Lab 02 CSC 120

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) 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.

**NOTE: 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

(+5) The following table shows a portion of 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

(+5) 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 assuming that the machine is started with its **program counter containing 00. *Use the emulator***

address content Instruction to execute

00 10 1002

01 02

02 24 2404

03 04

04 B4 B40A

05 0A

06 C0 C000

07 00

08 C0 C000

09 00

0A C0 C000

0B 00

0C C0 C000

0D 00

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

A. 2A B. 24 C. C4 D. 04

(+10) 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

* 1C Execute instruction 1C03
* 03
* 2B Execute 2B03
* 03
* 5A Execute 5ABC
* BC
* 3A Execute 3A00
* 00
* C0 Execute C000 (Halt)
* 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

(+10) Write a short program in the Vole architecture to perform the requested operation. Assume your program is placed in memory starting at address 00 (== 0x00) Remember to add C000 (== halt) to your program

Interchange the values stored at memory locations D8 and B3

As an example, to copy the value stored in memory address D8 to memory address B3 the following program would work:

Address Content Interpretation

* 14 Execute 14D8 R4 \*(D8)
* D8
* 34 Execute 34B3 B3 \*(R4)
* B3
* C0 Execute instruction C000 (== halt)
* C0

The above example can be interpreted as follows:

* Load the contents of address D8 into register 04
* Store the contents of register 04 (==\*(D8) ) into memory address B3
* Halt

|  |  |
| --- | --- |
| Address | Content |
| |  | | --- | | 00 | | 01 | | 02 | | 03 | | 04 | | 05 | | 06 | | 07 | | 08 | | 09 | | |  | | --- | | 14 | | D8 | | 11 | | B3 | | 34 | | B3 | | 31 | | D8 | | C0 | | 00 | |

(+25) Encode each of the following commands in terms of the machine language described in the language description table. **Use the following set of instructions**

17A5 5465 7465 B2B2

6456 27A5 8465 34AA A750

LOAD register 7 with the value A5. 27A5

LOAD register 7 with the contents of the memory cell at address A5. 17A5

ADD the contents of registers 5 and 6 as though they are values in 2’scomplement notation and leave the result in register 4. 5465

OR the contents of registers 5 and 6, leaving the result in register 4. 7465

AND the contents of registers 5 and 6, leaving the result in register 4. 8465

**(+10)** Why do modern CPU's have cache memory and a main memory? What is the function of the hard drive? Discuss how CPU speeds and main memory sizes have evolved from 1960's to 2021. Main memory is a larger source of storage for more permanent, longer term that outside of the CPU but attached to it while cache memory is a smaller but more easily accessible form of memory located within the CPU itself. Cache memory is based on temporary, currently relevant portions of main memory that have been consistently or recently accessed, and if edited in cache memory, will be updated in long-term memory at a more opportune time. A hard drive functions as secondary memory, often times used as a back-up for main memory. They tend to be more static as opposed to their more dynamic counterparts located internally. Hard drives are made to be more effectively swapped in and out when main memory is full. Overall, RAM capabilities and processing speeds have increased incredibly in terms of efficiency since the 1960’s but, simultaneously the data itself that needs to be stored has been condensed to a more efficient form. So not only has the CPU and memory become much better at doing what they do, they have also become much smaller in size. In the 1960’s, CPU’s that filled entire rooms were only able to store 100s of MB of storage at a time through main memory, in 2021, we have GBs of storage within our smartphones, a fraction of the size of their predecessors.

\*from the textbook primarily

Note: I am looking for your understanding and hence you may have to read some articles online. Be sure to include the links to the articles that you have read.\*

**(+15)** 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. The op-code is the indication of the operation to be executed (i.e. STORE, LOAD, SHIFT, JUMP) and is at the beginning of the instruction. An operand is more detailed information about the operation specified by the op-code. Load instructions request to fill register with the contents of a memory cell. There are 2 Load types in the VOLE, one that loads the address contents and one that loads the value.The general format of the instruction is 16 bits represented by four hexadecimal digits. The first four bits represent the op-code the remaining bits show the operand fields.

Example:

**Actual bit pattern: 0011 0101 1010 0111**

**Hexadecimal Form: 0x35A7**

**STORE** the bit pattern found in register 5 in the memory cell whose address is 0xA7

**Python code required for the next 2 problems**

(+10) 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.**

(+10) Write a python program to print the first 20 even numbers. For example

**NOTE: Please name this file as yourlastNameLab02P2.py. As an example if your last name was Euclid then you should submit EuclidLab02P2.py**

**What did I learn from this chapter?**

In chapter 2, I became a little more comfortable with various levels of conversions, how to interact with python and writing and executing some very basic code. It took a lot of trial and error, but by the end I felt comfortable in the understanding that there a number of ways to execute the same python code and it comes down to what is most efficient, but also what is most easily accessible to other reviewing it. Notes are incredibly important within python for this purpose.