The background is a dark, textured surface composed of a grid of squares, some of which are slightly raised or recessed, creating a 3D effect. In the center, there is a glowing, ethereal image of a hand holding a small, bright object, possibly a flame or a light source. The overall aesthetic is high-tech and digital.

FPGAs: COMPUTACIÓN & APLICACIONES

Jordi Carrabina
Lluís Ribas
Oscar Navas
David Castells
(Eds.)

serva04

FPGAs: COMPUTACIÓN & APLICACIONES

*Artículos seleccionados de las IV Jornadas de
Computación Reconfigurable y Aplicaciones
Bellaterra, 13-15 de Septiembre de 2004*

*Jordi Carrabina
Lluís Ribas
Oscar Navas
David Castells*

Editores

**Centre de Prototips i Solucions HW/SW
Escola Tècnica Superior d'Enginyeries
Universitat Autònoma de Barcelona**

Título: FPGAs: COMPUTACIÓN & APLICACIONES

I.S.B.N.: 84-688-7667-4

Depósito Legal: 40.194-2004

Editores: J. Carrabina, Ll. Ribas, O. Navas, D. Castells
Centre de Prototips i Solucions HW/SW
Escola Tècnica Superior d'Enginyeries
Universitat Autònoma de Barcelona
08193 Bellaterra
cephis@cephis.uab.es
<http://cephis.uab.es>

Impresión: Format Digital S.L.
C/ Berlin 11-13, Baixos
08014 Barcelona
Tel: 934090005
formatdigital@telefonica.net

Reservados todos los derechos. Ni la totalidad ni parte de este libro puede reproducirse o transmitirse por ningún procedimiento electrónico o mecánico, incluyendo fotocopia, grabación magnética o cualquier almacenamiento de información o sistema de reproducción, sin permiso previo y por escrito de los titulares del Copyright.

8.1 Sistemas bioinspirados

- Estudio de Arquitecturas para Verificadores Hamming en un Sistema de Identificación Biométrica por Iris.
Liu J., Sánchez R.585
- Generador Automático de Sistemas Bioinspirados de Visión en Hardware Reconfigurable.
Martínez A., Pelayo F., Morillas C., Romero S., Carrillo R., del Pino B......597
- Coprocesador de Extracción de Minutia para Microblaze
Cantó E., Cañellas N., Fons M., Fons F., López M......605

8.2 Redes neurales y algoritmos aproximativos

- Implementación Custom DSP en FPGAs basado en una Red Neuronal Digital SOM con Sincronización Self-Timed.
Raygoza J.J., Ortega S., Boemo E.613
- Implementación Hardware de Algoritmos Genéticos usando FPGAs: El TSP como Caso de Estudio.
Vega M., Gil R., Ávila J., Gómez J., Sánchez J......621
- Implementación de un Controlador de Estructura Variable mediante Lógica Reconfigurable.
Al-Hadithi B., Suardíaz J., Iborra A......629
- Capítulo 9: Aplicaciones específicas**641

- Sistema de Identificación Remota y Cronometraje basado en Lógica reconfigurable. (*Tutorial*)
Amuchastegui C., Ayuso N., Alvarez G., Vicario L., Pico J., Benítez N., Gerendáin A.643

Generador automático de sistemas bioinspirados de visión en hardware reconfigurable

Antonio Martínez, Francisco Pelayo, Christian A. Morillas, Samuel Romero,
Richard R. Carrillo, Begoña del Pino

amartinez@atc.ugr.es, fpelayo@ugr.es, {cmorillas, sromero, rcarrillo, bego}@atc.ugr.es
Dpto. Arquitectura y Tecnología de Computadores. Universidad de Granada. España.

Abstract. Se presenta una herramienta de síntesis automática ideada para el desarrollo de modelos bioinspirados de visión en hardware reconfigurable a partir de descripciones funcionales Matlab-Simulink. La herramienta incorpora elementos característicos de co-simulación y co-emulación para la comprobación de modelos, tales como la generación automática del software necesario para la interfaz con el hardware implementado. Este sistema está siendo utilizado para modelar y desarrollar una neuroprótesis visual capaz de emular el comportamiento de la retina humana y generar trenes de impulsos en formato neuromórfico.

1 Introducción

El trabajo se centra en los problemas de modelado e implementación en tiempo real, de tareas de procesamiento de información visual mediante técnicas inspiradas en los sistemas de visión de los seres vivos. Se presenta una herramienta capaz de proporcionar una solución final en hardware/software reconfigurable, que implemente todo el sistema a partir de una especificación de alto nivel y que posibilite la co-simulación y co-emulación del mismo. De esta manera, la herramienta posee dos funcionalidades bien diferenciadas, de una parte la destinada a sintetizar en VHDL el comportamiento requerido y de otra la generación automática de todo lo necesario para llevar a cabo una simulación congruente. A continuación detallamos algunas características de nuestra línea de trabajo, y en la sección siguiente describimos la herramienta propuesta.

1.1 Justificación de la implementación

Los requisitos de cómputo para el procesamiento en tiempo real de información visual son excesivos cuando se trata de extraer información de escenas naturales en presencia de movimiento, mientras que los sistemas biológicos son capaces de procesar esta información sin aparente esfuerzo, gracias al grado de paralelismo y especificidad existente en los distintos centros nerviosos de su sistema visual.

El paralelismo a que nos referimos es de granularidad fina y más fácil de obtener por medio de las arquitecturas actuales del tipo FPGA, que integran el equivalente a millones de puertas lógicas reconfigurables.

Una de las cuestiones clave en el desarrollo de aplicaciones de lógica reconfigurable orientadas a visión, es disponer de medios eficientes para la especificación y prueba de modelos, operando directamente con el flujo de imágenes captado de una o más cámaras. Los entornos de software de simulación funcional tales como Matlab son especialmente versátiles para este fin, pero se requieren módulos de interfaz adecuados para intercambiar información con el hardware, así como para realizar las transformaciones oportunas de los datos y algoritmos software para que proyecten bien sobre la lógica reconfigurable. La herramienta propuesta aborda éstos dos aspectos: permite el modelado, evaluación y síntesis automática en hardware reconfigurable orientado a tareas de procesamiento de la visión, a partir de una descripción del sistema total mediante Matlab/Simulink. También genera el software necesario para comprobar y comunicar los distintos dispositivos con un ordenador, (programas tipo *host* que habiliten una interfaz PC-Hardware reconfigurable y otros posibles programas de utilidad para ver los resultados y/o generar patrones de prueba). Partiendo del alto nivel de abstracción en la descripción de sistemas que podemos conseguir con herramientas de especificación funcional de alto nivel como Matlab/Simulink, accedemos a la implementación automática en hardware reconfigurable de dicho comportamiento. De forma análoga, utilizamos Matlab/Simulink como herramienta de simulación del sistema, haciendo especial hincapié en la posibilidad de ofrecer co-simulación del sistema, en la línea de herramientas como CodeSimulink/SMT6040 [1], System Generator de Xilinx [2] y Altera DSP Builder [3], orientadas a sistemas de control y procesamiento digital de señales.

1.2 Modelado del sistema

El presente modelo de trabajo se fundamenta en los resultados obtenidos hasta la fecha en el proyecto de investigación europeo CORTIVIS [4]. El consorcio Cortivis tiene como principal objetivo desarrollar una neuro-prótesis visual que actúe sobre el área visual de la corteza cerebral, destinada principalmente a personas que hayan sufrido una pérdida total de la visión. De este modo, se trata de implementar un sistema de procesamiento bioinspirado de información visual, que funcione a una velocidad suficiente para proporcionar una percepción visual casi continua. Dicho sistema será capaz de producir estímulos nerviosos similares a los que tienen lugar en la corteza visual primaria de una persona vidente normal y debe ser totalmente configurable porque ha de adaptarse en función de cada persona y de las condiciones de funcionamiento.

Una de las principales herramientas desarrolladas en el marco de Cortivis es Retiner, un entorno completo de simulación de esquemas de visión bioinspirados basado en Matlab. Esta herramienta puede probar y simular con diferentes niveles de detalle el tracto visual de un humano, llegando incluso a la generación de trenes de impulsos (o spikes) directamente aplicables a la corteza visual primaria. Dicho sistema se fundamenta en un modelo bioinspirado de retina descrito en [5].

El modelo de retina que utiliza Retiner se conforma mediante un conjunto de filtros espacio-temporales que el ingeniero debe ajustar usando el entorno. Una característica esencial es que todos los componentes del modelo estén parametrizados, pudiéndose variar el tamaño de los filtros, la resolución de los datos, etc. En la Fig. 1 mostramos un ejemplo del tipo de filtros que estamos modelando y sintetizando con nuestra herramienta. La entrada al sistema viene en este caso definida por una serie de canales cromáticos y de intensidad extraídos de una entrada de video concreta. Estas señales se van procesando y mezclando con una serie de filtros del tipo DoG (Difference of Gaussians), LoG (Laplacian of Gaussians), X (ponderador), T (filtrado temporal) etc; para finalmente obtener una cierta matriz de actividad o mapa de "sapiencia", que representa la actuación del modelo.

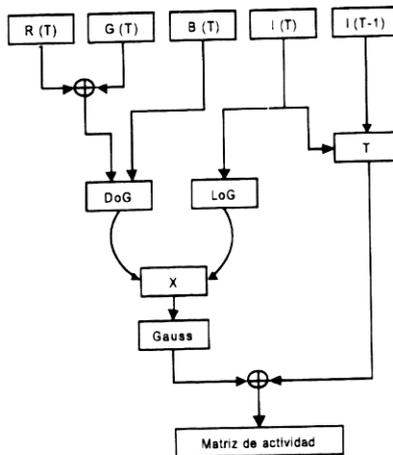


Fig. 1. Ejemplo de sistema de visión, definido como combinación de filtros de realce espacial con uno de realce temporal.

2 Descripción de la herramienta

La estructura software de la herramienta está diseñada siguiendo un modelo de componentes que ayude a una mejor depuración y organización de cada una de las tareas de la misma. Presentamos a continuación los esquemas para síntesis y para simulación que seguimos. Hemos llamado al sistema completo HSMaker, acrónimo de "Hardware-Software Maker" de ahora en adelante HSM.

La Fig. 2 aclara el flujo de diseño que estamos manejando en relación con los niveles de abstracción típicos en el diseño de circuitos integrados. La salida de HSMaker se realiza en VHDL para su síntesis lógica.

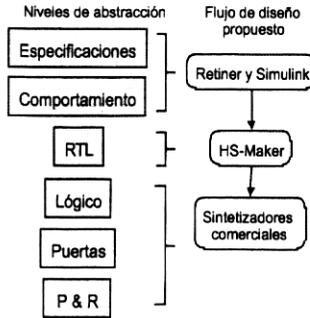


Fig. 2. Flujo de diseño típico con HSM

2.1 Organización de las tareas de síntesis

A continuación en la Fig.3 y Tabla 1 se presenta el esquema que sigue nuestra herramienta para la síntesis automática.

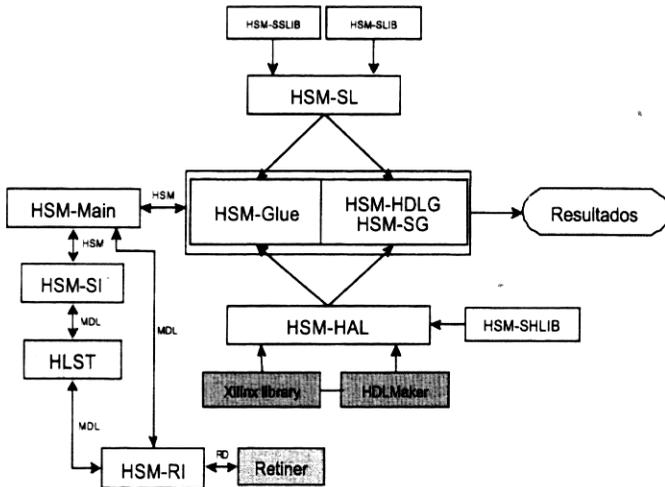


Fig. 3. Flujo de diseño para la síntesis.

El núcleo de la herramienta es la caja central de la Fig. 3 (cajas HSM-Glue, HSM-HDLG y HSM-SG). Aquí es donde se realiza toda la tarea de síntesis a partir de la especificación del modelo (dada por HSM-Main) y toda la batería de bibliotecas hardware y software. Dicho conjunto de bibliotecas de componentes nutren a la caja central a través de sendas capas de abstracción software y hardware (HSM-SL y HSM-HAL). Estas capas dotan a la herramienta de una adecuada independencia del hardware que se pretende desarrollar, además de facilitar la adición de nuevos componentes Hw o Sw. De esta forma, la aplicación es fácilmente actualizable sin modificar el núcleo de síntesis. Entre las bibliotecas suministradas por defecto, encontramos las orientadas al software, bien para simulación como en el caso de HSM-SSLIB (componentes de simulación bajo Simulink) o bien destinadas a desarrollar una interfaz adecuada con el hardware como HSM-SLIB (Ej. programas de interacción PC-Placa FPGA). De la parte hardware encontramos la biblioteca de Xilinx y la biblioteca de componentes del programa HDLMaker de Polybus [6]. El núcleo de síntesis se subdivide en 2 bloques, de una parte el módulo HSM-Glue y de otra el módulo conformado por los generadores de software y hardware, HSM-HDLG y HSM-SG respectivamente. HSM-Glue es una herramienta y a la vez un macrolenguaje que aún en un fichero una descripción funcional o estructural de un sistema concreto, incluyendo interfaz y módulos de software necesarios. Contiene toda la información necesaria para generar un sistema hardware-software concreto. A la derecha y dentro del núcleo de síntesis, tenemos los generadores de hardware y software. El HSM-HDLG se encarga de inferir módulos hardware en un lenguaje HDL como el VHDL, y el módulo HSM-SG proporciona las bibliotecas e incluso las aplicaciones ejecutables necesarias para interaccionar con el hardware soporte que se está utilizando, por ejemplo con la tarjeta RC1000 de AlphaData.

Acrónimo	Significado
HSM	HSMaker y extensión de los proyectos de HSM
MDL	Extensión de los archivos de modelos de Simulink
RD	Descripción en Retiner de una retina
HSM-Main	Aplicación principal de HSM
HSM-HAL	HSM Hardware Abstraction Layer
HSM-SL	HSM Software Layer
HLST	Herramienta de especificación de alto nivel
Retiner	Programa de simulación de retinas en Matlab
HSM-Glue	Aplicación HSM-Glue
HSM-HDLG	HSM Hardware Description Language Generator
HSM-SG	HSM Software Generator
HSM-SSLIB	HSM Simulink Software Library
HSM-SLIB	HSM Software Library
HSM-SHLIB	HSM Simulink Hardware Library
HSM-RI	HSM Retiner Interface
HSM-SI	HSM Simulink Interface

Tabla 1. Acrónimos utilizados

El núcleo de síntesis es comandado por el módulo HSM-Main, un “front-end” de las bibliotecas de HSM. Este módulo es, en última instancia, quien se encarga de “decidir” cuándo se ha llegado a un resultado apropiado a partir de las exigencias del desarrollador. Es decir, existe una realimentación entre el núcleo de síntesis y HSM-Main. Finalmente, se obtiene a la salida un diseño de proyecto susceptible de ser implementado por un herramienta de síntesis concreta como el Xilinx ISE.

Otra parte fundamental de la herramienta es la entrada de diseño. Dicha entrada se lleva a cabo a través del módulo HLST, es decir, de una herramienta como Simulink o bien a través de Retiner, usando para ello sendos módulos de adaptación que configuran un proyecto de HSM a partir de dichos bloques (HSM-SI y HSM-RI respectivamente). No obstante, el módulo HSM-Main puede establecer una realimentación con estas posibles entradas de síntesis incluso modificarlas en el formato original de las herramientas de origen.

2.2 Simulación del sistema

El esquema que seguimos para la co-simulación y la co-emulación (Fig. 4) se basa en la herramienta de especificación, concretamente en Matlab y Simulink. Usando las bibliotecas aportadas por HSM en forma de “toolboxes” de Simulink, es posible modelar un sistema por bloques para después proceder a su simulación y/o síntesis. También es posible a través de la utilidad HSM-RI, simular en Matlab/Simulink con HSM una retina que está descrita mediante Retiner.

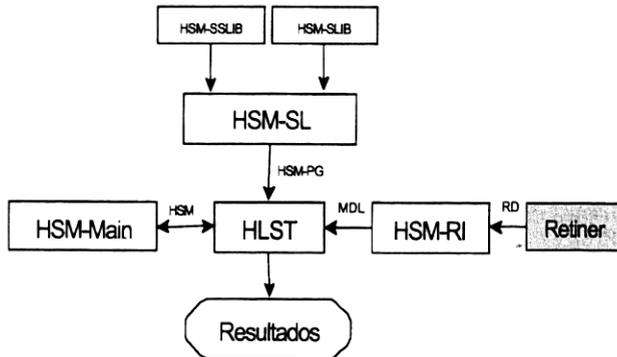


Fig 4. Flujo de diseño para la simulación.

3 Conclusiones

Se ha presentado una herramienta de síntesis, co-simulación y co-emulación de sistemas híbridos software/hardware de carácter general, aunque inicialmente concebida para facilitar el desarrollo de sistemas de visión bioinspirados. La herramienta produce código VHDL sintetizable en FPGAs u otras tecnologías de circuito integrado específicas. Creemos que esta estrategia es muy útil de cara a experimentar con diferentes modelos funcionales de sistemas ya que acorta los tiempos de diseño de hardware/software y posibilita un gran abanico de oportunidades de simulación. A su vez, hemos expuesto las cualidades del diseño de la herramienta, que a través de sendas capas de abstracción Sw/Hw, la hacen especialmente interesante a la hora de agregar nuevas bibliotecas y nuevos módulos; por ejemplo una interfaz con CodeSimulink [1].

Agradecimientos

Este trabajo está siendo financiado por la Comisión Europea con el proyecto de investigación CORTIVIS (*Cortical Visual Neuro-Prosthesis for the Blind*, QLK6-CT-2001-00279).

Referencias

1. L. Lavagno, B. Pino, L.M. Reyneri, A. Serra, "A Simulink-based Approach to System Level Design and Architecture Selection", Proc. of the 26th EUROMICRO Conference (EUROMICRO 2000), Vol. I, pp. 76-83 ISBN, 0-7695-0780-8, Maastricht, the Netherlands, September 5-7, 2000. Para más información sobre esta herramienta visitar la web <http://polimage.polito.it/groups/codesimulink.html>.
2. <http://www.xilinx.com/products/software/sysgen/features.htm> Página web principal de System Generator for DSP.
3. Altera DSP Builder – Reference Manual. Altera Corporation, Julio 2003.
4. <http://cortivis.umh.es> Página web principal del proyecto Cortivis.
5. F.J. Pelayo, C. Morillas, A. Martínez, S. Romero, and B. Pino E. Ros. A reconfigurable machine to model the first stages of the human visual pathway. In Actas de las III Jornadas sobre Computación Reconfigurable y Aplicaciones (JCRA'2003), pages 173–182. Madrid, 10–12 de Septiembre 2003. ISBN: 84-60099-28-8.
6. <http://www.polybus.com> Página web principal de Polybus Systems Corporation.