

DISEÑO Y SIMULACIÓN DE CBD CON ALLIANCE

D. PEÑALOSA, C. BAENA, M.P. PARRA y M. VALENCIA

Dpto. de Tecnología Electrónica. E.T.S. Ing. Informática (Ed. Rojo), Univ. Sevilla Avd^a. Reina Mercedes s/n Sevilla. . 41012 España.

Se presenta el diseño y simulación del procesador específico para bases de datos (CBD) usando un entorno de diseño de libre distribución: ALLIANCE. Dada la complejidad del procesador, se han desarrollado dos herramientas que facilitan la generación de componentes y la simulación del sistema. Con ellas se simplifica la realización de otros sistemas digitales complejos en ese entorno.

1. Introducción

En este artículo se presenta la forma en la que se ha validado un computador específico para base de datos (CBD), cuya organización y descripción funcional se explica en otro trabajo que presentamos en el TAAE'2002 titulado "Organización de un computador específico para base de datos (CBD)". El computador CBD es un sistema digital suficientemente complejo como para requerir un entorno de diseño CAD que permita su adecuada especificación y simulación. En este trabajo nosotros exploramos el uso de ALLIANCE [1] para el diseño y simulación de CBD. A pesar de que no está orientado al diseño de sistemas digitales tan complejos como CBD, el paquete ALLIANCE ha sido elegido por ser un entorno de diseño bastante completo, que abarca todos los pasos del proceso de diseño incluyendo la generación del *layout*, y por ser de distribución libre. Con el fin de facilitar diseños complejos hemos desarrollado tres herramientas compatibles con este entorno: una para especificar circuitos, otra para generar patrones de simulación de ALLIANCE y otra para ver los resultados de simulación, aunque las dos últimas van a ser tratadas como una sola.

En el siguiente apartado veremos cómo hemos diseñado CBD con ALLIANCE, tratando sucesivamente sus dificultades para realizar circuitos grandes, la herramienta GENVHDL.H que hemos desarrollado y algunos ejemplos de su aplicación al diseño de CBD. Posteriormente nos centraremos en la simulación describiendo primero las herramientas utilizadas y, después, algunas pruebas de simulación. Por último, presentamos nuestras conclusiones.

2. Diseño estructurado de CBD con ALLIANCE

El objetivo de este apartado es mostrar la forma de especificar el procesador CBD. Para ello en los siguientes subapartados se explicará, en primer lugar, por qué VHDL de ALLIANCE es inadecuado para especificar procesadores complejos; en segundo lugar, la herramienta

GENVHDL.H que hemos desarrollado para salvar esa inadecuación; y, por último, veremos un ejemplo de cómo se utiliza esta herramienta.

2.1. ALLIANCE y sus limitaciones

ALLIANCE es un entorno de aplicaciones y archivos de texto, que permite especificar en VHDL, simular y generar información adecuada para la creación de sistemas integrados. El problema que plantea el lenguaje VHDL de ALLIANCE a la hora de especificar computadores, es que es demasiado simple y funcional. Es simple porque el conjunto de instrucciones es el mínimo requerido a la hora de especificar circuitos, pues todas las operaciones se realizan partiendo de una señal de un bit: almacenamiento, bifurcaciones, alta impedancia, operaciones lógicas y declaraciones de señales. Estas condiciones hacen que este VHDL no se adecue a nuestro proyecto, pues especificar un computador de gran complejidad a partir de señales de un bit puede llegar a ser no sólo tedioso en exceso, sino también fuente de innumerables errores a la hora de su realización. Esto conlleva la necesidad de disponer de una forma estructurada para realizar las especificaciones y con las que podamos descomponer el sistema a estudiar en módulos más simples.

2.2. La herramienta “GENVHDL.H”

Aunque ALLIANCE presenta una librería de componentes, estos no son lo suficientemente complejos puesto que para nuestras necesidades requerimos registros con operaciones complejas como las de carga, incremento, desplazamientos, etc. Para superar esta dificultad, se ha desarrollado una herramienta que, utilizando el lenguaje “C”, permite especificar para ALLIANCE cualquier tipo de circuito de forma simple, clara y estructurada.

La herramienta que hemos desarrollado, denominada GENVHDL.H, consta de una librería de funciones del lenguaje “C”, que genera código VHDL para ALLIANCE, como se puede apreciar en la figura 1. Cada una de estas funciones se corresponde con un dispositivo electrónico (registros, memorias, multiplexores, sumadores,...), de forma que para cada uno de ellos se genera el correspondiente código VHDL. Esta forma de concebir la herramienta tiene una doble ventaja, pues: en primer lugar, a partir de una especificación estructural vamos a obtener una especificación funcional susceptible de ser simulada; y en segundo lugar, vamos a tener una especificación más clara sin perder flexibilidad.

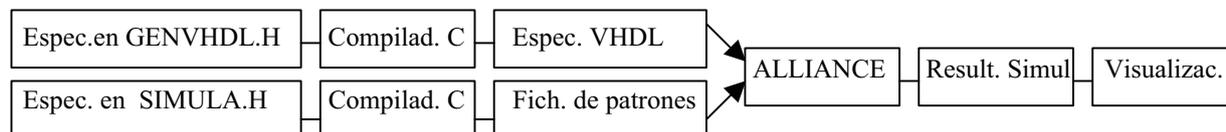


Figura 1: *Proceso para la creación y visualización de una prueba con las herramientas dadas.*

2.3. Diseño de CBD. Ejemplo.

En el siguiente ejemplo, vamos a ver cómo se especificaría un registro conectado a un bus:

```
REGISTRO("Reg1",32,CARGA,"LD","Cont1");  
BUSHI("Bus1","OE","Reg1");
```

Con la directiva REGISTRO, estamos indicando a la herramienta GENVHDL.H, que nos especifique en VHDL un registro, de nombre "Reg1", que sea de 32 bits, y que lo cargue con el contenido de "Cont1" en el flanco de subida de la señal "LD".

Con la directiva BUSHI se conecta el registro "Reg1" con el bus "Bus1". La información de este registro aparecerá cuando la señal "OE" se active en un nivel alto.

3. Simulación

Con la simulación podemos comprobar que el circuito se ajusta a las características preestablecidas en el diseño. Para esto ALLIANCE incluye un paquete de utilidades para la simulación y validación. Estas utilidades tratan al circuito como una caja negra; es decir, dadas unas entradas y salidas especificadas en un fichero de código VHDL y otro fichero con el valor de estas entradas, ALLIANCE tomará ambos ficheros y calculará los valores de las salidas. Cada conjunto de entradas se puede especificar tantas veces como se quiera. A cada conjunto de valores de estas entradas se le denomina patrón, el cual equivale a medio ciclo de reloj. Como para simular las instrucciones de una CPU necesitamos muchos ciclos de reloj, hemos de tener una herramienta que de forma fácil genere estos ficheros de patrones y que muestre de forma clara y elegante el resultado de la simulación. Pasemos a ver en qué consiste, en nuestro caso, esta herramienta, para explicar posteriormente cómo se utiliza sobre la generación del lote de pruebas de CBD.

3.1. Herramientas utilizadas para la simulación de CBD

La primera herramienta (figura 1) se trata de "SIMULA.H", la cual tiene un funcionamiento parecido a "GENVHDL.H". Esta herramienta consta de una serie de funciones del lenguaje "C" puestas de forma consecutivas en una función "main", cada cual es una directiva que construye una parte del fichero de patrones. Así, con "ENTRAD("Clk",1)" estamos indicando que el fichero de patrones tenga una señal de entrada llamada "Clk" de un bit de anchura, y con "SALID("obrDI,32)", estamos indicando que nos devuelva el valor del registro interno obrDI de 32 bits. De todas las funciones desarrolladas, merece destacar "ACTIVA_CLK(n,"snl")", cuya misión es la de generar los patrones necesarios para que la señal "snl" ejecute "n" ciclos de reloj.

Una vez obtenido el archivo de patrones con SIMULA.H, al ejecutar ALLIANCE se genera un archivo de texto con el valor de las señales de salida. Para ver este fichero de forma más ordenada y clara (ciclo a ciclo), lo podemos hacer empleando la función SIMULA() de la herramienta SIMULA (insertándolo al final del código, en la especificación del archivo de patrones), o bien, empleando la herramienta "SVHDL". Ambos programas se utilizan de la misma forma. La diferencia que hay entre ambos es que, mientras la función "SIMULA()" es

genérica para cualquier tipo de circuito, SVHDL está pensado para emplearse con CBD, mostrando así los registros más relevantes de este.

3.2. Pruebas del conjunto de instrucciones de CBD

Para poder probar nuestra CPU con las herramientas indicadas en el apartado anterior, hemos de conectarlas con un sistema de memoria (de código y datos) y de entrada/salida para que las instrucciones puedan transferir datos. Esto se especificará con la herramienta “GENVHDL.H” y se empleará para la simulación de todas las instrucciones.

Para cada instrucción, con la herramienta “SIMULA.H” se generará un archivo con el código de la instrucción en la primera posición de la memoria de código destinada a programas; con ACTIVA_RST(“Rst”) se hará un reset, y con ACTIVA_CLK(n,“Clk”), se indicará el número de ciclos de reloj que consume la instrucción. Cuando se ejecute este programa “C”, se generará un archivo de patrones que, junto con el de la especificación VHDL de la CPU con su sistema de memoria, se simulará con ALLIANCE para obtener un archivo con el resultado de la simulación. Este archivo, por último, se podrá inspeccionar detalladamente con “SVHDL”. En la figura 1 se puede ver reflejado el proceso llevado a cabo para la validación de las instrucciones:

4. Conclusiones

En este artículo hemos presentado una serie de herramientas que permiten la descripción y simulación de sistemas digitales complejos usando el entorno de diseño ALLIANCE y lo hemos aplicado al procesador CBD. Estas herramientas se han desarrollado en lenguaje C con el fin de facilitar la generación de ficheros de diseño y simulación.

La herramienta GENVHDL.H se puede considerar como un generador de componentes de nivel de complejidad media (registros, sumadores, multiplexores, memorias, etc.) resultando fácil de interconectar en la especificación. La herramienta posee una gran flexibilidad a la hora de realizar sistemas complejos a nivel RT ya que da mayor claridad a las descripciones de los módulos y, al mismo tiempo, evita una documentación propensa a errores propia de la descripción de módulos demasiado simples de VHDL-ALLIANCE. Por su parte, con SIMULA.H es fácil crear los ficheros de patrones de simulación, incluso en el caso de gran número de señales. Con estas herramientas, hemos diseñado y verificado por simulación el procesador CBD.

En conclusión este trabajo muestra que, efectivamente, el entorno ALLIANCE, así modificado, puede ser usado como entorno de diseño para sistemas digitales bastante complejos.

Referencias

- [1] Laboratoire MASI/CAO-VLSI Université Pierre et Marie Curie. *First Steps in Using ALLIANCE: The Addaccu Tutorial*. Edita Université Pierre et Marie Curie (1993).
- [2] D. Peñalosa. *Diseño de un Procesador Específico para Bases de Datos*. Proyecto Fin de Carrera (1998).
- [3] H. Schildt. *Programación en Turbo C*. Editorial Anaya (1989).