A Reconfigurable Simulator for Large-scale Heterogeneous Multicore Architectures

Jiayuan Meng, Kevin Skadron
Department of Computer Science, University of Virginia

I. Introduction

Future general purpose architectures will scale to hundreds of cores. In order to accommodate both latency-oriented and throughput-oriented workloads, the system is likely to present a heterogenous mix of cores. In particular, sequential code can achieve peak performance with an out-of-order core while parallel code achieves peak throughput over a set of simple, in-order (IO) or single-instruction, multiple-data (SIMD) cores. These large-scale, heterogeneous architectures form a prohibitively large design space, including not just the mix of cores, but also the memory hierarchy, coherence protocol, and on-chip network (OCN).

Because of the abundance of potential architectures, an easily reconfigurable multicore simulator is needed to explore the large design space. We build a reconfigurable multicore simulator based on M5, an event-driven simulator originally targeting a network of processors.

II. Key Features

A number of simulators have been developed to simulate various architectures. However, they are all limited in their capability to simulate large-scale, heterogeneous chip multiprocessors (CMPs) with tens or hundreds of cores. Two well-known simulators that model individual out-of-order (OOO) cores and simultaneous multithreaded (SMT) cores are SimpleScalar [3] and SMTSIM [18], respectively. As modern architectures employ CMPs, several other simulators have been released. They include PTLsim[20], Sesc [14], Simics [9], Gems [10], and SimFlex [6]. While the above simulators work well on a particular set of architectures, the large design space in heterogenous multicore and manycore architectures demands both diversity and flexibility in simulation configurations, which is what MV5 emphasizes. This means that even components with fundamentally different design principles should be supported and able to work together. Unfortunately, none of the above simulators support array-style SIMD cores like those in graphics processors (GPUs), let alone the associate runtime system that manages SIMD threads.

On the other hand, publicly available GPU simulators (*e.g.* Qsilver [16], Atilla [5], and GPGPUsim [1]) lack several important components for general purpose CMP simulation. Not only do these simulators lack general purpose hardware models such as OOO cores, caches, and OCN, the software stack that cross-compiles general purpose codes into binaries with SIMD threads is also missing.

So far as we know, no previous simulators can simulate a general purpose architecture that integrates array-style SIMD cores, coherent caches and OCN—all are likely to be important components in future heterogeneous architectures. These modules, together with an OpenMP-like programming API [11] that compiles SIMD codes and a simulated runtime that manages SIMD threads, are provided in MV5. Specifically, the SIMD cores in MV5 fall into the category of the array style or single-instruction, multiple-threads (SIMT) paradigm, where homogeneous threads are implicitly executed on scalar datapaths operating in lockstep, and branch divergence across SIMD units can be handled by the hardware. Due to the lack of OS support for managing SIMT threads, MV5 currently only supports system emulation mode, and uses its own runtime threading library to manage SIMD threads. Given that the M5 simulator, which MV5 is based upon, already supports OOO and IO cores, the additional modules provided by MV5 complete the set of components needed for large-scale heterogeneous architecture simulations.

III. Power and Area Modeling

We use Cacti 4.2 [17] to calculate both the dynamic energy for reads and writes as well as the leakage power of the caches. We estimate energy consumption of cores using Wattch [2]. The pipeline energy is divided into seven parts including fetch and decode, integer ALUs, floating point ALUs, register files, result bus, clock and leakage. Dynamic energy is accumulated each time a unit is accessed. Power consumption of OCN's routers are modeled after the work of Pullini et al. [15]. We assume the physical memory consumes 220 nJ per access [7].

To have realistic area estimates, we measure the sizes of different functional units in an AMD Opteron processor in 130nm technology from a publicly available die photo. We do not account for about 30% of the total area, which is dedicated to x86-specific circuites. We scale the functional unit area to 65nm with a 0.7 scaling factor per generation. Final area estimates are calculated from their constituent units. We derive the L1 cache sizes from the die photo as well, and assume a 11 mm^2/MB area overhead for L2 caches. Our future work includes integrating MV5 with a more recent power and area modeling framework such as

McPAT [8].

IV. Examples of System Configurations

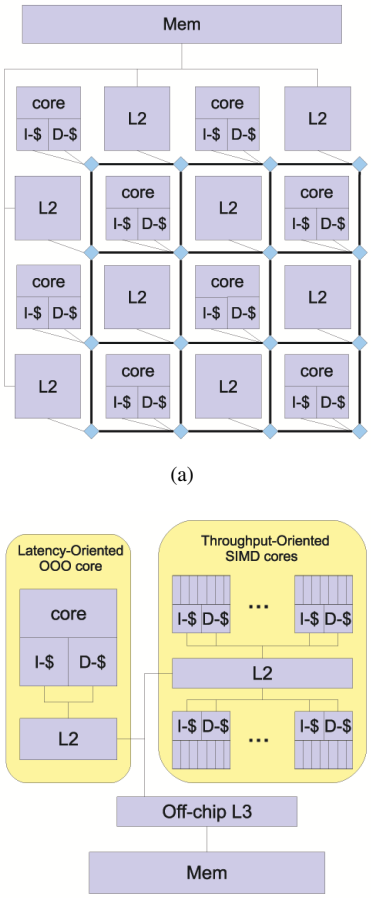

Fig. 1. Various system configurations: (a) tiled cores; (b) heterogeneous cores.

Figure 1 illustrates two examples of possible system configuration. Figure 1(a) shows a multicore architecture with eight IO cores that share a distributed L2 with eight banks through a 2-D mesh. Figure 1(b) demonstrates a heterogeneous multicore system with a latency-oriented OOO core, and a group of throughput oriented SIMD cores. The memory system contains two levels of on-chip caches and an off-chip L3 cache.

We have ported eight data-parallel benchmarks selected from Splash2 [19], Minebench [13], and Rodinia [4] to MV5's SIMD-compatible API. These benchmarks are re-implemented using our OpenMP-like programming API. SIMD cores can be configured with a SIMD width from one to 64, and the degree of multi-threading can be specified as well. Simulations have been conducted with up to 256 cores and up to 64 threads per core, operating over a directory-based coherent cache hierarchy with MESI protocol. MV5 was used to study a new technique for handling SIMD branch divergence and memory latency divergence, achieving an average speedup of 1.7X [12].

MV5 can be downloaded from https://sites.google.com/site/mv5sim/quick-start.

Acknowledgements This work was supported in part by SRC grant No. 1607, NSF grant nos. IIS-0612049 and CNS-0615277, a grant from Intel Research, and a professor partnership award from NVIDIA Research. We would like to thank Jeremy W. Sheaffer, David Tarjan, Shuai Che, and Jiawei Huang for their helpful inputs in power modeling, area estimation, and benchmark implementations.

References

[1] A. Bakhoda, G. L. Yuan, W. W. L. Fung, H. Wong, and T. M. Aamodt. GPGPU-Sim: A performance simulator for massively multithreaded processor. In *ISPASS*, 2009.
[2] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A Framework for Architectural-Level Power Analysis and Optimizations. In *ISCA 27*, June 2000.
[3] D. C. Burger, T. M. Austin, and S. Bennett. Evaluating future microprocessors: the SimpleScalar tool set. Tech. Report TR-1308, Univ. of Wisconsin, Computer Science Dept., July 1996.
[4] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, and K. Skadron. A performance study of general purpose applications on graphisc processors using CUDA. *JPDC*, 2008.
[5] V.M. del Barrio, C. Gonzalez, J. Roca, A. Fernandez, and Espasa E. ATTILA: a cycle-level execution-driven simulator for modern GPU architectures. *ISPASS*, 2006.
[6] N. Hardavellas, S. Somogyi, T. F. Wenisch, R. E. Wunderlich, S. Chen, J. Kim, B. Falsafi, J. C. Hoe, and A. G. Nowatzyk. SimFlex: a fast, accurate, flexible full-system simulation framework for performance evaluation of server architecture. *SIGMETRICS PER*, 31(4), 2004.
[7] I. Hur and C. Lin. A comprehensive approach to DRAM power management. *HPCA*, 2008.
[8] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi. Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *MICRO 42*, 2009.
[9] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hållberg, J. Högberg, F. Larsson, A. Moestedt, and B. Werner. Simics: A full system simulation platform. *Computer*, 35(2), 2002.
[10] M. M. K. Martin, D. J. Sorin, B. M. Beckmann, M. R. Marty, M. Xu, A. R. Alameldeen, K. E. Moore, M. D. Hill, and D. A. Wood. Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset. *CAN*, 33(4), 2005.
[11] J. Meng, J. W. Sheaffer, and K. Skadron. Exploiting inter-thread temporal locality for chip multithreading. In *IPDPS*, 2010.
[12] J. Meng, D. Tarjan, and K. Skadron. Dynamic warp subdivision for integrated branch and memory divergence tolerance. In *ISCA*, 2010.
[13] R. Narayanan, B. Ozisikyilmaz, J. Zambreno, G. Memik, and A. Choudhary. Minebench: A benchmark suite for data mining workloads. *IISWC*, 2006.
[14] P. M. Ortego and P. Sack. SESC: SuperESCalar Simulator. 2004.
[15] A. Pullini, F. Angiolini, S. Murali, D. Atienza, G. D. Micheli, and L. Benini. Bringing NoCs to 65 nm. *IEEE Micro*, 27(5), 2007.
[16] J. W. Sheaffer, D. Luebke, and K. Skadron. A flexible simulation framework for graphics architectures. *Graphics Hardware*, 2004.
[17] D. Tarjan, S. Thoziyoor, and N. P. Jouppi. Cacti 4.0. Technical Report HPL-2006-86, HP Laboratories Palo Alto, 2006.
[18] D. M. Tullsen. Simulation and modeling of a simultaneous multithreading processor. *CMG 22*, 1996.
[19] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta. The SPLASH-2 programs: Characterization and methodological considerations. *ISCA*, 1995.
[20] M.T. Yourst. PTLsim: A cycle accurate full system x86-64 microarchitectural simulator. *ISPASS*, April 2007.