William Lin

May 8, 2017

CSCI 2214-02

Project 2 Report

DLX Hardware Simulator

The project was successful. My simulator runs both test files correctly and other test files that I create. I use a Simulator class object to run the simulation and the Main Memory class object simulates the main memory.

**MainMemory:**

* loadProgramCode(string) – Takes an input filename as a string parameter. Attempts to open and read instructions from the given input file into the instruction section of main memory.
* readMemory(int, int) – Takes two ints as parameters. The first int is the address to read memory at. The second int is either 0, 1, or 2 which will specify whether to read a word, halfword, or a byte.
* writeMemory(int, int, int) – Takes the address of where to write into memory, the data to write into memory and the size of data that is to be written. The size of the data can either be 0 for a word, 1 for a halfword, or a 2 for a byte.

**Simulator:**

* run(bool) – Function to run the simulator after main memory has been loaded instructions. Takes a bool which will determine whether run the simulator step by step or all at once. If bool is true then it will run step by step. Using control signal functions, runs each line of instruction based on the opcode and by turning on and off control signals.
* void loadCode(string) – Takes an input file name and passes it to MainMemory loadProgramCode to load the instructions of the input file into main memory.
* void printStateHex – prints the state of the simulator all in Hex. The state includes all control signals and hardware component values like MDR and RF.
* void printStateDec – prints the state of the simulator all in decimal
* void printStatePretty – prints the state of the simulator with a mix of both hex and decimal. Hardware such as IR which is better viewed in hex is printed in hex and hardware that are better in decimal like register values are printed in decimal.
* int getInstrType(int) – takes an opcode as the parameter and returns an int that corresponds with the instruction type that the opcode is.
* void step1() – uses the correct control signal functions to use the program counter and get the corresponding instruction from memory and load it into the instruction register.
* void step2() – Uses the correct control signal functions to update the PC by 4.