# Parallelization, Customization and Automation

Jason Cong

Director, Center for Domain-Specific Computing
www.cdsc.ucla.edu

Chancellor's Professor
UCLA Computer Science Department
http://cadlab.cs.ucla.edu/~cong

**Abstract - In order to meet today's ever-increasing computing needs and overcome power density limitations, the computing industry has halted simple processor frequency scaling and entered the era of parallelization, with tens to hundreds of computing cores integrated in a single processor, and hundreds to thousands of computing servers connected in a warehouse-scale data center. However, such highly parallel, general-purpose computing systems still face serious challenges in terms of performance, power, heat dissipation, space, and cost. We believe that we need to look beyond parallelization and focus on domain-specific customization to provide capabilities that adapt architecture to application in order to achieve significant power-performance efficiency improvement. This paradigm shift requires a great deal of innovation in architecture, compilation, and runtime system design, and offers many exciting and challenging research opportunities. I shall discuss the research progress in this direction and its implications in the EDA industry.**

## I. Introduction

In order to meet today's ever-increasing computing needs and overcome power density limitations, the computing industry has halted simple processor frequency scaling and entered the era of parallelization, with tens to hundreds of computing cores integrated in a single processor, and hundreds to thousands of computing servers connected in a warehouse-scale data center. However, such highly parallel, general-purpose computing systems still face serious challenges in terms of performance, power, heat dissipation, space, and cost. We believe that we need to look beyond parallelization and focus on domain-specific customization to provide capabilities that adapt architecture to application in order to achieve significant power-performance efficiency improvement. Many decisions need to be made during the architecture and microarchitecture designs:

- For computing elements, shall we use a few powerful cores or many simple cores? What is the balance between general-purpose cores versus special-purpose accelerators?
- For on-chip memory, shall we use cache (managed by hardware) or scratchpad memory (managed by the application program)?
- For on-chip interconnects, what topology (e.g., bus, point-to-point, mesh-like network-on-chip) and bandwidth shall we provide?

The truth is that there is no best answer to these questions. The optimal choice depends on the application domain. We must develop the capability to adapt the architecture to the applications so that performance and power efficiency are optimized. In this talk I shall share the progress made in the Center for Domain-Specific Computing (CDSC) [1] on developing a customizable heterogeneous platform (CHP) that provides this customization capability.

## II. CHP Overview

*CHP* consists of a heterogeneous set of *adaptive* computational resources connected with high-bandwidth, low-power non-traditional reconfigurable interconnects. Specifically, a CHP includes 1) integration of customizable cores and co-processors that will enable power-efficient performance tuned to the specific needs of an application domain; and 2) reconfigurable high-bandwidth and low-latency on- and off-chip interconnects, such as RF-interconnects, which can be customized to specific applications. Figure 1 illustrates an example CHP configuration with a set of fixed cores, customizable cores, accelerators, programmable fabric, and a set of distributed cache banks ($).

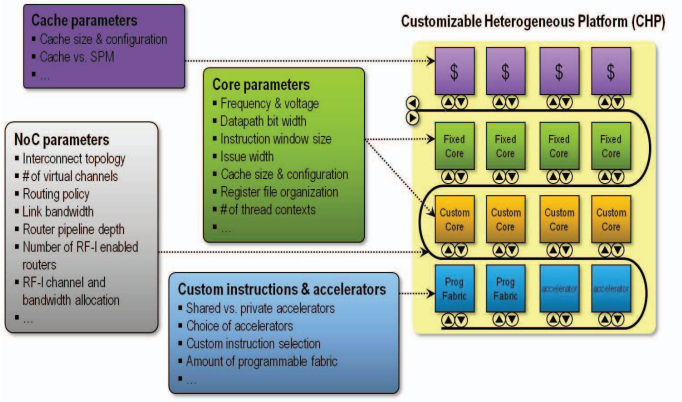


**Fig. 1. CHP illustration**

## III. Recent Progress on CHP Design

I shall highlight some of our results related to the CHP designs.

- We developed a novel approach to dynamically accelerate the performance of sequential application(s) on multiple cores. Execution is allowed to spill from one core to another when resources on one core have been exhausted. We proposed two techniques to enable low-overhead migration between cores: prespilling and locality-based filtering. We developed and analyzed an arbitration mechanism to intelligently allocate cores among a set of sequential applications on a CMP. On average, core spilling on an eight-core CMP can accelerate single-threaded performance by 35 percent. We further explored an eight-core CMP running a multiple application workload composed of the entire SPEC 2000 benchmark suite in various combinations and arrival times. By using core spilling to accelerate the current set of running applications in cases where there are idle cores, we achieved up to a 40% improvement in performance [2]. This shows that we can dynamically "fuse" several simple cores to form a powerful core.

- **Accelerator-rich architecture:** We have implemented AXR-CMP [3], a hardware architectural support for accelerator-rich CMPs. First, we implemented a hardware resource management scheme for accelerator sharing. This scheme supports sharing and arbitration of multiple cores for a common set of accelerators, and it uses a software-based priority mechanism to provide feedback to cores to indicate the wait time before a particular resource becomes available. Second, we proposed a lightweight interrupt system to reduce the OS overhead of handling interrupt—which happens often in an accelerator-rich platform. Third, we proposed an architectural support that allows us to compose a larger virtual accelerator out of multiple smaller accelerators, and chain multiple accelerators together with minimal intervention of the requesting core. We also implemented a complete simulation tool-chain to verify our AXR-CMP architecture. Experimental results show significant performance and energy improvement compared to approaches that use OS-based accelerator management. We achieved an average 123X in performance (up to 208X) and 112X in energy efficiency (up to 180X) over a software implementation, with minimal hardware overhead [3].

- **Adaptive hybrid L1 cache:** By reconfiguring part of the cache as software-managed scratchpad memory (SPM), hybrid caches manage to handle both unknown and predictable memory access patterns. However, existing hybrid caches provide a flexible partitioning of cache and SPM without considering adaptation to the run-time cache behavior. Previous cache set balancing techniques are either energy-inefficient or require serial tag and data array access. Recently, we developed an adaptive hybrid L1 cache that dynamically remaps SPM blocks from high-demand cache sets to low-demand cache sets. This achieves 19%, 25%, 18% and 18% energy-runtime-production reductions over four previous representative cache optimization techniques on a wide range of benchmarks [4].

- **Reconfigurable L2 cache with mixed memory technologies:** We proposed a novel reconfigurable hybrid cache architecture (RHC) in which NVM is incorporated in the L2 cache together with SRAM. RHC can be reconfigured by powering on/off SRAM/NVM arrays in a way-based manner. We investigated both the architecture and circuit design issues for RHC. Furthermore, we developed hardware-based mechanisms to dynamically reconfigure RHC on-the-fly based on the cache demand. Experimental results on a wide range of benchmarks show that the proposed RHC achieves an average 63%, 48% and 25% energy saving over non-reconfigurable SRAM-based L2 cache, non-reconfigurable hybrid L2 cache, and reconfigurable SRAM-based L2 cache, while maintaining the system performance (at most a 4% performance overhead) [5].

- **Customizable network-on-chip (NoC) designs:** On-chip interconnects need to deliver high bandwidth to node pairs with high communication demands. However, most often the communication patterns are either not known in advance, or change frequently during execution. Therefore, NoCs are usually over-designed to cover the worst-case, which happens rarely, with large area and power overhead. We showed that promising gains can be realized via integration of radio frequency interconnect (RF-I) through on-chip transmission lines with traditional interconnects implemented with RC wires. In addition to the latency advantage of RF-I, we demonstrated three further advantages of RF-I: 1) RF-I bandwidth can be flexibly allocated to provide an adaptive NoC, 2) RF- I can enable a dramatic power and area reduction by simplification of NoC topology, and 3) RF-I provides natural and efficient support for multicast. Based on these observations, we proposed a novel NoC design, exploiting dynamic RF-I bandwidth allocation to realize a reconfigurable network-on-chip architecture. We found that our adaptive RF-I architecture on top of a mesh with 4B links can match or even outperform the baseline with 16B mesh links, but reduces NoC power by approximately 65% including the overhead incurred for supporting RF-I [6].

As an example, we have chosen our application domain to be medical imaging, and we are in the process of finalizing the CHP design for this domain.

## IV. Opportunities for EDA

Although good progress has been made, our proposed customizable domain-specific computing methodology requires a great deal more innovations in architecture,

compilation, and runtime system design, and offers many exciting and challenging research opportunities. For example, the following are some of the open issues associated with CHP design and mapping:

- We need tools to automatically identify accelerator candidates. We also need the tools to automatically synthesize the accelerators once such candidates are identified (the C-to-RTL synthesis tools, such as xPilot [7] and AutoPilot [8], are part of the solution). Furthermore, we need tools to provide accelerator virtualization — that is, using available physical accelerators to implement more complex accelerators of the same or similar types (e.g., implementing a 1024-point FFT function using a 128-point FFT accelerator).
- We need compilation support to efficiently use the hybrid L1 cache with SPMs and the reconfigurable L2 cache with a mix of SRAM cache blocks and STT-RAM based cache blocks. We made some progress in this area [9], but more studies are needed.
- We need tools to synthesize customizable NoCs with RF-interconnects to decide how to dynamically add shortcuts and construct the route. For example, our work in [10] shows that routing in an irregular NoC (resulting from adding a RF-bus to the underlying mesh-based NoC) may deadlock, and we developed efficient NoC construction and routing algorithms to avoid the deadlock.
- Finally, simulation of such customizable heterogeneous is a large challenge. The existing architecture simulators (e.g., [11][12]) cannot model the complexity of CHP and nor provide sufficient simulation speed.

## Acknowledgements

The Center for Domain-Specific Computing (CDSC) is funded by the NSF Expedition in Computing Award CCF-0926127. Most of the summarized in this paper are the joint work with CDSC faculty and students. The list of CDSC faculty and students is available from www.cdsc.ucla.edu .

## References

[1] J. Cong, V. Sarkar, G. Reinman and A. Bui, "Customizable Domain-Specific Computing," *IEEE Design and Test of Computers*, Volume 28, Number 2, pp. 5-15, 2011.

[2] J. Cong, G. Han, A. Jagannathan, G. Reinman, and K. Rutkowski, "Accelerating Sequential Applications on CMPs Using Core Spilling," *IEEE Transactions on Parallel and Distributed Systems*, Volume 18, Number 8, pp. 1094- 1107, August 2007.

[3] J. Cong, M. A. Ghodrat, M. Gill, C. Liu, G. Reinman and Y. Zou, "AXR-CMP: Architecture Support in Accelerator-Rich CMPs," *Proceedings of the 2nd Workshop on SoC Architecture, Accelerators and Workloads (SAW-2)*, February 2011.

[4] J. Cong, K. Gururaj, H. Huang, C. Liu, G. Reinman and Y. Zou, "An Energy-Efficient Adaptive Hybrid Cache," In *Proc. of ISLPED*, pp. 67-72, 2011.

[5] Y. Chen, J. Cong, H. Huang, B. Liu, C. Liu, M. Potkonjak and G. Reinman, "Dynamically Reconfigurable Hybrid Cache: An Energy-Efficient Last-Level Cache Design," *to appear in Proc. of DATE*, 2012.

[6] M. F. Chang, J. Cong, A. Kaplan, C. Liu, M. Naik, J. Premkumar, G. Reinman, E. Socher, and R. Tam, "Power Reduction of CMP Communication Networks via RF Interconnects," In *Proc. of MICRO*, pp. 376-387, 2008.

[7] J. Cong, Y. Fan, G. Han, W. Jiang, and Z. Zhang, "Platform-Based Behavior-Level and System-Level Synthesis," In *Proc. of IEEE International SOC Conference*, pp. 199-202, 2006.

[8] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers and Z. Zhang, "High-Level Synthesis for FPGAs: From Prototyping to Deployment," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Volume 30, Number 4, pp. 473-491, 2011.

[9] J. Cong, H. Huang, C. Liu and Y. Zou, "A Reuse-Aware Prefetching Algorithm for Scratchpad Memory," In *Proc. of DAC*, pp. 960-965, 2011.

[10] J. Cong, C. Liu and G. Reinman, "ACES: Application-Specific Cycle Elimination and Splitting for Deadlock-Free Routing on Irregular Network-on-Chip," In *Proc. of DAC*, pp. 443-448, June 2010.

[11] D. Burger and T. M. Austin, "The SimpleScalar Tool Set, Version 2.0," Computer Architecture News, pp 13-25, 1997.

[12] M. Martin, D. Sorin, B. Beckmann, M. Marty, M. Xu, A. Alameldeen, K. Moore, M. Hill, and D. Wood. "Multifacet's General Execution-Driven Multiprocessor Simulator (GEMS) Toolset," In *Computer Architecture News*, pp. 92-99, 2005.