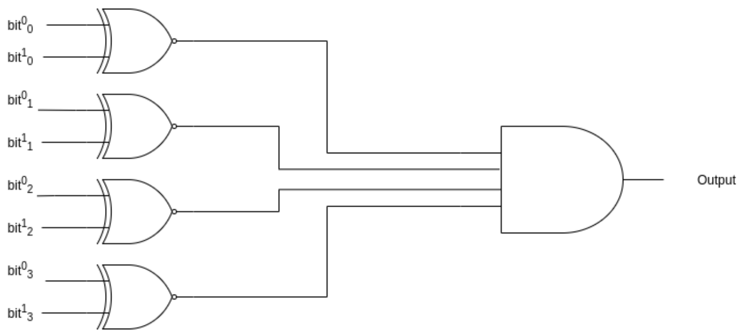
Lab 01
Arquitectura de Computadores
Sección 2

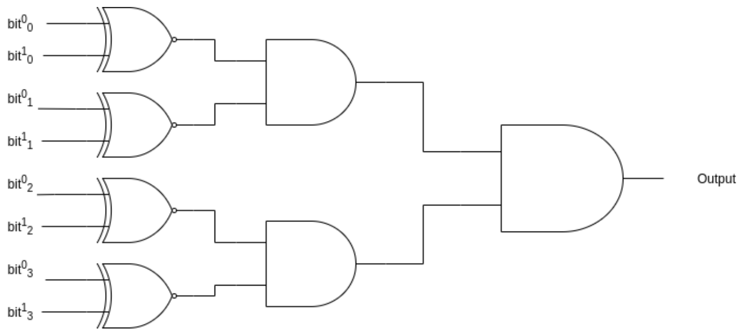
Joaquín Ramírez

April 24, 2020

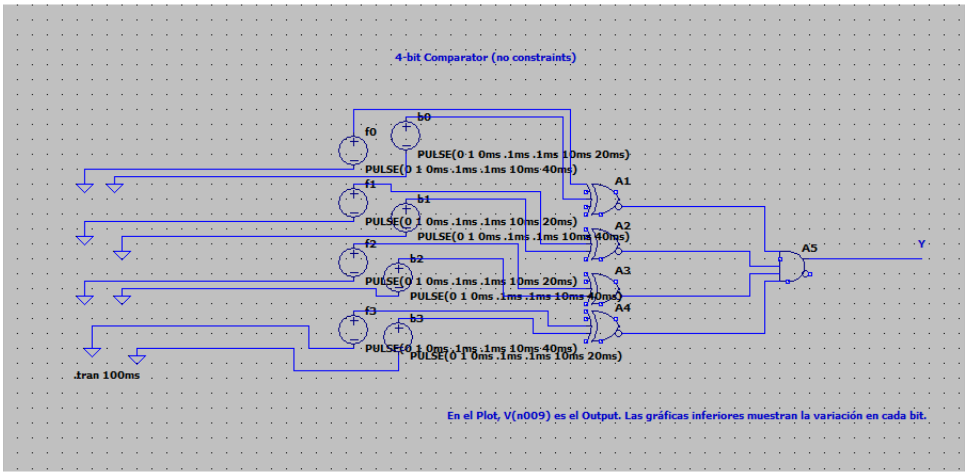
1. Dados dos números de 4-bits, se comparan los pares i-ésimos, $i = 0, 1, 2, 3$ con XOR gates. Finalmente se conectan en un AND gate, el cual solo se activará cuando todos los pares estén activados.

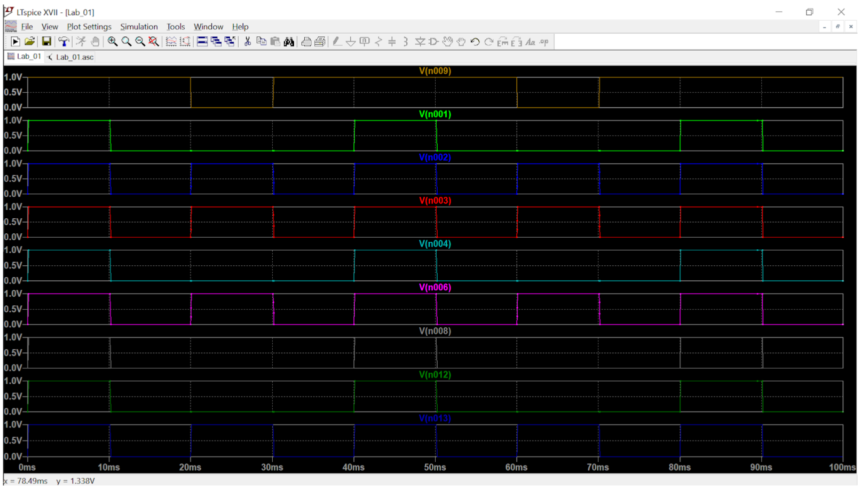


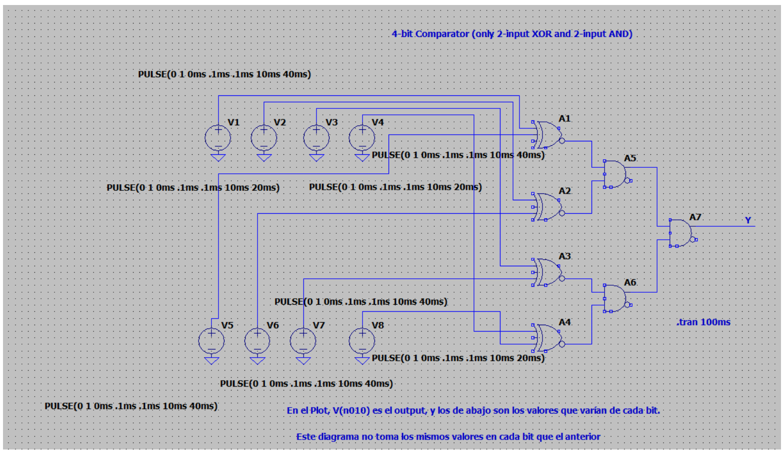
   Si solo se pueden utilizar 2-input XNOR y 2-input AND gates, entonces el diagrama cambia.

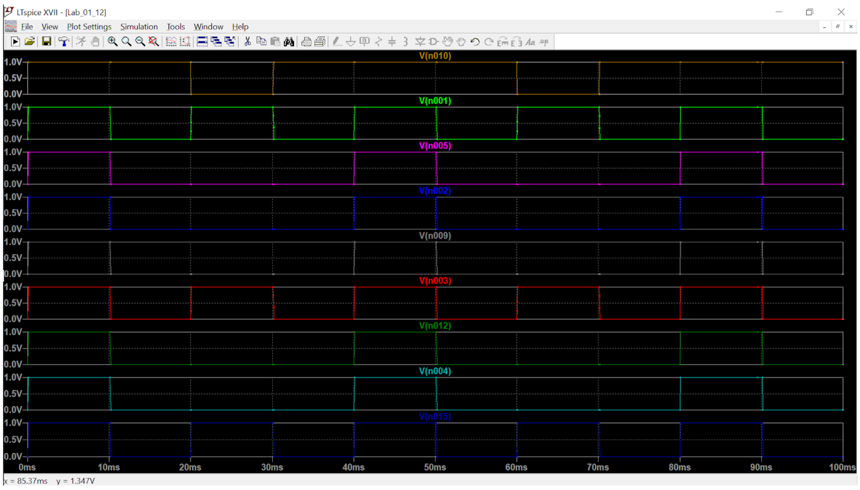


2. Se adjunta la simulación en LTSpice del ejercicio anterior.


4-bit Comparator (no constraints)
f0
b0
PULSE(0 1 0ms .1ms .1ms 10ms 20ms)
PULSE(0 1 0ms .1ms .1ms 10ms 40ms)
f1
b1
A1
A2
A3
A4
A5
Y
f2
b2
f3
b3
.tran 100ms
En el Plot, V(n009) es el Output. Las gráficas inferiores muestran la variación en cada bit.

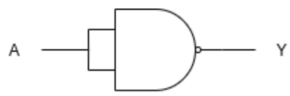

LTspice XVII - [Lab_01]
File View Plot Settings Simulation Tools Window Help
V(n009)
V(n001)
V(n002)
V(n003)
V(n004)
V(n006)
V(n008)
V(n012)
V(n013)
1.0V
0.5V
0.0V
0ms
10ms
20ms
30ms
40ms
50ms
60ms
70ms
80ms
90ms
100ms
x = 78.49ms y = 1.338V


4-bit Comparator (only 2-input XOR and 2-input AND)
PULSE(0 1 0ms .1ms .1ms 10ms 40ms)
V1
V2
V3
V4
A1
A2
A3
A4
A5
A6
A7
Y
PULSE(0 1 0ms .1ms .1ms 10ms 20ms)
V5
V6
V7
V8
.tran 100ms
En el Plot, V(n010) es el output, y los de abajo son los valores que varian de cada bit.
Este diagrama no toma los mismos valores en cada bit que el anterior

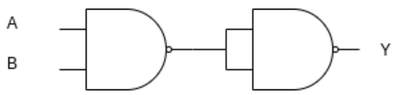


3. Los diagramas con NAND gates de ciertas funciones booleanas son:

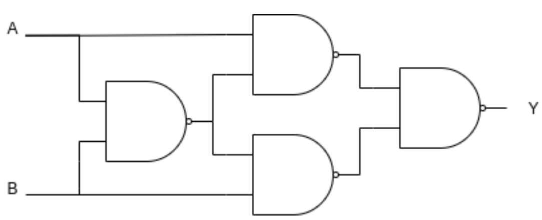
- NOT: se divide un input en dos y pasa por un NAND, lo que significa que simplemente se invertirán los posibles casos 0 y 1.



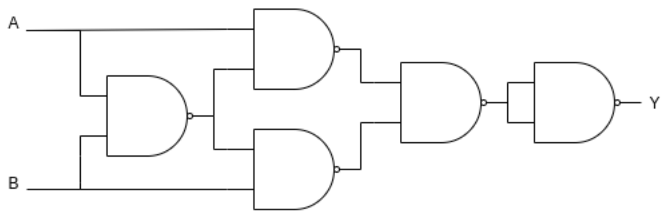
- AND: se niega un NAND con el NOT del ejercicio anterior.



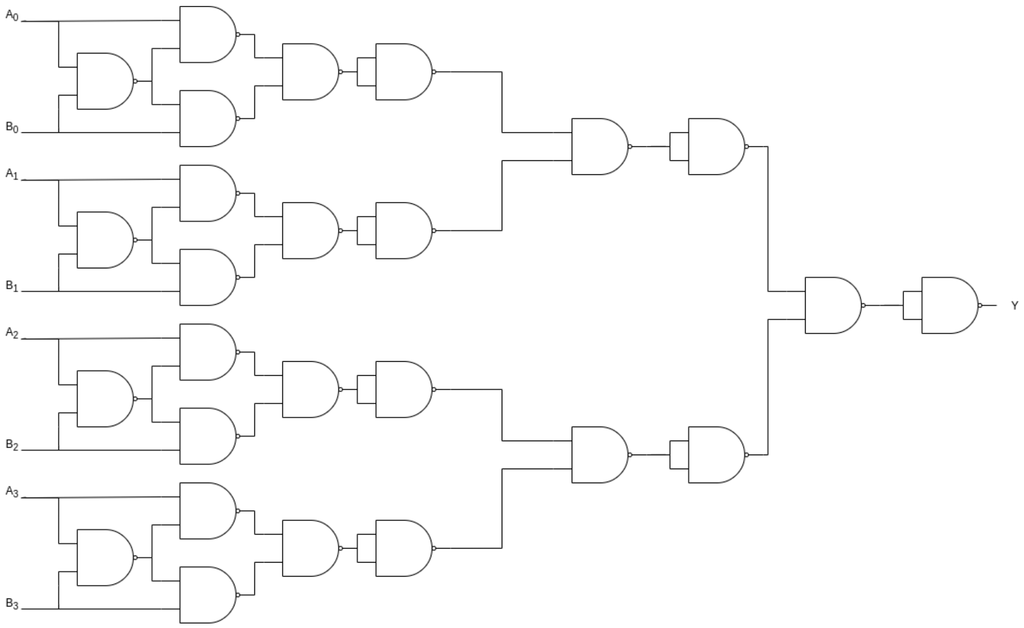
- XOR: Partiendo de la Suma de Productos podemos llegar a la forma simplificada del diagrama con NANDs. $Y = A\overline{B} + \overline{A}B = A\overline{B} + \overline{A}A + \overline{A}B + \overline{B}B = A(\overline{A} + \overline{B}) + B(\overline{A} + \overline{B}) = \overline{\overline{A(\overline{AB})}}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



- 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 ńumero 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

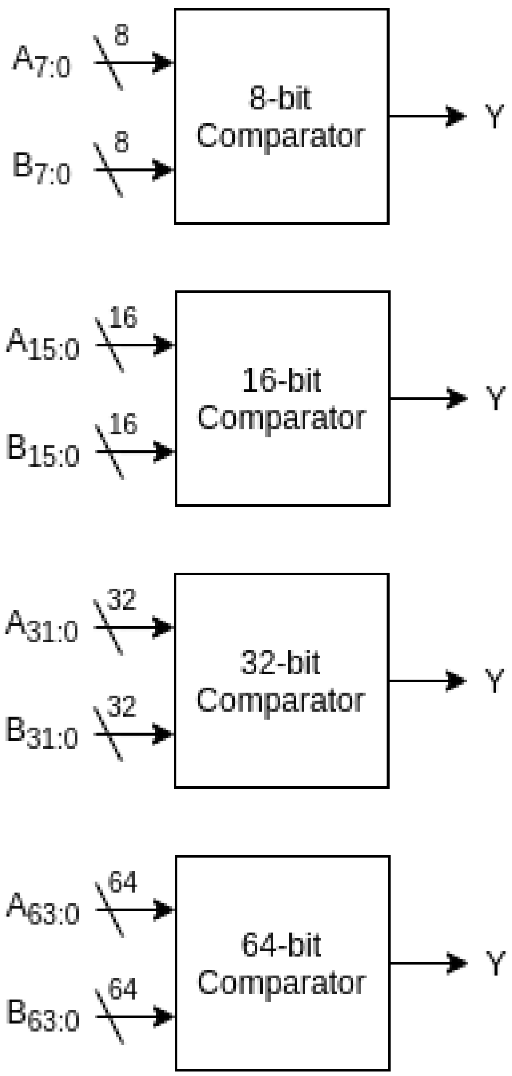
$$\# \; compuertas = n + 1$$

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

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

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.

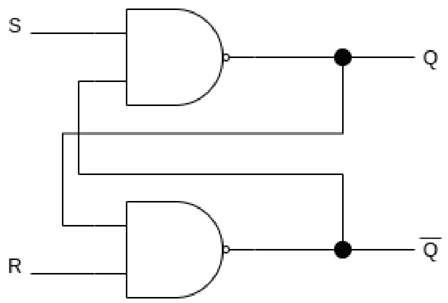
Comparator width	2-input XNOR gates	2-input AND gates	Logic Depth
8 bits	8	7	4
16 bits	16	15	5
32 bits	32	31	6
64 bits	64	63	7

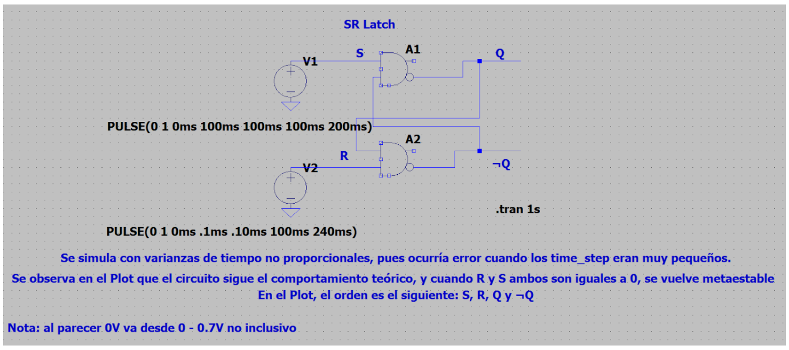
(d) Se puede proponer que $N = 2^i$, $i \in \{0, 1, 2, 3, 4, ...\}$, entonces la tabla previa se generaliza a:

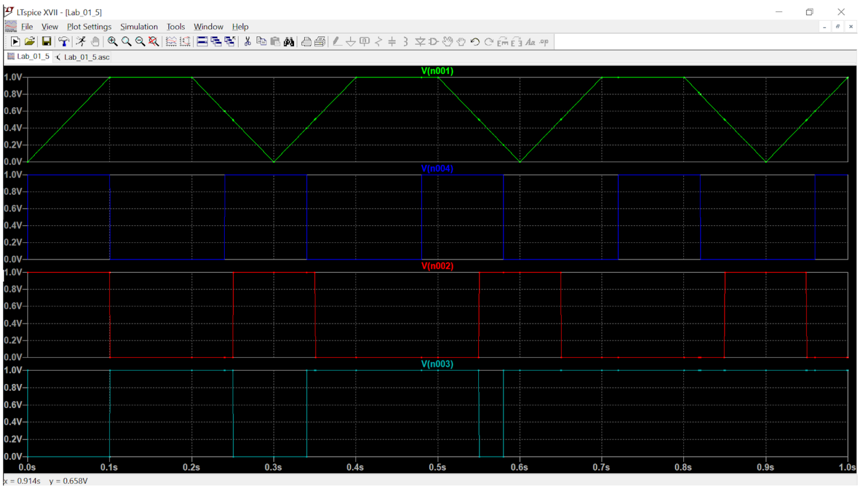
Comparator width	2-input XNOR gates	2-input AND gates	Logic Depth
N bits	N	N - 1	i + 1

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:

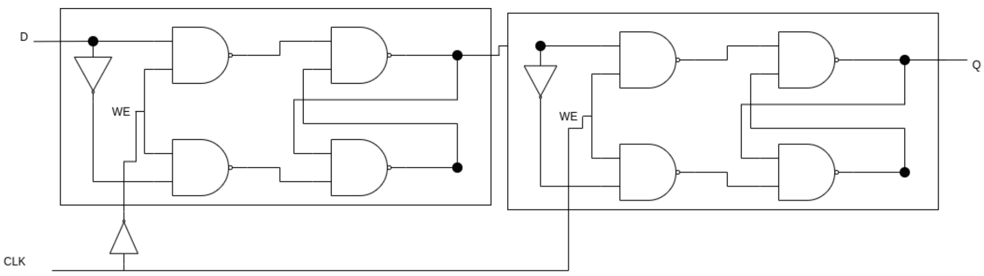
- $S = 1, R = 1$: este es el caso "ideal" o "callado", ya que el valor de Q será el mismo que el de Q_{prev}.
- $S = 0, R = 1$: este es el caso "reset", ya que el valor de Q será "bajado" a 0.
- $S = 1, R = 0$: este es el caso "set", ya que el valor de Q será "subido" a 1.
- $S = 0, R = 0$: este es el 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.

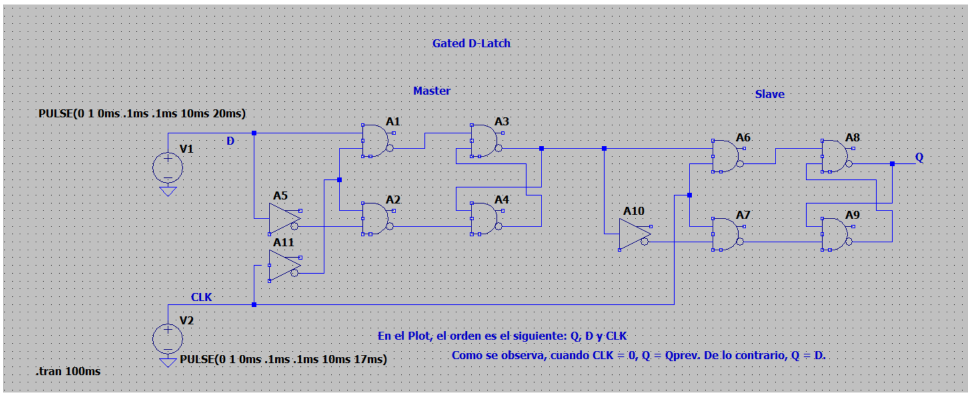


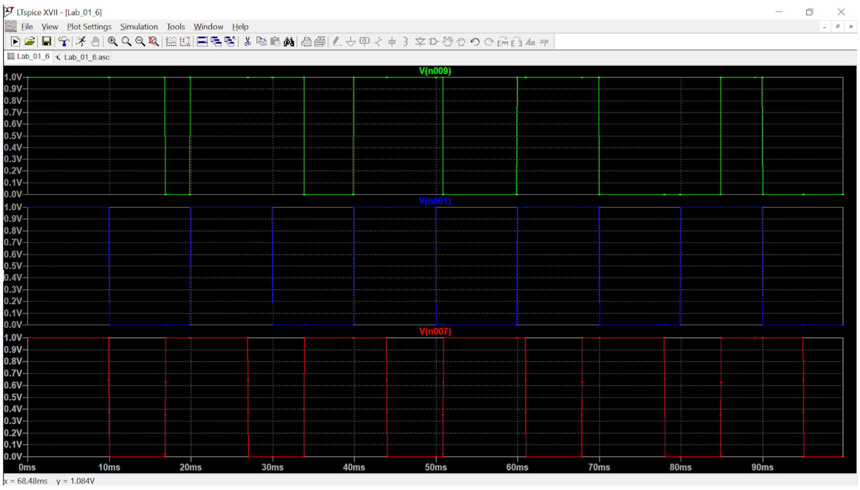




6. El Gated D Flip-Flop sirve para regularizar la grabación de datos en los registros, para que la data esté disponible durante todo el ciclo del clock. Se tienen dos D-Latches, la primera es la maestra y la segunda la esclava. El funcionamiento es simple: cuando el $CLK = 0$, el maestro pasa el valor 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 $0- > 1$, $Q = D$. En cualquier otro momento Q no varía.




Gated D-Latch
Master
Slave
PULSE(0 1 0ms .1ms .1ms 10ms 20ms)
V1
D
A1
A2
A3
A4
A5
A11
A6
A7
A8
A9
A10
Q
CLK
V2
PULSE(0 1 0ms .1ms .1ms 10ms 17ms)
.tran 100ms
En el Plot, el orden es el siguiente: Q, D y CLK
Como se observa, cuando CLK = 0, Q = Qprev. De lo contrario, Q = D.


V(n009)
V(n007)
0ms
10ms
20ms
30ms
40ms
50ms
60ms
70ms
80ms
90ms