

OpenPiton: An Emerging Standard for Open-Source EDA Tool Development

Jonathan Balkind, Alexey Lavrov, Michael McKeown¹, Yaosheng Fu², Tri Nguyen³, Mohammad Shahrads, Ang Li, Katie Lim⁴, Yanqi Zhou⁵, Ting-Jung Chang, Paul J. Jackson, Adi Fuchs, Samuel Payne², Xiaohua Liang⁶, Matthew Matl⁷, David Wentzclaff

Princeton University

{jbalkind, alavrov, mmckeown, yfu, trin, mshahrad, angl, kml4, yanqiz, tingjung, pjj, adif, spayne, xiaohua, mmatl, wentzclaf}@princeton.edu

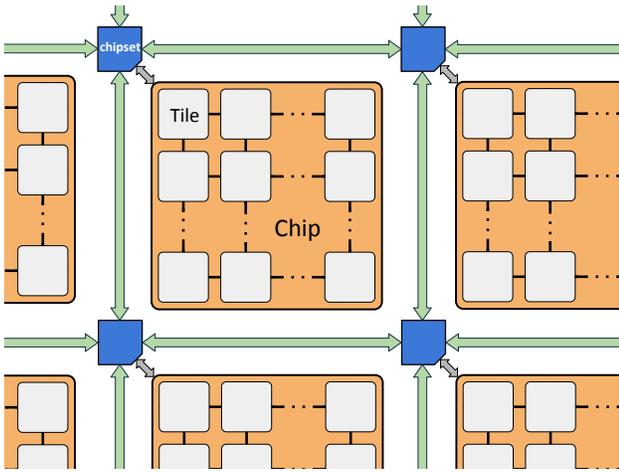


Fig. 1: OpenPiton Architecture, reproduced [1]. Multiple manycore chips are connected together with chipset logic and P-Mesh networks to build large scalable manycore systems. The P-Mesh cache coherence protocol extends off chip.

Abstract—With the transistor count of industrial ASICs reaching the tens of billions, EDA tools must have the scalability to handle such large designs. However, few open-source RTL designs reflect the scale that industrial ASICs are reaching. OpenPiton, on the other hand, has a scalable, tiled manycore design that can reach as many as 65,536 cores in a single chip. This can enable EDA tool developers to test their tools’ functionality at that extreme scale. With its many configurability options, extensive scalability, and open-source ASIC synthesis and back-end flow, the OpenPiton platform is well placed to supercharge open-source EDA tool development.

OpenPiton is the world’s first open-source, general-purpose, multithreaded, manycore processor and framework. The high-level OpenPiton architecture is shown in Figure 1. OpenPiton is designed to provide scalable meshes of tiles on a single chip, with the P-Mesh cache coherence system providing the ability to connect together up to 8192 chips in a single shared-memory system.

A single OpenPiton tile (as shown in Figure 2(a)) consists of a modified OpenSPARC T1 core [8], its CPU-Cache

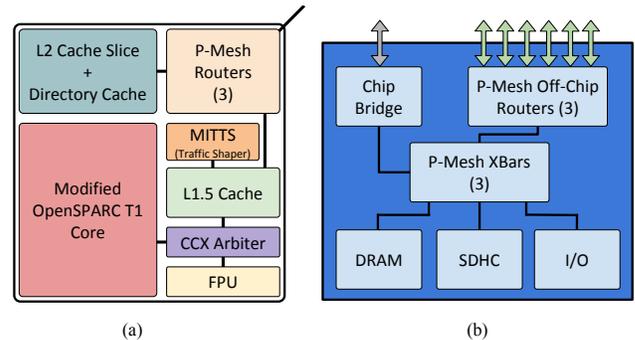


Fig. 2: Architecture of (a) a tile and (b) chipset, reproduced [1] and updated.

Crossbar (CCX) arbiter, an L1.5 cache, an FPU, a slice of the distributed, shared L2 cache, and three P-Mesh Network-on-Chip (NoC) routers. A Memory Inter-arrival Time Traffic Shaper (MITTS) [9] can also be optionally included. The open-source OpenSPARC T1 provided an industrial-strength basis for the development of OpenPiton, with its thorough validation suite containing thousands of assembly tests and the ability to boot a full-stack Linux distribution. We have since scratch-built an entirely new memory system using the P-Mesh cache coherence protocol and modernized and enhanced the codebase, including writing new synthesis and back-end scripts that target new foundry processes and modern FPGAs. We have had eight major releases since our first open source release in June 2015, and are providing active support and enhancements with a roadmap stretching far into the future.

The interconnection of OpenPiton chips is managed by the chipset, shown in Figure 2(b). Our current implementations of the OpenPiton chipset have been on FPGA, where the P-Mesh networks connect to the attached chip via the chip bridge. This has provided us with rapid innovation in chipset design, but the chipset may also be integrated on-chip, providing IP blocks such as memory controllers to create a single, realistic SoC. This enables researchers developing new EDA tools to investigate many aspects of an SoC design within a single, easy-to-use infrastructure.

OpenPiton provides an open-source ASIC synthesis and back-end flow, based on our 25-core chip Piton [6], which

¹Now at Esperanto Technologies. ²Now at NVIDIA. ³Now at Harvard Medical School. ⁴Now at Baidu SVAIL. ⁵Now at University of Washington. ⁶Now at Microsoft. ⁷ Now at University of California, Berkeley.

Component	Configurability Options
Cores (per chip)	Up to 65,536
Cores (per system)	Up to 500 million
Threads per Core	1/2/4
Floating-Point Unit	Present /Absent
Stream-Processing Unit	Present /Absent
TLBs	8/ 16 /32/64 entries
L1 I-Cache	8*/ 16 /32KB
L1 D-Cache	4*/ 8 /16KB
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
Inter-chip Topologies	2D Mesh, 3D Mesh, Crossbar, Butterfly Network
Bootloading	SD/SDHC Card, UART

TABLE I: Supported OpenPiton configuration options. Bold indicates default values. (*Associativity reduced to 2-ways at smallest size). Reproduced [1] and updated.

was taped-out in the IBM 32nm SOI process. This open-source flow remains a rarity among open-source processors. We have taken the time to remove foundry-specific references and release scripts for both leaf-level modules and for our top-level hierarchically placed and routed chip. The Piton Processor that was the result of this flow has recently been thoroughly characterized [7], including its power and energy characteristics, and an exceptionally detailed area breakdown. Our methodology and all data for this characterization are now open, as is our printed circuit-board (PCB). The availability of this data in conjunction with the open-source RTL provides a unique silicon-verified platform for development of new EDA tools. All of the characterization data, as well as the OpenPiton source code, FPGA bitfiles, OS images, development virtual machines and documentation are available for download at <http://www.openpiton.org>.

I. SCALABILITY, CONFIGURABILITY, AND MODULARITY

OpenPiton can be configured across a rich design space of parameters. Table I shows the options that are available for configuration by the user. Each of these options can be specified at design time through a change to a single parameter in the relevant configuration file. Many of these options are provided through the use of the Python Hypertext Processor (PyHP), which introduces a simple preprocessing step that executes Python code embedded inside the Verilog source files. This means that our tools only take standard Verilog code as input, while the RTL designer can make use of higher-level concepts in Python to produce the Verilog code that is input to the EDA tool flow. We have found this to be a more user-friendly and cross-compatible approach than introducing a new hardware description language.

On top of its scalability and configurability, OpenPiton is built from a diverse set of modules that will enable the developers of open-source EDA tools to stress-test their tools in a variety of different aspects. To evaluate a tool's handling of large memories, one can synthesize and evaluate the L1.5 or L2 caches across a variety of parameters. If long wires need special treatment, one can study the placement

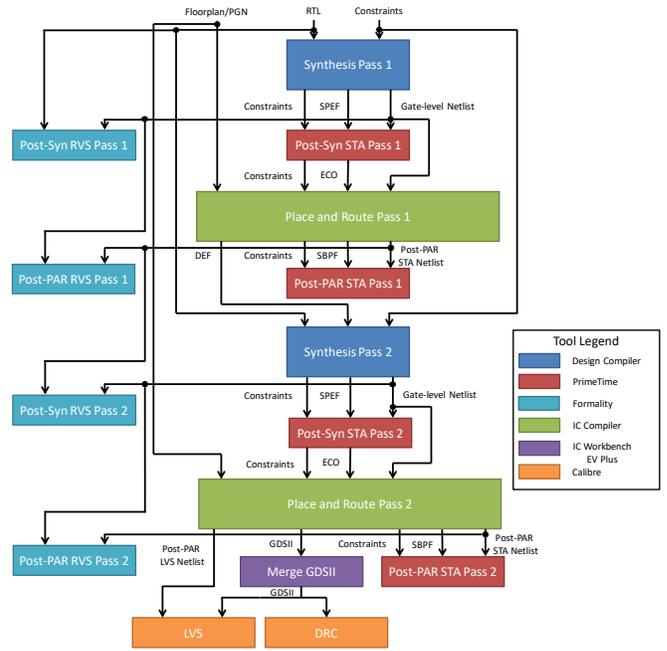


Fig. 3: OpenPiton's ASIC Synthesis and Back-end Flow

and routing of OpenPiton's P-Mesh NoC routers and their associated buses. Off-chip interfaces can be scrutinized in the context of OpenPiton's Chip Bridge. To evaluate a full-system including all of these components, we provide our hierarchical flow that includes everything contained in the physical Princeton Piton chip.

We have also developed OpenPiton with a keen eye towards portability. We support behavioral simulation across Synopsys VCS, Cadence Incisive, Mentor ModelSim, and the open-source Icarus Verilog. Our on-chip memories are designed to be replaceable with foundry-specific SRAMs and our FPGA implementations are written to provide correct inference across a variety of synthesis tools. Clocking and other IP blocks are also modularized for easy replacement when targeting a new technology.

II. OPENPITON'S OPEN ASIC SYNTHESIS AND BACK-END FLOW

The open ASIC synthesis and back-end flow that we provide with OpenPiton is detailed in Figure 3. This flow provides a quality reference for a desirable set of features in an open-source EDA tool flow. It features synthesis, static timing analysis, formal equivalence checking, place and route, design rule checking, and more. The flow's validated functionality means that EDA tool developers can replace individual tools in the flow to verify that they are functionally correct in the context of the full flow.

The open-source Piton characterization data and methodology also provide a well-documented design point reflecting a silicon-verified design that new EDA tools can target. The thorough area breakdown also enables researchers to evaluate the Quality of Results (QoR) and determine the expected QoR of their own tools.

Module Name	Description	Purpose
ffu	OpenSPARC T1 core floating-point front-end unit	Small module with one SRAM macro
sparc	OpenSPARC T1 core	Large module with many SRAM macros
dynamic_node	OpenPiton on-chip network router	Small module with no IP macros
tile	OpenPiton tile	Large, hierarchical module with many SRAM macros
chip	OpenPiton top-level chip	Large, top-level hierarchical module

TABLE II: Supported Modules for the OpenPiton ASIC Synthesis and Back-end Flow

Table II details the set of modules that we presently provide scripts for. Each module is included for a particular purpose, showing how a user can scale from a very small block with no IP macros, up to a large, hierarchical module with a variety of SRAM and other IP macros.

III. OPEN SOURCE EDA WITH OPENPITON

OpenPiton has been available at <http://www.openpiton.org> since June 2015. Since then, OpenPiton has been downloaded thousands of times, from more than 70 countries around the globe. We have made eight major releases, with a roadmap to continue releasing far into the future. OpenPiton is being actively used by multiple commercial EDA vendors and DARPA has identified OpenPiton as a benchmark for use in the POSH program. Our online discussion forum¹ has grown into a community where users can report issues and actively help each other to develop solutions. Our users have also identified bugs, helped us implement fixes, and have even begun to maintain their own ports for other FPGAs that we do not support by default. Users are choosing OpenPiton as a research platform thanks to its open-source ASIC synthesis and back-end scripts. These users would benefit greatly from a fully open-source tool flow, just as the quality of the OpenPiton platform would improve from being more broadly accessible.

We believe OpenPiton is an ideal framework for use in the development of open-source EDA tools through its high-quality code base, its growing user base, and its focus on usability to enable more efficient research.

A. Reducing Human Effort

In the process of developing OpenPiton for both our own use and external use, we have found it essential to reduce the cost of engineering time. Even for a small team aiming to tape out an academic chip, the cost of the tape-out can easily be dwarfed by the cost of paying students and researchers. As a result, we have focused on making all of our tools push-button, and have tried to move as much configurability to a single file or command-line option as we can. With our framework, users can easily replace the technology libraries and tools used in any step of the simulation and backend flows. The new, custom flow can then be validated hierarchically on a wide set of inputs ranging from a single module to the full design. We have also made use of our online discussion forum to determine the pain points for

our users in order to improve the user experience as much as possible in succeeding releases. One example of this is reducing Linux boot time on FPGA from around 45 minutes with the original OpenSPARC T1, to just 4 minutes with OpenPiton.

B. Visibility

With traditional, closed-source designs, tool vendors rely on their customers tracking down a bug and producing a minimal test case. Making use of an open-source and well-documented RTL design like OpenPiton during the development process enables the tool developer to identify which parts of the design might be causing issues. Given that OpenPiton is reflective of industrial-quality RTL that is portable across tools from a number of vendors, it provides an excellent basis for evaluating how well new tools can be expected to work in the context of real-world designs.

C. Enabled Research

The OpenPiton platform enables rapid prototyping and innovation for researchers in a variety of fields. Our own research has relied heavily on OpenPiton for prototyping purposes, and there are a number of examples of external teams making use of OpenPiton to take their EDA research ideas to functional full-system prototypes in a matter of months.

One such example is Elnaggar et al. [2] using performance counters from an OpenPiton system running Debian Linux on an FPGA to identify hardware trojans present in the design. Lerner et al. [5] produced a flow that uses OpenPiton and the gem5 simulator to improve an OpenPiton-based multicore IoT system's reliability lifetime by 4.1x. OpenPiton has also proved useful as a platform for investigating power analysis of scan testing [4].

Another novel application for OpenPiton in the EDA space was in the design of a scheme for system-level modeling and control of Electromagnetic Interference (EMI) produced by an SoC [3]. Here, an OpenPiton system was evaluated on FPGA, with the technique also being applicable to other ASICs, like Piton.

IV. A MUTUALLY BENEFICIAL CYCLE

We believe that OpenPiton is a great match for the development of open-source EDA tools, because of its streamlined configurability and its industrial-quality RTL. We also believe that OpenPiton (and the wider community) would benefit in return from the development of such open-source tools. We look forward to being able to open more of our

¹Join us at <https://groups.google.com/group/openpiton>

flow by targeting open-source EDA tools as they become available. Our goal is to work with the community to provide a fully open-source ASIC synthesis and back-end flow, which would be beneficial for education, research, and business.

ACKNOWLEDGMENTS

This material is based on research sponsored by the NSF under Grants No. CNS-1823222, CCF-1217553, CCF-1453112, and CCF-1438980, AFOSR under Grant No. FA9550-14-1-0148, Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) under agreement No. FA8650-18-2-7846 and FA8650-18-2-7852 and DARPA under Grants No. N66001-14-1-4040 and HR0011-13-2-0005. 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, AFOSR, DARPA, or the U.S. Government. We thank Fei Gao, Felix Madutsa, and Kathleen Feng for their important contributions to OpenPiton.

REFERENCES

- [1] J. Balkind, M. McKeown, Y. Fu, T. Nguyen, Y. Zhou, A. Lavrov, M. Shahrad, A. Fuchs, S. Payne, X. Liang, M. Matl, and D. Wentzlaff. OpenPiton: An open source manycore research framework. In *Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '16*, pages 217–232, New York, NY, USA, 2016. ACM.
- [2] R. Elnaggar, K. Chakrabarty, and M. B. Tahoori. Run-time hardware trojan detection using performance counters. In *2017 IEEE International Test Conference (ITC)*, pages 1–10, Oct 2017.
- [3] D. I. Gorman. *DEMIS: Dynamic EMI Shifting*. PhD thesis, University of California, Santa Cruz, 2018.
- [4] I. Indino. An open source platform and EDA tool framework to enable scan test power analysis testing. Master's thesis, University of Limerick, 2017.
- [5] S. Lerner and B. Taskin. Workload-aware asic flow for lifetime improvement of multi-core iot processors. In *2017 18th International Symposium on Quality Electronic Design (ISQED)*, pages 379–384, March 2017.
- [6] M. McKeown, Y. Fu, T. Nguyen, Y. Zhou, J. Balkind, A. Lavrov, M. Shahrad, S. Payne, and D. Wentzlaff. Piton: A manycore processor for multitenant clouds. *IEEE Micro*, 37(2):70–80, Mar 2017.
- [7] M. McKeown, A. Lavrov, M. Shahrad, P. Jackson, Y. Fu, J. Balkind, T. Nguyen, K. Lim, Y. Zhou, and D. Wentzlaff. Power and energy characterization of an open source 25-core manycore processor. In *High Performance Computer Architecture (HPCA), IEEE Int. Symposium on*, 2018.
- [8] Oracle. OpenSPARC T1. <http://www.oracle.com/technetwork/systems/opensparc/opensparc-t1-page-1444609.html>.
- [9] Y. Zhou and D. Wentzlaff. MITTS: Memory inter-arrival time traffic shaping. In *Proceedings of the 43rd International Symposium on Computer Architecture, ISCA '16*, pages 532–544, Piscataway, NJ, USA, 2016. IEEE Press.