CSC 345:

Project Five: Computer Architecture

February 25, 2025

LOGIC SYSTEMS

Contents

Summary of Work

Project Overview

Project 5 is the culmination of most of the previous projects. Throughout the past few weeks, GTM began to build the basic components of a computer's architecture. Project 5 was the implementation of all of this previous work to build the most significant components of a computer. In this project, GTM created the Memory unit, the CPU, and finally the hack computer. From this project, we were able to build a computer using elementary chips as well as advanced chips.

Key Achievements

Throughout the project, several milestones were achieved, including:

- Successful creation of the Hack Computer.
- Implementation of CPU.
- Implementation of the Memory chip.
- Implementation of the Hack Computer.

1 Research and necessary information!

The computer being built throughout Project Five is based on Von Neumann Architecture. In this hardware architecture style, the computer is built off of relatively basic building blocks, Along with relatively simple instruction sets. Von Neumann architecture was the building block of the modern computer space. Many companies have taken ideas from Von Neumann and created more complex systems. This style of architecture aids in creating comprehensible systems. This style is how we will be building the hack computer. In its basics, these Von Neumann systems take a CPU, Memory unit, and Input/Output system. Although Von Neumann ideas are relatively new to the GTM team, the central ideas can be seen throughout past projects. Similarly to the architecture, GTM has combined easy to understand chips to allow for complex operations. This process of building complex from relatively noncomplex objects has been seen throughout the entirety of our journey. Now let's begin to understand how we will create this. Our computer use the concept of stored memory. This means that "the computer's memory stores not only the data that the computer manipulates but also the instructions that tell the computer what to do."[1] Our computer will use the data structures built previously and combine them to create what is called Data Memory. We will also have instruction memory. Simply put, the instruction memory will tell the computer what to do and the data memory will store the result of what the computer has done. Data memory and operations done on Data Memory are reduced to reading or writing selected registers. For the instruction memory, the binary instructions will be loaded into the CPU. The CPU is the core of the computer. The CPU also known as the Central Processing Unit, is in charge of executing the instructions of a program. The CPU does this by, utilizing the ALU, registers(A,D), and a control unit (PC). We

[1] Noam Nisan and Shimon Schoken. "The Elements of Computing Systems." In: *MIT Press* (2021), p. 126

finally create the computer which is a combination of what we have done so far.

The Objective

During this project, we were tasked with creating the Memory unit, the CPU, and lastly the Computer.

2 Designed Chips

Using chips built in previous projects we will create the Memory Unit. The Memory unit will later be used in the finalized computer. We will also create the CPU, which will also be implemented in the computer.

2.1 Memory Unit

To begin the process of implementing our Memory chips we have to understand what will make up the memory chip. The memory chip has 16K bits of memory designated for general-purpose storage. The unit also has 8k bits of designated memory for the screen. And lastly, the memory chip also has a one-bit address for the keyboard. The memory chip takes in an input of 16 bits from the CPU which is what will be written to memory. The memory chip also takes and 15-bit address which is where the memory unit will write the data. The memory chip also takes a load value which, determines if the data will be written. The chip also has a 16-bit address which will be the CPU's inM input. We begin by creating the memory unit with a DMux4way. This gate is used for reading or writing to RAM, as well as for selecting the keyboard screen by taking the load value as input. From here, we use an Or gate to decide if we are reading or writing to ram. From there, we use a RAM16k to store data. We also use the built-in screen chip as well as the keyboard chip to finalize the memory storage. We tie all of the outputs from these various chips together using a Mux4Way16, this allows us to accurately output the correct 16-bit value.

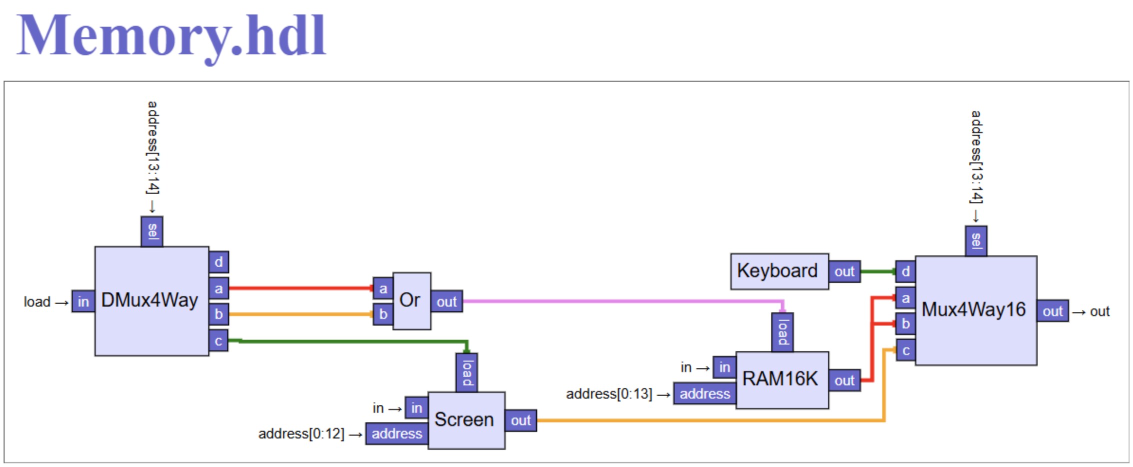

Figure 2.1: Memory Chip

2.2 Computer Processing Unit (CPU)

The CPU was by far the most challenging chip and task GTM has tackled so far. The CPU relies on many different components. Each of these components must have the proper instructions and outputs to function properly. The CPU has four main outputs. The outputs are the outM[16], writeM, addressM[15], and pc[15]. Translated these mean, the value of M's output, if we write to M, the address in data memory, and the address of the next instruction respectively. The CPU chip has 3 inputs the inM, where M is the contents of RAM[A]. It also has instruction[16], which is the instruction for execution. Lastly, we have reset which is the input that determines whether we should restart the current program or continue executing the program. Let's break this down a little further. The instruction [16] consists of 16 bits; however, we really only care about 13 of those bits. The instruction bits determine what operation will be done. You can break the instruction sets into the A instructions, C instructions, D instructions, and lastly J instructions. One of the largest issues we came across was incorrectly using these instruction bits. One of the bits we ended up incorrectly classifying was instruction [11]. This bit here is a C instruction used in the CPU to determine if the x inputs should be zeroed. This 'Zx' is integral to boolean algebra. We incorrectly set this bit to instruction[12]. This set the value of Zx to an A instruction bit. Not only did we get this section wrong all cascading inputs into the ALU were incorrect due to this error. We discovered this issue by breaking down the 16-bit binary instruction and cross-referencing it to the given hack language specification sheet. This sheet can be found in Figure 4.5 of the textbook. Another issue we had was with the program counter. We discovered that the program counter(PC) is what is used to determine the jump instructions i.e. JLE. The determination of the right gates to use in combination proved to be extremely difficult. We found a combination or Or, And, and Not gates, which gave us the correct output. You can see the chip diagram in figure 2.2, below to visualize our implementation.

CPU.hdl

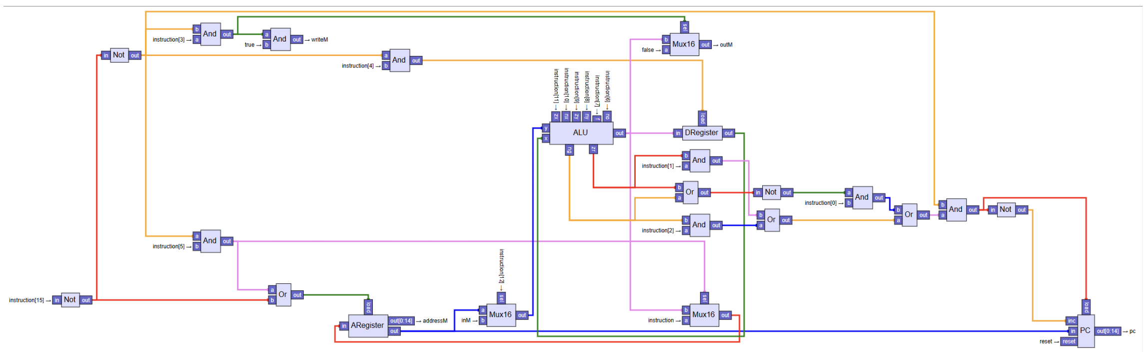

Figure 2.2: CPU Chip

2.3 Computer

The final chip we built was the computer. This was the simplest section out of all of the other processes to create. The computer only takes a reset input and no output. The computer is built off of a CPU, ROM32K, and a Memory chip. The ROM32k is read-only memory and will use instruction sets to run. For our implementation, we simply linked the chips together and created the computer. You can view our implementation below in Figure 2.3.

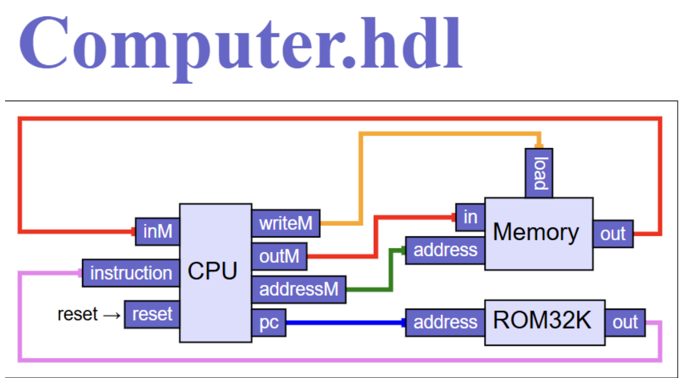

Figure 2.3: Computer

2.4 Other Considerations

Our hack machine treats all the storage as being of the same speed. In a traditional computer, this would not be the case. Some data may be slower than others due to the size of the memory unit, or distance to the CPU. Our computer does not take this into account. To address the different speeds, some computers, utilize larger caches to ensure no CPU starvation. They also implement a data hierarchy. This hierarchy is based on data that is frequently used. An abstracted version of this can be seen in the Windows file explorer, where recently used and often opened files are easily accessible. These methods, along with others, allow computers to address memory latency and differing speeds.

Conclusion

In Project 5, GTM created the CUP, Memory unit, and Computer. The Memory unit has three main sections with 16k available for data, 8k for the screen, and 1 bit for the keyboard. The CPU utilizes registers, ALU, and a program counter to function. The computer is built off of read-only memory, the main memory unit, and the CPU. All of this was done successfully throughout the project 5 timeline. This project was the realization of all of the previous projects. Now that GTM has finalized the computer we can go forward with new assembly programs which can be powered off of the CPU. In a previous project, GTM built the ALU which was an instrumental component of the CPU chip. All of the work done in the previous projects led us to what was accomplished here,

Bibliography

Nisan, Noam and Shimon Schoken.
"The Elements of Computing Systems."
In: *MIT Press* (2021), p. 126.

LOGIC SYSTEMS

Mereck McGowan