

Question 1

What is Lamport's optimization for loads that are enqueued after a store to the same location?

- 1.No such optimization is proposed.
- 2.The load is reordered to before the store for a faster response time.
- 3.The load returns the value of the store but remains in the queue.
- 4.The load immediately returns the value from the store and is not enqueued.

Question 2

Lamport writes "We need only require that all requests *to the same memory cell* be serviced in the order that they appear in the queue." Which of the following statements would be true if we defined a memory cell as a single bit?

- 1.It would be impossible to obtain mutual exclusion for any number of threads. E.g., Dekker's algorithm would no longer work.
- 2.Programming would be more difficult because if two threads write two different values V_0 and V_1 to the same byte B in memory, the resulting value for B could be a mixture of V_0 and V_1 .
- 3.Sequential consistency would no longer hold.
- 4.There would be increased memory parallelism as memory requests to distinct bits could complete in any order.

Question 4

Under TSO, write buffers must service requests in FIFO order. The FIFO order must be consistent with which one of the following orders?

1. program order
2. no order in particular
3. address order

Question 5

When would you need to use a FENCE instruction on a SC processor?

1. never
2. between loads and stores to different addresses
3. when implementing Dekker's algorithm for 3 threads
4. between loads and stores to the same address

Question 6

What is the main advantage of TSO over SC?

1. performance
2. clarity of specification
3. ease of understanding
4. fewer letters in acronym

Question 3

Just something to think about for next class: Lamport's scheme allows memory requests to different memory cells to be handled in an unordered fashion. But on the processor side, all memory requests must occur sequentially in program order (due to Requirement R1).

Can we extend Lamport's scheme to perform reordering on the processor side? That is, if we have two requests A and B and we know that A and B access distinct memory cells, can we safely reorder the accesses? Are there only certain conditions under which reordering is ok, or can we always reorder?



What is Total Store Order?

operational

formal

TABLE 4.4: TSO Ordering Rules. An “X” Denotes an Enforced Ordering. A “B” Denotes that Bypassing is Required if the Operations are to the Same Address. Entries that are Different from the SC Ordering Rules are Shaded and Shown in Bold.

		Operation 2			
		Load	Store	RMW	FENCE
Operation 1	Load	X	X	X	X
	Store	B	X	X	X
	RMW	X	X	X	X
	FENCE	X	X	X	X

SC sample execution

`x == 0 && y == 0`

`1 => x;`
`r1 <= y;`

`1 => y;`
`r2 <= x;`

SC sample execution

`x == 0 && y == 0`

`1 => x;`
`r1 <= y;`

`1 => y;`
`r2 <= x;`

can `r1 == 0 && r2 == 0`?

Dekker's algorithm

```
flag[0] = false
flag[1] = false
turn    = 0
```

```
flag[0] = true;
while (flag[1] == true) {
    if (turn != 0) {
        flag[0] = false;

        while (turn != 0) {}

        flag[0] = true;
    }
}
```

```
// critical section
```

```
turn    = 1;
flag[0] = false;
```

Flag-based synchronization

```
boolean ready = false;
```

```
// wait for condition  
while (!ready) {}
```

```
// go for it...
```

Flag-based synchronization

```
boolean ready = false;
```

```
// wait for condition  
while (!ready) {}
```

```
// go for it...
```

```
// initialize...
```

```
ready = true;
```