University of California, Berkeley – College of Engineering

Department of Electrical Engineering and Computer Sciences

Summer 2015 Instructor: Sagar Karandikar 2015-08-13

CS61C FINAL

After the exam, indicate on the line above where you fall in the emotion spectrum between "sad" & "smiley"...

Last Name	
First Name	
Student ID Number	
Login	`cs61c-`
*The name of your **SECTION** TA (please circle)*	Derek | Harrison | Jeffrey | Nathaniel | Rebecca
Name of the person to your Left	
Name of the person to your Right	
*All the work is my own. I had no prior knowledge of the exam contents nor will I share the contents with others in CS61C who have not taken it yet. **(please sign)***	

Instructions (Read Me!)

- This exam contains 16 numbered pages including the cover page. **The back of each page is blank and can be used for scratch-work but will not be looked at for grading.** (i.e. the sides of pages without the printed "SID: _____" header will not even be scanned into Gradescope).
- Please turn off all cell phones, smartwatches, and other mobile devices. Remove all hats & headphones. Place your backpacks, laptops and jackets under your seat.
- You have 80 minutes to complete this exam. The exam is closed book; you may not use any computers, phones, wearable devices, or calculators. You may use one page (US Letter, front and back) of handwritten notes in addition to the provided green sheet.
- There may be partial credit for incomplete answers; write as much of the solution as you can. We will deduct points if your solution is far more complicated than necessary. When we provide a blank, please fit your answer within the space provided. "IEC format" refers to the mebi, tebi, etc prefixes.

	M1-1	M1-2	M1-3	M2-1	M2-2	M2-3	F1	F2	F3	Total
Points Possible	11	20	18	20	15	19	12	20	17	152

SID: _____

M1-1) Number Representation (11 pts)

Part 1: Quick conversions:

a) What is 0xFFFFFFFC in decimal when read as an unsigned value?

b) Encode the decimal value -10 in 32 bits of two's complement. Give your answer in hex.

Part 2: We have a new number representation called Balanced Ternary that uses "Signed Digit Representation" in ternary trits. Each digit can individually represent a positive, zero or a negative value. The n^{th} digit (zero-indexed from the right) gets multiplied by 3^n – just like in regular ternary. Instead of having the three states represent 0, 1 and 2, they represent for **-1, 0, and 1**. **We will use + to denote +1, 0 to denote 0, and – to denote -1.**

Convert the following Balanced Ternary numbers to decimal.

c) 000+ _____

d) 00+- _____

e) +--0 _____

f) How many ways can you represent the decimal number 4? Write all of them out.

What do you get (expressed in balanced ternary) when you add the following numbers vertically? (Hint: use the result from (g)-(i) to do (j))

g)	+	h)	0+	i)	0-	j)	++0+ - -
	-		0+		0-		- +0++ -

_____ _____ _____ _____

SID: _____

M1-2) C Pointers, Arrays and Strings (20 pts)

a) Fill in the blanks below for the function `tokenize` which splits up a string into proper C strings whenever it encounters a specified delimiter (a single character).

Example:

```
char * string = "Summer*time";
printf("%s", tokenize(string, '*')[0]);    // prints: Summer
printf("%s", tokenize(string, '*')[1]);    // prints: time
```

Inputs:

`char * str` - a NULL terminated character array

`char delimiter` - a character to split on

- You may **assume that the delimiter appears at least once in the input str** and the first/last character in the input string are not the delimiter.

Outputs:

An array of strings split by the provided token (pointers and string data on the heap). See the example above. The delimiter should never appear in an output string and your output should never contain strings of length zero.

Requirements:

You may use **one** additional malloc call beyond the one provided in the skeleton.

Assume that you have access to the functions:

`int count_occurences(char * str, char token);` // Counts occurrences of a token in a string

`char * strcpy (char * destination, char * source);` // copies string `source` to buffer `dest`

```
_____________ tokenize(char * str, char token) {
    char * copy = malloc(sizeof(char) * strlen(str) + _____________________________);
    strcpy(copy, str);

    _________________ string_array = ____________________________________________;

    ____________________________________________________________________________;

    ____________________________________________________________________________;

    while(_____________________________________________________________ != '\0'){

        if (_____________________________________________________________){

            ____________________________________________________________

            ____________________________________________________________

            ____________________________________________________________

        } else if (_____________________________________________________________){

            ____________________________________________________________

        } else {_____________________________________________________________}
    }
    return string_array;
}
```

SID: _____

M1-2) C Pointers, Arrays and Strings (continued)

b) What does the following program print if the starting address of x is 1000_{ten}, and the starting address of array arr is 2000_{ten}? All addresses are byte-addressed.

```
#include <stdio.h>
int main() {
     int *p, x;
     int arr[5] = {1000, 2000, 3000, 4000, 5000};
     int *p2;
     p = NULL;
     x = 5000;
     p = &x;
     printf("%d %d %u %u\n", x, *p, p, &x);
     p2 = arr;
     *(p2+1) = *p;
     *p = *p2 + *(p2+2);
     p2++;
     printf("%d %d %d %u\n", x, *p, *p2, p2);
     return 0;
}
```

_____

_____

SID: _____

M1-3) What's happening? It's a MIPStery! (18 pts)

The code below does something funky. You can assume that we only need to run this code on a "bare" system – one with no address translation (i.e. no Paging or Base and Bounds).

```
0x000004  main: li  $a0, 1
0x000008        jal     mystery        # ← (A)
0x00000c        addu    $a0, $0, $v0
0x000010        jal mystery         # ← (B)
0x000014        addu    $a0, $0, $v0
0x000018        jal mystery         # ← (C)
0x00001c        addu    $a0, $0, $v0
0x000020        jal mystery         # ← (D)
          …
0x800004  mystery: lui    $t0, 0xffff
0x800008           lui    $t2, %Hi(mystery)
0x80000c           ori    $t2, %Lo(mystery)
0x800010           addiu $t1, $0, 0
0x800014           andi  $a0, $a0, 0xffff
0x800018           add    $v0, $a0, $t1
0x80001c           lw     $t3, 12($t2)
0x800020           and    $t3, $t3, $t0
0x800024           or     $t3, $t3, $a0
0x800028           sw     $t3, 12($t2)
0x80002c           jr     $ra
```

a) What is the value in `$v0` **after** returning from the function call in the line marked:

(A) _____ (B) _____ (C) _____ (D) _____

b) What does the function `mystery` return if we follow the pattern in `main` above, calling `mystery` `N` times? (You should use `N` in your answer.)

c) Write TAL MIPS code to "reset" the function `mystery` such that after a `jal reset_mystery` line, the next call to mystery with `$a0` = 1 behaves like the function call in (A). You might not need all of the lines provided. In your answer, you can use the %Hi and %Lo directives used above.

```
reset_mystery:
```

_____

_____

_____

_____

_____

_____

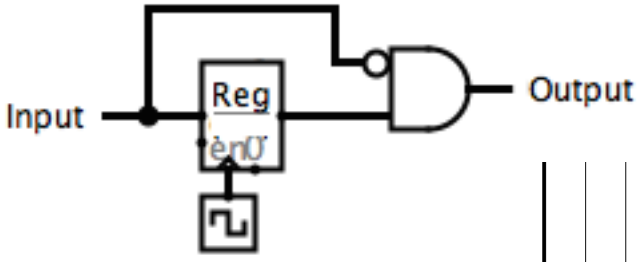
_____

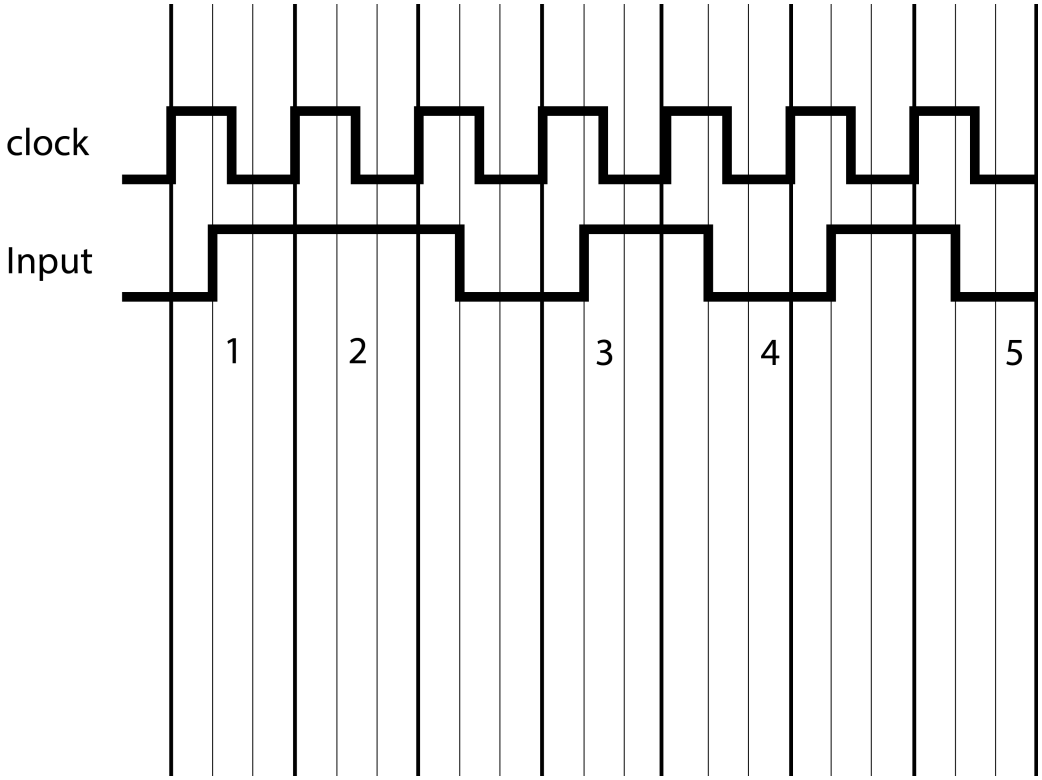
_____

SID: _____

M2-1) SDS, Gates, Timing, FSM (20 pts)

a) Assume that the clock period is 15ns, the input comes 5ns after each positive clock edge, the register has no hold and setup times but has a clock-to-Q delay of 5ns, and gates have a 5 ns delay (with or without bubbles in the input).

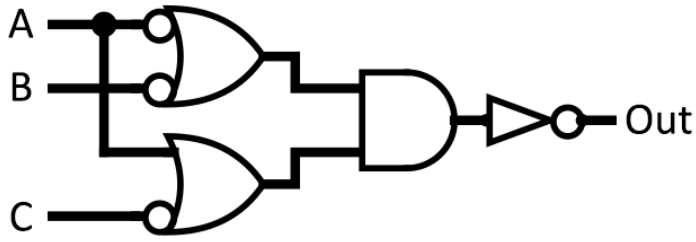




What is the value of the output signal at the numbered markers? **Write X for unknown/undefined**. We have provided space for you to sketch out signals, but we will not grade any sketches.

1) _____ 2) _____ 3) _____ 4) _____ 5) _____

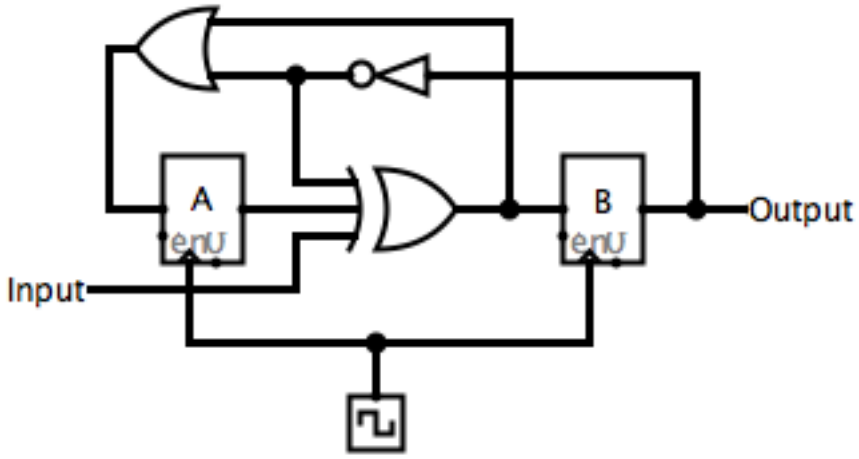
b) Write a Boolean expression for Out in terms of inputs A, B and C in the circuit below and simplify the expression so it can be built using only 4 basic gates without bubbles. Basic gates are: OR gates, AND gates, NOT gates.



SID: _____

M2-1) SDS, Gates, Timing, FSM (continued)

c) The circuit below has the following timing parameters:
Clock period: 20ns
XOR gate delay: 5ns
OR gate delay: 5ns
Inverter delay: 4ns
Register setup time: 2ns
Input comes 1ns after every clock edge



i) What is the maximum possible clk-to-Q delay for the registers?

ii) Assuming a clk-to-Q delay of 3ns, what is the maximum possible hold time for the registers?

M2-1) SDS, Gates, Timing, FSM (continued)

d) Using as few states as possible, **draw an FSM diagram** that satisfies the following:

- We are reading a buffer bit-by-bit containing data surrounded by a "header" and a "footer".
- The header is defined to be the bit sequence "10". Once we have seen "10", then we will read in bit-by-bit and always output the bit we read in last.
- The footer is also defined to be the bit sequence "10". If we are in the body, once we have seen a "10" we are at the end of the data section.
- The FSM is initialized to detect headers. Whenever we are not within the body of the data or the footer, it always outputs 0. Note that this buffer is infinitely long and should continue to detect a header after finishing a footer.

Use labels/words for your states instead of 1's and 0's to make your life easier.
Note the following sample input -> output scheme:
Cycle 01: 0 -> 0 # garbage value
Cycle 02: 1 -> 0 # header
Cycle 03: 0 -> 0 # header
Cycle 04: 0 -> 0 # data
Cycle 05: 0 -> 0 # data
Cycle 06: 1 -> 1 # data
Cycle 07: 1 -> 1 # footer
Cycle 08: 0 -> 0 # footer
Cycle 09: 1 -> 0 # garbage value
Cycle 10: 1 -> 0 # garbage value
Cycle 11: 1 -> 0 # garbage value
Cycle 12: 1 -> 0 # header
Cycle 13: 0 -> 0 # header
etc…

Place your solution in this box. Be sure to indicate which state is your start state.

SID: _____

M2-2) MIPS Pipeline (15 pts)

The code on the left is written for a single-cycle MIPS CPU. **Put a star to the left of lines that are involved in hazards** (both the lines that cause and are affected by hazards), reorganize the code order, and insert nops so that the code runs in as few cycles as possible on a 5-stage pipelined MIPS CPU. Assume that the CPU has complete forwarding in the pipeline with load, branch and jump delay slots; we have a branch comparator in the decode stage; but we do not have the ability to send bubbles in the pipeline. You might not need all of the provided lines.

```
Start:
      and  $t4 $t1 $t2
      ori  $t5 $s0 0xFFFF
      add  $t0 $t1 $t2
      add  $s0 $t2 $t3
      beq  $s0 $t0 Somewhere
      addi $s2 $0 0
Loop:
      lw   $s1 4($s0)
      addi $s1 $s1 1
      sw   $s1 0($s0)
      addi $s2 $s2 1
      jal  DoStuff
      addi $s0 $s0 4
      blt  $s1 $s2 Loop
      beq  $s1 $s2 Start
      jr $ra
```

```
Start:
      _____
      _____
      _____
      _____
      _____
      _____
      _____
      _____
Loop:
      _____
      _____
      _____
      _____
      _____
      _____
      _____
      _____
      _____
      _____
      _____
      _____
```

SID: _____

M2-3) Cache Performance/AMAT (19 pts)

Each subproblem in this question can be done without the previous subproblems.
We have caches with the following configurations:
Cache-size: 1KiB, Block-size: 128B, Memory-size: 4 GiB, Write-policy: Write back and write-allocate

```
int rand(x,y);  // defined elsewhere, returns a random integer in the range [x,y)
#define repcount 4
int A[1024*1024]; // block aligned
int B[8*1024]; // block aligned
...
// ← Start (assume cache is cold)
for(int rep = 0; rep < repcount; rep++)
     for (int i = 0; i < 1024; i++)
          A[8*i] = B[8*i] + A[256*i + 8*i + rand(0, 32)] + A[8*i];
// ← Stop
```

For the following scenarios calculate the hit-rate and specify which types of misses are encountered in the code between the lines marked start and stop. Assume all variables stay in registers and memory reads happen left to right in each line.

a) i) Direct Mapped: # Tag/Index/Offset bits: [_____ : _____ : _____]
ii) What is the best-case hit rate? Also, state the types of misses.

iii) What is the worst-case hit rate? Also, state the types of misses.

b) i) 2-way Set Assoc: # Tag/Index/Offset bits: [_____ : _____ : _____]
ii) What is the worst-case hit-rate? Assume we use LRU replacement. Also, state the types of misses.

c) Calculate the AMAT in cycles if the L1 local hit rate is 60%, with a hit time of 1 cycle, the L2 global hit rate is 20% with a hit time of 10 cycles and the main memory has a hit time of 200 cycles.

SID: _____

F1) MapReduce (12 pts)

You have been hired as a computer scientist for a retail company where your boss is trying to optimize the location of his products in the store by analyzing the products bought by customers. He decides that he wants you to take their existing code and add on MapReduce in order to analyze giant logs of information more quickly.

a) Right now, the machine that you work on finishes the old program in 30 seconds. Your boss gives you extra funds for a new machine that will be able to take existing code and have it finish in 10 seconds. You notice that 5/6 of the code is parallelizable. What is the minimum number of threads your new machine has to have in order to achieve this speedup?

b) You now move on to working on MapReduce. Your boss, who loves hats, complains that sales for hats are not great. Previously, he tried putting different colors of hats with items of similar type but sales did not improve.

So, he proposes to take advantage of spatial information and wants to move the hats according to the aisle number (10 in total) of other items that were bought by those who bought hats.

For example, if people have bought items in aisle 2 AND they bought a red hat, your boss wants to consider moving red hats to aisle 2 because customers who buy items in aisle 2 has a higher probability to buy a red hat.

You are given a customer log which contains:
`(customer_id, list(item_object))` where `item_object` is defined as:

```
item_object {
      int color      // Color of product encoded as an integer
      int product_id // The ID of the product; the ones digit of item_id
}                    // corresponds to aisle number it is in (only 10 aisles)
```

Your output should be in the form of N key-value pairs `(hat_color, aisle_number)` if there are N unique hat colors bought in the customer log. These pairs should tell your boss where the different colored hats should be put.

Also, **assume you are given the following functions**:

1) `isHat(product_id)` that takes a product ID and returns whether that product is a hat or not.
2) `indexMax(array)` that takes an array and outputs the **index** of the largest value in the array.
3) `set1.add(value1)` adds some value (`value1`) to a `set()` named `set1`

You can access fields in an `item_object` using dot notation (e.g. `item_obj.color`)

Write **pseudocode** in the provided MapReduce functions on the following page so that at the end of your reduce function, you will provide the best color hat that should be placed in each aisle.

SID: _____

F1) MapReduce (continued)

```
map(customer_id, item_list) {
      hats = set()                // empty set
      other_items = set()         // empty set
      for item in item_list:

            aisle_number = ____________________________________________________

            ____________________________________________________________________

            ____________________________________________________________________

            ____________________________________________________________________

            ____________________________________________________________________
      for hat in hats:
            for item in other_items:

                  emit(_____________________________________________________)
}


reduce(key, values_list) {
      totals = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

      for ____________________________________________________________:

            ____________________________________________________________________

     emit(_____________________________________________________________)

}
```

SID: _____

F2) Virtual Memory (20 pts)

For the following questions, assume the following (IEC prefixes are on your green sheet):

- **You can ignore any accesses to instruction memory (code)**
- 16 EiB virtual address space per process
- 256 MiB page size
- 4 GiB of physical address space
- Fully associative TLB with 5 entries and an LRU replacement policy
- All arrays of doubles are **page-aligned** (start on a page boundary) and do **not** overlap
- All arrays are of a size equivalent to some nonzero integer multiple of 256 MiB
- All structs are tightly packed (fields are stored contiguously)
- All accesses to structs and arrays go out to caches/memory (there is no optimization by reusing values loaded into registers)

```
----------------------------------------------------------------------
typedef struct { *double dbl; double fun;} doubleFun

void dblCpy(doubleFun* measurer, double* dblsToCpy) {
     measurer->fun = 0;
     for (uint32_t i = 0; i < ARRAY_SIZE; i+=4) {
          measurer->dbl[i] += dblsToCpy[i];
          measurer->fun += dblsToCpy[i];
     }
     measurer->fun /= ARRAY_SIZE;
}
...
/* Now, the code goes on to call the function dblCpy. Assume that space
for the array pointed to by measurer->dbl was allocated at some time in
the past and that all elements in the array were set to 0. The arrays
dblsToCpy and measurer->dbl are each of length ARRAY_SIZE.*/
... // dblCpy function call here
----------------------------------------------------------------------
```

a) Fill the following table:

Virtual Page Number Bits:	Virtual Address Offset Bits:

Physical Page Number Bits:	Physical Address Offset Bits:

b) Assume the TLB has just been flushed. What TLB hit to miss ratio would be encountered if sizeof(double) * ARRAY_SIZE = 256 MiB and we run the above code? Show your work.

F2) Virtual Memory (continued)

c) In the best-case scenario, how many iterations can be executed with no TLB misses? Use IEC prefixes when reporting your answer. Show your work.

[EXTRA ROOM TO SHOW YOUR WORK BELOW.]
You have to put your final result under the problem. Any final result below this text will not be graded.

SID: _____

F3) Potpourri (17 pts)

a) What is the first positive integer divisible by 5 that is not representable by floating point representation? (Hint: 2^{4n} divided by 5 gives a remainder of 1 for any integer value of n)

b) What is the fraction of integers in the range [2^{26}, 2^{27}) that are not representable in single-precision IEEE 754?

c) Encode the following data value using Hamming ECC: 0b10010001100. **Give your answer in hex.**

Hamming-Encoded value in Hex: _____

d) Suppose that we have a program that runs in 100 ns with 4 processors, but takes only 50ns when run on 8 processors. What term describes the behavior of this program as we increase the number of processors, extrapolating from the given data?

e) Which of the following choices correctly describes the given code? (Circle one). Explain your choice.

```
omp_set_num_threads(8);
#pragma omp parallel
{
     int thread_id = omp_get_thread_num();
     for(int count=0; count < ARR_SIZE/8*8; count+=8){
          arr[(thread_id%4)*2 + (thread_id/4) + count] = thread_id;
     }
}
```

A) Always correct, slower than serial
B) Always correct, speed depends on caching scheme
C) Always correct, faster than serial
D) Sometimes incorrect
E) Always incorrect

Explain:

F3) Potpourri (continued)

f) In class, we discussed the different types of RAID. Fill in the blanks with **all of the following options that apply**. Options can be reused for each question:

A) RAID1 B) RAID3 C) RAID4 D) RAID5 E) None

i) A single write can require 2 reads and 2 writes: _____

ii) Can execute independent writes in parallel: _____

iii) Uses block striping: _____

g) In RAID5, what logical gate is used to determine if new data does not match old data?

h) Consider attaching a hard disk to a CPU. Should we use polling or interrupts to deal with the hard disk? Should we use a DMA engine? Explain.

i) (1pt) **Extra Credit:** What does the Cisco logo represent?