Roll no.: CS22BTECH11061

Computer Architechture

Lab-3 (RISC-V Disassembler)

REPORT:

Program is divided into multiple parts for readability and functionality.

* In main() function -
  1) we call a fn to ready the file pointer.
  2) Then we go through file once to get the Instruction count (IC).
  3) we initialize a 2d array of strings Instructions[IC][33] to store binary instructions.
  4) we convert hex intruction to binary before storing in array.
  5) An int array labels[IC] is used to store labels.
     (Instead of L0 we store 0 in labels array ps that instruction which has that label. For no label we have value -1)
  6) After that we pass IC, labels arrays, instructions to disassembler() fn to disassemble.

★ In disassembler() function -

1) It runs the disassemble() fn for each instruction to seperately disassemble them one by one.

★ In disassemble() function -

1) From the given instruction it first checks the opcode
2) and calls the appropriate function for respective format type to disassemble the instruction (ex, R-format(), I-format(), etc)

★ In R_format(), I_format(), S_format(), B_format(), J_format(), U_format() functions -

1) This functions finds instructions respective rd, rs1, rs2 register numbers, imm value, etc. (acc. to format)
2) After that its funct3, funct7 values are checked to know the instruction name,
3) Then the instruction is stored in instructions[Ic][33] with instruction name as well as register names, imm value, labels etc, according to instruction type. (This instructions are stored & overwritten in instructions[Ic][33] array itself b/c 33 characters are enough to store any instruction)

4) We also label the instructions accordingly with appropriate value in labels array.

back in main() fn -

7) After all this, we call the print() function to print all the instructions with appropriate labels.
(with label data from labels array, labelling is done here)

8) Other functions like hexToBinary(), binaryToDecimalstring(), etc are created and used for convenience.

For Testing -

An input file with all the ~~asked~~ instructions in the problem statement is used. The hex code for them was obtained from ripes.