Microprocessors and Microcontrollers lab

This laboratory course is designed to complement the traditional Electrical Engineering course offerings in theMicroprocessors and Microcontrollers labcourse. This course include

1. To familiarize with the assembly level programming.
2. To gain knowledge in microprocessor architecture, programming and its various applications.
3. Design circuits for various applications using microcontrollers.
4. An in-depth knowledge of applying the concepts on real- time applications.

LIST OF EXPERIMENTS

S.No.	NAME OF EXPERIMENTS	Page No.
1	Induction to 8085 Microprocessor	3-13
2	(a) Addition of 2 - 8 bit numbers (b) Addition of 2 - 16 bit numbers (c) Addition of 2 - decimal numbers (d) Addition of 8 bit number (neglecting the carry)	14-18 19-23 24-26 27-30
3	Subtraction of 2 – 16 bit numbers	31-35
4	(a) Multiplication of 2 - 16 bit numbers (b) Multiplication of two 8bit numbers (bit rotation) (c) Multiply two 8-bit numbers by repetitive addition	36-40 41-44 45-48
5	(a) Division of 2 - 16 bit number (b) Division of 2 – 8 bit numbers (c) Factorial of 8 bit number	49-54 55-59 60-63
6	(a) Separation of hexadecimal number into two digits (b) Check the parity of hex numbers	64-67 68-71
7	Speed control of stepper motor	72-75
8	(a) 1’s complement of an 8 bit number (b) 2’s complement of an 8 bit number	76-79 80-83
9	(a) Generation of square wave (b) Generation of triangular wave (c) Generation of sawtooth wave	84-86 87-90 91-93
10	Generation of beep sound on a buzzer	94-96

11	Display name on the LCD display	97-99
12	(a) Addition of 2 - 8 bit numbers using microcontroller 8051 (b) Subtraction of 2 - 8 bit numbers using microcontroller 8051 (c) Multiplication of 2 - 8 bit numbers using microcontroller 8051 (d) Division of 2 - 8 bit numbers using microcontroller 8051	100-102 103-105 106-108 109-111

EXPERIMENT NO.:01

Introduction to the Microprocessor

Contents **Page No.**

1. Objectives:……………………………………………………………4
2. Expected outcomes of Experiment:……………………………………..4
3. Theory:………………………………………………………………..4
4. Equipment Required:………………..………………………………..13
5. Results: ……………………………………………………………….13
6. Assignments: …………………………………………………………13
7. Conclusions: …………………………………………………………13

1. Objective:

To study the microprocessor 8085.

2. Expected Outcomes of Experiment:

To gain knowledge in microprocessor architecture, programming and its various applications.

3. Theory:

Architecture of 8085 Microprocessor

a) General purpose register

It is an 8 bit register i.e. B,C,D,E,H,L. The combination of 8 bit register is known as register pair, which can hold 16 bit data. The HL pair is used to act as memory pointer is accessible to program.

b) Accumulator

It is an 8-bitregister, which hold one of the data to be processed by ALU and stored the result of the operation.

c) Program counter (PC)

It is a 16-bitpointer, which maintain the address of a byte entered to line stack.

d) Stack pointer (SP)

It is a 16-bit special purpose register, which is used to hold line memory address for line next instruction to be executed.

e) Arithmetic and logical unit

It carries out arithmetic and logical operation by 8-bit address it uses the accumulator content as input the ALU result is stored back into accumulator.

f) Temporary register

It is an 8-bit register associated with ALU hold data, entering an operation, used by the microprocessor and not accessible to programs.

g) Flags

Flag register is a group of fire, individual flip flops line content of line flag register will change after execution of arithmetic and logic operation. The line states flags are

i) Carry flag (C)
ii) Parity flag (P)
iii) Zero flag (Z)
iv) Auxiliary carry flag (AC)
v) Sign flag (S)

h) Timing and control unit

Synchronous all microprocessor, operation with the clock and generator and control signal from it necessary to communicate between controller and peripherals.

i) Instruction register and decoder

Instruction is fetched from line memory and stored in line instruction register decoder the stored information.

j) Register Array

These are used to store 8 bit data during execution of some instruction.

PIN Description

Address Bus

1. The pins $A_0 – A_{15}$ denote the address bus.
2. They are used for most significant bit

Address / Data Bus

1. $AD_0 – AD_7$ constitutes the address / Data bus
2. These pins are used for least significant bit

ALE : (Address Latch Enable)

1. The signal goes high during the first clock cycle and enables the lower order address bits.

IO / M

1. This distinguishes whether the address is for memory or input.
2. When this pins go high, the address is for an I/O device.

$S_0 – S_1$

S_0 and S_1 are status signal which provides different status and functions.

RD

1. This is an active low signal
2. This signal is used to control READ operation of the microprocessor.

WR

1. WR is also an active low signal
2. Controls the write operation of the microprocessor.

HOLD

1. This indicates if any other device is requesting the use of address and data bus.

HLDA

1. HLDA is the acknowledgement signal for HOLD
2. It indicates whether the hold signal is received or not.

INTR

1. INTE is an interrupt request signal .
2. IT can be enabled or disabled by using software.

INTA

1. Whenever the microprocessor receives interrupt signal .
2. It has to be acknowledged.

RST 5.5, 6.5, 7.5

1. These are nothing but the restart interrupts .
2. They insert an internal restart junction automatically.

TRAP

1. Trap is the only non-maskable interrupt
2. It cannot be enabled (or) disabled using program.

RESET IN

1. This pin resets the program counter to 0 to 1 and results interrupt enable and HLDA flip flops.

X_1, X_2

These are the terminals which are connected to external oscillator to produce the necessary and suitable clock operation.

SID

This pin provides serial input data

SOD

This pin provides serial output data

V_{CC} and V_{SS}

1. V_{CC} is +5V supply pin
2. V_{SS} is ground pin

Specifications

1. Processors

Intel 8085 at E144 MHz clock

2. Memory

Monitor RAM:	0000 – IFFF
EPROM Expansion:	2000 – 3FFF's
	0000 – FFF
System RAM:	4000 – 5FFF
Monitor data area	4100 – 5FFF
RAM Expansion	6000 – BFFF

3. Input / Output

Parallel: A8 TTL input timer with 2 number of 32-55 only input timer available in μ-85 EBI.

Serial: Only one number RS 232-C, Compatible, crucial interface using 8281A

Timer: 3 channel -16 bit programmable units, using 8253 channel '0' used for no band late. Clock generator. Channel '1' is used for single stopping used program.

Display: 6 digit – 7 segment LED display with filter 4 digit for adder display and 2 digit for data display.

Key board: 21 keys, soft keyboard including common keys and hexa decimal keys.

RES: Reset keys allow to terminate any present activity and retain to μ - 85 its on initialize state.

INT: Maskable interrupt connect to CPU's RST 7.5 interrupt

DEC: Decrement the adder by 1

EXEC: Execute line particular value after selecting address through go command.

NEXT: Increment the address by 1 and then display its content.

Key Functions:

E 0 SUB	i. Hex entry key '0' ii. Substituting memory content where "next" key is paused immediately after 1, take used to st cutting address. iii. Register key 'E'
RD 1 REG	i) Hex code entry (1) ii) Register key 'D'
C 2 TN	i) Hex code entry '2' ii) Retricre data from data 'memory' to data top iii) Register key 'C'

B 3 TR	i)	Hex code entry ‘3’
	ii)	Retricre data from memory to top
	iii)	Register key ‘B’

F 4 BLOC	i)	Hex key entry ‘C’
	ii)	Block search from byte
	iii)	Register key ‘F’

A 5 FILL	i)	Hex key entry ‘5’
	ii)	Fill block of RAM memory with desired data
	iii)	Register key ‘A’

L 6 SER	i)	Hex key entry ‘6’
	ii)	TN/Tl used for sending (or) receiving
	iii)	Register key ‘H’

H 7 F_2	i)	Hex key entry ‘7’
	ii)	Register key ‘H’

I 8 G_0	i) Register key ‘S’ ii) Register key ‘I’
PL 9 SNG	i) Hex key entry ‘A’ ii) Function key F_3 iii) Register key “ph”
A PH F_3	i) Hex key entry “y” ii) Signal step program (instruction by instruction)
C SH MOV	i) Hex key entry “c” ii) Much a block of memory from a linear block iii) Register key “S_H”
D CMP	i) Hex key D ii) Compare 2 memory block
B SL BC	i) Hex key entry ‘B’ ii) Check a block from flame iii) Register key “SPL”
E INS	i) Hex key ‘E’ ii) Insert by test into memory (RAM)
F DEL	i) Hex key ‘F’ ii) Delete byte from memory RAM

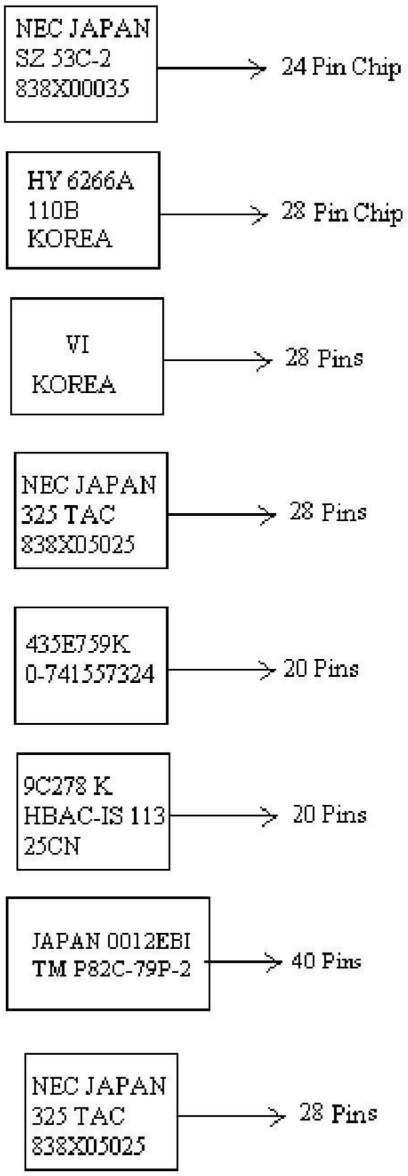
System Power Consumption

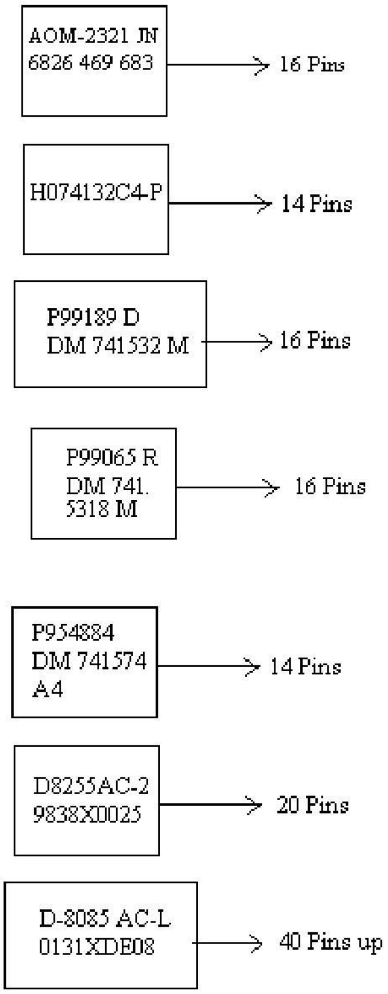
Micro BSEB2 | MICRO SSEB
+5V @ 1Amp | +5V@ 800 mA
+12V @ 200 mA
- 12V @ 100 mA

Power Supply Specification

MICRO SSEM 230V, AC @ 80 Hz +5V @ 600 mA

Key Function





IC's Used

8085	-	8 bit µp
8253	-	programmable internal timer
8255	-	programmable peripheral interface
8279	-	programmable key boards / display interface
8251	-	programmable communication interface
2764	-	8 KV VV EPROM
6264	-	8K STATIC PROM
7414	-	Hex inverter
7432	-	Quad 21/p OR GATE
7409	-	Quad 21/p AND GATE
7400	-	NAND Gate
7404	-	Dual D-FF
74373 -		Octal 'D' Latch
74139	-	Dual 2 to 4 line decoder
74138	-	3 to 8 line decoder

In Enter Program into Trainer Kit

1. Press ‘RESET’ key
2. Sub (key processor represent address field)
3. Enter the address (16 bit) and digit in hex
4. Press ‘NEXT’ key
5. Enter the data
6. Again press “NEXT”
7. Again after taking the program, are use HLT instruction its Hex code
8. Press “NEXT”

How to executive program

1. Press “RESET”
2. Press “GO”
3. Enter the address location in which line program was executed
4. Press “Execute” key

4. Equipments Required:

8085 microprocessor kit.

5. Results:

Thus 8085 microprocessor was studied successfully.

6. Assignments:

Expertise the Micro controller programming & its applications.

Expertise the concepts of theory and programming of microprocessors.

7. Conclusion:

In this experiment, the motive was to introduce students to the microprocessor. At the end, student should try to learn as much as they can from this experiment and should try different programming concepts.

EXPERIMENT NO.:02(a)

Addition of 8 bit numbers

Contents **Page no.**

1. Objectives :……………………………………………………………15
2. Expected outcomes of Experiment :……………………………………..15
3. Theory :…………………………………………………………………..15
4. EquipmentRequired :……………….……………………………………...15
5. Procedure : ………………………………………………………………16
6. Coding : …………………………………………………………………17
7. Results : …………………………………………………………………17
8. Assignments : …………………………………………………………...18
9. Conclusions : ……………………………………………………………18

1. Objective:

To write an assembly language for adding two 8 bit numbers by using microprocessor kit.

2. Expected Outcomes of Experiment:

Understanding of the addition of two eight-bit numbers in microprocessor.

3. Theory:

8-bit addition is much like decimal addition except that you are only adding 1s and 0s. When the sum exceeds 1, carry a 1 over to the next-more-significant column.

0 + 0 = 0 carry 0
0 + 1 = 1 carry 0
1 + 0 = 1 carry 0
1 + 1 = 0 carry 1

General form: $A_0 + B_0 = \Sigma_0 + C_{out}$

Where Summation symbol (Σ)

Carry-out (C_{out})

Truth table for addition of two binary digits			
A_0	B_0	$\sum_0$	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

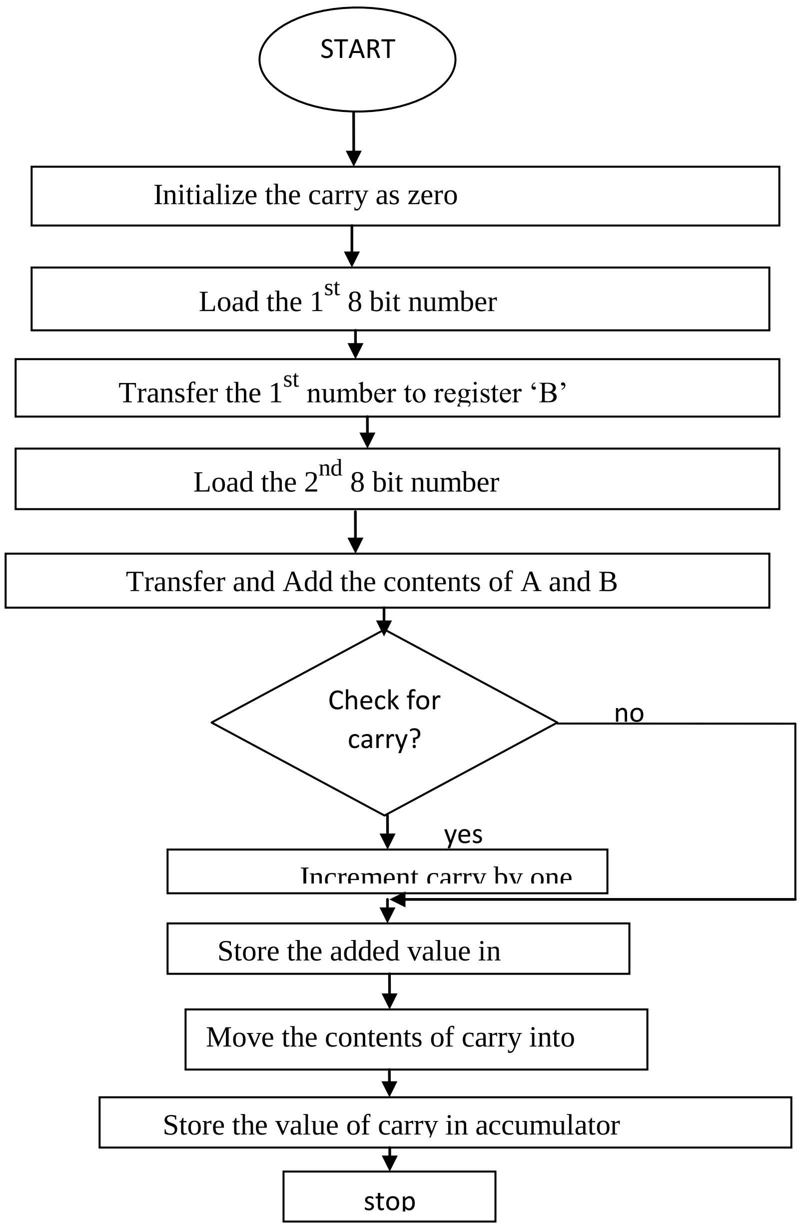
4. Equipments Required:

8085 micro processor kit
(0-5V) DC battery.

5. Procedure:

Algorithm:

Step1	:	Start the microprocessor.
Step2	:	Intialize the carry as ‘Zero’.
Step3	:	Load the first 8 bit data into the accumulator.
Step4	:	Copy the contents of accumulator into the register ‘B’
Step5	:	Load the second 8 bit data into the accumulator.
Step6	:	Add the 2 - 8 bit datas and check for carry.
Step7	:	Jump on if no carry.
Step8	:	Increment carry if there is.
Step9	:	Store the added request in accumulator.
Step10	:	More the carry value to accumulator.
Step11	:	Store the carry value in accumulator.
Step12	:	Stop the program execution.



6. Coding:

Address	Label	Mnemonics	Hex Code	Comments
4100		MVI C,00	OE, 00	Initialize the carry as zero
4102		LDA 4300	3A, (00, 43)	Load the first 8 bit data
4105		MOV, B,A	47	Copy the value of 8 bit data into register B
4106		LDA 4301	3A, (01, 43)	Load the second 8 bit data into the accumulator
4109		ADD B	80	Add the hoo values
410A		JNC	D2, 0E, 41	Jump on if no carry
410D		INR C	OC	If carry is there increment it by one
410E	Loop	STA 4302	32 (02, 43)	Stone the added value in the Accumulator
4111		MOV A,C	79	More the value of carry to the Accumulator from register C
4112		STA 4303	32 (03, 43)	Store the value of carry in the accumulator
4115		HLT	76	Stop the program execution

7. Results:

Input

Without carry

Input Address	Value
4300	04
4301	02

Output

Output Address	Value
4302	06
4303	00 (carry)

With carry

Input Address	Value
4300	FF
4301	FF

Output Address	Value
4302	FE
4303	01 (carry)

Calculation 1111 1111

1111 1111

8. Assignments:

Add two numbers 05 & 08where 05= [0000 0101] and 08 = [0000 1000].

9. Conclusion:

This experiment introduces concept of theory and programming ofBinary Arithmetic addition.

It helped students in applying such operation in real time scenario on arithmetic.

Student should try the mentioned operations and implement them on binary addition.

EXPERIMENT NO.: 02(b)

Addition of two 16 – bit numbers

Contents **Page No.**

1.Objectives:...20

2.Expected outcome of experiment:...20

3.Theory: 20

4.Equipments required: 20

5. Procedure: 20

6. Coding: …………………………………………………………………….22

7. Result: 22

8. Assignments: 23

9. Conclusion: ……………………………………………………………….23

1. Objective:

To write an assembly language program for adding two 16 bit numbers using 8085 micro processor kit.

2. Expected Outcome of Experiment:

Knowledge about addition of two 16 bit numbers.

To familiarize with the assembly level programming.

3. Theory:

16-bit addition is the addition of two 16-values. First, we must recognize that the addition of two 16-bit values will result in a value that is, at most, 17 bits long. Why is this so? The largest value that can fit in 16-bits is 256 * 256 - 1 = 65,535. If we add 65,535 + 65,535, we get the result of 131,070. This value fits in 17 bits. Thus when adding two 16-bit values, we will get a 17-bit value. Since the 8051 works with 8-bit values, we will use the following statement: "*Adding two 16-bit values results in a 24-bit value*". Of course, 7 of the highest 8 bits will never be used--but we will have our entire answer in 3 bytes. Also keep in mind that we will be working with unsigned integers.

Programming Tip: Another option, instead of using 3 full bytes for the answer, is to use 2 bytes (16-bits) for the answer, and the carry bit (C), to hold the 17th bit. This is perfectly acceptable, and probably even preferred. The more advanced programmer will understand and recognize this option, and be able to make use of it. However, since this is an introduction to 16-bit mathematics it is our goal that the answer produced by the routines be in a form that is easy for the reader to utilize, once calculated. It is our belief that this is best achieved by leaving the answer fully expressed in 3 8-bit values.

4. Equipments Required:

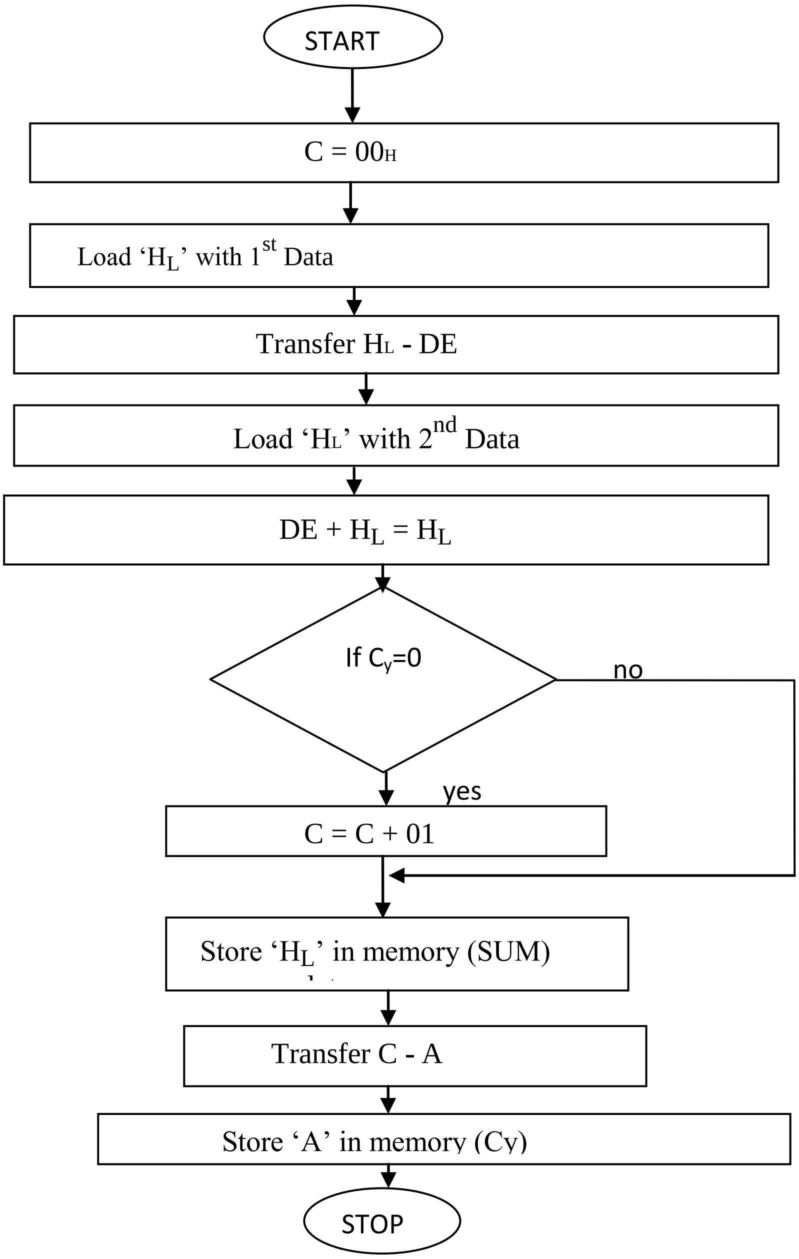
8085 micro processor kit
(0-5V) DC battery.

5. Procedure:

Algorithm:

Step1	:	Start the microprocessor
Step2	:	Get the 1st 8 bit in 'C' register (LSB) and 2nd 8 bit in 'H' register (MSB) of 16 bit number.
Step3	:	Save the 1st 16 bit in 'DE' register pair
Step4	:	Similarly get the 2nd 16 bit number and store it in 'HL' register pair.
Step5	:	Get the lower byte of 1st number into 'L' register
Step6	:	Add it with lower byte of 2nd number

Step7	:	tore the result in ‘L’ register
Step8	:	Get the higher byte of 1st number into accumulator
Step9	:	Add it with higher byte of 2nd number and carry of the lower bit addition.
Step10	:	Store the result in ‘H’ register
Step11	:	Store 16 bit addition value in ‘HL’ register pair
Step12	:	Stop program execution



6. Coding:

Address	Label	Mnemonics		Hex Code	Comments
4500		MVI	C,00	0E	C = 00_H
4501				00	
4502		LHLD	4800	2A	HL – 1st No.
4503				00	
4504				48	
4505		XCHG		EB	HL – DE
4506		LHLD	4802	2A	HL – 2nd No.
4507				02	
4508				48	
4509		DAD	D	19	Double addition DE + HL
450A		JNC	Ahead 450E	D2	If Cy = 0, G0 to 450E
450B				0E	
450C				45	
450D		INR	C	0C	C = C + 01
450E	AHEAD	SHLD	4804	22	HL – 4804 (sum)
450F				04	
4510				48	
4511		MOV	C,A	79	Cy – A
4512		STA	4806	32	Cy – 4806
4513				06	
4514				48	
4515		HLT		76	Stop execution

7. Result:

Input

Without

Input Address	Value
4800	01 (addend)
4801	04
4802	02 (augend)
4803	03 (augend)

Output

Output Address	Value
4804	03 (sum)
4805	07 (sum)
4806	00 (carry)

```
Calculation    0000   0100   0000   0001
               0000   0011   0000   0010
               --------------------------------
               0000   0111   0000   0011
               0      7      0      3
```

8. Assignment:

Write a program to add 0033 and 05FF.

9. Conclusion:

This experiment introduces concept of theory and programming ofBinary Arithmeticaddition.It helped students in applying such operation in real time scenario on arithmetic. Student should try the mentioned operations and implement them on binary addition.

EXPERIMENT NO.:02(c)

Addition of Two Decimal Numbers

Contents **Page no.**

1. Objectives: 25
2. Expected outcome of experiment: 25
3. EquipmentRequired:...25
4.Procedure:...25
5. Coding:...26
6.Results:..26
7.Assignments: 26
8.Conclusions:...26

1.Objective:

To write an assembly language program for addition of two decimal numbers using microprocessor kit.

2. Expected Outcome Of Experiment:

Knowledge of the addition of two decimal numbers.
Design circuits for various applications using Microcontrollers.

3. Equipments Required:

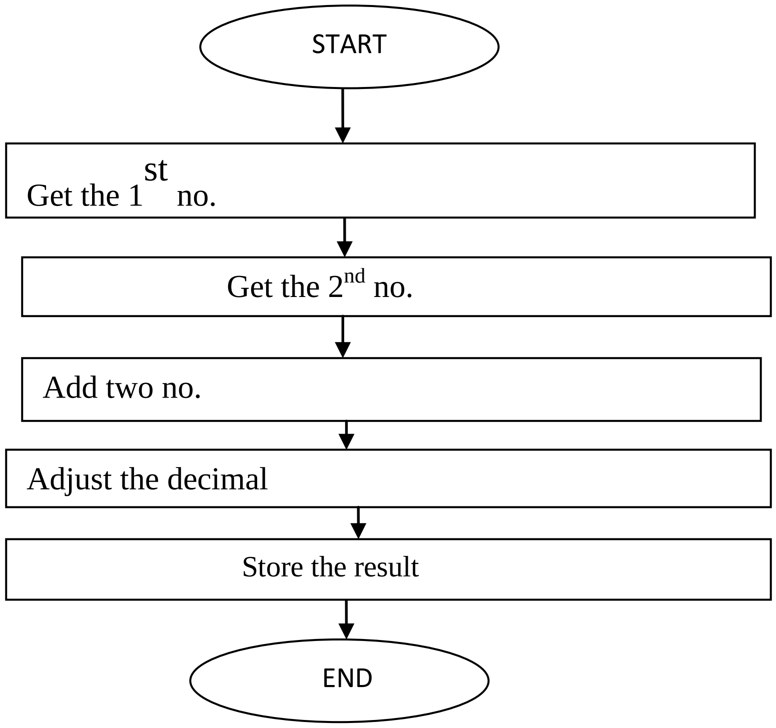
8085 microprocessor kit.

(0-5V) DC supply.

4. Procedure:

Algorithm:

Step1 Initialize HL Reg. pair with address where the first number islying.
Step 2 : Store the number in accumulator.
Step 3 : Get the second number.
Step 4 : Add the two numbers and store the result in 200B.
Step 5 : Go back to Monitor .



Address	Label	Mnemonics	Hex Code	Comments
2000	Data			Two decimal no. to be added
2001	Data			
2002	Result			Result
2003		LXI H, 2000	21,00,20	Point 1st no.
2006		MOV A, M	7E	Load the acc.
2007		INX H	23	Adv Pointer
2008		ADD M	86	ADD 2nd NO
2009		DAA	27	Convert to decimal
200A		INX H	23	Adv Pointer
200B		MOV M, A	77	Store Result
200C		RST 5		

5. Coding:

6. Results:

Example:

Address	Data	Comments
2100	05	The no. of hexadecimal no.
2101	10	1st hex no.
2102	02	2nd hex no.
2103	08	3rd hex no.
2104	04	4th hex no.
2105	01	5th hex no.
2106	1F	Result

7.Assignments:

Write an assembly language code for Addition two numbers 25 & 13.

8.Conclusion:

In this experiment, the motive was to provide students with solid foundation on interfacing the external devices to the processor according to the user requirements to create novel products and solutions for the real life problems.

EXPERIMENT NO.:2(d)

Addition of 8 bit number (neglecting the carry)

Contents **Page No.**

1.Objectives :……………………………………………………………..28

2.Expected outcomes of Experiment :……………………………………...28

3.Equipments Required :…………………………………………………28

4.Procedure : ………………………………………………………….….28

5.Coding : …………………………………………………………….….29

6.Results : ………………………………………………………………..30

7.Assignments : ………………………………………………………….30

8.Conclusions : …………………………………………………………..30

1. Objective:

Addition of 8 bit number series neglecting the carry generated.

2. Expected Outcomes of Experiment:

Ability to design and conduct experiments, as well as to analyze assembly programming.
An in-depth knowledge of applying the concepts on addition of two 8 bit numbers neglecting carry generated.

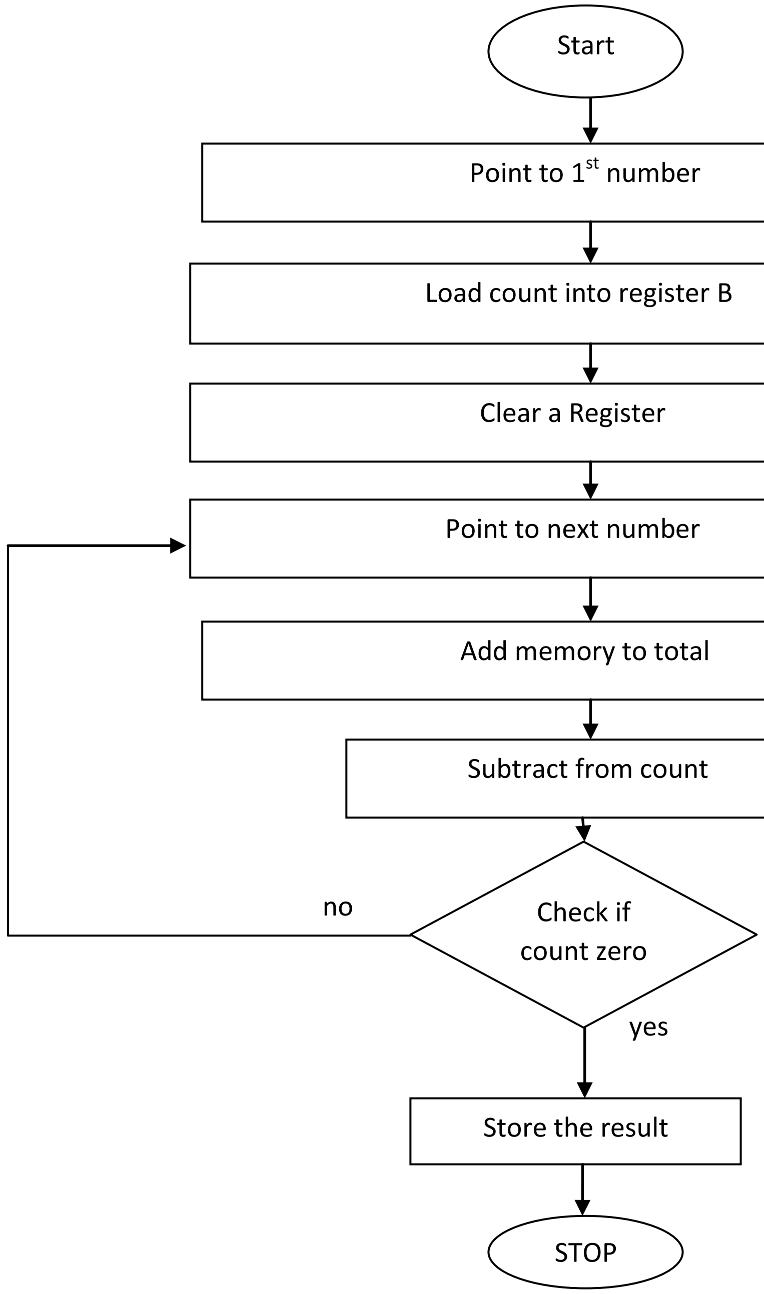
3. Equipments Required:

8085 microprocessor kit.
(0-5V) DC battery

4. Procedure:

Algorithm:

Step 1 : Load the content from 2100 location, give how many bytes are to be added.
Step 2 : Initialize the accumulator as the result will be stored in accumulator.
Step 3 : Let the memory to point the no. of the bytes to be added into partial register acc.
Step 4 : Decrement the Counter having no. of bytes.
Step 5 : Check if zero- No. repeat from point 3.
Step 6 : Store the result to 2100 location.
Step 7 : Go back to monitor.
Step 8 : Execute from 2000.



5. Coding:

Address	Label	Mnemonics	Hex Code	Comments
2000	Start	LXI H, 2100	21 00 21	Point to first no.
2003		MOV B,M	46	Load count into B register
2004		XRA A	AF	Clear A register
2005	Loop	INX H	23	Point to 1st number
2006		ADD M	86	Add memory to total
2007		DCR B	05	Subtract from count
2008		JMC LOOP	C2 05 20	Test to see if done

200B		STA 2100	32 00 21	Save the result
200E		RST 5	EF	

6. Results:

Memory Address	Data
2100	04
2101	16
2102	02
2103	08
2104	04

7.Assignments:

Pick the example from textbook to find addition of two 8 bit numbers neglect carry generated in assembly language by the advice of instructor lab.

Write a program to find addition of 23 and 44.

8. Conclusions:

This experiment introduces concept of theory and programming ofBinary Arithmetic addition. It helped students in applying such operation in real time scenario on arithmetic. Student should try the mentioned operations and implement them on binary addition.

EXPERIMENT NO.:03

Subtraction of two 16 – bit numbers

Contents **Page No.**

1. Objectives:…………………………………………………………32

2. Expected outcomes of Experiment:……………………………………..32

3.Theory:…………………………………………………………..32

4. Equipments Required:……………………………………………...32

5. Procedure: …………………………………………………………..32

6. Coding: ……………………………………………………………..33

7. Results: ……………………………………………………………..34

8. Assignments: ………………………………………………………..35

9. Conclusion: ………………………………………………………....35

1.Objective:

To write an assembly language program for subtracting two 16 bit numbers using 8085 microprocessor kit.

2. Expected outcome of experiment:

Knowledge of subtraction two 16 bit numbers using 8085 microprocessor kit.

3. Theory:

Binary subtraction is a similar process to decimal subtraction. To perform the binary subtraction of two numbers carry out the subtraction for each pair of corresponding bits, starting from the l.s.b. to m.s.b., using he following rules;

- 0 - 0 = 0.
- 1 - 0 = 1.
- 0 - 1 = 1 Subtract 1 from next column.
- 0 - 1 - 1 = 0 Subtract 1 from next column.
- 1 - 1 - 1 = 1 Subtract 1 from next column

4. Equipment Required:

8085 microprocessor kit (0-5V)
DC battery.

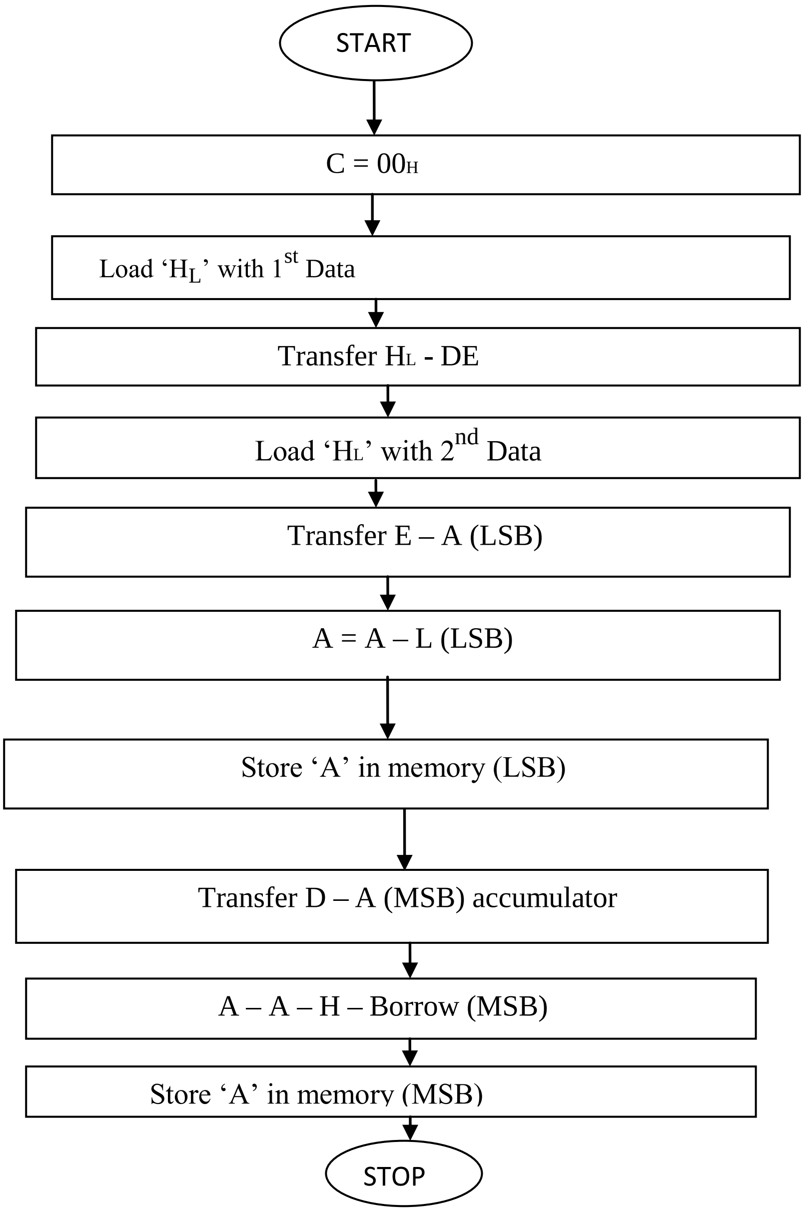
5. Procedure:

Algorithm:

Step	1	:	Start the microprocessor
Step	2	:	Get the 1st 16 bit in 'HL' register pair
Step	3	:	Save the 1st 16 bit in 'DE' register pair
Step	4	:	Get the 2nd 16 bit number in 'HL' register pair
Step	5	:	Get the lower byte of 1st number
Step	6	:	Get the subtracted value of 2nd number of lower byte by subtracting it with lower byte of 1st number
Step	7	:	Store the result in 'L' register
Step	8	:	Get the higher byte of 2nd number
Step	9	:	Subtract the higher byte of 1st number from 2nd number with Borrow

Step 10 : Store the result in 'HL' register

Step 11 : Stop the program execution



6. Coding:

Address	Label	Mnemonics		Hex Code	Comments
4500		MVI	C,00	0E	C = 00_H
4501				00	
4502		LHLD	4800	2A	L – 1st No.
4503				00	
4504				48	
4505		XLHG		EB	HL – DE
4506		LHLD	4802	2A	HL – 2nd No.

4507				02	
4508				48	
4509		MOV	A,E	7B	LSB of ‘1’ to ‘A’
450A		SUB	L	95	A – A – L
450B		STA	4804	32	A – memory
450C				04	
450D				48	
450E		MOV	A,D	7A	MSB of 1 to A
450F		SBB	H	9C	A- A – H
4510		STA	4805	32	A – memory
4511				05	
4512				48	
4513		HLT		76	Stop execution

7. Result:

Input

Without borrow

Input Address	Value
4800	07
4801	08
4802	05
4803	06

Output

Output Address	Value
4804	02
4805	02
4807	00

With borrow

Input Address	Value
4800	05
4801	06
4802	07
4803	08

Output Address	Value
4804	02
4805	02

4806	01

Calculation

05	06	-	07	08				
05	06	0101	0110		07	08	0111	1000
CMA		1010	1001		CMA		1000	0111
ADI		0000	0001		ACI		0000	0001
		---------------					--------------	
		1010	1010				1000	1000
05	06	+	07	08				
			1010	1010				
			1000	1000				
			---------------					
		(1)	0010	0010				
			02	02				

8.Assignment:

1. Perform and write down assembly language code forsubtraction of any two 16 bit numbers by the advice of lab instructor.

2.Write assembly language code forsubtraction of any two 16 bit numbers with help of book.

9. Conclusion:

This experiment introduces concept of theory and programming ofBinary Arithmetic subtraction.
It helped students in applying such operation in real time scenario on arithmetic.
Student should try the mentioned operations and implement them on coding.

EXPERIMENT NO.:04(a)

16 – Bit multiplication

Contents **Page No.**

1. Objectives:……………………………………………………………37

2.Expected outcomes of Experiment:……………………………………..37

3. Theory:…………………………………………………………………..37

4.Equipment Required:….……………………………………………...37

5. Procedure:…………………………………………………………..…37

6. Coding: ……………………………………………………………….39

7. Results: ……………………………………………………………….39

8. Assignments: …………………………………………………………40

9. Conclusion: …………………………………………………………...40

1. Objective:

To write an assembly language program for 16 bit multiplication by using 8085 microprocessor kit.

2. Expected outcome of experiment:

Knowledge of 16 bit multiplication.

An in-depth knowledge of applying the concepts on real- time applications.

3. Theory:

16-bit multiplication is the multiplication of two 16-bit value from another. Such a multiplication results in a 32-bit value.

Programming Tip: In fact, any multiplication results in an answer which is the sum of the bits in the two multiplicands. For example, multiplying an 8-bit value by a 16-bit value results in a 24-bit value (8 + 16). A 16-bit value multiplied by another 16-bit value results in a 32-bit value (16 + 16), etc.

For the sake of example, let's multiply **25,136** by **17,198**. The answer is 432,288,928. As with both addition and subtraction, let's first convert the expression into hexadecimal: **6230h** x **432Eh.**

4. Equipments Required:

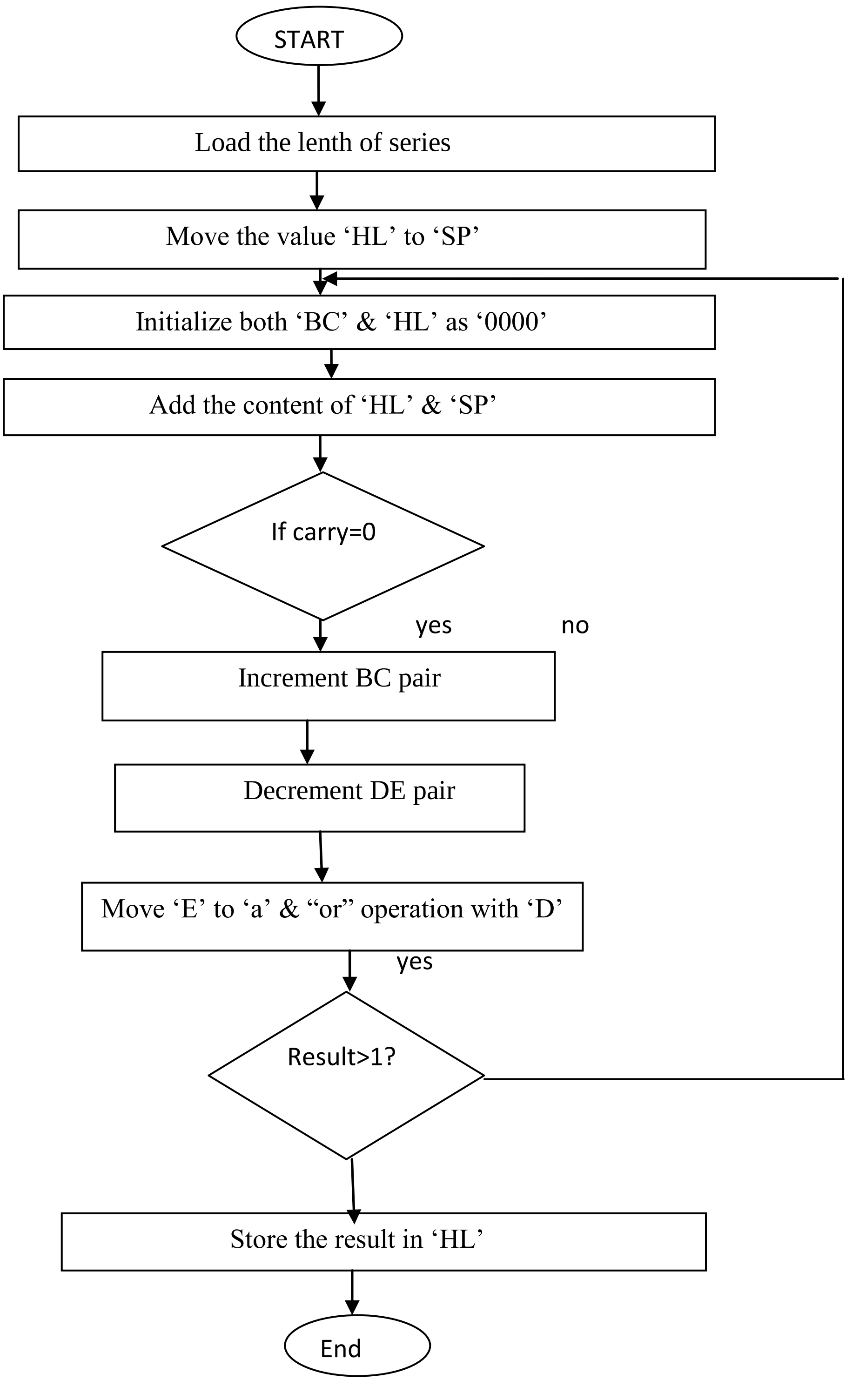
8085 microprocessorkit

(0-5V) DC battery

5. Procedure:

Algorithm:

Step1	:	Start the microprocessor
Step2	:	Load the 1st data in 'HL' register pair
Step3	:	Move content of 'HL' pair to stack pointer
Step4	:	Load the 2nd data in 'HL' and move it to 'DE'
Step5	:	Make 'HL' pair as '00' and '00'
Step6	:	Add 'HL' pair and 'SP'
Step7	:	Check for carry condition, if carry is present increment it by one else move to next step.
Step8	:	Decrement DE register
Step9	:	Then move E to 'A' and perform 'OR' operation with 'a' and 'D'
Step10	:	The value of operation is zero, then store the value else go to step 3
Step11	:	Stop the program


START
Load the lenth of series
Move the value ‘HL’ to ‘SP’
Initialize both ‘BC’ & ‘HL’ as ‘0000’
Add the content of ‘HL’ & ‘SP’
If carry=0
yes
no
Increment BC pair
Decrement DE pair
Move ‘E’ to ‘a’ & “or” operation with ‘D’
yes
Result>1?
Store the result in ‘HL’
End

6. Coding:

Memory Location	Hex Code	Label	Mnemonics		Comments
			Op code	Operand	
4100	2A,00,42		LHLD	4200	Get the 1st data in HL
4103	F9		SP HL		Save it in stack pointer4106
4106	2A,02,42		LHLD	4202	Get the 2nd data in HL
4107	EB		XCHG		Exchange 'HL' and 'DC'
4108	21,00,00		LXI H	0000	Make HL – 0000
410B	01,00,00		LXI B	0000	Make BC – 0000
410E	39	Next	DAD	SP	Add 'SP' and 'HL'
410F	D2, 13, 41		JNC	Loop	Jump to loop if no carry
4112	03		INX	B	Increment 'BC' by one
4113	1B	Loop	DCX	D	Decrement 'DE' By One
4114	7B		MOV	A,E	Make E – A
4115	B2		ORA	D	'OR' gate between A & D
4116	C2,0E,41		JNZ	Next	Jump on if Number zero
4119	22,04,42		SHLD	4204	Store the LSB In memory
411C	69		MOV	L,C	Make C to L
411D	60		MOV	H,B	Make B to H
411E	22,06,42		SHLD	4206	Store the MSB In memory
4121	76		HLT		Stop the program

7. Result:

Input

Input Address	Value
4200	04
4201	07
4202	02
4203	01

Output

Output Address	Value
4204	08
4205	12
4206	01
4207	00

8.Assignments:

Pick the example from textbook to multiply two 16 bit numbers in assembly language.

9. Conclusion:

This experiment introduces concept of theory and programming ofBinary Arithmetic multiplication.

It helped students in applying such operation in real time scenario on arithmetic.

Student should try the mentioned operations and implement them on coding.

EXPERIMENT NO.:4(b)

Multiplication of two 8bit numbers (bit rotation)

Contents **Page No.**

1.Objectives :…………………………………………………………………..42

2.Expected outcomes of Experiment :………………………………………42

3.Theory :……………………………………………………………………...42

4. Equipment Required :………………..…………………………………...42

5.Procedure : …………………………………………………………………..42

6.Coding : ……………………………………………………………………..44

7.Results : ……………………………………………………………………..44

8.Assignments : ……………………………………………………………….44

9.Conclusions : ………………………………………………………………..44

1. Objective:

Write a program to perform multiplication of two 8 bit numbers using bit rotation method.

2. Expected Outcomes of Experiment:

Ability to design and conduct experiment, as well as to analyze assembly programming. An in-depth knowledge of applying the concepts on multiplication of two 8 bit numbers using bit rotation method.

3. Theory:

We will multiply the numbers using add and shift method. In this method, you add number with itself and rotate the other number each time and shift it by one bit to left alongwith carry. If carry is present add the two numbers.Initialize the count to 4 as we are scanning for 4 digits. Decrement counter each time the bits are added. The result is stored. Display the result.

Example:

Steps	Product								Multiplier					Comments
	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0	CY	B_3	B_2	B_1	B_0	
	0	0	0	0	0	0	0	0	0	0	1	0	1	Initial Stage
Step 1	0	0	0	0	0	0	0	0	0	1	0	1	0	Shift left by 1
	0	0	0	0	0	0	0	0	0	1	0	1	0	Don’t add since CY= 0
Step 2	0	0	0	0	0	0	0	0	1	0	1	0	0	Shift
	0	0	0	0	1	1	0	0	1	0	1	0	0	Add multiplicand;CY=1
Step 3	0	0	0	1	1	0	0	0	0	1	0	0	0	Shift left by 1
	0	0	0	1	1	0	0	0	0	1	0	0	0	Don’t add since CY= 0
Step 4	0	0	1	1	0	0	0	0	1	0	0	0	0	Add multiplicand;CY=1

4. Equipments Required:

8085 microprocessor kit.
(0-5V) DC battery.

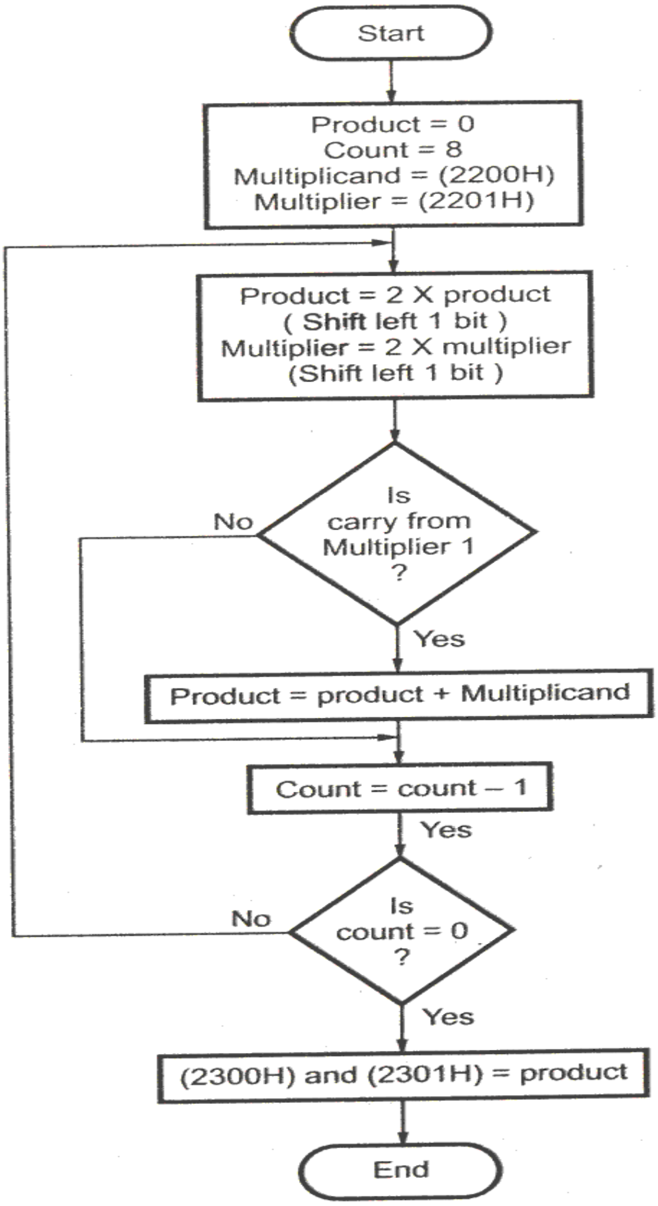
5. Procedure:

Algorithm:

1. Start the program by loading HL register pair with address of memory location.
2. Move the data to a register (E register).
3. Get the second data and load into Accumulator.

4. Add the two register contents.
5. Check for carry.
6. Increment the value of carry.
7. Check whether repeated addition is over and store the value of product and carry in memory location.

8. Terminate the program.



6. Coding:

Program:

```
            LXI H, 2200 H       ;Initialize the memory pointer
            MOV E , M           ;   Get multiplicand
            MVI D, 00 H         ;   Extend to 16 bits
            INX H               ;   Increment memory pointer
            MOV A , M           ;   Get Multiplier
            LXI H , 0000 H      ;   Product = 0
            MVI B, 08 H         ;Initialize counter with count 8
LOOP:  DAD H                    ;   Product = product X 2

            RAL
            JNC  XYZ            ;   Is carry from multiplier 1?
            DAD D               ;  Yes, product = product + multiplicand
XYZ:  DCR  B                    ;   Is counter = 0
            JNZ  LOOP           ;      No, repeat
            SHLD 2300 H         ;      Store the result
            HLT
```

7. Results:

Multiplication has been carried out between the data of 2200H and 2201 H.

8.Assignments:

Pick the example from textbook to find multiplication of two 8 bit numbers by bit rotation method in assembly language.

Write a program to find multiplication of 09 and 13 using bit rotation method.

9. Conclusions:

This experiment introduces concept of theory and programming ofBinary Arithmetic addition. It helped students in applying such operation in real time scenario on arithmetic. Student should try the mentioned operations and implement them on binary addition.

EXPERIMENT NO.:4(c)

Multiply two 8-bit numbers by repetitive addition

Contents **Page No.**

1.Objective :…………………………………………….…………………..46

2.Expected outcomes of Experiment :……………………………………...46

3.Theory :…………………………………………………………………...46

4.Equipments Required :……………………………………………………46

5.Procedure : ………………………………………………………………….46

6.Coding : ……………………………………………………………………..47

7.Results : ……………………………………………………………………..47

8.Assignments : ……………………………………………………………….48

9.Conclusions : ………………………………………………………………..48

1. Objective:

Write a program to multiply two 8-bit numbers by repetitive addition method using 8085.

2. Expected Outcomes of Experiment:

To familiarize with the assembly level programming to multiply two 8-bit numbers by repetitive addition method using 8085.

Simplified the binary multiplication.

3. Theory:

Multiplication is nothing but repeated addition. In successive addition method, one number is accepted and other number is taken as a counter. The first number is added with itself, till the counter decrements to zero.

4. Equipments Required:

8085 microprocessor kit.
(0-5V) DC battery.

5. Procedure:

1. Switch ON the microprocessor training kit with built in power supply.
2. When the kit is ON then check, “UP 85” display on the display board.
3. Now press the REL/EXMEM key on the key board and check that “.” (Dot) displayed.
4. Load machine code at the specific addresses in the user defined memory from 2000 to 3FFF.
5. Fill the machine code on next address location by pressing the NEXT key on keyboard and check the previous code by pressing PRE key.
6. After entering the whole program, verify it by execution.

6. Coding:

ADDRESS	OPCODE	LABEL	MNEMONIC	OPERAND	COMMENTS
2000	21,00,25		LXI	H, 2500 H	Address for count in H-L Pair
2003	4E		MOV	C, M	Count in register C
2004	3E,00		MVI	A, 00	Initial value of sum = 00
2006	23	LOOP	INX	H	Address of next data in H-L pair
2007	86		ADD	M	Previous sum + next no.
2008	0D		DCR	C	Decrement count.
2009	C2,06,20		JNZ	LOOP	If count is not zero, then jump to LOOP
200C	32,50,24		STA	2450 H	Store sum in 2450H
200F	76		HLT		Halt

7. Result:

The assembly language program for multiply two 8-bit numbers by repetitive addition method was executed using 8085 micro processing kit.

Input:
4150(10)
4151(02)

Output:
4152(20)
4153(00)

8.Assignments:

Write a program to multiply by repetitive addition 1101 0111 and 0111 1101.

9. Conclusions:

This experiment simplified Binary multiplication. It helped students in applying such operation in real time scenario on arithmetic. Student should try the mentioned operations and implement them.

EXPERIMENT NO.: 05(a)

16 – Bit division

Contents **Page No.**

1. Objectives:……………………………………………………………50

2.Expected outcomes of Experiment: ……………………………………..50

3.Theory:…………………………………………………………………..50

4. Equipments Required:…………………………………………...…51

5.Procedure: ……………………………………………………………51

6. Coding: ……………………………………………………………….53

7. Results: ………………………………………………………………..53

8.Assignments: …………………………………………………………53

9.Conclusions: …………………………………………………………..54

1.Objective:

To write an assembly language program for 16 bit division in 8085 microprocessor.

2. Expected Outcome of The Experiment:

To familiarize with the assembly level programming.
An in-depth knowledge of applying the concepts on binary division.

3. Theory:

16-bit division is the division of one 16-bit value by another 16-bit value, returning a 16-bit quotient and a 16-bit remainder. I used r1/r0 for dividend/remainder and r3/r2 for divisor/quotient.

Programming Tip: The number of bits in the quotient and the remainder can never be larger than the number of bits in the original divident. For example, if you are dividing a 16-bit value by a 2-bit value, both the quotient and the remainder must be able to handle a 16-bit result. If you are dividing a 24-bit value by a 16-bit value, the quotient and remainder must both be able to handle a 24-bit result.

So, again, let's remember how we did division in elementary school. For example, 179 divided by 8:

```
1 7 9 / 8 = 22 (quotient)
1 6
---
  1 9
  1 6
  ---
    3 (remainder)
```

It's necessary to follow this same process step by step. There is a 3-digit-dividend, so we expect 3 digits maximum for quotient. We "shift left" the divisor 2 digits (3-1) such that the number of digits in the divisor is the same as the number of digits in the dividend. So we get:

1 7 9 / 8 0 0 = ? ? ?

We divide the two numbers, multiply the result by the divisor and substract this result from the dividend. In this first step 179 can't be divided by 800, so the the result is 0. We subtract 0 from 179 and still have 179:

```
1 7 9 / 8 0 0 = 0 ? ?
    0
-----
1 7 9
```

We then "shift right" the divisor 1 digit and repeat the process. 179 divided by 80 results in an answer of 2. After we subtract 160 (2x80) we are left with a remainder of 19:

```
1 7 9 / 8 0 = 0 2 ?
1 6 0
-----
  1 9
```

We repeat the process again until the divisor has shifted into its original position:

```
1 7 9 : 8 = 0 2 2
1 6 0
-----
  1 9
  1 6
-----
    3
```

This may have been an unnecessary review of elementary school math, but it is important to remember exactly how the process is performed because we do *exactly* the same with the 8052 in binary system.

In this routine we will place the original dividend into R1 (high-byte) and R0 (low-byte) and the divisor in R3 (high-byte) and R2 (low-byte).

In the case of our example (179 divided by 8), the initial registers would be:

R1/R0 00000000 10110011
R3/R2 00000000 00001000

4. Equipments Required:

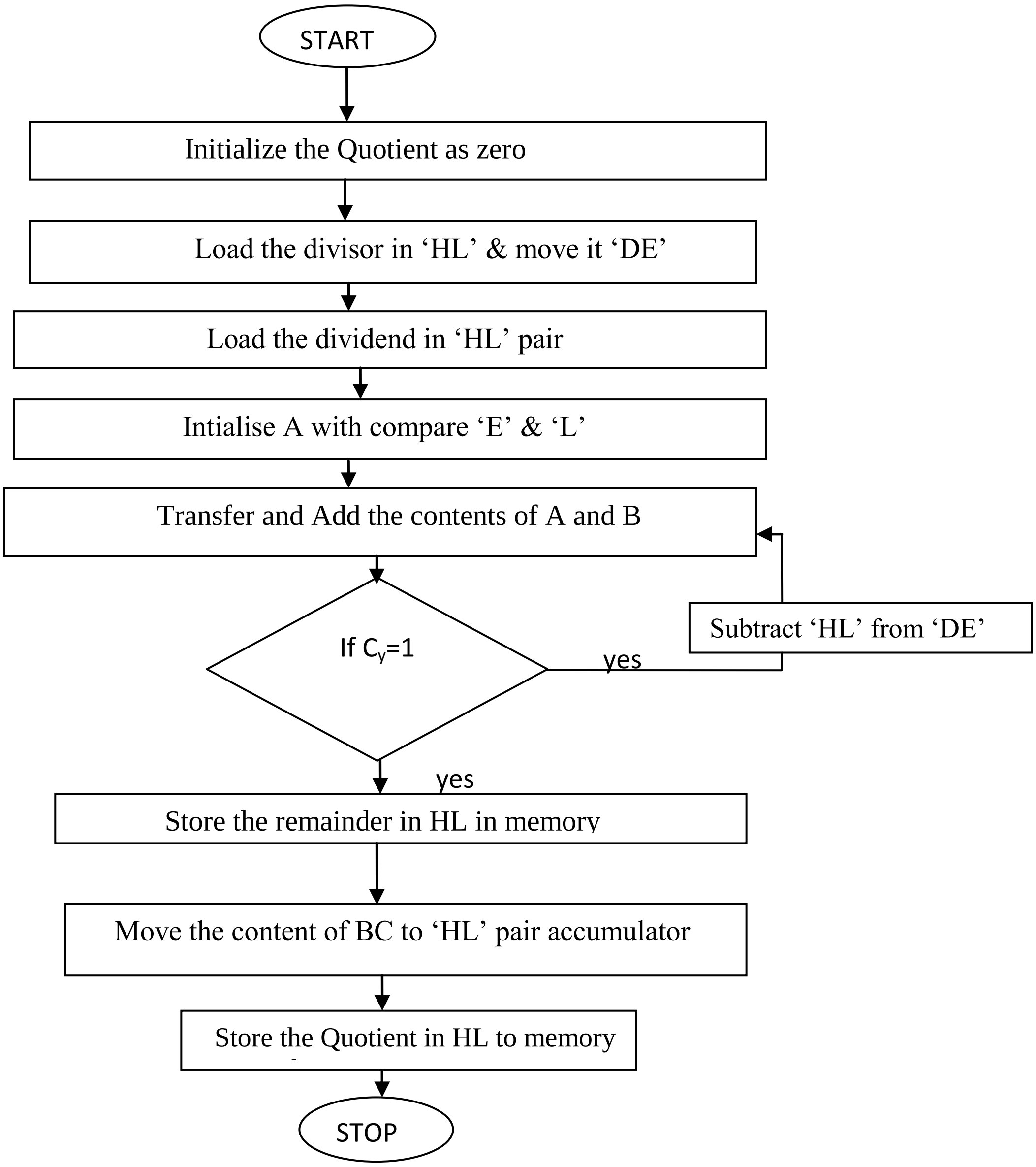
8085 microprocessor kit
(0-5V) DC battery

5. Procedure:

Algorithm:

Step1	:	Start the microprocessor
Step2	:	Intialise ‘BC’ as ‘0000’ for Quotient
Step3	:	Load the divisor in ‘HL’ pair and save it in ‘DE’ register pair
Step4	:	Load the dividend in ‘HL’ pair
Step5	:	Move the value of ‘a’ to register ‘E’
Step6	:	Subtract the content of accumulator with ‘E’ register
Step7	:	Move the content ‘A’ to ‘C’ & ‘H’ to ‘A’
Step8	:	Subtract with borrow, the content of ‘A’ with ‘D’

Step9	:	Move the value of ‘a’ to ‘H’
Step10	:	If cy = 1, go to step 12, otherwise next step
Step11	:	Increment ‘B’ register & jump to step ‘4’
Step12	:	Add both contents of ‘DC’ and ‘HL’
Step13	:	Store the remainder in memory
Step14	:	Move the content of ‘C’ to ‘L’ & ‘B’ to ‘H’
Step15	:	Store the Quotient in memory
Step16	:	Stop the program



6. Coding:

Address	Label	Mnemonics	Hex Code	Comments
4500		LXI B,0000	0,00,00	Intialise Quotient as ‘0000’
4503		LHLD 4802	2A,02,48	Load the divisor in ‘HL’
4506		XCHG	EB	Exchange ‘HL’ and ‘DE’
4507		LHLD 4800	2A,00,48	Load the dividend
450A	Loop 2	MOV A,L	7D	Move the ‘L’ value to ‘A’
450B		SUB E	93	(A-E) – A
450C		MOV L,A	6F	A- L (A value is move t L)
450D		MOV A,H	7C	H – A (a is stored with H)
450E		SBB D	9A	Subtract ‘D’ from ‘A’
450F		MOV H,A	67	Then A is moved to ‘H’
4510		JC loop 1	DA,17,45	If cy is present go to loop 1
4513		INX B	03	Increment BC pair by 1
4514		JMP loop 2	C3, 0A, 45	Jump to loop 2
4517	Loop 1	DAD ‘D’	19	‘DE’ and ‘HL’ pair all added
4518		SHLD 4806	22,06,48	HL is stored in memory
451B		MOV L,C	69	Move ‘C’ register data to ‘L’
451C		MOV H,B	60	Move ‘B’ register data to ‘H’
451D		SHLD 4804	22,04,48	Store the result in ‘HL’ pair
4520		HLT	76	Stop the program

7.Result:

Input

Input Address	Value
4800	04
4801	00
4802	02
4803	00

Output

Output Address	Value
4804	02
4805	00
4806	FE
4807	FF

8. Assignment:

Write an assembly language code fordivision of any two 16 bit numbers by the advice of instructor lab.

Write a program to divide 0022and 05EE.

9. Conclusion:

In this experiment, the motive was to introduce concept of theory and programming Binary Arithmetic division.

At the end, student should try to learn as much as they can from this experiment and should try problems in microprocessor.

EXPERIMENT NO.:05(b)

Division of two 8 – bit numbers

Contents **Page No.**

1. Objectives :……………………………………………………………….56
2. Expected outcomes of Experiment :……………………………………...56
3. Theory :…………………………………………………………………...56
4. EquipmentsRequired :…………………………………………………....56
5. Procedure : ………………………………………………………………56
6. Coding : …………………………………………………………………58
7. Results : …………………………………………………………………58
8. Assignments : …………………………………………………………..…59
9. Conclusions : ……………………………………………………………59

1. Objective:

To write an assembly language program for dividing two 8 bit numbers using microprocessor kit.

2.Expected Outcomes of Experiment:

Understand the architecture, instruction set and programming of 8085 microprocessor.

An in-depth knowledge of applying the concepts on binary division.

3. Theory:

A division algorithm can be characterized as follows:

Dividend = quotient ×divisor + remainder

To handle signed binary number division, we first convert both the dividend and the divisor to positive numbers to perform the division, and then correct the signs of the results as needed. We adopt a convention that the remainder and the dividend shall have the same sign. That is, if the dividend is positive, then the remainder will be positive. If the dividend is negative, then the remainder will be negative. As for the quotient, it will be positive if the divisor and the dividend have the same sign. Otherwise, it will be negative. Here are some examples that illustrate these conventions:

0111 ÷0011 = 0010 R 0001 (7 ÷3 = 2 reminder = 1)
0111 ÷1101 = 1110 R 0001 (7 ÷(−3)= −2 reminder = 1)
1001 ÷0011 = 1110 R 1111 (−7 ÷3 = −2 reminder = −1)
1001 ÷1101 = 0010 R 1111 (−7 ÷−3 = 2 reminder = −1)

4. Equipments Required:

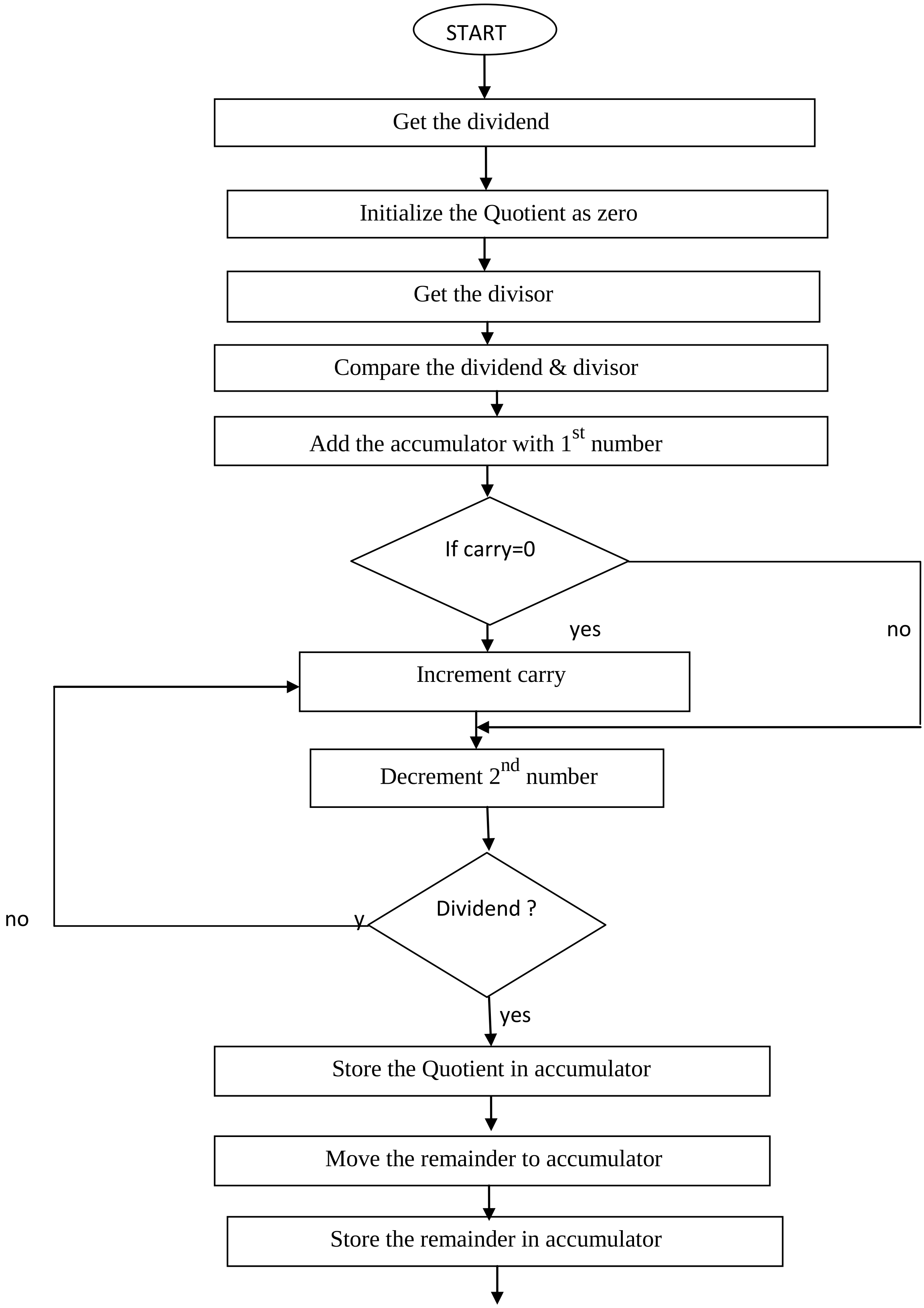
8085 microprocessor kit.
(0-5V) DC battery

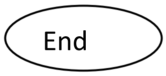
5. Procedure:

Algorithm:

Step1	:	Start the microprocessor
Step2	:	Initialize the Quotient as zero
Step3	:	Load the 1^{st} 8 bit data
Step4	:	Copy the contents of accumulator into register 'B'
Step5	:	Load the 2^{nd} 8 bit data
Step6	:	Compare both the values
Step7	:	Jump if divisor is greater than dividend
Step8	:	Subtract the dividend value by divisor value
Step9	:	Increment Quotient
Step10	:	Jump to step 7, till the dividend becomes zero

Step 11	:	Store the result (Quotient) value in accumulator
Step 12	:	Move the remainder value to accumulator
Step 13	:	Store the result in accumulator
Step 14	:	Stop the program execution





6. Coding:

Address	Label	Mnemonics	Hex Code	Comments
4100		MVI C, 00	0E, 00	Initialize Quotient as zero
4102		LDA, 4500	3A 00, 45	Get the 1^{st} data
4105		MOV B,A	47	Copy the 1^{st} data into register ‘B’
4106		LDA, 4501	3A 01, 45	Get the 2^{nd} data
4109		CMP B	B8	Compare the 2 values
410A		JC (LDP)	DA 12,41	Jump if dividend lesser than divisor
410D	Loop 2	SUB B	90	Subtract the 1^{st} value by 2^{nd} value
410E		INR C	0C	Increment Quotient (410D)
410F		JMP (LDP, 41)	C3, 0D, 41	Jump to Loop 1 till the value of dividend becomes zero
4112	Loop 1	STA 4502	32 02,45	Store The value in Accumulator
4115		MOV A,C	79	Move the value of remainder to accumulator
4116		STA 4503	32 03,45	Store the remainder value in Accumulator
4119		HLT	76	Stop the program execution

7. Results:

Input

Input Address	Value
4500	09
4501	02

Output

Output Address	Value
4502	04 (quotient)
4503	01 (reminder)

1001
0010 – I

0111
0010 – II

0101
0010 – III

0011
0010 – IV

0001 – carry

Quotient	- 04
Carry	- 01

8.Assignments:

Pick the example from textbook to multiply two 16 bit numbers in assembly language.

Write a program to divide two 8-bit numbers 06 by 03.

9. Conclusions:

In this experiment, the motive was to introduce concept of theory and programming Binary Arithmetic 8 bit division and To expertise the concepts of theory and programming of microprocessor.

EXPERIMENT NO.:05(c)

Factorial of 8 Bit Number

Contents **Page No.**

1. Objectives :……………………………………………………………….61

2. Expected outcomes of Experiment :……………………………………...61

3. Theory :…………………………………………………………………...61

4. Equipments Required :……………………………………………………61

5. Procedure : ……………………………………………………………….61

6. Coding : ………………………………………………………………….62

7. Results : ……………………………………………………………….…63

8. Assignments : ……………………………………………………….…63

9. Conclusions : ……………………………………………………………..63

1. Objective:

To write an program to calculate the factorial of a number (between 0 to 8).

2. Expected Outcomes of Experiment:

Understand the architecture, instruction set and programming of 8085 microprocessor.

An in-depth knowledge of applying the concepts on factorial of a number.

3. Theory:

The factorial, symbolized by an exclamation mark (!), is a quantity defined for all integers greater than or equal to 0.

For an integer *n* greater than or equal to 1, the factorial is the product of all integers less than or equal to *n* but greater than or equal to 1. The factorial value of 0 is defined as equal to 1. The factorial values for negative integers are not defined.

Mathematically, the formula for the factorial is as follows. If *n* is an integer greater than or equal to 1, then

$n\,! = n\,(\,n - 1)(\,n - 2)(\,n - 3)\ ...\ (3)(2)(1)$

If $p = 0$, then $p\,! = 1$ by convention.

The factorial is of interest to number theorists. It often arises in probability calculations, particularly those involving combinations and permutations. The factorial also arises occasionally in calculus and physics.

4. Equipments Required:

8085 microprocessor kit.
(0-5V) DC battery

5. Procedure:

Algorithm:

Step 1	:	Initialize the stack pointer
Step 2	:	Get the number in accumulator
Step 3	:	Check for if the number is greater than 1. If no store the result Otherwise go to next step.
Step 4	:	Load the counter and initialize result
Step 5	:	Now factorial program in sub-routine is called.
Step 6	:	In factorial,

		Initialize H_L RP with 0.
		Move the count value to B
		Add H_L content with R_p.
		Decrement count (for multiplication)
Step 7	:	Exchange content of Rp (DE) with HL.
Step 8	:	Decrement counter (for factorial) till zero flag is set.
Step 9	:	Store the result
Step 10	:	Halt

Memory address	Content
4250	05
4251	(120_{10})

6. Coding:

Memory Location	Hex Code	Label	Mnemonics		Comments
			Op code	Operand	
4200 4201 4202	3A 50 42		LDA	4250	Get the Number In accumulator
4203 4204	FE 02		CPI	02H	Compare data with 2 and check it is greater than 1
4205 4206 4207	DA 17 42		JC	Loop 1	If cy =1 jump to loop 1 If cy = 0 proceed
4208	5F		MOV	E,A	Move content of A to E
4209 420A	16 00		MVI	D,00	Load this term as a Resul t
420B	3D		DCR	A	Decrement accumulator by 1
420C	4F		MOV	C,A	Move ‘A’ content To ‘C’ (counter 1 less than A)
420D 420E 420F	CD 00 46		CALL	Facto	Call sub Routine programe Facto
4210	EB		XCHG		Exchange (DE) – (HL)
4211 4212 4213	22 51 42		SHLD	4251	Store content of HL in specified Memory location
4214 4215	C3 1D		JMP	Loop 3	Jump to Loop 3

4216	42				
4217 4218 4219	21 00 01	Loop 1	LXI	$H,0001_H$	HL is loaded with data 01
421A 421B 421C	22 51 42		SHLD	4251	Store the Result In memory
421D	76	Loop 3	HLT		Terminate the program
Sub Routine					
4600 4601 4602	21 00 00	Facto	LXI	H,0000	Initialize HL pair
4603	41		MOV	B,C	Content of ‘C’ Is moved to B
4604	19	Loop 2	DAD	D	Content of DE is added with HL
4605	05		DCR	B	‘B’ is decremented
4606 4607 4608	C2 04 46		JNZ	Loop 2	Multiply by successive addition till zero flag is set

7. Results:

Memory address	Content
4250	04
4251	18

1 x 2 x 3 x 4 = 24 Hexadecimal

16 24
1-8

8.Assignments:

Pick the example from textbook to find factorial of 8 bit number in assembly language.

Write a program to find factorial of 10.

9. Conclusions:

In this experiment, the motive was to introduce concept of theory and programming to find factorial and to expertise the concepts of theory and programming of microprocessor.

EXPERIMENT NO.:06(a)

Separation of hexadecimal number into two digits

Contents **Page No.**

1.Objectives :…………………………………………………………….65

2.Expected outcomes of Experiment :……………………………………...65

3.Theory :…………………………………………………………………...65

4.Equipments Required :……………………………………………..…..65

5.Procedure : ………………………………………………………………65

6.Coding : …………………………………………………………………..66

7.Results : …………………………………………………………………..67

8.Assignments : ……………………………………………………………67

9.Conclusions : ……………………………………………………………..67

1. Objective:

Separation of hexadecimal number into two digits.

2. Expected Outcomes of Experiment:

Design circuits for various applications using microcontrollers

An in-depth knowledge of applying the concepts on addition of two 8 bit numbers neglecting carry generated.

3. Theory:

In mathematics and computing, **hexadecimal** (also **base16**, or **hex**) is a positionalnumeral system with a radix, or base, of 16. It uses sixteen distinct symbols, most often the symbols **0**–**9** to represent values zero to nine, and **A**, **B**, **C**, **D**, **E**, **F** (or alternatively **a**, **b**, **c**, **d**, **e**, **f**) to represent values ten to fifteen.

Hexadecimal numerals are widely used by computer system designers and programmers. As each hexadecimal digit represents four binary digits (bits), it allows a more human-friendly representation of binary-coded values. One hexadecimal digit represents a nibble (4 bits), which is half of an octet or byte (8 bits). For example, a single byte can have values ranging from 00000000 to 11111111 in binary form, but this may be more conveniently represented as 00 to FF in hexadecimal.

In a non-programming context, a subscript is typically used to give the radix, for example the decimal value 10,995 would be expressed in hexadecimal as $2AF3_{16}$. Several notations are used to support hexadecimal representation of constants in programming languages, usually involving a prefix or suffix. The prefix "0x" is used in C and related languages, where this value might be denoted as 0x2AF3.

4. Equipments Required:

8085 microprocessor kit.

(0-5V) DC battery.

5. Procedure:

Algorithm:

Step 1 : Load the byte into acc.

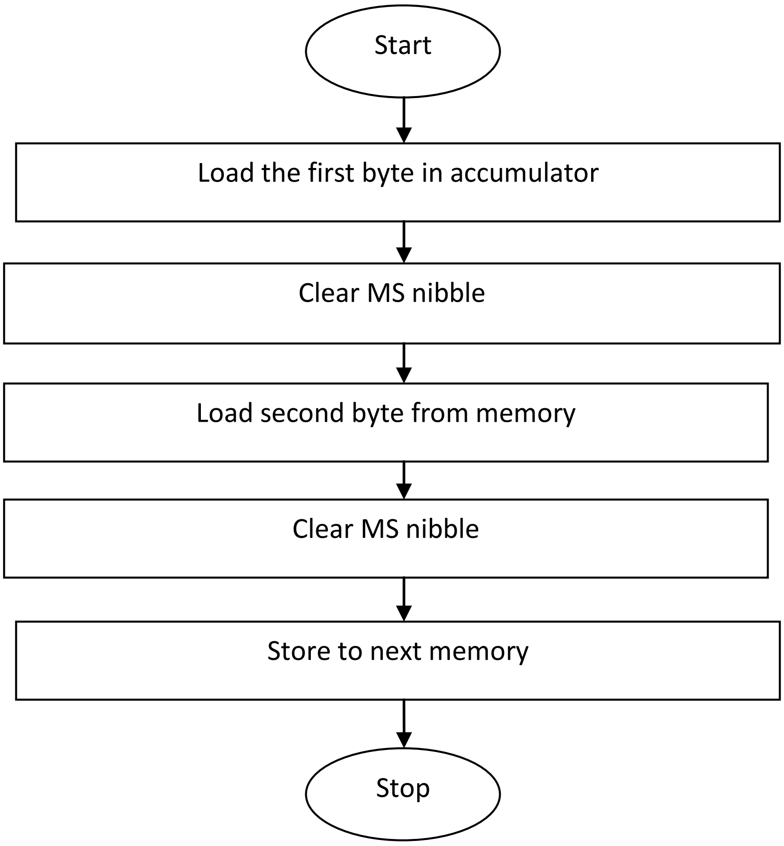
Step 2 : Clear the MS nibble and store it at 2101.

Step 3 : Load the byte from 2100.

Step 4 : Clear the LS nibble and store it at 2102.

Step 5 : Go back to monitor.

Step 6 : Execute from 2000.



6. Coding:

Address	Label	Mnemonics	Hex Code	Comments
2000	Start	LXI H, 2100	21 00 21	Point to first no.
2003		MOV B,M	46	Load count into B register
2004		XRA A	AF	Clear A register
2005	Loop	INX H	23	Point to 1st number
2006		ADD M	86	Add memory to total
2007		DCR B	05	Subtract from count
2008		JMC LOOP	C2 05 20	Test to see if done
200B		STA 2100	32 00 21	Save the result
200E		RST 5	EF	

7. Results:

Memory Address	Data
2100	AF
2101	00
2102	00

After the program executes:

2100	AF
2101	0F
2102	A0

8. Assignments:

Pick a hexadecimal number from textbook and separate it into two digits.

Write a program to separate AB into two digits.

9. Conclusions:

This experiment introduces concept of theory and programming of separation of a hexadecimal number into two digits.Student should try the mentioned operations and implement them.

EXPERIMENT NO.:06(b)

Check the parity of hex numbers

Contents **Page No.**

1.Objectives :…………………………………………………………………..69

2.Expected outcomes of Experiment :………………………………………69

3.Theory :……………………………………………………………………...69

4.Equipments Required :……………………………………………….…69

5.Procedure : ………………………………………………………………..69

6.Coding : …………………………………………………………………….…70

7.Results : ……………………………………………………………………..71

8.Assignments : ………………………………………………………………71

9.Conclusions : ……………………………………………………………..71

1. Objective:

Check the parity of hex numbers.

2. Expected Outcomes of Experiment:

An in-depth knowledge of applying the concepts to check the parity of hex numbers.

3. Theory:

Parity: Parity of a number refers to whether it contains an odd or even number of 1-bits. The number has “odd parity”, if it contains odd number of 1-bits and is “even parity” if it contains even number of 1-bits. Main idea of the below solution is – Loop while n is not 0 and in loop unset one of the set bits and invert parity.

4. Equipments Required:

8085 microprocessor kit.
(0-5V) DC battery.

5. Procedure:

Algorithm:

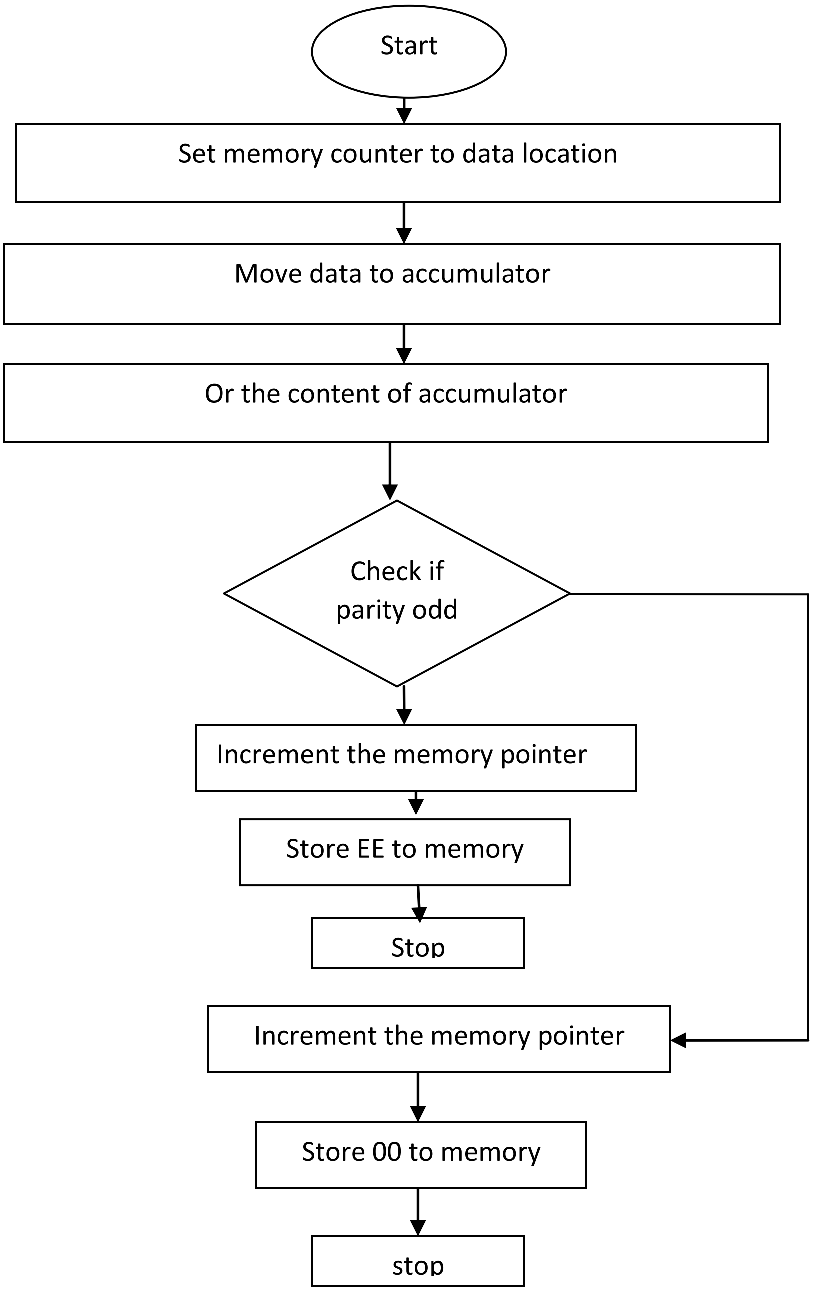
Step 1 : Set the memory counter to the data location 2010 and bring its contents to accumulator.

Step 2 : OR the contents of accumulator with itself,

Step 3 : Check the parity register for odd or even parity and store 00 or EE in location 2011 depending upon whether it is odd or even parity.

Step 4 : Go back to monitor.

Step 5 : Execute from 2000.



6. Coding:

Address	Label	Mnemonic	Hex code	Comment
2000		LXI H,2010	21 10 20	Set the memory counter.
2003		MOV A,M	7E	Get the first no. in accumulator.
2004		ORA A	B7	Set the flag.
2005		JPO ODD	E2 0C 20	If the parity is odd, jump to

				ODD.
2008		INR L	2C	Point to memory location for result.
2009		MVI M,EE	36 EE	Store EE in 2011.
200B		RST 5	EF	Go to monitor.
200C	ODD	INR L	2C	Point to memory location for result.
200D		MVI M,00	36 00	Store 00 in 2011.
200F		RST 5	EF	Go to monitor.
2010			DATA	
2011			RESULT	

7. Results:

Output:

Memory location	Data
2010	10
2010	30

8.Assignments:

Pick a hexadecimal number and check its parity.

Write a program to check the parity of 1F.

9. Conclusions:

This experiment introduces concept of theory and programming of to check the parity of hexadecimal number.

EXPERIMENT NO.:07

Speed Control of Stepper Motor

Contents **Page No.**

1.Objective :…………………………………………………………………...73

2.Expected outcomes of Experiment :……………………………………….73

3.Theory :……………………………………………………………….....73

4.Equipments Required :…………………………………………………….…74

5.Procedure : ……………………………………………………………….….74

6.Coding : ……………………………………………………………….….75

7.Results : ………………………………………………………………..75

8.Assignments : ……………………………………………………….75

9.Conclusions : …………………………………………………………..75

1. Objective:

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

2. Expected Outcomes of Experiment:

An in-depth knowledge of applying the concepts to run a stepper motor in forward and reverse direction.

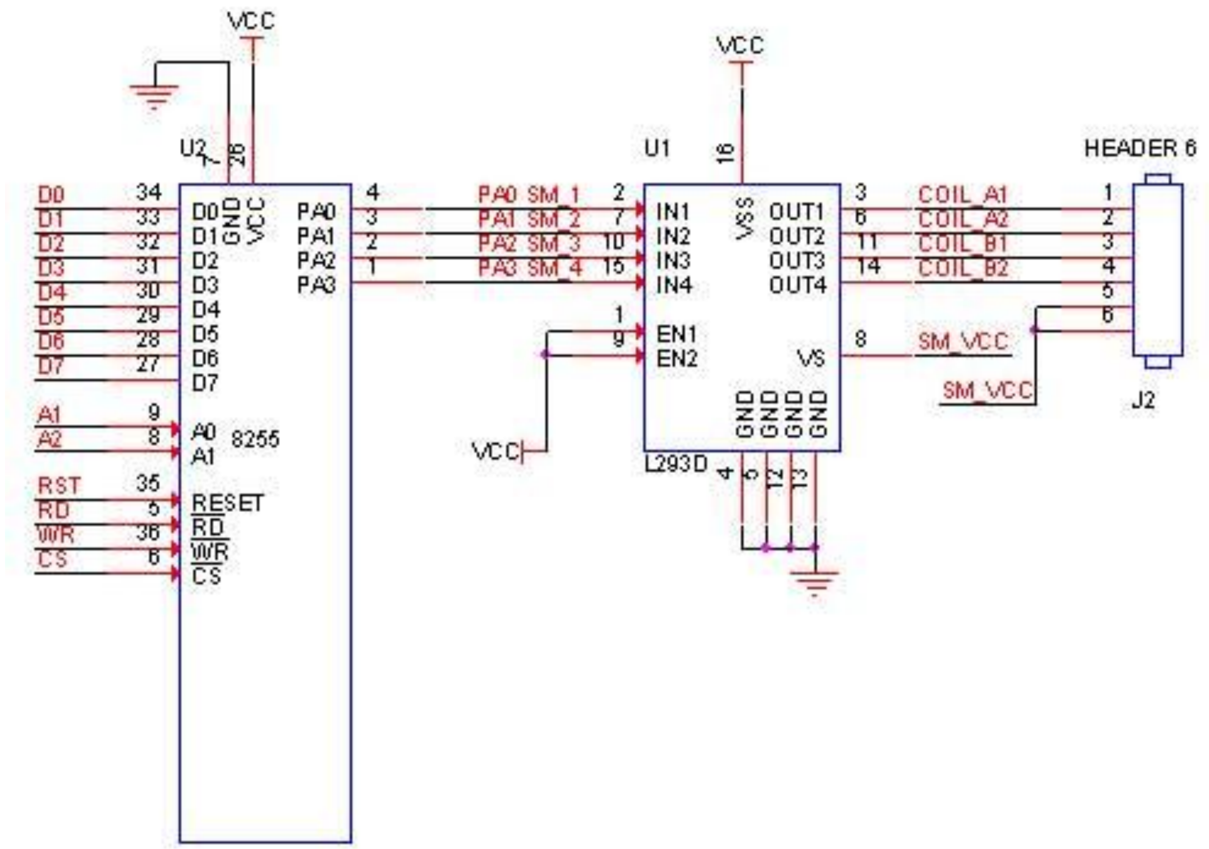
3. Theory:

STEPPER MOTOR

A **stepper motor** is a brushless, synchronous electric motor that converts digital pulses into mechanical shaft rotation. Every revolution of the **stepper motor** is divided into a discrete number of steps, and the motor must be sent a separate pulse for each step.

INTERFACING STEPPER MOTOR WITH 8085

We now want to control a **stepper motor** in 8085 trainer kit. It works by turning ON & OFF a four I/O port lines generating at a particular frequency.



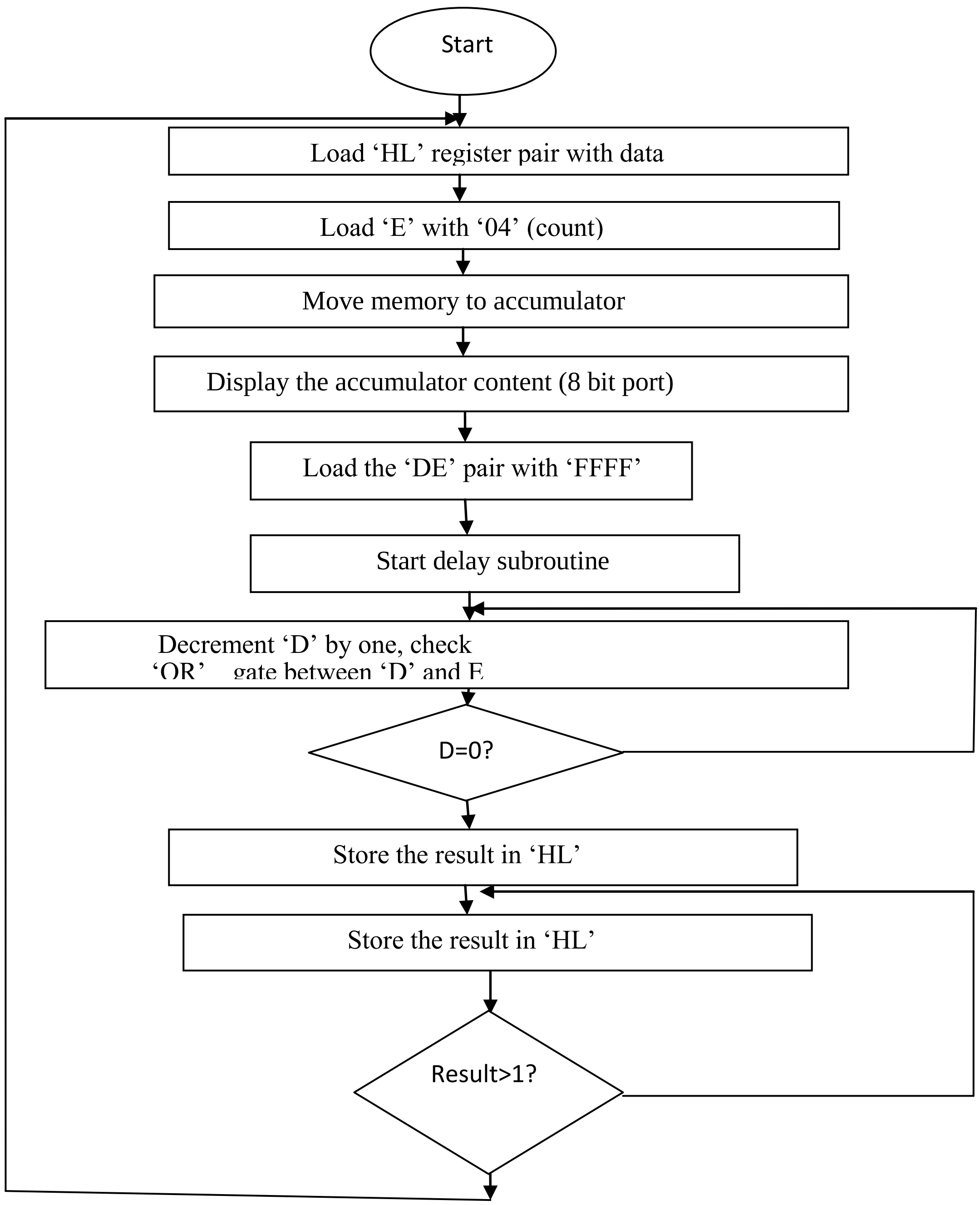
4. Equipments Required:

8085 microprocessor kit.
(0-5V) DC battery.
Stepper motor

5. Procedure:

Algorithm:

Step 1	:	Load the 'HL' pair wit value from table
Step 2	:	Move it to 'B' register for setting the counter
Step 3	:	Move the memory value to accumulator and display it by control word
Step 4	:	Load 'DE' register pair with FFFF for starting delay subroutine
Step 5	:	Run the delay loop control D-register becomes zero.
Step 6	:	Increment 'H' address for next value from table
Step 7	:	Jump on no zero
Step 8	:	When B = 0, go to start and restart the program.



6. Coding:

Memory Location	Hex Code	Label	Mnemonics		Comments
			Op code	Operand	
4100	Start	LXI	H,Look up	21,1A,41	Load the 'HL' with Data
4103		MVI	B,04	06,04	B = 04
4105	Repeat	MOV	A,M	7E	Memory value to 'A'
4106		OUT	C0	D3, C0	Display it
4108		LXI	D,03,03	11	Load 'DE' with FFFF
410B	Delay	NOP		00	Start delay loop
410C		DCX	D	1B	Decrement DE by 1
410D		MOV	A,E	7B	Move 'E' to 'A'
410E		ORA	D	B2	Check De = 0 or not
410F		JNZ	DELAY	C2, 0B,41	Jump on zero
4112		INX	H	23	Increment HL by 1
4113		DCR	B	05	Decrement B by 1
4114		JNZ	Repeat	C2,05,41	Jump on no zero
4117		JMP	START	C3,00,41	Jump to start

7. Result:

Input

Input Address	Value
411A	0A
411B	06
411C	05
411D	09

Reverse Direction

Output Address	Value
411A	09
411B	05
411C	06
411D	0A

8.Assignments:

Study about stepper motor in detail.

9. Conclusions:

This experiment introduces concept of theory and programming to run a motor in forward and reverse direction.

EXPERIMENT NO.:08(a)

1's complement of an 8 bit number

Contents **Page No.**

1.Objective :………..……………………………………………………………..77

2.Expected outcomes of Experiment :………………………………………77

3.Theory :……………………………………………………………………...77

4.Equipments Required :…………………………………………………….77

5.Procedure : ……………………………………………………………..77

6.Coding : ………………………………………………………………..78

7.Results : ……………………………………………………………...78

8.Assignments : …………………………………………………………..79

9.Conclusions : …………………………………………………………..79

1. Objective:

Write a program to find 1’s complement of an 8 bit number.

2. Expected Outcomes of Experiment:

An in-depth knowledge of applying the concepts of finding 1’s compliment of an 8 bit number.

3. Theory:

The **ones' complement** of a binary number is defined as the value obtained by inverting all the bits in the binary representation of the number (swapping 0s for 1s and vice versa). The ones' complement of the number then behaves like the negative of the original number in some arithmetic operations. To within a constant (of −1), the ones' complement behaves like the negative of the original number with binary addition. However, unlike two's complement, these numbers have not seen widespread use because of issues such as the offset of −1, that negating zero results in a distinct negative zero bit pattern, less simplicity with arithmetic borrowing, etc.

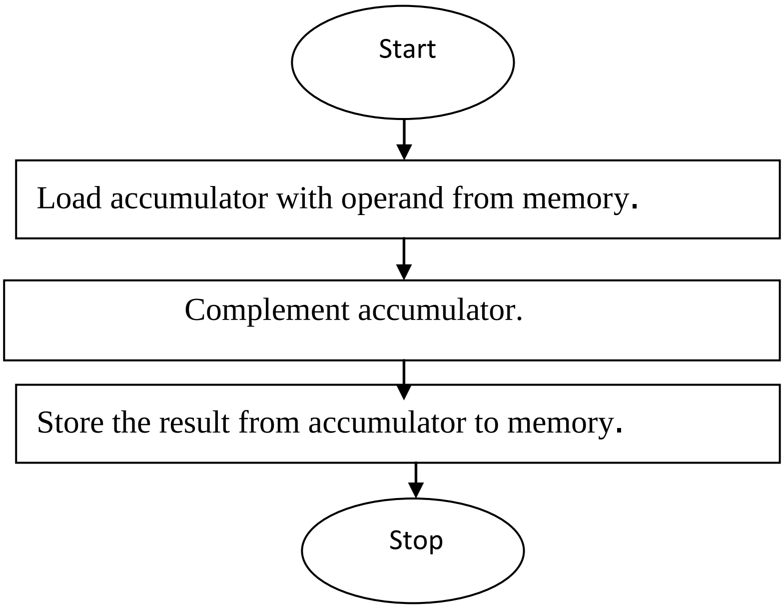
A **ones' complement system** or **ones' complement arithmetic** is a system in which negative numbers are represented by the inverse of the binary representations of their corresponding positive numbers. In such a system, a number is negated (converted from positive to negative or vice versa) by computing its ones' complement. An N-bit ones' complement numeral system can only represent integers in the range $-(2^{N-1}-1)$ to $2^{N-1}-1$ while two's complement can express -2^{N-1} to $2^{N-1}-1$.

4. Equipments Required:

8085 microprocessor kit.
(0-5V) DC battery.

5. Procedure:

1. This program finds the 1’s complement of an 8-bit number stored in memory location 3000H.
2. Let us assume that the operand stored at memory location 3000H is 85H.
3. The operand is moved to accumulator from memory location 3000H.
4. Then, its complement is found by using CMA instruction.
5. The result is stored at memory location 3001H.



6. Coding:

Address	Mnemonics	Operand	Opcode	comments
2000	LDA	3000H	3A	Load H-L pair with data from 3000H.
2001			00	Lower-order of 3000H.
2002			30	Higher-order of 3000H.
2003	CMA		2F	Complement accumulator.
2004	STA	3001H	32	Store the result at memory location 3001H.
2005			01	Lower-order of 3001H.
2006			30	Higher-order of 3001H.
2007	HALT		76	Halt.

7. Result:

Output:

Before Execution:
Memory Address Data

3000H: 85H

After Execution:

Memory address Data
3001H: 7AH

8.Assignments:

Take an eight bit number and find out its 1’s compliment.

Find out 1’s compliment of 11001001.

9. Conclusions:

This experiment introduces an alternative approach to signed magnitude in one’s complement where the first bit is again a sign bit.

EXPERIMENT NO.:08(b)

2's complement of an 8 bit number

Contents **Page No.**

1.Objective :……………………………………………………………….81

2.Expected outcomes of Experiment :……………………………………..81

3.Theory :…………………………………………………………………...81

4.Equipments Required :…………………………………………………...81

5.Procedure : ……………………………………………………………….81

6.Coding : ………………………………………………………………….82

7.Results : ………………………………………………………………….83

8.Assignments : ……………………………………………………………83

9.Conclusions : ……………………………………………………………83

1. Objective:

Write a program to find 2's complement of an 8 bit number.

2. Expected Outcomes of Experiment:

An in-depth knowledge of applying the concepts of finding 2's compliment of an 8 bit number.

Simplified the binary subtraction.

3. Theory:

Two's complement is a mathematical operation on binary numbers, as well as a binary signed number representation based on this operation. Its wide use in computing makes it the most important example of a radix complement.

The two's complement of an *N*-bit number is defined as the complement with respect to 2^N; This is also equivalent to taking the ones' complement and then adding one, since the sum of a number and its ones' complement is all 1 bits. The two's complement of a number behaves like the negative of the original number in most arithmetic, and positive and negative numbers can coexist in a natural way.

In two's-complement representation, positive numbers are simply represented as themselves, and negative numbers are represented by the two's complement of their absolute value; two tables on the right provide examples for $N = 3$ and $N = 8$. In general, negation (reversing the sign) is performed by taking the two's complement. This system is the most common method of representing signed integers on computers. An *N*-bit two's-complement numeral system can represent every integer in the range $-(2^{N-1})$ to $+(2^{N-1} - 1)$ while ones' complement can only represent integers in the range $-(2^{N-1} - 1)$ to $+(2^{N-1} - 1)$. The sum of a number and its two's complement will always equal 0 (since the last digit is truncated), and the sum of a number and its one's complement will always equal -0.

4. Equipments Required:

8085 microprocessor kit.

(0-5V) DC battery.

5. Procedure:

1. This program finds the 2's complement of an 8-bit number stored in memory location 3000H.

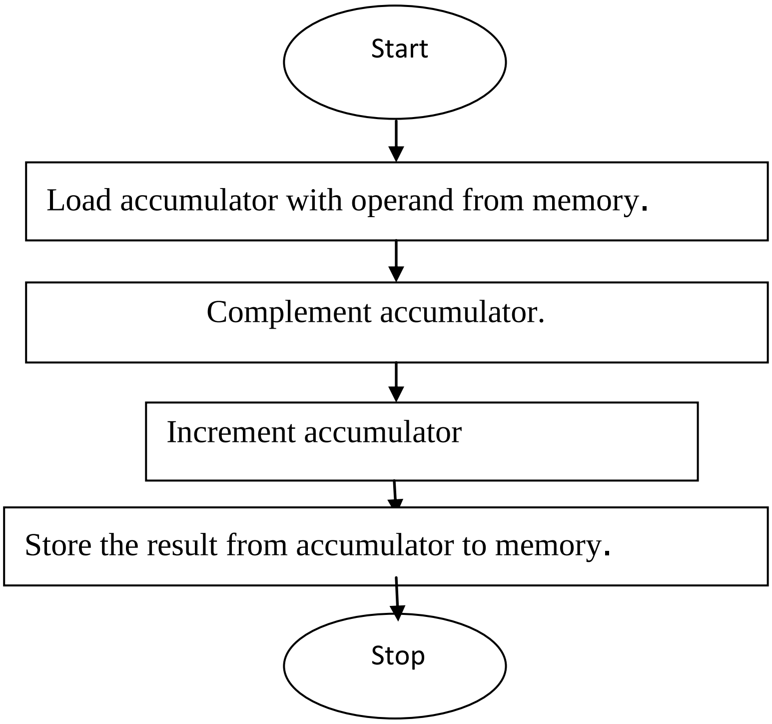
2. Let us assume that the operand stored at memory location 3000H is 85H.

3. The operand is moved to accumulator from memory location 3000H.

4. Then, its complement is found by using CMA instruction.

5. One is added to accumulator by incrementing it to find its 2’s complement.

6. The result is stored at memory location 3001H.



6. Coding:

Address	Mnemonics	Operand	Opcode	comments
2000	LDA	3000H	3A	Load H-L pair with data from 3000H.
2001			00	Lower-order of 3000H.
2002			30	Higher-order of 3000H.
2003	CMA		2F	Complement accumulator.
2004	INR	A	2C	Increment accumulator.
2005	STA	3001H	32	Store the result at memory location 3001H.
2006			01	Lower-order of 3001H.
2007			30	Higher-order of 3001H.
2008	HALT		76	Halt

7. Result:

Output:

Before Execution:

Memory address	Data
3000H:	85H

After Execution:

Memory address	Data
3001H:	7BH

8.Assignments:

Write a program to find 2’s compliment of 35_{10}.

9. Conclusions:

This experiment simplified Binary subtraction. It helped students in applying such operation in real time scenario on arithmetic.

Student should try the mentioned operations and implement them on 2’s compliment.

EXPERIMENT NO.:09(a)

Generation of square wave

Contents **Page No.**

1.Objective :……………………………………………..……………..85

2.Expected outcomes of Experiment :………………………………………85

3.Theory :…………………………………………………………………...85

4.Equipments Required :……………………………………………………85

5.Procedure : ………………………………………………………………85

6.Coding : ……………………………………………………………………85

7.Results : ……………………………………………………………………86

8.Assignments : ……………………………………………………………86

9.Conclusions : ……………………………………………………………86

1. Objective:

Write a program to generate square wave of 1 KHZ frequency using 8085 microprocessor kit.

2. Expected Outcomes of Experiment:

Students acquire knowledge about assembly language programming to generate square wave of 1 KHZ frequency using 8085 microprocessor kit.

3. Theory:

A square wave is a non-sinusoidal periodic waveform in which the amplitude alternates at a steady frequency between fixed minimum and maximum values, with the same duration at minimum and maximum.

The DAC 0800 is a monolithic 8 bit high speed current output digital to analog converters featuring setting time of 100nSEC. It also features high compliance complementary current outputs to allow differential output voltage of 20 Vp-p with simple resistor load and it can be operated both in unipolar and bipolar mode.

4. Equipments Required:

8085 microprocessor kit.
(0-5V) DC battery.

5. Procedure:

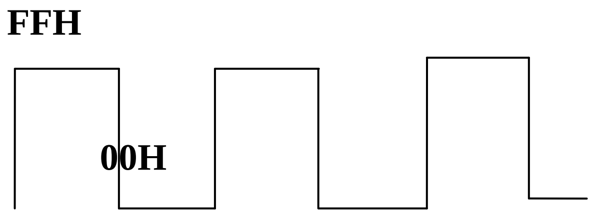
1. Load the initial value (00) to the accumulator and move it to DAC.
2. Call the delay program.
3. Load the final value (FF) to the accumulator and move it to DAC.
4. Call the delay program.
5. Repeat steps 2 to 5.

6. Coding:

LABEL	OPCODE	OPERAND	COMMENT
	LXI	SP,2000H	Load stack pointer with immediate data of 2000H.
	MVI	A,80H	Load A register with immediate data 80H.
	OUT	83H	Send content of A register to control word register.
REPEAT:	MVI	A,FFH	Load max. Count into A register.
	OUT	80H	Send data of A register to register to port A Of 8155.
	CALL	DELAY	Go to delay subroutine.

	MVI	A,00H	Load Min. count into A register.
	OUT	80H	Send data of A register to port A of 8155.
	CALL	DELAY	Go to delay subroutine.
	JMP	REPEAT	Jump to REPEAT to generate square wave continuously.
DELAY	MVI	B,"DELAY COUNT"(B,0FH)	Load delay count value into B register.
LOOP-1	DCR	B	Decrement delay count value.
	JNZ	LOOP-1	Jump to loop-1 if delay count value is not equal to zero.
	RET		End of subroutine and return back to main program.

7. Result:



8. Assignments:

Write a program to generate square wave of 2 KHZ frequency using 8085 microprocessor kit.

9. Conclusion:

The square wave was generated using 8085 microprocessor kit.

EXPERIMENT NO.:09(b)

Generation of triangular wave

Contents **Page No.**

1.Objective:…………………………………….………………………88

2.Expected outcomes of Experiment :…… …………………………88

3.Theory :…………………………………………………………..…88

4.Equipments Required :………………………………………………..88

5.Procedure : ……………………………………………………….....89

6.Coding : ………………………………………………………………89

7.Results : ………………………………………………………………90

8.Assignments : ……………………………………………………...90

9.Conclusions : ……………………………………………………...90

1. Objective:

Write a program to generate triangular wave of 1 KHZ frequency using 8085 microprocessor kit.

2. Expected Outcomes of Experiment:

Students acquire knowledge about assembly language programming to generate triangular wave of 1 KHZ frequency using 8085 microprocessor kit.

3. Theory:

Triangular wave is a periodic, non-sinusoidal waveform with a triangular shape. The most important feature of a triangular wave is that it has equal rise and fall times.

DAC 0800 is an 8–bit DAC and the output voltage variation is between–5V and +5V. The output voltage varies in steps of 10/256=0.04. The digital data input and the corresponding output voltages are presented-

Input Data in HEX	Output Voltage
00	-5.00
01	-4.96
02	-4.92
…	…
7F	0.00
…	…
FF	5.00

Referring to Table1, with 00H as input to DAC, the analog output is -5V. Similarly, with FFH as input, the output is +5V. Outputting Digital data 00 and FF at regular intervals to DAC.

4. Equipments Required:

8085 microprocessor kit.
(0-5V) DC battery.

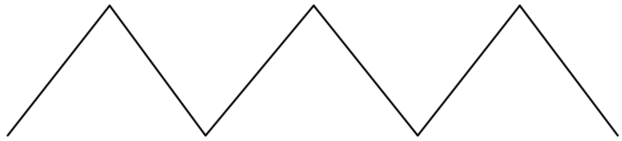
5. Procedure:

1. Load the initial value (00) to accumulator.
2. Move the accumulator content to DAC.
3. Increment the accumulator content by 1.
4. If the accumulator content is zero proceed to next step else go to step 3.
5. Load value (FF) to the accumulator.
6. Move the accumulator content to DAC.
7. Decrement the accumulator content by 1.
8. If the accumulator content is zero proceed to step 1 else go to step 6.

6. Coding:

LABEL	OPCODE	OPERAND	COMMENT
	MVI	A,80H	Load A register with immediate data 80H.
	OUT	83H	Send content of A register to control word register.
LOOP-3	MVI	A,00H	Load initial count.
	OUT	80H	Send count value to port A of 8155.
LOOP-1	INR	A	Increment count value.
	OUT	80H	Send count value to port A of 8155.
	CPI	FFH	Compare content of A register with max. count
	JNZ	LOOP-1	Jump to LOOP-1 if result of comparison is not equal to zero.
LOOP-2	DEC	A	Decrement count value.
	OUT	80H	Send count value to port A of 8155.
	CPI	00H	Compare content of A register with initial count.
	JNZ	LOOP-2	Jump to LOOP-2 if result of comparison is not equal to zero.
	JMP	LOOP-3	Jump to LOOP-3 to repeat process.

7. Results:



8. Assignments:

Write a program to generate triangular wave of 2 KHZ frequency using 8085 microprocessor kit.

9. Conclusion:

The triangular wave was generated using 8085 microprocessor kit.

EXPERIMENT NO.:09(c)

Generation of saw tooth wave

Contents Page No.

1.Objective:…………………………………..………………………….92

2.Expected outcomes of Experiment :………………………………….92

3.Theory :……………………………………………………………..92

4.Equipments Required :……………………………………………...92

5.Procedure : ………………………………………………………….92

6.Coding : …………………………………………………………….92

7.Results : …………………………………………………………….93

8.Assignments : ………………………………………………………93

9.Conclusions : ……………………………………………………….93

1.Objective:

Write a program to generate saw tooth wave of 1 KHZ frequency using 8085 microprocessor kit.

2. Expected Outcomes of Experiment:

Students acquire knowledge about assembly language programming to generate triangular wave of 1 KHZ frequency using 8085 microprocessor kit.

3. Theory:

A linear, non-sinusoidal, triangular shape waveform represents a sawtooth waveform in which fall time and rise time are different. A linear, non-sinusoidal, triangular shape waveform represents a pure triangular waveform in which fall time and rise times are equal. The Sawtooth Wave Generator is also known as asymmetric triangular waveform.

4. EquipmentRequired:

8085 microprocessor kit.

(0-5V) DC battery.

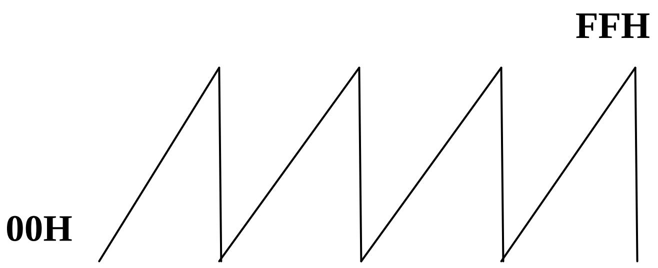
5. Procedure:

1. Load the initial value (00) to accumulator.
2. Move the accumulator content to DAC.
3. Increment the accumulator content by 1.
4. Repeat step 3 and 4.

6. Coding:

LABEL	OPCODE	OPERAND	COMMENT
	MVI	A,80H	Load A register with immediate data 80H.
	OUT	83H	Send content of A register to control word register.
LOOP-2	MVI	A,00H	Load initial count.
	OUT	80H	Send count value to port A of 8155.
LOOP-1	INR	A	Increment count value.
	OUT	80H	Send count value to port A of 8155.
	CPI	FFH	Compare content of A register with max. count.
	JNZ	LOOP-1	Jump to LOOP-1 if result of comparison is not equal to zero.
	JMP	LOOP-2	Jump to LOOP-2 to repeat process.

7. Results:



8. Assignments:

Write a program to generate triangular wave of 2 KHZ frequency using 8085 microprocessor kit.

9. Conclusion:

The triangular wave was generated using 8085 microprocessor kit.

EXPERIMENT NO.:10

Generation of beep sound on a buzzer

Contents **Page No.**

1.Objective:……………………..……………………………………………...95

2.Expected outcomes of Experiment :…… …………………………...95

3.Equipments Required :…………………………………………………95

4.Coding : ……………………………………………………………….95

5.Results : ……………………………………………………………….96

6.Assignments : …………………………………………………………96

7.Conclusions : ………………………………………………………….96

1.Objective:

To generate beep sound on a buzzer using microcontroller 8051.

2. Expected Outcomes of Experiment:

Differentiatebetween microprocessors & microcontrollers.
Identify the basic element and functions of microcontroller.

3. Equipment Required:

8051 microcontroller kit with buzzer interface .

Software for programming 8051.

4. Coding:

```
#include "REG52.h"
#define buz P1

sbit SW=P3^0;
long int i;
void main()
{
while(1)
{
if (SW==0)
{
for(i=0;i<=90000;i++);
if(SW==0)
{

while(SW==0);
buz=0x01;     // ON Buzzer
for(i=0;i<4500;i++);     // Delay
buz=0x00;      // OFF Buzzer

for(i=0;i<4500;i++);  // Delay
}  }
}  }
```

5. Results:

The Microcontroller 8051 was successfully programmed to generate “beep” sound on buzzer.

6. Assignments:

Build a microcontroller based project & do the programming on your own.

7. Conclusion:

Beep sound was successfully produced using 8051 microcontroller and buzzer.

EXPERIMENT NO.:11

Display name on the LCD display

Contents **Page No.**

1.Objective: ……………………………..…………………………….98

2.Expected outcomes of Experiment :…….. ……………………………98

3.Equipments Required :……………………………………………...98

4.Coding : ………………………………………………………….98

5.Results : ………………………………………………………….99

6.Assignments : ……………………………………………………99

7. Conclusions : …………………………………………………...99

1.Objective:

To display your name on the LCD display of microcontroller 8051.

2. Expected Outcomes of Experiment:

Differentiatebetween microprocessors & microcontrollers.
Developingthe assembly language program & C language program for microcontroller applications.

3. EquipmentRequired:

8051 microcontroller kit with LCD

Software for programming 8051.

4. Coding:

```
#include<reg52.h> //including sfr registers for ports of the controller
#include<lcd.h> //Can be download from bottom of this article

//LCD Module Connections
sbit RS = P0^0;
sbit EN = P0^1;
sbit D4 = P2^4;
sbit D5 = P2^5;
sbit D6 = P2^6;
sbit D7 = P2^7;
//End LCD Module Connections
void Delay(int a)
{
  int j;
  int i;
  for(i=0;i<a;i++)
  {
    for(j=0;j<100;j++)
    {
    }
  }
}
void main()
{
  int i;
  Lcd4_Init();
  while(1)
```

```
{
  Lcd4_Set_Cursor(1,1);
  Lcd4_Write_String("Name");
  for(i=0;i<15;i++)
  {
    Delay(1000);
    Lcd4_Shift_Left();
  }
  for(i=0;i<15;i++)
  {
    Delay(1000);
    Lcd4_Shift_Right();
  }
  Lcd4_Clear();
  Lcd4_Set_Cursor(2,1);
  Lcd4_Write_Char('e');
  Lcd4_Write_Char('S');
  Delay(2000);
 }
}
```

5. Results:

The Microcontroller 8051 was successfully programmed to display a name on the LCD display.

6. Assignments:

Build a microcontroller based project & do the programming on your own.

7. Conclusion:

"Name" was successfully written on LCD of 8051 microcontroller.

EXPERIMENT NO.:12(a)

Addition of 8 bit numbers using microcontroller 8051

Contents **Page no.**

1.Objective :……………………………..…………………….. …101

2.Expected outcomes of Experiment :…………………………... …101

3.Theory :……………………………………………………….....101

4.Equipments Required :………………………………………...101

5.Procedure : ……………………………………………………102

6.Coding : ………………………………………………………102

7.Results : ……………………………………………………….102

8.Assignments : ………………………………………………102

9.Conclusions : ……………………………………………….102

1.Objective:

To write an assembly language for adding two 8 bit numbers by using microcontroller 8051 kit.

2. Expected Outcomes of Experiment:

Understanding of the addition of two eight bit numbers in microcontroller 8051.

3. Theory:

8 bit addition is much like decimal addition except that you are only adding 1s and 0s. When the sum exceeds 1, carry a 1 over to the next-more-significant column.

0 + 0 = 0 carry 0
0 + 1 = 1 carry 0
1 + 0 = 1 carry 0
1 + 1 = 0 carry 1

General form: $A_0 + B_0 = \Sigma_0 + C_{out}$

Where Summation symbol (Σ)

Carry-out (C_{out})

Truth table for addition of two binary digits			
A_0	B_0	$\sum_0$	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

4. Equipments Required:

8051 microcontroller kit.
(0-5V) DC battery.

5. Procedure:

1. Clear C register for carry.

2. Get the data immediately.

3. Add the two data.

4. Store the result in memory pointed by DPTR.

6. Coding:

```
PROGRAM:
ORG 4100
CLR C
MOV A,#data1
ADD A,#data2
MOV DPTR,#4500
MOVX @DPTR,A
HERE: SJMP HERE
```

7. Results:

Input 66, 32

Output 98(4500)

8. Assignments:

Solve examples from textbook for adding two eight bit numbers using 8051 microcontroller.

9. Conclusion:

This experiment introduces concept of theory and programming ofBinary Arithmetic addition. It helped students in applying such operation in real time scenario on arithmetic. Student should try the mentioned operations and implement them on binary addition.

EXPERIMENT NO.:12(b)

Subtraction of 8 bit numbers using microcontroller 8051

Contents **Page no.**

1. Objective :……………………………………………………………..104

2. Expected outcomes of Experiment :……………………………………..104

3. Theory :………………………………………………………………..104

4. EquipmentRequired :……………………..……………………………...104

5. Procedure : ……………………………………………………………105

6. Coding : ………………………………………………………………105

7. Results : ………………………………………………………………105

8. Assignments : ………………………………………………………...105

9. Conclusions : …………………………………………………………105

1.Objective:

To write an assembly language for subtracting two 8 bit numbers by using microcontroller 8051 kit.

2. Expected Outcomes of Experiment:

Understanding of the subtraction of two eight bit numbers in microcontroller 8051.

3. Theory:

8 bit addition is much like decimal addition except that you are only adding 1s and 0s. When the sum exceeds 1, carry a 1 over to the next-more-significant column.

0 - 0 = 0 borrow 0
0 - 1 = 1 borrow 1
1 - 0 = 1 borrow 0
1 - 1 = 0 borrow 0

General form: A_0-B_0 =R_0+ B_{out}

Remainder is R_0

Borrow is B_{out}

Truth table for subtraction of two binary digits			
A_0	B_0	R_0	B_{out}
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

4. Equipments Required:

8051 microcontroller kit.

(0-5V) DC battery.

5. Procedure:

1. Clear C register for carry.

2. Get the data immediately.

3. Subtract the two data.

4. Store the result in memory pointed by DPTR.

6. Coding:

```
PROGRAM:
ORG 4100
CLR C
MOV A,#data1
SUBB A,#data2
MOV DPTR,#4500
MOVX @DPTR,A
HERE: SJMP HERE
```

7. Results:

Input 66, 32

Output 34(4500)

8. Assignments:

Solve examples from textbook for subtracting two eight bit numbers using 8051 microcontroller.

9. Conclusion:

This experiment introduces concept of theory and programming ofBinary Arithmetic subtraction. It helped students in applying such operation in real time scenario on arithmetic. Student should try the mentioned operations and implement them on binary subtraction.

EXPERIMENT NO.:12(c)

Multiplication of 8 bit numbers using microcontroller 8051

Contents **Page no.**

1.Objective :……………………………….……………………………107

2.Expected outcomes of Experiment :……………………………………..107

3.Theory :…………………………………………………………………..107

4.Equipments Required :………………………………………………..107

5.Procedure : …………………………………………………………………107

6.Coding : ……………………………………………………………………107

7.Results : ……………………………………………………………………108

8.Assignments : ………………………………………………………...108

9.Conclusions : …………………………………………………………108

1.Objective:

To write an assembly language for multiplying two 8 bit numbers by using microcontroller 8051 kit.

2. Expected Outcomes of Experiment:

Understanding of the multiplication of two eight bit numbers in microcontroller 8051.

3. Theory:

- Multiply the 2^0 bit of the multiplier times the multiplicand.
- Multiply the 2^1 bit of the multiplier times the multiplicand. Shift the result one position to the left.
- Repeat step 2 for the 2^2 bit of the multiplier, and for all remaining bits.
- Take the sum of the partial products to get the final product.

4. Equipments Required:

8051 microcontroller kit.

(0-5V) DC battery.

5. Procedure:

1. Get the data in A – reg.
2. Get the value to be multiplied in B – reg.
3. Multiply the two data.
4. The higher order of the result is in B – reg.
5. The lower order of the result is in A – reg.
6. Store the results.

6. Coding:

```
ORG 4100
CLR C
MOV A,#data1
MOV B,#data2
MUL AB
MOV DPTR,#4500
MOVX @DPTR,A
INC DPTR
```

```
MOV A,B
MOVX @DPTR,A
HERE: SJMP HERE
```

7. Results:

Input 80 , 80

Output 00(4500) 19(4501)

8. Assignments:

Solve examples from textbook for multiplying two eight bit numbers using 8051 microcontroller.

9. Conclusion:

This experiment introduces concept of theory and programming ofBinary Arithmetic multiplication. It helped students in applying such operation in real time scenario on arithmetic. Student should try the mentioned operations and implement them on binary multiplication.

EXPERIMENT NO.:12(d)
Division of 8 bit numbers using microcontroller 8051

Contents **Page no.**

1.Objective :………………………………………..…………………….110

2.Expected outcomes of Experiment :……………………………………...110

3.Equipment Required :…………..………………………………………...110

4.Procedure : ………………………………………………………………110

5.Coding : …………………………………………………………………110

6.Results : …………………………………………………………………111

7.Assignments : …………………………………………………………...111

8.Conclusions : ……………………………………………………………111

1.Objective:

To write an assembly language for division two 8 bit numbers by using microcontroller 8051 kit.

2. Expected Outcomes of Experiment:

Understanding of the division of two eight bit numbers in microcontroller 8051.

3. Equipment Required:

8051 microcontroller kit.

(0-5V) DC battery.

4. Procedure:

1. Get the data in A – reg.
2. Get the value to be divided in B – reg.
3. Divide the two data.
4. The quotient is in A – reg..
5. The remainder is in B – reg.
6. Store the results.

5. Coding:

```
PROGRAM:
ORG 4100
CLR C
MOV A,#data1
MOV B,#data2
DIV AB
MOV DPTR,#4500
MOVX @DPTR,A
INC DPTR
MOV A,B
MOVX @DPTR,A
HERE: SJMP HERE
```

6. Results:

Input 05 , 03

Output 01(4500) 02(4501)

7. Assignments:

Solve examples from textbook for division two eight bit numbers using 8051 microcontroller.

8. Conclusion:

This experiment introduces concept of theory and programming ofBinary Arithmetic division. It helped students in applying such operation in real time scenario on arithmetic. Student should try the mentioned operations and implement them on binary division.