

Fixed and Reconfigurable Multi-Core Device Characterization for HPEC

Jason Williams, Alan D. George, Justin Richardson, Kunal Gosrani, and Siddarth Suresh
NSF Center for High-Performance Reconfigurable Computing (CHREC)
ECE Department, University of Florida
{jwilliams, george, richardson, gosrani, suresh}@chrec.org

Abstract

As on-chip transistor counts increase, the computing landscape has shifted to multi- and many-core devices. HPEC systems have adopted this trend by incorporating both fixed and reconfigurable many-core and multi-core devices. As more, disparate devices enter the market, there is an increasing need for concepts, terminology, and classification techniques to understand device tradeoffs. Additionally, computational performance, memory system performance, and power metrics are needed to objectively compare devices. We present results with a hierarchical taxonomy of HPEC processing devices under study in terms of computational density and memory bandwidth.

Introduction

Over the last several years, multi-core devices have emerged as the leading technology to take advantage of increasing transistor counts. This *architecture reformation* is shifting the focus to exploiting explicit parallelism rather than relying upon instruction-level parallelism and higher clock rates to achieve high performance. The resulting *application reformation* is driving application developers to write explicitly parallel programs, rather than relying on automatic compiler optimizations for high performance. Multi-core devices are finding their way into new embedded technologies that are used to deliver high computational performance with low power consumption.

Multi-core devices have at least two major computational components in a single package. Many-core devices have many (e.g. hundreds) of computational components in a single package. We do not differentiate between multi-core and many-core devices and use the notation MC to refer to them collectively. We define two primary classes of MC architecture technology: Fixed MC (FMC) and Reconfigurable MC (RMC). FMC devices have a fixed hardware structure that cannot be changed after fabrication. RMC devices can change their hardware structure after fabrication to adapt to changing problem requirements. Multiple computational cores can be instantiated on the RMC fabric. Both FMC and RMC devices feature either a homogeneous or heterogeneous set of processing elements (PE) with various degrees of granularity. A large set of reconfigurability factors will be defined in the talk to further differentiate FMC vs. RMC.

Comparing disparate processing technologies impartially and objectively has been a challenge throughout the history of computing. It is an even greater challenge considering today's vast design space of FMC and RMC devices, and the number and variety of available architectures. We propose several forms of *computational density per Watt (CDW)* and *internal memory bandwidth (IMB)* metrics to facilitate device comparisons within and between architectural categories. CDW metrics are

featured in this abstract and will be even more so in the presentation. IMB metrics will be featured in the presentation but are not included here due to space limits.

CDW and IMB Metrics

We propose and feature several metrics to compare devices within and between taxonomy categories. For CDW, we evaluate bit-level, integer, and floating-point operations, the latter two in several sizes. *Computational density (CD)* describes the raw computational performance of a device on individual bits. It is normalized by power to produce CDW. *Bit-level CD* is an adaptation of work done in [1]. We deviate from the original metric by grouping devices into categories by process technology (130, 90, or 65 nm). CDW is calculated for coarse-grained FMC and RMC devices using operating frequency, PE bit-width, and number of PEs. For fine-grained RMC devices, primarily FPGAs, CDW is calculated using operating frequency, number of look-up tables (LUTs), and width and number of additional coarse-grained PEs on the chip.

To describe computational capabilities for integer calculations, *integer CD* is defined. FMC and coarse-grained RMC devices typically contain ALUs or coarse-grained processing elements for integer computation. In this case, the operating frequency, the number of integer execution units, and the average number of clock cycles per operation are used to determine integer CD.

For FPGAs, a methodology similar to the one described by Strenski is used [2]. This characterization is highly dependent on the performance of the IP cores. We assume that integer cores provided by the vendor are highly optimized and will provide a good basis for characterization. The parameters in Strenski's procedure are available as part of the core documentation from the vendor or via experimentation using vendor tools. Memory needs to be allocated to store two input operands per operation. The operands can be overwritten with the result in memory. Dual-port memory configurations are used to increase the internal bandwidth. Thus, the *memory-sustainable CD* is limited by the size of the operands and the amount of parallel paths to on-chip memory. For all metrics, the independent variable *parallel operations* is defined as the number of memory-sustainable operations that can be processed simultaneously.

In most cases, *floating-point CD* can be determined at the device level using similar methods as cited above for integer CD. Coarse-grained devices use the same metric as integer CD, but instead use the number of floating-point execution units and the average number of clock cycles per floating-point operation. Fine-grained FPGAs use a similar method as outlined in [2] using floating-point IP cores.

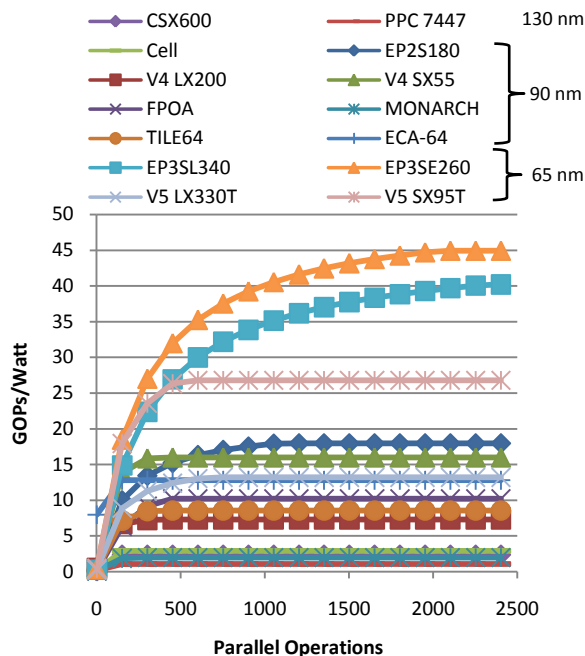


Figure 1. 16-bit Integer CDW

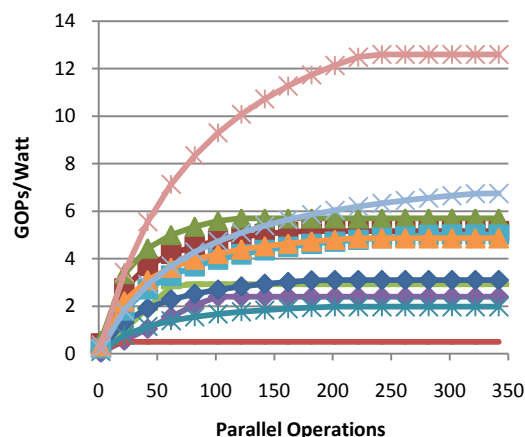


Figure 2. SPFP CDW

RMC devices can achieve much lower power consumption than their peak values, since only configured portions of the chip are active. Within a metric, we hold frequency constant at a realizable level based upon available cores. Therefore, for reconfigurable architectures, we assume that power scales linearly with resource utilization up to maximum power consumption specified in vendor documentation or using vendor estimation tools in the results that follow. CDW is calculated by taking the CD for each level of parallelism and dividing by the power consumption at that level of parallelism.

Characterizing devices based upon memory speed helps to realistically balance computational performance potential. High computational performance cannot overcome memory performance issues that cause the architecture to be continually stalled waiting for data to process. *Internal memory bandwidth (IMB)* describes the data transfer rate to on-chip memories. It is calculated using the number of parallel memory structures, the bit-width of each memory

structure, the clock frequency of memory, and the average number of clock cycles to access memory.

Results and Analysis

For results cited in this abstract, 16-bit integer and single-precision floating-point (SPFP) forms of CDW were calculated for a broad range of FMC and RMC devices of interest for HPEC. The full range of results and analyses in the presentation will include bit-level, 16-bit integer, 32-bit integer, SPFP, and double-precision floating-point (DPFP) forms of CDW, as well as IMB for the same devices featured. FMC devices under study include Cell Broadband Engine (IBM) [3-4], CSX600 (ClearSpeed) [5], and PowerPC MPC7447 with AltiVec extensions (Freescale) [6]. RMC devices include Arrix FPOA (MathStar) [7-8], ECA-64 (ElementCXI) [9-10], MONARCH FPCA processor (Raytheon) [11], Stratix-II EP2S180 [12], Stratix-III EP3SL340, and Stratix-III EP3SE260 FPGAs (Altera) [13], TILE64 (Tilera) [14], as well as Virtex-4 LX200 and SX55 [15], and Virtex-5 LX330T and SX95T FPGAs (Xilinx) [16].

Figure 1 shows 16-bit integer CDW for both FMC and RMC devices. For 90 nm devices, the leader for almost all levels of parallelism is the Stratix-II EP2S180, although the ECA-64 and Virtex-4 SX55 perform well. The Stratix-III FPGAs are the clear overall leader, due to their high performance at high levels of parallelism and low power consumption. The FMC devices tend to perform poorly in this metric due to their high, fixed power consumption.

SPFP CDW is shown in Figure 2. Devices in Figure 1 that are not intended for SPFP operations and would likely perform poorly are omitted in the SPFP results. Despite its significant performance advantage for raw CD performance for SPFP over other 90 nm devices, the Cell is extremely power-hungry and performs worse on CDW than most of the RMC devices. The 65 nm FPGAs have a major performance increase over the previous generation devices, while maintaining good power efficiency, so that they achieve the best CDW for all levels of parallelism, led by the Virtex-5 SX95T.

Conclusions

As highlighted in this abstract, our presentation will commence with a taxonomy for classifying MC devices for HPEC in terms of fixed vs. reconfigurable and degree of reconfigurability. This taxonomy will serve as the basis for a broad comparison and contrast of more than one-dozen promising device technologies for HPEC in terms of two important metrics. A broad range of results will be featured, including CDW for at least five different modes of operation and numerical precision, as well as IMB. Results will provide insight on inherent strengths and weakness of these diverse device technologies in addressing the needs of advanced HPEC systems.

Acknowledgements

This work was supported in part by the I/UCRC Program of the National Science Foundation under Grant No. EEC-0642422. Authors also acknowledge vendor equipment and/or tools provided by Altera, MathStar, and Xilinx.

References

- [1] A. DeHon. Reconfigurable Architectures for General Purpose Computing, PhD thesis, MIT AI Lab, Sept. 1996.
- [2] D. Strenski, "FPGA Floating Point Performance -- a pencil and paper evaluation," HPCWire, Jan. 12, 2007, <http://www.hpcwire.com/hpc/1195762.html>.
- [3] T. Chen, et al., "Cell Broadband Engine Architecture and its First Implementation--A Performance View," *IBM Journal of Research & Development*, vol. 51, no. 5, Sept. 2007, pp. 559-572.
- [4] D. Wang, "ISSCC 2005: the Cell Microprocessor," Real World Technologies, Feb. 2005, retrieved Jan. 2008, <http://www.realworldtech.com/page.cfm?ArticleID=rwt021005084318&p=2>.
- [5] ClearSpeed Technology PLC, *CSX600 Architecture Whitepaper*, 2007.
- [6] Freescale Semiconductor, Inc., *Altivec Technology Programming Environments Manual Rev. 3*, 2006.
- [7] Mathstar, Inc., *Arrix Family FPOA Architecture Guide*, 2007.
- [8] Mathstar, Inc., *Arrix Family Product Data Sheet & Design Guide*, 2007.
- [9] Element CXI, Inc., *ECA-64 Device Architecture Overview*, 2007.
- [10] Element CXI, Inc., *ECA-64 Product Brief*, 2007.
- [11] Raytheon Company, *World's First Polymorphic Computer – MONARCH*, 2006.
- [12] Altera Corporation, *Stratix II Device Handbook*, 2007.
- [13] Altera Corporation, *Stratix III Device Handbook*, 2007.
- [14] Tiler Corporation, *TILE64 Processor Product Brief*, 2008.
- [15] Xilinx, Incorporated, *Virtex-4 Family Overview*, 2007.
- [16] Xilinx, Incorporated, *Virtex-5 Family Overview*, 2008.