Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0021

Effect of different transistor technologies on physical design of arithmatic logic unit

Merve Özcan Korum, Fırat Kaçar
Department of Electrical and Electronics Engineering, Cerrahpaşa Faculty of Engineering, İstanbul University, İstanbul, Turkiye

Cite this article as: Özcan Korum, M., & Kaçar, F. (2025). Effect of different transistor technologies on physical design of arithmatic logic unit. *J Comp Electr Electron Eng Sci,* 3(1),1-7.

Received: 15.10.2024 • **Accepted**: 22.12.2024 • **Published**: 17.04.2025

ABSTRACT

Aims: The aim of this open source physical design tool is to observe the impact of miniaturization in transistor technology.
Methods: This study encompasses the utilization of free or open-source software and the design processes involved in both digital and physical design phases within the realm of VLSI design. Additionally, open-source software was utilized as a synthesis tool during the physical design process. This approach allowed us to observe the impact of transistor technology shrinkage not only with licensed software but also with open-source alternatives. One distinguishing feature of this study lies in the verification of the design process through simulations, conducted both after the digital design phase and following the physical design phase. The physical design phase employed the open-source qflow software. Designs for 8-bit, 16-bit, and 32-bit arithmetic logic units (ALUs) were synthesized using transistor technologies of 180 nm and 350 nm.
Results: The results of the design experiments revealed that reducing the transistor technology scale for designs of the same bit length did not lead to a significant alteration in the number of standard cells comprising the design. However, the number of standard cells increased directly in proportion to the length. As technology continued to shrink, the physical area occupied by the design decreased, while static time analysis (STA) also diminished concurrently with technology scaling.
Conclusion: To promote the proliferation of VLSI design process-es, it is imperative to foster the development and utilization of open-source software. This approach facilitates the expansion of research and education in the field of VLSI design, ultimately augmenting the pool of skilled professionals in this domain
Keywords: Open source, qflow, transistor technology, physical design, arithmetic unit logic

INTRODUCTION

Transistors serve as the building blocks of chips, enabling the functionality of technological products in all aspects of our lives. Thanks to advancements in semiconductor technology, VLSI technology has become an indispensable element in today's world. VLSI technology offers both advantages and comes with expensive and time-consuming production processes. Essentially, the design phase can be divided into two main parts: behavioral design and physical design.

VLSI designs can be crafted professionally using licensed computer-aided design (CAD) programs, as well as open-source software. Some of the licensed CAD programs include Cadence, Synopsys, and Mentor. On the other hand, there are open-source alternatives such as qFlow, OpenLANE, Coriolis, VTR, and SymbiFlow (Chupilko et al., 2021).

The data processing center of the central processing unit is called the arithmetic logic unit (ALU). The ALU is one of the most important units in a microprocessor, where arithmetic and logic operations are performed. Arithmetic operations include addition, subtraction, multiplication, and division, while logical operations include AND, OR, and NOT. It is inevitable that different processors will have different arithmetic logic units, but fundamentally, every ALU includes the basic logical operations listed above (Aydın, M., 2022).

ALU designs find applications in both military and civilian sectors. The study involved the use of 8-bit, 16-bit, and 32-bit ALU designs. Within the framework of this thesis, the behavioral design and physical design stages of a basic-functionality ALU's digital integrated circuit design process were undertaken (Bedir, N., 2018). Behavioral design was conducted using Quartus II in the Verilog language and simulated with the ModelSim tool. For the physical synthesis process, the open-source qflow tool was employed, with the IRSIM tool serving as the simulation tool (Başak, S., 2011).

The behavioral designs of 8-bit, 16-bit, and 32-bit ALU designs were physically synthesized using 180 nm and 350 nm transistor technologies. Parameters such as the number of standard cells, occupied area, and static time analysis (STA) values were assessed. Furthermore, in addition to reviewing the literature, the design was subjected to ModelSim simulation following behavioral modeling. After physical synthesis, the accuracy of the design was validated using the IRSIM tool. Simulations from these two stages were compared

Corresponding Author: Merve Özcan Korum, merve.ozcan@ogr.iuc.edu.tr

This work is licensed under a Creative Commons Attribution 4.0 International License.

across various bit lengths and synthesis processes in different transistor technologies. These comparisons revealed that transistor technology scaling did not have a detrimental impact on the design's functionality (Bakacak, M., 2021).

In the results of the thesis study, it was observed that as the bit length in the design increased, the number of cells used also increased in direct proportion to the bit length. The reduction in transistor technology scale employed during synthesis did not have an impact on the diversity of cells in the design. As transistor technology advances, i.e., as transistors shrink in size, the number of transistors within a unit area increases (Gaurav et al., 2022), (Purohit et al., 2021).

Through the synthesis of designs with varying bit lengths and two distinct transistor technologies, we observed their impact on the design's latency. Surprisingly, even as the bit length increased, the reduction in transistor technology managed to decrease the delay time. Furthermore, we examined the changes in latency resulting from the inclusion of capacitive and resistive values following physical synthesis in an open-source software. The obtained results align with previous studies in the literature (Sırmaçek, B., 2007).

The leading countries in the global semiconductor industry are China, the USA, Taiwan, and Japan. Taiwan meets approximately 60% of the world's semiconductor demand, while its closest competitor, South Korea, accounts for around 20%. The major global semiconductor manufacturers include Texas Instruments, Nvidia Corp, Micron Technology Inc. (MU), Qualcomm Inc. (QCOM), Intel, Samsung Electronics, Taiwan Semiconductor Manufacturing Company (TSMC), SK Hynix Inc., and Broadcom Inc. In Turkiye, significant advancements in chip design and production are being made through academic efforts led by METU MEMS and Bilkent UNAM. Additionally, under the supervision of TÜBİTAK YİTAL (semiconductor technologies research laboratory), companies such as ASELSAN, YONGATEK, and ERMEKSAN are actively involved in both production and sales within Turkiye (Ozcan, M.,2023) (**Table 1**).

Table 1. Semiconductor manufacturing facilities in Turkiye (Ozcan, M., 2023)

Product	Production facility	Production place
CMOS 250 nm	YİTAL Research Center	Gebze
CMOS/SiGe 130nm	YİTAL Research Center/ASELSAN	Gebze/Ankara
CMOS 90 u-65 nm	YİTAL A.Ş / ASELSAN	Gebze/Ankara
MEMS	ODTÜ-MEMS/ASELSAN	Ankara
GaN	AB Mikro Nano/ASELSAN	Ankara
GaAs (infrared)	ODTÜ Kanal, ASELSAN	Ankara
InGaAs (infrared)	ODTÜ Kanal, CÜNAM, ASELSAN	Ankara
HgCdTe (infrared)	ODTÜ Kanal, ASELSAN	Ankara
High power laser (500W CW)	ERMAKSAN	İstanbul

This study encompasses both behavioral design using a free version tool and the entire life cycle of VLSI design processes with the open-source Qflow tool. Consequently, it offers valuable insights and resources for individuals interested in the field of design and synthesis. Additionally, digital and physical designs were executed at different bit lengths and using two distinct transistor technologies, with the accuracy of these processes being confirmed through simulation studies conducted post-synthesis.

METHODS

Materials and methods should be described with sufficient detail to allow others to replicate and build on published results. New methods and protocols should be described in detail while well-established methods can be briefly described and appropriately cited. Give the name and version of any software used and make clear whether computer code used is available. Chapter name may be adapted to the topic of the manuscript or split into more chapters if it benefits the readability of the manuscript.

The first stage, known as behavioral synthesis, generates the gate-level (RTL) working state of the design. In the subsequent stage, the physical aspects of the design necessary to reach the chip size are executed. This stage is referred to as physical synthesis. The entire process represents the procedure for constructing application specific integrated circuits (ASICs). The ASIC design process is illustrated in **Figure 1**.

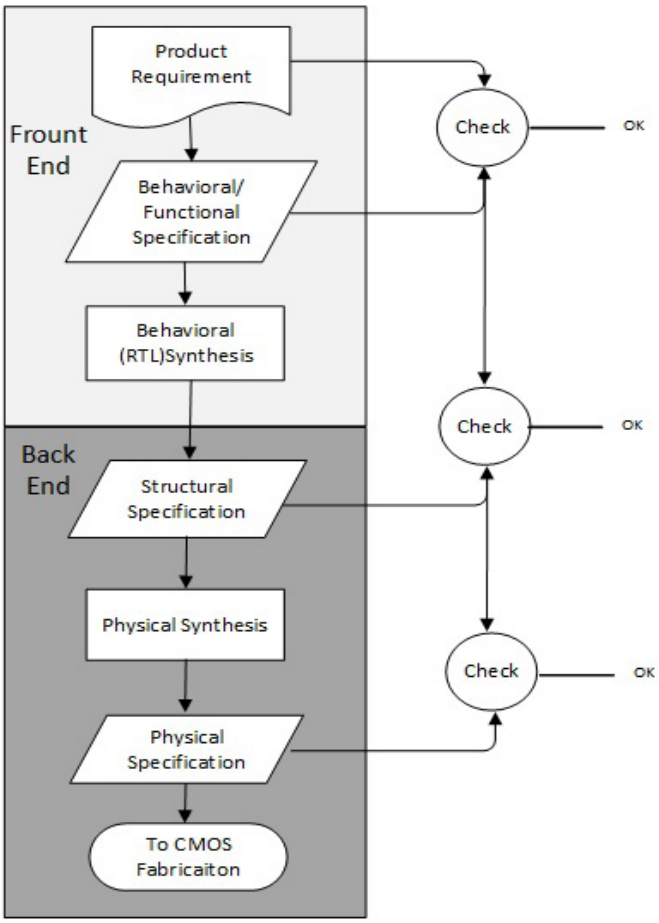

Figure 1. ASIC design process

ASIC: Application specific integrated circuit

Behavioral (Logic) Design

In ASIC development, behavioral design is often referred to as Front-end design. It typically involves four key steps (Purohit et al., 2021). The logic design process entails generating the RTL representation using a Hardware Description Language (HDL) to describe the functional behavior. Once the design steps are completed, each RTL module must undergo separate simulation to ensure accuracy.

Furthermore, IC (integrated circuit) design encompasses more than just logical designs; it also involves incorporating macros into the overall design. Fundamental macros include memory blocks, analog circuits, and input/output blocks.

During the logical design phase, modules for time adjustment and power management are also characterized.

The flowchart depicting the behavioral design processes is presented in **Figure 2**. The operations carried out in each step are explained sequentially.

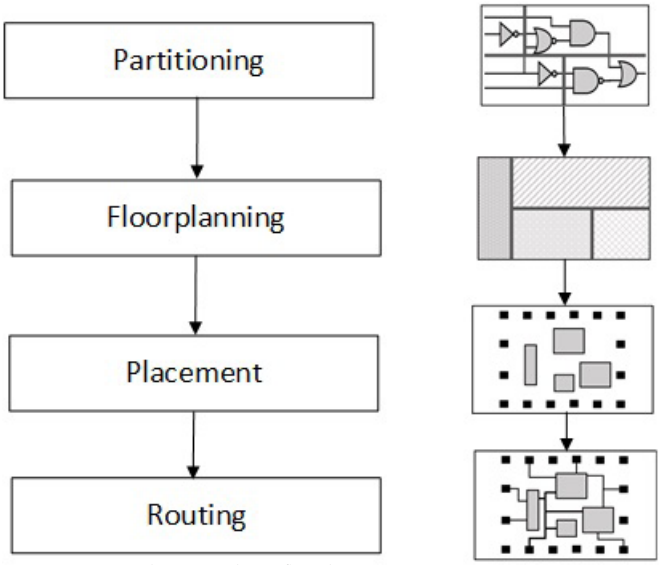

Figure 2. General gate synthesis flowchart

RTL-design entry: This is the stage where the design is represented using a HDL. During this phase, programming languages such as Verilog HDL or VHDL can be employed.

Generic netlist synthesis: This phase involves establishing connections between logic cells and cells generated from the HDL software. The logic synthesis phase encompasses steps that involve gate optimization and the conversion from a behavioral model to a structural model, regardless of transistor technology. The library utilized in this step obtains necessary timing and power information from the .lib file, which is based on previously defined data within the layout information (.lef) files.

Technology structure and optimization: It is mapped based on the alignment of logic gates from the standard cell library of the technology supported by the design software.

Gate level network list: This is the stage where the cells responsible for constructing the physical structure of the design are generated, along with the connection lists between them.

During the validation phase, if any errors are identified, the logic blocks are remapped. This process continues until the design criteria are enhanced either through adjustments made by the synthesis tool or optimizations achieved by resizing the gates.

In RTL design, timing-related errors often surface, which VLSI engineers address through extensive optimization efforts. In some cases, this optimization can be so profound that it necessitates a complete regeneration of the entire synthesis process.

Physical Design

In ASIC design, the phase referred to as physical design is commonly known as the back-end design. It essentially involves four main steps.

This process aims to create the detailed geometric layout required to transform the synthesized output at the gate level, obtained during the design phase, into the final form of the integrated circuit. It is also known as physical design, constituting a critical part of the digital VLSI workflow. During the physical layout, specific positions are assigned to the components that make up the design, and interconnections and routing are established either over or between metal layers using additive manufacturing technology. Once the graphic design system (GDSII) flow concludes as a result of the physical design, the output consists of a set of instructions for production.

The flowchart depicting the physical design processes is illustrated in **Figure 3**. The operations conducted at each step are subsequently elucidated. Modern physical production tools automate the entire process, significantly reducing the potential for human errors. This process is not limited to licensed physical design tools used in professional settings but also widely employs open-source tools.

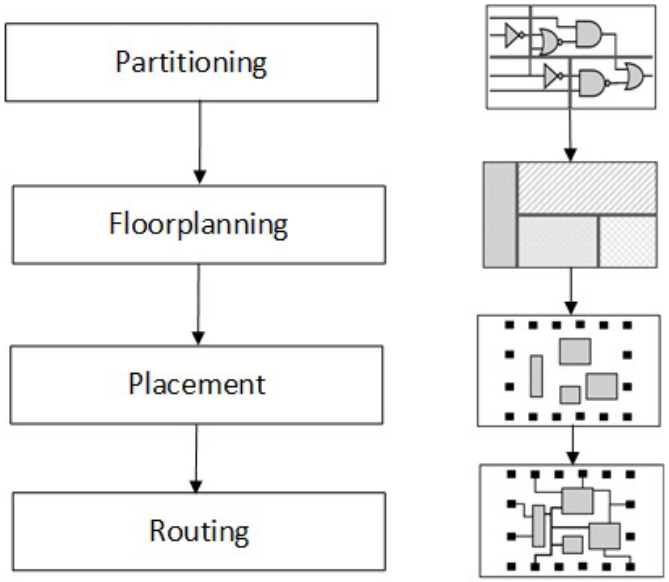

Figure 3. Physical design flowchart

Partitioning: Segmentation is a design approach wherein the larger modules or blocks constituting the design are broken down into smaller sub-components. This segmentation strategy is commonly employed to mitigate design complexity. Through this approach, each block can be individually designed or analyzed. Nevertheless, it's crucial to account for the connections between these blocks; otherwise, incorrect or incomplete connections can lead to performance degradation.

Floor planning: Each block structure can have either a fixed or flexible shape. Rigid blocks maintain fixed dimensions and areas, while flexible blocks have a fixed area but can vary in aspect ratios. This stage involves organizing both rigid and flexible blocks within the design according to the layout rules on the plane.

Placement: This stage involves making decisions regarding the placement of cells within the blocks of the design. All placements are predetermined to occupy specific areas. During this phase, timing constraints come into play, necessitating optimization to minimize the distance or connections between cells while preserving orientability.

An inefficient layout not only consumes more space but also gives rise to more timing errors. Furthermore, if time optimization is the primary objective during the placement step, it is possible to conserve space.

Routing: This stage involves establishing the connections between cells and blocks within the design. The routing process requires determining the topologies of the paths between standard cells and interconnects. The objective of concurrent routing optimization is to minimize the overall connection length and maximize timing efficiency.

Tool-QFLOW

The open-source Qflow tool is a toolchain that simulates the synthesis of a digital circuit design, originally modeled in Verilog, using specific manufacturing technology and processes.

In the commercial domain, widely-used and effective digital synthesis tools for chip design include software such as cadence and synopsys. These digital synthesis algorithms rely on closed-source software and are developed and controlled by various electronic design automation (EDA) software vendors. The prices of these software packages tend to be relatively high, making them less accessible for small and medium-sized digital synthesis projects.

Qflow 1.4 software is a computer-aided design (CAD) program used to transform RTL-modeled designs into GDS II layouts. This software is released under the General Public License (GPL), classifying it as open-source software. Qflow was originally developed by R. Timothy Edward in 2013, with the latest version published on 10th June 2021. Qflow comprises various sub-software components, including external tools such as Magic, Graywolf, and Yoys.

Figure 4 provides a schematic representation of the tools employed by the program in the synthesis process, including logical equivalence check (LEC) and Layout vs. Schematic (LVS) tools.

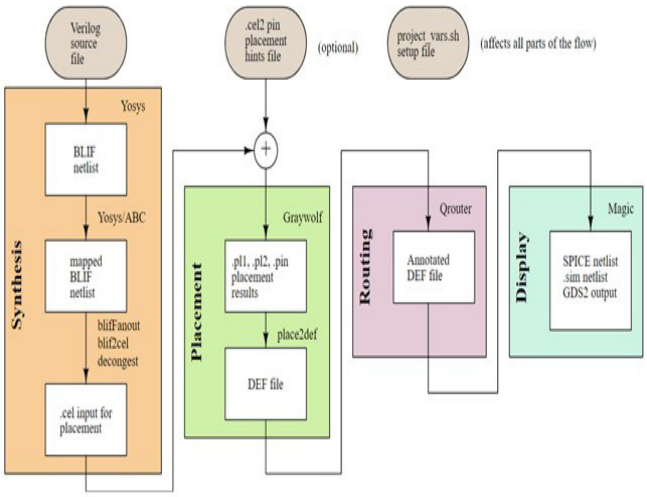

Figure 4. Qflow structural diagram

- Yosys (URL-1, 2022) and Odin II (URL-2, 2022) sub-softwares are used in the Logic Synthesis stage. As a result of this process, netlist links are created in BLIF format.
- The Graywolf (URL-3, 2022) subroutine in the layout and route phase is used in the initial block layouts and routing, and the qrouter (URL-4, 2022) subroutine is used in the specific routing step.
- OpenSTA (URL-5, 2022or vesta (URL-6, 2022) subroutines are used for Static Time Analysis (STA).
- The Magic (URL-7, 2022) subroutine is used for design check (DRC).
- Yosy and Netgen (URL-8, 2022) subprograms are used for LEC and LVC operations, respectively.

Qflow relies on MOSIS's "scalable CMOS rules" as the foundation, with open-source code serving as the standard cell library. Qflow specifically utilizes the open-source cell library from Oklahoma State University (OSU) This library encompasses OSU's 180-500 nm and FreePDK 45 nm technology cell libraries (Purohit et al., 2021). It's worth noting that industry leaders like X-Fab, IBM, or TSMC, which have their production facilities, typically do not disclose their standard cell libraries for sharing. The Oflow screen page is shown in Figure 5.

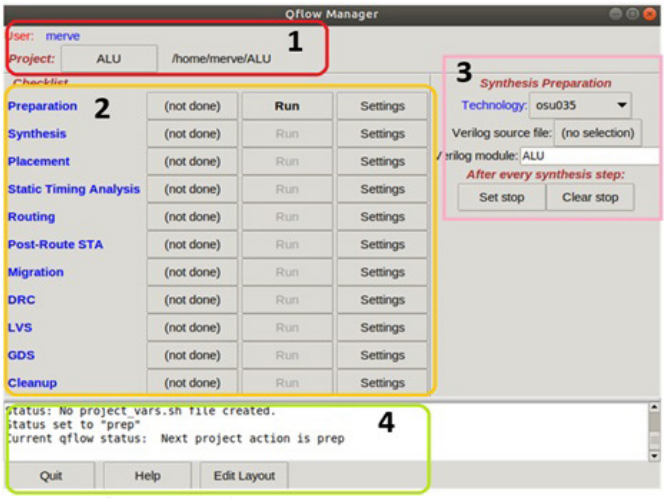

Figure 5. Qflow user interface

Section 1: This section provides the file path where the project is located.

Section 2: This section enumerates the steps conducted during the physical synthesis process. Each line includes checkboxes for initiating the operation, marking the operation as completed, and unique settings pertinent to that specific step.

Various transistor technologies were employed, including 180 nm and 350 nm technologies, for synthesizing ALU designs with different bit lengths. The ALU designs encompassed 8-bit, 16-bit, and 32-bit configurations.

Logic designs were developed using the Verilog language within the Quartus II environment. The logic design was simulated using Modelsim Altera 10.1d. Subsequently, for the physical design phase, the open-source Qflow 1.4 software was employed. Additionally, the accuracy of the physical design was confirmed through IRSIM simulation.

Data from the log/synth.log, log/place.log, and log/synth.log files have been tabulated for further analysis using graphic.

RESULTS

After synthesizing ALU designs with different bit lengths using the same transistor technology, we examined the impact of bit length on the number of standard cells. Following the synthesis of 8-bit, 16-bit, and 32-bit ALU designs with 180 nm transistor technology, the data regarding the standard

cell count is recorded in the place.log file, which is one of the Qflow tool's logs. The information extracted from this file is presented in Figure 6. It's evident that as the bit length of the design increases, there is a proportional increase in the number of cells composing the design. This finding aligns with the existing literature (Kang & Leblebici, 2005).

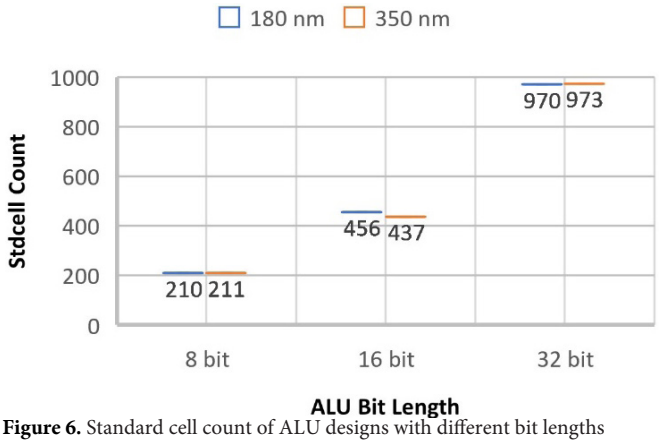

Figure 6. Standard cell count of ALU designs with different bit lengths
ALU: Arithmetic logic unit

Testbenches were created for IRSIM following the synthesis using 180nm and 350nm technologies with Qflow, and the designs were verified through simulations. The layout data in the generated log folder for different technology files is presented in Table 2. Similarly, the static timing analysis (STA) data in the log folder for different technology files is provided in Table 2.

DISCUSSION

As a result of synthesizing ALU designs with identical bit lengths but different transistor technologies, we observed the impact of changes in transistor technology on the occupied design area.

As a result of the physical synthesis of the 8-bit ALU design according to 180 nm transistor technology, the standard cell count was calculated as 211, the cell width as 5.89e+04 µm, the cell height as 2.11e+05 µm, the area covered by the design as 5.89e+07 µm² and the average cell height as 1.00e+03 µm. In addition, the maximum switching time of this design was calculated as 1114.15 ps and the minimum delay time as 187.453 ps.

As a result of the physical synthesis of the 16-bit ALU design according to 180 nm transistor technology, the standard cell count was calculated as 456, the cell width as 1.29e+05 µm, the cell height as 4.56e+05 µm, the area covered by the design as 1.29e+08 µm² and the average cell height as 1.00e+03 µm. In addition, the maximum switching time of this design was calculated as 1482.58 ps and the minimum delay time as 187.453 ps.

As a result of the physical synthesis of the 32-bit ALU design according to 180 nm transistor technology, the standard cell count was calculated as 970, the cell width as 2.74e+05 µm, the cell height as 9.70e+05 µm, the area covered by the design as 2.74e+08 µm² and the average cell height as 1.00e+03 µm. In addition, the maximum switching time of this design was calculated as 1769.47 ps and the minimum delay time as 187.453 ps.

As a result of the physical synthesis of the 8-bit ALU design according to 350 nm transistor technology, the standard cell count was calculated as 210, the cell width as 1.16e+05 µm, the cell height as 4.20e+05 µm, the area covered by the design as 2.31e+08 µm² and the average cell height as 2.00e+03 µm. In addition, the maximum switching time of this design was calculated as 1925.81 ps and the minimum delay time as 267.201 ps.

As a result of the physical synthesis of the 16-bit ALU design according to 350 nm transistor technology, the standard cell count was calculated as 437, the cell width as 2.45e+05 µm, the cell height as 8.74e+05 µm, the area covered by the design as 4.89e+08 µm² and the average cell height as 2.00e+03 µm. In addition, the maximum switching time of this design was calculated as 2451.97 ps and the minimum delay time as 267.201 ps.

As a result of the physical synthesis of the 32-bit ALU design according to 350 nm transistor technology, the standard cell count was calculated as 973, the cell width as 5.36e+05 µm, the cell height as 1.95e+06 µm, the design area as 1.07e+09 µm² and the average cell height as 2.00e+03 µm. In addition, the maximum switching time of this design was calculated as 3041.15 ps and the minimum delay time as 267.201 ps.

After successfully synthesizing them using Qflow, the area occupied by the designs is recorded in the place.log file. Figure 7 presents the area occupied by 8-bit, 16-bit, and 32-bit ALU designs with 180 nm transistor technology, along with synthesis results conducted with 350 nm transistor technology, which are also included in the same table.

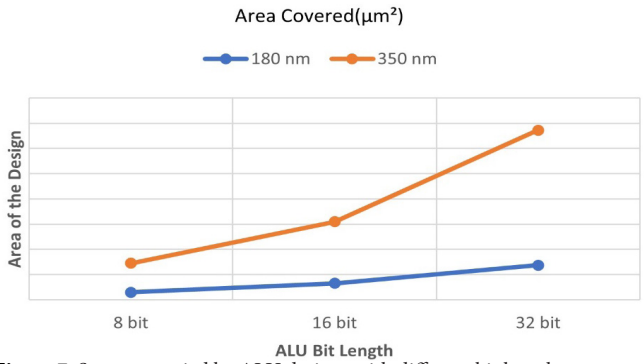

Figure 7. Space occupied by ALU designs with different bit lengths
ALU: Arithmetic logic unit

Table 2. Place.log and sta.log data in 180-350 nm technology

ALU design	Technology	SD cells	Cell width (µm)	Cell height (µm)	Design area (µm²)	Avg. cell height (µm)	Max transition time (ps)	Min delay (ps)
8-bit	180 nm	211	5.89e+04	2.11e+05	5.89e+07	1.00e+03	1114.15	187.453
16-bit	180 nm	456	1.29e+05	4.56e+05	1.29e+08	1.00e+03	1482.58	187.453
32-bit	180 nm	970	2.74e+05	9.70e+05	2.74e+08	1.00e+03	1769.47	187.453
8-bit	350 nm	210	1.16e+05	4.20e+05	2.31e+08	2.00e+03	1925.81	267.201
16-bit	350 nm	437	2.45e+05	8.74e+05	4.89e+08	2.00e+03	2451.97	267.201
32-bit	350 nm	973	5.36e+05	1.95e+06	1.07e+09	2.00e+03	3041.15	267.201

ALU: Arithmetic logic unit, SD: Standard deviation, Avg.: Averaj, Max: Maximum, Min: Minimum

As the transistor technology used in physical synthesis shrinks, the space occupied by designs with the same bit length decreases. However, when the bit length of the designs synthesized using the same transistor technology increases, the occupied design area also increases. These findings align with results reported in existing literature (Cao, 2021).

As a result of synthesizing ALU designs with varying bit lengths using the same transistor technologies, we observed the delays within the designs. Following successful synthesis with Qflow, delay information for the designs is recorded in the sta.log file. **Figure 8** provides pre and post static time analysis (STA) values for 8-bit, 16-bit, and 32-bit ALU designs using 180nm transistor technology. The preSTA value in the table represents the value calculated during the rough layout phase in physical synthesis. Subsequently, post STA values, accounting for capacitive effects, were calculated to observe delays using open-source software.

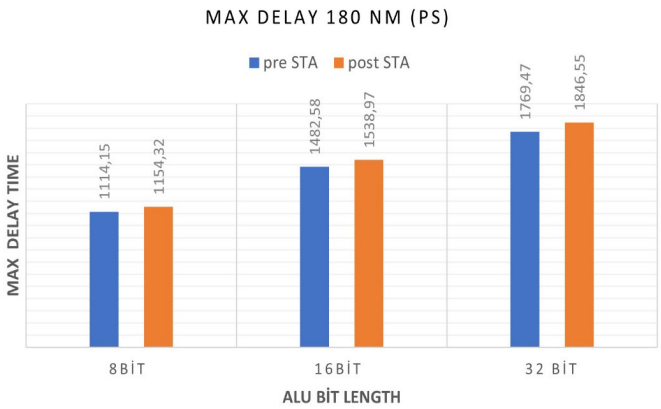

Figure 8. Pre and post STA times in 180 nm technology

STA: Static time analysis

As evident in the table, the post STA value for the 8-bit ALU design exceeds the preSTA value, and the delay time increases as the bit length increases. These observations align with findings reported in existing literature (Cao, 2021).

Using 350 nm transistor technology, we conducted physical synthesis for 8-bit, 16-bit, and 32-bit ALU designs and observed the latency of the designs. The results obtained from the sta.log file following the synthesis with 350 nm transistor technology are presented in **Figure 9** Notably, both pre and post static time analysis (STA) values have increased when compared to 180 nm transistor technology. This evaluation underscores a significant result that aligns with the trend of shorter processing times for designs as transistor technology scales down.

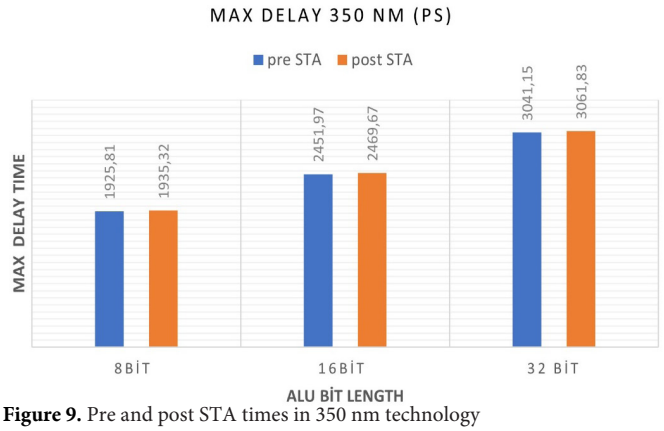

Figure 9. Pre and post STA times in 350 nm technology

STA: Static time analysis

In the thesis study, simulations were conducted to validate the accuracy of both the logic design and physical design stages. These simulations were replicated after synthesizing 8-bit, 16-bit, and 32-bit ALU designs using both 180 nm and 350 nm transistor technologies. The results obtained were found to align with the simulations conducted during the logic design and physical design phases. **Figure 10** provides an overview of the ALU design processes and the anticipated simulation results compared to the referenced initial values. For further clarification, **Figure 10** illustrates the outputs of the ModelSim simulation conducted for verifying the 8-bit ALU design, while **Figure 11** showcases the IRSIM simulation results following physical synthesis in the 180 nm transistor technology.

In both the ModelSim output shown in **Figure 10** and the IRSIM output displayed in **Figure 11**, the expected output for the "OP" value of "00110011" (hexadecimal 51) and "00000001" (hexadecimal 1) is indeed "00110011" (hexadecimal 51) and "00000001" (hexadecimal 1) in binary format, forming part of a sequence of 8 values being processed. This consistent result is observed across both simulations. Moreover, the alignment with the data in **Table 3** is maintained in other transactions as well.

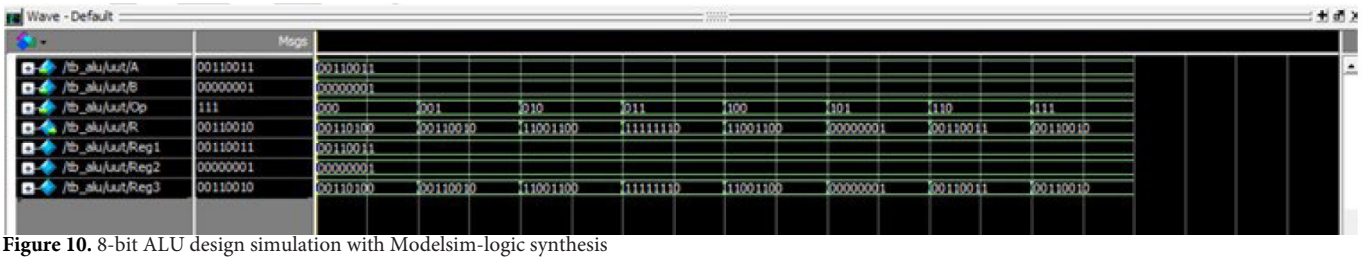

Figure 10. 8-bit ALU design simulation with Modelsim-logic synthesis

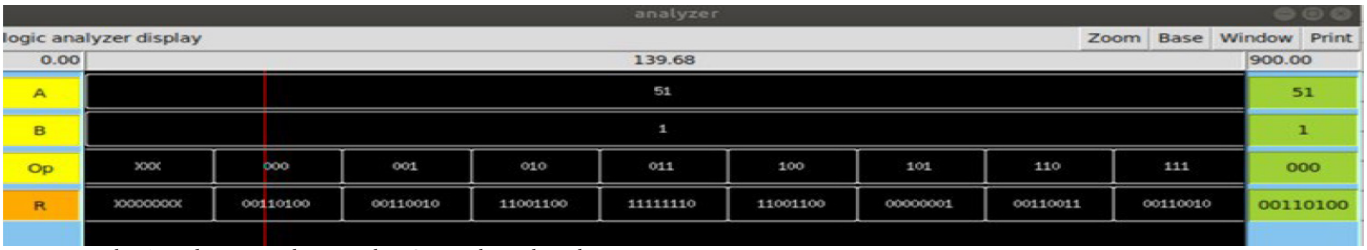

Figure 11. 8-bit ALU design simulation with IRSIM – physical synthesis

Table 3. Evaluation of Modelsim and Irsim results

Op	ALU process	Reg 1/A	Reg 2/B	ModelSim	Irsim
0	Addition	00110011	0000001	00110100	00110100
1	Subtraction	00110011	0000001	00110010	00110010
2	Not gate	00110011	0000001	11001100	11001100
3	Nand gate	00110011	0000001	11111110	11111110
4	Nor gate	00110011	0000001	11001100	11001100
5	And gate	00110011	0000001	00000001	00000001
6	Or gate	00110011	0000001	00110011	00110011

ALU: Arithmetic logic unit

CONCLUSION

This study encompassed a comprehensive simulation analysis, incorporating an open-source tool, following physical synthesis. Additionally, the accuracy of the designs was rigorously validated through simulation comparisons conducted after both logical and physical synthesis phases. The investigation examined the impact of shrinking transistor technology on standard cell counts for different bit lengths, the footprint of designs, and design delays, all assessed using open-source tools. Within the scope of this thesis, changes in transistor technology relative to design bit length were observed using open-source software. The reduction in transistor technology didn't yield a significant shift in the number of standard cells constituting the design. However, the advancement in transistor technology led to a reduction in the design's footprint, despite an increase in bit length. Furthermore, post-physical synthesis, design latency decreased as transistor technology scaled down. This study significantly contributes to the literature by verifying design accuracy through simulations conducted across different transistor technologies using open-source tools. To promote the proliferation of VLSI design processes, it is imperative to foster the development and utilization of open-source software. This approach facilitates the expansion of research and education in the field of VLSI design, ultimately augmenting the pool of skilled professionals in this domain.

ETHICAL DECLARATIONS

Referee Evaluation Process

Externally peer-reviewed.

Conflict of Interest Statement

The authors have no conflicts of interest to declare.

Financial Disclosure

The authors declared that this study has received no financial support.

Author Contributions

All of the authors declare that they have all participated in the design, execution, and analysis of the paper, and that they have approved the final version.

REFERENCES

Chupilko, M., Kamkin, A., & Smolov, S. (2021). Survey of open-source flows for digital hardware design. *Ivannikov Memorial Workshop (IVMEM)*, 11-16. doi:10.1109/IVMEM53963.2021.00008

Gaurav, T., Patel, K., & Parekh, R. (2022). RTL to GDSII implementation of RADIX-4 booth multiplier. *Int Confer Nanoelectr Nanophot Nanomater Nanobiosci Nanotechnol (5NANO)*, 1-5. doi:10.1109/5NANO53044.2022.9828885

Purohit, S., Laddha, P., & Parekh, R. (2021). Implementation and physical design of 8/4-bit signed divider. *8th Int Confer Signal Process Integrat Networks (SPIN)*, 829-834. doi:10.1109/SPIN52536.2021.9566020

URL-1. (2022). Yosys open synthesis suite. http://www.clifford.at/yosys (date of access: 03-10-2022).

URL-2. (2022). Odin II. https://docs.verilogtorouting.org/en/latest/odin (date of access: 03-10-2022).

URL-3. (2022). Graywolf. https://github.com/rubund/graywolf (date of access: 03-10-2022)

URL-4. (2022). Qrouter. http://opencircuitdesign.com/qrouter (date of access: 03-10-2022)

URL-5. (2022). OpenSTA engine. https://github.com/The-OpenROAD-Project/OpenSTA (date of access: 03-10-2022).

URL-6. (2022). vesta. https://github.com/RTimothyEdwards/qflow/blob/master/src/vesta.c (date of access: 03-10-2022).

URL-7. (2022). vesta. https://github.com/RTimothyEdwards/qflow/blob/master/src/vesta.c (date of access: 03-10-2022).

URL-8. (2022). Netgen 1.5. http://opencircuitdesign.com/netgen (date of access: 03-10-2022).

Kang, S., & Leblebici, Y. (2005). Introduction. In CMOS digital integrated circuits: analysis and design (Tata McGraw Hill ed., Chapter 1, pp. 1-8).

Cao, L. (2021). Multi-objective digital VLSI design optimisation (doctoral dissertation). University of York.

Aydın, M. (2022). Detection of printed circuit board errors with FPGA-based real-time image processing. İstanbul University-Cerrahpaşa.

Bedir, N. (2018). FPGA based 64-bit arithmetic logic unit design. Thesis (PhD), İstanbul University-Cerrahpaşa.

Başak, S. (2011). FPGA based synthesizable processor design. Thesis (PhD), Yıldız Teknik University.

Sırmaçek, B. (2007. Modeling learning algorithm for mobile robot with FPGA. Thesis (PhD), Yıldız Teknik University.

Bakacak, M. (2021). A basic microprocessor design and embedded system implementation for educational purposes. Thesis (PhD), Kırıkkale University.

Ozcan, M. (2023). Different nanometer technologies on ALU (aritmetic logic unit) design. İstanbul University-Cerrahpaşa.