

Análisis de eficiencia en sistemas de Computo de Alta Performance Reconfigurable

Daniel Martin Morales^{1,2}, Eduardo J. Kunysz^{1,2}, Jorge R. Osio^{1,2}, José Rapallini¹

¹CeTAD, Departamento Electrotécnica, Facultad de Ingeniería - Universidad Nacional de La Plata

²Instituto de Ingeniería y Agronomía - Universidad Nacional Arturo Jauretche

martin.morales@unaj.edu.ar, ekunysz@unaj.edu.ar, josio@unaj.edu.ar, josrap@ing.unlp.edu.ar.

Abstract

El objetivo de este trabajo, es presentar la arquitectura y características de un sistema para el estudio de las nuevas técnicas de procesamiento paralelo, en el desarrollo de sistemas de cómputo de aplicaciones específicas de altas prestaciones. Poniendo énfasis en la optimización de rendimiento y reducción de consumos energéticos, se propone una alternativa que representa un nuevo paradigma en el desarrollo de supercomputadoras. Dicha alternativa, viene de la mano de tecnología de arreglos de compuertas lógicas formando parte del cómputo de alta performance reconfigurable (HPCR).

Palabras Clave: FPGA, High Performance Reconfigurable Computing, Sistemas Embebidos, Aplicaciones específicas.

1. Introducción

En la medida que el manejo del volumen de la información avanza, los desafíos en cuanto a capacidad de cómputo y cantidad de datos aumenta drásticamente. Actualmente es normal pensar en múltiples procesadores para la industria masiva y supercomputadoras para usos específicos.

Muchos investigadores han demostrado que la computadora de propósitos generales con procesadores gráficos (GPUs), son una alternativa interesante para resolver problemas de cálculos intensivos[1][2][3]. El desafío en estos casos consiste en trasladar el problema concreto a las limitaciones de la programación gráfica. Éstos GPUs (Graphics Processing Unit), contienen múltiples procesadores simples, en su conjunto llamados many-cores. Este tipo de tecnología normalmente utiliza arquitecturas híbridas, donde parte del procesamiento se realiza en la computadora de propósitos generales y la

otra en la placa gráfica.

Con la intención de expandir este concepto hacia aplicaciones más específicas dejando de lado la versatilidad, los fabricantes de computadoras de altas prestaciones, han introducido unidades de co-procesamiento de arreglos de lógica programable. La aplicación principal se ejecuta en los microprocesadores, mientras que las FPGAs manejan las porciones de código que requieren mayor tiempo de ejecución. Estas porciones de procesamiento suelen ser datos en paralelo superpuestos, arquitecturas que se pueden implementar con una alta granularidad, una sola instrucción, una instrucción y múltiples datos (SIMD), entre otros [2].

Una de las ventajas de este tipo de arquitectura es la capacidad que tiene el procesador principal de reconfigurar las FPGAs en tiempo real. Esto permite la reutilización de partes de hardware en virtud de mejorar la performance del software, lo que crea un nuevo paradigma en el campo de los desarrolladores de aplicaciones. Los sistemas tradicionales utilizan técnicas de descripción de hardware como VHDL. Otros utilizan lenguajes de alto nivel adaptados ([7][8][9][10]) como C o Fortran, o entornos de programación gráficos [11][12]. Esas últimas alternativas permiten una transición natural entre el mundo de hardware y el software.

Con las primeras FPGAs, el reducido tamaño y capacidad que tenían, limitaba mucho la evolución de sistemas tan complejos como los que se pueden implementar en una ASIC. Pero hoy en día sabemos, que los recursos de los dispositivos de arreglos de lógica programable, se incrementan cada año debido a la Ley de Moore, con el agregado de memoria RAM embebida. Con estos incrementos se logran diseños de gran magnitud. Hasta tal punto es así que la tendencia actual es integrar dentro de una FPGA sistemas digitales completos que incluyen un microprocesador de propósito general y todo el hardware de propósito específico que requiere la aplicación [1]

Esto ha llevado a elevar el número de integrados de lógica programable en comparación con el número de procesadores. Con lo que hoy en día tenemos plataformas de altas prestaciones en donde casi todo el procesamiento se realiza en FPGAs.

Típicamente el clock de una FPGA es un orden de magnitud o más lento que el de un procesador. Sin embargo desde el punto de vista de la velocidad, las FPGAs obtienen su ventaja en los siguientes tres factores:

1. *Intensidad*: en el mejor de los casos un CPU puede realizar una operación entre enteros cada dos ciclos de reloj (caso ideal en donde se estaría utilizando cache, y con el pipeline funcionando sin interrupciones). El peor caso es significativamente más desfavorable. En una FPGA se implementa sólo la funcionalidad necesaria para la aplicación particular. Por lo tanto se prescinde del “overhead” producto de la arquitectura sofisticada que tiene un CPU de propósitos generales.
2. *Latencia*: con una FPGA se puede obtener una granulosidad fina sobre el control de donde se encuentran los datos en memoria.
3. *Paralelismo espacial*: se puede realizar un pequeño pipeline de propósitos especiales para realizar una operación particular, y luego replicarlo dentro del chip de la FPGA.

Entre algunas de las alternativas que se pueden explorar con computadoras paralelas reconfigurables tenemos tres parámetros:

- **Comunicación**: posibilidad de ensayar diversas topologías conocidas, o explorar nuevas alternativas que podrían optimizar el rendimiento general. Los dispositivos más modernos permiten interfaces de alta velocidad como GbEthernet, o PCI Express. Se pueden implementar protocolos de comunicación flexibles y variar el ancho de bus según la necesidad. Una de las características que se puede utilizar con esta tecnología es la capacidad de reconfiguración parcial de los dispositivos de lógica programable. Esto último permite crear redes virtuales reconfigurables en hardware en tiempo real que sirvan de soporte para enrutado en software [13].
- **Memoria**: cada problema específico requiere de una configuración de memoria determinada. Para ello se cuenta con memoria interna (en general limitada) la cual se podría utilizar para procesos locales y luego se puede interfacear con distintas tecnologías de memoria existentes. Los dispositivos más modernos tienen incorporados módulos de control para memorias

del tipo DDR3.

- **Software**: en este tipo de plataformas es el aspecto más difícil de estandarizar. El software necesariamente es híbrido entre partes de hardware (manejado por drivers) y lenguajes de alto nivel. En estos casos el desafío es encontrar que partes de alto nivel son las que generan mayor “overhead” sobre el procesamiento, identificarlas y luego trasladarlas a su versión de compuertas lógicas. (Implementarlas dentro de la FPGA)

Si bien el estudio de sistemas paralelos con múltiples procesadores, es una ciencia bien estudiada, el estudio de utilización de múltiples dispositivos reconfigurables no es un terreno completamente explorado.

2. Implementación

El grupo de investigación está explorando técnicas de procesamiento de HPRC con una placa que está compuesta por una FPGA (Xilinx Spartan 6), 3 memorias SRAM para maximizar el throughput de acceso aleatorio y 2 memorias del tipo DDR2 para almacenamiento de datos. Esto, junto a las fuentes de alimentación e interfaces de comunicación están integrados en un circuito impreso de 12 capas y de un tamaño de 8x12cm.

Sobre la base de esta placa se está trabajando en el diseño de una maqueta de simulación para evolucionar hacia tecnologías superiores con varias FPGAs por placa.

En una primera etapa se trabajará sobre una unidad de procesamiento (UP) compuesta por dos FPGAs

En la Fig 1, se observa la arquitectura propuesta para esta variante.

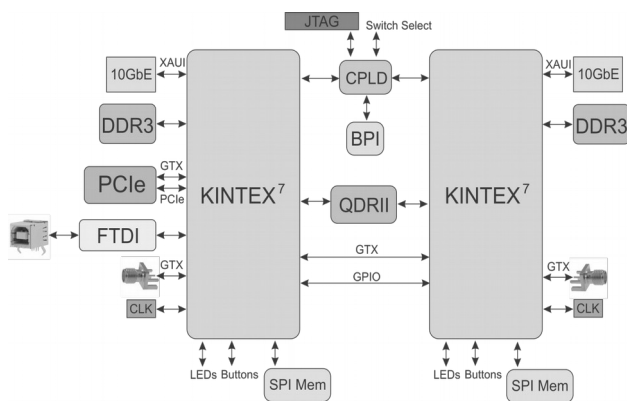


Fig. 1. Topología Backplane / Gigabit ethernet

Algunas de las características de este diseño son:

- Memorias DDR3
- SRAM Quad Data Rate (QDR11); memorias de alta velocidad y baja latencia.
- Memorias Flash BPI (Byte Peripheral Interface);

- memorias de configuración de alta velocidad.
- Interfaz integrada PCIe (Gen 2.0 5 GB/s), a través de los transceivers GTX.
- 10 Gb Ethernet, a través del protocolo XAUI (10 Attachment Unit Interface)

El objetivo es probar distintas opciones y topologías que puedan ser útiles en una versión más avanzada de 4 FPGAs por UP

Para una etapa posterior, se estudian distintas topologías de interconexión que permitan flexibilidad de proyectos y aplicaciones específicas.

3. Simulaciones

Uno de los grandes beneficios de la utilización de sistemas de arreglos de lógica programable es la gran versatilidad a la hora de simular comportamientos.

Existe un abanico enorme de aplicaciones de simulación, las cuales logran resultados no sólo adecuados, sino muy realistas del algoritmo. Una de las herramientas utilizadas para estos fines es el “Modelsim” que permite evaluar la descripción de hardware independientemente del fabricante.

Mediante dichas simulaciones no solo se puede verificar al flujo de la información sino que también es posible verificar los “constrains” temporales del circuito integrado.

Uno de los primeros trabajos que está realizando el grupo de investigación es la comparación a nivel de simulación de diversos algoritmos corriendo en una FPGA. El objetivo en esta instancia, es dimensionar de forma estimativa el consumo energético en función al número de ciclos de reloj requeridos. Alguno de los cálculos propuestos son multiplicaciones de matrices y filtrado de señales de forma digital. Luego éstos resultados serían comparados con los obtenidos al resolver los mismos cálculos en procesadores de usos generales.

4. Conclusiones

La tecnología FPGA está introduciéndose fuertemente en el procesamiento paralelo. Se presenta como una alternativa fuerte y confiable a la resolución de varios problemas de la ciencia como de la industria.

Como se analizó, las computadoras de alta performance son necesarias en la resolución de cálculos con gran densidad de parámetros y datos. Y también para la búsqueda de información dentro de grandes bases de

datos en donde se pueden implementar algoritmos de búsqueda por correlación [14].

Cada implementación o arquitectura está ligada directamente a la aplicación para la cual se desea implementar:

Teniendo en cuenta el nivel de comunicación se pueden utilizar topologías “backplane” (para requerimientos más leves) o topologías “cubo” para sistemas que presentan requisitos altos de comunicación y modularidad.

Teniendo en cuenta la simetría de los procesos, si se puede paralelizar en partes idénticas, ya sea la tecnología, como la comunicación es homogénea. Sin embargo, si el problema se puede paralelizar en partes diferentes, se puede pensar en sistemas heterogéneos, utilizando diferentes tecnologías de procesamiento por cada etapa.

Las necesidades comerciales y disminución de riesgos al desarrollar una supercomputadora, hacen de las supercomputadoras basadas en tecnologías FPGA una apuesta interesante de mercado.

5. Bibliografía

- [1] “CUBE: a 512-FPGA Cluster”, Oskar Mencer, Kuen Hung Tsoi, Stephen Craimer, Timothy Todman and Wayne Luk, Ming Yee Wong and Philip Heng Wai Leong , Dept. of Computing, Imperial College London , Dept. of Computer Science and Engineering The Chinese University of Hong Kong .
- [2] “The NAPA Adaptive Processing Architecture”, Charlé R. Rupp, Ph.D., Mark Landguth, Tim Garverick, Edson Gomersall, Harry Holt , National Semiconductor Corporation .
- [3] “grape-4 : A Massively Parallel Special-Purpose Computer for Collisional N-Body Simulations”, Junichiro Makino , Makoto Taiji, Toshikazu Ebisuzaki, and Daiichiro Sugimoto . IEEE
- [4] “Implementation trade-offs of Triple DES in the SRC-6e Reconfigurable Computing Environment ” ,Osman Devrim Fidanci, Hatim Diab, Tarek El-Ghazawi, Kris Gaj and Nikitas Alexandridis
- [5] “FPGAs vs. CPUs: Trends in Peak Floating-Point Performance ”, Keith Underwood , Sandia National Laboratories .
- [6] “High Performance Reconfigurable Computing for Science and Engineering Applications ”, Peter Leonard McMahon
- [7] "Implementing Simulink Designs on SRC-6 System", David Meixner, Volodymyr Kindratenko, David Pointer, Innovative Systems Laboratory, National Center for Supercomputing Applications,

University of Illinois at Urbana-Champaign
(<http://www.srccomp.com/carte-programming-environment>)

- [8] "Using Hardware Libraries with Impulse C", Ralph Bodenner, Director of Product Development, Impulse Accelerated Technologies, Inc. Application Note (<http://www.impulseaccelerated.com/>)
- [9] "Low Power Hybrid Computing for Efficient Software Acceleration", Mitronics, White Paper. (<http://www.mitronics.com/>)
- [10] "DK4 - Handel-C Language Reference Manual", Celoxica, (<http://www.celoxica.com/>)
- [11] "CoreFire™ Design Suite", Annapolis Micro Systems, Inc., DataSheet (<http://www.annapmicro.com/corefire.html>)
- [12] "Model-Based Design with Simulink, HDL Coder, and Xilinx System Generator for DSP", Kiran Kintali and Yongfeng Gu, MathWorks, White Paper (<http://www.mathworks.com/fpga-design/simulink-with-xilinx-system-generator-for-dsp.html>)
- [13] Customizing Virtual Networks with Partial FPGA Reconfiguration, Dong Yin , Deepak Unnikrishnan, Yong Liao, Lixin Gao and Russell Tessier . Dept. of Electrical and Computer Engineering University of Massachusetts
- [14] "High Performance Biological Pairwise Sequence Alignment: FPGA versus GPU versus Cell BE versus GPP", Khaled Benkrid, Ali Akoglu, Cheng Ling, Yang Song, Ying Liu, Xiand Tian. International Journal of Reconfigurable Computing 2012.