Organization of Digital Computer Lab
EECS112L/CSE 132L

Assignment 5
Simple Pipeline MIPS Processor

prepared by: Group_Sperry_Pynes
Student name:
Andrew Sperry
David Pynes

Student ID:
73195172
45557742

EECS Department
Henry Samueli School of Engineering
University of California, Irvine

March, 11, 2016

1 Synopsis

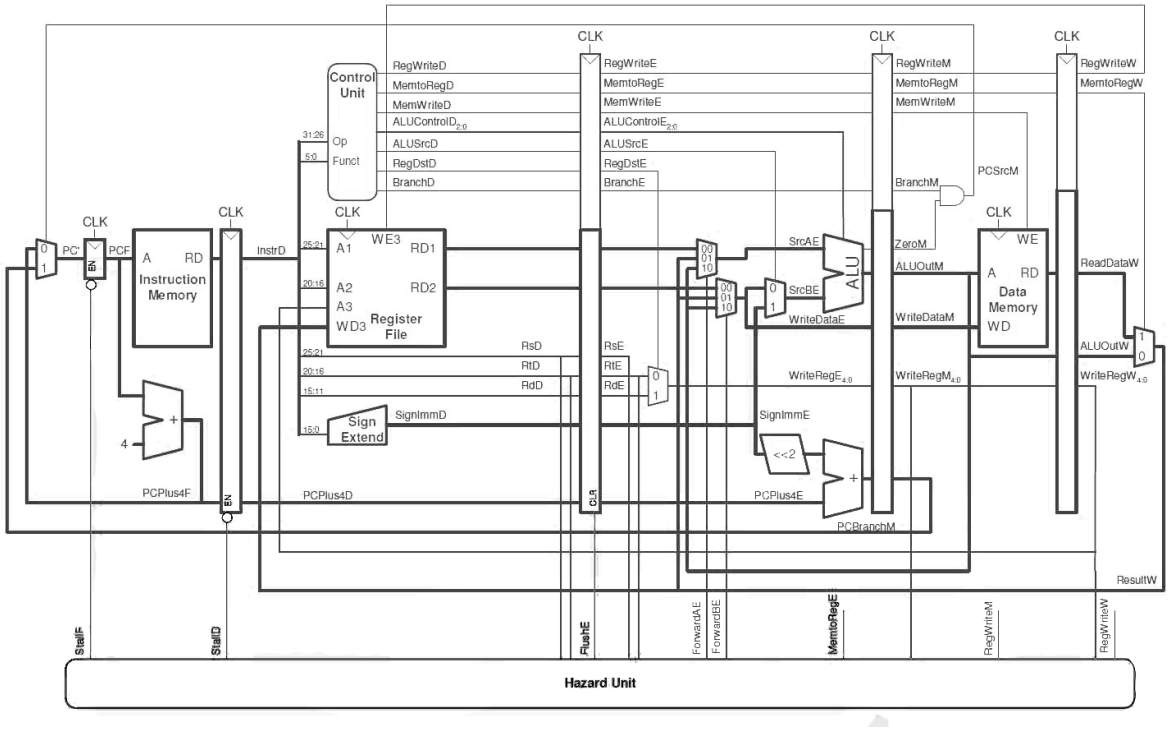
1.1 Description

This project consisted of converting the previously designed single-cycle MIPS processor into a pipe-lined MIPS processor. Hexadecimal format instructions are preloaded from an external file, converted to 32-bit binary, and loaded into the ROM at the start of the simulation. This new processor supports the following instructions: **add, sub, addi, addu, subu, addiu, and, andi, or, ori, nor, xor, xori, andi, ori, xori, sll, srl, sra, slt, sltu, slti, sltiu, beq, bne, bltz, bgez, blez, bgtz, lb, lbu, lh, lhu, lw, sb, sh, and sw**. This project has been coded in VHDL using the ISE Xlinix environment. Verilog was used for component test-benches and Xilinx / Synopsis Toolset were both used for synthesizing and optimization. Within the processor, test ports were added to ensure components were operating correctly at each stage of the datapath design during testing. Surprisingly, only simple syntax errors were discovered, with nothing major having to be fixed while testing.

To support the new design, read-only memory and random-access memory were both converted to a byte addressable design. Byte, half-word and word load logic were placed inside of the random-access memory to optimize the design. This change required an additional type bit be connected to random-access memory but also eliminated the need for the 2-bit load control needed in the previous design.

After synthesis, the single-cycle MIPS processor was compared to the new pipe-lined MIPS processor. The pipe-lined MIPS processor generated noticable improvements, particularly in the maximum combination path delay time and minimum input arrival time before clock. Comparison results are provided at the end of in the report and full synthesis reports are included with the other files submitted.

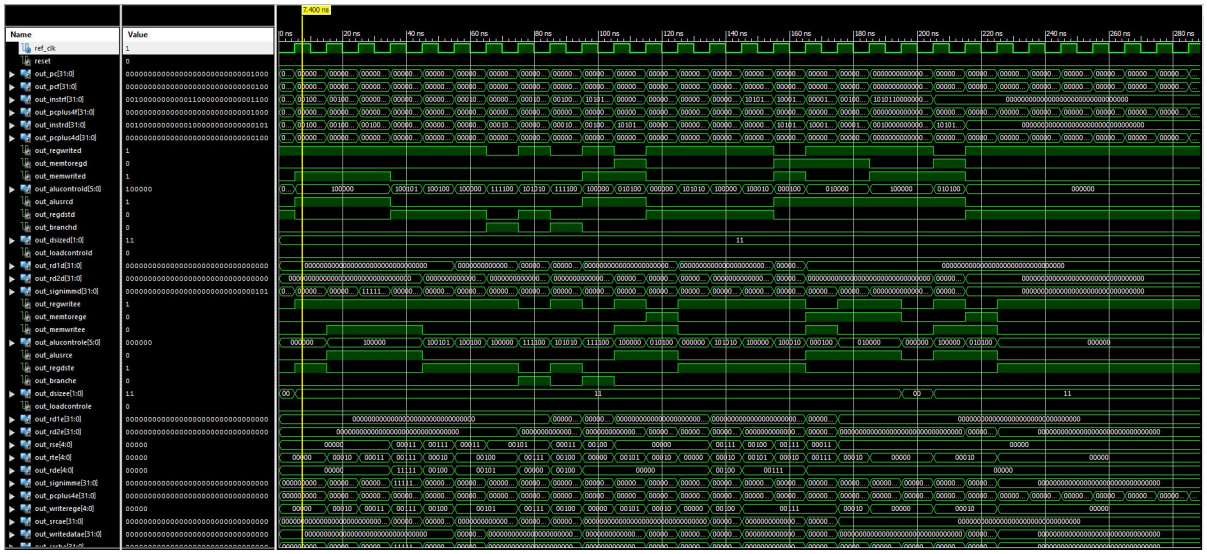
2 Datapath:



3 Simple Test:

The following instruction set was put through the pipe-lined MIPS processor and captured as a waveform shown in the image below. Closer analysis of this waveform showed that the processor did in fact operate as desired by the instruction set.

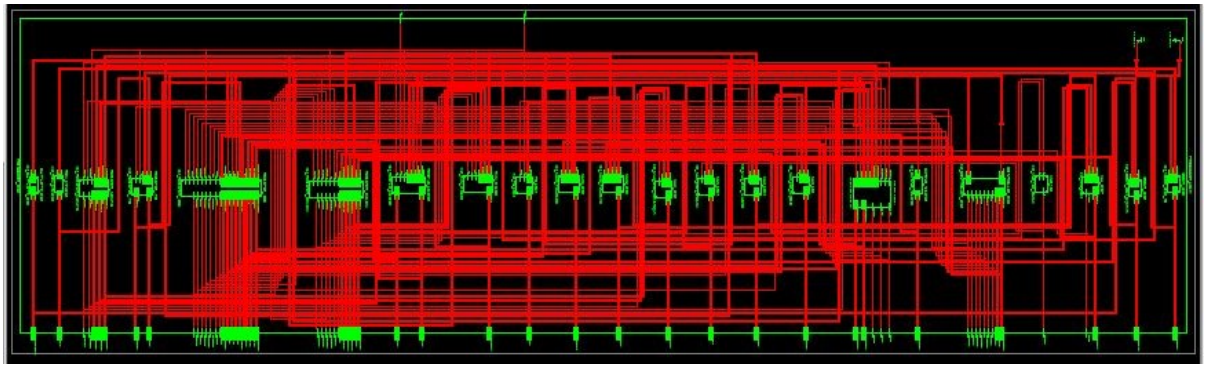
Instruction List: 20020005 / 2003000c / 2067fff7 / 00e22025 / 00642824 / 00a42820 / 10a7000a / 0064202a / 10800001 / 20050000 / 00e2202a / 00853820 / 00e23822 / ac670044 / 8c020050 / 08000011 /20020001 / ac020054



4 Problem Areas

During the development of this pipe-line MIPS processor, no major errors occurred. This was largely due to the fact that most of the components were already build for use in the single-cycle version. The stage dividing register logic was not very complex as well. The biggest task was re-wiring the new circuit design, but with a solid image to work from, everything came together fairly smooth. There were some simple syntax errors that were caught before synthesizing but that was the most that was ran into.

5 Synthesis Comparison



After successful synthesis and optimization of the single-cycle and pipe-lined MIPS processor builds, the following results were obtained for each:

5.1 Single-Cycle MIPS Processor

```
Timing Summary:
---------------
Speed Grade: -4

   Minimum period: 18.567ns (Maximum Frequency: 53.859MHz)
   Minimum input arrival time before clock: 29.747ns
   Maximum output required time after clock: 15.123ns
   Maximum combinational path delay: 12.211ns
```

5.2 Pipe-lined MIPS Processor

```
Timing Summary:
---------------
Speed Grade: -4

   Minimum period: 17.221ns (Maximum Frequency: 58.070MHz)
   Minimum input arrival time before clock: 3.128ns
   Maximum output required time after clock: 12.425ns
   Maximum combinational path delay: 5.051ns
```

The pipe-lined MIPS processor generated a 7.82% increase in maximum frequency, an 851% decrease in minimum input arrival time before clock, a 21.71% decrease in maximum output required time after clock, and a 142% decrease in maximum computational path delay. This also takes into consideration the pipe-lined versions added forwarding and hazard logic component.

6 Conclusion

After converting the single-cycle MIPS processor into a pipe-lined MIPS processor, it is interesting to analyze the results and visualize just how much of a difference dividing the circuit up into smaller uniform stages changes things.