

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

LABORATORIO # 10: STACK Y SUBRUTINAS EN EL 8085.

Mediante este experimento se practicarán las instrucciones de tiempo para el 8085 y entender las instrucciones de snack para hacer subrutinas en el 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 son utilizadas para crear subrutinas. Las instrucciones de stack (PUSH, POP) son utilizadas para guardar la última dirección que se accesó antes de acceder la subrutina. La subrutina se accesa con la instrucción CALL y se regresa de ésta con la instrucción RET. Las subrutinas son útiles cuando en un programa se necesitan acceder a cierto proceso varias veces ⁽¹⁾.

II. Experimentación

Se trabajó con un módulo en donde 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 SP, C099	31 99 C0	Cargar el address C099 al registro SP.
	C003 C004	MVI L, 00	2E 00	Mover 00 al registro L.
	C005	PUSH H	E5	Mover el contenido de H a la dirección establecida por el SP.
	C006	POP PSW	F1	Sustituir contenido actual del PSW por el almacenado en el "stack area"..
	C007	POP H	E1	Sustituir contenido actual de H por el

				almacenado en el "stack area".
	C008	MOV A, L	7D	Mover el contenido de L al registro A.
	C009	OUT PORT 0	D3 00	Mostrar el contenido de A.
	C00A C00B	MVI A, 00	3E 00	Mover a A 00.
	C00C	ORA A	B7	Hacer un OR del contenido de A con A.
	C00D	PUSH PSW	F5	Mover el contenido de PSW a la dirección establecida por el SP.
	C00E	POP H	E1	Sustituir contenido actual de H por el almacenado en el "stack area".
	C00F	MOV A, L	7D	Mover el contenido de L al registro A.
	C010 C011	ANI 40	E6 40	Hacer un AND con el contenido de A y 40.
	C012	OUT PORT 1	D3 01	Mostrar el contenido de A.
	C013	HLT	76	Terminar

Tabla 1: Instrucciones del programa 1

Al terminar el programa, el puerto 00 presentaba el 40.

Programa #2: Subrutinas

Label	Address	Assembly	Hex Code	Comments
	C000 C001 C002	LXI SP, C099	31 99 C0	Cargar el address C099 al registro SP.
	C003	MVI A, 20	3E 20	Mover 20 a A.
	C004 C005	OUT PORT 0	D3 00	Mostrar el contenido de A.
	C006 C007	MVI B, 0A	06 0A	Mover 20 a B.
	C008 C009 C00A	CALL DELAY	CD 50 C0	Llamar subrutina DELAY.
	C00B C00C	MVI A, 30	E3 30	Mover 30 a A.
	C00D C00E	OUT PORT 1	D3 01	Mostrar el contenido de A.
	C00F	HLT	76	Terminar
DELAY:	C050	PUSH D	D5	Mover el contenido de D en SP.
	C051	PUSH PSW	F5	Mover el contenido de PSW a la dirección establecida por el SP.
SEG	C052 C053 C054	LXI D, 28	11 28 00	Poner 0028 en el registro D.
LOOP:	C055	DCX D	1B	Decrementar D.
	C056	MOV A, D	7A	Mover contenido de D a A.
	C057	ORA E	B3	Hacer un OR entre A y E.
	C058 C059 C05A	JNZ LOOP	C2 55 C0	Brincar al LOOP si A no es 0.
	C05B	DCR B	05	Decrementar B.
	C05C C05D C05E	JNZ SEG	C2 52 C0	Brincar a SEG si A no es 0.
	C05F	POP PSW	F1	Sustituir contenido actual del PSW por el almacenado en el "stack area".
	C060	POP D		Sustituir contenido actual de D por el almacenado en el "stack area".
	C05F	HLT	76	Terminar

Tabla 2: Instrucciones del programa 2

Al terminar el programa, en el puerto de salida se vio primero 20 y un segundo después cambió a 30.

Programa #3: Semáforo con el 8085

Label	Address	Assembly	Hex Code	Comments
	C000 C001 C002	LXI SP, C099	31 99 C0	Cargar el address C099 al registro SP.
	C003 C004	MVI A, 41	3E 41	Mover al registro A el 41
	C005 C006	OUT 00	D3 00	Mostrar contenido de A
	C007 C008	MVI B, 13	06 13	Mover a B el 13
	C009 C00A C00B	CALL DELAY	CD 50 C0	Llamar subrutina DELAY
	C00C C00D	MVI A, 84	3E 84	Mover a A el 84
	C00E C00F	OUT PORT 01	D3 01	Mostrar contenido de A
	C010 C011	MVI B, 07	06 07	Mover a B el 07
	C012 C013 C014	CALL DELAY	CD 50 C0	Llamar subrutina DELAY
	C015 C016	MVI A, 90	3E 90	Mover a A 90
	C017 C018	OUT PORT 02	D3 02	Mostrar contenido de A
	C019 C01A	MVI B, 17	06 17	Mover a B el 17
	C01B C01C C01D	CALL DELAY	CD 50 C0	Llamar subrutina DELAY
	C01E C01F C020	JMP C003	C3 03 C0	Volver a la dirección C003

Tabla 3: Tabla de instrucciones del semáforo

Al terminar el programa, primero se presentaba el 41 que representa la luz verde y la de caminar activa. Después de 15 segundos se presentaba el 84 que representaba la luz amarilla y el “don’t walk” activo. Cuando pasen 5 segundos se presenta el 90 que representa la luz roja y el “don’t walk” activo.

IV. Conclusión

En este experimento se pudo observar la función del stack y las subrutinas. El primer experimento no dio el resultado esperado pero el segundo, nos presento como con es que se hace el call a una subrutina, en este caso al DELAY, y luego seguía con la programación principal.

V. Referencias

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