Midterm Report
ECE 437

Team: Fengjian Pan, Xihui Wang
Teaching Assistant: Eric Schrock
Due Date: October 16

Overview

In the past weeks, we designed and realized two different processors based on MIPS. One is single cycle processor, and the other one is pipeline processor. The pipeline processor is optimized based on the single cycle one, and brings us a big difference.

Each of the two processors has its own advantages and disadvantage. Single processor is easy to design and understand, since the instructions are run one by one. On the other side, pipeline processor has high complexity and cost, since multiple instructions will run within a same cycle, for our case, at most five instructions will run together. But it would decrease cycle time and increase CPI, therefore reduce the throughput, and improve the performance of groups of instructions.

In order to analyze the data gathered for each processor, we choose mergesort.asm as the test file, since it is complex, has long execution time, and has all types of instructions. In particular, we compare the performance of the two processors by comparing MIPS value, and we can calculate it by $MIPS = \frac{instruction\ \ count}{execution\ \ time \times 10^6} = \frac{frequency}{CPI \times 10^6}$. The result shows that pipeline processor has larger MIPS value than single cycle processor, so that pipeline performs better than the other.

Both of these processors have their merits, but based on the performance, pipeline processor is obviously better than single cycle processor.

Processor Design

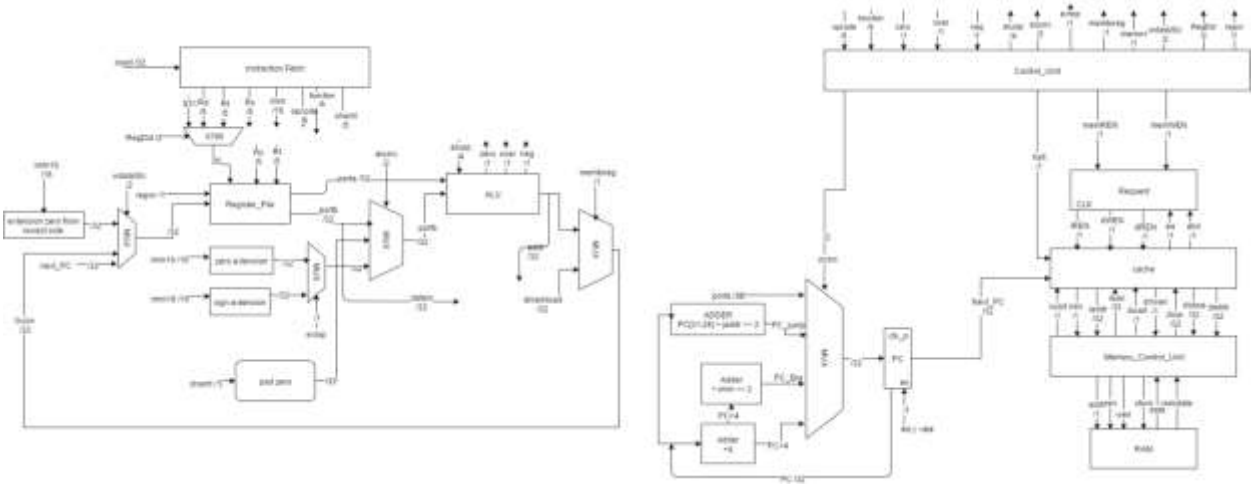

Fig-1 Block Diagram for Single Cycle Processor

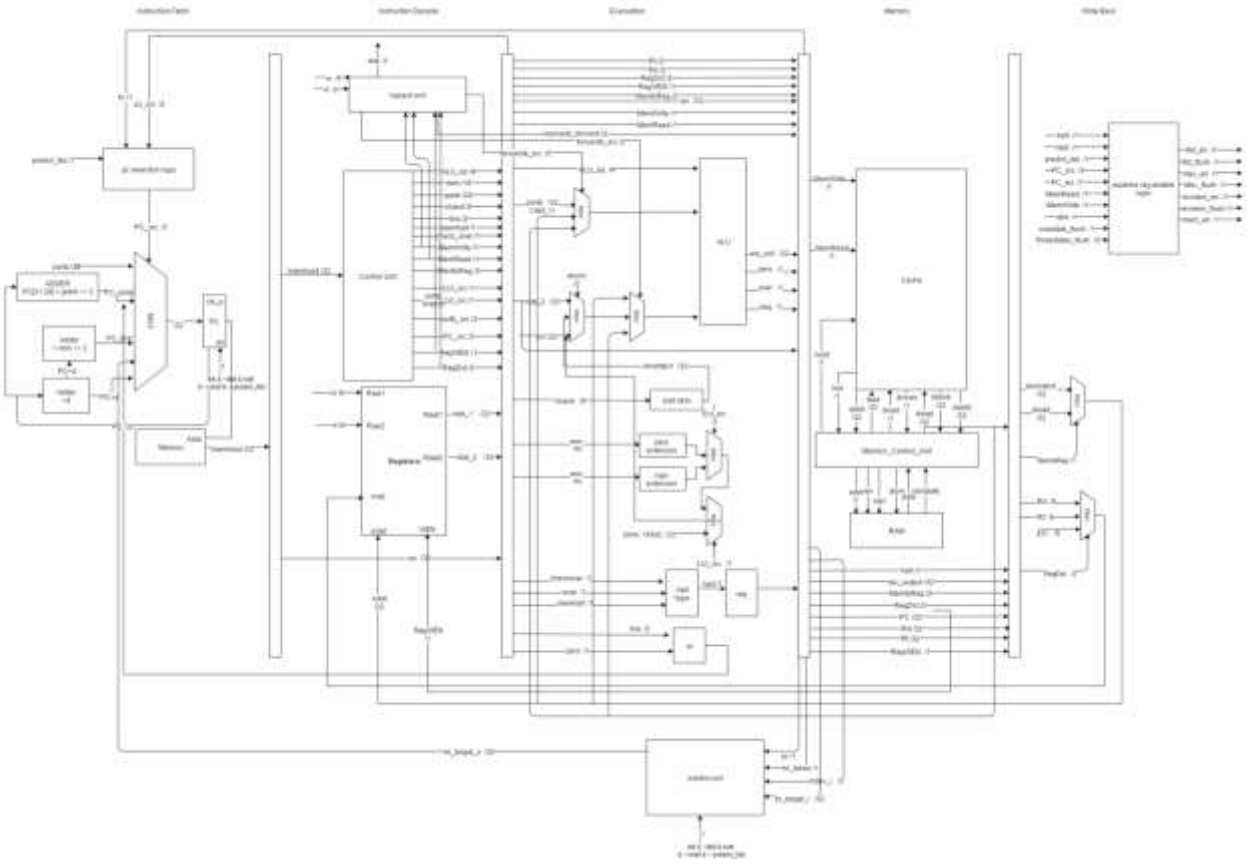

Fig-2 Block Diagram for Pipeline Processor

Results

	Single Cycle Processor	Pipeline Processor
Frequency (max)	25.173 MHz	56.96 MHz
IPC (average)	0.39	0.32
Latency	28.5 ns	87.78 ns
MIPS	9.855	18.048
Critical Path Delay	37.62 ns	8.358 ns
FPGA Total Registers	1285	2003
Total Combinational Functions	2892 / 114480	3477 / 114480

Conclusions

Pipeline processor is designed based on single cycle processor, so the stages for pipeline is the same in single cycle processor, thus the performance for single instruction does not change. Even more, the cycle time and latency for pipeline processor is much longer than single cycle processor, because we need to add stalls for branch, load, immediate used ALU operations, and it will long the execution time for single instruction. Besides that, pipeline processor uses more registers than single cycle processor as space usage, and therefore increases the cost, since we want to run multiple instructions together, and we need to have four groups of pipeline registers.

However, from MIPS and execution time for Critical Path, we can see that pipeline processor performs much better than single cycle processor. It worse to use pipeline processor for complex programs, and real world programs are always complex.

From previous discussion and comparison, we can easily come to the conclusion that single cycle processor is easier to design, but pipeline processor is better on performance.