Computer Sciences Department
University of Wisconsin-Madison
CS/ECE 552 – Introduction to Computer Architecture
In-Class Exercise (03/24)

Answers to all questions should be uploaded on Canvas.

1. [10 points] Consider the following MIPS assembly program:

```
ori  $s0, $zero, 0x2000
lw   $t3, 0xC($s0)
lw   $t2, 0x28($s0)
slt  $t3, $t3, $t2
lw   $t1, 0x50($s0)
lw   $t0, 0x34($s0)
sub  $t0, $t1, $t0
lw   $t1, 0x4C($s0)
add  $t1, $t1, $t2
lw   $t2, 0x10($s0)
```

The contents of some addresses in data memory are shown below, **before** the program is executed. Assume 32-bit memory addresses.

Memory Address (in hex)	Word Contents (in decimal)
0x00002000	-454
0x00002008	9
0x00002010	2
0x00002018	46
0x00002020	100
0x00002028	54
0x00002030	92
0x00002038	-434
0x00002040	-4
0x00002048	1
0x00002050	75
0x00002058	2

Memory Address (in hex)	Word Contents (in decimal)
0x00002004	77
0x0000200C	53
0x00002014	5899
0x0000201C	10
0x00002024	-44
0x0000202C	-78
0x00002034	20
0x0000203C	1111
0x00002044	7
0x0000204C	-9
0x00002054	3
0x0000205C	-410

You implement a **32B direct-mapped cache** for data memory, with **8-byte cache blocks**. Unfortunately, you find a design bug in your cache where you only have space to store 26 bits per tag. You decide to use the cache anyway, setting each tag to be **bits 31 to 6** of the memory address.

Assuming that the cache is initially empty, what are the contents of registers **$t0, $t1, $t2** and **$t3 after** executing the program above? You are encouraged to use the table on the next page to track the cache contents.

Load Address	Tag	Index	Offset	Hit?	Cache Contents After Load
0x200C	80	1	4	N	[], [9, 53, 0x80], [], []
0x2028	80	1	0	N	[]. [54, -78, 0x80], [], []
0x2050	81	2	0	N	[], [54, -78, 0x80], [75, 3, 0x81], []
0x2034	81	2	4	N	[], [54, -78, 0x80], [92, 20, 0x81], []
0x204C	81	1	4	N	[], [1, -9, 0x81], [92, 20, 0x81]
0x2010	80	2	0	N	[], [1,-9, 0x81], [2, 5899, 0x80], []