Introduction to Computer Architecture

Assignment 2 - Solution

Due April 15, 2014

1. [70 = 10 x 7] Use the following code fragment:

```
Loop:    LD           R1, 0(R2)      ; load R1 from address 0+R2
         DADDI        R1, R1, #1     ; R1 = R1 + 1
         SD           R1, 0, (R2)    ; store R1 at address 0+R2
         DADDI        R2, R2, #4     ; R2 = R2 + 4
         DSUB         R4, R3, R2     ; R4 = R3 - R2
         BNEZ         R4, Loop       ; branch to Loop if R4 != 0
```

Assume that the initial value of R3 is R2 + 396.

a. Data hazards are caused by data dependences in the code. List all of the data dependences in the code above. Record the register, source instruction, and destination instruction; for example, there is a data dependency fro register R1 from the LD to DADDI, that is,

   R1 LD DADDI

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 "forwards" through the register file, as shown in Figure C.6. Use a pipeline timing chart like that in Figure C.5. Assume that the branch is handled by flushing the pipelining. If all memory references take 1 cycle, how many cycles does this loop take to execute?

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

d. 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.5. Assume that the branch is handled by predicting it as taken. If all memory references take 1 cycle, how many clock cycles does this loop take to execute?

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.5. 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?

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?

g. Using you 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.

Solution

a.

R1	LD	DADDI
R1	DADDI	SD
R2	LD	DADDI
R2	SD	DADDI
R2	DSUB	DADDI
R4	BNEZ	DSUB

b. Forwarding is performed only via the register file. Branch outcomes and targets are not known until the end of the execute stage. All instructions introduced to the pipeline prior to this point are flushed.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
LD R1, 0(R2)	F	D	X	M	W													
DADDI R1, R1, #1		F	s	s	D	X	M	W										
SD 0(R2), R1					F	s	s	D	X	M	W							
DADDI R2, R2, #4								F	D	X	M	W						
DSUB R4, R3, R2									F	s	s	D	X	M	W			
BNEZ R4, Loop												F	s	s	D	X	M	W
LD R1, 0(R2)																	F	D

Since the initial value of R3 is R2 + 396 and equal instance of the loop adds 4 to R2, the total number of iterations is 99. Notice that there are 8 cycles lost to RAW hazards including the branch instruction. Two cycles are lost after the branch because of the instruction flushing. It takes 16 cycles between loop instances; the total number of cycles is 98 × 16 + 18 = 1586 . The last loop takes two addition cycles since this latency cannot be overlapped with additional loop instances.

c. Now we are allowed normal bypassing and forwarding circuitry. Branch outcomes and targets are known now at the end of decode.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
`LD R1, 0(R2)`	F	D	X	M	W													
`DADDI R1, R1, #1`		F	D	s	X	M	W											
`SD R1, 0(R2)`			F	s	D	X	M	W										
`DADDI R2, R2, #4`					F	D	X	M	W									
`DSUB R4, R3, R2`						F	D	X	M	W								
`BNEZ R4, Loop`							F	s	D	X	M	W						
(incorrect instruction)									F	s	s	s	s					
`LD R1, 0(R2)`										F	D	X	M	W				

Again we have 99 iterations. There are two RAW stalls and a flush after the branch since the branch is taken. The total number of cycles is $9 \times 98 + 12 = 894$. The last loop takes three addition cycles since this latency cannot be overlapped with additional loop instances.

d. See the table below.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
`LD R1, 0(R2)`	F	D	X	M	W													
`DADDI R1, R1, #1`		F	D	s	X	M	W											
`SD R1, 0(R2)`			F	s	D	X	M	W										
`DADDI R2, R2, #4`					F	D	X	M	W									
`DSUB R4, R3, R2`						F	D	X	M	W								
`BNEZ R4, Loop`							F	s	D	X	M	W						
`LD R1, 0(R2)`									F	D	X	M	W					

Again we have 99 iterations. We still experience two RAW stalls, but since we correctly predict the branch, we do not need to flush after the branch. Thus, we have only $8 \times 98 + 12 = 796$.

e. See the table below.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
`LD R1, 0(R2)`	F1	F2	D1	D2	X1	X2	M1	M2	W1	W2										
`DADDI R1, R1, #1`		F1	F2	D1	D2	s	s	s	X1	X2	M1	M2	W1	W2						
`SD R1, 0(R2)`			F1	F2	D1	s	s	s	D2	X1	X2	M1	M2	W1	W2					
`DADDI R2, R2, #4`				F1	F2	s	s	s	D1	D2	X1	X2	M1	M2	W1	W2				
`DSUB R4, R3, R2`					F1	s	s	s	F2	D1	D2	s	X1	X2	M1	M2	W1	W2		
`BNEZ R4, Loop`									F1	F2	D1	s	D2	X1	X2	M1	M2	W1	W2	
`LD R1, 0(R2)`										F1	F2	s	D1	D2	X1	X2	M1	M2	W1	W2

We again have 99 iterations. There are three RAW stalls between the LD and ADDI, and one RAW stall between the DADDI and DSUB. Because of the branch prediction, 98 of those iterations overlap significantly. The total number of cycles is $10 \times 98 + 19 = 999$.

f. 0.9ns for the 5-stage pipeline and 0.5ns for the 10-stage pipeline.

g. CPI of 5-stage pipeline: $796/(99 \times 6) = 1.34$.
CPI of 10-stage pipeline: $999/(99 \times 6) = 1.68$.
Avg Inst Exe Time 5-stage: $1.34 \times 0.9 = 1.21$.
Avg Inst Exe Time 10-stage: $1.68 \times 0.5 = 0.84$.

2. [30 = 10 + 20] 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 5 instructions, whereas the 12-stage pipeline experiences 3 stalls every 8 instructions. In addition, branches constitute 20% of the instructions, and the misprediction rate for both machines is 5%.

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

Solution

C.7 a. Execution Time = $I \times CPI \times$ Cycle Time

Speedup = $(I \times 6/5 \times 1)/(I \times 11/8 \times 0.6) = 1.45$

b. $CPI_{5\text{-stage}} = 6/5 + 0.20 \times 0.05 \times 2 = 1.22$,

$CPI_{12\text{-stage}} = 11/8 + 0.20 \times 0.05 \times 5 = 1.425$

Speedup = $(1 \times 1.22 \times 1)/(1 \times 1.425 \times 0.6) = 1.17$