ALEJANDRO REYES ALBILLAR 45931406-S

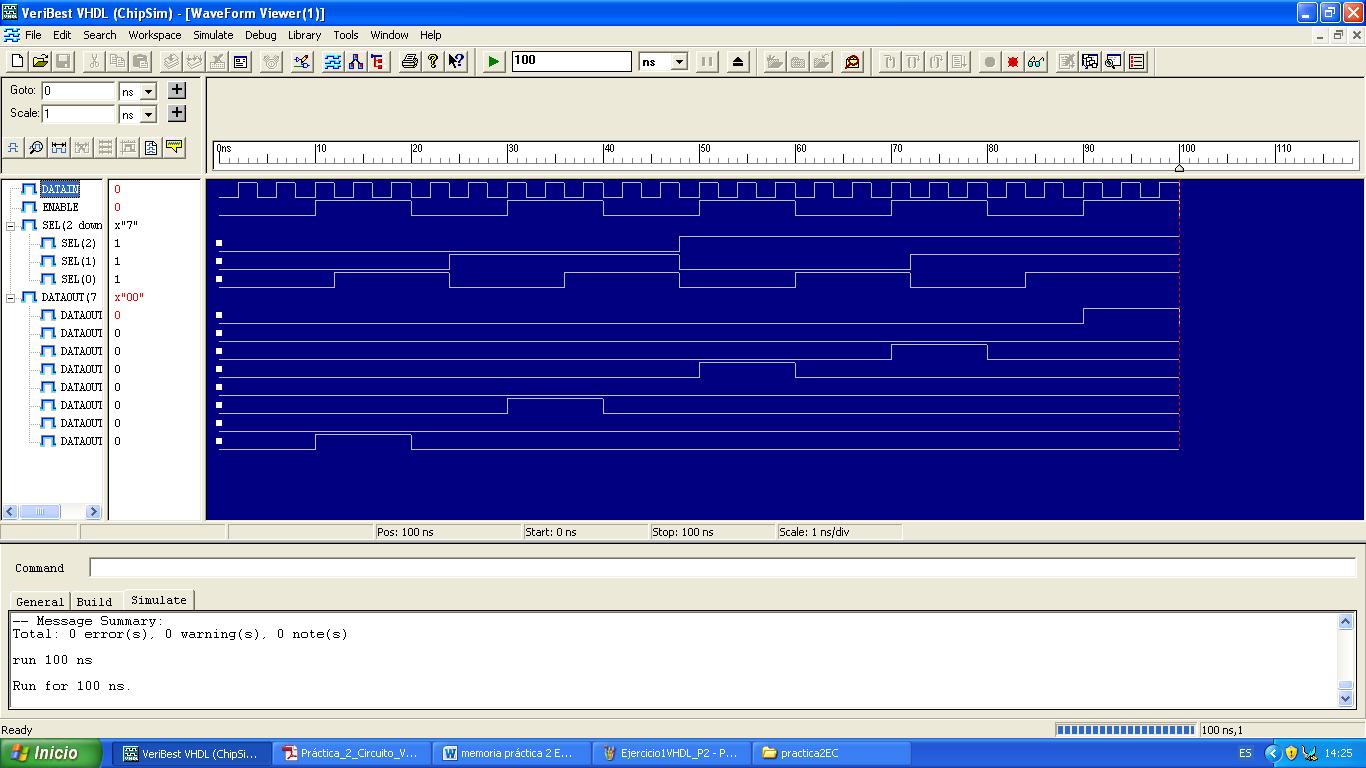
PRACTICA 2 ESTRUCTURA DE LOS COMPUTADORES

1º) En este primer ejercicio se nos pide que realicemos la implementación de un Demultiplexor de 1 a 8 bits con entrada de selección de 3 bits, siendo tanto la entrada de selección como la salida vectores con sus correspondientes tamaños.

Sabemos que la tabla de verdad de un biestable D es:

|  |  |  |  |
| --- | --- | --- | --- |
| Data(0/1) | Selección(3 bits) | Enable | Salida |
| D | --- | 0 | - |
| D | 000 | 1 | D |
| D | 001 | 1 | D |
| D | 010 | 1 | D |
| D | 011 | 1 | D |
| D | 100 | 1 | D |
| D | 101 | 1 | D |
| D | 110 | 1 | D |
| D | 111 | 1 | D |

Tras haber implementado la entidad, la arquitectura y el correspondiente testbech se ha hecho una simulación que ha dado como resultado la siguiente imagen:



Como podemos observar, cada uno de los valores de DataIN son copiados en la salida indicada en la selección Sel cuando Enable está activo a nivel alto, es decir, cuando su valor es 1.

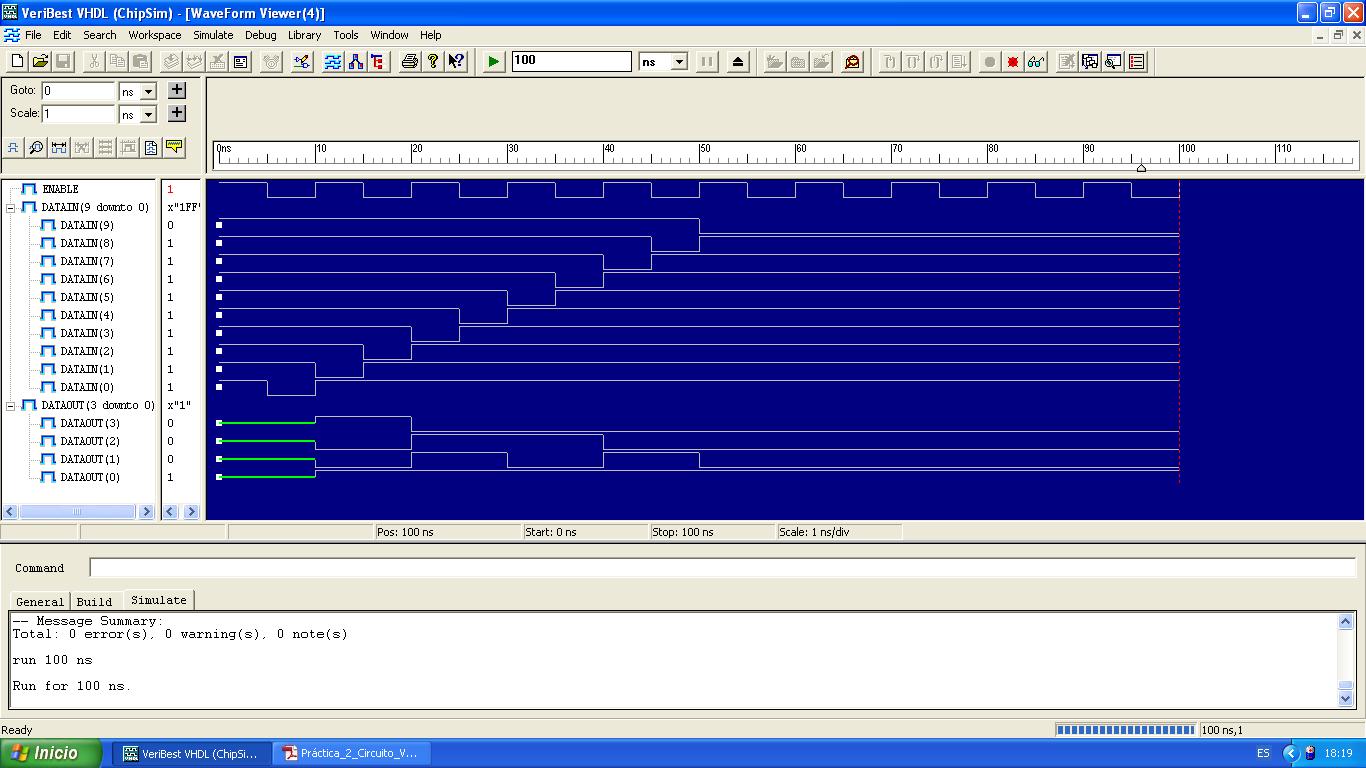
2º) En este segundo ejercicio, se nos pide que realicemos la entidad, estructura y testbech de un codificador de 10 a 4 entradas. Para ello, hacemos uso de la siguiente tabla de verdad:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ENA | A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | SALIDA |
| H | X | X | X | X | X | X | X | X | X | L | 0001 |
| H | X | X | X | X | X | X | X | X | L | H | 0010 |
| H | X | X | X | X | X | X | X | L | H | H | 0011 |
| H | X | X | X | X | X | X | L | H | H | H | 0100 |
| H | X | X | X | X | X | L | H | H | H | H | 0101 |
| H | X | X | X | X | L | H | H | H | H | H | 0110 |
| H | X | X | X | L | H | H | H | H | H | H | 0111 |
| H | X | X | L | H | H | H | H | H | H | H | 1000 |
| H | X | L | H | H | H | H | H | H | H | H | 1001 |
| H | L | H | H | H | H | H | H | H | H | H | 1010 |
| H | H | H | H | H | H | H | H | H | H | H | ZZZZ |
| L | X | X | X | X | X | X | X | X | X | X |  |

Siendo ENA la señal de enable, X cuando el valor es cualquiera, H cuando el valor es 1, L cuando el valor es 0 y Z cuando se encuentra en alta impedancia.

Como el ejercicio nos pide que realicemos la práctica para que la señal esté activada a nivel bajo, asignamos según la tabla de verdad unos valores a la salida de nuestro codificador, siendo el valor de alta impedancia Z cuando ninguna de las señales de entrada tome valor 0 mientras el enable está activo.

Se puede comprobar en la siguiente fotografía las entradas y la salida del circuito, siendo la zona verde fosforito la asignada a la alta impedancia Z.

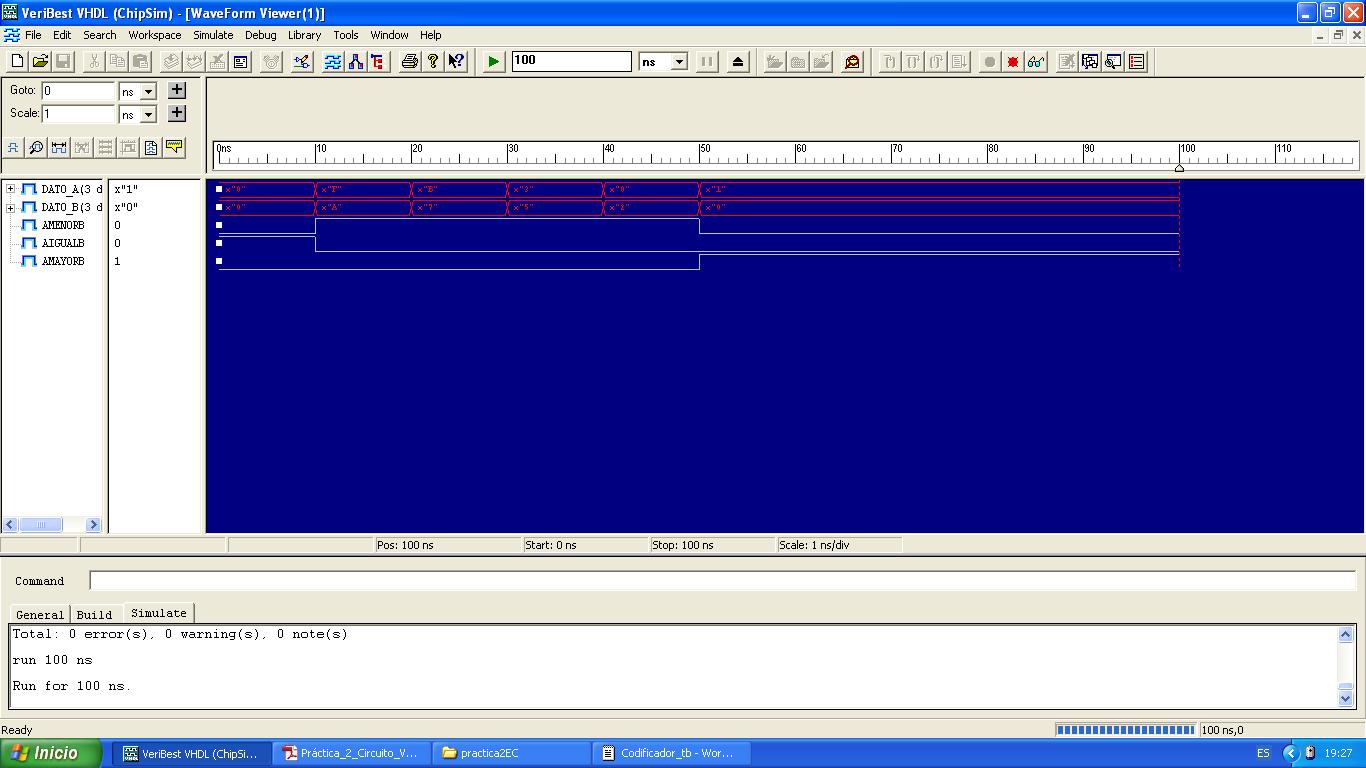


3º) Se nos pide que definamos la entidad, la arquitectura y un testbech para un comparador de 2 números de 4 bits en complemento a 2 según la siguiente tabla:

|  |  |
| --- | --- |
| COMPLEMENTO A 2 | DECIMAL |
| 0111 | 7 |
| 0110 | 6 |
| 0101 | 5 |
| 0100 | 4 |
| 0011 | 3 |
| 0010 | 2 |
| 0001 | 1 |
| 0000 | 0 |
| 1111 | -1 |
| 1110 | -2 |
| 1101 | -3 |
| 1100 | -4 |
| 1011 | -5 |
| 1010 | -6 |
| 1001 | -7 |
| 1000 | -8 |

Ésta tabla representa la conversión de decimal a binario complemento a 2 de los números de 4 bits existentes.

Cuando un número A es mayor que otro B la salida AmayorB queda con valor 1 mientras que el resto de salidas quedan a 0. Esto cambia según sea A mayor, menor o igual que B, dándole el valor correspondiente a cada salida como se puede comprobar en la siguiente ejecución del testbech:



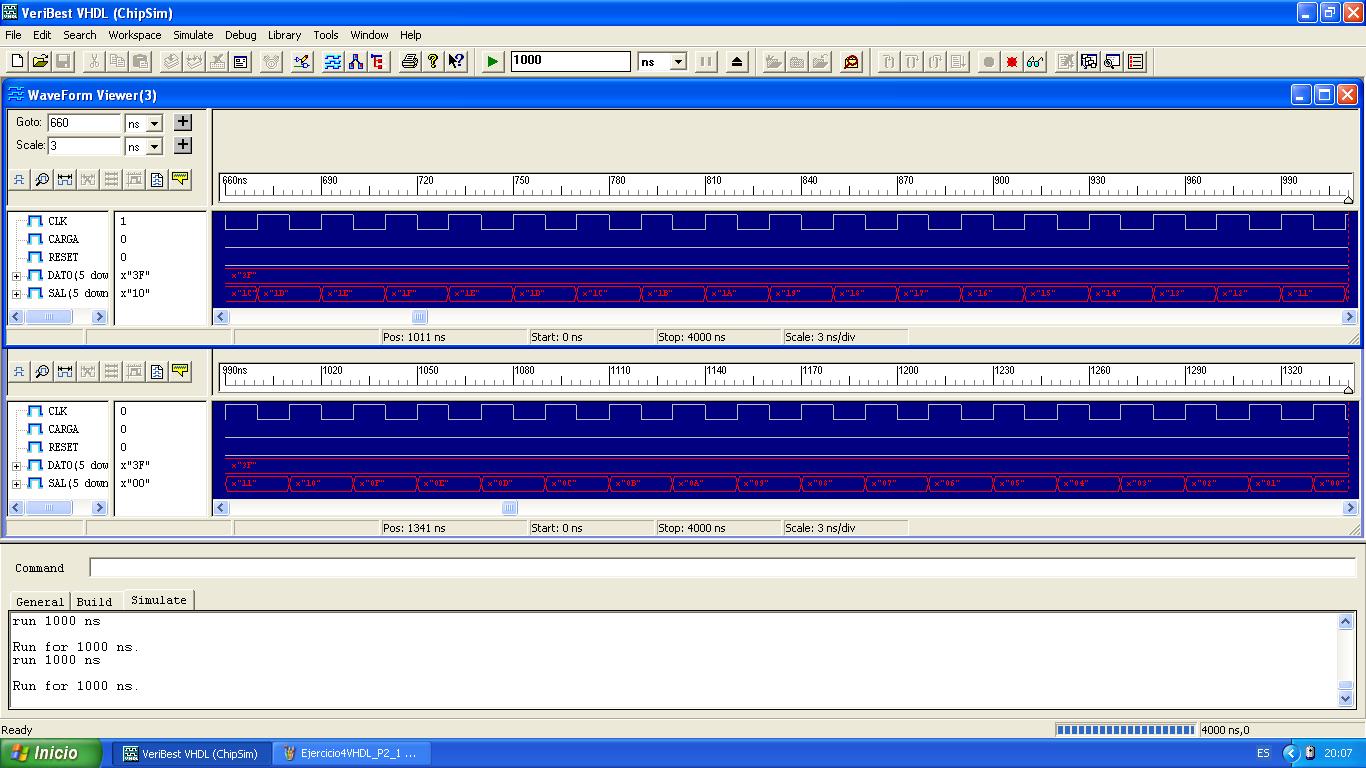
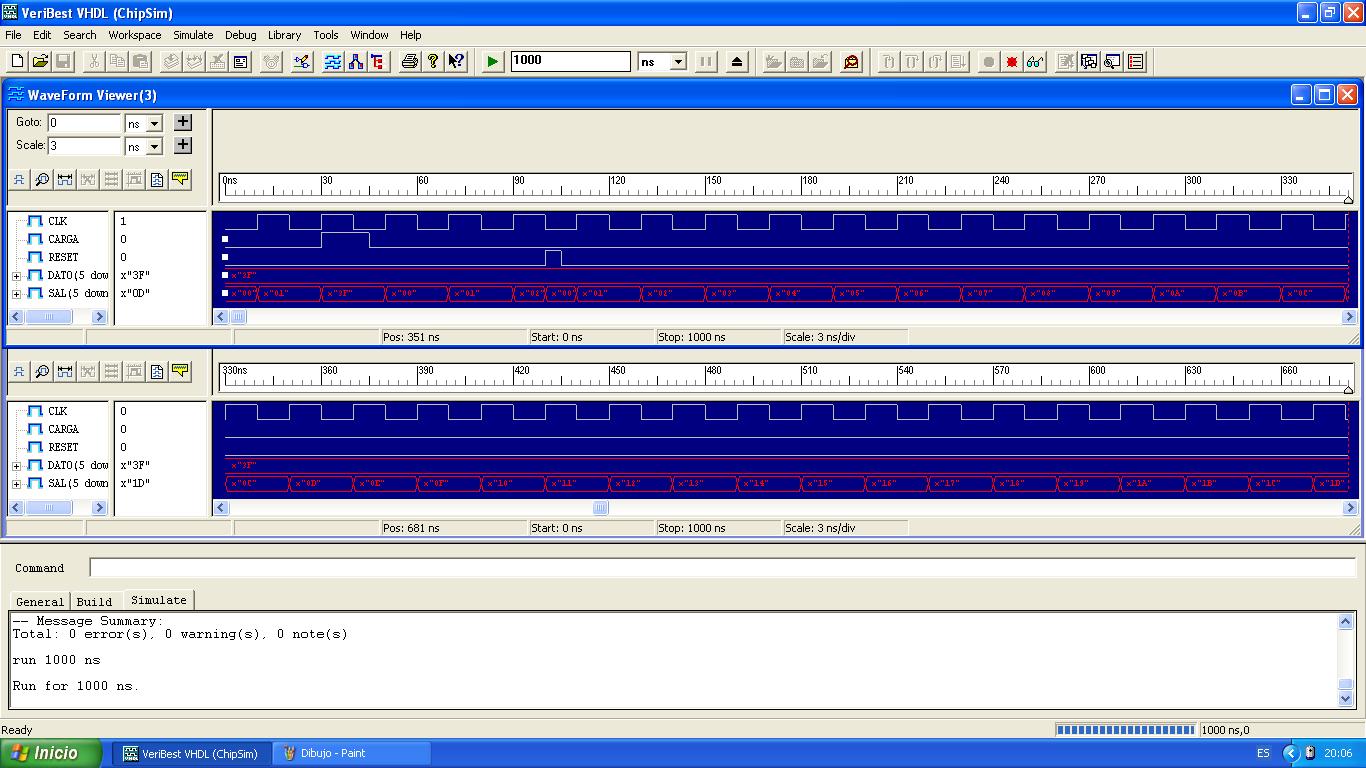
Las entradas son, enumerándolas de arriba abajo Dato\_A y Dato\_B, y las salidas enumeradas de la misma manera serían AmenorB, AigualB y AmayorB.

Las entradas dadas a los datos A y B expresadas en hexadecimal quedan explicitas en la siguiente tabla:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DATO\_A | 0 | F | B | 3 | 0 | 1 |
| DATO\_B | 0 | A | 7 | 5 | 2 | 0 |

4º) Se nos pide realizar un contador ascendente descendente que tenga una entrada de datos que cuando recibe la entrada síncrona carga copie la entrada datos a la salida, también que cuando reciba la entrada asíncrona reset ponga la salida a 0 y que aumente cuando se produce un evento en la señal de reloj Clk.

En las siguientes imágenes podemos ver la ejecución de cómo funciona el contador ascendente descendente.



Podemos ver en la primera imagen que se realiza correctamente la asignación de Dato a salida cuando carga es 1 mientras la señal de reloj está activa, sin embargo, debido a que el simulador de veribest no funciona correctamente solamente recibe la señal hasta el valor “1F” expresado en hexadecimal, es por eso que la simulación mostrada en la segunda imagen es hasta dicho valor y es por eso que el siguiente valor a “3F” tras la asignación es “00”.

**La arquitectura de este ejercicio es la que tiene como límite de cambio 3F, no con los valores usados para realizar la simulación mostrada en las imágenes.**

Todas las imágenes de este documento, las entidades, arquitecturas, testbech y entornos de trabajo de los ejercicios están dentro del zip que contiene a este documento.