

23VLS1401: Microcontroller and Computer architecture  
Lecture 1 (U2)

**Data Transfer instructions and  
programming for Microprocessor 8085**

A presentation by  
Dr. Shubhangi Rathkanthiwar  
Professor



Department of Electronics Engineering, YCCE, Nagpur, India

# Session objectives

---

- To learn various software instructions related to the process of data transfer
- To learn various addressing modes related to the process of data transfer
- To develop the programming technique in assembly language for given problem statement, store the source data, execute the program and observe the result in destination register or memory location.



# Data Transfer instructions

- MOV
- MVI
- LDA
- STA
- LDAX
- STAX
- LXI
- LHLD
- SHLD
- XCHG

# Data transfer instructions

Opcode	Operand	Meaning	Explanation	Addressing mode
<b>MOV</b>	<b>Rd,Rs</b> <b>Rd,M</b> <b>M,Rs</b>	<b>Copy the 8 bit data from the source to the destination</b>	<b>This instruction copies the contents of the source register into the destination register without any alteration.</b>	<b>Register addressing mode</b>

- ❑ M specifies a memory location whose address is specified by the HL pair
- ❑ Memory to Memory Data Transfer is not possible
- ❑ Rd is the destination register and Rs is the source register. Registers A, B, C,D,E, H and L can act as Source/Destination registers

# Examples of MOV instruction:

## MOV A,B

Before Execution of the instruction

A	11
B	22

After Execution of the instruction

A	22
B	22

MOV A,B



# Examples of MOV instruction:

## MOV A,M

Before Execution of the instruction

After Execution of the instruction

H		25		L		01	
A	11	MOV A,M		A	7F		
2501	7F			2501	7F		

# Examples of MOV instruction:

## MOV M,B

Before Execution of the instruction

After Execution of the instruction

H		25	L	01
B	22	MOV M,B	B	22
2501	7F		2501	22

# Data transfer instructions

Opcode	Operand	Meaning	Explanation	Addressing mode
MVI	R/M, 8 bit data	Move the 8 bit data to the destination register or memory location	This instruction copies the 8 bit data into the destination register or memory location whose address is given in HL pair	Immediate addressing mode



# Examples of MVI instruction:

## MVI A,7FH

---

Before Execution of the instruction

A	11
---	----

After Execution of the instruction

A	7F
---	----

MVI A,7FH

# Examples of MVI instruction:

## MVI M,11H

Before Execution of the instruction

H	25	L	01
---	----	---	----

2501	7F
------	----

MVI M,11H

After Execution of the instruction

2501	11
------	----

# Examples of MVI instruction:

## MOV B,66H

Before Execution of the instruction

B	22
---	----

MVI B,66H

After Execution of the instruction

B	66
---	----



# Data transfer instructions

Opcode	Operand	Meaning	Explanation	Addressing mode
LXI	Rp, 16 bits data (H – HL) (B – BC) (D – DE)	Load Register pair immediate	This instruction loads the register pair with 16 bits data	Immediate addressing mode

# Examples of LXI instruction:

## LXI H,2501H

Before Execution of the instruction

H	11
---	----

L	22
---	----

LXI H,2501H

After Execution of the instruction

H	25
---	----

L	01
---	----

# Examples of LXI instruction:

## LXI B,2601H

Before Execution of the instruction

B	11
---	----

C	22
---	----

LXI B,2601H

After Execution of the instruction

B	26
---	----

C	01
---	----



# Examples of LXI instruction:

## LXI D,2501H

Before Execution of the instruction

D	11
---	----

E	22
---	----

LXI D,2501H

After Execution of the instruction

D	25
---	----

E	01
---	----

# Data transfer instructions

Opcode	Operand	Meaning	Explanation	Addressing mode
LDA	16 bits address	Load the 8 bit data from the memory location to Accumulator	This instruction copies the contents of the memory location whose address is specified in the instruction, to the Accumulator	Direct addressing mode

# Example of LDA instruction: LDA 2501H

Before Execution of the instruction

A	11
---	----

2501	22
------	----

LDA 2501H

After Execution of the instruction

A	22
---	----

2501	22
------	----



# Data transfer instructions

Opcode	Operand	Meaning	Explanation	Addressing mode
STA	16 bits address	Store the Accumulator in memory location	This instruction copies the contents of the Accumulator to the memory location whose address is specified in the instruction	Direct addressing mode

# Example of STA instruction:

## STA 2501H

Before Execution of the instruction

A	11
---	----

2501	22
------	----

STA 2501H

After Execution of the instruction

A	11
---	----

2501	11
------	----

# Data transfer instructions

Opcode	Operand	Meaning	Explanation	Addressing mode
LDAX	Rp (B – BC) (D – DE)	Load Accumulator indirect	This instruction copies the contents of the memory location whose address is specified in the register pair, to the Accumulator	Indirect addressing mode



# Example of LDAX instruction: LDAX B

Before Execution of the instruction

BC	2501
----	------

A	11
---	----

2501	22
------	----

LDAX B

After Execution of the instruction

BC	2501
----	------

A	22
---	----

2501	22
------	----

# Data transfer instructions

Opcode	Operand	Meaning	Explanation	Addressing mode
STAX	Rp (B – BC) (D – DE)	Store Accumulator indirect	This instruction copies the contents of the Accumulator to the memory location whose address is specified in the register pair	Indirect addressing mode

# Example of STAX instruction: STAX B

Before Execution of the instruction

BC	2501
----	------

A	11
---	----

2501	22
------	----

**STAX B**

After Execution of the instruction

BC	2501
----	------

A	11
---	----

2501	11
------	----



# Data transfer instructions

Opcode	Operand	Meaning	Explanation	Addressing mode
LHLD	16 bits address	Load H and L registers direct	<p>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.</p> <p>This is a 3-byte instruction, the second byte specifies the low-order address and the third byte specifies the high-order address.</p>	Direct addressing mode

# Example of LHLD instruction: LHLD 2501H

Before Execution of the instruction

H	XX	L	XX
---	----	---	----

2501	11
------	----

2502	22
------	----

After Execution of the instruction

H	22	L	11
---	----	---	----

2501	11
------	----

2502	22
------	----

LHLD 2501H

# Data transfer instructions

Opcode	Operand	Meaning	Explanation	Addressing mode
SHLD	16 bits address	Store H and L registers direct	<p>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.</p> <p>This is a 3-byte instruction, the second byte specifies the low-order address and the third byte specifies the high-order address.</p>	Direct addressing mode



# Example of SHLD instruction: SHLD 2501H

Before Execution of the instruction

H	44	L	55
---	----	---	----

After Execution of the instruction

H	44	L	55
---	----	---	----

2501	11
------	----

2502	22
------	----

SHLD 2501H

2501	55
------	----

2502	44
------	----

# Data transfer instructions

Opcode	Operand	Meaning	Explanation	Addressing mode
XCHG		Exchange H and L with D and E	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.	Implicit addressing mode

# Example of XCHG instruction

## XCHG

Before Execution of the instruction

H	44	L	55
---	----	---	----

After Execution of the instruction

H	11	L	22
---	----	---	----

XCHG

D	11	E	22
---	----	---	----

D	44	E	55
---	----	---	----



# Programming Examples

**Problem Statement 1:** Write an Assembly language Program to load the data bytes 55H, 66H, 77H, 88H and 99H in Accumulator, Registers B, C, D and E and 16 bit data 1122H in HL pair.

---

MVI A,55H

MVI B,66H

MVI C,77H

MVI D,88H

MVI E,99H

LXI H,1122H

HLT

# Before and after execution

This screenshot shows the debugger's state before execution. The left sidebar contains icons for CPU, I/O, and other components. The main window is divided into two sections: 'Registers' and 'Flags'. The 'Registers' section lists A/PSW, BC, DE, HL, SP, and PC with their current values. The 'Flags' section lists S, Z, and AC with checkboxes. The instruction list on the right shows the program code.

Register	Value
A/PSW	0x 00 02
BC	0x 00 00
DE	0x 00 00
HL	0x 00 00
SP	0x FF FF
PC	0x 00 0E

Flag	Status
S	<input type="checkbox"/>
Z	<input type="checkbox"/>
AC	<input type="checkbox"/>

Instruction List:

- 1 MVI A,55H
- 2 MVI B,66H
- 3 MVI C,77H
- 4 MVI D,88H
- 5 MVI E,99H
- 6 LXI H,1122H
- 7 HLT

This screenshot shows the debugger's state after execution. The registers and flags have been updated. A tooltip 'Load & Run Ctrl + F5' is visible over the instruction list. The instruction list shows the program code.

Register	Value
A/PSW	0x 55 02
BC	0x 66 77
DE	0x 88 99
HL	0x 11 22
SP	0x FF FF
PC	0x 00 0E

Flag	Status
S	<input type="checkbox"/>
Z	<input type="checkbox"/>
AC	<input type="checkbox"/>

Instruction List:

- 1 MVI A,55H
- 2 MVI B,66H
- 3 MVI C,77H
- 4 MVI D,88H
- 5 MVI E,99H
- 6 LXI H,1122H
- 7 HLT

# Programming Examples

**Problem Statement 1:** Write an Assembly language Program to load the data bytes 55H, 66H, 77H, 88H and 99H in Accumulator, Registers B, C, D and E and 16 bit data 1122H in HL pair.

---

MVI A,55H

~~MVI B,66H~~

LXI B,6677H

~~MVI C,77H~~

~~MVI D,88H~~

LXI D,8899H

~~MVI E,99H~~

LXI H,1122H

HLT



# Before and after execution

The screenshot shows the 8086 emulator interface before execution. The assembly code window contains five lines: `1 MVI A,55H`, `2 LXI B,6677H`, `3 LXI D,8899H`, `4 LXI H,1122H`, and `5 HLT`. The registers window shows the following values: A/PSW (0x0002), BC (0x0000), DE (0x0000), HL (0x0000), SP (0xFFFF), and PC (0x0017). The flags window shows all flags (S, Z, AC, P) as unchecked.

Registers	Value
A/PSW	0x0002
BC	0x0000
DE	0x0000
HL	0x0000
SP	0xFFFF
PC	0x0017

Flags	Status
S	<input type="checkbox"/>
Z	<input type="checkbox"/>
AC	<input type="checkbox"/>
P	<input type="checkbox"/>

The screenshot shows the 8086 emulator interface after execution. The assembly code window remains the same. The registers window shows the updated values: A/PSW (0x5502), BC (0x6677), DE (0x8899), HL (0x1122), SP (0xFFFF), and PC (0x000C). The flags window shows all flags (S, Z, AC, P) as unchecked. A tooltip for the 'Load & Run' button (Ctrl + F5) is visible.

Registers	Value
A/PSW	0x5502
BC	0x6677
DE	0x8899
HL	0x1122
SP	0xFFFF
PC	0x000C

Flags	Status
S	<input type="checkbox"/>
Z	<input type="checkbox"/>
AC	<input type="checkbox"/>
P	<input type="checkbox"/>

Load & Run Ctrl + F5

# Instructions for incrementing contents of Register/Memory and Register pair

Opcode	Operand	Meaning	Explanation	Addressing mode
INR	R M	Increment R/M by 1	This instruction increments 8 bit register or memory location by 1	Register addressing mode
INX	Rp H – HL pair B – BC pair D – DE pair	Increment contents of register pair by 1	This instruction increments 16 bits data in register pair by 1, i.e. lower order register is incremented by 1 and there is no change in higher order register	Register addressing mode

# Instructions for decrementing contents of Register/Memory and Register pair

Opcode	Operand	Meaning	Explanation	Addressing mode
DCR	R M	Increment R/M by 1	This instruction decrements 8 bit register or memory location by 1	Register addressing mode
DCX	Rp H – HL pair B – BC pair D – DE pair	Increment contents of register pair by 1	This instruction decrements 16 bits data in register pair by 1, i.e. lower order register is decremented by 1 and there is no change in higher order register	Register addressing mode



# Programming Examples

**Problem Statement 2:** Write an Assembly language Program to load the data bytes 55H, 66H, 77H, 88H and 99H in memory locations 2501H, 2502H, 2503H, 2504H and 2505H

---

LXI H,2501H

MVI M,55H

INX H

MVI M,66H

INX H

MVI M,77H

INX H

MVI M,88H

INX H

MVI M,99H

HLT

# Before and after execution

The screenshot shows the 8086 simulator interface before execution. The top status bar displays the address `0x0`. The left sidebar contains icons for Memory, I/O, and a lightbulb. The main window is divided into three sections: Memory, Disassembler, and I/O. The Memory section shows a range of `0x2501 - 0x2505` with a table of memory locations. The Disassembler section shows a list of instructions. The I/O section is empty.

**Memory**

0x2501 - 0x2505

Displaying Memory Locations from 0x2501 to 0x2505

Double click the value to edit then press Enter to save the value or Tab to edit the next location.

Address	Value
0x2501	00
0x2502	00
0x2503	00
0x2504	00
0x2505	00

**Disassembler**

```
1 LXI H,2501H
2 MVI M,55H
3 INX H
4 MVI M,66H
5 INX H
6 MVI M,77H
7 INX H
8 MVI M,88H
9 INX H
10 MVI M,99H
11 HLT
```

The screenshot shows the 8086 simulator interface after execution. The top status bar displays the address `0x0`. The left sidebar contains icons for Memory, I/O, and a lightbulb. The main window is divided into three sections: Memory, Disassembler, and I/O. The Memory section shows a range of `0x2501 - 0x2505` with a table of memory locations. The Disassembler section shows a list of instructions. The I/O section is empty. A tooltip for the 'Load & Run' button is visible, showing the text 'Load & Run Ctrl + F5'.

**Memory**

0x2501 - 0x2505

Displaying Memory Locations from 0x2501 to 0x2505

Double click the value to edit then press Enter to save the value or Tab to edit the next location.

Address	Value
0x2501	55
0x2502	66
0x2503	77
0x2504	88
0x2505	99

**Disassembler**

```
1 LXI H,2501H
2 MVI M,55H
3 INX H
4 MVI M,66H
5 INX H
6 MVI M,77H
7 INX H
8 MVI M,88H
9 INX H
10 MVI M,99H
11 HLT
```

**I/O**

# Programming Examples

**Problem Statement 3:** Write a program to move the data byte 78H in register A and copy the contents of accumulator in register B and memory location 2501H.

---

**MVI A,78H**

**MOV B,A**

**STA 2501H**

**HLT**



# Programming Examples

**Problem Statement 4:** Write a program to move the contents of memory locations 2501H and 2502H in registers B and C respectively.

---

LXI H, 2501H

MOV B,M

INX H

MOV C,M

HLT

# Jump Instructions

Opcode	Operand	Meaning	Explanation	Addressing mode
JMP	16 bits address /Label	Unconditional jump	Jump to the address specified in the instruction	Implicit addressing mode
JC		Jump on carry	Jump to the specified address if carry flag is set	
JNC		Jump on no carry	Jump to the specified address if carry flag is reset	
JZ			Jump to the specified address if Zero flag is set	
JNZ			Jump to the specified address if Zero flag is reset	

# Programming Examples

**Problem Statement 3:** A block of 5 data bytes is stored in memory locations starting at 2501H. Write an Assembly language Program to copy the block from memory locations starting at 2601H.

---

LXI H,2501H

LXI D,2601H

MVI C,05H

L1: MOV A,M

STAX D

INX H

INX D

DCR C

JNZ L1

HLT



# Programming Examples

**Problem Statement 3:** A block of 5 data bytes is stored in memory locations starting at 2501H. Write an Assembly language Program to copy the block from memory locations starting at 2601H in reverse order.

---

LXI H,2501H

LXI D,2605H

MVI C,05H

L1: MOV A,M

STAX D

INX H

DCX D

DCR C

JNZ L1

HLT

Thank  
you