SoC Studio: A User-Centric Framework for Custom System-on-Chip Design, Emulation, and AI Integration

Shayan Hassan Baig[1], Shahzaib Kashif[1], Ali Ahmed[2] and Farhan Ahmed Karim[1]

[1]Department of Computer Science, Usman Institute of Technology (UIT)
[2]Department of Electrical Engineering, Usman Institute of Technology (UIT)

Abstract

Modern System-on-Chip (SoC) design demands flexible tools to bridge the gap between conceptualization and implementation. This paper introduces SoC Studio, an innovative application enabling users to configure, simulate, and emulate custom SoC designs through an intuitive interface. SoC Studio automates customized SoC Register-Transfer Level (RTL) code generation, simulator generation, FPGA bitstream compilation, and simulation workflows. Unique to SoC Studio is its support for both bare-metal software execution and ONNX-based AI model deployment on accelerator-enabled designs on the cloud. The framework further provides FPGA emulation on cloud or local hardware, alongside real-time power-performance-area (PPA) metrics and resource utilization analysis. By democratizing SoC development, SoC Studio reduces design cycles and empowers non-experts to prototype heterogeneous architectures efficiently.

Introduction

Slowing of Moore's Law has led computer architects to focus on making architectural changes to improve computational performance and energy efficiency, interconnecting electronic components such as processors, memory modules and peripheral interfaces onto a single chip.[5] Nevertheless, SoC design has historically been an intricate and labor-intensive task, requiring technical expertise and restricting adoption to a niche cohort of individuals and organizations.[1]

Recent innovations in SoC design tools have started to mitigate these challenges. A notable example is the open-source Chipyard platform[2], which is an integrated framework for configuring, generating, and simulating SoCs. Despite its advancements, the Chipyard platform remains somewhat reliant on users' technical proficiency, as navigating and leveraging its full suite of capabilities demands a foundational understanding of the SoC design principles. This highlights the ongoing need to bridge the gap between cutting-edge tools and user accessibility.

To bridge this accessibility gap, we present SoC Studio, a browser-based graphical interface engineered atop the Chipyard framework. It democratizes RISC-V[3]-centric SoC development by accommodating a spectrum of technical proficiency users, from novices to experts. Through its visual workflow and abstraction of low-level complexities, SoC Studio expedites the creation of RISC-V-compliant SoC architectures, broadening participation in semiconductor innovation. This paradigm shift holds transformative potential for academia, startups, and non-specialist practitioners alike.

SoC Studio enables users to generate RTL code[4], generate (functionally and cycle accurate) simulators, produce FPGA-ready bitstreams for emulation, and conduct simulations to evaluate design functionality. For simulation, users can deploy bare-metal programs for foundational testing or leverage ONNX-based AI models to validate accelerator performance when integrated into the design. FPGA emulation is supported across both cloud-based infrastructure and local hardware boards, offering flexibility in prototyping environments. Additionally, the platform provides detailed PPA analytics and insights into FPGA resource consumption including lookup tables, block RAM, and DSP slices, equipping users with actionable data to refine and optimize their SoC implementations.

In this extended abstract, we present a synopsis of SoC Studio, dissecting its architectural structure, functional capacities, and transformative implications for the SoC design ecosystem.

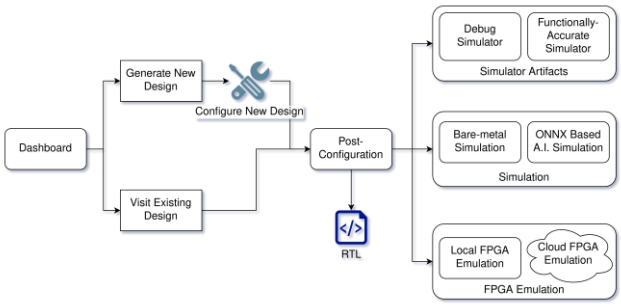

Figure 1 : SoC Studio Workflow

Core Elements

This section elucidates the platform's core subsystems.

User-Centric SoC Configuration Interface

SoC Studio offers a user-friendly web-based GUI in which a structured layout is provided for configuring SoC components, allowing selection and modification of core types, caches, memory controllers, interconnects, and accelerators through parameterized configuration of each element. The GUI approach helps users understand and visualize their design while avoiding errors that may arise from command line or file-based approaches in tools like Chipyard and X-HEEP.[6]

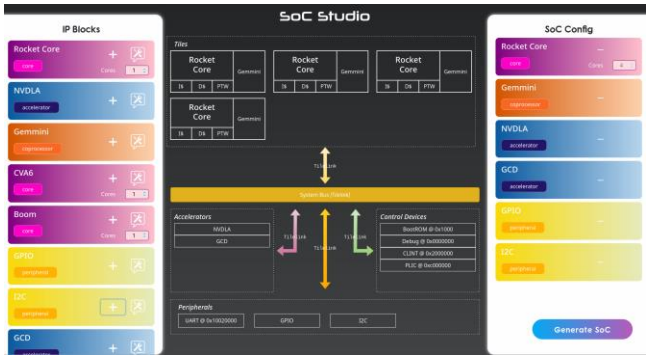

Figure 2 : SoC Designer of SoC Studio Web App

Simulator Artifacts

The RTL artifact is the SystemVerilog design for the SoC the user configured. The framework also generates a cycle-accurate simulator, called the Debug Simulator, which enables generation of waveforms of bare-metal programs and software running on the custom SoC, as well as a functional-accurate simulator which provides only the post-simulation terminal output using UART in the SoC.

Simulation

SoC Studio allows two types of simulations.

1. Bare metal
2. ONNX-based A.I. inference

In bare metal simulation, users write their own C program to run on the configured SoC, while ONNX-based A.I. inferencing simulation is possible if the design has an A.I. accelerator configured in it. The user can upload custom or select pre-trained ONNX-based A.I. models which are then run on a RISC-V based ONNX runtime for the simulation.

FPGA Emulation

SoC Studio offers FPGA emulation, enabling users to deploy designs on cloud-based FPGA instances (e.g., AWS F1) or local boards (e.g., Xilinx Zynq, Intel Altera) for rapid prototyping. Post-emulation, the platform generates comprehensive PPA reports and detailed FPGA resource utilization metrics, including lookup tables (LUTs), flip-flops (FFs), and block RAM consumption.

Custom User IP Integration

SoC Studio also supports the integration of custom intellectual property (IP) blocks into the SoC architecture. Users can dynamically augment the Memory-Mapped I/O (MMIO) address space with custom or pre-validated IPs or attach as a host (core) or a coprocessor, integrating them into their designs through guided workflows. The platform further accelerates third-party IP adoption via automated blackbox abstraction, enabling plug-and-play compatibility for externally developed RTL modules whether cores, accelerators, coprocessors, or peripherals. This process is protocol-agnostic, accommodating user-specified interconnect standards such as Tilelink and AMBA through configurable adapter connection and generation.

Conclusion and Future Work

SoC Studio represents a significant step forward in democratizing SoC design by providing an intuitive, automated framework for configuring, simulating, and emulating custom SoCs. Its unique features, such as RTL code generation, ONNX-based AI model deployment, and real-time PPA analysis, empower both experts and non-experts to prototype heterogeneous architectures. By reducing design cycles and enabling cloud-based FPGA emulation, SoC Studio bridges the gap between conceptualization and implementation. Looking ahead, future work will focus on integrating an **RTL-to-GDS-II flow** to support ASIC implementation, extending the tool's capabilities to cover the entire design process from RTL to physical layout. Additionally, enhancements in AI/ML support, multi-FPGA emulation, and advanced PPA optimization will further solidify SoC Studio as a comprehensive solution for modern SoC development.

References

[1] (Muhammad Shahzaib, Shahzaib Kashif, Talha Ahmed, Farhan Ahmed Karim, Hadir Khan, Usman Zain),(SoC-Now: An Open-Source Web based RISC-V SoC Generator), Article No. (6), Workshop on Open-Source EDA Technology (WOSET), 2022.

[2] Amid, Alon, David Biancolin, Abraham Gonzalez, Daniel Grubb, Sagar Karandikar, Harrison Liew, Albert Magyar et al. "Chipyard: Integrated design, simulation, and implementation framework for custom socs." IEEE Micro 40, no. 4 (2020): 10-21.

[3] Waterman A, Lee Y, Patterson D, Asanovic K, level Isa VI, Waterman A, Lee Y, Patterson D. The RISC-V instruction set manual. Volume I: User-Level ISA', version. 2014 Apr;2.

[4] Kashif, Shahzaib, Talha Ahmed, Mahnoor Ismail, and Farhan Ahmed Karim. "Chipshop: A cloud based gui for accelerating soc design."

[5] Patterson, David A., and John L. Hennessy. Computer organization and Design. Morgan Kaufmann,, 1994.

[6] Machetti, Simone, et al. "X-heep: An open-source, configurable and extendible risc-v microcontroller for the exploration of ultra-low-power edge accelerators." arXiv preprint arXiv:2401.05548 (2024).