

Lab 01

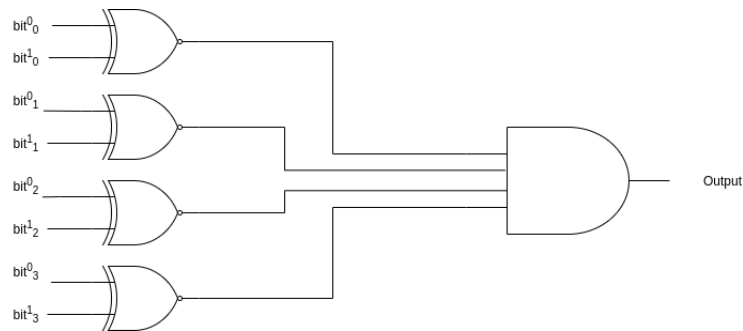
Arquitectura de Computadores

Sección 2

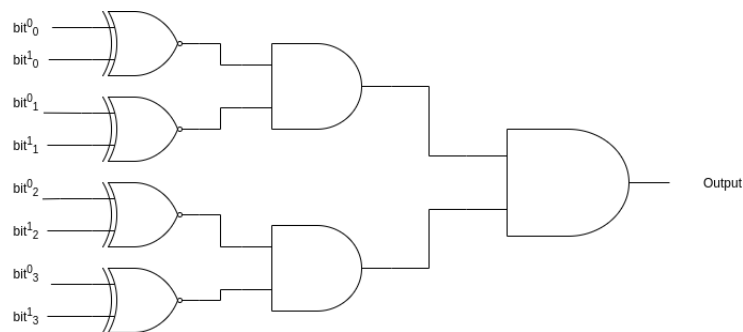
Joaquín Ramírez

April 22, 2020

1. Dados dos números de 4-bits, se comparan los pares i -ésimos, $i = 0, 1, 2, 3$ con XOR gates. Finalmente se conectan en un AND gate, el cual solo se activará cuando todos los pares estén activados.

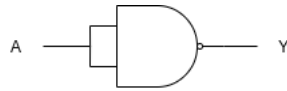


Si solo se pueden utilizar 2-input XNOR y 2-input AND gates, entonces el diagrama cambia.

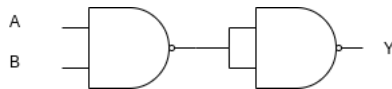


2. Se adjunta un archivo .asc con la simulación del ejercicio anterior.
3. Los diagramas con NAND gates de las funciones booleanas son los siguientes:

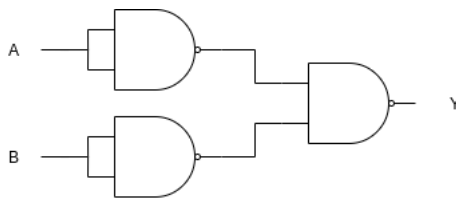
- NOT: se divide un input en dos y pasa por un NAND, lo que significa que simplemente se invertirán los posibles casos 0 y 1.



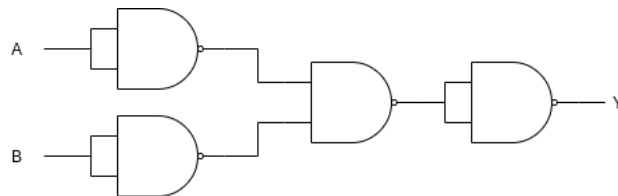
- AND: se niega un NAND con el NOT del ejercicio anterior.



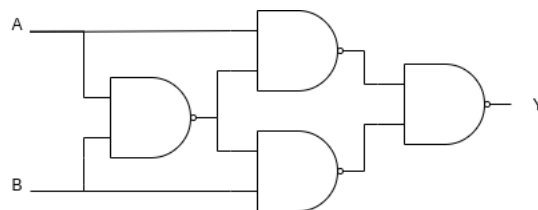
- OR: ya que la tabla de verdad del NAND es la misma que la del OR pero con los inputs invertidos, se niegan los inputs y se aplica un NAND después.



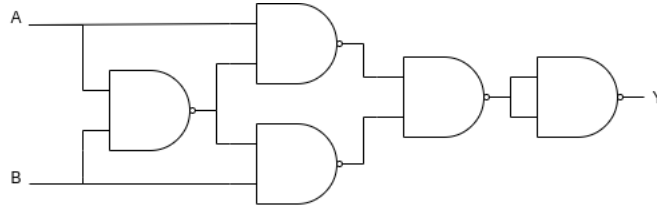
- NOR: al negar el OR, sacando dos inputs de su resultado, se obtiene el NOR.



- XOR: Partiendo de la Suma de Productos podemos llegar a la forma simplificada del diagrama con NANDs. $Y = \overline{A}\overline{B} + \overline{A}B + A\overline{B} + AB = \overline{A}(\overline{B} + B) + A(\overline{B} + B) = \overline{A}(\overline{\overline{B}}) + A(\overline{\overline{B}}) = \overline{A(\overline{B})} \overline{B(\overline{A})}$



- XNOR: se aplica un NAND a la respuesta del XOR.



4. (a) Siguiendo la lógica del ejercicio 1 y si no hay restricciones, se observa que la cantidad de compuertas XNOR es igual a n , donde n es el número de bits. Después, se añade una compuerta AND, la que verifica si en efecto todos los pares de bits son idénticos. Así, se puede decir que

$$\# \text{ compuertas} = n + 1$$

Por otra parte, si solo se permiten compuertas de tipo 2-input, entonces

$$\# \text{ compuertas} = n_{XOR} + (n - 1)_{AND}$$

bits	XNOR gate	AND gate	Total
8	8	1	9
16	16	1	17
32	32	1	33
64	64	1	65

(b) ...

(c) La tabla describe el comportamiento para distintos bits.

Comparator width	2-input XNOR gates	2-input AND gates	Logic Depth
8 bits	8	7	4
16 bits	16	15	5
32 bits	32	31	6
64 bits	64	63	7

- (d) Se puede proponer que $N = 2^i$, $i \in \{0, 1, 2, 3, 4, \dots\}$, entonces la tabla previa se generaliza a:

Comparator width	2-input XNOR gates	2-input AND gates	Logic Depth
N bits	N	N - 1	$i + 1$

5. El SR-Latch es un circuito secuencial simple que almacena 1 bit. Trabaja con dos inputs S y R, y con dos outputs Q y \overline{Q} . Se pueden presentar 4 casos:

- $S = 1, R = 1$: este es el caso “ideal” o “callado”, ya que el valor de Q será el mismo que el de Q_{prev} .
- $S = 0, R = 1$: este es el caso “reset”, ya que el valor de Q será “bajado” a 0.
- $S = 1, R = 0$: este es el caso “set”, ya que el valor de Q será “subido” a 1.
- $S = 0, R = 0$: este es el caso “prohibido” o “metaestable”, ya que el valor de Q será el mismo que el de \overline{Q} y los valores asignados a cada uno dependen de las propiedades eléctricas de los transistores que componen las compuertas, mas no de la lógica.

