Assignment 2

Parallel Architectures

Rodrigo C. O. Rocha
s1533346

1 Introduction

In this work, we implement a cache simulator to compared several cache-coherence protocol in a multi-core processor with a single level of cache. Section 2 we describe our implementation for the MSI snooping protocol. Section 3 we describe our implementation for the directory-based protocol. Finally, in Section 4 we describe our implementation for the MESI snooping protocol, including the addition of a second-level shared cache. For each cache-coherence protocol, we evaluate its performance in two work-load traces.

In all cases, we consider *direct mapped caches*. In direct mapped caches, the main memory is divided into blocks of memory locations, storing multiple words per block, and caches are organised by lines, called *cache lines*. Each cache line is capable of storing exactly one memory block, a tag that uniquely identifies the corresponding memory block, and the current state of the cache line. Considering this organisation, the direct mapping divides each memory address into *tag*, *slot* and *offset*, as illustrated by Figure 1. The offset represents the word index within the memory block, the slot represents the index

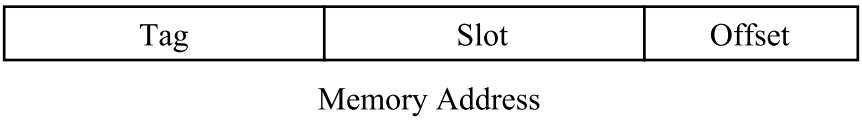

Figure 1: Memory address decomposed into a direct mapped cache.

of the cache line, and the tag uniquely identifies the memory address, given that we know the corresponding slot and offset.

In this report, we consider architectures with blocks of size 4, i.e., 2 bits of offset, and caches with 512 lines, i.e., 9 bits of slots, and the remaining 21 bits, considering a 32 bits memory address, for the tag.

2 MSI Snooping Protocol

For the MSI snooping protocol, we consider the architecture illustrated in Figure 2, where the cores are inter-connected by an unidirectional ring.

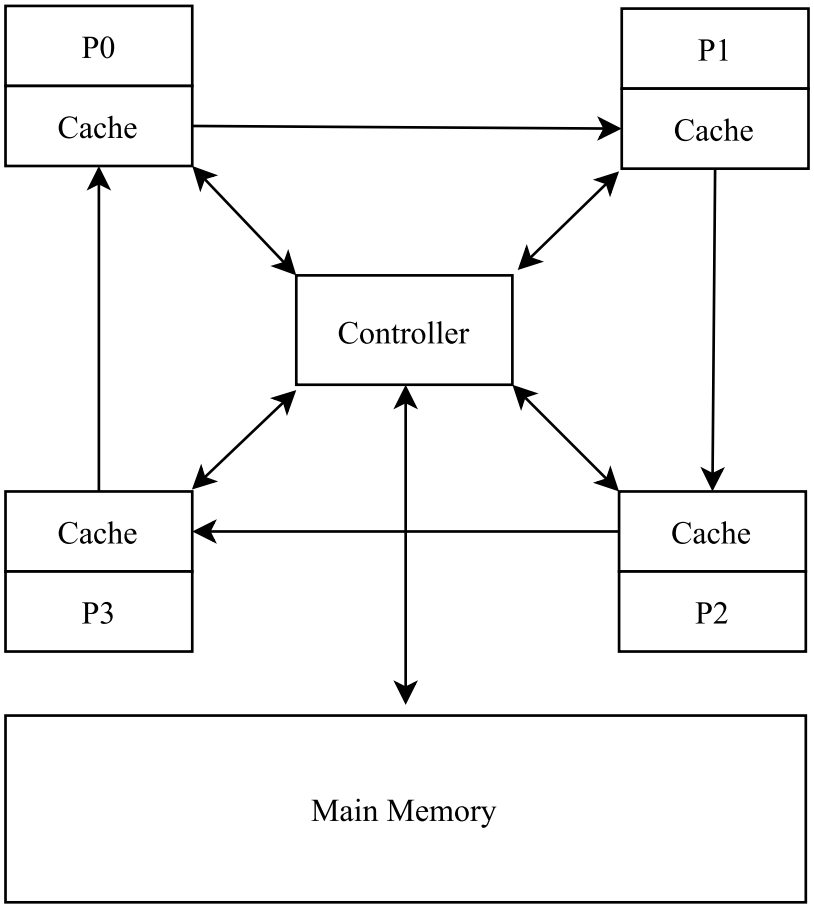

Figure 2: Diagram of the architecture used with the MSI snooping protocol.

In order to verify the correctness of the implementation, we show several small traces executed in verbose mode.

```
P1 R 10
Processor P1 reads word in address 10, with tag 0 and slot 2.
A cache miss occurred. Data transferred from main memory.
(33 cycles taken)
P1 R 10
Processor P1 reads word in address 10, with tag 0 and slot 2.
A cache hit occurred with cache line found in state Shared.
(2 cycles taken)
```

Figure 3: (MSI Snooping Protocol) Test case: Reads with cache miss and cache hit.

```
P1 R 10
Processor P1 reads word in address 10, with tag 0 and slot 2.
A cache miss occurred. Data transferred from main memory.
(33 cycles taken)
P3 R 10
Processor P3 reads word in address 10, with tag 0 and slot 2.
A cache miss occurred. Data transferred from processor P1,
found in state Shared.
(17 cycles taken)
P1 W 10
Processor P1 writes word in address 10, with tag 0 and slot 2.
Cache line found in state Shared. A total of 1 invalidations were required.
(17 cycles taken)
```

Figure 4: (MSI Snooping Protocol) Test case: Write at P1; Local cache line in S state, Remote cache line at P3 in S state; Expected 17 cycles.

3 Directory-based Protocol

For the directory-based protocol, we consider the architecture illustrated in Figure 8, where the cores are inter-connected by an bidirectional ring. The controller is supposed to be able to access the directory immediatly, i.e., within the same cycle.

In order to verify the correctness of the implementation, we show several small traces executed in verbose mode.

```
P1 R 10
Processor P1 reads word in address 10, with tag 0 and slot 2.
A cache miss occurred. Data transferred from main memory.
(33 cycles taken)
P2 W 10
Processor P2 writes word in address 10, with tag 0 and slot 2.
A cache miss occurred. Data transferred from processor P1,
found in state Shared. A total of 1 invalidations were required.
(18 cycles taken)
P3 R 10
Processor P3 reads word in address 10, with tag 0 and slot 2.
A cache miss occurred. Data transferred from processor P2,
found in state Modified. A protocol writeback was required by processor P2.
(18 cycles taken)
P1 W 10
Processor P1 writes word in address 10, with tag 0 and slot 2.
A cache miss occurred due to state Invalid.
Data transferred from processor P2, found in state Shared.
A total of 2 invalidations were required.
(18 cycles taken)
```

Figure 5: (MSI Snooping Protocol) Test case: Write at P1; Local cache line in I state, Remote cache line at P2, P3 in S state; Expected 18 cycles.

4 MESI Snooping Protocol with Second-Level Shared Cache

For the optimised snooping protocol, we implemented the MESI protocol with a second-level shared cache. In the results we show that the MESI protocol alone can improve over the MSI snooping protocol, while the 1KB second-level shared cache provides a very small improvement. We consider the architecture illustrated in Figure 8, where the cores are inter-connected by an unidirectional ring. The controller is supposed to be able to probe the shared cache in one cycle.

```
P1 R 10
Processor P1 reads word in address 10, with tag 0 and slot 2.
A cache miss occurred. Data transferred from main memory.
(33 cycles taken)
P2 W 10
Processor P2 writes word in address 10, with tag 0 and slot 2.
A cache miss occurred. Data transferred from processor P1,
found in state Shared. A total of 1 invalidations were required.
(18 cycles taken)
P1 R 10
Processor P1 reads word in address 10, with tag 0 and slot 2.
A cache miss occurred due to state Invalid.
Data transferred from processor P2, found in state Modified.
A protocol writeback was required by processor P2.
(16 cycles taken)
```

Figure 6: (MSI Snooping Protocol) Test case: Read at P1; Local cache line in I state, Remote cache line at P2 in M state; Expected 19 cycles.

5 Results

6 Conclusion

From the results we can see that the directory-based protocol has the best performance amongst the protocols. The shared cache has no major impact due to its limited size of 1KB and the simple prefetching mechanism based on keeping the replaced blocks and prefetching a single adjacent block when reading data from the main memory.

```
Trace 1                          Trace 2
Total Reads: 163840              Total Reads: 507093
Total Writes: 32768              Total Writes: 76462
Total Requests: 196608           Total Requests: 583555
Hit Rate: 0.912679036458         Hit Rate: 0.84227022303
Direct Write: 24440              Direct Write: 34942
Invalidations: 51                Invalidations: 52559
Invalidation Sent: 8328          Invalidation Sent: 41520
Private Accesses: 179440         Private Accesses: 491511
Remote Accesses: 8427            Remote Accesses: 91537
Off-chip Accesses: 8741          Off-chip Accesses: 507
Protocol Writeback: 51           Protocol Writeback: 39537
Replacement Writeback: 6357      Replacement Writeback: 0
Total Cycle Count: 790691        Total Cycle Count: 2547924
Mean Cycle Per Req.: 4.02166239421  Mean Cycle Per Req.: 4.36621055428
```

Figure 7: Statistics for the MSI protocol on Traces 1 and 2.

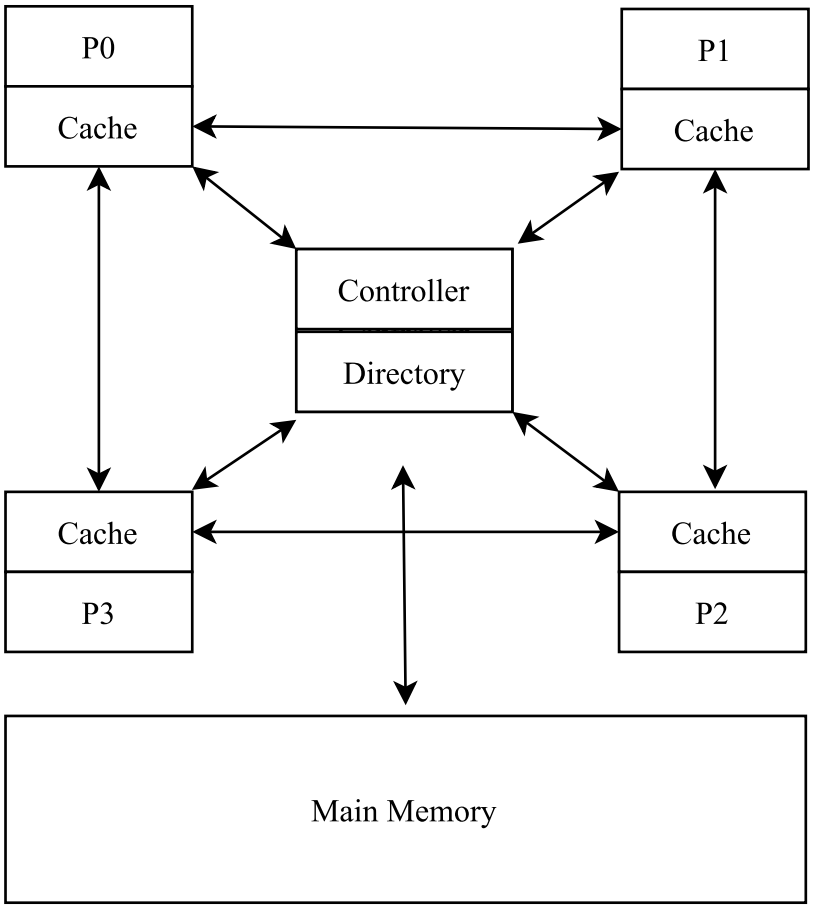

Figure 8: Diagram of the architecture used with the directory-based protocol.

```
P2 W 10
Processor P2 writes word in address 10, with tag 0 and slot 2.
A cache miss occurred. Data transferred from main memory.
A total of 0 invalidations were required.
(18 cycles taken)
P3 R 10
Processor P3 reads word in address 10, with tag 0 and slot 2.
A cache miss occurred. Data transferred from remote processor
in Exclusive state with distance 1.
A protocol writeback was required by processor P2.
(13 cycles taken)
P1 W 10
Processor P1 writes word in address 10, with tag 0 and slot 2.
A cache miss occurred. Data transferred from remote processor
in Shared state with distance 1.
A total of 2 invalidations were required.
(13 cycles taken)
```

Figure 9: (Directory-based Protocol) Test case: Write at P1; Local cache line in I state, Remote cache line at P2, P3 in S state.

```
Trace 1                          Trace 2
Total Reads: 163840              Total Reads: 507093
Total Writes: 32768              Total Writes: 76462
Total Requests: 196608           Total Requests: 583555
Hit Rate: 0.912679036458         Hit Rate: 0.995224100556
Direct Write: 24440              Direct Write: 74622
Invalidations: 0                 Invalidations: 0
Invalidation Sent: 0             Invalidation Sent: 0
Private Accesses: 179440         Private Accesses: 580768
Remote Accesses: 8427            Remote Accesses: 2280
Off-chip Accesses: 8741          Off-chip Accesses: 507
Protocol Writeback: 51           Protocol Writeback: 455
Replacement Writeback: 6357      Replacement Writeback: 0
Total Cycle Count: 584129        Total Cycle Count: 1195843
Mean Cycle Per Req.: 2.9710337321 Mean Cycle Per Req.: 2.04923786104
```

Figure 10: Statistics for the Directory-based protocol on Traces 1 and 2.

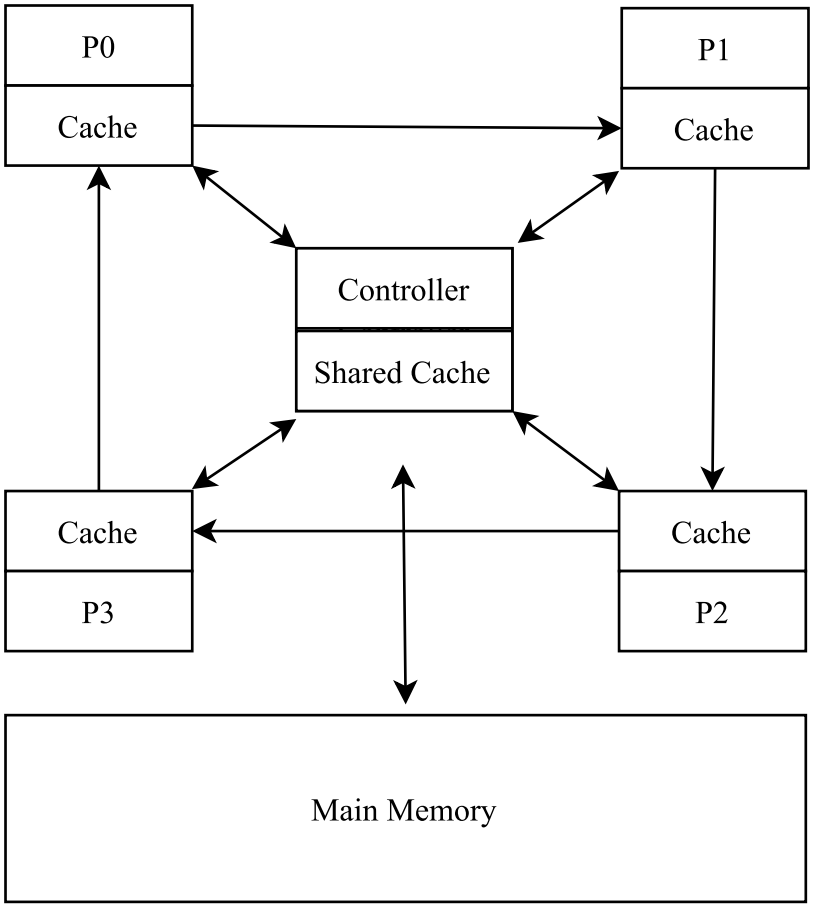

Figure 11: Diagram of the architecture used with the MESI snooping protocol with the second-level shared cache close to the memory controller.

```
Trace 1                              Trace 2
Total Reads: 163840                  Total Reads: 507093
Total Writes: 32768                  Total Writes: 76462
Total Requests: 196608               Total Requests: 583555
Hit Rate: 0.954778035482             Hit Rate: 0.855568027007
Direct Write: 32717                  Direct Write: 38839
Invalidations: 51                    Invalidations: 48557
Invalidation Sent: 153               Invalidation Sent: 111159
Private Accesses: 187717             Private Accesses: 499271
Remote Accesses: 150                 Remote Accesses: 83641
Off-chip Accesses: 8741              Off-chip Accesses: 643
Protocol Writeback: 51               Protocol Writeback: 35757
Replacement Writeback: 6357          Replacement Writeback: 0
Total Cycle Count: 663037            Total Cycle Count: 2432247
Mean Cycle Per Req.: 3.37238057454   Mean Cycle Per Req.: 4.16798245238
```

Figure 12: Statistics for the MESI Snooping protocol with 1KB of Shared Cache on Traces 1 and 2.

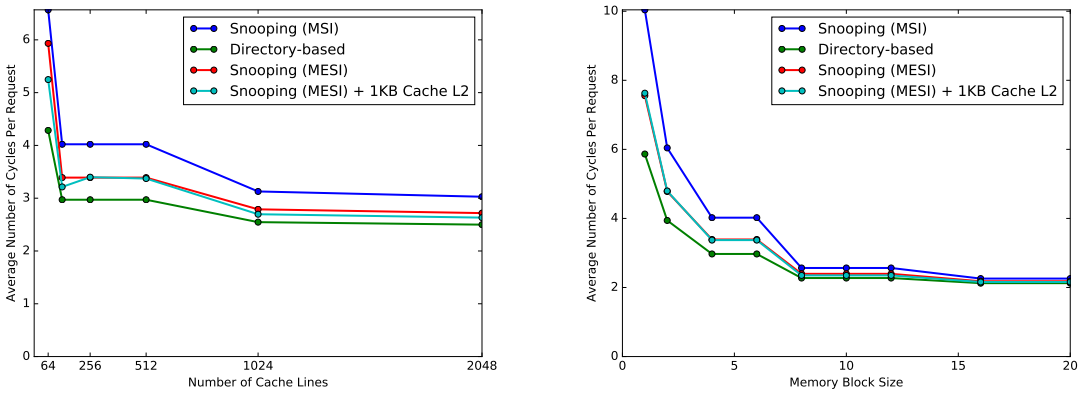

Figure 13: Comparison of the cache-coherence protocols on Trace 1.

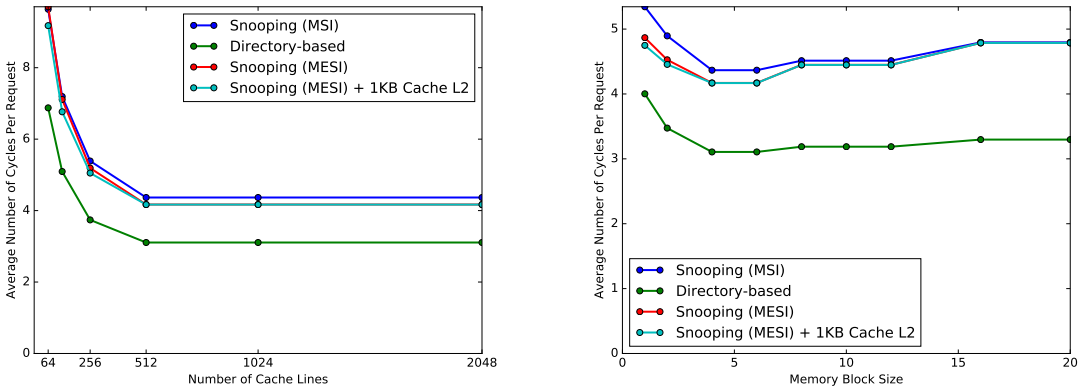

Figure 14: Comparison of the cache-coherence protocols on Trace 2.

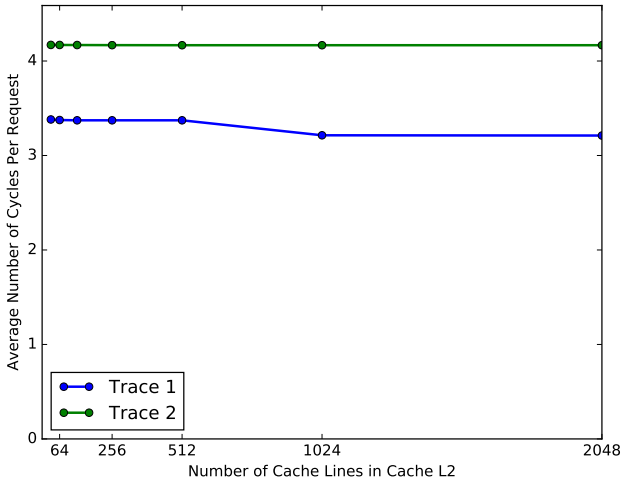

Figure 15: Performance of the MESI protocol with Second-Level Shared Cache, when varying the size of the shared cache.