CSCI 6461 Section 10 Fall 2023 Group 10 Project IIa – Design Document

Design Notes of the Code :

*Refer to the **CSCI 6461 Section 10 Fall 2023 Group 10 Project I – Hand Guide** for the information about design notes of project I. This document is a continuation of project I. Hence, we only consider the design notes/classes introduced as part of project IIa.

The objective of project IIa is to translate high-level instructions into hexadecimal codes. As part of this, the new classes that we introduced are :

<u>Translator.java:</u> This class is core to the functionality of translating high-level instructions to hexadecimal instructions. It makes use of the metadata file called "InstructionsMetaData.json". It reads an array of elements that are part of this file and convert it into an instance list of type "InstructionsMetaData" class. It makes use of fields present is each of the elements in the list to determine if the instruction is of 1 or 2 or 3 or 4 length and accordingly decodes the instructions.

We store the decoded instructions into a LinkedHashMap so that we return this object to FileHandler.java* to write into **HexFile.txt**.

<u>InstructionsMetaData.java:</u> InstructionsMetaData instance contains the information that each of the element that is contained in the InstructionsMetaData.json. It is just storing the content of the json file into the "memory" and using it later to convert high-level instructions into hexadecimal code.

<u>InstructionsMetaData.json:</u> This json file is a template to all the instructions that we incorporate as part of high-level instructions. It has three fields namely bytePattern, instructionsPart, opCode. Where instructionsPart determines number of fields that we sent in instruction other than opcode. opCode determines the opcode of the instruction and bytePattern determines the predefined byte pattern(16 bits) that we use to decode the instruction into hexadecimal.

<u>FileTypes.java:</u> Since we need to deal with different files like json and text, we introduced this **enum** to help our program determine which file to interpret. It has file types of JSON and TXT.

<u>HexFile.txt:</u> This is the result file where we store our instructions that are in the hexadecimal representation. This file is the actual input to our program. Our simulator only understands hexadecimal, and hence generation of text in this file is crucial to run our simulator.

Other Changes Include :

<u>Opcodes.java:</u> We have introduced more opcodes in this enum to determine the instructions.

*Keeping the InstructionsMetaData.json, HexFile.txt, ProgramLoadFile.txt and the jar file in a folder is essential for the program to run.