**KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY**

COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRICAL/ ELECTRONIC ENGINEERING



PROJECT: DESIGN OF AN 8 BIT MICROPROCESSOR

The Design Team

1. Michael Senanu Anku 20531136 9310817
2. Adrian Fiifi Wereko Cole 20557211 9313217
3. Cecil Manford 20521165 9315917
4. Alfred Kusi Adjei-Mensah 20544889 9308817
5. Emmanuel Amoaku 20508452 9310517
6. Quinton Smith-Assan 20515740 9319917
7. Oswald Sena Donya 20506436 9314117

## ACKNOWLEDGEMENT

We would like to thank the Almighty God for being our guide and protector throughout the project sessions. We would also our gratitude to Dr. Jephthah Yankey for giving us this challenging but mind opening project.

Contents

[ACKNOWLEDGEMENT 2](#_Toc22507314)

[Our Microprocessor 4](#_Toc22507315)

[How a Microprocessor Works 5](#_Toc22507316)

[Opcodes for our microprocessor 5](#_Toc22507317)

[**Team designed ALU** 6](#_Toc22507318)

[Operation of our ALU 6](#_Toc22507319)

[Team Designed 8 bit Adder 7](#_Toc22507320)

[Team designed subtractor 7](#_Toc22507321)

[Team designed Multiplier 8](#_Toc22507322)

[Team Designed 8-bit RAM 8](#_Toc22507323)

[Operation of the RAM 9](#_Toc22507324)

[Team designed Control Unit 9](#_Toc22507325)

[Operation of the Control Unit 9](#_Toc22507326)

## Our Microprocessor

The design team, being the members of this group, decided to build an 8 bit Microprocessor. Our design is relatively simple. We set out design a microprocessor that could do the four basic mathematical task. Them being addition, subtraction, multiplication and division. We ( The Design Team) set out to do some really extraordinary task, however, our level expertise could not permit us to put those extra thoughts into fruition.

Our microprocessor has some of the following features;

1. **Arithmetic Logic Unit**

An arithmetic logic unit (ALU) is a digital circuit used to perform arithmetic and logic operations. It represents the fundamental building block of the central processing unit (CPU) of a computer. Modern CPUs contain very powerful and complex ALUs. In addition to ALUs, modern CPUs contain a control unit (CU).

Most of the operations of a CPU are performed by one or more ALUs, which load data from input registers. A register is a small amount of storage available as part of a CPU. The control unit tells the ALU what operation to perform on that data and the ALU stores the result in an output register. The control unit moves the data between these registers, the ALU, and memory.

1. **Registers**

A register may hold an instruction, a storage address, or any kind of data (such as a bit sequence or individual characters). Some instructions specify registers as part of the instruction. For example, an instruction may specify that the contents of two defined registers be added together and then placed in a specified register.

A register must be large enough to hold an instruction - for example, in a 64-bit computer, a register must be 64 bits in length.

1. **Random Access Memory (R.A.M)**

Contributor(s): Kevin Ferguson, Stacey Peterson and Rodney Brown

RAM (Random Access Memory) is the hardware in a computing device where the operating system (OS), application programs and data in current use are kept so they can be quickly reached by the device's processor. RAM is the main memory in a computer, and it is much faster to read from and write to than other kinds of storage, such as a hard disk drive (HDD), solid-state drive (SSD) or optical drive.

Random Access Memory is volatile. That means data is retained in RAM as long as the computer is on, but it is lost when the computer is turned off.

1. **Control Unit**

## How a Microprocessor Works

The operation of a microprocessor, though abstracted, is relatively simple to follow. The Microprocessor, can be somehow likened to the brain of a human being. As so, it is often known as the Brain of the computer.

They work on based on digital logics. Every instruction is executed by a series of logic circuits that work on the principles of the different types of logic gates.

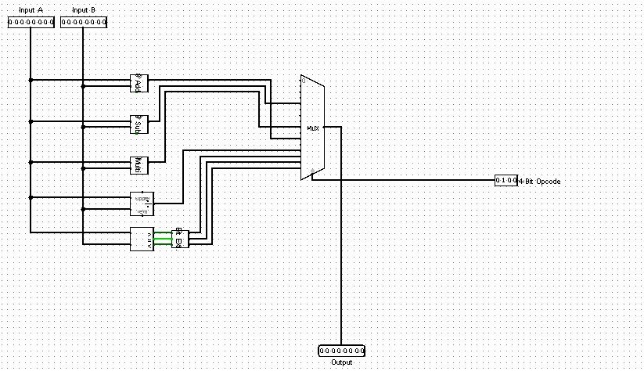
A microprocessor executes a collection of machine instructions that tell the processor what to do. Based on the instructions, a microprocessor does three basic things:

* Using its ALU (Arithmetic/Logic Unit), a microprocessor can perform mathematical operations like addition, subtraction, multiplication and division. Modern microprocessors contain complete floating point processors that can perform extremely sophisticated operations on large floating point numbers.
* A microprocessor can move data from one memory location to another.
* A microprocessor can make decisions and jump to a new set of instructions based on those decisions.

## Opcodes for our microprocessor

|  |  |
| --- | --- |
| **Opcode** | **Instruction** |
| 1010 | Addition |
| 0100 | Subtraction |
| 1000 | Multiplication |
| 1100 | Division |
| 1101 | Greater than |
| 1110 | Equal to |
| 1111 | Less than |
|  |  |

## **Team designed ALU**

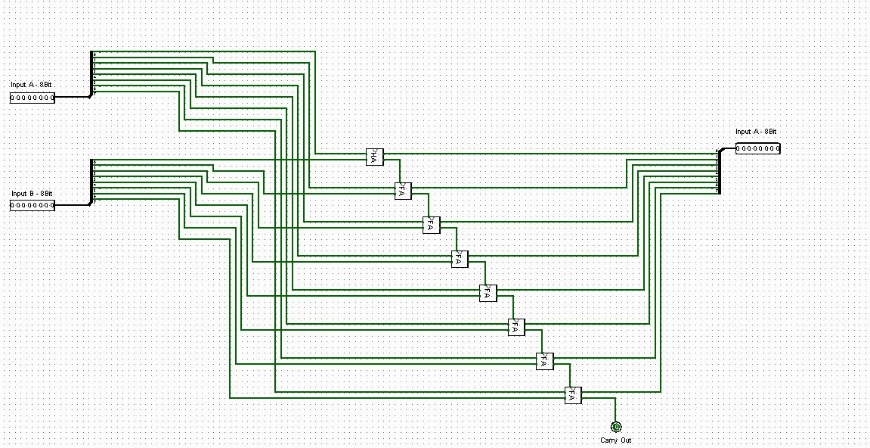


The operation of our team designed ALU is relatively simple. It was designed and modelled using Logisim. We used higher levels of abstraction to represent the addition logic circuits, subtraction logic circuits, multiplication logic circuits. All in the order of top to bottom. These are all connected to an 8 bit multiplexer, with the results of the operations shown on the right side of the multiplexer.

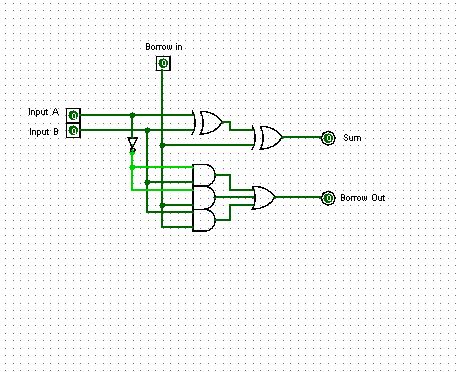
## Operation of our ALU

Two inputs A and B are connected to the abstracted operation logic circuits. The operation to be executed is manually input using the opcode, and adjusted using Logisim’s interface. Once an opcode is selected, the ALU executes that particular instruction. The result of the operation is sent into the multiplexer and from there connected to our 8 bit register.

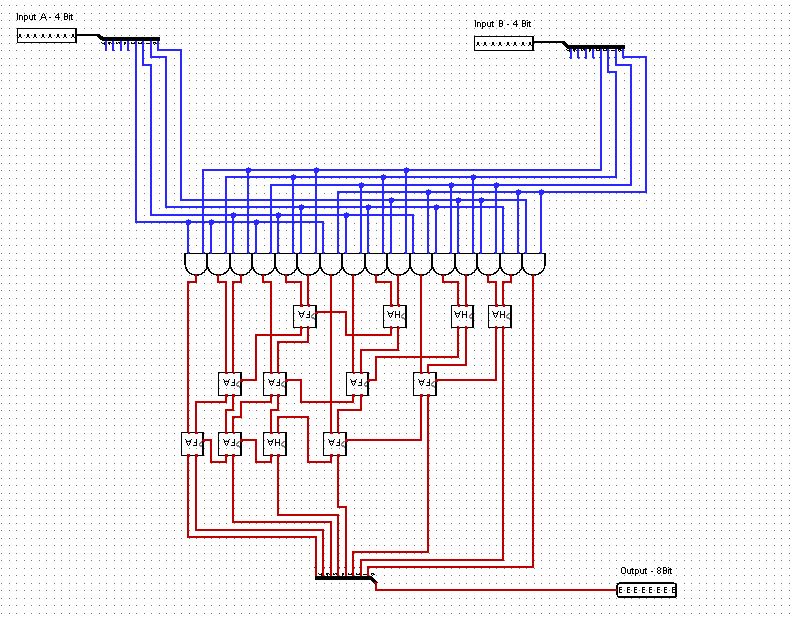
## Team Designed 8 bit Adder



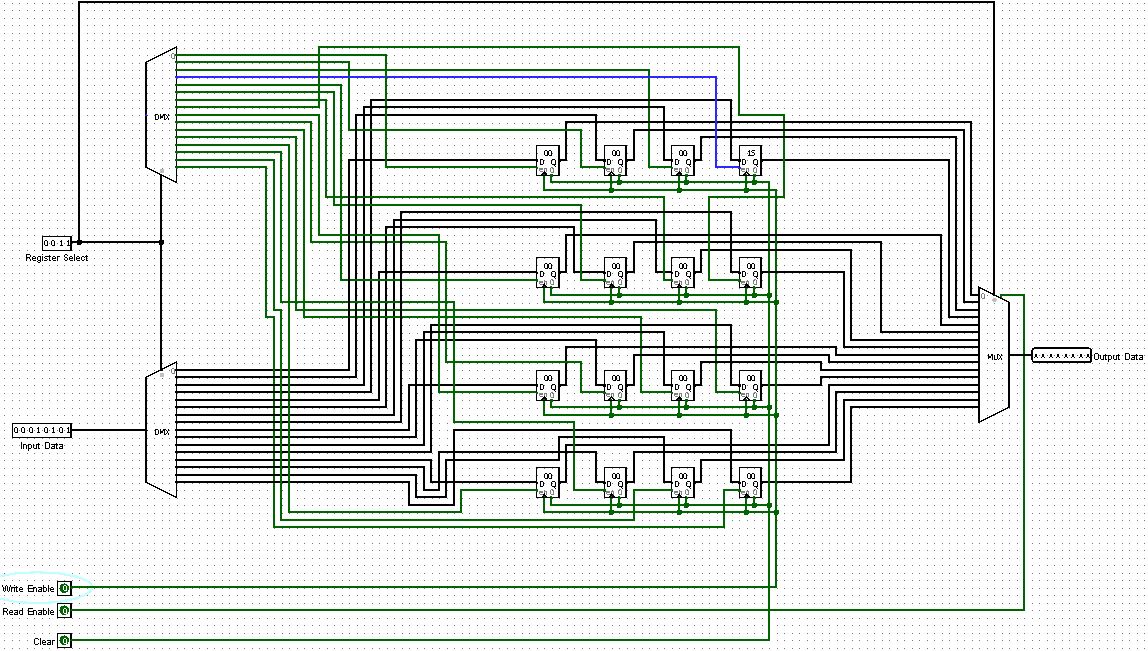
## Team designed subtractor



## Team designed Multiplier



## Team Designed 8-bit RAM



## Operation of the RAM

The work of our RAM, just like any other RAM, stores information. In this case we are storing data to be used for our ALU operations. Our RAM is made up of sixteen 8-bits registers, two De-multiplexers and two multiplexers. The De-multiplexers operate during the input stage of our RAM operation. Whereas the multiplexers operate during the output or display stage. The multiplexers display the binary values of the input stored in the register.

The top DMUX is the register selector DMUX. This is a 4-data bit DMUX with a 4-bit selector for our 8-bit registers. The register number, which must be in binary is inputted using the selector. This would make the desired register open/ready for data input and for storage. The second DMUX is for the inputting of the data you’d want to be stored in your selected register. The data to be stored is inputted via the input bit of the second DMUX, hence our data DMUX.

All the registers have an enable output and a clear output. The enable outputs are all connected to a clock. The clock must be enabled before data input in the second DMUX can be stored in the register selected. Also, all the clear outputs are connected to a 1-data bit input set at zero (0). When the set input value is changed to one, all the data in the registers are cleared.

After your desired data has been stored or loaded into the registers, you’d want to display them or use it to perform an operation. The multiplexers would be used to aid this function. Both MUX’s operate the same way, nonetheless not necessarily giving you the same output values. Also with the MUX, there’s the select bit (which is also 4-data bit input) which selects the register you would want to display its stored value. The clock enable is the read enable feature and this must be “ON” in order for the output to be displayed.

## Team designed Control Unit

## Operation of the Control Unit

The operation of the Control unit is divided into three main parts. The fetch process, decode process and the execute process. The control unit is attached to the ALU (Arithmetic and Logic Unit) and several registers. (The registers mainly being memory address register MAR, instruction address register and other internal rregisters to keep data.)

**FETCH**

The first step the Control unit carries out is to fetch some data and instructions (program) from main memory then store them in its own internal temporary memory areas. These memory areas are called 'registers'. Fetch decode execute cycle. This is called the 'fetch' part of the cycle. For this to happen, the Control Unitt makes use of a vital hardware path called the 'address bus'. The Control Unit places the address of the next item to be fetched on to the address bus.

Data from this address then moves from main memory into the control unit by travelling along another hardware path called the 'data bus'.

**DECODE**

The next step is for the control unit to make sense of the instruction it has just fetched. This process is called 'decode'.

The Control unit is designed to understand a specific set of commands. These are called the 'instruction set' of the Control unit. Each make of control unit has a different instruction set.

The Control unit decodes the instruction and prepares various areas within the chip in readiness of the next step.

**EXECUTE**

This is the part of the cycle when data processing actually takes place. The instruction is carried out upon the data (executed). The result of this processing is stored in yet another register. Once the execute stage is complete, the Control unit sets itself up to begin another cycle once more.