

# CS61C MIDTERM 2

<i>Last Name (Please print clearly)</i>	
<i>First Name (Please print clearly)</i>	
<i>Student ID Number</i>	
<i>Circle the name of your Lab TA</i>	<b>Alex   Brian   Jinglin   Nate   Reese   Steven</b>
<i>Name of the person to your: Left   Right</i>	
<i>All my work is my own. I had no prior knowledge of the exam contents nor will I share the contents with others in CS61C who haven't taken it yet. (please sign)</i>	

## Instructions

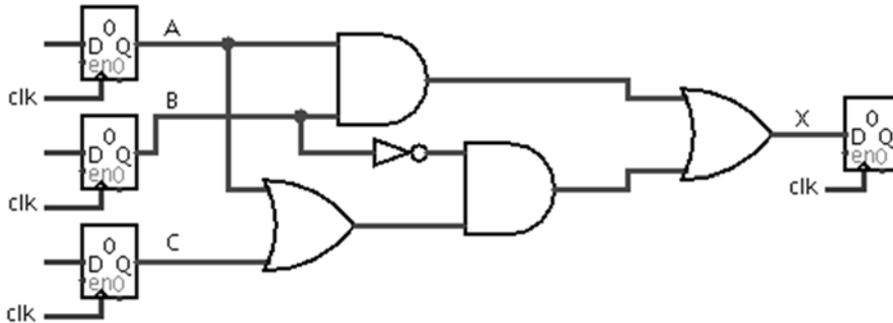
- This booklet contains 7 pages including this cover page. **The back of each page is blank and can be used for scratch work, but will not be graded** (i.e. not even scanned into Gradescope).
- Please turn off all cell phones, smartwatches, and other mobile devices. Remove all hats, headphones, and watches. Place *everything* except your writing utensil(s), cheat sheet, and beverage underneath your seat.
- You have 80 minutes to complete this exam. The exam is closed book: no computers, tablets, cell phones, wearable devices, or calculators. You are allowed one page (US Letter, double-sided) of *handwritten* notes.
- There may be partial credit for incomplete answers; write as much of the solution as you can.
- Please write your answers within the boxes and blanks provided within each problem!

Question	1	2	3	4	5	6	Total
<b>Possible Points</b>	10	10	11	11	10	4	56

If you have the time, **feel free to doodle on this front page!**

**Question 1: Synchronous Digital Systems (10 pts)**

Consider the circuit below for the following questions. Logical gates incur a 10 ns combinational logic delay. Registers have a CLK-to-Q delay of 5 ns and a setup time constraint of 15 ns.



A	B	C	X
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

a) Write the boolean expression for X in terms of A, B, and C. Only use the AND, OR and NOT logical operators, and leave the expression in its UNSIMPLIFIED form. (You are welcome to use the provided truth table, but it is **NOT** required and **NOT** graded.)

b) Given the following boolean expression for a different circuit, simplify it to use the *fewest* possible AND, OR, and NOT logical operators.

$$x = \overline{a + d} + a \cdot \bar{d} + \bar{b} \cdot \bar{c}$$

c) For the circuit above (part a), calculate the minimum clock period that will allow the circuit to function correctly. Remember to include units.

e) Assuming the hold time constraint of the registers is 20 ns, calculate the minimum combinational logic delay needed per logic gate to allow the circuit (part a) to function correctly. Remember to include units.

### Question 2: Datapath and Control (10 pts)

Consider adding the new instruction `lweq` to our existing MIPS datapath. This instruction is intended to load into `$rd` a word from memory at the address specified by the contents of `$rs` offset by the `shamt` value if the values in `$rt` and `$rs` match. Below you'll find the RTL-esque description as well as a set of regular MIPS instructions that are equivalent to a call to `lweq`.

Instruction	Operation
<code>lweq rd, rs, rt, shamt</code>	$\text{if } (R[rs] == R[rt]) \text{ RF}[rd] \leftarrow M[ R[rs] + (\text{SignExt}(\text{shamt}) \ll 2) ]$

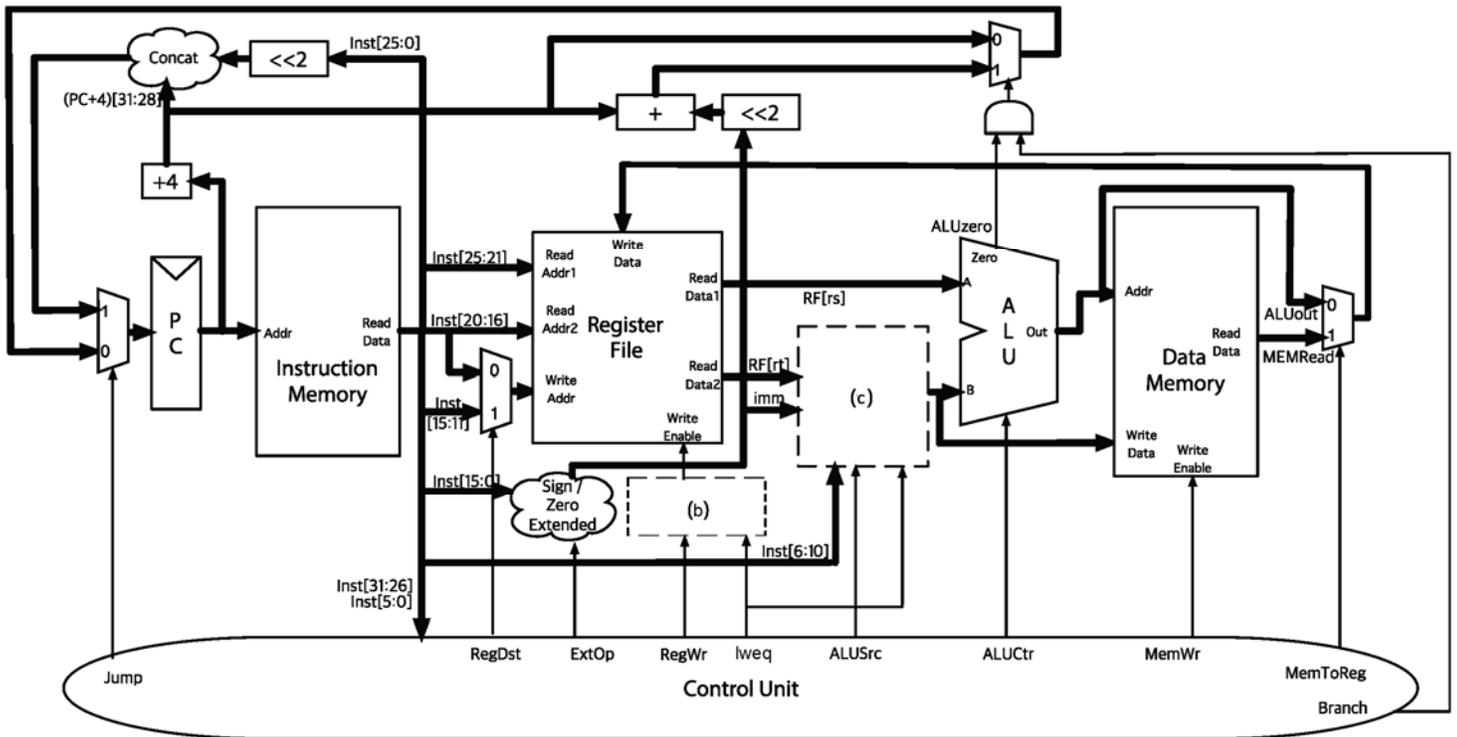
```

lweq $t0 $t1 $t2 -2    ⇔    bne $t1 $t2 skip
                               lw $t0 -8($t1)
                               skip: # next instruction
    
```

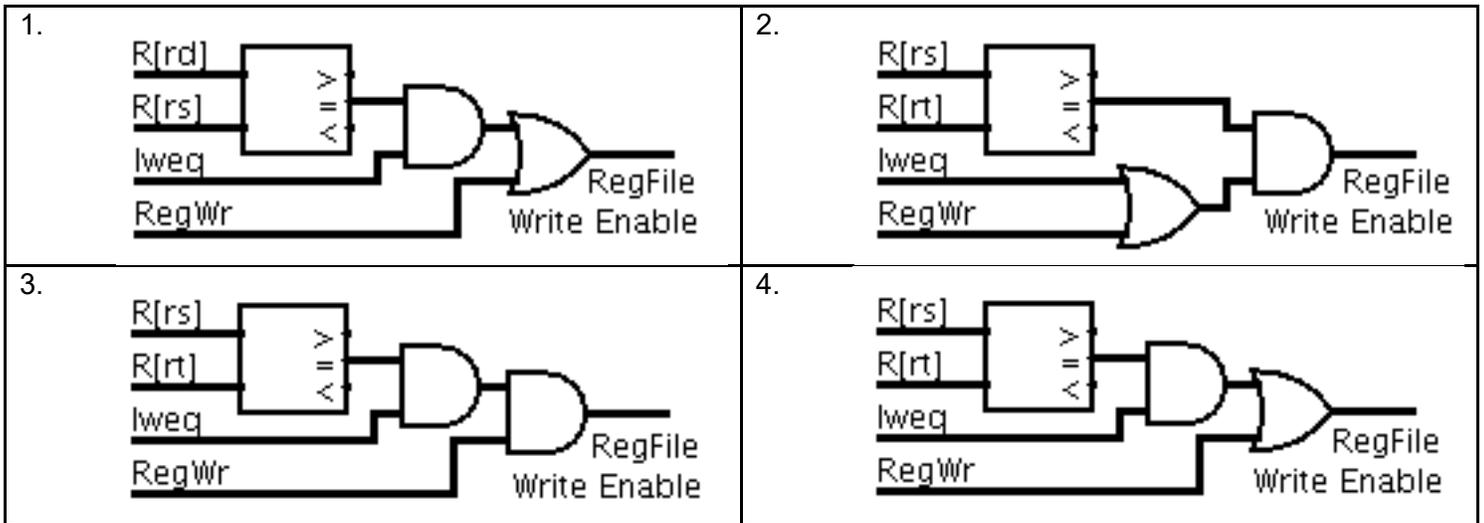
- a) If we want to use the pre-existing R-type instruction format for `lweq` and decide to repurpose the `shamt` field to hold the memory address offset, what is the largest and smallest possible byte offset that can be applied to the value stored in `$rs`?

Most Positive:
Most Negative:

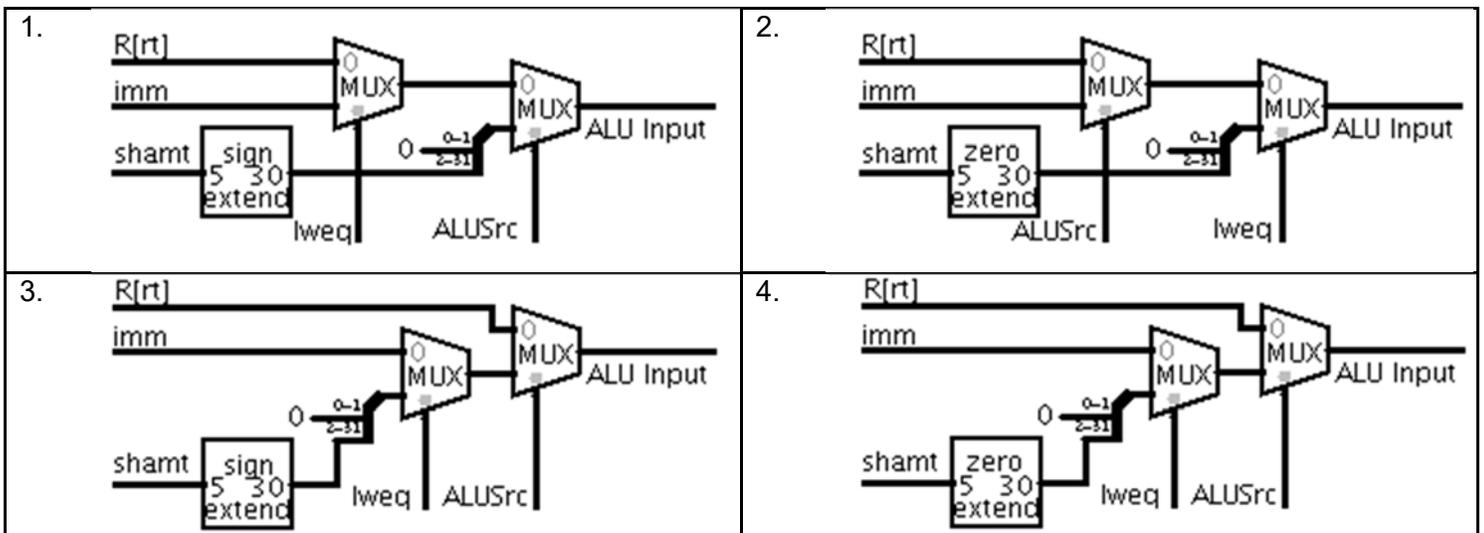
Based on the MIPS datapath diagram shown below, choose the modifications required in the dashed boxes to execute the new instruction correctly.



b) For box (b), circle the number below for the circuitry required:



c) For box (c), circle the number below for the circuitry required:



d) Based on the MIPS datapath diagram shown on the previous page, fill in the control signal values necessary to execute the `lweq` instruction correctly.

lweq	RegDst	ExtOp	MEMWr	MemToReg	Jump	Branch
1						



**Question 4: Caches (11 pts)**

We have a 64 KiB address space and two different caches. Both are 1 KiB, direct-mapped caches with random replacement and write-back policies. **Cache X** uses 64 B blocks and **Cache Y** uses 256 B blocks.

a) Calculate the TIO address breakdown for **Cache X**:

Tag	Index	Offset

b) During some part of a running program, **Cache Y**'s management bits are as shown below. Four options for the next two memory accesses are given (R = read, W = write). Circle the option that results in data from the cache being *written to memory*.

Slot	Valid	Dirty	Tag
00	0	0	1000 01
01	1	1	0101 01
10	1	0	1110 00
11	0	0	0000 11

(1) R 0x4C00, W 0x5C00

(2) W 0x5500, W 0x7A00

(3) W 0x2300, R 0x0F00

(4) R 0x3000, R 0x3000

c) The code snippet below loops through a character array. Give the value of LEAP that results in a Hit Rate of 15/16 for **Cache Y**.

```
#define ARRAY_SIZE 8192
char string[ARRAY_SIZE];           // &string = 0x8000
for(i = 0; i < ARRAY_SIZE; i += LEAP) {
    string[i] |= 0x20;              // to lower
}
```

d) For the loop shown in part (c), let LEAP = 64. Circle ONE of the following changes that increases the hit rate of **Cache X**:

Increase Block Size

Increase Cache Size

Add a L2\$

Increase LEAP

e) For the following cache access parameters, calculate the AMAT. All miss and hit rates are local to that cache level. Please simplify and include units.

L1\$ Hit Time	L1\$ Miss Rate	L2\$ Hit Time	L2\$ Hit Rate	MEM Hit Time
2 ns	40%	20 ns	95%	400 ns

**Question 5: Floating Point (10 pts)**

---

Assume integers and IEEE 754 single precision floating point are **32 bits wide**.

a) Convert from IEEE 754 to decimal: **0xC0900000**

b) What is the smallest *positive* integer that is a power of 2 that can be represented in IEEE 754 but not as a signed int? You may leave your answer as a power of 2.

c) What is the *smallest positive* integer  $x$  such that  $x + 0.25$  can't be represented? You may leave your answer as a power of 2.

d) We have the following word of data: **0xFFC00000**. Circle the number representation below that results in the *most negative number*.

Unsigned Integer

Two's Complement

Floating Point

e) If we decide to stray away from IEEE 754 format by making our Exponent field 10 bits wide and our Mantissa field 21 bits wide. This gives us (circle one):

MORE PRECISION // LESS PRECISION

**Question 6: Performance (4 pts)**

---

We are using a processor with a clock period of 1 ns.

a) Program A contains 1000 instructions with a CPI of 1.2. What is the CPU time spent executing program A?

b) Program B contains 500 instructions but accesses memory more frequently, what is the maximum CPI that program B can have without executing slower than program A?