FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

-

DEPARTMENT OF
ELECTRONICS AND COMMUNICATION ENGINEERING

EC6513 – MICROPROCESSOR AND MICROCONTROLLERLABORATORY

V SEMESTER - R 2013

LABORATORY MANUAL

firstranker.com
www.FirstRanker.com

Name : _____

Register No. : _____

Section : _____

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

EC6513 – MICROPROCESSOR AND MICROCONTROLLER LABORATORY

COURSE OBJECTIVES

C01 - To learn the working of ARM processor

C02 - To introduce ALP concepts and features

C03 - To write ALP for arithmetic and logical operations in 8086 and 8051

C04 - To differentiate Serial and Parallel Interface

C05 - To interface different I/Os with Microprocessors

C06 - To be familiar with MASM

.

COURSE OUTCOMES

1. Write programs in ARM for a specific Application
2. Write ALP Programs for fixed and Floating Point and Arithmetic
3. Interface different I/Os with processor and Generate waveforms using Microprocessors
4. Execute Programs in 8051
5. Assess the performance of optical devices: light sources, fibers and detectors from both physical and system point of view

INTRODUCTION

This course illustrates the concepts covered in course CS6412 and provides students with hands-on experience in Assembly language programs.

PREREQUISITE:

1. Digital Electronics

EC6513 – MICROPROCESSOR AND MICROCONTROLLER LABORATORY

Lecture No./Session No.	Name of the Experiment	Course objectives	Proposed date
1,2,3	Introduction to lab	CO1	
CYCLE I - 8086 PROGRAMS USING KITS AND MASM			
4,5,6	Basic arithmetic and Logical operations	CO2	

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

7,8,9	Decimal arithmetic and Code conversion	CO3	
10,11,12	Move a data block without overlap	CO3	
	Matrix operations		
13,14,15	Searching and Sorting a sting in a given array	CO2	
CYCLE 2 - PERIPHERALS AND INTERFACING EXPERIMENTS			
16,17,18	D/A interface and Waveform Generation	CO4	
19,20,21	A/D interface	CO5	
22,23,24	Key board and Display	CO5	
25,26,27	Stepper motor control	CO2	
	Traffic light control	CO5	
28,29,30	Digital clock	CO6	
	Serial interface and Parallel interface	CO4	
CYCLE III - 8051 PROGRAMS USING KITS AND MASM			
31,32,33	Basic arithmetic and Logical operations	CO1	
34,35,36	Square and Cube program, Find 2's complement of a number	CO5	
	Unpacked BCD to ASCII		
37,38,39	Password checking and Print RAM size with system date	CO6	
	Counters and Time Delay		
ADDITIONAL EXPERIMENTS BEYOND THE SYLLABUS			
40,41,42	Sorting of an array using 8051	CO2	
43,44,45	Separate odd and even numbers using 8086	CO3	
	Sum of N consecutive numbers using 8086		

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

VISION

College of Engineering is committed to provide highly disciplined, conscientious and enterprising professionals conforming to global standards through value based quality education and training.

MISSION

- To provide competent technical manpower capable of meeting requirements of the industry
- To contribute to the promotion of Academic Excellence in pursuit of Technical Education at different levels
- To train the students to sell his brawn and brain to the highest bidder but to never put a price tag on heart and soul

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VISION

To impart professional education integrated with human values to the younger generation, so as to shape them as proficient and dedicated engineers, capable of providing comprehensive solutions to the challenges in deploying technology for the service of humanity

MISSION

- To educate the students with the state-of-art technologies to meet the growing challenges of the electronics industry
- To carry out research through continuous interaction with research institutes and industry, on advances in communication systems
- To provide the students with strong ground rules to facilitate them for systematic learning, innovation and ethical practices

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. Fundamentals

To impart students with fundamental knowledge in Mathematics, Science and fundamentals of engineering that will would them to be successful professionals

2. Core Competence

To provide students with sound knowledge in engineering and experimental skills to identify complex software problems in industry and to develop practical solution for them

3. Breadth

To provide relevant training and experience to bridge the gap between theory and practice this enables to find solutions for real time problem in industry and organization and to design products requiring interdisciplinary skills

4. Professionalism skills

To bestow students with adequate training and provide opportunities to work as team that will build up their communication skills, individual leadership and supportive qualities and to develop them to adapt and work in ever changing technologies

5. Lifelong Learning

To develop the ability of students to establish themselves as professionals in Computer Science and Engineering and to create awareness about the need for lifelong learning and pursuing advanced degrees

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

PROGRAMME OUTCOMES (POs)

a) To apply basic knowledge of Mathematics, Science and Engineering fundamentals in Computer Science and Engineering field

b) To design and conduct experiments as well as to analyze and interpret and apply the same in the career

c) To design and develop innovative and creative software applications

d) To understand a complex real world problems and develop an efficient practical solutions

e) To create, select and apply appropriate technique, resources, modern engineering and IT tools

f) To understand their roles as professionals and give the best to the society

g) To develop a system that will meet expected need with realistic constraints such as economical, environmental, social, political, ethical, safe and sustainable

h) To communicate effectively and make others understand exactly what they are trying to convey in both verbal and written forms

i) To work in a team as team member or a leader and make unique contributions and work with coordination

j) To engage lifelong learning and exhibit their technical skills

k) To develop and manage projects in multidisciplinary environments

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

EC6513 – MICROPROCESSOR AND MICROCONTROLLER LABORATORY

SYLLABUS

COURSE OBJECTIVES

- To learn the working of ARM processor
- To introduce ALP concepts and features
- To write ALP for arithmetic and logical operations in 8086 and 8051
- To differentiate Serial and Parallel Interface
- To interface different I/Os with Microprocessors
- To be familiar with MASM

LIST OF EXPERIMENTS:

8086 Programs using kits and MASM

1. Basic arithmetic and Logical operations
2. Move a data block without overlap
3. Code conversion, decimal arithmetic and Matrix operations.
4. Floating point operations, string manipulations, sorting and searching
5. Password checking, Print RAM size and system date
6. Counters and Time Delay

Peripherals and Interfacing Experiments

7. Traffic light control
8. Stepper motor control
9. Digital clock
10. Key board and Display
11. Printer status
12. Serial interface and Parallel interface
13. A/D and D/A interface and Waveform Generation 8051

Experiments using kits and MASM

14. Basic arithmetic and Logical operations
15. Square and Cube program, Find 2"s complement of a number
16. Unpacked BCD to ASCII

COURSE OUTCOMES

- Write programs in ARM for a specific Application
- Write ALP Programs for fixed and Floating Point and Arithmetic
- Interface different I/Os with processor and Generate waveforms using Microprocessors
- Execute Programs in 8051
- Assess the performance of optical devices: light sources, fibers and detectors from both physical and system point of view

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

EC6513 – MICROPROCESSOR AND MICROCONTROLLER LABORATORY

CONTENTS

Sl. No.	Name of the Experiment	Page No.
CYCLE 1 - 8086 PROGRAMS USING KITS AND MASM		
1	Basic arithmetic and Logical operations	10
2	Decimal arithmetic and Code conversion	15
3	Move a data block without overlap	21
4	Matrix operations	24
5	Searching and Sorting a sting in a given array	27
6	Password checking and Print RAM size with system date	33
7	Counters and Time Delay	36
CYCLE 2 - PERIPHERALS AND INTERFACING EXPERIMENTS		
8	Stepper motor control	39
9	Traffic light control	42
10	Digital clock	47
11	Serial interface and Parallel interface	50
12	A/D interface	54
13	Key board and Display	57
14	D/A interface and Waveform Generation	60
CYCLE 3 - 8051 PROGRAMS USING KITS AND MASM		
15	Basic arithmetic and Logical operations	64
16	Square and Cube program, Find 2's complement of a number	69
17	Unpacked BCD to ASCII	71
ADDITIONAL EXPERIMENTS BEYOND THE SYLLABUS		
18	Sorting of an array using 8051	74
19	Separate odd and even numbers using 8086	77
20	Sum of N consecutive numbers using 8086	80

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.1 BASIC ARITHMETIC AND LOGICAL OPERATIONS

Aim:

To write an Assembly Language Program to perform Arithmetic operation and to execute using 8086 microprocessor

Apparatus required:

1. 8086 Kit
2. Power cable
3. Keyboard

Procedure:

To enter program in Trainer Kit

1. Press 'RESET' key
2. Press 'A(Address)' key
3. Press "Enter" key
4. Enter the starting address (16 bit)
5. Press "Enter" key
6. Type the Assembly Language Program
7. Press 'RESET' key

To executive the program

8. Press "GO"
9. Type the starting address to execute the program
10. Press 'RESET' key

To verify the result

11. Type "SB(Substitute Byte)"
12. Enter the output address location to check the result

Arithmetic Operation:

1. Addition:

Algorithm:

1. Clear BL register to store carry
2. Get the addend in AX register pair
3. Get the Augend in CX register pair
4. Add Augend and addend
5. Check for carry, If it is one go to step 6 else go to step 7

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

6. Increments carry (BL register by 1)
7. Store the sum in memory from AX register pair
8. Store the carry in memory from BL register
9. Stop

Program:

Address	Label	Mnemonics
1000		MOV BL, 00H
1003		MOV AX, 1010H
1007		MOV CX, 2020H
100B		CLC
100C		ADD AX, CX
100E		JNC Label1
1010		INC BL
1012	Label 1	MOV [1200], AX
1016		MOV [1202], BL
101A		HLT

Precaution:

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

Tabulation:

Input		Output	
Address	Data	Address	Data

firstranker.com
www.FirstRanker.com

2. Subtraction:

Algorithm:

1. Clear BL register for storing borrow
2. Get the minuend in AX register pair
3. Get the subtrahend in CX register pair
4. Subtract minuend and subtrahend
5. Check borrow, if borrow is one go to step 6 else go to step 8
6. Increment borrow (BL register by 1)

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

7. Negate AX register
8. Store the difference from AX in memory
9. Story the borrow from BL in memory
10. Stop

Program:

Address	Label	Mnemonics
1000		MOV BL, 00H
1003		MOV AX, 2024H
1007		MOV CX, 1010H
100B		SUB AX, CX
100D		JNC Label1
100F		INC BL
1011		NEG AX
1013	Label 1	MOV [1200], AX
1017		MOV [1202], BL
101B		HLT

Tabulation:

Input		Output	
Address	Data	Address	Data

3. Multiplication

Algorithm:

1. Clear DX register
2. Load the AX register pair with multiplicand
3. Load the CX register pair with multiplier
4. Multiply multiplicand and multiplier
5. Store the higher byte of product(from DX register pair) in memory specified
6. Store the lower byte of product(from AX register pair) in memory specified
7. Stop
8.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Program:

Address	Label	Mnemonics
1000		MOV DX, 0000H
1004		MOV AX, 00FFH
1008		MOV CX, 0002H
100C		MUL CX
100E		MOV [1200], AX
1012		MOV [1202], DX
1016		HLT

Tabulation:

Input		Output	
Address	Data	Address	Data

4. Division:

Algorithm

1. Clear DX register
2. Load the AX register pair with dividend
3. Load the CX register pair with divisor
4. Dividend divided by divisor
5. Store the quotient (from AX register pair) in memory specified
6. Store the remainder (from DX register pair) in memory specified
7. Stop

Program:

Address	Label	Mnemonics
1000		MOV DX, 0000H
1004		MOV AX, 0008H
1008		MOV CX, 0004H
100C		DIV CX
100E		MOV [1200], AX
1012		MOV [1202], DX
1016		HLT

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Tabulation:

Input		Output	
Address	Data	Address	Data

Result:

Thus Assembly language program are written to perform arithmetic operation using 8086 microprocessor. They are executed using 8086 processor and the outputs are verified.

Outcome:

At the end of this experiment the students able to write ALP programs for Arithmetic and Logical operations.

Viva – voce

1. Write the size of the data bus of 8086.
2. Write the size of the address bus of 8086.
3. What is meant by physical addressing in 8086?
4. What is meant by an Opcode?
5. What is meant by an Operand?
6. What is meant by a Mnemonics?
7. What are the other possibilities of writing ADD, SUB and MUL instructions in other addressing modes?
8. What is the difference between microprocessor and microcontroller?
9. What is meant by LATCH?
10. What is the difference between primary & secondary storage device?
11. What is the difference between static and dynamic RAM?
12. What is an interrupt?
13. Differentiate between RAM and ROM?
14. Define – Compiler
15. Define – Flag
16. Define – Stack
17. How clock signal is generated in 8086 microprocessor?
18. State the functions of queue status line QS0 and QS1 in 8086 microprocessor.
19. What is the purpose of BIU& EU?
20. List out the two examples of assembler directives.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Applications:

1. barrel shifters
2. Multiple-precision arithmetic
3. Calculation pipeline
4. binary multipliers
5. Calculation in a single clock

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.2 DECIMAL ARITHMETIC AND CODE CONVERSION

Aim:

To write an Assembly Language Program to perform Code conversion and Decimal Arithmetic operations and to execute using 8086 microprocessor

Apparatus required:

1. 8086 Kit
2. Power cable
3. Keyboard

Procedure:

To enter program in Trainer Kit

1. Press 'RESET' key
2. Press 'A(Address)' key
3. Press "Enter" key
4. Enter the starting address (16 bit)
5. Press "Enter" key
6. Type the Assembly Language Program
7. Press 'RESET' key

To executive the program

8. Press "GO"
9. Type the starting address to execute the program
10. Press 'RESET' key

To verify the result

11. Type "SB(Substitute Byte)"
12. Enter the output address location to check the result

Decimal Arithmetic:

1. Using DAA

Algorithm:

1. Get the first input and store in AL.
2. Get the second input and store in BL.
3. Clear the CL reg.
4. Add the contents of AL and BL.
5. Adjust decimal accumulator after addition.
6. Store the result.

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Program:

Address	Label	Mnemonics
1000		MOV AL, [1100]
1004		MOV BL, [1101]
1008		MOV CL, 00
100B		ADD AL,BL
100D		DAA
100E		JNC L1
1010		INC CL
1012	L1	MOV [1150], AL
1016		MOV [1151], CL
101A		HLT

Tabulation:

Input		Output	
Address	Data	Address	Data

2. Using AAM

Algorithm:

1. Get the first input and store in AL.
2. Get the second input and store in BL.
3. Multiply the contents of AL and BL.
4. Adjust decimal accumulator after multiplication.
5. Store the result.

Program:

Address	Label	Mnemonics
1000		MOV AL, [1100]
1004		MOV BL, [1101]
1008		MUL BL
100A		AAM
100C		MOV [1150], AH
1010		MOV [1151], AL
1014		HLT

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Tabulation:

Input		Output	
Address	Data	Address	Data

3. Using DAS

Algorithm:

1. Get the first input and store in AL.
2. Get the second input and store in BL.
3. Subtract the contents of BL from AL.
4. Adjust decimal accumulator after subtraction.
5. Store the result.

Program:

Address	Label	Mnemonics
1000		MOV AL, [1100]
1004		MOV BL, [1101]
1008		SUB AL,BL
100A		DAS
100B		MOV [1150], AL
100F		HLT

Tabulation:

Input		Output	
Address	Data	Address	Data

Code Conversion:

1. Binary to BCD

Algorithm:

1. Get the input and store in AL.
2. Clear the carry flag.
3. Rotate right the bits of input.
4. Find logical XOR between input and shifted input.
5. Store the result.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Program:

Address	Label	Mnemonics
1000		XOR AX,AX
1002		XOR CX,CX
1004		MOV BX,000B
1008	L2	CMP BX,00
100C		JZ L1
100E		DEC BX
100F		MOV AL,CL
1011		ADD AL,01
1014		DAA
1015		MOV CL,AL
1017		MOV AL,CH
1019		ADC AL,00
101C		DAA
101D		MOV CH,AL
101F		JMP L2
1022		MOV [1150],CX
1026		HLT

Tabulation:

Input		Output	
Address	Data	Address	Data

2. Binary to Gray

Algorithm:

1. Get the input and store in AL.
2. Clear the carry flag.
3. Rotate right the bits of input.
4. Find logical XOR between input and shifted input.
5. Store the result.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Program:

Address	Label	Mnemonics
1000		MOV AL, [1100]
1004		MOV BL, AL
1006		CLC
1007		MOV CL,01
100A		RCR AL,CL
100C		XOR BL,AL
100E		MOV [1150],BL
1012		HLT

Tabulation:

Input		Output	
Address	Data	Address	Data

Result:

Thus an Assembly Language Program for code conversion and decimal arithmetic operations is written. The program is executed using 8086 and the output is verified.

Outcome:

At the end of this experiment the students able to understand the concepts of code conversions and decimal arithmetic operations.

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Viva – voce

1. Write the function of the following 8085 instructions: JP, JPE, JPO, and JNZ.
2. What is the purpose of the following commands in 8086?
   a) AAD
   b) RCL
3. List out the addressing modes in 8086.
4. List out the various string instructions that are available in 8086.
5. What are the 8086 instructions used for BCD arithmetic?
6. What flags get affected after executing ADD instruction?
7. Which instruction is used to add immediate data?
8. What is BCD code? Where it is used?
9. What is ASCII code? Where it is used?
10. What is the difference between carry flag and overflow flag?
11. What are the special function register associated with interrupts?
12. List the flags affected by arithmetic instructions.
13. After executing ADDC instruction, what flags get affected?
14. How many bytes the instruction ADDC will add?
15. Name the signals used by the processor to communicate with an I/O processor
16. What is the function of IP?
17. What is the use of base pointer register?
18. Mention the index registers of 8086.
19. How many memory locations are available in 8086 microprocessor?
20. What are the flags in 8086? What are the various interrupts in 8086?

Applications:

1. Programmable calculators
2. Counting system using counts the tennis ball
3. Detecting and Correcting an Error with the Hamming Code
4. Power tool
5. Radio clocks

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.3 MOVE A DATA BLOCK WITHOUT OVERLAP

Aim:

To write an assembly language program to move a block of data to another memory location and execute it using 8086

Apparatus required:

1. 8086 Kit
2. Power cable
3. Keyboard

Algorithm:

1. Initialize the source and destination memory location to SI and DI register
2. Initialize the counter (CL register)
3. Move the data to designation memory through AL register
4. Decrement the count
5. Check for zero flag and if zero flag is set go to next step
6. else go to step 3
7. Stop

Procedure:

To enter program in Trainer Kit

1. Press 'RESET' key
2. Press 'A(Address)' key
3. Press "Enter" key
4. Enter the starting address (16 bit)
5. Press "Enter" key
6. Type the Assembly Language Program
7. Press 'RESET' key

To executive the program

8. Press "GO"
9. Type the starting address to execute the program
10. Press 'RESET' key

To verify the result

11. Type "SB(Substitute Byte)"
12. Enter the output address location to check the result

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Program:

Address	Label	Mnemonics
1000		MOV CL, 08
1003		MOV SI, 1200
1007		MOV DI, 1300
100B	L1	LOD SB
100C		MOV [DI], AL
100E		INC DI
100F		DEC CL
1011		JNZ L1
1013		HLT

Tabulation:

Input		Output	
Address	Data	Address	Data

firstranker.com
www.FirstRanker.com

Result:

Thus an Assembly Language Program for moving data to another location is written the program is executed using 8086 and the output is verified.

Outcome:

At the end of this experiment the students able to understand the concepts and write the ALP program for moving data to another location.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Viva – voce

1. List out the Flag manipulation instruction.
2. Define – Variables
3. Define – Segment Override Prefix
4. How is the memory segment accessed by 8086 microprocessor identified?
5. List out the advantages of using Direct Memory Access (DMA).
6. What is BIOS function call in 8086? (May 2012)
7. List out the difference between procedures and Macros.
8. What is meant by maskable interrupts& non-maskable interrupts?
9. What is the Maximum clock frequency in 8086?
10. Which Stack is used in 8086?
11. Define – Pipeline (Dec 2011)
12. How many address lines are available in 8086? What is the maximum address possible?
13. What is an assembler?(May 2012)
14. What is the purpose of LEA instruction in 8086? (May 2012)
15. Give the function of index and pointers in 8086.
16. What are the different instruction set of 8086?
17. Give the various addressing modes in 8086.
18. Give the differences between JUMP and LOOP instruction.
19. Give the physical address formation of any two addressing mode.
20. Give the use of "ASSUME" in 8086 programming.

Applications:

1. Priority Encoder
2. DVD players
3. Cellular telephones
4. Household appliances
5. Car equipment

FirstRanker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.4 MATRIX OPERATIONS

Aim:

To write an assembly language program to perform matrix addition and to execute it using 8086

Apparatus required:

1. 8086 Kit
2. Power cable
3. Keyboard

Algorithm:

1. Get the size (count) of the matrix
2. Initialize SI pointer for storing first matrix
3. Initialize DI pointer for storing second matrix
4. Move the contents pointed by SI pointer to accumulator
5. Move the contents pointed by DI pointer BL register
6. Add the contents of accumulator and BL register
7. Store result in the address pointed by DI pointer
8. Increment DI and SI pointer by 1 and Decrement count by 1
9. Check for zero. If no zero jump to step 4 else go to step 11
10. Stop the execution

Procedure:

To enter program in Trainer Kit

1. Press 'RESET' key
2. Press 'A(Address)' key
3. Press "Enter" key
4. Enter the starting address (16 bit)
5. Press "Enter" key
6. Type the Assembly Language Program
7. Press 'RESET' key

To executive the program

8. Press "GO"
9. Type the starting address to execute the program
10. Press 'RESET' key

To verify the result

11. Type "SB(Substitute Byte)"
12. Enter the output address location to check the result

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Program:

Address	Label	Mnemonics
1000		MOV CL, 09
1003		MOV SI, 1100
1007		MOV DI, 1200
100B	L1	MOV AL, [SI]
100D		MOV BL, [DI]
100F		ADD AL, BL
1011		MOV [DI], AL
1013		INC SI
1014		INC DI
1015		DEC CL
1017		JNZ L1
1019		HLT

Tabulation:

Input		Output	
Address	Data	Address	Data

Result:

Thus an assembly language program for matrix addition was written. It is executed using 8086 microprocessor and the output is verified.

Outcome:

At the end of this experiment the students able to understand the concepts of moving and adding the group of data from one location to another location.

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Viva – voce

1. Write an ALP for 8086 to multiply two 16 bit unsigned numbers.
2. What is an accumulator?
3. List out the segment register available in 8086.
4. List out any four program control instructions that are available in 8086.
5. What is program counter?
6. Give any four logical instructions in 8086.
7. How many memory locations are available in 8086 microprocessor?
8. What are the general purposes registers in 8086?
9. What are the functional units in 8086?
10. How much memory location allotted for the particular segments registers in 8086?
11. When the 8086 processor is in minimum mode and maximum mode?
12. Define – Segment Override Prefix.
13. Define – Macro and Procedure
14. Define – Assembler and assembler directives
15. Define – Compiler and Linker
16. What is meant by modular programming?
17. Explain the uses of PUSH and POP instruction.
18. Explain the uses of CALL and RET instruction.
19. Identify the addressing modes in the following instructions.

    AND AL, BL

    SUB AL, 24H

    MOV AL, (BP)

    MOV CX, 1245H
20. What are the 8086 instructions used for BCD arithmetic?

firstranker.com
www.FirstRanker.com

Applications:

1. Toys
2. Light switches and dimmers
3. Electrical circuit breakers
4. Smoke alarms
5. Battery packs

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.5 SEARCHING AND SORTING A STRING IN A GIVEN ARRAY

a) Searching a String in a Given Array:

Aim:

To write an assembly language program to search a string in the given array and to execute it using 8086 kit.

Apparatus required:

1. 8086 Kit
2. Power cable
3. Keyboard

Algorithm:

1. Initialize the source and destination memory in SI and DI register
2. Initialize the count (CL) and Load the string values to any register (DL)
3. Move string to AL register and Compare the register (DL and AL)
4. Check for zero flag and if it is set jump to step 7 else go to step 9
5. Store the value of AL register
6. Increment DI register by one, Increment SI register by one and Decrement CL register by one
7. Check for zero flag and if it is set jump to step 12 else go to step 4
8. Stop the execution

Procedure:

To enter program in Trainer Kit

1. Press 'RESET' key
2. Press 'A(Address)' key
3. Press "Enter" key
4. Enter the starting address (16 bit)
5. Type the Assembly Language Program
6. Press 'RESET' key

To executive the program

7. Press "GO"
8. Type the starting address to execute the program
9. Press 'RESET' key

To verify the result

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

10. Type "SB(Substitute Byte)"
11. Enter the output address location to check the result

Program:

Address	Label	Mnemonics
1000		MOV DI, 1300
1004		MOV SI, 1200
1008		MOV CL, 06
100B		MOV DL, 02
100E	L1	MOV AL, [SI]
1010		CMP DL, AL
1012		JNZ L2
1014		MOV [DI], AL
1016		INC DI
1017	L2	INC SI
1018		DEC CL
101A		JNZ L1
101C		HLT

Tabulation:

Input		Output	
Address	Data	Address	Data

Result:

Thus an assembly language program to search a string in the string in the given array is written. The program is executed using 8086 and the output is verified.

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

b) Sorting an Array of Data Using 8086:

Aim:

To write an assembly language program to sort an array of data in ascending and descending order and execute it using 8086 kit

Apparatus required:

1. 8086 Kit
2. Power cable
3. Keyboard

Procedure:

To enter program in Trainer Kit

1. Press 'RESET' key
2. Press 'A(Address)' key
3. Press "Enter" key
4. Enter the starting address (16 bit)
5. Press "Enter" key
6. Type the Assembly Language Program
7. Press 'RESET' key

To executive the program

8. Press "GO"
9. Type the starting address to execute the program
10. Press 'RESET' key

To verify the result

11. Type "SB(Substitute Byte)"
12. Enter the output address location to check the result

1. Ascending Order:

Algorithm:

1. Set SI as pointer for array
2. Set CL as outer loop counter
3. Initialize array pointer
4. Set CH as inner loop counter
5. Increment array pointer
6. Get the first element of array
7. Increment array pointer and Compare the two elements

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

8. Checks carry flag. If carry flag is set go to step 11 else go to step 10
9. Swap the content of the memory location
10. Decrement the inner loop counter
11. Check zero flag. If it is set go to step 13 else go to step 7
12. Decrement outer loop counter
13. Check zero flag. If it is set go to step 15 else go to step 3 and stop

Program:

Address	Label	Mnemonics
1000		MOV SI, 1100H
1004		MOV CL, [SI]
1006		DEC CL
008	Repeat	MOV SI, 1100H
100C		MOV CH, [SI]
100E		DEC CH
1010		INC SI
1011	Rep	MOV AL, [SI]
1013		INC SI
1014		CMP AL, [SI]
1016		JC Ahead
1018		XCHG AL, [SI]
101A		XCHG AL, [SI-1]
101D	Ahead	DEC CH
101F		JNZ Rep
1021		DEC CL
1023		JNZ Repeat
1025		HLT

Tabulation:

Input		Output	
Address	Data	Address	Data

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

2. Descending Order:

Algorithm:

1. Set SI as pointer for array
2. Set CL as outer loop counter
3. Initialize array pointer
4. Set CH as inner loop counter
5. Increment array pointer Get the first element of array
6. Increment array pointer
7. Compare the two elements
8. Checks carry flag. If carry flag is set go to step 11 else go to step 10
9. Swap the content of the memory location
10. Decrement the inner loop counter
11. Check zero flag. If it is set go to step 13 else go to step 7
12. Decrement outer loop counter
13. Check zero flag. If it is set go to step 15 else go to step 3
14. Stop

Program:

Address	Label	Mnemonics
1000		MOV SI, 1100H
1004		MOV CL, [SI]
1006		DEC CL
1008	Repeat	MOV SI, 1100H
100C		MOV CH, [SI]
100E		DEC CH
1010		INC SI
1011	Rep	MOV AL, [SI]
1013		INC SI
1014		CMP AL, [SI]
1016		JNC Ahead
1018		XCHG AL, [SI]
101A		XCHG AL, [SI-1]
101D	Ahead	DEC CH
101F		JNZ Rep
1021		DEC CL
1023		JNZ Repeat
1025		HLT

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Tabulation:

Input		Output	
Address	Data	Address	Data

Result:

Thus an assembly language program is written to sort an array of data in ascending and descending order. The program is executed using 8086 and the output is verified.

Outcome:

At the end of this experiment the students able to understand the concepts and write the Assembly Language Program for Searching and Sorting.

Viva – voce

1. What is the relation between 8086 processor frequency & crystal Frequency?
2. What is the position of the stack pointer after the POP instruction?
3. Compare CALL and JMP instructions.
4. Define – Baud Rate
5. What is the size of instruction queue in 8086?
6. Compare JNC and JMP instructions.
7. What happens when HLT instruction is executed in processor?
8. What is the maximum internal clock frequency of 8086 processor?
9. What are the functions of BIU?
10. Write an ALP program to search a number 05 from a given array.
11. What is cache memory?
12. Can ROM be used as stack?
13. What are the 8086 instructions used for BCD arithmetic?
14. What are the 8086 instructions used for ASCII arithmetic?
15. List the various string instructions available in 8086.
16. How will carry and zero flags reflect the result of instruction CMP BX, CX?
17. Give any four miscellaneous instructions in 16-bit Microprocessor
18. List the flags in 8086 and state its functions.
19. What is the purpose of segment registers in 8086?
20. What is virtual addressing mode?

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Applications:

1. Car keys
2. Power tool
3. Home security system
4. Dishwashers
5. Washing machines

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.6 PASSWORD CHECKING AND PRINT RAM SIZE WITH SYSTEM DATE

Aim:

To write an assembly language program to Check Password.

Apparatus required:

1. Personal Computer
2. MASM Software

Procedure:

1. Go to Start, Programs, MS-DOS Prompt (or Command Window). Once the DOS window is open, type the command EDIT. Type your file and save it to your .ASM directory. Be sure it has a .ASM extension. Do not close the DOS window or the editor.
2. Go to Start, Programs, MS-DOS Prompt (or Command Window). This will open a second DOS window. Change to your .ASM directory and issue the commands to run MASM.
3. Click compile, link and run it.
   a) C: cd \COP3402
   b) MASM/L FIRST.ASM
   c) LINK FIRST.OBJ
   d) FIRST.EXE
4. For errors after issuing the MASM command, fix the .ASM file and redo the above steps.
5. open the file FIRST.LST to clear all errors
6. Click compile, link and run it.
   a) MASM/L FIRST.ASM
   b) LINK FIRST.OBJ
   c) FIRST.EXE

Program:

```
.MODEL SMALL
.DATA
    PASS DB 'ABC'
    MES1  DB  10,13,'ENTER 3 CHARACTER PASSWORD  $'
    MES2  DB  10,13,'PASSWORD IS CORRECT $'
    MES3  DB  10,13,'PASSWORD ID WRONG $ '
.CODE
    START: MOV  AX,@DATA
           MOV  DS,AX
           MOV  AH,09H
           LEA DX,MES1
           INT  21H
```


FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

```
        MOV  CL,00
        MOV  DL,OOH
        XOR DI,DI
        .WHILE  CL!=3
          MOV AH,07H
            INT  21H
            LEA  BX,PASS
            MOV  AH,[BX+DI]
            .IF  AL==AH
                ADD DL,01
            .ENDIF
            INC DI
            INC  CL
        .ENDW
        .IF DL==3
              MOV  AH,09H
              LEA    DX,MES2
              INT  21H
        .ELSE
              MOV  AH,09H
              LEA  DX,MES3
              INT  21H
        .ENDIF
        MOV  AH,4CH
        INT  21H
    END  START
END
```

Result:

Thus the given password is checked and the result is verified.

Outcome:

At the end of this experiment the students able to understand the concepts of password checking and print ram size with system date.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Viva – voce

1. What is the role of Stack?
2. What is the difference between DOS and BIOS interrupts?
3. What is an interrupt vector Tabulation: of 8086?
4. What .model small stands for?
5. Define – Interrupt Vector Tabulation
6. What are the contents of AL and CY after the execution of the following segments?
7. What is the purpose of CLK signal in an 8086 system?
8. What is the need for MN/MX pin in 8086 system?
9. What is the purpose of QUEUE in 8086 processor?
10. Give the operation of TEST instructions of 8086?
11. List out few string instructions of 8086.
12. What is the use of LOCK prefix?
13. What is the purpose of REP prefix?
14. What are the types of Multiprocessor configuration?
15. Define – Co-processor
16. List any four program control instructions available in 8086?
17. How the data and address lines are demultiplexed?
18. Define – Instruction
19. Define – Machine Cycle
20. Define – T-State

Applications:

1. High-end coffee makers
2. Radio clocks
3. Televisions
4. VCRs
5. DVD players

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.7 COUNTERS AND TIME DELAY

Aim:

To write an assembly language program to display the current system time and date

Apparatus required:

1. Personal Computer
2. MASM Software

Procedure:

1. Go to Start, Programs, MS-DOS Prompt (or Command Window). Once the DOS window is open, type the command EDIT. Type your file and save it to your .ASM directory. Be sure it has a .ASM extension. Do not close the DOS window or the editor.
2. Go to Start, Programs, MS-DOS Prompt (or Command Window). This will open a second DOS window. Change to your .ASM directory and issue the commands to run MASM.
3. Click compile, link and run it.
   a) C:
   b) cd \ COP3402
   c) MASM/L FIRST.ASM
   d) LINK FIRST.OBJ
   e) FIRST.EXE
4. For errors after issuing the MASM command, fix the .ASM file and redo the above steps.
5. open the file FIRST.LST to clear all errors
6. Click compile, link and run it.
   a) MASM/L FIRST.ASM
   b) LINK FIRST.OBJ
   c) FIRST.EXE

Program:

```
.MODEL SMALL
.STACK 100H
.DATA
   PROMPT  DB  'Current System Time is : $'
   TIME    DB  '00:00:00$'
.CODE
  MAIN PROC
```


FirstRanker.com
Firstranker's choice
www.FirstRanker.com

```
                MOV AX, @DATA
                MOV DS, AX
              LEA BX, TIME
                  CALL GET_TIME
                LEA DX, PROMPT
                MOV AH, 09H
                INT 21H
                LEA DX, TIME
          MOV AH, 09H
          INT 21H
                MOV AH, 4CH
                INT 21H
        MAIN ENDP

GET_TIME PROC
          PUSH AX
          PUSH CX
          MOV AH, 2CH
          INT 21H
          MOV AL, CH
          CALL CONVERT
          MOV [BX], AX
          MOV AL, CL
          CALL CONVERT
          MOV [BX+3], AX
          MOV AL, DH
          CALL CONVERT
          MOV [BX+6], AX
          POP CX
          POP AX
        RET
GET_TIME ENDP
```

Result:

Thus the systems current date is fetched, displayed and the result is verified.

Outcome:

At the end of this experiment the students able to understand the concepts of counters and time delay.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Viva – voce

1. What are the 8086 instructions used for BCD arithmetic?
2. What is the function of BX register?
3. How Physical address is generated?
4. List out the pointers available in 8086
5. Compare PUSH and PULL instructions
6. What is ALE? When will the data bus AD0-AD7 be enabled?
7. Define – HOLD in 8086
8. Define – HLDA in 8086
9. Give the significance of RQ / GTO and IO / M signals.
10. Name any two coprocessors and their use.
11. State the importance of sample and hold circuit.
12. List the applications of programmable interval timer.
13. What is key denouncing? What are the methods to detect the denouncing?
14. Name the two modes of operation of DMA controller?
15. Give the different types of command words in 8259.
16. Give the comments for MOV r, M.
17. How many T-states are in MOV instruction?
18. Explain the addressing mode of MOV r, M.
19. How many machine cycles are in MOV instruction?
20. Give the comments for MOV M, r.

Applications:

1. Microwaves
2. Toasters
3. Ovens
4. Stoves
5. Thermostats

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.8 INTERFACING STEPPER MOTOR WITH 8086

Aim:

To write an Assembly Language Program to run the stepper motor in forward and reverse direction and execute it using 8086 microprocessor

Apparatus required:

1. Stepper motor Card
2. Interface Cable
3. Datasheets | App notes

Procedure:

1. Connect the 26 core FRC connector to the 8086 trainer at connector no CN4 and the interface module.
2. Connect the power mate connector to the interface module and the other side of the connector to the power supply.
3. 5- Way power mate is wired to the motor. This power mate is to be inserted into the male socket provided on the interface.
4. After the completion of the program and connections, enter the program as given in the listing below. G0< STARTING ADDRESS< ENTER (on the key board of trainer).

Algorithm:

1. Get the drive sequence using HL pointer
2. Initialize a counter for appropriate number of steps
3. Get the drive sequence from memory to accumulator
4. Send the data to stepper motor
5. Call delay
6. Get the next sequence to accumulator
7. Decrement the counter
8. Check zero flag, If zero flag is reset go to step 3 else go to step 9
9. Jump to step 1

Look Up Tabulation:

1200	09	05	06	0A
1210	0A	06	05	09

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Address Decoding:

A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	X	X	X

Half Wave Scheme:

Anticlock wise					Clock wise				
Step	A1	A2	B1	B2	Step	A1	A2	B1	B2
1	1	0	0	1	1	1	0	1	0
2	0	1	0	1	2	0	1	1	0
3	0	1	1	0	3	0	1	0	1
4	1	0	1	0	4	1	0	0	1

Program:

Address	Label	Mnemonics
1000	Start	MOV BL, 20H
1003	Next	MOV DI, 1200
1007		CALL 1025(Rotate)
100A		DEC BL
100C		JNZ 1003(Next)
100E		CALL 1037(Delay 2)
1011		MOV BL, 20H
1014	Next 1	MOV DI, 1210
1018		CALL 1025(Rotate)
101B		DEC BL
1010		JNZ 1014(Next 1)
101F		CALL 1037 (Delay 2)
1022		JMP 1000H (Start)
1025	Rotate	MOV CL, 04H
1028	Start 1	MOV AL, [DI]
102A		OUT 0C0(Port), AL
102C		MOV DX, 0FFFH
1030	Delay 1	DEC DX
1031		JNZ 1030H(Delay 1)
1033		INC DI
1034		LOOP 1028(Start 1)
1036		RET
1037	Delay 2	MOV DX, 0FFFH
103B	Delay	DEC DX
103C		JNZ 103B(Delay)
103E		RET

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Result:

Thus an Assembly Language Program to interface stepper motor with 8086 processor is written. It is executed using 8086 and the output is verified.

Outcome:

At the end of this experiment the students able to understand the working conditions of stepper motor and write the assembly language program for interfacing stepper motor with 8086 microprocessor.

Viva – voce

1. What is meant by Prefix?
2. Difference between small, medium, tiny, huge?
3. Define –DD, DW and DB.
4. List out the Interrupts in 8086
5. What is meant by half wave scheme?
6. Give examples for 8 / 16 / 32 bit Microprocessor?
7. What is 1st / 2nd / 3rd / 4th generation processor?
8. What is meant by interrupt?
9. What is meant by Scratch pad of computer?
10. What is NV – RAM?
11. Which interrupts are generally used for critical events?
12. What is the position of the Stack Pointer after the PUSH instruction?
13. What is the position of the Stack Pointer after the POP instruction?
14. Logic calculations are done in which type of registers?
15. Explain how to generate the physical address with respect to code segment and any other segment.

firstranker.com
www.FirstRanker.com

Applications:

1. Clothes washers
2. Stereo systems
3. Hand-held game devices
4. Thermostats
5. Video game systems

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.9 INTERFACING TRAFFIC LIGHT WITH 8086

Aim:

To Interface the Traffic light controller with 8086

Kit Includes:

1. Traffic Light Controller kit
2. Interface Cable
3. Datasheets | App notes

Procedure:

To interface the Trainer Kit with Traffic light controller

1. Connect Power supply to 8086 kit
2. Interfacing kit Connect to CN4 of 8086 using 26 pin bus.

To enter program in Trainer Kit

3. Press 'RESET' key
4. Press 'A(Address)' key
5. Press "Enter" key
6. Enter the starting address (16 bit)
7. Press "Enter" key
8. Type the Assembly Language Program
9. Press 'RESET' key

To executive the program

10. Press "GO"
11. Type the starting address to execute the program

To observe the output

12. Take the reading from the interfacing kit traffic light controller

Lookup Table:

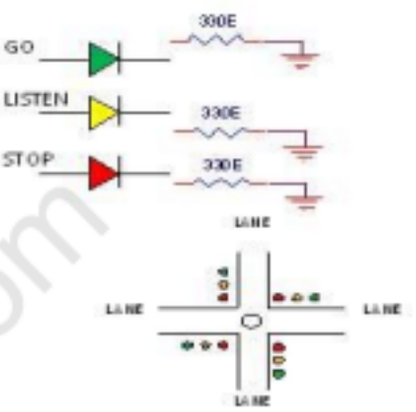
1200	80H
1201	21H,09H,10H,00H(SOUTH WAY)
1205	0CH,09H,80H,00H(EASTWAY)
1209	64H,08H,00H,04H(NOURTHWAY)
120D	24H,03H,02H,00H(WEST WAY)
1211	END

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

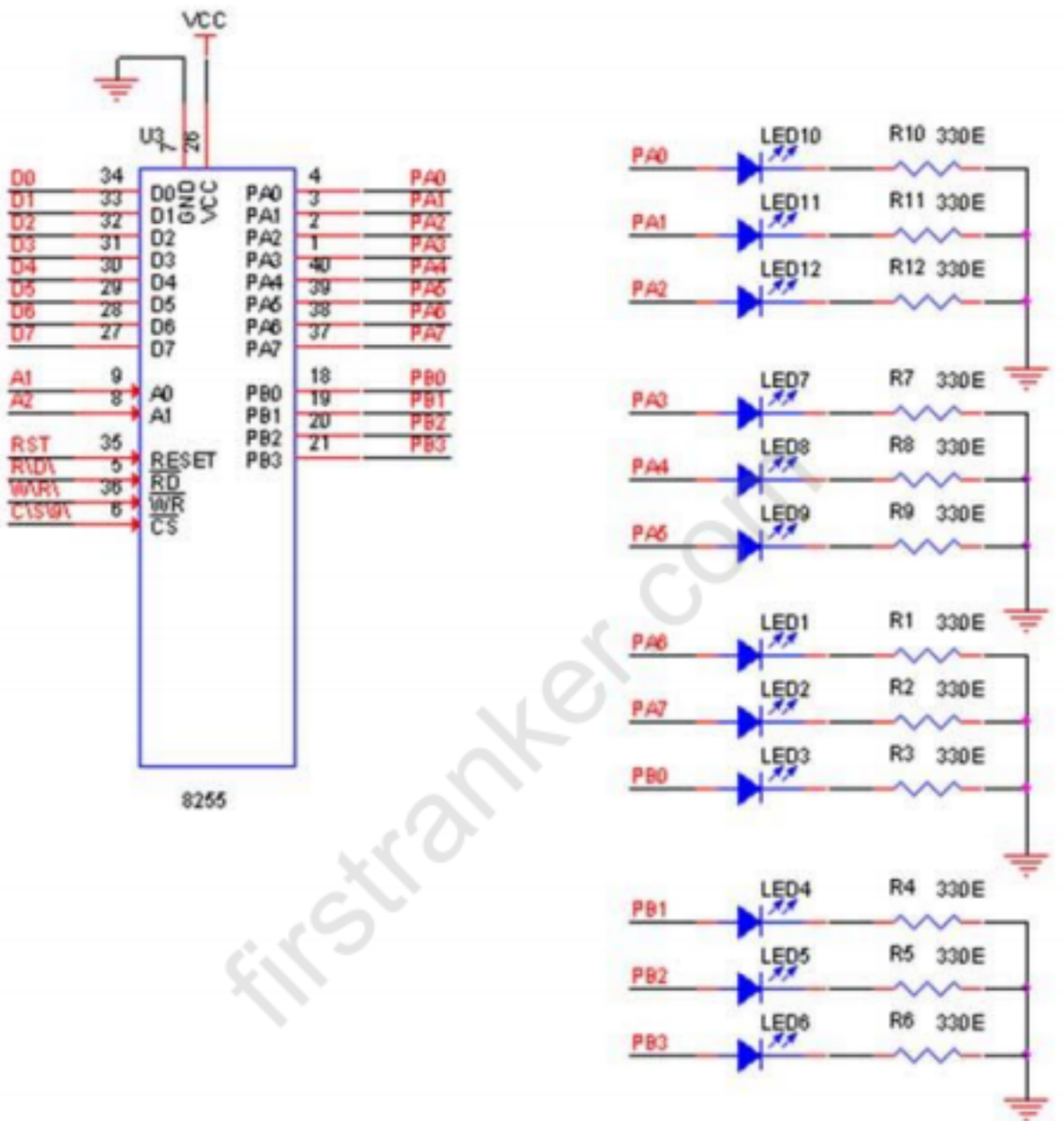
Pin Assignment With 8086

LAN Direction	8086 LINES	MODULES	Traffic Light Controller Card
SOUTH	PA.0	GO	 Make high to - LED On Make low to – LED Off
	PA.1	LISTEN	
	PA.2	STOP	
EAST	PA.3	GO	
	PA.4	LISTEN	
	PA.5	STOP	
NORTH	PA.6	GO	
	PA.7	LISTEN	
	PB.0	STOP	
WEST	PB.1	GO	
	PB.2	LISTEN	
	PB.3	STOP	
	13-16	NC	
PWR	17,19	Vcc	Supply form MCU/MPU/FPGA Kits
	18,20	Gnd	

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

CIRCUIT DIAGRAM TO INTERFACE TRAFFIC LIGHT WITH 8086



IN 8086 WE HAVE TWO 8255 IC'S

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

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

GPIO- II (8255)

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

Program:

Address	Label	Mnemonics
1000	START	MOV BX, 1200H
1004		MOV CX, 0008H
1008		MOV AL,[BX]
100A		MOV DX, CONTRL PORT
100E		OUT DX, AL
100F		INC BX
1010	NEXT	MOV AL,[BX]
1012		MOV DX, PORT A
1016		OUT DX,AL
1017		INC BX
1018		MOV AL,[BX]
101A		MOV DX,PORT B
101E		OUT DX,AL
101F		CALL DELAY
1022		INC BX
1023		JNZ NEXT
1025		JMP START
1028	DELAY	PUSH CX
1029		MOV CX,0005
102D	REPAT	MOV DX,0FFFFH
1031	LOOP2	DEC DX
1032		JNZ LOOP2
1034		LOOP REPAT
1036		POP CX
1037		RET

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Result:

Thus an Assembly Language Program to interface traffic light controller with 8086 processor is written. It is executed using 8086 and the output is verified.

Outcome:

At the end of this experiment the students able to understand the working conditions of traffic light controller and write the assembly language program for interfacing with 8086 microprocessor.

Viva – voce

1. List out the control ports in traffic light controller.
2. What are the functions of conditional instructions?
3. List out the LAN ports in traffic light controller.
4. What are the functions of Loop instructions?
5. List out the Modules in traffic light controller.
6. List out the control ports in traffic light controller
7. What are the functions of conditional instructions?
8. List out the LAN ports in traffic light controller.
9. What are the functions of Loop instructions?
10. List out the Modules in traffic light controller.
11. List out the difference between INT 0 and INT 4.
12. Describe the steps required in the execution of an assembly language program.
13. Explain the use of EXTRN and PUBLIC directives with an example.
14. Explain the memory structure in a general purpose desktop computer Illustrate the use of following assembler directives: DD, DW, EVEN, GROUP, ORG, ASSUME, ENDP, PTR, OFFSET.
15. Discuss how "even" and "odd" memory banks are accessed using control signals.

Applications:

1. Traffic control devices
2. Vehicle maintenance system
3. Railways
4. Airport

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.10 DIGITAL CLOCK AND STOP WATCH

Aim:

To Program the 8086microprrocessor that places a message on the screen every 10 sec, using INT 1A h

Apparatus required:

1. Personal Computer
2. MASM Software

Algorithm:

1. Write the timer delay program.
2. Write the timeout subroutine.
3. Clear the accumulator.
4. Give the message to be displayed.
5. Call the interrupt.
6. Display the result.

Procedure:

1. Go to Start, Programs, MS-DOS Prompt (or Command Window). Once the DOS window is open, type the command EDIT. Type your file and save it to your .ASM directory. Be sure it has a .ASM extension. Do not close the DOS window or the editor.
2. Go to Start, Programs, MS-DOS Prompt (or Command Window). This will open a second DOS window. Change to your .ASM directory and issue the commands to run MASM.
3. Click compile, link and run it.
   a) C:
   b) cd \ COP3402
   c) MASM/L FIRST.ASM
   d) LINK FIRST.OBJ
   e) FIRST.EXE
4. For errors after issuing the MASM command, fix the .ASM file and redo the above steps.
5. open the file FIRST.LST to clear all errors
6. Click compile, link and run it.
   a) MASM/L FIRST.ASM
   b) LINK FIRST.OBJ
   c) FIRST.EXE

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Program:

```
CODE SEGMENT
        TIMEDELAY:
                MOV SP,1000H
                MOV DI,10XD
        TIME OUT:
                MOV AH,00H
                INT 1AH
                MOV BX,DX
        TIMER:
                MOV AH, 00H
                INT 1AH
                SUB DX, BX
                CMP DX, 182XD
                JC TIMER
                MOV AH, 09H
        CS      MOV DX,MSG
                INT 21H
                DEC DI
                JNZ TIMEOUT
                MOV AX,4C00H
                INT 21H
        MSG:
                DB 'TEN MORE SECONDS HAVE PASSED $'
CODE ENDS
```

firstranker.com
www.FirstRanker.com

Result:

Thus 8086 microprocessor is programmed for digital clock and the outputs are verified.

Outcome:

At the end of this experiment the students able to understand the working conditions of digital clock and stop watch in 8086 microprocessor.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Viva – voce

1. What is the difference between near and far procedure?
2. What are the different string instructions of 8086?
3. What is the difference between near and far procedure?
4. What is the difference between macro and sub-routine?
5. What are the functions of SI and DI registers?
6. Discuss the use of following instructions:
    a. CLI
    b. LOOP
    c. CALL
    d. AAM
7. Define – ALE
8. Where is the READY signal used?
9. What is the need for timing diagram?
10. What operation is performed during first T-state of every machine cycle in 8085?
11. What is interrupt acknowledge cycle?
12. What is vectored and non-vectored interrupt?
13. List the software and hardware interrupts of 8085.
14. Define – TRAP
15. How clock signals are generated in 8085 and what is the frequency of the internal clock?

Applications:

1. Gasoline pumps
2. Credit-card processing units
3. Elevators
4. Computer servers
5. Surveillance systems

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.11 TRANSFER OF DATA BETWEEN TWO SERIAL PORTS USING 8251

Aim:

To write an assembly language program to transfer the data between two serial ports using 8251 and to execute it using 8086 microprocessor

Kit Includes:

1. Serial port Card
2. Interface Cable
3. Datasheets | App notes

Procedure:

To interface the Trainer Kit with 8251

1. Connect Power supply to 8086 kit
2. 8251Interfacing kit Connect to CN4 of 8086 using 26 pin bus.

To enter program in Trainer Kit

3. Press 'RESET' key
4. Press 'A(Address)' key
5. Press "Enter" key
6. Enter the starting address (16 bit)
7. Press "Enter" key
8. Type the Assembly Language Program
9. Press 'RESET' key

To executive the program

10. Press "GO"
11. Type the starting address to execute the program

To observe the output

12. Take the reading from the interfacing kit 8251.

Algorithm:

Transmitter end:

1. Start.
2. Write the command word into the control register of 8253 Initialize counter 0, read/write 16 bit character, mode 3-square wave rate generator, binary count to generate the baud rate.
3. Load the appropriate count to make the internal clock frequency of 8251-15 kHz as 1.5 kHz, in the counter 0 of 8253.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

4. Write the appropriate command word for asynchronous mode instruction format and command instruction format of 8251 in the control register of 8251.
5. Read the status word from the status register of 8251.
6. Check whether the 3rd bit (TXEMPTY) is set or reset. If it is set loop around the step 5, else go to step 7.
7. Move the content which is to be transmitted to the data register of 8251.
8. Stop.

Receiver end:

Repeat the steps from 1 to 5 in transmitter algorithm

9. Check whether the 2nd bit (RXRDY) is set or reset. If it is reset loop around the step 5, else go to step 7.
10. Read the content from the data register of 8251.
11. Store the content in a memory.
12. Stop.

Program

Transmitter end: 8086 - I

Address	Label	Mnemonics
1000		MOV AL,36
1003		OUT 0CE,AL
1006		MOV AL,0A
1008		OUT 0C8,AL
100A		MOV AL,00
100B		OUT 0C8,AL
100C		MOV AL,4E
100E		OUT 0C2,AL
100D		MOV AL,37
100E		OUT 0C2,AL
1010	L1	IN AL,0C2
1012		AND AL,04
1013		JNZ L1
1014		MOV AL,41
1017		OUT 0C0,AL
1018		HLT

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Receiver end: 8086 - II

Address	Label	Mnemonics
1000		MOV AL,36
1003		OUT 0CE,AL
1006		MOV AL,0A
1008		OUT 0C8,AL
100A		MOV AL,00
100B		OUT 0C8,AL
100C		MOV AL,4E
100E		OUT 0C2,AL
100D		MOV AL,37
100E		OUT 0C2,AL
1010	L2	IN AL,0C2
1012		AND AL,02
1013		JZ L2
1014		IN AL,0C0
1017		MOV BX,1500
1018		MOV [BX],AL
101C		HLT

Result:

Thus the assembly language program to transfer the data between two serial ports using 8251 is written. It is executed using 8086 and the output is verified.

Outcome:

At the end of this experiment the students able to understand the USART communications and write the assembly language program for serial port communication with 8086 microprocessor.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Viva – voce

1. What is the difference between min mode and max mode of 8086?
2. What is the difference between near and far procedure?
3. What is difference between shifts and rotate instructions?
4. Which are strings related instructions?
5. Which are addressing modes and their examples in 8086?
6. Discuss the use of following instructions:
    a. SCASB
    b. LAHF
    c. ROL
    d. SHR
    e. IDIV
7. List out the internal devices of 8255.
8. Define – USART
9. What is scanning in keyboard and what is scan time?
10. What is programmable peripheral device?
11. What are the tasks involved in keyboard interface?
12. How a keyboard matrix is formed in keyboard interface using 8279?
13. Define – GPIB
14. List out advantages of differential data transfer.
15. What are the modes used in keyboard display interface?

Applications:

1. Digital kiosks
2. Security systems
3. Surveillance systems
4. Even some doors with automatic entry
5. Credit-card processing units

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.12 INTERFACING ADC WITH 8086

Aim:

To write an assembly language program to interface ADC with 8086 microprocessor and to execute it using 8086 kit

Kit Includes:

1. ADC Card
2. Interface Cable
3. Datasheets | App notes

Procedure:

1. Connect the 26 core FRC connector to the 8086 trainer at connector no CN4 and the interface module.
2. Connect the power mate connector to the interface module and the other side of the connector to the power supply.
3. 5- Way power mate is wired to the motor. This power mate is to be inserted into the male socket provided on the interface.
4. After the completion of the program and connections, enter the program as given in the listing below. G0< STARTING ADDRESS< ENTER (on the key board of trainer).

Algorithm:

1. Initialize the ADC's channel 0.
2. Make the ALE of the ADC high by using the control word.
3. Set the SOC pulse to ADC.
4. Call delay.
5. Clear the SOC signal.
6. Continuously check for EOC from ADC.
7. If EOC is received, get the digital data from ADC.
8. Store the result in memory.
9. Stop the execution.

Program:

Address	Label	Mnemonics
1000		MOV AL, 10
1003		OUT C8, A
1005		MOV AL, 18

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

1008		OUT C8, AL
100A		MOV AL, 01
100D		OUT D0, AL
100E		MOV AL, 00
1012		MOV AL, 00
1015		MOV AL, 00
1018		MOV AL, 00
101B		OUT D0, AL
101D	LOOP	IN AL, D
101F		AND AL, 01
1022		CMP AL, 01
1025		JNZ LOOP
1027		IN AL, C
1029		MOV BX, 110
102D		MOV [BX], A
102E		HLT

Result:

Thus the assembly language program is written to interface ADC with 8086 microprocessor. The program is executed using 8086 and the output is verified.

Outcome:

At the end of this experiment the students able to understand the working conditions analog to digital converter and write the assembly language program for interfacing ADC with 8086 microprocessor.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Viva – voce

1. What is the difference between Macro and procedure?
2. Which is by default pointer for CS/ES?
3. Basic difference between 8085 and 8086?
4. Which operations are not available in 8085?
5. What is the difference between instructions RET & IRET?
6. What are the functions performed by 8279?
7. What is PPI?
8. Give the control word format for I/O mode of 8255.
9. Give the BSR mode format of 8255.
10. What are the registers present in 8259?
11. What are the applications of 8253?
12. Define – DMA process
13. Give the status word format of 8257
14. What are the features of 8279?
15. List some of the features of INTEL 8259 (Programmable Interrupt Controller).

Applications:

1. Radio clocks
2. Televisions
3. VCRs
4. DVD players
5. ECG Machine

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.13 INTERFACING 8279 WITH 8086 (KEYBOARD AND DISPLAY INTERFACE)

Aim:

To write an assembly language program to interface 8279 with 8086 microprocessor to display a word and to execute using 8086

Kit Includes:

1. Keyboard And Display Card
2. Interface Cable
3. Datasheets | App notes

Procedure:

1. Connect the 26 core FRC connector to the 8086 trainer at connector no CN4 and the interface module.
2. Connect the power mate connector to the interface module and the other side of the connector to the power supply. The connections to the power supply are given below. Connections: (power supply)
3. 5- Way power mate is wired to the motor. This power mate is to be inserted into the male socket provided on the interface.
4. After the completion of the program and connections, enter the program as given in the listing below. G0< STARTING ADDRESS< ENTER (on the key board of trainer).

Algorithm:

1. Start.
2. Load the SI pointer with the address of the data to be displayed.
3. Load the count register with count of data to be displayed.
4. Initialize the accumulator.
5. Load the appropriate command words in control register.
6. Load the appropriate words for the characters to be displayed, in the accumulator in BCD form.
7. Send the BCD data to 8279 display Register.
8. Increment the SI register for the next data.
9. Decrement the count, go to step 7, if the count is not zero. Go to step 1.
10. Stop execution.

Program:

Address	Label	Mnemonics
1000	Start	MOV SI, 1200
1004		MOV CX,000F

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

1008		MOV AL, 10
100B		OUT C2, AL
100D		MOV AL,0CC
1010		OUT C2, AL
1012		MOV AL,90
1015		OUT C2, AL
1017	L1	MOV AL,[SI]
1019		OUT C0, AL
101B		CALL 1500 (D1)
101E		INC SI
101F		LOOP 1017 (L1)
1021		JMP 1000
DELAY 1500	D1	MOV DX,0A0FF
1504	L2	DEC DX
1505		JNZ 1504 L2
1507		RET

Result:

Thus an assembly language program was written to interface 8279 with 8086 microprocessor. It was executed and the output was verified.

Outcome:

At the end of this experiment the students able to understand the working conditions of key board display and write the assembly language program for key board display with 8086 microprocessor.

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Viva – voce

1. What is the size of flag register?
2. Can you perform 32 bit operation with 8086? How?
3. What is the difference between instructions DIV & IDIV?
4. What is the size of each segment?
5. What is the difference between instructions MUL & IMUL?
6. What is meant by LED/LCD?
7. How do you place a specific value in DPTR register? (Dec 2013)
8. Which of the 8051 ports need pull-up registers to functions as I/O port ? (Dec 2013)
9. What are the control words of 8251A and what are its functions?
10. What are the display modes supported by the 8279 chip?
11. Give the format of program clock word of 8279 and mention its purpose.
12. What is 2 key lockout and n key rollover?
13. Define – PPI
14. What is the use of direction flag?
15. What are the alternate functions of port0, port1, port2 and port3?

Applications:

1. Oscilloscopes
2. Multi-meter
3. Leakage Current Tester
4. Data Acquisition and Control
5. ECG Machine

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.14 INTERFACING DAC WITH 8086

Aim:

To write an assembly language program to interface DAC with 8086 microprocessor and generate the following waveforms:

1. Square waveform
2. Saw tooth waveform
3. Triangular waveform
4. Sine waveform

Kit Includes:

1. DAC Card and Interface Cable
2. CRO
3. Datasheets | App notes

Procedure:

To interface the Trainer Kit with DAC converter

1. Connect Power supply to 8086 kit
2. 8255Interfacing kit Connect to CN4 of 8086 using 26 pin buses.
3. Connect the CRO probe to JP3 of 8255 kit
4. Keep the DIP switch in 1 & 7 on (8086kit), Change dip switch into 1 & 5on, once reset 8086 kit

To enter program in Trainer Kit

5. Press 'RESET' key
6. Press 'A(Address)' key
7. Press "Enter" key
8. Enter the starting address (16 bit)
9. Press "Enter" key
10. Type the Assembly Language Program
11. Press 'RESET' key

To executive the program

12. Press "GO"
13. Type the starting address to execute the program

To observe the output

14. Take the reading amplitude from the CRO
15. Take the reading time variations from the CRO

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Note:

1. Square waveform at the output of DAC-2
2. Saw tooth waveform at the output of DAC-1
3. Triangular waveform at the output of DAC-2
4. Sine waveform at the output of DAC-1

1. Triangular Wave Generation:

Algorithm:

1. Initialize the accumulator.
2. Send the data from the accumulator to the DAC port.
3. Go to delay subroutine.
4. Load the data FFh to the accumulator.
5. Send the data to the DAC port.
6. Go to delay subroutine.
7. Go to step 1.

Delay Subroutine

1. Load the count in CX register.
2. Loop around step 2 until CX becomes zero.
3. Return to the main program.

Program:

Address	Label	Mnemonics
1000	START:	MOV AL,00h
1003		OUT C8,AL
1005		CALL Delay
1007		MOV AL, 0FFh
100B		OUT C8,AL
100D		CALL Delay
1010		JMP START
1013	Delay	MOV CX,05FFh
1017		LOOP Delay
1019		RET

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

2. Saw tooth Waveform Generation:

Algorithm:

1. Initialize accumulator.
2. Send the data to DAC port.
3. Increment accumulator.
4. Check for zero flag. If it is set go to step 6, else go to step2.
5. Jump to step1.

Program:

Address	Label	Mnemonics
1000	START:	MOV AL,00h
1003	L1:	OUT C0,AL
1005		INC AL
1007		JNZ L1
1009		JMP START

3. Square Wave Generation:

Algorithm:

1. Initialize the accumulator.
2. Send the data from the accumulator to the DAC port.
3. Go to delay subroutine.
4. Load the data FFh to the accumulator.
5. Send the data to the DAC port.
6. Go to delay subroutine.
7. Go to step 1.

Delay Subroutine

1. Load the count in CX register.
2. Loop around step 2 until CX becomes zero.
3. Return to the main program.

Program:

Address	Label	Mnemonics
1000	START:	MOV AL,00h
1003		OUT C8,AL

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

1005		CALL Delay
1007		MOV AL,FFh
100B		OUT C8,AL
100D		CALL Delay
1010		JMP START
1013	Delay	MOV CX,05FFh
1017		LOOP Delay
1019		RET

Result:

Thus the assembly language program is written to interface timer with 8086 Microprocessor. The program is executed using 8086 and the output is verified.

Outcome:

At the end of this experiment the students able to understand the working conditions of digital to analog converter and write the assembly language program for DAC with 8086 microprocessor.

Viva – voce

1. Whether 8086 is compatible with Pentium processor?
2. Write an ALP program for multiplication of given number in location mode a) 0060, b) 0002
3. What is 8087? How it is different from 8086?
4. Write an ALP program for addition of multi byte numbers.
5. What is the size of flag register?
6. List the operating modes of 8253 timer.
7. Give the control word format of timer.
8. What is the use of USART?
9. Compare the serial and parallel communications.
10. What is the use of Keyboard and display controller?
11. What is meant by synchronous data transfer scheme?
12. Define – Interrupt I/O
13. Why interfacing is needed for I/O devices?
14. When the 8085 processor checks for an interrupt?
15. How the 8085 processor differentiates a memory access and I/O access?

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Applications:

1. Oscilloscopes
2. Multi-meter
3. Leakage Current Tester
4. Data Acquisition and Control
5. ECG Machine

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.15 BASIC ARITHMETIC AND LOGICAL OPERATIONS USING 8051MICROCONTROLLER

Aim:

To do the arithmetic and logical operations using 8051 microprocessor

Apparatus required:

1. 8051 Microcontroller Kit
2. Power cable
3. Keyboard

Procedure:

To enter program in Trainer Kit

1. Press 'RESET' key
2. Press 'A(Address)' key
3. Press "Enter" key
4. Enter the starting address (16 bit)
5. Press "Enter" key
6. Type the Assembly Language Program
7. Press 'RESET' key

To executive the program

8. Press "GO"
9. Type the starting address to execute the program
10. Press 'RESET' key

To verify the result

11. Type "SD(Substitute address)"
12. Enter the output address location to check the result

firstranker.com
www.FirstRanker.com

1. 8-bit Addition:

Algorithm:

1. Move 1st data to memory
2. Add or subtract 1stdata with 2nddata
3. Initialize data pointer.
4. Move result to memory pointed by DPTR.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Program:

Memory Location	Label	Mnemonics
4100	Start	CLR C
4101		MOV A, # data 1
4103		ADD A, # data 2
4105		MOV DPTR, # 4500
4108		MOVX @DPTR, A
4109		SJMP 4109

Tabulation:

Input		Output	
Address	Data	Address	Data

2. 8-bit Subtraction:

Algorithm:

1. Move 1st data to memory
2. Add or subtract 1stdata with 2nddata
3. Initialize data pointer.
4. Move result to memory pointed by DPTR.

Program:

Memory	Label	Mnemonics
4100	Start	CLR C
4101		MOV A, # data 1
4103		SUBB A, # data 2
4105		MOV DPTR, # 4500
4108		MOVX @DPTR, A
4109		SJMP 4109

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Tabulation:

Input		Output	
Address	Data	Address	Data

3. 8-bit Multiplication:

Algorithm

5. Get 1st data and 2nd data to memory
6. Multiply or divide 1st data with 2nd data
7. Initialize data pointer.
8. Move result to memory pointed by DPTR (first port)
9. Increment DPTR
10. Move 2nd part of result to register A
11. Move result to 2nd memory location pointer by DPTR

Program:

Memory	Label	Mnemonics
4100	Start	MOV A, # data 1
4101		MOV B, # data2
4105		MUL AB
4106		MOV DPTR, # 4500
4109		MOVX @DPTR, A
410B		MOV A,B
410D		MOVA @DPTR
410E		SJMP 410E

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Tabulation:

Input		Output	
Address	Data	Address	Data

4. 8-bit Division:

Program:

Memory	Label	Mnemonics
4100	Start	MOV A, # data 1
4102		MOV B,#data2
4105		DIV AB
4106		MOV DPTR, # 4500
4109		MOVX @DPTR, A
410A		INC DPTR
410B		MOVA,B
410D		MOV@ DPTR, A
410E		SJMP410E

Tabulation:

Input		Output	
Address	Data	Address	Data

Result:

Thus Assembly language program is written and executed to perform arithmetic operations using 8051 microcontroller. The output is verified.

Outcome:

At the end of this experiment the students able to write ALP programs for Arithmetic and Logical operations using 8051 microcontroller.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Viva – voce

1. List out the features of 8051 micro controller
2. What is the width of data bus?
3. What is the width of address bus?
4. What flags get affected after executing ADD instruction?
5. Which instruction is used to add immediate data?
6. What is the function of the instruction INC DPTR?
7. Name the flag register in 8051 microcontroller.
8. What is the on-chip memory size of 8051 microcontroller?
9. List the flags affected by arithmetic instructions.
10. What is the function of the instruction ADDC A, #00H?
11. Specify the size of memory systems used in 8051 microcontroller.
12. Mention the different types of operands used in 8051.
13. How the processor 8051 does know whether on-chip ROM or external program memory is used?
14. What is the difference between AJMP and LJMP instruction?
15. What is the necessary to have external pull-up for port 0 in 8051?
16. List the addressing modes of 8051.
17. Explain the instructions used to access external RAM.
18. List the features of 8051 microcontroller.
19. Explain the interrupts of 8051 microcontroller.
20. 20. What is the function of program counter in 8051?

Applications:

1. Radio clocks
2. Televisions
3. Oscilloscopes
4. VCRs
5. DVD players

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.16 FIND 2'S COMPLEMENT OF A NUMBER

Aim:

To write Assembly Language Program to find 2's complement of a number and to execute using 8051 microcontroller

Apparatus required:

1. 8051 Microcontroller Kit
2. Power cable
3. Keyboard

Procedure:

To enter program in Trainer Kit

1. Press 'RESET' key
2. Press 'A(Address)' key
3. Enter the starting address (16 bit)
4. Type the Assembly Language Program

To executive the program

5. Press "GO"
6. Type the starting address to execute the program
7. Press 'RESET' key

To verify the result

8. Type "SD(Substitute address)"
9. Enter the output address location to check the result

Algorithm:

1. Get the input data.
2. Find the complement of the given number.
3. Add one with the complement number.
4. Store the result.

Program:

Address	Label	Mnemonics
4100		MOV DPTR, #4100H
4103		MOVX A, @DPTR
4104		CPL A
4105		ADD A, #01H
4107		INC DPTR
4108		MOVX @DPTR, A
4109		SJMP 4019

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Tabulation:

Input		Output	
Address	Data	Address	Data

Result:

Thus Assembly language program is written and executed to find 2's complement of a number using 8051 microcontroller. The output is verified.

Outcome:

At the end of this experiment the students able to write ALP programs to find 2's complement of a number using 8051 microcontroller.

Viva – voce

1. What is the function of the instruction INC DPTR?
2. What is BCD code? Where it is used?
3. What is ASCII code? Where it is used?
4. What are the alternate functions of port0, port1, port2 and port3?
5. Write about the jump statement
6. Write about CALL statement in 8051
7. Explain the operating mode0 of 8051 serial ports
8. What happens in power down mode of 8051 microcontroller?
9. What are the different ways of operand addressing in 8051?
10. What is the difference between AJMP and LJMP instruction?
11. Define – SFR
12. Define – PSW
13. State the uses of I^2C bus standard.
14. What are the uses of PWM in motor control using Microcontroller?
15. Why are relays that use coils called electromagnetic relays

Applications:

1. Obstacle avoidance robotic vehicle
2. Patient health monitoring system with location details
3. Electronic voting machine
4. Oscilloscopes
5. Multi-meter

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.17 Unpacked BCD to ASCII

Aim:

To write Assembly Language Program to convert Unpacked BCD to ASCII and to execute using 8051 microcontroller

Apparatus required:

1. 8051 Microcontroller Kit
2. Power cable
3. Keyboard

Procedure:

To enter program in Trainer Kit

1. Press 'RESET' key
2. Press 'A(Address)' key
3. Press "Enter" key
4. Enter the starting address (16 bit)
5. Press "Enter" key
6. Type the Assembly Language Program
7. Press 'RESET' key

To executive the program

8. Press "GO"
9. Type the starting address to execute the program
10. Press 'RESET' key

To verify the result

11. Type "SD(Substitute address)"
12. Enter the output address location to check the result

Algorithm:

1. Get the input data.
2. Mask the lower nibble.
3. Add 32 with previous result.
4. Mask the higher nibble of input.
5. Move the number in higher position to lower position by rotating.
6. Add 30 with previous result and store.

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Program:

Address	Label	Mnemonics
4100		MOV A,#29
4101		MOV R2,A
4105		ANL A,#0FH
4106		ORL A,#30H
4109		MOV R6,A
410B		MOV A,R2
410D		ANL A,#0F0H
410E		RRA
410F		RRA
4100		RRA
4101		RRA
4102		ORL A,#30H
4104		MOV R2,A

Tabulation:

Input		Output	
Address	Data	Address	Data

Result:

Thus Assembly language program is written and executed to convert packed BCD to ASCII using 8051 microcontroller. The output is verified.

Outcome:

At the end of this experiment the students able to write ALP programs to convert packed BCD to ASCII using 8051 microcontroller.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Viva – voce

1. What are the special function register associated with interrupts?
2. Name the flag register in 8051 microcontroller.
3. What is the on-chip memory size of 8051 microcontroller?
4. List the flags affected by arithmetic instructions.
5. What is the function of the instruction ADDC A, #00H? After executing ADDC instruction, what flags get affected?
6. Specify the single instruction, which clears the most significant bit of B register of 8051, without affecting remaining bits. (May 2015)
7. Give the DJNZ instruction of Intel 8051 microcontroller(May 2015)
8. Give the schematic to interface a relay with microcontroller. (Dec 2014)
9. State the importance of relay coils. (May 2013)
10. What is PWM?
11. What is resolution?
12. Write about the design steps involved in using microcontroller for stepper motor. (May 2014)
13. Differentiate microprocessor from microcontroller in system design.(Dec 2010)
14. How is the microcontroller used for the stepper motor control application?
15. Why are relays that use coils called electromagnetic relays?

Applications:

1. Multi-meter
2. Leakage Current Tester
3. Data Acquisition and Control
4. ECG Machine
5. Accu-Check

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.18 Sorting of an array using 8051

Aim:

To write Assembly Language Program to sort an array in ascending order and execute it using 8051 microcontroller

Apparatus required:

1. 8051 Microcontroller Kit
2. Power cable
3. Keyboard

Procedure:

To enter program in Trainer Kit

1. Press 'RESET' key
2. Press 'A(Address)' key
3. Press "Enter" key
4. Enter the starting address (16 bit)
5. Type the Assembly Language Program

To executive the program

6. Press "GO"
7. Type the starting address to execute the program
8. Press 'RESET' key

To verify the result

9. Type "SD(Substitute address)"
10. Enter the output address location to check the result

Algorithm:

1. Get the count of the outer loop. Initiate data pointer with the memory address of the array.
2. Get the loop of inner loop and save lower byte of memory address.
3. Get the first number. Increment the data pointer and Get the next number and increment pointer
4. Check for equality. If equal, go to step 6 else go to step 5, Jump unconditionally to step 10
5. If carry flag is reset, jump to step 4, else swap the contents of the two memory locations.
6. Decrement the inner loop count. If the inner loop count is not equal to zero, goto step 4 else goto step 8

Program:

Address	Label	Mnemonics
4100		MOV R0, #05H
4101	AGAIN	MOV DPTR, #4600H

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

4105		MOV R1,#05H
4106		MOV R2, 02H
4109		MOVX A, @DPTR
410B		MOV FO, A
410D		INC DPTR
410E		MOVX A, @DPTR
410F		CJNE A,0F0,L1
4100		AJMP SKIP
4101	L1	JC SKIP
4102		MOV 82H,R2
4104		INC DPTR
4109		MOV A,0F0
410A		MOVX @DPTR,A
410E	SKIP	DJNZ R1, BACK
4110		DJNZ R0, AGAIN
4113		LCALL 00BB

Output:

Input		Output	
Address	Data	Address	Data

firstranker.com
www.FirstRanker.com

Result:

Thus Assembly language program sorting of an array is written and executed using 8051 and the output is verified.

Outcome:

At the end of this experiment the students able to write ALP programs for sorting of a given array using 8051 microcontroller.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Viva – voce

1. What is the function of 01h of Int 21h?
2. How connect the I^2C bus with microcontroller?
3. What is the function of 02h of Int 21h?
4. What do you mean by I^2C standard?
5. State the significance of using microprocessors in interfacing traffic limit control.
6. What is the function of 09h of Int 21h?
7. Whether micro reduces memory requirements?
8. What TD is?
9. What do you mean by emulator?
10. Give stack related instruction.
11. What is the function of 0Ah of Int 21h?
12. How does U differentiate between positive and negative numbers?
13. What is the IC numbers of ADC and DAC
14. What is the function of 4ch of Int 21h?
15. How many no. of ports available for 8051?

Applications:

1. Cell Phones
2. Telephone Sets
3. Answering Machines
4. Fax
5. Printers

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.19 Separate Odd and Even Numbers using 8086

Aim:

To separate odd and even numbers using 8086 microprocessor

Apparatus required:

1. 8086 Microcontroller Kit
2. Power cable
3. Keyboard

Procedure:

To enter program in Trainer Kit

1. Press 'RESET' key
2. Press 'A(Address)' key
3. Enter the starting address (16 bit)
4. Press "Enter" key
5. Type the Assembly Language Program
6. Press 'RESET' key

To executive the program

7. Press "GO"
8. Type the starting address to execute the program
9. Press 'RESET' key

To verify the result

10. Type "SB(Substitute Byte)"
11. Enter the output address location to check the result

Algorithm:

1. Load the CL register with count value.
2. Load the source and destination index registers with memory address.
3. Move the content of source index to accumulator.
4. Rotate right the bits of accumulator with carry.
5. If carry bit is one, then the input is odd number.
6. If carry bit is zero, then the input is even number.
7. Separate the odd and store the results.
8. Separate the even numbers and store the results.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Program:

Address	Label	Mnemonics
4100		MOV CL, 06
4101		MOV SI, 1600
4105		MOV DI, 1500
4106	Loop	LODSB
4109		ROR AL, 01
410B		JB Loop
410D		ROL AL,01
410E		MOV [DI], AL
410F		INC DI
4100		DEC CL
4101		JNZ Loop
4105		INT 3

Result:

Thus program for separating Odd & Even numbers was executed.

Outcome:

At the end of this experiment the students able to write ALP programs for separating Odd & Even numbers using 8086 microprocessor.

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Viva – voce

1. How many bytes the instruction ADDC will add?
2. List of the different modes of timer/counter in 8086?
3. What is meant by loader?
4. What is meant by compiler?
5. Which is the highest priority interrupt for 8086
6. What is the vector address for serial communication interrupt?
7. What is the difference between software and hardware interrupt?
8. What are an ISR and IVT?
9. Which instructions are used for reading & writing data from/to ports of 8051?
10. Which is the highest priority interrupt for 8051?
11. How many timers/ counters are available in 8051?
12. Give comparison of 8086, 286, 386, 486 and Pentium processors with respect to clock speed, data bus width, memory addressing capacity.
13. Explain the salient features of Pentium processor.
14. Differentiate between intra segment and inter segment operations with respect to branch instructions.
15. Differentiate between intra segments and inter segment operations with respect to Call instructions.

Applications:

1. Pedometer
2. Auto-breaking system
3. Mp3 Player
4. Multiple-precision arithmetic
5. Calculation pipeline

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Expt.No.20 Sum of N consecutive numbers using 8086

Aim:

To find the sum of N consecutive numbers with 8086 microprocessor

Apparatus required:

1. 8086 Microcontroller Kit
2. Keyboard
3. power cable

Procedure:

To enter program in Trainer Kit

1. Press 'RESET' key
2. Press 'A(Address)' key
3. Press "Enter" key
4. Enter the starting address (16 bit)
5. Press "Enter" key
6. Type the Assembly Language Program
7. Press 'RESET' key

To executive the program

8. Press "GO"
9. Type the starting address to execute the program
10. Press 'RESET' key

To verify the result

11. Type "SB(Substitute Byte)"
12. Enter the output address location to check the result

Algorithm:

1. Load the source and destination index registers with memory address.
2. Load the CL register with count value.
3. Clear the accumulator.
4. Initialize the BL register with one.
5. Add the content of AL and BL.
6. Increment the BL value.
7. Decrement the CL value.
8. If CL value is not equal to zero, go to step 5.
9. Store the result.

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Program:

Address	Label	Mnemonics
4100		MOV SI, 2000
4101		MOV CL, [SI]
4105		MOV AL, 00
4106		MOV BL, 01
4109	LOOP	ADD AL, BL
410B		INC BL
410D		DEC CL
410E		JNZ LOOP
410F		MOV DI, 2002
4100		MOV [DI], AX
4101		INT 3

Result:

Thus program for finding sum of 'n' consequent numbers was executed

Outcome:

At the end of this experiment the students able to write ALP programs for finding sum of 'n' consequent numbers using 8086 microprocessor.

firstranker.com
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Viva – voce

1. Write an alp program to perform an operation to find the squares of a given number using MP trainer kit
2. What is the disadvantage of microprocessor?
3. What is meant by LATCH?
4. Differentiate between RAM and ROM?
5. What is the difference between primary & secondary storage device?
6. Discuss the use of following instructions:
    a. REP
    b. LOCK
    c. ESC
    d. CLD
7. What are the uses of DOS and BIOS functions?
8. List out the functions of INT 21H.
9. List out the functions of BIOS interrupt INT 10H.
10. Why 2-passes are required for an assembler?
11. Mention the pins available "exclusively" in minimum mode. 49. Mention the pins available "exclusively" in maximum mode.
12. What are the functions of following pins of 8086?
    a. TEST
    b. ALE.
13. What are the functions of following pins of 8086?
    a. READY
    b. BHE
    c. S5, S6, S7.
14. Explain the instruction format of 8086 for data transfer instructions.
15. Is it possible to save the flag register? If yes, how?

Applications:

1. Security systems
2. Surveillance systems
3. Programmable calculators
4. Counting system using counts the tennis ball
5. *Detecting and Correcting an Error with the Hamming Code*

FirstRanker.com
Firstranker's choice
www.FirstRanker.com