NPTEL Online Certification Courses

Indian Institute of Technology Kharagpur

GPU Architectures and Programming
Assignment- Week 1
TYPE OF QUESTION: MCQ/MSQ

Number of questions: 10 **Total mark: 10 X 1 = 10**

MCQ Question

Common data for Questions 1-4:

Given a 5-stage pipelined processor with stage latencies of 3, 4, 4, 5, and 2 clock cycles for fetch, decode, execute, memory, and write-back respectively, and an additional 1 clock cycle of register latency between stages.

Question 1:
What is the clock cycle time for the pipelined processor?

A. 7 Clock cycle
B. 5 Clock cycle
C. 6 Clock cycle
D. 8 Clock cycle

Answer: C
Solution:
Clock cycle time=Max stage latency+Register latency
=max(3,4,4,5,2) clock cycle+1 clock cycle
=5 clock cycles+1 clock cycle
=6 clock cycles

Question 2:
What is the total number of clock cycles to complete 20 instructions using pipelining?

A. 21 clock cycle
B. 26 clock cycle
C. 30 clock cycle
D. 25 clock cycle

Answer: D
Solution:
It will require 6 clock cycles to fill the pipeline as the clock cycle time for one instruction is 6 as obtained in question 1.
After filling the pipeline, each instruction(n -1=19) completes every clock cycle
So,
Total clock cycle =6 + 19=25

Question 3:
What is the total number of clock cycles to complete 20 instructions without pipelining?

A. 400 clock cycles
B. 440 clock cycles
C. 470 clock cycles
D. 425 clock cycles

Answer: B

Solution:

$T_{non\text{-}pipelined}$= clock cycles for all stages + clock cycle for register latency for all stages except the last stage.

=3+4+4+5+2+4*1

=22 clock cycles per instruction

For 20 instructions:

$T_{non\text{-}pipelined}$=No. of instruction x clock cycle per instruction

=22×20

=440 clock cycles

Question 4:

What is the speedup achieved by pipelining?

A. 17.6
B. 16.5
C. 20.2
D. 15.8

Answer: A

Solution:

Speedup=$T_{non\text{-}pipelined}/T_{pipelined}$

=440/25

=17.6

Common block for Question 5-6:

In computer architecture, a branch predictor is a digital circuit that tries to guess which way a branch (e.g., an if–then–else structure) will go before this is known definitively. The purpose of the branch predictor is to improve the flow in the instruction pipeline and play a critical role in achieving high effective performance. Assume the following outcome of a branch: T, T, T, T, NT, T, T, T, T,NT. [T represents the branch that is taken and NT represents that the branch is not taken].

Question 5:

Consider a 1-bit branch predictor with 2 stages Taken and Not Taken. If the current state is "Taken" and the branch is taken, the state remains at "Taken". Similarly, if the current state is "Not Taken" and the branch is not taken, the state remains at "Not Taken". Conversely, if the current state is "Taken" and the branch is not taken, the state changes to "Not Taken". Similarly, if the current state is "Not Taken" and the branch is taken, the state changes to "Taken". Calculate what is the percentage of misprediction for the given predictor, given that the initial state is Not Taken.

branch predictor stages: NT, T, T, T, T, NT, T, T, T, T

A. 20%
B. 30%
C. 40%
D. 50%

Answer: C

Solution:

Misprediction = 4/10=40%

Question 6:
Consider a 2-bit branch predictor with 4 stages Strongly Taken, Weakly Taken, Weakly Not Taken, and Strongly Not Taken. If the current state is “Strongly Taken” and the branch is taken, the state remains at “Strongly Taken”, and if the current state is “Weakly Taken” and the branch is taken, the state reverts back to “Strongly Taken” . Similarly, if the current state is “Strongly Not Taken” and the branch is not taken, the state remains at “Strongly Not Taken”, and if the current state is “Weakly Not Taken” and the branch is not taken, the state reverts back to “Strongly Not Taken”. Conversely, if the current state is “Strongly Taken” and the branch is not taken, the state changes to “Weakly Taken”, and if the current state is “Weakly Taken” and the branch is not taken, the state changes to “Weakly Not Taken”. Similarly, if the current state is “Strongly Not Taken” and the branch is taken, the state changes to “Weakly Not Taken”, and if the current state is “Weakly Not Taken” and the branch is taken, the state changes to “Weakly Taken”. Calculate what is the percentage of misprediction for the given predictor, given that the initial state is Strongly Not Taken.

branch predictor stages: SNT, WNT, WT, ST, ST, WT, ST, ST, ST, ST, WT

A. 20%
B. 30%
C. 40%
D. 50%

Answer: C
Solution:
Misprediction = 4/10=40%

Question 7:
Assume a computer has a 32 bit address. Each block stores 64 bytes.A direct-mapped cache has 256 blocks. In what block of the cache would the address B108C432 be present.

A. 16
B. 32
C. 28
D. 12

Answer: A
Solution:
32 bit address where,

Tag=18 bits	Index=8 bits	Block offset=6 bits

B108C432 (Hexadecimal) = 1011 0001 0000 1000 11**00 0100 00**11 0010 (Binary)
The index is the 8 bits after the tag bits: 00010000 in binary, which is 16 in decimal.
Thus, the address B108C432 will map to block 16 in the cache.

Question 8:
What hit rate is required for the cache to reduce the effective memory access time from 100ns to 25ns, given that the cache access time is 15ns and cache access and main memory access occur simultaneously?

A. 0.75
B. 0.90

C. 0.88
D. 0.80

Answer: C

Solution:

t_{mm}= 100ns
t_{cm} = 15ns
t_{avg}=25ns
t_{avg}=H*t_{cm}+(1-H)*t_{mm}
t_{avg}=H*15+(1-H)*100
25=15H + 100 -100H
H=75/85=0.88

Question 9:

A CPI of 10 was observed from simulation experiments conducted with a perfect cache for a processor with clock rate 5Ghz. A 64KB L1 cache having a miss rate of 5% and cache miss penalty of 60ns is proposed to be used. What would be the nearest CPI with the 64KB L1 cache?

A. 5
B. 10
C. 15
D. 25

Answer: D

Solution:

5Ghz=0.2 ns
60ns=300 cycles
CPI = observed CPI + penalty for cache miss = 10 + (0.05) × 300 = 25

Question 10:

Given the following assembly code,which instruction will cause RAW(Read after Write) Hazard.

```
1. ADD R1, R2, R3      // R1 = R2 + R3
2. SUB R4, R1, R5      // R4 = R1 - R5
3. MUL R6, R7, R8      // R6 = R7 * R8
4. DIV R12, R9, R13    // R12 = R1 / R13
```

A. SUB R4, R1, R5
B. MUL R6, R7, R8
C. DIV R12, R9, R13
D. ADD R9, R10, R11

Answer: A

Solution:

Instruction 2 (SUB R4, R1, R5) will cause Read-After-Write (RAW) Hazard as R1 is written in instruction 1 and read in instruction 2

*********END**********