Syeda Reeha Quasar

14114802719

4C7

Aim

To demonstrate all the data transfer instructions of GNUSim 8085 microprocessor.

Experiment - 3

Computer Organization and Architecture

# **EXPERIMENT – 2**

## **Aim:**

To demonstrate all the data transfer instructions of GNUSim 8085 microprocessor.

## **Theory:**

Instructions Used-

* **MOV** (Copy from source to destination) - This instruction copies the contents of the source register into the destination register; the contents of the source register are not altered. If one of the operands is a memory location, its location is specified by the contents of the HL registers. Example: MOV B, C or MOV B, M
* **INX** SP instruction is used to increment the SP contents by 1. INX SP instruction is a special case of INX rp instruction which increases the content of the register pair. This instruction occupies only 1-Byte in memory.
* **ADD** (Add register or memory to accumulator) - The contents of the operand (register or memory) are added to the contents of the accumulator and the result is stored in the accumulator. If the operand is a memory location, its location is specified by the contents of the HL registers. All flags are modified to reflect the result of the addition. Example: ADD B or ADD M
* **LXI** (Load register pair immediate) - The instruction loads 16-bit data in the register pair designated in the operand. Example: LXI H, 2034H or LXI H, XYZ
* **LDA** or Load The accumulator loads the contents of a memory location, specified by a 16-bit address in the operand, are copied to the accumulator. Example − LDA 2034K
* **STA** - The contents of the accumulator are copied into the memory location specified by the operand. This is a 3-byte instruction, the second byte specifies the low-order address and the third byte specifies the high-order address. Example − STA 325K
* **IN** - Input data to accumulator from a port with 8-bit address. The contents of the input port designated in the operand are read and loaded into the accumulator. Example − IN5KL.
* **OUT** - The contents of the accumulator are copied into the I/O port specified by the operand. Example − OUT K9L.
* **LHLD** - The instruction copies the contents of the memory location pointed out by the address into register L and copies the contents of the next memory location into register H. Example − LHLD 3225K
* **SHLD** - The contents of register L are stored in the memory location specified by the 16-bit address in the operand and the contents of H register are stored into the next memory location by incrementing the operand. This is a 3-byte instruction, the second byte specifies the low-order address and the third byte specifies the high-order address. Example − SHLD 3225K
* **LDAX** - The contents of the designated register pair point to a memory location. This instruction copies the contents of that memory location into the accumulator. Example − LDAX K
* **PCHL** - it loads the program counter with HL data. the content of H placed into higher order byte and L placed at low order bytes.
* **STAX(Store accumulator indirect): -** The contents of the accumulator are copied into the memory location specified by the contents of the operand (register pair). The contents of the accumulator are not altered. Eg: - STAX B (the content of accumulator is stored into the memory location specified by the BC register pair.)
* **MVI** (Move immediate 8-bit) - The 8-bit data is stored in the destination register or M, data memory. If the operand is a memory location, its location is specified by the contents of the HL registers. Example: MVI B, 57H or MVI M, 57H
* **XCHG** - The contents of register H are exchanged with the contents of register D, and the contents of register L are exchanged with the contents of register E. Example – XCHG
* **STA** - The contents of the accumulator are copied into the memory location specified by the operand. This is a 3-byte instruction, the second byte specifies the low-order address and the third byte specifies the high-order address. Example − STA 325K

# **Source Codes:**

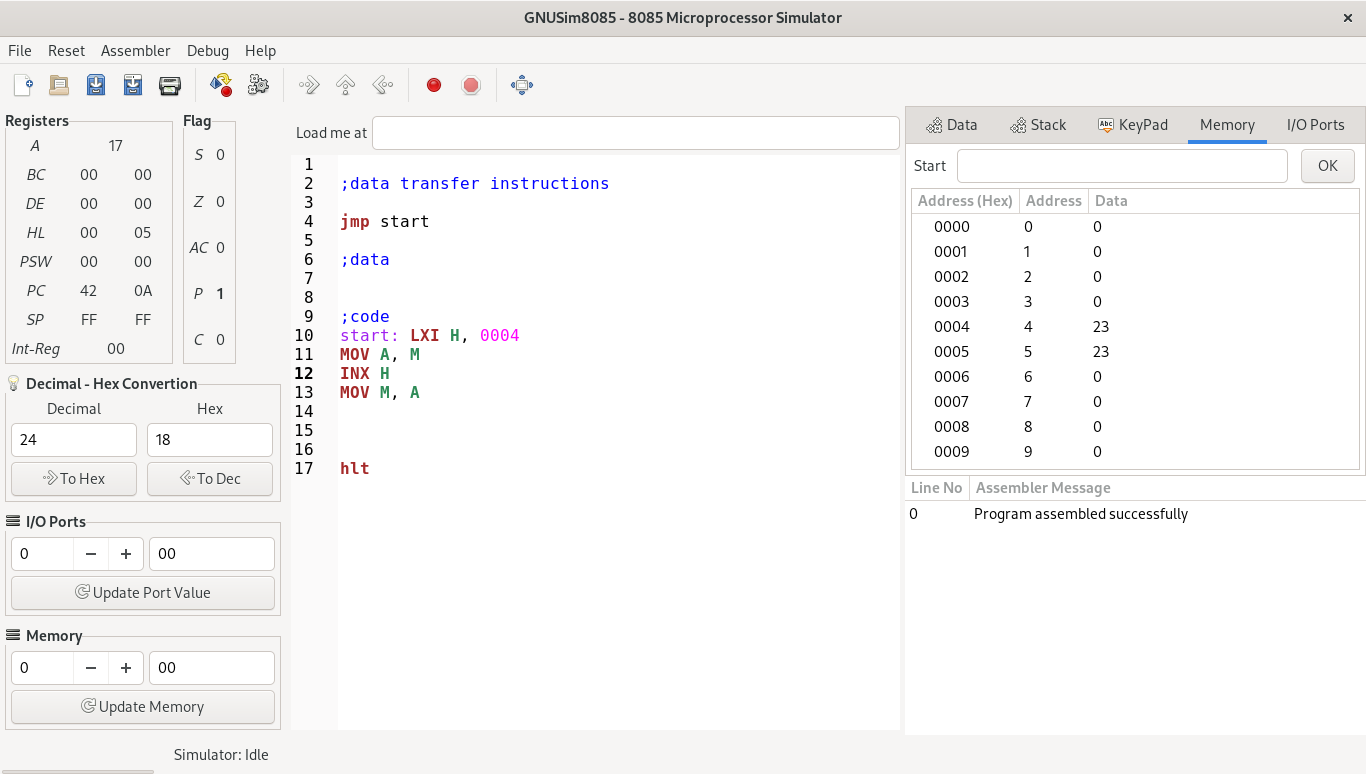
## MOV

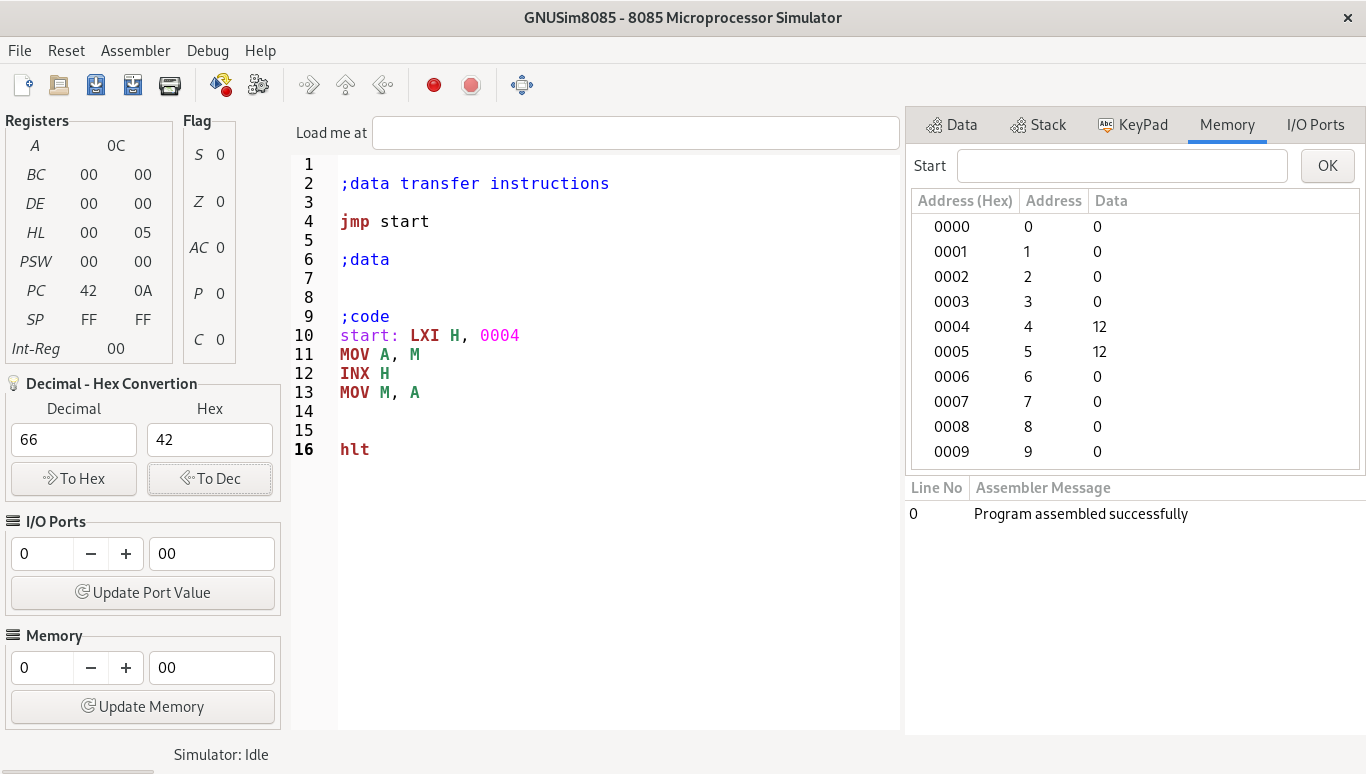
### Source Code

;data transfer instructions  
  
jmp start  
  
;data  
  
  
;code  
start: LXI H, 0004  
MOV A, M  
INX H  
MOV M, A

hlt

## **Output:**



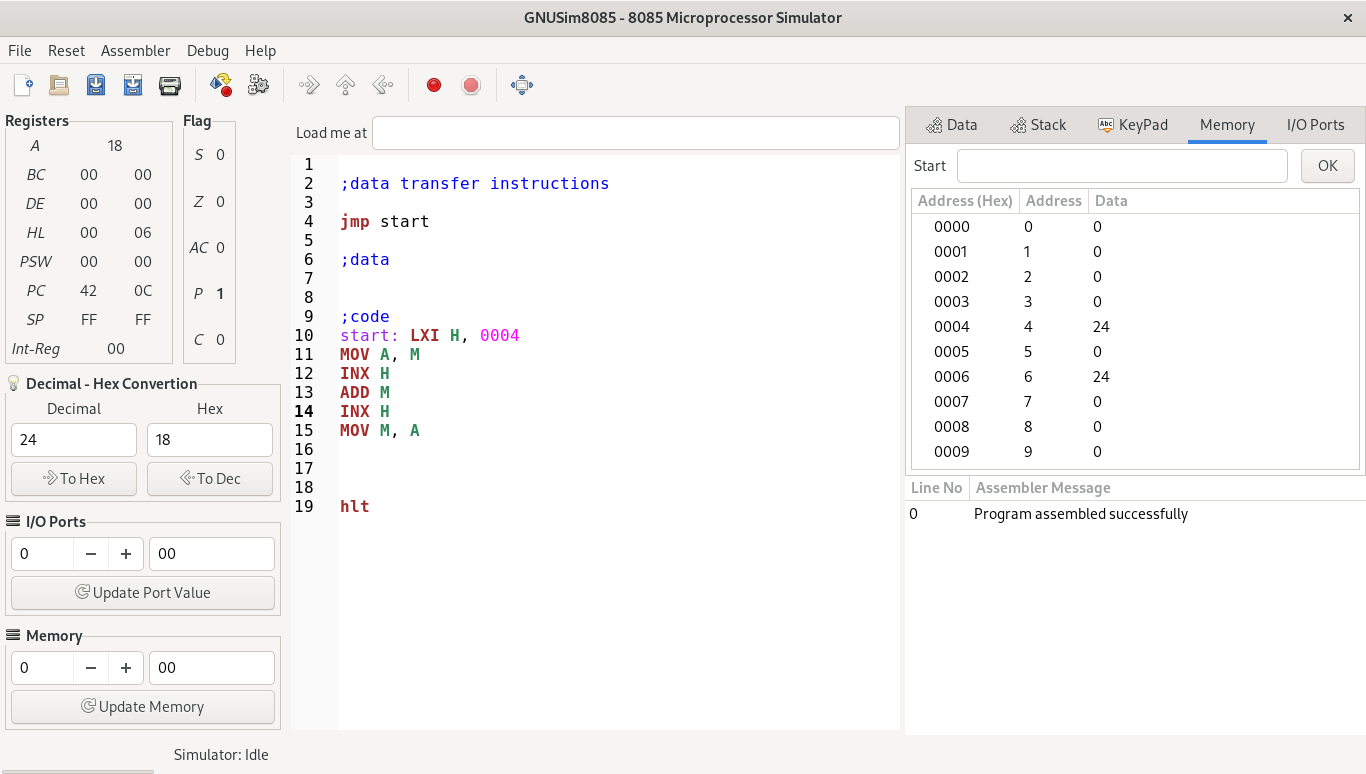


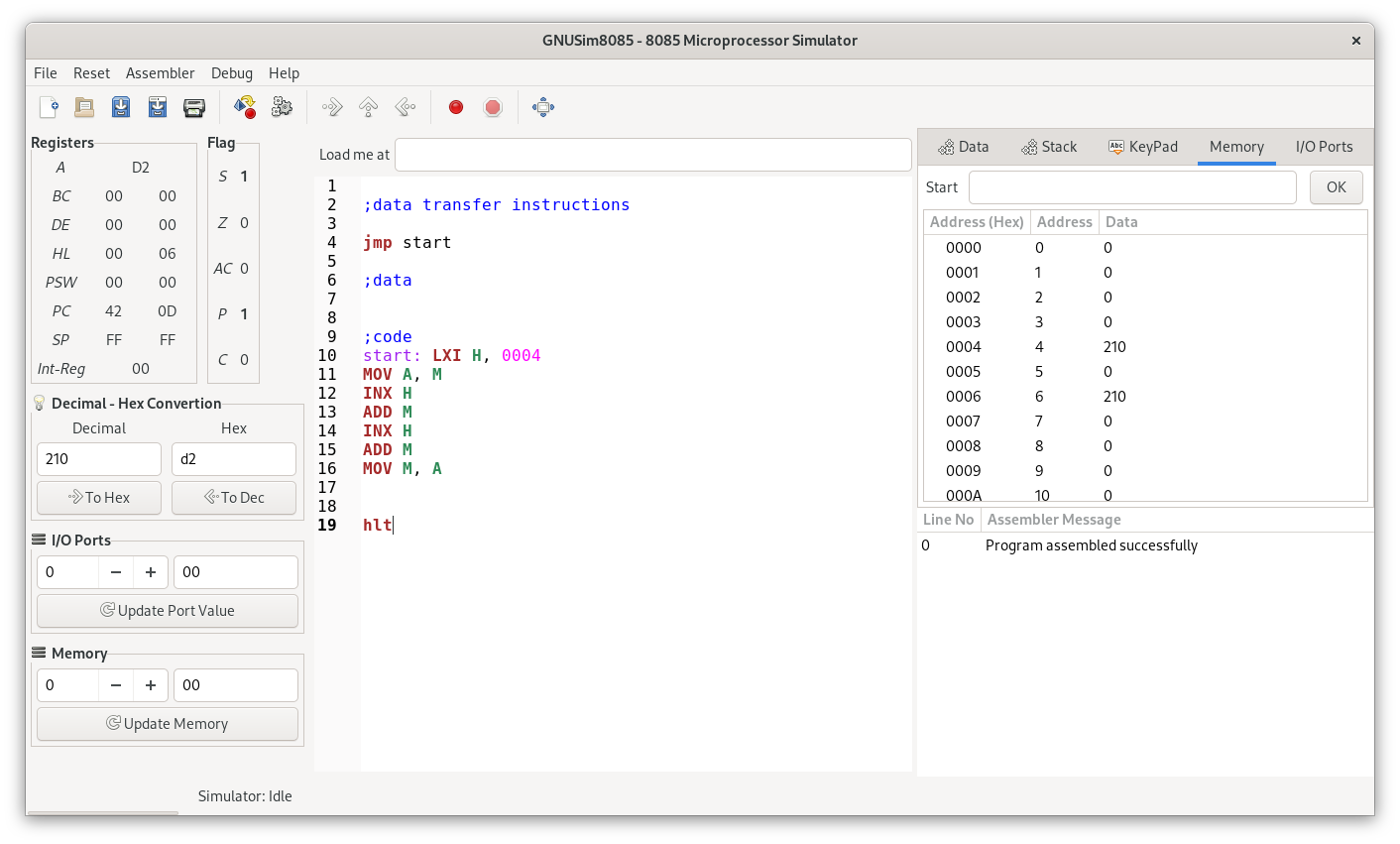
## Memory Transfer and LXI and ADD

### Source Code

;data transfer instructions  
  
jmp start  
  
;data  
  
  
;code  
start: LXI H, 0004  
MOV A, M  
INX H  
ADD M  
INX H  
ADD M  
MOV M, A  
  
  
hlt

## **Output:**





## Memory Transfer and LXI and ADD

### Source Code

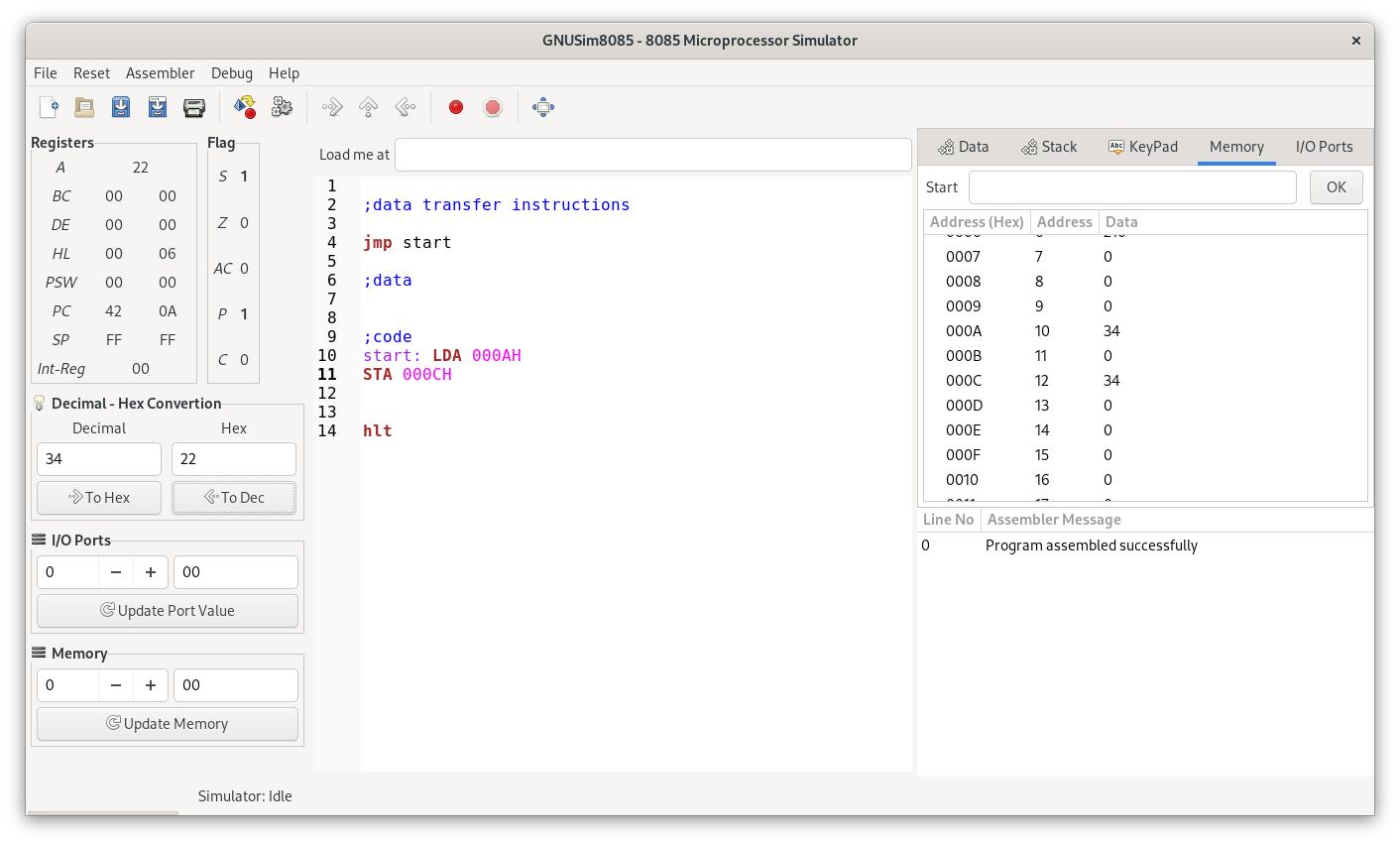
;LDA -> ACC TO DATA

;STA A TO MEMORY

;MEMORY OPERATIONS

;data transfer instructions  
  
jmp start  
  
;data  
  
  
;code  
start: LDA 000AH  
STA 000CH  
  
  
hlt

## **Output:**



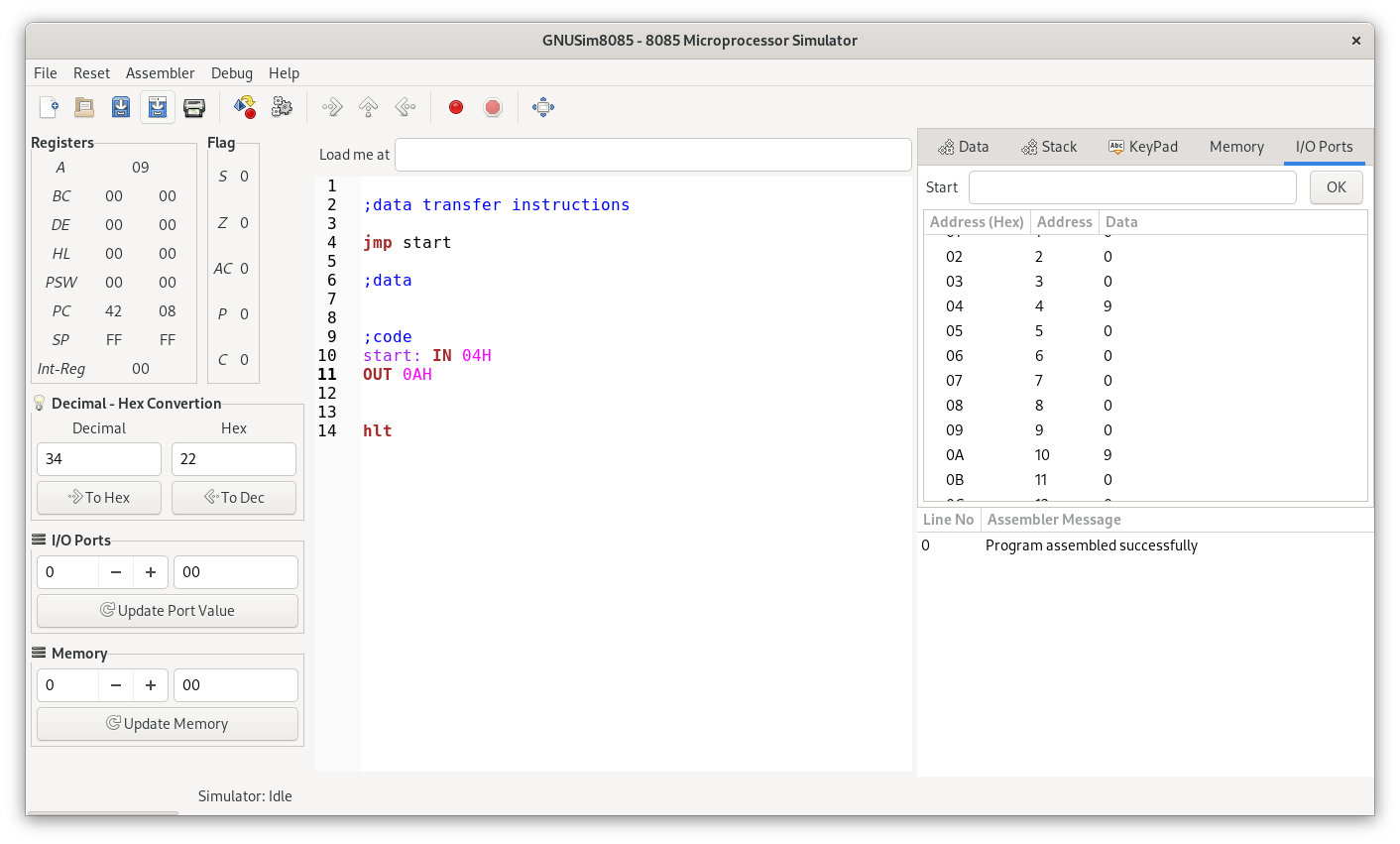
## I/P TO MEMORY COMMUNICATION , IN and OUT instructions

### Source Code

;I/P TO MEMORY COMMUNICATION

;data transfer instructions  
  
jmp start  
  
;data  
  
  
;code  
start: IN 04H  
OUT 0AH  
  
  
hlt

## **Output:**

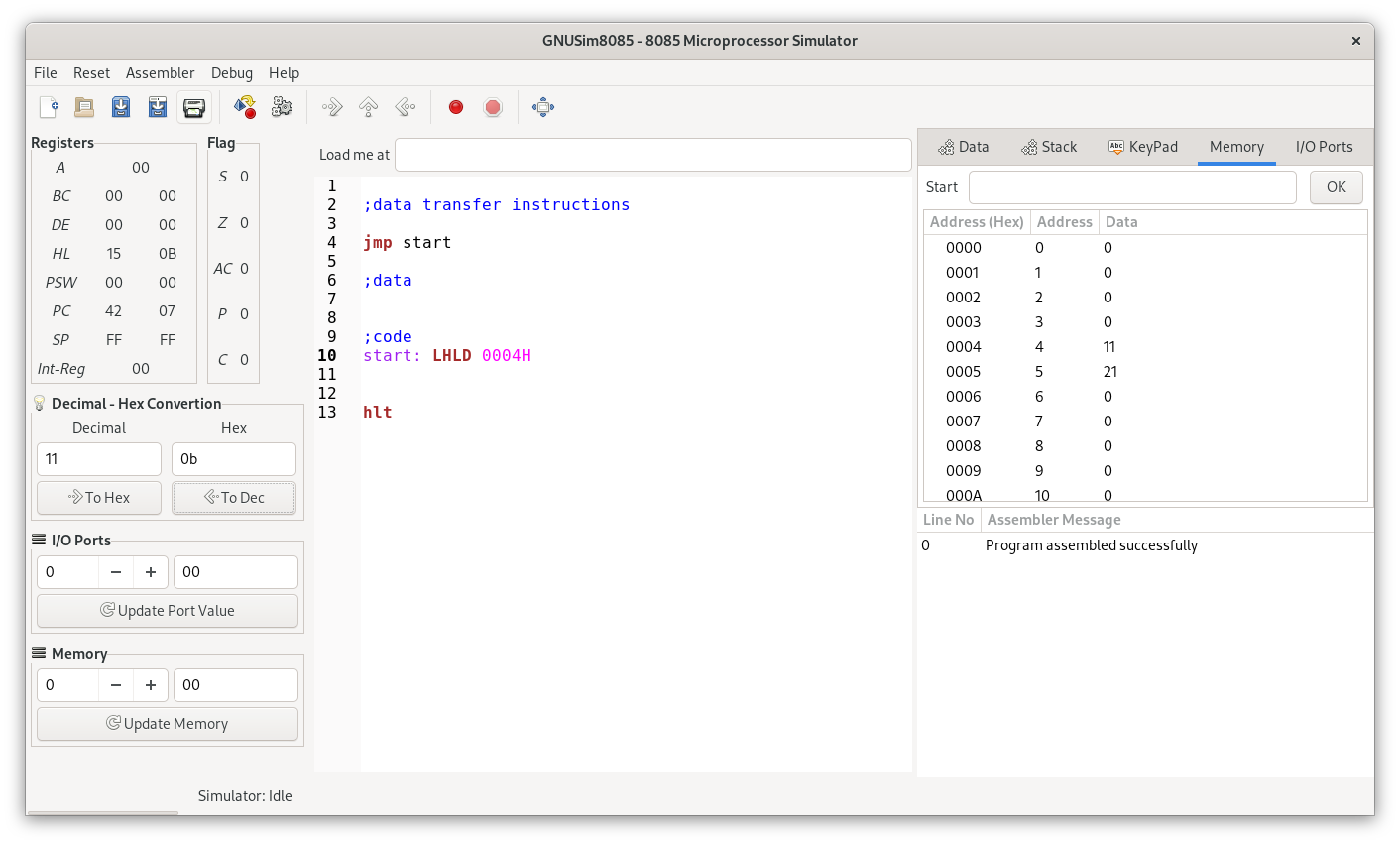


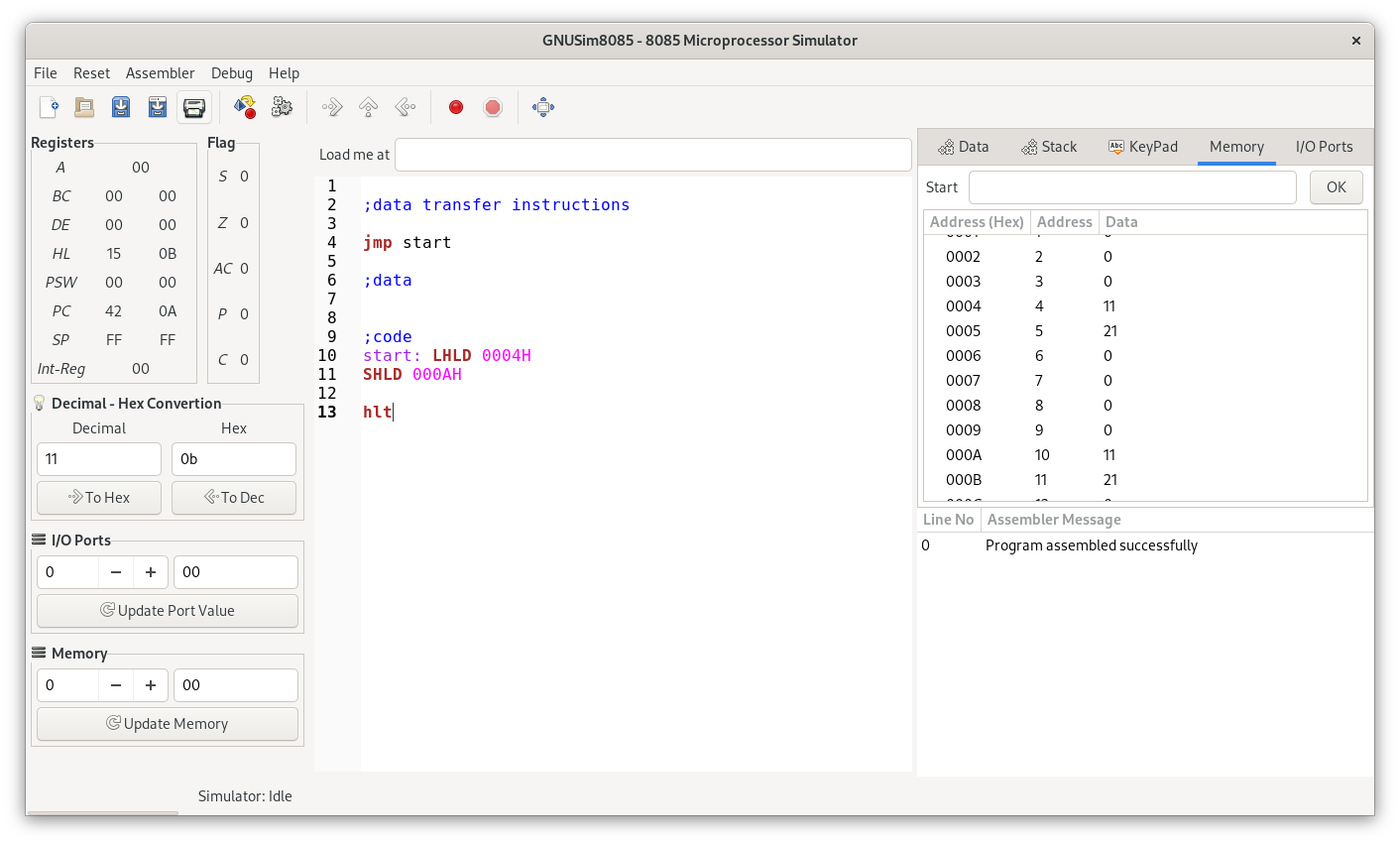
## LHLD and SHLD instructions

### Source Code

;data transfer instructions  
  
jmp start  
  
;data  
  
  
;code  
start: LHLD 0004H  
SHLD 000AH  
  
hlt

## **Output:**





## PCHL instruction

### Source Code

;data transfer instructions  
  
jmp start  
  
;data  
  
  
;code  
start: LXI H, 0004H  
PCHL  
  
hlt

## **Output:**

