**Project 2**

**Project description**

You will need to implement a trace-driven cache simulator, and use it to evaluate the performance of different cache architecture features. The project is described in detail in sim.pdf.

**File description**

In the attachment, you will find the following project files:

l cache.h - cache simulator definitions

l cache.c - cache simulator (put all your edits here)

l main.h - simulation driver definitions

l main.c - simulation driver

l Public tests

n public-block.trace - test cache block sizes

n public-assoc.trace - test cache associativity

n public-write.trace - test cache write policy

n public-instr.trace - test instruction cache

n spice10.trace - 1st 10 accesses in spice.trace

n spice100.trace - 1st 100 accesses in spice.trace

n spice1000.trace - 1st 1000 accesses in spice.trace

l Expected outputs for public tests

n public-block1.out

n public-assoc1.out

n public-write1.out

n public-instr1.out

n spice10.out

n spice100.out

n spice1000.out

l Makefile - makefile to create simulator

l runPublic - csh script to run public tests

l sim.pdf - detailed description of project

l tags.txt - example index & tag values for spice100.trace In addition, there are three large application traces:

l spice.trace - circuit simulator

l cc.trace - C compiler

l tex.trace - Tex document processor

**Requirements**

l Write a simple report about how you implement the cache simulator and answer the questions in sim.pdf

l Compress your project and submit it to XueZaiZheDa before the deadline. And hand in the paper report in class on 2025.1.2. Use your student ID as filename. Files you should submit:

n code/ directory

n Report

**Note**

l It’sokif you can’t implement a fully functional simulator, make your best effort and do what you can do.