# CIS655/CSE661: Advanced Computer Architecture

# Mock Exam. Spring 2017

1. *First, write down your name and SUID on the top of all pages.*
2. *Check the number of pages; if it is smaller than 14 pages, report.*
3. *Close the exam and wait until the exam starts.*

**Problem 1. [20 points] Cache Coherence**

In a shared memory, a snooping protocol is implemented among four processors (P0, P1, P2, P3). Consider just one memory block in this problem. Initially, the block is of value A.

**Main Memory**

|  |
| --- |
| A |

**P0**

|  |
| --- |
|  |

**P1**

|  |
| --- |
|  |

**P2**

**P3**

|  |
| --- |
|  |

|  |
| --- |
|  |

1. [10 points] Suppose the **MSI** protocol is being used, and the instruction sequence to access block is illustrated in the following table ordered in time. Fill the rest of table with the value and state (M, S or I) of the cached copy of block *right after* the instruction is being executed.

Time-line

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Time | Instruction | CPU P0 | P1 | P2 | P3 |
| t0 | P0: read | S/A (means: state ‘S’ and value ‘A’) | NA (Not Applicable) | NA | NA |
| t1 | P3: write A | I/A | NA | NA | M/A’ |
| t2 | P1: write A | I/A | M/A’’ | NA | I/A’ |
| t3 | P2: read | I/A | S/A’’ | S/A’’ | I/A’ |

1. [3 points] Suppose the **MESI** protocol is being used, and the instruction sequence to access block is illustrated in the following table ordered in time. Fill the rest of table with the value and state (M, E, S or I) of the cached copy of block *right after* the instruction is being executed.

Both of the following answers are correct

Answer 1

Time-line

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Time | Instruction | P0 | P1 | P2 | P3 |
| t0 | P0: read | E/A | NA | NA | NA |
| t1 | P3: write A | I/A | NA | NA | M/A’ |
| t2 | P1: write A | I/A | M/A’’ | NA | I/A’ |
| t3 | P2: read | I/A | I/A’’ | E/A’’ | I/A’ |

Answer 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Time | Instruction | P0 | P1 | P2 | P3 |
| t0 | P0: read | E/A | NA | NA | NA |
| t1 | P3: write A | I/A | NA | NA | M/A’ |
| t2 | P1: write A | I/A | M/A’’ | NA | I/A’ |
| t3 | P2: read | I/A | S/A’’ | S/A’’ | I/A’ |

**Problem 2. [20 points] Parallel Processing**

A shared-memory system with two processors is executing the following two code fragments in parallel (one on each processor). All variables are shared and are initially zeros. The system uses sequential consistency.

|  |  |  |
| --- | --- | --- |
| Processor P0  A=10; |  | Processor P1  Btmp=B; |
| Atmp=A; |  | C=Btmp; |
| B=Atmp-2; |  | if(C==0) |
| while(C==0); |  | C=3; |
| C=6; |  | A=3; |
| printf(“%d %d | %d\n”,A,B,C); |  |

(A) [6 points] Is it possible for the printout in P0 to be “10 8 6”. If yes, show the execution interleaving that produces this printout. If no, explain why not.

Yes, it’s possible

A=10

Atmp=10

Btmp=B

B=Atmp-2

C=Btmp

If C==0

C=3

While(C==0)

C=6

printf

A=3

(B) [6 points] Is it possible for P0 to get stuck and never print out anything? If yes, show the execution interleaving that produces this situation. If no, explain why not.

Not possible. P1 will update C to non-zero value in its final state.

(C) [8 points] Is it possible for P0 to print 3 as the value of C? If yes, show the execution interleaving that produces this printout. If no, explain why not.

Not possible. We now prove it

Let C1 be the statement of “while(C==0)”, C2 be “if(C==0)”, E1 be C=Btmp, E2 be C=3, E3 be C=6, P be “printf”.

From assumption of sequential consistency, we know that

E3🡪P, E1🡪E2, C1🡪E3, C2🡪E2, E1🡪C2 (1)

Because “P0 prints C as 3”, we know that

E3🡪E2🡪P (2)

We know that E3 can’t be executed before E1. Because initial value of C is zero, and E3 can’t be executed if the C1 keeps spinning over zero valued C. So that,

E1🡪E3 (3)

We know that C1 can’t be executed before E1 for the same reason.

E1🡪C1 (4)

Combine (1) (2) (3) (4), we have

E1🡪C1🡪E3🡪E2🡪P (5)

And we know that E1 updates C to an non-zero value.

Now we need to schedule C2. We find that the only place to insert C2 is before E1. Because after E1, C will be an non-zero value and E2 will not be executed. Which means C2🡪E1, this creates a contradiction with the sequential consistency in (1) !!

End of Proof.

**Problem 3. [20 points] Disks and RAID**

A 6,000RPM disk drive has 5 double-surface platters, with 100,000 tracks per surface, 1,000 sectors per tracks, and 512 data bytes per sector. Each sector also has a 128-bit error detection code that can detect all errors in its sector but cannot correct any errors. The head can takes one microsecond (one millionth of a second) to move from one cylinder to an adjacent cylinder, and multi-cylinder movements are done at the same speed (one cylinder per microsecond). The disk controller is very fast (assume zero latency for everything it does) and the I/O bus has very large (assume infinite) bandwidth.

(A) [2 point] How many heads does the disk drive have? Because we have 5 double-surface platters, and there’s one head for each surface, so the disk has10 heads.

(B) [4 points] Assuming that the disk controller and the drive itself are not servicing any other requests, what is the worst-case time needed to read a sector from the disk?

The disk spins at 6000 RPM, so to rotate a full circle, it takes (60seconds/6000) = 10,000 microseconds

Since there’re 1,000 sectors on a track, so that to scan 1 sector, it takes 10,000/1,000 = 10 miscroseconds

Worst case is that moving head across all cylinders, waiting for the rotation of a full circle and then scanning one sector. Because it takes 1 microsecond to move from one cylinder to an adjacent cylinder, so it takes 100,000 microseconds to go across all cylinders. Therefore, the worst case time is

100,000 + 10,000 + 10 = 110,010 micro-seconds.

(C) [4 points] Assuming that the disk controller and the drive itself are not servicing any other requests, what is the best-case time needed to read a sector from the disk?

The best case is when scanning one sector is the only operation needed, therefore, the best case time is,

10 microseconds

Notice: Original question was to ask the time needed to read the entire disk. In this case,

Time = Time to go across all the cylinders + Time to scan all the tracks

Time to go across all the cylinders is 100,000us = 0.1s

Time to scan all the tracks = time to scan all the tracks in one cylinder \* 100,000

= 10,000\*10\*100,000us = 10,000s

So the total time is 10,000.1 seconds

(D) [2 points] If we use two of these disk drives in a RAID0 configuration, what is the total data capacity of the resulting RAID array.

First, we calculate the capacity for one disk.

On each surface, the capacity is 512Byte/sector \* 1000sector \* 100,000

Because the disk has 10 surfaces, capacity for the disk is 512 \* 1000 \* 100,000 \* 10 = 512GB

In RAID0, there’s no data redundancy, which means all the disk capacity is used for storing payload, therefore, the answer is 2\*512GB = 1024GB

(E) [2 points] If we use two of these disk drives in a RAID0 configuration, can we recover all data if one sector is damaged on one of the disk drives? Explain your answer.

No, we can’t. In RAID0, there’s no data redundancy, so when data is damaged, there’s no recovery.

(F) [2 points] If we use two of these disk drives in a RAID1 configuration, can we recover all data if one disk drive is accidentally dropped into an active volcano? Explain your answer.

Yes. It RAID1, there’re two copies for each piece of data. If one disk is damaged, the data can be recovered from another disk.

(G) [2 points] If we use five of these disk drives in a RAID5 configuration, what is the total data capacity of the resulting RAID array? Explain your answer.

In RAID5, we need one disk to store the parity. Therefore, four disks can be used to store payload. The data capacity is 4\*512GB = 2048GB

(H) [2 points] If we use five of these disk drives in a RAID5 configuration, is it possible to damage only two sectors in a way that the RAID array cannot recover from? Explain your answer.

Yes. RAID5 can only tolerate one disk failure. If the two sectors are distributed in two different disks, then there will be no recovery.

**Problem 4. [20 points] Single-choice questions**

1. [1 point] Which cache write mechanism allows out of date data in the memory? [A]

A) Write back

B) Write allocate

C) Write through

D) Non-write allocate

2. [1 point] Which of the following is TRUE for the WSC? [C]

A) WSC is built upon a large number of high-end servers.

B) WSC is built upon a small number of low-end servers.

C) WSC is built upon a large number of low-end servers.

D) In WSC, local disk access is commonly faster than remote memory access.

3. [2 points] Which of the following statements is FALSE? [CD]

A) Memory-mapped IO maps device control registers directly to physical address space.

B) DMA can move data away from physical memory without CPU’s involvement.

C) In disk IO, there are multiple data copies before the application can read the data.

D) In disk IO, polling outperforms interrupt.

4. [2 points] Which of the following statements is TRUE? [E]

A) Cache coherence is about multiple memory locations.

B) Consistency is about one memory location.

C) MSI snoopy protocol requires 3 bits to store states for each memory block.

D) In the write-update cache coherence, there may be out-to-date cached copy.

E) None of the above is TRUE.

5. [1 point] For optimizing memory access, the size of the cache needs to be large and close to the processor. [B]

A) True

B) False

6. [1 point] In a 32-bit address space, with 4K(4096) pages and a page table entry of 8 bytes, what is the page-table size? [B]

A) 32 MB

B) 8 MB

C) 800 KB

D) 32 KB