Homework 9 ● GRADED

Q1 Cache Tracing

25 Points

You are given a 4-way set associative cache. Each cache set has four cache entries, valid bits, and "Least Recently Used (LRU)", which holds a sequence of caches ordered based on how recently were they used. For example, an entry of

$$C1 \rightarrow C2 \rightarrow C3 \rightarrow C0$$

indicates $C0$ as the least recently used cache entry.

In the tables below, an entry denotes the memory address whose data is stored at that location, not the data itself. Additionally, assume the entry in the "Valid" column for an index is the value of the valid bit for each entry at that index. An "O" means each entry in the index is valid, and an "X" means each entry in the index is invalid.

Q1.1 Cache Trace 1

8 Points

Given the current state of the cache below, what will happen if the CPU accesses the memory location of **51**? Assume that location 51 **has** been previously referenced in the cache.

Index	C0	C1	C2	C3	Valid	LRU
0	4	--	36	28	X	--
1	33	5	9	45	O	$C2 \rightarrow C3 \rightarrow C1 \rightarrow C0$
2	14	38	30	18	O	$C0 \rightarrow C2 \rightarrow C1 \rightarrow C3$
3	39	11	7	27	O	$C1 \rightarrow C3 \rightarrow C0 \rightarrow C2$

Hit/Miss & Type:

- ○ Hit
- ○ Miss - Cold/Compulsory
- ○ Miss - Capacity
- ◉ Miss - Conflict

ANSWER

Miss - Conflict

Modified Cache Entry (Choose cache & its index)

Cache

- ◉ C0
- ○ C1
- ○ C2
- ○ C3
- ○ None

ANSWER

C2

Index

- ○ Index 0
- ○ Index 1

QUESTION 1

Cache Tracing **18** / 25 pts

1.1 Cache Trace 1 R **4** / 8 pts

+ 8 pts Correct

Hit/Miss

✔ + 4 pts Marks conflict miss

+ 1 pt Marks cold/compulsory miss (previously referenced)

+ 0 pts Marked capacity miss (the cache is not full) or hit

Modified Cache

+ 2 pts Marks C2

+ 2 pts Marks correct cache for incorrect index (see below)

✔ + 0 pts Marks any other cache or None

Modified Index

+ 2 pts Marks Index 3

✔ + 0 pts Marks any other index or None

+ 0 pts Incorrect

Regrade Request Submitted on: **May 07**

For part 1 of 1.1 (Hit/Miss & Type), Gradescope shows "+4 pts Marks conflict miss" and I choose "Miss - Conflict"

Request Approved!

Reviewed on: **May 07**

ANSWER

Index 3

None

Q1.2 Cache Trace 2

8 Points

Given the current state of the cache below, what will happen if the CPU accesses the memory location of **11**? Assume that location 11 **has** been previously referenced in the cache.

Index	C0	C1	C2	C3	Valid	LRU
0	12	16	0	8	O	$C1 \rightarrow C3 \rightarrow C2 \rightarrow C0$
1	1	13	21	9	O	$C2 \rightarrow C1 \rightarrow C0 \rightarrow C3$
2	-	-	-	-	X	----
3	3	11	7	19	O	$C1 \rightarrow C2 \rightarrow C3 \rightarrow C0$

Hit/Miss & Type:

- (•) Hit
- ( ) Miss - Cold/Compulsory
- ( ) Miss - Capacity
- ( ) Miss - Conflict

Modified Cache Entry (Select the cache & its index)

Cache

- ( ) C0
- ( ) C1
- ( ) C2
- ( ) C3
- (•) None

Index

- ( ) Index 0
- ( ) Index 1
- ( ) Index 2
- ( ) Index 3
- (•) None

1.2 Cache Trace 2 — **8** / 8 pts

- ✔ **+ 8 pts** Correct

Hit/Miss

- **+ 4 pts** Marks hit
- **+ 0 pts** Marks any miss

Modified Cache

- **+ 2 pts** Marks None (no cache is modified on a hit)
- **+ 1 pt** Marks C1 (the correct cache), even though it is not modified
- **+ 0 pts** Marks any other cache

Modified Index

- **+ 2 pts** Marks None
- **+ 1 pt** Marks Index 3 (the correct index), even though it is not modified
- **+ 0 pts** Marks any other index

Q1.3 Cache Trace 3

8 Points

Given the current state of the cache below, what will happen if the CPU accesses the memory location of **12**? Assume that location 12 **has NOT** been previously referenced in the cache.

Index	C0	C1	C2	C3	Valid	LRU
0	32	4	16	8	O	$C2 \rightarrow C3 \rightarrow C1 \rightarrow C0$
1	5	13	33	41	O	$C3 \rightarrow C1 \rightarrow C2 \rightarrow C0$
2	18	14	26	22	O	$C0 \rightarrow C1 \rightarrow C3 \rightarrow C2$

Index	C0	C1	C2	C3	Valid	LRU
3	35	43	15	7	0	$C1 \rightarrow C0 \rightarrow C2 \rightarrow C3$

Hit/Miss & Type:

- ○ Hit
- ○ Miss - Cold/Compulsory
- ◉ Miss - Capacity
- ○ Miss - Conflict

Modified Cache Entry (Choose cache & its index)

Cache

- ◉ C0
- ○ C1
- ○ C2
- ○ C3
- ○ None

Index

- ◉ Index 0
- ○ Index 1
- ○ Index 2
- ○ Index 3
- ○ None

1.3 Cache Trace 3 — **5** / 8 pts

+ 8 pts Correct

Hit/Miss

+ 4 pts Marks cold/compulsory miss

✔ + 1 pt Marks any other miss

+ 0 pts Marks hit

Modified Cache

✔ + 2 pts Marks C0

+ 2 pts Marks correct cache for incorrect index (see below)

+ 0 pts Marks any other cache or None

Modified Index

✔ + 2 pts Marks Index 0

+ 0 pts Marks any other index or None

+ 0 pts Incorrect

Q1.4

1 Point

How many bits are required to keep track of the ordering of 8 way set associative?

- ○ 40320
- ○ 256
- ◉ 16
- ○ 8

1.4 (no title) — **1** / 1 pt

✔ + 1 pt Correct

+ 0 pts Incorrect

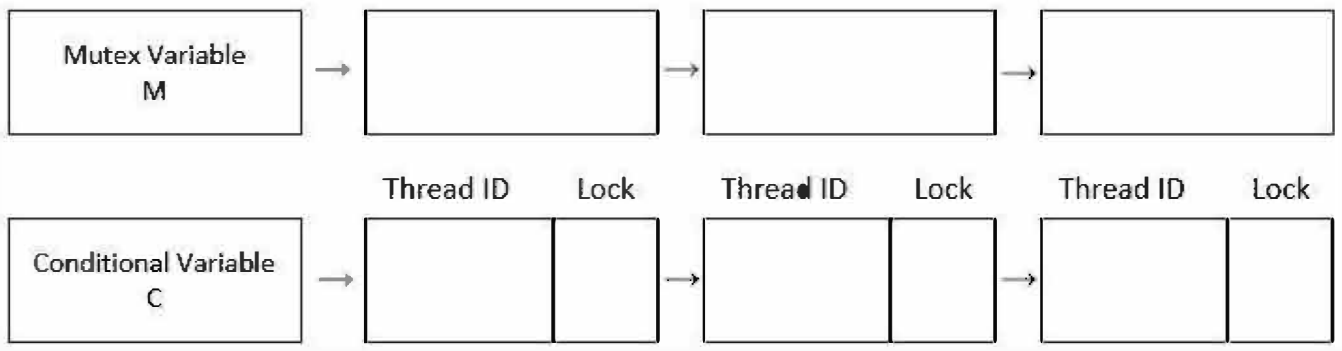
Q2 Conditional Variable

24 Points

Given a mutex lock m, and a conditional variable c, the following events happen in the order of occurrence stated below:

1. $T2$ executes mutex-lock(m).
2. $T1$ executes mutex-lock(m).
3. $T2$ executes cond-wait(c, m).
4. $T1$ executes cond-signal(c).
5. $T1$ executes mutex-unlock(m)

Fill in the boxes below. If there is no thread waiting for a lock or a signal, leave the boxes blank.



Q2.1 Before Step 4

8 Points

Write down the state of the two waiting queues after steps 1, 2, and 3 are completed.

For the mutex variable m, write your answers as a comma-separated list with the first entry being the thread that currently has the lock. For example, if T1 has the lock and T2 is waiting for the lock, you should answer "T1, T2". If a queue is empty, write down "NA"

For the conditional variable c, write a comma-separated list of threads waiting in order that they arrived. For instance, if T1 and T2 are waiting for mutex lock m, you should answer "T1 m, T2 m."

Mutex variable m

T1

Conditional variable c

T2 m

2.1 Before Step 4 **8** / 8 pts

- ✔ − **0 pts** Correct
  - m: T1
  - c: T2 m
- − **0 pts** Correct
- − **4 pts** Does not only have T1 holding Mutex
- − **4 pts** Incorrect condition variable (wrong thread, also T2, etc.)
- − **8 pts** Incorrect

Q2.2 Before Step 5

8 Points

Write down the state of the two waiting queues **before** step 5. Steps 1, 2, 3, and 4 have completed.

For the mutex variable m, write your answers as a comma-separated list with the first entry being the thread that currently has the lock. For example, if T1 has the lock and T2 is waiting for the lock, you should answer "T1, T2." If a queue is empty, write down "NA"

For the conditional variable c, write a comma-separated list of threads waiting in order that they arrived. For instance, if T1 and T2 are waiting for mutex lock m, you should answer "T1 m, T2 m."

Mutex variable m

T1, T2

Conditional variable c

T2 m

2.2 Before Step 5 **4** / 8 pts

- − **0 pts** Correct
  - m: T1, T2
  - c: NA
- − **0 pts** Correct

Mutex

- − **2 pts** Incorrect order in mutex queue (T2 then T1)
- − **2 pts** Missing one thread in the condition variable queue
- − **4 pts** Incorrect mutex queue (beyond or all of the above cases; i.e. do not take off more than 4 points for mutex)

- ✔ − **4 pts** Incorrect condition variable (anyone in the queue)
- − **8 pts** Incorrect

Q2.3 After All Steps

8 Points

Write down the state of the two waiting queues after all the steps are completed.

For the mutex variable m, write your answers as a comma-separated list with the first entry being the thread that currently has the lock. For example, if T1 has the lock and T2 is waiting for the lock, you should answer "T1, T2" If a queue is empty, write down "NA"

For the conditional variable c, write a comma-separated list of threads waiting in order that they arrived.

Mutex variable m

T2

Conditional variable c

2.3 After All Steps 8 / 8 pts

- ✔ – 0 pts Correct
  - m: T2
  - c: NA
- – 0 pts Correct
- – 4 pts Incorrect mutex
- – 4 pts Incorrect condition variable (anyone in the queue)
- – 8 pts Incorrect

Q3 Page Coloring

24 Points

Page coloring is used to make sure that a few least significant bits of the virtual page number (VPN) and physical frame number (PFN) remain unchanged during address translation.

Imagine the following memory hierarchy:

- 64-bit virtual address
- 32-bit physical address
- Virtually-indexed, physically-tagged, 2-way set associative cache
- Page size of 8 KB
- Memory is byte-addressable
- Total Cache Size of 256 KB
- Cache block size of 128 bytes

Assume K = 1024 and M = 1024 * 1024.

Q3.1 Overlapping Bits

8 Points

How many of the least significant bits of the VPN must remain unchanged in the VPN-PFN translation?

4

3.1 Overlapping Bits 8 / 8 pts

✔ – 0 pts Correct

Q3.2 Overlap in Cache Address

4 Points

Where in the cache address are the overlapping bits present? (End describes positions touching MSB, and beginning describes positions touching LSB)

- ○ End of the tag
- ◉ Beginning of the index
- ○ Middle of the index
- ○ End of the index
- ○ Beginning of the offset

3.2 Overlap in Cache Address 0 / 4 pts

+ 4 pts Correct

✔ + 0 pts Incorrect

Q3.3 Page Coloring Justification

12 Points

Page coloring provides multiple benefits when it comes to performance in a virtual memory system which implements caches. Identify and explain ONE benefit of page coloring.

Page Coloring guarantees that more of the virtual address bits will remain unchanged by the translation process by choosing the VPN to PFN mapping. It also allows the processor to have a larger virtually indexed physically tagged cache independent of the page size

3.3 Page Coloring Justification 12 / 12 pts

+ 12 pts Allows the processor to perform TLB lookups and cache lookups in parallel

+ 12 pts Correct

+ 0 pts Incorrect

+ 12 pts Ensures contiguous virtual pages are mapped to different cache entries and thus should not conflict (improve spatial locality usage)

✔ + 12 pts Allows for expanded cache sizes, since the cache index can be partially contiguous with the VPN/PFN without having to worry about the index changing after the VPN-PFN address translation

+ 6 pts Partially correct answer -- understands what page coloring does, but does not fully describe a benefit

+ 0 pts Incorrect/blank/no answer

Q3.4 Work (Optional)

0 Points

If you would like partial credit in case of an incorrect answer on the previous parts, show your work in the field below or attach it as a file:

No files uploaded

3.1:
offset bits = log2(page size) = log2(8 kb) = 13
64-13 = 49 VPN

2-way set associative * 128 bytes/block = 256bytes/set

256 kB total size / 256 bytes/set = 1k (1000) sets which implies 10 bits for the index

so we will need 4 bits to represent 10 values

Q4 SMP Cache Coherence

28 Points

We are using a symmetric multiprocessor (SMP) with the following specifications:

- Two processors, P1 and P2
- Each processor has a single cache which is initially empty
- The cache can contain x and y at the same time (i.e. no conflicts)
- Cache coherence protocol: **Write-invalidate**
- Cache to memory policy: **Write-back**
- Memory location x initially contains the value 2
- Memory location y initially contains the value 3

Consider the following memory accesses from P1 and P2:

Time	P1	P2
T1	Load x	Load y
T2	Store 4 to y	Store 6 to x
T3	---	Store 7 to y
T4	Load x	---

Answer the questions below to summarize the activities and values in the caches at **the specified units of time**. Use NP to represent that a value is not present in the cache or that the current entry is invalid.

Q4.1 After T2

12 Points

4.1 After T2 **12** / 12 pts

✔ – **0 pts** Correct

What is the value of x in the cache of P1?

NP

What is the value of x in the cache of P2?

6

What is the value of x in memory?

2

What is the value of y in the cache of P1?

4

What is the value of y in the cache of P2?

NP

What is the value of y in memory?

3

Q4.2 After T4

12 Points

What is the value of x in the cache of P1?

2

What is the value of x in the cache of P2?

NP

What is the value of x in memory?

2

What is the value of y in the cache of P1?

I

What is the value of y in the cache of P2?

7

What is the value of y in memory?

3

4.2 — After T4 **8** / 12 pts

– 0 pts	Correct
✔ – 2 pts	Incorrect value of x in P1's cache (6)
✔ – 2 pts	Incorrect value of x in P2's cache (6)
– 2 pts	Incorrect value of x in memory (2)
– 2 pts	Incorrect value of y in P1's cache (NP)
– 2 pts	Incorrect value of y in P2's cache (7)
– 2 pts	Incorrect value of y in memory (3)
– 12 pts	Incorrect

Q4.3 New Memory Access

4 Points

Now, assume that after T4, the process executing on **P2** writes the value of 12 to a new memory location z which is mapped to the **same cache entry** as y, causing y to be evicted from P2's cache. After this operation, what is the value of y in memory?

- 2
- 4
- 7
- 12
- Cannot be determined

4.3 — New Memory Access **4** / 4 pts

✔ + 4 pts	Correct
+ 0 pts	Incorrect

Q4.4 Work (Optional)

0 Points

If you would like partial credit in case of an incorrect answer on the previous parts, show your work (the full table) in the field below or attach it as a file:

▾ Screenshot 2022-04-13 235127.jpg Download

Time	Variables	Cache of P1	Cache of P2	Memory
T1	X	2	NP	2
	Y	NP	3	3
T2	X	I	6	2
	Y	4	I	3
T3	X	I	6	2
	Y	I	7	3
T4	X	2	I	2
	Y	I	7	3

Homework 9

GRADED

STUDENT
Eric Anders Gustafson

TOTAL POINTS
82 / 101 pts

QUESTION 1
Cache Tracing — **18** / 25 pts

1.1 Cache Trace 1 — R **4** / 8 pts

+ 8 pts Correct

Hit/Miss

✔ + 4 pts Marks conflict miss

+ 1 pt Marks cold/compulsory miss (previously referenced)

+ 0 pts Marked capacity miss (the cache is not full) or hit

Modified Cache

+ 2 pts Marks C2

+ 2 pts Marks correct cache for incorrect index (see below)

✔ + 0 pts Marks any other cache or None

Modified Index

+ 2 pts Marks Index 3

✔ + 0 pts Marks any other index or None

+ 0 pts Incorrect

Regrade Request Submitted on: **May 07**

For part 1 of 1.1 (Hit/Miss & Type), Gradescope shows "+4 pts Marks conflict miss" and I choose "Miss - Conflict"

Request Approved!

Reviewed on: **May 07**

1.2 Cache Trace 2 — **8** / 8 pts

1.3	Cache Trace 3	**5** / 8 pts
1.4	(no title)	**1** / 1 pt

QUESTION 2

	Conditional Variable	**20** / 24 pts
2.1	Before Step 4	**8** / 8 pts
2.2	Before Step 5	**4** / 8 pts
2.3	After All Steps	**8** / 8 pts

QUESTION 3

	Page Coloring	**20** / 24 pts
3.1	Overlapping Bits	**8** / 8 pts
3.2	Overlap in Cache Address	**0** / 4 pts
3.3	Page Coloring Justification	**12** / 12 pts
3.4	Work (Optional)	**0** / 0 pts

QUESTION 4

	SMP Cache Coherence	**24** / 28 pts
4.1	After T2	**12** / 12 pts
4.2	After T4	**8** / 12 pts
4.3	New Memory Access	**4** / 4 pts
4.4	Work (Optional)	**0** / 0 pts