LAB3: Disassembler Report

Gaurav Choudekar - CS22BTECH11015

October 12, 2023

1 Introduction

This report is regarding lab3 conducted on disassembler of RISC-V machine code.This code takes a file as input and outputs their respective assembly instruction

2 About Code

The code is written in C.Different types/formats of instruction including R/I/S/J/U/B can be obtained in assembly level instruction from its hexadecimal machine code.

3 Input format

A file is given as input,name of file is modified accordingly.in each line of file,8 hexadecimal digits are present and new line starts immediately after 8 digits rather than any space key.

4 Code Flow

4.1 Main Function

In the main function we are giving a file as input for which we want to obtain assembly instruction.for finding location of all labels required/used in B/J format,we will run traverse our file and store in array the pc(just for refernce) of instruction where we might jump during B/J instruction.we will not print anything during first traversal,now we get pc's of labels,now when traversing for second time we will print assembly instruction with,if there,label with its number.for each hexadecimal code,we are converting it into its equivalent binary 32-digit code,from this we can decode everything about that instruction.We have seperate functions for different formats,we find their format from its opcode and call function accordingly.

4.2 Other Functions

We have different functions for different formats,like I/J/U etc...from main we are giving every character in function 'machinecode' to form its equivalent 32-bit binary code.We have one function for forming array of pc(for reference) of labels.for printing label before instruction,we have one function which checks if this particular pc is present or not.

4.3 Format Functions

Different formats have their different structure for their instruction,funct 3 also plays important role in determining determining exact instruction.everything including,rs1,rs2,rd,imm,func3 etc was calculated from binary code and was printed as expected.

4.4 Accuracy Check

Before giving big inputs,i typed instruction in ripes and gave its respective hexadecimal machine code in input file.This was done for each and every instruction format seperately.When i got confident about my inputs,then i collected input text files from other and verified output result with them.necessary changes were made many times in between where i found something needs to be changed accordingly.

4.5 Label Handling

Label handling is done in this code,for jump,and also before instruction,labelling is used,for 2 different instructions,if they point to same instruction,then same labelling is used for them.

5 Run the program

1. Compile the program using gcc lab3.c -o lab3
2. run it by typing ./lab3

6 Results

This code outputs assembly level instruction for every hexadecmial machine code given in input file.labelling is done wherever needed.I will submit expected output with input files to cross check.

7 Conclusion

This lab helps us to understand about encoding and decoding better,it also helps us to get more familiar with assmebly level instructions and its different formats.