*Note all benchmark times were run with hyperfine					
Benchmark	Time (s)	Instructions	Rel to Start	Rel to Prev	Improvement
Big	804.187	2,113,497,561	1.000	1.000	
Small	33.524	85,070,522	1.000	1.000	No improvement (starting point)
Big	220.010	2,113,497,561	0.274	0.274	
Small	8.487	85,070,522	0.253	0.253	Compiled with link-time optimization turned on
Big	35.089	2,113,497,561	0.044	0.159	
Small	2.592	85,070,522	0.077	0.305	Added the check in load program to check if $r[B] != 0
Big	6.915	2,113,497,561	0.009	0.197	Changed the memory representation to be a Vec<Vec<u32>> instead of a HashMap u32, Vec<u32>. We found that the times while running hyperfine multiple times gave times anywhere from 6.5 to 8.5s.
Small	0.330	85,070,522	0.010	0.127	
Big	6.093	2,113,497,561	0.008	0.881	Replaced instanced of r[instr.ra] and for rb and rc in the machine.rs match statement with variables outside of the match statement. Seemed to have mixed results. Sometimes it would result in 7s while other times it would be around 6s wth a standard deviation of 0.337s.
Small	0.347	85,070,522	0.010	1.052	
Big	5.688	2,113,497,561	0.007	0.934	New get_instruction function for the new vec implementation of memory. Mainly reduced an unneccesary clone. Seemed to have the most improvements on larger inputs.
Small	0.328	85,070,522	0.010	0.945	
Big	6.567	2,113,497,561	0.008	1.155	Removed some instances of assert functions that were placed in the deallocate and allocate functions. It did not make any noticeable change in the timing of the programs.
Small	0.325	85,070,522	0.010	0.991	
Big	6.551	2,113,497,561	0.008	0.998	Combined the parse_opcode and decode functions to reduce function calls overhead. Decreased in sample size by 5% when testing with Samply, but it did not have a noticeable change when timing
Small	0.349	85,070,522	0.010	1.074	
Big	6.679	2,113,497,561	0.008	1.020	Made applicable match statements in run function loop inline. Did not make a signifcant change when timing and produced mixed results. Change reverted.
Small	0.360	85,070,522	0.011	1.032	
Big	6.404	2,113,497,561	0.008	0.959	Assigned memory.get_instruction(pc) to a variable so it wouldn't be checked more than once in the instr match statement. Did not have a noticeable change when timing.
Small	0.334	85,070,522	0.010	0.928	
Big	6.304	2,113,497,561	0.008	0.984	Instead of using the get and get_mut functions we decided to access directly through the index assuming that the indicies are always valid. This helps to avoid those funtion calls and the use of unwrap as well because we avoided the option from "get" call.
Small	0.285	85,070,522	0.009	0.853	