ECE 437L Midterm Report

Yutao Hu

Hanwen Huang

TA:Younghoon, Yutong

Due: 2017.10.15

1 Executive Overview

This report is to demonstrate the comparison between the single cycle and pipeline processor. The maximum clock frequency of both designs, the length of the critical path(in units of time), the latency of instructions, and the total execution time of the processor using the mergesort.asm program will be collected for the judgment of the processor's performance. The information is collected from the log files after synthesizing. The mergesort.asm program is quite complex since it uses a large number of the MIPS instructions and different dependencies in many situations.

Generally, the single cycle has the advantage of less latency of instructions and simple design because of fewer combination blocks in the design. The pipeline design require much more resouces including pipeline registers and combinational logic to resolve the various hazard, but it has the advantage of higher throughput and less latency between stages than a single cycle since the pipeline overlaps the instructions, which could possibly increase the clock speed. From our result, as the pipeline processor increases the latency and has the shorter critical path, it has a shorter execution time. In conclusion, the pipeline processor has a better performance and consume more resources than the single cycle processor.

2 Processor Design

In Figure 1, all the signals sent to RAM pass through the caches modport in caches interface. There are register file, control unit, request unit, alu, and pc unit in the single-cycle datapath. Simply speaking, PC unit provides the instruction address to caches to fetch the memory. Control unit is responsible for receiving and decoding the instruction for future use. The register file is able to read and write the 32 bits from/to registers. Request unit is in charge of wake up the caches to write/read data from/to specific memory address. ALU unit is to do the calculations following each instruction. All five units work in order for a single instruction after the control unit receives the high ihit signal. In Figure 2, the instruction corresponding signals communicate in the icache modport in caches interface and the data corresponding

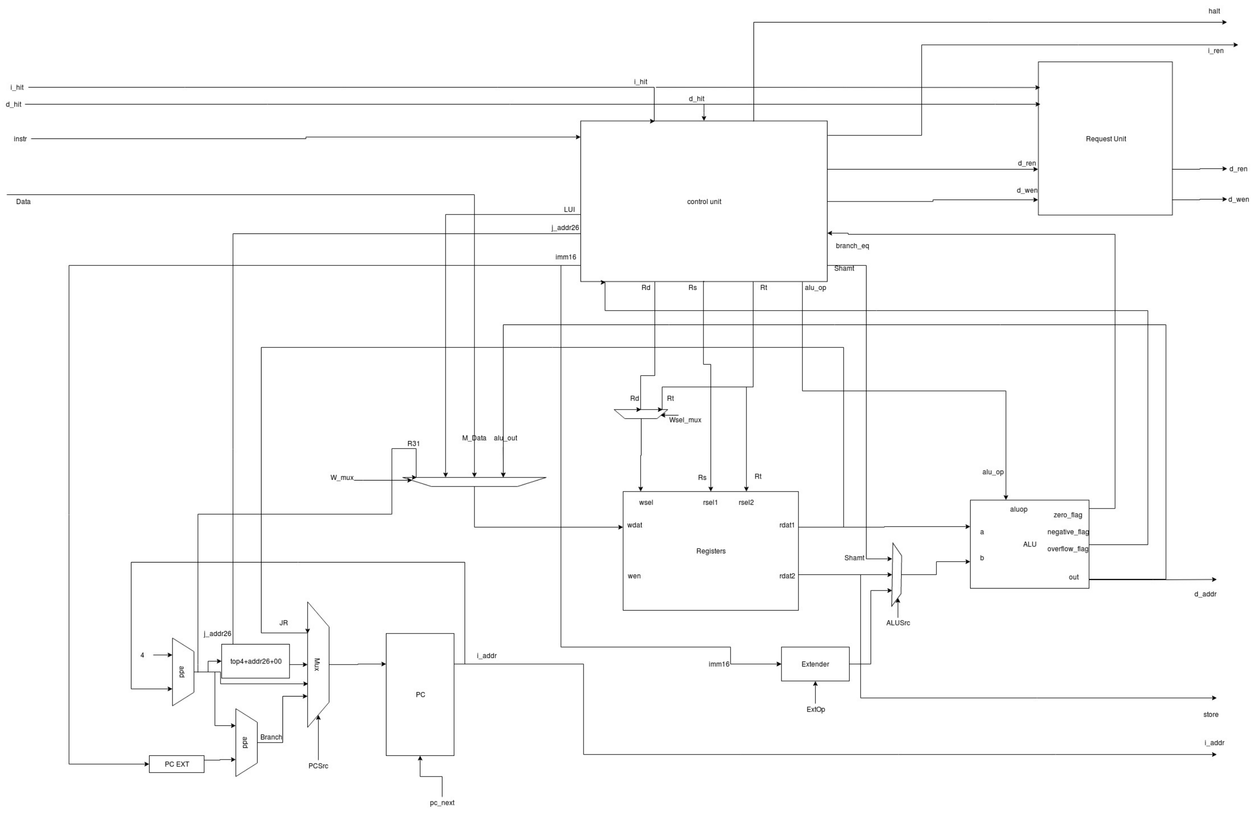

Figure 1: Single Cycle Block Diagram

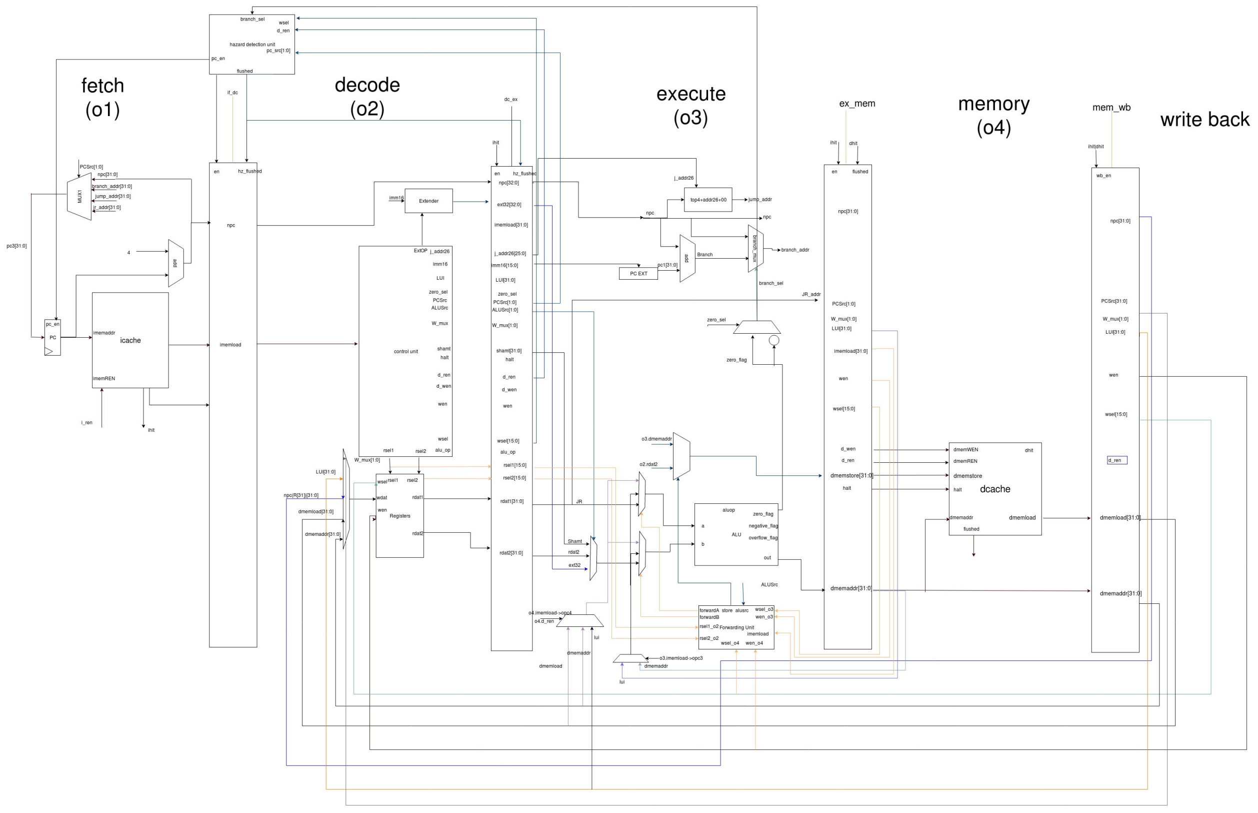

Figure 2: Pipeline Block Diagram

signal communicates in the dcache modport.The original single cycle is divided into 5 stages by latches. Latches save the result from the last stage and pass it to the next stage.

3 Processor Debug

When we simulated the mergesort.asm, we figured out there was part of results different from sim files. After we read through the asm file and understood how it works, we changed the total number of numbers for mergesort. In this way, we decreased the lines in the sim file to about 100 lines. Then we checked the map and found that there is order to plug in the bubble when there are dependencies between the instruction with multiple instructions which we hadn't considered before.

4 Results

	Single Cycle	Pipeline
The maximum clock frequency(MHZ)	34.75	51.92
Period(ns)	28.77	19.26
Length of the critical path(ns)	52.55	33.52
The Latency of of instructions(ns)	28.77	96.30
The total execution time(ns)	1190647	1168143
The instruction number	5399	5399
The Ram Latency	3	3
Average instructions per cycle	0.13	0.089
FPGA Resources		
Total combinational Functions	2,919 / 114,480 (0.03)	3,357 / 114,480 (0.03)
Total registers	1289	1770

Table 1: Processor Specs

From the Table 1, the maximum clock frequency is gained from the system log which shows the restricted Fmax in ”Slow 1200mV 85C Model Fmax Summary” section. The possible maximum clock frequency should be the min(CLK/2, CPUCLK).The period is calculated by 1/Fmax. Length of the critical path is the period - the minimum slack which is also gained from the system log file. The Latency of instructions is calculated by the period times the number of stages. The total execution time is calculated by period times the total cycle which gained from running the ”make system.sim” command. Instruction number is gained from run ”sim” command to simulate the mergesort. Average instruction per cycle is calculated by total instruction number divided by total cycle. FPGA resources required are found in the system FPGA log file in the Fitter Summary section.

5 Conclusion

Based on the result above, the pipeline design has a higher maximum frequency than the single cycle design. The 5 stage of instruction result a higher latency per instruction in pipeline design but the benefit gain from hold 5 instruction at the same time could make up that loss and have a lower total execution time. In terms of the resources, the combinational gates and registers used are higher in pipeline design. registers used in the pipeline is significantly higher because the stage is a huge block of registers to save the 5 instructions running at the same time. besides, there are several extra registers added in the pipeline to make sure it running correctly as a whole. For the combinational logic, the use of hazard unit, forwarding unit and it's corresponding mux consume most of the extra combinational. In conclusion, the same program run faster in pipeline design than single cycle design but pipeline design require more resources than single cycle design.

6 Contributions

Hanwen Huang: 1.Draw the plot 2.Write part of code 3.Debug for the issues along with the hazard unit 4. Solve the synthesizing issue 5.Write part of report

Yutao Hu: 1.Update the plot 2.Write part of code 3.Debug for the dependency issues

4.Solve the hardware issue and make the table 5.Write part of report