COMPUTER ARCHITECTURE & ASSEMBLY LANGUAGE
14:332:331
Rutgers University
Fall 2016

Homework 4 Solution

1. Please determine the control signals for the following instructions in the un-pipelined MIPS.
   a) andi $t0, $t1, #12
   b) lw $t0,12($t2)
   c) sw $t4,12($t7)

Sol.

a)

Control Signal	Value	Control Signal	Value
RegDst	0	ALUOp1	1
Jump	0	ALUOp0	0
Branch	0	MemWrite	0
MemRead	X	ALUSrc	1
MemtoReg	0	RegWrite	1

b)

Control Signal	Value	Control Signal	Value
RegDst	0	ALUOp1	0
Jump	0	ALUOp0	0
Branch	0	MemWrite	0
MemRead	1	ALUSrc	1
MemtoReg	1	RegWrite	1

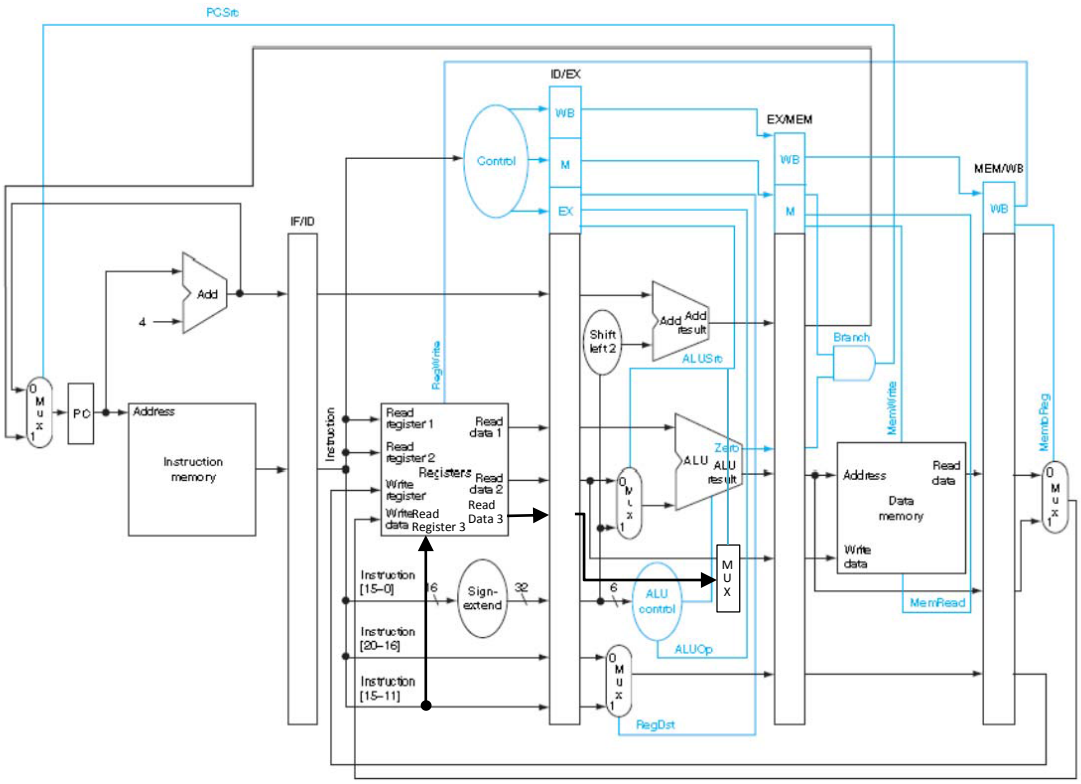
c)

Control Signal	Value	Control Signal	Value
RegDst	X	ALUOp1	0
Jump	0	ALUOp0	0
Branch	0	MemWrite	1
MemRead	X	ALUSrc	1
MemtoReg	X	RegWrite	0

2. Please apply required changes (if any) to the datapath and controller of singly cycle MIPS to support the following instruction:

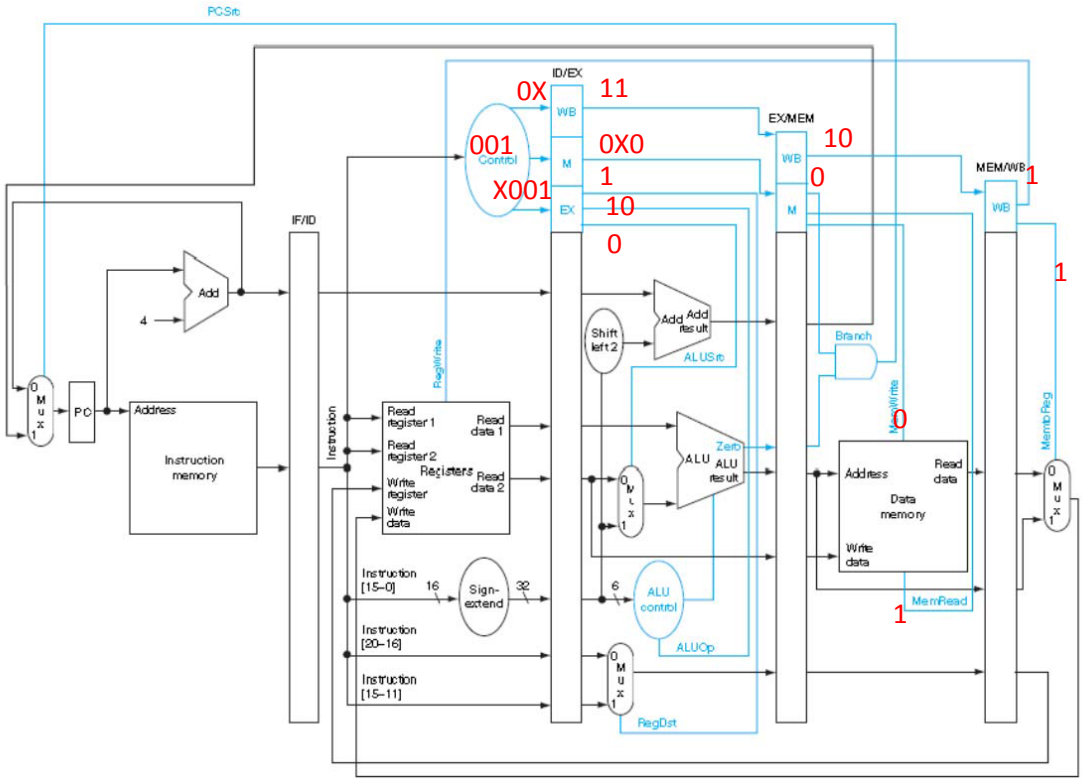
   swi $t1, $t2($t4) #memory address is ($t2) + ($t4)

Sol. The Register File should have a third read port and a third data output, because we read one register for the data that will be stored in the memory and two registers to calculate the address. A multiplexer is also added in the memory data input port that will either select the correct register compared to sw and swi. The select signal of that multiplexer can be driven from ALUSrc. The control signals here will be changed. The address calculation here is completed by adding the two registers together.



Control Signal	Value	Control Signal	Value
RegDst	X	ALUOp1	0
Jump	0	ALUOp0	0
Branch	0	MemWrite	1
MemRead	X	ALUSrc	0
MemtoReg	X	RegWrite	0

3. We would like to run the following code on the pipeline MIPS. Please show the values of all the control signals only in clock cycle 6 (you can find the figure in course lectures too). Note that the first instruction (add) is fetched in clock cycle 1.



add $1, $3, $5
and $10, $8, $3
lw $4, 16($3)
sub $11, $2, $7
sw $2,100($6)

Sol.

cycle	1	2	3	4	5	6	7	8	9	10
Instruction										
add $1, $3, $5	IF	ID	EX	MEM	WB					
and $10, $8, $3		IF	ID	EX	MEM	WB				
lw $4, 16($3)			IF	ID	EX	MEM	WB			
sub $11, $2, $7				IF	ID	EX	MEM	WB		

sw $2, 100($6)					IF	ID	EX	MEM	WB	

So in Clock 6, we have the control signals as shown in the figure.

4. Consider the following instruction sequence. We want to run this code on pipelined MIPs.

add R3, R2,R7
lw R4,20(R3)
and R6, R5, R4
sw R3, -40(R5)
add R4, R6, R3
sub R5, R6, R4

a) Show all the data hazards (assuming that there is no forwarding).
b) Add no operations between these instructions as required to resolve the data hazards. Assume that there is no forwarding mechanism. How many clock cycles needed to run this code?
c) Repeat part b but assume that there is a full forwarding mechanism.

For part b and part c show the pipeline stages as IF,ID,EX,MEM,WB for each instruction as below:
3 INST1: IF ID EX MEM WB INST2: IF ……..

d) To reduce clock cycle time, we are considering a split of the MEM stage into two stages. Please find all hazards in this 6-stage pipeline for the same instruction sequence with and without forwarding.

Sol.

lw R4,20(R3) : The value of R3 is required which is not yet computed
and R6,R5,R4 : The value of R4 is required but Load is not yet finished
add R4,R6,R3 : The value of R6 is not written back yet.
sub R5, R6, R4 : The values of R4 is required.

b) 17 clock cycles as below

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
add R3, R2, R7	IF	ID	EX	MEM	WB												
		ST															
			ST														
lw R4, 20(R3)				IF	ID	EX	MEM	WB									
					ST												
						ST											
and R6, R5, R4							IF	ID	EX	MEM	WB						
sw R3, -40(R5)								IF	ID	EX	MEM	WB					
									ST								
add R4, R6, R3										IF	ID	EX	MEM	WB			
											ST						
												ST					
sub R5, R6, R4													IF	ID	EX	MEM	WB

c) 11 clock cycles as below:

	1	2	3	4	5	6	7	8	9	10	11
add R3, R2, R7	IF	ID	EX	MEM	WB						
lw R4, 20(R3)		IF	ID	EX	MEM	WB					
			ST								
and R6, R5, R4				IF	ID	EX	MEM	WB			
sw R3, -40(R5)					IF	ID	EX	MEM	WB		
add R4, R6, R3						IF	ID	EX	MEM	WB	
sub R5, R6, R4							IF	ID	EX	MEM	WB

d) (without Forwarding):22 clock cycles as below:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
add R3, R2, R7	IF	ID	EX	M1	M2	WB																
		ST																				
			ST																			
				ST																		
lw R4, 20(R3)					IF	ID	EX	M1	M2	WB												
						ST																
							ST															
								ST														
and R6, R5, R4									IF	ID	EX	M1	M2	WB								
sw R3, -40(R5)										IF	ID	EX	M1	M2	WB							
											ST											
												ST										
add R4, R6, R3													IF	ID	EX	M1	M2	WB				

														ST								
															ST							
																ST						
sub R5, R6, R4																	IF	ID	EX	M1	M2	WB

With forwarding: 13 clock cycle as below

	1	2	3	4	5	6	7	8	9	10	11	12	13
add R3, R2, R7	IF	ID	EX	M1	M2	WB							
lw R4, 20(R3)		IF	ID	EX	M1	M2	WB						
			ST										
				ST									
and R6, R5, R4					IF	ID	EX	M1	M2	WB			
sw R3, -40(R5)						IF	ID	EX	M1	M2	WB		
add R4, R6, R3							IF	ID	EX	M1	M2	WB	
sub R5, R6, R4								IF	ID	EX	M1	M2	WB

5. Consider a 5-stage pipelined machine (M1) which employs data forwarding. Assuming that the frequency of Load, Store, and ALU instructions are 30%, 20%, 50%. Suppose the following:
   - 40% of all load instructions are followed immediately by an instruction that uses the value just loaded in.
   - 30% of all store instructions are immediately preceded by an R-type instruction that computes the value that is written to memory.
   - 20% of all ALU instructions are immediately followed by another ALU instruction that uses the value just computed.

   What is the throughput (in instructions per cycle) while executing this program? Assume all instructions go through all five stages and there is no additional pipelining overhead, except stalls

Sol.

For Load instructions followed by an instruction that uses the loaded value a stall is required. For example consider the following case. Here we need one stall between these 2 instructions.

LW R1,10(R2)

Add R3,R1,R4

For Store Instructions preceded by R-type instruction computing the value, no stall is required. For example consider the following case. Here we do not need any stall between these 2 instructions.

Add R3,R1,R4

SW R10,10(R3)

For ALU Instructions no stall is required. For example consider the following case. Here we do not need any stall between these 2 instructions.

Add R3,R1,R4

Add R10,R3,R5

Total CPI = ideal CPI + cycles added due to Stalls

CPI=1+0.0.3*0.40*1=1.12

Instruction per cycle = 1/1.12 = 0.8929