INDEX

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
4. 8085 Simulator and a PC

ALGORITHM:

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

PROGRAM TO ADD TWO 8-BIT DATA

Memory address	Label	Instruction	Opcode	Comments
		MVI C,00		**Clear C reg to count carry**
		LDA 4200H		Get 1st data in A and save in B.
		MOV B, A		
		LDA 4201H		Get 2nd data in A-register
		ADD B		Get the sum in A register
		JNC SKIP		If CY=0 Then skip next step
		INR C		Increment C register to count the carry
	SKIP	STA 4202H		Store the sum in memory
		MOV A,C		Move the carry to accumulator and store in memory
		STA 4203H		
		HLT		Stop the Execution

Input		Output	
Address	**Data**	**Address**	**Data**
4200		**4202**	**(Sum)**
4201		**4203**	**(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. 16-BIT ADDITION WITH CARRY

AIM:

To write an assembly language program to add two numbers of 16-bit data stored in memory locations 4200H, 4201H, 4202H & 4203H and store the results in 4204H, 4205H & 4206H with carry.

APPARATUS REQUIRED

1. 8085 microprocessor kit ---------- 1
2. Power card ---- 1
3. Keyboard---- 1
4. 8085 Simulator and a PC

ALGORITHM:

1. Clear B reg to count carry
2. Load the first data from memory to HL pair and move it to DE pair.
3. Load the second data from memory to HL pair.
4. Add the content of BC Pair to HL pair
5. If Carry flag = 0 then jump to step 6
6. Increment D register to count the carry
7. Store the sum in memory.
8. Move the carry to accumulator and store in memory.
9. Stop.

PROGRAM TO ADD TWO 8-BIT DATA

Memory address	Label	Instruction	Opcode	Comments
		MVI B,00		**Clear B reg to count carry**
		LHLD 4200H		Get 1st data in HL pair and save in DE.
		XCHG		
		LHLD 4202H		Get 2nd data in HL-register
		DAD D		ADD HL with DE and Get the sum in HL register pair
		JNC SKIP		If CY=0 Then skip next step

		INR B		Increment C register to count the carry
	SKIP	SHLD 4204H		Store the sum in memory
		MOV A, B		Move the carry to accumulator and store in memory
		STA 4206H		
		HLT		Stop the Execution

Input		Output	
Address	**Data**	**Address**	**Data**
4200		**4204**	**(Sum)**
4202		**4206**	**(Carry)**

RESULT:

Thus, an assembly language program for addition of given two 16-bit number with carry was written, executed and Verified the Result successfully using 8085 kit and Simulator.

3. 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
4. 8085 Simulator and a PC

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
		LDA 4201H		Get 2nd data in A and save in B.
		MOV B, A		
		LDA 4200H		Get 1st data in A-register
		SUB B		Subtract B-Reg from A register
		JNC SKIP		If CY=0 Then skip next two steps
		INR C		Increment C register to count the carry
		CMA		Take two's complement of difference
		INR A		
	SKIP	STA 4202H		Store the Difference in memory
		MOV A,C		Move the Borrow to accumulator and store in memory
		STA 4203H		
		HLT		Stop the Execution

SAMPLE DATA:

Input		Output	
Address	**Data**	**Address**	**Data**
4200		**4202**	**(Sum)**
4201		**4203**	**(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.

4. 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

4. 8085 Simulator and a PC

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

SAMPLE DATA

Address	Input Data	Address	Output Data
4200	(Data-1)	4202	(Lower byte of product)
4201	(Data-2)	4203	(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.

5. 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
4. 8085 Simulator and a PC

ALGORITHM:

1. Load the divisor in accumulator and move if to B-register
2. Load the dividend in accumulator.
3. Clear C-register to account for quotient
4. Check whether divisor is less than dividend
5. If divisor is less than dividend, go to step 9, otherwise go to next step
6. Subtract the content of B-register (quotient)
7. Increment the content of C-register (quotient)
8. Go to step 4
9. Store the content of accumulator (remainder) in memory.
10. Move the content of C-register (quotient) to accumulator and store in memory
11. Stop.

PROGRAM TO DIVIDE TWO NUMBERS OF 8-BIT DATA

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

Sample data

Address	Input Data	Address	Output Data
4200	(Dividend)	4202	(Quotient)
4201	(Divisor)	4203	(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.

6. 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
4. 8085 Simulator and a PC

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
		LDA	4200H	;Load the count value
		MOV	B,A	;Set counter for (N-1) repetitions
		DCR	B	;of (N-1) comparisons
	LOOP 2	LXI	H,4200H	;Set pointer for array

		MOV	C,M	;Set count for (N-1) comparisons
		DCR	C	
		INX	H	;Increment pointer
	LOOP 1	MOV	A,M	;Get one data of array in A
		INX	H	
		CMP	M	;Compare next data with A register
		JC	AHEAD	;If content of A is less than memory then go to AHEAD
		MOV	D,M	;If the content of A is greater than
		MOV	M,A	;then content of memory
		DCX	H	;pointed by HL and previous location
		MOV	M,D	
		INX	H	
	AHEAD	DCR	C	;Repeat comparisons until C count is zero
		JNZ	LOOP I	
		DCR	B	;Repeat until B count is zero
		JNZ	LOOP 2	
		HLT		Stop the Execution

Sample Data

Address	Data Array (Before sorting)	Address	Data Array (After sorting)
4200	(Count)	4200	(Count)
4201		4201	
4202		4202	
4203		4203	
4204		4204	
4205		4205	
4206		4206	
4207		4207	

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.

7. 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
4. 8085 Simulator and a PC

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 I	
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	(Count)	4200	(Count)
4201		4201	
4202		4202	
4203		4203	
4204		4204	
4205		4205	
4206		4206	
4207		4207	

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.

8. SEARCH THE SMALLEST NUMBER FROM AN ARRAY

AIM:

Write an assembly language program to search the smallest data in an array of N data stored in memory from 4200H to (4200H + N). The first element of the array gives the number of data in the array. Store the smallest data in 4300H.

APPARATUS REQUIRED:

1. 8085 microprocessor kit 1
2. Power card 1
3. Keyboard 1
4. 8085 Simulator and a PC

ALGORITHM

1. Load the address of the first element of the array in HL register pair. (Pointer)
2. Move the count to B-register
3. Increment the pointer
4. Get the first data in accumulator.
5. Decrement the count
6. Increment the pointer
7. Compare the content of memory addressed by HL pair with that of accumulator
8. If carry = 1, go to step 10 or if carry = 0, go to step 9
9. Move the content memory addressed HL to accumulator.
10. Decrement the count.
11. Check for zero of the count. If ZF = 0, Go to step 6, or If ZF = 1 go to next step
12. Store the smallest data in memory.
13. Stop.

PROGRAM TO SEARCH SMALLEST DATA IN AN ARRAY

MEMORY ADDRESS	LABEL	INSTRUCTION	OPCODE	COMMENTS
		LXI H,4200H		;set pointer for array
		MCV B,M		;set count for number of elements in array
		INX H		
		MCV A,M		;Set 1st element of array as smartest data
		DCR B		;Decrement the count.
	LOOP	INX H		;Compare on element of array
		CMP M		;with current smallest data
		JC AHEAD		;If CY = 1, go to AHEAD
		MOV A,M		;If CY = 0 then content of memory
				:is smaller than A. Hence if CY = 0,
				;Make memory as smallest by moving to A
	AHEAD	DCR B		
		JNZ LOOP		; Repeat Comparison until count is zero
		STA 4300H		;Store the smallest data in memory.
		HLT		Stop the Execution

Sample data

Address	Input Data	Address	Output Data
4200	(Count)	4300	(Smallest data in the array)
4201			
4202			
4203			
4204			
4205			
4206			

RESULT:

Thus, an assembly language program for searching a smallest number from an array of given 8-bit number was written, executed and Verified the Result successfully using 8085 kit and simulator.

9. SEARCH THE LARGEST NUMBER FROM AN ARRAY

AIM:

Write an assembly language program to search the largest data in an array of N data stored in memory from 4200H to (4200H + N). The first element of the array gives the number of data in the array. Store the smallest data in 4300H.

APPARATUS REQUIRED:

1. 8085 microprocessor kit 1
2. Power card 1
3. Keyboard 1
4. 8085 Simulator and a PC

ALGORITHM

1. Load the address of the first element of the array in HL register pair. (Pointer)
2. Move the count to B-register
3. Increment the pointer
4. Get the first data in accumulator.
5. Decrement the count
6. Increment the pointer
7. Compare the content of memory addressed by HL pair with that of accumulator
8. If carry = 1, go to step 10 or if carry = 0, go to step 9
9. Move the content memory addressed HL to accumulator.
10. Decrement the count.
11. Check for zero of the count. If ZF = 0, Go to step 6, or If ZF = 1 go to next step
12. Store the smallest data in memory.
13. Stop.

PROGRAM TO SEARCH SMALLEST DATA IN AN ARRAY

MEMORY ADDRESS	LABEL	INSTRUCTION	OPCODE	COMMENTS
		LXI H,4200H		;set pointer for array
		MCV B,M		;set count for number of elements in array
		INX H		
		MCV A,M		;Set 1st element of array as smartest data
		DCR B		;Decrement the count.
	LOOP	INX H		;Compare on element of array
		CMP M		;with current smallest data

		JNC AHEAD		;If CY = 0, go to AHEAD
		MOV A,M		;If CY = 1 then content of memory
				:is largest than A. Hence if CY = 1,
				;Make memory as largest by moving to A
	AHEAD	DCR B		
		JNZ LOOP		; Repeat Comparison until count is zero
		STA 4300H		;Store the smallest data in memory.
		HLT		Stop the Execution

Sample data

Address	Input Data	Address	Output Data
4200	07 (Count)	4300	(Largest data in the array)
4201			
4202			
4203			
4204			
4205			
4206			

RESULT:

Thus, an assembly language program for searching a smallest number from an array of given 8-bit number was written, executed and Verified the Result successfully using 8085 kit and simulator.

10. 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
4. 8085 Simulator and a PC

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
		LDA	4200		Get the ASCII date to A register
		SUI	30H		;Subtract 30h from the data
		CPI	0AH		Compare the result with 0A
		JC	STORE		;If CY = 1, Store the result
		SUI	07H		;Else then subtract 07H
	STORE	STA	4201		Store the result
		HLT			Stop the program

ASCII TABLE

ASCII	HEX	ASCII	HEX
30	0	38	8
31	1	39	9
32	2	41	A
33	3	42	B
34	4	43	C
35	5	44	D
36	6	45	E
37	7	46	F

SAMPLE DATA:

ASCII Input		Hex Output	
4200		4201	

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.

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
4. 8085 Simulator and a PC

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
		LDA 4200		Load A-register with Data
		ANI, 0FH		AND the content of A with 0FH
		STA 4201		Store the Result
		HLT		Stop the program

OUTPUT:

INPUT		OUTPUT	
Address	Data	Address	Data
4200H		4201H	

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

		STA 4201		Store the Result
		HLT		Stop the program

OUTPUT:

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

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:

iii) By using 8086 kit:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
		LDA 4200		Load AL-register with 1st Data
		CMA		NOT the content of AX
		STA 4201		Store the One's complement in memory location.
		INR A		Increment the content of AX.
		STA 4202		Store the Two's complement in memory location
		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 and Emulator8086.

APPARATUS:

1. 8086 microprocessor kit---------- 1
2. Power card---- 1
3. Keyboard --- 1
4. Emulator8086 and PC

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,[1200H]		Get the first data in AX register.
		MOV BX,[1202H]		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 [1206H],CX		Store the carry
		MOV [1204H],AX		Store the sum
		HLT		Stop the program

OUTPUT FOR ADDITION:

INPUT		OUTPUT	
Address	Data	Address	Data
1200		1204	
1202		1206	

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 and Emulator8086.

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 and Emulator8086.

APPARATUS:

1. 8086 microprocessor kit---------- 1
2. Power card---- 1
3. Keyboard --- 1
4. Emulator8086 and PC

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
		MOV CX,0000H		Initialize counter CX
		MOV AX,[1300H]		Get the first data in AX register
		MOV BX,[1302H]		Get the second data in BX register.
		SUB AX,BX		Subtract the contents of both the register AX & BX
		JNC SKIP		Check the Borrow.
		INC CX		If carry exists, increment the CX
		NEG AX		Take two's complement of the difference
	SKIP	MOV [1306H],CX		Store the Borrow.
		MOV [1304H],AX		Store the difference.
		HLT		Stop the program

OUTPUT FOR SUBTRACTION:

INPUT		OUTPUT	
Address	Data	Address	Data
1200		1204	
1202		1206	

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 and Emulator8086.

15. MULTIPLICATION OF 16 BIT NUMBERS

AIM

1. To write and execute an assembly language program to Multiply two 16-bit unsigned numbers in 8086 kit and Emulator8086.

.

APPARATUS:

1. 8086 microprocessor kit---------- 1
2. Power card---- 1
3. Keyboard --- 1
4. Emulator8086 and PC

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

OUTPUT:

INPUT		OUTPUT	
Address	Data	Address	Data
1200		1204	
1202		1206	

RESULT

Thus, an assembly language program for multiplication of given 16-bit numbers was written, executed and Verified the Result successfully using 8086 kit and Emulator8086.

16. DIVISION OF 32BIT BY 16 BIT NUMBER

AIM

To write and execute an assembly language program to Divide two 16-bit unsigned numbers in 8086 kit and Emulaotr8086.

APPARATUS:

1. 8086 microprocessor kit---------- 1
2. Power card---- 1
3. Keyboard --- 1
4. Emulator8086 and PC

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,[1200H]		Initialize DX-register with Lsb of Dividend
		MOV AX,[1202H]		Load AX-register with Dividend
		MOV BX, [1204H]		Load BX-register with Divisor
		DIV BX		Divide AX by BX-register
		MOV [1206H],AX		Store the Quotient
		MOV [1208H],DX		Store the Remainder
		HLT		Stop the program

OUTPUT:

INPUT		OUTPUT	
Address	Data	Address	Data
1200		1206	
1202		1208	
1204			

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:

1. 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 kit and Emulator8086.

APPARATUS:

1. 8086 microprocessor kit ---------- 1
2. Power card ---- 1
3. Keyboard---- 1
4. Emulator8086 and PC

MASKING OF BITS

ALGORITHM:

1. Load the Data in AX-register.
2. Logically AND the content of AX with 0F0FH.
3. Store the result in memory location.
4. Stop the program

PROGRAM:

iv) By using 8086 kit:

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

OUTPUT:

INPUT		OUTPUT	
Address	Data	Address	Data
1200H		1202H	

SETTING OF BITS

ALGORITHM:

1. Load the Data in AX-register.
2. Logically OR the content of AX with 0F0FH.
3. Store the result in memory location.
4. Stop the program

PROGRAM:

v) By using 8086 kit:

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
			MOV AX,[1200H]	Load AL-register with 1st Data
			OR AX, 0F0FH	AND the content of AX with 0F0FH
			MOV [1202H],AX	Store the Result
			HLT	Stop the program

OUTPUT:

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

RESULT:

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

18. 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 Emulator8086.

APPARATUS:

1. 8086 microprocessor kit ---------- 1
2. Power card ---- 1
3. Keyboard---- 1
4. Emulator8086 and PC

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	LABEL	MNEMONICS	OPCODE	COMMENTS
		MOV SI, 1150H		Initialize the source address.
		MOV DI,1250H		Initialize the destination address.
		MOV CX,0006 H		Initialize count value to the count register.
	REPEAT:	CLD		Clear the direction flag.
		MOVSB		Move the string byte.
		LOOP REPEAT		Unconditional loop to address specified by the label REPEAT.
		HLT		Stop the program

OUTPUT:

INPUT		OUTPUT	
Address	**Data**	**Address**	**Data**
1150.		**1250.**	
1151.		**1251.**	
1152.		**1252.**	
1153.		**1253.**	
1154.		**1254.**	
1155.		**1255.**	

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.

19.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 and Emulator8086

APPARATUS:

1. 8086 microprocessor kit ---------- 1
2. Power card ---- 1
3. Keyboard---- 1
4. Emulator8086 and PC

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
		MOV DX,0H		Clear DX
		MOV SI,1250H		Initialize the SI with source address
		MOV CX,03H		Load the count as N-1
		MOV AX, [SI]		Get the first data on AX
	A1:	INC SI		Increment the SI
		INC SI		Increment the SI
		ADD AX, [SI]		Add with next data
		JNC NEXT		If no carry jump to next addition
		INC DX		Else increment DX to count carry
	NEXT:	LOOP A1		Repeat N-1 Times
		MOV [1300H], AX		Store the Sum
		MOV [1302H], DX		Store the Carry
		HLT		Stop the process

OUTPUT:

INPUT		OUTPUT	
Address	**Data**	**Address**	**Data**
1250		**1300**	
1252		**1302**	
1254			
1256			

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. 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
	START	LXI H, 4200		Initialize HL with 4200H
		MVI C, 04		Copy the value 04 to C- register
	NEXT	MOV A, M		Copy the content M to A-register
		OUT C0		The content of A is moved to Out port
		LXI D, 1010		Copy the data 1010 to DE-reg Pair
	loop	DCX D		Decrement DE-register
		MOV A,E		
		ORA D		Check DE = 0000
		JNZ loop		Jump on no zero to loop
		INX H		Increment HL -register Pair
		DCR C		Decrement the count
		JNZ NEXT		Jump to NEXT if Z flag is zero
		JMP START		Jump to label START
		HLT		Stop the program.
	TABLE	09 05 06 0A		**clockwise direction**
	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.

21. 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
6.

ROLLING MESSAGE "HELP US"

ALGORITHM:

Display of rolling message "HELP US "

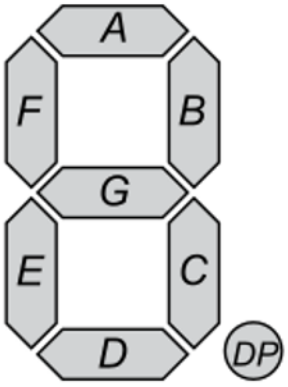
1. Initialize the counter
2. Set 8279 for 8 digit character display, right entry
3. Set 8279 for clearing the display
4. Write the command to display
5. Load the character into accumulator and display it
6. Introduce the delay
7. Repeat from step 1.

PROGRAM:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
	START	MOV SI,1200H		Initialize array
		MOV CX,000FH		Initialize array size
		MOV AL,10		Store the control word for display mode
		OUT C2,AL		Send through output port
		MOV AL,0CC		Store the control word to clear display
		OUT C2,AL		Send through output port
		MOV AL,90		Store the control word to write display
		OUT C2,AL		Send through output port
	NEXT	MOV AL,[SI]		Get the first data
		OUT C0,AL		Send through output port
	DELAY	MOV DX,0FFFFH		Store 16bit count value
	LOOP1	DEC DX		Decrement count value
		JNZ LOOP1		Loop until count values becomes zero
		INC SI		Go & get next data
		LOOP NEXT		Loop until all the data has been taken
		JMP START		Go to starting location
		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	MNEMONICS	OPCODE	COMMENTS
		MOV AL,00		Store the control word for display mode
		OUT C2,AL		Send through output port
		MOV AL,0CC		Store the control word to clear display
		OUT C2,AL		Send through output port
		MOV AL,90		Store the control word to write display

		OUT C2,AL		Send through output port
		MOV AL,8F		Get the first data
		OUT C0,AL		Send through output port
		HLT		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

22. INTERFACE SWITCHES WITH 8086 THROUGH 8255

AIM:

To write and execute an assembly language Program to Interface 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

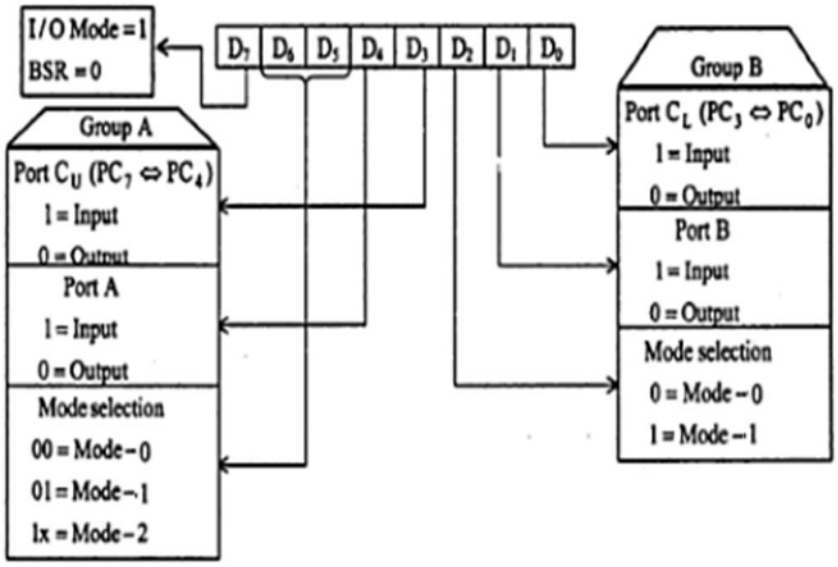
PROGRAM:

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

INPUT	OUTPUT	
VARY THE SWITCH POSITIONS ON OFF ON ON OFF ON OFF ON	Address	Data
	1250	

CONTROL WORD

1 0 0 1 0 0 0 0 = 90H



RESULT

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

23. TRAFFICLIGHT INTERFACE

AIM:

To write and execute an assembly language program for traffic light interfacing to handle the traffic using 8085 Microprocessor.

APPARATUS REQUIRED:

1. 8085 microprocessor kit ----------1
2. Traffic light Interface board 1
3. Power card ----1
4. Keyboard ----1

ALGORITHM:

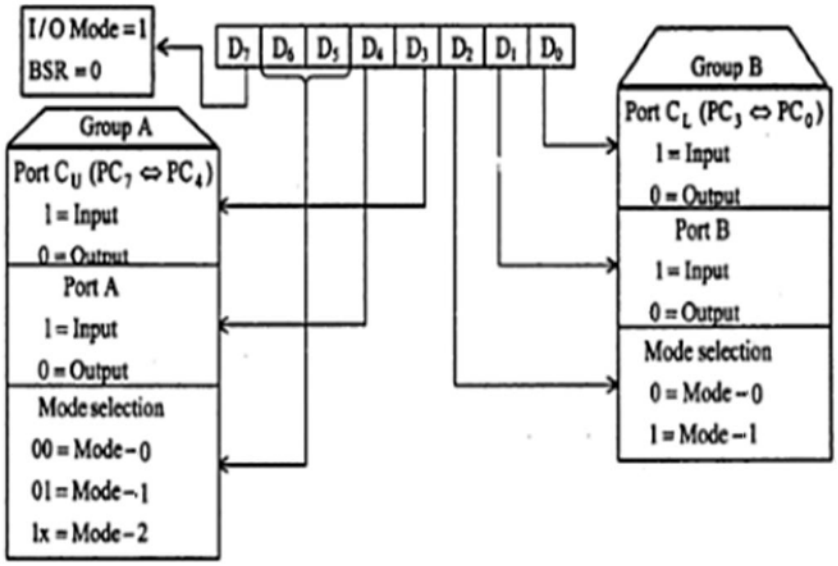
1. Get the control word
2. Send to control register
3. Get the data1
4. Send to PORT A
5. Get the data2
6. Send to PORT B
7. Get the data3
8. Send to PORT C
9. Stop the process

PROGRAM:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
		MVI A,80		Get the control word
		OUT 26		Send to control register
		MVI A, DATA1		Get the data1
		OUT 20		Send to PORT A
		MVI A, DATA2		Get the data2
		OUT 22		Send to PORT B
		MVI A, DATA3		Get the data3
		OUT 24		Send to PORT C
		HLT		Stop the process

CONTROL WORD

1 0 0 0 0 0 0 0 = 80H



INPUT:

Vehicles from North side to all other directions (Assumed Cable connected side as north)

	D8	D7	D6	D5	D4	D3	D2	D1	
Data1	1	0	0	0	0	1	0	0	84H
	D20	D19	D18	D17	DL78	DL56	DL34	DL12	
Data2	1	0	0	1	1	1	1	1	9FH
	D16	D15	D14	D13	D12	D11	D10	D9	
Data3	1	0	0	1	0	0	0	0	90H

OUTPUT:

Green LED's ON	**D16, D17 & D20**	**To direct Vehicles from North side to all other directions**
Red LED's ON	**D3, D8 & D13**	**To stop Vehicles from all other directions**
Dual LED's Colour	**RED**	**To stop pedestrians crossing**

RESULT:

Thus, an assembly language program for traffic light interfacing board with 8085 was written, executed and Verified the output successfully.

24. ADC INTERFACE

AIM:

To write and execute an assembly language program to convert an analog signal into a digital signal using an ADC interfacing.

APPARATUS REQUIRED:

1. 8086 microprocessor kit/MASM ----1
2. ADC Interface board ----1
4. Power card ----1
5. Keyboard ----1

ALGORITHM:

(i) Select the channel and latch the address.
(ii) Send the start conversion pulse.
(iii) Read EOC signal.
(iv) If EOC = 1 continue else go to step (iii)
(v) Read the digital output.
(vi) Store it in a memory location.

PROGRAM:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
		MOV AL,00		Load accumulator with value for ALE high
		OUT 0C8, AL		Send through output port
		MOV AL,08		Load accumulator with value for ALE low
		OUT C8,AL		Send through output port
		MOV AL,01		Store the value to make SOC high in the accumulator
		OUT D0,AL		Send through output port
		MOV AL,00		Introduce delay
		MOV AL,00		
		MOV AL,00		
		MOV AL,00		Store the value to make SOC low the accumulator
		OUT D0,AL		Send through output port
	L1	IN AL, D8H		Read the EOC signal from port & check for end of conversion
		AND AL,01		
		CMP AL,01		

		JNZ L1		If the conversion is not yet completed, read EOC signal from port again
		IN AL,C0H		Read data from port
		MOV BX,1110		Initialize the memory location to store data
		MOV [BX],AL		Store the data
		HLT		Stop

OUTPUT:

Potentiometer Knob Positions (From minimum to Maximum)	DIGITAL DATA ON LED DISPLAY	HEX CODE IN MEMORY LOCATION
1		
2		
3		
4		
5		
6		

RESULT:

Thus the ADC was interfaced with 8086 and the given analog inputs were converted into its digital equivalent and the same was stored in memory

25. ADDITION OPERATION USING 8051 MICROCONTROLLER

AIM:

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

APPARATUS:

1. 8051 microcontroller kit---- 1
2. Power card---- 1
3. Keyboard --- 1
4. Keil and PC

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	LABEL	MNEMONICS	OPCODE	COMMENTS
		MOV A,#data		Load data 1 in accumulator.
		MOV B,#data		Load data 2 in B-register
		ADDC A,B		Add the contents of accumulator and B-reg with carry.
		MOV DPTR,#4500_H		Initialize DPTR with address 4500_H
		MOVX @ DPTR,A		Store the Sum in 4500_H
	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

26. 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
4. Keil and PC

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	LABEL	MNEMONICS	OPCODE	COMMENTS
		MOV A,#data		Load data 1 in accumulator.
		MOV B,#data		Load data 2 in B-register
		SUBB A,B		Subtract the contents of B-reg from accumulator with borrow.
		MOV DPTR,#4500_H		Initialize DPTR with address 4500_H
		MOVX @ DPTR,A		Store the difference in 4500_H
	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

27. 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
4. Keil and PC

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	LABEL	MNEMONICS	OPCODE	COMMENTS
		MOV A,#data		Load data 1 in accumulator.
		MOV B,#data		Load data 2 in B-register
		MUL AB		A*B, Higher byte of result in B and lower byte of result in A.
		MOV DPTR,#4500_H		Initialize DPTR with address 1100_H
		MOVX @ DPTR,A		Store the LSB in 4500_H
		INC DPTR		Increment Data pointer
		MOV A,B		Copy the content of B-reg to A-register.
		MOVX @ DPTR,A		Store the MSB in 4501_H
	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

28. 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
4. Keil and PC

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	LABEL	MNEMONICS	OPCODE	COMMENTS
		MOV A,#data1		Load data 1 in accumulator.
		MOV B,#data2		Load data 2 in B-register
		DIV AB		Divide. Remainder in A and quotient in B
		MOV DPTR,#4500_H		Initialize DPTR with address 1100_H
		MOVX @ DPTR,A		Store the quotient in 4500_H
		INC DPTR		Increment Data pointer
		MOV A,B		Copy the content of B-reg to A-register.
		MOVX @ DPTR,A		Store the Remainder in 4501_H
	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

29. 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
4. Keil and PC

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	LABEL	MNEMONICS	OPCODE	COMMENTS
		MOV A,#data1		Load data 1 in accumulator.
		MOV B,#0F		Load data 2 in B-register
		ORL A,B		OR the contents of accumulator and B-reg.
		MOV DPTR,#4500_H		Initialize DPTR with address 4500_H
		MOVX @ DPTR,A		Store the Result in 4500_H
	STOP:	SJMP STOP		Stop the program

OUTPUT:

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

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	LABEL	MNEMONICS	OPCODE	COMMENTS
		MOV A,#data1		Load data 1 in accumulator.
		MOV B,#0F		Load data 2 in B-register
		ANL A,B		AND the contents of accumulator and B-reg.
		MOV DPTR,#4500_H		Initialize DPTR with address 4500_H
		MOVX @ DPTR,A		Store the Result in 4500_H
	STOP:	SJMP STOP		Stop the program

OUTPUT:

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

RESULT:

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