

LABORATORIO DE ARQUITECTURA DE COMPUTADORES,

I.T.I. SISTEMAS Y GESTIÓN

GUÍA DEL ALUMNO

SESIÓN 1: Simulación con VHDL y entorno V-System

Objetivo

- Comenzar a familiarizarse con los aspectos básicos de VHDL como lenguaje para el modelado y la simulación de Arquitecturas.
- Crear un entorno personalizado de ejecución del sistema V-System independiente y libre de interferencias con otros usuarios del mismo ordenador que sirva para el resto de las prácticas.

Método de trabajo

Lee cuidadosamente las indicaciones de la guía, realizando los pasos que se indican. La guía tiene como propósito que TU te asegures de que comprendes lo que va sucediendo mientras realizas lo que te indica que hagas: no tiene propósitos evaluativos.

Para todas las sesiones, necesitaremos textos de referencia del lenguaje para realizar programas o para comprender lo que ocurre durante las simulaciones. Por lo tanto se recomienda traer libros, apuntes y cualquier bibliografía de VHDL que sirva para hacer consultas. En el apartado “bibliografía” se proporcionan algunas referencias.

También traer un **diskette** para llevar copia de los ficheros que se generen y se deseen conservar ya que, al tratarse de una máquina compartida no hay garantías de que más tarde encontremos los que dejamos hoy. Pero tener en cuenta que V-System no trabaja bien con los diskettes, así que siempre habrá que **copiar de/al disco duro y trabajar en él**.

Bibliografía

- “VHDL: Lenguaje para síntesis y modelado de circuitos”. F.Pardo y J.A.Boluda. Ed. Ra-Ma, 1999.
- Lluís Terés y otros “VHDL: Lenguaje estándar de diseño electrónico” Mc Graw Hill, 1998
- Buscar en la WEB: Google encuentra referencias y pueden encontrarse libros en pdf

Paso I: Creación de un entorno de trabajo propio

1. Arrancar el ordenador del puesto de trabajo y dejar que inicie el Sistema Operativo Windows.
2. Entrar en una sesión de *windows 2000*

cuentas de usuario (Solo para el L4)

Son locales en cada uno de los puestos. Las cuentas están definidas para el dominio ESTEL4-PUESTOnn donde nn va de 1 a 14

Para las cuentas de Lab. Arquitectura de Computadores de ITIG

a1lacges-pnn con nn de 1 a 14, es el número de puesto

a2lacges-pnn con nn de 1 a 14

a3lacges-pnn con nn de 1 a 14

b1lacges-pnn con nn de 1 a 14
 b2lacges-pnn con nn de 1 a 14
 b3lacges-pnn con nn de 1 a 14

Para las cuentas de Lab. Arquitectura de Computadores I de ITIS

b1lacges-pnn con nn de 1 a 14 es el número de puesto
 b2lacges-pnn con nn de 1 a 14
 b3lacges-pnn con nn de 1 a 14

3. Crear una carpeta (o directorio) en el directorio raíz del disco C:/ combinando los nombres o los apellidos de quienes formen el equipo de trabajo. Dicho nombre NO DEBE TENER mas de 8 caracteres ni contener espacios. Evitar nombres muy comunes: el PC lo utilizan otras personas. Para el resto de la guía, llamaremos a este directorio **xyyy**.
4. Crear también el directorio **c:/xyyy/fuentes**. En este directorio puedes ubicar tus ficheros fuentes o copiar los que se proporcionan para las prácticas y modificarlos.
5. Desde Inicio-Programas etc., o con el explorador de windows, situándose en **c:/vhdl**, arranca la aplicación V-System (con el comando **vsystem.exe**).
6. Maximizar la ventana. Asegurarse de que son visibles las barras "**Tool**" y "**Status**" (de lo contrario seleccionar *Window>Tool Bar* y *Status Bar*).
7. Activar todas las opciones en *Window>settings* para así salvar la configuración al salir de la aplicación.
8. **CAMBIAR DIRECTORIO a c:/xyyy** de una de las siguientes formas:
 - vía línea de comandos en la "*Transcript window*": **cd c:/xyyy**
 - vía Menú: *File>Directory*, seleccionar el directorio **c:/xyyy**, y botón "Change to".
9. Este debe ser **siempre** el lugar desde el botón cual se ejecute la aplicación para mantener así las referencias a los recursos propios de **xyyy** y recuperar la configuración y preferencias personalizadas. Esto evita interferencias con otros grupos. SIEMPRE que arranques V-System debes inmediatamente cambiarte a tu directorio como acabas de hacer.
10. Comprobar directorio actual: comando "**pwd**" (en la ventana Transcript).
11. Crear **una librería de diseño** para que contenga los resultados de las compilaciones que hagas en las prácticas. Su nombre será "**work**". Utilizar *Library>New* para ello y, tras escribir el nombre "**work**", pulsar "**create**". NO puede usarse el sistema operativo para esto. No es un directorio convencional.
12. Comprobar las correspondencias entre nombres lógicos y *path* de ficheros en disco de las librerías del lenguaje. Para ello, en *Library>Mapping* debe verse:

Library	Directory
STD	C:\VHDL\STD
IEEE	C:\VHDL\IEEE
work	C:\XXYY\work

13. Seguramente no aparecerá work todavía. Pulsar en el botón "new" y definir entonces la relación entre el nombre lógico (*library*) "work" y el directorio "C:\XXYY\work". Al hacerlo ya se visualizarán las tres librerías y sus localizaciones en disco.
14. Crear un **proyecto** propio para mantener los parámetros del entorno de trabajo mediante *Project>New*; escribir **vsystem.ini**. Ese será el fichero que leerá el sistema tras cambiarse al directorio **xyyy**. Inmediatamente se recuperará la configuración personalizada.
15. Verificar que el sistema utiliza este fichero (ver submenú *Project* con este fichero explícitamente seleccionado en lugar de el de inicio por defecto).

16. Ahora vamos a establecer una opción de compilación que permite reconocer la sintaxis VHDL' 93: “*Options>Compile Options*” seleccionar **todas** las casillas **salvo** la que dice “*use explicit declaratiосn only*”. Esto también se guardara en el “proyecto” individualizado.

Paso II: Edición de fuentes VHDL

17. Arrancar el “bloc de notas” para crear un fichero fuente que llamaremos “mux.vhd”.
18. Crear una entidad llamada mux, con 3 entradas de tipo bit (a, b, sel) y una salida también de tipo bit (out).
19. Realizar una arquitectura RTL y otra comportamental que describan un multiplexor (esto es: **out** es **a** cuando **sel** es 0 y **out** es **b** cuando **sel** es 1), según lo visto en la introducción teórica del lenguaje. Puede usarse el ejemplo al final de la guía.
20. Incluir alguna línea de comentario.
21. Salvar (escribir) la fuente con el nombre indicado **mux.vhd** en el directorio de fuentes descrito anteriormente. Comprobar que el editor crea un fichero de texto puro (no enriquecido y sin extensión aparte de la de "vhd" que da el usuario).

Paso III: Compilación de mux.vhd en Vsystem

22. Compilar mux.vhd pulsando *File>Compile* o Botón *Compile* de la Barra de Herramientas; bajar a *fuentes*, seleccionar el ficheros fuente y pulsar *Compile*.
23. Corregir errores con el editor de texto, salvar y recompilar hasta completar la compilación sin errores. Salir con “Done”.
24. Se puede observar que la estructura work (se puede recorrer con el explorer como si se tratase de un directorio) contiene a su vez subestructuras con las arquitecturas, tablas de símbolos y una versión “instrumentada” del código fuente. Asimismo, la librería work contiene ahora la nueva entidad: *Library>Contents*.

Paso IV: Simulación de una arquitectura

25. Al pulsar SIMULATE, el sistema requiere que se seleccione la librería, entidad y arquitectura con la que se desea trabajar. De momento usamos solo la librería **work** que por defecto se ha creado en el directorio de trabajo, la entidad **mux** y la arquitectura que deseese. También especificamos nano-segundos (ns) como la unidad de tiempo elegida.
26. Para poner todas las señales en las ventanas Wave y List, escribir (en la ventana transcript)
 - list *
 - wave *
27. Aparecerán las 7 nuevas ventanas que intervienen en la simulación. Con Window>restore All, Window>Tile Vertically y re-dimensionando con el ratón manualmente, configurar el espacio según las preferencias, pero sugerimos lo siguiente para este primer ejemplo:
 - minimizar las ventanas de variables, estructura, proceso, source y signal
 - dejar wave y list ajustando el ancho de list al indispensable para ver las columnas que se muestran. Esto deja mas espacio a wave.
28. A continuación vamos a “forzar la señal **sel** al valor “1” a los 100 ns, luego a “0” a los 300 ns (desde el comienzo) y repetir este ciclo cada 400 ns (también referido al comienzo), como si fuera un reloj. Aplicamos este estímulo mediante el comando mostrado (tambien vía signal>force signal). Observar que por defecto todas las señales comienzan en 0.
 - Force sel 1 100, 0 300 –repeat 400

29. Pulsar el boton RUN y observar los cambios en las ventanas Signal, List y Wave. Añadir cursores en la ventana Wave que marquen las transiciones. Reiniciar (File> Restart) y repetir la simulación.
30. Ahora, sabiendo como se fuerza una señal, reiniciar la simulación variando los valores de las señales a y b a la entrada del mux, progresiva y controladamente para comprender lo que sucede. Para ello puede usarse tanto la ventana de transcript como la interfaz visual:
 - Signal> force signal (signal: /a, value: 1, delay:- , Freeze. Apply)
 - Force –freeze /a 1 0
31. Forzar las señales (/a o /b) introduciendo un valor para el tiempo de aplicación:
 - en modo delay 50 ns (ej: Force –freeze /a 1 50)
 - en modo absoluto (ej: Force –freeze /a 1 @nnn) siendo nnn algo cercano al momento de simulación en que se esté.
32. Observar y comprobar en la ventana Wave el uso de cursores, por ejemplo:
 - Al añadir un cursor se dibuja una línea vertical y se le asocia una banda horizontal en la parte inferior de la ventana con el valor del cursor y el tiempo entre éste y el siguiente cursor; la banda desaparece al borrarle.
 - Al mover el cursor, se muestra el valor de las señales de acuerdo al tiempo correspondiente a su movimiento
33. Observar y comprobar en la ventana List :
 - Si dos señales han cambiado a la vez, puede que haya dos líneas con el mismo tiempo y dos valores de “delta”.
34. Hacer una búsqueda en la ayuda de vsystem. Seleccionar Help en la tool bar a continuación Index, seleccionar Graphical Interface topics y ver todas las opciones de la tool bar.
35. Para salir de la aplicación: File Exit V-System.

NOTAS

1. Las mayúsculas y minúsculas no se distinguen en el lenguaje.
2. vsystem solo maneja nombres de ficheros de hasta 8 caracteres alfanuméricos puros.
3. Las barras “/” y “\” para comandos en Vsystem se usan indistintamente.
4. File> Restart re-inicia el proceso de simulación.
5. Las Señales elegidas pueden modificarse o borrarse de cada ventana desde el menú edit
 1. La “flecha hacia arriba” permite retroceder comandos en la ventana transcript. Las demas flechas también hacen lo propio.

MUY IMPORTANTE: Recuerda que los fuentes han de compilarse de forma que primero lo hagamos con los elementos primitivos y luego los derivados (que utilizan estos elementos primitivos).

Fuente: mux.vhd

```
----- Programa mux.vhd -----  
  
entity mux is  
  port (a  : in bit;  
        b  : in bit;  
        sel : in bit;  
        salida : out bit);  
end mux;  
  
-- descripcion comportamental de mux; a,b,sel lista de sensibles  
architecture comportamental of mux is  
begin  
  process (a, b, sel)  
  begin  
    if (sel = '0')  
      then salida <= a;  
      else salida <= b;  
    end if;  
  end process;  
end comportamental;  
  
-- descripción RTL_1  
  
architecture RTL_1 of mux is  
  signal nosel, ax, bx: bit;  
begin  
  nosel <= NOT sel;  
  ax <= a AND nosel;  
  bx <= b AND sel;  
  salida <= ax or bx;  
end RTL_1;  
  
--- otra descripción RTL mas elegante  
architecture rtl_mux of mux is  
begin  
  salida <= a WHEN sel ='0' else b;  
end rtl_mux;  
-----
```