DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING LAB MANUAL Microprocessors & Microcontrollers Lab II - B. Tech. II - Semester



PRASAD V POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY (Autonomous, Accredited by NBA & NAAC, an ISO 9001:2008 certified institution) (Sponsored by Siddhartha Academy of General & Technical Education) VIJAYAWADA – 520 007, ANDHRA PRADESH

Microprocessors & Microcontrollers Lab MANUAL

HARTHA INS

Prepared by Mr. K. Phani Rama Krishna & Dr. Haji. Habibulla Md.

PRASAD V POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING Microprocessors & Microcontrollers Lab

LIST OF EXPERIMENTS

Syllabus			
Expt. No.	Contents	Mapped CO	
Ι	16-bit Signed and unsigned Arithmetic operations, ASCII – arithmetic operations	CO1,CO4	
II	Arithmetic operations – Multi byte Addition and Subtraction	CO1,CO4	
III	Logical operations, Sum of Squares, Sum of Cubes	CO1,CO4	
IV	Write ALP to find smallest, largest number, arrange numbers in Ascending order, Descending order in a given series.	CO1,CO4	
V	Using string operation and Instruction prefix: Move Block, Reverse string, String comparison HAINS 775	CO1,CO4	
VI	Introduction to MSP430 launch pad and Programming Environment. (Study Experiment)	CO2, CO4	
VII	Read input from switch and Automatic control/flash LED (software delay).	CO2,CO3,CO4	
VIII	Read Temperature of MSP430 with the help of ADC.	CO2, CO3,CO4	
IX	Interrupts Programming Example Using GPIO	CO2, CO3,CO4	
Х	Use Of Comparator To Compare The Signal Threshold Level	CO2, CO3, CO4	

Additional Experiments:

1. Average of numbers

2. Conversion of Packed BCD to Unpacked BCD, Packed BCD to ASCII

INSTRUCTIONS TO THE STUDENTS

1. Students are required to attend all labs.

2. Students have to bring the lab manual cum observation book, record etc. along with them whenever they come for lab work.

3. Should learn the prelab questions. Read through the lab experiment to familiarize themselves with the components and assembly sequence.

4. Should utilize 3 hours' time properly to perform the experiment and to record the readings. Do the calculations, and take signature from the instructor.

5. If the experiment is not completed in the stipulated time, the pending work has to be carried out in the leisure hours or extended hours.

6. Should submit the completed record book according to the deadlines set up by the instructor.

7. For practical subjects there shall be a continuous evaluation during the semester for 15 internal marks and 35 end examination marks.

8. Out of 15 internal marks, 10 marks shall be awarded for day-to-day work and 5 marks to be awarded by conducting an internal laboratory test.



EXPERIMENT-1

16-bit Signed and unsigned Arithmetic operations, ASCII – arithmetic operations

AIM: To perform 16-bit Signed and unsigned Arithmetic operations, ASCII – arithmetic operations using TASM.

DHARTHA INS 7

පණුඩ්ථ් ද

VIJAYAWADA

ECHNOLOG'

Experimental Requirements: PC loaded with TASM software ,8086 microprocessor kit and power supply.

Procedure for doing **DEBUG** program:

Step1:

Open the dosbox icon placed in the desktop

Step 2:

type the following

mount c c:\8086

and press enter

then dos box will be mounted to the local directory.

Step 3:

type c:

and press enter

You will be getting the screen as:

c:\

Step4:

Now Type

debug

,and press enter

Now you are going to get hyphen symbol

Step5: Type

a ,and press enter

Type your program as shown in the attachment.

Step6:

Type

R IP

,and press enter

The instruction pointer should point to the starting address of the program.

If not, type the starting address of the program.For Example, 0100

Step7:

We need to do single step execution.

For single step execution:

type

t as shown in the attachment.

Till in the end of the program we need to repeat.

Procedure for TASM:

- 1. Switch on the PC, press windows+R then enter CMD.
- Find the folder where TASM is located. check whether TASM.EXE, TLINK.EXE, TD.EXE are present or not
- 3. Enter into the directory where TASM is located by using cd... or directory name:
- Type cd tasm in which the three files are present .Now we will be getting into c: \ or d:\ with tasm directory.
- 5. Type edit then a new window will be opened in which the program is entered.
- 6. After entering the program save the file with <filename.asm>.
- 7. Check for the errors or warnings by using TASM <filename> and press enter...
- 8. If there are no errors, then type TLINK <filename> to compile the file. If errors go back to the edit and do the necessary corrections and repeat the previous step.
- Next type td <filename > to debug the executable file then will be getting the message program has no symbol table, press ok and then write down the instructions, registers and flags status before execution.
- 10. For step by step execution press F8.and for direct execution press F9 and then write down the instructions, registers and flags status after execution .Go to dump if required for noting down the required inputs and outputs.

Programs	IONS			
8-BIT OPERAT	IUNS			
ASSUME CS: COD DATA SEGMENT OPR1 DB 78H OPR2 DB 23H RES DB 1 DUP (0H DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV AX, DATA MOV DS, AX MOV AL, OPR1 MOV BL, OPR2 ADD AL, BL MOV RES, AL INT 03H CODE ENDS END START END		DHARTHA INS 7/7	UTE OF TEC	
ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	RAS.	1	6	
	BR BR		GY	
	ġ.	S & S & S & S & S & S & S & S & S & S &		
		ఆత్మదీపోభన	//	
		VIJAYAWADA		

Result: INPUT: AL: OUTPUT: AL: FLAG STATUS: Theoretical Calculations:

BL:

AH:

2. SUBTRACTION:

ASSUME CS: CODE, DS: DATA DATA SEGMENT OPR1 DB 36H OPR2 DB 23H RES DB 1 DUP (0H) DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS, AX MOV AL, OPR1 MOV BL, OPR2 SUB AL, BL MOV RES, AL INT 03H CODE ENDS END START END

SIDDHARTHA INS TITU

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
			NHN	
	SA		OLO	
	R/A		OGV	
	\$			
		ఆత్రబ్రేశిభన		
		e)	
		VIJAYAWADA		

Result: INPUT: AL: OUTPUT: AL: FLAG STATUS: Theoretical Calculations:

BL:

AH:

3. MULTIPLICATION:

ASSUME CS: CODE, DS: DATA DATA SEGMENT OPR1 DB 15H OPR2 DB 05H RES DW 1 DUP (0H) DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS, AX MOV AL, OPR1 MOV BL, OPR2 MOV AH,00H MUL BL MOV RES, AX INT 03H CODE ENDS END START END

SIDDHARTHA INS TITU

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	RA	/ 7	OG	
	d Q	R R R R R R R R R R R R R R R R R R R	~	
		105 B-78055		
		00,00.00]]	
		VIJAYAWADA		

Result: INPUT: AL: OUTPUT: AL: FLAG STATUS:

Theoretical Calculations:

BL:

AH:

4. DIVISION:

ASSUME CS: CODE, DS: DATA DATA SEGMENT OPR1 DB 20H OPR2 DB 05H RES DW 1 DUP (0H) DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS, AX MOV AL, OPR1 MOV BL, OPR2 MOV AH,00H DIV BL MOV RES, AX INT 03H CODE ENDS END START

Standartha INS TITU

END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	V.	h	СН	
	AD		NO	
	AS	////	LOC	
	E Salar	e e	Ye	
	Ŷ.		9	
		. ఆత్కె బిఫో భర్		
		VIDAIANADA		

Result: INPUT: AL: OUTPUT: AL: FLAG STATUS: Theoretical Calculations:

BL:

AH:

1.Addition

-A 072A:0100 -A 400 072A:400 MOV AL,55 072A:4002 MOV BL,32 072A:4004 ADD AL,BL 072A:4006 -R IP IP 0100 -R IP 4000 -T

-G

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	G17	DHARTHA INS TIT		
	È			
	71		OF	
	ЬО		TE	
	>	III nor	СН	
	SAD		ION	
	3AS	/ \	_00	
		Ŕ		
		ఉత్త బికాటం		
		VIJAYAWADA		

Result: INPUT: AL: OUTPUT: AL: FLAG STATUS:

BL:

AH:

2.Subraction :

-A 072A:0100 -A 4000 072A:4000 MOV AL,37 072A:4002 MOV BL,36 072A:4004 SUB AL,BL 072A:4006 -R IP IP 0100 -R IP 4000 -T -G

ADDRESS OPCODE MNEMONIC OPERAND COMMENTS				
ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	5		in the second se	
	Ē	$\left(n \right)$	T	
	0 4			
	>	l Ing		
	AD		Z	
	AS	4	6	
	PR		GY	
	\$		\$	
		ఆత.బీఫేిచన		
	()	(d)	//	
		VIJAYAWADA		

Result: INPUT:	
AL:	BL:
OUTPUT:	
AL:	AH:
FLAG STATUS:	

3. Multiplication:

-A 072A:0100 -A 400 072A:400 MOV AL,54 072A:4002 MOV BL,21 072A:4004 MUL,BL 072A:4006 INT 03 -R IP IP 0100 -R IP 4000 -T -G

SIDDHARTHA INS TITU

	2		0	
ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	PG		TEC	
	2	nnør	HN	
	SAL		IOL	
	RA	/ 7	00	
	۵ C		~	
		ఆష్ట బిపిశింగన		
	(())	
		VIJAYAWADA	//	
			7	
	I	1		

Result: INPUT: AL: OUTPUT: AL: FLAG STATUS:

BL:

AH:

4. Divison -A 072A:0100 -A 400 072A:400 MOV AL,24 072A:4002 MOV BL,4 072A:4004 DIV,BL 072A:4006 -R IP IP 0100 -R IP 4000 -T -G

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	Š		m	
	14	$\left(\begin{array}{c} \\ \\ \\ \end{array} \right)$	F	
	PO		TEO	
	>		HZ	
	SAI		OL	
	RA		OG	
	¢ ا		\$	
		495 9 3 3 3 3 5		
	(00,00,00	11	
		VIJAYAWADA	/	
			7	

Result: INPUT:	
AL:	BL:
OUTPUT:	
AL:	AH:
FLAG STATUS:	

<u>16-bit ADDITION:</u>

ASSUME CS: CODE, DS: DATA DATA SEGMENT OPR1 DW 78BCH OPR2 DW 23FEH RES DW 1 DUP (0H) DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS, AX AX, OPR1 MOV MOV BX, OPR2 ADD AX, BX MOV RES, AX INT 03H CODE ENDS END START END

 ADDRESS
 OPCODE
 MNEMONIC
 OPERAND
 COMMENTS

 Image: Imag

BX:

DX:

SIDDHARTHA INS TIT

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

<u>16-bit SUBTRACTION:</u>

10-DIL DUDINA				
ASSUME CS: CO	ODE, DS: DATA			
DATA SEGMEN OPR1 DW 36BBI				
OPR2 DW 23CCI	Н			
RES DW 1 DUP ((0H)			
DATA ENDS				
CODE SEGMEN	NT			
START:				
MOV AX, DAT	ΓA			
MOV DS, AX				
MOV AX, OPR				
MOV BX, OPR				
SUB AX, BX	7			
MOV RES, AX INT 03H	GIP	DHARTHA INS TIT		
CODE ENDS	2			
	110		ିକ	
END START END	POTLU		OF TE	
END START	OPCODE	MNEMONIC	T I	COMMENTS
END START END	OPCODE	(dh)	F TEC	COMMENTS
END START END	OPCODE	(dh)	F TEC	COMMENTS
END START END	OPCODE	(dh)	OPERAND	COMMENTS
END START END	OPCODE	(dh)	OPERAND	COMMENTS
END START END	OPCODE	(dh)	OPERAND	COMMENTS
END START END	OPCODE	(dh)	OPERAND	COMMENTS
END START END	OPCODE	(dh)	OPERAND	COMMENTS
END START END	OPCODE	(dh)	OPERAND	COMMENTS
END START END	OPCODE	(())	OPERAND	COMMENTS
END START END	OPCODE	(())	OPERAND	COMMENTS
END START END	OPCODE	(())	OPERAND	COMMENTS
END START END	OPCODE	(())	OPERAND	

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

BX:

<u>16-bit MULTIPLICATION:</u>

ASSUME CS: C DATA SEGME OPR1 DW 1506F OPR2 DW 0AC0 RES1 DW 1 DUF RES2 DW 1 DUF DATA ENDS CODE SEGME START: MOV AX, DAT MOV AX, OPR MOV DS, AX MOV AX, OPR MOV BX, OPF MOV BX, OPF MOV BX, 000 MUL BX MOV RES1, A MOV RES1, A MOV RES2, D2 INT 03H CODE ENDS	H 5H 9 (0H) 9 (0H) NT TA TA R1 R2 0H X	DHARTHA INS 7/7	UTE OF TEC	
END	RASAD	11.751	TECHNOLOG	
ADDRESS	OPCODE 🔅	MNEMONIC	OPERAND	COMMENTS
		<i>ఆత్మ</i> బీపేశిభవ		
		VIJAYAWADA	//	

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

BX:

<u>16-bit</u> . <u>DIVISION</u> ASSUME CS: CO	ODE, DS: DATA			
DATA SEGMEN				
OPR1 DW 0F506 OPR2 DW 0AC50				
RES1 DW 1 DUP				
RES2 DW 1 DUP				
DATA ENDS	(011)			
CODE SEGMEN	NT			
START:				
MOV AX, DAT	Ϋ́A			
MOV DS, AX				
MOV AX, OPR	1			
MOV BX, OPR				
MOV DX,0000				
DIV BX		DHARTHA INS 7/7		
MOV DECI A			/ ,	
,	X			
MOV RES2,DX	X K Š		TH OF	
MOV RES2,DX INT 03H	X STILL		OF TE	
MOV RES2,DX INT 03H CODE ENDS	V. POTLUA	Mha	TECHI	
MOV RES2,DX INT 03H CODE ENDS END START	V. POTLUA	Mbe	TECHNOL	
MOV RES2,DX INT 03H CODE ENDS END START END	ASAD V. POTLUP	mbe	TECHNOLO OPER A ND	COMMENTS
MOV RES2,DX INT 03H CODE ENDS END START	V. POTLUA	MNEMONIC	OPERAND	COMMENTS
MOV RES2,DX INT 03H CODE ENDS END START END	COPCODE	MNEMONIC		COMMENTS
MOV RES2,DX INT 03H CODE ENDS END START END	COPCODE	MNEMONIC		COMMENTS
MOV RES2,DX INT 03H CODE ENDS END START END	COPCODE	MNEMONIC		COMMENTS
MOV RES2,DX INT 03H CODE ENDS END START END	COPCODE	MNEMONIC ఆక్షబిపోభవ		COMMENTS
MOV RES2,DX INT 03H CODE ENDS END START END	COPCODE	MNEMONIC ఆక్షబిపోభవ		COMMENTS
MOV RES2,DX INT 03H CODE ENDS END START END	COPCODE	MNEMONIC ఆక్షబిపోభవ		
MOV RES2,DX INT 03H CODE ENDS END START END	COPCODE	MNEMONIC ఆక్షబిపోభవ		
MOV RES2,DX INT 03H CODE ENDS END START END	COPCODE	MNEMONIC ఆక్షబిపోభవ		
MOV RES2,DX INT 03H CODE ENDS END START END	COPCODE	MNEMONIC ఆక్షబిపోభవ		
MOV RES2,DX INT 03H CODE ENDS END START END	COPCODE	MNEMONIC ఆక్షబిపోభవ		

Nesul. INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

BX:

DATA CECMEN	DE, DS: DATA			
DATA SEGMEN				
OPR1 DW 0BCDE				
OPR2 DW 0ABCE				
RES DW 1 DUP (0 DATA ENDS)H)			
CODE SEGMEN	Т			
START:	1			
MOV AX, DATA	4			
MOV DS, AX				
MOV AX, OPR1				
MOV BX, OPR				
STC		DHARTHA INS 7/7		
ADD AX, BX MOV RES, AX	2511			
INT 03H	Ē		·97	
CODE ENDS	POT			
END START	>	llino	TECHNO	
END	SAD			
END ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	0	MNEMONIC		COMMENTS
	0	MNEMONIC క		COMMENTS
	0	MNEMONIC క		COMMENTS
	0	MNEMONIC రాత్యదిపోధన VIJAYAWADA		COMMENTS
	0	MNEMONIC లక్కటిహాధక VIJAYAWADA		COMMENTS
	0	MNEMONIC రెక్కటిహాధన VIJAYAWADA		COMMENTS
	0	MNEMONIC కి. సరక్షదిహిధన VIJAYAWADA		COMMENTS
	0	MNEMONIC క రెక్కటిహిధన VIJAYAWADA		COMMENTS
	0	MNEMONIC రాక్యదిపౌశిధన VIJAYAWADA		COMMENTS
	0	MNEMONIC అత్యమిశిధన VIJAYAWADA		COMMENTS
	0	MNEMONIC రెక్యెటిహిభన VIJAYAWADA		COMMENTS
ADDRESS	0	MNEMONIC రావ్యదిపౌశిభవ VIJAYAWADA		COMMENTS
ADDRESS	0	ుత్యదిపోధన VIJAYAWADA		COMMENTS
ADDRESS	0	MNEMONIC ເອຣູພິສະດຸລ໌ VIJAYAWADA		COMMENTS

A COLIME CO. CO.				
ASSUME CS: CO	DDE, DS: DATA			
DATA SEGMEN	ΤI			
OPR1 DW 0BCDI	EH			
OPR2 DW 0ABCI	DH			
RES DW 1 DUP (0H)			
DATA ENDS				
CODE SEGMEN	JT			
START:				
MOV AX, DAT	A			
MOV DS, AX	1			
MOV AX, OPR				
MOV BX, OPR CLC	.2			
ADC AX, BX				
MOV RES, AX		DHARTHA INS TIT		
INT 03H	01511			
CODE ENDS			97 97	
END START	6	(1)		
END		111hB	ECH	
ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	SAS		0	
			- HGY	
		() () () () () () () () () ()		
	Ŷ			
	÷	ఆత్మదిపోభవ		
	÷	ఆత్మదీపేశభవ VIJAYAWADA		
Result:				

AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

BX:

DX:

2. SUBTRACTION:

ASSUME CS: CODE, DS: DATA DATA SEGMENT **OPR1 DW 0BCDEH OPR2 DW 0ABCDH** RES DW 1 DUP (0H) DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS, AX MOV AX, OPR1 MOV BX, OPR2 STC SUB AX, BX MOV RES, AX INT 03H CODE ENDS END START END

SIDDHARTHA INS TITUE

			l	
ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	RA		OG	
	÷			
		ఆశ్రవిపోభన		
		6))	
		VIJAYAWADA		

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

BX:

ASSUME CS: CODE, DS: DATA DATA SEGMENT OPR1 DW 0BCDEH **OPR2 DW 0ABCDH** RES DW 1 DUP (0H) DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS, AX MOV AX, OPR1 MOV BX, OPR2 CLC SBB AX, BX MOV RES, AX INT 03H CODE ENDS END START END

SIDDHARTHA INS TITU

	Ĕ			,
ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	SAD		NOI	
	SAS	/ \	_00	
	۵. B	Ŕ	Υ:	
	5	and first		
		.ఆత్రో ఐవించి సె		
		VIJAYAWADA		

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

BX:

3. MULTIPLICATION:

ASSUME CS: CO DATA SEGMEN	·			
OPR1 DW 1111H				
OPR2 DW 1111H	[
RES1 DW 1 DUP	(0H)			
RES2 DW 1 DUP	(0H)			
DATA ENDS				
CODE SEGMEN	NT			
START:				
MOV AX, DAT	Ϋ́Α			
MOV DS, AX				
MOV AX, OPR				
MOV BX, OPR				
MOV DX,0000	H			
MUL BX	SID	DHARTHA INSTITUS		
MOV RES1, A	X	m		
MOV RES2, D	X			
INT 03H				
CODE ENDS	>	ECHNO		
END START	ASAD			
END	RAS	/ א מ		
ADDRESS	OPCODE 💿	MNEMONIC OP	ERAND	COMMENTS
		ఆన బ్రీఫోతినన		
		ind the		
		VIJAYAWADA		

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

BX:

ASSUME CS: CC DATA SEGMEN OPR1 DW 1111H OPR2 DW 8888H RES1 DW 1 DUP RES2 DW 1 DUP	NT [[(0H)			
DATA ENDS				
CODE SEGMEN START:	N'I'			
MOV AX, DAT	`A			
MOV DS, AX				
MOV AX, OPR				
MOV BX, OPF				
MOV DX,000 IMUL BX	JH			
MOV RES1, A	X			
MOV RES2, D	7,	DHARTHA INS 7/7	Ur	
INT 03H	5		log n.	
CODE ENDS END START	POT	(1)		
END START END	AD V.		CHNO	
ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	Ē		~	
		. ఆత్య బిపాభన		
	(//	
		VIJAYAWADA		
Result: INPUT:				

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

BX:

DX:

ASSUME CS: CODE, DS: DATA DATA SEGMENT OPR1 DW 2224H OPR2 DW 1111H RES1 DW 1 DUP (0H) RES2 DW 1 DUP (0H) DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV AX, OPR1 MOV DS, AX MOV AX, OPR1 MOV BX, OPR2 MOV DX,00H DIV BX MOV RES1, AX MOV RES1, AX MOV RES2, DX INT 03H CODE ENDS END START					
DATA SEGMENT OPRI DW 2224H OPR2 DW 1111H RES1 DW 1 DUP (0H) RES2 DW 1 DUP (0H) DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV AX, OPR1 MOV AX, OPR1 MOV BX, OPR2 MOV DX,00H DIV BX MOV RES1, AX MOV RES2, DX INT 03H CODE ENDS END START END ADDRESS OPCODE MINEMONIC OPERAND COMMENTS VUAYAWADA UVAYAWAD	4. DIVISION:				
OPRI DW 2224H OPR2 DW 1111H RES1 DW 1 DUP (0H) RES2 DW 1 DUP (0H) DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV AX, DATA MOV DS, AX MOV AX, OPR1 MOV BX, OPR2 MOV DX,00H DIV BX MOV RES1, AX MOV RES2, DX INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE VUJAYAWADA	ASSUME CS: CO	ODE, DS: DATA			
OPR2 DW 1111H RES1 DW 1 DUP (0H) RES2 DW 1 DUP (0H) DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV AX, DATA MOV AX, OPR1 MOV BX, OPR2 MOV DX,00H DIV BX MOV RES1, AX MOV RES2, DX INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MNEMONIC OPERAND COMMENTS VUJAVAWADA	DATA SEGMEN	NT			
RESI DW 1 DUP (0H) RES2 DW 1 DUP (0H) DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV AX, DATA MOV AX, OPR1 MOV BS, AX MOV AX, OPR1 MOV BS, OPR2 MOV DX,00H DIV BX MOV RES1, AX MOV RES2, DX MOV RES2, DX MOV RES2, DX ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MNEMONIC OPERAND COMMENTS	OPR1 DW 2224H	1			
RES2 DW 1 DUP (0H) DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS, AX MOV AX, OPR1 MOV BX, OPR2 MOV DX,00H DIV BX MOV RES2, DX INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS UVUAYAWADA	OPR2 DW 1111H	I			
DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS, AX MOV AX, OPR1 MOV BX, OPR2 MOV DX,00H DIV BX MOV RES1, AX MOV RES2, DX INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS I I I I I I I I I I I I I I I I I I I	RES1 DW 1 DUP	• (0H)			
CODE SEGMENT START: MOV AX, DATA MOV DS, AX MOV DS, AX MOV AX, OPR1 MOV BX, OPR2 MOV DX,00H DIV BX MOV RES1, AX MOV RES2, DX INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MNEMONIC OPERAND COMMENTS	RES2 DW 1 DUP	'(0H)			
START: MOV AX, DATA MOV DS, AX MOV AX, OPR1 MOV BX, OPR2 MOV DX,00H DIV BX MOV RES1, AX MOV RES2, DX INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS VIANTAWADA	DATA ENDS				
MOV AX, DATA MOV DS, AX MOV DS, AX MOV AX, OPRI MOV BX, OPR2 MOV DX,00H DIV BX MOV RES1, AX MOV RES2, DX INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS UNATAWADA UNATAWADA	CODE SEGMEN	NT			
MOV DS, AX MOV AX, OPR1 MOV BX, OPR2 MOV DX,00H DIV BX MOV RES1, AX MOV RES2, DX INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS UVJAYAWADA	START:				
MOV AX, OPR1 MOV BX, OPR2 MOV DX,00H DIV BX MOV RES1, AX MOV RES2, DX INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS 	MOV AX, DAT	ΓA			
MOV BX, OPR2 MOV DX,00H DIV BX MOV RES1, AX MOV RES2, DX INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS 	MOV DS, AX				
MOV DX,00H DIV BX MOV RES1, AX MOV RES2, DX INT 03H CODE ENDS END START END ADDRESS OPCODE MINEMONIC OPERAND COMMENTS VIJAYAWADA	MOV AX, OPR	.1			
DIV BX MOV RES1, AX MOV RES2, DX INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE VIJAYAWADA	MOV BX, OPR	R2			
MOV RES2, DX INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MUAYAWADA Internet Internet Internet Inter	MOV DX,00H				
MOV RES2, DX INT 03H INT 03H OPCODE ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MNEMONIC OPERAND COMMENTS VUAYAWADA Internet Internet Internet Inter Inter <td< td=""><td>DIV BX</td><th>~</th><td></td><td></td><td></td></td<>	DIV BX	~			
INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MNEMONIC OPERAND COMMENTS	MOV RES1, A	X SIL			
INT OSIT CODE END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS OPCODE MNEMONIC OPERAND COMMENTS OPCODE MNEMONIC OPERAND COMMENTS OPCODE MNEMONIC OPERAND COMMENTS OPERAND COMENTS OPERAND	MOV RES2, D	X S		n' O	
END START Second se	INT 03H	Ë	(n)		
ADDRESS OPCODE MNEMONIC OPERAND COMMENTS	CODE ENDS	De la	(Chal	ПО	
ADDRESS OPCODE MNEMONIC OPERAND COMMENTS	END START			IZ	
ADDRESS OPCODE MNEMONIC OPERAND COMMENTS	END	SA	11, 251		
	ADDRESS	OPCODE	MNEMONIC		COMMENTS
		9		ð	
			. ఆత్త బీపేశిభన		
		()		//	
			VIJAYAWADA		
		1	1		

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

BX:

ASSUME CS: CO	DDE, DS: DATA			
DATA SEGME				
OPR1 DW 2224H	I			
OPR2 DW 1111H	I			
RES1 DW 1 DUP	P (0H)			
RES2 DW 1 DUP	P (0H)			
DATA ENDS				
CODE SEGMEN	NT			
START:				
MOV AX, DAT	ΓA			
MOV DS, AX				
MOV AX, OPR				
MOV BX, OPF	R2			
MOV DX,00H IDIV BX	l	DHARTHA INS TIT		
MOV RES1, A	v all	DHARTHA INS 7/7		
MOV RES1, A MOV RES2, D	x		In	
INT 03H			97	
CODE ENDS	2			
END START	>	IIInol	I I I I I I I I I I I I I I I I I I I	
END	ASAD		TECHNOLO	
ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
ADDRESS	OTCODE	IVIT VELIVIOI VIC	OILMIND	COMMENTS
		ఆత్యటిపోభవ		
			//	
		VIJATAWADA		
	1			1

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

BX:

ASCII OPERATIONS

1. AAA:

ASSUME CS: CODE CODE SEGMENT START: MOV AL, 35H MOV BL,39H MOV AH,00H ADD AL,BL AAA INT 03H CODE ENDS END START END

DHARTHA INSTIN,

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	771		OF	
	РО		TE	
		11 nor	СНІ	
	SAD		NOL	
	A SAS	/ / / /	00	
	<u>د</u>	Ŕ	Y	
		105 6 18 dd		
		00,000000		
		VIJAYAWADA 🦼		

Result: INPUT: AL: OUTPUT: AL: FLAG STATUS: Theoretical Calculations:

BL:

AH:

2. AAS:

ASSUME CS: CODE CODE SEGMENT START: MOV AL, 39H MOV BL,35H MOV AH,00H SUB AL,BL AAS INT 03H CODE ENDS END START END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	SIL		Ur,	
	LU.		0	
	10			
	К. Р	IIha	ECF	
	AD .	0	INC	
	AS/	7511	DLO	
	PR		GY	
	(Ç)			
		ఆత్తపిపోభన		
		VIJAYAWADA		

Result: INPUT: AL: OUTPUT: AL: FLAG STATUS: Theoretical Calculations:

BL:

AH:

3. AAM:

ASSUME CS: CODE CODE SEGMENT START: MOV AL, 05H MOV BL,09H MOV AH,00H MUL BL AAM INT 03H CODE ENDS END START END

DHARTHA INSTIN

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	771		97	
	РО		1 T	
	V.	n by	CH	
	SAD		101	
	SAS	/ / / / /	00	
	L S	Ê) i i i i i i i i i i i i i i i i i i i	
	1			
		అత్యబించిం	//	
		VIJAYAWADA		

Result:	
INPUT:	
AL:	
OUTPUT:	
AL:	
FLAG STATUS:	

BL:

AH:

4. AAD:

ASSUME CS: CODE CODE SEGMENT START: MOV AL, 05H MOV BL,06H MOV AH,03H AAD DIV BL INT 03H CODE ENDS END START END

DHARTHA INS TIT

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	771		OF	
	РО		TE	
		11 nor	СН	
	SAD		NOI	
	SAS	/]	_00	
	۵ ۵	¢	Y	
	~		-	
		ಆಡ್ಡಬಿಕ್ಕಾಭಾ		
		VIJAYAWADA		

Result:		
INPUT:		
AL:	AH:	
OUTPUT:		
AL:		
FLAG STATUS:		

Result: Arithmetic operation –Signed and unsigned Arithmetic operation, ASCII – arithmetic operations were performed.

AH:

BL:

EXPERIMENT-02

Arithmetic operations – Multi byte Addition and Subtraction,

AIM : To perform multibyte addition, subtraction, sum of squares and sum of cubes using TASM.

Experimental Requirements : PC loaded with TASM software

Procedure:

- 1. Switch on the PC, press windows+R then enter CMD.
- 2. Find the folder where TASM is located. check whether TASM.EXE, TLINK.EXE, TD.EXE are present or not
- 3. Enter into the directory where TASM is located by using cd... or directory name:
- 4. Type cd tasm in which the three files are present .Now we will be getting into c: \ or d:\ with tasm directory.
- 5. Type edit then a new window will be opened in which the program is entered.
- 6. After entering the program save the file with <filename.asm>.
- 7. Check for the errors or warnings by using TASM <filename> and press enter...
- 8. If there are no errors, then type TLINK <filename> to compile the file. If errors go back to the edit and do the necessary corrections and repeat the previous step.
- 9. Next type td <filename > to debug the executable file then will be getting the message program has no symbol table, press ok and then write down the instructions, registers and flags status before execution .
- 10. For step by step execution press F8.and for direct execution press F9 and then write down the instructions, registers and flags status after execution .Go to dump if required for noting down the required inputs and outputs.



1.MULTI BYTE ADDITION

ASSUME CS: CODE, DS: DATA DATA SEGMENT OPR1 DB 12H, 34H, 56H, 78H OPR2 DB 23H, 34H, 66H, 86H RES DW 1 DUP (0H) DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS, AX MOV SI, OFFSET OPR1 DI, OFFSET OPR2 MOV MOV BX, OFFSET RES MOV CX, 0004H MOV AH, 00H BACK: MOV AL, [SI] MOV DL, [DI] ADC AL, DL POTL MOV [BX], AL INC SI PRASAD V. INC DI INC BX LOOP BACK INT 03H CODE ENDS END START END



ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	SIL	DHARTHA INS 7/7	U.S.	
	5		IN OF	
	LOd			
	>	Hhb	9	
	SAD			
	RA	7	OG	
	¢			
		ఆత్రదీపిశిభవ		
		VIJAYAWADA		
		VIJAYAWADA		

Result: INPUT: SI: OUTPUT: RES: FLAG STATUS: Theoretical Calculations:

DI:

2. MULTI BYTE SUBTRACTION

ASSUME CS: CODE, DS: DATA DATA SEGMENT OPR1 DB 23H, 34H, 66H, 86H OPR2 DB 12H, 34H, 56H, 78H RES DW 1 DUP (0H) DATA ENDS CODE SEGMENT START:MOV AX, DATA MOV DS, AX MOV SI, OFFSET OPR1 DI, OFFSET OPR2 MOV MOV BX, OFFSET RES MOV CX, 0004H MOV AH, 00H BACK: MOV AL, [SI] MOV DL, [DI] AL, DL SBB POTL MOV [BX], AL INC SI PRASAD V. INC DI INC BX LOOP BACK INT 03H CODE ENDS END START END



ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	710	DHARTHA INS 7/3		
	8- 1		TTT I I I I I I I I I I I I I I I I I I	
	772		9	
	0d	(dlba)	TEC	
			HZ	
	RASA	1 791	6	
	L L L L L L L L L L L L L L L L L L L	(' ¢	ADGY A	
	<i>\$</i>	ఆశ్రవీపోచన	9	
		80,00-00))	
		VIJAYAWADA		

Result: INPUT: SI: OUTPUT: RES: FLAG STATUS: Theoretical Calculations:

DI:

Result: Multibyte addition, subtraction has been performed using TASM.

EXPERIMENT-03

Logic operations - Shift and rotate - Sum of Squares, Sum of Cubes

AIM: To perform logical operations on 16-bit using TASM.

Experimental Requirements: PC loaded with TASM software

Procedure:

- 1. Switch on the PC, press windows+R then enter CMD.
- 2. Find the folder where TASM is located. check whether TASM.EXE, TLINK.EXE, TD.EXE are present or not
- 3. Enter into the directory where TASM is located by using cd... or directory name:
- Type cd tasm in which the three files are present .Now we will be getting into c: \ or d:\ with tasm directory.
- 5. Type edit then a new window will be opened in which the program is entered.
- 6. After entering the program save the file with <filename.asm>.
- 7. Check for the errors or warnings by using TASM <filename> and press enter...
- 8. If there are no errors, then type TLINK <filename> to compile the file. If errors go back to the edit and do the necessary corrections and repeat the previous step.
- 9. Next type td <filename > to debug the executable file then will be getting the message program has no symbol table, press ok and then write down the instructions, registers and flags status before execution.
- 10. For step by step execution press F8.and for direct execution press F9 and then write down the instructions, registers and flags status after execution .Go to dump if required for noting down the required inputs and outputs.

Logical Instructions:

1.AND:

ASSUME CS: CODE CODE SEGMENT START: MOV AX, 3355H MOV BX, 5355H AND AX, BX INT 03H CODE ENDS END START END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
		HARTHA INS 7		
	SIL		Un	
	LU,		0	
	10			
		ha	I C	
	Q	0	UN-	
	4S/	751		
	L L L L L L L L L L L L L L L L L L L		GΥ	
	\$		8	
		ఆత్త బిపోభన		
			//	
		VIJAYAWADA		

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

BX:

2. **OR:**

ASSUME CS: CODE CODE SEGMENT START: MOV AX, 3355H MOV BX, 5355H OR AX, BX INT 03H CODE ENDS END START END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	SIL	DHARTHA INS TIJ	U~	
	Ŝ		m	
	1	$\left(\begin{array}{c} \end{array} \right)$	OF	
	PC		TE	
	>	nø	H	
	SAL		OL	
	RA	/]	OG	
	Ê ∥ ¢		\prec	
		CO.5 5-78-55		
		va we we	//	
		VIJAYAWADA 🏒		
Result:				

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

BX:

3. NOT:

ASSUME CS: CODE CODE SEGMENT START: MOV AX, 3355H NOT AX INT 03H CODE ENDS END START END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	SIL	DHARTHA INS 7/7	U ~	
	5		m	
	E.	$\left(\begin{array}{c} \end{array} \right)$	T	
	6			
	2		Ξ	
	SA	1195 T	2	
	RA		G	
	- \$		~	
		45 9:38:55		
Result:		- Calmarka		1
INPUT:		VIJAYAWADA	//	
AX·				

AX:

OUTPUT:

AX:

FLAG STATUS:

Theoretical Calculations:

4. XOR:

ASSUME CS: CODE CODE SEGMENT START: MOV AX, 3355H MOV BX, 5355H XOR AX, BX INT 03H CODE ENDS END START END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	SIL	DHARTHA INS TIJ		
	S		m	
	Z	$\left(\begin{array}{c} \end{array} \right)$	DT T	
	РО		TE	
	2	nø	H	
	SAI			
	RA	/ /	OG	
	¢ ا	(\sim	
		05 57855		
		100guo 40	//	
		VIJAYAWADA 🍃		

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

BX:

5.TEST:

ASSUME CS: CODE

CODE SEGMENT START: MOV AX, 3355H MOV BX, 5355H TEST AX, BX INT 03H CODE ENDS END START END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
		OHARTHA INS 7		
	SIL		Ur,	
	107		0	
	10			
	<u>с.</u> Р	ha	I C	
	Q	0	-INC	
	4S/	751		
	PR		GY	
	\$		8	
		ఆత్త బీఫోభన		
			//	
		VIJAYAWADA		

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

BX:

Shift and Rotate Instructions

1. SHR:

ASSUN	AE CS: CODE
CODE	SEGMENT
START	
MOV	AX, 0ABCDH
MOV	CL, 04H
SHR	AX, CL
INT	03H
CODE	ENDS
END	START
END	

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
		OHARTHA INS 71		
	SI		Ur,	
	5		0	
	5			
	d .	11ha	EC	
	9		I	
	SA	11.221		
	22		G	
	Ť Š		3	
		ఆత బీఫోచన		
			//	
	//	VIJAYAWADA		
Posult:			1	

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

CL:

2. SHL:

ASSUME CS: CODE CODE SEGMENT START: MOV AX, 0ABCDH MOV CL, 04H SHL AX, CL INT 03H CODE ENDS END START END

SIDDHARTHA INSTITUS

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	Ĕ	$\left(\begin{array}{c} 0 \end{array} \right)$	F	
	PC	(hal	EC	
			HZ	
	SA			
	RA		G	
			}	
		ఆత్రబ్రేశిభన		
	(//	
		VIJAYAWADA		

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

CL:

ASSUME CS: CODE CODE SEGMENT START: MOV AX, 0ABCDH MOV CL, 04H SAL AX, CL INT 03H CODE ENDS END START END

OPCODE	MNEMONIC	OPERAND	COMMENTS
	DHARTHA INS 7/7		
4			
Ec		-OF	
0			
	ha	CH	
AD		NO	
AS	/ \		
Ľ I A	e e e e e e e e e e e e e e e e e e e	× III	
		5	
	ASAD V. POTLUP	DHARTHA INS 7//	DHARTHA INS 7/7 OF TECHNOLOGY

පමෘඩනංගන VIJAYAWADA

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

CL:

3. ROTATE RIGHT:

ASSUME CS: CODE CODE SEGMENT START: MOV AX, 0ABCDH MOV CL, 04H STC ROR AX, CL INT 03H CODE ENDS END START END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	SIL	DHARTHA INS TI		
	Ŝ		m	
	1	$\left(\begin{array}{c} \end{array} \right)$	- FI	
	PO		TE	
	>	nø	H	
	SAL			
	RA	/]	OG	
	¢ ا	R R R R R R R R R R R R R R R R R R R	\prec	
		CO.5 5-78-55		
		va we we	//	
		VIJAYAWADA		
· · · · · · · · · · · · · · · · · · ·				

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

CL:

ASSUME CS: CODE
CODE SEGMENT
START:
MOV AX, 0ABCDH
MOV CL, 04H
CLC
ROR AX, CL
INT 03H
CODE ENDS
END START
END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	71.	DHARTHA INS 7/7		
	2-51			
	07		i og	
	100		H	
	Κ. Ε	IIIhA	ECH	
	AD		NO	
	AS	5	LO	
	PR		GY	
	9		5	
		. ఆత్మ బీ హా భవ		
			//	
		VIJATAWADA		

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

CL:

4. ROTATE LEFT:

ASSUME CS: CODE CODE SEGMENT START: MOV AX, 0ABCDH MOV CL, 04H STC ROL AX, CL INT 03H CODE ENDS END START END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	SIL	DHARTHA INS TI		
	Ŝ		m	
	1	$\left(\begin{array}{c} \end{array} \right)$	- FI	
	PO		TE	
	>	nø	H H	
	SAL			
	RA	/]	OG	
	Ê ∥ ¢		} <	
		CO.5 5-78055		
		va we we	//	
		VIJAYAWADA		
		VIJATAWADA		

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

CL:

ASSUME CS: CODE
CODE SEGMENT
START:
MOV AX, 0ABCDH
MOV CL, 04H
CLC
ROL AX, CL
INT 03H
CODE ENDS
END START
END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	710	DHARTHA INS 7/7	· .	
	2			
	2		OF	
	0			
	>	ha	ЮН	
	AD		NO	
	AS		LO	
	L'L A		GY	
	€r		5	
		. ఆత్య బిపో భవ		

VIJAYAWADA

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

CL:

5. ROTATE RIGHT THROUGH CARRY:

ASSUME CS: CODE CODE SEGMENT START: MOV AX, 0ABCDH MOV CL, 04H STC RCR AX, CL INT 03H CODE ENDS END START END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	SIL	DHARTHA INS TI	U.	
	S		m	
	1	$\left(\begin{array}{c} \end{array} \right)$	0F	
	PO		TE	
	>		H	
	SAI		IOL	
	RA	/]	OG	
	¢ ا	R R R R R R R R R R R R R R R R R R R	\rightarrow	
		CO.5 5 +8055		
		Noguo po	//	
		VIJAYAWADA 🏒		

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

CL:

ASSUME CS: CODE
CODE SEGMENT
START:
MOV AX, 0ABCDH
MOV CL, 04H
CLC
RCR AX, CL
INT 03H
CODE ENDS
END START
END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	710	DHARTHA INS 7/7	· .	
	2-5			
	07		97	
	0			
		here	СН	
	AD		NO	
	SAS		LOO	
	Ľ 🖌		YE	
	÷		5	
		. ఆత్య బిపోభన్		

VIJAYAWADA

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

CL:

6. ROTATE LEFT THROUGH CARRY:

ASSUME CS: CODE CODE SEGMENT START: MOV AX, 0ABCDH MOV CL, 04H STC RCL AX, CL INT 03H CODE ENDS END START END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	SIL	DHARTHA INS TI	(/~	
	Ŝ		m	
	1	$\left(\begin{array}{c} \end{array} \right)$	OF T	
	PO		TE	
	>	nø	HN	
	SAL			
	RA	/]	OG	
	Ê ∥ ¢		\rightarrow	
		CO.5 5-78055		
		va we we	//	
		VIJAYAWADA		

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

CL:

ASSUME CS: CODE
CODE SEGMENT
START:
MOV AX, 0ABCDH
MOV CL, 04H
CLC
RCL AX, CL
INT 03H
CODE ENDS
END START
END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	710	DHARTHA INS 7/7	· .	
	2			
	2		OF	
	0			
	>	ha	ЮН	
	AD		NO	
	AS		LO	
	L'A		GY	
	€r		5	
		. ఆత్య బిపో భవ		

VIJAYAWADA

Result: INPUT: AX: OUTPUT: AX: FLAG STATUS: Theoretical Calculations:

CL:

SUM OF SQUARES $(1^2 + 2^2 + 3^2 + ... + n^2)$

ASSUME CS:CODE CODE SEGMENT START: MOV CL,07H MOV DX,0000H MOV AH,00H L1: MOV AL,CL MUL CL ADD DX,AX LOOP L1 INT 03H CODE ENDS END START END

SIDDHARTHA INS TITU

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	110	$\langle \mathbf{u} \rangle$	FT	
	/. P		EC	
			INF	
	AS/	1 751	DLC	
	PR/		θGΥ	
	¢.			
		ఆత,బిపిశచన		
		0	//	
		VIJAYAWADA		

Result: INPUT: CL: OUTPUT: DX: FLAG STATUS: Theoretical Calculations:

SUM OF SQUARES IN AN ARRAY ASSUME CS:CODE,DS:DATA DATA SEGMENT ARRI DB 05H,07H,06H,04H DATA ENDS CODE SEGMENT START: MOV AX,DATA MOV DS,AX MOV SI,OFFSET ARRI MOV OX,0000H MOV AH,00H L1: MOV BL,[SI] MOV AL,BL MUL BL ADD DX,AX INC SI LOOP L1 INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADD RESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OP			.		
DATA SEGMENT ARRI DB 05H,07H,06H,04H DATA ENDS CODE SEGMENT START: MOV AX,DATA MOV DS,AX MOV SLOFFSET ARRI MOV CX,0004H MOV DX,0000H MOV AL,00H L1: MOV BL,[SI] MOV AL,BL MUL BL ADD DX,AX INC SI LOOP L1 INT 03H CODE ENDS END START END CODE ENDS END START END CODE INT CODE I			Y		
ARRI DB 05H,07H,06H,04H DATA ENDS CODE SEGMENT START: MOV AX,DATA MOV DX,007 MOV AX,DATA MOV S,AX MOV S,007 MOV AL,00H L1: MOV BL,[SI] MOV AL,BL MUL BL ADD DX,AX INC SI LOOP L1 INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS CODE ENDS END START END CODE ENDS CODE I CODE I					
DATA ENDS CODE SEGMENT START: MOV AX,DATA MOV DS,AX MOV SI,OFFSET ARRI MOV CX,0004H MOV AL,00H L1: MOV BL,[SI] MOV AL,BL MUL BL ADD DX,AX INC SI INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS COMMENT					
CODE SEGMENT START: MOV AX,DATA MOV DS,AX MOV SI,OFFSET ARRI MOV CX,0004H MOV AL,000H L1: MOV BL,[SI] MOV AL,BL MUL BL ADD DX,AX CODE ENDS END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MNEMONIC OPERAND ADDRESS OPCODE MOMANDA Intro Intro Intro <		п,00п,04п			
START: MOV AX,DATA MOV DS,AX MOV SI,OFFSET ARRI MOV CX,0004H MOV AL,00H L1: MOV BL,[SI] MOV AL,BL ADD DX,AX INC SI LOOP L1 INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS INC SI LOOP L1 INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS INC SI INC S		-			
MOV DS,AX MOV SI,OFFSET ARRI MOV CX,0004H MOV DX,0000H MOV AL,00H L1: MOV BL,[SI] MOV AL,BL MUL BL ADD DX,AX INC SI LOOP L1 INT 03H CODE ENDS END ADDRESS OPCODE MOLARES COMMENTS ADDRESS OPCODE MOLARES COMMENTS ADDRESS OPCODE MOLARES COMMENTS ADDRESS OPCODE MOLARES Income Income Income					
MOV SI,OFFSET ARRI MOV CX,0004H MOV DX,0000H MOV AL,00H L1: MOV BL.[SI] MOV AL,BL MUL BL ADD DX,AX INC SI LOOP L1 INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MULANDA Income Income Income		,DITTY			
MOV CX,0004H MOV AL,00H L1: MOV BL.[SI] MOV AL,BL MUL BL ADD DX,AX INC SI LOOP L1 INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MOMANDA Intervention Intervention Intervention Intervention Intervention ADDRESS OPCODE MNEMONIC OPERAND COMMENTS Intervention Intervention Intervention Interventi		ARR1			
MOV DX,0000H MOV AH,00H L1: MOV BL,[SI] MOV AL,BL MUL BL ADD DX,AX INC SI LOOP L1 INT 03H CODE ENDS END START END ADDRESS OPCODE MILL ADDRESS OPCODE MILL MUL BL ADDRESS OPCODE MILL ADDRESS OPCODE MILL ADDRESS OPCODE MILL MILL ADDRESS OPCODE MILL ADDRESS OPCODE MILL Interve					
MOV AH,00H L1: MOV BL,[SI] MOV AL,BL MUL BL ADD DX,AX INC SI LOOP L1 INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MUL ADDRESS OPCODE MOMENDIC OPERAND COMMENTS Intervention ADDRESS OPCODE MUL ADDRESS OPCODE MUL ADDRESS OPCODE Intervention Intervention Intervention Intervention <t< td=""><td></td><td></td><td></td><td></td><td></td></t<>					
MOV AL,BL MUL BL ADD DX,AX INC SI LOOP L1 INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MNEMONIC OPERAND COMMENTS MUL HL OPERAND ADDRESS OPCODE MNEMONIC OPERAND COMMENTS OPERAND ADDRESS OPCODE MUL HL OPERAND ADDRESS OPCODE MIATAWADA Interve Interve Interve					
MUL BL ADD DX,AX INC SI LOOP L1 INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC ADDRESS OPCODE MNEMONIC ADDRESS OPCODE	L1: MOV BL,[SI]				
ADD DX,AX INC SI LOOP L1 INT 03H CODE ENDS END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MNEMONIC OPERAND COMMENTS	MOV AL,BL				
INT OSH Interview	MUL BL				
INT OSH Interview	ADD DX,AX		NHARTHA INS		
INT OSH Interview		SI			
INT OSH Interview		S.		0	
END START Normalize Person Comments ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ADDRESS OPCODE MNEMONIC OPERAND COMMENTS Image:		Fo			
ADDRESS OPCODE MNEMONIC OPERAND COMMENTS Image: Comment of the second sec			Inhal		
ADDRESS OPCODE MNEMONIC OPERAND COMMENTS Image: Comment of the second sec					
Comparison Comp		S			COMMENTS
	ADDRESS	OPCODE	MINEMONIC	OPERAND	COMMENTS
		(<u>)</u>		<u>}</u>	
		11	ఆత్మబిపోభన		
				//	
			VIJATAVVADA		
Image: Constraint of the second se					
Image: Constraint of the second se					
Image: Constraint of the second se					<u> </u>
Image: Constraint of the second sec			1 1		1
Image: Constraint of the second se					
Result:					
Result:					
Result:					

INPUT: ARR1: OUTPUT: DX: FLAG STATUS: Theoretical Calculations:

SUM OF CUBES $(1^3 + 2^3 + 3^3 + ... + n^3)$

ASSUME CS:CODE CODE SEGMENT START: MOV CL,07H MOV DX,0000H MOV AH,00H L1: MOV AL,CL MUL CL MUL CL ADD DX,AX LOOP L1 INT 03H CODE ENDS END START END

SIDDHARTHA INS TITU

	<u> </u>			1 1
ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	AD		ION	
	SAS	/ \	_00	
	L PI	e e e e e e e e e e e e e e e e e e e	Υ Έ	
	(F)			
		ಆತ್ಮಐರ್ಭಾಂ		
		VIJAYAWADA		

Result: INPUT: CL: OUTPUT: DX: FLAG STATUS: Theoretical Calculations:

ASSUME CS:CODE, DS: DATA DATA SEGMENT ARR1 DB 05H,07H,06H,04H DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS,AX MOV SI, OFFSET ARR1 MOV CX,0004H MOV DX,0000H MOV AH,00H L1: MOV BL,[SI] MOV AL, BL MUL BL SIDDHARTHA INS TIT MUL BL ADD DX,AX INC SI POTL LOOP L1 **TECHNO** INT 03H > AD CODE ENDS LOG END START END ADDRESS OPCODE MNEMONIC OPERAND COMMENTS ఆత.బీఫి•చవ L VIJAYAWADA

Result: INPUT: ARR1: OUTPUT: DX: FLAG STATUS: Theoretical Calculations

Result: Logic operations – Shift and rotate – sum of squares and sum of cubes using TASM were performed.



EXPERIMENT-4

Smallest, largest number, arrange numbers in Ascending order, Descending order

AIM: To find smallest, largest number, arrange numbers in ascending order, descending order in a given series.

Experimental Requirements: PC loaded with TASM software

Procedure:

- 1. Switch on the PC, press windows+R then enter CMD.
- 2. Find the folder where TASM is located. check whether TASM.EXE, TLINK.EXE, TD.EXE are present or not
- 3. Enter into the directory where TASM is located by using cd... or directory name:
- 4. Type cd tasm in which the three files are present .Now we will be getting into c: \ or d:\ with tasm directory.
- 5. Type edit then a new window will be opened in which the program is entered.
- 6. After entering the program save the file with <filename.asm>.
- 7. Check for the errors or warnings by using TASM <filename> and press enter...
- 8. If there are no errors, then type TLINK <filename> to compile the file. If errors go back to the edit and do the necessary corrections and repeat the previous step.
- 9. Next type td <filename > to debug the executable file then will be getting the message program has no symbol table, press ok and then write down the instructions, registers and flags status before execution.
- 10. For step by step execution press F8.and for direct execution press F9 and then write down the instructions, registers and flags status after execution .Go to dump if required for noting down the required inputs and outputs.

Smallest number

ASSUME CS: CODE, DS: DATA DATA SEGMENT LIST DB 35H, 26H, 19H, 56H, 44H DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS, AX

MOV CX, 0004H
MOV SI, OFFSET LIST
MOV BL, [SI]
L2: MOV AL, [SI+1]
CMP BL, AL
JB L1
MOV BL, AL
L1: INC SI
LOOP L2
INT 03H
CODE ENDS
END START
END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
		NHARTHA INS 7		
	SI		UX.	
	5		·m	
	11		F	
	P			
	>	h h b	H L L	
	AD		Z	
	4S.		6	
	2K		G	
	\$			
		105 5-48-55		
		200 wo		
			//	

Result: INPUT: LIST: OUTPUT: BL: FLAG STATUS: Theoretical Calculations

Largest number

ASSUME CS: CODE, DS: DATA DATA SEGMENT LIST DB 35H, 26H, 19H, 56H, 44H DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS, AX MOV CX, 0004H MOV SI, OFFSET LIST MOV BL, [SI] L2: MOV AL, [SI+1] CMP BL, AL JA L1 MOV BL, AL SIDDHARTHA INS TIT L1: INC SI LOOP L2 9 . POTLUN INT 03H CODE ENDS П END START > END **MNEMONIC** OPERAND **COMMENTS** ADDRESS OPCODE 2 G L. ń ಆತ್ಮಬಿತ್ ಭಿತ್ರ . **VIJAYAWAD**

Result: INPUT: LIST: OUTPUT: BL: FLAG STATUS: Theoretical Calculations:

Ascending order

ASSUME CS: CODE, DS: DATA DATA SEGMENT STR1 DB 'BINDHU\$' DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS, AX MOV DX, 0005H L3: MOV CX, DX MOV SI, OFFSET STR1 L2: MOV AL, [SI] CMP AL, [SI+1] JB L1 XCHG AL, [SI+1] XCHG AL, [SI] L1: INC SI LOOP L2 DEC DX JNZ L3 INT 03H CODE ENDS END START END



ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	all	DHARTHA INS 7/7		
	5			
	H H			
	2		I III	
	2		Hz	
Result:	PRASAI	' ' ^y En		
INPUT:	AN I		OLOGY	
STR1:	Ť I Ø		» [~	
OUTPUT:		<i>ఆత్మ</i> బీపోభవ		
STR1: FLAG STATUS:			//	
Theoretical Calcula	tions	VIJAYAWADA		

Descending order

ASSUME CS: C	CODE, DS: DATA				
DATA SEGME					
STR1 DB 'BI					
DATA ENDS					
CODE SEGME	NT				
START: MOV	AX, DATA				
MOV DS, AX					
MOV DX, 0003	5H				
L3: MOV CX, I					
MOV SI, OFFS					
L2: MOV AL, [S					
CMP AL, [SI+1					
JA L1					
XCHG AL, [SI-	-1]				
XCHG AL, [SI]					
L1: INC SI	GID	DHARTHA INS TIT			
LOOP L2	2				
DEC DX	77		9		
JNZ L3	6		N <u>≓</u>		
INT 03H		IIIhal			
CODE ENDS	<u>a</u>				
	\leq				
END START	S				
END START END	RAS	5	TECHNOLOG		
	OPCODE Ø	DHARTHA INS 7/7 MNEMONIC	OPERAND	COMMENTS	
END	OPCODE 🚳	and Side it	OPERAND	COMMENTS	
END	OPCODE Ø	MNEMONIC &	OPERAND	COMMENTS	
END	OPCODE Ø	and Side it	OPERAND	COMMENTS	
END	OPCODE	and Side it	OPERAND	COMMENTS	
END	OPCODE Ø	and Side it	OPERAND	COMMENTS	
END	OPCODE	and Side it	OPERAND	COMMENTS	
END	OPCODE Ø	and Side it	OPERAND	COMMENTS	
END	OPCODE	and Side it	OPERAND	COMMENTS	
END	OPCODE Ø	and Side it	OPERAND		
END	OPCODE	and Side it	OPERAND		
END	OPCODE Ø	and Side it	OPERAND		
END	OPCODE	and Side it	OPERAND		
END	OPCODE Ø	and Side it	OPERAND		
END	OPCODE	and Side it	OPERAND		
END	OPCODE Ø	and Side it	OPERAND OPERAND		
END	OPCODE	and Side it	OPERAND Image: Comparison of the second se		
END	OPCODE Ø	and Side it	OPERAND Image: Comparison of the second se		
END	OPCODE	and Side it	OPERAND		
END	OPCODE Ø	and Side it	OPERAND OPERAND OPERAND		

Result: INPUT: STR1: OUTPUT: STR1: FLAG STATUS: Theoretical Calculations

<u>RESULT</u>: Finding the smallest, largest numbers and arranging given numbers in ascending and descending orders using TASM are performed.



EXPERIMENT-5

STRING OPERATIONS

Aim : String operation and Instruction prefix: Move Block, Reverse string, Inserting, Deleting, Length of the string, String comparison.

Experimental Requirements : PC loaded with TASM software

Procedure:

- 1. Switch on the PC, press windows+R then enter CMD.
- 2. Find the folder where TASM is located. check whether TASM.EXE, TLINK.EXE, TD.EXE are present or not
- 3. Enter into the directory where TASM is located by using cd... or directory name:
- 4. Type cd tasm in which the three files are present .Now we will be getting into c: \ or d:\ with tasm directory.
- 5. Type edit then a new window will be opened in which the program is entered.
- 6. After entering the program save the file with <filename.asm>.
- 7. Check for the errors or warnings by using TASM <filename> and press enter...
- 8. If there are no errors, then type TLINK <filename> to compile the file. If errors go back to the edit and do the necessary corrections and repeat the previous step.
- 9. Next type td <filename > to debug the executable file then will be getting the message program has no symbol table, press ok and then write down the instructions, registers and flags status before execution.
- 10. For step by step execution press F8.and for direct execution press F9 and then write down the instructions, registers and flags status after execution .Go to dump if required for noting down the required inputs and outputs.

STRING OPERATIONS

1. MOVING A BLOCK OF DATA

ASSUME CS: CODE, DS: DATA, ES:EXTRA DATA SEGMENT ORG 1000H STR1 DB 'HI FRIEND\$' COUNT EQU \$-STR1 DATA ENDS EXTRA SEGMENT **ORG 2000H** STR2 DB 1 DUP(?) EXTRA ENDS CODE SEGMENT START: MOV AX,DATA MOV DS,AX MOV AX,EXTRA MOV ES.AX MOV SI, OFFSET STR1 PO MOV DI,OFFSET STR2 PRASAD V. MOV CL,COUNT-1 REP MOVSB INT 03H CODE ENDS Ô END START END



ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	GIL	DHARTHA INS 7/7		
	2			
	2		9	
	6			
	2	INLAL	E	
	6		H.	
	SA		2	
		/ / /	8	
	He la	d d	Y	
	9		²	

Result: INPUT: STR1: OUTPUT: STR2: FLAG STATUS: Theoretical Calculations



2. REVERSE OF A STRING

ASSUME CS: CODE, DS: DATA DATA SEGMENT **ORG** 1000H STR1 DB 'HI FRIEND\$' COUNT EQU \$-STR1 DATA ENDS CODE SEGMENT START: MOV AX, DATA MOV DS, AX MOV SI, OFFSET STR1 MOV DI, OFFSET STR1+COUNT-2 MOV CL, COUNT/2 BACK: MOV AL,[SI] XCHG [DI], AL XCHG [SI], AL INC SI DEC DI PO7L LOOP BACK INT 03H PRASAD V. CODE ENDS **END START** END



ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	15	DHARTHA INS 7/7		
	3			
	071		<u> </u>	
	6			
	> 0	hhột	H	
	SAI		101	
	PRAS	/]	-0GY	
	L Ø			
		ఆత్రబిపోభన		
			//	
		VIJAYAWADA		

Result: INPUT: STR1: OUTPUT: STR1: FLAG STATUS: Theoretical Calculations

3. STRING COMPARISON

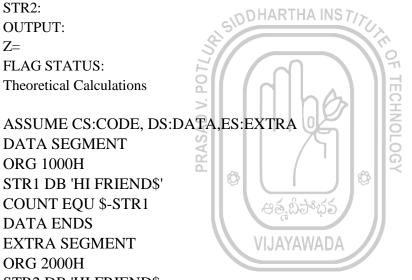
ASSUME CS:CODE, DS:DATA, ES:EXTRA DATA SEGMENT ORG 1000H STR1 DB 'HI FRIEND\$' COUNT EQU \$-STR1 DATA ENDS EXTRA SEGMENT ORG 2000H STR2 DB 'HIFRIEND' EXTRA ENDS CODE SEGMENT START: MOV AX,DATA MOV DS,AX MOV AX,EXTRA MOV ES,AX MOV SI, OFFSET STR1 MOV DI, OFFSET STR2 P01/ MOV CL, COUNT-1 REP CMPSB PRASAD V. INT 03H CODE ENDS END START END

SID	DHARTHA INS TITU	
5		OF TI
		ECHNO
ð l	T a	HNOLOGY
~	CO 5 8-38-55	

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
		VIDATAWADA		

Result: **INPUT:** STR1: STR2: OUTPUT: Z= FLAG STATUS: **Theoretical Calculations**

DATA SEGMENT ORG 1000H STR1 DB 'HI FRIEND\$' COUNT EQU \$-STR1 DATA ENDS EXTRA SEGMENT **ORG 2000H STR2 DB 'HI FRIEND\$** EXTRA ENDS CODE SEGMENT START: MOV AX,DATA MOV DS,AX MOV AX,EXTRA MOV ES,AX MOV SI, OFFSET STR1 MOV **DI,OFFSET STR2** MOV CL, COUNT-1 **REP CMPSB** INT 03H CODE ENDS END START END



ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	ر آلام	HARTHA INS ראד		
	25		1	
			0	
	Dod		H	
		Hhb	CH	
	N N N N N N N N N N N N N N N N N N N	11951		
	V V	/ / / /	Ó	
	Ê 🔅		GY	
	~	.ఆత్రబీఫిశిభన		
		00,00 m))	
		VIJAYAWADA 🍃	/	

Result: INPUT: STR1: STR2: OUTPUT: Z= FLAG STATUS: Theoretical Calculations:

Result: String operation and Instruction prefix: Move Block, Reverse string and String comparison were performed.

Introduction to MSP430 launch pad and Programming Environment. (Study Experiment)

Aim: To write an assembly language program to blink an LED

Experiment Requirements: PC loaded code composer studio, MSP430 LAUNCHPAD

Procedure:

- 1. Open code composer studio
- 2. Open file go to new and select CCS project
- 3. A CCS window opens.

Select MSP430G2253 in the target. ST/>

Establish the connection by selecting the TI MSP430 USB1 (default)

Give a project name.

Select empty project with main.c and press finish.

- 4. Write C code in main.c.
- 5. Select build project and build your program. It will check for errors. When it is error free go to next step, otherwise repeat until the program is error free.

ఆత.బీపోచవ

6. Go to target configuration

Select user defined VIJAYAWADA

Select new target

Select MSP430G2253 in the target.

Establish the connection by selecting the TI MSP430 USBI

Save.

7. Open New Target configuration

Right click on new target configuration

Click on Launch selected configuration

Select MSP430G2253 in the target.

Establish the connection by selecting the TI MSP430 USB1 (default) and Save.

8. Open the Run menu

Select connect project

Again open Run

Load Project

Browse the project

select the project.out

select, save and ok.

- 9. Run the program.
- 10. Observe the output in the console window or on the board.

```
THARTHA INS T
Program:
#include <msp430.h>
void main(void) {
  WDTCTL = WDTPW | WDTHOLD;
                                          Stop watchdog timer
                           PRAS
                               // P1.0 (Red LED), P1.1 (Green LED)
  P1DIR \models (BIT0+BIT6);
                                Ô
  while(1)
  {
                                     පෂ හිත් ස
    volatile unsigned long i;
                                    VIJAYAWADA
    P1OUT &= ~BIT6;
                              //Green LED -> OFF
    P1OUT \models BIT0;
                            //Red LED -> ON
    for(i = 0; i<10000; i++); //delay
    P1OUT &= ~BIT0;
                              //Red LED -> OFF
    P1OUT \models BIT6;
                            //Green LED -> ON
    for(i = 0; i<10000; i++); //delay
  }
}
```

Result: Blinking of LED on the MSP430 launch pad was performed.

Read input from switch and Automatic control/flash LED (soft-ware delay).

Aim: To read input from switch and Automatic control/flash LED (soft-ware delay).

Experiment Requirements: PC loaded code composer studio, MSP430 LAUNCHPAD Procedure:

- 1. Open code composer studio
- 2. Open file go to new and select CCS project
- 3. A CCS window opens.

Select MSP430G2253 in the target.

Establish the connection by selecting the TI MSP430 USB1 (default) Give a project name.

Select empty project with main.c and press finish.

- 4. Write C code in main.c.
- 5. Select build project and build your program. It will check for errors. When it is error free go to next step, otherwise repeat until the program is error free.
- 6. Go to target configuration

Select user defined

Select new target

Select MSP430G2253 in the target.

Establish the connection by selecting the TI MSP430 USBI

Save.

7. Open New Target configuration

Right click on new target configuration

Click on Launch selected configuration

Select MSP430G2253 in the target.

Establish the connection by selecting the TI MSP430 USB1 (default) and Save.

8. Open the Run menu

Select connect project

Again open Run

Load Project

Browse the project

select the project.out

select, save and ok.

- 9. Run the program.
- 10. Observe the output in the console window or on the board.

```
Program:
```

```
#include <msp430.h>
```

```
DHARTHA INS T
/*
* main.c
*/
void main(void)
{
    WDTCTL = WDTPW + WDTHOLD;
                                                     Stop watchdog timer
                                                           .OGY
    P1DIR = 0x01; // Set P1,0 to output direction
                                                      Ô
    P1OUT \models BIT3;
    P1REN \models BIT3;
                                       ఆత్తపిపోభక
    while (1) // Infinite Loop
     {
    if ((BIT3 & P1IN)) // active low switch
    P1OUT &= ~0x01; // if P1.3 is 1(not pressed), reset P1.0
     }else
    P1OUT |= 0x01; // else set P1.0
     }
     }
     }
```

Result: Reading input from switch and Automatic control/flash LED (soft-ware delay) has been performed.

Read Temperature of MSP430 with the help of ADC.

<u>AIM</u>: To Read Temperature of MSP430 with the help of ADC.

Experiment Requirements: PC loaded code composer studio, MSP430 LAUNCHPAD

Procedure:

- 1. Open code composer studio
- 2. Open file go to new and select CCS project
- 3. A CCS window opens.

Select MSP430G2253 in the target.

Establish the connection by selecting the TI MSP430 USB1 (default)

Give a project name.

Select empty project with main.c and press finish.

- 4. Write C code in main.c.
- 5. Select build project and build your program. It will check for errors. When it is error free go to next step, otherwise repeat until the program is error free.

OG

Ô

6. Go to target configuration

Select user defined

Select new target

Select MSP430G2253 in the target.

Establish the connection by selecting the TI MSP430 USBI

Save.

7. Open New Target configuration

Right click on new target configuration

Click on Launch selected configuration

Select MSP430G2253 in the target.

Establish the connection by selecting the TI MSP430 USB1 (default)

and Save.

Open the Run menu

Select connect project

Again open Run

Load Project

Browse the project

select the project.out

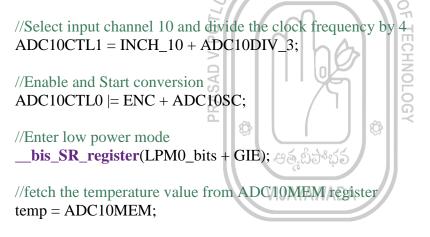
select, save and ok.

- 8. Run the program.
- 9. Observe the output in the console window or on the board.

program: #include <msp430g2353.h>

int temp = 0; int main(void){ WDTCTL = WDTPW | WDTHOLD; //stop the watchdog timer

//Select 1.5 V, 64 clock cycles, enable ADC interrupt, Turn on the reference generator ADC10CTL0 = SREF_1 + REFON + ADC10ON + ADC10SHT_3 + ADC10IE;



//convert it into degree <u>celsius</u> temp = ((temp * 27069L - 18169625L)>>16);

return 0;

}

```
//ISR
#pragma vector = ADC10_VECTOR
__interrupt void adc_interrupt(void)
```

```
{
__bic_SR_register_on_exit(CPUOFF);
}
```

<u>RESULT</u>: Hence read Temperature of MSP430 with the help of ADC

Interrupts Programming Example Using GPIO

AIM: To perform Interrupts Programming Example Using GPIO.

Experiment Requirements: PC loaded code composer studio, MSP430 LAUNCHPAD

PROCEDURE:

- 1. Open code composer studio
- 2. Open file go to new and select CCS project
- 3. A CCS window opens.

Select MSP430G2253 in the target.

Establish the connection by selecting the TI MSP430 USBI

Give a project name. DHARTHA INSTIT

Select empty project with main.c and finish.

- 4. Write C code in main.c.
- 5. Select build project and build your program.
- 6. Go to target configuration

Select user define

Select new target

Select MSP430G2253 in the target.

Establish the connection by selecting the TI MSP430 USBI Save.

TECHNOLOG

Ô

7. Open New Target configuration

Right click on new target configuration

Click on Launch selected configuration

Select MSP430G2253 in the target.

Establish the connection by selecting the TI MSP430 USBI

Save.

8. Open the Run menu

Select connect project

Again open Run

Load Project

Browse the project

select the project.out

select, save and ok.

- 9. Run the program.
- 10. Observe the output in the console window or on the board.

Program 199

```
#include <msp430g2353.h>
unsigned int wdtCounter = 0;
void main(void)
{
       WDTCTL = WDT_MDLY_32; // Set <u>Watchdog</u> Timer interval to ~32ms
       IE1 |= WDTIE; // Enable WDT interrupt
P1DIR |= BIT0; // Set P1.0 to output direction
       P1OUT |= BIT0; // Turn on LED at 1.0
       P1IE |= BIT3; // enable P1.3 interrupt
       __enable_interrupt();
                                                             TECHNOLOG
       for(;;)
                              PRASAD
       {
       }
                                   Ô
}
// Watchdog Timer interrupt service routine
                                       VIJAYAW
#pragma vector=WDT_VECTOR
interrupt void watchdog timer(void)
{
       if(wdtCounter == 249)
               P1OUT = 0x00; // P1.0 turn off
               wdtCounter = 0;
               _BIS_SR(LPM3_bits + GIE); // Enter LPM3 w/interrupt enabled
       }
       else
       {
               wdtCounter++;
       }
}
```

<u>RESULT</u>: Hence performed Interrupts Programming Example Using GPIO

Use of Comparator to Compare the Signal Threshold Level

AIM: Use Of Comparator To Compare The Signal Threshold Level.

Experimental Requirements : PC loaded code composer studio, MSP430 LAUNCHPAD

Procedure:

- 1. Open code composer studio
- 2. Open file go to new and select CCS project
- 3. A CCS window opens.

Select MSP430G2253 in the target.

Establish the connection by selecting the TI MSP430 USBI Give a project name.

TECHNOLOG

Ô

Select empty project with main.c and finish.

4. Write C code in main.c.

5. Select build project and build your program.

6. Go to target configuration

Select user define Select new target

Select MSP430G2253 in the target.

Establish the connection by selecting the TI MSP430 USBI Save.

ఆత్రబిపోభవ

7. Open New Target configuration

Right click on new target configuration

Click on Launch selected configuration

Select MSP430G2253 in the target.

Establish the connection by selecting the TI MSP430 USBI

Save.

8. Open the Run menu

Select connect project

Again open Run

Load Project

Browse the project

select the project.out

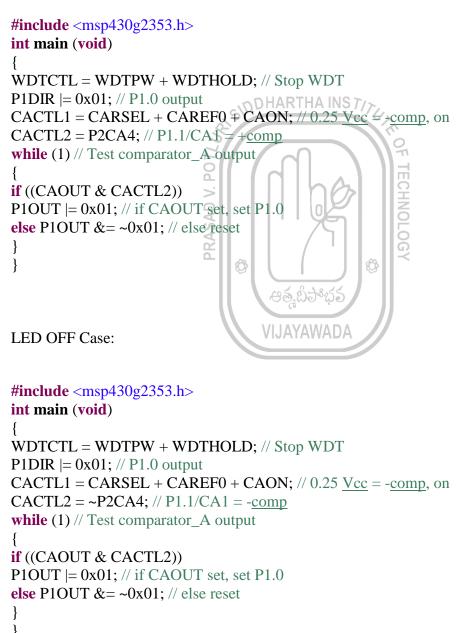
select, save and ok.

9. Run the program.

Observe the output in the console window or on the board

Program

LED ON Case:



Result: Hence used Comparator To Compare The Signal Threshold Level.

AVERAGE OF N NUMBERS

AIM : To perform average for a given series using TASM.

Experimental Requirements: PC loaded with TASM software

Procedure:

- 1. Switch on the PC, press windows+R then enter CMD.
- 2. Find the folder where TASM is located. check whether TASM.EXE, TLINK.EXE, TD.EXE are present or not
- 3. Enter into the directory where TASM is located by using cd... or directory name:
- 4. Type cd tasm in which the three files are present .Now we will be getting into c: \ or d:\ with tasm directory.
- 5. Type edit then a new window will be opened in which the program is entered.
- 6. After entering the program save the file with <filename.asm>.
- 7. Check for the errors or warnings by using TASM <filename> and press enter...
- 8. If there are no errors, then type TLINK <filename> to compile the file. If errors go back to the edit and do the necessary corrections and repeat the previous step.
- 9. Next type td <filename > to debug the executable file then will be getting the message program has no symbol table, press ok and then write down the instructions, registers and flags status before execution.
- 10. For step by step execution press F8.and for direct execution press F9 and then write down the instructions, registers and flags status after execution .Go to dump if required for noting down the required inputs and outputs.

Program:

VIJAYAWADA

ఆత్మదిపోభవ

1. AVERAGE OF N NUMBERS ((1+2+3+4+...N)/N

ASSUME CS:CODE CODE SEGMENT START: MOV AX,0000H MOV BL,08H MOV CL,BL L1: ADD AL,CL ADC AH,00H LOOP L1 DIV BL INT 03H CODE ENDS END START END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	710	DHARTHA INS 7/7		
	2-51			
			0	
	6 l			
		111hA	TECHNOLOGY	
	PRASAD V.		No No	
Result:	AS	11 7511	6	
INPUT: BL:	Here and the second sec		∬G≺	
OUTPUT:	÷ \$		2	
AX:		ఆత్త బిపోభన		
FLAG STATUS:			//	
Theoretical Calculation	ons	VIJAYAWADA		
			-	

2. AVERAGE OF N NUMBERS IN AN ARRAY

ASSUME CS:CODE, DS:DATA DATA SEGMENT LIST DB 12H,23H,45H,56H,70H DATA ENDS CODE SEGMENT START: MOV AX,DATA MOV DS,AX MOV AX,0000H MOV BL,05H MOV CL,BL MOV SI,OFFSET LIST L1: ADD AL,[SI] ADC AH,00H INC SI LOOP L1 DIV BL INT 03H CODE ENDS END START END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	210	DHARTHA INS 7/7		
	6			
			G	
	0		1) <u>–</u>	
	5	ha	i i i i i i i i i i i i i i i i i i i	
	Q		NI	
	SA			
	N/N		G	
	Ľ Ø			
		ఆత్రబిపోచన		
	(100,00°40	L)]	
		VIJAYAWADA	//	

Result: INPUT: ARR1: OUTPUT: AX: FLAG STATUS: Theoretical Calculations

Result: Average for a given series was found.

Conversion of Packed BCD to unpacked BCD and BCD to ASCII

AIM : To convert packed BCD to unpacked BCD and BCD to ASCII using TASM.

Experimental Requirements: PC loaded with TASM software

Procedure:

- 1. Switch on the PC, press windows+R then enter CMD.
- 2. Find the folder where TASM is located. check whether TASM.EXE, TLINK.EXE, TD.EXE are present or not
- 3. Enter into the directory where TASM is located by using cd... or directory name:
- 4. Type cd tasm in which the three files are present .Now we will be getting into c: \ or d:\ with tasm directory.
- 5. Type edit then a new window will be opened in which the program is entered.
- 6. After entering the program save the file with <filename.asm>.
- 7. Check for the errors or warnings by using TASM <filename> and press enter...
- 8. If there are no errors, then type TLINK <filename> to compile the file. If errors go back to the edit and do the necessary corrections and repeat the previous step.
- 9. Next type td <filename > to debug the executable file then will be getting the message program has no symbol table, press ok and then write down the instructions, registers and flags status before execution .
- 10. For step by step execution press F8.and for direct execution press F9 and then write down the instructions, registers and flags status after execution .Go to dump if required for noting down the required inputs and outputs.



Program:

1.PACKED BCD TO UNPACKED BCD

ASSUME CS: CODE CODE SEGMENT START: MOV AL, 56H MOV AH, AL SHR AH, 04H AND AL, 0FH INT 03H CODE ENDS END START END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
		در THARTHA INS		
	SIL		(r	
	5			
	F	$\langle \mathcal{A} \rangle$	T	
	PC			
	>	$ n \theta$	I I I	
	AD		6	
	AS		6	
	P.K.		GY	
	\$		\$	
		ఆవ బ్రేషిశ్రిక		
		action the	//	
		VIJAYAWADA		

Result: INPUT: AL: OUTPUT: AX: FLAG STATUS: Theoretical Calculations

2. BCD TO ASCII

ASSUME CS: CODE CODE SEGMENT START: MOV AL, 56H MOV AH, AL SHR AH, 04H AND AL, 0FH OR AX,3030H INT 03H CODE ENDS END START

END

ADDRESS	OPCODE	MNEMONIC	OPERAND	COMMENTS
	210	DHARTHA INS 7/7		
	07		0	
	5			
	L	Inhal	EC	
			H Z	
	A S		2	
	×.	/ 7		
	ā 🖓	e e	_ IIY	
	5		5	
		ఆత్త బిపోభన		
			//	
		VIJAYAWADA 🏒		
			r	

Result: INPUT: AL: OUTPUT: AX: FLAG STATUS: Theoretical Calculations

Result: packed BCD to unpacked BCD, BCD to ASCII conversion has been Performed.