Indian Institute of Technology Kharagpur

AUTUMN Semester, 2020-2021
COMPUTER SCIENCE AND ENGINEERING

Computer Organization Laboratory

Assignment-6: MIPS-32 Assembly Language Programming

Full Marks: 20

Time allowed: 6 hours

INSTRUCTIONS: **ATTEMPT ALL PROBLEMS. Make one submission per group of your source code on Moodle. Name your submitted source files following the format** `Assgn_6_Prob_1_Grp_<Group_no>.s` **(e.g.** `Assgn_6_Prob_1_Grp_25.s`**), etc. Inside each submitted file, there should be a clear header describing the assignment no., problem no., semester, group no., and names of group members. Liberally comment your code to improve its comprehensibility. Finally, submit a single zipped folder containing only your source files.**

1. Write a complete MIPS-32 program to collect two positive integers from the user, and then to calculate and display their **GCD by repeated subtraction. Display proper messages while collecting the input arguments from the user and and displaying the calculated value. DO NOT USE ANY INTEGER DIVISION INSTRUCTION, EVEN IF IT IS AVAILABLE IN MIPS-32**. After the two input numbers are collected from the user, there should be sanity checking to ensure that both the integers are positive. Your program may or may not have a procedure call. (5 marks)

2. Write a complete MIPS-32 program to collect a positive integer $n \geq 2$ from the user, and then to determine whether it is prime or composite. After the input number is collected from the user, there should be sanity checking to ensure that the integer is ≥ 2. Use any algorithm that you can think of (which might not be the most efficient one available). Display the finding of your program with a proper message at the end. Your program may or may not have a procedure call. (5 marks)

3. Write a complete MIPS-32 program to collect two 16-bit unsigned integers from the user, and then to determine their product using the **Sequential Unsigned Binary Multiplication (left-shift version) Algorithm** that we have covered earlier. After the input arguments are collected from the user, there should be sanity checking to ensure that they are within the proper numerical range. Display the product at the end of your program with a proper message. Your program should have a procedure call `seq_mult_unsigned`, with the two input arguments available in registers `$a0` and `$a1`, and the final product available in register `$v0`. (5 marks)

4. Write a complete MIPS-32 program to collect two 16-bit signed integers (encoded in 2's complement form) from the user, and then to determine their product using the **Booth's Multiplication Algorithm**. After the input arguments are collected from the user, there should be sanity checking to ensure that they are within the proper numerical range. Display the product at the end of your program with a proper message. Your program should have a procedure call `seq_mult_booth`, with the two input arguments available in registers `$a0` and `$a1`, and the final product available in register `$v0`. (5 marks)