



Princeton Computer Science Contest – Fall 2024

Problem 6: Sequential Consistency (15 points) [File Upload]

By August Ning

Problem Statement

The following three programs are running concurrently and read from/write to two shared memory addresses `mem0` and `mem1`. Find all possible unique outcomes of `mem0` and `mem1` after all three programs have finished completion assuming the programs are ran in a *sequentially consistent order*. `mem0` and `mem1` are both initialized to 0.

P1	P2	P3
p1.0: LOAD r1, mem0	p2.0: LOAD r2, mem0	p3.0: LOAD r3, mem1
p1.1: ADDI r1, 3	p2.1: ADDI r2, 5	p3.1: ADDI r3, 7
p1.2: STORE r1, mem0	p2.2: STORE r2, mem0	p3.2: STORE r3, mem1
p1.3: ADDI r1, 3		
p1.4: STORE r1, mem1		

Background: Sequential Consistency

Sequential consistency is a model in concurrent computing where the result of any execution appears as if all operations were executed in some sequential order, while maintaining the order specified by each individual process's program. It can be thought of as a guarantee that shared memory behaves like a single, global memory accessed by a switch that connects to one process at a time. Similarly, think of the programs as running on a single machine and the programs can be swapped in/out at any time, in any order after an instruction completes. Each individual program will run concurrently and independently, but instructions within a program may not be reordered.

For example, $\{p2.0, p1.0, p1.1, p3.0\}$ is a valid sequential ordering. $\{p1.0, p1.2\}$ and $\{p1.1, p1.0\}$ are **not valid** since instructions are skipped/reordered within a program.

Princeton Computer Science Contest – Fall 2024



CITADEL | CITADEL Securities





Princeton Computer Science Contest – Fall 2024

Background: The Programs' Pseudo-Assembly

For this problem, the programs are executing a psuedo-assembly with three types of instructions.

- **LOAD rX, memY**: load the value in memY into register rX
- **STORE rX, memY**: store the value in register rX into memY
- **ADDI rX, [INT]**: $rX = rX + [INT]$

Registers r1, r2, r3 are local variables per program.

Example

A valid sequentially consistent ordering is to run all of P1, then all of P2, then all of P3.

P1	P2	P3	mem0	mem1
LOAD r1, mem0			0	0
ADDI r1, 3			0	0
STORE r1, mem0			3	0
ADDI r1, 3			3	0
STORE r1, mem1			3	6
	LOAD r2, mem0		3	6
	ADDI r2, 5		3	6
	STORE r2, mem0		8	6
		LOAD r3, mem1	8	6
		ADDI r3, 7	8	6
		STORE r3, mem1	8	13

Princeton Computer Science Contest – Fall 2024



CITADEL | CITADEL Securities





Princeton Computer Science Contest – Fall 2024

How to Submit: For each possible unique $\{\text{mem0}, \text{mem1}\}$ outcome, provide a sequentially consistent ordering as well as the memory state after each instruction is executed, similar to how it is shown in the example.

To make it easier to grade, please sort your outcomes in ascending order. *e.g.*, if you think the $\{\text{mem0}, \text{mem1}\}$ outcomes are $(1, 2), (1, 3), (2, 5), (3, 1)$, please submit your outcomes and corresponding tables as shown.

Collect all of the outcome tables into an a single PDF and upload the PDF to the Google form.

Tips: Excel is your friend!

Princeton Computer Science Contest – Fall 2024



Jane
Street®

SCM STEVENS
CAPITAL
MANAGEMENT



Evergreen Statistical Trading

CITADEL | CITADEL Securities

SIG
SUSQUEHANNA
INTERNATIONAL GROUP, LLP



PRINCETON
Computer Science

CENTER FOR STATISTICS AND MACHINE LEARNING

thetadata

Princeton
Physics

PI CSciE

ORFE



Department of
Mathematics