**Mazen Amr EID**

**900161021**

**Khaled Soliman**

**900162031**

**RISCV 32 IC Simulator Report**

**Design**

Our purpose from the very start was to create a universal ISA simulator working cross-platform on any ISA the user gives to it. In fact, we have a human friendly configuration file which contains all the RV32IC needed information that we can replace if wanted to with any other instruction set and our simulator would still work.

**Implementation**

Our program is composed of these main classes and concepts: Fetcher, Decoder, Decompressor, Simulator (Executor). Thus, we process the raw binary file and fetch every instruction, then pass it to the Decoder which decodes the binary instructions into Instruction objects which contains all the information it needs about the instruction by matching with the yaml file(instruction name, format, opcode, funct3/funct4/funct7, register s1, register s2, register d, immediate). This allows the Decompressor to easily understand what are the different fields the instruction has and print it on screen. Afterwards, the simulator takes this same Instruction object with its encoded instruction and matches the name of the instruction with one of its keys mapping to the execution function, as we have an implementation for every instruction in the RV32IC in a class called Instructions (note that RV32C ‘s instructions map to its equivalent uncompressed instruction).

**Limitations**

Unfortunately, we didn’t finish the compressed isa. In addition to that, our simulator is hardcoded (unlike the Decoder, and the Disassembler) so it is only compatible with RV32IC architecture. We didn’t also have the time to implement a user interface or add additional support to any other ISA (for instance, MIPS) but all of that will come later.