Transferencia de Datos en Bloques

Maite Orama Miranda, William Neris Diaz
Departamento de Física y Electrónica

Abstracto

En este experimento estaremos practicando las instrucciones de transferencia de datos en bloque para el 8085 y de igual manera practicaremos los jump condicional.

I. Teoría

Una de las funciones principales del microprocesador es el copiar data, de un registro llamada la fuente, a otro registro llamado el destino. En literatura técnica, la función de copiar es frecuentemente llamada la función de transferencia de datos, lo que es algo engañoso. De hecho el contenido de la fuente no son transferidos, sino copiados en el registro de destino sin modificar el contenido de la fuente.

Algunas de las instrucciones que son utilizadas para copiar data son los siguientes:

MOV: Move
Copia un byte de data

MVI: Move Immediate
Carga directamente un byte de data

OUT: Out to Port
Envía un byte de data a una salida

IN: Input from Port
Lee un byte de entrada.

El termino copia es igualmente valido para una función de entrada o salida porque el contenido de la fuente no es alterado. De cualquier forma, el termino transferencia de data es utilizado tan comúnmente para indicar la función de copia de datos, que estos términos se pueden intercambiar cuando su significado no es ambiguo.(1)

Las instrucciones para copiar data de un microprocesador a una localización de memoria son similares a las descritas anteriormente, algunas de estas instrucciones son:(2)

1. MOV M,R: Move (from Register to Memory)
   - Esta es una instrucción de 1 byte que copia los datos de un registro R, hacia la localización de memoria especificada por el contenido del registro HL.

2. STAX B/D: Store Accumulator Indirect
   - Esta es una instrucción de 1 byte que copia los datos del acumulador hacia la localización de memoria especificada por el contenido de los registros BC o DE

3. STA 16-bit: Store Accumulador Direct

- Esta es una instrucción de 3 byte que copia los datos del acumulador hacia la localización de memoria especificada por un operando de 16 bits.

4. MVI M,8-bit: Load 8-bit data in memory

- Esta es una instrucción de dos bytes; el Segundo byte especifica el data de 8-bit
- La localización de memoria es especificada por el contenido del registro HL.

II. Experimento

En este laboratorio hicimos varios programas los cuales realizaban transferencias de datos con diferentes instrucciones.

Programa 1:

Direc	Instrucción	Opcode	Comentario
C000	LXIH,C050	2150C0	Carga el valor que esta en esa dirección
C003	LXID,C070	1170C0	Carga el valor que esta en esa dirección
C006	MVI B, 10	06 10	Mueve 10 al registro B
C008	NEXT: MOV A, M	7E	Mueve lo que esta en el acumulador a memoria
C009	STAX D	12	Guarda indirectamente lo que esta en el acumulador
C00A	INX H	23	Incrementa H
C00B	INX D	12	Incrementa D
C00C	DCR B	05	Decrementa el registro B
C00D	JNZ NEXT	C208C0	Brinca a la siguiente dirección si el registro B no es cero
C010	HLT	76	Terminar

En este programa grabamos previamente los siguientes datos en los espacios de memoria C0050 a C005F:

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

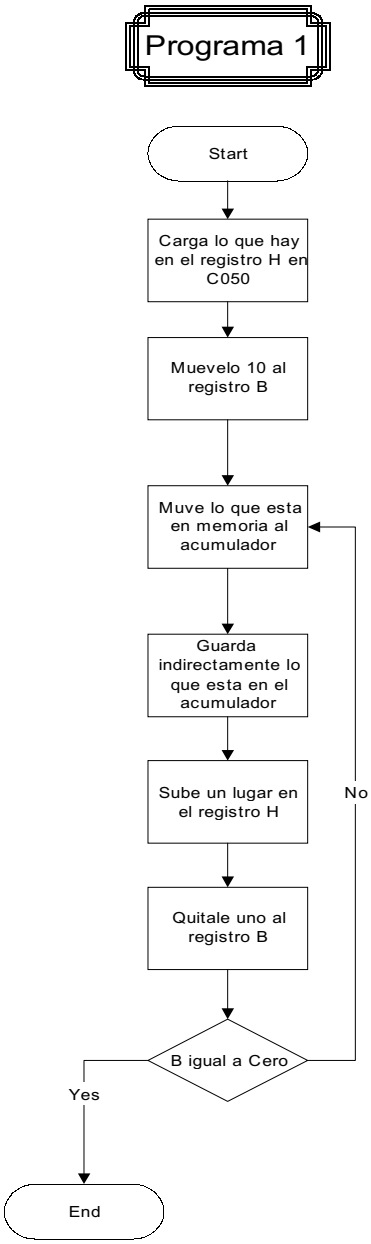
El *programa 2* fue el siguiente:

Direc	Instrucción	Opcode	Comentario
C000	XRA, A	AF	Carga el valor que esta en esa dirección
C001	MOV B, A	47	Mueve lo que esta en el acumulador al registro B
C002	MVI C, 06	0E, 06	Mueve 06 al registro C
C004	LXI H, C050	21C050	Carga el valor que esta en esa dirección
C007	ADD M	86	Suma lo que esta en el acumulador a memoria
C008	JNC NXTMEM	D20CC0	Brinca a la siguiente dirección si no hay carry
C00B	INR B	04	Incrementa B
C00C	INX H	23	Incrementa H
C00D	DCR C	0D	Decrementa C
C00E	JNZ NXTBYT	C207C0	Brinca si no es cero
C011	OUTPORT1	D3 FF	Muestra el resultado en los seven segment
C013	MOV A, B	78	Mueve lo que esta en A al registro B
C014	OUTPORT2	D3 00	Muestra el resultado en los seven segment

III. Análisis y Discusión

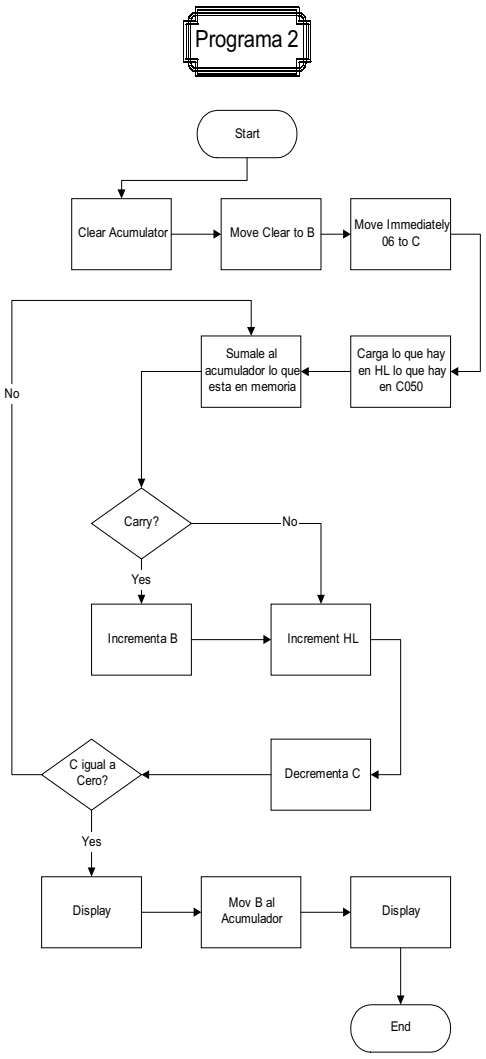
En el primer programa movíamos los datos que estaban en la dirección antes mencionada a la dirección C0070. Se podía ver claramente si nos dirigíamos a esos espacios en memoria como el microprocesador movía los datos instantáneamente.

El flow Chart correspondiente al programa 1 seria el siguiente:



Mientras el segundo programa sumaba todos los bytes de data. Se utilizo el registro B para grabar si en la suma había un carry. Esa suma se mostraba en los puertos de salida.

El flow Chart correspondiente al segundo programa seria el siguiente:



IV. Conclusión

En conclusión pudimos aprender como el microprocesador tiene la capacidad de mover datos entre espacios de memoria y de igual manera sumarlos. En general este laboratorio fue uno simple a pesar de que confrontamos dificultades con los puertos de salida en el segundo programa.

V. Referencias:

[1] Gaonkar, Armes S. Introduction to 8085 Instrcution: Data Transfer Operations, Microprocessor Architecture, Programind, and Aplications with the 8085 (Prentice Hall 2002), pag 176)

[2] Gaonkar, Armes S. Introduction to 8085 Instrcution: Data Transfer Operations, Microprocessor Architecture, Programind, and Aplications with the 8085 (Prentice Hall 2002), pag 235-236)