COAL Lab # 06

## Load & Store Instruction Assembly Code

## Name: Muhammad Usman

**Class:** BSCS 3C1

**Reg No:** cs211208

## Literature Review:

### Load instructions

### Load instructions are used to move data from memory to registers (before operation). Loads are encoded in the I-type format. The effective byte address is obtained by adding register rs1 to the sign-extended 12-bit offset. Loads copy a value from memory to register rd. The assembly representation for load instructions are:

### lw (destination\_register), (offset)(source\_register)

### or

### lw (rd), offset(rs1)

The LW instruction loads a 32-bit value from memory into rd. LH loads a 16-bit value from memory, then sign-extends to 32-bits before storing in rd. LHU loads a 16-bit value from memory but then zero extends to 32-bits before storing in rd. LB and LBU are defined analogously for 8-bit values.

Store Instructions:

Store instructions are used to move data from registers to memory (after operation). Stores are encoded in the S-type format. The effective byte address is obtained by adding register rs1 to the sign-extended 12-bit offset. Stores copy the value in register rs2 to memory.. The assembly representations for store instructions are:

sw (source\_register\_2), (offset)(source\_register\_1)

or

sw (rs2), offset(rs1)

The SW instruction stores a 32-bit value from the low bits of register rs2 to memory. SH stores a 16-bit value from the low bits of register rs2 to memory. SB stores a 8-bit value from the low bits of register rs2 to memory.

# Lab Exercise 01

## Task:

## Run the below assembly code on Venus Simulator

## li s0, 0x12345678 # Data to be store

## li s1, 0x00000020 # memory address

## sb s0, 0x0(s1)

## sh s0, 0x4(s1)

## sw s0, 0x8(s1)

## Machine Code:

**0000 0000 1000 0100 1000 0000 0010 0011**

**0000 0000 1000 0100 1001 0010 0010 0011**

**0000 0000 1000 0100 1010 0100 0010 0011**

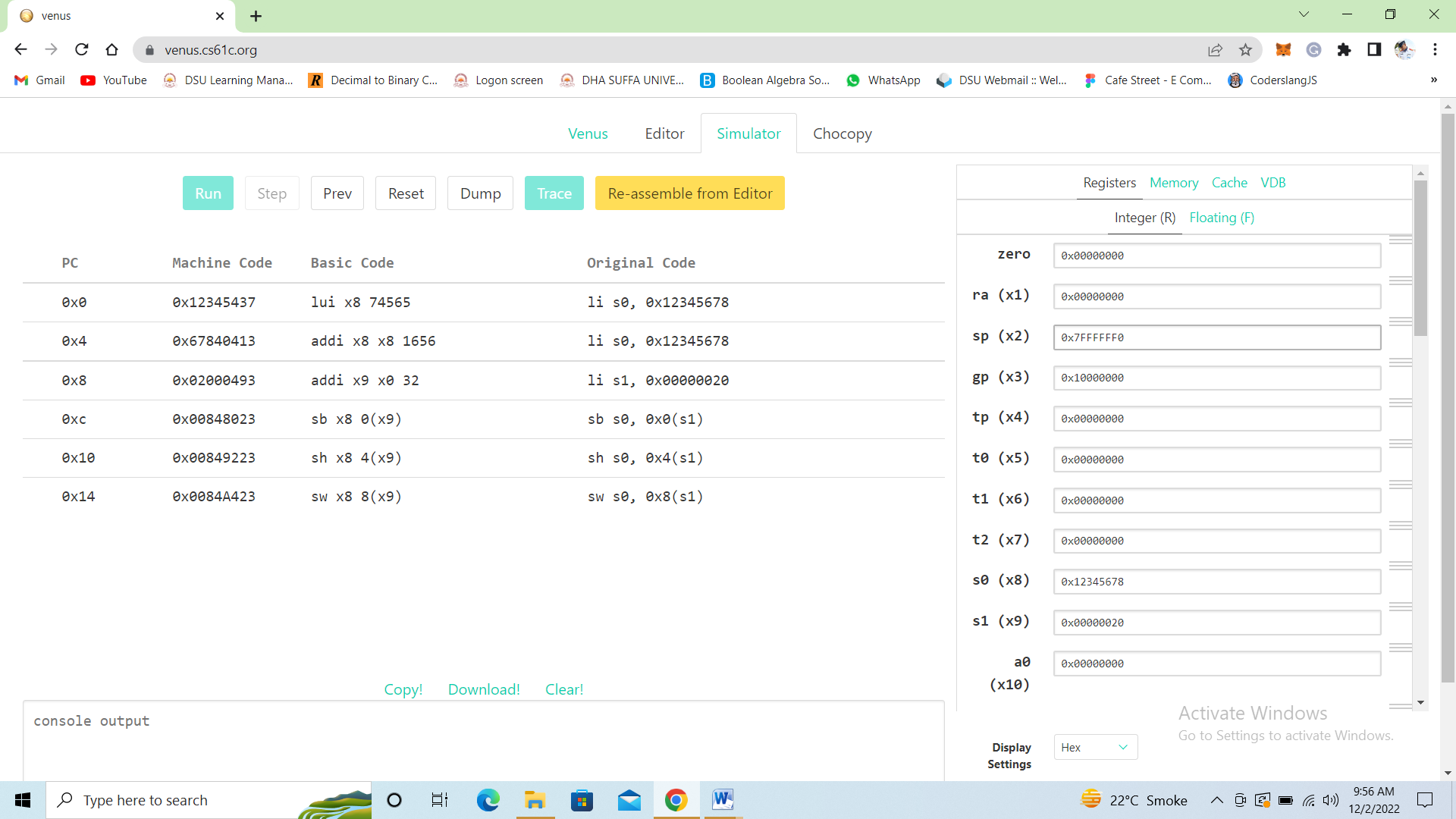
## Hexa Code:

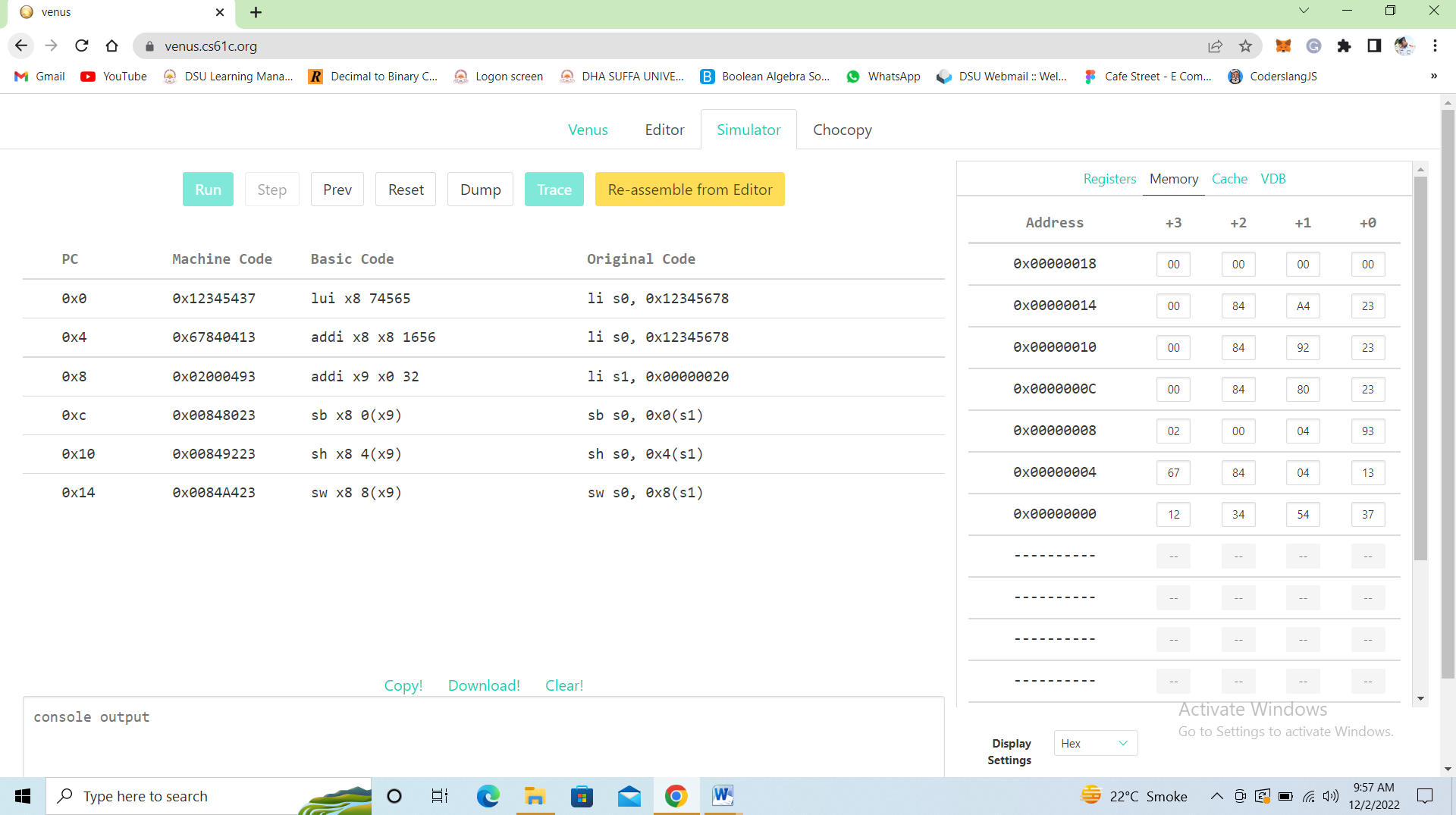
0x00848023

0x00849223

0x0084A423

# **Venus Simulation**





Lab Exercise 02

**Task:**

# Run the below assembly code on Venus Simulator

# lb t0, 0x0(x0)

# lbu t1, 0x4(x0)

# lh t2, 0x8(x0)

# lhu s0, 0xC(x0)

# lw s1, 0x10(x0)

## Machine Code:

## 0000 0000 0000 0000 0000 0010 1000 0011

## 0000 0000 0100 0000 0100 0011 0000 0011

## 0000 0000 1000 0000 0001 0011 1000 0011

## 0000 0000 1100 0000 0101 0100 0000 0011

## 0000 0001 0000 0000 0010 0100 1000 0011

## Hexa Code:

## 0x00000283

## 0x00404303

## 0x00801383

## 0x00C05403

## 0x01002483

# **Venus Simulation**

## 

# In Lab Task

# **Task 1**

# Write down a simple assembly program to add, and subtract two integer numbers and store their result into different memory locations. Stimulate the code on Venus

# **Assembly**

# addi x20,x20,0x20 # base address

# addi x18,x18,0x5

# addi x18,x18,0x5 # 5+5

# addi x19,x19,0xA

# addi x19,x19,-4 # 10-4

# sw x18,0x0(x20)

# sw x19,0x4(x20)

# **Venus Simulation**

# 

# **Task 2**

# Write down a simple assembly program to load the contents from memory into registers and perform the logical operations on them. Stimulate the code on Venus.

# **Assembly**

# addi x20,x20,0x20 # base address

# lhu x18, 0x0(x20)

# lhu x19, 0x4(x20)

# and x21,x18,x19

# or x22,x18,x19

# **Venus Simulation**

# 