

# Lab 01

## Arquitectura de Computadores

### Sección 2

Joaquín Ramírez

Abril 24, 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.

1.png

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

0.png

2. Se adjunta la simulación en LTSpice del ejercicio anterior.

dia11.png

plot11.png

dia12.png

plot12.png

3. Los diagramas con NAND gates de ciertas funciones booleanas son:

- 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.

2.png

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

3.png

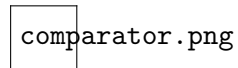
- XOR: Partiendo de la Suma de Productos podemos llegar a la forma simplificada del diagrama con NANDs.  $Y = A\overline{B} + \overline{A}B = \overline{\overline{A\overline{B} + \overline{A}B}} = \overline{\overline{A\overline{B}} \cdot \overline{\overline{A}B}} = \overline{A(\overline{A\overline{B}})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

6.png

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



A partir de esto, el comparador del inciso 1 quedaría de la siguiente forma:



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

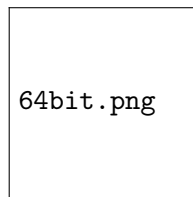
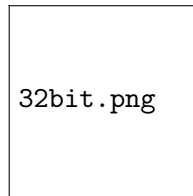
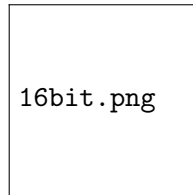
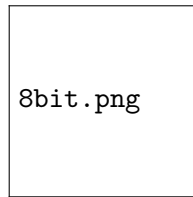
$$\# \text{ compuertas} = n + 1$$

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) Los comparadores en bloque son:



(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:

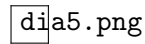
Comparator width	2-input XNOR gates	2-input AND gates	Logic Depth
N bits	N	N - 1	i + 1

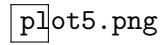
5. El SR-Latch es un circuito secuencial simple que almacena 1 bit. Trabaja con dos inputs S y R, y con dos outputs  $Q$  y  $\overline{Q}$ . Se pueden presentar 4 casos:

- $S = 1, R = 1$  : 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  $\bar{Q}$  y los valores asignados a cada uno dependen de las propiedades eléctricas de los transistores que componen las compuertas, mas no de la lógica.




8.png


dia5.png

plot5.png

6. El Gated D Flip-Flop sirve para regularizar la grabación de datos en los registros, para que la data esté disponible durante todo el ciclo del clock. Se tienen dos D-Latches, la primera es la maestra y la segunda la esclava. El funcionamiento es simple: cuando el  $CLK = 0$ , el maestro pasa el valor de  $D$  a la “puerta” del esclavo. Como no se activa su  $WE$ , entonces el valor de  $Q$  será el de  $Q_{prev}$ . En el caso contrario, cuando  $CLK = 1$ , el maestro no propaga ningún valor, por lo que el valor de  $Q$  será el que “esperaba en la puerta” del esclavo. Así, en el “rising edge” de  $0 \rightarrow 1$ ,  $Q = D$ . En cualquier otro momento  $Q$  no varía.

9.png

 dia6.png

 plot6.png