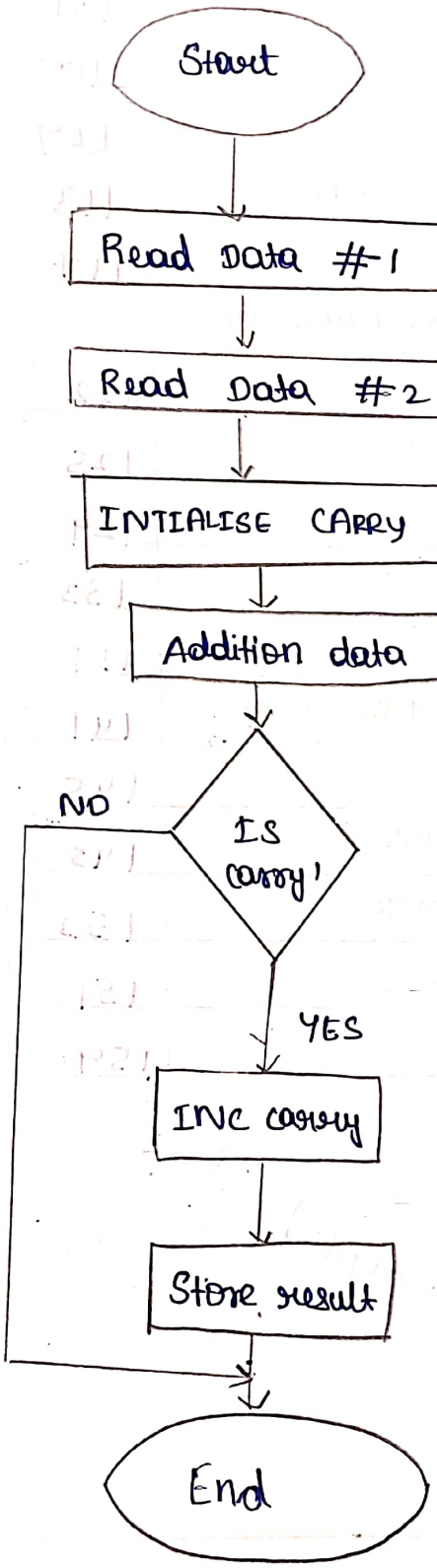
16 Bit addition (8086)



EX NO: 01

DATE: 16/8/21

16 BIT ADDITION USING

AIM:

To write a program for 16 bit addition using 8086 microprocessor

APPARATUS REQURIED:

1.8086 microprocessor kit	1
2. Adaptor	1
3. Opcode sheet	1

ALGORITHM:

1. Start the program
2. Get the first operand from the memory
3. Get the second operand from the memory
4. Add both operand
5. Store the result
6. Stop the program and store it in specific notation
7. Execute the program

WITHOUT CARRY :

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA.
8800	03	8804	04
8801	07	8805	0B
8802	01	8806	00
8803	04		

WITH CARRY :

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8800	04	8804	06
8801	09	8805	17
8802	02	8806	01
8803	08		

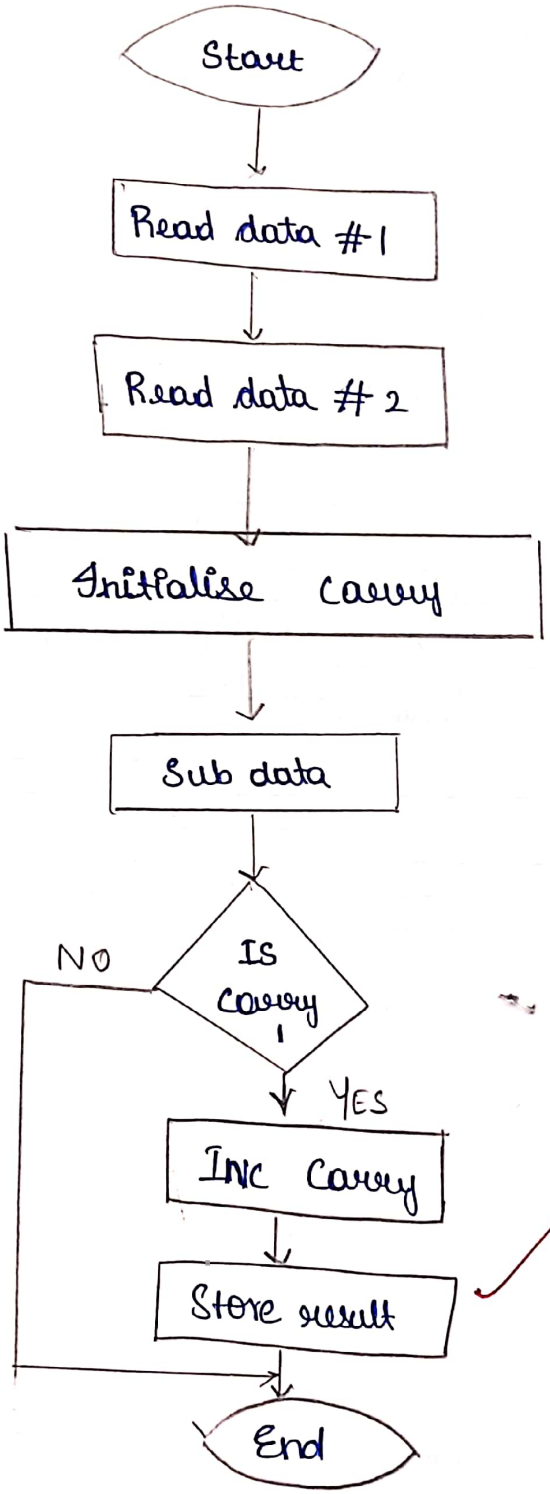
16 BIT ADDITION USING 8086

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	AX,[8800]	8B	MOVE DATA TO AX
8001				06	
8002				00	
8003				88	
8004		MOV	CL,00	B1	CLEAR THE CL REGISTER
8005				00	
8006		MOV	BX,[8802]	8B	MOVE DATA TO BX
8007				1E	
8008				02	
8009				88	
800A		ADD	AX,BX	03	ADD BX TO AX
800B				C3	
800C		MOV	[8804],AX	89	MOVE AX TO DATA
800D				06	
800E				04	
800F				88	
8010		JNC	LOOP	73	
8011				02	
8012		INC	CL	FE	INCREEMENT CL REGISTER
8013				C1	
8014	LOOP	MOV	[8806],CL	88	MOVE CL REGISTER TO DATA
8015				0E	
8016				06	
8017				08	
8018		HLT		F4	END OF THE PROGRAM

RESULT:

Thus the program for 16 bit addition using 8086 is executed and verified successfully

16 BIT SUBTRACTIONS



EX NO: 02

DATE: 16/8/21

16 BIT SUBTRACTION USING 8086

AIM:

To write the program for 16 bit subtraction using 8086 microprocessor kit.

APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Adaptor	1
3.Opcode sheet	1

ALGORITHM:

1.Start the program

2.Get the first operand from the memory

3.Get the second operand from the memory

4.Subtract both operand

5.Store the result

6.Stop the program and store it in specific notation

7.Execute the program

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8800	05	8804	02
8801	02	8805	01
8802	03		
8803	01		

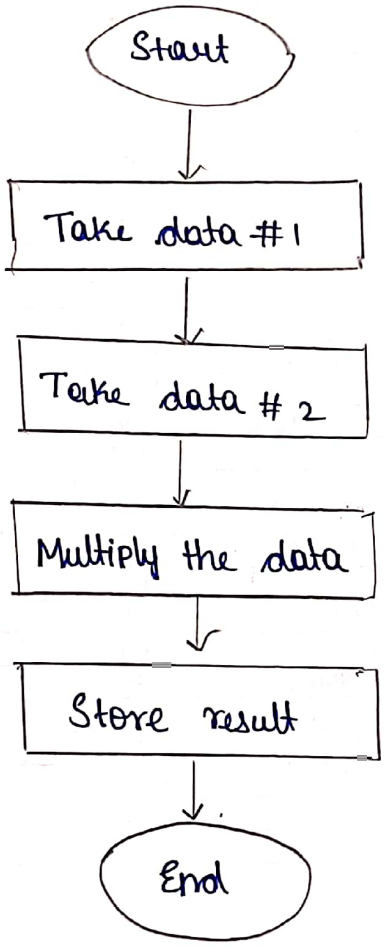
16 BIT SUBTRACTION USING 8086

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	AX,[8800]	8B	Move the 8800 to AX
8001				06	
8002				00	
8003				88	
8004		MOV	CL,00	B1	CLEAR THE REGISTER
8005				00	
8006		MOV	BX,[8802]	8B	MOVE THE DATA TO BX
8007				1E	
8008				02	
8009				88	
800A		SUB	AX,BX	2B	SUBTRACT BX FROM AX
800B				C3	
800C		MOV	[8804],AX	89	MOVE AX TO DATA
800D				06	
800E				04	
800F				88	
8010		JNC	LOOP	73	
8011				02	
8012		INC	CL	FE	INCREEMENT CL
8013				C1	
8014	LOOP	MOV	[8806],CL	88	MOVE CL TO DATA
8015				0E	
8016				06	
8017				08	
8018		HLT		F4	END OF THE PROGRAM

RESULT:

Thus the program for 16 bit subtraction using 8086 is executed and verified successfully

16 BIT MULTIPLICATIONS



EX NO: 03

DATE: 16/8/24

16 BIT MULTIPLICATION USING 8086

AIM:

To write the program for 16 bit multiplication using 8086 microprocessor kit.

APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Adaptor	1
3.Opcode sheet	1

ALGORITHM:

1.Start the program

2.Get the first operand from the memory

3.Get the second operand from the memory

4.Multiply both operand

5.Store the result

6.Stop the program and store it in specific notation

7.Execute the program

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8100	02	8200	04
8101	01	8201	02
8102	02		
8103	02		

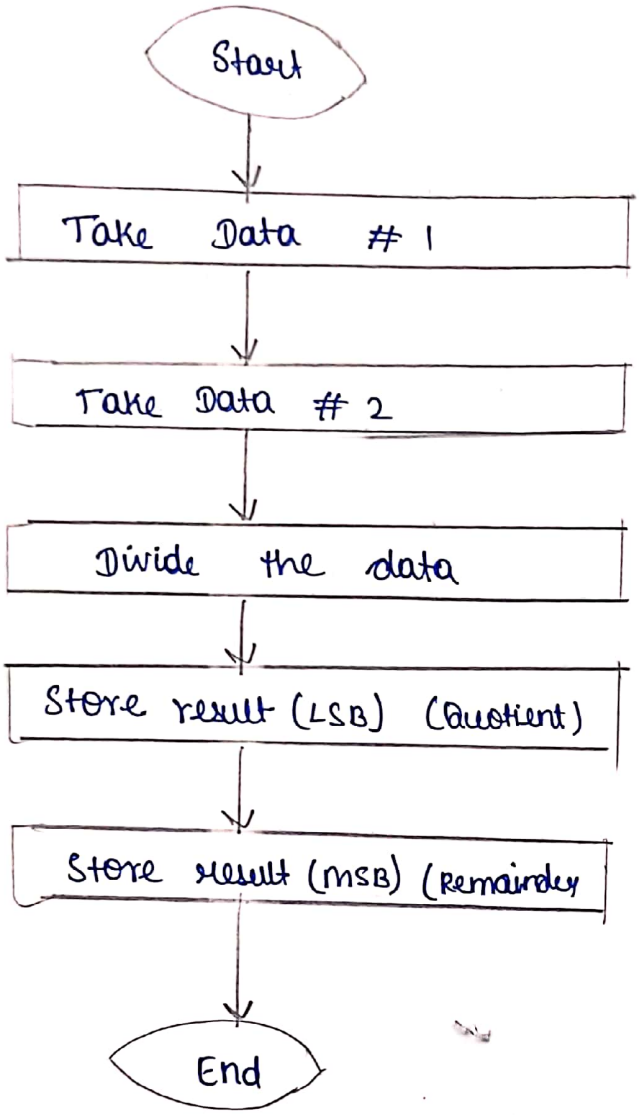
16 BIT MULTIPLICATION USING 8086

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	AX,[8100]	8B	MOVE THE DATA TO AX
8001				06	
8002				00	
8003				81	
8004		MOV	BX,[8102]	8B	MOVE THE DATA TO BX
8005				1E	
8006				02	
8007				81	
8008		MUL	BX	F7	MULTIPLY BX
8009				E3	
800A		MOV	[8200],AX	89	MOVE AX TO DATA
800B				06	
800C				00	
800D				82	
800E				89	
800F		MOV	[8202],DX	16	MOVE DX TO DATA
8010				02	
8011				82	
8012		HLT		F4	END OF THE PROGRAM

RESULT:

Thus the program for 16 bit multiplication using 8086 is executed and verified successfully

16 - Bit Division



EX NO: 04

DATE: 16/8/21

16 BIT DIVISION USING 8086

AIM:

To write the program for 16 bit division using 8086 microprocessor kit.

APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Adaptor	1
3.Opcode sheet	1

ALGORITHM:

1.Start the program

2.Get the first operand from the memory

3.Get the second operand from the memory

4.Divide both operand

5.Store the result

6.Stop the program and store it in specific notation

7.Execute the program

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8100	02	8200	02
8101	04	8201	01
8102	01		
8103	04		

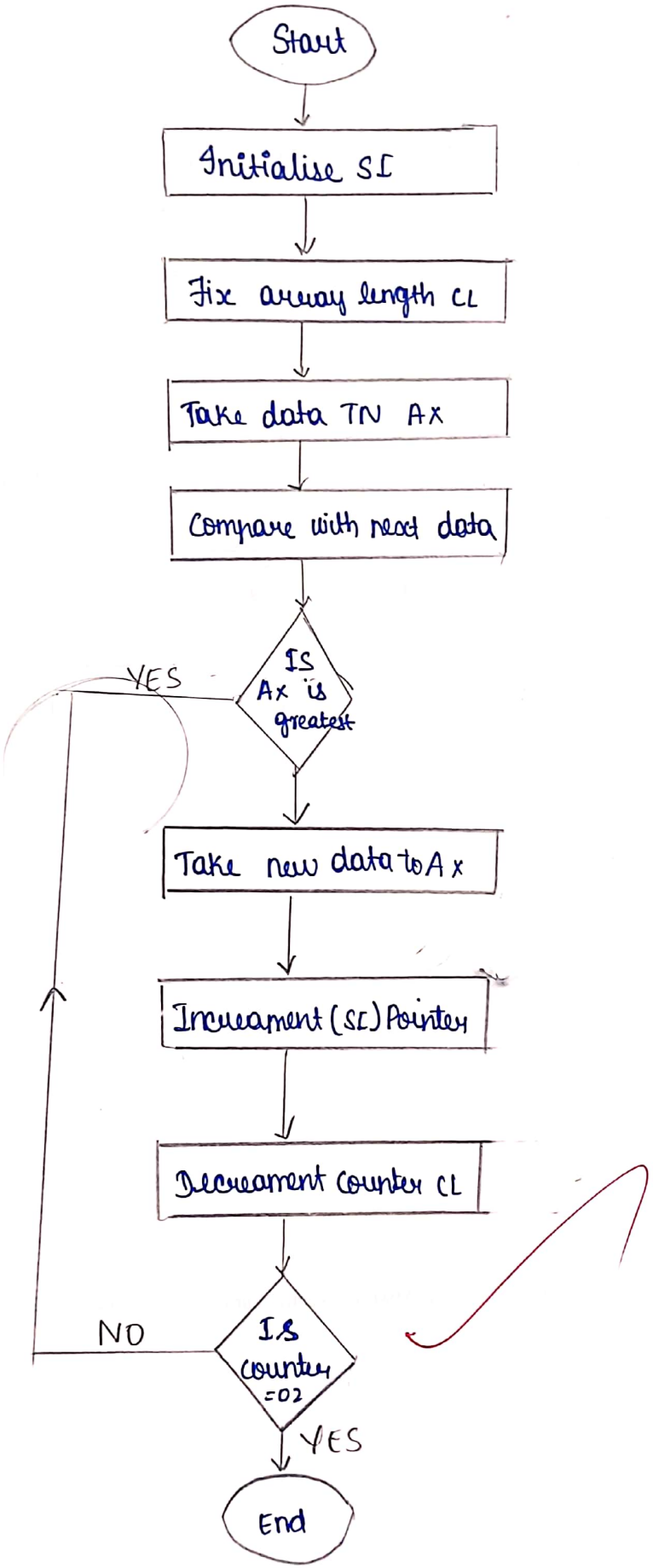
16 BIT DIVISION USING 8086

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	AX,[8100]	8B	MOVE THE DATA TO AX
8001				06	
8002				00	
8003				81	
8004		MOV	BX,[8102]	8B	MOVE DATA TO BX
8005				1E	
8006				02	
8007				81	
8008		DIV	BX	F7	DIVIDE BX
8009				E3	
800A		MOV	[8200],AX	89	MOVE AX TO DATA
800B				06	
800C				00	
800D				82	
800E		MOV	[8202],DX	89	MOVE DX TO DATA
800F				16	
8010				02	
8011				82	
8012		HLT		F4	END OF THE PROGRAM

RESULT:

Thus the program for 16 bit division using 8086 is executed and verified successfully

FINDING THE LARGEST OF THREE NUMBERS



EXNO: 1(a)(2)

DATE: 19/8/21

FINDING THE LARGEST OF THREE NUMBERS

AIM:

To write a program for assembly language to find the largest array in the list using 8086 microprocessor kit.

APPARATUS REQUIRED:

1.8086 Microprocessor kit 1

2.Power supply adapter

ALGORITHM:

1.Start the program

2.Load the number of passes an input each of number of elements in the array.

3.Move 1 to B register and increment BL register

4.Move data to accumulator and the compare with C register

5.In accumulator value is greater than C set array flag and decrement B register

6.If zero comes go to step 4

7.Stop the program

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8200	02	8300	07
8201	03	8301	04
8202	01		
8203	07		
8204	04		
8205	03		

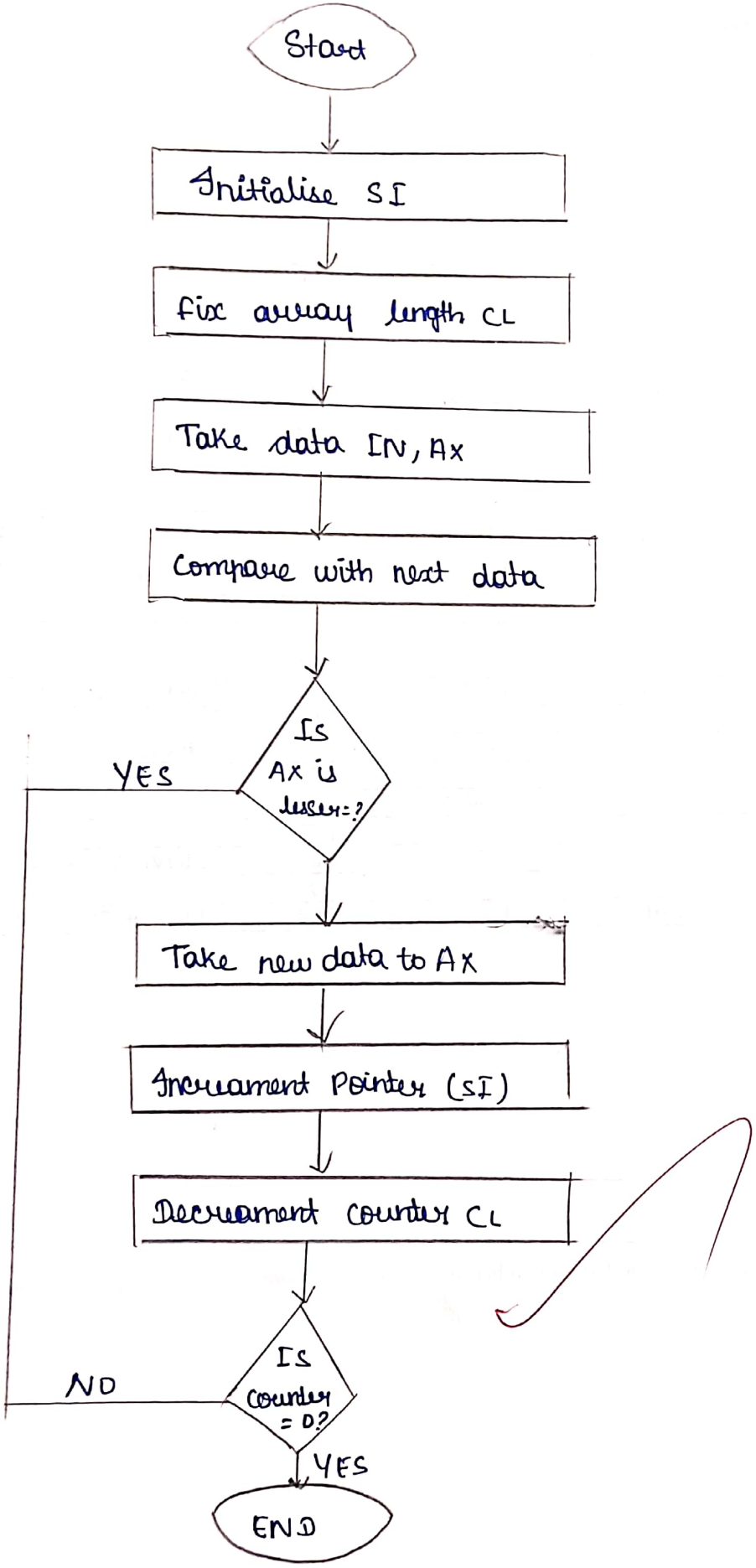
FINDING LARGEST NUMBER IN ARRAY

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	SI,8200	BE	MOVE THE DATA TO SI
8001				00	
8002				82	
8003		MOV	DI,8300	BF	MOVE DATA TO DESTINATION INDEX
8004				00	
8005				83	
8006		MOV	CL,[SI]	8B	MOVE SI TO CLEAR REGISTER
8007				0C	
8008		INC	SI	46	INCREEMENT SI
8009		MOV	AX,[SI]	86	MOVE SI TO AX
800A				04	
800B		DEC	CL	FE	DECREEMENT CL
800C				C9	
800D		INC	SI	46	INCREEMENT SI
800E		MOV	BX,[SI]	8B	MOVE SI TO BX
800F				1C	
8010		CMP	AX,BX	38	COMPARE AX AND BX
8011				D8	
8012		JNC	8016	73	
8013				02	
8014		MOV	AX,BX	88	MOVE BX TO AX
8015				D8	
8016		DEC	CL	FE	DECREEMENT CL
8017				C9	
8018		JNZ	800D	75	
8019				F3	
801A		MOV	[DI],AX	89	MOVE AX TO DI
801B				05	
801C		HLT		F4	END OF THE PROGRAM

RESULT:

Thus the program for 8 bit largest number in an array using 8086 microprocessor is executed successfully.

FINDING THE SMALLEST VALUES



EX NO: 06

DATE: 19/8/21

FINDING THE SMALLEST NUMBERS

AIM:

To write a program for assembly language to find the smallest number in an array using 8086 microprocessor kit.

APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Power supply adapter	1

ALGORITHM:

1.Start the program

2.Load the number of passes an input each of number of elements in the array.

3.Move 1 to B register and increment BL register

4.Move data to accumulator and the N compare with C register

5.In accumulator value is less than C set carry flag and decrement B register

6.If zero comes go to step 4

7.Stop the program

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8200	07	8300	02
8201	02	8301	01
8202	01		
8203	03		
8204	06		
8205	05		
8206	09		

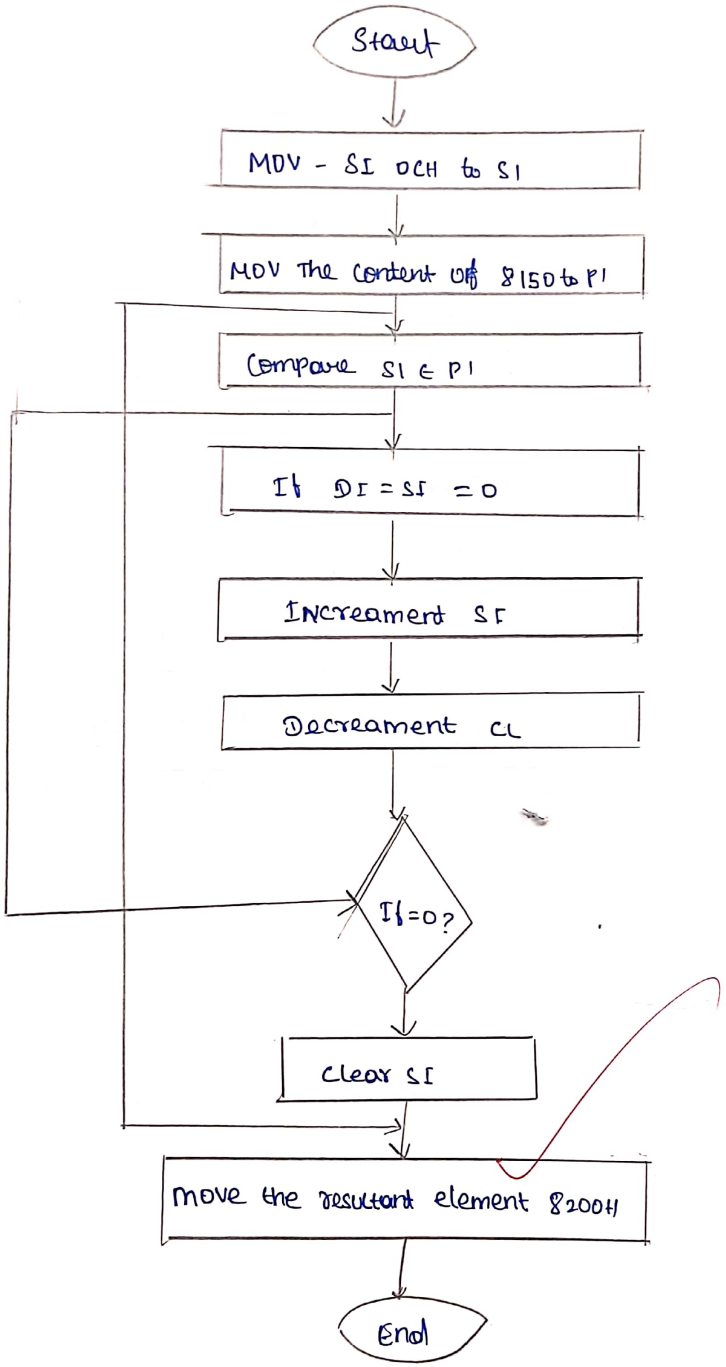
FINDING SMALLEST NUMBER IN ARRAY

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	SI,8200	BE	MOVE THE DATA TO SI
8001				00	
8002				82	
8003		MOV	DI,8300	BF	MOVE DATA TO DI
8004				00	
8005				83	
8006		MOV	CL,[SI]	8B	CLEAR THE SI
8007				0C	
8008		INC	SI	46	INCREEMENT SOURCE INDEX
8009		MOV	AX,[SI]	86	MOVE SI TO AX
800A				04	
800B		DEC	CL	FE	DECREEMENT CL
800C				C9	
800D		INC	SI	46	INCREEMENT SI
800E		MOV	BX,[SI]	8B	
800F				1C	
8010		CMP	AX,BX	38	COMPARE AX AND BX
8011				D8	
8012		JC	8016	72	
8013				02	
8014		MOV	AX,BX	88	MOVE BX TO AX
8015				D8	
8016		DEC	CL	FE	DECREEMENT CL
8017				C9	
8018		JNZ	800D	75	
8019				F3	
801A		MOV	[DI],AX	89	MOVE AX TO DI
801B				05	
801C		HLT		F4	END OF THE PROGRAM

RESULT:

Thus the program for 8 bit smallest number in an array using 8086 microprocessor is executed successfully

SEARCHING AN ELEMENT FROM THE GIVEN ARRAY



07 19/8/24	SEARCHING AN ELEMENT FROM THE GIVEN ARRAY

AIM:

To write the program for searching an element from the given array

APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Adaptor	1
3.Opcode sheet	1

ALGORITHM:

1.Start the program

2.Move the content of memory to BL and move data to SI

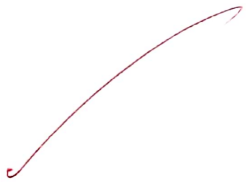
3.Move the content of memory to DI

4.Compare content of DI and SI

5.When ZF=0 jump to loop and decrement BL by 1

6.Move the content of SI to memory

7.Stop the program



SEARCH I/P

8150 0A
8151 00

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8100	05	8200	03
8101	04	8201	00
8102	00	8202	81
8103	05		
8104	00		
8105	09		
8106	00		
8107	0A		
8108	00		
8109	FF		
810A	00		

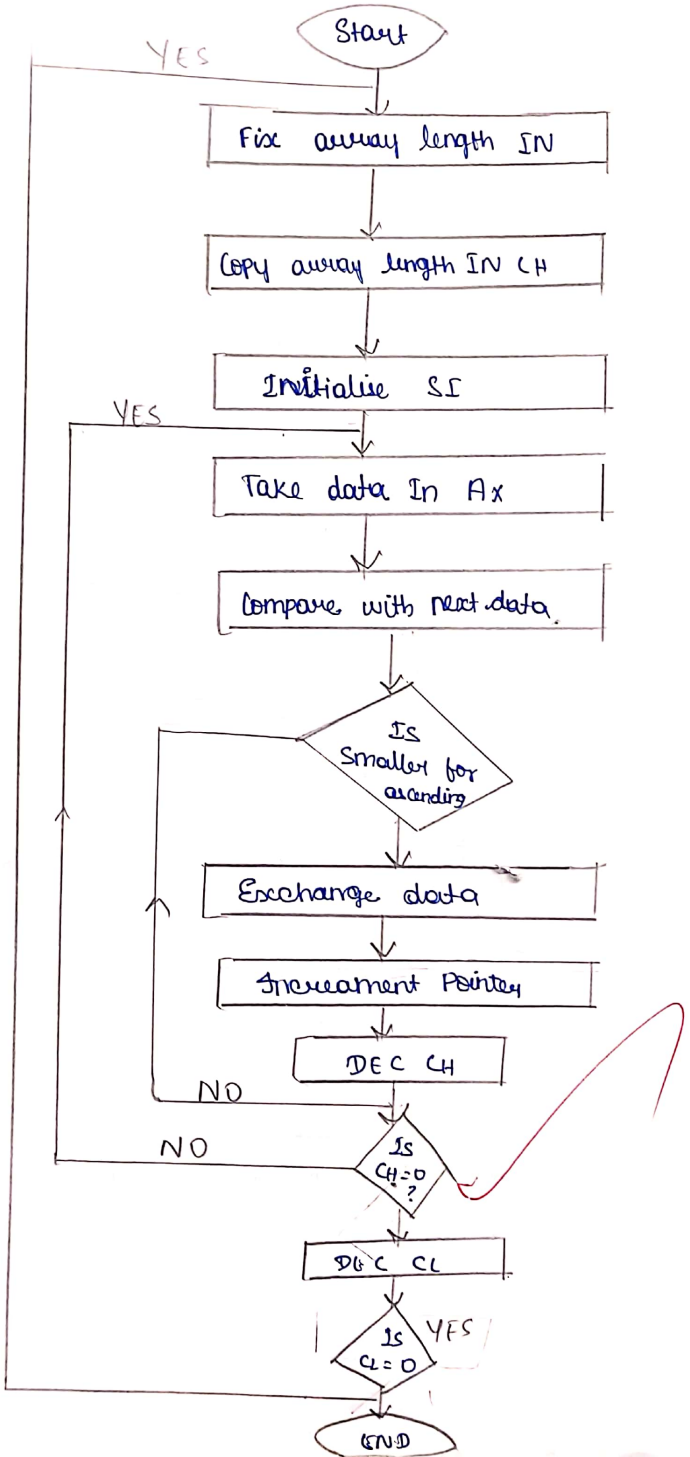
SEARCHING A NUMBER IN AN ARRAY

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	SI,8100	BE	MOVE THE DATA TO SOURCE INDEX
8001				00	
8002				81	
8003		MOV	CL,[SI]	8B	CLEAR THE SI
8004				0C	
8005		INC	SI	46	INCREEMENT SI
8006		MOV	AX,[8150]	8B	MOVE DATA TO AX
8007				06	
8008				50	
8009				81	
800A	LOOP2	CMP	AX,[SI]	39	COMPARE AX,SI
800B				04	
800C		JZ	LOOP1	74	
800D				09	
800E		INC	SI	46	INCREEMENT SI
800F		INC	SI	46	
8010		DEC	CL	FE	DECREEMENT CL
8011				C9	
8012		JNZ	LOOP2	75	
8013				76	
8014		MOV	SI,0000	BE	MOVE DATA TO SI
8015				00	
8016				00	
8017		MOV	[8200],SI	89	MOVE SI TO DATA
8018				36	
8019				00	
801A				82	
801B		HLT		F4	END OF THE PROGRAM

RESULT:

Thus the program for searching a number in an array using 8086 is executed and verified successfully

ASCENDING ORDER.



07

26/8/21

ASCENDING ORDER USING 8086

AIM:

To write the program for ascending order using 8086 microprocessor kit.

APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Adaptor	1
3.Opcode sheet	1

ALGORITHM:

1.Start the program

2.Load the content of accumulator directly from the given address

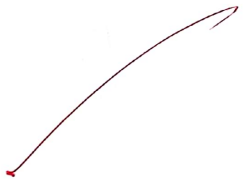
3.Move the content of pair by B to C and it load it to HL memory

4.Increment the content of pair by 1

5.Compare the value between A and D and the result is stored in flag

6.Store the ordered content in accumulator

7.Stop the program



15

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8100	44	8100	33
8101	44	8101	33
8102	55	8102	44
8103	55	8103	44
8104	66	8104	55
8105	66	8105	55
8106	77	8106	66
8107	77	8107	66
8108	33	8108	77
8109	33	8109	77

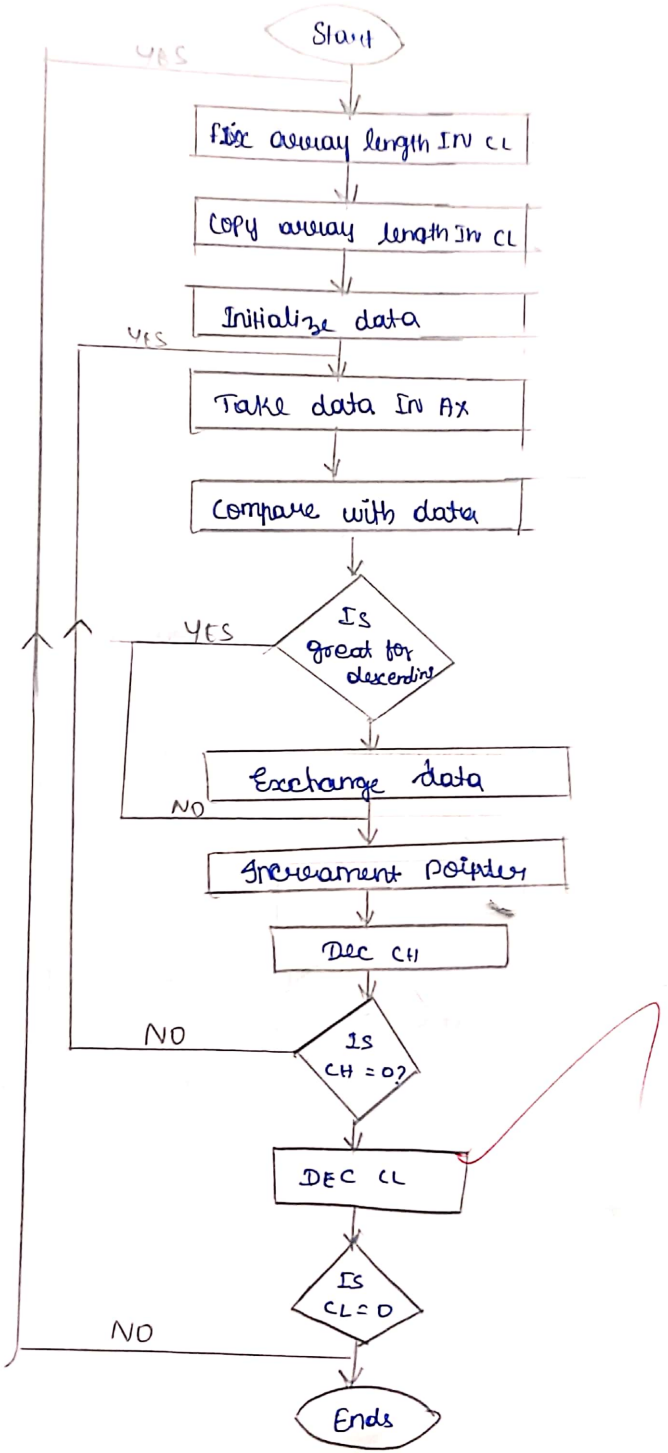
ASCENDING ORDER USING 8086

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	CL,04H	B1	MOVE THE DATA TO CLEAR REGISTER
8001				04	
8002	LOOP3	MOV	CH,04H	B5	MOVE THE DATA TO CH
8003				04	
8004		MOV	S1,8100H	BE	MOVE THE DATA TO SI
8005				00	
8006				81	
8007	LOOP2	MOV	AX,[SI]	8B	MOVE SI TO AX
8008				04	
8009		INC	S1	46	INCREEMENT SOURCE INDEX
800A		INC	S1	46	
800B		MOV	DX,[SI]	8B	MOVE SI TO DX
800C				14	
800D		CMP	AX,DX	39	COMPARE AX ,DX
800E				D0	
800F		JNC	LOOP1	72	
8010				0A	
8011		MOV	BX,[SI]	8B	MOVE SI TO BX
8012				1C	
8013		MOV	[SI],AX	89	MOVE AX TO SI
8014				04	
8015		DEC	SI	4E	DECREEMENT SOURCE INDEX
8016		DEC	SI	4E	DECREEMENT SOURCE INDEX
8017		MOV	[SI],BX	89	MOVE BX TO SI
8018				1C	
8019		INC	SI	46	INCREEMENT SI
801A		INC	SI	46	
801B	LOOP1	DEC	CH	FE	DECREEMENT CH
801C				CD	
801D		JNZ	LOOP2	75	
801E				E8	
801F		DEC	CL	FE	DECREEMENT CL
8020				C9	
8021		JNC	LOOP3	75	
8022				DF	
8023		HLT		F4	END OF THE PROGRAM

RESULT:

Thus the program for 16 bit storing number in ascending order using 8086 microprocessor is executed successfully

DESCENDING ORDER



09

26/8/21

DESCENDING ORDER USING 8086

AIM:

To write the program for descending order using 8086 microprocessor kit.

APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Adaptor	1
3.Opcode sheet	1

ALGORITHM:

1.Start the program

2.Get the data and more immediately to CX register

3.Increment the pointer by 1 to store the data in next memory location

4.Compare the data and store the data in descending order

5.The data which are stored should be arranged in order

6.Increment the pointer

7.Stop the program

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8100	44	8100	77
8101	44	8101	77
8102	66	8102	66
8103	66	8103	66
8104	55	8104	55
8105	55	8105	55
8106	77	8106	44
8107	77	8107	44
8108	33	8108	33
8109	33	8109	33

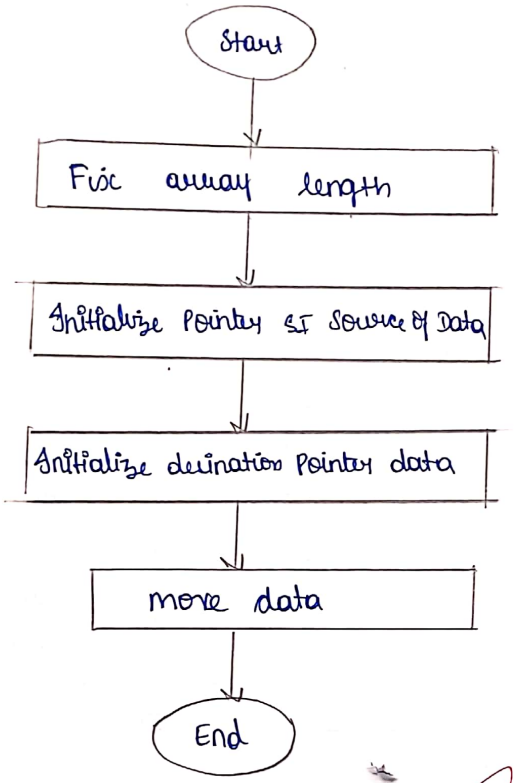
DESCENDING ORDER USING 8086

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	CL,04H	B1	MOVE DATA TO CLEAR REGISTER
8001				04	
8002	LOOP3	MOV	CH,04H	B5	MOVE THE DATA TO CH
8003				04	
8004		MOV	S1,8100H	BE	MOVE THE DATA TO SI
8005				00	
8006				81	
8007	LOOP2	MOV	AX,[SI]	8B	MOVE SI TO AX
8008				04	
8009		INC	SI	46	INCREEMENT SOURCE INDEX
800A		INC	SI	46	
800B		MOV	DX,[SI]	8B	MOVE SI TO DX
800C				14	
800D		CMP	AX,DX	39	COMPARE AX AND DX
800E				D0	
800F		JNC	LOOP1	73	
8010				0A	
8011		MOV	BX,[SI]	8B	MOVE SI TO BX
8012				1C	
8013		MOV	[SI],AX	89	MOVE AX TO SI
8014				04	
8015		DEC	SI	4E	DECREEMENT SI
8016		DEC	SI	4E	
8017		MOV	[SI],BX	89	MOVE BX TO SI
8018				1C	
8019		INC	SI	46	INCREEMENT SI
801A		INC	SI	46	
801B	LOOP1	DEC	CH	FE	DECREEMENT CH
801C				CD	
801D		JNZ	LOOP2	75	
801E				E8	
801F		DEC	CL	FE	DECREEMENT CL
8020				C9	
8021		JNC	LOOP3	75	
8022				DF	
8023		HLT		F4	END OF THE PROGRAM

RESULT:

Thus the program for 16 bit storing number in descending order using 8086 microprocessor is executed successfully

STRING MANIPULATION



10	
26/8/24	**STRING MANIPULATION USING 8086**

AIM:

To write the program for string manipulation using 8086 microprocessor kit.

APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Adaptor	1
3.Opcode sheet	1

ALGORITHM:

1.Start the program

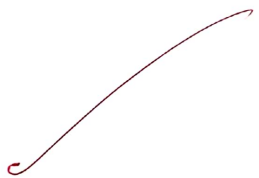
2.Move the content of memory to CX

3.Move the immediate data to SI

4.Move immediate data to DI

5.Repeatedly move the content between memory location through SI and DI by looping

6.Stop the program



INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8100	02	8202	05
8102	05	8203	06
8103	06	8204	07
8104	07	8205	08
8105	08		

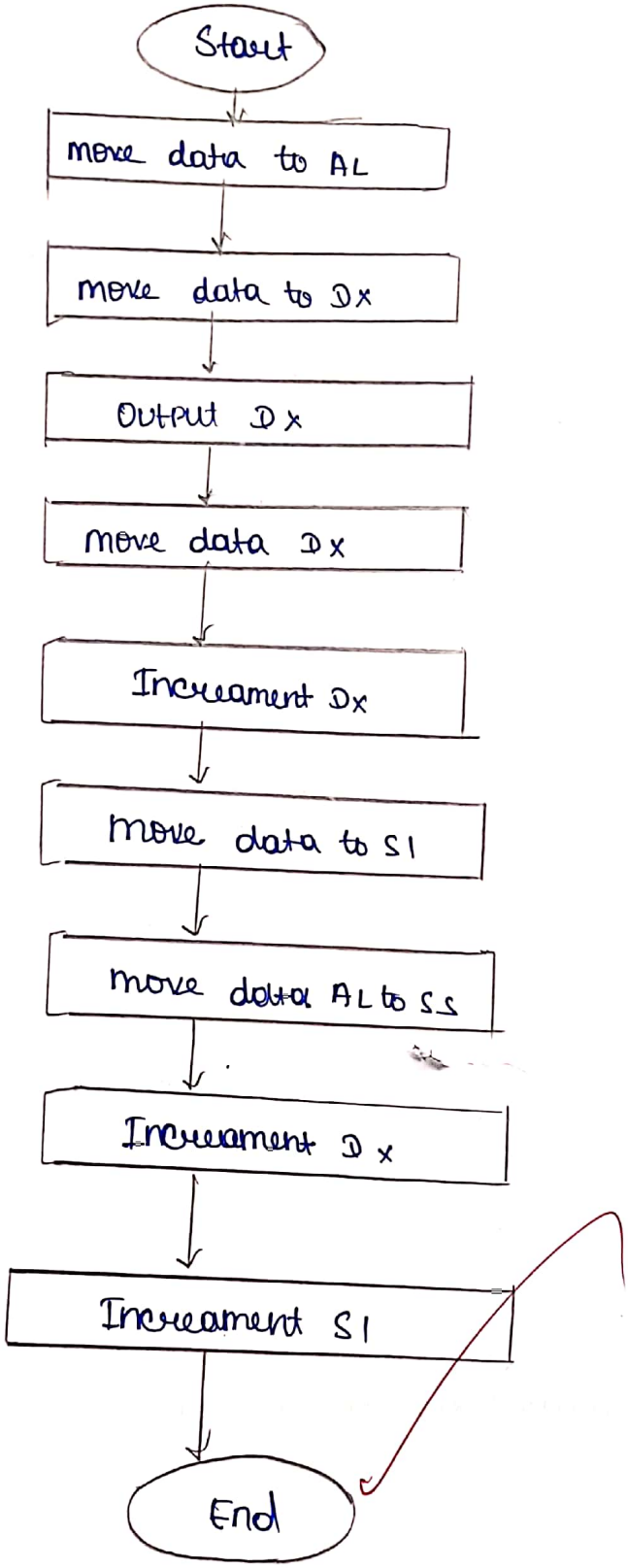
STRING MANIPULATION PROGRAM FOR BLOCK TRANSFER

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	CX,[8100H]	8B	MOVE THE DATA TO CX
8001				0E	
8002				00	
8003				81	
8004		MOV	SI,8102H	BE	MOVE DATA TO SI
8005				02	
8006				81	
8007		MOV	DI,8202H	BF	MOVE DATA TO DI
8008				02	
8009				82	
800A	REP	MOV	SW	F3	
800B				A5	
800C		HLT		F4	RND OF THE PROGRAM

RESULT:

Thus the program for block transfer using 8086 microprocessor is executed successfully.

PARALLEL COMMUNICATION



11

02/9/21

PARALLEL COMMUNNNICATION

AIM:

To write the program for transmitting and receiving the signal in the parallel communication system.

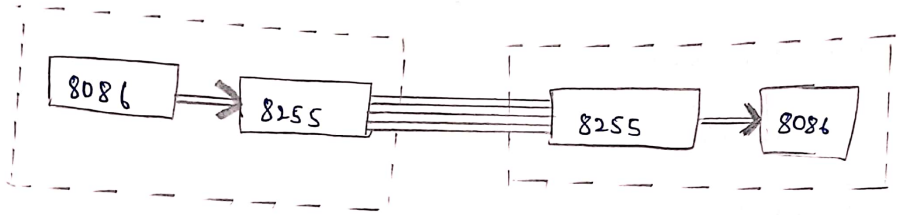
APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Power supply	1

ALGORITHM:

1.Start the program
2.In the transistor side more the content of data to AL
3.Get the port address to the register
4.Get the memory address
5.Use the 'OUT' operator
6.In receiver side, get the port address
7.Get the port data
8.Use the 'IN' operator
9.Increment the value
10.Stop the program

PARALLEL Communication.



CWR = FFE6

PA = FFE0

PB = FFE2

PC = FFEC

INPUT:

8007 04

8009 06

PROGRAM FOR TRANSMITTER

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	AL,80	B0	MOVE THE DATA TO AL
8001				80	
8002		MOV	DX,FFE6	BA	MOVE THE DATA TO DX
8003				E6	
8004				99	
8005		OUT	DX	EE	OUTPUT DX
8006		MOV	AL,04	B0	MOVE DATA TO AL
8007				04	
8008		MOV	DX,FFE0	BA	MOVE THE DATA TO DX
8009				E0	
800A				FF	
800B		OUT	DX	FE	OUT DX
800C		MOV	AL,06	B0	MOVE THE DATA TO AL
800D				06	
800E		MOV	DX,FFE2	BA	MOVE THE DATA TO DX
800F				E2	
8010				FF	
8011		OUT	DX	EE	OUTPUT DX
8012		HLT		F4	END OF THE PROGRAM

d/p

8500 Dy

1058 D6

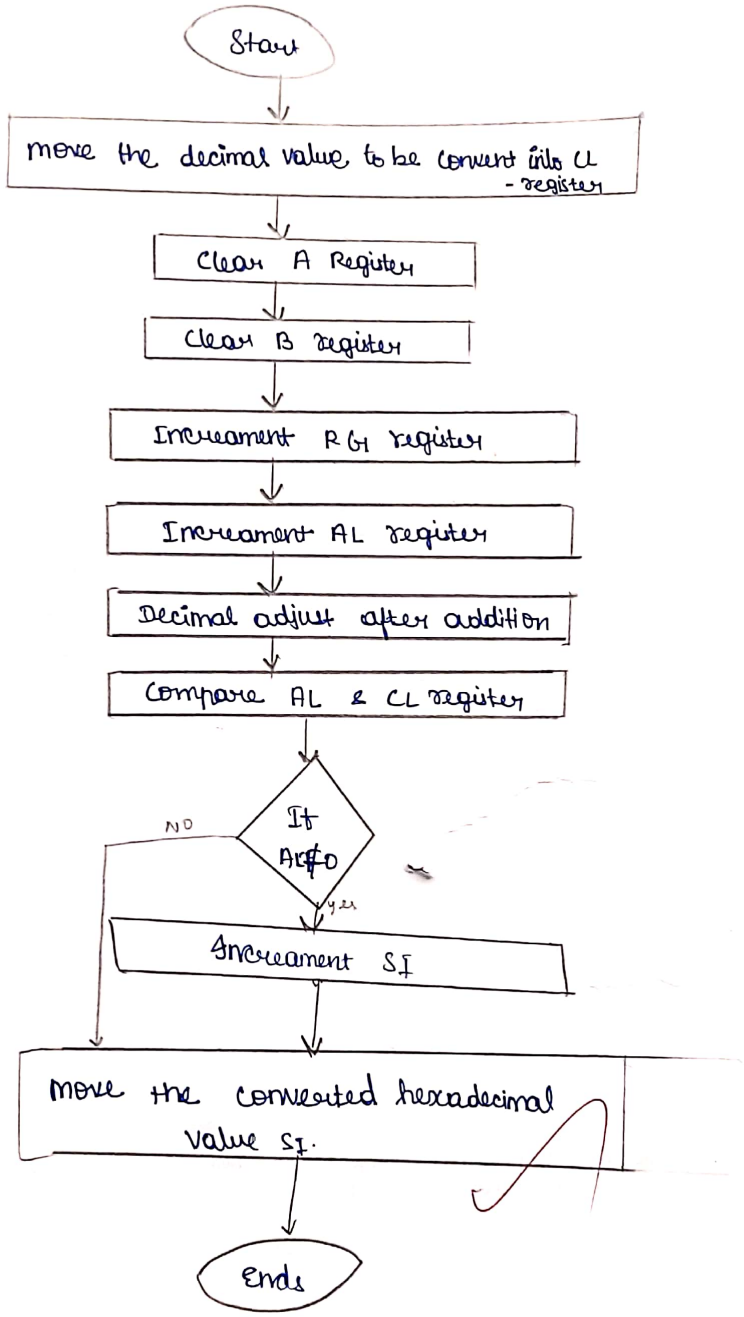
PROGRAM FOR RECEIVER

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	AL,92	B0	MOVE DATA TO AL
8001				92	
8002		MOV	DX,FFE6	BA	MOVE DATA TO DX
8003				E6	
8004				FF	
8005		OUT	DX	EE	OUTPUT DX
8006		MOV	DX,FFEO	BA	MOVE THE DATA TO DX
8007				E0	
8008				FF	
8009		IN	DX	EC	INCREEMENT DX
800A		MOV	SI,8500	BE	MOVE THE DATA TO SI
800B				00	
800C				85	
800D		MOV	[SI],AL	88	MOVE AL TO SI
800E				04	
800F		MOV	DX,FFE2	BA	MOVE DATA TO DX
8010				E2	
8011				FF	
8012		IN	DX	EC	INCREEMENT DX
8013		INC	SI	46	INCREEMENT SI
8014		MW	[SI],AL	88	
8015				04	
8016		HLT		F4	END OF THE PROGRAM

RESULT:

Thus the program for parallel communication between two microprocessor receives and transmits using 8086 microprocessor is executed successfully

DECIMAL TO HEXADECIMAL



12

02/9/21

HEXADECIMAL TO DECIMAL CONVERSION USING 8086

AIM:

To write the program for string manipulation using 8086 microprocessor kit.

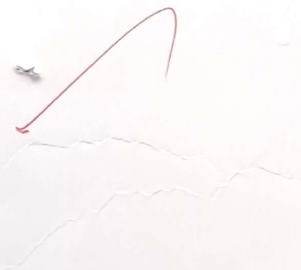
APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Adaptor	1
3.Opcode sheet	1

ALGORITHM:

1. Move the Data To Al and Move Al To Dl
2. Move Data To Al And Move Data To Cl
3. Increment Ac and Increment Cl
4. Decrement Dl
5. Move Al To Data and Move Cl To Data
6. End Of The Program

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8100	11	8101	0B



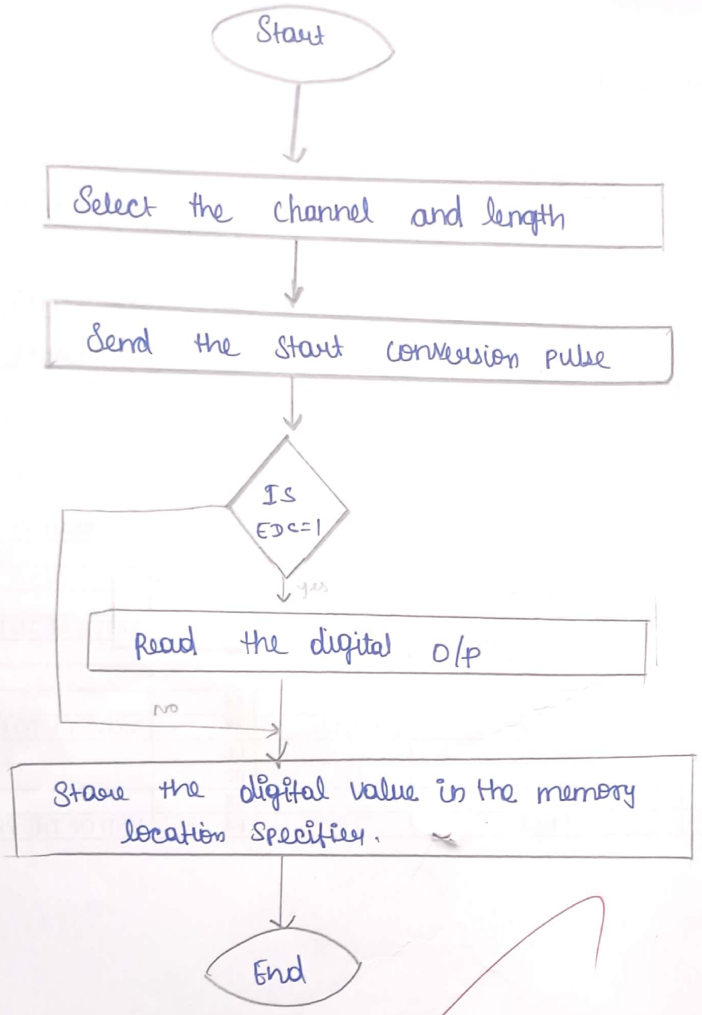
HEXADECIMAL TO DECIMAL

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	AL,[8800]	A0	MOVE THE DATA TO AL
8001				00	
8002				88	
8003		MOV	DL,AL	8A	MOVE AL TO DL
8004				D0	
8005		MOV	AL,00	B0	MOVE DATA TO AL
8006				00	
8007		MOV	CL,00	B1	MOVE DATA TO CL
8008				00	
8009	LOOP2	INC	AC	FE	INCREMENT AC
800A				C0	
800B		DAA		27	
800C		JNC	LOOP1	73	
800D				02	
800E		INC	CL	FE	INCREMENT CL
800F				C1	
8010	LOOP1	DEC	DL	FE	DECREMENT DL
8011				CA	
8012		JNZ	LOOP2	75	
8013				F5	
8014		MOV	[8801],AL	88	MOVE AL TO DATA
8015				06	
8016				01	
8017				88	
8018		MOV	[8802],CL	88	MOVE CL TO DATA
8019				0E	
801A				02	
801B				88	
801C		HLT		F4	END OF THE PROGRAM

RESULT:

Thus the program for hexadecimal to decimal using 8086 microprocessor is executed successfully.

SAWTOOTH WAVEFORM



51

13

9/9/21

SAWTOOTH WAVE FORM GENERATION

AIM:

To write the program to generate saw tooth wave from.

APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Adaptor	1
3.CRO	1

ALGORITHM:

1.Start the program

2.Set the ports as a output and initialize the control word register

3.Clear the accumulator content immediately

4.Increment the accumulator and send the output to the port

5.It will be incremented till it reaches to index value

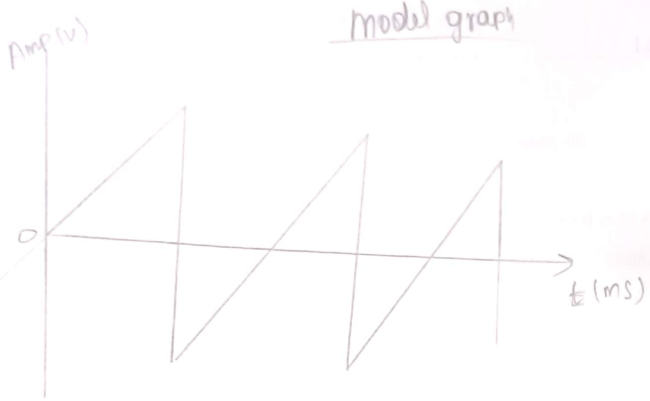
6.Further increment will jumps uncontionally

7.The process is repeated again and again

8.Stop the program

55

Model graph



Amplitude	Time Period
3 x 0.5 = 1.5V	1.2 x 1 = 1.2ms

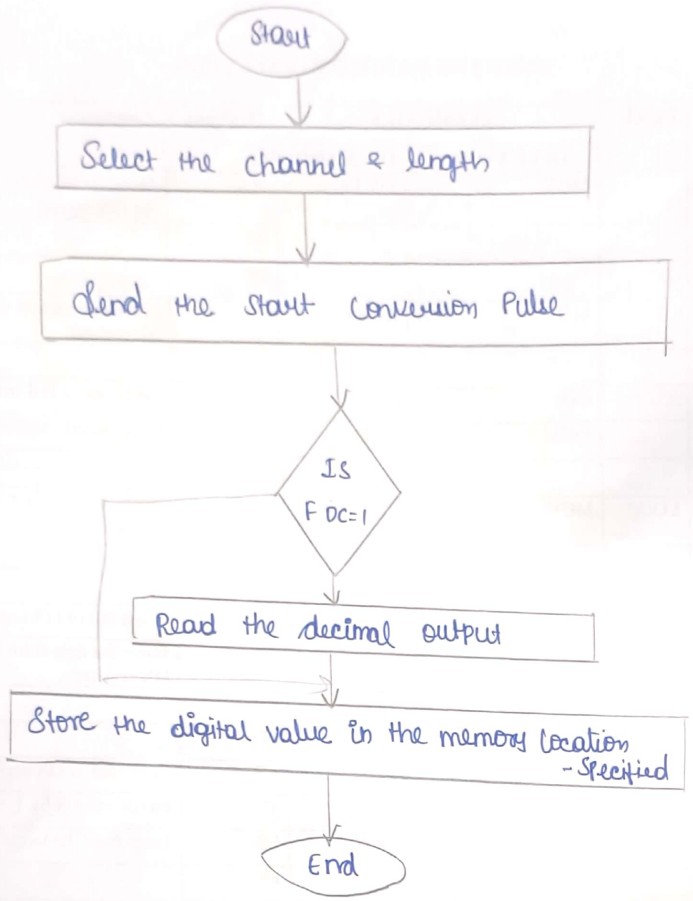
SAWTOOTH WAVEFORM GENERATION

Address	Label	MNEMONICS		Hex code	Comments
		OPERATOR	OPERAND		
8000H		MOV	DX,FFE6	BA	Move the content of FFE6 to the DX register
8001H				E6	
8002H				FF	
8003H		MOV	AL,80	B0	Move the content of 80 to the AL register
8004H				80	
8005H		OUT	DX	EE	Come out of DX register
8006H		MOV	AL,00	B0	Clear the AL register
8007H				00	
8008H	LOOP	MOV	DX,FFE0	BA	Move the data from FFE0 to DX register
8009H				E0	
800AH				FF	
800BH		OUT	DX	EE	Come out of DX register
800CH		MOV	DX,FFE2	BA	Move the data from FFE2 to DX register
800DH				E2	
800EH				FF	
800FH		OUT	DX	EE	Come out of DX register
8010H		INC	AX	40	Increment AX by 1
8011H		JMP	LOOP	EB	Jump from the loop
8012H				F5	

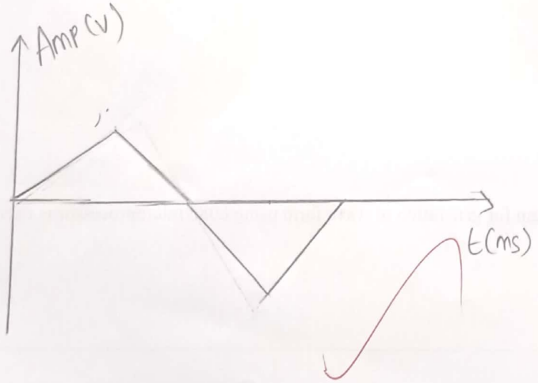
RESULT:

Thus the program for generation of wave form using 8086 microprocessor is executed successfully

TRIANGULAR WAVEFORM



MODEL GRAPH :



14

9/9/21

TRIANGULAR WAVE FROM GENERATION

AIM:

To write the program to generate the triangular wave form using 8086

APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Power supply	1

ALGORITHM:

1.Start the program

2.Move the data to the accumulator

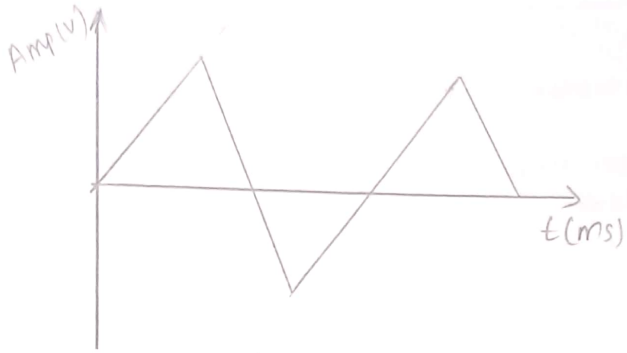
3. The accumulator content is set by the SWR

4.accumulator content is being incremented one by one

5.Once the maximum value is reached its starts decrementing till it reach zero

6.After every step, the accumulator sets out the content

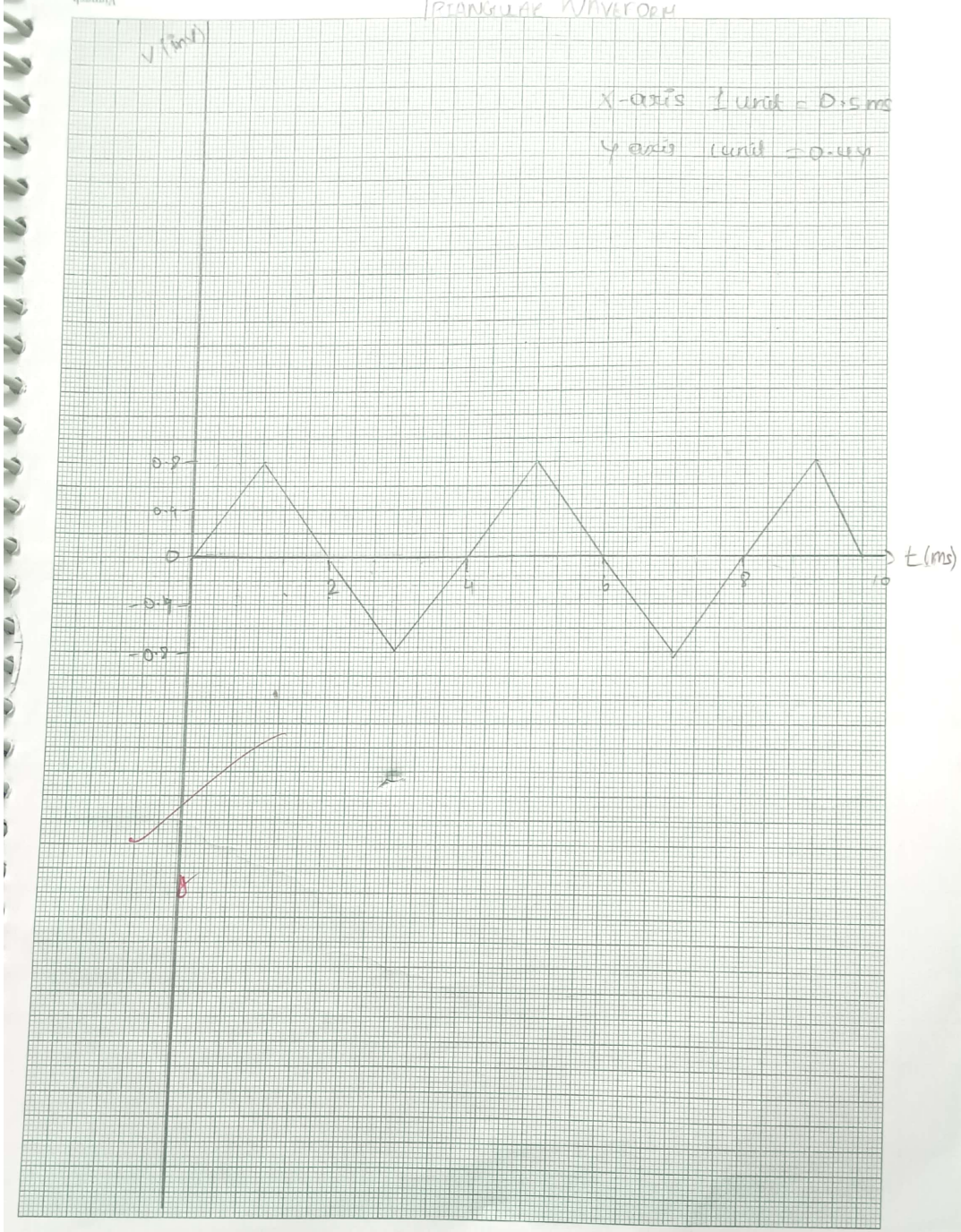
7.Stop the program



Amplitude	Time.
$3.2 \times 0.5 = 1.6V$	$4 \times 1 = 4ms$

Vignesh

TRIANGULAR WAVEFORM

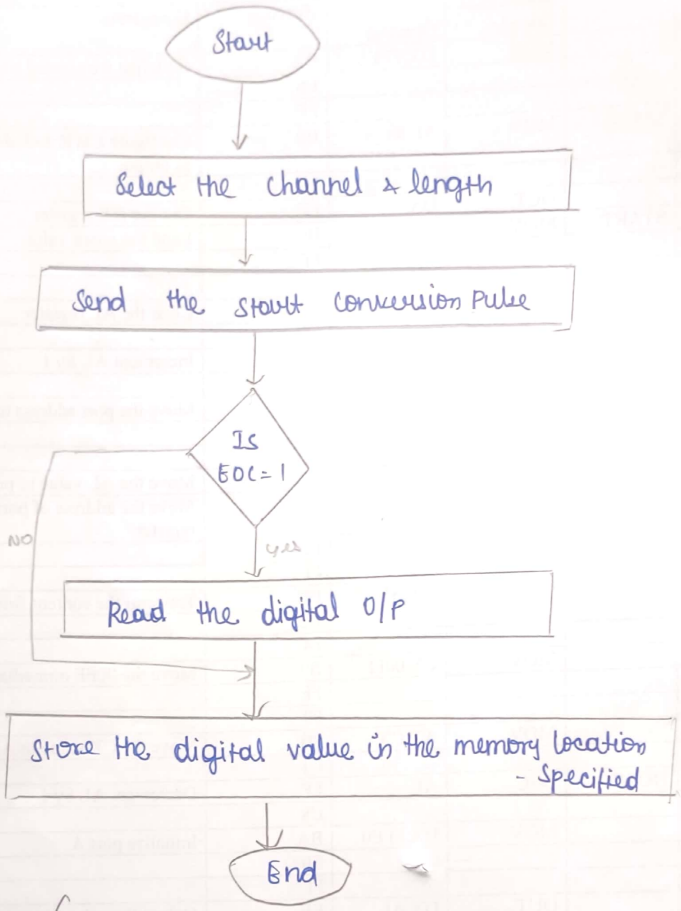


Address	Label	Mnemonics Opcode	Operand	Hexcode	Comments
8000H		MOV	DX,FFE6	BA	Initialize the control word register
8001H				E6	
8002H				FF	
8003H		MOV	AL,80	B0	Configure CWR and all the port as output
8004H				80	
8005H		OUT	DX	EE	Out the DX register
8006H	START	MOV	CX,00FF	B9	Load the count value
8007H				FF	
8008H				00	
8009H		MOV	AL,00	B0	Clear the AL register
800AH				00	
800BH	UP	INC	AL	40	Increment AL by 1
800CH				C0	
800DH		MOV	DX,FFE0	BA	Move the port address to DX register
800EH				E0	
800FH				FF	
8010H		OUT	DX,AL	EE	Move the AL value to port B
8011H		MOV	DX,FFE2	BA	Move the address of port B to DX register
8012H				E2	
8013H				FF	
8014H		OUT	DX,AL	EE	Transmit the content from port B
8015H		LOOP	UP	E2	
8016H				F4	
8017H		MOV	CX,00FF	B9	Move the 00FF immediate to CX
8018H				FF	
8019H				00	
801AH		MOV	AX,CX	8B	Move the CX value to AX
801BH				C1	
801CH	DOWN	DEC	AL	FE	Decrement AL by 1
801DH				C8	
801EH		MOV	DX,FFE0	BA	Initialize port A
801FH				E0	
8020H				FF	
8021H		OUT	DX,AL	EE	Output the value to port A
8022H		MOV	DX,FFE2	BA	
8023H				E2	
8024H				FF	
8025H		OUT	DX,AL	EE	
8026H		LOOP	DOWN	E2	
8027H				F8	
8028H		JMP	START	EB	
8029H				E0	

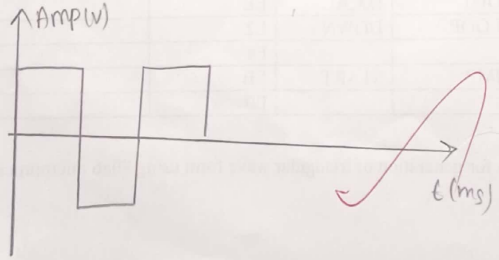
RESULT:

Thus the program for generation of triangular wave form using 8086 microprocessor is executed successfully.

SQUARE WAVEFORM



MODEL GRAPH:



1.5

9/9/21

SQUARE WAVE FORM GENERATION

AIM:

To write the program to generate square wave from.

APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Adaptor	1
3.CRO	1

ALGORITHM:

1.Start the program

2.Initialise output program

3.Set maximum value FF to A

4.Call uncondional delay for getting square shape

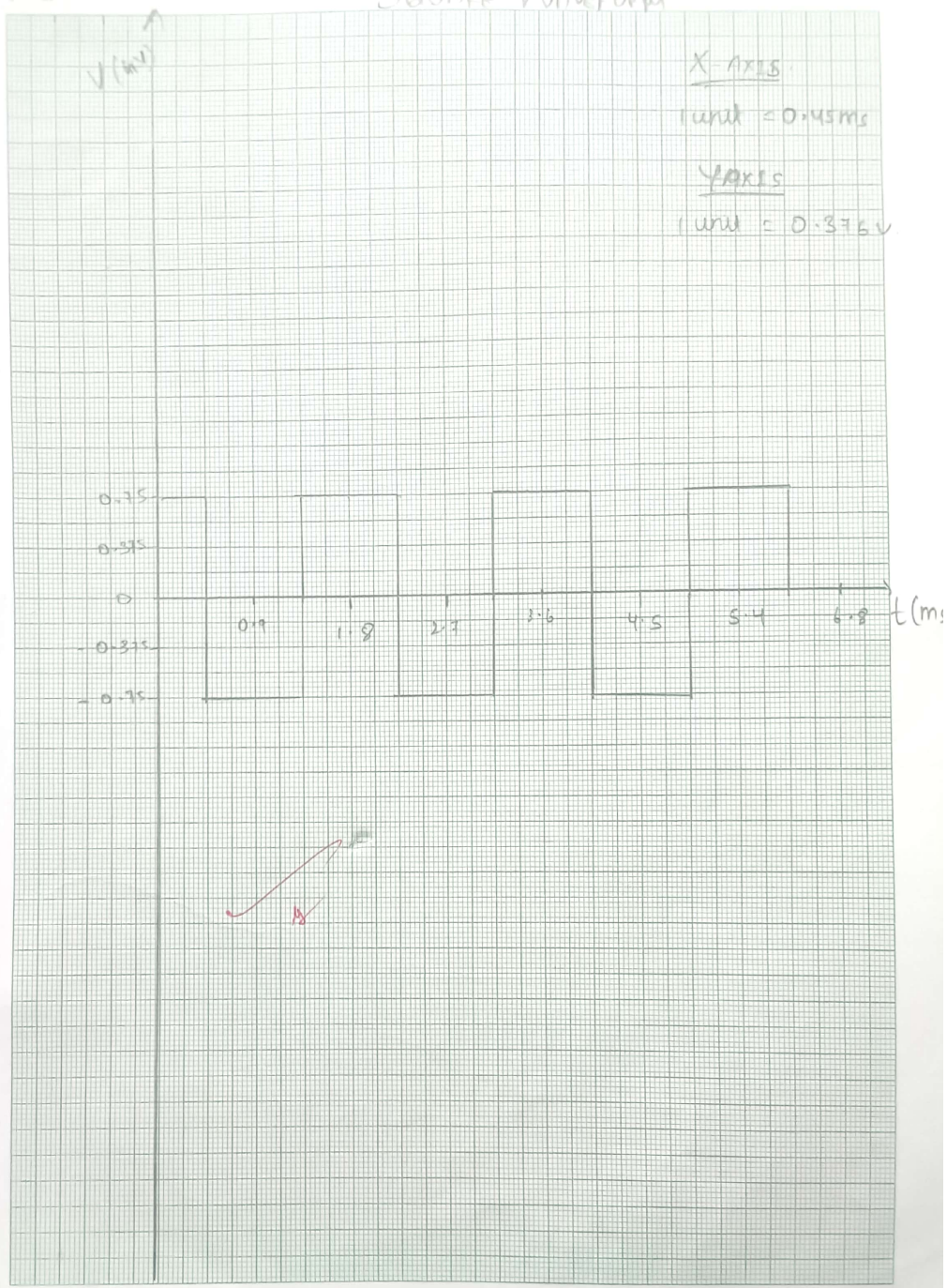
5.Tthen the value is decrement to 0

6.Again same steps are repeated

7.stop the program

Amplitude	Time Period
3×0·5 = 1·5V	1·8×1 = 1·8ms

Square Waveform

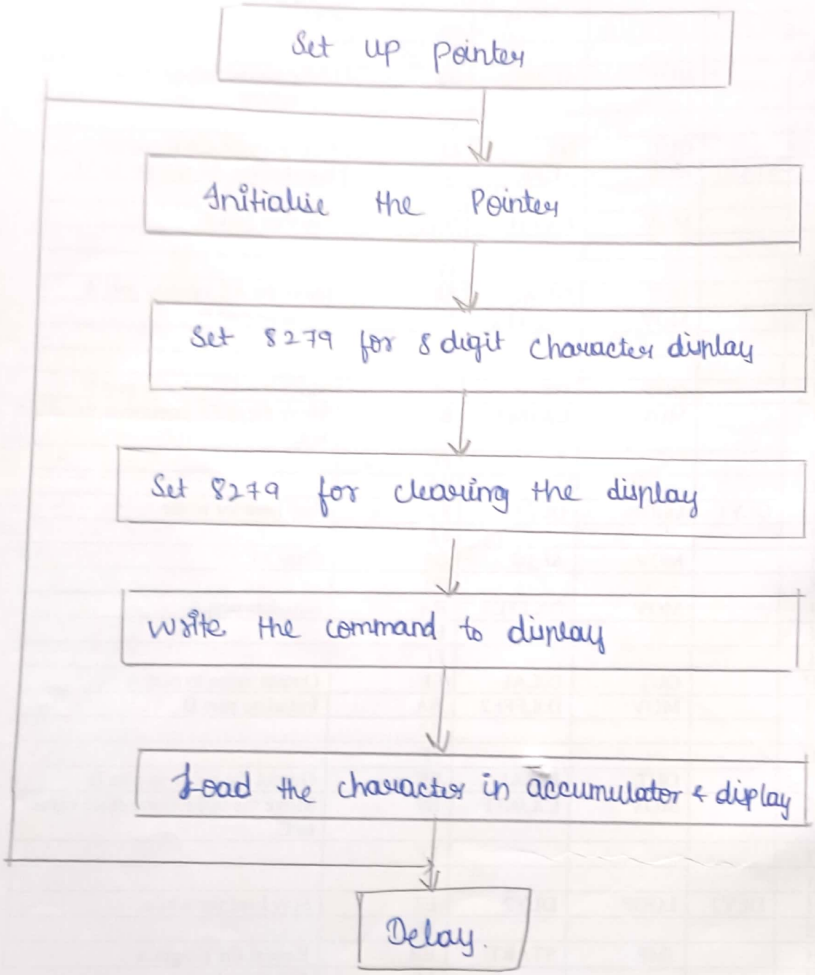


Address	Label	Mnemonics Opcode	Operand	Hex code	Comments
8000H		MOV	DX,FFE6	BA	Move the CWR register to DX register
8001H				E6	
8002H				FF	
8003H		MOV	AL,80	B0	Move the immediate value 80 to AL register
8004H				80	
8005H		OUT	DX	EE	Configure the CWD register
8006H	START	MOV	AL,FF	B0	Initialize the AL register by FF
8007H				FF	
8008H		MOV	DX,FFE0	BA	Initialize port A
8009H				E0	
800AH				FF	
800BH		OUT	DX,AL	EE	Move the AL value to port A
800CH		MOV	DX,FFE2	BA	Initialize port B
800DH				E2	
800EH				FF	
800FH		OUT	DX,AL	EE	Move the AL value to port B
8010H		MOV	CX,00FF	B9	Move the 00FF immediate value to C
8011H				FF	
8012H				00	
8013H	DLY1	LOOP	DLY1	E2	Stay here for while
8014H				FE	
8015H		MOV	AL,00	B0	Clear AL
8016H				00	
8017H		MOV	DX,FFE0	BA	Initialize port A
8018H				E0	
8019H				FF	
801AH		OUT	DX,AL	EE	Output value to port A
801BH		MOV	DX,FFE2	BA	Initialize port B
801CH				E2	
801DH				FF	
801EH		OUT	DX,AL	EE	Output the value to port B
801FH		MOV	CX,00FF	B9	Move the 00FF immediate value to C
8020H				FF	
8021H				00	
8022H	DLY2	LOOP	DLY2	E2	Stay here for while
8023H				FE	
8024H		JMP	START	EB	Repeat the program
8025H				E0	Indefinitely

RESULT:

Thus the program for generation of square wave form using 8086 microprocessor is executed successfully.

INTERFACING 8279



16

9/9/21

INTERFACING 8279 WITH 8086

AIM:

To display a characteristics of the first digit and using 8279 interfacing with 8086 microprocessor.

APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Adaptor	1
3.8279 interface	1

ALGORITHM:

1.Start the program.

2.Move the data into accumulator port clear the output port.

3.Move the data from accumulator to the output port.

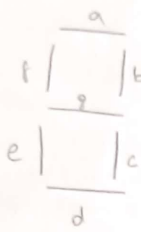
4.Display A in the output port.

5.Blank the rest of the display.

6.Stop the program.

Write	Display
1000	AA nn
1001	0000

Segment display section



a_0 1 1
a_1 1 2
a_2 1 3
a_3 1 4

h	g	f	e	d	c	b	a	ch	data
0	1	1	1	1	0	0	1	E	79
0	0	1	1	1	0	0	1	C	39
0	1	1	1	1	0	0	1	E	79
1	1	1	1	0	1	1	1	A	f7

O/P

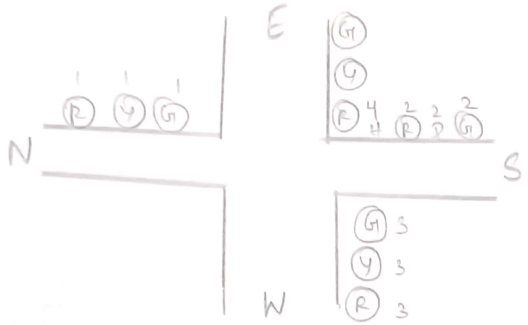
ECE-A

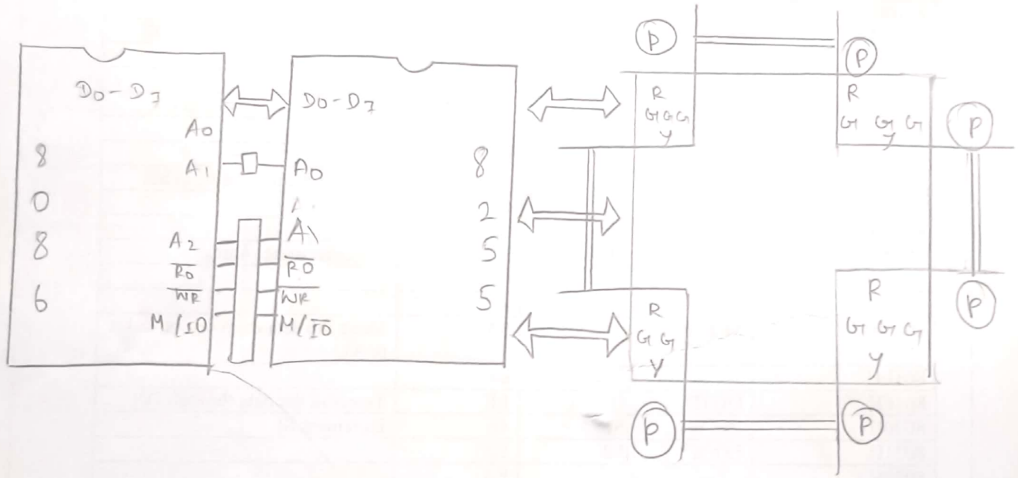
Address	Label	Mnemonics Opcode	Operand	Hex code	Comments
8000H		MOV	AL,90	B0	Move immediate value to acc
8001H				90	
8002H		MOVW	DX,0082	BA	Move the word to DX
8003H				82	
8004H				00	
8005H		OUTB	DX,AL	EE	Transmit the data through DX
8006H		MOVB	AL,00	B0	Clear the acc
8007H				00	
8008H		OUTB	DX,AL	EE	Transmit the data through DX
8009H		MOVW	CX,08	B9	Move immediate data to CX
800AH				08	
800BH				00	
800CH	L1	MOVB	AL,00	B0	initialize the acc
800DH				00	
800EH		MOVW	DX,80	BA	Move immediate data to DX
800FH				80	
8010H				00	
8011H		OUTB	DX,AL	EE	Transmit the data through DX
8012H		LOOP	L1	E2	
8013H				F8	
8014H		MOVW	CX,04	B9	Send the control word to CX
8015H				04	
8016H				00	
8017H		MOVW	AX,00	B8	Move immediate value to acc
8018H				00	
8019H				00	
801AH		MOVW	SI,8100	BE	Initialize source index
801BH				00	
801CH				81	
801DH	L2	MOVB	AL,[SI]	8A	Move the data from source index to AL
801EH				04	
801FH		OUTB	DX,AL	EE	Transmit the data through DX
8020H		INCW	SI	46	Increment SI
8021H		LOOP	L2	E2	
8022H				FA	
8023H		INT3		CC	interrupt

RESULT

Thus the program for interfacing display controller using 8086 microprocessor is executed successfully.

TRAFFIC LIGHT CONTROLLER





R - Red

P - Pedestrain crossing

Y - Amber

G - Green (left right straight)

17

9/9/21

TRAFFIC LIGHT CONTROLLER WITH 8086

AIM:

To interface traffic light conmtroller with 8086 microprocessor.

APPARATUS REQUIRED:

1.8086 Microprocessor kit	1
2.Adaptor	1
3.traffic light interface	1

ALGORITHM:

1.Start the program.

2.Transmit the control word to the traffic light interface through 8086

3.Display the signal.

4.Stop the program.

50

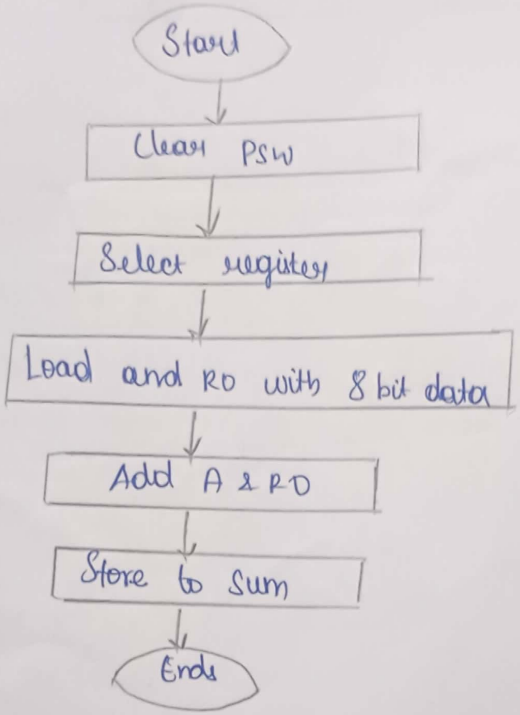
ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000	START:	MOV	BX, 8200H	BB	Move the memory address to bx
8001				00	
8002				82	
8003		MOV	CX, 0008H	B9	Move control word to cx
8004				08	
8005				00	
8006		MOV	AL,[BX]	8A	Move content in memory location
8007				07	
8008		MOV	DX, FF36	BA	Initialize port A
8009				36	
800A				FF	
800B		OUT	DX, AL	EE	Transmit data to interface
800C		INC	BX	43	Increment bx
800D	NEXT:	MOV	AL,[BX]	8A	Move content in memory location
800E				07	
800F		MOV	DX, FF30	BA	
8010				30	
8011				FF	
8012		OUT	DX,AL	EE	Transmit data to interface
8013		INC	BX	43	
8014		MOV	AL,[BX]	8A	Move content in memory location
8015				07	
8016		MOV	DX,FF32	BA	Initialize port B
8017				32	
8018				FF	
8019		OUT	DX,AL	EE	Transmit data to interface
801A		CALL	DELAY	E8	
801B				07	
801C				00	
801D		INC	BX	E2	Increment bx
801E				F1	
801F		LOOP	NEXT	E2	
8020				EB	
8021		JMP	START	EB	
8022				E4	
8023	DELAY:	PUSH	CX	51	Push the count value
8024		MOV	CX,0005H	B9	
8025				00	
8026				05	
8027	REPEAT:	MOV	DX,0FFFFH	BA	
8028				FF	
8029				FF	
802A	LOOP2:	DEC	DX	4A	decrement bx
802B		JNZ	LOOP2	75	
802C				FD	

[illegible]		LOOP	REPEAT	E2	
[illegible]				F8	
[illegible]		POP	DX	58	Pop the covent value
[illegible]		RET		C3	

Result:

Thus the traffic light controller is executed.

8-BIT ADDITION



19	
13/9/21	**8 BIT ADDITION USING 8051**

AIM:

To write the program for 8 bit addition using 8051 microcontroller kit.

APPARATUS REQUIRED:

1.8051 Microcontroller kit	1
2.Power supply	1
3.Opcode sheet	1

ALGORITHM:

1.Start the program
2.Move the data to memory pointer and then to accumulator
3.Move the accumulator value to R0 register
4.Increment DPTR by 1
5.Move DPTR to accumulator
6.Add accumulator and register value and jump to given loop
7Lload the pointer to accumulator
8.Increment DPTR by 1
9.Move the accumulator content to DPTR
10.Stop the program using interactive short jump

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8100	04	8200	06
8101	02		

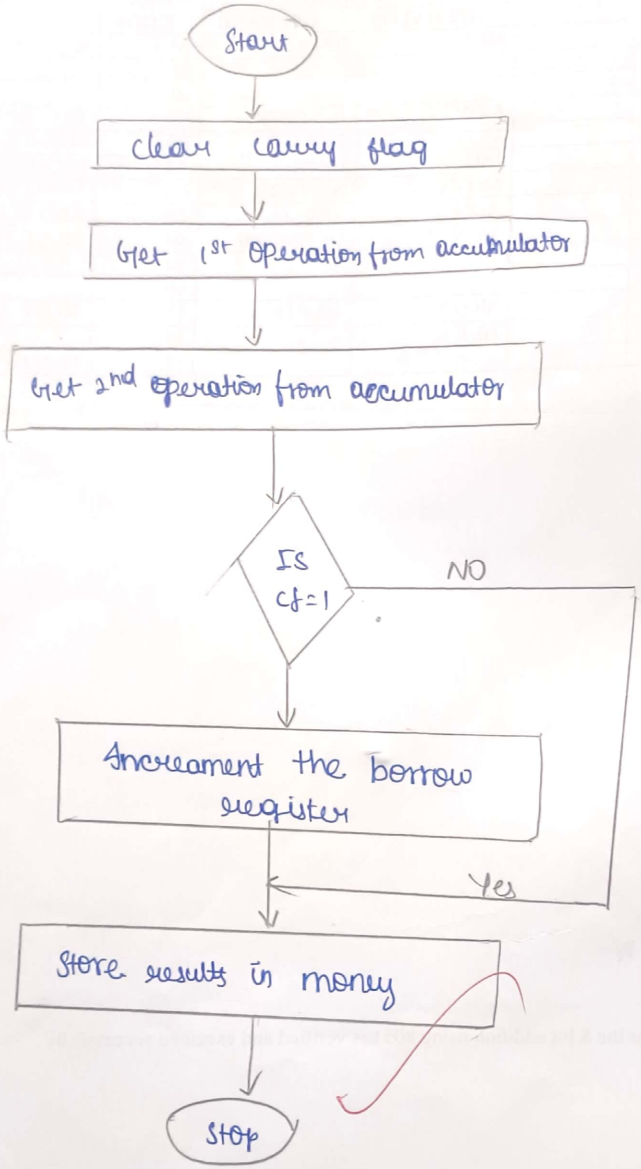
ADDITION PROGRAM FOR 8051

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	DPTR,#8100	90	MOVE THE DATA TO DPTR
8001				81	
8002				00	
8003		MOV	A,@DPTR	E0	MOV DPTR TO A
8004		MOV	R0,A	F8	MOVE A TO REGISTER
8005		INC	DPTR	A3	INCREEMENT DPTR
8006		MOVX	A,@DPTR	E0	MOVE DPTR TO A
8007		ADD	a,R0	28	ADD REGISTER TO A
8008		MOV	DPTR#8200	90	MOVE THE DATA TO DPTR
8009				82	
800A				00	
800B		MOVX	@DPTR,A	F0	MOVE A TO DPTR
800C		HLT		FF	
800D				FE	END OF THE PROGRAM

RESULT:

Thus the 8 bit addition using 8051 is verified and executed successfully.

8-BIT SUBTRACTION



19

13/9/21

8 BIT SUBTRACTION USING 8051

AIM:

To write the program for 8 bit subtraction using 8051 microcontroller

APPARATUS REQUIRED:

1.8051 Microcontroller kit 1

2.Power supply 1

3.Opcode sheet 1

ALGORITHM:

1.Start the program

2.Move the data to memory pointer and then to accumulator

3.Move the accumulator value to the R0 register

4.Increment the DPTR by 1

5.Move the DPTR to the accumulator

6.Subtract accumulator and register value and jump to given loop

7.Load the pointer to accumulator

8.Increment the DPTR by 1

9.Move the accumulator content to DPTR

10.Stop the program

AS

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8100	02	8200	02
8101	04		

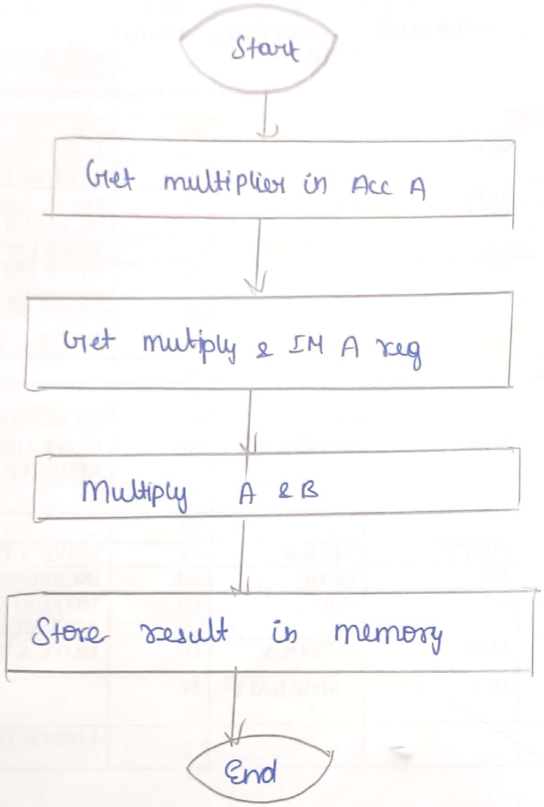
SUBTRACTION PROGRAM FOR 8051

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	DPTR,#8100	90	MOVE THE DATA TO DPTR
8001				81	
8002				00	
8003		MOVX	A,@DPTR	E0	MOVE DPTR TO ACCUMULATOR
8004		MOV	R0,A	F8	MOVE A TO REGISTER
8005		INC	DPTR	A3	INCREEMENT DPTR
8006		MOVX	A,@DPTR	E0	MOVE DPTR TO A
8007		MOV	R1,#00H	79	MOVE THE DATA TO REGISTER
8008				00	
8009		SUB	A,R0	98	SUBTRACT R0 FROM A
800A	LOOP1	JNC		50	
800B				01	
800C		INC	R1	09	INCREEMENT REGISTER
800D		MOV	DPTR#8200H	90	MOVE THE DATA TO REGISTER
800E				82	
800F				00	
8010		MOVX	DPTR,A	F0	MOVE A TO DPTR
8011		INC	DPTR	A3	INCREEMENT DPTR
8012		MOV	A,R1	E9	MOVE REGISTER TO ACCUMULATOR
8013		MOV	@DPTR,A	F0	MOVE A TO DPTR
8014		HLT	SJMP HALT	FE	
8015					END OF THE PROGRAM

RESULT:

Thus the program for 8 bit subtraction using 8051 is executed successfully

8 BIT MULTIPLICATION



20

13/9/21

8 BIT MULTIPLICATION USING 8051

AIM:

To write the program for 8 bit multiplication using 8051 microcontroller kit.

APPARATUS RQUIRED:

1.8051 Microcontroller kit	1
2. Power supply	1
3. Opcode sheet	1

ALGORITHM:

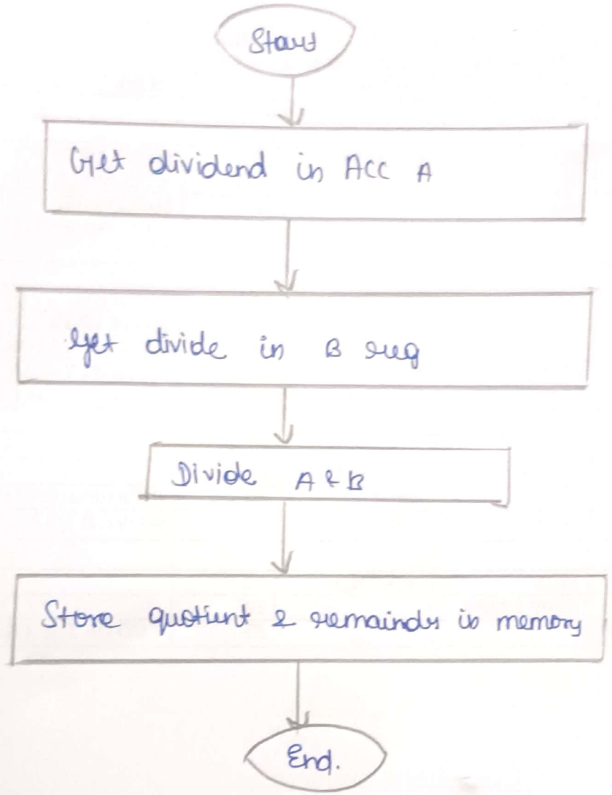
1. Start the program
2. Move the immediate data 8100H to data pointer
3. Store the address value as data pointer
4. Move the content of data pointer to a register through accumulator
5. Increment DPTR and multiply the two values
6. Store the results in 8200H address location
7. Stop the program

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8300	03	8200	06
8301	02		

RESULT:

Thus the program for 8 bit multiplication using 8051 is executed and verified successfully.

8 Bit Division



21

13/9/21

85

8 BIT DIVISION USING 8051

AIM:

To write the program for 8 bit division using 8051 microcontroller

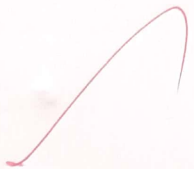
APPARATUS REQUIRED:

1.8051 microcontroller kit	1
2.power supply	1
3.Opcode sheet	1

ALGORITHM:

1.Start the program
2.Move immediate data to data pointer
3.Move the content of accumulator to B
4.increment the DPTR and move its content to A
5.Program division operation on A and B
6.Move the result to data pointer and then to accumulator
7.Increment DPTR and move the content to A
8.Stop the program

INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8300	02	8200	02
8301	04		



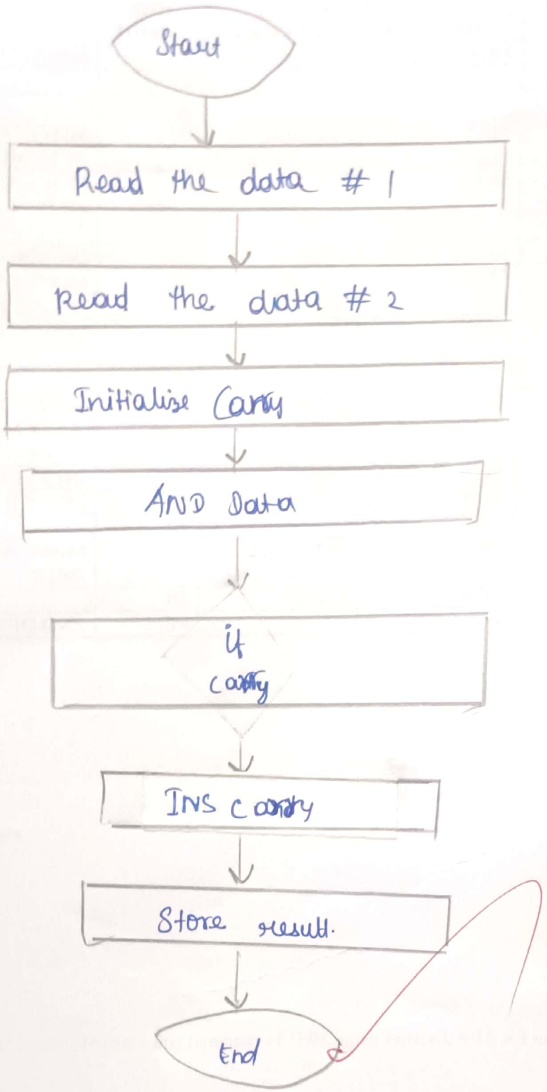
DIVISION PROGRAM FOR 8051

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	DPTR,#8300	90	MOVE THE DATATO DPTR
8001				83	
8002				00	
8003		MOV	A,@DMOVE ACCUMULATO	E0	MOVE DPTR TO ACCUMULATOR
8004		MOV	B,A	F5	MOVE ACCUMULATOR TO B
8005				F0	
8006		INC	DPTR	A3	INCREEMENT DPTR
8007		MOV X	A,@DPTR	E0	MOVE DPTR TO A
8008		DIV	A B	84	DIVIDE ABY B
8009		MOV	DPTR,#8200	90	MOVE THE DATA TO DPTR
800A				82	
800B				00	
800C		MOV X	@ DPTR , A	F0	MOVE THE ACCUMULATOR TO DPTR
800D		INC	DPTR	A3	INCREEMENT DPTR
800E		MOV	A,B	E5	MOVE B TO A
800F				F0	
8010		MOV X	@ DPTR,A	F0	MOVE ACCUMULATOR TO DPTR
8011	LABEL	SJMP	LABEL	80	
8012				FE	END OF THE PROGRAM

RESULT:

Thus the program for 8 bit division using 8051 is executed and verified successfully.

8-BIT AND OPERATION



22

7/10/21

AND OPREATION USING 8051

AIM:

To perform 8 bit logical AND operation using 8051 microcontroller

APPARATUS REQUIRED:

1.8051 microcontroller kit	1
2.Power supply	1
3.Opcode sheet	1

ALGORITHM:

1.Start the program

2.Get the input data from memory location into DPTR register

3.Move the content of DPTR into the accumulator

4.Store the content of accumulator into R0 register

5.Increment the content of DPTR

6.AND the content of the R0 register with the content of accumulator

7.Move the content of 8200H into DPTR

8.Stop the program.

INPUT :

8100 0 3 ⇒ 0000 0011

8101 0 2 ⇒ 0000 0001

OUTPUT :

8200 01 ⇒ 0000 0001

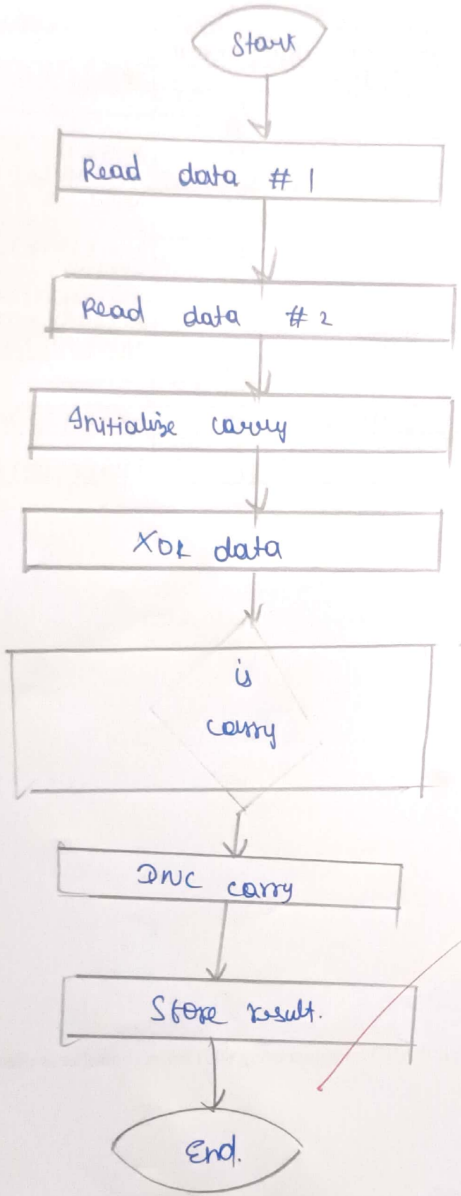
AND OPERATION PROGRAM FOR 8051

ADDRESS	LABEL	MNEMONICS		REL CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	DPTR,#01H	90	MOVE THE DATA TO DPTR
8001				01	
8002				00	
8003		MOV	A,@DPTR	E0	MOVE THE DPTR TO ACCUMULATOR
8004		MOV	R0,A	F8	MOVE THE A TO REGISTER0
8005		INC	DPTR	A3	INCREMENT DPTR
8006		MOVX	A, @DPTR	E0	MOVE DPTR TO ACCUMULATOR
8007		AND	A,R0	58	AND REGISTER TO ACCUMULATOR
8008		MOV	DPTR,#0200	90	MOVE THE DATA TO DPTR
8009				02	
800A				00	
800B		MOVX	@DPTR,A	F0	MOVE A TO DPTR
800C	HALT	SJMP HALT		80	
800D				FE	END OF THE PROGRAM

RESULT:

Thus the program for logical AND operation using 8051 microcontroller is executed successfully.

8-BIT XOR OPERATION



23

7/10/21

XOR OPERATION USING 8051

AIM:

To perform the 8 bit XOR operation using 8051 microcontroller

APPARATUS REQUIRED:

1.8051 microcontroller kit	1
2.Power supply	1
3.Opcode sheet	1

ALGORITHM:

1.Start the program

2.Stroe the content of 8100 to R0

3.store the content 8101 to A

4.Perform logical XOR operation of A and R0

5.Store the result in 8200 memory location

6.Stop the program

I/P :

8100 03 ⇒ 0000 0011

8101 01 ⇒ 0000 0001

O/P :

8200 02 ⇒ 0000 0010

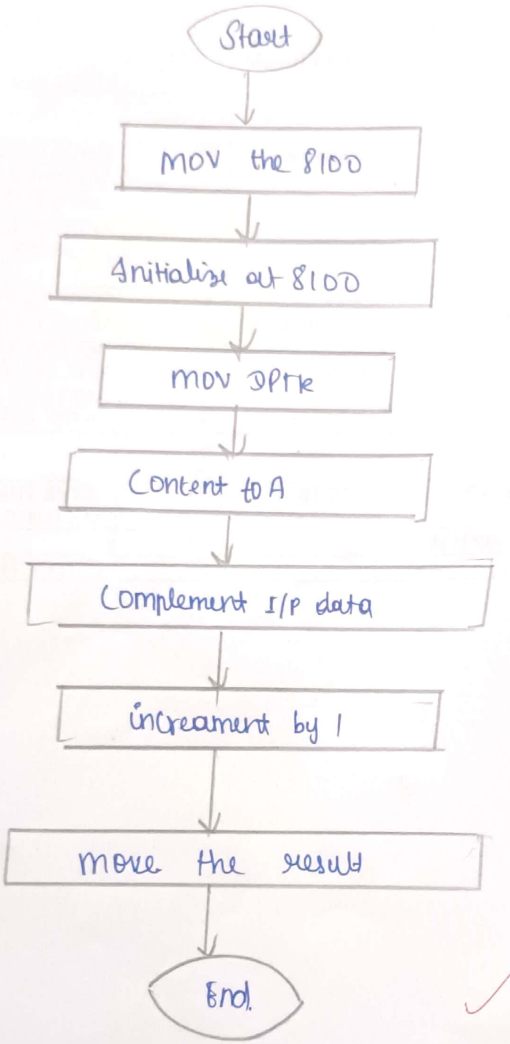
EX-OR OPERATION PROGRAM FOR 8051

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	DPTR,#8100	90	MOVE THE DATA TO DPTR
8001				81	
8002				00	
8003		MOV	A,@DPTR	E0	MOVE DPTR TO ACCUMULATOR
8004		MOV	R0,A	F8	MOVE THE ACCUMULATOR TO REGISTER
8005		INC	DPTR	A3	INCREEMENT DPTR
8006		MOVX	A, @DPTR	E0	MOVE DPTR TO ACCUMULATOR
8007		XOR	A,R0	68	XOR REGISTER TO ACCUMULATOR
8008		MOV	DPTR,#8200	90	MOVE THE DATA TO DPTR
8009				82	
800A				00	
800B		MOVX	@DPTR,A	F0	MOVE THE ACCUMULATOR TO DPTR
800C	HALT	SJMP HALT		80	
800D				FE	END OF THE PROGRAM

RESULT:

Thus the program for logical XOR operation using 8051 microcontroller is executed successfully

2's Complement



2's COMPLEMENT USING 8051

AIM:

To perform the 2's complement operation using 8051 microcontroller

APPARATUS REQUIRED:

1.8051 microcontroller kit	1
2.Power supply	1
3.Opcode sheet	1

ALGORITHM:

1.Start the program

2.Store the content DPTR

3.store the DPTR content to A

4.Complement the Accumulator content.

5.Add immediate date 01 to accumulator content.

6.Store the result in 8200 memory location

7.Stop the program

INPUT :

ADDRESS	DATA
8100	03

OUTPUT :

ADDRESS	DATA
8101	FD

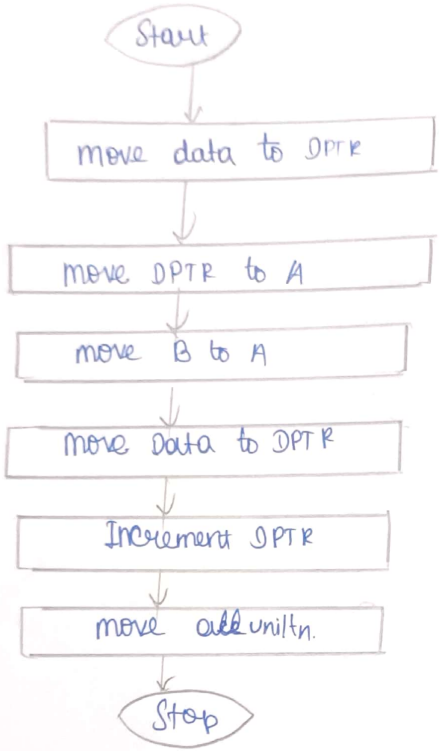
2'S COMPLEMENT PROGRAM FOR 8051

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	DPTR,#8100H	90	MOVE THE DATA TO DPTR
8001				81	
8002				00	
8003		MOVX	A,@DPTR	E0	MOVE THE DPTR TO ACCUMULATOR
8004		CPL	A	F4	
8005		ADD	A,#01	24	ADD THE DATA TO ACCUMULATOR
8006				01	
8007		MOV	DPTR,#8200H	90	MOVE THE DATA TODPTR
8008				82	
8009				00	
800A		MOVX	@DPTR,A	F0	MOVE THE ACCUMULATOR TO DPTR
800B	HALT	SJMP HALT		80	
800C				FE	END OF THE PROGRAM

RESULT:

Thus the program for 2's complement using 8051 microcontroller is executed successfully

SQUARE OF A GIVEN NUMBER



25

21/10/21

FINDING SQUARE OF A NUMBER USING 8051

AIM:

To perform the square of a given number using 8051 microcontroller.

APPARATUS REQUIRED:

1.8051 microcontroller kit	1
2.Power supply	1
3.Opcode sheet	1

ALGORITHM:

1.Start the program
2.Store the content to DPTR
3.Move the content of DPTR to A.
4.Move the content of A to B.
5.Multiply A and B
6.Store the result in 8200 memory location
7.Stop the program

INPUT :

ADDRESS	DATA
8100	02

OUTPUT :

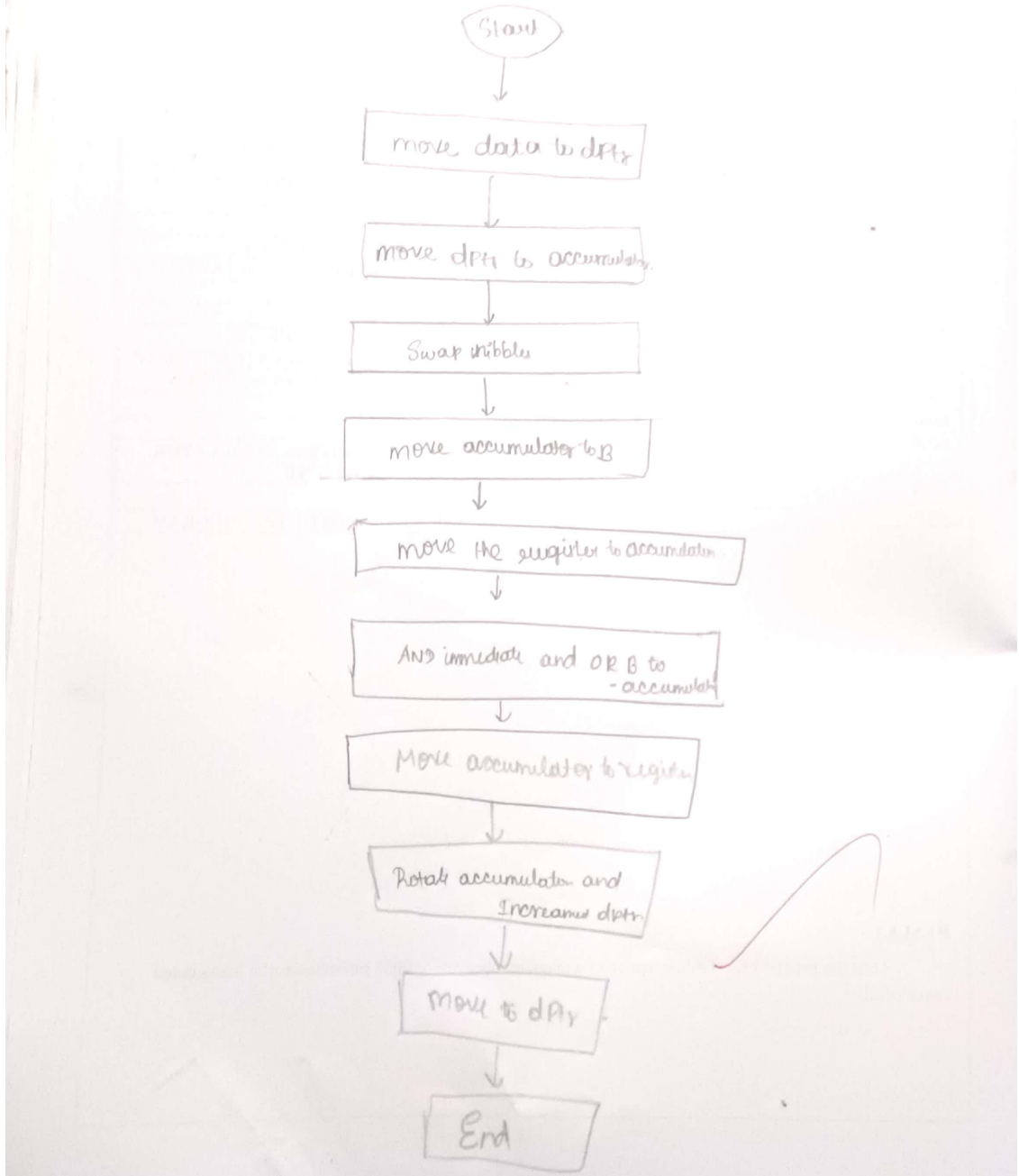
ADDRESS	DATA
8200	04

TO FIND SQUARE OF A GIVEN NUMBER PROGRAM FOR 8051

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	DPTR,#8300H	90	MOVE THE DATA TO DPTR
8001				83	
8002				00	
8003		MOV	A,@DPTR	E0	MOVE DPTR TO A
8004		MOV	B,A	F5	MOVE A TO B
8005				F0	
8006		MUL	A,B	A4	MOVE B TO A
8007		MOV	DPTR,#8200H	90	MOVE THE DATA TO DPTR
8008				82	
8009				00	
800A		MOVX	@DPTR,A	F0	MOVE THE ACCUMULATOR TO DPTR
800B		INC	DPTR	A3	INCREEMENT DPTR
800C		MOV	A,B	E5	
800D				F0	
800E		MOVX	@DPTR,A	F0	MOVE ACCUMULATOR TO DPTR
800F	LABEL	SJMP LABEL		80	
8010				FE	END OF THE PROGRAM

RESULT:

Thus the program for finding square of a given number using 8051 microcontroller is executed successfully



26

21/10/21

BCD TO ASCII CONVERSION USING 8051

AIM:

To perform the BCD to ASCII conversion using 8051 microcontroller

APPARATUS REQUIRED:

1.8051 microcontroller kit	1
2.Power supply	1
3.Opcode sheet	1

ALGORITHM:

1.Start the program

2. Move the data to dptr

3. Move dptr to accumulator

4. And immediate data to accumulator

5. Swap nibbles within the accumulator

6. Move accumulator to B and Move the register to accumulator

7.And immediate data to accumulator and Or B to accumulator

8.Move accumulator to register and AND immediate data to accumulator and move register to accumulator

9.Rotate accumulator right and Increment dptr

10.Move a to dptr

11.End of the program

Input Address	Data	Output Address	Data
8100	08	8101	38
		8102	33

BCD TO ASCII PROGRAM FOR 8051

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	DPTR,#8200	90	MOVE THE DATA TO DPTR
8001				82	
8002				00	
8003		MOV	A,@DPTR	E0	MOVE DPTR TOACC
8004		ANL	A,#0F	54	AND IMMEDIATE DATA TO ACCUMULATOR
8005				0F	
8006		SWAP	A	C4	SWAP NIBBLES WITHIN TH ACCUMULATOR
8007		MOV	B,A	F5	MOVE ACCUMULATOR TO B
8008				FO	
8009		MOV	A,R4	EC	MOVE THE REGISTER TO ACCUMULATOR
800A		ANL	A,#0F	54	AND IMMEDIATE DATA TO ACCUMULATOR
800B				0F	
800C		ORL	A,B	45	OR B TO ACCUMULATOR
800D				F0	
800E		MOV	R2,A	FA	MOVE ACC TO REGISTER
800F		ANL	A,#0F	54	AND IMMEDIATE DATA TO ACCUMULATOR
8010				0F	
8011		ORL	A,#30	44	OR IMMEDIATE DATA TO ACCUMULATOR
8012				30	
8013		M0V	R6,A	FE	MOVE ACCUMULATOR TO REGISTER
8014		MOV	A,R2	EA	MOVE REGISTERTO ACC
8015		ANL	A,#F0	54	
8016				F0	
8017		RRA		03	ROTATE ACCUMULATOR RIGHT
8018		RRA		03	
8019		RRA		03	
801A		RRA		03	
801B		ORL	A,#30	44	
801C				30	
801D		INC	DPTR	A3	INCREEMENT DPTR
801E		MOV	@DPTR,A	F0	MOVE A TO DPTR
801F	HERE	SJMP HERE		80	
8020				FE	END OF THE PROGRAM

RESULT:

Thus the program for BCD to ASCII conversion using 8051 microcontroller is executed successfully

30

Start
Move the data to accumulator
Rotate the register towards right
Complement the register
Add Swap
Store the result
END

27

21/10/21

BIT MANIPULATION USING 8051

AIM:

To write a program for a 8 bit manipulation using 8051 microcontroller

ALGORITHM:

1.8051 microcontroller kit 1

2.power supply

3.Opcode sheet 1

ALGORITHM:

1. Start the program.
2. Move the data to the accumulator.
3. Rotate the register towards the right.
4. Complement of the register.
5. AND swap the register.
6. Store the result.
7. Stop the program.

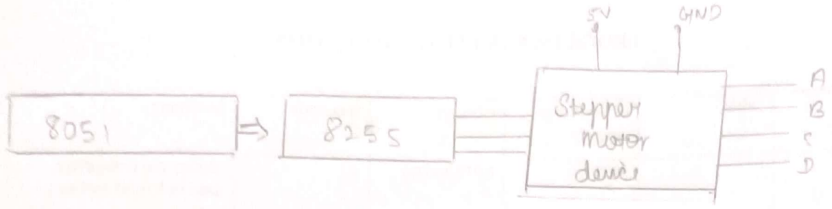
INPUT ADDRESS	DATA	OUTPUT ADDRESS	DATA
8100	03	8200	D7

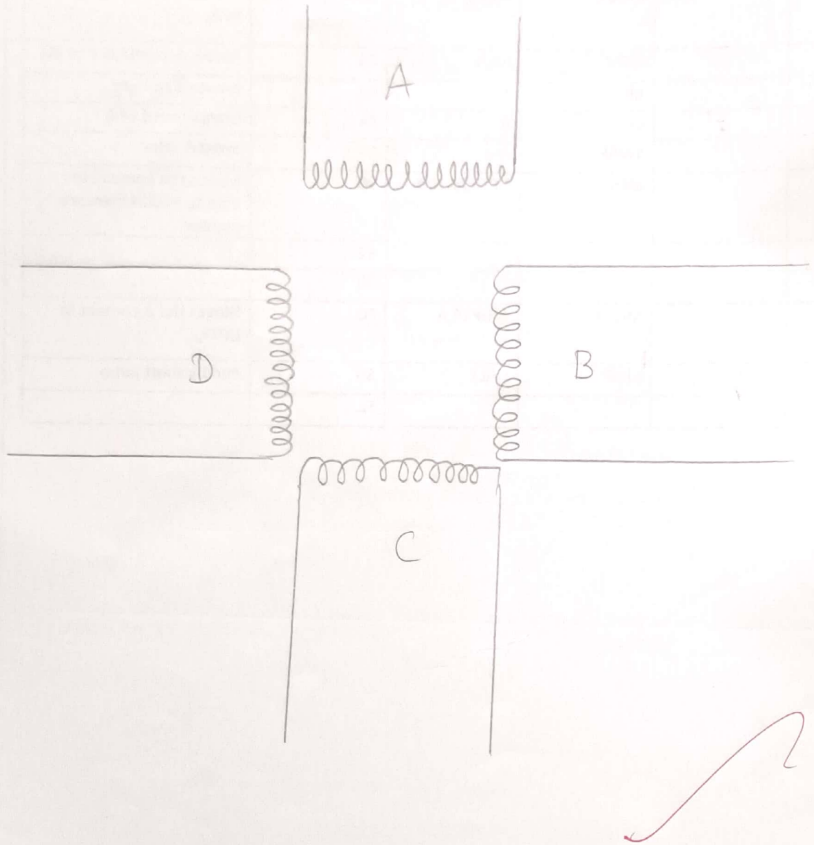
PROGRAM FOR BIT MANIPULATION

Address	Label	Mnemonics Opcode	Operand	Hex code	Comments
8000H		MOV	DPTR,#8100	90	Moves the immediate data to 8100H memory pointer.
8001H				81	
8002H				00	
8003H		MOVX	A,@DPTR	E0	Moves the DPTR content to A.
8004H		MOV	R0,A	F8	Moves accumulator to R0.
8005H		RR	A	03	Rotate A to right.
8006H		CPL	A	F4	Complement of A.
8007H		SWAP	A	C4	Swap A value.
8008H		MOV	DPTR,#8200	90	Moves the immediate data to 8200H memory pointer.
8009H				82	
800AH				00	
800BH		MOVX	@DPTR,A	F0	Moves the A content to DPTR.
800CH	HALT	SJMP	HALT	80	End by short jump.
800DH				FE	

RESULT:

Thus the program bit manipulation with 8051 microcontroller is executed.





28

28/10/21

STEPPER MOTOR USING 8051

AIM:

To write an ALP using 8051 microcontroller to run stepper motor with delay timer

ALGORITHM:

1.8051 microcontroller kit 1

2.power supply

3.Opcode sheet 1

4.Stepper motor 1

ALGORITHM:

1.Start the program

2.load data in A and CWR

3.Create the loop which transistor the control to the given when ZF=0

4.Send the data in A to port A and call the subroutine

5.Call the delay program

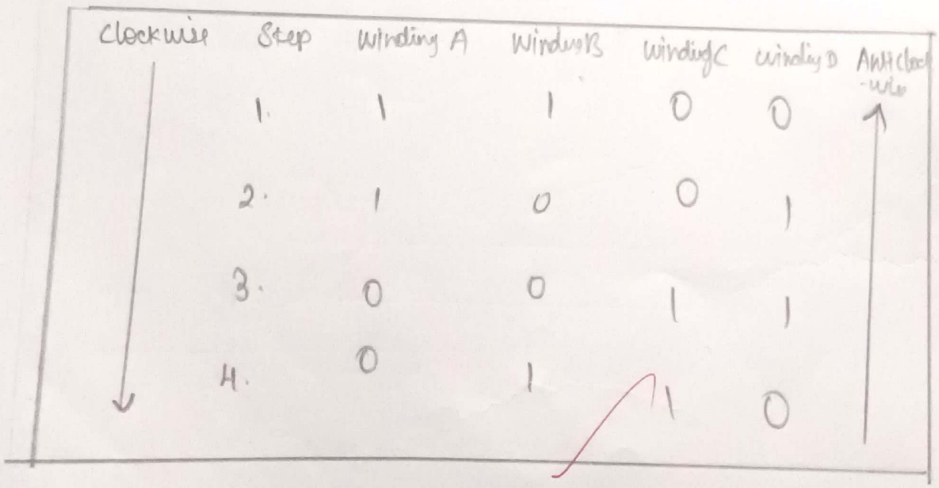
6.Create another loop in which logical OR operation is performed between accumulator and D register

7.Transfer the control to main program

8.Stop the program

Single angle	Step per revolution
0.72	500
1.8	200
2.0	180
2.5	144
5.0	72
7.5	48
15	24

NORMAL FOUR STEP SEQUENCE



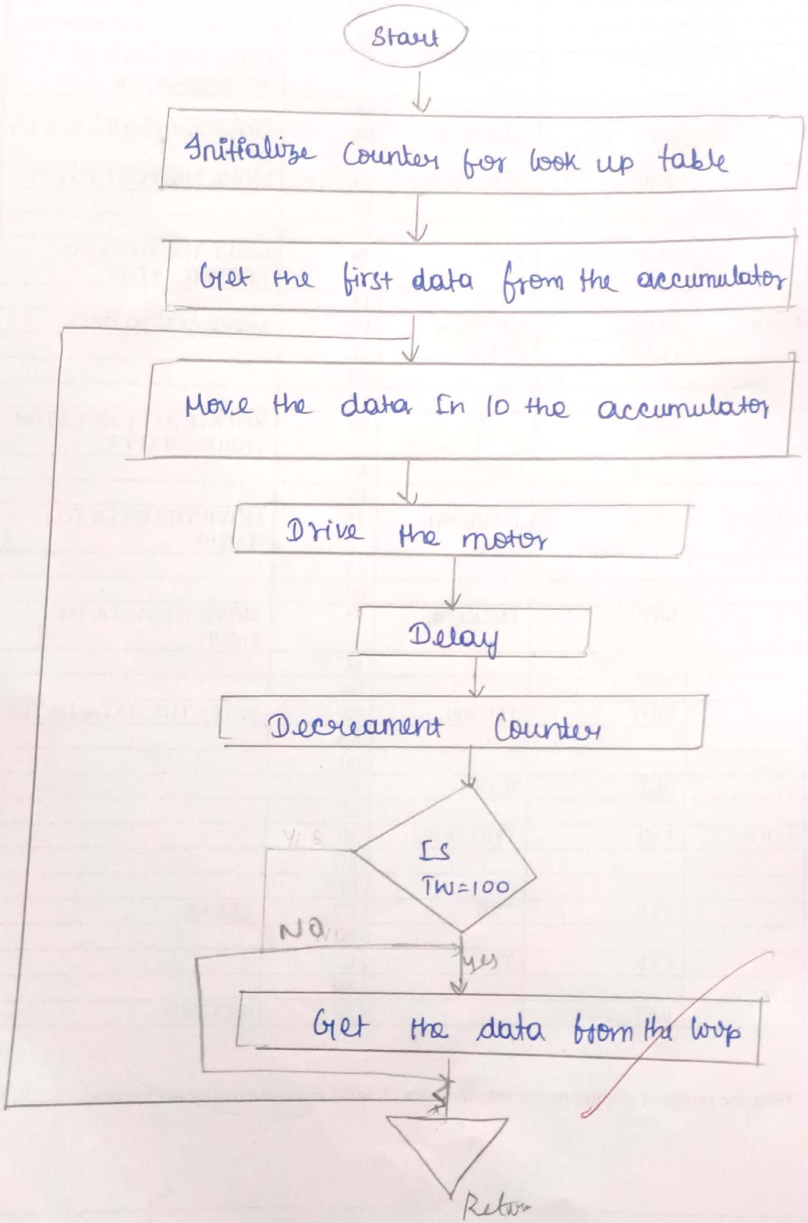
PROGRAM FOR STEPPER MOTOR CONTROL

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000		MOV	DPTR,#CNR	90	MOVE CNR TO DPTR
8001				E8	
8002				03	
8003		MOV	A,#80	74	MOVE THE DATA TO ACCUMULATOR
8004				80	
8005		MOV	@DPTR,A	F0	MOVE ACCUMULATOR TO DPTR
8006		MOV	DPTR,#PORT	90	MOVE THE PORT TO DPTR
8007				E8	
8008				00	
8009		MOV	A,#11	74	MOVE THE DATA TO ACCUMULATOR
800A				11	
800B	LOOP1	MOV X	@DPTR,A	F0	MOVE ACC TO DPTR
800C		LCALL	DELAY	12	
800D				80	
800E				12	
800F		RRA		03	ROTATE ACCUMULATOR THROUGH LEFT
8010		SJMP	LOOP1	80	
8011				F9	
8012	DELAY	MOV	TMOD,#01	75	MOVE THE DATA TO TMOD
8013				89	
8014				01	
8015		MOV	TMOD,#05	75	MOVE THEDATA TO TMOD
8016				8C	
8017				05	
8018		MOV	TL0,#00	75	MOVE THE DATA TO TL0
8019				8A	
801A				00	
801B		SET	B,TR0	D2	
801C				8C	
801D	LOOP2	JNB	TF0,LOOP2	30	
801E				8D	
801F				FD	
8020		CLR	TR0	C2	CLEAR
8021				8C	
8022		CLR	TF0	C2	
8023				80	
8024		RET		22	RETURN

RESULT:

Thus the program stepper motor interfacing with 8051 microcontroller is executed.

FLOW CHART TO EECUTE TIMER :



29

28/10/21

PROGRAM TO VERIFY TIMER

AIM:

To write a program to execute the timer using 8051 microcontroller kit.

ALGORITHM:

1.8051 microcontroller kit 1

2.power supply 1

3.Opcode sheet

ALGORITHM:

1.The counter is set for the multiple delay.

2. Assign TL1=08, TH1=01, set the timer as 1.

3. Stay until timer rolls over.

4. Stop the timer 1.

5. Complement p1 to toggle.

6. If r3 is not equal to zero they reload the timer.

7. Rotate the loop indefinitely.

8.Stop the program.

OUTPUT :

Hour : 002

Minutes : 05

VERIFYING TIMER IN 8051

ADDRESS	LABEL	MNEMONICS		HEX CODE	COMMENTS
		OPERATOR	OPERAND		
8000	HERE	MOV	TMOD,#10	75	MOVE THE DATA TO TMOD
8001				89	
8002				10	
8003		MOV	83,#C3	78	MOVE C3 TO DATA
8004				C8	
8005		AGAIN	TL1,08	75	MOVE THE DATA TO TL1
8006				8B	
8007				08	
8008		MOV	TH1,01	75	MOVE THEDATA TO TH1
8009				8D	
800A				01	
800B		SET	B TR1	D2	SET B TR1
800C				8E	
800D	BACK	JNB	TF1 , BACK	30	
800E				8F	
800F				FD	
8010		CLR	TR1	C2	CLEAR
8011				8E	
8012		CPL	P1.0	B2	
8013				90	
8014		CLR	TF1	C2	
8015				8F	
8016		DJNZ	R3,AGAIN	D3	
8017				ED	
8018		SJMP	HERE	80	
8019				E6	

RESULT :

Thus the program for verifying the timer using 8051 microcontroller is executed.

INPUT :

LIST 534, 154, 194, 02H

OUTPUT :

8300 01

30

4/4/24

8086 STRING MANIPULATION – SEARCH AWORD

AIM:

To search a word from the string using MASM.

ALGORITHM:

1. Load the source and destination index register with starting address and destination address respectively.
2. Initialize the counter with the total number of words to be copied.
3. Clear the destination with the total incrementing node.
4. Use the string manipulation instruction SCASW with the prefix REP to search a word from string.
5. If the match is found (z=1), display 01 in destination address. Otherwise, display 00 in destination address

PROGRAM:

```
ASSUME CS:CODE, DS : DATA
DATA SEGMENT
LIST DW 53H, 15H, 19H, 02H
DEST EQU 3000H
COUNT EQU O5H
DATA ENDS
CODE SEGMENT
START:      MOV AX, DATA
 MOV DS , AX
MOV AX, 15H
MOV SI, OFFSET LIST
MOV DI, DEST
MOV CX, COUNT
MOV AX, 00
CLD
REP         SCASW
JZ LOOP
MOV AX, 01
LOOP        MOV [DI], AX
MOV AH, 4CH
INT 21H
CODE ENDS
END START
```

RESULT:

Thus the program for string manipulation is implemented and the searching word is found from the string.

31

4/11/21

STRING TRANSFER

AIM:

To find and replace word from the string using MASM.

ALGORITHM:

1. Load the source index register with the starting address and the ending address respectively.
2. Initialize the counter with the total number of words to be copied.
3. Clear the direction flag for auto incrementing mode of transfer.
4. Use the string manipulation instruction SCASW with the prefix REP to search a word from string
5. If a match is found (z=1), replace the old word with the current word in the destination address, otherwise stop.

OUTPUT:

String 1 : DHSCF2 , DFM , OF . ECES.

PROGRAM:

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT
        STRING 1    DB      'BMSCE DEPT OF ECE$'
        LENGTH      EQU     ($-STRING 1)
        STRING 2    DB      LEN DUP (0)
DATA ENDS
CODE SEGMENT
START:  MOV AX,@DATA
        MOV DS, AX
        MOV ES, AX
        MOV CX, LENGTH
        CLD
        LEA SI,STRING 1
        LEA DI, STRING 2
        REP MOVSB
        MOV AH, 4CH
        INT 21H
CODE ENDS
END
```

RESULT:

Thus the program for string transfer is implemented and the word is found and replaced from the string.

32

4/11/21

8086 STRING MANIPULATION – SORTING

AIM:

To sort a group of data bytes using MASM.

ALGORITHM:

1. Place all the elements of an array named list(in the consecutive memory location)
2. Do the following steps until the counter C reaches 0
3. Compare the accumulator content with the next element present in the next memory location.

If the accumulator content is smaller go to next step, otherwise , swap the content of accumulator with the content of memory location.

4. Increment the memory pointer to point the next element.
5. Decrement the counter C by 1.
6. Stop the program.

I/P DATA:

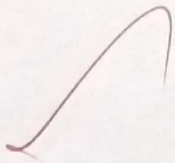
Descending

53H
25H
19H
02H

OUTPUT DATA:

Ascending

02H
19H
25H
53H



PROGRAM:

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT
LIST DW 53H, 25H, 19H, 02H
COUNT EQU 04H
DATA ENDS
CODE SEGMENT
START:  MOV AX,DATA
        MOV DS, AX
        MOV DX, COUNT-1
LOOP2:  MOV CX, DX
        MOV SI, OFFSET LIST
AGAIN:  MOV AX, [SI]
        CMP AX, [SI+2]
        JC LOOP1
        XCHG [SI+2], AX
        XCHG [SI], AX
LOOP1:  ADD SI, 02
        LOOP AGAIN
        DEC DX
        JNZ LOOP2
        MOV AH, 4CH
        INT 21H
CODE ENDS
END START
```

RESULT:

Thus the program for sorting is implemented and the given array is sorted successfully.

Find

Input : 05 = 0000 0101

0000 1111

Input = 0000 0110 06

0000 0100 04

output :

0000 0010 02.

32

25/11/21

LOGICAL CONTROLLER

AIM:

To write a program for performing logical AND operation using MASM.

ALGORITHM:

1. Start the program
2. Get input data from memory location
3. Store the content of accumulator to AX
4. Move the content of memory location
5. Increment content of DX
6. AND content of Al register with data
7. Stop the program

I/P :

0A = 0000 1010

O/P:

10.

PROGRAM:

```
CODE SEGMENT
ASSUME CS:CODE, DS: CODE, SS: CODE,ES: CODE
MOV AL,05H
MOV BL,06H
AND AL, 0FH
XOR BL,04H
MOV AH,4CH
INT 21H
CODE ENDS
END
```

RESULT:

Thus the program for logical controller is implemented successfully

INPUT :

0A : 0000 0010

OUTPUT :

10 : 0001 0000

34

25/11/24

BINARY TO BCD CONVERSION

AIM:

To write a program for performing binary to BCD conversion using MASM.

ALGORITHM:

1. Start the program.
2. Get input data from memory function
3. Store content of accumulator to ax
4. Move the content of memory location
5. Increment the content of C
6. OR the content of AL register with data
7. Stop the program.

PROGRAM:

```
        ASSUME CS: CODE, DS: DATA
        DATA SEGMENT
        BINARY      DB    63H
        ANS         DB    00H, 00H,, 00H
        DATA ENDS
        CODE SEGMENT
START:  MOV AX, @DATA
        MOV DS,AX
        MOV AX, 00H
        MOV AL, BINARY
        MOV CL, 64H
        DIV CL
        MOV BCD, AL
        MOV AL, AH
        MOV AH, 00H
        MOV CL, 0AH
        DIV CL
        MOV ANS+ 1, AL
        MOV ANS+2,AH
        OR ANS, 30H
        OR ANS+ 1, 30H
        OR ANS+ 2,30H
        MOV AH, 4CH
        INT 21H
        CODE ENDS
        END
```

RUSULT:

Thus the binary number is converted into BCD successfully.

35

25/11/21

8086 STRING MANIPULATION – FIND AND REPLACE THE WORD

AIM :

To find replace the word from the string using MASM.

ALGORITHM:

1. Load the source index register with the starting address and the ending address respectively.
2. Initialize the counter with the total number of words to be copied.
3. Clear the direction flag for auto incrementing mode of transfer.
4. Use the string manipulation instruction SCASW with the prefix REP to search a word from string
5. If a match is found (z=1), replace the old word with the current word in the destination address, otherwise stop.

INPUT :

LIST : 53H, 15H, 19H, 02H

OUTPUT :

Find the word 15H & replace by 30H

List : 53H, 30H, 19H, 02H.

PROGRAM:

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT
LIST DW 53H, 15H, 19H, 02H
REPLACE EQU 30H
COUNT EQU 05H
DATA ENDS
CODE SEGMENT
START:  MOV AX, DATA
        MOV DS, AX
        MOV AX,15H
        MOV SI, OFFSET LIST
        MOV CX, COUNT
        MOV AX, 00
        CLD
REP     SCASW
        ZNZ LOOP
        MOV DI, LABEL LIST
        MOV [DI], REPLACE
LOOP    MOV AH, 4CH
        INT 21H
CODE ENDS
END START
```

RESULT:

Thus the word is found and replaced from the string successfully.

36	
2/12/21	**16 BIT ADDITION USING MASM**

AIM:

To write a program for performing addition using MASM.

ALGORITHM:

1.start the program.

2. Get input data from memory function

3.Add the data

3. Store content of accumulator to memory location

4. Stop the program.

INPUT :

ADDRESS	SEGMENT	OFFSET
0B63	0000	33
	0001	33
	0002	22
	0003	22

OUTPUT :

ADDRESS	SEGMENT	OFFSET
	0004	55
	0005	55

PROGRAM:

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT
LIST DW  0102H, 1215H
SUM DW 2H DUP(0)
DATA ENDS
CODE SEGMENT
START:    MOV AX, DATA
          MOV DS, AX
          MOV AX,LIST
          MOV BX, 00H
          ADD AX,LIST+2
          JNC LOOP
          INC BX
LOOP:     MOV SUM AX
          MOV SUM+2,BX
          MOV AH, 4CH
          INT 21H
CODE ENDS
END START
```

RESULT:

Thus the program for Addition is implemented successfully.

16-BIT SUBTRACTION USING MASM

AIM:

To execute a program for performing subtraction using MASM.

ALGORITHM:

1. start the program
2. Get input data from memory function
3. Sub the data
3. Store content of accumulator to memory location
4. Stop the program

INPUT :

OFFSET	SEGMENT	DATA
DB b3		
	0000	33
	0001	33
	0002	22
	0003	22

OUTPUT :

SEGMENT	DATA
0004	11
0005	11

PROGRAM:

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT
LIST DW 8982H, 1215H
SUB DW 2H DUP(0)
DATA ENDS
CODE SEGMENT
START:  MOV AX, DATA
        MOV DS, AX
        MOV AX,LIST
        MOV BX, 00H
        SUB AX,LIST+2
        JNC LOOP
        INC BX
LOOP:   MOV SUB AX
        MOV SUB+2,BX
        MOV AH, 4CH
        INT 21H
CODE ENDS
END START
```

RESULT:

Thus the program for subtraction is implemented successfully.

38

2/12/24

16 BIT MULTIPLICATION USING MASM

AIM:

To write a program for performing multiplication using MASM.

ALGORITHM:

1.start the program.

2. Get input data from memory function

3.Multiply the data

3. Store lower order bit to memory location through accumulator.

4.Store higher order bit to memory location through DX

5. Stop the program.

INPUT :

OFFSET	SEGMENT	DATA
DB63	0000	42
	0001	00
	0002	22
	0003	00

OUTPUT :

SEGMENT	DATA
0004	C4
0005	08

PROGRAM:

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT
LIST DW  0002H, 0008H
MULTIPLY  DW 04H DUP(0)
DATA ENDS
CODE SEGMENT
START:      MOV AX, DATA
            MOV DS, AX
            MOV AX,LIST
            MOV BX, LIST+2
            MUL BX
            MOV MULTIPLY, AX
            MOV MULTIPLY+2,DX
            MOV AH, 4CH
            INT 21H
CODE ENDS
END START
```

RESULT:

Thus the program for Multiplication is implemented successfully.

39

9/12/24

16 BIT DIVISION USING MASM

AIM:

To write a program for performing division using MASM.

ALGORITHM:

1.start the program.

2. Get input data from memory function

3.Divide the data

3. Store quotient bit to memory location through accumulator.

4.Store remainder bit to memory location through DX

5. Stop the program.

INPUT :

OFFSET	SEGMENT	DATA
0B63	0000	49
	0001	00
	0002	05
	0003	00

OUTPUT :

SEGMENT	DATA
0004	0E
0005	00
0006	03
0007	00

PROGRAM:

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT
LIST DW 0002H, 0008H
QUO DW 2H DUP(0)
REM DW 2H DUP(0)
DATA ENDS
CODE SEGMENT
START:  MOV AX, DATA
        MOV DS, AX
        MOV AX,LIST
        MOV BX, LIST+2
        DIV BX
        MOV QUO, AX
        MOV REM,DX
        MOV AH, 4CH
        INT 21H
CODE ENDS
END START
```

RESULT:

Thus the program for division is implemented successfully.

40

9/12/21

3X3 MATRIX USING MASM

AIM:

To write a program form 3x3 matrix using MASM.

ALGORITHM:

1.start the program.

2. Get 9 input data from memory function

3.Arrange the data in rows and columns

4. Stop the program.

PROGRAM

```
ASSUME DS:DATA , CS:CODE, SS:STACK

DATA SEGMENT
MATRIX DB 1, 2, 3, 4, 5, 6, 7, 8, 9
DATA ENDS
STACK  SEGMENT
DW 128 DUP(0)
STACK ENDS
CODE SEGMENT

START:      MOV AX, DATA
            MOV DS, AX
            MOV SS, AX
            MOV BX, MATRIX
            MOV CH,03H
            MOV CL, 00H
COLUMN1:    MOV DL, [BX]
            ADD DL, 30H
            MOV AH, 02H
            INT 21H
            INC CL
            CMP CL, CH
            JGE ROW1
            ADD BX, 03H
            JMP COLUMN1
ROW1:       INC CL
            INC BX
            MOV DL, [BX]
            ADD DL, 03H
            MOV AH, 02H
            INT 21H
            CMP CL, 5
            JE COLUMN1
            JNE ROW1
COLUMN2:    INC CL
            SUB BX, 03H
            MOV DL, [BX]
            ADD DL, 03H
            MOV AH, 02H
            INT 21H
            CMP CL, 07H
            JNE COLUMN2

            JE ROW2
ROW2:       DEC BX
            MOV DL, [BX]
            ADD DL, 30H
            MOV AH, 02H
            INT 21H
            JMP MIDDLE
```

INPUT: 1 2 3 4 5 6 7 8 9

OUTPUT:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

```
MIDDLE:     ADD BX, 1
            MOV DL, [BX]
            ADD DL, 30H
            MOV AH, 02H
            INT 21H
            JMP END
END:
```

RESULT:

Thus the program for 3X3 matrix is implemented successfully.