SEMICONDUCTOR CHIPS DESIGNING PROCESS USING OPEN SOURCE SOFTWARES AND PROCESS DESIGN KITS: SKY130, XSCHEM, MAGIC, NGSPICE

Nguyen Huynh Phuoc Thien[1], Pham The Thinh[1], Duong Khanh Son[1], and Do Vinh Quang[1]

[1]*Can Tho University of Technology*
Email: nhpthien@ctuet.edu.vn

ARTICLE INFO
Received: 25/10/2024
Revised: 17/02/2025
Accepted: 19/02/2025

Keywords: *Magic VLSI, Ngspice, SKY130, Semiconductor chip design, Xschem*

ABSTRACT

The semiconductor industry and related fields are currently attracting significant attention from researchers and businesses. These fields require a highly skilled workforce with strong expertise and capabilities. To meet this demand, companies have developed open-source software to facilitate access to chip design. This study provides a brief overview of the process of designing, simulating, and laying out integrated circuits using open-source software such as SKY130, Xschem, Magic, and Ngspice. Additionally, this study evaluates the advantages and limitations of these tools, thereby demonstrating the usefulness and feasibility of open-source solutions in semiconductor design.

1. INTRODUCTION

In recent years, the semiconductor industry and semiconductor chip design research have gained significant attention (Amuru et al., 2023; Mohammad et al., 2023; Parmar et al., 2023). According to the statistics from the Semiconductor Industry Association, over 100 billion semiconductor chips are used daily (Semiconductor Industry Association, 2024). These chips rank as the fourth most traded goods worldwide, following crude oil, refined oil, and automobiles (Greater Phoenix Economic Council, 2021), contributing to a market value exceeding $500 billion in 2022 (Semiconductor Industry Association, 2024). This makes the semiconductor industry a promising area for scientific exploration and business development.

In chip production, packaging accounts for about 6% of a chip's value, while design contributes over 53%, and manufacturing accounts for 24% (Joyce Beebe, 2022). This highlights significant job opportunities for engineers in the design phase. However, a challenge in chip design is the need for specialized software, which limits access to this field. To address this, companies have developed open-source semiconductor chip design software to make the field more accessible and open new development opportunities (Chai et al., 2023; Bustany et al., 2023).

These open-source tools not only help beginners learn but also assist those with foundational knowledge in practicing and researching. This is expected to further expand the field.

This study explores open-source semiconductor chip design software in the context of a design workflow and examines how these tools interact to complete a semiconductor chip design.

2. OPEN-SOURCE SOFTWARE FOR SEMICONDUCTOR CHIP DESIGN

2.1. SkyWater Open-Source Process Design Kit

A Process Design Kit (PDK) is a dataset used in the semiconductor industry to model fabrication processes for chip design. PDKs are created by chip manufacturers to define specific technologies for their processes. Designers use PDKs to design, simulate, draft, and verify their designs before sending them to the manufacturer for production. These kits are customized to the specific processes of a production facility and are selected early in the design process based on market requirements for the chip.

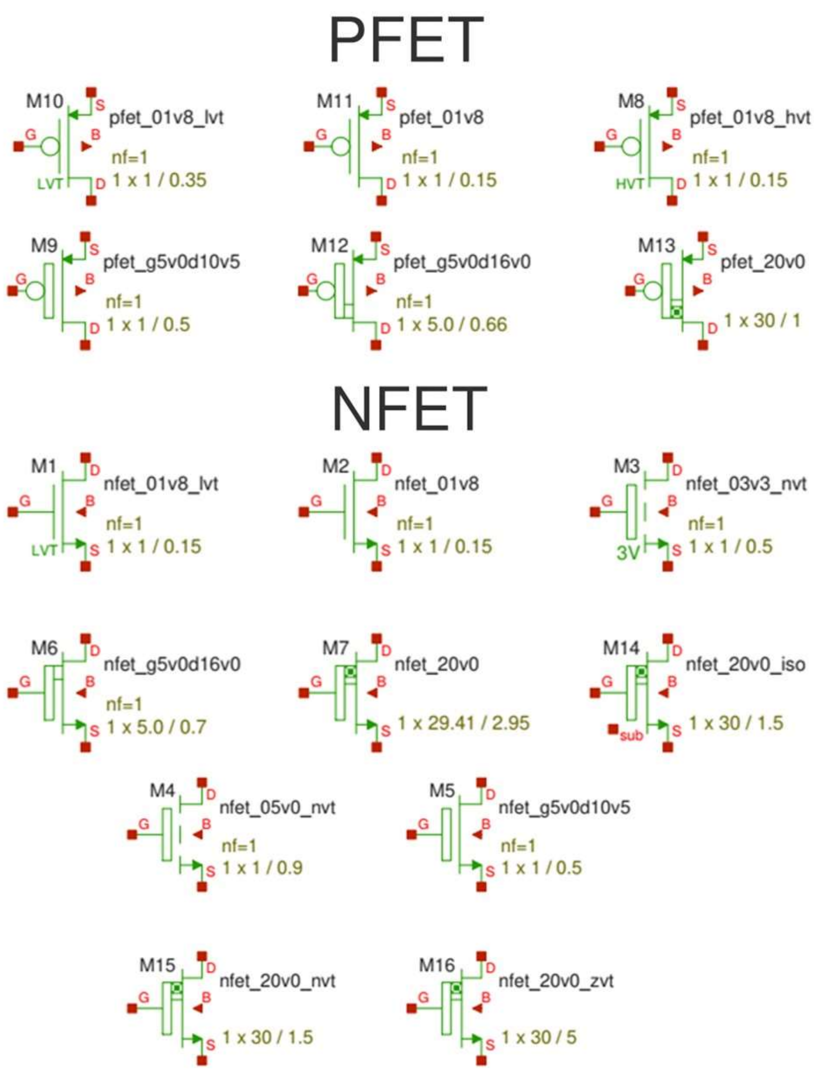


Figure 1. Some components of the SKY130 PDK in Xschem (PFET and NFET)

The SkyWater PDK is an open-source PDK developed in collaboration between Google and SkyWater Technology Foundry. It supports the creation of chip designs that can be manufactured at SkyWater's facilities. Currently, the SkyWater PDK uses the 130 nm process, known as SKY130, and has been widely adopted in various projects (Zhang et al., 2022; Rodriguez-Ferrendez et al., 2023; Cherivirala et al., 2023). If this process is wellutilized, Google and SkyWater Technology Foundry may introduce more advanced processes in the future.

2.2. Xschem

Xschem is a schematic capture and simulation tool that facilitates hierarchical, multi-level circuit design with a top-down approach (Stefan Schippers et al., 2023). Complex systems can be represented by simple blocks focusing on their attributes, hierarchy, and surface-level functionality. Key features of Xschem include its drawing tool written in C, which directly uses Xlib drawings for high performance and speed, even for large and complex circuits. Additionally, it supports simulation using VHDL, Verilog, or Spice and integrates seamlessly with the SkyWater PDK.

Figure 1 and Figure 2 illustrate some components of the SKY130 PDK in Xschem software.

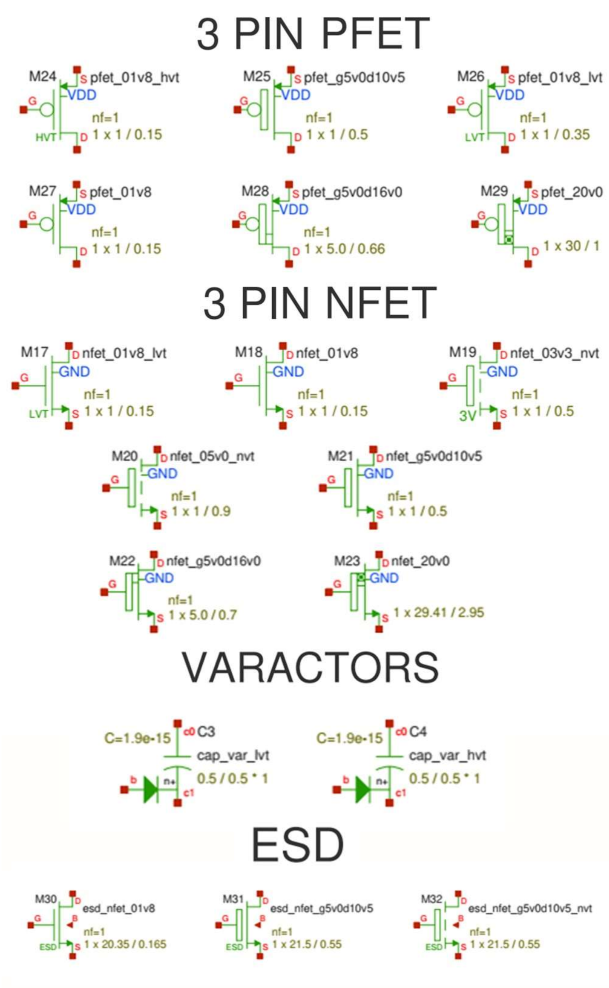
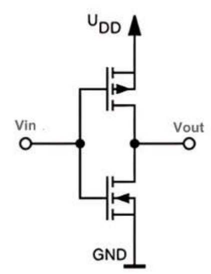


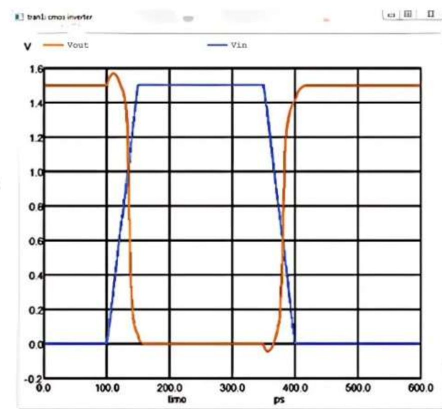
Figure 2. Some components of the SKY130 PDK in Xschem (3 pin FETs and some other component)

2.3. Ngspice

Ngspice (Vogt et al., 2023) is a circuit simulation tool that uses mathematical equations to describe circuits. Its utility has been demonstrated in numerous projects (Lannutti et al., 2012; Dakre et al., 2012). To illustrate how Ngspice works, in Figure 3 we have a CMOS inverter circuit. To simulate this inverter circuit, we have to create a netlist. The netlist is the input to Ngspice, describing to this software the circuit that needs to be simulated.




```
CMOS inverter

.include ./bsim4soi/nmos4p0.mod
.include ./bsim4soi/pmos4p0.mod
.option TEMP=27C

Vpower VD 0 1.5
Vgnd VS 0 0

Vgate Vin VS PULSE(0 1.5 100p 50p 50p 200p 500p)

MN0 Vout Vin VS VS N1 W=10u L=0.18u
MP0 Vout Vin VD VS P1 W=20u L=0.18u

.tran 3p 600ps

.control
  run
  plot Vout Vin
.endc

.END
```



the input

the output

Figure 3. Example of using Ngspice

Source: Vogt, (2023).

To create a netlist, users need to have certain knowledge about circuits and more importantly about Ngspice software. Xschem simplifies this process by generating netlists directly from circuit diagrams.

2.4. Magic

Magic is an open-source layout editor that allows users to create and modify chip layouts. It uses color-coded layers to represent different physical material layers for chip design. Though predominantly used in industry, Magic has also been applied in academic research (Jin et al., 2010; Concepcion et al., 1996). A particularly useful feature of Magic is its ability to convert Xschem-drawn circuit schematics into physical chip layouts, saving significant time and effort for designers.

3. SEMICONDUCTOR CHIP DESIGN WORKFLOW

3.1. Design steps

After installing the above software, users can proceed with the following steps:

1. Draw the circuit schematic using Xschem: Create the initial theoretical representation of the circuit, outlining its components and connections.

2. Simulate the schematic using Ngspice: To simulate the circuit, Ngspice requires a netlist that describes the circuit's components, connections, and their properties. Users can generate this netlist directly from Xschem after completing the schematic design. Once the netlist is ready, it is loaded into Ngspice for simulation. Key configurations include selecting appropriate simulation modes (e.g., transient or AC analysis) and defining voltage/current sources for input. Common issues during simulation include convergence errors, which can often be resolved by fine-tuning simulation parameters or ensuring accurate component values. The results, such as

waveforms or frequency responses, help verify the circuit's functionality and guide further optimization.

3. Convert the schematic into a chip layout using Magic: Develop a detailed layout representing the physical placement of components and connections.

4. Adjust the layout to match the schematic: Ensure the physical design aligns with the theoretical schematic to maintain functionality.

5. Generate the final production file: The production file, typically in GDS (Graphic Data System) format, serves as the blueprint for chip manufacturing. It includes detailed geometric and material layer information for the chip layout. This file ensures that the physical design aligns with the intended schematic and adheres to process design rules. The GDS file also contains metadata such as layer mapping and annotations to assist in the manufacturing process. Once generated, it can be sent to a foundry for fabrication, making it a critical step in turning a digital design into a physical chip.

3.2. Drawing a Schematic with Xschem

To demonstrate, the authors chose to design a NAND gate circuit using 2 PFETs and 2 NFETs (Figure 4). The schematic was created using Xschem, as shown in Figure 5. Once the schematic is correctly drawn with all components and connections, a netlist can be generated automatically by pressing the "Netlist" button at the top-right corner of the Xschem interface. Figure 6 illustrates the netlist generated by Xschem.

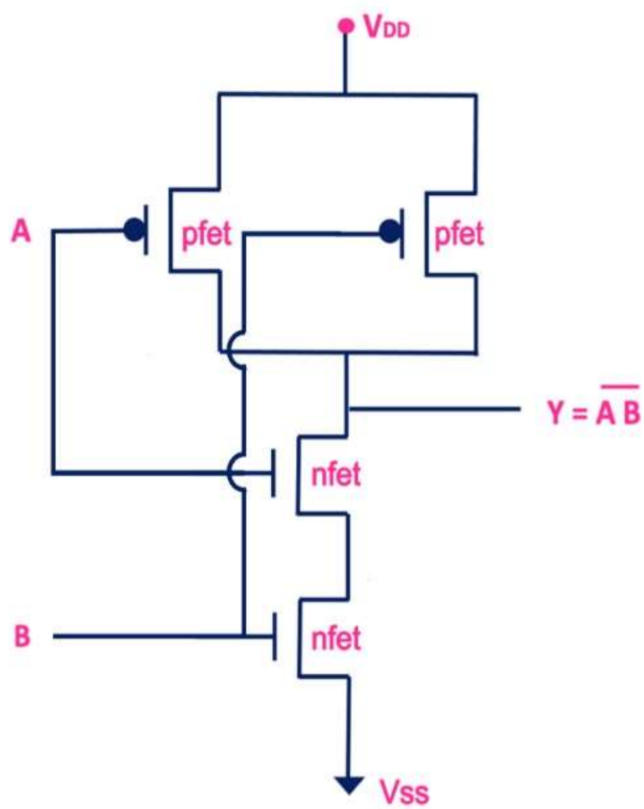


Figure 4. NAND logic gate using 4 FETs

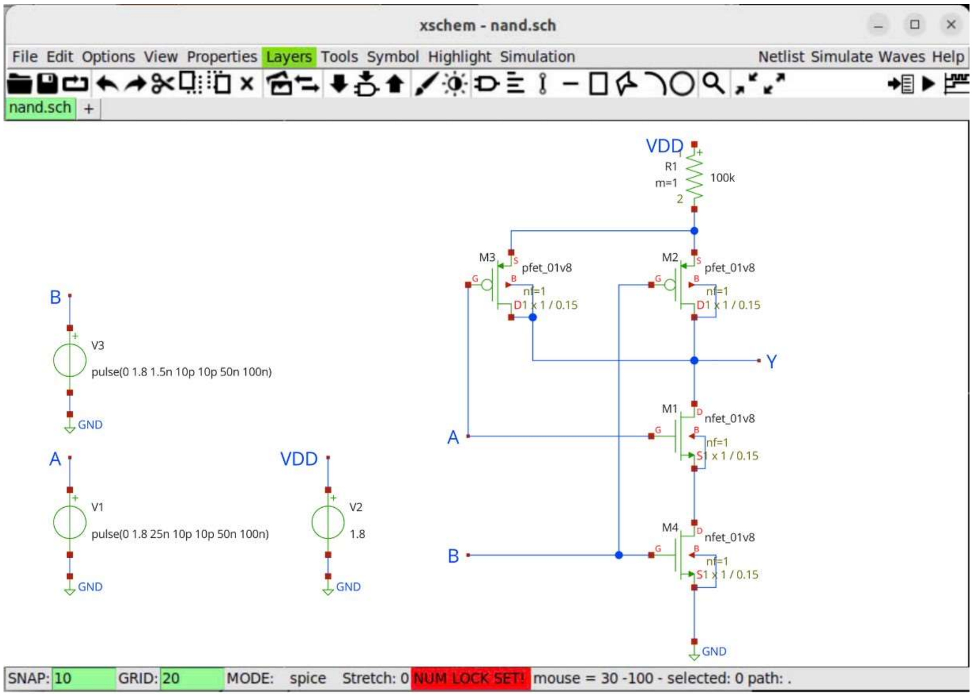


Figure 5. Created schematic using Xschem

```
** sch_path: /home/ttuser/project/nand.sch
**.subckt nand
XM1 Y A net1 net1 sky130_fd_pr__nfet_01v8 L=0.15 W=1 nf=1 ad='int((nf+1)/2) * W/
nf * 0.29' as='int((nf+2)/2) * W/nf * 0.29' pd='2*int((nf+1)/2) * (W/nf + 0.29)'
+ ps='2*int((nf+2)/2) * (W/nf + 0.29)' nrd='0.29 / W' nrs='0.29 / W' sa=0 sb=0 s
d=0 mult=1 m=1
XM2 Y B net2 Y sky130_fd_pr__pfet_01v8 L=0.15 W=1 nf=1 ad='int((nf+1)/2) * W/nf
* 0.29' as='int((nf+2)/2) * W/nf * 0.29' pd='2*int((nf+1)/2) * (W/nf + 0.29)'
+ ps='2*int((nf+2)/2) * (W/nf + 0.29)' nrd='0.29 / W' nrs='0.29 / W' sa=0 sb=0 s
d=0 mult=1 m=1
V1 A GND pulse(0 1.8 25n 10p 10p 50n 100n)
V2 VDD GND 1.8
V3 B GND pulse(0 1.8 1.5n 10p 10p 50n 100n)
XM3 Y A net2 Y sky130_fd_pr__pfet_01v8 L=0.15 W=1 nf=1 ad='int((nf+1)/2) * W/nf
* 0.29' as='int((nf+2)/2) * W/nf * 0.29' pd='2*int((nf+1)/2) * (W/nf + 0.29)'
+ ps='2*int((nf+2)/2) * (W/nf + 0.29)' nrd='0.29 / W' nrs='0.29 / W' sa=0 sb=0 s
d=0 mult=1 m=1
XM4 net1 B GND GND sky130_fd_pr__nfet_01v8 L=0.15 W=1 nf=1 ad='int((nf+1)/2) * W
/nf * 0.29' as='int((nf+2)/2) * W/nf * 0.29' pd='2*int((nf+1)/2) * (W/nf + 0.29)
'
+ ps='2*int((nf+2)/2) * (W/nf + 0.29)' nrd='0.29 / W' nrs='0.29 / W' sa=0 sb=0 s
d=0 mult=1 m=1
R1 VDD net2 100k m=1
**** begin user architecture code

.lib /home/ttuser/pdk/sky130A/libs.tech/ngspice/sky130.lib.spice tt
.tran 0.1n 100n
.save all

**** end user architecture code
**.ends
.GLOBAL GND
.end
```

Figure 6. Generated netlist by Xschem

3.3. Simulating the Schematic with Ngspice

The "Simulation" menu in Xschem allows to configure the necessary settings to run a simulation using Ngspice. Select "Ngspice Interactive" as the simulator, as shown in Figure 7. Clicking the "Simulation" button, the circuit will be simulated, and results such as waveforms will be displayed. For instance, the authors simulated the NAND gate circuit and compared the results with the truth table of the NAND logic gate (Figure 8). The simulation verified that the circuit operated correctly according to the expected truth table outputs.

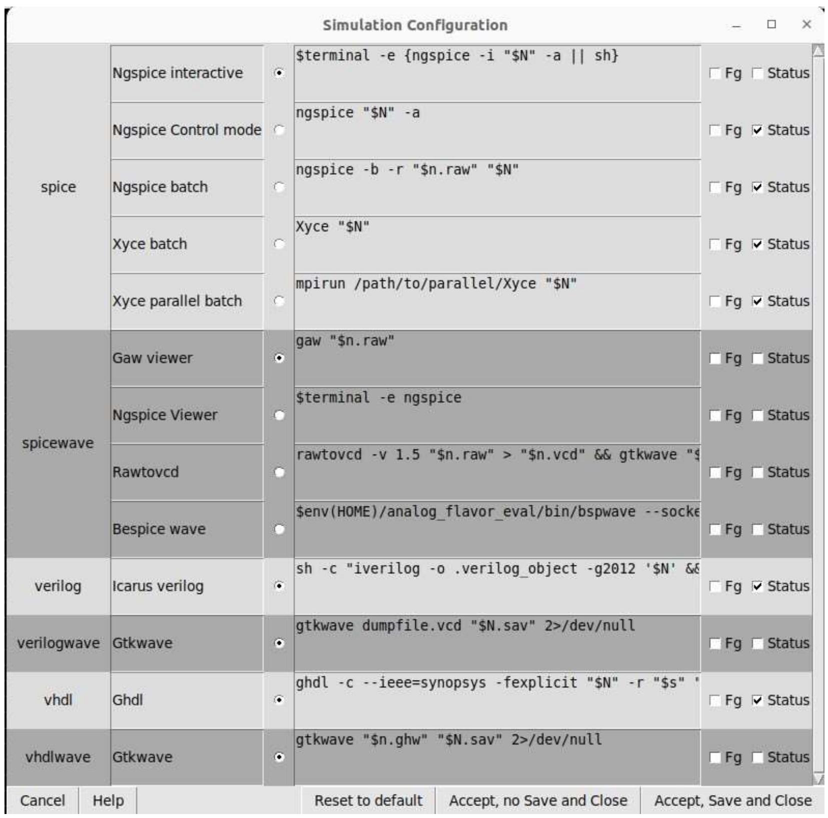
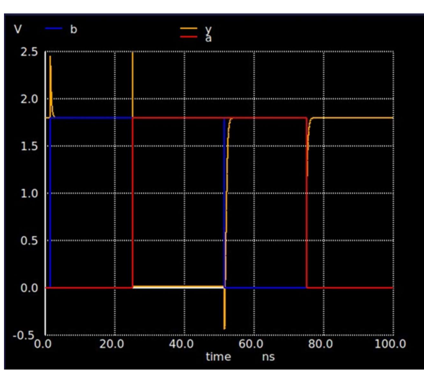


Figure 7. Simulation Configuration window



Input		Output
A	B	Y
0	0	1
1	0	1
0	1	1
1	1	0

Figure 8. Simulation result (left) and the NAND logic gate's truth table (right)

3.4. Converting the schematic to a layout with magic

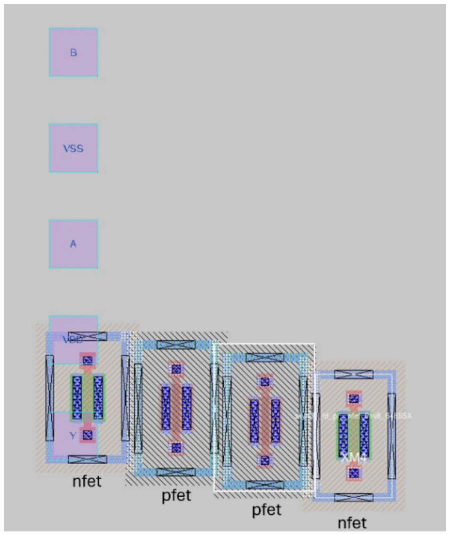


Figure 9. The initial layout generated by Magic

The chip layout is generated by opening the schematic's netlist in Magic. Figure 9 shows the initial layout generated by Magic, including the placement of FETs and interconnects. Users can then arrange the components and draw additional material layers to complete the layout. Figure 10 presents the finalized chip layout, where the authors utilized two distinct material colors representing two layers of the circuit.

3.5. Exporting the final production file

After completing the chip layout, users can export the production file by entering the command "gds write" in Magic's command window. If successful, Magic generates a GDS file, as shown in Figure 11. This file can then be sent to manufacturers, such as Tiny Tapeout, for chip fabrication.

```
% gds write nand.gds
   Generating output for cell sky130_fd_pr__nfet_01v8_648S5X
   Generating output for cell sky130_fd_pr__nfet_01v8_lvt_648S5X
   Generating output for cell sky130_fd_pr__pfet_01v8_XGS3BL
   Generating output for cell sky130_fd_pr__pfet_01v8_lvt_4QXNR3
   Generating output for cell nand
```

Figure 10. Production file exporting command

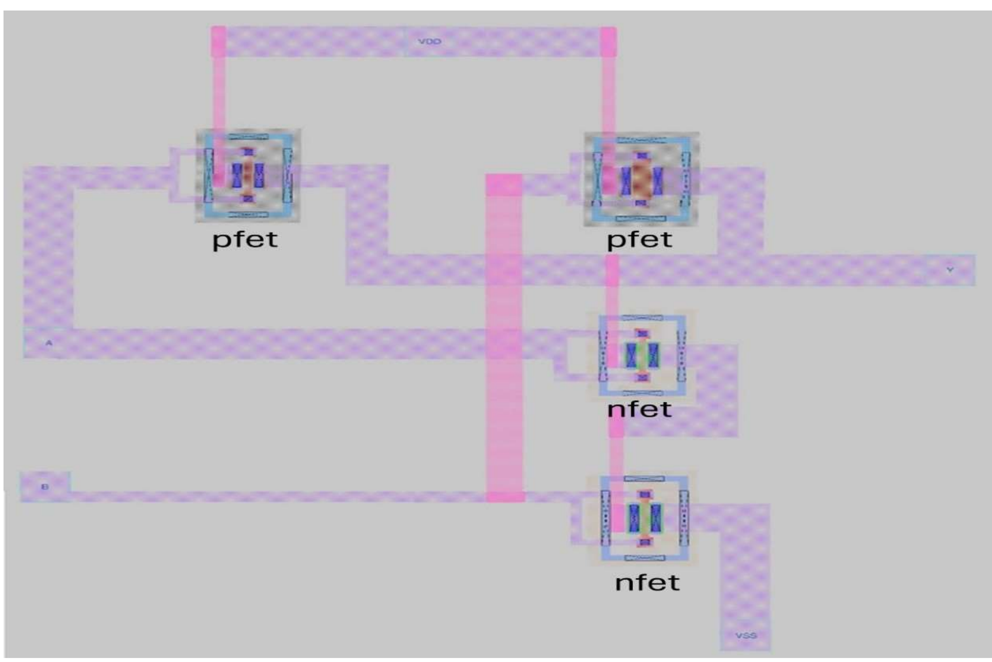


Figure 11. The finalized chip layout

4. CONCLUSION

This study successfully demonstrates the design, simulation, and layout of semiconductor chips using open-source tools such as Xschem, Ngspice, and Magic, along with the SkyWater PDK SKY130. These tools not only make semiconductor design accessible but also provide robust functionality for completing an end-to-end design process. The results highlight the feasibility and effectiveness of these tools in practical applications, including NAND logic gate design and generating GDS-format production files.

However, open-source tools have limitations, such as stability issues and access to advanced PDKs. For high-security and precision-demanding commercial applications, professional-grade software is still the optimal choice.

The results open up many potentials for future development:

- Expanding open-source tools to integrate advanced PDKs (e.g., 65 nm or smaller).

- Optimizing design and simulation processes through automation.

- Applying these tools in emerging fields such as IoT and embedded systems.

- Developing detailed learning resources to support students and beginners.

References

Amuru, D., Zahra, A., Vudumula, H. V., Cherupally, P. K., Gurram, S. R., Ahmad, A., & Abbas, Z., (2023), “AI/ML algorithms and applications in VLSI design and technology”, Integration, 93.

Bustany, Ismail et al, (2023), "An Open-Source Constraints-Driven General Partitioning Multi-Tool for VLSI Physical Design", 2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD), IEEE.

Chai, Z., Zhao, Y., Liu, W., Lin, Y., Wang, R., & Huang, R., (2023), “Circuitnet: An open-source dataset for machine learning in vlsi cad applications with improved domain-specific evaluation metric and learning strategies”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 42(12), 5034-5047.

Cherivirala, Yaswanth K., Mehdi Saligane, & David D. Wentzloff, (2023), "An open source compatible framework to fully autonomous digital ldo generation", 2023 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE.

Concepcion, A. I., & Millican, D. R, (1996), “Developing the VLSI laboratory for the computer architecture course”, In Proceedings of the twenty-seventh SIGCSE technical symposium on Computer science education, 47-52.

Dakre, A., & Chopde, A., (2012), “Design and Simulation of 1-Bit Sigma–Delta ADC Using Ngspice Tool”, International Journal of Advanced Research in Computer Science and Electronics Engineering, 1(2).

Greater Phoenix Economic Council, (2021), Industry Insight Report: Semiconductor.

Joyce Beebe, (2022), Semiconductors: Encouraging Innovation through Manufacturing and Tax Incentives, Issue brief no. 07.20.22. Rice University’s Baker Institute for Public Policy, Houston, Texas.

Jin, L., Liu, C., & Anan, M., (2010), “Open-source VLSI CAD tools: a comparative study”, In Anais: American Society for Engineering Education–2010 Illinois–Indiana Section Conference, West Lafayette.

Lannutti, F., Nenzi, P., & Olivieri, M., (2012), “KLU sparse direct linear solver implementation into NGSPICE", In Proceedings of the 19th International Conference Mixed Design of Integrated Circuits and Systems-MIXDES 2012, 69-73.

Mohammad, Ashif et al, (2023), "AI in VLSI Design Advances and Challenges: Living in the Complex Nature of Integrated Devices", Asian Journal of Mechatronics and Electrical Engineering 2.2, 121-132.

Parmar, Rushik et al, (2023), "Design of DNN-based low-power VLSI architecture to classify atrial fibrillation for wearable devices", IEEE Transactions on Very Large Scale Integration (VLSI) Systems 31.3, 320-330.

Rodriguez-Ferrandez, Ivan et al, (2023),"Space Shuttle: A Test Vehicle for the Reliability of the SkyWater 130nm PDK for Future Space Processors", 2023 IEEE 29th International Symposium on On-Line Testing and Robust System Design (IOLTS), IEEE.

Stefan Schippers et al, (2023), Xschem Manual.

Semiconductor Industry Association, (2024), 2024 State of the U.S. Semiconductor Industry Report.

Vogt, H., Atkinson, G., Nenzi, P., & Warning, D, (2023), Ngspice User's Manual Version 40 plus (ngspice release version).

Zhang, Q., Duan, W., Edwards, T., Ansell, T., Blaauw, D., Sylvester, D., & Saligane, M., (2022), "An open-source and autonomous temperature sensor generator verified with 64 instances in SkyWater 130 nm for comprehensive design space exploration", IEEE Solid-State Circuits Letters, 5, 174-177.

QUÁ TRÌNH THIẾT KẾ VI MẠCH BÁN DẪN SỬ DỤNG CÁC PHẦN MỀM VÀ BỘ THIẾT KẾ QUY TRÌNH MÃ NGUỒN MỞ: SKY130, XSCHEM, MAGIC, NGSPICE

TÓM TẮT

Hiện nay nền công nghiệp vi mạch bán dẫn và các lĩnh vực liên quan đang thu hút nhiều sự quan tâm của các nhà nghiên cứu và cả những doanh nghiệp đầu tư. Từ thiết kế - chế tạo - gia công - đào tạo đều cần sự đáp ứng về nhân lực, đặc biệt là nguồn nhân lực có chuyên môn cao và kỹ năng tốt. Vì vậy, các công ty thiết kế vi mạch bán dẫn đã phát triển các phần mềm mã nguồn mở nhằm giúp những người quan tâm có thể tiếp cận với lĩnh vực này dễ dàng hơn. Nghiên cứu này trình bày ngắn gọn về quá trình sử dụng bao gồm thiết kế, mô phỏng và vẽ vi mạch, trên các phần mềm mã nguồn mở đó. Ngoài ra, các nhận xét về ưu điểm, hạn chế của các phần mềm trên cũng được đưa ra. Qua đó có thể chứng minh tính hữu ích và thực tiễn của các phần mềm mã nguồn mở trong lĩnh vực thiết kế vi mạch bán dẫn

Từ khóa: *Magic VLSI, Ngspice, SKY130, thiết kế vi mạch bán dẫn, Xschem*