Program to ADD TWO 8-BIT DATA.

Address		Memonics	OP Code	Comments
4100		MVI C, 00	0E, 00	Clear C reg to Count carry
4102		LDA 4200 H	3A, 00, 42	Get 1st data in A and Save in B
4105		MOV B, A	47	
4106		LDA 4201 H	3A, 02, 42	Get 2nd data in A-register
4109		ADD B	80	Get the Sum in A register
410A		JNC SKIP	D2	If CY=0 then SKIP next step
410B		INR C	0C	Increment C register to Count the carry
410C	SKIP	STA 4202 H	41, 0C	Store the Sum in memory.
410E		MOV A, C	32	Move the carry
410F		STA 4203 H	02, 42	
4111		HLT	79	Stop the Execution

Input		Output	
Address	Data	Address	Data
4200	FF	4202	FE (sum)
4201	FF	4203	01 (carry)

(1) 8-BIT ADDITION WITH CARRY USING DIRECT ADDRESSING

Page No.

Aim:-

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

Apparatus Required:-

1. 8085 micro processor kit — 1
2. Power cord — 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 accumulator and move to B-register.
3. Load the second data from memory to accumulator.
4. Add the content at 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

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.

Program to ADD TWO 8-BIT DATA

Memory address	Label	Instruction	opcode	comments
4100		MVI B,00	06,00	Clear B reg to count carry
4102		LHLD 4200H	2A,00,42	Get 1st data in HL Pair and save in DE
4105		XCHG	EB	
4106		LHLD 4202H	8A,02,42	Get 2nd data in HL-register
4109		DAD D	19, D2	ADD HL with DE and Get the SUM in HL register
410A		JNC SKIP	D2	If CY=0 then skip next step
410B		INR B	0E	Increment C register to count the carry.
410C	SKIP	SHLD 4204H	41,04	Store the sum in memory
410E		MOV A,B	22	Move the carry to accumulator and store in memory
410F		STA 4206H	04, 42	
4111		HLT	78	Stop the Execution.

Input		Output	
Address	Data	Address	Data
4200	ff	4204	FE (sum)
4202	ff	4206	01 (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.

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.

PROGRAM TO SUBTRACT TWO 8-BIT DATA:

Memory address	Label	Instruction	opcode	Comments
4101		LDA 4201 H	3A, 01, 42	Get 2nd data in A and Save in B
4104		MOV B, A	3F	
4105		LDA 4200H	3A, 01, M2	Get 1st data in A-register
4108		SUB B	47	Subtract B-Reg from A register
4109		JNC SKIP	21, 00, 00	If CY=0 Then Skip next two steps
410C		INR C	16, 00	Increment C register to count the carry.
410E		CMA	19	Take two's complement of difference
410F		INR A	C2	
4113	SKIP	STA 4202H	OB	Store the Difference in memory
4116		MOV A, C	C2	Move the Borrow to accumulator and store in memory
4113		STA 4203 H	32, 03, 42	
4110		HLT	76	Stop the Execution

SAMPLE DATA:-

Input		Output	
Address	Data	Address	Data
4200	01 02	4202	04 06
4201	03 04	4203	00 00

(3) 8-BIT SUBTRACTION WITH BORROW USING DIRECT ADDRESSING

Page No.

Aim:-
To write an assembly language program to subtract two numbers of 8-bit data stored in memory locations 4200H and 4201H and store the result in 4202 H and 4203 H 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

Result:-
Thus an assembly language program for subtraction of given two 8-bit numbers with borrow was written executed and verified the result successfully using 8085 kit.

PROGRAM TO MULTIPLY TWO NUMBERS OF 8-BIT DATA:-

Memory address	Label	Instruction		opcode	Comments.
4100		LDA	4200H	3A,00,42	; Get 1st data in A
4103		MOV	E,A	5F	; Save 1st data in E
4104		LDA	4201 H	3A,01,42	; Get 2nd data in A
4107		MOV	B, A	47	; Save 2nd data in B
4108		LXI	H, 0000H	21,00,00	; Clear HL Pair (initial sum=0)
410B		MVI	D, 00H	16,00	; Clear E for accounting overflow
410D		DAD	D	19,05,C2	; Add the content of DE to Sum (HL)
4110		DCR	B	05	Decrement data 2 for every addition
4111		JNZ	NEXT	C2	; Repeat addition until count is zero
4112		SHLD	4202H	22,02,42	; Store the product in memory
4115		HLT		76	Stop the Execution.

Address	Input Data	Address	output Data.
4200	03	4202	06
4201	02	4203	00

8-BIT MULTIPLICATION OPERATIONS USING 8085 MICROPROCESSOR

Page No.

AIM:-

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

Apparatus Required:-

1. 8085 microprocessor kit — 1
2. Power cord — 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.
* Clear HL pair (initial sum)
* Clear D for overflow.
* 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 the next step.
* Store the content of HL in memory.
* Stop

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.

PROGRAM TO DIVIDE TWO NUMBERS OF 8-BIT DATA:-

Memory address	Label	Instruction		comments
4100		LDA	4201 H	
4103		MOV	B,A	; Get the divisor in B register
4104		LDA	4200 H	; Get the dividend in A register
4107		MVI	C,00H	; clean C register for quotient
4109	AGAIN:	CMP	B	
4100		JC	STORE	; If divisor is less than dividend goto
410E		SUB	B	; subtract divisor from dividend
410F		INC	C	; quotient by one for each subtraction
4112		JMP	AGAIN	
4115	STORE:	STA	4203 H	; store the remainder in memory
4116		MOV	A,C	
4119		STA	4202 H	; Store the quotient in memory
411A		HLT		Stop the Execution

Sample data:-

Address	Input Data	Address	output Data
4200	02	4202	01
4201	03	4203	00

5

8-BIT DIVISION OPERATION USING 8085 MICROPROCESSOR.

Page No.

Aim:-
To written an ALP to perform division of two 8 bit number stored in memory location 4200H 4201H and stored the remainder in 4202H and the quotient in 4203H.

Apparatus Required:-
1. 8085 microprocessor kit —①
2. Power cord —①
3. Keyboard —①
4. 8085 simulator and a PC.

Algorithm:-
1. Load the divisor in accumulator and move it to B register.
2. load the dividend in accumulator
3. Clean 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 C-register (quotient)
7. Increment the content of C-register (quotient)
8. Go to step 4.
9. Store the content of accumulator in memory.
10. Move the content of C- register to accumulator and store in memory.
11. Stop.

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.

PROGRAM TO SORT AN ARRAY OF DATA IN ASCENDING ORDER

Memory address	Label	Instruction	OP code	Comments
4100		LDA	4200 H	; Load the count value
4108		MOV	B, A	; Set counter for (N-1) repetition
4104		DCR	B	; of (N-1) comparisons.
4105	LOOP 2	LXI	H, 4200 H	; Set pointer for array.
4108		MOV	C, M	; Set count for (N-1) comparison
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 go to AHEAD
4110		MOV	DM	; If the content of A is greater than
4111		MOV	M, A	; then content of memory
4113		DCX	H	; Pointed content by HL and previous
4116		MOV	M, D	
4117		INX	H	
4118	AHEAD	DCR	C	Repeat comparison until C Count is zero.
411B		JNZ	Loop-1	; Repeat until B count is zero.
411C		JNZ	Loop-2	
411D		HLT		Stop the Execution.

Sample Data :-

Address	Data Array
4200	07
4201	AB
4202	92
4203	84
4204	4F
4205	60
4206	F2
4207	34

Address	Data Array.
4200	07
4201	34
4202	4F
4203	69
4204	84
4205	92
4206	AB
4207	F2

⑥ SORT AN ARRAY OF DATA IN ASCENDING ORDER.

Page No.

Aim:-

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

Apparatus required:-

1. 8085 microprocessor kit - ①
2. Power cord - ①
3. Keyboard - ①
4. 8085 simulator and a PC.

Algorithm:-

1. Load the count value from the memory to A-register and save in B.
2. Decrement B- register
3. Set HL pair as data address pointer.
4. Set C-register as counts for (N-1) comparisons.
5. Load the data at the array in accumulator using data address
6. Increment the HL pair.
7. Compare the data pointed by HL with acc.
8. Exchange the content of memory pointed HL and the A.
9. Decrement C-register. If zero flag is reset to go to step 6 or next step.
10. Decrement B register if zero flag is reset to go to step 5 or or next step.
12. STOP.

Result:- Thus, an Assembly language programme for sorting in ascending order of an unsorted array of given 8 bit number was written, executed and verified the Result using 8085 kit.

Program to Sort an Array of data in decreasing 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	LOOP2	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	LOOP1	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
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	
4117		INX	H	
4116	AHEAD	DCR	C	; Repeat comparisons until C count is zero
417		JNZ	LOOP1	
411A		DCR	B	; Repeat until B count is zero
411B		JNZ	LOOP2	
411E		HLT		Stop the execution.

SAMPLE DATA:-

Address	Data Array	Address	Data Array
4200	07 (count)	4200	07
4201	AB	4201	F2
4202	92	4202	AB
4203	84	4203	92
4204	8F	4204	84
4205	69	4205	69
4206	F2	4206	4F
4207	34	4207	34

(7) SORT AN ARRAY of DATA IN DESCENDING ORDER

Page No.

Aim:-
To write an ALP to sort and array in descending order sorted in memory in starting from 4200H. The first element of the array gives the count value for the number of element in the array.

Apparatus:-
* 8085 - Microprocessor kit - 1
* Power cord - 1
* Keyboard - 1
* 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 then go to step to otherwise go to next step.

Result:-
Thus an ALP from sorting in descending order of an unsorted array of given 8 bit number was written executed verified using 8085 kit.

Program to search smallest DATA IN AN ARRAY

MEMORY ADDRESS	LABEL	INSTRUCTION	COMMENTS
4100		ORG 4100H	; Assembly directive
4102		LXI H, 4200H	; Set pointer for array
4103		MCV B, M	; Set count for number of elemnt
4104		INX H	
4105		MCV A, M	; Set 1st element of array as smalle
4106		DCR B	; Decrement the count
4107	LOOP	INX H	; Compare an element of array
4108		CMP M	; With current smallest data
4108		JC AHEAD	; If CY=1, go to AHEAD.
410C		MOV A, M	; If CY=0 then content of memory
410D			; is smaller than A. Hence if CY=0
4110			; Make memory as smallest by moving A
4113	AHEAD	DCR B	
4114		JNZ LOOP	; Repeat comparison until count is 0
411A		STA 4300H	; Store the smallest data in memory
411B		HLT	Stop the Execution.

Sample data :-

Address	Input Data	Address	output Data
4200	07	4300	1C
4201	42		
4202	3A		
4203	1C		
4204	2F		
4205	B4		
4206	25		

(8) SEARCH THE SMALLEST NUMBER FROM AN ARRAY.

Page No.

Aim :- Write an ALP 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 give the number of data in the array. Store the smallest data in 4300H

Apparatus Required :-

1. 8085 microprocessor kit - (1)
2. Power Cord - (1)
3. Keyboard - (1)
4. 8085 simulator and a pc.

Algorithm :-

* Load the address of the first element of the array in HL register pair.
* Move the count to B-register.
* Increment the point.
* Get the first data in ACC.
* Decrement the count.
* Increment the point.
* Compare the content of memory addressed by HL pair with ACC.
* If carry=1, go to step 10 if Carry=0, go to step-9.
* Move the content memory addressed HL - to ACC
* Decrement the count
* Check the zero of the count.
* Store the smallest data in memory.
* Stop.

Result :-

Thus, An ALP of the searching the smallest number in the array has been executed, verified by using 8085 kit.

PROGRAM TO SEARCH SMALLEST DATA IN AN ARRAY

MEMORY ADDRESS	LABLE	INSTRUCTION	COMMENTS
4000		ORG 4100H	;Assembler directive
4003		LXI H,4200H	;set pointer for array
4004		MOV B,M	;set count for number of elemen
4005		INX H	
4006		MOV A,M	;Set 1st element of array of data
4007		DCR B	;Decrement the count
4008	LOOP	INX H	;Compare an element of array
4009		CMP M	;with current smallest data
400C		JNC AHEAD	;If CY=0 go to AHEAD
400D		MOV A,M	;If CY=1 then content of M is
400E			;is largest than A. Hence if CY=1
4010			;Make memory as largest number A
4014	AHEAD	DCR B	
4016		JNZ LOOP	;Repeat comparison until count is zero
4018		STA 4300H	;store the smallest data in memory
411A		HLT	stop the execution

Sample data:-

Address	Input Data	Address	output Data
4200	07	4300	B4
4201	42		
4202	3A		
4203	1C		
4204	24		
4205	B4		
4206	25		

(9) SEARCH THE LARGEST NUMBER FROM AN ARRAY

Page No.

Aim:-
write an ALP to search largest number in an array of N data stored in memory from 4200H - 10(4200H+N). The first element of array give the number of data in the array. Store the smallest number in 4300H

Apparatus Required:
1. 8085 Microprocessor kit - 1
2. Power Cord.
3. Key board.
4. 8085 Simulator and a PC.

Algorithm:-
1. Load the address of the first element of the array in HL register pair.
2. Mov the count to B - register.
3. Increment the pointer.
4. Get the first data in ACC.
5. Decrement the count.
6. Increment the pointer.
7. Compare the memory of content
8. If carry = 1, go to 10, If carry = 0 go to 9.
9. Mov the content address to HL to ACC.
10. Decrement the count.
11. Check for zero count. If ZF = 0, go to Step 6. If ZF = 1 go to next step
12. Store the smallest data in memory.
13. Stop.

Result:-
Thus, An ALP for searching largest number in Array is executed successfully and verified by using 8085 kit.

Program to convert ASCII CODE TO BINARY VALUE

Memory address	Label	Instruction	OP code	comments
4000		LDA 4200	3A00 42	Get the ASCII data to A register
4002		SUI 30H	D6 E	; subtract 30h from the data
4003		CPI 0AH	FE 0A	Compare the result with 0A
4005		JC STORE	DA 0C 40	;If CY=1, Store the result
4008		SUI 07H	D6, 07, 32	; Else then subtract 07H
400A	STORE	STA 4201	32, 01, 42	Store the result
400F		HLT	76	Stop the program

SAMPLE DATA:-

ASCII INPUT		HEX output	
4200	41	4201	0A

(10) ASCII CODE TO HEX CODE CONVERSION Page No.

AIM:-
To write an ALP to convert an array of ASCII to corresponding binary value in 8085 µP.

Apparatus required:-
1. 8085 microprocessor — 1
2. Power cord — 1
3. key board — 1
4. 8085 simulator and a PC.

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

Result:-
Thus, ALP to convert ASCII code to Hex has been executed and verified successfully by 8085 kit.

Program:-

i) By using 8085 kit

ADDRESS	LABLE	MNEMONICS	OPCODE	COMMENTS
4100		LDA 4200	3A, 00, 42	Load A-register with Data
4103		ANI, 0FH	E6, 0F	AND the content of A with 0FH
4105		STA 4201	32, 01, 42	Store the Result
4108		HLT	76	Stop the program

output:-

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

Program:-

ii) By using 8085 kit

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4100	3A, 00, 42		LDA 4200	Load A-register with Data
4103	F6, 0F		ORI, 0FH	OR the content of A with 0FH
4105	32, 01, 42		STA 4201	Store the Result
4108	FA 76		HLT	Stop the program

Output:-

Input		output	
Address	Data	Address	Data
4200H	C5	4201 H	CF

(10) MASKING AND SETTING OF LOWER NIBBLES ON GIVEN DATA.

Page No.

AIM:- To write and execute an ALP for performing Masking setting one's and Tow's compliment of given data of 8-bit using 8085 MP.

Apparatus Required:-

* 8085 MP - kit — 1
* Power Cord — 1
* Keyboard — 1
* 8085 simulator and a PC.

Masking of bits:-

Algorithm:-

1. Load the data in A.
2. Logically AND the content of A with of H.
3. Store the result in memory location.
4. Stop the program.

Setting of Bits:-

1. Load the data in A.
2. Logically ORI content of A with of H.
3. Store the result in memory location.
4. Stop.

Result:-

Thus, Am, ALP for masking and setting of the given data has been excuted, verified succesfully by 8085 kit.

PROGRAM:-

i) By using 8086 kit:-

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4100	3A 0042		LDA 4200	Load AL-register with 1st Data
4103	2F		CMA	Not the content of A Y
4104	3201 42		STA 4201	Store the one's complement in memory location
4107	3C		INR A	Increment the content of Ax
4108	32 0242		STA 4202	Store the TWO's complement in memory location
410B	F6		HLT	Stop the program

OUTPUT:-

Input		Out put	
Address	Data	Address	Data
4200H	AB	4201 H	54
		4202 H	55

(12) ONE'S AND TWO'S COMPLEMENT

Page No.

Aim:-

To write an ALP for performing one's & Two's compliment of given 8-bit number using 8085 MP

Apparatus:-

1. 8085 Mp kit — 1
2. Power Card — 1
3. keyboard — 1

Marking 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. Instrumment the content of A.
5. Store the TWO's complement in memory location.
6. STOP.

Result:-

Thus, An ALP to performing one's & TWO's compliment of bits were executed using 8085 kit.

PROGRAMM :-

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000		MOV CX, 0000H	B9, 00, 00	Initialized counter Cx.
1003		MOV AX, [1200]	A1, 00, 12	Get the first Data in AX register.
1006		MOV BX, [1202]	8B, 1E, 02, 12	Get the Second Data in BX register.
100A		ADD AX, BX	01, D8	Add the contents of both the register AX & BX
100C		JNC L1	73, 01	check for Carry
100E		INC CX	41	If carry exist, increament the Cx.
1010	L1	MOV [1206], CX	89, 0E, 06, 12	Store the carry.
1014		MOV [1204], AX	A3, 04, 12	Store the sum
1017		HLT	F4	Stop the program

OUTPUT FOR ADDITION :-

INPUT		OUTPUT	
Address	Data	Address	Data
1200	01 06	1204	05 09
1202	04 03	1206	00 00

13. ADDITION OF 16 BIT NUMBERS WITH CARRY

Page No.

Aim :-

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

Apparatus Required

1. 8086 microprocessor kit — 1
2. Power cord — 1
3. Key board — 1
4. Emulator 8086 and PC.

ALGORITHM :-

1. Load the first Data in AX-register.
2. Load the first Data in BX register.
3. Add the two data and get the sum in AX - register.
4. If C=0 then skip next step
5. Store the sum in memory location
6. Store the carry in memory location
7. Stop the programm

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 emulator 8086.

PROGRAMM :-

ADDRESS	LABEL	MNEMONICS	OP CODE	COMMENTS
1100		MOV CX, 0000H	C7, C1, 00, 00	Initialize counter CX
1104		MOV AX, [1300]	8B, 06, 00, 13	Get the first data in AX regi
1108		MOV BX, [1302]	8B, 1E, 02 13	Get the second data in BX regi
110C		SUB AX, BX	29, D8	sub the content of both AX & BX
110E		JNC SKIP	73, 03	Check the Borrow
1110		INC CX	41	If carry exist, increment the CX
1111		NEG AX	F4 D8	Take Two's complement of the difference.
1113	SKIP	MOV [1306], CX	89 0E 0613	Store the Borrow
1117		MOV [1304] AX	89, 06 04, 13	Store the difference
111B		HLT	F4	Stop the program

OUT FOR SUBTRACTION

~~OUT PUT~~ INPUT		OUT PUT	
Address	Data	Address	Data
1200	05 06	1204	04 04
1202	01 02	1206	00 00

(14) SUBTRACTION OF 16 BIT NUMBERS WITH BORROW Page No.

Aim:-

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

Apparatus Required:-

① 8086 microprocessor kit —— ①
② Power card - ①
③ keyboard —— ①
④ Emulator 8086 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 storage.
7. Store the Difference in memory.
8. Move the borrow to accumulator and store in memory.
9. Stop.

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 Emulator 8086.

Programm :-

ADDRESS	LABEL	MNEMONIC	OPCODE	COMMENTS
1200		MOV AX, [1200]	8B,06,00,14	load Ax-register with 1st Data
1204		MOV BX, [1202]	8B,1E,02,14	load Bx-register with 2nd Data
1208		MUL BX	F7 E3	Multiply the contents of Ax with Bx register
120A		MOV [1204], AX	89,06,04,14	Store the lower word
120E		MOV [1206] DX	89,16,06,14	Store the Higher word
1212		HLT	F4	Stop the program

OUTPUT :-

INPUT		OUTPUT	
Address	Data	Address	Data
1200	02 02	1204	08 04
1002	02 02	1206	00 04

(15) MULTIPLICATION OF 16 BIT NUMBERS

Page No.

AIM :-
To write and execute an assembly language program to multiply two 16-bit unsigned numbers in 8086 kit and Emulator 8086.

APPARATUS :-
1. 8086 microprocessor kit ——— (1)
2. Power Cord ——— (1)
3. Reg board ——— (1)
4. Emulator 8086 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.

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

PROGRAM:-

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000		MOV DX, [1200H]	B0, 04, 0012	Initialize Dx-register with LSB of Dividend
1004		MOV Ax, [1208]	B0, 06, 0212	Load Ax-register with Dividend
1008		MOV Bx, [1204]	B1, 1E 04 12	Load Bx-register with Divisor
100C		DIV CX	f3 f7	Divide Ax by Bx-register
100E		MOV [1206], Ax	89, 06 1612	Store the Quotient
1013		MOV [1209], Dx	89, 06 0912	Store the Remainder
1016		HLT	F4	Stop the program

OUTPUT:-

INPUT		OUTPUT	
Address	Data	Address	Data
1200	05 04	1206	14 02
1202	03 06	1208	E3 F7
1204	01 01		

(16) DIVISION OF 32 BIT BY 16 BIT NUMBER

Page No.

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

Apparatus Required:-
1. 8086 microprocessor kit — 1
2. Power Cord — 1
3. Keyboard — 1
4. Emulator 8086 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 Remainder in memory from Dx.
6. Stop.

Result:-
Thus an assembly language program for Division of given 16-bit numbers was written, executed and verified.

PROGRAM:-

i) By Using 8086 kit:

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
1100	8B 06 00 12		MOV AX, [1200]	Load AL-register with 1st Data
1104	81 E0 0F 0F		AND AX, 0F0FH	AND the Content of Ax with 0F0FH
1108	89 06 02 12		MOV [1202], AX	Store the Result
110C	F4		HLT	Stop the program

OUTPUT:-

INPUT		OUTPUT	
Address	Data	Address	Data
1200H	FF FF	1202 H	0F 0F

Program:-

i) By using 8086 kit:-

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
1300	8B 06 00 12		MOV AX, [1200]	Load AL-register with 1st data
1304	81 C8 0F 0F		OR AX, 0F0FH	AND the content of Ax with 0F0F H
1308	89 06 02 12		MOV [1202], AX	Store the Result
130C	F4		HLT	Stop the Program

INPUT		OUTPUT	
Address	Data	Address	Data
1200H	00 00	1202H	0F 0F

17. LOGICAL OPERATION

Page No.

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 kit and Emulator 8086.

Apparatus Required:-

1. 8086 microprocessor kit —— 1
2. Power Card —— 1
3. key board —— 1
4. Emulator 8086 and PC

Masking of Bits:-

ALGORITHM:-

1. Load the Data in Ax-register
2. Logically AND the content of Ax with 0F0F H.
3. Store the result in memory location.
4. Stop the program.

Setting of Bits

ALGORITHM:-

1. Load the Data in Ax-register.
2. Logically OR the content of Ax with 0F0F H
3. Store the result in memory location
4. Stop the program

Result:-

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

Program:-

i) By using 8086 kit

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
1000	C7C60012		MOV SI, 1150 H	Initialize the source addres
1004	C7C70013		MOV DI, 1250 H	Initialize the destination address
1008	C7C10600		MOV CX	Initialize count value to the count register
100C	FC	REPEAT:	CLD	clear the direction flag.
100D	A4		MOVSB	Move the stright byte
100E	E2, F3		LOOP REPEAT	Unconditional loop to addr specified by the table REPEA
1010	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

18. MOVE A DATA BLOCK WITHOUT OVERLAP

Page No.

Aim:-

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

Apparatus Required:-

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

Result:-

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

PROGRAM:-

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000		MOV DX, 0H	C7 C2 0000	Clear Dt
1004		MOV SI, 1250H	C7 C6 5012	
1008		MOV CX, 03 H	C7 C1 0300	
100C		MOV AX, [SI]	8B 04	
100E	A1:	INC SI	46	
100F		INC SI	46	
1010		ADD AX, [SI]	03 04	
1012		JNC NEXT	73 01	
1014		INC DX	42	
1015	NEXT:	LOOP A1	E2 F7	
1017		MOV[1300H], AX	89 06 0013	
101B		MOV[1302H], DX	89 16 0213	
101F		HLT	F4	

OUTPUT:-

INPUT		OUTPUT	
Address	Data	Address	Data
1250	02 01	1300	09 36
1252	01 01	1302	00 00
1254	02 03		
1256	03 01		

19. SUM OF N NUMBERS IN A WORD ARRAY

Page No.

AIM:-
To write and execute an assembly language program for adding N numbers in a word array using 8086 kit and Emulator, 8086.

Apparatus Required
1. 8086 microprocessor kit - 1
2. Power cord - 1
3. Keyboard - 1
4. Emulator 8086 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 pointer and decrement counter.
7. Repeat steps 4, 5 and 6 until counter equal to zero.
8. Stop.

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

PROGRAM :-

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
4100	START	LXI H, 4200	21,00,42	Initialize HL with 4200H
4103		MVI C, 04	06,04	Copy the value 04 to C-register
4105	NEXT	MOV A, M	7E	copy the content M to A-register
4106		OUT C0	D3,C0	The Content of A is Moved to o/p
4108		LXI D, 1010	21,10,10	copy the data 1010 to DE-register
410B	LOOP	DCX D	1B	Decrement DE register
410C		MOV A,E	7B	
410D		ORA D	B2	check DE = 0000
410E		JNZ LOOP	C2,0B,5023	Jump on zero to loop
4111		INX H	05	Increment HL-register pair
4112		DCR C	C2	Decrement the count
4113		JNZ NEXT	05,50	Jump to NEXT if Z flag is
4115		JMP START	C2	Jump to label START
4118		HLT	05,50	Stop the program
4200	TABLE	09 05 06 0A	C3,00	clockwise direction
4200	TABLE	0A 06 05 09	50	counter clockwise direction

OUTPUT :-

Input		output
Address	Data	Motor Rotated on.
4200	09 05 06 0A	clockwise direction
4200	0A 06 05 09	Anti clockwise direction

20. STEPPER MOTOR INTERFACING.

Page No.

AIM:-
To write an execute an assembly language programs to run a stepper motor at different speed, and to control its direction using 8085 mp.

Apparatus Required.
1. 8085 microprocessor kit : —— 1
2. Stepper Motor : —— 1
3. Stepper Motor Interface board —— 1
4. Power card —— 1
5. keyboard —— 1

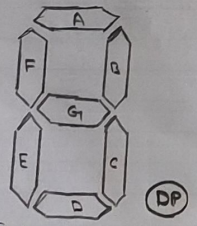
Result:-
Thus an assembly language program to run the stepper motor in both forward and revers direction with delay was executed and its overput was verified.

PROGRAM:-

ADDRESS	LABEL	PROGRAM	OPCODE	COMMENTS
1100		MOV SI, 1200H	C7, C6, 0012	Initialize array
1104		MOV CX, 000FH	C7, C1, 0F, 00	Initialize array size
1108		MOV AL, 10	C6, C0, 10	Store the control word for display
110B		OUT C2, AL	E6 C2	Send through output port
110D		MOV AL, 0CC	C6 C0 C0	Store the control word to clear
1110		OUT C2, AL	E6 C2	Send through output port
1112		MOV AL, 90	C6 C0 90	Send the control word to write
1115		OUT C2, AL	E6 C2	Send through output port
1117	NEXT	MOV AL, [SI]	8A, 04	Get the first data
1119		OUT C0, AL	E6, C0	Send through output port
111B	DELAY	MOV DX, 0FFFFH	E8, E2 0F	Store 16 bit count value
111F	LOOP1	DEC DX	4E	Decrement count value
1120		JNZ LOOP1	E2 FB	Loop until count value becomes zero
1122		INC SI	E9 0C FF	Go & get next data
1123		LOOP NEXT	CA C2 FF A0	Loop until all the data has taken
1125		JMP START	4A 75 ED	Go to starting location
1127		HLT	F6	Exit Program

LOOK-UP Table :-

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



ON-0 OFF-1

Page No.

21. KEYBOARD AND DISPLAY.

Aim:-

To write and execute an assembly language program to display a character, 7 and the rolling message "HELP US" to the display.

Apparatus Required:-

1. 8086 MP kit
2. 8279 interface board.
3. Power cord
4. Keyboard.

Rolling Message help us:-

Algorithm:-

1. Initilize 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 ACC and display it.
6. Introduce the delay.
7. Repeat from step-1

Display the character-7

Algorithm:-

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

OUTPUT

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
1209H	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

PROGRAM

ADDRESS	LABEL	Program	OP code	COMMENTS
1100		MOV AL,00	C6 C0 00	Store the control word for O/P's
1103		OUT C2,AL	E6 C2	Send through output port
1105		MOV AL,0CC	C6 C0 CC	Store the control word to clear
1108		OUT C2,AL	E6 C2	Send through output port
110A		MOV AL,90	C6 C0 90	Store the control word to write
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	FA	Stop Program

INPUT		OUTPUT	
Address	Data	Address	Data
1111	0B	Display	3

Page No.

Result:

Thus, the rolling message Helpers & 3 are displayed using 8279 interface kit with 8086 MP.

PROGRAM:-

Address	LABEL	Program	OPCODE	Comment
1100		MOV AL,90	C6,C0,90	load 90 in AL with control code
1103		OUT C2,AL	E6,C2	send to control reg address
1105		IN AL,C0	E4,C0	Read Port A
1108		MOV [1250],AL	89,06,12,02	Store the result
1114		HLT	F4	Stop

Output:-

Input	OUTPUT	
Vary the switch position on off on on off on off on	Address	Data
	1250	

22. INTERFACE SWITCHES WITH 8086 THROUGH 8255

Page No.

Aim:-

To write and execute ALP to interface 8 switches with 8086 MP through 8255 PPI

Apparatus required:-

1. 8086 microprocessor kit — 1
2. 8255 interface board — 1
3. Power cord — 1
4. ~~Stop~~ 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.

Result:-

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

PROGRAM:-

Address	Lable	Program	opcode	comments
1200		MOV AL,80	16,C6,80	
1202		OUT 26,AL	E6,26	
1205		MOV AL,84	C6,C0,84	
1208		OUT 80,AL	E6,20	
120A		MOV AL,9F	C6,C0,4F	
120D		OUT 22,AL	C6,22	
120F		MOV A,90	C6,C0,90	
1212		OUT 24,AL	C6,24	
1214		HLT	F4	

23. TRAFFIC LIGHT INTERFACE

Page No.

AIM:-

To write and execute an assembly language program for traffic ~~ting~~ 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

Result:-

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

PROGRAM :-

Address	OPCODE	LABEL	Program	comments.
4100	74,05		MOV A, #data	Load data 1 in Accumulator
4102	76,F0,05		MOV B, #data	Load data 2 in B-register
4105	35, 50		ADDC A, B	Add the content of acc
4107	90,11,00		MOV DPTR, #4500	DPTR=0 in which data
410A	F0		MOV @DPTR, A	store the sum in 4500
410B	80.fe		SJMP STORE	STOP

OUT PUT :-

Input		Output	
Register	data	Address	data
4101		4500	
4104			

24. ADDITION OPERATION USING 8085 MICROCONTROLLER.

Page No.

Aim :-

To write and execute an assembly language program to add of two 8-bit numbers using 8051 and keil software

Apparatus Required :-

① 8051 microcontroller kit — ①
② Power cord — ①
③ key board — 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.

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.

PROGRAM :-

ADDRESS	OP CODE	LABEL	PROGRAM	COMMENTS
4100	74,05		MOV A,#data	Load data 1 in Acc
4102	75,F0,04		MOV B,#data	load data 2 in B-reg
4105	95,F0		SUBB A,B	subtract the contents of R
4107	90,11,00		MOV DPTR,#4500	initialize DPTR address 4500
410A	F0		MOV X@DPTR,A	Store the difference to
410B	80,FE	STOP:	SJMP STOP	Stop the program

OUT PUT :-

INPUT		OUTPUT	
Address	Data	Address	Data
4101		4500	
4104			

25. SUBTRACTION OPERATION USING 8051 MICROCONTROLLER.

Page No.

Aim :-

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

Apparatus Required :-

1. 8051 microcontroller kit — 1
2. Power Card — 1
3. Key board — 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.

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.

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 Register
4105	A4		MUL AB	A*B. Higher byte of result in B and low bites of result in A
4106	90,11,00		MOV DPTR,#4500H	Initialize DPTR with address 1100 H
4109	F0		MOVX @DPTR,A	Store the LSB in 4500H
410A	A3		INC DPTR	Increment Data pointer
410B	E5,F0		MOV A,B	Copy the content of B-reg to A-reg.
410D	F0		MOVX @DPTR A	Store the MSB in 4501 H
410E	80,FE	STOP:	SJMP STOP	Stop the Program

OUTPOT :-

INPOT		Output	
Register	DATA	ADDRESS	DATA
4101		4500	
4104		4501	

26. MULTIPLICATION OPERATION USING 8051 MICROCONTROLLER

Page No.

Aim:

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

Apparatus Required:

1. 8051 microcontroller kit — 1
2. Power cord — 1
3. Key board — 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 location
5. Stop the program

Result:

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

PROGRAM:-

ADDRESS	OPCODE	LABEL	PROGRAM	COMMENTS
4100	74, data 1		MOV A, #data 1	Load data 1 in accumulator
4102	75, data 2		MOV B, #data 2	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	Initialize DPTR with address 1100H
4108	F0		MOVX @DPTR A	store the quotient in 4500H
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 4501H
410D	80, FE	STOP:	SJMP STOP	stop the Program

OUT POT:-

INPOT		OUTPOT	
REGISTER	DATA	ADDRESS	DATA
4101		4500	(quotient)
4104		4501	(remainder)

27. DIVISION OPERATION USING 8051 MICRO CONTROLLER

Page No.

AIM:-

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

Apparatus Required:-

1. 8051 Microcontroller kit —①
2. Power Cord —①
3. Keyboard —①
4. keil and pc —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.

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.

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	Initialize DPTR with adres
410A	F0		MOVX @DPTR, A	Store the Result in 4500H
410B	80,FE	STOP:	SJMP STOP	Stop program.

Out Put :-

Input		Output	
Register	Data	Address	Data
4101	C3	4500	CF

28. LOGICAL OPERATION USING 8051

Page No.

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

Apparatus Required:
1. 8051 Micro controller kit — 1
2. Power Card — 1
3. Key board — 1
4. Keil and PC.

SETTING OF BITS :-
ALGORITHM :-
1. Load the Data in A-register.
2. Load OF 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.

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 content of accumulator and B-reg.
4107	90,11,00		MOV DPTR,#4500	Initialize DPTR with address 4500H
410A	F0		MOVX@DPTR,A	Store the Result in 4500H
410B	80,FE	STOP:	SJMP STOP	Stop the program

OUTPUT:-

Input		output	
Register	Data	Address	Data
4104	4D	4500	0D

Result:-

Thus, an assembly language program for setting and masking of 8-bit number using 8051 were performed and its outputs were verified.