HW4

Zhijiia Chen

November 5, 2018

C.1 Use the following code fragment:

```
Loop:   ld x1,0(x2)       ;load x1 from address 0+x2
        addi x1,x1,1      ;x1=x1+1
        sd x1,0,(x2)      ;store x1 at address 0+x2
        addi x2,x2,4      ;x2=x2+4
        sub x4,x3,x2      ;x4=x3-x2
        bnez x4,Loop      ;branch to Loop if x4 != 0
```

Assume that the initial value of x3 is x2+396.

a. Data hazards are caused by data dependencies in the code. Whether a dependency causes a hazard depends on the machine implementation (i.e., number of pipeline stages). List all of the data dependencies in the code above. Record the register, source instruction, and destination instruction; for example, there is a data dependency for register x1 from the ld to the addi.

answer:

x1 ld addi
x1 addi sd
x2 ld addi
x2 sd addi
x2 sub addi
x4 bnez sub

b. Show the timing of this instruction sequence for the 5-stage RISC pipeline without any forwarding or bypassing hardware but assuming that a register read and a write in the same clock cycle "forwards" through the register file, as between the add and or shown in Figure C.5. Use a pipeline timing chart like that in Figure C.8. Assume that the branch is handled by flushing the pipeline. If all memory references take 1 cycle, how many cycles does this loop take to execute?

answer:

Number of iteration $= \frac{396}{4} = 99$. Loop 1 to loop 98 takes 16 cycles and the last loop takes 18 cycles, so this loop takes $98 \times 16 + 18 = 1586$ cycles.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ld x1,0(x2)	IF	ID	EX	MEM	WB												
addi x1,x1,1		IF	Stall	Stall	ID	EX	MEM	WB									
sd x1,0,(x2)					IF	Stall	Stall	ID	EX	MEM	WB						
addi x2,x2,4								IF	ID	EX	MEM	WB					
sub x4,x3,x2									IF	Stall	Stall	ID	EX	MEM	WB		
bnez x4,Loop												IF	ID	EX	MEM	WB	
ld x1,0(x2)													IF	ID	EX	MEM	WB

c. Show the timing of this instruction sequence for the 5-stage RISC pipeline with full forwarding and bypassing hardware. Use a pipeline timing chart like that shown in Figure C.8. Assume that the branch is handled by predicting it as not taken. If all memory references take 1 cycle, how many cycles does this loop take to execute?
answer:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
ld x1,0(x2)	IF	ID	EX	MEM	WB									
addi x1,x1,1		IF	ID	Stall	EX	MEM	WB							
sd x1,0,(x2)			IF	Stall	ID	EX	MEM	WB						
addi x2,x2,4					IF	ID	EX	MEM	WB					
sub x4,x3,x2						IF	ID	EX	MEM	WB				
bnez x4,Loop							IF	Stall	ID	EX	MEM	WB		
(instruction after the loop)									IF	Flush	Flush	Flush	Flush	
ld x1,0(x2)										IF	ID	EX	MEM	WB

Number of iteration = $\frac{396}{4} = 99$. Loop 1 to loop 98 takes 9 cycles and the last loop takes 12 cycles, so this loop takes $98 \times 9 + 12 = 894$ cycles.

d. Show the timing of this instruction sequence for the 5-stage RISC pipeline with full forwarding and bypassing hardware, as shown in Figure C.6. Use a pipeline timing chart like that shown in Figure C.8. Assume that the branch is handled by predicting it as taken. If all memory references take 1 cycle, how many cycles does this loop take to execute?
answer:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
ld x1,0(x2)	IF	ID	EX	MEM	WB									
addi x1,x1,1		IF	ID	Stall	EX	MEM	WB							
sd x1,0,(x2)			IF	Stall	ID	EX	MEM	WB						
addi x2,x2,4					IF	ID	EX	MEM	WB					
sub x4,x3,x2						IF	ID	EX	MEM	WB				
bnez x4,Loop							IF	Stall	ID	EX	MEM	WB		
ld x1,0(x2)									IF	ID	EX	MEM	WB	

Number of iteration = $\frac{396}{4} = 99$. Loop 1 to loop 98 takes 8 cycles and the

last loop takes 12 cycles, so this loop takes $98 \times 8 + 12 = 796$ cycles.

e. High-performance processors have very deep pipelines—more than 15 stages. Imagine that you have a 10-stage pipeline in which every stage of the 5-stage pipeline has been split in two. The only catch is that, for data forwarding, data are forwarded from the end of a pair of stages to the beginning of the two stages where they are needed. For example, data are forwarded from the output of the second execute stage to the input of the first execute stage, still causing a 1-cycle delay. Show the timing of this instruction sequence for the 10-stage RISC pipeline with full forwarding and bypassing hardware. Use a pipeline timing chart like that shown in Figure C.8 (but with stages labeled IF1, IF2, ID1, etc.). Assume that the branch is handled by predicting it as taken. If all memory references take 1 cycle, how many cycles does this loop take to execute?
answer:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
ld x1,0(x2)	IF1	IF2	ID1	ID2	EX1	EX2	MEM1	MEM2	WB1	WB2													
addi x1,x1,1		IF1	IF2	ID1	ID2	Stall	Stall	Stall	EX1	EX2	MEM1	MEM2	WB1	WB2									
sd x1,0,(x2)			IF1	IF2	ID1	Stall	Stall	Stall	ID2	EX1	EX2	MEM1	MEM2	WB1	WB2								
addi x2,x2,4				IF1	IF2	Stall	Stall	Stall	ID1	ID2	EX1	EX2	MEM1	MEM2	WB1	WB2							
sub x4,x3,x2					IF1	Stall	Stall	Stall	IF2	ID1	ID2	Stall	EX1	EX2	MEM1	MEM2	WB1	WB2					
bnez x4,Loop									IF1	IF2	Stall	Stall	Stall	Stall	ID1	ID2	EX1	EX2	MEM1	MEM2	WB1	WB2	
ld x1,0(x2)										IF1	Stall	Stall	Stall	Stall	IF2	ID1	ID2	EX1	EX2	MEM1	MEM2	WB1	WB2

Number of iteration = $\frac{396}{4} = 99$. Loop 1 to loop 98 takes 9 cycles and the last loop takes 22 cycles, so this loop takes $98 \times 9 + 22 = 904$ cycles.

f. Assume that in the 5-stage pipeline, the longest stage requires 0.8 ns, and the pipeline register delay is 0.1 ns. What is the clock cycle time of the 5-stage pipeline? If the 10-stage pipeline splits all stages in half, what is the cycle time of the 10-stage machine?
answer:

Cycle time for 5-stage pipeline = 0.8+0.1=0.9 ns.
Cycle time for 10-stage pipeline = $\frac{0.8}{2} + 0.1 = 0.5$ ns.

g. Using your answers from parts (d) and (e),determine the cycles per instruction (CPI) for the loop on a 5-stage pipeline and a 10-stage pipeline. Make sure you count only from when the first instruction reaches the write-back stage to the end. Do not count the start-up of the first instruction. Using the clock cycle time calculated in part (f), calculate the average instruction execute time for each machine.
answer:

CPI of 5-stage pipeline: $\frac{796}{99 \times 6} = 1.34$.
CPI of 10-stage pipeline: $\frac{904}{99 \times 6} = 1.52$.
Average instruction execution time of 5-stage: $1.34 * 0.92 = 1.21$ ns.
Average instruction execution time of 10-stage: $1.52 * 0.92 = 0.76$ ns.

C.2. Suppose the branch frequencies (as percentages of all instructions) are as follows:

Conditional branches	15%
Jumps and calls	1%
Taken conditional branches	60% are taken

a. We are examining a four-stage pipeline where the branch is resolved at the end of the second cycle for unconditional branches and at the end of the third cycle for conditional branches. Assuming that only the first pipe stage can always be completed independent of whether the branch is taken and ignoring other pipeline stalls, how much faster would the machine be without any branch hazards?

answer:

Without branch hazard, the speed up is $\frac{4}{1+0} = 4$.

Suppose the four pipeline stages are Instruction Fetch (IF), Instruction Decode (ID), Excecution (EX) and Write Back (WB). We need to calculate the CPI for the case of jumps and calls, taken conditional branches and untaken conditional branches respectively.

The following figure show how jumps and calls are handled. The IF in the second cycle of instruction i+2 is because the branch is resolved at the end of second cycle for unconditional branches and we realize that the instruction fetched in the second cycle is incorrect. We can see that jumps and calls instructions introduce one stall in the pipeline.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Jumps and calls	IF	ID	EX	WB				
i+1		IF	IF	ID	EX	WB		
i+2			Stall	IF	ID	EX	WB	
i+3					IF	ID	EX	WB

The following figure show how taken conditional branches are handled. The instruction i+1 is installed for once cycle because the branch is resolved at the end of third cycle for conditional branches and then we refetch instruction because the instruction fetched in the second cycle is incorrect. We can see that the taken conditional instructions introduce two stalls in the pipeline.

The following figure show how untaken conditional branches are handled. The instruction i+1 is installed for once cycle because the branch is resolved at the end of third cycle for conditional branches. We can see that the untaken conditional instructions introduce one stall in the pipeline.

The frequency of unconditional branches = 0.01.

The frequency of taken conditional branches = $0.15 \times 0.6 = 0.9$.

	1	2	3	4	5	6	7	8	9
Taken branches	IF	ID	EX	WB					
i+1		IF	Stall	IF	ID	EX	WB		
i+2				Stall	IF	ID	EX	WB	
i+3						IF	ID	EX	WB

	1	2	3	4	5	6	7	8
Taken branches	IF	ID	EX	WB				
i+1		IF	Stall	ID	EX	WB		
i+2				IF	ID	EX	WB	
i+3					IF	ID	EX	WB

The frequency of taken unconditional branches = $0.15 \times 0.4 = 0.6$.

Overall, the average number of stall cycles = $(1 \times 1\%) + (2 \times 9\%) + (1 \times 6\%) = 0.25$, so the speedup with branch hazards = $\frac{1 \times 4}{1+0.25} = 3.2$, and the machine is $\frac{4}{3.2} - 1 = 25\%$ faster without branch hazards.

C.3 We begin with a computer implemented in single-cycle implementation. When the stages are split by functionality, the stages do not require exactly the same amount of time. The original machine had a clock cycle time of 7 ns. After the stages were split, the measured times were IF, 1 ns; ID, 1.5 ns; EX, 1 ns; MEM, 2 ns; and WB, 1.5 ns.The pipeline register delay is 0.1 ns.

a. What is the clock cycle time of the 5-stage pipelined machine?
answer: The slowest stage is MEM which needs 2 ns, so the clock cycle time $= 2 + 0.1 = 2.1$ ns.

b. If there is a stall every four instructions, what is the CPI of the new machine?
answer: $\frac{5}{4} = 1.25$ cycles.

c. What is th espeedup of the pipelined machine over the single-cycle machine?
answer: $\frac{1+1.5+1+2+1.5}{1.25 \times 2.1} = 2.67$.

d. If the pipelined machine had an infinite number of stages, what would its speedup be over the single-cycle machine?
answer: The clock cycle time would be the pipeline register delay, so the speedup = $\frac{1+1.5+1+2+1.5}{0.1} = 70$.

C.7 In this problem, we will explore how deepening the pipeline affects performance in two ways: faster clock cycle and increased stalls due to data and control hazards. Assume that the original machine is a 5-stage pipeline with a 1 ns clock cycle. The second machine is a 12-stage pipeline with a 0.6 ns clock cycle. The 5-stage pipeline experiences a stall due to a data hazard every five instructions, whereas the 12-stage pipeline experiences three stalls every eight instructions. In addition, branches constitute 20% of the instructions, and the misprediction rate for both machines is 5%.

a. What is the speedup of the 12-stage pipeline over the 5-stage pipeline, taking into account only data hazards?

answer:

Average instruction execution time of the 5-staged machine = $\frac{5+1}{5} \times 1 = 1.2$.

Average instruction execution time of the 12-staged machine = $\frac{8+3}{8} \times 0.6 = 0.825$ ns.

Speedup = $\frac{1.2}{0.825} = 1.45$.

b. If the branch mispredict penalty for the first machine is 2 cycles but the second machine is 5 cycles, what are the CPIs of each, taking into account the stalls due to branch mispredictions?

answer:

Average CPI of the 5-staged machine = $\frac{5+1}{5} + 0.2 \times 0.05 \times 2 = 1.22$ cycles.

Average CPI of the 5-staged machine = $\frac{8+3}{8} + 0.2 \times 0.05 \times 5 = 1.425$ cycles.

Speedup = $\frac{1.22 \times 1}{1.425 \times 0.6} = 1.43$.