Volume: 10 Issue: VI Month of publication: June 2022

DOI: https://doi.org/10.22214/ijraset.2022.44249

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

Implementation of RISC-V SoC from RTL to GDS flow using Open-Source Tools

S Nikhil Kumar Reddy[1], Shashank Viswanath Hosmath[2], Sharanakumar[3], Sandeep[4], Vinay B K[5]
[1, 2, 3, 4]*Department of Electronics and Communication Engineering, Vidyavardhaka College of Engineering, Mysuru, India*
[5]*Assistant Professor, Department of Electronics and Communication Engineering, Vidyavardhaka College of Engineering, Mysuru, India*

Abstract: The VSD QFLOW project is working on an open-source RTL-to-GDS tool that can produce a manufacturable layout from a hardware description in less than 24 hours and without involving a person.
Opensource EDA tools are used to investigate and research an IC design for this purpose. This article explains how opensource EDA tools can assist researchers and students in learning.
How to design and construct their own integrated circuits. Many hurdles and collaboration among different entities are involved in such an undertaking. We now have an open-source foundry. PDK as well as a thriving open-source hardware design environment. Th over all process can help in VLSI design. Optimizes error free GDS file for the given specification

I. INTRODUCTION

Modern technologies provide a growing number of design requirements as well as increasingly demanding power, performance, and area targets. As advanced SOC devices necessitate more commercial EDA tool licenses and larger teams of experienced engineers, the cost of IC design continues to rise Moore's Law has been the driving force behind the semiconductor industry for many years.
Moore's Law states that every 18 to 24 months the number of transistors doubles. The International Technology Roadmap for Semiconductors has been based on Moore's Law since 1998.Electronic Design Automation (EDA) software development firms have long recognized this.
They put a lot of money into IP development, to the point where the two most well-known EDA tool companies are also the second and third largest IP produces.
In current generation, the complexity of IC design has skyrocketed in recent years. Silicon implementation is out of reach for system innovators because to growing design costs, intricacy, and risk.
For decades, custom ASIC design and fabrication for the masses has been a pipe dream due to a slew of significant roadblocks, including a lack of open-source EDA tools scaled to modern processes and supported by foundries, public-access foundry Process Design Kits, and design data and EDA format files that can be published in the public domain. The open-source community has made several efforts.
In this study, we primarily focus on how the outcomes of opensource EDA tools, such as the Open Lane flow and its application to Digital ASIC design, compare with the commercial flow.

II. LITERATURE SURVEY

A. VSD Flow

Using the Sky130 open source PDK, QFLOW is being developed to physically implement the picoRV32. The Open VSD flow's major purpose, as shown in Figure, is to provide a clean layout for any RTL design automatically.
VSD flow is made up of open-source EDA tools from a variety of projects, including Open ROAD and Open flow. VSD flow includes over seventy scripts and tools, as well as forty various configurations, sixteen of which are design-specific. VSD flow can be used to physically implement any design for any process technology, despite the fact that it was intended for use with the Sky130 PDK. It has been effectively shown with targeted designs.
All physical design steps in the Open ROAD project are governed by a single environment: the VSD flow. It's a collection of tools that cover everything from logic synthesis to layout extraction. An RTL Verilog file, design restrictions, and design enablement elements such as Liberty, LEF, and technology files are all inputs to this pipeline. The output is a GDSII file that may be taped.

B. RISC-V ISA

(RISC-V Instruction Set Architecture) RISC-V is an open standard instruction set architecture (ISA) based on the ideas of the reduced instruction set computer (RISC). Despite not being the first open architecture ISA, it is notable since it is intended for usage in modern computerised devices such as large-scale cloud computers, high-end mobile phones, and the tiniest embedded systems. The instruction set is intended for a variety of applications. The ISA enables variable length extensions, which let each instruction to be any number of 16-bit parcels in length. RISC-V is designed to be modular, with different base pieces and optional additions. The ISA base and extensions are the result of a collaborative effort involving industry, academia, and research organisations. Instructions, control flow, registers, memory and addressing, logic manipulation, and ancillaries are all specified in the base. With complete software support, including a general-purpose compiler, the base alone can create a simplified general-purpose computer.

C. Real-world Implementation

Physical implementation is a difficult stage that takes a long time to complete depending on the design complexity. It starts with CADENCE ENCOUNTER, INNOVUS, and IC Compiler importing the design, standard cells library, including layout and timing models, technology file, and timing restrictions. The tool then constructs its working environment, which includes Floor Plan, Power Plan, Placement, CTS, and Routing, to conduct all PnR stages. The purpose of this step is to arrange all of the standard cells, Macros, and I/O pads with the least amount of space, delay, and routing such that no DRC violations occur.

D. picoRV32

It is a RISC-V CPU with a small footprint. PicoRV32 is a RISC-V RV32IMC Instruction Set implementation CPU core. It comes with a built-in interrupt controller and can be configured as RV32E, RV32I, RV32IC, RV32IM, or RV32IMC core. PicoRV32 is free and open hardware licensed under the ISC license (which is equivalent to the MIT license or the 2-clause BSD license in terms of terms). This CPU is intended for use as an auxiliary processor in FPGA and ASIC systems. It can be fitted into most existing designs without crossing clock domains due to its high f max. It has a lot of timing slack when operating at a lower frequency, therefore it may be added to a system without sacrificing timing closure.

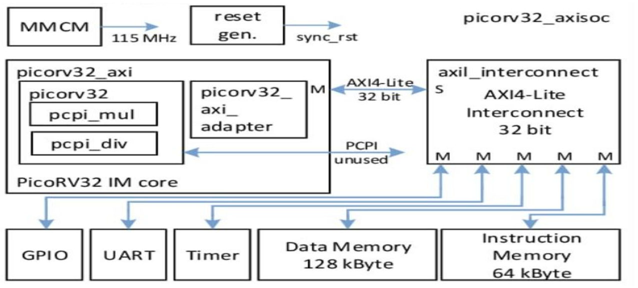


Fig1- Block diagram of picoRV32

III. PROPOSED METHODOLOGY

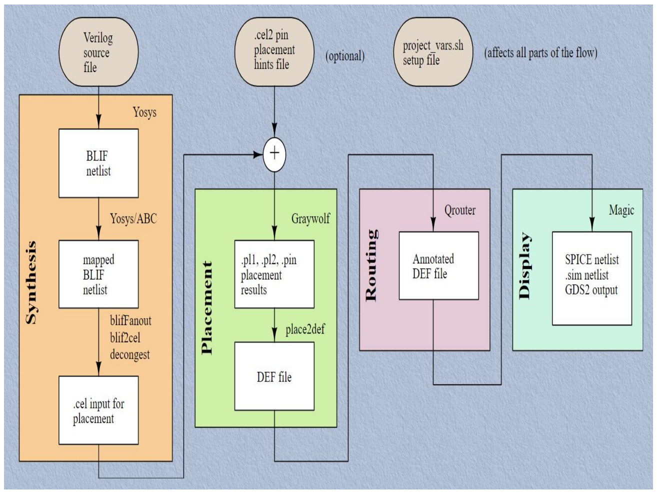
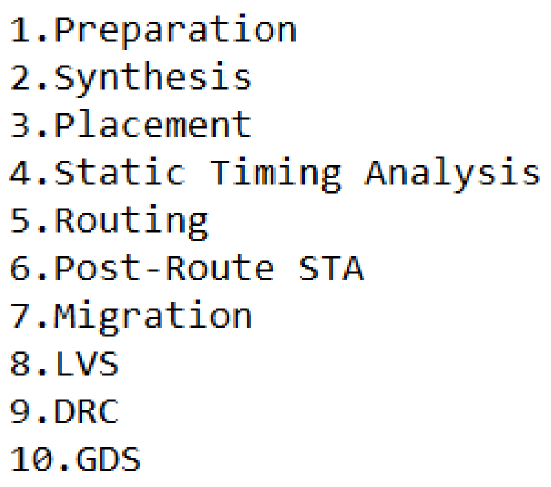

Verilog source file
.cel2 pin placement hints file
(optional)
project_vars.sh setup file
(affects all parts of the flow)
Synthesis
Yosys
BLIF netlist
Yosys/ABC
mapped BLIF netlist
blifFanout blif2cel decongest
.cel input for placement
Placement
Graywolf
.pl1, .pl2, .pin placement results
place2def
DEF file
Routing
Qrouter
Annotated DEF file
Display
Magic
SPICE netlist .sim netlist GDS2 output

Fig2- The design flow methodology

The process is based on a long-term collection of source code and its defects. Source code repositories keep track of when and where code defects first arose, as well as when and where they (dis)appeared.



```
1.Preparation
2.Synthesis
3.Placement
4.Static Timing Analysis
5.Routing
6.Post-Route STA
7.Migration
8.LVS
9.DRC
10.GDS
```

A. Synthesis

Yosys is in charge of the synthesis. To illustrate different degrees of area delay trade-offs, there are four default synthesis procedures that generate four different locations in the design space. If desired, a user can also add a custom synthesis script/strategy. We applied both techniques area and delay to evaluate how they affected abc logic synthesis and technology mapping. Open STA is used to perform static timing analysis on the resultant netlist.

B. HDL Synthesis

The design specification is accepted as System Verilog RTL by VSD synthesis process, which results in a library-mapped netlist implementation. Yosys is used by VSD for RTL. For logic synthesis and technology mapping, use synthesis and ABC . Some design limitations, such as the specification of the clock They can be stated, but only to a limited extent, and they do not follow the de facto standard Synopsys design constraints (SDC) format. A script that applies a series of modifications to the design controls ABC logic synthesis. It has been discovered that the order and quantity of For different designs, transformations generate different consequences VSD flow gives numerous synthesis options for this. For any given situation, there are strategies that can be automatically examined. design. The designer can make decisions based on the findings of the investigation. choose the optimal strategy based on the design goals (e.g., minimum area, delay, or power).

Assembly level language:

Bridge between software and hardware. The applications and software that operate on our laptops and computers are created in several languages, including C, C++, Python, and Java. The hardware cannot directly process and implement these many languages (Microprocessor Core). They must be transformed into a format that the core can comprehend. This is where the ISA (instruction set architecture) or assembly language (assembly language) comes into play. The compiler and assembler are special programmes that transform instructions from diverse languages to the target assembly language. The compiler translates the code into assembly language, which in our case is the RISC-V ISA, and the assembler converts the various textual instructions into binary representations that the CPU can understand.

C. Physical Inspection Physical Verification

Is carried out at numerous points during the cycle to ensure that the final layout is correct. Route collisions will be reported by the Qrouter, and Fast Route will check for potential antenna violations, which will be mitigated by feeding back into netlist updates and place-and-route processes. Although the final design is assumed to be DRC and LVS free, independent validation of the final layout through extraction and analysis is critical for trust in the mask data provided for foundry tape out. Magic, an open-source layout altering tool, has a number of features that can be used for this purpose. It reads DEF files, LEF macro libraries, the technology LEF file, and GDS mask data of library components, including the standard, and assembles the final design.

D. *Preparation*

User can control what's input for Synthesis. At what technology to target. once you select the file the file name will appear in the button and the module name will be set for the flow

E. *Verilog Synthesis*

YOSYS is an open-source Verilog synthesis and verification framework. It is compatible with all commonly used synthesis Ables. Verilog-2005 features, and it can target both FPGAs and microcontrollers. ASIC For logic optimization and LUT/cell, yosys employs ABC. Custom coarse-grained optimisations are integrated with mapping. as well as specific passes for inferring and mapping block-level information an distributed-RAM, flip-flops, and arithmetic structures are all examples of arithmetic structures. After logic elaboration, a typical FPGA flow would run. some coarse-grained optimizations and translate the outcomes to a collection of variable Hard-logic cells that are generic. After that, generic are utilised to infer. block-RAM, clock-enabled flip-flops, and set-reset flip-flops followed by architecture-specific arithmetic logic and more mapping of technology Any coarse-grain cells that remain are removed. Yosys turned the data to gates, which were then mapped to LUTs. ABC, followed by architecture-specific rules.

F. *Placement*

Many EDA projects are focused on the layout placement of analogue VLSI circuits. However, just a handful of them are talking about where high-voltage VLSI circuits should be placed. The design of high-voltage circuits differs from that of conventional circuits. For enhanced safety, isolation rings are frequently used around transistors. Because isolation rings will take up a lot of space on the chip, it's important to plan. create appropriate EDA tools for location optimization. To save money on chips, isolation rings are used. A placement is used in this paper. An optimization flow is suggested that takes both symmetry restrictions into account and isolation rings for high-voltage circuit layout automation. By altering the position of transistors within each isolator, Isolation rings of various shapes will be studied. to optimise the layout simultaneously throughout the placement algorithm area.

G. *PDN and Floorplan*

It includes two key components, does floor planning and power delivery network synthesis. The first is an integer programming-based macro block packing system that is aware of macro-level interconnection and is seeded by a global placement that is mixed-size (blocks and standard cells). The second component is the generation of a based power delivery network (PDN) using a safe-by-construction technique. IP global .cfg and IP local .cfg capture macro packing rules, while PDN. cfg captures safe-by-construction metal and via geometry information. The inability of academic open-source tool developers (or their tools) to see whole decrypted design enablement's from the foundry necessitates these config files.

H. *Synthesis of a Clock Tree Clock Tree Synthesis (CTS)*

Is performed using Triton CTS for low-power, low-skew, and low-latency clock distribution. based on the GH-Tree (generalised H-Tree) paradigm . A clock tree topology is discovered using a dynamic programming approach. with the smallest possible anticipated power, given the latency as well as biased targets Sinks are programmed using linear programming. Clustering and the location of clock buffers Routing at the leaf level is possible. be performed utilising either a single-trunk Steiner tree or a multi-trunk Steiner tree Algorithm of Prim-Dijkstra [1]. Triton CTS features interfaces in the layout generation flow. with the router and the placer (Replace) (Triton Route). The placer is used to legalise clock buffers that have been added. Sink pins are assigned to GCELLs that should be used by the router. for the routing of the clock tree LEF inputs are used in Triton CTS.

I. *Routing Qrouter*

Based on a global routing solution in route guide format, consumes LEF and placed DEF before performing detailed routing for both signal and clock nets. Qrouter pre-processes the global routing solution using a rapid approximation approach to ensure that each net has a Steiner tree structure. As a result, in the detailed routing stage, congestion and wirelength are decreased while net connectivity is kept. The detailed routing problem is then handled iteratively layer by layer, with each layer divided into separate routing panels. The panel routing problem is stated as a maximum weighted independent set (MWIS) problem, which is solved in parallel using a MILP-based technique. The MWIS formula distributes tracks in the most efficient way possible

J. *Migration*

The value of text is passed directly to the command line of magic when processing the migration script.

K. LVS

You must also run LVS (Layout vs Schematic) to ensure that your layout matches your source netlist before signing off a design. At a high level, this is accomplished by providing your layout and a Spice netlist to the LVS tool. The tool then attempts to match a netlist extracted from your layout to the source netlist. An LVS run-deck will be provided by the PDK you're using, which defines specific criteria for the LVS tool directory. LVS mistakes are frequently caused by metal shorts or problems with the power network in a completely digital flow. Before resolving DRC violations, some designers choose to focus on getting their design LVS clean first.

L. DRC (Design Rule Check)

You should run DRC on your design to see if it complies with the rules of the technology you're working with. A design rule manual is usually included with a PDK. There are numerous rules, some of which deal with metal spacing, metal density, antenna rules, and other topics. For a design to be transmitted to the foundry, it must be DRC clean, albeit certain restrictions may be relaxed in exceptional instances.
The DRC log Design will be displayed as a result of this action. Text reports can also be found in the drc-rundir directory. You won't need to remedy any DRC infractions, but it's a good practise to check over and understand a couple of them. You can scroll through the violations and click on certain of them. In the Design Review GUI, it will zoom in on the violation and highlight it. There may be a huge number of violations when you first go through a design, many of which can be addressed in a systematic manner by modifying some flags or commands in your flow. You might wind up addressing some of the infractions once the number of them is manageable.

M. GDS (Graphic Data Stream)

It is a binary file format that uses a hierarchical structure to represent layout data. Various CAD applications create data like as labels, forms, layer information, and other 2D and 3D layout geometric data in Integrated Circuit (IC) design.

N. Overview of VSD QFLOW

VSD QFLOW, an open-source end-to-end, is discussed in its current condition. The goal of VSD QFLOW is to make system designers' lives easier by lowering the costs, expertise, time, and risk they face today. The open-source flow, which is built around an industrial-strength open-source design database, includes design tools such as physical synthesis, floor planning, placement, clock tree synthesis, global routing, and detailed routing. Static timing analysis, parasitic extraction, power integrity analysis, and cloud deployment analysis and support tools are all included in the pipeline. Users can customise the entire flow and experiment with different tool parameters and optimizations using scripting.

IV. RESULTS AND DISCUSSION

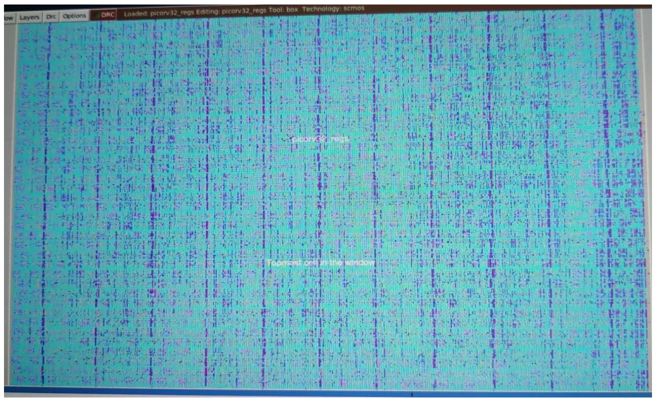


Fig3- Final Layout

The fig3 represents the final layout of the picorv32 RISC V processor core that we have obtained from the VSD qflow process. Although we are able to find out number of wires, memories , memory bits and number of gates that are used in the design. We have observed that the total area is 2.99e+09 computed at the maximum clock frequency of 1153.65 MHz .

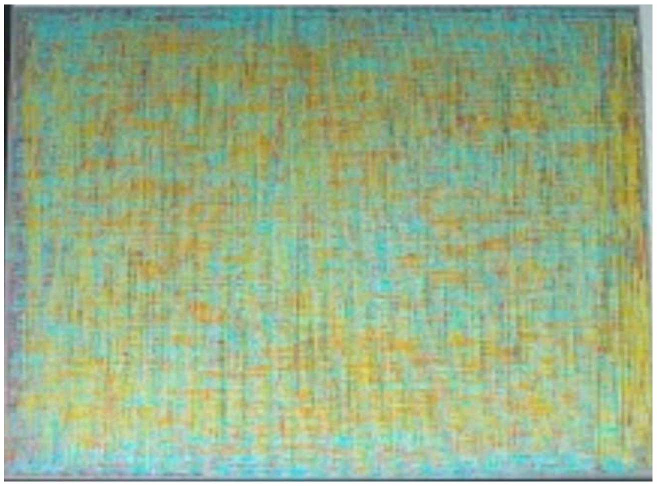


Fig.4 standard cell placement

V. CONCLUSION

VSD flow, a digital ASIC flow constructed with open-source EDA tools, was demonstrated. Given the correct set of design parameters, VSD flow can automatically produce manufacturable layout from RTL for many designs. It can be used to harden macros and put chips together. This paper outlines how open source EDA tools can assist students and researchers in learning how to design and construct their own Digital ASICs and Integrated Circuits. It also demonstrates how the EDA tool development community is alive and well, with new tools and methodologies becoming accessible, allowing them to be integrated into current workflows with a degree of flexibility that commercial tools lack. We will investigate the ability of training Machine Learning models to predict the optimal configuration given a design, as the quality of a layout is determined by the flow design configurations. The number of RISC-V processors has increased in the last years. This fact is explained by the recent efforts that were made on the RISC-V development with the support offered by the open-source community and companies that integrate the RISC-V Foundation. It is expected that in the future, several commercial products will include RISC-V cores instead of the current closed and proprietary solutions.

REFERENCES

[1] Mohamed Gaber, Manar Abdelatty and Mohamed Shalan, "Fault, an Open Source DFT Toolchain," Proceedings of the Workshop on OpenSource EDA Technology (WOSET19), November 2019
[2] Clifford Wolf and Johann Glaser, "Yosys - A Free Verilog Synthesis Suite“ Proceedings of Austrochip, 2013.
[3] Matthew R. Guthaus, James E. Stine, Samira Ataei, Brian Chen, Bin Wu and Mehedi Sarwar, "OpenRAM: An Open-Source Memory Compiler,“.
[4] Waterman, A., and K. Asanovic. “The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Document Version 2.2. CS Division, EECS Department.” University of California, Berkeley (2017).
[5] R. Timothy Edwards, M. Shalan and M. Kassem, “Real Silicon using Open Source EDA,” in IEEE Design & Test, doi: 10.1109/MDAT.2021.3050000.
[6] C. J. Alpert, W.-K. Chow, K. Han, A. B. Kahng, Z. Li, D. Liu, S. Venkatesh, “Prim-Dijkstra Revisited: Achieving Superior Timing-driven Routing Trees”, Proc. ACM/IEEE Intl. Symp. on Physical Design, 2018, pp. 10-17.
[7] S. Hashemi, C.-T. Ho, A. B. Kahng, H.-Y. Liu and S. Reda, “METRICS 2.0: A Machine-Learning Based Optimization System for IC Design”, Workshop on Open-Source EDA Technology, 2018, pp. 21:1-21:4.
[8] Andrew B. Kahng, Lutong Wang and Bangqi Xu, "TritonRoute: The Open Source Detailed Router," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (2020), to appear.
[9] “Magic VLSI Layout Tool,” https://github.com/RTimothyEdwards/magic, 2020.
[10] Abdelatty, Manar, Mohamed Gaber, and Mohamed Shalan. ”Fault: Open-Source EDA’s Missing DFT Toolchain.” IEEE Design & Test (Volume: 38, Issue: 2, April 2021).