

# Transferencia de Datos en Bloques

Jorge L. Morales Ortiz  
Héctor M. Solís Villodas

## Abstracto

El microprocesador Intel 8085 es capaz de transferir datos de un espacio de memoria a otro mediante una serie de instrucciones. En este experimento trabajaremos con las instrucciones de transferencia de datos para transferir datos de un bloque de memoria a otro, además de practicar las instrucciones de jump condicional aprendidas anteriormente.

## I. Introducción

La transferencia de datos en el 8085 es un proceso en el cual se mueven datos de un lugar en específico a otro. La transferencia de datos puede ser de registro a registro, de registro a memoria, o de memoria a registro. Este proceso se lleva a cabo mediante el uso de una serie de instrucciones que emplea el microprocesador. Las instrucciones más comunes son las siguientes:

- **MOV** – Mueve Datos.
- **MVI** – Mueve datos inmediatamente.
- **LDA** – Carga el acumulador directamente de memoria.
- **STA** – Guarda el acumulador directamente en memoria.
- **LHLD** – Carga el registro HL directamente de memoria.
- **SHLD** – Guarda el registro HL directamente en memoria.

En las siguientes instrucciones, la X implica que la transferencia de datos esta relacionada con un registro par (16 bits).

- **LXI** – Carga el registro par inmediatamente.
- **LDAX** – Carga el acumulador desde una dirección de un registro par.

- **STAX** – Guarda en el registro par lo que está en el acumulador.
- **XCHG** – Intercambio entre los registros pares DE y HL.

Con esta serie de instrucciones se puede programar el microprocesador 8085 para que transfiera los datos en bloques de un espacio de memoria a otro.

## II. Experimento

### Descripción del Problema #1:

16 bytes de data están almacenados en los espacios en memoria 0050H a 005FH. Transfiera todo el bloque de datos a un nuevo espacio en memoria comenzando desde 0070H.

Los datos en memoria (en hexadecimal) son los siguientes:

37, A2, F2, 82, 57, 5A, 7F, DA, E5, 8B, A7, C2, B8, 10, 19, 98

**Programa #1 –**

Address	Assembly	HexCode
C000	LXI H, C050	21
C001		50
C002		C0
C003	LXI D, C070	11
C004		70
C005		C0
C006	MVI B, 10	06
C007		10
C008	<b>NEXT:</b> MOV A, M	7E
C009	STAX D	12
C00A	INX H	23
C00B	INX D	13
C00C	DCR B	05
C00D	JNZ <b>NEXT</b>	C2
C00E		08
C00F		C0
C010	HLT	76

Haga un flujograma que describa cómo funciona el programa. Determine el estado de cada espacio en memoria comenzando desde 0070H.

Descripción del Problema #2:

6 bytes de data están acumulados en las localizaciones en memoria comenzando en 0050H. Sume todos los bytes de data. Utilice el registro B para salvar cualquier “carry” generado, mientras suma los datos. Muestre la suma completa almacenando el resultado en dos espacios de memoria consecutivos, 0070H y 0071H.

Los datos en memoria (en hexadecimal) son los siguientes:

A2, FA, DF, E5, 98, 8B

**Programa #2 –**

Address	Assembly	HexCode
C000	XRA, A	AF
C001	MOV B, A	47
C002	MVI C, 06	0E
C003		06
C004	LXI H, C050	21
C005		50
C006		C0
C007	<b>NXTBYT:</b> ADD M	86
C008	<b>JNC NXTMEM</b>	D2
C009		xx→0C
C00A		xx→C0
C00B	INR B	04
C00C	<b>NXTMEM:</b> INX H	23
C00D	DCR C	0D
C00E	<b>JNZ NXTBYT</b>	C2
C00F		07
C010		C0
C011	LXI H, C070	21
C012		70
C013		C0
C014	MOV M, A	77
C015	INX H	23
C016	MOV M, B	70
C017	HLT	76

Haga un flujograma para describir el funcionamiento del programa. Determine el estado de las direcciones en memoria 0070H y 0071H.

### III. Análisis de Datos

### Flujograma del Programa #1 –

#### Programa #1 –

Mnemonic	Comentarios
LXI H, C050	Carga el primer numero (37) al registro HL.
LXI D, C070	Carga en el registro DE lo que esta en la dirección C070.
MVI B, 10	Carga en el registro B 10.
<b>NEXT:</b> MOV A, M	Mover de memoria al acumulador.
STAX D	Guarda en DE lo que esta en el acumulador.
INX H	Incrementa el registro HL (carga el próximo número).
INX D	Incrementa el registro DE.
DCR B	Decrementa el registro B.
JNZ NEXT	Brinca si el resultado no es cero a <b>NEXT</b> .
HLT	Detenerse (se ejecuta sólo cuando el resultado es cero).

Al ejecutar el programa, éste transfiere los datos que están en el bloque de memoria de la dirección C050 hasta C05F al bloque de memoria de C070 en adelante. Los datos almacenados en estas localizaciones de memoria son: 37, A2, F2, 82, 57, 5A, 7F, DA, E5, 8B, A7, C2, B8, 10, 19, 98.

Inicializar.

Cargar el registro HL.

Cargar el registro DE.

Cargar dato en el registro B.

Mover datos de memoria al acumulador.

Guarda en DE lo que está en el acumulador.

Incrementa el registro HL.

Incrementa el registro DE.

Decrementa el registro B.

No

El resultado es cero?

Si

Detenerse.

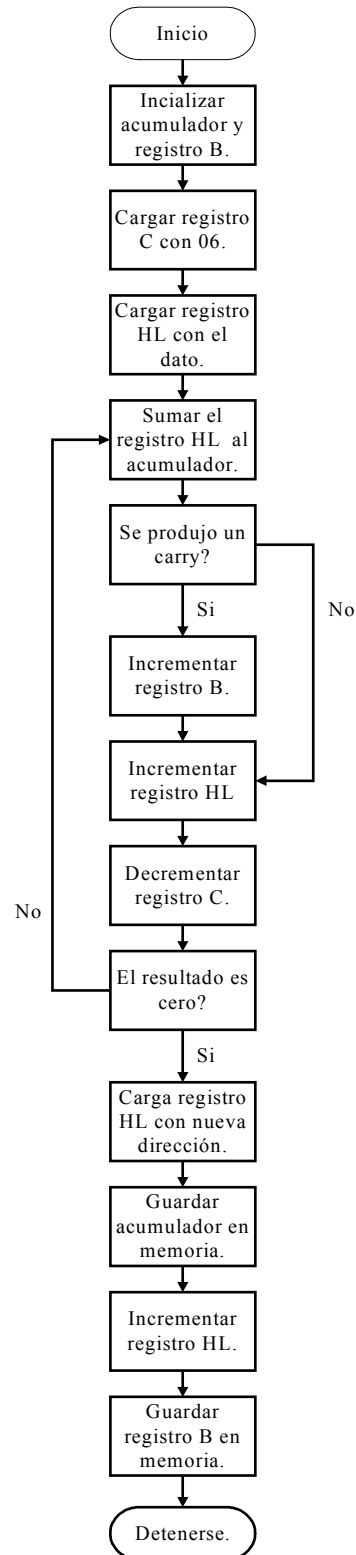
**Programa #2 –**

Assembly	Comentarios
XRA, A	Inicializa el acumulador con cero (0).
MOV B, A	Carga en el registro B lo que está en el acumulador (0).
MVI C, 06	Carga 06 en el registro C.
LXI H, C050	Carga en el registro HL lo que está en la dirección C050 (A2).
<b>NXTBYT:</b> ADD M	Le suma al acumulador lo que está en el registro HL y lo guarda en el acumulador.
JNC NXTMEM	Si el “carry” es cero salta a la dirección <b>NXTMEM</b> .
INR B	Incrementa el registro B (sólo se ejecuta si el carry no es cero).
<b>NXTMEM:</b> INX H	Incrementa el registro HL (carga el próximo número de la lista).
DCR C	Decrementa el registro C.
JNZ NXTBYT	Si el resultado no es cero, salta a la dirección <b>NXTBYT</b> .
LXI H, C070	Carga en el registro HL lo que está en la dirección C070 (sólo se ejecuta si el resultado es cero).
MOV M, A	Carga en el registro HL lo que está en el acumulador (el resultado).
INX H	Incrementa el registro HL (dirección C071).
MOV M, B	Carga en el registro HL lo que está en el registro B (el “carry”).
HLT	Detenerse.

El resultado de la suma de los datos es 483. Al finalizar el programa, en la dirección de

memoria 0070H aparece guardado 83, y en la dirección 0071H aparece guardado 04.

Flujograma del Programa #2 –



## IV. Conclusión

En este experimento trabajamos con el módulo 8085 para practicar las instrucciones de transferencia de datos en bloque. Estas instrucciones nos permiten mover un conjunto (bloque) de información de un espacio en memoria a otro fácilmente. Las instrucciones de jump condicional del 8085 también fueron utilizadas, además de las instrucciones de transferencia de datos, para la transferencia de información en bloque. Además del uso de tablas para analizar los programas, se emplearon flujogramas para describir el funcionamiento de los mismos.

## V. Referencias

(1) Intel 8085 Microprocessor Instruction Set