Computer Organization

Lab 1: von Neumann Architecture

& Data Representation

Lab Objective:

1) To study how a von Neumann Machine works.
2) To understand how different types of data are represented in modern electronic computers including unsigned integers, signed integers, floating-point numbers, characters and strings.

Introduction

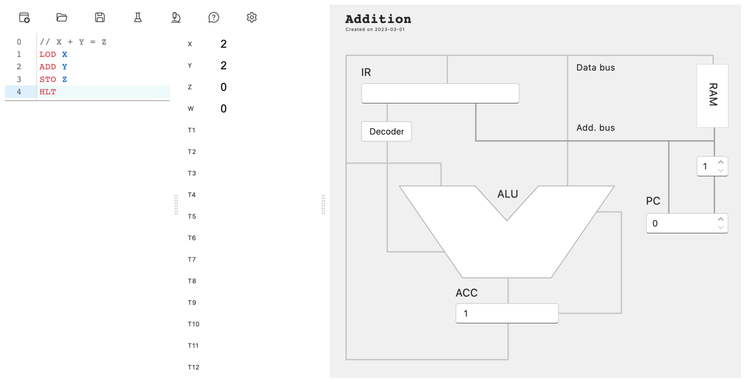
The von Neumann architecture is a computer architecture that is based on the concept of **stored-program computers**. It was developed by the mathematician and computer scientist John von Neumann in the 1940s and is used in most modern computers today.

The von Neumann architecture is based on the principle of having a single shared memory for storing both data and instructions that the computer needs to execute. The computer's CPU reads instructions from this memory (**fetch**), decodes them (**decode**), and then executes them (**execute**). This is called the **fetch-decode-execute cycle**.

This architecture also includes the use of a **control unit**, which manages the flow of data and instructions between the **memory** and the **CPU**. The von Neumann architecture is characterized by its use of a sequential processing model, where instructions are executed one at a time in a linear sequence. This is done by the automatic increment of PC right after the fetch action. It also allows for the concept of conditional branching, which allows the computer to make decisions based on the data it is processing.

von Neumann Machine Simulator

This von Neumann Simulator (https://www.vnmsim.app/) is a small tool for students and teachers to study how a Von Neumann Machine works. The right side shows the von Neumann architecture and the left side shows the memory that contains both the data (variables) and the program.



```
// X + Y = Z
LOD X
ADD Y
STO Z
HLT
```

```
X  2
Y  2
Z  0
W  0
```

It is fully compatible with the most common commands for the von Neumann Machine. It is able to load data (LOD), store data (STO), do additions (ADD), subtractions (SUB), multiplications (MUL), divisions (DIV) and perform jumps (JMZ/JMP).

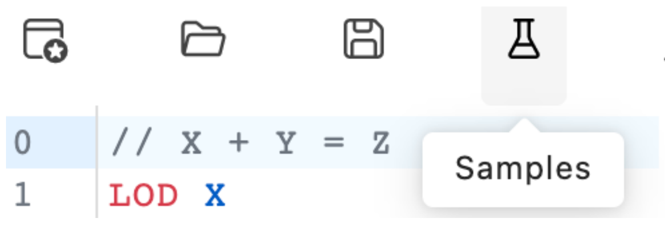
LOD *var/cell*	**LOD** *#num*	**STO** *var/cell*	**JMZ** *cell*	**JMP** *cell*
ADD *var/cell*	**ADD** *#num*	**SUB** *var/cell*	**SUB** *#num*	**NOP**
MUL *var/cell*	**MUL** *#num*	**DIV** *var/cell*	**DIV** *#num*	**HLT**

Lab Instruction

Step 1: Open the von Neumann Machine simulator page at

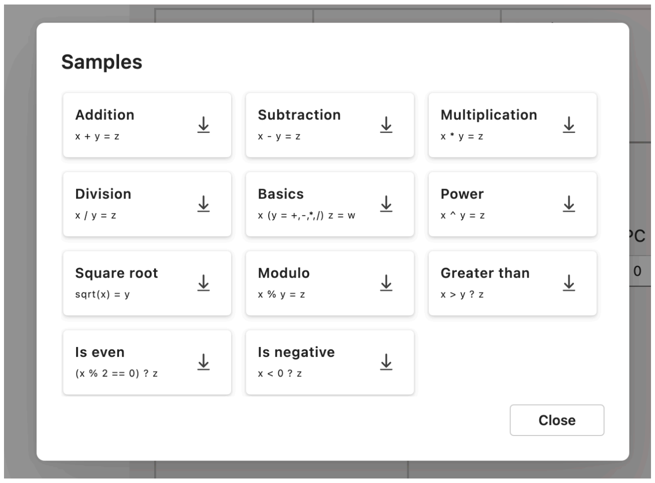
https://www.vnmsim.app/

Step 2: Click "Sample" on the menu bar:



```
// X + Y = Z
LOD X
```

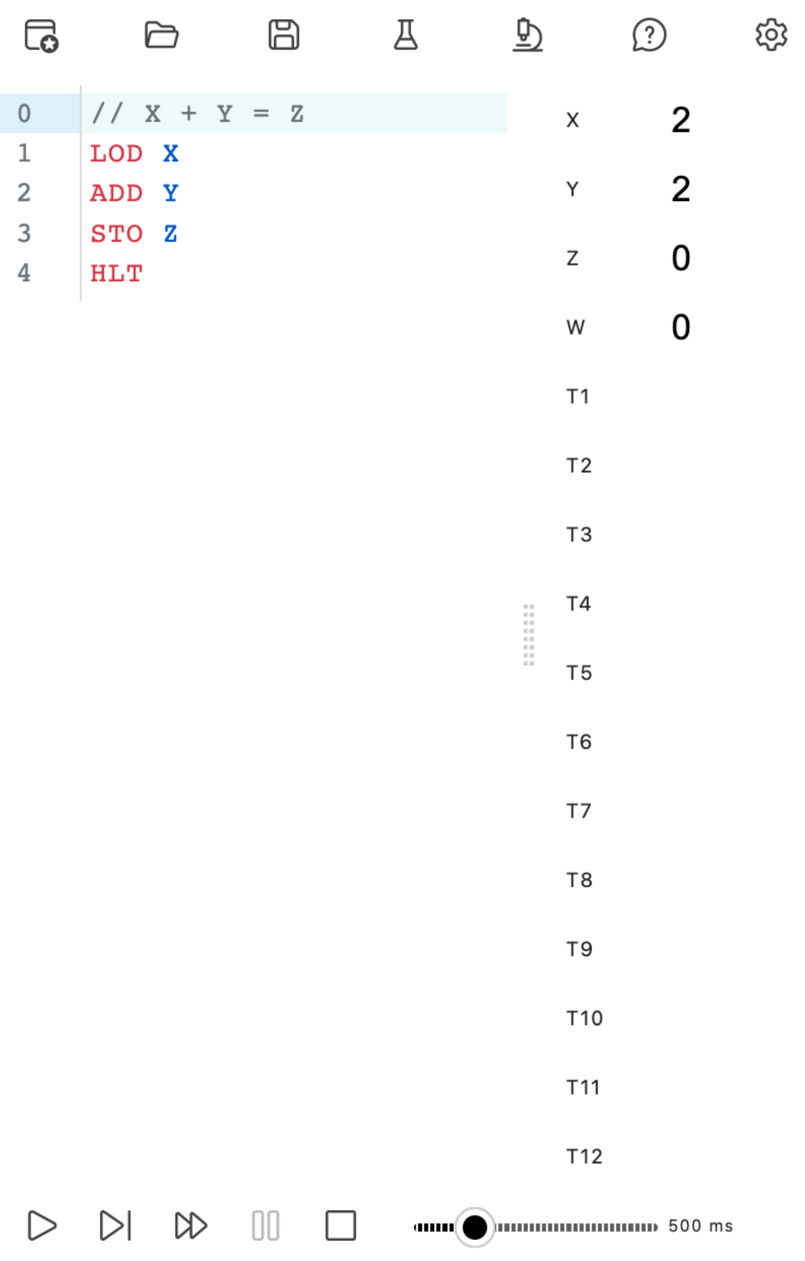
You will see the following 11 samples that you can load and run.



Step 3: Select the first sample "Addition" to load the program and data into the memory. This program totally has four instructions that are stored in the memory.

1. *LOD X*
2. *ADD Y*
3. *STO Z*
4. *HLT*

It will load the value X, add Y to it, store the result to Z, and then halt. Here X, Y and Z are three memory locations or variables. The initial values of X, Y and Z are 2, 2 and 0.



```
// X + Y = Z
LOD X
ADD Y
STO Z
HLT
```

```
X    2
Y    2
Z    0
W    0
T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12
```

Step 4: To run the program. You can click the "Run loop" button at the bottom of the page. Or you can click the "Single step" button to trace the operation of the von Neumann machine in a step-by-step manner.

Step 5: Try to play with the example and run the von Neumann Machine to see how the **fetch-decode-execute** cycle works for each instruction. You may also try other samples provided by the simulator.

Lab Exercise: （Solve all questions by chatGPT and verify the answers by hand）

1) Given 16 bits, what are the Signed-magnitude, 1’s complement and 2’s complement representation for -59? Write the answers in Hexadecimal.
2) What is the corresponding decimal form for the single point precision floating number?

   1101 1011 0101 0000 0010 0000 0110 0000

3) What is the single point precision floating number of -45.875 in computer system? Write the answers in Hexadecimal.
4) What is the ASCII representation for “5” , ‘t’ and ‘T’ in decimal?
5) What is the Unicode representation for “您好” in hexadecimal?

Submission

Submit a doc file containing your answers before next Thursday.