DRONACHARYA
College of Engineering

LABORATORY MANUAL

B.Tech. Semester- V

MICROPROCESSOR LAB
Subject code: LC-CSE- 321G

Prepared by:	Checked by:	Approved by:
Prof Renu Narwal	Dr. Ashima Mehta	Prof. (Dr.) Isha Malhotra
Sign.: …….	Sign.: …….	Sign.: …….

Table of Contents

Vision and Mission of the Institute

Vision:

"Empowering human values and advanced technical education to navigate and address global challenges with excellence."

Mission:

- M1: Seamlessly integrate human values with advanced technical education.
- M2: Supporting the cultivation of a new generation of innovators who are not only skilled but also ethically responsible.
- M3: Inspire global citizens who are equipped to create positive and sustainable impact, driving progress towards a more inclusive and harmonious world.

Vision and Mission of the Department

Vision:

"Steering the future of computer science through innovative advancements, fostering ethical values and principles through technical education."

Mission:

M1: Directing future innovations in computer science through revolutionary progress.

M2: Instilling a foundation of ethical values and principles in every technologist.

M3: Offering a comprehensive technical education to equip individuals for a meaningful and influential future.

Programme Educational Objectives (PEOs)

PEO1: Apply the technical competence in Computer Science and Engineering for solving problems in the real world.

PEO2: Carry out research and develop solutions on problems of social applications.

PEO3: Work in a corporate environment, demonstrating team skills, work morals, flexibility and lifelong learning.

Programme Outcomes (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and software tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: 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.

Programme Specific Outcomes (PSOs)

PSO1: Exhibit design and programming skills to develop and mechanize business solutions using revolutionary technologies.

PSO2: Learn strong theoretical foundation leading to brilliance and enthusiasm towards research, to provide well-designed solutions to complicated problems.

PSO3: Work effectively with diverse Engineering fields as a team to design, build and develop system applications.

University Syllabus

1 1. Write a program using 8085 and verify for:

a. Addition of two 8-bit numbers.

b. Addition of two 8-bit numbers (with carry).

2. Write a program using 8085 and verify for:

a. 8-bit subtraction (display borrow)

b. 16-bit subtraction (display borrow)

3. Write a program using 8085 for multiplication of two 8- bit numbers by repeated addition method. Check for minimum number of additions and test for typical data.

4. Write a program using 8085 for multiplication of two 8- bit numbers by bit rotation method and verify.

5. Write a program using 8086 for finding the square root of a given number and verify.

6. Write a program using 8086 for copying 12 bytes of data from source to destination and verify.

7. Write a program using 8086 and verify for:

a. Finding the largest number from an array.

b. Finding the smallest number from an array.

8. Write a program using 8086 for arranging an array of numbers in descending order and verify.

9. Write a program using 8086 for arranging an array of numbers in ascending order and verify.

10. Write a program to interface a two-digit number using seven-segment LEDs. Use 8085/8086 microprocessor and 8255 PPI.

Course Outcomes (COs)

Upon successful completion of the course, the student will be able to:

CO1. Do assembly language programming of 8085/ 8086.

CO2. Do assembly language programming of 8086/8085 for interfacing of peripherals.

CO-PSO Mapping

	PSO1	PSO2	PSO3	PSO4
CO1	2	2	3	2
CO2	2	2	3	2
Average	**2**	**2**	**3**	**2**

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	2	2	1	1		2		1	2	1
CO2	1	1	2	1	2	3				3	2	2
Average	**2**	**1**	**2**	**1.5**	**1.5**	**2**		**1**		**2**	**2**	**1.5**

Course Overview

The Microprocessors Lab course is designed to provide students with hands-on experience in working with microprocessors and microcontrollers and to enhance their understanding of embedded systems. This lab-based course complements the theoretical knowledge gained in the Microprocessors theory course, allowing students to apply their knowledge in practical applications. The course typically covers the topics: Introduction, overview of microprocessors and their applications. It gives differentiating information between microprocessors and microcontrollers based on their characteristics and applications. It gives information about writing assembly language programs to perform basic arithmetic operations and control flow. Practicing instruction sets, addressing mode s, and assembly language programming techniques specific to the microprocessors being used.

It gives information about interfacing microprocessors with peripheral devices such as switches, LEDs, LCDs, and sensors. Implementing different communication protocols (UART, SPI, I2C) to establish communication with external devices. Configuring and utilizing timers, interrupts, and other peripheral modules for efficient device control and data transfer.

.

List of Experiments mapped with COs

S. No.	Name of the Experiments	Course Outcome	Page No.
1	Write a program using 8085 and verify for: a). Addition of two 8-bit numbers. b). Addition of two 8-bit numbers (with carry).	CO1	1 3
2	Write a program using 8085 and verify for: a). 8-bit subtraction (display borrow) b).16-bit subtraction (display borrow)	CO1	6 8
3	3. Write a program using 8085 for multiplication of two 8- bit numbers by repeated addition method. Check for minimum number of additions and test for typical data.	CO1	10
4	Write a program using 8085 for multiplication of two 8-bit numbers by bit rotation method and verify.	CO1	12
5	Write a program using 8086 for finding the square root of a given number and verify.	CO1	14
6	Write a program using 8086 for copying 12 bytes of data from source to destination and verify.	CO1	16
7	Write a program using 8086 and verify for: Finding the largest number from an array.	CO1	18
8	Write a program using 8086 and verify for: Finding the smallest number from an array.	CO1	20
9	Write a program using 8086 for arranging an array of numbers in descending order and verify.	CO1	22
10	. Write a program using 8086 for arranging an array of numbers in ascending order and verify.	CO2	25

DOs and DON'Ts

DOs

1. Do follow the lab rules and guidelines provided by your instructor or lab supervisor.
2. Do wear appropriate safety gear, such as safety glasses or gloves, when working with electronic components or soldering equipment.
3. Do handle microprocessors and other electronic components with care to avoid damage.
4. Do organize your workspace and keep it clean and tidy to prevent accidents and confusion.
5. Do double-check your connections and circuit configurations before applying power to avoid short circuits or other electrical issues.
6. Do ask questions and seek clarification from your instructor or lab assistants if you are unsure about any aspect of the lab experiment or procedure.
7. Do backup your work regularly, especially if you are working on programming or design projects.
8. Do document your work, including circuit diagrams, code snippets, and observations, for future reference and troubleshooting.
9. Do collaborate and engage in discussions with your lab partners or classmates, as it can enhance your understanding and problem-solving skills.
10. Do explore additional resources, such as textbooks, online tutorials, or datasheets, to deepen your knowledge of microprocessors and related concepts.

DON'Ts

1. Don't eat, drink, or smoke in the lab, as it can pose a safety hazard and damage the equipment.
2. Don't leave your workspace unattended while your experiment is running or when equipment is powered on.
3. Don't attempt to repair or modify equipment without proper authorization and guidance.
4. Don't rush or take shortcuts in your experiments or projects. Take your time to understand the concepts and follow the correct procedures.
5. Don't ignore warning signs, such as abnormal smells, smoke, or overheating of equipment. Report any issues to your instructor or lab supervisor immediately.
6. Don't rely solely on your lab partners' work or solutions. Each student should actively participate and contribute to the lab activities.
7. Don't hesitate to seek help if you encounter difficulties or challenges during the lab. Your instructor or lab assistants are there to assist you.

General Safety Precautions

Precautions (In case of Injury or Electric Shock)

1. To break the victim with live electric source, use an insulator such as fire wood or plastic to break the contact. Do not touch the victim with bare hands to avoid the risk of electrifying yourself.

2. Unplug the risk of faulty equipment. If main circuit breaker is accessible, turn the circuit off.

3. If the victim is unconscious, start resuscitation immediately, use your hands to press the chest in and out to continue breathing function. Use mouth-to-mouth resuscitation if necessary.

4. Immediately call medical emergency and security. Remember! Time is critical; be best.

Precautions (In case of Fire)

1. Turn the equipment off. If power switch is not immediately accessible, take plug off.

2. If fire continues, try to curb the fire, if possible, by using the fire extinguisher or by covering it with a heavy cloth if possible isolate the burning equipment from the other surrounding equipment.

3. Sound the fire alarm by activating the nearest alarm switch located in the hallway.

4. Call security and emergency department immediately:

Emergency: Reception

Security : Main Gate

Guidelines to students for report preparation

All students are required to maintain a record of the experiments conducted by them. Guidelines for its preparation are as follows: -

1) All files must contain a title page followed by an index page. **The files will not be signed by the faculty without an entry in the index page.**

2) Student's Name, Roll number and date of conduction of experiment must be written on all pages.

3) For each experiment, the record must contain the following

(i) Aim/Objective of the experiment

(ii) Apparatus required with specification and Name plate details

(iii) Program, flowchart, memory locations.

(v) Results/ output

Note:

1. Students must bring their lab record along with them whenever they come for the lab.

2. Students must ensure that their lab record is regularly evaluated.

Lab Assessment Criteria

An estimated 10 lab classes are conducted in a semester for each lab course. These lab classes are assessed continuously. Each lab experiment is evaluated based on 5 assessment criteria as shown in following table. Assessed performance in each experiment is used to compute CO attainment as well as internal marks in the lab course.

Grading Criteria	Exemplary (4)	Competent (3)	Needs Improvement (2)	Poor (1)
AC1: Pre-Lab written work (this may be assessed through viva)	Complete procedure with underlined concept is properly written	Underlined concept is written but procedure is incomplete	Not able to write concept and procedure	Underlined concept is not clearly understood
AC2: Flow Chart / Connection	Flow chart must be neatly drawn and / program with logic should be sequencely written.	Flow chart drawn and program written but logic not appropriate.	program and flow chart to be given as per directions.	Unable to draw flow chart connection as per program.
AC3: Identification of problems in running the equipment and note down the reading	Able to identify the errors while running the machine code and note down the execution step wise.	Able to identify the errors while running the machine code and whole execution observed.	Only few steps of program are exectuted	Unable to identify the program.
AC4: Final Demonstration and Execution	All steps for execution of program are measured, experiment is well demonstrated and implemented concept is clearly explained	All steps for execution of program are not measured, experiment is demonstrated and implemented concept is clearly explained	Only few steps are measured, experiment is demonstrated and implemented concept is not clearly explained	Not well executed and not explained the concept
AC5: Lab Record Assessment	All the instructions are properly written and mathematical operations properly executed and performance analysis- results are shown with memory locations.	70 % calculations are done results.	Less than 70 % instructions are done results are executed.	Not completed

LAB EXPERIMENTS

LAB EXPERIMENT 1(a)

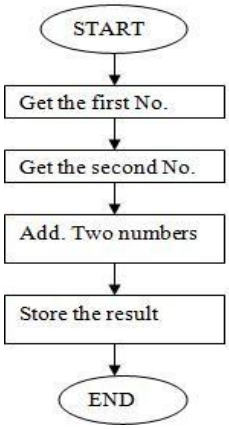
AIM : write a program using 8085 & verify for : Addition of two 8-bit numbers.

APPARATUS : 8085 microprocessor kit, 5V power supply, Keyboard.

THEORY (Program)

Memory address	Machine code	Mnemonics	Operands	Commands
7000	21,01,75	LXI	H,7501	Get address of 1st no. in HL pair
7003	7E	MOV	A,M	Move Ist no. in accumulator
7004	23	INX	H	HL points the address 7502H
7005	86	ADD	M	Add the 2nd no.
7006	23	INX	H	HL points 7503H
7007	77	MOV	M,A	Store result in 7503H.
7008	EF	RST 5		Terminate

FLOW CHART:



INPUT DATA

7501- 13H

7502- 12H

OUTPUT DATA

7503- 25H

PRECAUTIONS:-

1. Make sure that all the machine codes should be as per specified in the program.

2. Switch OFF the setup/ system when note in use.

EXPERIMENT NO. 1(b)

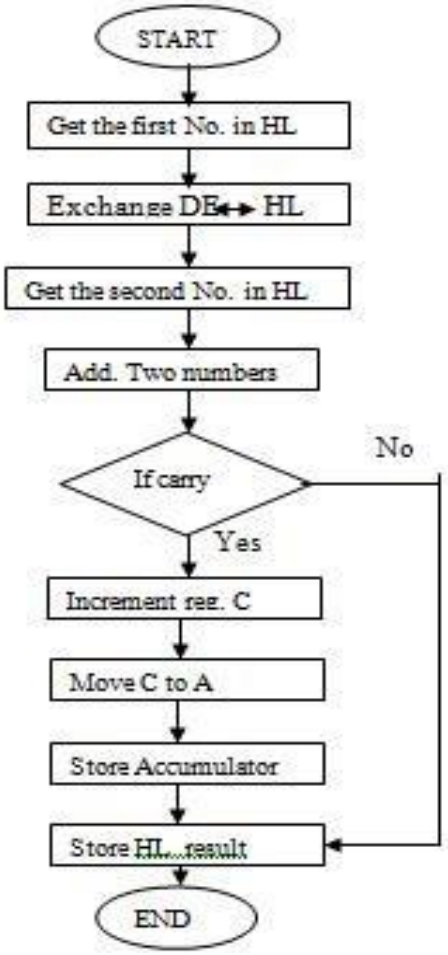
AIM : write a program using 8085 & verify for : Addition of two 16-bit numbers (with carry).

APPARATUS : 8085 microprocessor kit, 5V power supply, Keyboard.

THEORY (Program)

Memory address	Label	Machine code	Mnemonics	Operands	Commands
7000		2A,01,76	LHLD	7601H	Get 1st no. in HL pair from memory (7601)
7003		EB	XCHG		Exchange cont. of DE◆ HL
7004		2A,03,76	LHLD	7603H	Get 2st no. in HL pair from location 7603
7007		0E,00	MVI	C,00H	Clear reg. C.
7009		19	DAD	D	Get HL+DE & store result in HL
700A		D2,12,70	JNC	7012(loop)	If no carry move to loop/if carry then move to next step.
700D		0C	INR	C	Increment reg C
700E		79	MOV	A,C	Move carry from reg. C to reg. A
7011		32,02,75	STA	7502	Store carry at 7502H
7012	loop	22,00,75	SHLD	7500	Store result in 7500H.
7015		EF	RST 5		Terminate

FLOW CHART:-



INPUT DATA

7601 : 13H

7602 : 31H

7603 : 12H

7604 : 10H

OUTPUT DATA

7500 : 25H
7501 : 41H

7502 : 00H

PRECAUTIONS:-

1. Make sure that all the machine codes should be as per specified in the program.

2. Switch OFF the setup when note in use.

EXPERIMENT NO. 2(a)

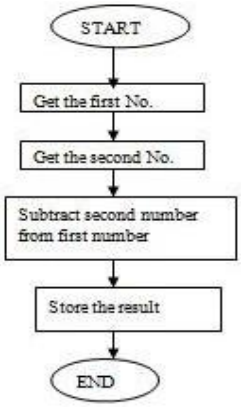
AIM : Write a program using 8085 & verify for : Subtraction of two 8-bit numbers. (display of barrow).

APPARATUS : 8085 microprocessor kit, 5V power supply, Keyboard.

THEORY(Program) :

Memory address	Opcode	Mnemonics	Operands	Comments
7000	21,01,75	LXI	H, 7501	Get address of ist no. in HL pair
7003	7E	MOV	A, M	Move Ist no. in accumulator
7004	23	INX	H	HL points 7502H.
7005	96	SUB	M	Substract 2nd no. from Ist no.
7006	23	INX	H	HL points 7503 H.
7007	77	MOV	M, A	Move contents of acc. to memory
7008	CF	RST 1		Stop

FLOW CHART :-



INPUT DATA

7501 : 20H

7502 : 10H

OUTPUT DATA

7503 : 10H

PRECAUTIONS:-

1. Make sure that all the machine codes should be as per specified in the program.

2. Switch OFF the setup when note in use.

EXPERIMENT NO. 2 (b)

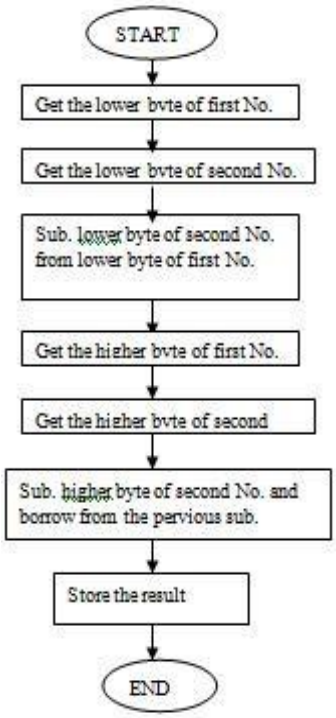
AIM : Write a program using 8085 & verify for : Subtraction of two 16-bit numbers. (display of barrow)

APPARATUS : 8085 microprocessor kit, 5V power supply, Keyboard.

THEORY (Program) :

Memory Address	Machine Code	Mnemonics	Operands	Comments
7000	2A, 01,75	LHLD	7501 H	Get 1st 16 bit no. in HL pair
7003	EB	XCHG		Exchange HL pair with DE.
7004	2A, 03,75	LHLD	7503 H	Get 2nd 16 bit no. in HL pair
7007	7B	MOV	A, E	Get lower byte of ist no.
7008	95	SUB	L	Subtract lower byte of 2nd no.
7009	6F	MOV	L, A	Store the result in reg. L
700A	7A	MOV	A, D	Get higher byte of Ist no.
700B	96	SBB	H	Subtract higher byte of 2nd no. with borrow
700C	67	MOV	H,A	Move from acc. To H
700D,E, F	22,05,75	SHLD	7505H	Store 16 bit result at 7505&7506
7010	EF	RST 5		Terminate

FLOW CHART:



INPUT DATA

7501 : 30H

7502 : 40H

7503 : 10H

7504 : 20H

OUTPUT DATA

7505 : 20H

7506 : 20H

PRECAUTIONS:-

1. Make sure that all the machine codes should be as per specified in the program.

2. Switch OFF the setup when note in use.

EXPERIMENT NO. 3

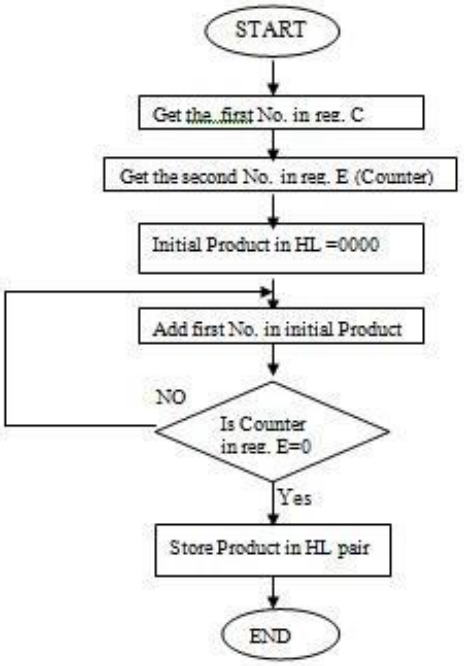
AIM : Write a program using 8085 for multiplication of two 8-bit numbers by repeated addition method check minimum number of addition & test for typical data.

APPARATUS : 8085 microprocessor kit, 5V power supply, Keyboard.

THEORY (Program) :

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
7000		0E,25	MVI	C,25	Move the no. in reg. C
7002		1E,05	MVI	E,05	Move the no. in reg. E
7004		06,00	MVI	B,00	Clear reg. B
7006		21,00,00	LXI	H,0000	Initial Product=0000
7009	UP1:	09	DAD	B	HL+BC=>HL
700A		1D	DCR	E	Decrement reg. E
700B		C2,09,70	JNZ	UP1(7009)	Jump if not zero to location up1
700E		22,00,75	SHLD	7500	Store HL at 7500
7011		CF	RST 1		Terminate

FLOW CHART:



INPUT DATA

Reg.C : 25H

Reg.E : 05H

Reg.B : 00H

OUTPUT DATA

HL pair : 00B9H

PRECAUTIONS:-

1. Make sure that all the machine codes should be as per specified in the program.
2. Switch OFF the setup when note in use

EXPERIMENT NO: 4

AIM: Write a program using 8085 for multiplication of two 8- bit numbers by bit rotation method and verify**.**

APPARATUS : 8085 microprocessor kit, 5V power supply, Keyboard.

Program:

Address	HEX Codes	Labels	Mnemonics	Comments
F000	21, 00, 80		LXI H,8000H	Point to first operand
F003	5E		MOV E,M	Load the first operand to E
F004	16, 00		MVI D,00H	Clear the register D
F006	23		INX H	Point to next location
F007	7E		MOV A,M	Get the next operand
F008	0E, 08		MVI C,08H	Initialize counter with 08H
F00A	21, 00, 00		LXI H, 0000H	Clear the HL pair
F00D	0F	LOOP	RRC	Rotate the acc content to right
F00E	D2, 12, F0		JNC SKIP	If carry flag is 0, jump to skip
F011	19		DAD D	Add DE with HL
F012	EB	SKIP	XCHG	Exchange DE and HL
F013	29		DAD H	Add HL with HL itself
F014	EB		XCHG	Exchange again the contents of DE and HL
F015	0D		DCR C	Decrease C register
F016	C2, 0D, F0		JNZ LOOP	if Z = 0, jump to LOOP
F019	22, 50, 80		SHLD 8050H	Store the result
F01C	76		HLT	Terminate the program

INPUT DATA:

8000: 25

8001: 2A

OUTPUT DATA

8051: 12

8050: 06

PRECAUTIONS:-

1. Make sure that all the machine codes should be as per specified in the program.
2. Switch OFF the setup when note in use

EXPERIMENT NO: 5

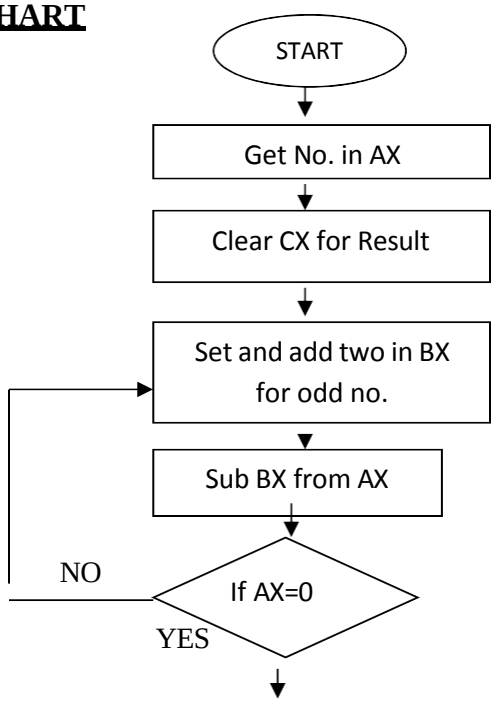
AIM: Write a program using 8086 for finding the square root of a given number and Verify.

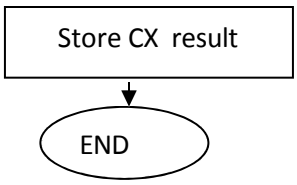
APPARATUS REQUIRED:- 8086 Microprocessor Kit (NV5586A), Power Supply, Keyboard.

THEORY/PROGRAM:-

CS	IP	Machine code	Mnemonics	Comments
0000	0400	8B,06,00,05	MOV AX,[500]	Move the No. from memory to AX
0000	0404	B9,00,00	MOV CX,0000 H	Initialize the counter
0000	0407	BB,FF,FF	MOV BX,FFFF H	Set all 16 bits one in BX register
0000	040A	81,C3,02,00 UP:	ADD BX,02	Add two to get the odd No.
0000	040E	41	INC CX	Increment CX by one
0000	040F	29,D8	SUB AX,BX	Sub BX from AX.
0000	0411	75,F7	JNZ UP	Jmp if AX is not zero
0000	0413	89,0E,00,06	MOV [600],CX	Move counter value to location
0000	0417	F4	HLT	

FLOW CHART





RESULT :

INPUT DATA	**OUTPUT DATA**
0500 : 09 H	0600 : 03 H

PRECAUTIONS:-

1. Make sure that all the machine codes should be as per specified in the program.
2. Switch OFF the setup when note in use

EXPERIMENT NO: 6

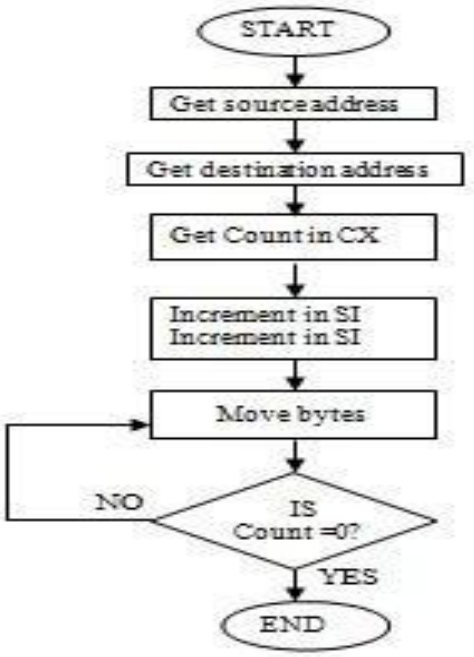
AIM: Write a program using 8086 for copying 12 bytes of data from source to destination & verify .

APPARATUS REQUIRED:- 8086 Microprocessor Kit (NV5586A), Power Supply, Keyboard.

THEORY/PROGRAM:-

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
0101		FC	CLD		Clear direction flag DF
0102		BE,00,03	MOV	SI,0300	Source address in SI
0105		BF,02,02	MOV	DI,0202	Destination address in DI
0108		8B,0C	MOV	CX,[SI]	Count in CX
010A		46	INC	SI	Increment SI
010B		46	INC	SI	Increment SI
010C	BACK	A4	MOV	SB	Move byte
010D		E2,FD	LOOP	BACK	Jump to BACK until CX =0
010F		CC	INT		Interrupt program

FLOWCHART:



RESULT :

INPUT DATA

0300 : 0B
0301 : 00
0302 : 03
0303 : 04
0304 : 05
0305 : 06
0306 : 15
0307 : 07
0308 : 12
0309 : 08
030A : 09
030B : 0A
030C : 0B
030D : 0E

OUTPUT DATA

0202 : 03
0203 : 04
0204 : 05
0205 : 06
0206 : 15
0207 : 07
0208 : 12
0209 : 08
020A : 09
020B : 0A
020C : 0B
020D : 0E

PRECAUTIONS:-

1. Make sure that all the machine codes should be as per specified in the program.
2. Switch OFF the setup when note in use

EXPERIMENT NO: 7

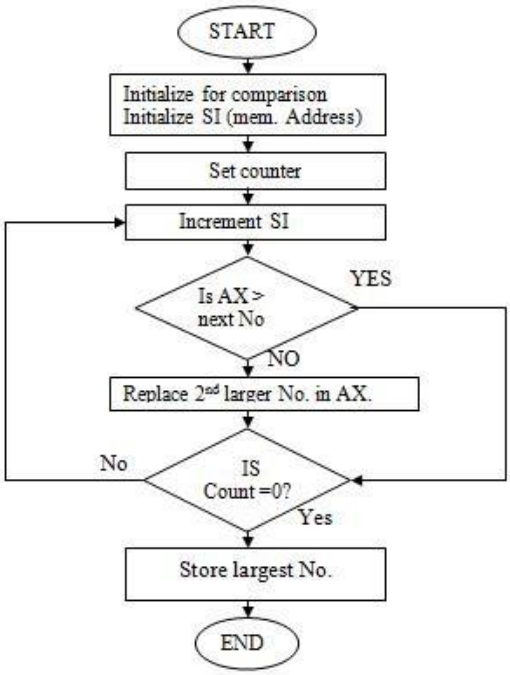
AIM: Write a program using 8086 & verify for finding the largest number from an data array.

APPARATUS REQUIRED:- 8086 Microprocessor Kit (NV5586A), Power Supply, Keyboard.

THEORY/PROGRAM:-

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
0101		B0,00,00	MOV	AX,0000	Initial value for comparison
0104		BE,00,02	MOV	SI,0200	Memory address in SI
0107		8B,0C	MOV	CX,[SI]	Count in CX
0109	BACK	46	INC	SI	Increment SI
010A		46	INC	SI	Increment SI
010B		3B,04	CMP	AX,[SI]	Compare previous largest number with next number
010D		73,02	JAE	GO	Jump if number in AX is larger i.eCF=0
010F		8B,04	MOV	AX,[SI]	Save next larger number in AX
0111	GO	E2,F6	LOOP	BACK	Jump to BACK until CX becomes zero
0113		A3,51,02	MOV	(0251),AX	Store largest number in memory
0116			HLT/INT 5		Interrupt program

FLOW CHART:



INPUT DATA

0200 : 05 H
0201 : 00 H
0202 : 41 H
0203 : 83 H
0204 : 58 H
0205 : 72 H
0206 : 39 H
0207 : 46 H
0208 : 53 H
0209 : 84 H
020A : 30 H
020B : 96 H

OUTPUT DATA

251 : 30 H
252 : 96 H

PRECAUTIONS:-

1. Make sure that all the machine codes should be as per specified in the program.
2. Switch OFF the setup when note in use

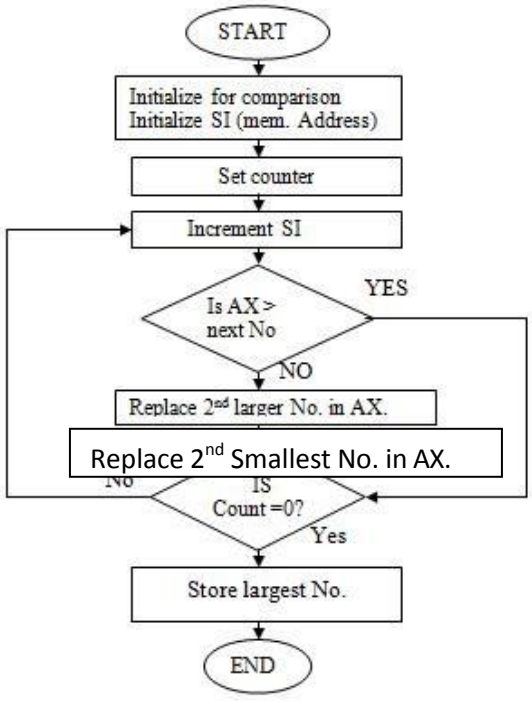
EXPERIMENT NO: 8

AIM: Write a program using 8086 & verify for finding the smallest number from an data array.

APPARATUS REQUIRED:- 8086 Microprocessor Kit (NV5586A), Power Supply, Keyboard.

THEORY/PROGRAM:-

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
0101		B0,00,00	MOV	AX,0000	Initial value for comparison
0104		BE,00,02	MOV	SI,0200	Memory address in SI
0107		8B,0C	MOV	CX,[SI]	Count in CX
0109	BACK	46	INC	SI	Increment SI
010A		46	INC	SI	Increment SI
010B		3B,04	CMP	AX,[SI]	Compare previous smallest number with next number
010D		73,02	JBE	GO	Jump if number in AX is larger i.eCF=0
010F		8B,04	MOV	AX,[SI]	Save next smaller number in AX
0111	GO	E2,F6	LOOP	BACK	Jump to BACK until CX becomes zero
0113		A3,51,02	MOV	(0251),AX	Store smallest number in memory
0116			HLT/INT 5		Interrupt program



FLOW CHART

INPUT DATA

0200 : 05 H
0201 : 00 H
0202 : 41 H
0203 : 83 H
0204 : 58 H
0205 : 72 H
0206 : 39 H
0207 : 46 H
0208 : 53 H
0209 : 84 H
020A : 30 H
020B : 96 H

OUTPUT DATA

251 : 39 H
252 : 46 H

PRECAUTIONS:-

1. Make sure that all the machine codes should be as per specified in the program.
2. Switch OFF the setup when note in use

EXPERIMENT NO: 9

AIM : Write a program using 8086 for arranging an array of numbers in descending order.

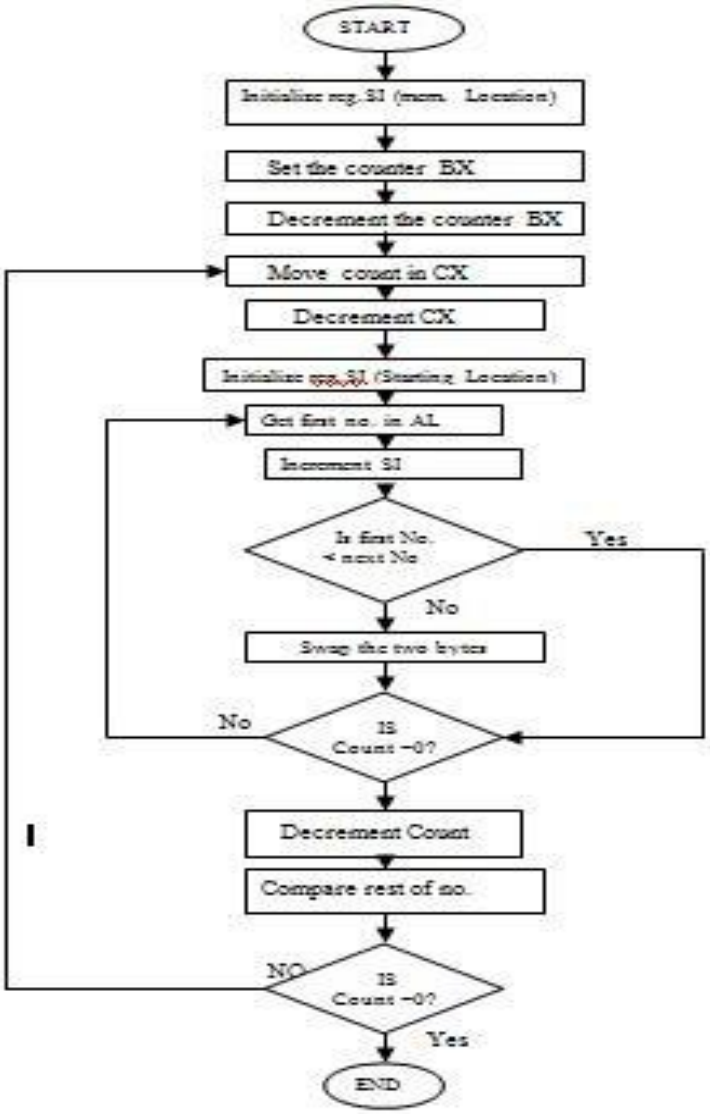
APPARATUS : 8086 microprocessor kit, 5V power supply, Keyboard.

THEORY/PROGRAM

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
0200		BE,00,03	MOV	SI,0300	Initialize SI Reg. with Memory Location. 0300.
0203		8B,1C	MOV	BX,[SI]	BX has no. of bytes
0205		4B	DEC	BX	Decrement the no. of bytes by one
0206	(3)	8B 0C	MOV	CX (SI)	Move no. of bytes in CX
0208		49	DEC	CX	Decrement the no. of bytes by one
0209		BE,02,03	MOV	SI,0303	Initialize SI reg. with the starting address of string
020C	(2)	8A,04	MOV	AL,[SI]	Move first data byte of string into AL
020E		46	INC	SI	Point at the next bytes of the string
020F		3A,04	COMP	AL,[SI]	Com. the two bytes of string.
0211		73,06	JAE	(1)	If two bytes are equal or 1st byte is above that the second byte branch to (1)
0213		86,04	XCHG	AL,[SI]	Else
0215		4E	DEC	SI	Second byte is less than first byte and swap the two bytes.
0216		88,04	MOV	[SI],AL	
0218		46	INC	SI	Point at next location of string
0219	(1)	E2,F1	LOOP	(2)	Loop if CX is not zero
021B		4B	DEC	BX	

021C		BE,00,03	MOV	SI,0300	
021F		75,E5	JNZ	(3)	
0221		F4	HLT		Halt.

FLOW CHART:



INPUT DATA

0300 : 05
0301 : 00

0302 : 20
0303 : 25
0304 : 28
0305 : 15
0306 : 07

OUTPUT DATA

0302 : 28
0303 : 25
0304 : 20
0305 : 15
0306 : 07

PRECAUTIONS:-

1. Make sure that all the machine codes should be as per specified in the program.
2. Switch OFF the setup when note in use

EXPERIMENT NO: 10

AIM : Write a program using 8086 for arranging an array of numbers in Aescending order.

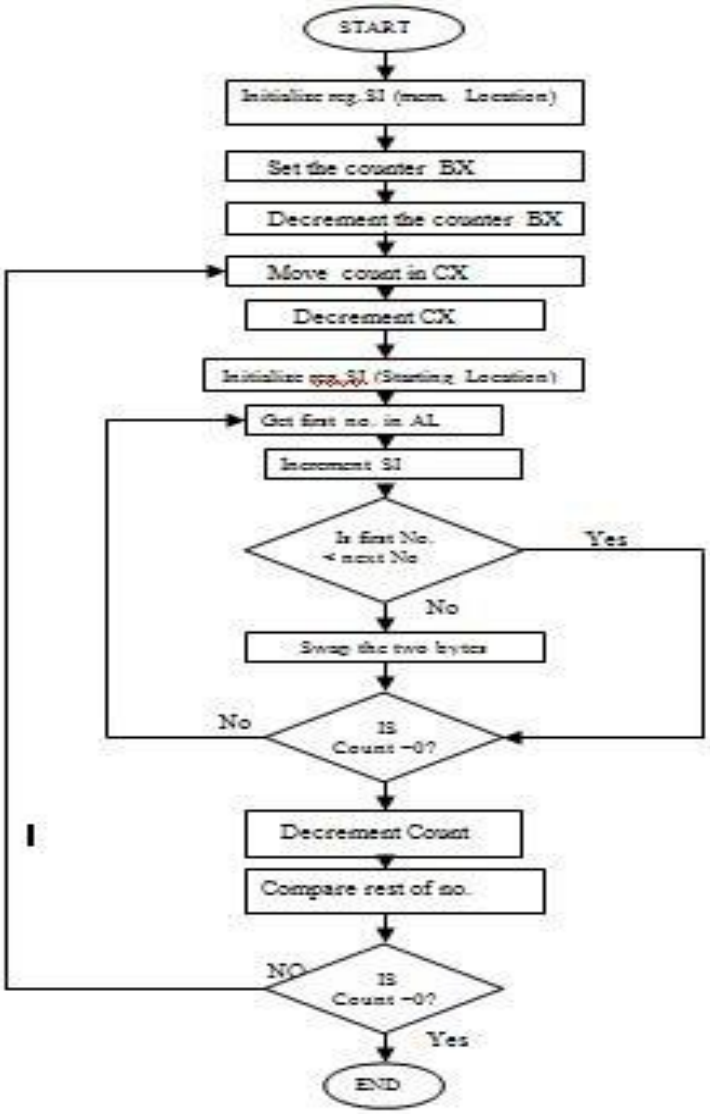
APPARATUS : 8086 microprocessor kit, 5V power supply, Keyboard.

THEORY/PROGRAM

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
0200		BE,00,03	MOV	SI,0300	Initialize SI Reg. with Memory Location. 0300.
0203		8B,1C	MOV	BX,[SI]	BX has no. of bytes
0205		4B	DEC	BX	Decrement the no. of bytes by one
0206	(3)	8B 0C	MOV	CX (SI)	Move no. of bytes in CX
0208		49	DEC	CX	Decrement the no. of bytes by one
0209		BE,02,03	MOV	SI,0303	Initialize SI reg. with the starting address of string
020C	(2)	8A,04	MOV	AL,[SI]	Move first data byte of string into AL
020E		46	INC	SI	Point at the next bytes of the string
020F		3A,04	COMP	AL,[SI]	Com. the two bytes of string.
0211		73,06	JBE	(1)	If two bytes are equal or 1st byte is above that the second byte branch to (1)
0213		86,04	XCHG	AL,[SI]	Else
0215		4E	DEC	SI	Second byte is less than first byte and swap the two bytes.
0216		88,04	MOV	[SI],AL	
0218		46	INC	SI	Point at next location of string
0219	(1)	E2,F1	LOOP	(2)	Loop if CX is not zero
021B		4B	DEC	BX	

021C		BE,00,03	MOV	SI,0300	
021F		75,E5	JZ	(3)	
0221		F4	HLT		Halt.

FLOW CHART:



INPUT DATA

0300 : 05
0301 : 00

0302 : 20
0303 : 25
0304 : 28
0305 : 15
0306 : 07

OUTPUT DATA

0302 : 07
0303 : 15
0304 : 20
0305 : 25
0306 : 28

PRECAUTIONS:-

1. Make sure that all the machine codes should be as per specified in the program.
2. Switch OFF the setup when note in use

Viva-Voce Questions

1. Expand PPI?
2. Where do we prefer the serial communication?
3. What is the function of instruction pointer (IP) register?
4. What is the difference between IN and OUT instructions?
5. What is MODEM?
6. What sort of pipelining concept is available in 8086/8088 µp?
7. What is the purpose of segment registers in 8086/8088 µp?
8. How physical address is generated in case of 8086/8088 µp?
9. Explain segment overlapping. What is the use of LOCK' signal in 8086?
10. What do you know about relocatability of program?
11. Explain Division instruction of 8086 µp?
12. Define any three directives of 8086 assembler.
13. What is difference between LOOP, REP and JMP in µp 8086?
14. What are the utility of conditional and control flags in µp 8086 & name them in above category?
15. What is the difference between Real & Protected mode of ×86 family µp's?
16. How µp 8086 reads or write from or in to memory 2 bytes of data in one cycle & at what address it check from memory whether even or odd for this?

SHORT Questions/ Answers:-

Q.l Explain MOV R1, R2 ?

A.1 Copy the content of register to register

Q.2 How many machine cycles and T-states are in MOV instruction?

A.2 One Machine cycle for Even address (16-bit operation), while two for Odd address.

Q.3 Which of the flag/s is/are affected in MOV instruction?

A.3 No Flags affected.

Q.4 What is XCHG ?

A.4 Swap the contents of the registers/Memory location data.

Q.6 Explain the addressing mode of XCHG?

A.6 Register Direct/Indirect modes are possible.

Q.7 What is ADD R1, R2 ?

A.7 Add the contents at operand register/memory operand and save result at destination operand.

Q.8 What is ADC R1, R2 and how it is different from ADD instruction?

A.8 Add the contents at operand register/memory operand with carry flag and save result at destination operand.

This lab manual has been updated by

Prof Monika Rani

(monika.rani@ggnindia.dronacharya.info)

Cross checked by

HOD CSE