 Daffodil International University (DIU)

Department of Electrical and Electronic Engineering

EEE 324: Microprocessors and Interfacing Laboratory

Experiment No: 08

**Name of the Experiment:   
Interfacing 8086 with serial monitor and writing logical instructions using 8086 and serial monitor.**

**Objective:**

The objectives of the experiment are-

* To interface Microprocessor kit 8086 with serial monitor.
* To be familiar with logical instructions.
* To verify the operation of AND, OR, XOR, NOT logical instructions.

**Apparatus:**

1. PC with Windows XP.
2. Microprocessor kit 8086.
3. MASM.EXE
4. LOD186.EXE
5. LINK.EXE
6. Wincomm software.

**Theory:**

## Serial monitor

Serial monitor is the basic monitor program which monitor program running in 8086 through communication port link between MDE-8086 and computer.

## Connecting the computer

The connector of computer **RS-232C** is 25 pin and RS-232c of **MDE-8086** is 9 pin is connected.

**MDE-8086** & **RS-232C** port of computer is also connected. So **RS-232C** is ready to do data communication with IBM comparator personal computer.

## Using serial monitor

To use serial monitor, jumper P1 is moved which located on the PCB like this:



p1

After moving the JP1 to serial monitor, if on a power up or pressing RES key, following message will be displayed on LCD and data communication is possible with computer.

Serial Monitor! !

Midas 335-0964/5

Midas 335-0964/5

At the same time the computer monitor shows the following display:

\*\*8086 Monitor 9.5 \*\*

\*\*Midas 335-0964/5\*\*

8086>

## Serial Monitor commands

To show the **HELP COMMAND** menu, we have to type a question mark (?) and press ENTER.

8086 >?

HELP COMMAND

E segment : offset ………………………...: Enter Data to Memory

D segment: offset length ………………….: Dump Memory Contents

R [Register name] ……………………….. : Register Display & Change

M address1, length, address2………………: Move Memory From 1 to 2

F address, length, data…………………….: Fill Memory With Any Data

L Return key………………………………: Program Down Load

G segment : offset…………………………: Execute program

T………………………………………………… : Program 1 step execute

**Memory Modify Command:**

To modify memory we use the command E segment:offset which tells us to enter data to memory. For example:

Type New Data and then Enter

Current Data

Address

Command “E segment:offset” then Enter

8086>E 0000: 3000

0000:3000 DD? 65

0000:3001 DD? 45

0000:3002 DD?AC

0000:3003 DD? 77

0000:3004 DD? 46

0000:3005 DD? /

0000:3004 46? /

0000:3003 77? **.**

Here “**/**” is used for offset Decrement and “**.**” is used for Escaping command.

**Memory Display Command:**

To display memory we use the command D segment**:**offset length which dumps memory contents. For example:

8086>D 0000:3000

0000:3000 65 45 AC 77 46 FF FF FF - FF FF FF FF FF FF FF FF eE-wF………..

0000:3010 FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF ……………….

0000:3020 FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF ……………….

**Block Move Command :**

The M command is used to move blocks of memory from one area to another.

Syntax for Move command is **“M segment:offset ‘no of bytes’ segment:offset”**

(source) (destination)

8086>M 0000:3000 0A 0000:3030

This will move copy data of 03000 to 03030

03001 to 03031

. . . . . to . . . . .

So on total 0A bytes

**Memory Fill Command** :

F command is used to fill memory with any data.

Syntax for Move command is **“F segment:offset ‘no of bytes’ Data”**

8086>F 0000:3000 10 00

8086>D 3000

0000:3000 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00 …..………..

0000:3010 FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF ……………….

0000:3020 FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF ……………….

0000:3030 65 45 AC 77 46 FF FF FF -FF FF FF FF FF FF FF FF eE-

The command “F 0000:3000 10 00” fills memory addresses from 0000:3000 to 0000:300F total 10H bytes all with 00H.

**Display Registers command :**

The R command is used to display the 8086 processor registers .

8086>R

AX=0000 BX=0000 CX=0000 DX=0000

SP=0000 BP=0000 SI=0000 DI= 0000

DS=0000 ES=0000 SS=0000 CS=0000

IP=1000 FL=0000 =………

To change register individually following are does:

8086>R CX 

CX=0000 1234

DX=0000 1358

SP=0540 

**PROCEDURE:**

1. In a text editor (Notepad) write the following code.

**Program 01:**

CODE SEGMENT

ASSUME CS: CODE

ORG 1000H

MOV AX, 3256H

MOV BX, 3465H

MOV CX, 5639H

MOV DX, 6789H

AND AX, 0CF3H

OR DX, 8ADCH

XOR BX, CX

NOT CX

HLT

CODE ENDS

END

1. Save the program as an asm file. i.e. file name should be like EXP5.asm
2. Save the program in a folder where MASM.EXE and LINK.EXE and LOD186 is present.
3. Write ‘cmd’ in RUN window and press ENTER.
4. To change drive write C: or D: or E: and ENTER to go any drive you want.
5. To change current directories, you will use the cd command. The cd command stands for “change directory”.
6. Run the file MASM.EXE with your asm file name after it.

MASM.EXE EXP5.asm or

MASM EXP5.asm

1. Press ENTER to create EXP5.OBJ file then you will see

Source listing [NUL.LST]:

Press ENTER if list file is not needed.

Press any name if list file is needed.

1. Then you will see

Cross-reference [NUL.CRF]:

Press ENTER if cross reference file is not needed.

Press any name if cross reference file is needed.

1. If there is any error or warning you will see what error happened in which line.
2. Fix error in the source file and run MASM again if there is any error.
3. Now for loading use lod186 command and press ENTER.
4. Then you will see

Object/ Command File [.OBJ]:

Press ENTER if object file is not needed.

Press any name if object file is needed.

1. Then you will see

Output Object File [.ABS]:

Press ENTER if abs file is not needed.

Press any name if abs file is needed.

1. Then you will see

Map Filename [NUL.MAP]

Press ENTER if map file is not needed.

Press any name if map file is needed.

1. To use serial monitor, set the jumper P1 in the Microprocessor kit 8086 and press RES button to reset.
2. Now open WINCOMM software and press L↵ to download.
3. Then press F3 and select abs file type.
4. Now open the created abs file.
5. Then press G↵ to interface with serial monitor.
6. Type R↵ and write down register values.
7. Type T↵ once. This will execute one instruction and show register status. Write down again which registers has been changed.
8. Continue until all instructions are executed
9. Now collect the values of register and verify them.