LAB8 : REPORT

ABHAY SHANKAR K : CS21BTECH11001

Program flow:

- Program reads from default input file (for ease of debugging) input.txt.
- numpy.loadtxt() converts the 0x-prefixed hexadecimal strings in the file to integers, storres them in a list s[].
- One pass through s generates another list ops, which contain only the opcodes of all instructions.
- Using a dict, these opcodes are matched to their respective handlers in a key-value pair.
- Each handler extracts the various fields of the instruction, and uses more dicts to determine the exact instruction being executed (based on funct3 and funct7).
- For the shift instructions, some references indicate another field named funct6, to differentiate berween arithmetic and logical shifts. This is not treated as a separate field and extracted later.
- For each line of machine code, the required handler returns a string containing the corresponding assembly. This gets stored in a list, temp.
- One pass through the assembly code allows us to look for branches and insert labels. The naming convention is L{n} for the n'th encountered branch/jump. The labelled code is stored in out[].
- The contents of out are printed to the terminal (for ease of debugging) as well as the default output file, output.txt.

Testing:

- The code was tested using three assembly programs provided for Lab7 : Caches.
- Testing of signed correctness required some modifications to the given programs, and input.txt still contains those modifications.