ECE 4750 Computer Architecture

Section 6: Problem-Based Learning on Processors

http://www.csl.cornell.edu/courses/ece4750
School of Electrical and Computer Engineering
Cornell University

revision: 2022-09-29-23-30

List of Problems

Problem 1. Single-Cycle vs. Iterative Multipliers in a TinyRV1 Processor

In this problem, we will consider a pipelined TinyRV1 processor that uses stalling to resolve data hazards. The baseline design includes a single-cycle integer multiplier and the alternative design includes a 4-cycle unpipelined iterative integer multiplier.

Part 1.A Multiplication Microbenchmark

We will evaluate the performance of both the baseline and alternative design by considering the following assembly sequence:

```
loop:
 lw   x5,  0(x11)
 mul  x6,  x5, x12
 sw   x6,  0(x11)
 addi x11, x11, 4
 addi x13, x13, -1
 bne  x13, x0, loop
```

Write a C function that clearly represents the functionality of this assembly sequence. Recall that arguments are passed in registers `x11`–`x17`, the return value is stored to `x10`, the return address is stored in `x1`, and `x5`–`x7`, `x28`–`x31` are used as temporary registers.

Part 1.B TinyRV1 Processor with Single-Cycle Integer Multiplier

The baseline design uses a single-cycle integer multiplier. Assume that this processor has a cycle time of 1ns, and that the critical path is through the single-cycle multiplier in the X stage. **Draw a pipeline diagram illustrating the first iteration of the loop.** Draw arrows in your pipeline diagram to indicate any data or control hazards. **Fill in the appropriate row in the table in Part 1.D.** Assume that `x13` is initially 64.

`lw x5, 0(x11)`																									
`mul x6, x5, x12`																									
`sw x6, 0(x11)`																									
`addi x11, x11, 4`																									
`addi x13, x13, -1`																									
`bne x13, x0, loop`																									

Part 1.C TinyRV1 Processor with 4-Cycle Unpipelined Iterative Integer Multiplier

The alternative design uses a 4-cycle unpipelined iterative integer multiplier. Assume that multiplier is no longer on the critical path, and the cycle time for the processor decreases to 0.75ns. **Draw a pipeline diagram illustrating the first iteration of the loop.** Draw arrows in your pipeline diagram to indicate any data or control hazards. **Fill in the appropriate row in the table in Part 1.D.** Assume that `x13` is initially 64.

`lw x5, 0(x11)`																									
`mul x6, x5, x12`																									
`sw x6, 0(x11)`																									
`addi x11, x11, 4`																									
`addi x13, x13, -1`																									
`bne x13, x0, loop`																									

Part 1.D Comparison of Processor Microarchitectures

Consider the results in the following table. Which microarchitecture has the highest performance on this microbenchmark? How would these results generalize to other workloads? Discuss some of the trade-offs in terms of area and energy between the processor microarchitectures. Consider what would happen if we used a 4-cycle *pipelined* integer multiplier.

Microarchitecture	**Inst/Prog**	**Cycles/Inst** (ns)	**Time/Cycle**	**Exec Time** (ns)
Processor w/ 1-cycle Mul			1.0ns	
Processor w/ 4-cycle Mul			0.7ns	