2014

M.Sc.

3rd Semester Examination

ELECTRONICS

PAPER—ELC-305 (Set-2)

(PRACTICAL)

Full Marks : 50

Time : 3 Hours

The figures in the right-hand margin indicate full marks.

Candidates are required to give their answers in their own words as far as practicable.

(*Microprocessor Programming*)

Answer any *one* question, selecting it by a lucky draw.

1. Write an assembly language program to calculate the square root of a given number with the following considerations :

   (a) check the number to determine whether it is positive or negative.

   (b) reject the negative number.

   (c) store the negative number and a message FFH in the consecutive memory locations.

   (d) calculate the square root of the positive number.

(e) store the number and the result in consecutive memory locations if the number is a perfect square ; otherwise an error message FEH in the result memory location.

Perform the experiment for five different numbers.

2. Write an assembly language program to compare two blocks of data bytes containing 10 bytes in each blocks. The blocks are started from memory locations XX50H and XX60H respectively. If a mismatch is found during comparison between two bytes, then store the two mismatch bytes alongwith the addresses of these data bytes. Assume a single mismatch to be obvious there.

Repeat the experiment using three sets of data.

3. A set of ten numbers is stored in memory locations starting at XX60H Write an assembly language program to :

(a) check each reading to determine whether it is positive or negative.

(b) reject all negative readings.

(c) add all positive readings.

(d) store FFH in the memory location XX70H when the sum exceeds eight bits to indicate overload ; otherwise, store the sum.

4. Write an assembly language program to convert binary-to-ACSII Hex code, following the condition stated below :

   "An 8-bit number is stored in a memory location. Transfer the byte to the accumulator. Separate the two nibbles. Call the subroutine to convert each nibble into ASCII Hex code. Store the codes in consecutive memory locations. Write the subroutine to convert a binary digit into ASCII Hex code."

   Repeat the experiment for five different numbers.

5. Write an assembly language program to find out the LCM of three 8-bit numbers stored in three consecutive memory locations starting from XX50H. Store the result in a memory location XX60H.

   Repeat the experiment with five different sets of data.

6. Two consecutive memory locations contain two numbers P and Q. Write an assembly language program to divide X by Y (without using repeated sustraction). Store the quotient and remainder in two consecutive memory locations.

   Repeat the experiment with 3 sets of dividend and divisor.

7. Write an assembly language program to add two 3×3 matrices and store the result in successive memory locations starting from XX50H.

   Repeat the experiment with two sets of data.

8. Write an assembly language program to blink an LED connected with 8085 μp by PPI 8255A interfacing.

9. An array of ten positive and negative numbers is stored in memory area in sign magnitude form. Write an assembly language program to count for the negative 8-bit numbers within the array and store only the negative numbers in a separate memory area in a 2's complement form.

   Repeat the experiment for three different arrays.

10. Write an assembly language program to multiply two 8-bit numbers using Left shift and Add method. The numbers are stored in two consecutive memory locations. Store the product in a suitable memory location. Find also X = P * Q - R, where P, Q and R are all 8-bit numbers. The number R may be available from a suitable memory location. Register the value of X into the location just after a location where the result of product P * Q is stored.

    Repeat the experiment using two sets of P, Q and R values.

11. Write an assembly language program to find the 2's complement of an 8-bit number using the following algorithm :

    "Copy all bits starting from LSB of the byte upto the first 1 bit of the number and then complement all the bits up to MSB".

    Repeat the experiment for 5 numbers.

12. Write an assembly language program to calculate the factorial of a whole number. Store the number and its result in consecutive memory locations. Find out the values for the number 0 and four other numbers selecting from 2 to 7.

13. A set of ten bytes are stored in memory locations starting with the address XX50H. Write an assembly language program to check each byte, and save the bytes that are higher than 60_{10} and lower than 100_{10} in memory locations starting from XX60H. Arrange the sorted numbers in ascending order from the memory location XX70H.

14. Write an assembly language program to transfer the bytes from a block of 16 bytes stored from location X050H to X05FH following the condition stated below :

    "The byte should be even but the upper nibble of the byte should be odd". The starting address of the destination should be X300H.

15. Write an assembly language program to convert a 2-digit BCD number stored at memory location X200H into its binary equivalent number and register the result in the memory address X300H.

    Repeat the experiment for 5 different data.

Distribution of Marks

Flow Chart	:	05 Marks
Assembly language program	:	10 Marks
Execution of the program	:	10 Marks
Result	:	05 Marks
Description of the program	:	05 Marks
Viva-voce	:	10 Marks
Laboratory note book	:	05 Marks
Total	:	50 Marks