Design of Parallel and High Performance Computing
HS 2017
Markus Püschel, Torsten Hoefler
Department of Computer Science
ETH Zurich

Homework 3

Sequential Consistency

Exercise 1

For the following executions, please indicate if they are sequentially consistent. All variables are initially set to 0.

1. P1: W(x,1);
   P2: R(x,0); R(x,1)
2. P1: W(x,1);
   P2: R(x,1); R(x,0);
3. P1: W(x,1);
   P2: W(x,2);
   P3: R(x,1); R(x,2);
4. P1: W(x,1);
   P2: W(x,2);
   P3: R(x,2); R(x,1);
5. P1: W(x,1);
   P2: W(x,2);
   P3: R(x,2); R(x,1);
   P4: R(x,1); R(x,2);
6. P1: W(x,1); R(x,1); R(y,0);
   P2: W(y,1); R(y,1); R(x,1);
   P3: R(x,1); R(y,0);
   P4: R(y,0); R(x,0);
7. P1: W(x,1); R(x,1); R(y,0);
   P2: W(y,1); R(y,1); R(x,1);
   P3: R(y,1); R(x,0);

Exercise 2

P1	**P2**	**P3**
x = 1;	y = 1;	z = 1;
print(y,z);	print(x,z);	print(x,y);

All variables are stored in a memory system which offers sequential consistency. All operations, even the print statements, are atomic. Atomicity means, that no operation can overlap with other operations. Are the following sequences legal outputs?

1. 001011
2. 001111
3. 001110

Explain your answer by showing one possible interleaving of the instructions that might lead to the legal output. In case of an illegal output, explain why no possible interleaving exists.

Design of Parallel and High Performance Computing
HS 2017
Markus Püschel, Torsten Hoefler
Department of Computer Science
ETH Zurich

Homework 3

Non-blocking caches

Consider a system with three processors. Each processor has a non-blocking cache (i.e., a processor does not stop upon encountering a cache miss; while the missing cache line is brought in from memory, the processor continues to execute instructions that are not data or control dependent upon a missing load).

The memory locations x and y are originally 0. The processors execute the following sequence:

P1: W(x,1); R(y,0);
P2: W(y,1); R(y,1); R(x,1);
P3: R(y,1); R(x,0);

Is this system sequentially consistent?

The x86 Memory Model: TLO+CC

1. Describe the memory model of the x86 architecture. Where does it differ from sequential consistency.
2. Two threads execute the following code (given in AT&T assembly syntax) on a machine using TLO+CC. Is it possible that the registers eax $= 0 \wedge$ ebx $= 0$ after both threads have finished?. Would it be possible in sequential consistency?

Initial: All registers and memory locations are 0	
Thread A	Thread B
movl $1, 0x42 movl 0x50, %eax	movl $1, 0x50 movl 0x42, %ebx