Michael Smith

Feburary 20th, 2018

Chapter 4 Homework

CSE 401

4.8)

1) In a pipeline processor all stages take a single clock tick to complete. So the clock cycle is set by the slowest stage of the operation. Making the time for the slowest stage as the cycle time, so the time for this example is 350 ps.

In a non-pipeline processor, the cycle time becomes the time for all the stages to complete. So the sum for this example is 250 + 350 + 150 + 300 + 200 = 1250ps.

2) Assuming the LW (load word) uses all 5 stages. In pipeline then the LW takes 5 cycles. Therefore, the time would be 350 \* 5 = 1750ps.

While the non-pipeline processor would just be the sum of all 5 stages again, so the time would become 1250ps, which was gathered from problem 1.

3) You would have to split the stages with the longest latency to increase performance, so splitting the ID stage would make it 175 each part. Now the clock cycle would become the second longest stage which would be memory, 300ps.

4) The utilization of the memory stage is taken into consideration during usage of load word and store word. Those stages have a 20% and 15% utilization respectfully. So the utilization becomes a 35% of the clock cycles.

5) The write register port may be used during the ALU and LW instructions. Each of those stages have a 45% and 20% respectfully. Making a total of 65% clock cycles.

6) Multi Cycle Execution: (5 x 20%) + (4 x ( 45% + 20% + 15%) = 4.2. While the single cycle execution: 1250ps / 350 ps = 3.5. Meaning in this case the multi cycle speed up is greater then the single cycle.

4.12)

1) The first instruction will have a two stall cycle due to dependencies of first and second instruction. So, the dependency will be on stall cycle with second as instruction. The equation would look like 1 + (0.35 \* 2) + (0.05 \* 1) + (0.1 \* 1) = 1.95. Therefore, the stall is 0.95 / 1.95 = 48.7%

2) WIth the full forwarding, only stalles can occur on the Read and Write stages. There will be one cycle stalled with the memory stage of one instruction to next instruction. So the equation would be = 1 + 0.2 + 0.1 + 0.1 = 1.4. So the stall percentage is 0.4 / 1.4 = 28.5%

3) The firs execution does not have stalls, but from the 2nd instruction, memory to first would have one stall. So the EX and Mem would be 0.05 + 0.05 + 0.20 = 0.3 and the mem and wb would be 0.05 + 0.10 + 0.20 = 0.35.

4) With the results from above. Speed up without the forwarding is = 1.95 x 130 ps = 253.5 ps. With forwarding = 1.20 \* 150ps = 180ps. Total would become = 1 ( 253.5 - 180/180) = 1.4ps.

5) Time with forwarding = 1.2 \* 150 = 180ps. While the time traveling would be = 1 \* 250 = 250ps.

6) The time per instruction from 4.12.3 with shorter time would be 1.55 \* 140 = 217 ps. And the memory and wb would be 1.45 \* 130 = 188.5ps.

4.15)

1) Each branch that is not correctly predicted will cause a 3 stall cycle, making the extra CPI: 3 x (1-0.4) x 0.15 = 0.27 and 3 x ( 1 – 0.60 ) x 0.10 = 0.12.

2) Repeating but with the always not taken predictor: 3 x (1 – 0.60) x 0.15 = 0.18 and 3 x (1-0.4) x 0.10 = 0.18.

3)Repeating again but with the 2 bit predictor: 3 x (1-0.8) x 0.15 = 0.009 and 3 x (1-0.95) x 0.1 = 0.015

4) If the branches are predicted correctly the CPI becomes 1 and they become ALU instructions with CPI of 1 as well. Incorrect predictions are converted and become ALU instructions with CPI of 1. Therefore we have: (Without conversion) 1 + 3 x (1-0.8) x 0.15 = 1.090 and 1 + 3 × (1 – 0.95) × 0.10 =1.015. (With conversion) 1 + 3 × (1 – 0.80) × 0.15 × 0.5 =1.045 and 1 + 3 × (1 – 0.95) × 0.10 × 0.5 =1.008. The speed up would be 1.090/1.045 = 1.043 and 1.015/1.008 = 1.007

5) Similarly to question 4, but replacing the instruction with 2 instructions for the alu would become 1 + (1\*(0.5\*0.25)) + ((1-0.85)\*(0.5\*0.25)\*2) = 1.1625. Which creates speed up of 1.075 / 1.1625 = 0.9247.

6) To create the average of 85% we would have to use the formula: 0.85 = (0.8 x 1) + (0.2 \* x) where x is the accuracy. Solving for x, you would receive x = 0.25 or 25% accuracy.