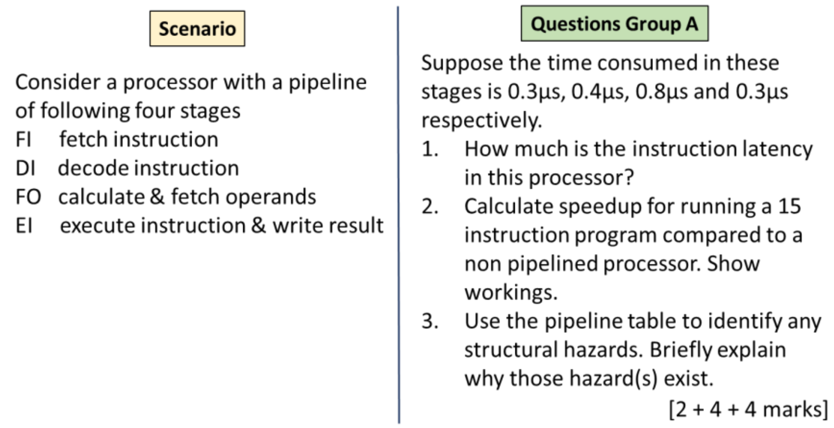


Scenario

Consider a processor with a pipeline of following four stages

FI fetch instruction
DI decode instruction
FO calculate & fetch operands
EI execute instruction & write result

Questions Group A

Suppose the time consumed in these stages is 0.3µs, 0.4µs, 0.8µs and 0.3µs respectively.

1. How much is the instruction latency in this processor?
2. Calculate speedup for running a 15 instruction program compared to a non pipelined processor. Show workings.
3. Use the pipeline table to identify any structural hazards. Briefly explain why those hazard(s) exist.

[2 + 4 + 4 marks]

Q1

Pipeline would wait for the slowest stage (0.8µs) before advancing.
So latency is 0.8 × 4 = 3.2µs.

Q2

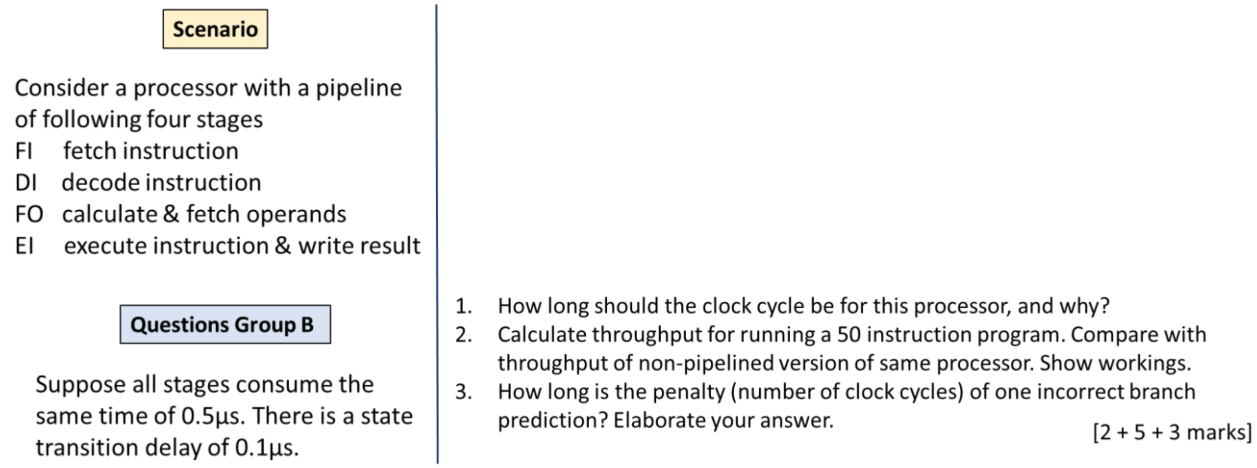
Non-pipelined latency is 0.3+0.4+0.8+0.3=1.8µs
So total program time is 15×1.8 = 27µs

Pipelined total time is (4+(15-1))×0.8 = 14.4µs
Speedup = 27/14.4 = 1.88

Q3

FI	DI	FO	EI	
	FI	DI	FO	WR

EI and FO stages can not run at the same time because EI includes writing result (to registers/memory) and FO will need to read from the same. A resource conflict exists here which causes structural hazard.



Scenario

Consider a processor with a pipeline of following four stages

FI fetch instruction
DI decode instruction
FO calculate & fetch operands
EI execute instruction & write result

Questions Group B

Suppose all stages consume the same time of 0.5µs. There is a state transition delay of 0.1µs.

1. How long should the clock cycle be for this processor, and why?
2. Calculate throughput for running a 50 instruction program. Compare with throughput of non-pipelined version of same processor. Show workings.
3. How long is the penalty (number of clock cycles) of one incorrect branch prediction? Elaborate your answer.

[2 + 5 + 3 marks]

Q1

Pipeline would wait for each stage to complete and outputs transition to next stage before advancing.
So, clock cycle is 0.5+0.1 = 0.6µs.

Q2

Non-pipelined latency is 0.5×4=2µs (no transition delay)
So, throughput = 0.5 instruction per µs

Pipelined total time is (4+(50-1))×0.6 = 31.8µs
Throughput = 50/31.8 = 1.57 instruction per µs

Q3

Incorrect prediction means an instruction already entered the pipeline and completed first three stages (FI, DO, FO). This instruction needs to be flushed from the pipeline and a new instruction is to be brought in.
So branch penalty is number of cycles of wasted work, which is three in this case.
Alternate explanation: Branch penalty is equal to differential between fetch and execute stages.