

# Desarrollo de Hardware Digital

## Seminario 2

### Iniciación a herramientas de especificación, verificación y síntesis automática

#### Objetivos:

- Conocer las diferentes fases del flujo de diseño con Quartus II
- Realizar síntesis automática RT, lógica y física, y comprender las tareas que implica cada etapa
- Realizar simulaciones funcionales y temporales con ModelSim-Altera
- Utilizar una herramienta de análisis temporal, y comprender su utilidad
- Adquirir hábitos recomendables de verificación

## 1. Introducción

En la Figura 1 se muestran las etapas básicas del flujo de diseño con FPLDs. El diseñador principalmente interviene en la especificación del diseño y en su verificación.

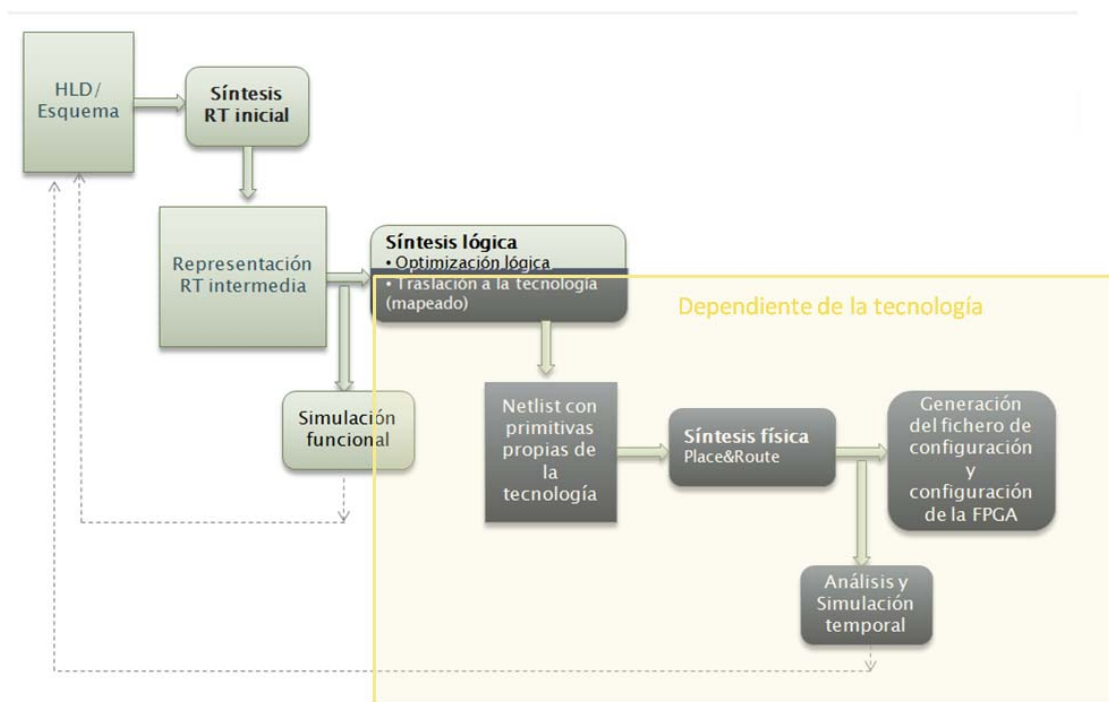


Figura 1: Flujo de diseño con FPLDs

En las siguientes secciones se describe cómo realizar las diferentes fases de desarrollo de hardware digital con Quartus II [1], cuando el diseño se especifica mediante descripciones en VHDL. Se empleará la herramienta ModelSim-Altera [2] para realizar las simulaciones funcional y temporal.

## 2. Flujo de diseño descendente con Quartus II

En esta sección se ilustra cómo acometer la especificación, síntesis automática y verificación de un componente, considerando como ejemplo una descripción funcional VHDL de un contador binario de 8 bits.

### 2.1 Creación de un nuevo proyecto

- Cada diseño se realiza como un proyecto en Quartus II. Para crear un nuevo proyecto se emplea la opción *File -> New Project Wizard*. Tras una primera ventana informativa, es necesario introducir el directorio de trabajo, y el nombre del proyecto (`contador_8bits`) y de la entidad en el nivel de abstracción superior (`contador_8bits`). Conviene asegurarse de que el path del directorio donde se cree el proyecto no contenga espacios en blanco.
- A continuación, se pueden asignar diferentes ficheros ya existentes al proyecto (en nuestro caso, añadir `contador_8bits.vhd`). Más adelante, una vez creado el proyecto, también se podrían asignar ficheros a un proyecto con la opción *Assignments -> Settings -> Files*.
- En la siguiente ventana se selecciona el dispositivo configurable con el que implementar el diseño. En caso de trabajar con la tarjeta DE2-115 [3], se selecciona el dispositivo EP4CE115F29C7 de la familia Cyclone IV E [4].
- Finalmente, existe la posibilidad de utilizar herramientas EDA (Electronic Design Automation) de especificación, síntesis automática y verificación de otros suministradores diferentes de Altera (indicar ModelSim-Altera como herramienta EDA para simulación, lenguaje VHDL).

### 2.2 Especificación del diseño

Puesto que ya se dispone del fichero `contador_8bits.vhd`, tan sólo es necesario abrirlo, pulsando dos veces sobre el nombre del fichero en la ventana de arriba a la izquierda (*Entity*). Se abrirá una ventana de texto con el contenido del fichero.

### 2.3 Síntesis RT-lógica

La síntesis RT-lógica se realiza seleccionando *Processing -> Start -> Analysis y Synthesis, Ctrl-K*. En la etapa inicial de síntesis RT se infieren los componentes con capacidad de almacenamiento (registros, memoria), las máquinas de estado y ciertos módulos de lógica combinacional. A continuación, durante la síntesis lógica se optimizan las redes lógicas combinacionales y se realiza una traslación a la tecnología infiriendo un circuito equivalente a partir de primitivas propias de la FPGA.

### 2.4 Visualización de la *netlist* resultante de la síntesis RT inicial

Como resultado de la síntesis RT inicial se genera un circuito con componentes genéricos (independientes de la tecnología, es decir de la FPGA seleccionada en el proyecto), que es posible visualizar con la opción *Tools -> Netlist Viewers -> RTL Viewer*. Resulta muy útil consultar este esquema antes de realizar la simulación funcional para, a partir de la descripción VHDL, localizar fundamentalmente los registros y máquinas de estado, diferenciándolos de la lógica combinacional.

## 2.5 Simulación funcional con ModelSim-Altera

La simulación funcional se realiza para constatar que el modelo VHDL describe correctamente el componente que se desea implementar, sin tener en cuenta ningún tipo de retardo de propagación.

La secuencia de tareas a realizar es la siguiente:

- Utilizar la opción *Tools → Run Simulation Tools → RTL Simulation* para ejecutar de forma nativa ModelSim-Altera<sup>1</sup>.
- Desde ModelSim-Altera, utilizar la opción *Simulate → Start Simulation*. En la pestaña “*Design*”, indicar que el fichero de diseño es `contador_8bits.vhd` desplegando *work* (referido al directorio de trabajo).
- Abrir una ventana *Wave* de formas de onda, seleccionando *View → Wave*.
- Definir las formas de onda de los estímulos de entrada. Accionar el botón derecho del ratón sobre el nombre de la señal `Reset` en la ventana *Objects* y seleccionar la opción *Create Wave*. A continuación, se elige la opción *Constant*, se fija un intervalo de tiempo p.e. de 26 ns, y se le asigna un valor constante igual a 0. Desde la ventana *Wave* de formas de onda, seleccionar también con el botón derecho la señal `Reset` y seleccionar la opción *Edit → Create/Modify Waveform*, para indicar que en el intervalo desde 0 a 50 ps dicha señal debe tomar el valor constante 1. Análogamente, se crea una onda para la señal `Reloj`, seleccionando en este caso la opción *Clock* después de *Create Wave*, fijando el mismo intervalo (de 0 a 26 ns) y dejando para este ejemplo los valores por defecto para el valor inicial y para los intervalos en alta y en baja.
- Añadir las señales de salida a la ventana de formas de onda, accionando el botón derecho del ratón sobre la señal `Salida` en la ventana *Objects* y seleccionando *Add → To Wave → Selected Signals*.
- Simular con la opción *Simulate → Run → Run -All*. Desde la ventana *Wave*, con la opción *View → Zoom Range* se puede determinar el rango a visualizar. Es posible representar en hexadecimal el valor de la señal `Salida` con la opción *Radix → Hexadecimal* (botón derecho del ratón posicionado sobre la señal).
- Una vez examinados los resultados de simulación, al cerrar la ventana *Wave* existe la posibilidad de guardar en un fichero de texto los comandos utilizados para definir los estímulos de simulación. Responda afirmativamente a la pregunta “*Do you want to save wave commands?*” e introduzca `wave_funcional.do` como nombre del fichero de comandos, que se almacenará en `<directorio_proyecto>/simulation/modelsim`. Podría repetir la simulación con los mismos estímulos reiniciándola (*Simulate → Restart*) y ejecutando el fichero de comandos `wave_funcional.do` mediante la opción *File → Load*.
- Una simulación finaliza con la opción *Simulate → End Simulation*.

## 2.6 Compilación

En Quartus II se ordena la compilación de un diseño con la opción *Processing → Start Compilation*. La compilación engloba principalmente el análisis y elaboración de la entidad, la síntesis RT-lógica (Análisis & Síntesis), la síntesis física (Fitter), la generación del fichero de configuración del FPLD (Assembler), y la extracción de parámetros temporales (TimeQuest Timing Analyzer). En la ventana de la izquierda se presenta el progreso o avance de las diferentes etapas, y a la derecha un resumen de los múltiples informes que se generan

---

<sup>1</sup> Antes de ejecutar ModelSim-Altera de forma nativa es necesario comprobar que en *Tools → Options → General → EdaTool Options* el campo “ModelSim-Altera” contiene el directorio donde se encuentra instalado el ejecutable de ModelSim-Altera (por ejemplo, `C:\altera\12.0sp2\modelsim_ase\win32aloem`)

## 2.7 Visualización de la netlist resultante de la síntesis física y localización en el plano de base del dispositivo

Quartus II ofrece también la posibilidad de visualizar el esquemático resultante de la compilación con celdas propias de la FPGA asignada al proyecto, y trabajar sobre el plano de base del dispositivo, tal como quedaría después de su configuración.

Para visualizar el resultado de la síntesis física se utiliza la opción *Tools → Netlist Viewers → Technology Map Viewer (Post-Fitting)*. El esquemático del circuito inferido se representa con celdas propias de la tecnología de la FPGA en el que se implementa el diseño.

Consultado la opción *Tools → Chip Planner (Floorplan and Chip Editor)* se puede conocer en qué bloques internos de la FPGA se ha implementado cada función. Para visualizar el plano de base completo de la FPGA, seleccionar *View → Fit in Window*. Actuando sobre las opciones de zoom o búsqueda se puede visualizar la ubicación de los elementos del diseño (pines, registros, etc.) en la FPGA. También se pueden localizar en el plano de base utilizando el botón derecho del ratón una vez nos posicionamos sobre el elemento en una netlist. Por ejemplo, colocando el ratón sobre el módulo *Contador* en el *RTL Viewer* y seleccionando con el botón derecho *Locate → Locate in Chip Planner (Floorplan ...)* se visualiza la ubicación de los bits correspondientes a la señal *Contador*.

## 2.8 Análisis temporal

El analizador temporal de Quartus II permite conocer algunos parámetros temporales del diseño, como por ejemplo la frecuencia máxima de funcionamiento (*Fmax*), el tiempo de establecimiento (*Setup*) o el tiempo de mantenimiento respecto al flanco activo del reloj (*Hold*). Para conocer el resultado del análisis temporal se puede consultar la entrada *TimeQuest Timing Analyzer* del informe de compilación.

En el ejemplo del contador de 8 bits se observan varias entradas en rojo. Se debe a que si no se establecen restricciones a la señal de reloj, por defecto se considera un reloj de 1GHz (se puede comprobar desde *TimeQuest Timing Analyzer → Clocks*). A esa frecuencia, el retardo de propagación del camino más largo no respeta el tiempo de setup (ver columna *Slack en Slow 1200mV 85C Model → Setup Summary*). Conviene ajustar el periodo de reloj de acuerdo con la estimación de frecuencia máxima que proporciona el analizador en *Slow 1200mV 85C Model → Fmax\_Summary*. Las restricciones temporales se almacenan en ficheros *.sdc*. Establecemos un periodo de reloj de 4ns del siguiente modo:

- Desde *TimeQuest Timing Analyzer → Clocks*, pulsamos el botón derecho del ratón sobre la señal *Reloj*, y seleccionamos la opción *Edit Clock Constraint...*
- Se abre una nueva ventana correspondiente al analizador temporal, y dentro de esta otra más pequeña con las restricciones para la señal de reloj. Editar el campo *Period* fijando el valor 4, y pulsar *Run*.
- Para que el comando que establece la restricción impuesta se almacene en un fichero *.sdc* (y sea tenido siempre en cuenta al realizar la compilación desde Quartus II), es preciso indicarlo en la opción *Constraints → Write SDC File*. Y a continuación, desde Quartus II añadir en *Assignment → Settings → TimeQuest Timing Analyzer* el nombre del fichero *.sdc* que se ha generado (*contador\_8bits.out.sdc*). No olvidar pulsar la opción *Add*. Al compilar de nuevo el diseño se observa cómo los resultados del análisis se ajustan a las restricciones impuestas.

## 2.9 Simulación temporal con ModelSim-Altera

Una vez analizados los parámetros temporales, se realiza una nueva simulación en la que aparecerán retardos en las respuestas a los estímulos de entrada teniendo en cuenta las estimaciones del analizador temporal, siguiendo los siguientes pasos:

- En primer lugar, asegurarse de que no se tiene ninguna ventana abierta con ModelSim.
- Utilizar la opción *Tools* → *Run Simulation Tools* → *Gate Level Simulation ... para ejecutar de forma nativa ModelSim-Altera*. Dejar el modelo temporal del dispositivo que aparece por defecto ("Slow -7 1.2V 85C"), y pulsar Run.
- Desde ModelSim-Altera, utilizar la opción *Simulate* → *Start Simulation*.
  - En la pestaña "SDF", añadir el fichero "*<directorio\_proyecto\_quartus>/simulation/modelsim/contador\_8bits\_vhd.sdo*".
  - En la pestaña "Libraries", añadir la biblioteca "cycloneive".
  - En la pestaña "Design", seleccionar el fichero de extensión .vho que aparece por defecto en el directorio de trabajo (desplegando work). Observar que el fichero que ahora se simula no es la descripción contador\_8bits.vhd sino el fichero contador\_8bits\_7\_1200mV\_85c\_slow.vho que se ha generado en el subdirectorio simulation/modelsim, en el que se incluye la información del modelo del dispositivo.
  - Definir los estímulos de simulación respetando las restricciones temporales del diseño. Al realizar la simulación temporal, se deben tener en cuenta los resultados del análisis temporal para definir los estímulos de entrada, en particular la señal de reloj debe respetar las restricciones de frecuencia máxima.
- Simular con la opción *Simulate* → *Run* → *Run -All*.

### 3. Actividades y ejercicios prácticos

**3.1** Considerar el diseño contador\_8bits.vhd y realizar las siguientes tareas:

- a) Dibujar un esquema de bloques del circuito que se sintetiza.
- b) ¿Cuál es el número de elementos lógicos (LEs) ocupados en la FPGA tras realizar la síntesis? ¿Cómo puede conocer este dato con Quartus II?
- c) ¿Cuál es la frecuencia máxima estimada por el analizador temporal?
- d) ¿Cómo ha definido las formas de onda de la señal reloj y de la entrada reset para realizar la simulación temporal? ¿Por qué?
- e) ¿Cómo se ven afectadas la ocupación de LEs y la frecuencia máxima si se realiza la síntesis cambiando el número de bits del contador a 4 y a 16?

**3.2** Considerar la descripción contador\_BCD, y responder las siguientes cuestiones:

- a) Dibujar el esquema de bloques del circuito genérico que se sintetiza (también para las instancias del componente contador\_ascendente).
- b) ¿Cuál es la secuencia de valores de salida que genera cada instancia de contador\_ascendente? ¿Qué utilidad tiene la entrada EN de dicho componente? ¿Qué secuencia de valores de salida genera el contador\_BCD? ¿Cuándo se activa su salida MAX? Comprobar las respuestas mediante simulación.
- c) ¿Cuál es el porcentaje de ocupación de LEs y flip-flops en la FPGA de la DE2-115?
- d) ¿Cuál es la frecuencia máxima estimada por el analizador temporal?
- e) ¿Qué LAB de la FPGA se ha asignado al flip-flop correspondiente al bit menos significativo del contador?

**3.3** *Opcional:* Implementar un divisor de frecuencia (denominado divisor\_1s) que genere un pulso de salida cada segundo a partir de un reloj de 50 MHz, y conectarlo al contador BCD para que en los visualizadores de 7 segmentos de la tarjeta DE2-115 se muestre la salida del contador, de modo que su valor cambie cada segundo. *NOTA: Puede tomar como referencia las descripciones div\_frec.vhd y top.vhd del Seminario 1.*

## 4. Documentación

### [1] Quartus II Web Edition 12.0 Service Pack 2

Descarga e instalación:

<https://www.intel.com/content/www/us/en/programmable/downloads/software/quartus-ii-we/120sp2.html>

### [2] Modelsim Altera-Starter v10.0d <sup>2</sup>

<https://www.intel.com/content/www/us/en/programmable/downloads/software//modelsim-starter/12.0.html>

### [3] Tarjeta DE2-115:

[https://www.intel.com/content/dam/altera-www/global/en\\_US/portal/dsn/42/doc-us-dsnbk-42-1404062209-de2-115-user-manual.pdf](https://www.intel.com/content/dam/altera-www/global/en_US/portal/dsn/42/doc-us-dsnbk-42-1404062209-de2-115-user-manual.pdf)

### [4] Cyclone IV Device Handbook:

<https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-iv/cyclone4-handbook.pdf>

---

<sup>2</sup> Es preciso descargar e instalar en el siguiente orden tres paquetes ModelSim-Altera Starter Edition:

1º) v10.0d Software Download for Quartus II v12.0

2º) v10.0d Service Pack 1 for Quartus II v12.0

3º) v10.0d Service Pack 2 for Quartus II v12.0