MICROPROCESSORS AND APPLICATIONS

LABORATORY MANUAL (ECE - 317)

III/IV ECE SEM - I

By
Mr.N.Srinivasa Naidu

Dr. V. Rajya Lakshmi
Professor & HOD, ECE

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY & SCIENCES (A)
(Affiliated to AU, Approved by AICTE & Accredited by NBA) Sangivalasa-
531 162, Visakhapatnam District, Phone: 08933-225083/84/87

Vision of the Institute

ANITS envisions to emerge as a world-class technical institution whose products represent a good blend of technological excellence and the best of human values.

Mission of the Institute

To train young men and women into competent and confident engineers with excellent communication skills, to face the challenges of future technology changes, by imparting holistic technical education using the best of infrastructure, outstanding technical and teaching expertise and an exemplary work culture, besides molding them into good citizens

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Vision of the Department

To become a centre of excellence in Education, research and produce high quality engineers in the field of Electronics and Communication Engineering to face the challenges of future technological changes.

Mission of the Department

To achieve vision department will

Transform students into valuable resources for industry and society by imparting contemporary technical education.

Develop interpersonal skills and leadership qualities among students by creating an ambience of academic integrity to participate in various professional activities

Create a suitable academic environment to promote research attitude among students.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates excel in their career in the domains of Electronics, Communication and Information Technology

PEO2: Graduates will practice professional ethics and excel in professional career through interpersonal skills and leadership qualities

PEO3: Graduates demonstrate passion for competence in higher education, research and participate in various professional activities

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

PO-1 **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization for the solution of complex engineering problems.

PO-2 **Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO-3 **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs.

PO-4 **Conduct investigations of complex problems:** An ability to design and conduct scientific and engineering experiments, as well as to analyze and interpret data to provide valid conclusions

PO-5 **Modern tool usage:** Ability to apply appropriate techniques, modern engineering and IT tools, to engineering problems.

PO-6 **The engineer and society:** An ability to apply reasoning to assess societal, safety, health and cultural issues and the consequent responsibilities relevant to the professional engineering practice

PO-7 **Environment and sustainability:** An ability to understand the impact of professional engineering solutions in societal and environmental contexts

PO-8 **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO-9 **Individual and team work:** Ability to function effectively as an individual, and as a member or leader in a team, and in multidisciplinary tasks.

PO-10 **Communication:** Ability to communicate effectively on engineering activities with the engineering community such as, being able to comprehend and write effective reports and design documentation, make effective presentations.

PO-11 **Project management and finance:** An ability to apply knowledge, skills, tools, and techniques to project activities to meet the project requirements with the aim of managing project resources properly and achieving the project's objectives.

PO-12 **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1 : Implement Signal & Image Processing techniques using modern tools.

PSO2 : Design and analyze Communication systems using emerging techniques.

PSO3 : Solve real time problems with expertise in Embedded Systems.

MICROPROCESSORS & APLLICATIONS LABORATORY	
ECE 317	Credits:2
Instruction: 3 Practical /Week	Sessional Marks:50
End Exam: 3 Hours	End Exam Marks:50

COURSE OUTCOMES:

At the end of the course, students will be able to

CO1: Program 8085 & 8086 microprocessor to meet the requirements of the user.

CO2: Interface peripherals like switches, LEDs, stepper motor, Traffic lights controller, etc..,

CO3: Apply concept & types of interrupts for the given context.

CO4: Design a microcomputer to meet the requirement of the user

CO-PO-PSO Mapping

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO 1	2	2	2	2	2			1	1	1					2
CO 2	3	2	2	2	2			1	1	1					2
CO 3	2	2	2	2	2			1	1	1					2
CO 4	2	2	3	3	2			1	1	1					3

3: high correlation, 2: medium correlation, 1: low correlation

List of Experiments:

Note: A student has to perform a minimum of 10 experiments.

Scheme of Evaluation

Lab Internal:

I. Observation – 5M
(Successful Wording/Algorithm/flowchart-1M, Successful Program verification – 1M, Successful Program Execution – 1M, Record Initial and Indexing – 2M)

II. Record – 10M
(Aim &Apparatus – 1M, Theory – 3M, Algorithm/flowchart – 2M(each experiment should have at least one flowchart), Hand Coding(Program Table) – 2M, Procedure & Precautions – 1M, Calculations, Input/Output observations & Result – 1M)

III. Daily Performance – 10M
(Based on regularity (A Student who fails to submit record in stipulated amount of time will attract a penalty), response, behavior, discipline,..etc award 1M for each experiment)

IV. Attendance – 5M

V. Internal End Exam – 20M
(Aim, Apparatus – 2M, Program – 10M (Mnemonics/code – 5M, Hand coding & relevant Comments – 2M, Algorithm/flow chart – 3M), Calculations, Input/Output observations & Result – 5M, Performance – 3M)

Lab External:

I. Writeup – 10M
(Aim– 2M, Apparatus – 1M, Theory – 2M, Algorithm/flowchart – 5M)

II. Program – 15M
(Mnemonics/Code – 10M, Comments – 3M, hex Codes – 2M)

III. Performance – 5M
(Experimentation skill – Connections.. etc.)

IV. Result – 10M
(Identifying & Showing the inputs and outputs – 2M and/or theoretical calculations – 2M, Output Verification – 6M (Partial output – 3M, No Output – 0M)

V. Viva – 10M

RUBRICS

(MP&A LABORATORY)

S.No	Competency	Performance Indicator
1	Demonstrate an ability to conduct experiments consistent with their level of knowledge and understanding.	Laboratory preparation (verification of Lab observation)
		Stating clearly the aim of the experiment, its scope and importance for purpose of doing experiment.(Based on viva)
		Experimental procedures (Based on contents in Lab observation)
		Ability to construct the circuit diagram on a bread board and use meters/ instruments to record the measured data according to the range selected.(Based on physical observation)
2	Demonstrate an ability to design experiments to get the desired output.	Finding the appropriate values of the components to meet the specifications.
3	Demonstrate an ability to analyze the data and reach valid conclusions.	Ability to gather materials and writing in lab record (Based on lab record)

S.No	Performance Indicator	Excellent (A) 100%	Good(B) 80%	Need improvement (C) 60%	Fail (D) <40%
1	Laboratory preparation (verification of Lab observation) (5M)	Read and understand the lab manual before coming to lab. Observations are completed with necessary theoretical calculations including the use of units and significant figures	Analyze data for trends and correlations, stating possible errors and limitations in choosing the component values.	Observations are incomplete	No effort exhibited
2	Experimental procedures Conclusions of the lab experiment performed. (Based on physical observation, contents of lab record, viva)(5M)	Clearly describes the purpose of doing experiment and its scope. Follow the given experimental procedures, to obtain the desired output. Able to correlate the theoretical concepts with the concerned lab results with appropriate reasons	Tabulate data (tabular form or in graphical form) from the results so as to facilitate analysis and explanations of the data, and draw conclusions. Follow the given experimental procedures, but obtained results with some errors. Able to correlate the theoretical concepts with the concerned lab results with some difficulties.	Some idea of doing experiment but not very clear. Lacks the appropriate knowledge of the lab procedures. Has no idea what to do Not able to correlate the theoretical concepts with the concerned lab results	No effort exhibited
3	Ability to write a code and verify its	Able to perform tasks accurately without	Able to perform tasks with some	Poor in performing tasks without	No effort exhibited

	output.(Based on physical observation)(5M)	assistance , obtain the stimuli after calculations	Difficulties, obtain the correct stimuli for only few components after calculations	assistance, obtain the incorrect stimuli.	
4	Presentation of record (Based on Lab record)(5M)	Well-organized, interesting, confident presentation of record	Presentation of record acceptable	Presentation of record lacks clarity and organized	No effort exhibited
5	Oral Presentation (Based on Viva)(5M)	Responds confidently, and precisely in giving answers to questions correctly	Responds in giving answers to questions but some answers are wrong.	Responds in giving answers to questions but all answers are wrong.	No effort exhibited

The objective of this lab is to impart skill (both Programming-Assembly level & Hardware) in designing microcomputer systems. This Lab has 8085, 8086 microprocessor trainer kits and 8051 micro controller trainer kits along with interfacing modules to demonstrate the detailed applications of microprocessors& microcontrollers.

The facilities in the laboratory enable students to build a firm background in microcomputer hardware as well as software. Students learn about assembly language programming, memory and I/O design, interfacing of programmable chips and peripherals such as stepper motors, analog – to – digital and digital – to – analog converters etc.

LIST OF MAJOR EQUIPMENT IN MP & MC LABORATORY

S.NO	NAME OF THE EQUIPMENT	MAKE	QUANTITY
1.	Intel 8085 Microprocessor kits	ESA	23
2.	8086 Microprocessor kits	ESA	15
3.	8051 Microcontroller kits	ESA	15
4.	Universal Programmer (iup-uxp)	ESA	01
5.	20MHz Dual trace Oscilloscopes	APLAB	04
6.	PC Systems	HCL	12

TOTAL EXPENDITURE OF THE LABORATORY: Rs. 10,75,309.97/-

Do's

1. Proper dress code has to be maintained while entering in to the Lab.
2. Students should carry observation notes and record completed in all aspects.
3. Assembly level program and its theoretical result should be there in the observation before coming to the next lab.
4. Student should be aware of next ALPs.
5. Students should be at their concerned desktop/bench, unnecessary moment is restricted.
6. Student should follow the procedure to start executing the ALP they have to get signed by the Lab instructor for theoretical result then with the permission of Lab instructor they need to switch on the desktop and after completing the same they need to switch off and keep the chairs properly.
7. After completing the ALP Students should verify the ALP by the Lab Instructor.
8. The Practical Result should be noted down into their observations and result must be shown to the Lecturer In-Charge for verification.
9. Students must ensure that all switches are in the OFF position, desktop is shut down properly.

Don'ts

1. Don't come late to the Lab.
2. Don't leave the Lab without making proper shut down of desktop and keeping the chairs properly.
3. Don't leave the Lab without verification by Lab instructor.
4. Don't leave the lab without the permission of the Lecturer In-Charge.

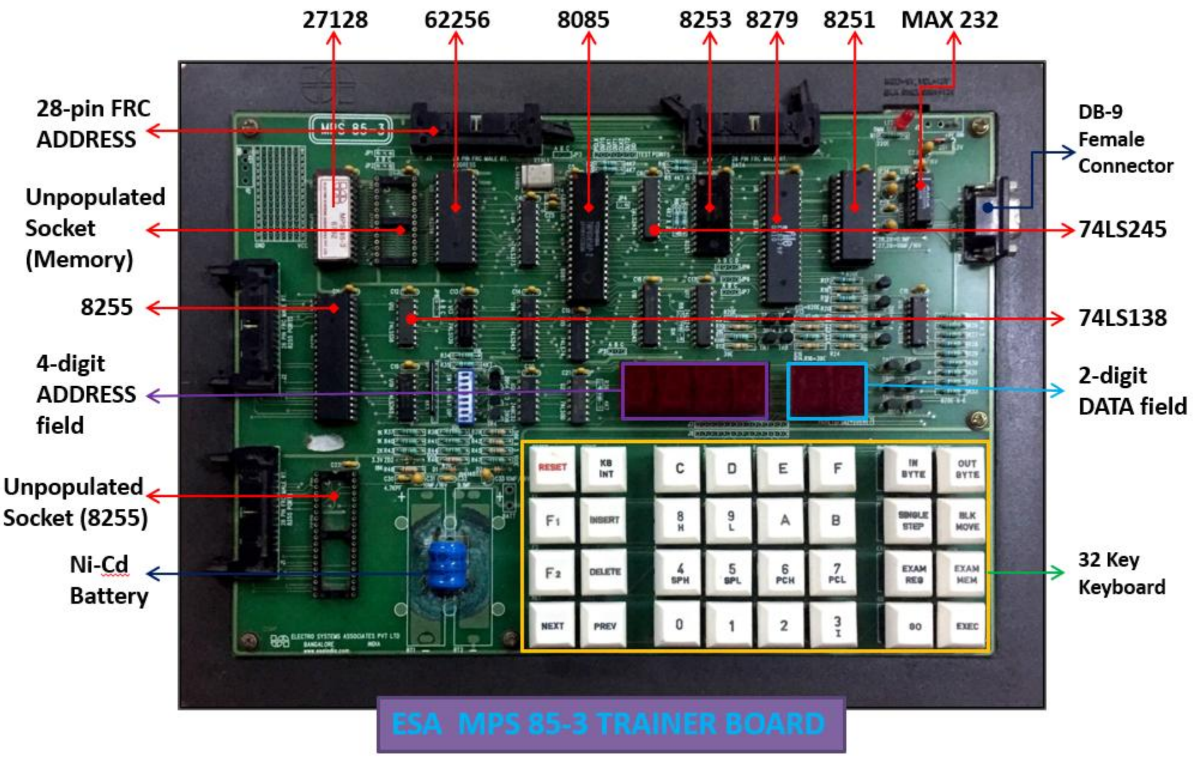
STUDY OF ESA 8085 MICROPROCESSOR KIT

AIM:

To study the 8085 (ESA-85/3) trainer kit and the details about various components present in the kit.

APPARATUS REQUIRED:

Item No	Name of the equipment / component	Specification	Quantity
1	8085 Microprocessor Kit	MPS 85 - 3	1
2	Power supply Adapter	5 V, 3 A	1



SPECIFICATIONS:

- 8085 operated at 3.072MHz
- 16Kbytes of firmware and 32Kbytes of Static RAM with battery backup.
- 4K/8K/16 K expansion memories.
- **8279**: 32 keyboard and 6 digit seven segment LED display.
- **8253**:3 programmable 16 bit interval timers.
- **8251**: Serial communication supporting for all standard bauds from 110 to 19200.
- **8255**: Programmable Peripheral interface, 24 programmable I/O lines.

KEYBOARD MONITOR COMMAND SUMMARY:

COMMAND	FUNCTION/FORMAT
Examine/Modify Memory	Displays/Modifies the contents of Memory location EXAM MEM<add > NEXT <data > [NEXT/PREV] EXEC
Examine/Modify Register	Displays/Modifies 8085 register contents EXAM REG < reg key > NEXT <data > [NEXT] EXEC
Single step	Execute a single user program instruction. SINGLE STEP <start add> NEXT <start add> NEXT EXEC
Go	Transfer control from monitor to user program Go <add> EXEC
Block Move	Moves a block of data from one portion to another. BLK <start add> NEXT <end add> NEXT <dest add> EXEC
Insert	Inserts one or more instructions in the user program INSERT <low limit> <high limit> NEXT <low insert add> NEXT, <No. of bytes> NEXT <data> EXEC
Delete	Deletes one or more instructions in the user program DELETE <low limit> <high limit> NEXT <low delete add> NEXT, <high delete add> EXEC

KEYBOARD MONITOR ROUTINES ACCESSIBLE TO USER:

CALLING ADDRESS	MNEUMONICS	FUNCTIONS
0440H	UPDAD	Updates address field of the display. The contents of the locations (CURAD), 8FEFH & 8FF0H are displayed in the address field. The contents of all the CPU registers and flags are affected. If Reg.B =1, dot at the right edge of the field; if B=0, no dot.
044CH	UPDDT	Updates data field of the display. The contents of the location (CURDT), 8FF1H are displayed in the data field. The contents of all CPU registers and flags are affected. If Reg. B=1, dot at the right edge of the field; if B=0, no dot.
0389H	OUTPUT	Output characters to display. The parameters for this routine are as follows: Reg.A=0-Use address field Reg.A=1-Use data field Reg.B=1-Dot at the edge of the field =0- No dot. Reg.HL=Starting address of character string to be displayed.
02BEH	CLEAR	Clears the display. This routine blanks the entire display field. Parameter is: Reg.B=1-dot at the right edge of the address field. =0-No dot.

Precautions:

1. Care must be exercised while converting assembly code into hexadecimal code.
2. Turn-off the kit when there is a power failure
3. Care must be exercised while inserting DB-9 connector.
4. Care must be exercised while inserting FRC
5. Turn-off the trainer kit when not in use.

RESULT:

Thus the study of ESA 85-3 trainer kit along with the peripherals connected on-board is carried out.

1. SIMPLE PROGRAMS ON 8085 MICROPROCESSOR

a) **Aim:** Write an assembly language program to move 8-Bit data into different registers.

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Data transfer instructions
ii. Register set of 8085 Microprocessor
iii. Program code generation

Pre – lab Questions:

1. What is the word length of 8085 Microprocessor?
2. What are the various registers in 8085 Microprocessor?
3. What is the purpose of program counter?

Equipment required:

1. MPA trainer Kit
2. +5V power supply

Program Table:

Memory Location	Mnemonic	Comments
8000	MVI A, 32H	Move 32H to register A
8002	MOV B, A	Moving the contents of A to B
8003	MOV C, B	Moving the contents of B to C
8004	MOV D, C	Moving the contents of C to D
8005	MOV E, D	Moving the contents of D to E
8006	MOV H, E	Moving the contents of E to H
8007	MOV L, H	Moving the contents of H to L
8008	HLT	Halt

Output Data:

Registers	Data
A	32H
B	32H
C	32H
D	32H
E	32H
H	32H
L	32H

b)Aim: Write an assembly language program to load registers E, D, C, A with the bytes from consecutive memory locations.

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Data transfer and arithmetic instructions
ii. Register set of 8085 Microprocessor
iii. Program code generation

Pre – lab Questions:

1. What is the addressing capacity of 8085 Microprocessor?
2. What is the clock frequency of 8085 Microprocessor?
3. What are the externally initiated operations of 8085 MP?

Equipment required:

1. MPA trainer Kit
2. +5V power supply

Program Table:

Memory Location	**Mnemonic**	**Comments**
8000	LXI H,9000H	Load HL with the contents in the memory location 9000H
8003	MOV E,M	Moving the contents of M to E
8004	INX H	Incrementing the HL pair
8005	MOV D,M	Moving the contents of M to D
8006	INX H	Incrementing the HL pair
8007	MOV C,M	Moving the contents of M to C
8008	INX H	Incrementing the HL pair
8009	MOV A,M	Moving the contents of M to A
800A	HLT	Halt

Input Data:

Memory Address	Data
9000H	10H
9001H	20H
9002H	30H
9003H	40H

Output Data:

Registers	Data
E	10H
D	20H
C	30H
A	40H

c) Aim: Write an assembly language program to check the status of D_5 bit in a given 8 bit number. If the bit is 1 place FFH in next memory location else not place 00H.

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Arithmetic and Logical instructions
ii. Register set of 8085 Microprocessor
iii. Program code generation

Pre – lab Questions:

1. What are the various conditional JUMP instructions?
2. How many t-states are required to execute unconditional JUMP instruction?
3. Name some, one byte, two-byte & three-byte instructions
4. What are the various addressing modes?

Equipment required:

1.MPA trainer Kit
2.+5V power supply

Program Table:

Memory Location	**Label**	**Mnemonic**	**Comments**
8000		LXI H,9000H	Load the HL with the contents of the memory location 9000H
8003		MOV A,M	Moving the contents of A to B
8004		ANI 20H	AND the contents of accumulator with 20H
8006		JNZ L1	Jump if no zero to L1
8009		INX H	Incrementing the HL pair
800A		MVI M,00H	Move 00H to the next memory Location
800C		HLT	Halt
800D	L1	INX H	Incrementing the HL pair
800E		MVI M,01H	Move 01H to the next memory Location
8010		HLT	Halt

Output Data:

Memory Address	Data
9001H	01H

d) Aim: Write an 8085 Assembly Language Program to check the status of byte

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Arithmetic and Logical instructions
ii. Register set of 8085 Microprocessor
iii. Program code generation

Pre – lab Questions:

1. What are the externally initiated operations of 8085 MP?
2. What is purpose of ALE signal in 8085 Microprocessor
3. What do you mean by the stack pointer register?

Equipment required:

1. MPA trainer Kit
2. +5V power supply

Program Table:

Memory Location	Label	Mnemonic	Comments
8000		LXI H,9000H	Load the HL with the contents of the memory location 9000H
8003		MOV A,M	Moving the contents of A to B
8004		ORI 00H	OR the contents of accumulator with 00H
8006		JNZ L1	Jump if no zero to L1
8009		INX H	Incrementing the HL pair
800A		MVI M,00H	Move 00H to the next memory Location
800C		HLT	Halt
800D	L1	INX H	Incrementing the HL pair
800E		MVI M,01H	Move 01H to the next memory Location
8010		HLT	Halt

Output Data:

Memory Address	Data
9001H	01H

e) **Aim:** Write an 8085 Assembly Language Program to count the number of 1‟s and 0‟s in given an 8- bit number

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Conditional Jump instructions
ii. Register set of 8085 Microprocessor
iii. Program code generation

Pre – lab Questions:

1. How is the instruction set classified?
2. What does “M” indicate in MOV A, M instruction?
3. What is the range of RAM area in the kit Microprocessor kit used?

Equipment required:

1. MPA trainer Kit
2. +5V power supply

Program Table:

Memory Location	Label	Mnemonic	Comments
9000		LXI H,8000H	Load the HL with the contents of the memory location 8000H
9003		MOV A,M	Move the contents of memory location to accumulator
9004		MVI B,08H	Move 08H to the register B
9006		MVI C,00H	Clearing register C
9008		MVI D,00H	Clearing register D
900A	L1	RAL	Rotate the contents of Acc to left
900B		JC L2	If carry bit is set jump to loop L2
900E		INR D	Increment the contents of D
900F		DCR B	Decrement the contents of register B
9010		JNZ L1	Jump if no zero to L1
9013		JMP L3	Jump to loop L3
9016	L2	INR C	Increment the contents of register C
9017		DCR B	Decrement the contents of B reg.
9018		JNZ L1	Jump if no zero to loop L1
901B	L3	INX H	Incrementing the HL pair register
901C		MOV M,C	Moving the contents of C to the memory location
901D		INX H	Incrementing the HL pair register
901E		MOV M,D	Moving the contents of D to the memory location
901F		HLT	Halt

OUTPUT DATA:

Memory Address	Data
8001H	03H
8002H	05H

2. PROGRAMS ON ADDITION & SUBTRACTION

a) Aim: Write an 8085 Assembly language program to ADD two 8-bit numbers result can be a 16-bit number.

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Branch instructions
ii. Register set of 8085 Microprocessor
iii. Program code generation

Pre – lab Questions:

1. What are the flags affected after ADD instruction?
2. What type of addressing mode does STA 16-bit add instruction represent?
3. What is the other Machine controlled instruction of 8085 Microprocessor?
4. How many machine cycles are required to execute LXIH 16-bit instruction? Explain what are those?

Equipment required:

1. MPA trainer Kit
2. +5V power supply

Input Data:	Address	Data
	A000H	A0H
	A001H	FFH

Program Table:

Memory Location	**Label**	**Mnemonic**	**Comments**
9000		LXI H,A000H	Load the HL with the contents of the memory location A000H
9003		MVI C,00H	Clearing register C
9005		MOV A,M	Move the contents of memory location to register A
9006		INX H	Incrementing the contents of HL pair
9007		ADDM	ADD the contents of memory location to accumulator
9008		JNC:L1	Jump if no carry to L1
900B		INR C	Incrementing the C register
900C	L1	INX H	Incrementing the contents of HL pair
900D		MOV M,A	Move the contents A to memory location.
900E		INX H	Incrementing the contents of HL pair
900F		MOV M,C	Move the contents of C to memory location
9010		HLT	Halt

Output Data:

Address	Data
A002H	9FH
A003H	01H

b) Aim: Write an 8085 Assembly language program to subtract two 8-bit numbers using 2"s complement store the actual result

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Branch instructions
ii. Register set of 8085 Microprocessor
iii. Program code generation

Pre – lab Questions:

1. In the program if LXIH B, 8050 is used instead of LXIH, 8050 then what is the instruction that has to be used in place of MOV A,M?
2. What is the addressing mode of SUB B instruction?
3. Are there two different flags for the borrow and the carry?
4. How do you perform 2"s complement subtraction?

Equipment required:

1. MPA trainer Kit
2. +5V power supply

Input Data:	Memory Address	Data
	8000H	B9H
	8001H	C3H

Program Table:

Memory Location	**Label**	**Mnemonic**	**Comments**
9000		LXI H,8000H	Load the HL with the contents of the memory location 8000H
9003		MOV B,M	Move the contents of memory location to register B
9004		INX H	Incrementing the contents of HL pair
9005		MOV A,M	Move the contents of memory location to register A
9006		CMA	Complement the contents of accumulator
9007		INR A	Incrementing the A register
9008		ADD B	Add the contents of accumulator to register B.
9009		JC L1	Jump if carry to L1
900C		CMA	Complement the contents of accumulator
900D		INR A	Incrementing the contents of register A
900E	L1	INX H	Incrementing the contents of HL pair

900F		MOV M,A	Move the contents A to memory location.
9010		HLT	Halt

Output Data:

Memory Address	Data
8002H	0AH

3. PROGRAMS ON LARGEST/SMALLEST AND SORTING

a) Aim: Write an 8085 Assembly language program to arrange a series of numbers in ascending order using Bubble sorting method.

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Instructions SET
ii. Register set of 8085 Microprocessor
iii. Stack and Subroutines
iv. Program code generation

.

Pre – lab Questions:

1. What type of addressing mode does STA 16-bit add instruction represent?
2. What happens internally when we give HLT instruction in our program in 8085 MP?
3. What is the other Machine controlled instruction of 8085 Microprocessor?

Equipment required:

1. MPA trainer Kit
2. +5V power supply

INPUT DATA:

Memory Address	Data
8000H	06H
8001H	DDH
8002H	CCH
8003H	BBH
8004H	FFH
8005H	AAH
8006H	EEH

OUTPUT DATA:

Memory Address	Data
8001H	AAH
8002H	BBH
8003H	CCH
8004H	DDH
8005H	EEH
8006H	FFH

PROGRAM:

Memory Location	Label	Mnemonic	Comments
9000		LXI H,8000H	Load the HL with the contents of the memory location 8000H
9003		MOV C,M	Move the contents of memory location to C .
9004		DCR C	Decrementing the contents of C register.
9005	L3	LXI H, 8000H	Load the HL with the contents of the memory location 8000H
9008		MOV D,M	Move the contents of memory location to register D
9009		DCR D	Decrementing the contents of register D.
900A		INX H	Incrementing the contents of HL pair
900B	L2	MOV A,M	Move the contents of memory location to register A
900C		INX H	Incrementing the contents of HL pair
900D		MOV B,M	Move the contents of memory Location to register B.
900E		CMP B	Compare the contents B with the contents of accumulator.
900F		JC L1	Jump if carry to L1
9012		DCX H	Decrementing the HL register pair
9013		MOV M,B	Move the contents of register B to memory location
9014		INX H	Incrementing the contents of HLpair
9015		MOV M,A	Move the contents of accumulator to memory Location
9016	L1	DCR D	Decrementing the D register
9017		JNZ L2	Jump if no zero to L2
901A		DCR C	Decrementing the C register
901B		JNZ L3	Jump if no zero to L3
901E		HLT	Halt

b) Aim: Write an 8085 Assembly language program to arrange a series of numbers in descending order using selection sort method

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Branch instructions
ii. Register set of 8085 Microprocessor
iii. Program code generation

Pre – lab Questions:

1. How many machine cycles are required to execute LXIH 16-bit instruction? Explain what are those?
2. What are the machine cycles of SHLD, 16-bit address instruction?
3. What are the flags affected after ADD instruction?

Equipment required:

1. MPA trainer Kit
2. +5V power supply

INPUT DATA:

Memory Address	Data
8000H	06H
8001H	DDH
8002H	CCH
8003H	BBH
8004H	FFH
8005H	AAH
8006H	EEH

PROGRAM:

Memory Location	Label	Mnemonic	Comments
9000		LXI H,8000H	Load the HL with the contents of the memory location 8000H
9003		LXI D,A000H	Load the DE with the contents of the memory location A000H
9006		MOV C,M	Move the contents of memory location to register C
9007	L1	CALL: LARGE	Call subroutine LARGE
900A		STAX D	Store the contents of accumulator in memory location pointed by DE register pair
900B		INX D	Increment the contents of DE pair
900C		CALL: CHECK	Call subroutine CHECK
900F		DCR C	Decrement the contents of reg C
9010		JNZ:L1	Jump on non zero to L1
9013		HLT	Halt

Subroutine: LARGE

Memory Location	Label	Mnemonic	Comments
B000		LXI H,8000H	Load the HL with the contents of the memory location 8000H
B003		XRA A	Clear Accumulator contents
B004		MOV B,M	Move the contents of memory location to register B
B005	L3	INX H	Increment the contents of HL pair
B006		CMP M	Compare the contents of memory location with the contents of accumulator.
B007		JNC:L2	Jump if no carry to L2
B00A		MOV A,M	Move the contents of memory location to Accumulator
B00B	L2	DCR B	Decrement the contents of reg B
B00C		JNZ:L3	Jump on non zero to L3
B00F		RET	Return to main program

Subroutine: CHECK

Memory Location	Label	Mnemonic	Comments
C000		LXI H,8000H	Load the HL with the contents of the memory location 8000H
C003		MOV B,M	Move the contents of memory location to register B
C004	L5	CMP M	Compare the contents of memory location with the contents of accumulator.
C005		JNZ:L4	Jump on non zero to L4
C008		DCR B	Decrement the contents of reg B
C009		JNZ:L5	Jump on non zero to L5
C00C	L4	MVI M,00H	Move 00H to memory location
C00E		RET	Return to main program

OUTPUT DATA:

Memory Address	Data
A000H	FFH
A001H	EEH
A002H	DDH
A003H	CCH
A004H	BBH
A005H	AAH

4. PROGRAMS ON MULTIPLICATION

a) Aim: Write an 8085 Assembly language program to perform multiplication of two 8-bit numbers using successive addition method result can be a16-bit.

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Branch instructions
ii. Register set of 8085 Microprocessor
iii. Program code generation

Pre – lab Questions:

1. If A< B while performing A-B then what is the status of the carry flag?
2. What are the flags affected after the instruction INX H?
3. What is the status of the AC and the CY flags after the ANI instruction?
4. Are there two different flags for the borrow and the carry?

Equipment required:

1. MPA trainer Kit
2. +5V power supply

INPUT DATA:

Memory Address	Data
9010H	22H
9011H	16H

PROGRAM:

Memory Location	Label	Mnemonic	Comments
F000		XRA A	Clear Accumulator contents
F001		MOV C,A	Store the contents of accumulator to register C
F002		LXI H,9010H	Load the HL with the contents of the memory location 9010H
F005		MOV D,M	Move the contents of memory location to register D.
F006		INX H	Incrementing the contents of HL Pair
F007		MOV B,M	Move the contents of memory location to register B.
F008	L2	ADD D	ADD the contents register D to Accumulator
F009		JNC L1	Jump if no carry to L1
F00C		INR C	Incrementing the contents C Register
F00D	L1	DCR B	Decrementing contents the B Register
F00E		JNZ L2	Jump if no zero to L2
F011		INX H	Incrementing the contents of HL Pair
F012		MOV M,A	Store the contents of accumulator to the memory location
F013		INX H	Incrementing the contents of HL Pair

F014		MOV M,C	Store the contents of register C to the memory location
F015		HLT	Halt

OUTPUT DATA:

Memory Address	Data
9012H	ECH
9013H	00H

b) Aim: Write an 8085 Assembly language program to perform multiplication of a 16-bit and a 8-bit numbers using shifting and adding method

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Branch instructions
ii. Register set of 8085 Microprocessor
iii. Program code generation

Pre – lab Questions:

1. Explain the operation performed by the instruction LHLD 16-bit address
2. Can you access the AC flag?
3. What is the operation performed by XCHG instruction?
4. What are the flags that are affected after DAD H instruction?

Equipment required:

3. MPA trainer Kit
4. +5V power supply

INPUT DATA:	Memory Address	Data
	8601H	37H
	8602H	25H
	8603H	12H

PROGRAM:

Memory Location	**Label**	**Mnemonic**	**Comments**
F000		LHLD 8601H	Load HL register pair with the contents of memory location 8601H and 8602H.
F003		XCHG	Exchange the contents of DE register pair with HL pair.
F004		LDA 8603H	Load Accumulator with the contents of 8603H
F007		LXI H,0000H	Load the HL with 0000H
F00A		MVI C,08H	Move 08H to the register C.
F00C	L2	DAD H	Add the contents HL pair with HL pair.
F00D		RAL	Rotate the contents of accumulator to left with carry.
F00E		JNC L1	Jump if no carry to L1
F011		DAD D	Add the contents HL pair with DE pair.
F012	L1	DCR C	Decrementing the contents Reg C
F013		JNZ L2	Jump if no zero to L2
F016		SHLD 8604H	Store the contents of HL pair in the memory location 8604H and 8605H
F019		HLT	Halt

OUTPUT DATA: Memory Address	Data
8604H	DEH
8605H	9DH

STUDY OF 8086 TRAINER KIT

AIM:

To study the 8086 (ESA 86 / 88E) trainer kit and the details about various components present in the kit.

APPARATUS REQUIRED:

Item No	Name of the equipment / component	Specification	Quantity
1	8086 Microprocessor Kit	MPS 86 - 3	1
2	Power supply Adapter	5 V, 3 A	1
3	Keyboard	Alphanumeric	1

THEORY:

ESA 86 / 88 E is an economical and powerful general-purpose microcomputer system that can be operated with 8086 or 8088 CPU. It may be used as an instructional, learning aid and also as an application development tool in R&D labs and industries. Following are the main features of ESA 86/88E:

Specifications:

- 8086 or 8088 CPU operating at 5 MHz in maximum mode
- Memory: ESA 86/88E provides a total of 128 K Bytes of on board memory
  - 64 K Bytes of ROM using two 27256 EPROMs
  - 64 K Bytes of RAM using two 62256 static RAM

Onboard peripherals & interfacing options:

  - 8251A – Universal Synchronous / Asynchronous Receiver / Transmitter supporting standard baud rates from 110 to 19,200. Baud rate is selected through on-board DIP switch setting
  - 8253 - 5 - programmable interval timer; Timer 0 is used for Baud clock generation, Timer 1 and Timer 2 are available for the user.
- 8255A – 3 Programmable Peripheral Interface provide up-to 72 programmable I/O line. One 8255 is used for controlling LCD and reading DIP switch. Two 8255s are for the user, of which one is populated by default and the other is optional.
- 8288 – Bus controller used for generating control in Maximum mode operation by receiving the status of 8086 through S2 – S0 lines.
- KBD CNTRL UPI (Universal Peripheral Interface)

External Interrupts

- KBINT Key: Non-maskable Interrupt (NMI) is provided through KBINT key on the trainer. It is considered as Type 2 interrupt in 8086 / 8088. The vectoring information for this interrupt is fully user defined.

- INTR: Maskable Interrupt (INTR) is available to user on system expansion connector J6

Internal Interrupts:

- Interrupt vectors Type 1 (Single step interrupt) and Type 3 (Break point interrupt) reserved for monitor. Type 3 interrupt is used with GO command and its use is left to the user's choice.

External Interface Signals:

- CPU bus: De-multiplexed and fully buffered, TTL compatible, Address, Data & control signals are available on two 26 pin ribbon cable connectors
- Parallel I/O: 48 Programmable parallel I/O lines (TTL compatible) through two 26 – pin ribbon cable connectors. Note that only one 8255 and its corresponding 26 – pin ribbon cable connector is available as default factory installation, which may additionally be used as a parallel printer interface. Further, ESA 86/88E firmware uses this 8255 for operations with ESA EPROM programmer interface also.
- Firmware : Serial and Keyboards monitor
  Centronics Printer interface driver software
  EPROM programming software
  Audio tape interface driver software

- Serial I/O: RS – 232C through on-board 9 – pin D – type female connector and self powered USB through USB type-B connector.
- PC Keyboard: A keyboard is to be interfaced to enter mnemonics and commands to the processor. This has to be connected through the PS/2 connector provided for interfacing the PC keyboard.
- 20x4 LCD: 15 – pin flow strip for interfacing the 20 x 4 LCD
- Power supply: + 5V at 3A (approximately)
- Battery back-up: 3.6 V Ni-Cd Battery as Power back-up to RAM (Optional)

System Capabilities:

- Assemble 8086/8088 Instruction mnemonics using ESA 86/88E Symbolic one-line assembler
- Disassembler Hex bytes from memory into 8086/8088 CPU instruction using

monitor resident disassembler.

- **Note:** A disassembler is a computer program that translates machine language into assembly language—the inverse operation to that of an assembler.
- Execute the user program at full speed or debug the program through Single step and break point facilities
- Examine / modify the contents of memory location in byte or word format
- Examine / modify the content of CPU registers
- Write or read data to or from I/O ports (byte or word format)
- Communicate with a host PC serially through RS-232C / USB interface at a baud rate of upto 19,200 and develop debug applications using the user-friendly Windows or DOS driver packages
- Support for uploading user programs to host PC and saving them as hex files on a PC
- Reads, programs, verify and blank check of popular EPROMs (2716 through 27512) using optional EPROM programmer interface module.
- Use the monitor resident Centronics compatible parallel Printer driver software and obtain hard copies of serial mode operations.

Dip switch settings:

The following table summarizes the Dip switch settings for different configurations of ESA 86/ 88E trainer.

DIP switch No	Functionality	Configuration	Remarks
1 – 3	Baud rate setting for RS 232 C / USB communication	000 – 19,200 001 – 9600 (default) 010 - 4800 011 - 2400 100 - 1200 101 - 600 110 - 300 111 – 110 Bits / sec.	ESA 86 / 88 E communicates at a max. baud rate of 19,200 with the Windows Driver package
4	Operational mode	ON – Serial mode OFF – Stand-alone mode (default setting)	
5	Printer driver enable	On – Enabled OFF – Disabled	As per default factory setting printer is disabled
6	-	-	Reserved
7	CPU selection	On- 8086 (default) OFF – 8088	Both CPU’s operate at 5MHz in maximum mode
8	-	-	Reserved

TRAINER KIT DIAGRAM:

Result:

Thus the study of ESA 86/ 88E trainer kit along with the peripherals connected on-board is carried out.

5. PROGRAMS ON ADDITION

a) **Aim:** Write an 8086 Assembly language program to ADD two 16-bit numbers result can be a 24-bit number.

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Branch instructions
ii. Register set of 8086 Microprocessor
iii. Program code generation/.ASM file execution

Pre – lab Questions:

1. What are the flags affected after ADD instruction?
2. What type of addressing mode does MOV AX,[BX] instruction represent?
3. What is the DW directive?

Equipment required:

1. MPA trainer Kit/PC with MASM
2. +5V power supply/AC Supply

Program:

```
DATA SEGMENT
  NUM DW 1234H, 0F234H
  SUM  DW 2 DUP(0)
DATA ENDS
CODE SEGMENT
    ASSUME CS: CODE, DS:DATA
START: MOV AX,DATA
       MOV DS,AX
       MOV AX,NUM             ; First number loaded into AX
       MOV BX,0H              ; For carry BX register is cleared
       ADD AX,NUM+2           ; Second number added with AX
       JNC DOWN               ; Check for carry
       INC BX                 ; If carry generated increment the BX
DOWN: MOV SUM,AX              ; Storing the sum value
       MOV SUM+2,BX           ; Storing the carry value
       MOV AH,4CH
       INT 21H
CODE ENDS
END START
```

```
  INPUT   : 1234H, F234H
OUTPUT :  10468H
```

b) Aim: Write an 8086 Assembly language program to ADD two 32-bit numbers result can be a 40-bit number.

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Branch instructions
ii. Register set of 8086 Microprocessor
iii. Program code generation/.ASM file execution

Pre – lab Questions:

1. What are the flags affected after ADC instruction?
2. Is MOV [AX],[BX] allowed
3. What is ASSUME?

Equipment required:

1. MPA trainer Kit/PC with MASM
2. +5V power supply/AC Supply

Program:

```
  DATA SEGMENT
    NUM1 DW 0FFFFH,0FFFFH
    NUM2 DW 1111H,1111H
    SUM  DW 4 DUP(0)
  DATA ENDS
  CODE SEGMENT
       ASSUME CS:CODE,DS:DATA
  START: MOV AX,DATA
             MOV DS,AX
             MOV AX,NUM1             ;Move LSB of  NUM1 to AX
            ADD AX,NUM2              ;Add  LSB of NUM2  to AX
            MOV SUM,AX               ;Store the LSB  in SUM
            MOV AX,NUM1+2            ; Move MSB of  NUM1 to AX
            ADC AX,NUM2+2            ; Add  MSB of NUM2  to AX

            JNC DOWN                   ; Check for carry
            MOV SUM+4,01H            ; Store the carry in SUM+4
DOWN:  MOV SUM+2,AX                  ; Store the MSB in SUM+2
           MOV AH,4CH
            INT 21H
  CODE ENDS
  END START
```

INPUT: 0FFFFFFFFH, 011111111H
OUTPUT: 0111111110H

6. PROGRAMS ON FACTORIAL OF A NUMBER, FIBONACCI SERIES

a) Aim: Write an 8086 Assembly language program to find factorial of a given number

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Branch instructions
ii. Register set of 8086 Microprocessor
iii. Program code generation/.ASM file execution

Pre – lab Questions:

1. What are the Control flags?
2. What is ALE
3. How CX will effect LOOP instruction.

Equipment required:

3. MPA trainer Kit/PC with MASM
4. +5V power supply/AC Supply

Program:

```
DATA SEGMENT
      X DW 06H
      FACT DW ?
DATA ENDS

CODE SEGMENT
  ASSUME CS:CODE,DS:DATA
  START: MOV AX,DATA
        MOV DS,AX
           MOV AX,01H              ;Set the value of AX as 01H.
           MOV CX,X                ;Move the i/p number to CX.
                                   ;Perform the Loop multiplication
     UP: MUL CX                    operation.
           LOOP UP
           MOV FACT,AX             ;Store the FACT value.
           MOV AH,4CH
           INT 21H

   CODE ENDS
   END START

   Input: 06
   Output: 2D0H
```

b) Aim: Write an 8086 Assembly language program to generate Fibonacci Series

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Branch instructions
ii. Register set of 8086 Microprocessor
iii. Program code generation/.ASM file execution

Pre – lab Questions:

1. What is LEA?
2. What type of addressing mode does DAA instruction represent?
3. What happnes to string retrival if STD?

Equipment required:

1. MPA trainer Kit/PC with MASM
2. +5V power supply/AC Supply

Program:

```
.MODEL SMALL

.DATA

RES DB ?

CNT DB 0AH ; Initialize the counter for the no of Fibonacci No needed

.CODE

START: MOV AX,@DATA

MOV DS,AX

LEA SI,RES

MOV CL,CNT ; Load the count value for CL for looping

MOV AX,00H ; Default No

MOV BX,01H ; Default No



;Fibonacci Part

L1:ADD AX,BX
```

```
DAA ; Used to Present the value in Decimal Form
MOV [SI],AX
MOV AX,BX
MOV BX,[SI]
INC SI
LOOP L1

INT 3H ; Terminate the Program
END START
```

Output:1, 1, 2, 3, 5, 8, etc.

7.. HEXADECIMAL AND DECIMAL COUNTERS

AIM: To Write 8086 ALP to realize Hexadecimal and Decimal counters
Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Branch instructions
ii. Register set of 8086 Microprocessor
iii. Program code generation/.ASM file execution

Pre – lab Questions:

1. What is LEA?
2. What type of addressing mode does DAA instruction represent?
3. What happnes to string retrival if STD?

APPARATUS: 1.8086 mp kit -1& Adopter-1
2. System-1
3. RS 232 Serial cable.
4.+5v Supply
5.86/88e Driver Software.

PROGRAM:

```
DATA SEGMENT
NUM DW 1234H
 RES DB 10 DUP ('$')
DATA ENDS
START:
MOV AX,DATA
MOV DS,AX
MOV AX,NUM
LEA SI,RES
CALL HEX2DEC
LEA DX,RES
MOV AH,9
INT 21H
MOV AH,4CH
INT 21H
CODE ENDS
HEX2DEC PROC NEAR
MOV CX,0
MOV BX,10
LOOP1: MOV DX,0
DIV BX
ADD DL,30H
 PUSH DX
```

```
INC CX
CMP AX,9
JG LOOP1
ADD AL,30H
MOV [SI],AL
LOOP2: POP AX
INC SI
MOV [SI],AL
LOOP LOOP2
RET
HEX2DEC ENDP
END START
```

Result: A program has been written to count 0,1,2,…. FF (hexadecimal 8-bit counter) and 0,1,2,3……..99(decimal 8-bit counter)

8. PROGRAM ON SORTING

Aim: Write an 8086 Assembly language program to sort the numbers in ascending/descending order

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Branch instructions
ii. Register set of 8086 Microprocessor
iii. Program code generation/.ASM file execution

Pre – lab Questions:

1. What are the flags affected after INC instruction?
2. What type of addressing mode does MOV AX,[SI]instruction represent?
3. What is the DT directive?

Equipment required:

1. MPA trainer Kit/PC with MASM
2. +5V power supply/AC Supply

Program:

```
DATA SEGMENT                             ;start of data segment
        x DW 42H,34H,26H,17H,09H
        LEN EQU 05
        ASCD DB 10 DUP(0)
DATA ENDS                                ;end of data segment

CODE SEGMENT                             ;start of code segment
      ASSUME CS:CODE,DS:DATA
      START:  MOV AX,DATA                ;initialize data segment
              MOV DS,AX
              MOV BX,LEN-1               ;load BX(counter1) with count
                                         value(number of data words in array - 1)
              MOV CX,BX                  ;make a copy of the count value in CX(counter2)
        UP1:  MOV BX,CX                  ;load the updated CX in BX
              LEA SI,X                   ;SI points to the first number in the array
              UP:MOV AX,[SI]             ;make a copy of the number pointed by SI in
                                         AX
              MOV DX,[SI+2]              ;make a copy of the next number in DX
              CMP AX,DX                  ;compare both the numbers
             JB DOWN/JA DOWN             ;if AX < DX/AX > DX retain them as it is
              MOV [SI],DX                ;if not sort the numbers in ascending order
              MOV [SI+2],AX
     DOWN:  INC SI                       ;point to the next number
              INC SI
              DEC BX                     ;decrement the counter1
              JNZ UP                     ;compare till the larger number is sorted at
                                         the end of the array
              DEC CX                     ;decrement counter2
              JNZ UP1                    ;compare till the numbers are sorted in
                                         ascending order
              MOV AH,4CH                 ;terminate the process
```

```
                INT 21H
CODE ENDS                                       ;end of code segment
END START
```

OUTPUT: 09 17 26 34 42

9. INTERFACING OF DIGITAL TO ANALOG CONVERTER

AIM: To write a program to generate saw-tooth waveform.

Pre – Requisites:

The student should have completed the following study before doing this experiment

- i. Branch instructions
- ii. Register set of 8086 Microprocessor
- iii. Program code generation/.ASM file execution
- iv. 8255

Pre – lab Questions:

1. Explain CWR of 8255
2. What are different Parallel data transfer methods.
3. What is the value of D7 to be sent to CWR to initialize Ports

APPARATUS:

1.DAC

2.8255 Study card

3. Adapter, Keyboard, Cables, CRO Etc.

An assembly language program to generate a sawtooth waveform of period 1ms with Vmax 5V.

```
ASSUME     CS:CODE
CODE SEGMENT
START:          MOV AL,80h              :Make all ports output
                OUT CW,AL
AGAIN:          MOV AL,00h              : Start voltage for ramp
BACK:           OUT PA,AL
                INC AL
                CMP AL,0FFH
                JB BACK
                JMP AGAIN
CODE ENDS
END:            START
```

Output: An 8086 program has been written to generate sawtooth waveform.

10.INTERFACING OF ANALOG TO DIGITAL CONVERTER

AIM: To Interface ADC0808 with 8086 using 8255ports.

Pre – Requisites:

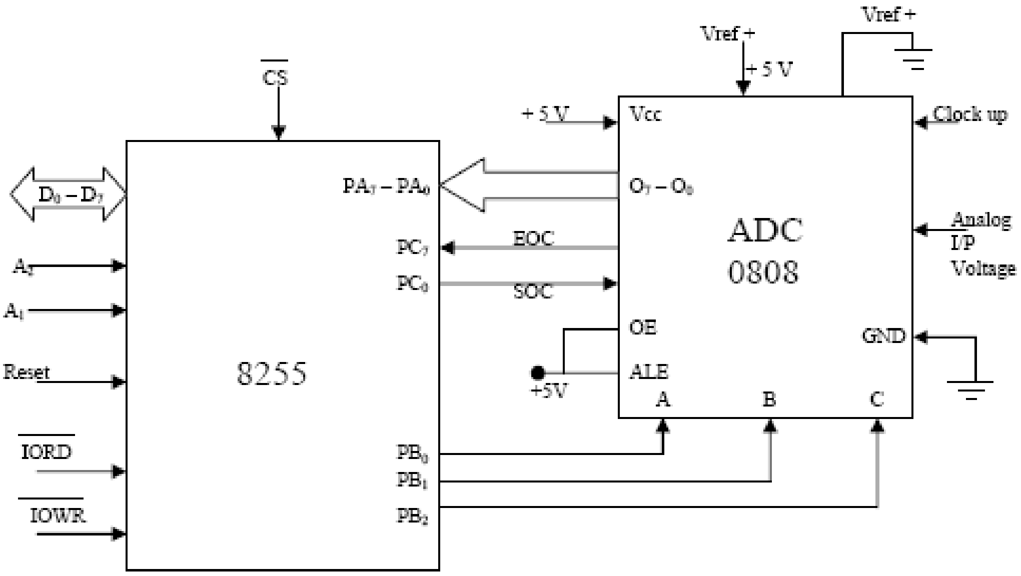
The student should have completed the following study before doing this experiment

i. Branch instructions
ii. Register set of 8086 Microprocessor
iii. Program code generation/.ASM file execution
iv. 8255

Pre – lab Questions:

1. Explain CWR of 8255
2. What are different Parallel data transfer methods.
3. What is the value of D7 to be sent to CWR to initialize Ports

APPARATUS:
1.DAC
2.8255 Study card
3. Adapter, Keyboard, Cables, CRO Etc.



Interfacing 0808 with 8086

The 8255 control word is written as follows

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	0

The required ALP is as follows:

```
                MOV       AL,98h          :Initialize 8255 as discussed above
                OUT       CWR,AL
                MOV       AL,02h          : Select I/P2 as analog input
                OUT       Port B,AL
                MOV       AL,00h          : Give start of conversion
                OUT       Port C, AL      : Pulse to the ADC
                MOV       AL,01h
                OUT       Port C, AL
                MOV      AL,00h
                OUT       Port C,AL
WAIT:           IN       AL,Port C        :Check for EOC by
                RCR
                JNC WAIT
                IN       AL,Port A
                HLT
```

Output: An 8086 program has been written to interface ADC with 8086 microprocessor.

11. INTERFACING 8255

Aim: Write an 8085 Assembly language program to take inputs from the 8-switches and send the same data to LED's by interfacing 8255 study card

Pre – Requisites:

The student should have completed the following study before doing this experiment

v. Branch instructions
vi. Register set of 8086 Microprocessor
vii. Program code generation/.ASM file execution
viii. 8255

Pre – lab Questions:

4. Explain CWR of 8255
5. What are different Parallel data transfer methods.
6. What is the value of D7 to be sent to CWR to initialize Ports

Equipment required:

1. MPA trainer Kit/PC with MASM
2. +5V power supply/AC Supply
3. 8255 study card
4. FRC

Theory:

The Intel 8255A is a general-purpose programmable, parallel I/O device designed for use with Intel Microprocessor. It can be programmed to transfer data under various conditions from simple I/O to Interrupt I/O.

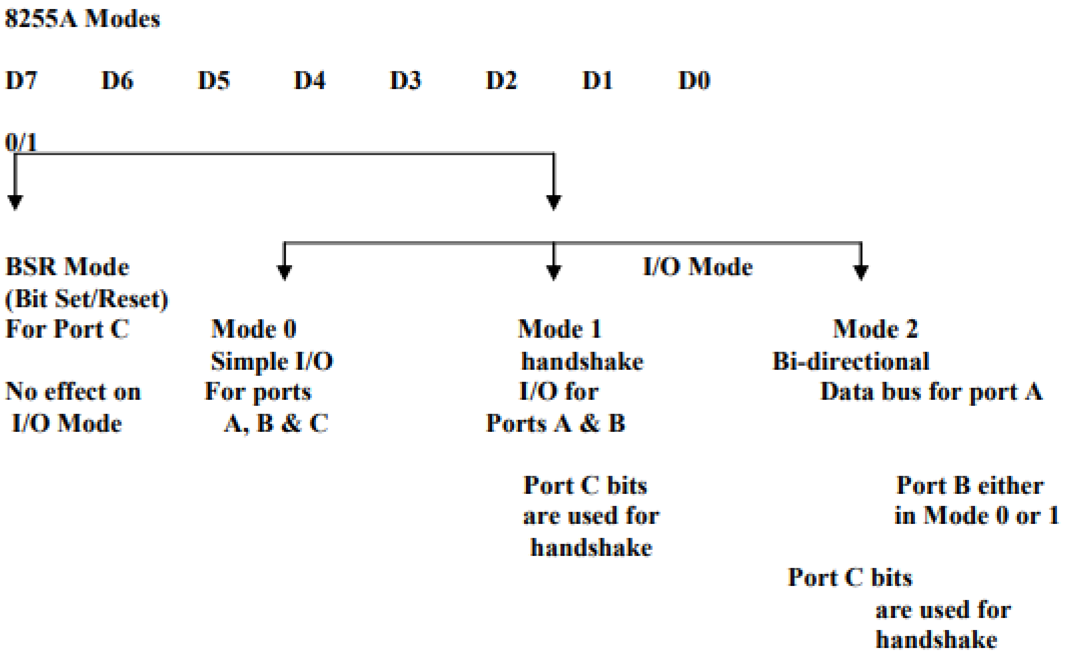


Figure 1: 8255 Modes

The 8255A has 24 I/O PINS that can be grouped primarily in two 8 Bit parallel ports A & B, with the remaining 8 bits as port C. The 8 bits of port can be used as individual bits or be grouped in two 4bit ports. C upper and C lower. The functions of these ports are defined by writing a control word in the control register.

Figure 1 shows all the functions of 8255A, classified according to two modes: the Bit Set/Reset (BSR) mode and the I/O mode. The I/O mode is further divided into three modes: Mode 0, Mode 1, and Mode 2. . In mode 0, all ports function as simple I/O ports. Mode 1 is a handshake mode whereby ports A and /or B use bits from port C as handshake signals. In the handshake mode, two types of I/O data transfer can be implemented: status checks and interrupt. In Mode 2, port A can be set up for bi-directional data transfer using handshake signals from port C, and port B can be set up either in Mode 0 or Mode 1. 8255A Modes D7 D6 D5 D4 D3 D2 D1 D0 0/1 BSR Mode I/O Mode (Bit Set/Reset) For Port C Mode 0 Mode 1 Mode 2 Simple I/O handshake Bi-directional No effect on For ports I/O for Data bus for port A I/O Mode A, B & C Ports A & B Port C bits Port B either are used for in Mode 0 or 1 handshake Port C bits are used for handshake FIG-1

The BSR Mode is used to set or reset the bits in port C. The I/O mode is further divided into three modes. Mode 0, Mode 1 and Mode 2. In Mode 0 all ports functions as simple I/O ports. Mode 1 is a Handshake mode where by ports A&B use bits from port C as Handshake signals.

Program:

```
    MVI A,90H; Initialize 8255's CWR with 10010000B
    OUT 83H;
L1: IN 80H; Take input from switches present at 80H port address
    OUT 81H: Send the switches data to LED's present at 81H port address
    STA 8FF1H; and to the 7-segment display in the data field
    CALL 044CH
    JMP L1; Repeat the process
```

Output: An 8085 program has been written to read switches at the port 80H and sent the same data to the LED's available at 81H.

`12. INTERFACING STEPPER MOTOR

Aim: Write an 8086 Assembly language program to interface stepper motor to rotate in clockwise direction.

Pre – Requisites:

The student should have completed the following study before doing this experiment

- i. Branch instructions
- ii. Register set of 8086 Microprocessor
- iii. Program code generation/.ASM file execution
- iv. Basics of stepper motor
- v. 8255

Pre – lab Questions:

1. Explain CWR of 8255
2. What are different Parallel data transfer methods.

Equipment required:

1. MPA trainer Kit/PC with MASM
2. +5V power supply/AC Supply
3. Stepper motor Study card and Adapter
4. FRC

Theory:

The Intel 8255A is a general-purpose programmable, parallel I/O device designed for use with Intel Microprocessor. It can be programmed to transfer data under various conditions from simple I/O to Interrupt I/O.

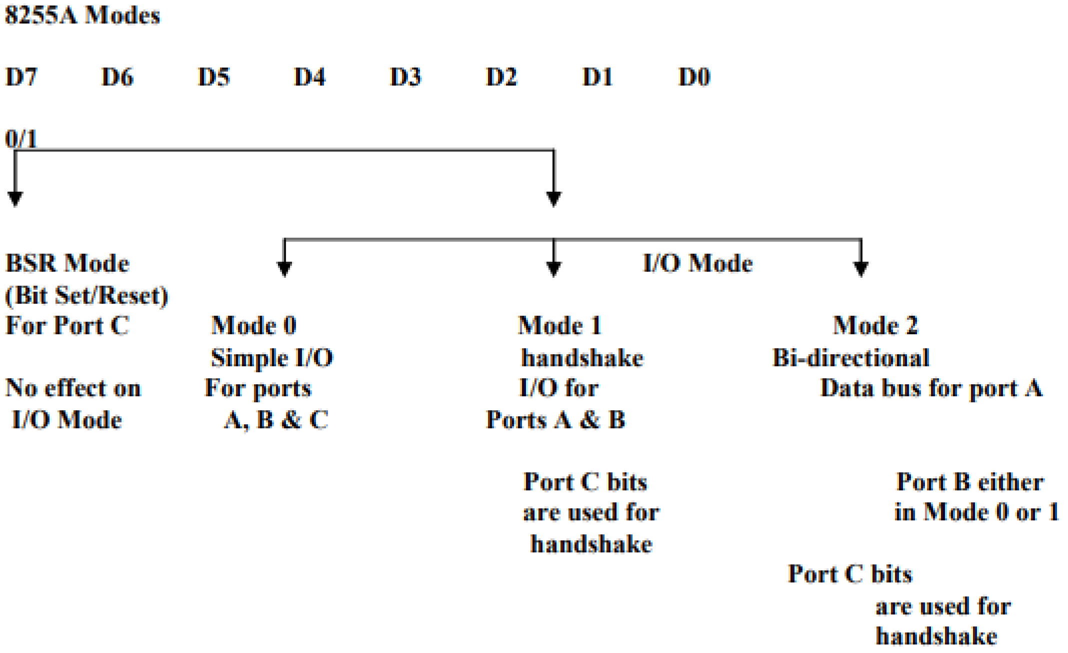


Figure 1: 8255 Modes

The 8255A has 24 I/O PINS that can be grouped primarily in two 8 Bit parallel ports A & B, with the remaining 8 bits as port C. The 8 bits of port can be used as individual bits or be grouped in two 4bit ports. C upper and C lower. The functions of these ports are defined by writing a control word in the control register.

Figure 1 shows all the functions of 8255A, classified according to two modes: the Bit Set/Reset (BSR) mode and the I/O mode. The I/O mode is further divided into three modes: Mode 0, Mode 1, and Mode 2. . In mode 0, all ports function as simple I/O ports. Mode 1 is a handshake mode whereby ports A and /or B use bits from port C as handshake signals. In the handshake mode, two types of I/O data transfer can be implemented: status checks and interrupt. In Mode 2, port A can be set up for bi-directional data transfer using handshake signals from port C, and port B can be set up either in Mode 0 or Mode 1. 8255A Modes D7 D6 D5 D4 D3 D2 D1 D0 0/1 BSR Mode I/O Mode (Bit Set/Reset) For Port C Mode 0 Mode 1 Mode 2 Simple I/O handshake Bi-directional No effect on For ports I/O for Data bus for port A I/O Mode A, B & C Ports A & B Port C bits Port B either are used for in Mode 0 or 1 handshake Port C bits are used for handshake FIG-1

The BSR Mode is used to set or reset the bits in port C. The I/O mode is further divided into three modes. Mode 0, Mode 1 and Mode 2. In Mode 0 all ports functions as simple I/O ports. Mode 1 is a Handshake mode where by ports A&B use bits from port C as Handshake signals.

Stepping Modes of a Stepper Motor:

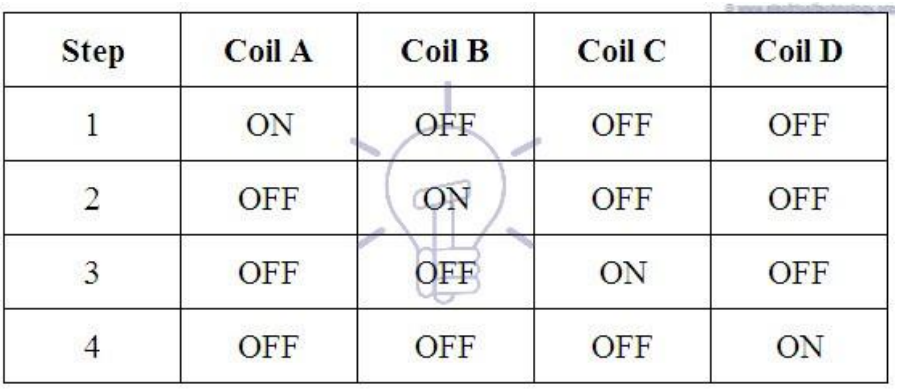
A typical stepping action causes the motor to step through a sequence of equilibrium positions in response to current pulses given to it. It is possible to vary the stepping action in different ways simply by changing the sequence through which stator windings are energized. The following are the most common operating or driving modes of stepper motors.

1. Wave step
2. Full step
3. Half step
4. Micro stepping

Wave Step Mode:

Wave step mode is the simplest of all other modes in which only one winding is energized at any given time. Each coil of the phase is connected to the supply alternatively. The table below shows the order through which coils are energized in a 4-phase stepper motor.

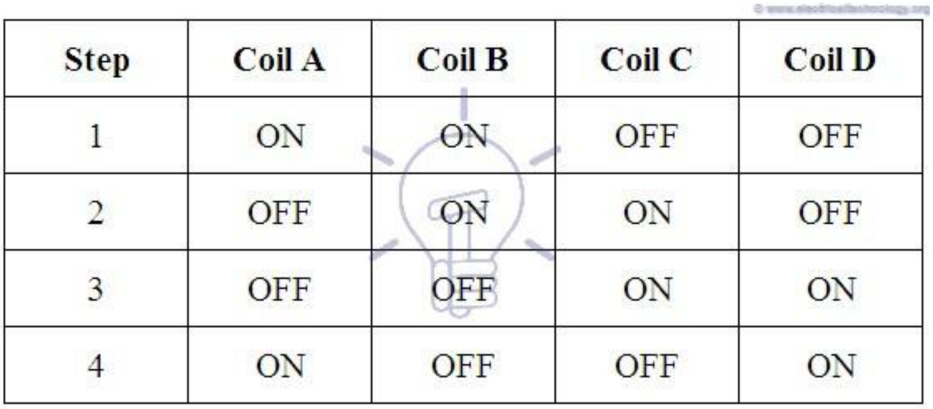
In this mode motor gives maximum step angle compared to all other modes. It is the simplest and most commonly used mode for stepping; however the torque produced is less as it uses some part of the total winding at a given time.



Step	Coil A	Coil B	Coil C	Coil D
1	ON	OFF	OFF	OFF
2	OFF	ON	OFF	OFF
3	OFF	OFF	ON	OFF
4	OFF	OFF	OFF	ON

Full Step Mode:

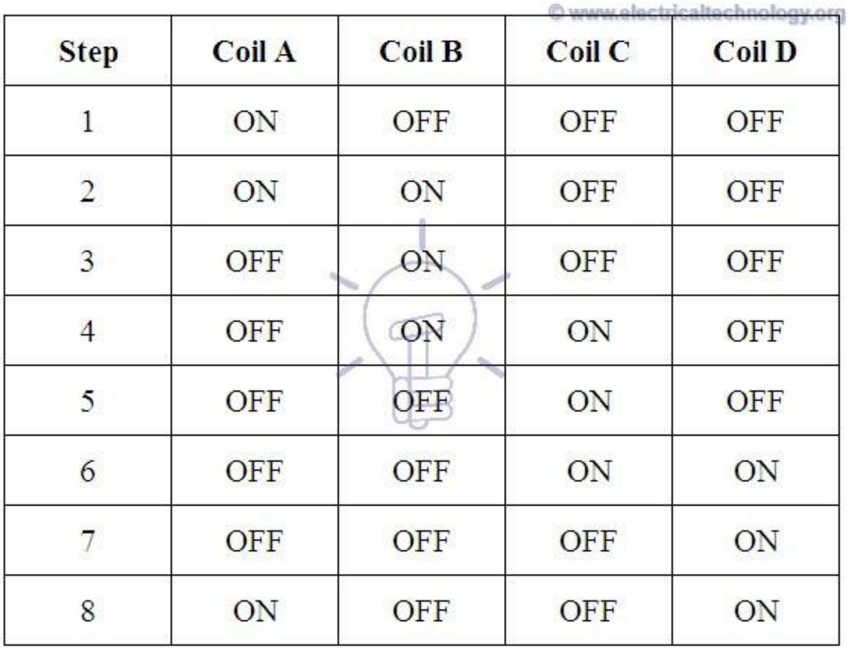
In this drive or mode, two stator phases are energized simultaneously at any given time. When two phases are energized together, the rotor will experience the torque from both phases and comes to the equilibrium position, which will be interleaved between two adjacent wave step positions or 1-phase excitations. So this step provides better holding torque than wave step. The table below shows the full step drive for 4-phase stepper motor.



Step	Coil A	Coil B	Coil C	Coil D
1	ON	ON	OFF	OFF
2	OFF	ON	ON	OFF
3	OFF	OFF	ON	ON
4	ON	OFF	OFF	ON

Half Step Mode:

It is the combination of both wave and full step modes. In this, single phase and dual phase excitations are carried out alternatively, i.e., one-phase ON, two-phases ON, and so on. The step angle in this mode becomes half of the full step angle. This drive mode has highest torque and stability compared to all other modes. The table containing phase pulsing sequence for a 4-phase motor in half stepping is given below.



Step	Coil A	Coil B	Coil C	Coil D
1	ON	OFF	OFF	OFF
2	ON	ON	OFF	OFF
3	OFF	ON	OFF	OFF
4	OFF	ON	ON	OFF
5	OFF	OFF	ON	OFF
6	OFF	OFF	ON	ON
7	OFF	OFF	OFF	ON
8	ON	OFF	OFF	ON

Micro stepping Mode:

In this mode, each motor step is subdivided into several small steps, even hundreds of fixed positions, therefore a greater positioning resolution is obtained. In this, currents through the windings are continually varied in order to get very small steps. In this, two phases are excited simultaneously, but with the unequal currents in each phase.

For example, the current through phase -1 is held constant while the current through phase-2 is incremented in steps till the maximum value of current, whether it is negative or positive. The current in the phase-1 is then decreased or increased in steps till zero. Thus, the motor will produce a small step size.

All these stepping modes can be obtained by each type of stepper motor discussed above. However, the direction of current in each winding during these steps can be varied depending on the type of motor and either it is unipolar or bipolar.

Program:

```
MOV AL,#80. ;Initialize 8255
MOV DX,#0FFE6 ;All ports output.
OUT DX,AX
MOV AL,#88 ;Initial bit pattern to be sent to motor
MOV DX,#0FFE0 ;pattern sent on
OUT DX,AX ;PA0-PA3.
CALL DELAY ;Delay between
XCHG AX,AX ;successive steps
```

```
            ROR AL,1 ;Next pattern to be sent
            JMP 2008 ;Go back to send ;the next pattern

   DELAY: PUSHF
            PUSH AX ; Subroutine for the delay between successive Steps
            MOVW BX,#0FFF
  LOOP: DECW BX
            JNZ LOOP
            POP AX
            RET
```

Result: A program has been written to rotate stepper motor in Clockwise direction

13. SEVEN SEGMENT DISPLAY

Aim:

Program to interface 8086 microprocessor kit with Seven Segment Display and make any digit to Blink.

Pre – Requisites:

The student should have completed the following study before doing this experiment

i. Branch instructions
ii. Register set of 8086 Microprocessor
iii. Program code generation/.ASM file execution
iv. Basics of stepper motor
v. 8255

Pre – lab Questions:

1. How does a 7 segment display work?
2. What diode is used in seven segment display?.

Equipment required:

1. MPA trainer Kit/PC with MASM
2. +5V power supply/AC Supply
3. 7-segment display

Program:

In this program we interface 8086 microprocessor with Seven Segment Display and diaplay a digit. To interface with Seven Segment Display,we must be familiar with 8255A PPI, which has threeports which can be used as I/O ports. The ports A, B and C are identified using addresses 19H, 1BH, and 1DH respectively and control register address is 1FH. We use Seven Segment Display structure using the following digit.

	Digit	Hexadecimal	Decimal	Binary
8	0	C0	192	11000000
	1	F9	249	11111001
	2	A4	164	10100100
	3	B0	176	10110000
	4	99	153	10011001
	5	92	146	10010010
	6	82	130	10000010
	7	F8	248	11111000
	8	80	128	10000000
	9	90	144	10010000

code:

Source program:

```
 MOV AL,10000000B
 OUT PPI_C, AL
STR:   MOV AL, 10011001B
 OUT PPIA,AL
 MOV CX, FFFFH
DEL:    NOP
 NOP
 NOP
 LOOP DEL
 MOV AL, 11111111B
 OUT PPIA, AL
 MOV CX, FFFFH
DEL1:    NOP
 NOP
 NOP
 LOOP DEL1
 JMP  STR
 HLT
```

Result: A program has been written to display number on seven segment display.