Indian Institute of Technology Kharagpur

AUTUMN Semester, 2019
COMPUTER SCIENCE AND ENGINEERING

Computer Organization Laboratory

Assignment-4: MIPS-32 Assembly Language Programming

Full Marks: 20

Time allowed: 6 hours

INSTRUCTIONS: ATTEMPT BOTH PROBLEMS. Make one submission per group of your source code on Moodle. Name your submitted source files following the format `Assgn_4_Prob_1_Grp_<Group_no>.s` **(e.g.** `Assgn_4_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.**

1. [**Insertion Sort in MIPS-32**] Write a MIPS-32 program to collect an array of eight integers from the user, and then sort the array in descending order using the *Insertion Sort* algorithm. Collect the numbers from the input console using a loop, and store in memory in an array called `array`. Do not store the numbers as scalars in eight different non-contiguous locations or in eight different registers. Write a function `InsertionSort` to implement the sorting algorithm. Pass the address of the 1-D array and the required parameters while implementing the function. After sorting, print the sorted array on the console with a proper message. (10 marks)

2. [**Saddle Point in MIPS-32**] Write a MIPS-32 program which identifies the saddle point in an 2-D matrix. In this program, create an 2-D integer array `twoDarray` of size 4 × 4, and populate it in the memory with integer values. Collect the numbers from the input console using a loop, and store in the memory contiguously to populate `twoDarray`. Do not store the numbers as scalars in different non-contiguous locations or in different registers. Find the saddle point(s) in the 2-D array `twoDarray` and print on the terminal. An element is a saddle point if it is the unique maximum (or minimum) in its row but the unique minimum (resp., maximum) in its column. If there is none, then print that message, otherwise print the saddle point value and the indices of the saddle point. Write a function `FindSaddle`, which accomplishes the task of finding saddle points. Pass the address of the 2-D array and the required parameters while implementing the function. (10 marks)