

Experimental Design

This document defines the experimental plan used to evaluate several cache-adjacent mechanisms implemented in SimpleScalar [sim-cache](#)

- Baseline caches (DL1 + unified L2)
- Victim Cache (VC)
- Miss Cache (MC)
- Stream Buffers (SB)
- Combined Mechanisms (VC+SB, MC+SB)

The goal is to quantify how each mechanism impacts:

1. DL1 miss rate
2. UL2 behavior
3. Indicators relevant to the specific mechanisms (VC hit rate, MC hit rate, etc.)

This experiment is a functional simulation, and not necessarily cycle-accurate. Therefore, results are interpreted in terms of miss behavior and traffic trends rather than precise runtime performance.

1. Workloads

Benchmarks

Workloads are small, repeatable programs invoked via [config/ss_benchmarks.txt](#). Each line defines:

- Benchmark name
- Program path (relative to project root, typically under [traces/](#))
- Optional args

Example format:

```
test-fmath traces/test-fmath
test-llong traces/test-llong
test-lswlr traces/test-lswlr
test-math traces/test-math
```

Rationale

These tests are intentionally lightweight to enable large parameter sweeps quickly and allow controlled comparisons across cache and mechanism settings.

2. Cache Hierarchy Under Test

The simulator models:

- IL1 (fixed)
- DL1 (swept)

- Unified L2 (UL2, swept)
- IL2 and DL2 are both pointed to the same unified cache configuration (UL2)

IL1 (fixed)

Instruction cache is held constant:

- `il1:32768:64:1:l` (32KB, 64B lines, direct-mapped)

DL1 sweep (primary size sweep)

DL1 is direct-mapped with 32B blocks, swept by changing the number of sets:

- `dl1:16:32:1:l`
- `dl1:64:32:1:l`
- `dl1:256:32:1:l`
- `dl1:1024:32:1:l`

This provides DL1 capacities of:

- 512B, 2KB, 8KB, 32KB ($n_{sets} \times 32B \times \text{assoc}$)

UL2 sweep (secondary size sweep)

Unified L2 uses 64B blocks and 4-way associativity, swept by changing number of sets:

- `ul2:8192:64:4:l`
- `ul2:16384:64:4:l`
- `ul2:32768:64:4:l`
- `ul2:65536:64:4:l`

This provides UL2 capacities of:

- 2MB, 4MB, 8MB, 16MB ($n_{sets} \times 64B \times 4\text{-way}$)

3. Cache Mechanisms Under Test

Each mechanism is enabled through the experiment mode and parameter settings.

Baseline

No extra structures are enabled to serve as a baseline

Victim Cache (VC)

A small fully associative buffer on DL1 read misses.

Parameter:

- `vc_entries = {2, 4, 8, 16}`

Primary metric:

- VC hit rate = `vc_hits/vc_lookups`

Miss Cache (MC)

A small fully associative miss buffer probed on DL1 read misses after VC (if VC is enabled)

Parameter:

- `mc_entries = {2, 4, 8, 16}`

Primary metric:

- MC hit rate = `mc_hits/mc_lookups`

Stream Buffers (SB)

Stream buffers are probed on DL1 read misses. On allocation, they trigger UL2 prefetch accesses.

Parameter:

- `sb_depth = {2, 4, 8, 16}`

Fixed parameters:

- `sb_count = 1`
- `sb_degree = 1`

Primary metric:

- SB hit rate = `sb_hits / sb_lookups`

Secondary metric:

- `sb_prefetches` (counts prefetch requests issued)

Combined mechanisms

`victim_stream` and `miss_stream` are also tested

- Hold SB depth while sweeping VC or MC
- Hold VC or MC while sweeping SB depth

4. Parameter Sweep Plan

Experiments were run over the full cross-product of:

- Benchmarks
- DL1 configs
- UL2 configs
- Mechanism modes and mechanism specific parameters

Modes run

- baseline

- victim (sweep `vc_entries`)
- miss (sweep `mc_entries`)
- stream (sweep `sb_depth`)
- victim_stream (sweep VC only, SB only, both)
- miss_stream (sweep MC only, SB only, both)

5. Outputs and Metrics

Raw simulator outputs

Each run produces:

- a text output per (benchmark, mode) run under the run-tag directory
- a per-directory CSV containing parsed metrics

Consolidated dataset

A summarization step merges all per-run CSVs into:

- `results/ss/ss_summary_all.csv`

Each row corresponds to one benchmark execution under a single run tag and mode.

Core metrics used for analysis

From sim-cache and parsing scripts:

- `dl1_miss_rate`
- `ul2_miss_rate`
- `vc_lookups`, `vc_hits`, derived VC hit rate
- `mc_lookups`, `mc_hits`, derived MC hit rate
- `sb_lookups`, `sb_hits`, `sb_prefetches`, derived SB hit rate
- `ul2_d_accesses`, `ul2_i_accesses`
- `ul2_total_accesses` (sum of UL2 D + I accesses)

Derived / normalized metrics (for presentation-quality comparisons)

Normalization is performed per (benchmark, DL1 config, UL2 config) relative to baseline:

- DL1 miss rate normalized = `dl1_miss_rate / baseline_dl1_miss_rate`
- UL2 miss rate normalized = `ul2_miss_rate / baseline_ul2_miss_rate`
- UL2 demand normalized = `ul2_total_accesses / baseline_ul2_total_accesses`

Normalization is used to compare mechanisms at the same cache geometry and workload while reducing baseline scale effects.

6. Reference Point for Comparisons

For bar-chart comparisons, a fixed reference cache geometry is used:

- DL1 reference: `dl1:1024:32:1:l`

- UL2 reference: **ul2:16384:64:4:l**
- SB depth reference: 4