Advanced Computer Architecture

Lecturer: Hajar Falahati

Frist Semester 98-99

دانشگاه علم و صنعت ایران

Iran University of Science & Technology

IUST

Homework 1 Computer Architecture Recall Deadline 1398/07/12

1. Direct-mapped i-cache.

Instruction cache, a.k.a., i-cache, acts as a memory between external memory and the processor. By storing instructions inside an i-cache, process fetches the instructions faster and can achieve throughput gain due to removing access to the external bus; especially when the same code is used frequently. Figure 1 illustrate a sample i-cache.

Features. We need to exploit a level 1 instruction cache with the following features:

- 32 KB
- **Direct-mapped cache**
- Each block contains four instructions; each instruction has 32-bit length.
- 32-bit addresses
- 1 valid bit per block
- 1 clock cycle hit time

Requirements. Please design a i-cache to support all the features.

- Draw the detailed architecture and microarchitecture for your cache.
- Draw the detailed architecture and microarchitecture for your cache. Indicate the required signals and their width, e.g., how wide is tag?.
- Implement your cache via Verilog code. Please note that your code should (1) be **parametric**, and (2) **have enough comments**.
- Test the implemented cache via a suitable test bench.

2. 4-way i-cache

We would like to evaluate different cache structures. Considering the previous cache, design a 4-way i-cache.

Features. We need to exploit a level 1 i-cache with the following features:

- 32 KB
- **4-way cache**

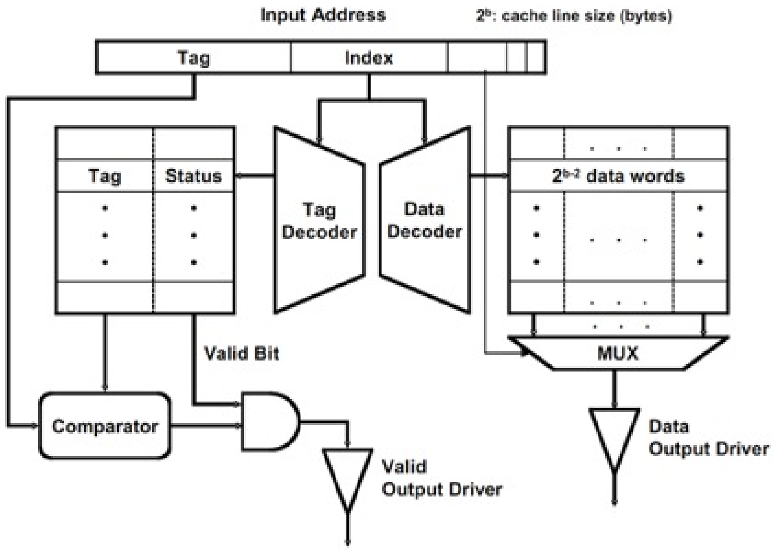

Figure 1: Sample i-cache.

- **Replacement policy: LRU**
- Each block contains four instructions; each instruction has 32-bit length.
- 32-bit addresses
- 1 valid bit per block
- 1 clock cycle hit time

Requirements. Please design an i-cache to support all the features.

- Draw the detailed architecture and microarchitecture for your cache. Indicate the required signals and their width, e.g., how wide is tag?.
- **Using hierarchical and structural design techniques, implement your cache built upon the cache implemented in question 1**. Please note that your code should (1) be **parametric**, and (2) **have enough comments**.
- Test the implemented cache via a suitable test bench.

Required Document

Please upload a zip file, titled **[ACA-HW1][Student Number].**

- All your Verilog codes.
- A document to report the detailed description and architecture of the problems.

General Rules

Please consider the course rules.

- Deadlines are tight.

- You have 3-day extra time for submitting assignments/projects in the whole semester.

- 20% penalty for your late assignment/project submissions after 3-day extra time!

- Zero score for the late assignment/project submissions after uploading the solution in the course website!

- Zero score for copied assignments/projects and academic misconduct.

Deadline

Friday 23:59. 1398/07/12.

Contact Information

Ask your questions via the course website or send an email to:

```
farahani_a@cmps2.iust.ac.ir
hfalahati@iust.ac.ir
```

Good Luck