CS2323-lab 8(DIASSEMBLER)

REPORT

CS21BTECH11055

SADINENI ABHINAY

CODE APPORACH:

1)convert hex-decimal instructions to binary instructions

2)Segregation of instructions into R, I, S, U, B, J formats

Using opcode (last 7-bits)

Note:

funt7, funt3, immd, rs1, rs2, rd can be separated from instruction string.(do according to the format)

3)R-format: Map the function using funt7 and funt3 and use conversion to get register numbers

4)I-format: Map the function using funt3 and use conversion to get intermediate and register numbers (rd, rs1)

5)S-format: Map the function using funt3 and use conversion to get offset and register numbers .

6)U-format: lui instruction ,first 5 hex digits are the immediate.

7)B-format: Map the function using funt3 and use conversion to get offset, now convert the offset to label using the line number of the instruction, use the conversion register numbers.

8)J-format: use conversion to get offset, now convert the offset to label using the line number of the instruction

9)if there is any J-type or B-type then get the label from the instruction (from the offset)

10)check for label and if present add at the start of instruction.

Conversion:(functions are written at start)

1)hexa-decimal to binary format

2)binary to decimal format

Testing:

1)I used an “input.txt” file to give instructions set

2)I used all the labs which done before and gave them as input.

3)first individually tested for each function in the risc-v card.

4)Used ripes simulator to get the hexadecimal instructions

From assembly instructions.

5)After that added multiple instructions and tested labels

Results:

1)All instructions are translated properly.

2)Labels are attached at appropriate line.