UNIVERSIDAD TECNOLOGICA NACIONAL

- FACULTAD REGIONAL CORDOBA -

TECNICAS DIGITALES I

Tutorial CPLD Versión 1.

Autores:

- **Francisco G. Gutiérrez**[1]
- **Juan Manuel KIRICHIAN**[2]
- **Emilio KOWALSKI** [3]

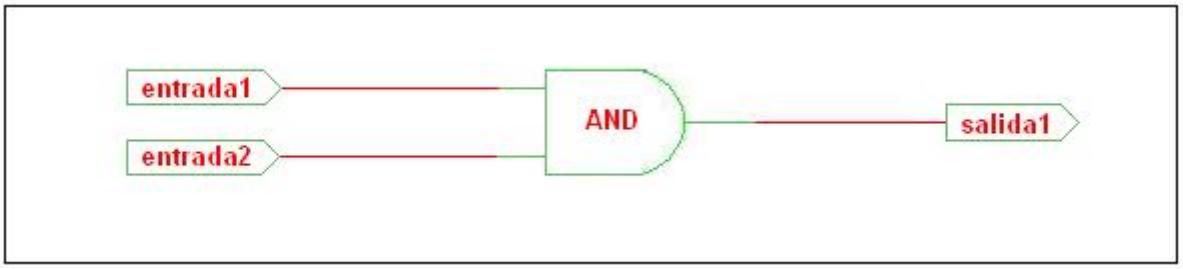
1 JTP Técnicas Digitales I
2 Alumno Técnicas Digitales I (2009).
3 Becario CUDAR

INTRODUCCION:

El objetivo de este material consiste en introducir al estudiante en la utilización de "kit de desarrollo CPLD" disponible en el laboratorio de Técnicas Digitales de la Universidad Tecnológica Nacional Facultad Regional Córdoba, esto no limita la aplicación del presente material, ya que el mismo permite tener un acercamiento rápido a la creación, simulación y transferencia de un proyecto desarrollado en VHDL a un dispositivo tipo CPLD o FPGA.

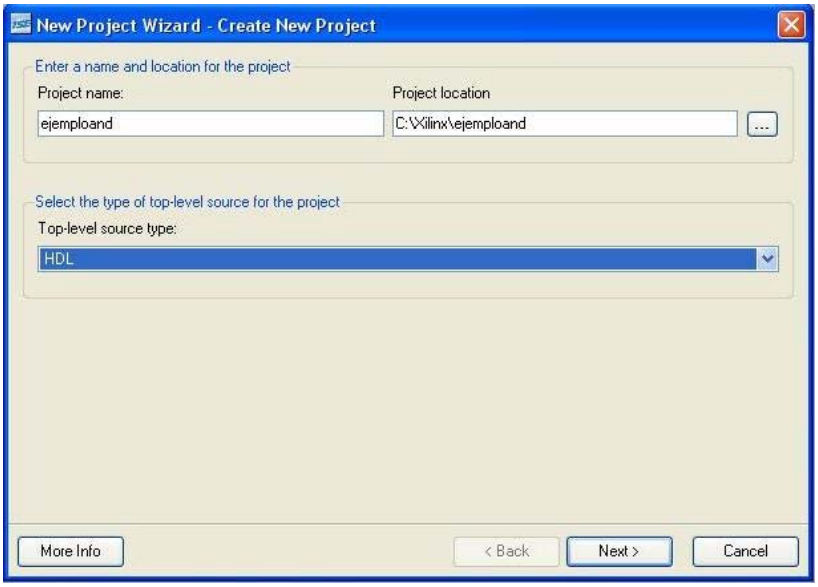
Los CPLD y FPGA disponibles para la práctica son fabricados por la empresa XILINX, debido a esto el presente material utiliza el ISE como plataforma de desarrollo.

Se presenta como ejemplo de implementación una compuerta AND. La finalidad del ejercicio es guiar al estudiante por todos los pasos necesarios para implementar la compuerta, simular el comportamiento en la PC y transferirla al CPLD. En el kit, las entradas de la compuerta serán simuladas mediante el uso de interruptores y la salida será visualizada mediante la utilización de un diodo emisor de luz (LED).



- **1er Paso: Crear un nuevo proyecto.**

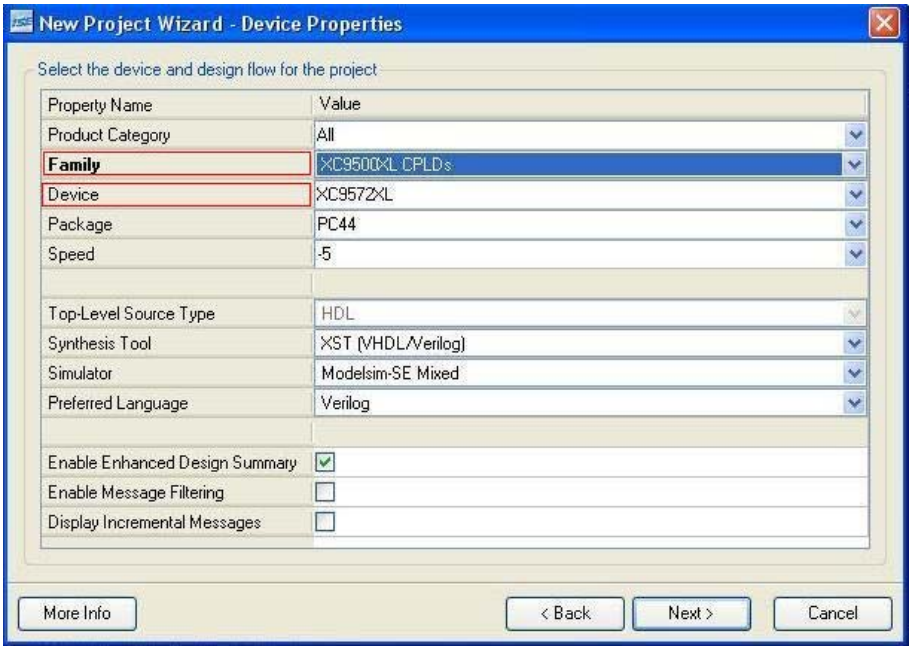
Una vez abierto el ISE, en el menú principal seleccionamos **File → New Project**. Con eso iniciamos el proceso del wizard que permitirá generar el proyecto.
En la siguiente pantalla ponemos el nombre de nuestro proyecto.
Algunos programas que utiliza el ISE durante el proceso de implementación y simulación no soportan espacios en los nombres ni en los directorios que contienen al proyecto, por lo tanto la ubicación del proyecto y el nombre del mismo NO PUEDEN CONTENER ESPACIOS NI CARACTERES ESPECIALES.



Una vez completo el nombre y la ubicación del proyecto pasamos a la siguiente pantalla en la que debemos colocar el tipo de CPLD o FPGA que vamos a utilizar. El Kit de Desarrollo esta basado en el XC9500CL. Para seleccionarlo seguimos los siguientes pasos:

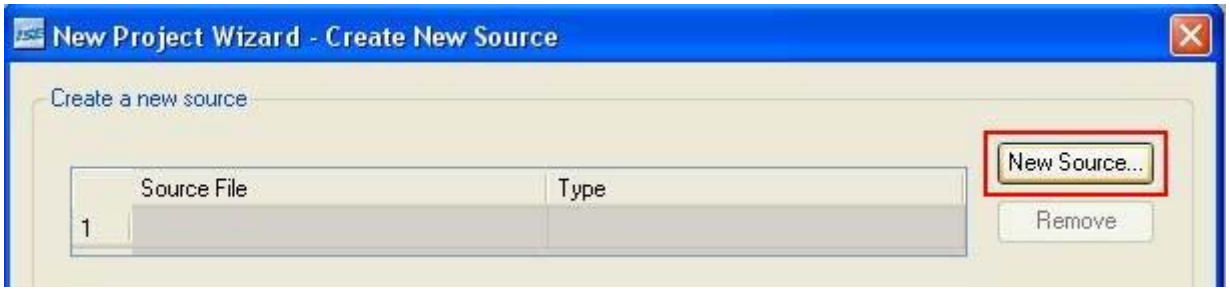
1. En **Family** buscamos la opción **XC9500XL CPLDs.**
2. En **Device** ponemos **XC9572XL.**
3. Hacemos click en **Next.**

- **2do**

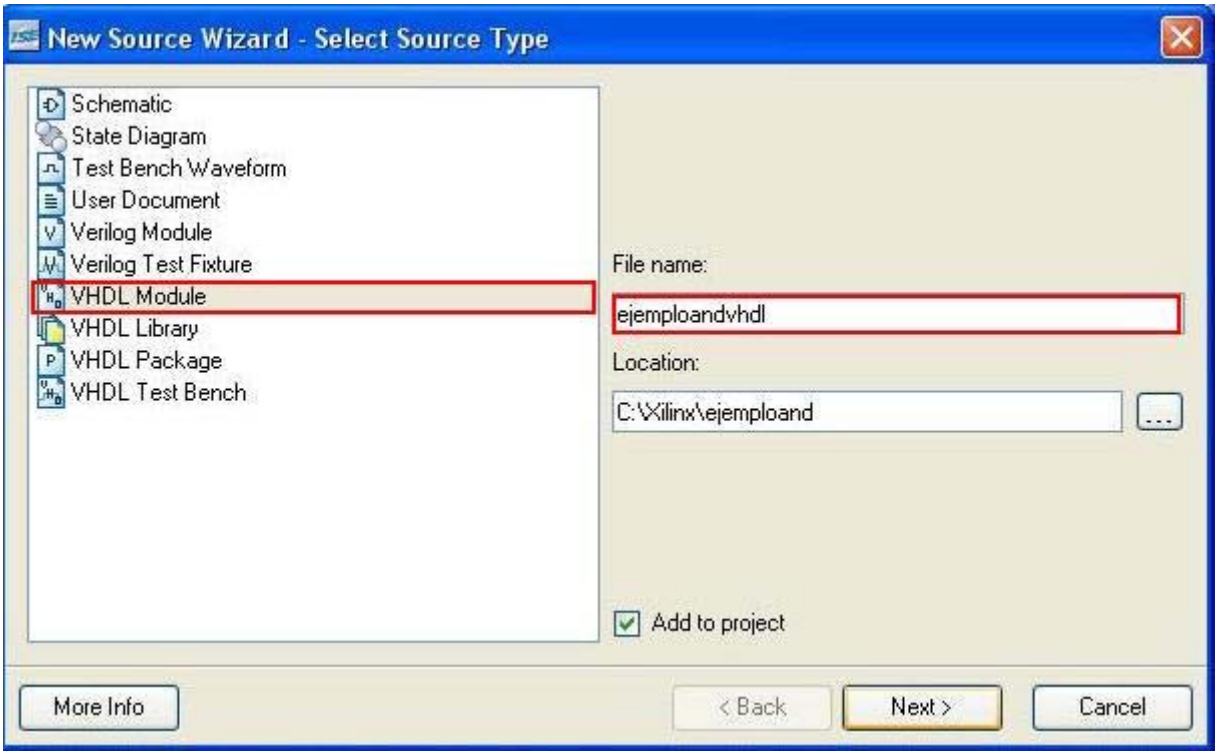


Paso: Crear una fuente VHDL y hacer la declaración de puertos

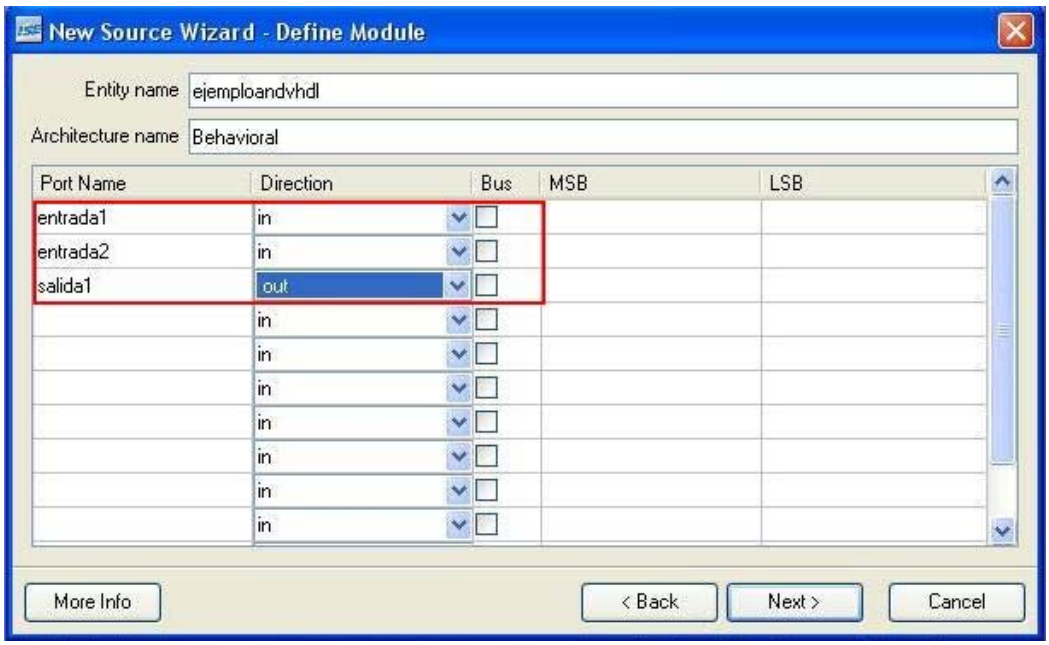
Click en **New Source**…



En la siguiente pantalla agregamos un **módulo VHD**L y el nombre del mismo, siguiendo la misma regla de evitar los espacios. Luego click en **Next**

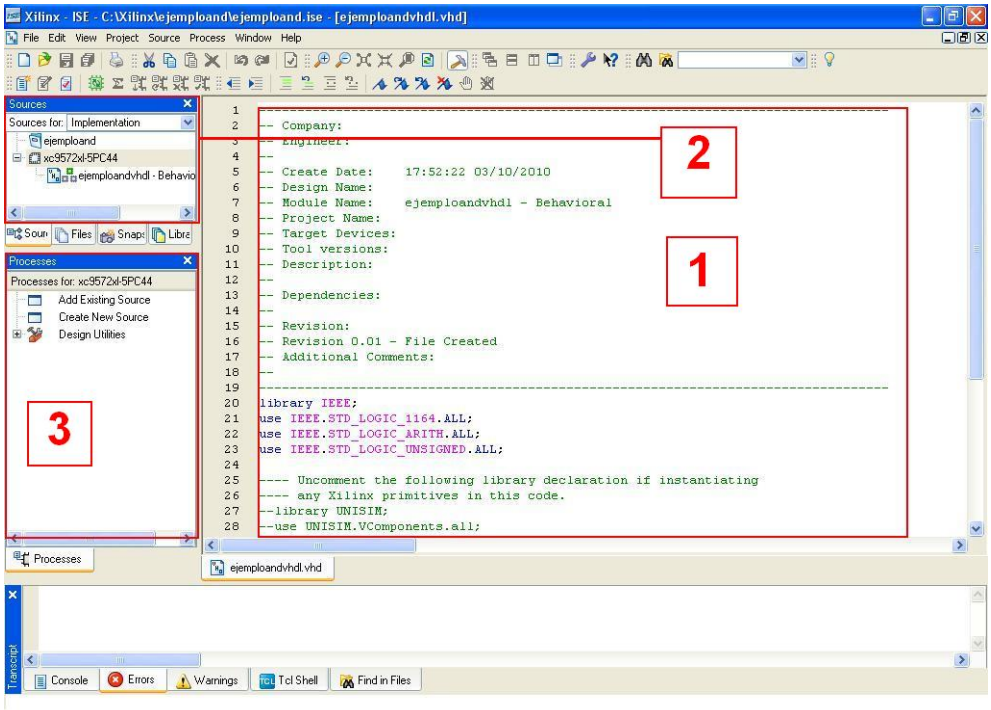


Al presionar el botón NEXT aparece la siguiente pantalla, debemos declarar los puertos a utilizar y además ingresar el tipo del mismo (entrada, salida o entrada/salida). (El casillero Bus NO se utiliza en el presente ejemplo, este casillero permite crear entradas y salidas del tipo vector).



Una vez declarado los puertos, hacemos click en **Next** y el programa genera un resumen del proyecto creado.
Hacemos click en **Finish** y el proyecto queda definido y se generara un témplate en VHDL con los datos que fuimos seleccionando durante las etapas anteriores.

En la pantalla podemos diferenciar tres grandes bloques. El 1 es el template que se genero para el proyecto, es donde se debe incluir el Código VHDL, el 2 es el menú de **fuentes** disponibles para el proyecto, y el 3 es el menú de los procesos que se le pueden aplicar a las distintas fuentes**.**



- **3er Paso: Escribir el código descriptivo de nuestra función.**

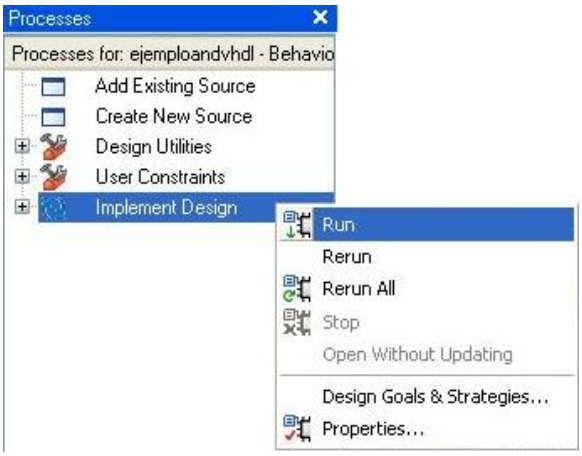
Ahora se debe describir el funcionamiento de una compuerta AND de dos entradas.
La función AND está incorporada en el VHDL así como también la OR, XOR, y NOT.
Para describir el funcionamiento de esta compuerta solo se realiza una asignación de entras a salida a través de una función lógica.
El código debe quedar de la siguiente manera.

```
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity ejemploandvhdl is
    Port ( entrada1 : in  STD_LOGIC;
           entrada2 : in  STD_LOGIC;
           salida1 : out  STD_LOGIC);
end ejemploandvhdl;

architecture Behavioral of ejemploandvhdl is

begin
 salida1<= ( entrada1 AND entrada2 );
end Behavioral;
```

Aqui se declaran las entidades

Nuestro código de una compuerta AND

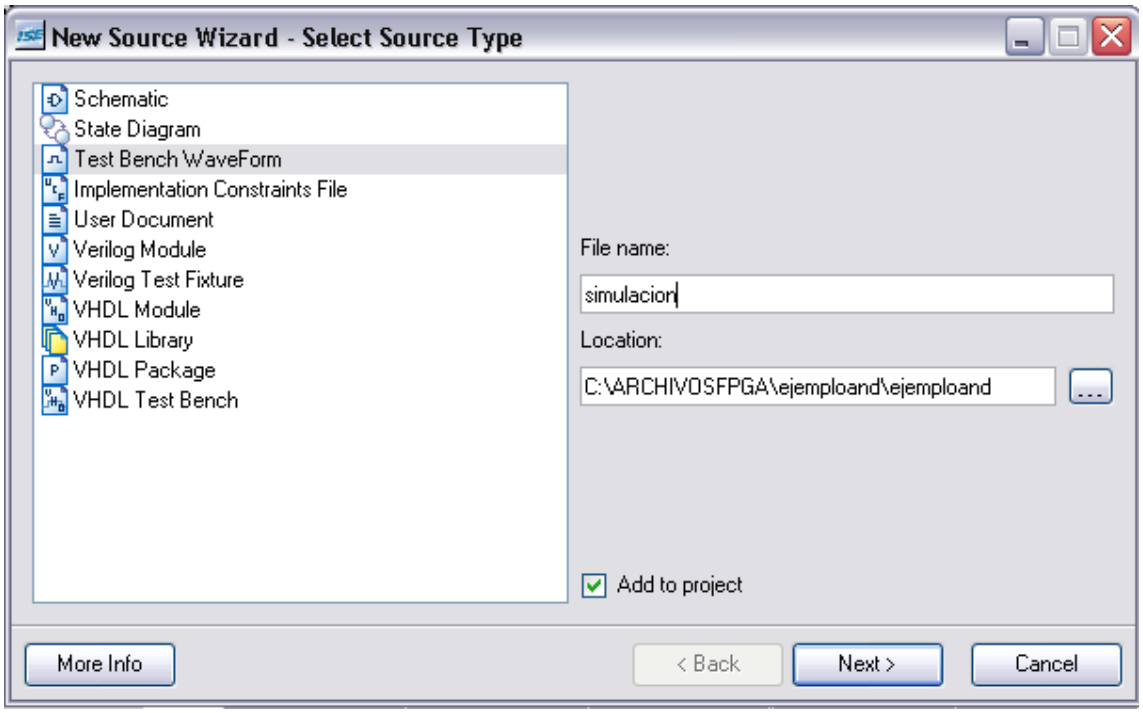
Una vez que terminado el código se lo compila e implementa en el dispositivo programable. Para ello en el menú **Processes** hacemos click derecho en **Implement Desing** y seleccionamos **Run**.



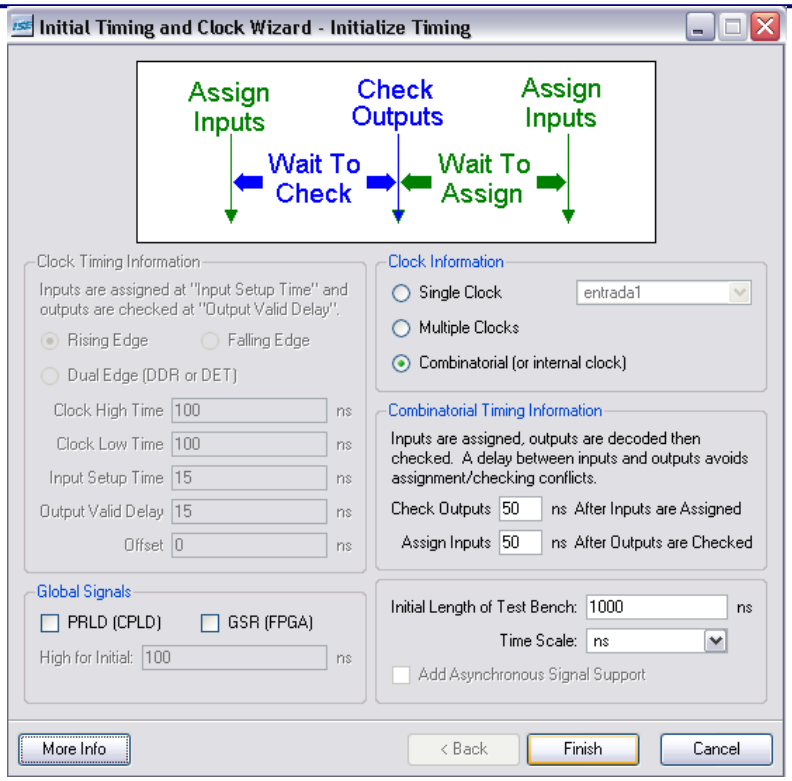
- **4to Paso: Simulación.**

Para realizar la simulación de nuestra compuerta primero debemos hacer doble click en **Create New Source** dentro de la ventana **Processes**.

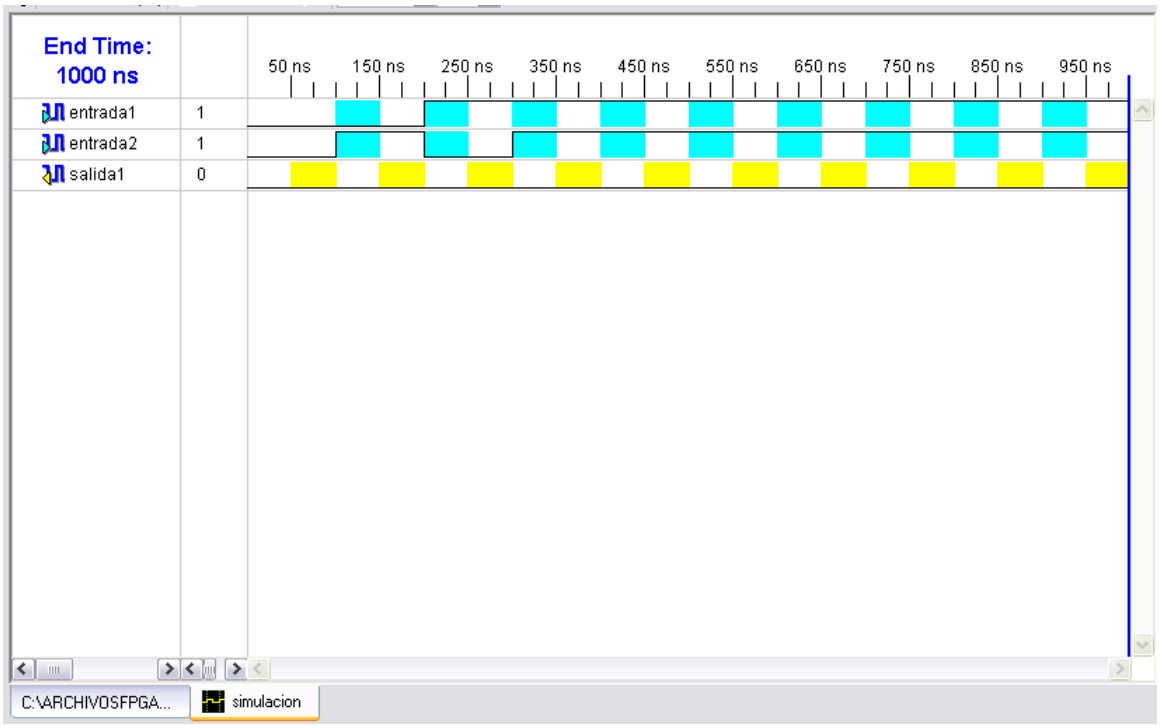
A continuación se abrirá la ventana **New Source Wizard**. Aquí debemos hacer click en **Test Bench WaveForm**, escribir un nombre a nuestro archivo de simulación y elegir la ruta de ubicación del mismo. Click en **Next** para esta ventana, la siguiente y **Finish** en la última ventana.



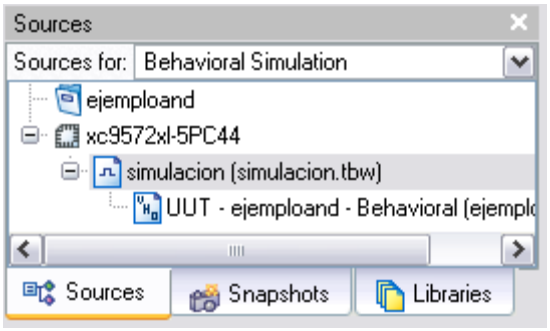
Seguidamente se deberá abrir automáticamente la ventana **Initial Timing and Clock Wizard**.



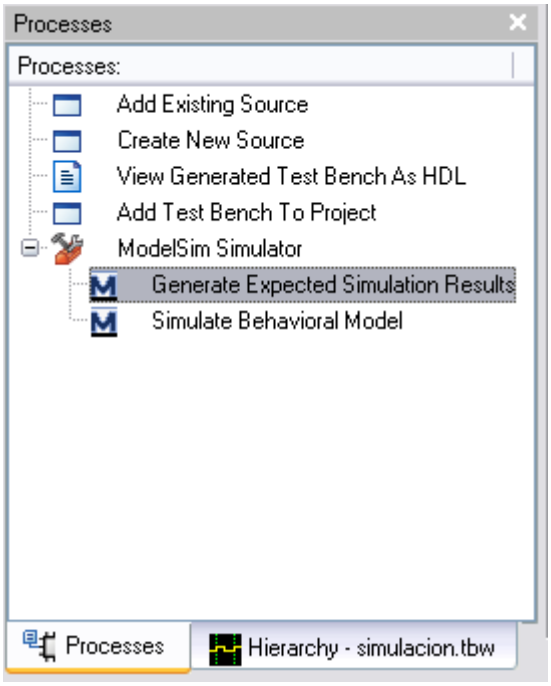
Aquí seleccionaremos dentro de **Clock Information** la opción **Combinatorial** debido a que se trata de un simple circuito combinacional. El resto de las opciones quedarán por defecto. Click en **Finish** para continuar.
El programa nos presentará en pantalla las señales de entrada y salida. Aquí debemos asignarles las formas de ondas manualmente a entrada1 y entada2, de modo de realizar todas las combinaciones posibles para la simulación. Para ello hay que hacer click sobre el área celeste definiendo el nivel de la señal a medida que transcurra el tiempo.



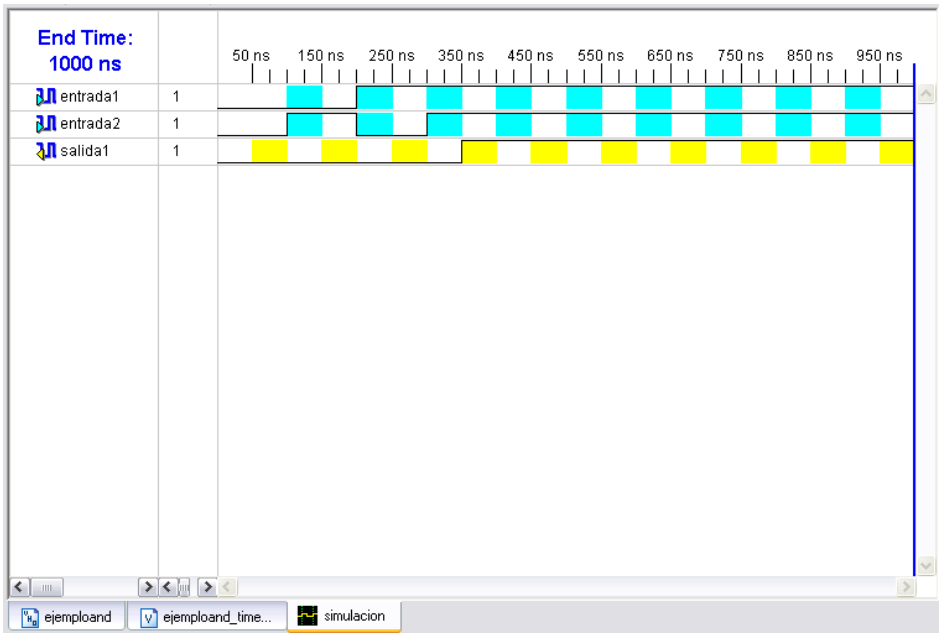
Posteriormente se debe seleccionar dentro de la ventana **Sources** la opción desplegable **Behavioral Simulation** y hacer click sobre **simulación** (o el nombre por el cual han llamado a la simulación).



Ahora se debe hacer doble click sobre **Simulate Behavoiral Model** dentro de la ventana **Processes**.



Este ultimo paso muestra el resultado de la simulación del sistema descripto en VHDL, al ser una simulación del tipo funcional (behavoiral) no tiene en cuenta los retardos producidos por las compuertas o por el mapeo del sistema en el dispositivo, esta simulación permite rápidamente ver si la descripción es correcta, pero no asegura que la misma funcionará en el dispositivo lógico programable.



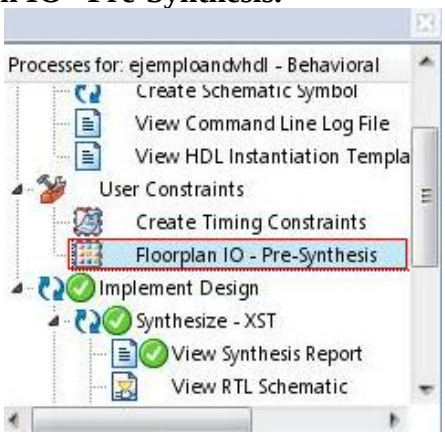
- **5to Paso: Asignación de puertos a pines reales del CPLD.**

Este paso permite mapear los pines del CPLD a los puertos definidos en nuestro diseño.
Para eso se debe modificar el archivo **ucf (users constraint file)**.
Para identificar la conexión de los pines del CPLD en el Kit se debe consultar el manual del mismo.(Pagina 20 Apéndice A.2)
En este ejemplo se utilizan dos entradas y una salida. Las entradas serán simuladas con dos llaves switch y la salida visualizada mediante un diodo Led. Entonces seleccionamos de la lista los siguientes ubicaciones:

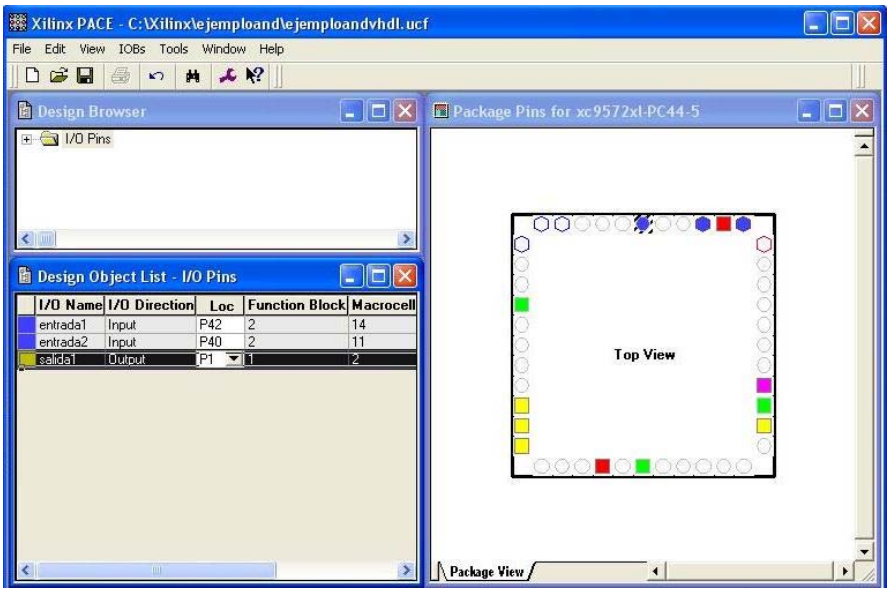
```
NET "leds<0>" LOC = "P1" ;
NET "llaves<0>" LOC = "P42" ;
NET "llaves<1>" LOC = "P40" ;
```

El texto anterior se interpreta de la siguiente manera
"NET" es una palabra clave que busca el programa de mapeo para identificar que la línea en cuestión tiene información referida a una determinada señal o puerto.
A continuación entre comillas se coloca el nombre del puerto o señal a la cual se hace referencia.
"LOC" es la palabra clave que le indica al programa de mapeo que la información corresponde a un pin físico del dispositivo y que debe llevar el puerto a ese pin.
"P1" es el pin uno del dispositivo.

El archivo USF se genera desde el menú **Processes**, seleccionamos en sub menú de **User Constraints**, la opción **Floorplan IO –Pre-Synthesis.**

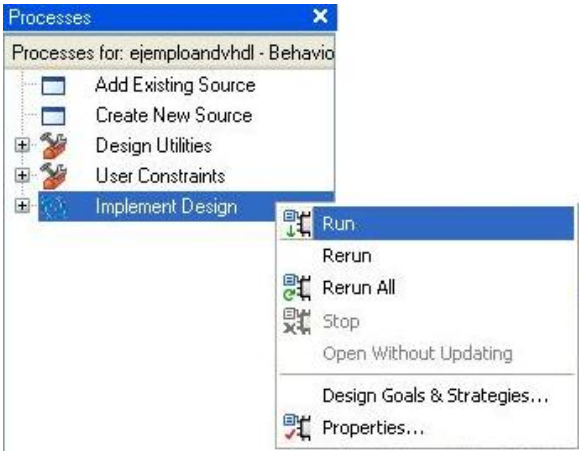


- Se nos abre una pantalla nueva en la ventana inferior izquierda ingresamos las asignaciones. Al finalizar, se guardan los cambios y se cierra la pantalla.

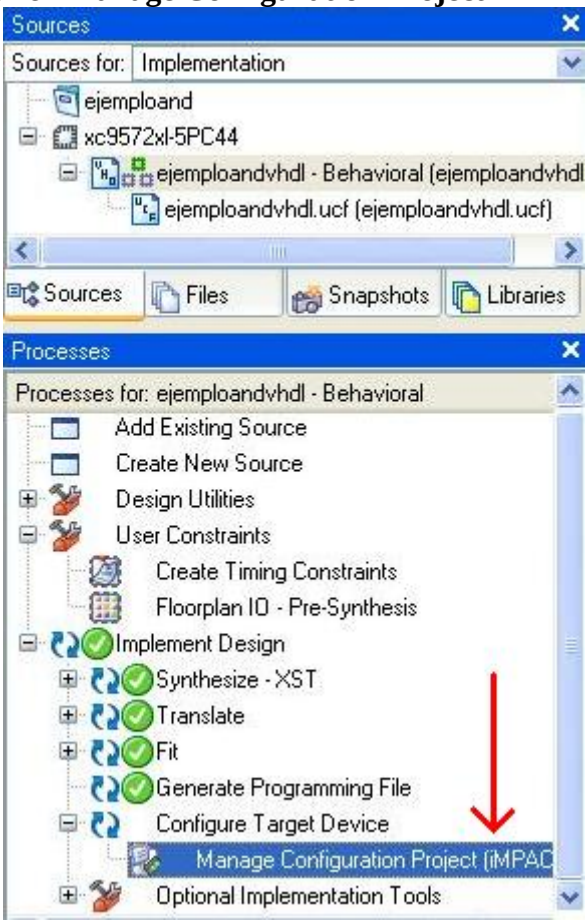


- **6to Paso: Generar el archivo de programación.**

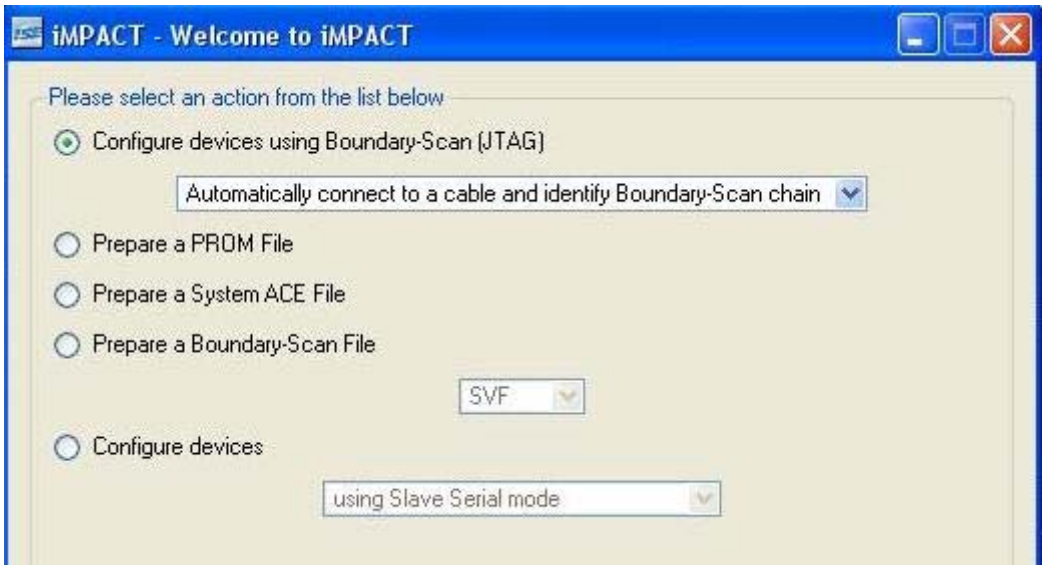
Una vez guardado el archivo **ucf**, debemos nuevamente hacer click derecho en el menú **Processes** y seleccionamos **Run**.



Una vez realizado esto dentro del menú de **Procceses**, dentro del sub menú **Configure Target Device**, hacemos doble click en **Manage Configuration Project**



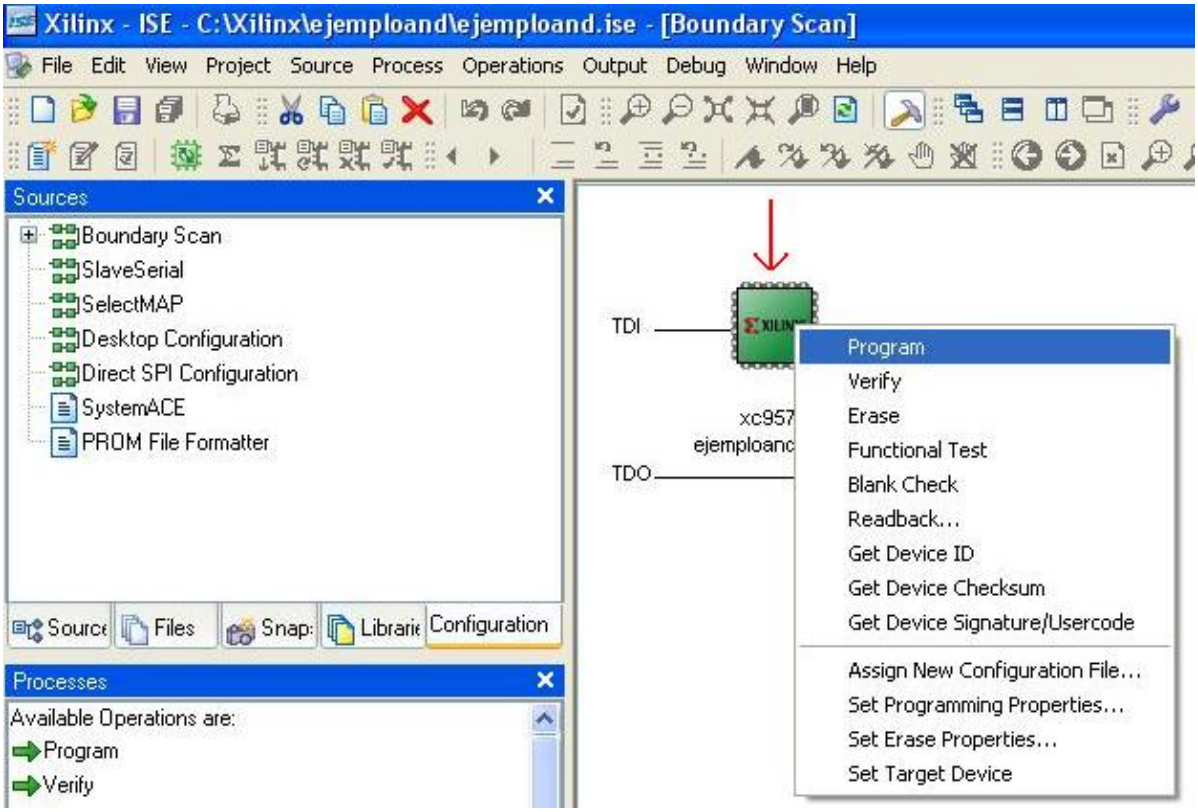
Esto nos lleva a una nueva pantalla en donde debe figurar la siguiente configuración:



Seleccionamos Finalizar y nos abre una nueva pantalla donde debemos seleccionar nuestro archivo con extensión **.jed**. El archivo se encuentra en la misma carpeta donde creamos nuestro proyecto. Lo seleccionamos y seleccionamos **Open**.

Una nueva pantalla se abre. En la misma se setean las configuraciones de programación. Sin modificar nada seleccionamos la opción **OK**.

En el siguiente paso vamos a programar el dispositivo. Como se muestra en la siguiente captura se debe hacer click derecho sobre nuestro dispositivo y seleccionar la opción **Program**.



Una vez realizado eso esperamos que nos aparezca el mensaje:

Y ya se puede operar sobre la placa de entrenamiento y verificar que lo que se programó funciona o no de la manera que uno esperaba.