

## 16-BIT DIVISION

EXP NO: 8

**AIM:** To write an assembly language program to implement 16-bit divided by 8-bit 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:

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&OUTPUT:

The screenshot displays the GNUSim8085 - 8085 Microprocessor Simulator interface. The main window shows the assembly code being executed, with line numbers 1 through 14. The code includes instructions like LDA, MOV, STA, JMP, and RST. The left panel shows the state of the 8085 registers (A, B, C, D, E, H, L, PSW, PC, SP, Int-Reg) and the flag register (S, Z, AC, P, C). The right panel shows the memory dump, with addresses ranging from 8500 to 8511. The bottom status bar indicates the simulator is in an 'Idle' state.

**Registers**

Register	Value
A	02
BC	0E 02
DE	00 00
HL	0D B1
PSW	00 00
PC	42 1A
SP	FF FF
Int-Reg	00

**Flag**

Flag	Value
S	1
Z	0
AC	0
P	1
C	1

**Assembly Code**

```
1 LDA 8501
2 MOV B,A
3 LDA 8500
4 MVI C,00
5 LOOP: CMP B
6 JC LOOP1
7 SUB B
8 INR C
9 JMP LOOP
10 LOOP1: STA 8503
11 MOV A,C
12 STA 8502
13 RST 1
14
```

**Memory Dump**

Address (Hex)	Address	Data
2134	8500	32
2135	8501	14
2136	8502	2
2137	8503	4
2138	8504	0
2139	8505	0
213A	8506	0
213B	8507	0
213C	8508	0
213D	8509	0
213E	8510	0
213F	8511	0

**Assembler Message**

```
0 Program assembled successfully
```

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