

INTRODUCCIÓN AL ACTIVE-HDL SIM

Juan Antonio Jaramillo Gómez
Isaac Guzmán Domínguez

Referencia de este artículo [1].

El propósito del presente artículo es el de dar una introducción al programa "Active-HDL™ SIM", mostrando sus principales características y los pasos requeridos para la realización de una simulación lógica de un archivo de VHDL.

Palabras clave.

Active-HDL, simulación y VHDL.

Abstract.

The purpose of this paper is to give an introduction to "Active-HDL™SIM", showing their principal characteristics and their requirements steps to realize a logic simulation with a VHDL file.

Keywords.

Active-HDL, simulation, VHDL.

Introducción.

El programa Active-HDL™SIM se encuentra dentro de Warp™, desarrollado por Cypress™. Es una herramienta de simulación gráfica que utiliza los modelos de tiempo producidos por Warp, es decir, es una herramienta de síntesis de VHDL o Verilog para los PLDs y CPLDs de Cypress™.

Este artículo se divide en tres partes.

1. Creando un modelo de simulación
2. El proceso de simulación
3. Aplicación de los estímulos

Creando un modelo de simulación

El archivo de entrada del Active-HDL Sim es un modelo de simulación post-síntesis. El entorno Warp puede generar el modelo de simulación post-síntesis en muchos diferentes formatos. El Active-HDL Sim necesita un modelo de simulación IEEE 1164/VHDL

Seleccionando el formato del archivo

Para los usuarios del Warp versión 6, en el menú Project>Compiler Options... activa la ventana de Opciones del Compilador (Compiler Options), que por default se posiciona en la pestaña de síntesis (synthesis tab). En la parte inferior derecha de la pestaña de síntesis hay un menú de persiana en donde se selecciona el modelo temporal que se utilizará en la simulación. Para éste caso seleccionar el Active-HDL Sim/Active-VHDL o el 1164/VHDL, como se muestra en la figura 1, y verificar que la casilla Enable Testbench Output esté seleccionada

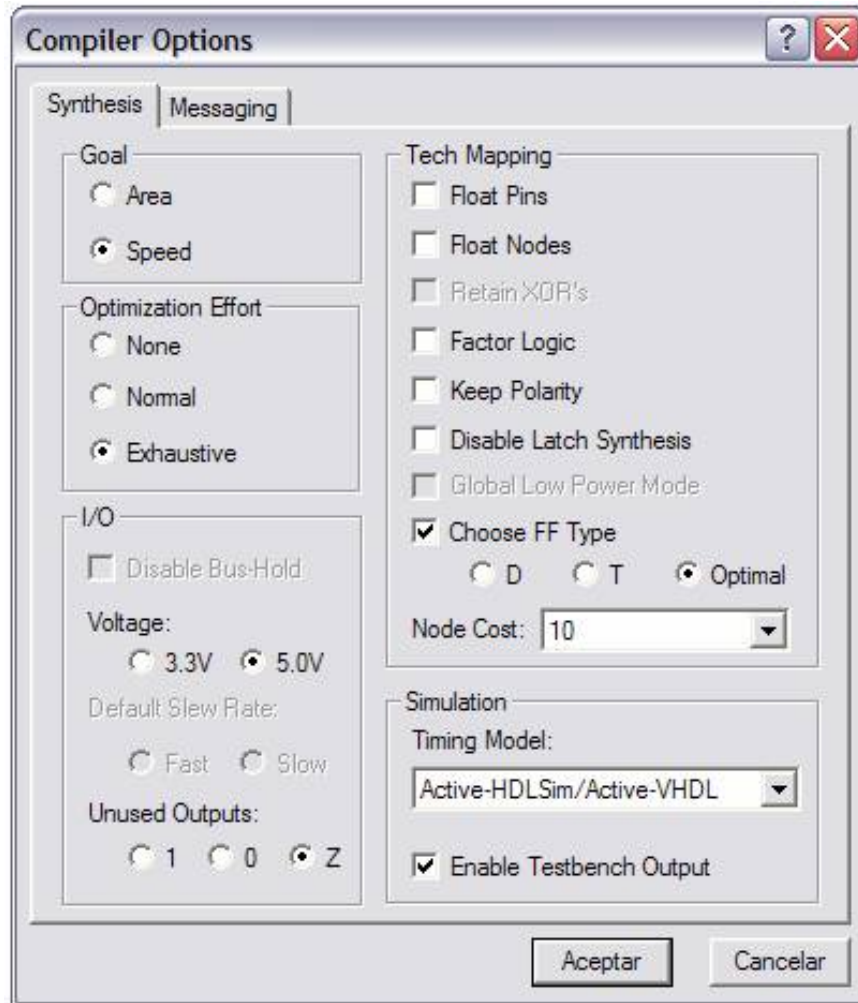


Figura 1: Selección del modelo temporal.

Cuando se tenga un proyecto ya capturado y listo para compilar, coloque el archivo de VHDL en “set top” (en la ventana de proyecto dar botón derecho sobre su archivo a compilar y seleccione “set top”), como se muestra en la figura 2a, y compile su código de VHDL. Esto generará un modelo de simulación con el mismo nombre y extensión, dentro de una nueva carpeta llamada “vhd”, para evitar que se sobre-escriba su archivo original, como se muestra en la figura 2b.

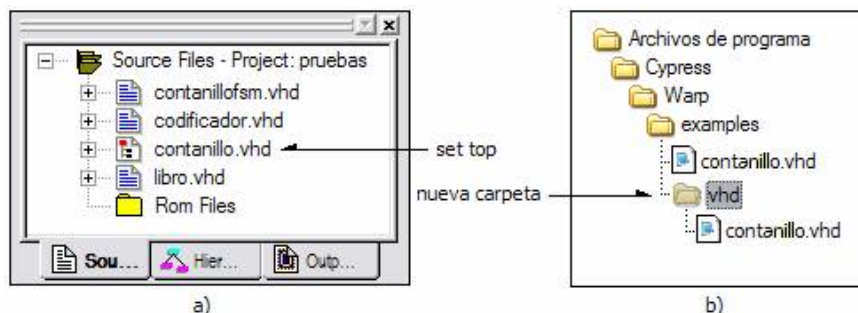


Figura 2: a) Muestra la forma en que queda un archivo con set top, b) Creación de la carpeta VHD para la simulación

El proceso de simulación

Una vez abierto el Active-HDL Sim, usted encontrará un menú, las barras de herramientas estándar y de formas de ondas (standar toolbar, waveform toolbar), mostradas en las figuras 3, 4 y 5 respectivamente.

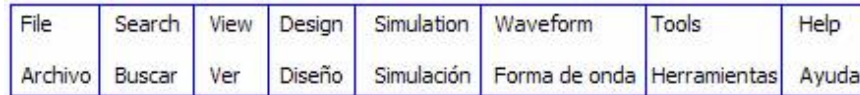


Figura 3: Barra de menú.



Figura 4: Barra de herramienta estándar.

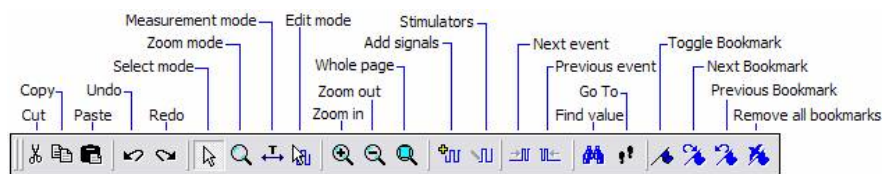


Figura 5: Barra de herramienta de formas de onda.

Los pasos a seguir para simular un diseño ya compilado con éxito en Galaxy™y teniendo abierto el programa Active-HDL Sim son:

1.- Abrir el archivo “vhd” que contiene el modelo de simulación (Active-HDL Sim/Active-VHDL o el 1164/VHDL) con el botón “Open VHDL file for simulation” o en en menú File¿Open VHDL..., recordando que este archivo es el que se encuentra en la subcarpeta “vhd”.

En caso de abrir el archivo original que se encuentra en la carpeta de trabajo, se pueden visualizar los errores generados en la ventana de la consola (ver menú View>Console).

2.- Inicializar la simulación en Simulation¿Initialize Simulation.

Esto abrirá una ventana para simular las formas de onda, y si se desea se puede abrir otra ventana utilizando el menú File¿New Waveform o File>Open Waveform.

3.- Añadir las señales con el botón “Add Signals” o en Waveform>Add Signals...

Esto abrirá una ventana con un listado de todas las señales de entrada, de salida y nodos internos disponibles para en su diseño compilado. Hay que seleccionar una o más señales y presionar el botón “Add”.

4.- Asignar los estímulos con el botón “Stimulators” o en Waveform>Stimulators..., activados sólo cuando se elige una señal, por lo menos.

Esto abrirá la ventana de estímulos (ver figura 6), en donde se puede elegir, en la pestaña de Signals, un tipo de estímulo (Clock, Custom, Formula, Hotkey, Predefined y Value) y asignarlo a la señal al presionar el botón “Apply”. Sin necesidad de cerrar la ventana, se pueden añadir más estímulos a las otras señales, sólo hay que seleccionar la señal y el nuevo tipo de estímulo.

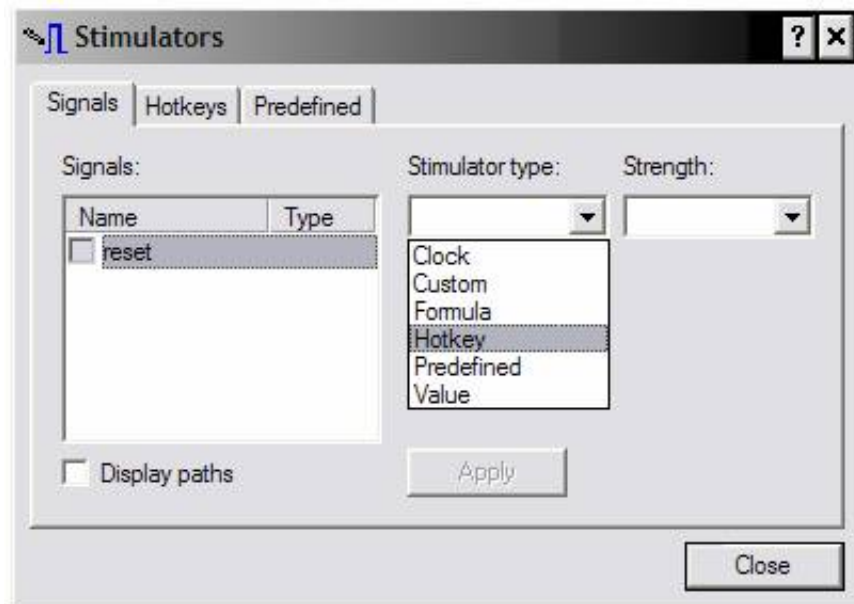


Figura 6: Ventana de estímulos.

C 5.- orrer la simulación en el menú Simulation o con los botones Run, Run until... o Run for (F5). Una vez teniendo los estímulos colocados e inicializada la simulación, se pueden realizar cualquiera de las siguientes ejecuciones: correr (Run) la simulación por un tiempo máximo de 2ms; correr hasta (Run until) el tiempo marcado por la ventana que se activa, y correr por (Run for) el tiempo estipulado en el espacio contiguo (Time to run). La simulación se parará una vez terminado el tiempo especificado en la solicitud. Para poder reinicializar la simulación hay que ir al menú Simulation; Restart simulation o presionar el botón con el mismo nombre.

6.- Observar la forma de onda y modificarla con los comandos en el menú Waveform. Las características de la forma de onda, como estilo, colores, engrillado, etc., se puede cambiar por medio de las propiedades de cada señal (clic derecho sobre la señal e ir a propiedades)

7.- Salvar y/o copiar la forma de onda. Es posible salvar las señales en un archivo con extensión "awf" a través de save, save as... o save all. Si se desean copiar la(s) forma(s) de onda, primero hay que seleccionarla(s) y después con el botón derecho seleccionar copiar, esto permitirá pegarlas en un documento de Word, CorelDraw, PaintBrush, etc.

Aplicación de los estímulos

Los tipos de estímulos existentes son:

Reloj (clock)

Este estímulo produce una señal rectangular definida por la frecuencia/periodo, un desvío inicial de tiempo (offset), ciclo de trabajo y un valor inicial.

Al gusto del cliente (Custom)

Este estímulo es creado utilizando el editor de formas de onda.

Fórmula

Una fórmula genera una forma de onda definida por una sintaxis simple. La forma de onda es definida por una secuencia de pares de valor-tiempo dados por default en picosegundos. Para repetir el estímulo en un periodo específico, hay que añadir el modificador -r.

La sintaxis de la fórmula es: <valor> <tiempo> [, <valor> <tiempo>...][-r <periodo>]

Hotkey

Este estímulo se asocia a una tecla del teclado de la computadora, que cambiará de valor cada vez que se presiona dicha tecla.

Predefinido

Los estímulos predefinidos son señales de reloj de tiempos ya establecidos o bien una fórmula, aunque también se pueden editar y crear nuevos estímulos.

Valor

Existen estímulos con un valor ya asignado como son 'U', 'X', '0', '1', 'Z', 'W', 'L', 'H', '-', colocándose desde uno hasta más bits

A continuación se muestra un ejemplo práctico de un circuito contador de anillo para visualizar la aplicación de los estímulos.

-- Archivo de un contador de anillo para ver la simulación

```
Library ieee;  
use ieee.std_logic_1164.all;  
use work.std_arith.all;  
ENTITY anillo IS PORT (  
  clk : in std_logic ;  
  reset : in std_logic ;  
  q : buffer std_logic_vector (3 downto 0));  
END anillo;  
architecture contador of anillo is  
  begin  
  process(clk, reset) begin  
    if reset = '1' then q <= "0001";  
    elsif (clk'event and clk = '1') then q <= q rol 1;  
    end if;  
  end process;  
end contador;
```

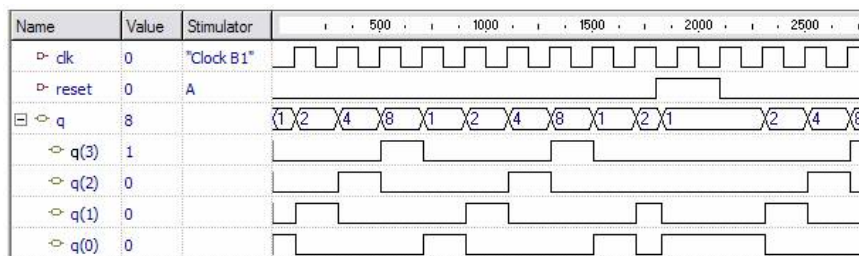


Figura 7:

Conclusiones

Se ha mostrado una breve introducción al simulador temporal Active-HDL Sim, que se utiliza para verificar el funcionamiento de los diseños realizados en VHDL dentro de Galaxy. Pueden utilizarse diseños sobre VHDL y Verilog para enfocarlos a los dispositivos de Cypress. Para más detalles contactar con los autores o con la ayuda del software.

Referencias

- [1] Kevin Skahill, VHDL for Programmable Logic, Addison Wesley Longman
- [2] An Introduction to Active-HDL™ SIM – AN1015 (12, Nov, 2001), www.cypress.com
- [3] J. A. Jaramillo G., I. Guzmán D., R. Herrera A., Apuntes del curso: Diseño Básico de Sistemas Digitales con VHDL, junio 2005.
- [4] J. A. Jaramillo G., I. Guzmán D., Apuntes del curso: Diseño Intermedio de Sistemas Digitales con VHDL, junio 2006.

Artículo realizado con y para el apoyo del programa CGPI 544 que involucra los proyectos CGPI20061916 y CGPI20061619