

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:

```
LDA 8500
MOV B, A
LDA 8501
ADD B
STA 8502
RST 1
```

INPUT:

The screenshot displays the 8085 Assembler interface. The 'Memory' window shows a table of memory addresses and their corresponding data values. The 'Assembler Message' window at the bottom indicates that the program was assembled successfully.

Address (Hex)	Address	Data
2134	8500	2
2135	8501	4
2136	8502	6
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
2142	8514	0

Line No	Assembler Message
0	Program assembled successfully

OUTPUT:

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers

Register	Value
A	06
BC	02 00
DE	00 00
HL	00 00
PSW	00 00
PC	42 0C
SP	FF FF
Int-Reg	00

Flag

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

Load me at

```
1  
2 LDA 8500  
3 MOV B,A  
4 LDA 8501  
5 ADD B  
6 STA 8502  
7 HLT
```

Decimal - Hex Conversion

Decimal: 0 Hex: 0

To Hex To Dec

I/O Ports

0 - + 00

Update Port Value

Memory

0 - + 00

Update Memory

Start 8500 OK

Address (Hex)	Address	Data
2134	8500	2
2135	8501	4
2136	8502	6
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
2142	8514	0

Line No Assembler Message

0 Program assembled successfully

Simulator: Idle

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