Avances en la síntesis de alto nivel para la generación de hardware en FPGA: Modelos y programabilidad

Maria Angélica Dávila-Guzmán, Rubén Gran-Tejero, María Villarroya-Gaudó, Darío Súarez-Gracia

Afiliación: Grupo de Arquitectura de Computadores
Instituto de Investigación en Ingeniería de Aragón (I3A)
Universidad de Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, Spain.
Tel. +34-976762707, e-mail: angelicadg@unizar.es

Resumen

Mejorar el rendimiento en sistemas de cómputo ha impulsado el uso de aceleradores como FPGAs. Este trabajo presenta 2 propuestas que aúnan su programabilidad y rendimiento utilizando síntesis de alto nivel, HLS, con FPGAs: 1) A través del análisis y modelado de las unidades funcionales generadas por los compiladores, con énfasis en la memoria y 2) Implementando frameworks que permitan el uso eficiente de los recursos de las FPGA en dominios específicos como la visión por computador.

Introducción

El interés por dispositivos re-programables como las FPGAs para acelerar aplicaciones de cómputo ha crecido no, principalmente como propuestas ante las limitaciones físicas en el escalado de los transistores, como se había predicho en la ley de Moore. Estas limitaciones han afectado el incremento de la frecuencia de los procesadores y la potencia, en lugar de permanecer constante, se ha incrementado de forma exponencial debido a las corrientes de fuga, poniendo fin a las proyecciones del escalado de Dennard [1].

Estos cambios en las bases de la evolución de los sistemas de cómputo ha supuesto una oportunidad para el uso de las FPGAs, que poseen gran cantidad de recursos para ser configurados, lo que permite tener hardware específico para cada aplicación. Las FPGAs han probado ser una solución eficiente energéticamente en computación de alto rendimiento, *machine learning*, procesamiento de imágenes, entre otros, favoreciendo la relación rendimiento/energía, comparada con procesadores de propósito general [2]. Además, las FPGAs están ahora preparadas para soportar aplicaciones con altas demandas de memoria con la inclusión de memorias on-chip con alto ancho, como por ejemplo las HBM.

A pesar de las ventajas de las FPGAs, la adopción de estas por parte de los programadores sigue siendo compleja, incluso con la introducción de HLS, que permite su programación con lenguajes de alto nivel como C/C++, OpenCL, SyCL, entre otros. La HLS es una solución parcial, porque los tiempos de desarrollo continúan siendo altos debido a la generación del bitstream y las técnicas de optimización en las FPGAs son diferentes a dispositivos como las CPUs y las GPUs.

Para mejorar la productividad usando HLS, se pueden optar por dos propuestas: 1) Estimar por medio de modelos analíticos, en etapas de pre-compilación, la eficiencia de un kernel para FPGA, que se puede obtener en minutos; 2) Usar patrones de programación bien conocidos para obtener hardware eficiente.

En este trabajo abordamos estas dos propuestas. En la primera hemos demostrado que en aplicaciones limitadas por la memoria externa se pueden predecir con precisión los tiempos de ejecución en dos tipos de memoria DRAM: DDR4 y HBM. En cuanto a la programación usando patrones, se ha realizado una propuesta de implementación del framework OpenVX diseñado para generar hardware eficiente para el procesamiento de imágenes, portando y mejorando una propuesta para FPGAs de Xilinx a FPGAs de Intel, para reducir la brecha de portabilidad de los estándares de programación entre estos, que sigue siendo muy notoria [5].

Estimación del tiempo de ejecución mediante modelado analítico

La comunicación entre el kernel, código, con la memoria DRAM a través del bloque de interacción de memoria global o GMI, influye enormemente en el tiempo de ejecución de las aplicaciones limitadas

por memoria. El modelo propuesto se basa en el análisis de los bloques funcionales que forman parte del GMI junto con las propiedades de las propias memorias y algunos parámetros del kernel. [6].

Las necesidades de cada tipo de acceso a memoria conllevan a latencias muy diferentes en el hardware generado, estos accesos son capturados por el modelo lo que permite lograr un error medio de 11% para DDR y del 10% para HBM. Comparado con otras propuestas, este modelo reduce el error al menos dos veces a lo estimado por trabajos previos [3][4].

Para 16 aplicaciones relevantes en HPC, la Figura 1 muestra los resultados del tiempo medido y estimado para DDR4 y HBM, usando un banco por variable global.

Aceleración de patrones para aplicaciones de visión artificial

Basados en el HiFlipVX[7], una implementación de kernels para FPGA basada en OpenVX, esta propuesta extiende sus funcionalidades para las FPGAs diseñadas por Intel. Esa librería está orientada a la generación de hardware en la FPGA, por lo cual su implementación base difiere de las usadas en CPU y GPU. Como resultado se obtiene la primera propuesta realmente portable, entre fabricantes y familias de FPGAs para el estándar OpenVX con un framework que permite emplear C++ simplificando la programación de aplicaciones para el procesamiento de imágenes. La Figura 2 muestra los resultados de recursos y aceleración de 3 aplicaciones comparadas con el estado del arte.

Conclusiones

Este trabajo demuestra a través de dos enfoques como HLS para FPGAs permite mejorar las opciones de programabilidad ayudando a los programadores a superar las barreras de la programación de hardware reconfigurable haciendo uso de modelos para predicción, y de frameworks que oculten los detalles requeridos por las FPGAs.

REFERENCIAS

1. SCHAFER B. C. and WANG Z. "High-Level Synthesis Design Space Exploration: Past, Present, and Future," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2628-2639, Oct. 2020.

2. CONG J., ET AL.. "Customizable Computing—From Single Chip to Datacenters," in *Proceedings of the IEEE*, vol. 107, no. 1, pp. 185-203, Jan. 2019.

3. WANG Z., HE B., ZHANG W. and JIANG S., "A performance analysis framework for optimizing OpenCL applications on FPGAs," in *HPCA*, 2016,

4. CHOI Y., ET.AL. 2017. HLscope+: fast and accurate performance estimation for FPGA HLS. In *ICCAD '17*. IEEE Press, 691–698.

5. VOSS N., ET AL. 2020. Performance Portable FPGA Design. *In Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '20)*. Association for Computing Machinery, New York, NY, USA, 324.

6. DAVILA-GUZMAN M., ET AL., "An Analytical Model of Memory-Bound Applications Compiled with High Level Synthesis," in *2020 FCCM*, Fayetteville, AR, USA, 2020 pp.

7. KALMS, L., ET AL. 2019. HiFlipVX: An Open Source High-Level Synthesis FPGA Library for Image Processing. *Applied Reconfigurable Computing*. 1(16), 17–31.

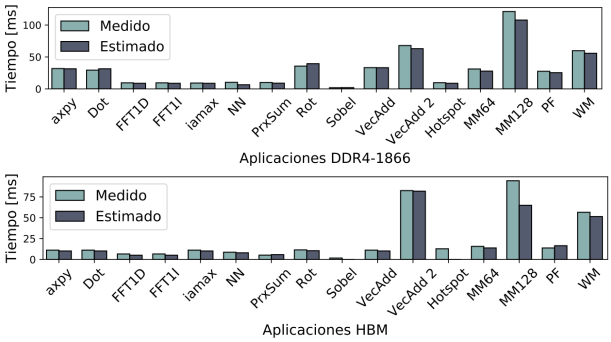

Figura 1. Tiempo Medido y estimado para 16 aplicaciones limitadas por memoria usando un modelo para FPGAs basado en memoria[6] para DDR4 1866 y HBM

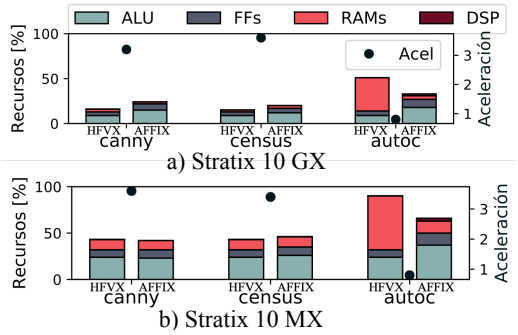

Figura 2. Recursos y aceleración para la propuesta de aceleración con patrones HiFlipVX(HFVX) comparado con el estado del arte (AFFIX) en dos FPGAS: a) Stratix10 Gx y b) Stratix 10 MX