Computer Organization
Assignment 3

Due date: September 8, 2022

Keep your answers concise as well as neat & clean.
Submit your MIPS codes via this Google Form: https://forms.gle/pmTYXUwuF9p3jiob9.
All MIPS codes should run as is in QtSPIM[1]. Clearly state your assumptions (if any).
Some MIPS resources are available at:
https://www.isical.ac.in/~rathin_r/uploads/CO/2022/MIPS_resources.html
[backup] https://ratcoinc.github.io/MIPS/

1. Given a four-variable switching function represented in the canonical PoS (Product of Sums) form as:

$$f(a, b, c, d) = \Pi(5, 7, 10, 12, 14, 15) + \Phi(1, 4, 13)$$

where $\Phi(i)$ denotes a don't-care condition that means for the input combination with binary value i, the output is to be ignored.

   i. Minimize the function using K-map.

   ii. Represent the same function $f(a, b, c, d)$ in canonical SoP form and minimize it.

   iii. Implement both minimized functions obtained in i and ii using AND, OR and NOT gates. With respect to hardware complexity which one is better? Justify your answer.

$[5 + (4 + 4) + (3 + 3 + 1) = 20]$

2. Recall that MIPS does not provide any instruction to obtain 1's complement of a bit pattern.

   i. Devise an *efficient*[2] way to obtain 1's complement of an integer in MIPS using only the existing instruction(s). Furthermore, you are restricted from specifying any constant explicitly (thus you cannot do $X \oplus 1$).

   ii. Now write MIPS code to load the year of your birth (YYYY) in $\$t0$ and then compute its 1's complement into the same $\$t0$ using your solution of part i. $[5 + 5 = 10]$

3. Let A be a constant that denotes the last two digits of your roll number (day of birth in case of JRFs). Now let $D = \max(A, 60-A)$. Suppose you need to load this value D into a register of the MIPS processor, but you cannot explicitly specify any constant values in your code (thus cannot use any immediate mode instructions and some others). Write a minimal MIPS code to load D into $\$t0$.

   [hint: MIPS provides the constant 0 in $\$zero/\$0$, if you can somehow generate +1 out of this 0, you are done. You might want to explore the arithmetic and logic operations.] [15]

4. i. Write an efficient C (or similar) function which receives a positive integer n as an argument then counts the number of 1s in n and returns that count.

   ii. Suppose register $\$t0$ contains n. Write an *efficient* MIPS code to count the number of 1s in n and store the count into the memory location X interpreted in HEX, where $X = 10000000 + 32 * A$ (A is defined in ques 3). That is if $A = 60$, the value of X is `0x10001920` in HEX.

   iii. Comment on the time complexity of your solution. $[5 + 5 + 2 = 12]$

5. Suppose there are n distinct integers all in the closed interval of $[0, n]$, that is only one number is absent, and all others occur exactly once. Your task is to find the missing number efficiently.

   i. Write an efficient C (or similar) function for this. Note that the value of n can be very high, say INT_MAX[3], so arithmetic operations may result into overflows[4].

   ii. Suppose, memory location `0x10001000` contains n and the given numbers reside in n consecutive locations starting at `0x10002000`. Write a MIPS code that writes the missing number at the memory location `0x10003000` using the method devised in part i. $[5 + 8 = 13]$

[1] QtSPIM version 9.1.23 available at: https://sourceforge.net/projects/spimsimulator/files/

[2] *efficient* means lesser number of instructions and/or involves lesser data transfer

[3] Sizes of various built-in integral types of C/C++: https://cplusplus.com/reference/climits/

[4] https://en.wikipedia.org/wiki/Integer_overflow#Origin