## CSC 120 Lab 03

Use the **emulator** to find the answers to your work. The goal of this lab is to understand how to use the emulator and run instructions on it. You should be able to make sense of the values returned by the emulator.

**Emulator link below**

https://joeledstrom.github.io/brookshear-emu/#AA01

**Appendix C link below**

https://blackboard.waketech.edu/bbcswebdav/pid-18088193-dt-content-rid-148874207\_1/xid-148874207\_1

### (10 points) The following table shows a portion of a machine's memory containing a program written in the language described in the language description table. See the first page of this lab Answer the questions below (there are two) assuming that the machine is started with its program counter containing 00. Recall the language requires two bytes per instruction

### Address Content Interpretation

### 00 21 Execute the instruction 210B

### 01 0B

### 02 14 Execute the instruction 1404

### 03 04

### 04 C0 Execute the instruction C0000

### 05 00

### What bit pattern will be in register 4 when the machine halts?

### A A5 B. C0 C. 27 D. C7

### 

### What bit pattern will be in the program counter when the machine halts?

### A 05 B. 06 C. 07 D. 04

**(10 points) The following table shows a portion of the machine's memory written in the language described in the language description table. Answer the questions below assuming that the machine is started with its program counter containing 00.**

**Address Content Interpretation**

00 25 Execute instruction 2503

01 03

02 A5 Execute instruction A502

03 02

04 35 Execute instruction 3503

05 03

06 24 Execute instruction 2400

07 00

08 34 Execute instruction 3404

09 04

0A B0 Execute instruction B003

0B 03

0C C0 Execute instruction C000

0D 00

What bit pattern will be in memory location 03 when the machine halts?

A C0 B. 05 C. 00 D. A0

**(15 points) The following table shows a portion of a machine's memory containing a program written in the language described in the language description table (adopted from Chapter review problem #15 ) Use the emulator**

**Address Content Interpretation**

0x00 1C Execute instruction 1C03

0x01 03

0x02 2B Execute 2B03

0x03 03

0x04 5A Execute 5ABC

0x05 BC

0x06 3A Execute 3A00

0x07 00

0x08 C0 Execute C000 (Halt)

0x09 00

What bit pattern will be in register A when the machine halts?

A. 30 B. 03 C. C4 D. 06

What bit pattern will be in memory address (cell) 00 when the machine halts?

A. 30 B. 03 C. C4 D. 06

### 

### (15 points) Explain briefly the concept of opcode/operand and the types of load instructions for the Vole architecture. What is the general format of the instruction? Give examples

Opcode and operand are a way of encoding binary strings into a meaningful instruction set that allows processors to move, calculate, and compare data. Vole architecture is an example of this instruction set which encodes 16 bits of binary into an easy-to-read encoding of four hexadecimal characters.

The first character is the opcode and is the base instruction for the processor to perform, ex. load, rotate, add. The next three characters are the operands which will detail which cells of memory data should be moved from, moved to, compared to, calculated, or placed in the register.

Vole architecture reserves opcode 1 and 2 for load instructions. The two instructions differ in that opcode 1 moves data from a memory cell location into the register while opcode 2 writes a value to the register, instead of from a memory cell. For example, if memory cell location 2B had a value of 01, the instruction 0x152B will result in register 5 having a value of 01. By contrast, if memory cell location 2B had a value of 01, 0x252B will result in register 5 having a value 2B.

### 

### Section B Exploration Challenge (20 points)

In this module, we learned about processor architecture. In recent years, deep learning and AI advances are creating a revolution in processor design. Researchers and engineers today are having to rethink the idea of chip design. It is a very exciting time to witness this change. Therefore, your role in this assignment is to research how the field of artificial intelligence and deep learning is revolutionizing chip design. Use google as well as Google Scholar or ACM Digital Library. You can start by reading the below article which was published in Communications of the ACM

[Making Chips Smarter](https://cacm.acm.org/magazines/2017/5/216326-making-chips-smarter/fulltext)

URL: https://cacm.acm.org/magazines/2017/5/216326-making-chips-smarter/fulltext

Your answer should be 2-3 paragraphs and should try to answer the following questions.

* Summarize 3 key points from the article.
* How are chips developed for deep learning different from traditional processors?
* What are some of the emerging technologies that are being used today in chip design?
* What is Moore’s law and why is it relevant in this context?
* What are some of the challenges ? (Price, power consumption, design)

Once again the aim is to promote self learning and exploration. Provide references for your sources and come up with your own answer. Do not copy.paste information directly from other articles.

**Note:** Google Scholar is a free search engine for searching research and academic articles. You should definitely try using this to understand how scientific articles are written,

In the article, Making Chips Smarter, it was highlighted that artificial intelligence, machine learning, and deep learning are posing unique challenges and opportunities in computer chip design. What was essentially a guideline of how computer processing power increases while cost decreases, exponentially, Moore’s Law is reaching a physical limitation as the size of a transistor is approaching that of an atom.

The traditional method of computer chip design is changing with the new possibilities of AI and deep learning. These new fields of advancement can benefit from custom made computer chips that are designed specifically for the algorithm or task they perform. However, this poses a problem of not being flexible enough to be used for a multitude of purposes, and this proves to be very costly. Traditional processors today are considered to have a high numerical precision, and scaling this for use with AI requires high amounts of power consumption and yet still does not achieve the preferred performance.

New forms of computer design are being created that intentionally aim for low numerical precision, and in some ways are even more capable for machine learning all while having the added benefit of less power consumption. Field programmable gate arrays are another design technology that is being used to boost efficiency of processing by allowing implementation of algorithms onto hardware which can be modified and adapted in almost a moments notice. As we move into a new era of science and technology, traditional computer chip design is changing to accommodate extremely high processing requirements with low power consumption.

Sources:

<https://cacm.acm.org/magazines/2017/5/216326-making-chips-smarter/fulltext>

<https://www.intel.com/content/www/us/en/silicon-innovations/moores-law-technology.html>

### Section C Programming Challenge (30 points)

1. (15 points) Write a python program to accept a number from a user. The program should print "even number" as an output if the number is even and print "odd number" as an output if the number is odd. If the number is negative, it should print "negative number".0 is considered even.

Hint: Use the modulus operator "%". Read online on what it does and how to use it. Part of the exercise is to learn to find information online. Do not copy code directly without understanding it.

#This program receives a number from the user and will display whether the number is even, odd, or negative.

#Get number from user

num = int(input("Please enter a number"))

#Calculate whether the number is positive, remainder is 0 or 1, and determine wheter the user's number is negative, even, or odd.

if num > 0:

  if num % 2 == 0:

    print("Even number!")

  else:

    print("Odd number!")

else:

  print("Negative number!")

1. (15 points) Write a python program to print the first 20 even numbers.

#This program prints the first 20 even numbers

#Establish variable

num = 0

#Perform loop in range 0 - 20 to see if number is even (remainder 0), if so, print number

for n in range(21):

  if num % 2 == 0:

    print(num)

  num = num + 1

<https://colab.research.google.com/drive/1w6M63-SY00cEJVDSAM0GBVSF6IMg8LN0?usp=sharing>

**Instructions: Upload the file with the screenshot on Blackboard with your firstname\_lastname.docx**