



Prácticas de Fundamentos de Diseño Digital

PRÁCTICA 2. Compuertas lógicas

FECHA: _____

GRUPO: _____

ALUMNOS: _____

INTRODUCCIÓN:

Por medio de la presente práctica guiada el alumno se familiarizará con el ambiente de desarrollo ISE creando un proyecto VHDL que tenga la función de una compuerta AND. Además, utilizando la herramienta ISim del software, simulará el circuito corroborando la teoría de las compuertas.

OBJETIVO:

Conocer el lenguaje de descripción de hardware y los dispositivos lógicos programables a través de la implementación de las compuertas lógicas básicas.

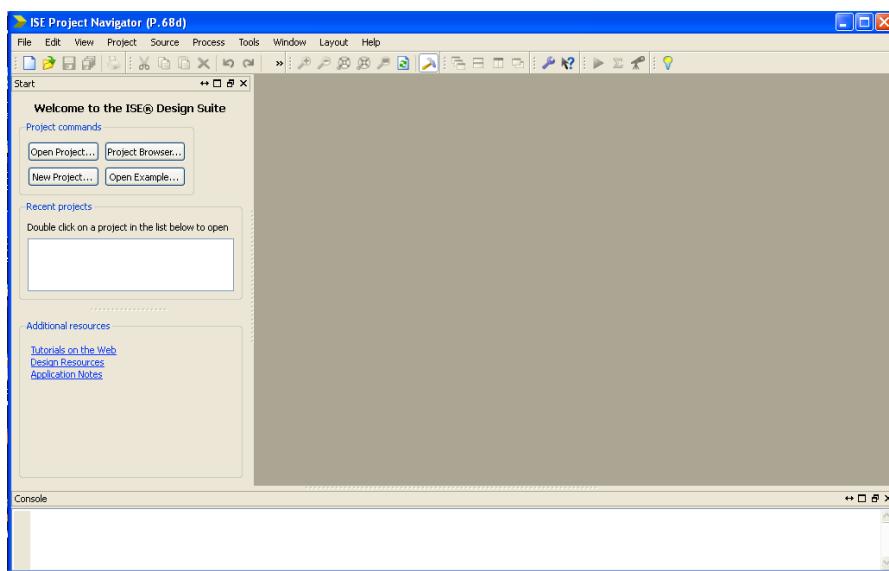
MATERIALES:

- Software ISE Project Navigator

DESARROLLO DE LA PRÁCTICA:

1. Crear nuevo proyecto.

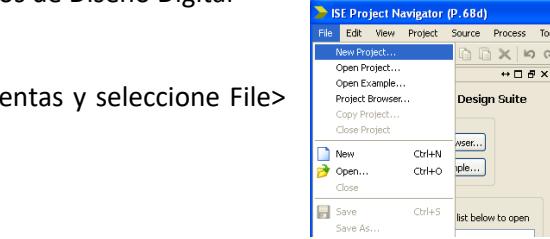
Ejecute el programa ISE Project Navigator, aparecerá la pantalla siguiente



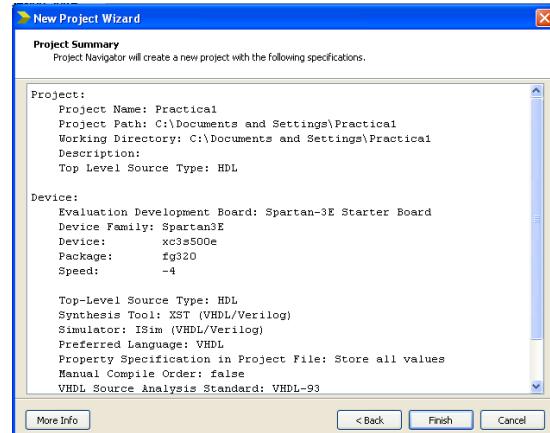


Prácticas de Fundamentos de Diseño Digital

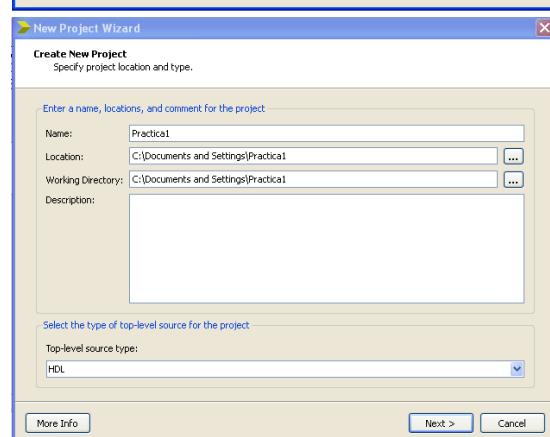
Para crear un proyecto vaya a la barra de herramientas y seleccione File> New Project.



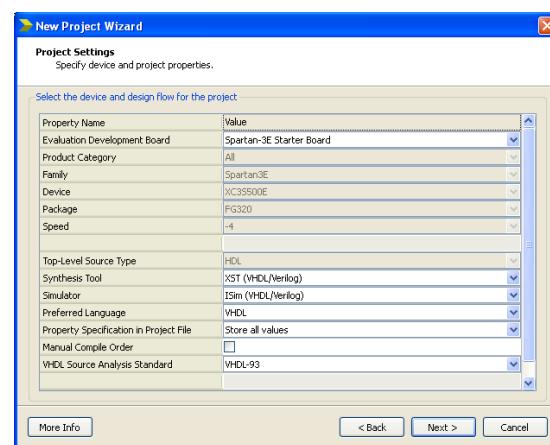
En la siguiente pantalla nombre al proyecto como “Práctica 1” en la opción *Name*, escoja la ubicación en donde se guardará el proyecto en *Location*, verifique que en la barra “Top-level source type” este la opción HDL como se muestra en la siguiente figura. Seleccione el botón *Next*.



En la siguiente pantalla que aparece tendrá que llenar los datos del FPGA que se utilizará. Para estas prácticas se utilizará el FPGA “Spartan 3E starter board kit”, seleccione esta opción en la pestaña *Evaluation development board*. Seleccione *Next*



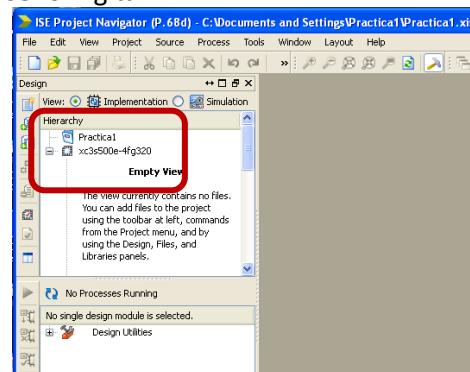
Por último, el programa desplegará un resumen de las opciones que se eligieron en la creación del nuevo proyecto. Seleccione la opción *Finish* para concluir este paso.





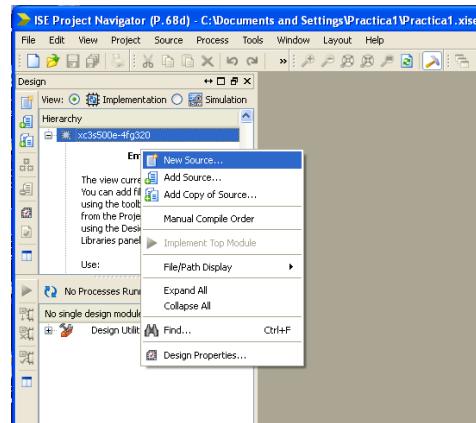
Prácticas de Fundamentos de Diseño Digital

El nuevo proyecto aparecerá en el cuadro izquierdo en el recuadro “Design” con el nombre que eligió, para este caso observamos “Practica 1”

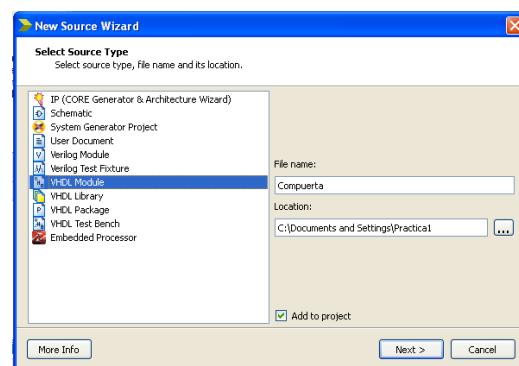


2. Crear un archivo VHDL

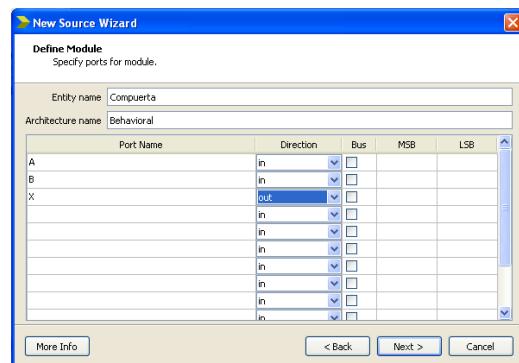
Presione click derecho sobre el dispositivo agregado al proyecto (XC3S500e) y seleccione “New Source”, o podemos ir a la barra de herramientas y seleccionar *Project> New Source*.



En esta ventana se muestran los diferentes tipos de fuentes, para esta práctica seleccionaremos *VHDL Module* con el nombre *COMPUERTA*, después marque la casilla *Add to Project* y presione *Next*.



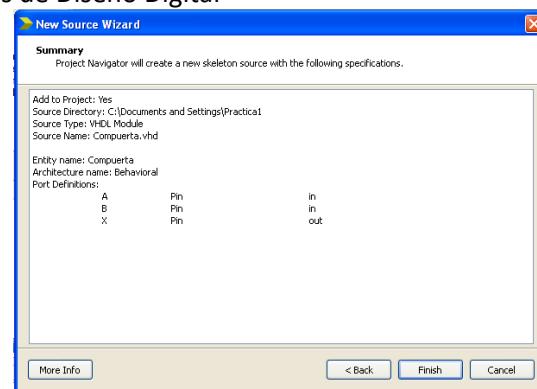
Ahora se tiene que definir los puertos de entrada y salida de nuestra fuente, esta sección generará la entidad del proyecto con los datos especificados en las pestañas. Para esta práctica asigne dos entradas (A y B) y una salida (X). Como las entradas y salidas son de un solo bit, no se habilita la opción de BUS. Al finalizar seleccione Next.



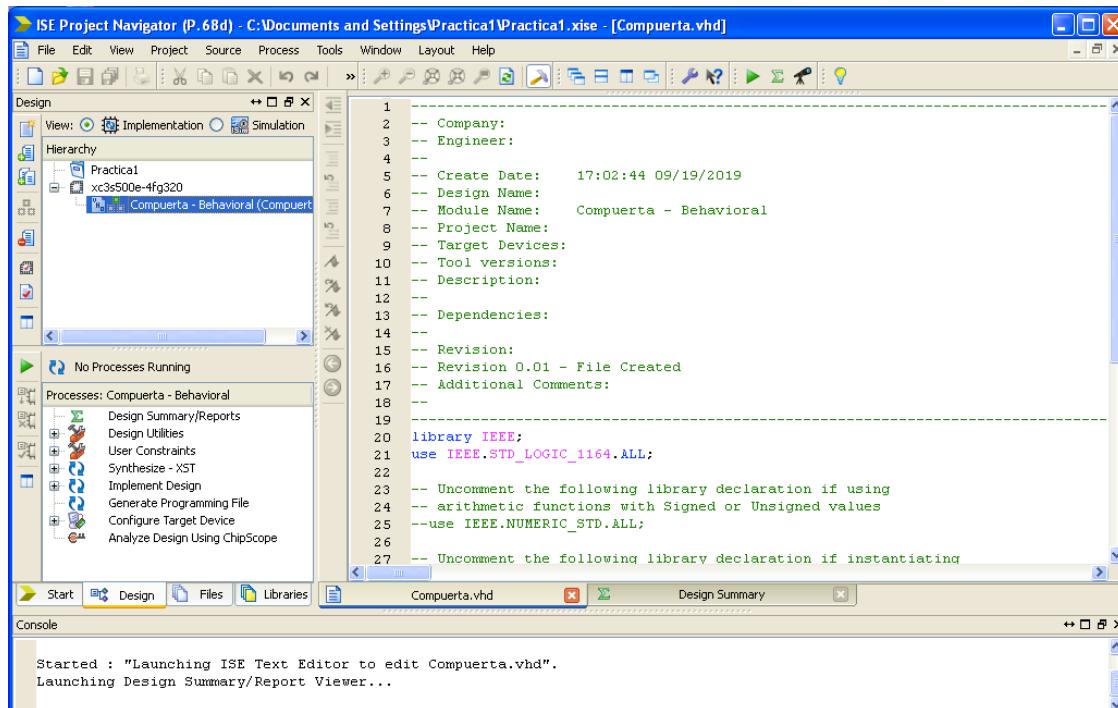


Prácticas de Fundamentos de Diseño Digital

La pantalla mostrada a continuación arroja el resumen de selecciones para nuestro archivo VHDL. Seleccione *Finish*.



Observamos que al proyecto “Práctica 1” se le ha asociado un archivo VHDL llamado compuerta, que se abre automáticamente con el código generado a manera de plantilla. En esta plantilla se encuentran las librerías principales, la entidad generada y las palabras reservadas para la arquitectura.





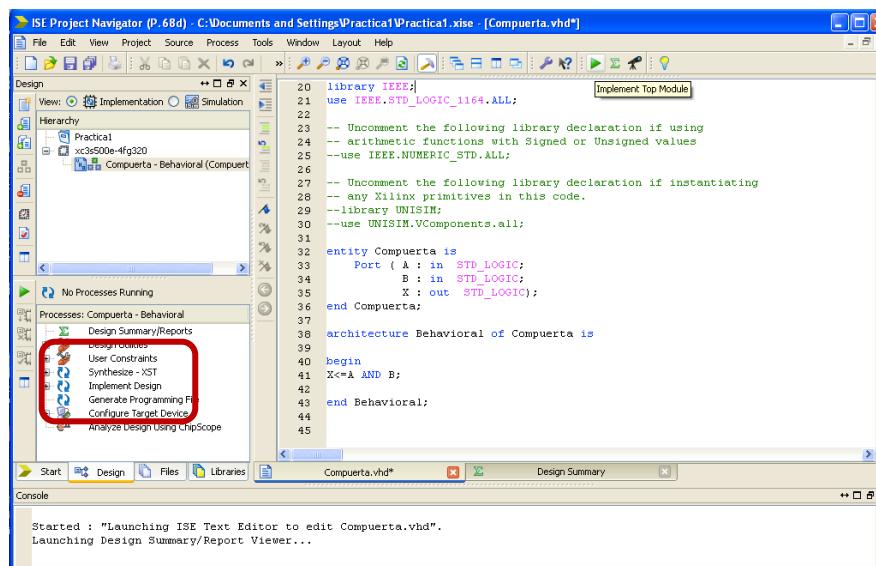
Prácticas de Fundamentos de Diseño Digital

3. Editar código.

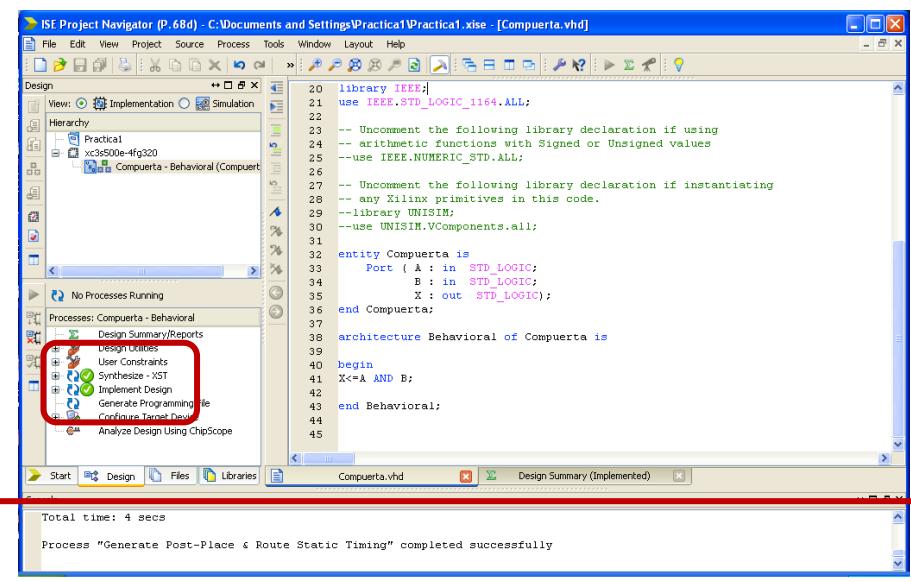
Agregue el código para el diseño de una compuerta AND entre BEGIN y END de la arquitectura y guarde los datos. El código para una AND de dos entradas A y B y salida X es:

$$X \leq A \text{ AND } B;$$

Para compilar el trabajo y revisar que no existan errores, presione click en el ícono de *Implement Top Module*, esta herramienta ejecutará los procesos *Synthesize-XST* e *Implement Design*



Si el proyecto no contiene errores se observarán los procesos mencionados anteriormente con un círculo verde a su lado y en la ventana inferior llamada Consola, se desplegará que el proyecto fue completado satisfactoriamente como se muestra a continuación



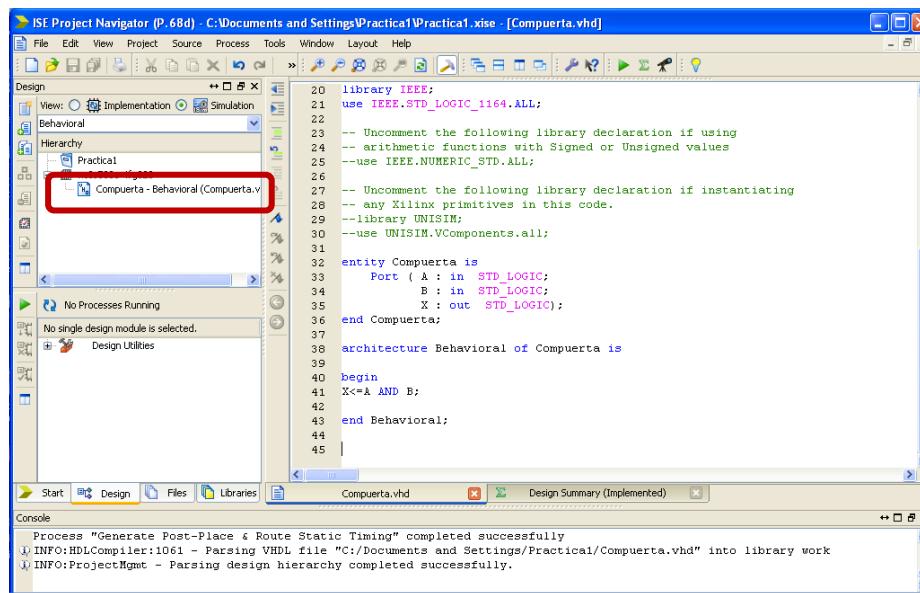


Prácticas de Fundamentos de Diseño Digital

Si existiera algún error se desplegarían en la misma ventana de consola, por ejemplo, si faltara un ";" en la descripción, al compilar se genera un error y se indica en qué línea se encuentra. Es importante depurar errores ya que de otra manera no se podrá simular o descargar al FPGA.

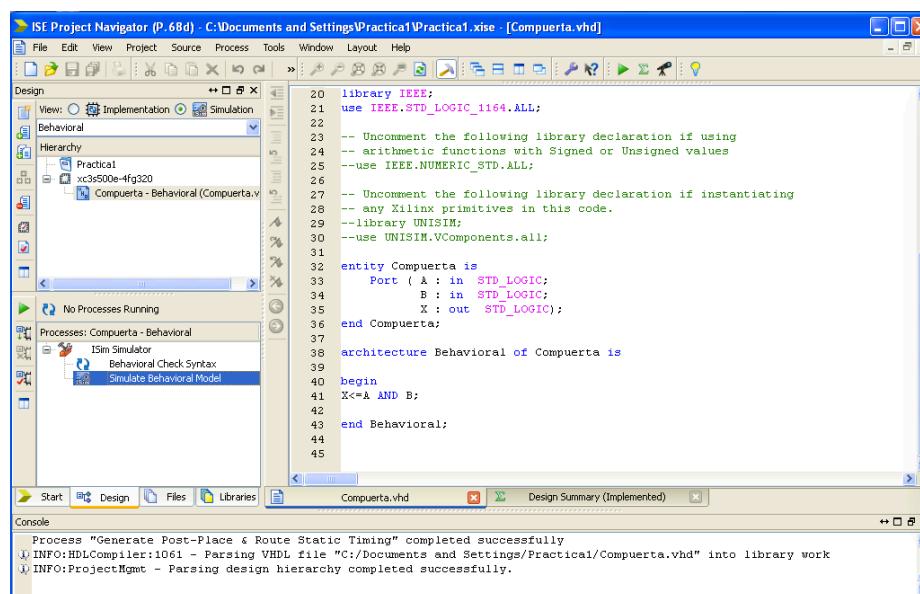
4. Simule el funcionamiento del circuito

En la ventana de diseño (esquina superior izquierda) seleccione la opción de Simulación para que el programa despliegue la ventana de fuentes a simular. En ésta ventana seleccione el módulo VHDL **COMPUERTA** que anteriormente se asoció al proyecto.



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity Compuerta is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           X : out STD_LOGIC);
end Compuerta;
architecture Behavioral of Compuerta is
begin
    X<=A AND B;
end Behavioral;
```

Dé click derecho en *Simulate Behavioral Model> Run* para que se ejecute ISim

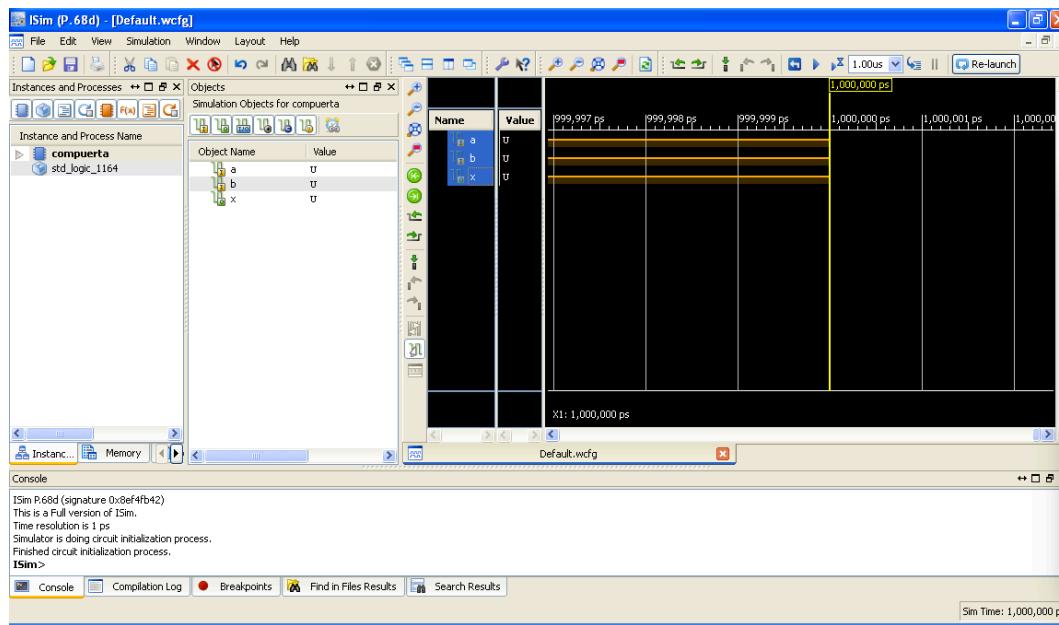


```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity Compuerta is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           X : out STD_LOGIC);
end Compuerta;
architecture Behavioral of Compuerta is
begin
    X<=A AND B;
end Behavioral;
```

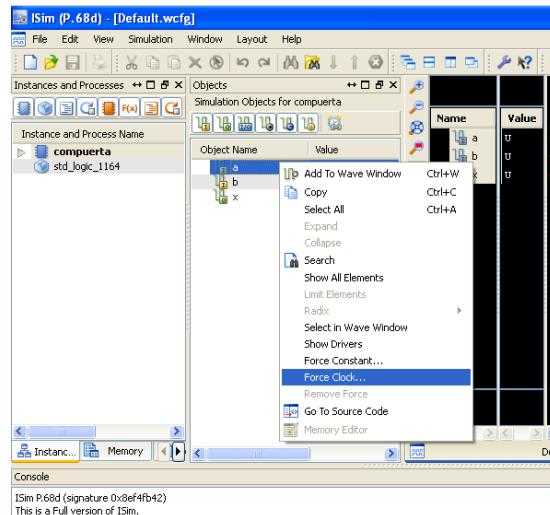


Prácticas de Fundamentos de Diseño Digital

Después de cargar los datos necesarios, se abrirá el ISim, en esta ventana se observa los puertos declarados en el módulo VHDL, así como sus propiedades



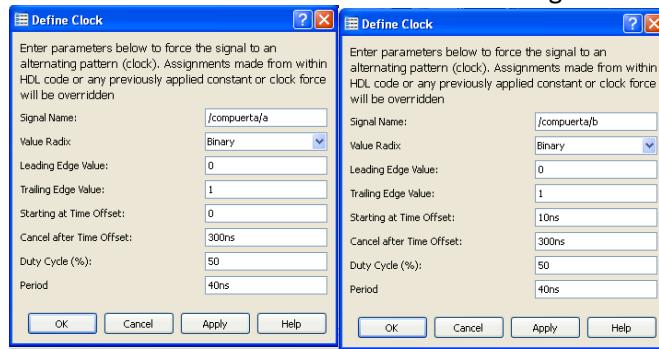
Para comprobar que el proyecto funciona como se diseñó, tendremos que simular con los datos de la tabla de verdad de la compuerta AND, para lograr esto se utilizarán dos señales de reloj desfasadas con un periodo de 40 ns, ciclo de trabajo del 50%. Las características de las señales se ingresan en el ISim dando click derecho sobre cualquiera de las señales de entrada y seleccionando Force Clock.



Proceda a configurar las características de cada señal. Las dos señales tendrán periodo de 40ns y un ciclo de trabajo del 50%, la segunda señal iniciará 10ns después para poder apreciar los valores de salida en la última señal, el análisis terminará en 300ns.



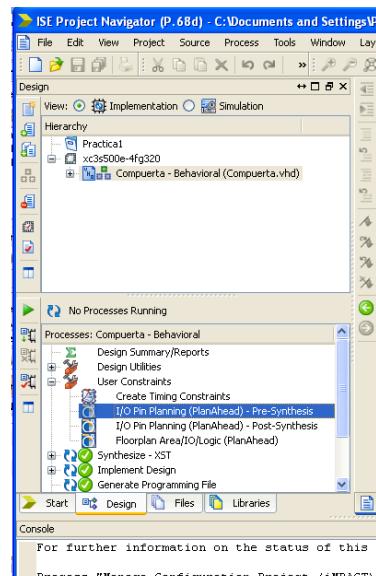
Prácticas de Fundamentos de Diseño Digital



Finalmente se obtiene la simulación de las dos señales de entrada y la de salida, donde podemos apreciar que en la señal de salida se cumple la tabla de verdad de la compuerta AND.

5. Asignación de pines del proyecto “Practica 1” con el sub-programa “PlanAhead” de la ISE Design Suit de Xilinx.

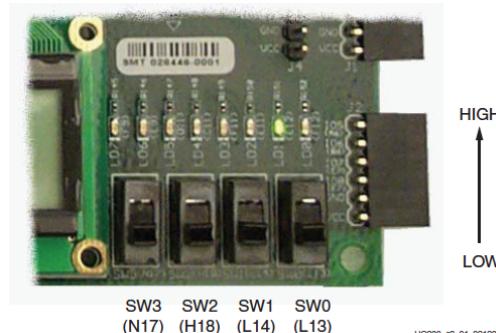
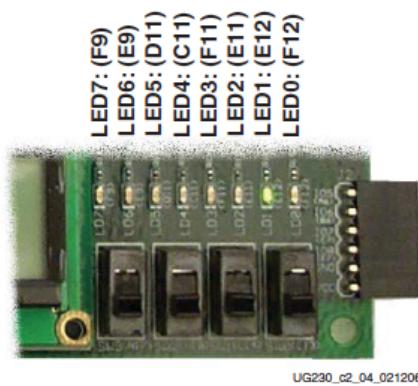
Se hará la asignación de pines de entrada y salida, con la opción “User Constraints”. Despliegue el submenú *User Constraints* y presione doble click en “I/O Planning (PlanAhead)- Pre-Synthesis”.



Se abrirá el programa PlanAhead, donde podremos seleccionar los pines de entrada y salida. Para elegir los pines de entrada y salida, se expandirá el menú “Scalar Ports” donde estarán listadas las entradas y salidas de nuestro circuito. Para la elección de entradas y salidas nos basaremos en el diagrama de la tarjeta Spartan 3E starter kit que se puede encontrar en el manual de usuario, en el cual se listan todos los puertos y periféricos disponibles. Para esta práctica elegiremos como entradas los interruptores SW0 y SW1 y como salida en led LED0.

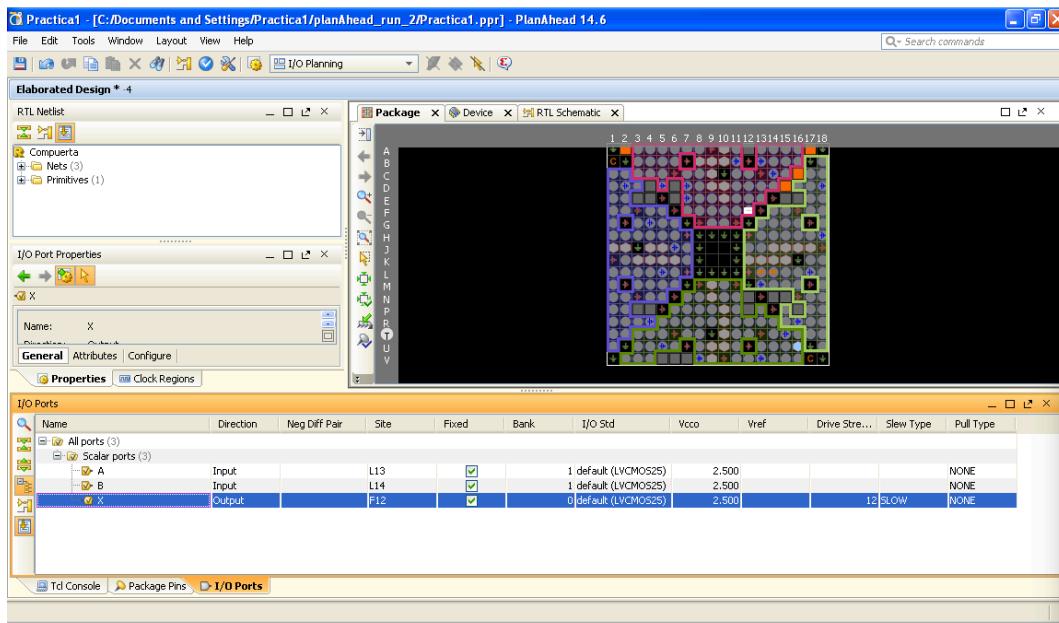


Prácticas de Fundamentos de Diseño Digital

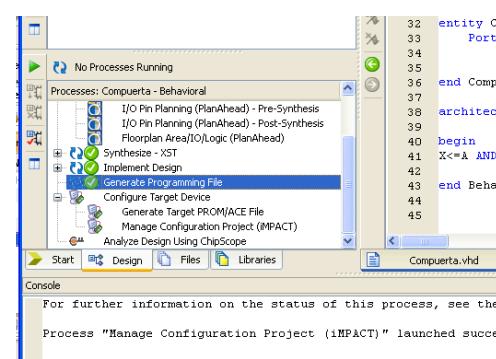


Entrada A: L13, Entrada B: L14, Salida X: F12

Mapeamos estos pines a la ventana de *PlanAhead* en la columna SITE, para cada entrada y salida



Guarde los cambios dando click en *File> Save Constraints*. Después de guardar el diseño de entradas y salidas, vuelva a *ISE Project Navigator*, y utilice *Run Top Module* para verificar que el proyecto no tenga errores de diseño. Después de una implementación exitosa generamos el programa .BIT, dando click en “*Generate Programming File*”. Si no hay errores, el archivo .BIT se llamará como el archivo fuente y se almacenará en la carpeta del proyecto.

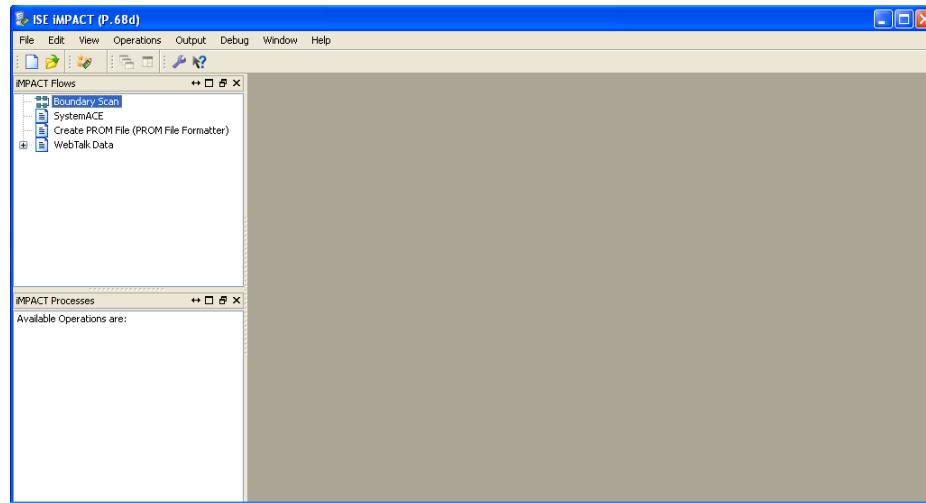




Prácticas de Fundamentos de Diseño Digital

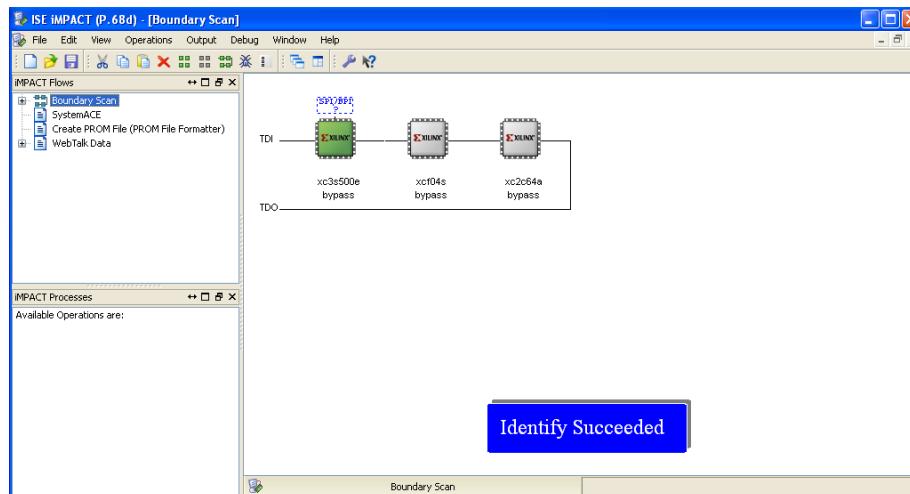
6. Cargue el programa en el FPGA

Para descargar el programa *compuerta.bit* al FGPA, este deberá estar conectada por USB a su computadora con el interruptor de encendido en la posición ON. Ahora dé doble click sobre la opción “*Manage Configuration Project (iMPACT)*” la cual abrirá la siguiente ventana:

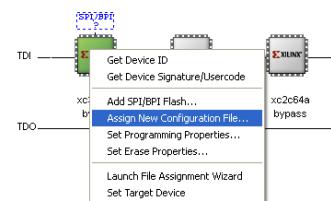


En esta ventana presione doble click en “*Boundary Scan*” y en seguida presione el botón “*Initialize Chain*” con el ícono

, cierre todas las ventanas de dialogo que puedan emerger. Si la tarjeta está bien conectada a la computadora, se desplegará el mensaje: “*Identify Succeeded*”. Además, en esta ventana se desplegarán los dispositivos programables disponibles en la tarjeta de desarrollo Spartan 3E. Nos enfocaremos en el dispositivo xc3s500e.



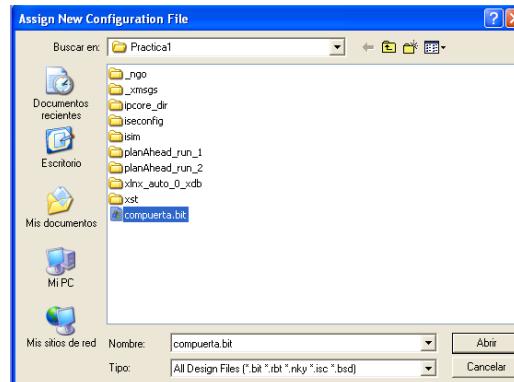
Dé click derecho sobre el dispositivo xc3s500e y seleccione la opción “*Assign New Configuration File*”





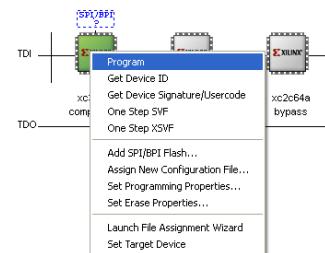
Prácticas de Fundamentos de Diseño Digital

Seleccione el archivo con extensión “.bit” que se generó en pasos anteriores, éste deberá estar en la carpeta del proyecto, cierre todas las ventanas de diálogo que puedanemerger.



Dé click derecho sobre el dispositivo xc3s500e y seleccione la opción “*Program*”, esto cargará la configuración con el programa a la tarjeta FPGA.

Si el programa se ha cargado de manera correcta, en esta ventana aparecerá el mensaje “*Program Succeeded*”.



Compruebe la tabla de verdad de la compuerta AND cambiando los estados de los interruptores seleccionados en la tarjeta. Deberá encender el led sólo cuando el interruptor SW0 y SW1 estén cerrados.

Ejercicio:

Realice la programación para las compuertas OR, NOR, NAND y XOR. Compruebe su funcionamiento por medio de simulación y directamente en la tarjeta por medio de los interruptores.

Agregar a reporte:

- Simulación de cada compuerta lógica
- Fotos de evidencia de la implementación en el FPGA.