Introducción al Diseño Lógico (E0301)

Ingeniería en Computación

Gerardo E. Sager

Clase 8 curso 2021

Clase 8

- Temas a tratar
  - Dualidad
  - Algebra Booleana y el mapa de Karnaugh como herramientas para simplificar y diseñar circuitos lógicos II
  - Operación de circuitos OR-Exclusiva y NOR-exclusiva. Aplicaciones.
  - Circuitos de Habilitación y Deshabilitación.
  - Codigos de Gray II

Mapas de Karnaugh o K-Maps

- ***Dualidad***:Si en una expresión booleana se reemplazan todos las variables por sus expresiones negadas, las opeeraciones OR por AND y viceverse, la expresión resultante es igual a la expresión original pero negada. A esta propiedad la llamamos DUALIDAD.
  - Basándose en la Dualidad se pueden construir K-Maps, orientados a simplificar las expresiones escribiéndolas como POS. donde las filas y columnas en vez de corresponder al AND de variables, corresponden al OR y ademas se agrupan los valores '0' en vez de los valores '1'.
  - No vamos a profundizar en este tema, porque vamos a utilizar otra manera de construir los K-Maps ya sea a partir de POS o SOP

Mapas de Karnaugh o K-Maps

- Cómo llenar un **mapa K** a partir de una expresión Booleana de tipo **SOP**
  - Si la expresión está expresada como SOP puede llenarse el K-MAP, poniendo '1's en los lugares correspondientes a los Productos y '0's en los restantes
- Cómo llenar un **mapa K** a partir de una expresión Booleana de tipo **POS**
  - Si bien se podria aplicar la propiedad de dualidad y construir un K-MAP basado en POS, lo más práctico es negar la expresión propuesta y aplicar deMorgan para convertirla en SOP. Esto describirá la función negada y a partir de alli podremos simplificarla:
  - $F=(A+B)(C+D) \Rightarrow \overline{F} = \overline{(A+B)(C+D)} = \overline{(A+B)} + \overline{(C+D)} = \overline{A}\,\overline{B} + \overline{C}\,\overline{D}$
  - A partir de alli podemos agrupar los valores necesarios, obtener la expresión simplificada y luego volverla a negar y aplicar DeMorgan nuevamente.

Ejemplo:

- Volvemos al ejemplo de la presentación 7, pagina 7. para el caso de POS
- El enunciado pedía diseñar un circuito lógico de tres entradas, A, B, y C. La salida debe ser HIGH cuando la mayoría de las entradas es HIGH.

A	B	C	F	$\overline{F}$	POS
0	0	0	0	1	$\overline{A}\ \overline{B}\ \overline{C}$
0	0	1	0	1	$\overline{A}\ \overline{B}\ C$
0	1	0	0	1	$\overline{A}\ B\ \overline{C}$
0	1	1	1	0	
1	0	0	0	1	$A\ \overline{B}\ \overline{C}$
1	0	1	1	0	
1	1	0	1	0	
1	1	1	1	0	

Habíamos obtenido a partir de

$\overline{F} = \overline{A}\ \overline{B}\ \overline{C} + \overline{A}\ \overline{B}C + \overline{A}B\overline{C} + A\ \overline{B}\ \overline{C}$

$F=(A+B+C)(A+B+\overline{C})(A+\overline{B}+C)(\overline{A}+B+C)$

Si aplicamos K-MAP a $\overline{F}$

	BC	00	01	11	10
A		$\overline{B}\ \overline{C}$	$\overline{B}\ C$	$B\ C$	$B\ \overline{C}$
0	$\overline{A}$	1	1	0	1
1	A	1	0	0	0

$\overline{F} = \overline{A\ B} + \overline{B\ C} + \overline{A\ C} = \overline{A+B} + \overline{B+C} + \overline{A+C}$

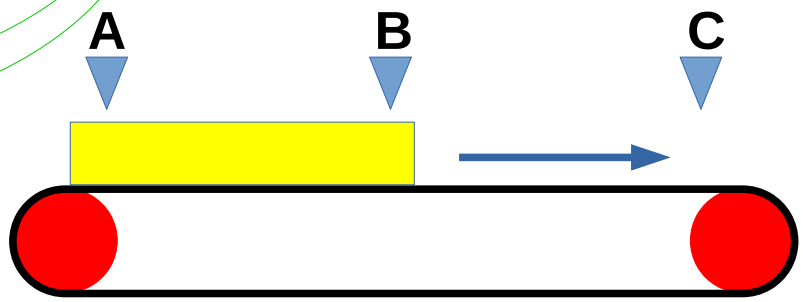
$\overline{F} = \overline{A+B} + \overline{B+C} + \overline{A+C} = (A+B)(B+C)(A+C)$

Se observa que utilizar el K-MAP nos dá una expresión más simple

Valores “Don't Care” o “No Importa”

- En algunos casos, cuando escribimos la TdV , hay situaciones en que no nos interesa el valor exacto de la salida cuando se produce cierta combinación de entradas.
- Por ejemplo, si tenemos un circuito de dos entradas, que sabemos que por una restricción externa nunca van a adoptar simultaneamente el valor '1'
- En ese caso, no nos importaría el valor que adopte la salida, y lo marcamos como “don't care” o “No importa” que se marca como X.

Valores “Don't Care” o “No Importa”



	$\overline{B}\,\overline{C}$	$B\,\overline{C}$	$B\,C$	$\overline{B}\,C$
$\overline{A}$	0	0	1	0
A	0	1	X	X

F= $AB\overline{C}+\overline{A}BC$ (X=0; X=0)
F= $AB+BC$ (X=1; X=0)

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

Supongamos un circuito con tres entradas en las cuáles no hay posibilidad de que las tres entradas tengan el mismo valor.

Por ejemplo podria ser una cinta transportadora donde las entradas son sensores que me indican la presencia de un objeto con un '1' y la ausencia con '0',

Si el sistema es como el de la figura, nunca vamos a tener tres sensores activados a la vez. Tampoco se activarán simultáneamente los sensores A y C

Si queremos que la salida valga 1 cuando hay dos sensores activados, escribimos la TdV y el K-Map

Cuando ubicamos la condición X en el K-MAP,depende si le asignamos valor 0 o 1, obtenemos expresiones más simples o más complejas

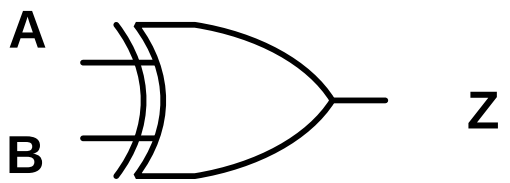
Circuitos XOR y XNOR

- OR exclusiva (**XOR**)

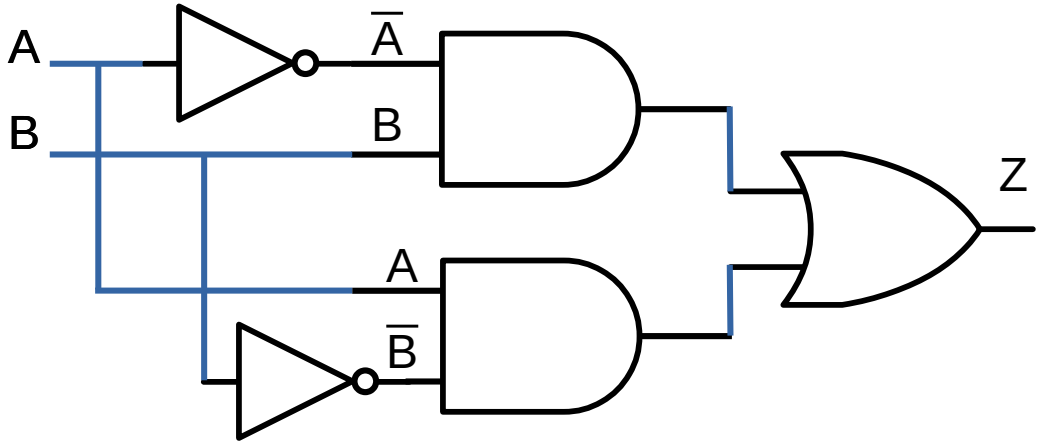
TdV		
A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

K-MAP		
	$\overline{B}$	B
$\overline{A}$	0	1
A	1	0

Símbolo



Implementación



$A \text{ XOR } B = \overline{A}B + A\overline{B}$

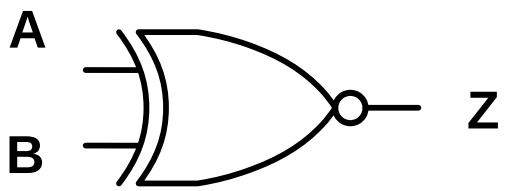
Circuitos XOR y XNOR

- NOR exclusiva (**XNOR**)

TdV		
A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

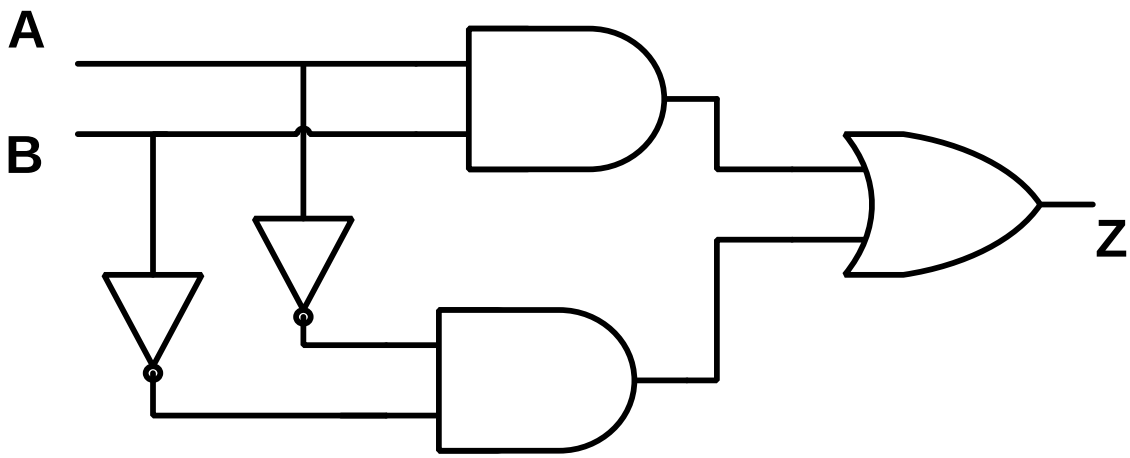
K-MAP		
	$\overline{B}$	B
$\overline{A}$	1	0
A	0	1

Símbolo



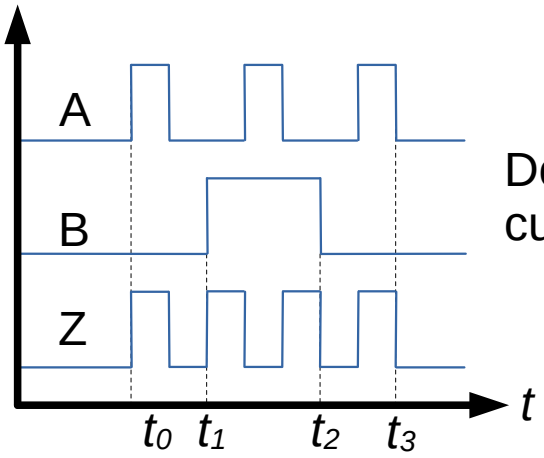
Implementación

$$A \text{ XNOR } B = \overline{A}\,\overline{B} + A\,B$$

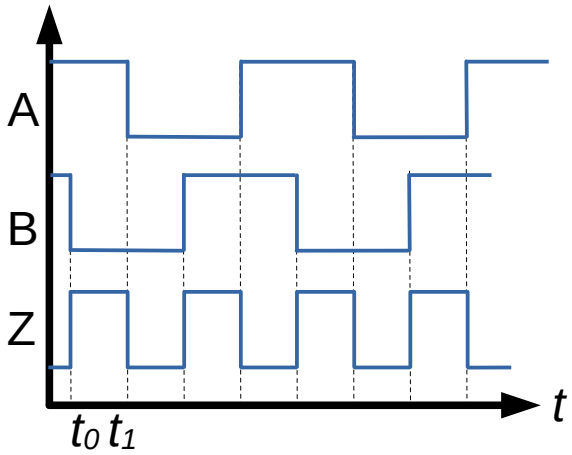


Circuitos XOR y XNOR

- OR exclusiva (**XOR**) Salida en el tiempo para diferentes entradas

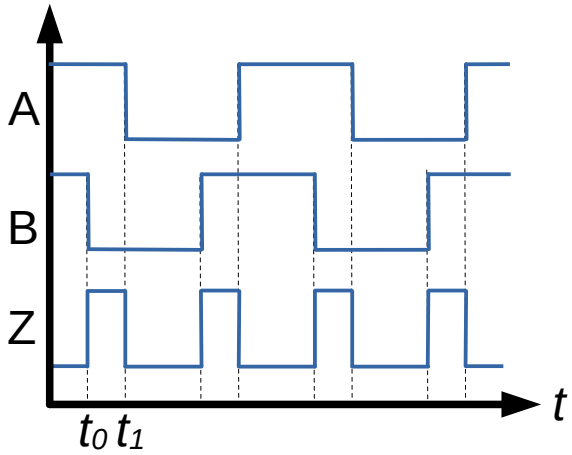


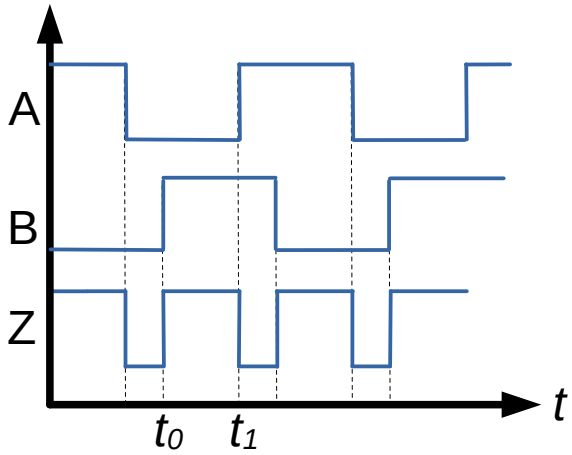
Dos entradas cualquiera

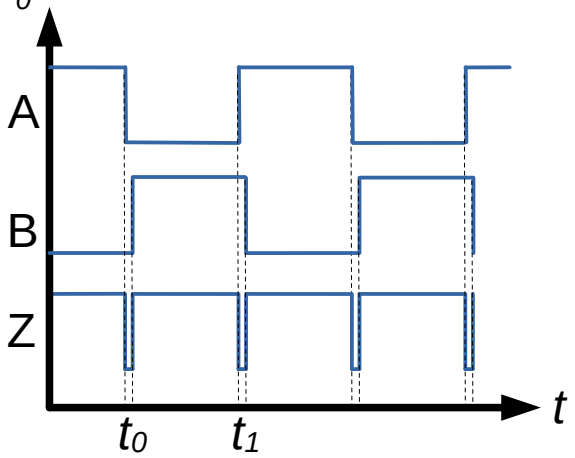


Las entradas son dos señales de la misma frecuencia, pero defasadas 90º

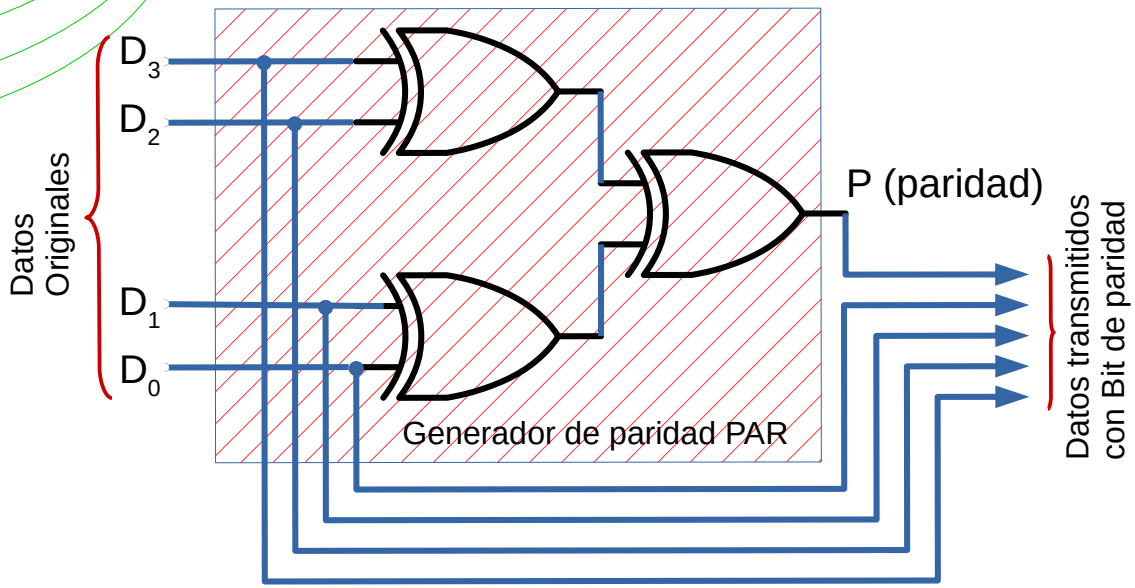
Seguimos con entradas de la misma frecuencia, con distintos defasajes
Notar como varía el ancho de los pulsos de salida $\Delta t = t_1 - t_0$

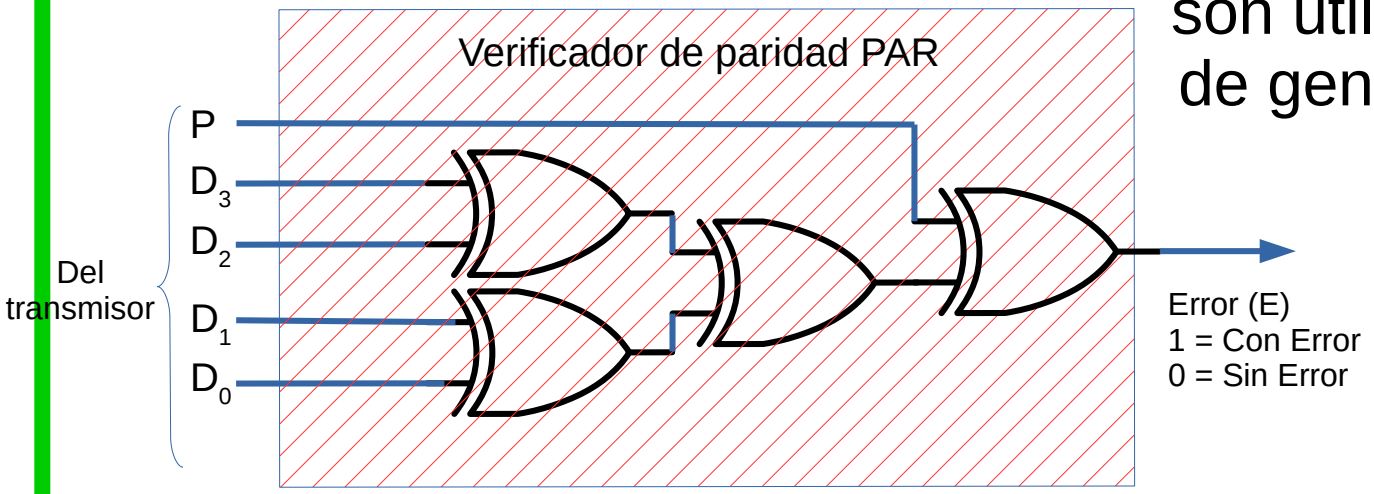






Generador y Verificador de Paridad



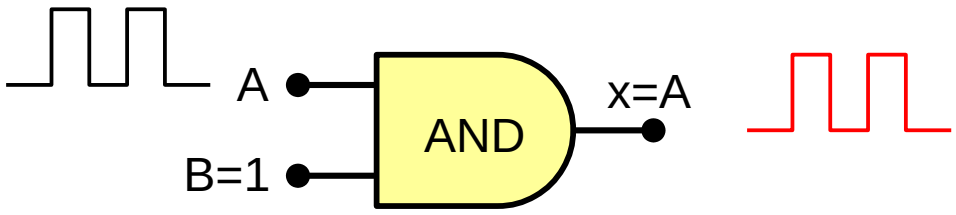


Las compuertas XOR y XNOR son útiles para diseñar circuitos de generación y verificación de paridad

Circuitos de habilitación y deshabilitación

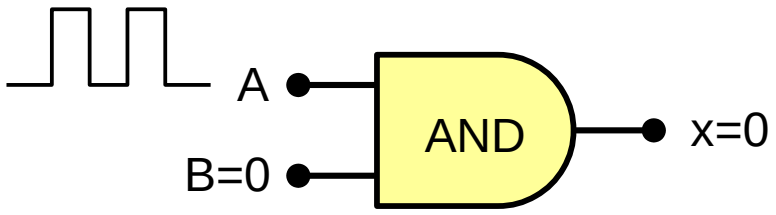
- En el diseño de circuitos digitales ocurren frecuentemente situaciones donde se requiere circuitos de habilitación/deshabilitación .
  - Un circuito está **habilitado** (**enabled**) cuando permite el paso de la señal de entrada a la salida.
  - Un circuito está **deshabilitado** (**disabled**) cuando evita el paso de la señal de entrada a la salida.

ENABLE



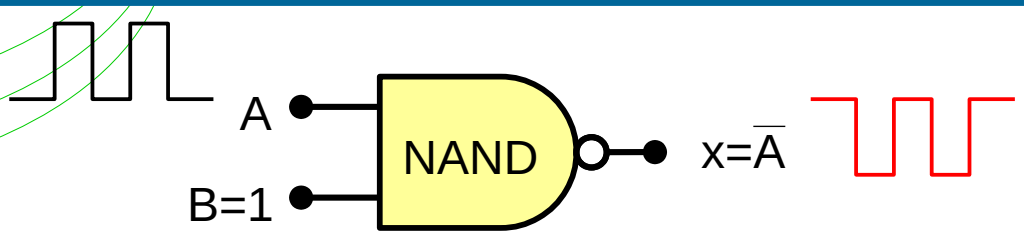
B=1, HABILITA que pase la señal A

DISABLE

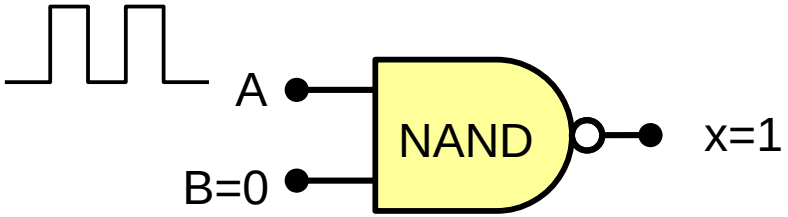


B=0, DESHABILITA el paso de la señal A

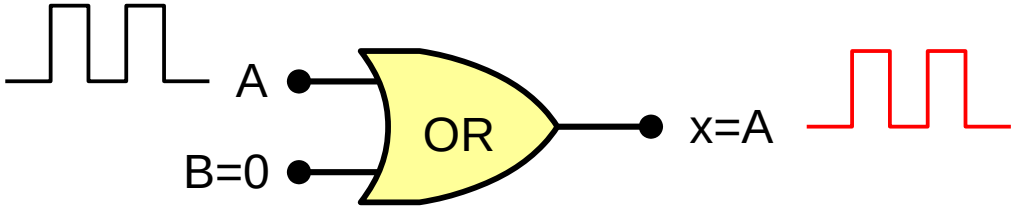
Circuitos de habilitación y deshabilitación



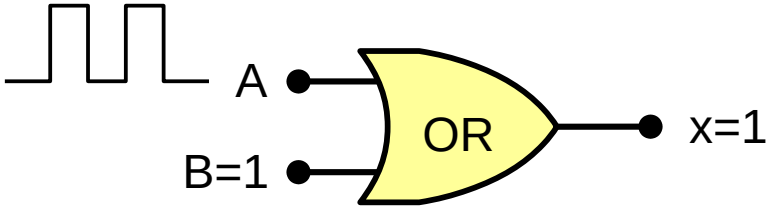
B=1, HABILITA que pase la señal A INVERTIDA a la salida



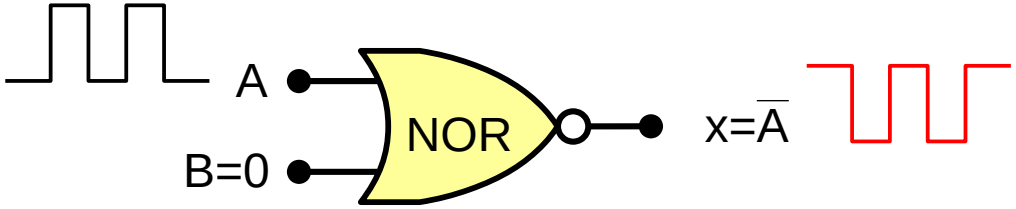
B=0, DESHABILITA que pase la señal A a la salida



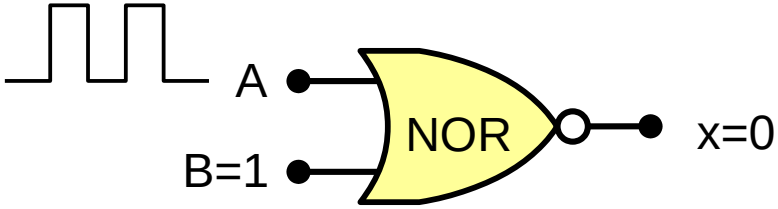
B=0, HABILITA que pase la señal A a la salida



B=1, DESHABILITA que pase la señal A a la salida



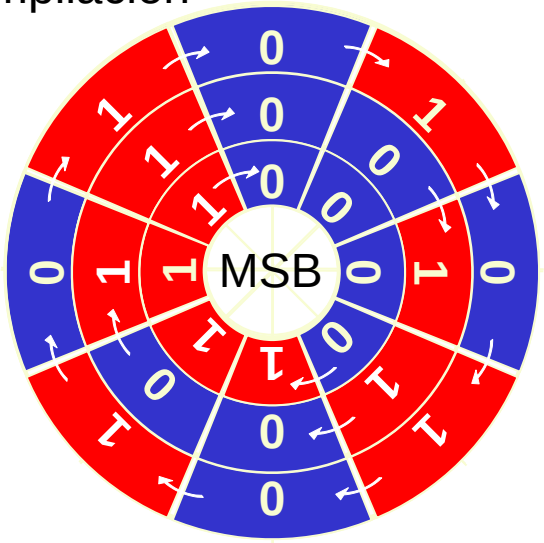
B=0, HABILITA que pase la señal A INVERTIDA a la salida



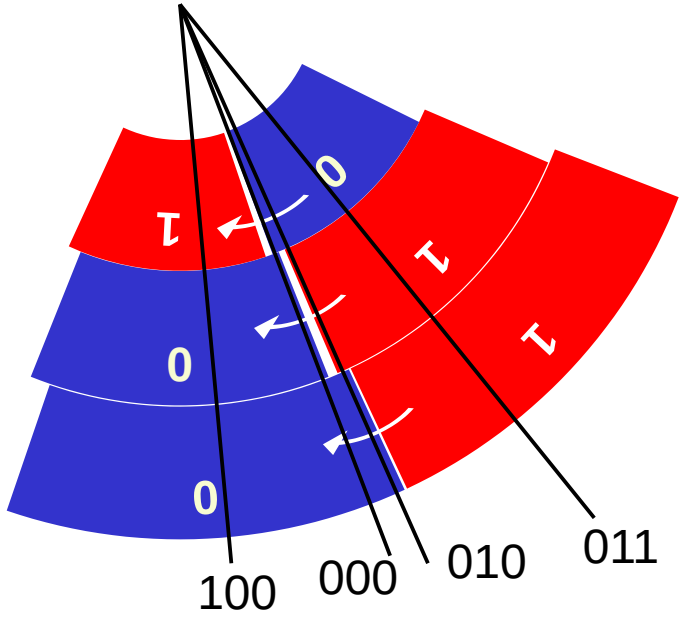
B=1, DESHABILITA que pase la señal A a la salida

Código de Gray

- Para determinar la posición de un eje, tomamos un disco, y lo dividimos en anillos y en sectores y usamos algun tipo de sensor que nos permita determinar si corresponde a un 1 o a un 0.
- Por ejemplo podriamos recortar los sectores donde hay '1's y ubicar sensores de luz bajo el disco y una fuente de luz arriba, cuando se detecta la luz hay un 1 y si no hay un 0
- Si analizamos los dos sectores 011 y 100 contiguos entre sí, vemos que cambian todos los bits al pasar de uno al otro.
- Esto puede producir que por imperfecciones de alineación o de tiempos de respuesta de los sensores, entre 011 y 100, se produzcan valores intermedios como tambien vemos en la ampliación



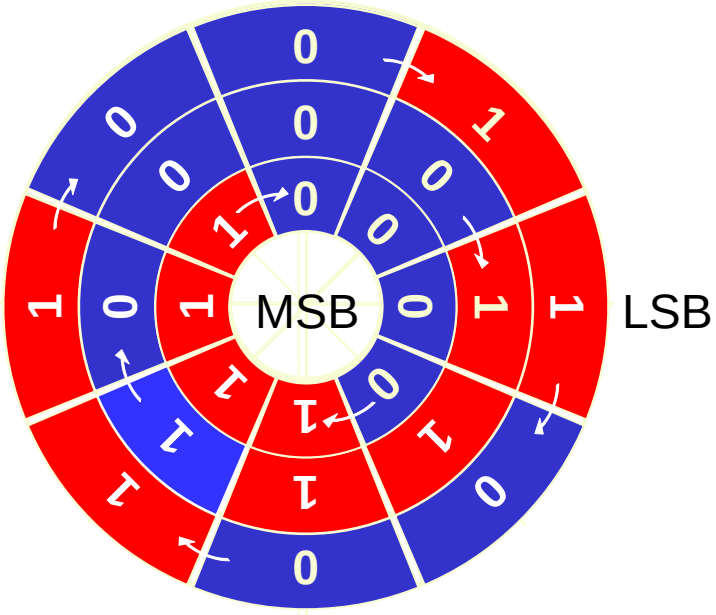
Orden Binario natural



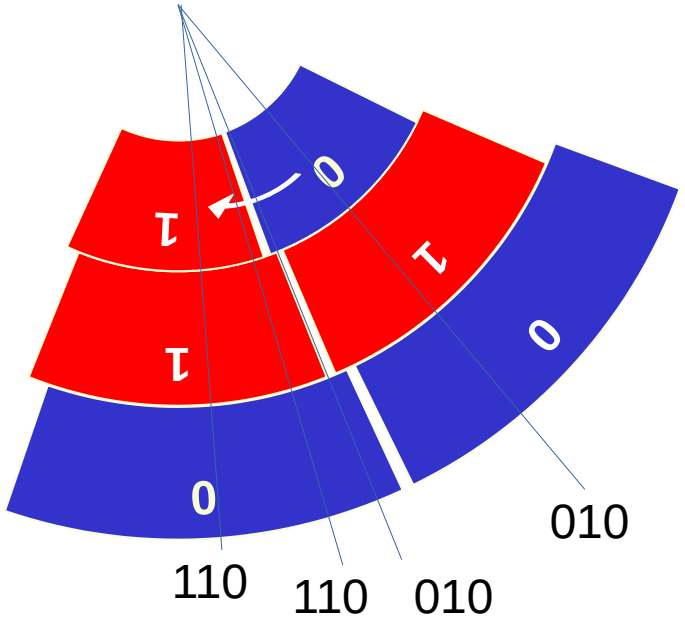
Imperfecciones en la transición generan lecturas erróneas

Código de Gray

- Estos valores no esperados pueden traernos problemas y vemos que una de las causas es que se produzcan multiples cambios de bits al pasar de un sector a otro, en el caso que analizamos hay tres cambios, en otros puede haber dos.
- Si conseguimos que entre sector y sector nunca se cambie más de un bit, eliminamos el problema.
- Necesitamos una manera de numerar los sectores que cumpla con esta propiedad, Esta manera de ordenar los valores, se llama un código de GRAY



Orden Código GRAY



Las imperfecciones en las transiciones NO causan errores

Código de Gray

La construcción del código es simple:

- Se comienza por el LSB, se le da valores. Al resto de los bits se los completa con 0. (a)
- Cuando no pueda hacer más combinaciones, copio abajo la imagen en espejo de lo que tenía hasta ahora, y al bit que sigue hacia el MSB lo pongo en 1 (b)
- Repito el proceso (c)

A	B	C	
0	0	0	a)
0	0	1	

A	B	C	
0	0	0	a)
0	0	1	
0	**1**	1	b)
0	**1**	0	

A	B	C	
0	0	0	a)
0	0	1	
0	**1**	1	b)
0	**1**	0	
1	1	0	c)
1	1	1	
1	0	1	
1	0	0	