MICROPROCESSOR AND COMPUTER ARCHITECTURE

Instruction Pipeline

Prof. Roopa Ravish

Example 1

The instruction pipeline of a RISC processor has the following stages: Instruction Fetch (IF), Instruction Decode (ID), Operand Fetch (OF), Perform Operation (PO) and Write back (WB), The IF, ID, OF and WB stages take 1 clock cycle each for every instruction. Consider a sequence of 100 instructions. In the PO stage, 40 instructions take 3 clock cycles each, 35 instructions take 2 clock cycles each, and the remaining 25 instructions take 1 clock cycle each. Assume that there are no data hazards and no control hazards. The number of clock cycles required for completion of execution of the sequence of instruction is _______ .

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)+20*(1-1)
= (100 + 4) + 40*2+35*1+20*0
= 104 + 115
= 219 cycles

Example 2

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	$R_5 \leftarrow R_2 - R_6$

Ans : 15

Example 3

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
```

Ans: 8

	1	2	3	4	5	6	7	8
I	F	D	E	WB				
I+1		F	D	E	E	E	WB	
I+2			F	D	--	--	E	WB

Example 4

Consider a pipeline having 4 phases with duration 60, 50, 90 and 80 ns. Given latch delay is 10 ns. Calculate-

Pipeline cycle time
Non-pipeline execution time
Speed up ratio
Pipeline time for 1000 tasks
Sequential time for 1000 tasks
Throughput

Solution-

Given-

Four stage pipeline is used

Delay of stages = 60, 50, 90 and 80 ns

Latch delay or delay due to each register = 10 ns

Part-01: Pipeline Cycle Time-

Cycle time

= Maximum delay due to any stage + Delay due to its register

= Max { 60, 50, 90, 80 } + 10 ns

= 90 ns + 10 ns

= 100 ns

Part-02: Non-Pipeline Execution Time-

Non-pipeline execution time for one instruction
= 60 ns + 50 ns + 90 ns + 80 ns
= 280 ns

Part-03: Speed Up Ratio-

Speed up
= Non-pipeline execution time / Pipeline execution time
= 280 ns / Cycle time
= 280 ns / 100 ns
= 2.8

Part-04: Pipeline Time For 1000 Tasks-

Pipeline time for 1000 tasks
= Time taken for 1st task + Time taken for remaining 999 tasks
= 1 x 4 clock cycles + 999 x 1 clock cycle
= 4 x cycle time + 999 x cycle time
= 4 x 100 ns + 999 x 100 ns
= 400 ns + 99900 ns
= 100300 ns

Part-05: Sequential Time For 1000 Tasks-

Non-pipeline time for 1000 tasks
= 1000 x Time taken for one task
= 1000 x 280 ns
= 280000 ns

Part-06: Throughput-

Throughput for pipelined execution
= Number of instructions executed per unit time
= 1000 tasks / 100300 ns

Example 5

```
LW R1, 0(R4) ;            R1 ← address (0+R4)
ADDI R2, R1, #8 ;         R2 ← R1+8
MULT R3, R1, R1 ;         R3 ← R1*R1
SW R3, 4(R2) ;            address(4+R2) ← R3
```

List all RAW (read-after-write) pipeline hazards in the code, regardless of whether they cause any stalls.

Example 6

Consider the following sequence of instructions.

LDR R1, # 0 [R2]
SUB R4, R1, R5
AND R6, R1, R7
OR R8, R1, R9

During the execution of these instructions using a five stage pipeline architecture, **what hazards occur?** Which instructions are affected? How are those hazards resolved?

Explain using diagram that shows various instructions being executed in different clock cycles.

Example 7

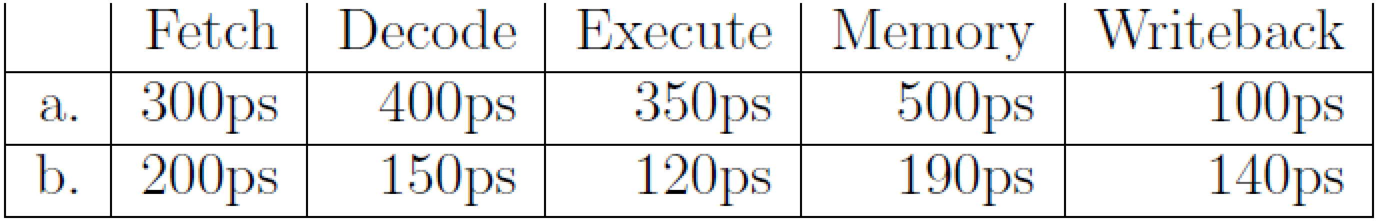
Assume the distribution of instructions that run on the processor is:
 50% ALU
 25%: BEQ
 15%: LW
 10%: SW
Assuming there are no stalls or hazards, what is the utilization of the data memory?
What is the utilization of the register block's write port? (Utilization in percentage of clock cycles used)

Data memory is utilized only by LW and SW instructions in the MIPS ISA. So the utilization is 25% of the clock cycles.

The write port may be utilized by ALU and LW instructions. The utilization is 65% of the clock cycles.

Example 8

The 5 stages of the processor have the following latencies:



	Fetch	Decode	Execute	Memory	Writeback
a.	300ps	400ps	350ps	500ps	100ps
b.	200ps	150ps	120ps	190ps	140ps

Assume that when pipelining, each pipeline stage costs 20ps extra for the registers between pipeline stages.

1. Non-pipelined processor: what is the cycle time? What is the latency of an instruction? What is the throughput?

Cycle-time: there is no pipelining, so the cycle-time has to allow an instruction to go through all the stages each cycle. Therefore:

a. CT = 1650ps

b. CT = 800ps

The latency for an instruction is also the same, since each instruction takes 1 cycle to go from beginning fetch to the end of write back. The throughput similarly is (1/ cycle time) instructions per second.

2. Pipelined processor: What is the cycle time? What is the latency of an instruction? What is the throughput?

Pipelining to 5 stages reduces the cycle time to the length of the longest stage. Additionally, the cycle time needs to be slightly longer to accommodate the register at the end of the stage.

a. CT = 520ps

b. CT = 220ps

The latency for both is 5 (cycle time), since an instruction needs to go through 5 pipeline stages, spending 1 cycle in each, before it commits.

The throughput for both is still 1 instruction/cycle. Throughput increases because the cycle time reduces.

3. If you could split one of the pipeline stages into 2 equal halves, which one would you choose? What is the new cycle time? What is the new latency? What is the new throughput?

Splitting the longest stage is the only way to reduce the cycle time. After splitting it, the new cycle time is based on the new longest stage.

a. Old longest stage is Memory. New CT = 420ps
b. Old longest stage is Fetch. New CT = 210ps

The new latency is 6 * (cycle time), since an instruction needs to go through 6 pipeline stages now.