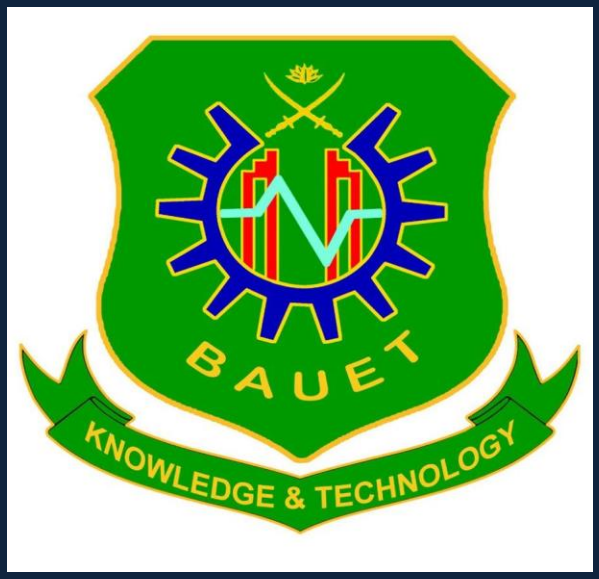
Bangladesh Army University of Engineering & Technology (BAUET)

Department of Computer Science & Engineering



Microprocessor User Lab Manual

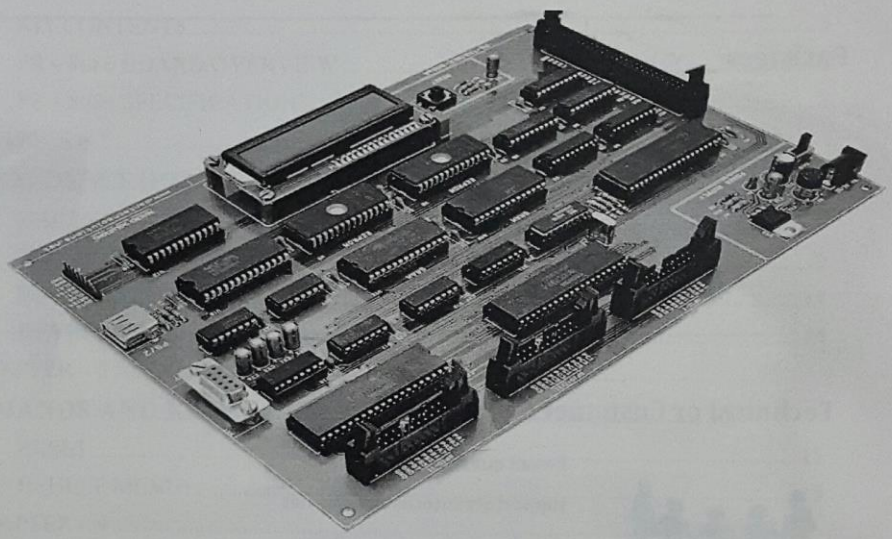
Course code : CSE - 3106

Course Title : Microprocessor and Microcontroller Sessional

Course Lecturer : Md. Sabbir Ejaz

8086 Lab Trainer Kit

PART NO: PS-TRAINER-8086A



USER MANUAL Ver 1.0

PANTECH PROLABS INDIA PVT LTD.

No:41 Santhosh nagar, Rajeswari street,

Kandhanchavadi, Chennai - 600 096

Ph: 044 – 6452 4445/46/47 | Fax: 044 – 4260 6350| www.pantechsolutions.net

BRANCHES @ CHENNAI | COIMBATORE | MADURAI | HYDERABAD | BANGALURU | COCHIN | PUNE

Introduction

Thank you for purchasing the Add-On Cards. You will find it useful in developing your Controller/Processor based applications.

Packages

- 8086 Lab Trainer Kit
- Adaptor -9V
- Serial Cable,User Manual

Technical or Customer Support

E-mail questions to

support@pantechsolutions.net

Send questions by mail to

Pantech Prolabs India Pvt Ltd.,

#41, Santhosh nagar, Rajeshwari street

kandhanchavadi

Chennai – 600 096. Tamilnadu, India

Phone : 91 -98409 74408 /7200083083

Website : www.pantechsolutions.net

For Video Demos

YouTube https://www.youtube.com/user/pantechsolutions

Table of Contents

CHAPTER-1 5
INTRODUCTION 5
1.1 KIT CONTENTS 5
1.2 PS – 8086 BOARD OVERVIEW 6
1.3 PS – 8086 SPECIFICATIONS 7
CHAPTER – 2 8
SYSTEM DESCRIPTION 8
2.1 HARDWARE 8
2.2 MAPPING DEVICE 9
2.3 POWER SUPPLY DETAILS 14
2.4 KEYBOARD DETAILS 14
CHAPTER – 3 15
COMMANDS AND KEYS 15
3.1 RESET 15
3.2 H (HELP MENU) 15
CHAPTER – 4 16
OPERATING INSTRUCTIONS 16
4.1 POWER ON 16
4.2 PROGRAM ENTRY USING ASSEMBLER: 16
4.3 DISASSEMBLER 20
4.4 PROGRAMMING THE 8086 TRAINER KIT: 25
CHAPTER – 5 27
PROGRAMMING DETAILS 27
5.1 PROGRAMMING 8086 OVERVIEW 27
5.2 REGISTERS 27
5.3 BREAK POINT DISPLAY IN LOCAL MODE 28
CHAPTER 6 29
EXAMPLE PROGRAMS 29
6.1 ADDITION 29
6.2 SUBTRACTION 31
6.3 MULTIPLICATION 33

6.4 DIVISION 35
6.5 HEX TO ASCII CODE CONVERSION 37
6.6 BCD TO HEXA DECIMAL CONVERSION 39
6.7 ASCII TO HEX CONVERSION 41
6.8 MATRIX ADDITION 43
6.9 SEPERATING ODD AND EVEN 46
6.10 FIBONACCI SERIES 48
6.11 FACTORIAL OF A NUMBER 50
6.12 LARGEST NUMBER IN AN ARRAY 52
6.13 ASCENDING ORDER: 55
6.14 DESCENDING ORDER 58
CHAPTER 7 61
ON BOARD INTRERFACE 61
7.1 TIMER INTRERFACE 61
7.2 SERIAL COMMUNICATION BETWEEN TWO MICROPROCESSORS 63
CHAPTER 8 66
ADDON INTERFACE CARDS 66
8.1 PS-ADDON 7 SEGMENT DISPLAY 66
8.2 PS-ADDON LED & SWITCH 68
8.3 PS-ADDON KEYPAD 70
8.4 PS-ADDON TRAFFIC LIGHT CONTROLLER 73
8.5 PS-ADDON 8251/8253 INTERFACE CARD 76
8.6 PS ADDON-8255 INTERFACE CARD 83
8.7 PS ADDON-8259 INTERFACE CARD 87
8.8 PS ADDON-8279 INTERFACE CARD 89
8.9 PS-ADDON ADC 0809 INTERFACE CARD 92
8.10 PS-ADDON DAC 0800 INTERFACE CARD 95
8.11 PS ADDON-I2C 7 SEGMENT INTERFACE CARD 98
8.12 PS ADDON STEPPER/DC MOTOR INTERFACE CARD 102
8.13 PS ADDON LCD/GLCD INTERFACE CARD 110

CHAPTER-1

INTRODUCTION

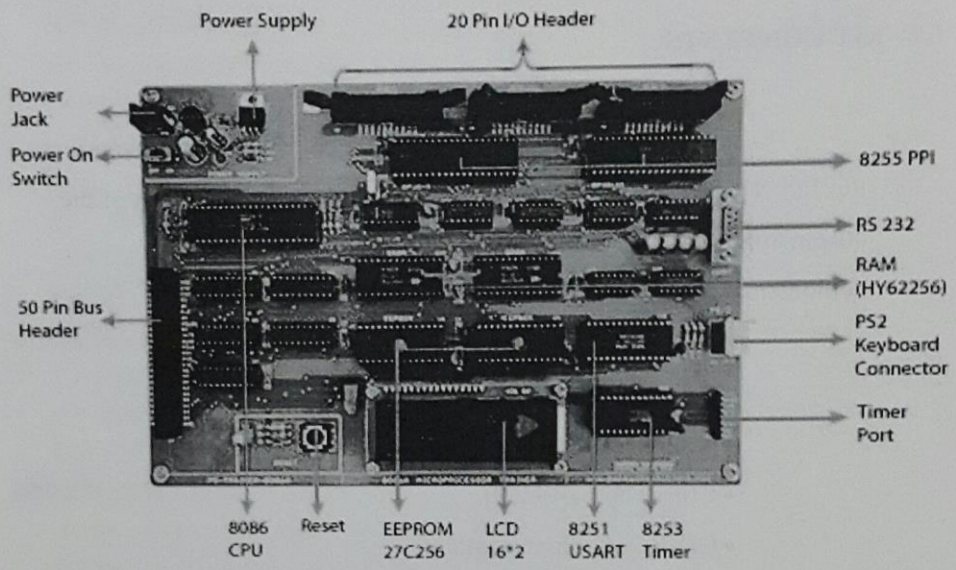
1.1 KIT CONTENTS

Thank you for purchasing PS-8086 Board from Pantech Solutions Pvt Ltd. The PS-8086 board which demonstrates the capabilities of the 40-pin 8086 (various families) Sample programs are provided to demonstrate the unique features of the supported devices.

The PS-8086 Kit comes with the following:

- PS-8086 Board
- Sample devices (INTEL 8086/NEC 8086)
- Cross cable (RS232)
- CD-ROM, which contains:
  a) Sample programs
  b) PS-8086 Board User manual
- Keyboard (101 keys)

1.2 PS – 8086 BOARD OVERVIEW



The PS – 86A board is based on Intel 8086 Microprocessor, which operates at 6.144 MHz using the crystal of 18.432. The board can operate using the 101/104 PC keyboard supplied along with the trainer kit and 2 Line by 16-character LCD display or from the PC (using the Terminal Emulation Software).

Microprocessor's Address, Data and Control bus pins are brought to the 50 pin FRC connector. PS -86A is equipped with powerful software monitor in two-27C256 EPROM.

The monitor supports Video terminal RS232C interface, local 101keyboard and LCD display. The board has 64KB CMOS static RAM (type 62256). PS -86A works on +9V DC.

1.3 PS – 8086 SPECIFICATIONS

- 8086 Microprocessor operating at 18.432 MHz
- 16KB powerful software monitor two 27C256 EPROM
- Three 16-bit programmable timers from 8253
- 48 programmable I/O lines from two nos. of 8255
- Serial interface using 8251
- 50 pin FRC connector for system bus expansion
- 20 pin FRC connector for user interface from 8255
- 9 pin D type connectors for RS 232 interface
- Six different selectable baud rates from 150 to 9600
- 101 PC type keyboard for entering user address/data and for commands
- Built in line-by-line assemble and disassemble
- User friendly software monitor for loading and executing programs with break point facility

CHAPTER – 2

SYSTEM DESCRIPTION

2.1 HARDWARE

PROCESSOR CLOCK FREQUNCY:

8086 operates at 18.432 MHz clock.

MEMORY:

Monitor EPROM: 0000 –FFFF (SEGMENT)

System RAM: 0000 –FFFF (SEGMENT)

1000 – FFFF (Reserved For Monitor program)

User RAM Area: 1100 – FFFF

ALLOCATION OF EPROM

START ADDRESS	END ADDRESS	SOCKET NO	IC USED	TOTAL CAPACITY
0000	FFFF	U9 U8	27256 27256	32 K BYTE 32 K BYTE

ALLOCATION OF RAM

START ADDRESS	END ADDRESS	SOCKET NO	IC USED	TOTAL CAPACITY
0000	FFFF	U10 U11	62256 62256	32 K BYTE 32 K BYTE

2.2 MAPPING DEVICE

PARALLEL INTERFACE

8255 - Programmable peripheral interface.

The following are the I/O addresses for 8255(GPIO I):

SOCKET.NO	FUNCTION	ADDRESS	CONNECTOR.NO
U22	CONTL REG	FF26	J8 GPIO I J9(GPIO I&GPIOII)
	PORT A	FF20	
	PORT B	FF22	
	PORT C	FF24	

The following are the I/O addresses for 8255(GPIO II):

SOCKET.NO	FUNCTION	ADDRESS	CONNECTOR.NO
U16	CONTL REG	FF36	J6 GPIO II J9(GPI0 I&GPIOII)
	PORT A	FF30	
	PORT B	FF32	
	PORT C	FF34	

TIMER INTERFACE

8253 - Programmable Interval Timer:

CHANNEL 2:

Input clock : 3 MHz

Output clock: Depends on selection of baud rate.

Used for : Baud rate generation for 8521 USART.

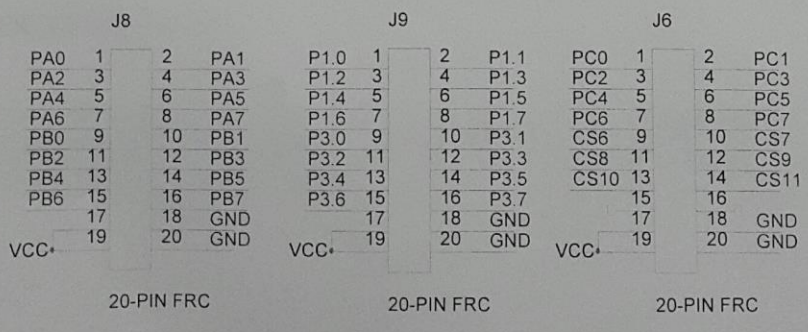
I/O ADDRESS:

SOCKET.NO	FUNCTION	ADDRESS	CONNECTOR.NO
U12	CONTL REG	FF06	J2
	CHENNAL 0	FF00	
	CHENNAL 1	FF02	
	CHANNEL 2	FF04	

CONNECTOR DETAILS

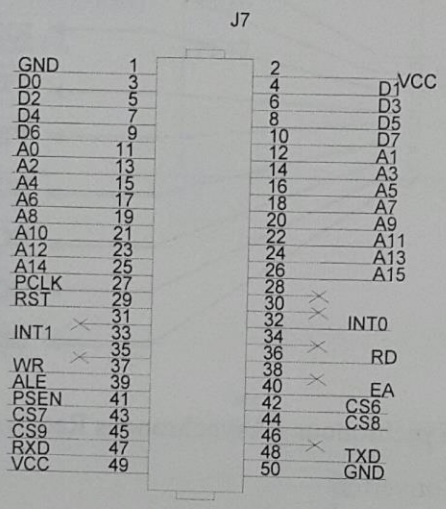
20 PIN EXPANSION CONNECTORS

The 20 Pin FRC connector is used to interconnect with the Interface cards like ADC, DAC, SWITCH/LED, RELAY buzzer Interfaces etc. Pin details are given below

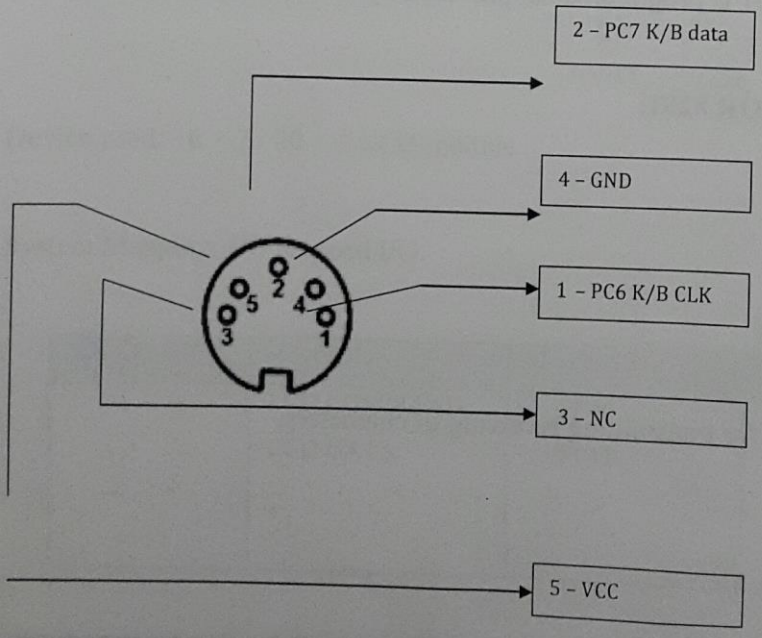


50 PIN EXPANSION CONNECTOR

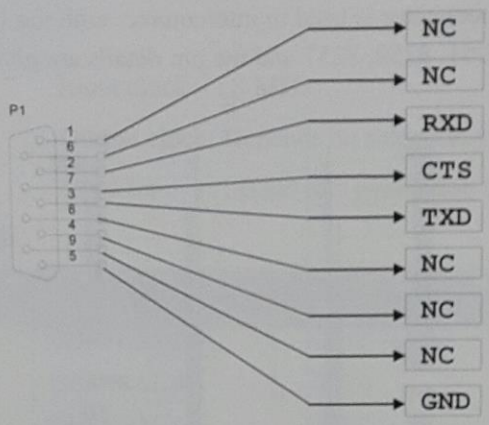
The 50 Pin FRC connector is used to interconnect with the Interface cards like 8255, 8279, 8253/8251, 8259, 8257 and the pin details are given below



KEYBOARD CONNECTOR



'9PIN 'D' TYPE (FEMALE):



8251 - Universal Synchronous / Asynchronous Receiver / Transmitter.

RS232 Bridge Converter

BAUD CLOCK:

Baud clock for 8251 is programmable, provided by Channel 2 of 8253

INPUT CLOCK FOR 8251:

3.072 MHz

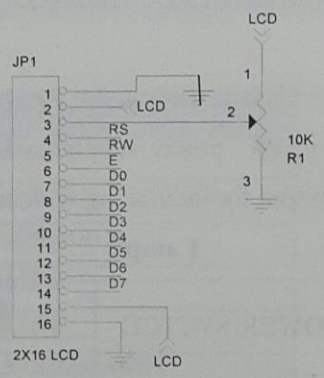
DRIVERS USED:

MAX 232 is used for transmitting receiving of characters.

8251 UART I/O ADDRESS:

SOCKET.NO	FUNCTION	ADDRESS	CONNECTOR.NO
U15	8251 CONTL REG 8251 DATA	FF10 FF12	D2(SKT)

LCD INTERFACE:



Device used: 16 × 2 / 20 × 4 LCD module

System Mapping: I/O mapped I/O.

SOCKET.NO	FUNCTION	ADDRESS	CONNECTOR.NO
----	LCD COMMAND LCD DATA	FF40 FF42	-----

RESET

This key is located in the main 8086 board. On depressing this key the program starts executing from the beginning or reset address 0000. On power on reset it. Display **PS - 86** in local LCD display.

2.3 POWER SUPPLY DETAILS

PS trainer kit will work at 0 – 5v (1 amp) from the PS power supply. Provision is made in PS power supply to bring out on the front panel DC regulated voltage output for interfacing with add-on cards.

+5V 1 amp

POWER SWITCH	■	Supply Turned OFF
	▬	Supply Turned ON

2.4 KEYBOARD DETAILS

101 PC type keyboard is interfaced to Microcontroller through its port pin. Communication between keyboard and Microcontroller takes place using 2 wires – one for serial clock and serial data **(P1.6 and P1.7).**

CHAPTER – 3

COMMANDS AND KEYS

3.1 RESET

This key is located in the main PS-86A board .On depressing this key the programs. Starts executing from the beginning or reset address 0000. On power on reset it. Displays **PS- 86A** in local LCD display

PS – 86
–

3.2 H (HELP MENU)

This key is used go PS- 86A help menu and it will display the following commands.

KEY	FUNCTIONS
A <BEG>	ASSEMBLE
B	BAUD RATE
D <BEG>	DISASSEMBLE
E <BEG><END>	EXAMINE
G <BEG>	EXECUTE
H	HELP COMMANDS
I <INSERT>	INTERNAL RAM
L <OFFSET>	DOWN LODE
M <ADDR>	MODIFY
N	NORMAL MODE
Q	QUIT
R <REG>	REGISTER DISPLAY
S	SERIAL TRANSFER
T <START><END><DS>	BLOCK TRENSFER
U <BEG><END>	UP LODE
X	DELETE BLACK MEMORY
?	INSTRUCTIONS

CHAPTER – 4

OPERATING INSTRUCTIONS

4.1 POWER ON

Connect the PS – 8051 board to the power having the following specifications.

+9V DC 1 Amp

Switch on the power supply after ensuring the correct voltages. Following message will appear on the LCD display.

PS -- 86
! ! ! !

On power on or after reset the display shows **PS – 86 as** a sign on message. The prompt character – is displayed in the next line informing the user, that the board is ready to accept the commands.

4.2 PROGRAM ENTRY USING ASSEMBLER:

Instruction follows

ENTERING MNEMONICS

EXAMPLE:

Press H for help
A1100

Enter the starting Address

Press Enter Key ↵ Enter

User program starts from address 1100 and displays the following and waits for the user data to be typed in the second line

EXAMPLE:

0000 : 1100:
MOV AX,1212

Enter the mnemonics

Press Enter Key ↵ Enter

0000 : 1103:
MOV BX,1212

Enter the mnemonics

Press Enter Key ↵ Enter

Program end.

EXIT COMMAND

Double Enter you get the main menu

PS 86
–

PROGRAM ENTRY USING OPCODE

ENTERING THE OPCODE

Modify Memory

Press H for help **_M1100**

Enter the starting Address

Press Enter Key ↵ Enter

0000:1100: **18 _**

0000:1100: **18 B8_**

Enter the opcode

Enter the Space Bar Key

0000:1101: **34 12_**

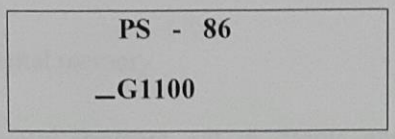
Enter the opcode

Enter the Space Bar Key

PROGRAM END. EXIT COMMAND:

Double Enter you get the Main Menu

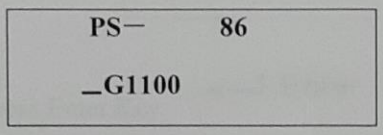
ENTERING 'G'EXECUTING COMMAND



Enter starting address

Press Enter Key ↵ Enter

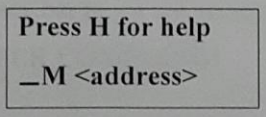
After executing display



Executing display

To EXIT Execution Mode PRESS 'RESET 'Switch

ENTERING RESULT COMMAND:



Enter the Memory Location

Press Enter Key ↵ Enter

0000:1200: **24 _**

You get the output

8-bit Data

4.3 DISASSEMBLER

Disassemble converts the hex byte stored in the memory into equivalent mnemonics. To enter into disassemble mode, type D in the command mode followed by the memory address.

EXAMPLE:

Press H for help **_D1100**

Enter the starting address

Press Enter Key ↵ Enter

1100: B8 12 12 **MOV AX,1212**

Enter the Space Bar Key

1103: BB 12 12 **MOV AX,1212**

Enter the space bar key

M (MODIFY EXTERNAL MEMORY)

Using this command the user can display/modify any external memory address.

Modify External memory

Press H for help **_M1100**

Enter the starting Address

R (REGISTER DISPLAY)

EXAMPLE:

Press H for help **_R**

Enter the starting Address

Press Enter Key ↵ Enter

AX=1104

Enter the Space Bar Key

BX=1204

Enter the space bar to see the remaining registers

T (TRANSFER COMMAND)

EXAMPLE:

Press **'T '**

The source segment addresses 0000. The above command transfer the memory content starting from source start address 1100 to destination start address 1200 till source end address 1500 is reached.

Src seg address	**0000**
Starting address	**1100**
End address	**1200**
Destination address	**1500**

Block Transfer
Src seg : 0000

Enter Press Enter Key ↵ Enter

start : 1100
end : 1200

Enter the 1100 address

Enter the 1200address

Press Enter Key ↵ Enter for exit command

dest : 0 : 1500

Enter Press Enter Key ↵ Enter

Transfer Complete

Press Enter Key ↵ Enter for exit command

N (LOCAL MODE)

When this key is depressed on PC keyboard, the PS – 8051 Kit starts working through local 101 keyboard. Serial communication is disabled. Following message will appear in the LCD display

! Normal Mode!

B (BAUD RATE)

Press the **'B'**

Cur BAUD : 2400 150

Enter the Space Bar Key

Cur BAUD : 2400 9600

Enter Press Enter Key ↵ Enter

To Set the 9600 baud rate

Baud rates : 150, 300, 600, 1200, 2400, 4800, **9600**

When using the serial Communication.

S (SERIAL MODE KEY)

When this key is depressed the system start communicating through connector. All keys are disabled except reset.

! Serial Mode!

The system displays the message SERIAL MODE. To come back to LCD mode (Normal Mode) user has to press the 'N' key in the computer keyboard otherwise press the Reset button.

4.4 PROGRAMMING THE 8086 TRAINER KIT:

PROCEDURE 1: TO ENTER THE MNEMONICS

Initially connect the 9V adaptor to J10 connector

- Switch **ON** the **PS-8086** kit using slide Switch **SW1**
- "**PS - 86**" will be displayed on the LCD
- Connect the **Keyboard** in **PS**/2 connector
- Depress "**A**" starting address of the program for Ex: **A1100**

  For ex: **A1100** enter key

  Type the mnemonics **MOV AX, 1212press** Enter key

  Type the mnemonics **MOV BX, 1212 press** Enter key and continue the same procedure till the end of the program

ADDRESS	OPCODES	MNEMONICS
1100	B8 12 12	MOV AX,1212
1103	BB 12 12	MOV BX,1212
1106	01 D8	ADD AX,BX
1108	BE 00 12	MOV SI,**1200**
110B	89 04	MOV [SI],AX
110D	F4	HLT

- To verify the code depress **D** starting address and depress space bar to see next memory location

  For Ex: **D1100** and press spacebar till the end of the program

- To execute the program Depress "G staring address for Ex: **G1100**
- To see the result depress "M result address" for Ex: **M1200.**
- To view the output in the Register depress **'R'** and press enter key in keyboard.

PROCEDURE 2: TO ENTER THE OPCODE

Follow the same procedure till step 4

- Depress "**M**" starting address of the program for Ex: **M1100**

  For ex: **M1100** press enter

  Type the opcode B8 space bar

  Type the opcode 12 space bar and continue the same till the end of the program

ADDRESS	OPCODES	MNEMONICS
1100	B8 12 12	MOV AX,1212
1103	BB 12 12	MOV BX,1212
1106	01 D8	ADD AX,BX
1108	BE 00 12	MOV SI,1200
110B	89 04	MOV [SI],AX
110D	F4	HLT / INT 03

- To view the code depress **D** starting address and depress space bar to see next memory location

  For Ex: **D1100** and press spacebar till the end of the program
- To execute the program Depress "G staring address for Ex: **G1100.**
- To see the result depress "M result address" for Ex: **M1200.**
- To view the output in the Register depress **'R'** and press enter key in keyboard

Note: 1) **"M"** is used for displaying the result, for Ex: **M8500**

2) **"M"** is used to entering the Opcode.

3) **"M"** is used for entering the data.

Note: There are two ways to enter the program

1) Mnemonics method

2) Opcode method

Sample program is given to enter the program in both the methods

CHAPTER – 5

PROGRAMMING DETAILS

5.1 PROGRAMMING 8086 OVERVIEW

- The 8086 Microprocessor uses a multiplexed 16 bit address and address bus
- During the first clock of machine cycle the 16 bit address s sent out on address/data bus
- These 16 bit addresses may be latched externally by the address latch enable signals(ALE)
- 8086 Microprocessor can access 1024kb of external memory using its 20 bit address and memory read/write signals
- The 8086 provide s0, s1 and s2 signals for bus control.
- The 8086 Microprocessor has a 16 bit program counter (IP) and 16 bit stack pointer (sp)

It has following set of 16 bit Registers:

AX –Accumulator

BX, CX, DX (These four register can be used as two 8 bit register individually)

5.2 REGISTERS

INDEX REGISTER

SI → Source index

DI → Destination index

BP → Base pointer index

SEGMENT REGISTER

CS → Code segment register

DS → Data segment register

ES → Extra segment register

SS → Stack segment register

FL → Flag register

INTERRUPTS

The 8086 have two interrupt

- External mask able interrupt (INTR)
- Non mask able interrupt (NMI)

5.3 BREAK POINT DISPLAY IN LOCAL MODE

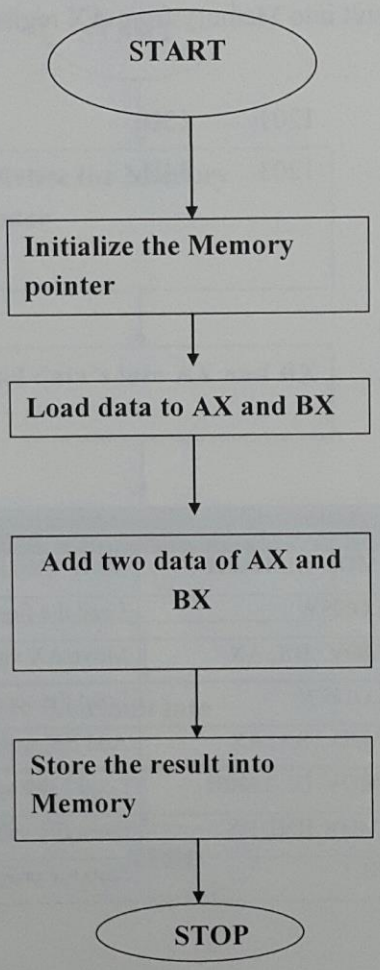
When break point is encountered, all the register values are saved and the Acc. "**AX=XXXX** "Value is displayed in the LCD display. Now use SPACE key to check register values one by one

CHAPTER 6

EXAMPLE PROGRAMS

6.1 ADDITION

FLOW CHART



ALGORITHM:

- Initialize the pointer to the memory for data and result.
- Load the data into AX, BX.
- Add the two data of AX and BX registers.
- Store the result into Memory from AX registers.

INPUT:

- 1200 13H 1201 13H.
- 1202 14H 1203 14H.

OUTPUT:

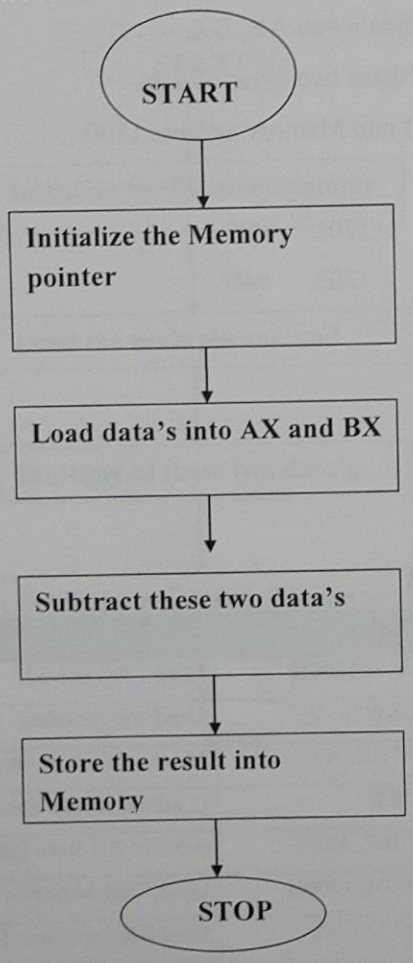
- 1300 27H.
- 1301 27H.

PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	BE 00 12	MOV SI, **1200H**	Move 1200 into SI pointer
1103	AD	LODSW	Load the first data into AX
1104	89 C3	MOV BX, AX	Move AX value into BX
1106	AD	LODSW	Load the second data into AX
1107	01 C3	ADD BX, AX	Add BX and AX registers
1109	BF 00 13	MOV DI, **1300H**	Load 1300 address location into DI
110C	89 1D	MOV [DI], BX	Store BX value into memory
110E	F4	HLT	Stop the program

6.2 SUBTRACTION

FLOW CHART:



ALGORITHM:

- Initialize the pointer to the memory for data and result.
- Load the two data's into AX, BX.
- Subtraction of these two bytes of data.
- Store the result into Memory address 1300.

INPUT:

- 1200 08H 1201 08H
- 1202 04H 1203 04H

OUTPUT:

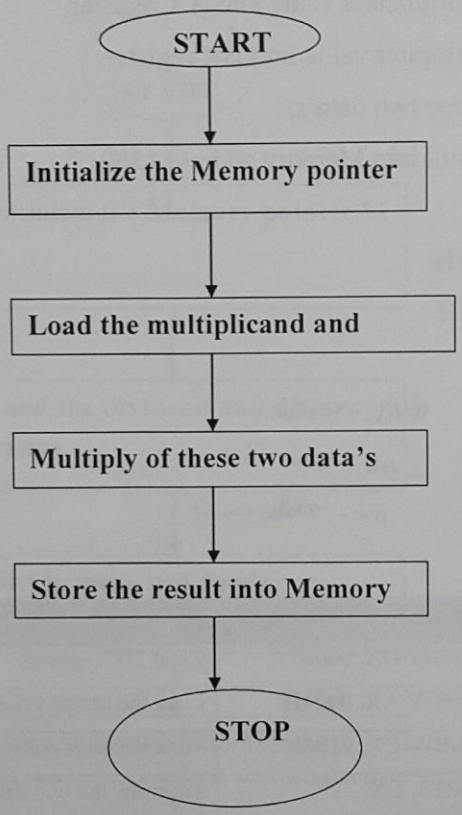
- 1300 04H
- 1301 04H

PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	BE 00 12	MOV SI,**1200H**	Load 1200 into SI
1103	AD	LODSW	Load the first data
1104	89 C3	MOV BX, AX	Move AX value into BX
1106	AD	LODSW	Load the second data
1107	29 C3	SUB BX, AX	subtract AX from BX
1109	BF 00 13	MOV DI, **1300H**	Load 1300 address into DI
110C	89 1D	MOV [DI],BX	Load BX value into DI
110E	F4	HLT	Stop the program

6.3 MULTIPLICATION

FLOW CHART:



ALGORITHM:

- Initialize the pointer to the memory for result.
- Load the multiplicand value into AX register.
- Load the multiplier value into BX register.
- Multiply these two data's.
- Store the result into Memory address 1300.

INPUT:

- AX 0404H
- CX 0202H

OUTPUT:

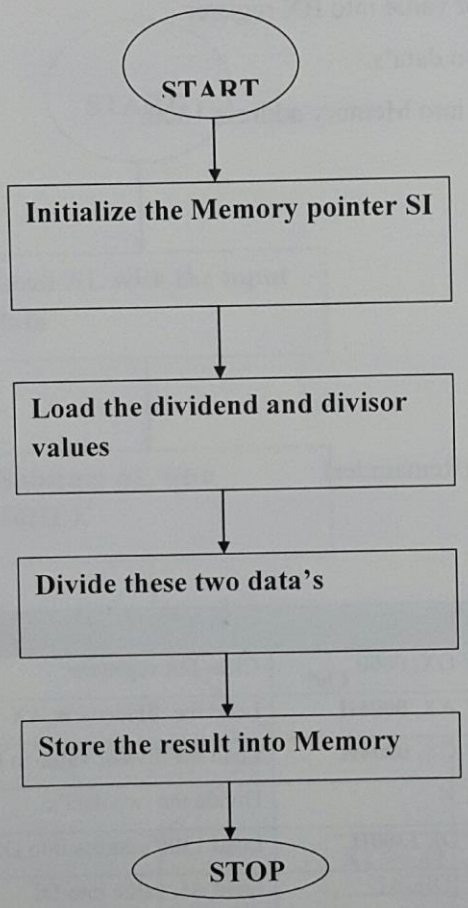
- 1300 08H AL - 8 bits
- 1301 10H AH - 8 "
- 1302 08H DX - 8 bits Extra)

PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	BA 00 00	MOV DX, 0000	Clear DX registers
1103	B8 06 00	MOV AX, **0404H**	Load the multiplicand in AX
1106	B9 02 00	MOV CX, **0202H**	Load the multiplier value in BX
1109	F7 F1	MUL CX	Multiply the two data's
110B	BF 00 13	MOV DI, **1300H**	Load 1300 address into DI
110E	88 05	MOV [DI], AL	Load AL value into DI
1110	47	INC DI	Increment DI 1300+1 =1301
1111	88 25	MOV [DI], AH	Load AH value into DI
1113	47	INC DI	Increment DI 1302
1114	89 15	MOV [DI], DX	Load DX value into DI
1116	F4	HLT	End

6.4 DIVISION

FLOW CHART:



ALGORITHM:

- Initialize the pointer to the memory for result.
- Load the dividend value into AX register.
- Load the divisor value into BX register.
- Divide these two data's.
- Store the result into Memory address 1300.

INPUT:

- AX 0006H
- CX 0004H

OUTPUT:

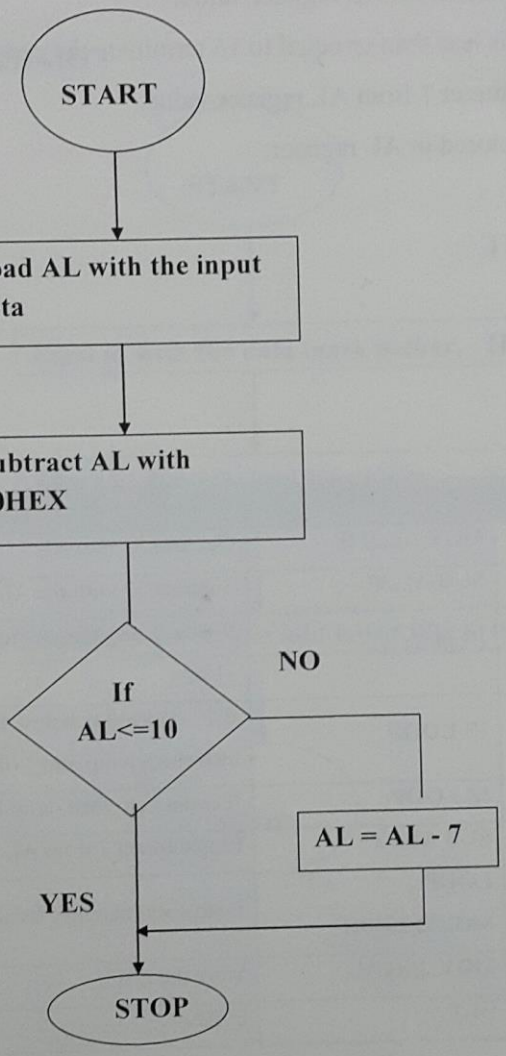
- 1300 01H
- 1301 00H
- 1302 02H (Remainder)

PROGRAM

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	BA 00 00	MOV DX, 0000	Clear DX registers
1103	B8 06 00	MOV AX, **0006H**	Load the dividend in AX
1106	B9 04 00	MOV CX, **0004H**	Load the divisor value in BX
1109	F7 F1	DIV CX	Divide the two data's
110B	BF 00 13	MOV DI, **1300H**	Load 1300 address into DI
110E	88 05	MOV [DI], AL	Load AL value into DI
1110	47	INC DI	Increment DI
1111	88 25	MOV [DI], AH	Load AH value into DI
1113	47	INC DI	Increment DI
1114	89 15	MOV [DI], DX	Load DX value into DI
1116	F4	HLT	End

6.5 HEX TO ASCII CODE CONVERSION

FLOW CHART:



ALGORITHM:

- Load the input data in AL register.
- Subtract 30 from AL register value.
- If data is less than or equal to 16 terminate the program.
- Else subtract 7 from AL register value.
- Result stored in AL register.

INPUT:

- AL 31H

OUTPUT:

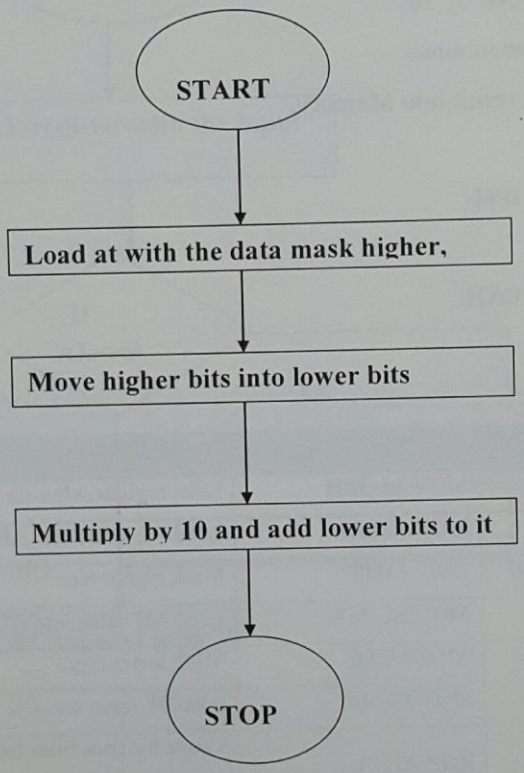
- 1300 01H

PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	B0 31	MOV AL,**31H**	Get data 31 into AL
1102	2C 30	SUB AL,30	Subtract 30 with the AL
1104	3C 10	CMP AL,10	If data is less than or equal to 16 go to 110C
1106	72 04	JB **LOOP**	If 1st operand is below the 2nd operand then short jump into 110C
1108	74 02	JZ **LOOP**	If count zero then jump into to 110C
110A	2C 07	SUB AL,07	Else subtract 7 from AL register value
110C	BE 00 13	**LOOP**: MOVSI,**1300H**	Load 1300 memory location
110F	88 04	MOV [SI],AL	Store the result
1111	F4	HLT	END

6.6 BCD TO HEXA DECIMAL CONVERSION

FLOW CHART



ALGORITHM:

- Load the data in AL register.
- Separate higher nibbles and (in) lower nibbles.
- Move the higher nibbles (in) to lower nibbles position.
- Multiply AL by 10.
- Add lower nibbles.
- Store the result into Memory.

INPUT:

- AL 10H

OUTPUT:

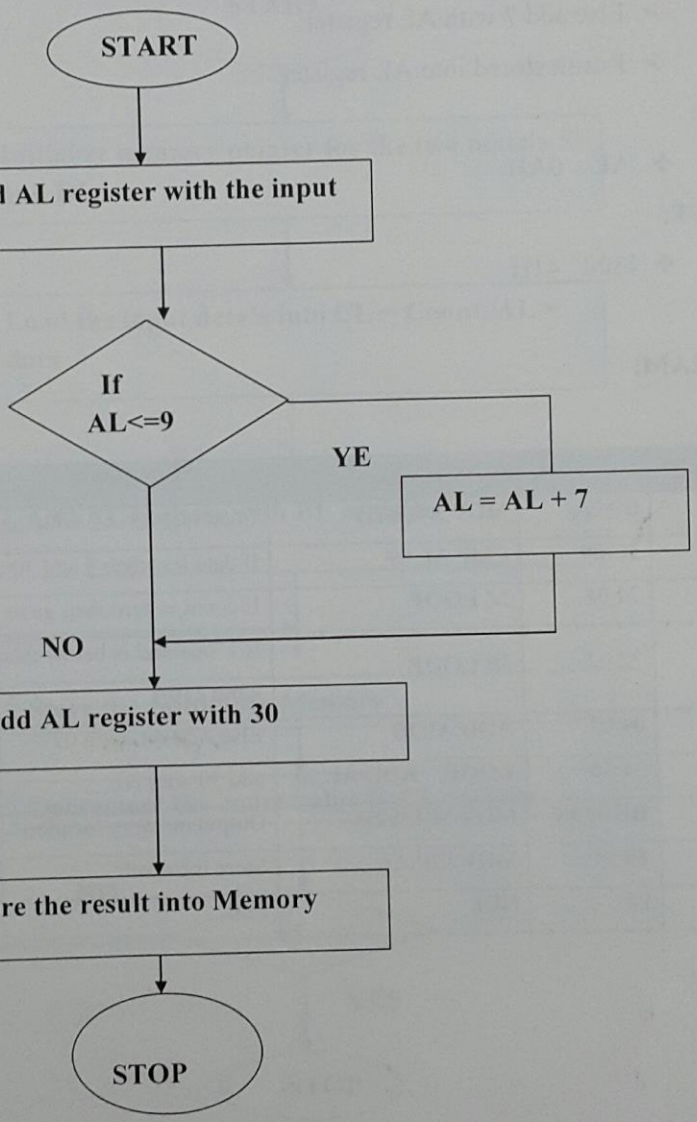
- 1300 0AH

PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	B0 10	MOV AL,**10H**	Load register AL with the data 10
1102	88 C4	MOV AH,AL	Load AL value into AH
1104	80 E4 OF	AND AH,0F	Mask higher bits
1107	88 E3	MOV BL,AH	Load AH value into BL
1109	24 F0	AND AL,F0	Mask lower bits
110B	B1 04	MOV CL,04	Load 04 value into CL
110D	D2 C8	ROR AL,CL	Rotate the data from last 4bits to first 4 bits
110F	B7 0A	MOV BH,0A	Load 10 value into BH
1111	F6 E7	MUL BH	Multiply by 10
1113	00 D8	ADD AL,BL	Add lower nibble to the multiplied data
1115	BE 00 13	MOV SI,**1300H**	Output memory location
1118	88 04	MOV [SI],AL	Store the result
111A	F4	HLT	End

6.7 ASCII TO HEX CONVERSION

FLOW CHART:



ALGORITHM:

- Load AL with the input data.
- Check If (AL<=9) then add 30 with AL register.
- Else add 7 with AL register.
- Result stored into AL register.

INPUT:

- AL 0AH

OUTPUT:

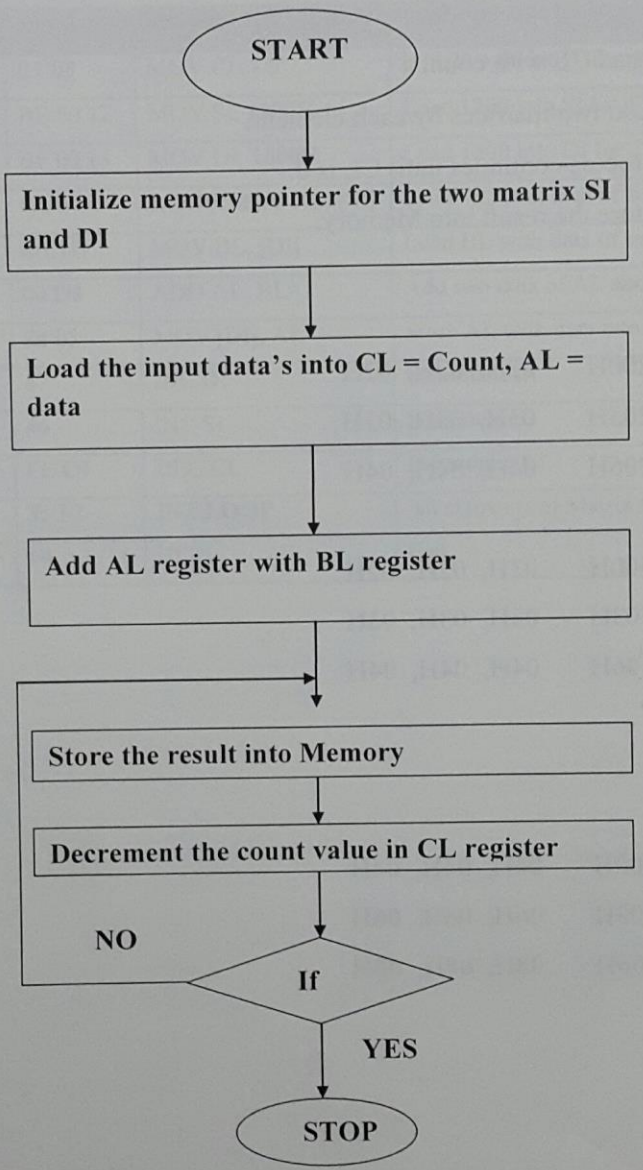
- 1300 41H

PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	B0 0A	MOV AL,**0AH**	Load register AL with the data 10
1102	3C 09	CMP AL,09	If data less than 9 add 30 to the data
1104	74 04	JZ **LOOP**	If count is zero then go to 110A
1106	72 02	JB **LOOP**	If 1st operand is below than 2nd operand jump 110A
1108	04 07	ADD AL,07	Else Add AL with 07
110A	04 30	**LOOP:** ADD AL,30	add 30 with AL
110C	BE 00 13	MOV SI,**1300H**	Output memory location
110F	88 04	MOV [SI],AL	Store the result
1111	F4	HLT	End

6.8 MATRIX ADDITION

FLOW CHART:



ALGORITHM:

- Initialize the pointer to memory for data and result.
- Load CL with count.
- Add two matrices by each element.
- Process continues until CL is 0.
- Store the result into Memory.

INPUT:

- 1200H 02H, 02H, 02H
- 1203H 03H, 03H, 03H
- 1206H 04H, 04H, 04H

- 1300H 02H, 02H, 02H
- 1303H 03H, 03H, 03H
- 1306H 04H, 04H, 04H

OUTPUT:

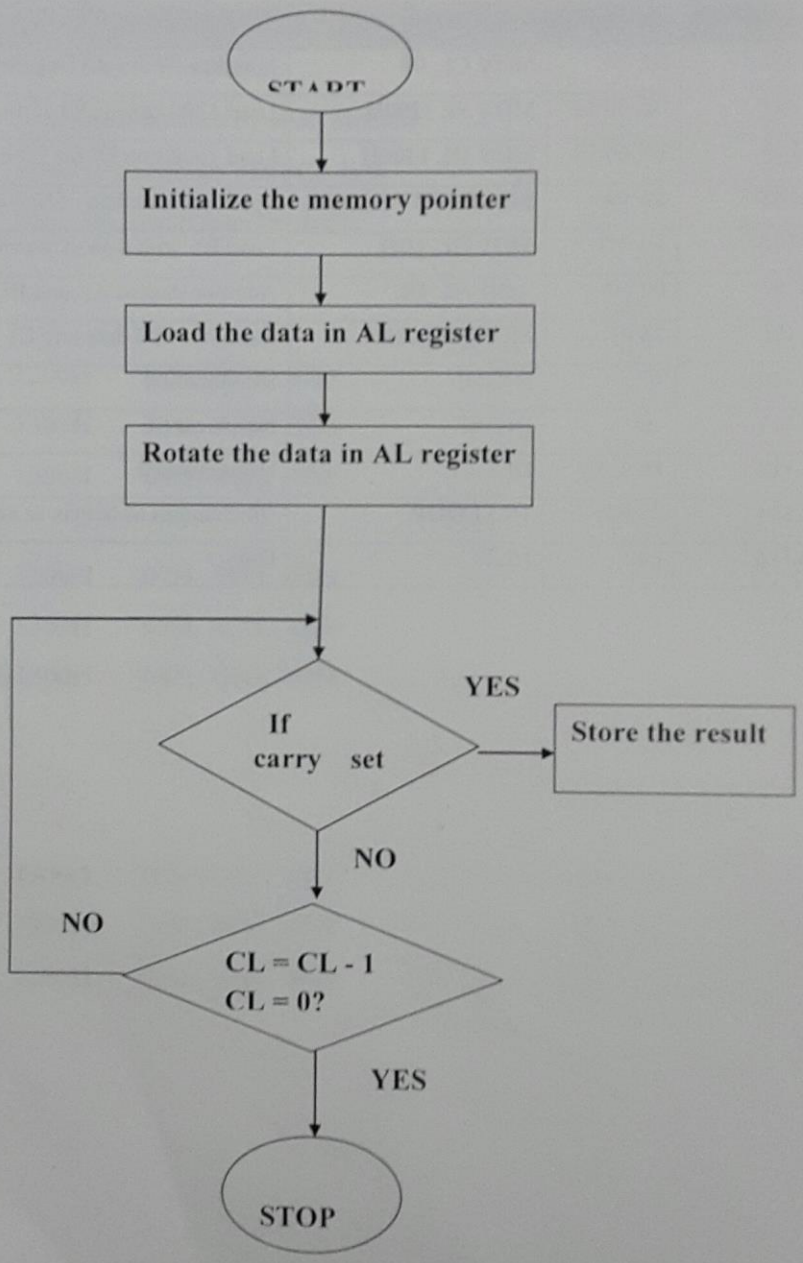
- 1300H 04H, 04H, 04H
- 1303H 06H, 06H, 06H
- 1306H 08H, 08H, 08H

PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	B1 09	MOV CL, 09	Initialize 09 into CL register
1102	BE 00 12	MOV SI, **1200H**	Load 1200 into SI for 1st matrix
1105	BF 00 13	MOV DI, **1300H**	Load 1300 into DI for 2nd matrix
1108	8A 04	**LOOP:**MOV AL, [SI]	Load AL with data of first matrix
110A	8A 1D	MOV BL, [DI]	Load BL with data of second matrix
110C	00 D8	ADD AL, BL	Add two data of AL and BL
110E	88 05	MOV [DI], AL	Store AL with data into DI
1110	47	INC DI	Increment DI
1111	46	INC SI	Increment SI
1112	FE C9	DEC CL	Decrement CL
1114	75 F2	JNZ **LOOP**	all elements of Matrix to added
1116	F4	HLT	End

6.9 SEPERATING ODD AND EVEN

FLOW CHART



ALGORITHM:

- Initialize the pointer to memory for data and result.
- Loaded the data in AL register from memory.
- Rotate the AL register by one bit.
- If carry flag is set then go to step2.
- Store the even number as a result into the Memory.

INPUT:

- 1200H 00H, 01H, 02H, 03H
- 1204H 04H, 05H, 06H,07H

OUTPUT:

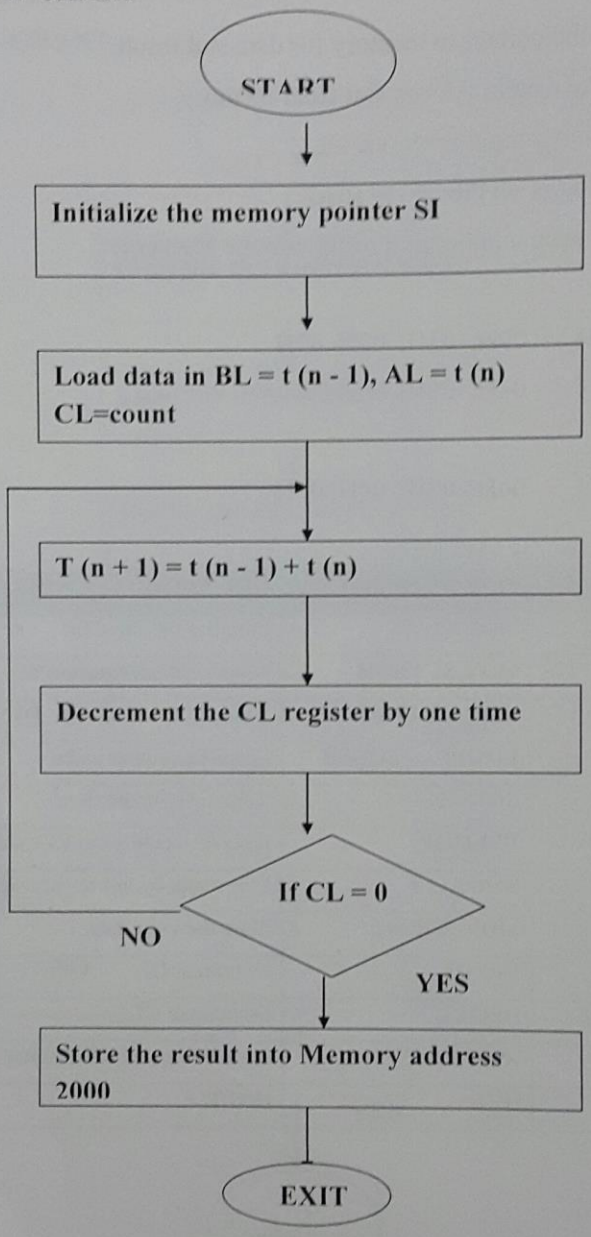
- 1300H 00H, 02H, 04H, 06H

PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	B1 08	MOV CL, 08	Initialize 08 into CL
1102	BE 00 12	MOV SI, **1200H**	Load 1200 address into SI
1105	BF 00 13	MOV DI, **1300H**	Load 1300 address into DI
1108	AC	**LOOP:** LODSB	Load the counter value
1109	D0 C8	ROR AL,1	Rotate AL in one time
110B	72 FB	JB **LOOP**	If carry occurs go to L1 (odd Data)
110D	D0 C0	ROL AL, 1	Else rotate by left to get original data
110F	88 05	MOV [DI], AL	Store the even data
1111	47	INC DI	Increment DI
1112	FE C9	DEC CL	Decrement CL
1114	75 F2	JNZ **LOOP**	Loop executes until counter is zero
1116	F4	HLT	HALT

6.10 FIBONACCI SERIES

FLOW CHART



ALGORITHM:

- Initialize the pointer to memory for storing result.
- Number of the counts loaded into CL register.
- T (n + 1) = t (n) + t (n - 1).
- Repeat the above process until count is 0.

INPUT:

- CL 10H

OUTPUT:

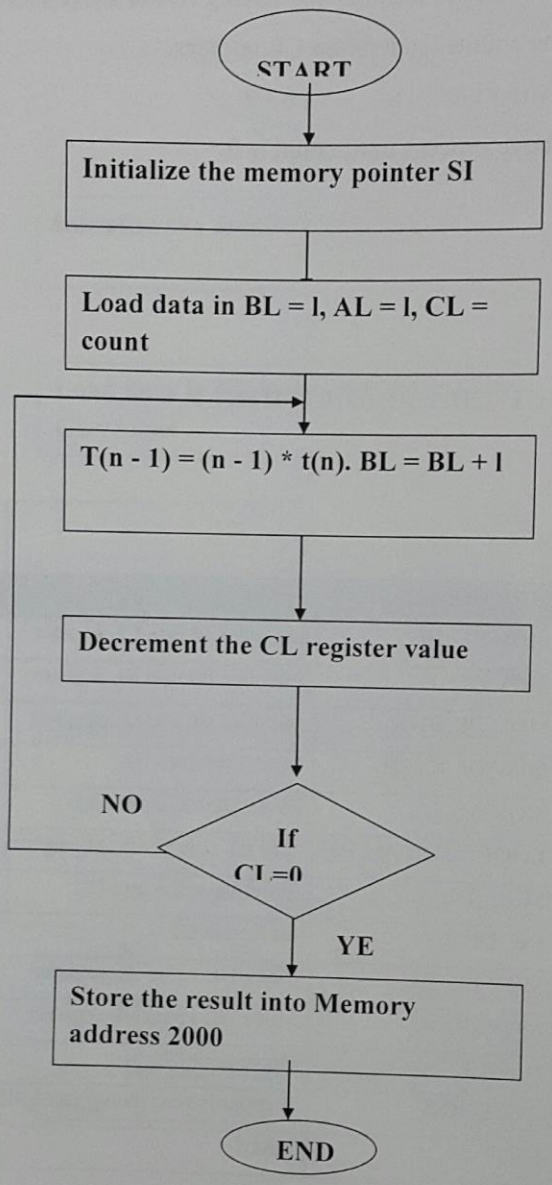
- 1300H 01H, 02H, 03H, 05H, 08H etc…

PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	B1 10	MOV CL, 10	Initialize 10 into CL register
1102	B3 00	MOV BL, 00	Initialize 00 into BL register
1104	B2 01	MOV DL, **01H**	Initialize 01 into DL register
1106	BF 00 13	MOV DI, **1300H**	Load 1300 into DI
1109	88 D0	MOV AL, DL	Move DL value into AL
110B	00 D8	**LOOP:** ADD AL, BL	Add BL value with AL register
110D	88 05	MOV [DI],AL	Store AL value into DI.
110F	47	INC DI	Increment DI
1110	88 D3	MOV BL, DL	Move DL value BL register
1112	88 C2	MOV DL, AL	Move AL value DL register
1114	FE C9	DEC CL	Decrement CL
1116	75 F3	JNZ **LOOP**	If count is zero then go to 110B
1118	F4	HLT	HALT

6.11 FACTORIAL OF A NUMBER

FLOW CHART:



n!=n*(n-1)*(n-2)*……..*1

ALGORITHM:

- Load the counter with value of n into CL register.
- T (n) = t (n - 1) * t (n - 2).
- Repeat the process until n becomes to store result.
- Initialize the pointer to memory to store result.
- Store the result into Memory address 1300.

INPUT:

- CL 04H (4 **factorial 1 X 2 X 3 X 4 =24 decimal value**)

OUTPUT:

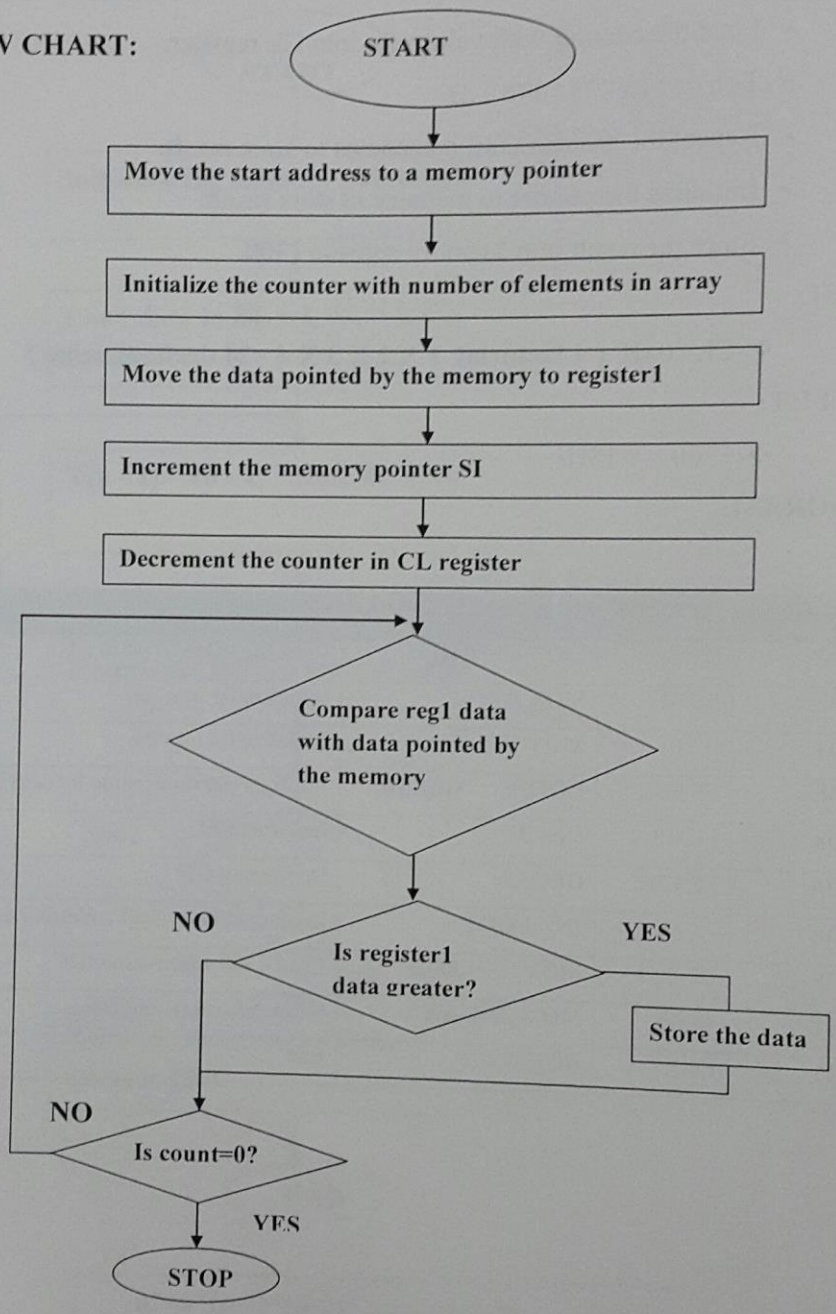
- 1300 18H

PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	B1 04	MOV CL, **04H**	Load the value of 04 in CL
1102	B0 01	MOV AL, 01	Initialize 01 into AL
1104	B3 01	MOV BL, 01	Initialize 01 into BL
1106	F6 E3	**LOOP**: MUL BL	Multiply previous value by next Value
1108	FE C3	INC BL	Increment BL
110A	FE C9	DEC CL	Decrement CL
110C	75 F8	JNZ **LOOP**	Loop continues until count is Zero
110E	BF 00 13	MOV DI, **1300H**	Load 1300 address into DI
1111	89 05	MOV [DI], AX	Store AX value into DI
1113	F4	HLT	End

6.12 LARGEST NUMBER IN AN ARRAY

FLOW CHART:



ALGORITHM:

- Take the first number of the array.
- Compare with next number.
- Take the bigger one of the them.
- Decrement the count in CL register.
- If the count is not zero then continue from step 2.
- Store the result into Memory address 1300.

INPUT:

- 1200H 05H, 09H, 01H, 02H,07H,08H

OUTPUT:

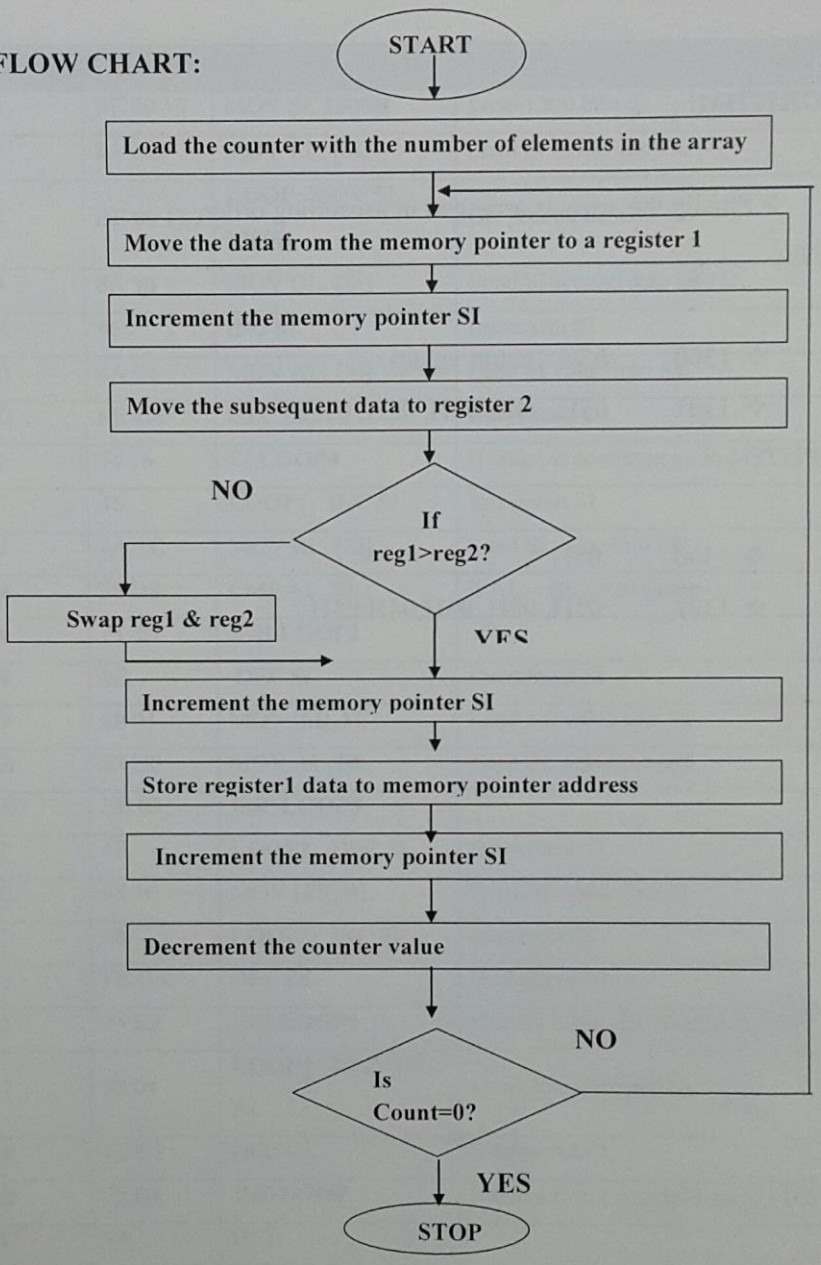
- 1300H 09H

PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	BE 00 12	MOV SI,**1200H**	Load 1200 address into SI
1103	8A 0C	MOV CL,[SI]	Load SI value into CL
1105	46	INC SI	Increment SI
1106	8A 04	MOV AL,[SI]	Move the first data in AL
1108	FE C9	DEC CL	Reduce the count
110A	46	**LOOP:** INC SI	Increment SI
110B	3A 04	CMP AL,[SI]	if AL> [SI] then go to jump1 (no swap)
110D	73 02	JNB **LOOP1**	If count is zero then jump into 1111
110F	8A 04	MOV AL,[SI]	Else store large no in to AL
1111	FE C9	**LOOP1**: DEC CL	Decrement the count
1113	75 F5	JNZ **LOOP**	If count is not zero then jump into 110A
1115	BF 00 13	MOV DI,**1300H**	Else store the biggest number at 1300
1118	88 05	MOV [DI],AL	Store the AL value into DI
111A	F4	HLT	End

6.13 ASCENDING ORDER:

FLOW CHART:



ALGORITHM:

- Finally the array is arranged in ascending order.

INPUT:

- 1300 05H (count value)
- 1301 05H, 04H, 03H, 02H, 01H

OUTPUT:

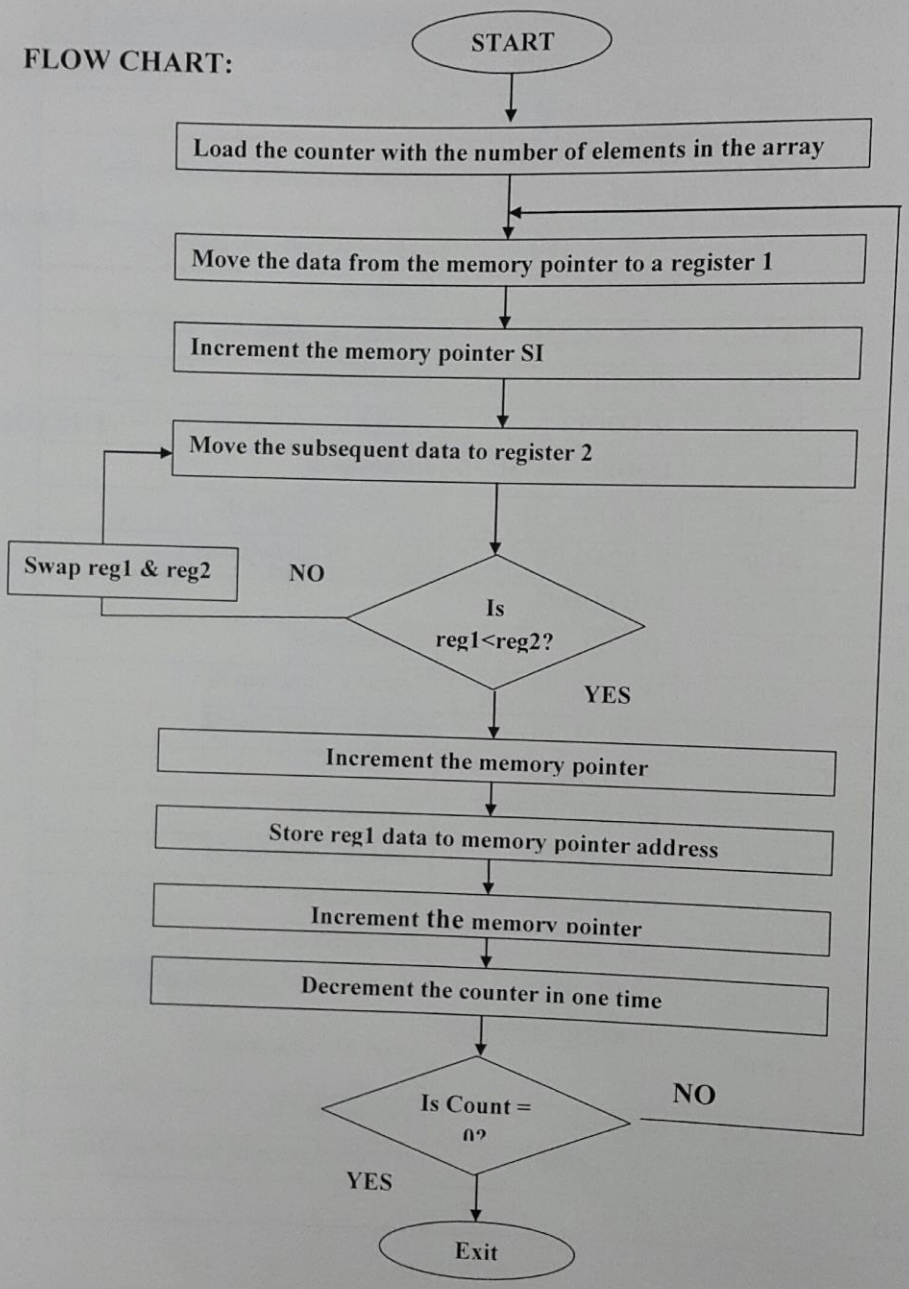
- 1300 05H
- 1301 01H, 02H, 03H,04H,05H

PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	BE 00 13	MOV SI, **1300H**	Load 1300 into SI
1103	8A 0C	MOV CL, [SI]	Load SI value into CL
1105	BE 00 13	**LOOP:** MOV SI, **1300H**	Get second data
1108	8A 14	MOV DL, [SI]	Load SI second data into DL
110A	46	INC SI	Increment SI
110B	8A 04	MOV AL, [SI]	Load SI value into AL
110D	FE CA	DEC DL	Decrement DL
110F	74 16	JZ **LOOP4**	If count is zero then go to 1127
1111	46	**LOOP1**: INC SI	Increment SI
1112	8A 1C	MOV BL, [SI]	Load SI value into BL
1114	38 D8	CMP AL, BL	if AL > BL go to (jump1)
1116	73 07	JNB **LOOP2**	
1118	4E	DEC SI	Decrement SI
1119	88 04	MOV [SI],AL	Load AL value into SI
111B	88 D8	MOV AL, BL	Load BL value into AL
111D	EB 03	JMP **LOOP3**	
111F	4E	**LOOP2:** DEC SI	Decrement SI
1120	88 1C	MOV [SI], BL	Load BL value into SI
1122	46	**LOOP3**: INC SI	Increment SI
1123	FE CA	DEC DL	Decrement DL
1125	75 EA	JNZ **LOOP1**	If count is not zero then go to 1111
1127	88 04	**LOOP4:** MOV [SI], AL	Load AL value into SI
1129	FE C9	DEC CL	Decrement CL
112B	75 D8	JNZ **LOOP**	If count is not zero then go to 1105
112D	F4	HLT	

6.14 DESCENDING ORDER

FLOW CHART:



ALGORITHM:

➢ Finally the array is arranged in Descending order.

INPUT:

- 1300 05H (count value)
- 1301 01H, 02H, 03H, 04H, 05H

OUTPUT:

- 1300 05H
- 1301 05H, 04H, 03H, 02H, 01H

PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	BE 00 13	MOV SI, **1300H**	Load 1300 into SI
1103	8A 0C	MOV CL, [SI]	Load SI value into CL
1105	BE 00 13	**LOOP:** MOV SI, **1300H**	Get second data
1108	8A 14	MOV DL, [SI]	Load SI second data into DL
110A	46	INC SI	Increment SI
110B	8A 04	MOV AL, [SI]	Load SI value into AL
110D	FE CA	DEC DL	Decrement DL
110F	74 16	JZ **LOOP4**	If count is zero then go to 1127
1111	46	**LOOP1**: INC SI	Increment SI
1112	8A 1C	MOV BL, [SI]	Load SI value into BL
1114	38 D8	CMP AL, BL	if AL > BL go to (jump1)
1116	72 07	JB **LOOP2**	
1118	4E	DEC SI	Decrement SI
1119	88 04	MOV [SI],AL	Load AL value into SI
111B	88 D8	MOV AL, BL	Load BL value into AL
111D	EB 03	JMP **LOOP3**	
111F	4E	**LOOP2:** DEC SI	Decrement SI
1120	88 1C	MOV [SI], BL	Load BL value into SI
1122	46	**LOOP3**: INC SI	Increment SI
1123	FE CA	DEC DL	Decrement DL
1125	75 EA	JNZ **LOOP1**	If count is not zero then go to 1111
1127	88 04	**LOOP4:** MOV [SI], AL	Load AL value into SI
1129	FE C9	DEC CL	Decrement CL
112B	75 D8	JNZ **LOOP**	If count is not zero then go to 1105
112D	F4	HLT	

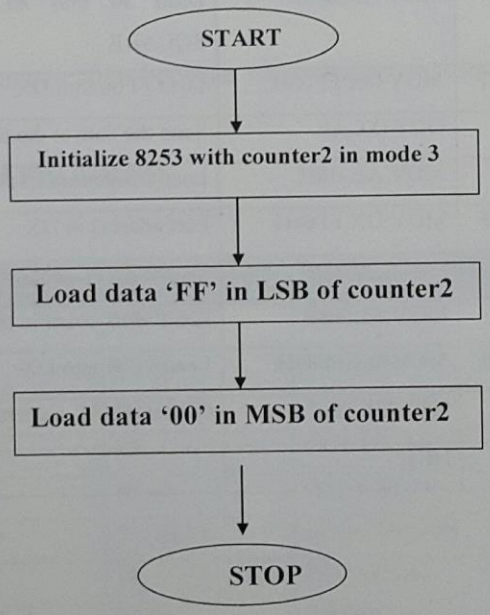
CHAPTER 7

ON BOARD INTRERFACE

7.1 TIMER INTRERFACE

SQUARE WAVE GENERATION USING 8253/54

FLOW CHART:



ALGORITHM:

- Initialize the Timer port J7 connecter result.
- Timer its channel 2 in mode3 operation

OUTPUT:

To generate a square wave of frequency 3 KHz at OUT2, CLK 0 of 8253 is connected to PCLK of frequency 3 MHz

PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
1100	B0 B6	MOV AL,B6H	Load 36 into AL for generating SQUARE
1102	BA 06 FF	MOV DX,FF06H	Load FF06 into DX
1105	EE	OUT DX,AL	Send the data to the timer
1106	B0 B8	MOV AL,B8H	Load LSB count in the AL
1108	BA 04 FF	MOV DX,FF04H	Port address in DX
110B	EE	OUT DX,AL	Output the AL contents to CLK 2
110C	B0 0B	MOV AL,0BH	Load MSB count in the AL
110E	BA 04 FF	MOV DX,FF04H	Load FF04 into DX
1111	EE	OUT DX,AL	Output the AL content to CLK 2
1112	F4	HLT	

CALCULATION

Square wave frequency $= \frac{\text{PCLK}}{\text{COUNT}}$

$= \frac{3.1\text{MHZ}}{3\text{KHZ}}$

$= 1.1$ KHz

$= 1.1$ KHz

7.2 SERIAL COMMUNICATION BETWEEN TWO MICROPROCESSORS

PROCEDURE

- Connect RS232 FEMALE to FEMALE (cross cable) between Two 8086 Microprocessor.
- Enter the program. First Execute Receiver program.
- After Execute transmitter program.
- Press reset switch on both the 8086 trainer kit.
- A set of characters are transmitted from one 8086 board to the other.

AIM

Serial communication between two microprocessors using 8086 trainer kit

TRANSMITTER PROGRAM

ADDRESS	OPCODE	MNEMONICS
1100	BE 00 12	MOV SI,1200
1103	B0 B6	MOV AL,B6
1105	BA 26 FF	MOV DX,FF26
1108	EE	OUT DX,AL
1109	B0 40	MOV AL,40
110B	BA 24 FF	MOV DX,FF24
110E	EE	OUT DX,AL
110F	B0 01	MOV AL,01
1111	BA 24 FF	MOV DX,FF24
1114	EE	OUT DX,AL
1115	B1 05	**RELOD** : MOV CL,05
1117	BA 12 FF	**CHECK** : MOV DX,FF12
111A	EC	IN AL,DX

111B	24 04	AND AL,04
111D	74 F8	JZ CHECK
111F	8A 04	MOV AL,[SI]
1121	BA 10 FF	MOV DX,FF10
1124	EE	OUT DX,AL
1125	46	INC SI
1126	3C 3F	CMP AL,3F
1128	75 EB	JNZ **RELOD**
112A	FE C9	DEC CL
112C	75 E9	JNZ **CHECK**
112E	CC	INT 03

NOTE:

The following characters are transmitted via Serial Port

1200 01H, 02H, 03H, 04H, 05H

RECEIVER PROGRAM:

ADDRESS	OPCODE	MNEMONICS
1100	BE 00 12	MOV SI,1200
1103	B0 B6	MOV AL,B6
1105	BA 26 FF	MOV DX,FF26
1108	EE	OUT DX,AL
1109	B0 40	MOV AL,40
110B	BA 24 FF	MOV DX,FF24
110E	EE	OUT DX,AL
110F	B0 01	MOV AL,01
1111	BA 24 FF	MOV DX,FF24
1114	EE	OUT DX,AL
1115	B1 05	**RELOD** : MOV CL,05

1117	BA 12 FF	**CHECK :** MOV DX,FF12
111A	EC	IN AL,DX
111B	24 02	AND AL,02
111D	74 F8	JZ CHECK
111F	BA 10 FF	MOV DX,FF10
1122	EC	IN AL,DX
1123	88 04	MOV [SI],AL
1125	46	INC SI
1126	3C 3F	CMP AL,3F
1128	75 EB	JNZ **RELOD**
112A	FE C9	DEC CL
112C	75 E9	JNZ **CHECK**
112E	CC	INT 03
112F	CC	INT 03

OUTPUT

Received characters are stored from 1200.

1200 01H, 02H, 03H, 04H, 05H

CHAPTER 8

ADDON INTERFACE CARDS

8.1 PS-ADDON 7 SEGMENT DISPLAY

In 8086 We have two no's of 8255 ic's

- GPIO- I (8255) **J8** Connector

PORTS	ADDRESS
Control port	FF26
PORT A	FF20
PORT B	FF22
PORT C	FF24

- GPIO- II (8255) **J6** Connector

PORTS	ADDRESS
Control port	FF36
PORT A	FF30
PORT B	FF32
PORT C	FF34

PROCEDURE:

- Connect a 20 Pin FRC Cable between 8086 Trainer Kits J6 (Third port) or J8 (First port) and 7 Segment Display Card.
- Connect USB/PS2 Keyboard on 8086 Trainer Kits JP1 port.
- Type and Execute the Program by using USB/PS2 Keyboard.
- The Segment will display the Numbers (0 to 9) continuously.

AIM:

Interface 7 Segment Display card with 8086 using 8255

PROGRAM:

ADDRESS	OPCODE	MNEMONICS
1100	B0 80	MOV AL, 80
1102	BA 36 FF	MOV DX, FF36
1105	EE	OUT DX, AL
1106	B0 7F	MOV AL, 7F
1108	BA 30 FF	MOV DX, FF30
110B	EE	OUT DX,AL
110C	EB 15 00	**AGN:** CALL **DLY**
110F	B1 0A	**START**: MOV CL,0A
1111	BE 00 12	MOV SI,1200
1114	8A 04	**REP**: MOV AL,[SI]
1116	BA 32 FF	MOV DX,FF32
1119	EE	OUT DX,AL
111A	E8 07 00	CALL **DLY**
111D	46	INC SI
111E	FE C9	DEC CL
1120	75 F2	JNZ **REP**
1122	EB EB	JMP **START**
1124	BA FF FF	**DLY**: MOV DX,FFFF
1127	4A	**M**: DEC DX
1128	75 FD	JNZ **M**
112A	C3	RET

digit selection.

Display data.

ORG 1300H

1200 C0H, CFH, A4H, B0H, 99H,

1205 92H, 82H, F8H, 80H, 98H

8.2 PS-ADDON LED & SWITCH

IN 8086 WE HAVE TWO 8255 IC'S

- **J8** Connector [GPIO- I (8255)]

PORTS	ADDRESS
Control port	FF26
PORT A	FF20
PORT B	FF22
PORT C	FF24

- **J6** Connector [GPIO- II (8255)]

PORTS	ADDRESS
Control port	FF36
PORT A	FF30
PORT B	FF32
PORT C	FF34

PROCEDURE:

- Connect a 20 Pin FRC Cable between 8086 Microprocessor J6 port(third port) or J8 port(first port) and the Led & Switch interface card.
- Connect USB/PS2 keyboard on 8051 Microcontroller JP2 port.
- Type and Execute the Program by using USB/Ps2 Keyboard.

- Now Turn ON/OFF the switches in LED & Switch Interface card and the corresponding output Led will be turned ON/OFF.

AIM:

Interface LED & Switch card with 8086 using 8255

PROGRAM:

ADDRESS	OPCODE	MNEMONICS
1100	B0 90	MOV AL,90
1102	BA 36 FF	MOV DX,FF36
1105	EE	OUT DX,AL
1106	BA 30 FF	~~LOP~~: MOV DX,FF30
1109	EC	IN AL,DX
110A	BA 32 FF	MOV DX,FF32
110D	EE	OUT DX,AL
110E	EB F6	JMP ~~LOP~~ 1106

Port or mode activation. I/O 8255
send control word.
Sw input.
LED out.

INPUT: give the input through Slide switch

OUTPUT: view the corresponding Digital output on LED

8.3 PS-ADDON KEYPAD

In 8086 We have Two no' of 8255

- U22- 8255 **&** U16-8255

PORTS	ADDRESS
CONTROL REG	FF26H
PORT C	FF24H

PROCEDURE:

- Connect a 20Pin FRC Cable between 8086 Microprocessor J9port (Middle port) and 4x4 Matrix keypad.
- Connect USB/PS2 Keyboard on 8086 Microprocessor JP1 port.
- Type and Execute the program. Press any key in 4x4 Matrix keyboard and the corresponding value will be displayed in 8086 Microprocessor board.
- The 8086 LCD displays output will be **A = 00.**(When SW1 in Matrix keyboard pressed)

PROGRAM

ADDRESS	OPCODE		MNEMONICS
1100	B0 88		MOV AL,88H
1102	BA 26 FF		MOV DX,0FF26H
1105	EE		OUT DX,AL
1106	B0 00	**START:**	MOV AL,00H
1108	BA 24 FF		MOV DX,0FF24H
110B	EE		OUT DX,AL
110C	EC		IN AL,DX
110D	24 0F		AND AL,0FH
110F	24 F0		AND AL,0F0H
1111	3C F0		CMP AL,0F0H
1113	72 02		JC **CONT**
1115	EB EF		JMP **START**
1117	E8 02 00	**CONT:**	CALL **DATA**
111A	EB EA		JMP **START**
111C	BE 00 12	**DATA:**	MOV SI,1200H
111F	BB 00 13		MOV BX,**1300H**
1122	B0 0E	FIRST ROW;	MOV AL,0EH
1124	E8 10 00		CALL **FIND1**
1127	B0 0D	SECOND ROW;	MOV AL,0DH
1129	E8 0B 00		CALL **FIND1**
112C	B0 0B	THIRD ROW	MOV AL,0BH
112E	E8 06 00		CALL **FIND1**
1131	B0 07	FOURTH ROW	MOV AL,07H
1133	E8 01 00		CALL **FIND1**
1136	C3		RET
1137	24 0F	**FIND1:**	AND AL,0FH
1139	BA 24 FF		MOV DX,0FF24H
113C	EE		OUT DX,AL
113D	EC		IN AL,DX
113E	24 F0		AND AL,0F0H
1140	D0 C8		ROR AL,01
1142	D0 C8		ROR AL,01
1144	D0 C8		ROR AL,01
1146	D0 C8		ROR AL,01
1148	8A C8		MOV CL,AL
114A	B2 04		MOV DL,04H
114C	8A 07	**FFN2:**	MOV AL,[BX]

114E	3A C1		CMP AL,CL
1150	75 04		JNZ **FFN3**
1152	8A 04		MOV AL,[SI]
1154	CC		INT 03H
1155	C3		RET
1156	46	**FFN3:**	INC SI
1157	43		INC BX
1158	FE CA		DEC DL
115A	75 F0		JNZ **FFN2**
115C	BB 00 13		MOV BX,**1300H**
115F	C3		RET

ORG 1200H

1200	00 01 02 03	DB 00H, 01H, 02H, 03H
1204	04 05 06 07	DB 04H, 05H, 06H, 07H
1208	08 09 0A 0B	DB 08H, 09H, 0AH, 0BH
120C	0C 0D 0E 0F	DB 0CH, 0DH, 0EH, 0FH

ORG 1300H

1300	0E 0D 0B 07	DB 0EH, 0DH, 0BH, 07H

8.4 PS-ADDON TRAFFIC LIGHT CONTROLLER

IN 8086 WE HAVE TWO 8255 IC'S

- GPIO- I (8255)

PORTS	ADDRESS
Control port	FF26
PORT A	FF20
PORT B	FF22
PORT C	FF24

- GPIO- II (8255)

PORTS	ADDRESS
Control port	FF36
PORT A	FF30
PORT B	FF32
PORT C	FF34

PROCEDURE:

- Connect a 20pin FRC cable between 8086 Microprocessor J6 port (Third port) or J8 port (First port) and the Traffic light controller.
- Program can be tested with GPIOII (8255) J6 or with GPIOI (8255) -J8.
- Type the Program by using USB/ PS2 Key board.
- Execute the Program. Now Traffic Light Led's are blinking.

MODEL TRAFFIC LIGHT CONTROLLER

AIM:

Interface Traffic light controller with 8086 using 8255

PROGRAM:

ADDRESS	OPCODE	MNEMONICS
1100	BB 00 12	**START:** MOV BX, 1200H
1103	B9 08 00	MOV CX, 0008H
1106	8A 07	MOV AL,[BX]
1108	BA 36 FF	MOV DX, FF36
110B	EE	OUT DX, AL
110C	43	INC BX
110D	8A 07	**NEXT**: MOV AL,[BX]
110F	BA 30 FF	MOV DX, FF30
1112	EE	OUT DX,AL
1113	43	INC BX
1114	8A 07	MOV AL,[BX]
1116	BA 32 FF	MOV DX,FF32
1119	EE	OUT DX,AL
111A	E8 06 00	CALL **DELAY**
111D	43	INC BX
111E	49	DEC CX
111F	75 ED	JNZ **NEXT**
1121	EB DD	JMP **START**
1123	51	**DELAY:** PUSH CX
1124	B9 0F 00	MOV CX,000F
1127	BA FF FF	**REPEAT:** MOV DX,FFFFH
112A	4A	**LOOP2:** DEC DX
112B	75 FD	JNZ **LOOP2**
112D	E2 F8	LOOP **REPEAT**
112F	59	POP CX
1130	C3	RET

ORG 1200H

1200	80	DB 80H
1201	21 09 31 09	DB 21H,09H,31H,09H
1205	0C 09 8C 09	DB 0CH,09H,8CH,09H
1209	64 08 64 0C	DB 64H,08H,64H,0CH
120D	24 03 26 03	DB 24H,03H,26H,03H

END

8.5 PS-ADDON 8251/8253 INTERFACE CARD

8251/8253 INTERFACE WITH 8086 TRAINER KIT

CONTROL REGISTER OF 8253(TIMER)

	ADDRESS
Control REG	FF26H
Channel 0	FF20H
Channel 1	FF22H
Channel 2	FF24H

CONTROL REGISTER OF 8251 (UART)

	ADDRESS
Control REG	FF10H
Control Data	FF12H

JUMPERS SETTINGS IN 8086 USING FOLLOWS

- Put the jumper in UP position. J3 in up position(clk0) and J10 MUST DOWN position for 8086 to select address and IO selection
- For interrupt terminal count program put the jumper (J3) in down position.

JUMPERS		8086
ADDRESS	**J6** (ADDR0)	**A1**
ADDRESS	**J7**(ADDR1)	**A2**
CLOCK	**J3**(CLK0)	**PCLK**
DEC6	**J5**(DEC6)	**NOT**
DEC4	**J4**(DEC4)	**M/IO**
CS TIMER	**J10**(TIMER)	**CS3**
CS UART	**J9**(USART)	**CS2**

8253 TIMER INTERFACE PROGRAM

INTERRUPT TERMINAL COUNT

AIM

To interface 8253 with 8086 using Mode 0 - Interrupt On Terminal Count:

PROGRAM:

ADDRESS	OPCODE	MNEMONICS
1100	B0 30	MOV AL,30
1102	BA 26 FF	MOV DX,FF26
1105	EE	OUT DX,AL
1106	B0 04	MOV AL,04
1108	BA 20 FF	MOV DX,FF20

110B	EE	OUT DX,AL
110C	B0 00	MOV AL,00
110E	BA 20 FF	MOV DX,FF20
1111	EE	OUT DX,AL
1112	EB FE	JMP 1112

PROCEDURE

- Place jumper J3 in SW2 position

OUTPUT

Initially in Channel 0 output is low. After giving five clock Pulses, you may notice that the output goes high (place the CRO probe point in J3 header in OUT 0).

RATE GENERATOR

AIM

To interface 8253 with 8086 using Mode 2 - Rate Generator:

PROGRAM

ADDRESS	OPCODE	MNEMONICS
1100	B0 74	MOV AL,74
1102	BA 26 FF	MOV DX,FF26
1105	EE	OUT DX,AL
1106	B0 0A	MOV AL,0A
1108	BA 22 FF	MOV DX,FF22
110B	EE	OUT DX,AL
110C	B0 00	MOV AL,00
110E	BA 22 FF	MOV DX,FF22
1111	EE	OUT DX,AL
1112	EB FE	JMP 1112

PROCEDURE

- Place jumper J3 in PCLK position

OUTPUT

Using mode 2 let us divide the clock present at channel 1 by 10. Connect the CLK 1 to PCLK. In a CRO observe simultaneously, the input clock to channel 1 and the output OUT 1 (place the CRO probe point in J8 header in OUT1).

SQURE WAVE GENERATOR

AIM

To interface 8253 with 8086 using mode 3 - square waves generator:

PROGRAM

ADDRESS	OPCODE	MNEMONICS
1100	B0 B6	MOV AL,B6
1102	BA 26 FF	MOV DX,FF26
1105	EE	OUT DX,AL
1106	B0 0B	MOV AL,0B
1108	BA 24 FF	MOV DX,FF24
110B	EE	OUT DX,AL
110C	B0 00	MOV AL,00
110E	BA 24 FF	MOV DX,FF24
1111	EE	OUT DX,AL
1112	F4	HLT

PROCEDURE

- Place jumper J3 in PCLK position

OUTPUT

To generate a square wave of frequency 270 KHz at OUT2, CLK 0 of 8253 is connected to PCLK of frequency 3 MHz

CALCULATION

Square wave frequency $= \frac{\text{PCLK}}{\text{COUNT}}$

$= \frac{3.1}{11}$

$= 0.280$ MHz

$= 280$ KHz

Where,

PCLK $\rightarrow$ 3.1 MHz, Count $\rightarrow$ 0x0B $[(11)_{10}]$

8251 UART INTERFACE PROGRAM

SERIAL COMMUNICATION BETWEEN TWO MICROPROCESSORS

PROCEDURE:

- Place a jumper in J3 between middle and PCLK in 8251 And 8253 interface card.
- Take One 8086 trainer kit And One 8251 & 8253 interface card. .
- Connect 50 Pin FRC Cable between 8086 and 8251 &8253 interface card.
- Connect RS232 FEMALE to FEMALE (cross) cable between 8251 & 8253 interface card and PC.
- Type and Enter the program. Open HyperTerminal at 9600 baud rate.
- The string "12345" will be displayed in HyperTerminal.

AIM:

Serial communication between 8251 & 8253 interface card and HyperTerminal.

TRANSMITTER PROGRAM

ADDRESS	OPCODE	LABEL	MNEMONICS
1100	BE 12 00		MOV SI,1200H
1103	B0 B6		MOV AL,0B6H
1105	BA FF 26		MOV DX,0FF26H
1108	EE		OUT DX,AL
1109	B0 40		MOV AL,040H
110B	BA FF 24		MOV DX,0FF24H
110E	EE		OUT DX,AL
110F	B0 01		MOV AL,01H
1111	BA FF 24		MOV DX,0FF24H
1114	EE		OUT DX,AL
1115	B0 00		MOV AL,00H
1117	E8 11 41		CALL SENDUART
111A	E8 11 41		CALL SENDUART
111D	E8 11 41		CALL SENDUART
1120	B0 40		MOV AL,040H
1122	E8 11 41		CALL SENDUART
1125	B0 4D		MOV AL,04DH
1127	E8 11 41		CALL SENDUART
112A	B0 05		MOV AL,05H
112C	E8 11 41		CALL SENDUART
112F	B3 05		MOV BL,05H
1131	BA FF 10		MOV DX,0FF10H
1134	8A 04	SEND CHAR:	MOV AL,[SI]
1136	EE		OUT DX,AL
1137	E8 11 45		CALL LOOP2
113A	46		INC SI
113B	FE CB		DEC BL
113D	75 F5		JNZ SENDCHAR
113F	EB FE	HERE:	JMP HERE
1141	BA FF 12	SEND UART:	MOV DX,0FF12H
1144	EE		OUT DX,AL
1145	B9 02 00	LOOP2:	MOV CX,200H
1148	E2 FE	DELAY1:	LOOP DELAY1
114A	C3		RET

INPUT:

1200 31H, 32H, 33H, 34H,35H

OUTPUT:

12345 (In HyperTerminal)

8.6 PS ADDON-8255 INTERFACE CARD

8255 INTERFACE WITH 8086 TRAINER KIT

	ADDRESS
Control REG	FF56H
PORT A	FF50H
PORT B	FF52H
PORT C	FF54H

AIM

To initialize Port A as an input port in mode – 0

PROGRAM

ADDRESS	OPCODE	MNEMONICS
1100	BE 00 12	MOV SI, 1200H
1103	B0 90	MOV AL, 90
1105	BA56 FF	MOV DX,FF56
1108	EE	OUT DX,AL
1109	BA 50 FF	MOV DX,FF50
110C	EC	**LOOP1:** IN AL,DX
110D	88 04	MOV [SI],AL
110F	EB FB	JMP **LOOP1**

PROCEDURE

- Connect 50 Pin FRC Cable 8086 Microprocessor J7 port (50 pin box connector) and PPI 8255 Interface Card.
- Connect USB/PS2 Keyboard on 8086 Microprocessor Jp1 port.
- Enter the program starting from the user Ram address 1100H. Set a known data at the SPDT switches. Execute the program.
- The above program initializes port A as an input port. The data as set by the SPDT switch settings is input into the accumulator and is stored at the location 1200H.
- Please verify whether the data in LCD at 1200 Location is the same as that set by SPDT switches.

AIM

To initialize Port A as input port and port B as output port in mode - 0

PROGRAM

ADDRESS	OPCODE	MNEMONICS
1100	B0 90	MOV AL,90
1102	BA 56 FF	MOV DX,FF56
1105	EE	OUT DX,AL
1106	BA 50 FF	**L1:** MOV DX,FF50
1109	EC	IN AL,DX
110A	BA 52 FF	MOV DX,FF52
110D	EE	OUT DX,AL
111D	EB F6	JMP **L1**

PROCEDURE

- Connect 50 Pin FRC Cable to the 8086 Microprocessor J7 port (50 pin box connector) and other end connect to the PPI 8255 Interface Card.
- Connect USB/PS2 Keyboard on 8086 Microprocessor Jp1 port.
- Enter the program starting from the user RAM address 1100H. Set a known data at the SPDT switches. Execute the program.
- The above program initializes port A as an input port and port B as output port. The data as set by the SPDT switch settings is input to the accumulator and is outputted to port B.
- Please verify visually that the data output at the LEDs is the same as that set by the SPDT switch settings

AIM

To initialize port C as output port in mode – 0

PROGRAM

ADDRESS	OPCODE	MNEMONICS
1100	B0 90	MOV AL,90
1102	BA 56 FF	MOV DX,FF56
1105	EE	OUT DX,AL
1106	B0 01	MOV AL,01
1108	BA 54 FF	MOV DX,FF54
110B	EE	OUT DX,AL
110C	F4	HLT

PROCEDURE

- 50 Pin FRC Cable connect to the 8086 Microprocessor J7 port (50 pin box connector) and other end connect to the PPI 8255 Interface Card.

- Connect USB/PS2 Keyboard on 8086 Microprocessor Jp1 port.

- Enter the program starting from the user RAM address 9100H. Execute the program. In the design used in our board, PC0, PC1, PC2, PC3 bits of Port C are connected to LEDs. So a data output glows the corresponding LED.

- In the above program, we output a "one" to the PC0 bit of port C. It can be visually checked by the LED display that the PC0 bit goes 'high' whereas the other bits of port C remain 'low'.

8.7 PS ADDON-8259 INTERFACE CARD

8259 INTERFACE WITH 8086

INTERRUPT PROGARM

AIM:

This program do Addition Operation If IRQ0 is raised.

MAIN PROGRAM:

ADDRESS	OPCODE	MNEMONICS
1100	BA 50 FF	MOV DX,FF50
1103	B0 1F	MOV AL,1F ; **ICW1**
1105	EE	OUT DX,AL
1106	BA 52 FF	MOV DX,FF52
1109	B0 10	MOV AL,10 ; **ICW2**
110B	EE	OUT DX,AL
110C	BA 52 FF	MOV DX,FF52
110F	B0 03	MOV AL,03 ; **ICW4**
1111	EE	OUT DX,AL
1112	BA 52 FF	MOV DX,FF52
1115	B0 80	MOV AL,80 ; **OCW1**
1117	EE	OUT DX,AL
1118	BA 50 FF	MOV DX,FF50
111B	FB	STI
111C	**HERE:**	JMP **HERE**

INTERRUPT VECTOR:

0040 Address for IRQ0 enable subroutine program address 1200H

0040	00
0041	12
0042	00
0043	00

INTERRUPT SERVICE ROUTINE:

1200	B8 13 00	MOV AX,0013H
1203	BB 14 00	MOV BX,0014H
1206	01 D8	ADD AX,BX
1208	CC	INT 03

JUMPER SETTINGS:

MODE (JP1)	CHIP SELECT (JP2)	SP/EN (JP3)	INTR (JP4)	INTA (JP5)
8086	86/51	8085/86	8085/86	8085/86

PROCEDURE:

- Set all the jumpers in 8086 Mode (JP1-JP5) in 8259 Interface Card..
- Connect a 50Pin FRC Cable between 8086 Trainer Kit J7 Port and 8259 Interface card.
- Connect USB/PS2 keyboard on 8086 Microprocessor Kit. Type and execute the program.
- Turn ON the IRQ0 (SW1) Switch, Now the Kit is waiting for the interrupt Request. Press the PULSE Switch (S1) to Create the Interrupt.
- Now the Program Control goes to Interrupt Service Routine. The Result is Stored in Accumulator.

OUTPUT AX=0027H

8.8 PS ADDON-8279 INTERFACE CARD

8086 MICRO PROCESSOR ON BOARD I/O DECODING ADDRESS

CONTROL	ADDRESS
Control REG	FF52H
Control Data	FF00H

READ A KEY

AIM

To read a key and store the key code in memory location 1200. This code will be entered into the FIFO whenever a key is pressed the value stored in 1200 location (4x4 matrix method

PROCEDURE

- Connect a 50 Pin FRC Cable between 8051 Trainer Kit and the 8279 Keyboard/Display Interface card.
- Connect USB/PS2 Keyboard on 8086 MicroprocessorJP1 port.
- Type and Execute the Program by using USB/PS2 Keyboard.
- Now press any key on 8279 keyboard/ display interface card the equal key value read and stored in 9200H memory location in 4x4 matrix method.

PROGRAM

ADDRESS	OPCODE	MNEMONICS
1100	BB 00 12	MOV SI,1200
1103	BA 50 FF	MOV DX,FF50
1106	EC	IN AL,DX
1107	88 04	MOV [SI],AL
1109	F4	HLT

ROLLING DISPLAY

AIM

To display the rolling message 'HELLO' in the display.

PROCEDURE:

- Connect a 50 Pin FRC Cable between 8051 Trainer Kit and the 8279 Keyboard/Display Interface card.
- Connect USB/PS2 Keyboard on 8086 Microprocessor JP1 port.
- Type and Execute the Program by using USB/PS2 Keyboard.
- Now the Rolling Message "HELLO" would be display on 8279 keyboard/ display interface card.

PROGRAM

ADDRESS	OPCODE	MNEMONICS
1100	BE 00 12	**START**:MOV SI,1200
1103	B9 0F 00	MOV CX,000F
1106	B0 10	MOV AL,10
1108	BA 52 FF	MOV DX,FF52
110B	EE	OUT DX,AL
110C	B0 CC	MOV AL,CC
110E	EE	OUT DX,AL
110F	B0 90	MOV AL,90
1111	EE	OUT DX,AL
1112	8A 04	**LOP1:** MOV AL,[SI]
1114	BA 50 FF	MOV DX,FF50
1117	EE	OUT DX,AL
1118	E8 05 01	CALL **DELAY**
111B	46	INC SI
111C	E2 F4	LOOP **LOP1**
111E	EB E0	JMP **START**
1120	BA FF A0	**DELAY**:MOV DX,A0FF
1123	4A	DEC DX
1124	75 FD	JNZ 1123
1126	C3	RET

LOOK-UP TABLE:-

1200 FFH FFH FFH FFH

1204 FFH FFH FFH FFH

1208 98H 68H 7CH 7CH

120C 0CH FFH FFH FFH

8.9 PS-ADDON ADC 0809 INTERFACE CARD

In 8086 We have two no's of 8255 ic's

- GPIO- I (8255)

PORTS	ADDRESS
Control port	FF26
PORT A	FF20
PORT B	FF22
PORT C	FF24

- GPIO- II (8255)

PORTS	ADDRESS
Control port	FF36
PORT A	FF30
PORT B	FF32
PORT C	FF34

ANALOG TO DIGITAL CONVERSION

AIM:

To Interface ADC 0809 with 8086 Trainer Kit and to capture the data at channel - 0 **and displays the output with the LEDs.**

PROGRAM:

ADDRESS	OPCODE	MNEMONICS
1100	B0 90	MOV AL,90
1102	BA 36 FF	MOV DX,FF36
1105	EE	OUT DX,AL
1106	B0 01	MOV AL,01
1108	BA 32 FF	MOV DX,FF32
110B	EE	OUT DX,AL
110C	B0 01	MOV AL,01
110E	BA 32 FF	MOV DX,FF32
1111	EE	OUT DX,AL
1112	B0 31	MOV AL,31
1114	BA 32 FF	MOV DX,FF32
1117	EE	OUT DX,AL
1118	B0 01	**LOOP**: MOV AL,01
111A	BA 32 FF	MOV DX,FF32
111D	EE	OUT DX,AL
111E	24 01	AND AL,01
1120	74 F6	JZ **LOOP**
1122	B0 40	MOV AL,40
1124	BA 32 FF	MOV DX,FF32
1127	EE	OUT DX,AL
1128	E8 05 00	CALL **DELAY**

112B	BA 30 FF	MOV DX,FF30
112E	EC	IN AL,DX
112F	CC	INT 03
1130	BA FF FF	**DELAY**: MOV DX,FFFF
1133	4A	**LOOP1**: DEC DX
1134	75 FD	JNZ **LOOP1**
1136	C3	RET

PROCEDURE:

- Connect a 20 Pin FRC cable between 8086 Trainer Kits J6 port (third port) or J9 port (First port) and the ADC Interface card.
- Place jumper JP2 of ADC Interface Card in 'I' position.
- Connect USB/PS2 keyboard on 8086 Microprocessor. Type and execute the program.
- Vary the analog input (Trim pot) and view the corresponding digital value in the LED (D0 –D7) and the corresponding hex value will be displayed in Trainer Kits LCD.

8.10 PS-ADDON DAC 0800 INTERFACE CARD

In 8086 We have two no's of 8255 ic's

- GPIO- I (8255)

PORTS	ADDRESS
Control port	FF26
PORT A	FF20
PORT B	FF22
PORT C	FF24

- GPIO- II (8255)

PORTS	ADDRESS
Control port	FF36
PORT A	FF30
PORT B	FF32
PORT C	FF34

PROCEDURE:

- Connect a 20 Pin FRC Cable between 8086 Microprocessor J6 port (Third port) or J8 port(First port) and the DAC interface card.
- Connect the 15v DC adapter with DAC interfacing board.
- Place the jumper J2 in down position.
- Verify the square wave output using CRO. The CRO Probe must be connected between BI polar (positive) and GND pin of J3 Header

DIGITAL TO ANALOG CONVERSION

AIM:

To create a square waveform at BI polar in 8086 Microprocessor

PROGRAM:

ADDRESS	OPCODE	MNEMONICS
1100	B0 80	MOV AL, 80H
1102	BA 36 FF	MOV DX,FF36
1105	EE	OUT DX, AL
1106	B0 FF	**START** : MOV AL,FF
1108	BA 30 FF	MOV DX, FF30
110B	EE	OUT DX,AL
110C	E8 08 00	CALL **DELAY**
110F	B0 00	MOV AL,00
1111	EE	OUT DX,AL
1112	E8 02 00	CALL **DELAY**
1115	EB EF	JMP **START**
1117	B3 05	**DELAY:** MOV BL,05
1119	B1 FF	**LOOP1:** MOV CL,FF
111B	FE C9	**LOOP2** : DEC CL
111D	75 FC	JNZ **LOOP2**

111F	FE CB	DEC BL
1121	75 F6	JNZ **LOOP1**
1123	C3	RET

AIM:

To Create a Saw-Tooth Wave at BI polar In 8086 Micro processor

PROGRAM:

ADDRESS	OPCODE	MNEMONICS
1100	B0 80	MOV AL, 80H
1102	BA 36 FF	MOV DX,FF36
1105	EE	OUT DX, AL
1106	B0 00	**START:** MOV AL,00
1108	BA 30 FF	MOV DX, FF30
110B	EE	**LOOP1 :** OUT DX,AL
110C	FE C0	INC AL
110E	75 FB	JNZ **LOOP1**
1110	EB F4	JMP **START**

PROCEDURE:

- Connect a 20 Pin FRC Cable between 8086 Microprocessor J6 port (Third port) or J8 port(First port) and the DAC interface card.
- Connect the 15v DC adapter with DAC interfacing board.
- Place the jumper J2 in down position.
- Verify the sawtooth wave output using CRO. The CRO Probe must be connected between BI polar (positive) and GND pin of J3 Header.

8.11 PS ADDON-I2C 7 SEGMENT INTERFACE CARD

1.1 I2C-7 SEGMENT INTRFACE WITH 8086 EXAMPLE PROGRAM

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
1100	B0 80		MOV AL,80H	8255 CONTROL WORD
1102	BA 26 FF		MOV DX,FF26H	
1105	EE		OUT DX,AL	
1106	E8 2B 00		CALL **I2CINIT**	
1109	E8 2F 00		CALL **STARTC**	
110C	B3 70		MOV BL,70H	SEG SLAVE ADDRESS
110E	E8 41 00		CALL **SEND**	
1111	B3 00		MOV BL,00H	SEG START BIT
1113	E8 3C 00		CALL **SEND**	
1116	B3 27		MOV BL,27H	SEG CONTROL WORD
1118	E8 37 00		CALL **SEND**	
111B	B3 4F		MOV BL,4FH	0 DATA SEND
111D	E8 32 00		CALL **SEND**	
1120	B3 5B		MOV BL,5BH	1 DATA SEND
1122	E8 2D 00		CALL **SEND**	
1125	B3 06		MOV BL,06H	2 DATA SEND
1127	E8 28 00		CALL **SEND**	
112A	B3 3F		MOV BL,3FH	3DATA SEND
112C	E8 23 00		CALL **SEND**	
112F	E8 13 00		CALL **STOP**	
1132	EB FE	**HERE:**	JMP **HERE**	

1134	B0 03	**I2CINIT:**	MOV AL,03	SDA SET
1136	EE		OUT DX,AL	
1137	B0 01		MOV AL,01	SCL SET
1139	EE		OUT DX,AL	
113A	C3		RET	

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
113B	B0 01	**START:**	MOV AL,01	SCL SET
113D	EE		OUT DX,AL	
113E	B0 02		MOV AL,02	SDA CLR
1140	EE		OUT DX,AL	
1141	B0 00		MOV AL,00	SCL CLR
1143	EE		OUT DX,AL	
1144	C3		RET	
1145	B0 00	**STOP:**	MOV AL,00	SCL CLR
1147	EE		OUT DX,AL	
1148	B0 02		MOV AL,02	SDA CLR
114A	EE		OUT DX,AL	
114B	B0 01		MOV AL,01	SCL SET
114D	EE		OUT DX,AL	
114E	B0 03		MOV AL,03	SDA SET
1150	EE		OUT DX,AL	
1151	C3		RET	

1152	B1 08	**SEND:**	MOV CL,08H	
1154	B0 00	**BACK:**	MOV AL,00H	SCL CLR
1156	EE		OUT DX,AL	
1157	D0 D3		RCL BL,01	
1159	72 05		JC **LOOPCRY**	
115B	B0 02		MOV AL,02H	SDA CLR
115D	EB 03 90		JMP **SENDBIT**	
1160	B0 03	**L00PCRY:**	MOV AL,03H	SDA SET
1162	EE	**SENDBIT:**	OUT DX,AL	
1163	B0 01		MOV AL,01H	SCL SET
1165	EE		OUT DX,AL	
1166	FE C9		DEC CL	
1168	75 EA		JNZ **BACK**	

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
116A	B0 00		MOV AL,00H	SCL CLR
116C	EE		OUT DX,AL	
116D	B0 03		MOV AL,03H	SDA SET
116F	EE		OUT DX,AL	
1170	B0 01		MOV AL,01H	SCL SET
1172	EE		OUT DX,AL	
1173	B0 00		MOV AL,00H	SCL CLR
1175	EE		OUT DX,AL	

1176	C3		RET	

PROCEDURE:

- Connect a 20Pin FRC Cable between 8086 Trainer Kit J9 Port and I2C Peripheral Interface Card.
- Turn ON DIP Switch SW2 pin 7SEG (I2C Add-on Card).
- Enable I2C Data (SDA0) and Clock (SCL0) pins of SW1 (I2C Add-on Card).
- Connect USB/PS2 Keyboard on 8086 Microprocessor Kit. Type and Execute the Program
- Now 7 Segment Displays the value **0 1 2 3.**

8.12 PS ADDON STEPPER/DC MOTOR INTERFACE CARD

- In 8086 We have two no's of 8255 ic's
- GPIO- I (8255)

PORTS	ADDRESS
Control port	FF26
PORT A	FF20
PORT B	FF22
PORT C	FF24

- GPIO- II (8255)

PORTS	ADDRESS
Control port	FF36
PORT A	FF30
PORT B	FF32
PORT C	FF34

PROCEDURE:

- Connect a 20 Pin FRC cable between 8086 Microprocessor J6 port (Third port) or J9 port (First port) and the DC MOTOR/STEPPER MOTOR CARD.
- Connect a DC motor at the MG1 connector or connect a Stepper Motor in J4.

- Connect USB/PS2 keyboard on 8086. Type and execute the DC Motor or Stepper Motor program.
- Now the DC Motor or the Stepper Motor is running.

DC MOTOR INTERFACE WITH 8086

AIM:

To interface DC motor with 8086 and to run the DC motor in forward direction

PROGRAM:

ADDRESS	OPCODE	MNEMONICS
1100	B0 80	MOV AL,80
1102	BA 36 FF	MOV DX,FF36
1105	EE	OUT DX,AL
1106	B0 05	**START:** MOV AL,05
1108	BA 32 FF	MOV DX,FF32
110B	EE	OUT DX,AL
110C	E8 02 00	CALL **DELAY**
110F	EB F5	JMP **START**
1111	B9 FF FF	**DELAY**: MOV CX,FFFF
1114	49	**LOOP:** DEC CX
1115	75 FD	JNZ **LOOP**
1117	C3	RET

RESULT:

Execute the program. Now you can see that the DC motor runs in forward direction with a delay.

STEPPER MOTOR INTERFACE WITH 8086

AIM:

To interface the stepper motor with 8086 trainer kit and to run a stepper motor at different speed in clockwise directions.

PROGRAM:

ADDRESS	OPCODE	MNEMONICS
1100	B0 80	MOV AL,80
1102	BA 36 FF	MOV DX,FF36
1105	EE	OUT DX,AL
1106	BE 00 12	**START :** MOV SI,1200
1109	B3 04	MOV BL,04
110B	8A 04	**REPEAT :** MOV AL,[SI]
110D	BA 30 FF	MOV DX,FF30
1110	EE	OUT DX,AL
1111	E8 07 00	CALL **DELAY**
1114	46	INC SI
1115	FE CB	DEC BL
1117	75 F2	JNZ **REPEAT**
1119	EB EB	JMP **START**
111B	B9 03 09	**DELAY**: MOV CX,0903

111E	49	**LOOP :** DEC CX
111F	75 FD	JNZ **LOOP**
1121	C3	RET

ORG 1200H

1200 03060C09 DB 03H, 06H, 0CH, 09H

END

RESULT:

Execute the program. Now we can see that the stepper motor runs in forward direction with a delay.

RERUIRED ANGLE IN STEPPER MOTOR PROGRAM IN 8086

AIM:

To interface the stepper motor with 8086 trainer kit and to run a stepper motor for required angle within 360°

PROGRAM:

ADDRESS	OPCODE	MNEMONICS
1100	B0 80	MOV AL,80
1102	BA 36 FF	MOV DX,FF36
1105	EE	OUT DX,AL
1106	B1 19	**START :** MOV CL,19

1108	BE 00 12	**CLKWI :** MOV SI,1200
110B	E8 09 00	CALL **ROTAT**
110E	FE C9	DEC CL
1110	75 F6	JNZ **CLKWI**
1112	E8 16 00	CALL **HLT**
1115	EB EF	JMP **START**
1117	B3 04	**ROTAT :** MOV BL,04
1119	8A 04	**REPEAT :** MOV AL,[SI]
111B	BA 30 FF	MOV DX,FF30
111E	EE	OUT DX,AL
111F	BA 03 09	MOV DX,0903
1122	4A	**LOOP :** DEC DX
1123	75 FD	JNZ **LOOP**
1125	46	INC SI
1126	FE CB	DEC BL
1128	75 EF	JNZ **REPEAT**
112A	C3	RET
112B	F4	**HLT:** HLT

ORG 1200H

1200 03060C09 DB 03H, 06H, 0CH, 09H

END

RESULT:

Execute the program. Now we can see that the Stepper motor runs in forward direction with required angle.

CLOCK WISE ANTI CLOCK WISE ROTATE PROGRAM IN 8086

AIM:

To interface the stepper motor with 8086 trainer kit and to run a stepper motor in both the direction

PROGRAM:

ADDRESS	OPCODE	MNEMONICS
1100	B0 80	MOV AL,80
1102	BA 36 FF	MOV DX,FF36
1105	EE	OUT DX,AL
1106	B1 32	**START :** MOV CL,32
1108	BE 00 12	**CLKWI :** MOV SI,1200
110B	E8 18 00	CALL **ROTAT**
110E	FE C9	DEC CL
1110	75 F6	JNZ **CLKWI**

1112	E8 25 00	CALL **DELAY**
1115	B1 32	MOV CL,32
1117	BE 50 12	**ANCKWI :** MOV SI,1250
111A	E8 09 00	CALL **ROTAT**
111D	FE C9	DEC CL
111F	75 F6	JNZ **ANCKWI**
1121	E8 16 00	CALL **DELAY**
1124	EB E0	JMP **START**
1126	B3 04	**ROTAT :**MOV BL,04
1128	8A 04	**REPEAT :** MOV AL,[SI]
112A	BA 30 FF	MOV DX,FF30
112D	EE	OUT DX,AL
112E	BA 10 10	MOV DX,1010
1131	4A	**LOOP :**DEC DX
1132	75 FD	JNZ **LOOP**
1134	46	INC SI
1135	FE CB	DEC BL
1137	75 EF	JNZ **REPEAT**
1139	C3	RET

113A	BA FF FF	**DELAY :** MOV DX,FFFF
113D	4A	**LOOP1 :** DEC DX
113E	75 FD	JNZ LOOP1
1140	C3	RET

ORG 1200H

1200 03060C09 DB 03H, 06H, 0CH, 09H

ORG 1250H

1250 090C0603 DB 09H, 0CH, 06H, 03H

END

RESULT:

Execute the program. Now we can see that the Stepper motor runs in clockwise and anti clockwise direction.

8.13 PS ADDON LCD/GLCD INTERFACE CARD

LCD INTERFACING WITH 8086 TRAINER KIT

- In 8086 We have two no's of 8255 ic's
  - GPIO- I (8255)

PORTS	ADDRESS
Control port	FF26
PORT A	FF20
PORT B	FF22
PORT C	FF24

  - GPIO- II (8255)

PORTS	ADDRESS
Control port	FF36
PORT A	FF30
PORT B	FF32
PORT C	FF34

20 FRC cable connect to the J6 port (third port)

Program using connector GPIO II(8255)

AIM

To Interface LCD with 8086 Trainer Kit and to display the string on LCD "PANTECH"

PROGRAM

ADDRESS	OPCODE	MNEMONICS
1100	B0 80	MOV AL,80
1102	BA 36 FF	MOV DX,FF36
1105	EE	OUT DX,AL
1106	B0 38	MOV AL,38
1108	E8 33 00	CALL **CMDWT**
110B	E8 52 00	CALL **DELAY**
110E	B0 0E	MOV AL,0E
1110	E8 2B 00	CALL **CMDWT**
1113	E8 4A 00	CALL **DELAY**
1116	B0 06	MOV AL,06
1118	E8 23 00	CALL **CMDWT**
111B	E8 42 00	CALL **DELAY**
111E	B0 01	MOV AL,01
1120	E8 1B 00	CALL **CMDWT**
1123	E8 3A 00	CALL **DELAY**
1126	B1 0F	**START :** MOV CL,0F
1128	BE 00 12	MOV SI,1200
112B	8A 04	**REPET:** MOV AL,[SI]
112D	E8 1F 00	CALL **DATWT**
1130	E8 2D 00	CALL **DELAY**
1133	46	INC SI
1134	FE C9	DEC CL
1136	75 F3	JNZ **REPET**
1138 113B	E8 02 00 EB E9	CALL **HLT** JMP **START**
113D	F4	**HLT**
113E	BA 32 FF	**CMDWT:** MOV DX,FF32

1141	EE	OUT DX,AL
1142	B0 04	MOV AL,04
1144	BA 30 FF	MOV DX,FF30
1147	EE	OUT DX,AL
1148	B0 00	MOV AL,00
114A	BA 30 FF	MOV DX,FF30
114D	EE	OUT DX,AL
114E	C3	RET
114F	BA 32 FF	**DATWT:** MOV DX,FF32
1152	EE	OUT DX,AL
1153	B0 05	MOV AL,05
1155	BA 30 FF	MOV DX,FF30
1158	EE	OUT DX,AL
1159	B0 01	MOV AL,01
115B	BA 30 FF	MOV DX,FF30
115E	EE	OUT DX,AL
115F	C3	RET
1160	BA FF FF	**DELAY:** MOV DX,FFFF
1163	4A	**LOP :** DEC DX
1164	75 FD	JNZ **LOP**
1166	C3	RET

ORG 9200H

9200 20202022 DB 20H, 20H, 20H,22H ("PANTECH")

9204 50414E54 DB 50H, 41H, 4EH,54H

9208 45434822 DB 45H, 43H, 48H,22H

920C 20202020 DB 20H, 20H, 20H,20H

END