Sohaib El-Araby Ali
Digital Desgin Verification Engineer

+201094646362
sohibalaraby9@gmail.com
es-SohaibElaraby2025@alexu.edu.eg
GitHub Profile
LinkedIn Profile

Education

•Faculty Of Engineering Alexandria University, Alexandria *2025-2020*
Communication and Electronics Department CGPA/Percentage: 3.3
High Education

Experience

•MIA Technical Team *2023-2022*
Electronics Engineer for 1 year Alexandria
- I was responsible for creating Python scripts to automate regular software tasks, as well as developing Python machine learning models and computer vision models.
- I was also responsible for designing the entire hardware system of the ROV and all related PCB circuits.
- Additionally, I participated in the MATE ROV regional competition, where our team won 3rd place.

•IEEE SSCS Alex SC *2025-2024*
Digital IC Member Alexandria
- Currently, I am a part of the IEEE SSCS Alex SC in the Digital IC subteam.
- I am training under the supervision of highly skilled Digital IC Design, Verification, and ASIC engineers who have years of experience in the industry.

Technical Projects

•Adder Verification Based on UVM
In SystemVerilog, UVM
- Tools & technologies used: Questasim, EDA Playgroun, UVM
- In this project, I applied most of the OOP concepts I had learned and utilized the UVM structure. I transitioned from design to verification, building a comprehensive UVM testbench with a focus on reusability. Additionally, I developed a coverage model to trace coverage, ultimately achieving 100% coverage.

•ALU Verification Based on UVM
In SystemVerilog, UVM
- Tools & technologies used: Questasim, EDA Playgroun, UVM
- In this project, I applied most of the OOP concepts and utilized the UVM structure and verification principles. I built a complete UVM environment for testing ALU functionality, including all UVM structural components and a coverage collector to address all corner cases, ultimately achieving 100% coverage.

•UART Verification Based on UVM
In SystemVerilog, UVM
- Tools & technologies used: Questasim, EDA Playgroun, UVM
- In this project, I built a complete UVM environment to verify the two main components of a UART protocol design: the transmitter and the receiver. I identified a bug in the receiver that prevented it from receiving data correctly, so I modified the receiver's FSM to fix the issue. After retesting, it worked correctly, ultimately achieving 99.4% coverage.

•Single Cycle 32-bit RISC V (RV32I) Microprocessor Integrated With Cache Memory Design
In SystemVerilog
- Tools & technologies used: Questasim from Mentor Graphics
- In this project, I studied the instruction set architecture of RISC-V, designed an architecture to support all integer instructions, developed RTL code for the RV32I architecture, integrated it with cache memory in SystemVerilog, and verified it through direct testing.

•Single Cycle 32-bits MIPS Microprocessor Design
In Verilog
- Tools & technologies used: Modelsim
- In this project, I have built all the blocks necessary for MIPS functionality and verified each block through direct testing by building a test bench for each block to ensure they meet the specifications. Finally, I built a test bench for the entire design and tested it using two programs, obtaining correct outputs.

•UART Design

In SystemVerilog

- Tools & technologies used: Questasim, EDA Playgroun
- UART (Universal Asynchronous Receiver/Transmitter) is a hardware communication protocol used for asynchronous serial communication between devices. In this project, I built the transmitter and receiver, then built a test bench for each of them. Afterward, I connected them together and built a test bench to verify the complete system.

•Circular FIFO Memory Design

In SystemVerilog

- Tools & technologies used: Questasim from Mentor Graphics
- FIFO is a very important memory type used in many applications like UART. In this project, I built a circular FIFO with two variables to keep track of the head and the tail of the FIFO. If the head or tail reaches the end of the memory, it returns to the beginning and vice versa. There are also two variables that monitor the FIFO status: if the FIFO is empty, it prevents any read operation, and if it is full, it prevents writing to it. I also built a test bench to verify the design through direct testing.

Technical Skills and Interests

Languages: Arabic (Native Speaker), English (Very Good)
Developer Tools: Modelsim, Questasim, EDA-playground
HDL: Verilog, Systemverilog, VHDL
Verification Methodology: UVM
Related Topics: CDC, STA, OOP, MIPs Architecture, RISC V Architecture
Soft Skills: Team Work, Communication Skills, Passion For Learning, Time Management
Operating Systems: Linux Operating System
Programming Languages: C Programming, Python
Areas of Interest: Design & Verification Of Digital Integrated Circuits