

Computer Architecture - CS2323. Autumn 2022
Lab-8/Exam (RISC-V Disassembler)
CS21BTECH11003

- I have first converted the given hexadecimal instruction into binary code and stored it in an array.
- If the last 7 bits match any opcode, I have checked the func3 and func7, and derived the respective instruction.
- After finding the respective instruction, according to the RISC-V manual, I have calculated, the instruction specific data, i.e. rd, rs1, rs2, or immediate.
- For R-type rd, rs1, rs2.
- For I-type rd, rs1, immediate
- For S-type immediate, rs1, rs2.
- For B-type immediate, rs1, rs2.
- For J-type rd, immediate.
- For I type rd, immediate.
- Since for each instruction the way of calculating respective immediate is different, I have followed the calculation according to the manual.
- In finding and renaming labels, I have traversed the instructions found out the line number of instructions, where label is needed and stored them in an array.
- While executing the code if the line number matches with any of the line number in the stored array a label is added.
- To synchronize B-type and J-type labels, the line number to which this instruction jump is checked whether it is present in the stored array if it is present then the same label is added here.
- To verify the code, I have considered test cases for each instruction.
- Immediate is also checked whether it is positive or negative according to msb,
- The test case given in the sheet is also verified.