Computer Organization and Assembly

Salma Soliman - 900182325

Mariam Mousa – 900183871

Language Programming

RISC-V Disassembler

Dr.Mohamed Shalan



Implementation:

In this project, we aim at receiving data from a binary file that represents the machine code of RISC-V assembly instructions and decode it to have the data written in RISC-V assembly language. In RISC-V, we have 6 different formats of instructions (R, I, S, B, U and J) that we can distinguish between them using the opcode and other functions. Our implementation is basically based on multiple if and switch statements that checks the opcode at first to distinguish between different formats. After the first check, we perform one or two other checks to differentiate between the different instructions from the same format using function 7 and/or function 3 and the immediate for some instructions. We found that we need to declare more variables for the immediate values in order to be able to arrange their different sections.

Functions of the program:

1. instDecExec: This function is the core of the program. As it contains all of the if and switch statements that separate each format from the others and distinguishes between the instructions of the same format and prints them out. This function takes a buffer that contains an instruction word as an input. It decodes a 32-bit instruction word into an assembly language instruction and produces it as an output.

* Parameters: it takes the instruction word and does not return anything (void type).
* Input: simply, the input to this function is an instruction word.
* Output: the output of this function is printing the RISC-V assembly instructions that corresponds to the binary input.

1. EmitError: It is a function that can detect the error that the user might do while writing the main arguments in the terminal such as writing the name incorrect or entering arguments less than the supposed number.
2. Main: we open the binary file in the main function and we test if it is opened correctly or not. We also keep updating the program counter (PC) till it reaches the end of the binary file.

Termination point of the program:

The program will end once the PC reaches the end of the binary file.

Results and tests:

This project passed the samples that are uploaded on blackboard successfully. We also provided the rest of the instructions.

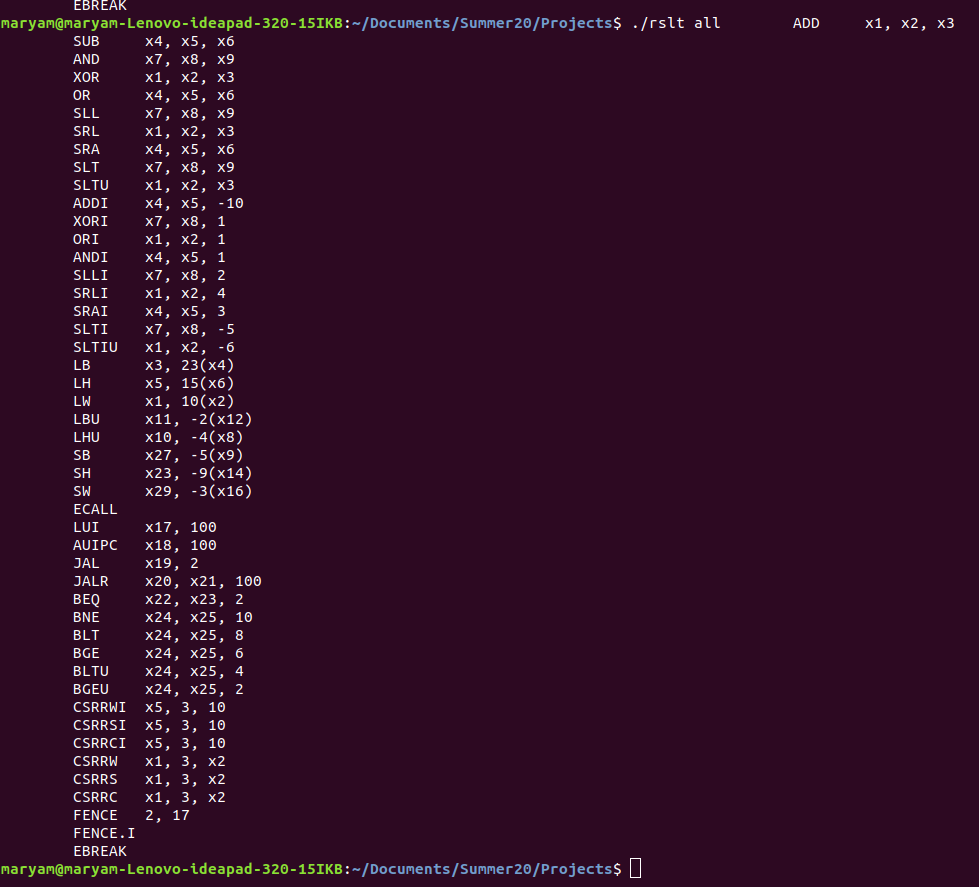


Fig (1) a screen shot showing the disassembled 47 instructions

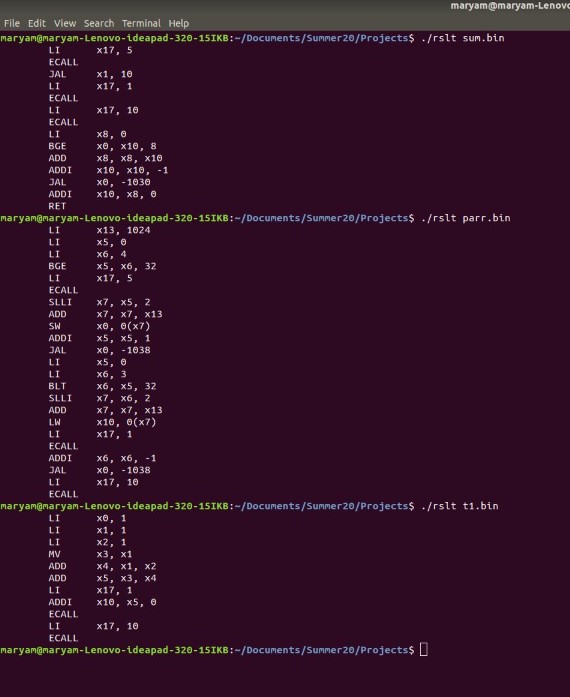


Fig (2) this screenshot shows some of the samples disassembled

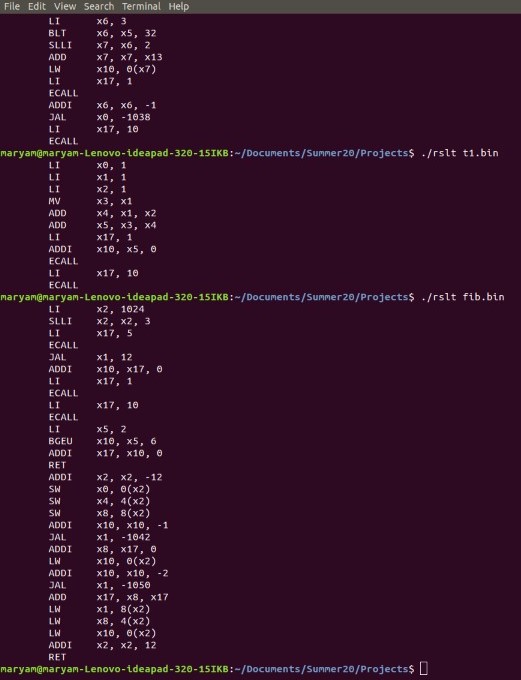


Fig (3) this screenshot shows some of the samples disassembled