**Bisrat Asefaw**

**CSS 422 (Hardware)**

**Prof. Yang**

**Individual Evaluation**

***Bisrat***

In the disassembler project that is submitted by our team, I implemented opcodes for NOP, RTS, NOT.B, NOT.W, NOT.L, JSR, LEA, MOVEM.W, and MOVEM.L. I have implemented the EA’s for those opcodes, and they are printing the required format to consul. I also come up with a strategic plan of divide and concur way of tackling the project so that every team member can have a chance to participate equal in the workload. Based on the strategic plan I designed the high-level architecture, and all my teammates accept it happily and takes it as a base diagram and work from that. Since every team member is responsible for testing his/her assigned opcodes, I have to test individually for both invalid and valid opcodes. By the end of the project, we as a team prepare the slides, final project document and presentation.

***Charlie***

worked on implementing IO, she did a good job of handling errors, user input handling, and providing the 16-bit hex decimal representation of the opcode. She also writes methods that can convert and print hex to character string and from character string to hex decimal for a purpose of printing and saving user input. She implements the opcodes Bcc, BLT, BGE, BEQ, BRA, and partially on SUB. She uses her own EA handling for different cases to print out the correct EA’S for BRA and Bcc’s. By the end of the project, we as a team prepare the slides, final project document and presentation. Since every team member is responsible for testing his/her assigned opcodes, she has to test individually for both invalid and valid opcodes.

***Sean***

worked on implementing the branch opcode to branch each opcodes to its subroutine depending on the first 4 bits of the instruction. He implemented the opcodes Muls.w, move.b/w/l, and sub. He also implemented the EA\_print subroutine to implement EA’s of opcodes that has both source and destination EA’s. He also did a good job of decoding every instruction in D5-D7. Since every team member is responsible for testing his/her assigned opcodes, he has to test individually for both invalid and valid opcodes. By the end of the project, we as a team prepare the slides, final project document and presentation.

***Tomomi***

Worked on implementing the opcodes ASR, LSL, ADD. She also implemented her own subroutine to implement her assigned opcodes. She participated in creating the high-level design in Lucidchart. Since every team member is responsible for testing his/her assigned opcodes, she has to test individually for both invalid and valid opcodes. By the end of the project, we as a team prepare the slides, final project document and presentation.

|  |  |  |
| --- | --- | --- |
| Member | Codes (%) | Task |
| Bisrat | 25 | * Implements both opcodes and EAs for NOP, RTS, NOT.B/W/L, JSR, LEA, MOVEM.W/L, MULS.L * Did high level architecture design and documentation |
| Charlie | 25 | * Implemented the input handling for getting the starting and ending addresses. * Decoded Bcc, BLT, BGE, BEQ, BRA * Did some decoding of SUB |
| Sean | 25 | * Decoded Muls.w, and, move.b/w/l, sub, * EA work for EA print * wrote code to break down the opcode and load info about the code into Dn’s 4-7. |
| Tomomi | 25 | * Opcode and EAs for ASR, LSL, ADD * some EA work with EA print. |