NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

School of Electrical Engineering and Computer Sciences

Microprocessor Systems (EE-222)
ASSIGNMENT No. 1

ASSIGNMENT- TITLE: *4 Bit Number Crunching Machine*

SUBMITTED TO: *Dr. Abid Rafique*

CLASS + SECTION: *BEE-13A*

SUBMISSION DATE: *22/02/2023*

SUBMITTED BY:

1. Muhammad Sami Ullah - 369138

2. Zohaib Irfan - 372692

4-bit Number Crunching Machine

Introduction:

In this project, our team was assigned the task of designing a 4-bit microprocessor system that involved building an Arithmetic Logic Unit (ALU) using a 4-bit binary adder IC and utilizing registers, multiplexers, and decoders to enable customized inputs and perform a range of operations. One of the key achievements of our project was successfully implementing the Fibonacci series program, which required careful consideration of the logic and sequence of operations to produce the desired output. However, throughout the project, we encountered several challenges, including designing efficient code and ensuring all components were correctly integrated. This project provided us with valuable knowledge and skills in digital logic design and programming, as well as a deeper understanding of microprocessors and computer systems. We are proud of the work we have accomplished and confident in our understanding of the code and hardware design involved.

Procedure:

Our approach involved building an Arithmetic Logic Unit (ALU) using a 4-bit binary adder IC (7483). Two inputs, A and B, were provided, and both were stored in separate registers, labeled RA and RB.

The B input was modified by passing it through an XOR gate, which compared it to a selection pin value. This allowed us to switch between adding and subtracting values. When the selection pin was set to 0, the adder would simply add the values stored in RA and RB. When it was set to 1, B was 2's complemented, and the adder produced the difference between the two values, including the carry-in bit. The selection pin value was set to the same value as the carry-in, to ensure correct operation.

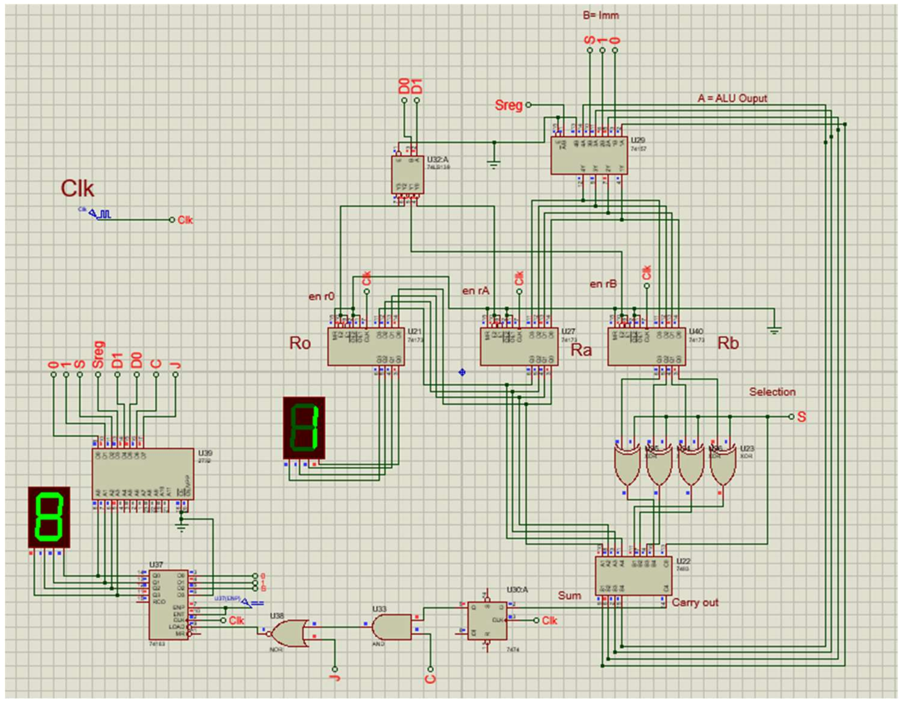
To enable customized inputs, we utilized three registers, all labeled 7495. RA and RB were used to store the two inputs, while the ALU performed arithmetic operations on their contents. The third register, R0, stored the final result. We used a 2-to-1 multiplexer to switch between RA and RB inputs based on the selected register's enable pin value, which was activated by a decoder. The decoder assigned addresses to each of the three registers separately, and the registers were enabled or disabled based on these addresses.

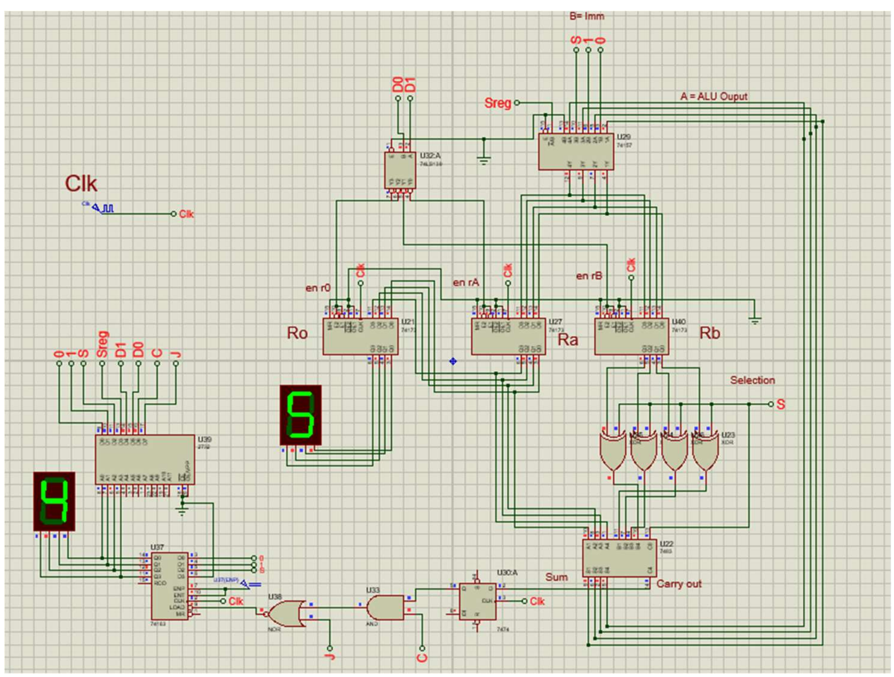
Our microprocessor system was capable of performing a wide range of operations, including addition, subtraction, swapping of numbers, jump conditions, and conditional jumps, among others. We tested the system by implementing the swapping function as specified in the assignment, and it performed successfully.

We also successfully implemented the Fibonacci series program, which was stored in an EEPROM. The program consisted of 8 lines of code, and the program counter (74163) counted from 0000 to 1000, executing the code stored in the EEPROM. The eight bits of each row of the EEPROM controlled the eight control signals of our microprocessor, enabling it to carry out the required operations.

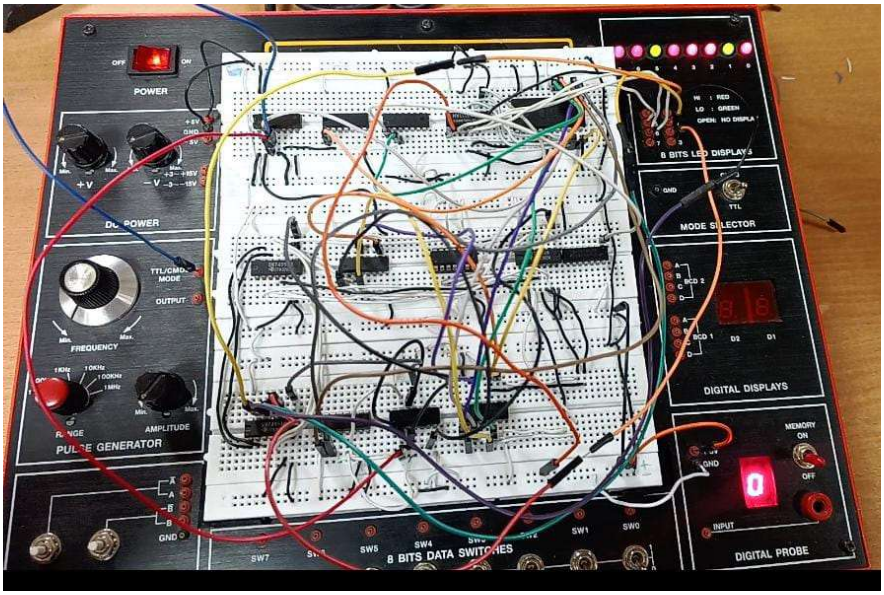
We have provided videos and images of our microprocessor system running the Fibonacci series program. We are confident in our understanding of the code and hardware design involved.

Proteus Simulation:





Hardware Implementation:



Challenges:

In addition to the technical details of our microprocessor design, we encountered several challenges along the way. One of the major challenges was designing the code for the Fibonacci series program, as it required us to carefully consider the logic and sequence of operations to produce the desired output. We started by examining the program requirements and breaking it down into smaller steps. We began by initializing the first two values of the Fibonacci series to 1, which were stored in RA and RB. We then used conditional jumps to determine when to add the two values and when to move to the next iteration. This involved comparing the current iteration count to the target iteration, and branching to the appropriate code block accordingly.

We also encountered some difficulties in implementing the conditional jumps, as it required us to carefully consider the conditions and ensure that the program counter was updated appropriately. We tested the program multiple times to ensure that it produced the correct output and met the program requirements.

Overall, we found the process of designing and building our microprocessor to be both challenging and rewarding. It allowed us to apply our knowledge of digital logic design and programming, and gave us a deeper understanding of the inner workings of microprocessors and computer systems.

Conclusion:

Through this project, we also gained experience in working with hardware components such as adders, registers, multiplexers, and decoders. We had to ensure that all the components were correctly wired and integrated to produce the desired functionality. We also had to troubleshoot and identify any issues that arose during testing and debugging, which required us to have a good understanding of the hardware components and their interconnections.

In addition, the project provided us with a better understanding of the importance of efficient programming and design practices. We had to carefully plan and optimize our code to ensure that it ran efficiently and produced the correct output. We also had to consider the limitations of our 4-bit microprocessor, such as the limited range of numbers it could handle, and design our program accordingly.

In conclusion, our microprocessor project was a challenging yet rewarding experience that allowed us to apply our knowledge of digital logic design and programming to build a functional microprocessor system. We gained valuable experience in working with hardware components and programming languages, and developed a deeper understanding of the inner workings of microprocessors and computer systems. The successful implementation of the Fibonacci series program demonstrates our understanding of the code and hardware design involved, and we are proud of the work we have accomplished.