Prof. Cristina Silvano	ACA2022_EXAM_FORM1 Data: 08/09/2022

ACA2022_FORM1_8Sept2022_SILVANO

ACA Course -- Prof. SILVANO

FORM1 is composed of 14 QUESTIONS to get UP TO 21 POINTS

Duration is 35 minutes!

IMPORTANT: What to do before leaving MS FORM1:

1. **Please click on:** “Inviami una conferma tramite posta elettronica delle risposte” (Send me a confirmation by email of the answers). This is **fundamental** to get an email with subject: **“My responses".** Open the email and click on the link to **VIEW YOUR RESPONSES.**

2. **Please click on SUBMIT** (Click on **INVIA**) and you will get the message: **“Your answer has been sent”.** Points will be assigned **manually** to **FORM1**. Therefore, the view of your results is **not** immediate: it will be enabled by prof. Silvano **after** results publication phase.

Question 1 (format Multiple Choice – Single answer)

Let's consider the following LOOP to be executed for 100 iterations on a processor with 1-entry 2-bit Branch History Table initialized at the state Weakly Taken:

```
LOOP: LD $F0, 0 ($R1)
      ADDI.D $F2, $F0, $F0
      SD $F2, 0 ($R1)
      ADDI.U $R1, $R1, 4
      BNE $R1, $R2, LOOP
```

How many branch mispredictions will occur by the BNE instruction?

(SINGLE ANSWER)
2 points

Answer 1: 1% **(TRUE)**
Answer 2: 2%
Answer 3: 99 %
Answer 4: 100 %
Answer 5: 98 %

Feedback
Given the initial state Weakly Taken, the 2-bit predictor fails only on the last branch therefore there are 99% correct predictions and 1% mispredictions.

Question 2 (format Multiple Choice – Single answer)

Let's consider the following LOOP to be executed for 100 iterations on a processor with 1-entry 2-bit Branch History Table initialized in the state Strongly Not-Taken:

```
LOOP: LD $F0, 0 ($R1)
      ADDI.D $F2, $F0, $F0
      SD $F2, 0 ($R1)
      ADDI.U $R1, $R1, 4
      BNE $R1, $R2, LOOP
```

How many branch mispredictions will occur by the BNE instruction?

(SINGLE ANSWER)
2 points

Answer 1: 3% **(TRUE)**
Answer 2: 2%
Answer 3: 99 %
Answer 4: 100%
Answer 5: 98 %

Feedback

Given the initial state Strongly Not-Taken, at the beginning the 2-bit predictor fails twice before changing the prediction to Taken and it also fails on the last branch. Therefore, there are 97% correct predictions and 3% mispredictions.

Question 3 (format True/False)

Write-through policy for caches uses a ReOrder Buffer to continue to service read requests while the writes are served towards the main memory.

(True/False)
1 point

Answer 1: **False**

Feedback
*Write through policy for caches always uses a **Write Buffer** to continue to service read requests while the writes are served towards the main memory.*

Question 4 (format Multiple Choice – Single answer)

Let’s consider an application kernel characterized by multiple consecutive writes to the same variables. Which *write protocol* should be better to use for the management of the memory hierarchy?

(SINGLE ANSWER)
1 point

Answer 1: Write-through
Answer 2: Write-back **(TRUE)**
Answer 3: Indifferent

Feedback:
With the write-though protocol, every write is done to memory to update with all the consecutive writes done to the same variables. The write-back protocol will write to memory

the value of the last write only when there is a cache miss: This is the best choice for the given kernel to avoid useless bus traffic due to multiple memory accesses.

Question 5 (Format Multiple Choice – Single answer)

What is the main purpose of a Branch History Table?

(SINGLE ANSWER)
1 point

Answer 1: To buffer the predicted Branch Target Address.
Answer 2: To buffer the predicted Branch Outcome before it is definitively known based on the previous branch behavior. **(TRUE)**
Answer 3: To buffer the correlation of the past behavior of another branch with respect to the given branch.

Question 6 (format Multiple Choice – Single answer)

Let us assume that, given an optimization technique, a fraction of instructions of an application code is parallelized to obtain, a CPI = 2.5 while the remaining fraction of instructions is executed sequentially with the original CPI = 5.
How much is the fraction of parallelized instructions to get an overall Speedup equal to 10?

(SINGLE ANSWER)
2 points

Answer 1: 2
Answer 2: 1.8 **(TRUE)**
Answer 3: 0.5
Answer 4: 0.2

Feedback:
To calculate the overall speedup we need to apply the Amdahl's law as follows:
$\mathit{Fraction}_E = x;$
$\mathit{Speedup}_E = \mathit{CPI}_o / \mathit{CPI}_E = 5 / 2.5 = 2;$
$\mathit{Speedup} = 1 / [(1-x) + (x / \mathit{Speedup}_E)] \Rightarrow 10 = 1 / [(1-x) + (x / 2)] \Rightarrow x = 9 / 5 = 1.8$

Question 7 (format Multiple Choice – Single answer)

Let’s consider a directory-based protocol for a distributed shared memory system with 4 Nodes (N0, N1, N2, N3) where we consider the block B2 in the directory of N2:

Directory N2 Block B2 | State: Shared | Sharer Bits: 1001

Which is the request sent for the block B2 to become:

Directory N2 Block B2 | State: Modified | Sharer Bits: 0100

(SINGLE ANSWER)
1 point

Answer 1: Write Hit B2 sent from N1 to N1
Answer 2: Write Miss B2 from N1 to N2 **(TRUE)**
Answer 3: Read Hit B2 from N1 to N2
Answer 4: Read Miss B2 from N1 to N2

Question 8 (format Multiple answers)

What are the true statements in a shared memory multiprocessor?

(MULTIPLE ANSWERS)
2 points

Answer 1: The access time to a shared memory variable could be either uniform or not uniform for all the processors. **TRUE**

Answer 2: Multiple processors communicate by explicit send/receive messages because they do not have direct access to the shared memory variables.

Answer 3: Multiple copies of shared memory variables generate cache coherence problems among the multiple processors. **TRUE**

Answer 4: The physical memory can be either centralized on a single node or distributed over the nodes. **TRUE**

Question 9 (format Multiple Choice – Multiple answers)

Let's consider the following loop code:

```
for (i=1; i<=100, i++) {
      X[i] = X[i-1] + Z[i-1];              /*S1*/
      Y[i] = Y[i-1] + W[i-1];              /*S2*/
      W[i] = W[i-1];                      /*S3*/
}
```

What are the loop-carried dependences in the code?

(MULTIPLE ANSWERS)

2 points

Answer 1: One in S1 because X[i] depends on X[i-1]; (**TRUE**)

Answer 2: One in S1 because X[i] depends on Z[i-1];

Answer 3: One in S2 because Y[i] depends on Y[i-1]; (**TRUE**)

Answer 4: One in S2 because Y[i] depends on W[i-1]; (**TRUE**)

Answer 5: One in S3 because W[i] depends on W[i-1]; (**TRUE**)

Feedback
Please note that the dependences of X[i] on Z[i-1] is not loop-carried dependences because the vector Z[] are never modified in the loop code.

Answer 1: True

Question 10 (Format Multiple Choice – Single answer)

Consider the following assembly code:

```
            ADDI $R1, $R0, 0
            ADDI $R2, $R0, 100

LOOP:       BEQ $R1, $R2, DONE
            ADD $R5, $R5, $R4
            ADDI $R1, $R1, 4
            J   LOOP
```

Let's assume *that* ***only*** *the BEQ instruction accesses the 1 entry 1-bit Branch History Table. How many loop iterations? How many accesses to the BHT are done?*

(SINGLE ANSWER)
2 points

Answer 1: 25 iterations, 26 BHT accesses **(TRUE)**
Answer 2: 100 iterations, 101 BHT accesses
Answer 3: 100 iterations, 100 BHT accesses
Answer 4: 25 iterations, 25 BHT accesses

Feedback:
There are 25 loop iterations.
For each iteration, the BEQ instruction accesses the BHT at each iteration plus one additional time at the end of the loop for a total of 26 BHT accesses.

Question 11 (format True/False)
In the Speculative Tomasulo architecture, the Reorder Buffer is combined to a Rename Table to solve WAW/WAR hazards

(True/False)
1 point

Answer 1: **True**

Question 12 (format Multiple Choice – Single answer)
*To obtain **a loop unrolling** version of a code, which technique we need to apply to solve output dependences (WAW) and anti-dependences (WAR)?*

(SINGLE ANSWER)
1 point

Answer 1: Register Renaming **True**
Answer 2: List-based Scheduling
Answer 3: Memory Aliasing
Answer 4: Trace Scheduling

Question 13 (format True/False)
Multithreading could be obtained even on a single-core processor architecture making the processor able to manage multiple threads concurrently: if a thread gets stalled, the processor can execute another thread by keeping the functional units busy. **True**

(True/False)
1 point

Answer 1: **True**

Question 14 (format True/False)
What are the true sentences related to ***multithreading:***

Answer 1: Each thread must preserve its computational state by a private PC and a set of private registers that are separated from the other threads. **True**

Answer 2: Thread switch must be managed by hardware support to get more efficient than process switch supported by the OS. **True**

Answer 3: Simultaneous multithreading is convenient for multiple-issue CPUs because there is a number of functional units that cannot be kept busy with instructions from a single thread. **True**

Answer 4: Multithreading requires a data-parallel programming model

(MULTIPLE ANSWERS)

2 points