PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

CIE Summer Workshop 2023,

June 2023 - July 2023

Report on

An Exploration of Open Source VLSI Tools for Simulation (Functional & Formal) and Synthesis (ASIC)

Submitted by,

1. Nandeesha Swamy - PES1UG21EC159

(pes1202102103@pesu.pes.edu)

2. Pooja R - PES1UG21EC182

(pes1202101997@pesu.pes.edu)

Mentored by,

1. Sai Govardhan - PES1UG19EC255 (saigovardhanmc@pesu.pes.edu)

Acknowledgements,

Sathya Prasad - Director, CIE

Madhukar Narasimha - Program Manager, CIE

Table of Contents:

1. Introduction

This project report provides a comprehensive overview of the enlightening 8-week journey through the CIE summer workshop. Throughout this program, we embarked on a captivating exploration of various open-source VLSI tools, encompassing both simulation (functional and formal) and synthesis. Notably, we delved into the realm of Verilator for simulation, Yosys for synthesis, and Symbiyosys for verification, gaining invaluable insights into their capabilities and functionalities.

Recognizing the indispensability of open-source tools in the field of VLSI, we further endeavored to equip VLSI enthusiasts with the means to expand their knowledge without constraints. To this end, we meticulously crafted a user-friendly installation and tool guide manual, empowering individuals to harness the full potential of these open-source tools and delve deeper into the realm of VLSI.

Moreover, our practical application of these tools extended beyond theory, as we demonstrated their prowess by implementing Adder, Mac, and Sorter designs. By immersing ourselves in the intricacies of these tools, we developed a profound understanding of the toolflow, gaining insights into their seamless integration within the VLSI design process.

While our focus remained on introducing essential and fundamental tools, we also embarked on a continuous exploration of additional tools. Our aim is to curate a comprehensive and cohesive package that encompasses all the necessary tools for an optimal VLSI tool flow. By undertaking this endeavor, we strive to streamline the VLSI design process and empower future VLSI enthusiasts with a holistic toolkit.

Overall, this project review encapsulates our journey through the CIE summer workshop, showcasing our deepened understanding of open-source VLSI tools, our commitment to knowledge dissemination through the creation of a user-friendly manual, and our dedication to expanding the scope of available tools for the VLSI community.

2. Problem Statement

Our project aims to enable VLSI enthusiasts to expand their knowledge without the burden of expensive licensed tools like Cadence, Mentor Graphics, and Synopsis. Through the exploration of open-source VLSI tools and the creation of a user-friendly installation and toolflow manual, we seek to provide a cost-effective alternative for learners. Open-source tools, being free and functional, offer a practical solution to perform basic operations efficiently. By reaching as many individuals as possible with our comprehensive manual, we hope to empower aspiring learners to pursue their passion for VLSI without financial constraints, fostering a thriving community of innovative contributors in the field.

3. EDA Industry Overview

The Electronic Design Automation (EDA) industry continues to be a vital part of the semiconductor and electronics manufacturing sector. It plays a crucial role in accelerating the design and development of complex integrated circuits (ICs) and other electronic systems, enabling innovations in various industries like consumer electronics, automotive, telecommunications, and more.

The EDA industry is characterized by a handful of major players, such as Cadence Design Systems, Synopsys, and Siemens EDA (formerly Mentor Graphics), along with smaller companies specializing in specific domains or offering niche tools. The market is competitive and subject to continuous innovation, as the demand for faster, smaller, and more power-efficient electronic devices continues to rise.

Our focus is on "Design and Verification Tools" in the EDA industry. These tools are at the core of the electronic design automation process and are essential for designing and verifying complex integrated circuits (ICs) and electronic systems. Let's break down this point further:

1. **Design Tools:** EDA companies offer a variety of design tools that enable engineers to create electronic circuits, systems, and components. These tools provide an environment where engineers can specify the functionality and behavior of the electronic design using hardware description languages (HDLs) such as Verilog and VHDL. Design tools allow engineers to describe the logical operation of the circuits at a high level of abstraction.
    - Hardware Description Languages (HDLs): HDLs are specialized programming languages used to model and describe digital systems.
    - Design Entry: Design entry tools allow engineers to enter their designs using HDLs or graphical user interfaces (GUIs) to create schematic representations of the circuits.

2. **Verification Tools:** After the design is created, it needs to be thoroughly tested and verified to ensure that it functions correctly and meets the desired specifications. Verification tools in the EDA industry play a critical role in this process.
    - Functional Verification: Functional verification involves testing the design's behavior under different scenarios and input conditions.
    - Simulation: Simulation tools allow engineers to run tests on the design virtually, simulating its behavior in a software environment.
    - Formal Verification: Formal verification techniques use mathematical methods to prove that the design meets specific requirements and behaves correctly under all possible conditions.
    - Hardware Emulation: Hardware emulation involves creating a physical prototype (emulator) of the design, enabling engineers to test the design at near-real-time speeds.
    - Accelerated Verification: Some verification tools use specialized hardware (e.g., Field-Programmable Gate Arrays - FPGAs) to accelerate the verification process, making it faster and more efficient.

These design and verification tools in the EDA industry significantly contribute to reducing design cycle times, increasing design reliability, and lowering development costs. By enabling engineers to work at higher levels of abstraction and providing robust verification capabilities, EDA tools play a crucial role in driving innovation and advancements in the electronics industry.

4. Open Source VLSI Industry

The open-source VLSI (Very Large Scale Integration) industry is a growing and emerging area with significant potential. The primary motivation behind the open-source VLSI movement is to democratize chip design and lower the barrier of entry for individuals, startups, and smaller companies. Traditional VLSI design tools and licenses from proprietary vendors can be prohibitively expensive, making it challenging for newcomers to participate in the chip design market.

Open-source VLSI tools and methodologies have gained traction in recent years due to several factors:

1. **Cost Reduction:** Traditional VLSI design tools and licenses from proprietary vendors can be expensive, making it difficult for smaller companies and startups to enter the market. Open-source tools provide a cost-effective alternative.

2. **Community Collaboration:** Open-source projects encourage collaboration and knowledge sharing among engineers and designers.

3. **Customization and Flexibility:** Open-source tools allow users to modify and customize the software to meet their specific needs.

4. **Education and Research:** Open-source VLSI tools are often used in academic institutions and research labs, providing students and researchers access to industry-standard design tools.

5. **ASIC and FPGA Design:** Open-source VLSI tools and methodologies are used for designing both ASICs (Application-Specific Integrated Circuits) and FPGAs (Field-Programmable Gate Arrays). While ASICs are custom chips designed for specific applications, FPGAs offer programmable logic that can be reconfigured for different tasks.

5. Open Source EDA Tools Explored

During our project, we delved into the exploration of three crucial open-source tools: Verilator for simulation, Yosys for synthesis, and SymbiYosys for verification. To ensure seamless functionality, we identified and integrated some essential prerequisites, including iVerilog and Gtkwave.

With a comprehensive understanding of these tools' capabilities, we successfully developed a user-friendly manual. This manual serves as a valuable resource, offering step-by-step installation guidelines and detailed demonstrations of each tool's functionalities. By providing this accessible documentation, we aimed to empower fellow enthusiasts to harness the full potential of these tools without unnecessary complexities.

Putting theory into practice, we implemented various designs, such as MAC (Multiplier-Accumulator), Adder Tree, and Sorter, utilizing the open-source tools.

1. **Verilator:** Verilator is a high-speed open-source HDL simulator widely used in digital design and verification. It employs "compile-time elaboration," converting Verilog/SystemVerilog code into optimized C++/SystemC code, enabling fast simulation of large and complex designs. It's a popular choice for efficient and accurate hardware design verification.

2. **Yosys:** Yosys is a versatile open-source synthesis tool widely used for converting RTL designs into optimized gate-level representations. It supports multiple hardware description languages, including Verilog and SystemVerilog, and offers

advanced algorithms for technology-independent synthesis. With its efficient netlist generation and FPGA/ASIC support, Yosys has become an essential tool in the VLSI community.

3. **SymbiYosys:** SymbiYosys is an advanced open-source formal verification tool designed for RTL designs. It utilizes Yosys synthesis capabilities and supports SystemVerilog Assertions (SVA) for checking design properties, finding bugs, and ensuring correctness. Its automation enhances the VLSI verification process, making SymbiYosys a valuable asset for designers seeking robust and bug-free hardware designs.

4. **Icarus Verilog or iVerilog:** iVerilog is a widely used open-source Verilog simulator for digital circuit simulation and verification. It offers a command-line interface, making it efficient and popular among VLSI designers for testing complex hardware designs written in Verilog HDL.

5. **GTKWave:** It is an open-source waveform viewer for digital simulation analysis and debugging. It provides a user-friendly interface to visualize simulation results in various formats.

6. Our Design Implementations (MAC, Adder, Sorter)

We have translated our knowledge of open-source EDA (Electronic Design Automation) tools into practical application by undertaking various designs on these platforms. The designs we have implemented include MAC, Adder tree and Sorter.

1. **MAC:** In VLSI, a MAC (Multiplier-Accumulator) is a fundamental building block used extensively in digital signal processing applications. It combines a multiplier and an accumulator into a single unit, performing multiplication and addition operations simultaneously. This efficient design reduces hardware complexity and execution time, making MAC units crucial in various applications

like audio and image processing, communication systems, and digital filters. Their compact nature and high performance make MAC units essential components in modern VLSI designs.

2. **Adder Tree:** An adder tree is a widely used digital circuit structure in VLSI design for fast addition of multiple numbers. It consists of cascaded adders, reducing the inputs by half at each stage until a single output is obtained. This parallel processing makes adder trees ideal for arithmetic-intensive tasks, contributing to enhanced performance and reduced power consumption in modern VLSI architectures.

3. **Sorter:** A bitonic sorter is a powerful parallel sorting algorithm utilized in VLSI and digital design for efficient data arrangement. It capitalizes on the "bitonicity" property, allowing it to perform comparisons and swaps on multiple sublists concurrently, resulting in faster sorting in parallel computing environments. Its ability to handle large datasets and offer scalability makes the bitonic sorter an essential tool for optimizing sorting tasks in diverse applications.

4. **FIFO:** A FIFO (First-In-First-Out) is a key data storage structure in VLSI and digital design. It manages data on a "first-in-first-out" basis, maintaining the order of data arrival. FIFOs are essential for managing data flow between different clock domains or components, ensuring efficient and reliable communication in digital systems.

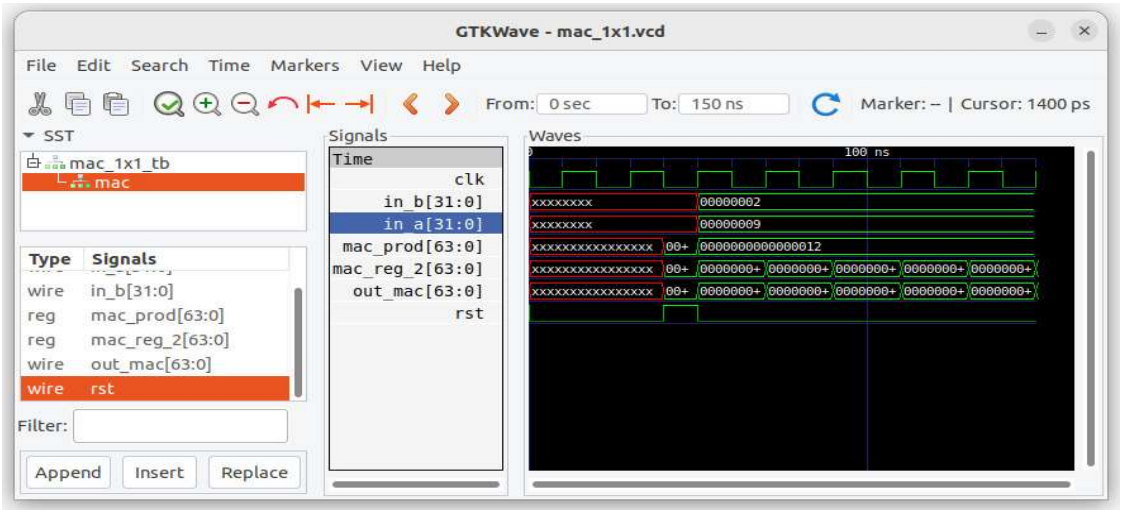
7. Demonstration on Verilator, Symbiyosys and Yosys

We have implemented the designs mentioned above utilizing these open source tools.

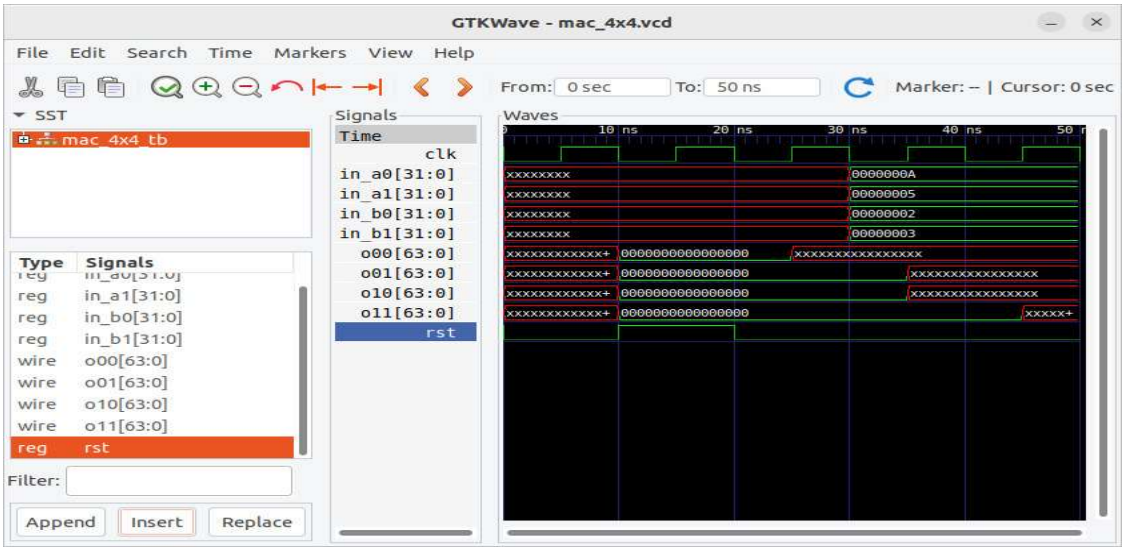
1. MAC

- Verilator:

mac_1x1

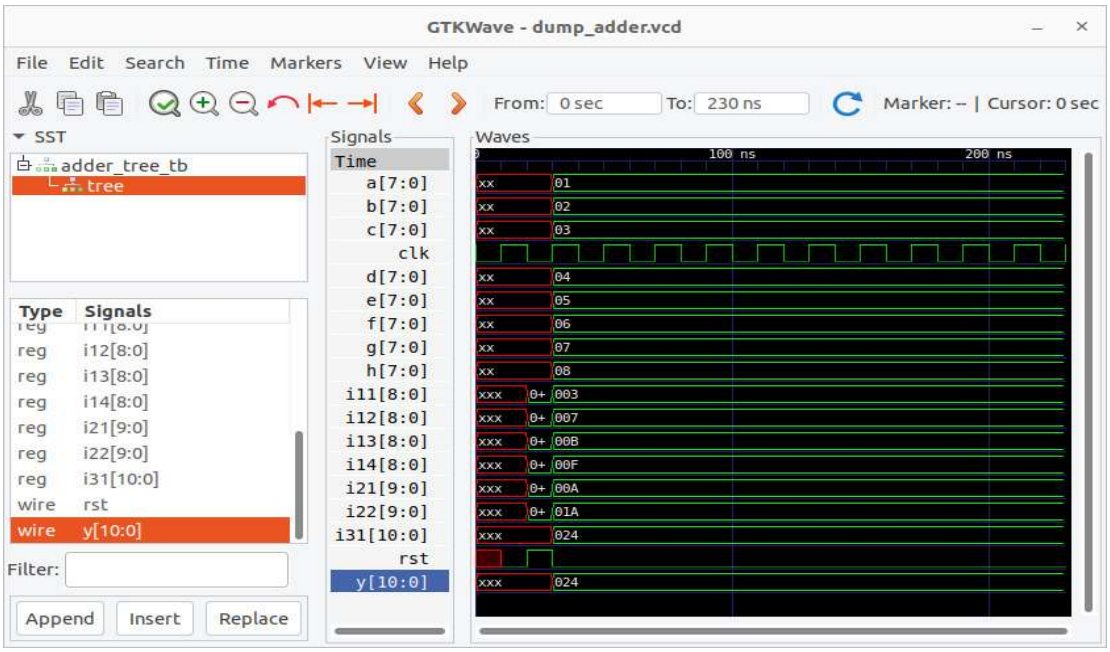


mac_4x4

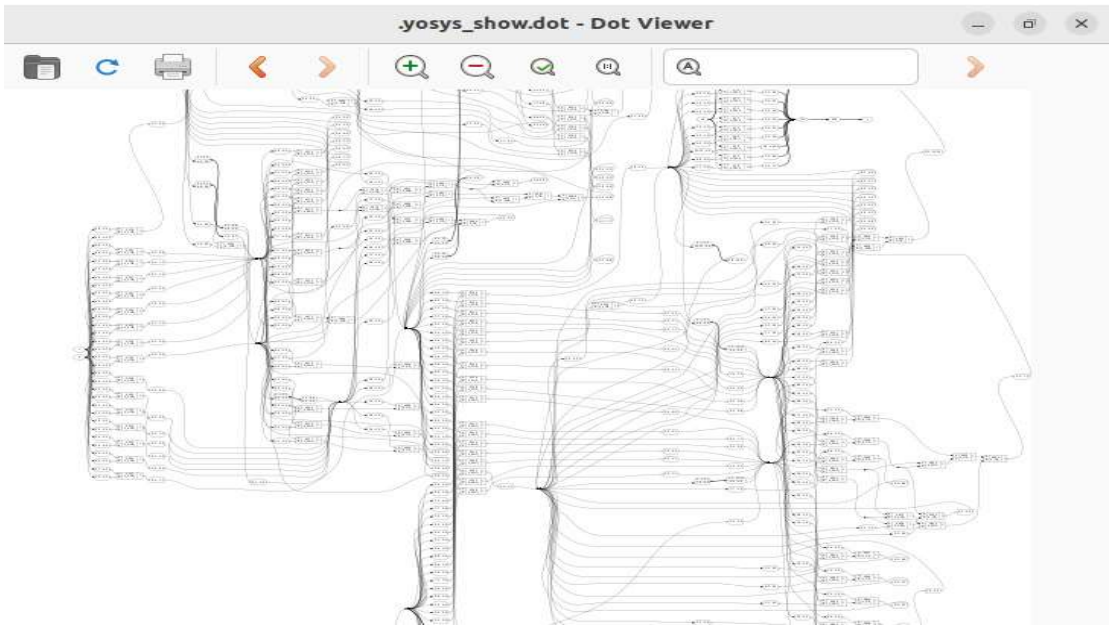


2. Adder tree

- Verilator



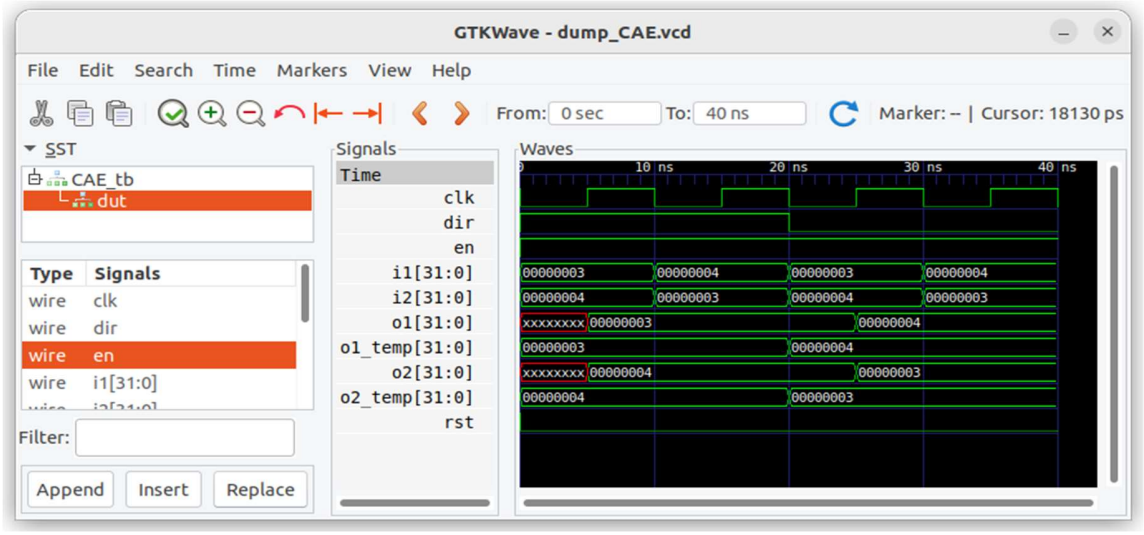
- Yosys



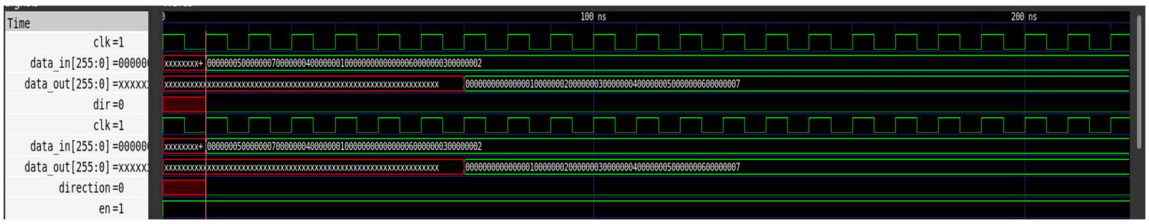
3. Sorter

- Verilator:

  CAE

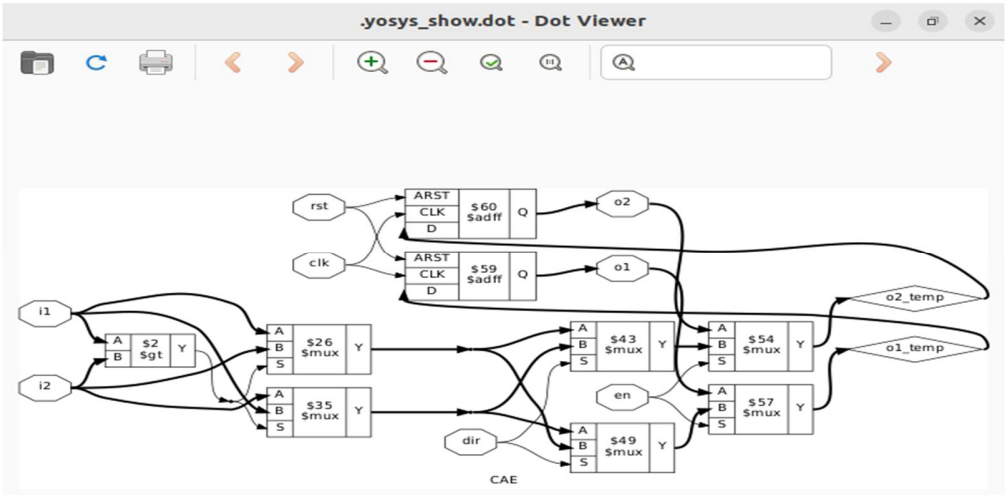


  Bitonic Sorter

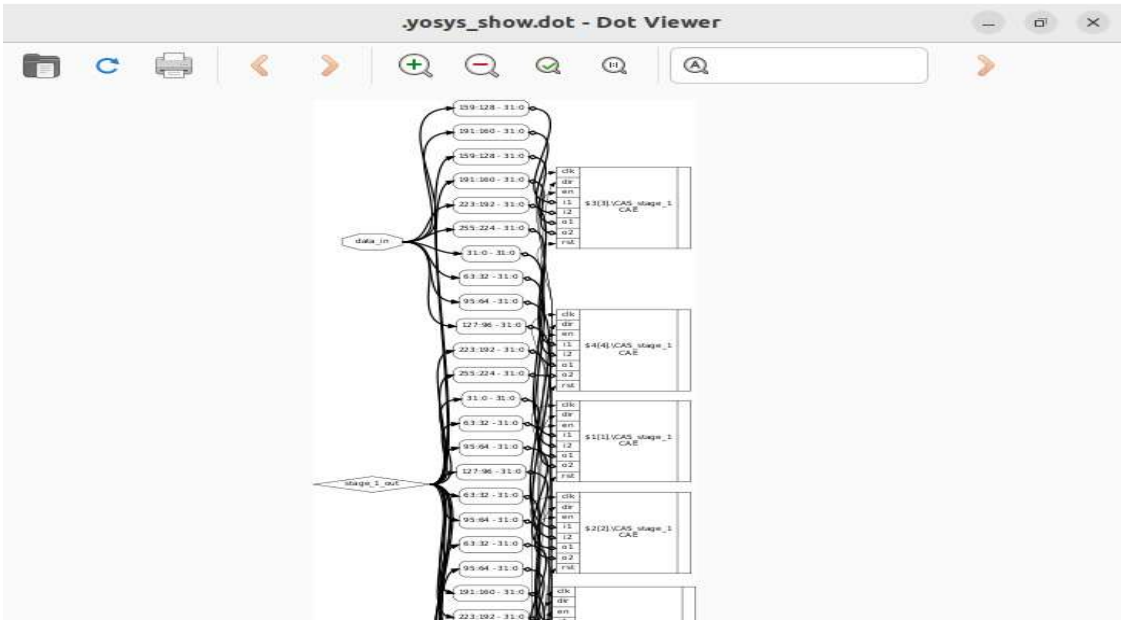


- Yosys:

  CAE



Bitonic Sorter

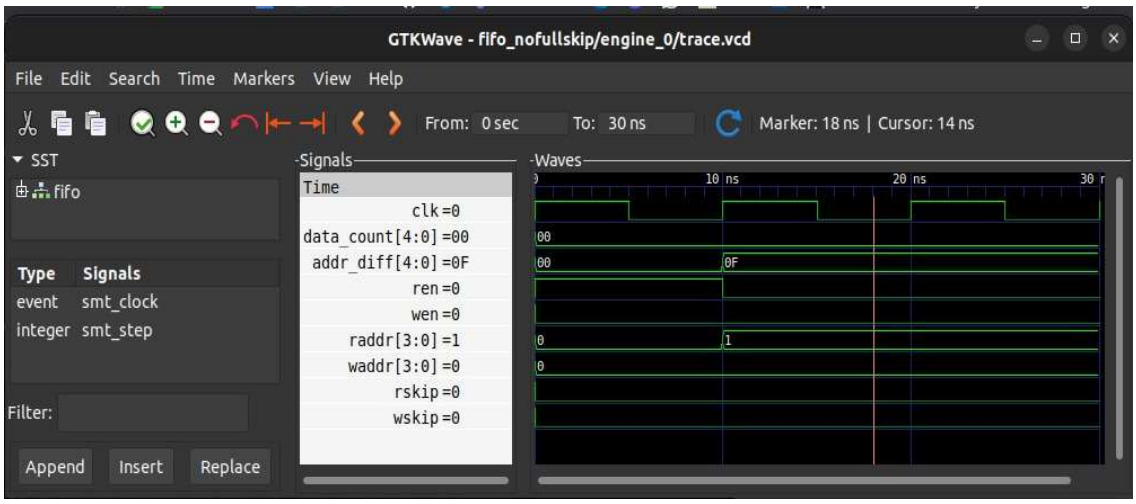


4. FIFO

- SymbiYosys

```
pooja@pooja-VirtualBox: ~/Desktop/Riscoveri/yosys/sby/do...
SBY 22:52:41 [fifo_nofullskip] engine_0.basecase: ##   0:00:00  Writing trace t
o Verilog testbench: engine_0/trace_tb.v
SBY 22:52:41 [fifo_nofullskip] engine_0.induction: ##   0:00:00  Trying inducti
on in step 16..
SBY 22:52:41 [fifo_nofullskip] engine_0.basecase: ##   0:00:00  Writing trace t
o constraints file: engine_0/trace.smtc
SBY 22:52:41 [fifo_nofullskip] engine_0.basecase: ##   0:00:00  Writing trace t
o Yosys witness file: engine_0/trace.yw
SBY 22:52:41 [fifo_nofullskip] engine_0.induction: ##   0:00:00  Trying inducti
on in step 15..
SBY 22:52:41 [fifo_nofullskip] engine_0.basecase: ##   0:00:00  Status: failed
SBY 22:52:41 [fifo_nofullskip] engine_0.basecase: finished (returncode=1)
SBY 22:52:41 [fifo_nofullskip] engine_0.basecase: Status returned by engine for
 basecase: FAIL
SBY 22:52:41 [fifo_nofullskip] engine_0.induction: terminating process
SBY 22:52:41 [fifo_nofullskip] summary: Elapsed clock time [H:MM:SS (secs)]: 0:
00:00 (0)
SBY 22:52:41 [fifo_nofullskip] summary: Elapsed process time [H:MM:SS (secs)]:
0:00:00 (0)
SBY 22:52:41 [fifo_nofullskip] summary: engine_0 (smtbmc boolector) returned FA
IL for basecase
SBY 22:52:41 [fifo_nofullskip] summary: counterexample trace [basecase]: fifo_n
ofullskip/engine_0/trace.vcd
SBY 22:52:41 [fifo_nofullskip] summary:   failed assertion fifo.a_count_diff at
 fifo.sv:106.13-107.71 in step 2
SBY 22:52:41 [fifo_nofullskip] DONE (FAIL, rc=2)
SBY 22:52:41 The following tasks failed: ['nofullskip']
(OSS CAD Suite) pooja@pooja-VirtualBox:~/Desktop/Riscoveri/yosys/sby/docs/examp
les/fifo$
```

- Gtkwave



8. Conclusions and Future Scope

The future scope of this project includes:

1. **Tool Expansion:** Exploring more advanced open-source VLSI tools for power analysis, formal verification, and custom layout generation to enhance the toolkit's capabilities.

2. **Integration Improvements:** Creating a unified and user-friendly package by streamlining tool integration and compatibility.

3. **Community Engagement:** Encouraging community collaboration and feedback to refine the toolkit and create an open-source platform for knowledge exchange.

4. **Educational Initiatives:** Conducting workshops, webinars, and educational resources to educate and inspire a broader audience in VLSI design.

5. **Design Examples:** Providing diverse design examples to showcase the toolkit's versatility and efficiency in real-world applications.

6. **Performance Optimization:** Investigating optimization techniques for implemented designs to enhance efficiency and resource utilization.

7. **Hardware Acceleration:** Exploring FPGA or ASIC platforms to accelerate simulation and synthesis processes for faster design iterations.

By pursuing these directions, we aim to make a lasting impact, empowering VLSI enthusiasts and fostering innovation in the open-source VLSI community

9. References

1. List of Open Source EDA tools by Andreas Olofsson - GitHub
2. Icarus Verilog by Cary R. - GitHub
3. Verilator by Aleksander Kiryk - GitHub
4. Gtkwave by Unai Martinez-Corral - GitHub
5. Yosys - GitHub - Website
6. SymbiYosys by Claire Xen - GitHub