PG/IIIS/EL - 2111/09 (Pr.)

M.Sc. 3rd Semester Examination, 2009

ELECTRONICS

(*Microprocessor Programming*)

PAPER—EL-2111

(Practical)

Full Marks : 50

Time : 3 hours

Answer any **one** question

Distribution of Marks :

Experiment	:	35 Marks
Viva voce	:	10 Marks,
Laboratory Note Book	:	5 Marks
Total	:	50 Marks

1. An array of ten bytes are stored from a memory location ARRAY. Write an assembly language program to develop another new array starting from a suitable memory location selecting only even bytes from the given array. Repeat this process for three different arrays.

2. An array of bytes are stored from a memory location ARRAY. The length of the array is stored in a location with address ARRAY - 1. Write an assembly language program to develop a new array starting from a suitable memory location selecting those bytes which are divisible by 4. Repeat this with 5 different arrays.

3. An array of ten bytes of value of each byte is less than 60_D are stored from a memory location ARRAY. Write an assembly language program to develop a new array with bytes which are twice the byte of the given array. Repeat this with 5 different arrays.

4. One 8-bit binary number is available from a memory location. Find the Gray code corresponding to the binary number and store the Gray code in a suitable memory location. Repeat this for 5 different binary numbers.

5. One 8-bit Gray code is available from a memory location. Find the binary code corresponding to the Gray code and store the result in a suitable memory location. Repeat this for 5 different Gray code.

6. An array contains ten bytes. All the bytes are in sign magnitude form. Write an assembly language program to find out the negative numbers present in the array and develop a new array containing the 2's complement of the negative numbers. Repeat for 5 different arrays.

7. Two consecutive memory locations contain two numbers X_1 and X_2. Write an assembly language program to divide X_1 by X_2 (without using repeated subtraction). Store the quotient and remainder in two consecutive memory locations. Repeat this for 3 sets dividend and divisor.

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

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

   Step 2 : Square $\leftarrow$ Square + Odd

Step 3 : Count ← Count - 1

Step 4 : If count = 0, then go to step 6

Step 5 : Odd ← Odd + 2

Step 6 : 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 5 numbers.

9. Write an assembly language program to find the 2's complement of an 8-bit numbers using the following algorithm :
"Copy all the bits starting from the L.S.B. of the byte up to the first 1 bit of the number and then complement all the bits up to M.S.B.." Repeat this experiment for 5 numbers.

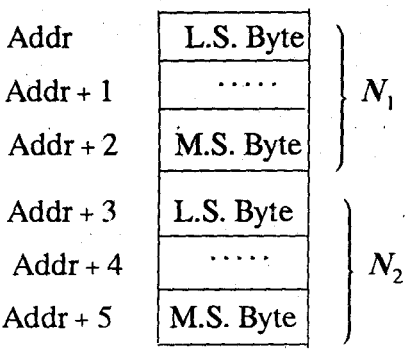
10. Write an assembly language program to convert a 2-digit BCD number into its equivalent binary number. The BCD number is available from a memory location NUMB and store the converted binary number into the location NUMB + 1.

11. Write an assembly language program to compute :

$$S = 2N_1 + 3N_2 + 7N_3,$$

where N_1, N_2 and N_3 are three 8-bit numbers available from three consecutive memory locations. Store the sum S in another memory location. Perform the experiment for three different sets of N_1, N_2 and N_3.

12. Write an assembly language program to add two 3-byte binary numbers N_1 and N_2. The least significant byte of N_1 and N_2 are loaded first and the most significant byte of the two numbers are loaded last as shown in the figure. Store the sum also in three consecutive memory locations. If final carry is generated due to this addition then that carry should also be loaded in the memory. Repeat this using at least 2 sets of N_1 and N_2.



13. Write an assembly language program to multiply two 8-bit numbers available from two consecutive memory locations. Use left-shift and Add method. Store the product in suitable memory location. Also find $X = N_1 * N_2 - N_3$, where N_1, N_2 and N_3 are all 8-bit numbers. The number N_3 may be available from a suitable memory location. Store the value of X into the location just after location where the result of product $N_1 * N_2$ is stored. Repeat the experiment using two sets of N_1, N_2 and N_3.

14. Write an assembly language program to convert on 8-bit binary to the Gray code. The 8-bit binary number is available from a memory location NUMB and the Gray code is to stored in the location GRAY. Repeat this experiment using 5 different 8-bit binary numbers.