

**Grado en Ingeniería de Tecnologías de Telecomunicación.
Escuela Técnica Superior de Ingeniería Industrial y de Telecomunicación.
Electrónica Digital I.**

Práctica nº 4. Herramienta de diseño digital Quartus II.

El objetivo de esta práctica es aprender el manejo de la herramienta de diseño de circuitos Quartus II (o Quartus Prime, desde la versión 15.0). Esta herramienta permite el diseño de un circuito digital a partir de una descripción gráfica del circuito o a partir de una descripción HDL (Verilog o VHDL) hasta la implementación física del diseño en un dispositivo comercial programable del fabricante Altera. En el laboratorio se dispone de la tarjeta de diseño ALTERA DE2 que contiene un dispositivo Cyclone II EP2C35F672C6N, que será utilizado en prácticas posteriores.

Los pasos necesarios para desarrollar un circuito en Quartus II podrían ser:

- Editar la descripción del circuito.
- Crear un proyecto para el circuito. Un proyecto contiene el fichero o los ficheros de descripción, de estímulos para simulación, etc.
- Compilar la descripción del circuito hasta eliminar los errores de descripción.
- Simular el comportamiento del circuito para comprobar que funciona correctamente.
- Asignar los entradas y salidas del circuito a pines del dispositivo. Volver a compilar y a simular.
- Realizar análisis de las prestaciones del circuito para comprobar que las variables físicas (tiempos de propagación, potencia, frecuencia máxima) están dentro del rango requerido por el diseñador.
- Programar el circuito en un dispositivo electrónico.
- Verificar que el circuito funciona correctamente.

En esta práctica se realizarán los pasos de diseño consistentes en la edición del circuito, compilación, análisis y simulación. La programación del dispositivo y su verificación será realizada en prácticas posteriores.

La versión Quartus II Web Edition es gratuita. En el laboratorio se utilizará la versión 9.1, que ya no se puede descargar directamente desde la página web del fabricante, aunque disponemos del instalador en el laboratorio. La última versión (que se llama Quartus Prime) y las versiones anteriores de este software desde la versión 13.0sp1 se pueden descargar desde la página web de Intel (se requiere registrarse como usuario) <https://fpgasoftware.intel.com/> (seleccionando en ella la versión deseada).

Quartus II es una herramienta bastante compleja que contiene un número alto de opciones. En esta práctica se utilizarán de forma sencilla los aspectos básicos de la herramienta sin profundizar mucho en su manejo. Debido a su complejidad no se hará un manual completo de

la herramienta, sino que se desarrollarán durante esta práctica los aspectos más relevantes de la herramienta. Más información se puede encontrar en algunos de los libros recomendados en la bibliografía básica de la asignatura, en la ayuda interactiva de Quartus II (que incluye tutoriales interactivos y en pdf) o, para las últimas versiones, en la página web de Intel:

<https://www.intel.com/content/www/us/en/programmable/products/design-software/fpga-design/quartus-prime/support.html>

La tarjeta DE2, que se usará en la práctica 6, incluye un CD con el programa Quartus II (versión 6.0), un CD con la descripción del microprocesador Nios II que se puede cargar en el dispositivo programable y un CD DE2 con manuales, hojas de características, esquemas, ejercicios guiados (de la tarjeta DE2 y de Quartus II), prácticas propuestas, demostraciones avanzadas, etc. En el caso de que no se domine la herramienta Quartus II se puede practicar con los tutoriales que se encuentran en el CD DE2 (directorio *DE2_tutorial*):

- *tut_quartus_intro_schem* para diseño mediante esquemáticos.
- *tut_quartus_intro_vhdl* para diseño mediante VHDL.
- *tut_quartus_simulation_vhdl* para simulación de una descripción VHDL.

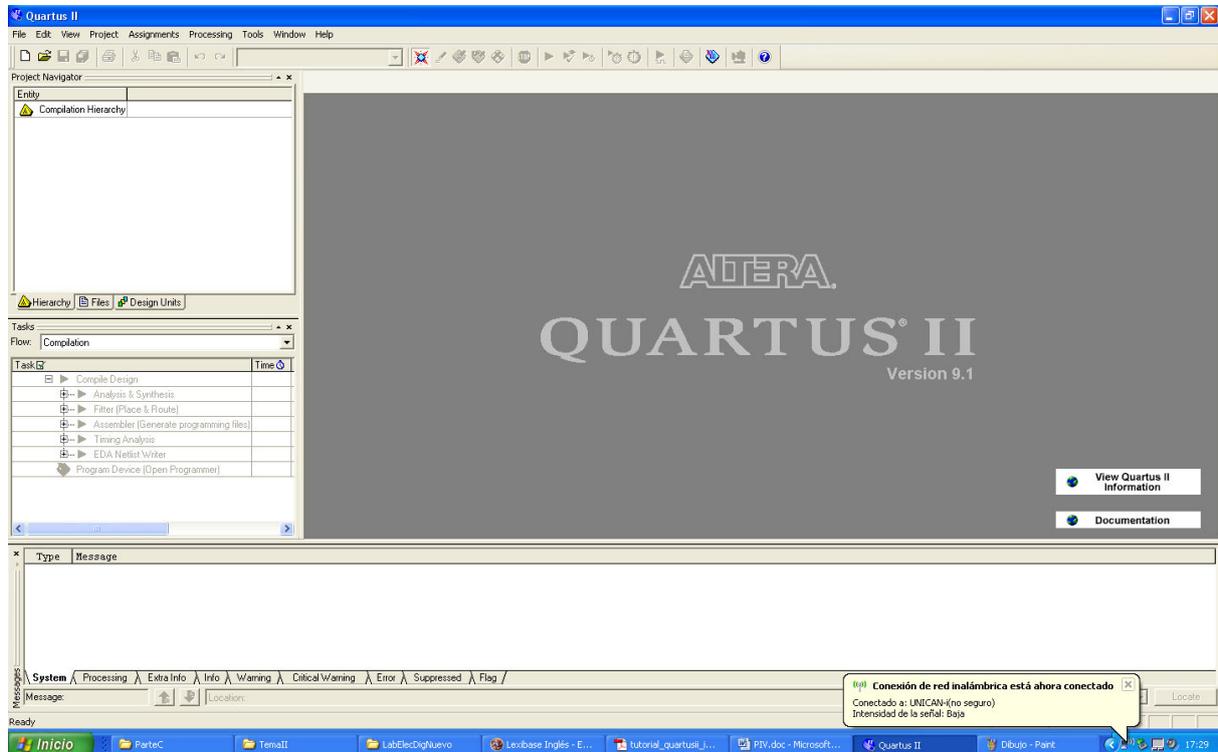
Todo el trabajo de la práctica debe realizarse en el directorio (o carpeta) Pr4 (o similar), que debe crearse en el directorio de trabajo de cada alumno. En la carpeta Pr4 debe utilizarse una subcarpeta para cada apartado y trabajar en ella, ya que Quartus II genera muchos ficheros para manejar un proyecto.

4.1. Descripción básica de Quartus II.

Para ejecutar Quartus II v9.1 basta con hacer un doble-click sobre el icono . La ejecución del programa genera una primera pantalla. En la línea superior de la pantalla aparece un menú en línea que permite el acceso a otros submenús del programa, pulsando con el botón izquierdo del ratón sobre cada palabra. Situada en la fila inmediatamente inferior aparece un menú de iconos o barra de herramientas, para activar con facilidad las opciones más importantes del simulador. Para activar una de estas opciones hay que pulsar sobre el icono correspondiente con el botón izquierdo del ratón. En ambas líneas hay una opción (*Help* o ) que permite acceder a un sistema de ayuda interactiva sobre las opciones de la herramienta.

En el resto de la pantalla, por defecto, está ocupada por ventanas. En la parte inferior la ventana *Message*, que muestra los mensajes generados cuando la herramienta está realizando o ha realizado alguna operación como por ejemplo la compilación de un diseño. Arriba a la izquierda se encuentra la ventana *Project Navigator*, que muestra en sus pestañas las diferentes partes que configuran un proyecto. Debajo de esta se encuentra la ventana *Tasks*, que indica cómo evoluciona la ejecución de un comando. Por último, la parte derecha superior de la ventana queda libre, en esta zona aparecerán las ventanas que se vayan generando durante el diseño del circuito: los ficheros gráficos o de texto para la edición del circuito, los ficheros de formas de

onda para la simulación y los resultados de las compilaciones o de las simulaciones. Se pueden modificar las ventanas que aparecen en pantalla mediante el comando *Utility Windows* del menú **View**.



El menú en línea contiene diversos menús expandibles:

- **File.** Permite crear ficheros nuevos o abrir ficheros existentes, crear o abrir proyectos, salvar ficheros, imprimir, crear ficheros de un tipo en base a otros, etc.
- **Edit.** Contiene comandos típicos de la edición de ficheros (de texto, de gráficos, de formas de onda, etc). Algunos comandos son comunes (*undo*, *redo*, *cut*, *copy*, *paste*, *find*, etc), mientras que otros dependen del tipo de editor activo en ese momento, por ejemplo, *Insert File*, *Insert Bookmark*, etc, para ficheros de texto; *Flip*, *Rotate*, etc para descripciones gráficas, *Insert Node*, *End Time*, *Time Bar*, etc para formas de onda.
- **View.** Contiene comandos que controlan lo que se muestra en pantalla. Algunos de los comandos son comunes y otros varían según la ventana activa. Por ejemplo, comunes son *Utility Windows* que abre un submenú para activar distintas ventanas de control en pantalla, y *Full Screen* que sitúa Quartus II en modo pantalla completa. Si se tiene activo un editor de textos en el menú aparecen unos comandos de edición; si se tiene abierto un editor de esquemáticos se dispone de comandos de *Zoom*, y de comandos para mostrar o no datos en pantalla; si la ventana activa es de formas de onda también hay comandos de *Zoom*, para comparar formas de onda, para indicar cuando pueden cambiar las señales de valor al editar o como se muestra el tiempo, etc.

- **Project.** Contiene comandos para realizar el control de un proyecto. Por ejemplo, *Add/Remove Files in Project*, abre una ventana donde se muestran los ficheros incluidos en el proyecto, y donde se puede añadir un nuevo fichero o eliminar alguno de los contenidos.
- **Assignments.** Contiene comandos que permiten realizar la asignación de determinadas características al circuito, como el dispositivo que se utiliza en la implementación (*Device*), sobre que pins del dispositivo se sitúan las entradas y las salidas (*Pins* o *Pin Planner*), restricciones temporales al circuito (*Timing Analysis Settings*) o acceso a un menú con todas las características (*Settings*).
- **Processing.** Permite acceder a los comandos que realizan la compilación (*Compiler Tool*), la simulación (*Simulation Tool*) y el análisis de las características del circuito: temporales (*Classic Timing Analyzer Tool*) y de potencia disipada (*PowerPlay Power Analyzer Tool*). Además, se puede realizar la compilación en un único paso (*Start Compilation*) o en varios pasos (menú *Start*), y se puede realizar la simulación (*Start Simulation*) y observar los resultados de la compilación (*Compilation Report*) y de la simulación (*Simulation Report*), que aparecen por defecto al finalizar la correspondiente operación.
- **Tools.** Contiene comandos de llamada a tareas de la herramienta, por ejemplo, *Programmer* que permite cargar los resultados de la compilación en un dispositivo físico, o *TimeQuest Timing Analyzer*, que invoca a un analizador temporal que sustituye al *Classic*, o *Chip Planner* que permite observar el interior de un dispositivo y visualizar su contenido o modificarle y cargar directamente un circuito, etc. El comando *Options* abre una ventana donde se pueden cambiar un gran número de características básicas de algunas herramientas de Quartus II, así como colores, fuentes de letras, navegador para conectarte a Altera, ficheros de licencias (*License Setup*) si fuesen necesarios, etc.
- **Window.** Muestra las ventanas activas en la herramienta y permite seleccionar una de ellas. También permite controlar como se muestran las diferentes ventanas abiertas en Quartus II (*Cascade*, *Tile Horizontally*, *Tile Vertically*), dentro del espacio de Quartus II, o como una ventana independiente (*Detach Window*).
- **Help.** Activa la ventana de ayuda, donde se puede encontrar información de los distintos comandos de Quartus II, bien sea por capítulos (*Contents*) o por un índice (*Index*), con posibilidad de buscar por palabras claves (*Search*). También se encuentran en este menú tutoriales de la herramienta (*Tutorial*) en forma interactiva o en pdf (*Pdf Tutorial*).

La línea de iconos que se muestra en pantalla hace llamadas a algunos de los comandos más utilizados normalmente. Situando el ratón sobre el icono aparece un mensaje con el comando correspondiente, al igual que en el menú aparece junto al comando el icono correspondiente. Por ejemplo el icono  realiza una compilación completa, el icono  realiza una simulación, y el icono  aborta la compilación o simulación que se esté realizando.

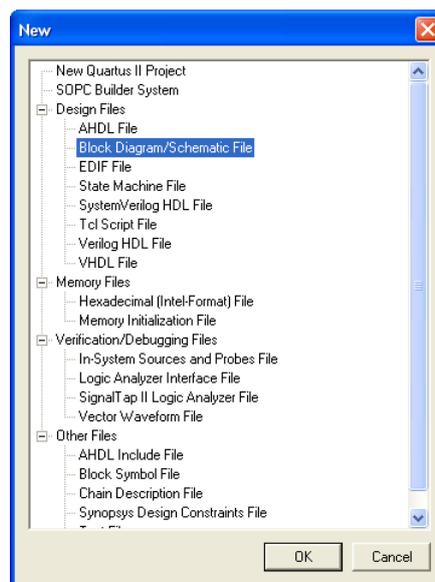
4.2. Diseño de un circuito mediante descripción de esquemáticos.

El primer diseño sobre Quartus II se hará utilizando una descripción esquemática del tipo usada en Circuit Maker. Para este apartado todos los ficheros deben guardarse en la carpeta Mux2graf dentro de la carpeta Pr4 de la carpeta de trabajo de cada alumno. El circuito a diseñar será un multiplexor de dos entradas, un circuito de dos entradas de datos I0 e I1, una entrada de selección S y una salida Z. La salida Z toma el valor de la entrada I0 si S es 0 y el valor de la entrada I1 si S es 1. Planteando este problema y resolviéndolo en una función lógica nos queda que:

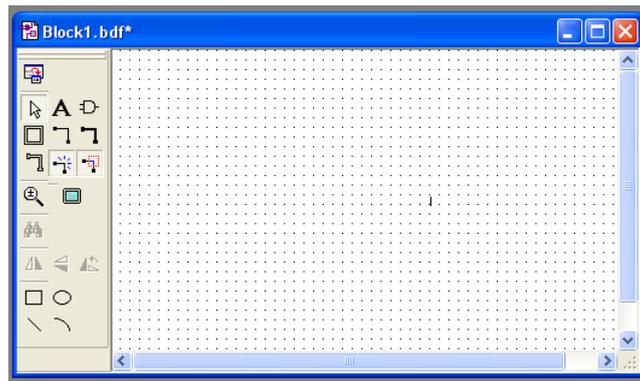
$$Z = I0 \cdot \bar{S} + I1 \cdot S$$

que se puede implementar con un circuito con 1 puerta NOT, 2 puertas AND de dos entradas y 1 puerta OR de 2 entradas. Para describir y simular este circuito digital en Quartus II hay que seguir estos pasos.

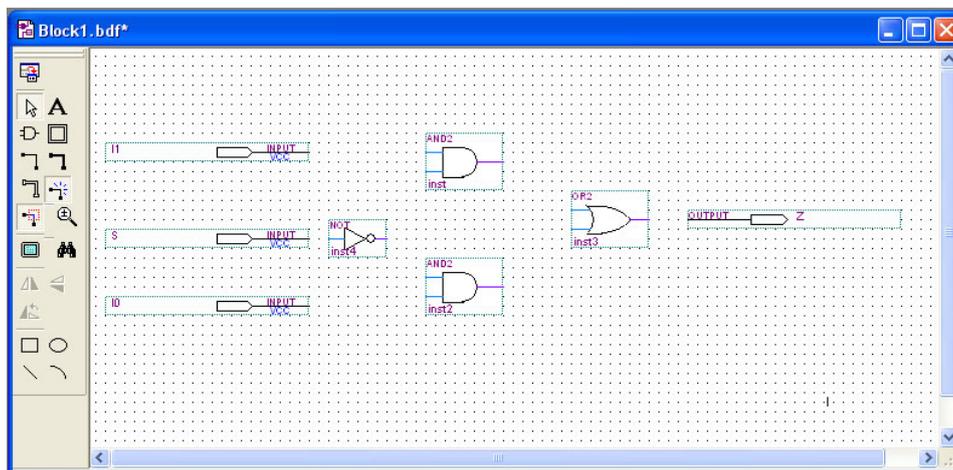
- Arrancar Quartus II haciendo un doble-click sobre el icono  de la herramienta Quartus II 9.1 Web Edition.
- Crear un nuevo fichero desde el comando *New* del menú **File** (o pulsar en el icono ). Este comando abre una ventana de selección del tipo de fichero que se desea crear, elegir el tipo *Block Diagram/Schematic File* y pulsar en OK. Otros tipos que se utilizarán en esta práctica serán por ejemplo el *VHDL File* y el *Vector Waveform File*. Cuando se abre un fichero ya existente con el comando *Open* del menú **File** (o el icono ) , para seleccionar el fichero hay que indicar el tipo de fichero (*Device Design Files, Project Files, Graphic Files, etc*) en el campo *Tipo* .:



Aparece una ventana para la edición de esquemáticos, con iconos para introducir dispositivos, rotarles, conectarles, hacer Zoom o editar figuras (rectángulos, círculos, líneas), etc.

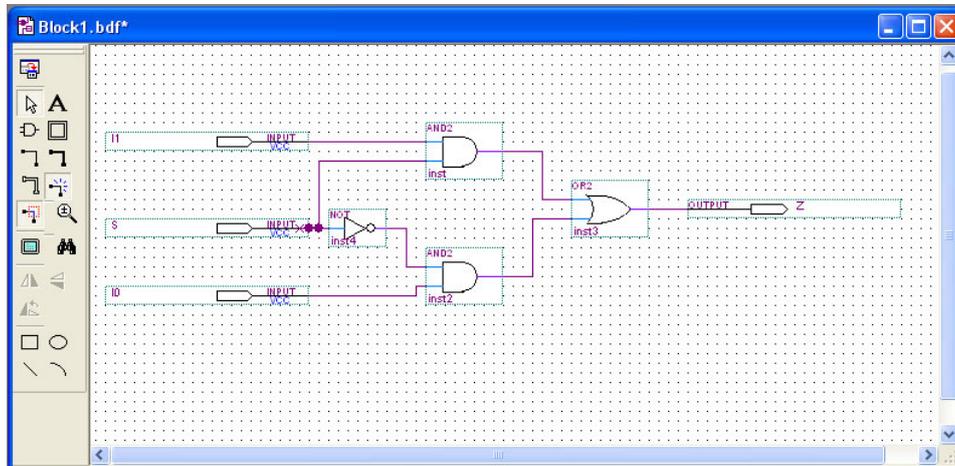


• Pulsar en el icono  de la ventana de esquemáticos. Sobre el menú desplegable pulsar en  sobre el catálogo principal, y desplegar igualmente los submenús *primitives* y *logic*. Sobre el catálogo desplegado sucesivamente localizar los elementos not, and2 y or2, pulsar en OK y situar los elementos en la pantalla de esquemáticos (pulsar ESC al finalizar con cada elemento cuando el modo *Repeat-insert mode* está activado). Con el icono  activo, estos elementos se pueden seleccionar (pulsando sobre uno, o una zona completa pulsando y desplazando el ratón) y moverlos con el ratón, o rotarlos con los iconos   , o borrarlos pulsando la tecla **Supr.** Mover los elementos en la ventana para que queden ordenados y se puedan conectar bien. Volver a pulsar en los  para desplegar ahora los submenús *primitives* y *pin*. Seleccionar sucesivamente los elementos input y output, y situar 3 inputs (para las entradas I0, I1, S) y 1 output (para la salida Z). Mover los elementos en la ventana para que queden ordenados y se puedan conectar bien. Haced doble-click sobre cada input y output, y en la ventana que se abra introducir el nombre de la entrada o salida en el campo *pin_name(s)*.



• Pulsar en el icono  de la ventana de esquemáticos, con lo que se entra en modo conexión. En este modo al pulsar con el ratón en un punto de la ventana y, sin soltar, desplazar el ratón, al soltarlo en otro punto de la ventana se genera una conexión entre esos dos puntos, formada por una línea horizontal y/o una línea vertical. Realizar las conexiones entre las entradas y salidas de los elementos del circuito intentando asegurarse de que están bien hechas: no hay cables que se quedan cortos ni que, al ser largos, realizan conexiones no deseadas. Si hace falta usar el Zoom (pulsar en el icono , al pulsar con el ratón en la ventana aumenta la resolución,

con la tecla mayúsculas pulsada disminuye). Si se desea cambiar la forma de un cable, moverlo o borrarlo hay que pulsar el icono  y seleccionar el cable, y moverlo con el ratón o pulsar la tecla **Supr** para eliminarlo.



- Una vez editado hay que guardar el diseño con el comando *Save As* del menú **File** (o el icono ). Guardar el fichero de nombre Mux2graf.bdf en la carpeta de nombre Mux2graf de la capeta Pr4 del directorio de trabajo de cada alumno. El nombre del fichero puede ser distinto del directorio, pero usar los mismos nombres hace que el proceso sea más automático. Al guardar el fichero, Quartus II pregunta si se quiere realizar un proyecto asociado a este fichero. Un proyecto maneja todos los ficheros asociados a un diseño, ya que los diseños complejos necesitan usar varios ficheros de distintos tipos, librerías, características distintas de síntesis, simulación, análisis y restricciones temporales, asignación de pines, etc; los diseños sencillos normalmente solo requieren un fichero de descripción y otro de simulación.

Pulsar en *Si*. Si se pulsase en *No* el proyecto debería generarse desde el comando *New Project Wizard* del menú **File**. El proceso sería parecido pero los ficheros a incluir en el proyecto deberían indicarse explícitamente.

- La creación del proyecto en realiza en diferentes pasos:

- La primera ventana que aparece es únicamente de información. Pulsar en *Next*.
- La segunda ventana indica el directorio de trabajo del proyecto que debe coincidir con el directorio de trabajo del alumno, el nombre asociado al proyecto, y la entidad de más alto nivel en la jerarquía del proyecto. Por defecto, todo se fija a Mux2graf aunque podría variarse, bien desde teclado o buscando en otras carpetas al pulsar en el icono . Pulsar en *Next*.
- La tercera ventana permite añadir ficheros al proyecto. Esta ventana corresponde al comando *Add/Remove Files in Project* del menú **Project** o a las pestañas *Files* y *Libraries* del comando *Settings* del menú **Assignments**. Por defecto ya se ha añadido el fichero del esquemático. Pulsar en *Next*.
- La cuarta ventana corresponde a la selección del dispositivo ALTERA donde se va a implementar el diseño. Esta ventana corresponde al comando *Device* del menú

Assignments. Seleccionar en la pestaña *Family*, la familia Cyclone II y en la ventana *Available Devices*: el dispositivo EP2C35F672C6. Pulsar en *Next*.

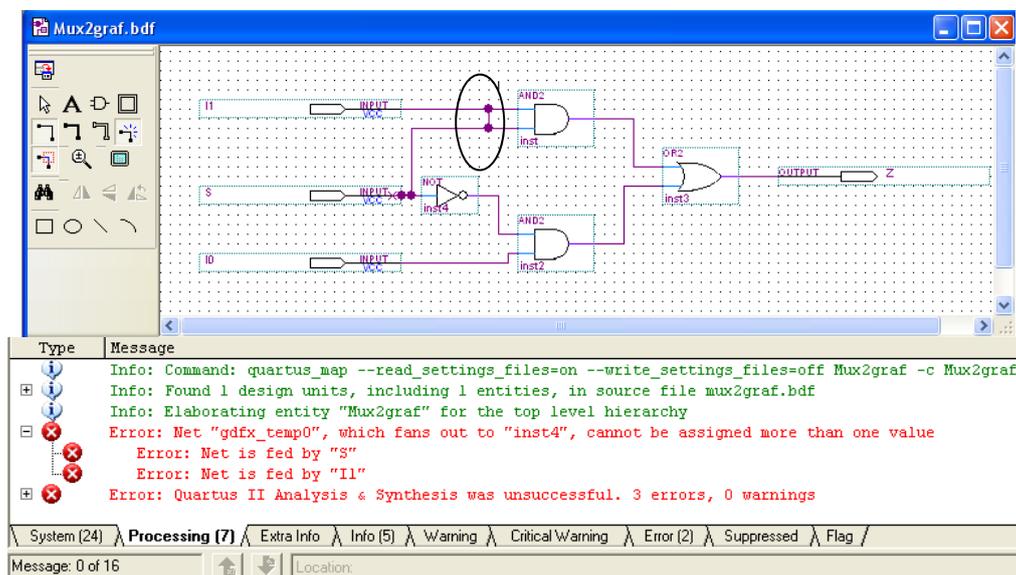
- La quinta ventana permite seleccionar otras herramientas distintas de las de Quartus II para realizar la síntesis lógica del circuito, la simulación o el análisis temporal. Esta ventana corresponde al comando *EDA Tools Settings* del menú **Assignments**. Pulsar en *Next*.
- La sexta ventana es el resumen del proyecto. Pulsar en *Finish*.

En la ventana *Project Navigator* de Quartus II en la pestaña *Hierarchy* deben aparecer el dispositivo seleccionado, y la jerarquía del proyecto, en la pestaña *Files* los ficheros de descripción de los que consta el proyecto y en la pestaña *Design Units* las unidades asociadas al proyecto que estará vacío hasta después de compilar.

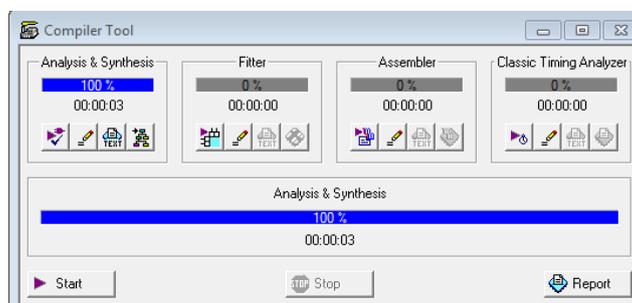
Desde el menú **File** se pueden crear proyectos (*New Project Wizard*), guardarlos (*Save Project*), abrirlos (*Open Project*) o cerrarlos (*Close Project*).

• Una vez creado el proyecto el siguiente paso es compilar o sintetizar el diseño. En un primer análisis del diseño nos puede interesar solo conocer si la descripción del circuito es correcta. Para ello se debe hacer una compilación basado en dos pasos:

- Activar el comando *Compiler Tool* del menú **Processing**. Pulsar en  para realizar el Analysis & Synthesis. En la ventana *Message* aparecen una serie de mensajes sobre las tareas que está realizando la herramienta. Si existe algún error en el diagrama (líneas mal conectadas o cortocircuitos) aparece un mensaje de error en color rojo. Pulsando en el mensaje de error se marca en la ventana del diagrama el punto erróneo. A veces para entender mejor el error hay que pulsar en el cuadro  asociado a la línea, con lo que aparecen varios mensajes asociados al error. También pueden aparecer mensajes de aviso en azul, no son errores, pero pueden indicar algo incorrecto o no bien definido en el circuito.



- Si la compilación acaba sin error, realizar el comando *Generate Funcional Simulation Netlist* del menú **Processing**. Esto permite realizar únicamente una simulación funcional de la descripción lógica del circuito, en lugar de la simulación temporal que se realiza sobre el circuito (se añaden tiempos de propagación de los dispositivos utilizados).

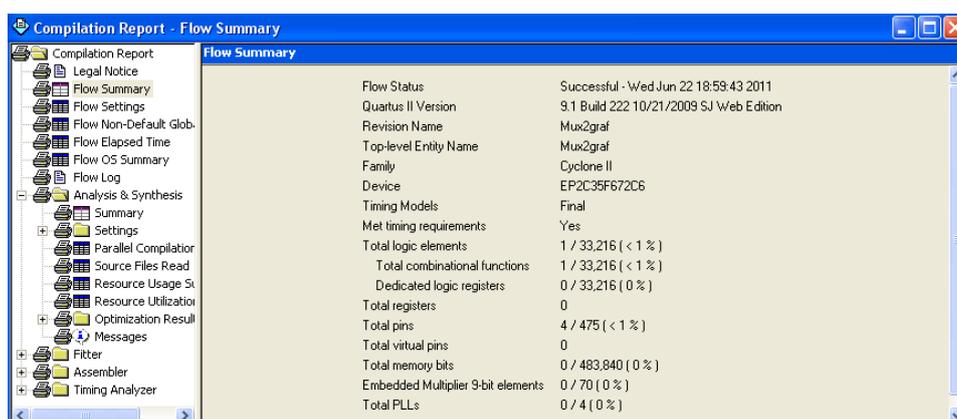


La compilación puede tener hasta cuatro tareas, que se realizan en el siguiente orden:

- Analysis & Synthesis. Comprueba que la descripción no tiene errores y realiza la síntesis lógica del circuito.
- Fitter. Fija la descripción lógica en determinadas celdas de dispositivo ALTERA y decide como interconectarlas.
- Assembler. Genera la información para cargar el diseño en el dispositivo.
- Timing Analyzer. Estudia las características temporales del circuito resultante.

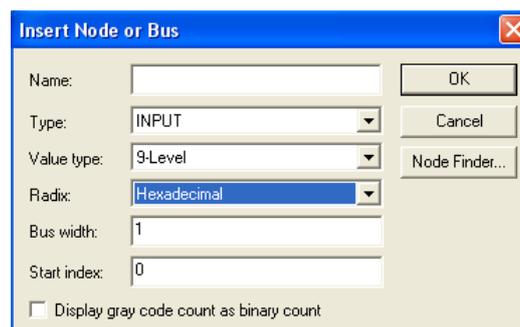
Se puede compilar de dos formas:

- En un solo paso. Pulsar en el icono . En la ventana *Tasks* se muestra la operación que se está realizando, el porcentaje de la operación realizado y el tiempo de ejecución transcurrido. Al finalizar aparece en pantalla una ventana con los resultados de la compilación (*Compilation Report – Flow Summary*), con el número de celdas utilizadas del dispositivo, número de pines, etc. En el margen izquierdo aparecen otros resultados de la compilación que se pueden observar en pantalla pulsando en los iconos . Algunos de estos resultados pueden guardarse usando el comando *Save Current Report Section As* del menú **File**.



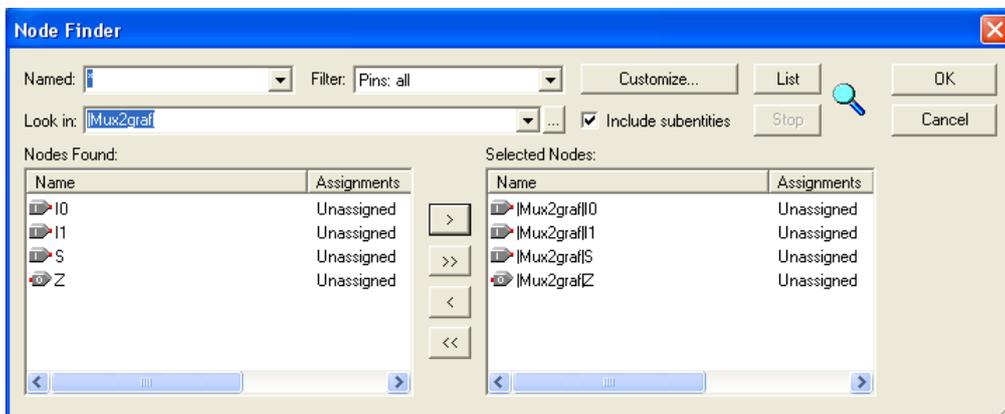
- Por pasos. Activar el comando *Compiler Tool* del menú **Processing**. Abre una ventana que muestra los cuatro pasos de la compilación. Pulsando en Start se realizan los cuatro pasos. Para hacer la compilación paso por paso hay que pulsar por orden en  (Analysis & Synthesis),  (Fitter), y  (Assembler) o  (Timing Analyzer). Los iconos  muestran las características (Settings) en cada paso, los iconos  muestran un fichero de texto con el informe del resultado de cada paso. El resto de los iconos abre una aplicación para cada operación (por ejemplo, el programador para Assembler). Pulsando en *Report* aparece la misma ventana de resultados de compilación que en la compilación en un único paso.
- Una vez compilado el diseño se debe comprobar si funciona bien mediante simulación. Para ello primero hay que editar los estímulos que aplicar al circuito. Esta versión de Quartus II dispone de una herramienta de simulación. Algunas versiones posteriores de Quartus II no disponen de este simulador, sino que hay que descargarse además el simulador ModelSim. Los estímulos para la simulación pueden hacerse desde una ventana de formas de onda. En este apartado de la práctica se editarán en la ventana todas las entradas de la tabla de verdad del problema para realizar una simulación exhaustiva. Los pasos a seguir serán:

- Aplicar el comando *New* del menú **File**. Pulsar sobre *Vector/Waveform File* para seleccionar el tipo. Pulsar en OK. Se abre la ventana de formas de onda de nombre Waveform1.vwf.
- Utilizar el comando *End Time* del menú **Edit**. Este comando indica el tiempo final de la simulación. Si se van a introducir todas las combinaciones distintas en 3 entradas (8 combinaciones) en un tiempo de 50 ns (para evitar problemas temporales posteriores), entonces se necesitan 400 ns. Situar ese valor en el campo *Time*. El campo *Extension Pattern* se utiliza para indicar los valores a los que se fijan las señales al incrementar el *End Time*.
- Utilizar el comando *Insert Node or Bus* de la pestaña *Insert* del menú **Edit**. Este comando permite fijar en la ventana de ondas nudos a los que fijar valor (entradas) o en los que observar valor (salidas o nudos internos). Se puede introducir un nombre de un nudo (1 bit) o bus (asociado a varios bits), y sus características: tipo (*Type*: entrada, salida, etc), formato (*Radix*: binario, hexadecimal, decimal sin signo o con signo, etc), anchura o número de bits (*Bus width*), pero lo más adecuado es realizar una búsqueda automática pulsando en *Node Finder*.

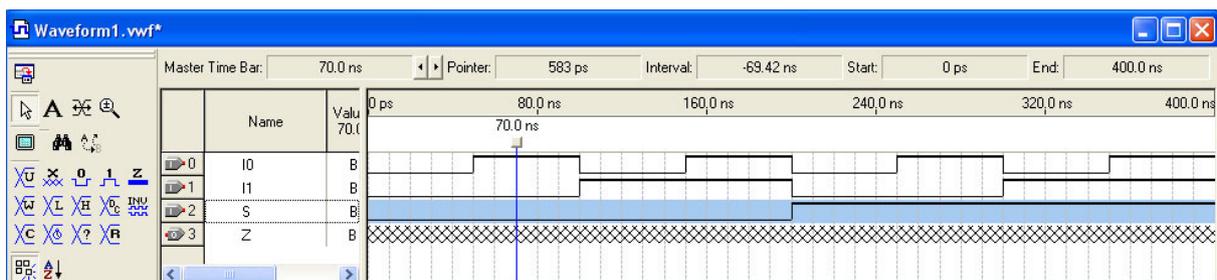


Situar en la pestaña *Filter*: el valor *Pins: all* para buscar solo nudos de entrada y salida (otras opciones son posible que permiten acceder a nudos internos), pulsar en *List* para que aparezcan en *Node Found* los nudos disponibles, seleccionar en con el ratón uno o varios nudos pulsando sobre ellos (con la tecla mayúsculas apretada se seleccionan varios) y pulsar en . En la ventana *Selected Nodes*: aparecen los nudos que se fijarán en la ventana de formas de onda. Pulsar en OK en la ventana *Node Finder* y, en la ventana *Insert Node or Bus* situar el campo *Radix*: a Binary y pulsar en OK.

El orden de los puertos en la ventana se puede cambiar pulsando con el ratón sobre el icono de conector ( entrada,  salida) que aparece al lado del nombre y, sin soltar, desplazar el ratón hacia arriba o hacia abajo en la ventana. Para eliminar un nudo basta con seleccionar su nombre con el ratón, y pulsar la tecla **Supr**. Se pueden cambiar las propiedades de un puerto seleccionándolo y pulsando sobre él con el botón derecho del ratón. Varios puertos pueden agruparse (y desagruparse después) seleccionándolos y usando el comando *Group* (y *Ungroup*) del submenú *Grouping* del menú **Edit**.



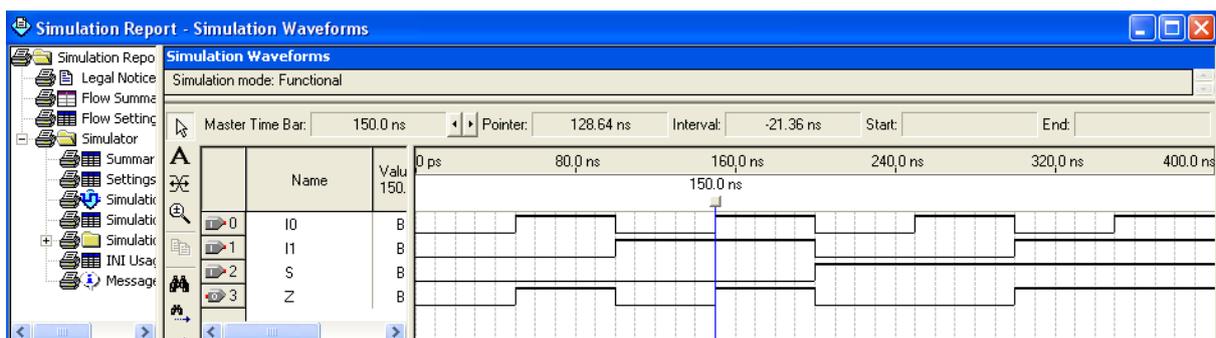
- Ahora se deben generar los estímulos para las entradas. Una forma rápida de generar la tabla de verdad es introducir señales de reloj en los estímulos, son señales periódicas de un periodo fijo. En la ventana de formas de onda pulsar sobre I0, de forma que quede toda la línea seleccionada. Pulsar sobre el icono , y en la ventana que aparece situar el campo *Period*: a 100 ns. Realizar la misma operación sobre I1 con *Period*: a 200 ns, y con S con *Period*: a 400 ns. Aplicar el comando *Fit in Window* del menú **View** para ver los estímulos en todo el rango de tiempo. Cada 50 ns aparece en las entradas uno de los valores de la tabla de verdad. Guardar el fichero con el comando *Save as* del menú **File** (o el icono ) con el nombre Mux2graf.vwf, de esta forma el fichero queda incorporado al proyecto.



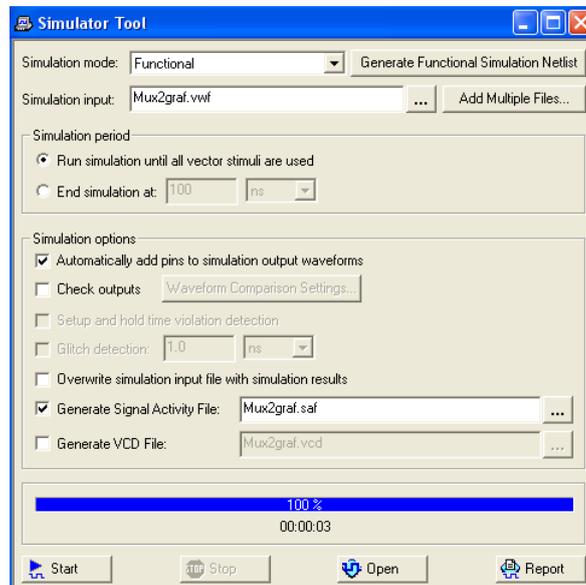
Existen otras formas de introducir estímulos en una entrada. Se puede hacer con los iconos de valores: seleccionar con el ratón una zona temporal de una señal (pulsar y, sin soltar desplazar el ratón hasta soltar, la zona se marca en azul) e introducir un valor típico del tipo `std_logic` de VHDL: 0 (0), 1 (1), Z (Z) o X (X), H (H), L (L), W (W), U (U), don't care (Xc), respectivamente. Sobre la zona seleccionada, para una señal de uno o varios bits, también se puede aplicar los iconos `INV`: complementa el contenido de las señales, `Xc`: genera una señal periódica, `X+`: genera un valor que se incrementa en el valor fijado en el campo *Increment by*: cada tiempo indicado en el campo *Count every*: de la pestaña *Timing*, `X?`: carga en todo el rango el valor indicado en el campo *Numeric or named value*:, en el formato indicado en *Radix*:, `XE`: genera un valor aleatorio en función de la selección que se haga (intervalos fijos, intervalos aleatorios, en función del *grid* o rejilla).

Para hacer este tipo de edición pueden ayudar los comandos *Grid size* del menú **Edit**: el tiempo entre las marcas verticales de la rejilla que aparece en la ventana, y *Snap Grid* del menú **View** (icono `88`): la selección temporal se puede realizar únicamente entre las marcas de la rejilla. Sobre las señales también se pueden realizar operaciones de edición como copy-paste y se puede utilizar el zoom (icono `89`) para aumentar o disminuir la resolución en pantalla. Para medir tiempo se dispone de una barra Master Time Bar, en la ventana se indica la posición de esa barra, la posición del ratón (*Pointer*:) y la diferencia en tiempo entre ambas (*Interval*:); se pueden introducir más barras desde el comando *Insert Time Bar* del submenú *Time Bar* del menú **Edit**.

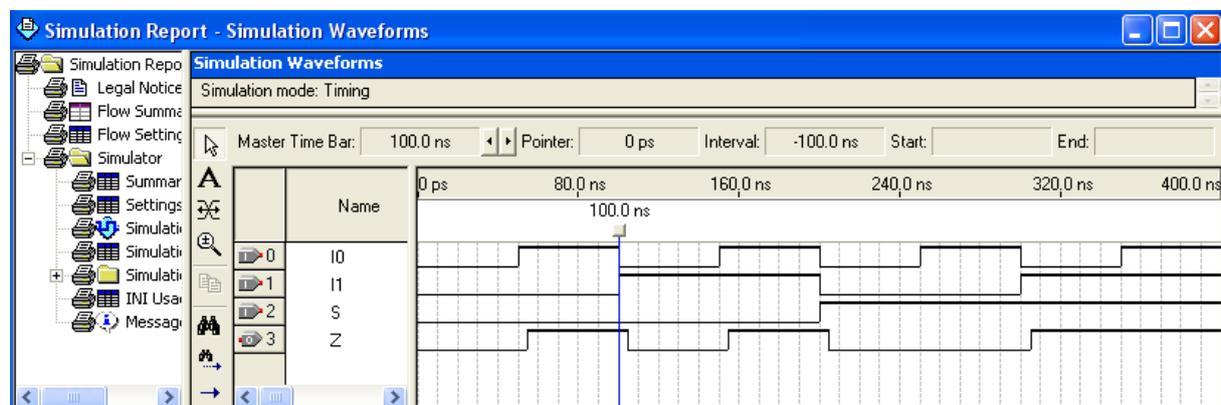
- Realizar la simulación. La forma más directa sería pulsando directamente sobre el icono `90`. Entonces se realizaría directamente una simulación temporal. Para ello se requiere haber hecho una compilación completa. Como estamos haciendo primero una simulación funcional se lo indicamos abriendo la herramienta de simulación mediante el comando *Simulator Tool* del menú **Processing** y fijando el campo *Simulation Mode* a valor *Functional*. Pulsar ahora en `91` o en *Start* y realizar la simulación. Si la simulación se realiza sin errores aparecerá una ventana *Simulation Report* con el resultado de la simulación, salvo cuando la ventana *Simulator Tool* está activa hay que pulsar en *Report* para activar la ventana de resultados. Con la ventana de resultados activa, realizar *Fit in Window* del menú **View** para visualizar todo el rango de tiempos. Comprobar sobre esa ventana que el circuito funciona correctamente: si S es 0 entonces Z es igual a I0, si S es 1 entonces Z es igual a I1, al ser simulación funcional los cambios en la salida se realizarán sin retraso con respecto a la entrada. Se pueden guardar los resultados desde el comando *Save Current Report Section as* del menú **File**.



El simulador tiene otras opciones: por ejemplo, permite comparar formas de onda, permite generar ficheros de actividad para evaluar el consumo de potencia, etc.



- Una vez comprobado que el circuito funciona bien se realiza un análisis más completo no solo de la descripción sino de su operación en el dispositivo ALTERA. Para ello, pulsar en el icono  para realizar una compilación completa. En la ventana Simulator Tool se fija el campo *Simulation Mode* a valor Timing, y se activa el campo *Generate Activity Signal*: con nombre de fichero Mux2graf.saf, para poder medir más adelante la potencia disipada, y se pulsa en el icono . Pulsar en *Report* y comprobar que el circuito opera correctamente como en la simulación anterior. Ahora se observa que las señales de salida están retrasadas con respecto a los cambios en las señales de entrada. Realizar un zoom sobre un cambio de la salida con el icono  activo, logrando más resolución fijar la Master Bar en el tiempo en el que cambia la entrada y situar el ratón en el punto en el que cambia la salida, en el campo Interval: aparece la diferencia de tiempo o tiempo de propagación.

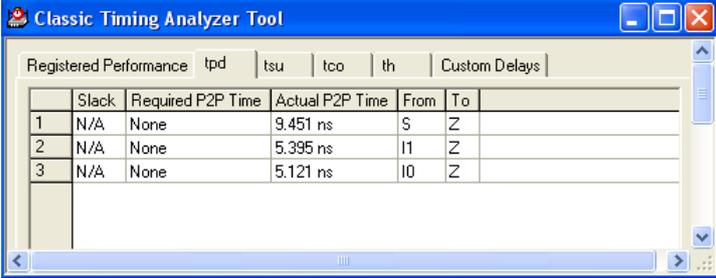


- Indicar el número de celdas del dispositivo utilizadas al sintetizar el circuito: campo *Total logic elements* en la ventana *Compilation Report*, que aparece al realizar un compilación con el botón  (o si se tiene la ventana *Compiler Tool* abierta al pulsar en su botón *Report*).

Además, usar los comandos *Netlist Viewer* del menú **Tools** para observar cuál ha sido el resultado de la síntesis del circuito. El formato *RTL Viewer* muestra el circuito con símbolos de formato RTL (Register-Transfer-Level), mientras que los *Technology Map Viewer* muestra el circuito como un bloque que se puede expandir haciendo doble-click sobre él a bloques más pequeños, o se pueden ver las ecuaciones que realizan pinchando sobre él con el ratón (& -> AND, # -> OR, \$ -> XOR, ! -> NOT).

- Por último, se va a realizar un análisis físico del circuito utilizando herramientas de Quartus II. Los análisis circuitales no serán del todo ajustados hasta que no se haga una implementación final en un dispositivo en el que se asignen las entradas y salidas del circuito a pines del dispositivo, pero si indica alguna de las opciones de Quartus II para analizar y refinar un diseño. Si no hay asignación explícita de pines, la asignación es realizada por el compilador.

- Hacer un análisis temporal del circuito encontrando las características temporales que limitan la velocidad de operación. En el caso de un circuito combinacional se puede comprobar el mayor tiempo de propagación mediante un analizador temporal. Ejecutar el comando *Classic Timing Analyzer Tool* del menú **Processing**. En realidad, este comando ya se ha ejecutado al realizar la compilación, así que ya se deberían tener los resultados sin necesitar pulsar en *Start*. Activar la pestaña *tpd* y en la columna Actual P2P aparecen los tiempos de propagación desde las entradas hasta la salida de mayor a menor. Medir el mayor tiempo de propagación.

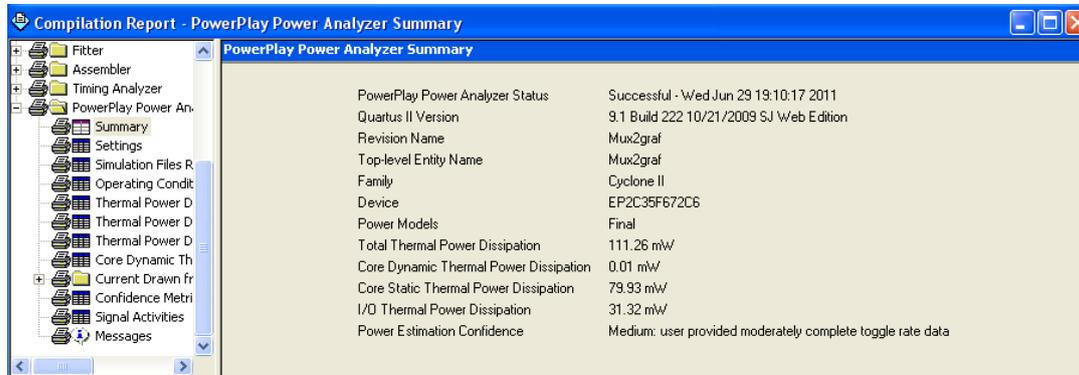


	Slack	Required P2P Time	Actual P2P Time	From	To
1	N/A	None	9.451 ns	S	Z
2	N/A	None	5.395 ns	I1	Z
3	N/A	None	5.121 ns	I0	Z

Quartus II tiene dos analizadores temporales: Classic y TimeQuest, para elegir uno de los dos por defecto al compilar ejecutar el comando *Timing Analysis Settings* del menú **Assignments** y activar una de las dos opciones. En nuestro caso debería ser Classic.

- También se puede obtener la potencia aproximada disipada por el circuito. Ejecutar el comando *PowerPlay Power Analysis Tool* del menú **Processing**. Activar el campo *Use input file(s) to initialize toggle rates and static probabilities during power analysis* y pulsar en *Add Power Input File(s)*. En la ventana que aparece pulsar en *Add*, en la nueva ventana seleccionar el *Input Type File* como *Signal Activity File*, pulsar en **...** del campo *File Name:* y seleccionar el fichero *Mux2graf.saf*. Pulsar en **OK** en todas las ventanas abiertas hasta volver la ventana *PowerPlay* donde se pulsa en *Start*. Al acabar el análisis pulsar en *Report* y medir la potencia total en *Total Thermal Power Dissipation*.

El análisis de potencia se puede realizar desde o partir de una ratio determinado de cambio en las señales, o de una simulación adecuada que genera ficheros de actividad de las señales (.saf o .vcd) a partir de los cuales se hace una estimación de la disipación en potencia.



- Se pueden mejorar los parámetros físicos mediante restricciones impuestas en la compilación. Por ejemplo, se quiere mejorar el mayor tiempo de propagación del circuito. Introducir el comando *Settings* del Menú **Assignments**; pulsar en de *Timing Analysis Settings* y seleccionar con el ratón *Classic Timing Analyzer Settings*. Sobre la ventana introducir en el campo tpd el valor 5 ns. Pulsar en OK y compilar mediante . Si no se pudiesen conseguir los requerimientos temporales la compilación debería dar un aviso. Al ejecutar *Classic Timing Analyzer Tool* del menú **Processing**, en la pestaña tpd los valores aparecen en rojo indicando que no se ha obtenido el requerimiento (*slack* negativo). Repetir el procedimiento con 5.5 ns, 6 ns, 6.5 ns, 7 ns, etc, hasta conseguir un circuito que cumpla los requerimientos y medir el mayor tiempo de propagación del circuito.

- Guardar y cerrar el proyecto mediante los comandos *Save Project* y *Close Project* del menú **File**.

4.3. Diseño de un circuito multiplexor de dos entradas mediante descripción VHDL.

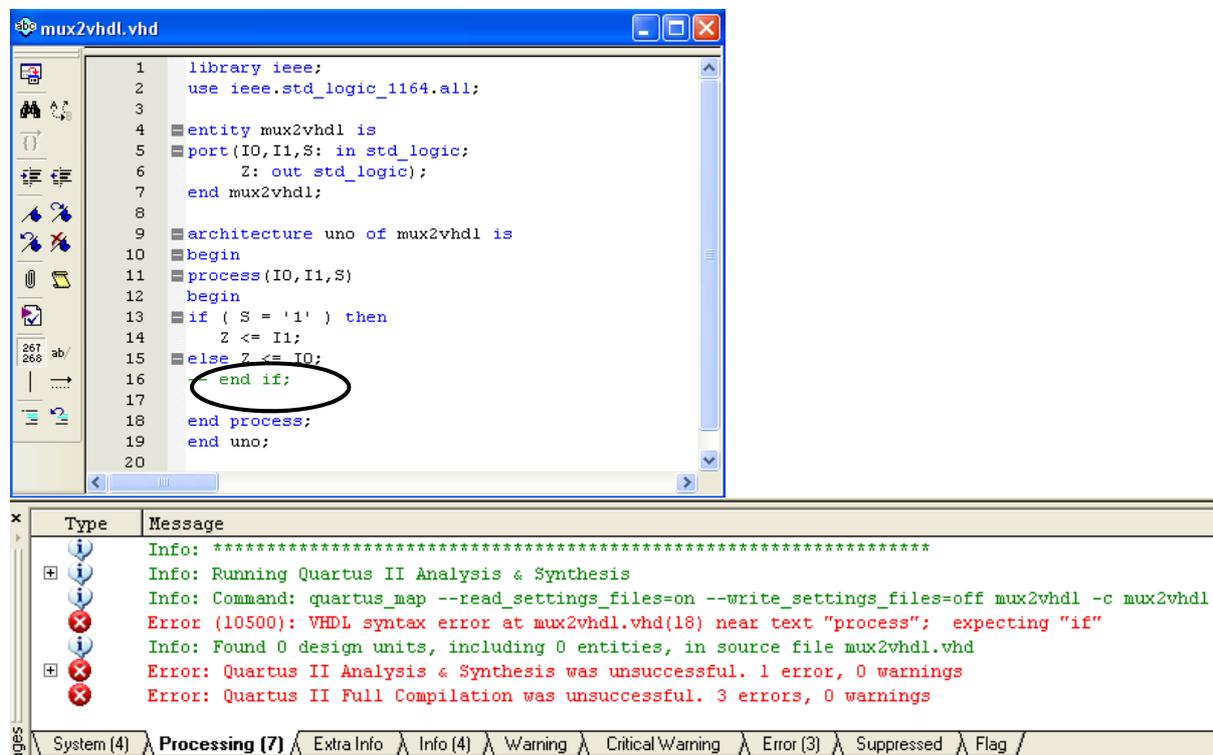
En este apartado se va a repetir la descripción del circuito del apartado anterior, pero realizando la descripción del circuito desde un fichero de texto en lenguaje VHDL. El circuito se puede describir de muchas formas: mediante operadores lógicos, mediante tabla de verdad a través de una sentencia CASE, etc. En este apartado la descripción debe hacerse mediante una sencilla sentencia del tipo IF-ELSE, que cargue en la salida Z el valor de I0 si S es 0, o el valor de I1 si S es 1 (la descripción VHDL está disponible en las diapositivas de las clases teóricas). El desarrollo del trabajo en este apartado hará referencia a parte del trabajo hecho en el apartado anterior, por lo que en muchos casos no se indicará explícitamente cómo hacer cada operación. Para este trabajo los ficheros deben guardarse en la carpeta Mux2vhdl dentro de la carpeta Pr4 de la carpeta de trabajo de cada alumno.

- Con Quartus II activado, crear un nuevo fichero desde el comando *New* del menú **File** (o pulsar en el icono). Este comando abre una ventana de selección del tipo de fichero que se

desea crear, elegir el tipo *VHDL File* y pulsar en OK. Se abre una ventana con un editor de texto que tiene la particularidad de colorear en azul palabras clave del código, en verde los comentarios, en morado los *strings*, etc. Editar la entidad con nombre mux2vhd1 (igual que la carpeta, igual que el fichero de texto, no es obligatorio, pero automatiza la creación del proyecto) y la arquitectura de la descripción en el fichero de texto.

El editor también dispone de iconos para facilitar tareas de edición, por ejemplo los iconos  sirven para insertar o eliminar sangrado en la línea activa o en las líneas seleccionadas; los iconos  convierten las líneas de código seleccionadas en comentarios o viceversa; el icono  permite incluir un fichero de texto completo en el fichero que se está editando, etc. El icono  permite introducir la sintaxis básica de las construcciones de uno de los lenguajes utilizables en Quartus II, entre ellos VHDL. Para introducir la construcción hay que seleccionarla a través del menú desplegable y pulsar en *Insert* con lo que su sintaxis aparece en el fichero de texto y, a continuación, hay que rellenar los campos específicos del diseño.

- Guardar el diseño con el comando *Save As* del menú **File** (o el icono ). Guardar el fichero de nombre Mux2vhd1.vhd en la carpeta de nombre Mux2vhd1 de la capeta Pr4 del directorio de trabajo de cada alumno. El nombre del fichero puede ser distinto del directorio, pero usar los mismos nombres hace que el proceso sea más automático. Al igual que en el apartado anterior al guardar el fichero Quartus II pregunta si se quiere realizar un proyecto asociado a este fichero. Responder que sí y generar un proyecto para el dispositivo EP2C35F672C6 de la familia Cyclone II (hay que pulsar siempre en *Next* y salvo en la última ventana que se pulsa en *Finish*).



- Compilar la descripción en un único paso pulsando en el icono . En caso de errores en la descripción aparecen líneas rojas de error en la ventana *Message* con información sobre el tipo

de error, como se muestra en la figura anterior. También pueden aparecer líneas azules de aviso. Pulsando sobre la línea en rojo se muestra en el fichero de texto la línea relacionada con el error. Una vez que la compilación no da error, indicar el número de celdas del dispositivo utilizadas al sintetizar el circuito (campo *Total logic elements* en la ventana *Compilation Report*) y el tiempo máximo de propagación (ejecutar el comando *Classic Timing Analyzer Tool* del menú **Processing**, pestaña tpd).

- Editar un fichero de formas de onda para simulación del circuito. Se puede utilizar el comando *New* del **menú File** para crear y editar el fichero mux2vhdl.vwf de estímulos, igual que se hizo en el apartado anterior, o bien copiar directamente desde sistema operativo el fichero del apartado anterior en la carpeta de trabajo de este apartado con el nombre mux2vhdl.vwf. Una vez editado o copiado el fichero pulsar en el icono , y comprobar los resultados de la simulación en la ventana *Simulation Report*. Guardar los resultados de la simulación mediante el comando *Save Current Report Section As* del menú **File**.

- Guardar y cerrar el proyecto mediante los comandos *Save Project* y *Close Project* del menú **File**.

En este apartado también podrían repetirse las tareas realizadas en el apartado anterior: simulación funcional, medida de tiempos de propagación, de la estimación de la disipación en potencia, aplicación de restricciones al diseño, etc, pero no se realizan ya que no aportan nada nuevo sobre el manejo de la Quartus II con respecto al apartado anterior.

4.4. Diseño de un circuito que genere la distancia de Hamming mediante descripción VHDL.

Siguiendo los pasos de los apartados anteriores realizar la descripción VHDL de un circuito que obtenga la distancia de Hamming de dos palabras de 8 bits (la descripción VHDL está disponible en las diapositivas de las clases teóricas). Para este trabajo los ficheros deben guardarse en la carpeta DHgen dentro de la carpeta Pr4 de la carpeta de trabajo de cada alumno. Los pasos a seguir son: edición de la descripción VHDL en el fichero DHgen.vhd con nombre de la entidad DHgen, creación del proyecto DHgen, creación un fichero de formas de onda DHgen.vwf y edición de los estímulos en él, simulación y comprobación de que el circuito opera correctamente. Una vez compilada sin error la descripción, indicar el número de celdas del dispositivo utilizadas al sintetizar el circuito (campo *Total logic elements* en la ventana *Compilation Report*) y el tiempo máximo de propagación (ejecutar el comando *Classic Timing Analyzer Tool* del menú **Processing**, pestaña tpd).

La principal diferencia con el proyecto anterior radica en la edición de los estímulos, ya que en el multiplexor de dos entradas las entradas y la salida eran de 1 bit. En este apartado las entradas son de 8 bits y la salida de 4. Probar exhaustivamente el circuito requeriría introducir 2^{16} valores distintos (65536 valores), así que se va a probar el circuito con 20 valores aleatorios, aplicados cada 50ns. Después de editar la descripción y compilarla mediante el icono  editar el fichero de formas de onda siguiendo estos pasos:

- Crear un fichero de formas de ondas desde el comando *New* del menú **File** (o pulsar en el icono .
- Comprobar (o fijar) que el tiempo de simulación está a 1 us (20 * 50 ns) con el comando *End Time* del menú **Edit**.

Situar los buses de entrada y salida en la ventana con el comando *Insert Node or Bus* de la pestaña *Insert* del menú **Edit**. Situar en la pestaña *Filter*: el valor *Pins: all* para buscar solo nudos de entrada y salida (otras opciones son posible que permiten acceder a nudos internos), pulsar en *List* para que aparezcan en *Node Found* los nudos disponibles, seleccionar con el ratón de uno en uno los nudos de entrada y salida en modo bus (con todos sus bits, icono ) y pasarlos pulsando en  a la ventana *Selected Nodes*: Pulsar en OK en la ventana *Node Finder* y, en la ventana *Insert Node or Bus* situar el campo *Radix*: a *Binary* y pulsar en OK.

- Pulsar con el botón derecho del ratón sobre el nombre DH y seleccionar *Properties* en el menú desplegable que aparece cambiar formato (*Radix*:) del bus de salida a valor *Unsigned Decimal*.
- Seleccionar una entrada y pulsar en el icono  para aplicar valores aleatorios; elegir *At fixed intervals* fijando el campo *Interval period* a 50 ns desde teclado. Aparece un valor distinto cada 50ns. Usar el zoom o el comando *Fit in Window* del menú **View** para ver bien los valores en pantalla. Realizar lo mismo con la otra entrada.
- Guardar el fichero con el comando *Save as* del menú **File** (o el icono ) con el nombre DHgen.vwf.

Una vez editado las formas de onda, simular el circuito pulsando en el icono  y comprobar que opera correctamente. Usar los comandos del submenú *Netlist Viewer* del menú **Tools** (*RTL Viewer* y *Technology Map Viewer*, en este último haciendo doble-click en las cajas de las celdas aparece su contenido, y situando el ratón sobre su salida aparece la función lógica que realiza) para observar cuál ha sido el resultado de la síntesis del circuito en formato esquemático. Guardar y cerrar el proyecto mediante los comandos *Save Project* y *Close Project* del menú **File**.

4.5. Diseño de un circuito multiplicador a partir de circuitos sumadores.

En la práctica 2 se ha realizado el diseño de un circuito multiplicador 4*4 a partir de puertas AND y de sumadores de 4 bits. En este apartado se va a realizar el diseño desde una descripción VHDL usando las sentencias de descripción del sumador sin signo (*unsigned*) del tipo de las que aparecen en las diapositivas de las clases teóricas y sentencias para representar las puertas AND (explícitamente o, por ejemplo, un *process* con sentencias de tipo *for ...loop*). Para este trabajo los ficheros deben guardarse en la carpeta Mult4_4 dentro de la carpeta Pr4 de la carpeta de trabajo de cada alumno, utilizar como nombre del fichero VHDL Mult4_4.vhd, como nombre de la entidad Mult4_4 y como nombre del fichero de estímulos de formas de onda Mult4_4.vwf.

La descripción VHDL debe hacerse mediante 16 operaciones AND y 3 sumas, y sin utilizar explícitamente el operador de multiplicación (*). Se recomienda definir las entradas como vectores de 4 bits y la salida como un vector de 8 bits. Las operaciones AND se pueden hacer mediante un lazo *for ... loop*, que recorra cada bit de A y haga el AND con cada bit explícito de B, generando las variables (o señales, como se prefiera) P0, P1, P2 y P3 de 4 bits de cada fila de la figura. Se recuerda que para un vector X, en VHDL se puede acceder a uno de sus bits usando una notación del tipo X(1), bit de índice 1 de X por ejemplo, y a un rango de sus bits utilizando una notación del tipo X(3 *downto* 1) (3 bits de X: bits 3, 2, y 1). Obtenidas las señales P, estas se deben sumar ordenadamente como en la figura obteniendo 3 variables (o señales como se prefiera) S0, S1 y S2 de 5 bits, usando los operadores + (suma) y & (concatenación); se recuerda que los operandos de entrada de la suma se deben expandir a 5 bits. Por último, hay que asignar al puerto de salida los bits ordenados mediante el operador de concatenación.

Realizar los pasos de edición del fichero de texto Mult4_4.vhd, creación del proyecto Mult4_4, compilación sin error, creación un fichero de formas de onda Mult4_4.vwf y edición de los estímulos en él, simulación y comprobación de que el circuito opera correctamente.

Una vez compilada sin error la descripción, indicar el número de celdas del dispositivo utilizadas al sintetizar el circuito (campo *Total logic elements* en la ventana *Compilation Report*) y el tiempo máximo de propagación (ejecutar el comando *Classic Timing Analyzer Tool* del menú **Processing**, pestaña tpd).

Para probar el circuito insertar las entradas y la salida en modo bus y fijar su valor en *Radix*: a *Unsigned Decimal*, y generar 20 números aleatorios como se hizo en el apartado anterior. Usar los comandos del submenú *Netlist Viewer* del menú **Tools** (*RTL Viewer* y *Technology Map Viewer*) para observar cuál ha sido el resultado de la síntesis del circuito en formato esquemático. Guardar y cerrar el proyecto mediante los comandos *Save Project* y *Close Project* del menú **File**. Salir de Quartus II mediante el comando *Exit* del menú **File**.

