7.4 Runtime memory optimization and GPU/manycore architectures

**Date:** Wednesday, March 26, 2014  
**Time:** 14:30 - 16:00  
**Location / Room:** Konferenz 2

Chair:  
Alberto Nannarelli, DTU Copenhagen, DK, Contact Alberto Nannarelli

Co-Chair:  
Alberto Macii, PoliTo Torino, IT, Contact Alberto Macii

The session starts with memory design techniques under PVT variation and ageing for DRAMs and SRAM caches. Afterwards, bus, memory and partitioning techniques for 2D and 3D GPUs and manycores are presented.

<table>
<thead>
<tr>
<th>Time</th>
<th>Label</th>
<th>Presentation Title</th>
<th>Authors</th>
</tr>
</thead>
</table>
| 14:30 | 7.4.1 | EXPLOITING EXPENDABLE PROCESS-MARGINS IN DRAMS FOR RUN-TIME PERFORMANCE OPTIMIZATION | (Paper/SoftConf ID: 763)  
Speakers:  
Karthik Chandrasekar¹, Sven Goossens², Christian Weis³, Martijn Koedam³, Benny Akesson⁴, Norbert Wehn³ and Kees Goossens⁵  
¹Delft University of Technology, NL; ²Eindhoven University of Technology, NL; ³University of Kaiserslautern, DE; ⁴Czech Technical University in Prague, CZ; ⁵Eindhoven university of technology, NL |

Abstract  
Manufacturing-time process (P) variations and runtime variations in voltage (V) and temperature (T) can affect a DRAM’s performance (internal delays) severely. To counter the effects of these variations, DRAM vendors provide substantial design-time PVT margins to guarantee correct DRAM functionality under worst-case conditions. Unfortunately, with technology scaling these design margins have become large and very pessimistic for a majority of the manufactured DRAMs. While run-time variations are specific to operating conditions and their margins difficult to optimize, process variations are manufacturing-time effects and excessive process-margins can be reduced on a per-device basis, if properly identified. In this paper, we propose a generic post-manufacturing performance characterization methodology for DRAMs that identifies this excess in process-margins for any given DRAM device at runtime, while retaining the requisite margins for voltage (noise) and temperature variations. By doing so, the methodology ascertains the actual impact of process-variations on the particular DRAM device and optimizes its access latencies, thereby improving its overall performance. We evaluate this methodology on 48 DDR3 devices (from 12 DIMMs) and verify the derived timings under worst-case operating conditions, showing up to 33.3% and 25.9% reduction in DRAM read and write latencies, respectively.

<table>
<thead>
<tr>
<th>Time</th>
<th>Label</th>
<th>Presentation Title</th>
<th>Authors</th>
</tr>
</thead>
</table>
| 15:00 | 7.4.2 | CACHE AGING REDUCTION WITH IMPROVED PERFORMANCE USING DYNAMICALLY RE-SIZABLE CACHE | (Paper/SoftConf ID: 604)  
Speakers:  
Haroon Mahmood, Massimo Poncino and Enrico Macii, Politecnico di Torino, Torino Italy, IT |

Abstract  
Aging of transistors is a limiting factor for long term reliability of devices in sub-100nm technologies. It’s a worst-case metric where the lifetime of a device is determined by the earliest failing component. Impact is more serious on memory arrays, where failure of a single SRAM cell would cause the failure of the whole system. Previous works have shown that partitioning based strategies based on power management techniques can effectively control aging effects and can extend lifetime of the cache significantly. However, such a benefit comes as a trade-off with performance which reduces proportionally as the time elapses. To address this problem and provide a single solution to concurrently improve aging, energy and performance of the cache, we propose an architectural solution based on the dynamically re-sizable cache and cache partitioning approaches. By this strategy, cache is dynamically re-sized and reconfigured whenever a cache block becomes unreliable. Coupling such aging mitigation technique along with dynamically re-sizable cache approach provides on average 30% lifetime improvement with less than 0.4x degradation.
**Abstract**

The complex buses consume significant power in graphics processing units (GPUs). In this paper, we demonstrate how the power consumption of buses in GPUs can be reduced with 3D IC technologies. Based on layout simulations, we found that partitioning and floorplanning of 3D ICs affect the power benefit amount, as well as the technology setup, target clock frequency, and circuit switching activity. With 3D IC technologies, we achieved the total power reduction of up to 21.5% for our GPU.

**Abstract**

High-level programming languages have transformed graphics processing units (GPUs) from domain-restricted devices into powerful compute platforms. Yet many "general-purpose GPU" (GPGPU) applications fail to fully utilize the GPU resources. Executing multiple applications simultaneously on different regions of the GPU (spatial multitasking) thus improves system performance. However, within-die process variations lead to significantly different maximum operating frequencies (Fmax) of the streaming multiprocessors (SMs) within a GPU. As the chip size and number of SMs per chip increase, the frequency variation is also expected to increase, exacerbating the problem. The increased number of SMs also provides a unique opportunity: we can allocate resources to concurrently-executing applications based on how those applications are affected by the different available Fmax values. In this paper, we study the effects of per-SM clocking on spatial multitasking-capable GPUs. We demonstrate two factors that affect the performance of simultaneously-running applications: (i) the SM partitioning algorithm that decides how many resources to assign to each application, and (ii) the assignment of SMs to applications based on the operating frequencies of those SMs and the applications characteristics. Our experimental results show that spatial multitasking that partitions SMs based on application characteristics, when combined with per-SM clocking, can greatly improve application performance by up to 46% on average compared to cooperative multitasking with global clocking.

**Abstract**

One memory-logic-integration design platform is developed in this paper with thermal reliability analysis provided for 2.5D through-silicon-interposer (TSI) and 3D through-silicon-via (TSV) based integrations. Temperature-dependent delay and power models have been developed at microarchitecture level for 2.5D and 3D integrations of many-core microprocessors and main memory, respectively. Experiments are performed by general-purpose benchmarks from SPEC CPU2006 and also cloud-oriented benchmarks from Phoenix with the following observations. The memory-logic integration by 3D RC-interconnected TSV I/Os can result in thermal runaway failures due to strong electrical-thermal couplings. On the other hand, the one by 2.5D transmission-line-interconnected TSI I/Os has shown almost the same energy efficiency and better thermal resilience.

**Abstract**

Coherent data prediction is introduced as a promising architectural technique for reducing cache-to-cache accesses in directory protocols. However, limited on-chip resources cause the accuracy of current prediction to be generally low. Low accuracy would result in a large number of unnecessary or incorrect predictions, which would consequently generate excessive network traffic. This leads to large power and performance overhead for coherent memory access. This paper proposes an early abort mechanism (EBT) that leverages NoC design to reduce the negative effect of wrong prediction operations, thus facilitating overall performance improvement and traffic reduction. Using detailed full-system simulations, we conclude that EBT provides a cost-effective solution for designing efficient multicore processors. To the best of our knowledge, this study is the first to leverage on-chip network for the prediction optimization on multicore coherence.

**Abstract**

A thermal resilient integration of many-core microprocessors and main memory by 2.5D TSI I/Os. The complex buses consume significant power in graphics processing units (GPUs). In this paper, we demonstrate how the power consumption of buses in GPUs can be reduced with 3D IC technologies. Based on layout simulations, we found that partitioning and floorplanning of 3D ICs affect the power benefit amount, as well as the technology setup, target clock frequency, and circuit switching activity. With 3D IC technologies, we achieved the total power reduction of up to 21.5% for our GPU.
Abstract

Data mining, bioinformatics, knowledge discovery, social network analysis, are emerging irregular applications that exploit data structures based on pointers or linked lists, such as graphs, unbalanced trees or unstructured grids. These applications are characterized by unpredictable memory accesses and generally are memory bandwidth bound, but also presents large amounts of inherent dynamic parallelism because they can potentially spawn concurrent activities for each one of the element they are exploring. Hybrid architectures, which integrate general purpose processors with reconfigurable devices, appears promising target platforms for accelerating irregular applications. These systems often connect to distributed and multi-ported memories, potentially enabling parallel memory operations. However, these memory architectures introduce several challenges, such as the necessity to manage concurrency and synchronization to avoid structural conflicts on shared memory locations and to guarantee consistency. In this paper we present an adaptive Memory Interface Controller (MIC) that addresses these issues. The MIC is a general and customizable solution that can target several different memory structures, and is suitable for High Level Synthesis frameworks. It implements a dynamic arbitration scheme, which avoids conflicts on memory resources at runtime, and supports atomic memory operations, commonly exploited for synchronization directives in parallel programming paradigms. The MIC simultaneously maps multiple accesses to different memory ports, allowing fine grained parallelism exploitation and ensuring correctness also in the presence of irregular and statically unpredictable memory access patterns. We evaluated the effectiveness of our approach on a typical irregular kernel, graph Breadth First Search (BFS), exploring different design alternatives.