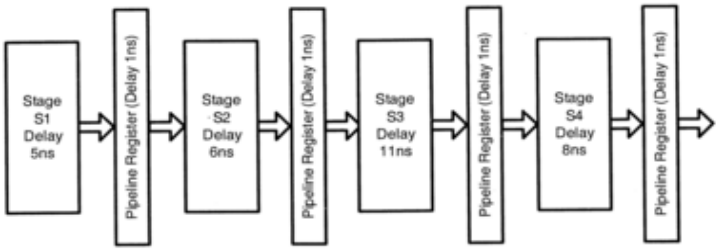
Tutorial Sheet
Computer Organization & Architecture
MIPS Pipelining

Q1. Consider an instruction pipeline with five stages without any branch prediction: Fetch Instruction (FI), Decode Instruction (DI), Fetch Operand (FO), Execute Instruction (EI) and Write Operand (WO). The stage delays for FI, DI, FO, EI and WO are 5 ns, 7 ns, 10 ns, 8 ns and 6 ns, respectively. There are intermediate storage buffers after each stage and the delay of each buffer is 1 ns. A program consisting of 12 instructions I1, I2, I3… I12 is executed in this pipelined processor. Instruction I4 is the only branch instruction and its branch target is I9. If the branch is taken during the execution of this program, the time (in ns) needed to complete the program?

Q2. Consider an instruction pipeline with four stages (S1, S2, S3 and S4) each with combinational circuit only. The pipeline registers are required between each stage and at the end of the last stage. Delays for the stages and for the pipeline registers are as given in the figure.



What is the approximate speed up of the pipeline in steady state under ideal conditions when compared to the corresponding non-pipeline implementation?

Q3. A 5-stage pipelined processor has Instruction Fetch (IF), Instruction Decode (ID), Operand Fetch (OF), Perform Operation (PO) and Write Operand (WO) stages. The IF, ID, OF and WO stages take 1 clock cycle each for any instruction. The PO stage takes 1 clock cycle for ADD and SUB instructions, 3 clock cycles for MUL instruction, and 6 clock cycles for DIV instruction respectively. Operand forwarding is used in the pipeline. What is the number of clock cycles needed to execute the following sequence of instructions?

Instruction	Meaning of instruction
I_0 :MUL R_2 ,R_0 ,R_1	$R_2 \leftarrow R_0 * R_1$
I_1 :DIV R_5 ,R_3 ,R_4	$R_5 \leftarrow R_3 / R_4$
I_2 : ADD R_2 ,R_5 ,R_2	$R_2 \leftarrow R_5 + R_2$
I_3 :SUB R_5 ,R_2 ,R_6	$R_5 \leftarrow R_2 - R_6$

Q4. Consider a pipelined processor with the following four stages:

IF: Instruction Fetch
ID: Instruction Decode and Operand Fetch
EX: Execute
WB: Write Back

The IF, ID and WB stages take one clock cycle each to complete the operation. The number of clock cycles for the EX stage depends on the instruction. The ADD and SUB instructions need 1 clock cycle and the MUL instruction needs 3 clock cycles in the EX stage. Operand forwarding is used in the pipelined processor. What is the number of clock cycles taken to complete the following sequence of instructions?

ADD R2, R1, R0 R2 ← R1 + R0
MUL R4, R3, R2 R4 ← R3 * R2
SUB R6, R5, R4 R6 ← R5 - R4

Q5. Consider the sequence of machine instructions given below:

MUL R5, R0, R1
DIV R6, R2, R3
ADD R7, R5, R6
SUB R8, R7, R4

In the above sequence, R0 to R8 are general purpose registers. In the instructions shown, the first register stores the result of the operation performed on the second and the third registers. This sequence of instructions is to be executed in a pipelined instruction processor with the following 4 stages:

1. Instruction Fetch and Decode (IF),
2. Operand Fetch (OF),
3. Perform Operation (PO) and
4. Write back the Result (WB).

The IF, OF and WB stages take 1 clock cycle each for any instruction. The PO stage takes 1 clock cycle for ADD or SUB instruction, 3 clock cycles for MUL instruction and 5 clock cycles for DIV instruction. The pipelined processor uses operand forwarding from the PO stage to the OF stage. The number of clock cycles taken for the execution of the above sequence of instructions?

Q6: Consider a pipeline with the following stages and latencies (in ps):

I1	I2	EX1	EX2	MEM1	MEM2	MEM3	WB1	Pipeline Register
100	120	150	190	200	250	100	70	10

Consider the following code is run on the processor with the pipeline described above:

```
addi $r4, $zero, 100
loop:
lw $r0, 0($r1)
addi $r1, $r0, 1
sw $r1, 0($r3)
addi $r4, $r4, -1
bne $r4, $zero, loop
```

a. What is the latency of a single instruction in a fully pipelined implementation? In a single cycle implementation?

b. Assuming no forwarding, add NOP instructions to the code to remove any data hazards. The register file is read in the EX1 stage, and instructions must exit from WB for other instructions to use its value.
c. What is the speedup of the fully pipelined implementation over the single cycle implementation? Assume steady-state execution.
d. What are the possible forwarding paths? Assume data is ready to be forwarded at the end of its final stage (EX2, MEM3).
e. Assuming all possible forwarding paths are implemented, what is the new speedup of the fully pipelined implementation over the single cycle implementation? Assume again steady state execution.