

INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

**LABORATORIO DE
FUNDAMENTOS DE DISEÑO DIGITAL**

PRÁCTICA No. 1

“USO BASICO DEL IDE GALAXY”

GRUPO: _____

EQUIPO: _____

INTEGRANTES: _____

PROFESOR: _____

FECHA DE REALIZACIÓN: _____

FECHA DE ENTREGA: _____

COMENTARIOS DEL PROF: _____

I OBJETIVO GENERAL:

Al finalizar la práctica, el alumno será capaz de implantar una función lógica en un dispositivo lógico programable (PLD), del tipo GAL, empleando el Ambiente de Desarrollo Integrado (IDE de Integrated Development Environment) Galaxy, de Cypress Semiconductor Corporation. Adicionalmente el alumno entenderá los conceptos de herramientas CAD-EDA y comprobará físicamente el correcto funcionamiento de la función implantada.

II OBJETIVOS ESPECÍFICOS:

- I. Aprender las características básicas del IDE de Galaxy.
- II. Aprender a describir un circuito, empleando un HDL.
- III. Programar el dispositivo o PLD.
- IV. Distinguir los beneficios proporcionados por las herramientas CAD.
- V. Confirmar el conocimiento adquirido en clase.
- VI. Comprobar físicamente, sobre un protoboard, el correcto funcionamiento de la función programada.

III MATERIAL Y EQUIPO EMPLEADO

- Mesa de instrumentación, del laboratorio de sistemas digitales
- Programador universal
- Fuente de 5V
- 1 GAL22V10
- 1 DIP switch de 8
- 6 Resistencias de 1K Ω
- 1 Resistencia de 220 Ω
- 1 Resistencia de 330 Ω

IV INTRODUCCIÓN TEÓRICA:

4.1 Las herramientas CAD-EDA.

Agregar una breve descripción de dichos temas.

4.2 Lenguajes de descripción de hardware (HDL)

Agregar una breve descripción de dicho tema.

4.3 Los dispositivos lógicos programables (PLD's)

Agregar una breve descripción de dichos dispositivos.

4.3.1 La GAL 22V10

Agregar una descripción del dispositivo, tal como la arquitectura, registros, etc. (Ver hoja de especificaciones de dicho dispositivo).

4.4 Herramientas de desarrollo

Hablar de las herramientas de desarrollo a emplear o empleadas. Investigar sobre otras...

Ejemplo:

*El ambiente de desarrollo aquí utilizado es el **Galaxy ver. 6.3**, de Cypress Semiconductor Corporation. El ambiente es un "Entorno de Desarrollo Integrado" conformado por "tales" herramientas, etc.....*

V DESARROLLO

5.1 Problema

Implementar los circuitos:

1. Un circuito multiplexor 4X1, con dos entradas de selección. La tabla de verdad característica de un multiplexor de 2X1 se muestra en la tabla 1. **Recordar:** Un multiplexor tiene 2^n posibles entradas y una única salida, donde n son la(s) entrada(s) de selección.
2. El circuito que se describe con la tabla de verdad, mostrada en la tabla 2.

Se pide:

- 1) Escribir la función de salida:
 - a) En su forma canónica o normalizada.
 - b) Simplificada, empleando álgebra de Boole.
 - c) En función de compuertas universales NAND.
- 2) Dibujar el circuito resultante:
 - a) Empleando compuertas básicas.
 - b) Empleando compuertas universales NAND.
- 3) Crear un proyecto en VHDL, simular su comportamiento y programar la GAL22V10.
- 4) Dibujar el diagrama del circuito y probar su correcto funcionamiento, empleando un protoboard.

ENTRADAS			SALIDA
Sel	a	b	Sal
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Tabla 1. Tabla de verdad de un multiplexor de 2X1.

Entrada				Salida
A	B	C	D	Sal
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
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

Tabla 2. Tabla de verdad del circuito 2.

5.2 Solución del problema

Se pide:

- 1) Escribir la función de salida:
 - a) En su forma canónica o normalizada.
 - b) Simplificada, empleando álgebra de Boole.
 - c) En función de compuertas universales NAND.
- 2) Dibujar el circuito resultante:
 - a) Empleando compuertas básicas.
 - b) Empleando compuertas universales NAND.
- 3) Crear un proyecto en VHDL, simular su comportamiento y programar la GAL22V10.
- 4) Dibujar el diagrama del circuito y probar su correcto funcionamiento, empleando un protoboard.

La metodología de diseño empleada fue:

(Nota: Dibujar el diagrama a flujo de la metodología de diseño)

Figura 5.1. Diagrama a flujo de la metodología de diseño.

5.2.1 Solución del problema 1.

5.2.1.1 Función de salida.

5.2.1.2 Circuito resultante.

5.2.1.3 Creando proyecto con el IDE Galaxy.

Una vez que el software esta instalado en el computador y listo para ser utilizado, los pasos a seguir para trabajar con Galaxy son los descritos a continuación:

1. Ejecutar el software Galaxy.

2. Se crea un archivo de texto mediante: **File→New→Text File** y después **<OK>**.
3. Se escribe el código, mostrado en el cuadro 1, y después se salva el archivo, preferentemente con el nombre de la ENTIDAD, y con extensión *.vhd. (En este caso se llama ***mux.vhd***).

```

-- Circuito multiplexor 2 a 1

ENTITY mux IS
PORT (a, b, sel: IN bit;
      sal: OUT bit);
END mux;

ARCHITECTURE flujo2 OF mux IS
BEGIN
  sal <= a WHEN sel = '0' ELSE
  b;
END flujo2;

```

Cuadro 1. Código de un multiplexor 2 a 1.

4. Una vez salvado el archivo se crea un proyecto, incluyendo dicho archivo. Esto se hace de la siguiente manera:
 - a) **File→New→ Project [Target-Device]**. Esto abre una ventana, como se muestra en la figura 1. En **Browse...** se introduce la ruta donde se salvará el proyecto, en este caso con el mismo nombre (***mux***). Además, se selecciona el lenguaje que se va a utilizar (**VHDL** o **Verilog**), que en este caso es VHDL.

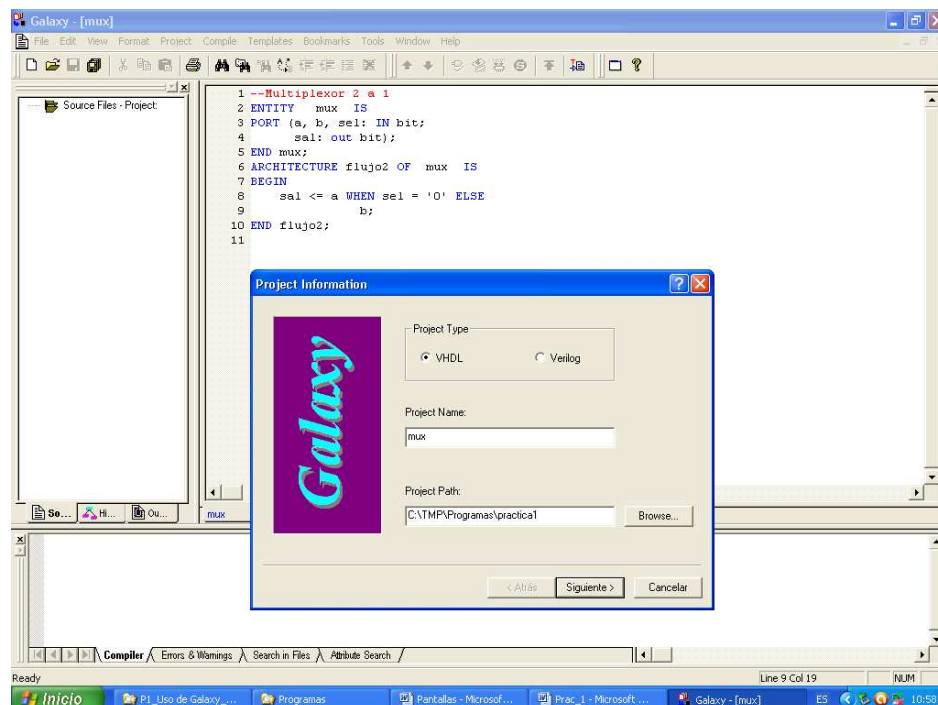


Figura 1. Ruta y nombre del proyecto (mux).

- b) Se da clic en **<siguiente>** y aparece una nueva ventana que da la opción para agregar el archivo ***mux.vhd***, creado anteriormente, dando clic en **<add>** y

<siguiente>. Esto último abre una nueva ventana que permite seleccionar el dispositivo a utilizar, ver figura 2. En ésta práctica se utilizará un dispositivo del tipo GAL22V10, que es un dispositivo **SPLD** del tipo **C22V10**. En este caso se selecciona el PALCE22V10-25PC ó -15PC, según sea su retardo de programación (**speed (ns)**) y se da <Finalizar>.

- c) Finalmente, aparece una nueva ventana que pregunta si se desea salvar el proyecto, dando en <si>.

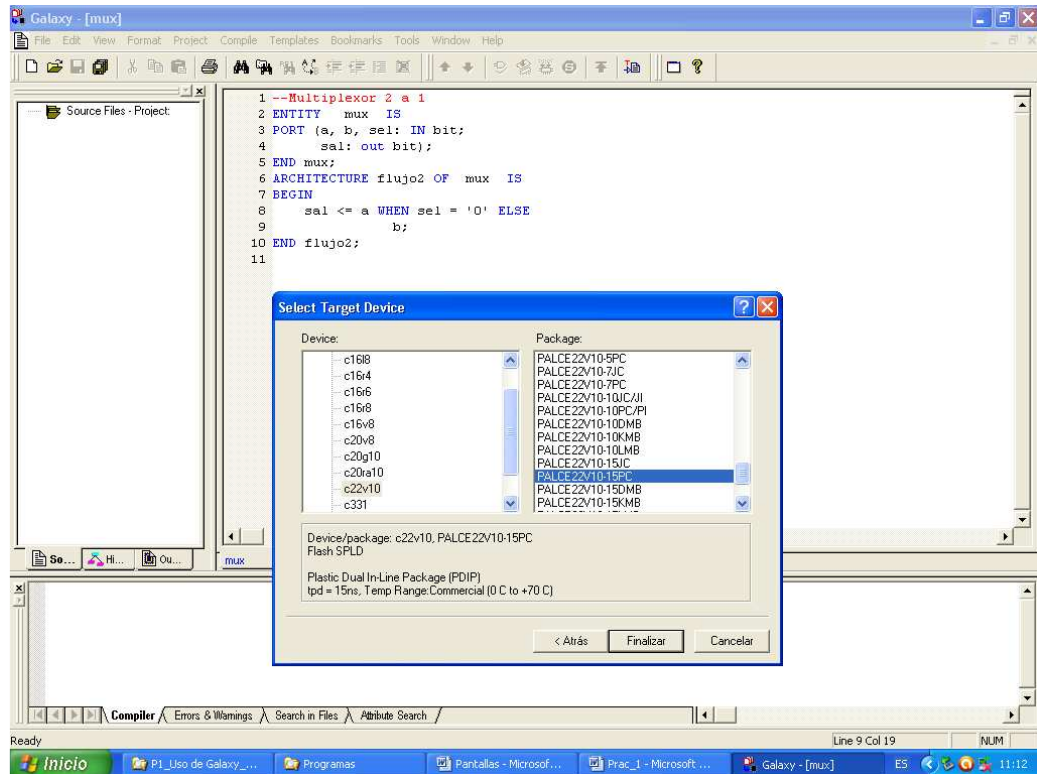


Figura 2. Selección del tipo de dispositivo.

5. Se puede observar en la figura 3, 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 **mux.vhd**, y se selecciona **Set Top**. Todo esto, como se muestra en la figura 3a). Una vez hecho esto es posible observar que el icono del archivo creado **mux.vhd** cambió, como se puede observar en la figura 3b).

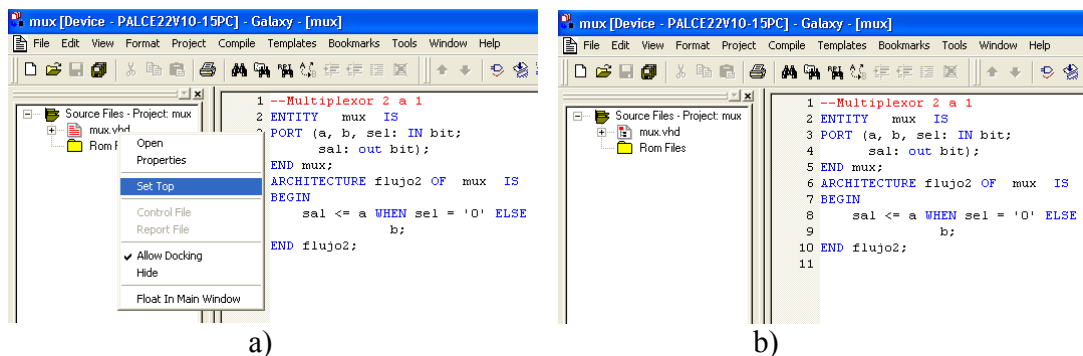


Figura 3. Indicar el archivo principal o de mayor nivel (top level).

6. Ya con esto se compila el archivo en: **Compile → Selected File(s)** y se ejecuta dicho proceso. En la parte inferior de la ventana se despliega la información del resultado de la compilación. En caso de que no fuese satisfactoria la compilación es posible saber en donde están los errores o advertencias. Para tener acceso a dicha información es necesario cambiar de pestaña. La pestaña en donde se despliegan los errores o advertencia dice: **Errors & Warnings**. La pestaña está localizada en la parte inferior de la ventana de abajo.
7. Para corroborar el funcionamiento correcto del programa es posible realizar la **SIMULACION FUNCIONAL** el mismo. Para esto, se va a: **Tool → Active-HDL Sim**, abriéndose una nueva ventana.
8. En esta nueva ventana se abre el archivo **mux.vhd**, esto mediante: **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, como se muestra en la figura 4.

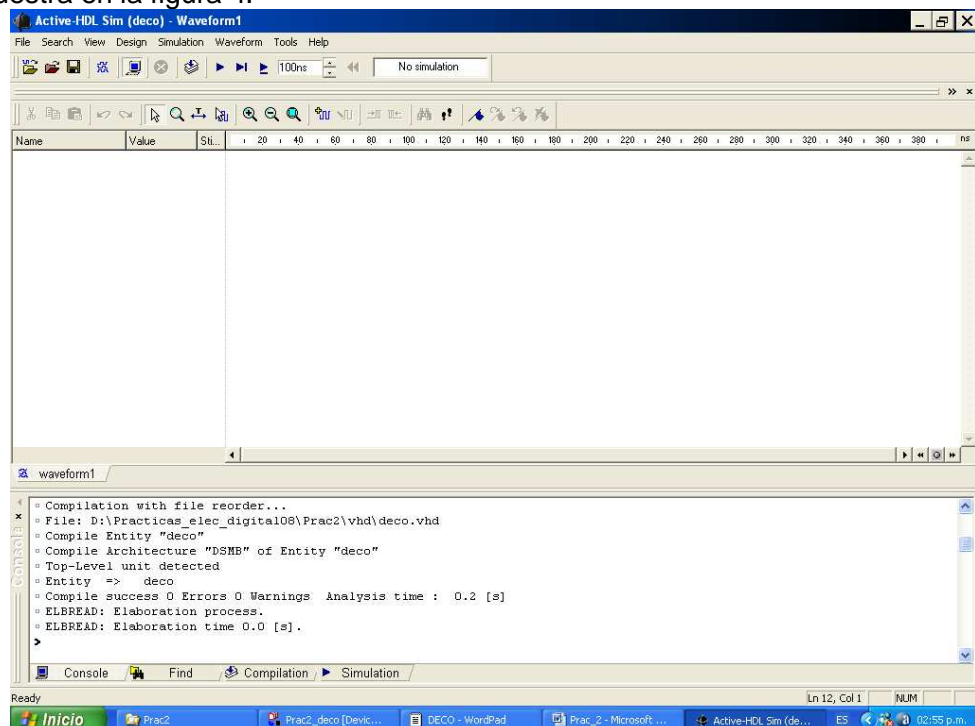


Figura 4. Ventana que muestra la compilación, en el software para la simulación.

9. Para visualizar las señales a simular es necesario agregarlas a la ventana de simulación. Para agregarlas es en: **Waveform → Add signal**.
10. Al ejecutar el paso anterior se abre una nueva ventana, como se muestra en la figura 5, y se seleccionan las señales a monitorear.
11. En este caso se seleccionan las entradas **Sel**, **a**, **b** y la salida **Sal**, para posteriormente agregarlas dando clic en el botón **<Add>**. Una vez hecho esto, aparecerán todas las señales en pantalla, como se muestra en la figura 6. Es posible ordenarlas como al usuario convenga.
12. En el caso de las señales de entrada es necesario excitar o asignar valores para su simulación. Esto se hace dando clic con el botón derecho sobre cada señal, por ejemplo la señal **b**, y dando clic en **Stimulators...**, esto hace que se abra una nueva ventana, como se muestra en la figura 7.

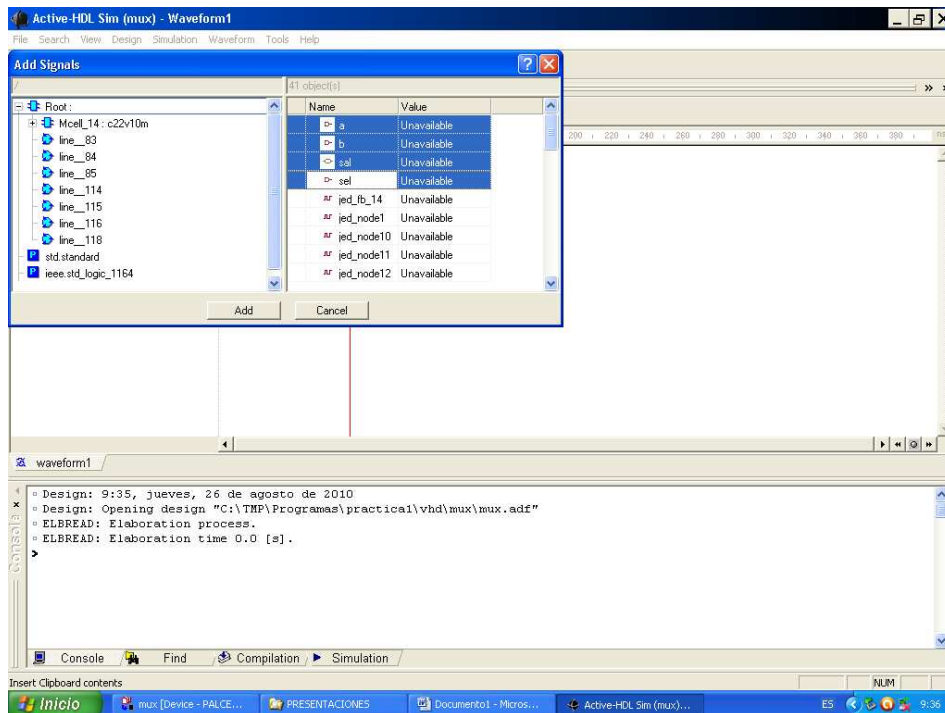


Figura 5. Ventana que muestra las señales a monitorear en la simulación.

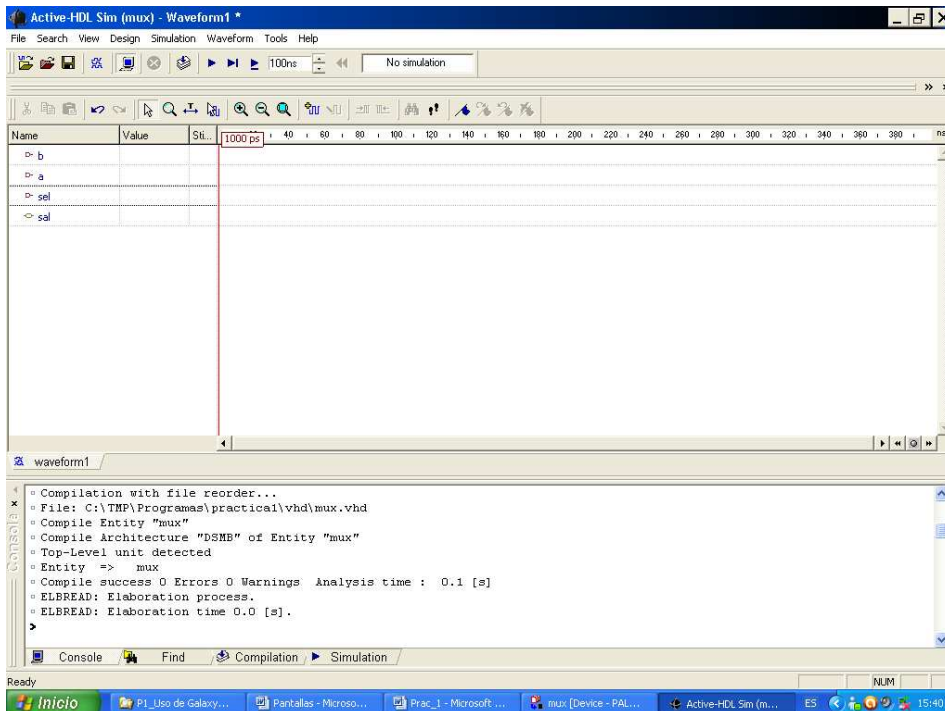


Figura 6. Ventana de simulación con las señales a excitar y monitorear

13. En la nueva ventana, se indican el tipo de estímulo, en esta caso daremos:
Stimulator type: value, Strength: Override, Force value: (valor que deseamos asignar, en este caso será **0**). Para que tenga efecto se da clic en **<Apply>**. Esto se hace en cada señal (**b, a, Sel**) y en cada asignación se da **<Apply>**, como se muestra en la figura 8.

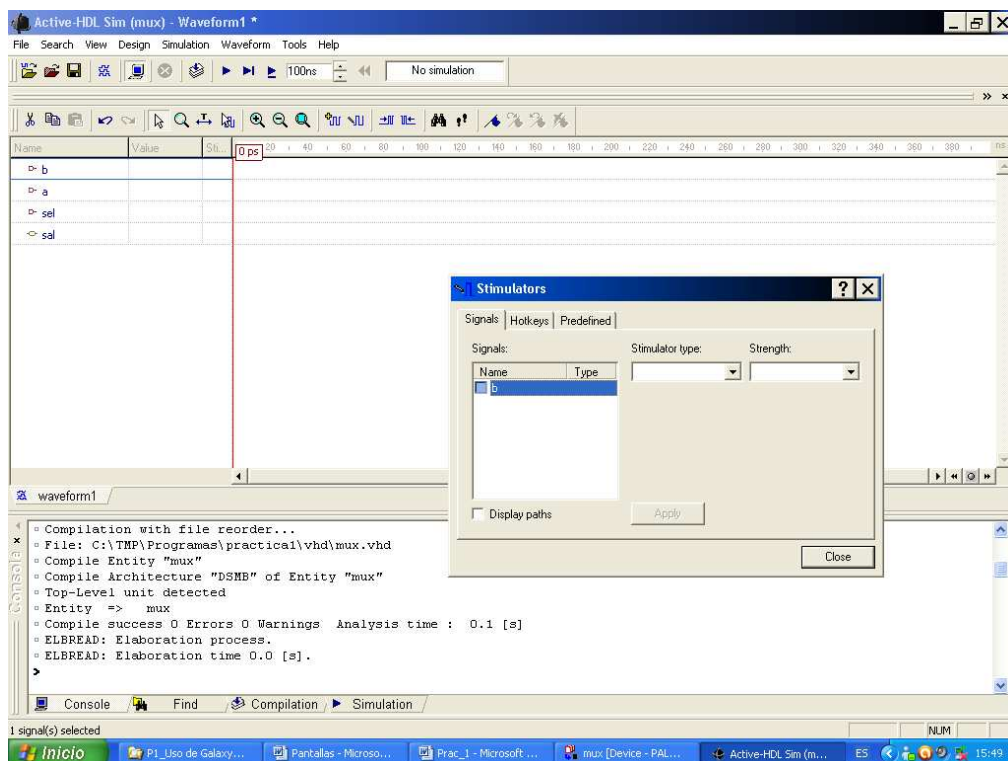


Figura 7. Ventana de simulación para asignar los estímulos a las entradas.

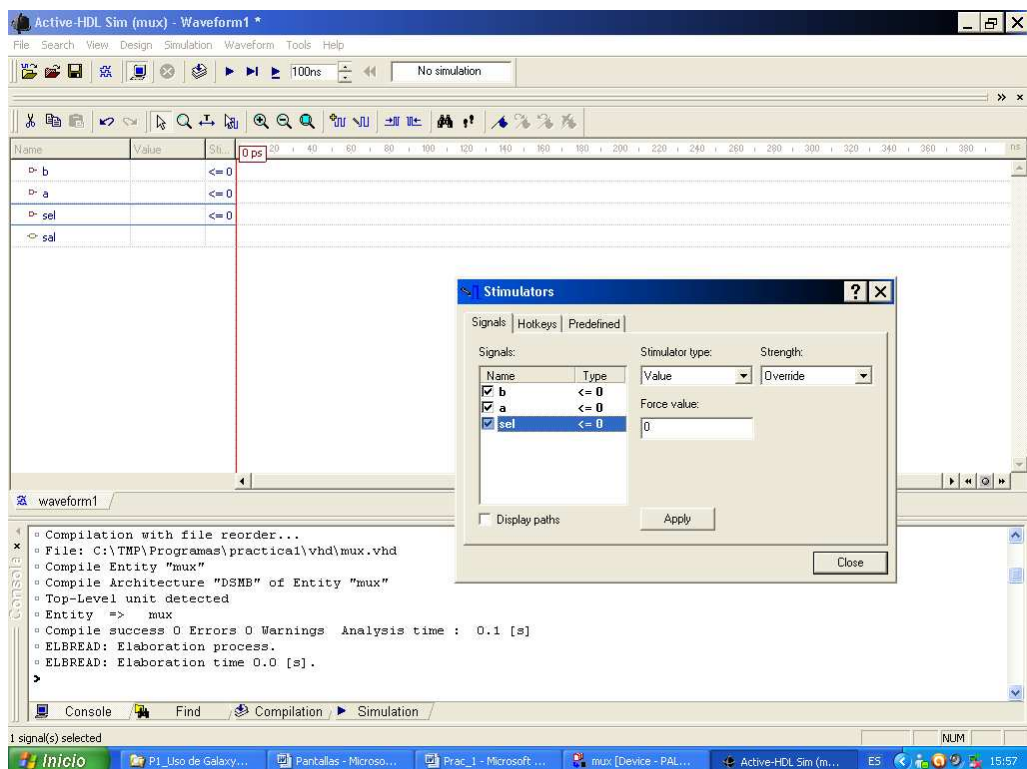


Figura 8. Ventana de simulación con los primeros estímulos asignados.

14. Para observar la simulación, con dicho estímulo asignados, es necesario ejecutar dicha opción. Esto se hace en: **Simulation → Run for o F5**. Al hacer esto se

muestran primeros valores de entrada asignados y se genera el valor de la salida (**Sal**), como se aprecia en la figura 9.

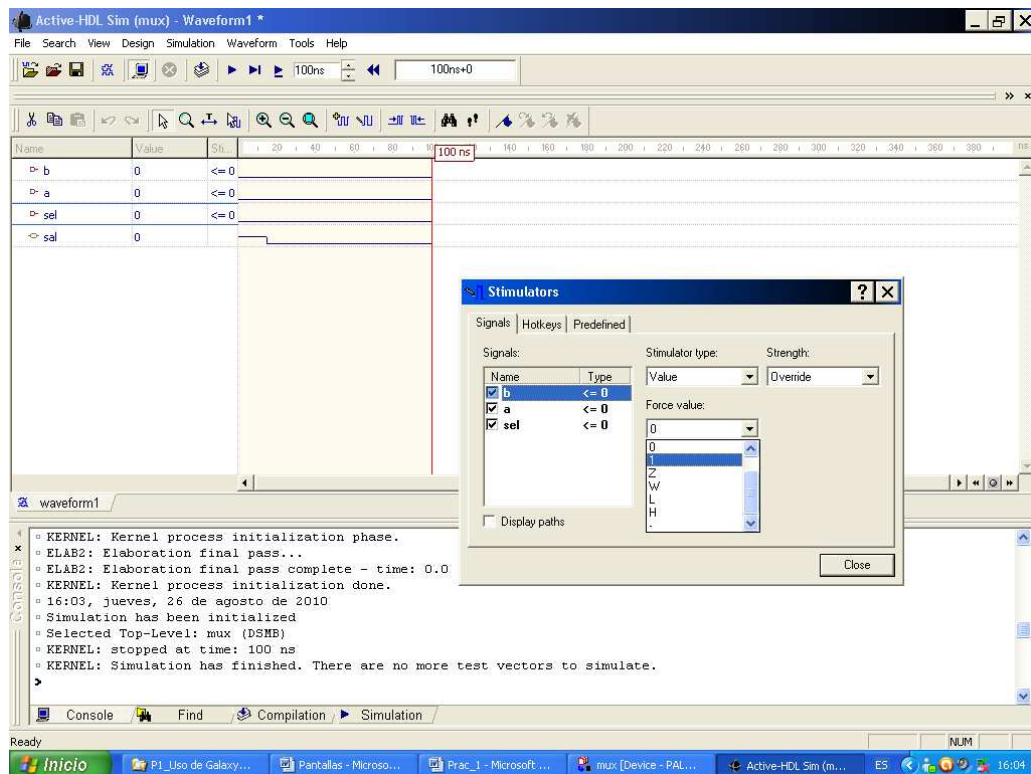


Figura 9. Ejecución de la simulación de los primeros vectores de prueba.

15. Se pueden hacer todas las posibles combinaciones de entrada, para observar si el circuito funciona correctamente. De esta manera se obtiene la figura 10.

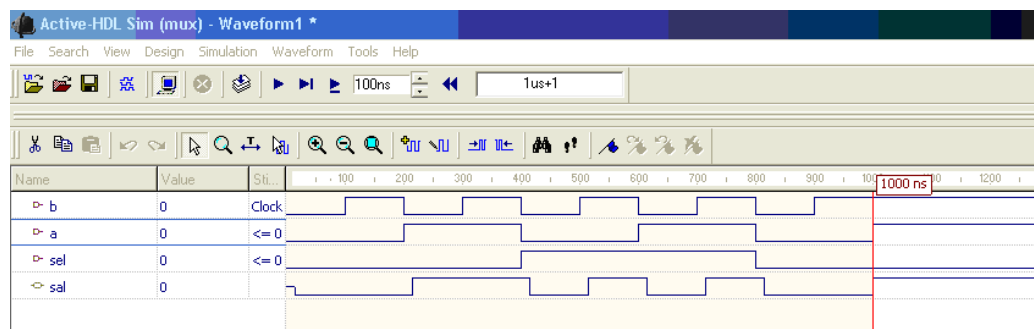


Figura 10. Simulación completa.

16. Una vez que se haya revisado el correcto funcionamiento, en simulación, del sistema diseñado es necesario saber en que terminales se encuentra cada señal de entrada/salida. El archivo para su conexión, es el reporte de salida **Output Files**, que se localiza en la ventana izquierda del proyecto, en donde se tiene el archivo **mux.vhd**. Para tener acceso al archivo de reporte se tiene que cambiar de pestaña, localizada en la parte inferior de la misma ventana, como se muestra en la figura 11. Dando doble clic en el archivo **mux.rpt** se puede observar su contenido, el cual indica las terminales asignadas a cada archivo y el uso del dispositivo, como muestra en el cuadro 2 y adicionalmente se encuentra la

información de las terminales asignadas a cada señal de nuestro proyecto, como se muestra en el cuadro 3.

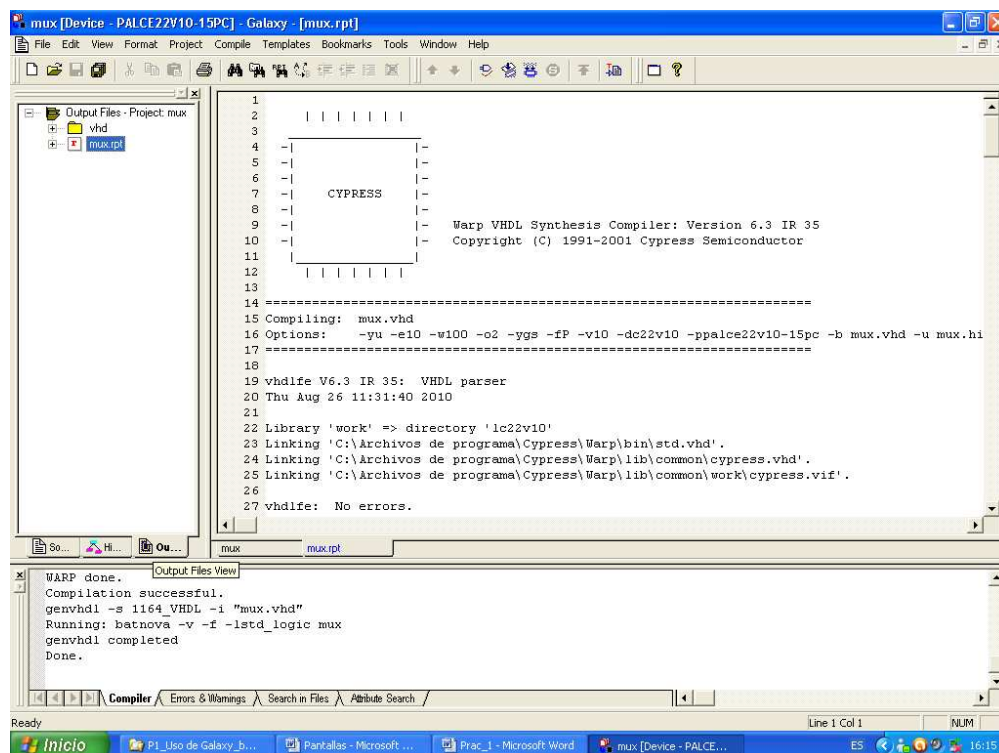


Figura 11. Reporte del diseño localizado en Output Files View.

Information: Output Logic Product Term Utilization.

Node# Output Signal Name Used Max

14	display(6)	2	8
15	display(1)	0	10
16	display(3)	0	12
17	display(0)	0	14
18	Unused	0	16
19	Unused	0	16
20	Unused	0	14
21	display(2)	0	12
22	display(5)	0	10
23	display(4)	0	8
25	Unused	0	1

2 / 121 = 1 %

Cuadro 2. Información de los términos productos que posee máximos (*Max*) y los utilizados (*Used*) por el dispositivo, para esta aplicación.

Information: Checking for duplicate NODE logic.
None.

C22V10

sel = 1	24 * not used
b = 2	23 * not used
a = 3	22 * not used
not used * 4	21 * not used
not used * 5	20 * not used
not used * 6	19 * not used
not used * 7	18 * not used
not used * 8	17 * not used
not used * 9	16 * not used
not used * 10	15 * not used
not used * 11	14 = sal
not used * 12	13 * not used

Cuadro 3. Información de las terminales asignadas automáticamente.

17. **PROGRAMACION DEL DISPOSITIVO:** Para descargar el programa al dispositivo es necesario utilizar un programador. Para esto se requiere portar el archivo que se desea programar, el cual tiene terminación jedec (***mux.jed***), e ir al programador del laboratorio. En el laboratorio se tiene un programador con un software para realizar dicho propósito. En el software se selecciona el archivo e indica a que dispositivo se va a descargar, para esto previamente se habrá insertado el dispositivo en la base del programador y se procede a programarlo.

5.2.1.4 Diagrama del circuito y prueba.

Figura 5.2. Diagrama de la interconexión del circuito 1.

5.2.2 Solución del problema 2.

(Repetir el procedimiento de edición, prueba y programación anterior).

VI OBSERVACIONES Y CONCLUSIONES

(Son individuales)

VII BIBLIOGRAFÍA

- El alumno puede consultar bibliografía y anotar la referencia. Si consulta una página de Internet ésta debe ser seria y deberá anotar dicha dirección.