UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Lógica Digital – Parte 1

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Introducción

En esta Unidad veremos la metodología de diseño lógico de circuitos electrónicos, digitales, combinacionales y secuenciales, así como de dispositivos lógicos programables.

Circuitos lógicos de sistemas digitales

Una expresiones booleanas permiten la representación algebraica del valor de una función que representa la salida de un circuito.

En los circuitos lógicos de sistemas digitales los estados 0 o 1 determinan dos niveles de tensión.

Circuitos combinacionales

Cada compuerta es un circuito lógico combinacional en sí. De esta manera se puede “armar” el esquema del comportamiento total del circuito sin necesidad de conocer los elementos electrónicos que lo constituyen.

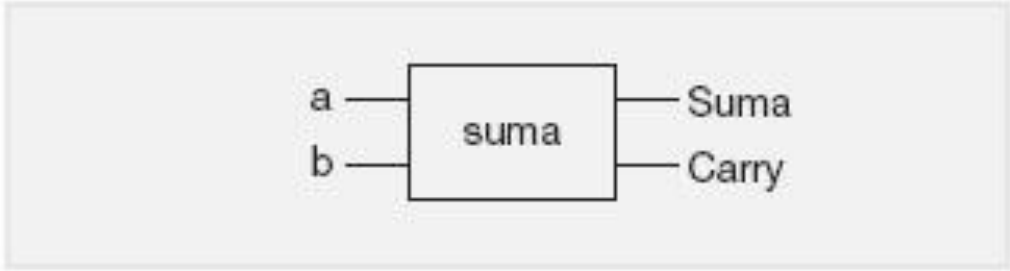
Los circuitos lógicos se clasifican en dos:

- Combinacionales
- Secuenciales

Circuitos combinacionales

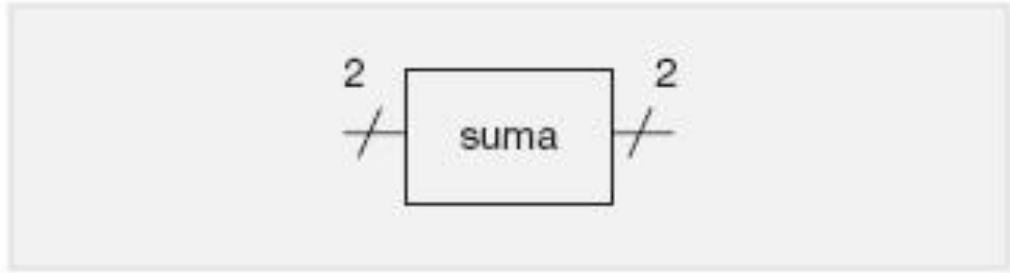
Circuito Semi-Sumador:

Diagrama de bloque



a	*b*	*Suma*	*Carry*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

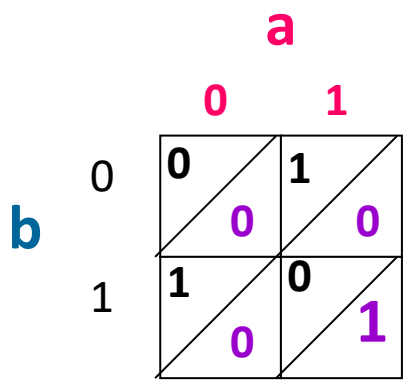
Diagrama de bloque (otra opción)



Sumador Elemental: Semisumador

El sumador elemental será capaz de sumar un par de bits “a” y “b”.

La Tabla de Verdad se deriva a partir de la Tabla de Suma Aritmética.



a	b	S suma	C acarreo
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

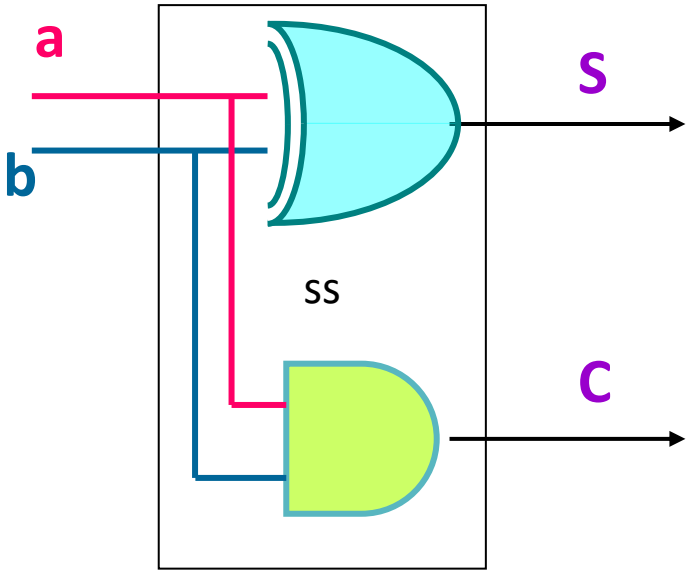
$S = a \oplus b$

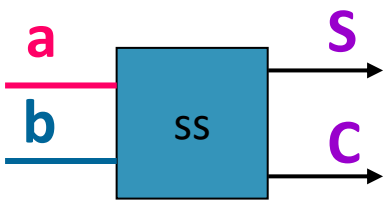
$C = a \bullet b$

Sumador Elemental: Semisumador

Para sumar dos cantidades de *n* bits es necesario disponer de *n* bloques funcionales como el obtenido.

Si bien permiten sumar bit a bit, al no tener en cuenta el acarreo no implementan correctamente la suma. Por este motivo este bloque se denomina **semisumador** o *half adder*.





UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Circuitos combinacionales

Niveles de descripción de un circuito combinacional

De menor a mayor nivel de abstracción encontramos en primer lugar la descripción del circuito electrónico, en segundo lugar los diagramas de lógica, luego la representación algebraica y por último el nivel algorítmico de diseño de componentes.

Así como se utilizan lenguajes de programación como FORTRAN, Pascal y C para describir programas de computadora de naturaleza secuencial.

Wn el campo del diseño digital, los diseñadores sintieron la necesidad de un lenguaje estándar para describir los circuitos digitales. Por lo tanto, surgieron los lenguajes de descripción de hardware (HDL).

Los HDL permitieron a los diseñadores modelar la concurrencia de procesos encontrados en elementos de hardware.

Los lenguajes de descripción de hardware como Verilog HDL y VHDL son los mas populares. Verilog HDL se originó en 1983 en Gateway Design Automation. Más tarde, VHDL se desarrolló bajo contrato de DARPA. Los simuladores Verilog y VHDL para simular grandes circuitos digitales ganaron rápidamente la aceptación de los diseñadores.

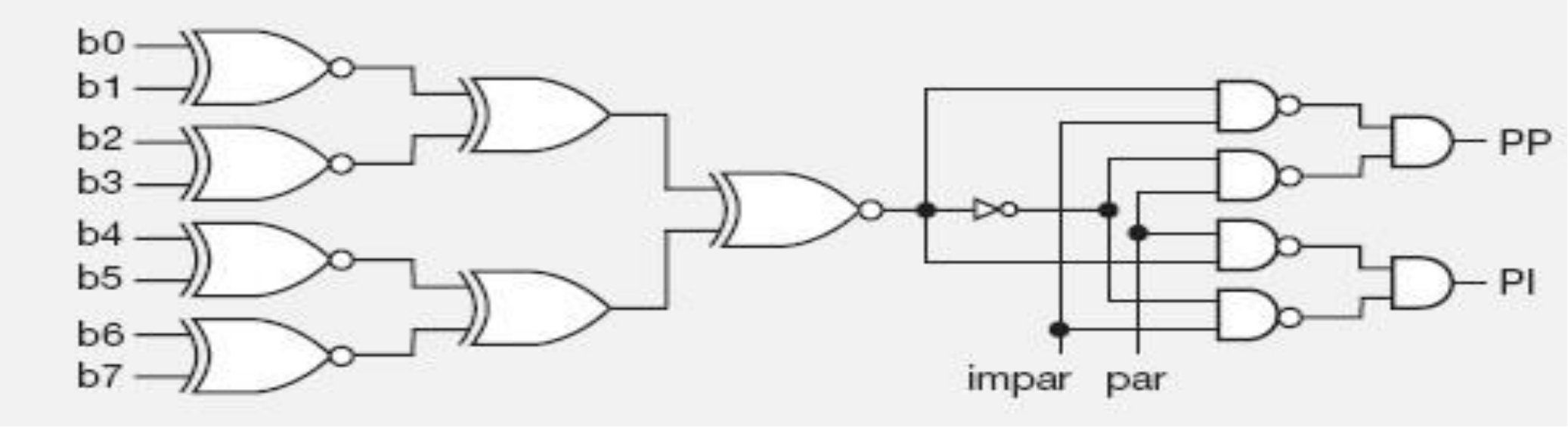
VHDL se estandarizó con los estandares 13641995,1364-2001, 1364-2005.

http://ieeexplore.ieeee.org/xls/standardstoc.jsp?isnumber=33945

Circuitos combinacionales

Circuito generador de paridad

Un circuito generador de paridad permite la detección de error en la transmisión de información entre un emisor y un receptor de información binaria cuyo coeficiente de probabilidad de error sea menor o igual a un bit.



Circuitos combinacionales

Circuito comparador de magnitud

Es común que se necesite comparar dos números binarios de *n* bits para saber si éstos son iguales, o si uno es mayor o menor que otro. Podemos simplificar el problema analizando la comparación de dos bits.

Diagrama de bloque

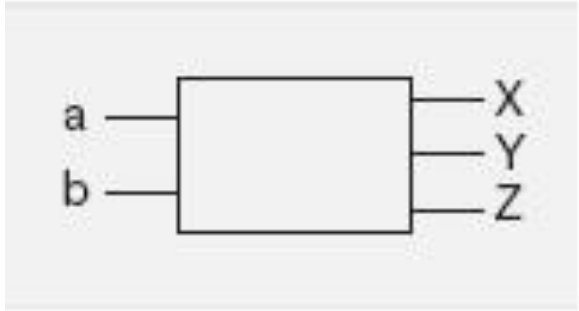
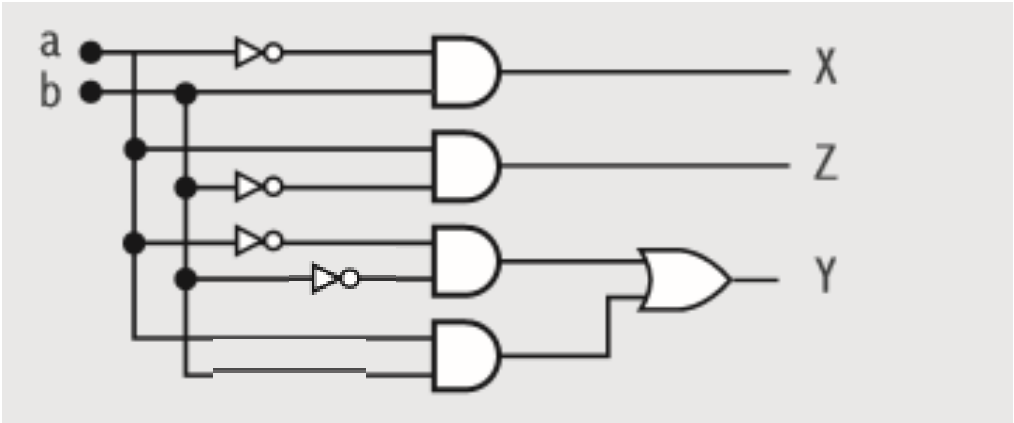


Tabla de verdad

		Z	X	Y
a	*b*	*a* > *b*	*a* < *b*	*a* = *b*
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1



Circuitos combinacionales

Circuito codificador

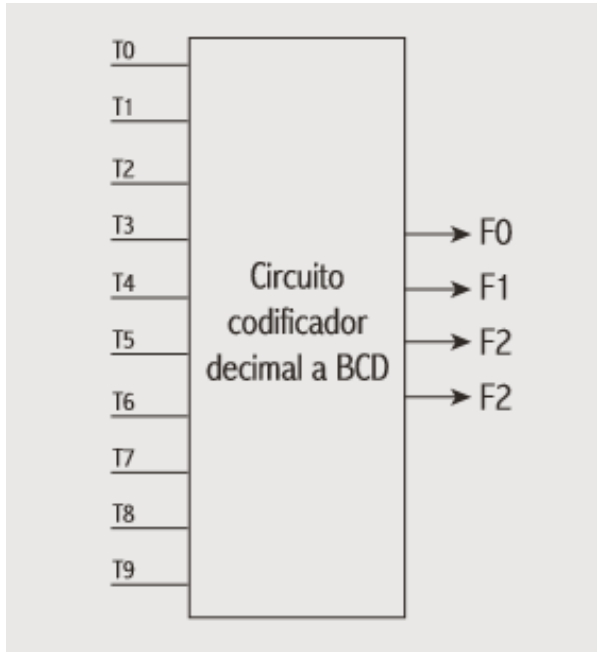
Las señales binarias provenientes de un teclado numérico pueden transformarse en salidas codificadas, por ejemplo, en *BCD*. Consideremos teclas numéricas sin tener en cuenta las teclas de operación (+, -, *, etc.) comunes en todo teclado.

Uno de los códigos más usados en la representación de números es el *BCD 8421* o *BCD* puro.

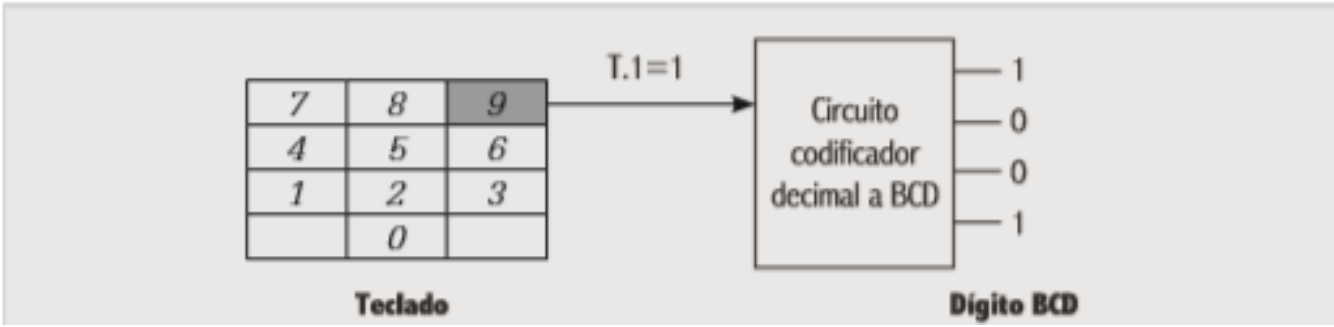
La siguiente tabla de verdad muestra *10* combinaciones de entrada en decimal y la salida en BCD.

Circuitos combinacionales

Circuito codificador



Entradas que identifican las teclas										Salidas BCD			
										8	4	2	1
T9	T8	T7	T6	T5	T4	T3	T2	T1	T0	F_3	F_2	F_1	F_0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1



Circuitos decodificador de códigos

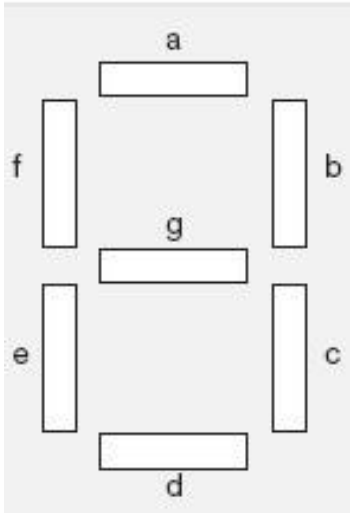
Circuito decodificador

Opera en forma inversa al codificador en el siguiente sentido: la información que ingresa en el circuito implica combinaciones binarias pertenecientes a alguna convención de representación de caracteres (numéricos o alfanuméricos) y las convierte a señales binarias para representar en algún dispositivo de salida de información

Circuitos decodificador de códigos

Circuito decodificador

Diagrama de bloque



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	0	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

Circuitos decodificador de códigos

Circuito decodificador n.2^n

Existe otro tipo de decodificadores que permiten señalar en una de sus salidas qué combinación binaria se dio en la entrada. Los decodificadores de este tipo se denominan "*nX2^n*" (se lee "*n x 2 a la n*" o "*n a 2 a la n*") y contemplan una salida activa para cada posible combinación de entradas.

Diagrama de bloque sin entrada de habilitación

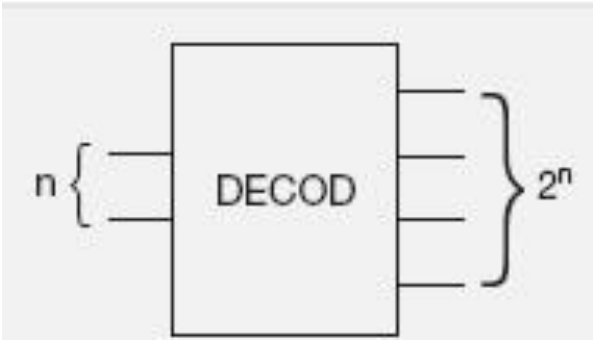
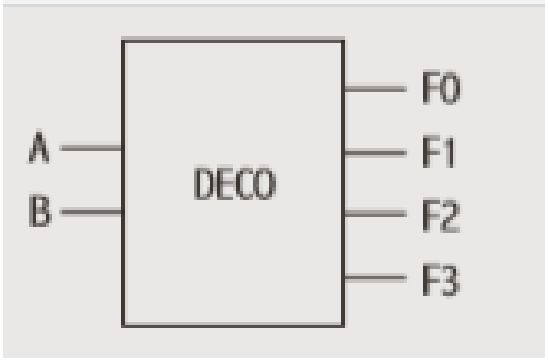


Diagrama de bloque del decodificador $n \cdot 2^n = 2 \cdot 2^2 = 2 \cdot 4$ sin entrada de habilitación

Circuitos decodificador de códigos

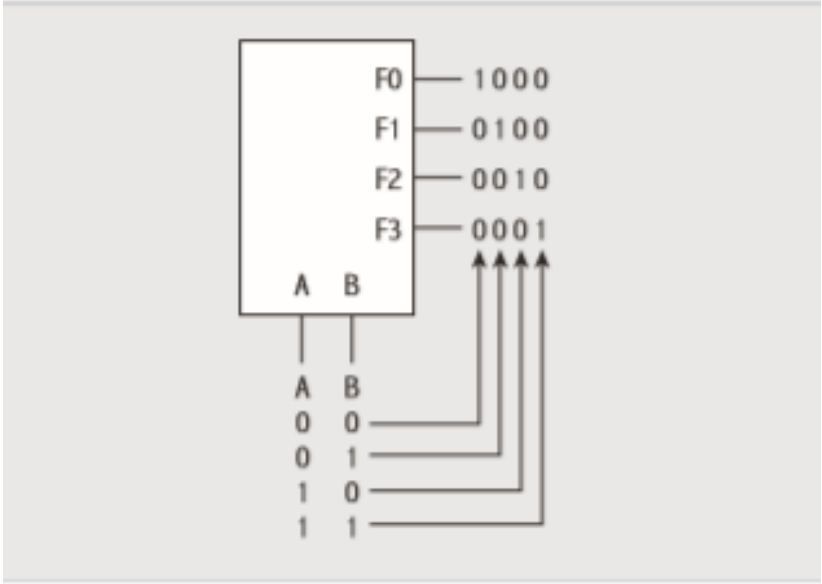


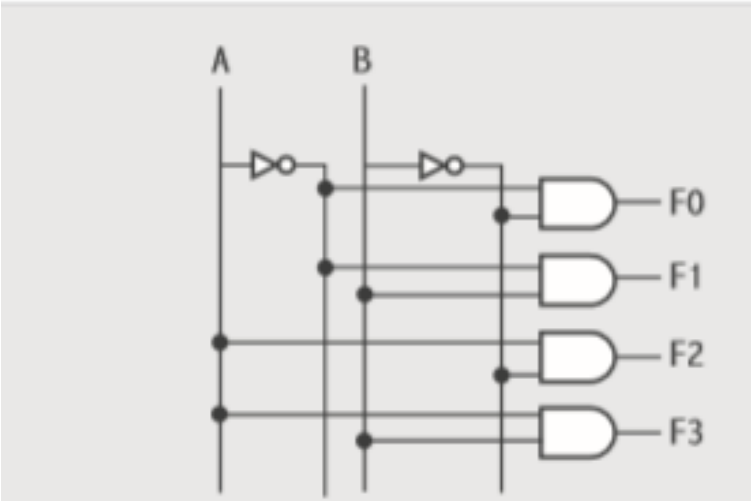
A	B	F0	F1	F2	F3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$$F0 = \bar{A} \cdot \bar{B}$$
$$F1 = \bar{A} \cdot B$$
$$F2 = A \cdot \bar{B}$$
$$F3 = A \cdot B$$

Se arma la tabla con todas las combinaciones de entradas y salidas esperadas.

Luego se obtienen los miniterminos (para este caso) y luego se construye el circuito.



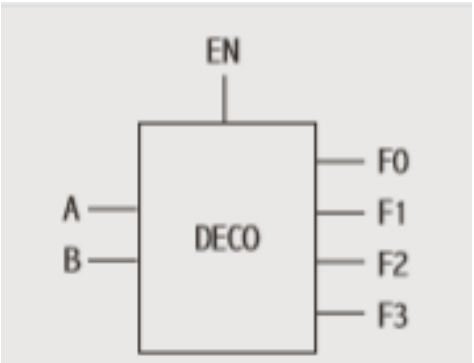


Circuitos decodificador de códigos

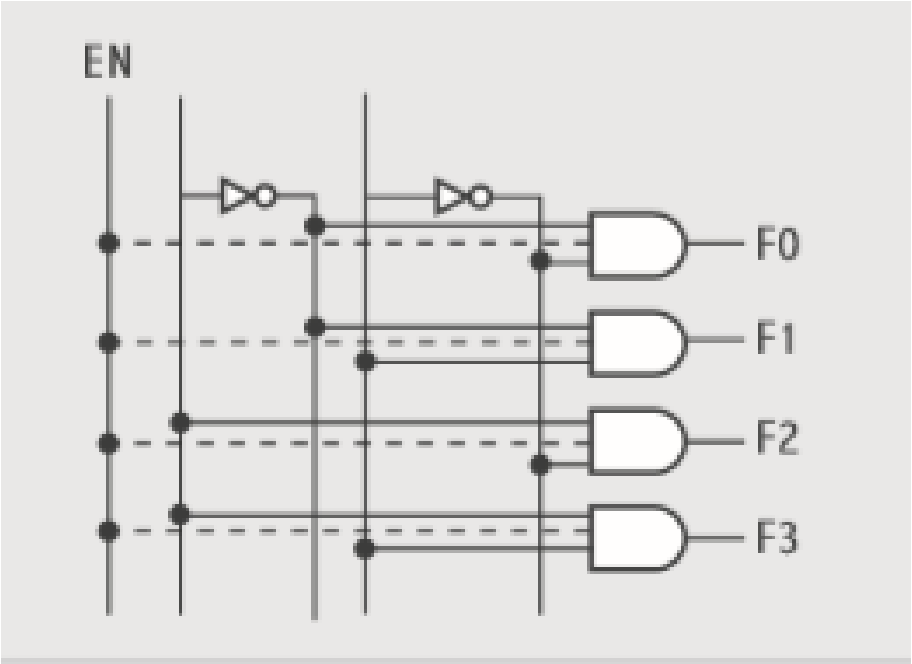
Se puede utilizar un **EN** = habilitación del circuito.

En las cuatro primeras combinaciones no decodifica

En términos algebraicos, si EN =1, entonces los minitérminos adquieren una nueva variable y las funciones se representan así



EN	a	b	F0	F1	F2	F3
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	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1



$$F0 = EN \cdot \bar{A} \cdot \bar{B} = 1 \cdot \bar{0} \cdot \bar{0} = 1 \cdot 1 \cdot 1 = 1$$

$$F1 = EN \cdot \bar{A} \cdot B = 1 \cdot \bar{0} \cdot 0 = 1 \cdot 1 \cdot 0 = 0$$

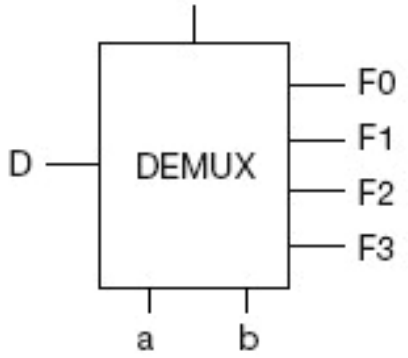
$$F2 = EN \cdot A \cdot \bar{B} = 1 \cdot 0 \cdot \bar{0} = 1 \cdot 0 \cdot 1 = 0$$

$$F3 = EN \cdot A \cdot B = 1 \cdot 0 \cdot 0 = 1 \cdot 0 \cdot 0 = 0$$

Circuitos Multiplexor y demultiplexor

Un demultiplexor es un circuito cuya función digital es “encaminar” la información que viene en una sola línea de entrada, en 1 de 2n. Se puede pensar en un decodificador en el que la entrada de habilitación es la portadora del bit que se ha de “encaminar” y la selección de la línea de salida (habilitada por líneas de selección) determina qué salida es la encargada de transmitirlo.

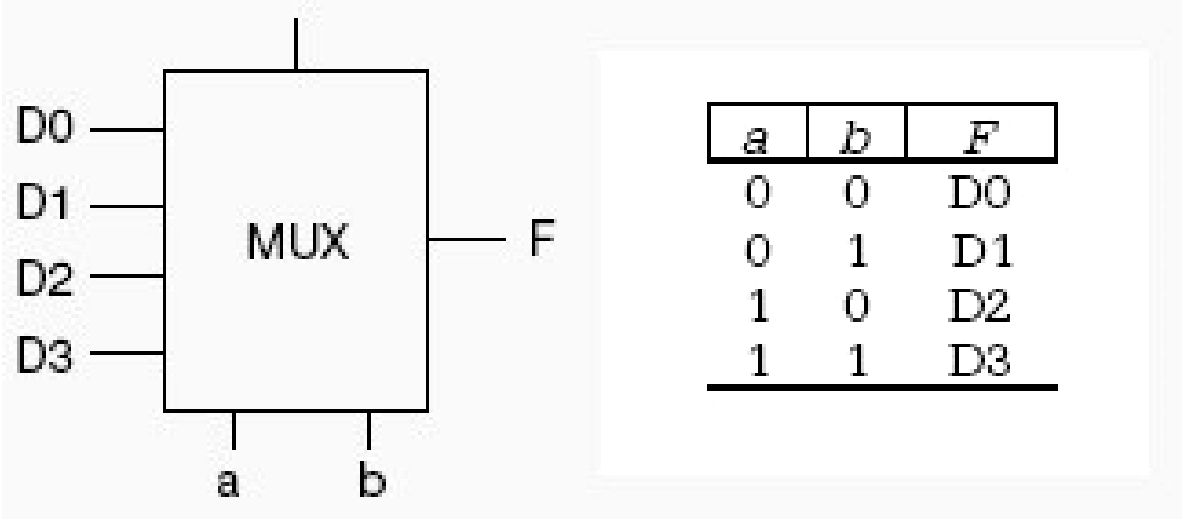
Por este motivo se lo conoce como decodificador/demultiplexor. Algunas de las aplicaciones de un demultiplexor son, por ejemplo, distribuir una entrada de datos en serie a una salida en paralelo o transferir bits de una línea de bus a un registro determinado; por eso también se lo denomina distribuidor.



a	*b*	*F0*	*F1*	*F2*	*F3*
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

Circuitos Multiplexor y demultiplexor

Un multiplexor es un circuito combinacional cuya función digital es “encaminar” las señales binarias de 1 de 2n líneas posibles de entrada en una única línea de salida. La línea de entrada de dato se “elige” a partir de los valores que puedan tomar las n líneas de selección.



a	b	F
0	0	D0
0	1	D1
1	0	D2
1	1	D3

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

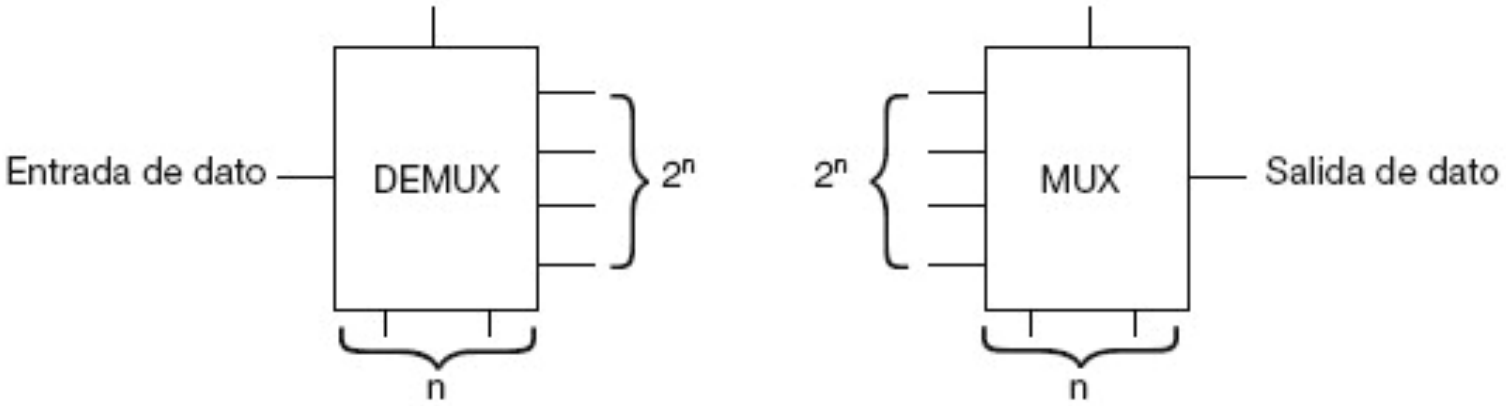
Circuitos Multiplexor y demultiplexor

Aplicaciones de los multiplexores

1. Un MUX se puede implementar con cierto número de compuertas lógicas, porque básicamente es un decodificador con sus salidas asociadas a una única compuerta.
2. Los MUX de pocas entradas se utilizan en los dispositivos de lógica programables descriptos más abajo.
3. El MUX se utiliza para la conversión de bits transferidos en paralelo a serie, esto es, de a uno por vez, utilizando una sola línea.
4. La multiplexación de bits de dirección en memorias se emplea, por ejemplo, cuando un bloque de caché supera la capacidad del bus, si el bloque es de 128 bits y el bus de 64; entonces, primero se transfiere un grupo de 64 bits y luego el otro.
5. Se puede armar un módulo desplazador que permite mover bits a derecha e izquierda en registros de desplazamiento bidireccionales, bajo microoperaciones de control, por ejemplo, las que genera la unidad de control cuando se ejecuta una instrucción de desplazamiento (ver sección "Registros con facilidad de desplazamiento").

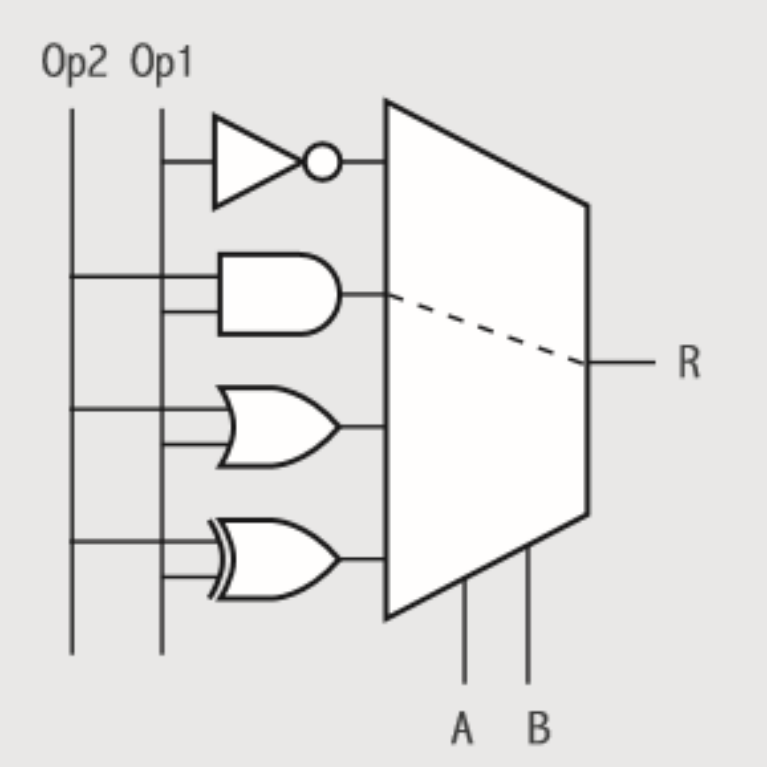
Circuitos Multiplexor y demultiplexor

Resumen



Circuitos Multiplexor y demultiplexor

Ejemplo de una unidad aritmético-lógica (ALU) para 4 operaciones de tipo lógico utilizando un multiplexor. Supongamos las operaciones NOT (negación o complemento), AND (conjunción o producto lógico), OR (disyunción inclusiva o suma lógica), XOR (disyunción excluyente o suma exclusiva), que identificamos con los números 00, 01, 10, 11, respectivamente. Las señales de control indican cuál de las "instrucciones" pasan por la línea de resultado, al operar los bits Op1 y Op2. Para este ejemplo, A = 0 y B = 1, entonces, el resultado R es el producto lógico entre ambos operadores: Op1 AND Op2.

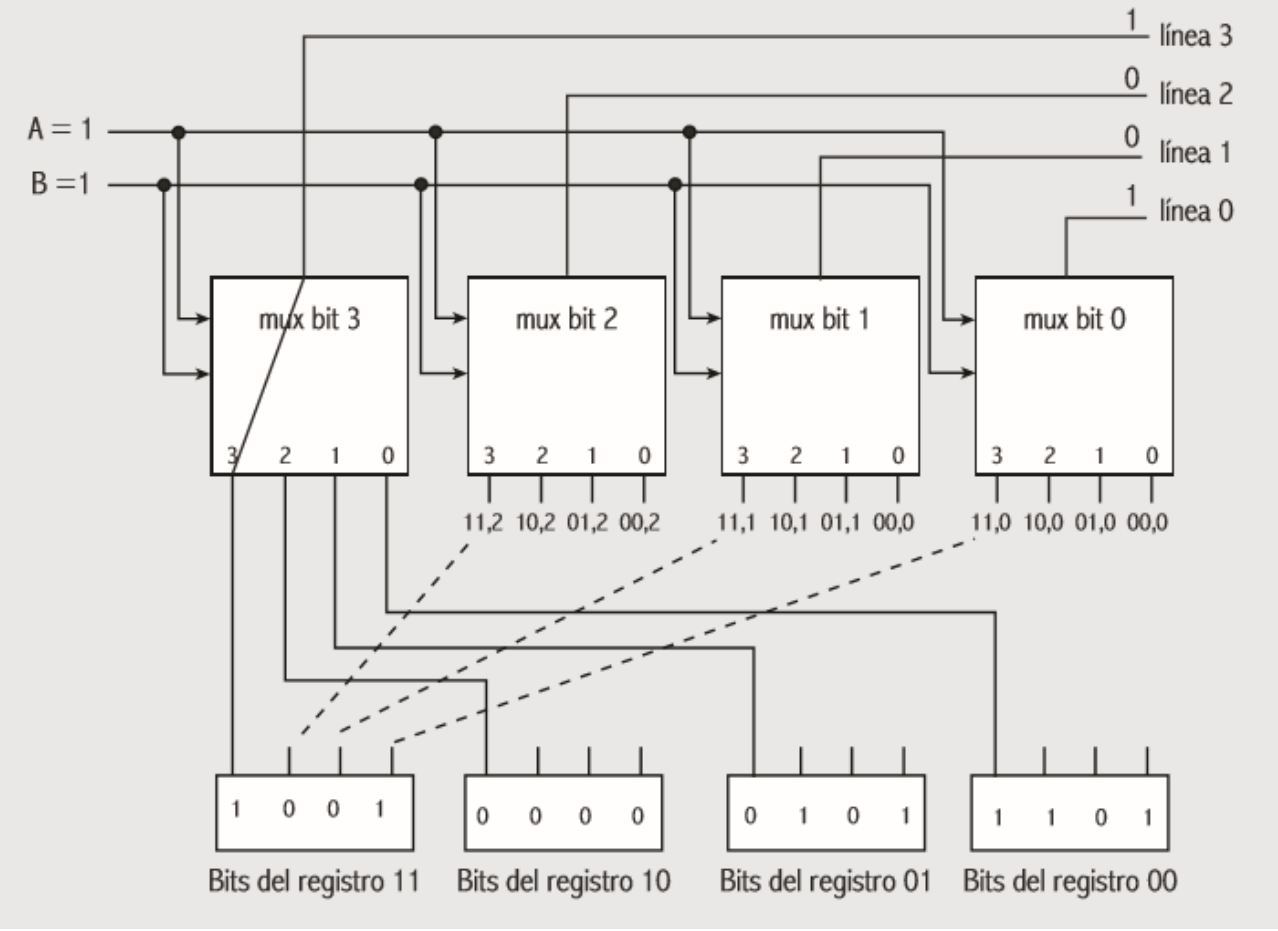


Circuitos Multiplexor y demultiplexor

Bus asociado a un multiplexor-demultiplexor

Un bus es un conjunto de conductores que permiten relacionar registros que actúan de emisores o de receptores de bits. En este caso se indica que los registros están asociados “a un bus común o bus único”.

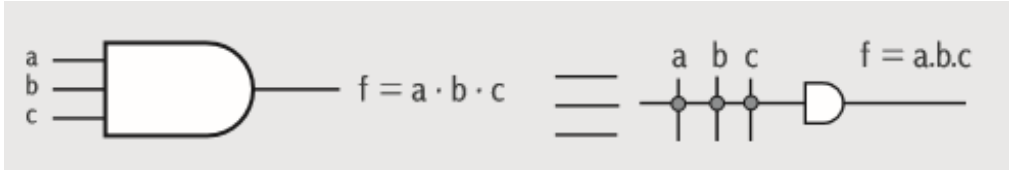
Cada multiplexor habilita el paso de una señal sobre el bus comandado por la dirección en sus entradas de selección. Así: A = 1 y B = 1 indican que D3 es la línea de entrada seleccionada para salir por F.



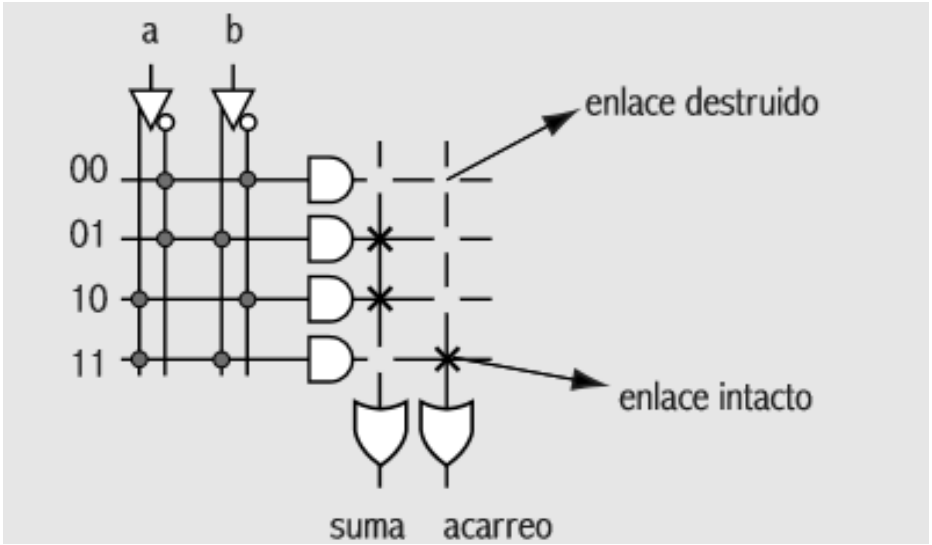
Circuitos programables

Circuitos “programables” para múltiples funciones

Si se considera que las salidas de un decodificador n . 2n son entradas a una compuerta OR, se puede diseñar una estructura de compuertas AND-OR sobre la cual “programar” una función digital. Para que el circuito resulte programable las salidas de las compuertas AND se asocian a un enlace electrónico, y éste, a su vez, a la entrada de la compuerta OR. Entonces, la lógica AND-OR no es fija, sino que puede programarse.



a	b	Suma	Acarreo
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Memoria sólo lectura

El arreglo de compuertas AND-OR, programado de esta manera, es un ejemplo simple de memoria ROM (Read Only Memory). Desde el punto de vista de los circuitos lógicos, forma parte de los dispositivos de lógica programable y se caracterizan por tener conexiones fijas en el arreglo de compuertas AND y conexiones programables en el arreglo de compuertas OR.

Una vez establecida la combinación binaria de las salidas, éstas se mantienen inalterables.

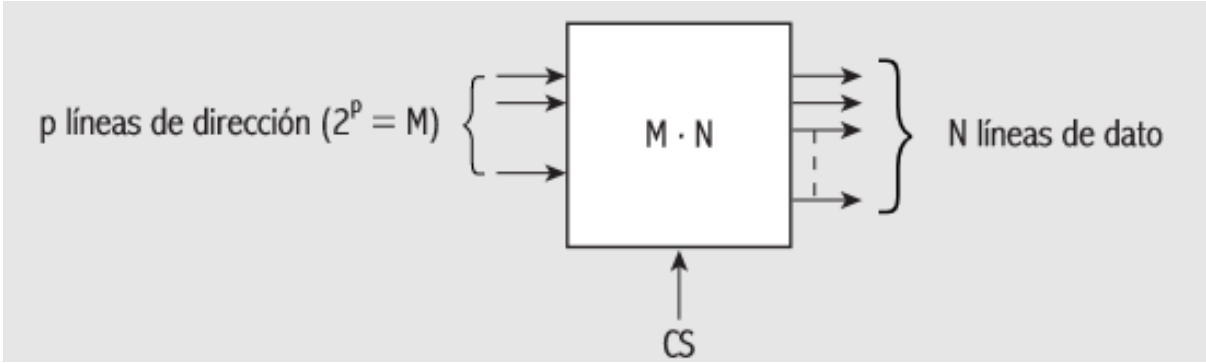
Memoria sólo lectura

Memorias sólo de lectura

Distribución de líneas en una memoria de sólo lectura

- La distribución de una memoria forma una matriz de M x N bits (M direcciones de N bits cada una).
- El bus de direcciones tiene p líneas, tal que 2p = M.
- El bus de datos tiene N líneas para transferir los N bits leídos.
- El bus de control tiene una línea que habilita el chip y normalmente se denomina CS (Chip Select).
- En cualquiera de los ROM, la red AND es inalterable y la red OR es programable.

Diagrama de bloque de cualquiera de las memorias de sólo lectura.

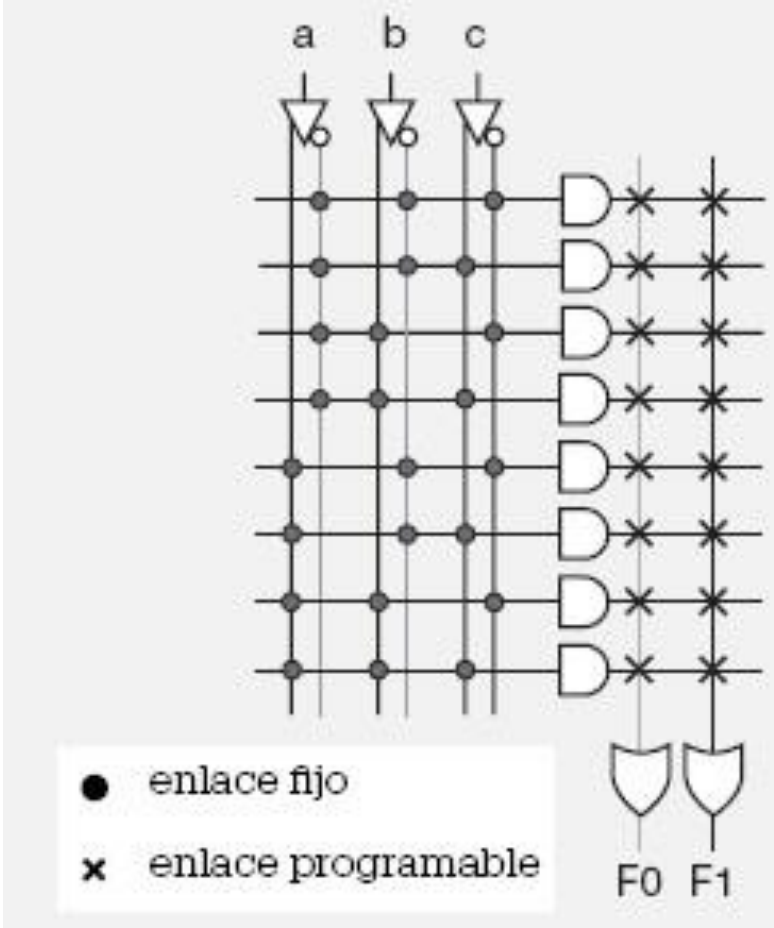


Circuitos combinacionales

Memorias sólo de lectura

Diagrama de lógica de una ROM sin programar

Una memoria ROM no es más que otra forma de representar los minitérminos para cada función de p variables; así, en el diagrama siguiente se representan las ocho combinaciones posibles de tres variables (a, b y c) como enlaces fijos en una red AND que, acoplada a una red OR, podrá “programarse” para cada una de las dos funciones de salida F0 y F1

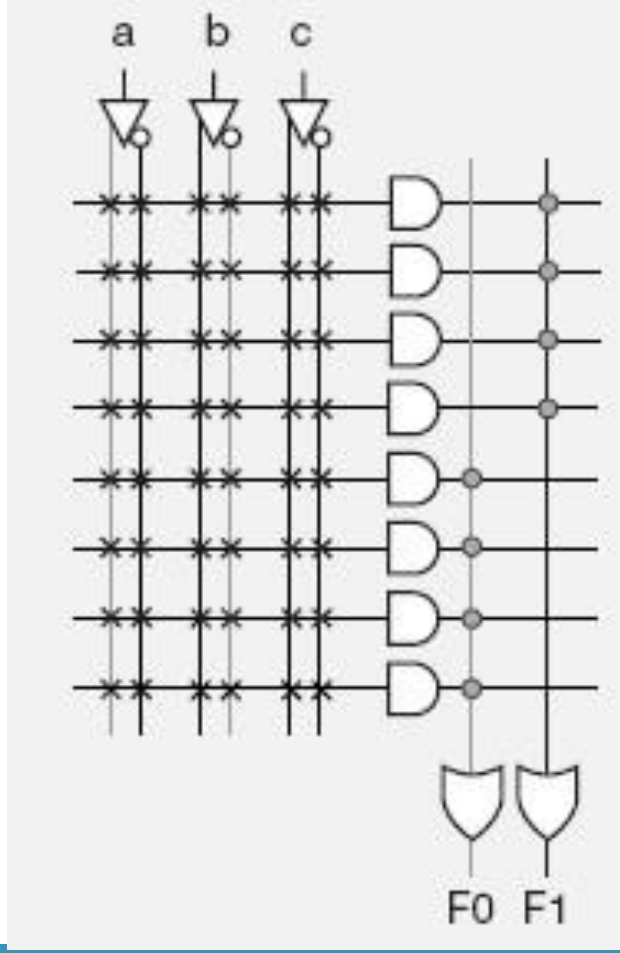


Memoria sólo lectura

Memorias sólo de lectura

Dispositivos tipo PAL

En una nueva categoría de dispositivos de lógica programable, se encuentran los denominados PAL (Programable Array Logic), más flexibles y rentables que las típicas ROM. Su función es similar a la de la ROM, pero en este caso se invierten los papeles de las redes ANDOR. La red AND es programable y la OR es fija. El diagrama para el ejemplo anterior puede ser:



Memoria sólo lectura

Memorias sólo de lectura

Dispositivos tipo PLA o F-PLA (field-PLA)

Los dispositivos tipo F-PLA (field = campo) constituyen otra categoría de los dispositivos de lógica programable, y tienen aún mayor flexibilidad, dado que ambas redes son programables.
Diagrama de lógica para un PLA no programado, esta vez de tres entradas y tres salidas: F0, F1 y F2.