Grenze International Journal of Engineering and Technology, June Issue

Comparative Analysis of Open-Source EDA Tool for VLSI Physical Design

Hardi Sangani[1] and Dipesh Panchal[2]
[1-2]Department of Electronics and Communication Engineering, Nirma University, Ahmedabad, India
Email: hardi.sanagani111@gmail.com, dipesh_panchal@nirmauni.ac.in

Abstract*—Open-source EDA tools are becoming very popular nowadays and they are also becoming more efficient. Academic researchers can use these open-source EDA tools which are free of cost and still give the idea about the whole chip design flow. These open-source EDA tools can also be used for real chip tap out so we can also manufacture a chip with the layout made with such tools. Commercial tools require very high license costs and are used in industries for high efficiency. The arithmetic and Logic Unit is the fundamental building block in any processor which performs arithmetic and logical operations. ALU is widely used for signal processing tasks where lots of additions and multiplications need to be performed at very high speed and also in communication-related applications. In this paper, the whole flow of chip design is performed on 8-bit ALU with both open-source tools that are Qflow and commercial tool that is Cadence Encounter. The RTL design is first written in Verilog and also functionally verified. Then Synthesis is done which converts RTL code into a gate-level netlist. Then different physical design steps like partitioning, floorplanning, power planning, placement, CTS, routing, timing closure are performed. After STA, LVS, DRC, the final layout is generated that is GDS-II file which can be given for tape out. The 180nm technology node is used in the physical design. With so many advantages of open-source tools, to know how efficient they are, design is done in Qflow and Cadence Encounter. Results are compared based on area, speed, and power which are the most important parameters in VLSI design. From the result, it can be seen that the area requires in design made with Qflow is almost 4 times larger than the design made with Cadence Encounter. Power required for design with Qflow is 25 times higher than design with Cadence Encounter.

***Index Terms*— Physical Design, ALU, Open-source EDA tool, Commercial tool, Performance Comparison.**

I. INTRODUCTION

The ALU reads two operands A and B, performs various arithmetic and logical operations on it depending on the opcode value which behaves like a control input as shown in "Fig. 1". Depending on various opcode values different operations are performed and the truth table for it is shown in TABLE 1.
The physical design flow of any digital design converts higher-level design like RTL e.g. Verilog, VHDL code to the physical layout [1]. Qflow is a toolchain that synthesizes any digital circuit and performs various physical design steps on it and finally gives GDS-II file depending on the processing technology selected [2]. For power analysis, a tool designed by the VSD intern is used [3]. Cadence Encounter is a commercial tool used in industries for real chip tap-outs [4]. There are so many commercial tools available for physical design like

Grenze ID: 01.GIJET.8.2.8

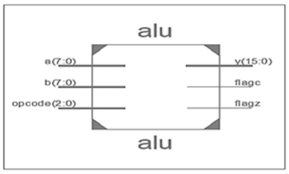


Figure 1. RTL design of 8-bit ALU

TABLE I. TRUTH TABLE OF 8-BIT ALU OPERATIONS

Opcode	Operation
000	Y= a + b
001	Y= a - b
010	Y= a * b
011	Y= a & b
100	Y= a | b
101	Y= ~ (a & b)
110	Y= ~ (a | b)
111	Y= a ^ b

Cadence Encounter, Synopsys, Mentor graphics but they are very expensive so startups, academic researchers can't afford them. Nowadays, open-source tools are used widely as they are also efficient in chip designing but there is a lack of efficiency in designing very complex designs in terms of area, power, and delay [5].

II. PHYSICAL DESIGN FLOW

The basic flow of physical design is shown in "Fig. 2" [6]. The first step in the physical design flow is a synthesis that converts RTL design into a gate-level netlist. This netlist is given as input to physical design steps. After synthesis, the next step is floor planning which estimates the area of the whole design by arranging logical blocks tentatively. Width, height, and aspect ratio are given as inputs to floor planning. Core utilization is also given which defined how much total core area is used for standard cell placement and the remaining is used for routing, generally, it is selected as 70%.

The next step is power planning to distribute power equally to each part of the chip. The problem of voltage droop and ground bounce arises if we use a single power source. So that a grid-like structure is created to distribute power uniformly. First power rings are created around the chip for VDD and VSS and then power strips are created which gives power to the core area [7]. For rings, the topmost metal layer is used as it is thick and hence contains low resistance and low IR drops. The lowest metal layers are used for rails as it is directly connected to standard cells which are designed with lower metal layers.

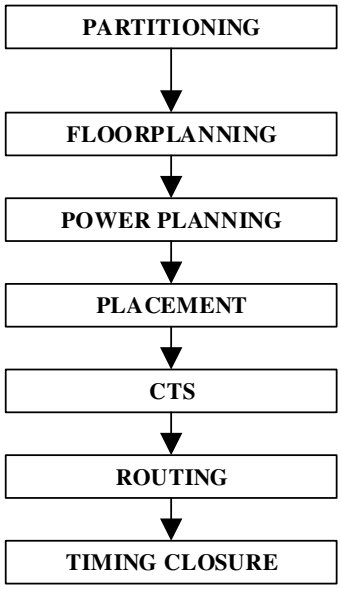
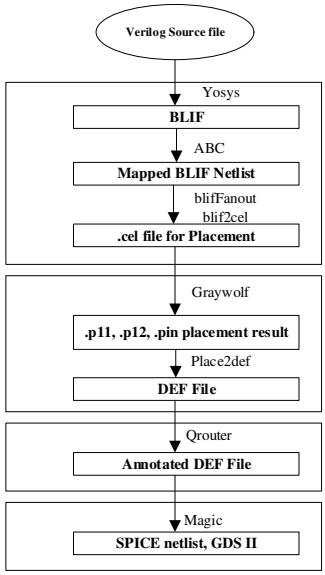


Figure 2. VLSI Physical Design Flow



Figu re 3. Qflow GUI Design Synthesis Flow

After power planning, placement is done in which all standard cells are arranged in such a way that wire length is minimized and timing constraints are met, the area is minimized. In Qflow, a Graywolf placer is used which

uses a min-cut algorithm that first partitions netlist and moves cells such that minimum wire length is obtained between two partitions.
Clock tree synthesis is performed after placement in which the clock is distributed in all parts of a chip. If the clock is not distributed uniformly then the problem of skew arises which is the difference between clock arrival time at different places in the chip. Different structures like H and grid are used to avoid the problem of skew. Buffers are also added in the path to balance the delay.
After placement, we do routing which interconnects standard cells as per design. Routing is done with multiple metal layers. Routing is done in such a way that interconnect length is minimized. Two steps are performed in routing those are global routing and detailed routing. Interconnect length adds capacitance and resistance which accounts for the delay of the chip. The next step in the flow is timing closure in which we check for setup and hold violations in the design with STA. In case of any violation, the design is optimized by adding buffers or decreasing path delay in the design to meet timing.
At last Design rule check (DRC) and Layout versus schematic (LVS) are performed. In DRC it is checked that the final layout follows all the rules made by the foundry such that it can be manufactured. LVS checks that physical design is the same as gate-level netlist that is no change in functionality of the design is assured after all physical design steps.

III. Detailed Flow Of Tool

A. Qflow GUI Design Synthesis flow

Qflow GUI provides an interface to simplify the whole process of RTL to GDS-II conversion including synthesis, placement, routing, etc. as shown in "Fig. 3". The Qflow GUI is written in Python language. In this tool, Oklahome State University (OSU) standard cell libraries are used.
Synthesis is done with Yosys tools which read Verilog source file and convert it into gate-level netlist in BLIF format [8]. The abc reads logic format file and optimize the design and map it to standard cells and produce output as mapped BLIF netlist. The blifFanout analyzes the output node capacitance of every gate and adds buffers and cells with high drive strength if latency is not within the constraint specified.
The blif2cel script converts the final netlist to the file required by the Graywolf placement tool. The placement tool takes the .cel2 file as an input which gives information about the location and order of pins. The Graywolf performs placement of all cell instances of design and generates three files at the output. The place2def tool converts this output of Graywolf into a DEF file which contains all placement and netlist information and also generates a .cfg file containing information of routing layers.
Routing is done by the Qrouter tool which performs detailed routing on the input DEF file and generates annotated DEF file at the output. The Magic tool reads the LEF file containing information of routing layers and DEF file and converts display it as layout. Finally, the SPICE netlist and GDS-II file are generated as output.
The vesta tool is used to perform static timing analysis. The complete tool flow from the RTL design to the final GDS-II file is shown in the flowchart above.

IV. Experiment Results

The 8-bit ALU is designed with both the tools Qflow and cadence encounter. After completing all the steps, the result is compared. For the design of ALU, 180nm technology is used. In the placement step aspect ratio, 1 and chip density 1 are selected. In power planning, top metal layers 5 and 6 were selected.
The design is first simulated to functionally verify it and the NCLaunch tool is used for it and the simulation result is shown in "Fig. 4". As seen from waveforms, for a = 8'hAA and b = 8'h55, output y= 16'h3872 for opcode = 010.

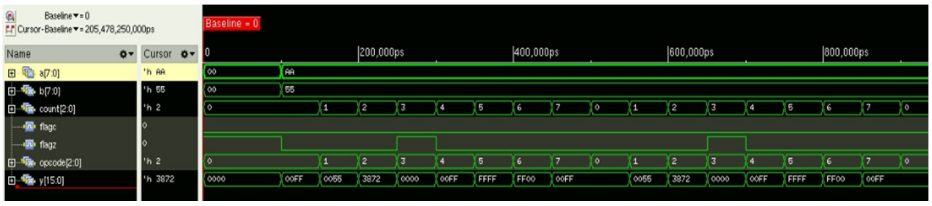


Figure 4. Simulation result of ALU in SimVision

Table II. Area Report Generated In Cadence Encounter

Instance	Cells	Cell Area
alu	342	7857
csa_tree_mul_23_9	127	3360
final_adder_mul_23_9	14	945

TABLE III. POWER REPORT GENERATED IN CADENCE ENCOUNTER

Instance	Cells	Leakage Power (nW)	Dynamic Power (nW)	Total Power (nW)
alu	342	43.180	874188.662	874231.842
csa_tree_mul_23_9	127	22.101	366465.850	366487.951
final_adder_mul_23_9	14	6.959	152268.719	152275.678

After functional verification synthesis is done with the Yosys synthesizer, the synthesis step generates a gate-level netlist which is technology-dependent. The area report in TABLE II shows the number of the sequential, inverter, and logic elements used in the design and also gives the total area. The power report in TABLE III shows leakage, dynamic, and total power consumption of the design. Yosys gives several wires and cells used in design as an output of synthesis as shown in "Fig. 5".

In "Fig.6", the left side figure shows the floorplanning result which decides the core-to-die boundary of design, and also aspect ratio decides the shape of the chip. The right side figure in "Fig.6" shows the power planning result where power rings and power rails can be seen.

The left side figure of "Fig.7" shows the placement result where green-colored standard cells are placed in predefined areas specified by floorplanning. The right side figure of "Fig.7" shows routing results that connect all standard cells as per design with interconnects. The Qflow tool uses a Graywolf placer for placement which also performs power planning and clock tree synthesis (CTS).

```
=== alu ===

   Number of wires:                583
   Number of wire bits:            629
   Number of public wires:           7
   Number of public wire bits:      38
   Number of memories:               0
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:                609
     $_ANDNOT_                      83
     $_AND_                         82
     $_AOI3_                        56
     $_AOI4_                         8
     $_DFF_P_                       18
     $_MUX_                         12
     $_NAND_                        37
     $_NOR_                         33
     $_NOT_                         29
     $_OAI3_                        30
     $_OAI4_                         9
     $_ORNOT_                       11
     $_OR_                          68
     $_XNOR_                        57
     $_XOR_                         76
```

Figure 5. Yosys Synthesis Result

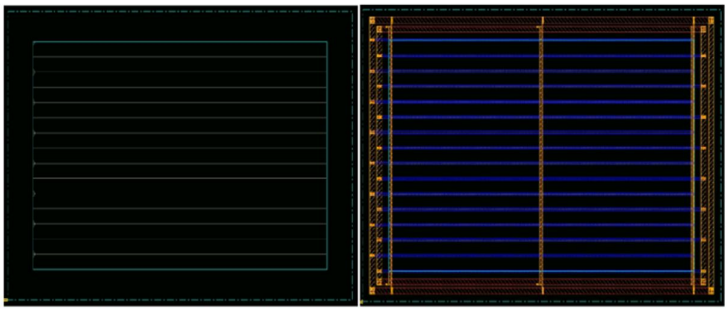


Figure 6. Result of Floorplan and Power planning in Encounter

A simulated annealing algorithm is used by Graywolf which places blocks such that it minimizes routing wire length. Modules go through different iterations before the final place is decided which is shown in the right side

image of “Fig.8”. The left side figure of “Fig.8” shows a magnified image of part of the layout after placement, in which different cells are seen in rows, and also power strips can be seen.

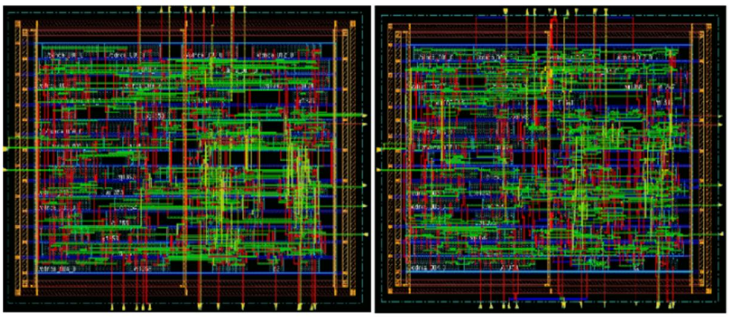


Figure 7. Result of Placement and Routing in Encounter

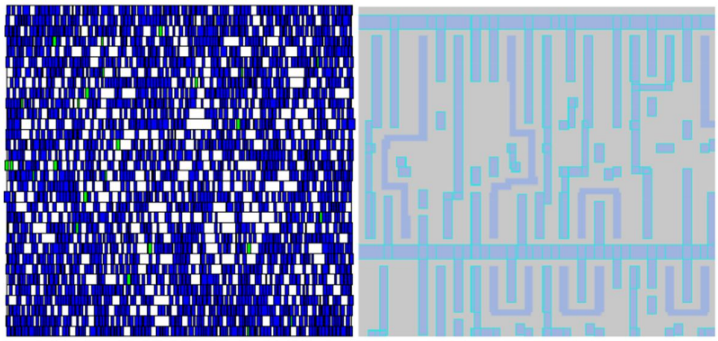


Figure 8. Result of Placement in Qflow

The Qrouter tool is used for routing purposes and performs detailed routing using Lee’s algorithm. The result of the routing step is shown in “Fig. 9”. The left side of “Fig.9” shows an image when routing was begin performed and the right side image shows the final magnified layout, interconnects can be seen between cells.
After all the steps like LVS and DRC, the final layout e.g. GDS-II file can be shown graphically with the Magic tool which is shown in “Fig. 10”. The magnified view of the final layout can be seen in “Fig.11” which shows a connection between cells as well as pins connected at the boundary of a chip.

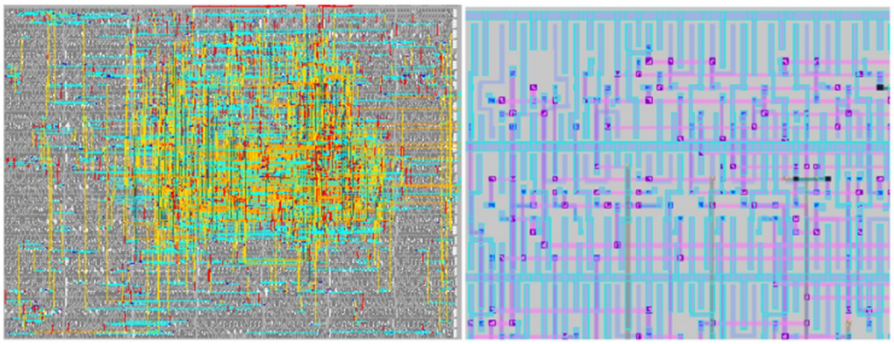


Figure 9. Result of Routing in Qflow

The comparison chart of the two tools is shown in “Fig. 12”. It can be observed from the chart that for the same design of 8-bit ALU, more cells are needed in Qflow and hence more area is required. The power required by design with Qflow is also high compared to Cadence Encounter as more number of cells and long routing wire length in Qflow. The maximum clock frequency that is the maximum speed with which the chip can be run is also less in the case of design with the Qflow tool.

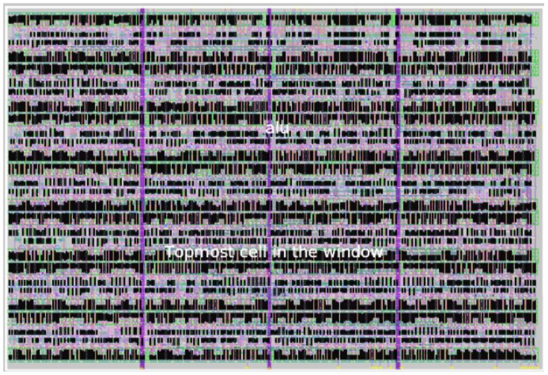


Figure 10. GDS-II of 8-bit ALU in Magic Software

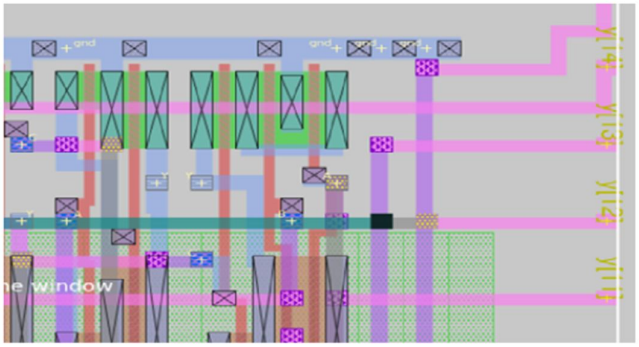


Figure 11. Magnified view of GDS-II layout in Magic Software

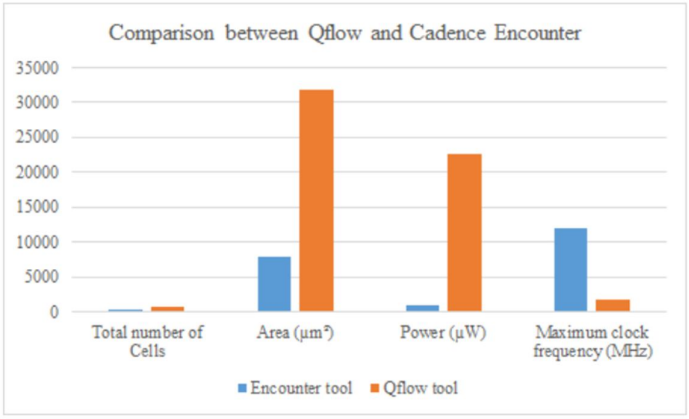


Figure 12. Comparison between Cadence Encounter and Qflow tool

V. Conclusion

In this paper, a commercial tool like Cadence encounter is compared with open-source tools like Qflow. Open-source tools are widely becoming popular among researchers and academia as they are free of cost but the main drawback of using such a tool is that they are not developed by considering all the aspects unlike the sophisticated tools and so we achieve less efficiency in terms of area, speed, and power. It is observed from the comparison chart that the area and power required by the Qflow tool is much more than what is required by the Cadence Encounter tool. The main reason for low efficiency is that open-source tools are not optimizing the

design by using sophisticated algorithms which commercial tools are using based on research done for many years. Qflow still uses some old algorithms like simulated annealing and Lee's algorithm. Yosys synthesizer tool does not support timing-driven synthesis and hence we cannot give timing constraints in SDC format which is used in the encounter tool to specify minimum and maximum delay as well as different delay corners. Hence such open-source tools need to be further developed with specifications that meet industry requirements so that they can be used for real chip tap out. When very high performance is not required and the chip is designed for academic and research purposes, an open-source tool can be used to understand physical design flow and can also be used for chip tap out. The Raven RISC-V microprocessor developed by efabless using Qflow is the best example of using an open-source tool for real chip tap out.

REFERENCES

[1] K. P. Ghosh and A. K. Ghosh, "Technology mediated tutorial on RISC-V CPU core implementation and sign-off using revolutionary EDA management system (EMS) — VSDFLOW," 2018 China Semiconductor Technology International Conference (CSTIC), 2018, pp. 1-3.

[2] Open Circuit Design Qflow tool, http://opencircuitdesign.com/qflow.

[3] Power calculator tool by VSD intern, https://github.com/CharuGupta-eng/vsdOSPowerCalc.

[4] Cadence Encounter tool, https://www.cadence.com.

[5] A. Rovinski, T. Ajayi, M. Kim, G. Wang and M. Saligane, "Bridging Academic Open-Source EDA to Real-World Usability," 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), 2020, pp. 1-7.

[6] A. Suresh and S. Nandi, "Physical Bridge Design for High Performance to Peripheral Bus: An Open Source Approach," 2020 Third International Conference on Advances in Electronics, Computers, and Communications (ICAECC), 2020, pp. 1-5.

[7] Andrew B Kahng, Jens Lienig VLSI Physical Design: From Graph partitioning to Timing Closure, Publication: Springer.

[8] M. Shalan and T. Edwards, "Building OpenLANE: A 130nm OpenROAD-based Tapeout- Proven Flow : Invited Paper," 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), 2020, pp. 1-6.

[9] Sadiq M Sait, Habib Youssef ,VLSI Physical Design Automation: Theory and Practice , Publication: World Scientific.

[10] Sung Kyu Lim, Practical Problems in VLSI Physical Design Automation, 2008 edition, Springer.