OpenPiton+Ariane: The First SMP Linux-booting RISC-V System Scaling From One to Many Cores

Jonathan Balkind*
Katie Lim
Fei Gao
Jinzheng Tu
David Wentzlaff
jbalkind@princeton.edu
katielim@cs.washington.edu
feig@princeton.edu
jinzheng@princeton.edu
wentzlaf@princeton.edu
Princeton Parallel Group, Princeton University
Princeton, New Jersey

Michael Schaffner*
Florian Zaruba
Luca Benini
schaffner@iis.ee.ethz.ch
zarubaf@iis.ee.ethz.ch
benini@iis.ee.ethz.ch
Integrated Systems Laboratory, ETH Zurich
Zurich, Switzerland

ABSTRACT

This paper introduces OpenPiton+Ariane, a permissively-licensed open-source framework designed to enable scalable architecture research prototypes. With the recent addition of SMP Linux running on FPGA, OpenPiton+Ariane is the first SMP Linux-booting, open-source RISC-V system that scales from single-core to manycore. OpenPiton+Ariane inherits capabilities from both the Ariane and OpenPiton projects, bringing with it simulation and FPGA emulation infrastructure, as well as synthesis and back-end scripts for ASIC development. The P-Mesh cache system from OpenPiton was enhanced with support for RISC-V atomic operations but remains otherwise unmodified, thus providing a mature, well-validated manycore memory system. Likewise, Ariane's cache subsystem was adapted to connect to P-Mesh but the core remains otherwise unmodified, which made Linux bring-up straightforward. This paper gives an overview of the system architecture and the modifications that were necessary to join the two open-source projects. Further, we describe the supported simulation and emulation flows and provide FPGA synthesis results for several different system configurations.

CCS CONCEPTS

• **Computer systems organization** → **Reduced instruction set computing**; **Multicore architectures**; • **Hardware** → *Reconfigurable logic and FPGAs*; *Very large scale integration design.*

KEYWORDS

RISC Processors, Computer Architecture, Multicore Architecture, Coherency, Cache

1 INTRODUCTION

Less than one year ago, Ariane was a single-core processor and OpenPiton relied purely on the OpenSPARC T1 core. Through our teams' mutual interest in building an open-source RISC-V manycore platform, we decided to integrate Ariane and OpenPiton, and we made rapid progress in doing so. Thanks to the thorough validation of the independent systems, it took less than three months to build the combined manycore which could run bare metal tests in simulation without noticeable validation issues. A further two months of work brought us to booting Linux on 4 cores on FPGA. In fact, it took only one day from the first successful Linux boot on a single-core to doing the same with a dual-core system.

The combined OpenPiton+Ariane platform [17], is a permissively-licensed open-source framework designed to enable scalable architecture research prototypes. With the recent addition in release 11 of SMP Linux running on FPGA, OpenPiton+Ariane is the world's first SMP Linux-booting, open-source, RISC-V system that scales from single-core to manycore. This makes OpenPiton+Ariane the ideal RISC-V hardware research platform.

OpenPiton began as the world's first open source, general-purpose, multithreaded manycore processor and framework [4]. It leverages the industry-hardened OpenSPARC T1 core [15, 16] with modifications and builds upon it with a scratch-built, scalable uncore (known as P-Mesh) creating a flexible, modern manycore design. OpenPiton provides a complete verification infrastructure of over 8000 tests, is supported by mature software tools, runs full-stack multiuser Debian Linux, and is written in industry standard Verilog. In addition, OpenPiton provides synthesis and back-end scripts for ASIC and FPGA to enable other researchers to bring their designs to implementation. Multiple implementations of OpenPiton have been created including a taped-out 25-core implementation in IBM's 32 nm process and multiple Xilinx FPGA prototypes [13, 14].

It is on this mature foundation that we built OpenPiton+Ariane. Ariane [22] is a 64-bit RISC-V application processor, which implements the RV64GC ISA. Ariane has been taped-out in multiple technologies including GlobalFoundries' 22 nm FDSOI process, and is capable of booting Linux single-core. By modifying the L1 cache for Ariane to support the P-Mesh cache-coherence protocol, we built OpenPiton+Ariane into an SMP Linux-booting, RISC-V manycore. OpenPiton+Ariane inherits all of the capabilities of OpenPiton and of Ariane, bringing them together in a single scalable, configurable, and easy-to-use platform ideal for rapid prototyping of ideas.

*Both authors contributed equally to this work.

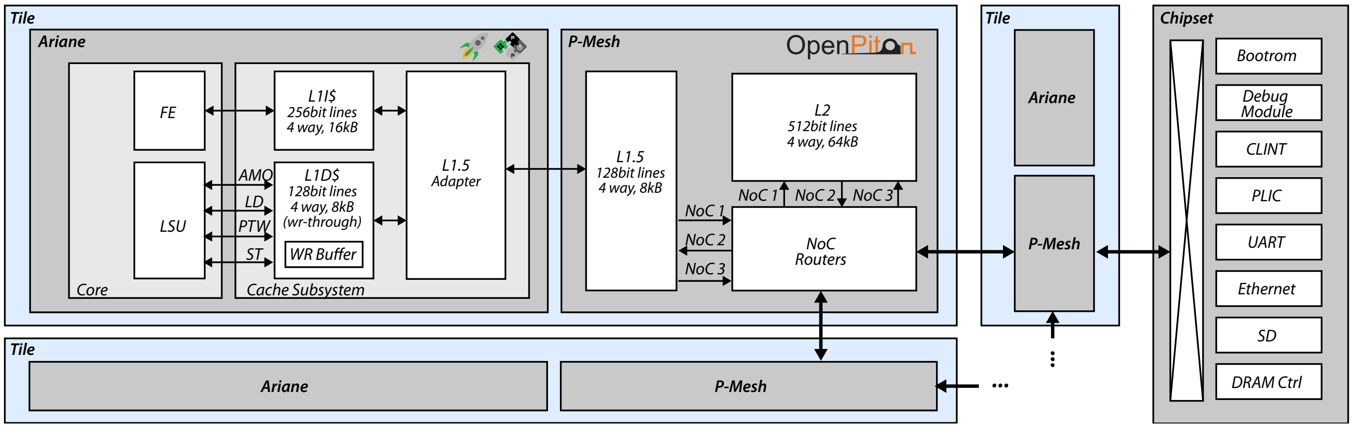

Figure 1: Overview of the OpenPiton+Ariane architecture. Each tile contains one Ariane core, private L1 data and instruction caches, a private L1.5 cache, 3 NoC Routers and a shared L2 cache slice. The chipset contains important platform peripherals such as the DDR memory controller, UART, and the RISC-V-specific peripherals.

Table 1: Supported OpenPiton+Ariane configuration options. Bold indicates defaults. Reproduced [4] and updated.

Component	Configurability Options
Cores (per chip)	Up to 65,536
Cores (per system)	Up to 500 million
Floating-Point Unit	**Present**/Absent
TLBs	Number of entries (**16**)
L1 I-Cache	Number of sets, ways (**16kB, 4 ways**)
L1 D-Cache	Number of sets, ways (**8kB, 4 ways**)
L1.5 Cache	Number of sets, ways (**8kB, 4-way**)
L2 Cache (per tile)	Number of sets, ways (**64kB, 4-way**)
Intra-chip Topologies	**2D Mesh**, Crossbar
Bootloading	SD/SDHC Card, UART, JTAG

This paper gives an overview of the architecture and modifications of the Ariane core and P-Mesh cache subsystem in Section 2, followed by an overview of the simulation and emulation environments in Section 3. Current limitations and future improvements are outlined in Section 4.

2 ARCHITECTURE

The OpenPiton processor system is a flexible, tiled architecture that supports different network-on-chip (NoC) topologies to interconnect a configurable number of processor tiles. The default configuration leverages a 2D mesh topology as shown in Figure 1, and the OpenPiton+Ariane release enables the instantiation of an Ariane RISC-V core within the tiles. Each tile further contains a private L1.5 cache, the NoC routers and a shared L2 cache slice. The chipset contains important platform peripherals such as the DDR memory controller, UART, and RISC-V specific peripherals. The full configuration space is summarised in Table 1, and the above described components are explained in more detail below.

2.1 RISC-V Core

Ariane is a 64 bit, single-issue, in-order RISC-V core (RV64GC) and its block-diagram is shown in Figure 2. It has support for hardware multiply/divide, atomic memory operations as well as an IEEE compliant Floating Point Unit (FPU). Moreover, it has support for the compressed instruction set extension as well as the full privileged instruction set extension. It implements the 39 bit, page-based virtual memory scheme SV39 and boots Linux single-core on FPGA.

To keep Instruction per Cycle (IPC)-losses moderate due to its six stage pipelined design it has a complete branch-prediction infrastructure. The instruction front-end which includes PC generation and instruction fetch from the private instruction cache is decoupled from the processor's back-end which consists of the instruction decode, issue, execute and commit stages. The issue stage tracks operand dependencies in a scoreboard and issues decoded and ready instructions in program order to the execute stage. All execution units are ready-valid hand-shaked and support retiring instructions out-of-order into a lightweight Re-order Buffer (ROB) which commits instructions in issue order to enable precise exception trapping.

The core's Load Store Unit (LSU) manages all integer and floating-point loads and stores as well as address translation and atomic memory operations. It has a split Translation Lookaside Buffer (TLB) for the instruction fetch and the data port. A hardware Page Table Walker (PTW) transparently manages TLB refills on TLB misses. Loads, stores and PTW requests are served on three different ports on a private, write-through data cache which is described in more detail in Section 2.2. Atomic memory operations bypass the regular load/store path and are handled on a separate interface on the data cache. When the core requests an atomic memory operation, the read copy is invalidated and the L1 data cache (re-)requests the data from the memory system together with the corresponding memory operation. The standard configuration of low-level core parameters that is used in OpenPiton is shown in Table 2.

2.2 Cache Subsystem

The Transaction-Response Interface (TRI) is the interface between the L1 caches in the core and the L1.5 cache of OpenPiton's P-Mesh cache subsystem. TRI is a generic and simple interface for cores adhering to a write-through cache protocol, and has been developed as part of JuxtaPiton [10, 11], a previous evolution of OpenPiton where T1 cores could be replaced with PicoRV32 cores to build the world's first open-source, heterogeneous-ISA processor. Since the

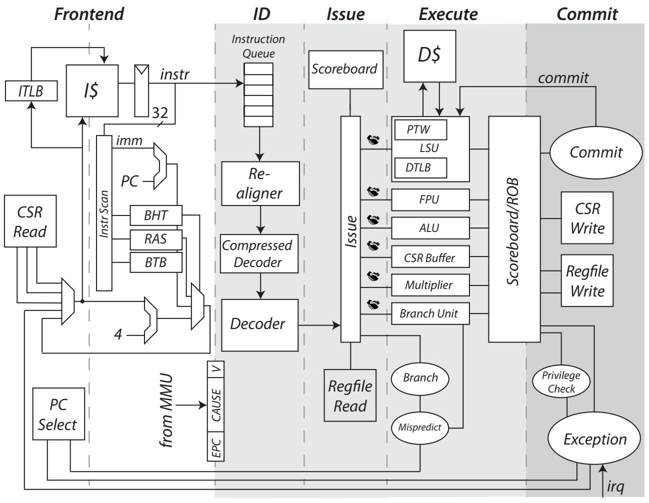

Figure 2: Ariane Core Architecture

original L1 cache of Ariane adhered to a write-back protocol and did not have support for invalidation messages and atomics, a new parametric write-through L1 cache system has been designed that is compatible to the P-Mesh TRI[1]. The L1 data cache is equipped with a merging write-buffer with forwarding capability to ensure good write performance. This write-buffer is currently parameterised to be eight 64 bit words deep and support two outstanding write transactions towards the L1.5. The number of outstanding transactions is currently aligned with the number of hardware threads that the T1 processor uses, since the T1 can only have one pending store per hardware thread.

Beyond the L1 cache, OpenPiton+Ariane uses the same P-Mesh cache subsystem provided in OpenPiton, with few modifications. OpenPiton provides a local private cache per tile (known as the L1.5, but equivalent to other systems' L2 caches) and a slice of the distributed, shared last-level cache (known as the L2) and directory per tile. The L1.5 and L2 are both inclusive and can be parameterised completely independently of each other in terms of size and associativity. The P-Mesh coherence protocol is a four-hop protocol where L1.5 caches only communicate with L2 caches and vice versa. P-Mesh provides three physical NoCs which carry equivalence classes of traffic designed to eliminate deadlock. The NoCs can be replaced with other network topologies (as shown in Table 1) provided that point-to-point ordering is maintained.

2.3 Support for RISC-V Atomic Operations

P-Mesh originally only supported SPARC atomic operations (swap and compare-and-swap) and hence we modified the cache subsystem to support the standard atomic instructions for RISC-V. First, the L1 data cache drains the write-buffer and pending loads in the MSHR, and issues the atomic operation sequentially consistently to P-Mesh (as though all atomics have both `aq` and `rl` bits set). Upon receiving a fetch-and-op atomic operation, the L1.5 cache first invalidates the corresponding cache-line in both the L1.5 and L1 and then forwards the operation to the L2. The L2 further invalidates all

[1]The new L1 cache system also contains an option to switch the interface to an AMBA AXI protocol adapter that has support for atomics.

Table 2: Ariane Standard Configuration

Parameter	Standard	Parameter	Standard
BTB Entries	64	SAQ Depth	4
BHT Entries	128	LAQ Depth	1
RAS Depth	2	Store Latency	0
ROB Size	8	Load Latency	1
Int. Regfile WR Ports	2	–	–

other sharers, if any. The read-modify-write process is finished in the L2 data array with a small atomic ALU we added into the path used by the existing swap operation. Finally, the old value which was read from the L2 is returned to the core.

Differently from those fetch-and-op atomic instructions, load-reserved/store-conditional (LR/SC) is handled within the L1.5. After receiving an LR, the L1.5 requests an upgrade for the line to the "M" MESI state and sets the LR/SC flag to high. From then, any operation that changes the line's MESI state will clear the LR/SC flag (e.g. a load from another core which downgrades the MESI state to "S"). The later SC returns 0 (meaning the store succeeded) only when the LR/SC flag is still high, otherwise the store fails and 1 is returned.

2.4 Platform Peripherals

The OpenPiton framework supports the following essential platform peripherals:

- **UART:** We use the Xilinx AXI UART16550 IP core, which is connected using a P-Mesh to AXI-Lite bridge. It is used both for standard serial I/O for interaction with the user and for bootloading using `pitonstream` (as described in section 3.1).
- **SD/SDHC:** The FPGA emulation can boot from our provided SD/SDHC controller, which comes from OpenPiton. The controller is based on an open-source Wishbone controller [6] (connected to P-Mesh) and includes a hardware driver to initialise the device. The full SD card is then mapped into the memory space for straightforward access for OS bootloading.
- **DRAM:** We make use of the FPGA-specific DDR3 and DDR4 controllers provided by Xilinx, wrapped in a P-Mesh NoC to Xilinx interface bridge.
- **Ethernet:** As for OpenPiton, we use the Xilinx AXI Ethernet Lite 10/100 MAC in OpenPiton+Ariane. Standalone Ariane also supports the lowRISC Ethernet MAC, which is a port of Alex Forencich's GHz RGMII design [7, 12]. We have the option to switch to the lowRISC Ethernet MAC in future to achieve higher bandwidth.

The RISC-V ecosystem specifies a handful of peripherals and additional core infrastructure which we implement and support in our system. This includes:

- **Debug:** The RISC-V draft spec v0.13 [19] compliant debug module governs external, multi-hart, run-control debug. The cores use their existing pipeline to facilitate debug functionality. An external debug request signal redirects the core to a "programmable" debug ROM which injects debug instructions into the core's pipeline.

Table 3: Some of the supported FPGA build configurations. Both cores have the same default cache configuration (see Table 1). The results have been generated with Vivado 2018.2, using OpenPiton r11 / Ariane v4.1 including additional development patches that will be part of upcoming releases.

Board Name / FPGA Type	Clock [MHz]	Config X × Y	Core Type	FPU [y/n]	LUTs [k]	Registers [k]	RAM Tiles [#]	DSPs [#]
Digilent NexysVideo Artix 7 7a200tsbg484	30	1 × 1	Ariane	no	95 (71%)	72 (27%)	66 (18%)	16 (2%)
	30	1 × 1	Ariane	yes	110 (82%)	75 (28%)	66 (18%)	27 (4%)
	30	1 × 1	OpenSPARC T1	yes	115 (86%)	96 (36%)	59 (16%)	13 (2%)
Digilent Genesys2 Kintex 7 7k325tffg900-2	67	1 × 1	Ariane	no	86 (42%)	72 (17%)	66 (15%)	16 (2%)
	67	1 × 1	Ariane	yes	99 (49%)	75 (18%)	66 (15%)	27 (3%)
	67	1 × 1	OpenSPARC T1	yes	105 (52%)	91 (22%)	59 (13%)	16 (2%)
	67	2 × 1	Ariane	no	141 (69%)	113 (28%)	124 (28%)	16 (4%)
	67	2 × 1	Ariane	yes	167 (82%)	120 (30%)	124 (28%)	54 (6%)
	67	2 × 1	OpenSPARC T1†	yes	160 (79%)	137 (33%)	112 (25%)	32 (4%)
Xilinx VC707 Virtex 7 7vx485tffg1761-2	60	1 × 1	Ariane	no	99 (33%)	73 (12%)	63 (6%)	16 (<1%)
	60	1 × 1	Ariane	yes	114 (37%)	77 (13%)	63 (6%)	27 (1%)
	60	1 × 1	OpenSPARC T1	yes	119 (39%)	97 (16%)	53 (5%)	16 (<1%)
	60	2 × 2	Ariane	no	284.1 (94%)	202 (33%)	237 (23%)	64 (2%)
	60	3 × 1	Ariane	yes	268 (88%)	169 (28%)	179 (17%)	81 (3%)
	60	3 × 1	OpenSPARC T1†	yes	255 (84%)	208 (34%)	158 (15%)	48 (2%)
Xilinx VCU118 Virtex US+ xcvu9pflga2104-2L	100	1 × 1	Ariane	no	90 (8%)	81 (3%)	88 (4%)	19 (<1%)
	100	1 × 1	Ariane	yes	103 (9%)	84 (4%)	89 (4%)	30 (<1%)
	100	1 × 1	OpenSPARC T1	yes	108 (9%)	100 (4%)	79 (4%)	19 (<1%)
	100	4 × 4	Ariane	no	923 (78%)	704 (30%)	963 (45%)	259 (4%)
	100	4 × 2	Ariane	yes	583 (49%)	399 (17%)	495 (23%)	219 (3%)

† Without Coherence Domain Restriction [8] in caches.

- **CLINT:** The Core Local Interrupt Controller (CLINT) provides Inter Processor Interrupts (IPI) and a common time-base. Each core has its own timer compare register which triggers an external timer interrupt when it matches the global time-base.
- **PLIC:** The Platform Level Interrupt Controller (PLIC) is a interrupt controller which manages external peripheral interrupts. It provides a context for each privilege level and core. The software can configure different priority thresholds for each context. The PLIC is still subject to official standardisation. However, there is already an implementation including a Linux driver, which is agreed upon.

2.5 Automatic Device Tree Generation

In order to capture the different platform configurations that OpenPiton+Ariane provides, we added an automatic device tree generation script to the PyHP preprocessor from OpenPiton. This script parses an XML description of the system address map and platform peripherals (which is also used to generate the chipset crossbar), and together with the information about the number of cores and the clock frequency it generates a device tree that is compiled into a bootrom attached to the peripheral space. The "zero-stage" bootloader stored in that bootrom initialises the cores and loads a pointer to the device tree blob into register `a1` as per RISC-V convention. With this automatic device tree generation, the same Linux image can be booted on differently parameterised instances, automatically adapting to the platform at runtime.

3 SIMULATION & EMULATION PLATFORMS

Ariane plugs into the `sims` simulation infrastructure provided in OpenPiton. This handles the building of simulation models with each of the supported simulators (at present, Mentor QuestaSim, Synopsys VCS and Verilator), as well as running one test or an entire test suite against the compiled model. We have enhanced `sims` to support compilation of RISC-V assembly and C tests, and the direct use of pre-compiled binaries. The primary bare-metal test suite is the publicly available `riscv-tests` repository [20]. Beyond bare-metal testing, we also simulate Linux boot for debugging, which takes approximately 4 days to boot for a single core (DRAM reduced to 128MB to speed up the memory initialisation phase in simulation).

3.1 FPGA Flows

The Ariane core option has been integrated into the OpenPiton `protosyn` build flow and is available for the Digilent Nexys Video and Genesys2 boards, as well as the Xilinx VC707 and VCU118 development boards. The resource consumption of a set of builds with the standard cache configuration and different numbers of cores is shown in Table 3. Since the Ariane FPU pipeline registers have not been optimised for FPGA mapping, enabling the FPU will result in a somewhat lower core clock frequency. The LUT distribution for single-core Genesys2 builds is shown in Figure 3. The core amounts to around 22%-41% of the total resources, depending on the actual configuration (Ariane with or without FPU, OpenSPARC T1 with FPU). Further, we note that the T1 is around 23% and 93% larger than Ariane with and without FPU, respectively. This area

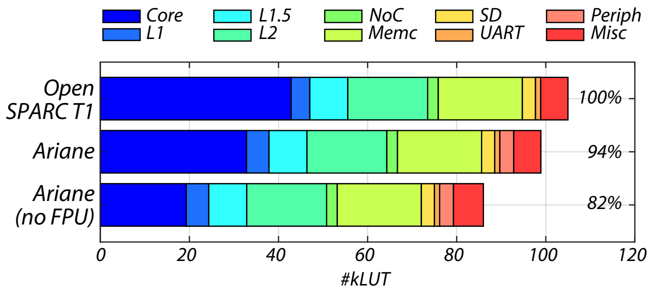

Figure 3: LUT distribution of single-core builds for the Genesys2 board (Kintex 7k325tffg900-2). The percentages are normalized with respect to the OpenSPARC total size.

difference can be attributed in part to the T1's register windows and its reliability features[2].

`pitonstream` is a tool and bootloading option for testing designs on FPGA. The user specifies a set of tests that they want to run, and those are compiled on the host machine and streamed into the memory of the FPGA over the UART link. `pitonstream` will run one test at a time and log its output and success or failure.

Since OpenPiton+Ariane is equipped with a RISC-V compliant Debug Module, the FPGA configurations can be in-system debugged via JTAG using, e.g., OpenOCD with GDB. This can also be used as a bootloading option for Linux as an alternative to UART and SD.

3.2 ASIC Flows

OpenPiton is a rare example of an open-source processor which also provides open-source synthesis and back-end scripts for ASIC development. Alongside an ongoing internal effort to refactor the flow to be more process- and tool-generic, we have begun to run the Ariane tile design through the flow. In particular, we have performed synthesis of the core in GlobalFoundries 14 nm technology and are working through the rest of the flow. Once the tile has been fully placed and routed, we intend to release the updated scripts to the community in the same way as for the SPARC-based OpenPiton.

4 ROADMAP

OpenPiton+Ariane is actively being improved and several extensions and features are planned, as described below. We are also open to any any input from the community in that regard.

4.1 Software and Testing

- **Litmus testing:** We have begun to test the implementation of memory consistency using a variant of the open-source litmus/herd/diy suites [1–3], both in simulation and on FPGA. This testing framework is giving us valuable pre-silicon feedback to enable us to avoid memory consistency violations and will be open-sourced once it is mature.
- **Torture tests:** Another common testing framework is `riscv-torture` [5] which is used to perform many random tests of a RISC-V processor. Ariane supports this infrastructure, but we have not yet included it in OpenPiton+Ariane.

[2]Note that on FPGA, we synthesise the T1 only with one hardware thread, thus there is no hardware multithreading overhead.

- **RISC-V-DV:** This is a recently released UVM-based verification framework for RISC-V RV32IMC/RV64IMC processors by Google [9]. Adding support for this framework could further increase the test coverage of our verification suite.
- **OpenSBI and Bootloading:** The RISC-V community is moving towards a common approach to firmware and bootloading. We plan to move to the community-supported OpenSBI [18] from the Berkeley Bootloader (BBL) that we currently use for firmware. We also plan to add support for a more mainstream bootloader such as U-Boot [21], rather than relying on BBL's bootloading capabilities.
- **Linux Distribution:** Several mainstream Linux distributions, including Fedora and Debian, are beginning to provide RISC-V ports. With OpenSBI and U-Boot in place, we will be able to support these distributions with little to no modification to the software or hardware.

4.2 Hardware

- **Cache Evolution:** The current cache is still parameterised according to the T1 configuration (Table 1), but as we move away from that particular instance, we plan to add further options to OpenPiton. E.g., we plan to add an option to change the cache-line size of all caches to 512 bit as it is done in several other cache-systems today. Further, we plan to optimise the default parameters to provide the optimum configuration in terms of line size, associativity and capacity.
- **FPGA Flows:** We plan to add support for other FPGA targets such as the BittWare XUPP3R (similar to the VCU118) and Amazon's AWS F1 platform. Additionally, a RISC-V recreation of PicoPiton [4] fits on the Digilent Nexys A7 (formerly Nexys 4 DDR) Artix-7 100T FPGA and would be very useful for educational purposes.

5 CONCLUSION

We have presented the extensions and modifications that were necessary to bring together Ariane and OpenPiton, and have given an overview of the main characteristics and supported infrastructure of the resulting OpenPiton+Ariane system. The permissive BSD, Apache 2.0 and Solderpad 0.51 licenses enable everyone to freely use this platform - be it to study computer architecture, try out new ideas, or even to develop next-generation processor platforms. Going forward there are still a lot of exciting features and modifications to be incorporated into this platform and we look forward to engaging with the community to make those happen.

ACKNOWLEDGEMENTS

This material is based on research sponsored by the NSF under Grants No. CNS-1823222, CCF-1453112, and CCF-1823032, Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) under agreements No. FA8650-18-2-7846, FA8650-18-2-7852, and FA8650-18-2-7862. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force

Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA), the NSF, or the U.S. Government.

REFERENCES

[1] Jade Alglave, Luc Maranget, Susmit Sarkar, and Peter Sewell. 2011. Litmus: Running Tests against Hardware. In *Tools and Algorithms for the Construction and Analysis of Systems*, Parosh Aziz Abdulla and K. Rustan M. Leino (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 41–44.

[2] Jade Alglave, Luc Maranget, Susmit Sarkar, and Peter Sewell. 2012. Fences in weak memory models (extended version). *Formal Methods in System Design* 40, 2 (01 Apr 2012), 170–205. https://doi.org/10.1007/s10703-011-0135-z

[3] Jade Alglave, Luc Maranget, and Michael Tautschnig. 2014. Herding Cats: Modelling, Simulation, Testing, and Data Mining for Weak Memory. *ACM Trans. Program. Lang. Syst.* 36, 2, Article 7 (July 2014), 74 pages. https://doi.org/10.1145/2627752

[4] Jonathan Balkind, Michael McKeown, Yaosheng Fu, Tri Nguyen, Yanqi Zhou, Alexey Lavrov, Mohammad Shahrad, Adi Fuchs, Samuel Payne, Xiaohua Liang, et al. 2016. Openpiton: An open source manycore research framework. In *ACM SIGARCH Computer Architecture News*, Vol. 44. ACM, 217–232.

[5] Berkeley Architecture Research (UC Berkeley). 2012. riscv-torture. https://github.com/ucb-bar/riscv-torture. (2012).

[6] John Clayton and Marek Czerski. 2013. Wishbone SD Card Controller. https://opencores.org/projects/sd_card_controller. (2013).

[7] Alex Forencich. 2019. Verilog Ethernet Components. https://github.com/alexforencich/verilog-ethernet. (2019).

[8] Yaosheng Fu, Tri M. Nguyen, and David Wentzlaff. 2015. Coherence domain restriction on large scale systems. In *Proceedings of the 48th International Symposium on Microarchitecture (MICRO-48)*. Association for Computing Machinery, 686–698. https://doi.org/10.1145/2830772.2830832

[9] Google. 2019. RISCV-DV. https://github.com/google/riscv-dv. (2019).

[10] Katie Lim, Jonathan Balkind, and David Wentzlaff. 2018. JuxtaPiton: Enabling Heterogeneous-ISA Research with RISC-V and SPARC FPGA Soft-cores. *CoRR* abs/1811.08091 (2018). arXiv:1811.08091 http://arxiv.org/abs/1811.08091

[11] Katie Lim, Jonathan Balkind, and David Wentzlaff. 2019. JuxtaPiton: Enabling Heterogeneous-ISA Research with RISC-V and SPARC FPGA Soft-cores. In *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '19)*. ACM, New York, NY, USA, 184–184. https://doi.org/10.1145/3289602.3293958

[12] lowRISC. 2019. RGMII Ethernet MAC. https://github.com/lowRISC/ariane-ethernet. (2019).

[13] Michael McKeown, Yaosheng Fu, Tri Nguyen, Yanqi Zhou, Jonathan Balkind, Alexey Lavrov, Mohammad Shahrad, Samuel Payne, and David Wentzlaff. 2017. Piton: A manycore processor for multitenant clouds. *Ieee micro* 37, 2 (2017), 70–80.

[14] Michael McKeown, Alexey Lavrov, Mohammad Shahrad, Paul J Jackson, Yaosheng Fu, Jonathan Balkind, Tri M Nguyen, Katie Lim, Yanqi Zhou, and David Wentzlaff. 2018. Power and Energy Characterization of an Open Source 25-Core Manycore Processor.. In *HPCA*. 762–775.

[15] Oracle. 2008. OpenSPARC T1 Microarchitecture Specification. https://www.oracle.com/technetwork/systems/opensparc/t1-01-opensparct1-micro-arch-1538959.html. (2008).

[16] Ishwar Parulkar, Alan Wood, James C Hoe, Babak Falsafi, Sarita V Adve, Josep Torrellas, and Subhasish Mitra. 2008. OpenSPARC: An open platform for hardware reliability experimentation. In *Fourth Workshop on Silicon Errors in Logic-System Effects (SELSE)*. Citeseer, 1–6.

[17] Princeton University. 2019. OpenPiton Research Platform. https://github.com/PrincetonUniversity/openpiton. (2019).

[18] RISC-V Foundation. [n. d.]. OpenSBI. ([n. d.]).

[19] RISC-V Foundation. 2018. RISC-V External Debug Support Version 0.13 - DRAFT. https://github.com/riscv/riscv-debug-spec/releases/download/task_group_vote/riscv-debug-draft.pdf. (2018).

[20] RISC-V Foundation. 2019. riscv-tests. https://github.com/riscv/riscv-tests. (2019).

[21] Wolfgang Denk. [n. d.]. Das U-Boot. ([n. d.]).

[22] Florian Zaruba and Luca Benini. 2019. The Cost of Application-Class Processing: Energy and Performance Analysis of a Linux-ready 1.7GHz 64bit RISC-V Core in 22nm FDSOI Technology. *arXiv e-prints*, Article arXiv:1904.05442 (April 2019), arXiv:1904.05442 pages. arXiv:cs.AR/1904.05442