Cache Coherence Simulator with MESI and MOESI Protocols

• Author info

Student name: Guangxing Hu

UnityID: ghu4

• Overview

This project implements a cache coherence simulator for a multiprocessor system using the MESI and MOESI protocols.

The simulator models cache behavior across multiple processors, supporting realistic evaluation of:

- Read and write operations with cache hits and misses.
- Cache invalidations during coherence actions.
- Memory transactions, including flushes and write-backs.
- Execution time estimation b◊ased on latency for various operations.

• Processor Access

1. MESI_Processor_Access:
    - Handles processor read and write requests.
    - Simulates cache coherence actions based on the MESI protocol.
    - Updates execution time, cache hits/misses, and transitions cache states.
2. MOESI_Processor_Access:
    - Similar to MESI_Processor_Access but includes the Owner state.
    - Simulates cache-to-cache transfers for dirty blocks.

• Bus Snooping

1. MESI_Bus_Snoop:
    - Handles bus transactions (BusRd, BusRdX, BusUpgr) in the MESI protocol.
    - Invalidates or transitions cache states in response to coherence requests from other processors.
2. MOESI_Bus_Snoop:
    - Extends MESI_Bus_Snoop with support for the Owner state.
    - Includes logic for handling dirty blocks and reducing memory bandwidth usage.

• Usage

1. Compilation

    To clean all:

    make clean

    To compile the simulator, use the provided Makefile

    make all

2. Running the simulator

    make test_mesi

    or

    make test_moesi

    If you want your output to a file:

```
make test_moesi > output_moesi.out 2>&1
```

- ## Output

The simulator prints cache statistics at the end, including:

- Number of reads and writes.
- Cache hit and miss counts.
- Total invalidations and flushes.
- Total execution time.

Example output:

```
============ Simulation results (Cache 3) ============
    01. number of reads:                          113428
    02. number of read misses:                       8922
    03. number of writes:                          12108
    04. number of write misses:                        51
    05. number of write hits:                       12057
    06. number of read hits:                       104506
    07. total miss rate:                          7.15%
    08. number of memory accesses (exclude writebacks):     8973
    09. number of invalidations:                     1981
    10. number of flushes:                          6257
```