

**Department of Electrical & Computer Engineering
University of California, Davis**

**EEC 170 – Computer Architecture
Fall Quarter 2024**

Laboratory Exercise 1: Learning RISC-V Assembly Language

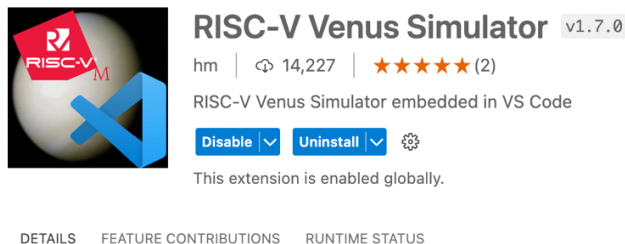
Due Date: [Friday October 10]
Full Points 150

Objectives of Lab 1

- Set up the Venus RISC-V simulator
- Learn some basic RISC-V assembly language instructions
- Learn how to print to the terminal with system calls

Step 1 - Download and install the Venus RISC-V Simulator [25 points]

- Download Microsoft Visual Studio Code from the link below:
<https://code.visualstudio.com/>
- After installation, you will need to install the Venus RISC-V extension from the extensions marketplace on Visual Studio Code, which looks like this.



RISC-V Venus Simulator embedded in VS Code

This Visual Studio Code extension embeds the popular [Venus RISC-V simulator](#). It provides a standalone learning environment as no other tools are needed. It runs RISC-V assembly code with the standard debugging capabilities of VS Code.

To use it as educational tool, further views are added as described below.

Step 2 - Run your first program

- Open lab1_skel.s program (that comes with this lab assignment) in VSCode.
- Read the statements that begin with #, which denote comments. Note that an assembly language program has two sections

.data where you declares all your variables.

.text where you put your program in the form of RISC-V instructions, one per line

- Printing is done by passing the pointer to the string or the value to be printed in register a1 or x11 and a code that tells the system what to print (integer, string, character, etc). in register a0 or x10. This is followed by the command **ecall**.
- See <https://github.com/ThaumicMekanism/venus/wiki/Environmental-Calls> for more documentation system calls and other aspects of the Venus simulator.
- Run the program in the debug mode. You can single step through each instruction.
- The program shows you several examples of printing strings and integers and how to do a simple computation.

Step 3 - Your first RISC-V Assembly Language Program [25 points]

- Modify the program to compute the subtract 2 numbers.
- Modify the message so that it prints X-Y = whatever the answer is

Step 4 – Translate a C function into RISC-V Assembly [100 points]

```
int countOccurrences(char *str, char ch) {  
    int count = 0;  
  
    // Loop through the string  
    for (int i = 0; str[i] != '\0'; i++) {  
        if (str[i] == ch) {  
            count++;  
        }  
    }  
  
    return count;  
}
```

// Program to test the function countOccurrences.

```
#include <stdio.h>
int main() {
    char str[100], ch;
    int count;

    // Input string from user
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin); // Read string

    // Input character to search from user
    printf("Enter a character to find its occurrences: ");
    scanf("%c", &ch);

    // Count occurrences of the character
    count = countOccurrences(str, ch);

    // Print result
    printf("The character '%c' occurs %d times in the string.\n", ch, count);
    return 0;
}
```

- Follow the instructions in the lab skeleton file to translate the function countOccurrences shown above.
- See the example on Page 116 of your textbook. The function you will write is very similar.
- The RISC-V assembly for data declarations and testing the program are given to you in the skeleton file
- We are going to hardcode the string and the character in the data declarations instead of getting it from the user
- Try your code with different strings and characters to make sure you get the expected result

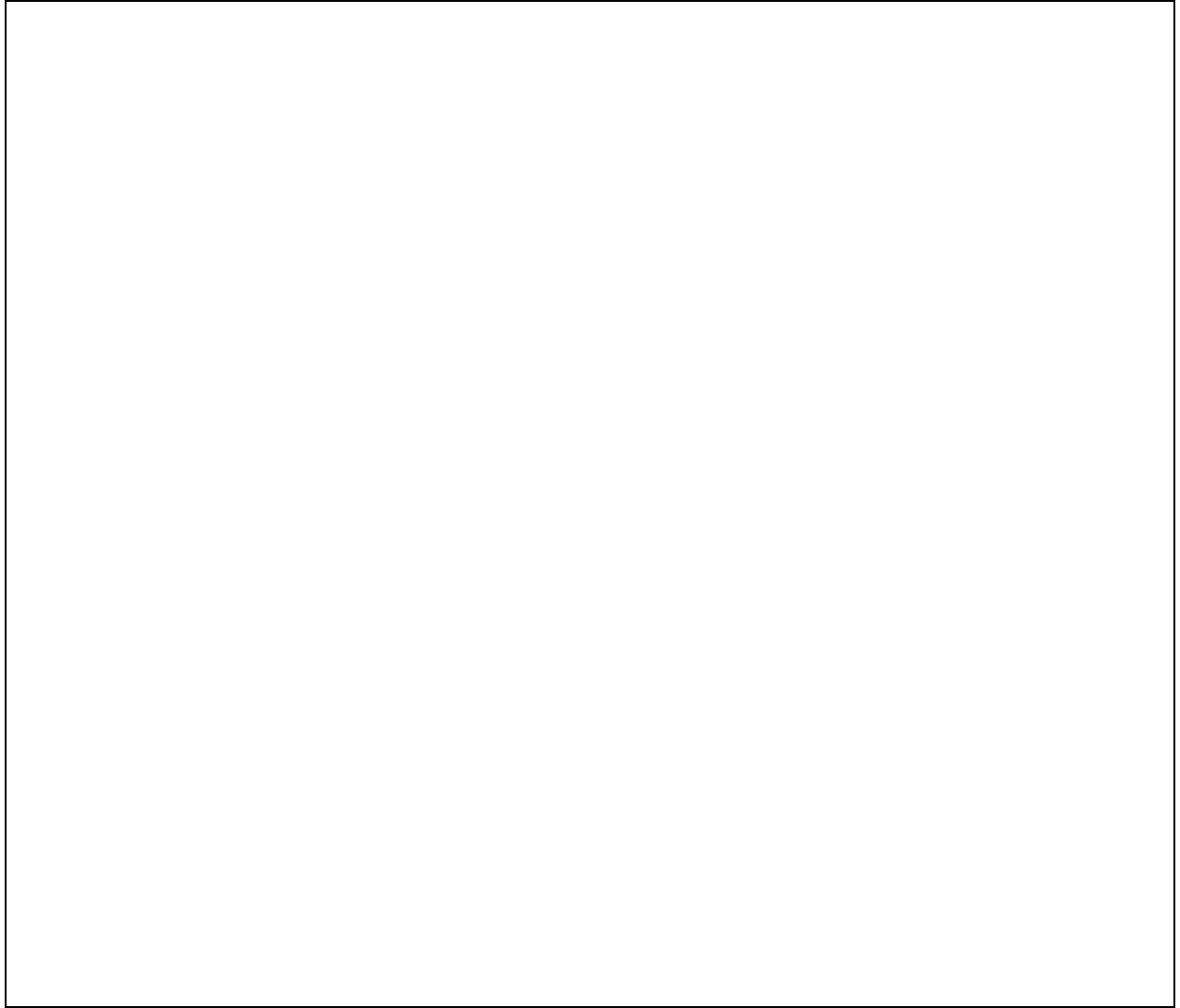
What to submit?

The attached report sheet with screenshots and code provided. Please limited your inserted code and screenshots to the specified areas (screenshot your code and shrink the images when needed)

Report Sheet

Step 2 - Run your first program

```
-----  
Starting program C:\Users\tetra\OneDrive\Des  
Welcome to EEC 170, Fall 2024  
X = 130  
Y = 15  
X+Y =145  
  
The character a appears 0 times in the  
  
Thanks. We are done.  
  
Exited with error code 0  
Stop program execution!  
-----
```



Step 3 - Your first RISC-V Assembly Language Program [25 points]

- Modify the program to compute the subtract 2 numbers.

```

# printing X-Y initially
    la a1, msg7
    li a0, 4
    ecall

# performing the computation X-Y
    lw a1, X
    lw a2, Y
    sub a1, a1, a2

# Printing the result from a2
    li a0, 1
    ecall

    la a1, newln
    li a0, 4
    ecall

##### PART 1 END #####

```

```

Welcome to EEC 170, Fall 2024
X = 130
Y = 15
X+Y =145
X-Y =115
The character a  appears 0

Thanks. We are done.

Exited with error code 0
Stop program execution!

```

- Modify the message so that it prints $X - Y =$ whatever the answer is

Created Message 7 to print $X - Y =$

```
welcome:      .string "Welcome to EEC 170, Fall 2024 \n"
msg1:         .string "X = "
msg2:         .string "Y = "
msg3:         .string "X+Y ="
msg7:         .string "X-Y ="
newln:        .string "\n"
thank:        .string "Thanks. We are done.\n \n \n"
```

Welcome to EEC 170, Fall 2024

$X = 130$

$Y = 15$

$X+Y = 145$

$X-Y = 115$

The character a appears 0

Thanks. We are done.

Exited with error code 0

Stop program execution!

Step 4 – Translate a C function into RISC-V Assembly [100 points]

```
1 ##### PART 2 BEGIN #####
2 la x5, str      # x5 = &str[0]
3 li x6, 0        # x6 = 0 or count initialized to 0
4 la x7, ch       # x7 = &ch address of ch is in register
5 lbu x7, 0(x7)   # x7 = ch the character ch is in register
6 li x8, 0        # x8 = 0 x8 is the index variable of
7
8 #your code goes here. Call your function countOccurrences
9
10 countOccurrences:
11     lb x9, 0(x5) # x9 = str[0]
12     beq x9, zero, printCount # if str[0] = /0 end loop
13     beq x9, x7, matchincrement # if str[0] == ch count
14     addi x5, x5, 1
15
16     j countOccurrences # jump back to start
17
18 matchincrement:
19     addi x6, x6, 1
20     addi x5, x5, 1
21     j countOccurrences
22
23 # Print routine is given for you. So, don't have to m
```

Starting program C:\Users\tetra\OneDrive\Desktop\lab1_skel.S

Welcome to EEC 170, Fall 2024

X = 130

Y = 15

X+Y =145

The character a appears 5 times in the string --- The goal of this lab is to learn RISC-V Asse

Thanks. We are done.

Exited with error code 0

Stop program execution!

PART 2 BEGIN

la x5, str # x5 = &str[0]

li x6, 0 # x6 =0 or count initialized to 0

la x7, ch # x7 = &ch address of ch is in register x7

lbu x7, 0(x7) # x7 = ch the character ch is in register x7

li x8, 0 # x8 = 0 x8 is the index variable of the for loop i, which is set to 0

#your code goes here. Call your function countOccurences

countOccurences:

lb x9, 0(x5) # x9 = str[0]

beq x9, zero, printCount # if str[0] = /0 end loop

beq x9, x7, matchincrement # if str[0] == ch counter ++ else continue

addi x5, x5, 1

```
j countOccurrences # jump back to start
```

```
matchincrement:
```

```
    addi x6, x6, 1
```

```
    addi x5, x5, 1
```

```
    j countOccurrences
```

```
# Print routine is given for you. So, don't have to modify anything, below this
```

```
printCount:
```

```
    la a1, msg4
```

```
    li a0, 4
```

```
    ecall
```