

Cache Simulation Report

1. Aim

To analyse the behaviour of a **2-way set-associative cache** (T1) and evaluate the impact of the **FIFO (First-In-First-Out) block replacement policy** (T2) on cache performance.

2. Introduction

Cache memory reduces access latency by storing frequently used data. This report investigates:

- **Set-Associative Cache Mapping (T1)**
- **FIFO Block Replacement Policy (T2)**
- **Cache Hits and Misses**
- **Performance Impact of FIFO Policy**

A **2-way set-associative cache** allows two blocks per set, improving efficiency. The **FIFO policy** replaces the block that was loaded first when new data needs to be placed.

3. System Parameters

Parameter	Value
Address Width	8 bits
Cache Size	32 bytes
Block Size	8 bytes
Associativity	2-way set-associative
Write Policy	Write-back
Write Miss Policy	Write-allocate
Replacement Policy	FIFO (First-In-First-Out)

4. Methodology

1. Cache Accesses Performed:

- Read instructions executed: R(0x00), R(0x10), R(0x00), R(0x20)

- Cache hits and misses recorded.
- Address Breakdown:**
 - **Tag, Index, and Offset** extracted for each memory request.
 - FIFO Block Replacement:**
 - Oldest block in the set is replaced when new data needs space.
 - Cache State Analysis:**
 - Studied how data was stored and replaced using **FIFO**.

5. Observations

351 Cache Simulator

System Parameters:
Address width: bits
Cache size: bytes
Block size: ☐ 2 ☐ 4 ☒ 8 bytes
Associativity: ☐ 1 ☒ 2 ☐ 4 way(s)
Write Hit:
Write Miss:
Replacement:

☐ Explain

Manual Memory Access:
☐ Explain Addr: 0x20
 Addr: 0x, Byte: 0x

Tag	Index	Offset	Cache Hits	Cache Misses
0010	0	000	1	3

History:

R(0x00) = M
R(0x10) = M
R(0x00) = H
R(0x20) = M

 ||

Simulation Messages:
Block read into cache from memory at address 0x20.
FIFO statuses updated.
Data: 0xcd

m = 8, C = 32
K = 8, E = 2
Write back
Write-allocate
Eviction: FIFO

V D T Cache Data

Set	0	1
0	1 0 2 cd 4a f6 48 1a 6f 7e 63	2
1	1 0 1 b8 bd 1a ca 35 95 cb 80	1
0	0 0 - - - - - - - - - -	0
1	0 0 - - - - - - - - - -	0

Physical Memory

0x00	20 f6 ef ea a2 5e 9f 1a
0x08	a2 d0 4f c4 a0 0c f7 27
0x10	b8 bd 1a ca 35 95 cb 80
0x18	84 3f 02 4f 8e f3 f6 e5
0x20	cd 4a f6 48 1a 6f 7e 63
0x28	e9 36 ae 32 0d 37 bc c9

Cache Access Breakdown

Address Tag Index Offset Cache Hit/Miss

0x00	0000	0	000	Miss (M)
0x10	0001	0	000	Miss (M)
0x00	0000	0	000	Miss (M)
0x20	0010	0	000	Hit (H)

Set Associativity (T1)

- Set 0 contains two blocks:

- **Block 1 (Tag = 0010) → Loaded from 0x20**
- **Block 2 (Tag = 0001) → Loaded from 0x10**
- **Blocks are mapped to sets using index bits** to allow multiple blocks in a set.

FIFO Block Replacement (T2)

- **FIFO Eviction Occurred:**
 - **The oldest block (Tag = 0000 from 0x00) was replaced** when 0x20 was accessed.
 - **FIFO ensures blocks are evicted in order of arrival, without considering recent usage.**

Key Findings

- **Set Associativity (T1) reduced direct-mapped conflicts**, improving cache performance.
 - **FIFO (T2) replaced the oldest block**, even if it was still in use, potentially causing unnecessary evictions.
 - **Performance improved after initial cold misses, but FIFO may not always be optimal compared to LRU.**
-

6. Conclusion

- **Set Associativity (T1) enhances cache efficiency** by allowing multiple blocks in a single set.
- **FIFO (T2) is simple but can lead to suboptimal evictions**, as it does not track block usage frequency.
- **Cache performance improves after the initial cold misses**, but FIFO may not always be the best choice compared to LRU.

This analysis highlights **trade-offs in cache replacement strategies**, demonstrating the importance of selecting the right policy based on workload patterns.