**Meeting Minutes**

**Date:** March 15, 2014

**Start Time:** 2:00pm

**End Time:** 5:00pm

**Members Present:** Drew Aaron, Michael Beaver, Clay Boren,

Chad Farley, Andrew Hamilton, and Travis Hunt

**Members Absent:** N/A

**Topics** **Discussed**

* Frontend Development
* Backend Detailed Design

**Decisions and Actions Taken**

Drew, Clay, and Travis worked on Frontend development. For the moment the problem of line numbers has been shelved in favor of completing simpler tasks. Drew, Clay, and Travis were able to get most of the external file I/O functionality working. For example, they implemented the project open and save functionality. The print functionality is proving to be somewhat more challenging than initially expected.

Chad, Michael, and Andrew began developing the ErrorDetection class Detailed Design. The team has opted to remove the segregation of the Assembler and Simulator ErrorDetection classes and has opted to consolidate them into one class. Also, an ErrorDetection module object will be instantiated in the Backend Processor to be used by the Assembler and the Simulator. The team needs to ask the client if the Simulator must track the Program Status Word. The team must also ask the client about the “IM” column of memory dumps. Similarly, the team needs to know if 10 is a sufficient number of instructions to track in an instruction trace (for crash dumps). As part of the ErrorDetection class, there will also be an InstructionsTrace class. There will be an InstructionsTrace object in the ErrorDetection class’ data members. See the attached design particulars.

There will be an optional team meeting Monday, March 17, 2014 at 10:00am in the Christian Student Center. The next mandatory team meeting will be Tuesday, March 18, 2014 at 3:30pm in the Commons building.

**Supplementary Information**

**ErrorDetection Data Members**

Exception Throwables;

Queue InstructionsTrace;

**ErrorDetection Methods**

public ErrorHandler(int errorCode, int lineNum)

Process errorCode (switch)

Call appropriate error method, pass in lineNum

private AddressingErrSim()

private DataErrSim()

private DecimalDivErrSim()

private DecimalOverflowErrSim()

private FixPntDivErrSim()

private FixPntOverflowErrSim()

private OperationErrSim()

private ProtectionErrSim()

private SpecificationErrSim()

Note that exceptions have a one-to-one relationship with errors for the Simulator.

Assembler error methods will be determined at a later date.

**InstructionsTrace Design**

The InstructionsTrace is essentially a Queue of “Listings.”

Listing is a struct:

string IM

string Instruction

string Location

const int MAX\_INST\_TO\_LIST = 10;

The InstructionsTrace will modify the push() method of the Queue class to pop first if there are MAX\_INST\_TO\_LIST items in the queue.

**Short-term Goals and Schedule**

Remaining:

Refine Translator

Refine Library

Assembler Errors

Frontend Functionality

Refine Data

Symbol and Literal Tables

Schedule:

Monday, 10:00-1:30, CSC

Tuesday, 3:30-6:00, Commons

Thursday, 3:30-6:00, Commons

Friday, TBD