Lab Assignment 1: Performance Measurements Report

Jianwei Sun (1000009821), Yi Fan Zhang (1000029284)

In this lab, sim-safe.c is modified to count cycle stalls due to RAW hazards on two different pipelined CPUs, and output their respective % performances drops vs the idea CPI of 1.0. A microbenchmark, "mbq1.c" containing all cases of 1-Cycle and 2-Cycle delays is written for verification.

CPI Slowdown:

The CPI slowdown is calculated using the following equation:

CPI Slowdown = [(number_of_1_cycle_stall_hazards) + 2*(number_of_2_cycle_stall_hazards)] / (total_number_of_instructions)

Slowdowns are caused by hazard stall cycles, thus the total number of extra cycles generated by hazards, proportional to the total number of instructions gives the slowdown ratio. In the case where there are zero hazards, the slowdown equation evaluates to zero, as expected.

Microbenchmark and Results

Appendix A and B details the pipelined cycles for each case of data hazards. **Case A** shows common hazards where a computation instruction (such as addu) followed a load instruction. Because there are two stages between Decode and Writeback, q1 must stall for two cycles to for data to arrive at the next Decode stage from the previous Writeback. While q2 has foward/bypass, load is computed at the Memory stage; thus, with 2 Execute stages, there must be 2 cycle delays. **Case E** shows two computations. For q1, this case is the same as above. For q2, as results from computation instructions is finished after X2, with forwarding to the next X1, only 1 cycle stall is needed. **Case B** shows a corner case when the 3rd instruction has a dependency with the first. However, because the 2nd instruction is has a data hazard with the first, and must stall as a result; this eliminates the data hazard of the 3rd instructions. **Case C and D** are variations of the case where one independent instruction is inserted between two dependencies. This decreases the number of stall cycles needed by 1 as the independent instruction takes 1 cycle. **Case F** involves a store. As q2 contains forwarding and bypassing, data can be forwarded from one Memory stage to the next, thus eliminating the need for stall.

The micro benchmark contains a total of 49 instructions within the main loop

There is a total of 2 1-Cycle slowdowns and 14 2-Cycle slowdowns for the q1 pipelined CPU, which has 5 stages and no forwarding or bypassing. Using the CPI slowdown equation, this gives a calculated value of 61% slowdown. From sim-safe, the simulated slowdown is 61.22%.

There is a total of 8 1-Cycle slowdowns, and 5 2-Cycle slowdowns for the q2 pipelined CPU, which has an extra EX stage and full forwarding and bypassing. Using the CPI slowdown equation, this gives a calculated value of 37% slowdown. From sim-safe, the simulated slowdown is 36.73%.

The simulated slowdown matches the predicted slowdown based on the CPI Slowdown equation. The compiler uses many initialization instructions, which will change the total number of instructions and hazards. However, because the microbenchmark was looped 1,000,000 times, these extra instructions outside the loop can be considered as "noise", and therefore, negligible.

APPENDIX A: PIPELINE DIAGRAMS FOR Q1

Case	Instruction	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	Comments
A	Lw $2 16($fP)	F	D	X	M	W								
	Slt $9 $3 $2		F	*d	*d	D	X	M	W					2 Cy Stall
B	Lw $3 0($5)	F	D	X	M	W								
	Addu $4 $3 1		F	*d	*d	D	X	M	W					2 Cy Stall
	Addu $3 $3 1					F	D	X	M	W				No Stall
C	Lw $3 0($5)	F	D	X	M	W								
	Addu $4 $4 1		F	D	X	M	W							No dependency
	Addu $3 $3 1			F	*d	D	X	M	W					1 Cy Stall
D	Addu $3 $3 1	F	D	X	M	W								
	Addu $4 $4 1		F	D	X	M	W							No dependency
	Addu $3 $3 1			F	*d	D	M	W						1 Cy Stall
E	Lw $8 24($fp)	F	D	X	M	W								
	Addu $2 $8 1		F	*d	*d	D	X	M	W					2 Cy Stall
	Move $8 $2					F	*d	*d	D	X	M	W		2 Cy Stall
F	Move $8 $2	F	D	X	M	W								
	Sw $8 28($fp)		F	*d	*d	D	X	M	W					2 Cy Stall

APPENDIX B: PIPELINE DIAGRAMS FOR Q2

Case	Instruction	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	Comments
A	Lw $2 16($fP)	F	D	X1	X2	M	W							
	Slt $9 $3 $2		F	*d	D	X1	M	W						1 Cy Stall
B	Lw $3 0($5)	F	D	X1	X2	M	W							
	Addu $4 $3 1		F	*d	*d	D	X1	X2	M	W				2 Cy Stall
	Addu $3 $3 1					F	D	X1	X2	M	W			No stall
C	Lw $3 0($5)	F	D	X1	X2	M	W							
	Addu $4 $4 1		F	D	X1	X2	M	W						No dependency
	Addu $3 $3 1			F	*d	D	X1	X2	M	W				1 Cy Stall
D	Addu $3 $3 1	F	D	X1	X2	M	W							
	Addu $4 $4 1		F	D	X1	X2	M	W						No dependency
	Addu $3 $3 1			F	D	X1	X2	M	W					No stall
E	Lw $8 24($fp)	F	D	X1	X2	M	W							
	Addu $2 $8 1		F	*d	*d	D	X1	X2	M	W				2 Cy Stall
	Move $8 $2					F	*d	D	X1	X2	M	W		1 Cy Stall
F	Move $8 $2	F	D	X	M	W								
	Sw $8 28($fp)		F	D	X	M	W							No stall