CS 530 Fall 2014
Computer Architecture

Homework 3 Solutions

1. Problem 3.3: *Consider a multiple-issue design with two execution pipelines and full forwarding. How many cycles does the loop require?*

Instruction type	Latency
LD	+4
SD	+1
Integer ADD, SUB	+0
Branches	+1
ADDD	+1
MULTD	+5
DIVD	+12

Instructions	Issues at clock cycle number	Executes at clock cycle number	Memory access at clock cycle number	Write CDB at clock cycle number	Comment
LD F2, 0(Rx)	1	2	3	7	
DIVD F8, F2, F0	1	7	-	20	Wait: LD
MULTD F2, F6, F2	2	7	-	13	Wait: LD
LD F4, 0(Ry)	2	3	4	8	
ADDD F4, F0, F4	3	8	-	10	Wait: LD
ADDD F10, F8, F2	3	20	-	22	Wait: MULTD, DIVD
ADDI Rx, Rx, #8	4	5	-	6	
ADDI Ry, Ry, #8	4	5	-	6	
SD F4, 0(Ry)	5	6	10	-	Wait: ADDD
SUB R20, R4, Rx	5	6	-	7	
BNZ R20, Loop	6	7	-	-	

The loop requires 22 cycles to complete.

2. Problem 3.4: *Recite at least two reasons why that could be hazardous and will require special considerations in the microarchitecture. Give an example of two instructions from the code in Figure 3.48 that demonstrates this hazard.*

   a. WAW hazards – Instruction N+1 broadcast register value via CDB prior to instruction N.
   b. Imprecise exceptions and interrupt handling – $DIVD$ execution cycles 7-20 and $MULTD$ execution cycles 7-13.

3. Problem 3.11: Assume 5-stage pipeline. All ops are one cycle except LW and SW, which are 1+2 cycles, and branches, which are 1+1 cycles. There is no forwarding.

	Clock cycle number																				
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
LW R3,0(R0)	IF	ID	EX	M	M	M	WB															
LW R1,0(R3)		IF	St	St	St	St	ID	EX	M	M	M	WB										
ADDI R1,R1,#1							IF	St	St	St	St	ID	EX	M	WB							
SUB R4,R3,R2												IF	ID	EX	M	WB						
SW R1,0(R3)													IF	St	ID	EX	M	M	M	WB		
BNZ R4,Loop															IF	ID	ID	EX	St	M	WB	
	Next instruction, ideal																					
Next instr.																IF	…					

Assumptions: Multi-cycle operations are not pipelined.

a. How many clock cycles per loop iteration are lost to branch overhead?

	Clock cycle number																				
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
LW R3,0(R0)	IF	ID	EX	M	M	M	WB															
LW R1,0(R3)		IF	St	St	St	St	ID	EX	M	M	M	WB										
ADDI R1,R1,#1							IF	St	St	St	St	ID	EX	M	WB							
SUB R4,R3,R2												IF	ID	EX	M	WB						
SW R1,0(R3)													IF	St	ID	EX	M	M	M	WB		
BNZ R4,Loop															IF	ID	ID	EX	St	M	WB	
	Next instruction																					
Next instr.																					IF	…

The branch instruction incurs in a 5 cycle overhead. The next instruction is fetched at clock cycle 21 (wait until WB stage). The ideal case is that the next instruction will perform the instruction fetch at clock cycle 16.

$$21 - 16 = 5\ cycles$$

b. *Assume a static branch predictor, how many clock cycles are wasted on branch overhead?*

	Clock cycle number																				
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
LW R3,0(R0)	IF	ID	EX	M	M	M	WB															
LW R1,0(R3)		IF	St	St	St	St	ID	EX	M	M	M	WB										
ADDI R1,R1,#1							IF	St	St	St	St	ID	EX	M	WB							
SUB R4,R3,R2												IF	ID	EX	M	WB						
SW R1,0(R3)													IF	St	ID	EX	M	M	M	WB		
BNZ R4,Loop															IF	ID	ID	EX	St	M	WB	
	Next instruction																					
Next instr.																		IF	...			

The branch instruction incurs in a 2 cycle overhead. The static branch predictor is capable of recognizing the backwards branch in the ID stage. The next instruction can be fetched at cycle 18.

$$18 - 16 = 2\ cycles$$

c. *Assume a dynamic branch predictor, how many clock cycles are lost on a correct prediction?*

	Clock cycle number																				
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
LW R3,0(R0)	IF	ID	EX	M	M	M	WB															
LW R1,0(R3)		IF	St	St	St	St	ID	EX	M	M	M	WB										
ADDI R1,R1,#1							IF	St	St	St	St	ID	EX	M	WB							
SUB R4,R3,R2												IF	ID	EX	M	WB						
SW R1,0(R3)													IF	St	ID	EX	M	M	M	WB		
BNZ R4,Loop															IF	ID	ID	EX	St	M	WB	
	Next instruction																					
Next instr.																IF	...					

The dynamic branch predictor incurs in 0 cycle overhead since the next instruction is executed without delay.

4. Problem 3.15: *Tomasulo's algorithm.*

FU Type	Cycles in EX	Number of FUs	Number of RS
Integer	1	1	5
FP adder	10	1	3
FP multiplier	15	1	2

Assumptions: FU are not pipelined, no forwarding, pipeline is IF/ID/IS/EX/WB, loads require 1 cycle, IS and WB require 1 cycle, 5 load buffers, 5 store buffers, BNEZ requires 1 cycle.

a. *Use the single-issue Tomasulo MIPS pipeline; show the number of stall cycles for each instruction and what clock cycle each instruction begins execution for three iterations of the loop. How many cycles does each loop iteration take?*

Iteration	Instruction	Issues	Executes	Memory access	CDB	Comment	RS in use (int,add,mult,L,S)
1	L.D F2, 0(R1)	1	2	3	3		(0,0,0,1,0)
1	MUL.D F4, F2, F0	2	4	-	19	Wait: L.D	(0,0,1,1,0)
1	L.D F6, 0(R2)	3	4	5	5		(0,0,1,1,0)
1	ADD.D F6, F4, F6	4	20	-	30	Wait: L.D, MUL.D	(0,1,1,1,0)
1	S.D F6, 0(R2)	5	6	31	-	Wait: ADD.D	(0,1,1,0,1)
1	DADDIU R1, R1, #8	6	7	-	8		(1,1,1,0,1)
1	DADDIU R2, R2, #8	7	8	-	9		(2,1,1,0,1)
1	DSLTU R3, R1, R4	8	9	-	10		(2,1,1,0,1)
1	BNEZ R3, foo	9	11	-	-	Wait: DSLTU	(2,1,1,0,1)
2	L.D F2, 0(R1)	10	12	13	13	Wait: BNEZ	(0,1,1,1,1)
2	MUL.D F4, F2, F0	11	19	-	34	Wait: L.D, MUL.D	(0,1,2,1,1)
2	L.D F6, 0(R2)	12	13	14	14		(0,1,2,2,1)
2	ADD.D F6, F4, F6	13	35	-	45	Wait: L.D, MUL.D	(0,2,2,1,1)
2	S.D F6, 0(R2)	14	15	46	-	Wait: ADD.D	(0,2,2,0,2)
2	DADDIU R1, R1, #8	15	16	-	17		(1,2,2,0,2)
2	DADDIU R2, R2, #8	16	17	-	18		(2,2,2,0,2)
2	DSLTU R3, R1, R4	17	18	-	20	Stall: CDB	(2,2,2,0,2)
2	BNEZ R3, foo	18	21	-	-	Wait: DSLTU	(2,2,2,0,2)
3	L.D F2, 0(R1)	19	22	23	23	Wait: BNEZ	(2,2,1,1,2)
3	MUL.D F4, F2, F0	20	34	-	49	Wait: L.D, MUL.D	(1,2,2,1,2)
3	L.D F6, 0(R2)	21	23	24	24	Stall: L.D in EX	(0,2,2,2,2)
3	ADD.D F6, F4, F6	22	50	-	60	Wait: L.D, MUL.D	(0,3,2,2,2)
3	S.D F6, 0(R2)	23	24	61	-	Wait: ADD.D	(0,3,2,1,3)
3	DADDIU R1, R1, #8	24	25	-	26		(1,3,2,0,3)
3	DADDIU R2, R2, #8	25	26	-	27		(2,3,2,0,3)
3	DSLTU R3, R1, R4	26	27	-	28		(2,3,2,0,3)
3	BNEZ R3, foo	27	29	-	-	Wait: DSLTU	(2,3,2,0,3)

Cycles per loop iteration based on latency of last loop instruction:

Iteration 1 – 31 cycles

Iteration 2 – 16 cycles

Iteration 3 – 16 cycles

b. *Repeat part (a) using a two-issue Tomasulo algorithm and a fully pipelined floating-point unit. How many cycles does each loop iteration take?*

Iteration	Instruction	Issues	Executes	Memory access	CDB	Comment	RS in use (int,add,mult,L,S)
1	L.D F2, 0(R1)	1	2	3	3		(0,0,1,1,0)
1	MUL.D F4, F2, F0	1	4	-	19	Wait: L.D	
1	L.D F6, 0(R2)	2	3	4	4		(0,1,1,2,0)
1	ADD.D F6, F4, F6	2	20	-	30	Wait: L.D, MUL.D	
1	S.D F6, 0(R2)	3	4	31	-	Wait: ADD.D	(1,1,1,1,1)
1	DADDIU R1, R1, #8	3	4	-	5		
1	DADDIU R2, R2, #8	4	5	-	6		(3,1,1,0,1)
1	DSLTU R3, R1, R4	4	6	-	7	Wait: DADDIU	
1	BNEZ R3, foo	5	8	-	-	Wait: DSLTU	(3,1,1,0,1)
2	L.D F2, 0(R1)	6	9	10	10	Wait: BNEZ	(2,1,2,1,1)
2	MUL.D F4, F2, F0	6	11	-	26	Wait: L.D	
2	L.D F6, 0(R2)	7	10	11	11	Stall: L.D in EX	(1,2,2,2,1)
2	ADD.D F6, F4, F6	7	27	-	37	Wait: L.D, MUL.D	
2	S.D F6, 0(R2)	8	9	38	38	Wait: ADD.D	(1,2,2,2,2)
2	DADDIU R1, R1, #8	8	9	-	10		
2	DADDIU R2, R2, #8	9	10	-	12	Stall: L.D in CDB	(3,2,2,2,2)
2	DSLTU R3, R1, R4	9	11	-	12	Wait: DADDIU	
2	BNEZ R3, foo	10	13	-	-	Wait: DSLTU	(3,2,2,1,2)
3	L.D F2, 0(R1)	11	14	15	15	Wait: BNEZ	(3,2,3,1,2)
3	MUL.D F4, F2, F0	11	16	-	31	Wait: L.D	
3	L.D F6, 0(R2)	12	15	16	16	Stall: L.D in EX	(1,3,3,2,2)
3	ADD.D F6, F4, F6	12	32	-	42	Wait: L.D, MUL.D	
3	S.D F6, 0(R2)	13	14	43	43	Wait: ADD.D	(1,3,3,2,3)
3	DADDIU R1, R1, #8	13	14	-	15		
3	DADDIU R2, R2, #8	14	15	-	17	Stall: L.D in CDB	(3,3,3,2,3)
3	DSLTU R3, R1, R4	14	16	-	17	Wait: DADDIU	
3	BNEZ R3, foo	15	18	-	-	Wait: DSLTU	(3,3,3,1,3)

Cycles per loop iteration based on latency of last loop instruction:

Iteration 1 – 31 cycles

Iteration 2 – 8 cycles

Iteration 3 – 6 cycles

5. Problem 3.16: *Use the hardware configuration and latencies from Problem 3.14 and find a code sequence of no more than 10 instructions where Tomasulo's algorithm must stall due to CDB contention.*

Instruction	Issues	Executes	Memory access	CDB
MUL.D F4, F2, F0	1	2	-	17
DADDIU R1, R1, #8	2	3	-	4
L.D F0, 0(R1)	3	5	6	6
ADD.D F6, F2, F0	4	7	-	~~17~~ 18

6. Problem C.1

a. List all of the data dependencies in the code.

Register	Source	Destination
R1	LD R1, 0(R2)	DADDI R1, R1, #1
R1	LD R1, 0(R2)	SD R1, 0(R2)
R1	DADDI R1, R1, #1	SD R1, 0(R2)
R2	DADDI R2, R2, #4	LD R1, 0(R2)
R2	DADDI R2, R2, #4	SD R1, 0(R2)
R2	DADDI R2, R2, #4	DSUB R4, R3, R2
R4	DSUB R4, R3, R2	BNEZ R4, Loop
R4 (via R2)	DADDI R2, R2, #4	BNEZ R4, Loop

b. Show timing of instruction sequence without any forwarding or bypassing hardware but assume that register read and a write in the same clock cycle "forwards" through the register file. How many cycles does this loop take to execute?

	Clock cycle number																	
Instruction	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
LD	*IF*	*ID*	*EX*	*M*	*WB*													
DADDI		*IF*	*St*	*St*	*ID*	*EX*	*M*	*WB*										
SD					*IF*	*St*	*St*	*ID*	*EX*	*M*	*WB*							
DADDI								*IF*	*ID*	*EX*	*M*	*WB*						
DSUB									*IF*	*St*	*St*	*ID*	*EX*	*M*	*WB*			
BNEZ												*IF*	*St*	*St*	*ID*	*EX*	*M*	*WB*
Next instr.															*flush*	*IF*	*…*	

The first iteration takes 18 clock cycles and the remaining iterations take 15 cycles. Branch result is known at end of ID stage. The number of loops is $\frac{396}{4} = 99$, hence, the loop takes 1488 clock cycles.

$$(18\ cycles \times 1\ iteration) + (15\ cycles\ \times 98\ iterations) = 1488\ cycles$$

c. *Show timing of instruction sequence with full forwarding and bypassing hardware. Assume that the branch is handled by predicting it as not taken. How many cycles does this loop take to execute?*

	Clock cycle number												
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13
LD	IF	ID	EX	M	WB								
DADDI		IF	ID	St	EX	M	WB						
SD			IF	St	ID	EX	M	WB					
DADDI					IF	ID	EX	M	WB				
DSUB						IF	ID	EX	M	WB			
BNEZ							IF	St	ID	EX	M	WB	
Next instr.									IF	idle	idle	idle	idle
Next instr.										IF	…		

The first iteration takes 12 clock cycles and all other iterations take 9 clock cycles to execute. The loop takes 894 clock cycles to execute.

$$(12\ cycles \times 1\ iteration) + (9\ cycles\ \times 98\ iterations) = 894\ cycles$$

d. *Show timing of instruction sequence with full forwarding and bypassing hardware. Assume that the branch is handled by predicting it as taken. How many cycles does this loop take to execute?*

	Clock cycle number											
Instruction	1	2	3	4	5	6	7	8	9	10	11	12
LD	IF	ID	EX	M	WB							
DADDI		IF	ID	St	EX	M	WB					
SD			IF	St	ID	EX	M	WB				
DADDI					IF	ID	EX	M	WB			
DSUB						IF	ID	EX	M	WB		
BNEZ							IF	St	ID	EX	M	WB
Next instr.									IF	…		

The first iteration takes 12 clock cycles and all other iterations take 8 clock cycles to execute. The loop takes 796 clock cycles to execute.

$$(12\ cycles \times 1\ iteration) + (8\ cycles\ \times 98\ iterations) = 796\ cycles$$

7. Problem C.7

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

   Assume that the pipeline is full and running and will continue running, that is, we are not considering the cases for filling and flushing the pipeline. With this assumption an instruction is completed in every clock cycle. Consider X instructions with the corresponding stall cycle ratios for data hazards, then the total number of clock cycles, D, required to execute X instructions is

$$D_{5-stage} = X\left(1+\frac{1}{5}\right) \qquad D_{12-stage} = X\left(1+\frac{3}{8}\right)$$

   To compute the speedup we find the execution times by multiplying the number of cycles with the clock period,

$$T_{5-stage} = 1ns \times D_{5-stage} \qquad T_{12-stage} = 0.6ns \times D_{12-stage}$$
$$T_{5-stage} = 1.20X\ ns \qquad T_{12-stage} = 0.825X\ ns$$

$$Speedup_{12-stage} = \frac{1.20X\ ns}{0.825X\ ns} \approx 1.45$$

   b. *What are the CPIs of each, taking into account the stalls due to branch mispredictions?*

   Using the same assumption as in part (a), CPI equals 1 when no stalls occur. In part (a) we computed the number of clock cycles required by each pipeline for executing X instructions considering only data hazard stalls,

$$D_{5-stage} = 1.20X \qquad D_{12-stage} = 1.375X$$

   Let us compute branch stall cycles, B, due to mispredictions,

$$B_{5-stage} = X(0.2 \times 0.05 \times 2\ cycles)$$
$$B_{12-stage} = X(0.2 \times 0.05 \times 5\ cycles)$$

   Combine the data hazard and branch stalls to find the total number of cycles, C,

$$C = D + B$$
$$C_{5-stage} = 1.20X + 0.02X = 1.22X$$
$$C_{12-stage} = 1.375X + 0.05X = 1.425X$$

   The CPI follows,

$$CPI_{5-stage} = \frac{1.22X}{X} = 1.22$$

$$CPI_{12-stage} = \frac{1.425X}{X} = 1.425$$

8. Problem C.12

   a. *How many clock cycles do each loop iteration take, counting from when the first instruction enters the WB stage to when the last instruction enters the WB stage?*

Functional Unit	Latency	Initiation Interval
Integer ALU	0	1
Loads	0	1
FP add	3	1
FP multiply	6	1

	Clock cycle number																					
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
L.D	IF	ID	EX	M	WB																		
MUL.D		IF	ID	St	M1	M2	M3	M4	M5	M6	M7	M	WB										
L.D			IF	St	ID	EX	M	WB															
ADD.D					IF	ID	St	St	St	St	St	A1	A2	A3	A4	M	WB						
S.D						IF	St	St	St	St	St	ID	St	St	St	EX	M	WB					
DADDIU												IF	St	St	St	ID	EX	M	WB				
DADDIU																IF	ID	EX	M	WB			
SGTIU																	IF	ID	EX	M	WB		
BEQZ																		IF	St	ID	EX	M	WB

The DAXPY loop takes 19 clock cycles to execute.