CHALMERS TEKNISKA HÖGSKOLA
Institutionen för data- och informationsteknik
Avdelning för datorteknik

Exam in EDA284 (Chalmers) and DIT361 (GU) Parallel Computer Architecture, Monday, October 29th, 2018, 8:30h-12:30h

---

Teacher/Lärare: Miquel Pericàs, tel 7721705

Language/Språk: Answers shall be given in English.

Solutions/Lösningar: Solutions will be posted on Wednesday, Oct 31st, on the course pingpong page.

Exam review/Granskning: The review date will be posted on the course pingpong page by the time you receive the email from LADOK.

---

**Grades:**

| **Chalmers** | | | | |
|---|---|---|---|---|
| **Points** | 0-23 | 24-35 | 36-47 | 48-60 |
| **Grade** | Failed | 3 | 4 | 5 |

| **GU** | | | | |
|---|---|---|---|---|
| **Points** | 0-23 | 24-41 | 42-60 | |
| **Grade** | Failed | G | VG | |

**Good Luck!**

**Problem 1 (12 points)**

The goal of this exercise is to reason about three locking schemes designed to protect a shared data structures in three different scenarios. The underlying parallel computer is a sequentially consistent shared memory bus-based multiprocessor that uses the MSI-invalidate protocol to manage coherence across private caches.

The three scenarios under consideration are:

- Scenario #1: the shared data structure is accessed by a single thread.
- Scenario #2: the lock is accessed by multiple threads but lock contention is very low.
- Scenario #3: the lock is accessed by many threads, and lock contention is high.

The following codes show the implementation of the three locking schemes. In the following assume that R0=0, and that the lock is taken when its value is 1. The variable holding the lock is `_lock`.

| Action/Label | **Lock #1** | **Lock #2** | **Lock #3** |
|---|---|---|---|
| `Lock:` | `T&S R1, _lock`<br>`BNEZ R1, Lock` | `LW R1, _lock`<br>`BNEZ R1, Lock`<br>`T&S R1, _lock`<br>`BNEZ R1, Lock` | `ADDI R1,R0,1`<br>`LL R2,_lock`<br>`SC R1,_lock`<br>`BEQZ R1, Lock`<br>`BNEZ R2, Lock` |
| `Unlock:` | `SW R0, _lock` | `SW R0, _lock` | `SW R0, _lock` |

The task is to describe, for each of the three locking schemes:

(a) How well do the locking schemes apply to the three scenarios shown above?

(b) What synchronization hardware needs to be added to the basic bus-based multiprocessor (as described in Lecture 4) to support the locking schemes?

(c) Assume now that the memory consistency model is a synchronization-based memory consistency model such as weak ordering. Will the locking schemes still work correctly?

**Problem 2 (6p)**

Amdahl's law and Gustafson's law are two speed-up models used to assess the expected impact of an architectural improvement, and to guide the efforts to improve the performance of parallel computer system.

(a) Describe Amdahl's law and provide the mathematical formulation to determine the speed-up as a function of the parallelizable program fraction and the number of processors used to run the program.

(b) Describe Gustafson's law conceptually and discuss what motivated its development

**Problem 3 (6p)**

In this problem you are asked to describe basic metrics of interconnection networks and to analyze the performance of some basic network topologies.

(a) Describe in a few sentences the following latency and bandwidth measures used for interconnection networks: (a) routing distance, (b) network diameter, (c) bandwidth per node, and (d) bisection bandwidth.

(b) Consider the following network topologies, all connecting $N$ nodes: a bus, a crossbar switch, a bidirectional ring, and a $n$-by-$n$ 2D mesh such that $N=n*n$. The task is to determine the network diameter and bisection width of these four network topologies, as a function of $N$ (or $n$).

**Problem 4 (6p)**

To effectively use multicores, simplifying the development of parallel software while providing high performance is an important goal. A recent proposal to achieve these goals is Hardware-based Transactional Memory (TM). The task is to compare HW Transactional Memory with traditional single-global locks and fine-grained lock-based programming in terms of:

(a) Programming complexity
(b) Performance
(c) Hardware support

You should write <u>about 3-5 sentences</u> for each item

**Problem 5 (10p)**

A design team needs to choose an appropriate cache coherence protocol to be used for a shared memory multiprocessor with a number of processor/private cache units connected by a shared single-transaction bus. The team has narrowed down the design to two protocol choices: MSI-invalidate and MESI-invalidate. Given the higher complexity of MESI, the designers consider that at least 5% better performance and 5% lower traffic need to be achieved to justify the implementation of the MESI protocol. In order to select the protocol, the team is analyzing the following representative sequence of accesses happening on the bus in the following order:

R1/X, R3/Z, R2/Y, W1/X, W2/Y, W3/Z, R1/Z, R4/Z, W2/Z, R3/Z

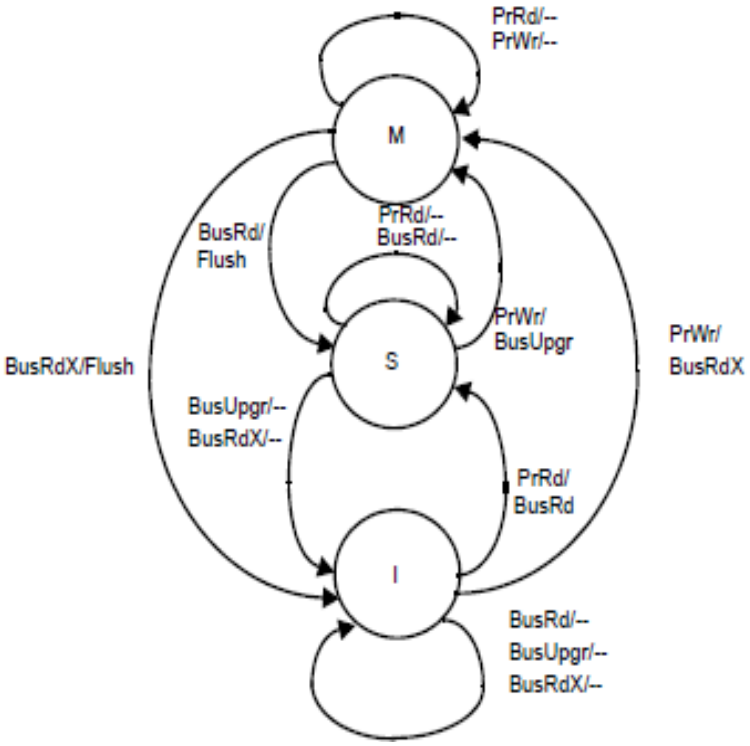
where the notation Ri/X, Wi/X means a read and write from processor i to cache block X. The latency and traffic costs of different operations considering a block size B bytes are:

| **Operation** | **Latency** | **Bus Traffic** |
|---|---:|---:|
| Read hit | 1 | N/A |
| Write hit | 1 | N/A |
| Request serviced by next level (BusRd,Flush) | 50 | 6+B |
| Bus upgrade (BusUpgr) | 10 | 10 |
| Snoop action | 5 | N/A |

Assumptions:

1. A bus upgrade consists of transferring the request on the bus and making a snoop action in each cache (the latency of the latter is shown in the table)
2. There is no contention on the tag directory (duplicated tag directory)
3. X, Y and Z are not the same cache block
4. The block size B is 32 bytes
5. Read-exclusive requests (BusRdX) have the same latency and traffic as Read requests

The MSI state-transition diagram is shown below

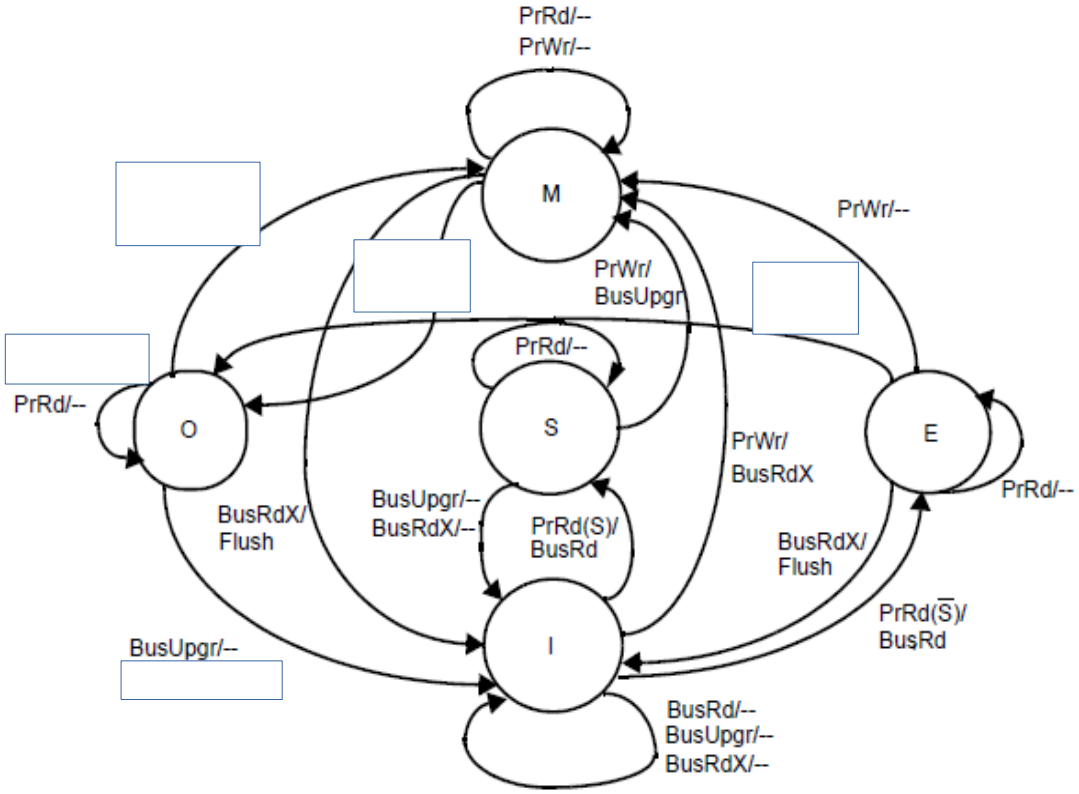



To assess the performance of both protocols the team must construct a table with the latency (cycles) and bus traffic (bytes) needed to complete the aforementioned access sequence (shown below). Considering the complexity-performance trade-off (5%), which protocol should the design team ultimately select?

| Protocol | Latency | Traffic |
|---|---|---|
| MSI | | |
| MESI | | |
| Improvement | % | % |

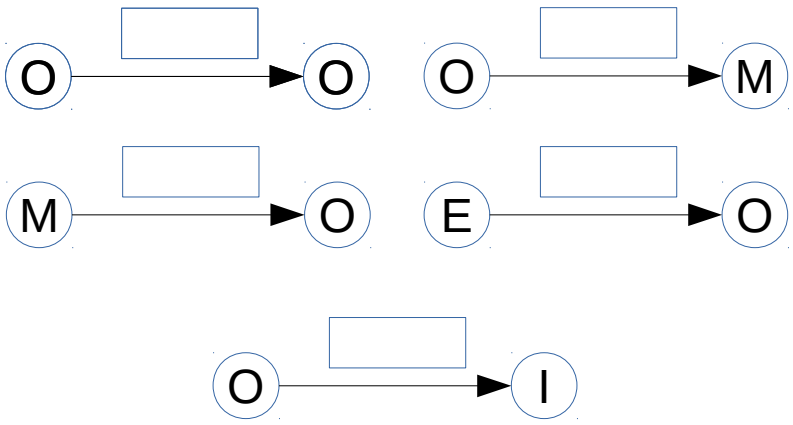
**Problem 6 (6p)**

The MOESI protocol adds a new state called Owner (O). A designer has been given the task to implement the MOESI protocol by adding the necessary transitions on top of MESI. The designer has so far enumerated all the transitions and added them to the MESI diagram (as shown below), but not all coherence protocol actions have yet been determined.




Your task is to:

(a) Describe the meaning of the state (O)wner, and show one example in which it improves performance compared to MSI or MESI.
(b) Add the missing protocol actions to the five transitions shown below



**Problem 7 (8p)**

Consider a future 8-way CMP on which you can power off some cores to allow the rest to operate at a higher frequency. You can use either 1, 2, 4, or 8 cores at the respective frequencies:

- when only one core is running it operates at 0.3ns clock cycle
- when two cores are running they operate at 0.4ns clock cycle
- when 4 cores are running they operate at 0.6ns clock cycle,
- when all 8 cores are running they operate at 1ns clock cycle,

Consider a partially parallel application consisting of a serial part of 100 Instructions and a parallel part that can be parallelized at will which is 200 Instructions. Parallelizing however requires you to create threads in the serial part of the application and 10 Instructions are added for every thread you create. The serial and parallel part of the application all run one instruction per cycle regardless of the frequency.

(a) Calculate the runtime of the application for the above processor when using 1, 2, 4 or 8 cores at their respective frequency without reconfiguring the processor while the application is running. Which configuration is better?

(b) What if you could reconfigure the processor to run the serial part with one core at 0.3 ns clock cycle and then configure it to 2 or 4 or 8 cores for the parallel part? What is the runtime for each case? Consider for all cases that creating each thread takes 10 Instructions that are added to the serial part (and run on the single core at 0.3ns clock cycle).

**Problem 8 (6p)**

We consider a scalable implementation of a shared memory multiprocessor using a set of nodes such that each contains a processor, a private cache, and a portion of the memory. Cache coherence is maintained using a directory cache protocol. The following costs apply:

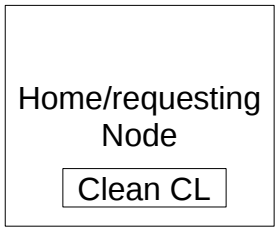
- The block size is 64 bytes.
- The time it takes to process a directory request is 20 cycles.
- The time it takes to install a new entry in a directory is also 20 cycles.
- A remote read request takes 10 Cycles and induces 8 bytes of traffic
- A flush of a cache line from one node to another costs 40 Cycles and 8 + 64 Bytes
- Ignore the time it takes to read/write to a cache.

Also:

- The home node is the one that holds the directory entry for a cache line.
- The requesting node is the node that produces a cache miss for a cache line.
- The remote node is the one that holds the dirty copy of a cache line when the cache line is dirty.

Your task is to describe the sequence of steps, as well as the number of cycles and traffic, needed to handle a cache miss for the following scenarios:

(a) The home node is the same as the requesting node and the memory copy is clean.




(b) The home node is the same as the requesting node and the memory copy is dirty


(c) The home node is different from the requesting node and the memory copy is clean


(d) The home node is different from the requesting node, the memory copy is dirty and the remote node is is the same as the home node


(e) The home node is different from the requesting node, the memory copy is dirty and the remote node is different from the home node