İzmir Katip Çelebi University

Department of Electrical and Electronics Engineering

EEE316 Microprocessors

Spring 2020

Experiment IV

Memory Operations

Pre-Lab Report

- Please study related topics in reference notes.
- Answer the questions under the lab activities. Prepare report in the specified format. Reports must be completed before coming to lab.
- Submit your report to CANVAS until March 18, 23:59.

Experimental Work

- Please explain your code step by step to instructors during lab hours.

Lab Objectives

- Memory operations using a MPLAB simulator.
- Constructing loops

References

- Lecture notes
- Mazidi, McKinlay, Causey "PIC Microcontroller and Embedded Systems," Chapter 5 and Chapter 6

Lab Activities

Reminder : To let the user enter data in the MPLAB simulator, set a breakpoint to the next line after "ORG 0H". So when you run the program, it will stop at the beginning of the program. Then you can go to "File Registers" and enter the data to any address. After that, you can run your program step by step, or click "continue" (press F5) to see the output. **All three tasks need to be implemented with memory operations, otherwise you will not receive any credits.**

1. Write a program that combines two memory regions into one memory region with writing nth elements of each memory region to the new memory region consecutively. The first and second lists are started from the address 0x20 and 0x30, respectively. The lengths of lists are the same and stored in the address 0x10 as one byte. The start address of the result should be the same as the address of the first list. In other words, the result list should be written on the first list (Therefore you can use a temporary list for a combination of two lists, then move it to the first list's location). Next, the program should find the average of

the new list (combination of two lists) and store the result to the 50H location (You should use the pointer algorithm).
Note : Numbers and length of lists (n) should be entered as hex format via MPLAB File Register Section. The length of the lists (n) should be entered to 0x10 location.

Example : M[.] indicates the content of address. Old value means the value before the execution of the program.
After execution of the program,
New value of M[0x20] = Old value of M[0x20] (Write the first element of the first list)
New value of M[0x21] = Old value of M[0x30] (Write the first element of the second list)
New value of M[0x22] = Old value of M[0x21] (Write the second element of the first list)
New value of M[0x23] = Old value of M[0x31] (Write the second element of the second list)
...
Let's assume that the numbers (hex) are 1, 3, 5, 7 and 9 (n=5) are stored in the first list which starts from address 0x20 to 0x24. The second list contains (hex) 2, 4, 6, 8 and A (n=5) are stored in the address 0x30 to 0x34. After the execution of the program, the numbers need to be displayed as (hex) 1, 2, 3, 4, 5, 6, 7, 8, 9, A in the address from 0x20 to 0x29. Then, average of the numbers (hex) (1+2+3+...+A)/2n = 5H. Quotient (5H) should be assigned to the 50H location.

Note : These numbers are given just for demonstration. You can use the above numbers to test your code. However, the length and the numbers will be defined by the user during lab hours.

2. Write a program that adds two lists of 16-bit unsigned numbers "one by one" and put the 24-bit sums into result list starting at $0x50. The first list starts at 0x10 and the second one starts at $0x30. The number of 16 bit numbers in each list, in other words length of the lists, is same and kept in location 0x00 as 1 byte. The length and the numbers will be defined by the **user**.

M[0x10, 0x11]+ M[0x30, 0x31] ➔ M[0x50, 0x51,0x52]
M[0x12, 0x13]+ M[0x32, 0x33] ➔ M[0x53, 0x54,0x55]
M[0x14, 0x15]+ M[0x34, 0x35] ➔ M[0x56, 0x57,0x58]
…

3. Write a program that rewrites a given string's characters in inverse order. The list starts at 0x10. That string must be ended with "/0" character. The inverse ordered string must be written to the address of original string. The string will be defined by the **user**.

Example: M[0x10, …,0x15]= "MICRO/0"
After execution of the program:
M[0x10, …,0x15]= "ORCIM/0"

4. **(BONUS)** Write a program that saves your name, surname, position, department, university and your initials (First letter of name and surname) and your logo to the ASCII part in MPLAB's File Register section.

- Your name should be saved 000 Address section's ASCII part.
- Your surname should be saved 010 Address section's ASCII part.
- Your position should be saved 020 Address section's ASCII part.
- Your department should be saved 030 Address section's ASCII part.
- Your university should be saved 040 Address section's ASCII part.
- Your initials should be saved starting from 050 Address section's ASCII part.
- Your logo should be saved starting from 070 Address section's ASCII part.

Note : If they don't fit in the required area, you should use abbreviations. Logo can be movies, tv series, university's logo etc.

Name, surname, position, department, university and your initials should be specified as 'DB directive' in your program (Look up Table). You should send it (each) your related address. Your information is burned into ROM locations starting at 400H as 'DB directive' and that program is burned into ROM locations starting at 05H.

Example :