PG/IIIS/EL-2111/10 (Pr.)

M.Sc. 3rd Semester Examination, 2010

ELECTRONICS

(Microprocessor Programming)

PAPER—EL-2111

(Practical)

Full Marks : 50

Time : 3 hours

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

1. Write an assembly language program to compare two blocks of data bytes containing 10 bytes in each block. The blocks are started from two different memory locations. If a mismatch is found during comparison between two bytes, then store the two mismatch bytes along with the addresses of these data bytes. Assume a single mismatch to be obvious there. Repeat the experiment using three sets of data.

2. Two consecutive memory locations contain two 8-bit numbers X and Y. Write an assembly language program to divide X by Y (without using repeated subtraction). Store the quotient and remainder in two consecutive memory locations. Repeat the experiment with three sets of dividend and divisor.

3. Write an assembly language program to multiply two 8-bit numbers using Left-shift and Add method. The numbers are stored at 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.

4. An array of ten positive and negative numbers is stored in a 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 this for five different arrays.

5. Write an assembly language program to find a factorial of a whole number N. Store the number and its result in consecutive memory locations. Verify your result for the number 0 and four other numbers selecting from 2 to 7.

6. Write an assembly language program to calculate the square root of a given real number. Store the number and FFH if the number is a perfect square; otherwise an error message FEH, and the result of the perfect square number in the consecutive memory locations. Execute the program for five different numbers.

7. Two 8-bit numbers are stored in two consecutive memory locations. Write an assembly language program to find the highest common factor (HCF) of the two numbers. Store the result in a memory location just after the memory locations of the two numbers. Repeat the process with three different sets of data.

8. An 8-bit binary number is stored in a memory location. Write an assembly language program to find the Gray code corresponding to the binary number and store the Gray code in a suitable memory location. Repeat this for six different binary numbers.

9. An 8-bit Gray code is available from a memory location. Write an assembly language program to find the binary code corresponding to the Gray code and store the result in a suitable memory location. Repeat this for six different Gray codes.

10. A set of ten current readings is stored in memory locations starting at XX60H. The readings are expected to be positive ($< 127_{10}$). Write an assembly language program to :

(*i*) Check each reading to determine whether it is positive or negative.

(*ii*) Reject all negative readings.

(*iii*) Add all positive readings.

(*iv*) Store FFH in the memory location XX70H when the sum exceeds eight bits ; otherwise, store the sum.

Data (H) : 28, D8, C2, 21, 24, 30, 2F, 19, F2, 9F.

11. A set of ten bytes are stored in memory starting with 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. Sort the selected bytes in ascending order. Data (H): 6F, 28, 5A, 49, C7, 3F, 37, 4B, 78, 64.

12. Write an assembly language program to convert a 2-digit BCD number into its equivalent binary number. The BCD number is available from a memory location NUMB and store the converted binary number into the location NUMB + 1. Repeat the process for five numbers.

13. Write an assembly language program to calculate the square of a given number ($< 16_{10}$) using the following algorithm :

Step I : Square $\leftarrow 0$, Count $\leftarrow$ given number
Odd $\leftarrow 1$

Step II : Square $\leftarrow$ Square + Odd

Step III : Count $\leftarrow$ Count - 1

Step IV : If Count $= 0$, then go to step VI

Step V : Odd $\leftarrow$ Odd + 2

Step VI : Store the current value of square.

The number to be squared is available from a memory location NUMB and store square in a memory location RESULT. Repeat this experiment for five different numbers.

14. Interfacing two LEDs using common anode technique, write an assembly language program to blink them alternately. Assume 8255 A is interfaced in IO mapped IO.

Distribution of Marks

Description of the program	: 05 Marks
Flowchart	: 05 Marks
Assembly language program	: 10 Marks
Execution of the program	: 15 Marks
Viva-voce	: 10 Marks
Laboratory Note Book	: 05 Marks
Total	: 50 Marks