LAB - 8 REPORT

Roll Number: CS21BTECH11022
Name: Janga Tushita Sharva

The program runs on file systems. So initially, the FILE * data type is used. Then other functions which help us access files are used, such as `fopen`, `fgets`, `fscanf`.

The program's inputs are 8 digit hexa-decimal strings. So, first it is converted into an array of 32 integers, and each is a binary bit. That array was named as `inputb[32]`.

1) Determining the opcode

First of all, the type of instruction is determined using the opcode. We know that the last 7 bits, or the bits occurring after `inputb[25]` determine the opcode. So a function `detType` is invoked, which returns a character depending upon the opcode.

2) Function Finding

Once the opcode is determined, there is a separate function for each format. We send the array of 32 integers to the respective functions. The new arrays of integers are created depending on the format. They are all assigned certain bits from the `inputb` array. Each format has several functions which are determined by `funct3` and `funct7` (or sometimes only `funct3` or sometimes nothing is required or sometimes a part of immediate value is required). Depending on the requirement, each function invokes another function, where the function is determined using the parameters required.

3) Finding the register numbers, immediate values etc

Once the function is found, most of the work is done. For this, a simple `for` loop and the `power` functions are used.

After the end of the program, the file is closed and the program terminates.

Description of the predicate functions:

`power(int base, int index)` : returns $(base)^{(index)}$

`convert(char input[], int inputb[])`: It does not return anything, but arrays are subject to change when they are passed through functions. So here, after the function, the `inputb[]`, which initially has nothing, will become an array of 32 integers.