Project Report: "Readboard: Scroll-It“

Date: June 8, 2025
Name: AMANALIYEVA SHYNAIYM
Student num: 24101212
Major: Computer Engineering
Date: 2025.06.08
Professor: 최성록
Course Name: Digital Logic and System

Problems and Solutions

Misunderstanding and INC_8 Implementation: At the beginning, it seemed difficult to develop the INC_8 module as well as comprehend its inner workings. Solution: I followed the modular route. Firstly, I designed a subcircuit for INC_1 with the help of logic gates (XOR, AND gate) to form one-bit full adder with Cin (carry-in), A0 (bit input), Cout (carry-out), and F0 (bit output) outputs. Subsequently, taking the pre-designed and debugged INC_1 subcircuit as the basis, I designed the INC_8 subcircuit with eight copies of INC_1 cascaded one after another, routing the Cout from one INC_1 as the Cin to the subsequent one. This facilitated the proper execution of an 8-bit incrementer.

Uncontrolled LED Matrix Output: Text was initially scrolling continuously on the LED matrix, iterating through all ROM content instead of discrete character output. Solution: Analysis revealed the issue was incorrect control logic. Instead of a register to hold display data, I mistakenly used a Counter in the output loop. I corrected this by replacing the Counter with a proper register (D-Flip-Flop) to store the current ASCII code. Furthermore, I made necessary adjustments to the INC_8 logic and its connection with the keyboard, which enabled control over the output and stopped the continuous scrolling.

Mirrored Text Display: Text appeared mirrored on both sides of the matrix (left and right). Solution: The cause was incorrect mapping of data bits from the ROM to the physical pixel arrangement on the LED matrix. I rectified this by carefully checking and adjusting the connection order of the 8-bit ROM outputs to the inputs of the registers controlling the matrix, ensuring proper bit-to-pixel correspondence for correct character display.

7-bit ASCII to 8-bit Conversion: The keyboard outputted 7-bit ASCII code, but an 8-bit format was required for further processing and ROM addressing. Solution: I designed the ASCII subcircuit to directly accept a 7-bit input from the keyboard. Inside this subcircuit, I used a Bit Extender, configured to extend the 7-bit input to an 8-bit output by adding a leading zero (zero-extend). This ensured the correct data format conversion for subsequent circuit blocks.

Knowledge Gained

Deep understanding of Registers and Counters: Learned not only to use standard components but also to create complex functional blocks, such as multi-bit incrementers, and correctly apply them as counters and registers.

Modular Design: Practically mastered the principles of building complex circuits from simpler subcircuits (INC_1 -> INC_8).

Memory Architecture Fundamentals: Understood ROM operation, data storage, and addressing principles.

Digital Circuit Synchronization: Realized the critical importance of clock signals and their

proper organization for stable sequential circuit operation.
Debugging Methodology: Acquired systematic troubleshooting skills for complex digital circuits.
Proficiency in Logisim Evolution: Significantly improved my skills in using Logisim Evolution, mastering subcircuit creation and component usage.

My Feelings and Future Plans

The "Readboard: Scroll-It" project was incredibly captivating and inspiring. Seeing the immediate results of my work in simulation brought immense satisfaction. This experience showed me that computer engineering is not just about coding, but also about fascinating hardware design. I've developed an interest in further exploring digital circuit design and computer architecture.

Expected Project Score

Given that I successfully overcame most challenges and implemented all core project functionalities in Logisim Evolution, I expect to receive an excellent grade". I submitted all the required materials, including the demo video, short report, poster, and circuit file. While I believe my project is complete and shows some creativity—especially in the poster design—I think the overall difficulty and originality are not very high compared to other students' work, as I still have limited experience in digital circuit design.
And this project was completed individually. I was solely responsible for all development stages.

Use of External Resources / Open Sources

The primary conceptual reference for implementing the ASCII character display system was a YouTube video tutorial: https://youtu.be/fQ8HK4YcWdY?si=L7P6tLXatFeoFJc-
But I independently solved numerous debugging, adaptation, and specific ROM addressing/display control challenges. Specifically, I developed a modular structure for the incrementer (INC_1 -> INC_8), implemented precise 7-bit to 8-bit data conversion using a Bit Extender within the ASCII subcircuit, and independently debugged the keyboard synchronization logic and resolved mirroring issues, which required a deep understanding of bit addressing and register control. These modifications and independent problem-solving allowed me to thoroughly grasp the material and create a fully functional circuit.