



Universidad  
de Alcalá



## Tutorial Quartus Prime Lite Edition

# ASIGNATURA Electrónica Digital

Grado en Ingeniería en Tecnologías de la  
Telecomunicación

Grado en Ingeniería Electrónica de Comunicaciones

Grado en Ingeniería Telemática

Grado en Ingeniería en Sistemas de  
Telecomunicación

**Universidad de Alcalá**

**Curso Académico 2019/2020**  
**Curso 2º – Cuatrimestre 2º**



**Dpto. Electrónica**





## Índice

1.	Introducción.....	1
2.	Creación de un proyecto en Quartus Prime Lite.....	1
3.	Realización del diseño con el editor de esquemáticos .....	6
3.1.	Creación de un fichero BDF.....	6
3.2.	Añadir texto.....	8
3.3.	Añadir componentes y moverlos.....	8
3.4.	Añadir y eliminar conexiones .....	11
3.5.	Añadir puertos de entrada y salida.....	11
3.6.	Compilación.....	13
3.7.	Compilación funcional.....	14
4.	Simulación del diseño con el editor de ondas .....	15
4.1.	Creación de un fichero VWF (vector de formas de onda).....	15
4.2.	Añadir las señales .....	17
4.3.	Modificación del tamaño de la cuadrícula y tiempo de finalización .....	19
4.4.	Modificación manual del fichero VWF.....	19
4.5.	Simulación funcional y temporal .....	21
4.6.	Agrupación de señales y generación de señales con <i>Count Value</i> y <i>Clock Value</i> .....	22
5.	Programación del CPLD .....	26
5.1.	Asignación de pines.....	26
5.2.	Configuración de pines no usados en triestate .....	28
5.3.	Programación del CPLD.....	29
6.	Descripción del circuito con HDL.....	31
7.	Bibliografía .....	33





## 1. Introducción

El objetivo de este tutorial es guiar al alumno, mediante una práctica sencilla, en el uso de la herramienta Quartus Prime Lite, de manera que sea capaz de realizar un diseño, simularlo y descargarlo en la tarjeta EPM240, basada en un CPLD de Intel de la familia MAX II de Intel.

Quartus Prime Lite es una herramienta de diseño de circuitos digitales para dispositivos programables mediante esquemas lógicos o utilizando lenguajes de descripción hardware, HDLs. Esta herramienta permite diseñar, simular y sintetizar diseños para los dispositivos lógicos programables de Intel. Todas las herramientas en este entorno tienen una interfaz gráfica de usuario que facilita su manejo.

El sistema que se va a implementar es sencillo, ya que lo que se pretende es explicar los pasos necesarios para su implementación. En concreto se desea implementar la función lógica  $Y = A \cdot \bar{B} + \bar{C}$ , empleando dos puertas AND, una puerta OR y dos inversores.

Los pasos que se seguirán para implementar la función lógica anterior en un CPLD son los siguientes:

- Creación de un proyecto y configuración de la ventana de Quartus.
- Realización del diseño con el editor de esquemáticos.
- Simulación del diseño con el editor de ondas.
- Programación del CPLD.

Al final del tutorial se indican los cambios que habría que hacer para diseñar un circuito utilizando el lenguaje VHDL en lugar del editor de esquemáticos.

Este tutorial es una adaptación del tutorial “*Quartus Tutorial with Basic Graphical Gate Entry and Simulation*”, del Dr. Eric M. Schwartz, 22-Jan-2018, Department of Electrical & Computer Engineering, University of Florida, [https://mil.ufl.edu/3701/docs/quartus/quartus\\_18.0\\_tutorial.pdf](https://mil.ufl.edu/3701/docs/quartus/quartus_18.0_tutorial.pdf)

## 2. Creación de un proyecto en Quartus Prime Lite

El primer paso consiste en la creación de un proyecto nuevo, siguiendo los pasos que se detallan a continuación:

1. Con el explorador de archivos cree una carpeta para almacenar el proyecto, los esquemas y los ficheros de simulación, llamado para este caso *tutorial*.
2. Ejecute Quartus Prime Lite.
3. Abra el asistente para crear un nuevo proyecto (*New Project Wizard*) desde el ícono de la ventana *Home*, o bien desde *File -> New Project Wizard...* (Figura 1).

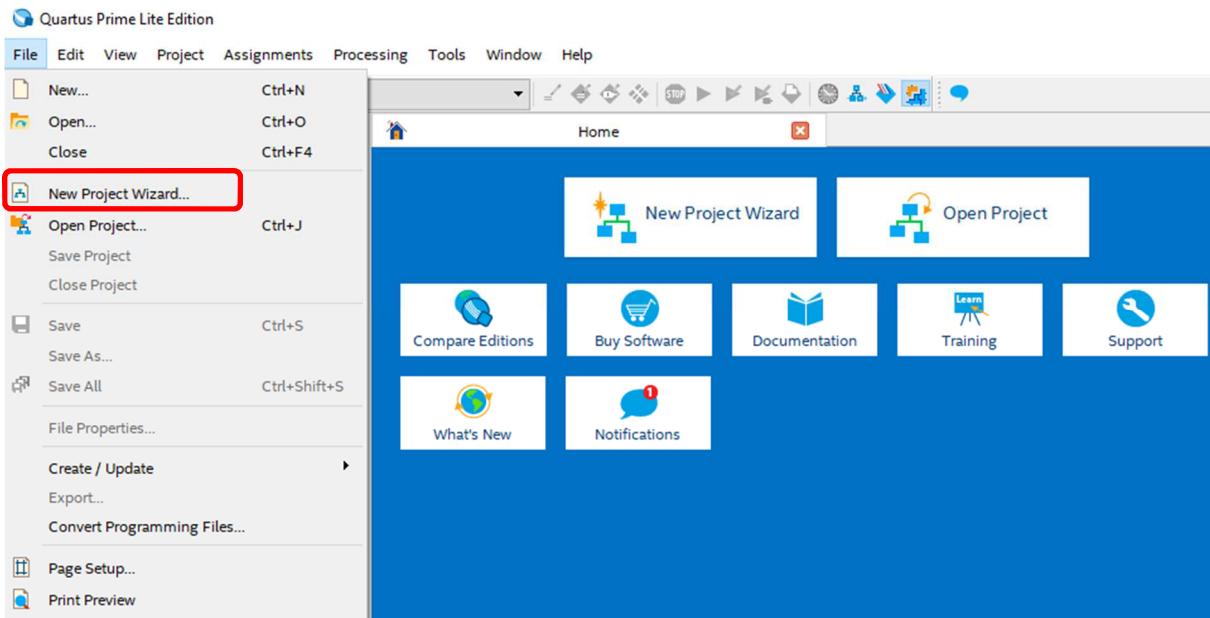


Figura 1. Abrir el asistente de creación de un proyecto.

4. Aparecerá una primera ventana de información que se puede desactivar para que no se muestre en lo sucesivo. Se hace clic sobre *Next*.
5. A continuación, se selecciona la carpeta del proyecto que se ha creado en paso 1. Se le da al proyecto el mismo nombre que la carpeta, que será también el nombre de la entidad principal de este proyecto (Figura 2). Se hace clic en *Next*.

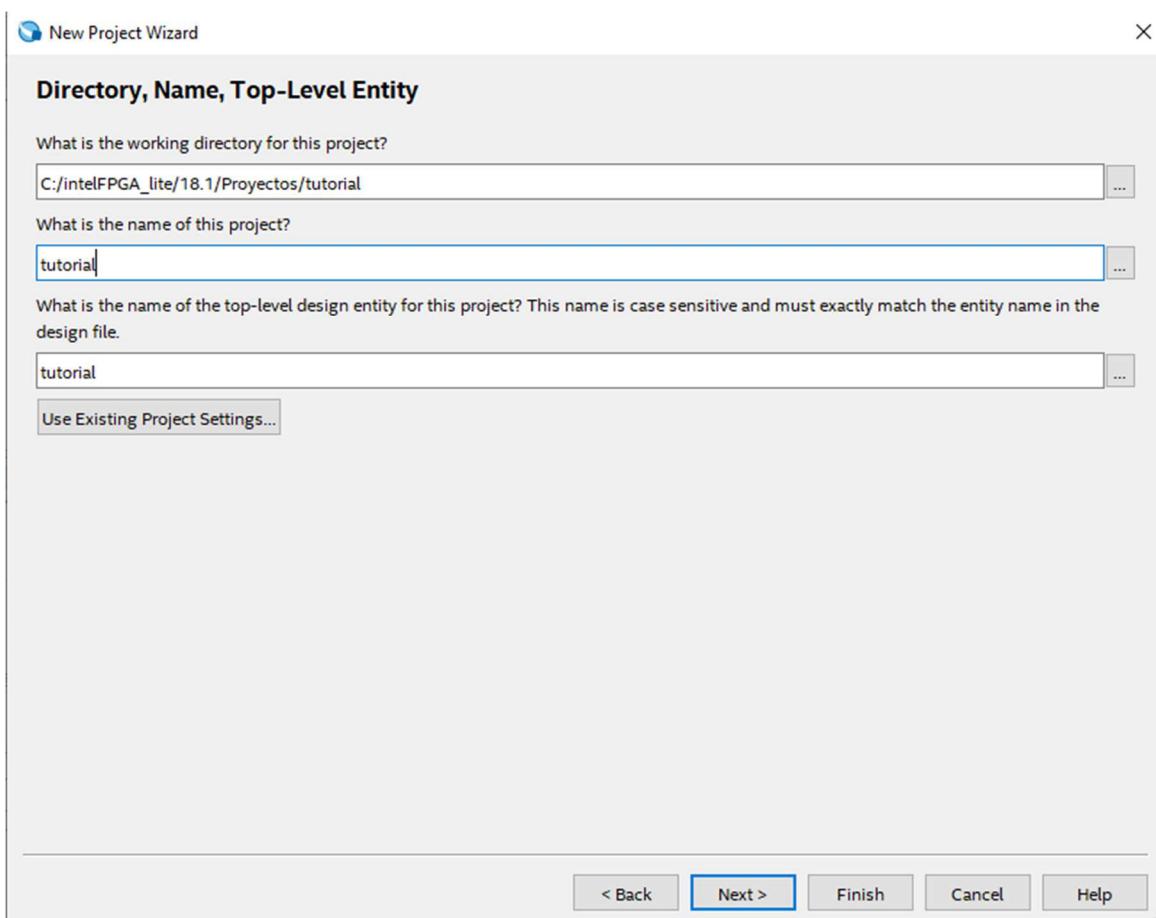
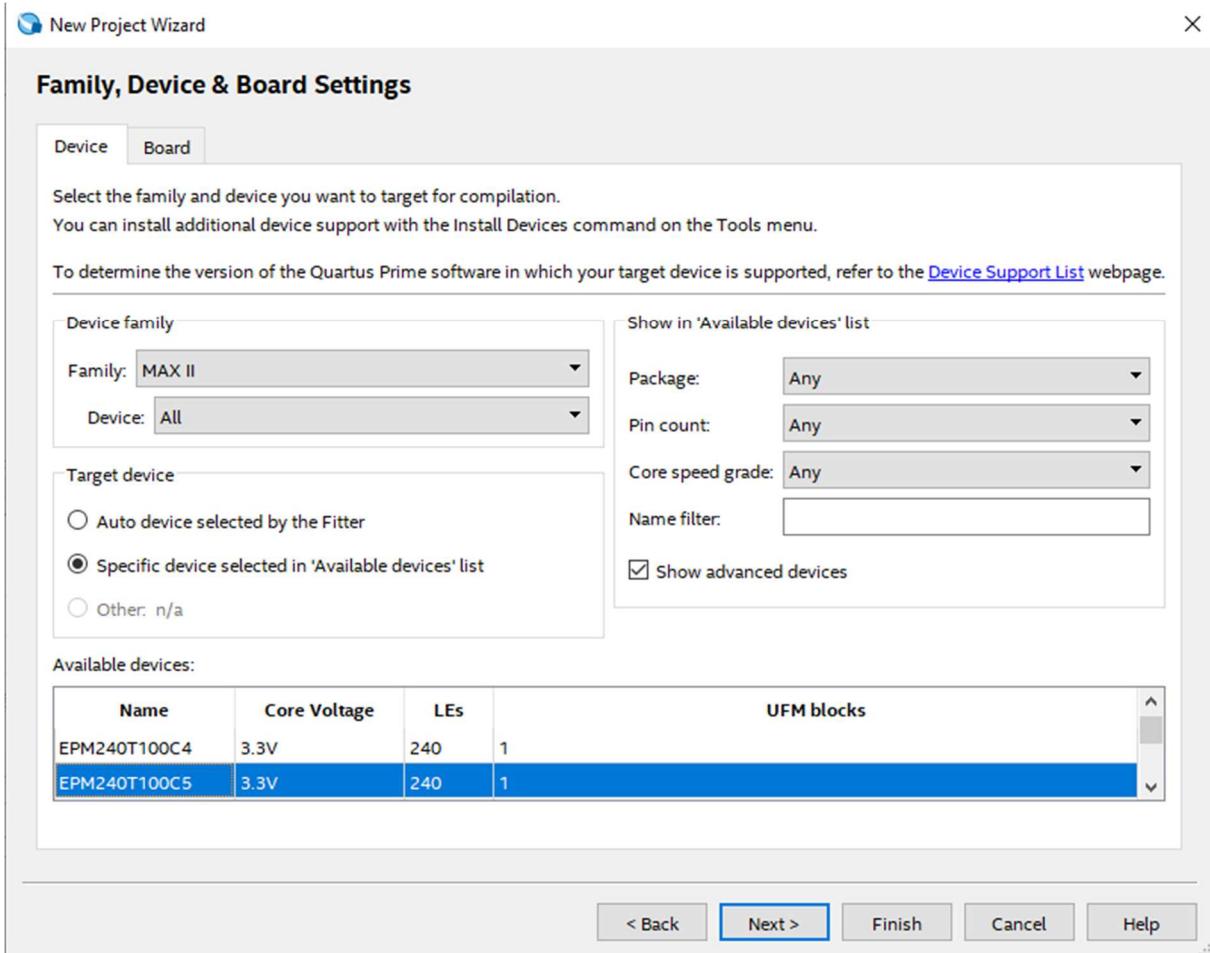


Figura 2. Ventana *Directory, Name, Top-Level Entity*.

6. En la ventana siguiente (*Project Type*) se deja marcada la opción de que cree un proyecto vacío (*Empty project*) y se pulsa *Next*.
7. En la ventana siguiente (*Add Files*) no se requiere ningún cambio y se hace clic en *Next*.
8. En la ventana *Family, Device & Board Settings* se selecciona el dispositivo que se va a utilizar en el laboratorio. En el campo *Family* se selecciona la familia MAX II, y en la lista de dispositivos disponibles se selecciona el EPM240T100C5 (Figura 3). Se hace clic en *Next*.



**Figura 3.** Ventana *Family, Device & Board Settings*.

9. En la ventana *EDA Tool Settings* (Figura 4) se selecciona como herramienta de simulación *ModelSim-Altera*, y en formato, *Format(s)*, se escoge VHDL y se hace clic en *Next*.



New Project Wizard

X

### EDA Tool Settings

Specify the other EDA tools used with the Quartus Prime software to develop your project.

EDA tools:

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synth...	<None>	<None>	<input type="checkbox"/> Run this tool automatically to synthesize the current design
Simulation	ModelSim-Altera	VHDL	<input type="checkbox"/> Run gate-level simulation automatically after compilation
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

< Back    Next >    Finish    Cancel    Help

**Figura 4.** Ventana *EDA Tool Settings*.

10. Para finalizar, en la ventana *Summary*, se muestra un resumen de las opciones del proyecto que se han seleccionado (Figura 5). Al hacer clic en *Finish* se crea el proyecto.

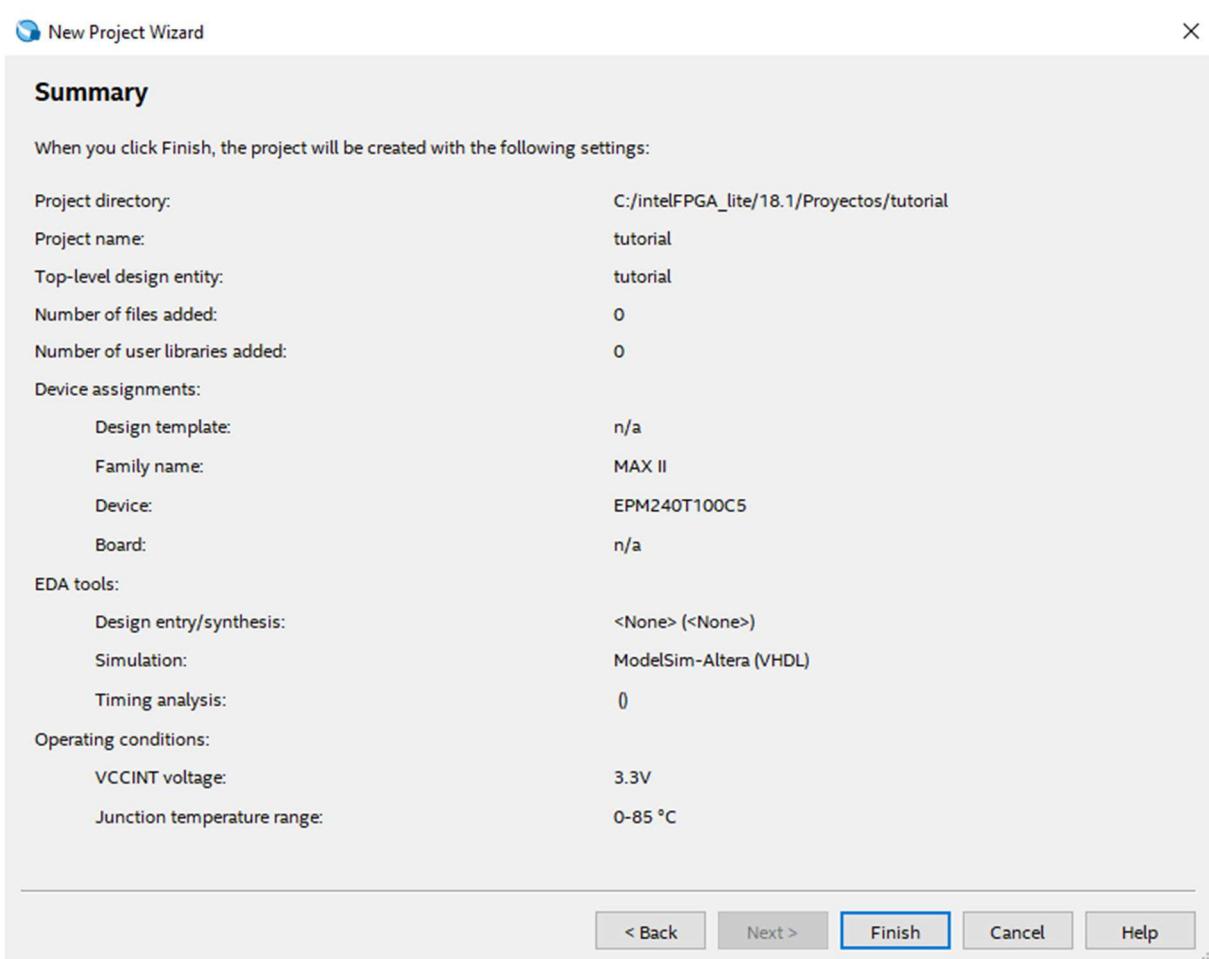


Figura 5. Ventana *Summary*.

Una vez creado el proyecto es necesario configurar las ventanas visibles. Para ello se accede a *View* -> *Utility Windows*, y se debe comprobar que están seleccionadas las ventanas *Project Navigator*, *Messages* y *Tasks* (Figura 6).

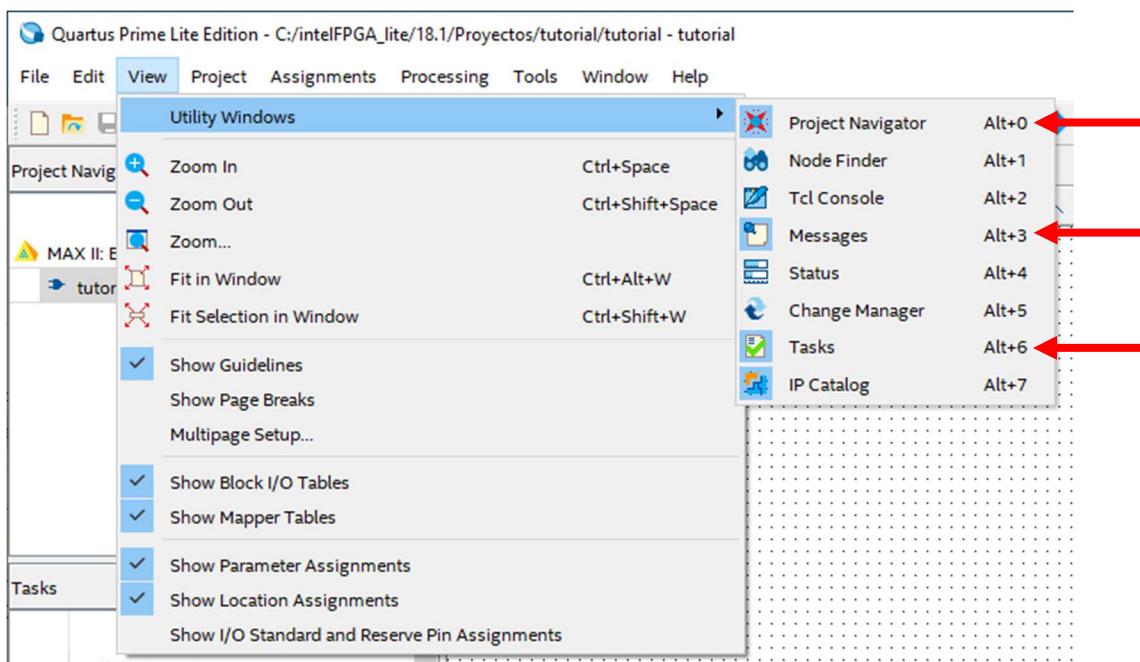
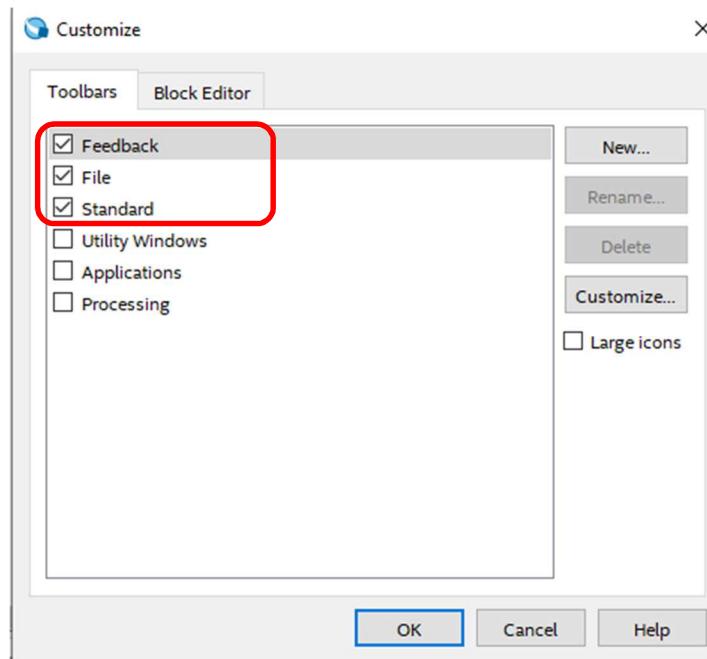


Figura 6. Configuración ventanas visibles.

A continuación, se comprueba que están accesibles algunas de las barras de herramientas, para ello se accede al menú *Tools -> Customize*, y se comprueba que están marcadas las opciones *Feedback, File y Standard*. (Figura 7).



**Figura 7.** Configuración de las barras de herramientas.

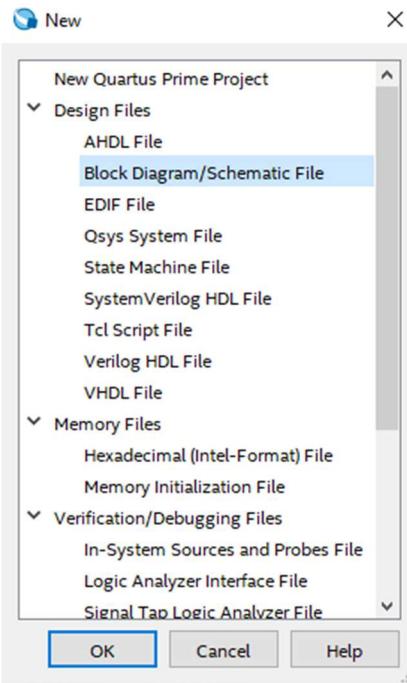
### 3. Realización del diseño con el editor de esquemáticos

La realización de un diseño con esquemáticos se puede dividir en las siguientes fases:

1. Creación de un fichero BDF (*Block Diagram File*).
2. Añadir texto.
3. Añadir componentes y moverlos.
4. Añadir y eliminar conexiones.
5. Añadir puertos de entrada y salida.
6. Compilación.
7. Compilación solo para simulación funcional.

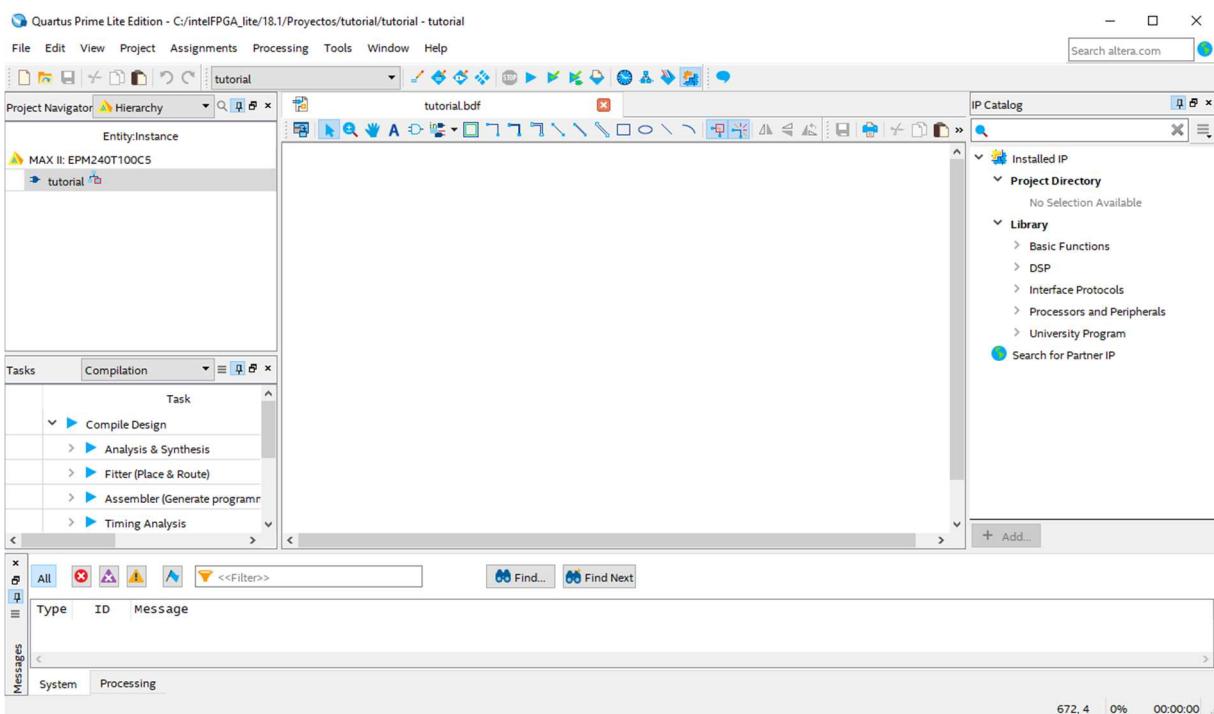
#### 3.1. Creación de un fichero BDF

1. Una vez creado el proyecto se debe crear un fichero para el esquemático. Para ello se accede a *File -> New* y en la ventana que se abre se selecciona *Block Diagram/Schematic File* y se hace clic en *OK* (Figura 8).



**Figura 8.** Selección del tipo de fichero a crear.

2. Ya se ha creado el proyecto y un fichero de esquemático, llamado por defecto Block1.bdf. A continuación, se guarda el fichero esquemático con el nombre tutorial.bdf, desde *File -> Save As...*
3. La cuadrícula de puntos se puede desactivar desde el menú *View -> Show Guidelines*. Finalmente la ventana de Quartus Prime Lite tendrá un aspecto como el mostrado en la Figura 9.

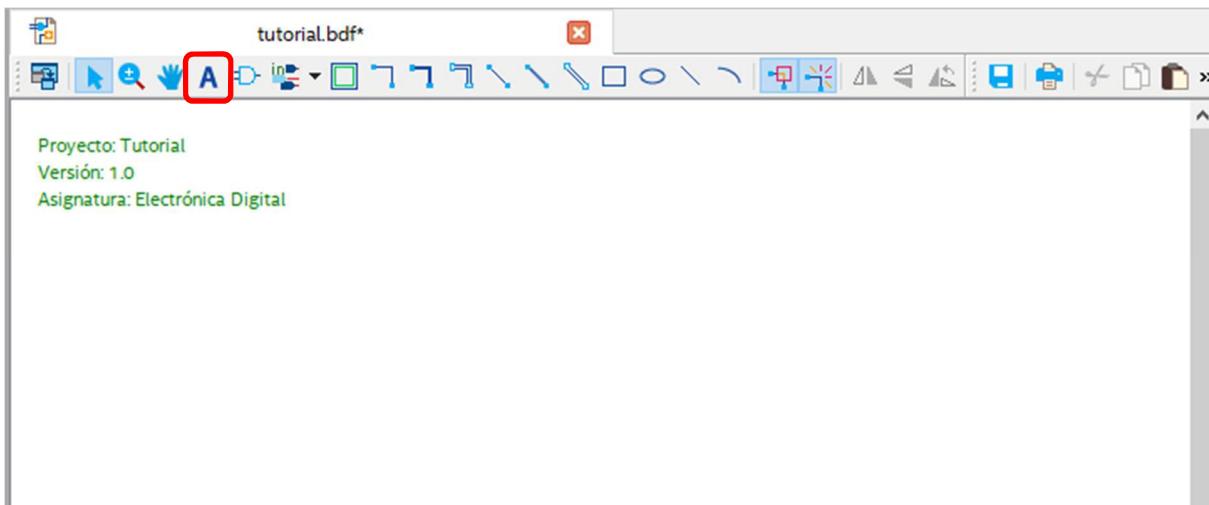


**Figura 9.** Ventana Quartus Prime Lite.

### 3.2. Añadir texto

En muchas ocasiones es necesario añadir un texto al fichero del esquemático para indicar el nombre del autor, el proyecto del que se trata, la versión, etc. Esto se hace a través de los pasos siguientes:

1. De la barra de herramientas de la ventana del esquemático, se selecciona el símbolo “A”.
2. Se hace clic donde se quiera situar el texto, por ejemplo, en la esquina superior izquierda, y se escribe el texto.
3. Se pueden añadir líneas pulsando la tecla ENTER. Para finalizar la entrada de texto se pulsa la tecla ESC. Para este tutorial se han insertado las líneas de texto que se muestran en la Figura 10.

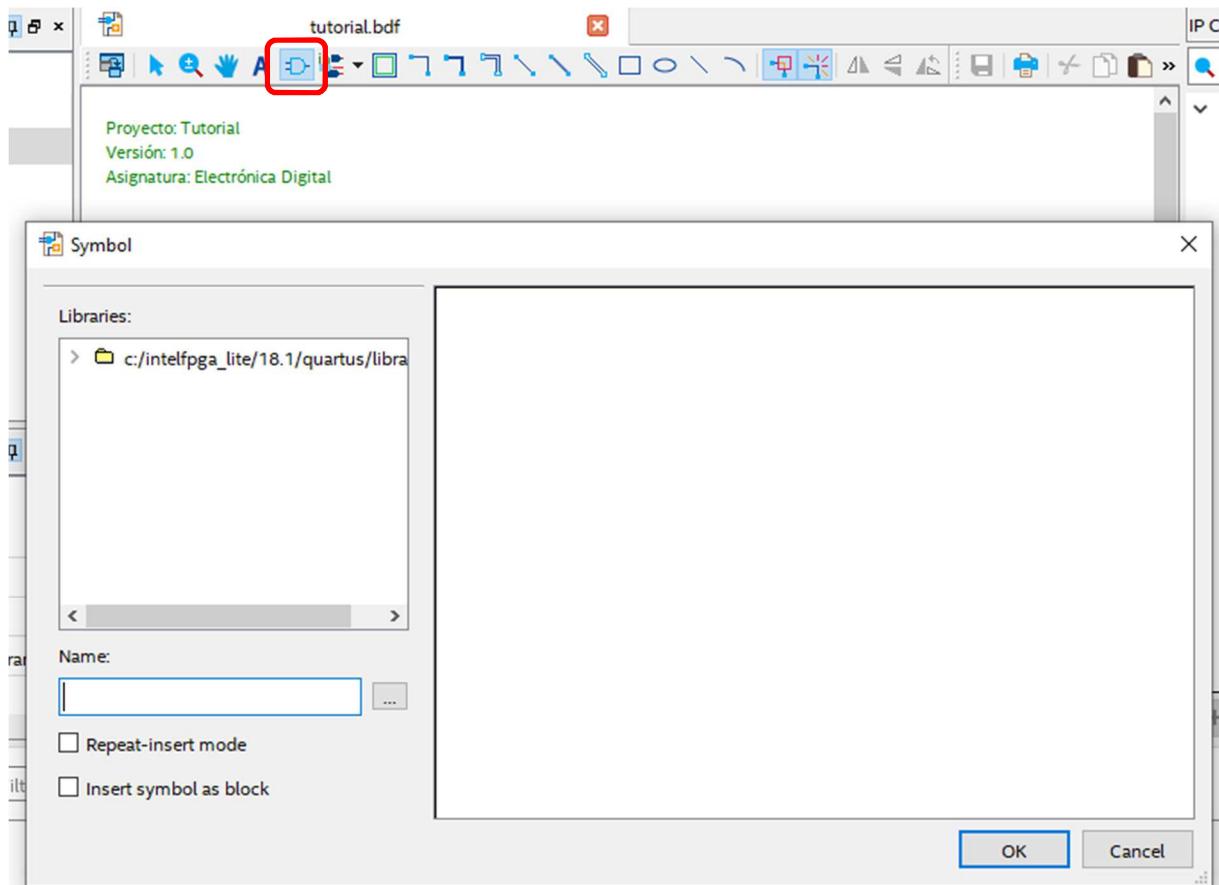


**Figura 10.** Añadir texto a un fichero de esquemático.

### 3.3. Añadir componentes y moverlos

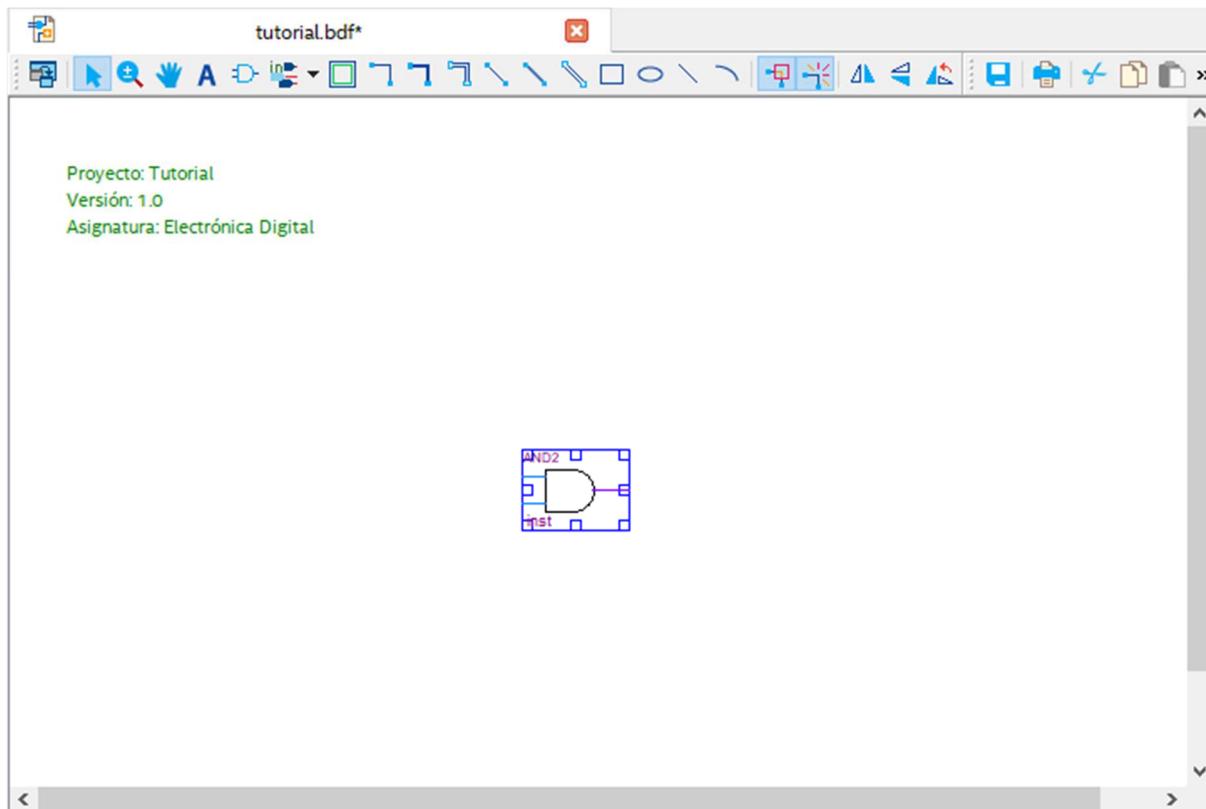
Una vez rotulado el documento del esquemático se va a proceder a insertar los componentes (*Symbols*) que se necesitan para el diseño. Para ello se siguen los pasos siguientes:

1. Se hace doble clic en la ventana de edición del esquemático, o bien se hace clic con el botón derecho y se selecciona la opción *Insert -> Symbol...* Otra opción es pulsar sobre el símbolo de una puerta lógica que hay en la barra de herramientas del editor de esquemáticos, . De cualquiera de las tres maneras se mostrará la ventana de símbolos de la Figura 11.



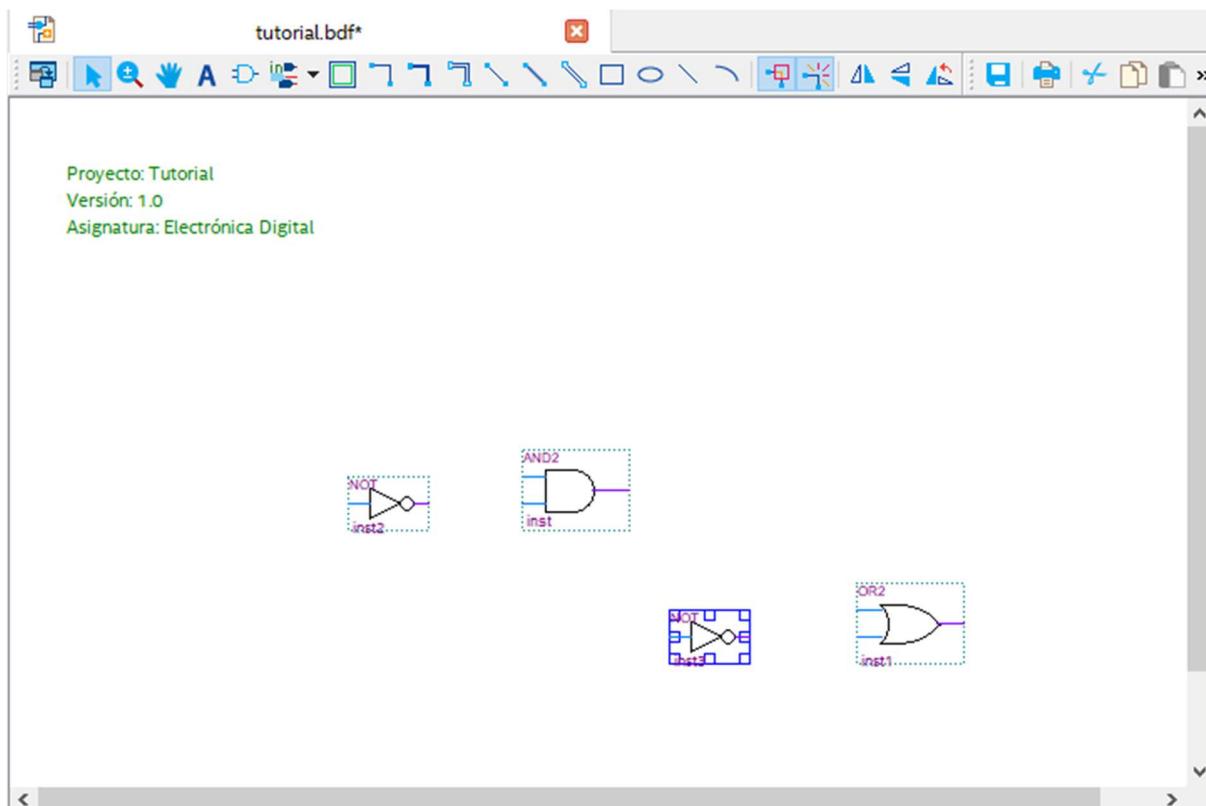
**Figura 11.** Selección del símbolo a insertar.

2. La ventana de símbolos muestra los posibles símbolos (componentes) a utilizar organizados en categorías, como si de un árbol de directorios se tratase. El árbol se despliega cuando se hace clic en el símbolo '>'. Se debe acceder a la carpeta *C:/intelfpga\_lite/18.1/quartus/libraries -> primitives -> logic* y se hace doble clic en el símbolo *and2*. De esa manera se ha seleccionado el componente *and2*. Se sitúa el puntero del ratón en la posición donde se quiera ubicar el componente *and2* y se hace clic (Figura 12). Para salir de la opción de insertar símbolo se pulsa la tecla ESC.



**Figura 12.** Símbolo insertado en un esquemático.

3. Se repite el proceso para situar el símbolo or2 y dos inversores (not), quedando el fichero de esquemático como se muestra en la Figura 13.



**Figura 13.** Símbolos necesarios para el proyecto *tutorial*.

Una vez situados los componentes, estos se pueden mover para colocarlos en la posición más adecuada para el diseño.

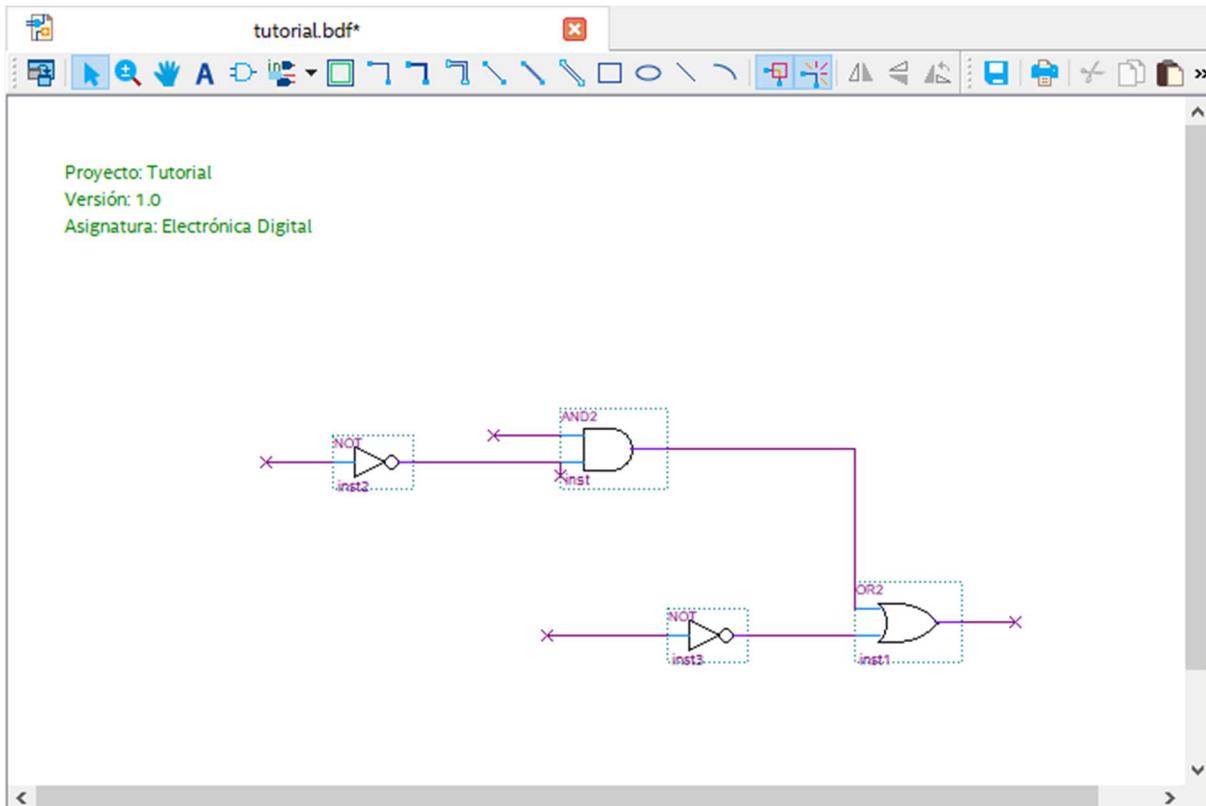
Tras situar los componentes se procederá a guardar el fichero *tutorial.bdf* para no perder el avance.

### 3.4. Añadir y eliminar conexiones

Para conectar los componentes que se han situado en el esquemático se selecciona la herramienta *Orthogonal Node Tool*, , en la barra de herramientas del editor de esquemáticos. A continuación, se hace clic en el extremo del terminal que se desea conectar y se arrastra el ratón, manteniendo el botón izquierdo pulsado, hasta el terminal con el que se desea conectar.

En caso de que se haya realizado una conexión incorrecta, se selecciona con el ratón y se pulsa la tecla de suprimir.

En las entradas y salidas del circuito que se conectarán a pines del CPLD se trazará una conexión con el extremo al aire, como se muestra en la Figura 14.



**Figura 14.** Conexiones entre los símbolos del esquemático.

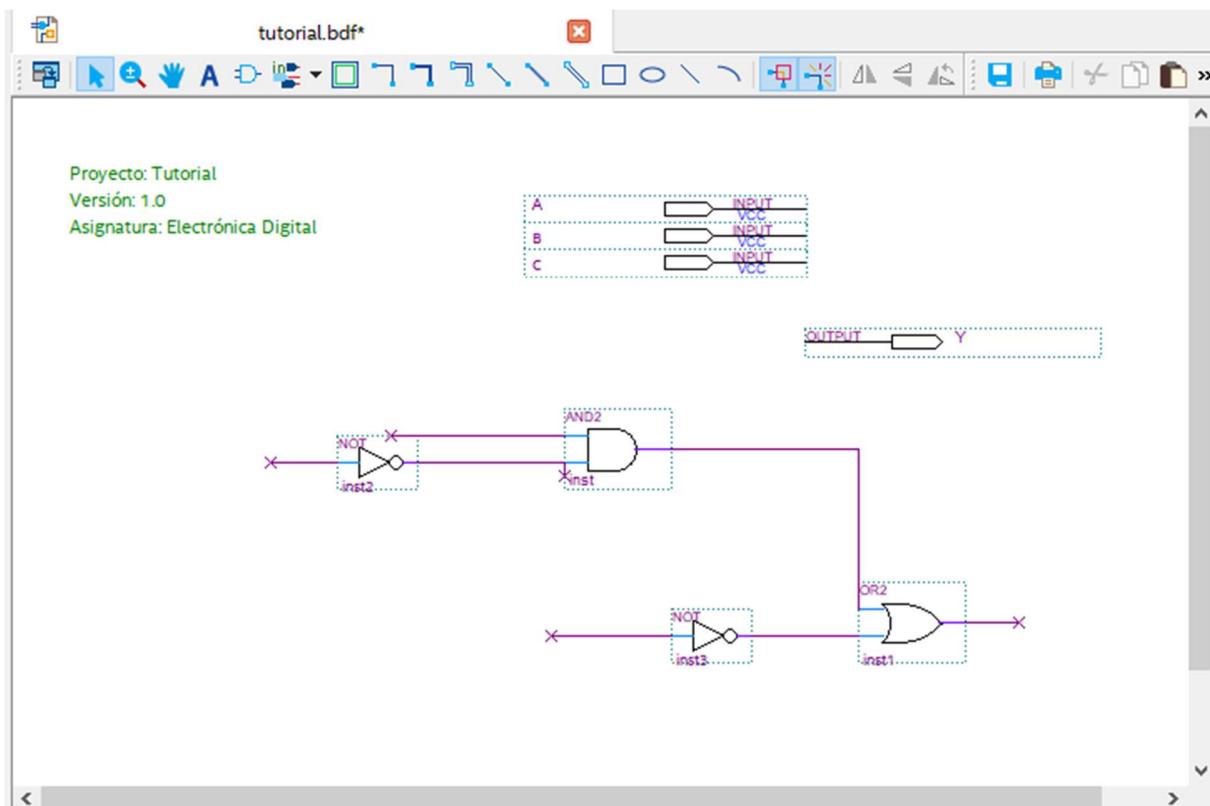
### 3.5. Añadir puertos de entrada y salida

Para conectar las señales del circuito con el exterior por medio de los pines del CPLD es necesario insertar unos componentes denominados puertos, *ports*, o pines, *pins*. Estos puertos pueden ser de entrada, salida o bidireccionales.

Los puertos se insertan de la misma manera que otros componentes, salvo que en esta ocasión no están en la carpeta *logic*, si no en la carpeta *pin*. Otra forma más directa de añadir los puertos es desde la herramienta Pin Tool, , desde donde se puede seleccionar también el tipo de puerto a insertar: entrada, salida o bidireccional.

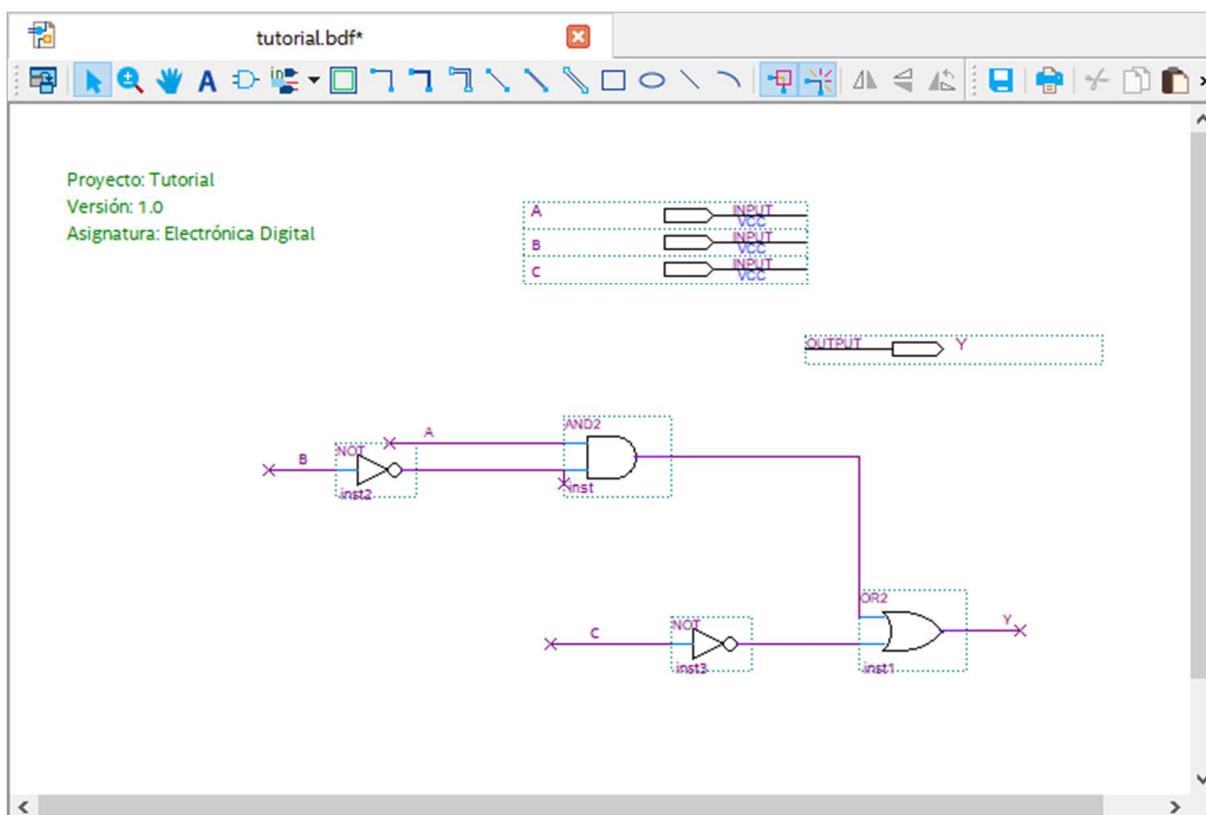
Los puertos se pueden situar próximos a cada una de las señales que se tienen que conectar a los pines del CPLD, o bien todos juntos en una zona del esquemático, para posteriormente relacionarlos con etiquetas. En este tutorial se ha optado por situar todos los puertos juntos. Para ello se procede de la siguiente manera:

1. Se selecciona el símbolo *input* y se sitúan 3 puertos input en la parte superior del esquemático.
2. Se hace doble clic en el nombre del primer pin, *pin\_name1*, y se le cambia el nombre a 'A'. Se repite este proceso para los otros dos puertos, llamándolos 'B' y 'C'.
3. Se inserta un puerto de salida, se hace doble clic en el nombre del puerto y se cambia a 'Y'. (Figura 15)



**Figura 15.** Puertos añadidos al esquemático.

4. A continuación, se etiquetan las conexiones de las puertas que se han dejado al aire para relacionarlas con los puertos. Para ello se hace clic en la conexión de la entrada de la puerta and2 que se ha dejado al aire y se teclea 'A'. Se selecciona conexión de la entrada del inversor conectado a la puerta and2 y se teclea 'B'. Se selecciona conexión de la entrada del inversor conectado a la puerta or2 y se teclea 'C'. Se selecciona conexión de la salida de la puerta or2 y se teclea 'Y', quedando el esquema como muestra la Figura 16.



**Figura 16.** Asociación entre los puertos y señales del circuito mediante etiquetas.

### 3.6. Compilación

Una vez finalizado el diseño es necesario compilarlo (sintetizarlo). Para realizar esta tarea existen varias posibilidades. Una de ellas es, desde la ventana de tareas, hacer doble clic en *Compile Design*. Otra es hacer clic en el símbolo del triángulo, en la barra *Standard*. La tercera es desde la barra de menús, en *Processing -> Start Compilation*. De cualquiera de las maneras se invoca al compilador, y en caso de que no se produzca ningún error se indicará en la ventana de tareas con ticks (Figura 17).

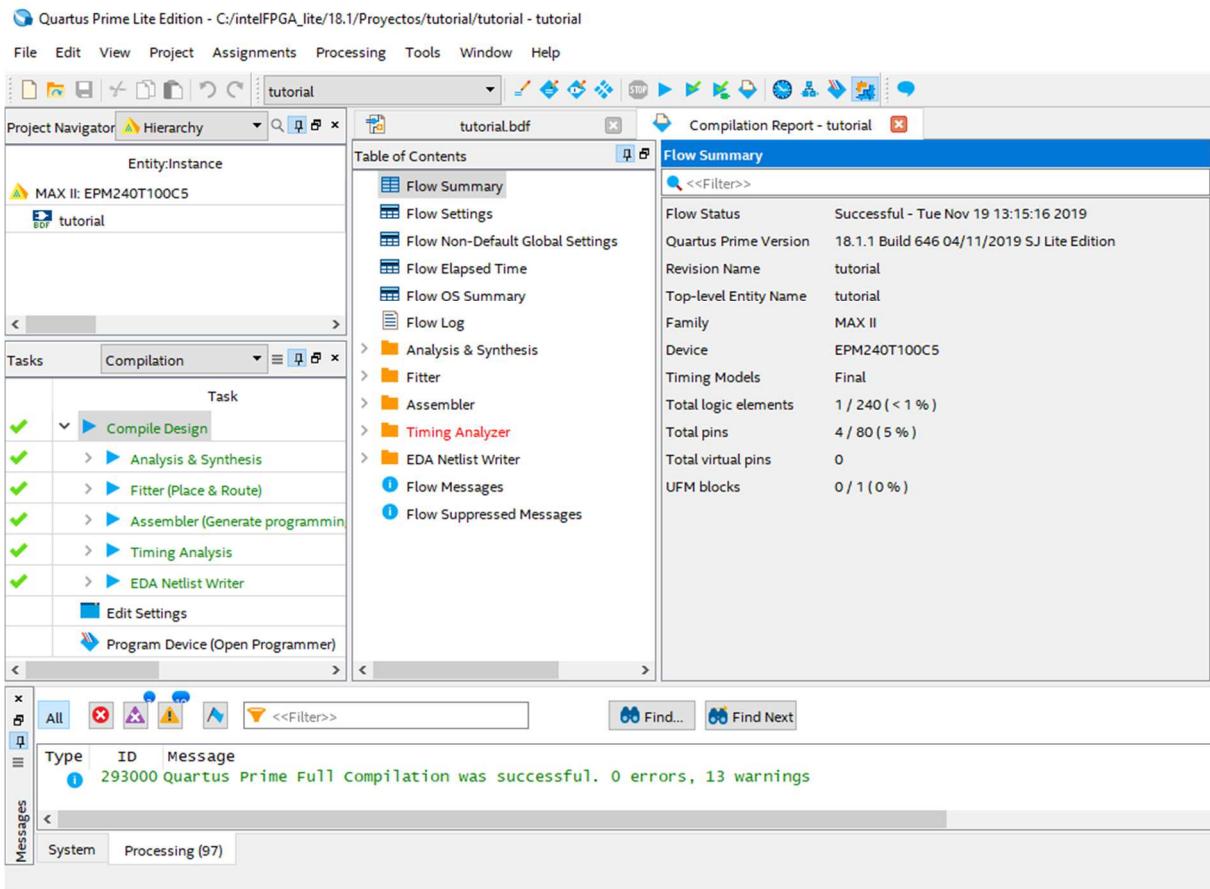


Figura 17. Resultado de la compilación sin errores.

### 3.7. Compilación solo para simulación funcional

La compilación es un proceso que consume bastante tiempo, pero se puede hacer una compilación funcional que es más rápida. Para hacer una compilación funcional se hace doble clic en *Analysis & Synthesis*, dentro de la ventana de tareas (Figura 18).

Se debe tener en cuenta que cuando se hace una compilación funcional únicamente se puede hacer una simulación funcional.

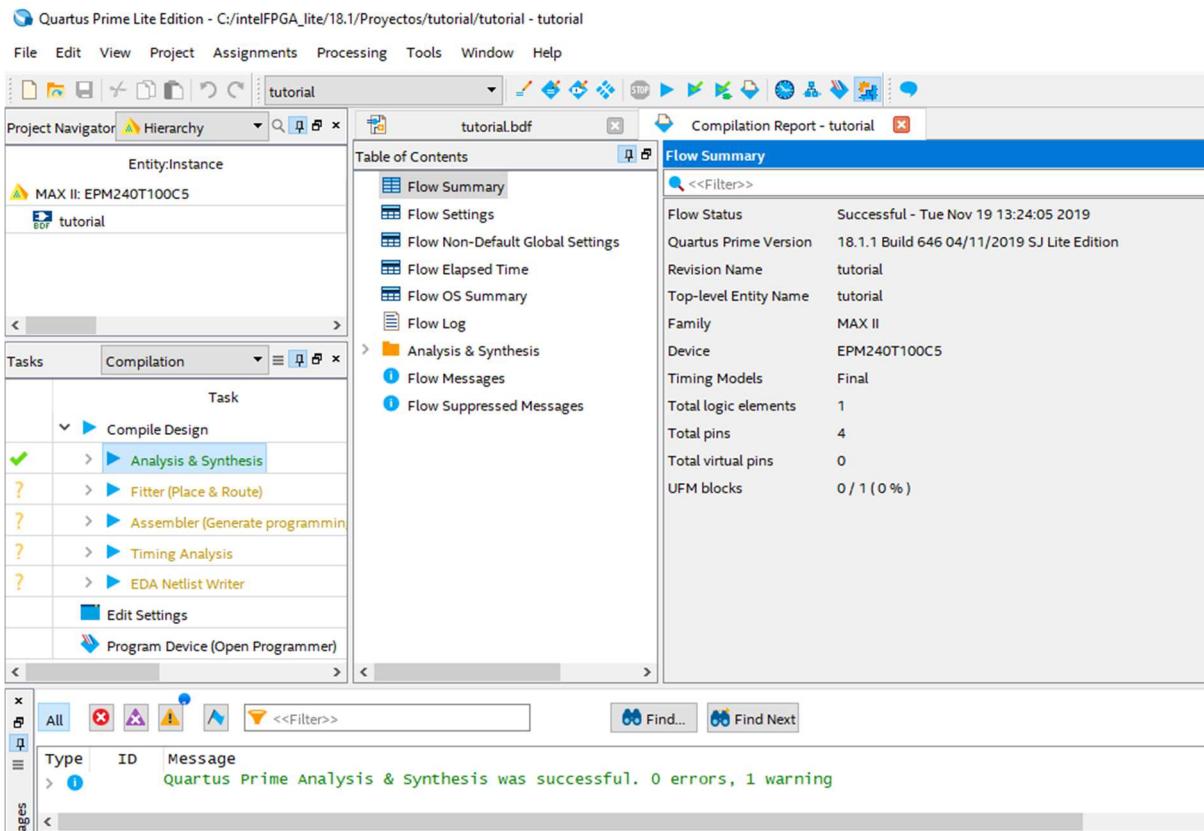


Figura 18. Compilación funcional del proyecto.

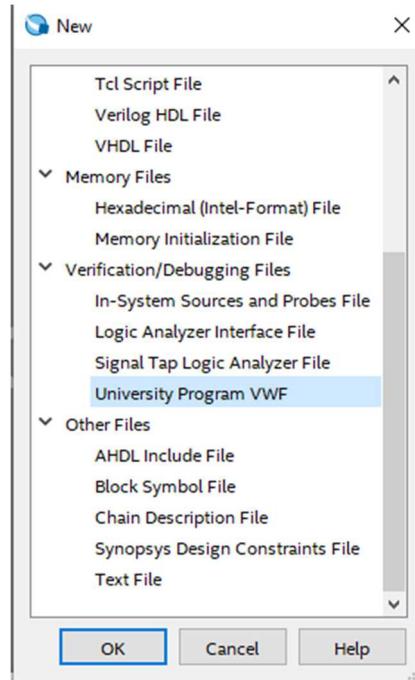
## 4. Simulación del diseño con el editor de ondas

La simulación de un diseño con el simulador de formas de onda requiere de los pasos siguientes:

1. Creación de un fichero VWF (vector de formas de onda).
2. Añadir las señales y/o puertos.
3. Modificación del tamaño de la cuadrícula y tiempo de finalización.
4. Modificación manual del fichero VWF.
5. Simulación funcional y temporal.
6. Agrupación de señales y generación de señales con *Count Value* y *Clock Value*.

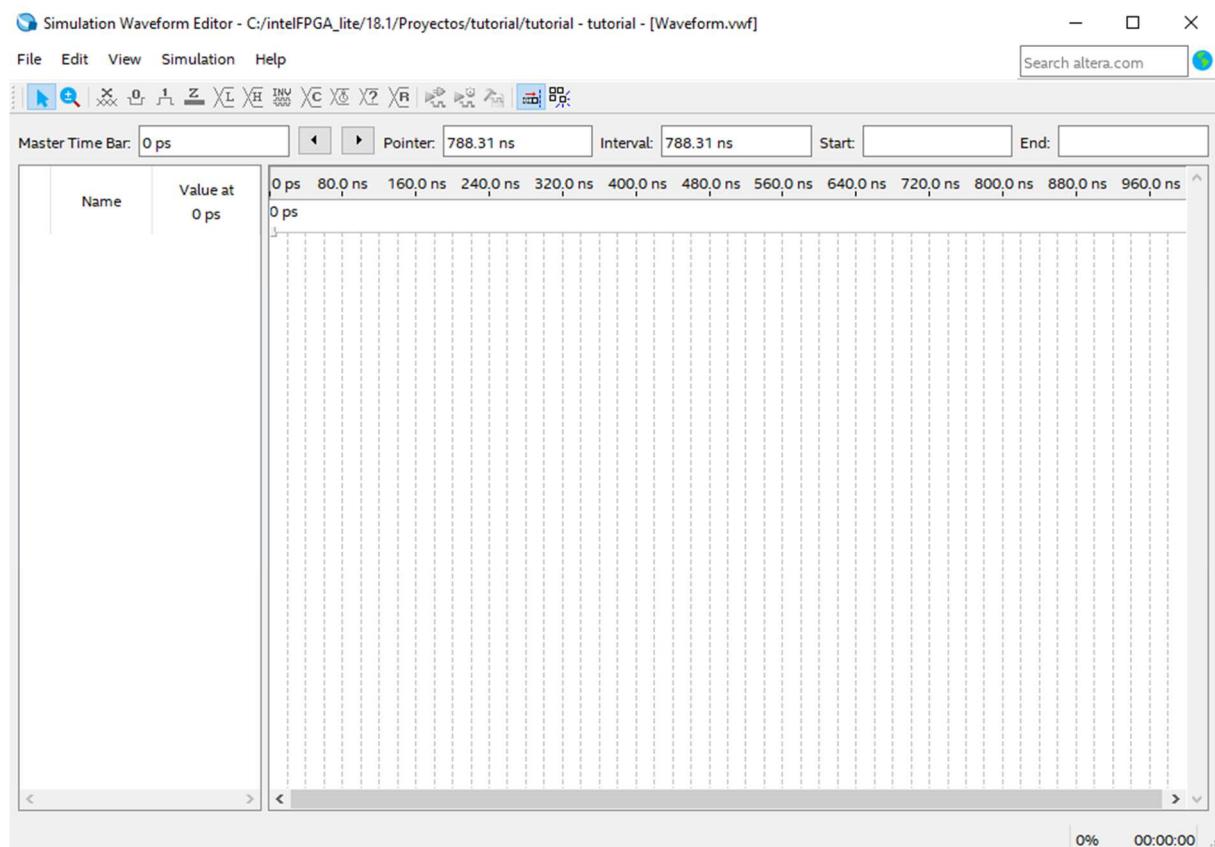
### 4.1. Creación de un fichero VWF (vector de formas de onda)

Para crear el fichero de vector de ondas se selecciona *File -> New* y en el apartado *Verification/Debugging* se selecciona *University Program VWF* y se hace clic en *OK* (Figura 19).



**Figura 19.** Nuevo fichero *University Program VWF*.

Esta acción abre el simulador de formas de onda, desde el que se realizará la simulación (Figura 20). A continuación, se guarda el fichero con el nombre por defecto, *Waveform.vwf*, desde *File -> Save As...* en la misma carpeta del proyecto.



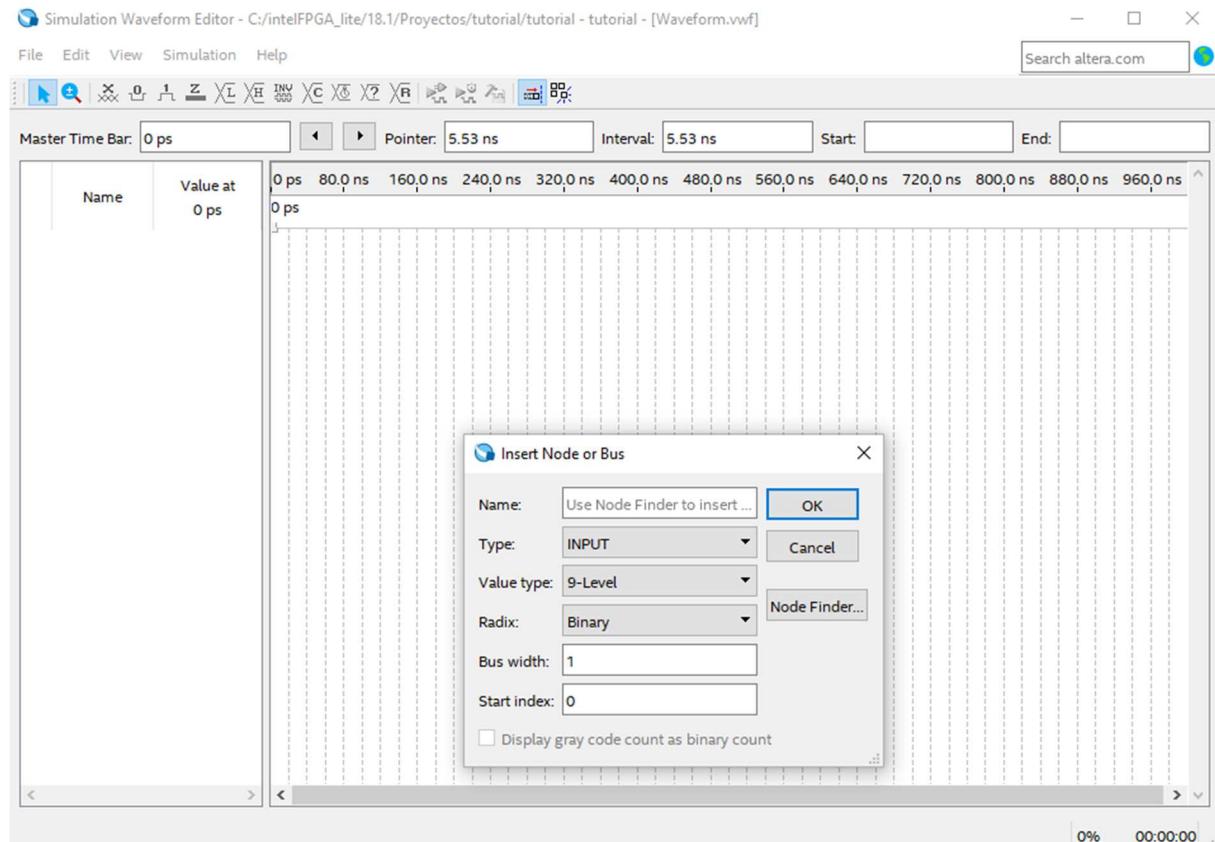
**Figura 20.** Ventana principal del simulador de formas de onda.

Si al simular el diseño se detectan fallos y es necesario realizar modificaciones y volver a simular, para abrir de nuevo la ventana del simulador de formas de ondas desde la ventana principal de

Quartus Prime Lite se debe abrir el fichero .VWF creado anteriormente, desde *File -> Open*, se selecciona en la parte inferior derecha ver todos los archivos (*All Files \*.\**), y se abre el archivo con extensión .VWF.

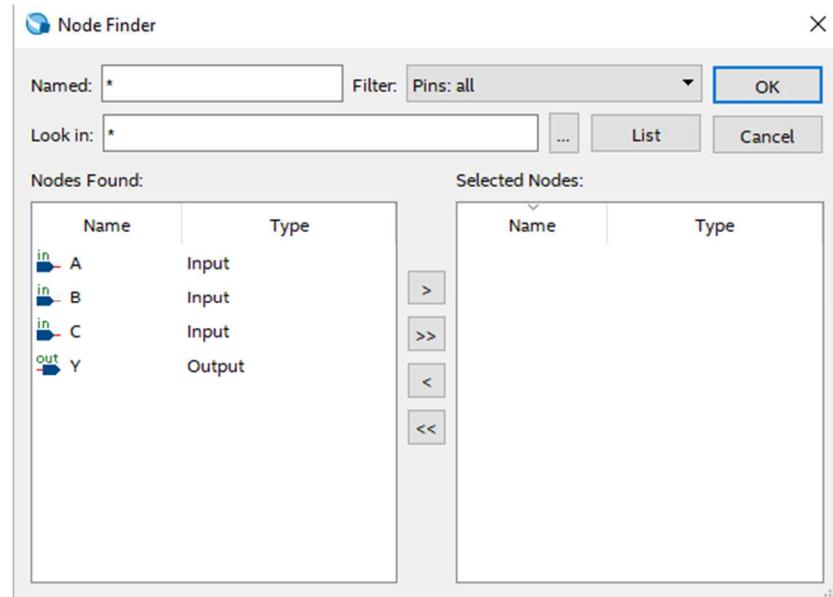
## 4.2. Añadir las señales

Para añadir las señales que se quieren representar en el simulador de formas de onda, se hace doble clic en la parte izquierda, debajo de *Name*. Se muestra la ventana *Insert Node or Bus*, en la que se hace clic en el botón *Node Finder...* (Figura 21).



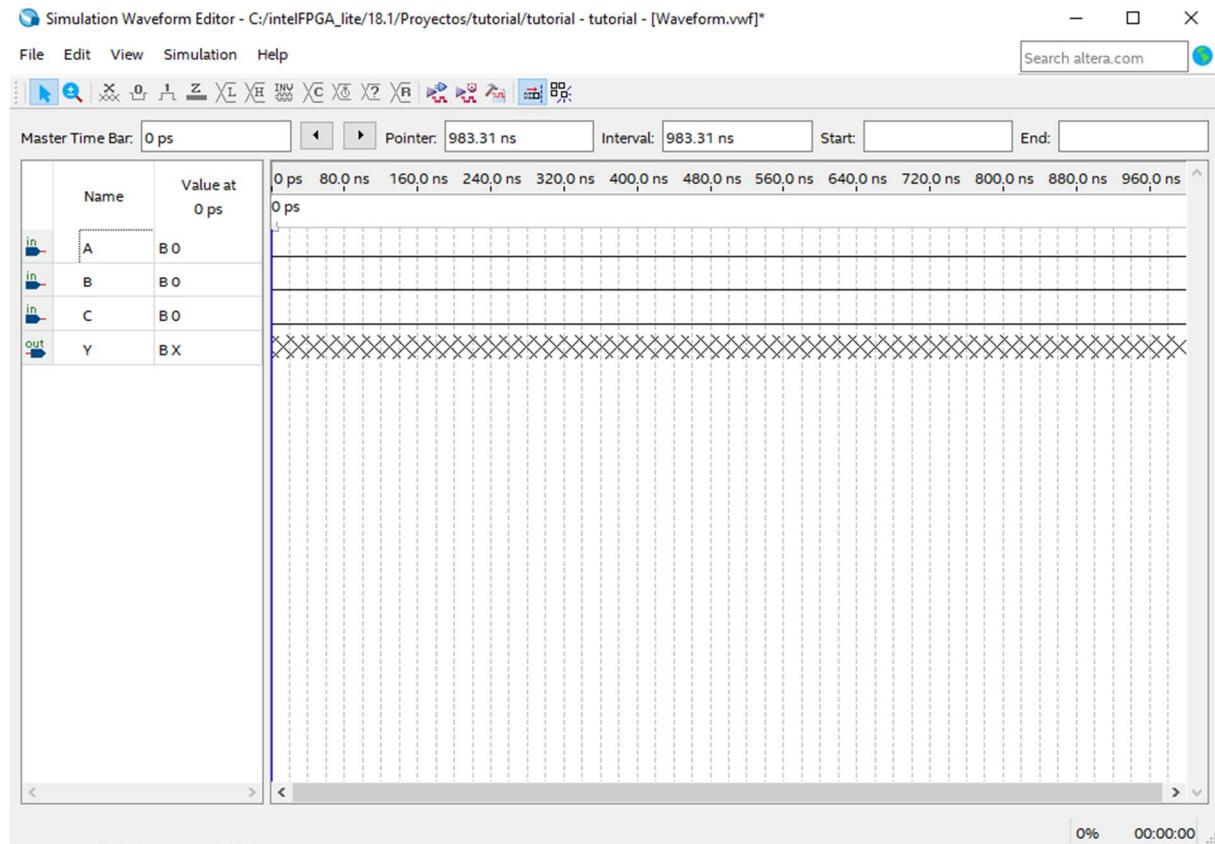
**Figura 21.** Añadir señales a la simulación.

En el campo *Filter* se selecciona *Pins: all* y se hace clic en *List*. Se mostrarán todos los pines (puertos) que se han incluido en el esquemático (Figura 22).



**Figura 22.** Ventana *Node Finder*.

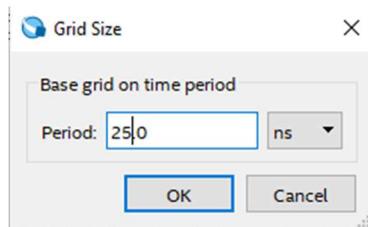
Se deben seleccionar las señales que se quieren representar en el cronograma y llevarlas a la ventana de la derecha, *Selected Nodes*, con los botones “>” (una señal individual o las señales seleccionadas) o “>>” (todas las señales). Para este tutorial se seleccionan *A*, *B*, *C* e *Y*. Cuando se hayan seleccionado las señales se hace clic en el botón *OK* y en la ventana *Insert Node or Bus* que sigue abierta, se hace clic también en *OK*. La ventana del simulador de formas de onda tendrá un aspecto como el mostrado en la Figura 23.



**Figura 23.** Ventana principal con las señales a visualizar.

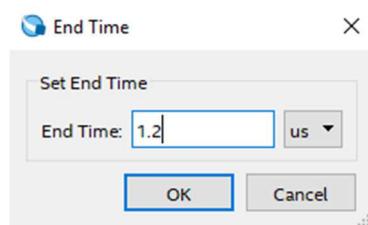
#### 4.3. Modificación del tamaño de la cuadrícula y tiempo de finalización

La escala de tiempos por defecto tiene incrementos de 10 ns, que es muy pequeño para la simulación ya que los componentes tienen un tiempo de propagación de unos 10 ns. Por ello en primer lugar se debe cambiar el incremento de tiempo (*grid size*). Desde el menú se accede a *Edit -> Grid Size*, y se modifica a 25 ns (Figura 24).



**Figura 24.** Configuración del intervalo de simulación.

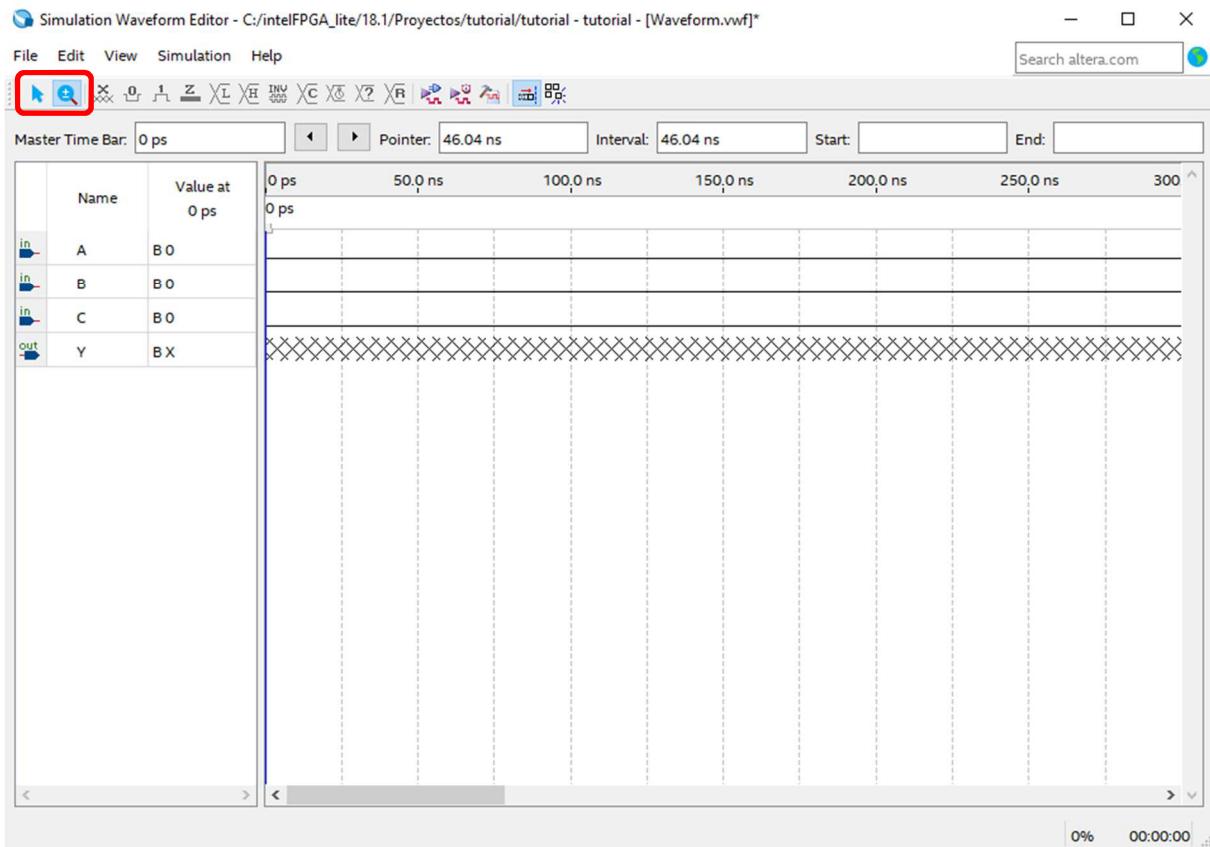
A continuación, se modifica el tiempo de finalización de la simulación. Desde el menú *Edit -> Set End Time...* se cambia el valor de fin de simulación a 1,2 µs (Figura 25).



**Figura 25.** Configuración del tiempo de finalización de la simulación.

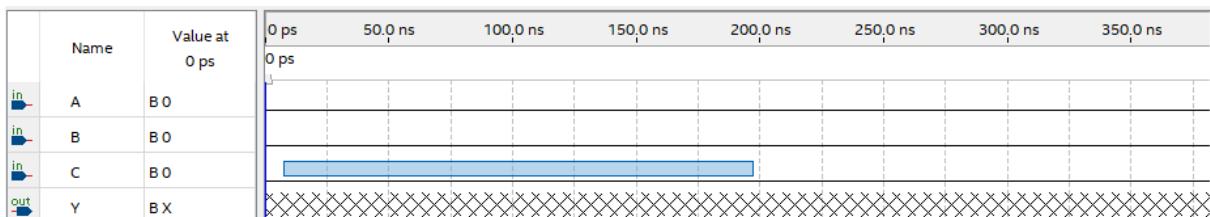
#### 4.4. Modificación manual del fichero VWF

En primer lugar, se debe seleccionar la herramienta lupa,  *Zoom Tool*, y hace clic con el botón izquierdo en los ejes de tiempos de las señales para aumentar la escala hasta que las divisiones de tiempos sean cada 50 ns (Figura 26).

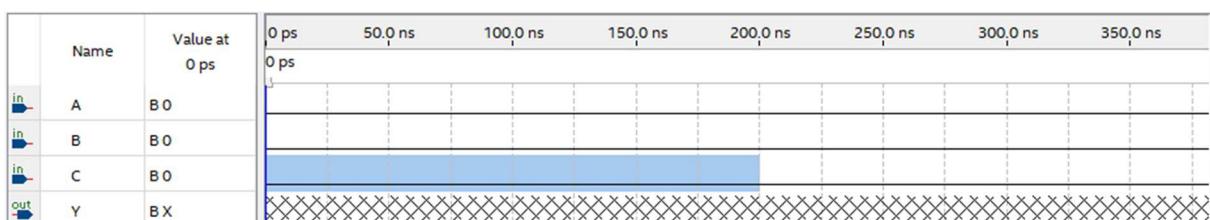


**Figura 26.** Ajuste del eje de tiempos para ver un incremento de 50 ns.

Se selecciona ahora la herramienta puntero, **Selection Tool**, y se hace clic y se arrastra el ratón entre los tiempos 0 ns y 200 ns del eje de la señal *C* (Figura 27). Al soltar al botón izquierdo debe quedar resaltado el eje de tiempos de la señal *C* entre 0 ns y 200 ns (Figura 28).

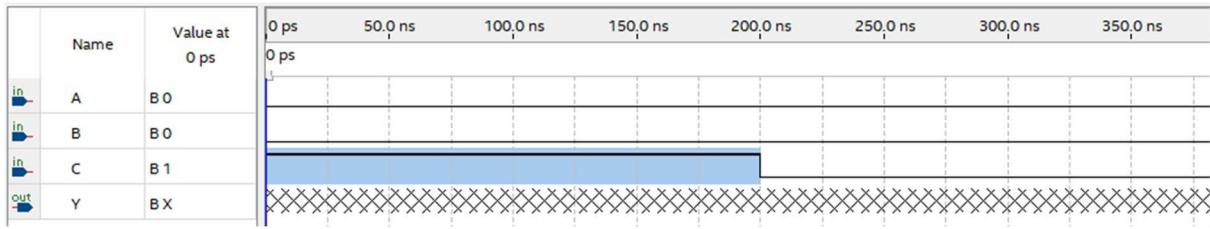


**Figura 27.** Selección del intervalo de tiempo para definir valor de la señal.



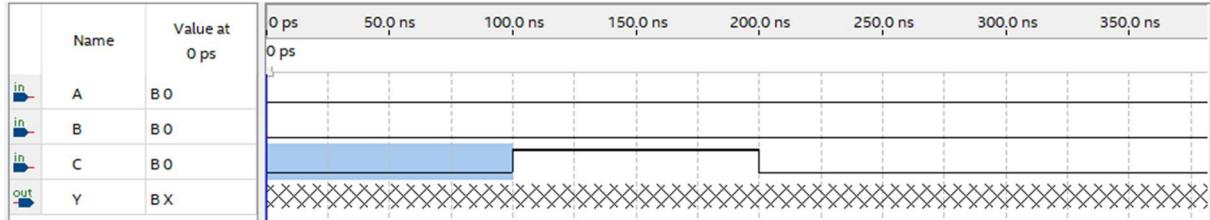
**Figura 28.** Intervalo de tiempo seleccionado.

Se pulsa a continuación el botón de la barra de herramientas para forzar la señal *C* a nivel alto en el intervalo seleccionado (Figura 29).



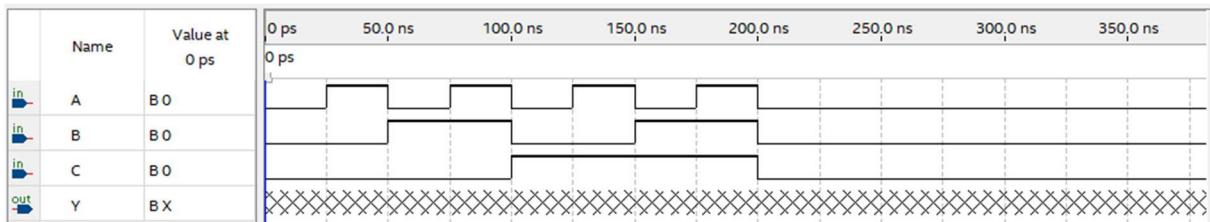
**Figura 29.** Señal definida a nivel alto.

Siguiendo el mismo procedimiento se debe fijar a valor 0 el intervalo entre 0 ns y 100 ns de la señal C (Figura 30).



**Figura 30.** Definición del tiempo a nivel bajo.

Se debe repetir el proceso con las señales A y B para obtener una evolución de las señales como la mostrada en la Figura 31.

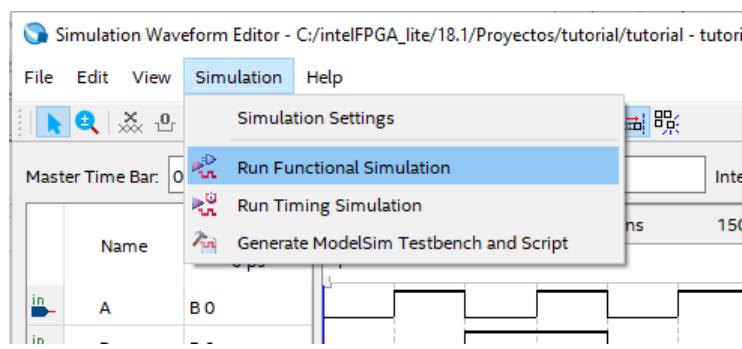


**Figura 31.** Señales de entrada definidas.

Se han definido las señales de entrada para que evolucionen desde 000 hasta 111, siendo A el bit de menor peso. Guarde el fichero *Waveform.vwf* de nuevo para no perder el avance.

#### 4.5. Simulación funcional y temporal

Una vez definidas las señales de entrada se puede iniciar la simulación funcional. Desde el menú se selecciona *Simulation -> Run Functional Simulation* (Figura 32).



**Figura 32.** Ejecución de la simulación.

Se mostrará una ventana del progreso de simulación como la de la Figura 33, y una vez finalizado el proceso se podrán ver los resultados en una nueva ventana de simulador de formas de onda (Figura 34). Se puede comprobar que el valor de la salida en cada instante de tiempo coincide con el resultado de la función lógica que se ha implementado,  $Y=A \cdot \bar{B} + \bar{C}$ .

Simulation Flow Progress

Generating netlist...

```

write_settings_files=off tutorial -c tutorial --
vector_source=C:/intelFPGA_lite/18.1/Proyectos/tutorial/Waveform.vwf --
testbench_file=C:/intelFPGA_lite/18.1/Proyectos/tutorial/simulation/qsim/Waveform.vwf.vt
Info (119006): Selected device EPM240T100C5 for design "tutorial"
Warning (18236): Number of processors has not been specified which may cause overloading on shared
machines. Set the global assignment NUM_PARALLEL_PROCESSORS in your QSF to an appropriate value
for best performance.

Completed successfully.

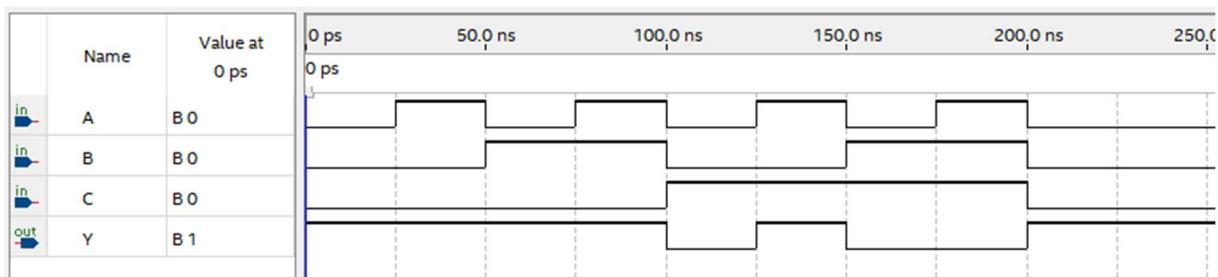
Completed successfully.

**** Generating the functional simulation netlist ****

quartus_edt --write_settings_files=off --simulation --functional=on --flatten_buses=off --
tool=modelsim_oem --format=verilog --
output_directory="C:/intelFPGA_lite/18.1/Proyectos/tutorial/simulation/qsim/" tutorial -c tutorial

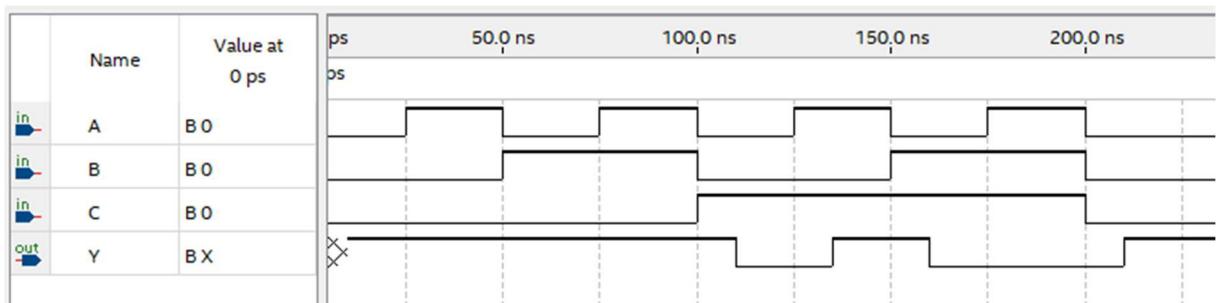
```

**Figura 33.** Compilación de la simulación.



**Figura 34.** Resultado de la simulación funcional.

Para realizar una simulación temporal se accede al menú *Simulation -> Run Timing Simulation*, y una vez finalizado el proceso se puede observar un resultado como el mostrado en la Figura 35.



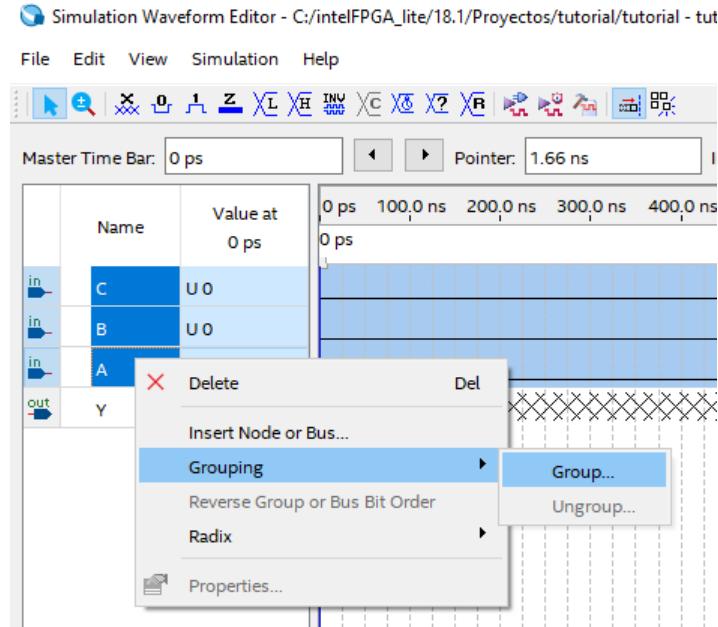
**Figura 35.** Resultado de la simulación temporal.

En la simulación temporal se puede observar cómo la salida tiene un retardo aproximado de 10 ns respecto a los cambios en las entradas.

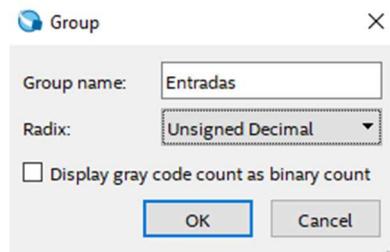
#### 4.6. Agrupación de señales y generación de señales con *Count Value* y *Clock Value*

El editor de formas de onda permite agrupar varias señales formando un bus. Es necesario resaltar que al agrupar un conjunto de líneas se va a considerar como línea de mayor peso la primera de ellas. En este tutorial se ha considerado que la señal *C* es la de mayor peso, por lo que debería estar la primera. Haciendo clic sobre cada una de las señales y moviendo el ratón sin soltar el botón se puede reorganizar las señales del simulador de formas de onda.

Una vez reorganizadas las señales, para agruparlas se seleccionan en la ventana izquierda del simulador de formas de onda manteniendo la tecla *Shift* pulsada las que se desean agrupar y se hace clic con el botón derecho del ratón. En el menú contextual se selecciona *Grouping -> Group* (Figura 36) y en la ventana que aparece se indica el nombre para el grupo de señales y el sistema de numeración a emplear (Figura 37). Para este tutorial se van a llamar *Entradas* y se empleará el sistema de numeración *Unsigned Decimal*.

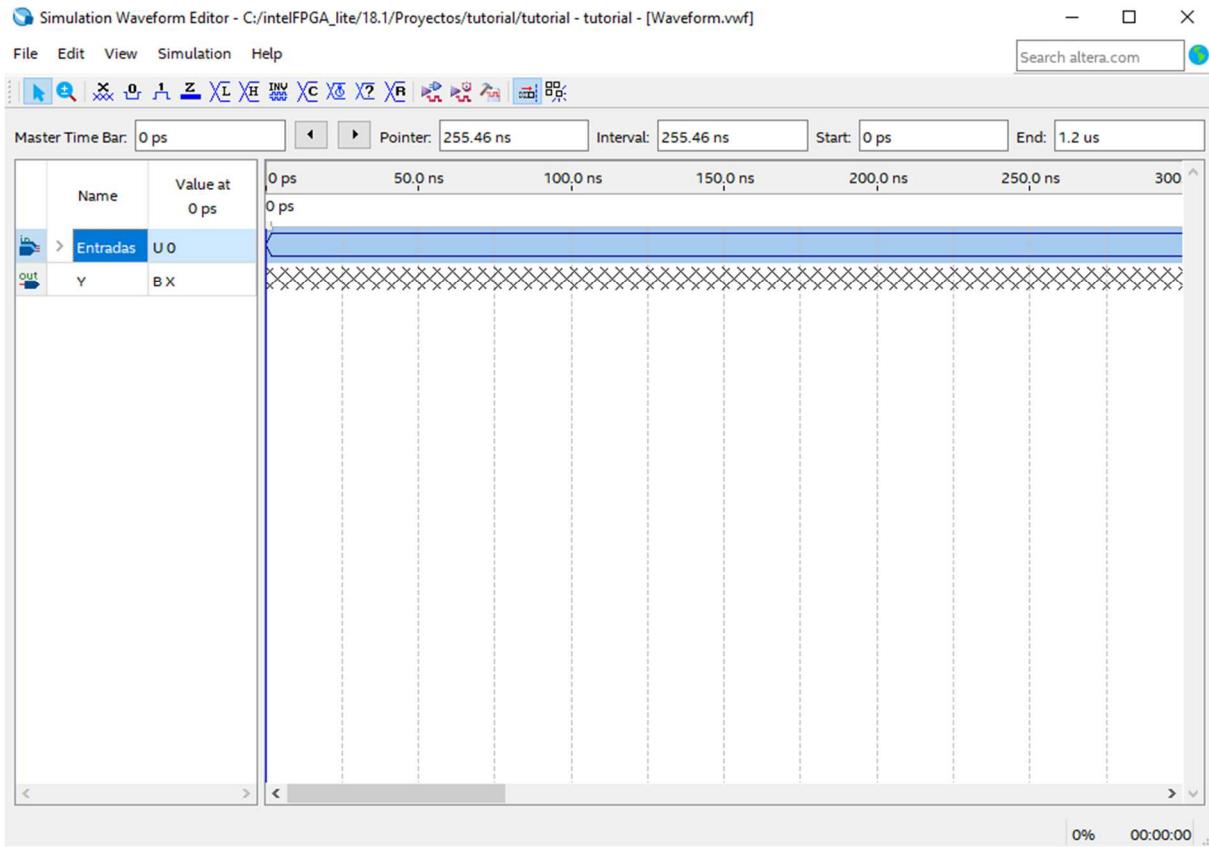


**Figura 36.** Menú de agrupación de señales.



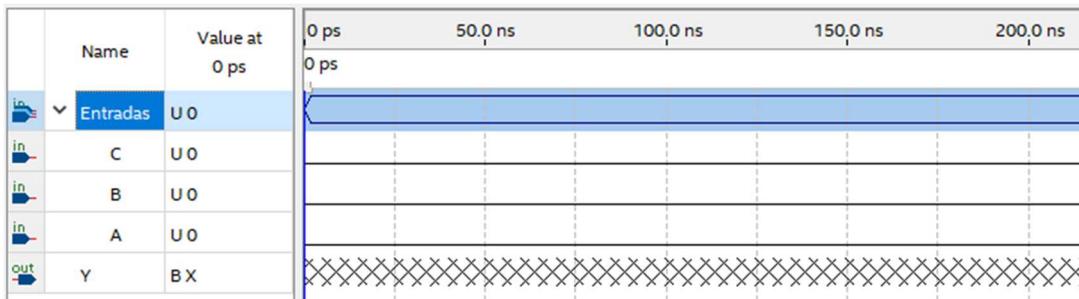
**Figura 37.** Configuración del nombre del bus y sistema de numeración.

Una vez creado el grupo de señales la ventana del editor de ondas tendrá un aspecto como el mostrado en la Figura 38.



**Figura 38.** Ventana principal con las señales agrupadas.

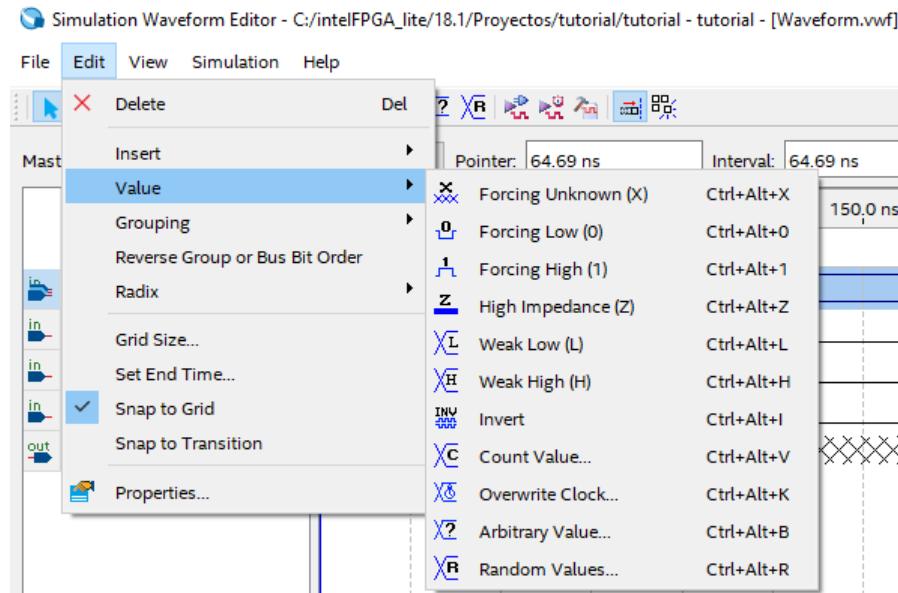
Se pueden expandir las señales que se han agrupado para poder ver el valor de cada una de ellas. Para ello se hace clic en la flecha '›' situada a la izquierda del nombre del bus, *Entradas*.



**Figura 39.** Expansión de las señales agrupadas.

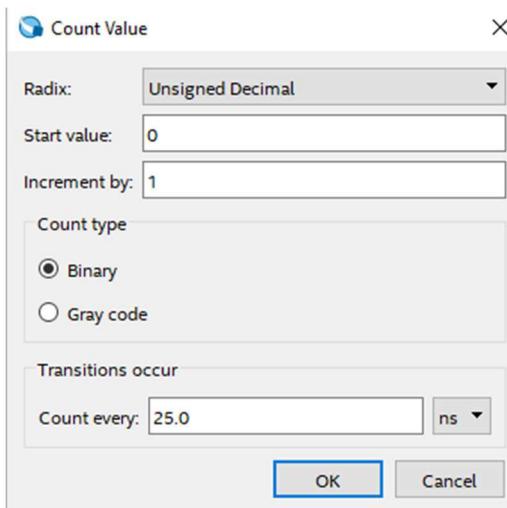
Para probar este sistema combinacional se han introducido los códigos binarios desde 000 hasta 111, manipulando las tres señales de entrada individualmente, pero existe una forma más fácil de hacerlo. En primer lugar, se deben borrar las señales de entrada, para ello se selecciona el bus *Entrada* y se pulsa sobre la herramienta .

A continuación se selecciona el bus de entrada y se pulsa sobre el botón 'C' de la barra de herramientas de la ventana del editor de ondas, . A esta funcionalidad también se puede acceder desde *Edit -> Value -> Count Value...* (Figura 40).

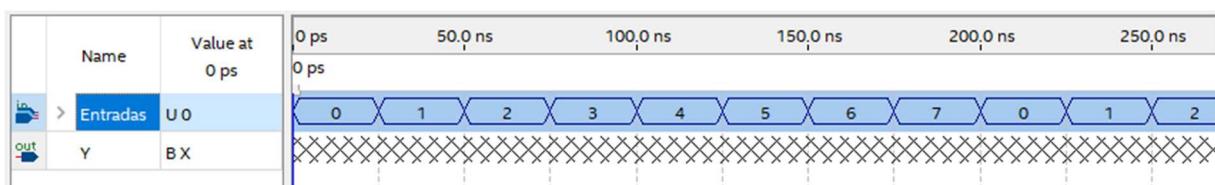


**Figura 40.** Acceso al menú de definición de los valores de entrada en el bus.

En la ventana de configuración de la cuenta se debe indicar cuál es el sistema de numeración, el valor inicial, el incremento, el tipo de cuenta y cada cuánto tiempo se incrementa la cuenta. Para este tutorial se realizará una cuenta en decimal sin signo, desde 0 con incrementos de 1 unidad, y la cuenta se incrementará cada 25 ns (Figura 41). De esta manera se definen las señales de entrada hasta el tiempo de finalización que se había configurado previamente, 1,2  $\mu$ s, (Figura 41)



**Figura 41.** Ventana de definición de los valores de entrada.



**Figura 42.** Bus de entrada con los valores definidos.

Otra funcionalidad que incluye el simulador de formas de onda es generar una señal de reloj mediante la herramienta , en la que se puede configurar el periodo, desplazamiento y ciclo de trabajo de la señal de reloj.

## 5. Programación del CPLD

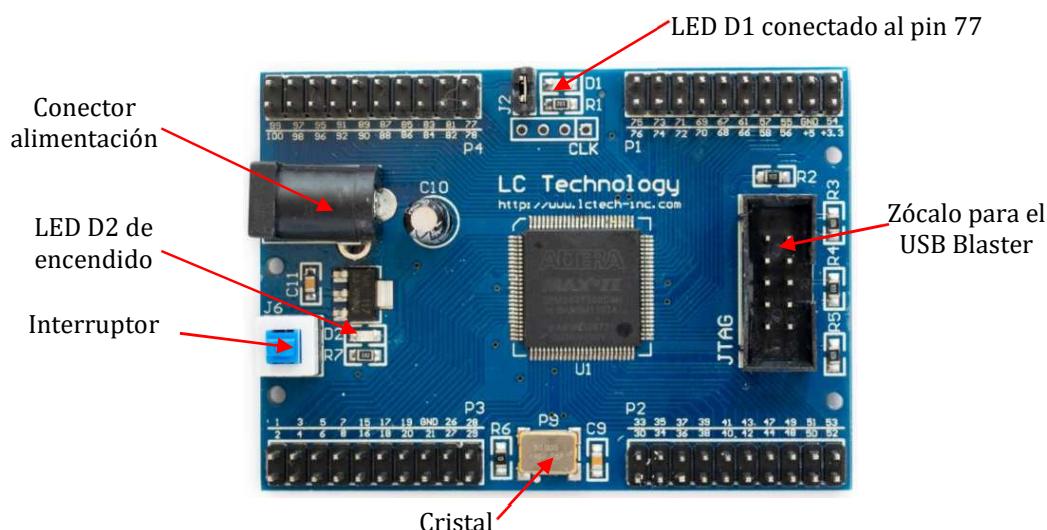
Una vez que se ha comprobado mediante simulación que el circuito funciona según las especificaciones, el siguiente paso es programar el CPLD. Este proceso consta de las siguientes fases:

1. Asignación de pines.
2. Configuración de pines no usados en tri-state.
3. Programación del CPLD.

Se debe tener en cuenta que la programación del CPLD solo se podrá llevar a cabo en caso de contar con un USB Blaster y una tarjeta EPM240.

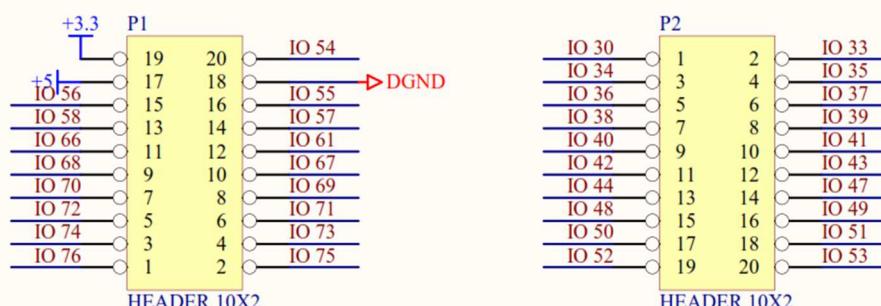
### 5.1. Asignación de pines

Para asignar los pines es imprescindible tener un mínimo de conocimiento de la tarjeta que se va a emplear. La tarjeta en cuestión es la EPM240, basada en CPLD EPM240T100C5. Esta dispone de una conexión de alimentación a 5 voltios, un oscilador que permite obtener una frecuencia de 50 MHz, un interruptor de encendido, el zócalo para la conexión del USB Blaster, un LED D2 para indicar que está alimentado y un LED D1 conectado al pin 77 (Figura 43).



**Figura 43.** Aspecto de la tarjeta EPM240.

Además de esos elementos dispone de cuatro conectores, P1 a P4, con los que se puede acceder a los pines del CPLD. En las Figura 44 y Figura 45 se muestran estos conectores. Se recomienda consultar el esquemático de la tarjeta para observar la conexión de los demás elementos que incorpora.



**Figura 44.** Conectores P1 y P2 de la tarjeta EPM240.

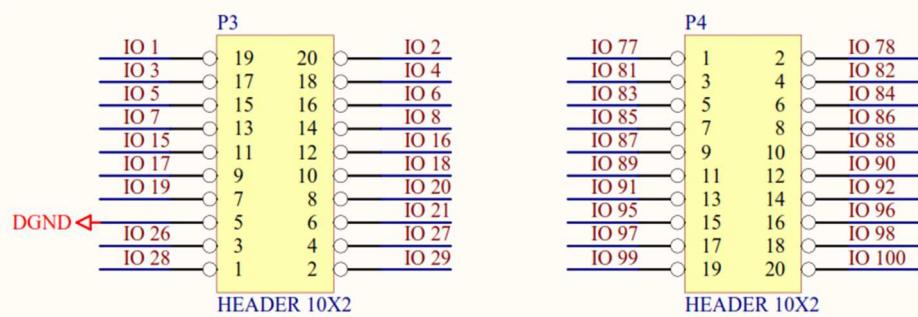


Figura 45. Conectores P3 y P4 de la tarjeta EPM240.

Una vez que se ha decidido en qué terminales se van a conectar cada una de las entradas y salidas del circuito, desde el menú de Quartus Prime Lite se accede a *Assignments -> Pin Planner* (Figura 46).

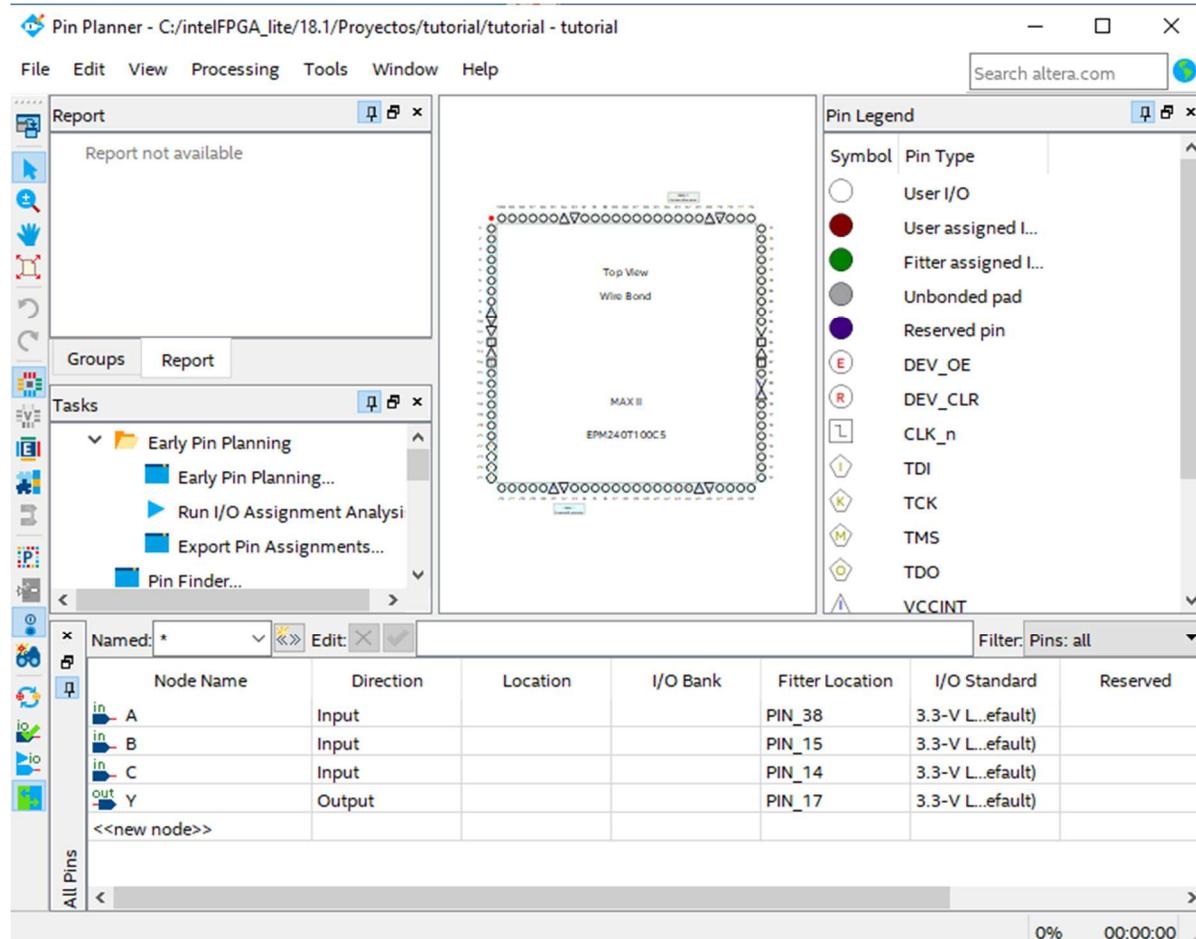


Figura 46. Ventana Pin Planner.

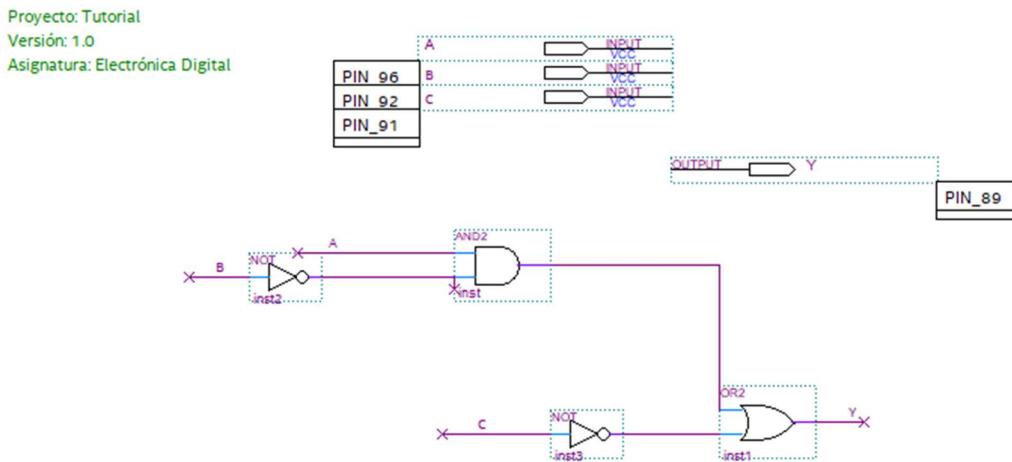
En la parte inferior de la ventana *Pin Planner* aparece el listado de las señales de nuestro esquemático que se han conectado a puertos. Para cada una de ellas en el campo *Location* se debe indicar a qué pin se van a conectar. Para este tutorial las señales A, B, C e Y se van a conectar a los pines 96, 92, 91 y 89, respectivamente. Se hace doble clic en el campo *Location* de cada una de las señales y en el menú desplegable se selecciona el pin (Figura 47).

Una vez definidos los pines se sale de la ventana del *Pin Planner*, *File -> Close*, y se vuelve a compilar el proyecto.

Named: * Filter: Pins: all						
	Node Name	Direction	Location	I/O Bank	Fitter Location	I/O Standard
	in A	Input	PIN_96	2	PIN_38	3.3-V L..fault)
	in B	Input	PIN_92	2	PIN_15	3.3-V L..fault)
	in C	Input	PIN_91	2	PIN_14	3.3-V L..fault)
	out Y	Output	PIN_89	2	PIN_17	3.3-V L..fault)
	<<new node>>					

**Figura 47.** Asignación de pines.

Después de compilar de nuevo el proyecto si se abre el fichero esquemático, *tutorial.bdf*, se observará que junto a los puertos que se han incluido aparecen los números de pines que se les ha asignado (Figura 48).

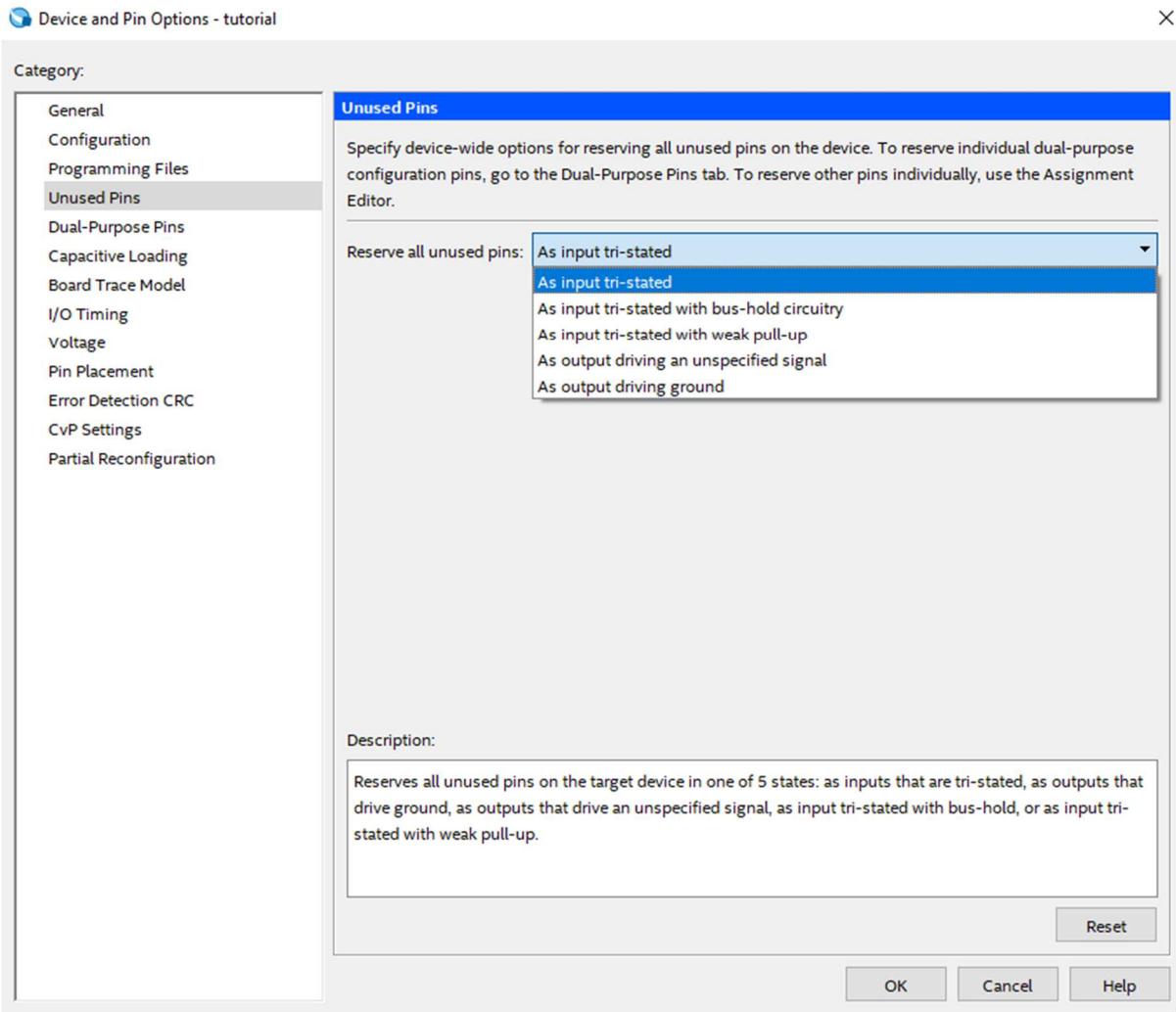


**Figura 48.** Esquemático con asignación de pines.

## 5.2. Configuración de pines no usados en triestate

Por defecto Quartus Prime Lite programa los pines no utilizados como salidas. Esto puede provocar que se destruyan en caso de una mala manipulación. Como prevención es recomendable configurar los pines no utilizados como entrada triestado.

Esta configuración se realiza accediendo a *Assignments -> Device*, se hace clic sobre el botón *Device and Pin Options...*, en la parte de la izquierda se pulsa sobre *Unused Pins*, y en el menú desplegable se selecciona *As input tri-estated* (Figura 49).

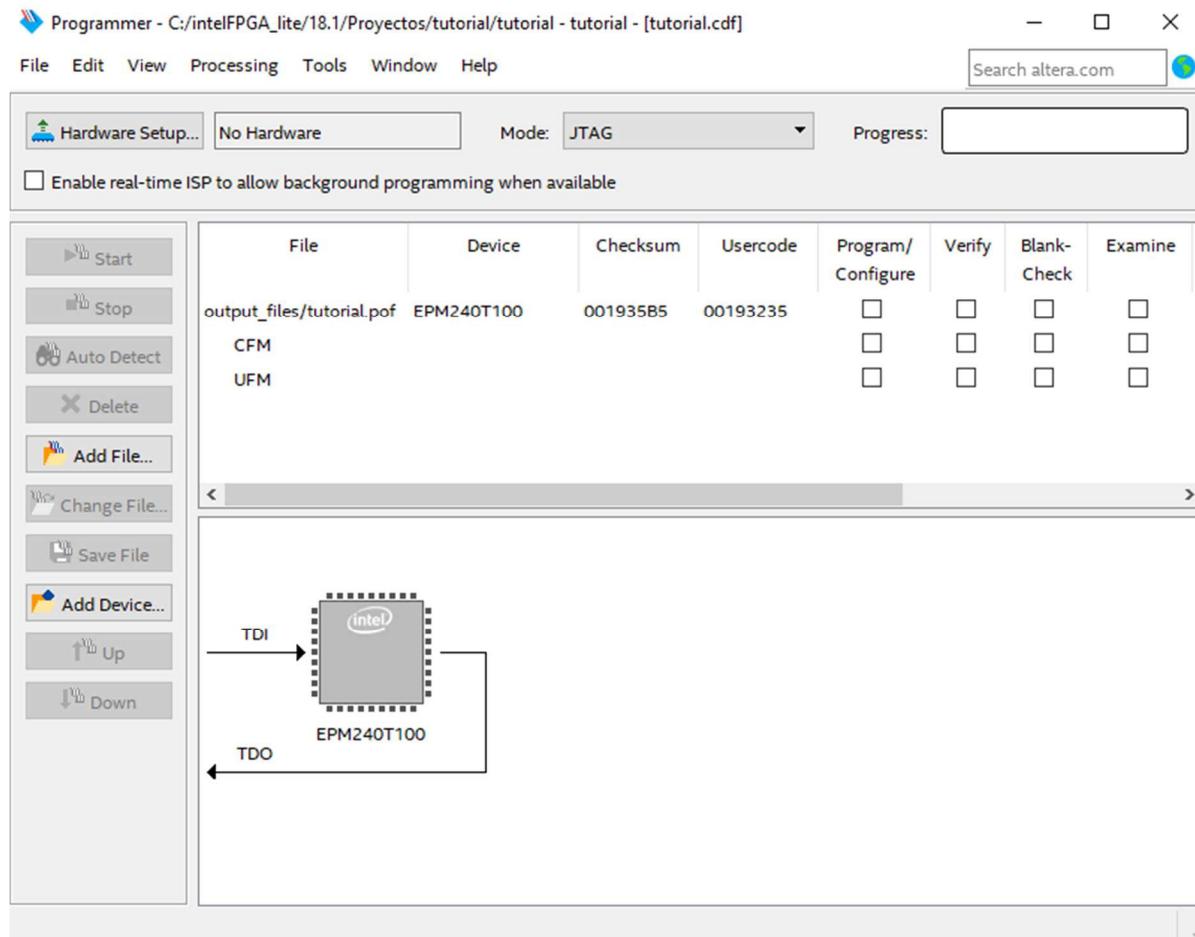


**Figura 49.** Configuración de los pines no usados.

### 5.3. Programación del CPLD

La última fase consiste en la programación de la tarjeta EPM240. Es recomendable que la tarjeta EPM240 no tenga cables de conexión con la placa board para prevenir posibles cortocircuitos. Conecte la alimentación de la tarjeta y enciéndala. Se debe iluminar el LED D2 de encendido. A continuación, conecte el USB Blaster al ordenador personal y al conector de la tarjeta EPM240.

Desde el menú de Quartus Prime Lite seleccione *Tools -> Programmer*. Se abrirá la ventana del programador mostrada en la Figura 50.



**Figura 50.** Ventana del programador.

Se selecciona el botón *Hardware Setup...* de la esquina superior izquierda y en la lista de hardware disponible se hace doble clic en *USB-Blaster*, y se pulsa *Close*. En el modo, *Mode*, es necesario asegurarse que está configurado a JTAG.

En la parte superior de la ventana del programador aparecerá por defecto el fichero *tutorial.pof*, que es el que se va a programar. En caso de que esa ventana apareciera en blanco se debe añadir manualmente. Se hace clic sobre el botón *Add File...* y en la ventana de selección de archivo que se abre se cambia a la carpeta *output\_files* y se selecciona el fichero *tutorial.pof*.

Finalmente se marcan los *checkbox* de Program/Configura y Verify según se muestra en la Figura 51.

Para proceder a programar el CPLD se pulsa el botón *Start* y en la barra de progreso de la esquina superior izquierda deberá indicar *100% Successful*. Si en la barra de progreso se indica *Fail* es necesario revisar la conexión del USB Blaster y la configuración del programador.

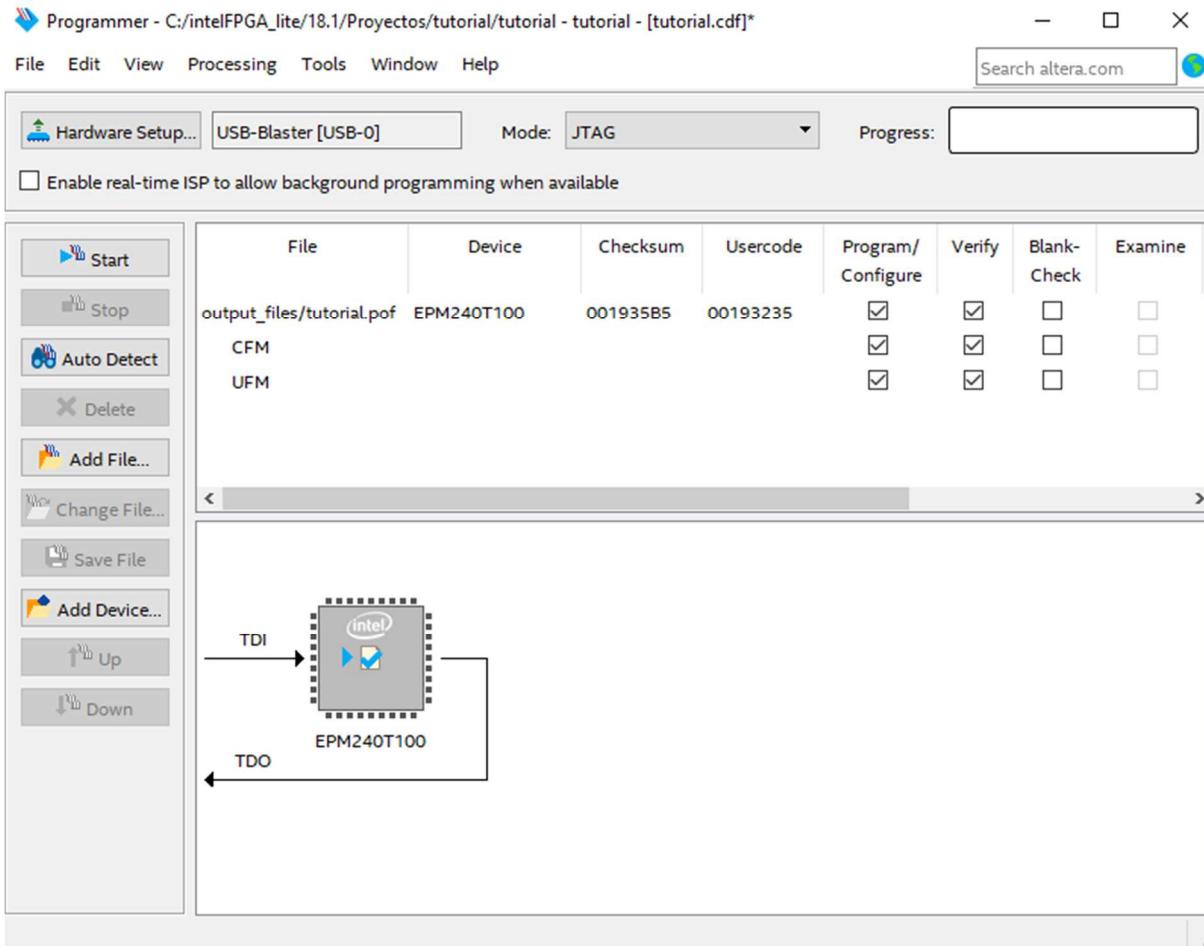


Figura 51. Compilación funcional del proyecto.

## 6. Descripción del circuito con HDL

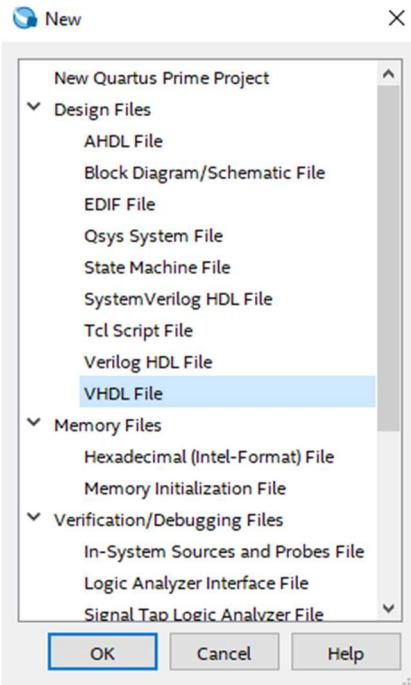
La funcionalidad de un sistema digital en Quartus Prime Lite se puede definir también con un lenguaje de descripción hardware, HDL, en lugar de con un esquemático. De los lenguajes HDL más extendidos, Verilog y VHDL, en esta asignatura se va a emplear VHDL para la programación del CPLD. Por ello se va a exponer a continuación cómo definir una funcionalidad digital con VHDL.

El primer paso, descrito en el apartado 2 y referente a la creación de un proyecto para Quartus Prime Lite, es exactamente igual, independientemente de emplear un esquemático o una descripción VHDL.

A continuación, y esto es en lo que cambia, se debe crear un fichero de texto VHD que se añadirá al proyecto, en lugar del fichero de esquemático BDF.

A partir de ahí, los demás pasos desde el descripto en el apartado 3.6, son independientes de cómo se ha descrito la funcionalidad.

Para hacer un ejemplo de proyecto descrito en VHDL se hará a partir de la misma función lógica  $Y = A \cdot \bar{B} + \bar{C}$ . Para ello se creará un nuevo proyecto, llamado *tutorialVHDL*. Una vez creado el proyecto se crea un nuevo archivo VHD, seleccionando *File -> New* y dentro del apartado *Design Files* se selecciona el tipo *VHDL File* (Figura 52). En la ventana principal de Quartus Prime Lite se abre un editor de texto en el que se podrá escribir el código VHDL que describe la funcionalidad del sistema.



**Figura 52.** Nuevo fichero VHD.

En este editor de texto se va a escribir el código mostrado a continuación, que implementa la función lógica  $Y=A \cdot \bar{B} + \bar{C}$ . Una vez escrito el código se guarda el fichero con el nombre *tutorialVHDL.vhd* (Figura 53).

```
library IEEE;  
  
use IEEE.STD_LOGIC_1164.ALL;  
  
entity tutorialVHDL is  
    Port ( A,B,C : in STD_LOGIC;  
           Y : out STD_LOGIC);  
end tutorialVHDL;  
  
architecture Behavioral of tutorialVHDL is  
begin  
    Y <= (A AND (not B)) OR (not C);  
  
end Behavioral;
```

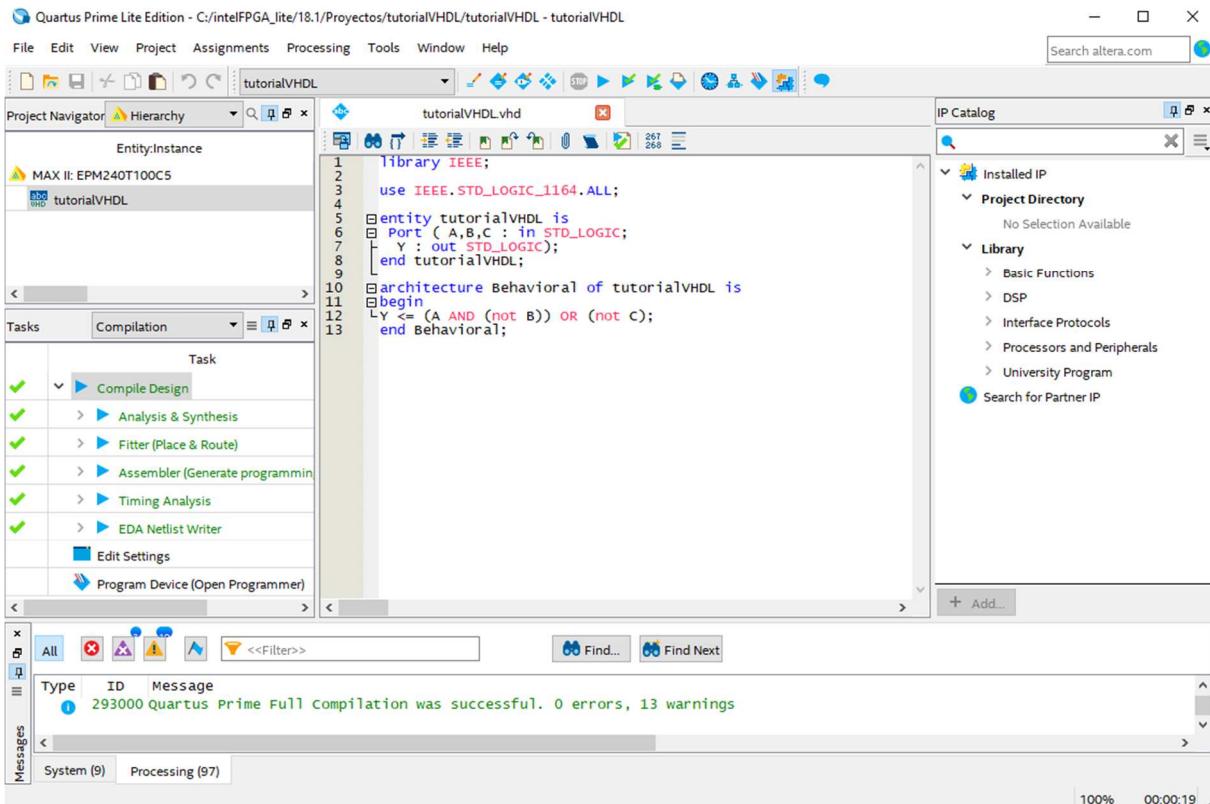


Figura 53. Ventana principal con fichero VHD.

A partir de este punto se sigue el mismo procedimiento que con la descripción con esquemático: compilación, simulación con el simulador de formas de onda, asignación de pines, compilación de nuevo y descarga a la tarjeta.

## 7. Bibliografía

Existen gran cantidad de videos y tutoriales de Quartus. La mayor parte son de una versión anterior, Quartus II, pero son extrapolables a Quartus Prime Lite a pesar de que el aspecto de la interfaz se ha modificado.

No obstante, hay un usuario, biblioman09, que ha subido varios video tutoriales a Youtube sobre la versión Quartus Prime Lite, que son los siguientes:

- Quartus Prime Lite & CPLD Altera/Intel MAX II (1)  
<https://www.youtube.com/watch?v=PtgQMq5M22A>
- Quartus Prime Lite & CPLD Alter/Intel MAX II (2)  
<https://www.youtube.com/watch?v=2LIEQ2q41ow>
- Quartus Prime Lite & CPLD MAX II. Creación de esquemáticos (1)  
<https://www.youtube.com/watch?v=uiBLOYJbm4U>
- Quartus Prime Lite & CPLD MAX II. Creación de esquemáticos (2)  
[https://www.youtube.com/watch?v=3M1vx\\_vtI\\_M](https://www.youtube.com/watch?v=3M1vx_vtI_M)



- Quartus Prime Lite & CPLD MAX II. Diseño Jerárquico (1)  
<https://www.youtube.com/watch?v=DIUzwj00H3o>
- Kit FPGA y CPLD de Altera (Intel). Bajo coste  
[https://www.youtube.com/watch?v=ee9l\\_xnIPPC](https://www.youtube.com/watch?v=ee9l_xnIPPC)

Además de esos tutoriales también puede ser necesarios algunos documentos para ampliar contenidos, algunos de los cuales son:

- Intel® Quartus® Prime Standard Edition User Guide. Getting Started  
<https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug-qps-getting-started.pdf>
- Quartus Prime Introduction. Using Schematic Designs  
[ftp://ftp.intel.com/pub/fpgaup/pub/Intel\\_Material/18.1/Tutorials/Schematic/Quartus\\_II\\_Introduction.pdf](ftp://ftp.intel.com/pub/fpgaup/pub/Intel_Material/18.1/Tutorials/Schematic/Quartus_II_Introduction.pdf)
- Quartus Prime Introduction. Using VHDL Designs  
[ftp://ftp.intel.com/pub/fpgaup/pub/Intel\\_Material/18.1/Tutorials/VHDL/Quartus\\_Std\\_Introduction.pdf](ftp://ftp.intel.com/pub/fpgaup/pub/Intel_Material/18.1/Tutorials/VHDL/Quartus_Std_Introduction.pdf)
- Introduction to Simulation of VHDL Designs  
[ftp://ftp.intel.com/pub/fpgaup/pub/Intel\\_Material/18.1/Tutorials/VHDL/Quartus\\_II\\_Simulation.pdf](ftp://ftp.intel.com/pub/fpgaup/pub/Intel_Material/18.1/Tutorials/VHDL/Quartus_II_Simulation.pdf)
- Quartus Prime Standard Edition Handbook Volume 1: Design and Synthesis  
<https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/qts/archives/qts-qps-5v1-17-0.pdf>