Topic :

PIONEER

Page :

Date : / /

Assignment No. - 1

GENERATE FIRST 10 FIBONACCI NUMBER

Topic :

PIONEER®

Page :

Date : / /

- OBJECTIVE: Display First 10 Fibonacci Number

- PROBLEM STATEMENT :- Write an Assembly Language Program in 8085 µp to Find First 10 Fibonacci Numbers.

- PROBLEM ANALYSIS : The first 2 bytes of fibonacci sequence are stored from memory location D000H The sequence is stored from memory location D003H and onwards. Note that, this algorithm and program generates fibonacci series in hexadecimal numbers.
  D000 = 00 D001 = 01 D002 = 01 D003 = 02 D004 = 03...

- ALGORITHM : The formal algorithm steps of the given problem are as follows.

Step 1 : The first two bytes of fibonacci series are stored at memory location D000H and D001H respectively. The memory location is pointed by H-L register pair.

Step 2 : Initialize register C with 08H

Step 3: Move the content of memory to accumulator

Step 4: Increment the H-L register pair to the next memory location.

Step 5: The content of memory is placed in register B.

Step 6: Add the content of register B with the content of accumulator.

Step 7: Again increment the H-L register pair to the next memory location and store the content of accumulator to that location.

Step 8: Move content of register B to accumulator.

Step 9: Move the content of memory location (pointed by H-L register pair) to register B.

Step 10: Decrement registe C by one.

Step 11: Check register's (C) content is zero or not

If it is not zero go to step 6 otherwise go to step 12

Step 12 : Terminate the execution.

TABLE :-

Memory Location	Lebel	Mnemonics	Hex Code	Comment
C000		LXI H, D000H	21	HL register pair
C001			00	pointed to memory
C002			D0	location D000H.
C003		MVI C, 08 H	0E	Initialize register
C004			08	C with 08H.
C005		MOV A, M	7E	Move the content of memory to acc.
C006		INX H	23	Increment H-L pair
C007		MOV B, M	46	Move memory content to B
C008	LOOP	ADD B	80	Add reg. B with accumulator
C009		INX H	23	Increment H-L pair
C00A		MOV M, A	77	Store accumulator content to memory
C00B		MOV A, B	78	Register B content is shifted to acc.
C00C		MOV B, M	46	Memory content shifts to reg. B

Memory Location	Lebel	Mnemonics	Hex Code	Comments
C00D		DCR C	0D	Decrement reg. C
C00E		JNZ C008H (LOOP)	C2	Jump to LOOP
C00F			08	Level when reg. C
C010			C0	is not zero.
C011		HLT	76	Terminate the program.

• INPUT AND OUTPUT :-

Input : D000 : 00 D001 : 01

Output :

Memory Location	D000	D001	D002	D003	D004	D005	D006	D007
Data	00	01	01	02	03	05	08	0d
Memory Location	D008	D009						
Data	15	22						

• DISCUSSION :-

i) The 1 byte instructions are used in this program like MOV, INX, ADD, DCR and HLT

ii) 2 byte instructions are used like MVI and 3 byte instructions LXI and JNZ are used in this program.

iii) Total 18 byte instructions are used in this program.

Topic :

PIONEER®

Page :

Date : / /

Assignment No: 2

ODD EVEN BYTE COUNTING

Topic :

PIONEER®

Page :

Date : / /

OBJECTIVE : Count the Number of Odd & Even Bytes from A Set of N Bytes Number

› PROBLEM STATEMENT : Write an Assembly Level Program to count the no of odd and even bytes in a set of N bytes.

› PROBLEM ANALYSIS : A set of 8 bit numbers are stored from location D001H. The number (N) of 8 bit numbers are stored at location D000H. Count no. of odd bytes and even bytes number and store it location D010H and D011H respectively.

› ALGORITHM : The formal algorithm steps of the given problem are as follows —

Step 1 : Load H-L register pair with D000H location address so that it works as a memory pointer.

Topic :

PIONEER

Page :

Date : / /

Step 2 : Move the contents of memory to C register. It acts as a counter and specifies how many bytes are stored from location D001H

Step 3 : Register B and D are initialized by zero. These are used to count the no. of even and odd byte respectively.

Step 4 : Increment H-L register pair by one moves its content to accumulator.

Step 5 : Logical AND operation with 01H is perform -ed with the content of accumulator.

Step 6 : If the result of AND operation is zero; then the control of execution jumps to EVEN level and increment register B by one. Otherwise register D is incremented by one.

Step 7 : After that, control of execution jumps to NEXT_OP level and decrement register C by one. Check register C is zero or not. If it is

go to LOOP level and perform the above steps again for next byte.

Step 8 : The no. of even byte which is stored at B register moves to accumulator and is stored at location D010H and no. of odd byte which is stored at location D register. and is stored at D011H location.

Step 9 : Stop executing the program and halts any further execution.

TABLE :

Memory Location	Lebel	Mnemonics	Hex Code	Comment
C000		LXI H, D000H	21	HL register pair
C001			00	pointed to memory
C002			D0	location D000H
C003		MOV C, M	4E	Move register C with no. of bytes
C004		MVI B, 00H	06	Move 00H to
C005			00	register B

Topic :

PIONEER®

Page :

Date : / /

Memory location	Lebel	Mnemonics	Hex Code	Comment
C006	D,	MOV D, B	50	Move the contents of B to D
C007	LOOP	INX H	23	Increment H-L pointer register to next location.
C008		MOV A, M	7E	Move the contents of memory to accumulator
C009		ANI 01_H	E6	Logical AND operation between accumulator and 01_H
C00A			01	
C00B		JZ $C012_H$	CA	Jumps to EVEN level if result is zero
C00C		(EVEN)	12	
C00D			C0	
C00E		INR D	14	Increment D by 1 if result is non zero
C00F		JMP $C013_H$	C3	Jump to location $C013_H$
C010		(NEXT_OP)	13	
C011			C0	
C012	EVEN	INR B	04	Increment B by one if result is zero
C013	NEXT_OP	DCR C	0D	Decrement register C by one
C014		JNZ C007	C2	Check register C is zero or not. If not zero go to LOOP level
C015		(LOOP)	07	
C016			C0	
C017		MOV A, B	78	Move the content of B to accumulator.

Topic :

PIONEER®

Page :

Date : / /

Memory location	Label	Mnemonics	Hex Code	Comment.
C018		STA D010H	32	Store no. of even bytes is stored at location D010H.
C019			10	
C01A			D0	
C01B		MOV A, D	7A	The content of D moves to A
C01C		STA D011H	32	Store no of odd bytes is stored at location D011H.
C01D			11	
C01E			D0	
C01F		HLT	76	

INPUT AND OUTPUT :

INPUT	OUTPUT
D000 : 06 (Counter)	D010 : 04 (Even)
D001 : 02	D011 : 02 (odd)
D002 : 05	
D003 : 08	
D004 : 0A	
D005 : 06	
D006 : 03	

Topic :

PIONEER®

Page :

Date : / /

• DISCUSSION :-

i) Properly connect the 8085 microprocessor kit with power supply terminals.

ii) Switch on the power supply after checking connections.

iii) 3 byte instructions - LXI, STA, JNZ, JZ, JMP are used

iv) 2 byte instructions — MVI, ANI are used in this program.

v) 1 byte instructions — DCR, MOV, INX, HLT are used here.

vi) Total 32 bytes instructions are used in this program.

Topic :

PIONEER®

Page :

Date : / /

OBJECTIVE : 8bit number and 8-bit number Multiplication.

- **PROBLEM STATEMENT :** Write an Assembly Language Program to multiply two 8bit numbers.
- **PROBLEM ANALYSIS :** The input is given in D000H and D001H and the input are given into H-L register pair. Then this two numbers are multiply using repeatative addition method. The result is stored in location D008H and D009H respectively (carry)
- **ALGORITHM :** The formal algorithm steps of two 8-bit number multiply in 8085 MPU are as follows —.

Step 1 : Load H-L register pair in the given address location.

Step 2 : Move the data content of memory to a register B.

Step 3 : Increment the memory pointer H-L

Step 4 : The content of accumulator is going to be zero and move the content of accumulator to register C.

Step 5 : Add the content of memory with the content of accumulator.

Step 6 : If carry occures then go to step 7 otherwise go to step 8

Step 7 : Increement the content of C register.

Step 8 : Decrement the content of B register.

Step 9 : Go to step 5 and repeat it until register B becomes zero.

Step 10 : Point H-L register pair to location $D008_H$ and store the content of register C (carry bit) to that memory location.

Step 11 : Increment the register pair H-L and store the accumulator content to memory location.

Step 12 : Stop executing the program and halts any further execution.

TABLE : The table of two 8 bit multiplication in 8085 MPU is as follows —

Memory Location	Lebel	Mnemonics	Hex Code	Comment
C000		LXI H, D000H	21	Load first
C001			00	operand address
C002			D0	
C003		MOV B, M	46	Store first operand to B
C004		INX H	23	Increase HL pair
C005		XRA A	AF	Clear accumulator
C006		MOV C, A	4F	Store 00H at register C
C007	LOOP	ADD M	86	Add memory content with acc.
C008		JNC SKIP	D2	When Carry
C009		(C00A)	C0	flag is 0, skip

Topic :

PIONEER

Page :

Date : / /

Memory Location	Lebel	Mnemonics	Hex Code	Comment
C00A			C0	next task
C00B		INR C	0C	Increase C when carry is 1
C00C	SKIP	DCR B	05	Decrease register B
C00D		JNZ $C007_H$	C2	Jump to loop when
C00E		(LOOP)	07	Z flag is not 1.
C00F			C0	
C010		LXI H, $D008_H$	21	Load destination
C011			08	address
C012			D0	
C013		MOV M, C	71	Store C register content into memory
C014		INX H	23	Increase HL pair.
C015		MOV M, A	77	Store acc content to memory
C016		HLT	76	Terminate the program.

INPUT AND OUTPUT :

Set	Input		Output	
	D000	D001	D008(Carry)	D009
Set-1	07	03	00	15

Topic :

PIONEER®

Page :

Date : / /

Set	Input		Output	
	D000	D001	D008 (Carry)	D009.
Set-2	0A	0A	00	64
Set-3	50	06	01	E0
Set-4				
Set-5				

• DISCUSSION :-

a. The 3-bytes instruction which is used here are LXI, JNC, JNZ and STA

b. The 1 bytes instruction used here are MOV, XRA ADD, INR, DCR, HLT.

c. Total 15 instructions are used in the program and the total bytes are

d. The 8085 has no multiplication operation to get the result of multiplication, the repeatative addition method should be kept.

e. After multiplying two 8 bit number it may generate 1 byte or 2 byte number, so two registers are used to hold the result.

f. The HLT instruction the value of program counter. Since program counter is not getting updated and keeps its value as it is, fetching if any instruction is not possible. This will keep MPU at stedy position.

Topic :

PIONEER®

Page :

Date : / /

Assignment No. 4

SORTING IN ASCENDING & DESCENDING ORDER

Topic :

PIONEER®

Page :

Date : / /

OBJECTIVE : Sorting an Array in Ascending & Descending Order.

- **PROBLEM STATEMENT :** Write an Assembly Language Program to arrange an Array of 8 bit numbers in Ascending and Descending Order.

- **PROBLEM ANALYSIS :-** The 8 bit numbers are stored in an array starting at location D000H Here, we adapt 'Bubble Sort' technique to sort in ascending order. Then by 'Block Transfer Mechanism' we arrange an array in descending order. 8 bit numbers of array are stored in descending order at location D050H.

- **ALGORITHM :**

Step 1 :- Load H-L register pair with D000H location address so that it works as a memory pointer.

Step 2 : Moves the content of memory to C

and decrement the content of C by one.

Step 3 :- Initialize register D with 00H and increase H-L register pair by one again.

Step 4 : The content of memory is moved to accumulator and increase HL register pair by one so that the pointer points to next memory location.

Step 5 : Compare the content of memory location with content of accumulator. If borrow occures, that means content of accumulator is smaller than memory content, go to NEXT level

Step 6 : If borrow not occures, swapping operation is performed. Content of memory is moved to register B. Content of accumulator is moved to memory location. Then H-L pointer is decremented by one. In that location, register B content is stored.

Step 7 : Again increment HL pointer by one. Move

Topic :

PIONEER

Page :

Date : / /

01H to register D.

Step 8 : Decrement register C by one. If it is not zero go to LOOP level. (Step 1)

Step 9 : Move content of register D is moved to accumulator. Logical OR operation is performed with accumulator itself to check whether any swapping operation is performed or not. If no swapping operation performed, stop the execution, otherwise go to NEXT_PASS.

Step 10 : To arrange in descending order, load H-L register pair with location D000H. Move content of memory to C register and B register.

Step 11 : Increment HL pointer register to next location and decrement register C by one every time to send memory pointer to last location (nth). Check register C content is zero or not. If it is not zero perform the previous mentioned operation.

Step 11: Decrement register B by one and point DE register pair to D050H memory location.

Step 12: Move the content of memory location pointed by H-L register pair to accumulator.

Step 13: Store it to DE register pair, decrement register pair H-L by one and similarly increment DE register pair by one.

Step 14: Decrement register B by one and check whether it is zero or not. If it is not zero perform the previous operation and stop performing any further execution if HLT instruction encountered.

TABLE :- The table for sorting an array of 8 bit numbers is as follows :

Memory Location	Lebel	Mnemonics	Hex Code	Comments
C000	NEXT_PASS	LXI H, D000H	21	Load H-L register
C001			00	pair with D000H
C002			D0	memory location.
C003		MOV C, M	4E	Move content of memory to regis

Topic :

PIONEER

Page :

Date : / /

Memory Location	Lebel	Mnemonics	Hex Code	Comment
C004		DCR C	0D	Decrement C by one
C005		MVI D, 00H	16	Move 00H to regis
C006			00	ter D
C007		INX H	23	Increment H-L reg. pair by one.
C008	LOOP	MOV A, M	7E	Move memory to acc.
C009		INX H	23	Again increment by 1
C00A		CMP M	BE	Compare accumulator content with memory
C00B		JC C015H (NEXT)	DA	If borrow occures goto NEXT level other
C00C			15	wise perform the
C00D			C0	following operations.
C00E		MOV B, M	46	Move the content of memory to reg. B
C00F		MOV M, A	77	Move acc content to memory
C010		DCX H	2B	Decrement HL reg. pointer
C011		MOV M, B	70	Reg. B content moves to memory
C012		INX H	23	Increment HL reg. pair
C013		MVI D, 01H	16	Move 01 H to D register
C014			01	
C015	NEXT	DCR C	0D	Decrement C register by one.
C016		JNZ C008 (LOOP)	C2	Check whether it is zero or not; go to LOOP level
C017			08	
C018			C0	
C019		MOV A, D	7A	Move the content of D to accumulator

Topic :

PIONEER

Page :

Date : / /

Memory Location	Label	Mnemonics	Hex Code	Comments
C01A		ORA A	B7	Logical OR operation with acc. itself
C01B		JNZ C008H (NEXT_PASS)	C2	Check whether it is zero or not; if not go to NEXT_PASS level
C01C			00	
C01D			C0	
C01E		LXI H, D000H	21	Load HL register pair with D000H to arrange in descending order.
C01F			00	
C020			D0	
C021		MOV C, M	4E	Move memory to reg. C
C022		MOV B, M	46	Move memory to reg. B
C023	LOOP1	INX H	23	Increment HL register pair by one.
C024		DCR C	0D	Decrement reg. C by one
C025		JNZ LOOP1 (C023H)	C2	If it is not zero go to LOOP1 level. otherwise next instruction
C026			23	
C027			C0	
C028		DCR B	05	Decrement reg. B by one
C029		LXI D, D050H	11	Load D-L register pair with D050H location
C02A			50	
C02B			D0	
C02C	LOOP2	MOV A, M	7E	Move content of memory to accumulator.
C02D		STAX D	12	Store accumulator content to DE register pair
C02E		DCX H	2B	Decrement HL reg. pair
C02F		INX D	13	Increment DE reg. pair

Topic :

PIONEER®

Page :

Date : / /

Memory Location	Lebel	Mnemonics	Hex Code	Comments
C030		DCR B	05	Decrement reg. B by one
C031		JNZ C02CH (LOOP2)	C2	Check if it is zero then HLT instruction executed otherwise go to LOOP2 lerel
C032			2B	
C033			C0	
C034		HLT	76	Terminate the execution.

- INPUT AND OUTPUT :

Set-1

INPUT	OUTPUT (Ascending)	OUTPUT (Descending)
D000 : 05 (Counter)		
D001 : 20	D001 : 10	D050 : 50
D002 : 40	D002 : 20	D051 : 40
D003 : 50	D003 : 30	D052 : 30
D004 : 10	D004 : 40	D053 : 20
D005 : 30	D005 : 50	D054 : 10

Set-2

INPUT	OUTPUT (Ascending)	OUTPUT (Descending)
D000 : 05 (Counter)		
D001 : 02	D001 : 02	D050 : 0C

Topic :

PIONEER®

Page :

Date : / /

D002 : 0A	D002 : 04	D051 : 0A
D003 : 06	D003 : 06	D052 : 06
D004 : 04	D004 : 0A	D053 : 04
D005 : 0C	D005 : 0C	D054 : 02

• DISCUSSION :

i) LXI H, JZ, LXI D, JNZ, are 3 byte instructions used in the program.

ii) MOV B, M, DCR, INX, DCX, CMP, ORA, HLT are 1 byte instruction are used here.

iii) There are two 2 byte instructions, MVI D, 00H
MVI D, 01H

iv) Total 53 bytes instructions are used in this program.

v) We should be careful about using instructions load inputs and different types of logic and we should keep knowledge about pin diagram of 8085 microprocessor descriptions of pins, carry and other flags, register uses etc. to execute the program perfectly.

Topic :

PIONEER

Page :

Date : / /

• OBJECTIVE : Identification of Binary Palindrome

• PROBLEM STATEMENT :- Write an Assembly Language Program to Identify Whether a Binary Number Palindrome or Not.

• PROBLEM ANALYSIS : In this program we are taking the number from location B000H. The program will return 00H if the number is not palindrome otherwise it will return FFH. Let the input is 18H in binary value is (0001 1000) this is a palindrome The number 52H (01010010) it is not a palindrome.

ALGORITHM :

Step 1 : Load in accumulator from memory location B000H. Move accumulator to H register.

Step 2 : Initialize register C with 08 H. It acts as a counter. Move again H register to accumulator

Step 3 : Rotate left the content of the accumulator without carry. Move accumulator content to H register

Step 4 : Load D content to accumulator. Then rotate it right through carry. Move it to again D register.

Topic :

PIONEER®

Page :

Date : / /

Step 5: Decrement register C by one. Check if it is zero or not If it is not zero go to LOOP level.

Step 6: Load H register data to accumulator. and compare it with D register. If the result produces zero then jump to TRUE level

Step 7: Load 00H to accumulator and store it to D001H location if the result of compare is not zero. ~~and~~ otherwise load FFH to accumulator and store it to D001H location and terminate the program

TABLE :

Memoryloc	Lebel	Mnemonics	Hex code	Comments.
C000		LDA D000H	3A 00, D0	Load the number into A
C003		MOV H, A	67	Move Acc to H
C004		MVI C, 08H	0E, 08	Initialize Counter.
C006	LOOP	MOV A, H	7C	Load H to Acc.
C007		RLC	07	Rotate left
C008		MOV H, A	67	Get back Acc to H
C009		MOV A, D	7A	Load D content to acc.
C00A		RAR	1F	Rotate accumulator right through carry
C00B		MOV D, A	57	Get back Acc to D
C00C		DCR C	0D	Decrease C
C00D		JNZ C006 (LOOP)	C2, 06 C0	Jump to LOOP if Z = 0
C010		MOV A, H	76	Load H data to Acc
C011		CMP D	BA	Compare D with acc

Topic:

PIONEER

Page:

Date: / /

Memory Loc	Lebel	Mnemonics	Hex Code	Comments
C012		JZ C01A (TRUE)	CA	If both are same it is palindrome.
C015		MVI A, 00H	3E, 00	Load 00H in A
C017		JMP C01C (EXIT)	C3, 1C C0	Jump to Exit
C01A	TRUE	MVI A, FFH	3F, FF	Load FFH to A
C01C	EXIT	STA D001H	32, 01, D0	Store the result in memory
C01F		HLT	76	Terminate the program.

• INPUT & OUTPUT :-

Input	Output
D000 : 52 (Set-I)	D001 : 00
D000 : BD (Set-II)	D001 : FF
D000 : 18 (Set-III)	D001 : FF

• DISCUSSION :

i) The 3 byte instructions LDA, STA, JMP, JZ, JNZ are used in this program. ii) 2 byte instructions like MVI are used and 1 byte instructions RAL, RAR, HLT MOV are used in this program iii) Total 31 byte instructions are needed to execute this program.

iv)

Topic :

PIONEER

Page :

Date : / /

- **OBJECTIVE:** Binary to Gray Code Convert

- **PROBLEM STATEMENT:** Write an Assembly Language Program to convert a Binary Number into its Gray Code.

- **PROBLEM ANALYSIS:** In this program, we are converting binary to gray code. The procedure is simple. At first, we have to shift the content to the right, then perform XOR operation with the shifted content. and the actual content. Then we will get the gray code. For an example if the number is ABH, then the binary value will be (10101011), after shifting the value we will be (01010101) = 55H, now by XORing ABH and 55H, the result will be (11111110) = FEH.

- **ALGORITHM:** Assume that a 8 bit number is stored in memory location D000H.

Step 1: The 8 bit number is loaded to the accumulator.

Step 2: Move accumulator's content to register B.

Step 3: Logical OR operation is performed with the accumulator's content itself. to set carry flag.

Step 4: Rotate accumulator right through carry so that bit D_0 becomes carry bit and the previous bit becomes bit D_7. Each bit is shifted to adjacent right position.

Step 5: Exclusive-OR is performed with register B

Step 6: Store the result in memory location D001H and stops execution.

• TABLE :- The table of binary to gray code conversion is shown below :-

Memory Location	Lebel	Mnemonics	Hex Code	Comments
C000		LDA D000H	3A	Load accumulator
C001			00	content

Topic :

PIONEER

Page :

Date : / /

Memory Location	Lebel	Mnemonics	Hex Code	Comments.
C002			D0	
C003		Mov B, A	47	Move the content of accumulator to reg. B
C004		ORA A	B7	Logical OR performed
C005		RAR	1F	Rotate acc. Right through carry
C006		XRA B	A8	Exclusive OR is performed with accumulator
C007		STA D001H	32	Store the content
C008			01	of accumulator to
C009			D0	memory.
C00A		HLT	76	Stop Execution

INPUT AND OUTPUT :-

Set	Input	Output
Set-1	0A (00001010)	0F (00001111)
Set-2	AB (10101011)	FE (11111110)
Set-3	77 (01110111)	4C (01001100)
Set-4	CD (11001101)	AB (10101011)
Set-5	04 (00000100)	06 (00000110)

Topic :

PIONEER

Page :

Date : / /

Memory Location	Lebel	Mnemonics	Hex Code	Comments.
C002			D0	
C003		Mov B, A	47	Move the content of accumulator to reg B
C004		ORA A	B7	Logical OR performed
C005		RAR	1F	Rotate acc. Right through carry
C006		XRA B	A8	Exclusive OR is performed with accumulator
C007		STA D001H	32	Store the content
C008			01	of accumulator to
C009			D0	memory.
C00A		HLT	76	Stop Execution

INPUT AND OUTPUT :-

Set	Input	Output
Set -1	0A (00001010)	0F (00001111)
Set-2	AB (10101011)	FE (11111110)
Set-3	77 (01110111)	4C (01001100)
Set-4	CD (11001101)	AB (10101011)
Set-5	04 (00000100)	06 (00000110)

Topic

PIONEER

Page

Date / /

OBJECTIVE: 16 bit + 16 bit Addition

- PROBLEM STATEMENT: Write an Assembly Language Program to find the addition of two 16 bit Numbers and Check the Carry Flag.

- PROBLEM ANALYSIS: Two 16 bit numbers are taken in the two register H-L and DE respectively. Then this two numbers are add. The result is stored in the memory location D004H and D005H and the carry (if generated) is stored in memory location D006H. The two input are taken in the memory location D000, D001, D002 and D003H respectively.

- ALGORITHM: The formal algorithm steps for two 16 bit number addition in 8085 microprocessor are as follows —

Step 1: Load the given address to the L register and the next location address to the H register.

Step 2: Exchange the content of HL register to DE register.

Topic

PIONEER

Page

Date : / /

OBJECTIVE : 16 bit + 16 bit Addition

- **PROBLEM STATEMENT :** Write an Assembly Language Program to find the addition of Two 16 bit Numbers and Check the Carry Flag.

- **PROBLEM ANALYSIS :** Two 16 bit numbers are taken in the two register H-L and DE respectively. Then this two numbers are add. The result is stored in the memory location D004H and D005H and the carry (if generated) is stored in memory location D006H. The two input are taken in the memory location D000, D001, D002 and D003H respectively.

- **ALGORITHM :** The formal algorithm steps for two 16 bit number addition in 8085 microprocessor are as follows —

Step 1 : Load the given address to the L register and the next location address to the H register.

Step 2 : Exchange the content of HL register to DE register.

Topic

Page

Date / /

Step 3: Load the given address to the L register and the next location address to the H register.

Step 4: Add the content of H with D and L with E and store the result in H-L register.

Step 5: Store the result at the given memory location

Step 6: Move 00 H to the accumulator.

Step 7: Transfer program control to the specified address if the carry flag is 0 and go to step 9.

Step 8: If the carry flag is 1, then increment the content of accumulator.

Step 9: Store the content of accumulator to the specified memory location

Step 10: STOP

Memory Location	Lebel	Mnemonics	Hex Code	Comments
C000		LHLD D000H	2A	Load D000H to L register and D001 to H register
C001			00	
C002			D0	
C003		XCHG	EB	Exchange the content of HL register to DE register
C004		LHLD D002H	2A	Load D002 H to L register and D003H to H register.
C005			02	
C006			D0	
C007		DAD D	19	Add the content of HL register with DE register
C008		SHLD D004H	22	Store the result at the D004 and D005 memory location (D004 for L register and D005 for H register).
C009			04	
C00A			D0	
C00B				
C00B		MVI A, 00H	3E	Move 00H to accumulator.
C00C			00	
C00D		JNC C011H (NEXT)	D2	Transform the program control to C011 location if carry flag is zero.
C00E			11	
C00F			C0	
C010		INR A	3C	Increment the content of accumulator if carry flag is not zero
C011	NEXT	STA D006H	32	Store the content of accumulator

Topic :

PIONEER

Page :

Date : / /

Memory Location	Lebel	Mnemonics	Hex Code	Comments
C012			06	into D006 memory
C013			D0	location.
C014		HLT	76	Stop the program.

Set	Input				Output		
	D000	D001	D002	D003	D004	D005	D006 (Carry)
Set 1	86	20	3D	2C	C3	4C	00
Set 2	98	56	4A	3B	E2	91	00
Set 3	65	3C	19	5A	7E	96	00
Set 5	75	5A	45	B7	BA	11	01
Set 6	93	27	82	F1	15	19	01

• DISCUSSION :

a) The 3 bytes instruction which is used here are LHLD, SHLD, JNC, STA

b) The 2 bytes instruction which used here is MVI.

DISCUSSION :

c) The single byte instruction which used here are XCHG, DAD, INR, HLT

d) The total 10 instruction which used here (in the program) and the total bytes are 22.

e) The HLT that is halt instruction the value of program counter. Since Program counter is not getting update and keeps its value as its is, fetching of any Instruction is not Possible. This will keep microprocessor is stedy position.