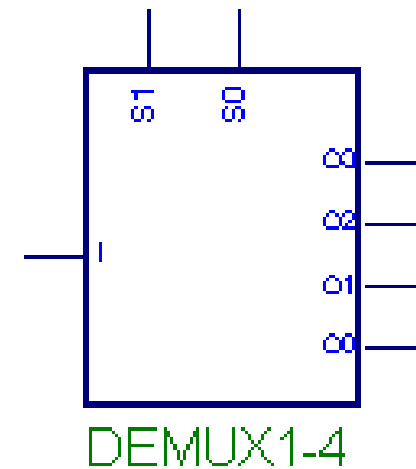
Demultiplexor

- Demultiplexor 1-4. **Circuito:**

Circuito esquemático



Esquema de bloque



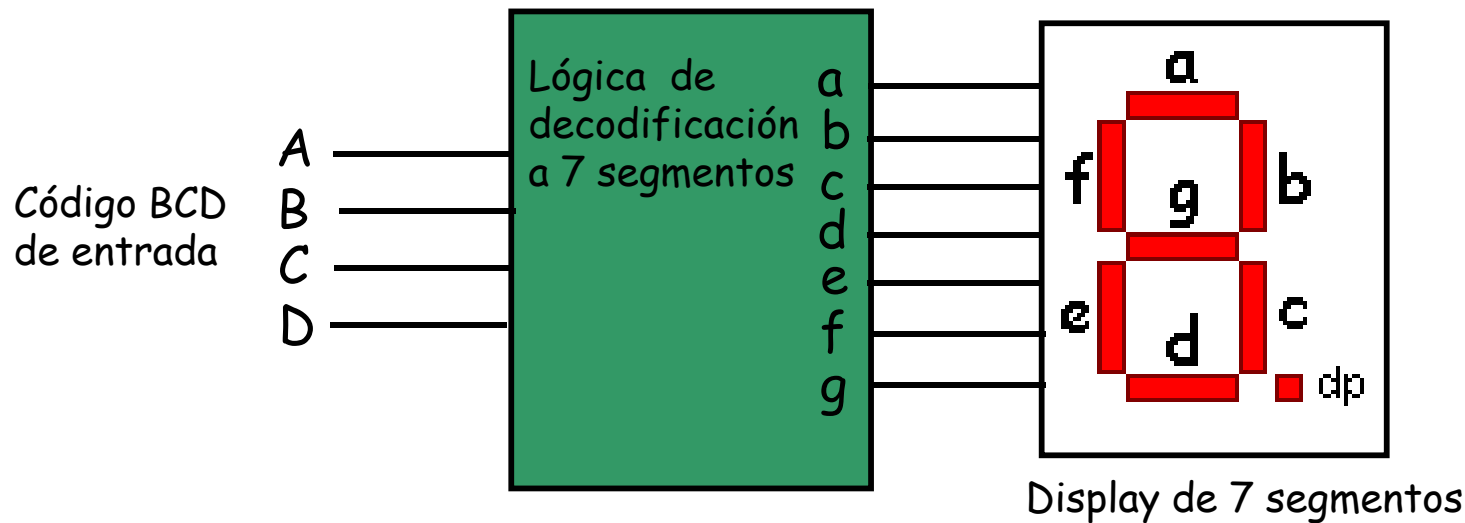
Convertor de código

- Dada una palabra de n bits a la entrada se convierte o traduce a otra palabra de m bits a la salida.
- No existe una relación entre el número de líneas de entrada y de salida.
- Ambas palabras representan la misma información en distintos códigos.

Convertor de código

- Convertor BCD a 7 segmentos

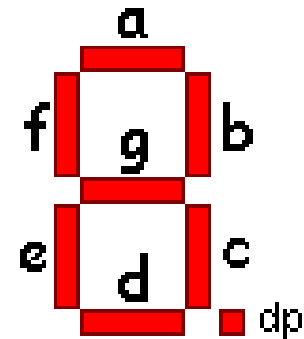
- Acepta código BCD (0..9) en sus 4 entradas y proporciona 7 salidas capaces de excitar un *display* de 7 segmentos que indican el dígito decimal de la entrada.
- Diagrama de bloques de la lógica:



Conversor de código

- Conversor BCD a 7 segmentos
 - Segmentos activados para cada dígito decimal

Dígito	Segmentos activados
0	a,b,c,d,e,f
1	b,c
2	a,b,d,e,g
3	a,b,c,d,g
4	b,c,f,g
5	a,c,d,f,g
6	a,c,d,e,f,g
7	a,b,c
8	a,b,c,d,e,f,g
9	a,b,c,d,f,g



Convertor de código

- Convertor BCD a 7 segmentos

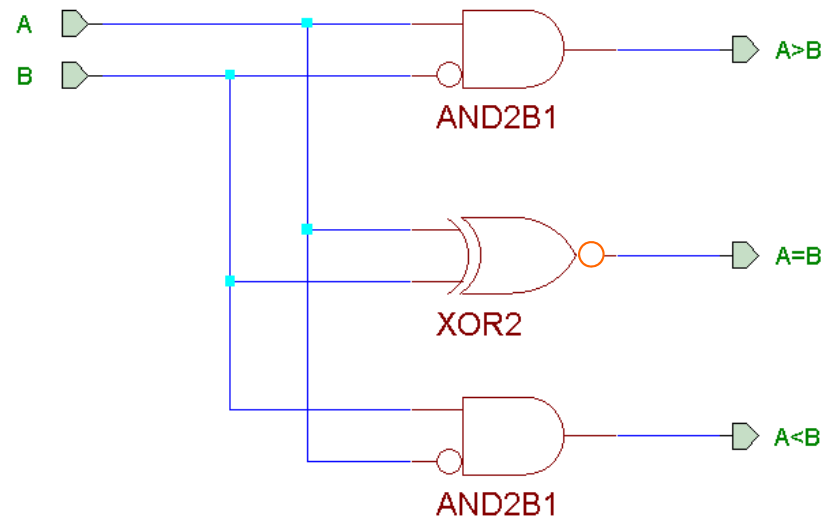
– Tabla de verdad:

Digito Decimal	Entradas				Salidas de segmentos						
	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	1	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	1	0	1	1
10	1	0	1	0	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X

Comparador de bits

- **Comparador de bits:** circuito con 2 entradas y 3 salidas que se utiliza para comparar bits
- Comparador de 2 bits. **Tabla de la verdad, ecuaciones y circuito**

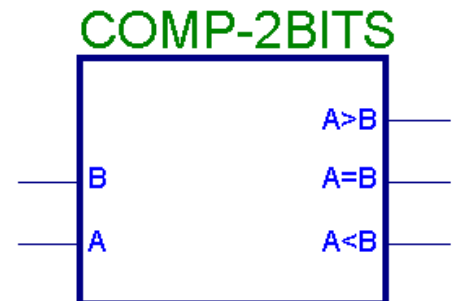
A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0



$$A > B = A\bar{B}$$

$$A = B = \overline{A \oplus B}$$

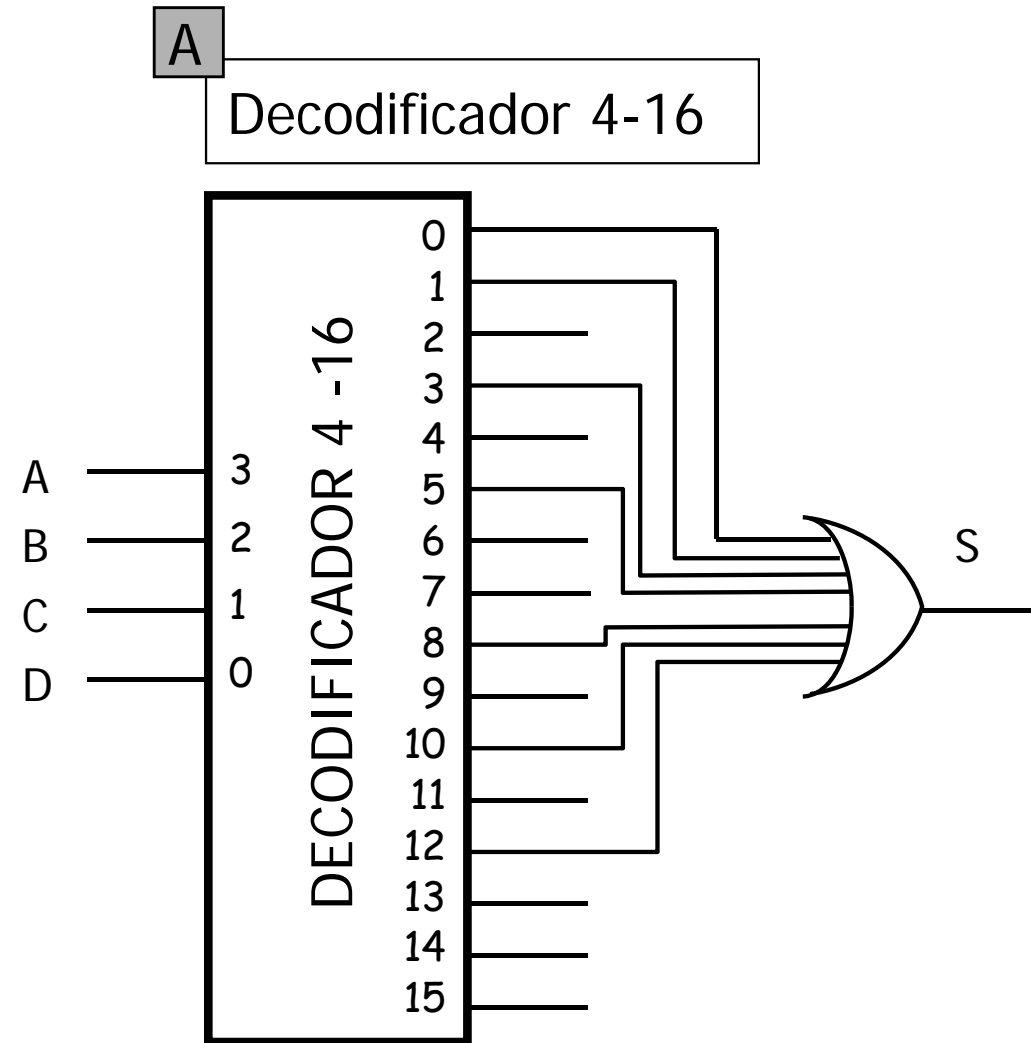
$$A < B = \bar{A}B$$



Funciones lógicas con DEC o MUX

Ejemplo: A) Diseñar F mediante un Decodificador 4-16 y una puerta OR

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	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0



- Asociar las variables de la función lógica a las entradas del decodificador.
- Equivalente a suma de productos.

Funciones lógicas con DEC o MUX

Ejemplo: B) Diseñar F mediante un Multiplexor 8-1

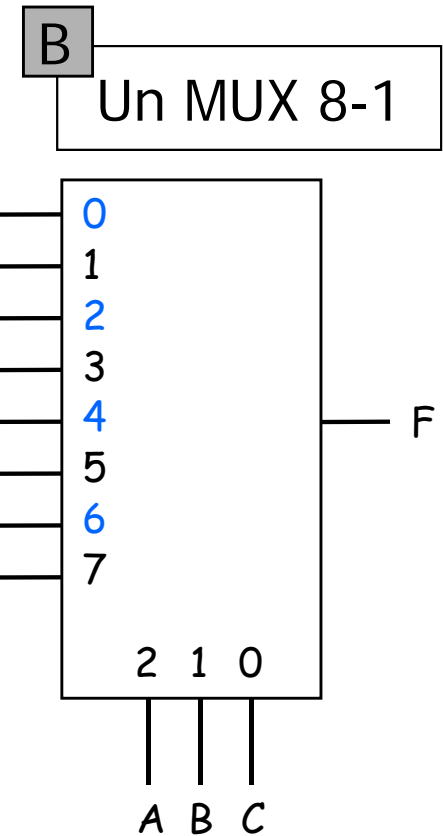
A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Si es 11, se pone 1 en su entrada

Si es 01, se pone D en su entrada

Si es 00, se pone 0 en su entrada

Si es 10, se pone /D en su entrada



1. Asociar variables (A,B,C,D,...) a las entradas de control ($S_n...S_2, S_1, S_0$).
2. Las variables no asociadas al control, forman parte de las entrada del MUX
3. Las variables asociadas al control, NUNCA forman parte las entrada del MUX
4. Algunas entradas al MUX pueden ser '0' ó '1'