Digital Logic Design

Lab Project



**Mini Computational Logic Unit**

Submitted by:

|  |  |
| --- | --- |
| Muhammad Ahmad Raza  Maira Izhar | 22L-6700  22L-6626 |

Submitted to:

Professor Ahmad Hamza

May 11, 2023

Department of Electrical Engineering

National University of Computer and Emerging Sciences, Lahore

## Introduction

1. **Problem Analysis**

## Design Requirements

1. **Feasibility Analysis**

## Possible Solutions

1. **Preliminary Design**

## Design Description

## Software Simulation

## Experimental Results

1. **Performance Analysis**
2. **Conclusion References**

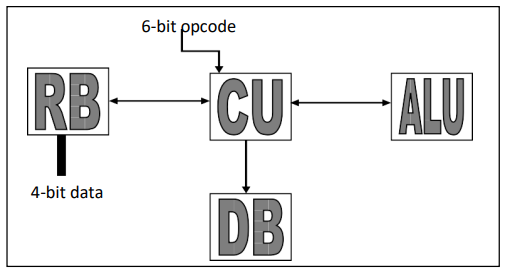
We were assigned to develop a Mini Computational Unit that would receive a unique opcode to perform specific functions. The project was based on the concepts of logic gates, decoders, registers, computational blocks like adder and subtractor. The project also required a good knowledge of software LogicWorks 5 developed by DesignWorks Solutions Inc. LogicWorks 5 for Windows is the leading schematic drawing and interactive digital simulation package setting the standard for demonstrating logic design principles and practices within the education sector and industry. Apart from this, the project demanded a good grip on basic logical concepts of the course Digital Logic Design. The main idea of the project was to design a mini computational unit that would perform specific tasks after receiving opcode provided by the user. The unit would have to store values and perform tasks like shifting bits within registers and addition and subtraction of values.

The mini computational unit had to be designed in a way that it would perform three major functions; storing input values from the user into registers, performing left and right shift on specific registers and adding or subtracting values of two registers and storing them in another register. The unit would work on opcode provided by the user and show output via LED lights specified for each register. Other than that, the unit also required a specific register for indicating certain things being computed. This could be designed with the help of decoders, registers, adder/subtractor blocks and different logic gates. The unit would run on a single clock that would control the working of all four registers of the computational unit. Overall, the unit can be divided into four parts; the memory block that would be comprised of all four registers, an arithmetic logic unit that would handle all operations like addition, subtraction and shifting of bits, a display box that would display the values stored and finally a control unit that would control the operations and send out signals to all other parts.

The unit requires four registers, three 4-to-16-bit decoders, one 2-to-4-bit decoder, two adder/subtractor blocks, OR gates, AND gates, NOT gates (inverters), binary clock, binary input switches, LED lights to display what values each register is holding at each moment and wires to connect all parts. The opcode would comprise of 6 unique bits, out of which the first two Most Significant Bits (MSBs) would decide what function needs to be performed. The other four bits of opcode would decide what specific task needs to be performed within the main function. All registers would work on a single binary clock input and perform tasks at runtime. The unit would require the user to provide another four-bit input that would be needed to initially store values in the registers. The display box would be displaying all values of each register at runtime.

The project did not require any kind of cost management as the only thing needed for this was the LogicWorks 5 software which was provided earlier by the institute. Other than that, the project required a substantial amount of time and knowledge of different concepts being used. The basic reason for the project requiring time was its complexity at several points throughout the journey of its completion. But, nevertheless, the project was completed earlier than due date which allowed us to test its functionality several times with different inputs and provided us the chance to check if there were errors in it.

The project could have been completed by using different strategies and approaches. The best solution among all others seemed to design the computational unit with the help of decoders and other computational blocks like adder/subtractor block. Other than using decoders, the unit could have been constructed by using Multiplexers instead of decoders but that would have required a lot more complex logic and working. The design of the unit with decoders involved is easier to understand and develop. The concepts and working of decoders are easier to get and perceive. Hence, our choice was to use decoders for designing the unit.



The basic idea was to divide the computational unit into 4 parts namely:

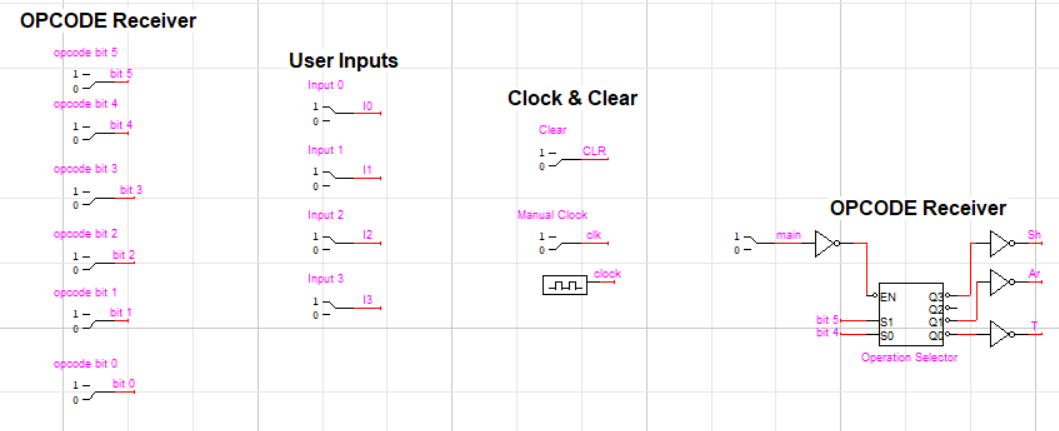
* Control Unit (CU)
* Arithmetic and Logic Unit (ALU)
* Register Box (RB)
* Display Box (DB)

The Control Unit would receive 6-bit opcode from user and control all parts of the unit. It would decide what operation is to be performed. The Register Box would also need 4-bit input from user to initially store some values into the registers before it can perform any other computational function like addition, subtraction or shifting of bits to left and right, all within the registers. The Display Box would simply display the values that are being held by the registers via different colored LEDs.

The details of each part of the unit are as follows:

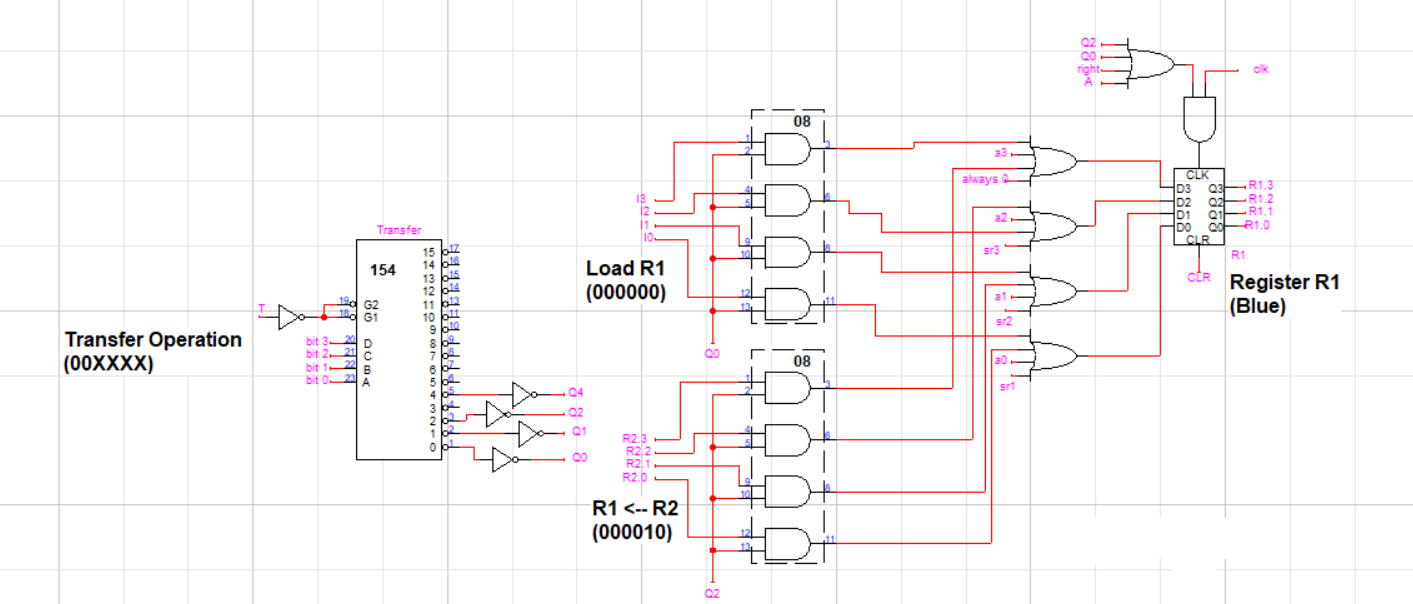
**Control Unit (CU):**

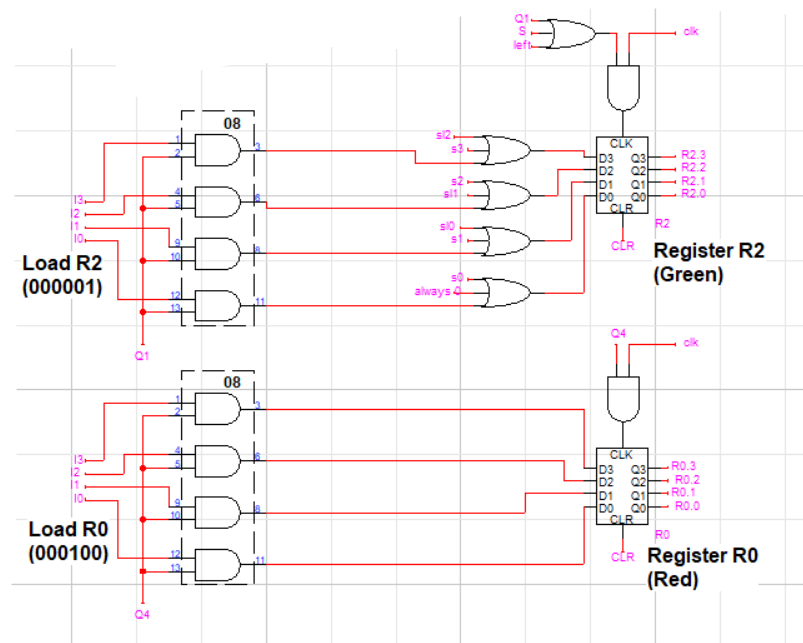
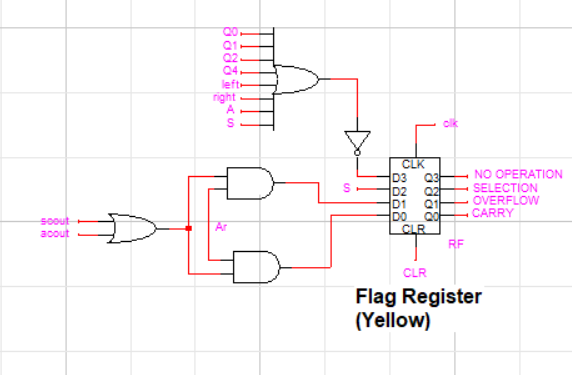
The control unit is comprised of input switches and decoders to control what part of the computational unit is on. Six switches are for the opcode while another four are input switches for the user to enter initial values of the register. A clear/reset switch is for clearing out previously stored values in all registers at once. The clock is also a binary switch for ease and to gain full control or each clock cycle. The clock can be changed to automatic binary clock at any time by changing a few names. The first 2-to-4 decoder controls the other 4-to-16 decoders. Each 4-to-16 decoders works as an enable for every specific task according to the opcode provided by the user.



**Register Block (RB):**

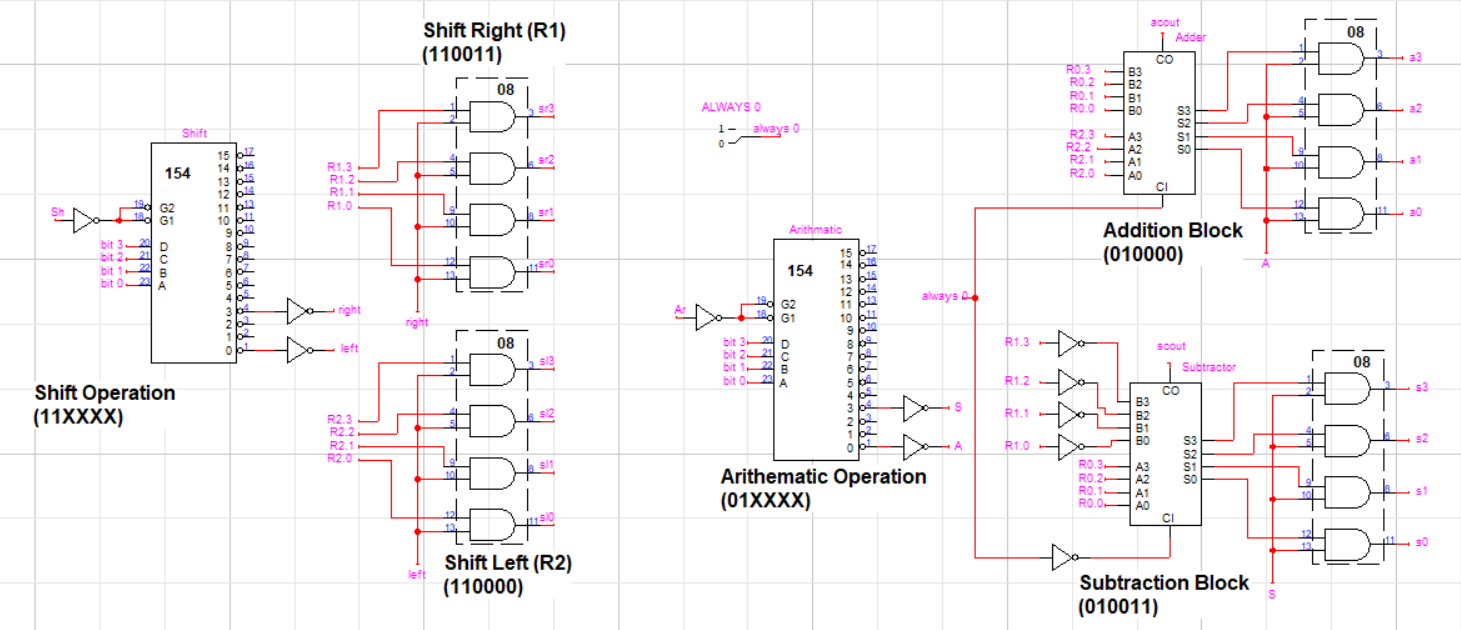
The register block is comprised of four registers which work on the same clock cycle. Three of the registers are for user to store values and perform different operations on them. The fourth register is named as a flag register which is set to display specific values as required and mentioned in the project description/manual.





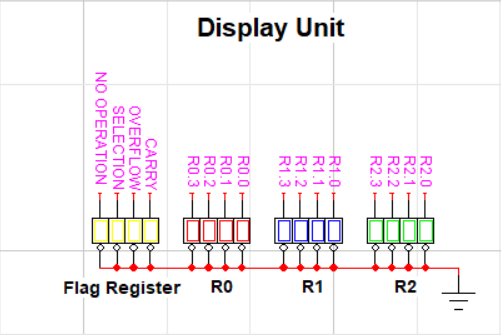
**Arithmetic and Logic Unit (ALU):**

This part of the unit consists of all the gates and computational blocks required for different tasks. This specific part is responsible for performing bit transfers and inputs for registers, additions, subtractions and shifting bits (left and right). This unit is further categorized into three main parts, each part for each specific function (Transfer, Shift and Arithmetic Operations). Each sub-part has its own gates and computational blocks for specific tasks and functions, all of which are powered by the output from one of the three 4-to-16 decoder of the CU.



**Display Box (DB):**

This part of the unit is set up to check what value each of the four register is storing at runtime. The unit has 16 LEDs which indicate specific output bits of each of the four registers.



Software simulation of the project was as follows:

1. The design of the unit was made on the software by using various concepts and logics, one part of the computational unit at a time.
2. Each part was individually tested by comparing its results with correct on-paper working, before combining it with the other parts.
3. Final design/version of the unit was designed on the software and tested multiple time to check for errors.
4. A few software and design issues were noticed and solved upon testing the final design/version of the unit.
5. Final results were checked and compared to correct on-paper working of the unit. The results matched and the unit designed on the software showed accurate results and worked efficiently.

Initial experimentation showed accurate results for the transfer and arithmetic functions but the shift function was not working. Upon further testing and investigation, it was found that the issue was not in the design of unit, but in the automatic binary clock. The clock kept changing from 1 to 0 and vice versa which caused the shifting bits part of the unit to continue shifting bits (left or right) until all values of the register were shifted. This issue was solved by introducing a binary switch which would work as a clock that can be controlled by the user/operator. The registers work only when the clock gets a positive edge. Automatic clock keeps getting positive and negative edges automatically in a fraction of time, hence making the registers work by themselves in shifting bits part. The binary switch working as a clock allows user/operator to control when the clock and registers work by providing positive and negative edges when the switch flips.

The design performs each task as it is supposed to. The earlier versions of the design had a few flaws which were tested time and again and fixed in the final design. The final version/design of the computational unit performs all functions according to the manual/description provided. No errors were detected. All parts of the computational unit work accurately and efficiently as originally intended.

The project was a success. The results of the final design/version of the computational unit were tested again to rule out all possibilities of any errors. Overall, working on this project was fun and challenging at the same time. The project helped in learning many different concepts of the course which were not known earlier. The Mini Computational Unit is ready and fully functional to perform all tasks it is made to perform with highest efficiency and solid, free-of-error results.

Software (LogicWorks 5) used:

https://www.designworkssolutions.com/

***Note:***

*All images used in this report are original. Only the authors of this report and Department of Electrical Engineering, National University of Computer and Emerging Sciences, Lahore have the authority to claim the rights to information and images provided above.*