**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

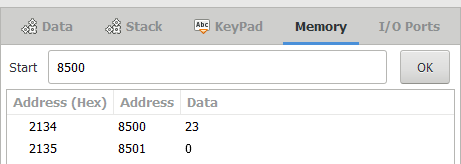
LDA 8501

ADD B

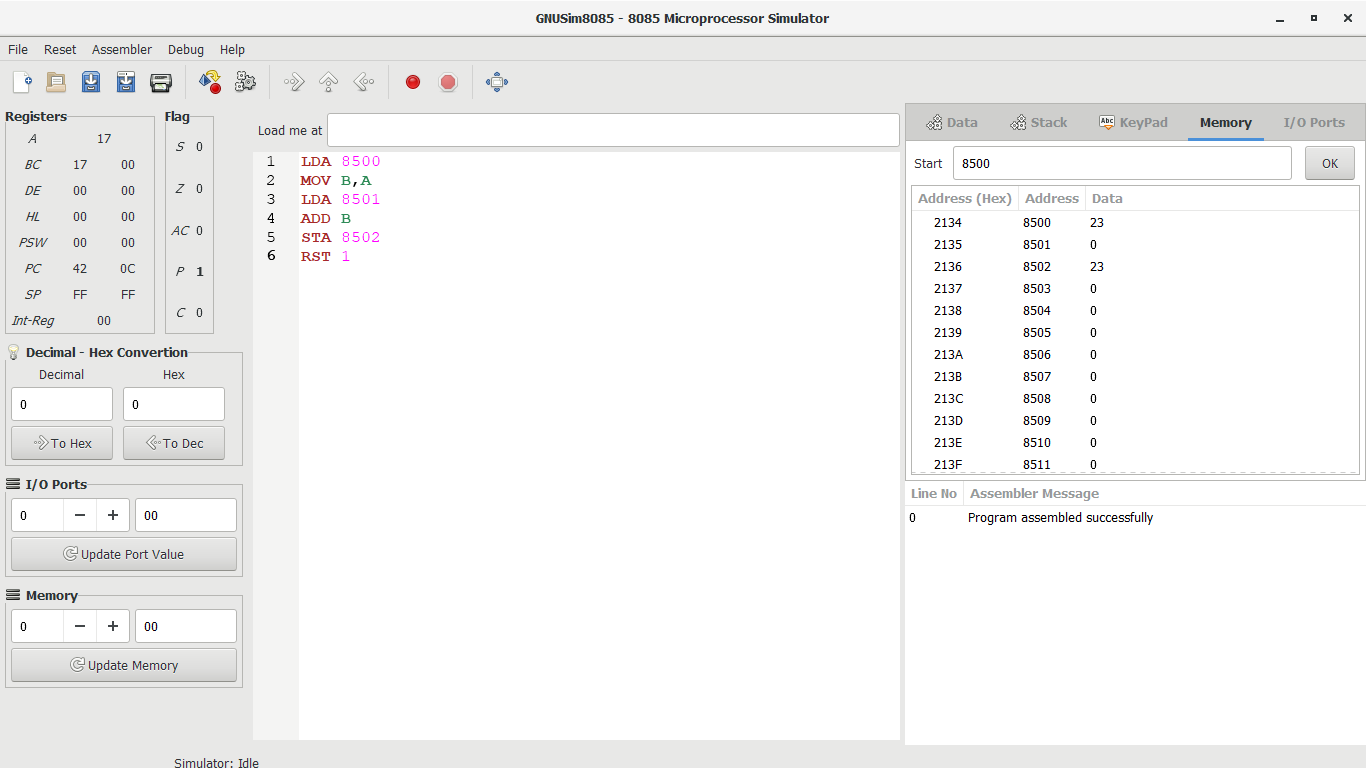
STA 8502

RST 1

**INPUT:**

****

**OUTPUT:**



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

**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

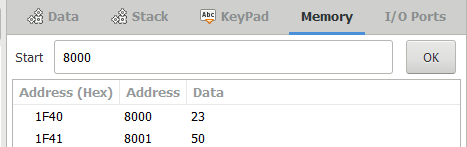
LDA 8001

SUB B

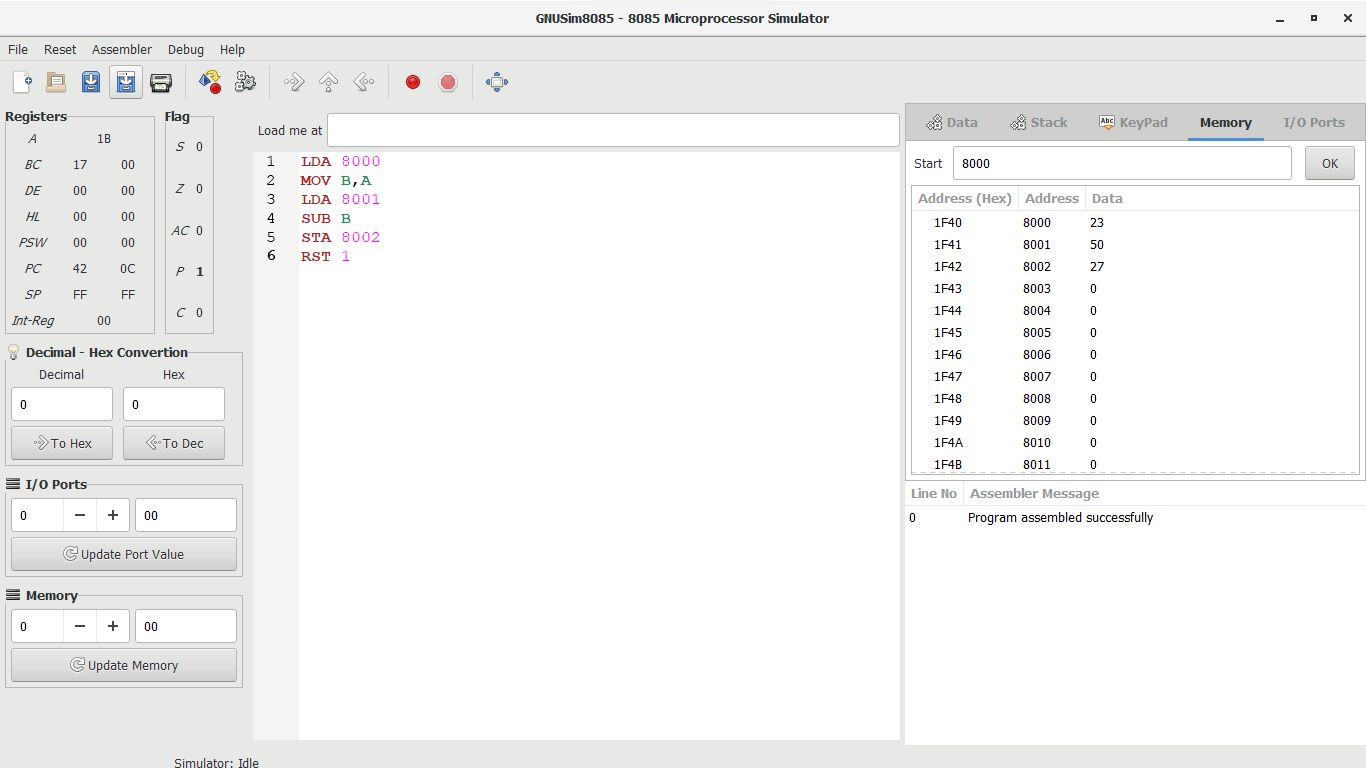
STA 8002

RST 1

**INPUT:**



**OUTPUT:**



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

**8-BIT MULTIPLICATION**

**EXP NO: 3**

**AIM:** To write an assembly language program to implement 8-bit multiplication using 8085 processor.

**ALGORITHM:**

1. Start the program by loading a register pair with the address of memory location.
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. Increment the value of the carry.
6. Check whether the repeated addition is over.
7. Store the value of product and the carry in the memory location.
8. Halt.

**PROGRAM:**

LDA 8500

MOV B, A

LDA 8501

MOV C, A

CPI 00

JZ LOOP

XRA A

LOOP1: ADD B

DCR C

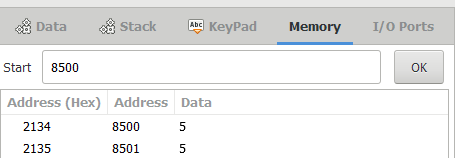
JZ LOOP

JMP LOOP1

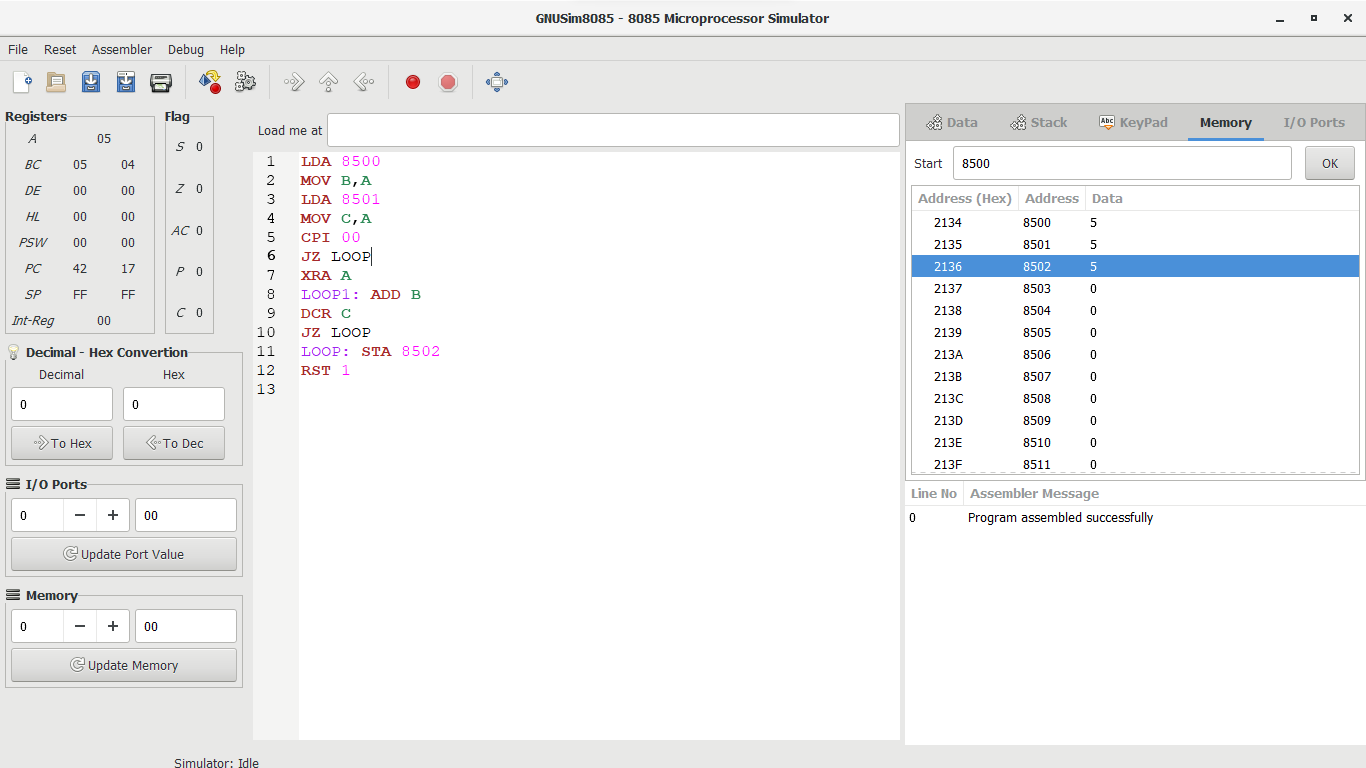
LOOP: STA 8502

RST 1

**INPUT:**

****

**OUTPUT:**



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

**8-BIT DIVISION**

**EXP NO: 4**

**AIM:** To write an assembly language program to implement 8-bit division using 8085 processor.

**ALGORITHM:**

1. Start the program by loading a register pair with the address of memory location.
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. Increment the value of the carry.
6. Check whether the repeated subtraction is over.
7. Store the value of quotient and the reminder in the memory location.
8. Halt.

**PROGRAM:**

LDA 8501

MOV B, A

LDA 8500

MVI C,00

LOOP:CMP B

JC LOOP1

SUB B

INR C

JMP LOOP

STA 8503

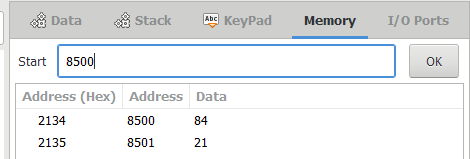
DCR C

MOV A, C

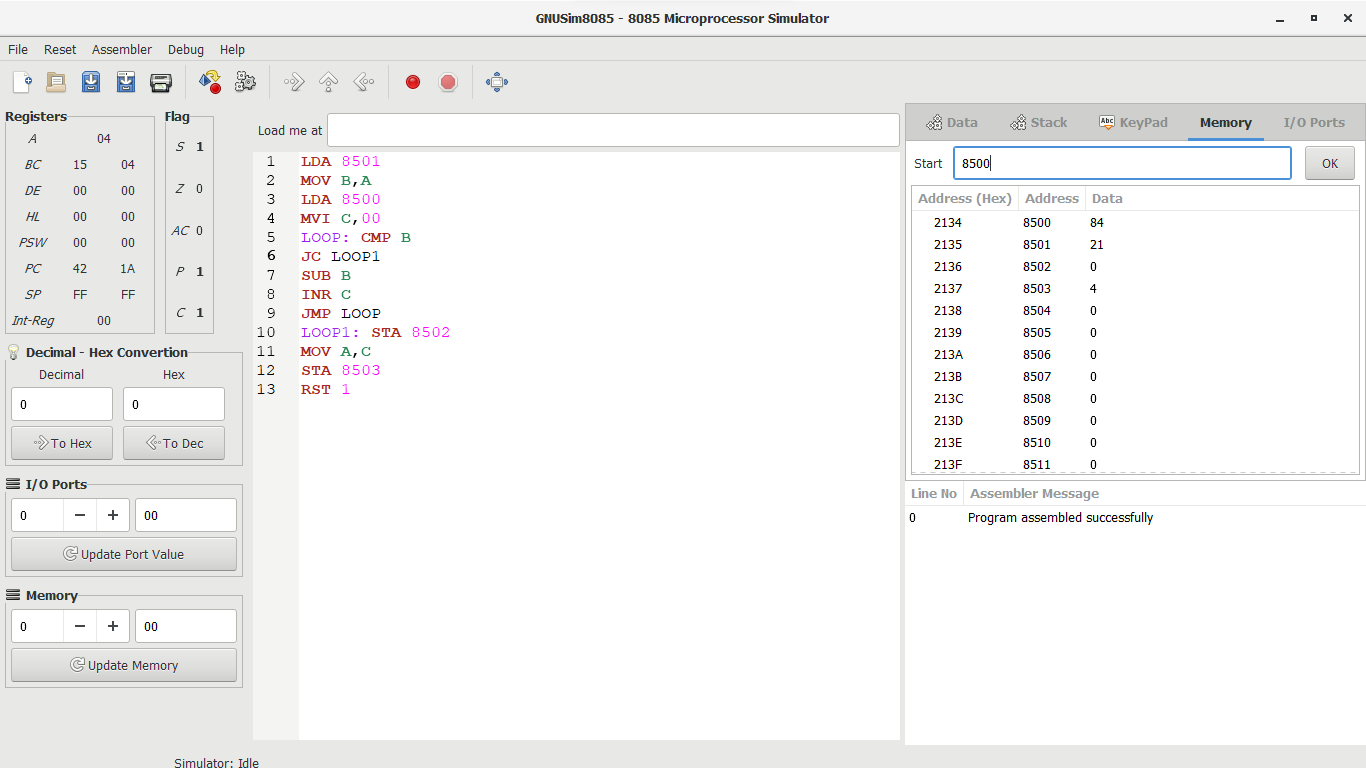
LOOP1: STA 8502

RST 1

**INPUT:**

****

**OUTPUT:**

****

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

**16-BIT ADDITION**

**EXP NO: 5**

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

**ALGORITHM:**

1) Start the program by loading a register pair with address of 1st number.

2)     Copy the data to another register pair.

3)      Load the second number to the first register pair.

4)      Add the two register pair contents.

5)      Store the result in memory locations.

6)      Terminate the program.

**PROGRAM:**

LHLD 2500

XCHG

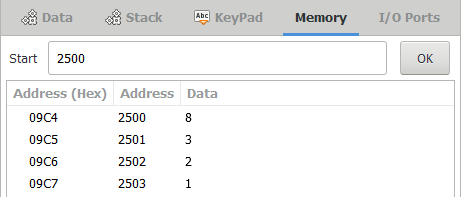
LHLD 2502

DAD D

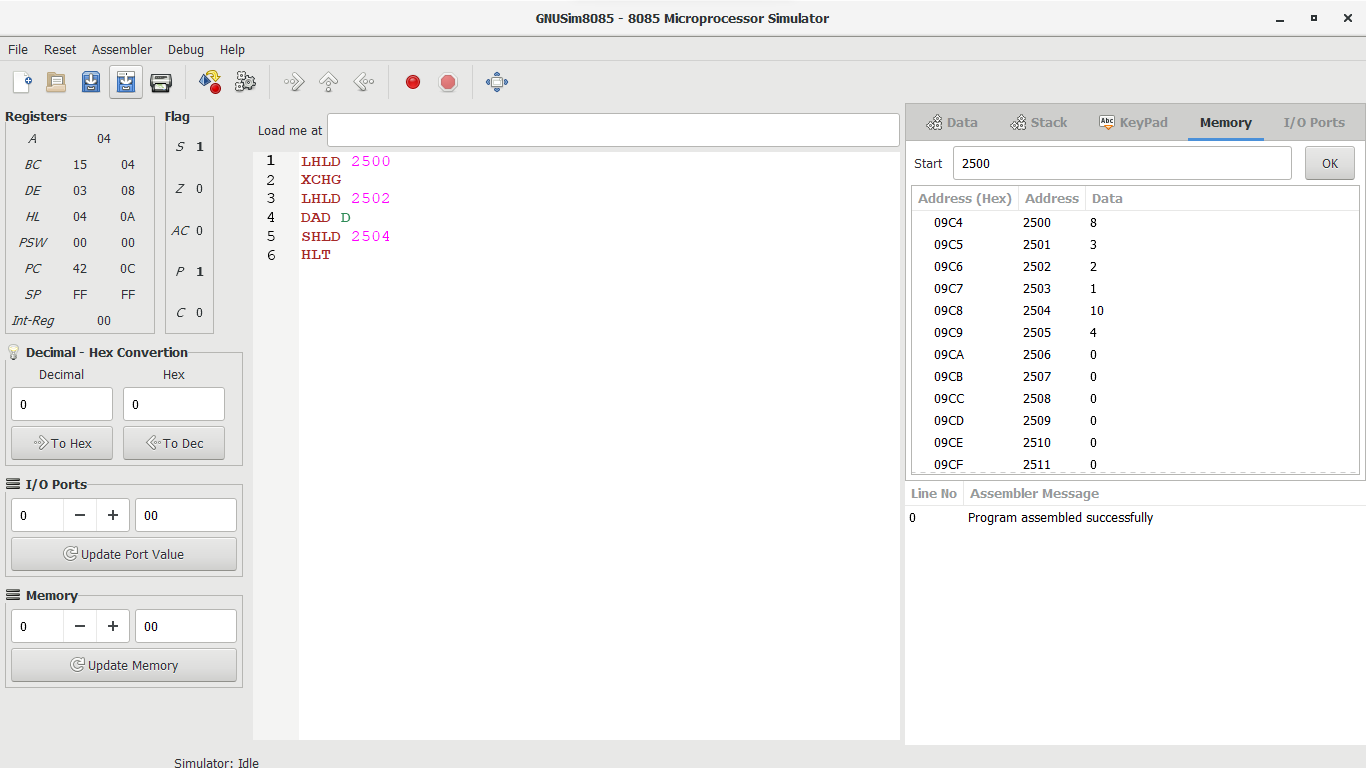
SHLD 2504

HLT

**INPUT:**

****

**OUTPUT:**

****

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

**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 1st number.

2)     Copy the data to another register pair.

3)      Load the second number to first registre 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

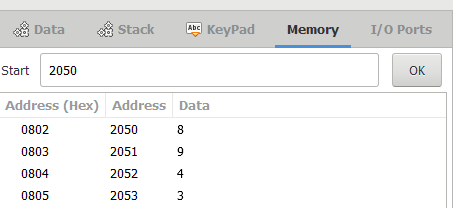
MOV A, D

SUB H

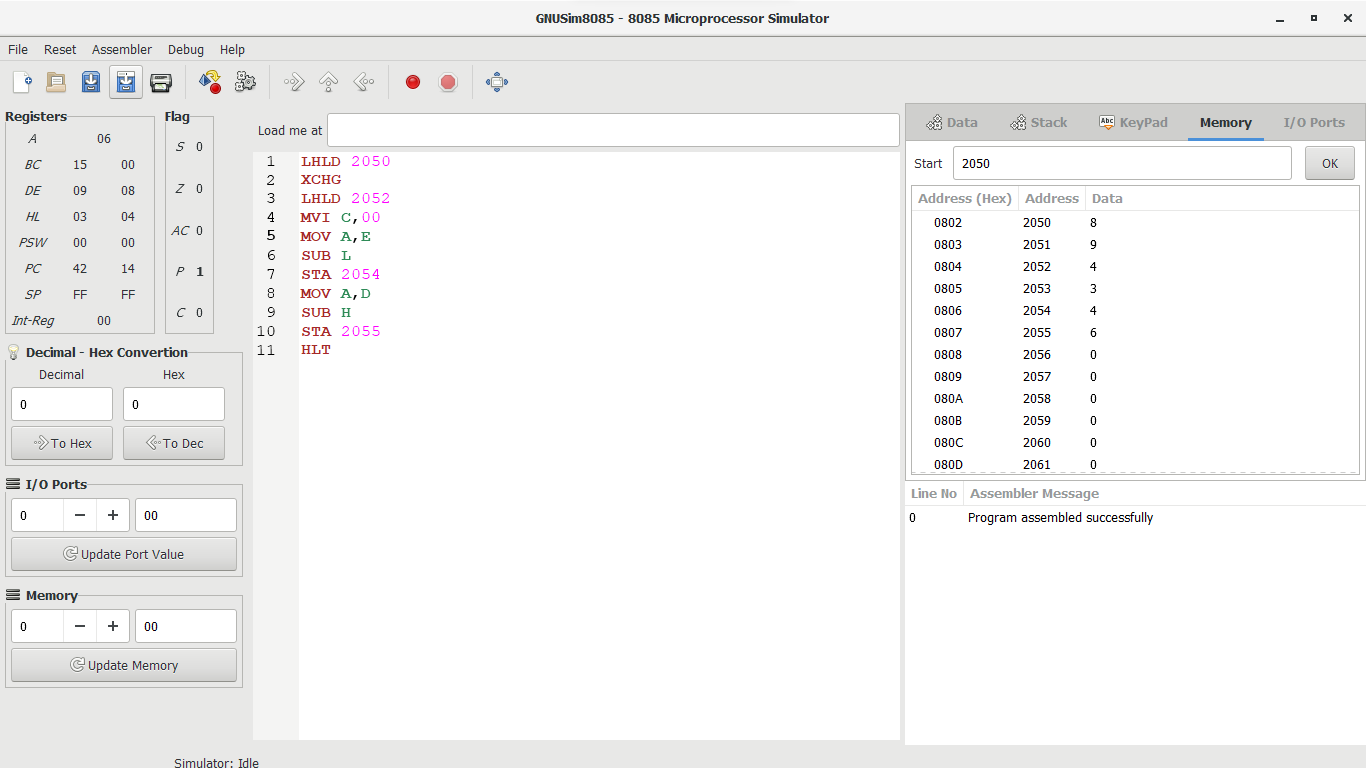
STA 2055

HLT

**INPUT:**

****

**OUTPUT:**

****

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

**16-BIT MULTIPLICATION**

**EXP NO: 7**

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

**ALGORITHM:**

1. Load the first data in HL pair.
2. Move content of HL pair to stack pointer.
3. Load the second data in HL pair and move it to DE.
4. Make H register as 00H and L register as 00H.
5. ADD HL pair and stack pointer.
6. Check for carry if carry increment it by 1 else move to next step.
7. Then move E to A and perform OR operation with accumulator and register D.
8. The value of operation is zero, then store the value else go to step 3.

**PROGRAM:**

LHLD 2050

SPHL

LHLD 2052

XCHG

LXI H,0000H

LXI B,0000H

AGAIN: DAD SP

JNC START

INX B

START: DCX D

MOV A,E

ORA D

JNZ AGAIN

SHLD 2054

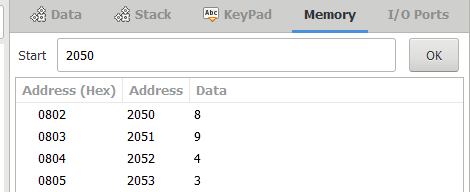
MOV L,C

MOV H,B

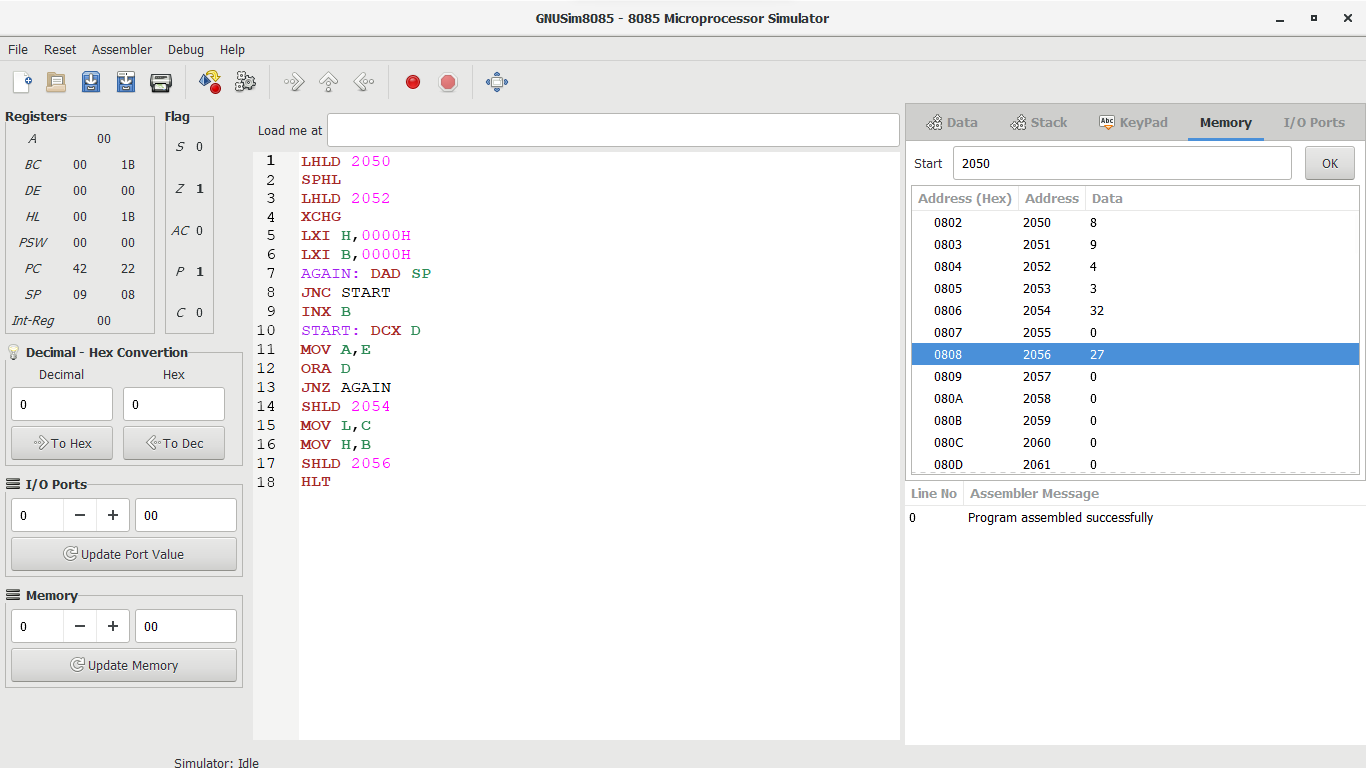
SHLD 2056

HLT

**INPUT:**

****

**OUTPUT:**

****

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

**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 <- 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 <- 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

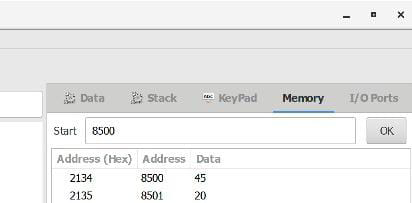
DCR C

MOV A,C

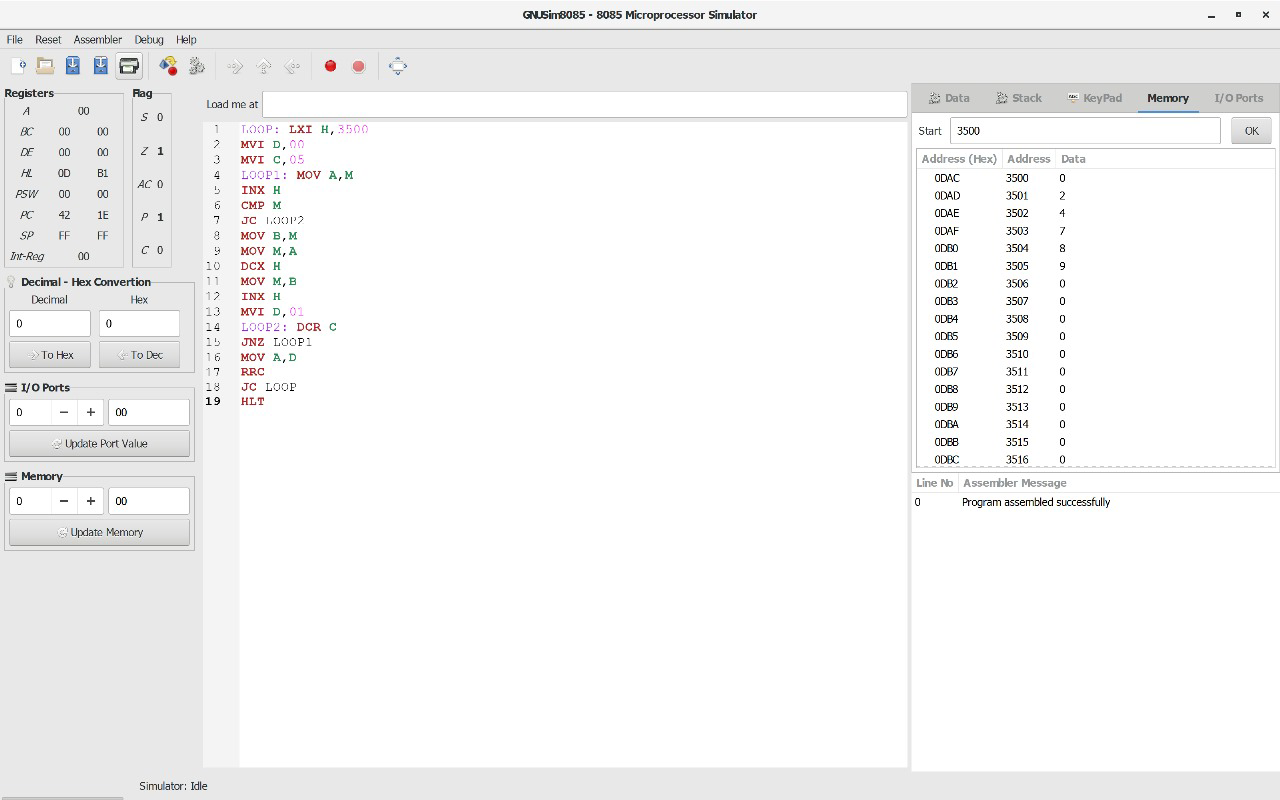
LOOP1: STA 8502

RST 1

**INPUT:**

****

**OUTPUT:**

****

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