ECE 585

MICROPROCESSOR SYSTEM DESIGN

(Winter 2024)

A PROJECT REPORT

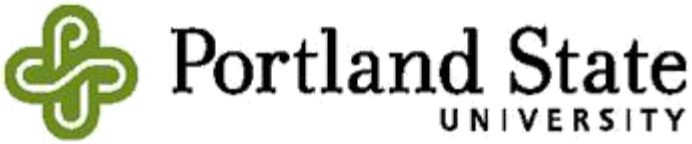
submitted in partial fulfillment of the requirements
for the award of the degree of

Master of Science
in
ELECTRICAL AND COMPUTER ENGINEERING

Guide: Professor Yuchen Huang

AUTHORS

SUGGU V S S K K L SAI TAGORE (989262683)
SURYA VAMSI TIRUVEEDHULA (934707602)
DIVYA SRI AYLURI (976089658)
RAFATH ACHUGATLA (934857968)



CONTENTS

1) Objective

A cache is a component in the computing environment that stores the information to be served quickly when requested by the processor. They are high speed memory elements that reside closer to the microprocessor for servicing all the memory request by the processor quickly.
Design and simulate a split L1 cache for a new 32-bit processor that can be used in a shared memory arrangement with up to three other processors. The system employs MESI Protocol to ensure cache coherence.

2) Design Specification

This is a design of a first level split cache of a core in a multi-processor environment using a shared memory configuration. The specifications of the cache are as follows:

1. 8-way Set Associative Data Cache and 4-way Set Associative Instruction Cache
2. There are 16K sets in each cache.
3. The cache line size is 64bits.
4. Implements MESI protocol to ensure cache coherence.
5. The cache must employ inclusivity.
6. Two modes: Mode 0- only display statistics, Mode 1- do all operations and display L2 cache interactions.
7. The caches employ write-back policy except when it is the first write to that line, it will be a write-through.
8. LRU replacement policy is used for eviction strategy and backed by a shared L2 cache.

At the end of simulation, the following must be printed:

- Number of cache reads
- Number of cache writes
- Number of cache hits
- Number of cache misses
- Cache hit ratio.

Cache Coherence: It can be defined as the uniformity of shared resource data that ends up getting stored in numerous local caches.

MESI Protocol: It is one of the most popular Cache Coherence Protocols that has four states which are described as follows:

- ***Modified State:***

  In a modified state, the cached copy is the only valid copy, and no other processor contains the same information.

  The cached content won't be present in the main memory, that is, the memory copy is out of date.

- ***Exclusive State:***

  The exclusive state implies that the cached contents are exclusive to a particular processor and are not present in any of the other processors.

  The cached content of the processor is like that present in the main memory.

- ***Shared State:***

  When the same cached contents are present in two or more processors, then the processors are said to be in a Shared state.

  In this case, the cached contents are also present in the main memory.

- ***Invalid State:***

  This state simply pimples that the cache entry is invalid because it doesn't hold a copy of that line.

The simulator reads the trace file and performs operations that correspond to the trace number on the specified address in the trace file.

Simulator considers:

- A read data request to the L1 data cache is indicated by ***trace number 0***.
- ***Trace number 1*** denotes a request to write data to the L1 data cache.
- Instruction fetch is shown by ***trace number 2*** (a read request to L1 instruction cache)
- ***Trace number 3*** denotes a command from L2 that was invalidated.
- ***Trace number 4*** denotes an L2 data request (in response to snoop).
- ***Trace number 8*** denotes a clear cache and state reset (and statistics).
- The print contents and cache status are indicated by ***trace number 9*** (allow subsequent trace activity).

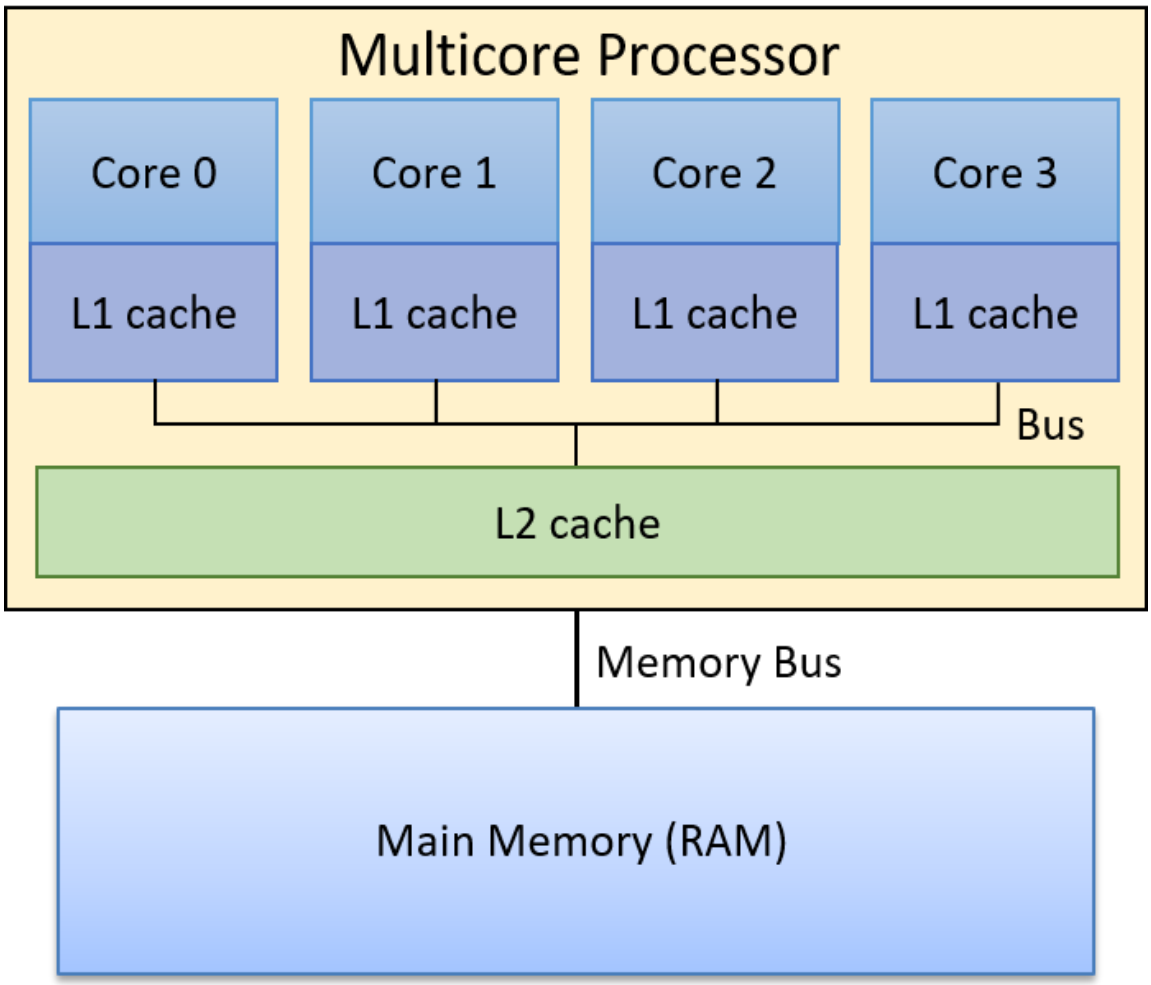
The simulator has two operating modes:

1. Mode 0 shows data usage and responds to trace number 9, and

2. Mode 1 shows everything in Mode 0 along with the messages used to communicate with the L2 cache. Without recompiling the code, the mode can be changed in the middle of a runtime.

The results of the simulation are printed in the transcript.

3) Block Diagram



4) Assumptions

1. We assume that all the caches, when reset are in the INVALID state.
2. During a write operation, if there is a miss we read the necessary cache line from L2 cache. If the cache line, we are writing into in L1 cache is a first write into that cache we are assuming that the cache line goes into an "EXCLUSIVE" state.
3. Since we do not have an L2 cache, we are simulating the read and write from L2 with display statements.

5) Operations of Cache

The operations with trace numbers 0, 1, 3, 4, 8, 9 involve the data cache and the ones with trace number 2, 8, 9 involve instruction cache.

Split L1 Cache

1.Data Cache:

Each CPU has its own data cache, that it can either write to and read from. Taking this into account, the following possibilities are possible (without considering the MESI protocol and LRU bits into account):

CPU Data Read (Command: 0)

- **Hit:**

  The L1 data cache will be read by the CPU.

- **Miss:**

  Prior to reading data from the L1 data cache, the processor reads L2 cached data into the L1 data cache.

CPU Data Write (Command: 1)

- **Hit:**

  The dirty data is moved to the L2 cache first if the dirty bit is set, and then new data that has the dirty bit cleared is stored to the L1 data cache.

  The dirty bit is set if it is not already set, and the new data is written into the L1 cache.

- **Miss:**

  In the case that the dirty bit is set, the dirty data is written back to the L2 cache first, after which the relevant cache line is read into the L1 data cache, and the new data is stored in.

  New data is written to the required cache line after reading it from the L1 data cache, if the dirty bit is not set.

Invalidate command from L2(Command: 3)

- ❖ The dirty data is first written back to the L2 if the dirty bit is set, and then the dirty bit and valid bit are both set to 0.
- ❖ If the dirty bit is zero, the valid bit is set to zero.

Data request from L2(Command: 4)

- ❖ The dirty data is written to the L2 first, then the dirty bit and valid bit are set to zero if the dirty bit is set.
- ❖ Once the dirty bit is set to zero, the valid bit is also set to zero.

2.Instruction Cache:

CPU Instruction Read (Command: 2)

- ➢ **Hit:**
  The L1 data cache will be read by the CPU.
- ➢ **Miss:**
  Prior to reading data from the L1 data cache, the processor reads L2 cached data into the L1 data cache.

Common operations between data and instruction cache:

1. Reset all states and clear the cache (and statistics) (Trace no. 8). The whole L1 cache including the valid bits, dirty bits, and usage statistics is cleared.
2. The state and content of the cache should be printed (to allow further tracing activity) (Trace number 9) The cache's entire contents, as well as usage data, are printed.

6) LRU Replacement Policy of Cache:

1.Data Cache:

The data cache is an associative eight-way set cache. In the 8-way set associative, we require 3 bits to retain LRU position.
The LRU bits are initially set to hex 0 (binary 3'b0). Upon reset, all of the ways' LRU bits and the set's LRU bits will be set to 0.
According to the implementation of the Cache Simulator, LRU 7 is the most recently accessed location and LRU 0 is the least recently accessed place. When any way of a specific set is accessed, the LRU bit changes to 7. The way with LRU = 0 is evicted when the cache is full during eviction, and a new tag with LRU = 7 is loaded into the cache.

2.Instruction Cache:

The instruction cache is a four-way associative set. To maintain LRU status in the four-way set associative, two bits are required.
The LRU bits are initially set to hex 0 (binary 2'b0). Upon reset, the LRU bits of each method and the set will both be set to 0.
The implementation of the Cache Simulator shows that LRU 3 is the most recently accessed location and LRU 0 is the least recently accessible location. When any way of a certain set is accessed with a 9, the LRU bit changes to 3. The way with LRU = 0 is evicted when the cache is full during eviction, and a new tag with LRU = 3 is loaded into the cache.

7) Test cases and Results:

Test Cases:

2 02020202
1 10305008
2 9abcdef0
1 f3f3f2ff
2 0000ffff
1 ceaddeef
2 cafebabe
1 8000000
0 7fffffff
0 00112233
0 ffeeddcc
0 55ab55c5
0 aaaaaaaa
0 01234567
0 89abcdef
0 7f7f7f7f
0 80808080
1 0f0f0f0f
2 f0f0f0f0
3 12341234
4 56785678
2 9abc9abc
3 def0def0
3 deadface
2 baadf00d
4 feedface
4 12344321

4 67896789
3 abcdabcd
2 ef00ef00
1 7fffffff
1 00112233
1 ffeeddcc
1 55555555
1 aaaaaaaa
1 01234567
1 89abcdef
1 7f7f7f7f
1 80808080
0 12345678
0 ffffffff
0 deadbeef
0 80000000
0 0f0f0f0f
0 00000000
0 9abcdef0
0 0000ffff
0 babecabe
0 f0f0f0f0
0 9abc9abc
0 faadd00d
0 ef00ef00
2 00000000
2 9abcdef0
2 0000ffff
2 cafebabe
2 f0f0ff0
2 9abc9abc
2 baadf00d
2 ef00ef00
8 abcdef01
9 543210ef

Results:

```
#
# 1 481C7631
#
# -----TAG BITS DONT MATCH - MISS-----
#
# 1 481C7631
#
# ------TAG BITS MATCH, HENCE HIT-------
#
# 0 582C7632
#
# -----TAG BITS DONT MATCH - MISS-----
#
# 0 594C7615
#
# -----TAG BITS DONT MATCH - MISS-----
#
# 0 621C7600
#
# -----TAG BITS DONT MATCH - MISS-----
#
# 1 111C762C
#
# -----TAG BITS DONT MATCH - MISS-----
#
# 9 0000
#
# ------------------------------Contents of INSTRUCTION CACHE------------------------------
# TRACE Address = 0
# SET NUMBER          = 0
# TAG3 = 0       TAG2 = 0       TAG1 = 0       TAG0 = 0
# STATE3 = INVALID  STATE2 = INVALID  STATE1 = INVALID  STATE0 = INVALID
# LRU3 = 0       LRU2 = 0       LRU1 = 0       LRU0 = 0
# VALID3 = 0     VALID2 = 0     VALID1 = 0     VALID0 = 0
#
# ---INSTRUCTION CACHE statistics---
# Total Number of instruction reads  = 0
# Total Number of instruction writes = 0
# Total Number of instruction hits   = 0
# Total Number of instruction misses = 0
# Denominator cannot be zero (ie, data_miss and data_hit is zero)
# ------------------------------------------------------------------------------Contents of DATA CACHE-------------------------------------------------------------------------------
# TRACE ADDRESS = 0
# SET NUMBER   = 0
# TAG7 = 0       TAG6 = 0       TAG5 = 0       TAG4 = 0       TAG3 = 0       TAG2 = 0       TAG1 = 0       TAG0 = 0
# STATE7 = INVALID  STATE6 = INVALID  STATE5 = INVALID  STATE4 = INVALID  STATE3 = INVALID  STATE2 = INVALID  STATE1 = INVALID  STATE0 = INVALID
# LRU7 = 0       LRU6 = 0       LRU5 = 0       LRU4 = 0       LRU3 = 0       LRU2 = 0       LRU1 = 0       LRU0 = 0
```

```
#
# -----TAG BITS DONT MATCH - MISS-----
# ------Communication with L2------
# Data Cache : Read from L2 <582c7632>
#
#
# 0 594C7615
#
# -----TAG BITS DONT MATCH - MISS-----
# ------Communication with L2------
# Data Cache : Read from L2 <594c7615>
#
#
# 0 621C7600
#
# -----TAG BITS DONT MATCH - MISS-----
# ------Communication with L2------
# Data Cache : Read from L2 <621c7600>
#
#
# 1 111C762C
#
# -----TAG BITS DONT MATCH - MISS-----
# -------------Communication with L2-------------
# Data Cache : Read for Ownership(RFO) from L2 <111c762c>
# Data Cache : Write to L2 <111c762c>
#
#
# 9 0000
#
# ------------------------------Contents of INSTRUCTION CACHE------------------------------
# TRACE Address = 0
# SET NUMBER          = 0
# TAG3 = 0       TAG2 = 0       TAG1 = 0       TAG0 = 0
# STATE3 = INVALID  STATE2 = INVALID  STATE1 = INVALID  STATE0 = INVALID
# LRU3 = 0       LRU2 = 0       LRU1 = 0       LRU0 = 0
# VALID3 = 0     VALID2 = 0     VALID1 = 0     VALID0 = 0
#
# ---INSTRUCTION CACHE statistics---
# Total Number of instruction reads  = 0
# Total Number of instruction writes = 0
# Total Number of instruction hits   = 0
# Total Number of instruction misses = 0
# Denominator cannot be zero (ie, data_miss and data_hit is zero)
# ------------------------------------------------------------------------------Contents of DATA CACHE-------------------------------------------------------------------------------
# TRACE ADDRESS = 0
# SET NUMBER   = 0
```

```
# STATE7 = INVALID  STATE6 = INVALID  STATE5 = INVALID  STATE4 = INVALID  STATE3 = INVALID  STATE2 = INVALID  STATE1 = INVALID  STATE0 = INVALID
# LRU7 = 0       LRU6 = 0       LRU5 = 0       LRU4 = 0       LRU3 = 0       LRU2 = 0       LRU1 = 0       LRU0 = 0
# DIRTY7 = 0     DIRTY6 = 0     DIRTY5 = 0     DIRTY4 = 0     DIRTY3 = 0     DIRTY2 = 0     DIRTY1 = 0     DIRTY0 = 0
# VALID7 = 0     VALID6 = 0     VALID5 = 0     VALID4 = 0     VALID3 = 0     VALID2 = 0     VALID1 = 0     VALID0 = 0
#
# -----DATA CACHE statistics-----
# Total Number of data reads  = 3
# Total Number of data writes = 3
# Total Number of data hits   = 1
# Total Number of data misses = 5
# DATA CACHE HIT ratio = 16.666667
#
#
# Trace filename formatting is incorrect
# -------------------------------Contents of INSTRUCTION CACHE--------------------------------
# TRACE Address = 0
# SET NUMBER          = 0
# TAG3 = 0       TAG2 = 0       TAG1 = 0       TAG0 = 0
# STATE3 = INVALID  STATE2 = INVALID  STATE1 = INVALID  STATE0 = INVALID
# LRU3 = 0       LRU2 = 0       LRU1 = 0       LRU0 = 0
# VALID3 = 0     VALID2 = 0     VALID1 = 0     VALID0 = 0
#
# ---INSTRUCTION CACHE statistics---
# Total Number of instruction reads  = 0
# Total Number of instruction writes = 0
# Total Number of instruction hits   = 0
# Total Number of instruction misses = 0
# Denominator cannot be zero (ie, data_miss and data_hit is zero)
# ------------------------------------------------------------------Contents of DATA CACHE-------------------------------------------------------------------
# TRACE ADDRESS = 0
# SET NUMBER    = 0
# TAG7 = 0       TAG6 = 0       TAG5 = 0       TAG4 = 0       TAG3 = 0       TAG2 = 0       TAG1 = 0       TAG0 = 0
# STATE7 = INVALID  STATE6 = INVALID  STATE5 = INVALID  STATE4 = INVALID  STATE3 = INVALID  STATE2 = INVALID  STATE1 = INVALID  STATE0 = INVALID
# LRU7 = 0       LRU6 = 0       LRU5 = 0       LRU4 = 0       LRU3 = 0       LRU2 = 0       LRU1 = 0       LRU0 = 0
# DIRTY7 = 0     DIRTY6 = 0     DIRTY5 = 0     DIRTY4 = 0     DIRTY3 = 0     DIRTY2 = 0     DIRTY1 = 0     DIRTY0 = 0
# VALID7 = 0     VALID6 = 0     VALID5 = 0     VALID4 = 0     VALID3 = 0     VALID2 = 0     VALID1 = 0     VALID0 = 0
#
# -----DATA CACHE statistics-----
# Total Number of data reads  = 3
# Total Number of data writes = 3
# Total Number of data hits   = 1
# Total Number of data misses = 5
# DATA CACHE HIT ratio = 16.666667
#
# ** Note: $stop    : testbench.sv(57)
#    Time: 125 ns  Iteration: 0  Instance: /SplitL1Cache_tesbench
```

```
# VALID7 = 0     VALID6 = 0     VALID5 = 0     VALID4 = 0     VALID3 = 0     VALID2 = 0     VALID1 = 0     VALID0 = 0
#
# -----DATA CACHE statistics-----
# Total Number of data reads  = 3
# Total Number of data writes = 3
# Total Number of data hits   = 1
# Total Number of data misses = 5
# DATA CACHE HIT ratio = 16.666667
#
#
# Trace filename formatting is incorrect
# -------------------------------Contents of INSTRUCTION CACHE--------------------------------
# TRACE Address = 0
# SET NUMBER          = 0
# TAG3 = 0       TAG2 = 0       TAG1 = 0       TAG0 = 0
# STATE3 = INVALID  STATE2 = INVALID  STATE1 = INVALID  STATE0 = INVALID
# LRU3 = 0       LRU2 = 0       LRU1 = 0       LRU0 = 0
# VALID3 = 0     VALID2 = 0     VALID1 = 0     VALID0 = 0
#
# ---INSTRUCTION CACHE statistics---
# Total Number of instruction reads  = 0
# Total Number of instruction writes = 0
# Total Number of instruction hits   = 0
# Total Number of instruction misses = 0
# Denominator cannot be zero (ie, data_miss and data_hit is zero)
# ------------------------------------------------------------------Contents of DATA CACHE-------------------------------------------------------------------
# TRACE ADDRESS = 0
# SET NUMBER    = 0
# TAG7 = 0       TAG6 = 0       TAG5 = 0       TAG4 = 0       TAG3 = 0       TAG2 = 0       TAG1 = 0       TAG0 = 0
# STATE7 = INVALID  STATE6 = INVALID  STATE5 = INVALID  STATE4 = INVALID  STATE3 = INVALID  STATE2 = INVALID  STATE1 = INVALID  STATE0 = INVALID
# LRU7 = 0       LRU6 = 0       LRU5 = 0       LRU4 = 0       LRU3 = 0       LRU2 = 0       LRU1 = 0       LRU0 = 0
# DIRTY7 = 0     DIRTY6 = 0     DIRTY5 = 0     DIRTY4 = 0     DIRTY3 = 0     DIRTY2 = 0     DIRTY1 = 0     DIRTY0 = 0
# VALID7 = 0     VALID6 = 0     VALID5 = 0     VALID4 = 0     VALID3 = 0     VALID2 = 0     VALID1 = 0     VALID0 = 0
#
# -----DATA CACHE statistics-----
# Total Number of data reads  = 3
# Total Number of data writes = 3
# Total Number of data hits   = 1
# Total Number of data misses = 5
# DATA CACHE HIT ratio = 16.666667
#
# ** Note: $stop    : testbench.sv(57)
#    Time: 125 ns  Iteration: 0  Instance: /SplitL1Cache_tesbench
# Break in Module SplitL1Cache_tesbench at testbench.sv line 57
```

8) Contribution towards the project:

Each one of us made sure that we equally contribute towards this project.
We started working on this project as early as possible so to make sure we achieve expected result. Below are our following contributions.

Common contributions by all:

- L1-L2 communication messages.
- Implementation of two modes in the project.
- Find corner cases to verify project design.
- Collect relevant project related information.
- Active participation in project team meetings every week.

Individual Contribution Towards Project:

- **SUGGU V S S K K L SAI TAGORE:** MESI Implementation, Designing test cases.
- **DIVYA SRI AYLURI**: Data Cache Implementation.
- **RAFATH ACHUGATLA**: Instruction Cache Implementation.
- **SURYA VAMSI TIRUVEEDHULA**: Testbench Coding, LRU Implementation, Documentation.

9) References

- Memory and Cache slides by Prof. Yuchen Huang.
- https://www.geeksforgeeks.org/cache-coherence-protocols-in-multiprocessor-system/
- https://en.wikipedia.org/wiki/Cache_replacement_policies