

ADS2 Assessed Exercise: Modeling a Cache – Part II

Wim Vanderbauwhede

1 Aim

The aim of this coursework is to create a Java model for a simple cache as used in modern computer systems.

2 What to submit

For the second part of the assignment, you have to write the code for the `FullyAssocLiFoCache` class based on the provided skeleton and the required Java datastructures.

- You *must* start from the provided code and you should *only add your own code* in `FullyAssocLiFoCache.java`, you *must* not modify or removed any part of the provided code.
- You *must* use the `Status` class for status reporting as it will be used by the testbench to verify the correctness of your code.
- If you use `println` statements in your code, they *must* be guarded by an `if (VERBOSE)` guard, for example:

```
if (VERBOSE) System.out.println("Flushing cache");
```

- You *must* submit this code in a *gzipped tar archive* (other formats will *not* be accepted) through the Moodle submission system, and the filename *must* be *<your matric + 1st char of your name in lowercase>.tgz*, so for example if your matric number is *1107023m* then your archive *must* be named *1107023m.tgz*. This archive *must* contain a single folder named *<your matric + 1st char of your name in lowercase>*. This folder *must* contain a subfolder `src/ads2/cw1/` with following files:

- `Cache.java` as provided
- `FullyAssocLiFoCache.java` in which you must implement the functionality for the cache model
- `Main.java` as provided

- `Memory.java` as provided
- `Status.java` as provided
- `TestBench.java` as provided

3 What is provided

You can get the coursework source files and the coursework description at <https://github.com/wimvanderbauwhede/ADS2>. If any updates would be required I will push them to this repository.

4 How to test your code

You should unit test every method you implement. Once your code is fully implemented you can use the provided testbench to see if it works as expected.

5 Marking

- Please note that whenever the specification (this document) says *must*, then you will lose marks if you don't. In particular, if you do not follow the required structure and naming conventions for your submission, you will be penalised.
- Your code will be marked using a test script. This script will test if your code builds correctly and runs correctly for a number of use cases (70% of the marks). I will not modify the provided testbench code but I may modify the memory and cache parameters. I will also run an additional test.
- The script will also perform source code analysis to see if you have used the correct API calls, data structures etc (30% of the marks). In particular, I will check
 - Correctness of the constructor of the `FullyAssocLiFoCache` class.
 - Correct use of the various attributes of the `FullyAssocLiFoCache` class in the implementation of its methods.
 - Correct implementation and use of the various methods of the `FullyAssocLiFoCache` class.
 - Correct use of the `Status` class to report the status of the cache at every operation.