Name: (as it would appear on official course roster)
UCSB email address: **@ucsb.edu**
Lab Section Time:
Optional: name you wish to be called if different from above
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone")

Lab 02: Binary Arithmetic & Introduction to MIPS

Assigned: *Thursday, January 16th, 2020*
Due: *Tuesday, January 21st, 2020*
Points: *60 (PDF questions), 11 (Programming questions)*

- You may collaborate on this homework with AT MOST one person, an optional "homework buddy".
- MAY ONLY BE TURNED ON **GRADESCOPE as a PDF file**.
- There is NO MAKEUP for missed assignments.
- We are strict about enforcing the LATE POLICY for all assignments (see syllabus).

Don't use a calculator or online solvers when working these problems. You will not be able to use them in exams either, so it's good practice to know how to do these!

Binary Subtraction

The following problems ask you to subtract one binary number from another. While this can be done directly, it is typically easier to negate the second number (using the 2's complement method), and then add the two together. **You must express the answer in 8 bits.** There will be several steps, but only show the answer. You will ALSO NEED TO SPECIFIY the overflow (**V**) and carry (**C**) bits.

1. What is

```
    11001101
  - 01100100
```

   _____(answer)

2. What is

```
    01101001
  - 10110101
```

   _____(answer)

3. What is

```
    10010110
  - 11111111
```

   _____(answer)

Name:
(as it would appear on official course roster)

4. What is

```
   10110010
 - 00000001
```

_____(answer)

Basic Bitwise Operations

Recall that bitwise AND is specified with `&`, bitwise OR with `|`, and bitwise XOR with `^`.

5. What is

```
   10010011
 & 10111011
```

_____(answer)

6. What is

```
   11100001
 & 10010001
```

_____(answer)

7. What is

```
   10011011
 | 11100101
```

_____(answer)

8. What is

```
   11110101
 ^ 00001011
```

_____(answer)

Give your answers for the following set in **2-digit hexadecimal**. Recall that bitwise AND is specified with `&`, bitwise OR with `|`, and bitwise XOR with `^`.

9. Calculate **0x05 & 0xF0** _____(answer)
10. Calculate **0x2A & 0x55** _____(answer)
11. Calculate **0x5D | 0xAA** _____(answer)
12. Calculate **0xBE | 0xEF** _____(answer)

Name:

(as it would appear on official course roster)

13. Calculate **0x79 ^ 0x97** _____(answer)

14. Calculate **0x16 ^ 0x9C** _____(answer)

Give your answers for the following set in **8-bit binary.**

15. What is 11010001 >> 1 for logical shift right? _____(answer)

16. What is 11010111 >> 1 for arithmetic shift right? _____(answer)

17. What is 10010011 >> 2 for logical shift right? _____(answer)

18. What is 10010011 >> 2 for arithmetic shift right? _____(answer)

19. What is 10010011 << 2 for shift left? _____(answer)

20. What is 00000011 << 2 for shift left? _____(answer)

Advanced Bitwise Operations

The entries shown as 'X' are general binary values (i.e. they can be either '1' or '0'). Your answer can have '1' or '0' or 'X' in it.

21. What is

```
  XXXXXXXX
| 11001010
```

_____(answer)

22. What is

```
  XXXXXXXX
& 10110110
```

_____(answer)

Name:

(as it would appear on official course roster)

For the following three questions, remember that bit numbers start numbering from 0. Assume that these hexadecimal numbers are always unsigned. In addition to showing the mask you would use, also show the operation you would use, like so: **& 0x0020**. (i.e., AND the original number with the hexadecimal mask 0x0020.)

23. Specify the mask you would need to inspect **bit 13** of the unknown number. Express it as a 4-digit hexadecimal number (i.e. in the form of 0xhhhh).

24. Specify the mask you would need to set just **bit 13** of the unknown number to zero. That is, the result of this operation results in a new number, which the unknown number should be subsequently set to. Again, express it as a 4-digit hexadecimal number.

25. Specify the mask you would need to set **bit 13** of the unknown number to one. That is, the result of this operation results in a new number, which the unknown number should be subsequently set to. Express it as a 4-digit hexadecimal number.

Name:

(as it would appear on official course roster)

26. (8 pts) Complete the MIPS assembly language program below that is supposed to take the values in **$t0**, **$t1**, and **$t2** and first calculate the value of **($t1 + $t2 - $t0)**, then logic **AND** it to the value of **($t0 + $t2)**. The entire calculated value should be placed in **$t3**. Finally, the program should print the value of **$t3** to *standard output* and exit correctly. You should try it out and test it, like the code(s) I demonstrated in class to be sure it has the correct syntax.

```
.text
main:
      li $t0, 42
      li $t1, 83
      li $t2, 214
# To-Do: the rest of the program here (except the last 2 lines)!
```

```
      li $v0, 10
      syscall
```

27. (2pts) What is the value of **$t3** (in 32-bit hexadecimal) after the above program is executed? Show it with a hand-calculation (i.e. not by running the program only, but by calculating it yourself).

Name:
(as it would appear on official course roster)

Bitwise Operations in C/C++

See the lab description on our main class website for details on these next tasks. This requires some programming knowledge of C/C++ (which you should know from CS 16, etc…) You will complete these C/C++ programs and then upload them to **Gradescope**, per the instructions on **lab02** on the website.