CS 61C Midterm #2 — November 2, 1994

Your name ____

login cs61c–________

Discussion section number ________

TA's name ____

This exam is worth 25 points, or 18.7% of your total course grade. The exam contains six substantive questions, plus the following:

Question 0 (1 point): Fill out this front page correctly and put your name and login correctly at the top of each of the following pages.

This booklet contains five numbered pages including the cover page. Put all answers on these pages, please; don't hand in stray pieces of paper. This is an open book exam.

Our expectation is that many of you will not complete one or two of these questions. If you find one question especially difficult, leave it for later; start with the ones you find easier.

0	/1
1	/6
2	/3
3	/2
4	/3
5	/4
6	/6
total	/25

Question 1 (6 points): Given the physical address `0x0f1653f8`, compute its block offset, set number, and tag for each of the following caches:

size (words)	width (words)	policy	offset	set	tag
1024	8	direct mapped			
4096	16	4-way set associative			
4096	4	fully associative			

Question 2 (3 points): Disassemble the following machine instructions into TAL:

```
0x02282822
0xAFB00008
0x0541FFFE
```

Question 3 (2 points): What is the use of the dirty bit in a page table entry or a TLB entry?

Question 4 (3 points): For each of the following tasks, indicate whether it is always done by hardware, always done by the operating system, or could be either way:

a. Reading a page table entry from main memory.

b. Reading a TLB entry.

c. Setting the write enable bit for a page.

Question 5 (4 points): Consider the following configuration:

Virtual address:	20 bits
Physical address:	16 bits
Page size:	4K bytes (1K words)
Cache size:	256 bytes (64 words)
Cache width:	4 word
Cache type:	direct-mapped
Cache policy on write:	write-through
TLB size:	4 entries

The machine is byte addressable.

a. How many entries are there in the page table?

(This question continues on the next page.)

Question 5 continued:

b. The current data store in this computer is as follows:

TLB:

	Virtual page #	Physical page #
0	0x1a	0x2
1	0x01	0x8
2	0x05	0xb
3	0x2f	0x0

Cache:

	Valid	Tag	Word #0	Word #1	Word #2	Word #3
0	1	0x20	0x10	0x20	0x30	0x40
1	1	0x08	0x11	0x22	0x33	0x44
2	0					
3	0					
4	1	0xb6	0x90	0xa0	0xb0	0xc0
5	0					
6	0					
7	0					
8	1	0xb5	0x50	0x60	0x70	0x80
9	0					
a	0					
b	0					
c	0					
d	0					
e	0					
f	1	0x81	0x55	0x66	0x77	0x88

What is the data contained in virtual address `0x05584`? (Hint: You do not need to know what the content of the page table or physical memory is.) **Show your work! Indicate which TLB and/or cache entries you use.**

Question 6 (6 points): Convert the following C function into a MAL procedure. Use $16, $17, $18, and $19 for local variables. Parameters are passed in $4. Make sure you follow the conventions discussed in class. You will not get full credit if you do not follow the conventions.

```
int finalGrade(int studentId)
{
    int hwGrade;
    int mtGrade;
    int projGrade;
    int grade;

    hwGrade = getHwGrade(studentId);
    mtGrade = adjustCurve(getMtGrade(studentId));
    projGrade = getProjGrade(studentId);
    grade = hwGrade + mtGrade + projGrade;
    return(grade);
}
```