

**Laboratory Manual**  
**of**  
**MICROPROCESSOR**  
**AND**  
**INTERFACING LAB**  
**(B24-ECE– 214)**



**DEPARTMENT OF**  
**ELECTRONICS & COMMUNICATION**  
**ENGINEERING**  
**STATE INSTITUTE OF ENGINEERING &**  
**TECHNOLOGY**  
**NILOKHERI 132117**

B24-ECE-214	Microprocessor & Interfacing Lab						
	Lecture	Tutorial	Practical	Credit	Internal Assessment	Practical exam	Total
0	0	3	1.5	40	60	100	3 Hour
Purpose	Write the efficient Assembly Language Program for different problem statements and implement different system interfacing.						
Course Outcomes							
CO 1	Understanding different steps to develop program such as Problem definition, Analysis, Design of logic, Coding, Testing, Maintenance (Modifications, error corrections, making changes etc.)						
CO 2	To be able to apply different logics to solve given problem.						
CO 3	To be able to write program using different implementations for the same problem						
CO 4	Use of programming language constructs in program implementation						

**LIST OF EXPERIMENTS: (Verification of atleast 3 experiments may also be done using TASM)**

- I. Familiarization with 8086 Trainer Kit
  - b) Familiarization with Digital I/O, ADC and DAC Cards
  - c) Familiarization with Turbo Assembler and Debugger S/Ws.
- II. Write a program to arrange block of data in
  - (i) ascending and (ii) descending order.
- III. Write a program to find out any power of a number such that  $Z = X^N$ . Where N is programmable and X is unsigned number.
- IV. Write a program to generate.
  - (i) Sine Waveform (ii) Ramp Waveform (iii) Triangular Waveform Using DAC Card.
- V. Write a program to measure frequency/Time period of the following functions.
  - (i) Sine Waveform (ii) Square Waveform (iii) Triangular Waveform using ADC Card
- VI. Write a program to increase, decrease the speed of a stepper motor and reverse its direction of rotation using stepper motor controller card.
- VII. Write a programmable delay routine to cause a minimum delay = 2MS and a maximum delay = 20 minutes in the increments of 2 MS
- VIII. Write a program that takes any two numbers as Input from the user through the input device (Keyboard) & Prints their sum on the standard output device (Screen).
- IX. Write a program that takes any two numbers as Input from the user through the input device (Keyboard) & Prints their sum on the standard output device (Screen) by giving appropriate messages to the user
- X. Write a program that initializes 100 positions in an array and loads them with 0.
- XI. Write a program that prints a Blinking character in the middle of the screen.
- XII. Write a program that accepts a number from the user through the input device (Keyboard), calculates its factorial and prints the result on the screen.
- XIII. ON/OFF control of SSR (Solid State Relay) using interface with 8255
- XIV. Interfacing of LM35/RTD temperature sensor with 8086 and display the temp value on LCD
- XV. To interface traffic light system using 8086 & 8255.

# INDEX

S.NO	EXPERINEMT NAME	DATE	REMARKS
1.	a. Familiarization with VMC 8609 Trainer Kit b. Familiarization with Turbo Assembler and Debugger software		
2.	Write a program for Addition and Subtraction of 16 bit number with using 8086 microprocessor		
3.	Write a program for multiplication and division of number by using 8086 microprocessor.		
4.	Write a program for logical operation like logical AND, logical OR, logical NOT, logical XOR, logical XNOR.		
5.	Write a program to find the largest number and Smallest no given string of data using 8086 microprocessor		
6.	TO find factorial of given number using 8086 microprocessor.		
7.	TO FIND 1'S AND 2'S COMPLIMENT USING 8086 MICROPROCESSOR.		
8.	Write a program to arrange a block of data in ascending or descending order using 8086 microprocessor		
9.	WRITE A PROGRAM GCD OF TWO 8 BIT NUMBERS IN TASM SOFTWARE.		
10.	PROGRAM TO CONVERT 8-BIT HEXADECIMAL TO BCD		
11.	Program to transfer Block of Data using string instruction		
12.	Assembly language program to exchange two data memory blocks of 10 bytes, which are store in 5000H:1000H and 5000H:2000H.		

# EXPERIMENT NO. 1A

**Aim:-** Familiarization with VMC 8609 Trainer Kit

To study and understand the hardware architecture, functional blocks, and basic operation of the VMC 8609 Microprocessor Trainer Kit, and to learn how to enter, execute, and observe a program on the kit.

## Apparatus Required

1. VMC 8609 Trainer Kit
2. Regulated Power Supply (230 V AC)
3. Connecting cables (if required)
4. User manual of VMC 8609 Trainer Kit

## Theory

The VMC 8609 Trainer Kit is an educational microprocessor-based training system designed to understand the architecture and programming of the 8086 microprocessor.

It provides facilities for:

- Program development
- Memory inspection
- Register analysis
- Execution of machine-level programs

The kit integrates the 8086 CPU, memory units, input/output interfaces, and a hexadecimal keyboard with LED display, making it suitable for learning microprocessor concepts practically.

## The trainer kit allows users to:

- Enter machine language programs
- Modify memory locations
- Execute programs step-by-step or continuously
- Observe outputs using LEDs and displays

## **Main Blocks of a Standard VMC 8609 Trainer Kit**

### **1. 8086 Microprocessor**

- Acts as the central processing unit
- Executes instructions and controls all operations

### **2. Memory Unit**

- EPROM / Flash ROM: Stores monitor program
- RAM: Used for user program storage and data

### **3. Keyboard Section**

- Hexadecimal keypad (0–9, A–F)
- Used for:
  - Data entry
  - Address entry
  - Command execution

### **4. Display Section**

- 7-segment LED display
- Displays:
  - Addresses
  - Data
  - Register contents
  - Program outputs

### **5. Input / Output Interface**

- Supports interfacing with external devices
- LEDs, switches, and ports for input/output experiments

### **6. System Monitor**

- A built-in program stored in ROM
- Provides commands like:
  - Memory examine
  - Modify memory
  - Run program

- Single step execution

## **7. Power Supply Unit**

- Converts AC input to regulated DC supply
- Supplies required voltages to all components

### **Procedure**

1. Connect the VMC 8609 Trainer Kit to the power supply.
2. Switch ON the power and observe the display.
3. Ensure the monitor program is loaded automatically.
4. Identify the keypad, display, and main functional blocks.
5. Use the monitor commands to:
  - Examine memory
  - Enter data
  - Modify memory locations
6. Load a sample machine-level program.
7. Execute the program using RUN or SINGLE STEP mode.
8. Observe the output on the display or LEDs.

### **Steps to Follow for Programming on VMC 8609 Trainer Kit**

#### **Step 1: Reset the Kit**

- Press RESET key
- Monitor prompt appears on the display

#### **Step 2: Enter Starting Address**

- Use hexadecimal keys to enter the desired memory address
- Press ENTER

#### **Step 3: Enter Program Data**

- Enter machine code instruction byte-by-byte
- Press NEXT / ENTER after each byte

#### **Step 4: Verify Program**

- Use MEMORY EXAMINE command
- Check entered data at each address

#### **Step 5: Execute the Program**

- Press RUN to execute the entire program  
OR
- Press SINGLE STEP to execute instruction by instruction

#### **Step 6: Observe Output**

- Check output on:
  - Display
  - LEDs
  - Registers (if required)

#### **STEP TO FOLLOWING PROGRAM ON 8086 KITS**

- Press ENTER and then press A for Assemble mode.
- ENTER address in the form of base add. off set add CS:IP at which program into be stored.
- ENTER program at location 0400 then press ENTER.
- Press F7 to save the program.
- To ENTER data press D. enter the address at which data is to be saved
- Press F7 to save the data
- For execution of program press G
- Enter to add. At which program is saved.
- Press RESET & then press ENTER.
- Press D to find the results

## **Observations**

- The trainer kit successfully accepts program instructions.
- Memory locations can be modified and verified.
- Program execution can be controlled using RUN and SINGLE STEP modes.
- Output is correctly displayed on the LED display.
- Registers and memory contents change according to the program logic.

## **Result**

Thus, the VMC 8609 Trainer Kit was successfully studied, and the basic operations, functional blocks, and programming procedure were understood. The program was entered, executed, and verified correctly on the trainer kit.

# EXPERIMENT NO. 1B

**Aim:** Familiarization with Turbo Assembler and Debugger software

To study and understand the use of Turbo Assembler (TASM) and Debugger (Debug.exe) software for writing, assembling, debugging, and executing 8086 assembly language programs.

## Apparatus Required:

- Computer system
- Turbo Assembler (TASM)
- Turbo Linker (TLINK)
- MS-DOS or DOSBox environment
- Debugger software (DEBUG.EXE)

## Theory:

Turbo Assembler (TASM) is an assembler used to convert 8086 assembly language programs into machine code. Debugger software such as DEBUG.EXE allows users to load, test, and analyze programs by examining registers, memory, and flags.

1. Turbo Assembler (TASM): TASM converts a .ASM file into an object file (.OBJ) and then an executable (.EXE) using TLINK.

## Typical steps:

1. Write the assembly program using any text editor and save as PROGRAM.ASM.
2. Assemble using TASM:
  3. TASM PROGRAM.ASM
4. Link using TLINK:
  5. TLINK PROGRAM.OBJ
6. Run the executable:
  7. PROGRAM.EXE

## 2. Turbo Debugger (DEBUG.EXE):

DEBUG allows step-by-step execution and provides access to registers, memory, and machine instructions.

Common Debug commands:

Command	Meaning
R	View/modify registers
D	Dump memory contents
E	Edit memory
U	Unassemble machine code
T	Trace (execute one instruction)
P	Proceed (execute procedure)
G	Go (run program)
Q	Quit

**Procedure:**

**Using Turbo Assembler (TASM):**

1. Create an .ASM file using a text editor.
2. Open DOS/DOSBox environment.
3. Type TASM FILENAME.ASM to assemble the program.
4. Check for error messages during assembly.
5. After successful assembly, type TLINK FILENAME.OBJ to create an executable file.
6. Run the program using FILENAME.EXE.
7. Observe the output.

**Using DEBUG.EXE:**

1. Open DOS/DOSBox environment.
2. Load the program using command:
3. DEBUG FILENAME.EXE
4. Use R to view registers.
5. Use U to view unassembled code.
6. Use T or P to execute instructions step by step.
7. Observe changes in registers and flags.
8. Exit the debugger using Q.

**Observations:**

- TASM successfully assembled .ASM files into object code.
- TLINK generated executable .EXE files.
- Debug allowed examination of internal CPU registers, memory contents, and instruction execution.
- Stepwise execution demonstrated how each instruction affects flags and registers.

**Result:**

Familiarization with Turbo Assembler and Debugger was successfully completed. The process of editing, assembling, linking, executing, and debugging 8086 assembly programs was understood.

## EXPERIMENT NO. 2

**Aim:** Write a program for Addition and Subtraction of 16 bit number with using 8086 microprocessor

**Apparatus Required:** 8086 microprocessor trainer kit, keyboard, power cord

**Program:**

**Addition:**

Memory Address	Mnemonics	Operands
0400	MOV	AX,0000
0403	MOV	DS,AX
0405	MOV	SI,0500
0408	MOV	CX,0000
040B	MOV	AX,[SI]
040D	MOV	SI,0502
0410	ADD	AX,[SI]
0412	JNC	0415
0414	INC	CX
0415	MOV	SI,0504
0418	MOV	[SI],AX
041A	MOV	AX,CX
041C	MOV	SI,0506
041F	MOV	[SI],AX
0421	HLT	

**Result:**

**Addition:**

**I/P Memory Address                  Data**

0500	
0501	
0502	
0503	

**O/P Memory Address                  Data**

0504	
0505	
0506	
0507	

**Program: Subtraction:**

Memory Address	Mnemonics	Operands
0400	MOV	AX,0000
0403	MOV	DS,AX
0405	MOV	SI,0500
0408	MOV	CX,0000
040B	MOV	AX,[SI]
040D	MOV	SI,0502
0410	SUB	AX,[SI]
0412	JC	0415
0414	INC	CX
0415	MOV	SI,0504
0418	MOV	[SI],AX
041A	MOV	AX,CX
041C	MOV	SI,0506
041F	MOV	[SI],AX
0421	HLT	

**Result:**

**Subtraction:**

**I/P      Memory Address      Data**

0500		
0501		
0502		
0503		

**O/P      Memory Address      Data**

0504		
0505		
0506		
0507		

## EXPERIMENT NO. 3

**AIM:** Write a program for multiplication and division of number by using 8086 microprocessor.

**APPARATUS:** 8086 microprocessor trainer kit, keyboard, power cord.

### **PROGRAM FOR MULTIPLICATION:**

```
MOV SI, 1100H
MOV AX, [SI]
MOV BX, [SI+2]
MOV BX
MOV [SI+4], AX
MOV [SI+6], DX
HLT
```

**PROGRAM:** DIVISION (8086 Program for divide a 16 bit number by 8 bit number)

```
MOV SI, 5000
MOV DI, 6000
MOV BL, [SI]
INC SI
MOV AX, [SI]
DIV BL
MOV [DI], AX
HLT
```

**RESULT:**

**INPUT:**

**MEMORY ADDRESS                  CONTENT**

1100	
1101	
1102	
1103	

**OUTPUT:**

**MEMORY ADDRESS                  CONTENT**

1104	
1105	
1106	
1107	

**DIVISION:**

**INPUT:**

**MEMORY ADDRESS                  CONTENT**

5000	
5001	
5002	

**OUTPUT:**

**MEMORY ADDRESS                  CONTENT**

6000	
6001	

# EXPERIMENT NO. 4

**AIM:** Write a program for logical operation like logical AND, logical OR, logical NOT, logical XOR, logical XNOR.

**APPARATUS:** 8086 Microprocessor trainer kit, keyboard, power cord.

**Program for logical AND operation:**

```
MOV AX, [3000]
MOV BX, [3002]
AND AX, BX
MOV [2000], AX
HLT
```

**Program for logical OR operation:**

```
MOV AX, [3000]
MOV BX, [3002]
OR AX, BX
MOV [3004], AX
HLT
```

**Program for NOT operation:**

```
MOV AX, [3000]
NOT AX
MOV [3002], AX
HLT
```

**Program for NAND operation:**

```
MOV AX, [3000]
MOV BX, [3002]
MOV AX, BX
MOV [3004], AX
NOT AX
```

MOV [3006], AX

HLT

**Program for NOR operation**

MOV AX, [3000]

MOV BX, [3002]

OR AX, BX

MOV [3004], AX

NOT AX

MOV [3006], AX

HLT

**Program for XOR operation:**

MOV AX, [3000]

MOV BX, [3002]

XOR AX, BX

MOV [3004], AX

HLT

**Program for XNOR operation**

MOV AX, [3000]

MOV BX, [3002]

XOR AX, BX

MOV [3004], AX

NOT AX

MOV [3006], AX

HLT

**OUTPUT:**

**AND OPERATION:**

**I/P M/M    CONTENT    O/P M/M    CONTENT**

3000		2000	
3001		2001	
3002			
3003			

**OR OPERATION:**

**I/P M/M    CONTENT    O/P M/M    CONTENT**

<b>3000</b>		<b>2000</b>	
<b>3001</b>		<b>2001</b>	
<b>3002</b>			
<b>3003</b>			

**NOT OPERATION:**

**I/P M/M    CONTENT    O/P M/M    CONTENT**

3000		3002	
3001		3003	

**NAND OPERATION:**

**I/P M/M    CONTENT    O/P M/M    CONTENT**

3000		3004	
3001		3005	
3002		3006	
3003		3007	

**NOR OPERATION:**

**I/P M/M    CONTENT    O/P M/M    CONTENT**

3000		3004	
3001		3005	
3002		3006	
3003		3007	

**XOR OPERATION:**

**I/P M/M    CONTENT    O/P M/M    CONTENT**

3000		3004	
3001		3005	
3002			
3003			

**XNOR OPERATION:**

**I/P M/M    CONTENT    O/P M/M    CONTENT**

3000		3004	
3001		3005	
3002		3006	
3003		3007	

# EXPERIMENT NO. 5

**Aim:** Write a program to find the largest number and Smallest no given string of data using 8086 microprocessor

**Apparatus Required:** 8086 microprocessor trainer kit, keyboard, power cord

**Program:**

**Largest number**

```
MOV SI,2000
```

```
MOV CL,04
```

```
MOV AL,[SI]
```

```
INC SI
```

```
MOV BL,[SI]
```

```
CMP AL,BL
```

```
JNC 0410
```

```
MOV AL,BL
```

```
DEC CL
```

```
JNZ 0407
```

```
INC SI
```

```
MOV [SI],AL
```

```
HLT
```

**Program:**

**Smallest number**

```
MOV SI,2000
```

```
MOV CL,04
```

```
MOV AL,[SI]
```

```
INC SI
```

MOV BL,[SI]

CMP AL,BL

JC 0410

MOV AL,BL

DEC CL

JNZ 0407

INC SI

MOV [SI],AL

HLT

**Result:**

**Largest:**

**I/P    Memory Address                    Data**

2000	
2001	
2002	
2003	
2004	

**O/P    Memory Address                    Data**

2000	
------	--

**Result:Smallest:**

<b>I/P</b>	<b>Memory Address</b>	<b>Data</b>
2000		
2001		
2002		
2003		
2004		

<b>O/P</b>	<b>Memory Address</b>	<b>Data</b>
2005		

# EXPERIMENT NO. 6

**AIM:** TO find factorial of given number using 8086 microprocessor.

**APPARATUS:** 8086 MICROPROCESSOR KIT, keyboard, power cord.

**PROGRAM :**

**FACTORIAL:**

```
MOV SI , 3000
```

```
MOV AL , [SI]
```

```
MOV BL , AL
```

```
DEC BL
```

```
JZ 0410
```

```
MUL BL
```

```
JMP 0407
```

```
INC SI
```

```
MOV [SI] , AL
```

```
HLT
```

**RESULT:**

**I/P M/M**

**DATA**

3000	
------	--

**O/P M/M**

**DATA**

3001	
------	--

# EXPERIMENT NO. 07

**AIM:** TO FIND 1'S AND 2'S COMPLIMENT USING 8086 MICROPROCESSOR.

**APPARATUS :** 8086 MICROPROCESSOR TRAINER KIT, KEYBOARD, POWERBOARD

**PROGRAM:**

MOV AL, [3000]

NOT AL

MOV [3001], AL

INC AL

MOV [3002], AL

HLT

**Result:**

I/P add.	Data	O/P add.	Data
3000		3001	
		3002	

# EXPERIMENT NO. 8

**Aim:** Write a program to arrange a block of data in ascending or descending order using 8086 microprocessor

**Apparatus Required:** 8086 microprocessor trainer kit, keyboard, power cord

**Program:**

**Descending Order:**

Memory Address	Mnemonics	Operands
0400	MOV	SI,0500
0403	MOV	BX,[SI]
0405	DEC	BX
0406	MOV	CX,BX
0408	MOV	SI,0502
040B	MOV	AL,[SI]
040D	INC	SI
040E	COMP	AL,[SI]
0410	JC	0418
0412	XCHG	AL,[SI]
0414	DEC	SI
0415	MOV	[SI],AL
0417	INC	SI
0418	LOOP	040B
041A	DEC	BX
041B	JNZ	0406
041D	HLT	

**Result:**

I/P	Memory Address	Data
	0500	
	0501	
	0502	
	0503	
	0504	
	0505	

0506	
------	--

**O/P    Memory Address                      Data**

0502	
0503	
0504	
0505	
0506	

**Program:**

**Ascending Order:**

<b>Memory Address</b>	<b>Mnemonics</b>	<b>Operands</b>
0400	MOV	SI,0500
0403	MOV	BX,[SI]
0405	DEC	BX
0406	MOV	CX,BX
0408	MOV	SI,0502
040B	MOV	AL,[SI]
040D	INC	SI
040E	CMP	AL,[SI]
0410	JNC	0418
0412	XCHG	AL,[SI]
0414	DEC	SI
0415	MOV	[SI],AL
0417	INC	SI
0418	LOOP	040B
041A	DEC,BX	
041B	JNZ	0406
041D	HLT	

**Result:**

**I/P Memory Address**

**Data**

0500	
0501	
0502	
0503	
0504	
0505	
0506	

**O/P Memory Address**

**Data**

0502	
0503	
0504	
0505	
0506	

# EXPERIMENT NO. 9

**AIM:** WRITE A PROGRAM GCD OF TWO 8 BIT NUMBERS IN TASM SOFTWARE.

DATA SEGMENT

X DB 0B2H

DB 100 DUP (00)

DATA ENDS

CODE SEGMENT

ASSUME CS: CODE, DS: DATA

START: MOV AX, DATA

MOV DS, AX

MOV AX, 0000H

MOV BX, 0000H

MOV AL, X

MOV CL, 0AH

REPEAT: DIV CL

MOV[BX+8], AH

INC BX

MOV AH, 00H

CMP AL, CL

JNC REPEAT

MOV[BX+8], AL

INT 3H

CODE ENDS

END START

**RESULT:-**

# EXPERIMENT NO. 10

AIM: PROGRAM TO CONVERT 8-BIT HEXADECIMAL TO  
BCD

DATA SEGMENT

X DB 48

Y DB 40

DB 100 DUP(00)

DATA ENDS

CODE SEGMENT

ASSUME CS: CODE, DS: DATA

START: MOV AX, DATA

MOV DS, AX

LEA BX, X

MOV AL, X

MOV CL, Y

BACK: CMP AL, CL

JZ EQUAL

JC AL\_LESS

SUB AL, CL

JMP BACK

AL\_LESS: SUB CL, AL

JMP BACK

EQUAL: MOV [BX+8], AL

MOV AX, 0000H

MOV AL, X

MUL Y

DIV CL

MOV [BX+16], AX

INT 3

CODE ENDS

END START

**RESULT:-**

# EXPERIMENT NO. 11

**AIM:** Program to transfer Block of Data using string instruction

CODE:

```
CODE SEGMENT
ASSUME CS:CODE
START: MOV AX, 5000H
      MOV DS, AX
      MOV ES, AX

      MOV SI, 1000H
      MOV DI, 2000H
      MOV CX, 10
      CLD
      REP MOVSB
```

```
      INT 3H
CODE ENDS
END START
```

**RESULT:-**

## EXPERIMENT NO. 12

**AIM:** Assembly language program to exchange two data memory blocks of 10 bytes, which are store in 5000H:1000H and 5000H:2000H.

CODE:-

CODE SEGMENT

ASSUME CS: CODE

START: MOV AX, 5000H

MOV DS, AX

MOV ES, AX

CLD

MOV SI, 1000H

MOV DI, 3000H

MOV CX, 10

REP

MOVSB

MOV SI, 2000H

MOV DI, 1000H

MOV CX, 10

REP

MOVSB

MOV SI, 3000H

MOV DI, 2000H

MOV CX, 10

REP MOVSB

INT 3H

CODE ENDS

END START

**RESULT:-**