Industrial Project (236504)

Cache loading sequence generator

Project Requirements

Students: Ohad Edelstain, Itai Gannon
Supervisors: Eyal Bin, Allon Adir and Amir Nahir
Introduction

- Modern IBM memory sub-systems include a hierarchy of caches designed to enhance the overall performance of the system.
- Maintaining the coherency of these caches when used by multiple processors is a complex task that must be taken care of by the memory sub-system.
- Verifying the successful maintenance of cache coherency is thus an important goal of processor verification.
Our Objective

- This project involves the development of a tool that receives a statically generated state of the cache hierarchy and outputs an assembly program that would bring the caches to the desired state.

- This program could later be used by, for example, hardware exercisers as a means of reaching cache states that are interesting for verification purposes.
Methodology

- There are a number of protocols for synchronizing between the caches of the different processors.
- The project will implement a refined MESI protocols.
- MESI protocol:
  - In MESI protocol every cache line had a state:
    - **Modified**: The cache line is present only in the current cache, and is dirty; it has been modified from the value in main memory.
    - **Exclusive**: The cache line is present only in the current cache, but is clean; it matches main memory.
    - **Shared**: The cache line may be stored in other caches of the machine, and is clean.
    - **Invalid**: The cache line is invalid and must be ignored.
The MESI Diagram
Milestones

- **Preliminary Analysis (Week 2)**
  Getting acquainted with the industry needs and outlining the project roadmap.

- **Analyze & Design (Week 6)**
  Fully understanding the system workflow and the interaction with the environment (input and output formats).
  Designing the program down to the class level.

- **Coding (Week 11)**
  Writing the code, debugging and minor testing.
  The program will be developed in Java using eclipse. Emphasis will be placed on portability and proper documentation.

- **Testing & Finalizing (Week 13)**
  Testing the program, fixing any bugs found.
Development Environment

- The program will be written in Java.
- This would allow it to be executed on different environments and operating systems.
Deliverables

- The final version of the project will be able to receive a required state of the caches and deliver an output file that would get the real hardware as close as possible to that desired outcome.