## **Computer Sciences Department**

## **University of Wisconsin-Madison**

## **CS/ECE 552 – Introduction to Computer Architecture**

## **In-Class Exercise (04/23)**

**Answers to all questions should be uploaded on Canvas.**

1. [2 points] From the textbook:
   1. [1 point] (6.10) So far in class we have discussed the 3C’s for classifying cache misses (cold/compulsory, capacity, conflict). However, in multiprocessor systems there is a 4th ‘C’ – coherence misses. Name at least one scenario where a coherence miss occurs.

Different/multiple cores try to access the same data. They each keep a copy in its own cache but that copy does match with each other.

* 1. [1 point] (6.10) What mechanism determines when a write will be seen by read from a different core?

Invalidat3e-based protocols

2. [8 points] Consider a program with four threads running in parallel on a quad-core multiprocessor. Assume that all memory contents are **initially 0**:

**# thread A**

**addi $t0, $zero, 4096**

**addi $t1, $zero, 1**

**sw $t1, 0($t0)**

**# thread B**

**addi $t0, $zero, 4096**

**addi $t2, $zero, 1**

**sw $t2, 4($t0)**

**# thread C**

**addi $t0, $zero, 4096**

**lw $s0, 0($t0)**

**lw $s1, 4($t0)**

**# thread D**

**addi $t0, $zero, 4096**

**lw $s2, 4($t0)**

**lw $s3, 0($t0)**

Fill in the table below, specifying whether or not each program outcome (i.e., set of register contents after executing all threads) is **possible** if the multiprocessor enforces **sequential consistency**.

|  |  |  |
| --- | --- | --- |
| **Register Contents After Execution** | | **Possible With Sequential Consistency (Y/N)?** |
| **Thread C** | **Thread D** |
| [$s0] = 0  [$s1] = 0 | [$s2] = 0  [$s3] = 0 | Y |
| [$s0] = 0  [$s1] = 0 | [$s2] = 0  [$s3] = 1 | Y |
| [$s0] = 0  [$s1] = 1 | [$s2] = 0  [$s3] = 1 | Y |
| [$s0] = 1  [$s1] = 0 | [$s2] = 0  [$s3] = 0 | Y |
| [$s0] = 1  [$s1] = 0 | [$s2] = 1  [$s3] = 0 | N |
| [$s0] = 1  [$s1] = 1 | [$s2] = 0  [$s3] = 0 | Y |
| [$s0] = 1  [$s1] = 1 | [$s2] = 0  [$s3] = 1 | Y |
| [$s0] = 1  [$s1] = 1 | [$s2] = 1  [$s3] = 0 | Y |
| [$s0] = 1  [$s1] = 1 | [$s2] = 1  [$s3] = 1 | Y |