

* Program To ADD Two 8-bit DATA:

memory address	Label	Instruction	opcode	Comments
8000		MVI C, 00	0F, 00	Get 1 st data in A and Save in B.
8002		LDA H200H	3A 00 42	↳ clear C reg to Count Carry
8005		MOV B,A	A7	
8006		LDA H201 H	3A 01 42	Get 2 nd data in A-reg
8009		ADD B	80	Get the Sum in A-reg
800A		TNC SKIP	D2 0E 80	If Cr=0 Then Skip next step
800D		INRC	0C	Increment C register to Count the Carry
800E	SKIP	STA H202 H	32 08 42	Store the Sum in memory
8011		MOV A,C	79	Move the Carry to accumulator and store in memory.
8012		STA H203 H	32 03 42	
8015		HLT	76.	Stop the Execution.

Input			
Address	DATA	Address	DATA
H200	F5	H202	BB
H201	C6	H203	01

1. 8-Bit ADDITION WITH CARRY USING DIRECT ADDRESSING

ADDRESSING*

* AIM :- To write an assembly language program to add two numbers of 8-bit data stored in memory location H200H and H201H and store the result in H202H and H203H with carry using direct addressing.

APPARATUS REQUIRED:-

- * 8085 Microprocessor Kit - 1
- * Power Card - 1
- * Key board - 1
- * 8085 Simulator card a pic.

* ALGORITHM:

- 1) clear Creg to Count Carry
- 2) Load the first data from memory
- 3) Increment C register to Count the Carry
- 4) Load the Second data from memory
- 5) Add the Content B - register to accumulator
- 6) If Carry flag = 0 then jump to step 7
- 7) Increment C register to Count the Carry
- 8) Store the sum in memory
- 9) Move the Carry to accumulator and store in memory.
- 10) STOP.

* RESULT:- Thus, an assembly language program 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:-
4000		MVI B, 00	06 00	Clear B reg to Count Carry
4002		LHLD H200 H	2A 00 42	Get 1st data in HL Pair and Save in DE
4005		XCHG	EB	
4006		CHLD H202 H	2A 00 42	Get 2nd data in HL - reg.
4009		DADD	19	ADD HL with DE and Get the Sum in HL register pair.
400A		JNC SKIP	D2 0E 40	If Cr=0 then skip next step.
400D		INR B	04	Increment C reg to Count the Carry.
400E	SKIP	SHLD H204 H	22 04 42	Store the Sum in memory
4011		Mov A,B	78	move the carry to accumulator and Store in memory
4012		STA H206 H	32 06 42	
4015		HLT	76	STOP the Execution.

Input		Output	
Address	DATA	Address	DATA
H200	BB, C2, F3	H204	9DFE BB 80 00
H202	DA, F1	H206	01 00

* 2w 16-bit ADDITION with CARRY *

* Aim :- To write an assembly language Program to two numbers of 16-bit data stored in memory locations H200H, H201H, H202H & H203H and store the result in H204H, H205H & H206H with Carry.

* APPARATUS REQUIRED :- * 8085 MicroProcessor Kit +

* Power Cord - 1

* Key board - 1

* 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 HL to DE Pair.
3) Load the Second data from memory to DE Pair.

4) Add the Count of BC Pair to HL Pair

5) If Carry Flag = 0 then jump to step.

6) Increment C register to Count the

7) Store the Sum in memory

8) Move the Carry to accumulator and in memory.

9) STOP

* RESULT :- Thus, an assembly language Program addition of given two 16-bit number with 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
4000		LDA 4261 H	3A 01 42	Get 3rd data in A and Save in B
4003		Mov B,A	47	
4004		LDA 4000 H	3A 00 42	Get 1st data in A
4007		SUB B	40	Subtract B-Reg from A-Reg
4008	JNC SKIP		D2 0E 40	If C-reg > 0 then skip next two
400B	WRC		0C	Increment C-reg by one
400C	CMA		2F	Take two's Complement
400D	INR A		3C	
400E	SKP	STA 4202 H	32 02 42	STORE the difference in memory
400F	Mov A,C		79	move the borrow to Accumulator and Store
4010		STA 4003 H	32 03 42	
4015	HLT		76.	STOP the Execution
<u>Input</u>		<u>Output</u>		
Address	DATA	Address	DATA	
H200	D5	H202	0F	
H201	C6	H203	00	

DIRECT ADDRESSING

Ques :- To write an assembly language program to Subtract two numbers of 8-bit data stored in memory locations 800H and 801H and store the result in 802H and 803H with borrow using direct addressing.

APPARATUS REQUIRED :- * Z80 MP Kit - 1
* Power Cord - 1
* Key board - 1
* Z80 Simulator and a PC.

ALGORITHM :-
1) Load the second data from memory to accumulator and move it to a 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 6.
5) Increment register to clear 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.

	<u>Input</u>		<u>Output</u>
Address	DATA	Address	DATA
H200	D5	H202	OF
H201	C6	H203	00

* Program To Multiply Two Number of 8-Bit DATA.

Memory Address	Label	Instruction	Op-code	Comments
4000	LDA	H200H	3A 00	Get 1st data in A
4003	MUL	E.A	5F	Save 1st data in E
4004	LOA	H201H	3A 42	Get 2nd data in A
4007	MUL	B.A	47	Save 2nd data in B
4008	LDI	H10000H	21 00 00	Load HL = 00000000H
400B	MVI	D,00H	16 00	Clear C for avoiding overflow.
400D	NEXT :	DAD	0	Add the content of DE to sum (HL).
400E	DCR	B	05	Decrement data for every addition
400F	JNZ	NEXT	C2 0D	Repeat addition until count is zero.
4012	SHLD	H20002H	82 02	Store the product in memory
4015	HLT		76	Stop the execution.

Address	Input DATA	Address	Output DATA
H200	A2	H202	AA
H201	C5	H203	7C

~~Result:~~ Thus in an assembly language program to multiply two number of 8-bit data was written, executed verified & Result successfully using 8085 kit.

- * Aim :- To write an assembly language to multiply two numbers of 8-bit data stored in memory H200H and 0201H and store the product in 0202H and 0203H.
- * APPARATUS Required :
 - * 8085 microprocessor kit ----- 1.
 - * Power Cord ----- 1
 - * Keyboard ----- 1
 - * 8085 Simulator and a PC.
- * ALGORITHM :
 - * Load the first data in HL and move to E.
 - * Load the second data in ECX and move to B (low)
 - * Clean D for overflow (Carry).
 - * Add the content of DE to HL.
 - * Decrement the count.
 - * Check whether count has reached zero.
 - * If ZF=0, repeat addition (or) if ZF=1, got to next step.
 - * Store the content of HL in memory.
 - * Clear Significant 16 bit of the prod.
 - * STOP.

Ex: 8-bit division operation using 8085 microProcessor

PROGRAM To DIVIDE Two Numbers of 8-Bit DATA

Memory Address	Label	Instruction	OpCode	Comments
4000		LDA 4201H	3A 01 42	
4003		Mov B, A	47	; Get the divisor in B-reg
4004		LDA 4200H	3A 00 42	; Get the dividend in A-reg
4007		MVI C, 00H	OE 00	; Clear C register for quotient.
4009	AGAIN:	CMP B	BB	
400A		JTC STORE	DA 12	; If divisor is less than dividend go to store.
400D		SUB B	90	; Subtract divisor from dividend increment
400E		INR C	0C	; Quotient by one for each subtraction
400F		JMP AGAIN	C3 09 40	
4012	STORE :	STA 4203H	32 03 42	; STA the remainder in memory
4015		Mov A, C	79	
4016		STA 4202H	32 02 42	; STA the quotient in memory.
4019		HLT	76	STOP the Execution.

Address	Input DATA	Address	Output data
4200	A4	4202	02
4201	40	4203	24

* Aim :- To write an ALP to perform division of two 8 bit numbers stored in memory (Location 4000H and 4001H) and store the remainder in 4003H and the quotient in 4002H.

APPARATUS Required :-

- 1. 8085 microProcessor Kit — 1
- 2. Power Cord 1
- 3. Keyboard — 1
- 4. 8085 Simulator and a PC.

ALGORITHM:-

- * Load the divisor in accumulator and move it 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 otherwise go to next step.

- * Subtract the content of B-register (quotient) from dividend.
- * Increment the content of C-register (quotient).
- * Go to step 4.
- * Store the content of C-register (remainder) in memory.
- * Move the content of C-register (quotient) to accumulator and store in memory.
- * 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



Program:-

memory Address	label	Instruction	Opcode	Comments
4000		LDA 4200H	3A 42 00	; Load the Count value
4003		Mov B,A	47	; Set Counter for (N-1) repetitions
4004		DCR B	05	; of (N-1) Comparisons
4005	Loop2	LXI H,4200H	21 42 00	; set Pointer for array
4008		Mov C,m	4E	; set count for (N-1) comparisons.
4009		DCR C	0D	
400A		INX H	23 05	; Increment Pointer.
400B	Loop1	Mov A,m	7E	; Get one data of memory in A
400C		INX H	2B	
400D		Cmp m	BE	; Compare next data with A register.
400E		Jc AHEAD	0A 40 16	; If Content of A is less than memory then go to AHEAD.
4011		Mov D,m	50	; If the Content of A is greater than
4012		Mov M,A	77 04 00	; Then Content of memory pointed by HL and Previous locations
4013		DCX H	2B	
4014		Mov M,D	72	
4015		INX H	23	
4016	AHEAD	DCR C	0D	; Repeat comparisons until C count is zero.
4017		JNZ Loop1	C2 40 0B	
4018		DCR B	05	; Repeat until B count is zero
4019		JNZ Loop2	C2	
401E		HLT	05 40	STOP the Execution.

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 from 4200H. The first element of the array gives the Count value for the number of elements in the array.

Apparatus Required:

- * 8085 microProcessor kit.
- * Power Card 1
- * Keyboard 1
- * 8085 Simulator and a PC.

Algorithm:

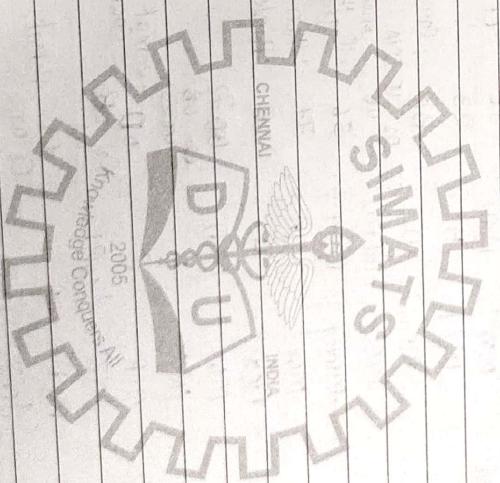
- * Load the Count value form memory to A-register.
- * Decrement B-register (B is a Count for (N-1) repetition).
- * Set
- HL Pair as data address pointer.
- * Set C-register as Counter for (N-1) Comparisons.
- * Load a data Pointed by HL wait accumulator.
- * Increment the HL Pair (data address pointer).
- * Compare the data Pointed by HL wait accumulator.
- * If Carry flag is set (if the Content of accumulator is Smaller than memory) then go to Step 6.
- * Otherwise Go to next Step.
- * Exchange the Content of memory Pointed by HL and accumulator.
- * Decrement C-register, if Zero flag is reset go to next Step.
- * Decrement B-register, if Zero flag is reset the Step 3 otherwise go to next step.



* STOP

RESULT :- Thus, an assembly language program for sorting in ascending order of an unsorted array of given & user written, executed and verified the result successful using Z80S kit.

Address	DATA Array (Before sorting)	Address	DATA Array (After sorting)
4200	07	4200	07
4201	AB	4201	34
4202	92	4202	4F
4203	84	4203	69
4204	4F	4204	84
4205	69	4205	92
4206	F2	4206	AB
4207	34	4207	C3
			01
			67



Program To Sort an array of DATA in Descending Order

Page No.

Memory Label	Label	Instruction	OpCode	Comments
4000		LOD	4200H	Load the count value
4003		MOV	B,A	set Counter for (n-1) repetition
4004		DCR	B	: of (n-1) comparisons
4005	Loop2	LXI	H,4200H	Set Pointer for array
4008		MOV	C,M	Set Count for (n-1) comparisons
4009		DCR	C	
400A		INX	H	Increment Pointer
400B	Loop1	MOV	A,M	Get one data of array in A.
400C		INX	H	
400D		CMP	M	
400E		JNC	AHEAD	Compare next data with A
4011		MOV	D,M	D register
4012		MOV	M,A	if content of A is less than memory then go to AHEAD.
4013		DCX	H	56 : if the content of A is greater than
4014		MOV	M,D	77 : then Content of memory
4015		INX	H	2B : Pointed by M Previous location.
4016	AHEAD	DCR	C	00 72 : Repeat comparisons until C
4017		JNZ	Loop1	00 0B : count is zero.
401A		DCR	B	00 0B : Repeat until B count is zero
401B		JNZ	Loop2	00 05 : Stop the execution?
401E		HLT		

Sample DATA:-

DATA	Array	Address	DATA	Array
07	94	4200	07	94
FB	AC	4201	F2	E5
92	BD	4202	AB	C2
84	87	4203	92	BD
4F	E5	4204	84	AC
69	69	4205	69	87
F2	05	4206	4F	69
34	C2	4207	34	45

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 sorted in memory stored in memory stored from 4200H. The first element of the array gives the count value for the number of elements in the array.

Apparatus Required :-

- * 8085 MicroProcessor Kit
- * Power Card 1
- * Hex board 1
- * 8085 Simulator and a PC.

Algorithm :-

The algorithm is same as algorithm of example Program except Step 8.

Step 8 : If Carry flag is reset then go to step 10, otherwise go to next step.

Result :-

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



Scanned with OKEN Scanner

Q. Search the Smallest Number From An Array.

PROGRAM TO SEARCH DATA IN AN ARRAY.	Memory Address	Label	Instruction	OpCode	Comments
	4000		ORI 7 W00H		Assembly directive
	4000	LX1 H, W00H	21 00	Set Pointer for array	
	4003	Mov B, M	42 0B	Set Counter for numbers of elements	
	4004	INX H	23	Set Count for numbers of elements	
	4005	Mov A, M	7E	Decrement the Count	
	4006	DCR B	05	With Current, Smallest Data	
	4007	INX H	2F	ACLRTHM:-	
	4008	CMP M	BE	If Cy = 1, then Content of memory.	
	4009	JC AHEAD	0A 0D	In all registers CHAIN	
	400A	Mov A, M	7E	* Load the address of the first element of the array in all registers	
	400B			* Move to Count to B- register	
	400C			* Increment the Pointer	
	400D			* Get the first data in accumulator	
	400E			* Decrement the Count	
	400F			* Increment the Pointer	
	4010			* Compare the content of memory addressed by B- reg with that of accumulator	
	4011	STA 400H	32 00	* If carry = 1, go to step 10 or if carry = 0, go to step 9.	
	4012	HALT	43	* Move the content memory addressed by B to accumulator	
				* Decrement the Count	
				* Check for zero of the count. If 2F = 0, go to step 6, or if 2F = 1 go to next step.	
				* Store the smallest data in memory.	

Result:-

thus, an assembly language program for searching a smallest number from an array of given 8-bit numbers was written.

SAMPLE DATA

Address	Input Data	Address	Output Data
4200	65	4300	10 (smallest data in the array)
4201	42	4301	32
4202	39	4302	52
4203	10	4303	A1
4204	24	4304	20
4205	34	4305	44
4206	25	4306	6C

Page No.

executed and verified the result successfully using 8085
kit and Simulator.



Scanned with OKEN Scanner

Program:-

Memory Address	Label	Instruction	OpCode	Comments
8000		ORG 4000H	21 00	Assemble directive
8003	LXH H, 4000H	21 42	Set pointer for array	
8004	Mov B M	46	Set count for number	
8005	INX H	23		
8006	MCR A M	7E	Set 1st Element of array	
8007	DCR B	05	Decrement the Count	
8008	loop	23	Compare the Element of array with Current smallest data	
8009	CMP M	B5		
800A	D2 00	4F	If Cy=0 Go to @HEAD	
800B	TRC @HEAD	7E	If memory content of memory is largest than A then	
800C	Mov A M	80	If largest cy = 1 make memory as largest	
800D	AHEAD	05		
800E	OCR B	05		
800F	JNZ loop	02 80	Repeat Comparison until	
8010	STA A300H	32	Store the smallest data in	
8011	HLT	STOP	Stop the Execution	

9. Search the largest number from an array.

Page No.

Aim:-

Write an assembly language program to search the largest data in an array of 16 data stored in memory from 400H to (400H+N). The first element of the array gives the number of data in the array. Store the smallest data in 4300H.

APPARATUS REQUIRED:-

- * 8085 microprocessor kit
- * Power Card 1
- * Key board 1
- * 8085 Simulator and a PC.

Algorithm:-

- * Load the address of the first element of the array in HL register pair (Pointer).
- * Move the Count to B-Register
- * Increment the Pointer
- * Get the first data in accumulator.
- * Decrement the Count.
- * Increment the Pointer.
- * Compare the content of memory addressed by HL pair with that of accumulator
- * If carry = 1, go to step 10 or if carry=0, go to step 9.
- * Move the content memory addressed by to accumulator.
- * Decrement the Count.
- * Check for zero of the Count. If ZF=0, go to Step 6,
- * or if ZF=1 go to next Step.
- * Store the smallest data in memory
- * STOP.

Result:-

Thus, an assembly language program for searching a smallest number from an array of given 8-bit numbers was written.

4204 34

4205 25



PROGRAM:-

label	Instruction	memory Address	op Code	Comments
LDA 4200		4000	31 00 42	Get the Ascii data to A reg
SUI 30H		4003	06 30	Subtract 30H from the data
CPL 0AH		4005	FF 0A	Compare the result with 0A
JC STORE		4007	DA 0C 40	If Cr = 1 Store the result.
SUI 07H		400A	D6 07	else then Subtract 07H
STORE STA 4201		400C	32 01 42	Store the result
HLT		400E	7F	Stop the Program

SAMPLE DATA:-

ASCII INPUT	Hex Output:-	
4200	41	35

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

(10) ASCII CODE TO HEX CODE CONVERSION

Page No.

Aim:-

To write an assembly language program to convert an array of Ascii codes to corresponding binary (Hex value) in 8085 microprocessor.

APPARATUS Required:-

- * 8085 MP Kit - 0
- * Power Card - 1
- * Key board - - - 1
- * 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 0A

4) If Cr = 1 Go to step 6 If Cr=0, Go to next Step

5) STORE the Hex into 4202H.

6) STOP Executing the Program.

RESULT:-

Thus, an assembly program for converting Ascii to Hex given 8-bit number was written and verified the result successfully using 8085 kit.



MASKING:-

Address	Label	Mnemonics	OPCODE	Comments
4100		LDA 4200	3A 00 42	Load A-register with data
4103		ANL (OFH)	F6 OF	And the Content of A with OFH
4105		STA 4201	32 01 42	STORE the Result
4108		HLT	76.	STOP the Program.

Input		Output	
Address	DATA	Address	DATA
4200	EA	4201	OF

Address	OPCODE	Label	MNE MONICS	Comments
4100	3A 00 42		LDA 4200	load A-register with data
4103	F6 OF		ORI (OFH)	OR the Content of A with OFH
4105	32 01 42		STA 4201	STORE the Result
4108	76		HLT	STOP the Program

Input		Output	
Address	DATA	Address	DATA
4200	A9	4201	AF

11. MASKING and Setting of Lower nibble on Given DATA.

Aim: To write Execute an assembly language program for performing masking getting one's and Two's Complement of given data to 8-bit number using 8085 MP.

APPARATUS Required

- * 8085 MP - Kit - 1
- * Power Cord --- 1
- * Key board --- 1
- * 8085 Simulator and a PC.

MISSING OF BitsAlgorithm:-

- * Load the DATA in A-Register.
- * logically AND the Content of A with OFH.
- * STORE the result in memory location.
- * STOP the Program

SETTING OF Bits:-Algorithm:-

- * Load the Data in A-reg
- * logically ORI the Content of A with OFH.
- * STORE the result in memory location.
- * STOP the Program.

RESULT:-

Thus, an assembly language Program for performing logical masking and setting of bits was Executed Using 8085 kit.



* Program:- By using 8086 Kit.

Address	Opcode	Label	Mnemonics	Comments
4100	3A 00		LOD A 4200H	Load AL-reg with 1st data
4103	2F		CMA	NOT the Contents of AX.
4104	32 01		STA 4201	Store the one's Complement in memory location
4107	3C		INR A	Increment the Content of AX.
4108	32 02		STA 4202	Store the two's complement in memory
410B	76		HLT	STOP the Program.

* SAMPLE DATA :-

Address	DATA	Output	
		Address	DATA
4200 H	CD	4201 H	32
		4202 H	33

12. ONE'S AND TWO'S COMPLEMENT

Page No.

* Aim:- To write and execute an assembly language program for performing one's and two's complement of given 8-bit numbers using 8086 kit.

* APPARATUS:-

- * 8085 microprocessor kit - 1.
- * Power Card --- 1
- * Key board --- 1
- * A PC.

FLASHING OF BITS:-

ALGORITHM:-

- * Load the data in a register.
- * Logically not the Content of A.
- * STORE the one's Complement in memory
- * Increment the Content of A.
- * Store the two's Complement in memory location.
- * STOP the Program.

RESULT:-

- * Thus, an assembly language program for performing one's and two's complement of bits were Executed using 8085 bit.



PROGRAM:-

Address	LABEL	MNEMONICS	Opcode	Comment
1000		MOV CX, 0000H	B9 00 00	Initialize Counter
1003		MOV AX, [1200]	A1 12	Get the first data in BX
1006		MOV BX, [1202]	8B 02 1E	Get the second data in BX.
1007		ADD AX, BX	03 C0	Both the AX & BX add the content of
100B		JNC L1	CB	Check for Carry
100C		INC CX	73 01	Increment the CX if carry exits.
100F	L1	MOV [1206], CX	89 06 0F 12	STORE the Carry
1013		MOV [1204], AX	A3 04 12	STORE the sum
1016		HLT	F4	Stop the Program.

Out For Addition:-

Address	DATA	Address	DATA
1200	5623	1204	EE AB
1202	9878	1206	0000

(13). ADDITION OF 16 BIT NUMBERS WITH CARRY

Page No.

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

- * APPARATUS:- * 8086 MP Kit - - 1
- * Power cord - - 1
- * Key board - - 1
- * Emulator 8086 and PC.

- * ALGORITHM:- * Load the first data in AX-reg.
- * Load the first data in BX-reg.
- * Add the two data and get the sum.
- * C=0 then SKIP next step.
- * Increment CX Reg for Carry.
- * STORE the sum in memory location.
- * STORE the carry in memory location.
- * STOP the Program.

* 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.



Program

Address	Label	Mnemonic	Opcode	Comment
1000		MOV CX, 0004H	B9 00 00	Initialize Counter CX
1003		MOV AX, [1300]	A1 00 13	Get the first data AX
1006		MOV BX, [1302]	8B 1E 0L 13	Get the second in BX
100A		SUB AX, BX	2B C3	Subtract the contents of both the registers
100C		JNC SKIP	73 03	Check the Borrow
100D		INX CX	41	If Carry, Exit increment
100F		NEG AX	F7 08	Take two's complement of different
1011	SKIP	MOV [1306]CX	89 0F 06 13	STORE the Borrow
1015		MOV [1304], AX	A3 04 13	Store the difference
1018		HLT	F4	Stop the Program

Output

for subtraction

INPUT			
Address	DATA	Address	DATA
1200	54 67	1204	31 33
1202	84 34	1206	00 01

IV. Subtraction of 16 bit number with

Page No.

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

APPARATUS:

- * Power Grid --- 1
- * Key Board --- 1
- * Emulator 8086 and PC.

Algorithm:- * Load the second data from memory to accumulator and move it to B-reg.

* Load the first data from memory to CHENNAI accumulator INDIA

* Subtract the content of B-reg from accumulator.

* If Carry flag = 0 then jump to step 5 of 6.

* Increment C reg to count the borrow.

* Take two's complement of the difference

* STORE the difference in memory

* move the borrow to accumulate a Sotre in the memory

* STOP.

Result:-

Thus, an assembly language program for subtraction with borrow of given numbers was written and verified successfully using 8086 bit and Emulator 8086.



(15) Multiplication of 16 bit Number:

Address	Label	Mnemonic	Op code	Comments
1000		mov Ax[1200]	A1 00 12	Load Ax-reg with 1st data
1003		mov Bx[1202]	8B 1E 12	Load Bx-reg with 2nd data
1006		mULBx	F7 E8 8B	Multiply the Content of Ax with Bx-reg
100A		mov [1204]Ax	AB 04 12	Store the lower word
100C		mov [1206]Dx	89 16 0B 12	Store the higher word
100E		HLT	F4	Stop the Program

Simple DATA :-

Input			
Address	Data	Address	Data
1200	221A	1204	93EE
1202	4C3B	1206	0A87

Aim:- To write the code an assembly language Programs to multiply two 16-bit unsigned numbers in 8086 kit and Emulator 8086.

APPARATUS:-

- * 8086 MP - Kit - 1
- * Power Card - 1
- * Key board - 1
- * Emulator 8086 and PC.

ALGORITHM:- Load the multiplier from memory to Accumulator.

& Load the multiplicand from memory to BX

& multiply AX with BX.

& STORE the lower word in memory from DX.

& STORE the higher word in memory from DX.

& STOP.

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 Emulator 8086.

16 DIVISION OF 32 BIT numbers by 16 bit Number

Address	Label	MNEMONICS	OPCODE	Comments
1000		MOV DX, [1200H]	8B 16 00 12	Initialize DX- register with Lsb of Dividend
1004		MOV AX, [1202]	A0 02 12	load Ax- register with dividend
1007		MOV BX, [1204]	8B 1E 04 12	load BX- register with Divisor
100B		DIV CX	F7 F3	divide AX by BX register
100D		MOV [1206], AX	A3 06 12	Store the Remainder
1010		MOV [1209], DX	89 16 08 12	Store the Remainder
1014		HLT	F4	Stop the Program

Output :-

Input		Output	
Address	DATA	Address	DATA
1200	0098	1206	00A8
1202	3604	1208	0142
1204	8587		

AIM :- To write and execute an assembly language program to divide two 16-bit unsigned number in 8086 MP Kit Emulator 8086.

APPARATUS :- * 8086 MP Kit - I
* Power Cord - I
* Key board --- I
* Emulator --- and PC.

Algorithm :- * Load the Divisor from memory to Accumulator
* Load the Divisor from memory to BX Register
* Divide DX / AX by BX.
* STORE the Quotient in memory from AX
* Store the Remainder in memory from D2.
* STOP.

~~Result :-~~ thus, an assembly program for division of given 16-bit number was written and executed and verified the result successfully using.

Program:-

Address	OPCODE	Label	MNE MONICS	Comments.
1000	A1 00 12		move AX [1200]	Load AL-reg with 1st
1003	25 OF OF		AND AX, OF0FH	AND the Content of Ax with OF0FH
1006	A3 02 12		MOV [1202], AX	Store the Result
1009	F4		HLT	Stop the Program.

Output:-

Input		Output	
Address	DATA	Address	DATA
1200H	5476	1202H	0006

Program:-

Address	OPCODE	Label	MNEMONICS	Comments.
1000	A1 00 12		MOV AX, [1200]	Load AL - register with 1st data.
1003	0D OF OF		OR AX, OF0FH	AND the Content of Ax with OF0FH
1006	A3 02 12		MOV [1202], AX	STORE the Result
1009	F4		HLT	STOP the Program

Output:-

INPUT		OUTPUT	
Address	DATA	Address	DATA
1200H	2312	1202H	2F1F

17. Logical Operation

Page No.

Aim :- To write and execute an assembly language programming masking, setting and twin complement of given 16-bit number using 8086 MP Kit and Emulator 8086.

APPARATUS :-

- * 8086 MP - 1
- * Power Card -- - 1
- * Key board -- - 1
- * Emulator 8086 and PC.

masking of BITS :-

Algorithm :-

- * Load → the DATA in Ax - register.
- * logically AND the Content of Ax with OF0FA.
- * STORE the result in memory location
- * STOP the Program.

Result :- Thus, an assembly language program for performing logical masking and setting of bits were executed using 8086 kit.



PROGRAM:-

Address	Opcode	Label	MNEMONICS	Comments
1000	C7 C6 0011		MOV SI, 1150H	Initialize the source address
1003	C7 C7 0013		MOV DI, 1250H	Initialize the destination address
1006	C7 C1 0600		MOV CX, 0006H	Initialize Count reg value to the
1009	FC	REPEAT	CLD	clear the direction flag
100A	A4		MOV SB	Move the string byte
100B	E2 F3		LOOP REPEAT	on Conditional loop to address
100C	F4		HIT	STOP the Program

* Output:-

INPUT		OUTPUT	
Address	DATA	Address	DATA
1150	S2	1250	S2
1151	S3	1251	S3
1152	S4	1252	S4
1153	S5	1253	S5
1154	S6	1254	S6

⑧ Move # DATA Block without Overlap.

Page No.

Aim :- To write and execute and assembly language program for transferring data from one block to another block without overlap using 8086 kit and Emulator 8086.

APPARATUS :-

- * 8086 microProcessor kit
- * Power Card
- * Key board.
- * Emulator.

Algorithm:

- * Initialize Counter
- * Initialize Source block Pointer
- * Initialize destination block pointer.
- * Get the byte from Source blo
- * Increment Source, destination Point and element Counter.
- Repeat steps 4,5 and 6 until
- Equal to zero.

~~Result :-~~ Thus
transferring
another block
using 8086 kit.

an assembly language program
data from one block
without overlapping way



PROGRAM:-

Address	Label	MNEMONICS	OPCODE	Comments
1000		Mov DX, 0H.	BA 00 00	Clear DX
1003		Mov SI, 1250 H	BE 50 12	Initialize the Source address
1006		Mov CX, 03H	B9 03 00	Initialize the Count value
1009		Mov AX, [SI]	8B 04	Loop AX-reg
100B	A1:	INC SI	46	Increment
100C		INC SI	46	Addition
100D		ADD SX, [SI]	03 04	Addition of AX
100F		JNC NEXT	73 01	JMP 8700 carry Next
1012		INC DX	42	Increment DX
1014	NEXT:-	Loop A1	E2 F1	
1017		Mov [1200H], AX	A3 00	move to AX to address
1018		Mov [1302H], DX	89 16 02 13	load and execute
101B		HLT	F4	Stop Program.

Output:-

INPUT:-		OUTPUT	
Address	DATA	Address	DATA
1250	65 87	1300	7565
1252	03 65	1302	01
1254	45 83		
1256	87 56		

Q. Sum of N Number 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:-

- * 8086 MP Kit - - - 1
- * Power Cord - - - 1
- * Key Board - - - 1
- * Emulator 8086 and PC.

Algorithm:-

- * Initialize Counter.
- * Initialize Source block Pointer.
- * Initialize destination block Pointer.
- * Get the byte from Source block
- * Store the byte in the destination block
- * Increment Source destination Pointers and decrement Counter.
- * Repeat the steps 4,5 and 6 until - loop and decrement Counter.
- * Stop.

Result:- Thus, assembly language Program for transfering data from one block to another block with out overlapping was Executed using 8086 kit.



PROGRAM:-

Address	LABEL	MNEMONICS	OPCODE	Comments
4100	START	(A) H, 4200		Initialize HL with 4000
4103		MUL, 04		Copy the value 04 to C-reg
4105	NEXT	MOV A, M		Copy the Content M to A-reg
4106		OUT C0		- the Content of A is moved to port 04
4108		LXI D, 1010		Copy the data 1010 to DE-reg
410B	LOOP	DCX D		Decrement DE-reg
410C		MOV A, E		
410D		ORA D		Check out DE = 0000
410E		JNZ LOOP		JMP on no zero to loop
4111		INX H		Increment HL reg pair
4112		DCR C		Decrement the count
4113		JNZ NEXT		Jump to next
4115		JNC 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 Clock wise direction

Input		Output	
Address	Data	Address	Data
4200			
4200			

20. STEPPER MOTOR INTERFACING.

Page No.

Objetive :- To write and Execute an assembly language Program to run a Stepper motor at different speed and to Control direction using 8085 microprocessor.

APPARATUS :-

- * 8085 microprocessor kit - 1
- * Stepper motor --- 1
- * Stepper motor - Interface board - 1.
- * Power board - 1.
- * Key board - 1.

Result :- Thus an assembly Program to run the stepper motor in both forward and reverse direction with delay was Executed and its output was verified.



PROGRAM:-

Address	Label	Program	Opcode	Comments
1100		MOV SI, 1200H		Initialize array
1104		MOV CX, 000FH		Initialize array size
1108		MOV AL, 10		Store the control word
110B		OUT C2, AL		Send through output port.
110D		MOV AL, ACC		Store the control word to clear display
1110		OUT C2, AL		Send through output port.
1112		MOV AL, 0CC		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, 0F FF FH		Store 16 bit Count value.
111F	loop1	DEC DX		Decrement Count value
1120		JNZ loop1		Loop unit became Count value
1122		INC SI		Get & get Next data.
1123		loop NEXT		Loop until all the data has taken.
1125		JMP START		Go to starting location
1127		HLT		

Input

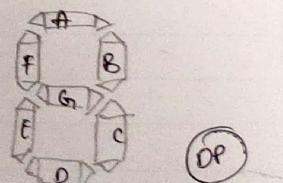
Address	DATA
1111	

Output

Address	DATA
Display	

* Look-up table :-

1200	98	68	B7C	C8
1204	FF	1C	29	FF



21. Keyboard HND Disp FN.

Page No.

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

APPARATUS:-

- * 8086 microprocessor kit --- 1.
- * 8279 interface board --- 1.
- * Power Card
- * Keyboard --- 1.

ROLLING MESSAGE "HELP US"

Algorithm:-

- * Display of rolling message "Help us".
- * Initialize the Counter.
- * Set 8279 for 8 digit Character display, right entry.
- * Set 8279 for 9 clearing the display.
- * write the command to display.
- * Load the character in accumulator and display.
- * Introduce the delay.
- * Repeat from step 1.

Display the "3" character:-

Algorithm:-

- * Set 8279 for 8-digit character display, right.
- * Set 8279 for clearing the display.
- * write the command to display.
- * Load the character into accumulator and display it.
- * Repeat from Step 1.



Memory location	Message	7-segment LED Format							HXF CODE	
		D	C	B	A	D _P	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	1	FF

Program:-

Address	Label	Program	OPCODE	Comments
1100		mov AL,00	C6,0000	Store the control word for display
1103		OUT C ₂ ,AL	E6,C ₂	Send through oct port.
1105		MOV AL,0CC	C6 CC	Store the control word to clear display.
1108		OUT C ₁ ,AL	E6 C ₂	Send through oct port now
110A		MOV AL,90	C6 C ₀ 90	Store the connected to display.
110D		OUT C ₂ ,AL	E6 C ₂	Send oct port
110F		MOV AL,8F	C6 C ₀ 8F	Get the first data.
1112		OUT C ₀ ,AL	E6 C ₀	Send through port
1114		HLT	F4	Stop the Execution.

Result:- Thus, the rolling message "Help us" and the character "3" are displayed using 8279 interface kit with 8086 microprocessor.

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
1105		IN AL, C0		Read Port A
1108		MOV [1250], AL		Store the result in memory
1114		HLT	F4	Stop the program.

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

Interface Switches with 8086 through 8255 Page No.

Aim:- To write and execute an assembly language program to interface 8 switches with 8086 microprocessor through 8255 PPI.

APPARATUS:-

- * 8086 microprocessor kit - - - 1
- * 8255 Interface Card - - - 1
- * Power Card - - - 1
- * Key board - - - 1

Algorithm:-

- * Configure the 8255 Port A as Input Port with the Control reg value as "90H".
- * Read the Port A switch status through
- * Store the output in t10.
- * Stop.

Result:-

thus, an assembly language Program for Interfacing of Switches with 8086 through 8255 PPI was written created and verified the Result Successfully.



* Program :-

* ADDITION

<u>Address</u>	<u>opcode</u>	<u>Label</u>	<u>Program</u>	<u>Comments</u>
4100	74,05		MOV A, #03	Load data 1 in accumulator.
4102	75, F0, 10F		MOV B, #05	Load data 2 in B register.
4105	35, F0		ADDC , A B	Add the contents of accumulator B
4107	90, 11, 00		MOV DPTR #4500H	Initialise DPTR with address 4500H
410A	F0		MOVC @DPTR, A	Store the sum in 4500H
410B	80, FE	STOP:	SJMP STOP	STOP → the Program

Input

Register	DATA	out Put	Address	DATA
4101	03		4500	08
4104	05			

24. Addition Operation using 8051 MicroProcessor

Aim :- To write and execute an assembly language program to add of two 8-bit numbers using 8051 and KELI software.

APPARATUS :-

- * 8051 microcontroller kit ... 1
- * Power Card ... 1
- * Key board ... 1
- * KELI and PC

ALGORITHM :-

- * Load the first data in A - register.
- * Load the second data in B - register.
- * Add the two data with carry.
- * Store the sum in memory location.
- * STOP → the Program.

Result :-

Thus, an assembly language program for given two 8-bit was written and verified using 8051 Kit.

28. Logical Operations using 8051.

PROGRAM:-

Address	OP Code	Label	Program	Comments
H100	7U, 05		MOV A #08	load data 1 in accumulator
H102	45, F0, 05		MOV B #0F	load data 2 B-reg
H105	35, F0		ORL A,B	OR the contents of accumulator
H107	90, 11, 00		MOV DPTR #US00H	Initialize DPTR with address
H10A	F0		MOVX @DPTR, A	STORE the result US00H
H10B	80, FE	STOP:	SJMP STOP	STOP - the Program.

OUTPUT:-

Input		Output	
Register	DATA	Address	Data
4101	08	4500,	0F

Aim:-

To write and execute an assembly language for setting and masking of given 8-bit number using 8051.

Apparatus Required:-

- * 8051 microcontroller kit --- 1.
- * Power Cord --- 1
- * Key board --- 1
- * Keil and PC.

Setting of Bits:-

- Algorithm:-
- * Load the Data in A-register
 - * Load DF to set the lower nibble in B-register.
 - * Perform OR operation with B-register.
 - * STORE the Result in memory location
 - * STOP the Program.

masking of Bits

- Algorithm:-
- * Load the Data in A-register
 - * Load DF to mask the higher nibble in B register
 - * Perform AND operation with B-register.
 - * STORE the Result in memory location.
 - * STOP the Program.

Result:-

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

PROGRAM:-

Address	opcode	Label	Program	Comments
4100	74, 05		MOV A, #93	load data in the accumulator
4102	75, F0, 05		MOV B, #0F	load data 2 in B-reg
4105	35, F0		ANL A, B	AND the content of accumulator
4107	90, 11, 00		MOV DPTR # 4500H	Initialize DPTR with address 4500H
410A	F0		MOV X @DPTR, A	store the result in 4500H
410B	80, FE	STOP	SJMP STOP	stop the program

Output:-

Register	Input		Output
Address	DATA	Address	DATA
4101	93	4500	03