ECE 437L Midterm Report

Pooja Kale (mg273)

Brian Rieder (mg267)

TA: Nick Pfister

October 14, 2016

1 Executive Overview

This report compares the performance of a single cycle processor and a pipeline processor on a merge-sort assembly (asm) file, and analyzes the results. The comparisons of the two processor designs will be based on the maximum clock frequencies, average instructions per clock cycle, the performance of the designs in MIPS and the FPGA resources required for each design. The comparisons will be further tested using variable latency RAM in order to ensure independence from memory timing. The single cycle processor is a simple design that takes in one instruction at a time to decode it and produce a result. This design does not have any instruction dependencies eliminating the possibility of instruction or data hazards. The pipeline processor is a more intricate design with five stages: instruction fetch, instruction decode, execute, memory, and write-back. This allows the design to process five instructions simultaneously within different stages of their execution.

The two designs will be tested using the merge-sort assembly file. The benefit of using the merge-sort assembly program, is it includes all possible assembly instructions that both processors can decode. The merge-sort file also includes multiple branches to test branch logic in both designs. The order of instructions in the program will test if the pipeline handles data hazards, and structural hazards with inserting stalls or bubbles in the correct places and flushing data at the right time. The execution results show that the pipeline processor is the favorable design because it increases the throughput (or decreases the execution time) as compared to the single cycle. While the single cycle handles one instruction at a time the pipeline processor design processes 5 instructions in different stages in one cycle.

2 Processor Design

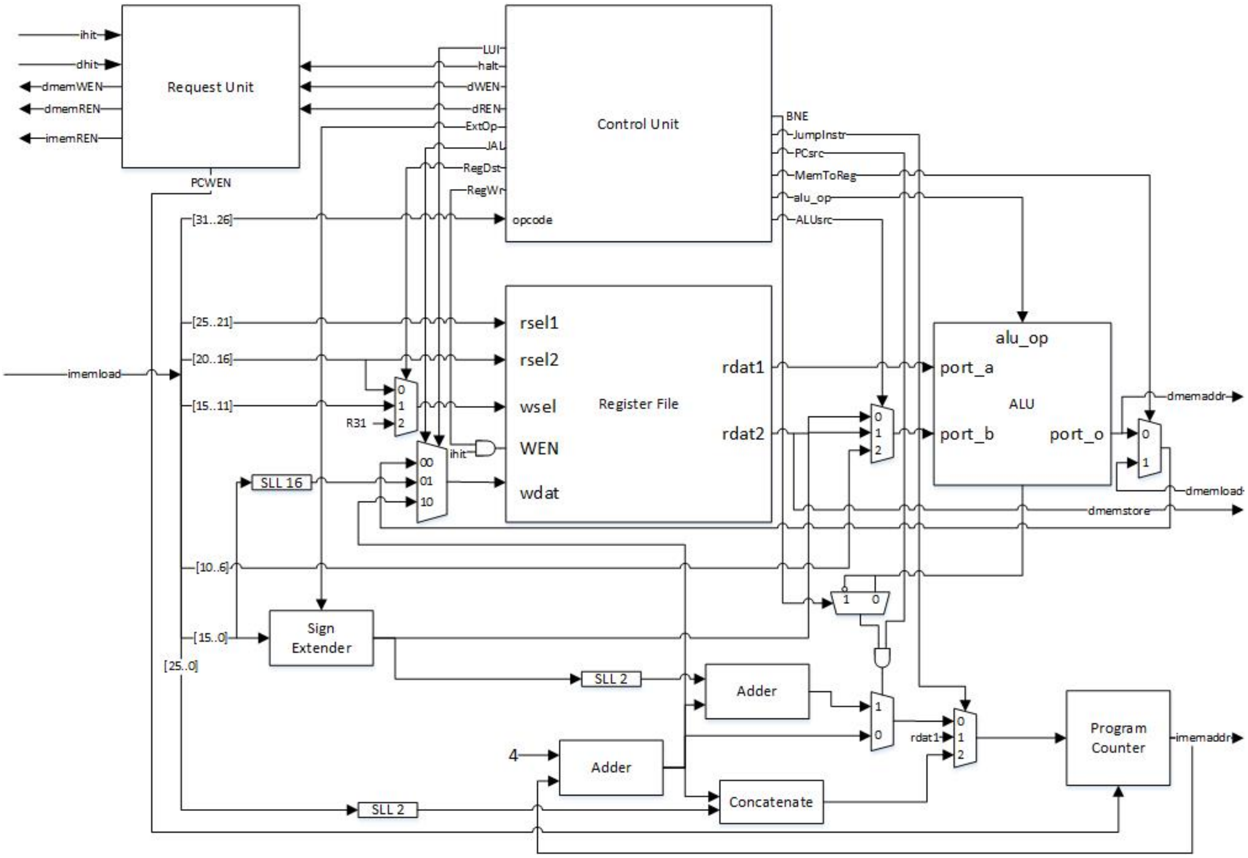

Figure 1: Single Cycle Processor Block Diagram

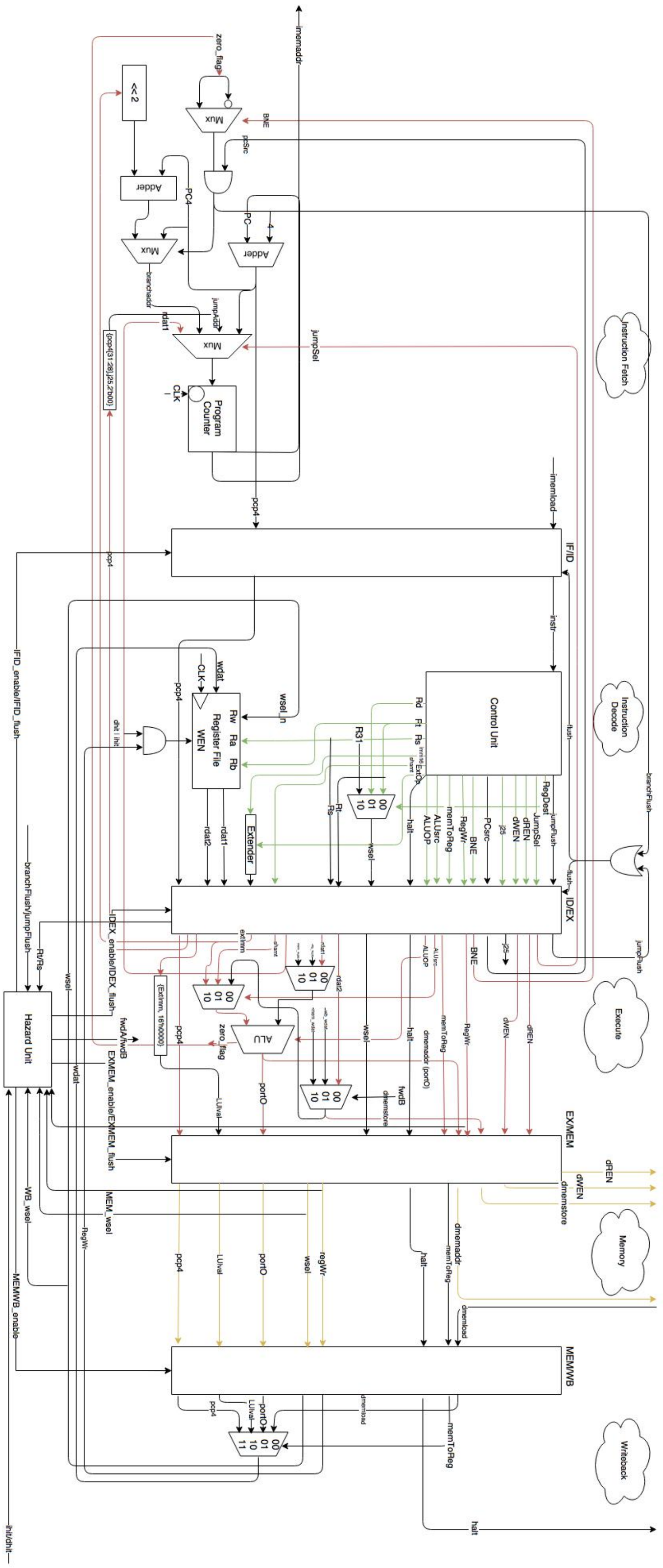

Figure 2: Pipelined Processor Block Diagram

3 Results

	Single Cycle	**Pipeline**
Maximum Possible Frequency (MHz)	37.24 (CLK/2)	48.88 (CPUCLK)
Highest Achieved Frequency (MHz)	55.56	100
Average Instructions per Clock Cycle (CPI)	2.17	3.216
Critical Path Length (ns)	26.939	20.774
Instruction Latency (ns)	26.85	102.29
Mergesort Number of Instructions	5,399	5,399
Mergesort Number of Cycles	13,791	17,365
MIPS	17.164	15.197
Total Utilized Combinational Functions	2,844	3,265
Utilized Dedicated Logic Registers	1,278	1,716

Table 1: Processor Specs

3.1 Sources of Result Values

- **Maximum Possible Frequency (MHz)** - Calculated by synthesis tools and retrieved from system.log

- **Highest Achieved Frequency (MHz)** - Determined through variation of PERIOD parameter within testbench/system_tb.sv

- **Average Instructions per Clock Cycle (CPI)** - Calculated using the total number of instructions retrieved from “sim -t” divided by the total number of cycles shown by “make system.sim”

- **Critical Path Length (ns)** - Calculated by synthesis tools and retrieved by system.log

- **Instruction Latency (ns)** - Calculated by multiplying period by a multiplier based on processor design - single cycle processors have a latency multiplier of 1 and a five-stage pipelined processor has a latency multiplier of 5 times

- **Mergesort Number of Instructions** - Determined using the total number of instructions output by “sim -t” when executing asmFiles/mergesort.asm

- **Mergesort Number of Cycles** - Determined using the total number of cycles output by “make system.sim” when executing asmFiles/mergesort.asm

- **MIPS** - Calculated using the inverse of CPI multiplied by the frequency of the processor

- **Total Utilized Combinational Functions** - Determined by synthesis tools and retrieved from system.log

- **Utilized Dedicated Logic Registers** - Determined by synthesis tools and retrieved from system.log

4 Conclusion

Text for conclusion

5 Contributions

Text for individual contributions