Midterm Report

ECE 437: Computer Design and Prototyping

Kyle Woodworth

Allen Chien

Adam Hendrickson

3/6/2015

Overview

For the lab portion of the ECE 437, we are required to design a single cycle processor and to implement a pipelining system for the overall design using MIPS instruction set (ISA). The idea of implementing the pipelining system is to increase the processor's throughput of instructions. Although the single cycle processor design is relatively simple, it only carries out one instruction per one clock cycle, resulting in a very low throughput of instructions and a slow overall speed of the processor. By implementing a pipelining system, we are able to increase the speed of the processor but also introduce different types of hazards. In order to resolve the hazards, we implemented a hazard detection unit and a forwarding unit to correctly handle each hazard and forward the correct values when needed.

MergeSort program is a perfect choice to compare our processors because it not only includes multiple branch and jump instructions but it also contains instructions that require special data handlings and detections in various orders. Using the MergeSort program, we can thoroughly test our single cycle processor and pipelined processor. The test will be carried out by modifying the clock period of the system testbench with memory latency of zero. By analyzing the average instructions per cycle, average instruction latency, and the FPGA resource usage from the REPORT generated by the synthesized processor, we are able to compare the two processors' performance. After running our tests, we concluded that implementing a pipelined processor increased throughput, CPI and decreased the clock period.

Processor Design

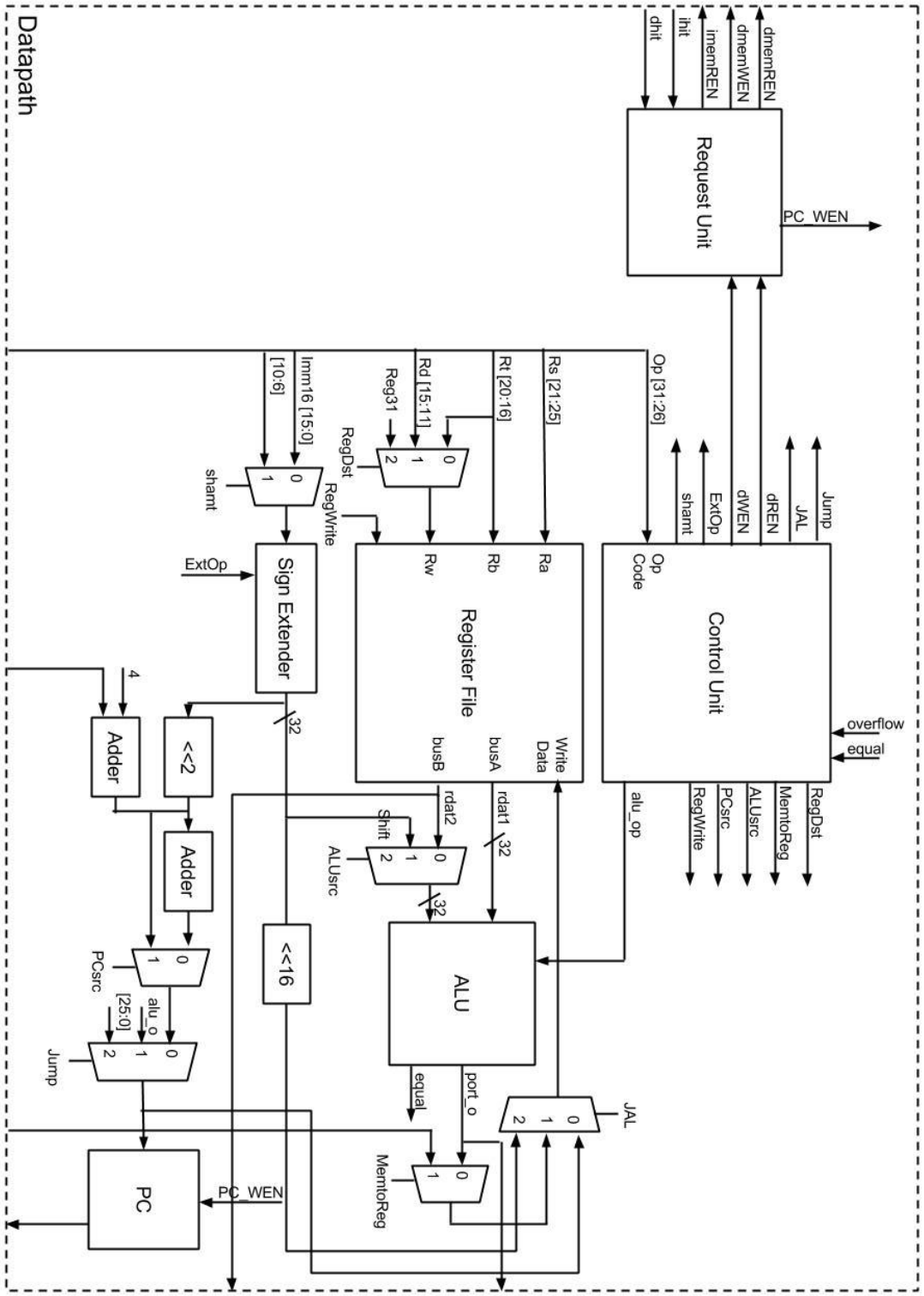

Figure 1: Single Cycle Processor Block Diagram

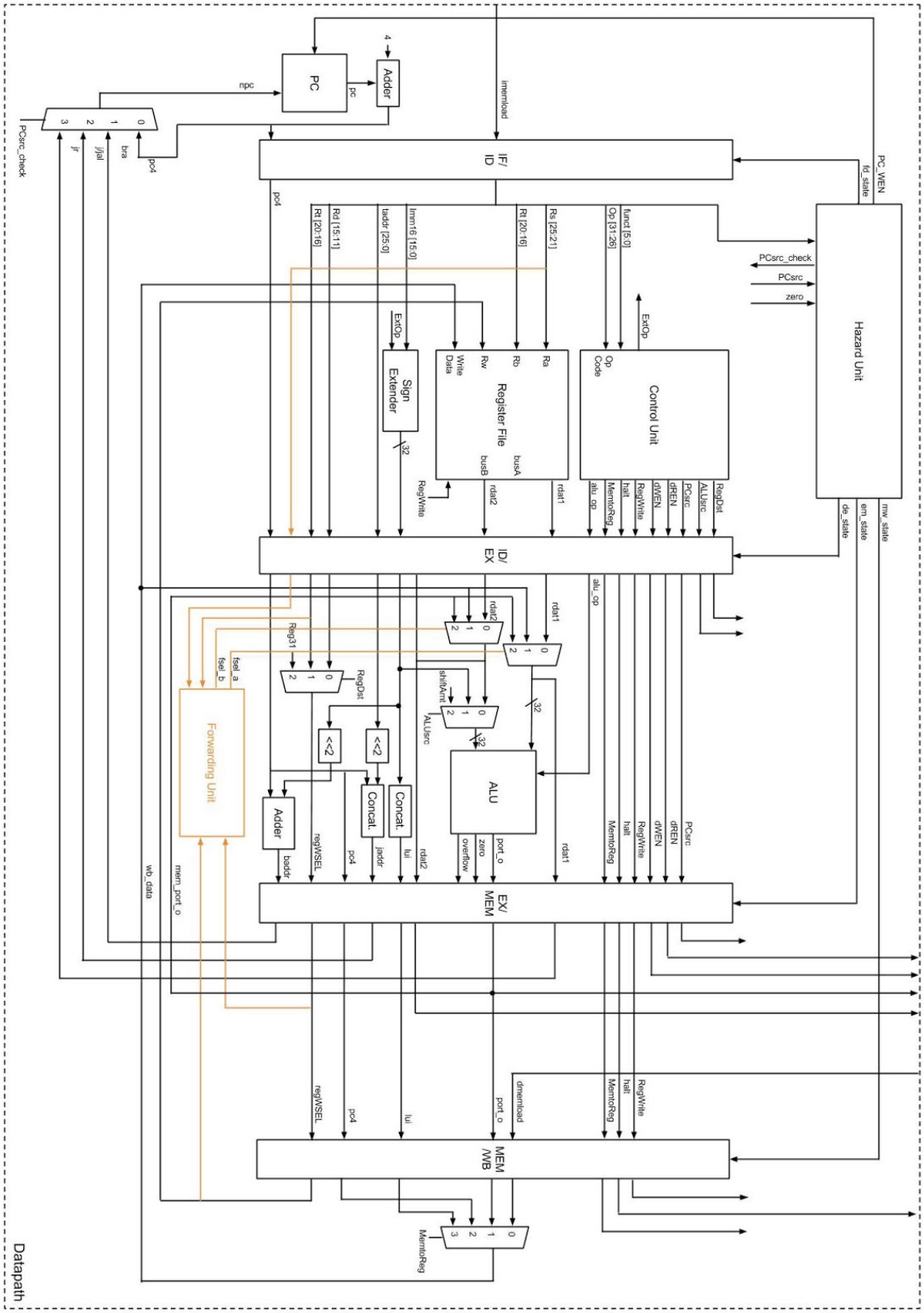

Figure 2: Pipelined Processor Block Diagram

Results

	Pipelined Processor	**Single-cycle Processor**
Frequency	83 MHz	50 MHz
Avg. Instr./Cycle	0.309 Instr./Cycle	0.391 Instr./Cycle
Avg. Instruction Latency	38.8 ns	51.17 ns
MIPS	25.647 MIPS	19.55 MIPS
FPGA Resource Usage: - Total logic elements - Total combinational functions - Dedicated logic registers	 4,428 3,309 1,789	 3,834 2,847 1,279

Conclusions

From the data above, we can see that our pipelined processor had a significant improvement in terms of performance. Our single cycle processor had a much higher average instruction latency and that the pipelined processor was able to run a little over 6 million more instructions per second than the single cycle processor can. Since the average instructions per clock cycle is the inverse of the CPI, we can conclude that our single cycle has a better CPI than our pipelined CPI. This is expected since it takes 5 clock cycles in a 5 stage pipeline to complete most instructions. Pipeline implementation allows us to increase our clock speed, this really means that we are segmenting the critical path of the single-cycle into stages. After the implementation of the pipes, our critical path will reside in one of the five stages.

Since we are using a common binary file, our MIPS metrics become a useful measurement. The MIPS metric simply describes the throughput of the processor. The single-cycle processor MIPS is 19.55, and after the pipeline implementation the MIPS increases to 25.647. The execution time of the single-cycle was 500740 ns, while the execution time for the pipeline processor was 379692 ns. We can calculate the performance and say that the pipelined processor is 32% faster than the single-cycle processor. In conclusion, the implementation of pipelines from a single-cycle processor gave an increase in clock speed and throughput, but made sacrifices in area and CPI.