SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
SAVEETHA SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ECA10

MICROPROCESSORS AND MICROCONTROLLERS

LAB MANUAL

Vision of the Department	To facilitate academic excellence and research attributes among Electronics and Communication Engineers adapting to the latest technologies that meet the global standards with high competence and professionalism.
Mission of the Department	M1: To provide a curriculum that practice quality teaching-learning process for academics and research enhancement. M2: To produce a qualified Engineer with sound technical knowledge and skillto compete the changing industry requirements. M3: To create an environment that helps to develop professional and ethical values for the growth of the society.
Program Educational Objectives (PEOs)	
PSO1	To implement the concepts of Electronics and Communication in designing the solutions for the applications in the field such as Embedded Systems, Robotics, Sensors, Wireless communication and Nano- technology.
PSO2	To apply modern design tools to develop software and hardware solutions occurringin the field of Electronics and Communication.
PSO3	To address the societal challenges in the field of Communication Engineering that supports in achieving a successful career.

CO1	Acquire the knowledge on the fundamentals, architecture, features and basic instructions of 8085 Microprocessor	K02	PO1
CO2	Apply knowledge and demonstrate programming proficiency using the various addressing modes and instructions set of 8085 Microprocessor	K03	PO2
CO3	Understand the architecture, instructions of 8086 Microprocessor and apply assembly language code to solve arithmetic, logic and string problems	K03	PO3
CO4	Understand the architecture of 8051 Microcontroller and Interfacing of Peripheral ICs	K02	PO6
CO5	Understand the fundamentals of RISC and ARM processors architecture and register organization	K02	PO12
CO6	Analyse and compare the features of microprocessors and microcontrollers	K02	PO1
CO7	Apply knowledge to perform arithmetic, logical operations, Interfacing of Peripheral IC's, transferring of data and string operations of 8085, 8086 & 8051 microprocessors & microcontroller	K01,K02	PO9
CO8	Ability to express theoretical and practical knowledge with industrial experts	K02	PO10

EXPERIMENT LIST

S.No	EXPERIMENT NAME	**CO**	**PO**	**BL**
8085 Microprocessor				
1.	8-bit Addition using direct addressing with carry in 8085 Microprocessor	CO7	PO7	K01,K02
2.	8-bit Addition using indirect addressing with carry in 8085 Microprocessor	CO7	PO7	K01,K02
3.	8-bit Subtraction without borrow in 8085 Microprocessor	CO7	PO7	K01,K02
4.	8-bit Subtraction with borrow in 8085 Microprocessor	CO7	PO7	K01,K02
5.	8-bit Multiplication operation using 8085 Microprocessor	CO7	PO7	K01,K02
6.	8-bit Division operation using 8085 Microprocessor	CO7	PO7	K01,K02
7.	ASCII to HEX (Binary) Conversion using 8085 Microprocessor	CO7	PO7	K01,K02
8.	Ascending Order of an Unsorted Array using 8085 Microprocessor	CO7	PO7	K01,K02
9.	Descending Order of an Unsorted Array using 8085 Microprocessor	CO7	PO7	K01,K02
10.	Program transferring a block of data between memory locations using 8085 instructions	CO7	PO7	K01,K02
11.	Perform Masking and Setting of lower nibbles on given data using 8085 Microprocessor	CO7	PO7	K01,K02
12.	Find One’s and Two’s complement of given data using 8085 Microprocessor	CO7	PO7	K01,K02
8086 Microprocessor				
13.	16-bit Addition using indirect addressing with carry in 8086 Microprocessor	CO7	PO7	K01,K02
14.	16-bit Subtraction with borrow in 8086 Microprocessor	CO7	PO7	K01,K02
15.	16-bit Multiplication operation using 8086 Microprocessor	CO7	PO7	K01,K02
16.	16-bit Division operation using 8086 Microprocessor	CO7	PO7	K01,K02
17.	String Operation using 8086 Microprocessors (Move a block of data from source to destination)	CO7	PO7	K01,K02
18.	Logical operations in 8086 Microprocessor [a] Masking of bits using AND operation. [b] Setting of bits using OR operation.	CO7	PO7	K01,K02
19.	1’s and 2's complement of a 16-bit number using 8086 Microprocessor	CO7	PO7	K01,K02
20.	Sum of N Numbers in a word array using 8086 Microprocessor	CO7	PO7	K01,K02
Interfacing with 8085/8086				
21.	Interface Stepper Motor with 8085 Microprocessor to run in Clockwise and Anticlockwise direction	CO7	PO7	K01,K02
22.	Interface 8279 with 8086 for the following [a]. Character display using Keyboard Display Controller [b]. Rolling display using Keyboard Display Controller	CO7	PO7	K01,K02
23.	Interface switches with 8086 via 8255 Programmable Peripheral Interface	CO7	PO7	K01,K02
24.	Interface the Timer 8253 with 8085 Microprocessor to generate square wave	CO7	PO7	K01,K02
8051 Microcontroller				
25.	Addition operation using 8051 Microcontroller	CO7	PO7	K01,K02
26.	Subtraction operation using 8051 Microcontroller	CO7	PO7	K01,K02
27.	Multiplication operation using 8051 Microcontroller	CO7	PO7	K01,K02
28.	Division operation using 8051 Microcontroller	CO7	PO7	K01,K02
29.	Setting and Masking of bits using 8051 Microcontroller	CO7	PO7	K01,K02

INDEX

Arithmetic Operations in 8085 Microprocessors

1. 8-BIT ADDITION WITH CARRY USING DIRECT ADDRESSING

AIM:

To write an assembly language program to add tow numbers of 8-bit data stored in memory locations 4200H and 4201H and store the result in 4202H and 4203H with carry using direct addressing.

APPARATUS REQUIRED:

1. 8085 microprocessor kit -------------- 1
2. Power card ---- 1
3. Keyboard---- 1

ALGORITHM:

1. Load the first data from memory to accumulator and move it to B register.
2. Load the second data from memory to accumulator.
3. Add the content of B – register to accumulator
4. If Carry flag = 0 then jump to step 6
5. Increment C register to count the carry
6. Store the sum in memory.
7. Move the carry to accumulator and store in memory.
8. Stop.

PROGRAM TO ADD TWO 8-BIT DATA

Memory address	Label	Instruction	Opcode	Comments
4100		LDA 4200H		Get 1st data in A and save in B.
4103		MOV B, A		
4104		LDA 4201H		Get 2nd data in A-register
4107		ADD B		Get the sum in A register
4108		JNC SKIP		If CY=0 Then skip next step
410B		INR C		Increment C register to count the carry
410C	SKIP	STA 4202H		Store the sum in memory
410F		MOV A,C		Move the carry to accumulator and store in memory
4110		STA 4203H		
4113		HLT		Stop the Execution

Input		Output	
Address	**Data**	**Address**	**Data**
4200	**CF**	**4202**	**6C (Sum)**
4201	**9D**	**4203**	**01 (Carry)**

RESULT:

Thus, an assembly language program for addition of given two 8-bit number with carry was written, executed and Verified the Result successfully using 8085 kit.

2. 8-BIT ADDITION USING INDIRECT ADDRESSING

AIM:

To write an assembly language program to add tow numbers of 8-bit data stored in memory locations 4200H and 4201H and store the result in 4202H and 4203H using indirect addressing modes

APPARATUS REQUIRED:

1. 8085 microprocessor kit -------------- 1
2. Power card ---- 1
3. Keyboard---- 1

ALGORITHM:

1. Load the first data from memory to accumulator.
2. Add the content of Memory to accumulator
3. If Carry flag = 0 then jump to step 5
4. Increment C register to count the carry
5. Store the sum in memory.
6. Store the carry in memory from C-register.
7. Stop.

PROGRAM TO ADD TWO 8-BIT DATA

Memory address	Label	Instruction	Opcode	Comments
4100		LXI H,4200H		Load the HL with data address
4103		MVI C,00		Clear C-Reg to count carry
4105		MOV A, M		Get 1st data in A
4106		INX H		
4107		ADD M		Add reg-A with Memory and Get the sum in A register
4108		JNC SKIP		If CY=0, Skip next step
410B		INR C		Increment C-Reg
410C	SKIP	INX H		
410D		MOV M, A		Store the sum in memory
410E		INX H		
410F		MOV M, C		Store the Carry in memory
4110		HLT		Stop the Execution

SAMPLE DATA:

Input		Output	
Address	**Data**	**Address**	**Data**
4200	**2F**	**4202**	**18 (Sum)**
4201	**E9**	**4203**	**01 (Carry)**

RESULT:

Thus, an assembly language program for addition of given two 8-bit number with carry was written, executed and Verified the Result successfully using 8085 kit.

3. 8-BIT SUBTRACTION WITHOUT BORROW

AIM:

To write an assembly language program to subtract two numbers of 8-bit data stored in memory 4200H and 4201H. Store the magnitude of the result in 4202H using 8085 Microprocessor.

APPARATUS REQUIRED:

1. 8085 microprocessor kit -------------- 1
2. Power card ---- 1
3. Keyboard---- 1

ALGORITHM:

1. Load the subtrahend (the data to be subtracted) from memory to accumulator and move if to B-register.
2. Load the minuend from memory to accumulator.
3. Subtract the content of B-register (subtrahend) from the content of accumulator (minuend).
4. Store the difference in memory.
5. Stop.

PROGRAM TO SUBTRACT TWO 8-BIT DATA

Memory address	Label	Instruction	Opcode	Comments
4100		LDA 4201H		; Get the subtrahend in B register.
4103		MOV B,A		
4104		LDA 4200H		;Get the minuend in A register
4107		SUB B		; Get the difference in A register.
4108		STA 4202H		Store the result in memory
410B		HLT		Stop the Execution

Sample data

Address	**Input Data**	**Address**	**Output Data**
4200	**D5**	**4202**	**8B (Difference)**
4201	**4A**		

RESULT:

Thus, an assembly language program for subtraction of given two 8-bit number without borrow was written, executed and Verified the Result successfully using 8085 kit.

4. 8-BIT SUBTRACTION WITH BORROW USING DIRECT ADDRESSING

AIM:

To write an assembly language program to subtract tow numbers of 8-bit data stored in memory locations 4200H and 4201H and store the result in 4202H and 4203H with borrow using direct addressing.

APPARATUS REQUIRED:

1. 8085 microprocessor kit -------------- 1
2. Power card ---- 1
3. Keyboard---- 1

ALGORITHM:

1. Load the second data from memory to accumulator and move it to B register.
2. Load the first data from memory to accumulator.
3. Subtract the content of B – register from accumulator
4. If Carry flag = 0 then jump to step 5 & 6
5. Increment C register to count the borrow
6. Take two’s complement of the difference
7. Store the Difference in memory.
8. Move the borrow to accumulator and store in memory.
9. Stop.

PROGRAM TO SUBTRACT TWO 8-BIT DATA

Memory address	Label	Instruction	Opcode	Comments
4100		LDA 4201H		Get 2nd data in A and save in B.
4103		MOV B, A		
4104		LDA 4200H		Get 1st data in A-register
4107		SUB B		Subtract B-Reg from A register
4108		JNC SKIP		If CY=0 Then skip next two steps
410B		INR C		Increment C register to count the carry
410C		CMA		Take two’s complement of difference
410D		INR A		
410E	SKIP	STA 4202H		Store the Difference in memory
4111		MOV A,C		Move the Borrow to accumulator and store in memory
4112		STA 4203H		
4115		HLT		Stop the Execution

SAMPLE DATA:

Input		**Output**	
Address	**Data**	**Address**	**Data**
4200	**CF**	**4202**	**0E (Sum)**
4201	**DD**	**4203**	**01 (Borrow)**

RESULT:

Thus, an assembly language program for subtraction of given two 8-bit number with borrow was written, executed and Verified the Result successfully using 8085 kit.

5. 8-BIT MULTIPLICATION OPERATIONS USING 8085 MICROPROCESSOR

AIM:

To write an assembly language program to multiply two numbers of 8-bit data stored in memory 4200H and 4201H and store the product in 4202H and 4203H.

APPARATUS REQUIRED:

1. 8085 microprocessor kit -------------- 1
2. Power card ---- 1
3. Keyboard---- 1

ALGORITHM:

- ✸ Load the first data in ACC and move to E.
- ✸ Load the second data ACC and move to B (count)
- ✸ Clear HL pair (Initial sum)
- ✸ Clear D for overflow (carry)
- ✸ Add the content of DE to HL
- ✸ Decrement the count.
- ✸ Check whether count has reached zero.
- ✸ Check the zero flag. If ZF = 0, repeat addition or If ZF = 1, go to next step
- ✸ Store the content of HL in memory. (Least significant 16 bits of the product)
- ✸ Stop.

PROGRAM TO MULTIPLY TWO NUMBERS OF 8-BIT DATA

Memory address	Label	Instruction	Opcode	Comments
4100		LDA 4200H		;Get 1 st data in A
4103		MOV E, A		;Save 1st data in E
4104		LDA 4201H		;Get 2nd data in A
4107		MOV B, A		;save 2nd data in B
4108		LXI H,0000H		;Clear HL pair(initial sum=0)
410B		MVI D,00H		;Clear E for accounting overflow.
410D	NEXT:	DAD D		;Add the content of DE to sum(HL)
410E		DCR B		Decrement data 2 for every addition
410F		JNZ NEXT		;Repeat Addition until count is zero.
4112		SHLD 4202H		;Store the product in memory
4115		HLT		Stop the Execution

Sample data

Address	Input Data	Address	Output Data
4200	6D (Data-1)	4202	26 (Lower byte of product)
4201	FE (Data-2)	4203	6C (Higher byte of product)

RESULT:

Thus, an assembly language program to multiply two numbers of 8-bit data was written, executed and Verified the Result successfully using 8085 kit.

6. 8-BIT DIVISION OPERATIONS USING 8085 MICROPROCESSOR

AIM:

To write an ALP to perform division of two 8 bit numbers Stored in memory location 4200H, 4201H and Store the remainder in 4202H and the quotient in 4203H.

APPARATUS REQUIRED:

1. 8085 microprocessor kit ---------------------- 1
2. Power card 1
3. Keyboard ------------ 1

ALGORITHM:

- ✷ Load the divisor in accumulator and move if to B-register
- ✷ Load the dividend in accumulator.
- ✷ Clear C-register to account for quotient
- ✷ Check whether divisor is less than dividend
- ✷ If divisor is less than dividend, go to step 8, otherwise go to next step
- ✷ Subtract the content of B-register (quotient)
- ✷ Increment the content of C-register (quotient)
- ✷ Go to step 4
- ✷ Store the content of accumulator (remainder) in memory.
- ✷ Move the content of C-register (quotient) to accumulator and store in memory
- ✷ Stop.

PROGRAM TO DIVIDE TWO NUMBERS OF 8-BIT DATA

Memory address	Label	Instruction		Comments
4100		LDA	4201H	
4103		MOV	B,A	;Get the divisor in B register
4104		LDA	4200H	;Get the dividend in A register
4107		MVI	C,00H	;Clear C register for quotient
4109	AGAIN:	CMP	B	
410A		JC	STORE	;If divisor is less than dividend go to store
410D		SUB	B	;Subtract divisor from dividend. Increment
410E		INR	C	;quotient by one for each subtraction.
410F		JMP	AGAIN	
4112	STORE:	STA	4203H	;Store the remainder in memory
4115		MOV	A,C	
4116		STA	4202H	;Stare the quotient in memory
4119		HLT		Stop the Execution

Sample data

Address	Input Data	Address	Output Data
4200	9F (Dividend)	4202	0F (Quotient)
4201	0A (Divisor)	4203	09 (Remainder)

RESULT:

Thus, an assembly language program to Divide two numbers of 8-bit data was written, executed and Verified the Result successfully using 8085 kit.

7. SORT AN ARRAY OF DATA IN ASCENDING ORDER

AIM:

To write an assembly language program to sort an array of data in ascending order. The array is stored in memory starting form 4200H. The first element of the array gives the count value for the number of elements in the array.

APPARATUS REQUIRED:

1. 8085 microprocessor kit 1
2. Power card 1
3. Keyboard 1

ALGORITHM:

1. Load the count value from memory to A-register and save it in B-register
2. Decrement B-register (B is a count for (N-1) repetitions)
3. Set HL pair as data address pointer
4. Set C-register as counter for (N-1) comparisons.
5. Load a data of the array in accumulator using the data address pointer
6. Increment the HL pair (data address pointer)
7. Compare the data pointed by HL wait accumulator
8. If carry flag is set (If the content of accumulator is smaller than memory) then go to step 10, otherwise go to next step
9. Exchange the content of memory pointed by HL and the accumulator
10. Decrement C-register. If zero flag is reset go to step 6 otherwise go to next step
11. Decrement B-register. If zero flag is reset go the step 3 otherwise go to next step
12. Stop.

PROGRAM TO SORT AN ARRAY OF DATA IN ASCENDING ORDER

Memory address	Label	Instruction	Opcode	Comments
4100		LDA	4200H	;Load the count value
4103		MOV	B,A	;Set counter for (N-1) repetitions
4104		DCR	B	;of (N-1) comparisons
4105	LOOP 2	LXI	H,4200H	;Set pointer for array
4108		MOV	C,M	;Set count for (N-1) comparisons
4109		DCR	C	
410A		INX	H	;Increment pointer
410B	LOOP 1	MOV	A,M	;Get one data of array in A
410C		INX	H	
410D		CMP	M	;Compare next data with A register
410E		JC	AHEAD	;If content of A is less than memory then go to AHEAD
4111		MOV	D,M	;If the content of A is greater than
4112		MOV	M,A	;then content of memory

4113	DCX	H	;pointed by HL and previous location
4114	MOV	M,D	
4115	INX	H	
4116	AHEAD DCR	C	;Repeat comparisons until C count is zero
4117	JNZ	LOOP l	
411A	DCR	B	;Repeat until B count is zero
411B	JNZ	LOOP 2	
411E	HLT		Stop the Execution

Sample Data

Address	Data Array (Before sorting)	Address	Data Array (After sorting)
4200	07 (Count)	4200	07 (Count)
4201	AB (Data -1)	4201	34 (Data -1)
4202	92 (Data -2)	4202	4F (Data -2)
4203	84 (Data -3)	4203	69 (Data -3)
4204	4F (Data -4)	4204	84 (Data -4)
4205	69 (Data -5)	4205	92 (Data -5)
4206	F2 (Data -6)	4206	AB (Data -6)
4207	34 (Data -7)	4207	F2 (Data -7)

RESULT:

Thus, an assembly language program for sorting in Ascending order of an unsorted array of given 8-bit number was written, executed and Verified the Result successfully using 8085 kit.

8. SORT AN ARRAY OF DATA IN DESCENDING ORDER

AIM:

To write an assembly language program to sort the array of data in descending order. The array is stored in memory stored in memory starting from 4200H. The first element of the array gives the count value for the number of elements in the array.

APPARATUS REQUIRED:

1. 8085 microprocessor kit 1
2. Power card 1
3. Keyboard 1

ALGORITHM:

The algorithm is same as algorithm of example program 15 except step 8.

Step 8: If carry flag is reset (If content of accumulator is larger than memory) then go to step 10, otherwise go to next step

PROGRAM TO SORT AN ARRAY OF DATA IN DESCENDING ORDER

Memory address	Label	Instruction	Opcode	Comments
4100		LDA	4200H	;Load the count value
4103		MOV	B,A	;Set counter for (N-1) repetitions
4104		DCR	B	;of (N-1) comparisons
4105	LOOP 2	LXI	H,4200H	;Set pointer for array
4108		MOV	C,M	;Set count for (N-1) comparisons
4109		DCR	C	
410A		INX	H	;Increment pointer
410B	LOOP 1	MOV	A,M	;Get one data of array in A
410C		INX	H	
410D		CMP	M	;Compare next data with A register
410E		JNC	AHEAD	;If content of A is less than memory then go to AHEAD
4111		MOV	D,M	;If the content of A is greater than
4112		MOV	M,A	;then content of memory
4113		DCX	H	;pointed by HL and previous location
4114		MOV	M,D	
4115		INX	H	
4116	AHEAD	DCR	C	;Repeat comparisons until C count is zero
4117		JNZ	LOOP l	
411A		DCR	B	;Repeat until B count is zero
411B		JNZ	LOOP 2	
411E		HLT		Stop the Execution

SAMPLE DATA:

Address	Data Array (Before sorting)	Address	Data Array (After sorting)
4200	07 (Count)	4200	07 (Count)
4201	AB (Data -1)	4201	F2 (Data -7)
4202	92 (Data -2)	4202	AB (Data -6)
4203	84 (Data -3)	4203	92 (Data -5)
4204	4F (Data -4)	4204	84 (Data -4)
4205	69 (Data -5)	4205	69 (Data -3)
4206	F2 (Data -6)	4206	4F (Data -2)
4207	34 (Data -7)	4207	34 (Data -1)

RESULT:

Thus, an assembly language program for sorting in descending order of an unsorted array of given 8-bit number was written, executed and Verified the Result successfully using 8085 kit.

9. ASCII CODE TO HEX CODE CONVERSION

AIM:

To write an assembly language program to convert an array of ASCII codes to corresponding binary (Hex) value in 8085 Microprocessor

APPARATUS REQUIRED:

1. 8085 microprocessor kit -------------- 1
2. Power card ---- 1
3. Keyboard---- 1

ALGORITHM:

1. Get the ASCII data in A register from 4200H
2. Subtract 30H from A-register
3. Compare the content of A-register with 0AH
4. If CY = 1 go to step 6. If CY = 0, go to next step
5. Subtract 07H from A-register
6. Store the HEX into 4202H
7. Stop the program

PROGRAM TO CONVERT ASCII CODE TO BINARY VALUE

Memory address	Label	Instruction	OP Code	Comments
4100		LDA 4200		Get the ASCII date to A register
4103		SUI 30H		;Subtract 30h from the data
4105		CPI 0AH		Compare the result with 0A
4107		JC STORE		;If CY = 1, Store the result
410A		SUI 07H		;Else then subtract 07H
410C	STORE	STA 4201		Store the result
410F		HLT		Stop the program

SAMPLE DATA:

ASCII Input		Hex Output	
4200	41	4201	0A

RESULT:

Thus, an assembly language program for converting ASCII to HEX of given 8-bit number was written, executed and Verified the Result successfully using 8085 kit.

10. TRANSFERRING A BLOCK OF DATA BETWEEN MEMORY LOCATIONS

AIM:

To write an assembly language program to transfer a block of data from source to destination in 8085 Microprocessor

APPARATUS REQUIRED:

1. 8085 microprocessor kit -------------- 1
2. Power card ---- 1
3. Keyboard---- 1

ALGORITHM:

1. Initialize BC pair with 4200H
2. Initialize DE pair with 4300H
3. Initialize H Reg with number of bytes to be transferred
4. Load a Byte from source address and Store the same on Destination address
5. Check all the data bytes transferred if no repeat step 4
6. Stop the program

Memory address	Label	Instruction	Opcode	Comments
4100		LXI B,	4200H	Initialize BC pair with 4200H
4103		LXI D,	4300H	Initialize DE pair with 4300H
4106		MVI	H,07	Initialize H Reg with number of bytes to be transferred
4108	NEXT	LDAX B		Load a Byte from source address
4109		STAX D		Store the same on Destination address
410A		INX B		Repeat the above steps to transfer the entire string
410B		INX D		
410C		DCR H		
410D		JNZ NEXT		
4110		HLT		Stop the process

SAMPLE DATA:

Address	Source Array	Address	Destination Array
4200	11	4300.	11
4201	22	4301.	22
4202	33	4302.	33
4203	AA	4303.	AA
4204	BB	4304.	BB
4205	CC	4305.	CC
4206	DD	4306.	DD
4207	EE	4307.	EE

RESULT:

Thus, an assembly language program for Transferring of given 8-bit numbers was written, executed and

Verified the Result successfully using 8085 kit.

11.MASKING AND SETTING OF LOWER NIBBLES ON GIVEN DATA

AIM:

To write and execute an assembly language program for performing Masking, Setting, One's and Two's Complement of given data of 8-bit numbers using 8085 Microprocessor.

APPARATUS REQUIRED:

1. 8085 microprocessor kit -------------- 1
2. Power card ---- 1
3. Keyboard---- 1

MASKING OF BITS

ALGORITHM:

1. Load the Data in A-register.
2. Logically AND the content of A with 0FH.
3. Store the result in memory location.
4. Stop the program

PROGRAM:

i) By using 8086 kit:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
4100		LDA 4200		Load A-register with Data
4103		ANI, 0FH		AND the content of A with 0FH
4105		STA 4201		Store the Result
4108		HLT		Stop the program

OUTPUT:

INPUT		OUTPUT	
Address	Data	Address	Data
4200H	4A	4201H	0A

SETTING OF BITS

ALGORITHM:

1. Load the Data in A-register.
2. Logically ORI the content of A with 0FH..
3. Store the result in memory location.
4. Stop the program

PROGRAM:

ii) By using 8086 kit:

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4100			LDA 4200	Load A-register with Data
4103			ORI, 0FH	OR the content of A with 0FH

4105			STA 4201	Store the Result
4108	F4		HLT	Stop the program

OUTPUT:

INPUT		**OUTPUT**	
Address	**Data**	**Address**	**Data**
4200H	**C5**	**4201H**	**CF**

RESULT:

Thus, an assembly language program for performing logical Masking and Setting of bits were executed using 8085 kit.

12.ONE'S AND TWO'S COMPLEMENT

AIM:

To write and execute an assembly language program for performing One's and Two's Complement of given 8-bit numbers using 8085 Microprocessor.

APPARATUS:

1. 8085 microprocessor kit ------------- 1
2. Power card ---- 1
3. Keyboard---- 1

MASKING OF BITS

ALGORITHM:

1. Load the Data in A-register.
2. Logically NOT the content of A.
3. Store the One's complement in memory location.
4. Increment the content of A.
5. Store the Two's complement in memory location.
6. Stop the program

PROGRAM:

By using 8086 kit:

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4100			LDA 4200	Load AL-register with 1st Data
4103			CMA	NOT the content of AX
4104			STA 4201	Store the One's complement in memory location.
4107			INR A	Increment the content of AX.
4108			STA 4202	Store the Two's complement in memory location
410B	F4		HLT	Stop the program

OUTPUT:

INPUT		OUTPUT	
Address	**Data**	**Address**	**Data**
4200H	**AB**	**4201H**	**54**
		4202H	**55**

RESULT:

Thus, an assembly language program for performing One's and Two's Complement of bits were executed using 8085 kit.

13. ADDITION OF 16 BIT NUMBERS WITH CARRY

AIM:

To write and execute an assembly language program to add two 16-bit unsigned numbers with carry in 8086 kit.

APPARATUS:

1. 8086 microprocessor kit-------------- 1
2. Power card---- 1
3. Keyboard --- 1

ALGORITHM:

1. Load the First Data in AX-register.
2. Load the First Data in BX-register.
2. Add the two data and get the sum in AX-register.
3. If C=0 then skip next step.
4. Increment CX Reg for carry
5. Store the sum in memory locations.
6. Store the Carry in memory location.
7. Stop the program.

PROGRAMM

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENT
		MOV CX,0000H		Initialize counter CX
		MOV AX,[1200]		Get the first data in AX register.
		MOV BX,[1202]		Get the second data in BX register.
		ADD AX,BX		Add the contents of both the register AX & BX
		JNC L1		Check for carry
		INC CX		If carry exists, increment the CX
	LI	MOV [1206],CX		Store the carry
		MOV [1204],AX		Store the sum
		HLT		Stop the program

OUTPUT FOR ADDITION:

INPUT		OUTPUT	
Address	Data	Address	Data
1200	ABCD	1204	9ADF
1202	EF12	1206	0001

RESULT

Thus, an assembly language program for addition with carry of given 16-bit numbers was written, executed and Verified the Result successfully using 8086 kit.

14.SUBTRACTION OF 16 BIT NUMBERS WITH BORROW

AIM

To write and execute an assembly language program to subtract two 16-bit unsigned numbers with borrow in 8086 kit.

APPARATUS:

1. 8086 microprocessor kit-------------- 1
2. Power card---- 1
3. Keyboard --- 1

ALGORITHM:

1. Load the second data from memory to accumulator and move it to B register.
2. Load the first data from memory to accumulator.
3. Subtract the content of B – register from accumulator
4. If Carry flag = 0 then jump to step 5 & 6
5. Increment C register to count the borrow
6. Take two's complement of the difference
7. Store the Difference in memory.
8. Move the borrow to accumulator and store in memory.
9. Stop.

PROGRAMM

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENT
1100		MOV CX,0000H		Initialize counter CX
1104		MOV AX,[1300]		Get the first data in AX register
1108		MOV BX,[1302]		Get the second data in BX register.
110C		SUB AX,BX		Subtract the contents of both the register AX & BX
110E		JNC SKIP		Check the Borrow.
1110		INC CX		If carry exists, increment the CX
1111		NEG AX		Take two's complement of the difference
1113	SKIP	MOV [1306],CX		Store the Borrow.
1117		MOV [1304],AX		Store the difference.
111A		HLT		Stop the program

OUTPUT FOR SUBTRACTION:

INPUT		OUTPUT	
Address	**Data**	**Address**	**Data**
1200	**ABCD**	**1204**	**4345**
1202	**EF12**	**1206**	**0001**

RESULT

Thus, an assembly language program for subtraction with borrow of given 16-bit numbers was written, executed and Verified the Result successfully using 8086 kit.

15.MULTIPLICATION OF 16 BIT NUMBERS

AIM

To write and execute an assembly language program to Multiply two 16-bit unsigned numbers in 8086 kit.

APPARATUS:

1. 8086 microprocessor kit-------------- 1
2. Power card---- 1
3. Keyboard --- 1

ALGORITHM:

1. Load the multiplier from memory to accumulator.
2. Load the Multiplicand from memory to BX Reg .
3. Multiply AX with BX.
4. Store the Lower word in memory from AX.
5. Store the Higher word in memory from DX.
6. Stop.

PROGRAMM

ADDRESS	LABEL	MNEMONIC	OPCODE	COMMENTS
		MOV AX, [1200]		Load AX-register with 1st data
		MOV BX,[1202]		Load BX-register with 2nd data
		MUL BX		Multiply the contents of AX with BX-register
		MOV [1204],AX		Store the Lower word
		MOV [1206],DX		Store the Higher word
		HLT		Stop the program

OUTPUT:

INPUT		OUTPUT	
Address	**Data**	**Address**	**Data**
1200	**ABCD**	**1204**	**776A**
1202	**EF12**	**1206**	**A070**

RESULT

Thus, an assembly language program for multiplication of given 16-bit numbers was written, executed and Verified the Result successfully using 8086 kit.

16.MULTIPLICATION OF 16 BIT NUMBERS

AIM

To write and execute an assembly language program to Divide two 16-bit unsigned numbers in 8086 kit.

APPARATUS:

4. 8086 microprocessor kit-------------- 1
5. Power card---- 1
6. Keyboard --- 1

ALGORITHM:

1. Load the Divisor from memory to accumulator.
2. Load the Divisor from memory to BX Reg .
3. Divide DXAX by BX.
4. Store the Quotient in memory from AX.
5. Store the Reminder in memory from DX.
6. Stop.

PROGRAMM

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
		MOV DX,0000		Initialize DX-register with Lsb of Dividend
		MOV AX,[1200]		Load AX-register with Dividend
		MOV BX, [1202]		Load BX-register with Divisor
		DIV CX		Divide AX by BX-register
		MOV [1204],AX		Store the Quotient
		MOV [1206],DX		Store the Remainder
		HLT		Stop the program

OUTPUT:

INPUT		OUTPUT	
Address	Data	Address	Data
1200	**EF12**	**1204**	**0001**
1202	**ABCD**	**1206**	**4345**

RESULT

Thus, an assembly language program for Division of given 16-bit numbers was written, executed and Verified the Result successfully using 8086 kit.

17.LOGICAL OPERATION

AIM:

To write and execute an assembly language program for performing Masking, Setting, One’s and Two’s Complement of given 16-bit numbers using 8086 Microprocessor.

APPARATUS:

4. 8086 microprocessor kit -------------- 1
5. Power card ---- 1
6. Keyboard---- 1

MASKING OF BITS

ALGORITHM:

5. Load the Data in AX-register.
6. Logically AND the content of AX with 0F0FH.
7. Store the result in memory location.
8. Stop the program

PROGRAM:

iii) By using 8086 kit:

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
1100			MOV AX,[1200]	Load AL-register with 1st Data
1104			AND AX, 0F0FH	AND the content of AX with 0F0FH
1108			MOV [1202],AX	Store the Result
110C	F4		HLT	Stop the program

OUTPUT:

INPUT		OUTPUT	
Address	**Data**	**Address**	**Data**
1200H	**ABCD**	**1202H**	**0B0D**

SETTING OF BITS

ALGORITHM:

9. Load the Data in AX-register.
10. Logically OR the content of AX with 0F0FH.
11. Store the result in memory location.
12. Stop the program

PROGRAM:

iv) By using 8086 kit:

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
1100			MOV AX,[1200]	Load AL-register with 1st Data
1104			OR AX, 0F0FH	AND the content of AX with 0F0FH

1108			MOV [1202],AX	Store the Result
110C	F4		HLT	Stop the program

OUTPUT:

INPUT		OUTPUT	
Address	**Data**	**Address**	**Data**
1200H	**ABCD**	**1202H**	**FBFD**

RESULT:

Thus, an assembly language program for performing logical Masking and Setting of bits were executed using 8086 kit.

18.ONE’S AND TWO’S COMPLEMENT

AIM:

To write and execute an assembly language program for performing One’s and Two’s Complement of given 16-bit numbers using 8086 Microprocessor.

APPARATUS:

1. 8086 microprocessor kit -------------- 1
2. Power card ---- 1
3. Keyboard---- 1

MASKING OF BITS

ALGORITHM:

7. Load the Data in AX-register.
8. Logically NOT the content of AX.
9. Store the One’s complement in memory location.
10. Increment the content of AX.
11. Store the Two’s complement in memory location.
12. Stop the program

PROGRAM:

v) By using 8086 kit:

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
1100			MOV AX,[1200]	Load AL-register with 1st Data
1104			NOT AX	NOT the content of AX
1108			MOV [1202],AX	Store the One’s complement in memory location.
110C			INC AX	Increment the content of AX.
110D			MOV [1204],AX	Store the Two’s complement in memory location
1110	F4		HLT	Stop the program

OUTPUT:

INPUT		OUTPUT	
Address	**Data**	**Address**	**Data**
1200H	**12AB**	**1202H**	**ED54**
		1204H	**ED55**

RESULT:

Thus, an assembly language program for performing One’s and Two’s Complement of bits were executed using 8086 kit.

19. MOVE A DATA BLOCK WITHOUT OVERLAP

AIM:

To write and execute an assembly language program for transferring data from one block to another block without overlapping using 8086 kit and MASM.

APPARATUS:

1. 8086 microprocessor kit ------------- 1
2. Power card ---- 1
3. Keyboard---- 1
4.

ALGORITHM:

1. Initialize counter.
2. Initialize source block pointer.
3. Initialize destination block pointer.
4. Get the byte from source block.
5. Store the byte in destination block.
6. Increment source, destination pointers and decrement counter.
7. Repeat steps 4, 5 and 6 until counter equal to zero.
8. Stop.

PROGRAM:

i) By using 8086 kit:

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
	C7 C6 0012		MOV SI, 1150H	Initialize the source address.
	C7 C7 0013		MOV DI,1250H	Initialize the destination address.
	C7 C1 0600		MOV CX,0006 H	Initialize count value to the count register.
	FC	REPEAT:	CLD	Clear the direction flag.
	A4		MOVSB	Move the string byte.
	E2,F3		LOOP REPEAT	Unconditional loop to address specified by the label REPEAT.
	F4		HLT	Stop the program

OUTPUT:

INPUT		OUTPUT	
Address	**Data**	**Address**	**Data**
1150.	**52.**	**1250.**	**52.**
1151.	**53.**	**1251.**	**53.**
1152.	**54.**	**1252.**	**54.**
1153.	**55.**	**1253.**	**55.**
1154.	**56.**	**1254.**	**56.**
1155.	**57.**	**1255.**	**57.**

VIVA QUESTIONS:

1. What is the fabrication technology used for 8086?
2. What are the functional units available in 8086 architecture?
3. Write the flags of 8086.
4. What are control bits?
5. What are the flag manipulation instructions of 8086?
6. What is Macro?
7. Which bus controller used in maximum mode of 8086?
8. What is the size of data bus and address bus in 8086?
9. What are the various segment registers in 8086?
10. What is the maximum memory addressing capability of 8086?

RESULT:

Thus, an assembly language program for transferring data from one block to another block without overlapping was executed using 8086 kit.

20.SUM OF N NUMBERS IN A WORD ARRAY

AIM:

To write and execute an assembly language program for adding N Numbers in a word array using 8086 kit.

APPARATUS:

1. 8086 microprocessor kit -------------- 1
2. Power card ---- 1
3. Keyboard---- 1

ALGORITHM:

1. Initialize counter.
2. Initialize source block pointer.
3. Initialize destination block pointer.
4. Get the byte from source block.
5. Store the byte in destination block.
6. Increment source, destination pointers and decrement counter.
7. Repeat steps 4, 5 and 6 until counter equal to zero.
8. Stop.

PROGRAM:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1100		MOV DX,0H		Clear DX
1104		MOV SI,1250H		
1108		MOV CX,03H		
110C		MOV AX, [SI]		
110E	A1:	INC SI		
110F		INC SI		
1110		ADD AX, [SI]		
1112		JNC NEXT		
1114		INC DX		
1115	NEXT:	LOOP A1		
1117		MOV [1300H], AX		
111B		MOV [1302H], DX		
111F		HLT		

OUTPUT:

INPUT		OUTPUT	
Address	**Data**	**Address**	**Data**
1250	**ABCD**	**1300**	**0B64**
1252	**EF98**	**1302**	**0003**
1254	**DCBA**		
1256	**9345**		

VIVA QUESTIONS:

1. What is the fabrication technology used for 8086?
2. What are the functional units available in 8086 architecture?
3. Write the flags of 8086.
4. What are control bits?
5. What are the flag manipulation instructions of 8086?
6. What is Macro?
7. Which bus controller used in maximum mode of 8086?
8. What is the size of data bus and address bus in 8086?
9. What are the various segment registers in 8086?
10. What is the maximum memory addressing capability of 8086?

RESULT:

Thus, an assembly language program for transferring data from one block to another block without overlapping was executed using 8086 kit .

11. STEPPER MOTOR INTERFACING

AIM:

To write and execute an assembly language Program to run a stepper motor at different speed, and to control its direction using 8085 Microprocessor

APPARATUS:

1. 8085 microprocessor kit --------------1
2. Stepper Motor ----1
3. Stepper Motor Interface board----1
4. Power card --- 1
5. Keyboard----1

PROGRAM:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
4100	START	LXI H, 4200		Initialize HL with 4200H
4103		MVI C, 04		Copy the value 04 to C- register
4105	NEXT	MOV A, M		Copy the content M to A-register
4106		OUT C0		The content of A is moved to Out port
4108		LXI D, 1010		Copy the data 1010 to DE-reg Pair
410B	loop	DCX D		Decrement DE-register
410C		MOV A,E		
410D		ORA D		Check DE = 0000
410E		JNZ loop		Jump on no zero to loop
4111		INX H		Increment HL -register Pair
4112		DCR C		Decrement the count
4113		JNZ NEXT		Jump to NEXT if Z flag is zero
4115		JMP START		Jump to label START
4118		HLT		Stop the program.
4200	TABLE	09 05 06 0A		**clockwise direction**
4200	TABLE	0A 06 05 09		**Counter clockwise direction**

OUTPUT

INPUT		OUTPUT
Address	**Data**	**Motor Rotates on**
4200	09 05 06 0A	clockwise direction
4200	0A 06 05 09	Counter clockwise direction

RESULT:

Thus an assembly language Program to run the stepper motor in both forward and reverse direction with delay was executed and its output was verified.

12. KEYBOARD AND DISPLAY

AIM:

To write and execute an assembly language Program to display a character “7” and the rolling message “HELP US” in the display.

APPARATUS:

1. 8086 microprocessor kit -------------1
2. 8279 Interface board---- 1
4. Power card --- 1
5. Keyboard ----1

ROLLING MESSAGE “HELP US”

ALGORITHM:

Display of rolling message “HELP US “

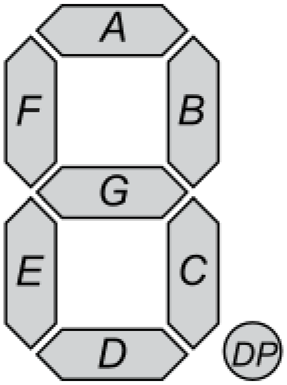
1. Initialize the counter
2. Set 8279 for 8 digit character display, right entry
3. Set 8279 for clearing the display
4. Write the command to display
5. Load the character into accumulator and display it
6. Introduce the delay
7. Repeat from step 1.

PROGRAM:

ADDRESS	LABEL	PROGRAM	OPCODE	COMMENTS
1100	START	MOV SI,1200H		Initialize array
1104		MOV CX,000FH		Initialize array size
1108		MOV AL,10		Store the control word for display mode
110B		OUT C2,AL		Send through output port
110D		MOV AL,0CC		Store the control word to clear display
1110		OUT C2,AL		Send through output port
1112		MOV AL,90		Store the control word to write display
1115		OUT C2,AL		Send through output port
1117	NEXT	MOV AL,[SI]		Get the first data
1119		OUT C0,AL		Send through output port
111B	DELAY	MOV DX,0FFFFH		Store 16bit count value
111F	LOOP1	DEC DX		Decrement count value
1120		JNZ LOOP1		Loop until count values becomes zero
1122		INC SI		Go & get next data
1123		LOOP NEXT		Loop until all the data has been taken
1125		JMP START		Go to starting location
1127		HLT		

LOOK-UP TABLE:

1200	98	68	7C	C8
1204	FF	1C	29	FF



OUTPUT:

ON – 0 OFF - 1

MEMORY LOCATION	Message	7-SEGMENT LED FORMAT								HEX CODE
		D	**C**	**B**	**A**	**DP**	**G**	**F**	**E**	
1200H	**H**	1	0	0	1	1	0	0	0	**98**
1201H	**E**	0	1	1	0	1	0	0	0	**68**
1202H	**L**	0	1	1	1	1	1	0	0	**7C**
1203H	**P**	1	1	0	0	1	0	0	0	**C8**
1204H		1	1	1	1	1	1	1	1	**FF**
1205H	**U**	0	0	0	0	1	1	0	0	**1C**
1206H	**S**	0	0	1	0	1	0	0	1	**29**
1207H		1	1	1	1	1	1	1	1	**FF**

DISPLAY THE CHARACTER “3”

ALGORITHM:

1. Set 8279 for 8-digit character display, right entry
2. Set 8279 for clearing the display
3. Write the command to display
4. Load the character into accumulator and display it
5. Repeat from step 1.

PROGRAM:

ADDRESS	LABEL	PROGRAM	OPCODE	COMMENTS
1100		MOV AL,00	C6 C0 00	Store the control word for display mode
1103		OUT C2,AL	E6 C2	Send through output port
1105		MOV AL,0CC	C6 C0 CC	Store the control word to clear display
1108		OUT C2,AL	E6 C2	Send through output port
110A		MOV AL,90	C6 C0 90	Store the control word to write display

110D		OUT C2,AL	E6 C2	Send through output port
110F		MOV AL,8F	C6 C0 8F	Get the first data
1112		OUT C0,AL	E6 C0	Send through output port
1114		HLT	F4	Stop the program

INPUT		**OUTPUT**	
Address	**Data**	**Address**	**Data**
1111	**0B**	**Display**	**3**

VIVA QUESTIONS:

1. What are the types of interfacing?
2. Compare memory interfacing and IO interfacing.
3. What is the difference between memory mapped IO and IO mapped IO interfacing?
4. What IC 8279 is?
5. What are the tasks involved in keyboard interface?
6. What is scanning in keyboard and what is scan time?
7. What is the difference between 2-key and n-key rollover?
8. What is the control registers available in 8279?
9. What is key debouncing?
10. What are the command words available in 8279?

RESULT:

Thus, the rolling message “HELP US” and the character “3” are displayed using 8279 interface kit with 8086 Microprocessor

13. INTERFACE SWITCHES WITH 8086 THROUGH 8255

AIM:

To write and execute an assembly language Program to Interface (Port A) 8 switches with 8086 Microprocessor through 8255 PPI.

APPARATUS:

1. 8086 microprocessor kit --------------1
2. 8255 Interface board---- 1
3. Power card --- 1
4. Keyboard ----1

ALGORITHM:

1. Configure the 8255 port A as input port with the control reg value as “90H”
2. Read the port A switch status through C0.
3. Store the output in 1250.
4. Stop

PROGRAM:

ADDRESS	LABEL	PROGRAM	OPCODE	COMMENTS
1100		MOV AL,90		Load the AL with control word
1103		OUT C2,AL		Send the control word to control reg of 8255
1105		IN AL,C0		Read port A
1108		MOV [1250],AL		Store the result on memory
1114		HLT	F4	Stop the program

INPUT	OUT PUT	
VARY THE SWITCH POSITIONS ON OFF ON ON OFF ON OFF ON	**Address**	**Data**
	1250	**B5**

RESULT

Thus, an assembly language program for Interfacing of switches with 8086 through 8255 PPI was written, executed and Verified the Result successfully.

14. TIMER INTERFACING

AIM:

To write and execute an assembly language program for timer interfacing to generate a square wave using 8085b Microprocessor.

APPARATUS REQUIRED:

1. 8085 microprocessor kit -------------1
2. 8253/54 Interface board 1
3. Power card --- 1
4. Keyboard ----1

ALGORITHM:

(i) Send the control word.
(ii) Introduce suitable delay.
(iii) Send high value to DAC.
(iv) Introduce delay.
(v) Repeat the above procedure.

PROGRAM: SQUARE WAVE

ADDRESS	LABEL	PROGRAM	OPCODE	COMMENTS
4100		MVI A,36		
4102		OUT CE		
4104		MVI A,0A		
4106		OUT C8		
4108		MVI A,00		
410A		OUT C8		
410C		HLT		

OUTPUT:

WAVEFORM GENERATION

WAVEFORMS	AMPLITUDE	TIME PERIOD
Square wave		

RESULT:

Thus, an assembly language program for Interfacing of Timer 8253/54 with 8085 was written, executed and Verified the Result successfully.

15. ADDITION OPERATION USING 8051 MICROCONTROLLER

AIM:

To write and execute an assembly language program to Add of two 8-bit numbers using 8051.

APPARATUS:

1. 8051 microcontroller kit---- 1
2. Power card---- 1
3. Keyboard --- 1

ALGORITHM:

1. Load the First Data in A-register.
2. Load the Second Data in B-register.
3. Add the two data with carry.
4. Store the sum in memory location.
5. Stop the program.

PROGRAM:

ADDITION

ADDRESS	OPCODE	LABEL	PROGRAM	COMMENTS
4100	74,05		MOV A,#data	Load data 1 in accumulator.
4102	75,F0,05		MOV B,#data	Load data 2 in B-register
4105	35,F0		ADDC A,B	Add the contents of accumulator and B-reg with carry.
4107	90,11,00		MOV DPTR,#4500$_H$	Initialize DPTR with address 4500$_H$
410A	F0		MOVX @ DPTR,A	Store the Sum in 4500$_H$
410B	80, FE	STOP:	SJMP STOP	Stop the program

OUTPUT:

INPUT		OUTPUT	
Register	**Data**	**Address**	**Data**
4101		**4500**	
4104			

RESULT:

Thus, an assembly language program for addition of given two 8-bit number was written, executed and Verified the Result successfully using 8051 kit

16. SUBTRACTION OPERATION USING 8051 MICROCONTROLLER

AIM:

To write and execute an assembly language program to subtract two 8-bit numbers using 8051.

APPARATUS:

1. 8051 microcontroller kit---- 1
2. Power card---- 1
3. Keyboard --- 1

SUBTRACTION

ALGORITHM:

1. Load the First Data in A-register.
2. Load the Second Data in B-register.
3. Subtract the two data with borrow.
4. Store the sum in memory location.
5. Stop the program.

PROGRAM:

ADDRESS	OPCODE	LABEL	PROGRAM	COMMENTS
4100	74,05		MOV A,#data	Load data 1 in accumulator.
4102	75,F0,04		MOV B,#data	Load data 2 in B-register
4105	95,F0		SUBB A,B	Subtract the contents of B-reg from accumulator with borrow.
4107	90 11 00		MOV DPTR,#4500$_H$	Initialize DPTR with address 4500$_H$
410A	F0		MOVX @ DPTR,A	Store the difference in 4500$_H$
410B	80, FE	STOP:	SJMP STOP	Stop the program

OUTPUT:

INPUT		OUTPUT	
Register	**Data**	**Address**	**Data**
4101		**4500**	
4104			

RESULT:

Thus, an assembly language program for subtraction of given two 8-bit number was written, executed and Verified the Result successfully using 8051 kit

17. MULTIPLICATION OPERATION USING 8051 MICROCONTROLLER

AIM:

To write and execute an assembly language program to multiply two 8-bit numbers using 8051.

APPARATUS:

1. 8051 microcontroller kit---- 1
2. Power card---- 1
3. Keyboard --- 1

MULTIPLICATION

ALGORITHM:

1. Get the multiplier in the accumulator.
2. Get the multiplicand in the B register.
3. Multiply A with B.
4. Store the product in memory locations.
5. Stop the program.

PROGRAM:

ADDRESS	OPCODE	LABEL	PROGRAM	COMMENTS
4100	74,05		MOV A,#data	Load data 1 in accumulator.
4102	75,F0,05		MOV B,#data	Load data 2 in B-register
4105	A4		MUL AB	A*B, Higher byte of result in B and lower byte of result in A.
4106	90,11,00		MOV DPTR,#4500$_H$	Initialize DPTR with address 1100$_H$
4109	F0		MOVX @ DPTR,A	Store the LSB in 4500$_H$
410A	A3		INC DPTR	Increment Data pointer
410B	E5,F0		MOV A,B	Copy the content of B-reg to A-register.
410D	F0		MOVX @ DPTR,A	Store the MSB in 4501$_H$
410E	80, FE	STOP:	SJMP STOP	Stop the program

OUTPUT:

INPUT		OUTPUT	
REGISTER	**DATA**	**ADDRESS**	**DATA**
4101		**4500**	
4104		**4501**	

RESULT:

Thus, an assembly language program for multiplication of given two 8-bit number was written, executed and Verified the Result successfully using 8051 kit

18. DIVISION OPERATION USING 8051 MICROCONTROLLER

AIM:

To write and execute an assembly language program to divide two 8-bit numbers using 8051.

APPARATUS:

1. 8051 microcontroller kit---- 1
2. Power card---- 1
3. Keyboard --- 1

DIVISION

ALGORITHM:

1. Get the Dividend in the accumulator.
2. Get the Divisor in the B register.
3. Divide A by B.
4. Store the Quotient and Remainder in memory.
5. Stop the program.

PROGRAM:

ADDRESS	OPCODE	LABEL	PROGRAM	COMMENTS
4100	74,data1		MOV A,#CF	Load data 1 in accumulator.
4102	75,data2		MOV B,#21	Load data 2 in B-register
4104	84		DIV AB	Divide. Remainder in A and quotient in B
4105	90,11,00		MOV DPTR,#4500$_H$	Initialize DPTR with address 1100$_H$
4108	F0		MOVX @ DPTR,A	Store the quotient in 4500$_H$
4109	A3		INC DPTR	Increment Data pointer
410A	E5,F0		MOV A,B	Copy the content of B-reg to A-register.
410C	F0		MOVX @ DPTR,A	Store the Remainder in 4501$_H$
410D	80, FE	STOP:	SJMP STOP	Stop the program

OUTPUT:

INPUT		OUTPUT	
REGISTER	**DATA**	**ADDRESS**	**DATA**
4101		**4500**	**(quotient)**
4104		**4501**	**(remainder)**

RESULT:

Thus, an assembly language program for Division of given two 8-bit number was written, executed and Verified the Result successfully using 8051 kit

19. LOGICAL OPERATIONS USING 8051

AIM:

To write and execute an assembly language program for Setting and Masking of given 8-bit number using 8051.

APPARATUS REQUIRED:

1. 8051 microcontroller kit---- 1
2. Power card---- 1
3. Keyboard --- 1

SETTING OF BITS

ALGORITHM:

1. Load the Data in A-register.
2. Load 0F to set the lower nibble in B-register.
3. Perform OR operation with B-register.
4. Store the Result in memory location.
5. Stop the program.

PROGRAM:

ADDRESS	OPCODE	LABEL	PROGRAM	COMMENTS
4100	74,05		MOV A,#C3	Load data 1 in accumulator.
4102	75,F0,05		MOV B,#0F	Load data 2 in B-register
4105	35,F0		ORL A,B	OR the contents of accumulator and B-reg.
4107	90,11,00		MOV DPTR,#4500$_H$	Initialize DPTR with address 4500$_H$
410A	F0		MOVX @ DPTR,A	Store the Result in 4500$_H$
410B	80, FE	STOP:	SJMP STOP	Stop the program

OUTPUT:

INPUT		OUTPUT	
Register	**Data**	**Address**	**Data**
4101	**C3**	**4500**	**CF**

MASKING OF BITS

ALGORITHM:

1. Load the Data in A-register.
2. Load 0F to mask the higher nibble in B-register.
3. Perform AND operation with B-register.
4. Store the Result in memory location.
5. Stop the program.

PROGRAM:

ADDRESS	OPCODE	LABEL	PROGRAM	COMMENTS
4100	74,05		MOV A,#4D	Load data 1 in accumulator.
4102	75,F0,05		MOV B,#0F	Load data 2 in B-register
4105	35,F0		ANL A,B	AND the contents of accumulator and B-reg.
4107	90,11,00		MOV DPTR,#4500_H	Initialize DPTR with address 4500_H
410A	F0		MOVX @ DPTR,A	Store the Result in 4500_H
410B	80, FE	STOP:	SJMP STOP	Stop the program

OUTPUT:

INPUT		OUTPUT	
Register	**Data**	**Address**	**Data**
4101	**4D**	**4500**	**0D**

RESULT:

Thus, an assembly language program for Setting and Masking of 8-bit numbers using 8051 were performed and its outputs were verified.