

## 16-BIT SUBTRACTION

### EXP NO: 6

**AIM:** To write an assembly language program to implement 16-bit subtraction using 8085 processor.


### ALGORITHM:

- 1) Start the program by loading a register pair with address of 1<sup>st</sup> number.
- 2) Copy the data to another register pair.
- 3) Load the second number to first register pair.
- 4) Subtract the two register pair contents.
- 5) Check for borrow.
- 6) Store the value of difference and borrow in memory locations.
- 7) End.

### PROGRAM:

```
LHLD 2050
XCHG
LHLD 2052
MVI C,00
MOV A, E
SUB L
STA 2054
MOV A, D
SUB H
STA 2055
HLT
```

### INPUT:



The screenshot shows a memory editor window with a table of memory addresses and data. The 'Start' address is 2050. The table lists addresses from 0002 to 000D with their corresponding data values. Below the table, the 'Assembler Messages' section shows a single message: 'Program assembled successfully'.

Address (Hex)	Address	Data
0002	2050	20
0003	2051	0
0004	2052	5
0005	2053	0
0006	2054	15
0007	2055	0
0008	2056	0
0009	2057	0
000A	2058	0
000B	2059	0
000C	2060	0
000D	2061	0

Line No: Assembler Messages  
0 Program assembled successfully

### OUTPUT:

VS 18Code 55% faster with GitHub Copilot

Registers

Register	Value	Flag
A	00	Z 0
PC	00	
SP	00	14 Z 1
AC	00	05
PSW	00	00 AC 0
PC	42	14 P 1
SP	00	FF
SP-Reg	00	C 0

Decimal - Hex Conversion

Decimal

Hex

0

0

To Hex

To Dec

I/O Ports

0

00

Update Port Value

Memory

2054

00

Update Memory

Load new file

```
1 LARLD 2050
2 XCHG
3 LARLD 2052
4 MOV C, 20
5 MOV A, 5
6 INC A
7 STA 2054
8 MOV A, B
9 INC B
10 STA 2055
11 HLT
```

Stack

Stack

Keypad

Memory

Use Binary

Start 2050

OK

Address (Hex)	Address	Data
0000	2050	20
0001	2051	0
0002	2052	5
0003	2053	0
0004	2054	15
0005	2055	0
0006	2056	0
0007	2057	0
0008	2058	0
0009	2059	0
000A	2060	0
000B	2061	0

Show the Assembly Message

0

Program assembled successfully

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