A RISC-V Adaptive Processor with a Vector of Reconfigurable Energy-Efficient Accelerators

Minwoo (Josh) Kang
Advisor: Duane A. Bailey

Department of Computer Science, Williams College
October 24, 2019

Moore's Law is dead and Dennardian scaling is over. Today, only a fraction of a chip can be driven at maximum clock speed due to the power utilization wall [1] imposed on modern processors. Increasing the die density is no longer effective, since it only implies that a greater percentage of the die must be kept idle to meet the power budget. Therefore, in this era of *dark silicon*, processors must be designed to optimize energy consumption and silicon utilization [2, 3]. To this end, recent literature has highlighted a specific approach that involves accelerators—specialized hardware units that can perform certain tasks with higher performance and/or greater energy-efficiency. A number of groups have developed heterogeneous System-on-a-Chip (SoC) architectures that couple accelerators with general-purpose cores and have reported notable gains in speed-up and energy savings [4, 5, 6]. However, many of these designs incorporate a fixed set of accelerators that allow enhanced performance for only a few limited applications. On the other hand, real-world users demand processors to perform optimally for a variety of user-specific workloads—we therefore believe there is considerable potential for the *personalization of hardware*.

This research will focus on implementing an adaptive processor that will enable personalization by making self-aware decisions about how accelerators are used. Ideally, a processor would like to have access to as many accelerators as possible for maximal energy-efficiency, but realistically there is a constraint on on-chip area and power. Instead, our proposed design will at run-time pick a set of accelerators to be activated on the chip based on readings from sensors monitoring power usage and accelerator utilization rates. At the next hardware compile-time, the processor will further decide which accelerators should be kept and which should be evicted to provide space for more useful units. In a sense, we are optimizing the area associated with accelerators and dynamically tailoring our vector of accelerators to the running application to improve overall energy-efficiency.

Building such an architecture first requires a general-purpose core. This work uses the *Rocket core*, which is a RISC-V 64-bit in-order processor developed at UC Berkeley and released as open-source [7]. The Rocket core is particularly helpful because it is part of a larger ecosystem that includes, among other things, a parameterizable hardware design language [8], a cloud-based cycle-accurate hardware simulator [9], and even some accelerators, such as a vector processor [10], a memory copy accelerator [11], and a neural-net accelerator [12]. At Williams, we are designing a broader palette of accelerators that our adaptive processor may potentially use. Development and implementation of accelerators will be a collective, open-source effort among students in the Bailey research group.

Given a collection of accelerators, this work will focus on the hardware infrastructure that supports self-aware decisions. This system comes in three parts. (1) We will build an accelerator invocation protocol for making decisions about whether to execute a function in hardware or in software. Previous research, such as that by Venkatesh et al., propose a *fall-back* system in which the processor first attempts to execute the function in hardware, and if the hardware is unavailable, falls back to executing in software [1, 4, 13]. We are considering a similar scheme for our architecture that possibly uses trap mechanisms to detect functions that can be accelerated. Hardware availability will then be checked by referencing a *capability vector* that records the current list of activated accelerators, and a *performance vector* will simultaneously record the accelerator use statistics. (2) Our system requires a collection of sensors to constantly monitor power usage and accelerator hit/miss rates. Recent reports on heterogeneous SoC designs have demonstrated the use of integrated Power Management Units (PMUs), and we expect to build a similar yet simplified unit that focuses on gauging the on-chip power demands [14, 15]. (3) We will devise a mechanism that will intelligently guide the processor personalization that depends on sensor readings.

We believe autonomous *personalization of hardware* will allow users to maximize the efficiencies of dark silicon.

References

[1] G. Venkatesh, J. Sampson, N. Goulding, S. Garcia, V. Bryksin, J. Lugo-Martinez, S. Swanson, and M. B. Taylor, "Conservation cores: Reducing the energy of mature computations," *SIGPLAN Not.*, vol. 45, pp. 205–218, Mar. 2010.

[2] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *2011 38th Annual International Symposium on Computer Architecture (ISCA)*, pp. 365–376, June 2011.

[3] M. B. Taylor, "Is dark silicon useful? Harnessing the four horsemen of the coming dark silicon apocalypse," in *DAC Design Automation Conference 2012*, pp. 1131–1136, June 2012.

[4] S. Swanson and M. B. Taylor, "Greendroid: Exploring the next evolution in smartphone application processors," *IEEE Communications Magazine*, vol. 49, pp. 112–119, April 2011.

[5] V. Govindaraju, C.-H. Ho, T. Nowatzki, J. Chhugani, N. Satish, K. Sankaralingam, and C. Kim, "DySER: unifying functionality and parallelism specialization for energy efficient computing," *IEEE Micro*, vol. 33, no. 5, 2012.

[6] M. Lin, S. Cheng, R. F. DeMara, and J. Wawrzynek, "ASTRO: Synthesizing application-specific reconfigurable hardware traces to exploit memory-level parallelism," *Microprocessors and Microsystems*, vol. 39, 03 2015.

[7] K. Asanović, R. Avizienis, J. Bachrach, S. Beamer, D. Biancolin, C. Celio, H. Cook, D. Dabbelt, J. Hauser, A. Izraelevitz, S. Karandikar, B. Keller, D. Kim, J. Koenig, Y. Lee, E. Love, M. Maas, A. Magyar, H. Mao, M. Moreto, A. Ou, D. A. Patterson, B. Richards, C. Schmidt, S. Twigg, H. Vo, and A. Waterman, "The Rocket Chip Generator," Tech. Rep. UCB/EECS-2016-17, EECS Department, University of California, Berkeley, Apr 2016.

[8] J. Bachrach, H. Vo, B. Richards, Y. Lee, A. Waterman, R. Avižienis, J. Wawrzynek, and K. Asanović, "Chisel: Constructing hardware in a Scala embedded language," in *DAC Design Automation Conference 2012*, pp. 1212–1221, June 2012.

[9] S. Karandikar, H. Mao, D. Kim, D. Biancolin, A. Amid, D. Lee, N. Pemberton, E. Amaro, C. Schmidt, A. Chopra, Q. Huang, K. Kovacs, B. Nikolic, R. Katz, J. Bachrach, and K. Asanović, "FireSim: FPGA-accelerated cycle-exact scale-out system simulation in the public cloud," in *Proceedings of the 45th Annual International Symposium on Computer Architecture*, ISCA '18, (Piscataway, NJ, USA), pp. 29–42, IEEE Press, 2018.

[10] Y. Lee, A. Waterman, H. Cook, B. Zimmer, B. Keller, A. Puggelli, J. Kwak, R. Jevtic, S. Bailey, M. Blagojevic, P. Chiu, R. Avizienis, B. Richards, J. Bachrach, D. Patterson, E. Alon, B. Nikolic, and K. Asanović, "An agile approach to building RISC-V microprocessors," *IEEE Micro*, vol. 36, pp. 8–20, Mar 2016.

[11] H. Mao, "Hardware acceleration for memory to memory copies," Master's thesis, EECS Department, University of California, Berkeley, Jan 2017.

[12] S. Eldridge, A. Waterland, M. Seltzer, J. Appavoo, and A. Joshi, "Towards general-purpose neural network computing," in *2015 International Conference on Parallel Architecture and Compilation (PACT)*, pp. 99–112, Oct 2015.

[13] S. Davidson, S. Xie, C. Torng, K. Al-Hawai, A. Rovinski, T. Ajayi, L. Vega, C. Zhao, R. Zhao, S. Dai, A. Amarnath, B. Veluri, P. Gao, A. Rao, G. Liu, R. K. Gupta, Z. Zhang, R. Dreslinski, C. Batten, and M. B. Taylor, "The Celerity open-source 511-core RISC-V tiered accelerator fabric: Fast architectures and design methodologies for fast chips," *IEEE Micro*, vol. 38, pp. 30–41, Mar 2018.

[14] S. Eldridge, T. J. Watson, V. Verma, R. S. Joshi, and P. Bose, "A low voltage RISC-V heterogeneous system boosted SRAMs, machine learning, and fault injection on VELOUR," in *First Workshop on Computer Architecture Research with RISC-V (CARRV 2017) at the 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-50)*, Oct 2017.

[15] B. Keller, M. Cochet, B. Zimmer, J. Kwak, A. Puggelli, Y. Lee, M. Blagojević, S. Bailey, P. Chiu, P. Dabbelt, C. Schmidt, E. Alon, K. Asanović, and B. Nikolić, "A RISC-V processor SoC with integrated power management at submicrosecond timescales in 28 nm FD-SOI," *IEEE Journal of Solid-State Circuits*, vol. 52, pp. 1863–1875, July 2017.