CÓDIGOS BINARIOS

FUNDAMENTOS DE DISEÑO DIGITAL
OPTATIVA I. ISISA

CODIGOS BINARIOS

Cuando se representan números, letras o palabras por medio de un grupo especial de símbolos, se dice que se encuentran codificados, y al grupo de símbolos se le llama código.

CODIGO: representación unívoca de una cantidad

Los códigos binarios son usados por el hardware para el procesamiento de información.

CODIGOS BINARIOS

CATEGORÍAS DE CÓDIGOS USADOS EN HARDWARE

CATEGORÍA I. Códigos utilizados por los circuitos lógicos para realizar operaciones Booleanas. Ejemplo código binario.

CATEGORÍA II. Códigos utilizados para representar los números decimales (0 al 9) o para codificar cantidades binarias. Ejemplo código BCD, Gray, EXC3, etc. A este tipo de códigos de forma genérica se les llama:Códigos Binarios.

CATEGORIA III. Códigos utilizados para convertir los números decimales, las letras del alfabeto, símbolos y operadores en forma binaria o hexadecimal. Ejemplo código ASCII.

CATEGORIA IV. Códigos de instrucciones utilizados en los procesadores y PLDs que hacen que estos realicen una serie de instrucciones. Ejemplo lenguaje C, HDL, etc.

CLASIFICACION DE CÓDIGOS BINARIOS

Los códigos binarios se pueden dividir en dos tipos:

1. Cíclicos y continuos (GRAY)

- Continuos: sÍ existe adyacencia entre sus combinaciones consecutivas.
- Cíclicos: sÍ existe adyacencia entre la primera y última representación.

2. BCD

BCD ➡ DECIMAL CODIFICADO EN BINARIO
ESTOS CÓDIGOS SIGUEN LAS REGLAS DEL SISTEMA NUMÉRICO DECIMAL

- Ponderados (BCD): cada bit tiene un peso bien definido dependiendo del lugar que ocupa en la cantidad.
- No ponderado (EXC3): el nombre del código indica como se van a representar las cantidades

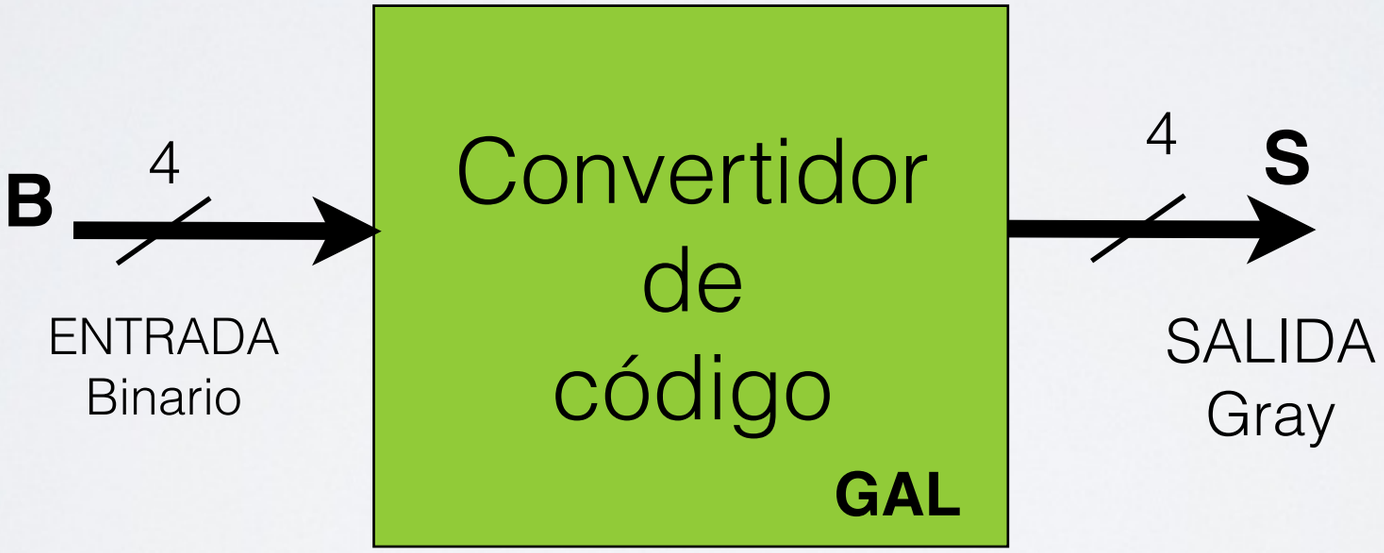
Tabla 1. CODIGOS BINARIOS

SIS. DECIMAL	BINARIO	BCD		Cíclicos y continuos
		Ponderados	No ponderados	
		BCD	EXC3	GRAY
0	0000	0000	0011	0000
1	0001	0001	0100	0001
2	0010	0010	0101	0011
3	0011	0011	0110	0010
4	0100	0100	0111	0110
5	0101	0101	1000	0111
6	0110	0110	1001	0101
7	0111	0111	1010	0100
8	1000	1000	1011	1100
9	1001	1001	1100	1101
10	1010	0001 0000	0100 0011	1111
11	1011	0001 0001	0100 0100	1110
12	1100	0001 0010	0100 0101	1010
13	1101	0001 0011	0100 0110	1011
14	1110	0001 0100	0100 0111	1001
15	1111	0001 0101	0100 1000	1000

CONVERTIDOR DE CÓDIGO

Un convertidor de código es un circuito lógico que convierte los valores binarios de entrada, que representan un código binario, a otro tipo de código binario.

Convertidor de código BIN/GRAY



DISEÑO DEL CONVERTIDOR BINARIO A GRAY DE 4 BITS

Binario				Código Gray			
B3	B2	B1	B0	S3	S2	S1	S0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Tabla 2. Tabla de verdad del convertidor de código Bin/Gray.

Usando las columnas de código binario y Gray (Tabla 1) se obtiene la tabla de verdad del convertidor de código.

De la Tabla se tiene las siguiente funciones canónicas

$S_3=\sum m(8,9,10,11,12,13,14,15)$
$S_2=\sum m(4,5,6,7,8,9,10,11)$
$S_1=\sum m(2,3,4,5,10,11,12,13)$
$S_0=\sum m(1,2,5,6,9,10,13,14)$

DISEÑO DEL CONVERTIDOR BINARIO A GRAY DE 4 BITS

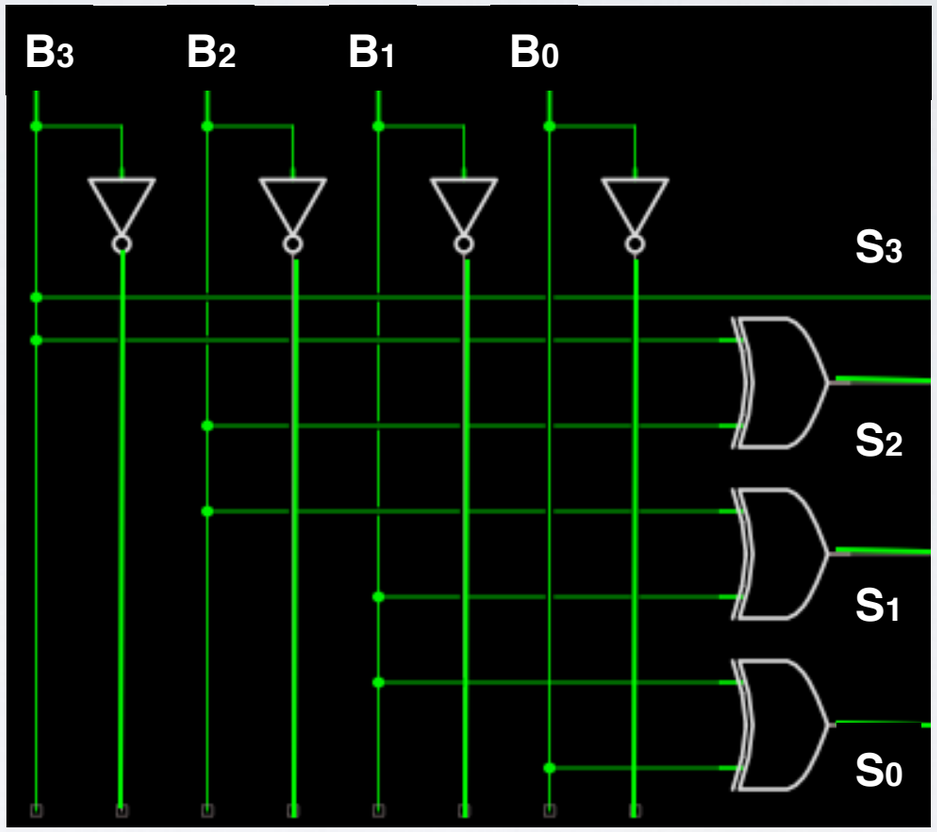
Minimizando las ecuaciones por mapas de Karnaugh se obtienen las siguientes funciones y su circuito lógico :

$S_3=B_3$

$S_2=B_3 \oplus B_2$

$S_1=B_2 \oplus B_1$

$S_0=B_1 \oplus B_0$



Circuito Lógico del convertidor de código binario a Gray de 4 bits.

DISEÑO DEL CONVERTIDOR BINARIO A GRAY

Descripción del circuito en VHDL

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY CON IS

PORT ( B:IN STD_LOGIC_VECTOR (3 DOWNTO 0);
       S:OUT STAD_LOGIC_VECTOR ( 3 DOWNTO 0)
      );
END ENTITY;

ARCHITECTURE A_CON OF CON IS
BEGIN

S(3) <= B(3);
S(2) <= B(3) XOR B(2);
S(1) <= B(2) XOR B(1);
S(0) <= B(1) XOR B(0);

END A_CON;
```

NOTA:

La instrucción

B: IN STD_LOGIC_VECTOR (3 DOWNTO 0);

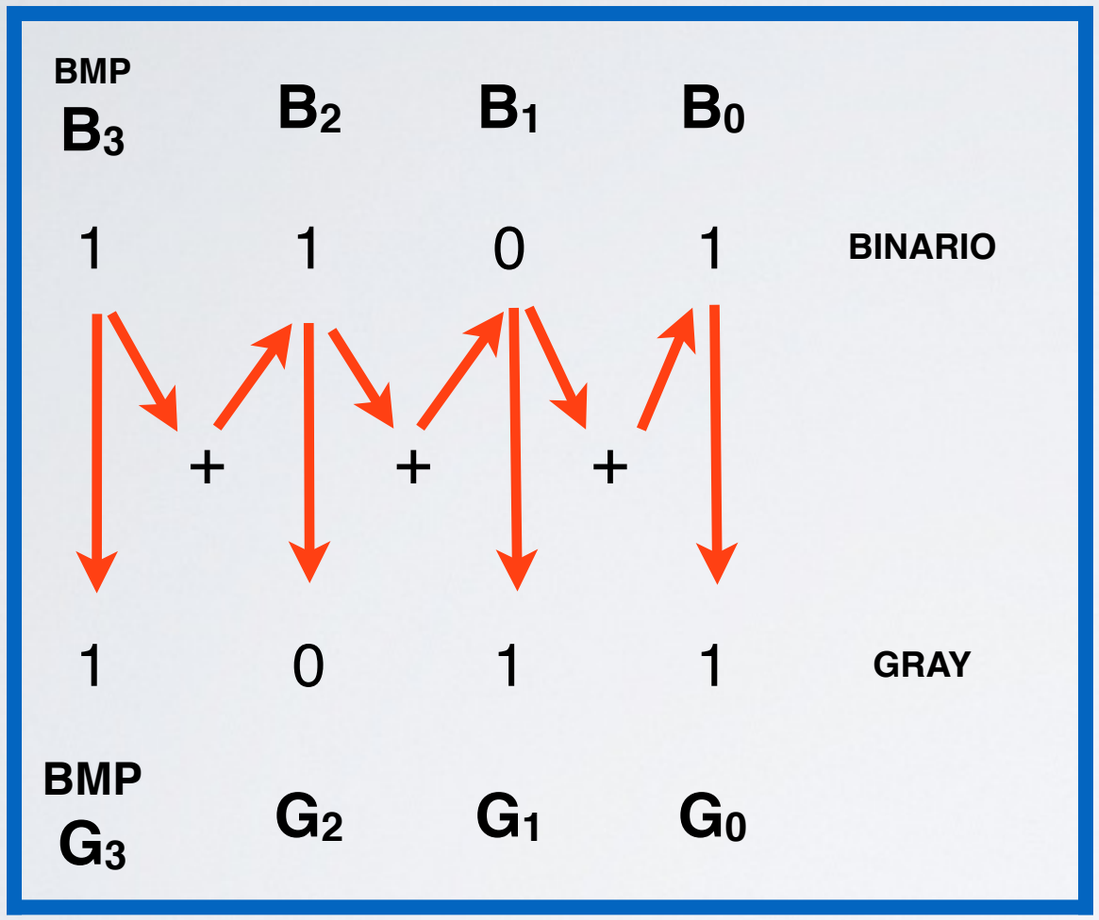
Genera el siguiente arreglo, llamado vector

B [B(3) B(2) B(1) B(0)]

Se pueden aplicar, por ejemplo, operadores lógicos o aritméticos a los vectores ya sean como bloques de datos o a cada elemento del vector por separado, como en el caso de la descripción anterior.

DISEÑO DEL CONVERTIDOR BINARIO A GRAY

Otra forma de diseñar el convertidor es analizar el método de conversión de cantidades binarias a Gray.



Método para convertir cantidades binarias

1. El BMP de la cantidad en Gray será igual al BMP de la cantidad en binario.

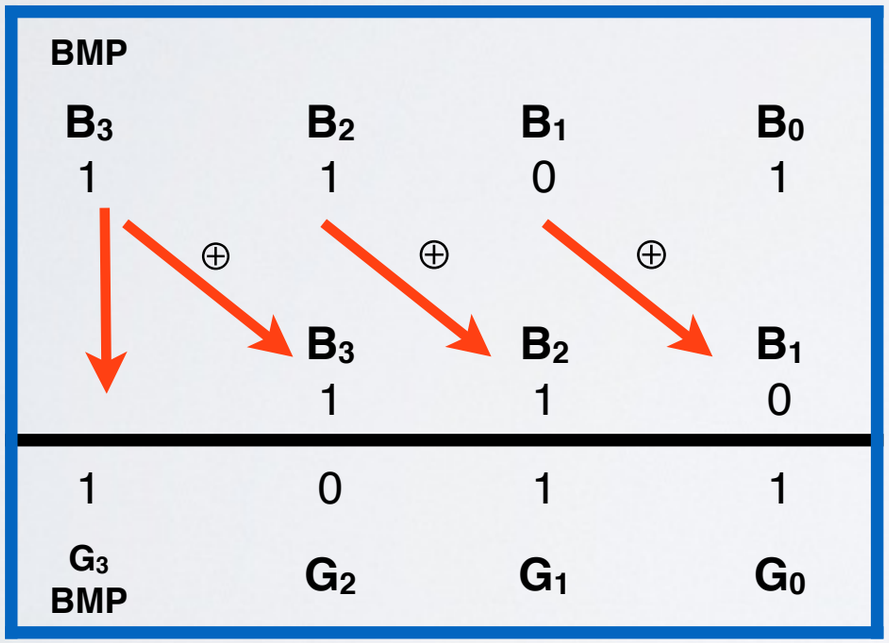
2. El siguiente bit en la cantidad en Gray será igual a la suma aritmética del bit de igual peso de la cantidad en binario y del bit a la izquierda de la misma cantidad; y así sucesivamente para todos los bits de la cantidad en Gray.

3. En caso de que la suma tenga residuo este se despreciara.

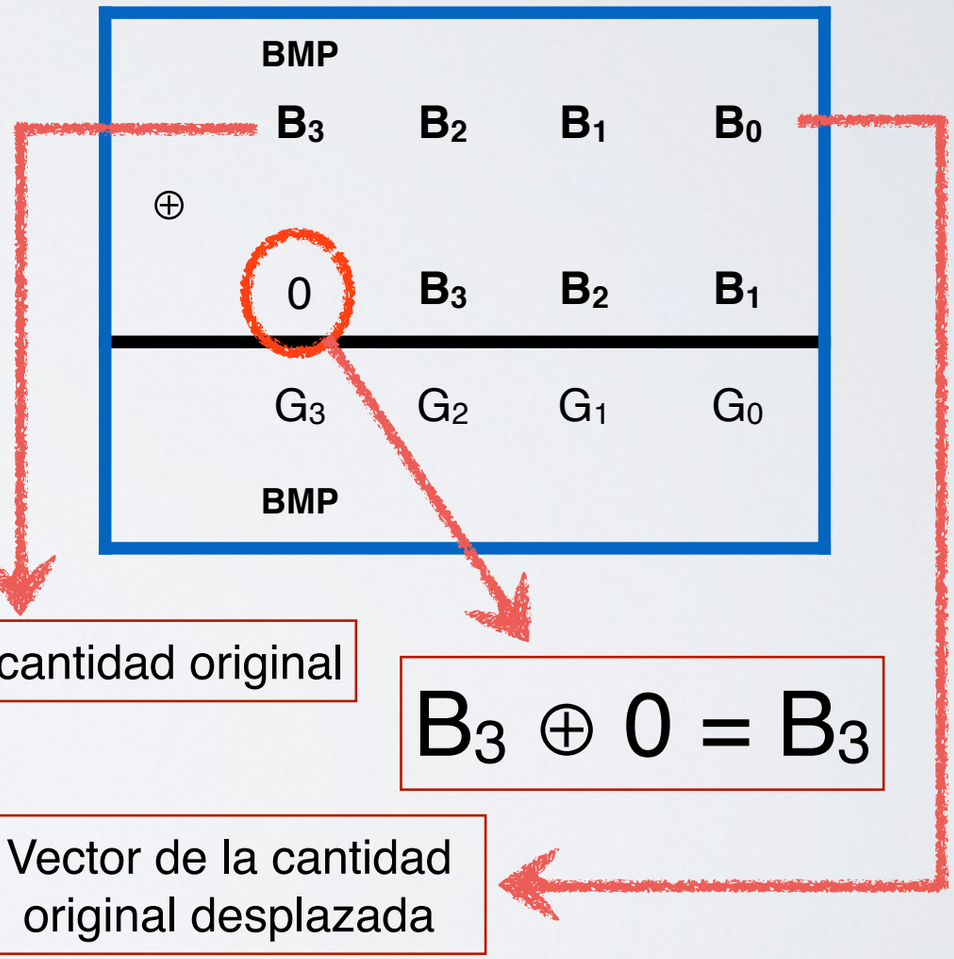
Método para convertir cantidades binarias a Gray

Analizando el método anterior se tiene:

1. La operación lógica que representa mejor la suma aritmética de dos bits sin acarreo es la XOR.
2. Por lo cual se puede decir que el método consiste en aplicar una XOR a la cantidad en binario y a esta misma pero desplazada una posición a la derecha, como se puede ver continuación.



Considerando las propiedades de los vectores en VHDL, se puede resumir:



DISEÑO DEL CONVERTIDOR BINARIO A GRAY

Para la descripción en VHDL del método de conversión de binario a Gray visto anteriormente, usaremos los "***operadores de corrimiento o desplazamiento"***.

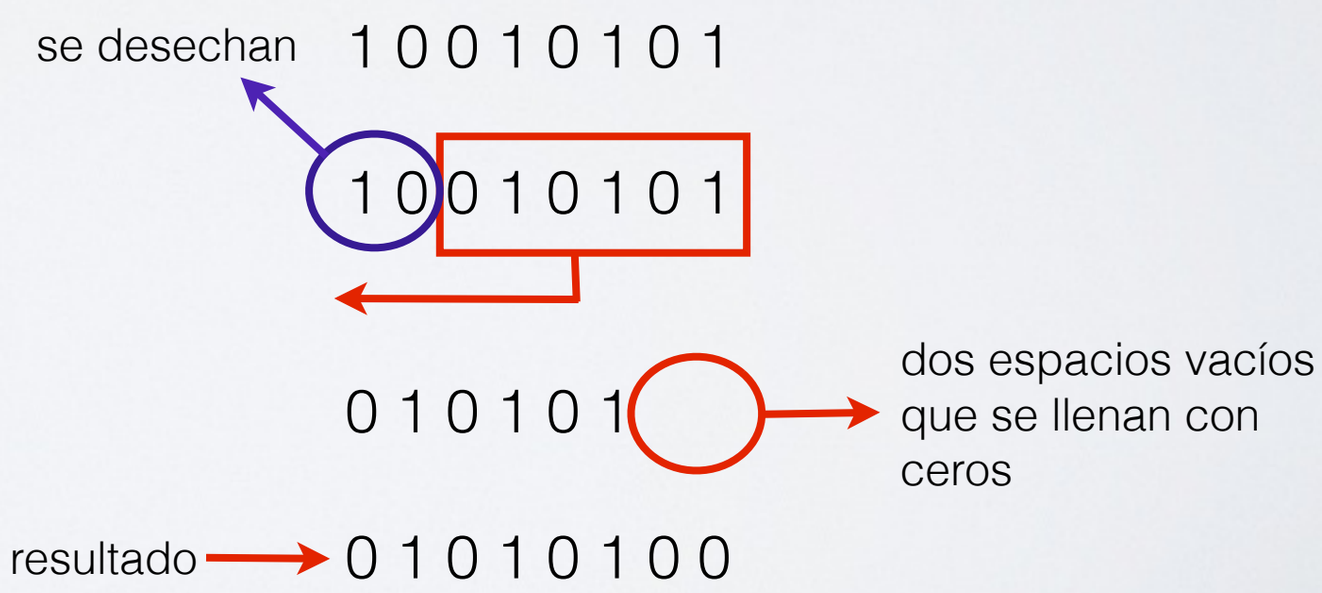
Los operadores de corrimiento son instrucciones que desplazan un vector n posiciones de bits a la derecha, izquierda o en rotación. Los espacios que quedan vacíos después de los desplazamientos se llenan con '1s' o '0s' dependiendo del tipo de desplazamiento.

Operadores de corrimiento

sll → desplazamiento lógico (llenado con ceros) hacia la izquierda.

Si A=10010101

A sll 2 = 01010100

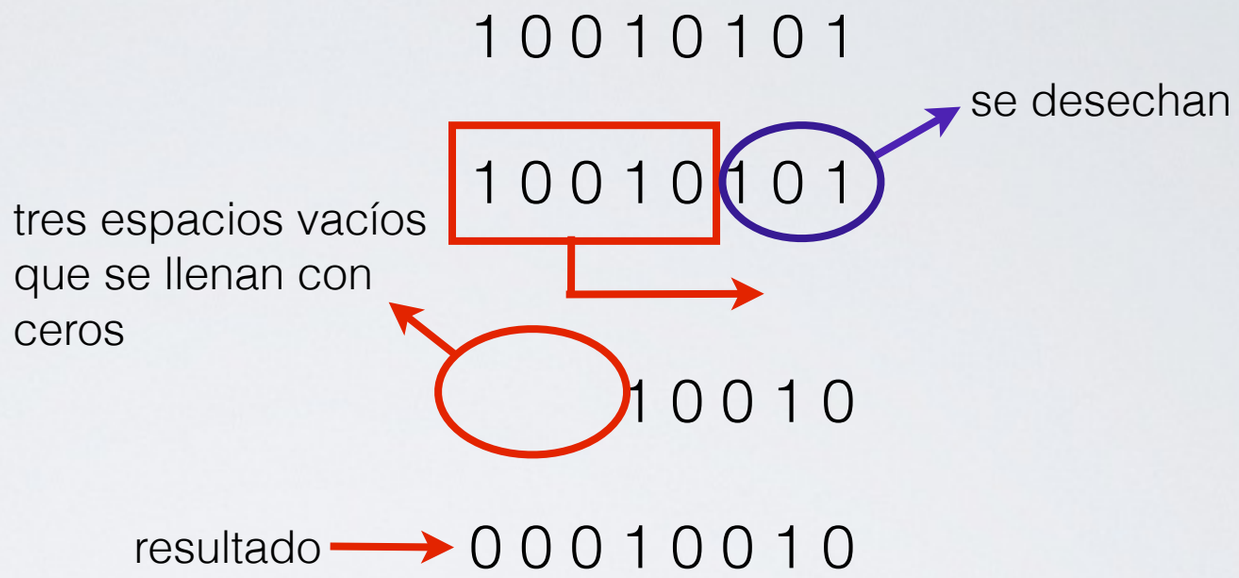


DISEÑO DEL CONVERTIDOR BINARIO A GRAY

srl → desplazamiento lógico (llenado con ceros) hacia la derecha.

Si A=10010101

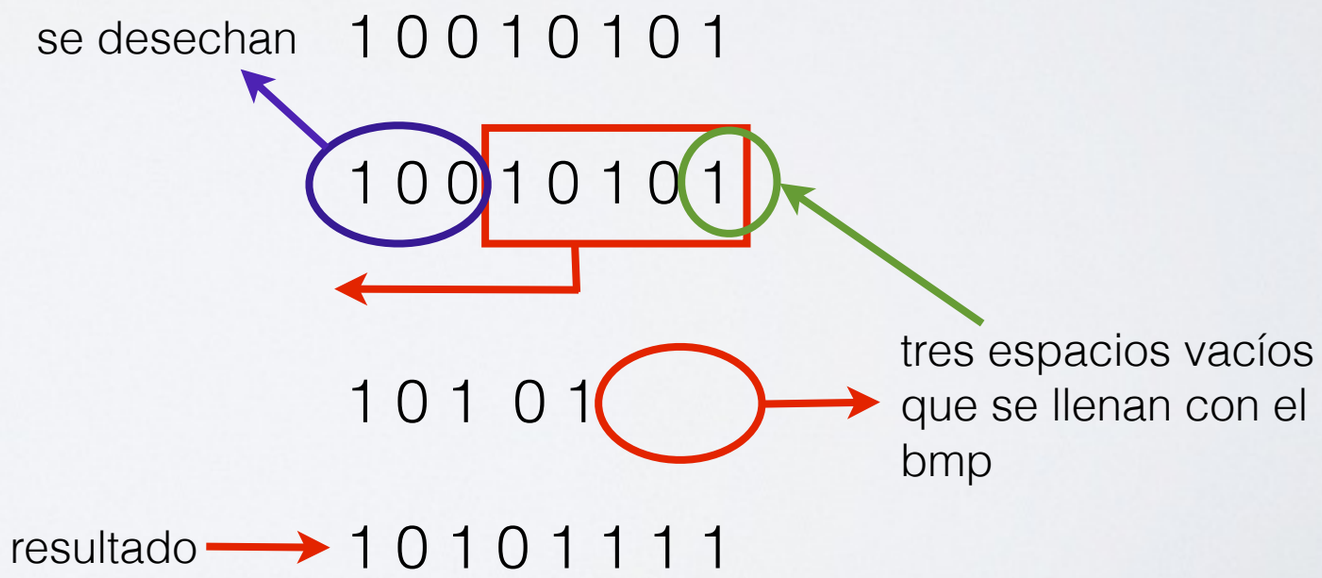
A srl 3 = 00010010



sla → desplazamiento aritmético (llenado con el último bit a la derecha o el bmp) hacia la izquierda.

Si A=10010101

A sla 3 = 10101111

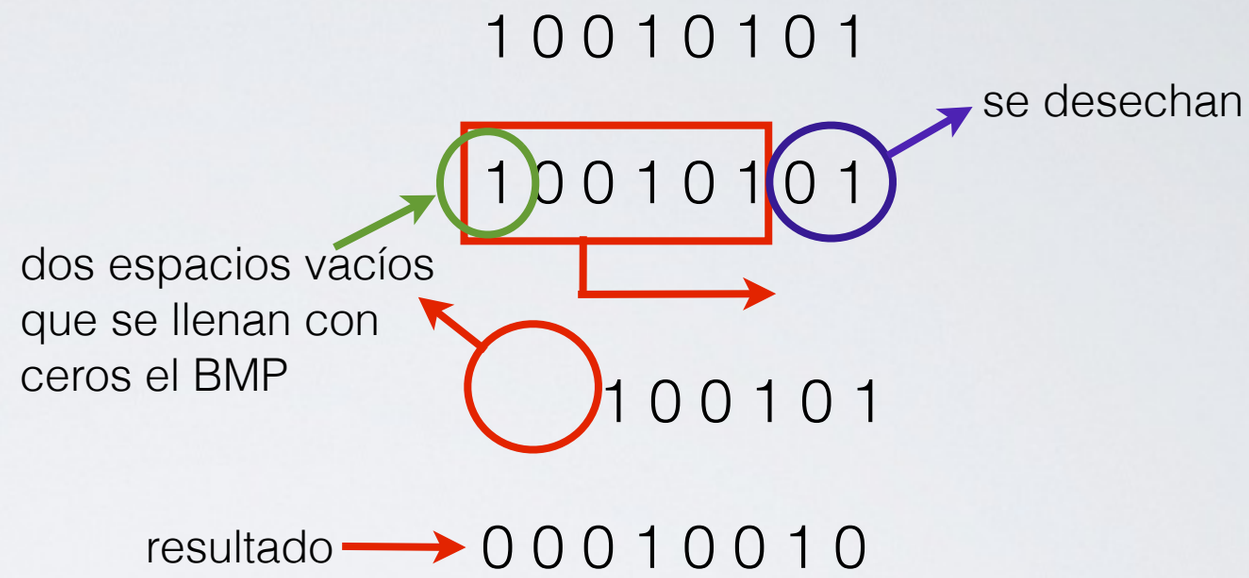


DISEÑO DEL CONVERTIDOR BINARIO A GRAY

sra → desplazamiento aritmético (llenado con el último bit a la izquierda o el BMP) hacia la derecha.

Si A=10010101

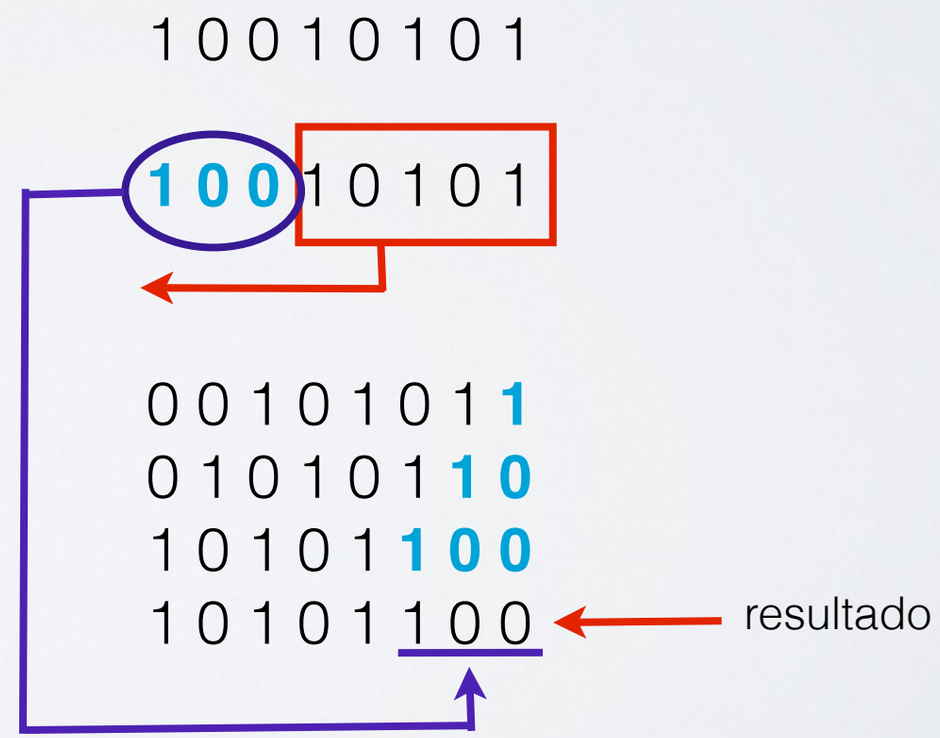
A sra 2 = 11100101



rol → rotación a la izquierda

Si A=10010101

A rol 3 = 10101100

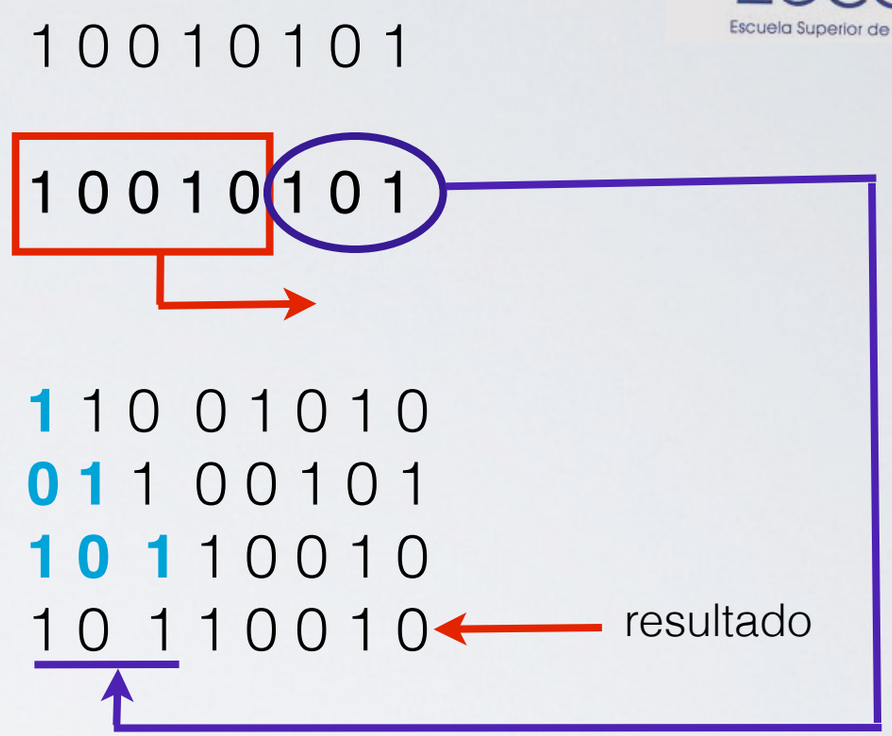


DISEÑO DEL CONVERTIDOR BINARIO A GRAY

ror → rotación a la derecha

Si A=10010101

A ror 3 =



NOTA: los operadores de desplazamiento o corrimiento sólo pueden ser usados con tipo de datos **BIT_VECTOR**. Por lo cual, si en la descripción se usa otro tipo de dato la instrucción "***TO***_" permite hacer el cambio de tipo de datos dentro de la arquitectura.

DISEÑO DEL CONVERTIDOR BINARIO A GRAY

Descripción en VHDL del método para convertir de binario a Gray usando operadores de corrimiento.

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY CON IS

PORT ( B:IN STD_LOGIC_VECTOR (3 DOWNTO 0);
          S:OUT STAD_LOGIC_VECTOR ( 3 DOWNTO 0)
         );
END ENTITY;

ARCHITECTURE A_CON OF CON IS
SIGNAL AUNX: STD_LOGIC_VECTOR (3 DOWNTO 0);

BEGIN

AUX<= TO_STDLOGIC( TO_BITVECTOR (B) SRL 1);
S<= B XOR AUX;


END A_CON;
```

PRÁCTICA

Convertidor de código BIN/GRAY y GRAY/BIN DE n BITS

- Realiza la descripción, compilación y simulación en VHDL de un convertidor de código de binario a Gray, cuando control sea 0, y de Gray a binario cuando control sea 1.
- El convertidor debe ser de 8 bits.
- Puedes utilizar para el convertidor binario a Gray lo revisado en esta presentación.
- Puedes utilizar la instrucción IF-THEN-ELSE para hacer la descripción de CONTROL.

CONTROL	SALIDA
0	BIN/GRAY
1	GRAY/BIN

PRÁCTICA

- Puedes usar algunos de los métodos de diseño revisados en la presentación para hacer la descripción del bloque que convierte de Gray a binario.
- A continuación se presenta el método para convertir cantidades de código Gray a código binario.

Método para convertir cantidades en código Gray a binario