CS2323
LAB-3 REPORT

SOMALARAJU BHAVYA SHLOKA
CS22BTECH11056

The code is written in C language and it's aim is to convert the Machine level code into Disassembled code(RISC-V).

1 IMPLEMENTATION

- Firstly, the code opens the input file and reads each line(8 hex digits i.e,32 binary bits instruction) as a hex string and converts it into a binary string by converting each hex digit into 4 binary bits.
- The last 7 characters of the binary string (opcode) are compared with the instruction types such as R-type, I-type, B-type, S-type, J-type and U-type.
- As the 32 Bit instruction is stored as a string i.e, each bit 0 or 1 is considered as a character.So I wrote a function that takes the string and unsigned or signed extension and calculates the value of the string in ASCI by subtraching each character by '0'.
- Based on the instruction type their specific Source registers, Destination registers, Immediates, Offsets are extracted from the binary string (instruction) and are printed thereafter using the function stringToValue.
- For every line after it is printed , I am updating the value of global variable Program counter increasing it by 4.

2 LABELING

- I declared an array which stores the value of $(Programcounter + Offset)$ of an instruction only if it is a J-type or a B-type.
- A global variable $label_count$ is intialised to 0 which increases it's value by 1 only when there is no element in that array having the same value as program counter of the present instruction (when the present instruction is B-type or J-type only).
- After reading each instruction ,I am traversing the array from it's 0 index to the $label_count$ and comparing with present instruction's Program counter. If it matches then that label should be printed.

3 TESTING

Various Input files have been used to check the output of the code where the output is then cross-checked in RIPES simulator for efficiency and correctness of the code.