SANDIA REPORT
SAND2010-8063
Unlimited Release
Printed September 2010

Using Reconfigurable Functional Units in Conventional Microprocessors

Arun F. Rodrigues

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2010-8063
Unlimited Release
Printed September 2010

Using Reconfigurable Functional Units in Conventional Microprocessors

Arun Rodrigues
Scalable Computer Architecture
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS1319

Abstract

Scientific applications use highly specialized data structures that require complex, latency sensitive graphs of integer instructions for memory address calculations. Working with the Univeristy of Wisconsin, we have demonstrated significant differences between the Sandia's applications and the industry standard SPEC-FP (standard performance evaluation corporation-floating point) suite. Specifically, integer dataflow performance is critical to overall system performance. To improve this performance, we have developed a configurable functional unit design that is capable of accelerating integer dataflow.

CONTENTS

1. INTRODUCTION

Scientific applications at Sandia use highly specialized data structures that require complex, latency sensitive graphs of integer instructions for memory address calculations. In prior work [4], we have demonstrated significant differences between Sandia's applications and the industry standard SPEC-FP suite. The integer dataflow in Sandia's applications average 40% more instructions and 50% more operands than the SPEC benchmarks. Furthermore, 92% of the data produced by these graphs is used to generate a memory address. Therefore, integer dataflow performance is critical to overall system
performance. In the past, this problem could be handled with specialized address generation units. However, Sandia's large and complex applications do not have identical access patterns. We demonstrated that each application has between 16 and 32 unique important graphs. It is not feasible to implement unique address generation units for each graph type, and each new application may use a different set of graphs. Thus, a Reconfigurable Hardware (RH) is an attractive accelerator because its functionality can change to accelerate a variety of different graphs. We use RH tightly integrated with a microprocessor to accelerate integer dataflow, improve the number of outstanding memory operations and therefore improve overall system performance.
We have developed compile-time algorithms that detect and select common graphs for acceleration. In addition, we have developed an RH execution model that efficiently encodes a large number of inputs and outputs (needed for the complex Sandia graphs) for execution in reconfigurable units. The execution model also includes architectural techniques to improve performance by improving both issue time, output pipelining, and minimizing register file bandwidth. Ongoing research is working to maximize the number of cases where our techniques may be applied. Future research will investigate the potential of run-time techniques for detecting common dataflow graphs. The work is in collaboration with the University of Wisconsin-Madison.

1.1. OS Work

There are multiple ways that a thread scheduler's behavior can impact the behavior of the RH allocator. Thus far, we have addressed two of these: the issue of hardware pre-emption in response to software context-switches, and the problem that recent RH kernel requests near a context switch does not correctly predict near-future RH kernel needs. To evaluate the system performance of these techniques, we also developed metrics for heterogeneous systems.

When the OS preempts a thread to perform a context switch, that application's RH kernels cannot progress because they can no longer access virtual memory addresses (which require the owning CPU thread to perform translation). The "blocked" kernel sits idle until its owner thread is restored, but the OS cannot reallocate those resources. The OS could save the context information from the RH and restore at a later time, but this operation is prohibitively expensive. We have developed alternative preemption techniques that yield the benefits of preemptable hardware, but without the overhead associated with full save and restore of configuration and data.

The OS thread scheduler performs thread context switches at periodic intervals to time-share access to the CPU. Similarly, the RH allocator periodically determines the allocation of hardware based on recent requests for access to RH resources. However, a thread context switch affects the hardware needs of applications that will execute in the near future. We have developed two RH allocator techniques that consider application thread context switches when allocating RH so that the allocation is based both on the hardware use of the recent past and on the predicted needs of applications that will execute in the near future.
This work on OS matters is detailed in [1][2].

1.2. Metrics

Evaluating the performance of reconfigurable computing applications in multi-tasking systems using simulation (as can be needed in early design-space exploration) faces several challenges. The complexity of full-system, cycle-accurate simulation prevents executing applications of any appreciable size to completion. One must sample only a portion of execution; yet unless care is taken, the measured performance for the sampled interval will not be indicative of the complete execution. Although this is generally a problem for simulation-based evaluation, the problem is exacerbated for multi-tasking systems. Therefore, we have developed a performance evaluation methodology that accurately measures hybrid (both hardware and software) application performance, accounts for additional overhead introduced by hybrid resource management (such as run-time allocation of reconfigurable hardware), and correctly compensates for momentary imbalances in processor time allocation that are only artifacts of the (necessarily) short simulated execution timespan and would balance out over time.
The contribution of this work is the definition of a new method to calculate speedup in simulated hybrid multi-tasking systems. These techniques could also be applied to measurements such as power consumption or energy-delay product. The new method provides the following benefits:

- Normalizes for imbalanced processor time allocation that is only an artifact of simulating a short span of execution (a few billion simulated cycles, days of simulation). These imbalances are eliminated over longer execution.
- Correctly accounts for the added overhead of hybrid resource management.
- Incorporates multiple, heterogeneous forms of computation into a single measure of application performance for any given span of execution, even when “time to completion” is not feasible to measure.

This work is detailed in [3].

2. CONCLUSIONS

The technical achievements of this work have the potential to influence system design of next-generation supercomputers and lower development effort to efficiently use the heterogeneous resources likely to be available in these systems. Next-generation CPUs, and therefore, supercomputers will be multicore, and these cores will likely be heterogeneous. The use of reconfigurable hardware to accelerate integer computation, coupled with effective OS scheduler support, could significantly increase application performance. .

Sandia's missions in nuclear security and scientific discovery and innovation require high performance science and engineering codes. A major obstacle to performance is the rate of memory addresses that can be generated. This work will address this bottleneck, and allow major improvements in the performance of Sandia's applications.

3. REFERENCES

1. Rupnow, K. and Compton. K, S*PY vs. SLY: Run-time thread-scheduler aware reconfigurable hardware allocators*, International Conference on Field-Programmable Technology, 2009. p353-356. FPT 2009. December 2009
2. Rupnow, K.. *Operating system management of reconfigurable hardware computing systems*. International Conference on Field-Programmable Technology, 2009. p477-478. FPT2009. December 2009
3. Rupnow, K, Adriaens, J., Fu, W. and Compton, K. *Accurately evaluating application performance in simulated hybrid multi-tasking systems*. FPGA2010. p134-144. February 2010.
4. Rupnow, K and Underwood, K. D. and Compton, K. *Scientific Application Demands on a Reconfigurable Functional Unit Interface*. ACM Transactions on Reconfigurable Technology. To Appear.

DISTRIBUTION

1	MS1319	James Ang	Org. Number 1422
1	MS1319	Arun Rodrigues	Org. Number 1422
1	MS0899	Technical Library	9536 (electronic copy)

For LDRD reports, add:

1	MS0359	D. Chavez, LDRD Office	1911

For CRADA reports add:

1	MS0115	OFA/NFE Agreements	10112

For Patent Caution reports, add:

1	MS0161	Legal Technology Transfer Center	11500


Sandia National Laboratories