CS3853: Computer Architecture Spring 2017

Homework 2

Due in class Feb/16 (No delayed submission is allowed)

Total points: 38

Name:

Banner Id:

1. [3x6 points] Consider the 5-stage RISC pipeline that we discussed in the class. In an attempt to improve it, a student splits Mem stage into MemR (memory read) and MemW (memory write) stage. So, now the order of stages is IF, ID, EX, MemR, MemW, and WB. At any clock cycle, data memory can be accessed by at most one of the memory stages. A branch instruction is resolved in EX stage. Lets assume that of all instructions, 30% are loads, 20% are stores, 10% are branches and the rest are other type of instructions (ALU instructions).
   - For any ALU and load instruction i, there is a probability of 15%, 10%, 5% and, 3% that instruction i+1, i+2, i+3 and i+4 will depend on it respectively. When instruction i+1 depends on instruction i, no later instructions (i+2, i+3, ...) depend on i. The same is true for i+2, i+3 etc. In all cases, there is a 10% chance that the dependent instruction is a store operation with dependence only on register rt.
   - For any store instruction i, there is a probability of 10% that instruction i+1 is a load instruction.

   Assume full forwarding support. Now calculate the following quantities:
   a) Maximum and average number of stall cycles due to a load instruction
   b) Maximum and average number of stall cycles due to a store instruction
   c) Maximum and average number of stall cycles due to an ALU instruction
   d) Maximum and average number of stall cycles due to a branch instruction
   e) What is the CPI?
   f) If clock rate is 1 GHz, what is the MIPS rating?

a) i: LD R1, 0(R2) F D E MR MW W
90% i+1: ADD R3, R1, R2 F D S E MR MW W → 1 stall
10% i+1: SD R1, 0(R2) F D E MR MW W → 0 stall

i+2: 0 stall
i+3: 0 "
i+4: 0 "

Avg stall = 0.15 (0.9×1 + 0.1×0) = 0.135

Maximum stall = 1

b) i: SD R1, 0(R2) F D E MR MW W
10% i+1: LD R3, 0(R4) F D E S MR MW W → 1 stall
90% i+1: ADD R3, R2, R1 F D E MR MW W → 0 stall

Avg stall = 0.1×1 + 0.9×0 = 0.1 Max stall = 1

c) i: ADD R1, R2, R3 — F D E MR MW W

90% i+1: ADD R3, R1, R4 — F D E MR MW W → 0 Stall

10% i+1: SD R1, 0(R2) — F D E MR MW W → 0 "

i+2, i+3, i+4 → 0 Stall

∴ Avg & Max Stall = 0

d) BEQZ R1, Li — F D E MR MW W

ADD .... — F S S D F D E MR MW W → 2 Stall

Avg & Max Stall = 2

e) CPI = 1 + Avg Stall per instruction

$= 1 + 0.3 \times 0.135 + 0.2 \times 0.1 + 0.1 \times 2 + 0.4 \times 0$

$= 1.2605$

f) MIPS rating $= \frac{\text{clock rate}}{\text{CPI} \times 10^6}$

$= \frac{1 \times 10^9}{1.2605 \times 10^6}$

$= 793.34$

2. Basic Pipelining [15 points]

Consider the following code fragment:

```
Loop:   LW R1, 0(R2)
        DADDI R1, R1, 1
        SW R1, 0(R2)
        DADDI R2, R2, 4
        DADDI R4, R4, -4
        BNEZ R4, Loop
```

Consider the standard 5 stage pipeline machine (IF ID EX MEM WB). Assume the initial value of R4 is 396 and all memory accesses hit in the cache.

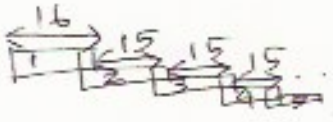
a. [6 points] Show the timing of the above code fragment for one iteration **as well as for the load of the second iteration**. For this part, assume there is no forwarding or bypassing hardware. Assume a register write occurs in the first half of the cycle and a register read occurs in the last half of the cycle. Also, assume that branches are resolved in the *memory* stage and are handled by flushing the pipeline. Use a pipeline timing chart to show the timing as below (expand the chart if you need more cycles). How many cycles does this loop take to complete (for all iterations, not just one iteration)?

Instruction	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	16 17 18 19 20
LW R1, 0(R2)	F	D	X	M	W											
DADDI R1, R1, 1		F	S	S	D	E	M	W								
SW R1, 0(R2)					F	S	S	D	E	M	W					
DADDI R2, R2, 4								F	D	E	M	W				
DADDI R4, R4, -4									F	D	E	M	W			
BNEZ R4, Loop										F	S	S	D	E	M	W
LW R1, 0(R2)																F D E M W

Num of iteration $= \frac{396}{4} = 99$

Number of overlapping cycle = 1/iteration

Total cycle = 16 + 15 × 98



= 1486

b. [6 points] Show the timing for the same instruction sequence for the pipeline with full forwarding and bypassing hardware (as discussed in class). Assume that branches are resolved in the MEM stage and are predicted as not taken. How many cycles does this loop take to complete?

Instruction	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
LW R1, 0(R2)	F	D	X	M	W										
DADDI R1, R1, 1		F	D	S	E	M	W								
SW R1, 0(R2)			F	S	D	E	M	W							
DADDI R2, R2, 4					F	D	E	M	W						
DADDI R4, R4, -4						F	D	E	M	W					
BNEZ R4, Loop							F	D	E	M	W				
LW R1, 0(R2)								S	S	S	F	D	E	M	W

Total cycle = 11 + 10 × 98
= 991

c. [2+1 points] How does the branch delay slot improve performance? Point out where in your solution for part b that it would be beneficial.

Branch delay slot improves performance by executing something useful (most of the time) in cycles that would, otherwise, have been stall cycles.

For part (b), we have 3 delay slots. We can move DADDI, SW and DADDI instructions into the delay slot. As a result, the first iteration would take 6 cycles. Total loop would take 6 + 5 × 98 = 496 cycles.

d. [2 pt] What are the different types of hazard in a pipeline?

3 types of hazard → Structural hazard
Data hazard
control hazard

e. [3 pt] What are the types of data hazard? Which ones are possible in MIPS R4000? Give examples.

2 types of data hazards are possible

1: RAW

LD R1, D(R2) IF IS RF EX DF DS TC WB
ADD R3, R1, R2 IF IS RF S S EX DF DS TC WB

2: WAW

	1	2	3	4	5	6	7	8
MUL F1, F2, F3	U	E+M	M	M	M	N	N+A	R
ADD F1, F3, F4		U	S+A	A+R	R+S	Stall	Stall	Stall