Indian Institute of Technology Kharagpur

Department of Computer Science and Engineering

Exam-1, Autumn 2021-22

Computer Organization and Architecture (CS31007)

Students: 133 **Date**: 06-September-2021

Full marks: 50 **Time**: 12:00 noon – 1:45 PM

Credit: 30%

INSTRUCTIONS: This is an OPEN-BOOK, OPEN-NOTES test. The questions are such that they require either numerical answers or very short answers (in a few sentences or a few lines of code). Please write ONLY THE ANSWERS on a sheet of paper, scan it (or take a photo), and submit the image of the solution sheet on Moodle. DO NOT FORGET TO WRITE YOUR NAME AND ROLL NUMBER AT THE TOP OF YOUR ANSWER SHEET. You may use calculators if required. ANSWER ALL QUESTIONS.

1. In an assembly-language program P, 20% instructions require sequential execution because of data dependence. Two machines $M1$ and $M2$ are given: $M1$ comprises one core, i.e., a single processor; $M2$ is a multi-core machine with 80 copies of $M1$ on-chip that are capable of performing parallel processing. Both of them operate with the same clock frequency.

   (a) Calculate the maximum achievable speed-up with respect to the execution time needed by the new machine $M2$ compared to that of $M1$ for the program P, following *Amdahl's Law.* [**2**]

   (b) Next, compute the same following the *Gustavson-Barsis Law.* If the estimated speed-up values differ in these two cases, explain the reason behind them. For machine $M2$, you may ignore inter-processor communication time and other overheads. [**4**]

2. Consider the following MIPS code segment being executed on machine M:

```
      lw $t1, 10($t2)
      srl $t1, $t1, 16
      lui $t3 0x7FFF
      ori $t3 0xFFFF
Loop: xor $t4, $t1, $t3
      add $t1, $t1, $t4
      sll $t1, $t1, 16
      bgt $t1, $zero,    Loop
      sw  $t1, 1000($t3)
```

   Assume M has a clock frequency is 3.5 GHz. Also, assume that `lw` needs 5 clock cycles, `sw` 4 clock cycles, `bne` 2 clock cycles, and all other instructions need 3 clock cycles to execute. Calculate the total amount of CPU-time (in nanoseconds) required to execute the above code. [**6**]

3. Consider the following C function:

```
int func (int x, int y, int z) {
  /* All integer variables are 32-bit long */
  int f, MASK1, MASK2;
  MASK1 = (((!x) ^ 0x01) << 31) >> 31; /* ^ denotes bitwise XOR operation */
  MASK2 = ((!x) << 31) >> 31;
  f = (y & MASK1) + (z & MASK2);
  return f;
}
```

   (a) Express the relationship between `f`, `x`, `y` and `z` using a single C programming language statement, without involving any other variable or constant. [**2**]

(b) Write an equivalent MIPS code segment using maximum four instructions, assuming variables `x`, `y`, `z` and `f` are bound to registers `$t1`, `$t2`, `$t3` and `$t0` respectively. There is no need to write a MIPS procedure body. [**4**]

4. For a given program to be executed, a RISC machine is likely to offer you the following features compared with a CISC machine (choose the correct option):

   (a) More instruction count, higher CPI, lower chip yield, better performance;

   (b) Lesser instruction count, higher CPI, higher chip yield, better performance;

   (c) More instruction count, less CPI, lower chip yield, worse performance;

   (d) More instruction count, less CPI, higher chip yield, better performance;

   (e) None of the above.

   Assume the same value of cock cycle time (CCT) for both machines. [**2**]

5. Let the content of register `$t1` represent a 2's complement number N. Write a MIPS code fragment such that the register `$t2` stores the absolute value of N. In other words, register `$t2` has a copy of register `$t1` if N is positive, and `$t2` should hold the negative of N, if N is negative. (Hint: you are encouraged to attempt it using at most three instructions, however you can write longer code if necessary). [**5**]

6. Read the following MIPS program and answer the questions that follow:

```
##########Data Segment########################
.data
msg:
      .asciiz "The value is: "
newline:
      .asciiz "\n"
###########Code Segment#####################
.text
.globl main
main:
    li $a0, 81 # first value, hard-coded
    li $a1, 24 # second value, hard-coded
subtract_loop: sub $t1, $a0, $a1
               ble $t1, $zero, check_condition
               move $a0, $t1
               j   subtract_loop
check_condition: beq $t1, $zero, print_val
# otherwise, swap
             move $t1, $a0
             move $a0, $a1
             move $a1, $t1
             j    subtract_loop
print_val:    la $a0, msg
              li $v0,4
              syscall # print a message
              move $a0, $a1
              li $v0, 1
              syscall
# Exit
              li $v0, 10
              syscall
```

   (a) What does the program compute in general? [**3**]

   (b) What would be the output of this run of the program? [**1**]

7. A student wanted to load the constant 0x1234dcba in the register `$t0`. For this, he wrote the following MIPS code:

```
lui  $t0, 0x1234  # to get MSB-side half-word
addi $t0, $t0, 0xdcba # to get LSB side half-word
```

However, when the student assembled and executed the program, he did not find the expected value in register `$t0`. Can you explain (in 1-2 sentences) what went wrong, and what value did he actually find in register $t0? (Note: no credit will be given for the second part of the question, if the explanation provided for the first part is incorrect.) [**2 + 3 = 5**]

8. Write a MIPS code segment to initialize the `$t1` register to the value 71234_{10}. You should use maximum two MIPS instructions, and you are not allowed to use any pseudoinstruction. [**4**]

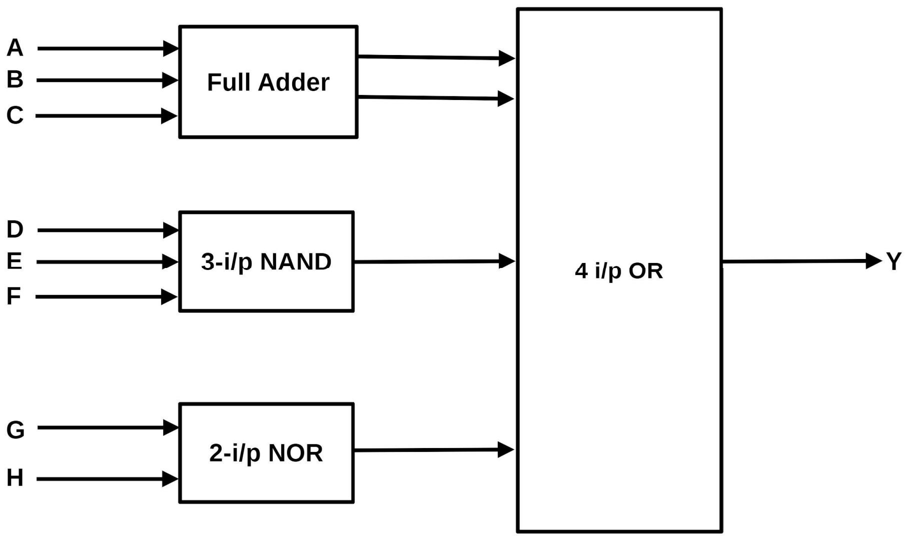

Figure 1: A simple eight-input combinational logic circuit.

9. Consider the logic circuit shown in Fig. 1, with eight Boolean inputs: A through H, and one Boolean output Y. The functionalities of the individual modules in the given circuit are self-evident. Out of 256 possible input combinations, how many would cause the output Y to assume logic value 1? Give brief justification in support of your answer. Note that no credit would be given for the first part of the question, of the explanation provided for the second part is incorrect.
(Hint: Please **do not** try to construct the truth-table for an eight variables Boolean function, or simplify the Boolean expressions for Y, as those approaches would be time-consuming.) [**2 + 5 = 7**]

10. Consider a null-terminated string of lower-case letters stored in contiguous bytes in memory, with the starting address of the string in register `$a0`. Write a complete MIPS procedure `toupper` that reads the characters of the string from memory, converts each to upper-case, and then stores it back to memory, overwriting the original string. The procedure need not use any callee-saved register during execution. Assume the string does not include any non-alphabetic character. (Hint: ASCII code for 'A' is 65, and ASCII code for 'a' is 97.) [**5**]