

lación: Grado en Ingeniería de Computadores
Asignatura: Tecnología de Computadores



tema 2: Sistemas combinacionales
tema 6: Módulos combinacionales básicos

Pablo Huerta Pellitero

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



CE

grafía

cción

adores y decodificadores

tesis de funciones de conmutación con decodificadores

lexores y demultiplexores

tesis de circuitos combinacionales con multiplexores

zadores

adores

itivos lógicos programables

os aritméticos básicos

mador

stador

mador/Restador

d aritmético-lógica

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



BIBLIOGRAFÍA

Hermida, Ana M^o del Corral, Enric Pastor, Fermín Sánchez

Elementos de Computadores” , cap 3

Prentice Hall

James L. Floyd

Elementos de Sistemas Digitales”, cap 6,7

Prentice Hall

D. Gajski

Principios de Diseño Digital”, cap 5

Prentice Hall

Thomas Harris Mano

Arquitectura Digital”, cap 4,5

Prentice Hall

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagenanet.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.



PRODUCCIÓN

terializaciones en forma de redes de puertas básicas no son
adadas cuando la complejidad del diseño es grande. **En estos casos se
diseño jerárquico en base a redes de módulos combinacionales**
s, y no mediante redes de puertas básicas.

eño jerárquico y modular se puede llevar a cabo si se dispone de
os que realicen funciones más complejas que las puertas básicas y
rmitan dividir el diseño en partes más sencillas.

ue sigue de tema se explican algunos de los módulos
nacionales de los que es conveniente disponer y cómo utilizarlos
módulos básicos para realizar diseños más complejos.

hódulos son: decodificador, codificador, multiplexor, demultiplexor,
zador, comparador, algunos módulos aritméticos y dispositivos
mables.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



CE

rafía

ucción

ADORES Y DECODIFICADORES

SÍNTESIS DE FUNCIONES DE CONMUTACIÓN CON DECODIFICADORES

Multiplexores y demultiplexores

Síntesis de circuitos combinacionales con multiplexores

Encodificadores

Decodificadores

Dispositivos lógicos programables

Algoritmos aritméticos básicos

Sumador

Restador

Sumador/Restador

Arquitectura aritmético-lógica

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

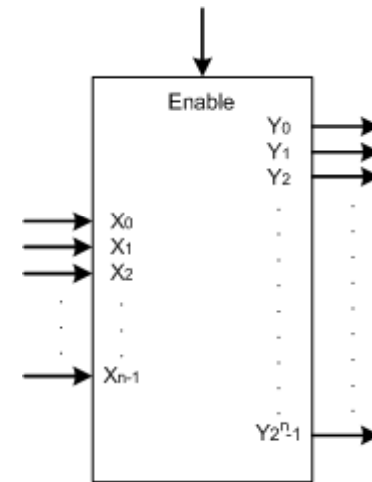
CODIFICADORES

Un codificador (o decodificador de n a 2^n) es un módulo combinacional con n entradas (X_0, X_1, \dots, X_{n-1}), y 2^n salidas (Y_0, Y_1, Y_2, \dots), además de una línea de activación (Enable).

El funcionamiento del decodificador es el siguiente:

Si la línea Enable está desactivada todas las salidas Y_i toman el valor '0'.

Si la línea Enable está activada se activará la salida de índice i correspondiente al número decimal codificado en las entradas X . El resto salidas toman valor '0'.



La expresión de conmutación que describe cada una de las salidas es:

$$Y_i = Enable \cdot m_i(X_{n-1}, \dots, X_0)$$

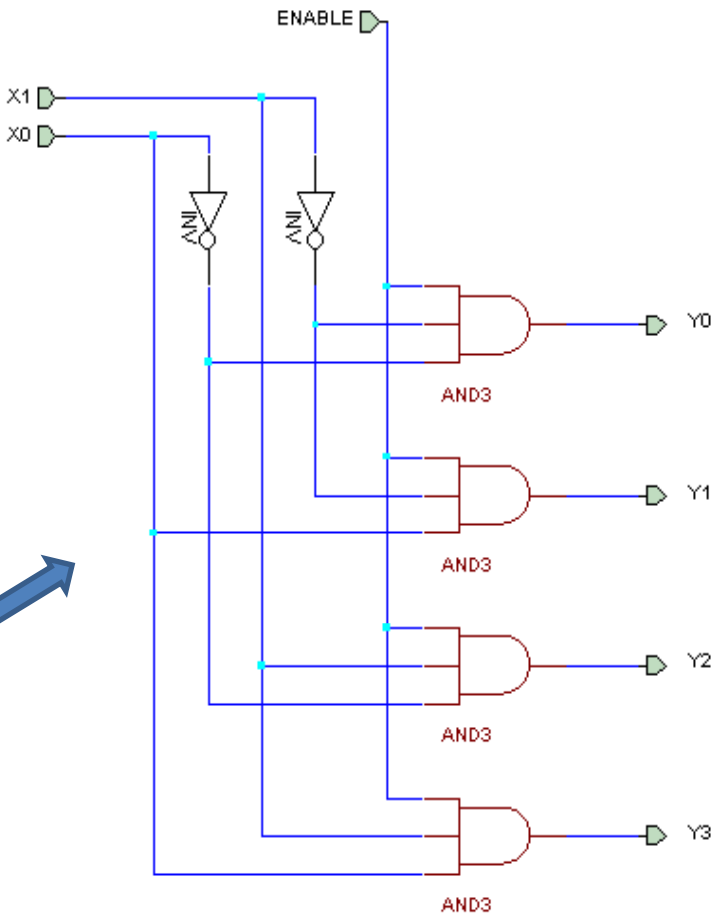


CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

DECODIFICADORES

o: implementación de un decodificador de 2 a 4 con puertas



	Y ₃	Y ₂	Y ₁	Y ₀
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	1	0	0	0
1	0	0	0	0

$$\begin{aligned}
 Y_0 &= X_1 \cdot X_0 \\
 Y_1 &= X_1 \cdot \overline{X_0} \\
 Y_2 &= \overline{X_1} \cdot X_0 \\
 Y_3 &= \overline{X_1} \cdot \overline{X_0}
 \end{aligned}$$

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



MODIFICADORES

o (continuación): descripción VHDL.

```

use ieee;
use ieee.std_logic_1164.all;

entity deco2a4 is
  port (enable: in std_logic;
        x: in std_logic_vector(1 downto 0);
        y: out std_logic_vector(3 downto 0));
end deco2a4;

architecture puertas of deco2a4 is
  signal not_x0, not_x1: std_logic;

  1: entity work.not1 port map(x(0), not_x0);
  2: entity work.not1 port map(x(1), not_x1);
  1: entity work.and3 port map(enable, not_x0, not_x1, y(0));
  2: entity work.and3 port map(enable, x(0), not_x1, y(1));
  3: entity work.and3 port map(enable, not_x0, x(1), y(2));
  4: entity work.and3 port map(enable, x(0), x(1), y(3));
end puertas;

```

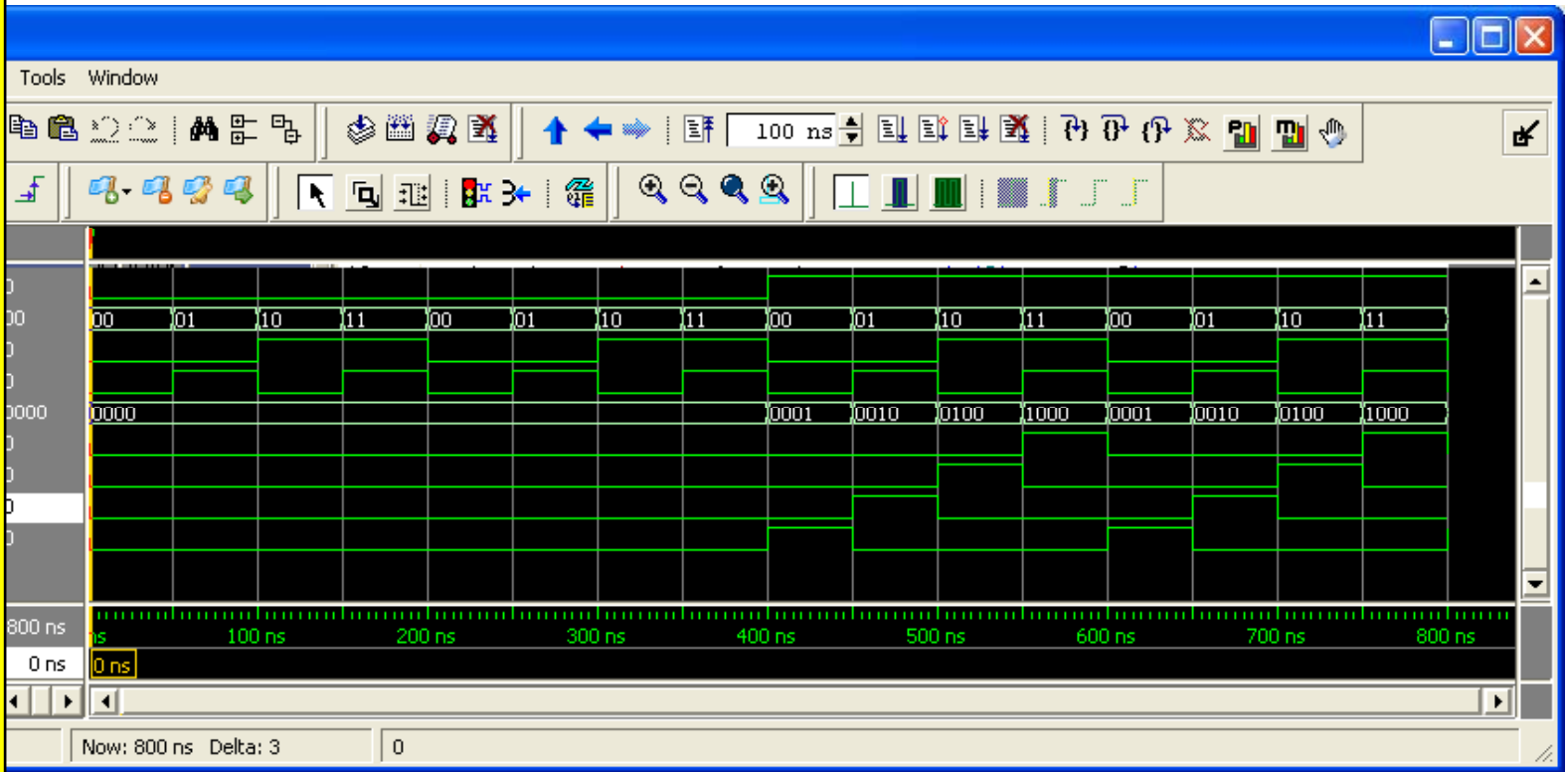
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



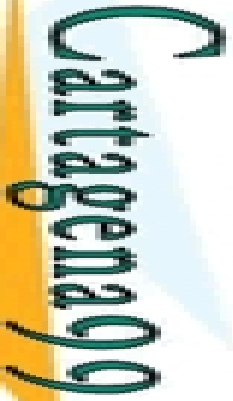
CODIFICADORES

o (continuación): resultado de la simulación.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

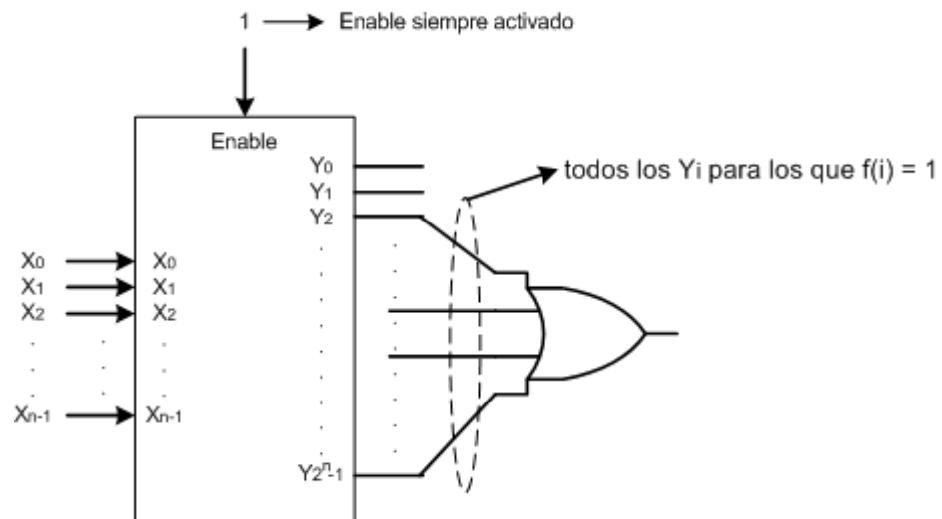


ANÁLISIS DE FC CON DECODIFICADORES

Un decodificador de n a 2^n materializa todos los minterms de una función de n variables:

$$Y_i = Enable \cdot m_i(X_{n-1}, \dots, X_0)$$

Por tanto se puede materializar cualquier FC de n variables expresada como una suma de minterms con **un decodificador de n a 2^n y una puerta Or** de tantas entradas como sumandos tenga la expresión de la FC.



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70



ANÁLISIS DE FC CON DECODIFICADORES

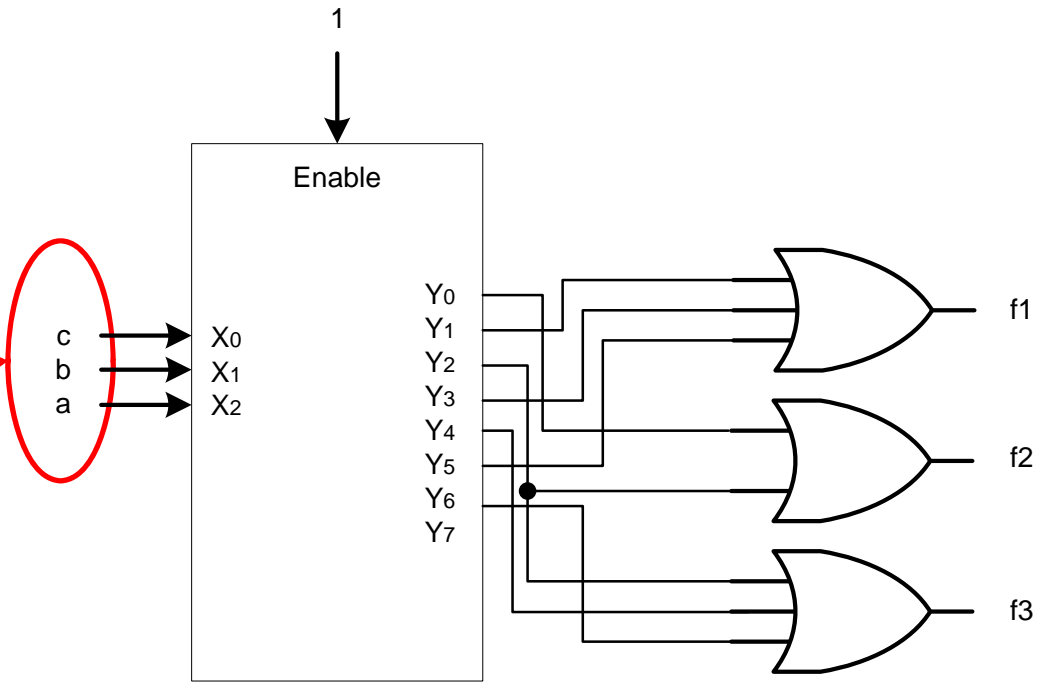
Objetivo: sintetizar las funciones f_1 , f_2 y f_3 con un decodificador de 3 a 8.

$$f_1(b, c) = \sum m(1,3,5)$$

$$f_2(b, c) = \sum m(0,2)$$

$$f_3(b, c) = \sum m(2,4,6)$$

IMPORTANTE:
Conectar con el orden de
prioridad de las entradas



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



IFICADORES SIN PRIORIDAD

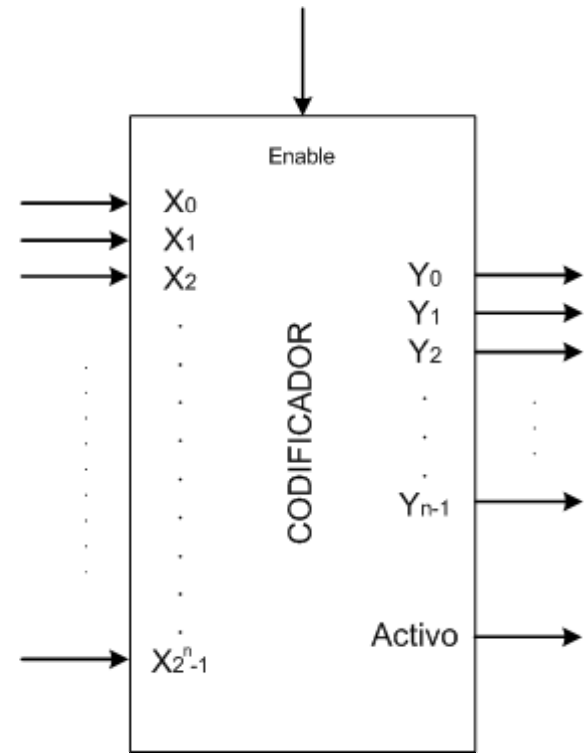
Un codificador (codificador de 2^n a n) es un módulo con 2^n entradas y n salidas de datos, una entrada de activación (Enable) y una salida de estado para diseño modular.

El funcionamiento del codificador sin prioridad es el siguiente:

Si la entrada Enable está desactivada todas las salidas Y_i tendrán el valor '0'.

Si la entrada Enable está activada las salidas Y_i codificarán el número I , siendo I el índice de la entrada X_i que esté activa (sólo una entrada X puede valer '1' en un instante determinado)

La salida Activo vale '1' si alguna de las entradas X_i vale '1' y Enable está activa. En caso contrario la salida Activo vale '0'.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

DECODIFICADORES SIN PRIORIDAD

Expresiones de conmutación de las salidas Y_i y Activo son:

$$Y_j = E \cdot \sum X_i \quad / \quad X_j \text{ se incluye en la suma si el bit } i\text{-ésimo de la representación binaria de } j \text{ es } 1$$

$$E = (X_{2^{n-1}} + X_{2^{n-2}} + \dots + X_1 + X_0)$$

Ejemplo: codificador de 8 a 3:

Entrada	Y_2	Y_1	Y_0	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	0	1
4	0	0	1	1
5	1	0	0	1
6	1	0	1	1
7	1	1	0	1
7	1	1	1	1

$$y_0 = E \cdot (x_1 + x_3 + x_5 + x_7)$$

$$y_1 = E \cdot (x_2 + x_3 + x_6 + x_7)$$

$$y_2 = E \cdot (x_4 + x_5 + x_6 + x_7)$$

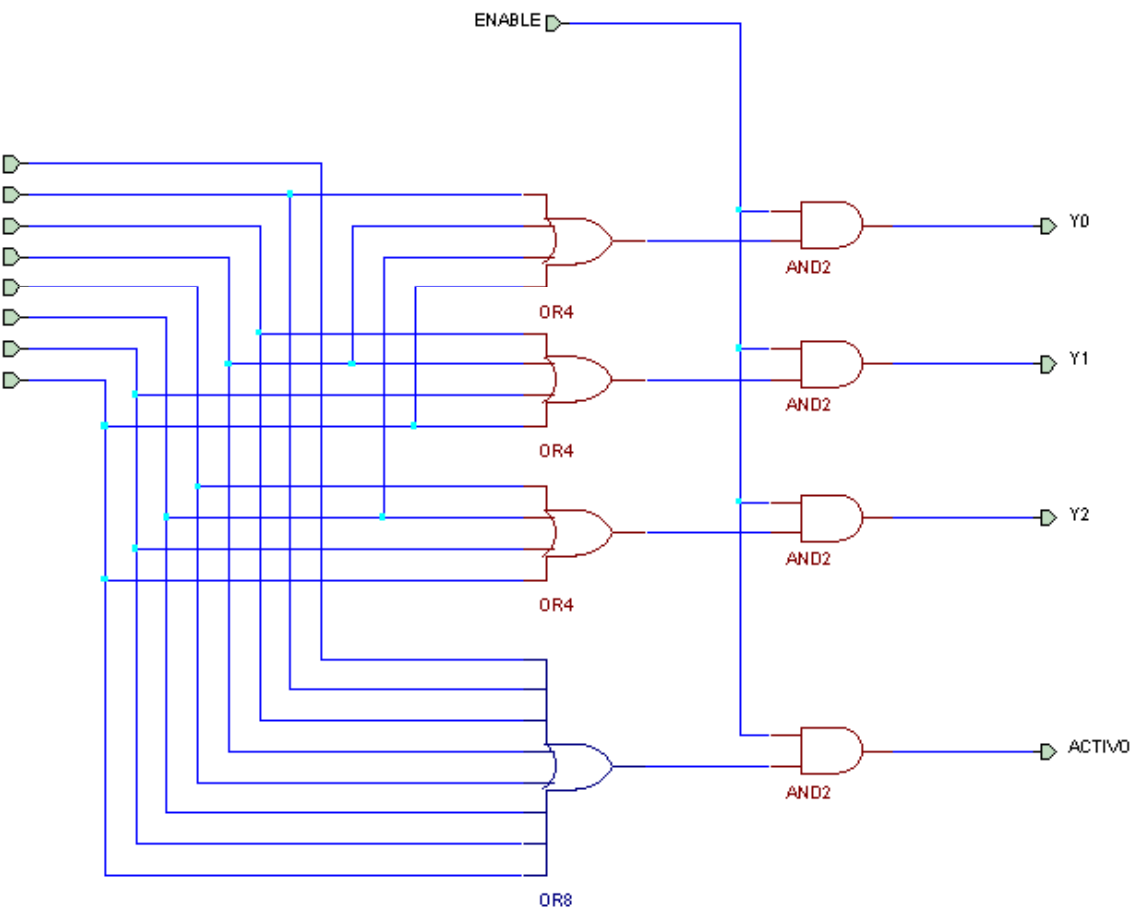
$$A = E \cdot (x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7)$$



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

DECODIFICADORES SIN PRIORIDAD

o: codificador de 8 a 3 (continuación):



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

IFICADORES SIN PRIORIDAD

o (continuación): descripción estructural en VHDL

```

entity ieee;
    use ieee.std_logic_1164.all;

    entity deco2a4 is
        port (enable: in std_logic;
              x: in std_logic_vector(1 downto 0);
              y: out std_logic_vector(3 downto 0));
    end deco2a4;

    architecture puertas of deco2a4 is
        signal not_x0, not_x1: std_logic;

        1: entity work.not1 port map(x(0), not_x0);
        2: entity work.not1 port map(x(1), not_x1);
        1: entity work.and3 port map(enable, not_x0, not_x1, y(0));
        2: entity work.and3 port map(enable, x(0), not_x1, y(1));
        3: entity work.and3 port map(enable, not_x0, x(1), y(2));
        4: entity work.and3 port map(enable, x(0), x(1), y(3));
    end puertas;

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



IFICADORES SIN PRIORIDAD

o (continuación): otra posible descripción del circuito.

Structure concurrente of cod8a3 is

```

) <= enable and (x(1) or x(3) or x(5) or x(7));
) <= enable and (x(2) or x(3) or x(6) or x(7));
) <= enable and (x(4) or x(5) or x(6) or x(7));
ativo <= enable and (x(0) or x(1) or x(2) or x(3) or x(4) or
x(5) or x(6) or x(7));
ncurrente;

```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

IFICADORES SIN PRIORIDAD

o (continuación): test-bench.

```

ry ieee;
eee.std_logic_1164.all;

y test_deco2a4 is
est_deco2a4;

--
structure test of test_deco2a4 is

enable: std_logic;
x: std_logic_vector(1 downto 0) := "00";
y: std_logic_vector(3 downto 0);

entity work.deco2a4(puertas) port map(enable, x, y);
e <= '0', '1' after 400 ns;
x = not x(0) after 50 ns;
x = not x(1) after 100 ns;

est;

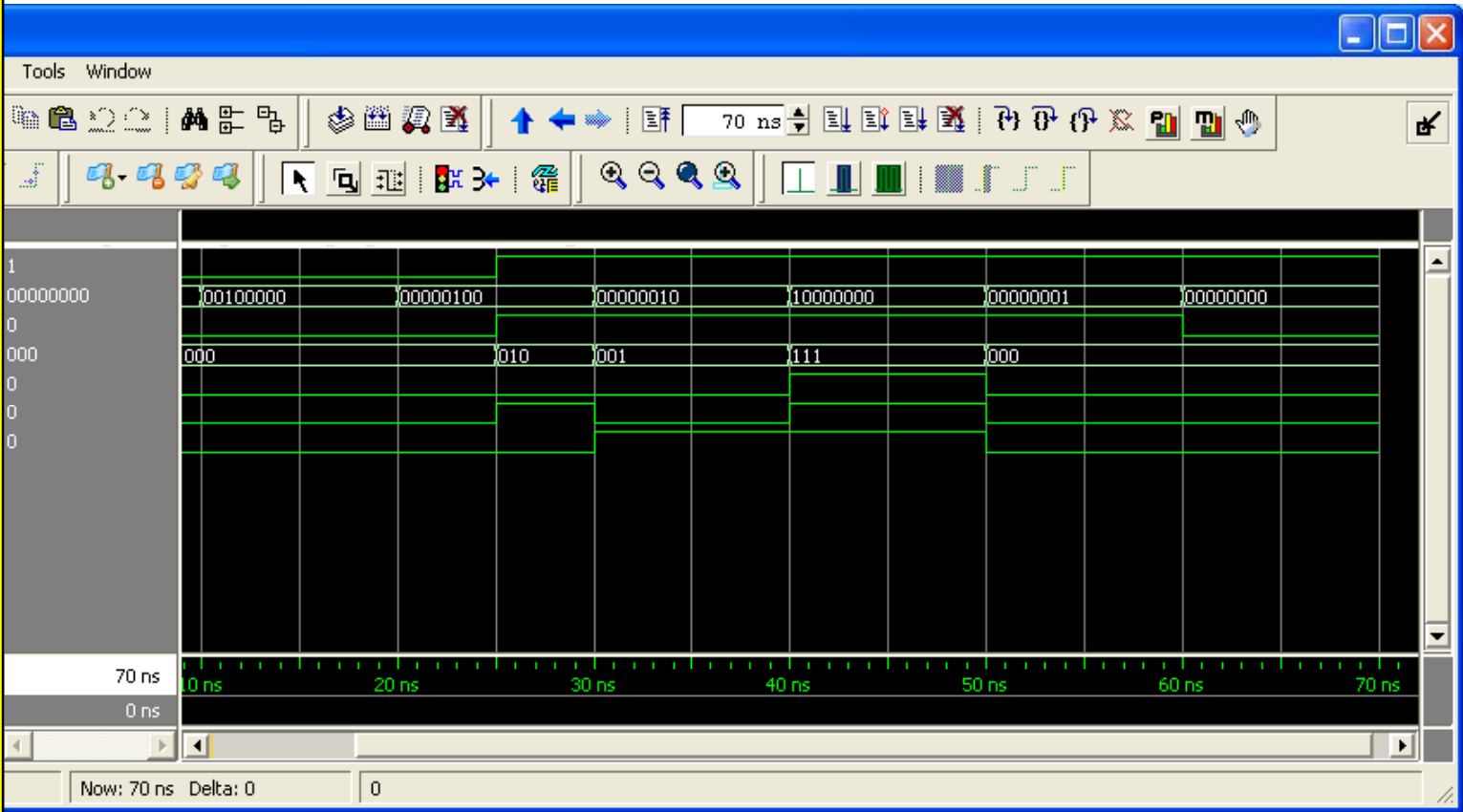
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

IFICADORES SIN PRIORIDAD

o (continuación): resultado de la simulación.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
-- --
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70



IFICADORES CON PRIORIDAD

Los codificadores con prioridad reciben varias entradas de las cuales pueden estar activas más de una a la vez. La salida codifica el índice de la entrada activa más alta (o más baja si es un codificador de bit menos significativo).

Internamente realizan una red de resolución de prioridades que decide cuál de las entradas tiene más prioridad.

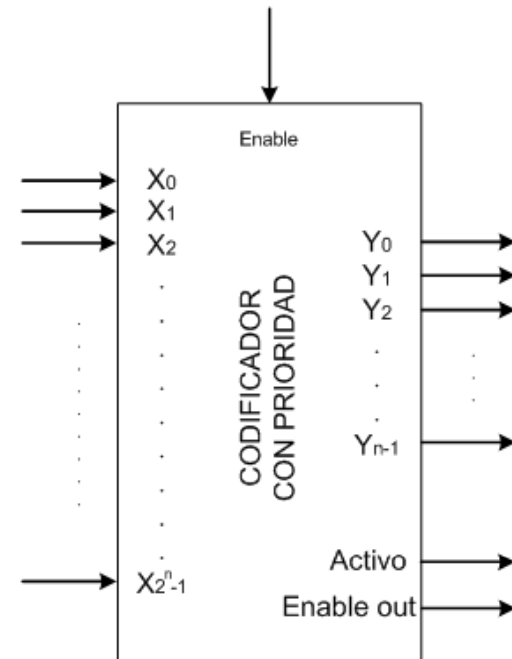
Funcionamiento:

Si Enable está inactiva: Y_i , Activo y Enable out serán todas '0'.

Si Enable está activa:

Si hay algún X_i activo, la salida Y codifica el índice del mayor X_i que esté activo. La salida Activo vale '1' y la salida Enable out vale '0'.

Si no hay ningún X_i activo, la salida Activo vale '0' y la salida Enable out vale '1'.

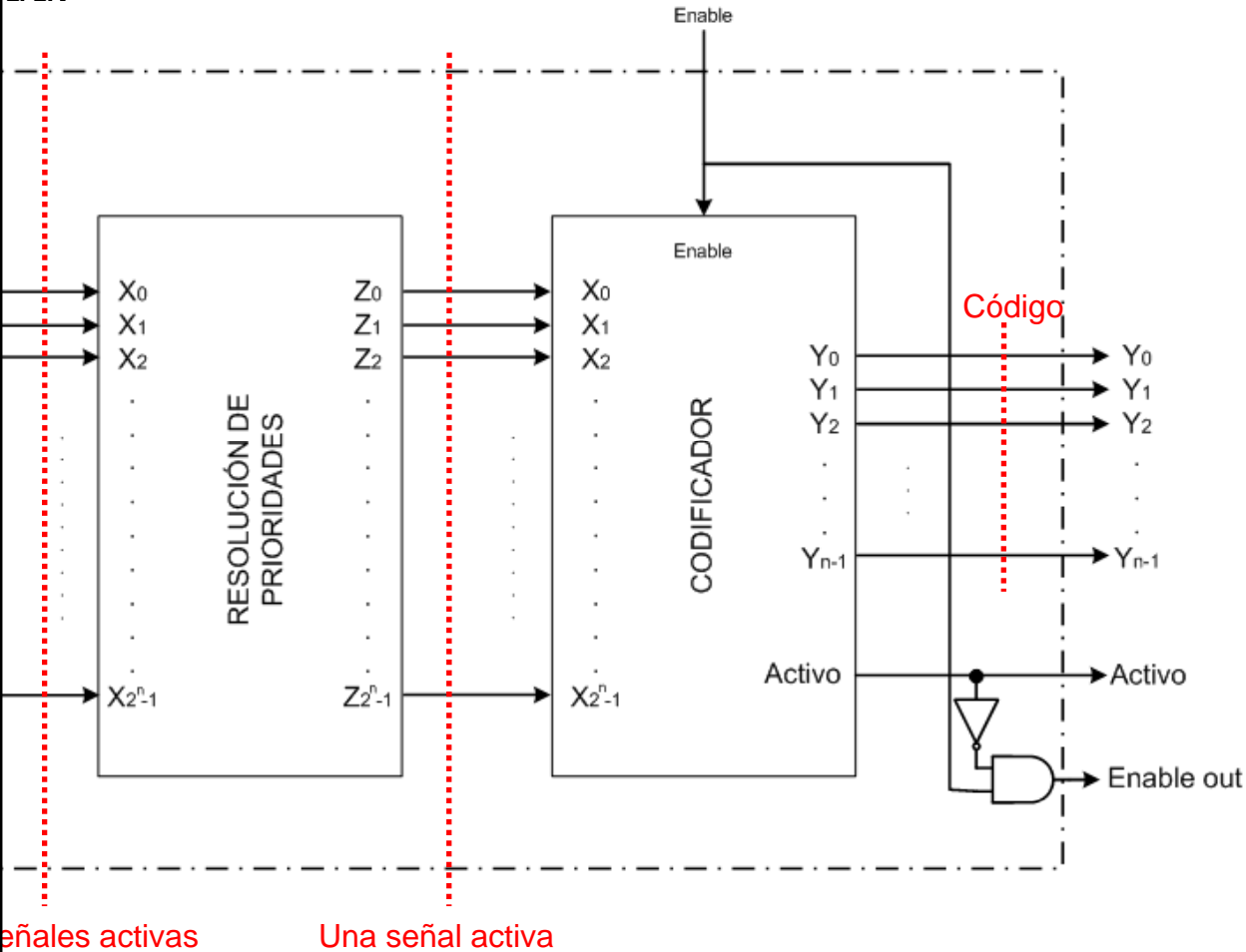


CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

IFICADORES CON PRIORIDAD

o: codificador con prioridad en función de un codificador sin

ad.



- - -

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**



IFICADORES CON PRIORIDAD

ulo de resolución de prioridades de bit más significativo se realiza con las siguientes expresiones de conmutación:

$$Z_{2^n-1} = X_{2^n-1}$$

$$Z_{2^n-2} = X_{2^n-2} \cdot \overline{X_{2^n-1}}$$

$$Z_{2^n-3} = X_{2^n-3} \cdot \overline{X_{2^n-2}} \cdot \overline{X_{2^n-1}}$$

.

.

$$Z_i = X_i \cdot \overline{X_{i+1}} \cdot \dots \cdot \overline{X_{2^n-2}} \cdot \overline{X_{2^n-1}}$$



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 - - -
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

DECODIFICADORES CON PRIORIDAD



Ejemplo: sintetizar un codificador con prioridad de 4 a 2.

Tabla de resolución de prioridades:

X_3
$X_2 \cdot \overline{X_3}$
$X_1 \cdot \overline{X_2} \cdot \overline{X_3}$
$X_0 \cdot \overline{X_1} \cdot \overline{X_2} \cdot \overline{X_3}$

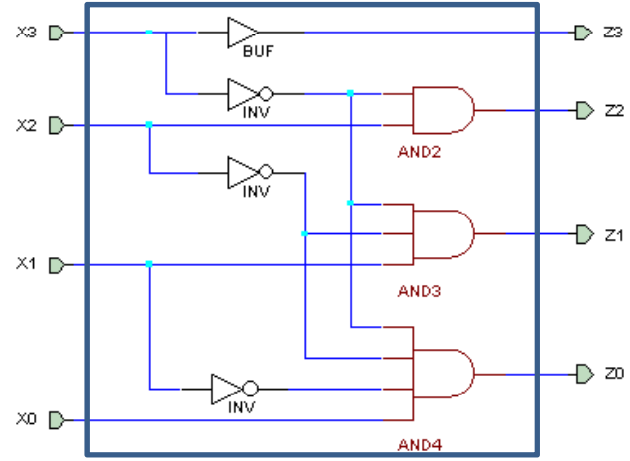
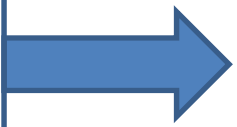
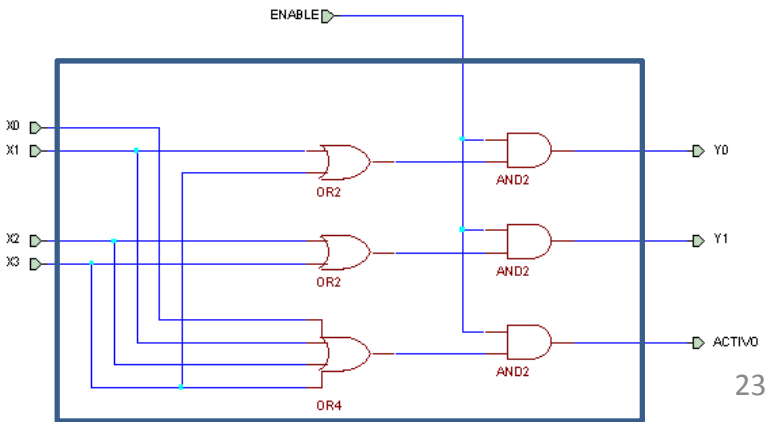


Tabla de resolución de prioridades:

$(X_1 + X_3)$
$(X_2 + X_3)$
$(X_0 + X_1 + X_2 + X_3)$

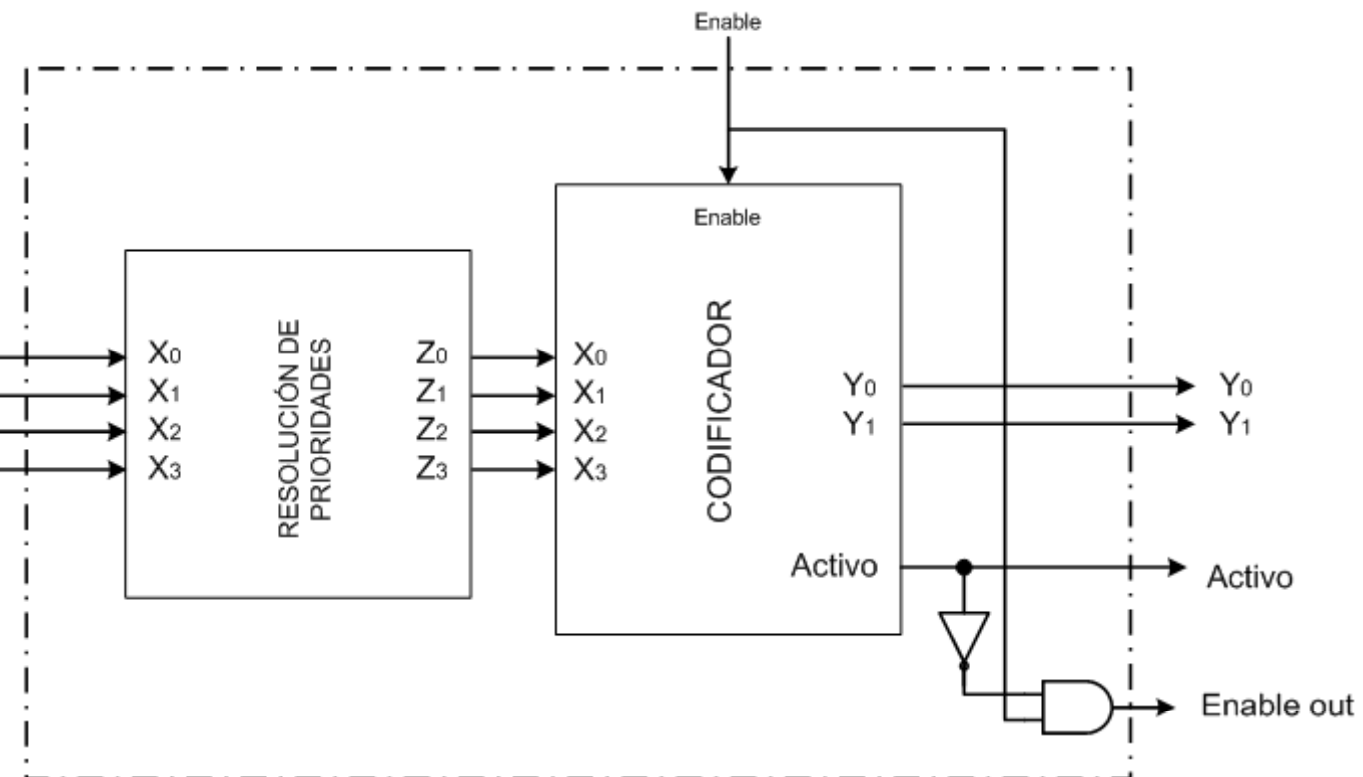


CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

IFICADORES CON PRIORIDAD



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



CE

rafía

ucción

adores y decodificadores

tesis de funciones de conmutación con decodificadores

lexores y demultiplexores

tesis de circuitos combinacionales con multiplexores

zadores

radores

itivos lógicos programables

os aritméticos básicos

mador

stador

mador/Restador

d aritmético-lógica



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



.TIPLEXORES

Un multiplexor (o multiplexor de 2^n a 1) es un módulo combinacional con 2^n entradas de datos, 'n' entradas de selección y una entrada de habilitación (Enable). Tiene una única salida.

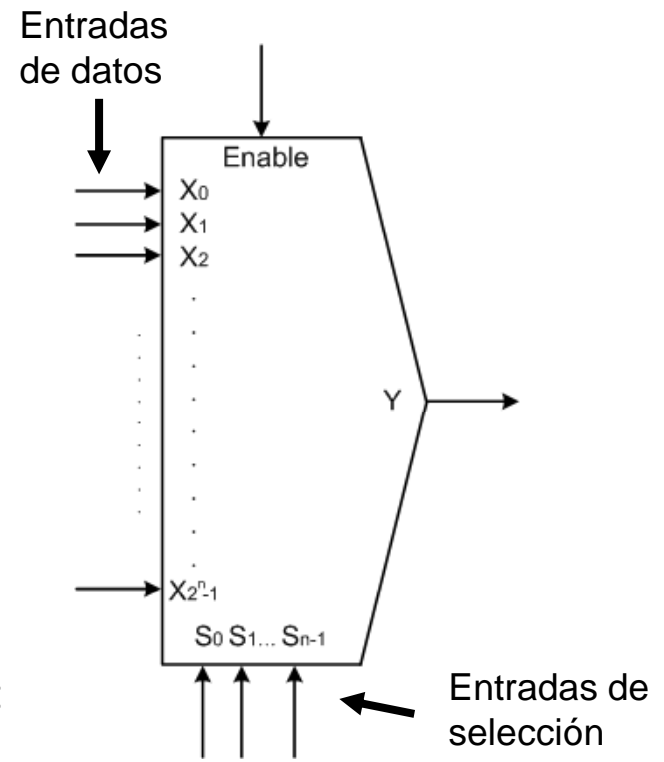
El multiplexor conecta una de las 2^n entradas de datos a la salida en función de una palabra de selección S:

Si las entradas de selección S_i codifican el número I, la salida Y tomará el valor de X_i siempre que Enable esté activo.

Si Enable está inactivo la salida Y vale '0'.

La función de salida en forma de expresión de conmutación:

$$Y = x_0 \cdot m_0(s_{n-1}, \dots, s_0) + x_1 \cdot m_1(s_{n-1}, \dots, s_0) + \dots = E \cdot \sum_{i=0}^{2^n-1} x_i \cdot m_i(s_{n-1}, \dots, s_0)$$



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

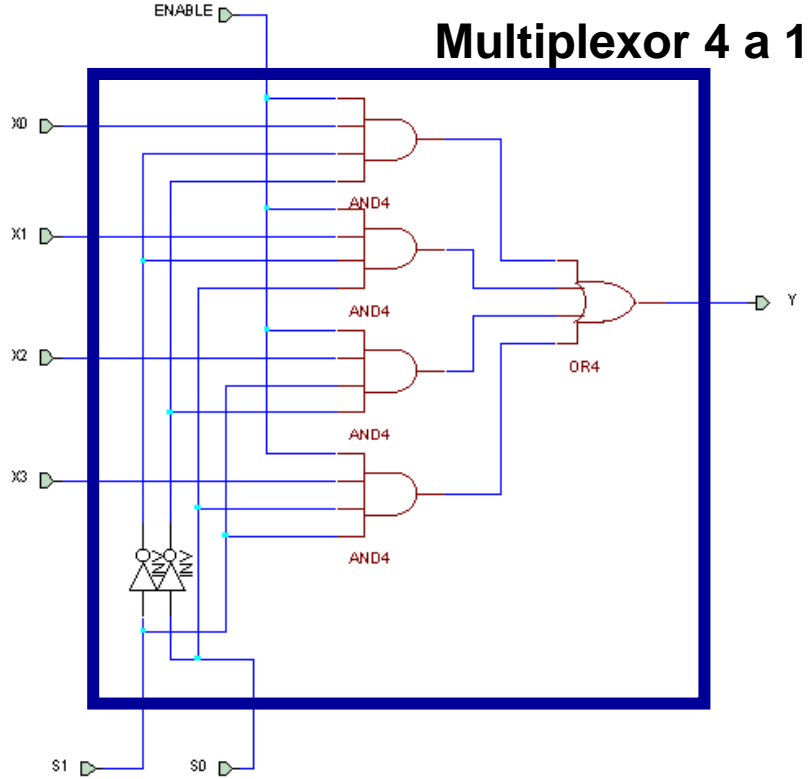


.TIPLEXORES

o: implementar un multiplexor de 4 entradas.

$$Y = x_0 \cdot m_0(s_1, s_0) + x_1 \cdot m_1(s_1, s_0) + x_2 \cdot m_2(s_1, s_0) + x_3 \cdot m_3(s_1, s_0) =$$

$$x_0 \cdot \overline{s_1} \cdot \overline{s_0} + x_1 \cdot \overline{s_1} \cdot s_0 + x_2 \cdot s_1 \cdot \overline{s_0} + x_3 \cdot s_1 \cdot s_0$$



- - -

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.



.TIPLEXORES

o (continuación): descripción estructural en VHDL

```

eee;
std_logic_1164.all;

mux4a1 is
enable: in std_logic;
sel: in std_logic_vector(3 downto 0);
sel: in std_logic_vector(1 downto 0);
out: out std_logic);
end mux4a1;

architecture puertas of mux4a1 is
not_sel1, not_sel0: std_logic;
s0, s1, s2, s3: std_logic;

entity work.not1 port map(sel(0), not_sel0);
entity work.not1 port map(sel(1), not_sel1);
entity work.and4 port map(enable, not_sel0, not_sel1, x(0), s0);
entity work.and4 port map(enable, sel(0), not_sel1, x(1), s1);
entity work.and4 port map(enable, not_sel0, sel(1), x(2), s2);
entity work.and4 port map(enable, sel(0), sel(1), x(3), s3);
entity work.or4 port map(s0, s1, s2, s3, y);
end architecture puertas;

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

.TIPLEXORES

o (continuación): otra posible descripción en VHDL

```

structure concurrente of mux4a1 is
  = enable and (
    ( not sel(0)) and (not sel(1)) and x(0) ) or
    ( sel(0) and (not sel(1)) and x(1) ) or
    ( (not sel(0)) and sel(1) and x(2) ) or
    ( sel(0) and sel(1) and x(3) )
  );
concurrente;

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



.TIPLEXORES

o (continuación): y otra posible descripción más en VHDL utilizando **process**.

```

architecture funcional of mux4a1 is
  process(x, sel)
  begin
    if enable = '0' then
      y <= '0';
    else
      case sel is
        when "00" => y <= x(0);
        when "01" => y <= x(1);
        when "10" => y <= x(2);
        when others => y <= x(3);
      end case;
    end if;
  end process;
end funcional;

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70





.TIPLEXORES

o (continuación): test-bench

```

eee;
std_logic_1164.all;

test_mux4a1 is
mux4a1;

architecture test of test_mux4a1 is
enable, y: std_logic;
x: std_logic_vector(3 downto 0) := "0000";
sel: std_logic_vector(1 downto 0);

entity work.mux4a1(puertas) port map(enable, x, sel, y);

x <= '1', '0' after 400 ns;
enable not x(0) after 2 ns;
enable not x(1) after 4 ns;
enable not x(2) after 8 ns;
enable not x(3) after 16 ns;
enable "00", "01" after 100 ns, "10" after 200 ns, "11" after 300 ns;

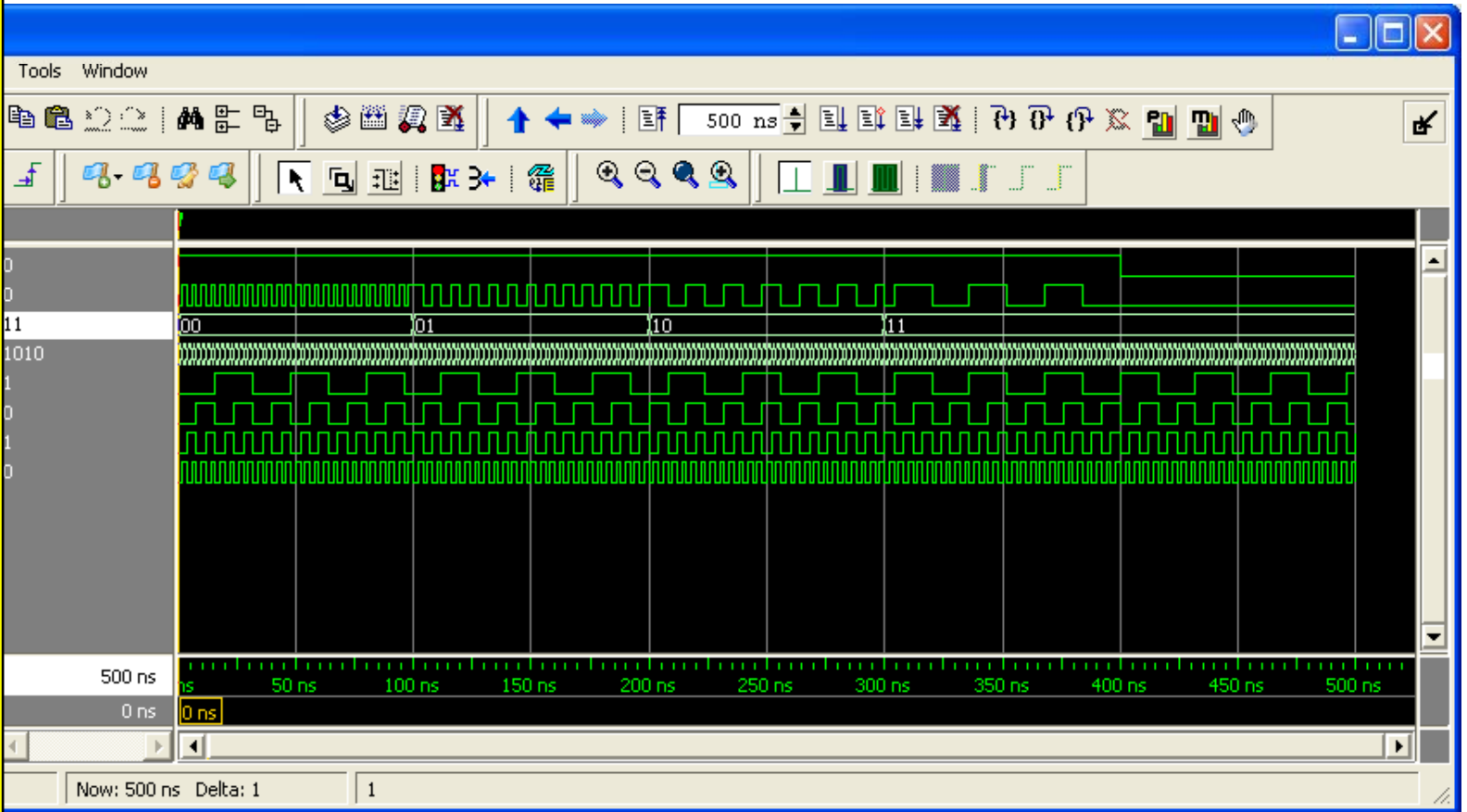
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

.TIPLEXORES

o (continuación): resultado de la simulación



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
-- --
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



ANÁLISIS DE FC CON MULTIPLEXORES

Un multiplexor de 2^n a 1 permite materializar cualquier función de combinación de 'n' variables.

Una FC como suma de productos consiste en la suma de los minterms m_i para los que la FC, $f(i)$ toma valor cierto, es decir:

$$f(a_{n-1}, \dots, a_0) = \sum_{i=0}^{2^n-1} f(i) \cdot m_i(a_{n-1}, \dots, a_0)$$

Un multiplexor es:

$$y = E \cdot \sum_{i=0}^{2^n-1} x_i \cdot m_i(s_{n-1}, \dots, s_0)$$

Entonces, podemos hacer que $y = f(a_{n-1}, \dots, a_0)$ si hacemos:

Enable = '1'.

Las entradas S_i sean las variables de la función.

Cada entrada X_i del multiplexor sea igual a $f(i)$ (el valor de la fila i de la tabla de verdad).



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

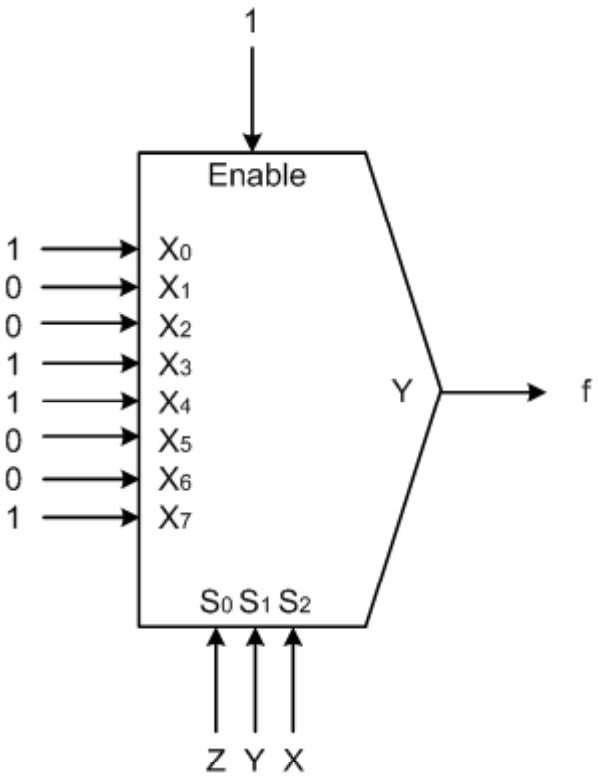
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

ANÁLISIS DE FC CON MULTIPLEXORES

o: sintetizar $f(x,y,z)$ mediante un único multiplexor.

$$f(x, y, z) = \sum m(0,3,4,7)$$

Z	F
0	1
1	0
0	0
1	1
0	1
1	0
0	1
1	0



ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

 CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

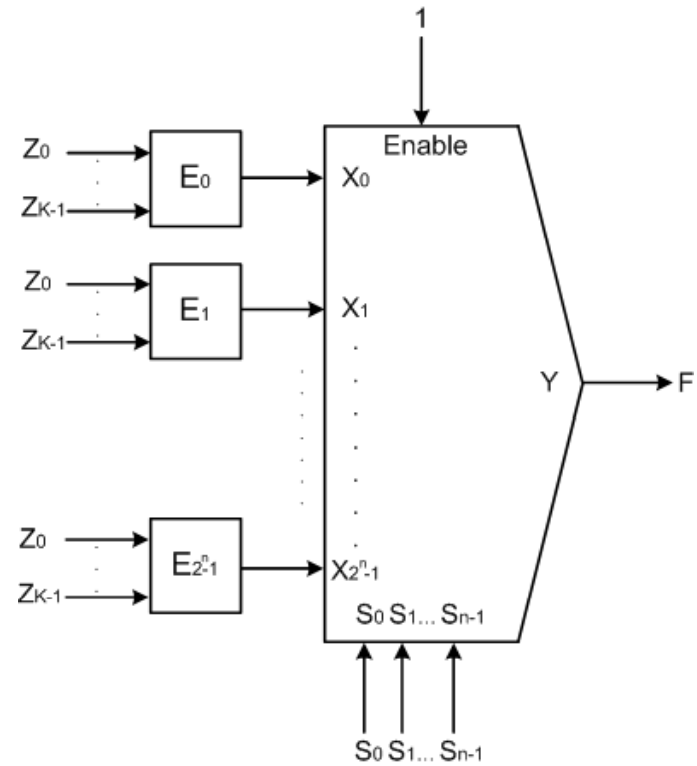




ANÁLISIS DE FC CON MULTIPLEXORES

La implementación de FCs mediante multiplexores no es directa cuando no dispone de un multiplexor con tantas entradas de control como variables binarias tiene la FC.

Se utiliza un multiplexor de 2^n a 1 (es decir, con 'n' entradas de selección) para implementar una función con 'n+k' variables binarias, se seleccionan 'n' variables binarias como entradas de control y se usan las restantes como variables de función de entrada al multiplexor.



$$F(z_{k-1}, \dots, z_0, x_{n-1}, \dots, x_0) = f(z_{k-1}, \dots, z_0, s_{n-1}, \dots, s_0) = \sum_{i=0}^{2^n-1} E_i(z_{k-1}, \dots, z_0) \cdot m_i(s_{n-1}, \dots, s_0)$$

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70



ANÁLISIS DE FC CON MULTIPLEXORES

Una función de k entradas puede realizarse con multiplexores de k bits de control (árbol de multiplexores), con decodificadores, con puertas lógicas, etc.

Ejemplo: implementar f mediante un multiplexor de 4 a 1 y puertas lógicas. (Se ha escogido c y d como entradas de selección del multiplexor)

$$f(a,b,c,d) = \sum m(1,3,4,6,7,9,10,11,14)$$

$$\bar{a}\bar{b}\bar{c}d + \bar{a}\bar{b}c\bar{d} + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + a\bar{b}\bar{c}\bar{d} + a\bar{b}\bar{c}d + a\bar{b}c\bar{d} + a\bar{b}cd =$$

$$(\bar{a}\bar{b} + a\bar{b})\bar{c}d + (\bar{a}\bar{b} + a\bar{b} + a\bar{b})\bar{c}\bar{d} + (\bar{a}\bar{b} + a\bar{b} + a\bar{b})\bar{c}d$$

E1

E2

E3

$$\left. \begin{aligned} a \cdot \bar{b} &= \bar{b} \\ a \cdot \bar{b} + a \cdot b &= a + \bar{b} \\ \bar{a} \cdot b + a \cdot \bar{b} &= \bar{a} + \bar{b} \end{aligned} \right\}$$

Simplificación algebraica

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

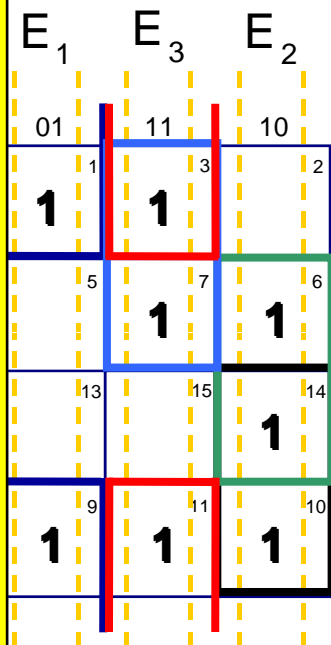


ANÁLISIS DE FC CON MULTIPLEXORES

Este análisis también se puede resolver de forma gráfica con mapas de Karnaugh.

Se trata de dividir el mapa en regiones en las que las variables que se usan como entradas de selección del multiplexor tienen valores constantes.

Ejemplo: para la misma función que la transparencia anterior:



Se divide el mapa en las regiones en las que 'c' y 'd' valen '00' => E0; '01' => E1, etc.

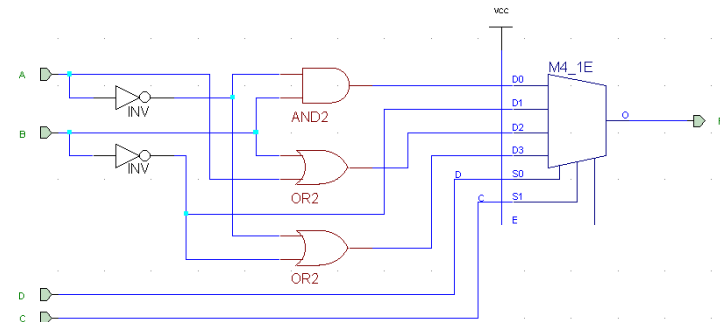
Cada región se trata como un mapa de Karnaugh de 2 variables ('a' y 'b')

$$E_0 = \bar{a} \cdot \bar{b}$$

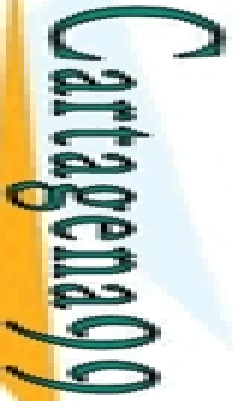
$$E_1 = \bar{a} \cdot b$$

$$E_2 = a \cdot \bar{b}$$

$$E_3 = a \cdot b$$



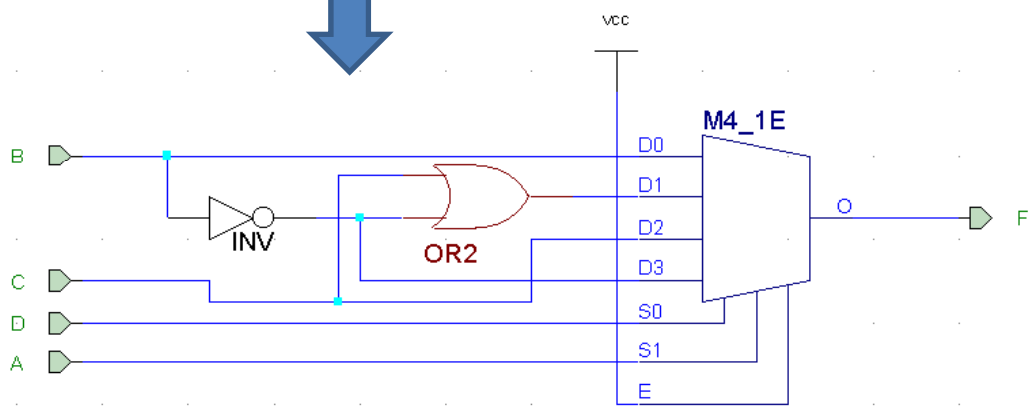
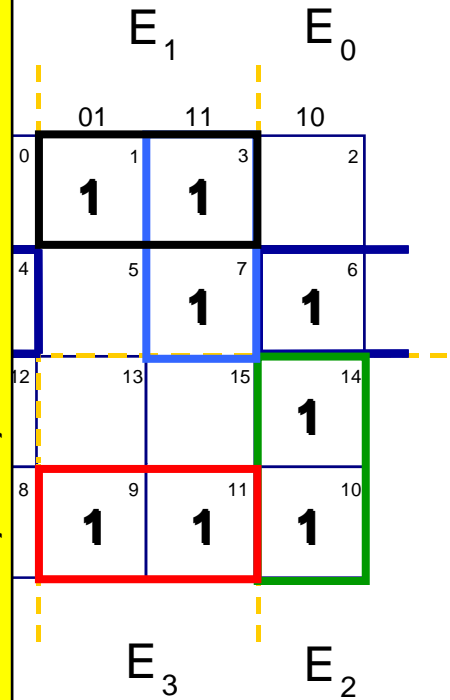
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70



DESIGN DE FC CON MULTIPLEXORES

Escogen otras variables de selección ('a' y 'd' en este ejemplo) el resultado resultante es diferente.

$$\begin{aligned}
 E_0 &= b \\
 E_1 &= \bar{b} + c \\
 E_2 &= c \\
 E_3 &= \bar{b}
 \end{aligned}$$



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

ANÁLISIS DE FC CON MULTIPLEXORES

o: descripción en VHDL de los dos circuitos anteriores.

```

entity ieee;
    port (
        eee: std_logic_1164.all;
    );
end ieee;

entity sintesis_mux is
    port (
        a,b,c,d: in std_logic;
        z: out std_logic);
end sintesis_mux;

architecture arq_1 of sintesis_mux is
    signal E: std_logic_vector(3 downto 0);
    signal sel: std_logic_vector(1 downto 0);

    E(0) <= (not a) and b;
    E(1) <= (not b);
    E(2) <= a or b;
    E(3) <= (not a) or (not b);
    z(0) <= d;
    z(1) <= c;
end arq_1;

entity work.mux4a1(funcional) port map('1', E, sel, z);
end work.mux4a1;

```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

ANÁLISIS DE FC CON MULTIPLEXORES

o (continuación): descripción en VHDL de los dos circuitos

pres.

```

architecture arq_2 of sintesis_mux is
  type E: std_logic_vector(3 downto 0);
  type sel: std_logic_vector(1 downto 0);
  signal z: std_logic;
  z <= b;
  z <= (not b) or c;
  z <= c;
  z <= (not b);
  z(0) <= d;
  z(1) <= a;
  entity work.mux4a1(concurrente) port map('1', E, sel, z);
  end arq_2;

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



ANÁLISIS DE FC CON MULTIPLEXORES

o (continuación): test-bench de las dos arquitecturas

```

library ieee;
use ieee.std_logic_1164.all;

entity test_sintesis_mux is
test_sintesis_mux;

architecture test of test_sintesis_mux is
    signal a: std_logic := '0';
    signal b: std_logic := '0';
    signal c: std_logic := '0';
    signal d: std_logic := '0';
    signal z1, z2: std_logic;

    t_1: entity work.sintesis_mux(arq_1) port map(a,b,c,d,z1);
    t_2: entity work.sintesis_mux(arq_2) port map(a,b,c,d,z2);

    not d after 2 ns;
    not c after 4 ns;
    not b after 8 ns;
    not a after 16 ns;

end test;

```

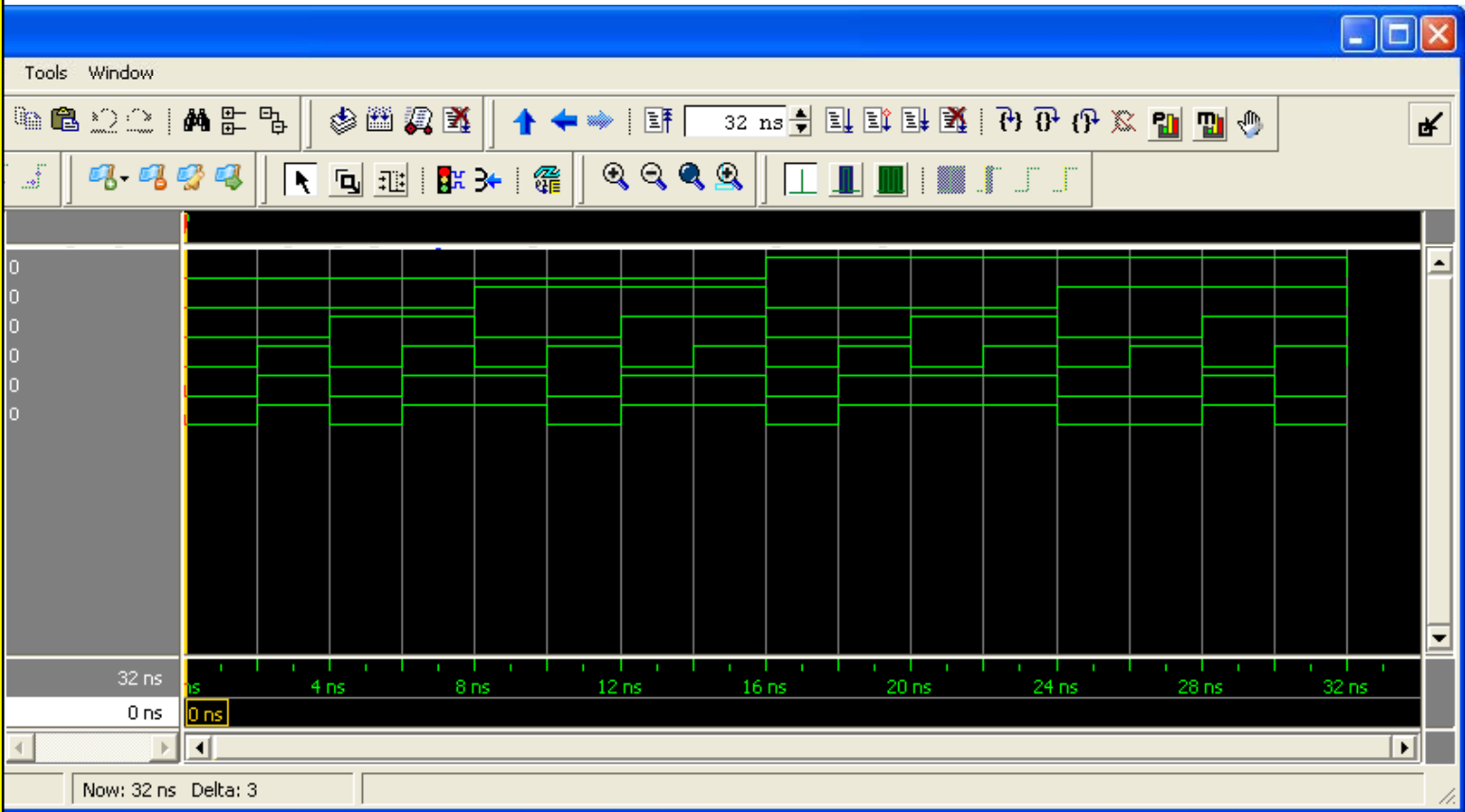
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



ANÁLISIS DE FC CON MULTIPLEXORES

o (continuación): resultado de la simulación



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70





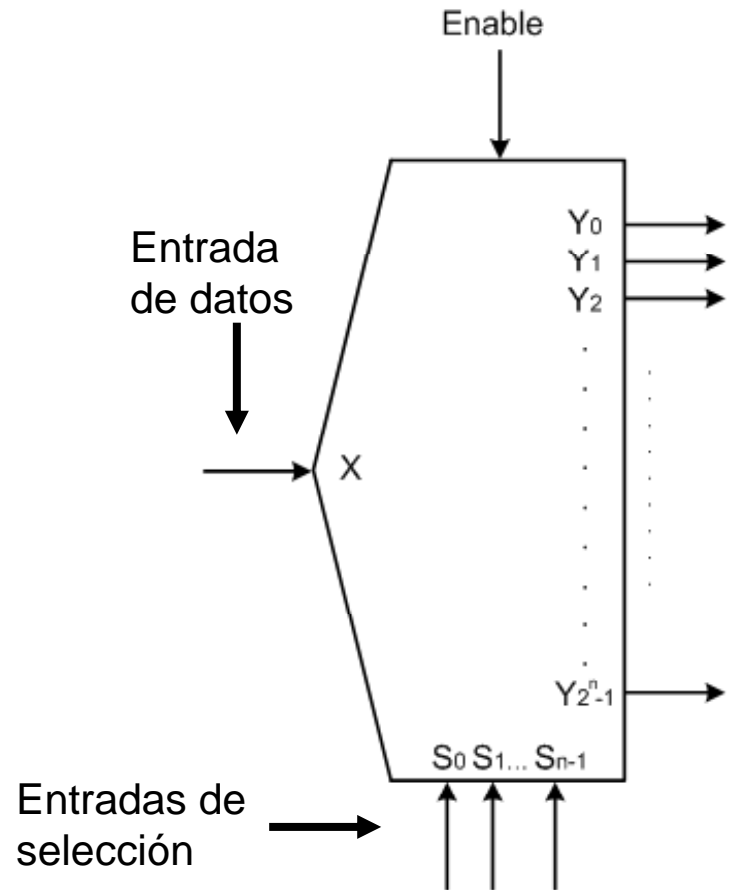
MULTIPLEXORES

un multiplexor (o demultiplexor de 1 a 2^n) es un módulo combinacional con una entrada y 2^n salidas, además de una entrada de activación y 'n' entradas de control.

La salida de índice 'i' vale lo mismo que la entrada X, siempre que Enable esté activo y las entradas de selección estén modificando el número decimal 'i'. Las salidas que no están seleccionadas valen '0'.

La función de activación de un multiplexor se expresa en forma de expresión de minterminos:

$$m_i(s_{n-1}, \dots, s_0)$$



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

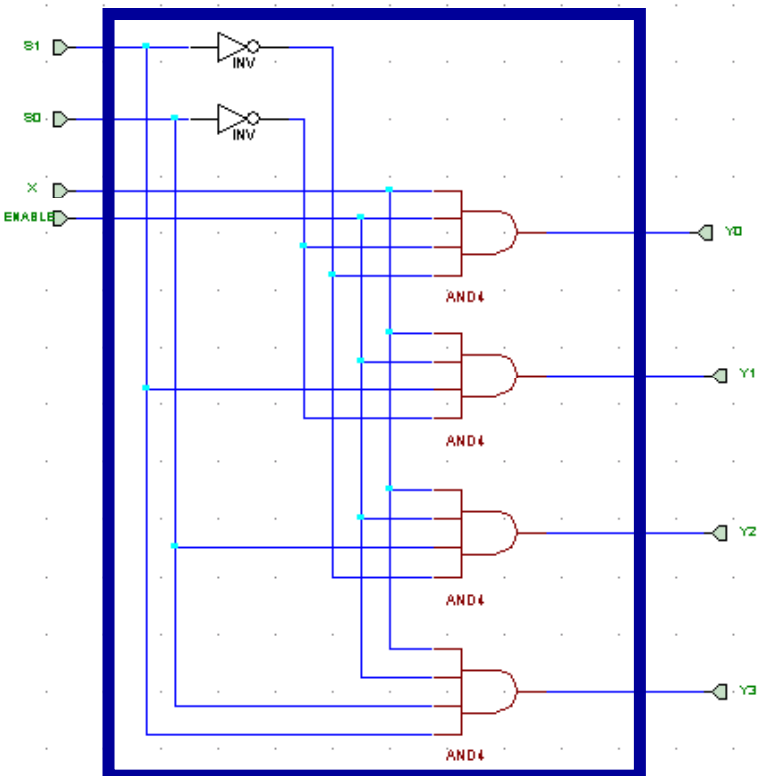
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

MULTIPLEXORES



o: implementar un demultiplexor de 1 a 4.

Demux 4a1



$$x \cdot m_i(s_{n-1}, \dots, s_0)$$

$$\begin{array}{cccc}
 | s_0 & | s_0 & | s_0 & | s_0 \\
 | s_1 & | s_1 & | s_1 & | s_1 \\
 \cdot X & \cdot X & \cdot X & \cdot X
 \end{array}$$

- - -

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

MULTIPLEXORES

o (continuación): descripción concurrente en VHDL.

```

entity demux1a4 is
    port (enable: in std_logic;
          x: in std_logic;
          sel: in std_logic_vector(1 downto 0);
          y: out std_logic_vector(3 downto 0));
end demux1a4;

architecture concurrente of demux1a4 is
    signal y0, y1, y2, y3: std_logic;
begin
    y0 <= enable and x and (not sel(0)) and (not sel(1));
    y1 <= enable and x and sel(0) and (not sel(1));
    y2 <= enable and x and (not sel(0)) and sel(1);
    y3 <= enable and x and sel(0) and sel(1);
end concurrente;

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



MULTIPLEXORES

o (continuación): una posible descripción funcional del mismo multiplexor.

```

architecture funcional_1 of demux1a4 is
  process(x, sel)
  begin
    y <= "0000";
    if enable = '1' then
      case sel is
        when "00" => y(0) <= x;
        when "01" => y(1) <= x;
        when "10" => y(2) <= x;
        when others => y(3) <= x;
      end case;
    end if;
  end process;
funcional_1;

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



MULTIPLEXORES

o (continuación): otra posible descripción funcional del mismo multiplexor.

```

architecture funcional_2 of demux1a4 is
    process(x, sel)
    in
        sel <= "0000";
    if enable = '1' then
        y(conv_integer(sel)) <= x;
    end if;
    process;
end architecture funcional_2;
  
```

La función *conv_integer(parametro)* convierte un vector de bits a su valor decimal. Esta función se encuentra en la librería *ieee.std_logic_unsigned*



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

MULTIPLEXORES

o (continuación): test-bench para probar las 3 arquitecturas
 áneamente.

```

library ieee;
use ieee.std_logic_1164.all;

entity test_demux1a4 is
  port
    enable: std_logic;
    x: std_logic := '0';
    sel: std_logic_vector(1 downto 0);
    y1, y2, y3: std_logic_vector(3 downto 0);
end entity test_demux1a4;

architecture concurrente of test_demux1a4 is
  component work_demux1a4_concurrente
    port
      enable: std_logic;
      x: std_logic;
      sel: std_logic_vector(1 downto 0);
      y1, y2, y3: std_logic_vector(3 downto 0);
    end port;
  end component work_demux1a4_concurrente;

  component work_demux1a4_funcional_1
    port
      enable: std_logic;
      x: std_logic;
      sel: std_logic_vector(1 downto 0);
      y1, y2, y3: std_logic_vector(3 downto 0);
    end port;
  end component work_demux1a4_funcional_1;

  component work_demux1a4_funcional_2
    port
      enable: std_logic;
      x: std_logic;
      sel: std_logic_vector(1 downto 0);
      y1, y2, y3: std_logic_vector(3 downto 0);
    end port;
  end component work_demux1a4_funcional_2;

  concurrente: work_demux1a4_concurrente
    port map
      (enable => enable, x => x, sel => sel, y1 => y1, y2 => y2, y3 => y3);

  funcional_1: work_demux1a4_funcional_1
    port map
      (enable => enable, x => x, sel => sel, y1 => y1, y2 => y2, y3 => y3);

  funcional_2: work_demux1a4_funcional_2
    port map
      (enable => enable, x => x, sel => sel, y1 => y1, y2 => y2, y3 => y3);

begin
  enable <= '1', '0' after 200 ns;
  x <= '0', '1' after 5 ns;
  sel <= "00", "01" after 50 ns, "10" after 100 ns, "11" after 150 ns;
end architecture test_demux1a4;

```

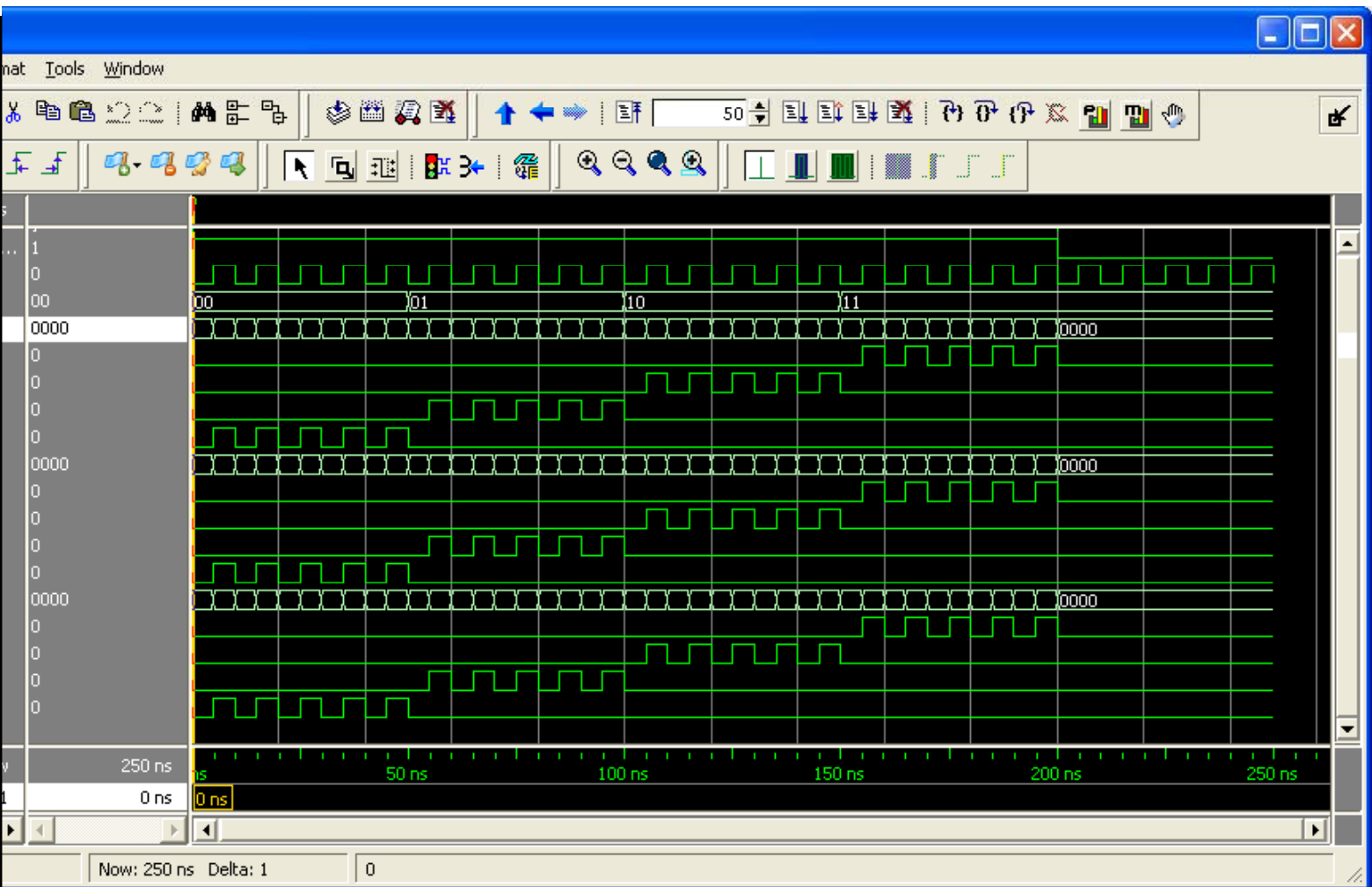


CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

MULTIPLEXORES

o (continuación): resultado de la simulación.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70





CE

rafía

ucción

adores y decodificadores

tesis de funciones de conmutación con decodificadores

lexores y demultiplexores

tesis de circuitos combinacionales con multiplexores

zadores

adores

itivos lógicos programables

os aritméticos básicos

mador

stador

mador/Restador

d aritmético-lógica

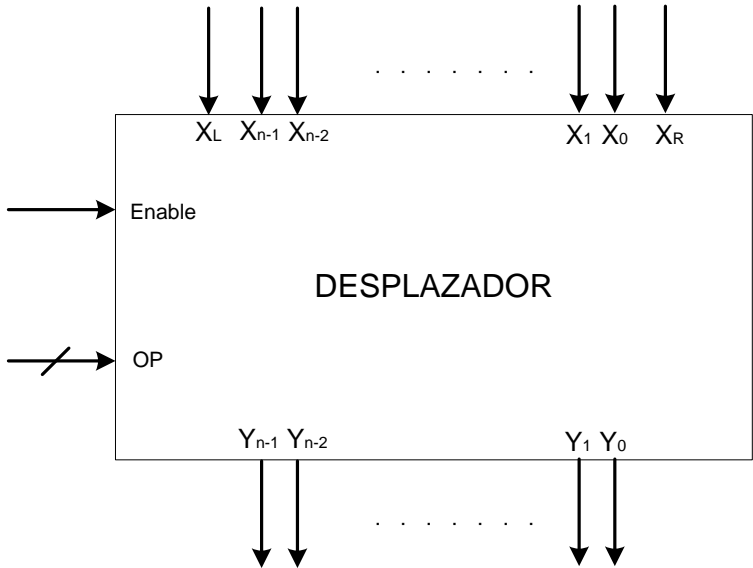
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



DESPLAZADORES

Un desplazador (*shifter*) es un módulo combinacional con $n+2$ entradas y n salidas, además de una señal de activación y varias señales de control. Este desplazador mueve a derecha o izquierda en desplazamientos abiertos o cerrados (rotaciones) en función de las señales de control.

Estos se pueden materializar mediante expresiones de implementación a través de puertas lógicas. La construcción habitual puede consistir en un conjunto de multiplexores.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

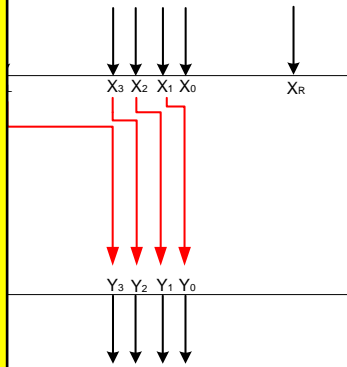
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

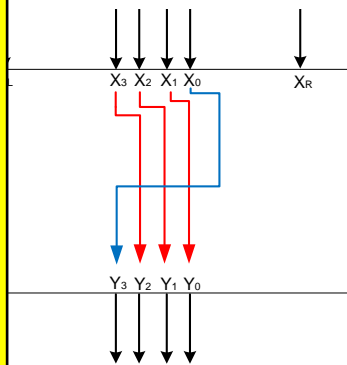
DEPLAZADORES

o: desplazador de 4 bits que realiza las siguientes operaciones:

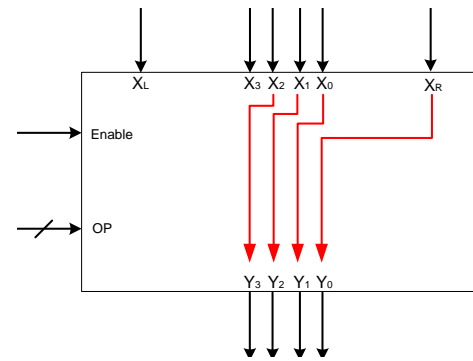
=> Desplazamiento a derecha



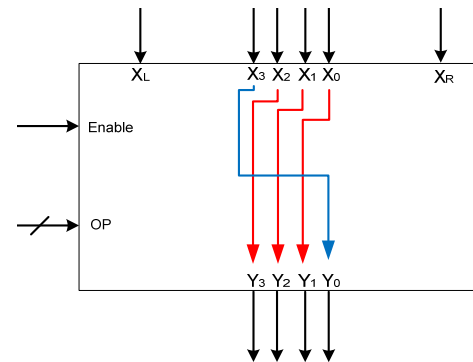
=> Rotación a derecha



OP = "01" => Desplazamiento a izquierda



OP = "11" => Rotación a izquierda

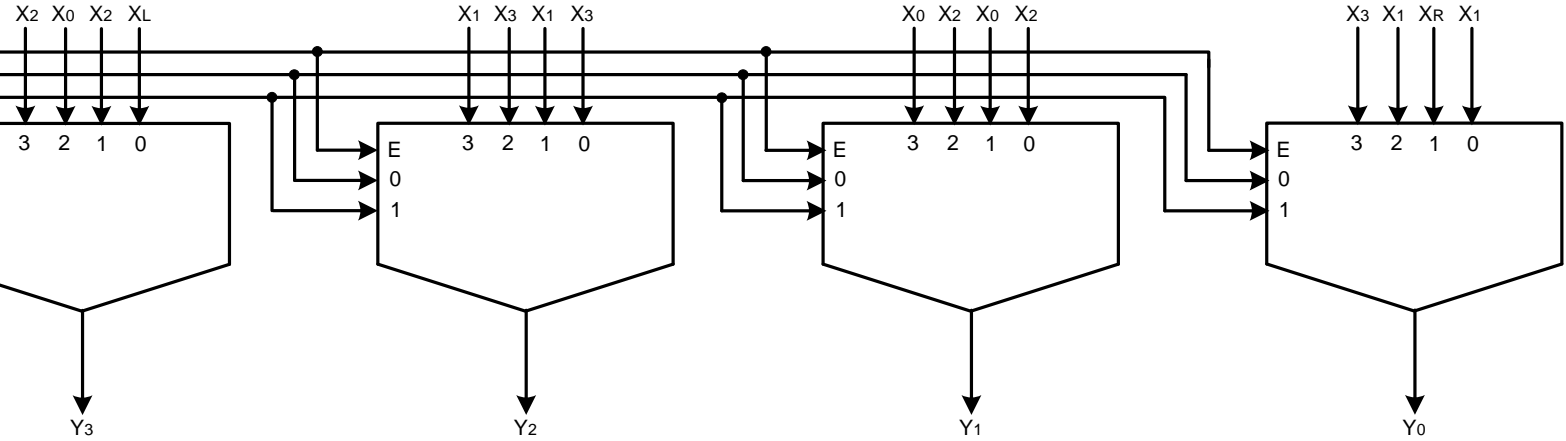


CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

PLAZADORES

o (continuación): se puede implementar utilizando multiplexores:



- - -

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70**

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.





CE

grafía

cción

adores y decodificadores

tesis de funciones de conmutación con decodificadores

lexores y demultiplexores

tesis de circuitos combinacionales con multiplexores

zadores

radores

itivos lógicos programables

os aritméticos básicos

mador

stador

mador/Restador

d aritmético-lógica

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



1 PARADORES

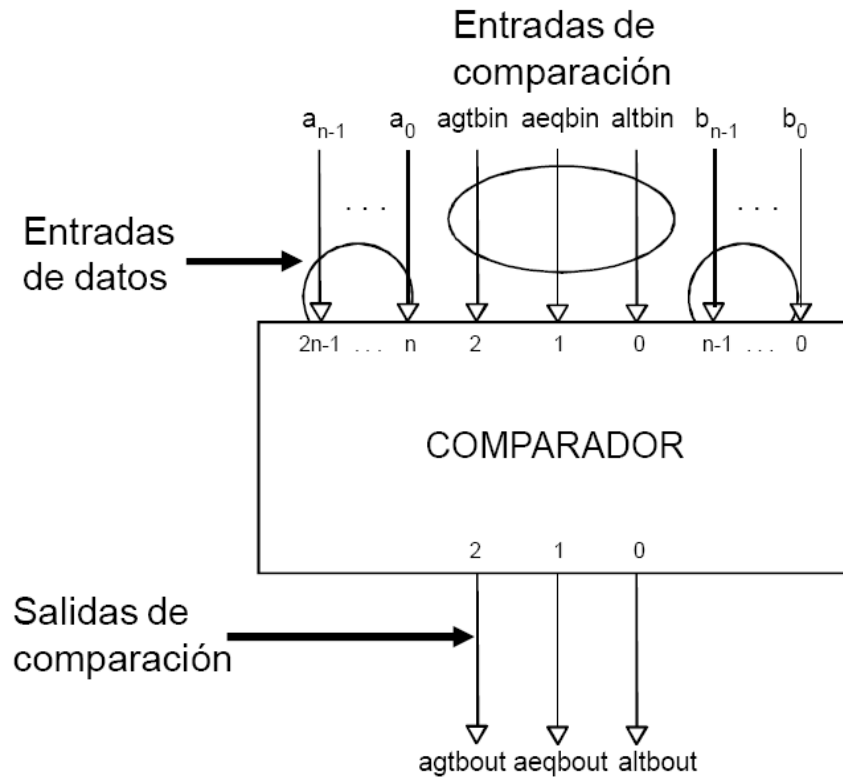
Comparador es un módulo combinacional con $2n$ entradas de datos (a y b, n bits cada una), 3 entradas para encadenamiento de comparación (entradas menos a más significativas) y 3 salidas de comparación.

El comparador determina la relación de magnitud de dos números binarios indicando si el primero es mayor (**agtb**), menor (**altb**) o igual (**aeqb**) al segundo.

$$agtb = (a > b) + (a = b) \cdot agtbin$$

$$aeqb = (a = b) \cdot aeqb_{in}$$

$$altb = (a < b) + (a = b) \cdot altbin$$



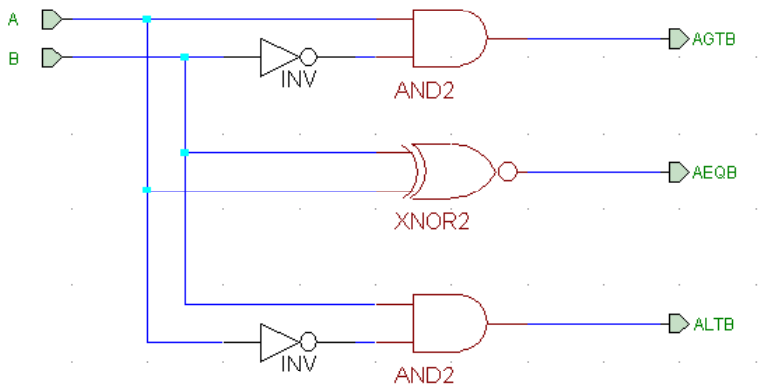
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70



1 PARADORES

o: comparador de un bit encadenable.

os primero de un comparador de un bit NO encadenable:



(a>b)	(a=b)	(a<b)
0	1	0
0	0	1
1	0	0
0	1	0

$$\overline{a} \cdot b$$

$$a \text{ xnor } b$$

$$a \cdot \overline{b}$$

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



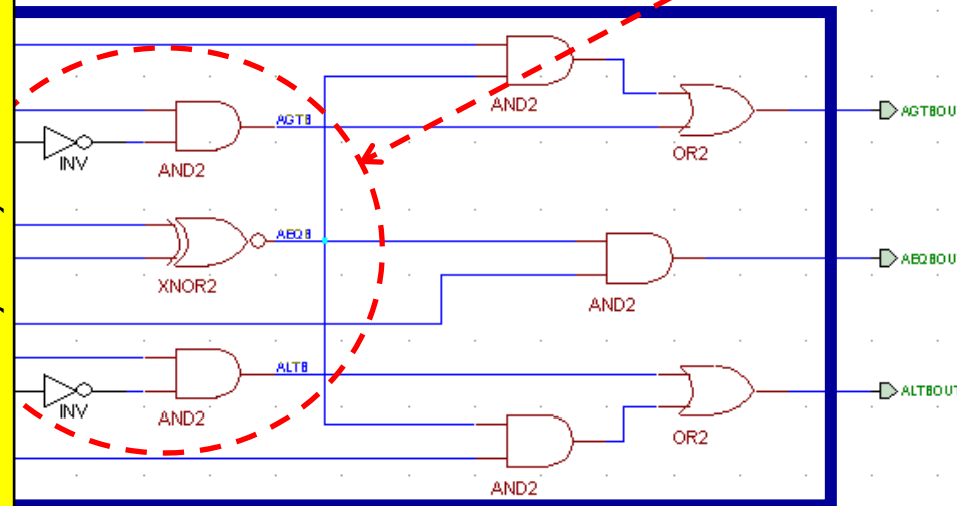
1 PARADORES

o (continuación): a partir del comparador no encadenable nos las entradas de comparación se calculan las ecuaciones de las teniendo en cuenta las entradas de comparación.

$$\text{out} = (a > b) + (a = b) \cdot \text{agtbin}$$

$$\text{out} = (a = b) \cdot \text{aeqbin}$$

$$\text{out} = (a < b) + (a = b) \cdot \text{altbin}$$



Comparador de 1 bit

Comparador de 1 bit encadenable

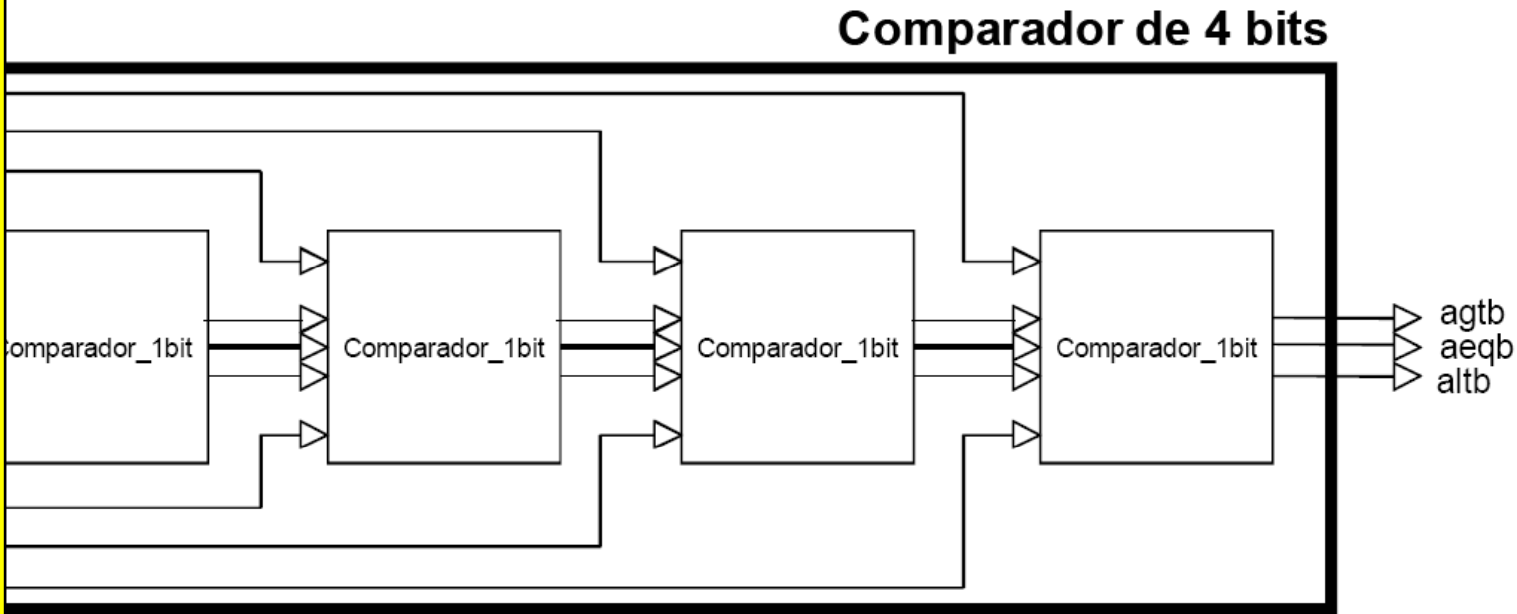
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

1 PARADORES

enando varios comparadores se pueden construir otros de mayor

o.
o: comparador de 4 bits a partir del comparador encadenable de 1



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



CE

rafía

ucción

adores y decodificadores

tesis de funciones de conmutación con decodificadores

lexores y demultiplexores

tesis de circuitos combinacionales con multiplexores

zadores

radores

itivos lógicos programables

os aritméticos básicos

mador

stador

mador/Restador

d aritmético-lógica

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

DISPOSITIVOS LÓGICOS PROGRAMABLES

La denominación de dispositivos lógicos programables se encuadran **junto de circuitos integrados** formados por cierto número de **chips lógicos y/o módulos básicos y/o biestables cuyas conexiones pueden ser personalizadas o programadas, bien sea por el fabricante o por el usuario.**

La ventaja de estos dispositivos reside en que los fabricantes pueden hacer grandes tiradas de estos CI lo que abarata sus costes de producción y los usuarios posteriormente pueden personalizar sus diseños en sus propios laboratorios sin grandes inversiones.

Dentro de ese conjunto de dispositivos haremos mención de las memorias programables por campo (PLA) y las PAL.

Las diferencias fundamentales entre sendos tipos de dispositivos estriba en los grados de libertad que disponen cada uno de ellos en cuanto a su capacidad y flexibilidad de programación.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

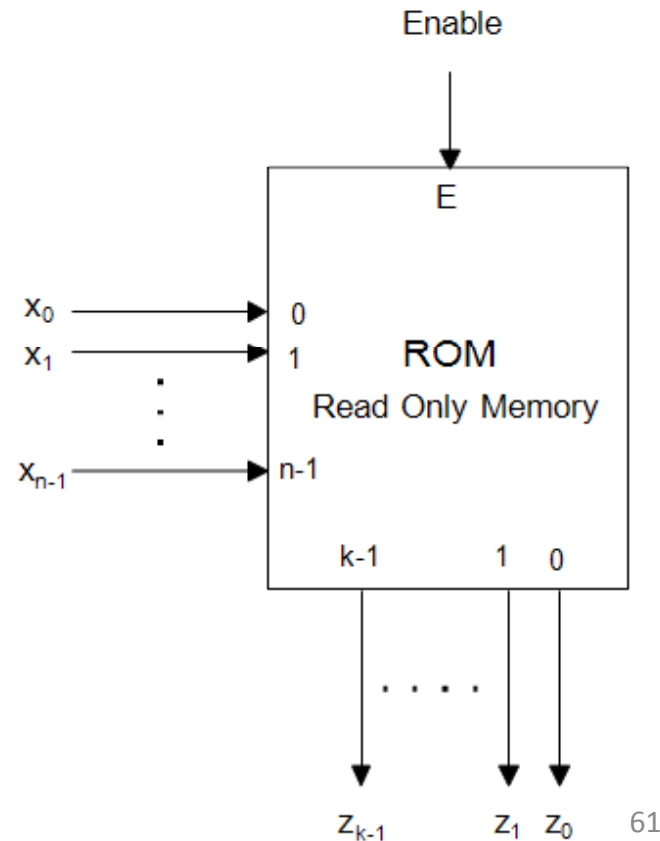
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

DISPOSITIVOS LÓGICOS PROGRAMABLES: ROM

La memoria ROM (Read Only Memory - memoria de sólo lectura) es un dispositivo combinatorial con 'n' entradas de direcciones y 'k' salidas de datos, además de una o varias señales de activación.

Existen distintos tipos, según se puedan o no programar, borrar, como se realiza el borrado, etc:

- ROM Read-Only Memory
- PROM Programmable ROM
- EPROM Erasable and Programmable ROM
- EEPROM Electrically erasable PROM



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

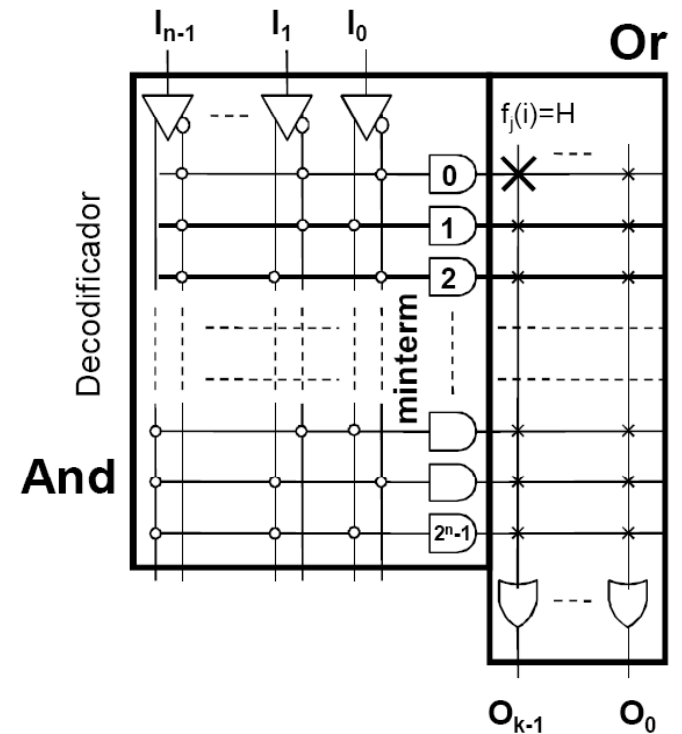
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

POSITIVOS LÓGICOS PROGRAMABLES: ROM

Una ROM se compone internamente de dos grupos de puertas: un grupo de puertas And (de n entradas y 2^n salidas) y un grupo de puertas Or (de 2^n entradas y k salidas).

Los grupos de puertas And están programadas internamente y conectadas de forma fija, mientras que el grupo de puertas Or es programable por el usuario.

Generalmente, una ROM con ' n ' entradas de datos y ' k ' salidas de datos consta de un grupo de puertas And (' n ' inversores y 2^n puertas And de n entradas) y un grupo de puertas Or (' k ' puertas Or de 2^n entradas, como puertas And).



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

POSITIVOS LÓGICOS PROGRAMABLES: ROM

puertas And están conectadas de tal forma que cada una de ellas implementa uno de los 2^n minterms de las 'n' variables de entrada (es decir, un decodificador de n a 2^n).

Cada grupo de puertas Or es programable, de tal forma que cada una de las puertas Or puede implementar una suma de algunos de los 2^n minterms.

Cada salida de datos de la ROM implementa la siguiente expresión de combinación:

$$F_j = \sum_{i=0}^{2^n-1} m_i(x_{n-1}, \dots, x_0) \cdot f_j(i)$$

$f_j(i) = 1$ si existe conexión (fila i, columna j) en el grupo Or

$f_j(i) = 0$ si no existe dicha conexión



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70



POSITIVOS LÓGICOS PROGRAMABLES: ROM

o: implementar las siguientes funciones de conmutación con una

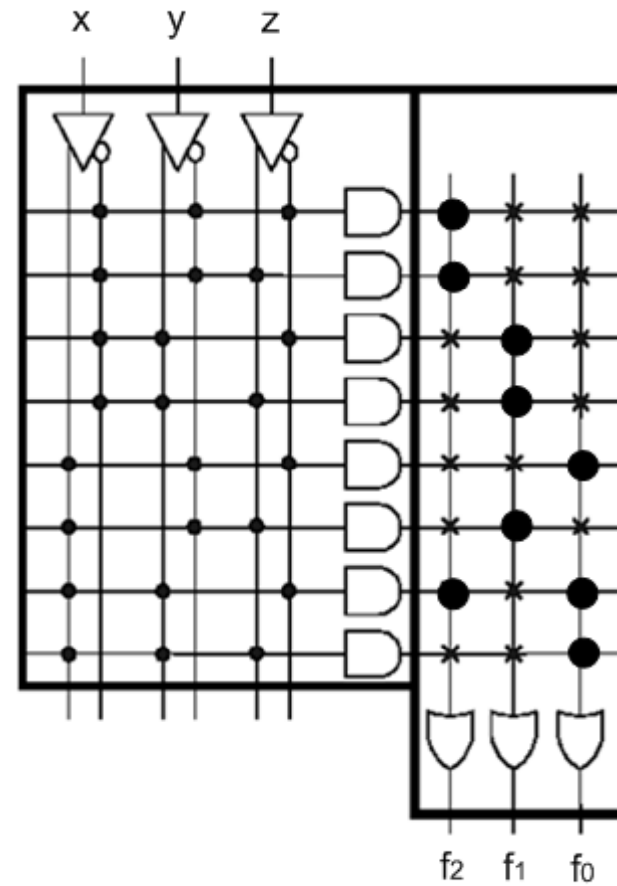
$$f_2(x, y, z) = \sum m(0,1,6)$$

$$f_1(x, y, z) = \sum m(2,3,5)$$

$$f_0(x, y, z) = \sum m(4,6,7)$$

son funciones de 3 variables, por lo que se necesita una ROM con 3 entradas de direcciones.

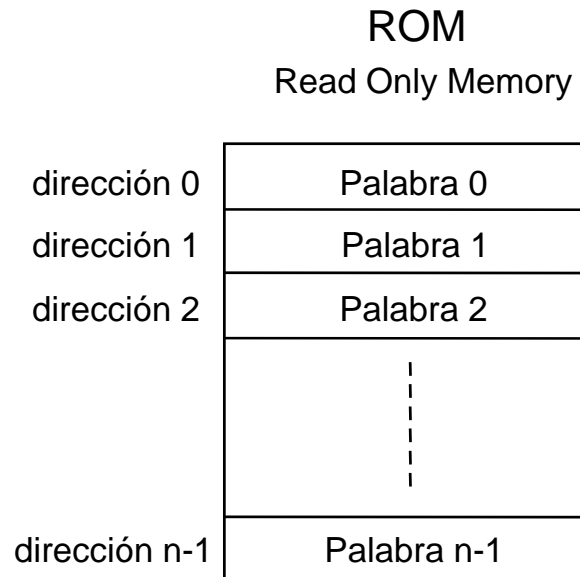
Para implementar las 3 funciones se necesita una ROM con 3 salidas de datos.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

DISPOSITIVOS LÓGICOS PROGRAMABLES: ROM

Una ROM se puede ver como una tabla que almacena datos con la siguiente estructura interna abstracta, donde cada dato ocupa una fila de la tabla denominada dirección.



La única parte programable es la Or se suele representar mediante una matriz de conexiones Or con 1 y 0 indicando conexión o no conexión respectivamente, de nuevo materializando directamente la tabla de



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

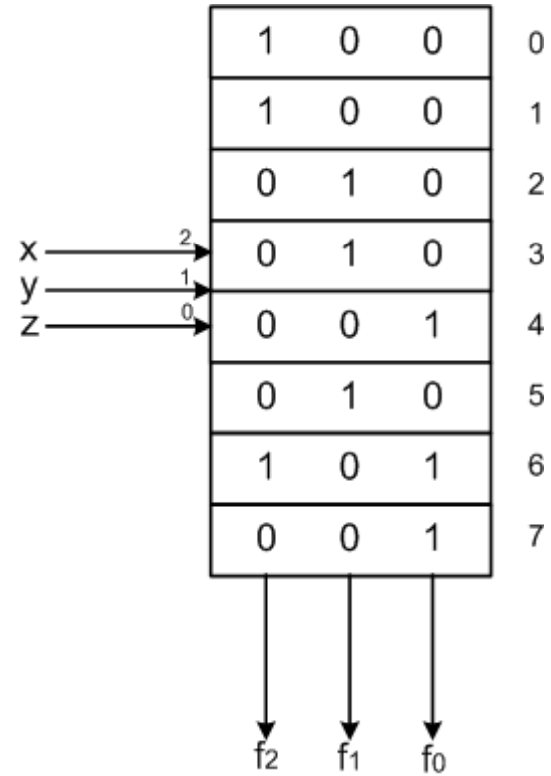
POSITIVOS LÓGICOS PROGRAMABLES: ROM

o: las mismas tres funciones que en el ejemplo anterior:

$$f_2(x, y, z) = \sum m(0,1,6)$$

$$f_1(x, y, z) = \sum m(2,3,5)$$

$$f_0(x, y, z) = \sum m(4,6,7)$$



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

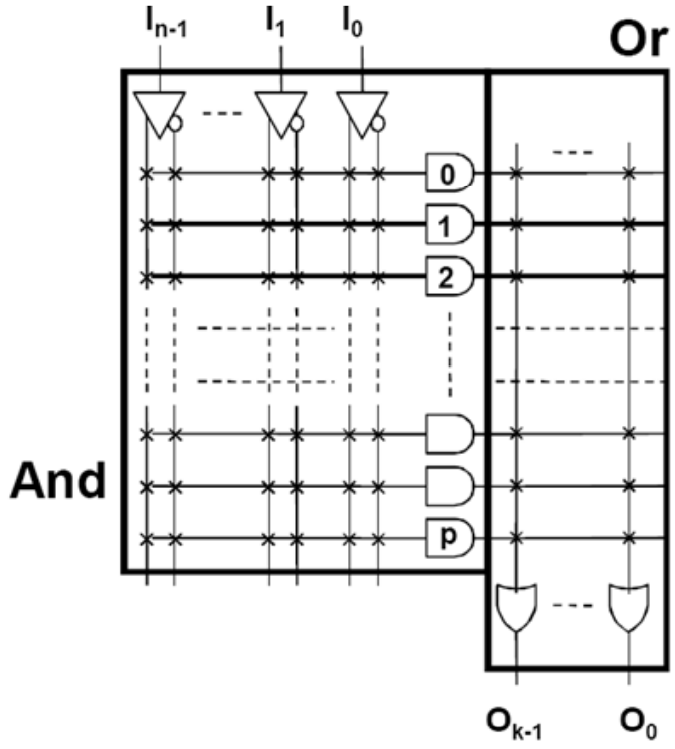
DISPOSITIVOS LÓGICOS PROGRAMABLES: PLA



La memoria ROM materializa FCs simplemente como suma de minterms ya que el grupo de puertas And está prefijado.

En una FC sólo utiliza unos pocos minterms o admite una fuerte simplificación de la que una ROM puede ser un despilfarro.

En este tipo de situaciones se utilizan dispositivos PLAs (Programmable Logic Array) que admiten conexiones programables tanto en el grupo de puertas And como en el grupo de puertas Or.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



POSITIVOS LÓGICOS PROGRAMABLES: PAL

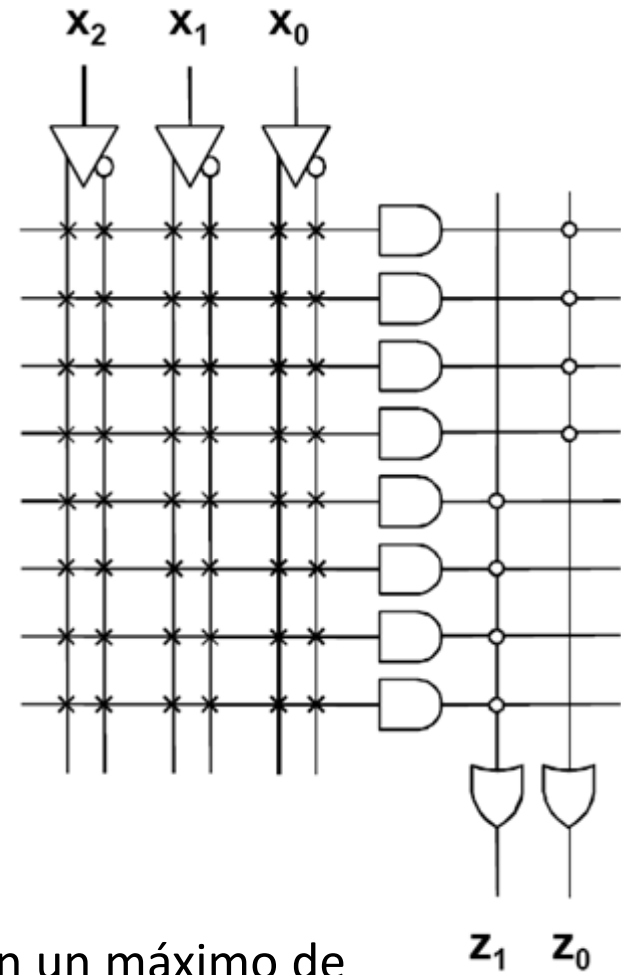
Los PAL (Programmable Array Logic) son un particular de PLA con conexiones programadas.

Un PAL con 'n' entradas y 'k' salidas tiene un grupo de 'm' puertas And de 2^n entradas y un grupo de 'k' puertas Or de p salidas, usualmente $p=m/k$.

Con estos parámetros una PAL puede implementar 'k' funciones de conmutación de variables que se expresen como suma de productos de un máximo de 'p' sumandos.

Ejemplo: representar una PAL con $n=3$, $m=8$
 ¿Qué FCs se pueden materializar?

¿Pueden implementar 2 FCs de 3 entradas con un máximo de 4 términos producto cada una.

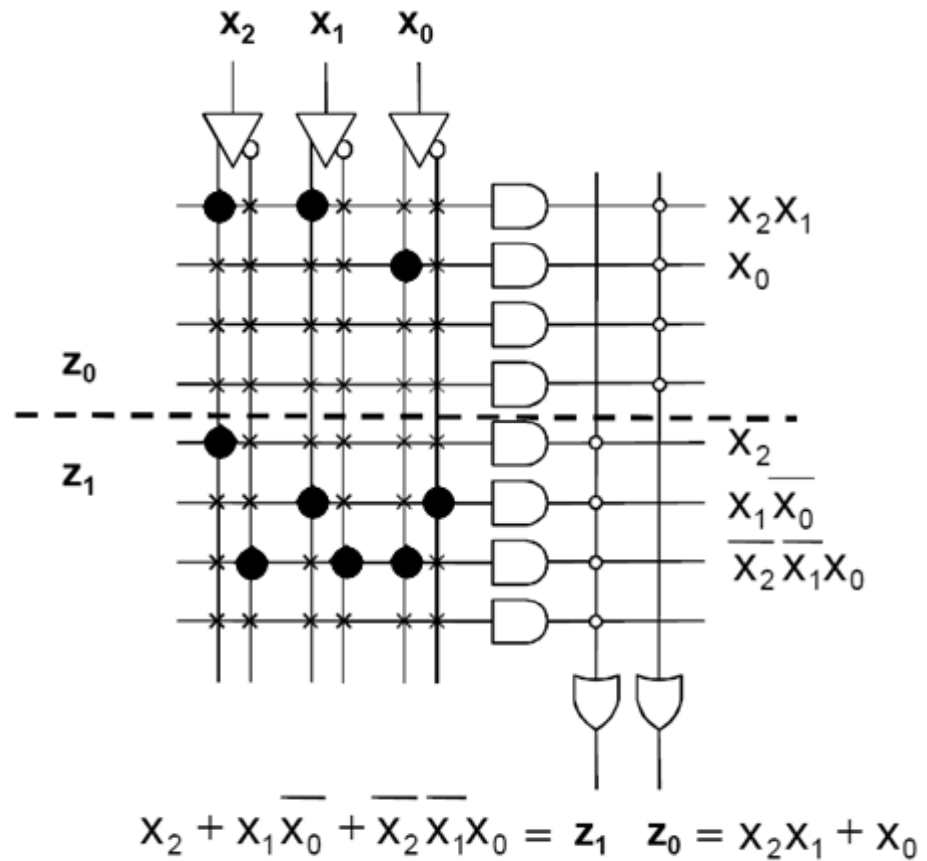


CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

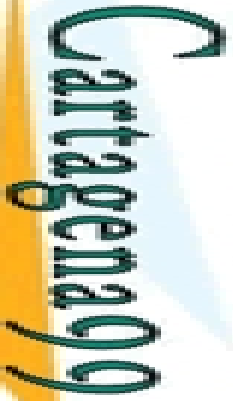
POSITIVOS LÓGICOS PROGRAMABLES: PAL

o: materializar las siguientes funciones de conmutación con una PAL en las características anteriores.



$$z_0 = x_2x_1 + x_0$$

$$z_1 = x_2 + x_1x_0 + x_2x_1x_0$$



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70



CE

rafía

ucción

adores y decodificadores

tesis de funciones de conmutación con decodificadores

lexores y demultiplexores

tesis de circuitos combinacionales con multiplexores

zadores

adores

itivos lógicos programables

os aritméticos básicos

mador

stador

mador/Restador

d aritmético-lógica

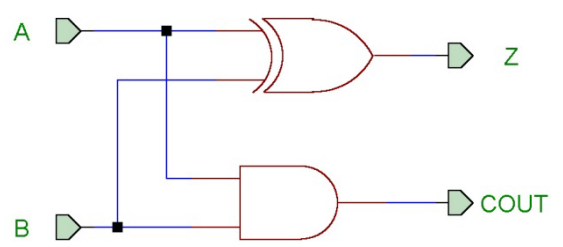
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



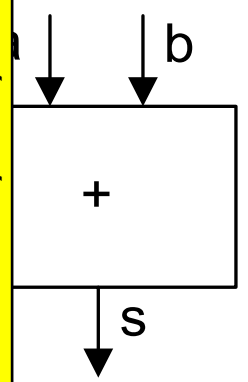
ADADORES

El sumador (*half adder*) es un circuito que suma dos bits de entrada A y B y devuelve un bit de resultado S y un bit de acarreo c_{out} .

S	Cout
0	0
1	0
1	0
0	1



Se representa con este símbolo:



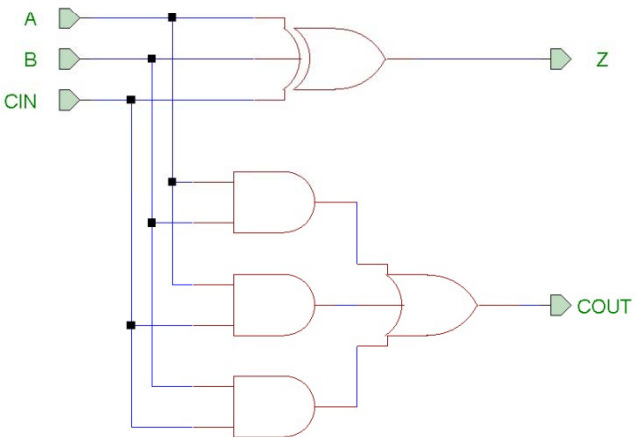
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

ADADORES

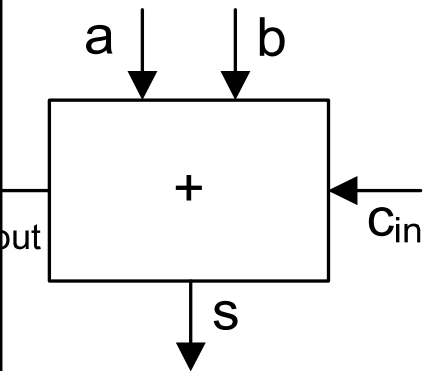
Adador completo (*full adder*) es un circuito que suma dos bits de entrada A y B más un acarreo de entrada C_{in} y devuelve un bit de resultado Z y un bit de acarreo C_{out} .



C_{in}	S	C_{out}
0	0	0
1	0	0
1	0	1
0	1	0
1	1	0
0	0	1
0	1	1
1	1	1



Se puede representar con este símbolo:



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70



ADADORES

o: sumador completo de 1 bit en VHDL

```

entity ieee;
    port (
        a, b, cin: in std_logic;
        s, cout: out std_logic);
end ieee;

architecture concurrente of add1 is
    component add1;
    port (
        a, b, cin: in std_logic;
        s, cout: out std_logic);
end component;

    s <= (a and b) or (a and cin) or (b and cin);
    cout <= a xor b xor cin;
end concurrente;

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

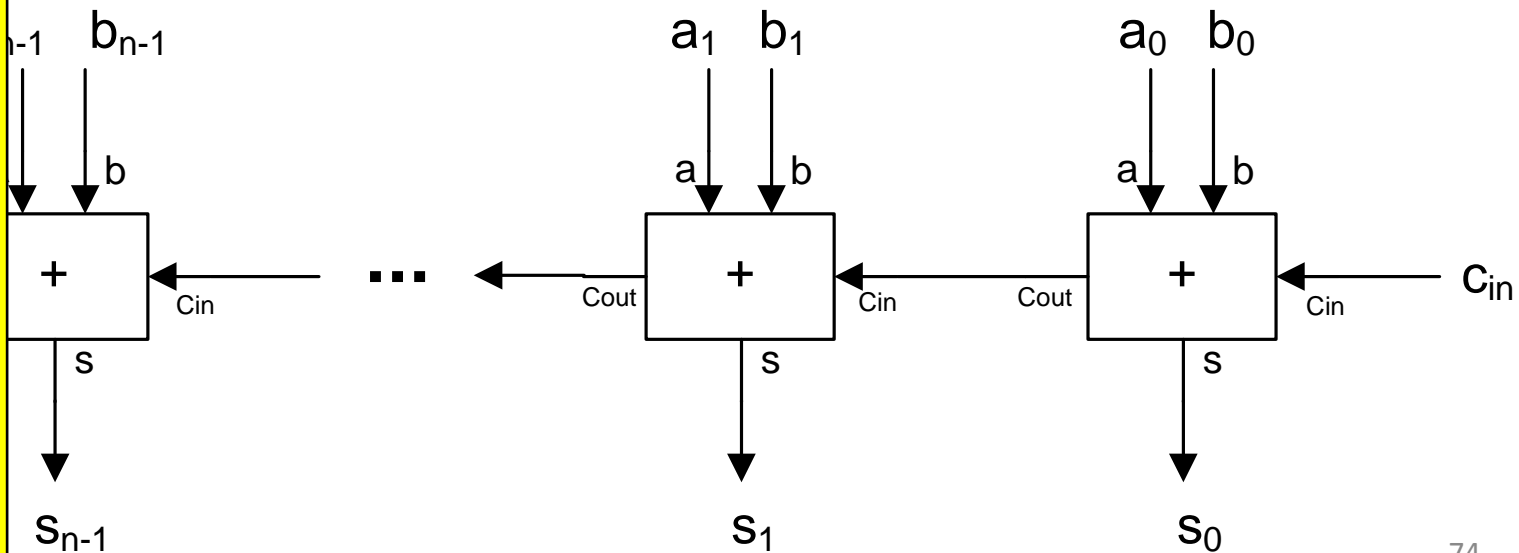
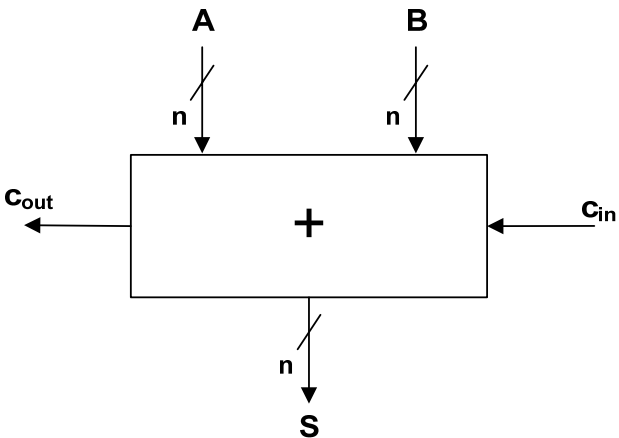
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



ADADORES

or de 'n' bits con propagación de acarreo:

onstruye conectando en cascada varios
nadores completos de 1 bit.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



ADADORES

o: sumador de 4 bits en VHDL.

```

library ieee;
use ieee.std_logic_1164.all;

entity add4 is
    port (a,b: in std_logic_vector(3 downto 0);
          cin: in std_logic;
          s: out std_logic_vector(3 downto 0);
          cout:out std_logic);
end add4;

architecture estructural of add4 is
    signal c: std_logic_vector(4 downto 0);

    c(0) <= cin;
    c(1): entity work.add1 port map(a(0), b(0), c(0), s(0), c(1));
    c(2): entity work.add1 port map(a(1), b(1), c(1), s(1), c(2));
    c(3): entity work.add1 port map(a(2), b(2), c(2), s(2), c(3));
    c(4): entity work.add1 port map(a(3), b(3), c(3), s(3), c(4));

    cout <= c(4);
end estructural;

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



ADADORES

o (continuación): cuando la estructura interna de un circuito es recursiva, como en el sumador de 4 bits, se puede utilizar una instrucción del lenguaje que permite crear varias instancias iguales (**for generate**).

```

structure estructural_2 of add4 is
  type s: std_logic_vector(4 downto 0);
  cin: std_logic;
  sum: std_logic_vector(4 downto 0);
  for i in 0 to 3 generate
    work.add1 port map(a(i), b(i), c(i), s(i), c(i+1));
  end generate;
  sum(4) = c(4);
  structural_2;

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



ADADORES

o (continuación):

```

structure estructural of addn is
  : std_logic_vector(n downto 0);
  = cin;
  for i in 0 to n-1 generate
  sum: entity work.add1 port map(a(i), b(i), c(i), s(i), c(i+1));
  end generate;
  = c(n);
  uctural;

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



ADADORES

o: es a la hora de utilizar el sumador de 'n' bits en otro circuito, o test-bench, cuando se le dará valor al parámetro 'n'.

```

e;
d_logic_1164.all;
d_logic_arith.all;

addn is
n;

--
e test of test_addn is
s: std_logic_vector(7 downto 0);
cout: std_logic;

, '1' after 100 ns;
std_logic_vector(123,8), conv_std_logic_vector(63,8) after 50 ns;
std_logic_vector(31,8);
ntity work.addn generic map(8) port map(a,b,cin,s,cout);

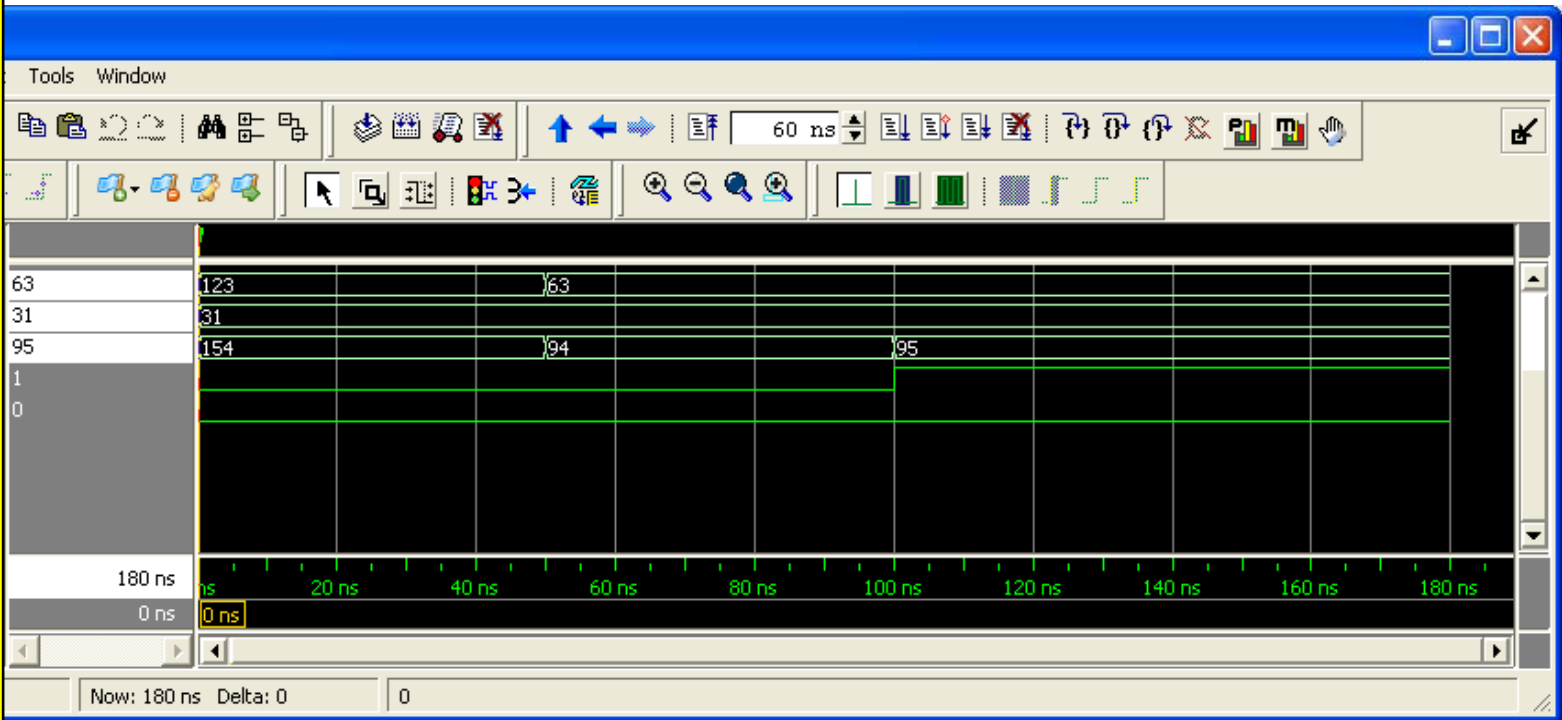
```

ción *conv_std_logic_vector(valor, n)* convierte el número 'valor' a un *std_logic_vector* de 'n' bits. Esta función se encuentra en la librería *std_logic_arith*

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

ADORES

o (continuación): resultado de la simulación.



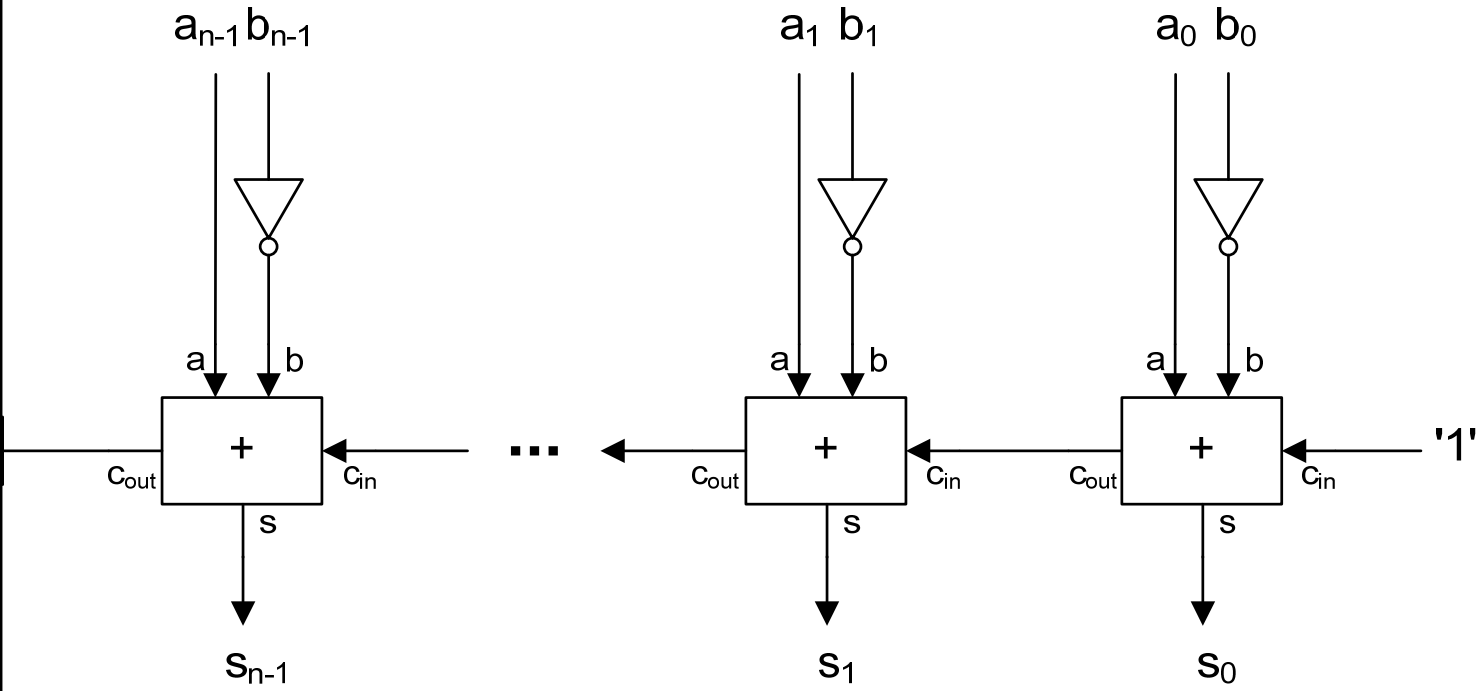
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70



ADDORES

laborio: para restar dos números en binario se hace la suma del
ndo con el complemento a 2 del sustraendo. El complemento a 2 es
plemento a 1 más '1'. En la resta binaria hay que invertir el acarreo



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

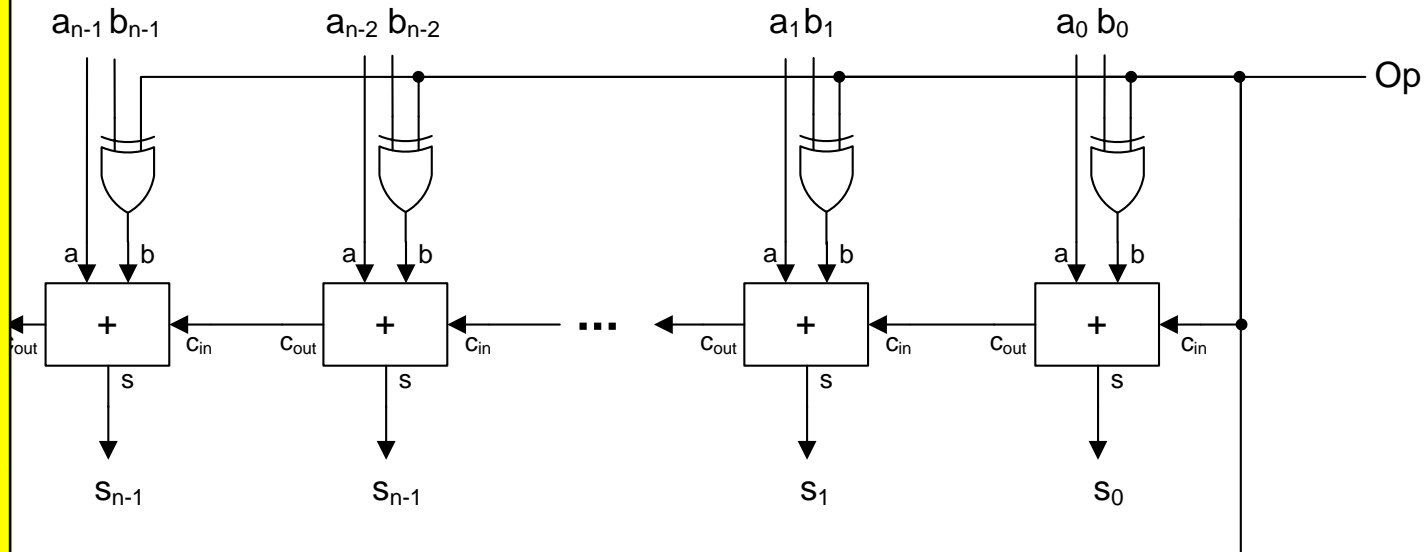
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

ADADORES RESTADORES

den combinar el sumador y el restador en un único circuito que la operación de suma o de resta en función de una señal de

1. Recordar que: $X \oplus 0 = X$

$$X \oplus 1 = \overline{X}$$



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



CE

rafía

ucción

adores y decodificadores

tesis de funciones de conmutación con decodificadores

lexores y demultiplexores

tesis de circuitos combinacionales con multiplexores

zadores

radores

itivos lógicos programables

os aritméticos básicos

mador

stador

mador/Restador

l aritmético-lógica

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

IDAD ARITMÉTICO LÓGICA

unidad aritmético lógica (ALU) es un circuito combinacional que se encarga de hacer distintas operaciones aritméticas (sumas, restas, desplazamientos) y lógicas (AND, OR, NOT, XOR, etc) entre dos operandos, bajo el control de unas señales de control.

Estas operaciones están implementadas con sumadores-restadores, multiplexores y otros circuitos adicionales.

Esquema de diseño: ALU con dos entradas de datos de 4 bits, que realiza las siguientes operaciones en función de una entrada de control de 2 bits:

OP ₁	OP ₀	OPERACIÓN
0	0	X + Y
0	1	X - Y
1	0	X AND Y
1	1	X OR Y



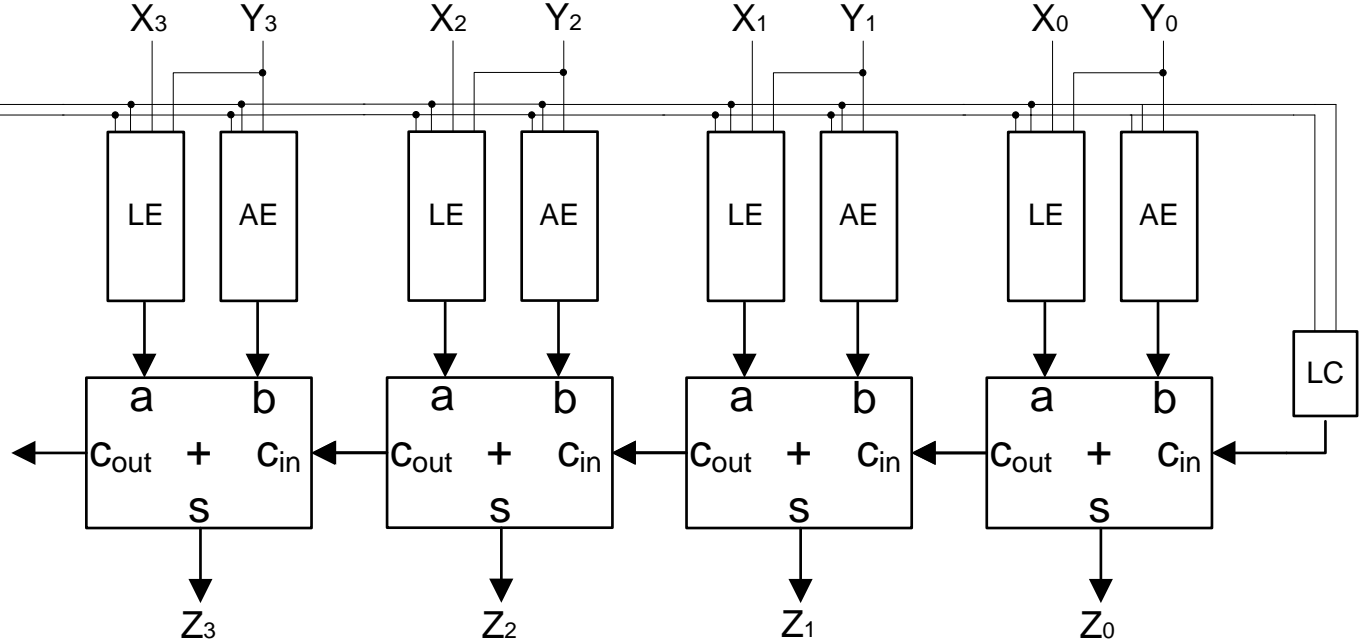
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



ADAD ARITMÉTICO LÓGICA

o (continuación):



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

ADAD ARITMÉTICO LÓGICA

En la estructura propuesta, los bloques LE y AE se encargan de proporcionar a las entradas A y B de los sumadores los datos necesarios para que se realice la operación deseada. El bloque LC se encarga de proporcionar el acarreo de entrada a la cadena de sumadores.

OP1 y OP0 = "00"

$A_i = X_i$, $B_i = Y_i$, $LC = 0$, y por tanto los sumadores harán la operación $X + Y + 0$.

OP1 y OP0 = "01"

$A_i = X_i$, $B_i = \neg Y_i$, $LC = 1$, y por tanto los sumadores harán la operación $X + \neg Y + 1$, es decir $X - Y$.

OP1 y OP0 = "10"

$A_i = X_i \text{ AND } Y_i$, $B_i = 0$, $LC = 0$, y por tanto los sumadores harán la operación $(X \text{ AND } Y) + 0 + 0$, es decir $X \text{ AND } Y$.

OP1 y OP0 = "11"

$A_i = X_i \text{ OR } Y_i$, $B_i = 0$, $LC = 0$, y por tanto los sumadores harán la operación $(X \text{ OR } Y) + 0 + 0$, es decir $X \text{ OR } Y$.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

ADAD ARITMÉTICO LÓGICA



LE

	01	11	10
0	0	1	1
0	0	1	1
1	1	1	1
0	0	1	0



$$\cdot y_i + x_i \cdot y_i + \overline{OP_1} \cdot x_i + OP_0 \cdot x_i$$

=> vale '1' cuando OP = "01" => $LC = \overline{OP_1} \cdot OP_0$

AE

	OP ₁ OP ₀	00	01	11	10
OP ₁ OP ₀ \ y _i	00	0	1	0	0
	01	1	0	0	0



$$b_i = \overline{OP_1} \cdot \overline{OP_0} \cdot y_i + \overline{OP_1} \cdot OP_0 \cdot y_i$$

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70