

# Guía para realizar el primer diseño con QUARTUS II

## Introducción

A través de esta guía aprenderás cómo hacer un diseño utilizando el programa QUARTUS II y la placa DE0 del curso de Diseño Lógico de la Facultad de Ingeniería de la Universidad de la República.

Mediante un ejemplo sencillo veremos cómo ingresar un diseño, cómo simularlo, cómo asignar las señales deseadas a los pines adecuados del chip y finalmente cómo programarlo.

El circuito que diseñaremos consiste en un circuito combinatorio que tiene como entrada la posición de 4 switches y como salida el estado de 2 leds.

Las salidas deberán cumplir con las siguientes ecuaciones:

1.  $d1 = a!bc + !abc + bd$
2.  $d2 = ac + !b!d$

Siendo abcd el estado de los switches SW[3..0].

Construir la tabla de verdad de estas funciones y verificar que las expresiones son mínimas.

## Antes de Comenzar..

Antes de comenzar debemos instalar el software necesario para realizar nuestro primer diseño.

Insertar el DVD que viene con la placa en la computadora donde vamos a trabajar, si no se despliega el menú de instalación automáticamente, ejecutar el archivo *install.exe* que se encuentra en la carpeta *Altera Complete Design Suite*. De aquí tenemos que instalar el programa *Quartus II Web Edition*.

El otro programa que debemos instalar es el *USB Blaster*, las instrucciones de cómo hacerlo se encuentran en la guía *Getting Started with Altera DE0 board* incluida en el DVD de la placa.

### Algunos errores del manual *Getting Started with Altera DE0 board*:

- Página 5, Figura 3: la ruta correcta en *Include this location in the search* es C:\altera\90\quartus\drivers\usb-blaster.
- Página 8, Step 3: el switch de ON/OFF es el SW10 y no el SW5 como aquí se indica.
- Página 10, Step 3: el switch de ON/OFF es el SW10 y no el SW11 como aquí se indica.

**Nota:** Si luego de instalado el *USB Blaster* nos aparece un mensaje de error relacionado con que el controlador no está firmado digitalmente y que Windows no permite su uso. Cuando vayamos a trabajar con la placa, antes que nada, debemos reiniciar la computadora, presionar F8 cuando está booteando el sistema y seleccionar la opción que permite utilizar controladores sin firma digital. Esto soluciona el problema.

# 1-Crear un Proyecto

Para este diseño vamos a crear 3 proyectos, uno para cada ecuación y uno para unir los bloques, comencemos por la primera ecuación, proyecto que llamaremos *dl1*.

Al iniciar el Quartus II se despliega un menú para crear un proyecto nuevo o abrir uno existente, allí elegiremos la opción *Create a New Project (New Project Wizard)*. También podemos acceder a esta opción a través de *File>New Project Wizard*.

A continuación llenaremos los recuadros de la siguiente manera:

1. Elegimos el directorio de trabajo (*c:\directorio\_de\_trabajo\tutorial\dl1*), el nombre del proyecto (*dl1*) y el nombre del top level entity de este proyecto (*dl1*). Ver Figura 1.
2. En el siguiente recuadro nos consulta si vamos a agregar algún archivo de diseño a nuestro proyecto, en este caso no vamos a agregar ninguno así que solo presionamos *Next*. Ver Figura 2.
3. Seleccionamos el chip que vamos a utilizar. En la sección *Device family* elegimos *Cyclone III*, y luego en la lista de *Available devices* elegimos el chip de la placa *EP3C16F484C6*. Ver Figura 3.
4. En el siguiente recuadro no vamos a cambiar nada así que solo presionamos *Next*. Ver Figura 4.
5. Al final vemos un resumen de las opciones elegidas. Ver Figura 5.
6. Por último presionamos en *Assignments > Settings*, se despliega un menú donde seleccionamos en *Category: Timing Analysis Settings*, elegir la opción *Use Classic Timing Analyzer during compilation* y presionar *Ok*. Ver Figura 6.

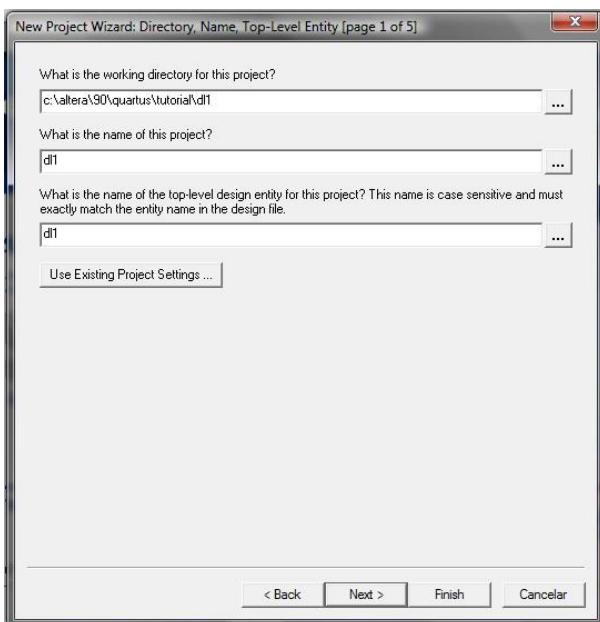


Figura 1

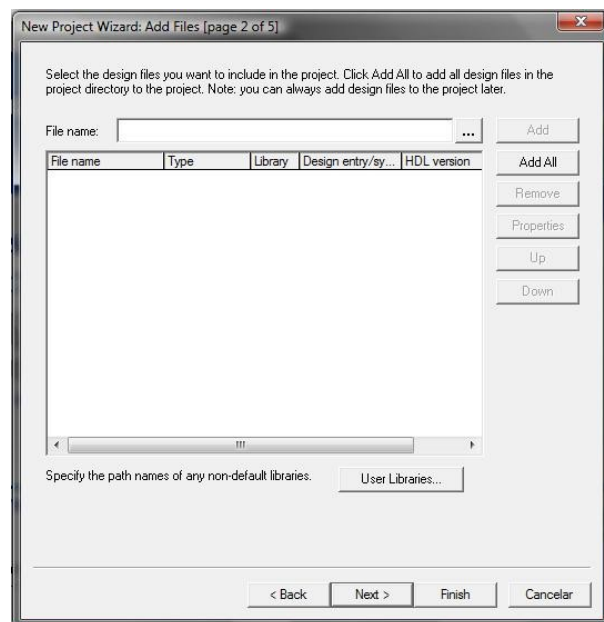


Figura 2

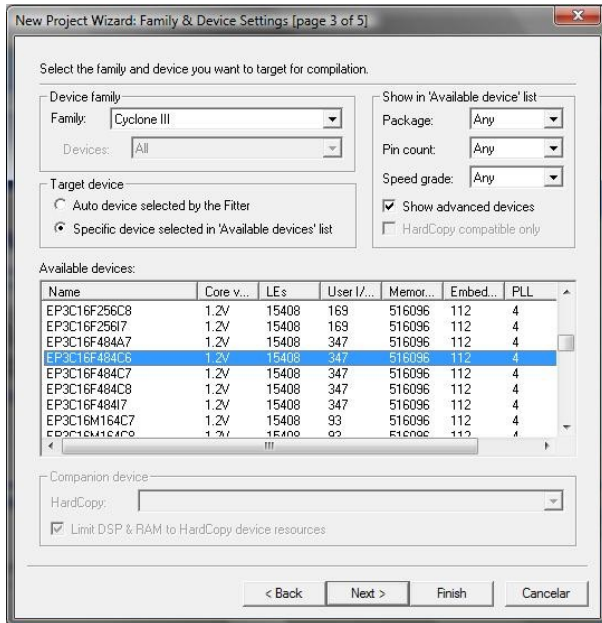


Figura 3

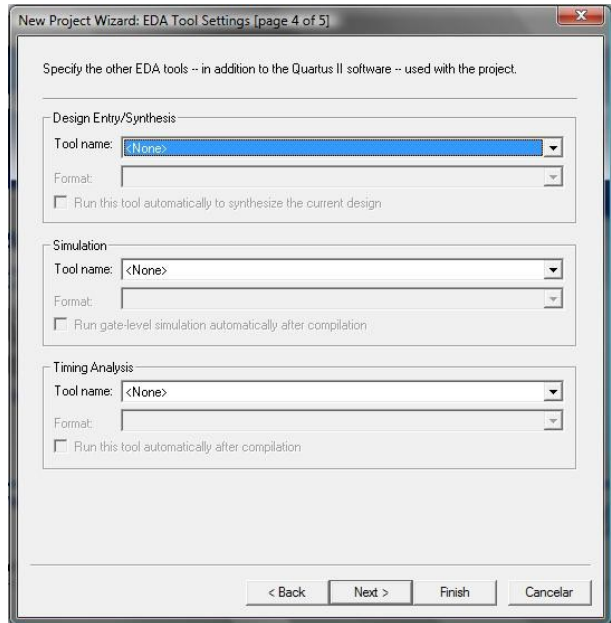


Figura 4

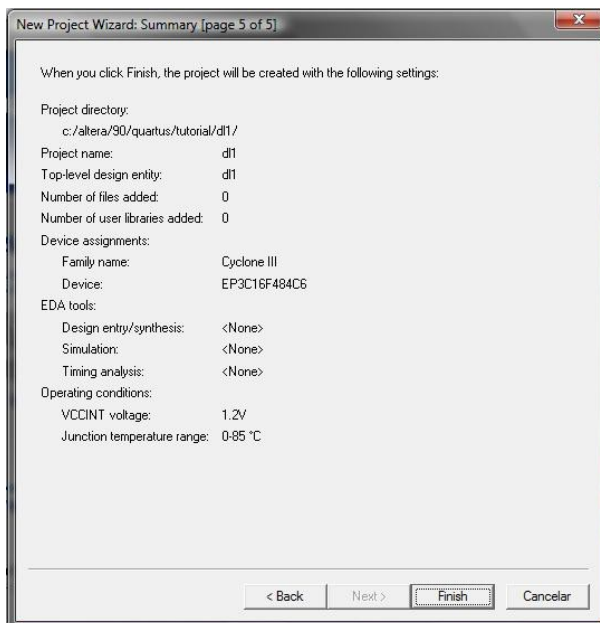


Figura 5

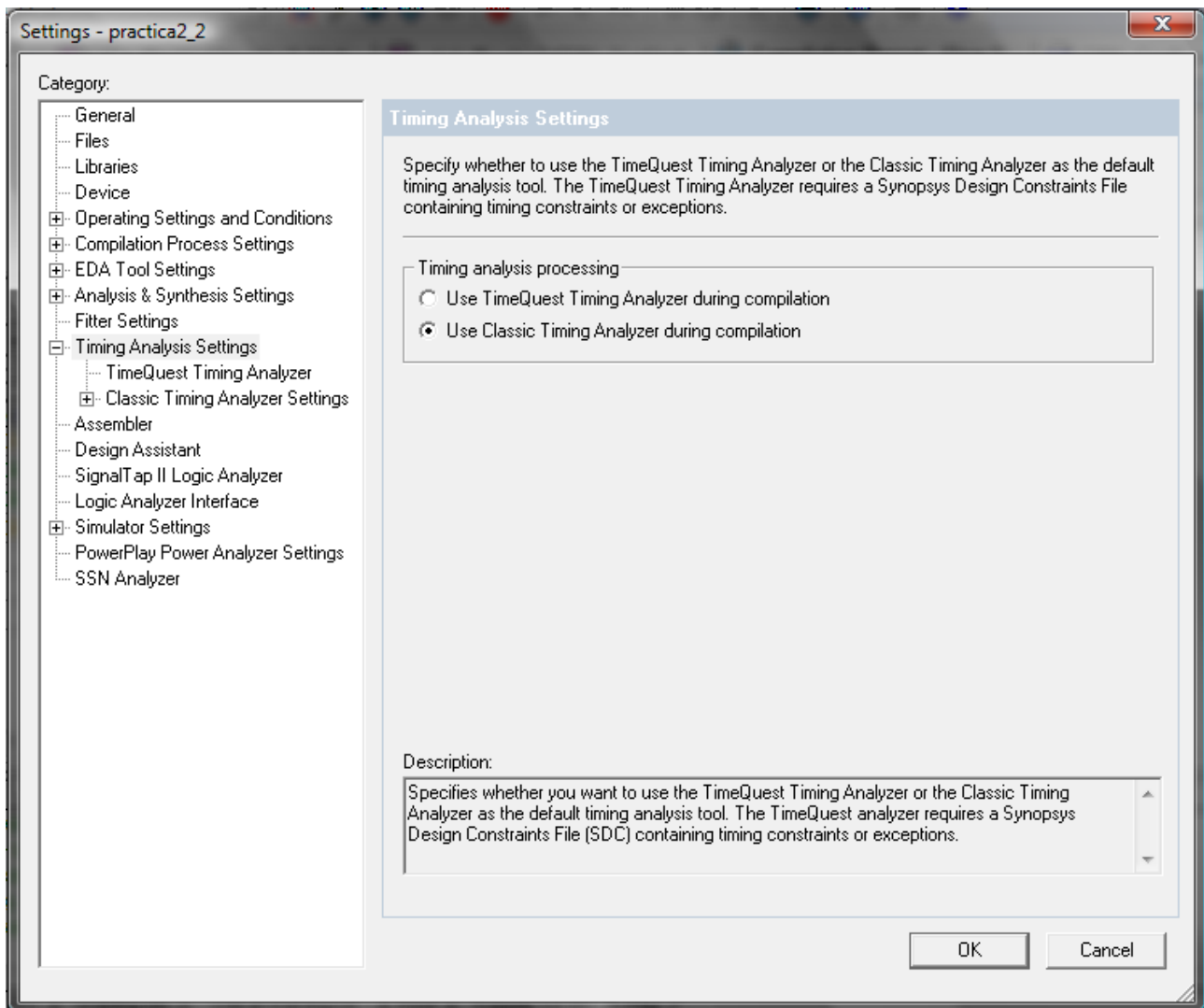


Figura 6

## 2-Crear un Archivo de Diseño

Luego de haber creado el proyecto vamos a comenzar con el diseño.

Seleccionamos *File>New...* y en el menú desplegado elegimos *Design files>Block Diagram/Schematic File*. Se abrirá una ventana en la cual se podrá comenzar a ingresar el circuito en forma gráfica. Guardar este archivo como *dl1.bdf*. En este archivo dibujaremos el circuito combinatorio correspondiente a la ecuación 1.

- Para ingresar una compuerta, hacer doble click en la zona de dibujo. Se abrirá una ventana que permite seleccionar el componente a ingresar. Teclear en el cuadro *Symbol Name* el tipo de compuerta indicando cantidad de entradas. Ejemplos: *or2* (OR de 2 entradas), *and4* (AND de 4 entradas), *not* (inversor), etc.
- Las entradas y salidas del circuito se ingresan como cualquier componente, y sus nombres son *input* y *output* respectivamente. Los nombres de dichas señales se pueden cambiar haciendo doble click sobre su etiqueta.
- Para conectar los distintos componentes hacer click en uno de los terminales de la compuerta y arrastrar el mouse hasta el punto que se desea conectar, al final llegaremos a un diseño similar a la Figura 7.
- Una vez dibujado el circuito y salvado, se puede proceder a la primera compilación. Para esto solo tenemos que presionar en *Processing > Start Compilation*.
- En caso que el Quartus reporte algún error este debe ser corregido. Una vez que se obtenga una compilación libre de errores vamos a crear un símbolo de nuestro diseño para utilizar en el proyecto final (este paso solo lo hacemos para los diseños de las ecuaciones 1 y 2). Para esto solo debemos hacer doble click en el archivo *.bdf* y luego presionar en *File>Create/Update>Create Symbol Files for Current File*. Esto nos crea un archivo *.bsf* (Block Symbol File) en la carpeta que estamos trabajando .

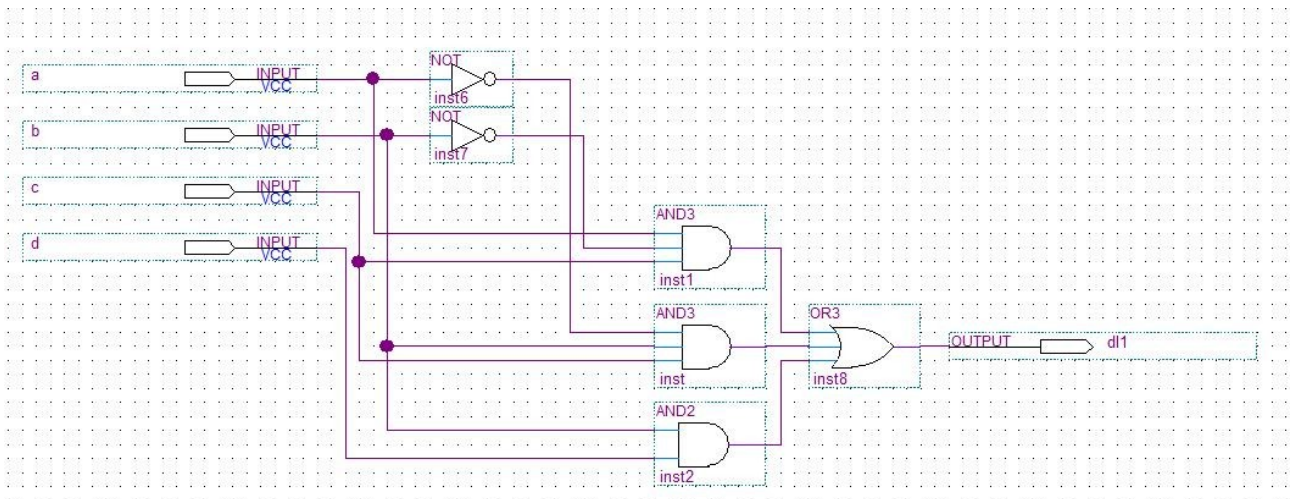


Figura 7

### 3-Simulación

Ahora estamos en condiciones de realizar una simulación del primer bloque que hemos diseñado. Primero debemos seleccionar *File>New...* y en el menú desplegado elegimos *Verification/Debugging Files>Vector Waveform File*. Se abrirá el editor de formas de onda en el cual se pueden dibujar las entradas para la simulación. Dos valores importantes para fijar inicialmente son el tamaño de la grilla (*Edit>Grid size...*) y el tiempo total de simulación (*Edit>End time...*). Vamos a utilizar los siguientes valores, *Grid size* en 100ns y *End time* en 1.6us.

Para elegir las señales que veremos en nuestra simulación hacer doble click abajo de la columna de *Name*, se despliega un menú donde presionamos el botón *Node Finder...*, una vez dentro del *Node Finder* verifiquemos que en la ventana *Filter* este seleccionado *Pins: all* y si no lo seteamos de esta forma (quizás para otros proyectos es útil usar algún filtro), ahora presionamos el botón *List* y se despliega una lista de todos los nodos del diseño, solo queda hacer doble click sobre la señales que queremos ver en la simulación (en este caso todas) para ponerlas dentro de la lista de *Selected Nodes* y presionar *Ok*. Ver Figura 8.

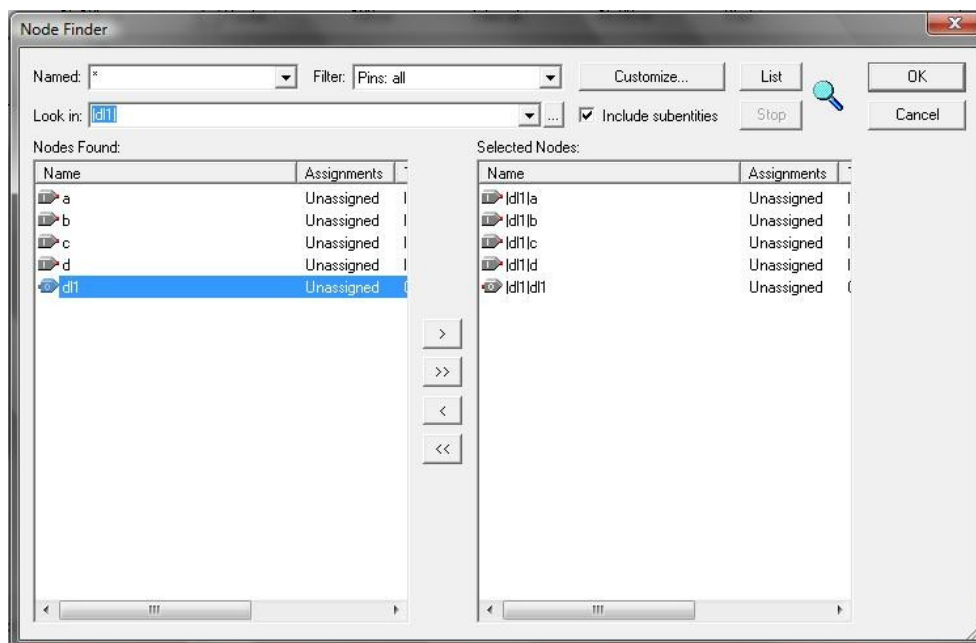


Figura 8

Las formas de onda se pueden dibujar con ayuda de las herramientas que aparecen sobre la izquierda de la pantalla (permiten sobrescribir con 0, 1, X o Z la zona seleccionada, además de ingresar relojes, contadores, etc.) Algo útil es agrupar señales y verlas como un número en alguna base adecuada. Por ejemplo en este caso se pueden agrupar las señales abcd. Primero se deben ordenar de forma que la señal a sea la que está en la parte superior y la señal d en la inferior. Luego se deben seleccionar todas las señales. Presionar el botón derecho sobre la selección y hacer click sobre *Grouping>Group*, darle un nombre al grupo y seleccionar la base en la que se desea mostrar el número que forman las señales. Sobre la izquierda existe la opción de generar un contador binario que puede ser útil para generar todas las entradas posibles al circuito.

Habiendo dibujado las formas de onda deseadas guardamos el archivo y primero vamos a realizar una simulación Funcional en donde no aparecen los retardos de los componentes pero así podemos

verificar la lógica de nuestro diseño (no siempre es necesario hacer este estudio, en varias ocasiones se puede pasar directo a la simulación con retardos). Para esto debemos presionar en *Processing>Generate Functional Simulation Netlist*, luego abrimos *Processing>Simulator Tool*, en *Simulation mode* seleccionamos *Functional* y en *Simulation input* buscamos con el explorador el archivo del *Wave form* que acabamos de crear. Ver Figura 9.

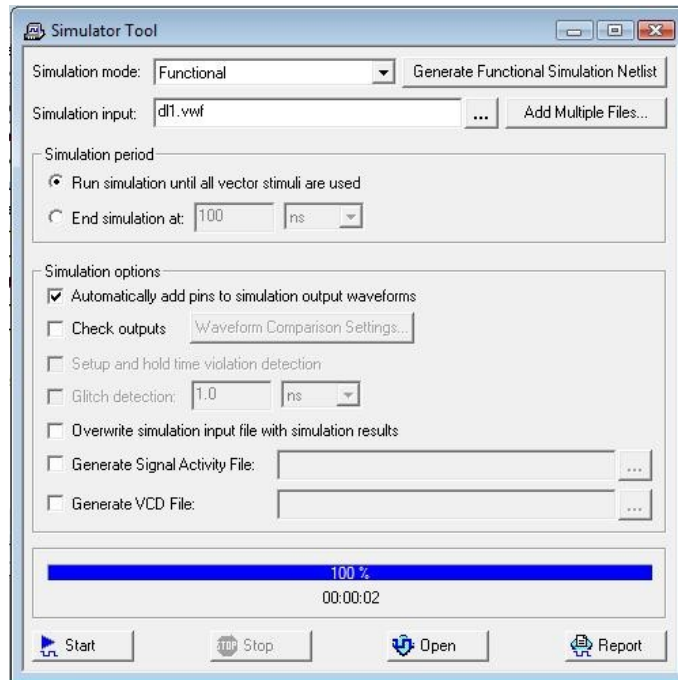


Figura 9

Ahora solo queda presionar *Start* y terminada la simulación presionar *Report* para ver los resultados. Ver Figura 10.

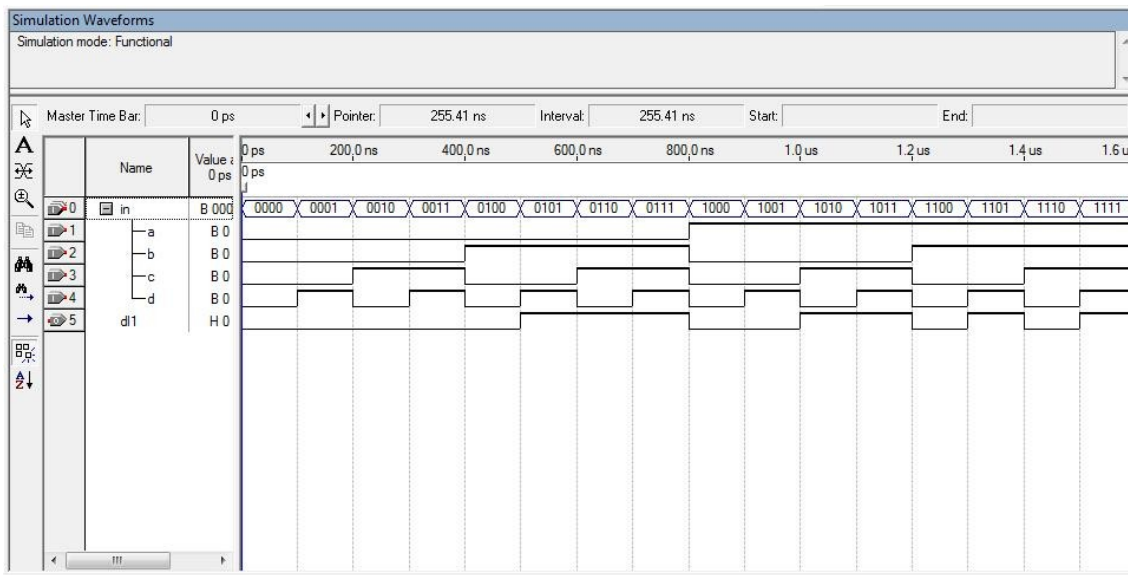


Figura 10

Para realizar una simulación teniendo en cuenta los retardos de los componentes en el *Simulator Tool*, en *Simulation mode* seleccionamos *Timing* y luego procedemos de igual manera que anteriormente.

**Repetir los pasos 1, 2 y 3 para la ecuación 2 con las siguientes diferencias:**

**Paso 1.1:**

Elegimos el directorio de trabajo (*c:\directorio\_de\_trabajo\tutorial\dl2*), el nombre del proyecto (*dl2*) y el nombre del top level entity de este proyecto (*dl2*)

**Paso 2:**

Guardar el archivo de diseño como *dl2.bdf*

El diagrama de bloques de esta ecuación nos quedará algo similar a la Figura 11.

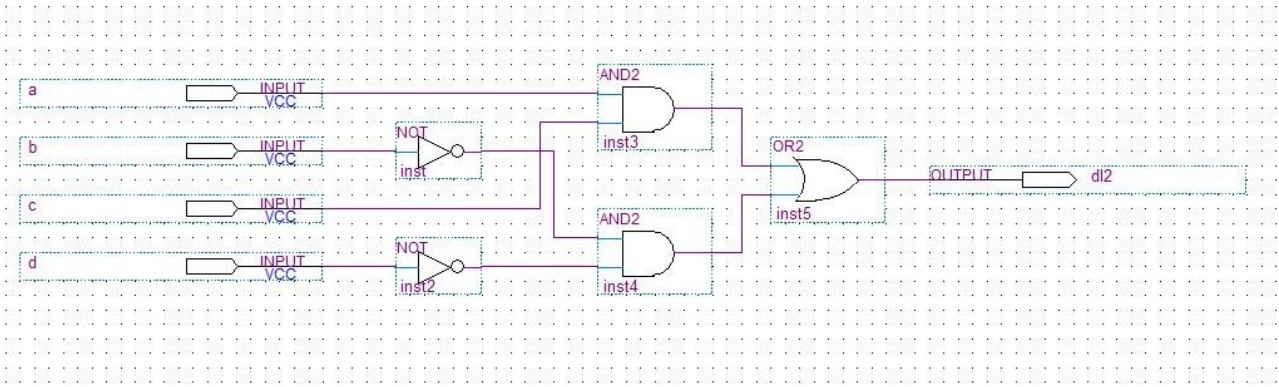


Figura 11

**Ahora repetir los pasos 1, 2 y 3 para el proyecto que une los diseños 1 y 2, con las siguientes diferencias:**

**Paso 1.1:**

Elegimos el directorio de trabajo (*c:\directorio\_de\_trabajo\tutorial*), el nombre del proyecto (*tutorial*) y el nombre del top level entity de este proyecto (*tutorial*)

**Paso 1.2:**

Aquí debemos agregar los archivos que ya hemos diseñado, buscar con el explorador dentro de las carpetas *c:\directorio\_de\_trabajo\tutorial\dl1* y *c:\directorio\_de\_trabajo\tutorial\dl2* los archivos *dl1.bdf* y *dl2.bdf* respectivamente y luego presionar *Add*. Ver Figura 12.

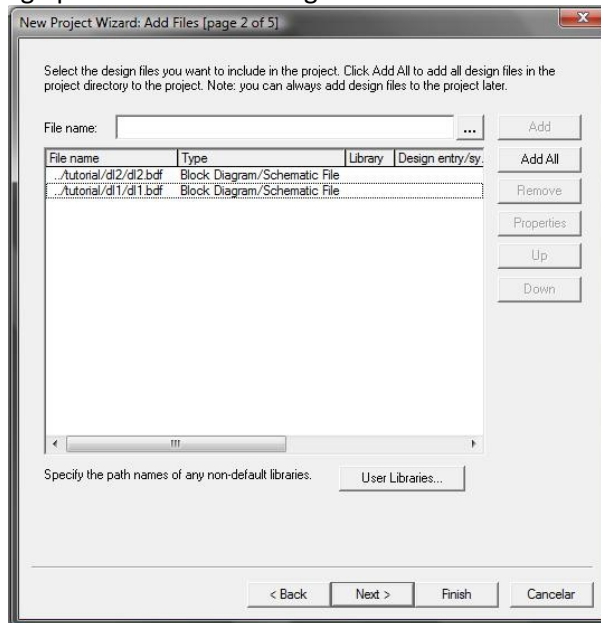


Figura 12

**Paso 2:**



Para el diseño final vamos a utilizar los símbolos que hemos creado previamente, para ello hacemos doble click en la zona de dibujo y en la ventana de *Name*: buscar con el explorador en las carpetas *c:\directorio\_de\_trabajo\tutorial\d11* y *c:\directorio\_de\_trabajo\tutorial\d12* los archivos *d11.bsf* y *d12.bsf* respectivamente. Guardar el archivo de diseño como *tutorial.bdf* y en la ventana de *Project Navigator* en la pestaña de *Files* presionar el botón derecho sobre el archivo tutorial y elegir en el menú desplegado *Set as Top-Level Entity*.

El diagrama de bloques nos quedará algo similar a la Figura 13.

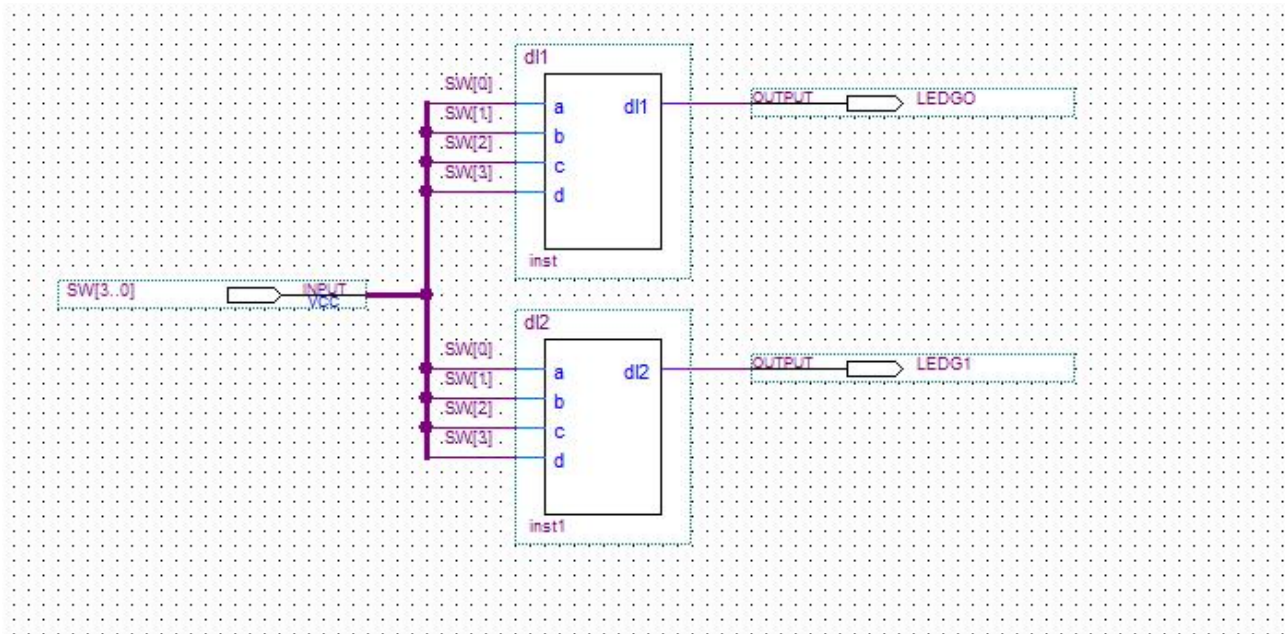


Figura 13

En la Figura se puede ver que se ha insertado un bus. Notar que el nombre de la entrada es *SW[3..0]*. Esto le dice al compilador que es una entrada de 4 bits. Para conectarse a un pin del bus se debe primero dibujar un trozo de cable, luego apretar el botón derecho sobre ese trozo, seleccionar *Properties* y escribir el nombre especificando entre paréntesis rectos a que número corresponde (ejemplo: *SW[0]*).

## 4-Asignar Pines

Luego de los pasos 1, 2 y 3 del proyecto tutorial, siendo este el que vamos a programar en la placa, vamos a continuar asignando los pines de entrada y salida.

### 4.1-Pines sin usar.

Para evitar conflictos con elementos no utilizados en la placa se debe indicar al Quartus 2 que todos los pines no utilizados se configuren como entradas.

Para eso seleccionar el menú "Assignments/Device". Hacer click en el boton "Device and Pin Options" (ver fig. 14), seleccionar la solapa "Unused Pins" y asegurarse que para "Reserve all unused pins" esté seleccionada la opción "As input tri-stated with weak pull-up resistor" (fig. 15)

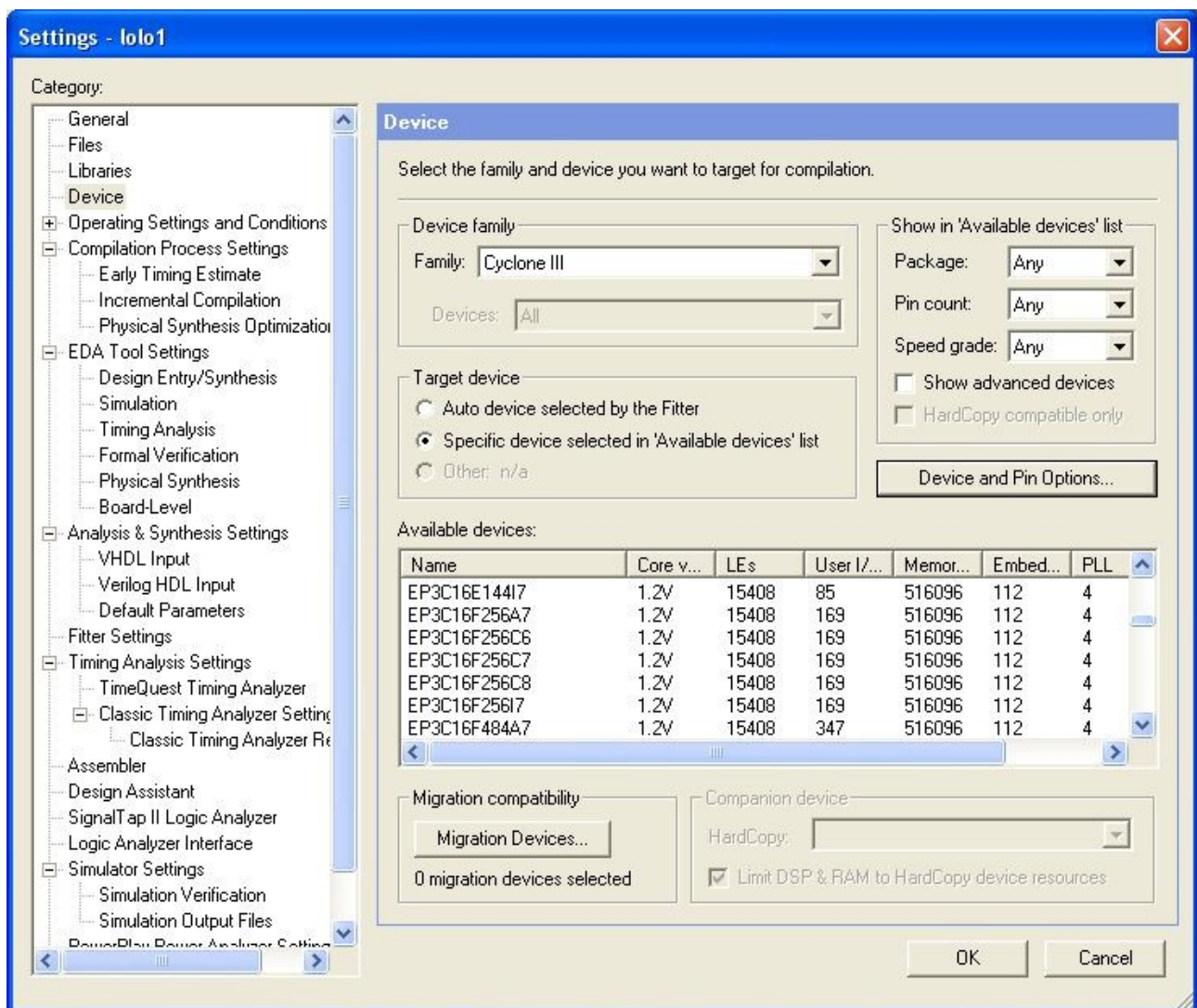


Figura 14

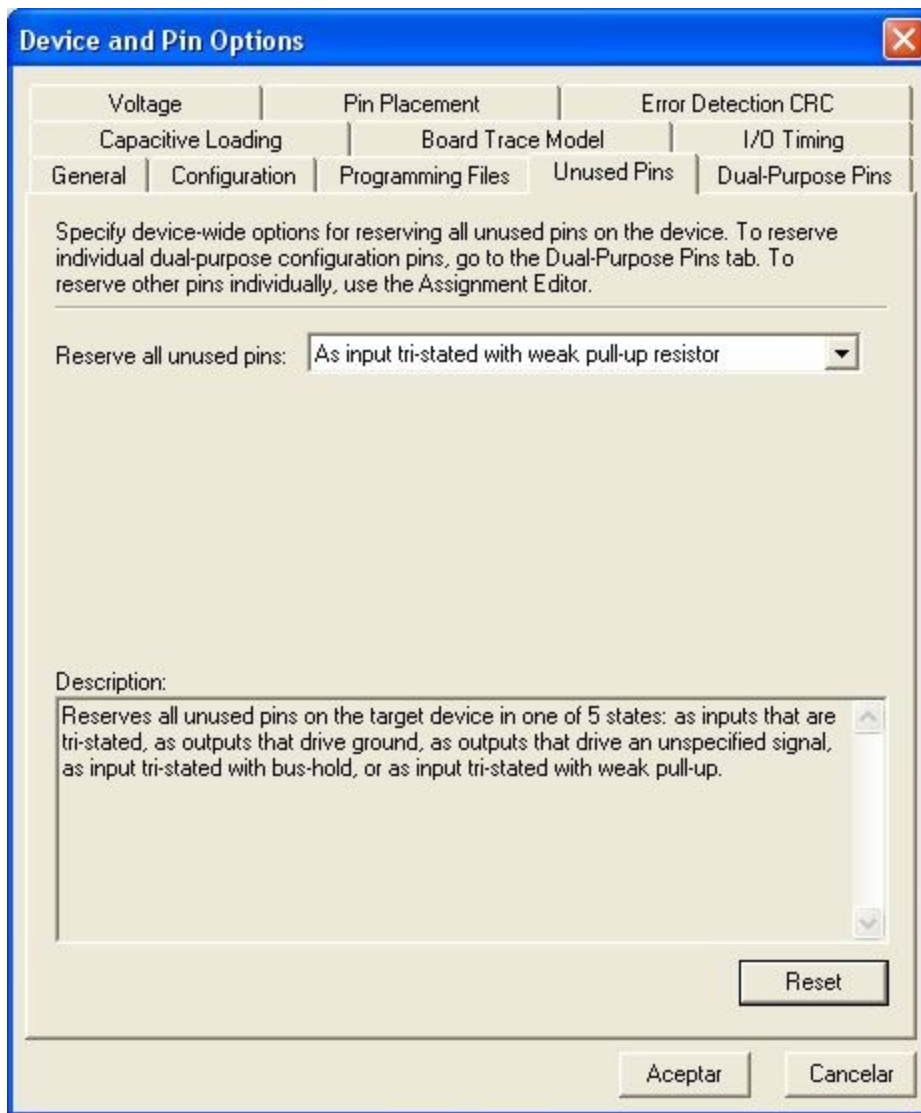


Figura 15

## 4.2-Pines del diseño

Las tablas que indican qué pines se corresponden con qué elementos de las placas se encuentran en el *DE0 User Manual* disponible en el DVD que viene con la placa.

Signal Name	FPGA Pin No.	Description
LEDG[0]	PIN_J1	LED Green[0]
LEDG[1]	PIN_J2	LED Green[1]
LEDG[2]	PIN_J3	LED Green[2]
LEDG[3]	PIN_H1	LED Green[3]
LEDG[4]	PIN_F2	LED Green[4]
LEDG[5]	PIN_E1	LED Green[5]
LEDG[6]	PIN_C1	LED Green[6]
LEDG[7]	PIN_C2	LED Green[7]
LEDG[8]	PIN_B2	LED Green[8]
LEDG[9]	PIN_B1	LED Green[9]

Tabla 1

Signal Name	FPGA Pin No.	Description
SW[0]	PIN_J6	Slide Switch[0]
SW[1]	PIN_H5	Slide Switch[1]
SW[2]	PIN_H6	Slide Switch[2]
SW[3]	PIN_G4	Slide Switch[3]
SW[4]	PIN_G5	Slide Switch[4]
SW[5]	PIN_J7	Slide Switch[5]
SW[6]	PIN_H7	Slide Switch[6]
SW[7]	PIN_E3	Slide Switch[7]
SW[8]	PIN_E4	Slide Switch[8]
SW[9]	PIN_D2	Slide Switch[9]

Tabla 2

Como entradas vamos a utilizar los switches 0, 1, 2 y 3 y como salida los LEDs 0 y 1. Ver Tablas 1 y 2.

Para comenzar con la asignación de pines debemos presionar en *Assignments>Pins*. Luego en la columna *Location* indicamos que pines vamos a utilizar, por ejemplo para el LED 0 basta con escribir *J1* y presionar *Enter*. Ver Figura 14.

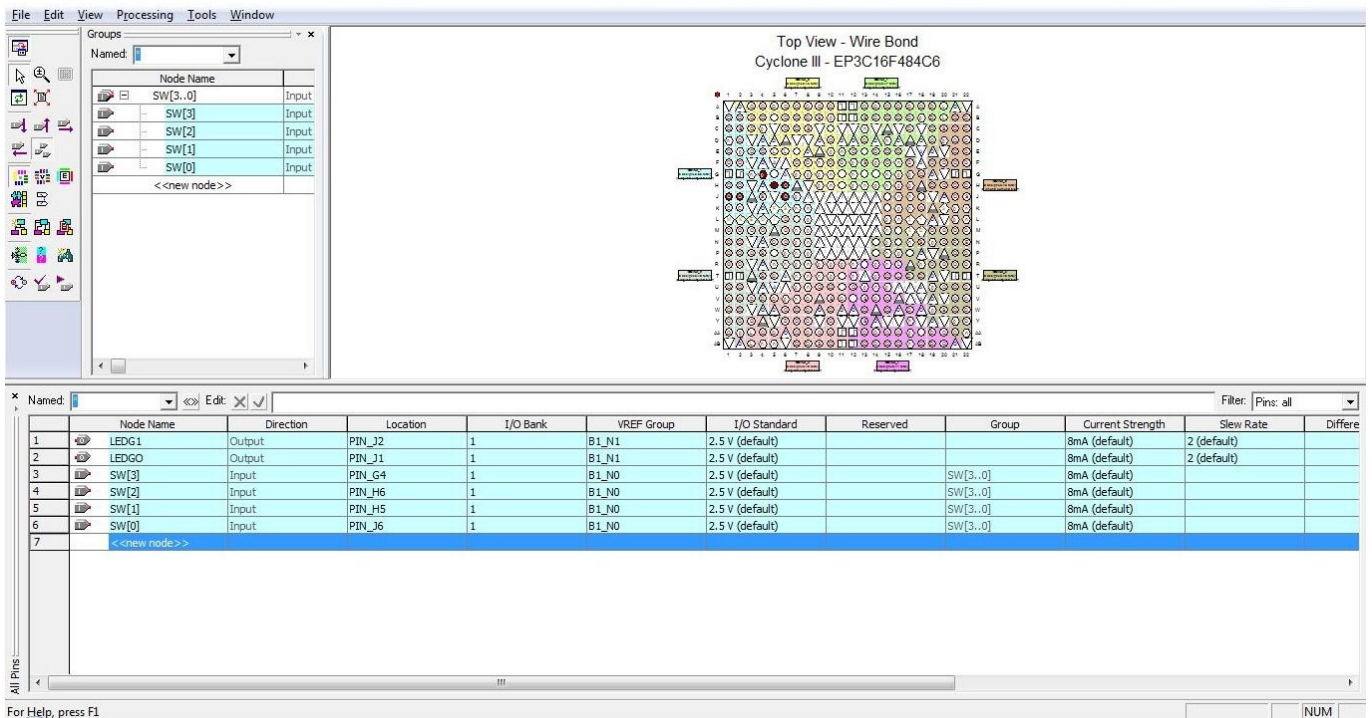


Figura 16

Finalizada la asignación volver a compilar el proyecto y verificar la asignación de pines en *Compilation Report>Fitter>Resource Section>Input Pins* y *Compilation Report>Fitter>Resource Section>Output Pins*. Ver Figura 15.

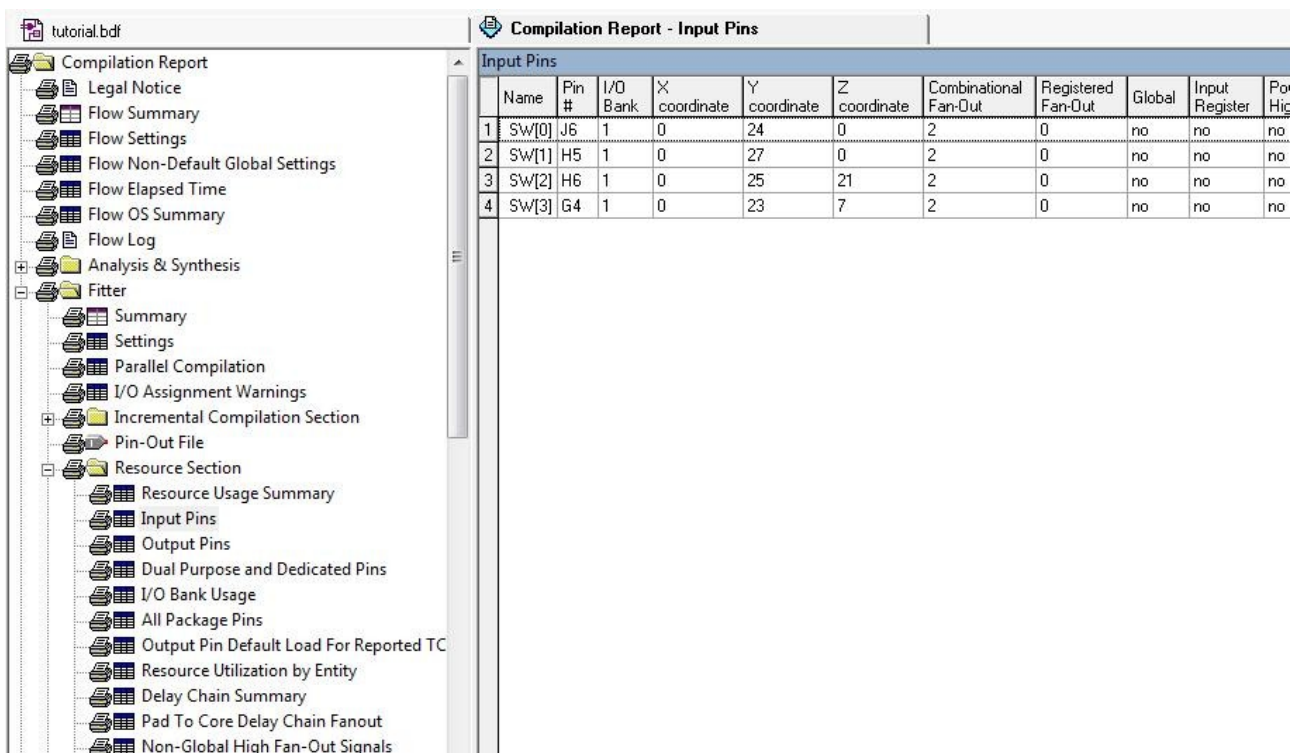


Figura 17

## 5-Programar la Placa

Antes de programar recordar que si tuvimos problemas al instalar el *USB-Blaster* debemos reiniciar la computadora, presionar F8 cuando está booteando la máquina y elegir la opción que nos permite usar controladores no firmados.

- 1- Conectar el cable de alimentación a la placa en el conector J7 y a una toma de corriente.
- 2- Conectar el cable USB a la computadora y al puerto USB blaster (J8) de la placa.
- 3- Verificar que el switch 11 se encuentre en la posición de run y encender la placa presionando el botón rojo de ON/OFF (SW 10).
- 4- Abrir el programador de Quartus II presionando en *Tools>Programmer*.
- 5- Hacer click en *Hardware Setup...*
- 6- Elegir de la lista de *Available hardware items: USB-Blaster [USB-0]*.
- 7- Ya nos debería aparecer el archivo *tutorial.sof*, en caso de que no sea así presionar en *Add File...* y seleccionarlo.
- 8- Presionar *Start...* y la programación comenzará. Ver Figura 16.

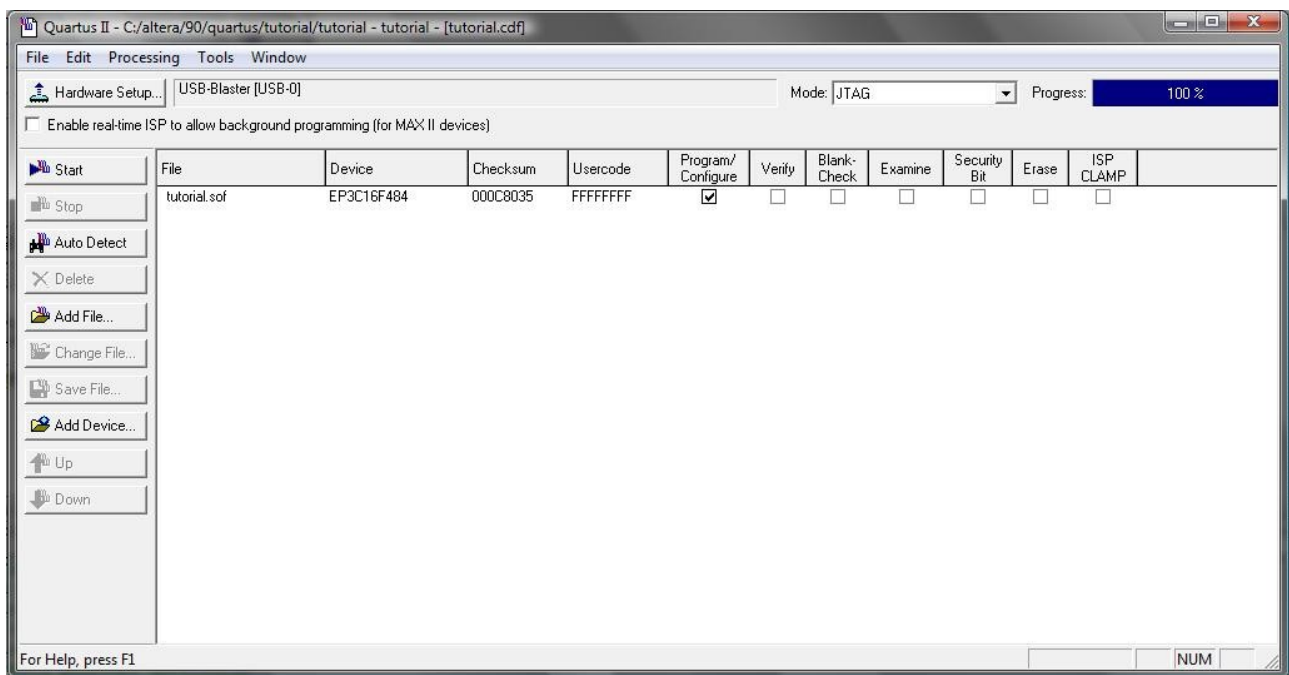


Figura 18

## 6-Verificar el Diseño

Una vez programada la placa lo único que nos queda por hacer es verificar que nuestro diseño es correcto, para esto darle distintas entradas al sistema variando las posiciones de los switches SW[3..0] y verificar que la salida es la esperada visualizando el estado de los leds LEDG0 y LEDG1 (dado que los leds son activos por alto, si el led está encendido tenemos un 1 en la salida y si está apagado un 0).

## 7-Sugerencias para el manejo de señales vectoriales (buses)

A la hora de realizar esquemáticos en el Quartus que involucren señales vectoriales, es de utilidad tener en cuenta los siguientes ejemplos.

Si se conecta un bus a una compuerta, el Quartus lo interpreta como una operación vectorial generando una compuerta por cada línea del bus, como se muestra en la siguiente figura, en la que ambos circuitos son equivalentes:



Figura 19

El Quartus permite también conectar un bus y una línea simple (escalar) a una misma compuerta, como se ve en la figura 20. En el caso del ejemplo se generan tres compuertas con el escalar (señal b) conectado a las tres compuertas.

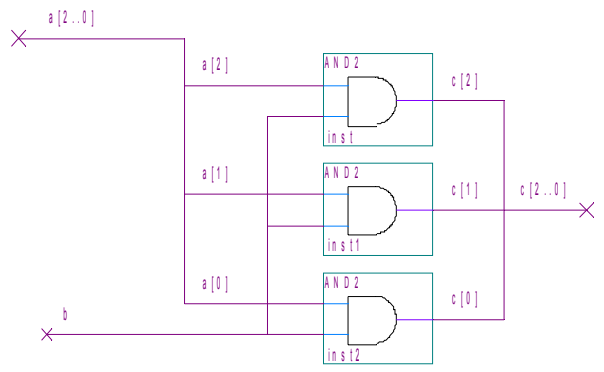
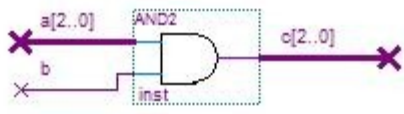


Figura 20



También es posible asignar nombres a diversas señales y definir las conexiones a un bus listando los nombres de las señales individuales separados con comas, como se muestra en el siguiente ejemplo.

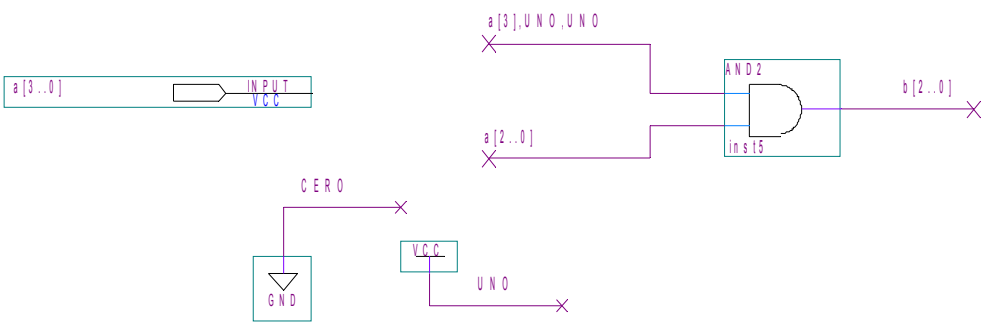


Figura 21