ESTD:2008

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Approved by AICTE, New Delhi & Permanently Affiliated to JNTUGV, Vizianagaram)

Recognized under Section 2(f) & 12(B) of the UGC Act, 1956| Accredited by NAAC with 'A' Grade

SONTYAM, ANANDAPURAM, VISAKHAPATNAM - 531173

Department of Electronics and Communication Engineering

MICROPROCESSORS AND MICROCONTROLLERS
STUDENT LABORATORY RECORD

For

III B.Tech, II Semester

(AR20 Regulations)

Prepared by

Mrs. M.V.S. Roja Ramani., M.Tech, (Ph.D)

Assistant Professor, Dept. of ECE

NAME OF THE STUDENT_____

ROLL.NO_____

BRANCH_____

YEAR_____SEMESTER_____

ESTD:2008

NSRIT

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Approved by AICTE, New Delhi & Permanently Affiliated to JNTUGV, Vizianagaram)

Recognized under Section 2(f) & 12(B) of the UGC Act, 1956| Accredited by NAAC with 'A' Grade

SONTYAM, ANANDAPURAM, VISAKHAPATNAM - 531173

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

This is to certify that **Mrs. M.V.S.ROJARAMANI, Assistant Professor, Dept. of ECE** has prepared the **Microprocessors and Microcontrollers Lab** manual as per Nadimpalli Satyanarayana Raju Institute of Technology (Autonomous) AR20 Regulation with utmost care, and I thoroughly verified the manual and approved it.

Prepared by

Mrs. M.V.S. ROJARAMANI

Assistant Professor

Approved by

HOD

ESTD

NSRIT

NS RAJU INSTITUTE OF TECHNOLOGY

AUTONOMOUS

SONTYAM, PENDURTI - ANANDAPURAM HIGHWAY, VISAKHAPATNAM - 531 173

(Apporved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada)

(AN ISO 9001 : 2008 Certified Institution)

CERTIFICATE

This is certify that is bonafied record of Practical work done in the Department of .....B.Tech..... Year.....Semester during the academic year -20 by

Name :.....

Roll No.....Branch.....Subject.....

Staff Member ***Head of the Department***

Record Submitted for the Practical Examination

Held on :.....Batch No.....

INTERNAL EXAMINER ***EXTERNAL EXAMINER***

INSTITUTE

VISION

To promote societal empowerment and become an institution of excellence in the field of engineering educationand research.

MISSION

- To develop the students into outstanding professionals through innovative Teaching-Learning process.
- To uphold research through long term Academia-Industry interaction.
- To inculcate ethical standards and moral values.

ECE DEPARTMENT

VISION

To become recognized forerunner in Electronics and Communication Engineering by producing competent and responsible graduates.

MISSION

- To prepare technically competent graduates by establishing a conducive learner centric academic environment that uses innovative teaching learning processes
- To create research interests in the graduates by bringing in real time engineering challenges through industry collaborations
- To make the graduates socially responsible citizens who provide sustainable solutions maintaining ethical and professional-standard.

CODE OF CONDUCT FOR THE LABORATORIES

A. All students must observe the Dress Code while in the laboratory.

B. Sandals or open-toed shoes are NOT allowed.

C. All bags must be left at the indicated place.

D. The lab timetable must be strictly followed.

E. Be PUNCTUAL for your laboratory session.

F. Program must be executed within the given time.

G. Noise must be kept to a minimum.

H. Workspace must be kept clean and tidy at all time.

I. Handle the systems and interfacing kits with care.

J. All students are liable for any damage to the accessories due to their own negligence.

K. All interfacing kits connecting cables must be RETURNED if you taken from the lab supervisor.

L. Students are strictly PROHIBITED from taking out any items from the laboratory.

M. Students are NOT allowed to work alone in the laboratory without the Lab Supervisor

N. USB Ports have been disabled if you want to use USB drive consult lab supervisor.

O. Report immediately to the Lab Supervisor if any malfunction of the accessories, is there.

Before leaving the lab

1. Place the chairs properly.
2. Turn off the system properly
3. Turn off the monitor.

Please check the laboratory notice board regularly for updates

PC 20EC606 Microprocessors and Microcontrollers Lab 0 0 3 1.5

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

Code	Course Outcomes	Mapping with POs	
		PO4	PO9
20EC606.1	Develop assembly language programs to perform arithmetic, logical operations, string operations using TASM and 8086 Microprocessor boards	3	2
20EC606.2	Design interfacing circuits using 8086 Microprocessor	2	3
20EC606.3	Construct different waveforms using 8086 Microprocessor and 8051 Microcontroller	3	3
20EC606.4	Develop and implement assembly language programs to perform real time interfacing using 8051Microcontrollers	3	3
20EC606.5	Implement assembly language programs to perform arithmetic operations using ARM Processors	3	3

1. Weakly Contributing | 2. Moderately Contributing | 3. Strongly Contributing, for the attainment of respective Pos

List of Experiments

1. Programs for 16 -bit arithmetic operations using 8086 programs
2. Perform BCD Addition
3. Write an assembly program for finding factorial of a given number
4. Program for sorting an array
5. Interfacing ADC to 8086
6. Interfacing DAC to 8086
7. Interfacing stepper motor to8086
8. Finding number of 1's and number of 0's in a given 8-bit number
9. Program to find Average of n-numbers
10. Interfacing Traffic Light Controller to8051
11. Timer Mode Programming
12. Write an assembly program to multiply of 2 16-bit binary numbers
13. Write an assembly program to find the sum of first 10 integers numbers
14. Write a program to toggle LED every second using timer interrupt

References

1. Lab Manual for Microprocessors and Microcontrollers of Electronics and Communication Engineering, NSRIT

CONTROL COPY ATTESTED

Chairman
Board of Studies (ECE)

Head of the Department
Dept. of Electronics & Communication Engg.,
N.S.Raju Institute of Technology
Sontyam, Visa[illegible] 531 173

Program Educational Objectives (PEOs)

PEO 1: Demonstrate the application of domain knowledge in solving real time problems and provide research based sustainable solutions in different specializations of Electronics and Communication Engineering or allied branch of engineering and technology and lead a satisfactory job employment with 21stcentury skills.

PEO 2: Involve themselves in life-long learning by enriching his/her competency in the chosen field of interest through professional experience, advanced studies leading to research, learning new age skills that demands dynamism for a continued better prospect to accomplish their professional and career goals.

PEO 3: Demonstrate the skill sets that are very much essential to work successfully for a rewarding career in an interdisciplinary environment.

PROGRAMME OUTCOMES (PO) :

PROGRAM OUTCOMES (POs):

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

PO 2: **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.

PO 3: **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.

PO 4: **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.

PO 5: **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO 6: **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.

PO 7: **Environment and sustainability:** Understand the impact of the professional engineering solutionsin societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

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**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 10: **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 reportsand design documentation, make effective presentations, and give and receive clear instructions.

PO 11: **Project management and finance:** Demonstrate knowledge and understanding of the Engineering and management principles and apply these to one's own work, as a memberand leader in a team, to manage projects and in multidisciplinary environments.

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 (PSO's)

PSO 1: To demonstrate the ability to design and develop complex systems in the areas of next generation Communication Systems, IoT based Embedded Systems, Advanced Signal and Image Processing, latest Semiconductor technologies, RF and Power Systems

PSO 2: To demonstrate the ability to solve complex Electronics and Communication Engineering problems usinglatest hardware and software tools along with analytical skills to contribute to useful, frugaland eco-friendly solutions.

CO-PO Mapping														
	Programme Outcomes													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1				**3**					**2**					
CO2				**2**					**3**					
CO3				**3**					**3**					
CO4				**3**					**3**					
CO5				**3**					**3**					

Strength of Correlation:
Weak-1
Medium-2
Strong-3

INDEX

Sl.No	Date	Name of the Experiment	Pg.No	Marks	Remarks

Sl.No	Date	Name of the Experiment	Pg.No	Marks	Remarks

1.1 INTRODUCTION TO MASM/TASM

ASSEMBLY LANGUAGE PROGRAMMING USING MASM/TASM SOFTWARE:

This software used to write a program (8086, Pentium processors etc.)

The programs are written using assembly language in editor then compile it. The complier converts assembly language statements into machine language statements/checks for errors. Then execute the compiled program. Programs for different processor instructions (Pentium, 8086) programming manner differ for each model.

There are different software's developed by different companies for assembly language programming are:

- □ **MASM** - Microsoft Company.
- □ **TASM** - Bore Land Company.

MERIT OF MASM/TASM:

1. produces binary code
2. Referring data items by their names rather than by their address.

HOW TO ENTER INTO MASM EDITOR:

□ Click "**Start"** on the desktop.

□ Then select **Run**

□ Then it Shows inbox

□ Then type Command (CMD) which enters You into **DOS prompt**

□ Path setting

Suppose it display path as **C:\ DOCUME-\ADMIN>**
Then type **CD**
i.e.; **C:\DOCUME\\ADMIN>CD**
Then the path is **C :\>**

Then type **CD TASM**

Then the path is **C: TASM>**

Then type edit i.e.; **C: TASM>edit**

Then you enter into **TASM** text editor.

Then enter to file and select New.
And name it and then write the **ALP** (Assembly Language Program) in this editor.
After that save it as **filename's**
Then exit from the editor and go to prompt.
Then type TASM filename.ASM
I.e. C: MASM>MASM filename.ASM **or C: TASM filename.ASM, , ;**
Then link this file using **C: TASM>TINK filename.OBJ**
or **C: TASM>TLINK filename.OBJ , , ;**
i.e., link the program in assembly with **DOS**
then to debug to create exe file
C:TASM>debug filename. EXE
Then it display **"--"** on the screen
After that type **„R"** displays the registers contents steps and starting step of the program.
"**T"** Tracing at contents of program step by step
Suppose you need to go for break point debugging. Then type that instruction no where you need to check your register. For example T_{10} it will display the contents of register after executing 10 instructions.

DEBUG:

This command utility enables to write and modify simple assembly language programs in an easy fashion. It provides away to run and test any program in a controlled environment.

We can change any part of the program and immediately execute the program with an having to resemble it. We can also run machine language (Object files) directly by using DEBUG

DEBUG COMMANDS:

ASSEMBLE A [address] ; Assembly the instructions at a particular address

COMPARE C range address ; Compare two memory ranges

DUMP D [range] ; Display contents of memory

ENTER E address [list] ; Enter new or modifies memory contents beginning at Specific Location

FILL F range list ; Fill in a range of memory

GO G [=address] [addresses] ; Execute a program in memory
HEX H value1 value2 ; Add and subtract two Hex values

INPUT I port

LOAD L [address] [drive] [first sector] [number]

MOVE M range address

NAME N [pathname] [arg list]

OUTPUT O port byte

PROCEED P [=address] [number]

QUIT Q

REGISTER R [register]

SEARCH S range list

TRACE T [=address] [value]

UNASSEMBLE U [range]

WRITE W [address] [drive] [first sector] [number]

ALLOCATE expanded memory XA [#pages]

DEALLOCATE expanded memory XD [handle]

MAP expanded memory pages XM [Lpage] [Ppage] [handle]

EXPERIMENT-1

ARITHMETIC OPERATIONS ON 16BIT DATA

AIM: Assembly language program to perform all Arithmetic operations on 16bit data

SOFTWARE USED: TASM SOFTWARE,PC

ALGORITHM:

Step1: Start

Step2: Initialize data segment

Step3: Load SI with memory location 5000

Step4: Move the contents from memory location [0000] to AX

Step5: Move the contents from memory location [0001] to BX

Step6: Perform addition

Step7: Move the result to the memory location specified

Step8: Copy the contents from the memory location [0000] to AX

Step9: Perform subtraction

Step10: Move the result to the memory location [SI+02]

Step11: Copy the contents from the memory location [0000] to AX

Step12: Perform multiplication

Step13: Move the result to the memory location [SI+04] &[SI+06]

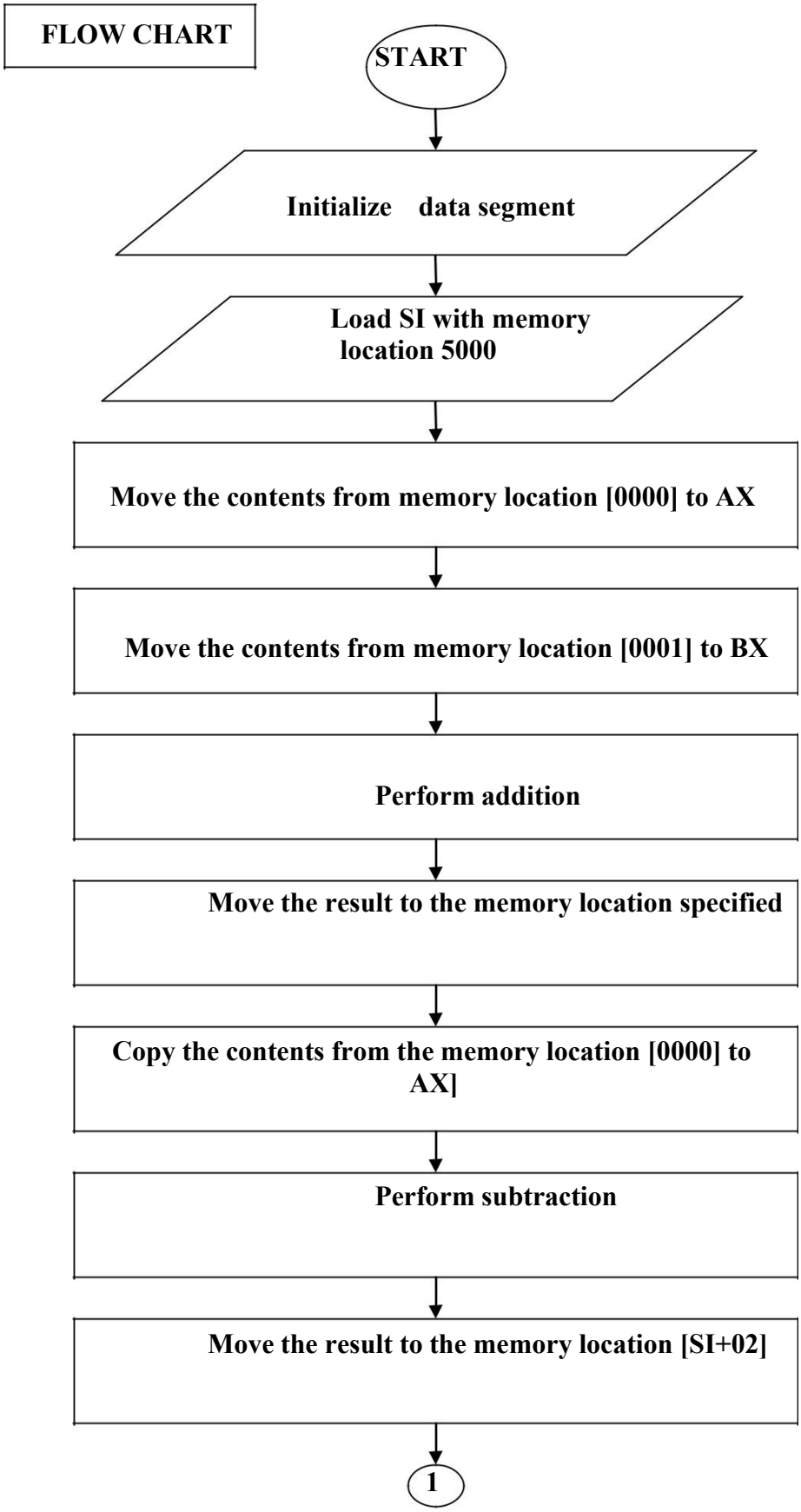
Step14: Copy the contents from the memory location [0000] to AX

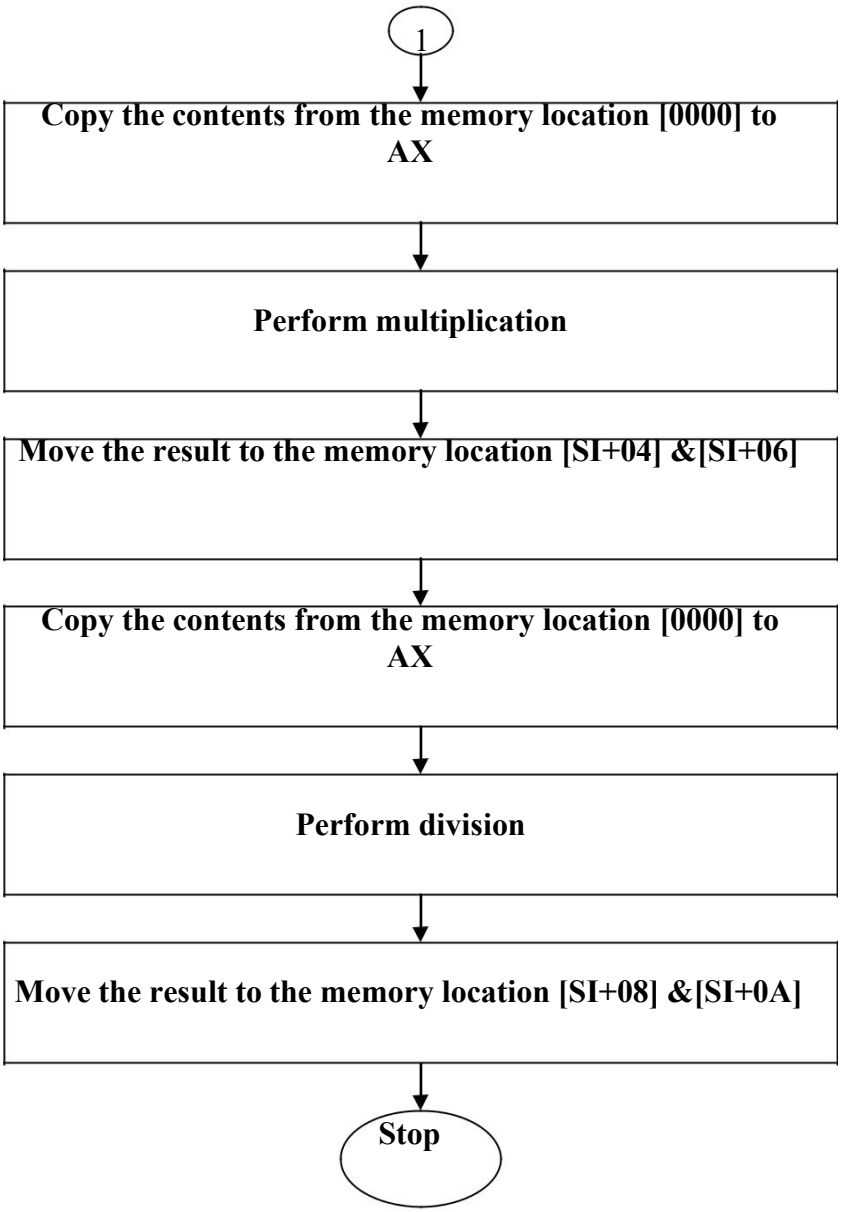
Step15: Perform division

Step16: Move the result to the memory location [SI+08]&[SI+0A]

Step17: Stop.

FLOW CHART


START
Initialize data segment
Load SI with memory location 5000
Move the contents from memory location [0000] to AX
Move the contents from memory location [0001] to BX
Perform addition
Move the result to the memory location specified
Copy the contents from the memory location [0000] to AX]
Perform subtraction
Move the result to the memory location [SI+02]
1


1
Copy the contents from the memory location [0000] to AX
Perform multiplication
Move the result to the memory location [SI+04] &[SI+06]
Copy the contents from the memory location [0000] to AX
Perform division
Move the result to the memory location [SI+08] &[SI+0A]
Stop

PROGRAM:

```
ASSUME      CS: CODE,    DS: DATA

DATA    SEGMENT
N1      EQU         8888H
N2      EQU         4444H
DATA ENDS

CODE    SEGMENT

START: MOV          AX, DATA
           MOV          DS, AX
           MOV          SI, 5000H
           MOV          AX, N1
           MOV          BX, N2
           ADD          AX, BX
           MOV          [SI], AX
           MOV          AX, N1
           SUB          AX, BX
           MOV          [SI+2], AX
           MOV          AX, N1
           MUL          BX
           MOV          [SI+4], AX
           MOV          [SI+6], DX
           MOV          AX, N1
           MOV          DX, 0000
           DIV          BX
           MOV          [SI+8], AX
           MOV          [SI+0AH], DX
           MOV          AH, 4CH
           INT          21H

CODE ENDS

END START
```

CODE TABLE:

Physical Address		Label	Hex code	Mnemonic Operands	Comments
Segment address	Effective address				

Observations:

VIVA QUESTIONS:

1. What are the instructions used to perform signed multiplication and division.

2. Distinguish Macros and Procedures

RESULT:

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

Department Electronics and Communication Engineering
Rubrics for evaluation of Laboratory

Name of student:
Roll No:
Experiment No:

	Not Satisfactory	Satisfactory	Good	Excellent	Score
Problem Underas -tanidng (2)	Unable to explain the problem statement and its requirements	Able to explain the problem statement	Able to explain the problem statement, fairlyon other requirements	Able to give full explanation on problem statement and on input and output	
Ability to conduct experim-ent (4)	Has poor understanding of software tools including its purpose and unable to conduct experiment on his own , lab instructor provides help in every step of the experiment	Has good understanding of software tools including its purpose and able to conduct experiment with some help of , lab instructor provides help in every step of the experiment	Has good understandin g of software tools including its purpose and able to conduct experiment with negligible help from the lab instructor	Fully understanding th e software tools including its purpose and able to conduct the experiment by own	
Program ,Simulati -ons and Results (4)	Has poor simulation skills unable to write code and has no ability to identify and remove the coding errors even with instructor's help. Unable to produce output results	Has good Simulations Skills and be able to write the code with less errors and Unable to Explain the output	Has good simulation skills and able to make the acceptable coding logic with less errors.	Has good simulation skills and able to make the acceptable coding logic with desired outputs for a given problem.	
				Total	
				Signature	

EXPERIMENT-2

ADDITION OF BCD NUMBERS

AIM: Assembly language program to perform addition of two 8 bit BCD numbers

SOFTWARE: TASM SOFTWARE

ALGORITHM:

Step 1: Start

Step 2: Initialize Data Segment

Step 3: Move the N1value into AL register

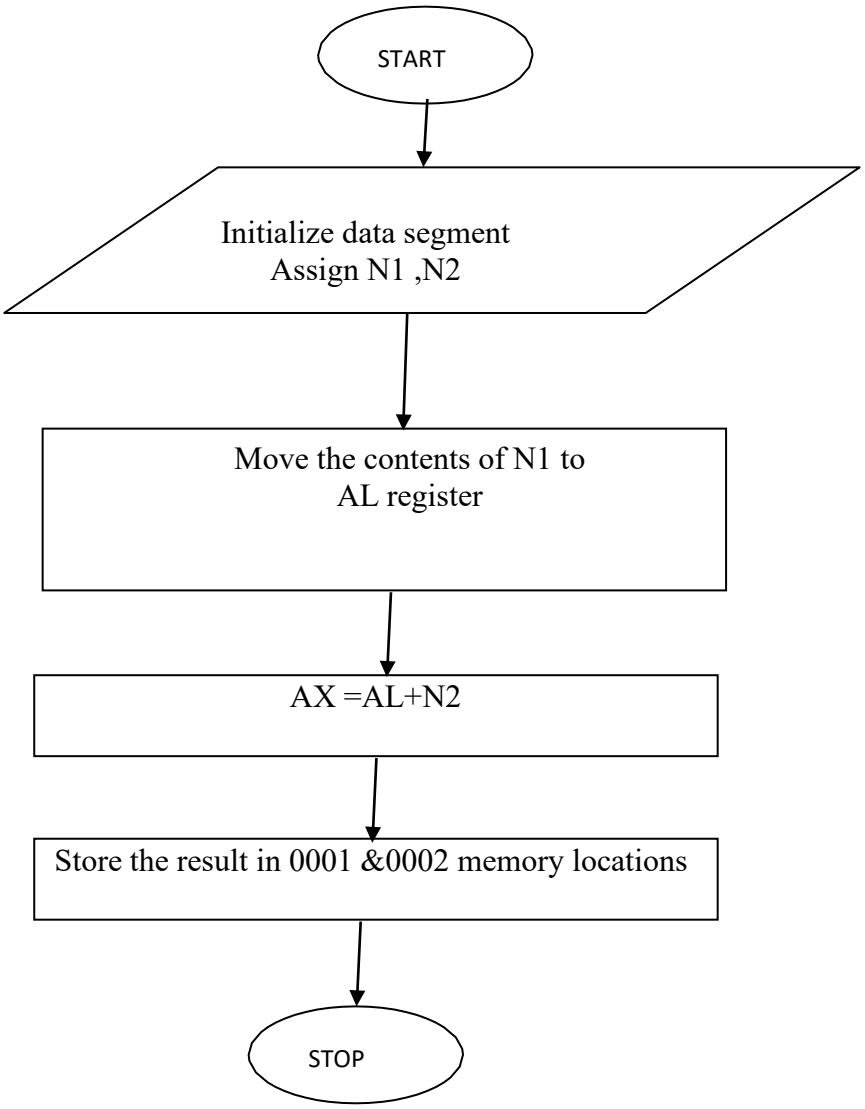
Step 3: Add the N2 Value to AL register

Step 4: Perform ASCII Adjust after Addition on the result

Step 5: Store the result in the consecutive memory locations

Step 6: Stop

FLOW CHART



PROGRAM:

```
ASSUME CS:CODE,DS:DATA
DATA SEGMENT
N1 DB 05
N2 DB 06
RESULT DB 04H DUP(00)
DATA ENDS
CODE SEGMENT
START:
MOV AX,DATA
MOV DS,AX
MOV AL,N1
ADD AL,N2
MOV [RESULT],AL
MOV AH,0
AAA
MOV [RESULT+1], AL
MOV [RESULT+2], AH
MOV AH,4CH
INT 21H
CODE ENDS
END START
```

CODE TABLE:

Physical Address		Label	Hex code	Mnemonic Operands	Comments
Segment address	Effective address				

Observations:

VIVA QUESTIONS:

1. Compare EQU and DB directives.
2. Explain AAA, AAS, AAM & AAD instructions.
3. What is the instruction used for finding 2'scomplement of a number.

RESULT:

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

Department Electronics and Communication Engineering
Rubrics for evaluation of Laboratory

Name of student:
Roll No:
Experiment No:

	Not Satisfactory	Satisfactory	Good	Excellent	Score
Problem Underas -tanidng (2)	Unable to explain the problem statement and its requirements	Able to explain the problem statement	Able to explain the problem statement, fairlyon other requirements	Able to give full explanation on problem statement and on input and output	
Ability to conduct experim-ent (4)	Has poor understanding of software tools including its purpose and unable to conduct experiment on his own , lab instructor provides help in every step of the experiment	Has good understanding of software tools including its purpose and able to conduct experiment with some help of , lab instructor provides help in every step of the experiment	Has good understandin g of software tools including its purpose and able to conduct experiment with negligible help from the lab instructor	Fully understanding th e software tools including its purpose and able to conduct the experiment by own	
Program ,Simulati -ons and Results (4)	Has poor simulation skills unable to write code and has no ability to identify and remove the coding errors even with instructor's help. Unable to produce output results	Has good Simulations Skills and be able to write the code with less errors and Unable to Explain the output	Has good simulation skills and able to make the acceptable coding logic with less errors.	Has good simulation skills and able to make the acceptable coding logic with desired outputs for a given problem.	
				Total	
				Signature	

EXPERIMENT-3

FACTORIAL OF A GIVEN NUMBER

AIM: Assembly language program to perform Factorial of a given number

SOFTWARE: TASM SOFTWARE

ALGORITHM:

Step1: Start

Step2: Initialize data segment

Step3: Move the contents from N1 to CX register

Step4: Move the contents 0001H to AX register

Step 5: Multiply the contents of AX & CX registers

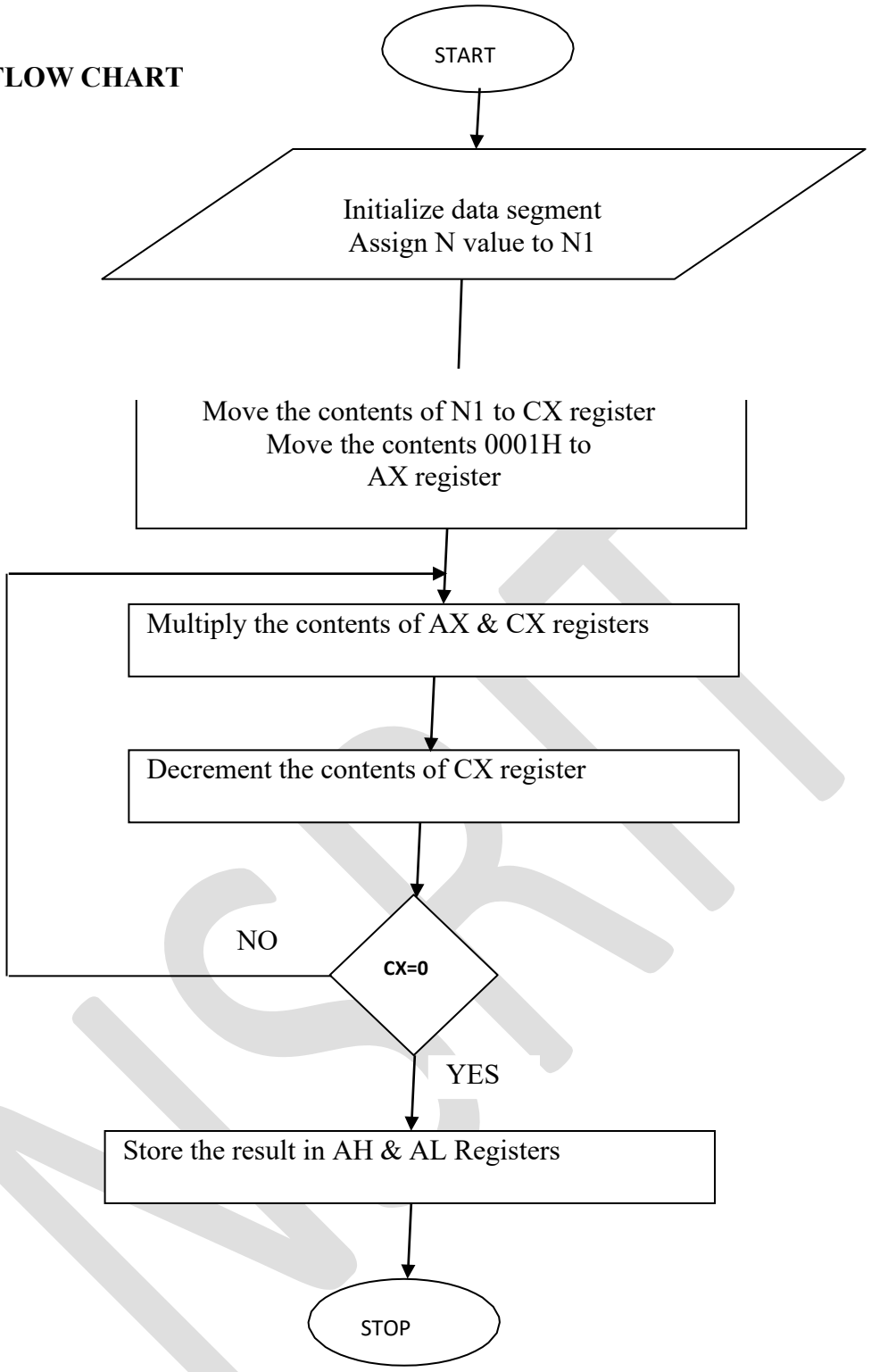
Step6: Decrement CX register

Step7: Repeat step 5 till CX register content become zero

Step8: Move the result to the AX register

Step9: Stop

FLOW CHART


START
Initialize data segment
Assign N value to N1
Move the contents of N1 to CX register
Move the contents 0001H to
AX register
Multiply the contents of AX & CX registers
Decrement the contents of CX register
CX=0
NO
YES
Store the result in AH & AL Registers
STOP

PROGRAM:

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT
N1        EQU         08
RESULT DB  01H    DUP (00)
DATA ENDS
CODE SEGMENT
START:
MOV CX, N1
MOV AX,0001H
LOOP1:MUL CX
DEC CX
JNZ  LOOP1
MOV [RESULT],AL
MOV [RESULT+1],AH
MOV     AH, 4CH
INT       21H
CODE ENDS
END START
```

CODE TABLE:

Physical Address		Label	Hex code	Mnemonic Operands	Comments
Segment address	Effective address				

Observations:

VIVA QUESTIONS:

1. What are branching and loop instructions.
2. Explain XCHG,XLAT instructions

RESULT:

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

Department Electronics and Communication Engineering
Rubrics for evaluation of Laboratory

Name of student:
Roll No:
Experiment No:

	Not Satisfactory	Satisfactory	Good	Excellent	Score
Problem Underas -tanidng (2)	Unable to explain the problem statement and its requirements	Able to explain the problem statement	Able to explain the problem statement, fairlyon other requirements	Able to give full explanation on problem statement and on input and output	
Ability to conduct experim-ent (4)	Has poor understanding of software tools including its purpose and unable to conduct experiment on his own , lab instructor provides help in every step of the experiment	Has good understanding of software tools including its purpose and able to conduct experiment with some help of , lab instructor provides help in every step of the experiment	Has good understandin g of software tools including its purpose and able to conduct experiment with negligible help from the lab instructor	Fully understanding th e software tools including its purpose and able to conduct the experiment by own	
Program ,Simulati -ons and Results (4)	Has poor simulation skills unable to write code and has no ability to identify and remove the coding errors even with instructor's help. Unable to produce output results	Has good Simulations Skills and be able to write the code with less errors and Unable to Explain the output	Has good simulation skills and able to make the acceptable coding logic with less errors.	Has good simulation skills and able to make the acceptable coding logic with desired outputs for a given problem.	
				Total	
				Signature	

EXPERIMENT – 4

SORTING OF "N" NUMBERS

AIM: Assembly language program to do sorting of numbers in a given series

SOFTWARE USED: TASM SOFTWARE, PC

ALGORITHM:

Step1: Start

Step2: Initialize data segment

Step3: Load CX register with count

Step4: Copy the contents from CX to DX

Step5: Load SI with offset list

Step6: Copy the contents from DX to CX

Step7: Move the contents from memory location SI to AL

Step8: Increment SI

Step9: Compare AL contents with [SI]

Step10: Jump to step15 if carry

Step11: Exchange the contents of AL with [SI]

Step12: Decrement SI

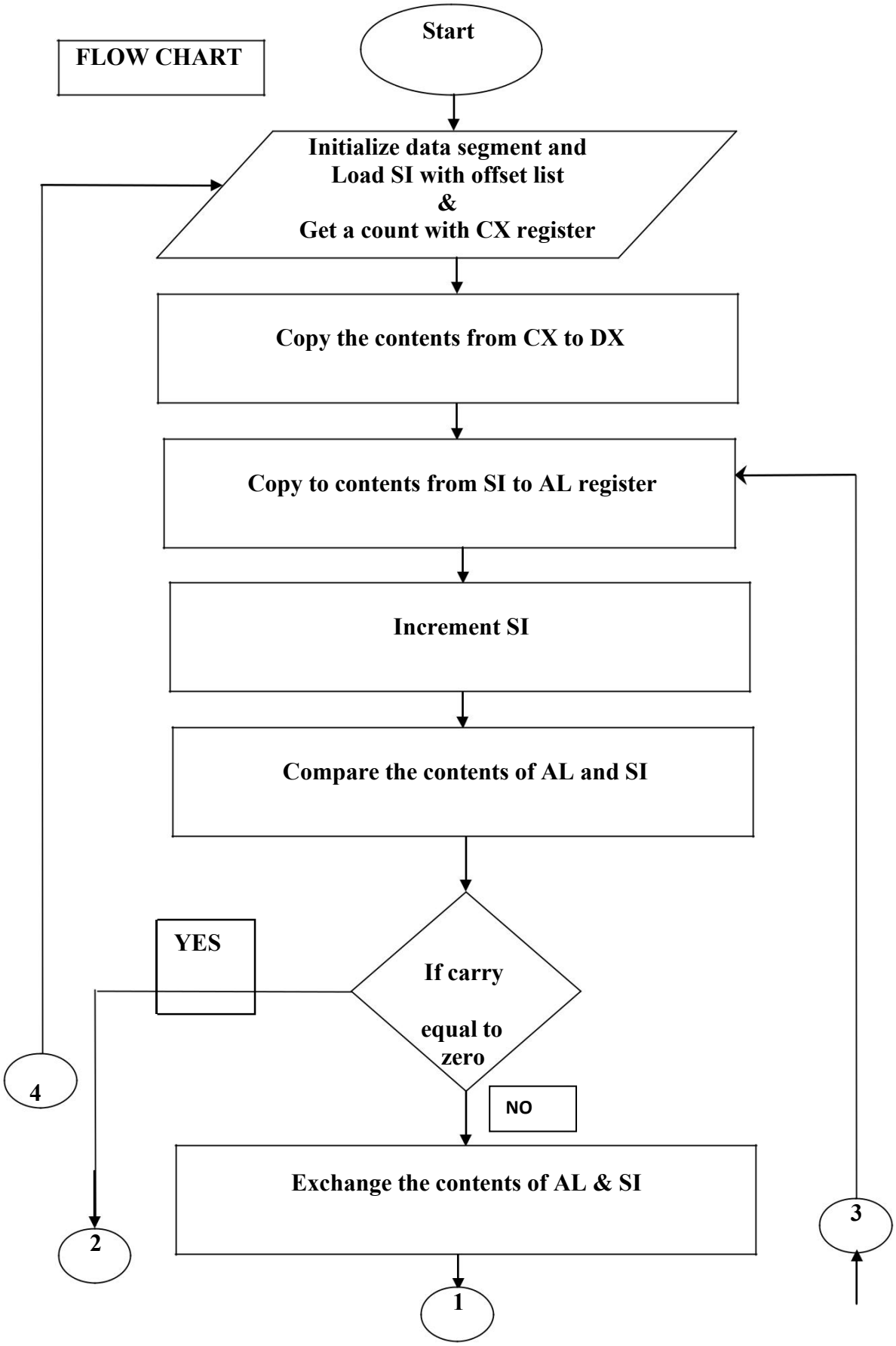
Step13: Move the contents from AL to memory location SI

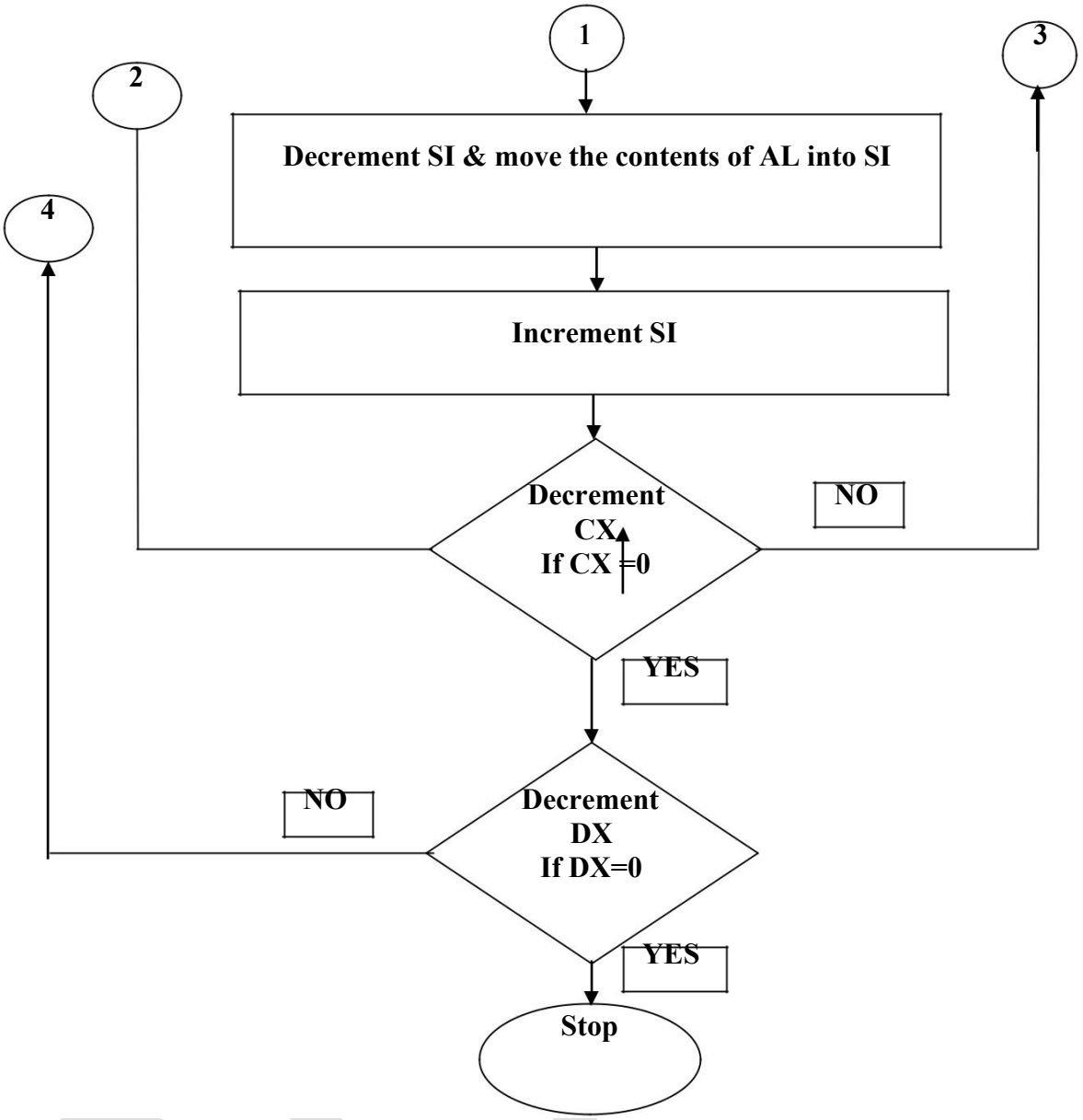
Step14: Increment SI

Step15: Decrement CX and jump to step7 if no zero

Step16: Decrement DX and jump to step5 if no zero

Step17: Stop


FLOW CHART
Start
Initialize data segment and
Load SI with offset list
&
Get a count with CX register
Copy the contents from CX to DX
Copy to contents from SI to AL register
Increment SI
Compare the contents of AL and SI
If carry
equal to
zero
YES
NO
Exchange the contents of AL & SI
1
2
3
4


1
2
3
4
Decrement SI & move the contents of AL into SI
Increment SI
Decrement CX If CX =0
NO
YES
Decrement DX If DX=0
NO
YES
Stop

PROGRAM:

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT
LIST DB 56H, 12H, 72H, 32H
COUNT EQU 0004H
DATA ENDS
CODE SEGMENT
ORG 1000H
START:  MOV AX, DATA
        MOV DS, AX
        MOV CX, COUNT
        MOV DX, CX
AGAIN:  MOV SI, OFFSET LIST
        MOV CX, DX
BACK:   MOV AL, [SI]
        INC SI
        CMP AL, [SI]
        JC NEXT
        XCHG [SI], AL
        DEC SI
        MOV [SI], AL
        INC SI
NEXT:   LOOP BACK
        DEC DX
        JNZ AGAIN
        MOV AH,4CH
        INT 21H
        CODE ENDS
        END START
```

CODE TABLE:

Physical Address		Label	Hex code	Mnemonic Operands	Comments
Segment address	Effective address				

Observations:

VIVA QUESTIONS:

1. What Is A Microprocessor
2. What Are The Functions Of Segment Registers
3. Explain The Assembly Directives Used In The Sorting Program

RESULT:

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

Department Electronics and Communication Engineering
Rubrics for evaluation of Laboratory

Name of student:
Roll No:
Experiment No:

	Not Satisfactory	Satisfactory	Good	Excellent	Score
Problem Underas -tanidng (2)	Unable to explain the problem statement and its requirements	Able to explain the problem statement	Able to explain the problem statement, fairlyon other requirements	Able to give full explanation on problem statement and on input and output	
Ability to conduct experim-ent (4)	Has poor understanding of software tools including its purpose and unable to conduct experiment on his own , lab instructor provides help in every step of the experiment	Has good understanding of software tools including its purpose and able to conduct experiment with some help of , lab instructor provides help in every step of the experiment	Has good understandin g of software tools including its purpose and able to conduct experiment with negligible help from the lab instructor	Fully understanding th e software tools including its purpose and able to conduct the experiment by own	
Program ,Simulati -ons and Results (4)	Has poor simulation skills unable to write code and has no ability to identify and remove the coding errors even with instructor's help. Unable to produce output results	Has good Simulations Skills and be able to write the code with less errors and Unable to Explain the output	Has good simulation skills and able to make the acceptable coding logic with less errors.	Has good simulation skills and able to make the acceptable coding logic with desired outputs for a given problem.	
				Total	
				Signature	

EXPERIMENT-5

Interfacing ADC to 8086

AIM: To Convert the given Dual slope Analog voltage to Digital form using 8255 PPI.

APPARATUS:

1. 8086 microprocessor kit and ADC kit
2. Power supply
3. Key board

THEORY:

Analog to digital conversion can be done in many ways. One of the methods is the Dual slope method used to achieve high noise immunity. The input voltage V_X is integrated for a fixed time Tr. Also a known reference voltage Vr is integrated for a time T_X. Now the input voltage V_X is given by

r X X r V V T T □

But (a constant) rr V K T □

Hence .X X VKT□

i.e the input voltage is proportional to the measured time TX

If the integration time T_r is chosen as 20ms, the mains hum can be suppressed. The microprocessor measures time as the number of counts, namely

$T_X = N_X.T_C$

Where N_X = Number of counts

TC = Counter period.

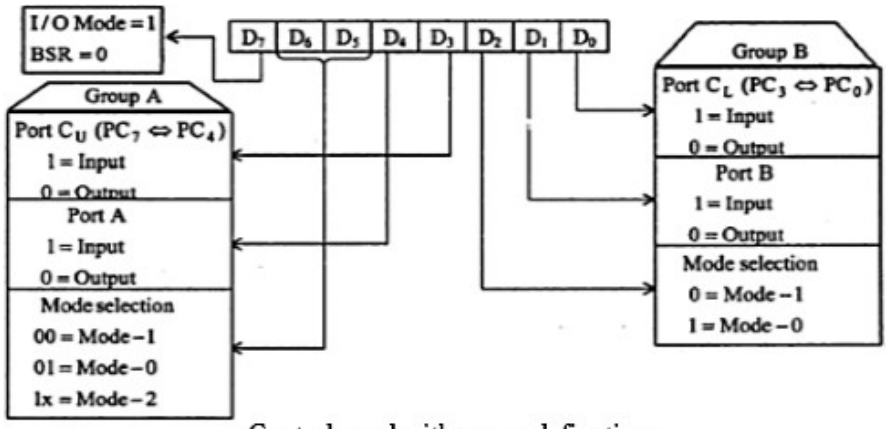
By proper choice of the counter period T_C, scaling can be incorporated into the counter such that a software multiplication can be avoided

DESCRIPTION OF THE CIRCUIT:

This interface has LM339, which is a quad differential comparator of which three comparators are used to trigger the bi – lateral switches present in CD4066. The LM308 OP-AMP is used as an integrator. Finally the LM311 is used as the differential comparator. The steps carried out by the microprocessor for A/D conversion are as follows.

1. Switch SW1 is ON to reset the integrator. This can be accomplished by setting pin7 of the LM339 high (i.e. by sending a high on bit 1 of port A), which in turn triggers the bilateral switch of CD 4066, and the required delay can be introduced in software.
2. Switch SW2 is ON and start the counter at count N. This is done by sending a high on bit 0 of port A and by decrementing the count to zero. It is during this that the analog input is integrated.
3. After the counter counts down to zero, Switch SW3 is ON and starts the counter again. This is done by sending a high on bit 2 of port A. It is during this period that the reference voltage is integrated.
4. Stop the counter when the LM311 comparator output changes its state and read the timer count. This is done by sampling the bit 0 of port B which inputs the comparator output.

CONTROL WORD (8255):



Control word with group defination

	D7	D6	D5	D4	D3	D2	D1	D0
BITS	1	0	0	0	0	0	1	0
HEX CODE	8				2			

PROGRAM:

THE PROGRAM STARTS AT 3000(ORIGIN) MEMORY LOCATION

LABEL	MNEMONICS		COMMENTS
	MOV	DX,0FFE6	; Initialize 8255
	MOV	AL,82	
	OUT	DX,AL	; Port A as output and Port B as input port
START:	MOV	AL,02	
	MOV	DX, 0FFE0	; Reset integrator
	OUT	DX,AL	
	NOP		
	NOP		
	NOP		
	NOP		
	NOP		
	MOV	AL,01	
	MOV	DX, 0FFE0	
	OUT	DX, AL	; Integrate Vin for
	MOV	CX,1000	; Known value of time
$1:	LOOP	$1	
	MOV	AL,04	
	MOV	DX,0FFE0	; Feed Vref to
	OUT	DX,AL	; Comparator
	MOV	BL, 00	; Start counter
TVAR:	MOV	DX,0FFE2	
	IN	AL, DX	; Check for comparator
	AND	AL, 01	; output. If high,
	JE	DISP	; display digital value
	INC	BL	; else increment
	JMP	TVAR	; counter and continue

DISP:	MOV	AL, BL	; Routine to output
	CALLS	0FE00:0052	; digital value
	MOV	AL, 02	; Reset integrator
	MOV	DX, 0FFE0	
	OUT	DX, AL	
	JMP	START	; for continuous operation

EXECUTION:

OUTPUT:

NSRIT

Internal Quality Assurance Cell (IQAC)

Rubrics for Laboratory Course

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical excercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	**Criterion #1** Ability to setup, conduct experiments and take measurements/readings and present data	Unable to identify theoretical framework, measurement techniques, testing apparatus or model. Could not discuss experimental processes and protocols	Able to identify theoretical framework, measurement techniq Developing ues, testing apparatus or model. Could discuss experimental processes and protocols	Able to use theoretical framework, measurement techniques, testing apparatus or model. Able to apply constraint and assumption into the experimental design. Able to conduct experiment correctly and collect data.	Able to develop contingency or alternative plans. Able to evaluate data and relate to engineering phenomena for decision-making.	Able to develop contingency or alternative plans and anticipate problems during experiment. Able to formulate controls and evaluate alternatives of the experiment. Able to evaluate data and relate to engineering phenomena for decision-making.	
B. Results & Interpretation	**Criterion #2** Ability to analyze the data theoretically and logically to conclude experimental results	-	Unable to select and describe the techniques or methods of analyzing the data.	Able to select and use and apply appropriate techniques or methods to analyze the data.	Able to evaluate or compare data and make meaningful conclusion	Able to combine /orgranise more than one set of data, interpret data and make meaningful conclusion.	
	Criterion #3 Ability to interpret and discuss any discrepancies between theoretical and experimental results	-	Unable to identify how results relate/differ from theory or previous results.	Able to identify and verify how results relate/differ from theory or previous results	Able to verify and/or validate data and relate to engineering phenomena for decision making.	Able to verify and/or validate several sets of data and relates to engineering phenomena for decision making.	
C. Viva Voce	**Criterion #4** Demonstrate the ability to effectively respond to questions	-	Unable to listen carefully to questions and does not provide an appropriate answer, or is unable to answer questions	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C)/20] x 10**			

EXPERIMENT 6

Interfacing DAC to 8086

AIM: Programs to generate Sine and Triangular waveforms DAC interface.

APPARATUS:

1. 8086 microprocessor kit and DAC kit
2. Power supply
3. Key board

THEORY:

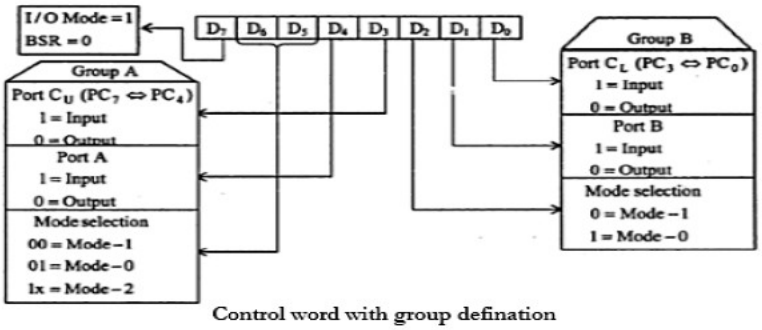
The digital to analog converter (DAC) is a device widely used to convert digital pulses to analog signals. There are two methods of creating a DAC; one is Binary weighted method and other one is R/2R Ladder method. The vast majority of integrated circuit DACs, including the DAC0808 uses the R/2R method since it can achieve a much higher degree of precision. The first criterion for judging a DAC is its resolution, which is a function of the number of binary inputs. In the MC1408 (DAC0808), the digital inputs are converted to current (I_{out}), and by connecting a resistor to the I_{out} pin, we convert the result to voltage. The total current provided by the Iout pin is a function of the binary numbers at the D0-D7 inputs of the DAC0808 and the reference current (I_{ref}), and is follows:

$$I_{out} = I_{ref}\left(\frac{D7}{2}+\frac{D6}{4}+\frac{D5}{8}+\frac{D4}{16}+\frac{D3}{32}+\frac{D2}{64}+\frac{D1}{128}+\frac{D0}{256}\right)$$ Generally $I_{ref} = 2mA$

DESCRIPTION OF THE CIRCUIT:

As can be seen in the circuit only 17 lines from the connectors are used totally. The port- A and port-B 8255 programmable. Peripheral interface are used as o/p ports. The digital inputs to the DAC are provided through port-A and port-B of 8255. The reference voltage for the DACs is derived from an on-board voltage regulator µA 723. It generates the voltage of about 8V. The offset voltage balancing of op-amps is done by making use of the two 10K pots provided. The analog outputs of DACs connecting to inverting inputs of op-amps µA 741 which act as current to voltage converter.

CONTROL WORD (8255):



Control word with group defination

	D7	D6	D5	D4	D3	D2	D1	D0
BITS	1	0	0	0	0	0	0	0
HEX CODE	8				0			

Sine Wave Generation:

AIM: Program to get sine wave in port-A using DAC.

APPARATUS:
1. 8086 microprocessor kit and 8259 kit

2. Power supply

3. Key board

Generating a sine wave:
To generate a sine wave, we first need a table whose values represent the magnitude of the sine of angles between 0 and 360 degrees. The values for the sine function varies from –1.0 to +1.0 for 0 to 360 degree angles. Full – scale output of the DAC is achieved when all the data inputs of the DAC are high. So to achieve the full scale, we use the following equation
$V_{out} = 5V + (5 X \sin \theta)$

Angle vs. voltage Magnitude for Sine Wave.

Angle θ (Degrees)	Sin θ	Vout (voltage Magnitude) 5V + (5V X sin θ)	Values sent to DAC(decimal) (voltage mag. X 25.6)	Equivalent Hex values
0	0.00	5.00	128	80H
30	0.50	7.50	192	C0H
60	0.866	9.33	238	EEH
90	1.00	10.00	255	FFH
120	0.866	9.33	238	EEH
150	0.50	7.50	192	C0H
180	0.00	5.00	128	80H
210	-0.50	2.50	64	40H
240	-0.866	0.669	17	11H
270	-1.00	0.00	0	0H
300	-0.866	0.669	17	11H
330	-0.50	2.50	64	40H
360	0.00	5.0	128	80H

ALGORITHM:
1. Load AL register with Control Word Register contents.

2. Initialize Control Word Register address in to DX and out the contents of AL into Control Word Register.
2. Make SI as pointer to memory to provide the input values and get counter value in CL
a. register.
3. Initialize DX register with port-A and out the AL contents to the port address mentioned in DX.

4. Repeat the loop until all values got outputted through the port A address.

5. Repeat the above procedure to get continuous waveform

PROGRAM:

PROGRAM STARTS AT 3000 (ORIGIN) MEMORY LOCATION

LABEL	MNEMONICS		COMMENTS
	MOV	AL,80	; load AL with Control Word Register contents i.e., 80 Mode 0 operation, port-A as output port.
	MOV	DX, 0FFE6	; initialize DX register with Control Word Register address
	OUT	DX,AL	; out contents of AL through Control Word Register address
AGAIN:	MOV	SI,2000	; initialize SI to memory to provide the inputs
	MOV	CL,0D	; get counter in CL register
START:	MOV	DX,0FFE0	; initialize DX register with port-A address
	MOV	AL,[SI]	; get the first hex value(equivalent to Angle) to AL from SI
	OUT	DX,AL	; out the contents of AL through port-A address
	INC	SI	; increments the memory of SI to get next value
	LOOP	START	; repeat the loop until all values got outputted through port A address
	JMP	AGAIN	; jumps to the specified label to repeat the steps

Code Table:

Physical Address		Label	Hex code	Mnemonic Operands	Comments
Segment address	Effective address				

OUTPUT:

Triangular Waveform Generation

AIM: Program to get triangular wave in port-A using DAC.

APPARATUS:
1. 8086 microprocessor kit and DAC kit

2. Power supply

3. Key board

ALGORITHM:
1. Load AL register with Control Word Register contents.
2. Initialize Control Word Register address into DX and out the contents of AL through the Control Word Register address.
3. Load initial value i.e., 00 to AL.
4. Initialize DX register with port-A and out the AL contents to the port A address.
5. Increment the AL contents and output through the port A address.
6. Compare AL with FF if it is not equal continue step-5, else continue step – 7.
7. Load AL with FF and output through the port A address.
8. Decrement the AL contents and output through the port A address.
9. Compare AL with 00 and check for equal condition if not equal repeat step-8.
10. Repeat from step-3.

PROGRAM:

PROGRAM STARTS AT 3000 (ORIGIN) MEMORY LOCATION

LABEL	MNEMONICS		COMMENTS
	MOV	AL,80	; load AL with Control Word Register contents ie., 80 Mode 0 operation, port-A as output port.
	MOV	DX, 0FFE6	; initialize DX register with Control Word Register address
	OUT	DX,AL	; out contents of AL through Control Word Register address
START:	MOV	AL,00	; load AL with initial value of triangular wave
BACK:	MOV	DX,0FFE0	; initialize DX register with port-A address
	OUT	DX,AL	; out the contents of AL through port-A address
	INC	AL	; increments AL contents to get next digital value
	CMP	AL, FF	; compare AL with FF
	JNZ	BACK	; check for Zero condition, if no zero jump to the specified label
	MOV	AL,0FF	; load AL with FF
BACK 1:	MOV	DX, 0FFE0	; initialize DX register with port A address
	OUT	DX, AL	; out the contents of AL through the port A address
	DEC	AL	; decrement AL contents
	CMP	AL, 00	; compare AL with 00
	JNZ	BACK 1	; if no zero jump to the specified label
	JMP	START	; jump to the specified label to obtain continuous wave form

EXECUTION:

G 3000

CODE TABLE: -

Physical Address		Label	Hex code	Mnemonic Operands	Comments
Segment address	Effective address				

OUTPUT:

(Square Wave)

a) AIM: Program to get square wave in port-A using DAC.

ALGORITHM:

1. Load AL register with Control Word Register contents.
2. Initialize Control Word Register address in to DX and out the contents of AL into Control Word Register.
3. Load initial value i.e., 00 to AL.
4. Initialize DX register with port-A and out the AL contents to the port address mentioned in DX.
5. Provide a delay.
6. Complement the contents of accumulator and repeat the step 4.

PROGRAM:

```
        MOV     AL,80
        MOV     DX, 0FFE6
        OUT     DX,AL
        MOV     AL,00
START:  MOV     DX,0FFE0
        OUT     DX,AL
        MOV     BX,00FF
BACK:   DEC     BX
        JNZ     BACK
        NOT     AL
        JMP     START
```

Physical Address		Label	Hex code	Mnemonic Operands	Comments
Segment address	Effective address				

NSRIT

Internal Quality Assurance Cell (IQAC)

Rubrics for Laboratory Course

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical excercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	**Criterion #1** Ability to setup, conduct experiments and take measurements/readings and present data	Unable to identify theoretical framework, measurement techniques, testing apparatus or model. Could not discuss experimental processes and protocols	Able to identify theoretical framework, measurement techniq Developing ues, testing apparatus or model. Could discuss experimental processes and protocols	Able to use theoretical framework, measurement techniques, testing apparatus or model. Able to apply constraint and assumption into the experimental design. Able to conduct experiment correctly and collect data.	Able to develop contingency or alternative plans. Able to evaluate data and relate to engineering phenomena for decision-making.	Able to develop contingency or alternative plans and anticipate problems during experiment. Able to formulate controls and evaluate alternatives of the experiment. Able to evaluate data and relate to engineering phenomena for decision-making.	
B. Results & Interpretation	**Criterion #2** Ability to analyze the data theoretically and logically to conclude experimental results	-	Unable to select and describe the techniques or methods of analyzing the data.	Able to select and use and apply appropriate techniques or methods to analyze the data.	Able to evaluate or compare data and make meaningful conclusion	Able to combine /orgranise more than one set of data, interpret data and make meaningful conclusion.	
	Criterion #3 Ability to interpret and discuss any discrepancies between theoretical and experimental results	-	Unable to identify how results relate/differ from theory or previous results.	Able to identify and verify how results relate/differ from theory or previous results	Able to verify and/or validate data and relate to engineering phenomena for decision making.	Able to verify and/or validate several sets of data and relates to engineering phenomena for decision making.	
C. Viva Voce	**Criterion #4** Demonstrate the ability to effectively respond to questions	-	Unable to listen carefully to questions and does not provide an appropriate answer, or is unable to answer questions	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C)/20] x 10**			

EXPERIMENT 7

STEPPER MOTOR INTERFACING

AIM: Program to rotate the stepper motor 180 clockwise and 180 anti - clock wise

APPARATUS:

1. 8086 microprocessor kit and stepper motor control kit

2. Power supply

3. Key board

THEORY:
Data acquisition and control represent the most popular applications of microprocessors, stepper motor control is a very popular application of microprocessor in control area, as stepper motor is capable of accepting pulses directly from the microprocessor and move accordingly.

There are two types of stepper motors

(a) Permanent magnet (PM)

(b) Variable reluctance (VR)

Permanent magnet stepper motors:

PM stepper motor consists of two stator windings A,B and a motor having two magnetic poles N and S.When a voltage +V is applied to stator winding A, a magnetic field Fa is generated. The rotor positions itself such that its poles lock with corresponding stator poles.
With the winding A is excited before, winding 'B' is now switched on to a voltage +V. This produces a magnetic field Fb in addition to Fa. The resulting magnetic field F makes an angle of 45 degrees. The motor consequently moves through 45 in anti-clock-wise direction, again to cause locking of rotor poles with corresponding stator poles.
While winding 'B' has voltage +V applied to it, winding 'A' is switched off. The rotor then moves through a further 45 degrees in anti-clockwise direction to align itself with stator field Fb. With voltage +V on winding B, a voltage –V is applied to winding A. Then the stator magnetic field has two components: Fa, Fb and their resultant F makes an angle of 135 degrees position.
A practical PM stepper motor will have 1.8 degrees step angle and 50 tooth on its rotor; There are eight main poles on the stator, each having five tooth in the pole face. The step angle is given by

A = 360/(N*K) degrees

Where N = number of rotor tooth
K = excitation sequence factor

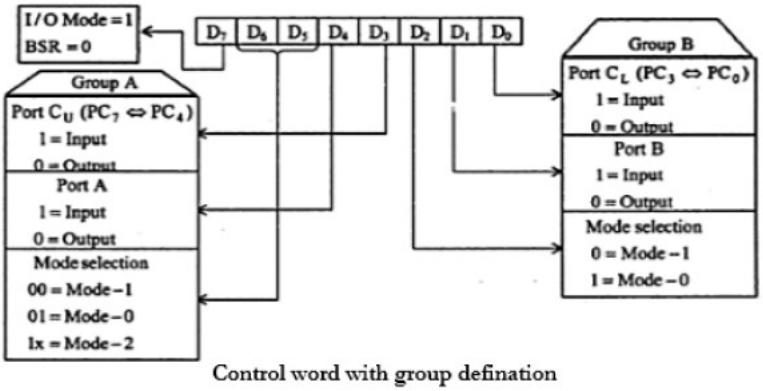
Variable Reluctance (VR) Stepper Motors:
There are twelve tooth on the stator and eight on the rotor. The rotor does not carry either a permanent magnet or winding; it is assembled from soft iron punchings. The stator is also assembled from soft iron punchings, and carries stator windings A,B and C. When stator winding A is excited, it creates a patterns of N and S poles. The rotor then positions itself as shown in fig(i), so as to minimize the reluctance of the magnetic circuit. When phase B is excited next, the rotor will move through 15 degrees to again seek minimum reluctance position. The step angle is given by
360(1 2) 1* 2 N N B N N□ □

Where N1 = number of stator tooth
N2 = number of rotor tooth
N1 and N2 are related by
N1 = N2+n = p*n
Where n = number of stator tooth per phase
p = number of phases

DESCRIPTION OF THE CIRCUIT:

The stepper motor interface uses four transistor pairs (SL100 & 2N3055) in a darlington pair configuration. Each darling ton pair is used to excite the particular winding of the motor connected to 4-pin connector on the interface. The inputs to these transistors are from the 8255 PPI I/O lines of the microprocessor trainer kit or from digital I/O card plugged in the PC. "Port A" lower nibble PA0, PA1, PA2, PA3 are the four lines brought out to the 26 pin FRC male connector (J1) on the interface module. The free – wheeling diodes across each winding protect transistors from switching transients.

CONTROL WORD (8255):



Control word with group defination

	D7	D6	D5	D4	D3	D2	D1	D0
BITS	1	0	0	0	0	0	0	0
HEX CODE	8				0			

ALGORITHM:

1. Load AL with Control Word.
2. Load DX with control word register address and out the contents of AL to the port address of Control Word Register
3. Load BX with (0064) and AL with 11.
4. Load DX with port A address (FFE0) and out the contents of AL to rotate the stepper motor for 1 step.
5. Provide delay.
6. Rotate left accumulator contents by one step.
7. Decrement BX contents and check for zero condition. If it is not zero continue step 4.
8. Again load BX with number of excitations given to the stepper motor (0064) and AL with 11 to get anti clockwise rotation.
9. Load DX with port A address (FFE0) and out the contents of AL.
10. Provide delay.
11. Rotate right AL contents and decrement BX register, if BX is not zero repeat step 9.

PROGRAM 1:

PROGRAM STARTS AT 3000 (ORIGIN) MEMORY LOCATION

LABEL	MNEMONICS		COMMENTS
	MOV	AL,80	; load Al with Control Word Register contents mode 0 operation, port A as output port.
	MOV	DX,8006	; initialize DX register with Control Word Register address
	OUT	DX,AL	; out the contents of AL through Control Word Register address
	MOV	AL,88	; initialize AL with 11 to enable step
START:	MOV	DX,8000	; initialize DX register with port A address
	OUT	DX,AL	; out the contents of AL through port A address
	CALL	DELAY	; check for zero condition
	ROL	AL,01	; rotate AL contents to left by one time to excite next step
	DEC	BX	; decrement BX register
	JN	AGAIN	; check for zero condition, if no zero jump to the specified label
	MOV	AL,11	; initialize AL with 11 to enable step
	MOV	BX,0064	; initialize BX register with 0064 (180° clock wise rotation)
	MOV	DX,0FFE 0	; initialize DX register with port A address
AGAIN2 :	OUT	DX,AL	; out the contents of AL through port A address
	MOV	CX,0FFF	; initialize CX register with some count to provide delay
BACK1:	LOOP	BACK1	; check for zero condition
	ROR	AL,1	; rotate AL contents to right by one time to excite next step
	DEC	BX	; decrement BX register
	JNZ	AGAIN2	; check for zero condition, if no zero jump to the specified label
	INT	03	; it is a software interrupt which tells the end of the program

EXECUTION:

OUTPUT:

PROGRAM 2

AIM: Program to rotate the stepper motor continuously clock wise.

ALGORITHM:

1. Load AL with 80.
2. Load DX with Control Word Register address (FFE6) and out the contents of AL to the port address of Control Word Register.
3. Load AL with 11.
4. Load DX with port A address (FFE0) and out the contents of AL to rotate the stepper motor for 1 step.
5. Provide delay.
6. Rotate right accumulator contents to rotate next step.
7. Jump unconditionally to step 4.

PROGRAM:

PROGRAM STARTS AT 3000 (ORIGIN) MEMORY LOCATION

LABEL	MNEMONICS		COMMENTS
	MOV	AL,80	; load Al with Control Word Register contents mode 0 operation, port A as output port.
	MOV	DX,0FFE6	; initialize DX register with Control Word Register address
	OUT	DX,AL	; out the contents of AL through Control Word Register address
	MOV	AL,11	; initialize AL with 11 to enable step
	MOV	DX,0FFE0	; initialize DX register with port A address
AGAIN:	OUT	DX,AL	; out the contents of AL through port A address
	MOV	CX,0FFF	; initialize CX register with some count to provide delay
BACK:	LOOP	BACK	; check for zero condition
	ROR	AL,01	; rotate AL contents to right by one time to excite next step
	JMP	AGAIN	; jump unconditionally to specified label to rotate the motor continuously

EXECUTION:

OUTPUT:

NSRIT

Internal Quality Assurance Cell (IQAC)

Rubrics for Laboratory Course

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical excercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	**Criterion #1** Ability to setup, conduct experiments and take measurements/readings and present data	Unable to identify theoretical framework, measurement techniques, testing apparatus or model. Could not discuss experimental processes and protocols	Able to identify theoretical framework, measurement techniq Developing ues, testing apparatus or model. Could discuss experimental processes and protocols	Able to use theoretical framework, measurement techniques, testing apparatus or model. Able to apply constraint and assumption into the experimental design. Able to conduct experiment correctly and collect data.	Able to develop contingency or alternative plans. Able to evaluate data and relate to engineering phenomena for decision-making.	Able to develop contingency or alternative plans and anticipate problems during experiment. Able to formulate controls and evaluate alternatives of the experiment. Able to evaluate data and relate to engineering phenomena for decision-making.	
B. Results & Interpretation	**Criterion #2** Ability to analyze the data theoretically and logically to conclude experimental results	-	Unable to select and describe the techniques or methods of analyzing the data.	Able to select and use and apply appropriate techniques or methods to analyze the data.	Able to evaluate or compare data and make meaningful conclusion	Able to combine /orgranise more than one set of data, interpret data and make meaningful conclusion.	
	Criterion #3 Ability to interpret and discuss any discrepancies between theoretical and experimental results	-	Unable to identify how results relate/differ from theory or previous results.	Able to identify and verify how results relate/differ from theory or previous results	Able to verify and/or validate data and relate to engineering phenomena for decision making.	Able to verify and/or validate several sets of data and relates to engineering phenomena for decision making.	
C. Viva Voce	**Criterion #4** Demonstrate the ability to effectively respond to questions	-	Unable to listen carefully to questions and does not provide an appropriate answer, or is unable to answer questions	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C)/20] x 10**			

INTRODUCTION TO KEIL SOFTWARE

Keil compiler is software used where the machine language code is written and compiled. After compilation, the machine source code is converted into hex code, which is dumped into the micro controller for further processing. Keil compiler also supports C language code.

Steps to write an assembly language program in KIEL and how to compile it:

1. Install the Keil software in the PC in any of the drives.

2. After installation, an icon will be created with the name 'Keil uvision3'. Just drag this icon onto the desktop so that it become easy whenever you try to write programs in keil.

3. Double click on this icon to start the keil compiler.

4. A page opens with different options in it showing the project workspace at the leftmost corner side, output window in the bottom and an ash coloured space for the program to be written.

5. Now to start using the keil, click on the option 'project'

6. A small window opens showing the options like new project, import project, open project etc. Click on 'new project'.

7. A small window with the title bar "create new project" opens. The window asks the user to give the project name with which it should be created and the destination location. The project can be created in any of the drives available. You can create a new folder and then new file or can create new file directly.

8. After the file is saved in the given destination location, a window opens where a list of vendors will be displayed and you have to select the device for the target you have created.

9. The most widely used vendor is Atmel. So click on Atmel and now the family of micro controllers manufactured by Atmel opens. You can select any one of the microcontrollers according to the requirement.

10. When you click on any one of the micro controllers, the features of that particular micro controller will be displayed on the right side of the page. The most appropriate micro controller with which most of the projects can be implemented is the AT89C51. Click on this micro controller and have a look at its features. Now click on "OK" to select this micro controller

11. A small window opens asking whether to copy the startup code into the file you have created just now. Just click on "NO" to proceed further

12. Now we can see "TARGET" and "SOURCE GROUP" created in the project work space.

13. Now click on File and then New, a new page opens and you can start writing program in it.

14. Save the program with .asm extension.

15. Add this file to the target by giving right click on the source group. A list of options open , in that select "add files to source group"

16. Right click on the topic and select first option, "options for target". "Click on target".

17. Enter the crystal frequency 11.0592 MHz in the text area.Put tick on the use on chip ROM.

18. Now click the option “output” and give any name to the hex file to be created “Name of the executable” text area and tick “ Create hex file” option present in the same window.

19. Now to check the errors in the program click on icon below the “open file” icon, which is ‘build target’ icon.

20. To check for the output there are several windows like, serial window, memory window, project window etc. Depends on icon select appropriate window.

21. Icon with letter ”d” indicates debug mode.

22. Click on this icon and now click on the option “view” and select the appropriate window to check for the output.

23. Click on icon debug again to come out of debug mode.

PROLOAD:

1. PROLOAD is a software which accepts only hex files used to dump hex code into microcontroller

2. Connect programmer kit to PC through serial cable. Power up the programmer kit.

3. Place microcontroller in GIF socket.

4. Click on PROLOAD icon in the PC, Click on browse option to select hex file to be dumped into microcontroller and then click on “Auto program”. To program the microcontroller with hex file

5. Microcontroller status can be seen on the window in the bottom of the page

EXPERIMENT -8

FINDING NUMBER OF 1'S & NUMBER OF 0'S

AIM: To write an Assembly Program to Find number of 1's and number of 0's in a given 8-bit number.

APPARATUS: Keil U Vision IDE software
Personal Computer
Keyboard

ALGORITHM:

Step1: Start.

Step2: Initialize R1 register with 00H

Step3: Initialize R2 register with 00H

Step 4: Move 08H into R7 register

Step5: Load accumulator with value 97H.

Step6: Rotate left the contents of Accumulator with carry.

Step7: Jump if there is carry to the label NEXT.

Step8: Increment the contents of R1 register.

Step9: short jump to the label HERE

Step10: Increment the contents of R2 register

Step11: Decrement the contents of R7 register and jump if not equal to zero**.**

Step12: stop

PROGRAM:

```
        MOV  R1,#00H
        MOV  R2,#00H
        MOV  R7,#08H
        MOV A,#97H
AGAIN:  RLC A
        JC NEXT
        INC R1
        SJMP HERE
NEXT:   INC R2
HERE:   DJNZ R7,AGAIN
        END
```

VIVA QUESTIONS:

1. Discuss the different register set in 8051 Microcontroller
2. Specify the internal RAM internal ROM of 8051 Microcontroller

Observations:

RESULT:

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

Department Electronics and Communication Engineering
Rubrics for evaluation of Laboratory

Name of student:
Roll No:
Experiment No:

	Not Satisfactory	Satisfactory	Good	Excellent	Score
Problem Underas -tanidng (2)	Unable to explain the problem statement and its requirements	Able to explain the problem statement	Able to explain the problem statement, fairlyon other requirements	Able to give full explanation on problem statement and on input and output	
Ability to conduct experim-ent (4)	Has poor understanding of software tools including its purpose and unable to conduct experiment on his own , lab instructor provides help in every step of the experiment	Has good understanding of software tools including its purpose and able to conduct experiment with some help of , lab instructor provides help in every step of the experiment	Has good understandin g of software tools including its purpose and able to conduct experiment with negligible help from the lab instructor	Fully understanding th e software tools including its purpose and able to conduct the experiment by own	
Program ,Simulati -ons and Results (4)	Has poor simulation skills unable to write code and has no ability to identify and remove the coding errors even with instructor's help. Unable to produce output results	Has good Simulations Skills and be able to write the code with less errors and Unable to Explain the output	Has good simulation skills and able to make the acceptable coding logic with less errors.	Has good simulation skills and able to make the acceptable coding logic with desired outputs for a given problem.	
				Total	
				Signature	

EXPERIMENT -9

AVERAGE OF N-Numbers

AIM: To write an assembly language program(ALP) for average of N- numbers in an array using KEIL Software.

APPARATUS:

1. Personal Computer
2. Keil U Vision IDE software
3. Keyboard

ALGORITHM:

Step1: Start.

Step2: Assign R0 register with Memory location 050H,R1 with 0AH

Step3: Initialize R2 register with 00H

Step 4: Move 0AH into R3 register

Step5: Load accumulator with value available in Memory Location 050H pointed by R0 Register.

Step6: Add the contents of Accumulator with R2.

Step7: Increment R0.

Step8: Repeat the process till R1 Count become zero.

Step9: Load the sum value into Accumulator

Step10: Load the value in R3 into B Register.

Step11: Perform Division

Step12: Load the quotient available in Accumulator into R4 Register and Remainder in B Register into R5 Register

Step13: Stop

PROGRAM:

```
        ORG 0000h
        LJMP main
        ORG 0000h
 main :MOV R0,#050h
        MOV R1,#0Ah
        MOV R2,#00h
        MOV A,R1
        MOV R3,A
LOOP:MOV A,@R0
```

```
ADD A,R2
        MOV R2,A
        INC R0
        DJNZ R1,LOOP
        MOV B,R3
        MOV  A,R2
        DIV AB
        MOV R4,A
        END
```

VIVA QUESTIONS:

1. Discuss the various Addressing Modes in 8051 Microcontroller
2. Illustrate different Arithmetic instructions supported by 8051 Microcontroller

RESULT:

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

Department Electronics and Communication Engineering
Rubrics for evaluation of Laboratory

Name of student:
Roll No:
Experiment No:

	Not Satisfactory	Satisfactory	Good	Excellent	Score
Problem Underas -tanidng (2)	Unable to explain the problem statement and its requirements	Able to explain the problem statement	Able to explain the problem statement, fairlyon other requirements	Able to give full explanation on problem statement and on input and output	
Ability to conduct experim-ent (4)	Has poor understanding of software tools including its purpose and unable to conduct experiment on his own , lab instructor provides help in every step of the experiment	Has good understanding of software tools including its purpose and able to conduct experiment with some help of , lab instructor provides help in every step of the experiment	Has good understandin g of software tools including its purpose and able to conduct experiment with negligible help from the lab instructor	Fully understanding th e software tools including its purpose and able to conduct the experiment by own	
Program ,Simulati -ons and Results (4)	Has poor simulation skills unable to write code and has no ability to identify and remove the coding errors even with instructor's help. Unable to produce output results	Has good Simulations Skills and be able to write the code with less errors and Unable to Explain the output	Has good simulation skills and able to make the acceptable coding logic with less errors.	Has good simulation skills and able to make the acceptable coding logic with desired outputs for a given problem.	
				Total	
				Signature	

EXPERIMENT – 10

INTERFACING TRAFFIC LIGHT CONTROL TO 8051

AIM: To control the traffic by interfacing traffic light controller with 8051 microcontroller.

APPARATUS:

1.Personal Computer

2.Keyboard

3.Keil U vision IDE Software

4.Hardware Interfacing Kit

PROGRAM:

```
#include <reg51.h>
sbit RA=p1^0;
sbit YA=p1^1;
sbit GA=p1^2;
sbit RB=p3^2;
sbit YB=p3^3;
sbit GB=p3^4;
sbit RC=p3^5;
sbit YC=p3^6;
sbit GC=p3^7;
sbit RD=p1^3;
sbit YD=p1^4;
sbit GD=p1^5;
void delay (void)
{
 unsigned int i,j
for(i=0;i<200;i++)
for(j=0;j<500;j++)
}
void superdelay ()
{
unsigned int i;
for(i=0;i<25;i++)
delay
}
void main ()
{
 p3=0; while(1)
 {
   RA=0;
  GA=1;
  YA=0;
  RB=1;
  GB=0;
  YB=0;
  RC=1;
  GC=0;
  YC=0;
  RD=1;
  GD=0;
```

```
YD=0;
superdelay()
GA=0;
YA=1;
delay();
RA=1;
   GA=0;
  YA=0;
  RB=0;
  GB=1;
  YB=0;
  RC=1;
  GC=0;
  YC=0;
  RD=1;
  GD=0;
  YD=0;
superdelay();
GB=0;
YB=1;
delay();
 RA=1;
 GA=0;
 YA=0;
 RB=1;
 GB=0;
 YB=0;
 RC=0;
 GC=1;
 YC=0;
 RD=1;
 GD=0;
 YD=0;
superdelay();
GC=0;
YC=0;
RD=0;
GD=1;
YD=0;
superdelay(); GD=0; YD=1; delay();
}
}
```

RESULT:

.

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

Department Electronics and Communication Engineering
Rubrics for evaluation of Laboratory

Name of student:
Roll No:
Experiment No:

	Not Satisfactory	Satisfactory	Good	Excellent	Score
Problem Underas -tanidng (2)	Unable to explain the problem statement and its requirements	Able to explain the problem statement	Able to explain the problem statement, fairlyon other requirements	Able to give full explanation on problem statement and on input and output	
Ability to conduct experim-ent (4)	Has poor understanding of software tools including its purpose and unable to conduct experiment on his own , lab instructor provides help in every step of the experiment	Has good understanding of software tools including its purpose and able to conduct experiment with some help of , lab instructor provides help in every step of the experiment	Has good understandin g of software tools including its purpose and able to conduct experiment with negligible help from the lab instructor	Fully understanding th e software tools including its purpose and able to conduct the experiment by own	
Program ,Simulati -ons and Results (4)	Has poor simulation skills unable to write code and has no ability to identify and remove the coding errors even with instructor's help. Unable to produce output results	Has good Simulations Skills and be able to write the code with less errors and Unable to Explain the output	Has good simulation skills and able to make the acceptable coding logic with less errors.	Has good simulation skills and able to make the acceptable coding logic with desired outputs for a given problem.	
				Total	
				Signature	

EXPERIMENT – 11

TIMER MODE PROGRAMMING

AIM: To generate square wave using Timer Mode Programming.

APPARATUS:

1. Personal Computer
2. Keil U Vision IDE software
3. Keyboard

PROGRAM:

```
MOV TMOD #01
HERE : MOV TL0,#0F2H
MOV TH0,#FFH
CPL P1.5
SETB TR0
AGAIN : JNB TF0,AGAIN
CLR TR0
CLR TF0
SJMP HERE
```

RESULT:

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

Department Electronics and Communication Engineering
Rubrics for evaluation of Laboratory

Name of student:
Roll No:
Experiment No:

	Not Satisfactory	Satisfactory	Good	Excellent	Score
Problem Underas -tanidng (2)	Unable to explain the problem statement and its requirements	Able to explain the problem statement	Able to explain the problem statement, fairlyon other requirements	Able to give full explanation on problem statement and on input and output	
Ability to conduct experim-ent (4)	Has poor understanding of software tools including its purpose and unable to conduct experiment on his own , lab instructor provides help in every step of the experiment	Has good understanding of software tools including its purpose and able to conduct experiment with some help of , lab instructor provides help in every step of the experiment	Has good understandin g of software tools including its purpose and able to conduct experiment with negligible help from the lab instructor	Fully understanding th e software tools including its purpose and able to conduct the experiment by own	
Program ,Simulati -ons and Results (4)	Has poor simulation skills unable to write code and has no ability to identify and remove the coding errors even with instructor's help. Unable to produce output results	Has good Simulations Skills and be able to write the code with less errors and Unable to Explain the output	Has good simulation skills and able to make the acceptable coding logic with less errors.	Has good simulation skills and able to make the acceptable coding logic with desired outputs for a given problem.	
				Total	
				Signature	

EXPERIMENT 12

MULTIPLY OF 2 16-BIT BINARY NUMBERS

AIM: To write an assembly program to multiply of 2 16-bit binary numbers.

APPARATUS:
l. ARM CORTEX M3 Board
2. KEIL MDK ARM

PROCEDURE:

a. First create a folder on desktop with some name.
b. Open KEIL Âμ Visions & go to project, click on class Project.
c. Click on project and click onÂμ Vision Project & give some name and save.
d. A window will appear and go to intel and extract intel options and select A T8965 l and Press ok.
e. A window will appear and click on No, expand Target-1 & source group-1
f. Go to file , click on new &type the program.
g. Go to file. Click on "save as" . Now give file name .asm and save.
h. Right click on "source group" and go to add & files to group.
i. A window will appear and change type of files to asm source file, click on ADD &CLOSE.
j. Click on Target-1 and go to options for "Target-1" & change crystal oscillator (XTAL) frequency as 11.0592 Mhz.
k. Click on output tab ⟶ Create HEX file.
⟶ Click on Ok.
l. Click on "translate current file" to check others
m. Click on "Build Targett" which size of program.
n. Click on" Rebuild all target files" and go to debug & select start/stop sessions where the warning will be generated click on ok.
o. Click on "step into" { } and check the program with the register. Again repeat the same process until it completes all the instructions in the program.
p. Now go to view& click on memory window then address box will be appeared at the bottom.
q. ln the address type the given Address location & enter .
r. Then the result will be stored in desired address locations & it is observed.
s. Now go to created folder & click on HEX file.
t. HEX code will be generated

PROGRAM:

```
AREA MULTIPLY, CODE, READONL Y ENTRY
START
LOR Rl OX1234
LOR R2 Ox2345
Mui r3, rl, r2
NOP
NOP
END
```

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

Department Electronics and Communication Engineering
Rubrics for evaluation of Laboratory

Name of student:
Roll No:
Experiment No:

	Not Satisfactory	Satisfactory	Good	Excellent	Score
Problem Underas -tanidng (2)	Unable to explain the problem statement and its requirements	Able to explain the problem statement	Able to explain the problem statement, fairlyon other requirements	Able to give full explanation on problem statement and on input and output	
Ability to conduct experim-ent (4)	Has poor understanding of software tools including its purpose and unable to conduct experiment on his own , lab instructor provides help in every step of the experiment	Has good understanding of software tools including its purpose and able to conduct experiment with some help of , lab instructor provides help in every step of the experiment	Has good understandin g of software tools including its purpose and able to conduct experiment with negligible help from the lab instructor	Fully understanding th e software tools including its purpose and able to conduct the experiment by own	
Program ,Simulati -ons and Results (4)	Has poor simulation skills unable to write code and has no ability to identify and remove the coding errors even with instructor's help. Unable to produce output results	Has good Simulations Skills and be able to write the code with less errors and Unable to Explain the output	Has good simulation skills and able to make the acceptable coding logic with less errors.	Has good simulation skills and able to make the acceptable coding logic with desired outputs for a given problem.	
				Total	
				Signature	

EXPERIMENT 13

SUM OF FIRST 10 INTEGERS

AIM: To write an assembly program to find the sum of first 10 integers.

APPARATUS:
l. ARM CORTEX M3 Board
2. KEIL MDK ARM

PROCEDURE:

a. First create a folder on desktop with some name.
b. Open KEIL Âμ Visions & go to project, click on class Project.
c. Click on project and click onÂμ Vision Project & give some name and save.
d. A window will appear and go to intel and extract intel options and select A T8965 l and Press ok.
e. A window will appear and click on No, expand Target-1 & source group-1
f. Go to file , click on new &type the program.
g. Go to file. Click on "save as" . Now give file name .asm and save.
h. Right click on "source group" and go to add & files to group.
1. A window will appear and change type of files to asm source file, click on ADD &CLOSE.
j. Click on Target-1 and go to options for "Target-1" & change crystal oscillator (XTAL) frequency as 11.0592 Mhz.
k. Click on output tab ⟶ Create HEX file.
⟶ Click on Ok.
l. Click on "translate current file" to check others
m. Click on "Build Targett" which size of program.
n. Click on" Rebuild all target files" and go to debug & select start/stop sessions where the warning will be generated click on ok.
o. Click on "step into" { ⤵} and check the program with the register. Again repeat the same process until it completes all the instructions in the program.
p. Now go to view& click on memory window then address box will be appeared at the bottom.
q. ln the address type the given Address location & enter .
r. Then the result will be stored in desired address locations & it is observed.
s. Now go to created folder & click on HEX file.
t. HEX code will be generated

PROGRAM: -

```
AREA addsum, CODE, READONL Y ENTRY
START
MOY RO, #10
MOVRl, RO
ADDIT: SUBS Rl, Rl, #1
CMP Rt, #0
BEQ STOP
ADD R3, RO,Rl
MOVRO,R3
BNEADDIT
STOP NOP
NOP
NOP
END
```

RESULT:

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

Department Electronics and Communication Engineering
Rubrics for evaluation of Laboratory

Name of student:
Roll No:
Experiment No:

	Not Satisfactory	Satisfactory	Good	Excellent	Score
Problem Underas -tanidng (2)	Unable to explain the problem statement and its requirements	Able to explain the problem statement	Able to explain the problem statement, fairlyon other requirements	Able to give full explanation on problem statement and on input and output	
Ability to conduct experim-ent (4)	Has poor understanding of software tools including its purpose and unable to conduct experiment on his own , lab instructor provides help in every step of the experiment	Has good understanding of software tools including its purpose and able to conduct experiment with some help of , lab instructor provides help in every step of the experiment	Has good understandin g of software tools including its purpose and able to conduct experiment with negligible help from the lab instructor	Fully understanding th e software tools including its purpose and able to conduct the experiment by own	
Program ,Simulati -ons and Results (4)	Has poor simulation skills unable to write code and has no ability to identify and remove the coding errors even with instructor's help. Unable to produce output results	Has good Simulations Skills and be able to write the code with less errors and Unable to Explain the output	Has good simulation skills and able to make the acceptable coding logic with less errors.	Has good simulation skills and able to make the acceptable coding logic with desired outputs for a given problem.	
				Total	
				Signature	

EXPERIMENT 14

TOGGLE LED EVERY SECOND USING TIMER INTERRUPT

AIM: To write an assembly program to toggle LED every second using timer interrupt.

APPARATUS:
l. ARM CORTEX M3 Board
2. KEIL MDK ARM

PROCEDURE:

a. Create a folder on desktop with some name.
b. Open KEIL Âµ Visions & go to project, click on class Project.
c. Click on project and click on µ Vision Project & give some name and save.
d. A window will appear and go to intel and extract intel options and select A T8965 l and Press ok.
e. A window will appear and click on No, expand Target-1 & source group-1
f. Go to file , click on new &type the program.
g. Go to file. Click on "save as" . Now give file name .asm and save.
h. Right click on "source group" and go to add & files to group.
i. A window will appear and change type of files to asm source file, click on ADD &CLOSE.
j. Click on Target-1 and go to options for "Target-1" & change crystal oscillator (XTAL) frequency as 11.0592 Mhz.
k. Click on output tab ⟶ Create HEX file.
⟶ Click on Ok.
l. Click on "translate current file" to check others
m. Click on "Build Targett" which size of program.
n. Click on" Rebuild all target files" and go to debug & select start/stop sessions where the warning will be generated click on ok.
o. Click on "step into" { ⬚} and check the program with the register. Again repeat the same process until it completes all the instructions in the program.
p. Now go to view& click on memory window then address box will be appeared at the bottom.
q. ln the address type the given Address location & enter .
r. Then the result will be stored in desired address locations & it is observed.
s. Now go to created folder & click on HEX file.
t. HEX code will be generated

PROGRAM: -

```
sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim1, &sMasterConfig) != HAL_OK)
  {
Error_Handler();
  }
}
/** Configure pins as
     * Analog
     * Input
     * Output
     * EVENT_OUT
     * EXTI
```

```
*/
static void MX_GPIO_Init(void)
{
GPIO_InitTypeDefGPIO_InitStruct;
  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOE_CLK_ENABLE();
  /*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_5, GPIO_PIN_RESET);
  /*Configure GPIO pin : PE5 */
GPIO_InitStruct.Pin = GPIO_PIN_5;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOE, &GPIO_InitStruct);
}
/* USER CODE BEGIN 4 */
/* USER CODE END 4 */
/**
  * @brief  This function is executed in case of error occurrence.
  * @param  None
  * @retvalNone
  */
voidError_Handler(void)
{
  /* USER CODE BEGIN Error_Handler */
  /* User can add his own implementation to report the HAL error return state */
while(1)
  {
  }
  /* USER CODE END Error_Handler */
}
#ifdef USE_FULL_ASSERT
/**
   * @brief Reports the name of the source file and the source line number
   * where the assert_param error has occurred.
   * @param file: pointer to the source file name
   * @param line: assert_param error line source number
   * @retvalNone
   */
voidassert_failed(uint8_t* file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
  /* User can add his own implementation to report the file name and line number,
ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
  /* USER CODE END 6 */
}
```

```
#endif
/**
  * @}
  */
/**
  * @}
*/
static uint32_t counter = 0;
if (counter >= 500)
{
HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
counter = 0;
 }
counter++;
```

RESULT:

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

Department Electronics and Communication Engineering
Rubrics for evaluation of Laboratory

Name of student:
Roll No:
Experiment No:

	Not Satisfactory	Satisfactory	Good	Excellent	Score
Problem Underas -tanidng (2)	Unable to explain the problem statement and its requirements	Able to explain the problem statement	Able to explain the problem statement, fairlyon other requirements	Able to give full explanation on problem statement and on input and output	
Ability to conduct experim-ent (4)	Has poor understanding of software tools including its purpose and unable to conduct experiment on his own , lab instructor provides help in every step of the experiment	Has good understanding of software tools including its purpose and able to conduct experiment with some help of , lab instructor provides help in every step of the experiment	Has good understandin g of software tools including its purpose and able to conduct experiment with negligible help from the lab instructor	Fully understanding th e software tools including its purpose and able to conduct the experiment by own	
Program ,Simulati -ons and Results (4)	Has poor simulation skills unable to write code and has no ability to identify and remove the coding errors even with instructor's help. Unable to produce output results	Has good Simulations Skills and be able to write the code with less errors and Unable to Explain the output	Has good simulation skills and able to make the acceptable coding logic with less errors.	Has good simulation skills and able to make the acceptable coding logic with desired outputs for a given problem.	
				Total	
				Signature	