Alexander Cleveland – Personal “Essay”

* Education

This is my last semester at UWRF and when I graduate, I will be receiving my Bachelors of Computer Science and Information Systems. I had started my education over at Century College and had transferred here after I had received my Associates of Computer Science.

* Career Plans

My one and only career plan is to find work as soon as I graduate from UWRF.

* Experience with and knowledge of C programming and Linux

The only experience I have with C programming and Linux are all from classes I took while attending UWRF.

* Understanding of the material in the Computer Organization and Assembly Language course including:

1. **Binary, octal, and hexadecimal numbers**

Binary is information that can be expressed by combinations of 0’s and 1’s such as the number 21 being represented as 10101. This Binary can then be converted to either Octal or Hexadecimal numbers. The octal number for 21 would be 25 and hexadecimal number would be 15.

1. **Hardware needed to add two numbers**

An Adder is used to add two numbers together.

1. **Combinatorial and Sequential Circuits**

A sequential circuit is affected by clock time where a combinational circuit is not. The Combinational Circuit is Independent whereas a sequential circuit is dependent on previous results.

1. **How D latches are related to CPU registers, RAM and ROM**

I know that when the clock input reaches zero D latches will hold information and then replace it or keep it depending on other inputs when the clock input is one.

1. **The number of address and data lines needed to access a 1 GB memory in which each memory location is 16 bits**

To find Address lines it as simple as finding the = 10 and the data line being even simpler, being 16 for the size of the memory locations.

1. **The time duration of a clock cycle of a 5 GHz clock**

I either do not remember how to calculate this or we were moving to online because of Covid-19 and this was skipped.

1. **The assembly language instruction needed for a for loop.**

Mov cx, 4

Example:

Dec cx

Loop example