

Ballast: Implementation of a Large MP-SoC on 22nm ASIC Technology

Antti Rautakoura*, Timo Hämäläinen*, Ari Kulmala*, Tero Lehtinen†, Mehdi Duman†, Mohamed Ibrahim†

*Tampere University, Finland –

antti.rautakoura@tuni.fi, timo.hamalainen@tuni.fi, ari.kulmala@tuni.fi

†Nokia, Finland –

tero.lehtinen@nokia.com, mehdi.duman@nokia.com, mohamed.7.ibrahim@nokia.com

Abstract—Chips have become the critical asset of the technology, and increasing effort is put to design System-on-Chips (SoC) faster and more affordable. Typically the focus of the research has been on the Power, Performance and Area optimization of the specific component or sub-system. To improve the situation we report design effort for complex SoC counted from specification to ASIC tape-out to lay out a solid reference for the community. Ballast is the first SoC-Hub chip taped out on 22nm technology. It includes six sub-systems on 15 mm² area and reaches 1.2GHz top speed. The design team included 24 persons and spent 21 200 person hours to tape-out in one calendar year from scratch. This is an outstanding achievement and sets the baseline to SoC design productivity development.

Index Terms—System-on-Chip, ASIC, System architecture, Methodology development

I. INTRODUCTION

Chips have become a critical asset in the modern world, which has increased the overall volume of System-on-Chip design projects. Two new trends can be observed. New companies that have not been before in the business have started own SoC designs to differentiate in competition. On the other hand, open source cores, design frameworks and tools are gaining more interest to lower development costs. Both trends call for increased design productivity, which often counts for shorter calendar time schedules [1]. SoC-Hub Finland¹ is a recent multi-partner research opening aiming at sustaining "one chip per year cadence" productivity. To achieve this, SoC-Hub seeks for new design methodologies, collaboration approaches and creation of reusable IPs and subsystems. The main focus is on large and complex SoCs in which integration expertise is essential. The main goals are 1) creation of a versatile SoC template and 2) development of agile HW development design process. SoC-Hub started in 2020 by building the Electronic Design Automation (EDA) tool environment, contracting ASIC technology for the samples, making agreements between the stakeholders and building the development team. Ballast is the first chip in a series of three chips completed by 2023. It is Edge capable, general-purpose SoC with four processor subsystems, machine learning accelerator and versatile interfaces. It allows product-level demonstrators as well as next generation SoC prototyping through extension interfaces. This paper reports the Ballast SoC architecture, the design

process, project organization and design effort. To the best of our knowledge, this paper presents the most accurate design effort analysis among the SoC implementation publications.

This paper is organized as follows. The next section II discusses the related work. Project setup is described in section III followed by design methodology at section IV. SoC architecture is covered in section V and results are introduced in section VI. Finally Conclusions are drawn in section VII.

II. RELATED WORK

The Quentin [2] ASIC is based on an open-source Pulpissimo SoC which is parallel ultra-low power (PULP) based 32-bit single core RISC-V MCU implemented on 22nm Fully-depleted Silicon-On-Insulator (FD-SOI) technology. The work reports 2,3mm² SoC area and maximum clock frequency of 930 Mhz

Chip called Mr. Wolf [3] is PULP platform based SoC for energy-precision scalable IoT Edge Processing. The design includes energy-efficient RISC-V CPU sub-system with IO connectivity and memory shared with eight core parallel computing clusters. The ASIC flow results report 10mm² and maximum clock frequency of 450Mhz with 40nm LP CMOS technology.

The Vega chip [4] is a 10-core SoC for IoT End-Nodes with non-volatile internal MRAM and two ML accelerators. The 12 mm² layout has been implemented with 22nm FD-SOI and 450 Mhz as maximum clock frequency.

Bailey et al. [5] introduce 25 mm² SoC implemented with 16nm FinFET technology at 410Mhz maximum frequency (taped out at 300Mhz). The SoC is targeted for Digital Signal Processing (DSP) applications and includes general-purpose RISC-V core with vector extensions, fixed-function DSP accelerator and mixed signal analog to digital converters. Importantly, this work reports results about design productivity and organization setup of two separate teams. Design productivity was measured with release frequency of selected sub-system, and total design time of 14 00 engineering hours was reported. However, it was not clearly specified what was included in the design time i.e. the RTL design or full project execution from architectural pre-work to ASIC physical design.

BlackParrot presented by Petrisko et al. [6] is a heterogeneous cache-coherent tile-based platform architecture with Linux capable 64-bit RISC-V host CPU. Other tile or

This work has been funded by Business Finland SoC-hub project.

¹www.sochub.fi

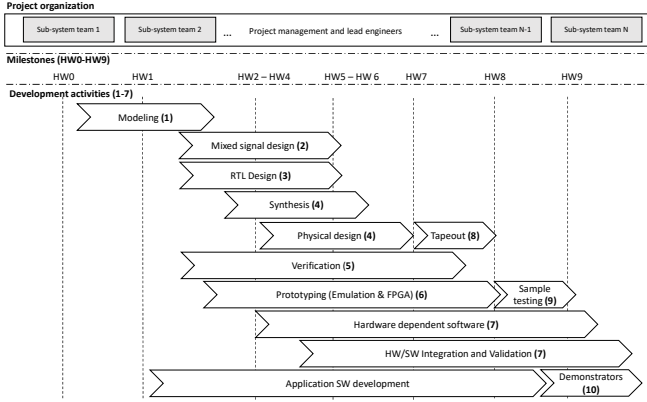


Fig. 1. SoC-Hub development process and organization

components provided are for example coherent accelerator tile, streaming accelerator tile and Network-on-Chip (NoC) components. The BlackParrot has been taped out on 12 nm technology including four CPU tiles. The purpose of the tape-out has been to close full design cycle and get important feedback on the design quality.

Gonzalez et al. report 16mm² heterogeneous multi-core SoC at 22nm technology [7]. The chip includes 64-bit general-purpose superscalar out-of-order RISC-V CPU with vector accelerator and the domain dedicated to machine learning includes 64-bit in-order RISC-V CPU with systolic array DNN accelerator.

SiFive Unleashed 540 SoC was (2020) first RISC-V based commercial Linux-capable multi-core SoC with DDR4 memory sub-system [1]. The SoC has been fabricated with 28nm technology and is able to run CPUs on 1,5 Ghz frequency. The die area of the chip is 30mm². The work reports capability to build large system with fewer people, but no exact measures are given.

As a summary state of the art related work reports complexity close to our Ballast SoC design. Although SoC research focus on productivity and cost reduction improvements, detailed measures of design time are not reported. The summary of the SoCs presented in the related work can be found from the section VI-C. The purpose is not a technical comparison, instead more details are provided to be able to get better visibility to chip complexity.

III. DEVELOPMENT PROCESS AND PROJECT ORGANIZATION

The common understanding of the development process is that it is an important tool for efficient managing of large SoC multi-team projects with multiple sub-systems. For our purpose defined development process also enables accurate measurements of the design time and effort.

TABLE I
MILESTONES DEFINITIONS USED IN BY THE PROJECT.

Milestone	Description
HW0	Kick-off
HW1	Architecture and project setup
HW2	early design and design environments for RTL Design, verification, physical design tools
HW3	Qualified sub-system RTL
HW4	Qualified top level RTL
HW5	Sub-system DRC clean GDSII
HW6	Top level DRC clean GDSII
HW7	Tapeout
HW8	Wake-up and testing of fabricated IC samples
HW9	Application level demonstrators

The SoC development process we used is depicted in Fig. 1. The process is based on Reuse Methodology Manual formulated by Keating and Bricaud [8] already in 2002. The baseline of the process is waterfall, but includes iterations and speedup by modularity. The process model is updated based on SoC-Hub's cumulative expertise in the industry, research and teaching.

Fig. 1 illustrates the high-level view of the **project organisation, the development activities** and the timeline as proceeding **milestones**.

The process includes ten **development activities** from modeling (1) to application level demonstrators with fabricated chip (10) depicted as horizontal arrows. The lengths of the activities are illustrative and the point is to highlight parallelism of the different activities instead of strictly gated waterfall process. On this level we are not focusing on defining used tool flows, but in most of the cases one activity involves specific tool flow and domain expertise.

Milestones (HW0-HW9) reflect the readiness of the SoC from the beginning of the project to the demonstrators with the physical chip samples. The milestones are listed in Table I, but details beyond that are out of scope for this paper. The reason for milestones is that under multiple parallel activities with long tool run times, milestones can help to synchronize the work between the development activities, and acts as quality control points. For the project management, milestones give visibility to the schedule feasibility. As using milestones typically converge projects toward non-agile project setups, we enhanced the situation by taking milestones as guiding tools instead of as gates that would hinder the progress. In practise that means in example that in milestone checkpoint we recorded the target vs. actual and instead of schedule extension we analyzed the consequences adapted to changed situation.

Typically in large SoC project organizations different activities form often dedicated teams. That can cause local optimization and slower the communication because topics must cross organization and discipline specific terminology. Instead our **project organization** was mainly based on system composition so that each sub-system formed own team with dedicated lead person owning the responsibility. In addition to sub-system leads we had physical design lead and FPGA prototyping lead. The project management, lead architect and

verification lead roles were carried out by single person.

The SoC development included members from three different companies and the university. The engineering headcount was 24 persons, some of them part-time. The team expertise roughly followed pattern of 33%-33%-33% ratio between junior-senior-expert level knowledge.

Also project was run completely during the COVID-19 pandemic which limited us to work mainly remotely. Although the working conditions were exceptional, the remote work with IT-based communication reflects the situation of a large enterprise setup with multiple different sites in different locations.

IV. TEMPLATE BASED DESIGN METHODOLOGY

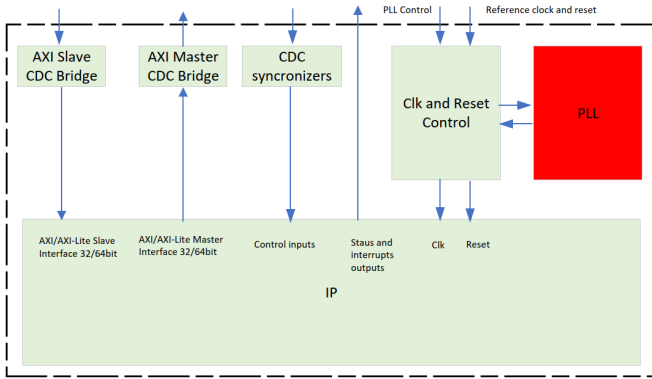


Fig. 2. Sub-system architecture template.

In the system design clear and efficient component composition is a key. The sub-system is our key entity when working with SoC architecture and integration. The sub-system architecture and the design methodology are based on well-defined template illustrated in Fig. 2. The key purpose of the sub-system is to create a fully independent clock domain including also a Phase-Locked Loop (PLL) based on-chip clock generator. The PLL is highly configurable mixed-signal design with the maximum output clock frequency of 3 Ghz. The PLL design was done as a part of this SoC project by the collaborating Intellectual Property (IP) provider company.

The independent clock tree of the sub-system enables hierarchical ASIC design flow which is independent from other sub-systems because the timing, the clock tree and the layout can be closed on the sub-system level. The hierarchical ASIC flow enables concurrent development of the sub-system from the RTL to the DRC cleaned GDSII layout. When synthesis, place and route, clock tree synthesis and static timing analysis are run with a moderate-sized sub-system instead of single chip level physical design flow, the tool run time and the environment complexity can be reduced. The shortened run time together with the enabled parallelism can significantly improve the iteration time of the ASIC flow.

The template also harmonizes the AXI interface parameters, the clock domain crossing (CDC), the clocking and the resetting implementations. The Clk and Reset Control module

takes care of clock gating, clock muxing, reset blanking and PLL control.

The AXI interface options include AXI4 and AXI4-Lite protocols with 32-bit and 64-bit data widths. For all of the provided alternatives, dedicated FIFO-based CDC bridge components were developed. The amount of these interfaces is a design decision and Fig. 2 illustrates only one master and one slave as an example. The used protocol of the IPs vary and the potentially needed protocol conversions are handled on a template level.

The harmonization provided by the sub-system template is a key for productive verification due to reuse and the correct-by-construct approach. The common architecture also helps in communication and flexible dynamic engineer resource allocation which is important especially for large-scale projects.

The hierarchical verification was the key verification strategy used in the project. The sub-systems were verified as separate entities and focused on functionality relevant on that level. The top-level verification focused only to integration of the sub-systems and system level functionality such as global address map. The verification methodologies included Universal Verification Methodology (UVM) and HW-SW co-simulation. The HW-SW co-simulation complicates the verification environment through SW cross-compilation, but on the other hand tests can be reused to between RTL simulation, FPGA prototyping and ASIC Sample testing.

We used FPGA platforms especially to validate IO devices toward real devices instead of inaccurate simulation models. The SoC boot process was also validated on the FPGA platforms. The FPGAs also gained from the modularity provided by the sub-systems because different configurations of the system could be created by including only needed sub-systems to the FPGA synthesis. That speed-ups the FPGA synthesis time and simplifies debugging.

V. SOC ARCHITECTURE

The SoC architecture is illustrated in Fig. 3. In this paper, we focus to the SoC top block which describes the chip content on a high level, but the figure includes also an example system-level setup for prototyping. The SoC is composed of six sub-systems that are based on a common sub-system architecture template. The sub-systems included are High Performance CPU (HPC), Medium Performance CPU (MPC), System Control (SysCtrl), Chip-to-chip (C2C), Artificial Intelligence (AI), Ethernet (Eth) and Digital Signal Processing (DSP). The interconnect components and minimal amount of configuration components are integrated to top-level outside sub-systems.

The strategy with the architecture has been to keep the CPU sub-systems as autonomous as possible. Due to that, each RISC-V based sub-system has their own JTAG and local SRAM for verification, debug and chip validation purposes.

The inter-processor communication is supported on architecture level by having a fully global memory and IO address map and global SW interrupts visible for all RISC-V cores.

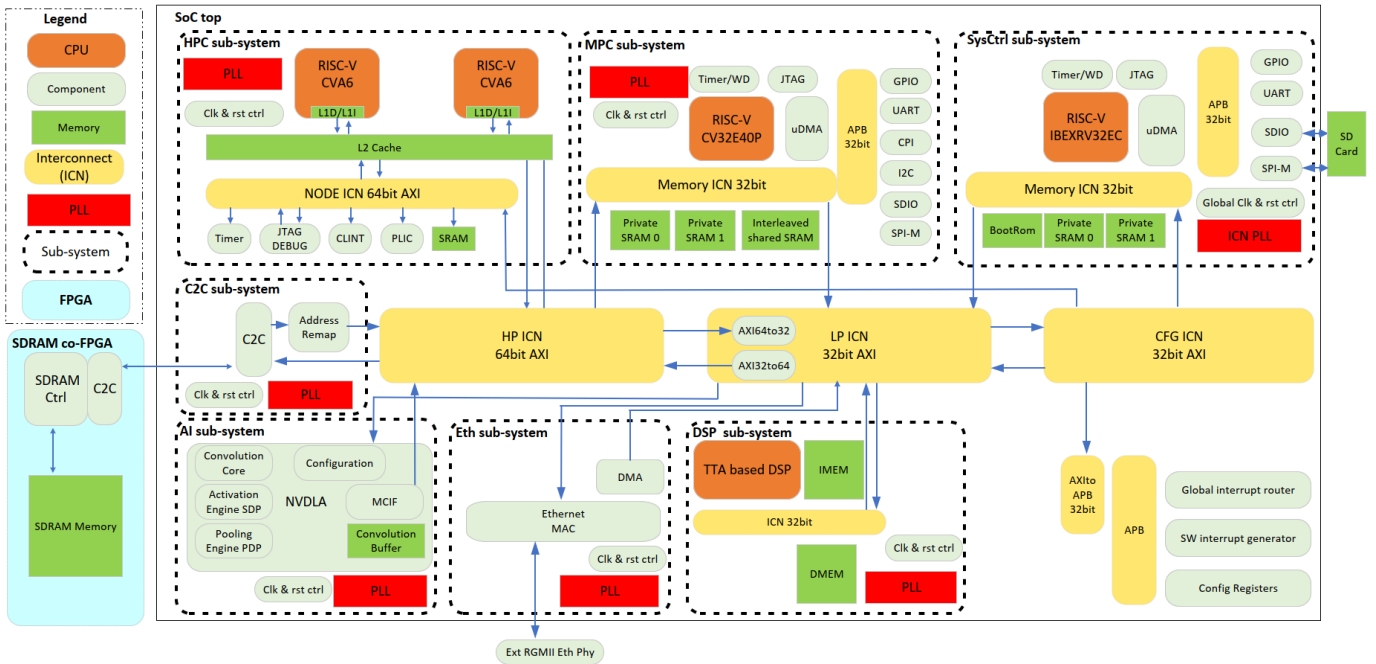


Fig. 3. Ballast SoC architecture and an example prototyping setup

A. HPC sub-system

The HPC sub-system reuses CVA6 (formerly known as Ariane) application class RISC-V core. The CVA6 implements RV64IMAC ISA extensions [9]. The memory system consist of 32 kB 8-way L1 data cache and 16 kB 4-way L1 instruction cache for both cores. Shared L2 cache is 256 kB 8-way and it was designed as a part of this SoC project.

The CVA6 support virtual memory and three privilege levels Machine (M), Supervisor (S) and User (U). These features make CVA6 Linux-capable RISC-V core.

B. MPC sub-system

The MPC sub-system is reused from pulp platform SoC design called Pulpissimo [2]. The MPC uses CV32E40P (formerly known as RI5CY) 4-stage pipeline in-order 32 bit RISC-V core with RV32IMC instruction set architecture (ISA), and the Xpulp custom extensions. The peripherals included QSPI, I2C, UART and parallel camera interface and IO data streams utilize peripheral to memory DMA engine (uDMA). SRAM based memories include two 32 KB Private banks and four 114 KB interleaved memory banks which enables four memory accesses in parallel. The interleaved memory is the largest shared memory in the SoC and can be utilized by other sub-systems as well.

C. SysCtrl sub-system

The SysCtrl is also based on Pulpissimo design, but has been heavily tailored toward minimal configuration. The 32-bit RISC-V CPU is 2-stage IBEX (formerly known as Zeroiscy), with RV32IMC ISA [10]. The SysCtrl is responsible for autonomous SoC boot and first stage of boot-loader is executed

from BootROM and perform application binary transfers from SD card to private SRAM. In addition to BootROM, JTAG based boot is supported. The SysCtrl operates on 30 Mhz clock driven by external crystal oscillator to make boot independent from PLL functionality. The ICN PLL in the SysCtrl is dedicated for top-level interconnect and due to small size of the SysCtrl sub-system, the design is place and routed as part of top level.

D. C2C sub-system

The C2C is a novel IP design developed as part of the Ballast SoC project. The C2C sub-system provides the high performance bi-directional memory mapped communication between two external chips while decreasing the amount of I/O ports needed by the AXI4 interface. The C2C has configurable address and data widths for the AXI4 interface and also configurable width of the physical layer. The implementation is based on asynchronous TX and RX FIFOs with two-way handshaking i.e. there is no clock dependency between chips nor between physical interface and the AXI4 interface. The C2C support AXI4 fixed and incremental burst types and multiple outstanding operations for performance. The C2C protocol is very light-weight without additional encoding, error detection nor error recovery which gives very good payload efficiency. The C2C protocol packet of one transaction consists of control (16bits), address (32bits) and data section dependent on AXI bus width and burst length ($BurstLength * NumberOfBytes * 8bits$).

The C2C instance in Ballast has 64-bit AXI4 interfaces and 16-bit wide physical layer per direction. Together with handshaking of the physical layer that leads to 36 pins.

With 200 Mhz physical layer clock we obtain uni-directional bandwidth of 3,2 Gbit/s.

The C2C has been prototyped with commercial FPGAs and thus C2C provides connectivity to high speed memories and other additional devices on connected FPGA as illustrated on Fig. 3.

E. AI sub-system

The AI sub-system is based on Nvidia Deep Learning accelerator NVDLA [11]. The NVDLA was configured to use 256 MAC units, 8-bit integer data format and 128 KB convolution buffer. The cross channel data processor (CDP) for local response normalization and reshape engine (RUBIK) as well as bridge DMA for accessing optional secondary memory were removed from the design for area optimization reasons and also because we analyzed the external memory bandwidth to limit factor instead of computation performance.

F. Eth sub-system

The Eth sub-system is an Ethernet Medium Access Layer (MAC) controller and conforms IEEE 802.3 with Reduced Gigabit Media Independent Interface (RGMI) for data which offers 1 Gbit/s Ethernet connectivity and Management Input Output (MDIO) interface for external Ethernet physical layer implementation. The Eth sub-system includes dedicated DMA engine coupled with TX and RX FIFOs.

G. DSP sub-system

The DSP sub-system is a statically scheduled VLIW-like core based on the Transport Triggered Architecture (TTA) [12] and its programming model follows an explicit datapath paradigm [13]. The sub-system is intended for image and audio processing tasks. The implementation includes a programmable dictionary based compression scheme to save instruction stream energy [14]. The memory hierarchy consists of two 64 KB compiler-controlled scratchpad memories for instructions and data separately. The implementation has been designed and generated with the TTA-based Co-design Environment (TCE) tools [15], which also provides C compiler support for the core.

H. Interconnect

The interconnect architecture is a combination of crossbar and bus topology. The hierarchical composition can be divided to sub-system and top-level interconnects. The sub-systems have autonomy for interconnect design, and the adaptation to the top-level interconnect is harmonized with the sub-system template architecture.

The top-level interconnect is composed of three fully connected crossbars (HP ICN, LP ICN and CFG ICN). The top-level interconnect topology has been the design choice between performance and area. The selected implementation allows parallel data streams between sub-system connected to same crossbar. The interconnect has been divided to 64-bit and 32-bit wide data regions and each crossbar forms own clock domain i.e. crossbars can operate on different frequencies

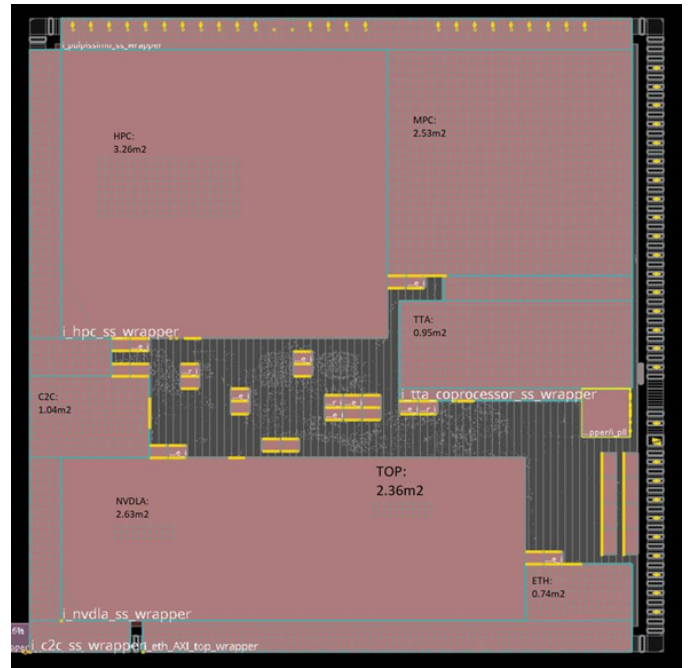


Fig. 4. Ballast SoC ASIC layout implemented on 22nm technology with total area of 15mm²

divided from ICN PLL. This clock tree and related CDC components has been omitted from figure.

VI. RESULTS

A. Power, Performance and Area

Power consumption and maximum operating frequency and area of the of sub-systems are recorded to Table III. Due to focus on fast execution we left most of the optimizations in performance, power and area for upcoming chip design iterations.

The reported frequency is the one used as constraint for physical design i.e. reflects maximum operational frequency for that sub-system. The area is effective area and utilization (area of cells / area occupied) varied between 40%-69% for sub-systems and the sub-systems with IOs (MPC, C2C and Eth) had lowest utilization. The layout with sub-system boundaries is visible in Fig. 4. The SysCtrl was flattened to top level. The layout also indicates some non-utilized area in the middle.

The measurements have been done with statistical estimation methods provided by commercial ASIC tool flow. Estimations have done for place and routed layout with 22nm technology library. The used corner settings in measurements has been 0,9V logic VDD, 1,8V for IOs and 20C temperature which is representing operation in normal lab condition. The switching activity for the all flip-flop outputs was set to 20% and activity was propagated through combinatorial logic. The PLL power consumption was also excluded. Gate-level simulation (GLS) based activity annotated to power measurement flow would give much more realistic switching activity, but GLS for such large sub-system included to our SoC are very

TABLE II
COMPARISON OF THE RELEVANT SOCS

Project	Technology	Max clock frequency	Area	Level of heterogeneity: no. of different computation architectures	Internal memory	External memory interface	Design effort	Other
Quentin [2]	22nm FD-SOI	938Mhz	2,3mm ²	1 (1x RV32IMFC-Xpulp extensions MCU)	520kB	Quad SPI 400Mbits, HyperBus DDR 800Mbit/s	NA	Pulp based single-core
Mr. Wolf [3]	40nm LP CMOS	450Mhz	10mm ²	2 (1x RV32IMF, 8x RV32IMF-Xpulp)	576kB	Quad SPI 400Mbits, HyperBus DDR 800Mbit/s	NA	Pulp based multi-core
Vega [4]	22nm FD-SOI	450Mhz	12mm ²	2 (10x RVC32IMF-Xpulp)	1,7MB SRAM + 4MB non-volatile MRAM	4xSPI, 1,6Gbit/s	NA	Pulp based multi-core with ML accelerators and two switchable power domain
Bailey et. al. [5]	16nm Fin-FET	410Mhz	25mm ²	3 (1x RV64GC, HWACHA vector accelerator, Streaming DSP accelerator chain)	10MB	Custom ADC with 3Ghz clock for DSP chain	14 000 engineering hours	RocketChip based multi-core
Gonzales et. al. [7]	22nm Fin-FET	961Mhz	16mm ²	4 (1x RV64GC, HWACHA vector accelerator, 1x RV64GC, GEMMINI DNN accelerator)	1MB L2-cache, 256KB scratchpad	low-speed SerDes (bandwidth not reported)	NA	Chipyard based multi-core with ML accelerators
SiFive U540 [1]	28nm	1,5GHz	30mm	2 (1x RV64IMAC, 4x RV64GC)	2MB	64bit DDR4 2400M transfers/s, QSPI	NA	RocketChip based commercial reference
This work	22nm	1,2Ghz	15mm ²	5 (2x RV64IMAC, 1x RV32IMC with Xpulp extensions, 1xRV32IMC, 1x DSP VLIW ASIP, 1x NVDLA 32 x 8bit MAC)	1MB	Chip2Chip 3,2Gbits/s, QSPI	12 months calendar time, 21 200 engineering hours	First chip of the SoC-Hub

TABLE III
POWER CONSUMPTION OF THE SUB-SYSTEMS

Sub-system	Frequency (MHz)	Power (mW)	Area (mm ²)
HPC	500	496,9	3,26
MPC	500	118,8	2,53
SysCtrl	30	8,0	Part of interconnect area
C2C	400 logic, 100Mhz inputs	27,9	1,04
AI	750	439,3	2,63
ETH	125	14,2	0,74
DSP	600	82,4	0,95
interconnect	Three clock domains: 1200/637/166	223,9mW	2,37

slow to simulate and debug. Our plan is to measure more accurate power measurements based on lab measurements with

actual fabricated ASICs.

When interpreting the results it needs to be understood that both operation conditions, statistical power estimation method with estimated switching activity affects results heavily.

B. Design effort

The 12 months was the calendar time for the HW1 to HW7 milestone, but more accurate results is obtained when engineering hours per design activity is reported together with amount of engineers participated at Table IV. As a summary, we report 21 200 scaled engineering hours which divides between RTL design and verification 45%, ASIC physical design 22%, FPGA 21%, IT infrastructure 4% and project management 8%.

Our project setting included many undergraduates and engineers with no former ASIC project background which is common for academic SoC work. The level of expertise has

TABLE IV
SCALED DESIGN EFFORT PER SOC DESIGN ACTIVITY

Activity	No. of engineers participated	Expertise level avg.	Scaled eng. hours	hours/total %
RTL design and verification	13	0,69	9 440	45%
ASIC physical design	6	0,75	4 720	22%
FPGA	5	0,65	4 440	21%
IT infrastructure	3	0,75	900	4%
Project management	4	0,875	1 700	8%
Total	24 Different engineers	0,74	21 680	100%

been taken into account with three different expertise level scaling factors for junior, senior and expert levels: junior 0,5 (undergraduate or no former SoC expertise), senior 0,75 (graduate or 1-6 years of SoC expertise) and expert 1,00 (over 6 years of SoC expertise). The average of scaling factor is reported per design activity. Scaled engineering hours = collected hours * scaling factor. Rather even distribution of expertise level is healthy and we achieved that by planning teams so that there is at least one more senior person who can support engineers on junior level.

The RTL design and the verification are combined because most of the engineers in that category participated in both. However, we had two dedicated verification engineers as well. Also, many engineers participated to multiple activities and thus total amount of engineers differs from the sum of engineers at Table IV.

The mixed signal PLL design was excluded from these results. The PLL design was performed by the collaborating company during the project, but we do not have the same level of visibility to the hours spent on that work.

The work with tool, IP and ASIC technology licensing was done during the project also. However, we excluded agreements related work from design effort calculation while that work can be reused by multiple projects. ASIC technology licensing took a long time and we got the full ASIC technology access only four months before the tapeout.

The IT infrastructure work included setup and maintenance of development servers, version control system (gitlab), Continuous integration pipelines for test automation and tool installations.

To enable comparison of the design effort to related work we need to take the SoC complexity in to account. We selected the level of heterogeneity as the key attribute to indicate the complexity. The level of heterogeneity at Table. II list amount of different architectures. Amount of parallel cores e.g. 2x RV64IMAC is given as additional information although parallelism not affecting this measure. We felt that the amount of logic gates, area or number of parallel cores would not work

as good measure of complexity because the amount of such homogeneous structures can be increased rather easily through design parameters such as memory size or number of parallel cores. As a results, reported SoCs have one to five different computation engines and our work having largest complexity measured on given method.

The summary of the results comparison has been recorded to Table II. The results indicate that complexity of our work is larger than any of the related work and design effort results with similar accuracy is not reported because activities included to design effort are not specified.

C. Discussion of the results

From the design effort results we notice that outside IT infrastructure and project management the engineering effort is quite equally divided if we assume that RTL design and verification is shared with 50%-50% ratio. The sub-system development consisted of reused open source design and novel designs developed as part of the project. The experience with open source reuse was mixed. With reused RISC-V CPU cores we didn't found severe issues. The problems raised when we need to change something on reused larger entity. The lack of documentation and poor code quality was common issues encountered.

Despite large complexity of the design and organization of multiple teams project management took surprisingly little amount of time. That might be explained by the responsibilities of the lead engineers. Leads engineers and ASIC project manager participated to actual development as well. Due to that some project management work might be reported as development work.

Also the component composition based organization instead of the development activity based organization helped on common understanding of the project matters and provided flexibility. During the project many engineers worked on multiple different development activities instead on focusing to single subject only. We argue that such common understanding and flexibility is necessity for fast execution of large SoC projects, although we understand that domain knowledge is essential for many activities like ASIC physical design and due to that flexibility possibilities are limited.

The FPGA related work too surprisingly long effort although FPGAs as typically regarded as platform for fast prototyping. To understand this, it is good to differentiate FPGA based prototype design and FPGA based ASIC design emulation. FPGA based prototypes are typically build on top of FPGA development platforms i.e. the design is FPGA friendly by nature. The ASIC emulation on FPGA is instead opposite. In the emulation, ASIC implementation is a primary target and FPGA work needs to adapt used design structures. This adaptation causes lot of work because technology specific components such as SRAM memories and clock tree components needs to be changed.

The PPA optimizations were not primary target for the this chip and we focused to creation of reusable template based SoC flow instead. Lack of optimizations naturally speeds up

development while ASIC physical design iterations can be kept limited. The PPA optimizations remains an important topic and target is address that in upcoming chip versions. We believe that due to complexity of the SoCs both creation of new architecture and optimizations are hard to achieve in one chip iteration.

As a summary we have demonstrated that flexible organization and template based SoC architecture can lead to fast execution of the SoC project in academic driven project including industry collaborators.

VII. CONCLUSIONS

This paper presented SoC design effort results of one year off calendar time and 21 200 engineering hours with accuracy not presented earlier. To amplify relevancy off the design effort results, a large and complex 15mm² SoC with six template-based sub-systems were implemented on 22nm ASIC technology. The complexity of SoCs compared was judged based on the number of different computation architectures. The SoC project team included 24 engineers and level of the SoC knowledge was taken into account when reporting engineering hours.

REFERENCES

- [1] Y. Lee and A. Waterman, "Managing chip design complexity in the domain-specific soc era," in *2020 IEEE Symposium on VLSI Circuits*. IEEE, 2020, pp. 1–2.
- [2] P. D. Schiavone, D. Rossi, A. Pullini, A. Di Mauro, F. Conti, and L. Benini, "Quentin: an ultra-low-power pulchissimo soc in 22nm fdx," in *2018 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*. IEEE, 2018, pp. 1–3.
- [3] A. Pullini, D. Rossi, I. Loi, G. Tagliavini, and L. Benini, "Mr. wolf: An energy-precision scalable parallel ultra low power soc for iot edge processing," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 7, pp. 1970–1981, 2019.
- [4] D. Rossi, F. Conti, M. Eggiman, S. Mach, A. Di Mauro, M. Guermendi, G. Tagliavini, A. Pullini, I. Loi, J. Chen *et al.*, "4.4 a 1.3 tops/w@ 32gops fully integrated 10-core soc for iot end-nodes with 1.7 μ w cognitive wake-up from mram-based state-retentive sleep mode," in *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64. IEEE, 2021, pp. 60–62.
- [5] S. Bailey, P. Rigge, J. Han, R. Lin, E. Y. Chang, H. Mao, Z. Wang, C. Markley, A. M. Izraelevitz, A. Wang *et al.*, "A mixed-signal risc-v signal analysis soc generator with a 16-nm finfet instance," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 10, pp. 2786–2801, 2019.
- [6] D. Petrisko, F. Gilani, M. Wyse, T. Jung, S. Davidson, P. Gao, C. Zhao, Z. Azad, S. Canakci, B. Veluri *et al.*, "Blackparrot: An agile open source risc-v multicore for accelerator socs," *IEEE Micro*, 2020.
- [7] A. Gonzalez, J. Zhao, B. Korpan, H. Genc, C. Schmidt, J. Wright, A. Biswas, A. Amid, F. Sheikh, A. Sorokin *et al.*, "A 16mm 2 106.1 gops/w heterogeneous risc-v multi-core multi-accelerator soc in low-power 22nm finfet," in *ESSCIRC 2021-IEEE 47th European Solid State Circuits Conference (ESSCIRC)*. IEEE, 2021, pp. 259–262.
- [8] M. Keating and P. Bricaud, *Reuse Methodology Manual for System-on-a-Chip Designs: For System-on-a-chip Designs*. Springer Science & Business Media, 2002.
- [9] 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 Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 11, pp. 2629–2640, Nov 2019.
- [10] P. D. Schiavone, F. Conti, D. Rossi, M. Gautschi, A. Pullini, E. Flamand, and L. Benini, "Slow and steady wins the race? a comparison of ultra-low-power risc-v cores for internet-of-things applications," in *2017 27th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. IEEE, 2017, pp. 1–8.
- [11] "Nvidia deep learning accelerator (nvdlia)," <http://nvdla.org/index.html>, accessed: 2022-04-08.
- [12] H. Corporaal, *Microprocessor Architectures: from VLIW to TTA*. John Wiley & Sons, Inc., 1997.
- [13] P. Jääskeläinen, H. Kultala, T. Viitanen, and J. Takala, "Code density and energy efficiency of exposed datapath architectures," *Journal of Signal Processing Systems*, vol. 80, pp. 49–64, Jul 2014.
- [14] J. Multanen, K. Hepola, and P. Jääskeläinen, "Programmable dictionary code compression for instruction stream energy efficiency," in *2020 IEEE 38th International Conference on Computer Design (ICCD)*, 2020, pp. 356–363.
- [15] P. Jääskeläinen, T. Viitanen, J. Takala, and H. Berg, "HW/SW co-design toolset for customization of exposed datapath processors," pp. 147–164, 2017.