INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

DEPARTAMENTO DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

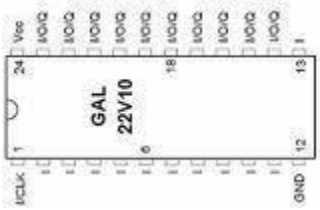
PRACTICA No. 4

USO DE GALAXY

Nombre: Silva Hernandez Noe Jasiel

Materia: Fundamentos de Diseño Digital

Grupo: 2CV1



USO DE GALAXY

I OBJETIVO GENERAL:

Al finalizar la práctica, el alumno aprenderá a programar un dispositivo lógico programable (PLD). Para esto estudiará y aprenderá a utilizar las herramientas adecuadas para dicho objetivo. Así mismo, reafirmará el conocimiento adquirido en clase al realizar un programa que ejecute una función específica, sobre el dispositivo utilizado, y comprobar físicamente el correcto funcionamiento del programa desarrollado. Finalmente, sabrá lo que es un PLD y como programar una función específica sobre el dispositivo, utilizando un lenguaje de descripción de hardware (HDL).

II OBJETIVOS ESPECÍFICOS:

I. Distinguir los beneficios proporcionados por las herramientas CAD.
II. Confirmar el conocimiento adquirido en clase y realizar un programa en un HDL que se ejecute sobre un PLD.
III. Comprobar físicamente el correcto funcionamiento del sistema diseñado, el cual será implementando sobre un proto-board.

III MATERIAL Y EQUIPO EMPLEADO.

- Fuente de 5V.
- Circuito Integrado GAL22V10.
- DIP switch de 8.
- 6 Resistencias de 1KΩ.
- 1 Resistencia de 330Ω.
- 1 Led.
- Tablilla de Prueba (proto-board).
- Alambre telefónico.
- Pinzas de punta.
- Pinzas de corte.
- Cables Banana-Caimán.
- Programador universal.

IV DESARROLLO EXPERIMENTAL.

El problema propuesto es simple, ya que la finalidad es relacionarse con las herramientas CAD, comprender la mecánica del proceso de diseño utilizando PLD's y realizar la implementación física de la función deseada.

De esta manera, la práctica consiste en programar el código para implementar las ya conocidas compuertas lógicas en la GAL, usando VHDL. El circuito tendrá 4 entradas y 1 salida, las cuales corresponden a las compuertas lógicas; y debe mostrar a la salida el resultado de las operaciones lógicas ya conocidas.

V EDICIÓN DEL PROGRAMA.

Los pasos a seguir para trabajar con Galaxy se describen a continuación:

USO DE GALAXY

1. Ejecutar el software Galaxy (**Inicio→ Todos los programas→ Cypress→ Galaxy**).

2. Una vez en la interfaz del programa, se crea un proyecto de la siguiente manera:

a) **File→ New→ Project [Target-Device].** Esto abre una ventana, como se muestra en la figura 1, en donde se introduce la ruta en que se salvará el proyecto, el nombre de este (**compuertas para este caso**) y además se selecciona el lenguaje que se va a utilizar (VHDL o Verilog HDL). Se selecciona VHDL.

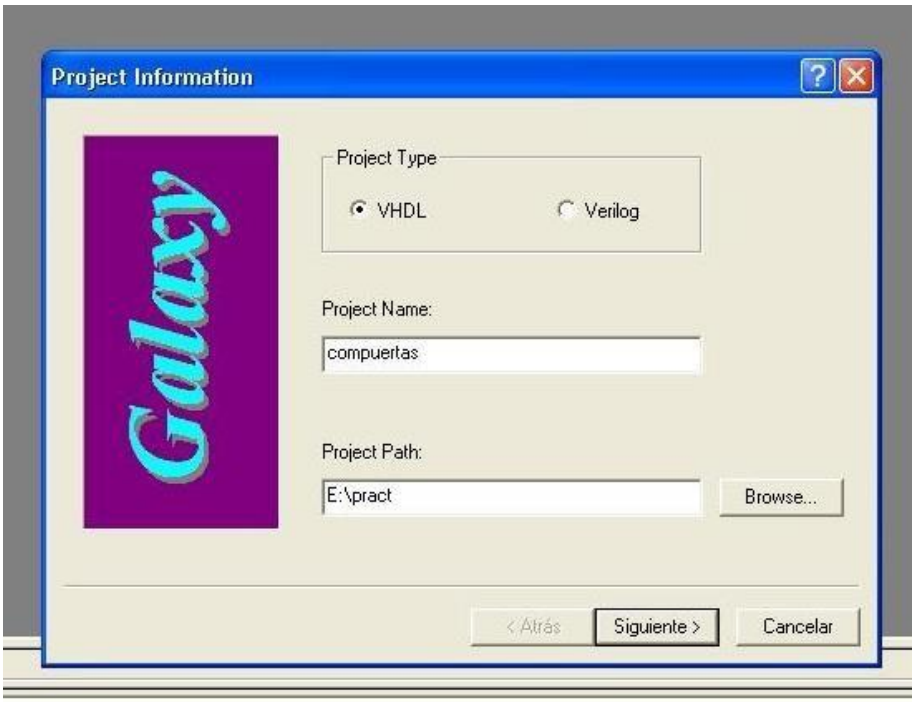


Figura 1. Nombre del proyecto (*compuertas*) y ruta.

b) Se da clic en **<siguiente>** y aparece una nueva ventana que da la opción para agregar el archivo ***.vhd***, lo cual haremos después, por lo que se da clic en **<siguiente>**, lo que abre una nueva ventana que permite seleccionar el dispositivo a utilizar (ver figura 2). En esta práctica se utilizará un dispositivo del tipo GAL22V10, que es un **SPLD**, localizado dentro de **C22V10** y se selecciona el PALCE22V10-25PC ó -15PC, según sea su retardo de programación (**speed (ns)**) y se da clic en **<Finalizar>**.

c) Después aparece una ventana que pregunta si se quiere salvar el nuevo proyecto, y se selecciona **<si>.**

3. Para crear un archivo de texto, se hace mediante: **File→ New→ Text File** y después **<OK>**.

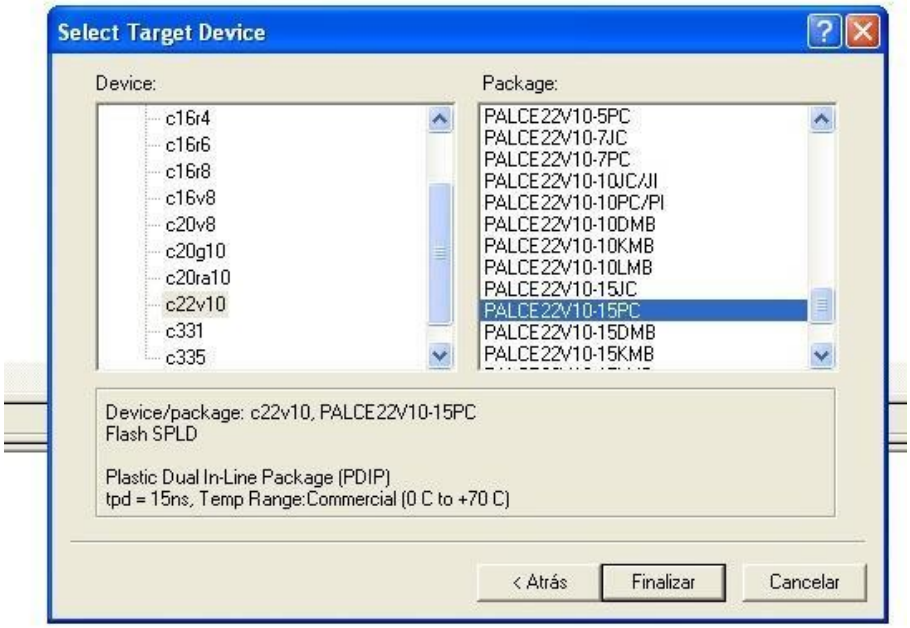


Figura 2. Selección del tipo de dispositivo.

4. Se escribe el código que se muestra en la figura 3 y después se salva el archivo, preferentemente con el nombre de la ENTIDAD, con terminación ***.vhd** (en este caso se llama **and.*vhd***), para posteriormente agregarlo al proyecto haciendo clic en **Project → Add Files**, como se muestra en la figura 4.

```
1 LIBRARY ieee;
2 USE ieee.std_logic_1164.all;
3
4 ENTITY compuerta_and IS
5     PORT( a, b, c, d: in std_logic;
6           salida : out std_logic
7           );
8 end compuerta_and;
9
10 architecture funcion of compuerta_and is
11 begin
12
13     salida <= a and b and c and d;
14
15     end compuerta_and;
16
```

Figura 3. Código del programa en VHDL.

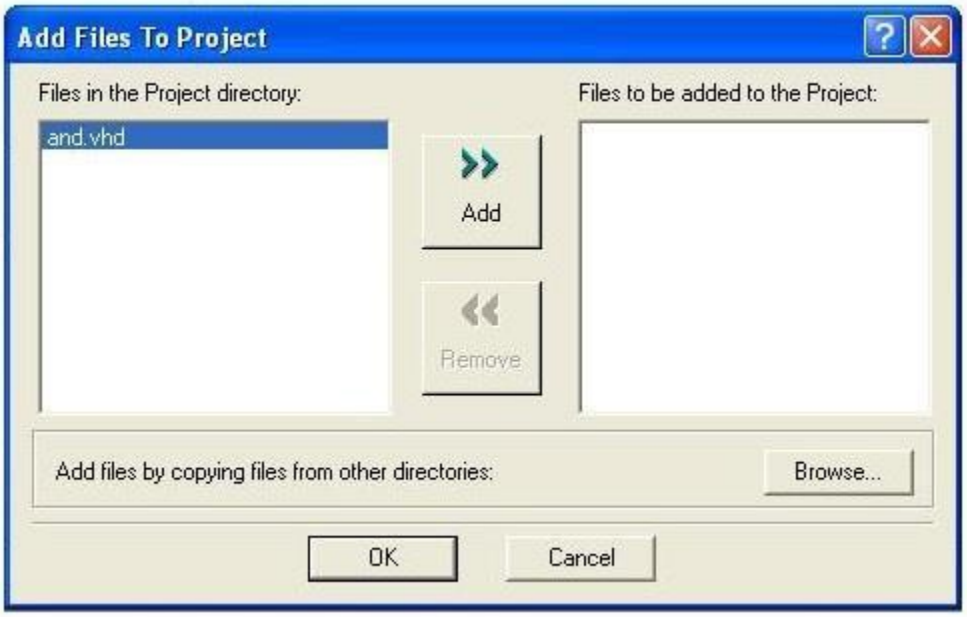


Figura 4. Agregar el archivo .vhd al proyecto.

5. En la figura 5 se puede observar que la ventana izquierda cambia. Esto es debido al hecho de que ya es un proyecto. Se da clic con el botón derecho del ratón sobre el icono del archivo creado **and.vhd** y se selecciona **Set Top**. Se puede observar, en la figura 5b, que después de esto cambia dicho icono.

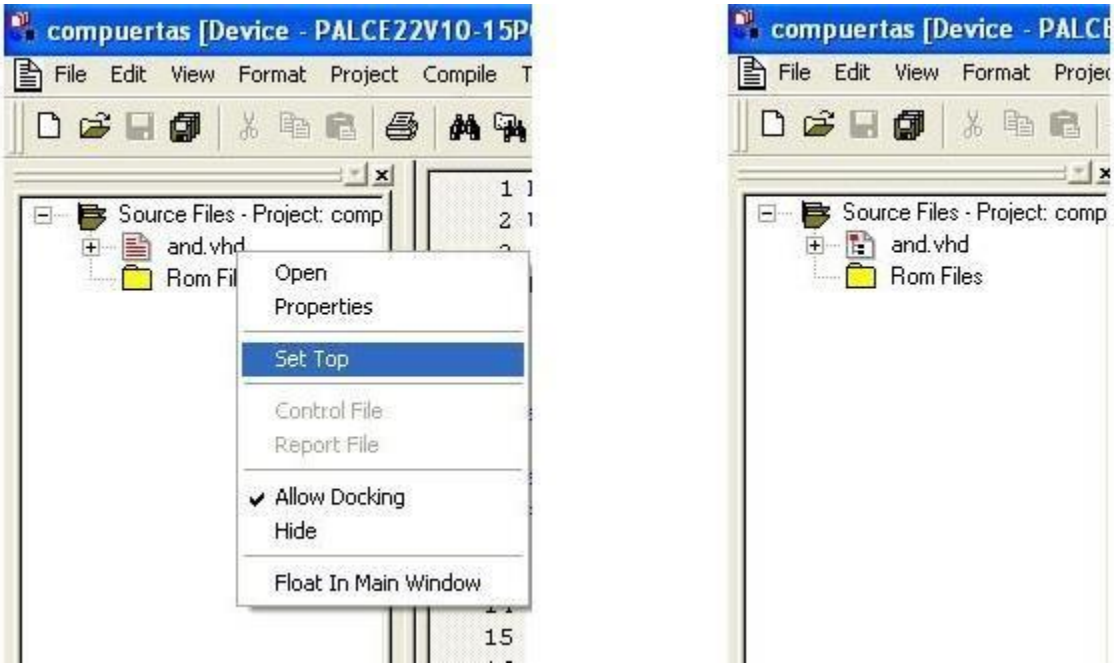


Figura 5 a) y b).

6. Una vez hecho esto, se compila el archivo en: **Compile→ Selected File(s)**. En la parte inferior de la ventana se despliega la información de la compilación. En caso de que esta no fuera satisfactoria, es posible ver los errores o advertencias. Para esto es necesario dirigirse a la pestaña de nombre **Errors & Warnings** que está localizada en la parte inferior de la ventana de trabajo.

7. Para corroborar el funcionamiento correcto del programa es posible realizar su simulación. Para esto, debemos ir a: **Tool→ Active-HDL Sim**, abriéndose la ventana que se muestra en la figura 6.

8. Se abre el archivo **and.*vhd***, esto en: **File→ Open VHDL file for simulation** y se selecciona el archivo, localizado en la carpeta ***vhd*, *creada por el proyecto***. Se da <**Abrir**>. Al hacer eso se compila el programa, apareciendo comentarios en la consola de la ventana de simulación.

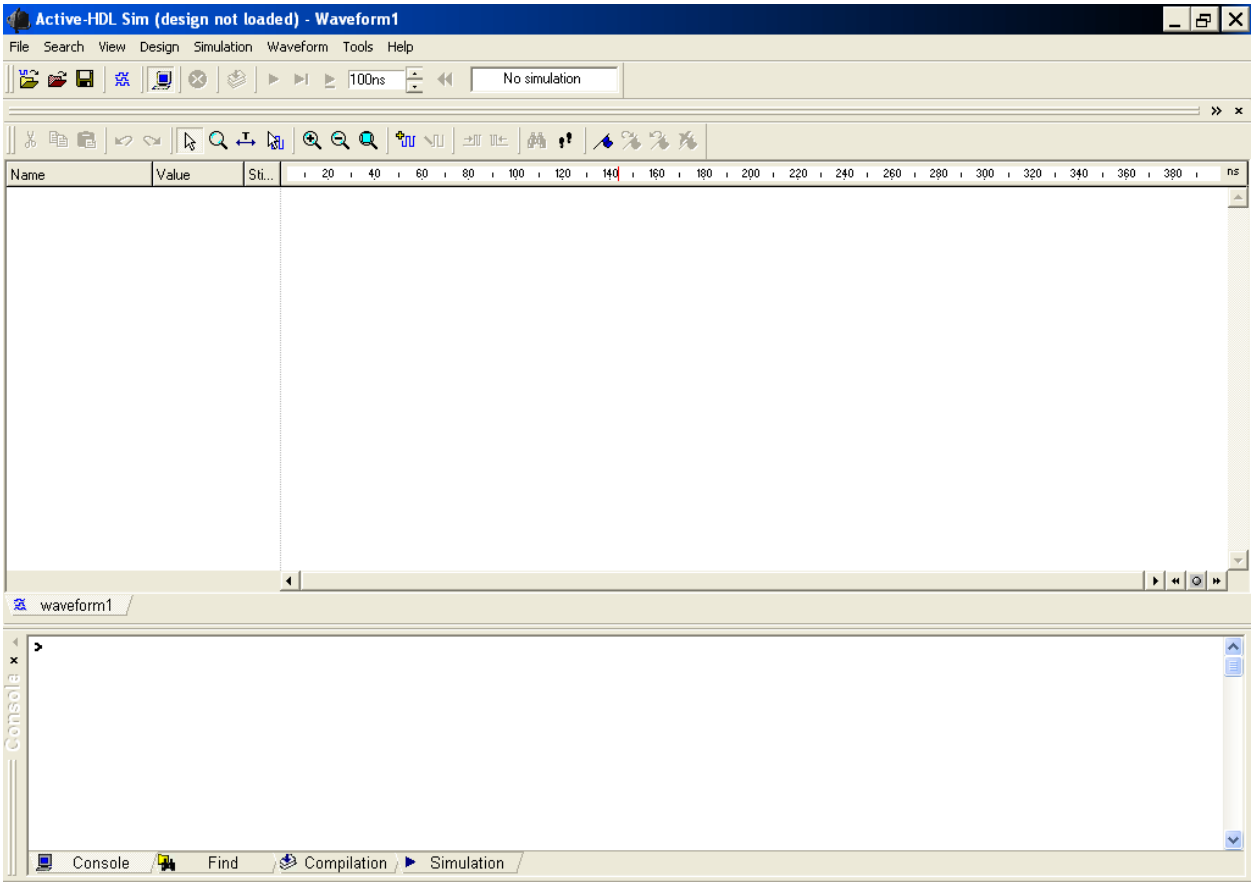


Figura 6. Interfaz de simulación.

9. Para visualizar las señales a simular es necesario agregarlas a la ventana de simulación. Para esto seleccionamos **Waveform→ Add signal**, como se muestra en la figura 7a. Al hacer esto, se abre la ventana que se muestra en la figura 7b, donde se seleccionan las señales a monitorear; en este caso, las entradas a, b, c, d y la salida.

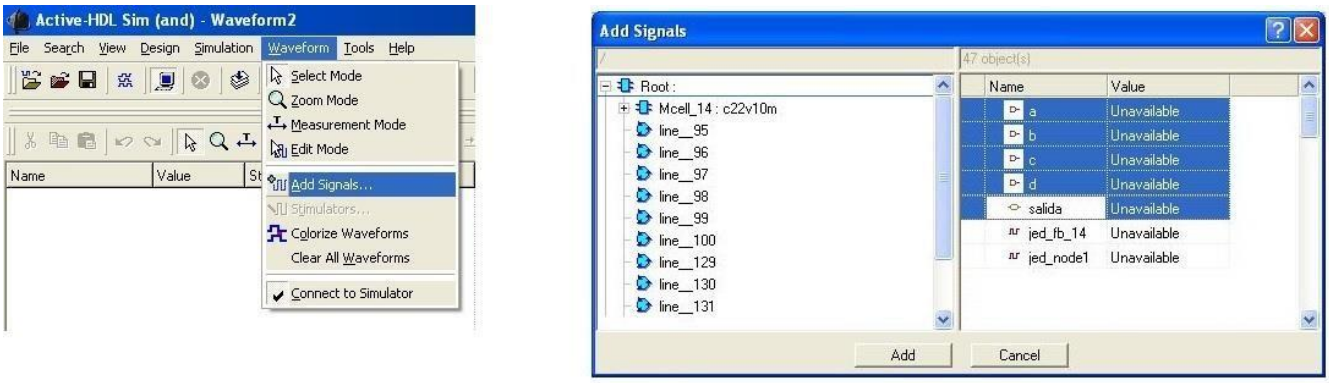


Figura 7 a) y b). Agregar señales a la simulación.

10. En el caso de las entradas es necesario asignar valores para la simulación. Esto se hace dando clic con el botón derecho sobre las señales. Al hacer esto se abre un menú desplegable y se da clic en **Stimulators**, abriéndose una nueva ventana, que se muestra en la figura 8.

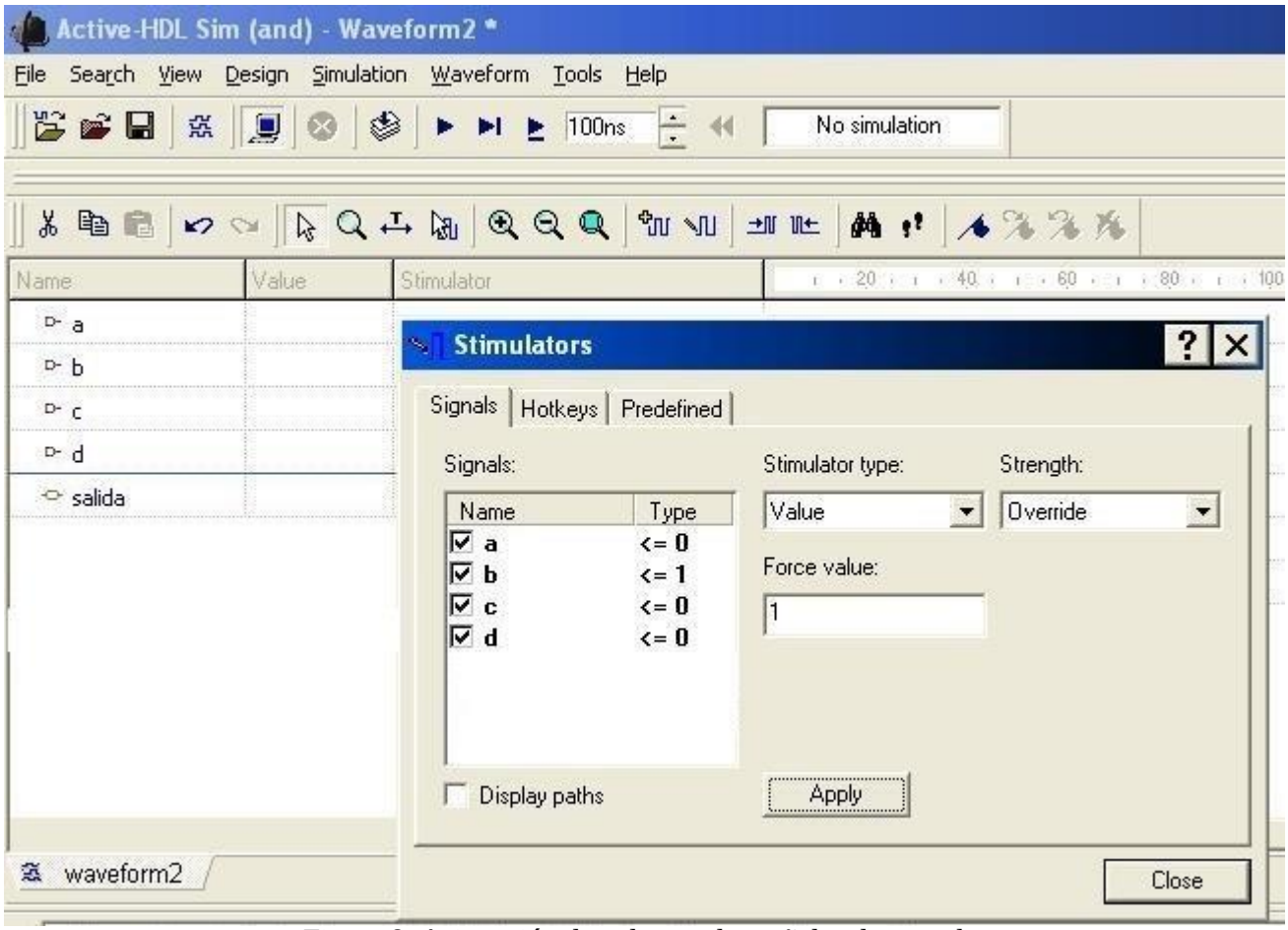


Figura 8. Asignación de valores a las señales de entrada.

11. En esta ventana se indican el tipo de estímulo, en este caso elegimos **Stimulator type: value, Strength: Override, Force value:** (valor que deseamos asignar, el cual será **0's** o **1's** para las entradas a, b, c, y d). Para que tenga efecto se da clic en <**Apply**>.

12. Para observar la simulación con los estímulos asignados es necesario correr el tiempo de la simulación. Esto se hace en: **Simulation→ Run for**. Al hacer esto se muestran los valores de entrada y se generan los valores de la salida, como se puede ver en la figura 9. Se verifica si los valores obtenidos son los deseados.

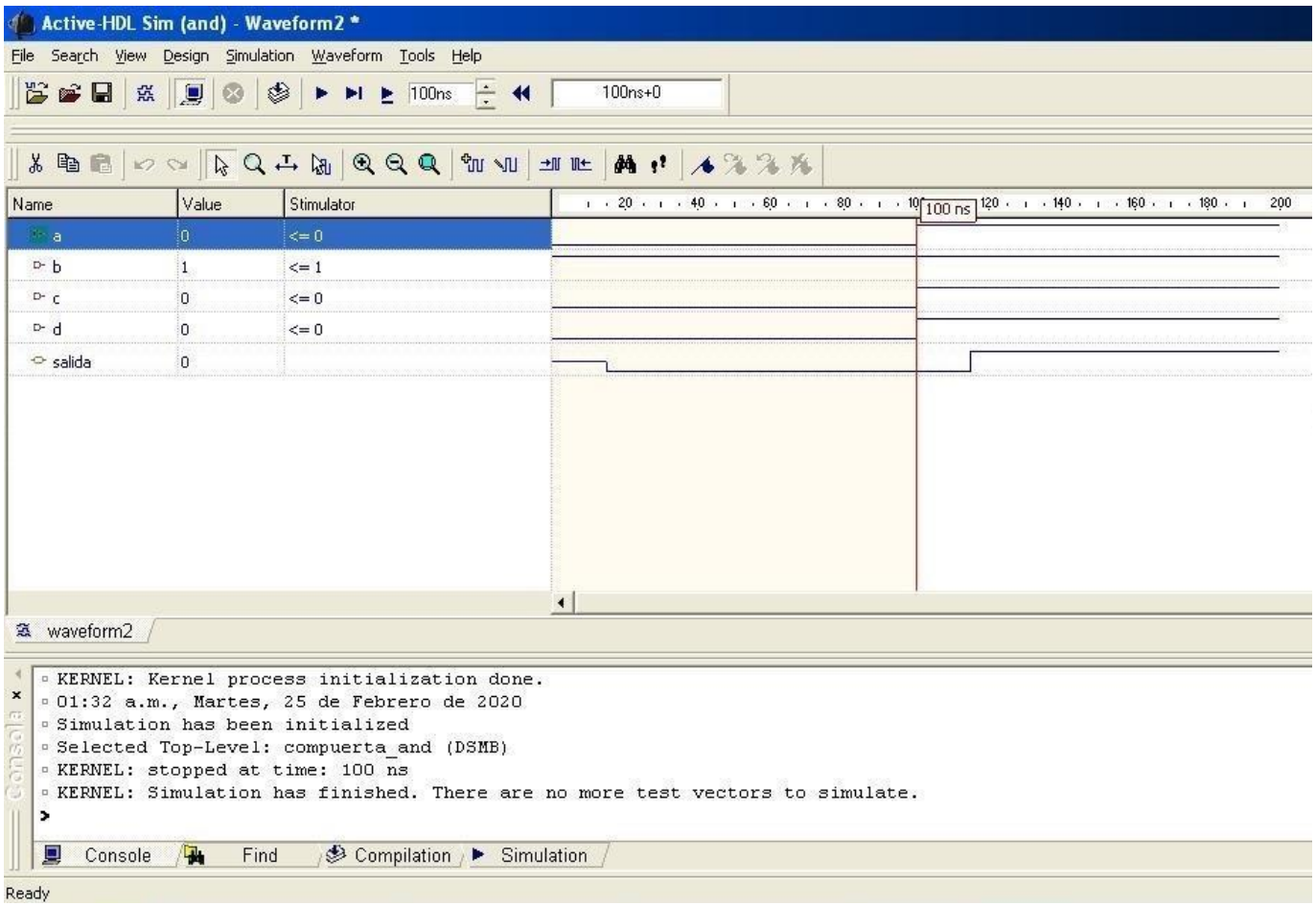


Figura 9. Simulación.

13. Si se desea hacer una nueva simulación con otro valor u otro estimulo de entrada, es necesario asignar el nuevo valor y correr otra vez la simulación.

14. Una vez que se ha revisado el correcto funcionamiento del sistema diseñado mediante la simulación, es necesario saber en qué terminales se encuentra cada señal de entrada/salida. El sitio donde aparece esta información es el reporte de salida ***Output Files***, localizado en la ventana izquierda del proyecto, dando clic derecho en el archivo **and.*vhd***. Para acceder al archivo de reporte se tiene que ir a la pestaña localizada en la parte inferior de la misma ventana, como se muestra en la figura 10. Dando clic en el archivo **and.*rpt*** se pueden observar las terminales asignadas a cada señal y el uso del dispositivo, como muestra la figura 11.

USO DE GALAXY

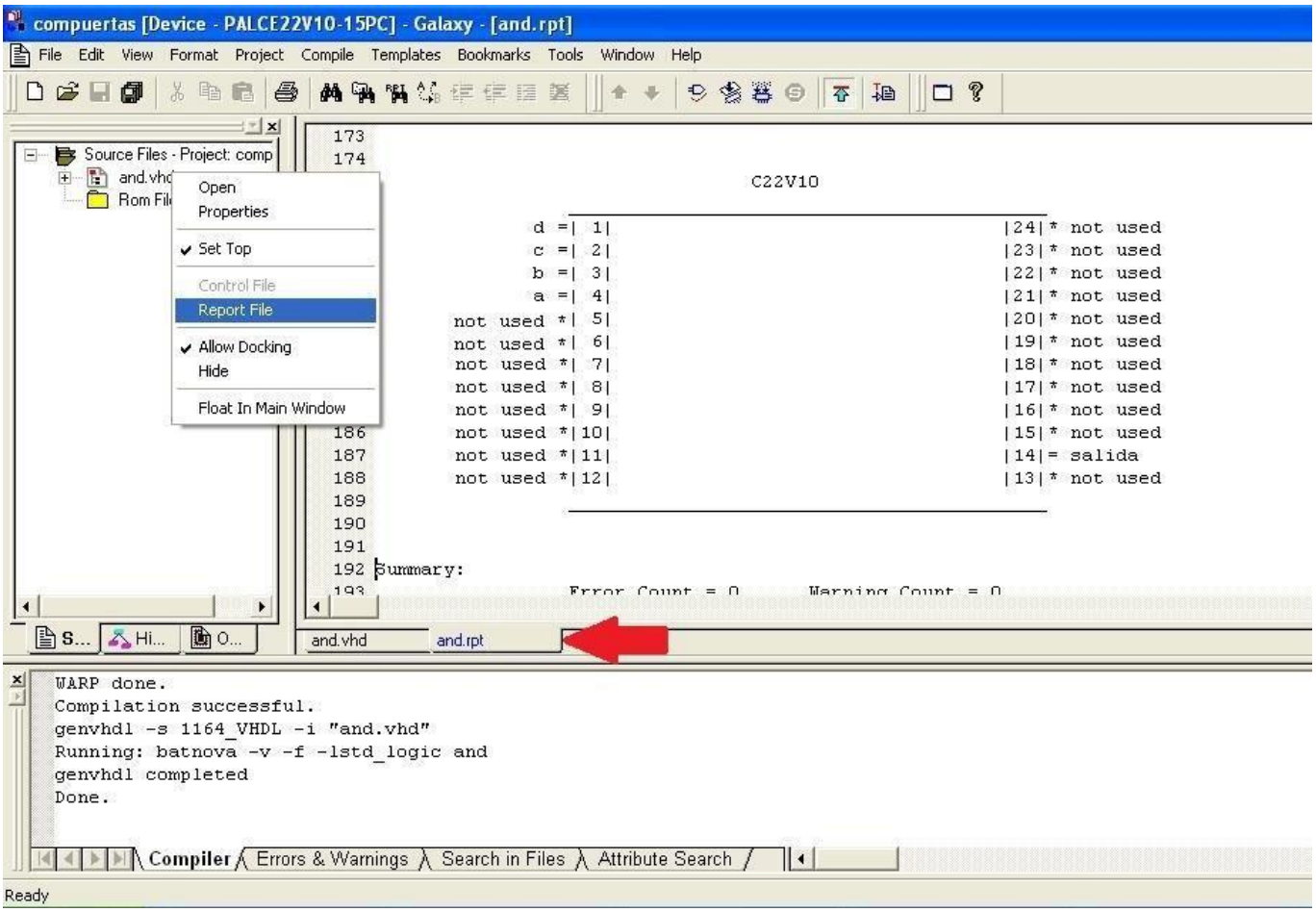


Figura 10. Reporte del diseño.

```
<CYPRESSTAG name="Utilization" icon=FILE_RPT_UTILIZATION>
RESOURCE UTILIZATION (01:20:50)
</CYPRESSTAG>
  Information: Macrocell Utilization.

                    Description        Used     Max
                 ____________________________________
                 | Dedicated Inputs   |    5  |   11  |
                 | Clock/Inputs       |    1  |    1  |
                 | I/O Macrocells     |    1  |   10  |
                 ____________________________________
                                       7  /   22   = 31  %
```

```
Information: Output Logic Product Term Utilization.

                 Node#  Output Signal Name  Used   Max
                 ____________________________________
                 | 14  |  salida          |   1  |   8  |
                 | 15  |  Unused          |   0  |  10  |
                 | 16  |  Unused          |   0  |  12  |
                 | 17  |  Unused          |   0  |  14  |
                 | 18  |  Unused          |   0  |  16  |
                 | 19  |  Unused          |   0  |  16  |
                 | 20  |  Unused          |   0  |  14  |
                 | 21  |  Unused          |   0  |  12  |
                 | 22  |  Unused          |   0  |  10  |
                 | 23  |  Unused          |   0  |   8  |
                 | 25  |  Unused          |   0  |   1  |
                 ____________________________________
                                           1  / 121   = 0   %
```

```
Completed Successfully
-----------------------------------------------------------------------------
PLD Compiler Software:        PLA2JED.EXE    31/03/2000  [v4.02 ] 6.3 IR 35

<CYPRESSTAG name="Equations" icon=FILE_RPT_EQUATION>
DESIGN EQUATIONS            (01:20:50)
</CYPRESSTAG>

    salida =
          a * b * c * d
```

Figura 11.

USO DE GALAXY

Desarrollo General

Ocuparemos el software de Galaxy solo disponible para S.O con XP
Empezaremos con creando un nuevo proyecto y seleccionaremos el CI GAL22v10 y seleccionamos una de sus variantes a convenir.
Después creamos un archivo de texto en el cual ira nuestro código…
Donde en la primera línea Definimos la librerías a utilizar… posteriormente utilizamos la palabra reservada USE para definir sobre que protocolo trabajaremos.

Entity : se define como la abstracción de un circuito esta definirá la forma externa del circuito, aquí se declaran las entradas y salidas del circuito
-define externamente el circuito
-nombre y numero de puertos
Las salidas y entradas se definen in, out respectivamente

ARCHITECTURE: aquí se describe la funcionalidad del circuito, en otras palabras conlleva la lógica de ese circuito

Y para terminar tras finalizar cada ENTITY y cada ARCHITECTURE se procede a finalizarla

Después de esto se salva el proyecto con extensión .vhd, agregamos un nuevo archivo desde add file y seleccionamos el archivo recién creado y se procede a compilarlo

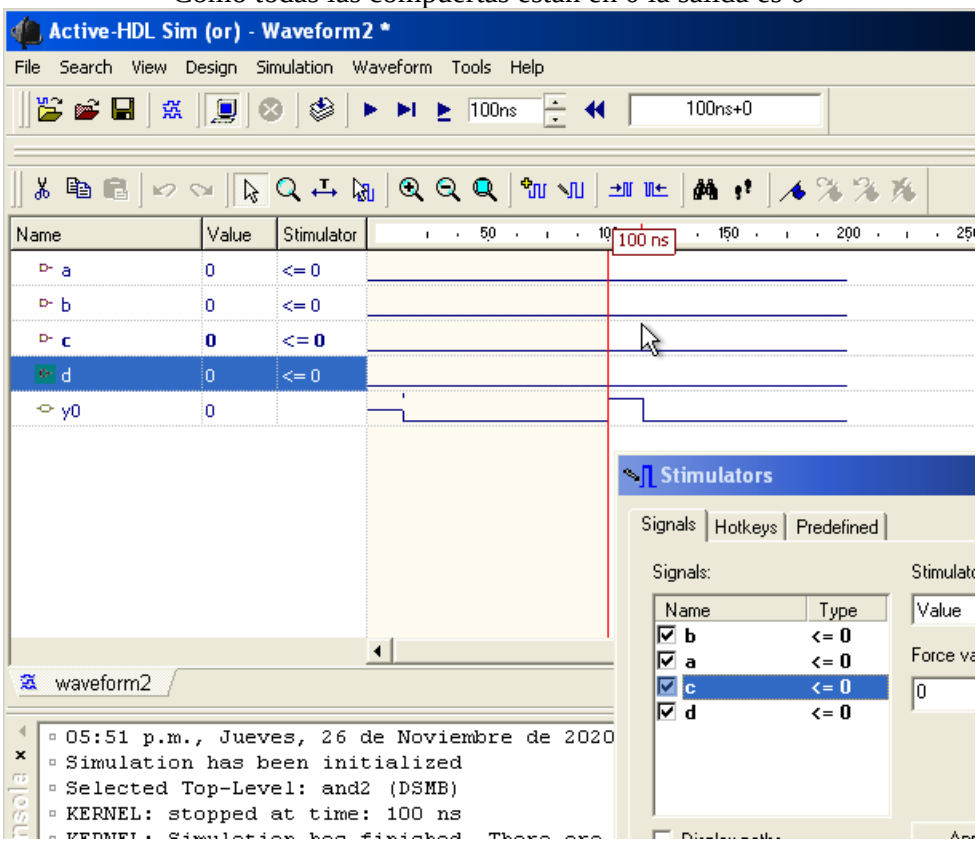
Despues del correcto funcionamiento se abre la pestaña **Active-HDL Sim** que se encuentra en tools en este programa se agregan las entradas generadas además de las salidas y se procede a darles un valor a las entradas… corremos el programa y anotamos la salida en la tabla de verdad

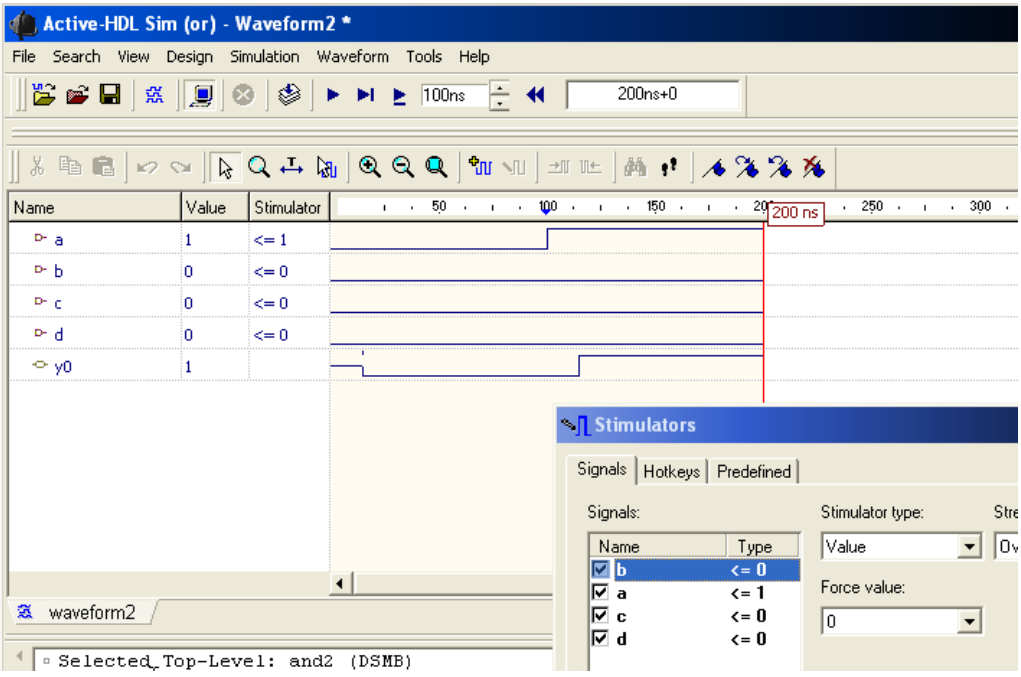
Este proceso se hace con todas las compuertas solo varea en el código.

USO DE GALAXY

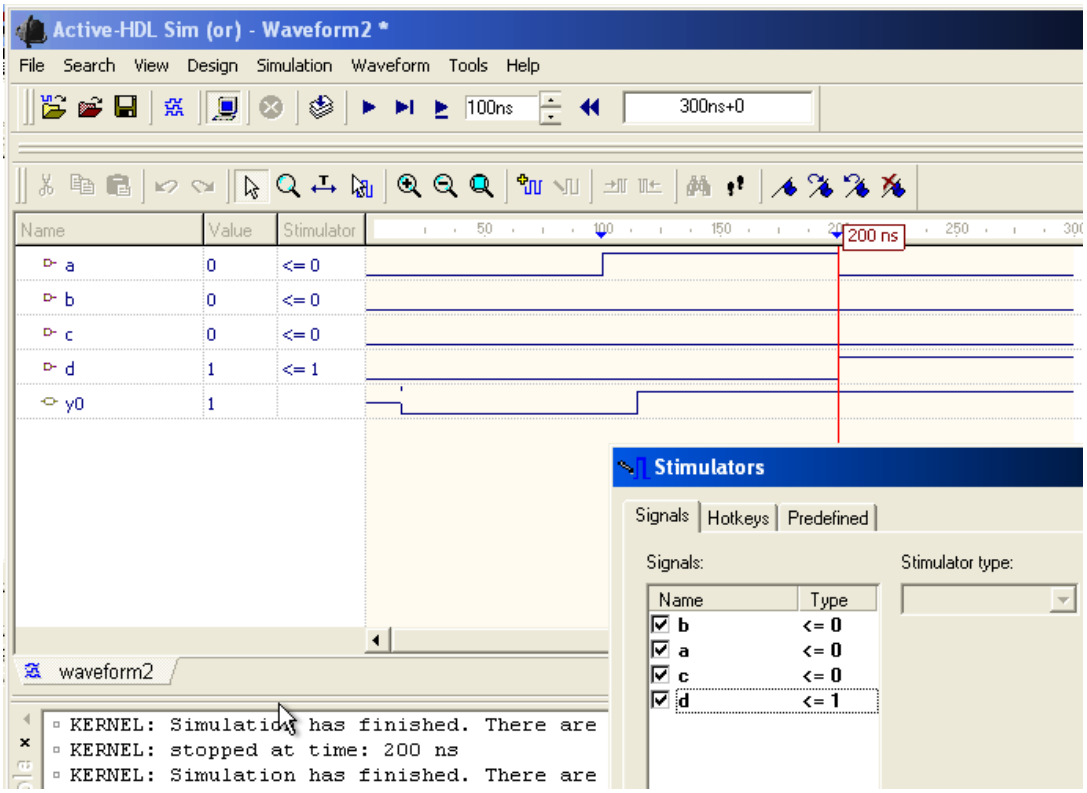
Capturas de la Compuerta OR

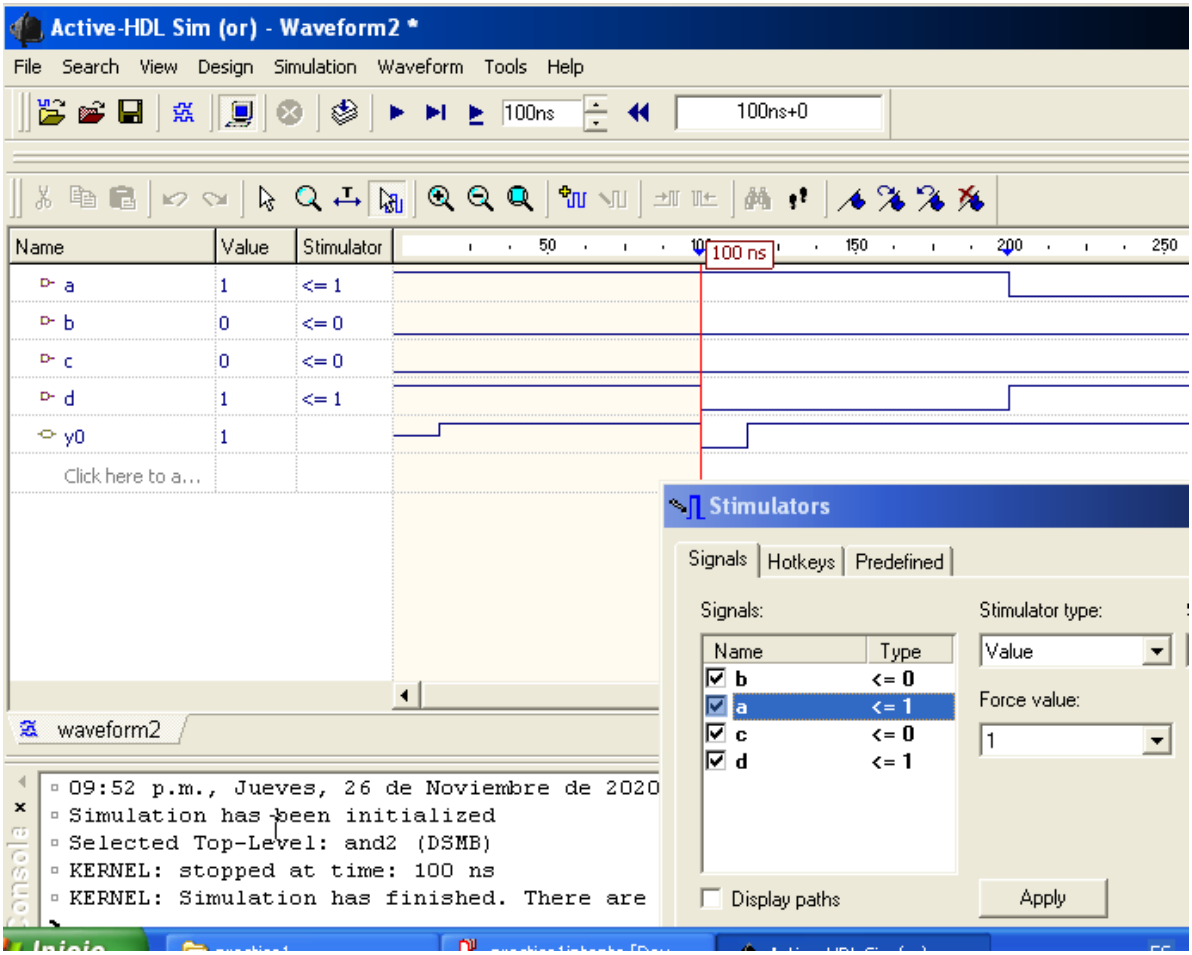
Como todas las compuertas están en 0 la salida es 0



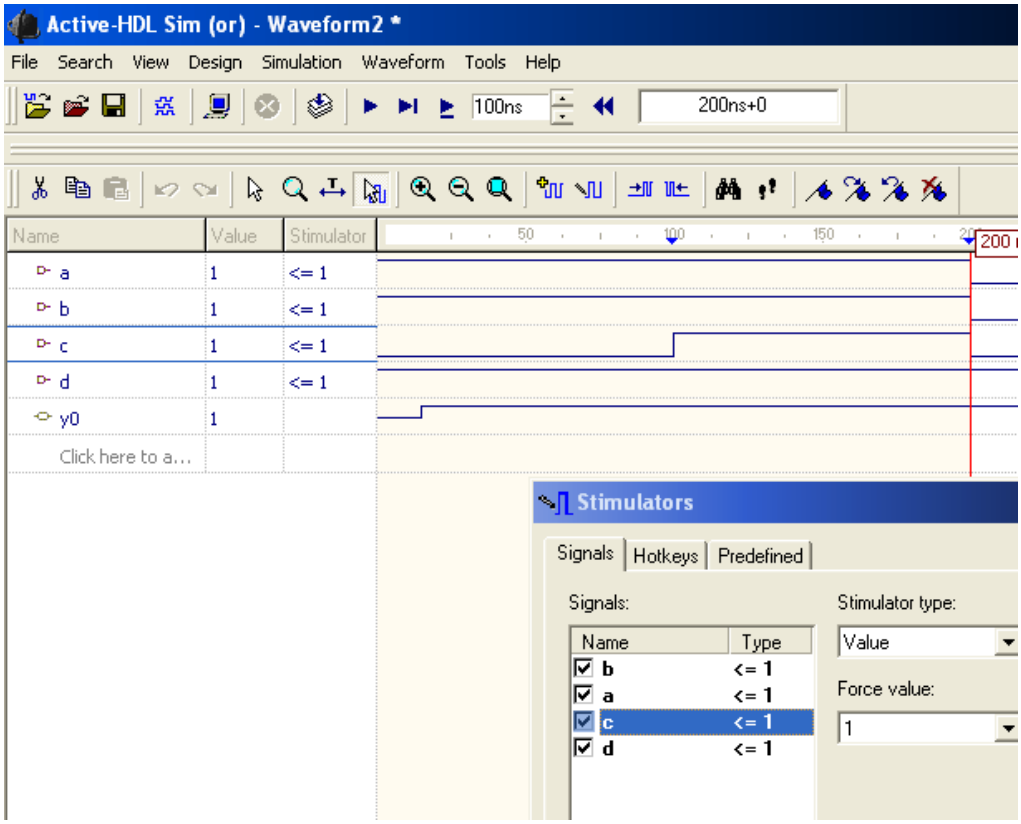


USO DE GALAXY



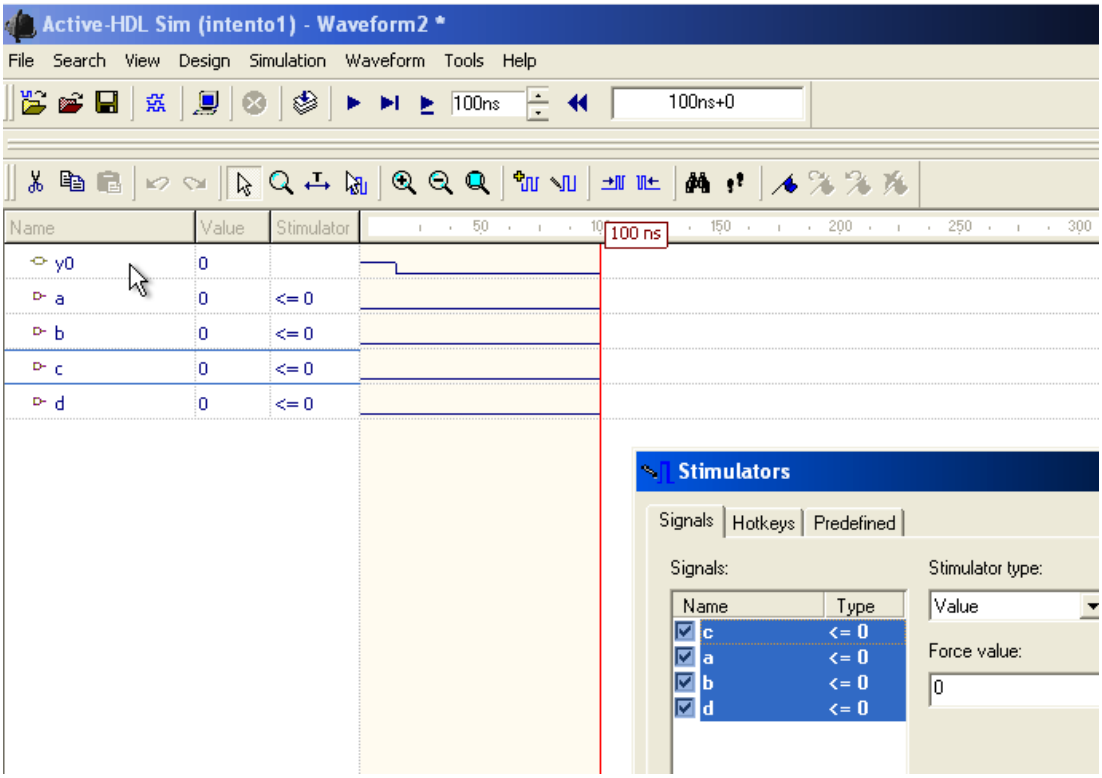


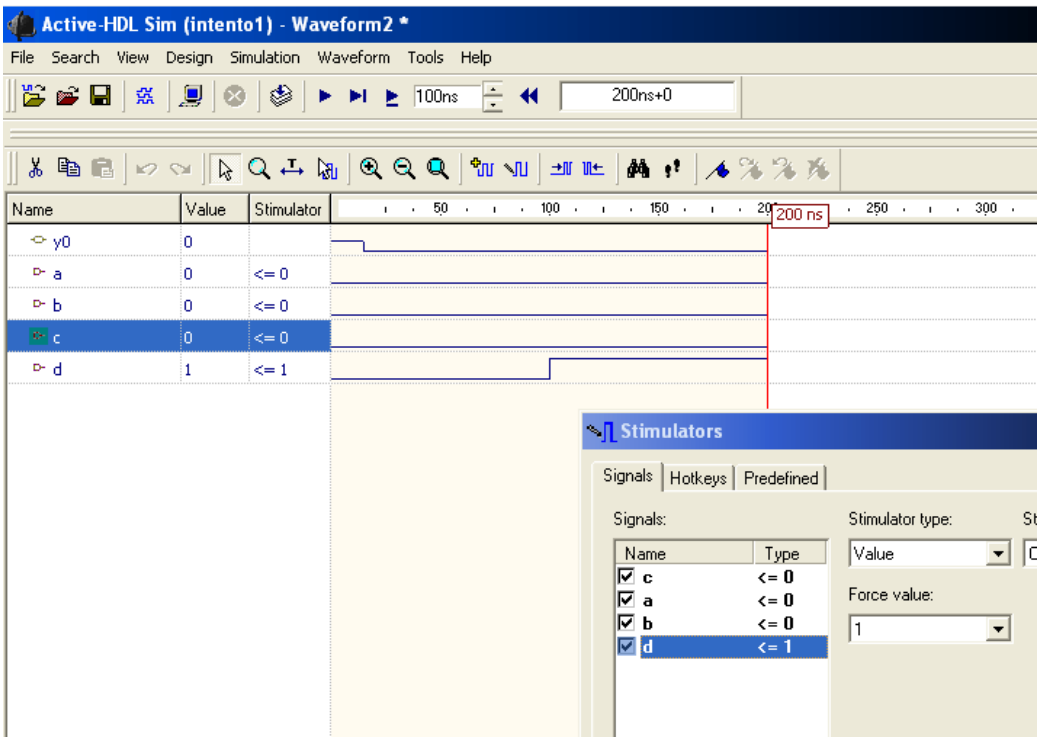
USO DE GALAXY



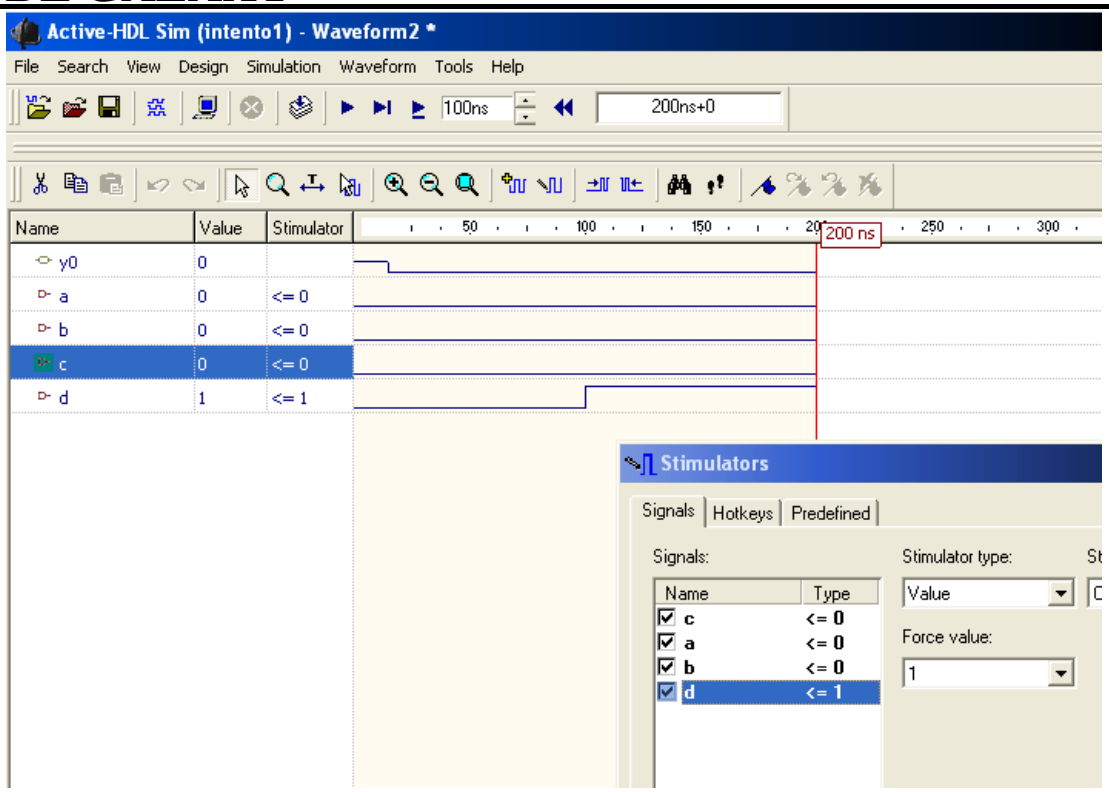
A	B	C	D	F0
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

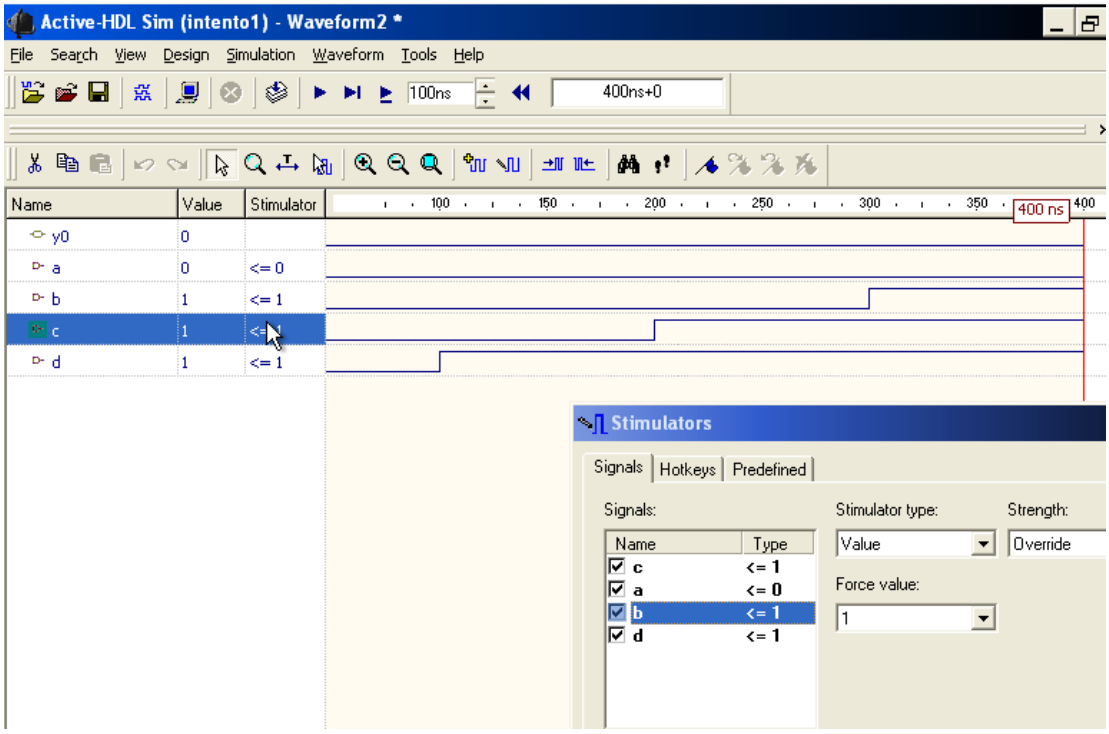
COMPUERTA AND





USO DE GALAXY





USO DE GALAXY

Como vemos en la entrada 1111 en la salida nos da 1 que es el funcionamiento de la compuerta AND

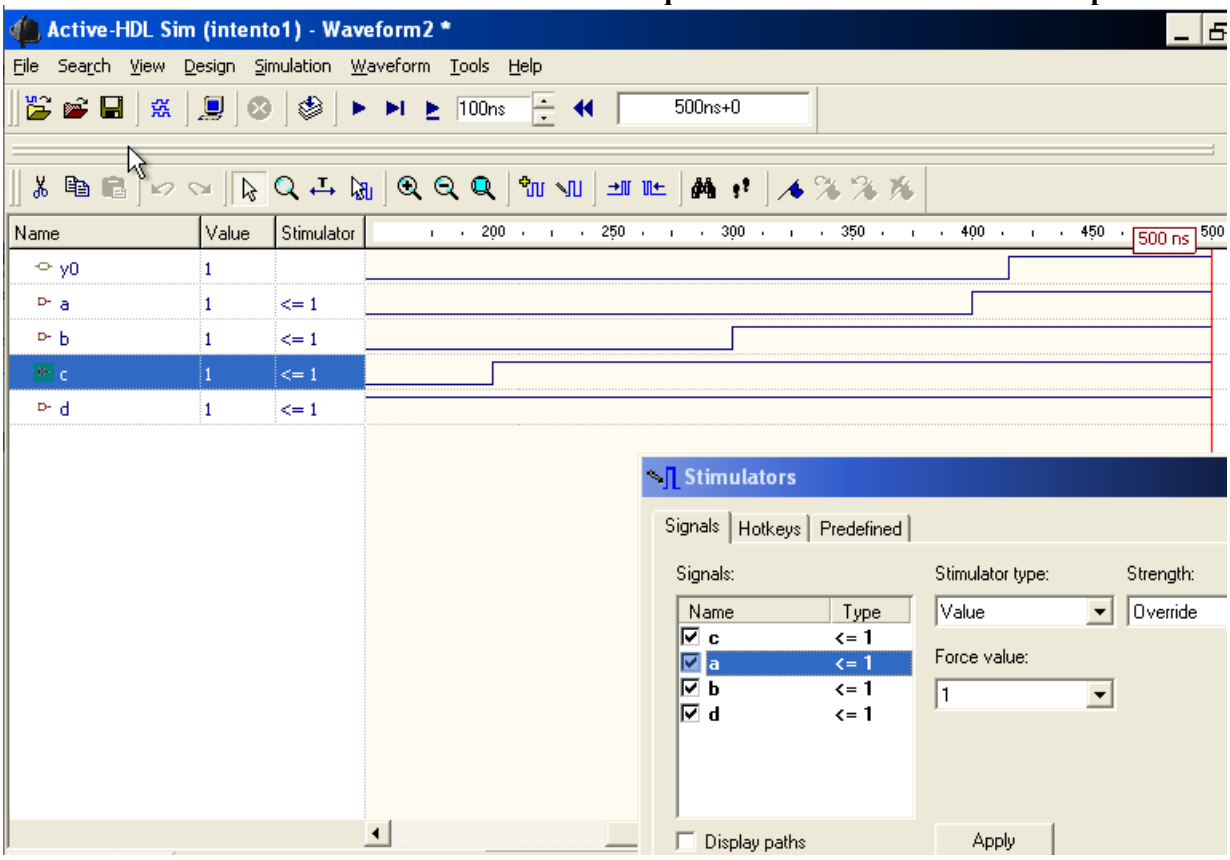
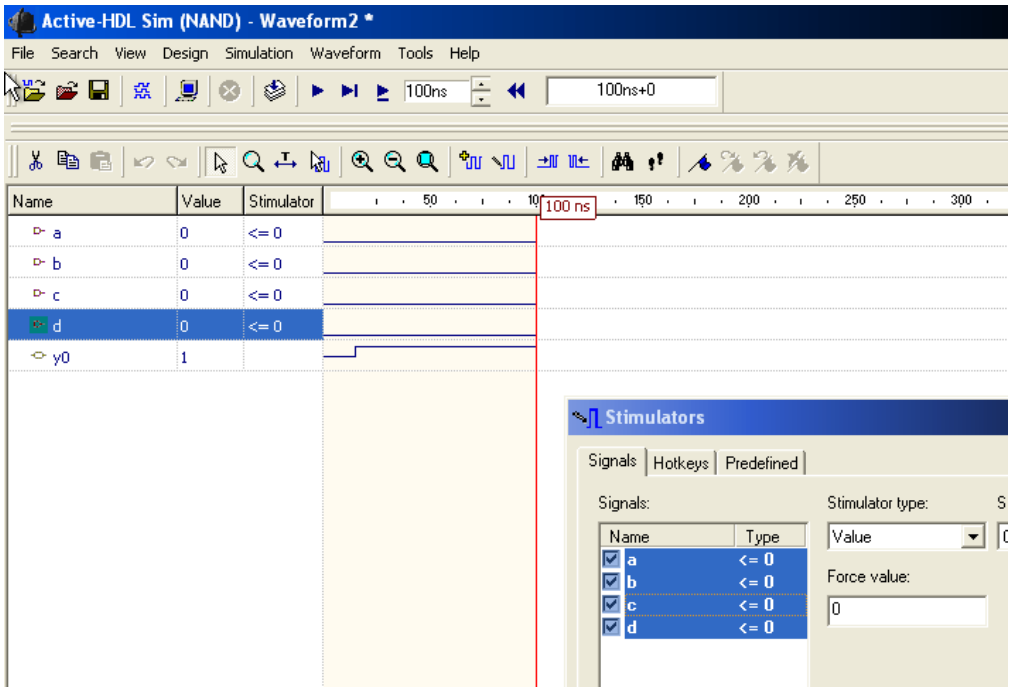


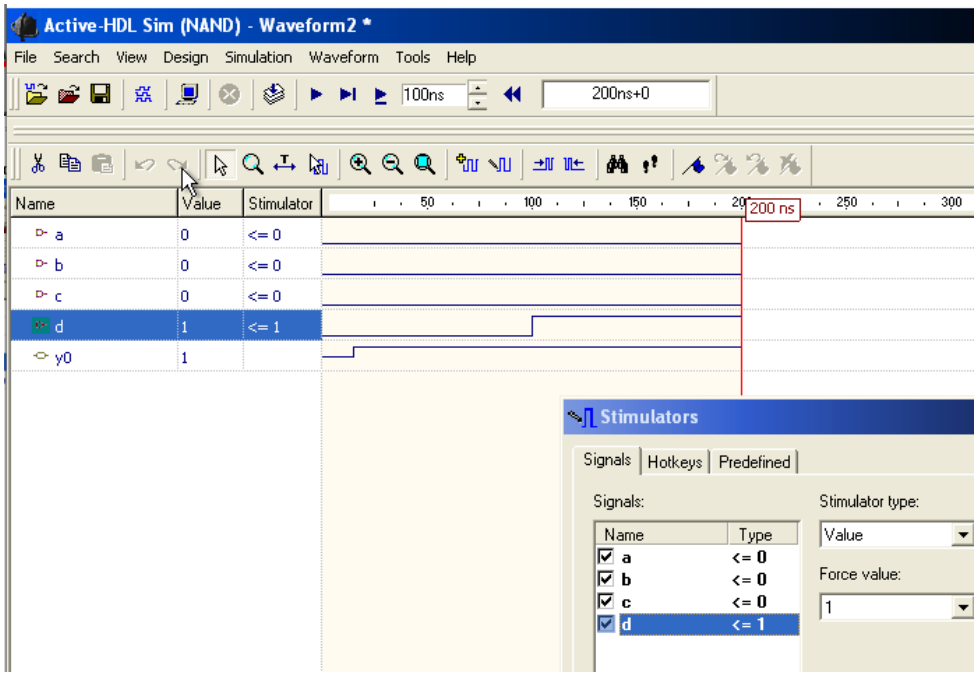
TABLA DE VERDAD

A	B	C	D	F0
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

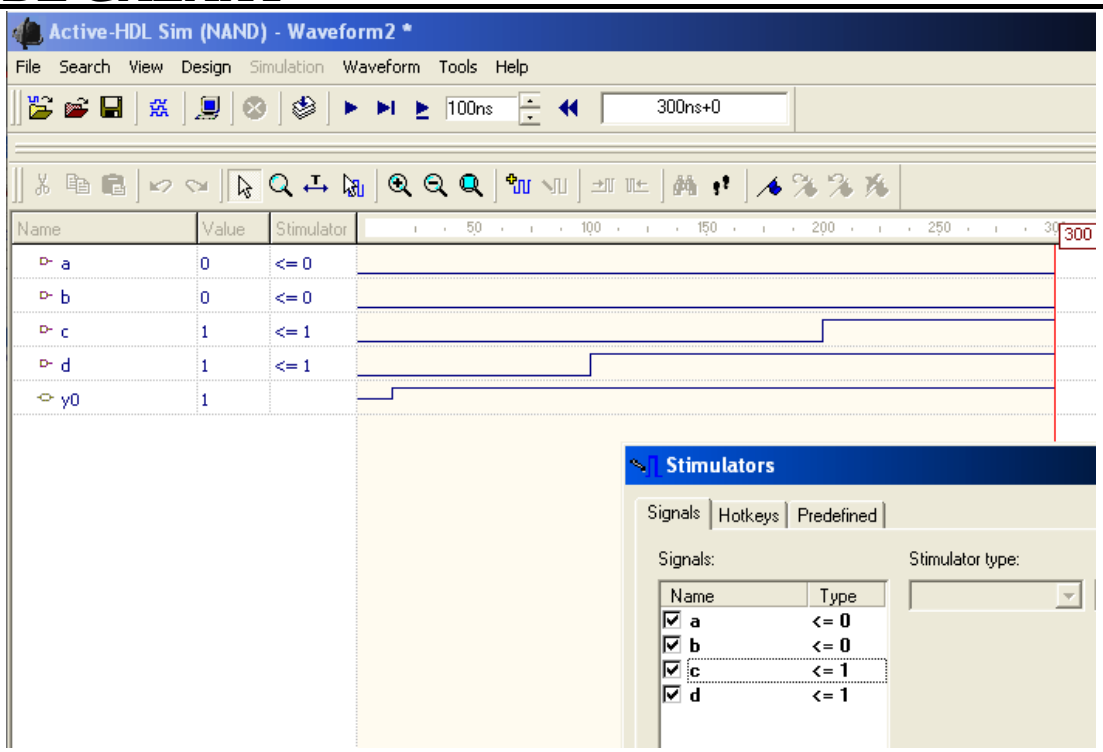
COMPUERTA NAND
0000 =>1, 1111=>0

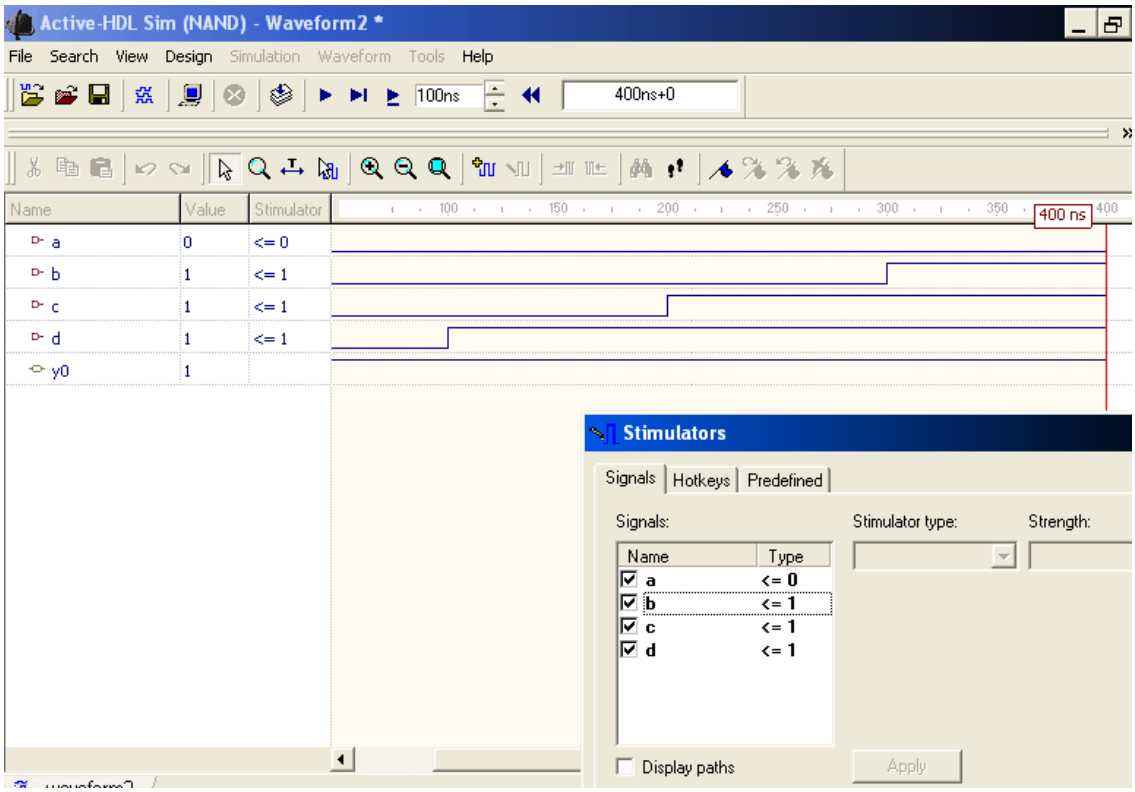
Como todas las entradas son 0 la salida será 1





USO DE GALAXY





USO DE GALAXY

Para la entrada 1111 la salida tendrá que ser 0 como lo dicta la tabla de verdad

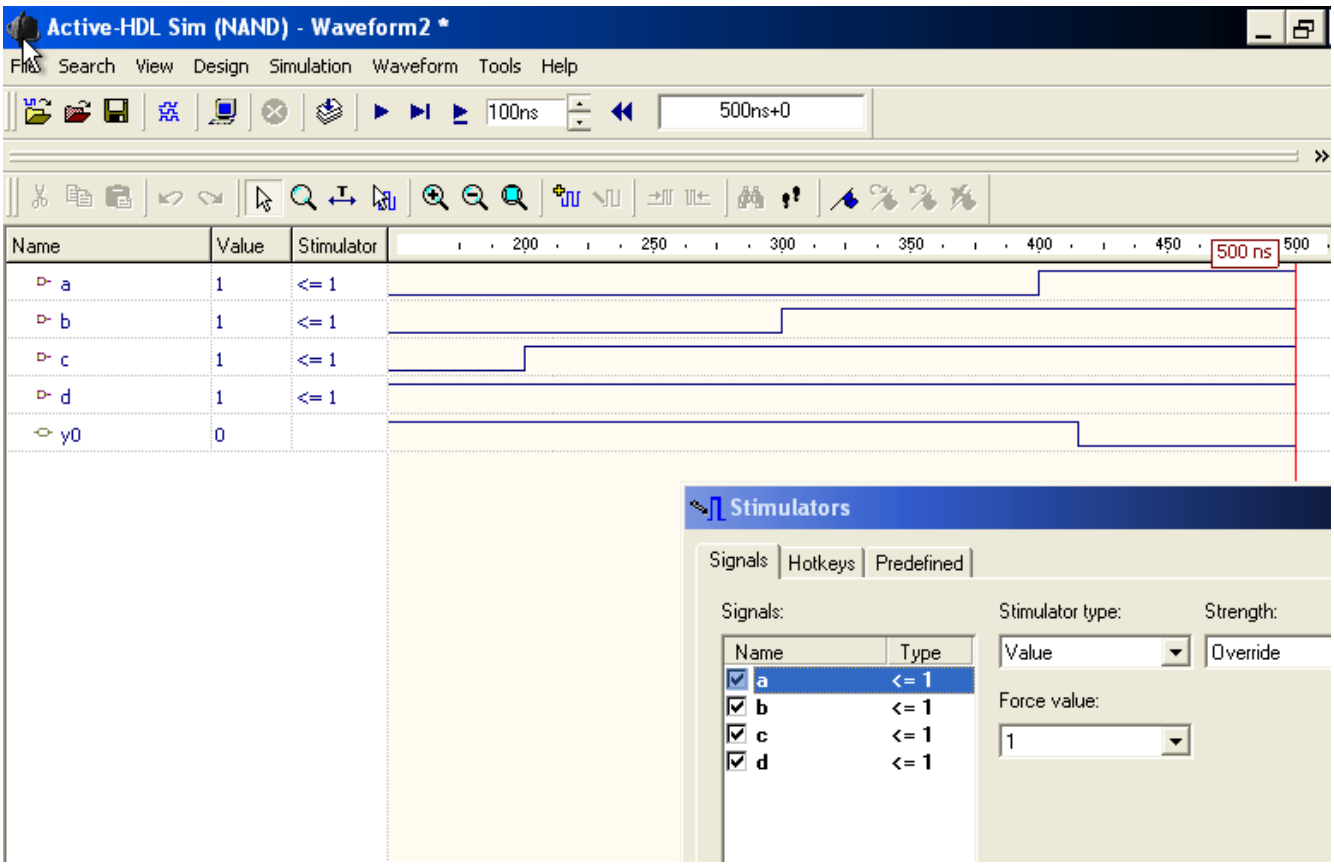


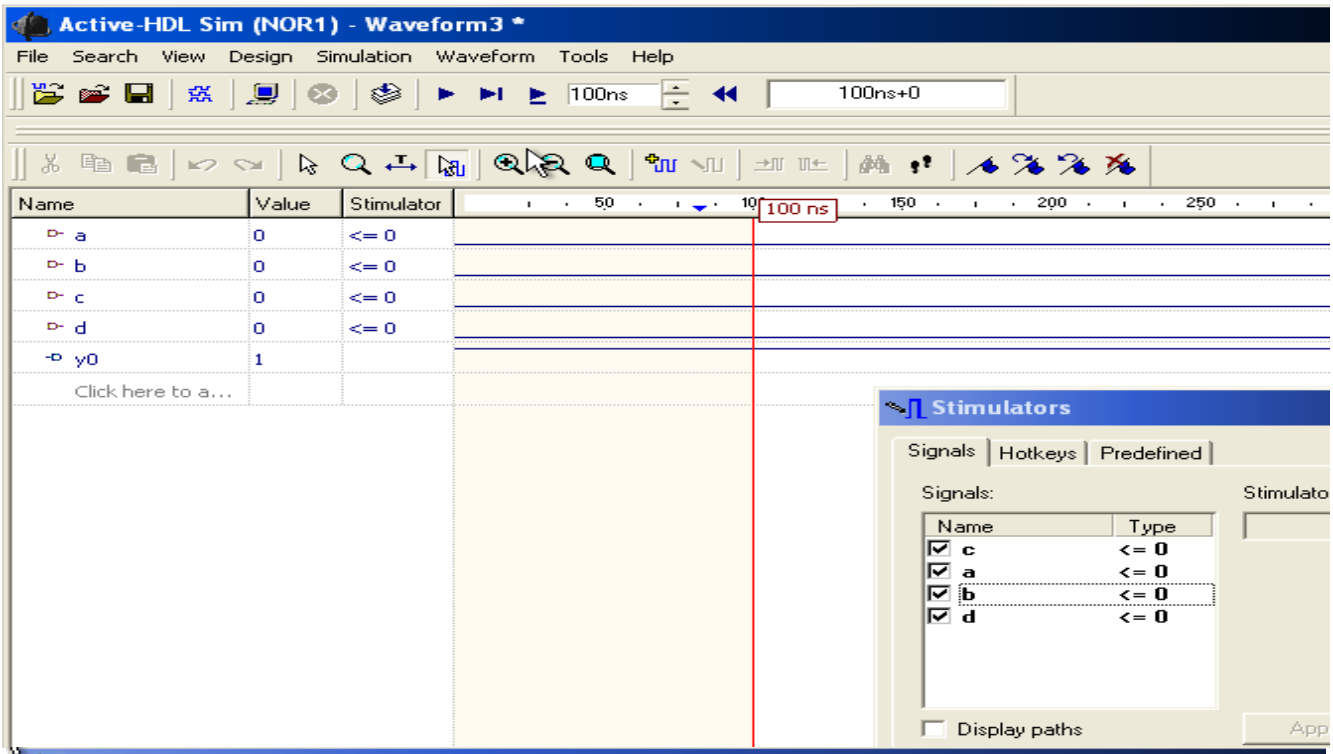
TABLA DE VERDAD

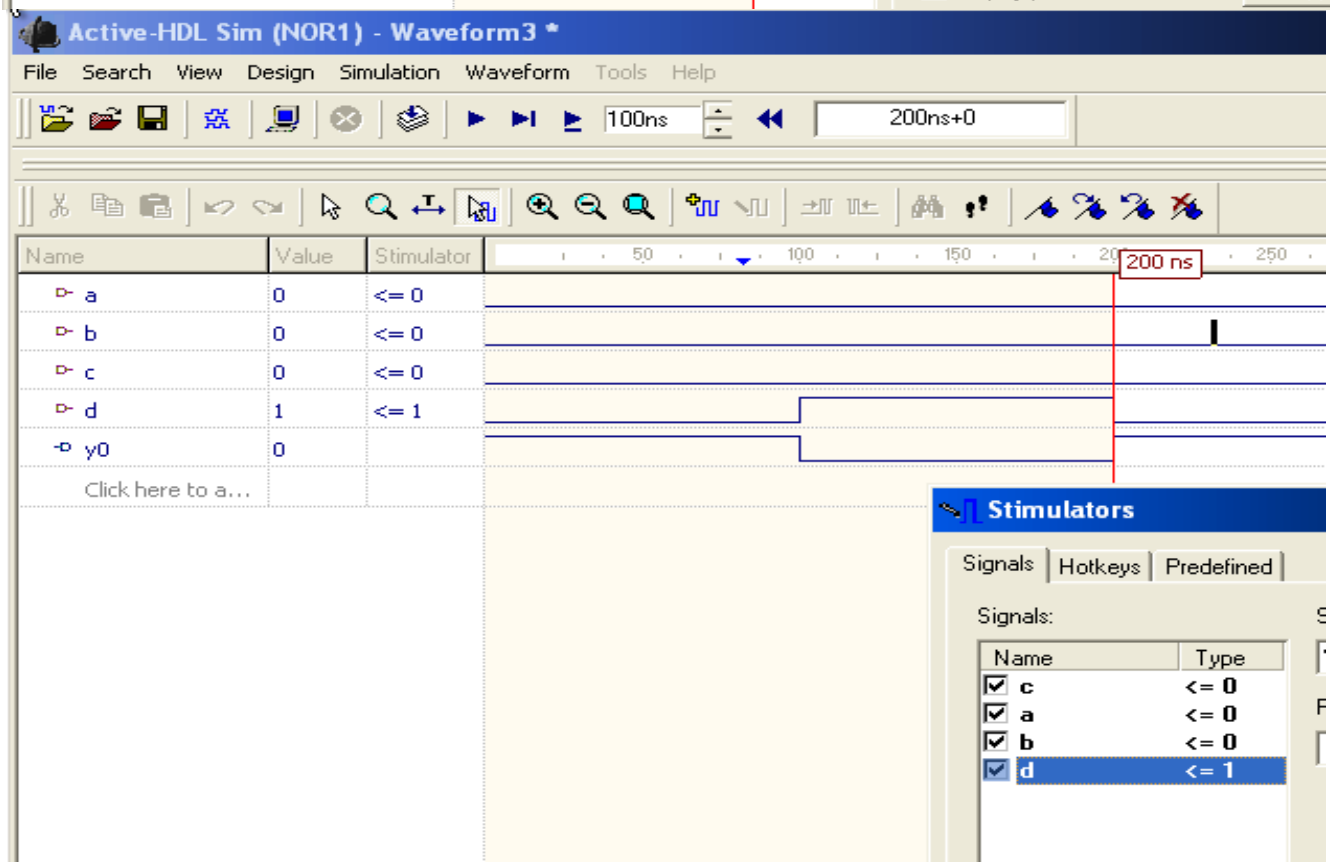
A	B	C	D	F0
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

USO DE GALAXY

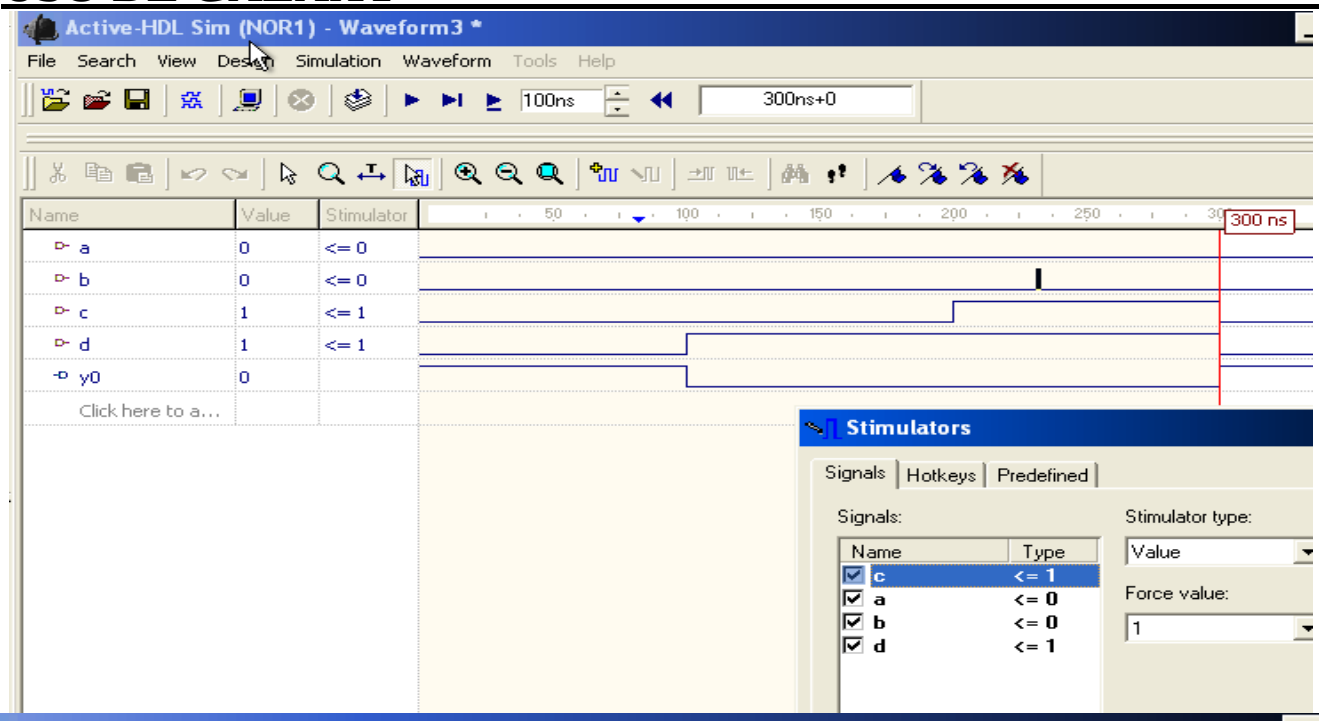
COMPUERTA NOR

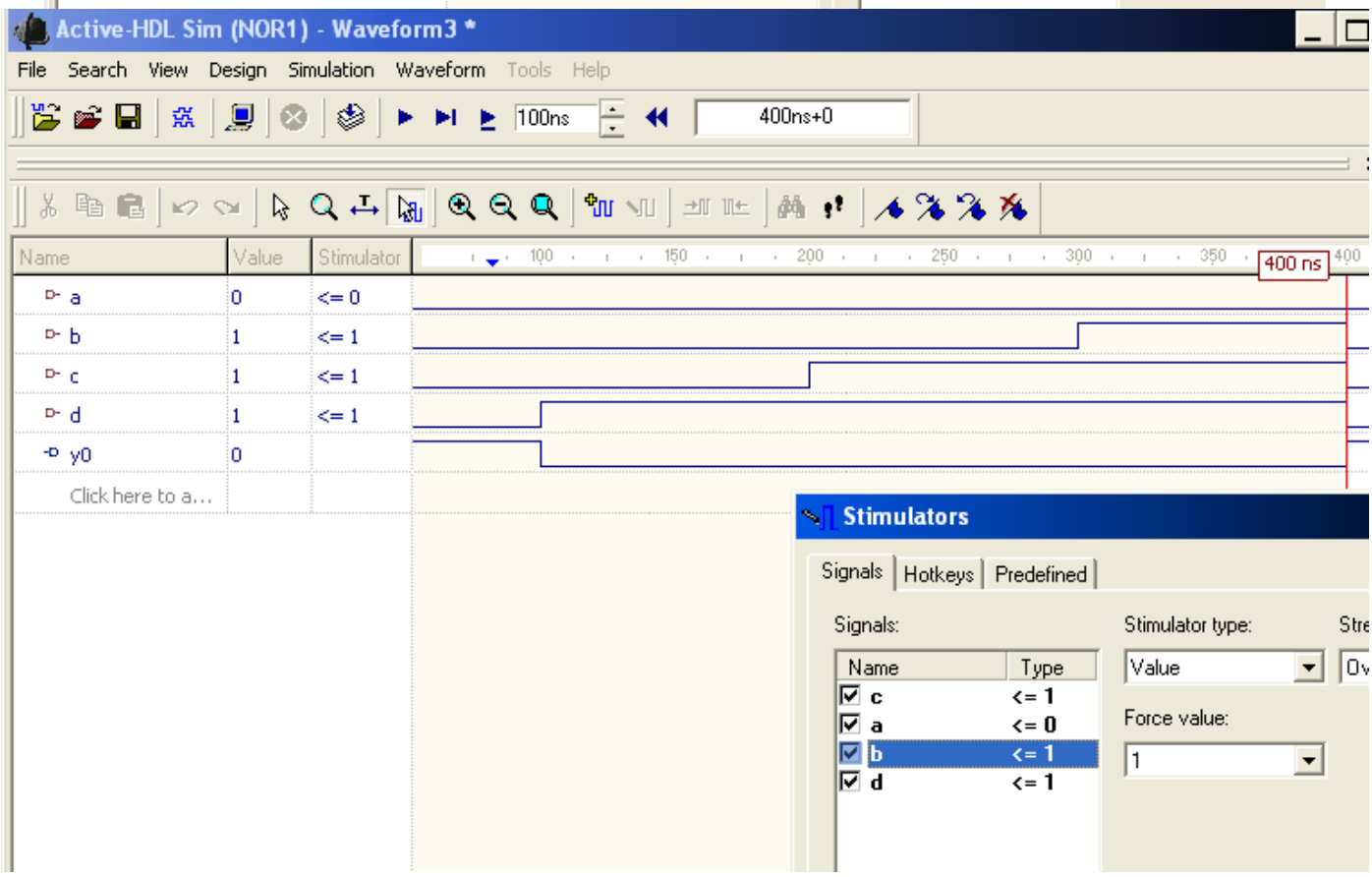
0000 =>1 , 1111 =>0





USO DE GALAXY





USO DE GALAXY

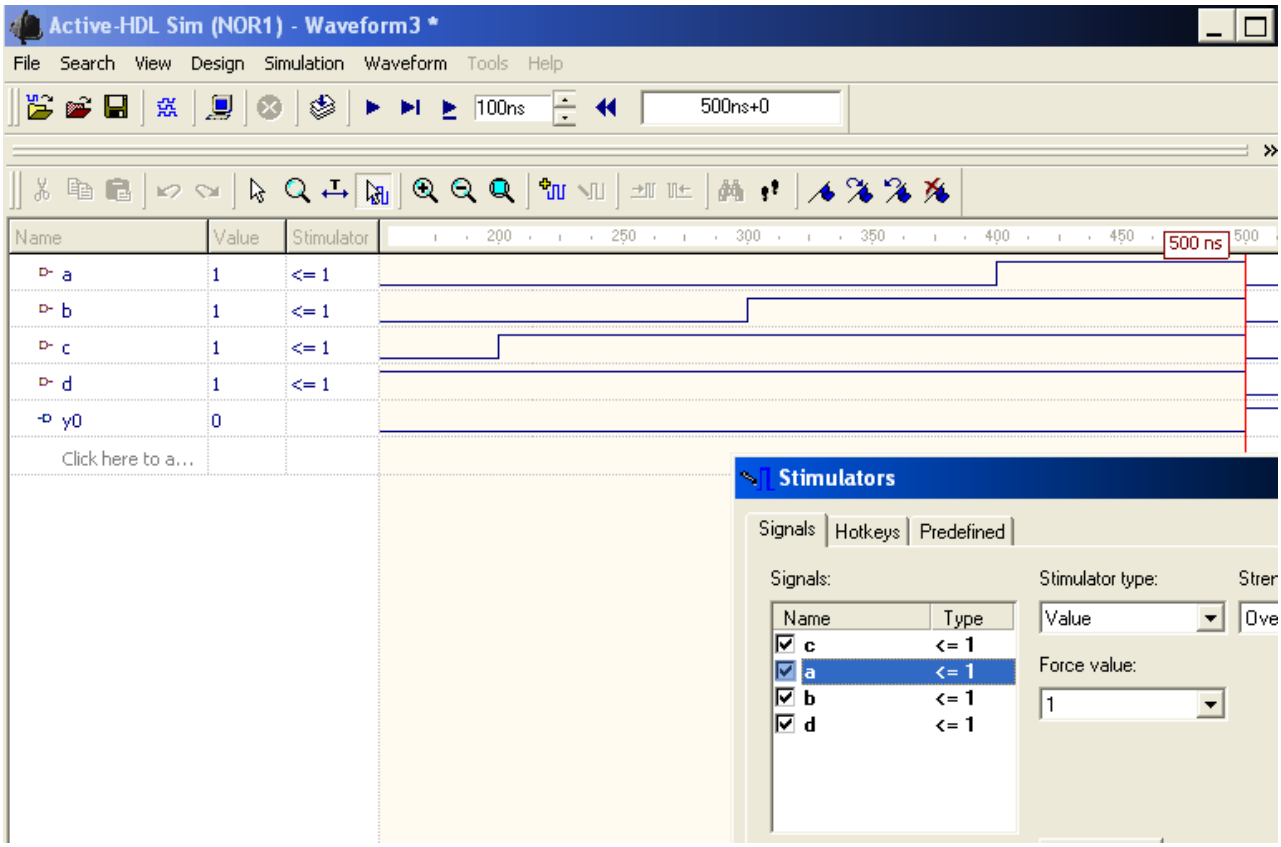
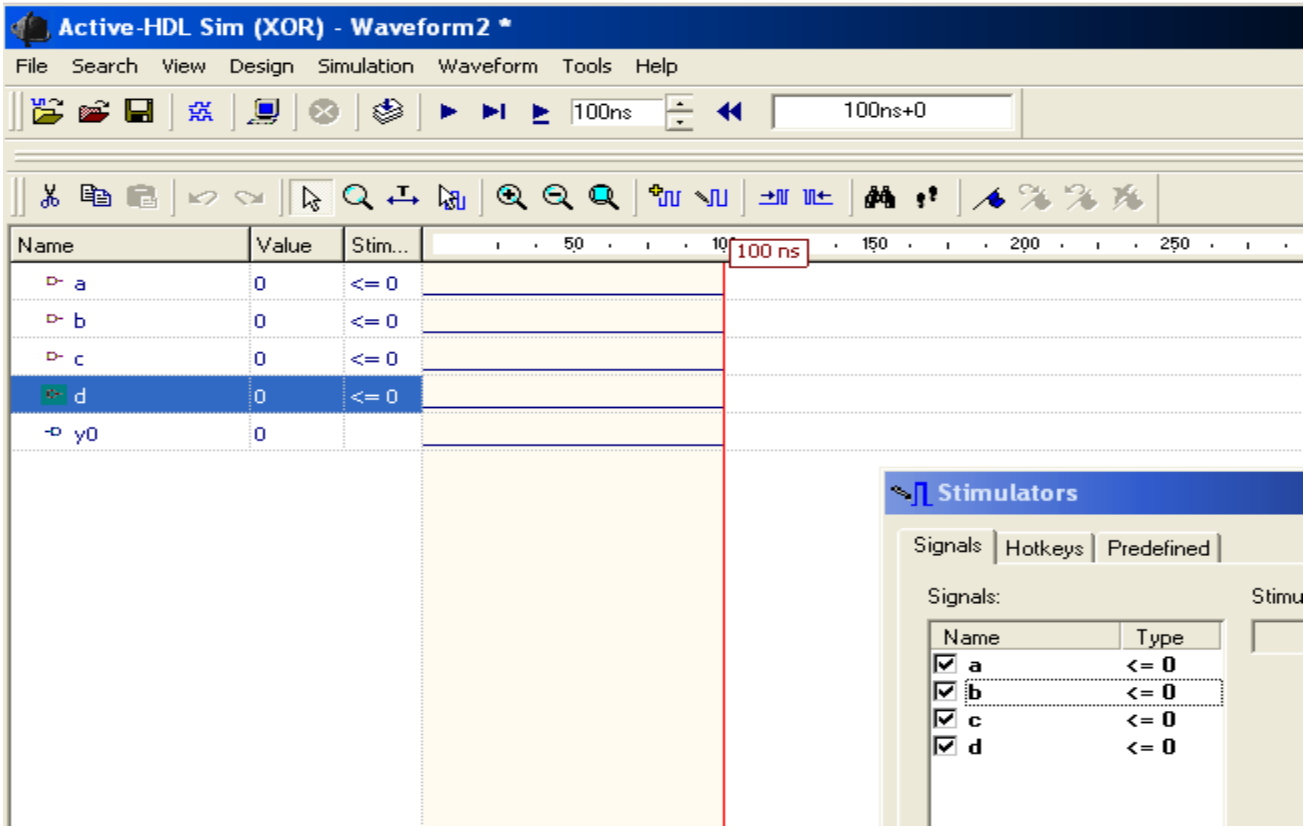


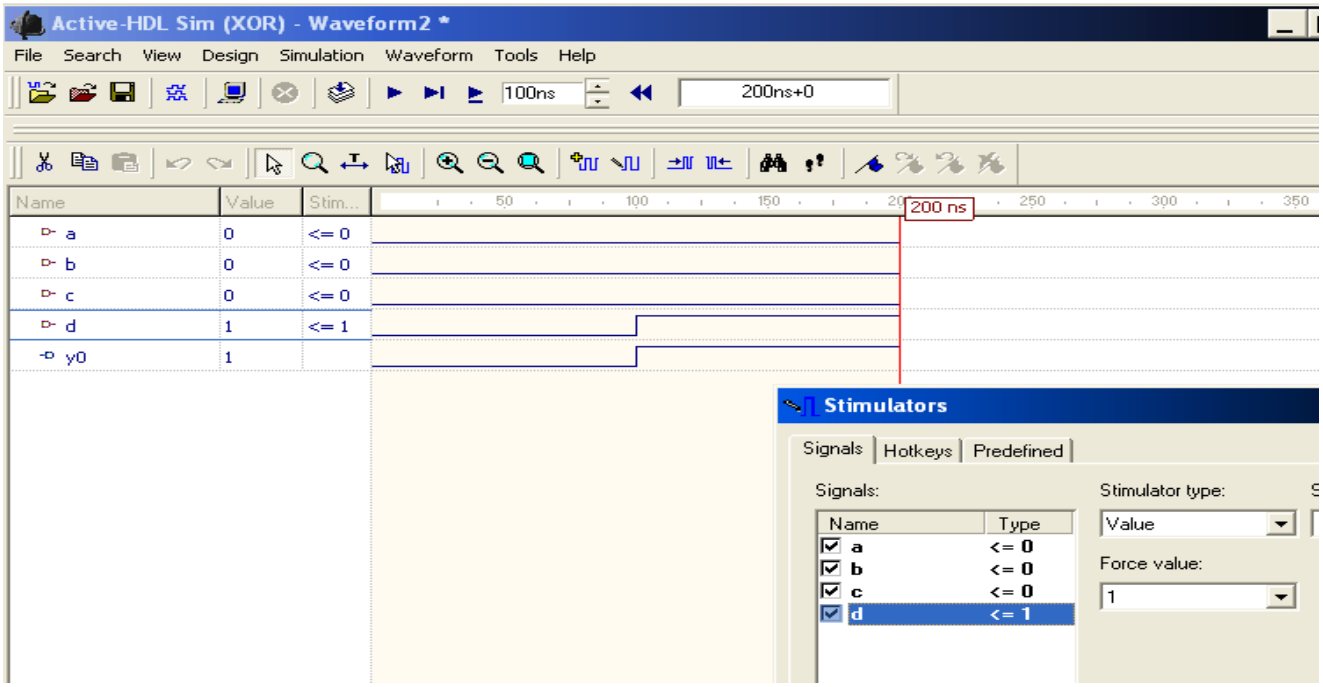
TABLA DE VERDAD

A	B	C	D	F0
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

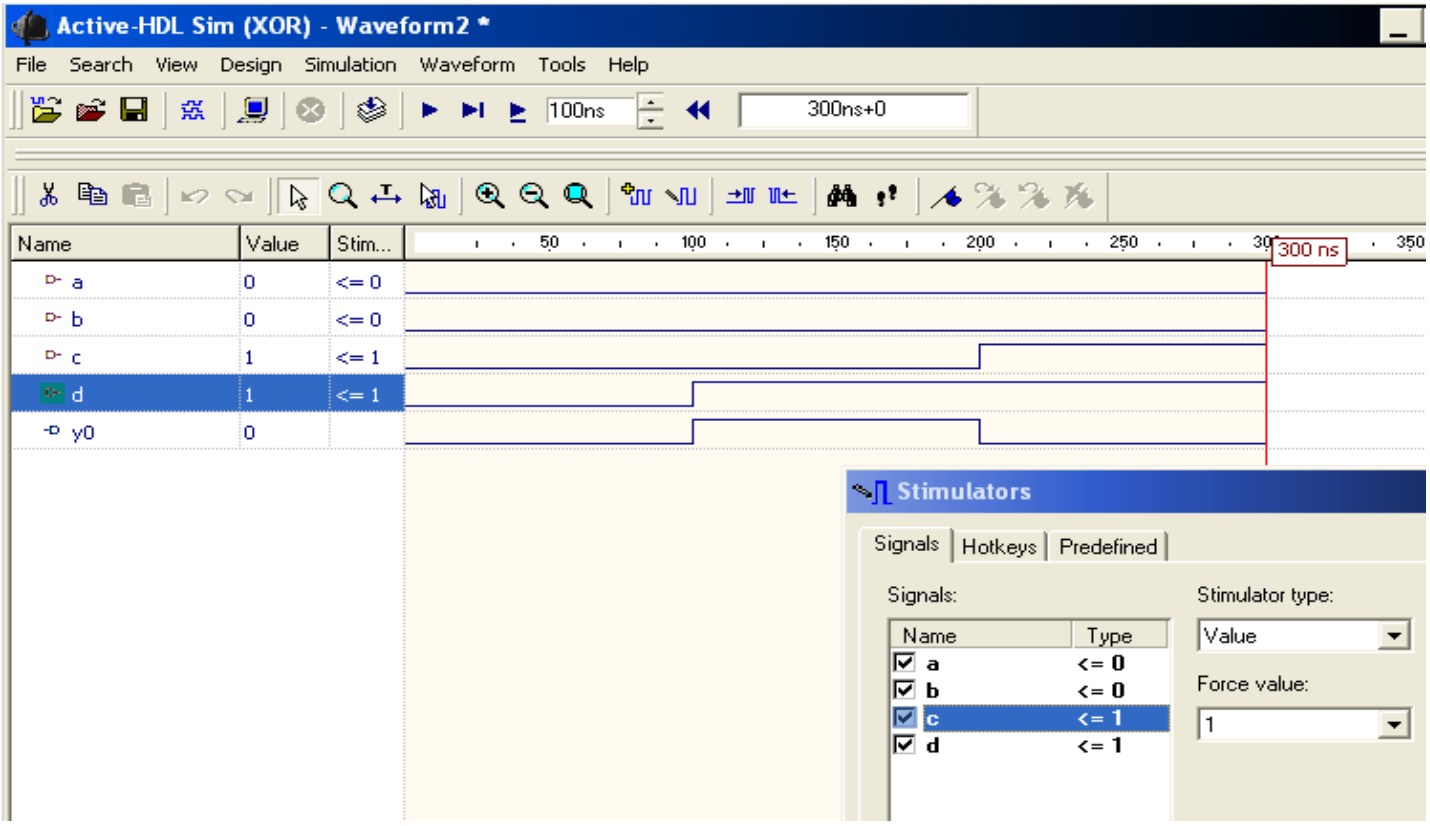
COMPUERTA XOR

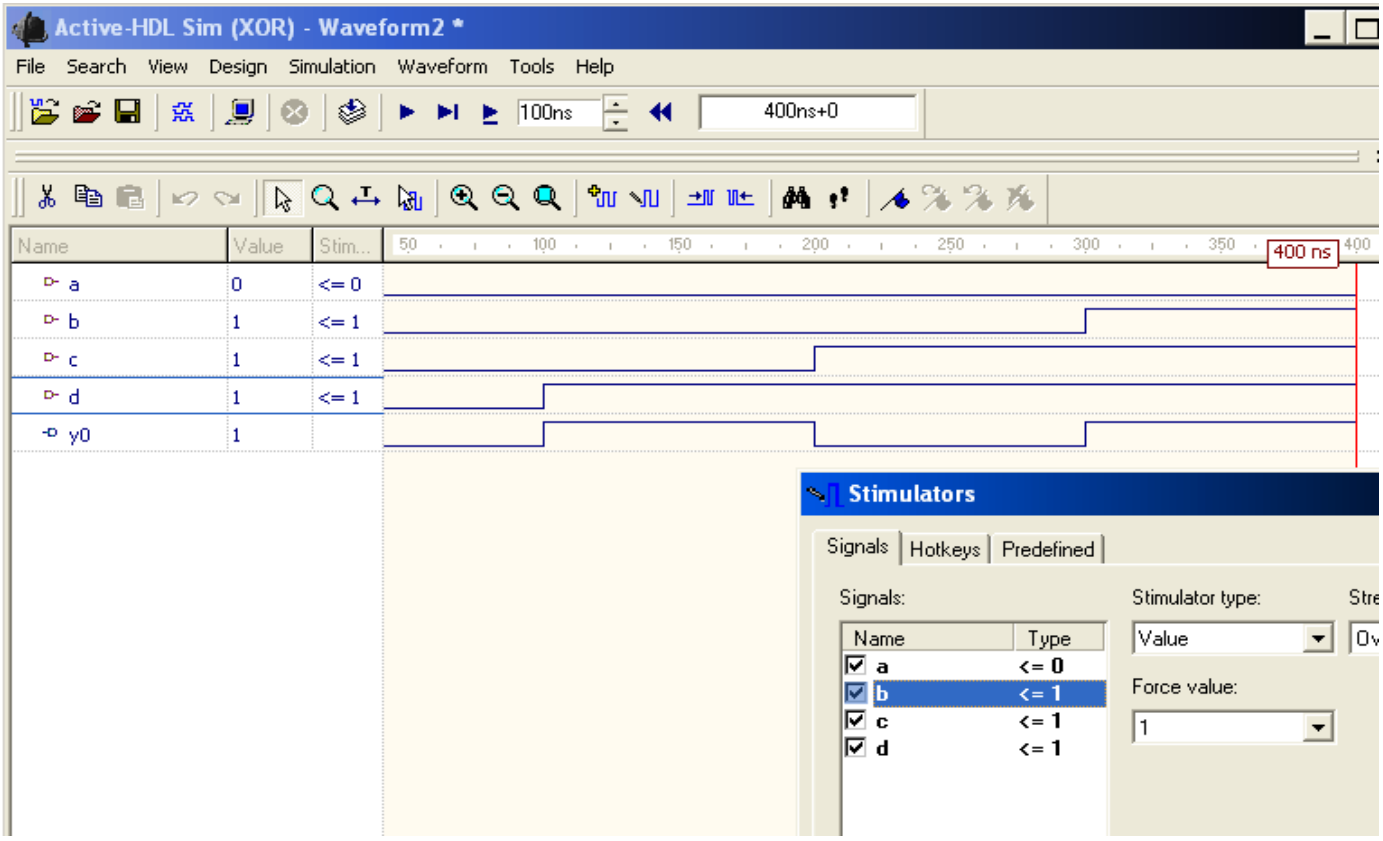
1111=>0, 0000 => 0





USO DE GALAXY





USO DE GALAXY

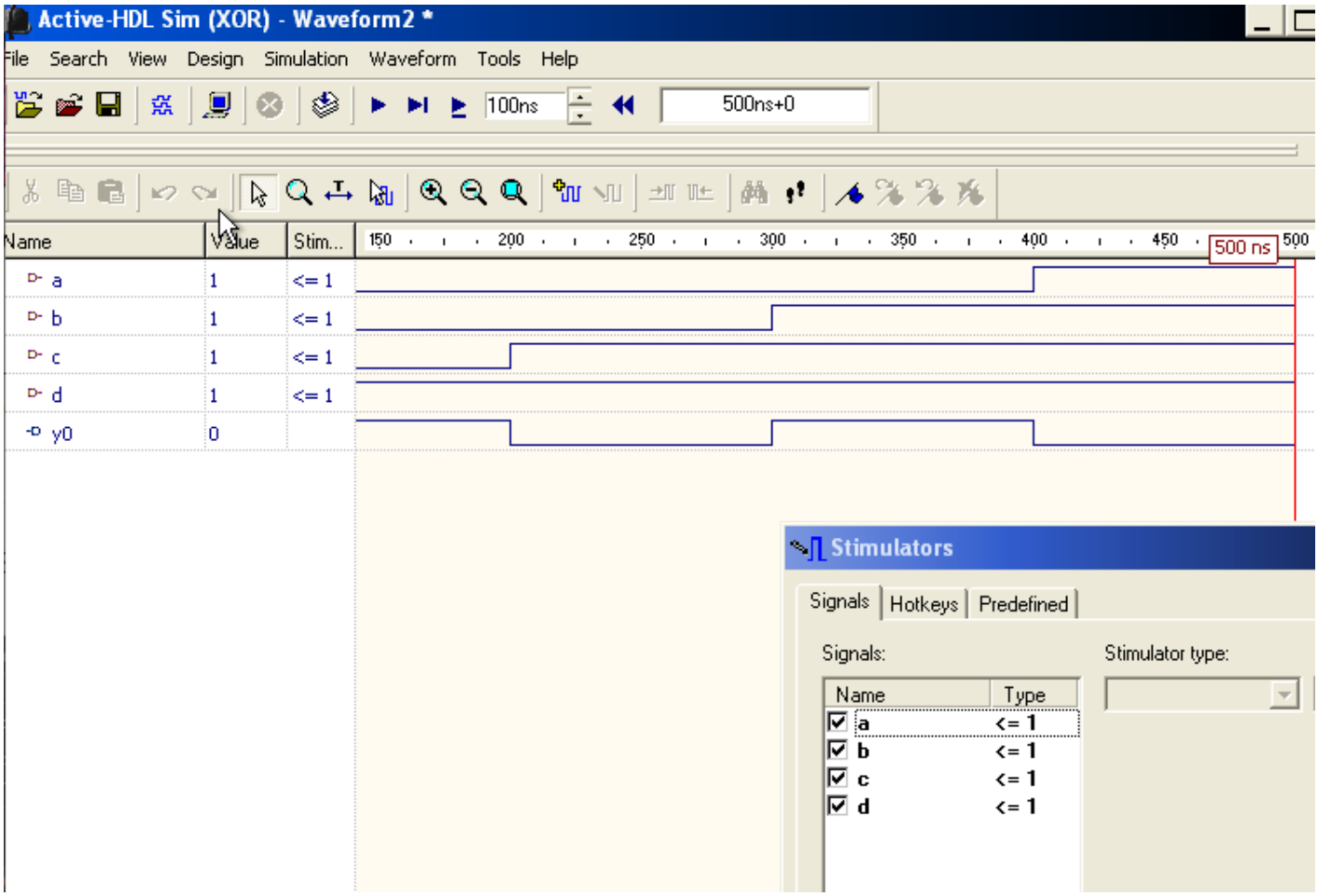
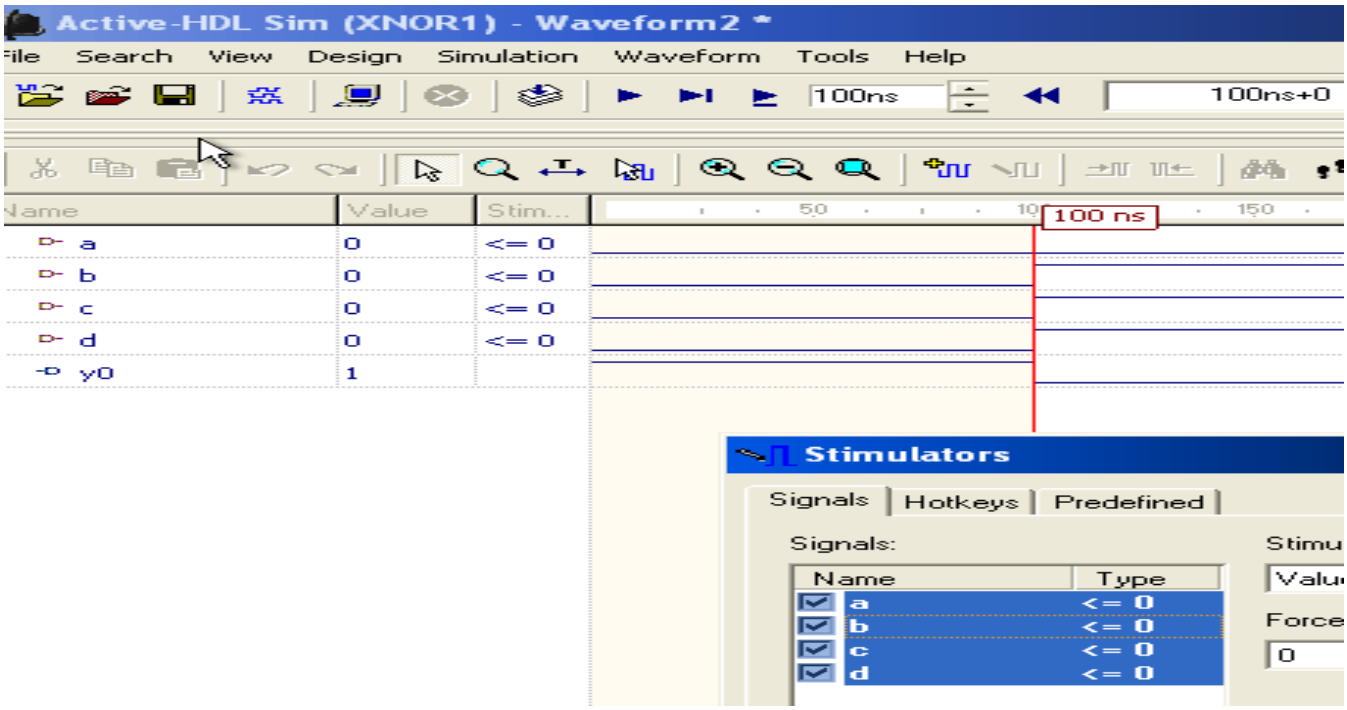


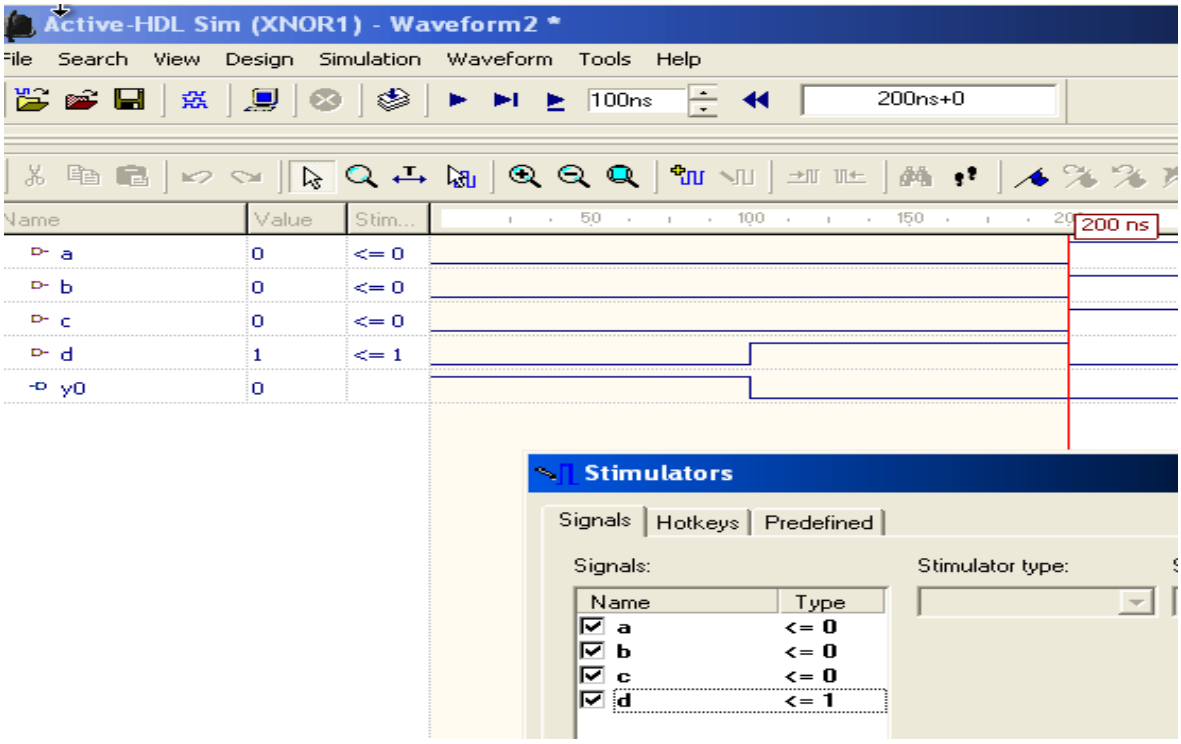
TABLA DE VERDAD

A	B	C	D	F0
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

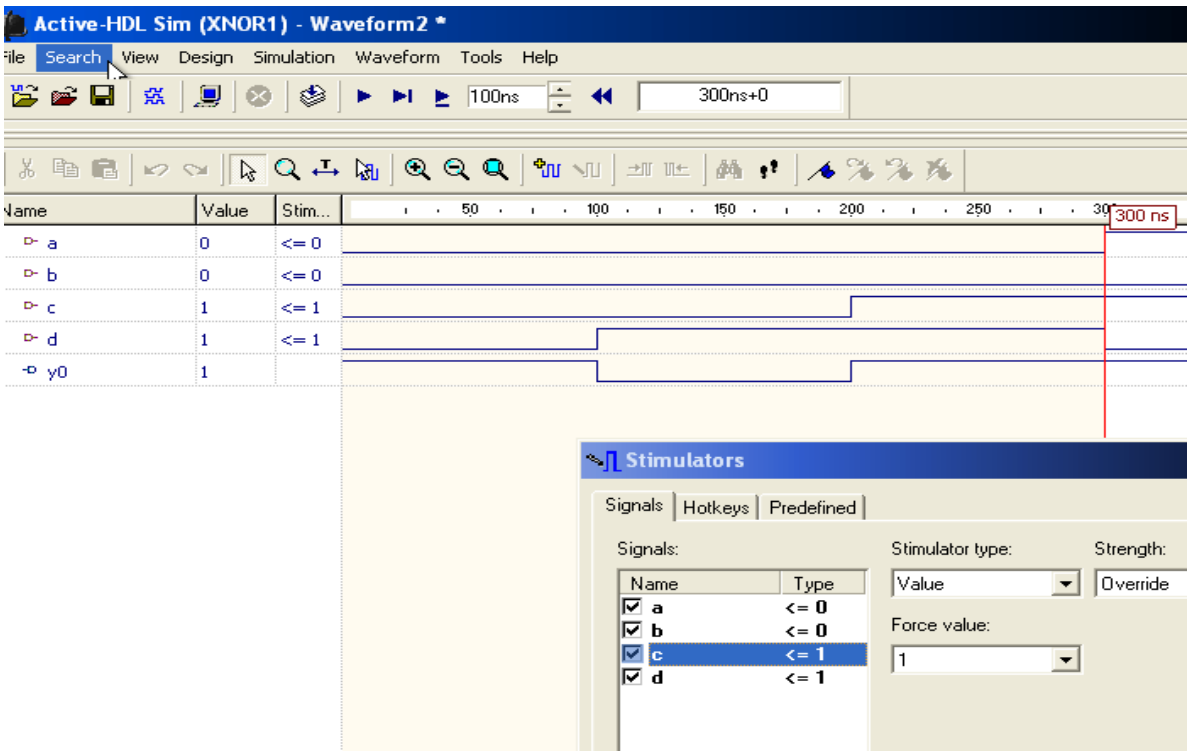
USO DE GALAXY

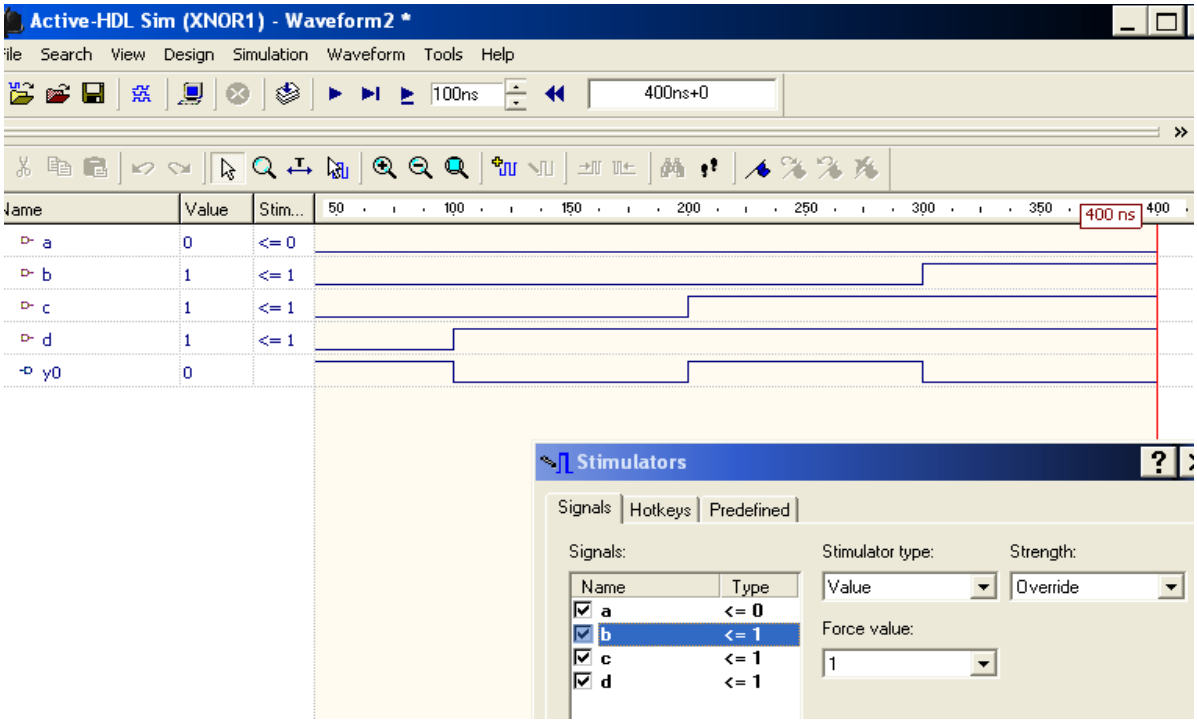
COMPUERTA XNOR





USO DE GALAXY





USO DE GALAXY

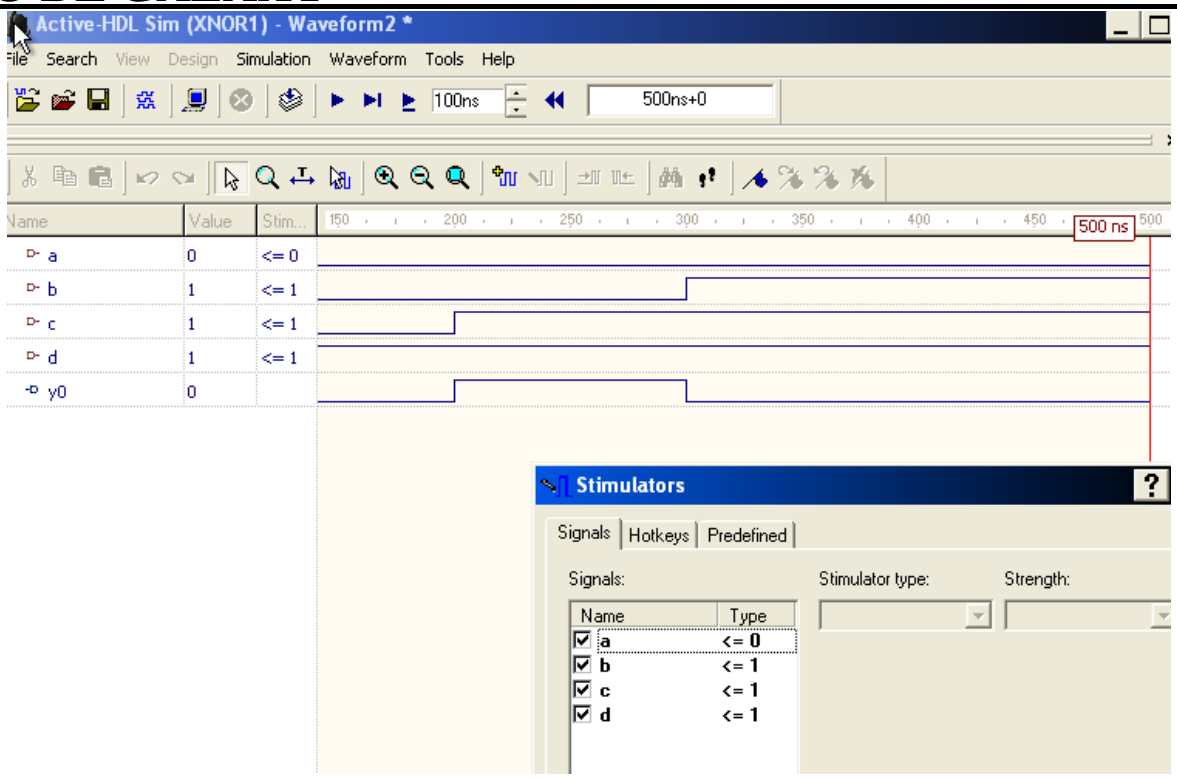


TABLA DE VERDAD

A	B	C	D	F0
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

USO DE GALAXY

OBSERVACIONES Y CONCLUSIONES:

En esta ocasión las cosas cambiaron, ahora en vez de ocupar las compuertas lógicas

de toda la vida utilizamos un software que solo se puede utilizar en WXP si bien es

complicado, al final nos ahorraría demasiado esfuerzo (nos ahorra conexiones, compuertas físicas, menor coste) ya que este programa que se “instalara ” en el GAL ya trae todo lo que necesitamos

previamente definido en el codigo.. en este se por medio de ARCHITECTURE se definiría que es y lo que hará nuestro circuito, y en ENTITY se definen las variables para decirlo de una manera muy comprensible

si bien en los circuitos que hicimos es complicado e inpractico pero a la larga en un circuito ya mas extenso nos ahorraría muchos recurso como se menciono antes. E ahí la ventaja de vhd

USO DE GALAXY