Midterm Report : Single Cycle v. Pipeline

ECE 437 : Computer Design and Prototyping

Katie Bauschka

Apoorv Gaur

Chuan Yean Tan

Friday, March 4, 2016

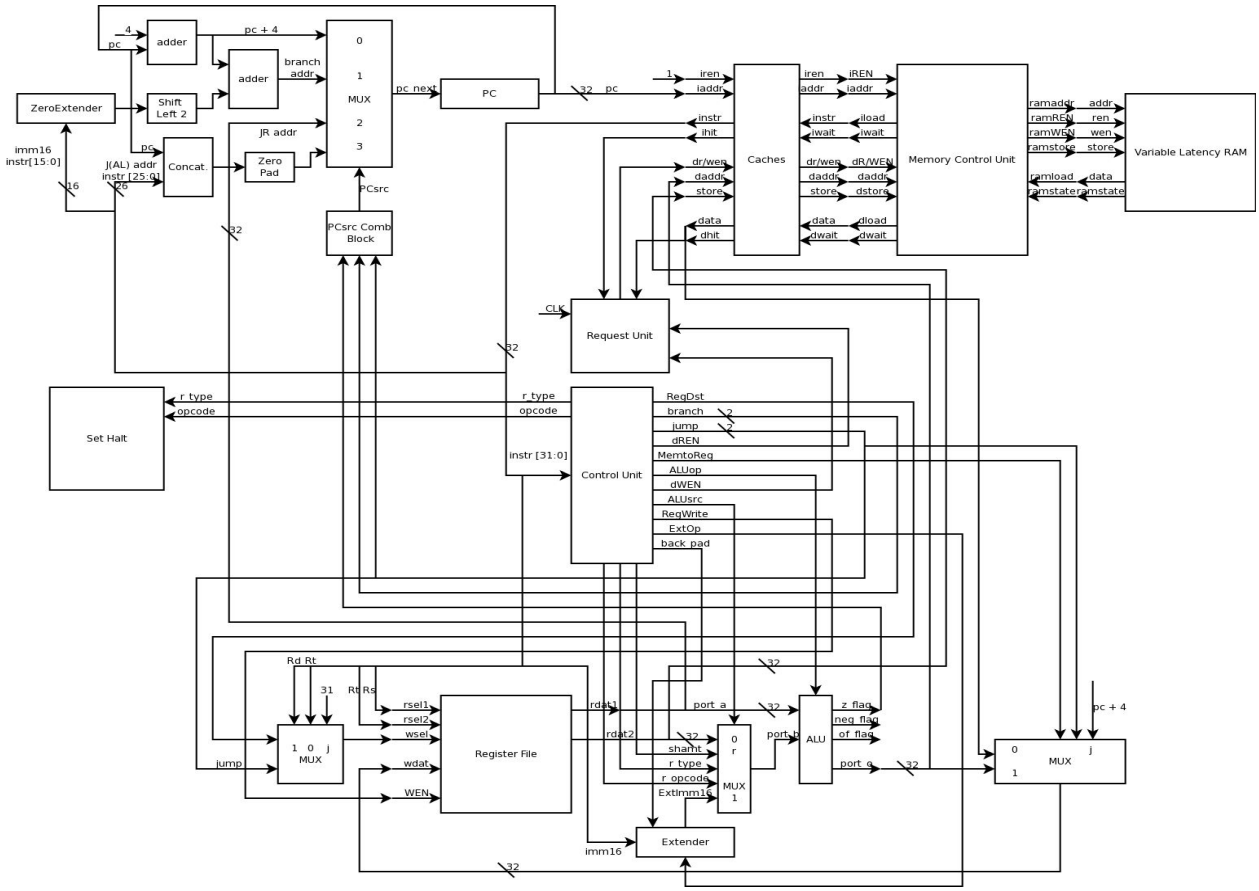
Overview

The throughput of each design is the main feature being compared between the two designs. The biggest benefit of the pipelined processor is that the throughput of any given program will be much higher than that of the single cycle. More instructions can be executed at a time, rather than one instruction at a time. On the other hand, the single cycle design is comparatively simplistic. There is no need for hazard or forwarding units.

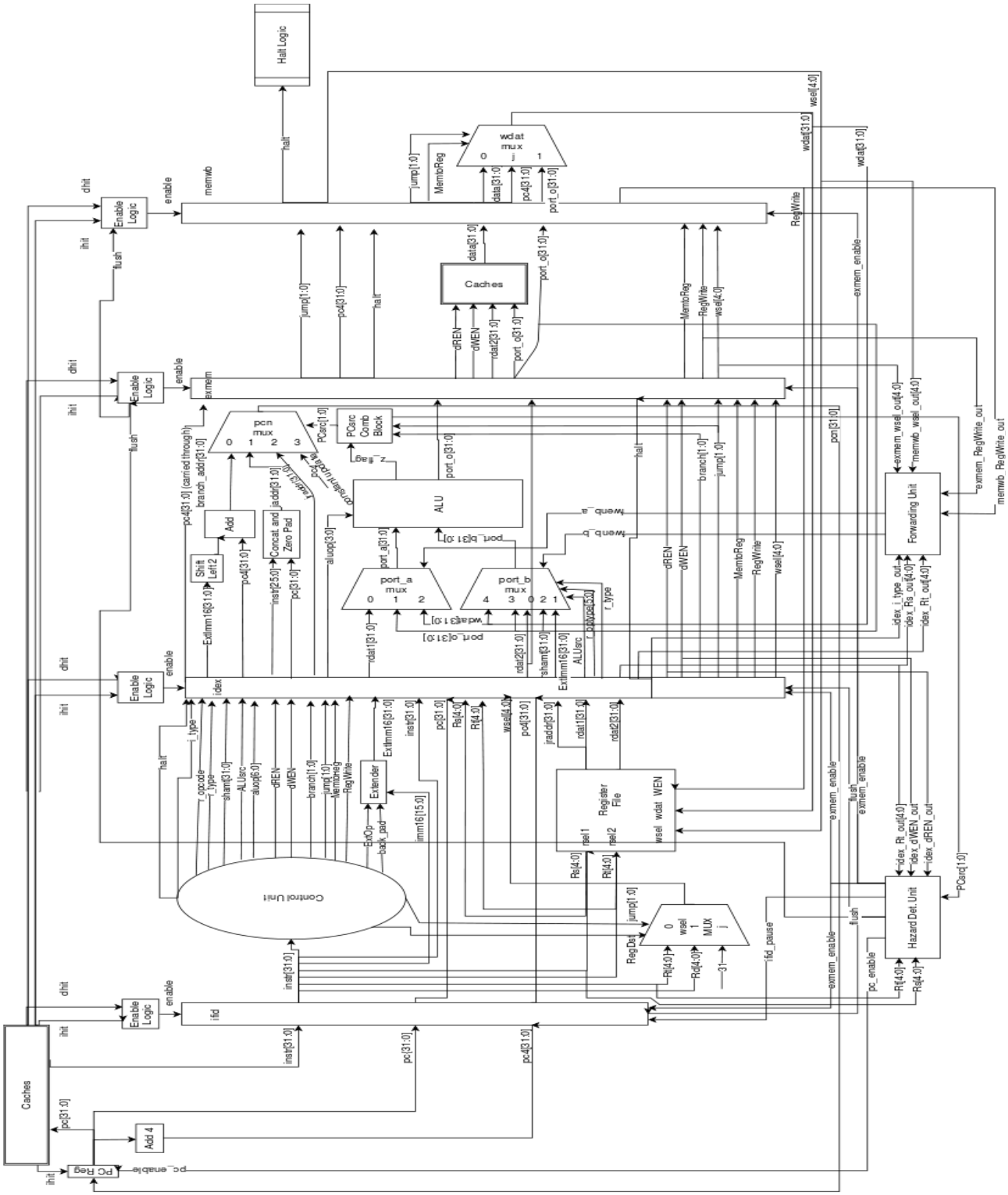
We are using mergesort.asm to test our two processor designs. This program will test all the types of instructions. For the pipeline design, it thoroughly tests the effectiveness of the hazard and forwarding units. At the end of the testing, we expect to see that the pipeline design has a higher throughput than the single cycle design.

Processor Design

Single Cycle



Pipeline


Halt Logic
wdat mux
Caches
ALU
Add
Add 4
Shift Left 2
Concat. and Zero Pad
pcn mux
port_a mux
port_b mux
PCsrc Comb Block
Register File
Control Unit
Extender
Forwarding Unit
Hazard Det. Unit
PC Reg
Enable Logic
memwb
exmem
idex
ifid
Caches

Results

Single Cycle Processor

	Single Cycle Processor
Frequency (maximum possible)	min(36.46, 34.395) = 34.395 MHz (CLK)
Frequency (highest achieved)	1 / (10ns)
Latency (of one instruction)	1 / 34.395MHz = 0.029ms
Performance in MIPS*	0.0135 (13791 cycles)
Length of critical path	27.360ns
FPGA resources	
Total Combinational functions	2781
Total Registers	1279

Pipeline Processor

	Pipeline Processor
Frequency (maximum possible)	min(76.91, 50.77) = 50.77 MHz (CLK)
Frequency (highest achieved)	1/ (5ns)
Latency (of one instruction)	5 * (1/50.77MHz) = 0.0984ms
Performance in MIPS*	0.00271 (20221 cycles)
Length of critical path	19.615ns
FPGA Resources	
Total Combinational functions	3384
Total Registers	1725

*Formula Used:

$$MIPS = \frac{n \cdot 10^{-6}}{latency \cdot cycles}$$

where n is total number of instructions used by mergesort (5399).

Conclusions

As seen from the tables above, a pipelined processor can operate at a higher frequency because fewer instructions need to be executed to per clock cycle.

The latency, however, increases for a pipelined processor as compared to a single cycled processor because while trying to execute several instructions at once, we introduce stalls and flushes to avoid overlaps, forwarding and hazarding issues. This is expressed in performance in MIPS, too, as an increase in latency leads to fewer number of instructions per cycle.

The length of the critical path decreases, which is explained by the increase in number of registers that are used in a pipelined processor. This is also depicted by the increase in number of registers used for a synthesized simulation (FPGA Resources) of the pipeline processor as compared to a single cycle processor.

Contributions

Single Cycle Processor

- Block Diagram - Katie
- Final Design - Katie

Pipeline Processor

- Block Diagram - Katie
- Final Design
    - Hazard Unit - Katie and Apoorv
    - Forwarding Unit - Katie and Apoorv
    - Datapath - Apoorv and Katie

Midterm Report Result Tables - Apoorv