Michael Lankford Homework 7 4/16/2023

1. Exercise 4.7
   1. Exercise 4.7.2

The latency of lw is calculated by adding the latency of the Register Read, I-Mem, Register File, Mux, ALU, I-Mem, Mux, and Register Setup. This comes out to 30+250+150+25+200+250+25+20 which equals 950ps.

* 1. Exercise 4.7.3

The latency of sw is calculated by adding the latency of the Register Read, I-Mem, Register File, ALU, I-Mem, and Mux. This comes out to 30+250+150+200+250+25 which equals 905ps.

* 1. Exercise 4.7.5

The latency of an arithmetic, logical, or shift I-type (non-load) instruction is calculated by adding the latency of the Register Read, I-Mem, Register File, Mux, ALU, Mux, and Register Setup. This comes out to 30+250+150+25+200+25+20 which equals 700ps.

* 1. Exercise 4.7.6

The minimum clock period of this CPU is 950ps because clock period is found from the longest delay in the CPU.

1. Exercise 4.9
   1. Exercise 4.9.1

The clock cycle time without the improvement is 950ps, as seen from our previous values in 4.7.

The clock cycle time with the improvement is calculated by taking the original 950ps and adding 300ps to it, which comes out to 1250ps.

* 1. Exercise 4.9.2

The speedup achieved by this addition is calculated by taking the total 1250ps and multiplying it by 95% (for the 5% reduction in number of instructions). The comes out to 1187.5ps. Compared to the original run time, 950/1187.5 results in a speedup of .8, which shows that the new addition is slower than the original.

1. Exercise 4.16
   1. Exercise 4.16.1

The clock cycle time in a pipelined process depends on the stage that takes the longest, so in this case it would be 350ps.

The clock cycle time in a non-pipelined process is calculated by adding the latency of IF, ID, EX, MEM, and WB. This comes out to 250+350+150+300+200 which equals 1250ps.

* 1. Exercise 4.16.3

If we could split one stage of the pipelined data path into two new stages, each with half the latency of the original stage, I would split the ID stage. This would cause the new clock cycle time to be 300ps.

* 1. Exercise 4.16.5

The utilization of the write-register port of the “Registers” unit comes out to 65%. This is calculated by adding the ALU/Logic and Load breakdowns, which is the data path that is used for the write-register.

1. Exercise 4.18

Addi $s0, $s1, 5 $s0 = 27

Add $s2, $s0, $s1 $s2 = 49

Addi $s3, $s0, 15 $s3 = 42

The final value of register $s2 would be 49 and the final value of register $s3 would be 42.

1. Exercise 4.20

Addi $s0, $s1, 5 NOP

NOP

Add $s2, $s0, $s1 Add $s3, $s0, 15 NOP

Add $s4, $s2, $s1

1. Exercise 4.28
   1. Exercise 4.28.1

The extra CPI due to mispredicted branches with the always-taken predictor is calculated by taking the percentage that the predictor is correct and multiplying it by the percentage of the branch instructions and the amount of stall cycles that will be produced, in this case 3. So, the extra CPI will be (1-.45)(.25)(3) which comes out to

.4125.

* 1. Exercise 4.28.3

The extra CPI due to mispredicted branches for the 2-bit predictor is calculated by taking the percentage that the predictor is correct and multiplying it by the percentage of the branch instructions and the amount of stall cycles that will be produced. So, the extra CPI will be (1-.85)(.25)(3) which comes out to .1125.

* 1. Exercise 4.28.5

To calculate speedup, you first have to calculate the increase in the CPI with the new conditions and with the old conditions. For the old conditions, take the 2-bit predictor when its correct and multiply it by the branch instructions , then add 1 to it for the total. This equation can be seen: 1+(.15(.25)) = 1.0375. For the new conditions, repeat but factor in the new conditions. The equation can be seen: 1+(.15(.111)) = 1.01665. For

speedup divide the new by the old with the instruction increase, 1.0375/(1.125\*1.01665) results in .91 speedup.

1. Exercise 4.29
   1. Exercise 4.29.1

The accuracy of always-taken predictors for this sequence is 3/5 or 60%. The accuracy of always-not-taken predictors for this sequence is 2/5 or 40%.

* 1. Exercise 4.29.2

If starting at predict not taken, the accuracy of the predictor will be 25% due to it predicting NOT TAKEN when in the 2cd step of the pattern (T, NT, T, T) but not successfully predicting for any of the other predictions.

* 1. Exercise 4.29.3

The accuracy of the predictor if this pattern was repeated forever would be 60% due to each time the pattern runs through the graph, there are 3 successful predictions out of 5 or 60% success.