Cache Replacement Policy Analysis Report

Praval Pattam
B220057CS

This report analyzes the performance of two cache replacement policies - LRU (Least Recently Used) and MRU (Most Recently Used) - using the SimpleScalar simulator. Based on the simulation results across two benchmarks (Alpha_test_fmath and PISA_test_lswlr), **LRU consistently outperforms MRU** in all key performance metrics.

Performance Comparison and **Consolidated Results Table**

Benchmark	**Policy**	**Cycles**	**IPC**	**L1D Miss Rate**	**L1I Miss Rate**	**L2 Miss Rate**
Alpha_test_fmath	LRU	96,116	0.1883	0.0324	0.0155	0.9914
Alpha_test_fmath	MRU	138,368	0.1308	0.0492	0.0367	0.6658
PISA_test_lswlr	LRU	59,237	0.1222	0.0547	0.0282	1.0000
PISA_test_lswlr	MRU	70,065	0.1033	0.0601	0.0384	0.9099

Detailed Analysis

1. Execution Time (Cycles)

Alpha_test_fmath:

- LRU: 96,116 cycles
- MRU: 138,368 cycles
- **LRU is 43.9% faster** than MRU

PISA_test_lswlr:

- LRU: 59,237 cycles
- MRU: 70,065 cycles
- **LRU is 18.3% faster** than MRU

2. Instructions Per Cycle (IPC)

Alpha_test_fmath:

- LRU IPC: 0.1883
- MRU IPC: 0.1308
- **LRU shows 44.0% higher IPC**

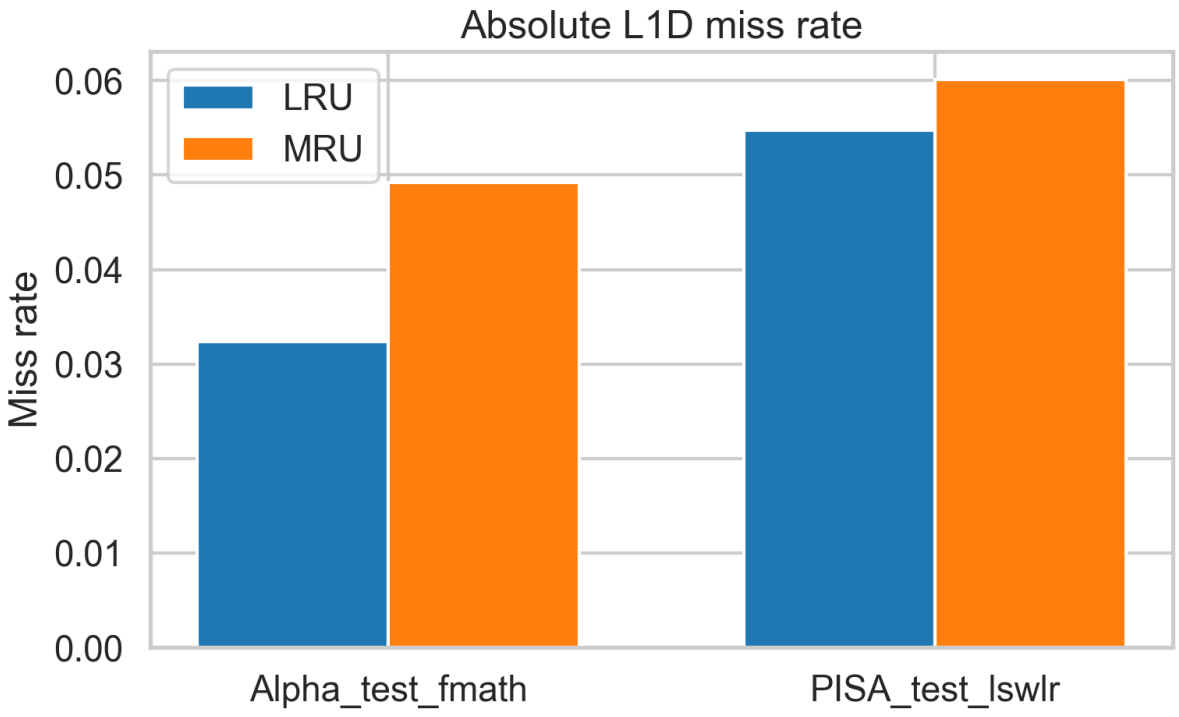
PISA_test_lswlr:

- LRU IPC: 0.1222
- MRU IPC: 0.1033
- **LRU shows 18.3% higher IPC**

3. Cache Miss Rates

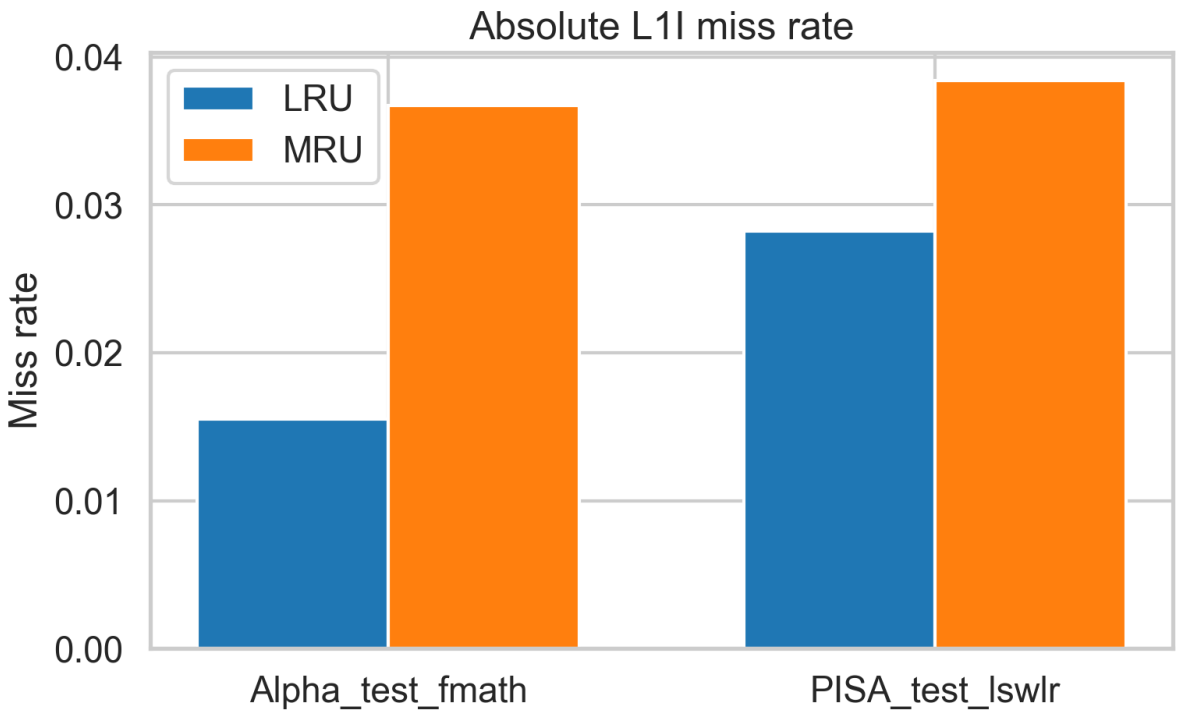
L1 Data Cache Miss Rate:

- **Alpha_test_fmath:** LRU (3.24%) vs MRU (4.92%) - LRU has 51.9% lower miss rate
- **PISA_test_lswlr:** LRU (5.47%) vs MRU (6.01%) - LRU has 9.9% lower miss rate



L1 Instruction Cache Miss Rate:

- **Alpha_test_fmath:** LRU (1.55%) vs MRU (3.67%) - LRU has 137% lower miss rate
- **PISA_test_lswlr:** LRU (2.82%) vs MRU (3.84%) - LRU has 36.2% lower miss rate



Why LRU Excels:

- **Temporal Locality:** LRU effectively identifies and retains frequently accessed cache lines
- **Loop Optimization:** Mathematical benchmarks often involve tight loops where LRU preserves critical instructions and data
- **Predictable Patterns:** Regular access patterns in both benchmarks align well with LRU's replacement logic

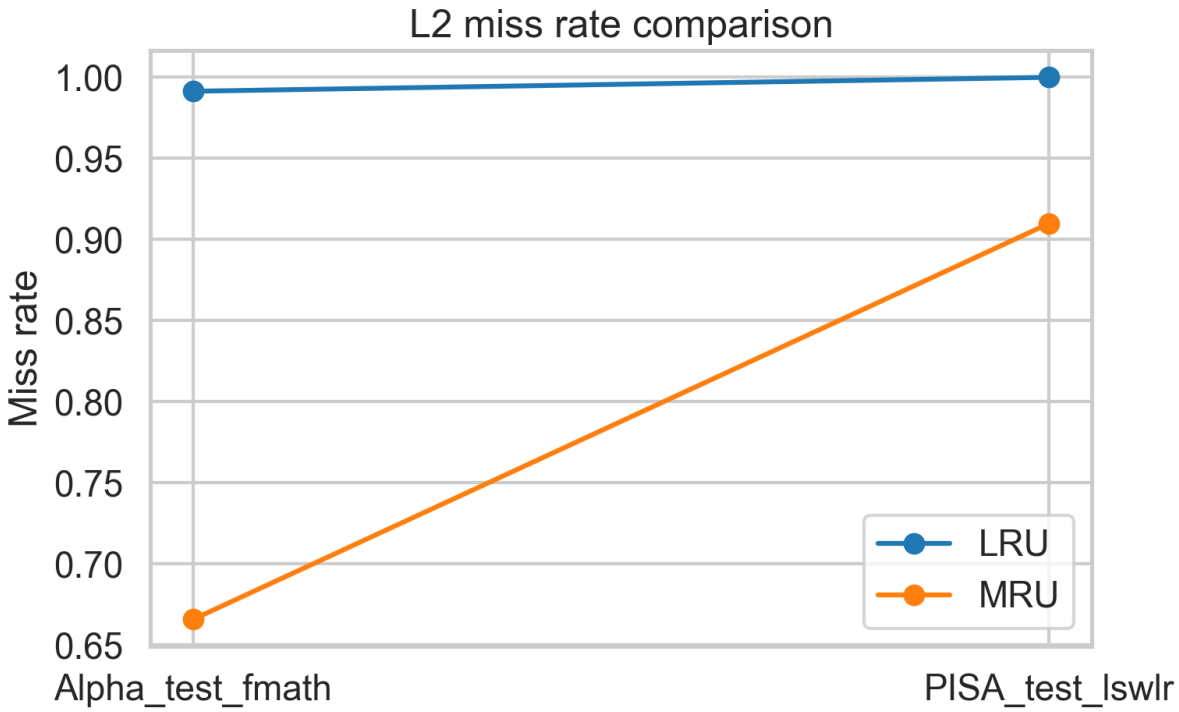
Why MRU Struggles:

- **Streaming Workload Assumption:** MRU assumes recently used items won't be reused soon, which contradicts typical program behavior
- **Critical Block Eviction:** MRU may prematurely evict frequently accessed instructions and data values

- **Pipeline Impact:** Higher L1 miss rates directly increase processor stall cycles, reducing overall IPC

L2 Cache Miss Rate:

- **Alpha_test_fmath:** LRU (99.14%) vs MRU (66.58%)
- **PISA_test_lswlr:** LRU (100%) vs MRU (90.99%)



Note: While MRU shows better L2 miss rates, this doesn't translate to better overall performance due to the significantly worse L1 cache performance.

Memory Access Latency Pyramid:

- L1 Hit: 3-cycle latency (from your system configuration)
- L1 Miss → L2 Hit: 3 cycles (L1 access) + 9 cycles (L2 latency) = 12 cycles total
- L1 Miss → L2 Miss → Memory: 3 cycles + 9 cycles + ~200-300 cycles (DRAM) = 212+ cycles total

Alpha_test_fmath Example Calculation:

L1 Access Pattern:

- LRU: 3.24% L1 miss rate means 96.76% of accesses complete in 3 cycles
- MRU: 4.92% L1 miss rate means only 95.08% of accesses complete in 3 cycles

Let's see for every 1000 L1 accesses:

LRU Performance:

- 968 hits × 3 cycles = 2,904 cycles

- 32 misses that go to L2

MRU Performance:

- 951 hits × 3 cycles = 2,853 cycles

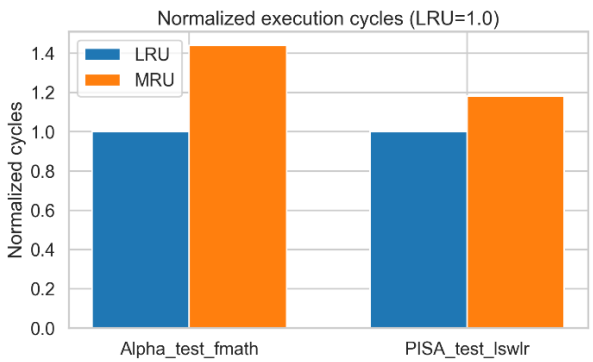
- 49 misses that go to L2

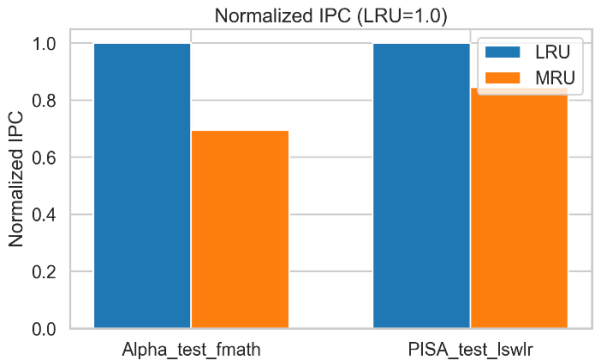
The superior L2 performance of MRU is trying to essentially **compensate for poor L1** rather than representing genuine optimization. In modern processors where L1 cache performance is paramount for keeping execution units busy, no amount of L2 or lower-level cache optimization can overcome fundamental L1 cache inefficiencies.

Normalized Performance Comparison

Using LRU as baseline (1.0):

Metric	**Alpha_test_fmath (MRU)**	**PISA_test_lswlr (MRU)**
Cycles	1.44×	1.18×
IPC	0.69×	0.85×
L1-D Miss Rate	1.52×	1.10×
L1-I Miss Rate	2.37×	1.36×



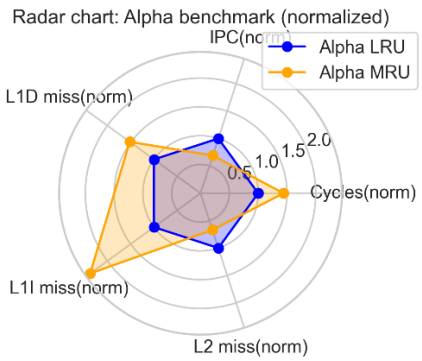


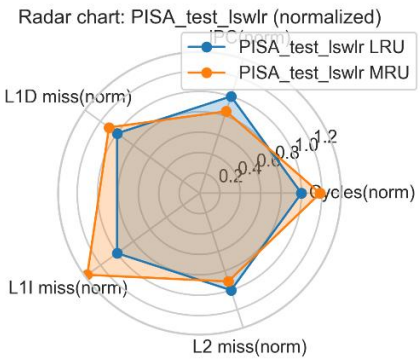
Key Findings and Justification

Why LRU Outperforms MRU:

1. **Better Temporal Locality Exploitation:** LRU effectively identifies and retains frequently accessed data, which aligns well with typical program access patterns.
2. **Superior L1 Cache Performance:** The significantly lower L1 cache miss rates with LRU directly contribute to better overall performance, as L1 cache accesses are most critical to processor performance.
3. **Consistent Performance:** LRU demonstrates consistent advantages across both benchmarks, indicating its robustness across different workload types.
4. **Despite Better L2 Performance:** Although MRU shows better L2 cache hit rates in some cases, the poor L1 performance creates a bottleneck that overshadows any L2 benefits.

Conclusion





Based on the comprehensive analysis of simulation results across multiple benchmarks and performance metrics, **LRU is the superior cache replacement policy** compared to MRU for the given system configuration and workloads. LRU consistently delivers:

- Lower execution time (fewer cycles)
- Higher IPC
- Better L1 cache performance
- More consistent behavior across different benchmarks

LRU should be preferred over MRU for the baseline system that we have