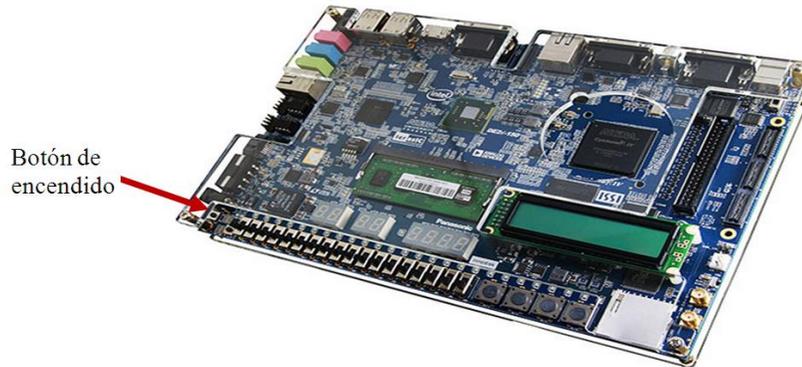


Manejo del FPGA de la tarjeta Intel DE2i-150 en la plataforma de diseño QUARTUS

M.I. Nortma Elva Chávez Rodríguez

Pasos para trabajar en la tarjeta intel DE2i-150 FPGA:

Lo primero que se debe hacer es prender la tarjeta y esperar a que se cargue el sistema operativo “UBUNTU”. Ver la siguiente figura:



Posteriormente se crea una carpeta en el escritorio con el nombre del proyecto a realizar y ejecutar los siguientes comandos que se llevan a cabo dentro del menú “terminal.” Y escribir:

```
sudo su root---- enter
```

```
Password: padrecestial---- enter
```

```
cd altera ---- enter
```

```
cd 12.1 ---- enter
```

```
cd quartus ---- enter
```

```
cd bin ---- enter .
```

```
./quartus ---- enter .
```

Nota: Espere un momento ya que tarda al iniciar.

1 HERRAMIENTAS DE DISEÑO “QUARTUS”

La plataforma QUARTUS integra herramientas de desarrollo necesarias para procesar diseños en forma amigable e incluso manejar proyectos jerárquicos. También cuenta con métodos poderosos de síntesis lógica, compilación, partición, simulación funcional, simulación en tiempo y simulación enlazada con varios dispositivos.

Para el manejo de esta plataforma se requiere introducir un diseño, sintetizarlo y finalmente configurarlo y grabarlo en el dispositivo seleccionado; sin embargo, es recomendable, antes de crear el diseño, que éste se simule y se analice en el tiempo.

2 PROYECTO

Un proyecto contiene todos los archivos de la jerarquía de un diseño. Los módulos que contenga un proyecto son llamados fuentes. La plataforma QUARTUS realiza la compilación, síntesis, simulación, análisis en el tiempo y programación de un dispositivo en un solo proyecto. Para compilar un proyecto con archivos independientes se debe especificar, primero, qué archivo se desea compilar.

3 INTRODUCCIÓN DEL DISEÑO

Introducir un diseño significa el proceso de describir la arquitectura del diseño, utilizando algún método que sea soportado por QUARTUS. La selección del dispositivo a utilizar se hace en el momento de introducir el diseño a esta plataforma. Existen muchos métodos para introducir un diseño; sin embargo, algunas reglas simples a seguir son las siguientes:

- La captura de cualquier esquemático se efectúa en el editor gráfico.
- La captura de diagramas de estado se efectúa en el editor gráfico.
- La captura de diseños hechos en lenguaje de descripción AHDL se efectúa en el editor de texto.
- La captura de diseños hechos en lenguaje de descripción VHDL se efectúa en el editor de texto.
- La captura de diseños hechos en lenguaje de descripción Verilog-HDL se efectúa en el editor de texto.
- La captura de diseños hechos en forma de vectores se efectúa en el editor de forma de onda.
- La captura de archivos generados en Edit y netlist que son diseños hechos con herramientas EDA de otras industrias se efectúa en el editor de texto.

4 SÍNTESIS DEL DISEÑO

Sintetizar un diseño significa traducirlo a código creador de hardware. Para sintetizar un proyecto se requiere compilarlo. Compilar un diseño significa evaluar el manejo correcto de la sintaxis necesaria para la creación de hardware. Puede darse el caso de que un diseño sea compilado exitosamente y sin embargo no sea sintetizable debido a que la plataforma QUARTUS no tenga los constructores necesarios para alguna instrucción de dicho diseño.

Después de sintetizar el diseño se requiere hacer una asignación a los pines de entrada y salida.

5 CONFIGURACIÓN O PROGRAMACIÓN DEL DISEÑO

Después de sintetizar cualquier diseño, éste queda listo para programarse en un CPLD o para configurarse en un FPGA, utilizando el cable JTAG.

6 ESPECIFICACIONES DE UN PROYECTO

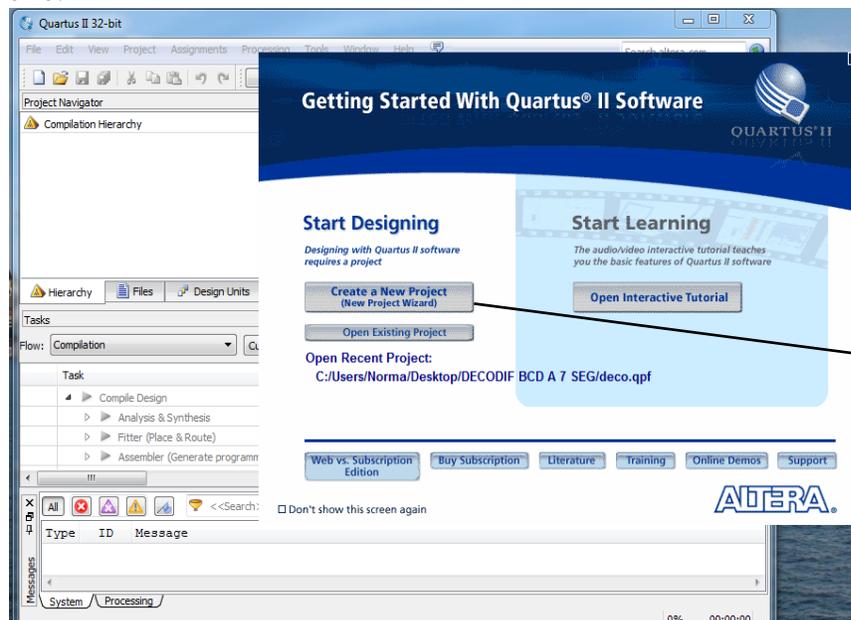
Considérese un proyecto que consta de dos módulos. El primer módulo será un divisor de frecuencia y el segundo será un contador del cero al nueve.

Para el diseño de cada módulo se utilizará la herramienta del editor de texto utilizando el lenguaje “VHDL” y la unión de ambos módulos se hará en el editor de esquemáticos.

7 CREACIÓN DE UN PROYECTO

Lo primero que se requiere es crear una carpeta en donde se tendrán todos los archivos generados durante el desarrollo del proyecto, posteriormente, se requiere abrir la plataforma de Quartus.

Esta acción inicia la ejecución del entorno de desarrollo de la plataforma Quartus, que permite acceder a los recursos y procesos de la plataforma. La figura 1 muestra el menú principal del entorno de desarrollo.



→ Presione aquí

Figura 1. Menú inicial del entorno de desarrollo Quartus

Para crear un nuevo proyecto se requiere de las acciones listadas a continuación:

- En la opción “Create a new project” del menú inicial, seleccione “Next”. Al hacer esto aparece una ventana de ayuda para introducir el nombre y la localización del proyecto, tal como se muestra en la figura 2 y seleccione “Next”.
- Al proyecto se le llamará “contador”.
- Si ya tiene hecho el VHDL, busque y abra ese archivo Presionando la opción “Add”. de otra forma presione la opción “Next”.

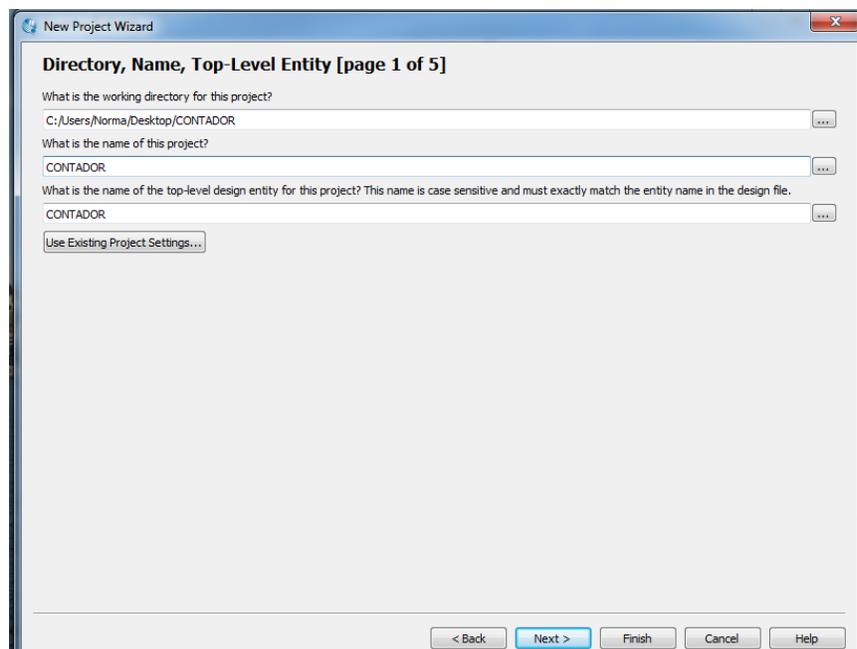


Figura 2. Creación de un nuevo proyecto.

En la página de propiedades del proyecto se requiere escribir los datos del dispositivo que se desee utilizar. El dispositivo incluido en la tarjeta de desarrollo Intel “DE2i-150 board”, con el que cuenta la Facultad de Ingeniería, en el laboratorio de Dispositivos Lógicos Programables tiene las siguientes propiedades:

| | |
|----------------|-----------------|
| Device Family: | Cyclone IV GX |
| Device: | EP4CGX150DF31C7 |

La tabla de propiedades del proyecto aparece como se muestra en la figura 3:

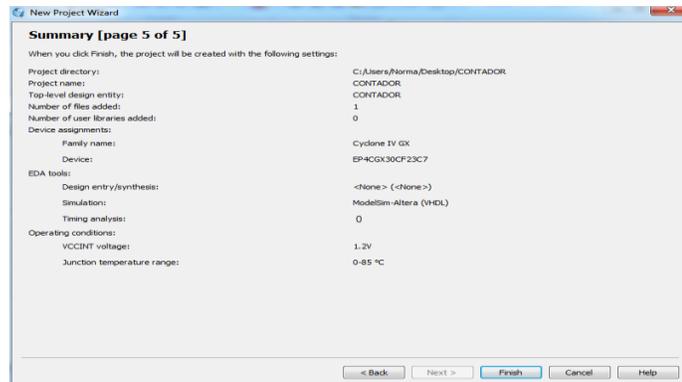


Figura 6.3. Tabla de propiedades del proyecto

Creación del código en VHDL su análisis y síntesis

Una vez que se introducen los datos solicitados en la página de propiedades, se requiere la creación del módulo; por lo que se presiona la opción “siguiente”, para que aparezca otra ventana en la que se debe presionar dos veces el nombre del proyecto que estamos desarrollando para ver o escribir el código del proyecto, posteriormente se debe compilar presionando el botón “analysis & synthesis” tal como lo muestra la figura 4. Para escribir el código se debe hacer click en el botón “FILE” de la barra superior y luego en “NEW”. Por lo que se abre una ventana donde se debe seleccionar el lenguaje HDL a utilizar en nuestro caso (VHDL FILE).

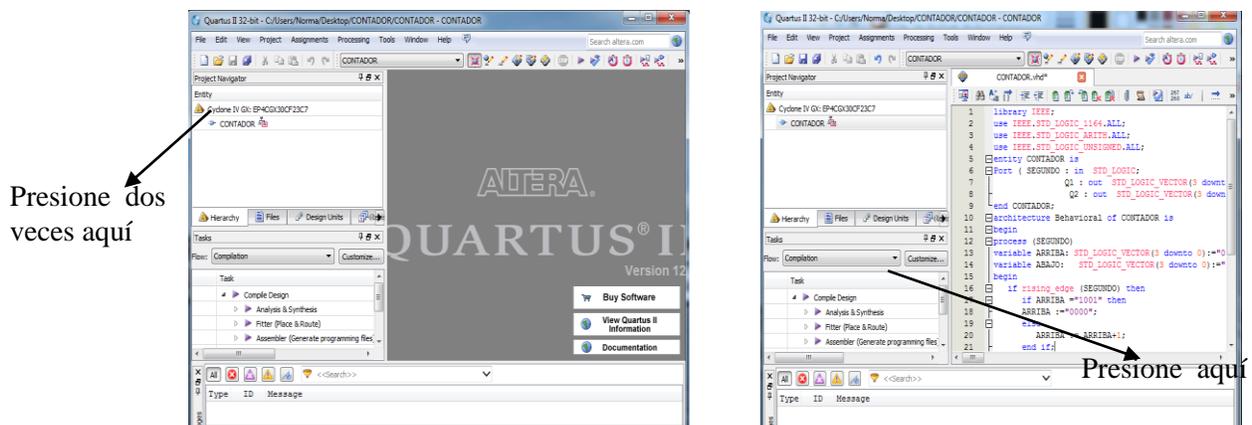


Figura 4. Selección, escritura y compilación del archivo que se desea crear.

Una vez terminado de escribir el código se selecciona en el menú superior la opción “start analysis and synthesis” y si el proyecto no tiene errores de sintaxis aparecerá la ventana que se muestra en la figura 4.

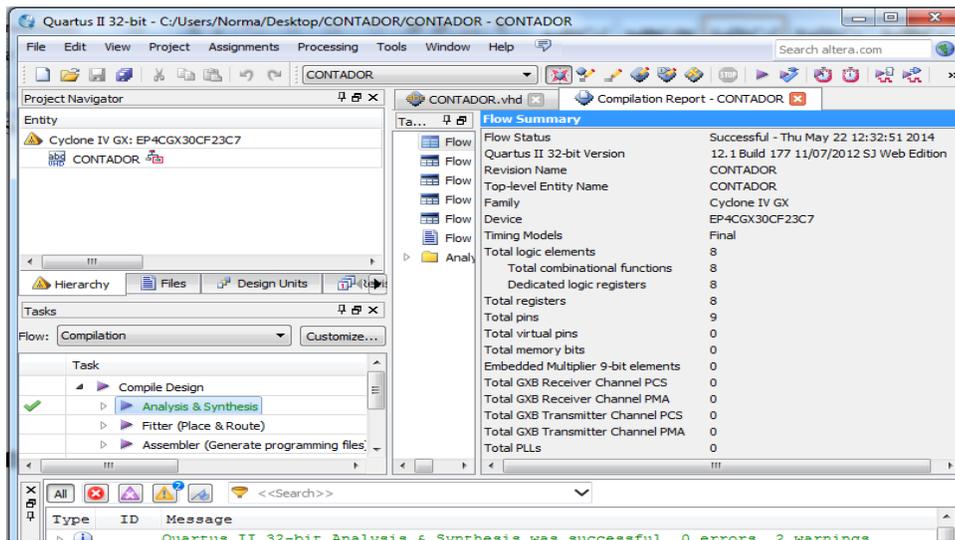


Figura 4. Compilación correcta del proyecto.

- Para abrir la ventana que permite definir los puertos de entrada y salida del diseño en VHDL, se debe oprimir el botón “Pin Planner”, como se muestra en la figura 5.

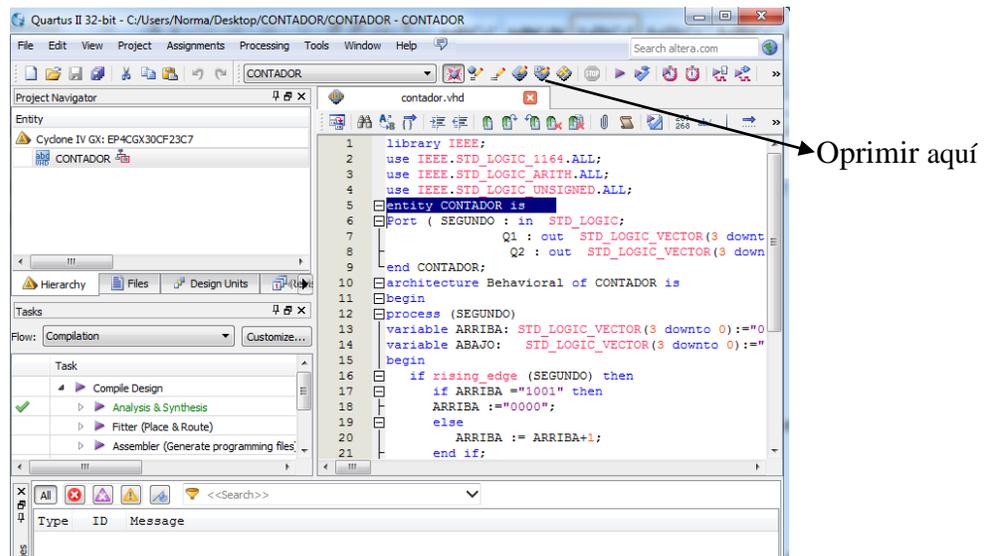


Figura .5. Acceso al Pin Planner Editor.

- Aparecerá la ventana que se muestra en la figura 6 y en la columna que dice “location” se debe anotar el pin que se asignará en el FPGA correspondiente a cada salida y a cada entrada.

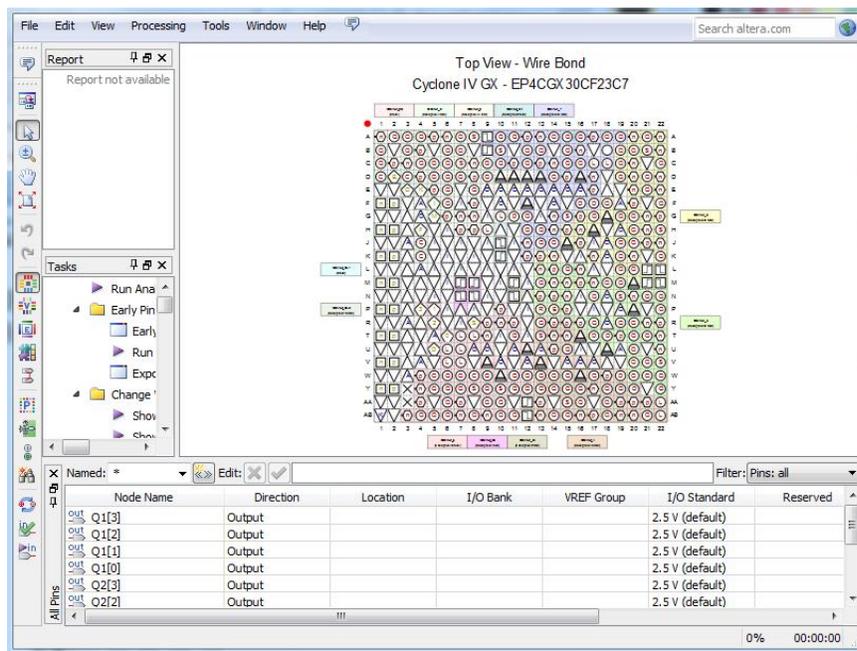
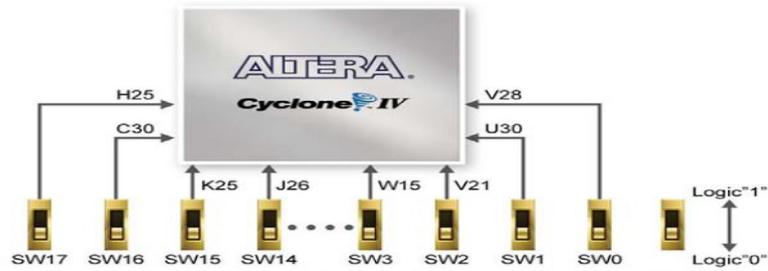


Figura.6. Pin Planner Editor

- Los periféricos más usados se muestran a contin

uación:

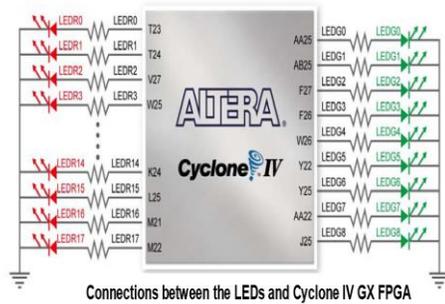


Connections between the slide switches and Cyclone IV GX FPGA

| Signal Name | FPGA |
|-------------|------|
| SW[0] | V28 |
| SW[1] | U30 |
| SW[2] | V21 |
| SW[3] | C2 |
| SW[4] | AB31 |
| SW[5] | U21 |
| SW[6] | T28 |
| SW[7] | R30 |
| SW[8] | P30 |
| SW[9] | R29 |
| SW[10] | R26 |
| SW[11] | N26 |
| SW[12] | M26 |
| SW[13] | N25 |
| SW[14] | J26 |
| SW[15] | K25 |
| SW[16] | C30 |
| SW[17] | H25 |

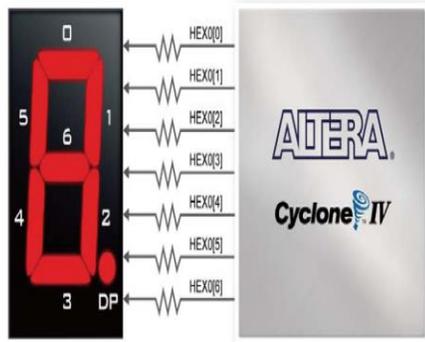
Pin Assignments for Push-buttons

| Signal Name | FPGA |
|-------------|------|
| KEY[0] | AA26 |
| KEY[1] | AE25 |
| KEY[2] | AF30 |
| KEY[3] | AE26 |



Pin Assignments for LEDs

| Signal Name | FPGA |
|-------------|------|
| LEDR[0] | T23 |
| LEDR[1] | T24 |
| LEDR[2] | V27 |
| LEDR[3] | W25 |
| LEDR[4] | T21 |
| LEDR[5] | T26 |
| LEDR[6] | R25 |
| LEDR[7] | T27 |
| LEDR[8] | P25 |
| LEDR[9] | R24 |
| LEDR[10] | P21 |
| LEDR[11] | N24 |
| LEDR[12] | N21 |
| LEDR[13] | M25 |
| LEDR[14] | K24 |
| LEDR[15] | L25 |
| LEDR[16] | M21 |
| LEDR[17] | M22 |
| LEDG[0] | AA25 |
| LEDG[1] | AB25 |
| LEDG[2] | F27 |
| LEDG[3] | F26 |
| LEDG[4] | W26 |
| LEDG[5] | Y22 |
| LEDG[6] | Y25 |
| LEDG[7] | AA22 |
| LEDG[8] | J25 |



Connections between the 7-segment display HEX0 and Cyclone IV GX FPGA

| Signal Name | FPGA | HEX4[2] | C7 |
|-------------|------|---------|-----|
| HEX0[0] | E15 | HEX4[3] | C6 |
| HEX0[1] | E12 | HEX4[4] | C5 |
| HEX0[2] | G11 | HEX4[5] | C4 |
| HEX0[3] | F11 | HEX4[6] | C3 |
| HEX0[4] | F16 | HEX5[0] | D3 |
| HEX0[5] | D16 | HEX5[1] | A10 |
| HEX0[6] | F14 | HEX5[2] | A9 |
| HEX1[0] | G14 | HEX5[3] | A7 |
| HEX1[1] | B13 | HEX5[4] | A6 |
| HEX1[2] | G13 | HEX5[5] | A11 |
| HEX1[3] | F12 | HEX5[6] | B6 |
| HEX1[4] | G12 | HEX6[0] | B9 |
| HEX1[5] | J9 | HEX6[1] | B10 |
| HEX1[6] | G10 | HEX6[2] | C8 |
| HEX2[0] | G8 | HEX6[3] | C9 |
| HEX2[1] | G7 | HEX6[4] | D8 |
| HEX2[2] | F7 | HEX6[5] | D9 |
| HEX2[3] | AG30 | HEX6[6] | E9 |
| HEX2[4] | F6 | HEX7[0] | E10 |
| HEX2[5] | F4 | HEX7[1] | F8 |
| HEX2[6] | F10 | HEX7[2] | F9 |
| HEX3[0] | D10 | HEX7[3] | C10 |
| HEX3[1] | D7 | HEX7[4] | C11 |
| HEX3[2] | E6 | HEX7[5] | C12 |
| HEX3[3] | E4 | HEX7[6] | D12 |
| HEX3[4] | E3 | | |
| HEX3[5] | D5 | | |
| HEX3[6] | D4 | | |
| HEX4[0] | A14 | | |
| HEX4[1] | A13 | | |



Block diagram of the clock distribution

Una vez que se anotaron los pines de entrada y de salida como se muestra en la figura 7, se salva y cierra esa ventana.

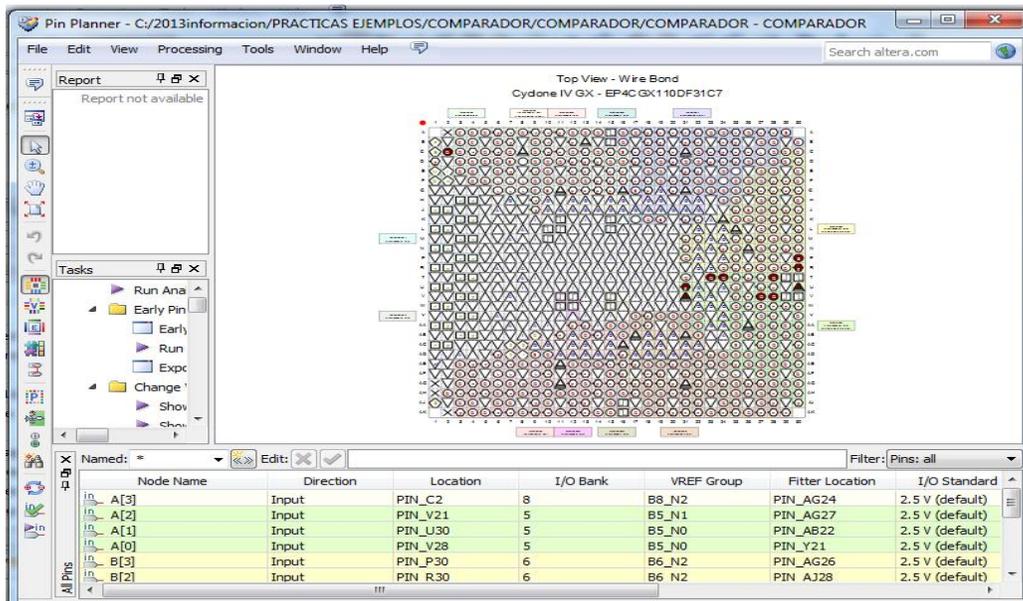


Figura.7. Anotación de pines de entrada y de salida.

Una vez cerrado la ventana del “pin planner” se hace click en el botón “start compilation” del menú de la barra superior.

Posteriormente hacer click en el botón programmer, de la misma barra superior, para iniciar la programación del chip, ver la figura 8.

Hacer click aqui



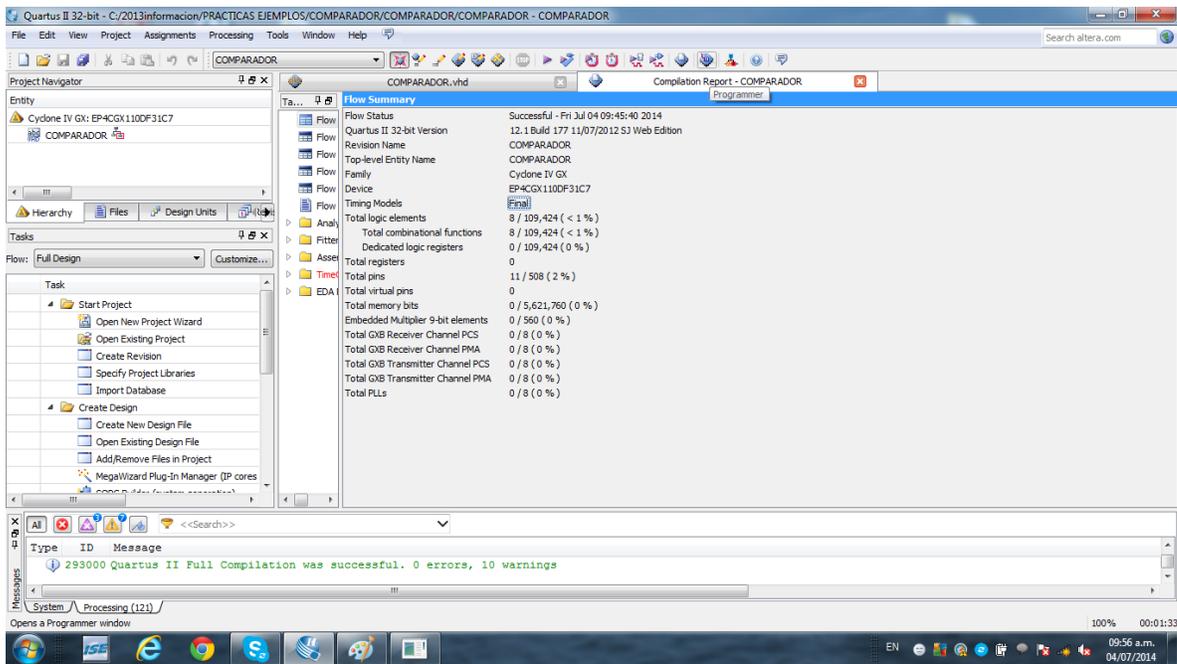
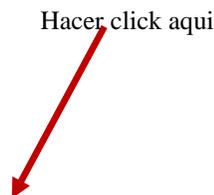


Figura.8. Botón para acceder al programador.

Posteriormente aparecerá el siguiente menú, en donde se debe hacer click en el botón start, para iniciar la programación del FPGA. Ver figura 9.



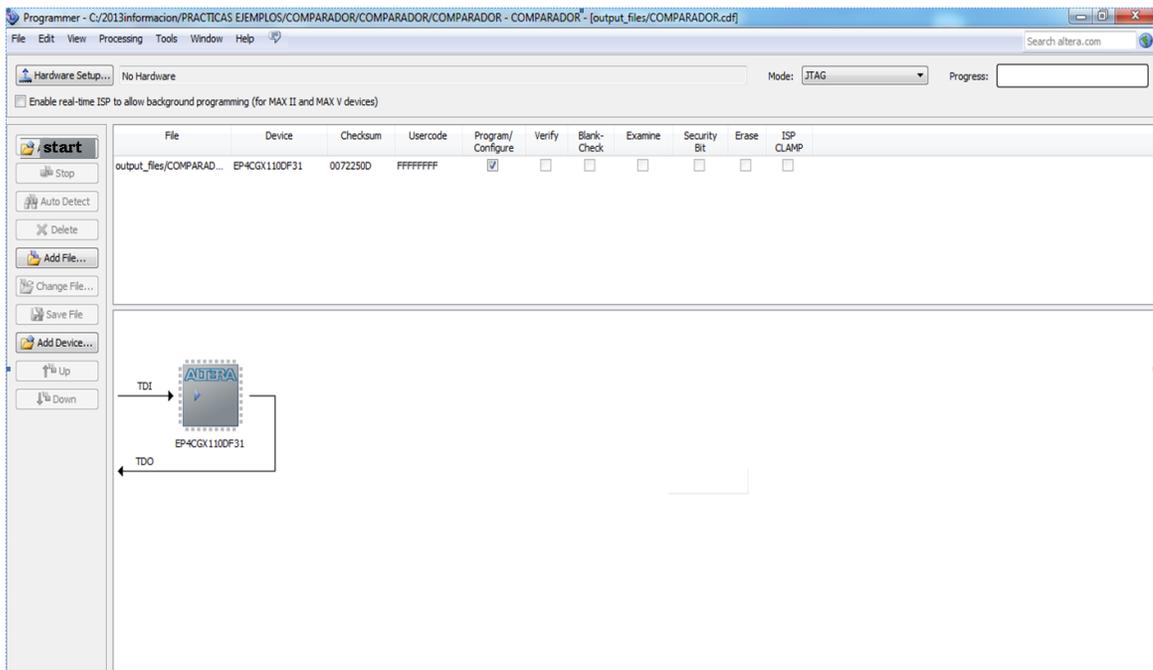


Figura 9. Programmer Editor