

WEST GODAVARI INSTITUTE OF SCIENCE AND ENGINEERING (Approved by AICTE & Affiliated to JNTU Kakinada.) Prakasaraopalem, Avapadu, Tadepalligudem, W.G.Dist, A.P, INDIA-534112

MICROPROCESSOR & MICROCONTROLLERS LABORATORY MANUAL (R20)III- B.Tech., II-Semester ECE



<u>Prepared By</u> T.Hima Bindu Assistant Professor Department of ECE <u>Approved By</u> L. sankar Associate Professor HOD,ECE

Name of the student:_____

H.T.No:_____

DEPARTMENTOFELECTRONICSANDCOMMUNICATIONENGINEERING WISE COLLEGEOFENGINEERING,TADEPALLIGUDEM-534112 (Approved by AICTE, New Delhi &Affiliated to JNTUK: Kakinada) West Godavari District, Andhra Pradesh



WEST GODAVARI INSTITUTE OF SCIENCE AND ENGINEERING (Approved by AICTE & Affiliated to JNTU Kakinada.) Prakasaraopalem, Avapadu, Tadepalligudem, W.G.Dist, A.P, INDIA-534112

> Microprocessor and Microcontroller lab (R20)III-B.Tech., ECE II-Semester List of Experiments as per University

PART-A : (Minimum of 5 Experiments has to be performed)

8086 Assembly Language Programming and Interfacing

1. Programs for16-bit arithmetic operations(using Various Addressing Modes).

- (a) Addition of n-BCD numbers.
- (b) Multiplication and Division operations.
- 2. Program for sorting an array.
- 3. Program for Factorial of given n-numbers.
- 4. Interfacing ADC to 8086
- 5. Interfacing DAC to 8086.
- 6. Interfacing stepper motor to 8086.

PART-B: (Minimum of 5 Experiments has to be performed)

8051 Assembly Language Programming and Interfacing

- 1. Finding number of 1's and number of 0's in a given 8-bit number
- 2. Average of n-numbers.
- 3. Program and verify Timer/Counter in 8051.
- 4. Interfacing Traffic Light Controller to 8051.
- **5.** UART operation in 8051
- 6. Interfacing LCD to 8051.

PART-C: (Minimum of 2 Experiments has to be performed)

Conduct the following experiments using ARM CORTEX M3PROCESSOR USING KEIL MDK ARM

- 1. Write an assembly program to multiply of 2 16-bit binary numbers.
- 2. Write an assembly program to find the sum of first10integers numbers.
- 3. Write a program to toggle LED every second using timer interrupt.



WEST GODAVARI INSTITUTE OF SCIENCE AND ENGINEERING (Approved by AICTE & Affiliated to JNTU Kakinada.) Prakasaraopalem, Avapadu, Tadepalligudem, W.G.Dist, A.P, INDIA-534112

Microprocessor & Microcontroller Lab (R20)III- B.Tech.,ECE II-Semester

	Name of the Experiment	Date	Marks	Signature
PART-A: 8086 Assembly Language Programming and Interfacing	 Programs for 16 -bit arithmetic operations (using Various Addressing Modes). a. Addition of n-BCD numbers. b. Multiplication and Division operations. 2. Program for sorting an array. 3.Program for Factorial of given n-numbers 4.Interfacing ADC to 8086 			
	5.Interfacing DAC to 80866.Interfacing stepper motor to8086			
PART-B: 8051 Assembly Language	 1.Finding number of 1's and number of 0's in a given 8-bit numbers 2. Average of n-numbers. 3.Program and verify Timer/Counter 			
Programming and Interfacing	in80514. Interfacing Traffic Light Controllerto8051.			
	5.UARToperation in 80516. Interfacing LCD to 8051.			
PART-C: ARM CORTEX M3 PROCESSOR USING KEIL MDKARM	 Write an assembly program to multiply of 2 16-bit binary numbers. Write an assembly program to find the sum of first 10 integers numbers. Write a program to toggle LED avery accord using times intermut. 			
PART-D: ADD ON	1.Transfer a block of data			
EXPERIMENTS	2. I acked DCD to ullpacked DCD			

Index



Institution

Vision:

• Promote academic excellence, research, innovation, and entrepreneurial skills to produce graduates with human values and leadership qualities to serve the nation.

Mission:

• Provide student-centric education and training on cutting-edge technologies to make the students globally competitive and socially responsible citizens. Create an environment to strengthen the research, innovation and entrepreneurship to solve societal problems.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

GENERAL INSTRUCTIONS:

- 1. The experiments have been designed to be performed with in the laboratory time.
- 2. To successfully complete the experiment in one lab turn, come prepared to the laboratory.
- 3. Read the experiment in advance.
- 4. Listand collect the components for the experiment.
- 5. Be sure that the specifications and values of the components are as per design.
- 6. Follow the experimental steps judiciously.
- 7. Record stepwise observations using proper test instruments.
- 8. Get the observation signed by the instructor.
- 9. Always take safety precautions while performing experiments.

GUIDANCE FOR THE LABORATORY REPORT:

1. Format of the report

Expt. No:

Date: Expt. Title: Objective: List of instruments and components: Theory in brief Procedure, Observations, Graph if any Result

- 2. Write the experimental observations and measurements stepwise.
- Plot the graph neatly. Always label the axes and indicate units too. Wherever frequency response is to be drawn, use the semi-log graph paper.
- 4. Compare the results with theoretical values with remarks/comments.
- 5. Wherever necessary, sketch the circuit diagram neatly and label the components.

PART-A:

INTRODUCITION TO TASM

ASSEMBLY LANGUAGE PROGRAMMING USING 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 softwares developed by different companies for assembly language programming are:

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

MERIT OF TASM:

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

HOW TO ENTER INTO MASM EDITOR:

- \rightarrow Click "**Start**" on the desktop.
- →Then select **Run**
- \rightarrow Then it Shows inbox

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

 \rightarrow 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 **CDTASM**

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

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

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:TASM>TASM file name.AS Mor C:TASM filename.ASM,, ; Then link this file using C:TASM>LINKfilename.OBJ Or C:TASM>LINKfilename.OBJ,,; i.e link the program in assembly with DOS then to debug to create exe file C:TASM>debugfilename.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 youneed 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] at specific Location	; Enter new or modifies memory contents beginning
FILL F range list	; Fill in arrange 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[L page][P page][handle]

DISPLAY expanded memory status XS

PART-A Programs for 16-bit arithmetic operations (using Various Addressing Modes).

Experiment Number: Date:

(a) Addition of N-BCD numbers

Aim: To write an ALP to perform addition of 'n' BCD numbers

APPARATUS:

Personal Computer TASM software

ALGORITHM:

Step1: Start

Step2: Initialize data segment

Step3: Move 00H into AHregister

Step4: Load N1 into AL register

Step5: Add AL with N2 and the result stored in AL

Step6: Add AL with N3 and the result is stored inAL

Step7: Decimal Adjust Accumulator after addition

Step8: Store the result from AL register to memory location.

Step9: Stop

PROGRAM ASSUME	и: CS: С	ODE, DS:D	ATA	
DATA SE	EGME	NT		
BCD	DB	05H,	24H, 29H	I, 50H, 70H, 44H
RESULT	DW	?		
DATA EN	VDS			
CODE SE	GMEN	T		
		START:	MOV	AX,DATA
			MOV	DS,AX
			MOV	АН,00Н
			MOV	SI,OFFSET BCD
			MOV	CL,[SI]
			MOV	СХ,00Н
			INC	SI
			MOV	AL,[SI]
		BACK:	INC	SI
			ADC	AL,[SI]
			LOOP	BACK
			DAA	
			MOV	RESULT, AL
			MOV	AH,4CH
			INT	21H
CODE EN	NDS			

END START

CODETABLE:

Physical PhysicaPhysic	Physical address			Mne	emonic	
Segment address	Effective address	Label	Hex code	Opcode	Operands	Comments
1	1			1	1	

Physic addre	Physical address			Mne	emonic	
Segment address	Effective address	Label	Hex code	Opcode	Operands	Comments

PROCEDURE:

- Start the computer. Press Windows Button+'R'. The DOS command window will be opened.
- ➤ Type "cd\" and press Enter. On that you observe c:\>_.
- > Type "cd TASM" and press Enter. On that you observe c:\TASM>_.
- > Type "cd BIN" and press Enter. On that you observe c:\TASM\BIN>_.
- > Type *EDIT*(*space*) *FILENAME.ASM* (type edit and press space bar, after that type your current program name **ex: EDIT sum.asm**). Edit window will be opened. Type your assembly language program here and save it by using shortcut key (press) ALT+F+S. After completion of saving the program to exit press ALT+F+X. Again you will be back to command window.
- > To compile the program use command TASM_(space)*filename.asm*. Here you observe ASSEMBLING FILE: FILENAME.ASM

ERROR MESSAGES: NONE WARNING MESSAGES: NONE

Otherwise if any error messages are observed then go back to the edit window by using **edit filename.asm** command and correct it and again compile it. This process will be continued till you get no error messages.

Next compilation step is TLINK_(space)filename.obj. Here you observe WARNING MESSAGES: NO STACK

If it is not then you have to re-verify the program and re-compile till no stack warnings.

- > Next compilation step is TD_(space) filename.exe. Now you Observe "MS-DOS" window.
- Click on ok. Observe the highlighted line, press F7 or go to RUN and select SINGLE STEP EXECUTION.
- It will continue up to INT 21h instruction. When INT 21h is highlighted then go to VIEW and select DUMP. Then you get output
- > Note down the source and destination content and flag register contents.

RESULT:

Se	ource Dat		Destination Data									
Address	Giv	Given Data			Address			Obta	ined l	Result		
lags						1		1			1	
					1							F

Viva:

- 1. What is an Interrupts
- 2. What is an Opcode?
- 3. What is an Operand?
- 4. Explain the difference between a JMP and CALL instruction?
- 5. What is meant by Interrupt?
- 6. What is an Instruction?
- 7. What is Microcontroller and Microcomputer?
- 8. What is Assembler?
- 9. Define Variable?
- 10. Define Pipelining?

Experiment Number: Date:

(b) Multiplication and Division operations

Aim: To write an ALP to perform multiplication and division operations on 8 bit & 16 bit data

APPARATUS: Personal Computer TASM software

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 multiplication

Step7: Move the result to the memory location [SI] & [SI+02]

Step8: Copy the contents from the memory location [0000] toAX

Step9: Perform division

Step10: Move the result to the memory location[SI+04]&[SI+06]

Step11: Stop



Dept. of ECE



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 BX MUL [SI], AX MOV [SI+2], DX MOV MOV AX, N1 DX, 0000 MOV DIV BX MOV [SI+4], AX [SI+6], DX MOV MOV AH, 4CH 21H INT

CODE ENDS END START

CODETABLE:									
Physic addre	cal ess			Mne	emonic				
Segment address	Effective address	Label	Hex code	Opcode	Operands	Comments			

address			
Segment address Effective address Label Hex code Opcode Operands Comm	ments		

PROCEDURE:

- Start the computer. Press Windows Button +'R'. The DOS command window will be opened.
- ➤ Type "cd\" and press Enter. On that you observe c:\>_.
- > Type "cd TASM" and press Enter. On that you observe c:\TASM>_.
- > Type "cd BIN" and press Enter. On that you observe c:\TASM\BIN>_.
- Type *EDIT*(*space*) *FILENAME.ASM* (type edit and press space bar, after that type your current program name ex: EDIT sum.asm). Edit window will be opened. Type your assembly language program here and save it by using shortcut key (press) *ALT+F+S*. After completion of saving the program to exit press *ALT+F+X*. Again you will be back to command window.
- > To compile the program use command TASM_(space)*filename.asm*. Here you observe ASSEMBLING FILE: FILENAME.ASM

ERROR MESSAGES: NONE WARNING MESSAGES: NONE

Otherwise if any error messages are observed then go back to the edit window by using **edit filename.asm** command and correct it and again compile it. This process will be continued till you get no error messages.

Next compilation step is TLINK_(space)filename.obj. Here you observe WARNING MESSAGES: NO STACK

If it is not then you have to re-verify the program and re-compile till no stack warnings.

- > Next compilation step is TD_(space)filename.exe. Now you Observe "MS-DOS" window.
- Click on ok. Observe the highlighted line, press F7 or go to RUN and select SINGLE STEP EXECUTION.
- It will continue up to INT 21h instruction. When INT 21h is highlighted then go to VIEW and select DUMP. Then you get output
- > Noted own the source and destination content and flag register contents.

RESULT:

	Source Data							Destination Data							
	Address			Given Data			A	Address			Obta	ined]	Result		
															-
ľ															
F	ags				T				n	T	T	n	T	T	1
								_							

Viva:

- 1. What is assembly language?
- 2. What are machine language and assembly language programs?
- 3. What is the drawback in machine language and assembly language programs?
- 4. Define bit, byte and word.
- 5. What is a bus?
- 6. Why data bus is bi-directional?
- 7. Why address bus is unidirectional?
- 8. What are the modes in which 8086 can operate?
- 9. What is the data and address size in 8086?
- 10. Explain the function of M/IO in 8086.

Experiment Number:

Date:

SORTING AN ARRAY (a) Ascending Order

AIM: To write an ALP to perform sorting in ascending order of 'N' numbers.

APPARATUS: Personal Computer TASM software

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 (for ascending Order)

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: DecrementDX and jump to step5 if no zero

Step17: Stop





PROGRAM:									
ASSUME CS:CODE, D	S:DAT	A							
LIST DB 56H 12H 72H 32H									
COUNT FOU0003H	/211, 52	11							
CODE SEGMENT									
ORG 0000H									
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									

COI	DETABLE :	:				
Physic addre	cal ess	Mnemonic		emonic		
Segment address	Effective address	Label	Hex code	Opcode	Operands	Comments

Physic addre	cal ess			Mnemonic		
Segment address	Effective address	Label	Hex code	Opcode	Operands	Comments

PROCEDURE:

- Start the computer. Press Windows Button +'R'. The DOS command window will be opened.
- ➤ Type "cd\"and press Enter. On that you observe c:\>_.
- > Type "cd TASM" and press Enter. On that you observe c:\TASM>_.
- > Type "cd BIN" and press Enter. On that you observe c:\TASM\BIN>_.
- Type *EDIT*(*space*) *FILENAME.ASM* (type edit and press space bar, after that type your current program name ex: EDIT sum.asm). Edit window will be opened. Type your assembly language program here and save it by using shortcut key (press) *ALT+F+S*. After completion of saving the program to exit press *ALT+F+X*. Again you will be back to command window.
- > To compile the program use command TASM_(space) *filename.asm*. Here you observe ASSEMBLING FILE: FILENAME.ASM

ERROR MESSAGES: NONE WARNING MESSAGES: NONE

Otherwise if any error messages are observed then go back to the edit window by using **edit filename.asm** command and correct it and again compile it. This process will be continued till you get no error messages.

Next compilation step is TLINK_(space)filename.obj. Here you observe WARNING MESSAGES: NO STACK

If it is not then you have to re-verify the program and re-compile till no stack warnings.

- > Next compilation step is TD_(space)filename.exe. Now you Observe "MS-DOS" window.
- Click on ok. Observe the highlighted line, press F7 or go to RUN and select SINGLE STEP EXECUTION.
- It will continue up to INT 21h instruction. When INT 21h is highlighted then go to VIEW and select DUMP. Then you get output
- > Noted own the source and destination content and flag register contents.

RESULT:

	Source				Data				Destination Data						
	Address			Given Data			Α	Address			Obtained Result				
															_
															_
F	lags														
	Ĩ														

Viva

- 1. What is Physical Address?
- 2. What are the flags in 8086?
- 3. Why crystal is a preferred clock source?
- 4. What is Program counter?
- 5. What is1st/2nd/3rd/4th generation processor?

(b) Descending Order

AIM: To write an ALP to perform sorting in descending order of 'N' numbers.

APPARATUS:

Personal Computer TASM software

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 no carry (for descending Order)

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 tostep5 if no zero

Step17: Stop

PROGRAM:													
ASSUME	CS:CODE, DS	S:DATA	A										
DATA SEGM LIST	1ENT DB 56H, 12H, 7	72H, 32	Н										
COUN	COUNT EQU 0003H												
DATA ENDS CODE SEGM	IENT												
ORG 0000H													
STAR	T:	MOV	AX,DATA										
		MOV	DS,AX										
		MOV	CX,COUNT										
		MOV	DX,CX										
AGAI	N:	MOV	SI,OFFSET LIST										
		MOV	CX, DX										
BACK	Χ:	MOV	AL,[SI]										
		INC	SI										
		CMP	AL, [SI]										
		JNC	NEXT										
		XCHG	i [SI],AL										
		DEC	SI										
		MOV	[SI],AL										
		INC	SI										
NEXT		LOOP DEC	BACK DX										
		JNZ	AGAIN										
		MOV	AH,4CH										
		INT	21H										
CODE ENDS													
END START													

CODETABLE:												
Physic addre	cal ess			Mne	emonic							
Segment address	Effective address	Label	Hex code	Opcode	Operands	Comments						

Physic addro	cal ess			Mne	emonic					
Segment address	Effective address	Label	Hex code	Opcode	Operands	Comments				
	1									

PROCEDURE:

- Start the computer. Press Windows Button+'R'. The DOS command window will be opened.
- > Type "cd\"and press Enter. On that you observe c:\>_.
- > Type "cd TASM" and press Enter. On that you observe c:\TASM>_.
- > Type "cd BIN" and press Enter. On that you observe c:\TASM\BIN>_.
- > Type *EDIT*(*space*) *FILENAME.ASM* (type edit and press space bar, after that type your current program name **ex: EDIT sum.asm**). Edit window will be opened. Type your assembly language program here and save it by using shortcut key (press) ALT+F+S. After completion of saving the program to exit press ALT+F+X. Again you will be back to command window.
- > To compile the program use command TASM_(space)*filename.asm*. Here you observe ASSEMBLING FILE:FILENAME.ASM

ERROR MESSAGES: NONE WARNING MESSAGES:NONE

Otherwise if any error messages are observed then go back to the edit window by using **edit filename.asm** command and correct it and again compile it. This process will be continued till you get no error messages.

Next compilation step is TLINK_(space)filename.obj. Here you observe WARNING MESSAGES:NO STACK

If it Is not then you have to re-verify the program and re-compile till nostackwarnings.

- > Next compilation step is TD_(space)filename.exe. Now you Observe "MS-DOS" window.
- Click on ok. Observe the highlighted line, press F7 or go to RUN and select SINGLE STEP EXECUTION.
- It will continue up to INT 21h instruction. When INT 21h is highlighted then go to VIEW and select DUMP. Then you get output
- > Note down the source and destination content and flag register contents.

RESULT:

		S	Source	Data	Data				Destination Data						
	Address			Given Data			A	Address			Obtained Result				
Fl	ags														
										1					

Viva:

- 1. What is a Microprocessor?
- 2. What is Instruction Set?
- 3. What are the features of Intel 8086 ?
- 4. What is Logical Address?
- 5. What is The Effective Address?
FACTORIAL OF A GIVEN n- NUMBERS

AIM: To develop assembly language program to find out factorial of given 8 bit number and place result in specified location

APPARATUS:

Personal Computer TASM software

ALGORITHM:

Step1: Start

Step2: Initialize the addresses for input and result as 5000h, 6000h respectively.

Step3: Load AX register with the data 0001h.

Step4: Load SI, DI registers with input, result addresses respectively.

Step5: Load the input at SI location into CL register.

Step 6: Check whether the input is zero.

Step 7: If input is zero, go to step8. Otherwise calculate the factorial using recursive formula until input becomes 1.

Step8: Store the content of AX register in to the memory location pointed by DI.

Step 9: End of the program

FLOWCHART



PROGRAM: ASSUME CS:	CODE	, DS:I	DATA	
DATA SEGM	ENT			
FACT	DW		05H	
RESULT	DW	02H	DUP(00)	
DATA ENDS				
CODE SEGM	ENT			
START:	MOV	7	AX,DATA	
	MOV	7	DS,AX	
	MOV	7	SI,OFFSET FACT	
	MOV	7	AX,0001H	
	MOV	7	CL,[SI]	
	CMP		CL,00H	
	JZ		XX	
AGAIN:	MUL	4	CL	
	DEC		CL	
	JNZ		AGAIN	
XX:	MOV	7	[RESULT],AX	
	MOV	7	[RESULT+2],DX	
	MOV	7	AH,4CH	
	INT		21H	
CODE ENDS				

END START

CODETABLE:

Physic addre	cal ess			Mnemonic		
Segment address	Effective address	Label	Hex code	Opcode	Operands	Comments

Physic addre	cal ess			Mnemonic		
Segment address	Effective address	Label	Hex code	Opcode	Operands	Comments
		1	1			

PROCEDURE:

- Start the computer. Press Windows Button +'R'. The DOS command window will be opened.
- > Type "cd\"and press Enter. On that you observe c:\>_.
- > Type "cd TASM" and press Enter. On that you observe c:\TASM>_.
- > Type "cd BIN" and press Enter. On that you observe c:\TASM\BIN>_.
- > Type *EDIT*(*space*) *FILENAME.ASM* (type edit and press space bar, after that type your current program name **ex: EDIT sum.asm**). Edit window will be opened. Type your assembly language program here and save it by using shortcut key (press) ALT+F+S. After completion of saving the program to exit press ALT+F+X. Again you will be back to command window.
- > To compile the program use command TASM_(space)*filename.asm*. Here you observe ASSEMBLING FILE:FILENAME.ASM

ERROR MESSAGES: NONE WARNING MESSAGES:NONE

Otherwise if any error messages are observed then go back to the edit window by using **edit filename.asm** command and correct it and again compile it. This process will be continued till you get no error messages.

Next compilation step is TLINK_(space)filename.obj. Here you observe WARNING MESSAGES:NO STACK

If it Is not then you have to re-verify the program and re-compile till no stack warnings.

- > Next compilation step is TD_(space)filename.exe. Now you Observe "MS-DOS" window.
- Click on ok. Observe the highlighted line, press F7 or go to RUN and select SINGLE STEP EXECUTION.
- It will continue up to INT 21h instruction. When INT 21h is highlighted then go to VIEW and select DUMP. Then you get output
- > Note down the source and destination content and flag register contents.

RESULT:

So	urce Data	Des	stination Data
Address Given Data		Address	Obtained Result

Flags

-	0							

Viva:

- 1. What is the use of HLDA?
- 2. Explain about "LEA"?
- 3. Difference between "Shift" and "Rotate".
- 4. Difference between JMP and JNC?
- 5. List the String Manipulation Commands?
- 6. What are the 4 Segments?
- 7. What is the main use of ready pin?
- 8. Explain about Direction Flag?
- 9. What are the basic units of a microprocessor ?
- 10. What is Software and Hardware?

MICROPROCESSOR (8086) INTERFACING PROCEDURE WITH DIFFERENT INTERFACING MODULES

1) Switch on the PC.

2) Connect **ESA 86/88E** Trainer to the Host PC using 9 to 9 pin serial cable supplied by kit manufacturer.

3) Give power to **ESA 86/88E** Trainer.

- 4) Make the Dip switches on **ESA 86/88E** Trainer for 9600 baud rate and for serial Monitor (Select Dip switches 3, 4 and 7 should be ON for Serial Mode).
 - 5) Start **Win86E** application on Host PC. You should then see the following Dialog Box.



6) If the Trainer is connected to COM1 port on your Host PC & configured for 9600-baud rate, click **No**. Click **Yes** otherwise. If yes, you will see the following dialog box.

Port Settings BaudRate	Data Bits	<u>_</u>
Com Part	Stop Bits	-
	Parity Bits	-

7) Select the Baud Rate as per the dipswitch configuration on **ESA 86/88E** Trainer. Select theComPortwherethetrainerisconnected.ClickApply.Youshouldthenseethefollowingwindow.

8) Click on the 'download' option in the above window and browse '.hex' file of the current program and click ok. After downloading the file click on 'cmd' option in the above window and then to execute the current program give the command 'g 0000:2000' (i.e., starting address of the program) from keyboard. Check out the output on the interface or study card kit which has connected to the 8086 microprocessor kit.



9) In some cases if the communication is not established you will see the following Dialog box. In this case please check the Power on **ESA 86/88E** Trainer. Check whether Serial cable is connected to thePC&**ESA86/88E**Trainer and check the parameter by clicking Settings.

ŧ	Serial Communication	×
	Message Serial Communication Time Out !	
	Re <u>t</u> ry <u>S</u> ettings <u>C</u> ancel	

10) If the communication is established properly, from the displayed window you can work with **ESA 86/88E** Trainer. You can view the Registers, Memory Dumps, Download the hex images, Upload the hex dump to the Host PC, Run the downloaded hex images, single stepping, break points, watch windows, Memory modification etc., These options are explained in detail in online help of **Win86E** (**Win86E.hlp**).

DIGITAL TO ANALOG INTERFACE USING 8255

AIM: Interface Digital to Analog Interface Using 8255 (Generate Square Wave) **APPARATUS:** 8086 Kit

D/A interface module with FRCPower Supply,Cathode Ray Oscilloscope with CablesPersonal Computer with Interacting tool ESA86/88–2E

DESCRIPTIONOFDEMO PROGRAM

As can be seen from the circuit only 17 lines from the connector are used totally. The port A and port B of 8255 programmable peripheral interface are used as output ports. The digital inputs to the DACs are provided through the port A and port B of 8255. The analog outputs of the DACs are connected to the inverting inputs of opamps uA741 which act as current to voltage converters.

The outputs from the opamps are connected to points marked Xout & Yout at which the wave forms are observed on a CRO. (Port A is used to control Xout and Port B is used to control Yout). The reference voltage for the DACs is derived from an on-board voltage regulator uA723. It generates a voltage of about 8V.

The offset balancing of the opamps is done by making use of the two 10K pots provided. The output waveforms are observed at Xout and Yout on an oscilloscope. To use DAC initialize 8255 for Mode0 with Port A and Port B as output. Output the data on the appropriate port, and observe the output waveform at 'Xout' and 'Yout' on an oscilloscope. The Dual DAC interface can be used to generate different interesting waveforms using microprocessor. There are two eight bit digital to analog converters provided, based on DAC 0800. The digital inputs to these DACs are provided through the port A and port B of 8255 used as output ports. The analog output from the DACs are given to operational amplifiers which act as current to voltage converters. Two 10K ohms pots are provided for the offset balancing of op-amps.

The reference voltage needed for the DACs is obtained from a onboard voltage regulator uA723. The voltage generated by this regulator is about 8V. The outputs from the DACs vary between 0 to 5V corresponding to values between 00 to FF. Different waveforms can be observed at the opamp outputs depending upon the digital input patterns.

PROGRAM:

; Assume the interface is connected over J4 of the trainer.

; The trainer can be in KEYBOARD MODE or SERIAL MODE.

cs = 0

MODE EQU 80H PORTA EQU FFE0H

PORTB EQU FFE2H

CMD_PORT EQU FFE6H

ADDRESS	OPCODE	LABLE	MNEMONIC	COMMENTS
2000	BAE6 FF	DDAC:	MOVWDX,#CMD_PORT	
2003	B0 80		MOV BAL,#Mode	
2005	EE		OUTBDX	; Setup 8255
				; for Mode0,
				; Ports A,B,
				;C output
2006	B0 00		MOVB AL,#00H	; Start with
				; value of00
2008	BAE0 FF	Loop:	MOVW DX,#PortA	
200B	EE		OUTB DX ; Out to DAC 1	
200C	BA E2		MOVW DX,#PortB	
200F	FFEE		OUTB DX	;Out to DAC 2
2010	40		INCW AX	
2011	EBF5		JMP Loop	;Repeat forever

RESULT:

PART-B 8051 ASSEMBLY LANGUAGE PROGRAM AND INTERFACING

Experiment Number: Date:

Find number of 1's and 0's in a given 8-Bit Number

AIM: To write assembly language program to find number of 1's and 0's in a given 8-bit number

APPARATUS:

1. Personal Computer 2. KEIL Software

ALGORITHM:

- 1. Initialize R2,R3 Registers to store output values
- 2. Enter Number into R0 register
- **3.** Assign count value toR1register
- 4. Copy the given number into A register from R0 register
- 5. A register contents are Rotate right with carry one time. If no carry increment the content of R3 register otherwise increment R2 Register.
- **6.** Decrement count value. If it is zero stop the process and observe the result at R2&R3 registers. Otherwise repeat step 5 till count value is equal to zero.

PROGRAM:

START:

	MOV	R2,#00H
	MOV	R3,#00H
	MOV	R0,#56H
	MOV	R1,#08H
	MOV	A,R0
BACK:	RRC	А
	JNC	SKIP
	INC	R2
	JMP	GO
SKIP:	INC	R3
GO:	DJNZ	R1,BACK
	SJMP	START

CODETABLE:

Dhuminal			Mn	emonic	
address	Label	Hex code	Opcode	Operands	Comments

Dhysical	Mnemonic		emonic		
address	Label	Hex code	Opcode	Operands	Comments

Microprocessor & Microcontroller Lab

PROCEDURE:

- 1. Click on the Keil icon (Keil MicroVision)
- 2. Click on the Project(at the top menu)
- 3. Click on the new micro vision project.(Create New Project window displayed)
- 4. Select the desktop icon and create the new folder and to open the created folder
- 5. Give the filename and save it, then Select Device for Target 'Target1' window will be displayed.
- 6. In the database window double click on The Atmel make (Company name) and select the device window(AT89C51)then click on ok button.
- 7. MicroVision3windowwillbeappeared.
- 8. open the new file, type the program and save it with extensionExamplesarada.asm
- 9. click on+ symbol in PROJECT WORKSPACE then "source group will be appeared
- 10. Right click on source group and select the type of files (Ex .asm/.c)
- 11. click on ADD and CLOSE buttons then the file will be added to source group
- 12. click on + symbol beside of the "source group" then file will be appeared
- 13. Right click on the appeared file and build the target (shows the errors and warnings) if there is errors correct them
- 14. Debug the program by click on debug at the top of menu
- 15. Run the program (press F5)
- 16. Select the corresponding ports see the results

RESULT:

Source Data		Destination Data	
Address	Given Data	Address	Obtained Result

Viva:

- 1. Write the flags of 8086.
- 2. What are the interrupts of 8086?
- 3. How clock signal is generate in 8086? What is the maximum internal clock frequency of 8086?
- 4. Write the special functions carried by the general purpose registers of 8086.
- 5. What is pipelined architecture?
- 6. What are the functional units available in 8086 architecture?
- 7. List the segment registers of 8086.
- 8. Define machine cycle.
- 9. Why interfacing is needed for 1/0devices?
- 10. What is the difference between CPU bus and system bus?

Find Average of N-numbers

AIM: To write assembly language program to find average of N-numbers APPARATUS:

1. Personal Computer 2. KEIL Software ALGORITHM:

- **1.** Take the numbers from 5000h location using DPTR register and send first number into R0 register through A register.
- 2. Assign count value to R1 register
- 3. Move the count value into B registers through A register
- 4. Move second number into A register
- **5.** Add the value of A register to R0 register value. And the result is stored in A register. The result moved to R0 through A register.
- **6.** Increment data pointer and the next number moved to A register and again it is added with R0 register and the result copied in toR0 through A register.
- **7.** This operation is continued till the count value is equal to zero. After that R0 value is moved to A register.
- 8. Divide the A register value with B Register. The result stored in 6000h location.

PROGRAM:

START:	MOV	DPTR,#	#5000H
		MOVX A	,@DPTR
		MOV	R0,A
		MOV	R1,#06H
		MOV	A,R1
		MOV	B,A
		INC	DPTR
BACK:	MOVX	A,@DI	PTR
	ADD	A,R0	
	MOV	R0,A	
	INC	DPTR	
	DJNZ	R1,BAC	CK
	MOV	A,R0	
	DIV	AB	
	MOV	DPTR,#	6000H
	MOVX	@DPTH	R,A
	SJMP	START	

CODETABLE

			Mnemonic		
Physical address	Label	Hex code	Opcode	Operands	Comments
	ļ				

Dhusiaal			Mn	emonic	
address	Label	Hex code	Opcode	Operands	Comments

PROCEDURE:

- 1. Click on the Keil icon (Keil MicroVision)
- 2. Click on the Project(at the top menu)
- 3. Click on the new micro vision project.(Create New Project window displayed)
- 4. Select the desktop icon and create the new folder and to open the created folder
- 5. Give the filename and save it, then Select Device for Target 'Target1' window will be displayed.
- 6. In the database window double click on he Atmel make (Company name) and select the device window(AT89C51)then click on ok button.
- 7. MicroVision3windowwillbeappeared.
- 8. open the new file, type the program and save it with extensionExamplesarada.asm
- 9. click on+ symbol in PROJECT WORKSPACE then "source group will be appeared
- 10. Right click on source group and select the type of files (Ex .asm/.c)
- 11. click on ADD and CLOSE buttons then the file will be added to source group
- 12. click on + symbol beside of he "source group" then file will be appeared
- 13. Right click on the appeared file and build the target (shows the errors and warnings) if there is errors correct them
- 14. Debug the program by click on debug at the top of menu
- 15. Run the program (press F5)
- 16. Select the corresponding ports see the results

RESULT:

Source Data		Destination Data	
Address	Given Data	Address	Obtained Result

Viva:

- 1. What does memory-mapping mean?
- 2. What is interrupt 1/0?
- 3. Why EPROM is mapped at the beginning of memory space in 8085system?
- 4. What is the need for timing diagram?
- 5. Define opcode and operand.

VERIFY TIMER/ COUNTER IN 8051

AIM: To observe the timer in different modes

APPARATUS:

1. Personal Computer 2. KEIL Software

ALGORITHM:

- 1. Select mode of timer
- 2. Select register and enter the data
- 3. Set timer
- 4. Clear timer flag
- 5. Clear timer register
- 6. Jump start

PROGRAM:

START:	MOV	TMOD,#01H
HERE:	MOVE	TL0,#0FEH
DELAY:	SETBIT	TR0
AGAIN:	JNB	TF0
	CLR	TR0
	CLR	TF0
	JMP	START
	END	

CODETABLE

Physical			Mn	emonic			
address	Label	Hex code	Opcode	Operands	Comments		

			Mn	emonic	
address	Label	Hex code	Opcode	Operands	Comments

PROCEDURE:

- 1. Click on the Keil icon (Keil MicroVision)
- 2. Click on the Project(at the top menu)
- 3. Click on the new micro vision project.(Create New Project window displayed)
- 4. Select the desktop icon and create the new folder and to open the created folder
- 5. Give the filename and save it, then Select Device for Target 'Target1' window will be displayed.
- 6. In the database window double click on he Atmel make (Company name) and select the device window(AT89C51)then click on ok button.
- 7. MicroVision3windowwillbeappeared.
- 8. open the new file, type the program and save it with extensionExamplesarada.asm
- 9. click on+ symbol in PROJECT WORKSPACE then "source group will be appeared
- 10. Right click on source group and select the type of files (Ex .asm/.c)
- 11. click on ADD and CLOSE buttons then the file will be added to source group
- 12. click on + symbol beside of he "source group" then file will be appeared
- 13. Right click on the appeared file and build the target (shows the errors and warnings) if there is errors correct them
- 14. Debug the program by click on debug at the top of menu
- 15. Run the program (press F5)
- 16. Select the corresponding ports see the results

RESULT:

Source Data		Destination Data	
Address	Given Data	Address	Obtained Result

Viva:

- 1. What is opcode fetch cycle?
- 2. Why status signals are provided in microprocessor?
- 3. What is interrupt acknowledge cycle?
- 4. How the interrupts are affected by system reset?
- 5. What is Software interrupts?

TRAFFIC LIGHT INTERFACE

AIM: To write an ALP program to interface traffic light system to 8051 APPARATUS: ESA 8051 Microcontroller kit Keyboard Power supply Traffic light Interface card INTERFACE IMAGE:





PROGRAM:

Assume the interface is connected to J10of trainer. The trainer ESA-51can be in KEYBOARD MODE or in SERIAL MODE. The Traffic System moves from one state to next state after a fixed time delay. PORT A EQU E800H PORT_B EQU E801H PORT_C EQU E802H CWR EQU E803H ADDRESS OPCODE LABEL **MNEMONIC COMMENTS** 8000 ORG8000H 8000 90 E8 03 MOVDPTR.#CWR :Initialise 8255 ;for MODE0. 8003 74 80 MOVA,#80H 8005 F0 MOVX@DPTR,A :All Ports as ;O/P Ports. 8006 90 0000 AGAIN: MOVDPTR,#0000H ;Table of Port 8009 E0 NEXTST: MOVXA,@DPTR ;values. 800A C0 82 PUSHDPL 800C C0 83 PUSHDPH 800E 90 E8 00 MOV DPTR, #PORT_A; PortA value. MOVX@DPTR,A 8011 F0 8012 D0 83 POPDPH D0 82 POPDPL 8014 8016 A3 **INCDPTR** E0 MOVXA,@DPTR 8017 8018 C0 82 PUSHDPL 801A C0 83 PUSHDPH 801C 90 E8 01 MOVDPTR, #PORT_B ;Port B value. 801F F0 MOVX@DPTR,A 8020 D0 83 POPDPH D0 82 8022 POPDPL 8024 A3 **INCDPTR** 8025 E0 MOVXA,@DPTR 8026 C0 82 PUSHDPL C0 83 8028 PUSHDPH 802A 90 E8 02 MOVDPTR, #PORT C ;Port C value. MOVX@DPTR,A 802D F0 802E D0 83 POPDPH POPDPL 8030 D0 82 8032 A3 **INCDPTR** 12 803D 8033 CALLDELAY ;Provide delay. 8036 E5 82 MOVA, DPL 8038 B4 01 CE CJNEA,#1EH,NEXTST 80 C9 **SJMPAGAIN** 803B DELAY: MOVR2,#07H 803D 7A07 ;Delay routine. 803F 7CFF LOOP3: MOVR4,#0FFH 7BFF 8041 LOOP2: MOVR3,#0FFH 8043 1B LOOP1: DECR3 8044 BB00 FC CJNER3,#00,LOOP1 8047 1CDECR4 8048 BC00 F6 CJNER4,#00,LOOP2 804B 1A DECR2 **BA00 F0** 804C CJNER2,#00,LOOP3 804F 22 RET

ENTER THE BELOW GIVEN DATA FROM 0000 TO 001E AT DATA MEMORY 0000 10 81 7A PORTS : DB 10H,81H,7AH ;State1 0003 44 44 F0 DB 44H,44H,0F0H ;All Ambers ON. 0006 08 11 E5 DB 08H,11H,0E5H ;State2 0009 44 44 F0 DB 44H,44H,0F0H ;All Ambers ON. 000C 81 10 DA DB 81H,10H,0DAH ;State 3 000F 44 44 F0 DB 44H,44H,0F0H ;All Ambers ON. 0012 11 08 B5 DB 11H,08H,0B5H ;State4 0015 44 44 F0 DB 44H,44H,0F0H ;All Ambers ON. 0018 88 8800 DB 88H,88H,00H ; State5 001B 44 44 F0 DB 44H,44H,0F0H ;All Ambers ON. 001E 00 DB 00H ;DUMMY.

RESULT:

PACKED BCD TO UNPACKED BCD CONVERSION

ABSTRACT: Write a program to convert packed BCD number into Unpacked BCD number.

REGISTERS USED: AL, BL

PORTS USED: None.

ALOGARITHM:

Step1: Start

Step2: Initialize the data segment

Step3: Move packed number into AL register

Step4: Move packed number into BL register

Step5: Initialize the count CX with 04h.

Step6: AND operation AL with 0Fh.

Step7: AND operation BL with 0F0h.

Step8: Rotate right without carry operation on BL by CL times.

Step9: Move the result into location 0000 and 0001.

Step10: Stop.



PROGRAM:

ASSUME CS: CODE, DS: DATA

DATA SEGMENT

N EQU 29H

RESULT DB 02H DUP (0)

DATA ENDS

CODE SEGMENT

ORG 2000h

START: MOV AX, DATA

MOV DS, AX

MOV AL, N

MOV BL, N

MOV CL, 04H

AND AL, 0Fh

AND BL, 0F0h

RORBL, CL

MOV [RESULT], BL

MOV [RESULT+1], AL

MOV AH, 4Ch

INT 21h

CODE ENDS

END START

CODETABLE:

Dhusiaal			Mne	emonic	
address	Label	Hex code	Opcode	Operands	Comments

			Mne	emonic	
address	Label	Hex code	Opcode	Operands	Comments

PROCEDURE:

- Start the computer. Press Windows Button +'R'. The DOS command window will be opened.
- ➤ Type "cd\" and press Enter. On that you observe c:\>_.
- > Type "cd TASM" and press Enter. On that you observe c:\TASM>_.
- > Type "cd BIN" and press Enter. On that you observe c:\TASM\BIN>_.
- Type EDIT_(space) FILENAME.ASM (type edit and press space bar, after that type your current program name ex: EDIT sum.asm). Edit window will be opened. Type your assembly language program here and save it by using shortcut key (press) ALT+F+S. After completion of saving the program to exit press ALT+F+X. Again you will be back to command window.
- > To compile the program use command TASM_(space)*filename.asm*. Here you observe ASSEMBLING FILE: FILENAME.ASM

ERROR MESSAGES: NONE WARNING MESSAGES: NONE

Otherwise if any error messages are observed then go back to the edit window by using **edit filename.asm** command and correct it and again compile it. This process will be continued till you get no error messages.

Next compilation step is TLINK_(space)filename.obj. Here you observe WARNING MESSAGES: NO STACK

If it is not then you have to re-verify the program and re-compile till no stack warnings.

- > Next compilation step is TD_(space)filename.exe. Now you Observe "MS-DOS" window.
- Click on ok. Observe the highlighted line, press F7 or go to RUN and select SINGLE STEP EXECUTION.
- It will continue up to INT 21h instruction. When INT 21h is highlighted then go to VIEW and select DUMP. Then you get output
- > Noted own the source and destination content and flag register contents.

RESULT:

Source Data				Destination Data									
Address		Given Data		Α	Address			Obtained Result					
lags		1											

TRANSFERBLOCKOFDATA

ABSTRACT: Assembly language program to transfer a block of data.

PORT USED: None.

REGISTERS USED: AX, BL.

ALGORITHM:

Step1: Start

Step2: Initialize data segment & extra segment

Step3: Load CX register with count

Step4: Initialize DI with memory location

Step5: Load SI with offset list

Step6: Repeat the process of moving string byte from SI to DI until count equals to zero

Step7: Stop



PROGRAM:

ASSUME CS: CODE, DS: DATA, ES: DATA

DATA SEGMENT

LIST DB 'ADITYA'

COUNT EQU 06H

DATA ENDS

CODE SEGMENT

ORG 1000H

MOV	DS, AX
	· · · · · ·

- MOV ES, AX
- MOV CX, COUNT
- MOV DI, 5000H
- LEA SI, LIST
- **REP** MOVSB

MOV AH, 4CH

INT21H

CODE ENDS

END START

CODETABLE:						
Physical		Mn	emonic			
address	Label	Hex code	Opcode	Operands	Comments	
1		1	1			
Physical address	Label	Hex code	Mne	emonic		
---------------------	-------	----------	---------	----------	----------	
			Op code	Operands	Comments	

PROCEDURE:

- Start the computer. Press Windows Button +'R'. The DOS command window will be opened.
- ➤ Type "cd\" and press Enter. On that you observe c:\>_.
- > Type "cd TASM" and press Enter. On that you observe c:\TASM>_.
- > Type "cd BIN" and press Enter. On that you observe c:\TASM\BIN>_.
- Type EDIT_(space) FILENAME.ASM (type edit and press space bar, after that type your current program name ex: EDIT sum.asm). Edit window will be opened. Type your assembly language program here and save it by using shortcut key (press) ALT+F+S. After completion of saving the program to exit press ALT+F+X. Again you will be back to command window.
- > To compile the program use command TASM_(space)*filename.asm*. Here you observe ASSEMBLING FILE: FILENAME.ASM

ERROR MESSAGES: NONE WARNING MESSAGES: NONE

Otherwise if any error messages are observed then go back to the edit window by using **edit filename.asm** command and correct it and again compile it. This process will be continued till you get no error messages.

Next compilation step is TLINK(space)filename.obj. Here you observe WARNING MESSAGES: NO STACK

If it is not then you have to re-verify the program and re-compile till no stack warnings.

- > Next compilation step is TD_(space)filename.exe. Now you Observe "MS-DOS" window.
- Click on ok. Observe the highlighted line, press F7 or go to RUN and select SINGLE STEP EXECUTION.
- It will continue up to INT 21h instruction. When INT 21h is highlighted then go to VIEW and select DUMP. Then you get output
- ▶ Noted own the source and destination content and flag register contents.

RESULT:

Source Data						Destination Data								
Address		Given Data			A	Address				Obtained Result				
													_	
ags														