

# INDEX

[illegible]

# **Lab Report : 1**

**TITLE** : DATA TRANSFER INSTRUCTION

**OBJECTIVES** :

The main objective of this lab report is to transfer data.

**MATERIALS REQUIRED** :

GNU 8085 Simulator and a Computer .

**THEORY** :

Data transfer instructions are such instructions that copies data from source to destination in which data will not be changed.

**Types of data transfer instructions :**

- 1) Between registers.
- 2) Between register and memory
- 3) Between peripheral and processor register or accumulator.

4) Immediate data to register or register pair

### **Some Instructions:**

#### **a) MOV Rd, Rs**

This instruction copies the contents of the source register into the destination register, the content of the source register are not altered. If one of the operands is a memory location, it is specified by the contents of HL registers.

#### **b) MVI Rd, 8bit data**

The 8 bit data are stored in the destination register or memory if the operand is a memory location, it is specified by the contents of HL registers.

#### **c) LXI Rp, 16 bit data**

The instruction loads 16 bit data in the register pair designated in the operands. This is a 3 byte instructions, the second byte specified the low order byte and the third byte specified the high order byte

#### **d) IN port address**

The contents of the input port designated in the operands are read and loaded into the accumulator.

#### **e) OUT port address**

The contents of the accumulators are copied into the output port specified by the operands .

**f) LDA 16 bit address**

The contents of the memory locations, specified by a 16 bit address in the operand, are copied to the accumulator. The contents of the source are not altered. This is a 3 byte instructions, the second byte specified the low order address and the third byte specifies the high order address

**g) LDAX B/D Rp**

The contents of the designated register pair point to a memory location. This instruction copies the contents of that memory location into the accumulator. The contents of either the register pair or the memory location are not altered.

**h) STA 16 bit**

The contents of the accumulator are copied to a memory location specified by the operands. This is a three byte instructions, the second byte specifies the low order address and the third byte specified the high order address.

**i) STAX B/D Rp**

The contents of the accumulator are copied to the memory location specified by the contents of operands

(Register pairs). The contents of the accumulator are not altered.

#### **j) LHLD 16 bit address**

The instruction copies the content of the memory location pointed out by 16 bit address in register L and copies the contents of the next memory location in register H. The content of source memory location are not altered.

#### **k) SHLD 16 bit address**

The contents of the register L are stored in the memory location specified by the 16 bit address in the operands, and the contents of H register are stored in the next memory location by incrementing the operands. The contents of registers HL are not altered. This is a three byte instructions, the second byte specifies the low order address and the third byte specifies the high order address.

#### **l) XCHG**

The contents of register H are exchanged with the contents of register D, And the contents of the register L are exchanged with the contents of the register E

**HLT :**

The MPU finishes executing the current instruction and halts any further execution. The MPU enters the halts acknowledge machine cycle and wait states are inserted in every clock period. The address and the data bus are placed in the high impedance state. The contents of the registers are unaffected during the HLT state. An Interrupt or reset is necessary to exit from the Halt state.

## **LAB EXPERIMENT :**

**#1** Write an instruction to move 78h data to Accumulator

**<source code>**

MVI A, 78H

HLT

**#2** Write an instruction to move 69h data to Accumulator and copy to register B.

**<source code>**

MVI A, 69H

MOV B, A

HLT

**#3** Write an instruction to store FE26H in B register pair.

**<source code>**

LXI B, 0FE26H

HLT

**#4** Write an instruction to store the data of port address 02h to accumulator.

**<source code>**

IN 02H

HLT

(**Note** : we should first write the decimal data in I/O port 02h before running the command)

**#5** Write an instruction to transfer 62H data to the I/O port address 04h.

**<source code>**

MVI A, 62H

OUT 04H

HLT

**#6** Write an instruction to copy the data of memory location 0002H to accumulator.

**<source code>**

LDA 0002H

HLT

(**Note** : we should first write the decimal data in memory location 0002h before running the command)

**#7** Write an instruction to copy the data of address stored in B register pair to accumulator.

**<source code>**

LXI B, 0002H

LDAX B

HLT



**#8** Write an instruction to load FEH data to accumulator and stored it in 0004H memory address.

**<source code>**

MVI A, 0FEH

STA 0004H

HLT

**#9** Write an instruction to load the data FEH in accumulator and store the address 0004H in D register pair and move the data to the address stored in the register pair.

**<source code>**

MVI A, 0FEh

LXI D, 0004h

STAX D

HLT

**#10** Write an instruction to copy the data of memory address 0004H and 0005H to HL register pair.

**<source code>**

LHLD 0004H

HLT

**#11** Write an instruction to load the data F326H in HL register pair and copy the data to the memory address 0006H and 0007H.

**<source code>**

LXI H, 0F326H

SHLD 0006H

HLT

**#12** Write an instruction to load the data 2676H and 5768H to H and D register pair respectively and exchange the data between H & D register pair.

**<source code>**

LXI H, 2676H

LXI D, 5768H

XCHG

HLT

**CONCLUSION :**

Hence, we are able to know about data transfer instructions.