8-BIT ADDITION

EXP NO: 1

AIM:
To write an assembly language program to implement 8-bit addition using 8085 processor.

ALGORITHM:
1) Start the program by loading the first data into the accumulator.

2) Move the data to a register.

3) Get the second data and load it into the accumulator.

4) Add the two register contents.

5) Check for carry.

6) Store the value of sum and carry in the memory location.

7) Halt.

PROGRAM:

MNEMONICS	EXPLANATION
LDA 8050	Load accumulator with first number in the address 8085
MOV B, A	Move the data from accumulator to'B' register
LDA 8051	Load accumulator with second number in the address 8051
ADD B	Add the data of 'B' register with accumulator
STA 8052	Store the data (Output) of the accumulator in address 8052
HLT	Halt

INPUT:

ADDRESS	DATA
8050	1
8051	2

OUTPUT:

ADDRESS	DATA
8052	3

RESULT: Thus the program was executed successfully using 8085 processor simulator

8-BIT SUBTRACTION

EXP NO: 2

AIM: To write an assembly language program to implement 8-bit subtraction using 8085 processor.

ALGORITHM:

1) Start the program by loading the first data into the accumulator.

2) Move the data to a register.

3) Get the second data and load it into the accumulator.

4) Subtract the two register contents.

5) Check for borrow.

6) Store the difference and borrow in the memory location.

7) Halt.

PROGRAM:

MNEMONICS	EXPLANATION
LDA 8000	Load accumulator with the first number in the address
MOV B, A	Move the data from accumulator to B register
LDA 8001	Load accumulator with the second number in the address
SUB B	Subtract the data B register with accumulator
STA 8002	Store the data (Output) of the accumulator in the address
RST 1	HALT

INPUT:

ADDRESS	DATA
8000	4
8001	5

OUTPUT:

ADDRESS	DATA
8002	1

RESULT: Thus the program was executed successfully using 8085 processor simulator

8-BIT MULTIPLICATION

EXP NO: 3

AIM: To write an assembly language program to implement 8-bit multiplication using 8085 processor.

ALGORITHM:

1) Start the program by loading a register pair with the address of memory location.

2) Move the data to a register.

3) Get the second data and load it into the accumulator.

4) Add the two register contents.

5) Increment the value of the carry.

6) Check whether the repeated addition is over.

7) Store the value of product and the carry in the memory location.

8) Halt.

PROGRAM:

MNEMONICS	**EXPLANATION**
LDA 2200	Load the accumulator with the first number in the address 8500
MOV E,A	Move the data from accumulator to 'E' register
MVI D,00	Move the immediate value 00 into
LDA 2201	Load the accumulator number in the address 2201
MOV C,A	Move the data from a ccumulator to 'C' register
LXI H,0000	Load the immediate value 0000 into the HL register pair
BACK: DAD D	Back : Label for the loop D ADD: Add the value in register D
DCR C	Decrement register E by 1
JNZ BACK	In register E, is not 0,jump back to the beginning of the loop
SHLD 2202	Store the value on the HL register pair at memory address 2202
HLT	HALT

INPUT:

ADDRESS	DATA
2200	4
2201	2

OUTPUT:

ADDRESS	DATA
2202	8

RESULT: Thus the program was executed successfully using 8085 processor simulator.

8-BIT DIVISION

EXP NO: 4

AIM: To write an assembly language program to implement 8-bit division using 8085 processor.

ALGORITHM:

1) Start the program by loading a register pair with the address of memory location.

2) Move the data to a register.

3) Get the second data and load it into the accumulator.

4) Subtract the two register contents.

5) Increment the value of the carry.

6) Check whether the repeated subtraction is over.

7) Store the value of quotient and the reminder in the memory location.

8) Halt.

PROGRAM:

MNEMONICS	**EXPLANATION**
START: NOP	It is often used for code alignment
LDA 8500	Load the accumulator with first number in address 8500
MOV B, A	Move data from accumulator to 'B' register
LDA 8501	Load the accumulator with second number in the address 8501
MVI C,00 H	Move the immediate value 00 into register 'C'
LOOP:CMP B	Loop : Label for loop CMP B: Compare the value in accumulator (A) with (B)
JC LOOP1	If the carrying is (A<B) jump to label loop 1
SUB B	Subtract the value in register (B) from the accumulator (A)
INR C	Increment register C by 1
JMP LOOP	Jump back to the beginning of the loop
STA 8502	Store the data in accumulator 8502
MOV A, C	Move the data from `C' register to accumulator
STA 8503	Store the data in the accumulator in 8503
RST 1	Typically transfer control to a predefined interrup service routine
HLT	HALT

INPUT:

ADDRESS	DATA
8500	2
8501	6

OUTPUT:

ADDRESS	DATA
8502	0
8503	3

RESULT: Thus the program was executed successfully using 8085 processor simulator.

16-BIT ADDITION

EXP NO: 5

AIM:-

To write an assembly language program to implement 16-bit addition using 8085 processor.

ALGORITHM:-

1) Start the program by loading a register pair with address of 1st number.

2) Copy the data to another register pair.

3) Load the second number to the first register pair.

4) Add the two register pair contents.

5) Check for carry.

6) Store the value of sum and carry in memory locations.

7) Terminate the program.

PROGRAM:-

MNEMONICS	Explanation
LDA 3050	Load the content of the memory location 3050H into the accumulator (A).
MOV B,A	Move the content of the accumulator (A) into register B. Now, register B contains the value from 3050H.
LDA 3051	Load the content of the memory location 3051H into the accumulator (A).
ADD B	Add the content of register B (value from 3050H) to the accumulator (A).
STA 3052	Store the result of the addition (from the accumulator) into memory location 3052.
LDA 3053	Load the content of the memory location 3053H into the accumulator (A).
MOV B,A	Move the content of the accumulator (A) into register B.
LDA 3054	Load the content of the memory location 3054H into the accumulator (A).
ADC B	Add the content of register B to the accumulator (A)
STA 3055	Store the result of the addition into memory location 3055.
HLT	Halt the execution of the program.

INPUT:-

Address	Data
3050	2
3051	3
3053	5
3054	5

OUTPUT:-

Address	Data
3052	5
3055	10

RESULT:-

Thus the program was executed successfully using 8085 processor simulator.

16-BIT SUBTRACTION

EXP NO: 6

AIM:-

To write an assembly language program to implement 16-bit subtraction using 8085 processor.

ALGORITHM:-

1) Start the program by loading a register pair with address of 1st number.

2) Copy the data to another register pair.

3) Load the second number to the first register pair.

4) sub the two register pair contents.

5) Check for carry.

6) Store the value of sum and carry in memory locations.

7) End.

PROGRAM:-

MNEMONICS	Explanation
LHLD 2050	Load the 16-bit data from memory locations 2050H and 2051H into the HL register pair.
XCHG	Exchange the contents of the HL and DE register pairs.
LHLD 2052	Load the 16-bit data from memory locations 2052H and 2053H into the HL register pair.
MVI C,00	Move the immediate value 00H into register C.
MOV A, E	Move the content of register E into the accumulator (A).
SUB L	Subtract the content of register L from the accumulator (A).
STA 2054	Store the result of the subtraction into memory location 2054H.
MOV A, D	Move the content of register D into the accumulator (A).
SUB H	Subtract the content of register H from the accumulator (A).
STA 2055	Store the result of the subtraction into memory location 2055H.
HLT	Halt the execution of the program.

INPUT:-

Address	Data
2050	2
2052	3

OUTPUT:-

Address	Data
2054	1
2055	1

RESULT:-

Thus the program was executed successfully using 8085 processor simulator.

16-BIT MULTIPLICATION

EXP NO: 7

AIM:-

To write an assembly language program to implement 16-bit multiplication using 8085 processor.

ALGORITHM:-

1) Load the first data in HL pair.

2) Move content of HL pair to stack pointer.

3) Load the second data in HL pair and move it to DE.

4) Make H register as 00H and L register as 00H.

5) ADD HL pair and stack pointer.

6) Check for carry if carry increment it by 1 else move to next step.

7) Then move E to A and perform OR operation with accumulator and register D.

8) The value of operation is zero, then store the value else go to step

PROGRAM:-

MNEMONICS	Explanation
LHLD 2050	Loads the contents of memory location 2050H and 2051H into the HL register pair.
SPHL	Sets the Stack Pointer (SP) to the value in the HL register pair.
LHLD 2052	Loads the contents of memory location 2052H and 2053H into the HL register pair.
XCHG	Exchanges the contents of the HL and DE register pairs.
LXI H,0000H	Loads the value 0000H into the HL register pair.
LXI B,0000H	Loads the value 0000H into the BC register pair.
AGAIN: DAD SP	Marks the beginning of a loop, Adds the contents of the SP register pair to the HL register pair.
JNC START	Jumps to the START label if the carry flag is not set (i.e., no overflow occurred in the previous operation).
INX B	Increments the BC register pair by 1.
START: DCX D	Marks the start of another loop, Decrements the DE register pair by 1.
MOV A,E	Moves the contents of register E into register A.

ORA D	Performs a logical OR operation between A and D.
JNZ AGAIN	Jumps back to the AGAIN label if the Zero flag is not set (i.e., if the result of the OR operation is non-zero).
SHLD 2054	Stores the contents of the HL register pair into memory locations 2054H and 2055H.
MOV L,C	Moves the contents of register C into register L.
MOV H,B	Moves the contents of register B into register H.
SHLD 2055	Stores the contents of the HL register pair into memory locations 2055H and 2056H.
HLT	Halts the program.

INPUT:-

Address	Data
2050	10
2052	5

OUTPUT:-

Address	Data
2054	50
2055	5

RESULT:-

Thus the program was executed successfully using 8085 processor simulator.

16-BIT DIVISION

EXP NO: 8

AIM:-

To write an assembly language program to implement 16-bit division using 8085 processor.

ALGORITHM:-

1) Read dividend (16 bit)

2) Read divisor

3) count <- 8

4) Left shift dividend

5) Subtract divisor from upper 8-bits of dividend

6) If CS = 1 go to 9

7) Restore dividend

8) Increment lower 8-bits of dividend

9) count <- count - 1

10) If count = 0 go to 5

11) Store upper 8-bit dividend as remainder and lower 8-bit as quotient

12) Stop

PROGRAM:-

MNEMONICS	Explanation
LDA 8501	loads the accumulator (A) with the contents of memory location 8501H.
MOV B,A	moves the contents of register A into register B.
LDA 8500	loads the accumulator (A) with the contents of memory location 8500H.
MVI C,00	loads the immediate value 00H into register C.
LOOP:CMP B	the beginning of a loop, compares the contents of register A (accumulator) with register B.
JC LOOP1	Jumps to the LOOP1 label if the carry flag is set (i.e., if A is less than B after the comparison).
SUB B	Subtracts the contents of register B from the accumulator (A) and stores the result in A.
INR C	Increments register C by 1.
JMP LOOP	Jumps back to the LOOP label, effectively continuing the loop.
STA 8503	Stores the contents of the accumulator (A) into memory location 8503H.

DCR C	Decrements the contents of register C by 1.
MOV A,C	Moves the contents of register C into the accumulator (A).
LOOP1: STA 8502	Marks the continuation point of the program after the jump, Stores the contents of the accumulator (A) into memory location 8502H.
RST 1	a software interrupt and jumps to address 0024H (based on the 8085 architecture).

INPUT:-

Address	Data
8051	20
8050	2

OUTPUT:-

Address	Data
8502	10
8503	2

RESULT:-

Thus the program was executed successfully using 8085 processor simulator