FACULTAD DE INGENIERIA
Universidad de Buenos Aires

86.01 Técnica Digital

Decodificadores y Multiplexores

Ing. Jorge H. Fuchs

Introducción

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Objetivos de la clase:

Analizar el concepto de Módulo Universal o bloques lógicos combinacionales básicos.

Estudiar los conceptos de codificación y decodificación como también los circuitos que los llevan a cabo (Encoder y Decoder).

Ver la necesidad de la conversión entre distintos tipos de códigos.

Analizar el concepto de multiplexación y demultiplexación analógica y digital como también las características de los circuitos Mux y Demux digitales.

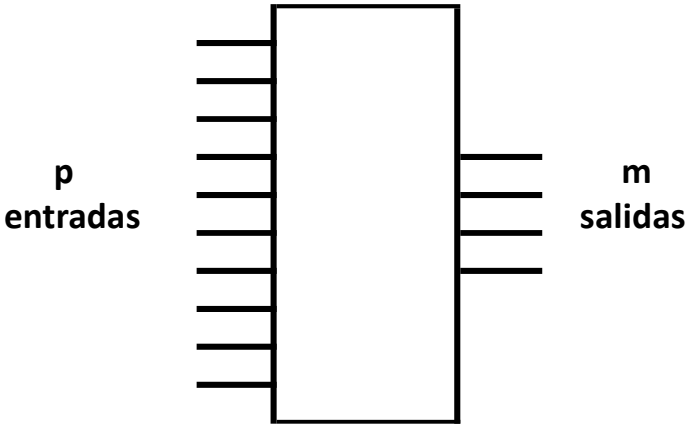
Utilizar los bloques universales (CIs) anteriores para la implementación alternativa de funciones lógicas.

Codificación y decodificación

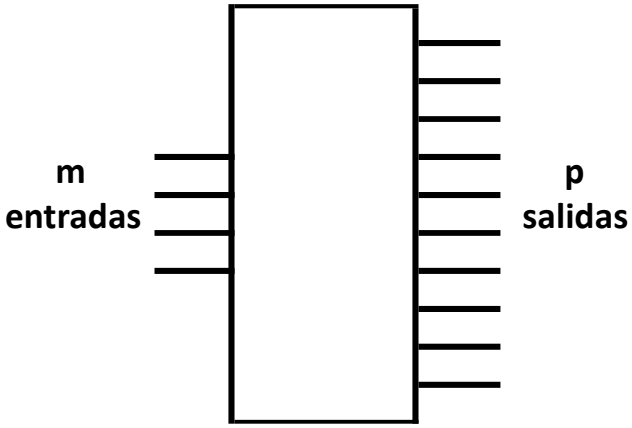
FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Es un proceso en el cual se “compacta” la información para luego de transmitirla y / o almacenarla, volverla a su estado original.
En el área digital tendremos circuitos **combinacionales** que realizarán esas tareas: Codificadores o Encoders y Decodificadores o Decoders.
En general los Codificadores tienen más entradas que salidas mientras que los Decodificadores tienen más salidas que entradas.

Encoder



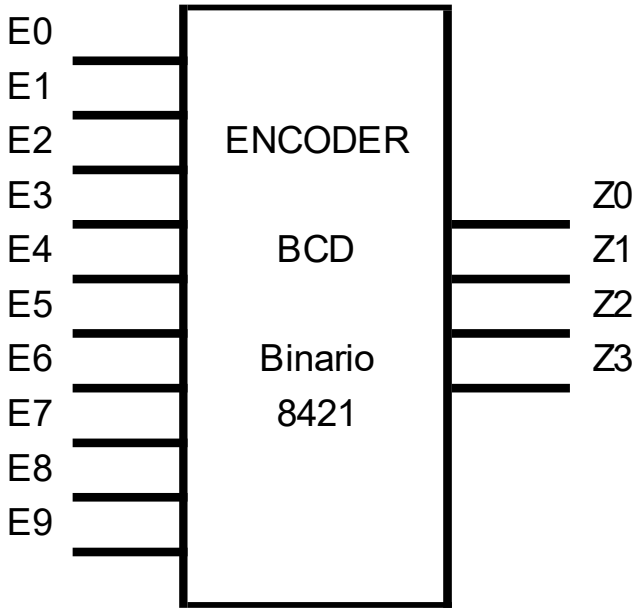
Decoder



Codificador (Encoder)

Es un circuito **combinacional** que asigna un **código** binario de salida único a cada señal de entrada aplicada al dispositivo. Si un codificador tiene **n** entradas, el número de salidas **s** es tal que se debe cumplir la siguiente condición: $\mathbf{2^S \geq n}$. Para los comerciales, salvo para el BCD, se cumple $\mathbf{2^S = n}$ (ej: 4/2, 8/3, 16/4, etc.)

Encoder **BCD** a binario 8421:

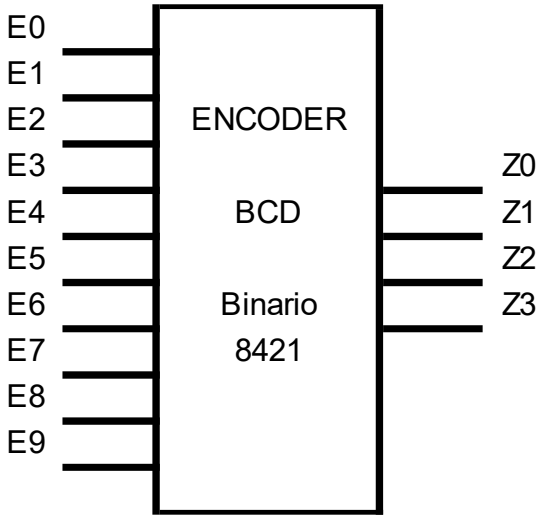


Dígito decimal	Código BCD A_3	A_2	A_1	A_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Codificador (Encoder)

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	Z3	Z2	Z1	Z0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	**1**	0	0	0	0	0	0	0	0	0	0	0	**1**
0	0	**1**	0	0	0	0	0	0	0	0	0	**1**	0
0	0	0	**1**	0	0	0	0	0	0	0	0	**1**	**1**
0	0	0	0	**1**	0	0	0	0	0	0	**1**	0	0
0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	**1**
0	0	0	0	0	0	**1**	0	0	0	0	**1**	**1**	0
0	0	0	0	0	0	0	**1**	0	0	0	**1**	**1**	**1**
0	0	0	0	0	0	0	0	**1**	0	**1**	0	0	0
0	0	0	0	0	0	0	0	0	**1**	**1**	0	0	**1**



$$Z_0 = E_1 + E_3 + E_5 + E_7 + E_9$$

$$Z_1 = E_2 + E_3 + E_6 + E_7$$

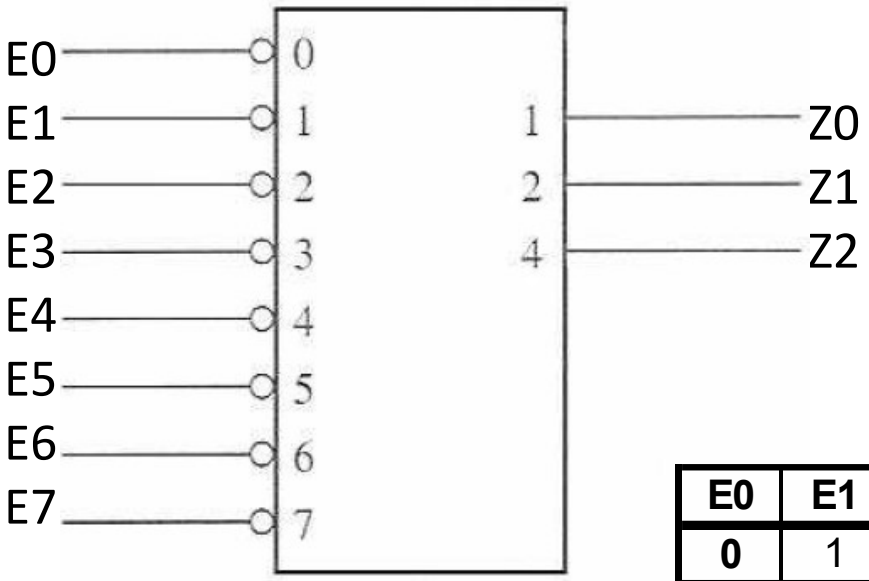
$$Z_2 = E_4 + E_5 + E_6 + E_7$$

$$Z_3 = E_8 + E_9$$

Codificador (Encoder)

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Veamos ahora el caso de un Encoder de "**8 a 3**" con entradas **negadas**:



$$Z_0 = \overline{E_1} + \overline{E_3} + \overline{E_5} + \overline{E_7}$$

$$Z_1 = \overline{E_2} + \overline{E_3} + \overline{E_6} + \overline{E_7}$$

$$Z_2 = \overline{E_4} + \overline{E_5} + \overline{E_6} + \overline{E_7}$$

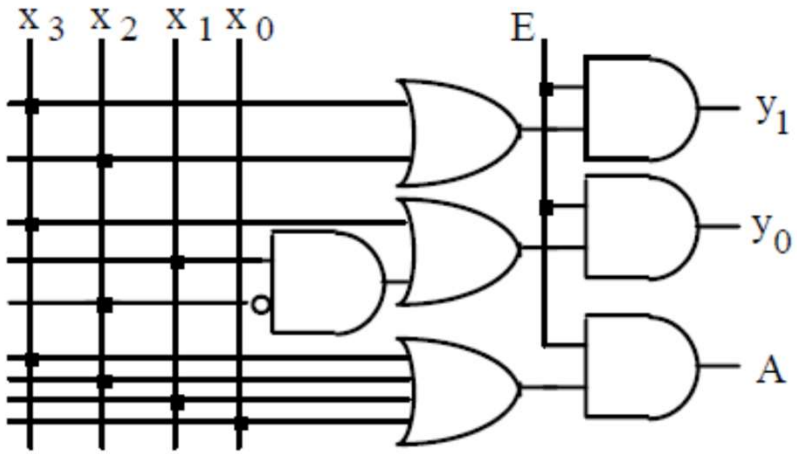
E0	E1	E2	E3	E4	E5	E6	E7	Z2	Z1	Z0
0	1	1	1	1	1	1	1	0	0	0
1	**0**	1	1	1	1	1	1	0	0	**1**
1	1	**0**	1	1	1	1	1	0	**1**	0
1	1	1	**0**	1	1	1	1	0	**1**	**1**
1	1	1	1	**0**	1	1	1	**1**	0	0
1	1	1	1	1	**0**	1	1	**1**	0	**1**
1	1	1	1	1	1	**0**	1	**1**	**1**	0
1	1	1	1	1	1	1	**0**	**1**	**1**	**1**

Codificador (Encoder)

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

En previsión de que haya **más de una entrada activa a la vez**, se puede definir una **prioridad** de entradas, actuando siempre de acuerdo con la entrada activa de prioridad más alta.
Ejemplo de codificador 4 a 2 con prioridad. La entrada E es de habilitación. La salida A en 1 indica que hay alguna entrada activa.

E	x_3	x_2	x_1	x_0	A	y_1	y_0
0	X	X	X	X	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0
1	0	0	1	X	1	0	1
1	0	1	X	X	1	1	0
1	1	X	X	X	1	1	1

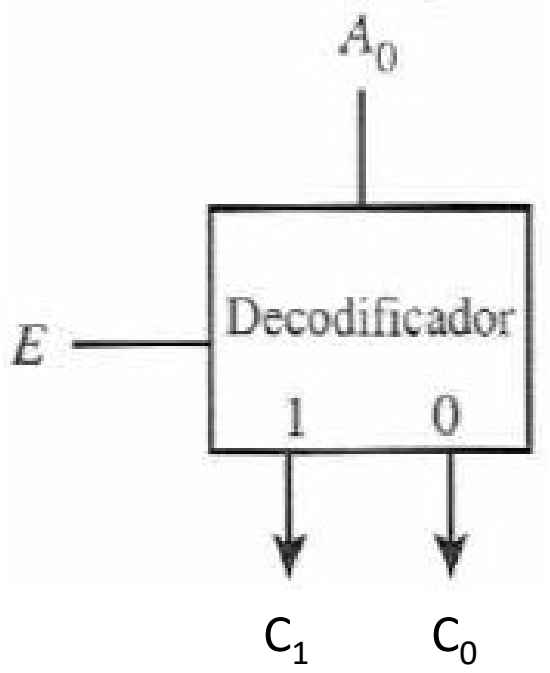


Decodificador (Decoder)

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Es un circuito **combinacional** que para una combinación de valores en sus entradas activa **solo una** de sus **salidas**. Si tiene **s** entradas, el número de salidas **n** es tal que se debe cumplir la siguiente condición: $\mathbf{2^S \geq n}$. Generalmente tienen una entrada de **habilitación E**. Para los comerciales, salvo para el BCD, se cumple $\mathbf{2^S = n}$ (ej: 2/4, 3/8, 4/16, etc.)

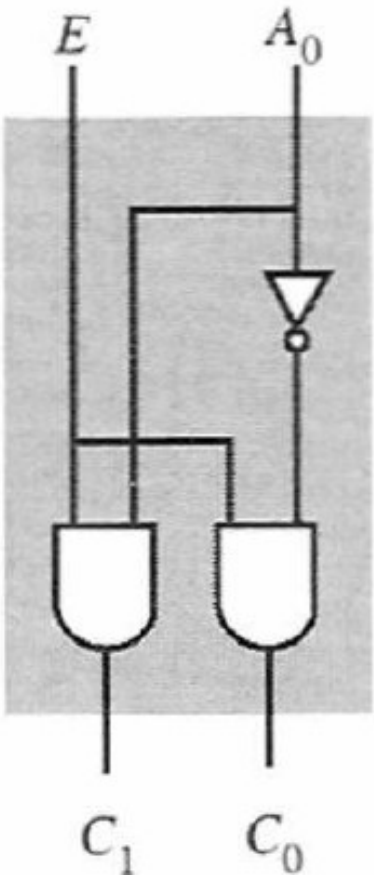
Diseño de un Decoder de **1 entrada** y **2 salidas**:



E	A_0	C_1	C_0
1	0	0	1
1	1	1	0
0	X	0	0

$$C_0 = E \cdot \overline{A}_0$$

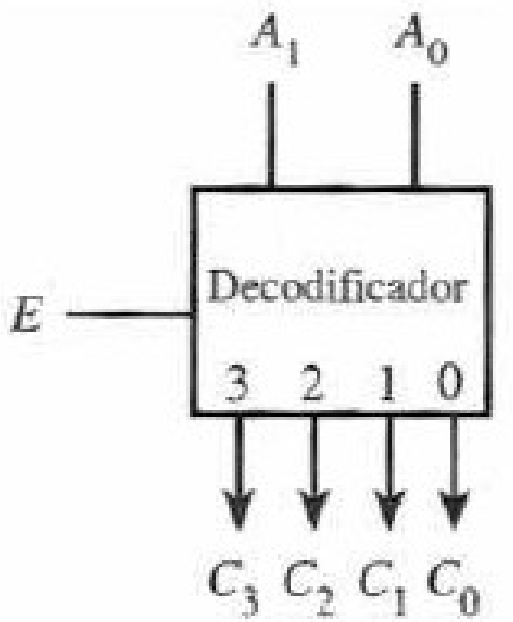
$$C_1 = E \cdot A_0$$



Decodificador (Decoder)

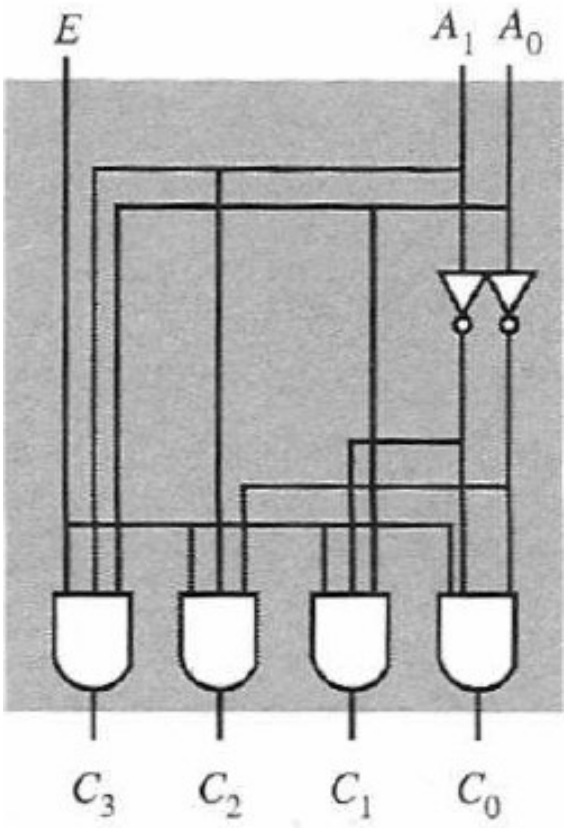
FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Diseño de un Decoder de
2 entradas y **4 salidas**:



E	A_1	A_0	C_3	C_2	C_1	C_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
0	X	X	0	0	0	0

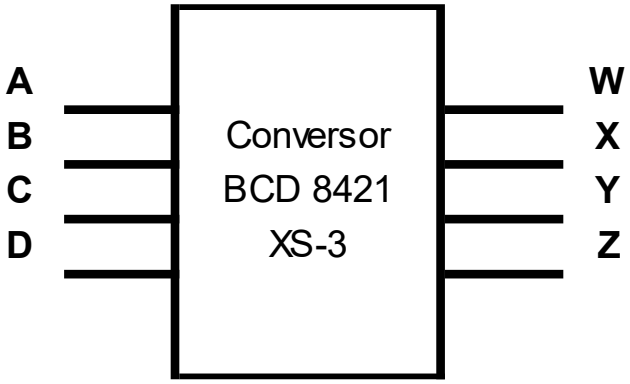
$$C_0 = E \cdot \overline{A}_1 \cdot \overline{A}_0$$
$$C_1 = E \cdot \overline{A}_1 \cdot A_0$$
$$C_2 = E \cdot A_1 \cdot \overline{A}_0$$
$$C_3 = E \cdot A_1 \cdot A_0$$



Conversor de código

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Es un circuito **combinacional** que convierte información de un código a otro.
Por ejemplo del código **BCD 8421** al código **XS-3 8421**.



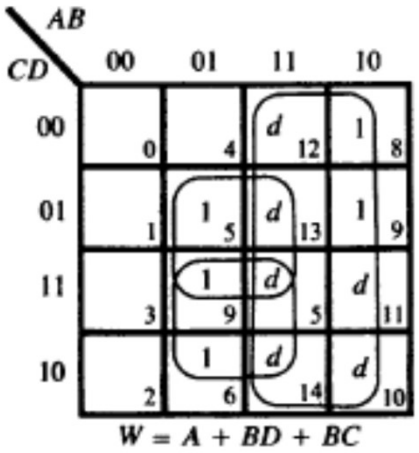
La tabla de verdad será:

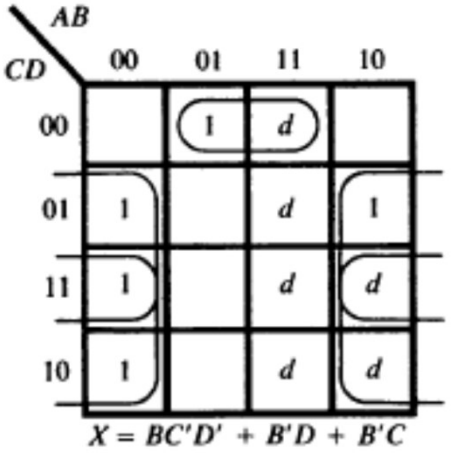
Inputs (BCD) A	B	C	D	Outputs (Excess-3) W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	*d*	*d*	*d*	*d*
1	0	1	1	*d*	*d*	*d*	*d*
1	1	0	0	*d*	*d*	*d*	*d*
1	1	0	1	*d*	*d*	*d*	*d*
1	1	1	0	*d*	*d*	*d*	*d*
1	1	1	1	*d*	*d*	*d*	*d*

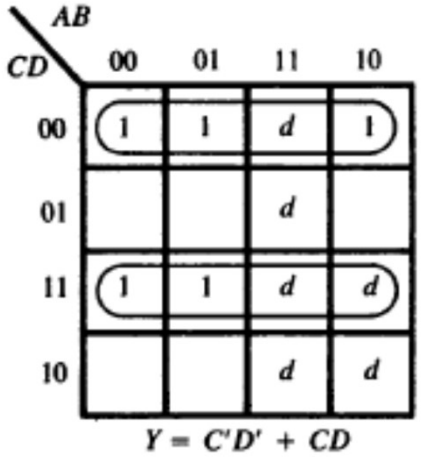
Conversor de código

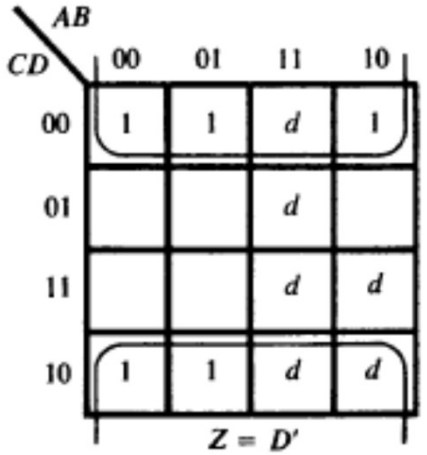
FACULTAD DE INGENIERIA
Universidad de Buenos Aires

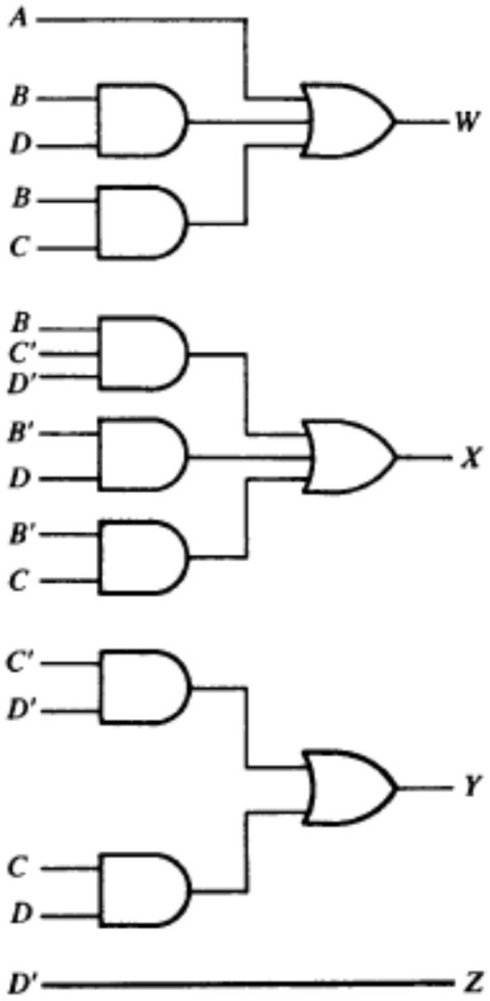
Resolviendo obtenemos el circuito interno del conversor:







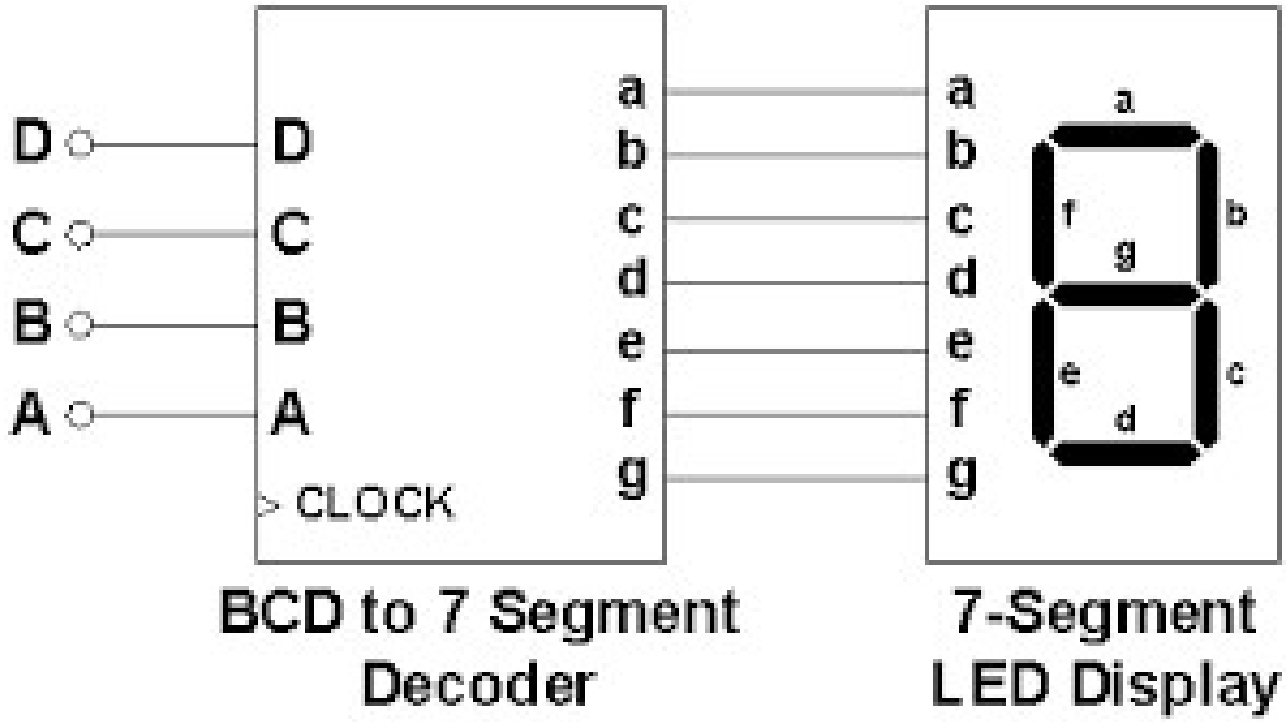




Conversor de código

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Conversor del código **BCD 8421** a **7 segmentos**:

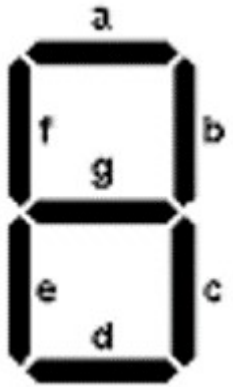


Conversor de código

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

La TV quedaría:

A	B	C	D	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
1	0	1	0	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x



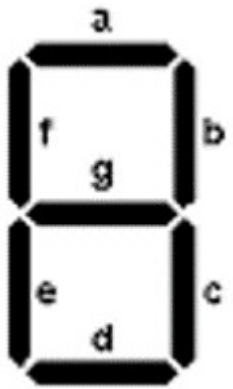
La solución la obtenemos resolviendo los 7 K de 4 variables c/u teniendo en cuenta las x según sea más conveniente para la simplificación.

Conversor de código

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Si queremos que al aparecer una combinación no esperada indique **E** (error):

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

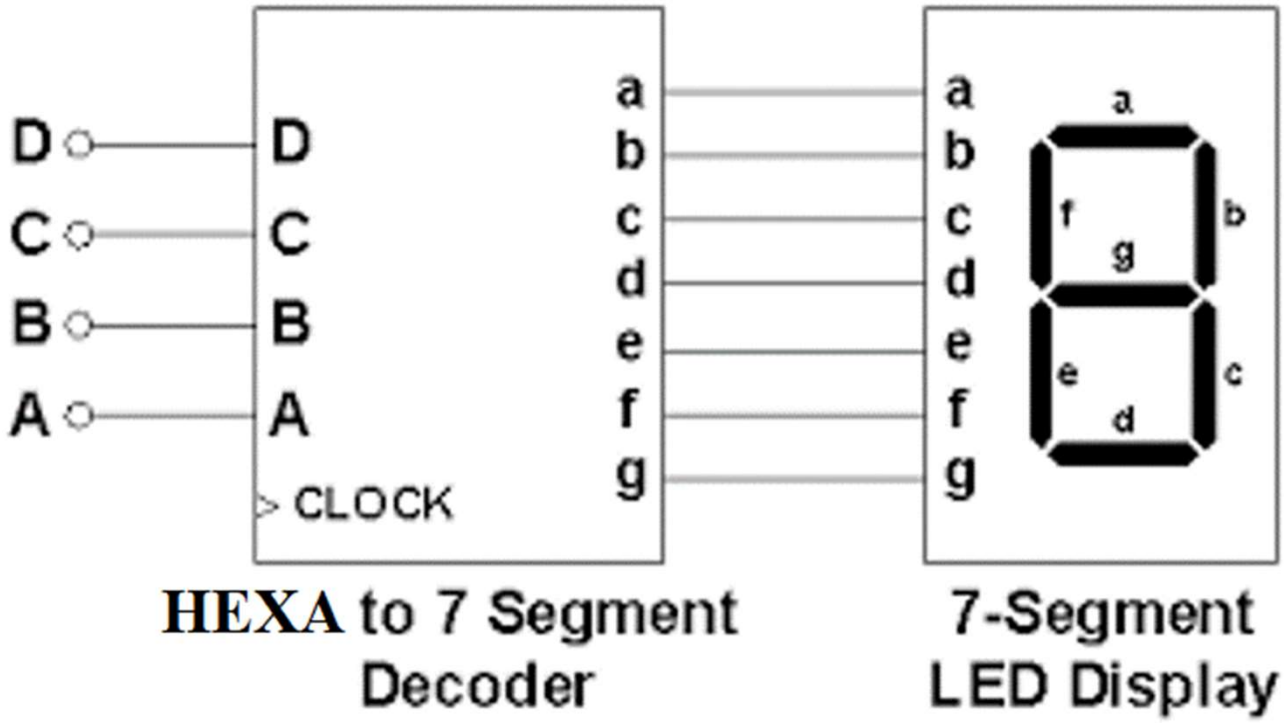


Nuevamente la solución la obtenemos resolviendo los 7 K de 4 variables c/u.

Conversor de código

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Conversor del código **Hexadecimal** a **7 segmentos**:

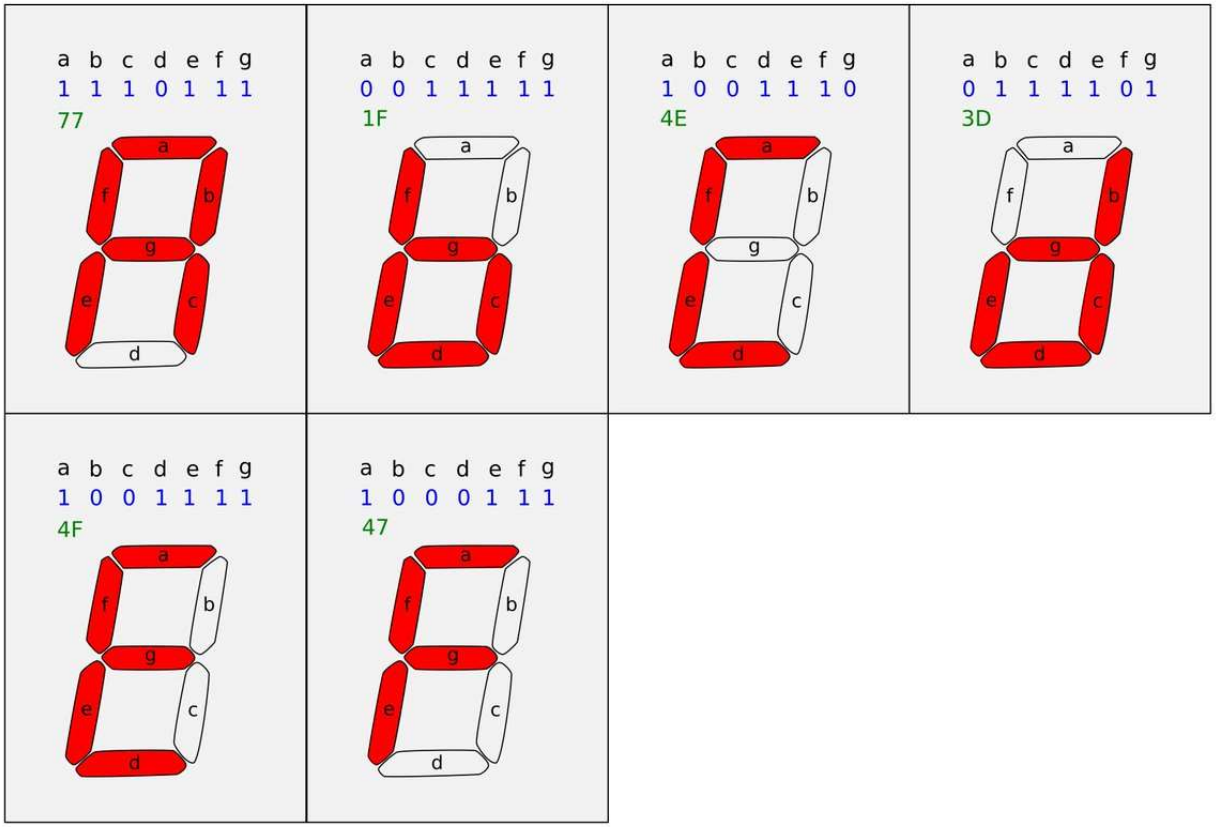


HEXA to 7 Segment Decoder

7-Segment LED Display

Conversor de código

Ahora debemos también representar los 6 casos del **10** al **15** (**A** a **F**):

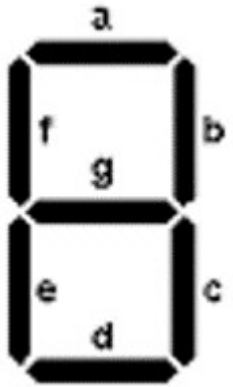


Conversor de código

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Entonces tenemos las 16 combinaciones posibles (sin x):

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



Nuevamente la solución la obtenemos resolviendo los 7 K de 4 variables c/u.

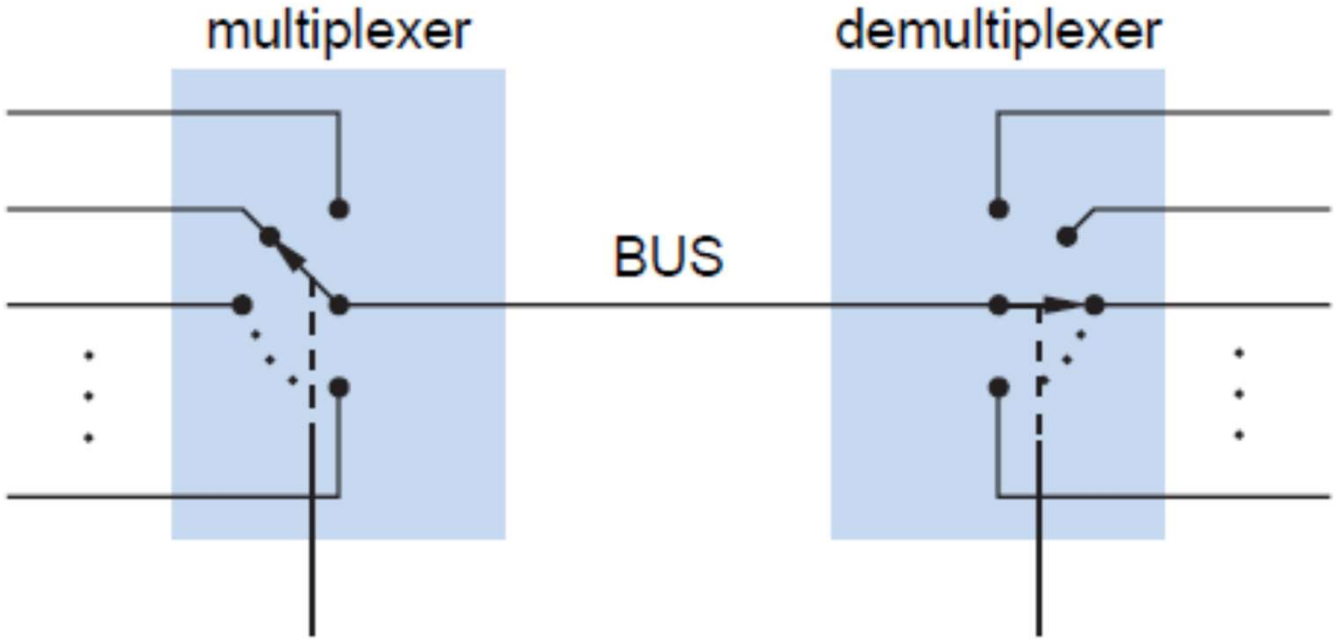
Multiplexación y Demultiplexación

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Un **multiplexor (MUX) analógico** es un circuito que permite seleccionar cual de sus canales de entrada enviará la información que presenta hacia su única salida.

Un **demultiplexor (DEMUX) analógico** es un circuito que permite seleccionar cual de sus salidas recibirá la información que se presenta en su única entrada.

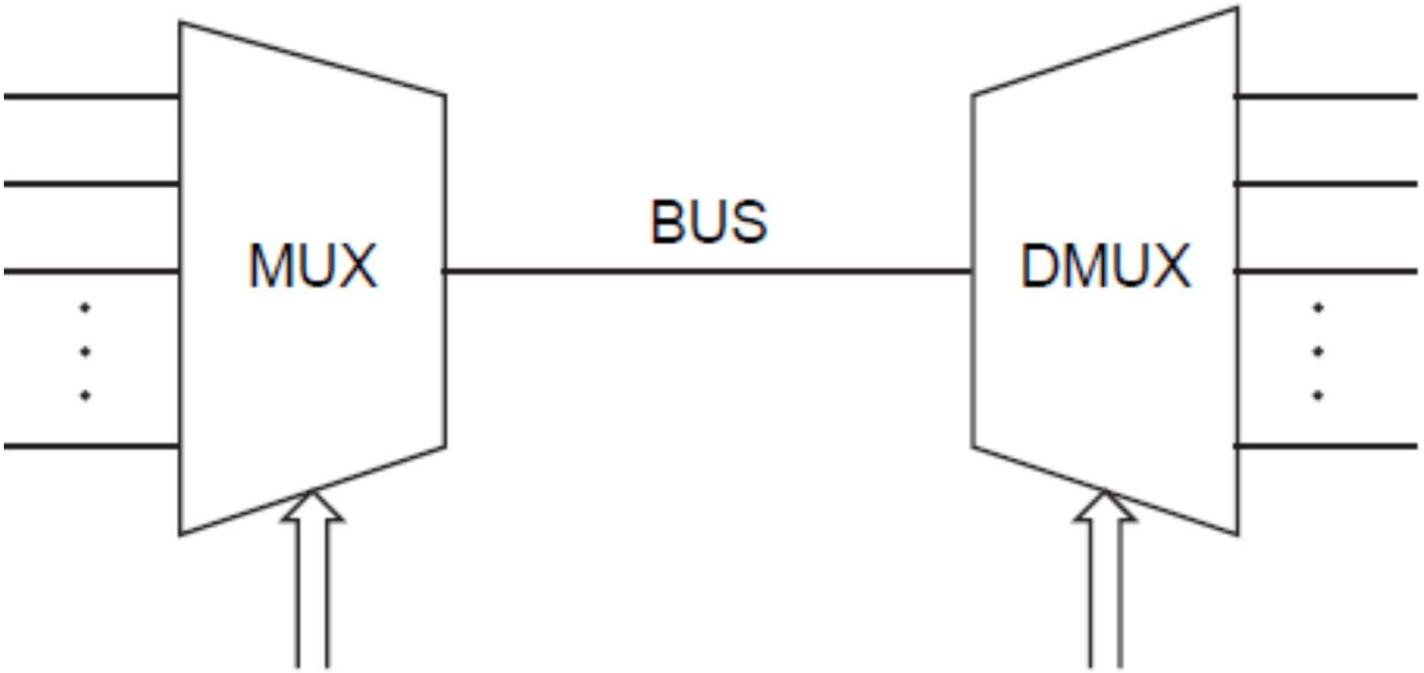
Ambos pueden ser bidireccionales.



Multiplexación y Demultiplexación

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

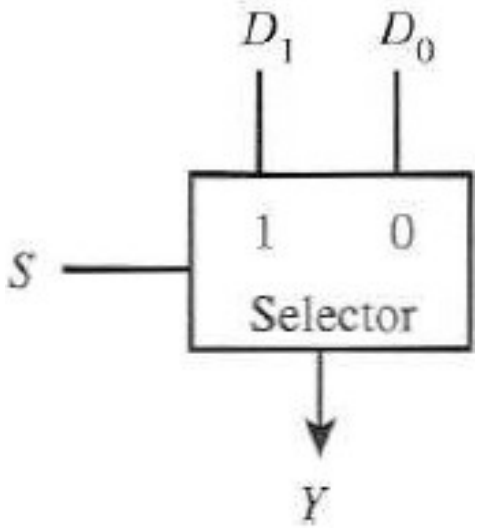
El **multiplexor** y el **demultiplexor digitales** son circuitos lógicos combinacionales que permiten realizar las mismas funciones que los analógicos pero con información binaria.
Tienen definidas sus entradas y salidas, por lo tanto son **unidireccionales**.



Multiplexor (MUX)

Un **multiplexor** es un circuito combinacional que permite transmitir la información digital presente en sus entradas (de datos) hacia una única salida.
Para poder transmitir la información es necesario entradas de selección o control, mediante las cuales se selecciona qué entrada se conecta a la salida.
Un multiplexor de $\mathbf{2^n}$ entradas de datos tendrá **n** entradas de control.

Diseño de un MUX de **2 entradas de datos** y **1 entrada de control**:



S	Y
0	D_0
1	D_1

S	D1	D0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Multiplexor (MUX)

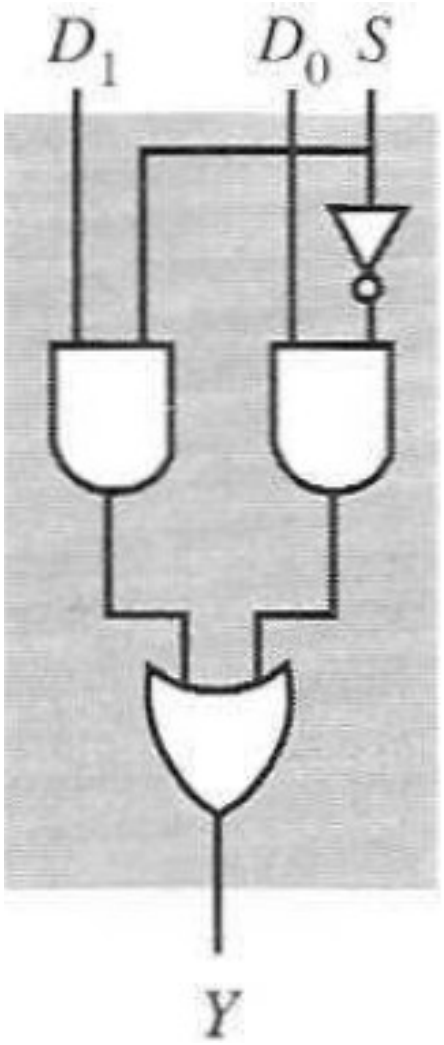
FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Resolviendo:

S	D1	D0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

S \ D1D0	00	01	11	10
0	0	1	1	0
1	0	0	1	1

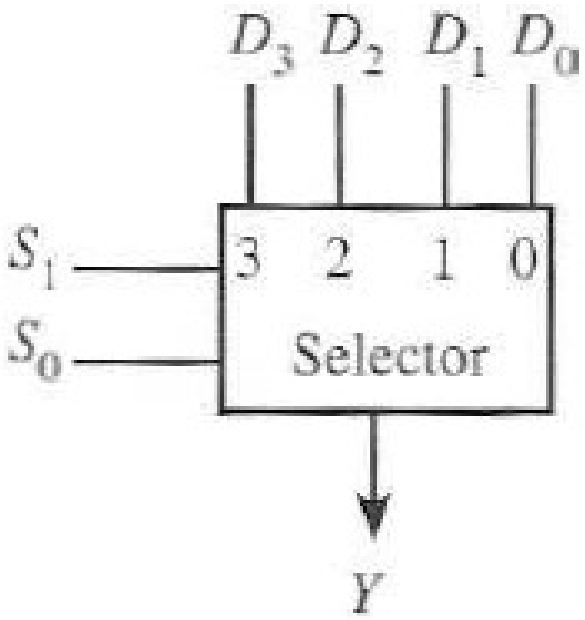
$$Y = \overline{S} \cdot D_0 + S \cdot D_1$$



Multiplexor (MUX)

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Diseño de un MUX de **4 entradas de datos** y **2 entradas de control**:



S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

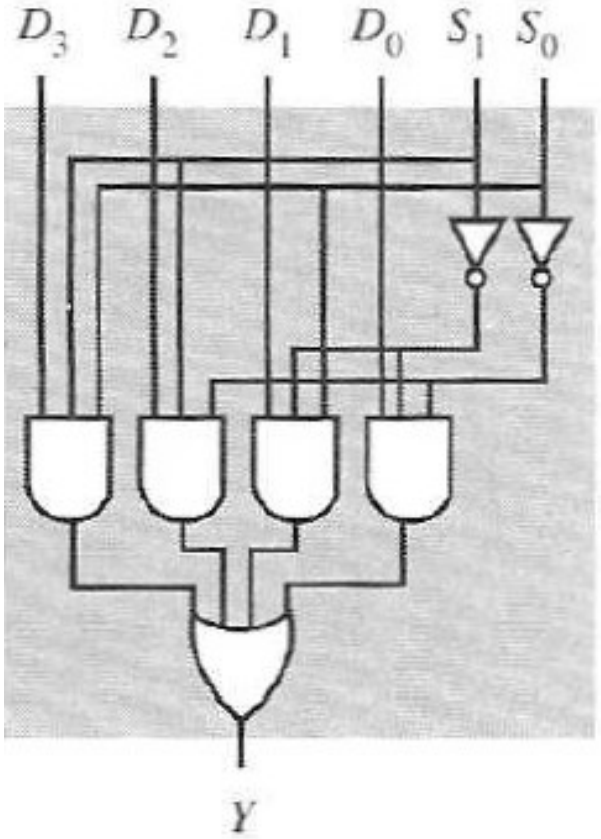
S1	S0	D3	D2	D1	D0	Y
0	0	x	x	x	0	0
0	0	x	x	x	1	1
0	1	x	x	0	x	0
0	1	x	x	1	x	1
1	0	x	0	x	x	0
1	0	x	1	x	x	1
1	1	0	x	x	x	0
1	1	1	x	x	x	1

Multiplexor (MUX)

Resolviendo:

S1	S0	D3	D2	D1	D0	Y
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

$$Y=\overline{S_0}\,\overline{S_1}D_0+ S_0\overline{S_1}D_1+ \overline{S_0}S_1D_2+ S_0S_1D_3$$

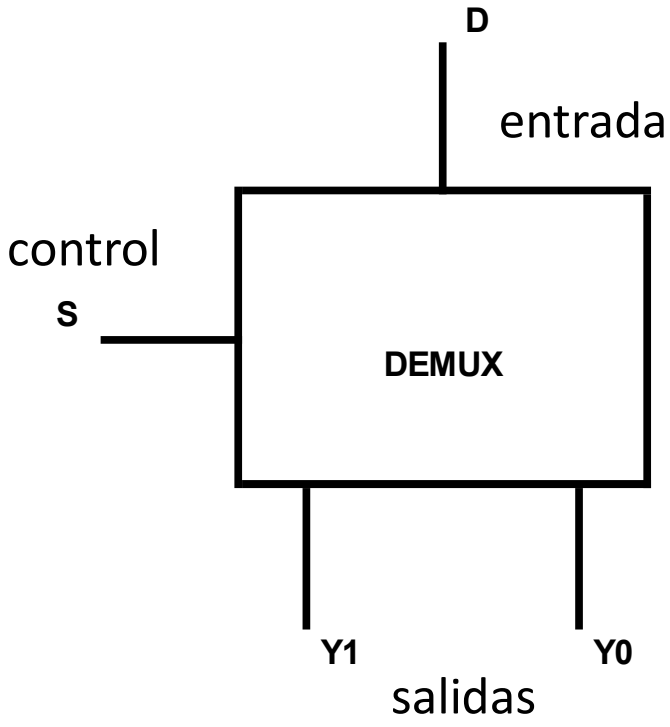


Demultiplexor (DEMUX)

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Un **demultiplexor** realiza la operación inversa a la que realiza un multiplexor. Distribuye los datos provenientes de una entrada entre varias salidas.
Un multiplexor de 2^n salidas tendrá **n** entradas de control.

Diseño de un DEMUX de **2 salidas** y **1 entrada de control**:



S	Y1	Y0
0	0	D
1	D	0

S	D	Y1	Y0
0	0	0	0
0	1	0	1
1	0	0	0
1	1	1	0

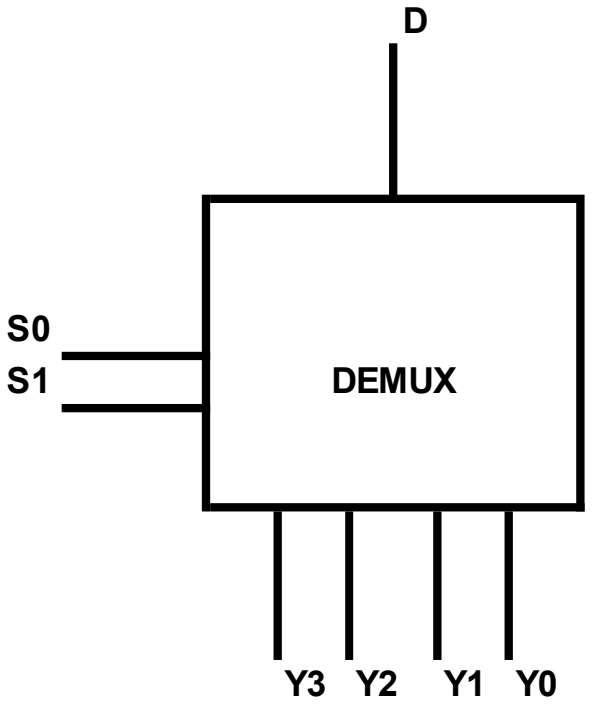
$$Y_0 = \bar{S}\, D$$

$$Y_1 = S\, D$$

Demultiplexor (DEMUX)

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Diseño de un DEMUX de **4 salidas** y **2 entradas de control**:



S1	S0	Y3	Y2	Y1	Y0
0	0	0	0	0	D
0	1	0	0	D	0
1	0	0	D	0	0
1	1	D	0	0	0

S1	S0	D	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	0	0
0	1	1	0	0	1	0
1	0	0	0	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	0	0
1	1	1	1	0	0	0

Demultiplexor (DEMUX)

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Resolviendo:

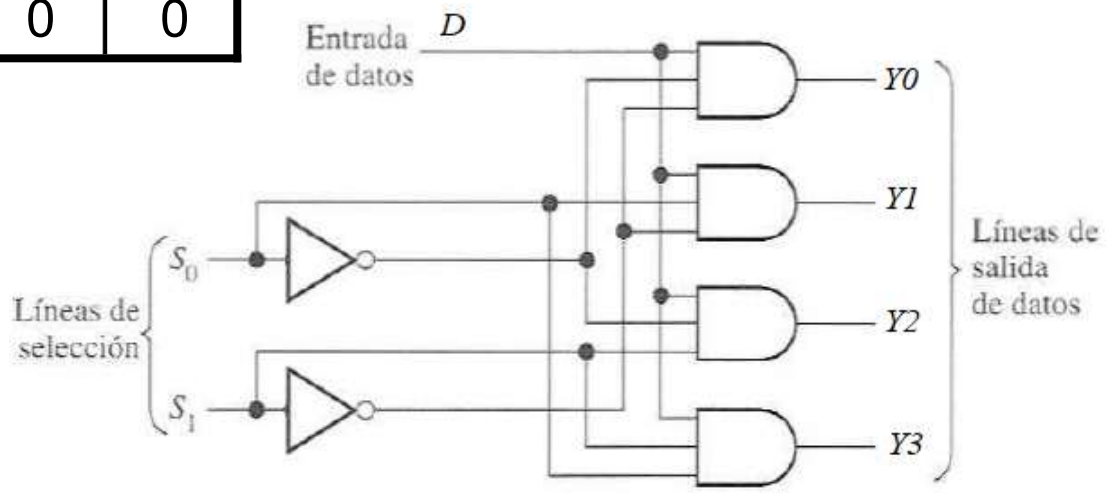
S1	S0	D	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	0	0
0	1	1	0	0	1	0
1	0	0	0	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	0	0
1	1	1	1	0	0	0

$$Y_0 = \bar{S}_1 \bar{S}_0 D$$

$$Y_1 = \bar{S}_1 S_0 D$$

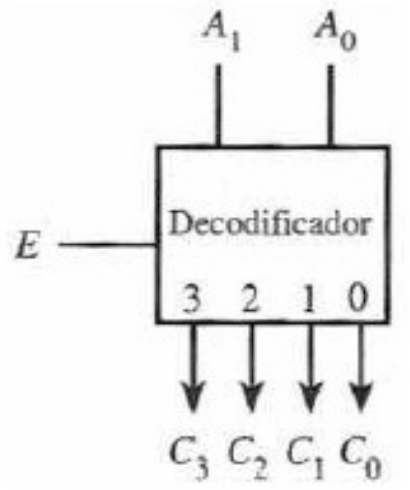
$$Y_2 = S_1 \bar{S}_0 D$$

$$Y_3 = S_1 S_0 D$$



Decoder / Demultiplexer

Analizando vemos que si bien son conceptos distintos, las TV son similares y las funciones y circuitos son iguales:

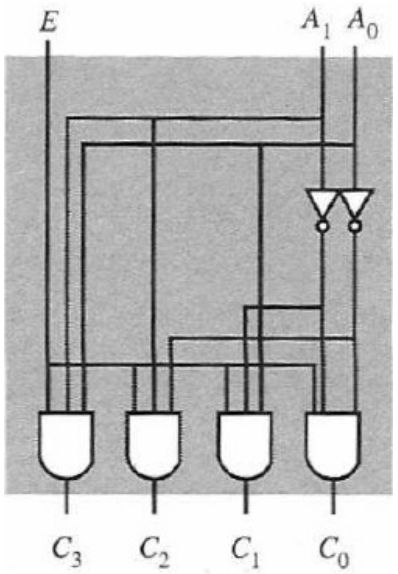


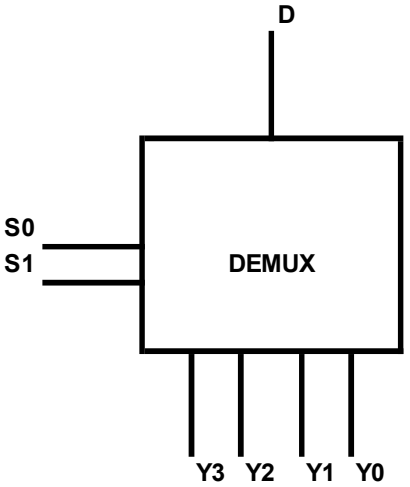
$$C_0 = E \cdot \overline{A}_1 \cdot \overline{A}_0$$

$$C_1 = E \cdot \overline{A}_1 \cdot A_0$$

$$C_2 = E \cdot A_1 \cdot \overline{A}_0$$

$$C_3 = E \cdot A_1 \cdot A_0$$



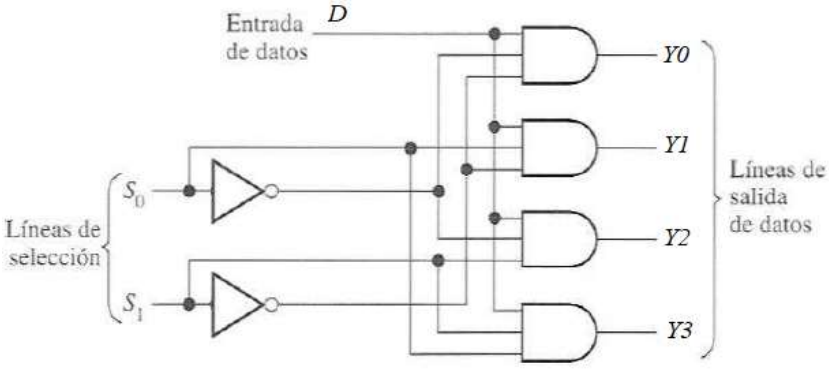


$$Y_0 = \overline{S}_1 \, \overline{S}_0 \, D$$

$$Y_1 = \overline{S}_1 \, S_0 \, D$$

$$Y_2 = S_1 \, \overline{S}_0 \, D$$

$$Y_3 = S_1 \, S_0 \, D$$



Decoder / Demultiplexer

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Ejemplo comercial:

74HC/HCT138

3-to-8 line decoder/demultiplexer; inverting

INPUTS						OUTPUTS							
$\overline{E}_1$	$\overline{E}_2$	E_3	A_0	A_1	A_2	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

Implementación con Decoder / Demux

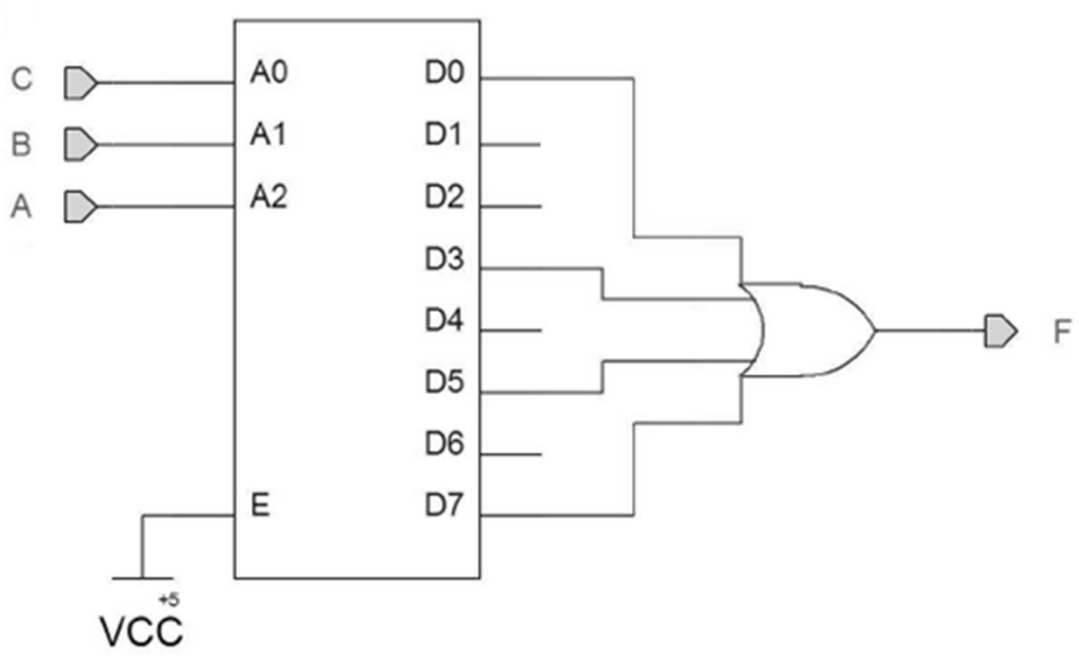
FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Podemos **implementar funciones** a partir de su TV con **Decoder / Demux** ya que estos generan los minitérminos de una función de **n** variables. Luego los sumamos mediante una compuerta **OR**.

Ejemplo: Implementar con un Decoder / Demux la función dada.

$$F(A,B,C) = \sum(0,3,5,7)$$

	A	B	C	F
0)	0	0	0	1
1)	0	0	1	0
2)	0	1	0	0
3)	0	1	1	1
4)	1	0	0	0
5)	1	0	1	1
6)	1	1	0	0
7)	1	1	1	1

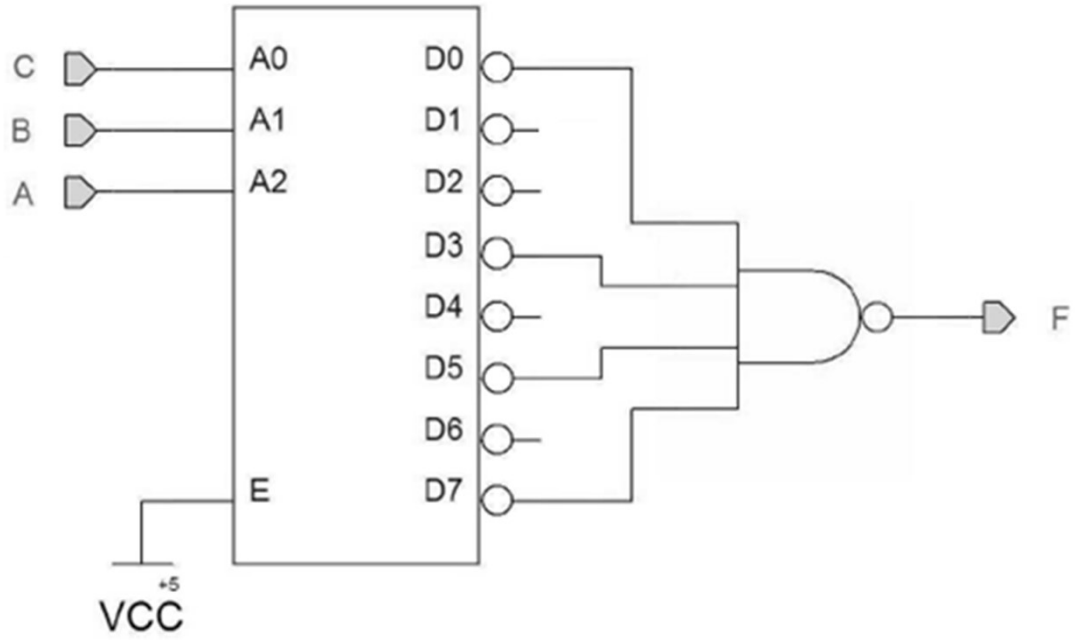


Implementación con Decoder / Demux

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Si el **Decoder / Demux** tiene salidas negadas, aplicando De Morgan tenemos:

	A	B	C	F
0)	0	0	0	1
1)	0	0	1	0
2)	0	1	0	0
3)	0	1	1	1
4)	1	0	0	0
5)	1	0	1	1
6)	1	1	0	0
7)	1	1	1	1



Implementación con MUX único

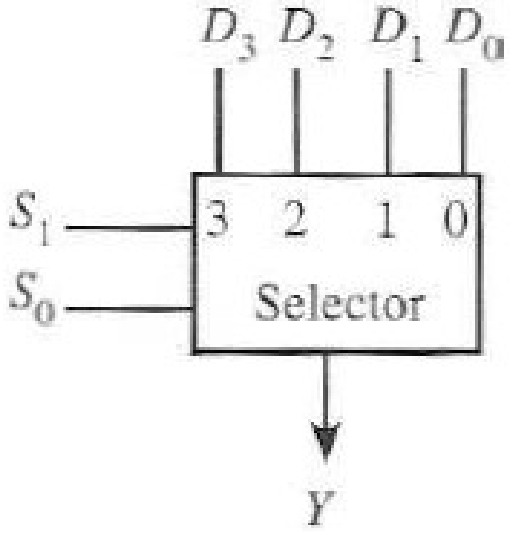
FACULTAD DE INGENIERIA
Universidad de Buenos Aires

También podemos implementar funciones de **n** variables con **MUX** de **n-1** entradas de control, colocando **n-1** variables de la función en las entradas de control, quedando las entradas de datos como funciones de la variable no usada.
Ejemplo: Implementar la función de 3 variables con **un solo MUX**.
La función es de 3 variables, entonces el MUX debe tener 2 entradas de control.

$F(A,B,C) = \sum(1,2,4,5)$

$F(A,B,C) = \prod(0,3,6,7)$

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



AB \ C	0	1	
00	0	1	D0
01	1	0	D1
11	0	0	D3
10	1	1	D2

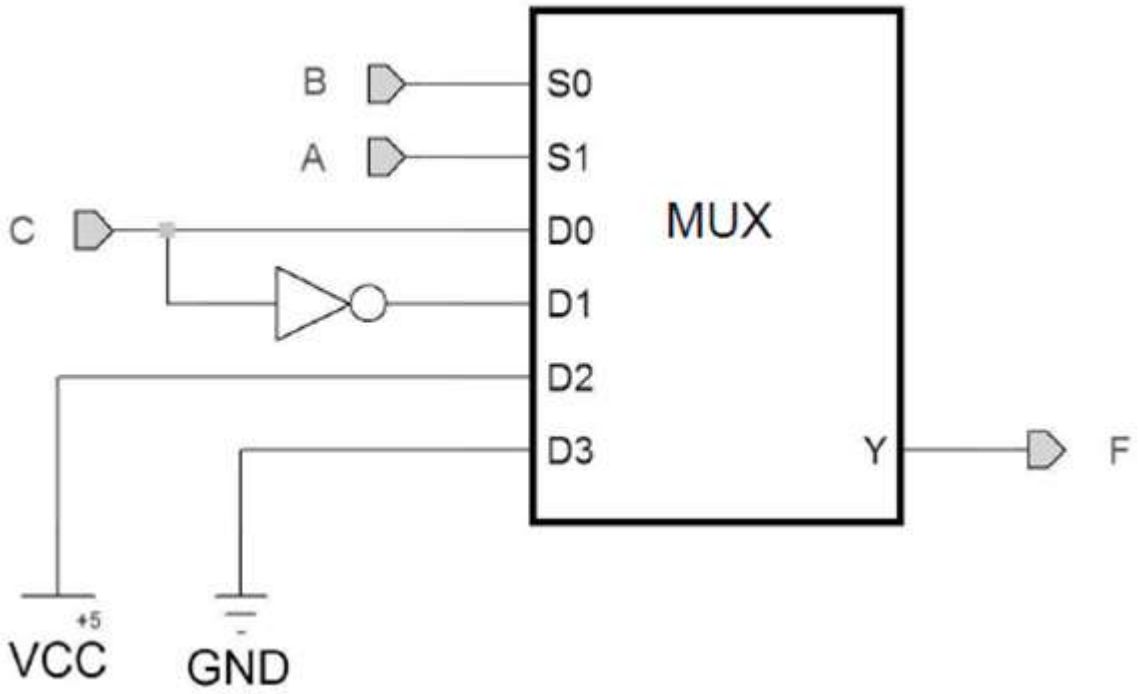
Implementación con MUX único

Analizando el mapa K nos queda:

AB \ C	0	1	
00	0	1	D0
01	1	0	D1
11	0	0	D3
10	1	1	D2

$F(A,B,C) = \sum(1,2,4,5)$

$F(A,B,C) = \prod(0,3,6,7)$



Implementación con MUX único

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Ejemplo: Implementar la función de 4 variables con **un solo MUX**.

$F(A,B,C,D) = \sum(1,2,4,5,9,10,11)$
$F(A,B,C,D) = \prod(0,3,6,7,8,12,13,14,15)$

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

AB \ CD	00	01	11	10
00	0	1	0	1
01	1	1	0	0
11	0	0	0	0
10	0	1	1	1

DO	D1
D2	D3
D6	D7
D4	D5

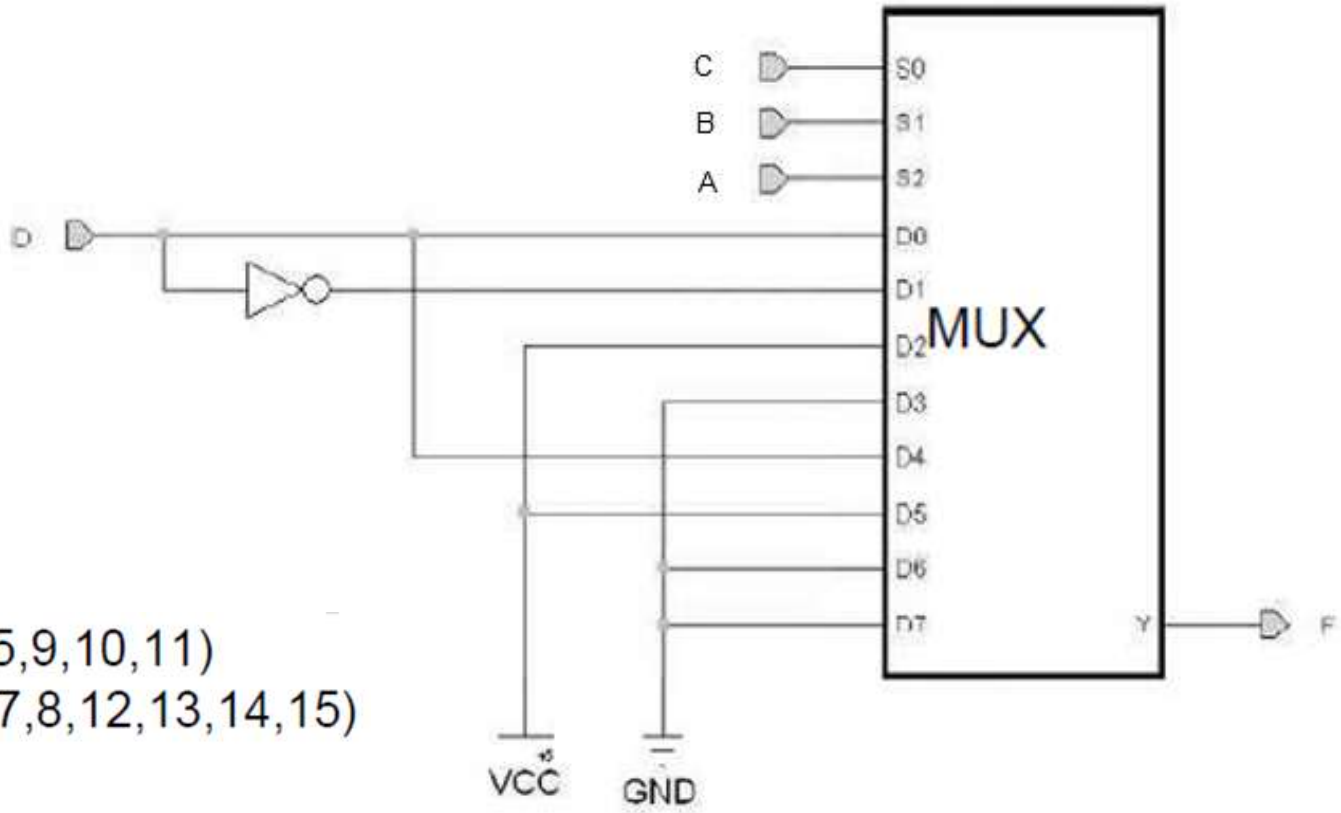
D	Not D
1	0
0	0
D	1

Implementación con MUX único

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Otra forma de visualizar el mapa K y el circuito final:

ABC \ D	0	1	
000	0	1	D0
001	1	0	D1
011	0	0	D3
010	1	1	D2
110	0	0	D6
111	0	0	D7
101	1	1	D5
100	0	1	D4



$F(A,B,C,D) = \sum(1,2,4,5,9,10,11)$

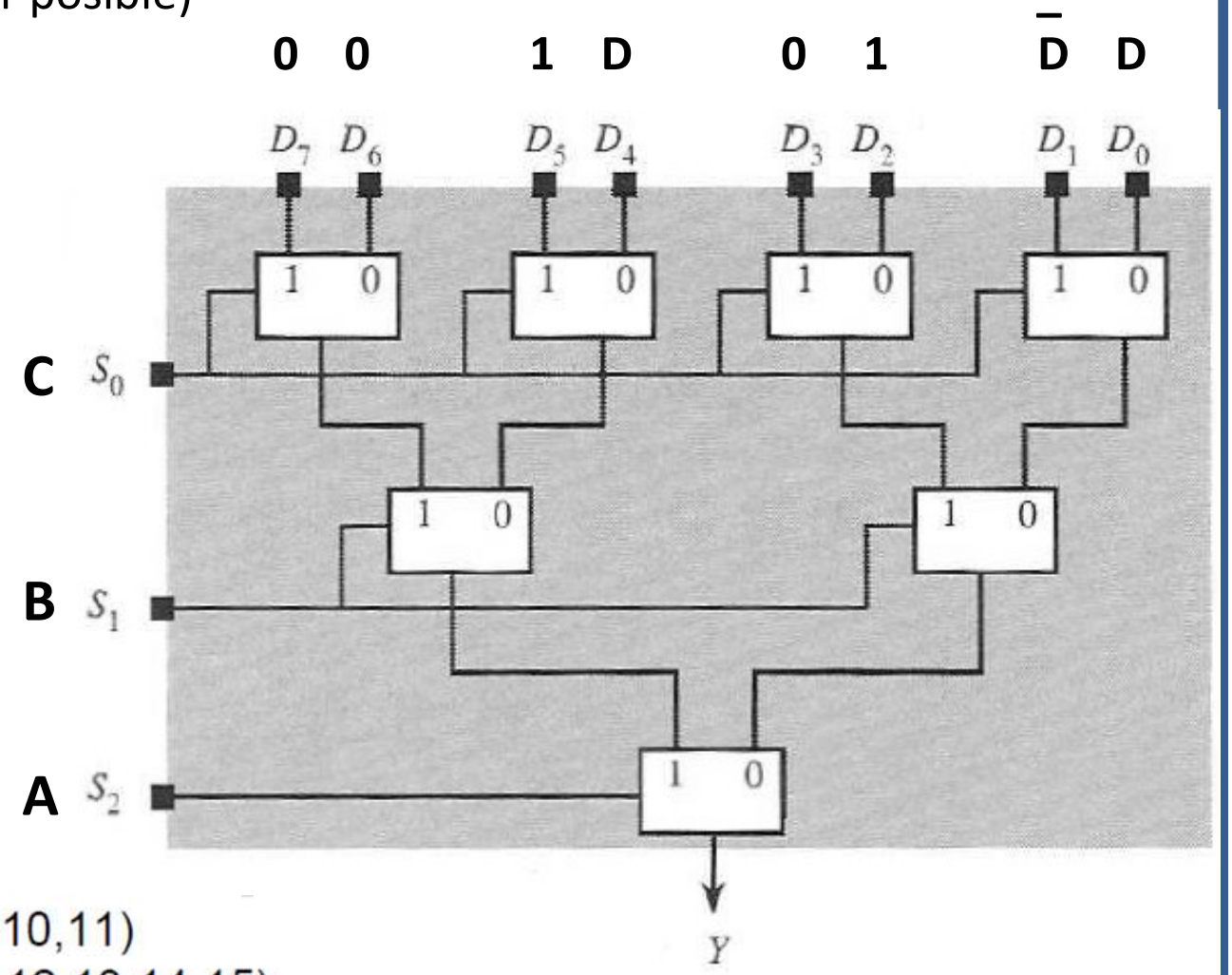
$F(A,B,C,D) = \prod(0,3,6,7,8,12,13,14,15)$

Implementación c/ MUX de 1 var. (árbol)

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Ejemplo: Implementar la misma función pero solo con MUX de **1 variable de control**. (Simplificar de ser posible)

ABC \ D	0	1	
000	0	1	D0
001	1	0	D1
011	0	0	D3
010	1	1	D2
110	0	0	D6
111	0	0	D7
101	1	1	D5
100	0	1	D4



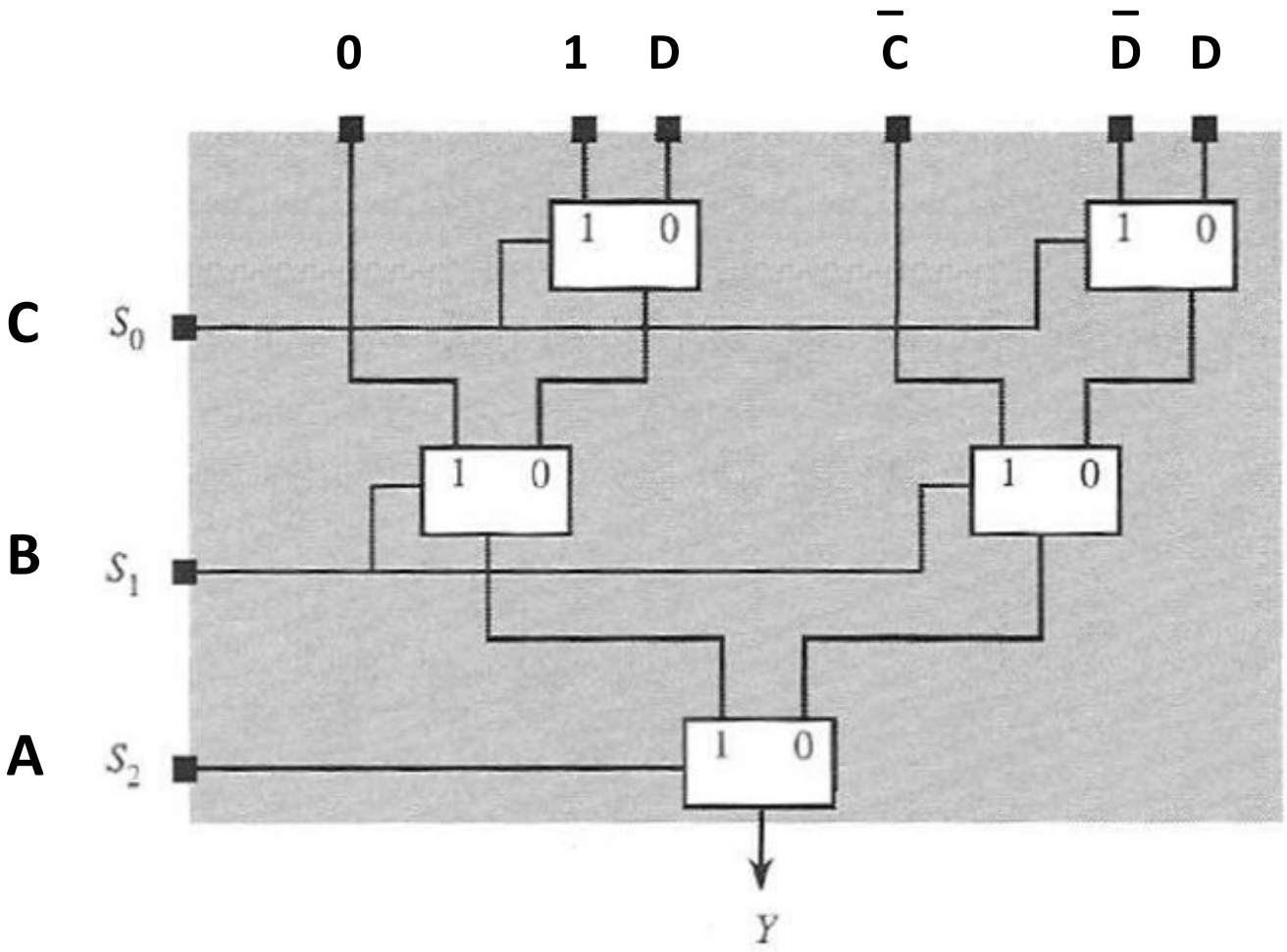
$F(A,B,C,D) = \sum(1,2,4,5,9,10,11)$

$F(A,B,C,D) = \prod(0,3,6,7,8,12,13,14,15)$

Implementación c/ MUX de 1 var. (árbol)

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

El circuito simplificado quedará:



Conclusiones

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Análisis y síntesis de:

- Encoder y Decoder
- Conversor de código
- Mux y Demux

Síntesis de circuitos combinacionales utilizando:

- Decoder / Demux (+ compuerta)
- Mux
  - Único Mux
  - En árbol

Conclusiones

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Estamos en condiciones de sintetizar funciones:

- Como SP (Mapa K por 1s)
- Como PS (Mapa K por 0s)
- Solamente con NAND
- Solamente con NOR
- Con Decoder / Demux (+ compuerta)
- Con Mux (único / en árbol)

Más adelante:

- Con PLDs (Dispositivos Lógicos Programables)

FACULTAD DE INGENIERIA
Universidad de Buenos Aires