Date: 02/28/16

To: CSS 422, Winter 2016

From: Easy Riders

Subject: Team Progress Report 3

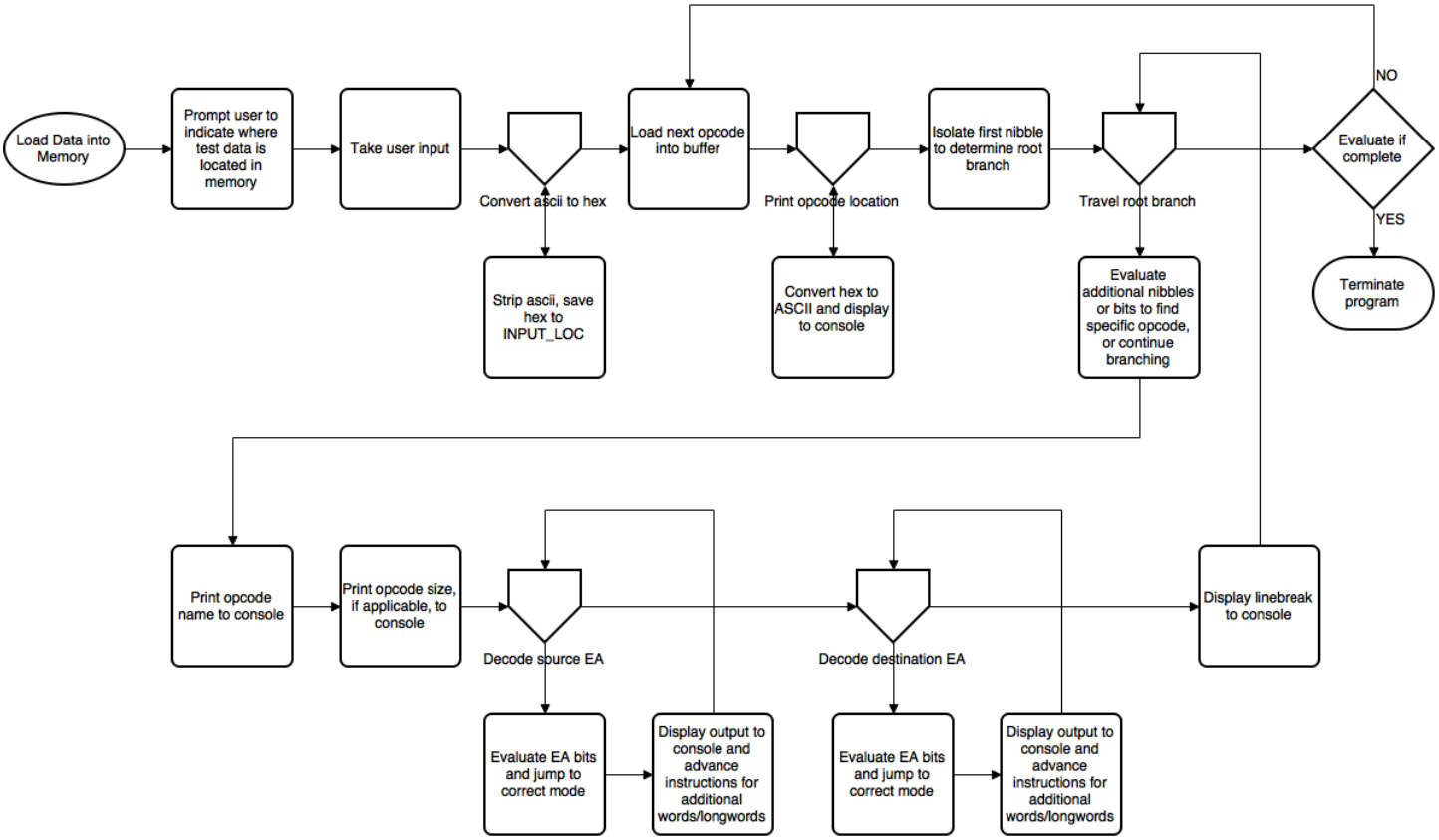
This week did not see a lot of direct rapid progress on the project because of time resource limitations, however the important task of merging the prototype into the mainline branch did occur, and input branching and opcode processing are nearing completion.

Work Completed:

Sprint 3

- Full audit of project specifications and deliverables
    - Delayed until sprint 4. td 2/28
- Need to build full project timeline
    - Removing from project list due to lack of time. td 2/28
- Built interactive client interface
    - Status:
        - Imported work from HW5 into project to decode ASCII and take user input to define location of imported tests. td 2/24
        - Added ability to pause output when screen becomes full. td 2/27
- Implement minimum 5 new opcodes.
    - Status:
        - Delayed until sprint 4. td 2/28
        - Did full pass standardizing register use and merged prototype into mainline code instead. td 2/27
- Opcode processing
    - Status:
        - Opcode detection and routing is almost complete. td 2/27
        - A testing module with all assigned opcodes can be loaded and processed to evaluate basic decoding. td 2/27
- Finish effective address decoding subroutine
    - completed EA mode 000 rh 2/28
    - completed EA mode 001 rh 2/28
    - completed EA mode 010 rh 2/28
    - completed EA mode 011 rh 2/28
    - completed EA mode 100 rh 2/28

Special emphasis was placed on branching execution for both opcode decodes, but for program execution in general. In particular, we verified that while we travel and branch inside of subroutines, program flow remains in a single main loop which we return to. When it is detected that there are no additional opcodes to process, the loop breaks and the program terminates immediately outside of the main loop.



The above flowchart illustrates the general flow of the disassembler, including subroutine movements. It does not show the detailed analysis of how an opcode is evaluated, which has been shown in previous team progress reports.

Problems:

While the disassembler project is coming along, both Ross and I have been overwhelmed with exercises and homework, much of it from this class. And while much of it is useful, it has taken priority over this project. I feel that even though coding progress has had a slow start, what we do have is very well executed, robust, and modular. However, the scope of the project is very big and regardless of how well thought-out our implementation is, there are unavoidable aspects of coding it that are time consuming busy work which may prove hard to accomplish if other large assignments are introduced.

Work Scheduled:

- Full audit of project specifications and deliverables
    - Notes: Needs to be done so detail specific goals are met and can be taken into account. td 2/28
- Finish input branching
- Finish effective address decode subroutines
- Implement and test opcodes, minimum 10.

Self Evaluation:

Our foundation is solid, but significant progress on opcode processing needs to happen in sprint 4 to allow time for test and debug.