A 12nm Agile-Designed SoC for Swarm-Based Perception with Heterogeneous IP Blocks, a Reconfigurable Memory Hierarchy, and an 800MHz Multi-Plane NoC

Tianyu Jia[1,*], Paolo Mantovani[2,*], Maico Cassel dos Santos[2,*], Davide Giri[2], Joseph Zuckerman[2], Erik Jens Loscalzo[2], Martin Cochet[3], Karthik Swaminathan[3], Gabriele Tombesi[2], Jeff Jun Zhang[1], Nandhini Chandramoorthy[3], John-David Wellman[3], Kevin Tien[3], Luca Carloni[2], Kenneth Shepard[2], David Brooks[1], Gu-Yeon Wei[1], Pradip Bose[3]

[1]Harvard University, Cambridge, MA, [2]Columbia University, New York, NY, [3]IBM Research, Yorktown Heights, NY

* These authors have equal contributions. Email: pbose@us.ibm.com

ESSCIRC 2022- IEEE 48th European Solid State Circuits Conference (ESSCIRC) | 978-1-6654-8494-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ESSCIRC55480.2022.9911456

***Abstract*— This paper presents an agile-designed domain-specific SoC in 12nm CMOS for the emerging application domain of swarm-based perception. Featuring a heterogeneous tile-based architecture, the SoC was designed with an agile methodology using open-source processors and accelerators, interconnected by a multi-plane NoC. A reconfigurable memory hierarchy and a CS-GALS clocking scheme allow the SoC to run at a variety of performance/power operating points. Compared to a high-end FPGA, the presented SoC achieves 7× performance and 62× efficiency gains for the target application domain.**

I. Introduction

The slowdown of CMOS scaling and limited effectiveness of parallelism via homogeneous multi-core processors have pushed modern computing systems toward heterogeneous SoC architectures. Heterogeneous architectures deliver superior energy-efficient performance by combining general-purpose processors with fixed-function accelerators. Heterogeneity, however, increases the complexity of the design and verification process. Open-source hardware (OSH) addresses this complexity challenge by promoting design reuse [1]. This work focuses on the emerging application domain of vehicular swarm perception (Fig. 1), which expands the vehicle perceptive field by sharing neighbors' sensor data through wireless V2V (vehicle-to-vehicle) communication and reduces false predictions [2-3]. Compared to the computations for the autonomous driving of a single vehicle [4], which include CNNs for object detection and general-purpose computing for decision making, swarm-based perception additionally relies on FFT and Viterbi decoding for sensor signal processing and wireless communication. Hence, this leads to the design of an SoC architecture with a highly heterogeneous architecture.

This paper presents a domain-specific SoC with a tile-based architecture for the target application domain. We designed the SoC with an agile design methodology that promotes the reuse of existing OSH IP blocks and simplifies the development of new ones. Fig. 1 shows its main steps: 1) the SoC components are selected from a library of reusable OSH IPs based on extensive workload analysis; 2) the tile sockets seamlessly integrate the OSH IPs, and the generation of the full SoC RTL is automated based on parameterized configurations; 3) a hierarchical physical design strategy leverages the modularity of the tile-based architecture and clocking scheme. Compared to agile design approaches for homogeneous multi-core chips [5], our methodology mitigates the complexity of heterogeneous SoC design by decoupling the design and integration of the heterogeneous IPs. Our approach scales up for the development of SoCs with larger and more heterogeneous arrays of tiles.

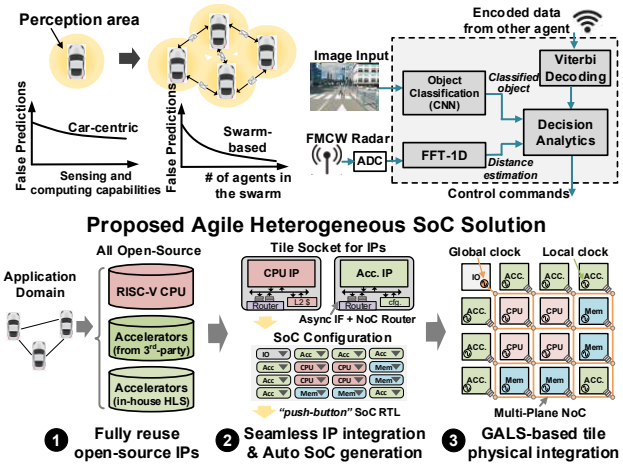

Fig. 1 Swarm-based perception in autonomous vehicle with its key computation kernels, and the agile-designed heterogeneous SoC with a tile-based architecture.

II. SoC Architecture and Agile Design Methodology

Fig. 2 shows the overall SoC architecture comprising an array of 4×4 tiles connected by a 2D-mesh multi-plane network-on-chip (NoC). One of the four RISC-V CPU cores [6] acts as the host and boots the Linux operating system. To support parallel processing of camera and sensor data inputs, three NVDLA DNN inference accelerators [7] and three FFT accelerators are deployed to perform object detection and distance estimation tasks. One Viterbi accelerator is deployed to decode the incoming vehicle messages. Both the FFT and Viterbi accelerators are designed in-house using high-level synthesis (HLS) [8]. For higher modularity, each IP block is encapsulated within a tile socket, which connects it to a local bus

(e.g., AXI4) as a master and is connected to the NoC via asynchronous interfaces. The tile socket also implements system-level services, including services specific to the particular type of tile: e.g. DMA and configuration registers for an accelerator tile. The NoC, tile sockets, and distributed reconfigurable memory hierarchy are extended from ESP, an open-source SoC platform [9].

The last-level cache (LLC) is partitioned into four memory tiles, each containing a 64-bit wide off-chip memory link. Combined, the four off-chip links support the real-time workload bandwidth requirements. Any subset of the memory tiles can be selected at runtime, and the memory hierarchy is reconfigurable to support different cache-coherence modes. The test-chip prototype relies on a modular FPGA system to connect each memory tile via FMC connectors to a 2GB DDR3 card, which stores image and sensor data for swarm perception. There is one IO tile in the SoC containing ROM and peripheral IO, such as Ethernet and UART.

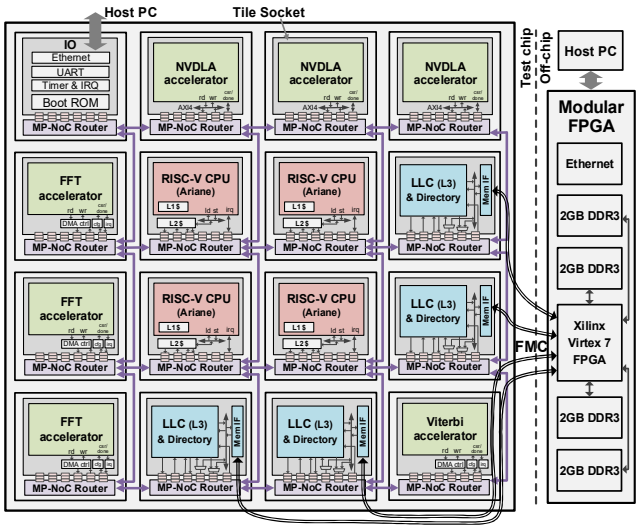

Fig. 2 The architecture of the domain-specific heterogeneous SoC.

Fig. 3 illustrates the details of the agile design methodology, which leverages the modularity of the tile-based architecture. Driven by the analysis of the target swarm-based perception application[1], the key computation kernels, e.g. CNN, FFT, Viterbi decode, are identified. Existing OSH IPs are evaluated and selected for these key computations. Reusing existing OSH IP blocks can significantly reduce the SoC design cycle. Each IP is seamlessly integrated into the SoC by the tile socket, which decouples its design from the rest of the system, thus simplifying the integration of heterogeneous blocks. Instanced from the open-source ESP SoC platform [9], the full SoC RTL, including the NoC and system-level services, is automatically generated. Co-generation of corresponding testbenches is also provided to enable rapid evaluation of the SoC performance and architecture optimizations by FPGA emulation.

During physical design, the inherent regularity of the tile-based architecture decouples each tile from its location in the top-level floorplan, i.e., the same tile can be replicated to meet workload requirements. A hierarchical timing-closure flow is adopted for independent timing signoff between the local clock frequency of each tile and the global NoC frequency. The physical design of all tiles is conducted in parallel, while the global NoC timing is closed later based on the interface logic model (ILM) timing models. Such timing closure flow allows flexible reuse or respin of pre-existing IPs, further trimming design time. The entire SoC exclusively uses synthesizable designs to avoid any manual layout effort.

Thanks to our agile SoC design methodologies, the proposed SoC was designed in 4 months by less than 10 full-time designers. This design cycle is considerably shorter than the 7 month cycle of a prior agile-designed multi-core processor [5] and was achieved despite the additional challenges posed by the higher heterogeneity of the SoC architecture.

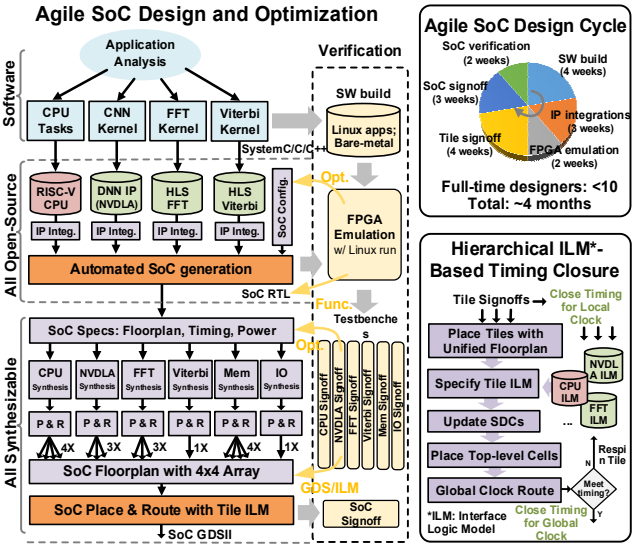

Fig. 3 Agile SoC design/optimization and ILM-based timing closure.

III. System-Level Services and Dynamic Reconfiguration

The tile socket provides each IP with system-level services such as DMA access and local reconfigurability. To support data exchange among heterogeneous tiles, the mesh NoC has six physical planes, as shown in Fig. 4. Planes 1-3 provide coherence channels between CPUs, accelerators, and LLC partitions. Planes 4-5 support DMA access for the accelerators. Plane 6 is dedicated to interrupts and memory-mapped IO and registers. Asynchronous buffers in the tile sockets connect the NoC routers in the six planes to the logic inside each tile.

In the SoC, the accelerators can communicate with the memory hierarchy via three dynamically configurable cache-coherence modes [10]: non-coherent DMA (accelerator bypasses the cache hierarchy and accesses main memory directly), LLC-coherent DMA (memory requests are sent directly to the LLC and coherence is enforced by software), and the coherent DMA (memory requests are sent directly to the LLC and the hardware maintains full coherence). Each mode supports different degrees of hardware coherence and offers distinct benefits depending on the active workloads, system-level contention, and accelerator properties.

[1] Target workload Mini-ERA: https://github.com/IBM/mini-era

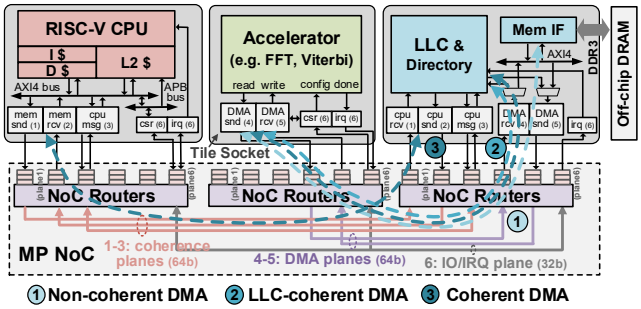

Fig. 4 The multi-plane NoC interconnection for heterogeneous tiles and three reconfigurable cache coherence modes.

The SoC implements a communication synchronous GALS (CS-GALS) clocking strategy, as shown in Fig. 5. Each IP tile is synchronous to a local clock (*clk_tile*) with a local power supply. The NoC, which sits in a global power domain, is synchronously driven by a chip-wide global clock (*clk_noc*). During the timing closure, only the clock skews between neighboring tiles need to be constrained, which relaxes traditional timing-closure constraints. To support the heterogeneity of the IP blocks, the frequency of each tile, as well as the NoC, can be adjusted dynamically and independently of one another, making them globally asynchronous. The NoC router consists of crossbar switches routed to four neighbors synchronously and to the local tile via asynchronous interfaces. Together with look-ahead router design, the CS-GALS enables single-hop-per-cycle throughput for each plane, which outperforms the asynchronous NoC strategy in prior tile-based SoCs [11, 13-14].

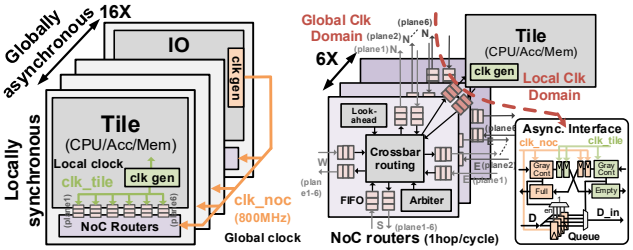

Fig. 5 CS-GALS clocking and NoC router w/ asynchronous interface.

IV. Measurement Results

The presented domain-specific SoC is fabricated using 12nm FinFET technology. Fig. 6 shows the die photo and its test setup with a modular FPGA system. In the SoC, each tile occupies an identical 1×1mm area for flexible placement and consistent timing closure between the NoC routers. The IP designs occupy more than 90% of the area in each tile, while the NoC logic is placed on the periphery for straight tile-to-tile routing. Only two accelerators (FFT, Viterbi) slightly underutilize tile area, and the entire design incurs less than 15% overhead compared to a bespoke design with custom sizes for each tile. The total active area of the SoC is 21.6mm^2.

The chip is assembled on a flip-chip package containing 18 power domains. During testing, the test board is connected to the FPGA motherboard through 3 FMC connectors. The FPGA test system is modular, with the flexibility of utilizing different daughter cards. An Ethernet link provides a debug interface from a PC for accessing memory-mapped regions of the SoC.

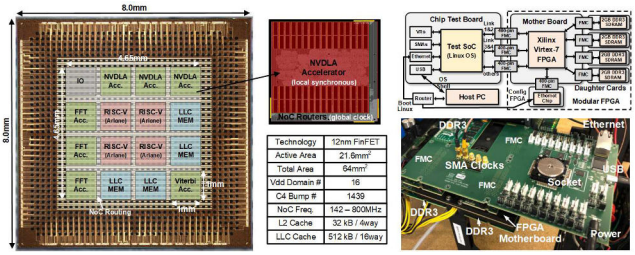

Fig. 6 Die photo and the test setup with modular FPGA system.

We first evaluate the performance and benefits of each tile. As shown in Fig. 7, the operating frequencies of each accelerator were measured across a range of supply voltages, from 0.5V to 1V. We run the CPU with a minimum supply voltage of 0.7V for stable operation of the operating system. We present the benchmark measurements at the nominal 0.8V. The deployment of accelerators significantly improves the performance, i.e. workload latency, compared to standalone CPU operations. At 0.8V, offloading computation to an FFT accelerator achieves 71× and 233× latency and energy reductions, respectively. Similarly, offloading the Viterbi decode kernel to its dedicated accelerator obtains a 20× latency and 56× energy improvement.

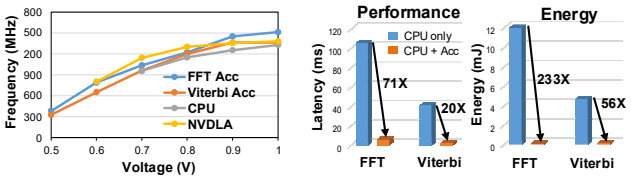

Fig. 7 (Left) V/F scaling for each tile, (Right) Benefits of offloading tasks to dedicated accelerators.

The benefit of the reconfigurable memory hierarchy is evaluated across different workload sizes, as shown in Fig. 8. When the accelerator and CPU share an LLC partition (e.g., when using one memory tile to save energy), the non-coherent DMA mode performs best by avoiding LLC contention, as it accesses DRAM directly. When the accelerator owns its own dedicated LLC partition, there are significant performance benefits from the coherent-DMA and LLC-coherent DMA modes, in which the accelerator performs DMA directly to the LLC and potentially avoids off-chip DRAM access.

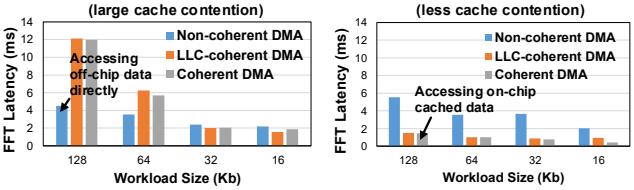

Fig. 8 (Left) Accelerator share one LLC partition with the CPU, (Right) Accelerator owns its dedicated LLC partition.

The accelerator performance is highly correlated with the memory bandwidth, which varies at runtime depending on the SoC operations. The proposed tile-based architecture simplifies the dynamic provisioning of the available four LLC partitions and corresponding off-chip memory links to meet workload demands, e.g. by scaling them up to match the parallel execution of accelerators for performance improvement. As illustrated in Fig. 9, the LLC memory partitions and the corresponding off-

chip links are scaled together with the FFT accelerator parallelism to avoid a memory bottleneck. The CS-GALS approach also allows each accelerator to run at its optimal frequency, independently from the rest of the SoC. For example, when the workload is memory bound, the frequency of the FFT accelerator can be reduced from the maximum 1.2GHz to 470MHz while maintaining similar workload latency, thus achieving a 2× energy reduction.

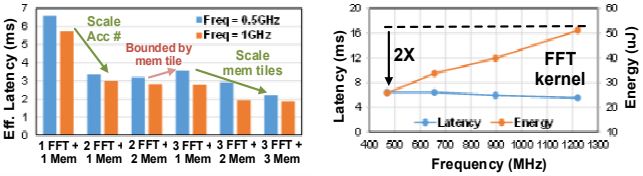

Fig. 9 (Left) Accelerator performance with scalable LLC partition and memory links, (Right) Scale frequency under CS-GALS.

Altogether, for the swarm perception workload Mini-ERA, our SoC achieves a 7× performance and 62× energy improvement compared to an implementation of the same design on a high-end Xilinx Virtex UltraScale XCVU440 FPGA, as shown in Fig. 10. For this workload, the SoC consumes 1.36W at 0.8V, with 7.2% attributable to the NoC. The measured NoC frequency reaches 800MHz at 0.8V.

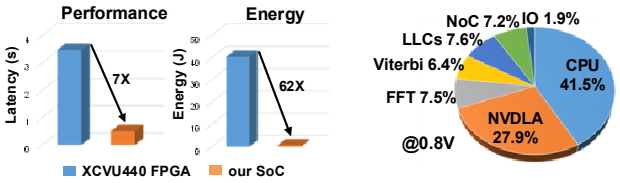

Fig. 10 (Left) Benefits for the target swarm perception application, (Right) SoC power breakdown.

Fig. 11 compares this work to prior tile-based chip designs, which feature *homogeneous* arrays of either processors [11-12] or accelerators [13]. In contrast, the proposed SoC contains a variety of *heterogeneous* OSH IPs and achieves much higher heterogeneity than prior open-source SoCs [15]. The NoC enables data transfers between the tiles with a maximum 281Gb/s throughput, while supporting reconfigurability of the memory hierarchy for a range of performance/power operating points. The scalability of the proposed SoC architecture and the associated methodology benefit engineering productivity when designing SoCs with even larger tile arrays.

	JSSC'17 [11]	VLSI'19 [12]	JSSC'20 [13]	This work
Process	32nm	16nm	16nm	**12nm**
Area	57.41 mm^2	15.25 mm^2	6 mm^2	**21.6 mm^2**
Design Target	CPU Processor	CPU Processor	DNN Accelerator	**Domain-Specific SoC**
Tile Array Size	1000	496	16	**16**
Tile Property	Homogeneous	Homogeneous	Homogeneous	**Heterogeneous**
Tile(s)	16b RISC CPU	32b RISC-V CPU	MAC Array	**64b RISC-V, NVDLA, FFT, Viterbi, Mem, IO**
GALS Clocking	Yes	No	Yes	**Yes**
Tile-to-Tile Clock	Asynchronous	Synchronous	Asynchronous	**Synchronous**
Clock Domains	2012	3	20	**17**
NoC planes x BW	1x16b	1x32b	1x64b	**5x64b + 1x32b**
Tile-to-Tile BW	45.5 Gb/s	32 Gb/s	70 Gb/s	**281 Gb/s**
NoC Power in SoC	~7%	Not reported	10%	**7.2%**
Frequency	115 - 1770 MHz	10 – 1400 MHz	161 – 2001 MHz	**165 - 1520MHz**
Voltage	0.56 – 1.10V	0.6 – 0.98 V	0.41 – 1.2 V	**0.5 - 1.0V**
Power	1.3 - 39.6 W	7.47 W	30mW – 4.16 W	**240mW* – 1.83 W**

* Memory links and NoC at 0.8V

Fig. 11 The comparison table with other tile-based designs.

V. Conclusion

We presented an SoC with a tile-based architecture for the application domain of swarm-based perception. We developed the SoC with an agile design methodology that simplifies the reuse of OSH IPs. The heterogeneous IPs are integrated with tile sockets, which enable system-level services, and are interconnected by a multi-plane NoC. The CS-GALS clocking and reconfigurable memory hierarchy allow flexible performance tuning based on workload demands. The SoC delivers 7× performance and 62× efficiency gains compared to a high-end FPGA implementation.

Acknowlegement

This research was developed with funding from DARPA. The views, opinions and/or other findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Distribution Statement A. Approved for public release: distribution unlimited.

References

[1] D. Daly, "EE5: Is an open-source hardware revolution on the horizon?," *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 555-557, Feb. 2020.

[2] T. Wang, et al., "V2VNet: vehicle-to-vehicle communication for joint perception and prediction", *European Conference on Computer Vision (ECCV)*, pp. 605-621, 2020.

[3] E. Sisbot, A. Vega, et al., "Multi-vehicle map fusion using GNU radio", *Proceedings of the GNU Radio Conference*, vol. 4, no. 1, 2019.

[4] K. Matsubara, et al., "A 12nm autonomous-driving processor with 60.4TOPS, 13.8TOPS/W CNN executed by task-separated ASIL D control", *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 56-57, Feb. 2021.

[5] C. Schmidt, et al., "An eight-core 1.44GHz RISC-V vector machine in 16nm FinFET", *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 58-59, Feb. 2021.

[6] F. Zaruba and L. Benini, "The cost of application-class processing: energy and performance analysis of a Linux-ready 1.7-GHz 64-Bit RISC-V core in 22-nm FDSOI technology", *IEEE Transactions on VLSI Systems (TVLSI)*, vol. 27, no. 11, pp. 2629-2640, Nov. 2019.

[7] Nvidia, NVIDIA Deep Learning Accelerator (NVDLA). http://nvdla.org/primer.html, 2018.

[8] B. Khailany, et al., "A modular digital VLSI flow for high-productivity SoC design", *Design Automation Conference (DAC)*, 2018.

[9] P. Mantovani, et al., "Agile SoC development with open ESP", *IEEE International Conference on Computer-Aided Design (ICCAD)*, 2020.

[10] J. Zuckerman, et al., "Cohmeleon: learning-based orchestration of accelerator coherence in heterogeneous SoCs", *IEEE International Symposium on Microarchitecture (MICRO)*, pp. 350-365, Oct. 2021.

[11] B. Bohnenstiehl, et al., "KiloCore: A 32-nm 1000-processor computational array", *IEEE Journal of Solid-State Circuits (JSSC)*, vol. 52, no. 4, pp. 891-902, Apr. 2017.

[12] A. Rovinski, et al., "A 1.4 GHz 695 Giga Risc-V inst/s 496-core manycore processor with mesh on-chip network and an all-digital synthesized PLL in 16nm CMOS", *IEEE Symposium on VLSI Circuits (VLSI)*, pp. C30-C31, Jun. 2019.

[13] B. Zimmer, et al., "A 0.32–128 TOPS, scalable multi-chip-module-based deep neural network inference accelerator with ground-referenced signaling in 16nm", *IEEE Journal of Solid-State Circuits (JSSC)*, vol. 55, no. 4, pp. 920-932, Apr. 2020.

[14] M. Fojtik, et al., "A fine-grained GALS SoC with pausible adaptive clocking in 16 nm FinFET", *IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pp. 27-35, May 2019.

[15] A. Gonzalez, et al., "A 16mm^2 106.1 GOPS/W heterogeneous RISC-V multi-core multi-accelerator SoC in low-power 22nm FinFET", *IEEE European Solid State Circuits Conference (ESSCIRC)*, Sep. 2021.