Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Mini-Project Report

on

Add Two 16 bit numbers with carry using Assembly Language

[Microprocessor and Assembly Language - COMP 231]

Submitted by:

Sadikshya Pokharel (36)
Ayush Regmi (39)
Roshan Sahani (42)
Soniya Sharma (47)
Reewaj Khanal (61)

Submitted to:

Dr. Gajendra Sharma

Department of Computer Science and Engineering

Submission Date:
2023/05/15

Table of Contents

Task

10. Write a program to add Two 16 bit numbers with carry.

The program uses the HL, DE, C, and accumulator registers to store the numbers and perform the addition. The carry flag is used to handle overflow, which occurs when the result of the addition exceeds the maximum value that can be represented by 16 bits.

At first, assuming the two 16 bit numbers are already stored in the memory, we load the DE and HL register pairs. Initially we set the value in the C register, which will be used to store carry(if any) to 0. Then we perform addition and store the result in the HL pair itself. If there is a carry, then value in the C register is incremented by 1. Then both carry and the sum present in C and HL registers are stored in the memory.

Flowchart

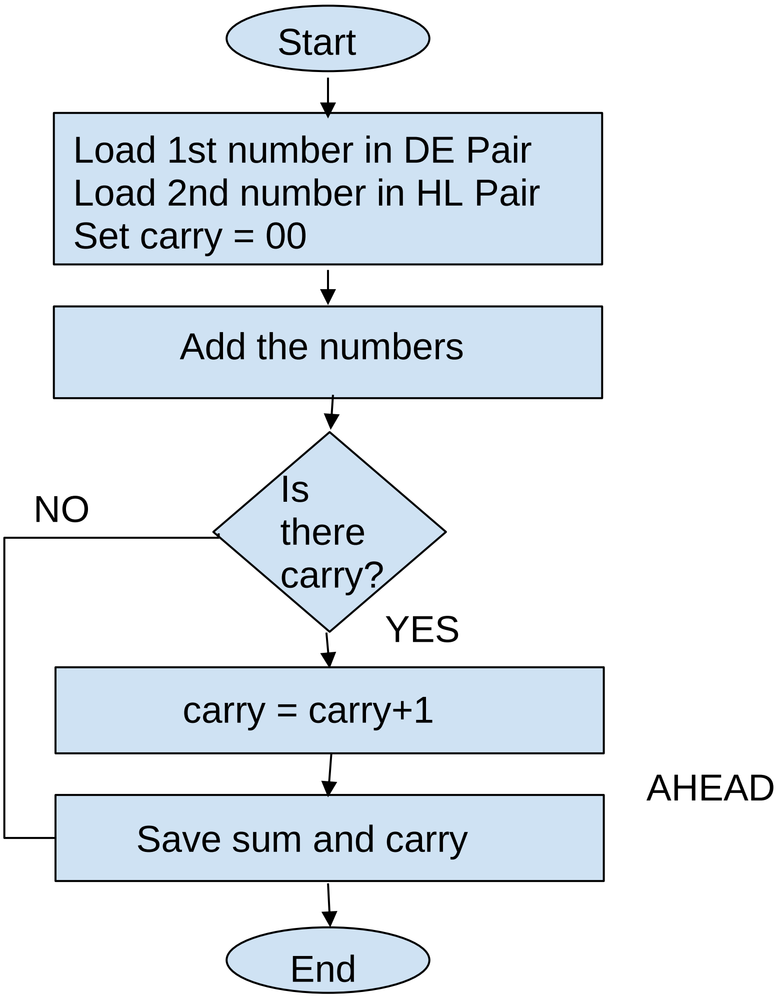


Figure 1: Flowchart

Instructions

- LHLD : stands for "Load H and L Direct". The LHLD instruction is used to load the contents of a memory location into the H and L registers of the microprocessor simultaneously.

  Eg: LHLD 2000H ; Load contents of memory location 2000H into H and L register

- XCHG : stands for "Exchange H and L with D and E". The XCHG instruction is used to swap the contents of the H and L registers with the contents of the D and E registers of the microprocessor.

  Eg: XCHG ; Swap the contents of H and L with D and E

- MVI : stands for "Move Immediate". The MVI instruction is used to move an 8-bit data value directly into a register or memory location.

  Eg: MVI B, 42H ; Load register B with 42H

- DAD : stands for "Double Add". The DAD instruction is used to add the contents of the H and L registers with the contents of the D and E registers and store the result in the H and L registers.

  Eg: DAD D ; Add the contents of H and L with the contents of D and E

- JNC : stands for "Jump if No Carry". The JNC instruction is used to transfer control to a specific memory address if the Carry flag is not set.

  Eg: JNC NEXT ; Jump to NEXT if Carry flag is not set

- INR : stands for "Increment". The INR instruction is used to increment the value of a register or memory location by 1.

  Eg: INR A ; Increment the value of accumulator by 1

- SHLD : stands for "Store H and L Direct". The SHLD instruction is used to store the contents of the H and L registers into a specific memory address.

Eg: SHLD 3000H ; Store the contents of the H and L registers at memory address 3000H

- MOV : stands for "Move". The MOV instruction is used to copy the contents of a source register or memory location into a destination register or memory location.

  Eg: MOV B, A ; Copy the contents of accumulator to register B

- STA : stands for "Store Accumulator". The STA instruction is used to store the contents of the accumulator register into a specific memory address.

  Eg: STA 3000H ; Store the contents of the accumulator at memory address 3000H

- HLT : stands for "Halt". The HLT instruction is used to stop the execution of the program and put the microprocessor in a halt state.

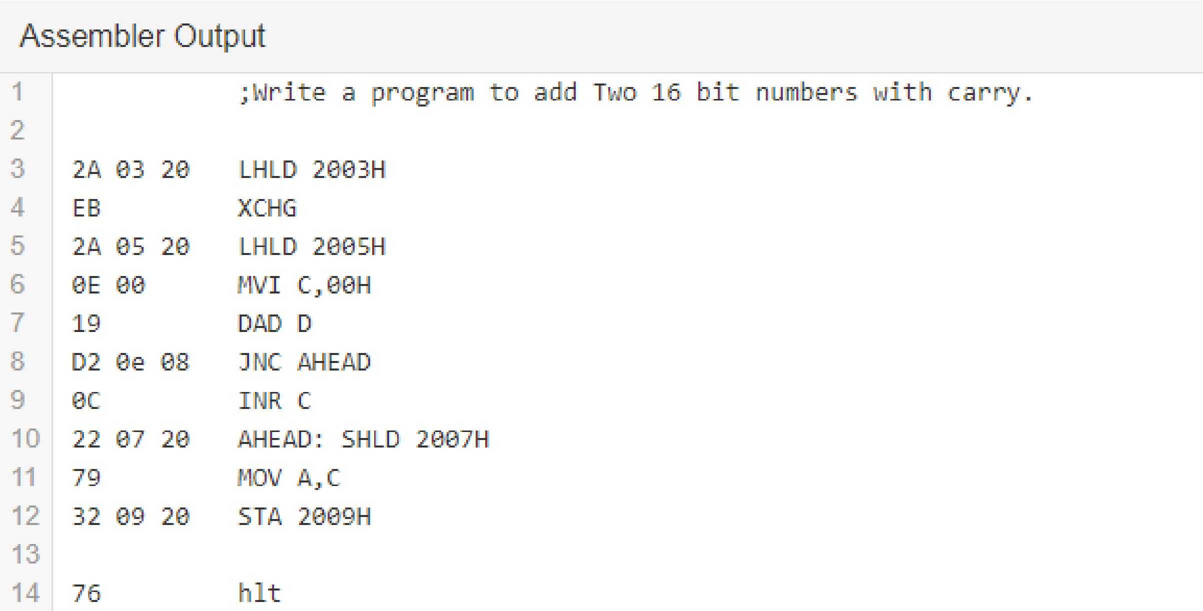
  Eg: HLT ; Halt the program

Source Code

Following is the source code for the addition of two 16- bit numbers.

Label	OPCODE + OPERAND	Comment
	LHLD 2003H	; Load HL pair direct from 2003H [L] = [2003H] & [H] = [2004H] I.e. load 1st number to HL pair
	XCHG	; Exchange contents of HL pair with contents of DL pair I.e. move 1st number to DE pair
	LHLD 2005H	; Load HL pair direct from 2005H [L] = [2005H] & [H] = [2006H] i.e. load 2nd number to HL pair
	MVI C,00H	; Move Immediately 00 to register C i.e. set carry=00
	DAD D	; Double add contents of HL pair with contents of DE pair [HL]=[HL]+[DE]
	JNC AHEAD	; Jump if No Carry to AHEAD
	INR C	; Increment value of C i.e. carry = carry+1 => [C] = carry = 01
AHEAD :	**SHLD 2007H**	; Store contents of HL pair to 2007H [2007H] =[L] & [2008H] =[H]
	MOV A,C	; Move the carry to Accumulator
	STA 2008H	; Store contents of accumulator i.e. carry in 2008H [2008H] = [A]
	HLT	; Halt the execution

Assembler Output



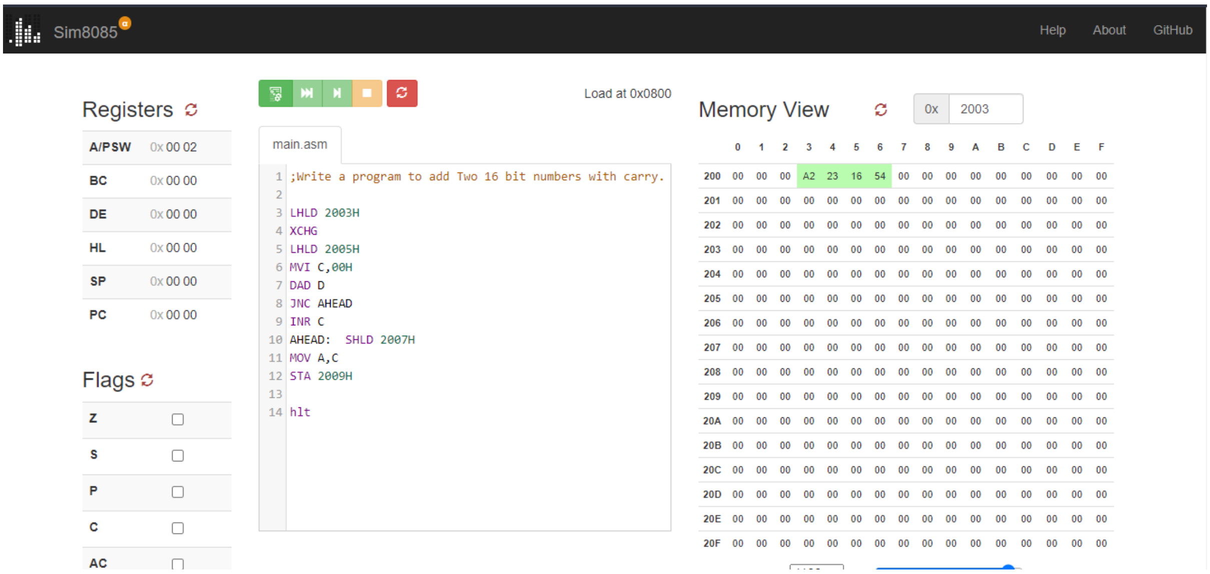
Assembler Output

```
                ;Write a program to add Two 16 bit numbers with carry.

2A 03 20        LHLD 2003H
EB              XCHG
2A 05 20        LHLD 2005H
0E 00           MVI C,00H
19              DAD D
D2 0e 08        JNC AHEAD
0C              INR C
22 07 20        AHEAD: SHLD 2007H
79              MOV A,C
32 09 20        STA 2009H

76              hlt
```

Figure 2 : Assembler Output

Simulation in Sim8085

Example for when there is no carry in addition:



Registers

A/PSW	0x 00 02
BC	0x 00 00
DE	0x 00 00
HL	0x 00 00
SP	0x 00 00
PC	0x 00 00

Flags

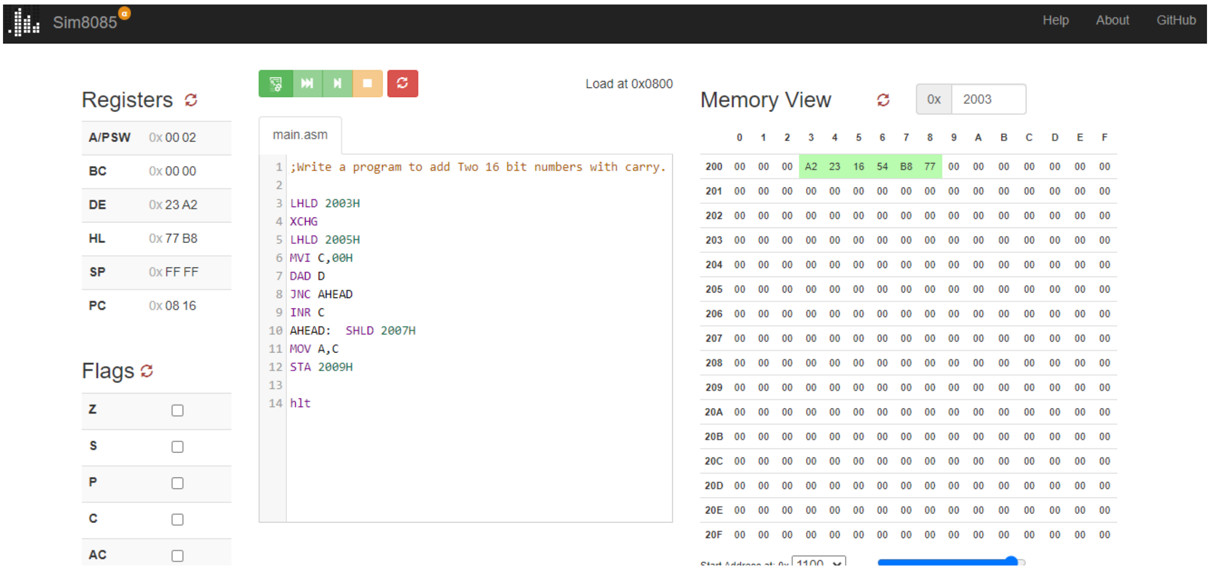
Z	□
S	□
P	□
C	□
AC	□

```
;Write a program to add Two 16 bit numbers with carry.

LHLD 2003H
XCHG
LHLD 2005H
MVI C,00H
DAD D
JNC AHEAD
INR C
AHEAD:  SHLD 2007H
MOV A,C
STA 2009H

hlt
```

Memory View 0x 2003

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
200	00	00	00	A2	23	16	54	00	00	00	00	00	00	00	00	00
201	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
202	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
203	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
204	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
205	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
206	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
207	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
208	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
209	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 3 : Before execution



Registers

A/PSW	0x 00 02
BC	0x 00 00
DE	0x 23 A2
HL	0x 77 B8
SP	0x FF FF
PC	0x 08 16

Flags

Z	□
S	□
P	□
C	□
AC	□

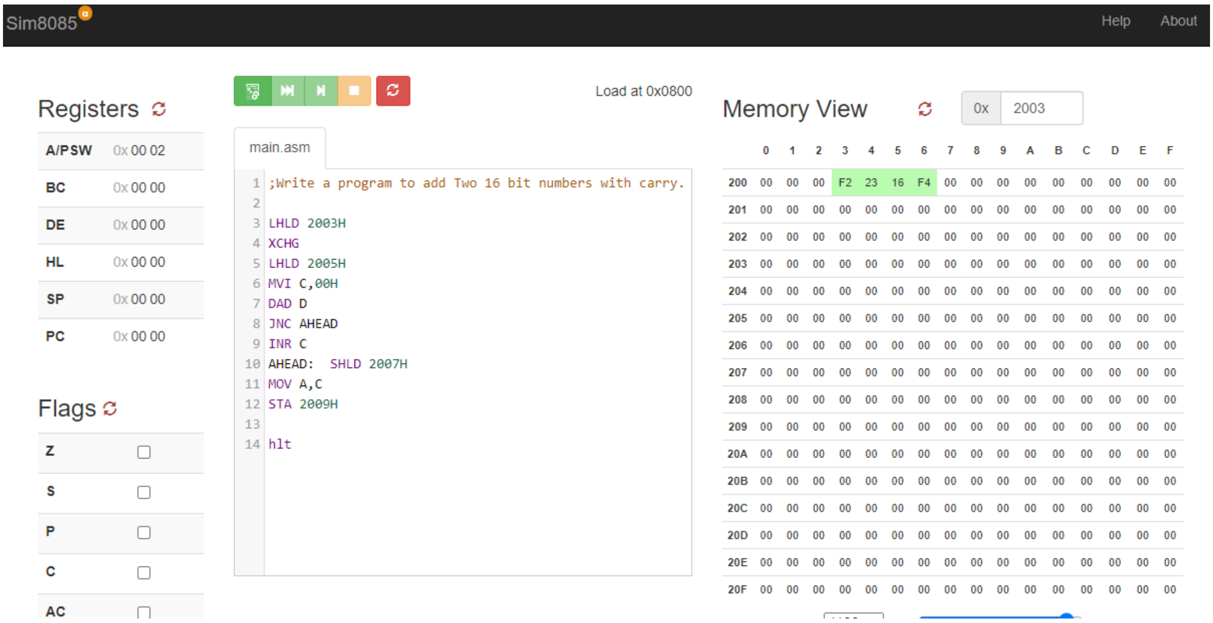
```
;Write a program to add Two 16 bit numbers with carry.

LHLD 2003H
XCHG
LHLD 2005H
MVI C,00H
DAD D
JNC AHEAD
INR C
AHEAD:  SHLD 2007H
MOV A,C
STA 2009H

hlt
```

Memory View 0x 2003

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
200	00	00	00	A2	23	16	54	B8	77	00	00	00	00	00	00	00
201	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
202	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
203	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
204	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
205	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
206	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
207	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
208	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
209	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 4 : After execution

Example for when there is a carry in addition:



```
;Write a program to add Two 16 bit numbers with carry.

LHLD 2003H
XCHG
LHLD 2005H
MVI C,00H
DAD D
JNC AHEAD
INR C
AHEAD:  SHLD 2007H
MOV A,C
STA 2009H

hlt
```

Registers

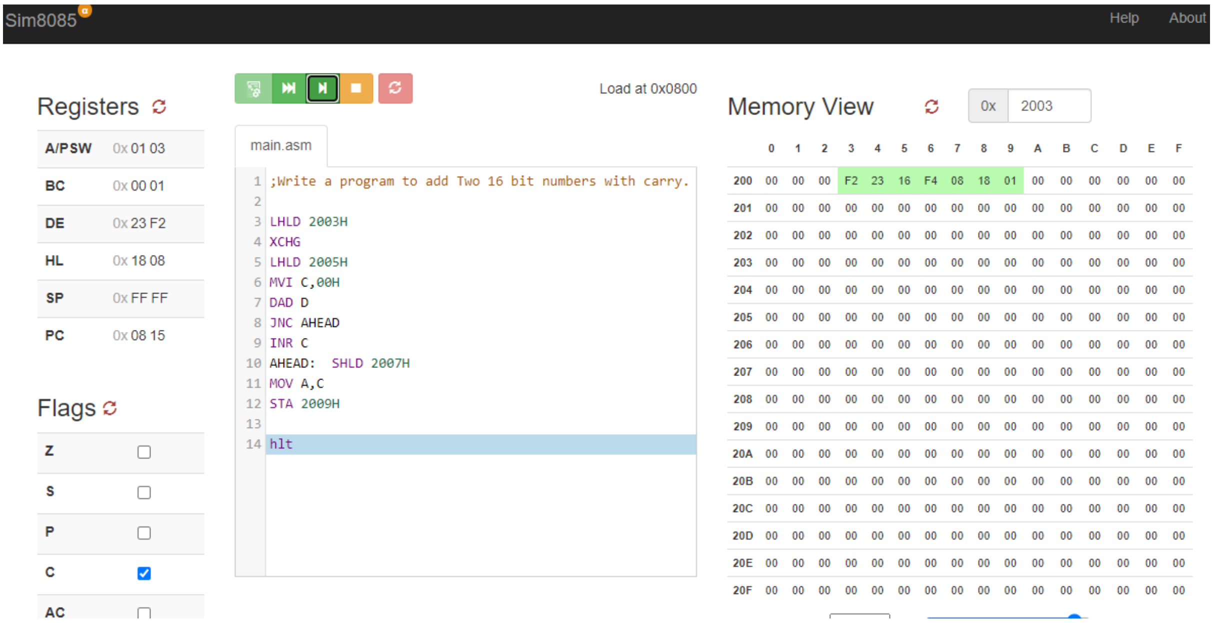
A/PSW	0x 00 02
BC	0x 00 00
DE	0x 00 00
HL	0x 00 00
SP	0x 00 00
PC	0x 00 00

Flags

Z	□
S	□
P	□
C	□
AC	□

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
200	00	00	00	F2	23	16	F4	00	00	00	00	00	00	00	00	00
201	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
202	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
203	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
204	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
205	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
206	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
207	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
208	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
209	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 5 : Before Execution



```
;Write a program to add Two 16 bit numbers with carry.

LHLD 2003H
XCHG
LHLD 2005H
MVI C,00H
DAD D
JNC AHEAD
INR C
AHEAD:  SHLD 2007H
MOV A,C
STA 2009H

hlt
```

Registers

A/PSW	0x 01 03
BC	0x 00 01
DE	0x 23 F2
HL	0x 18 08
SP	0x FF FF
PC	0x 08 15

Flags

Z	□
S	□
P	□
C	☑
AC	□

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
200	00	00	00	F2	23	16	F4	08	18	01	00	00	00	00	00	00
201	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
202	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
203	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
204	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
205	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
206	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
207	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
208	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
209	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 6 : After execution

Conclusion

Our team was successful in achieving the desired outcome on adding two 16 bits numbers with carry by implementing 8085 codes in SIM8085.The project helped to improve the programming skills of the team members and provided practical knowledge of working with microprocessors. Overall, the project demonstrated the importance and relevance of microprocessors in modern computing systems and provided valuable insights into the inner workings of these devices.