

UNIVERSIDAD DE PUERTO RICO
RECINTO DE HUMACAO
DEPARTAMENTO DE FÍSICA Y ELECTRÓNICA
Glendalys Figueroa Freytes e Isaura Rivera Meléndez

LABORATORIO #7: TRANSFERENCIA DE DATOS EN BLOQUES.

Mediante este experimento se practicarán las instrucciones de transferencia de datos en bloque y las instrucciones de “jump” incondicional del 8085.

I. Introducción

Todo microprocesador posee un set de instrucciones el cual puede ejecutar. La colección de instrucciones es implementada como patrones de bit, de los cuales cada uno posee un significado diferente cuando son cargados en los registros. Estos patrones de bit están definidos como un set de palabras cortas las cuales son llamadas el lenguaje de assembly del procesador. Un assembler puede traducir las palabras en sus patrones de bit fácilmente colocando la salida del assembler en memoria para que el microprocesador la ejecute.

Parte de estas instrucciones lo son las de aritmética (ADD, ADI, SUI, SUB, INX, INR, DCX, DCR) y las de jump incondicional (JMP). Estas instrucciones también se utilizan para mover los datos en bloque en la memoria. Esto se puede hacer accedando los datos uno por uno y asignándole la dirección que sea necesaria ⁽¹⁾.

II. Experimentación

Se trabajó con un módulo en donde se guardó en los espacios de memoria los siguientes datos:

Programa #1	
Address	Datos
C050	37
C051	A2
C052	F2
C053	82
C054	57
C055	5A
C056	7F
C057	DA
C058	E5
C059	8B
C05A	A7
C05B	C2
C05C	B8
C05D	10
C05E	19
C05F	98

Programa #2	
C050	A2
C051	FA
C052	DF
C053	E5
C054	98
C055	8B

Luego se entraron los programas de los problemas 1-2 y se corrieron para ver los resultados. Estos fueron luego comparados con los resultados del análisis hecho a mano para corroborar los resultados presentados en el módulo.

III. Análisis y datos

Programa #1

Label	Address	Assembly	Hex Code	Comments
	C000 C001 C002	LXI H, C050	21 50 C0	Cargar el address C050 al registro H
	C003 C004 C005	LXI D, C070	21 70 C0	Cargar el address C070 al registro D
	C006 C007	MVI B, 10	06 10	Mover el # 10 a B
NEXT	C008	MOV A, M	7E	Mover el contenido de memoria al acumulador
	C009	STAX D	12	Guardar una copia del contenido del acumulador a la localización de memoria almacenada en D
	C00A	INX H	23	Incrementar H
	C00B	DCR B	05	Decrementar B
	C00C C00D C00E	JNZ C008	C2 08 C0	Brincar al address C008 si el contenido de B no es cero
	C00F	HLT	76	Terminar

Tabla 1: Instrucciones del programa #1

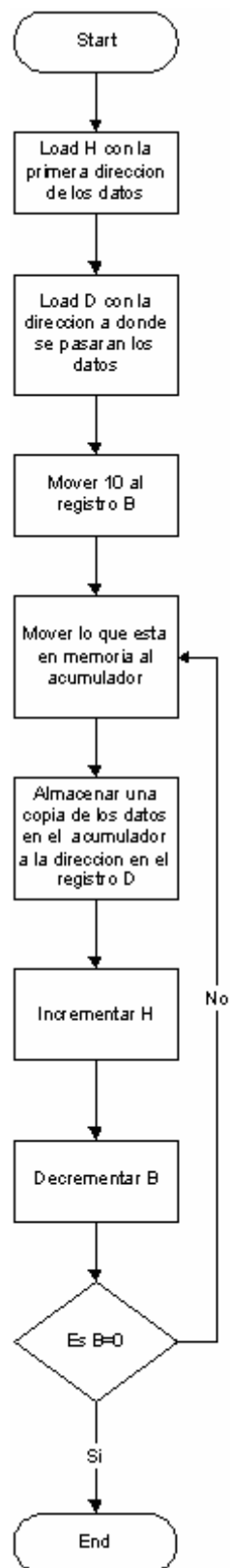


Figura 1: “Flowchart” del programa #1

Estado cada espacio de memoria desde C070:

C070 37
 C071 A2
 C072 F2
 C073 82
 C074 57
 C075 5A
 C076 7F
 C077 DA
 C078 E5
 C079 8B
 C07A A7
 C07B C2
 C07C B8
 C07D 10
 C07E 19
 C07F 98

Programa #2

Label	Address	Assembly	Hex Code	Comments
	C000	XRA , A	AF	Borrar el contenido de A
	C001	MOV B , A	47	Mover el contenido del A a B
	C002	MVI C, 06	0E	Mover 06 a C
	C003		06	
	C004	LXI H,	21	Poner en H el
	C005	C050	50	address de
	C006		C0	memoria
NXTBYT	C007	ADD M	86	Añadirle 1 a memoria
	C008	JNC	D2	Brincar si no
	C009	C00C	0C	hay carry al
	C00A		C0	address C00C
	C00B	INR B	04	Incrementar B
NXTMEM	C00C	INX H	23	Incrementar H
	C00D	DCR C	0D	Decrementar C
	C00E	JNZ C007	C2	Brincar al
	C00F		07	address C007
	C010		C0	si C no es cero
	C011	OUT (00)	D3	Display High
	C012		00	Byte
	C013	MOV A, B	78	Mover B a A
	C014	OUT (FF)	D3	Display Low
	C015		FF	Byte
	C016	HLT	76	Terminar

Tabla 2: Instrucciones del programa 2

Al terminar con el OUT (00) el resultado presentado es 04 y cuando se utilizo OUT (FF) apareció 83 en el display. El resultado completo es 0483H.

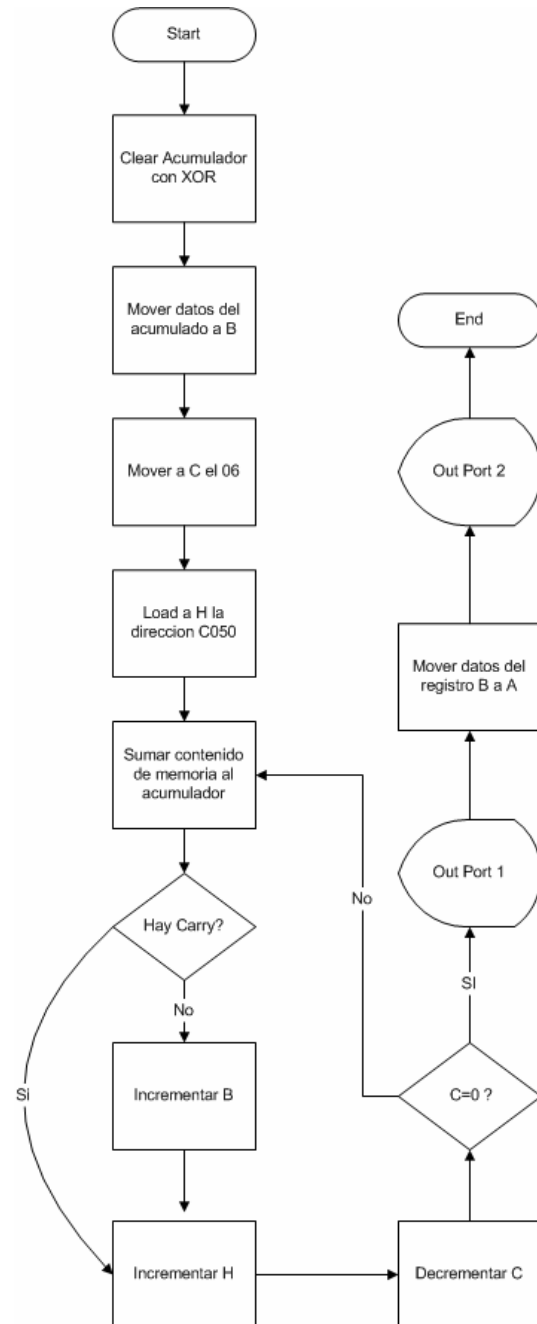


Figura 2: "Flowchart" de Programa #2

El código para que los resultados aparezcan en dos espacios de memoria correspondientes a C070 y C071 se encuentra redactado en la Tabla 3.

Label	Address	Assembly	Hex Code	Comments
	C000	XRA , A	AF	Borrar el contenido de A
	C001	MOV B , A	47	Mover el contenido de A a B
	C002 C003	MVI C, 06	0E 06	Mover 06 a C
	C004 C005 C006	LXI H, C050	21 50 C0	Poner en H el address de memoria
<i>NXTBYT:</i>	C007	ADD M	86	Añadirle 1 a memoria
	C008 C009 C00A	JNC C00C	D2 0C C0	Brincar si no hay carry a C00C
	C00B	INR B	04	Incrementar B
<i>NXTMEM:</i>	C00C	INX H	23	Incrementar H
	C00D	DCR C	0D	Decrementar C
	C00E C00F C010	JNZ C007	C2 07 C0	Brincar al ardes C007 si el resultado de C no es 0
	C011 C012 C013	LXI H C070	21 70 C0	Cargar en H la dirección C070
	C014	MOV M, A	77	Mover lo que esta en el Acumulador a memoria
	C015	INX H	23	Incrementar H
	C016	MOV M, B	70	Mover lo que esta en B a memoria
	C017	HLT	76	Terminar

Tabla 3: Instrucciones del programa #2 modificada

IV. Conclusión

En este experimento se pudo observar la función de mover bloques de datos de un lugar de memoria a otro. También se vio como es que se suman datos que están en direcciones cercanas accesándolos uno a uno.

V. Referencias

1) <http://computer.howstuffworks.com>