

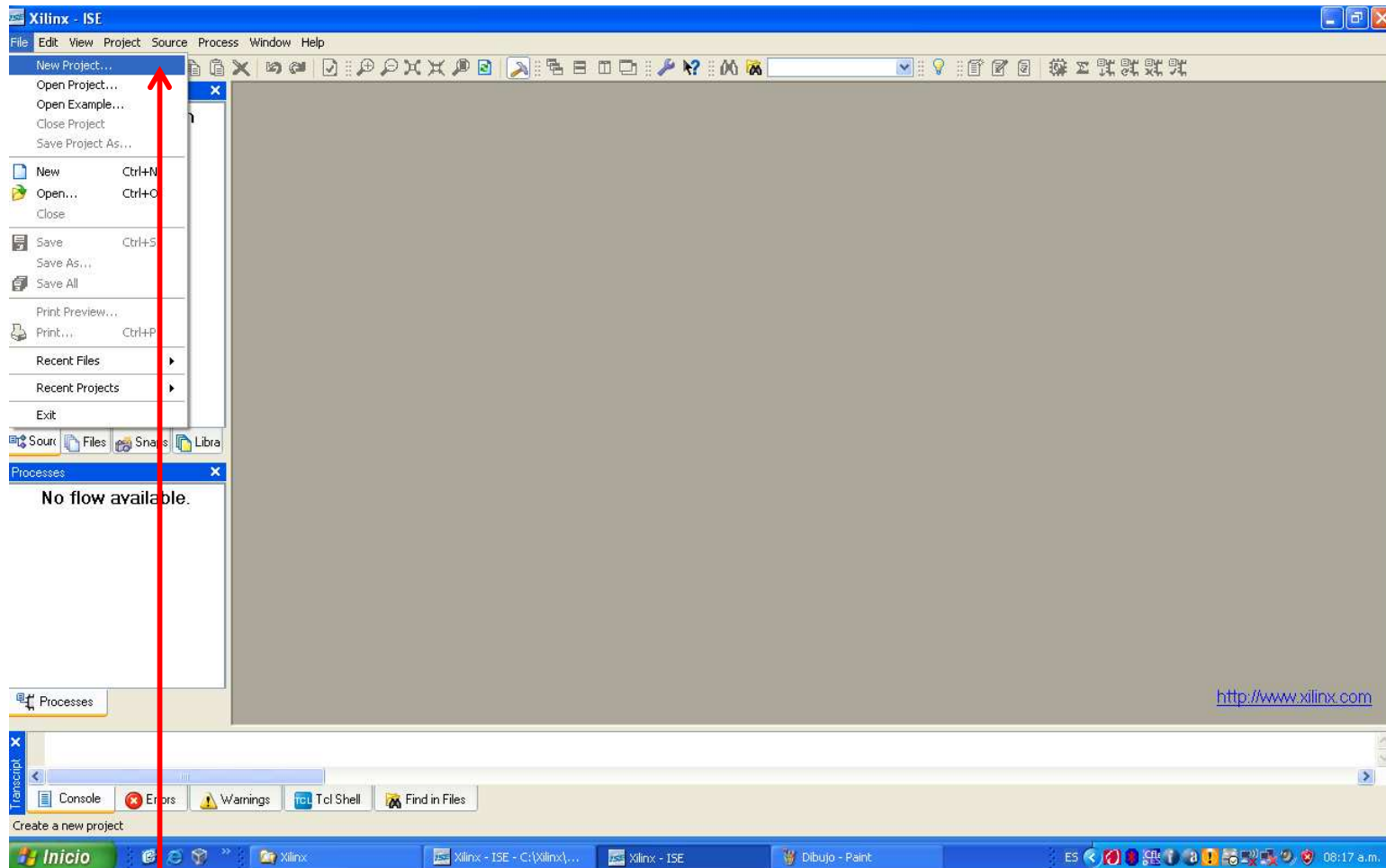
Tutorial para creación de un proyecto en ISE de Xilinx



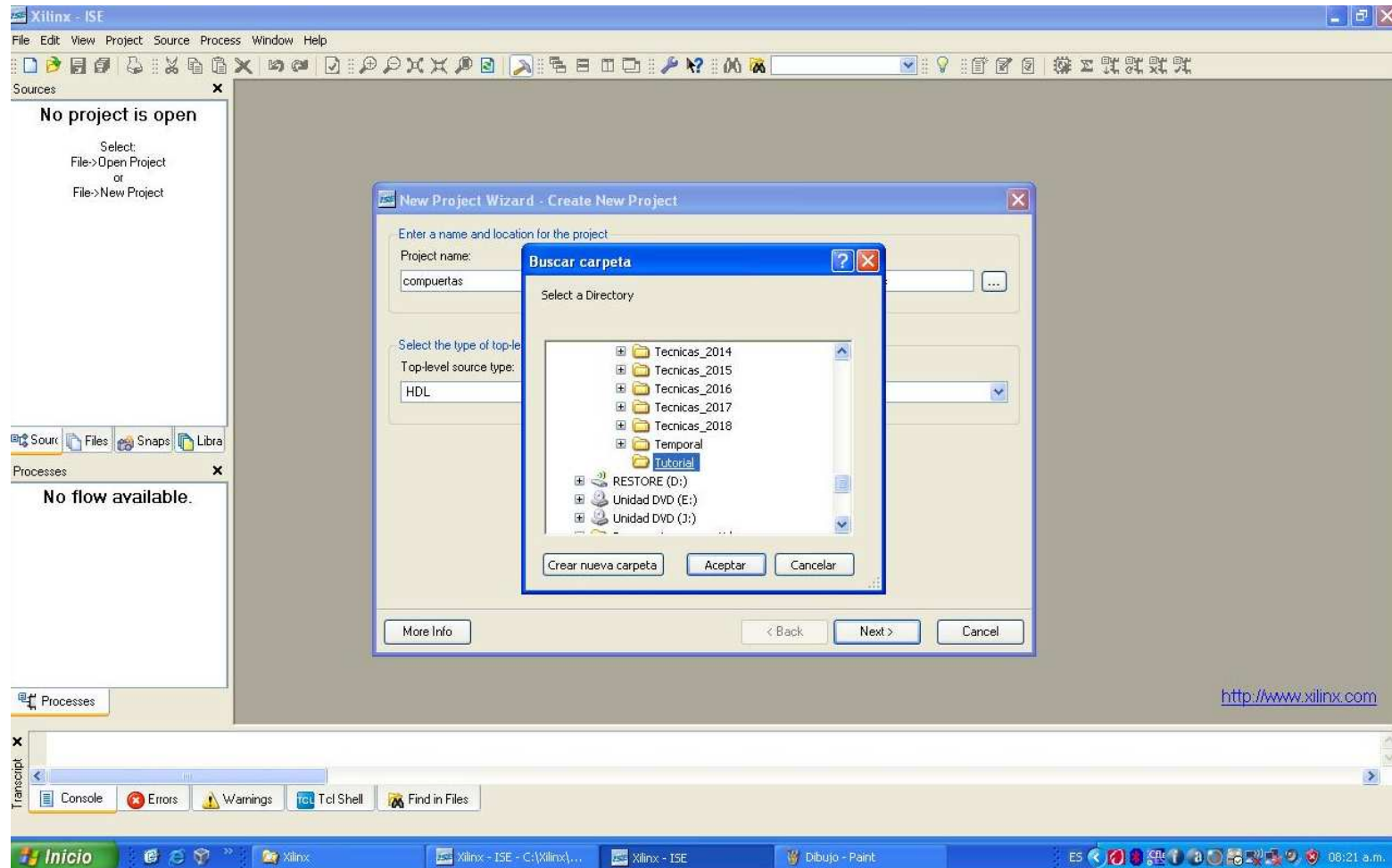
Este es un tutorial para introducirse en el manejo del entorno ISE de Xilinx, en el mismo se creará un simple proyecto que consiste en la descripción de una función lógica, compilación, simulación, asignación de pines y carga en el KIT CPLD.

Ante alguna duda escribir a la dirección de correo electrónico:

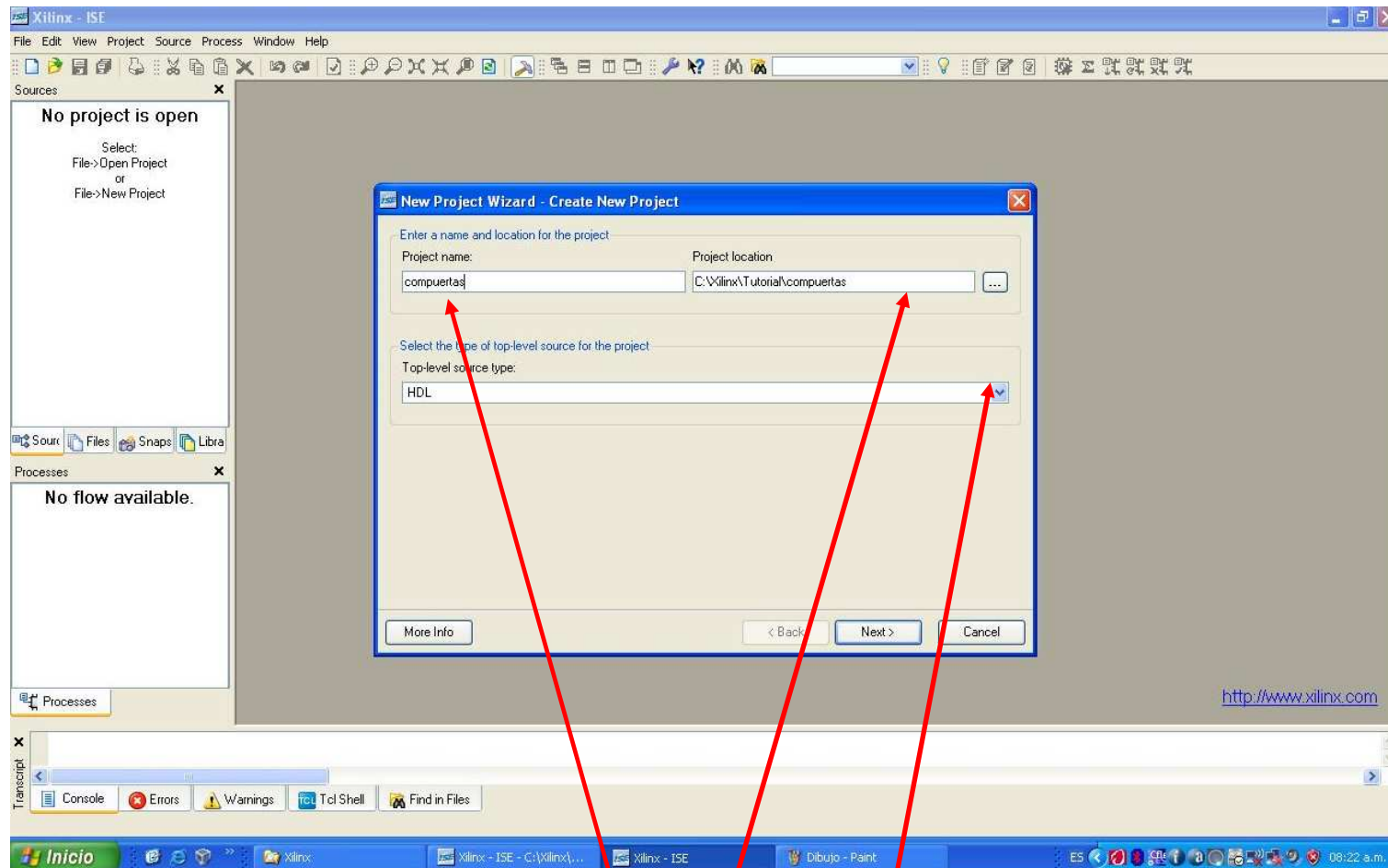
marcelocasasnovas@gmail.com



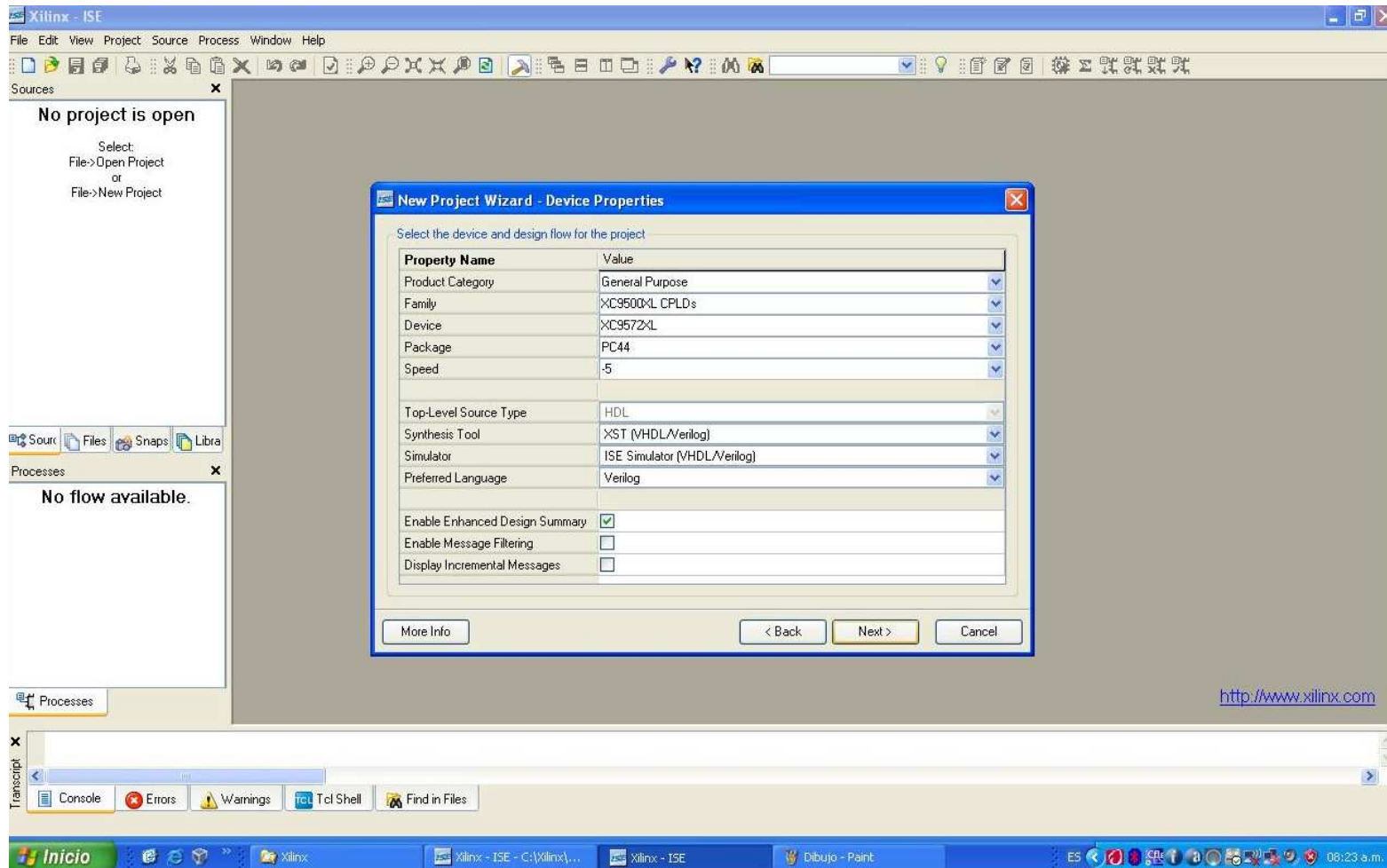
Hacer doble clic en el icono de Xilinx ISE 10.1, una vez abierto ir a FILE -> NEW PROYECT



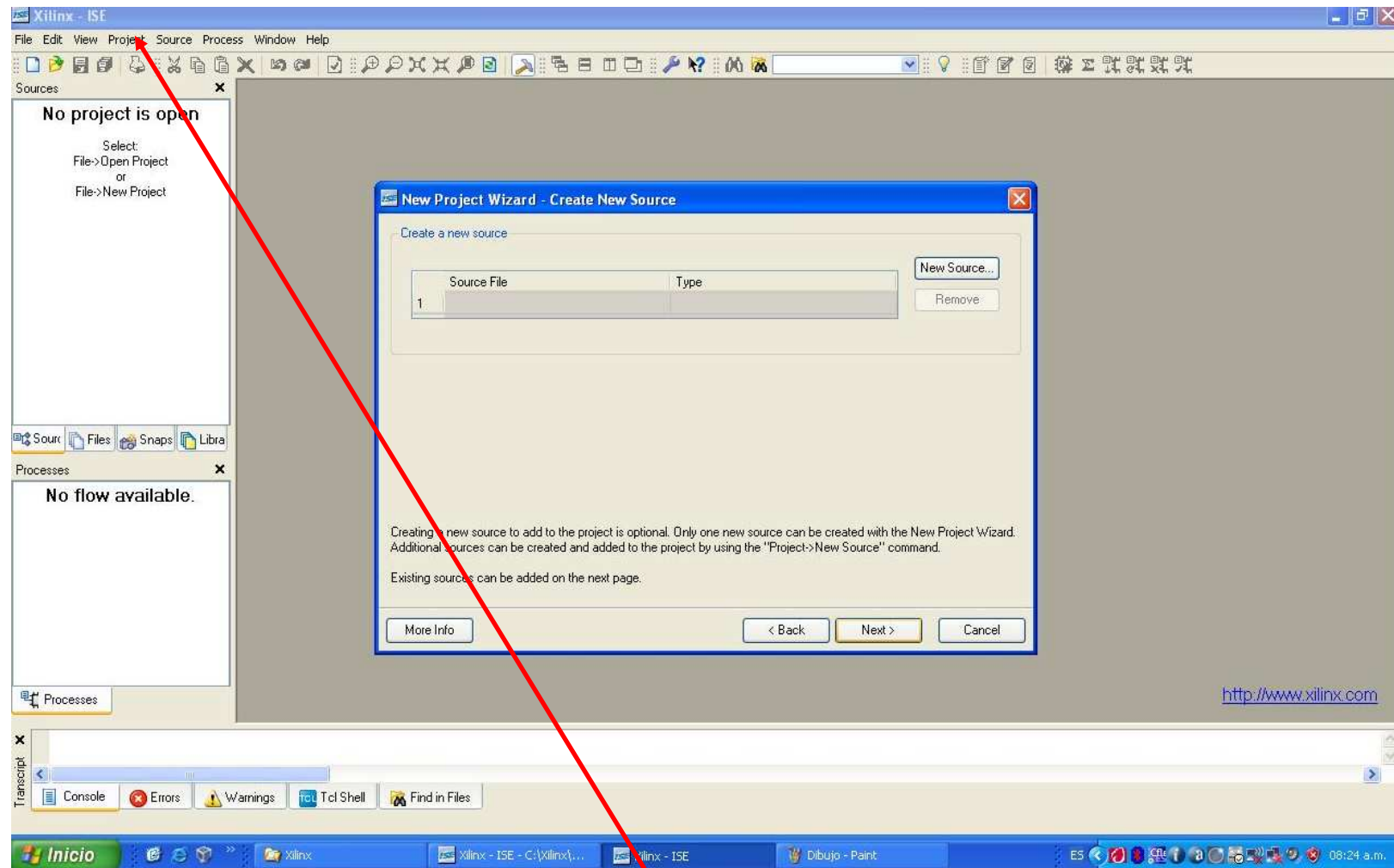
Seleccionar el directorio, que **NO DEBE TENER ESPACIOS** y darle nombre al proyecto, en este caso: **compuerta**



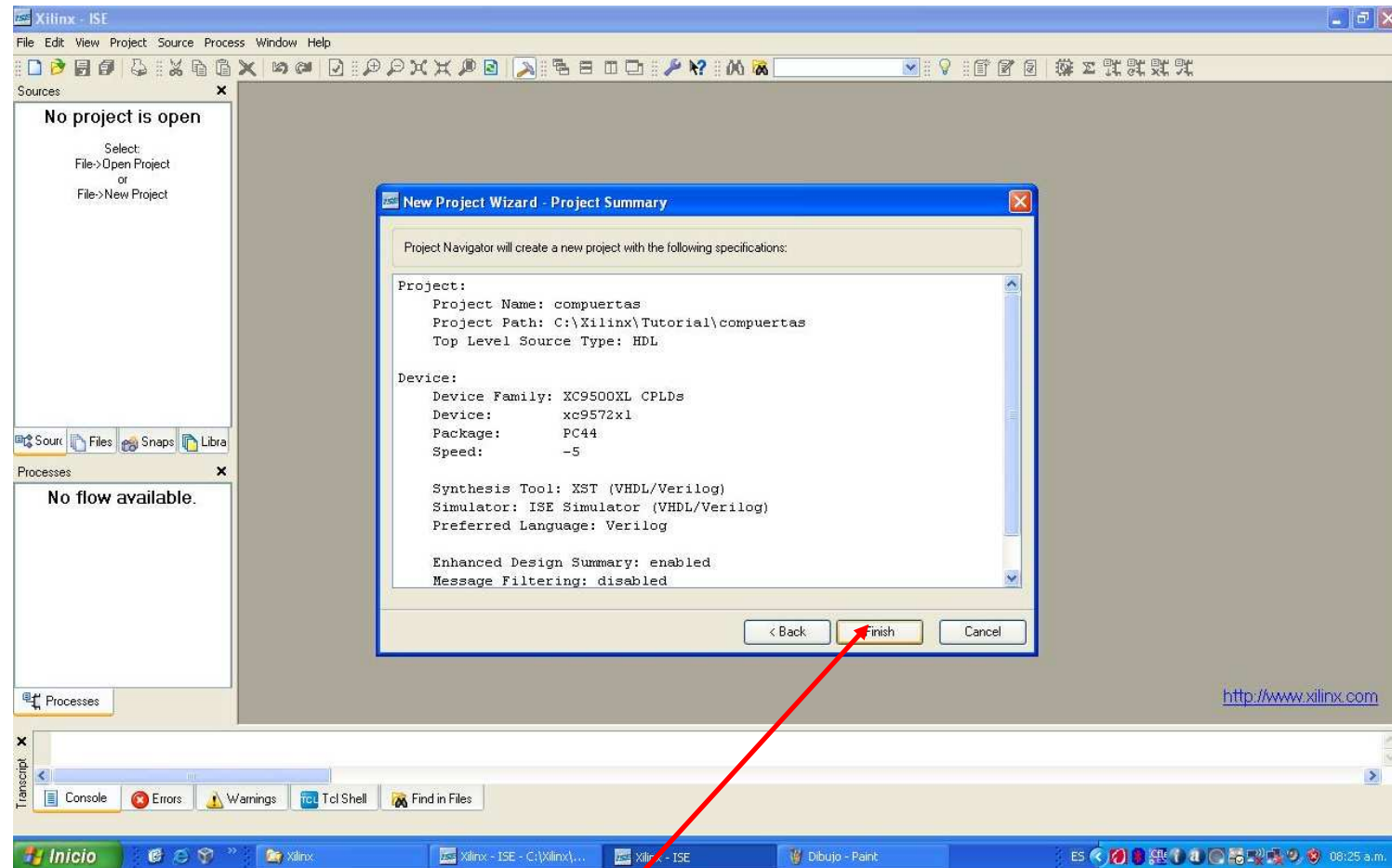
Dar el nombre al proyecto, elegir HDL asegurarse de que no haya espacios en directorios ni nombres, y hacer click en **Next**



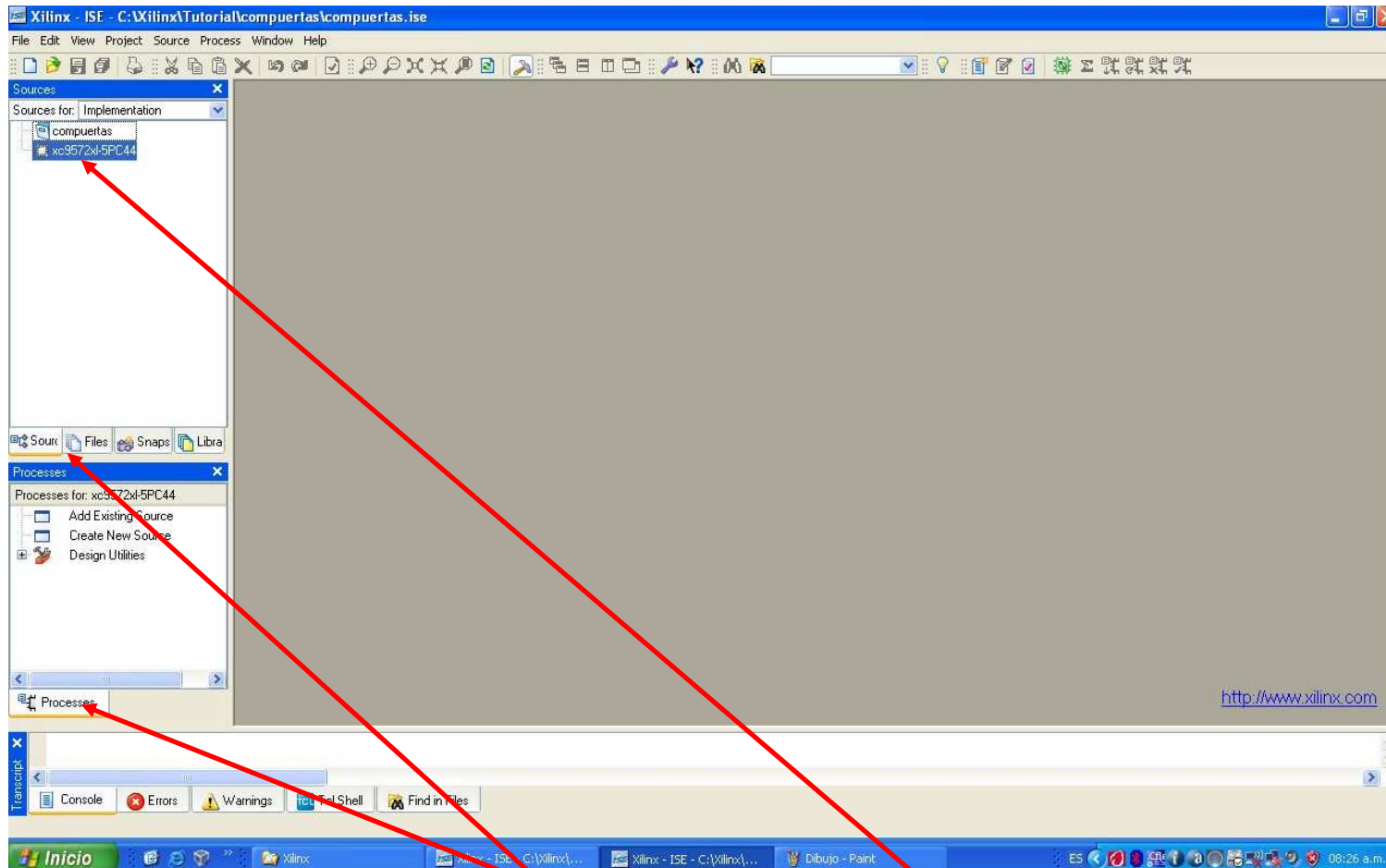
En esta etapa se selecciona el modelo de CPLD a utilizar. Y configuramos como se ve en la figura, **General Purpuse** , familia **XC9500XL CPLD**, device **XC9572XL**, simulador **ISE** y lenguaje **Verilog** y hacer click en **Next**



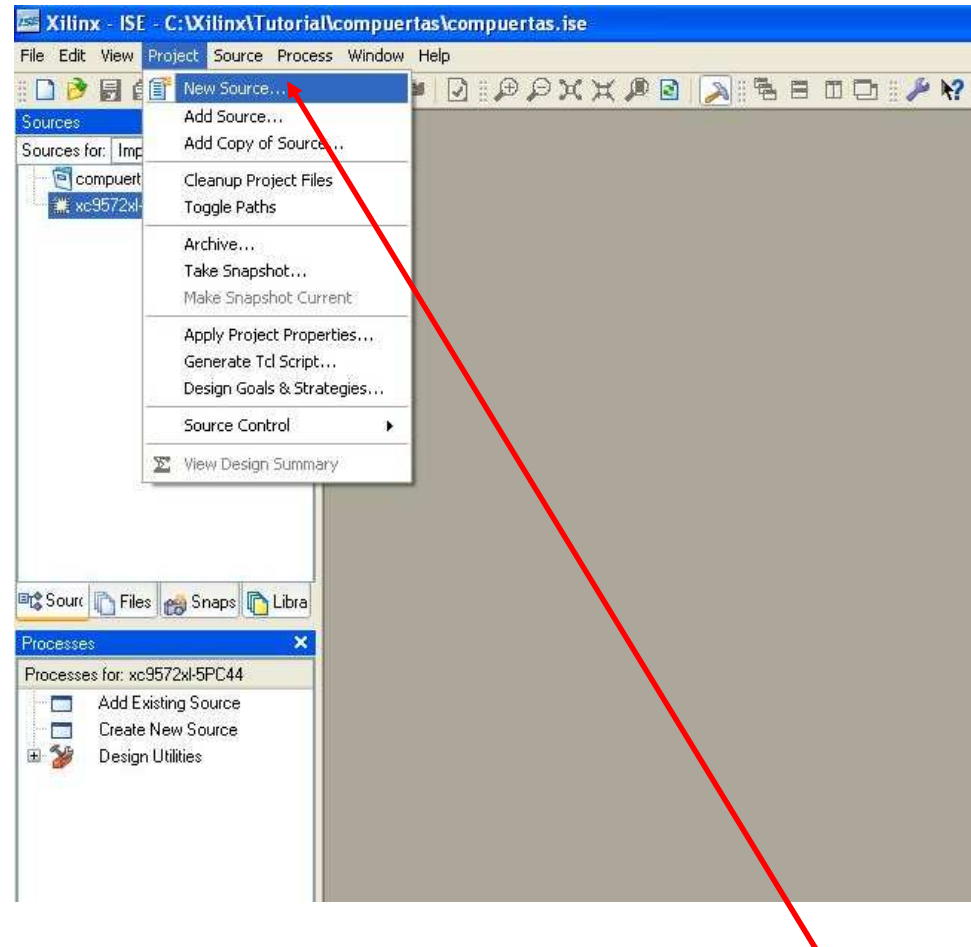
En el menú principal elegir **Project** y **New Source** y a continuación seleccionamos **Next**



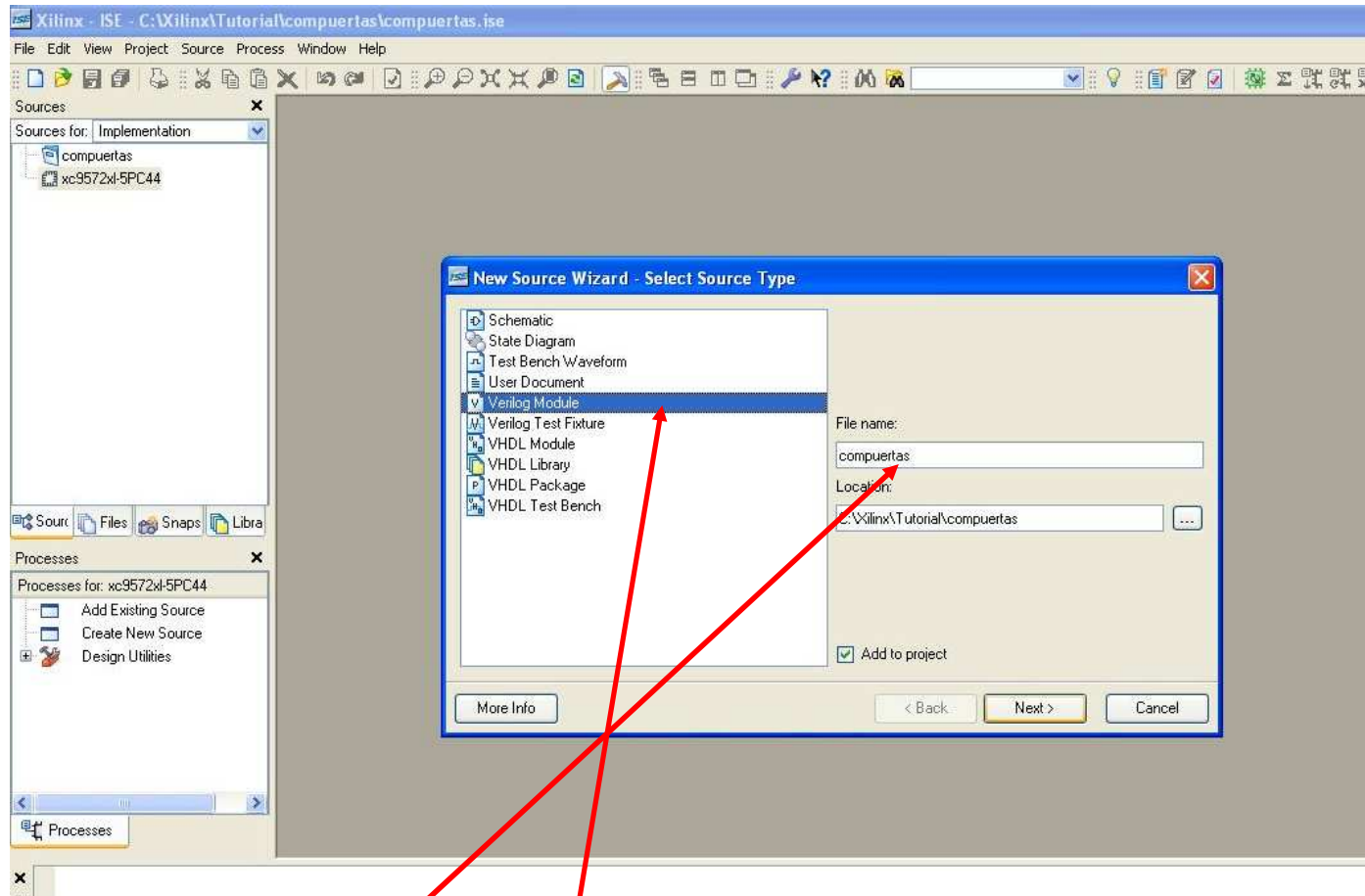
En este momento se observa un reporte de lo seleccionado hasta el momento, a continuación hacer click en **Finish**.



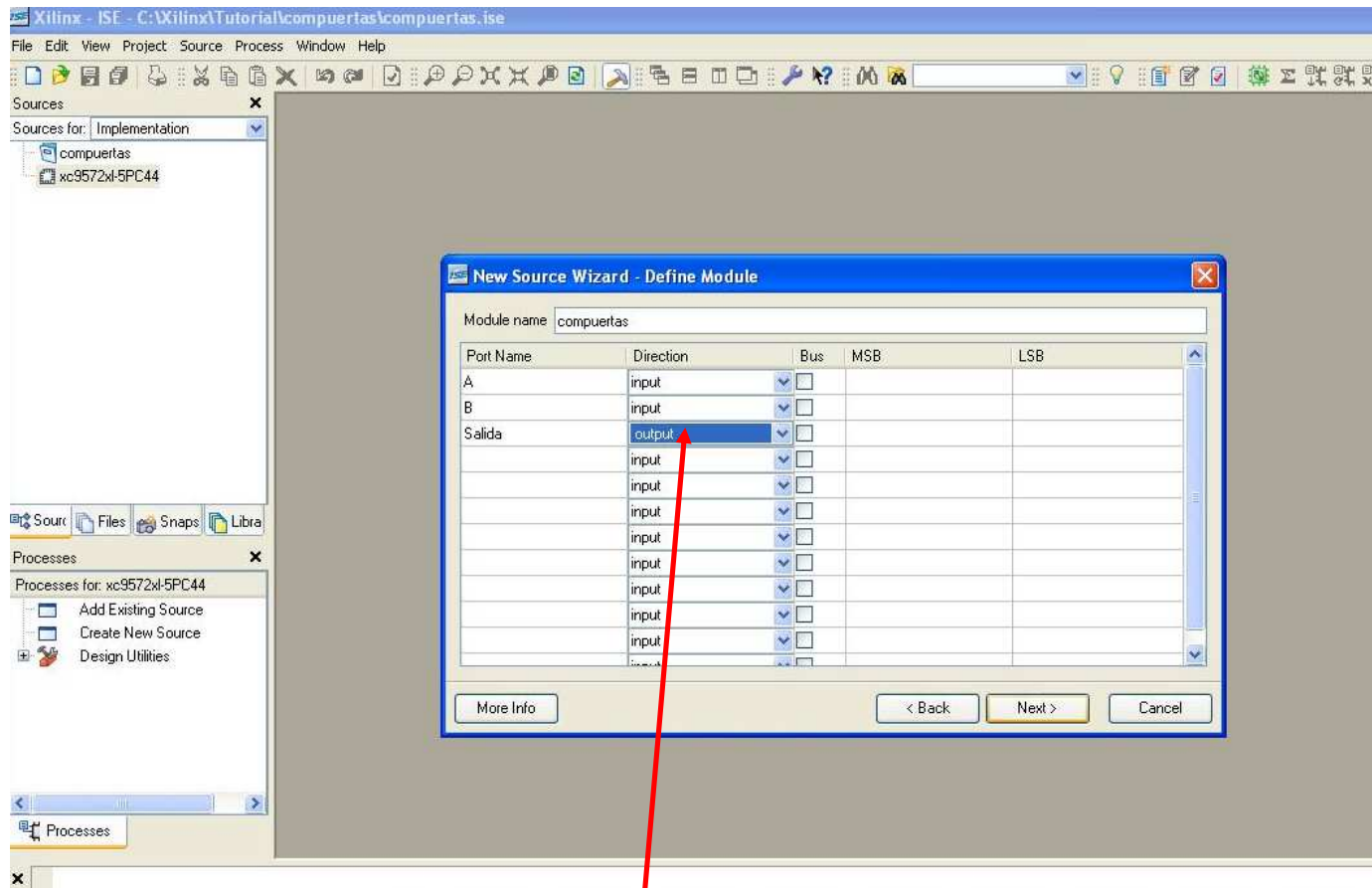
Hasta acá tenemos un proyecto vacío al cual hay que agregarle el archivo fuente donde se va a realizar la descripción de hardware. Siempre debemos tener seleccionadas las solapas **Source** y **Process**. Hacer click en el dispositivo seleccionado



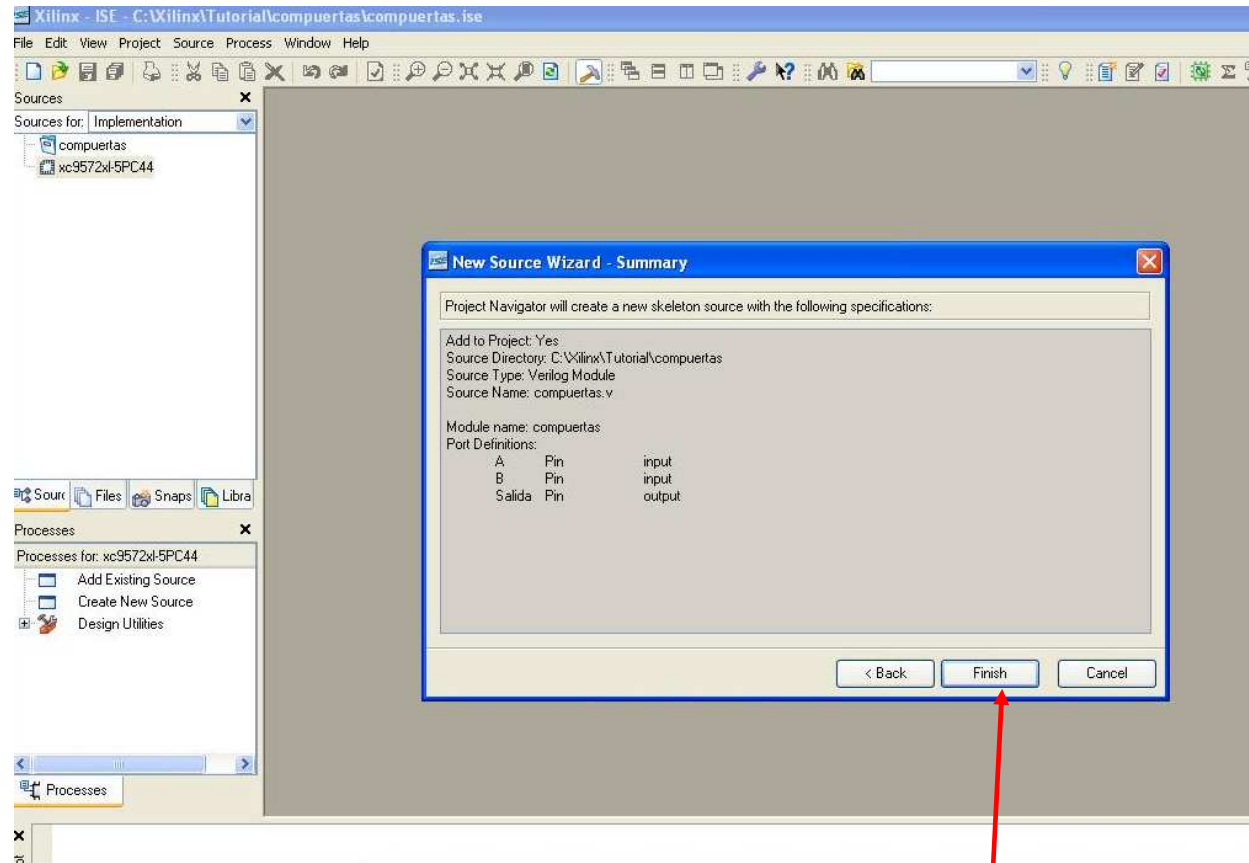
Seguidamente en el menú principal elegimos **Project** y **New Source** para crear nuestro archivo fuente donde realizaremos la descripción en Verilog y que tendrá extensión **.V**



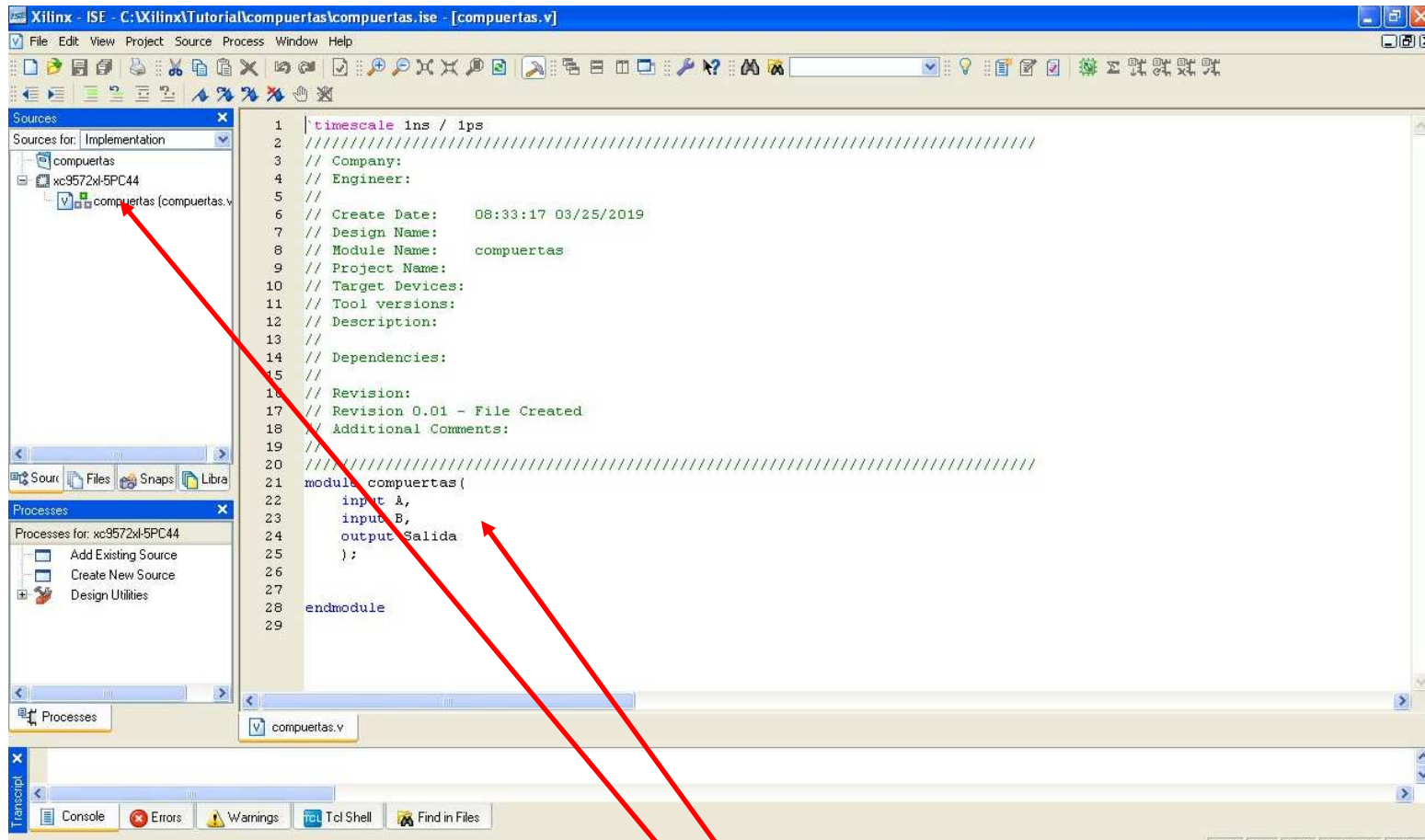
A continuación seleccionamos **Verilog Module** , que será nuestro archivo fuente y le colocamos el nombre que elijamos y damos **Next**.



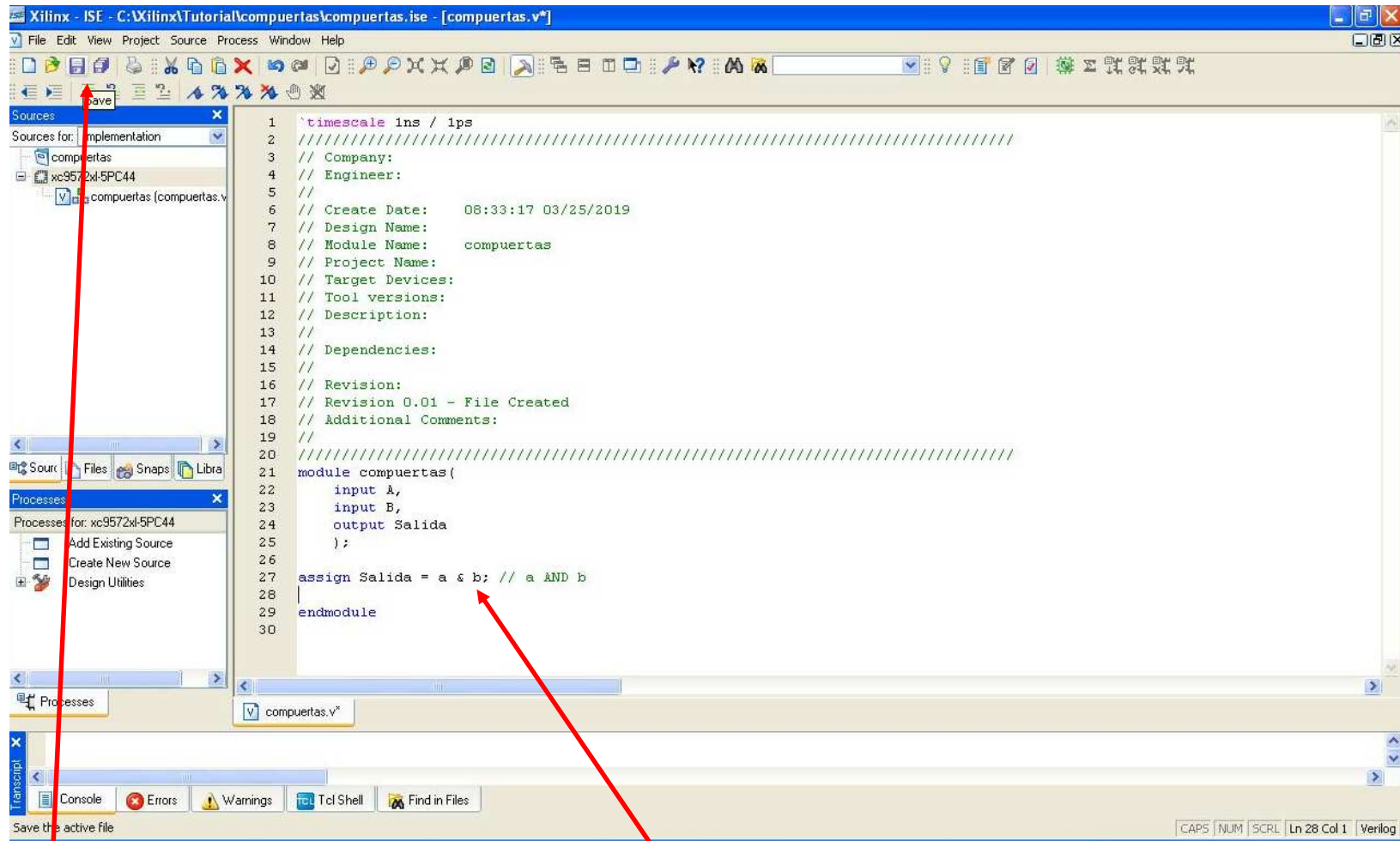
En esta secuencia debemos definir las entradas y salidas que tendrá nuestro circuito, ya que describiremos una compuerta **AND** tenemos dos entradas **A** y **B** y una salida que le llamaremos **Salida** y hacemos click en **Next**.



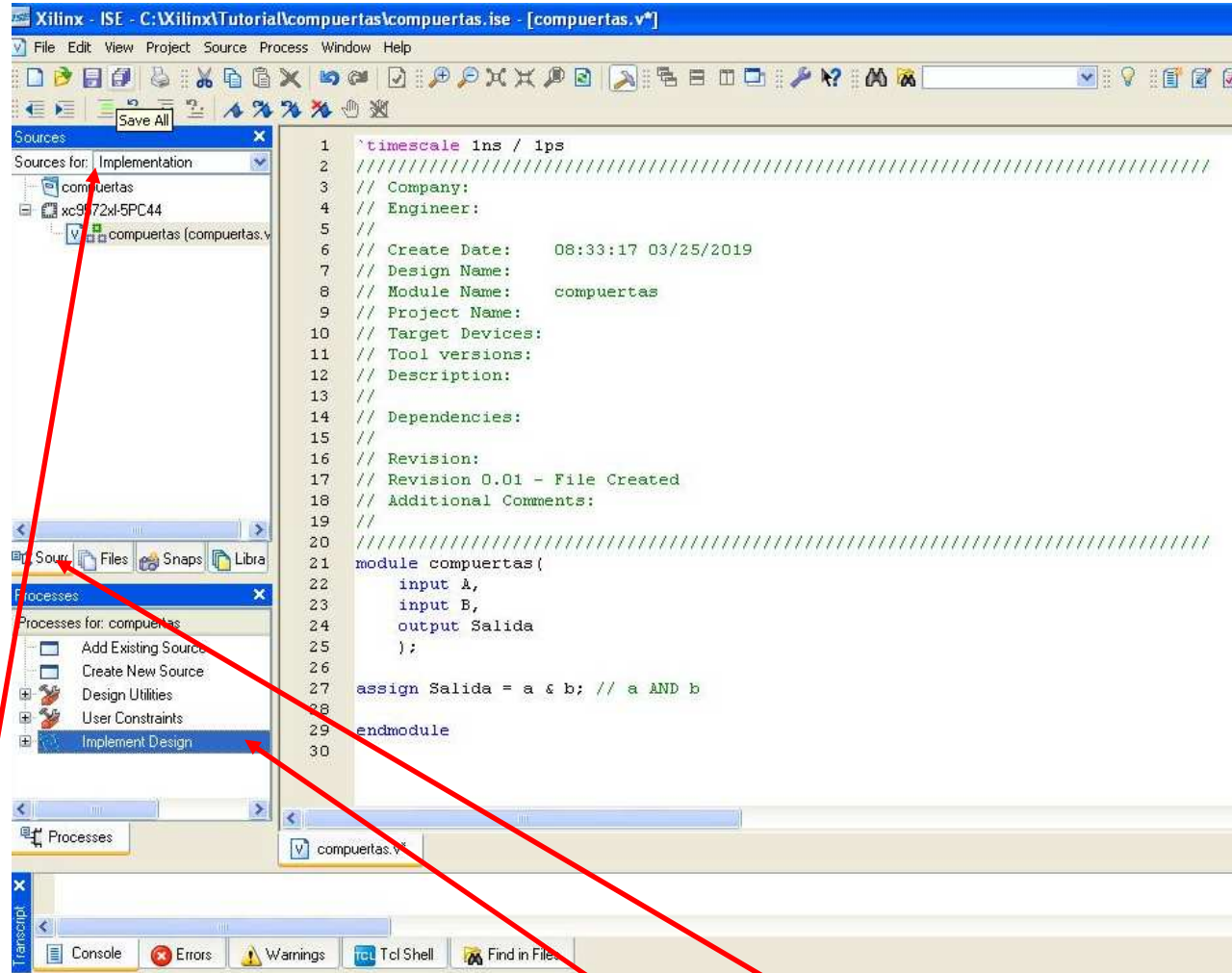
Finalmente el entorno nos muestra un resumen de lo que hemos seleccionado. Las entradas, salidas, el directorio donde se encuentra nuestro proyecto, el tipo de lenguaje etc. A continuación hacemos click en **Finish**.



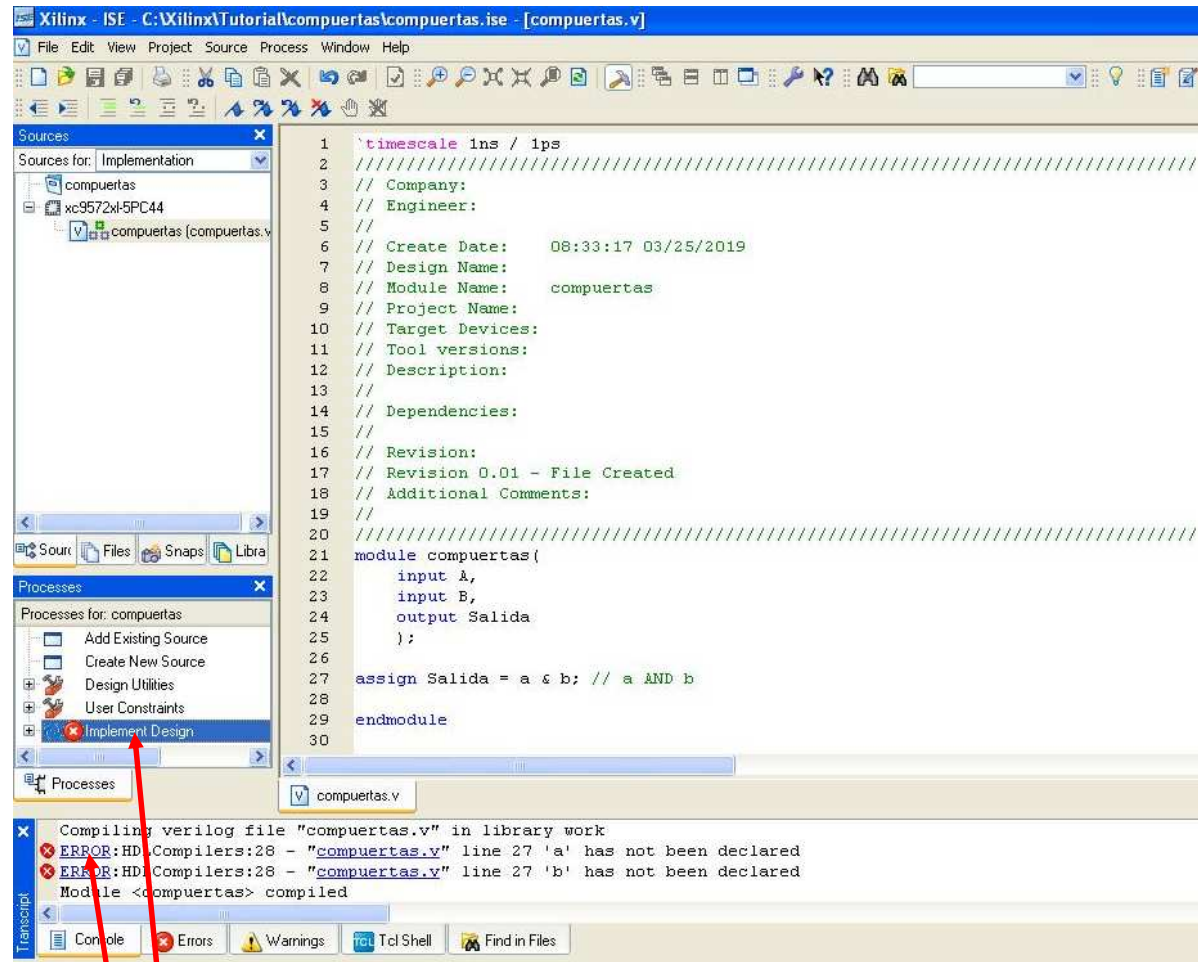
Como podemos ver en la ventana **Source** se creó la arborescencia con el archivo .V dentro de nuestro proyecto y en el lado derecho de la imagen el esqueleto del programa con las entradas y salidas definidas y a continuación nos encontramos en condiciones de comenzar a describir, nuestro código va después de la definición del módulo y antes de la palabra **endmodule**.



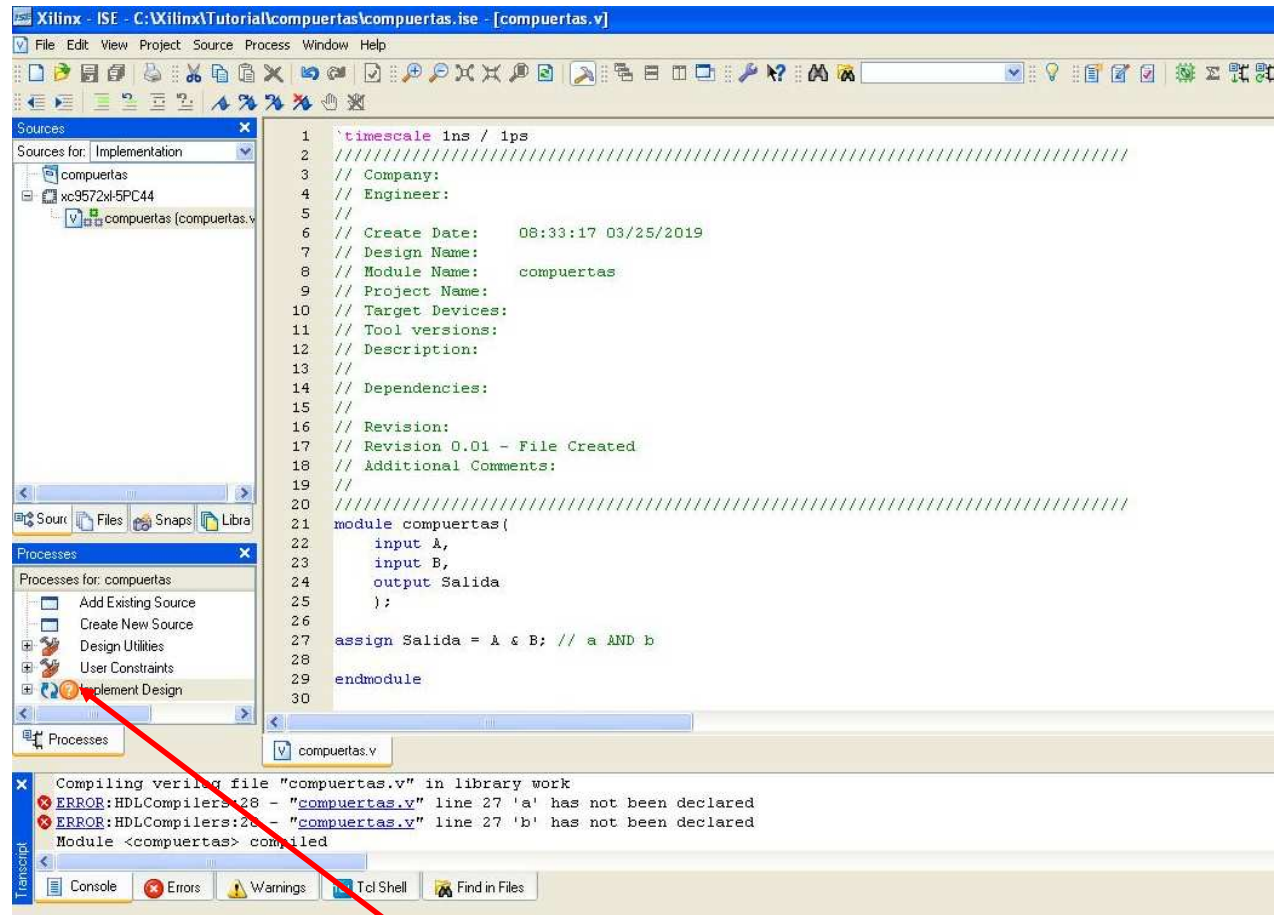
A continuación vamos a describir nuestra compuerta AND para lo que tipeamos el siguiente código : **assign Salida = a & b ;** , se deben respetar mayúsculas y minúsculas y los comentarios van entre **//** , paso siguiente salvamos el proyecto.



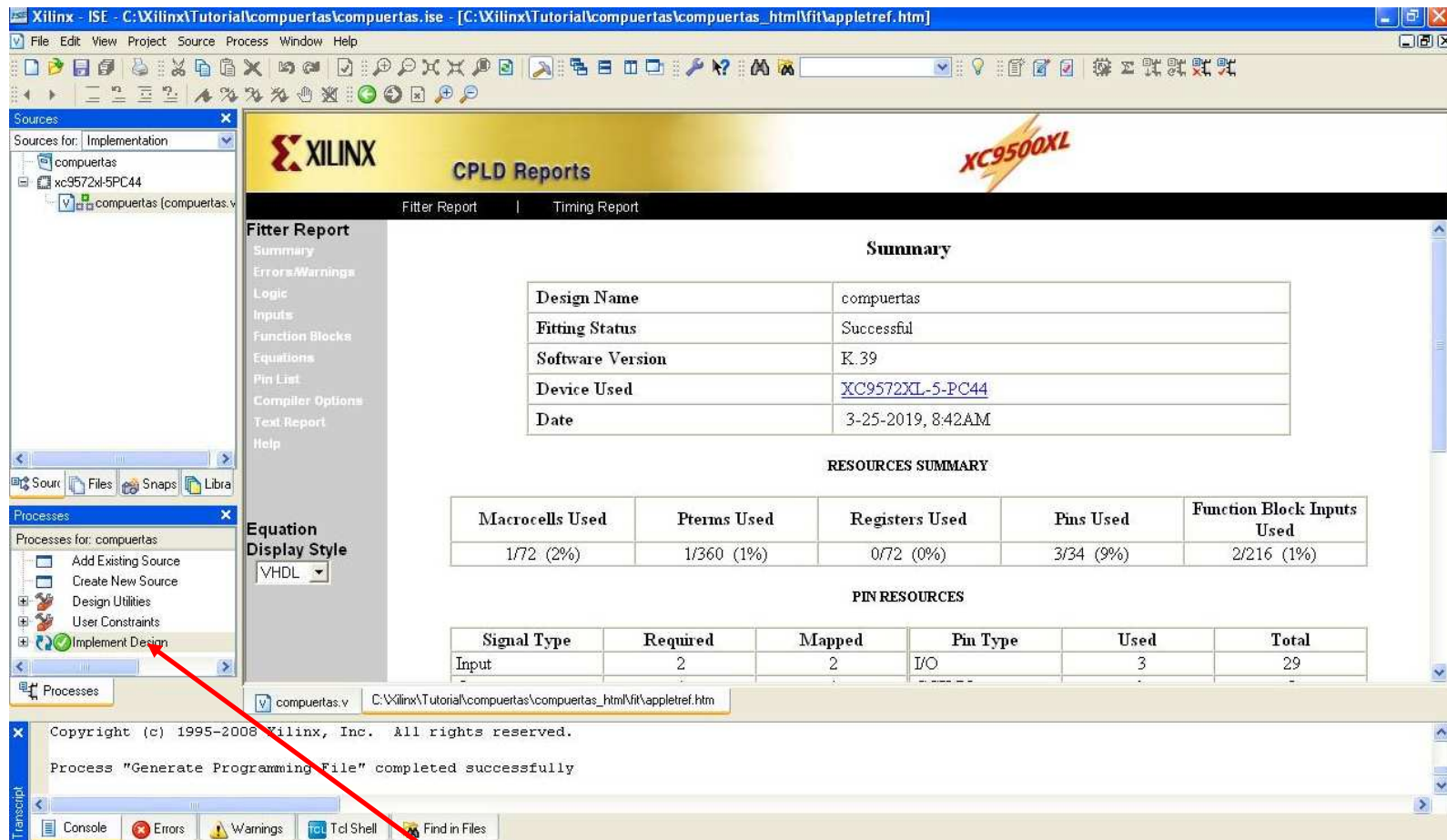
Paso siguiente es compilar nuestra descripción, para lo que debemos tener nuestro archivo seleccionado en la solapa **Sources** y la selección **Implementation**, y hacer doble click en **Implement Design** en la solapa **Process**



Si hemos seguido paso a paso el instructivo, la compilación nos debe haber arrojado el error que se muestra, debido a que la operación AND fue ejecutada con las entradas en minúsculas y nosotros las definimos en mayúsculas, por lo tanto modificamos y volvemos a compilar.



Al modificar y salvar el proyecto la opción de **Implement Design** aparece como una burbuja en color naranja, para lo que se debe hacer doble click nuevamente para iniciar la compilación.



Una vez compilado, si no hay error alguno se ve una imagen como la de la figura, donde se ve un reporte de los recursos de hardware que insumió nuestra descripción. La burbuja aparece en verde si la compilación fue exitosa.

The screenshot shows the Xilinx ISE 10.1 interface. The 'Project' menu is open, with 'New Source...' highlighted. The 'CPLD Reports' window is displayed, showing a 'Summary' table with the following data:

| Summary | |
|------------------|-------------------|
| Design Name | compuertas |
| Fitting Status | Successful |
| Software Version | K.39 |
| Device Used | XC9572XL-5-PC44 |
| Date | 3-25-2019, 8:42AM |

Below the summary is the 'RESOURCES SUMMARY' table:

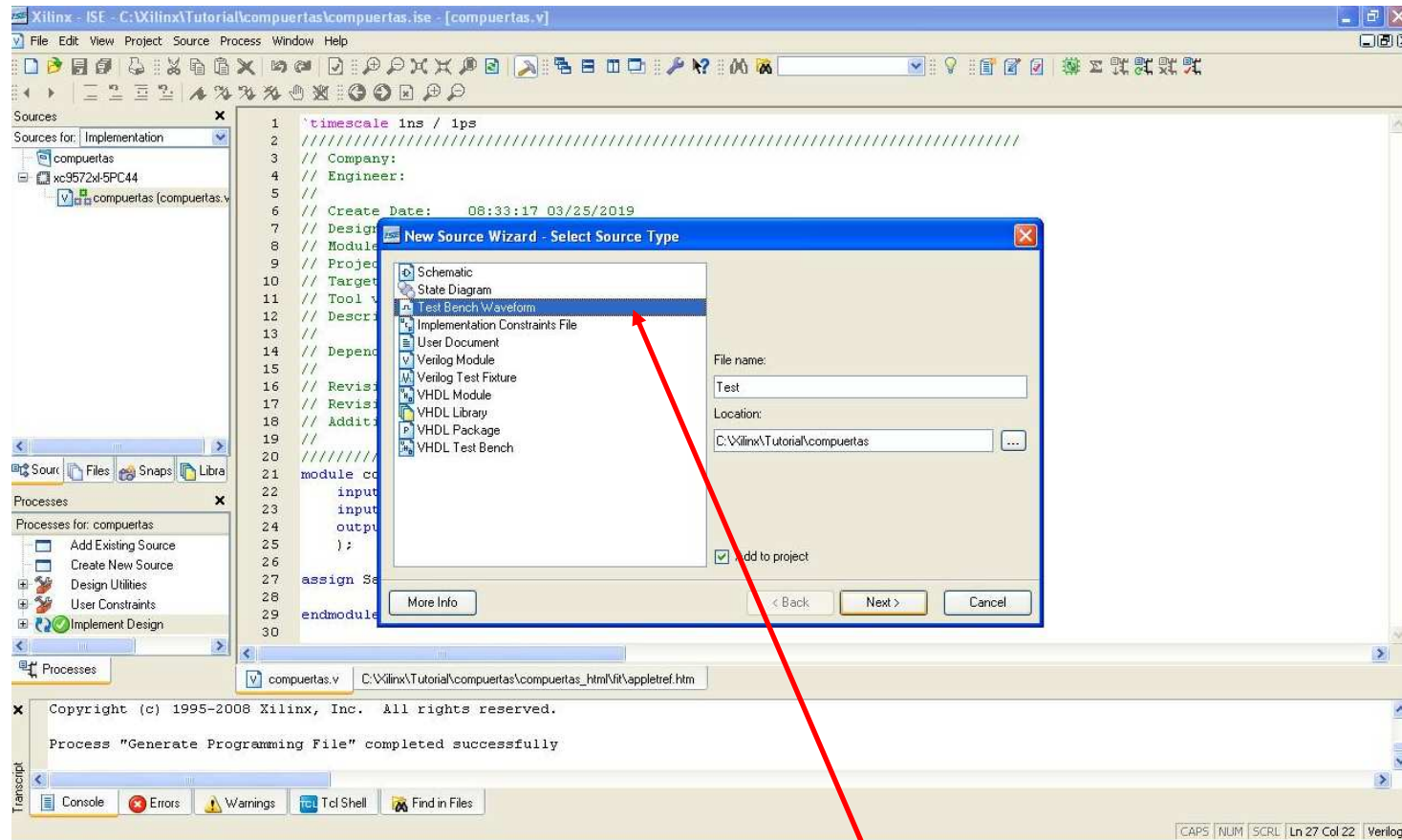
| Macrocells Used | Prtnms Used | Registers Used | Pins Used | Function Block Inputs Used |
|-----------------|-------------|----------------|-----------|----------------------------|
| 1/72 (2%) | 1/360 (1%) | 0/72 (0%) | 3/34 (9%) | 2/216 (1%) |

At the bottom is the 'PIN RESOURCES' table:

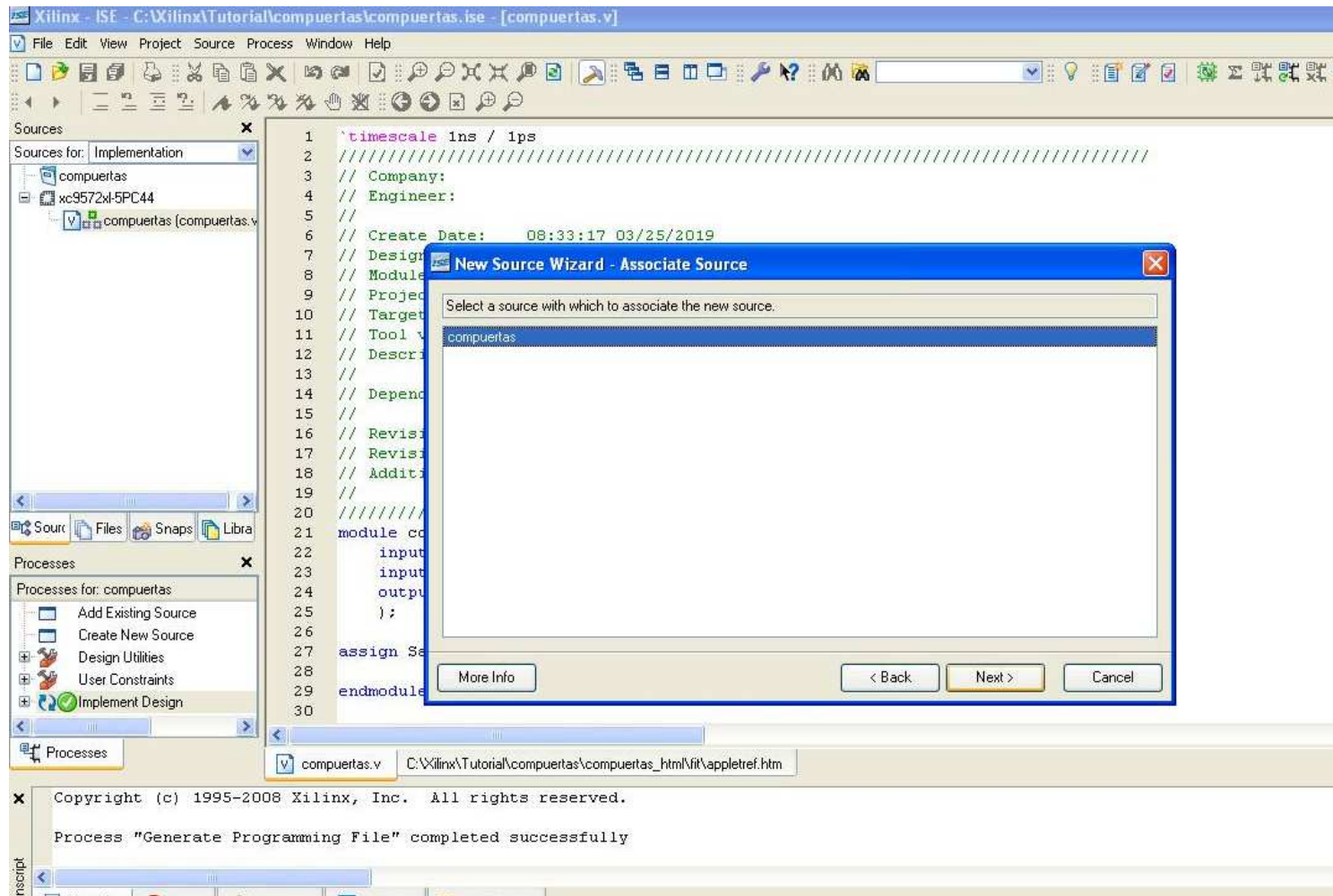
| Signal Type | Required | Mapped | Pin Type | Used | Total |
|-------------|----------|--------|----------|------|-------|
| Input | 2 | 2 | I/O | 3 | 29 |

The status bar at the bottom indicates 'Process "Generate Programming File" completed successfully'.

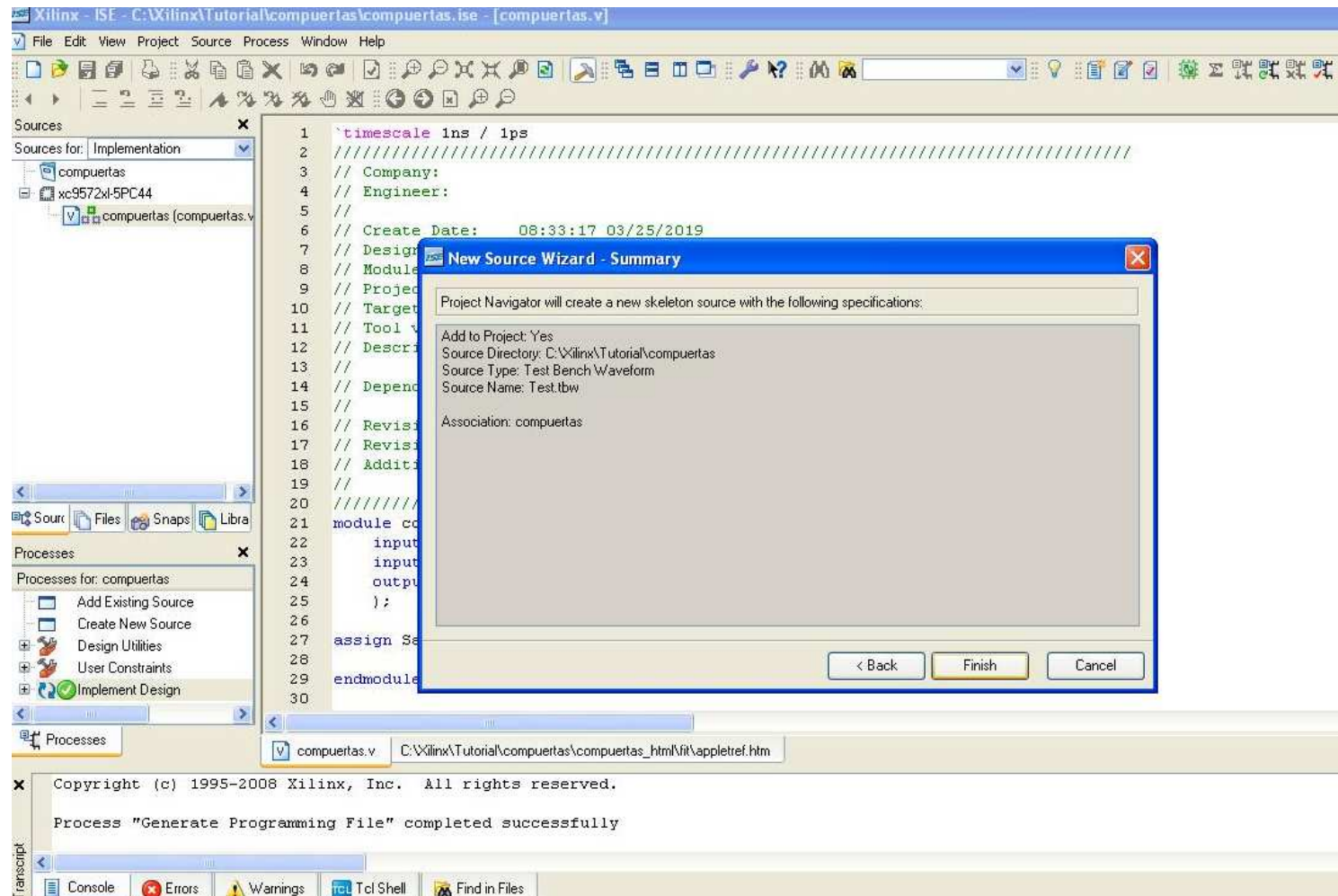
Una vez con la compilación exitosa el paso siguiente es crear un archivo para realizar la simulación de nuestra descripción, para lo cual vamos nuevamente al menú principal **Project -> New Source**



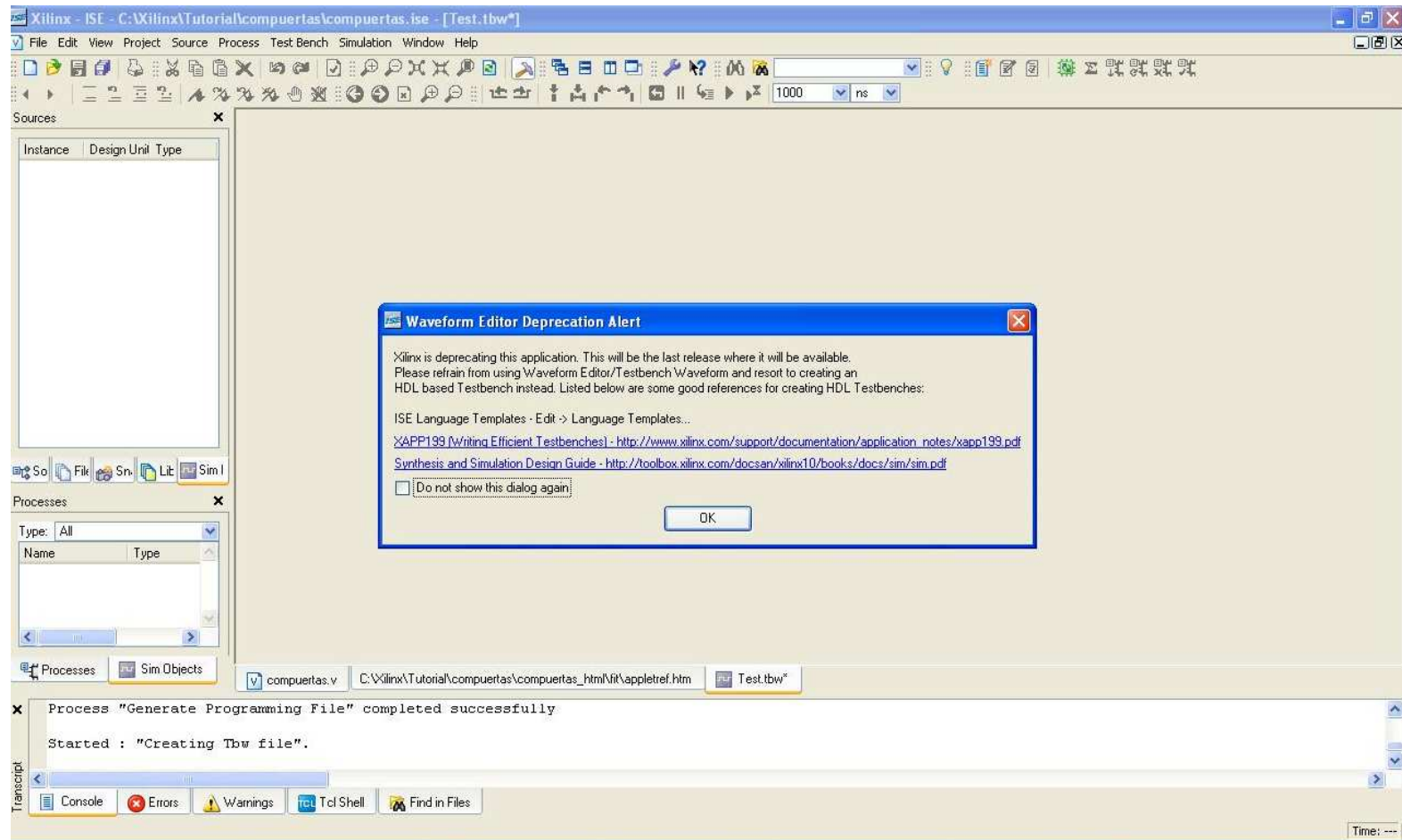
Seguidamente seleccionamos la opción **Testbench Waveform** que será el archivo de simulación y le asignamos un nombre, la extensión que le dará el sistema es **.TBW** , luego hacemos click en **Next**.



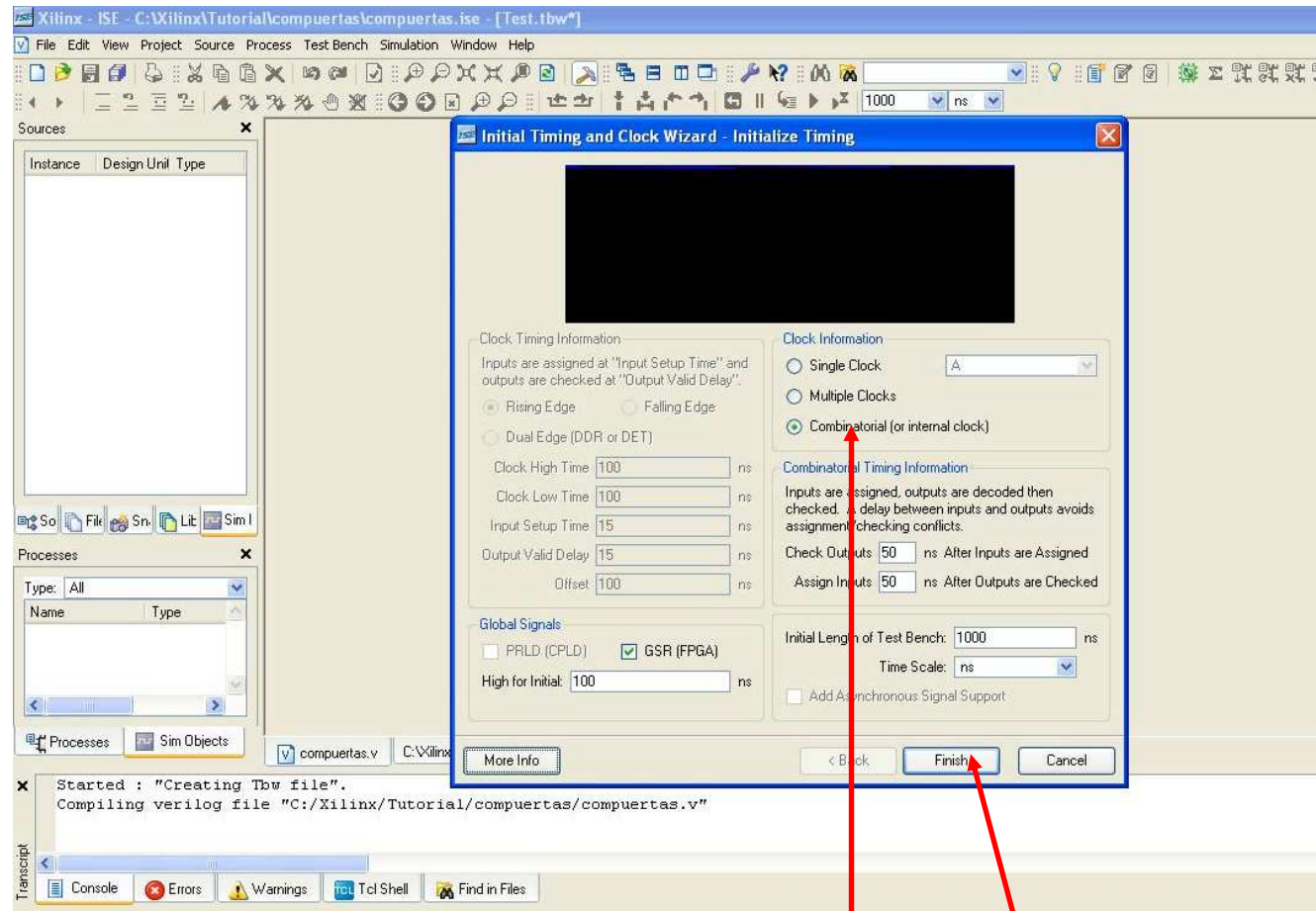
A continuación aparecerá el archivo que vamos a asociar a la simulación y damos Next.



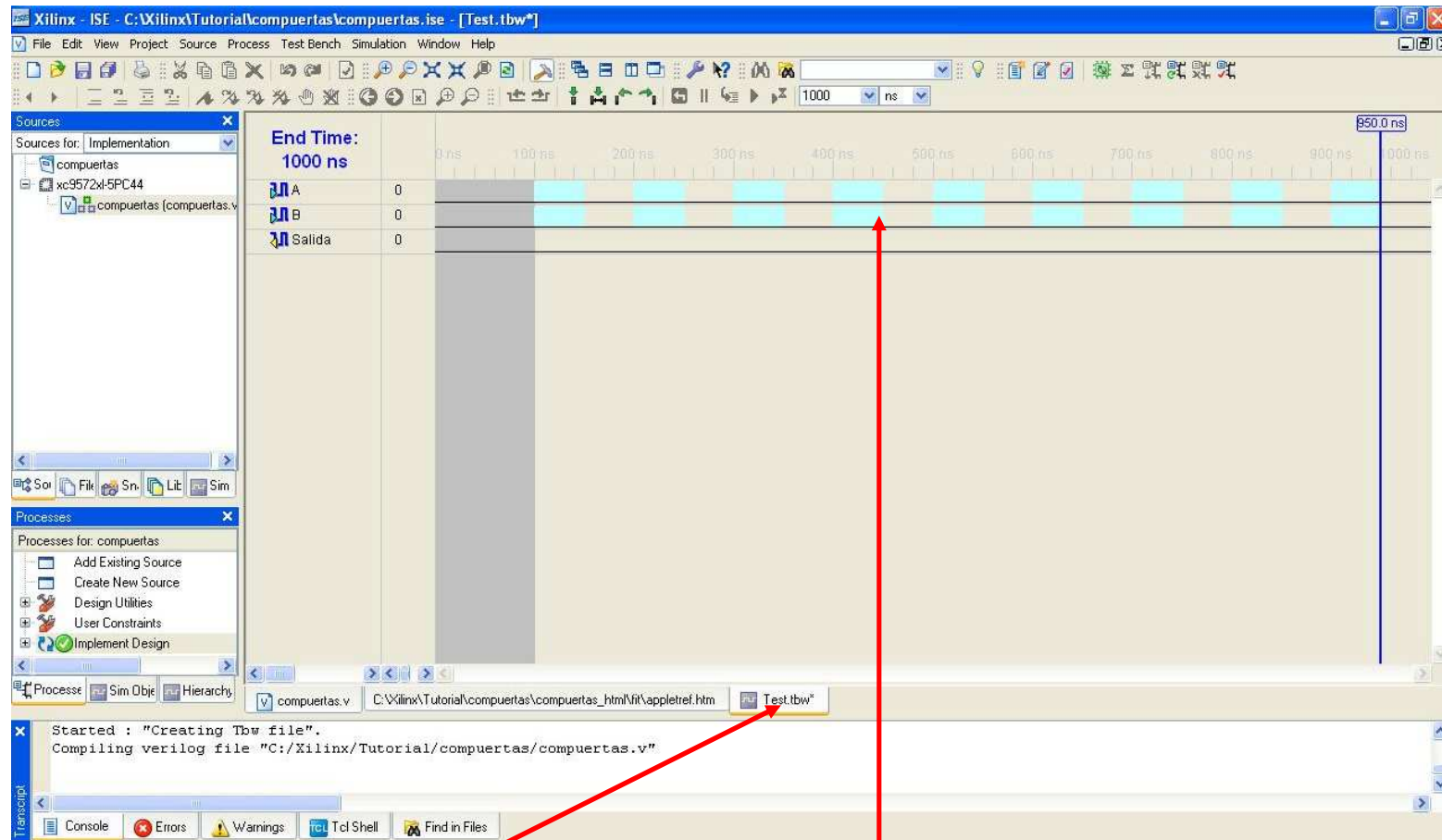
Como en los casos anteriores el sistema nos muestra un reporte de lo que hemos seleccionado, damos **Finish**



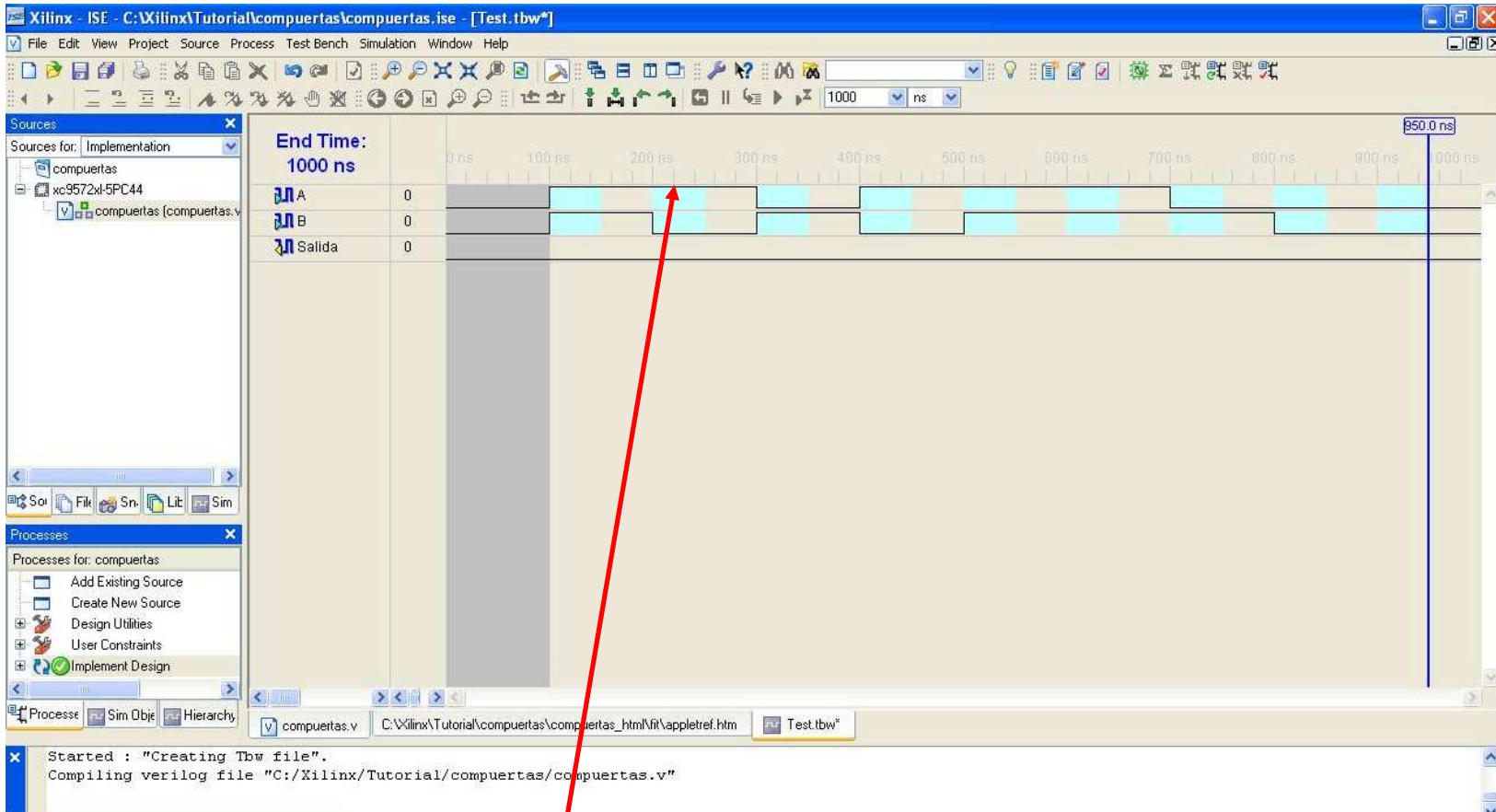
Seleccionamos OK



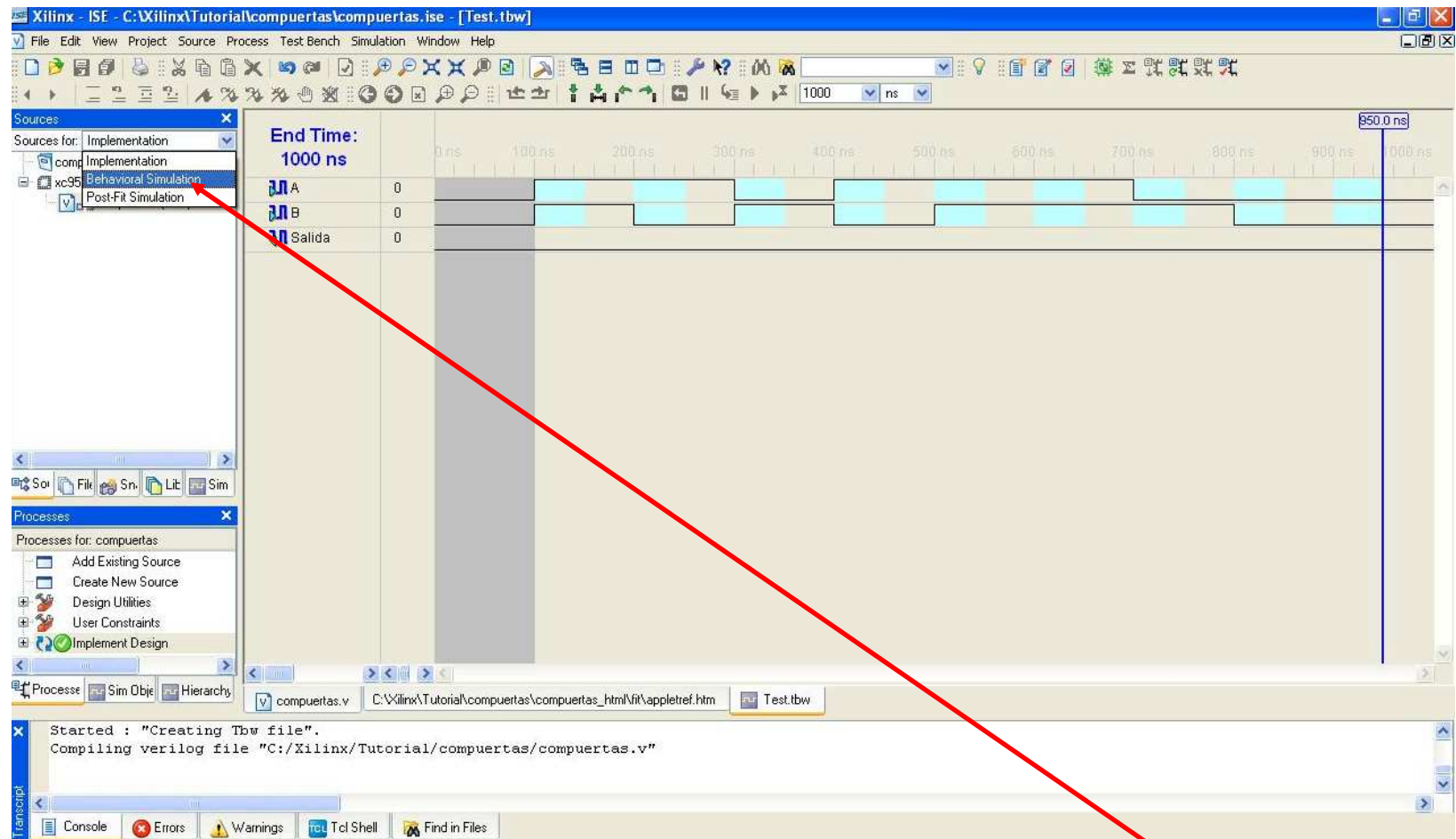
A continuación se selecciona la opción **Combinational**, ya que en esta etapa no utilizamos señales de clock o reloj. Luego damos **Finish**.



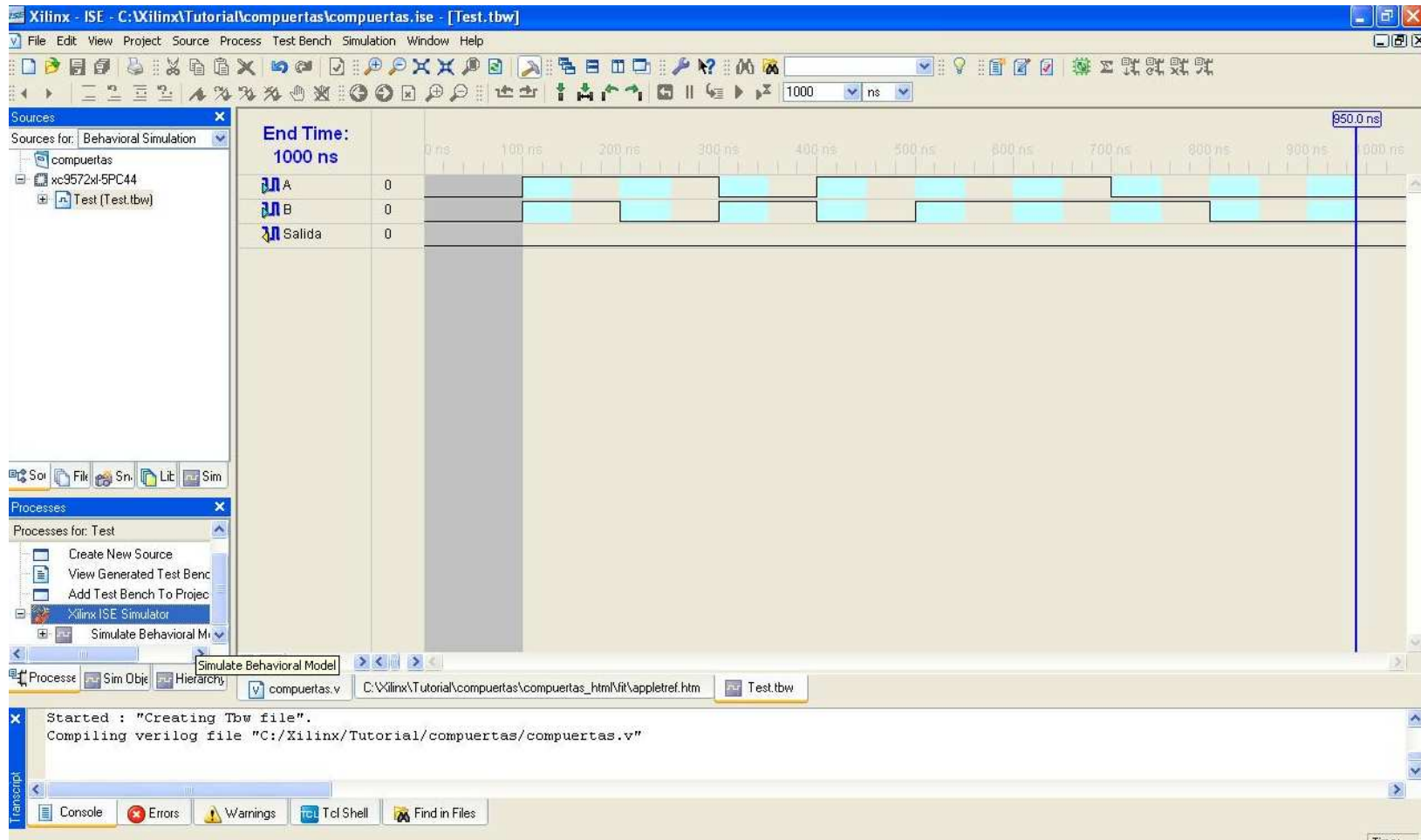
Nos aparece una solapa la cual contiene las entradas y salidas que definimos en nuestro proyecto. Lo que hacemos es dar estímulos a las entradas para luego lanzar la simulación y ver como se comporta nuestra descripción.



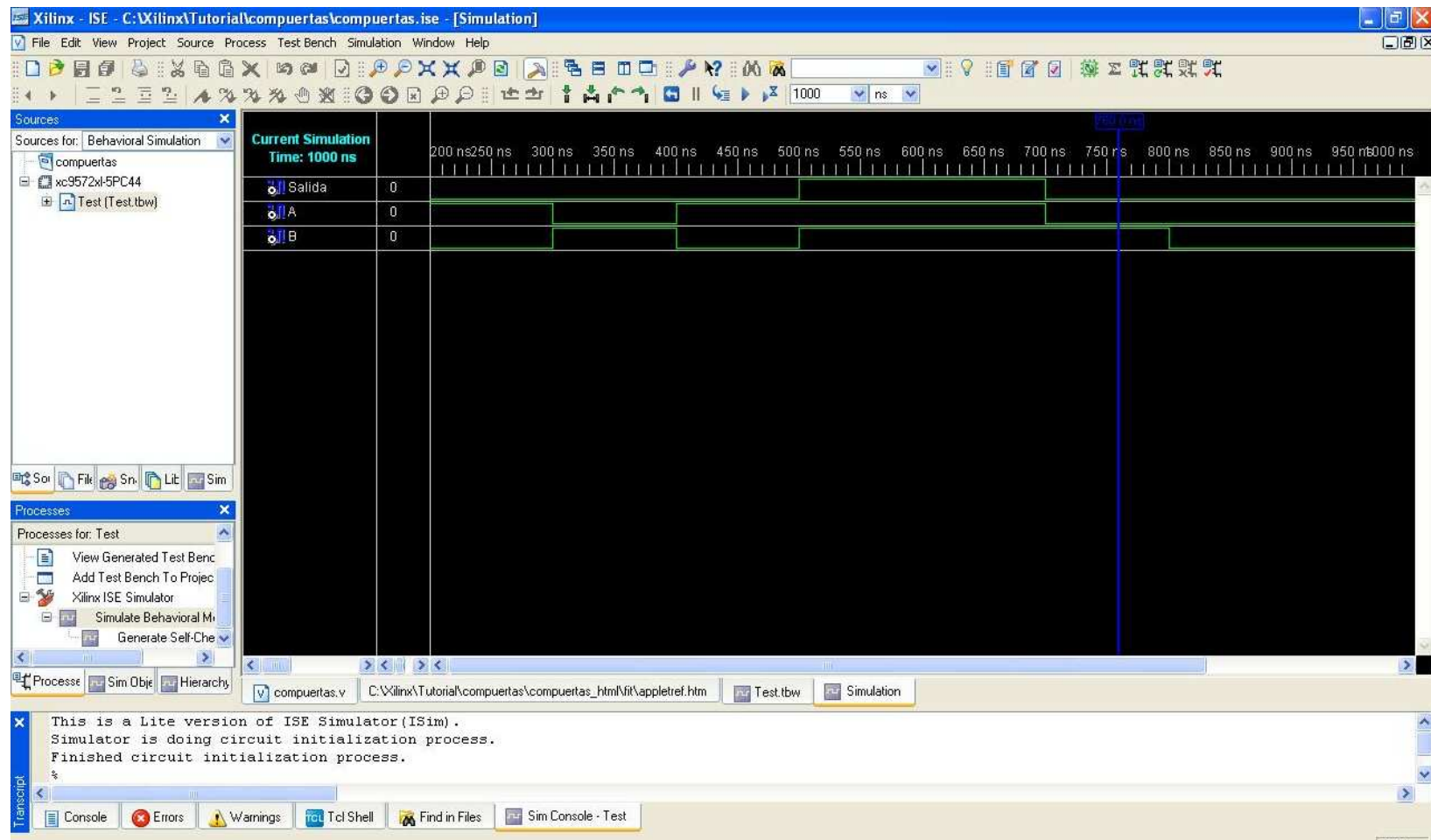
Damos los estímulos haciendo clic en las partes celestes en el área de las entradas. Básica mente es como tener representada una tabla de verdad.



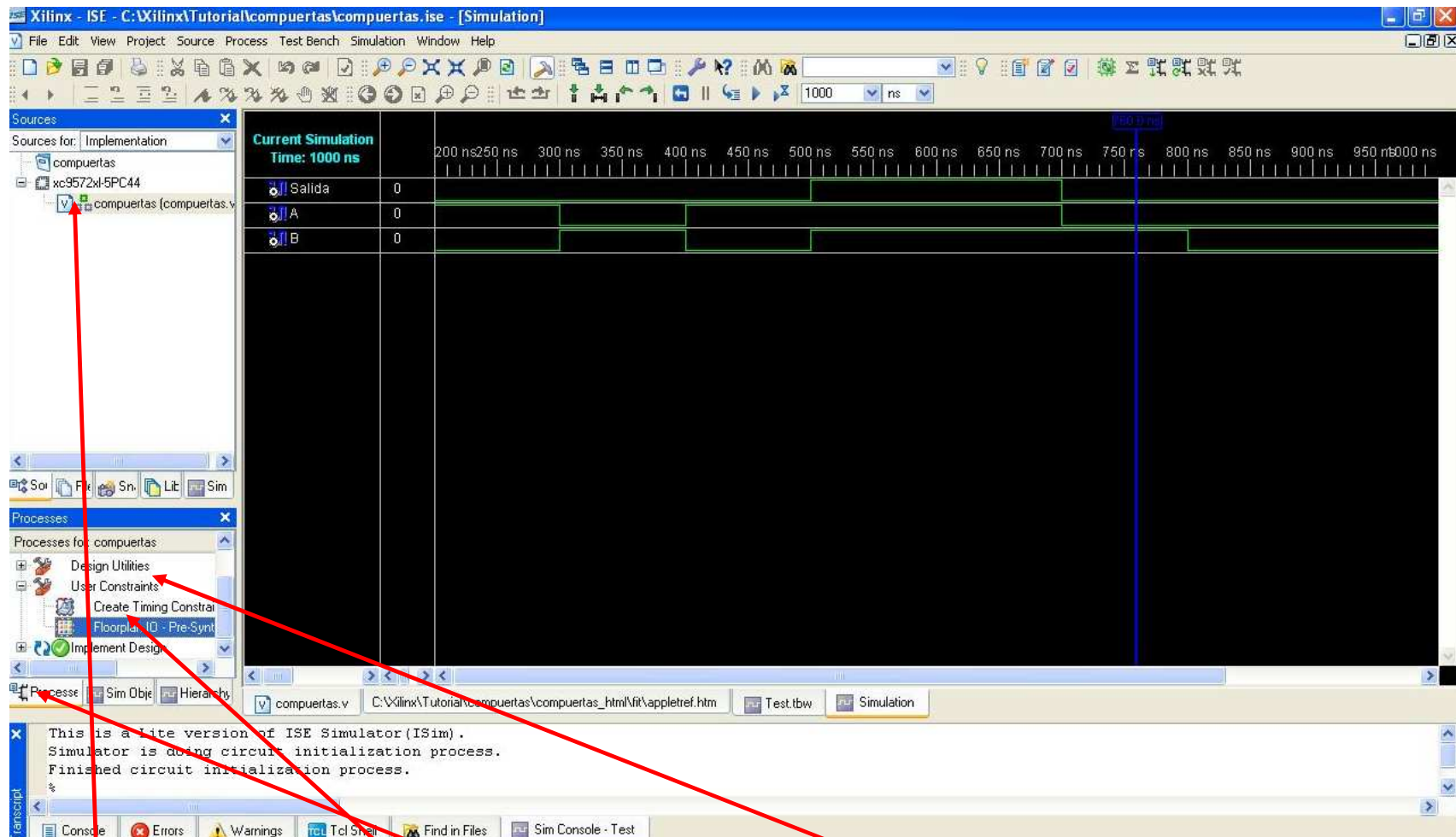
Salvamos el proyecto y en la ventana **Sources** seleccionamos **Behavioral Simulation**.



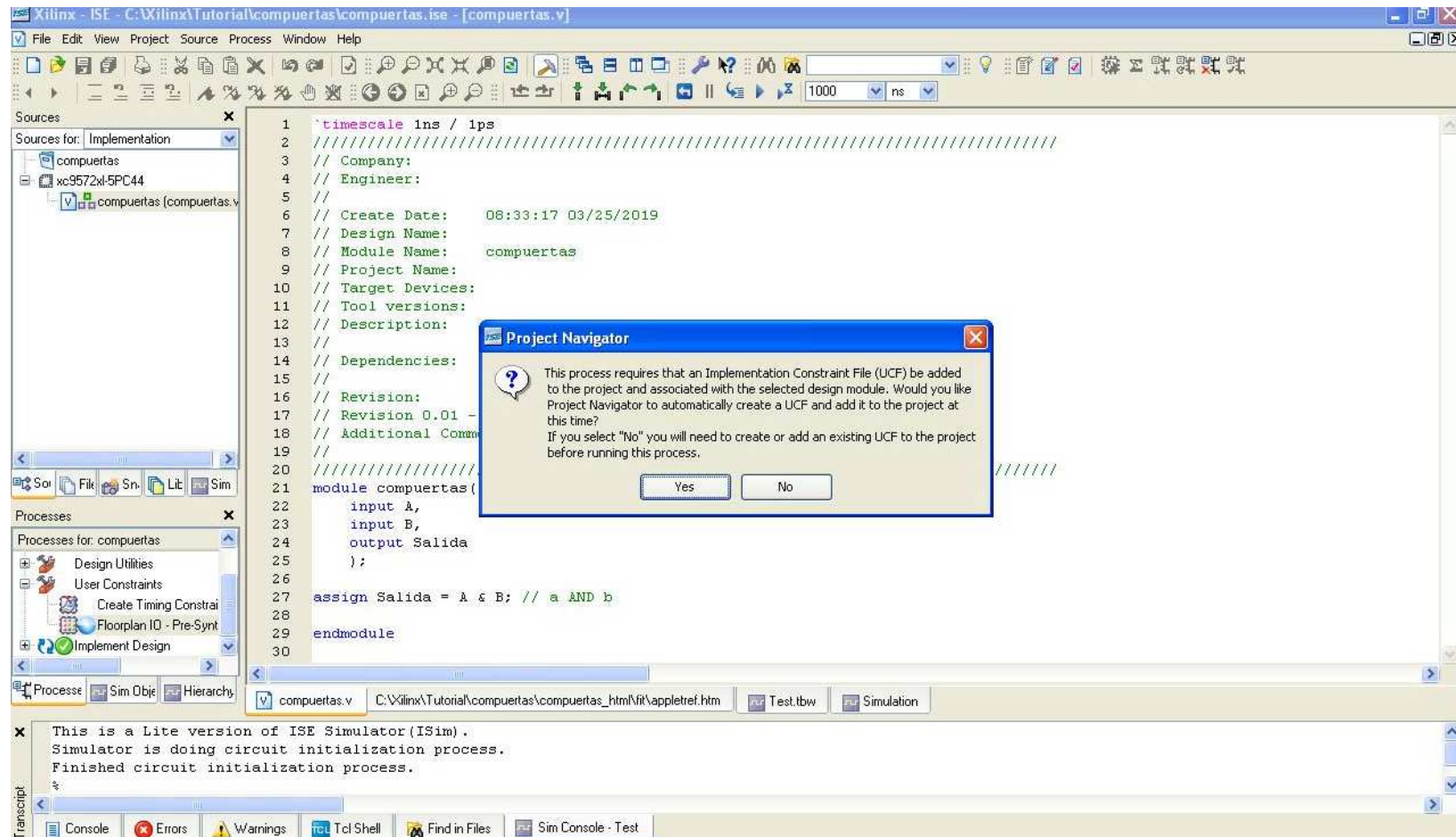
Hacemos un click sobre el archivo **Test.tbw** y en la ventana **Process** elegimos **Xilinx ISE Simulator**, expandimos y hacemos doble click en **Simulate Behavioral Model**, esta acción lanzará la simulación.



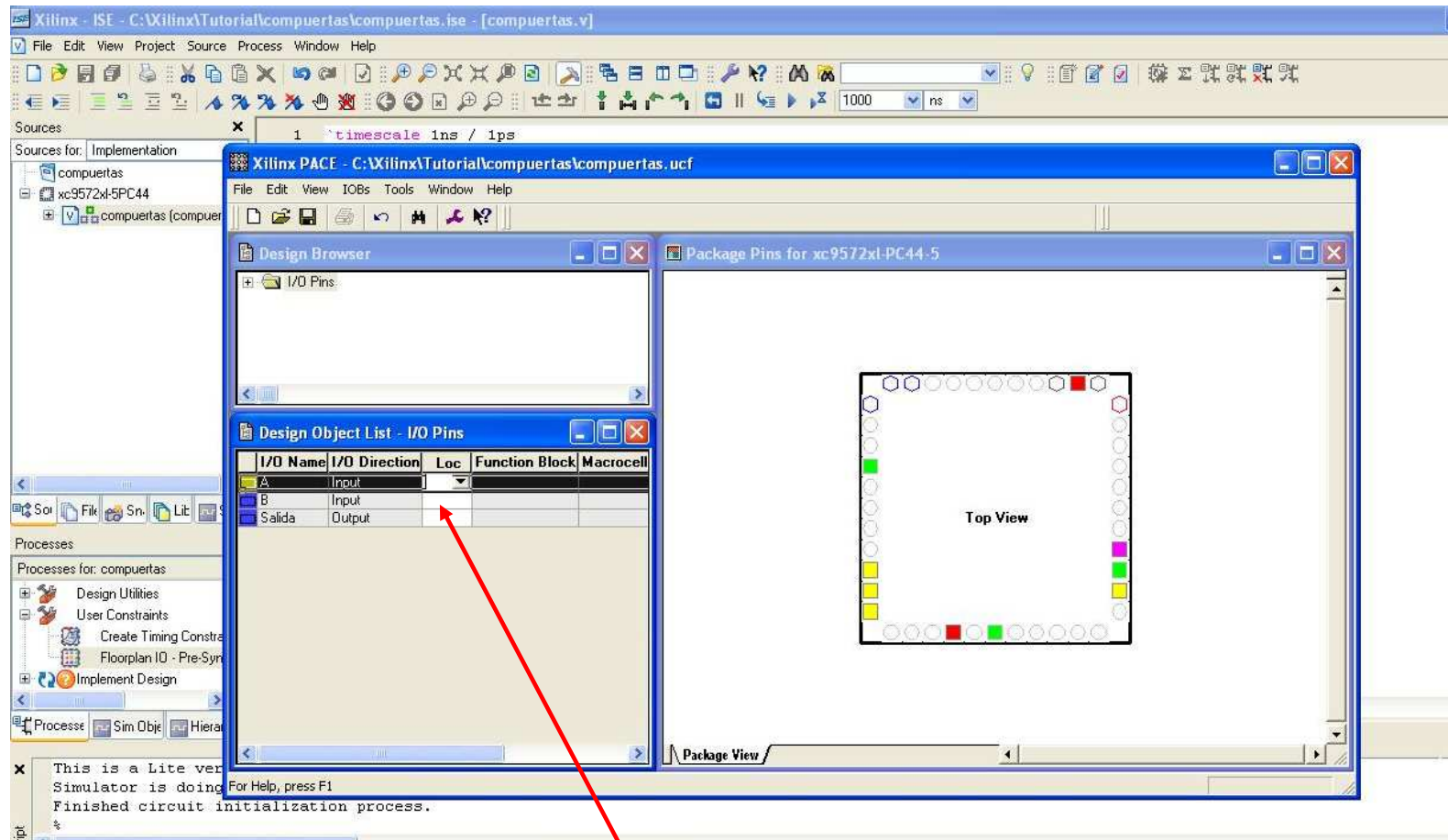
Si no se produjo ningún error se observa la simulación como en la imagen, donde podemos ver que se cumple el comportamiento de la función AND



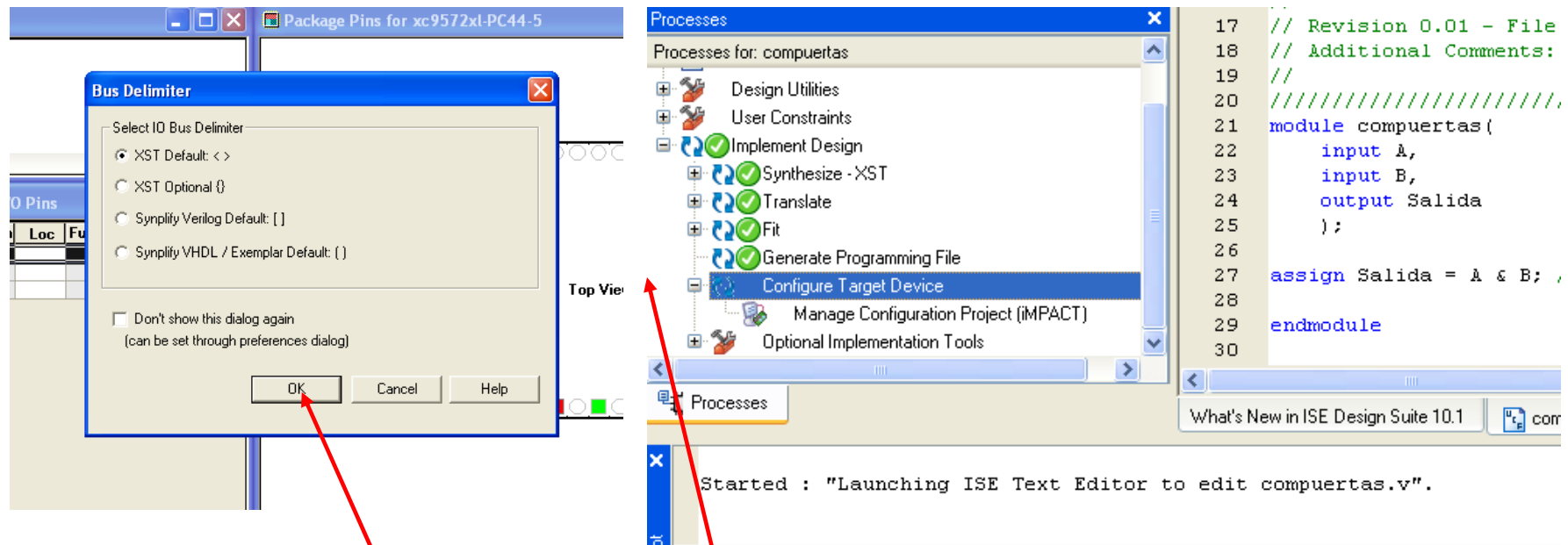
Seguidamente debemos realizar la asignación de pines para luego poder descargar nuestra descripción en nuestro CPLD. Para ello seleccionamos el archivo compuerta, luego en la solapa **Process** expandimos **Users Constraints** y hacemos doble click en **Floor IO Pre Synteis**



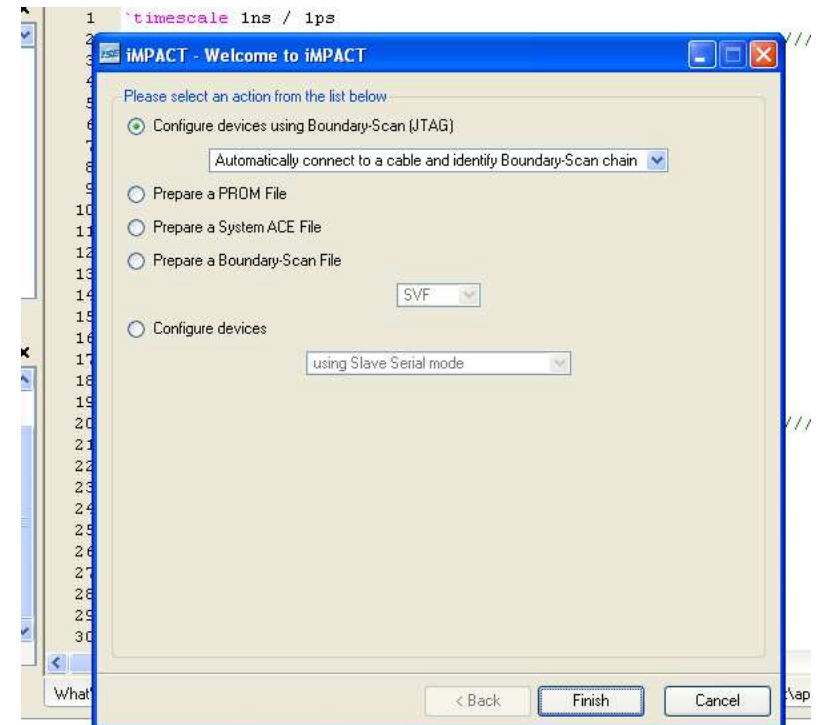
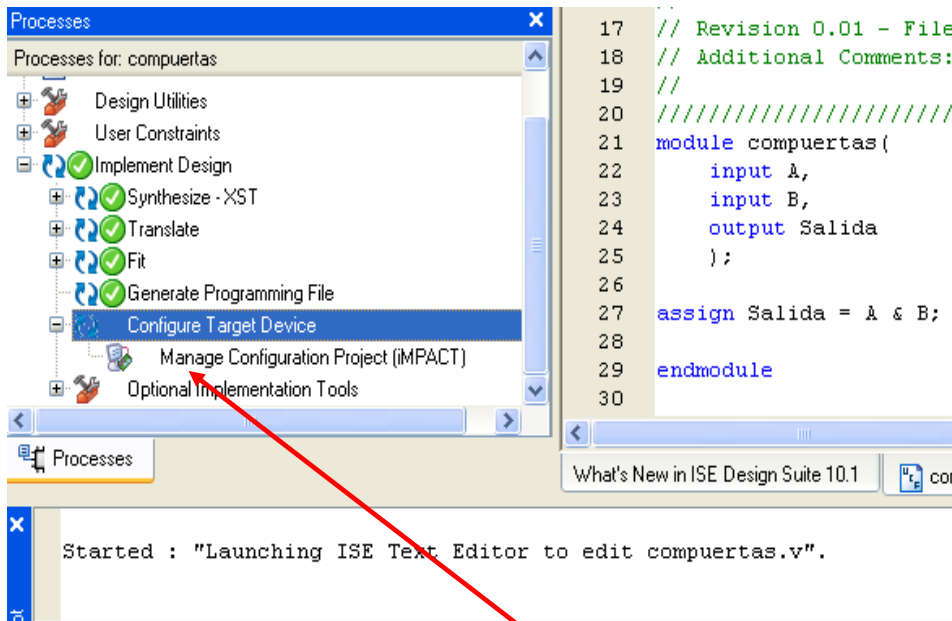
Hacemos click en **Yes** y de esta manera pasamos a la asignación de



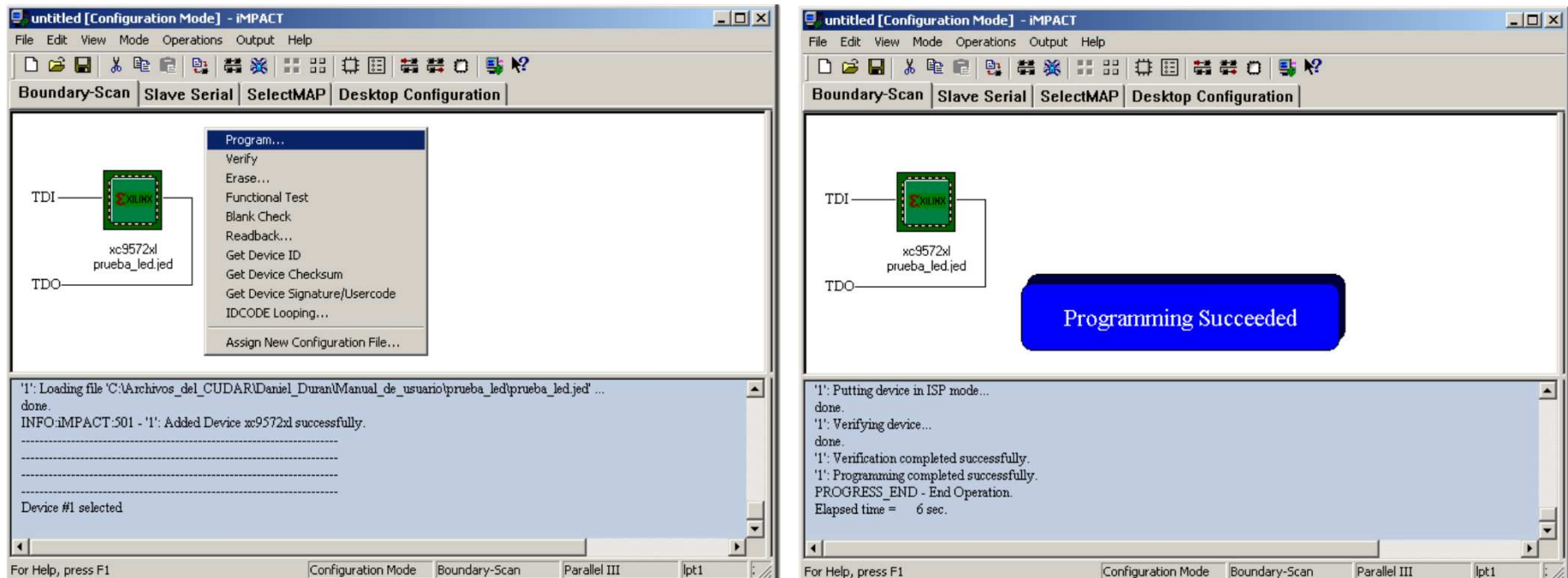
Seguidamente se abre la utilidad **Xilinx Pace** donde colocaremos los pines de entrada y salida de nuestro CPLD. Podemos ver en el lado izquierdo la representación del CPLD y en color los pines que están usados. Una vez asignados los pines de entrada y salida, salvamos y volvemos a compilar.



A continuación damos OK y compilamos, y vamos a la ventana de proceso en la opción **CONFIGURE TARGET DEVICE**, en esta etapa ya estamos en condiciones de programar el CPLD.



Hacemos doble click en **Manage configuration Project** y seguidamente damos **Finish**.



Nos aparecerá una ventana como la siguiente donde nos mostrará el dispositivo al que programaremos si el KiT se encuentra conectado. Hacemos click derecho sobre el CPLD, seleccionamos el archivo con extensión .JED y damos programar. Si la transferencia fue exitosa se mostrará un mensaje como en la imagen de la derecha.

FIN