

# Counter y Delays en el 8085

Jorge L. Morales Ortiz  
Héctor M. Solís Villodas

## Abstracto

El propósito de este experimento es estudiar y aplicar las instrucciones de tiempo del microprocesador Intel 8085. Estas instrucciones son utilizadas para producir “counters” y “delays” condicionales dentro del programa, los cuales serán ejecutados luego por el microprocesador.

### I. Introducción

Los “counters” son usados principalmente para controlar eventos dentro del microprocesador. Los “time delays” son utilizados para establecer un intervalo específico entre dos eventos. Los procesos de diseñar *counters* y *time delays* utilizan instrucciones de programación y son más convenientes que el diseño con componentes electrónicos.

#### Counters

El *counter* está diseñado simplemente para cargar un número deseado en uno de los registros, y utiliza una de las instrucciones de incrementar o decrementar para cambiar el contenido del registro. El ciclo (*loop*) está establecido para actualizar el conteo, y cada conteo es revisado para determinar si ha llegado al número final; de lo contrario, el ciclo se repite nuevamente. Para poder observar el conteo claramente, debe establecerse un intervalo de tiempo (*time delay*) apropiado para cada conteo.

#### Time Delays

El proceso utilizado para diseñar un *delay* específico es similar al utilizado para empezar un *counter*. Un registro es cargado con un número, dependiendo de un *time delay* requerido. El registro entonces es decrementado hasta que llegue a cero, a través de un ciclo (*loop*) utilizando una instrucción de “jump” condicional (JNZ).

El ciclo que causa el delay depende del periodo del reloj del microprocesador. En el 8085 la frecuencia ( $f$ ) del reloj es de 2MHz, o sea que el periodo ( $T$ ) del microprocesador es de  $0.5\mu s$ :

$$T = \frac{1}{f} = \frac{1}{2MHz} = 0.5\mu s$$

Para calcular el “Time Delay” (TD) se utiliza la siguiente fórmula:

$$TD = (T) \times (LoopT - States) \times (N10)$$

donde  $T$  es el periodo del reloj del CPU,  $LoopT-States$  es la suma de todos los  $T-States$  de cada instrucción, y  $N10$  es el valor del delay cargado en el registro en decimal.

### II. Experimento

**Programa #1** – *Time Delay* utilizando un registro par

Address	Assembly	HexCode	T-States
C000	LXI B, FFFF	01	10T
C001		FF	
C002		FF	
C003	<b>LOOP:</b>	D3	10T
C004	OUT 00	00	
C005	DCX B	0B	6T
C006	MOV A, C	79	4T
C007	ORA B	B0	4T
C008	JNZ <b>LOOP</b>	C2	10/7T
C009		03	
C00A		C0	
C00B	HLT	76	5T

Calcule el *Time Delay* para este programa. Haga un flujograma para este programa de cómo debe ir organizado. Anote los resultados.

**Programa #2** – *Time Delay* con dos *Loops*

Address	Assembly	HexCode	T-States
C000	MVI B, 38	06	7T
C001		38	
C002	<b>LOOP2:</b>	0E	7T
C003	MVI C, FF	FF	
C004	<b>LOOP1:</b>	D3	10T
C005	OUT 00	00	
C006	DCR C	0D	4T
C007	JNZ LOOP1	C2	10/7T
C008		04	
C009		C0	
C00A	DCR B	05	4T
C00B	JNZ LOOP2	C2	10/7T
C00C		02	
C00D		C0	
C00E	HLT	76	5T

Calcule el *Time Delay* para los dos ciclos del programa. Haga un flujograma para el programa de cómo debe ir organizado. Anote los resultados.

Descripción del problema:

Escriba un programa (Programa #3) que decremente de FFH a 00H con un período de 0.5 segundos en cada cambio con lo aprendido en el laboratorio.

### III. Análisis de Datos

**Programa #1** –

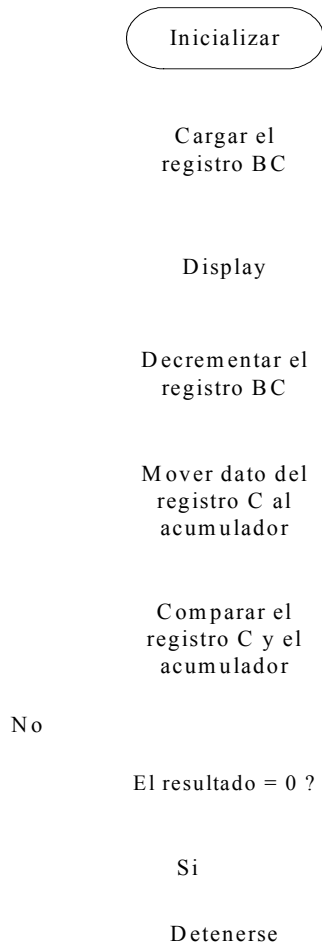
Assembly	Comentarios
LXI B, FFFFH	Carga en el registro BC FFFF.
<b>LOOP:</b> OUT 00	Muestra el dato.
DCX B	Decrementa el registro BC.
MOV A, C	Mueve lo que está en el registro C al acumulador (A).
ORA B	Hace una operación lógica OR entre el registro C y A.
JNZ LOOP	Salta a <b>LOOP</b> si el resultado no es cero.
HALT	Detenerse (solo se ejecuta si el resultado es cero).

$$T = \frac{1}{2MHz} = 0.5\mu s$$

$$T - States = 49T$$

$$TD = (0.5\mu s) \times (49) \times (65535) = 1.61s$$

Flujograma del Programa #1 –



**Programa #2 –**

Assembly	Comentarios
MVIB, 38	Carga 38 en el registro B.
<b>LOOP2:</b>	Carga FF en el registro C.
MVI C, FF	
<b>LOOP1:</b>	Muestra lo que está en el registro C.
OUT 00	
DCR C	Decrementa el registro C.
JNZ LOOP1	Si el resultado no es cero, salta a la dirección <b>LOOP1</b> .
DCR B	Decrementa el registro B (sólo se ejecuta si el resultado es cero).
JNZ LOOP2	Si el resultado no es cero, salta a la dirección <b>LOOP2</b> .
HLT	Detenerse (sólo se ejecuta si el resultado es cero).

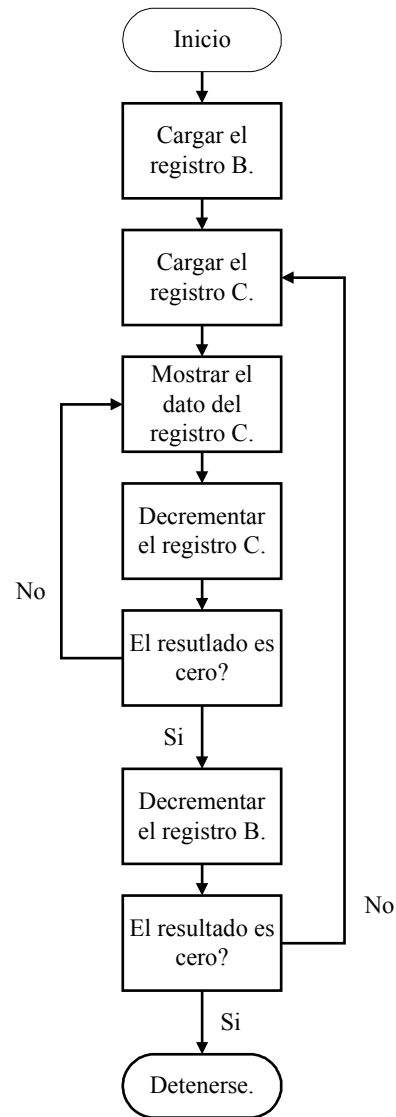
$$T = \frac{1}{2MHz} = 0.5\mu s$$

$$T - States = 57T$$

$$TDregB = (0.5\mu s) \times (57) \times (56) = 1.60ms$$

$$TDregC = (0.5\mu s) \times (57) \times (255) = 7.27ms$$

Flujograma del Programa #2 –



**Programa #3 –**

El siguiente programa muestra los números en orden descendente desde FFH hasta 00H con un delay de 0.5 segundos entre el cambio de un número y el próximo:

Address	Assembly	HexCode	T-States
C000	MVI D, FF	16	7T
C001		FF	
C002	<b>LOOP2:</b> MOV A, D	7A	4T
C003	OUT 00	D3	10T
C004		00	
C005	LXI B, AA00	01	10T
C006		00	
C007		AA	
C008	<b>LOOP1:</b> DCX B	0B	6T
C009	MOV A, C	79	4T
C00A	ORA B	B0	4T
C00B	JNZ <b>LOOP1</b>	C2	10/7T
C00C		08	
C00D		C0	
C00E	DCR D	15	4T
C00F	JNZ <b>LOOP2</b>	C2	10/7T
C010		02	
C011		C0	
C012	MOV A, D	7A	4T
C013	OUT 00	D3	10T
C014		00	
C015	HLT	76	5T

Para calcular el time delay de 0.5 segundos se utilizaron los T-States del segmento de programación desde C005 hasta C00D (34T) junto con la ecuación del time delay, despejando para obtener el número decimal:

$$N10 = \frac{TD}{(T) \times (LoopT - States)} = \frac{0.5s}{(0.5 \times 10^{-6} s) \times (34T)} = 29412 = 72E4H$$

El número obtenido se utilizó como referencia, y después se fue ajustando para producir el time delay deseado (para que el time delay estuviera bastante cerca de 0.5s se utilizó AA00H).

## IV. Conclusión

En este experimento trabajamos con las instrucciones para producir *counters* y *delays* en el 8085. Un counter es diseñado cargando el dato (número) deseado en un registro, y el ciclo se produce decrementando o incrementando la información dentro del registro. Un *time delay*, similarmente, es diseñado cargando un registro con un dato y utilizando un ciclo para decrementar el conteo hasta que llegue a cero. El *time delay* producido depende del periodo del reloj del microprocesador. Los *counters* y los *delays* son técnicas comúnmente utilizadas para aplicaciones como señales de tráfico, relojes digitales, para controlar procesos y para transferencia de datos.

## V. Referencias

- (1) Intel 8085 Microprocessor Instruction Set
- (2) Gaonkar, Ramesh S. Microprocessor Architecture, Programming, and Applications with the 8085. Fifth Edition.