

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 \leftarrow 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 \leftarrow 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:

```
8500-13
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 and **C** (Carry) as 1.
- Decimal - Hex Conversion:** A section for converting between decimal and hexadecimal values. The decimal input is 0, and the hex output is 0.
- I/O Ports:** A section for updating port values. The current value is 0.
- Memory:** A section for updating memory values. The current value is 0.
- Assembly Code:** A list of instructions being executed, including `LDA 8500H`, `MOV D, A`, `LDA 8501H`, `MOV B, A`, `MVI C, 00H`, `MOV A, D`, `LOOP: CMP B`, `JC END`, `SUB B`, `INR C`, `JMP LOOP`, `END: STA 8503H`, `MOV A, C`, `STA 8502H`, and `RST 1`.
- Memory Window:** A table showing the memory address, hex value, and data. The address range is from 8501 to 850E. The data values are 34049, 34050, 34051, 34052, 34053, 34054, 34055, 34056, 34057, 34058, 34059, 34060, 34061, and 34062.
- Assembler Message:** A message box indicating that the program was assembled successfully.

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