

## **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 remainder in the memory location.
- 8) Halt.

### **PROGRAM:**

```
LDA 8501  
  
MOV B, A  
  
LDA 8500  
  
MVI C,00
```

LOOP: CMP B

JC LOOP1

SUB B

INR C

JMP LOOP

STA 8503

DCR C

MOV A, C

LOOP1: STA 8502

RST 1

**INPUT:**

**8500-9**

**8501-2**

## OUTPUT:

The screenshot displays the 8085 processor simulator interface. The main window is divided into several sections:

- Registers:** A table showing the status of various registers. The **Flag** section shows **S** (Sign) as 1, **Z** (Zero) as 0, **AC** (Auxiliary Carry) as 0, **P** (Parity) as 1, and **C** (Carry) as 1.
- Decimal - Hex Conversion:** A section for converting between decimal and hexadecimal values. Both fields show 0.
- I/O Ports:** A section for updating port values. The value field shows 0.
- Memory:** A section for updating memory values. The value field shows 0.
- Assembly Code:** A list of instructions being executed:

```
1 LDA 8501
2 MOV B, A
3 LDA 8500
4 MVI C, 00
5
6 LOOP: CMP B
7 JC LOOP1
8 SUB B
9 INR C
10 JMP LOOP
11
12 STA 8503
13 DCR C
14 MOV A, C
15 LOOP1: STA 8502
16 RST 1
17
```
- Memory Window:** A table showing memory addresses and data:

Address (Hex)	Address	Data
2134	8500	9
2135	8501	2
2136	8502	1
2137	8503	0
2138	8504	0
2139	8505	0
213A	8506	0
213B	8507	0
213C	8508	0
213D	8509	0
213E	8510	0
213F	8511	0
2140	8512	0
2141	8513	0
- Assembler Message:** A message box showing "0 Program assembled successfully".

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