Date: 3/5/16

To: CSS 422, Winter 2016

From: Easy Riders

Subject: Team progress report 4

Time spent doing thorough advance planning has paid off with rapid development this week. We met our goal of implementing 10 opcodes with all effective addressing modes now working. It became clear as the logic was being implemented for the various opcodes that several of them fall into similar patterns where much of the implementation can be passed off to common subroutines and it's just a matter of calling the correct subroutine order to get proper decode output.

Work Completed:

Sprint 4

- Project General
    - Notes: Full audit of project specifications and deliverables.
    - Status:
        - Completely refactored code into separate files, standardized commenting, standardized naming. td 2/29
        - Added several evalbits and print subroutines for opcodes. td 3/2
        - EA decoding completed, needs integration. rh 3/5
- I/O
    - Notes: Input branching complete. Need to test on non-required opcodes.
    - Status:
        - Found and corrected decode errors for some opcodes. Input branching tested and complete for assigned and chosen optional opcodes. td 2/29
        - Updated root branch diagram. td 3/5
        - Updated branch 0 diagram. td 3/5
        - Updated branch 4 diagram. td 3/5
        - Created branch 5 diagram. td 3/5
        - Updated branch 8 diagram. td 3/5
- EA
    - Notes: Finish immediate addressing this week
    - Status:
        - Formatting and standardization coding pass for EA and subroutines. td 2/29
        - Completed Immediate address decoding -rh 3/5
- Opcodes
    - Notes: Finish minimum 10 this week
    - Blockers: Need Immediate Data EA mode to complete testing. td 3/2

- Status:
  - ADD opcode complete, fails extended test 19. td 3/1
  - ADDA opcode complete. td 3/2
  - ADDI opcode complete. td 3/2
  - AND, ANDI, ASL, ASR, LSL, LSR, ROL, ROR opcodes complete. All SHIFTROT opcodes fail memory shift tests. td 3/2
  - Updated MOVE.B, MOVE.W, and MOVE.L, needs testing. rh 3/5

The specifications audit brought to light some implementation details that we need to address, such as the requirement for loading additional alternate addresses with subsequent decode executions. Fortunately, our codebase should support this without much fuss.

We are currently looking at a split focus to complete remaining work:

- Thomas's first priority will be ensuring that the command parser performs and handles all interaction and output according to the project specifications. Second priority is to fold in data from all progress reports to build the project final report and presentation.
- Ross's first priority is to finish coding on the remaining opcodes. Second priority is in-depth testing of each opcode under all effective addressing modes.

Third priority for both of us is bug triage.

- All priority 1 (causes a crash) bugs must be eliminated.
- All priority 2 (improper output in general testing) must be addressed: fixed, or acknowledged and deferred.
- Select priority 3 (improper output under specific conditions) can be addressed as time allows.

Problems:

There are no problems to report for sprint 4.

Work Scheduled:

- Implement, check, or modify the disassembler as required for items identified during the specifications audit.
- Complete remaining opcodes: MOVEQ, MOVEM, SUB, MULS, DIVU, LEA, CLR, CMP, BCC, BGT, BLE, and JSR
- Create materials and documentation for the final project report and presentation.
- Extended testing and bug triage.

Self Evaluation:

We have completed enough of the disassembler project where it is possible to implement the remaining requirements and opcodes in the following week before the project is due. However, it remains to be seen how much test and debug will happen with other competing final projects from other classes also due next week. We have made good progress and seem to be on track, but I don't expect to be done early.