**Memo: Current Status of Processor**

Section 1, Team 1a

Adam Finer, Runzhi Yang, Katrina Kerrick, Fred Zhang

October, 6, 2015

We have designed a load-store instruction and a stack. Together, these can do function calls, recursion, exception handling and complex mathematics.

We have 4 different instruction formats -- R-type, I-type, B-type and J-type. Every instruction is one of those types.

We have 16 registers, including 8 $t’s, 3 $s’s, 2 $v’s, $ra, $sp, and $at.

We used our set of assembly language instructions to write several example programs.

We have written an assembler to help us convert assembly language into binary instructions.

We might make some changes to our instruction set, like deleting subi and including push and pop to make the common case fast (this is because you must increment or decrement $sp individually for each stack access anyway). This is extremely important because though our instruction set is mostly complete, it is not set in stone. We might end up needing shift instructions, too.