

VISIÓN INVESTIGADORA
Clasificación del artículo:
INVETIGACIÓN

DISEÑO DE CIRCUITOS VLSI A PARTIR DE VHDL

Jaime Alberto Parra Plaza*
Andrés Farfán**
Rafael Mauricio Becerra Peña***

Resumen

Este documento presenta los resultados obtenidos durante el diseño de un microprocesador RISC de 16 bits utilizando la herramienta Alliance CAD System. El objetivo de la investigación se centró en validar las capacidades de la herramienta para la obtención del conjunto de máscaras de fabricación, a partir de un conjunto de descripciones dadas en un subconjunto del lenguaje para descripción de hardware VHDL.

Palabras clave
ALLIANCE, Circuito integrado, CMOS, Microprocesador, RISC, VHDL, VLSI.

Summary
This paper summarizes the results obtained during the design of a 16-bit RISC microprocessor using the Alliance CAD System tool. The purpose of this research pointed out to validate the capacities of the tool to obtain the set of fabrication masks from a set of descriptions given in a subset of the hardware description language VHDL.

Keywords
ALLIANCE, Integrated Circuit, CMOS, Microprocessor, RISC, VHDL, VLSI.

1. Introducción

El diseño de circuitos de alta escala de integración (VLSI) tradicionalmente parte del uso de herramientas con las que el diseñador trabaja al nivel de capas de silicio, para generar las máscaras de fabricación, más o menos de la misma manera como un programador de microprocesadores puede trabajar al nivel del lenguaje ensamblador. Ejemplos de este tipo de herramientas pueden ser, en el dominio público, Magic o LASI, y como herramientas comerciales, Cadence, Mentor Graphics, Synopsis o Tanner [3]. Alliance es un conjunto de herramientas, algunas públicas otras comerciales, para el diseño de circuitos integrados que basa su metodología en la síntesis

* Docente de la Facultad de Ingeniería en la Pontificia Universidad Javeriana Cali, ,Ingeniero Electricista de la Universidad del Valle en 1991 y Magíster en Automática de la misma Correo electrónico: jparra@puj.edu.co.

** Ingeniero Electrónico de la Pontificia Universidad Javeriana Cali. Estudiante de Doctorado en Microelectrónica, Universidad de Sao Paulo, Brasil. Correo electrónico: nafraf@linuxmail.org.

*** Ingeniero Electrónico de la Pontificia Universidad Javeriana Cali. Correo electrónico: rafabe_vlsi@mail.com.

de configuraciones de máscaras, a partir de descripciones de hardware en VDHL y C (Alliance CAD System, Avertec Technologies).

El propósito es, continuando con la analogía anterior, diseñar un sistema digital a partir de una descripción comportamental, una abstracción en alto nivel de la función y estructura del sistema, y a partir de una herramienta de *compilación en silicio,* llegar a la descripción en bajo nivel del sistema que, en este caso, es el conjunto de máscaras adecuadas para ordenar la fabricación del circuito integrado.

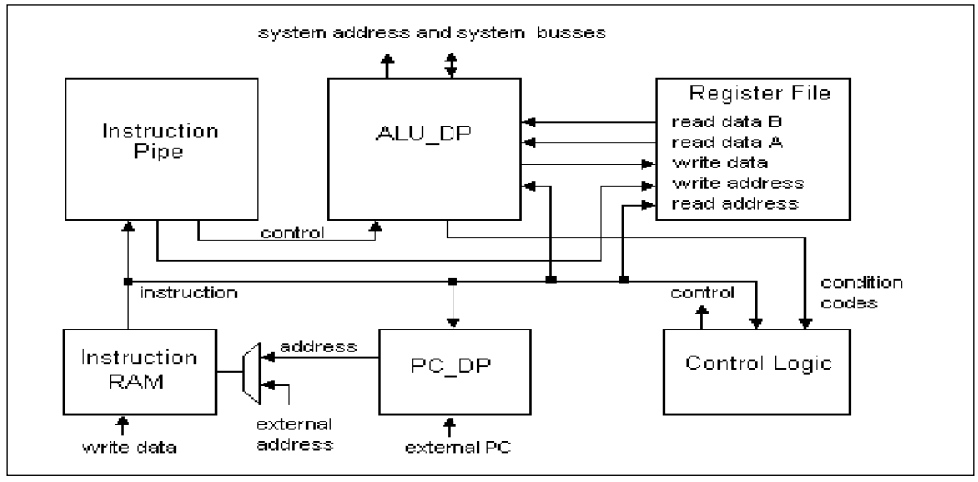
Para explorar las características de la herramienta y documentar claramente el proceso de diseño, se implementó un microprocesador RISC de 16 bits diseñado por B. Edwards y N. Weste (1994). El diseño parte de la especificación del conjunto de instrucciones que soportará el microprocesador, con base en las cuales se definen los bloques principales que constituirán el dispositivo. Posteriormente, se detallará la funcionalidad lógica de cada uno de estos bloques que se implementan aprovechando los generadores y sintetizadores que el sistema ofrece. El producto final de la investigación es un archivo de descripción de máscaras en un formato estándar de la industria (CIF), así como las conclusiones a las cuales los autores llegaron con el desarrollo de la investigación.

2. Arquitectura del microprocesador

El microprocesador consta de seis bloques principales: una Unidad Aritmético-Lógica (ALU_DP) que es la encargada de ejecutar las instrucciones; un conjunto de registros de trabajo (REGFILE), en el que se almacenan los operandos con los cuales la ALU trabaja; un contador de Programa (PCDP) con su respectiva circuitería de actualización, que sirve para calcular y almacenar la dirección de la siguiente instrucción a ejecutar; una Tubería de Instrucciones (INSTPIPE) en la cual las diferentes etapas en que se ejecuta una instrucción se van haciendo en paralelo y desplazadas para varias instrucciones al tiempo; una memoria de lectura y escritura (INSTRUCTION RAM), en la que se almacenan las instrucciones que el procesador realiza; una Unidad de Control (CONTROL) que decodifica la instrucción actual y genera las señales de control respectivas para las otras unidades.

El procesador cuenta con un pipe de cuatro etapas: decodificación de instrucción (I), lectura (R), ejecución (E) y escritura (W.) El esquema del microprocesador se presenta en diagrama 1.

Diagrama 1. Diagrama de bloques del procesador



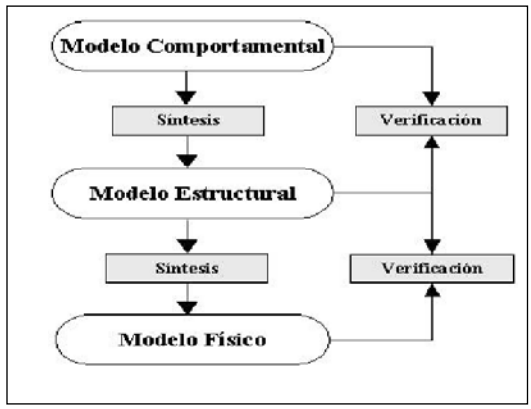
3. Flujo de diseño

El diseño parte de una descripción comportamental que es procesada por diferentes herramientas de síntesis hasta llegar a un modelo al nivel físico. El modelo comportamental describe la funcionalidad del sistema a través de descripciones de alto nivel, como condiciones *if*, selectores tipo case, bucles *for*.

Además de la funcionalidad, el modelo permite jerarquizar los diferentes módulos del sistema mediante una descripción estructural, en la que se establece qué módulos son internos a qué otros módulos. Ambos tipos de descripciones pueden hacerse en Alliance utilizando una mezcla de archivos escritos en VHDL y en C; para ambos casos, la herramienta soporta un conjunto limitado de la potencialidad plena de los lenguajes; en particular, se da soporte a las instrucciones y estructuras que puedan ser directamente sintetizables en hardware.

Después de ingresada esta descripción en alto nivel, el conjunto de herramientas la traduce sucesivamente a una descripción estructural y luego a una descripción física. El proceso de síntesis es acompañado por etapas de verificación entre los diferentes niveles de abstracción, que permiten validar violaciones a reglas de diseño, tanto en funcionalidad como en temporalidad, como en geometría. El esquema general de dicho proceso se muestra en el diagrama 2.

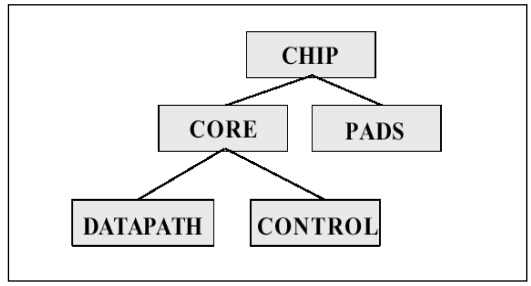
Diagrama 2. Flujo de diseño



4. Implementación del microprocesador

La concepción del diseño parte de una metodología en parte funcional y en parte estructural [4]. En primer lugar, se asume que el diseño va a ser sintetizado en un circuito integrado (el CHIP), así que se hace una división grande entre el circuito en sí mismo (el CORE) y los terminales que conectarán sus líneas de entrada y de salida con el exterior (o PADS.) El diseño del Core asume una división funcional [5] entre una unidad combinatoria que ejecuta las distintas labores (el DATAPATH) y una unidad secuencial que escoge qué operación se hace, en qué momento, con qué operandos y en dónde se almacena su resultado (el CONTROL). La creación del microprocesador sigue la estructura jerárquica mostrada en diagrama 3.

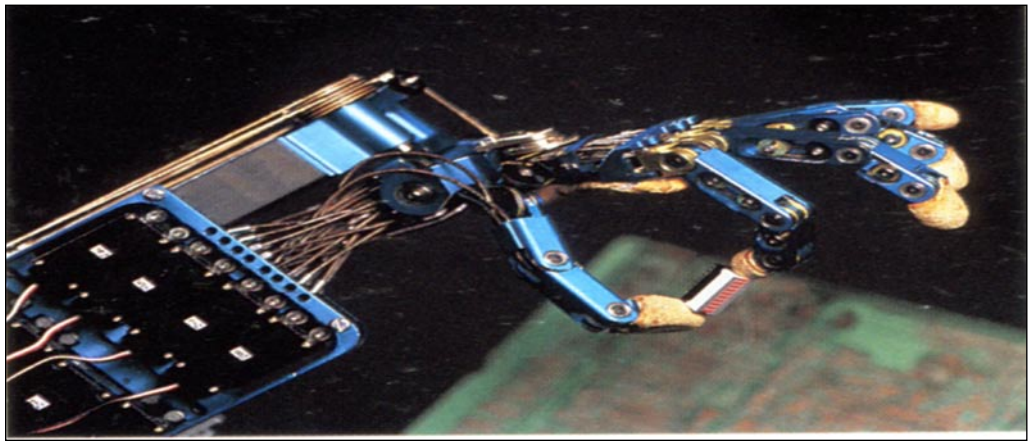
Diagrama 3. Estructura jerárquica del chip



El datapath comprende las unidades operativas del microprocesador, siendo éstas un conjunto de operadores que manejan los datos organizados a manera de buses. Los operadores fueron creados por herramien-

tas de generación automática o instanciando componentes parametrizados de una biblioteca de operadores. Cada operador es un conjunto de n celdas idénticas apiladas, cada una de ellas dedicada a la modificación de un solo bit.

La unidad de control del microprocesador reúne la lógica que rige el comportamiento del datapath. El control es una estructura que no posee arreglos regulares de operadores, su síntesis se hace sobre celdas estándar, a partir de descripciones en VHDL. El núcleo del circuito instancia e interconecta las unidades operativa y de control. La conexión es realizada automáticamente, con base en un archivo estructural y los modelos físicos de los bloques a conectar.



Por último, se crea el chip que comprende el núcleo del circuito y los pads de conexión con el exterior. El archivo se genera a partir del modelo físico del núcleo, un archivo estructural en el que se asocian los pads a los puertos del circuito y los modelos físicos de los pads tomados de una biblioteca.

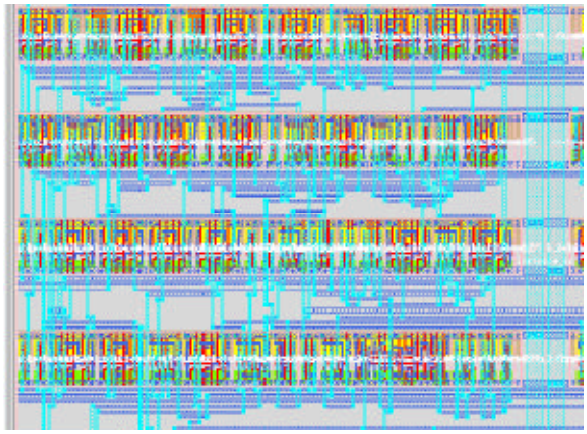
Alliance genera modelos físicos lógicos basados en unidades lambda, estos modelos serán dimensionados con unidades reales en la última etapa del proceso en el que se realiza la conversión de modelo simbólico a un modelo real, archivo en formato CIF que puede ser enviado a fabricación.

Cabe recordar que para el diseño de circuitos integrados se suele utilizar el concepto de *celda estándar* que es un pequeño circuito, generalmente compuesto de algunos transistores que funcionan de manera equivalente a un pequeño circuito combinatorio (por ejemplo un arreglo de dos compuertas AND seguido por una compuerta OR, o tres compuertas OR en cascada, etc.) que se usan como bloques constitutivos de un circuito más grande.

La ventaja de usar celdas estándar en lugar de diseñar el circuito de manera plana (partir del transistor o del par complementario como elemento primitivo) es doble: en primer lugar, las celdas estándar han sido diseñadas cuidadosamente por expertos para garantizar uniformidad en sus parámetros, como tiempos de respuesta, corriente drenada, capacitancia parásita, etc.; en segundo lugar, el tamaño de todas las celdas estándar se hace uniforme y la ubicación de los buses de potencia y de señal se hace siempre en los mismo lugares, de manera que cuando se unan unas celdas con otras naturalmente queden unidos los buses de potencia y sea fácil unir los buses de señales.

El uso de celdas estándar se combina de manera óptima con la metodología Bit Slice en la cual se elabora una unidad funcional que actúa sobre un solo bit. Ante una pa-

Figura 1. Sección de la unidad de ejecución

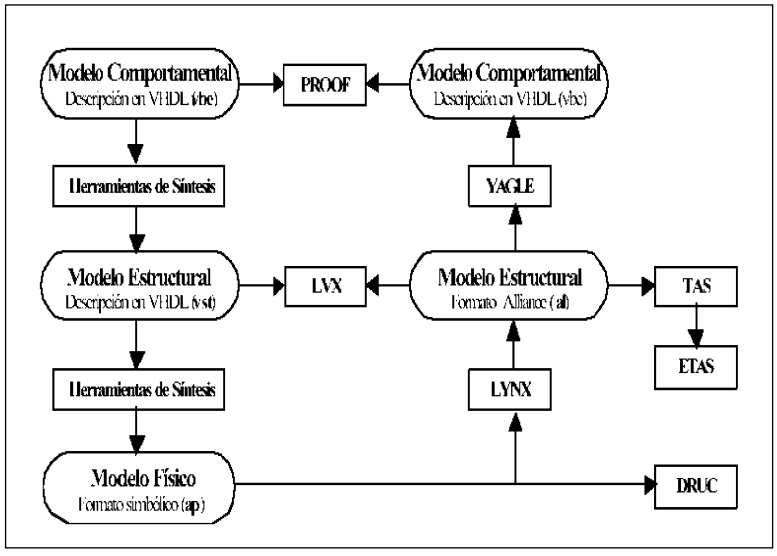


labra de datos de n bits, simplemente se generan n instancias de la unidad base; esta concepción permite que el diseño no tenga que cambiarse drásticamente ante cambios en el número de líneas de datos. El patrón repetitivo que se produce en la geometría del circuito es una consecuencia natural de esta metodología, lo que puede evidenciarse para una sección de la *unidad de ejecución* en la figura 1.

Otra consideración es las unidades *lambda* en las que se suelen expresar todas las medidas de un circuito integrado en su fase de diseño. La razón es facilitar esta fase y postergar para la fase última la elección de una determinada tecnología (método de fabricación) con la que se generará el circuito físico como tal. Dado que las tecnologías que usa cada fabricante suelen ser incompatibles, el hecho de tener unidades adimensionales posibilita la fácil comparación entre una implementación y otra sin tener que cambiar en absoluto el diseño.

El proceso de síntesis debe ir acompañado de etapas de verificación que garanticen que los modelos entregados son correctos. La validación se realiza en los diferentes niveles del proceso; esta tarea involucra el uso de herramientas de extracción y comparación de descripciones. El esquema general del proceso se muestra en diagrama 4.

Diagrama 4. Análisis y verificación



A nivel físico se verifica que no existan violaciones a las reglas de diseño, tarea efectuada por Design Rule Checker (DRUC). Una regla de diseño especifica aspectos geométricos, como la mínima distancia que debe existir entre un componente y otro, el área que debe ocupar un elemento, ya sea mínima o exacta, la necesidad de que dos capas coexistan en determinado sitio, para que tenga sentido funcional esa construcción, etc. Al nivel estructural se extrae un modelo a partir de la descripción física y se realiza una comparación con el modelo estructural inicial con la herramienta LVX. Para realizar la verificación a nivel comportamental es necesario extraer un modelo comportamental con YAGLE y

realizar una prueba formal con el modelo inicial con la utilidad PROOF.

Es posible hacer un análisis de los tiempos de respuesta esperados del circuito integrado con las herramientas TAS y ETAS. TAS entrega una lista detallada de los retardos entre diferentes puntos del circuito; el análisis de dicho archivo se realiza con la utilidad ETAS. Alliance incluye también herramientas para la simulación de circuitos descritos en VHDL estructural y comportamental. El programa ofrece utilidades para la generación de patrones de prueba (GENPAT), la simulación misma (ASIMUT) y el despliegue en modo gráfico de las respuestas (XPAT.) Cada uno de los bloques generados y el microprocesador final fueron simulados para corroborar que cumplen con la especificación inicial.

5. El sistema microprocesado

Una vez diseñado y verificado el correcto funcionamiento de cada parte del microprocesador se procedió a integrarlos en un solo bloque. Para efectos de probar el microprocesador como una unidad, se hace necesario añadirle una memoria en la que se pueda almacenar el programa. El conjunto de microprocesador y memoria es lo que se denomina aquí *sistema microprocesado*.

Para facilitar las labores de depuración se creó un ensamblador específico para el microprocesador diseñado y con ayuda de una herramienta de Metaensamblaje (Team Fire Development Group) se escribieron diversos programas que verificaran las diversas partes del procesador, tanto individualmente (ALU, Register File, Pipeline, PC) como en conjunto e interactuando con la memoria.

Programas como la generación de la serie de Fibonacci, el ordenamiento de un conjunto de números, la obtención de la parte entera de las raíces de una ecuación cuadrática, la comparación de dos cadenas, etc. fueron exitosamente ejecutados por el microprocesador, a través de las herramientas de simulación de Alliance.

6. Resultados

El proyecto concluye con el modelo al nivel de máscaras del microprocesador RISC, acompañado de su modelo comportamental en VHDL, un programa ensamblador para la generación de memorias para prueba funcional del circuito y todos los archivos para la generación y prueba de cada uno de sus bloques. Dentro de los programas de prueba se incluyen algoritmos que verifican que las instrucciones aritméticas, lógicas y de control de programa funcionan correctamente; además de un conjunto de programas que ejecutan diversos algoritmos de complejidad mediana. Las características del microprocesador se resumen en la tabla 1.

Tabla 1. Características del RISC

Arquitectura	RISC
Pipeline	4 etapas
Longitud Instrucción	32 bits
Bus de datos	16 bits
Tecnología	1 micra
Frecuencia de Operación	30 MHz
Número de Transistores	51350 t
Dimensiones Chip	15.2x4.8 mm
Dimensiones Núcleo	13.0x2.6 mm
Dimensiones Datapath	13.0x1.6 mm
Dimensiones Control	7.5 x 0.5 mm

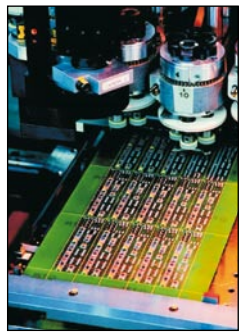
El proyecto concluye con el modelo al nivel de máscaras del microprocesador RISC, acompañado de su modelo comportamental en VHDL, un programa ensamblador para la generación de memorias para prueba funcional del circuito y todos los archivos para la generación y prueba de cada uno de sus bloques.

Dicha tabla permite comparar el microprocesador diseñado contra microprocesadores comerciales y contra otros diseños realizados con el sistema Alliance por diversas universidades a lo largo del mundo [1] y comprobar que el diseño es de una calidad entre buena y excelente. El trabajo y comentarios de los autores conllevaron una revisión a las herramientas LYNX, GENPAT y DPR. En algunos casos la solución a los problemas fue dada por los desarrolladores de Alliance y en otros fue tarea de los autores escribir programas que corrigen errores en los archivos generados por algunas herramientas. El programa de mayor relevancia soluciona los errores estructurales generados por FPGEN al instanciar elementos de memoria.

7. Conclusiones

Se demostró que es posible implementar circuitos CMOS VLSI con Alliance. Aunque el programa es un paquete de software no comercial, el hecho de ser desarrollado por una institución educativa reconocida y tener más de diez años de existencia, garantiza respaldo para proyectos de investigación en el área de la microelectrónica.

Alliance es un paquete en constante desarrollo, razón por la cual sus herramientas pueden presentar diversos problemas. Esta situación puede solucionarse si se reportan dichos errores a los desarrolladores de la herramienta. Sin embargo, en algunos casos estos errores no son corregidos, lo que obliga al diseñador a crear sus propias soluciones, ya sea modificando los archivos fuente de distribución de Alliance o creando aplicaciones que corrijan los archivos defectuosos generados.



Alliance ofrece varios caminos para la implementación de circuitos CMOS VLSI; la ruta a seguir es flexible de acuerdo con el tipo de aplicación que se esté realizando y al formato de descripción inicial que el diseñador elija (C o VHDL). Sin embargo, este enfoque no garantiza que los circuitos entregados posean un rendimiento óptimo en tiempo y hagan un buen uso del área. Por lo tanto, cuando se desee implementar un circuito para fabricación, se deberá guiar la herramienta para las tareas de ubicación y conexión o inclusive editar las máscaras a mano.

Este trabajo permitió realizar un estudio al nivel exploratorio de la metodología de implementación con Alliance, lo que proporciona una primera aproximación que sirve como base para futuras investigaciones en el área. La implementación del microprocesador RISC exigió la utilización de gran parte de las herramientas de Alliance, pues el circuito requiere la síntesis de componentes sobre celdas estándar, la síntesis sobre estructuras de operadores (datapath) y el uso de los programas para la optimización, verificación y simulación de cada una de ellas.

El uso de Alliance no se limita únicamente a la implementación de circuitos lógicos VLSI. Varias de sus herramientas pueden ser utilizadas de manera individual para diferentes propósitos, desde la simulación de descripciones VHDL hasta las pruebas formales entre modelos comportamentales. Esta característica permite que Alliance sea una herramienta para uso estudiantil, útil en el aprendizaje

El uso de Alliance no se limita únicamente a la implementación de circuitos lógicos VLSI. Varias de sus herramientas pueden ser utilizadas de manera individual para diferentes propósitos, desde la simulación de descripciones VHDL hasta las pruebas formales entre modelos comportamentales

y práctica de asignaturas que involucren el diseño y validación de circuitos lógicos digitales.

Referencias bibliográficas

[1] *Alliance CAD System*. Extraído de la World Wide Web: http://www-asim.lip6.fr/recherche/alliance/

[2] *Avertec*. Extraído de la World Wide Web: http://www.avertec.com

[3] Becerra, R. y Farfán, A. (1999). *Evaluación de las herramientas L-Edit, Magic y Alliance para el diseño de circuitos VLSI.* Cali: Pontificia Universidad Javeriana.

[4] European Space Research and Technology Centre (1994). *Modelling Guidelines*. S.d.: The Netherlands.

[5] Nagle, T., Nelson, V. (1996). *Análisis y diseño de circuitos lógicos digitales.* México: Prentice-Hall.

[6] Patterson, D. y Hennesy, J. (1996). *Computer Architecture*. S.d.: Morgan Kaufmann.

[7] *Team Fire Development Group*. Extraído de la World Wide Web: http://users.ece.gatech.edu/~hamblen/book/wintim/

[8] Weste, N. y Eshraghian, K. (1994). *Principles Of CMOS VLSI Design*. S.d.: Addison-Wesley.