# Project Progress report: Pipeline Mechanics

## Abstract

To progress with the evaluation of the MIPS64 Pipeline mechanics, the MIPS64 EDU environment was examined. TALK ABOUT HOW WE STARTED TO CHANGE FORWARDING. Also, a set of test bench assembly files were created that generate read after write dependencies that can be resolved with forwarding.

## Work Completed

### The MIPS64 EDU environment

The MIPS64 EDU environment is a platform of choice to examine the various functional differences of the MIPS64 pipeline. This java base simulator provides a set of tools to examine the pipeline, where instructions are stalled, the contents of the registers, and the overall structure of memory. Using all of these tools we are able to evaluate the performance of our pipeline as we make changes to its structure.

### Implementing forwarding

### Creating a Test Bench

To determine if the different forwarding schemes applied to the standards MIPS64 pipeline offer any sort of improvement, a set of standard test cases would be required.

In order to test if the pipeline will stall on a read after write (RAW), the test bench should be able to generate this type of error. After implementing forwarding, these test cases can be run again, and the number of stall cycles compared for each type of pipeline.

Two RAW data hazard test benches were written. The first creates a true data dependency between two registers that can be solved by forwarding to the EX stage. The second creates a dependency between two registers needed for a load and store operation, which can be solved by forwarding from the EX stage to the MEM stage and from the MEM stage to the EX stage. For the second test case, since memory is being used, there will still be at best one RAW stall.

The following code snippet shows a portion of data hazard test bench 1:

;Do Useful work

;Without forwarding 2 RAW stalls expected due to R1 dependency

;Adding forwarding from EX to back to EX will solve this stall

DADD R1, R2, R3 ;Put something in R1

DADD R4, R1, R5 ;Dependency on R1

DADDI R3, R3, 0 ;Execute filler instruction

SYSCALL 0 ;Exit program

### Work Remaining

### Issues Encountered