Computer Architecture HW #2

Mark F

Due: Friday, Sept. 11 (5 PM in ITT 305 mailbox or under my office door, ITT 313)

1. You are to assume the same 6-stage pipeline discussed in class (and textbook) when answering these questions. Assume that the first register in an arithmetic operation is the destination register, e.g., in "ADD R3, R2, R1" register R3 receives the result of adding registers R2 and R1.

a. What would the timing be **without** bypass-signal paths/forwarding (use "stalls" to solve the data hazard)? (This code might require more or less that 15 cycles)

	Time →														
Instructions	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
ADD R3, R2, R1	FI	DI	CO	FO	EI	WO									
STORE R3, 8(R4)		FI	DI	CO	–	–	FO	EI	WO						
LOAD R4, 16(R3)			FI	–	–	–	DI	CO	FO	EI	WO				
SUB R5, R3, R4							FI	DI	CO	–	–	FO	EI	WO	
LOAD R1, 4(R5)								FI	–	–	–	–	–	–	DI
MUL R6, R1, R8															FI
ADD R7, R6, R9															
SUB R2, R3, R4															

(Assume that a register **cannot** be written and the new value read in the same stage.)

b. What would the timing be **with** bypass-signal paths?
(This code might require more that 15 cycles)

	Time →														
Instructions	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
ADD R3, R2, R1	FI	DI	CO	FO	EI	WO									
STORE R3, 8(R4)		FI	DI	CO	FO	EI	WO								
LOAD R4, 16(R3)			FI	DI	–	CO	FO	EI	WO						
SUB R5, R3, R4				FI	–	DI	CO	FO	EI	WO					
LOAD R1, 4(R5)						FI	DI	–	–	CO	FO	EI	WO		
MUL R6, R1, R8							FI	–	–	DI	CO	EI	EI	WO	
ADD R7, R6, R9										FI	DI	CO	FO	EI	WO
SUB R2, R3, R4											FI	DI	CO	FO	EI WO

(Assume that a register **cannot** be written and the new value read in the same stage.)

Computer Architecture HW #2

Mark F.

Due: Friday, Sept. 11 (5 PM in ITT 305 mailbox or under my office door, ITT 313)

1. You are to assume the same 6-stage pipeline discussed in class (and textbook) when answering these questions. Assume that the first register in an arithmetic operation is the destination register, e.g., in "ADD R3, R2, R1" register R3 receives the result of adding registers R2 and R1.

a. What would the timing be **without** bypass-signal paths/forwarding (use "stalls" to solve the data hazard)? (This code might require more or less that 15 cycles)

(Continued)	16	17	18	19	20	21	22	Time 23→24	25	26					
Instructions	~~1~~	~~2~~	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
ADD R3, R2, R1	FI	DI	CO	FO	EI	WO									
STORE R3, 8(R4)		FI													
LOAD R4, 16(R3)															
SUB R5, R3, R4															
LOAD R1, 4(R5)	CO	FO	EI	WO											
MUL R6, R1, R8	DI	CO	–	–	FO	EI	WO								
ADD R7, R6, R9	FI	DI	–	–	CO	–	–	FO	EI	WO					
SUB R2, R3, R4		FI	–	–	DI	–	–	CO	FO	EI	WO				

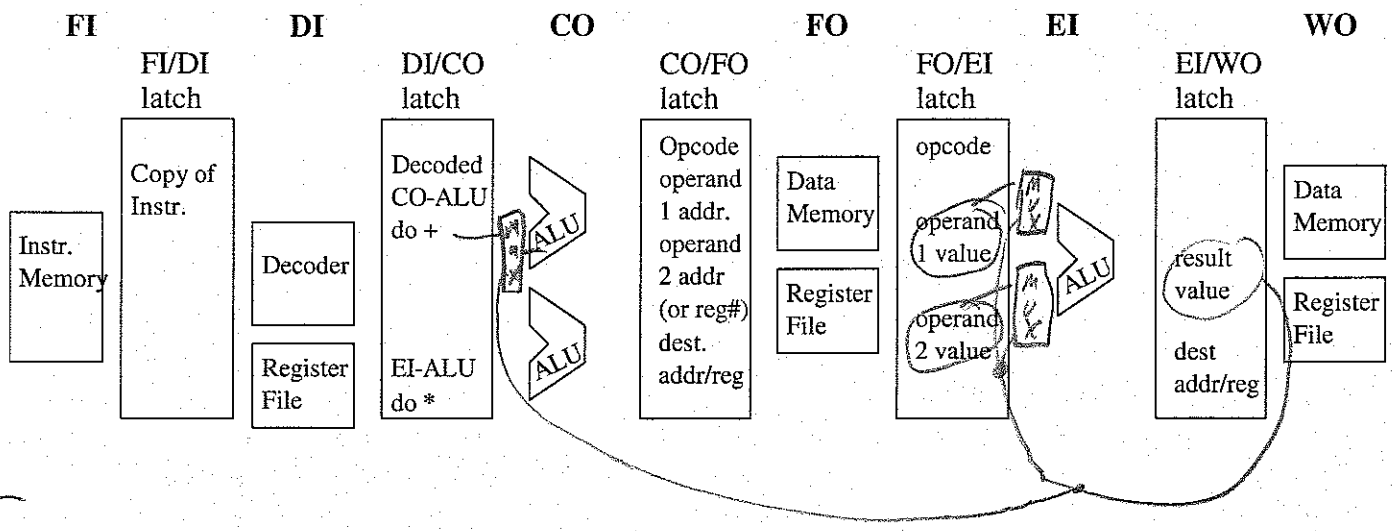
(Assume that a register **cannot** be written and the new value read in the same stage.)

b. What would the timing be **with** bypass-signal paths?
(This code might require more that 15 cycles)

	Time →														
Instructions	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
ADD R3, R2, R1	FI	DI	CO	FO	EI	WO									
STORE R3, 8(R4)		FI													
LOAD R4, 16(R3)															
SUB R5, R3, R4															
LOAD R1, 4(R5)															
MUL R6, R1, R8															
ADD R7, R6, R9															
SUB R2, R3, R4															

(Assume that a register **cannot** be written and the new value read in the same stage.)

c. Draw ALL the bypass-signal paths needed for the above example.



HW2-3