ECE485

Cash Controller Design Project

Tachchai Buraparatana

Jinho Park

Anthony Romano

Hoa Quach

Project Report

1. Objective

Design a 4-way set associative L2 Cache for a 32-bit processor and implement the design using verilog HDL.

Simulate random, LRU and Pseudo-LRU replacement policies and measure the performance of each policies

1. Requirement Specification

|  |  |
| --- | --- |
|  | Requirements |
| Cache | Line Size MUST BE greater than 4 bytes (32 bits) |
| MUST implement write-back and write allocate policies |
| MUST implement all of the following replacement policies   1. Random 2. LRU 3. Pseudo-LRU |
| Testbench | Must read memory access from text file |
| Must Keep track of following key parameters   1. Total number of memory references 2. Number of reads 3. Number of writes 4. Number of cache hits 5. Number of cache misses 6. Hit ratio |
| Should emulate burst mode in memory |

1. Assumptions Made
   1. All memory references are world-aligned and the processor addresses items by word.
   2. Full range of 32 bit address space is accessed.
   3. Write-through cache. All the memory access from L1 goes through L2.
   4. Burst mode is ensured to be completed without any interrupt.
2. Block Diagram



1. Module Description

|  |  |
| --- | --- |
| Module Name | L1Cache |
| Input Ports | stall, debug |
| Output Ports | addr[31:0], addrstb, we |
| Bidirectional Ports | data[31:0] |
| Description | Emulates the L1 cache side interface by reading in memory references from the trace file and make read/write request to L2 cache. |
| Interface mechanism with L2Cache | This module outputs 32 bit memory address to addr[31:0] and assert (we=0) or de-assert (we=1) the active low we signal according to the type of the memory reference. When valid address, we, and data (in case of memory write request) are available on the bus, it toggles addrstb, which triggers L2 cache to latch in the address, we and data.  L2 cache assert active high stall (stall = 1) when L2 is processing request from L1 cache. L1 cache stalls until stall is de-asserted by L2. |

|  |  |
| --- | --- |
| Module Name | L2Cache |
| Input Ports | Stb  we\_L1, addrstb\_L1, addr\_L1 |
| Output Ports | stall we\_MEM, addr\_MEM, addrstb\_MEM |
| Bidirectional Ports | data\_L1[31:0], data\_MEM[63:0] |
| Description | Processes cache request from L1Cache and manage L2 cache.  Employs one of the replacement policies.  Keep track of test statistics. |

|  |  |
| --- | --- |
| Module Name | MainMemory |
| Input Ports | we, addrstb, addr[31:0] |
| Output Ports | stb |
| Bidirectional Ports | data[63:0] |
| Description | Emulates the main memory side interface by processing memory access from L2 cache.  Supports burst read. |
| Interface mechanism with L2Cache | When addrstb is toggled, this module latches the address and we values from addr[31:0] and we. L2 stalls until stb is toggled. MainMemory toggles stb as it bursts out each chunk of data to data[63:0] bus or toggles stb when process is done. |

|  |  |
| --- | --- |
| Module Name | L2CacheTestBench |
| Input Ports | N/A |
| Output Ports | N/A |
| Description | Instantiate L1Cache, L2Cache, and MainMemory modules.  Provide interface to debug and rep port of L2 cache. |
| Interface mechanism with L2Cache | Assert debug to display debug flags in the codes.  Assign appropriate values of replacement policy to rep[1:0] to enable a replacement policy. |

1. L2 cache Design Decisions



* 1. Cache line size is 64 bytes because the graph above from class lecture slides show that the cache line size of 64 byte gives the best performance and current processors use cache size of 32 – 128 bytes.
  2. Main memory’s burst length is eight to support cache line fill of 64-byte-long line over 64 bit data bus.
  3. The number of lines is 64 to ensure number of replacement reasonably high. When the number of liens are high, replacement does not happen often enough to yield data to compare. When the numbers are too small, cache miss rate goes too high.

1. Algorithm



1. Testing
   1. Cache Initiation
      1. Cache Parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **# of ways** | **# of lines** | **# of words** | **Burst length** | **Replacement** |
| 2 | 2 | 4 | 2 | N/A |

* + 1. Description:

Initiate cache\_data array so that the **MSB** of each word represents the **line number** and the **4 LSB’s** represent the **word number** in the line.

* + 1. Result:

SIM: ----------------------------------------------

SIM: Way: 0

SIM: Index: 0

SIM: Word: 0: Content: 00000000

SIM: Word: 1: Content: 00000001

SIM: Word: 2: Content: 00000002

SIM: Word: 3: Content: 00000003

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

SIM: ----------------------------------------------

SIM: Way: 1

SIM: Index: 0

SIM: Word: 0: Content: 00000000

SIM: Word: 1: Content: 00000001

SIM: Word: 2: Content: 00000002

SIM: Word: 3: Content: 00000003

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

SIM: =================================================

* 1. Cache Line Fill
     1. Fill cache line 0 by causing a cache miss on line 0 with address 0xFFFc0000.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | F | | | | C | | | | 0 | | | | 0 | | | | 0 | | | | | 0 | | | | | |
| 16383 | | | | | | | | | | | | | | 0 | | | | | | | | | | | | | 0 | | | | | | 0 | |

Dram output increment the address by 4(0x00000004) after each burst, so that each item stored represent the starting address of the 32 bit word for debugging purpose.

* + 1. Expected Result of cache line fill:

|  |  |  |  |
| --- | --- | --- | --- |
| Word 0 | Word 1 | Word 2 | Word 3 |
| FFFF0000 | FFFF0004 | FFFF0008 | FFFF000C |

* + 1. Result:

SIM: command:2 address:fffc0**000**([17:6])

SIM: Tag: 16383 Index: 0 Word 0

SIM: L2 MISS

SIM: Task : Cache\_Line\_Fill

SIM: Way: 0, Tag: 16383, Line: 0

SIM: Way: 0

SIM: Index: 0

SIM: Word: 0: Content: **fffc0000**

SIM: Word: 1: Content: **fffc0004**

SIM: Word: 2: Content: **fffc0008**

SIM: Word: 3: Content: **fffc000c**

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

SIM: Way: 1

SIM: Index: 0

SIM: Word: 0: Content: 00000000

SIM: Word: 1: Content: 00000001

SIM: Word: 2: Content: 00000002

SIM: Word: 3: Content: 00000003

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

* 1. Cache Look Up (Hit)
     1. After fill the cache line as in b. with address FFFC0000 (Tag = 16383, Index = 0, Word = 0). Access FFFC0004, FFFC0008, FFFC000C (Tag = 16383, Index =1, Word = 1~3) and see if each addresses that match the tag and index gives cache hit.

* + 1. Result

SIM: command:2 address:fffc0004

SIM: Tag: 16383 Index: 0 Word 1

SIM: **L2 HIT**

SIM: L2 outputs

SIM: Way: 0, Index: 0 Word 1

SIM: **Content: fffc0004**

SIM: ======================================================

SIM: command:2 address:fffc0008

SIM: Tag: 16383 Index: 0 Word 2

SIM: **L2 HIT**

SIM: L2 outputs

SIM: Way: 0, Index: 0 Word 2

SIM: **Content: fffc0008**

SIM: ======================================================

SIM: command:2 address:fffc000c

SIM: Tag: 16383 Index: 0 Word 3

SIM: **L2 HIT**

SIM: L2 outputs

SIM: Way: 0, Index: 0 Word 3

SIM: **Content: fffc000c**

======================================================

* 1. Cache Look Up (Miss)
     1. After fill the cache line as in b. with address FFFC0000 (Tag = 16383, Index = 0, Word = 0). Access FFFC0040 (Tag = 16383, Index =1, Word = 0) and see if it gives miss and result in cache line fill in line 1.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | F | | | | C | | | | 0 | | | | 0 | | | | 4 | | | | | 0 | | | | | |
| 16383 | | | | | | | | | | | | | | **1** | | | | | | | | | | | | | 0 | | | | | | 0 | |

* + 1. Result:

SIM: ======================================================

SIM: command:2 address:fffc0040

SIM: Tag: 16383 **Index: 1** Word 0

SIM: **L2 MISS**

SIM: Task : **Cache\_Line\_Fill**

SIM: Way: 0

SIM: Index: 0

SIM: Word: 0: Content: fffc0000

SIM: Word: 1: Content: fffc0004

SIM: Word: 2: Content: fffc0008

SIM: Word: 3: Content: fffc000c

**SIM: Index: 1**

**SIM: Word: 0: Content: fffc0040**

**SIM: Word: 1: Content: fffc0044**

**SIM: Word: 2: Content: fffc0048**

**SIM: Word: 3: Content: fffc004c**

SIM: Way: 1

SIM: Index: 0

SIM: Word: 0: Content: 00000000

SIM: Word: 1: Content: 00000001

SIM: Word: 2: Content: 00000002

SIM: Word: 3: Content: 00000003

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

SIM:

* 1. Replacement
     1. Random
        1. After fill Index 0 of way 0 and way 1 and cause read miss on the same line and see if replacement occurs on either one of the way.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | F | | | | C | | | | 0 | | | | 0 | | | | 0 | | | | | 0 | | | | | |
| 16383 | | | | | | | | | | | | | | **0** | | | | | | | | | | | | | 0 | | | | | | 0 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | F | | | | 8 | | | | 0 | | | | 0 | | | | 0 | | | | | 0 | | | | | |
| 16382 | | | | | | | | | | | | | | **1** | | | | | | | | | | | | | 0 | | | | | | 0 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | F | | | | 4 | | | | 0 | | | | 0 | | | | 0 | | | | | 0 | | | | | |
| 16381 | | | | | | | | | | | | | | **0** | | | | | | | | | | | | | 0 | | | | | | 0 | |

* + - 1. Result

SIM: =================================================

SIM: command:2 address:fff80000

SIM: Tag: **16382** Index: 0 Word 0

SIM: L2 MISS

SIM: Task : Cache\_Line\_Fill

SIM: Way: 1, Tag: 16382, Line: 0

SIM: Miss:2

**SIM: Way: 0**

**SIM: Index: 0**

**SIM: Word: 0: Content: fffc0000**

**SIM: Word: 1: Content: fffc0004**

**SIM: Word: 2: Content: fffc0008**

**SIM: Word: 3: Content: fffc000c**

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

**SIM: Way: 1**

**SIM: Index: 0**

**SIM: Word: 0: Content: fff80000**

**SIM: Word: 1: Content: fff80004**

**SIM: Word: 2: Content: fff80008**

**SIM: Word: 3: Content: fff8000c**

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

SIM: Data from L2: fff80000

SIM: =================================================

SIM: command:2 address:fff40000

SIM: Tag: **16381** Index: 0 Word 0

SIM: L2 MISS

**SIM: Task: Replacement\_Way\_Lookup\_Random**

**SIM: Replace Way: 0**

SIM: Task : Cache\_Line\_Fill

SIM: Way: 0, Tag: 16381, Line: 0

SIM: Miss:3

**SIM: Way: 0**

**SIM: Index: 0**

**SIM: Word: 0: Content: fff40000**

**SIM: Word: 1: Content: fff40004**

**SIM: Word: 2: Content: fff40008**

**SIM: Word: 3: Content: fff4000c**

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

**SIM: Way: 1**

**SIM: Index: 0**

**SIM: Word: 0: Content: fff80000**

**SIM: Word: 1: Content: fff80004**

**SIM: Word: 2: Content: fff80008**

**SIM: Word: 3: Content: fff8000c**

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

SIM: Data from L2: fff40000

SIM: Hit:0

SIM: Miss:3

* + 1. PLRU
       1. Parameters: 4 Ways, 2 Lines, 4 Words, Burst Length = 2
       2. Steps
          1. fill Index 0 of way 0 - way 3 (PLRU = 000 Way 0)
          2. access index 0 of way 0 (PLRU = 110 Way 2)
          3. access index 0 of way 3 (PLRU = 010 Way 1)
          4. cause read miss on index 0 and see if way 1 is replaced.
       3. Test Vectors

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | F | | | | C | | | | 0 | | | | 0 | | | | 0 | | | | | 0 | | | | | |
| 16383 | | | | | | | | | | | | | | **0** | | | | | | | | | | | | | 0 | | | | | | 0 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | F | | | | 8 | | | | 0 | | | | 0 | | | | 0 | | | | | 0 | | | | | |
| 16382 | | | | | | | | | | | | | | **1** | | | | | | | | | | | | | 0 | | | | | | 0 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | F | | | | 4 | | | | 0 | | | | 0 | | | | 0 | | | | | 0 | | | | | |
| 16381 | | | | | | | | | | | | | | **0** | | | | | | | | | | | | | 0 | | | | | | 0 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | F | | | | 0 | | | | 0 | | | | 0 | | | | 0 | | | | | 0 | | | | | |
| 16380 | | | | | | | | | | | | | | **0** | | | | | | | | | | | | | 0 | | | | | | 0 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | E | | | | C | | | | 0 | | | | 0 | | | | 0 | | | | | 0 | | | | | |
| 16379 | | | | | | | | | | | | | | **0** | | | | | | | | | | | | | 0 | | | | | | 0 | |

* + - 1. RESULT

SIM:

===========================================================

SIM: command:2 address:fff00000

SIM: Tag: 16380 Index: 0 Word 0

SIM: L2 MISS

SIM: Task : Cache\_Line\_Fill

SIM: Way: 3, Tag: 16380, Line: 0

SIM: Way: 0

SIM: Index: 0

SIM: Word: 0: Content: fffc0000

SIM: Word: 1: Content: fffc0004

SIM: Word: 2: Content: fffc0008

SIM: Word: 3: Content: fffc000c

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

SIM: Way: 1

SIM: Index: 0

SIM: Word: 0: Content: fff80000

SIM: Word: 1: Content: fff80004

SIM: Word: 2: Content: fff80008

SIM: Word: 3: Content: fff8000c

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

SIM: Way: 2

SIM: Index: 0

SIM: Word: 0: Content: fff40000

SIM: Word: 1: Content: fff40004

SIM: Word: 2: Content: fff40008

SIM: Word: 3: Content: fff4000c

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

SIM: Way: 3

SIM: Index: 0

SIM: Word: 0: Content: fff00000

SIM: Word: 1: Content: fff00004

SIM: Word: 2: Content: fff00008

SIM: Word: 3: Content: fff0000c

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

SIM: Data from L2: fff00000

SIM: ===========================================================

SIM: command:2 address:fffc0000

SIM: Tag: 16383 Index: 0 Word 0

**SIM: L2 HIT**

SIM: Data from L2: fffc0000

SIM: ===========================================================

SIM: command:2 address:fff00000

SIM: Tag: 16380 Index: 0 Word 0

**SIM: L2 HIT**

SIM: Data from L2: fff00000

SIM: ===========================================================

SIM: command:2 address:ffec0000

SIM: Tag: 16379 Index: 0 Word 0

SIM: L2 MISS

**SIM: Task: Replacement\_Way\_Lookup\_PLRU**

**SIM: Replace Way: 1**

SIM: Task : Cache\_Line\_Fill

SIM: Way: 1, Tag: 16379, Line: 0

* + 1. LRU
       1. After fill Index 0 of way 0 - way 3 (LRU = Way 0)
       2. access index 0 of way 0 (LRU = Way 1)
       3. access index 0 of way 1 (LRU = Way 2)
       4. cause read miss on index 0 and see if way **2** is replaced.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | F | | | | C | | | | 0 | | | | 0 | | | | 0 | | | | | 0 | | | | | |
| 16383 | | | | | | | | | | | | | | **0** | | | | | | | | | | | | | 0 | | | | | | 0 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | F | | | | 8 | | | | 0 | | | | 0 | | | | 0 | | | | | 0 | | | | | |
| 16382 | | | | | | | | | | | | | | **1** | | | | | | | | | | | | | 0 | | | | | | 0 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | F | | | | 4 | | | | 0 | | | | 0 | | | | 0 | | | | | 0 | | | | | |
| 16381 | | | | | | | | | | | | | | **0** | | | | | | | | | | | | | 0 | | | | | | 0 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | F | | | | 0 | | | | 0 | | | | 0 | | | | 0 | | | | | 0 | | | | | |
| 16380 | | | | | | | | | | | | | | **0** | | | | | | | | | | | | | 0 | | | | | | 0 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tag[31:18] | | | | | | | | | | | | | | Index[17:6] | | | | | | | | | | | | Word  [5:4] | | | | | 00 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  | |  |
| F | | | | F | | | | E | | | | C | | | | 0 | | | | 0 | | | | 0 | | | | | 0 | | | | | |
| 16379 | | | | | | | | | | | | | | **0** | | | | | | | | | | | | | 0 | | | | | | 0 | |

2 fffc0000

2 fff80000

2 fff40000

2 fff00000

2 fffc0000

2 fff80000

2 ffec0000

* + - 1. RESULT

SIM: =================================================

SIM: command:2 address:fffc0000

SIM: Tag: 16383 Index: 0 Word 0

SIM: L2 MISS

SIM: Task : Cache\_Line\_Fill

SIM: Way: 0, Tag: 16383, Line: 0

SIM: Data from L2: fffc0000

SIM: =================================================

SIM: command:2 address:fff80000

SIM: Tag: 16382 Index: 0 Word 0

SIM: L2 MISS

SIM: Task : Cache\_Line\_Fill

SIM: Way: 1, Tag: 16382, Line: 0

SIM: Data from L2: fff80000

SIM: =================================================

SIM: command:2 address:fff40000

SIM: Tag: 16381 Index: 0 Word 0

SIM: L2 MISS

SIM: Task : Cache\_Line\_Fill

SIM: Way: 2, Tag: 16381, Line: 0

SIM: Data from L2: fff40000

SIM: =================================================

SIM: command:2 address:fff00000

SIM: Tag: 16380 Index: 0 Word 0

SIM: L2 MISS

SIM: Task : Cache\_Line\_Fill

SIM: Way: 3, Tag: 16380, Line: 0

SIM: cache\_lru[way=0][line=0]: 0

SIM: cache\_lru[way=1][line=0]: 1

SIM: cache\_lru[way=2][line=0]: 2

SIM: cache\_lru[way=3][line=0]: 3

SIM: Data from L2: fff00000

SIM:

=================================================SIM: command:2 address:fffc0000

SIM: Tag: 16383 Index: 0 Word 0

SIM: Task: Look\_For\_Match

SIM: Matching Way: 0

SIM: L2 HIT

SIM: cache\_lru[way=0][line=0]: 3

SIM: cache\_lru[way=1][line=0]: 0

SIM: cache\_lru[way=2][line=0]: 1

SIM: cache\_lru[way=3][line=0]: 2

SIM: Data from L2: fffc0000

SIM: =================================================

SIM: command:2 address:fff80000

SIM: Tag: 16382 Index: 0 Word 0

SIM: Task: Look\_For\_Match

SIM: Matching Way: 1

SIM: L2 HIT

SIM: cache\_lru[way=0][line=0]: 2

SIM: cache\_lru[way=1][line=0]: 3

SIM: cache\_lru[way=2][line=0]: 0

SIM: cache\_lru[way=3][line=0]: 1

SIM: Data from L2: fff80000

SIM: =================================================

SIM: command:2 address:**ffec0000**

SIM: Tag: 16379 Index: 0 Word 0

SIM: L2 MISS

SIM: Task: Replacement\_Way\_Lookup\_LRU

SIM: Replace Way: 2

SIM: Task : Cache\_Line\_Fill

SIM: Way: 2, Tag: 16379, Line: 0

SIM: Miss:5

SIM: Way: 0

SIM: Index: 0

SIM: Word: 0: Content: fffc0000

SIM: Word: 1: Content: fffc0004

SIM: Word: 2: Content: fffc0008

SIM: Word: 3: Content: fffc000c

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

SIM: Way: 1

SIM: Index: 0

SIM: Word: 0: Content: fff80000

SIM: Word: 1: Content: fff80004

SIM: Word: 2: Content: fff80008

SIM: Word: 3: Content: fff8000c

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

SIM: Way: 2

SIM: Index: 0

SIM: Word: 0: Content: ffec0000

SIM: Word: 1: Content: ffec0004

SIM: Word: 2: Content: ffec0008

SIM: Word: 3: Content: ffec000c

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

SIM: Way: 3

SIM: Index: 0

SIM: Word: 0: Content: fff00000

SIM: Word: 1: Content: fff00004

SIM: Word: 2: Content: fff00008

SIM: Word: 3: Content: fff0000c

SIM: Index: 1

SIM: Word: 0: Content: 10000000

SIM: Word: 1: Content: 10000001

SIM: Word: 2: Content: 10000002

SIM: Word: 3: Content: 10000003

SIM: cache\_lru[way=0][line=0]: 1

SIM: cache\_lru[way=1][line=0]: 2

SIM: cache\_lru[way=2][line=0]: 3

SIM: cache\_lru[way=3][line=0]: 0

1. Result
2. Conclusion

LRU does not worth it.

1. MESI
2. Performance