

Scenario

Consider a processor with a pipeline of following four stages

FI fetch instruction
DO decode instruction and
fetch operands
EI execute instruction
WR write result

Question Group A

Suppose the time consumed in each step is 0.2ms, 0.5ms, 0.1ms and 0.4ms respectively.

1. How much is the instruction latency?
2. Calculate speedup for running a 10 instruction program compared to a non pipelined processor. Show workings.

[3 + 7 marks]

Q1

Pipeline would wait for the slowest stage (0.5ms) before advancing.

So latency is $0.5 \times 4 = 2\text{ms}$.

Q2

Non-pipelined latency is $0.2+0.5+0.1+0.4=1.2\text{ms}$

So total program time is $10 \times 1.2 = 12\text{ms}$

Pipelined total time is $(4+(10-1)) \times 0.5 = 6.5\text{ms}$

Speedup = $12/6.5 = 1.85$

Scenario

Consider a processor with a pipeline of following four stages

FI fetch instruction
DO decode instruction and
fetch operands
EI execute instruction
WR write result

Question Group B

- How much is the penalty of one incorrect branch prediction (in number of clock cycles)?
- Analyze the given program to see how much will be the total branch penalty (in cycles) if the branch prediction strategy is
(i) predict always taken (ii) predict never taken
Show workings: for each conditional branch, give line number and if penalty is applicable (yes/no).

[2 + 4 + 4 marks]

```

1  mov cx, 10
2  mov dx, 5
3  again:
4  cmp dx, 12
5  jg finish
6  add dx, 3
7  sub cx, 1
8  jnz again
9  finish:
10 ; terminate

```

Q1

Incorrect prediction means an instruction already entered the pipeline and completed first two stages (FI and DO). 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 two in this case.
(can also prove it using pipeline table)

Q2

This is a kind of dry run problem with additional branch prediction analysis

(i) Predict always taken

Loop Iteration	Line #	DX	CX	Penalty?
1	5	5	10	Yes (predicted taken but actually not taken)
	8	8	9	No (actual outcome is branch taken, matching the prediction)
2	5	8	9	Yes
	8	11	8	No
3	5	11	8	Yes
	8	14	7	No
4	5	14	7	No (will jump to finish as predicted)

Three incorrect predictions => penalty of 6 cycles.

Note: DX and CX columns are just for convenience in dry run.

(ii) Predict never taken

Loop Iteration	Line #	DX	CX	Penalty?
1	5	5	10	No (did not take branch as predicted)
	8	8	9	Yes (did take branch, incorrect prediction)
2	5	8	9	No
	8	11	8	Yes
3	5	11	8	No
	8	14	7	Yes
4	5	14	7	Yes

Four incorrect predictions => penalty of 8 cycles.