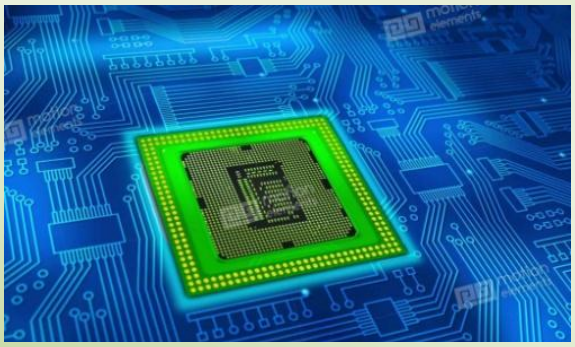
Lakshmi Narain College of Technology, Bhopal



Microcontroller & Embedded System (EC606)

Lab Manual

Department of Electronics & Communication Engineering

VISION AND MISSION OF DEPARTMENT

Vision of the Department:

To be recognized as Centre of Academic Excellence by imparting quality teaching and strengthening research and development activities with world class infrastructure in the field of Electronics and Communication Engineering.

.

Mission of the Department

- **M1** To establish a quality teaching learning process to provide application oriented, in-depth knowledge consistently.
- **M2** To establish state-of-the-art laboratories for academic excellence and to develop infrastructure through collaboration for quality research.
- **M3** To equip the students by blending theoretical knowledge and practical skills with employability and entrepreneurship traits for a bright successful career.
- **M4** To inculcate team spirit and leadership qualities to produce socially acceptable, eco-friendly and responsible citizens.

Program Educational Objectives (PEO's)

Students will be able to

- **PEO1** Apply knowledge of mathematics, science and engineering as appropriate in the field of Electronics & Communication Engineering as proficient learners in the domains such as Electronic Circuits, Embedded Systems, Communication Systems, Digital Signal Processing, VLSI Design, Data Networks, IOT, and Simulation etc.
- **PEO2** Seek admissions at Institutes of repute for higher education in Engineering & Technology and Management to the tune of 10%, seek employment in core and IT domains to the extent of 80% with remaining 10% opting for entrepreneurship.
- **PEO3** Use the skills, latest techniques, tools for modern engineering and ICT which are necessary to analyze industrial problems related to Electronics & Communication Engineering with focus to Global, Economical and Environmental Issues.
- **PEO4** Understand engineering solutions, exhibit professionalism, ethical attitude, team work, effective written and oral communication skills to practice in their profession with high regards to societal issues and responsibilities.

Program Specific Outcomes (PSO's)

Student will be able to:

- **PSO1** Apply basic concepts of science and engineering, to undertake theoretical learning of Electronic Devices and Circuits, Analog & Digital Communication, Signals & Systems, Embedded Systems, VLSI Design etc.
- **PSO2** Develop the ability to acquire hands-on skills such as Circuit Simulation, MATLAB, HDL Programming, Embedded Systems, DSP and PCB Designing etc.
- **PSO3** Develop team spirit and professional ethics to undertake research oriented projects, especially developmental projects and a few industry sponsored projects.
- **PSO4** Learn extra-curricular courses such as soft-skills, personality development, and groom them as responsible citizen with professional ethics blended with human values, engineering economics and ability to handle real life issues.

Course outcomes

- CO1. Apply arithmetic, logical and bit manipulation instructions of 8051 for programming.
- CO2. Simulate Serial Communication between 8051 kit and PC.
- CO3. Implement interface seven segment with 8051.
- CO4. Implement the interfacing of DAC and ADC with 8051.
- CO5. Verify the program of stepper motor and traffic light Controller.

Code of Conducts for the Laboratory

1. Switch off the equipment and disconnect the power supplies from the circuit before leaving the laboratory.

2. Do not install any software or delete any system files on any lab computers.

3. Switch on the supply, only after getting the circuit checked by the proper person guiding the experiment.

4. Equipment Failure - If a piece of equipment fails while being used, report it immediately to your lab instructor. Never try to fix the problem yourself because you could harm yourself and others.

5. If the trainer board catches fire, turned off the power and notify the instructor immediately. An electronic instrument catches fire but extinguish quickly after the power is shut off. Avoid using fire extinguishers on electronic instruments.

6. Read carefully all the instructions in the lab manual before conducting any experiment.

Rubrics for Assessment of student performance during Experiments

Area of Direct Assessment	Poor (0-2 Marks)	Fair (3-4 Marks)	Average (5-6 Marks)	Good (7-8 Marks)	Excellent (9-10 Marks)
Aim & Theory	Aim is not clear and irrelevant theory written Concept was not explained.	Aim is clear and Incomplete theory written. Concept could not be explained.	Aim is clear and Theory written but is unorganised Concept is explained.	Aim is clear and Theory written properly. Concept is explained.	Aim is clear and Theory written properly. Concept is explained with neat diagrams.
Performance and Working with Others	Did not conduct the experiment and none of the member recorded the observations.	Followed few steps to conduct the experiment. But few members recorded the observations.	Followed few steps to conduct the experiment. Few members recorded the observations.	Followed step by step method to conduct the experiment. Sufficient observations recorded by all team members.	Followed step by step method to conduct the experiment. Many observations recorded by all team members.
Safety Measures	None of the team member knew safety measures and did not followed.	Team members had knowledge of safety measures and followed few of them.	Team members had fair knowledge of safety measures and followed them.	Team members were well acquainted with safety measures and followed.	Team members were well acquainted with safety measures and followed all of them.
Result and Conclusion	No data recorded. Conclusion can not be drawn.	Analysis does not follow data the data. Conclusion can not be drawn.	Analysis as recorded somewhat lacks in insight. Results is poorly recorded to make sense. Conclusion can not be drawn.	Analysis as recorded somewhat lacks in insight. But clearly recorded as Results. Conclusion is properly drawn.	Observations are analysed accurately and clearly recorded as Results. Conclusion is properly drawn.
Observations and Calculations	No observations recorded and no calculation done.	Insufficient number of observations recorded. So calculations are Inaccurate.	Sufficient number of observations recorded but calculations are Inaccurate.	Almost all observations recorded. Calculations are accurate and well organised.	Many observations recorded in the table. Calculations are accurate and well organised.
Internal Viva	Student does not have grasp on the experiment and could not answer the questions about the experiment.	Student mumbles incorrectly, pronouns terms and speak too quietly for teachers to hear.	Student is uncomfortable but is able to answer basic questions about the experiment.	Student is at ease and able to answer expected questions, but fails to elaborate.	Student demonstrated full knowledge by answering all questions with explanations and elaboration.

Lakshmi Narain College of Technology, Bhopal

Enrollment No._____ Name of Student: _____

List Of Experiments

EC-606 MICROCONTROLLER AND EMBEDDED SYSTEM EXPERIMENTS

Exp. No.	Name of Experiments	Date of Experiment	Date of Submission	Remark
1	Write a program of arithmetic operations using 8051 microprocessor			
2	Assembly Language Program for finding largest no. from a given array of 8-bit numbers.			
3	Write an assembly language program to convert a HEX number to its equivalent ASCII code and display the result in the address field.			
4	To write an assembly language program to find the square root of a given data			
5	Transfer data serially between two kits			
6	Seven segment display			
7	Write a program to interface DAC with Microcontroller			
8	Write a program to interface ADC with Microcontroller			
9	Write a program to interface Stepper motor with Microcontroller			
10	Write a program for traffic light controller			

Date of Experiment: _____

Experiment No 1

Aim: To do the arithmetic operations using 8051 microprocessor

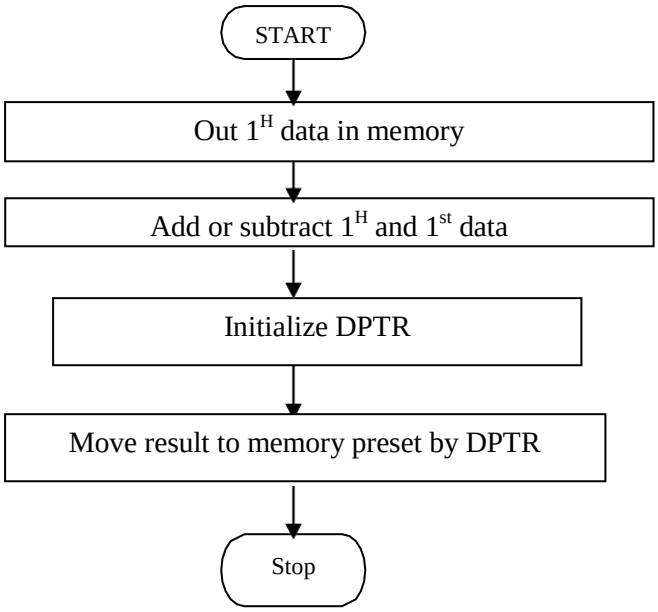
Apparatus required:

8085
microprocessor
kitDAC
interface kit
Keyboard

Algorithm:

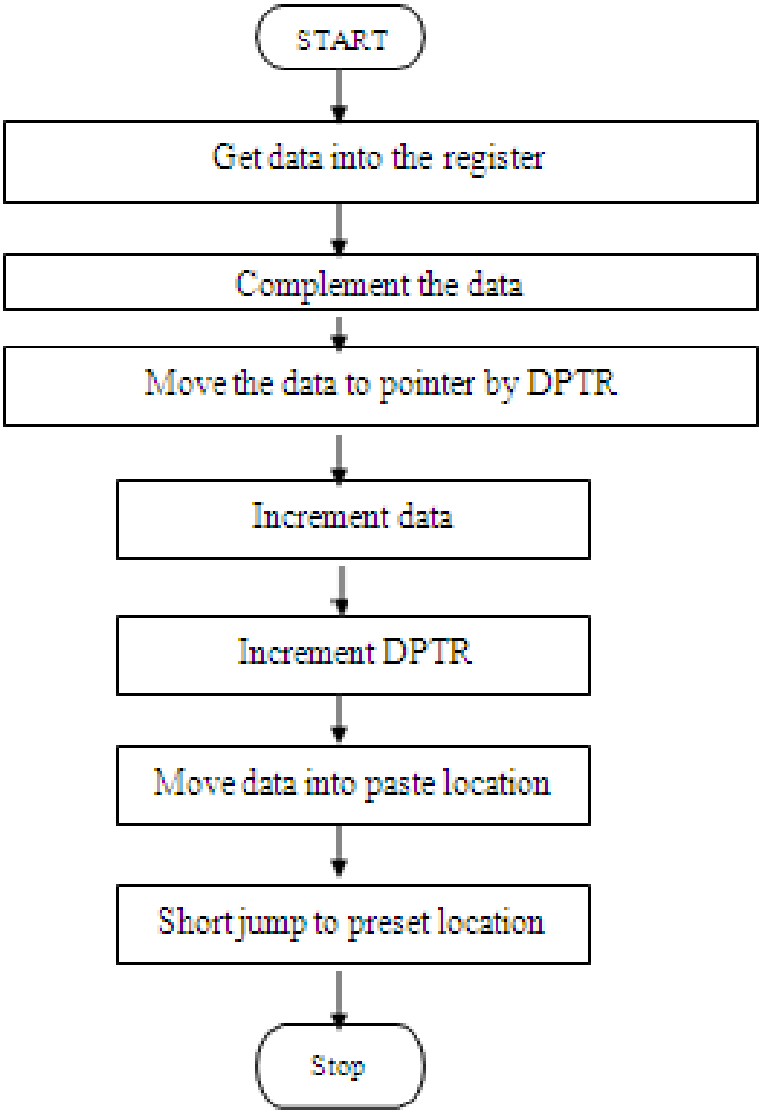
Addition / Subtraction

Step 1 : Move 1^H data to memory
Step 2 : Add or subtract 1^H data with
2^{nd} dataStep 3 : Initialize data pointer.
Step 4 : Move result to memory pointed by DPTR.



Multiplication / Division

Step 1 :Get 1^H data and 2^{nd} data to memory Step 2:Multiply or divide 1^H data with 2^{nd} dataStep : Initialize data pointer.
Step 4 : Move result to memory pointed by DPTR (first port Step 5 :Increment DPTR
Step 6 : Move 2^{nd} part of result to register A
Step 7 :Move result to 2^{nd} memory location pointer by DPTR



Program: 8-bit Addition:

Memory Location	Label	Opcode	Mnemonics	Comments
4100	Start	~~C3~~	~~CLR C~~	Clear the carry flat
4101		74DA	MOV A, #01	Moves data 1 to register A
4103		24DA	ADD A, #02	Add content of A and data 2 and store in A
4105		464500	MOV DPTR,#4500	Moves data 4500 to DPTR
4108		F0	MOVX @DPTR,A	Moves control of A to location pointed DTPR
4109		80 FE	$JMP 4109	Short jump to 4109

Execution:
Addition:

ML	Input
4103	
4109	

ML	Output
4500	

Program: 8-bit Subtraction

Memory Location	Label	Opcode	Mnemonics	Comments
4100	Start	~~C3~~	~~CLR C~~	Clear the carry flat
4101		74DA	MOV A,#05	Moves data 1 to register A
4103		24DA	SUBB A,#02	Subtract data 2 from content of A and store result in A
4105		464500	MOV DPTR,#4500	Moves 4500 to DPTR
4108		F0	MOVX @DPTR,A	Moves result by location by DTPR
4109		80 FE	SJMP 4109	Short jump to 4109

:

Execution:
Subtraction:

ML	Input
4101	
4103	

ML	Output
4500	

Program: 8-bit Multiplication:

Memory Location	Label	Opcode	Mnemonics	Comments
4100	Start	7403	MOV A,#03	Move immediate data to accumulator
4101		75F003	MOV B,#02	Move 2nd data to B register
4105		A4	MUL AB	Get the product in A & B
4106		904500	MOV DPTR, # 4500	Load data in 4500 location
4109 410A		F0	MOVX @DPTR,A INC DPTR	Move A t ext RAM
410B		E5F0	MOV A,B	Move 2nd data in A
410D		F0	MOVX @DPTR,A	Same the ext RAM
410E		80FE	SJMP 410E	Remain idle in infinite loop

Execution:
Multiplication:

ML	Input
4101	
4103	

Output Address	Value
4500	

Program: 8-bit Division:

Memory Location	Label	Opcode	Mnemonics	Comments
4100	Start	7408	MOV A,#04	Move immediate data to accumulator
4102		75F002	MOV B,#02	Move immediate to B reg.
4105		84	DIV AB	Divide content of A & B
4106		904500	MOV DPTR, # 4500	Load data pointer with 4500 location
4109		F0	MOVX @DPTR,A	Move A to ext RAM
410A		A3	INC DPTR	Increment data pointer
410B		ESF0	MOV A,B	Move remainder to A
410D		F0	MOVX @DPTR,A	Move A to ext RAM
410E		80FE	SJMP 410E	Remain idle in infinite loop

Execution:

Division:

ML	Input
4101	
4103	

Output Address	Value
4500	

Result:

Thus 8-bit addition, subtraction, multiplication and division is performed using 8051.

Date of Experiment: _____

Experiment No 2

Aim: Write an assembly language program to find the biggest number in an array of 8-bit unsigned numbers of predetermined length.

Apparatus required:

8051 microcontroller
kit(0-5V) DC battery

Algorithm:

1. Initialize pointer and counter.
2. Load internal memory location 40H as zero.
3. Move the first element of an array to r5 register.
4. Compare the data stored in memory location 40H is equal to or less than the value of first element of an array.
5. If it is lesser, then move the data of first element to 40H memory location ELSE increment pointer and decrement counter.
6. Check the counter. If counter is not equal to zero, repeat from the 2nd step else Move the R5 register to 40H memory location.
7. Stop the program.

Program:

Memory Location	Label	Opcode	Mnemonics
4100		90 42 00	MOV DPTR,#4200H
4103		75 40 00	MOV 40H,#00H
4106		7D 0A	MOV R5,#0AH
4108	LOOP2:	E0	MOVX A,@DPTR
4109		B5 40 08	CJNE A,40H,LOOP1
410C	LOOP 3	A3	INC DPTR
410D		DD F9	DJNZ R5,LOOP2
410F		E5 40	MOV A,40H
4111		F0	MOVX @DPTR,A

4112	HLT	80 FE	SJMP HLT

SAMPLE INPUT AND OUTPUT:

4114	LOOP1	40 F6	JC LOOP3
4116		F5 40	MOV 40H,A
4118		80 F2	SJMP LOOP3

INPUT:

Memory address	**Data**
4200	

OUTPUT:

Memory address	**Data**

RESULT:

Thus the assembly language program was written to find the largest element in anarray and executed using 8051 microcontroller

Date of Experiment: _____

Experiment No 3

Aim: Write an assembly language program to convert a binary number to its equivalent ASCII code and display the result in the address field.

Apparatus required:

8051 microcontroller kit
(0-5V) DC battery

Algorithm:

1. Get the decimal number in the range 00 to 99 as input
2. Separate the higher and lower nibble of the two digit number
3. Add 30h to the lower nibble and store the result
4. Bring the higher nibble to the ones position, add 30h to it and display the result.

Program:

Memory Location	Label	Opcode	Mnemonics	Comments
4100		90 42 00	MOV DPTR,#4200H	Input a Hex Value
4103		E0	MOVX A, @DPTR	
4104		F8	MOV R0,A	
4105		94 0A	SUBB A, #0AH	Compare Value 0-9
4107		50 05	JNC LOOP1	Values A-F go to Loop 1
4109		E8	MOV A,R0	
410A		24 30	ADD A,#30H	0-9 Add 30H
410C		80 03	SJMP LOOP	
410E	LOOP 1	E8	MOV A, RO	
410F		24 37	ADD A, #37H	A-F Add 37H
4111	LOOP	90 45 00	MOV DPTR, #4500H	
4114		F0	MOVX @DPTR, A	ASCII Value Output

4115		80 FE	SJMP 4115	

SAMPLE INPUT AND OUTPUT:

INPUT:

Memory address	Data
4200	Hex Data=

OUTPUT:

Memory address	Data
4500	ASCII Data=

Result:

Thus the assembly language program was written to converter Hexadecimal number to equivalent ASCII Code and executed using 8051 microcontroller.

Date of Experiment:_____

Experiment No 4

Aim: To write an assembly language program to find the square root of a given data

Apparatus required:

8051 microcontroller kit

(0-5V) DC battery

Algorithm:

1. Enter a program.
2. Enter the input hex value to location 4200h.
3. Execute the program.
4. The output square root value stored in a location 4500h.

PROGRAM:

Memory Location	Label	Opcode	Mnemonics	Comments
4100	Origin:	90 42 00	MOV DPTR,#4200h	Get a input data
4103		e0	MOVX A,@DPTR	
4104		f9	MOV R1,a	
4105		7a 01	MOV R2, #01h	Initialize counter
4107	LOOPI:	e9	MOV A,R1	
4108		8a f0	MOV B,R2	
410a		84	DIV AB	divide the given value and counter
410b		fb	MOV R3,A	
410c		ac f0	MOV R4,B	
410e		9a	SUBB A ,R2	compare
410f		60 03	JZ RESULT	Dividend and counter
4111		0a	INC R2	
4112		80 f3	SJMP L1	

SAMPLE INPUT AND OUTPUT:

ML	Input
4200	40(hex value)=64(decimal)

ML	Output
4500	8

Result:

Thus an assembly language program is written to find the square root of a given data and executed successfully

Date of Experiment___

Experiment 5

Aim: To write an assembly language program Transmitting and Receiving the data between twokits.

Apparatus required:

8051 microcontroller kit

(0-5V) DC battery

Algorithm:

1. Initialize TMOD with 20H
2. Set the values for TCON and SCON
3. Set the input address to DPTR
4. Based on the bit value on SCON store the data in SBUF
5. Increment DPTR and check for the loop end value

PROGRAM FOR RECEIVER.

Memory Location	Label	Opcode	Mnemonics	Comments
4100		75 89 20	MOV TMOD, #20H	
4103		75 8D A0	MOV TH1, #0A0H	
4106		75 8B 00	MOV TL1, #00H	
4109		75 88 40	MOV TCON, #40H	
410C		75 98 58	MOV SCON, #58H	
410F		90 45 00	MOV DPTR, #4500H	
4112	RELOAD	7D 05	MOV R5, #05H	
4114	CHECK	30 98 FD	JNB SCON.0, CHECK	
4117		C2 98	CLR SCON.0	
4119		E5 99	MOV A, SBUF	
411B		F0	MOVX @DPTR, A	
411C		A3	INC DPTR	
411D		B4 3F F2	CJNE A, #3FH, RELOAD	
4120		DD F2	DJNZ R5, CHECK	
4122		E4	CLAR A	
4123		12 00 20	LCALL 0020H	

Algorithm for Transmitter:

1. Initialize TMOD with 20H
2. Set the values for TCON and SCON
3. Set the input address to DPTR
4. Based on the bit value on SCON store the data in SBUF and move the data to register 'A'.
5. Increment DPTR and check for the loop end value

PROGRAM FOR TRANSMITTER.

Memory Location	Label	Opcode	Mnemonics
4100		75 89 20	MOV TMOD, #20H
4103		75 8D A0	MOV TH1, #0A0H
4106		75 8B 00	MOV TL1, #00H
4109		75 88 40	MOV TCON, #40H
410C		75 98 58	MOV SCON, #58H
410F		90 45 00	MOV DPTR, #4500H
4112	RELOAD	7D 05	MOV R5, #05H
4114	REPEAT	E0	MOVX A, @DPTR
4115		F5 99	MOV SBUF, A
4117	CHECK	30 99 FD	JNB SCON.1, CHECK
411A		C2 99	CLR SCON.1
411C		A3	INC DPTR
411D		B4 3F F2	CJNE A, #3FH, RELOAD
4120		DD F2	DJNZ R5, REPEAT
4122		E4	CLAR A
4123		12 00 20	LCALL 0020H

SAMPLE INPUT AND OUTPUT:

Sl.No	Transmitter Input (Hex Values)	Receiver Output (Hex Values)
1	**00**	**00**
2	**11**	**11**
3	**22**	**22**
4	**33**	**33**

Result:

Thus an assembly language program displaying characters on seven segment display has been executed.

Date of Experiment___

Experiment 6

Aim: To write an assembly language program to display characters on a seven display interface.

Apparatus required:

8051 microcontroller kit

(0-5V) DC battery

Algorithm:

1. Enter a program.
2. Initialize number of digits to Scan
3. Select the digit position through the port address C0
4. Display the characters through the output at address C8.
5. Check whether all the digits are display.
6. Repeat the Process.

PROGRAM:

Memory Location	Label	Opcode	Mnemonics	Comments
4100	START	90 41 2B	DPTR, #TABLE	Display message
4103		AA 82	MOV R2, DPL	
4105		AB 83	MOV R3, DPH	
4107		78 07	MOV R0, #07H	
4109		7F 08	MOV R7, #08H	Initialize no.of digits to scan
410B	L1	E8	MOV A, R0	Select digit position
410C		90 FF C0	MOV DPTR, #0FFC0H	
410F		F0	MOVX @DPTR, A	
4110		8A 82	MOV DPL, R2	
4112		8B 83	MOV DPH, R3	
4114		E0	MOVX A, @DPTR	
4115		90 FF C8	MOV DPTR, #0FFC8H	
4118		F0	MOVX @DPTR, A	
4119		12 41 22	LCALL DELAY	
411C		0A	INC R2	
411D		18	DEC R0	Check if 8 digits are displayed
411E		DF EB	DJNZ R7, L1	If not repeat
4120		21 00	AJMP START	Repeat from the 1st digit
4122	DELAY	7C 02	MOV R4, #02H	
4124	L3	7D FF	MOV R5, #0FFH	
4126	L2	DD FE	DJNZ R5, R2	
4128		DC FA	DJNZ R4, L3	
412A		22	RET	
412B	TABLE	3E 06 00 55	DB 3EH, 06H, 00H, 55H	
412F		06 39 50 3F	DB 06H, 39H, 50H, 3FH	
4133			END	

SAMPLE INPUT AND OUTPUT:

Sl.No	Input (hex Values)	Output (Characters)

Result:

Thus an assembly language program displaying characters on seven segment display has been executed.

Date of Experiment:

Experiment 7

Aim: To write an assembly language program to display Characters on a seven display interface.

Apparatus required:

8051 microcontroller

kit(0-5V) DC battery

Algorithm:

1. Move the Port Address of DAC 2 FFC8 to the DPTR.
2. Move the Value of Register A to DPTR and then Call the delay.
3. Move the Value of Register A (FFh) to DPTR and the call the dalay.
4. Repeat the steps 2 and 3.

PROGRAM TO GENERATE SQUARE WAVEFORM

Memory Location	Label	Opcode	Mnemonics
4100		90 FF C8	MOV DPTR, #0FFC8H
4103	START:	74 00	MOV A, #00H
4105		F0	MOVX @DPTR, A
4106		12 41 12	LCALL DELAY
4109		74 FF	MOV A, #0FFH
410B		F0	MOVX @DPTR, A
410C		12 41 12	LCALL DELAY
410F		02 41 03	LJMP STTART
4112		79 05	MOV R1, #05H
4114		7A FF	MOV R2, #0FFH
4116		DA FE	DJNZ R2, HERE
4118		D9 FA	DJNZ R1, LOOP
411A		22	RET
411B		80 E6	SJMP START

PROGRAM TO GENERATE SAW-TOOTH WAVEFORM

Memory Location	Label	Opcode	Mnemonics
4100		90 FF C0	MOV DPTR, #0FFC0H
4103		74 00	MOV A, #00H
4105		F0	MOVX @DPTR, A
4106		04	INC A
4107		80 FC	SJMP LOOP

PROGRAM TO GENERATE TRIANGULAR WAVEFORM

Memory Location	Label	Opcode	Mnemonics
4100		90 FF C8	MOV DPTR, #0FFC8H
4103		74 00	MOV A, #00H
4105		F0	MOVX @DPTR, A
4106		04	INC A
4107		70 FC	JNZ LOOP1
4109		74 FF	MOV A, #0FFH
411B		F0	MOVX @DPTR, A
410C		14	DEC A
410D		70 FC	JNZ LOOP2
410F		02 41 03	LJMP START

Result:

Thus an assembly language program for Digital to Analog has been executed.

Date of Experiment:

Experiment -8

Aim:To write an assembly language program to display Characters on a seven display interface.

Apparatus required:

8051 microcontroller kit

(0-5V) DC battery

Algorithm:

1. Make ALE low/high by moving the respective data from A register to DPTR.
2. Move the SOC(Start Of Conversion) data to DPTR from FFD0
3. Check for the End Of Conversion and read data from Buffer at address FFC0
4. End the Program.

PROGRAM:

Port Address for
74LS174 Latch: FFC8
Port Address for
SOC:FFD0
Port Address for EOC 1: FFD8
Port Address for 74LS 244 Buffer: FFC0

Memory Location	Label	Opcode	Mnemonics	Comments
4100		90 FF C8	MOV DPTR, #FFC8	
4103		74 10	MOV A, #10	Select Channel 0
4105		F0	MOVX @DPTR, A	Make ALE Low
4106		74 18	MOV A, #18	Make ALE High
4108		F0	MOVX @DPTR, A	
4109		90 FF D0	MOV DPTR, #FFD0	
410C		74 01	MOV A, #01	SOC Signal High
410E		F0	MOVX @DPTR, A	
410F		74 00	MOV A, #00	SOC Signal Low
4111		F0	MOVX @DPTR, A	
4112		90 FF D8	MOV DPTR, #FFD8	
4115		E0	MOVX A, @DPTR	
4116		30 E0 FC	JNB E0, WAIT	Check For EOC
4119		90 FF C0	MOV DPTR, #FFC0	Read ADC Data
411C		E0	MOVX A, @DPTR	
4110		90 41 50	MOV DPTR, #4150	Store the Data
4120		F0	MOVX @DPTR, A	
4121		90 FE	SJMP HERE	

Result:

Thus an assembly language program is executed for analog to digital conversion.

Date of Experiment:

Experiment-9

Aim: To write an assembly program to make the stepper motor run in forward and reverse direction.

Apparatus required:

Stepper motor
8051 microprocessor kit
(0-5V) power supply

Algorithm:

1. Fix the DPTR with the Latch Chip address FFC0
2. Move the values of register A one by one with some delay based on the 2-Phase switching Scheme and repeat the loop.
3. For Anti Clockwise direction repeat the step 3 by reversing the value sequence.
4. End the Program

Memory Location	Label	Opcode	Mnemonics
4100		90 FF C0	MOV DPTR, #FFC0
4103		74 09	MOV A, #09
4105		E0	MOVX @DPTR, A
4106		12 41 3B	LCALL DELAY
4109		74 05	MOV A, #05
410B		E0	MOVX @DPTR, A
410C		12 41 3B	LCALL DELAY
410F		74 06	MOV A, #06
411B		E0	MOVX @DPTR, A
411C		12 41 3B	LCALL DELAY
411F		74 0A	MOV A, #0A
412B		E0	MOVX @DPTR, A
412C		12 41 3B	LCALL DELAY

412F			SJMP 412F
413B	DELAY		
413B	L2		MOV R0, #55
413D	L1		MOV R1, #FF
413F			DJNZ R1, L1
413B			DJNZ R0, L2
413D			RET

Result:

Thus an assembly language program to control of stepper motor was executed successfully using 8051 Microcontroller kit.

Date of Experiment:_____

Experiment-10

Aim: To write an assembly language program to display Characters on a seven display interface.

Apparatus required:

8051 microcontroller kit

(0-5V) DC battery

Algorithm:

1. Fix the control the control and move the control word to control register.
2. Move the Traffic Light LED Position values to Port A, Port B and Port C respectively based on the logic.
3. Fix the delay based on the requirement.
3. Execute the program.

PROGRAM:

4100		ORG	4100
	CONTRL	EQU	0FF0FH
	PORT A	EQU	0FF0CH
	PORT B	EQU	0FF0DH
	PORT C	EQU	0FF0EH

Memory Location	Label	Opcode	Mnemonics
4100		74 80	MOV A, #80H
4102		90 FF 0F	MOV DPTR, #CONTRL
4105		F0	MOVX @DPTR, A
4106	START	7C 04	MOV R4, #04H
4108		90 41 9B	MOV DPTR, #LOOK1
410B		AA 83	MOV R2, DPH
410D		AB 82	MOV R3, DPL
410F		90 41 8F	MOV DPTR, #LOOK
4112		A8 83	MOV R0, DPH
4114		A9 82	MOV R1, DPL
4116	GO	E0	MOVX A, @DPTR
4117		A8 83	MOV R0, DPH
4119		A9 82	MOV R1, DPL
411B		90 FF 0C	MOV DPTR, #PORT A
411E		F0	MOVX @DPTR, A
411F		09	INC R1
4120		88 83	MOV DPH, R0
4122		89 82	MOV DPL, R1
4124		E0	MOVX A, @DPTR
4125		A8 83	MOV R0, DPH
4127		A9 82	MOV R1, DPL
4129		90 FF 0D	MOV DPTR, #PORT B
412C		F0	MOVX @DPTR, A
412D		09	INC R1
412E		88 83	MOV DPH, R0
4130		89 82	MOV DPL, R1
4132		E0	MOVX A, @DPTR
4133		A8 83	MOV R0, DPH
4135		A9 82	MOV R1, DPL
4137		90 FF 0E	MOV DPTR, #PORT C
413A		F0	MOVX @DPTR, A
413B		09	INC R1
413C		12 41 75	LCALL DELAY
413F		8A 83	MOV DPH, R2
4141		8B 82	MOV DPL, R3
4143		E0	MOVX A, @DPTR
4144		AA 83	MOV R2, DPH
4146		AB 82	MOV R3, DPL
4148		90 FF 0C	MOV DPTR, #PORT A
414B		F0	MOVX @DPTR, A
414C		0B	INC R3
414D		8A 83	MOV DPH, R2
414F		8B 82	MOV DPL, R3
4151		E0	MOVX A, @DPTR
4152		AA 83	MOV R2, DPH
4154		AB 82	MOV R3, DPL
4156		90 FF 0D	MOV DPTR, #PORT B
4159		F0	MOVX @DPTR, A

415A		0B	INC R3
415B		8A 83	MOV DPH, R2
415D		8B 82	MOV DPL, R3
415F		E0	MOVX A, @DPTR
4160		AA 83	MOV R2, DPH
4162		AB 82	MOV R3, DPL
4164		90 FF 0E	MOV DPTR, #PORT C
4167		F0	MOVX @DPTR, A
4168		0B	INC R3
4169		12 41 82	LCALL DELAY1
416C		88 83	MOV DPH, R0
416E		89 82	MOV DPL, R1
4170		DC A4	DJNZ R4, GO
4172		12 41 06	LCALL START
4175	DELAY	7D 12	MOV R5, #12H
4177	L3	7E FF	MOV R6, #0FFH
4179	L2	7F FF	MOV R7, #0FFH
417B	L1	DF FE	DJNZ R7, L1
417D		DE FA	DJNZ R6, L2
417F		DD F6	DJNZ R5, L3
4181		22	RET
4182	DELAY1	7D 12	MOV R5, #12H
4184	L6	7E FF	MOV R6, #0FFH
4186	L5	7F FF	MOV R7, #0FFH
4188	L4	DF FE	DJNZ R7, L4
418A		DE FA	DJNZ R6, L5
418C		DD F6	DJNZ R5, L6
418E		22	RET
418F	LOOK	44 27 12	DB 44H, 27H, 12H
4192		92 2B 10	DB 92H, 2BH, 10H
4195		84 9D 10	DB 84H, 9DH, 10H
4198		84 2E 48	DB 84H, 2EH, 48H
419B	LOOK1	48 27 12	DB 48H, 27H, 12H
419E		92 4B 10	DB 92H, 4BH, 10H
41A1		84 9D 20	DB 84H, 9DH, 20H
41A4		04 2E 49	DB 04H, 2EH, 49H

Result:

Thus an assembly language program for the Traffic Light Control has been executed.