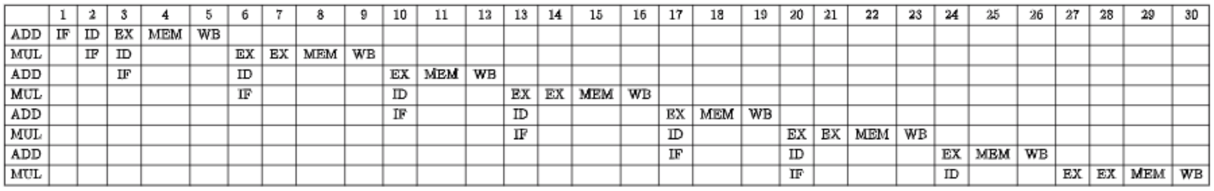
Q1)

$$\text{Speedup(def in question)} = \frac{\text{Time without Operand Forwarding}}{\text{Time with Operand Forwarding}}$$

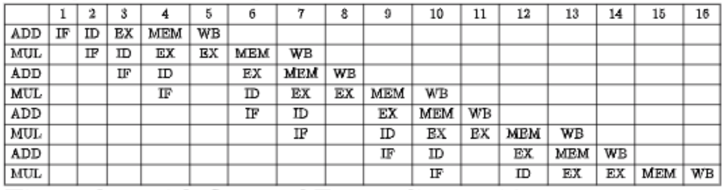
Without Operand Forwarding:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
ADD	IF	ID	EX	MEM	WB																									
MUL		IF	ID			EX	EX	MEM	WB																					
ADD			IF			ID				EX	MEM	WB																		
MUL						IF				ID			EX	EX	MEM	WB														
ADD										IF			ID				EX	MEM	WB											
MUL													IF				ID			EX	EX	MEM	WB							
ADD																	IF			ID				EX	MEM	WB				
MUL																				IF				ID			EX	EX	MEM	WB

$$\text{Time taken without Operand Forwarding} = 30$$

With Operand Forwarding:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADD	IF	ID	EX	MEM	WB											
MUL		IF	ID	EX	EX	MEM	WB									
ADD			IF	ID		EX	MEM	WB								
MUL				IF		ID	EX	EX	MEM	WB						
ADD						IF	ID		EX	MEM	WB					
MUL							IF		ID	EX	EX	MEM	WB			
ADD									IF	ID		EX	MEM	WB		
MUL										IF		ID	EX	EX	MEM	WB

$$\text{Time taken with Operand Forwarding} = 16$$

$$\text{Speedup} = \frac{\text{Time without Operand Forwarding}}{\text{Time with Operand Forwarding}} = \frac{30}{16} = 1.875$$

Q2)

Given, total number of instructions (n) = 100

Number of stages (k) = 5

Since, if n instructions take c cycle, so (c-1) stalls will occur for these instructions.

Therefore, the number of clock cycles required = Total number of cycles required in general case + Extra cycles required (here, in PO stage),

= (n + k - 1) + Extra cycles

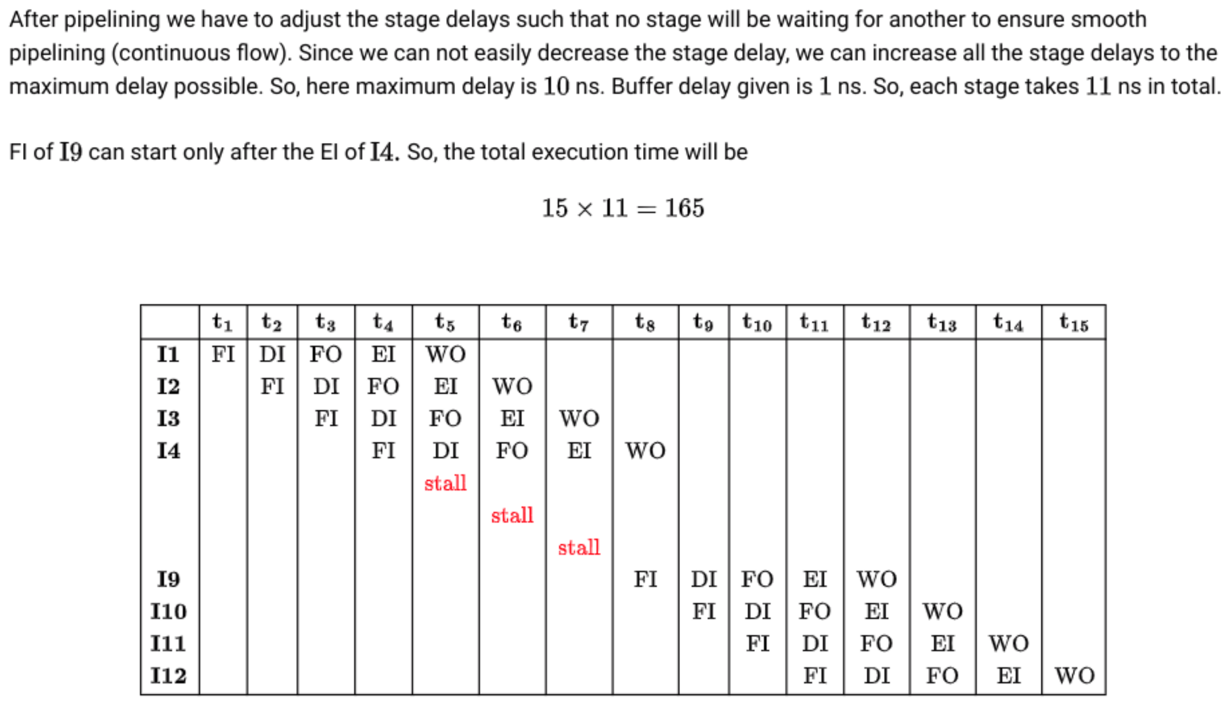
= (100 + 5 -1) + 40*(3-1)+35*(2-1)+25*(1-1)

= (100 + 4) + 40*2+35*1+25*0

= 104 + 115

= 219 cycles

Q3)



After pipelining we have to adjust the stage delays such that no stage will be waiting for another to ensure smooth pipelining (continuous flow). Since we can not easily decrease the stage delay, we can increase all the stage delays to the maximum delay possible. So, here maximum delay is 10 ns. Buffer delay given is 1 ns. So, each stage takes 11 ns in total.

FI of $\mathrm{I9}$ can start only after the EI of $\mathrm{I4}$. So, the total execution time will be

$$15 \times 11 = 165$$

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}
I1	FI	DI	FO	EI	WO										
I2		FI	DI	FO	EI	WO									
I3			FI	DI	FO	EI	WO								
I4				FI	DI	FO	EI	WO							
					stall										
						stall									
							stall								
I9								FI	DI	FO	EI	WO			
I10									FI	DI	FO	EI	WO		
I11										FI	DI	FO	EI	WO	
I12											FI	DI	FO	EI	WO

Q4)

Read After Write :

1. ADD -> MUL (R5)
2. MUL -> SUB (R6)
3. SUB -> DIV (R5)
4. DIV -> STORE (R6)

Write After Read :

1. MUL -> SUB (R5)
2. DIV -> STORE (R6)

Write After Write :

1. ADD -> SUB (R5)
2. MUL – DIV (R6)

Q5)

Execution Time = No.of Instructions x Clocks per Instructions x Clock cylce time

$$Speedup = \frac{\text{Old system execution time}}{\text{New system execution time}}$$

When no.of Instructions and clock cycle time are same, then

$$Speedup = \frac{\text{Old system CPI}}{\text{New system CPI}}$$

Normal processor CPI = 1, without any pipeline hazards.

Given that, Program P has 30% branch instructions where each instruction will lead to 2 stall cycles.

Processor X1 has NO BPU, therefore CPI = $1 + \overbrace{(0.30 * 2)}^{\text{branch instructions penalty}}$ = 1.60

Processor X2 has BPU, therefore CPI = $1 + \overbrace{(0.30 * (\underbrace{0.80 * 0}_{\text{BPU correctly predicted}} + \underbrace{0.2 * 2}_{\text{BPU wrongly predicted}}))}^{\text{branch instructions penalty}}$ = 1.12

$$Speedup = \frac{1.60}{1.12} = 1.4285$$