FabScalar-RISCV

Rangeen Basu Roy Chowdhury, Anil Kumar Kannepalli, Eric Rotenberg
North Carolina State University, Raleigh, NC, 27695, USA {rbasuro@ncsu.edu}

1 Introduction

FabScalar is a toolset for automatically generating synthesizable register-transfer-level (RTL) descriptions of arbitrary superscalar cores within a canonical superscalar template. The cores differ in three major superscalar dimensions: fetch/issue widths, pipeline depth, and sizes of units involved in exposing instruction-level parallelism (ILP) (issue queue, load and store queues, physical register file, reorder buffer, *etc*.).

FabScalar was conceived to reduce the design and verification effort of single-ISA heterogeneous multi-core processors, which are comprised of many microarchitecturally diverse core types [3]. Furthermore, superscalar processor design automation spurs innovation in today's highly stratified computing market, by streamlining the production of specialized processors and opening such ventures to many smaller players [4]. (We discuss forces for processor diversification in Appendix A.) These principles are shared by the RISC-V open-ISA movement [19]. In addition, FabScalar is used by many researchers worldwide. Since its beta release in 2010 and first major publication in 2011, 230 researchers, from 33 U.S. universities, 45 international universities, 6 industry sites, and 22 countries, have downloaded the FabScalar toolset. (Detailed user data is provided in Appendix B.) Finally, several chips have been fabricated using the FabScalar toolset [8]. (Another chip from Mei University is pending publication.)

The released FabScalar toolset implements the SimpleScalar PISA ISA [2]. It is MIPS-like, circa 1996. There are two problems with PISA. First, there is no longer a software ecosystem for it. The gcc compiler for PISA is severely outdated and fails to compile many SPEC 2006 benchmarks, and there is no Fortran compiler for PISA. Second, PISA does not have a specification for a system co-processor ISA, such as the MIPS co-processor 0, which is needed for all system-level support (interrupts, MMU) and running linux kernels. We have ported FabScalar to MIPS64, including implementations of co-processor 0 ISA (interrupts, MMU) [14] and co-processor 1 ISA (floating-point) [16]. Unfortunately, based on cautions by Asanovic and Patterson in their EE Times article [19], FabScalar-MIPS64 could expose NCSU to litigation (MIPS IP was sold to a UK firm).

RISC-V is the ideal replacement to PISA. It is open source. There is commitment to maintaining the software ecosystem (compilers, kernels). It is a truly minimal ISA amenable to complex microarchitectures (no delay slots, no predication, *etc*.). It uses a strategy of optional co-processor ISA extensions for floating-point, SIMD, MMU, and accelerators.

Thus, RISC-V solves key problems that have held back the FabScalar toolset from reaching its full potential. Moreover, there have been many improvements to FabScalar since the beta release, which have not been publicly released owing to ISA indecision. These will become available for the first time with the FabScalar-RISCV release.

We look forward to the opportunity to present FabScalar-RISCV, our port of the FabScalar toolset to the RISC-V ISA and software ecosystem.

2 FabScalar Background and Recent Improvements

FabScalar's core generator has three parts [3]. Its Canonical Superscalar Template defines canonical pipeline stages and interfaces among them. A Canonical Pipeline Stage Library (CPSL) provides many implementations of each canonical pipeline stage, that differ in their superscalar width and depth of sub-pipelining. An RTL generation tool references the template and CPSL to automatically generate an overall core of desired configuration.

FabScalar includes two other tools: FabMem [15] and FabFPGA [6]. Since highly-ported RAMs and CAMs are prevalent in superscalar processors and significantly impact area, power, and cycle time, the

FabMem tool was developed for automatically generating the physical designs (layouts) of multiported RAMs and CAMs. In contrast, commercial memory compilers are limited to a modest number of ports. FabFPGA is a configurable, automatically FPGA-synthesizable, and register-transfer-level (RTL) model of an out-of-order superscalar processor (superset core). FabFPGA enables FPGA modeling of diverse superscalar processors out-of-the-box. Moreover, its direct RTL implementation yields the fidelity of a hardware prototype.

There have been many enhancements to FabScalar since the beta release. These will become available for the first time with the release of FabScalar-RISCV. For example:

1. *Superset core*: FabScalar's previous approach of "building up" cores from a library of stage designs implies that propagating a change to arbitrary core configurations requires reimplementing it in each stage design. When FabFPGA was developed, FabScalar was transformed into a single highly-parameterized System Verilog design, called the superset core. Changes to the superset core, if done properly, extend to all of its possible configurations.
2. *SoC and multi-core support*: Key partners from Mei University, Japan, developed highly-parameterized RTL models of an AMBA bus (FabBus) and coherent L1 and L2 caches (FabCache) [47]. These models extend the FabScalar toolset to SoCs and multi-core processors.
3. *Crowd-sourcing with GitHub*: Research productivity of the entire community will dramatically improve if users can commit their changes to a community-shared design (as well as EDA scripts, *etc*.), and if other users can "cherry pick" desired features. We have begun using NCSU GitHub internally and with our partners from Mei University (developers of FabBus and FabCache). We will go live on GitHub with the release of FabScalar-RISCV.

3 Status of FabScalar-RISCV

The RISC-V port of the FabScalar RTL model is progressing well. It successfully runs small benchmarks, as they do not require a sophisticated verilog testbench to launch them. The limiter for testing larger benchmarks is not the RTL model itself. The limiter is replacing FabScalar's C++/verilog co-simulation environment with the RISC-V equivalent, referred to as the "tethered test harness". This term refers to using the Host Target Interface (HTIF) for program loading and system calls.

The FabScalar microarchitecture was originally designed for a RISC ISA. Consequently, minimal modifications were required to port it to RISC-V. Owing to the hierarchical instruction encoding of RISC-V, changes had to be made in the way instructions are decoded (both in the Decode stage and the Execution Lanes). The simpler encoding of the immediates and the fixed positions of the source and destination registers, made the logic less complex. Integer and floating-point are unified (unified issue queue and physical register file) owing to RISC-V being a 64-bit architecture.

We also developed a cycle-accurate C++ simulator which successfully runs the SPEC2006 benchmarks. The C++ simulator uses a tethered test harness similar to the Berkeley designs, *i.e.*, program loading and system calls are handled via the Host Target Interface (HTIF) as explained earlier. The same tethered test harness will be implemented in the verilog testbench.

In order to support system software, we will make both the RTL model and the C++ simulator fully compliant with the latest version of the RISC-V privileged ISA. FabFPGA will also be ported to RISC-V and released along with the other tools.

4 References

Please see Appendix C.

Appendix A: Forces for processor diversification

After years of consolidation in the desktop market, the processor business is experiencing a wave of diversification. Outwardly, this is driven by smart phones. Qualcomm designs its Snapdragon application processors. Recently, the company assembled a dedicated CPU research team. Apple acquired a CPU company and designs application processors for iPhones. ARM designs and sells a wide array of scalar in-order cores through superscalar out-of-order cores as soft IP. NVIDIA is developing their own CPUs, as well, under Project Denver.

Other forces are at play, that will further accelerate processor diversification, not just across company lines but also within.

- *ISA challengers and open ISA movements*. The near-monopoly of the x86 ISA in desktops and servers was a force for consolidation. Intel made it all but certain through its passion for microarchitecture innovation and cutting-edge manufacturing. Now, the formidable trio of ISA, microarchitecture, and manufacturing, is being skirted by ISA challengers in newer markets. Currently, the ARM ISA is the dominant challenger. Open ISA movements, such as Berkeley's RISC-V [19], are even more favorable for diversification. RISC-V in particular is truly minimal and consequently amenable to complex microarchitectures, extensible through optional co-processor extensions, cognizant of accelerator trends, and free of litigation. All of these factors are ingredients for diversity.
- *Stratified market*: With mobile computing, cloud computing, social media, and e-commerce, the market for processors is far more diverse today than it was even a decade ago. Anecdotally, an engineer from a large e-commerce company just recently requested access to the FabScalar toolset, citing "we're interested in processors for data centers". Separately, there is speculation about Amazon designing their own ARM-based processors in the future (some of the circumstantial evidence is that Amazon recently hired former Calxeda engineers) [21].
- *Frequency*: ISA factors aside, Intel's technical success can be attributed to deftly balancing instructions per cycle (IPC) and frequency, while mercilessly pursuing frequency outright with highly-optimized physical designs and best-in-class manufacturing. Today, peak frequency is stable between 3 and 4 GHz and, interestingly, frequencies of application processors are diverse. Processor companies are competing on other performance factors (*e.g.*, IPC), power, and functionality. Frequency is still important, but it is more about tightening-up frequency for a given design complexity, not pushing the frequency envelope itself. IPC is once again open for consideration [20]. Instruction-level parallelism (ILP) is being aggressively pursued in application processor design teams. Semi-custom designs made by smaller design teams are competitive. All of these factors are favorable for processor diversity.
- *Dark silicon*: Dark silicon is the prospect of having more cores on a chip than can be reliably powered-on [7][9]. One implication of dark silicon is that simply adding more of the same core type is of little value if the additional cores cannot be powered-on. This situation is fertile for processor research. It opens the door to single-ISA heterogeneous multi-core processors (HMP) [10][11][12], accelerators for general-purpose codes (ASIC blocks [5], programmable/reconfigurable hardware fabrics [1], GPUs, vector/SIMD units, *etc*.), and better conventional cores. An HMP is comprised of multiple functionally equivalent but microarchitecturally diverse core types. Accelerators co-exist with these general-purpose core types. This new processor paradigm presents a rich research agenda that will be on-going for years to come, as microarchitects explore the vast design space of core types and accelerators, and co-design "optimal" ensembles of core types along with algorithms for effectively scheduling program phases to core types [13].

Appendix B: FabScalar user data

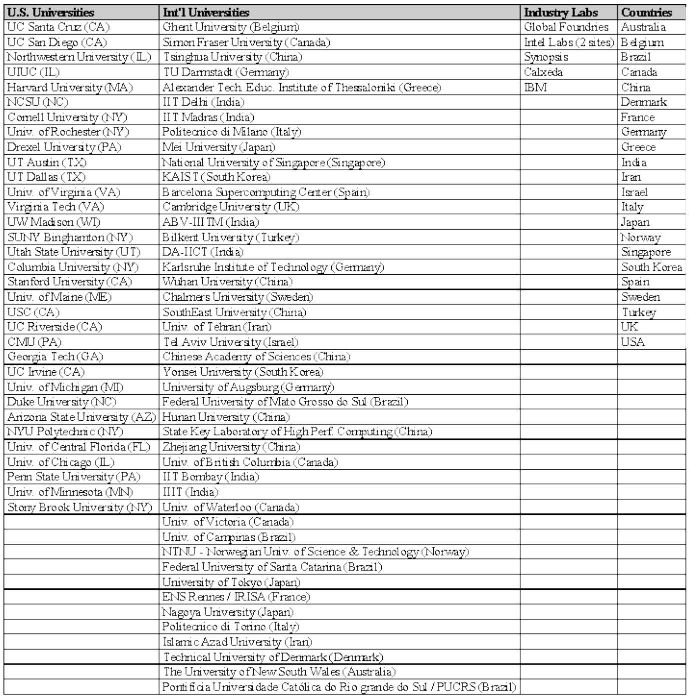
The following user data is up-to-date through November of 2014.

Since its beta release in 2010 and first major publication in 2011, 230 researchers, from 33 U.S. universities, 45 international universities, 6 industry sites, and 22 countries, have downloaded the FabScalar toolset. A complete list of affiliations is shown in Figure 1(a).

Figure 1(b) shows the number of new members added to the FabScalar Google group [17] and site [18] over time. 18 new members in 2010, 41 in 2011, 79 in 2012, 50 in 2013, and 62 in 2014. Two peaks in new memberships can be seen during the same months when the ISCA’11 paper and follow-up IEEE Micro Top Picks paper came out, in June 2011 and May/June 2012, respectively. This may be coincidence, or it may be that member spikes correlate with dissemination. A third spike occurred in February 2014 when a Penn State class used FabScalar for projects.

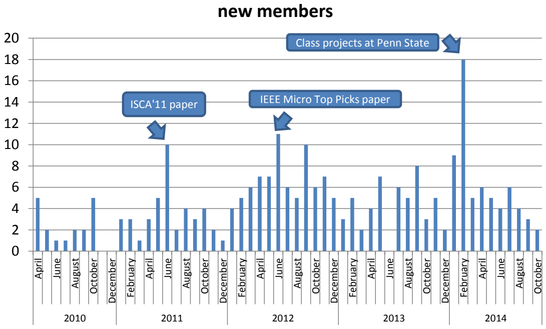
According to Google Scholar, 35 external papers or theses (not affiliated with NCSU) cite FabScalar [22]–[56], and a majority of these seem to use it in their experimental methodology. Two of the papers were nominated for best paper in HPCA 2012 [27][49].

Aside from papers, which are arduous to produce and get published, it is challenging to gauge the amount of activity by FabScalar users. One indirect measure is activity in the Google group Q&A forum, summarized in Figure 1(c). A total of 98 topics (threads) have been created, to which there have been 412 posts (4.2 posts/topic, on average) and 2,983 views (30 views/topic, on average).



U.S. Universities	Int'l Universities	Industry Labs	Countries
UC Santa Cruz (CA)	Ghent University (Belgium)	Global Foundries	Australia
UC San Diego (CA)	Simon Fraser University (Canada)	Intel Labs (2 sites)	Belgium
Northwestern University (IL)	Tsinghua University (China)	Synopsis	Brazil
UIUC (IL)	TU Darmstadt (Germany)	Calxeda	Canada
Harvard University (MA)	Alexander Tech. Educ. Institute of Thessaloniki (Greece)	IBM	China
NCSU (NC)	IIT Delhi (India)		Denmark
Cornell University (NY)	IIT Madras (India)		France
Univ. of Rochester (NY)	Politecnico di Milano (Italy)		Germany
Drexel University (PA)	Mei University (Japan)		Greece
UT Austin (TX)	National University of Singapore (Singapore)		India
UT Dallas (TX)	KAIST (South Korea)		Iran
Univ. of Virginia (VA)	Barcelona Supercomputing Center (Spain)		Israel
Virginia Tech (VA)	Cambridge University (UK)		Italy
UW Madison (WI)	ABV-IIITM (India)		Japan
SUNY Binghamton (NY)	Bilkent University (Turkey)		Norway
Utah State University (UT)	DA-IICT (India)		Singapore
Columbia University (NY)	Karlsruhe Institute of Technology (Germany)		South Korea
Stanford University (CA)	Wuhan University (China)		Spain
Univ. of Maine (ME)	Chalmers University (Sweden)		Sweden
USC (CA)	SouthEast University (China)		Turkey
UC Riverside (CA)	Univ. of Tehran (Iran)		UK
CMU (PA)	Tel Aviv University (Israel)		USA
Georgia Tech (GA)	Chinese Academy of Sciences (China)		
UC Irvine (CA)	Yonsei University (South Korea)		
Univ. of Michigan (MI)	University of Augsburg (Germany)		
Duke University (NC)	Federal University of Mato Grosso do Sul (Brazil)		
Arizona State University (AZ)	Hunan University (China)		
NYU Polytechnic (NY)	State Key Laboratory of High Perf. Computing (China)		
Univ. of Central Florida (FL)	Zhejiang University (China)		
Univ. of Chicago (IL)	Univ. of British Columbia (Canada)		
Penn State University (PA)	IIT Bombay (India)		
Univ. of Minnesota (MN)	IIIT (India)		
Stony Brook University (NY)	Univ. of Waterloo (Canada)		
	Univ. of Victoria (Canada)		
	Univ. of Campinas (Brazil)		
	NTNU - Norwegian Univ. of Science & Technology (Norway)		
	Federal University of Santa Catarina (Brazil)		
	University of Tokyo (Japan)		
	ENS Rennes / IRISA (France)		
	Nagoya University (Japan)		
	Politecnico di Torino (Italy)		
	Islamic Azad University (Iran)		
	Technical University of Denmark (Denmark)		
	The University of New South Wales (Australia)		
	Pontificia Universidade Católica do Rio grande do Sul / PUCRS (Brazil)		

(a) Affiliations.



(b) New members over time.

# topics	98
# posts to topics	412
average posts/topic	4.2
# views of topics	2,983
average views/topic	30

(c) Google group activity.

Figure 1. FabScalar usage data.

Appendix C: References

General references:

[1] J. Benson, R. Cofell, C. Frericks, C.-H. Ho, V. Govindaraju, T. Nowatzki, and K. Sankaralingam. Design, integration and implementation of the DySER hardware accelerator into OpenSPARC. *18th International Symposium on High Performance Computer Architecture*, pp. 1-12, Feb. 2012.

[2] D. Burger, T. M. Austin, S. Bennett. Evaluating Future Microprocessors: The SimpleScalar ToolSet. University of Wisconsin-Madison Technical Report CS-TR-1308, 1996.

[3] N. K. Choudhary, S. V. Wadhavkar, T. A. Shah, H. Mayukh, J. Gandhi, B. H. Dwiel, S. Navada, H. H. Najaf-abadi, and E. Rotenberg. FabScalar: Composing Synthesizable RTL Designs of Arbitrary Cores within a Canonical Superscalar Template. *38th IEEE/ACM International Symposium on Computer Architecture*, pp. 11-22, June 2011.

[4] N. K. Choudhary, S. V. Wadhavkar, T. A. Shah, H. Mayukh, J. Gandhi, B. H. Dwiel, S. Navada, H. H. Najaf-abadi, and E. Rotenberg. FabScalar: Automating Superscalar Core Design. *IEEE Micro, Special Issue: Micro's Top Picks from the Computer Architecture Conferences*, 32(3):48-59, May-June 2012.

[5] J. Cong, M. A. Ghodrat, M. Gill, B. Grigorian, K. Gururaj, and G. Reinman. Accelerator-Rich Architectures: Opportunities and Progresses. *51st Design Automation Conference*, pp. 1-6, June 2014.

[6] B. H. Dwiel, N. K. Choudhary, and E. Rotenberg. FPGA Modeling of Diverse Superscalar Processors. *2012 IEEE International Symposium on Performance Analysis of Systems and Software*, pp. 188-199, April 2012.

[7] H. Esmaeilzadeh, E. Blem, R. Amant, K. Sankaralingam and D. Burger. Dark Silicon and the End of Multicore Scaling, *38th IEEE/ACM International Symposium on Computer Architecture*, June 2011.

[8] E. Forbes, R. Basu Roy Chowdhury, B. Dwiel, A. Kannepalli, V. Srinivasan, Z. Zhang, R. Widialaksono, T. Belanger, S. Lipa, E. Rotenberg, W. R. Davis, and P. D. Franzon. Experiences with Two FabScalar-based Chips. *6th Workshop on Architectural Research Prototyping (WARP-6)*, June 14, 2015.

[9] N. Goulding, J. Sampson, G. Venkatesh, S. Garcia, V. Bryskin, J. Martinez, S. Swanson and M. Taylor. GreenDroid: A Mobile Application Processor for a Future of Dark Silicon, *HotChips*, 2010.

[10] 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. *International Symposium on Microarchitecture*, Dec. 2003.

[11] R. Kumar, D. M. Tullsen, P. Ranganathan, N. P. Jouppi, K. I. Farkas. Single-ISA Heterogeneous Multi-Core Architectures for Multithreaded Workload Performance. *31st International Symposium on Computer Architecture*, June 2004.

[12] R. Kumar, D. M. Tullsen, and N. P. Jouppi. Core Architecture Optimization for Heterogeneous Chip Multiprocessors. *15th International Symposium on Parallel Architecture and Compilation Techniques*, Sep. 2006.

[13] S. Navada, N. K. Choudhary, S. V. Wadhavkar, and E. Rotenberg. A Unified View of Non-monotonic Core Selection and Application Steering in Heterogeneous Chip Multiprocessors. *22nd IEEE/ACM International Conference on Parallel Architectures and Compilation Techniques*, pp. 133-144, September 2013.

[14] S. Sabharwal. Microarchitectural Implementation of the MIPS System Coprocessor in FabScalar-generated Superscalar Cores. M.S. Thesis, Department of Electrical and Computer Engineering, North Carolina State University, August 2013.

[15] T. A. Shah. FabMem: A Multiported RAM and CAM Compiler for Superscalar Design Space Exploration. M.S. Thesis, Department of Electrical and Computer Engineering, North Carolina State University, May 2010.

[16] A. V. Shastri. Microarchitectural Implementation of the MIPS Floating-point ISA in FabScalar-generated Superscalar Cores. M.S. Thesis, Department of Electrical and Computer Engineering, North Carolina State University, August 2012.

[17] https://groups.google.com/forum/?fromgroups#!forum/fabscalar-users
(sign up at http://people.engr.ncsu.edu/ericro/research/fabscalar/pre-release.htm)

[18] https://sites.google.com/a/ncsu.edu/fabscalar/

[19] K. Asanovic and D. Patterson. RISC-V: An Open Standard for SoCs (The case for an open ISA). *EE Times*, Aug. 2014. http://riscv.org/

[20] L. Gwennap. Soft Machines Targets IPC Bottleneck. *Microprocessor Report*, October 2014. http://www.softmachines.com/

[21] http://www.enterprisetech.com/2014/05/01/amazon-going-make-arm-processors/

The following non-NCSU references cite FabScalar (also see Appendix B – user data):

[22] Allred, J. M., Roy, S., & Chakraborty, K. (2014). Dark silicon aware multicore systems: Employing design automation with architectural insight. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 22(5):1192-1196.

[23] Allred, J. M., Roy, S., & Chakraborty, K. (2013, October). Long term sustainability of differentially reliable systems in the dark silicon era. In Computer Design (ICCD), 2013 IEEE 31st International Conference on (pp. 70-77). IEEE.

[24] Allred, J. M. (2013). Designing, optimizing, and sustaining heterogeneous chip multiprocessors to systematically exploit dark silicon (Doctoral dissertation, UTAH STATE UNIVERSITY).

[25] Ancajas, D. M., Nickerson, J. M., Chakraborty, K., & Roy, S. (2013, May). HCI-Tolerant NoC router microarchitecture. In Design Automation Conference (DAC), 2013 50th ACM/EDAC/IEEE (pp. 1-10). IEEE.

[26] Balasubramanian, R., & Sankaralingam, K. Understanding the Impact of Gate-level Physical Reliability Effects on Whole Program Execution. In High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on (pp. 60-71). IEEE.

[27] Battle, S., Hilton, A. D., Hempstead, M., & Roth, A. (2012, February). Flexible register management using reference counting. In High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on (pp. 1-12). IEEE.

[28] Battle, S. J., & Hempstead, M. (2013, October). Register allocation and VDD-gating algorithms for out-of-order architectures. In Computer Design (ICCD), 2013 IEEE 31st International Conference on (pp. 108-114). IEEE.

[29] Benson, J., Cofell, R., Frericks, C., Ho, C. H., Govindaraju, V., Nowatzki, T., & Sankaralingam, K. (2012, February). Design, integration and implementation of the DySER hardware accelerator into OpenSPARC. In High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on (pp. 1-12). IEEE.

[30] Bojnordi, M. N., & Ipek, E. (2012, June). PARDIS: A programmable memory controller for the DDRx interfacing standards. In Computer Architecture (ISCA), 2012 39th Annual International Symposium on (pp. 13-24). IEEE.

[31] Bojnordi, M. N., & Ipek, E. (2013). A programmable memory controller for the DDRx interfacing standards. ACM Transactions on Computer Systems (TOCS), 31(4), 11.

[32] Chakraborty, K., Cozzens, B., Roy, S., & Ancajas, D. M. (2013, May). Efficiently tolerating timing violations in pipelined microprocessors. In Proceedings of the 50th Annual Design Automation Conference (p. 102). ACM.

[33] Chen, H., Roy, S., & Chakraborty, K. (2014, March). Exploiting static and dynamic locality of timing errors in robust L1 cache design. In Quality Electronic Design (ISQED), 2014 15th International Symposium on (pp. 9-15). IEEE.

[34] Chen, H., Roy, S., & Chakraborty, K. (2014, March). DARP: dynamically adaptable resilient pipeline design in microprocessors. In Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014 (pp. 1-6). IEEE.

[35] Corbetta, S., & Fornaciari, W. (2013, May). Performance/reliability trade-off in superscalar processors for aggressive NBTI restoration of functional units. In Proceedings of the 23rd ACM international conference on Great lakes symposium on VLSI (pp. 221-226). ACM.

[36] Dadashi, M., Rashid, L., Pattabiraman, K., & Gopalakrishnan, S. Hardware-Software Integrated Diagnosis for Intermittent Hardware Faults. Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on (pp. 363-374). IEEE.

[37] Fang, K., Ni, Y., He, J., Li, Z., Mu, S., & Deng, Y. (2013, October). FastLanes: An FPGA accelerated GPU microarchitecture simulator. In Computer Design (ICCD), 2013 IEEE 31st International Conference on (pp. 241-248). IEEE.

[38] Guevara, M., Lubin, B., & Lee, B. C. (2014). Market mechanisms for managing datacenters with heterogeneous microarchitectures. ACM Transactions on Computer Systems (TOCS), 32(1), 3.

[39] Guevara, M., Lubin, B., & Lee, B. C. (2014, February). Strategies for anticipating risk in heterogeneous system design. In Proc. 20th IEEE International Symposium on High-Performance Computer Architecture (HPCA).

[40] M. Hayenga, D. Johnson, and M. Lipasti. Pitfalls of ORION-Based Simulation. *10th Annual Workshop on Duplicating, Deconstructing, and Debunking (WDDD-10)*, Portland, Oregon June 2012.

[41] Luo, Y., Hsu, W. C., & Zhai, A. (2013). The design and implementation of heterogeneous multicore systems for energy-efficient speculative thread execution. ACM Transactions on Architecture and Code Optimization (TACO), 10(4), 26.

[42] K. Mysur, M. Pricopi, T. Marconi, and T. Mitra. Implementation of Core Coalition on FPGAs. IFIP/IEEE International Conference on Very Large Scale Integration, October 2013.

[43] Nakabayashi, T. (2014). Researches on fabrication of low-energy heterogeneous multi-core processors (Doctoral dissertation, 三重大学).

[44] Nakabayashi, T., Sasaki, T., & Kondo, T. (2013, December). Dynamic BTB Resizing for Variable Stages Superscalar Architecture. In Computing and Networking (CANDAR), 2013 First International Symposium on (pp. 352-358). IEEE.

[45] Naresh, V. R. K., Palframan, D. J., & Lipasti, M. H. (2011, December). CRAM: coded registers for amplified multiporting. In Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture (pp. 196-205). ACM.

[46] Nathan, R., & Sorin, D. J. (2014, March). Nostradamus: Low-cost hardware-only error detection for processor cores. In Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014 (pp. 1-6). IEEE.

[47] Okamoto, T., Nakabayashi, T., Sasaki, T., & Kondo, T. (2013, December). FabCache: Cache Design Automation for Heterogeneous Multi-core Processors. In Computing and Networking (CANDAR), 2013 First International Symposium on (pp. 602-606). IEEE.

[48] Pricopi, M., & Mitra, T. Bahurupi: A polymorphic heterogeneous multi-core architecture. *ACM Transactions on Architecture and Code Optimization (TACO)*, 8(4), 22, 2012.

[49] Sartori, J., Ahrens, B., & Kumar, R. (2012, February). Power balanced pipelines. In High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on (pp. 1-12). IEEE.

[50] Sartori, J., & Kumar, R. (2013). Exploiting Timing Error Resilience in Processor Architecture. ACM Transactions on Embedded Computing Systems (TECS), 12(2s), 89.

[51] Sartori, J., & Kumar, R. (2014, August). Software canaries: software-based path delay fault testing for variation-aware energy-efficient design. In Proceedings of the 2014 international symposium on Low power electronics and design (pp. 159-164). ACM.

[52] Sartori, J. (2012). Programmable stochastic processors (Doctoral dissertation, University of Illinois at Urbana-Champaign).

[53] Swaminathan, K., Liu, H., Li, X., Kim, M. S., Sampson, J., & Narayanan, V. (2014, June). Steep Slope Devices: Enabling New Architectural Paradigms. In Proceedings of the 51st Annual Design Automation Conference on Design Automation Conference (pp. 1-6). ACM.

[54] Waksman, A., Suozzo, M., & Sethumadhavan, S. (2013, November). FANCI: identification of stealthy malicious logic using boolean functional analysis. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (pp. 697-708). ACM.

[55] Waksman, A. (2014). Producing Trustworthy Hardware Using Untrusted Components, Personnel and Resources (Doctoral dissertation, COLUMBIA UNIVERSITY).

[56] Zhang, W., Zhang, H., & Lach, J. (2014, August). Adaptive front-end throttling for superscalar processors. In Proceedings of the 2014 international symposium on Low power electronics and design (pp. 21-26). ACM.