EE323: Microprocessor Systems Design
Lab. N° 2

Arithmetic and Branch Operations

1 Objective

The objective of this lab is to learn how to use Z80 arithmetic and branch instructions to write assembly program. You will also learn how to implement loops and conditional statements using conditional jump instructions.

2 Before the Lab

1. Download from EE321course website, "Z80 User Manual" and read the section about DAA instruction.
2. Write a program to add two decimal values (BCD representation) stored in memory locations 2500H and 2501H. The result (BCD value) should be stored at address 2502H. Test your program using Z80 simulator.
3. Explain the function implemented by the program of task1. Test the program using Z80 simulator. Give the obtained results.
4. Review the function of CP instruction. Write your program for the task 2 and test it using Z80 simulator.
5. Execute manually the pseudocode of task 3 using a table of 5 values: [1, 4, 5, 2, 3].
6. Give the pseudocode of your program for the task 4 and test it using Z80 simulator.

3 Lab grading

This lab is scheduled for **1 week (3 hours)** and will be graded based on:

1) Preparation.
2) Completion of the experiment, obtained results and their interpretation. During the lab, you must show the obtained results to your teacher. Before moving to a new lab experiment (task), you have to demonstrate that your implementation is working correctly, and you must show your understanding of the experiment. To evaluate your comprehension; you may have to answer oral questions.
3) Answers to the questions.
4) Attendance to the lab.
5) Report sheet.

For this lab, a report sheet will be used as lab report. You must put all your experiment results and the answers to the questions on this sheet and **return it back at the end of the lab**. When you finish a task and before moving to the next task, you should get your teacher's signature for the results of the finished task. To evaluate your comprehension, during the demonstration, you may have to answer other questions not included in the lab assignment.

4 Tasks

4.1 Task 1: BCD operations and DAA instruction

The objective of this task is to get familiar with Binary Coding Decimal (BCD) operations and Digital Adjust Accumulator (DAA) instruction. DAA is used to adjust the result of Add/Sub operations. The result of DAA instruction depends on H, N, and C flags.

1) Explain the function implemented by the assembly program of Figure 1.Which results we should obtain using 00 as initial value?
2) Load the program into the memory starting at address 2000H and execute it. Which memory locations are modified by this program? Which memory location is used for the initial value?
3) Test your program with 00 and 09 as initial value. Give the obtained results. Using step by step execution, explain why the obtained results are different from the expected ones.
4) Correct the given program and test it again with different initial values. Give the obtained results. Explain how you corrected the program.
5) ***Get your work checked by your teacher.***

```
      .ORG 2000H
      LD HL, 2500H     ; HL is used as pointer
      LD B,10          ; B is used as counter
      LD A, (HL)       ; Load initial BCD value
LOOP: INC A            ; Increment current value
      DAA              ; Convert result to BCD
      INC HL           ; Update HL
      LD (HL), A       ; Save result
      DJNZ LOOP        ; Repeat loop until B=0
      HALT             ; Program end
```

Figure 1: Assembly program using DAA.

4.2 Task 2: Jump operations

The objective of this task is to get familiar with jump and compare operations. Jump instructions are used to implement loops and conditional statements.

1) Write a program to find the position (index from starting address) of the highest unsigned number stored in 10 continuous memory locations starting from 2500H. The result should be stored at address 2600H.
2) Assemble and test your program using Z80 simulator.
3) Load the program into the memory starting at address 2200H. Test your program using the following values: ***56H, 2AH, D3H, 07H, B9H, 37H, 1FH, C1H, 9FH***, ***55H***. Give the obtained result.
4) ***Get your work checked by your teacher.***
5) Put your assembly program on the report sheet. What is the size of your program? What is the number of cycles needed to execute your program?
6) Explain how compare (CP) instruction function.
7) Explain how you can modify your program to use it with 2's complement signed numbers

4.3 Task 3: Sorting algorithm

The objective of this task is to learn how to sort (order) unsigned numbers using Bubble Sort algorithm. The pseudocode of Bubble Sort algorithm is given in Figure 2.

1) Using the given pseudocode, write assembly program to sort in ascending order the unsigned numbers stored in 10 continuous memory locations starting from 2500H.
2) Assemble and test your program using Z80 simulator. Give the number of cycles of your program when the elements are in reverse order and when they are already ordered.
3) Load the program into the memory starting at address 2300H. Test your program using the following values: ***56H, 2AH, D3H, 07H, B9H, 37H, 1FH, C1H, 9FH***, ***55H***.
4) ***Get your work checked by your teacher.***
5) Put your assembly program on the report sheet. Is-it possible to optimize the program to reduce the number of cycles when the elements are already ordered?

```
Initialize PT to starting memory location
Initialize NB to the number of elements to sort
FOR i=0 to NB-1
     FOR j=0 to NB-i-1
          A = content of memory location (PT+j)
          B = content of memory location (PT+j+1)
          IF A > B THEN
               Copy A to memory location (PT+j+1)
               Copy B to memory location (PT+j)
```

Figure 2: Pseudocode for Bubble Sort algorithm.

4.4 Task 4: 8-bit multiplication

Z80 does not have built-in multiplication instructions, when a programmer wants to do multiplication; he has to use a program routine. The objective of this task is to learn how to multiply two unsigned 8-bit numbers.

1) Write a program to multiply two 8-bit unsigned numbers stored in memory locations 3000H and 3001H. The low byte of the result should be stored at address 3002H and the high byte at address 3003H.
2) Assemble and test your program using Z80 simulator.
3) Load the program into the memory starting at address 2400H. Test your program using the following values: a) ***08H*** and ***10H***, b) ***FFH*** and ***FFH***, and c) ***00H*** and ***11H***. Give the obtained result.
4) ***Get your work checked by your teacher.***
5) Explain how you can modify your program to multiply two 8-bit signed numbers.