Abstract

In the lab experiment the intel 8086 microprocessor were introduced with some basic programs. Some basic commands such as ADD, SUB, XCHG etch were used on the four codes done in the lab.

1 Introduction

The main purpose of this experiment was to introduce us with the architecture, operation, working procedure etc of Intel 8086 microprocessor. Here we were introduced with the basic architecture and working procedure of Intel 8086 microprocessor. We also simulated a few simple programs in EMU8086 emulator and executed them in a MTS-86c trainer board.

2 Pre-Lab Homework

- The basic architecture of 8086 was studied.

3 Apparatus

- Microprocessor 8086 Trainer Board (MTS-86c)
- EMU8086
- PC having Intel Microprocessor

4 Precautions

- Input of MTS-86c should be entered carefully
- Pen drives should not be used in the PC.

5 Calculation

- Exchange Program
  - AX = 1234H
  - BX = 5678H
  - CX = AX = 1234H
  - AX = BX = 5678H
  - BX = CX = 1234H
- Addition Program

- BX = 1234H
- CX = 5678H
- BX = BX + CX = 1234H + 5678H = 068AH
- AL = 13H
- DL = 01H
- AL = AL + DL = 13H + 01H = 14H

- Subtraction Program

  - BX = 1234H
  - CX = 5678H
  - BX = BX - CX = 1234H - 5678H = BBBC
  - AL = 13H
  - DH = 014H
  - AL = AL - DH = 13H - 01H = 11H

6 Result

All the register value in the training kit matched with the calculated and simulated value of the program. So, the experiment was successful.

7 Answer to Report Question

1. Write an assembl program to complete the following task.
   CX = 15+AX-17+BX ; AX = 10, BX = 5

```
CODE SEGMENT
        ASSUME CS:CODE, DS:CODE

    MOV AX, 10
    MOV BX, 5

    MOV CX, 15
    ADD CX, AX
    SUB CX, 17
    ADD CX, BX

    HLT
    CODE ENDS
END
```

2. Write an assembl program to complete the following task.
   BX = 5+2-3+4

```
CODE SEGMENT
        ASSUME CS:CODE, DS:CODE
    MOV BX, 5
    ADD BX, 2
    SUB BX, 3
    ADD BX, 4

    HLT
    CODE ENDS
END
```

8 Discussion

- All the codes were verified on the simulator before running them on the training kit.
- Some keys on the training kit was not clear, so we were careful wile inputing the opt code.
- In case of inputing the value of General Purpose Registers the last 4bits was entered first, followed by the first 4bits.
- All the register value in the training kit was similar to that of the emulator and as expected.
- In one of the trainer kit we were using the "," key was problematic, which caused a little trouble at once stage of the experiment.

9 Conclusions

After inputing the opt code in the training kit we checked the register values. All the register value matched the calculated value of those registers. Whic proves that the experiment was successful.

References

[1] Download emu8086 4.08. `http://8086-microprocessor-emulator.soft32.com/`. (Visited on 06/02/2015).

[2] Instruments techno test inc. `http://catalogue.techno-test.com/products/6-Educational_Products/38-Electricity___Electronics/354-K_H_MTS_86C-MTS_86C_8086_Microcomputer_Trainer.html`. (Visited on 06/02/2015).

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

10 Appendices

10.1 Exchange Programs

```
 1  0000                          CODE SEGMENT
 2                                    ASSUME CS:CODE, DS:CODE
 3
 4  0000   B8 1234                   MOV AX, 1234H
 5  0003   BB 5678                   MOV BX, 5678H
 6
 7  0006   8B C8                     MOV CX, AX
 8  0008   8B C3                     MOV AX, BX
 9  000A   8B D9                     MOV BX, CX
10
11  000C   F4                        HLT
12  000D                             CODE ENDS
13                                END
```

```
 1  0000                          CODE SEGMENT
 2                                    ASSUME CS:CODE, DS:CODE
 3  0000   BB 1234                   MOV BX, 1234H
 4  0003   B9 5678                   MOV CX, 5678H
 5  0006   87 D9                     XCHG BX, CX
 6  0008   F4                        HLT
 7  0009                             CODE ENDS
 8                                END
```

10.2 Addition Program

```
 1  0000                          CODE SEGMENT
 2                                    ASSUME CS:CODE, DS:CODE
 3  0000   BB 1234                   MOV BX, 1234H
 4  0003   B9 5678                   MOV CX, 5678H
 5  0006   03 D9                     ADD BX, CX
 6  0008   B0 13                     MOV AL, 13H
 7  000A   B2 01                     MOV DL, 01H
 8  000C   02 C2                     ADD AL, DL
```

```
 9  000E   F4                HLT
10  000F                     CODE ENDS
11                       END
```

10.3 Subtraction Program

```
 1  0000                     CODE SEGMENT
 2                               ASSUME CS:CODE, DS:CODE
 3
 4  0000   BB 1234           MOV BX, 1234H
 5  0003   B9 5678           MOV CX, 5678H
 6
 7  0006   2B D9             SUB BX, CX
 8  0008   B0 1B             MOV AL, 1BH
 9  000A   B6 02             MOV DH, 02H
10  000C   2A C6             SUB AL, DH
11
12  000E   F4                HLT
13  000F                     CODE ENDS
14                       END
```