CoreGenesis: Erasing Core Boundaries for Robust and Configurable Performance

Shantanu Gupta, Shuguang Feng, Amin Ansari, Ganesh Dasika and Scott Mahlke
Advanced Computer Architecture Laboratory
University of Michigan - Ann Arbor, MI
{shangupt, ansary, shoe, gdasika, mahlke} @umich.edu

ABSTRACT

Single-thread performance, power efficiency and reliability are critical design challenges of future multicore systems. Although point solutions have been proposed to address these issues, a more fundamental change to the fabric of multicore systems is necessary to seamlessly combat these challenges. Towards this end, this paper proposes CoreGenesis, a dynamically adaptive multiprocessor fabric that blurs out individual core boundaries, and encourages resource sharing across cores for performance, reliability and customized processing. Further, as a manifestation of this vision, the paper provides details of a unified performance-reliability solution that can assemble variable-width processors from a network of (potentially broken) pipeline stage-level resources.

Categories and Subject Descriptors

B.8.1 Hardware [**Performance and Reliability**]: Reliability, Testing, and Fault-Tolerance; C.1.4 Computer System Organization [**Processor Architectures**]: Parallel Architectures

General Terms

Performance, Reliability, Design

Keywords

Chip Multiprocessors, Fault Tolerance, Configurable Performance, Reconfigurable Architectures

1. INTRODUCTION

Motivation. As a result of growing power, thermal and complexity concerns in monolithic processors, major hardware vendors have lead a migration to multicore processors composed of relatively simple cores. Today, the sizes of multicores vary from 2-6 cores in desktop systems to 16-64 cores in throughput-oriented computing domains, e.g. SUN UltraSparc T1/T2, Intel Larrabee and Tilera TILE64. Despite having been attenuated, several challenges pertaining to performance, power and reliability still remain in the multicore paradigm. First, multiple cores are effective for throughput computing, but they provide little to no gains for sequential applications. Even if a major transition towards parallel programming occurs in the future, Amdahl's law dictates that the sequential component of an application will present itself as a performance bottleneck. Second, power constraints limit the number of cores / resources that can be kept active on a chip, motivating the need for customized and power-proportional processing. Finally, the increasing vulnerability of transistors with each passing technology generation, can jeopardize the objective of throughput sustainability over the lifetime of a multicore chip.

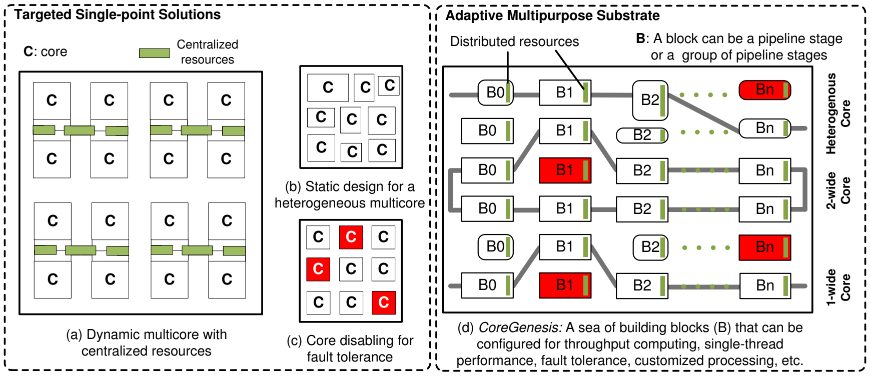
Prior Work. In this landscape of multicore challenges, prior research efforts have focused on addressing these issues in isolation. For example, to tackle single-thread performance, a recent article by Hill and Marty [2] introduces the concept of *dynamic multicores* (Figure 1(a)) that can allow multiple cores on a chip to work in unison while executing sequential codes. This notion of *configurable performance* allows chips to efficiently address scenarios requiring throughput computing, high sequential performance, and anything in between. Core Fusion [3], Composable Lightweight Processors and Federation are representative works with this objective. However, the scope of present day *dynamic multicore* solutions is limited as they cannot provide customized processing, as in [4], or better throughput sustainability, as achieved by techniques in [1]. The customized processing is typically accomplished by introducing heterogeneity of types and number of functional units, execution models (in-order, OoO), prediction structures, etc., into different cores (Figure 1(b)), in order to run workloads in a power-proportional fashion [4]. Whereas, better throughput sustainability can be provided by fine-grained reliability solutions like StageNet [1], that disable broken pipeline stages, instead of entire cores (Figure 1(c)), within a multicore.

Despite their abundance, by virtue of being independent efforts, combining existing performance, power and reliability solutions for multicores is neither cost-effective nor straightforward. The overheads quickly become prohibitive as the changes required for each solution are introduced, with very little that can be amortized across multiple techniques. Configurable performance requires dedicated centralized structures (adding drawbacks such as access contention/latency, global wiring, single point of failure), customization requires a variety of static core designs, and fine-grained reliability requires either large amounts of area for cold spares or the flexibility to share resources across cores.

2. COREGENESIS

Instead of targeting one challenge at a time, the goal of this proposal is to devise a design philosophy that can naturally be extended to handle a multitude of multicore challenges seamlessly, while overlapping costs, maintaining efficiency and avoiding centralized structures. Towards this end, we propose the CoreGenesis (CG) architecture (see Figure 1(d)), an adaptive computing substrate that is inherently flexible, and can best align itself to the immediate

PACT'10, September 11–15, 2010, Vienna, Austria.
ACM 978-1-4503-0178-7/10/09.


Targeted Single-point Solutions
C: core
Centralized resources
C
Adaptive Multipurpose Substrate
Distributed resources
B: A block can be a pipeline stage or a group of pipeline stages
B0
B1
B2
Bn
Heterogenous Core
2-wide Core
1-wide Core

(a) Dynamic multicore with centralized resources

(b) Static design for a heterogeneous multicore

(c) Core disabling for fault tolerance

(d) *CoreGenesis:* A sea of building blocks (B) that can be configured for throughput computing, single-thread performance, fault tolerance, customized processing, etc.

Figure 1: Contemporary solutions for multicore challenges (a,b,c) and vision of this work (d). In (a), centralized resources are used to assist in fusing neighboring cores. In (b) and (d), different shapes/sizes denote heterogeneity. In (c) and (d), dark shading marks broken components.

system needs. CG eliminates the traditional core boundaries and organizes the chip multiprocessor as a dynamically configurable network of building blocks. This sea of building blocks can be symmetric or heterogeneous in nature, while varying in granularity from individual pipeline stages to groups of stages. Further, the CG pipeline microarchitecture is decoupled at block boundaries, providing full flexibility to construct logical processors from any complete set of building blocks. Another key feature of the CG proposal is the use of distributed resources to coordinate instruction execution across decoupled blocks, without any significant changes to the ISA or the execution model. This is a major advancement over prior efforts, and addresses the shortcomings of centralized resources.

Resources from CG's sea of blocks can be fluidly allocated for a number of performance, power and reliability requirements. Throughput computing can be optimized by forming many single-issue pipelines, whereas sequential performance can be accelerated by forming wider-issue pipelines. Power and performance characteristics can be further improved by introducing heterogeneous building blocks in the fabric, and appropriately configuring them (dynamically or statically) for active program phases or entire workloads. This enables a dynamic approach to customized processing. Finally, fault tolerance in CG can be administered at the block granularity, by disabling the broken components over time.

To better understand the benefits and challenges in realizing this architectural vision, we designed and evaluated a CG instance that targets configurable performance and fine-grained reliability. For the fabric, an in-order pipeline model is used with single pipeline stages as its building blocks. As a first step, we defined mechanisms for decoupling pipeline stages from one another (inspired by the StageNet architecture [1]). This enabled salvaging of working stages from different rows of the fabric to form logical processors, thereby tackling the throughput sustainability challenge. To address configurable performance, we generalized the notion of logical processors to form processors of varying issue widths.

The engineering of distributed resources to support the assembly of decoupled pipeline stages into a wide-issue processor is especially hard due to the heavy co-ordination and communication requirements of an in-order superscalar. We adopted a best effort strategy here, speculating on control and data dependencies across pipeline ways, and falling back to a light-weight replay in case of a violation. To register these violations, hardware schemes were formulated for distributed control, register and memory data flow management. The frequency of data flow violations from instructions executing on two different pipeline ways was found to be a leading cause of performance loss. This was addressed by incorporating compiler hints for instruction steering in the program binary.

Overall, this manifestation of CG relies on interconnection flexibility, microarchitectural innovations, and compiler directed instruction steering, to provide a unified performance-reliability solution. The chip consists of a pool of pipeline stage-level resources that can be fluidly allocated for accelerating single-thread performance, throughput computing, or tolerating failures. Performance simulations show that merging pipelines within CoreGenesis improves single-thread IPC by 1.5X, on average, over a standalone pipeline. On the other hand, lifetime reliability experiments demonstrate that an 8-core CoreGenesis chip increases the cumulative work done by 41% over a traditional 10-core CMP (area-neutral comparison).

3. ACKNOWLEDGMENTS

The authors acknowledge the support of the Gigascale Systems Research Center, one of five research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation program. This research was also supported by National Science Foundation grants CCF-0916689 and ARM Limited.

4. REFERENCES

[1] S. Gupta, S. Feng, A. Ansari, J. A. Blome, and S. Mahlke. The stagenet fabric for constructing resilient multicore systems. In *Proc. of the 41st Annual International Symposium on Microarchitecture*, pages 141–151, 2008.

[2] M. D. Hill and M. R. Marty. Amdahl's law in the multicore era. *IEEE Computer*, 41(1):33–38, 2008.

[3] E. Ipek, M. Kirman, N. Kirman, and J. Martinez. Core fusion: Accommodating software diversity in chip multiprocessors. In *Proc. of the 34th Annual International Symposium on Computer Architecture*, pages 186–197, 2007.

[4] R. Kumar, K. I. Farkas, N. P. Jouppi, P. Ranganathan, and D. M. Tullsen. Single-ISA Heterogeneous Multi-Core Architectures: The Potential for Processor Power Reduction. In *Proc. of the 36th Annual International Symposium on Microarchitecture*, pages 81–92, Dec. 2003.