INTRODUCCIÓN AL ACTIVE-HDL FSM

.

El propósito de éste artículo es el de dar una introducción al programa "Active-HDL^{TM} FSM", resaltando sus principales características y mostrando los pasos a seguir para la realización de una máquina de estados finitos.

Palabras clave.

Active-HDL, FSM, VHDL, diagramas de estado, máquinas de estado finitos.

Abstract.

The purpose of this paper is to give an introduction to "Active-HDL^{TM} FSM", highlights their principal characteristics and showing their requirements steps to realize a finite state machine.

Keywords.

Active-HDL, FSM, VHDL, state diagrams, finite state machines.

Introducción.

El programa Active-HDL^{TM} FSM, que se encuentra dentro de Warp^{TM} desarrollado por Cypress^{TM}, es una herramienta gráfica para la realización de las máquinas de estado finito (FSM), cuya principal ventaja es generar códigos en VHDL y en Verilog a partir de los diagramas de estado gráficos introducidos. Permite la creación de múltiples máquinas, se pueden utilizar señales registradas (secuenciales con FF) y señales combinacionales. Dichos códigos pueden ser sintetizados dentro de los PLD o CPLD que soporta Cypress^{TM} utilizando el programa Warp^{TM}. El Active-HDL^{TM} FSM se puede llamar del entorno Galaxy^{TM} (Tools) o bien desde la carpeta de Warp^{TM}.

El editor de estados permite la creación de cualquier tipo de máquina de estados a través de su interfaz gráfica de usuario (GUI) completamente amigable con representación de diagramas de burbujas.

Tipos de máquinas de estado.

Generalmente existen tres tipos de máquinas de estado: Mealy, Moore y mixtas. Siendo sus principales características que las salidas de una máquina de Mealy dependen del estado en que se encuentran y de sus entradas, mientras que las máquinas de Moore dependen sólo del estado en que se encuentra. Y las máquinas mixtas combinan estas características. Ambas máquinas pueden tener salidas combinatorias o registradas, ser asíncronas o síncronas, trabajar con flanco de subida o de bajada.

Desarrollo de un nuevo diseño con la herramienta Active-HDL FSM

La forma más sencilla de crear un diseño es utilizando el asistente (Design Wizard), que muestra paso a paso las declaraciones de la máquina de estados. Cuando se abre el Active-HDL FSM desde windows o desde Galaxy, lanza automáticamente el editor de estados (ver figura 1), que al aceptar (OK) da paso al asistente (ver figura 2). En éste hay que seleccionar el lenguaje (VHDL o Verilog), el nombre, los puertos y el número de máquinas para nuestro diseño.

Figura 1: Ventana del editor de estados.

Figura 2: Asistente para la creación de la máquina de estados.

Lenguaje VHDL o Verilog El editor puede generar dos tipos de código, VHDL y Verilog (ver figura 3), ambos aceptados por la IEEE (1164 para VHDL y 1364 para Verilog) y sintetizados por Warp.

Nombre del archivo

En la figura 4, se muestra la ventana en donde hay que ponerle nombre a la máquina que se diseña, así como el lugar donde se guardará.

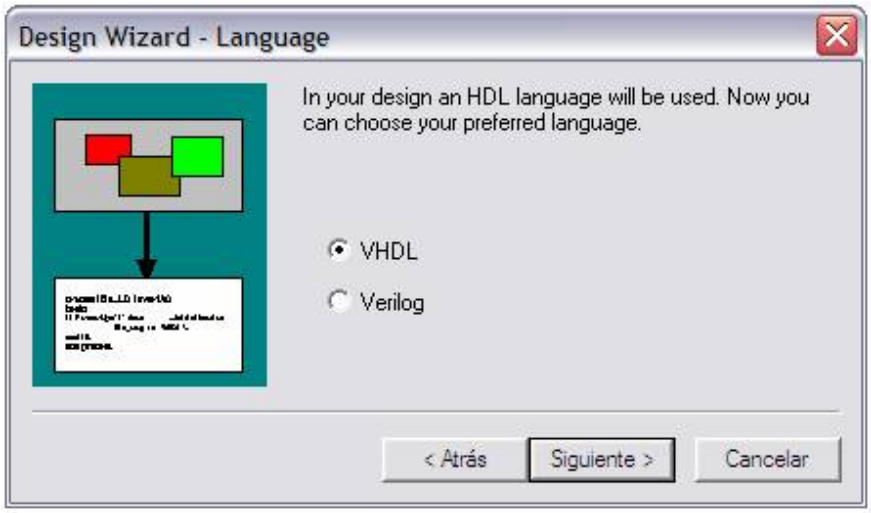

Figura 3: Ventana para seleccionar el lenguaje a ser utilizado.

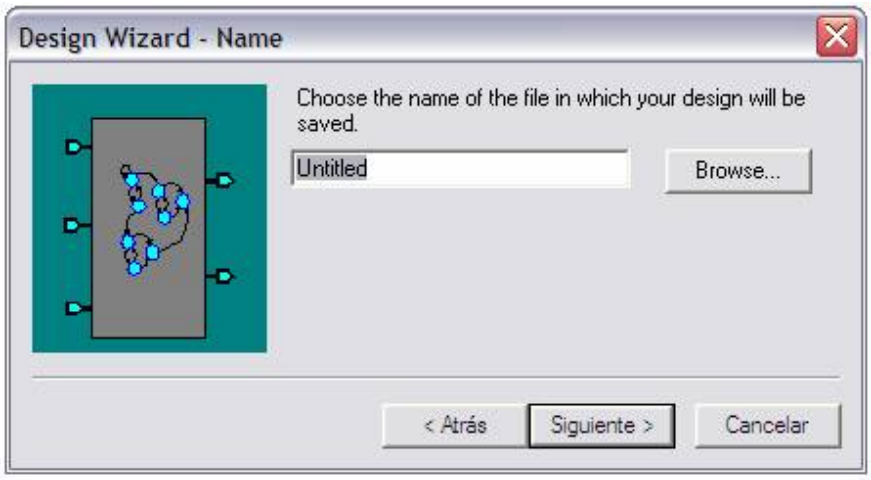

Figura 4: Nombre de la máquina y selección del directorio a guardar.

Puertos

En la figura 5 se muestra la ventana en donde se colocarán los nombres de los puertos de entrada, salida y los bidireccionales, así como la declaración de los buses. Utilizando el botón “new” y escribiendo en el área de texto se declaran los nombres de los puertos, seleccionando el puerto deseado (entrada, salida o bidireccional), así como su tamaño si es tipo bus. Con el botón “Advanced…” se puede seleccionar la señal de reloj (para entradas) y si se desea salida registrada o combinatoria.

Número de máquinas

En la figura 6 se selecciona el número de máquinas a trabajar. Por default, siempre marca una.

Una vez finalizado el asistente, aparece una hoja en blanco, donde hay que hacer uso de los botones de la figura 7 para colocar los estados (círculos), las transiciones (flechas), las condiciones (variables de entrada), las acciones (variables de salida), etc., que irán a definir nuestro diagrama de estados completo. Todos los botones son activados y ejecutados con un clic del ratón.

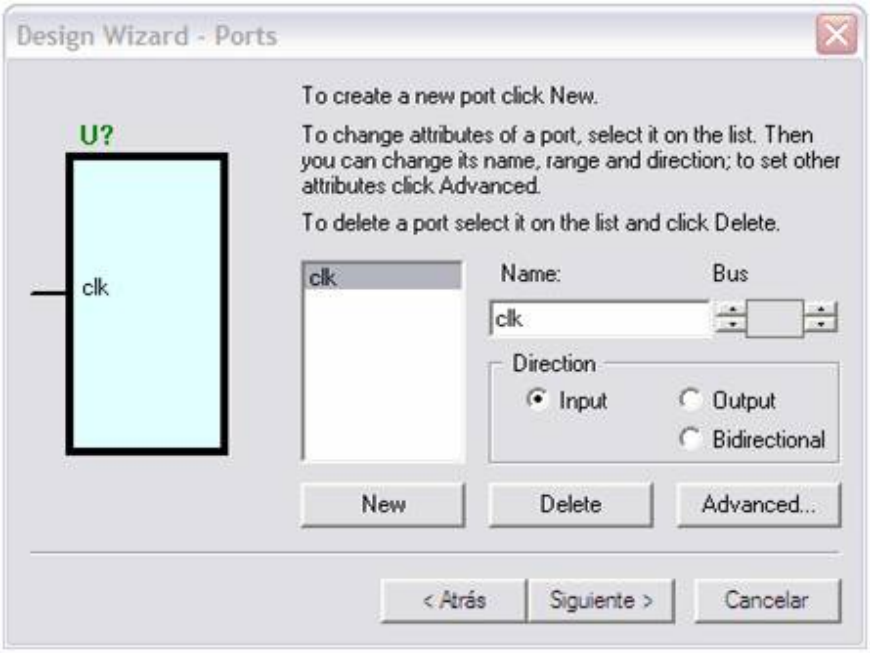

Figura 5: Selección de los puertos y sus propiedades.

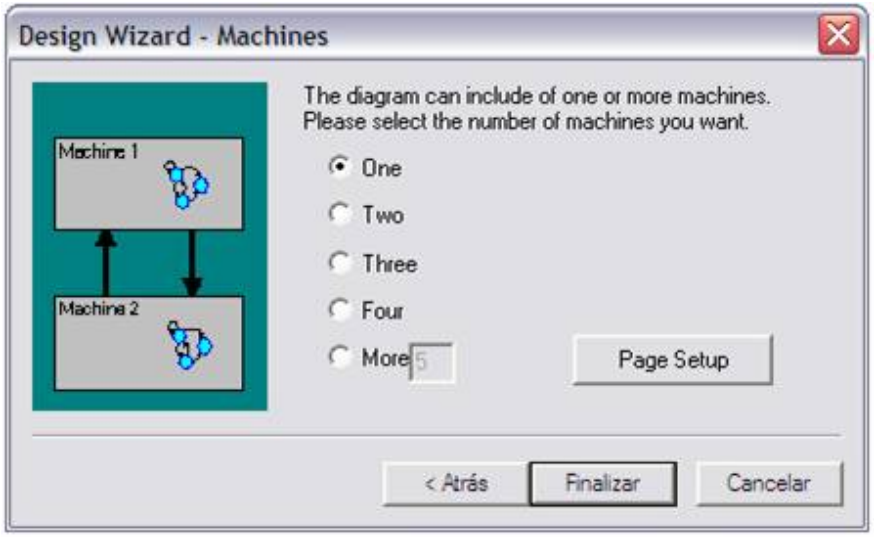

Figura 6: Selección del número de máquinas deseadas.

En la figura 8 se tiene un ejemplo con máquina Moore síncrona con reset, que soluciona el problema de encendido y apagado de un motor por medio de un botón (tipo push boton), con salidas registradas.

Una vez terminado el diagrama, para la generación del código en VHDL hay que seleccionar Synthesis¿HDL Code Generation o CTRL+H y contestar que se desea ver el código generado, si es que no existen errores (ver listado abajo "sin comentarios"). Si se desea generar el código en Verilog hay que cambiar la configuración en Synthesis¿Configuration. Cabe mencionar que el editor no contiene revisor de sintaxis, y todos los errores hay que revisarlos en el Galaxy.

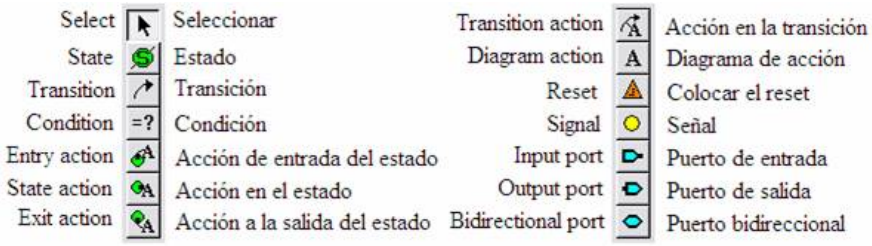

Figura 7: Botones para diseñar la máquina de estados.

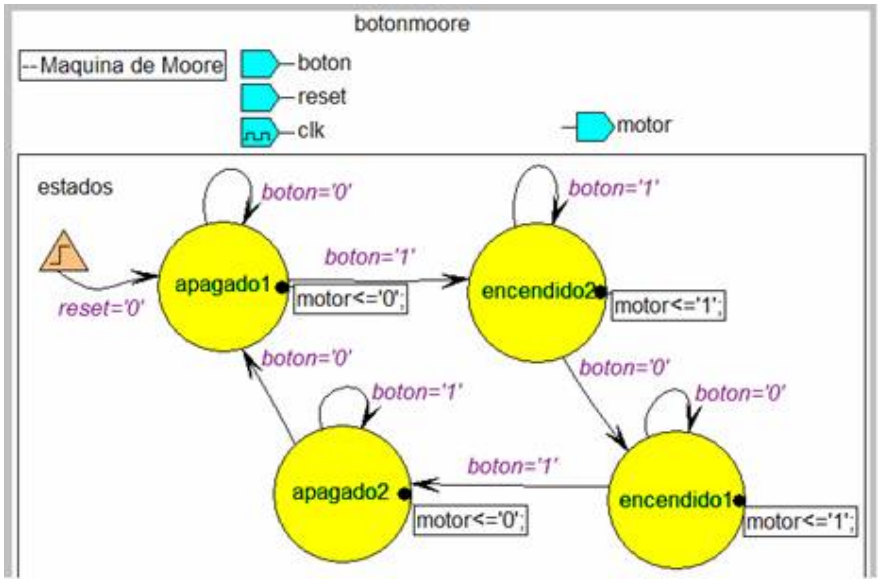

Figura 8: Ejemplo de control de encendido y apagado de un motor por medio de un botón, utilizando máquina de Moore. Tres entradas (botón, reset y clk), una salida (motor)

–Listado de VHDL del motor activado por botón, con máquina de Moore, sin comentarios.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity botonmoore is
port (boton: in STD_LOGIC;
clk: in STD_LOGIC;
reset: in STD_LOGIC;
motor: out STD_LOGIC);
end;
architecture botonmoore_arch of botonmoore is
type estados_type is (apagado1, apagado2, encendido1, encendido2);
signal estados: estados_type;
begin
estados_machine: process (clk) begin
if clk'event and clk = '1' then
if reset='0' then estados ¡= apagado1;
else
case estados is
when apagado1 =¿motor ¡= '0';
if boton='1' then estados ¡= encendido2;
elsif boton='0' then estados ¡= apagado1;
end if;
when apagado2 =¿motor ¡= '0';
if boton='1' then estados ¡= apagado2;
elsif boton='0' then estados ¡= apagado1;
end if;
when encendido1 =¿motor ¡= '1';
if boton='0' then estados ¡= encendido1;
elsif boton='1' then estados ¡= apagado2;
end if;
when encendido2 =¿motor ¡= '1';
if boton='1' then estados ¡= encendido2;
elsif boton='0' then estados ¡= encendido1;
end if;
when others =¿null;
end case;
end if;
end if;
end process;
```

end botonmoore_arch;

Conclusiones

Se ha mostrado una breve introducción al editor de máquinas de estado Active-HDL FSM, que puede ser utilizado para la creación de distintas máquinas. Pueden obtenerse códigos para VHDL y para Verilog, y utilizarse en el Galaxy de Warp. 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™ FSM – AN1016 (12, Nov, 2001), www.cypress.com

[3] 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 proyecto CGPI20061916