THE UNIVERSITY OF DA NANG
UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF AVANCED SCIENCE AND TECHNOLOGY

FINAL PROJECT REPORT:

Design and simulation of a split L1 cache for a new 32-bit processor

MEMBERS:
Nguyen The Nhan
Tran Phuoc Dien

Da Nang, 5/12/2024

Contents

I. INTRODUCTION: 2
II. PROJECT SPECFICATIONS AND REQUIREMEMTS 2
III. CACHE STRUCTURE: 4
IV. LRU POLICY 4
V. DESIGN ASSUMPTIONS 5
VI. Test Plan and Result: 6
VII. Appendix: 10

I. INTRODUCTION:

Cache is a type of high-speed memory that stores frequently accessed data and instructions for quick access by the processor. The primary purpose of a cache is to reduce the time it takes to access data from the main memory, which is much slower compared to the cache memory.

Our project involves the design and simulation of a split L1:

- L1 instruction cache: two-way set associative
- L2 data cache: four-way set associative

Both cache consisting of 16K sets of 65-byte lines and use a Least Recently Use (LRU) replacement policy. The cache hierarchy will employ inclusivity. The aims of out project are to optimize cache performance to improve the overall system performance

II. PROJECT SPECFICATIONS AND REQUIREMEMTS

The split L1 cache is designed for a new 32-bit processor:

- The processor is not used in a multicore or multiprocessor configuration and does not need to support cache coherence.
- The L1 instruction cache is 2-way set associative, with 16K sets and 64-byte lines.
- The L1 data cache is 4-way set associative, with 16K sets of 64-byte lines.
- The L1 data cache uses write allocate and is write-back except for the first write to a line which is write-through.
- Both caches employ the Least Recently Used (LRU) replacement policy.
- The cache hierarchy employs inclusivity.
- Both caches are backed by a shared L2 cache.

Our design will be implemented in C++, and the code should maintain and report the following key statistics for each cache upon completion of execution of each trace file:

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

The L1 caches must communicate with the shared L2 cache to maintain inclusivity. To simulate this communication, the following messages (where <address> is a hexadecimal address) should be displayed:

- **Write to L2 <address>:** This operation is used to write back a modified line to L2 upon eviction from the L1 cache.

- **Read from L2 <address>:** This operation is used to obtain the data from L2 on an L1 cache miss.

- **Read for Ownership from L2 <address>:** This operation is used to obtain the data from L2 on an L1 cache write miss.

These specifications ensure that the L1 caches are properly communicating with the shared L2 cache

We will be using two modes:
- **Mode 0:** Display only the required summary of usage statistics and responses to 9s in the trace file.
- **Mode 1:** Display everything from Mode 0, as well as the communication messages to the L2 cache described above.

The simulation reads cache access caches/events from a trace text file, and it can support any trace file provided without recompiling the program so long as the provided file has the correct format

The address will be a hex value. For example:

2 408ed4
0 10019d94
2 408ed8
1 10019d88
2 408edc

N	Semantics
0	Read data request to L1 data cache
1	Write data request to L1 data cache

2	Instruction fetch (a read request to L1 instructions cache
3	Evict command from L2 (for L2 evictions and inclusivity)
8	Clear the cache and reset all state (and statistics) but continue reading trace file
9	Print contents and state of the cache (continue reading traces, don't clear cache)

III. CACHE STRUCTURE:

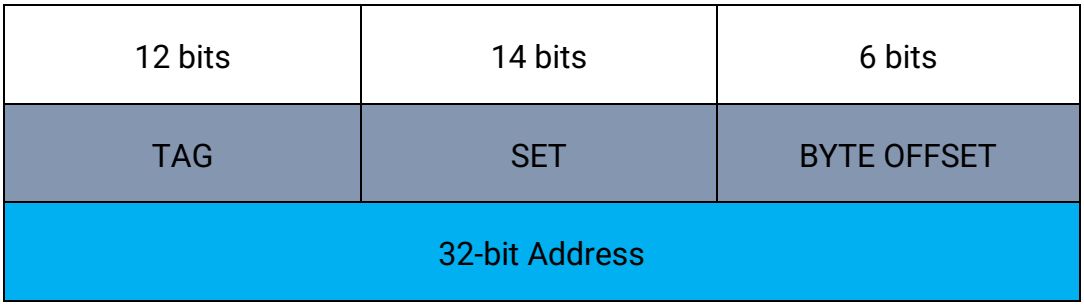
Design and simulate a split L1 cache for a new 32-bit processor:

- Data cache (4-way Set Associative, 16K set, 64-byte lines)
- Instruction cache (2-way Set Associative, 16K set, 64-byte lines)

- Requirement analysis:
  - **32-bit processor** => **32 address bits**.
  - **64-byte lines** => **6 address bits** (2^6) for Byte Select bits.
  - **16K sets** cache is reflected in the **14 address bits** (2^{14}) for index.
  - The number of **tag** bits is 32 - 6 - 14 = **12 bits**

Summary:

- 12 bits tag
- 14 bits set
- 6 bits byte



IV. LRU POLICY

LRU (Least Recently Used) policy is a caching strategy where the least recently used item gets evicted first. This means each item needs to be identified/organized in terms of its usage.

In our case, we are assigning the LRU default to 0 and the most recently used (MRU) item to 4.

For example, let's say item C has just been accessed:

	A	B	C	D
LRU value	1	2	3	4

Original

Since C just became MRU, it will be assigned LRU value 11. For every item that is valued higher than C, we will shift its corresponding LRU value left once (decrement). For every item that is valued less than C, its LRU value will not change.

	A	B	C	D
LRU value	1	2	4	3

After Accessing C

Items D were valued higher than C and have been decremented. A and B were valued less than C so they remain unchanged.

Another example, let's say we need to evict the LRU in the original lineup to add Item F:

	F	B	C	D
LRU value	4	1	3	2

After Adding Item F

All the other LRU value will be decremented, and Item F becomes 4 (MRU).

v. DESIGN ASSUMPTIONS

- Init system:
    - The L1 cache is empty
    - All operations in beginning is cache miss
- Reading data
    - Processor first checks the L1 cache, is the line is not found, a copy is retrieved from L2 and written to the L1 cache
    - Change state to valid if the previous state is invalid

- Write data:
    - Write hit:
        - Write back old line to L2 and then turn on dirty bit
    - Write miss (It will read for ownership from L2)
        - Processor first check the empty line in L1 cache. Write through at the first write
        - If no empty line. Check the line has LRU value.
            - If it has dirty bit. Write back that line to L2 cache
            - Change the state to Valid (because first write is write through)
- Evict data from L2:
    - Check the same address in L1 cache, evict it and change state to Invalid but not change LRU value

VI. Test Plan and Result:

We have created 3 test cases for our cache design:

Test 1: Data cache miss/hit of a particular set (0x2841)

```
0   003A1045
0   012A1062
1   030A1079
1   392A106B
0   030A106F
0   012A1076
1   030A1078
1   452A107A
9
```

Test 2: Evict data of particular set (0x2841)

```
3   012A105A
0   516A1056
0   193A1055
0   120A1057
3   193A1054
1   810A105D
9
```

Test 3: Test write data in set **0x3780** and **0x3781**. Read instructions in sets **0x3781** and **0x3782**. Clear operation at the end

```
1   562DE000
1   213DE001
1   213DE040
2   562DE045
2   562DE087
```

2 203DE084
9
8
9

All test cases are saved in only file “test.txt”

Result: We have used mode 1 in program to present both statistics and communicate with L2 cache. The statistics continue to calculate after each test case instead of recalculating from the beginning.

- **Test 1:**

```
                            Simulation of an L1 split cache of a 32-bit Processor

                             Final Project - Team Nguyen The Nhan and Tran Phuoc Dien

  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Mode 0: Display only the required summary of usage statistics and responses to 9s in the trace file.
  Mode 1: Display everything from mode 1, as well as the communication messages to the L2 cache.

  Please enter the mode number you'd like to select (0,1): 1

Operation: 0, Address: 3a1045

        [Data Read Miss] Read from L2: L1 cache miss, obtain data from L2 3a1045

Operation: 0, Address: 12a1062

        [Data Read Miss] Read from L2: L1 cache miss, obtain data from L2 12a1062

Operation: 1, Address: 30a1079

        [Data Write Miss] Read for Ownership from L2 30a1079

        [Data Write Through] Write to L2: we have a data Cache Miss 30a1079

Operation: 1, Address: 392a106b

        [Data Write Miss] Read for Ownership from L2 392a106b

        [Data Write Through] Write to L2: we have a data Cache Miss 392a106b

Operation: 0, Address: 30a106f

Operation: 0, Address: 12a1076

Operation: 1, Address: 30a1078

        [Data Write back] Write to L2: We have Data Cache Hit 30a1078

Operation: 1, Address: 452a107a

        [Data Write Miss] Read for Ownership from L2 452a107a

        [Data Write back] Write to L2: we have a data Cache Miss 3a1045

Operation: 9: Print
```

```
        ** KEY CACHE USAGE STATISTICS **

        -- DATA CACHE STATISTICS --
 number of Cache Reads:         4
 number of Cache Writes:        4
 number of Cache Hits:          3
 number of Cache Misses:        5
 Cache Hit Ratio:               0.375
 Cache Hit Ratio percentage:    37.5 %

 More information in valid line:
Line valid : 452A107A   |        Way : 0        |       Set :2841      |       State : V      |       LRU bit : 4    |       Tag : 452
Line valid : 012A1076   |        Way : 1        |       Set :2841      |       State : V      |       LRU bit : 2    |       Tag : 012
Line valid : 030A1078   |        Way : 2        |       Set :2841      |       State : D      |       LRU bit : 3    |       Tag : 030
Line valid : 392A106B   |        Way : 3        |       Set :2841      |       State : V      |       LRU bit : 1    |       Tag : 392

       There are 4 valid lines
```

- **Test case 2:**

```
 The cache instruction was not used/not operated on
Operation: 3, Address: 12A105A

Operation: 0, Address: 516A1056

        [Data Read Miss] Read from L2: L1 cache miss, obtain data from L2 516A1056

Operation: 0, Address: 193A1055

        [Data Read Miss] Read from L2: L1 cache miss, obtain data from L2 193A1055

Operation: 0, Address: 120A1057

        [Data Read Miss] Read from L2: L1 cache miss, obtain data from L2 120A1057

        [Data write back modified line] write to L2: we have a data cache miss: 30A1078

Operation: 3, Address: 193A1054

Operation: 1, Address: 810A105D

        [Data Write Miss] Read for Ownership from L2 810A105D

        [Data Write Through] Write to L2: we have a data Cache Miss 810A105D

Operation: 9: Print
```

```
        ** KEY CACHE USAGE STATISTICS **

        -- DATA CACHE STATISTICS --
  number of Cache Reads:         7
  number of Cache Writes:        5
  number of Cache Hits:          3
  number of Cache Misses:        9
  Cache Hit Ratio:               0.25
  Cache Hit Ratio percentage:    25 %

  More information in valid line:
 Line valid : 452A107A   |       Way : 0        |       Set :2841      |       State : V      |       LRU bit : 1    |       Tag : 452
 Line valid : 516A1056   |       Way : 1        |       Set :2841      |       State : V      |       LRU bit : 2    |       Tag : 516
 Line valid : 120A1057   |       Way : 2        |       Set :2841      |       State : V      |       LRU bit : 3    |       Tag : 120
 Line valid : 810A105D   |       Way : 3        |       Set :2841      |       State : V      |       LRU bit : 4    |       Tag : 810

        There are 4 valid lines
```

- **Test case 3:**

```
 The cache instruction was not used/not operated on
Operation: 1, Address: 562DE000

        [Data Write Miss] Read for Ownership from L2 562DE000

        [Data Write Through] Write to L2: we have a data Cache Miss 562DE000

Operation: 1, Address: 213DE001

        [Data Write Miss] Read for Ownership from L2 213DE001

        [Data Write Through] Write to L2: we have a data Cache Miss 213DE001

Operation: 1, Address: 213DE040

        [Data Write Miss] Read for Ownership from L2 213DE040

        [Data Write Through] Write to L2: we have a data Cache Miss 213DE040

Operation: 2, Address: 562DE045

 [Instruction Read Miss] Read from L2: L1 cache miss, obtain data from L2 562DE045

Operation: 2, Address: 562DE087

 [Instruction Read Miss] Read from L2: L1 cache miss, obtain data from L2 562DE087

Operation: 2, Address: 203DE084

 [Instruction Read Miss] Read from L2: L1 cache miss, obtain data from L2 203DE084

Operation: 9: Print
```

```
        ** KEY CACHE USAGE STATISTICS **

        -- DATA CACHE STATISTICS --
 number of Cache Reads:         7
 number of Cache Writes:        8
 number of Cache Hits:          3
 number of Cache Misses:        12
 Cache Hit Ratio:               0.2
 Cache Hit Ratio percentage:    20 %

 More information in valid line:
Line valid : 452A107A   |       Way : 0        |       Set :2841      |       State : V      |       LRU bit : 1    |       Tag : 452
Line valid : 516A1056   |       Way : 1        |       Set :2841      |       State : V      |       LRU bit : 2    |       Tag : 516
Line valid : 120A1057   |       Way : 2        |       Set :2841      |       State : V      |       LRU bit : 3    |       Tag : 120
Line valid : 810A105D   |       Way : 3        |       Set :2841      |       State : V      |       LRU bit : 4    |       Tag : 810
Line valid : 562DE000   |       Way : 0        |       Set :3780      |       State : V      |       LRU bit : 3    |       Tag : 562
Line valid : 213DE001   |       Way : 1        |       Set :3780      |       State : V      |       LRU bit : 4    |       Tag : 213
Line valid : 213DE040   |       Way : 0        |       Set :3781      |       State : V      |       LRU bit : 4    |       Tag : 213

       There are 7 valid lines

        -- INSTRUCTION CACHE STATISTICS --
 number of Cache Reads:         3
 number of Cache Hits:          0
 number of Cache Misses:        3
 Cache Hit Ratio:               0
 Cache Hit Ratio percentage:    0 %

 More information in valid line:
 Line valid : 562DE045  |       Way : 0        |       Set :3781      |       State : V      |       LRU bit : 2    |       Tag : 562
 Line valid : 562DE087  |       Way : 0        |       Set :3782      |       State : V      |       LRU bit : 1    |       Tag : 562
 Line valid : 203DE084  |       Way : 1        |       Set :3782      |       State : V      |       LRU bit : 2    |       Tag : 203

       There are 3 valid lines
Operation: 8: Clear all cache


        Clear the cache to the initial state and reset the statistics

Operation: 9: Print
```

Our results are correct when compared with our handwritten test (you can see in the Appendix

VII. Appendix:

Test 1: Data cache miss/hit of a particular set (0x2841)

0 003A1045
0 012A1062
1 030A1079
1 392A106B
0 030A106F
0 012A1076
1 030A1078
1 452A107A
9

Step 1: 0 003A1045 (Read Miss)

➔ Load to way 0 with 003

Set	Cache content table				
0x0x2841	Way	0	1	2	3
	Tag	003			
	State	V	I	I	I

	Data	instructions
Reads	1	0
Write		
Hit	0	0

	LRU value	4	0	0	0

Misses	1	0
Hit Ratio	0	0

Step 2: 0 012A1062 (Read Miss)

➔ Load to way 1 with 012

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	003	012		
	State	V	V	I	I
	LRU value	3	4	0	0

	Data	instructions
Reads	2	0
Write		
Hit	0	0
Misses	2	0
Hit Ratio	0	0

Step 3: 1 030A1079 (Write Miss)

➔ Load to way 2 with 030

➔ First write is write through

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	003	012	030	
	State	V	V	V	I
	LRU value	2	3	4	0

	Data	instructions
Reads	2	0
Write	1	
Hit	0	0
Misses	3	0
Hit Ratio	0	0

Step 4: 1 392A106B (Write Miss)

➔ Load to way 3 with 392

➔ First write is write through

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	003	012	030	392
	State	V	V	V	V
	LRU value	1	2	3	4

	Data	instructions
Reads	2	0
Write	2	
Hit	0	0
Misses	4	0
Hit Ratio	0	0

Step 5: 0 030A106F (Read Hit)

➔ Read hit does not change state

Set	Cache content table

	Data	instructions

0x2841	Way	0	1	2	3
	Tag	003	012	030	392
	State	V	V	V	V
	LRU value	1	2	4	3

Reads	3	0
Write	2	
Hit	1	0
Misses	4	0
Hit Ratio	0.2	0

Step 6: 0 012A1076 (Read Hit)

➔ Read hit does not change state

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	003	012	030	392
	State	V	V	V	V
	LRU value	1	4	3	2

	Data	instructions
Reads	4	0
Write	2	
Hit	2	0
Misses	4	0
Hit Ratio	0.33	0

Step 7: 1 030A1078 (Write Hit)

➔ Write back so set Dirty bit

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	003	012	030	392
	State	V	V	D	V
	LRU value	1	3	4	2

	Data	instructions
Reads	4	0
Write	3	
Hit	3	0
Misses	4	0
Hit Ratio	0.43	0

Step 8: 1 452A107A (Write Miss - conflict)

➔ Write back way 0 to L2 to for evict L1. Because this is the first write so this is write through

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	452	012	030	392
	State	V	V	D	V
	LRU value	4	2	3	1

	Data	instructions
Reads	4	0
Write	4	
Hit	3	0
Misses	5	0
Hit Ratio	0.375	0

Test 2: Evict data of particular set (0x2841)

3 012A105A
0 516A1056
0 193A1055
0 120A1057
3 193A1054
1 810A105D
9

Step 1: 3 012A105A (Evict from L2)

➔ Evict way 1 and reset State to Invalid but do not need reset LRU value

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	452		030	392
	State	V	I	D	V
	LRU value	4	2	3	1

	Data	instructions
Reads	4	0
Write	4	
Hit	3	0
Misses	5	0
Hit Ratio	0.375	0

Step 2: 0 516A1056 (Read miss)

➔ Load to way 1

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	452	516	030	392

	Data	instructions
Reads	5	0
Write	4	

	State	V	V	D	V
	LRU value	3	4	2	1

Hit	3	0
Misses	6	0
Hit Ratio	0.33	0

Step 3: 0 193A1055 (Read miss - conflict)

➔ Way 3 is evicted. Then load to way 3

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	452	516	030	193
	State	V	V	D	V
	LRU value	2	3	1	4

	Data	instructions
Reads	6	0
Write	4	
Hit	3	0
Misses	7	0
Hit Ratio	0.3	0

Step 4: 0 120A1057 (Read miss - conflict)

➔ Way 2 is evicted. However, way 2 have dirty bit, so way 2 is written back to L2 before implementing read new block from L2.

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	452	516	120	193
	State	V	V	V	V
	LRU value	1	2	4	3

	Data	instructions
Reads	7	0
Write	4	
Hit	3	0
Misses	8	0
Hit Ratio	0.27	0

Step 5: 3 193A1054 (Evict way 3)

➔ Evict way 3 and reset State to Invalid but do not need reset LRU value

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	452	516	120	
	State	V	V	V	I
	LRU value	1	2	4	3

	Data	instructions
Reads	7	0
Write	4	
Hit	3	0
Misses	8	0
Hit Ratio	0.27	0

Step 6: 1 810A105D (Write miss)

➔ Write to way 3. This is the first way so this is way through

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	452	516	120	810
	State	V	V	V	V
	LRU value	1	2	3	4

	Data	instructions
Reads	7	0
Write	5	
Hit	3	0
Misses	9	0
Hit Ratio	0.25	0

Test 3: Test write data in set **0x3780** and **0x3781**. Read instructions in sets **0x3781** and **0x3782**. Clear operation at the end

1 562DE000
1 213DE001
1 213DE040
2 562DE045
2 562DE087
2 203DE084
8

- Data cache:

Step 1: 1 562DE000 (Write miss)

➔ Write through to way 0 in set 3780 with ta 562

Set	Cache content table				
0x3780	Way	0	1	2	3
	Tag	562			
	State	V	I	I	I
	LRU value	4	0	0	0

	Data	instructions
Reads	7	0
Write	6	
Hit	3	0
Misses	10	0
Hit Ratio	0.23	0

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	452	516	120	810
	State	V	V	V	V
	LRU value	1	2	3	4

Step 2: 1 213DE001(Write miss)

➔ Write through way 1 in the set 3780 with tag 213

Set	Cache content table				
0x3780	Way	0	1	2	3
	Tag	562	213		
	State	V	V	I	I
	LRU value	3	4	0	0

	Data	instructions
Reads	7	0
Write	7	
Hit	3	0
Misses	11	0
Hit Ratio	0.21	0

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	452	516	120	810
	State	V	V	V	V
	LRU value	1	2	3	4

Step 3: 1 213DE040(Write miss in another set)

➔ Write through to way 1 in set 3781 with tag 213

Set	Cache content table				
0x3780	Way	0	1	2	3
	Tag	562	213		
	State	V	V	I	I
	LRU value	3	4	0	0

	Data	instructions
Reads	7	0
Write	8	
Hit	3	0
Misses	12	0
Hit Ratio	0.2	0

Set	Cache content table				
0x3781	Way	0	1	2	3
	Tag	213			
	State	V	I	I	I
	LRU value	4	0	0	0

Set	Cache content table				
0x2841	Way	0	1	2	3
	Tag	452	516	120	810
	State	V	V	V	V
	LRU value	1	2	3	4

- **Instruction cache:**

Step 4: 2 562DE045

➔ Load to way 0 with tag 562 in set 0x3781

Set	Cache content table		
0x3781	Way	0	1
	Tag	562	0
	State	V	I
	LRU value	2	0

	Data	instructions
Reads	7	1
Write	8	
Hit	3	0
Misses	12	1
Hit Ratio	0.2	0

Step 5: 2 562DE087

➔ Load to way 0 with tag 562 in another set (0x3782)

Set	Cache content table		
0x3781	Way	0	1
	Tag	562	
	State	V	I
	LRU value	2	0

	Data	instructions
Reads	7	2
Write	8	
Hit	3	0
Misses	12	2
Hit Ratio	0.2	0

Set	Cache content table		
0x3782	Way	0	1
	Tag	562	
	State	V	I
	LRU value	2	0

Step 6: 2 203DE084

➔ Load to way 1 with tag 203 in set 3782

Set	Cache content table		
0x3781	Way	0	1
	Tag	562	
	State	V	I
	LRU value	2	1

	Data	instructions
Reads	7	3
Write	8	
Hit	3	0
Misses	12	3
Hit Ratio	0.2	0

Set	Cache content table		
0x3782	Way	0	1
	Tag	562	203
	State	V	V
	LRU value	1	2

Step 7: Clear for both cache and reset statistics

	Data	instructions
Reads	0	0
Write	0	0
Hit	0	0
Misses	0	0
Hit Ratio	0	0