

Tutorial Breve: uso de Quartus II para compilar diseños VHDL haciendo uso de ModelSim como simulador

Presentación

En la PEC2 utilizamos ModelSim para compilar y simular código VHDL. Como comentamos, ModelSim es un producto extendido en la industria e independiente de fabricante. Ahora bien, si queremos acabar haciendo funcionar un diseño sobre un dispositivo concreto, habrá que hacer uso de una herramienta de fabricante.

En la asignatura, y en concreto a la PEC3, veremos cómo pasar un diseño a un dispositivo comercial de Altera. Este fabricante proporciona un software – Quartus – para trabajar con sus dispositivos. Con este tutorial, pretendemos exponeros de forma breve como trabajar con el entorno Quartus, pero a la vez, usando ModelSim como herramienta de simulación, puesto que Quartus permite invocarlo.

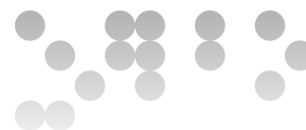
Objetivos

- Compilar y validar diseños VHDL mediante Quartus.
- Utilizar ModelSim de forma integrada con Quartus para llevar a cabo la simulación.

1. Introducción

En este tutorial se describe brevemente como modelar un diseño electrónico con VHDL y simularlo correctamente. Para hacerlo, se analizará como generar un proyecto con el software Quartus. Como veremos a continuación, Quartus es un software de diseño de Altera que nos permite implementar diseños a base de código VHDL.

Para comprobar el correcto funcionamiento de nuestros diseños – y como vimos a la PEC2 – el proceso consiste al compilar por – una vez no hay errores – simular el diseño. En este tutorial básicamente se quiere hacer entender qué se tiene que hacer para poder compilar con Quartus y simular correctamente el código que implementamos, así como que hay varias formas de hacerlo. A pesar de que Quartus tiene la posibilidad de uso de un simulador propio, usaremos ModelSim para ser muy extendido en el mercado, y soportado por diferentes fabricantes de dispositivos.



2. Funcionamiento del Quartus II

Esta sección presenta el funcionamiento del entorno Quartus II. Supondremos que hemos generado un diseño con VHDL (a modo de ejemplo, el planteado como “Hello world” en su punto 2.2. del Módulo 3 de los materiales de la asignatura, y que os adjuntamos también en el punto 2.1. de este tutorial).

Antes de empezar, indicar que en caso de ser necesario ampliar la información dada, se puede ir al menú ayuda o el botón sensible de ayuda (lo podéis ver a la Figura 1 a la barra de herramientas de Quartus II, con un interrogante).

Llegados a este punto, empezamos a trabajar con el Quartus II. Si abrimos el programa nos encontramos la ventana que se muestra a la Figura 1 (la figura corresponde a la versión 13.1 del software. Puede haber ligeras diferencias en función de la versión):

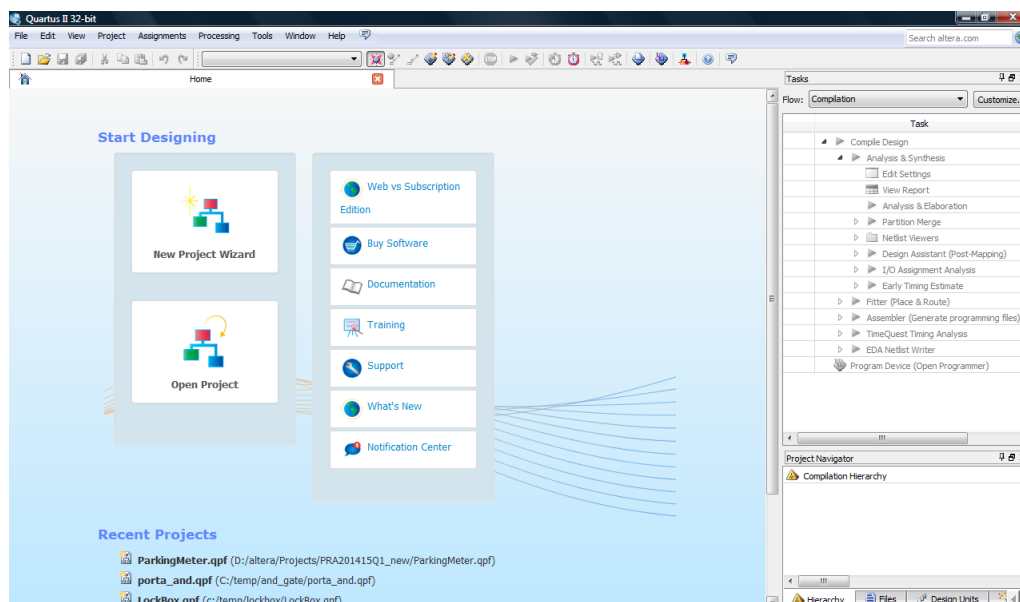
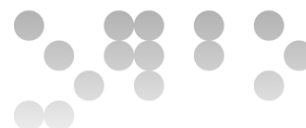


Figura 1: Ventana principal del Quartus II

Caso que hubierais trabajado ya con proyectos dentro del programa, os aparecerán a la pantalla anterior donde dice “Recent projects”). Supondremos ahora que partimos de cero y que el único que tenemos es un fichero (digamos hello.vhd) dónde hemos copiado el código VHDL de ejemplo.

2.1. Creación de un proyecto

Un proyecto de Quartus II tiene que ser creado con el fin de describir, compilar, simular y programar un diseño digital. Para crear un proyecto sólo



hace falta, en la pantalla que se muestra a la figura 1, simplemente haced click en “New Project Wizard”. Os aparecerá la figura 2.

Nota: Si al aparecer aquella pantalla habéis pulsado la tecla “ESC”, o bien estáis dentro del programa con otro proyecto, siempre podéis crear un proyecto nuevo yendo al menú y escogéis la opción “File → New Project Wizard” y hacéis clic en Next. La primera vez sale una ventana introductoria, y simplemente tenemos que hacer “Next”, para dar los datos del proyecto:

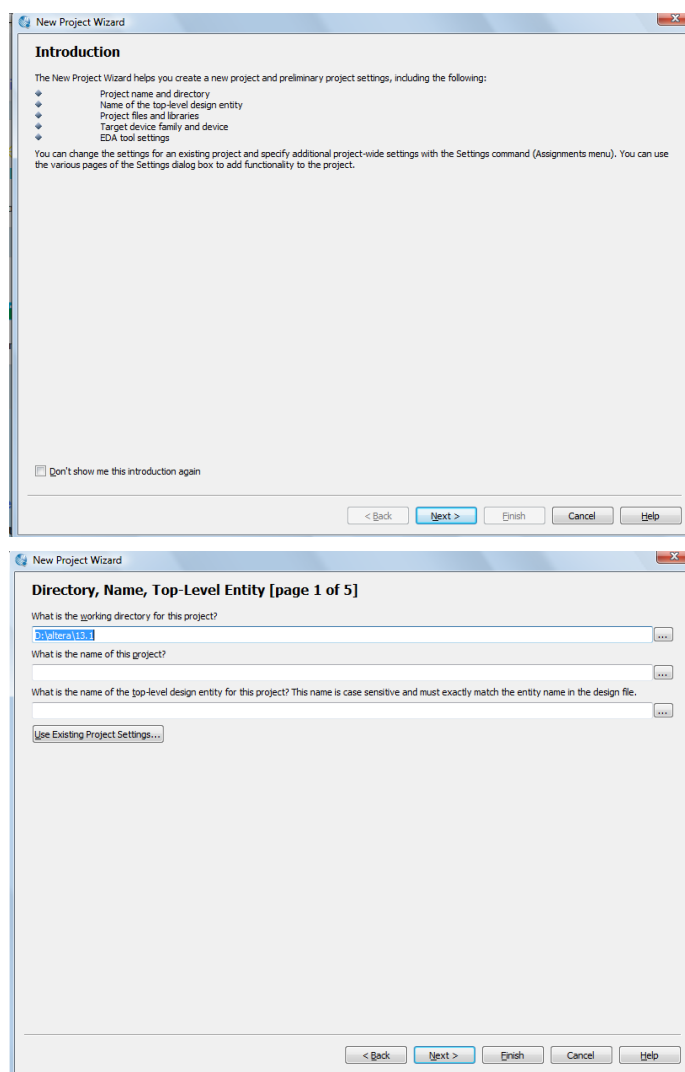

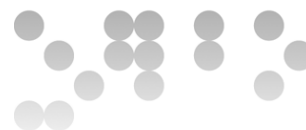


Figura 2(a y b): Creación de un nuevo proyecto

2. Introducís el nombre y el path de vuestro proyecto, o seleccionáis mediante la casilla  algún proyecto que ya esté creado.



3. Se puede ver que hay dos espacios vacíos para introducir el nombre, el primero es para introducir el nombre del proyecto y el segundo para introducir el nombre de la entidad superior. En el ejemplo que nos ocupa, si tomamos la entidad puerta_and, podemos poner este nombre como nombre del proyecto.

4.- Haced click en “Next”. Obtendréis la figura 3:

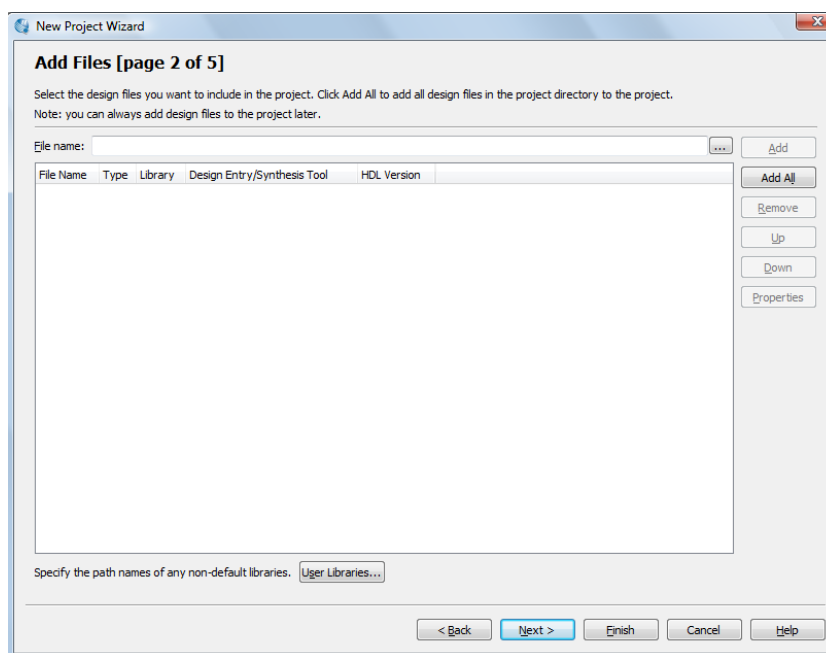


Figura 3: Selección de ficheros

Haced click en icono de los “...” y elegís el fichero .vhd donde tenéis el código para compilar y simular. Una vez lo tengáis haced click a “Add”.

4. Podéis hacer clic a finalizar (Finish), puesto que a continuación veremos como compilar y simular. La opción Next nos preguntaría sobre en qué tipo de dispositivo queremos implementar el diseño, pero no es ahora nuestro objetivo. Habremos llegado a una ventana del tipo:

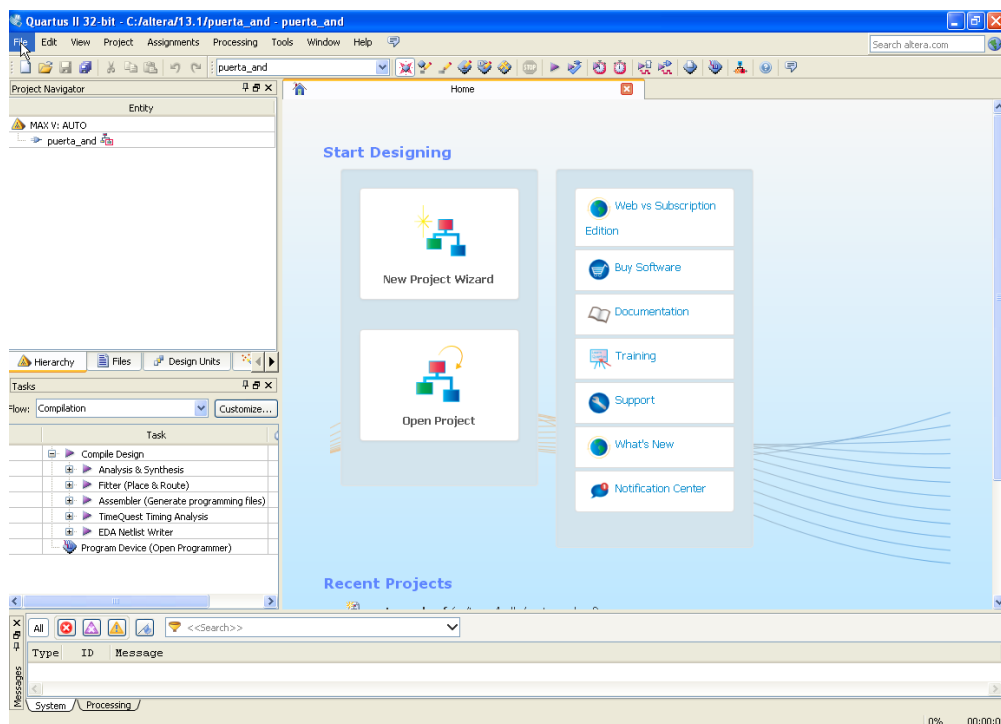
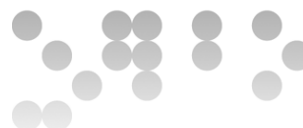


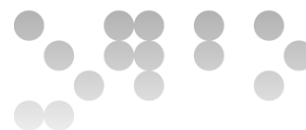
Figura 4: Entorno Quartus con proyecto abierto

La distribución de bloques de la pantalla puede depender de vuestra configuración. En todo caso, veréis que hay un bloque que es el “Proyecto Navigator” (a la Figura 4 es a la parte inferior derecha) . Si hacéis click en este cuadrante inferior encima la pestaña “File”, veréis el fichero que habéis añadido. Si hacéis doble-click sobre el fichero “vhd” se abrirá el código VHDL. Utilizamos en este caso el código de la puerta_and que encontramos también en los materiales (Módulo 3). En concreto:

```
library ieee;
use ieee.std_logic_1164.all;

-- entity definition
entity puerta_and is
    port(
        in0, in1: in bit;
        out0: out bit
    );
end entity puerta_and;

-- architecture definition
architecture arquit_and of puerta_and is
begin
    out0 <= in0 and in1;
end architecture arquit_and;
```



En el entorno gráfico lo vemos cómo:

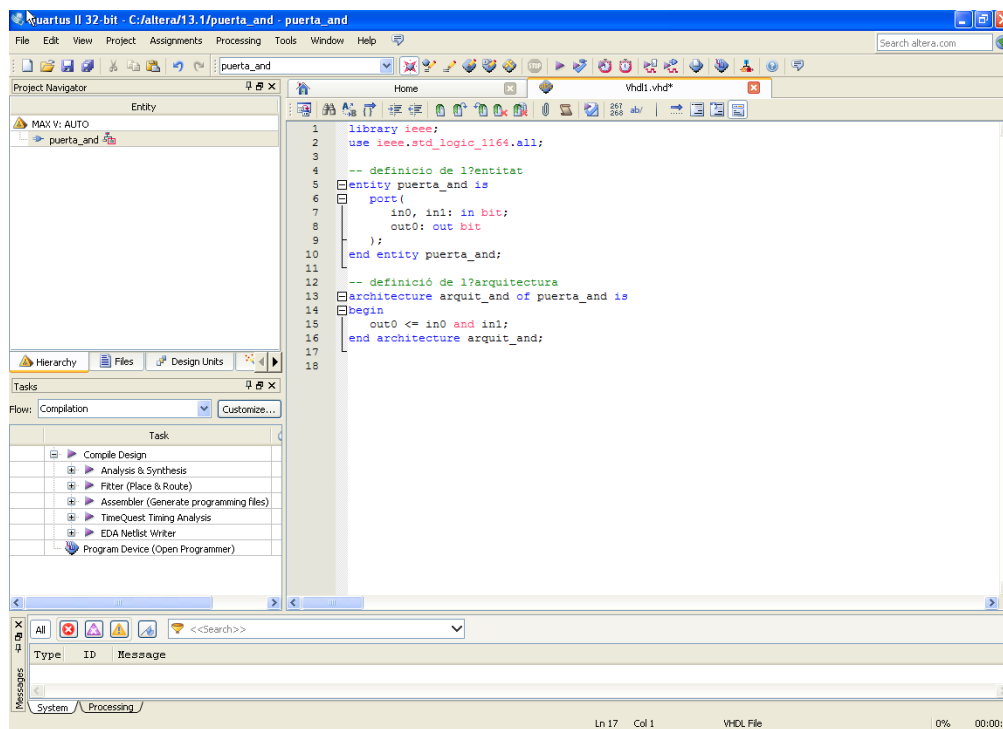
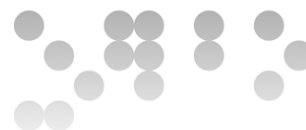


Figura 5: Apertura del fichero .vhd

2.2. Compilación

A efectos de hacer una compilación básica sólo tendréis que hacer “Processing – Start Compilation”. Con esto podréis compilar el código, y el Quartus os informará de si hay errores al diseño o por el contrario todo es correcto. En este caso, todo es correcto (obviaremos el warning para nuestro estudio):



Flow Summary	
Flow Status	Successful - Fri Oct 09 07:58:19 2015
Quartus II 32-bit Version	13.1.0 Build 162_10/23/2013 SJ Web Edition
Revision Name	puerta_and
Top-level Entity Name	puerta_and
Family	MAX V
Total logic elements	1 / 40 (3 %)
Total pins	3 / 30 (10 %)
Total virtual pins	0
UFM blocks	0 / 1 (0 %)
Device	5M40ZM6-4C4
Timing Models	Final

Figura 6: Compilación

2.3. Simulación

2.3.1. Configuración del entorno para utilizar ModelSim como simulador

Ha llegado el momento de simular. Como hemos dicho, Quartus ofrece diferentes posibilidades para el uso de simuladores. Si habéis utilizado el paquete de software que encontraréis en el aula, o bien habéis ya descargado conjuntamente de la web ambos programas, veréis que Quartus ya incluye la ruta hacia el simulador ModelSim. Lo podemos ver al menú Tools – Options y en el apartado “EDA tool Options” dentro del bloque “General” como muestra la figura siguiente:

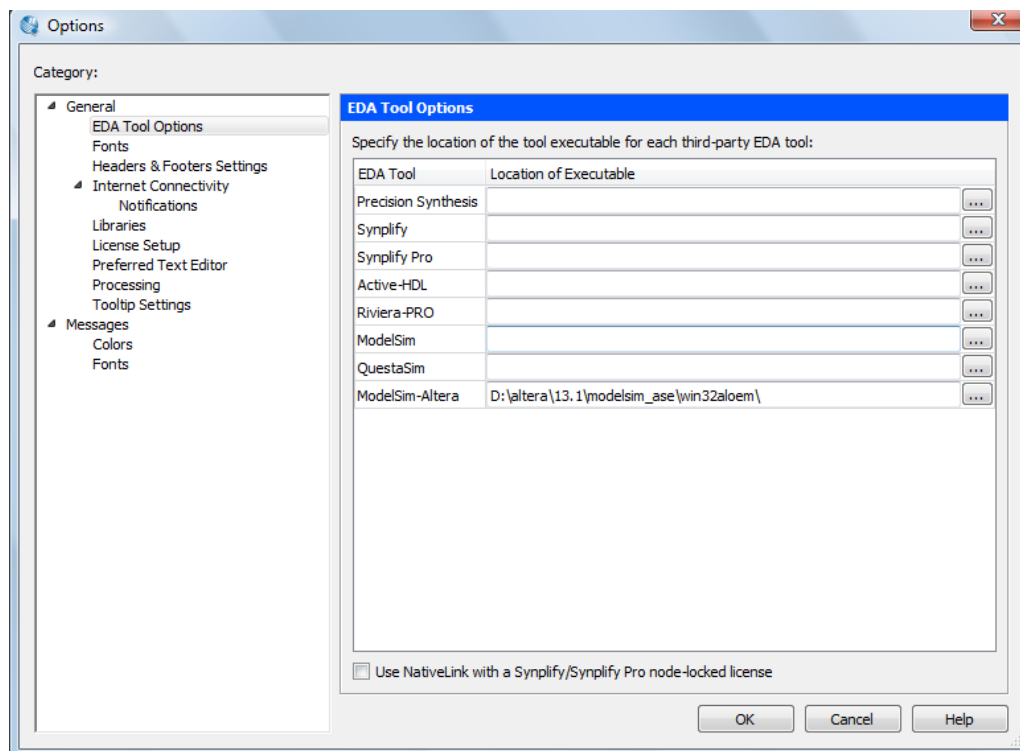
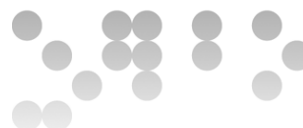


Figura 7 : Indicación del path donde se encuentra ModelSim

Cómo veis, el path hacia el ModelSim-Altera se encuentra ya indicado. Si no fuera el caso, tenéis que poner la ruta donde se encuentra (en el caso de la figura `d:\altera\13.1\modelsim_ase\win32aloem\`). Fijaos que la pantalla ya indica que son rutas hacia “third party tools” (programas terceros que se integran con el entorno Quartus). Para nuestras necesidades tenemos bastante con este. Para usuarios Windows, por favor, validad que hay la ‘\’ final al path, puesto que si no, no encuentra el ejecutable.

Una vez hecho esto, nos hace faltar indicar a Quartus que en el momento de simular, tendrá que utilizar la herramienta. Por eso vamos a Assignments – Settings – EDA Tool Settings – Simulation. Allí especificaremos las opciones de simulación que serán las de la figura siguiente:

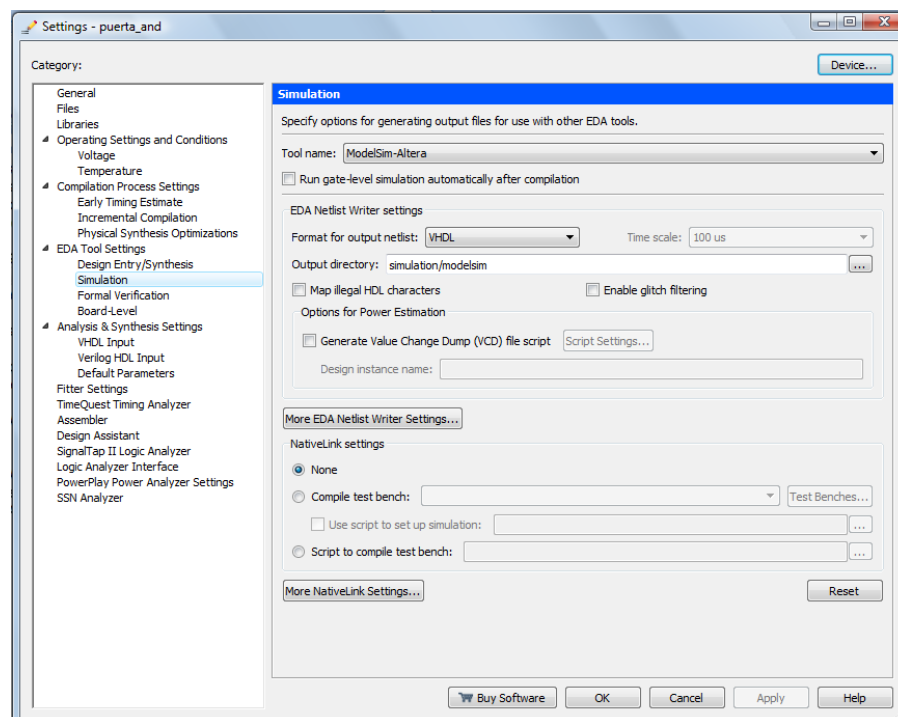
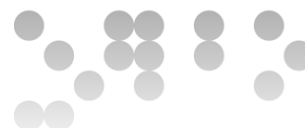


Figura 8 : Uso de ModelSim para simular

Fijaos que explícitamente elegimos “ModelSim-Altera” como herramienta de simulación. Dejad inalteradas el resto de opciones, tal y cómo se muestran en la figura anterior (podéis cambiar si queréis –a pesar de no ser necesario– el directorio de salida –Output Directory– pero dejad el resto de opciones como figuran).

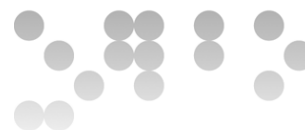
Con esto estamos listos para ejecutar una simulación con ModelSim desde el entorno Quartus.

2.3.2. Proceso de simulación

Simular con Quartus es bastante sencillo. Sólo habrá que añadir un fichero de formas de onda. Quartus como tal no trabaja con ficheros de bancos de pruebas. Conceptualmente, el que haremos es generar las formas de onda y hacer que ModelSim lo simule, todo de una forma gráfica.

Añadir las formas de onda tendrá dos partes: por un lado, añadir las entradas que queremos modificar. Por otro lado, añadir las salidas que tienen que generarse en base a cómo se han modificado las entradas. Es decir, añadimos primero los ‘inputs’ y vemos como se generan los ‘outputs’.

Una vez configurado Quartus por utilizar ModelSim cómo se describe en el punto anterior, simplemente haremos Tools – Run Simulation Tool – RTL



Simulation. Si hemos configurado correctamente se abrirá ModelSim, software que ya conocemos, con una ventana similar a la siguiente:

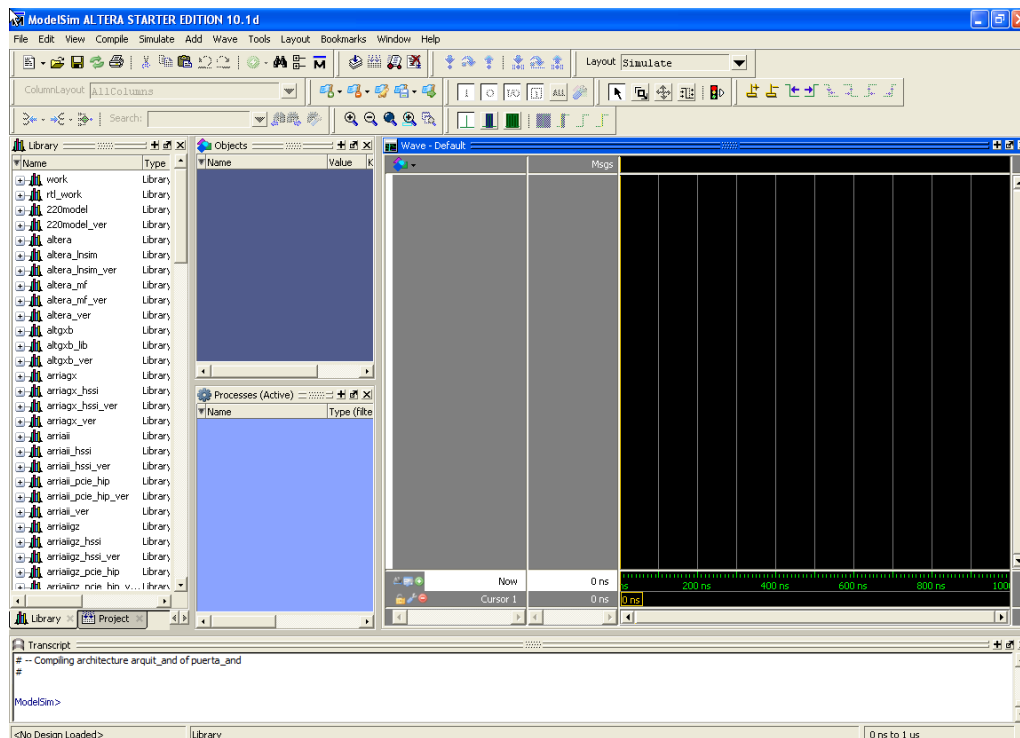
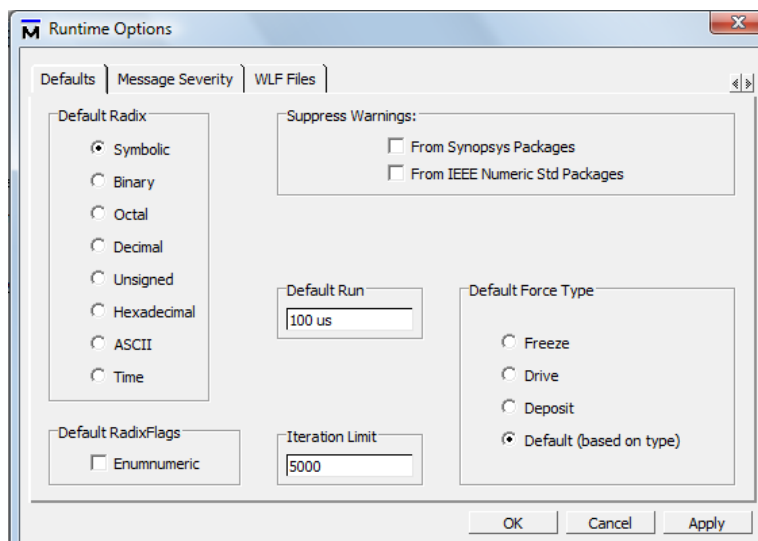


Figura 9: Ventana de ModelSim invocada desde Quartus

En primer lugar, configuraremos la duración de la simulación. Podemos hacerlo al menú Simulate – Runtime Options, donde por ejemplo indicaremos 100 microsegundos:



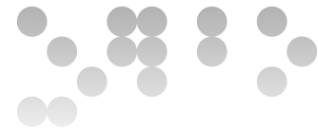


Figura 10: Duración de la simulación

Hecho esto, indicaremos a ModelSim que queremos simular. Vamos al menú Simulate – Start Simulation. Nos saldrá la ventana que conocemos de ModelSim, pero ahora sin banco de pruebas. Seleccionamos la entidad que queremos simular:

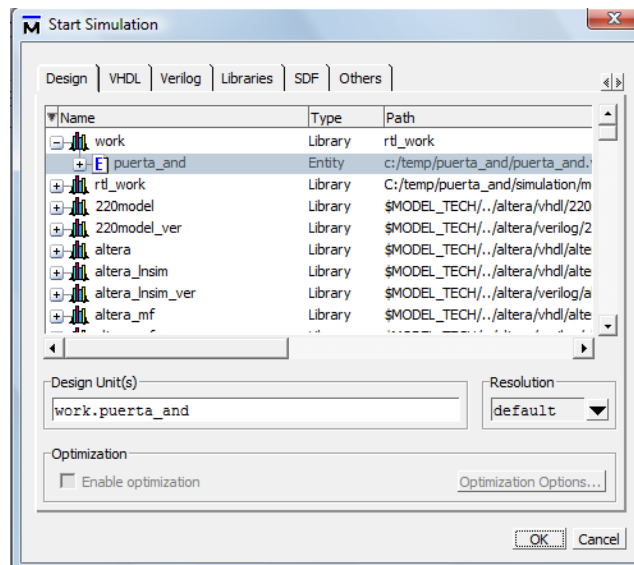


Figura11: Selección de la entidad a simular

Llegamos así a una pantalla del tipo:

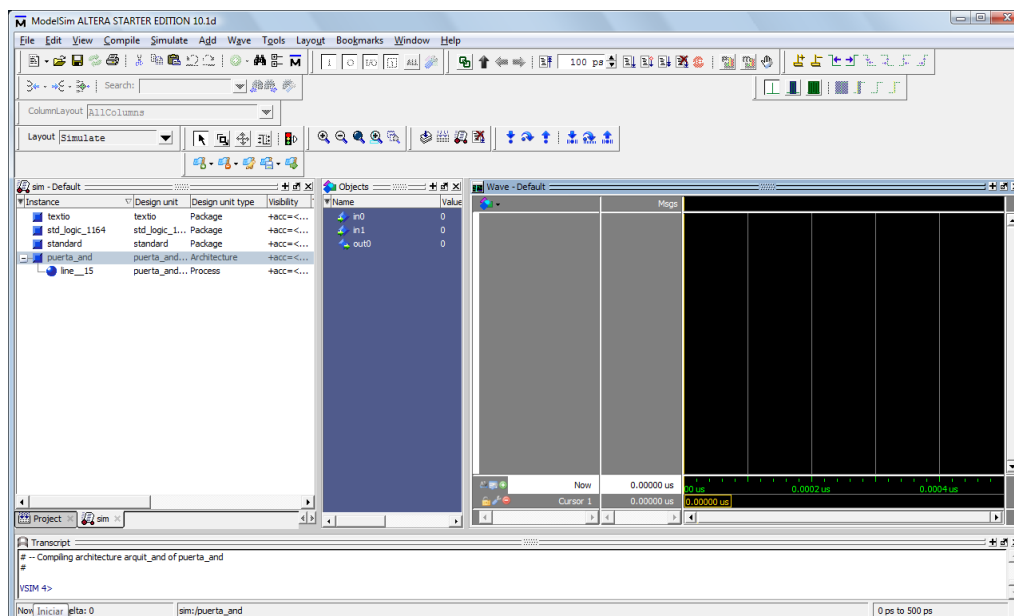
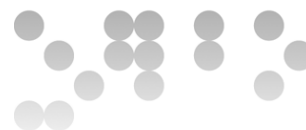


Figura 12: Pantalla de ModelSim con la entidad preparada para simular

2.3.3. Añadir las entradas a la simulación.

Hemos dicho puesto que el proceso tendrá dos partes: añadir las entradas y ver el resultado a la salida. Para añadir las entradas, antes que nada garantizaremos que a la pantalla anterior sale la pestaña Objects. Si no sale por defecto haremos View – Objects.

Para añadir las entradas deseadas (en este caso in0 e in1), simplemente hay que ir encima los objetos, y con el botón derecho, elegir la opción Modify – Apply Wave. En este caso lo haremos para in0 e in1:

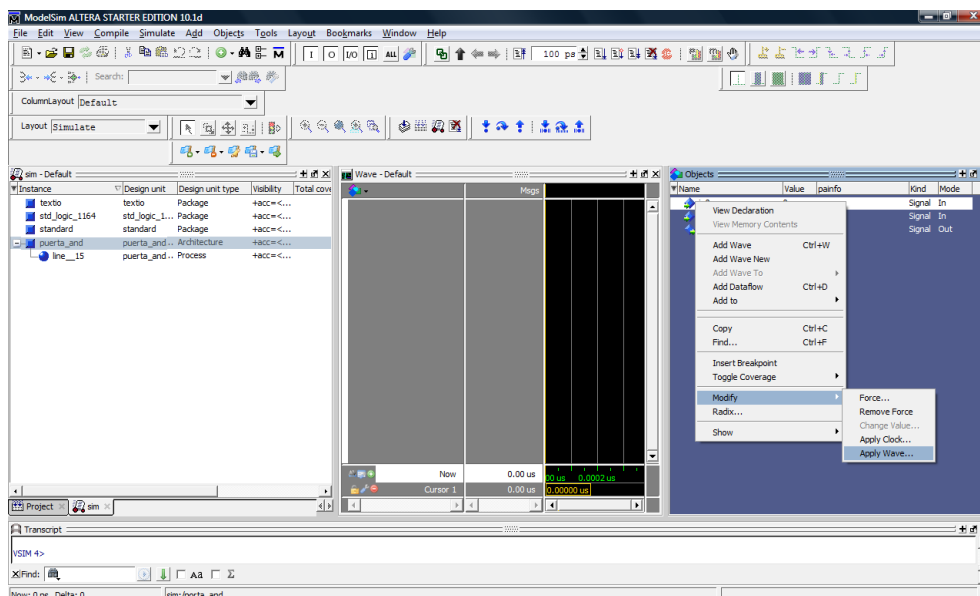
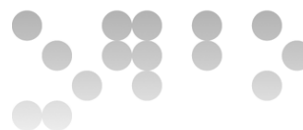


Figura 13: Añadido de formas de onda mediante edición

Se nos pregunta qué tipo de señales serán:

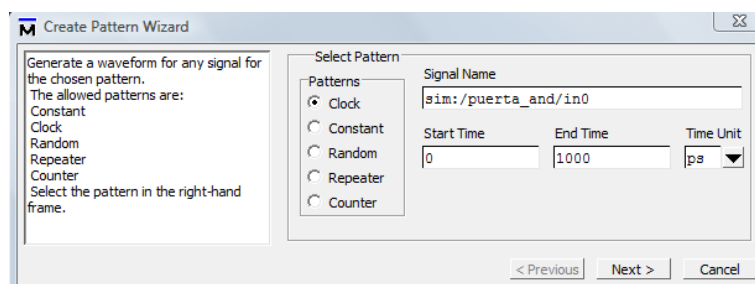


Figura 14: Tipo de formas de onda

Hacemos click en “Next” y damos valor de constante:

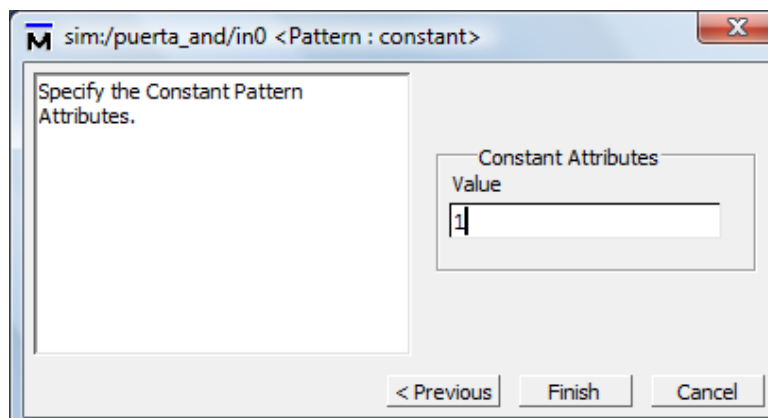
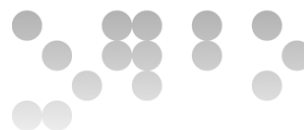


Figura 15: Valores de la forma de onda para una constante



Vemos como la señal queda añadida:

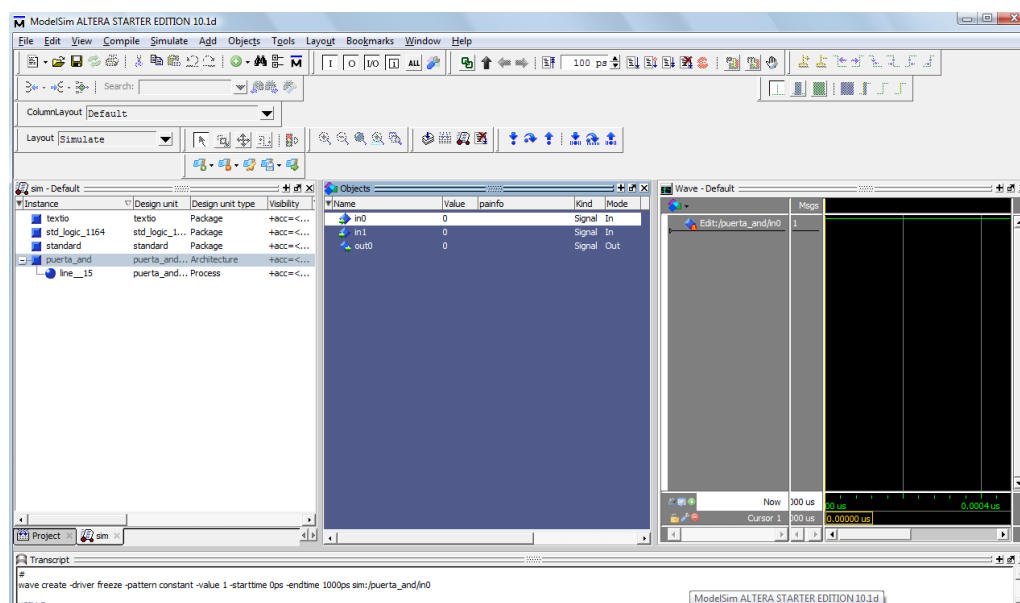
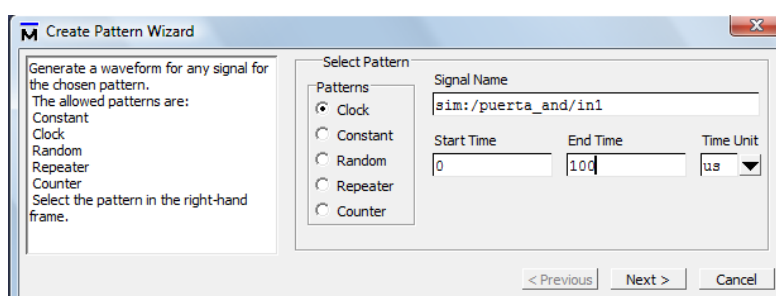


Figura 16: Forma de onda resultante

En el in1 añadimos, a título de ejemplo, una señal de tipo clock. No tiene sentido (conceptualmente) para nuestro ejemplo, pero lo añadimos puesto que nos será útil cuando simulamos sistemas que requieren un reloj – sistemas secuenciales –. Añadiremos un clock de 100 microsegundos con periodo de 20 microsegundos:



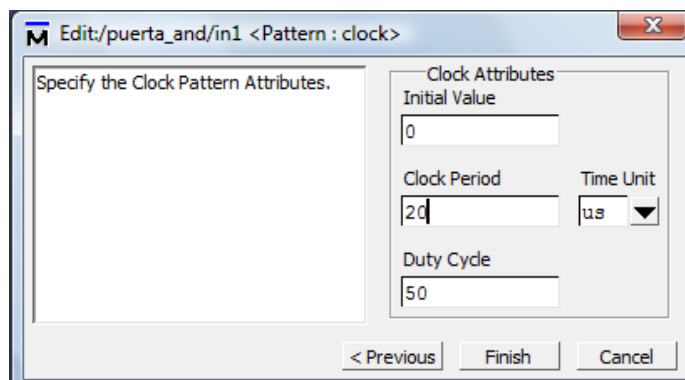


Figura 17 (a y b): Añadiendo una señal clock

Ya tenemos las señales de entrada. Podemos editarlos desde la ventana 'Wave'. En la ventana Wave, usando el botón derecho, podremos modificar las formas de onda, yendo al menú "Edit – Create/Modify Waveform". Probadlo. A modo de ejemplo, modificad la señal in0, dando un valor constante de cero entre 20 y 100 microsegundos. Veréis que se modifica la ventana Wave tomando este valor. Lo vemos a la figura siguiente:

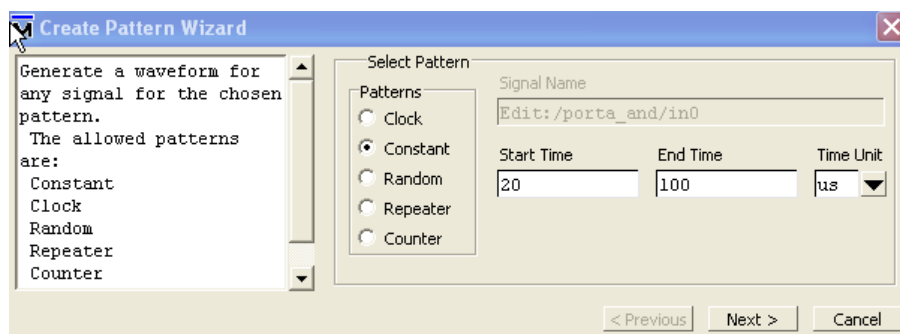
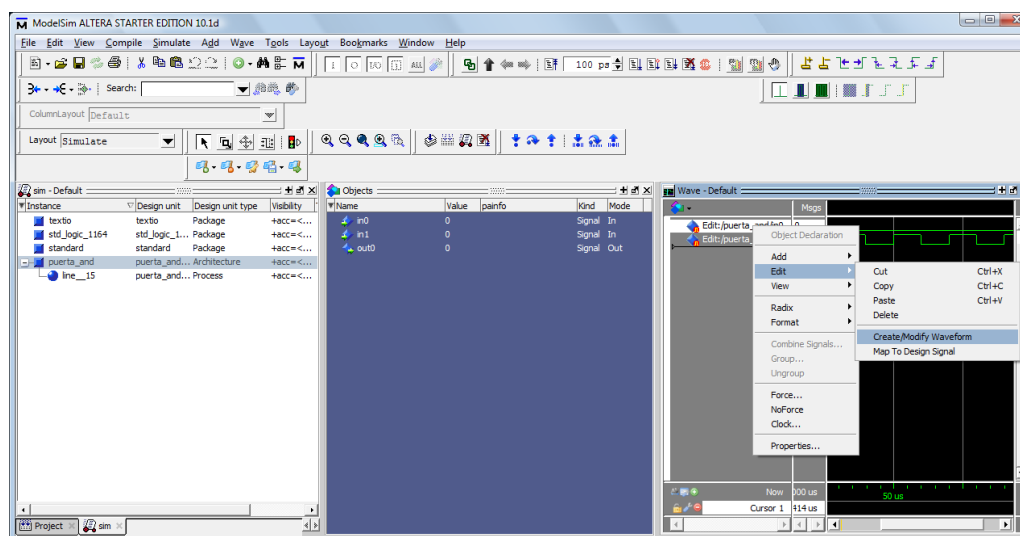
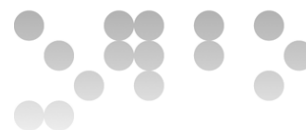


Figura 18 (a y b): Modificación de una señal de entrada



Falta sólo añadir las señales de salida que tiene que calcular la simulación. Lo hacemos en su punto siguiente.

2.3.4. Añadir los nodos de salida

En este caso, sólo hay la salida out0. Para añadir el nodo, a diferencia de las entradas, hacemos botón derecho a sobre la señal, pero ahora elegimos “Add to – Wave – Selected Signals”. Tenemos así el entorno ya preparado, como podéis ver a la figura siguiente:

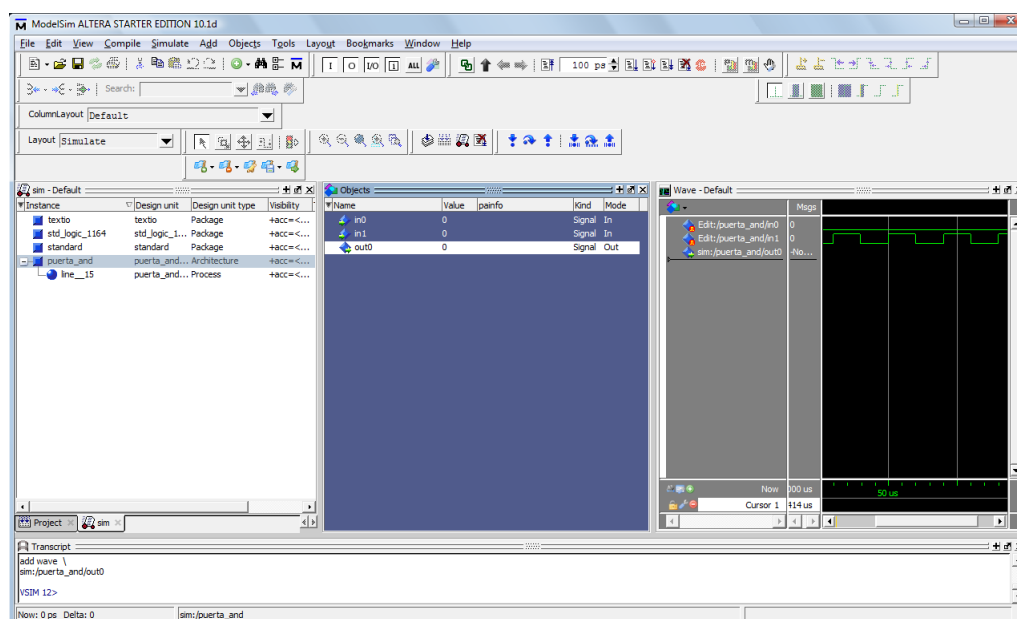


Figura 19: Añadiendo señales de salida

En la ventana anterior vemos como se ha añadido la señal out0

Con esto ya tenemos entradas y salidas preparadas para simular. Fijémonos que hemos sustituido el proceso del banco de pruebas.

2.3.5. Simulación y visualización de resultados

Ahora sólo hay que hacer – como en el supuesto de que vimos de ModelSim – Simulate – Run – Run –All. Vigilad si tenéis alguna ventana “maximizada” y no estáis dentro de la ventana principal de ModelSim pues el menú puede no aparecer. Si estáis en ModelSim y ejecutáis la opción anterior, podemos ver que se actualiza la ventana Wave, y muestra el resultado de la simulación. Cómo hicimos en el tutorial de ModelSim, jugad con el zoom para ver el rango de tiempo que queréis:

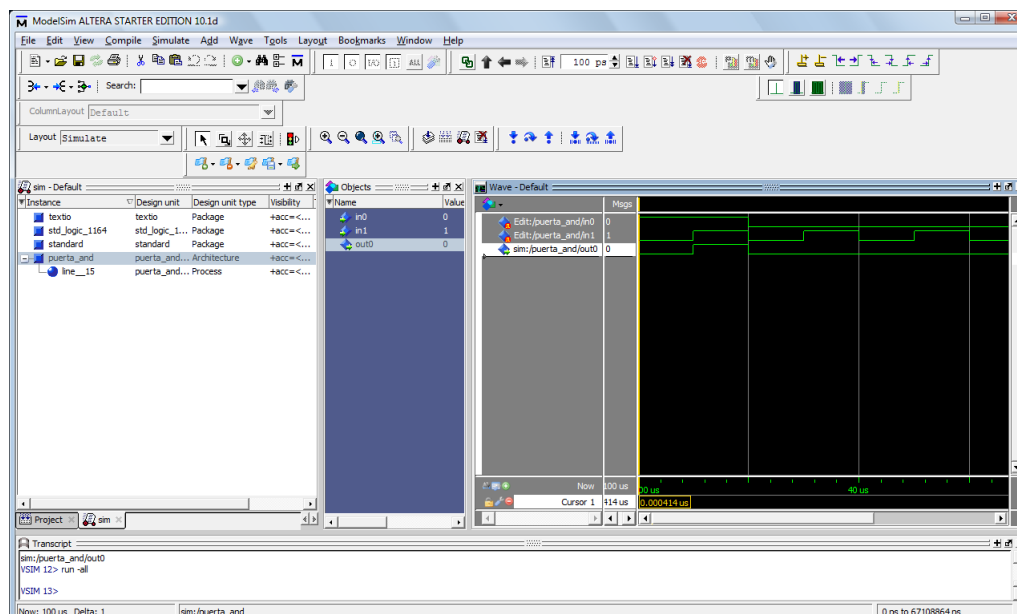
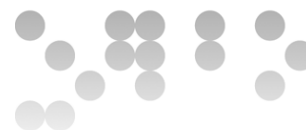


Figura 20: Resultado de la simulación

Sólo queda ahora comprobar el resultado, y si el resultado no es satisfactorio, volver a repetir el proceso. De forma simple, lo podéis hacer yendo a “Simulate-Restart” (cosa que ‘reseteará’ la forma de onda de salida), modificar las entradas, y volver a simular. De lo contrario, si queréis tocar el código, haced “Simulate-End simulation”, volved al entorno Quartus II y repetid el proceso.

3. Resumen

Si ahora repasamos lo que hemos visto, conceptualmente el proceso es simple:

- 0.- Preparación del entorno para que utilice ModelSim como simulador. Sólo hay que hacerlo la primera vez.
- 1.- Edición de un fichero vhd con el código deseado.
- 2.- Creación de un nuevo proyecto donde incluiremos el código anterior.
- 3.- Compilación del código y validación correcta.
- 4.- Invocación del simulador.
- 5.- Incluir y editar las formas de onda de las entradas.
- 6.- Añadir las salidas.



- 7.- Simular el circuito y comprobar el resultado.
- 8.- Caso que el comportamiento no sea el esperado, rehacer el código y repetir los puntos 3 al 7.