Abstract

In this experiment some advanced instructions of 8086 microprocessor are introduced. Basic use of array, loop, div were demonstrated to get summation of a serries and average of some random numbers.

1 Introduction

This experiment was based on the advanced instructions of an intel 8086 micro processor. The main objectives are as follows:

- To work with advanced 8086 instructions.
- To learn how to write assemble programs using 8086 instructions and arrays.

2 Pre-Lab Homework

The codes were emulated before the lab experiment.

3 Apparatus

1. Microprocessor 8086 Trainer Board (MTS-86c)
2. Emu8086 [ver.408 (32 bit WINOS compatible)]
3. PC having Intel Microprocessor

4 Precautions

1. The data in MTS-86c with care.
2. In case of the 16 bit numbers the last 8 bits should be entered first.

5 Calculation

- Series Summation
  - BX = 0
  - CX = 9
  - BX = BX + CX = 0 + 9 = 9
  - BX = BX + CX = 9 + 8 = 17
  - BX = BX + CX = 17 + 7 = 24
  - ... ...

  - ... ...
  - BX = BX + CX = 44 + 1 = 45

- Average from Array
  - AX = 0
  - AX = CX = Code Segment address
  - DX = AX = Code Segment address
  - AX = 0
  - SI = Offset address of A
  - CX = 3
  - BL = 3
  - AL = AL + [SI] = 0 + 3 = 3
  - SI = SI + 1
  - CX = CX - 1 = 2
  - AL = AL + [SI] = 3 + 1 = 4
  - SI = SI + 1
  - CX = CX - 1 = 1
  - AL = AL + [SI] = 4 + 2 = 6
  - SI = SI + 1
  - CX = CX - 1 = 0
  - AL = $\frac{AL}{Bl} = \frac{6}{3} = 2$

6 Result

The calculated, simulated and implemented values of all the registers matched with each other.

7 Answer to Report Question

1. Write an assembly code to solve the problem: BX = 5!

```
FAC SEGMENT
    ASSUME CS:FAC, DS: FAC

    MOV AX, 1
    MOV CX, 5

    L1:
```

```
        MUL CL
    LOOP L1

    MOV BX, AX

    HLT
    FAC ENDS
END
```

2. Write an assemble code to solve the following problem (without using array):
   DX = 2+4+6+ ..... + 20

```
SERIES SEGMENT
    ASSUME CS:SERIES, DS: SERIES

    MOV AX, 2D
    XOR DX, DX

    L1:
        ADD DX, AX
        ADD AX, 2
        CMP AX, 22
    JNE L1
    HLT
    SERIES ENDS
END
```

3. Write an assemble code to solve the following problem (with using array):
   AX = 15+19+7+20+2

```
ARRAY SEGMENT
    ASSUME CS:ARRAY, DS: ARRAY

    XOR AX, AX
    MOV AX, CS
    MOV DS, AX
    MOV AX, 0

    MOV SI, OFFSET A
    MOV CX, 5

    L1:
        ADD AL, [SI]
        INC SI
    LOOP L1
    HLT
```

```
    A DB 15,19,7,20,2
    ARRAY ENDS
END
```

8 Discussion

- The values were checked after inserting them in the training kit
- One or two mistakes were corrected by re-entering the value on the training kit.

9 Conclusions

After completing process the value of the register matched with the expected value for those registers. So, the experiment was successful.

References

[1] K & h mfg. co., ltd.-8086 microcomputer trainer - for educational teaching, training, and learning of 8086 chip architecure and programming, 8255, 8259, 8253, 8251, 8279, adc0809, dac0808, ram 62256, rom 27256. http://www.taiwantrade.com.tw/EP/kandh/products-detail/en_US/609531/8086_MICROCOMPUTER_TRAINER_-_for_educational_teaching,_training,_and__learning_of_8086_chip_architecure_and_programming,_8255,_8259,_8253,_8251,_8279,_ADC0809,_DAC0808,_RAM_62256,_ROM_27256/. (Visited on 06/02/2015).

[2] A.P. Malvino and J.A. Brown. *Digital Computer Electronics*. McGraw-Hill electricity & electronics series. Glencoe, 1992.

10 Appendices

10.1 Series Summation

```
1  0000                          SERIES SEGMENT
2                                    ASSUME CS:SERIES, DS: SERIES
3  0000   33 DB                     XOR BX, BX
4  0002   B9 0009                   MOV CX, 9
5  0005                             START:
6  0005   03 D9                         ADD BX, CX
7  0007   E2 FC                     LOOP START
```

```
 8
 9 0009   F4                    HLT
10 000A                         SERIES ENDS
11                          END
```

10.2 Average from Array

```
 1 0000                     ARRAY SEGMENT
 2                              ASSUME CS:ARRAY, DS: ARRAY
 3
 4 0000   33 C0                 XOR AX, AX
 5 0002   8C C8                 MOV AX, CS
 6 0004   8E D8                 MOV DS, AX
 7 0006   B8 0000               MOV AX, 0
 8
 9 0009   BE 0019 R             MOV SI , OFFSET A
10 000C   B9 0003               MOV CX, 3
11 000F   B3 03                 MOV BL, 3
12
13 0011                         L1:
14 0011   02 04                      ADD AL, [SI]
15 0013   46                        INC SI
16 0014   E2 FB                  LOOP L1
17 0016   F6 F3                  DIV BL
18 0018   F4                    HLT
19
20 0019   03 01 02              A DB 3,1,2
21 001C                         ARRAY ENDS
22                          END
```