

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:

LDA 8000

MOV B, A

LDA 8001

SUB B

STA 8002

RST 1

INPUT:

Start			8000
Address (Hex)	Address	Data	
1F40	8000	3	
1F41	8001	9	

OUTPUT:

The screenshot displays the GNUSim8085 - 8085 Microprocessor Simulator interface. The main window is divided into several sections:

- Registers:** Shows the state of various registers. For example, A is 06, BC is 03 00, DE is 00 00, HL is 00 00, PSW is 00 00, PC is 42 10, SP is FF FF, and Int-Reg is 00. Flags S, Z, AC, P, and C are also shown.
- Decimal - Hex Conversion:** A section for converting between decimal and hexadecimal values.
- I/O Ports:** A section for updating port values.
- Memory:** A section for updating memory values.
- Assembly Code:** A central area for writing and editing assembly code. The code includes:

```
1 ;<Program title>
2
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: nop
11 LDA 8000
12
13
14 MOV B, A
15
16 LDA 8001
17
18
19 SUB B
20
21
22 STA 8002
23
24
25 RST 1
26
27
28
29
30
31
32 hlt
```
- Memory Window:** A table showing memory addresses and data:

Address (Hex)	Address	Data
1F40	8000	3
1F41	8001	9
1F42	8002	6
1F43	8003	0
1F44	8004	0
1F45	8005	0
1F46	8006	0
1F47	8007	0
1F48	8008	0
1F49	8009	0
1F4A	8010	0
1F4B	8011	0
1F4C	8012	0
1F4D	8013	0
- Assembler Message:** A section showing the result of the assembly process, indicating "Program assembled successfully".

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