



Practica 9

Comparador

Profesor: Barrón Vera José Emanuel

Materia: Fundamentos de diseño digital

Grupo: 3CV6

Alumno: Cazares Cruz Jeremy Sajid

Boleta: 2021630179



Galaxy

Copyright © 1998-2001 Cypress Semiconductor. All Rights Reserved.

Para la realización de la practica del comparador de 4 bits se tuvo como medida las siguientes formas:

```
1 LIBRARY ieee;
2 USE IEEE.std_logic_1164.all;
3
4 ENTITY comparador IS PORT(
5     A,B: in std_logic_vector(3 downto 0);
6     signal mayor, menor, igual: std_logic;
7     display: out std_logic_vector(6 downto 0)
8 );
9 END comparador;
10
11 ARCHITECTURE resultado of comparador IS
12 BEGIN
13
14     mayor <= '1' when (A>B) else '0';
15     menor <= '1' when (A<B) else '0';
16     igual <= '1' when (A=B) else '0';
```

figura 1 código

De tal manera que compara directamente los 4 bits del primer número “A” y los 4 bits del segundo número “B” de tal manera que si el primer número es mayor al segundo la señal “mayor” será 1, de caso contrario la señal “menor” será igual a 1 y finalmente mediante el uso de un else pero de manera ilustrativa se tiene el que los dos números sean iguales donde entonces la señal “igual” será igual 1.

```
10
11 ARCHITECTURE resultado of comparador IS
12 BEGIN
13
14     mayor <= '1' when (X(3)>Y(3) or X(2)>Y(2) or X(1)>Y(1) or X(0)>Y(0)) else '0';
15     menor <= '1' when (X(3)<Y(3) or X(2)<Y(2) or X(1)<Y(1) or X(0)<Y(0)) else '0';
16     igual <= '1' when (X(3)=Y(3) or X(2)=Y(2) or X(1)=Y(1) or X(0)=Y(0)) else '0';
17
```

figura 2 código 2

Mediante la realización de la practica se pidió el uso de que el comparador iniciara en el bit más significativo al menos significativo por lo que se hace la implementación de ir comparando de alguna manera bit por bit tal como se muestra en la figura 2.

De tal manera que para saber cuándo un número es mayor a otro se usa la suma “OR” para que esto sea cierto al momento de tener que un bit sea mayor a otro

Finalmente, para la parte del decodificador del display se utilizó una sentencia “if” de tal manera que se puede ir “preguntando” que posición esta en 1, si es mayor entonces la señal “mayor” deberá estar en 1 y de caso contrario la señal “menor” deberá estar en 1, al tener esto claro se puede saber que dato se tendrá que imprimir dependiendo del estado de las señales, esto se puede observar mejor en la siguiente figura:

```

PROCESS ( mayor, menor)
BEGIN

    IF( mayor = '1' ) THEN
        display <= "1111000";    --MAYOR QUE
    ELSIF ( menor = '1' ) THEN
        display <= "1001110";    --MENOR QUE
    ELSE
        display <= "0001001";    --IGUAL
    END IF;

END PROCESS;

```

figura 3 código display

Al final el “else” mostrara el signo de igual ya que al no encontrar que uno sea mayor al otro quiere decir que ambos son iguales, en un caso inicial donde ambos números sean igual a 0 querrá decir que ambos son iguales por lo cual es correcto decir que desde un caso inicial los dos números son iguales.

Para la forma del igual se manejó la forma de:

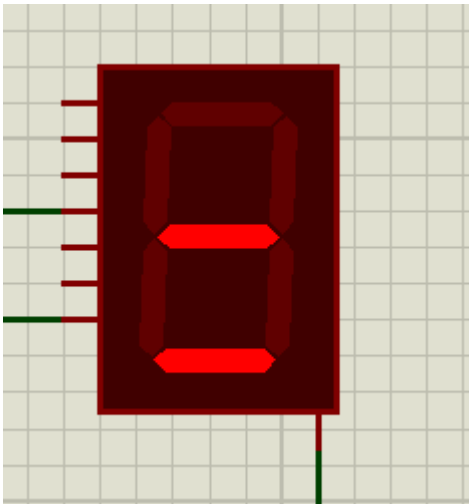


figura 4 símbolo igual manejado

De tal manera que el código BCD para la figura es: “0001001”

Esto sin contar el punto del display

A continuación, las simulaciones:

Para las partes de las simulaciones al momento de utilizar el display se tiene un error de escritura en la señal por lo que las pruebas solo serán sin el display

Simulación bit a bit

Para las simulaciones con el código bit a bit la señal igual será siempre será a igual a 1 ya que siempre encontrará un dato parecido, ya sea un 1 o 0 a menos que el primer número sea 15 y el segundo sea 0 o viceversa

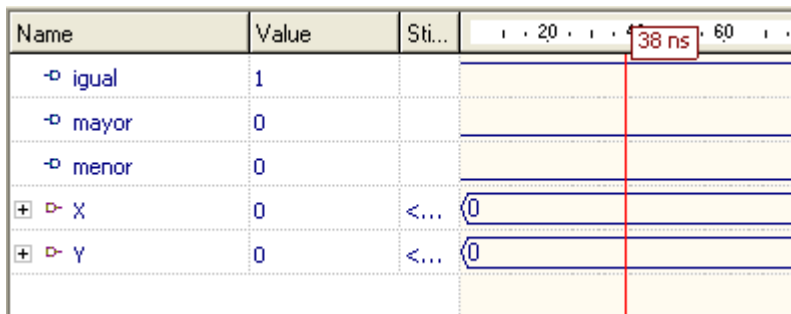


figura 5 simulación 1

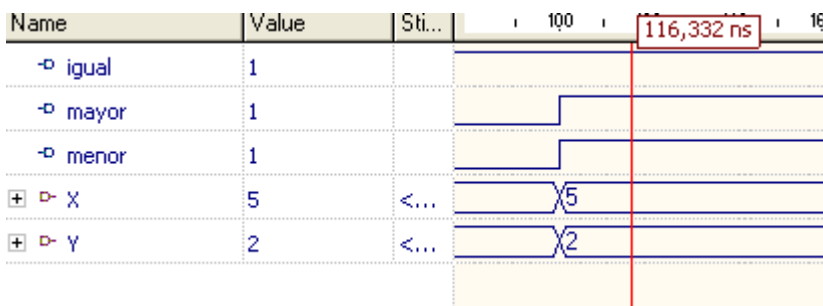


figura 6 simulación 2

Esto también pasa con los siguientes dígitos al indicar que la suma de los datos sea 1 o 0 siempre se tendrán bits mayores y menores con respecto a los números

Con la excepción de cuando ambos números sean 0, cuando esto suceda ninguno de los dos números sea mayor o menor respecto al otro solo serán iguales

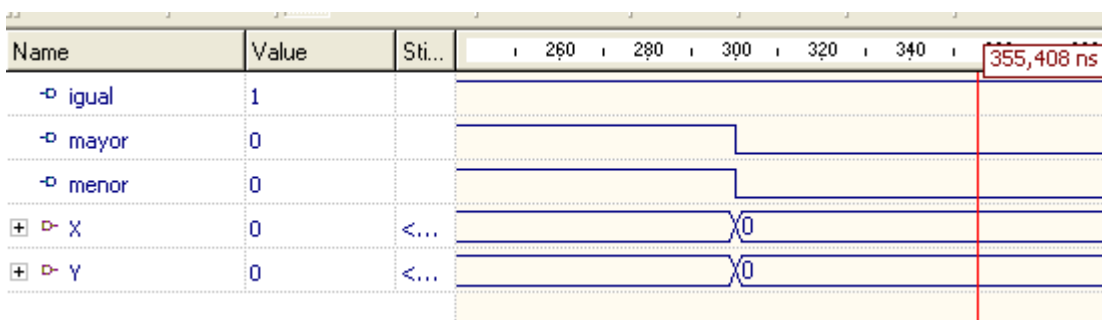


figura 7 simulación 3

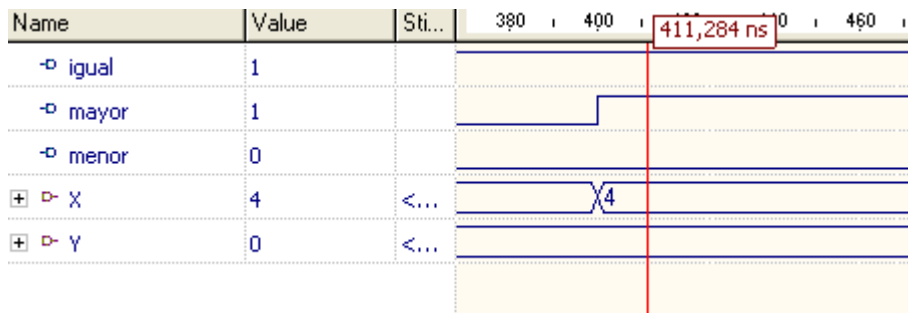


figura 8 simulación 4

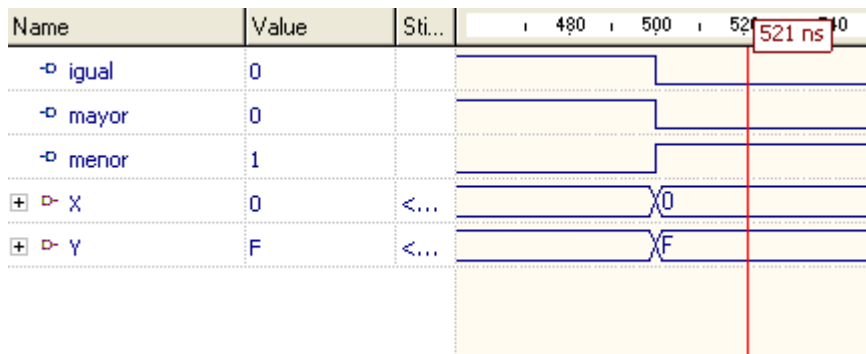


figura 9 simulación 5

Simulación con comparador completo del número

A diferencia de lo anterior se compara el valor completo del número y no bit por bit, de tal manera que la sentencia queda reducida a solo tener que comparar las variables con los símbolos de "<" y ">"

De igual manera siempre al entrar al programa y tener como valores iniciales 0 la señal de igual será 1

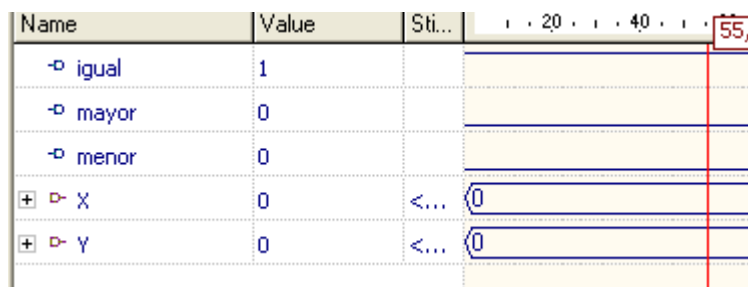


figura 10 simulación 6

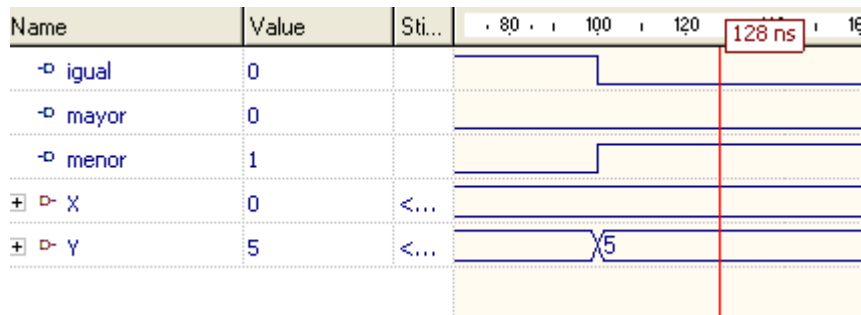


figura 11 simulación 7

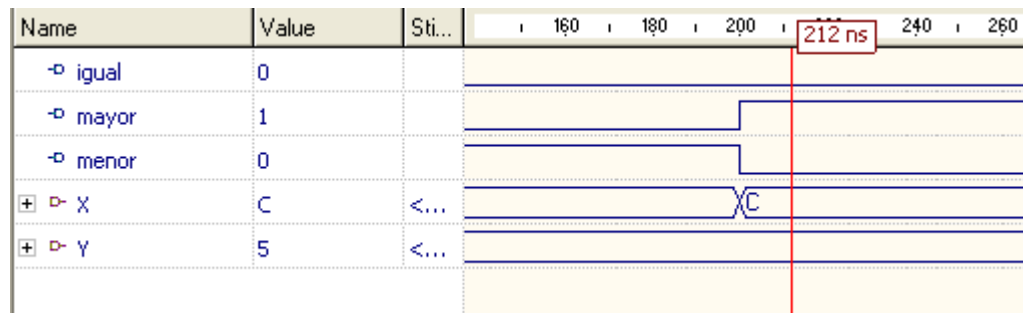


figura 12 simulación 8

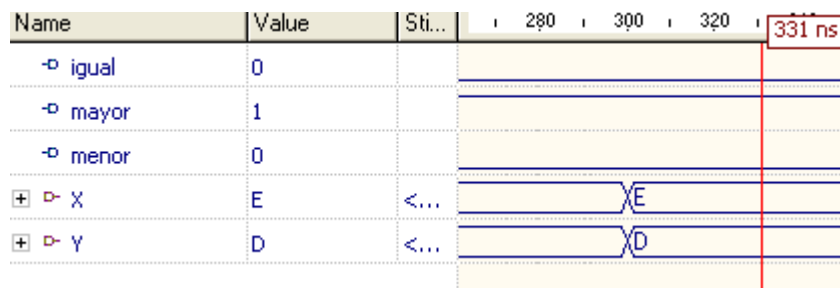


figura 13 simulación 9

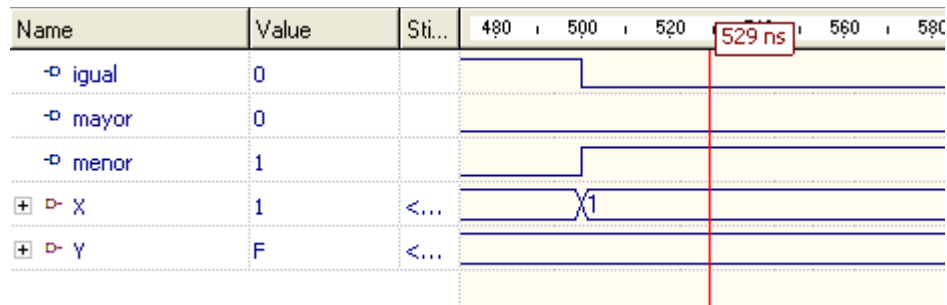


figura 14 simulación 10

Name	Value	Sti...	600	620	636,162
- igual	0				
- mayor	0				
- menor	1				
+ - X	5	<...		5	
+ - Y	D	<...		D	

Conclusiones

Con ayuda de la práctica se pudo comprender de mejor manera la implementación de los comparadores así como la forma de las que puedan trabajar ya sea compara completo el número o comparar bit a bit, siendo que la comparación bit a bit se debe tener más cuidado para indicar cuales son los bits que son mayores o menores respecto al otro número