

PRÁCTICA 2. Programas residentes. Programación en ensamblador de la Rutina de Servicio a la Interrupción

TEORÍA:

TEMA 3: ALMACENAMIENTO DE LA INFORMACIÓN

TEMA 5: CONTROL EN EL SISTEMA COMPUTADOR

TEMA 6: COMUNICACIÓN CON EL EXTERIOR

Los dispositivos periféricos llaman la atención de la CPU, a través de los dispositivos de E/S, por medio de las interrupciones. El objetivo de esta práctica consiste en la familiarización con el concepto de interrupción que, de manera general, se define como “una ruptura de secuencia no programada”. En esta práctica se estudian los tipos que podemos encontrar en un sistema basado en la CPU 8086; se muestra la forma en que la CPU actúa cuando se genera una interrupción; y, se diseña una rutina típica de tratamiento de interrupción, explicando previamente el concepto de programa residente.

2.1. Programas residentes

Fundamento teórico

A lo largo de la asignatura de Estructura de Computadores hemos visto cómo la tarea del microprocesador consiste en leer instrucciones, decodificarlas y ejecutarlas continuamente según la secuencia marcada por un programa. Pero en algunas ocasiones, esta secuencia se ve interrumpida debido a un suceso externo o a un error en la ejecución de alguna instrucción. En esta práctica veremos cómo actúa la CPU ante este tipo de situaciones; se verá que rompe la secuencia del programa que se encuentra ejecutando, para pasar a ejecutar el programa destinado a atender la interrupción.

Los tipos de interrupciones para la familia 8086 son:

- Interrupciones internas, excepciones, cepos o traps: desbordamiento, división por cero, instrucción ilegal, etc. Los modernos microprocesadores tienen implementada una circuitería para detectar este tipo de situaciones; si se presentan, reaccionan rompiendo la secuencia de programa y bifurcando a una dirección que, de manera directa o indirecta, tienen definida internamente.
- Interrupciones software: ejecución de una instrucción de tipo INT *i*. Estas interrupciones bien podemos denominarlas “falsas interrupciones” puesto que

contradican la definición de interrupción dada de “ruptura de secuencia no programada”. Estas instrucciones, como todas las instrucciones, son programadas puesto que se producen cuando la secuencia de programa llega a la posición en la que se encuentra dicha instrucción. Pero si es verdad que son un tipo de instrucción en la que el sistema actúa de la misma forma que cuando se produce una interrupción externa o interna; razón por la que a este tipo de instrucciones se les denomina “interrupciones software”.

- Interrupciones externas (interrupciones propiamente dichas). Las interrupciones externas a su vez pueden ser enmascarables (se comunican a la CPU a través de la línea INTR) y no enmascarables (línea NMI). Las interrupciones enmascarables las atiende el microprocesador en función de la inhibición o no de las mismas; en cambio, las interrupciones no enmascarables siempre son atendidas por el microprocesador. Los microprocesadores, de manera general, contemplan estos dos tipos de líneas de petición de interrupción y, es el usuario quien decide qué tipo de dispositivo externo es el que conecta a una u otra línea de petición de interrupción, en función de la importancia y prioridad que tengan los acontecimientos de los que informan los dispositivos externos.

En cualquiera de estos casos, la CPU procede de la siguiente forma:

1. La CPU finaliza la ejecución de la instrucción en curso (*Instrucción i* sobre la Figura 2.1). La atención de las interrupciones externas, si se atienden, siempre se hará cuando se termine de ejecutar la instrucción en curso; nunca antes de terminar todas las fases asociadas a la misma. Con las de tipo software, si las contempla el sistema, ocurrirá exactamente lo mismo. Ante las interrupciones internas, el sistema suele reaccionar de forma diferente, no teniendo por qué terminar completamente y exactamente igual todas las operaciones que implican la instrucción en curso.
2. Se genera la interrupción (línea INT sobre la Figura 2.2) y, si es aceptada, la CPU almacena el contenido del registro de estado SR y del contador de programa (CS e IP) en la pila (Figura 2.2). Aunque hay sistemas computadores que contemplan dos tipos de pilas, la del sistema y la del usuario, el microprocesador 8086 únicamente dispone de una pila que, tanto el sistema como el usuario pueden hacer uso de ella. Para este microprocesador 8086, la pila funciona de la siguiente forma:

- a. la pila crece hacia direcciones decrecientes (esto quiere decir que si el último dato que se introdujo en la pila está en la dirección de memoria i , el siguiente dato se introducirá en la dirección $i-1$);
 - b. el registro puntero de pila (SS:SP) contiene la dirección de memoria del último elemento almacenado en la pila (si el último dato introducido en la pila se encuentra en la dirección i , (SS:SP) apuntará a la dirección i); y,
 - c. para cuando el sistema tiene que almacenar el registro de estado y contador de programa (que cuando se produce una interrupción el sistema tiene que guardar en la pila estos dos valores), se almacena siempre primero el registro de estado (primero se almacena el registro de estado y a continuación el contador de programa).
3. La aceptación de la interrupción se comunica a los dispositivos externos activando el microprocesador su línea INTA (Figura 2.3). En el 8086, el código de la interrupción que se haya provocado, determina la posición de la tabla de vectores de interrupción a partir de la cual, la CPU tiene definida la dirección de comienzo de la rutina asociada a dicha interrupción (Figura 2.3). En este caso, si la interrupción es interna a la CPU, el vector de interrupción lo suministra la propia CPU. Si la interrupción tiene un origen externo, es el propio dispositivo que interrumpe el que suministra a través de las ocho líneas de menos peso del bus de datos, el vector de interrupción (interrupción vectorizada).
4. La CPU bifurca a la dirección de comienzo de la rutina asociada a dicha interrupción, Rutina de Servicio a la Interrupción (RSI), ver Figura 2.4. Para esto, es necesario que la rutina haya sido previamente definida y almacenada en memoria. Para que la RSI pueda ejecutarse cada vez que se produzca la interrupción, es necesario que esté siempre almacenada en memoria; es decir, es programa RSI debe ser lo que se conoce como *Programa Residente*.
5. Para que al finalizar la ejecución de la RSI, se pueda continuar con el programa (*Programa Principal* sobre la Figura 2.5) que se encontraba ejecutando la CPU cuando se produjo la interrupción, debe ser posible una nueva bifurcación a la dirección de memoria a partir de la cual se encuentra la instrucción siguiente a la última que se ejecutó. La forma de realizar esta bifurcación es mediante la instrucción IRET (Retorno de Interrupción) al final de la RSI (Figura 2.5). Cuando la CPU ejecuta la instrucción IRET, recupera de la pila el valor del contador de programa y del registro de estado, en orden inverso a como fueron almacenados.
6. La CPU continúa ejecutando el programa principal (*Instrucción $i+1$* sobre la Figura 2.6).

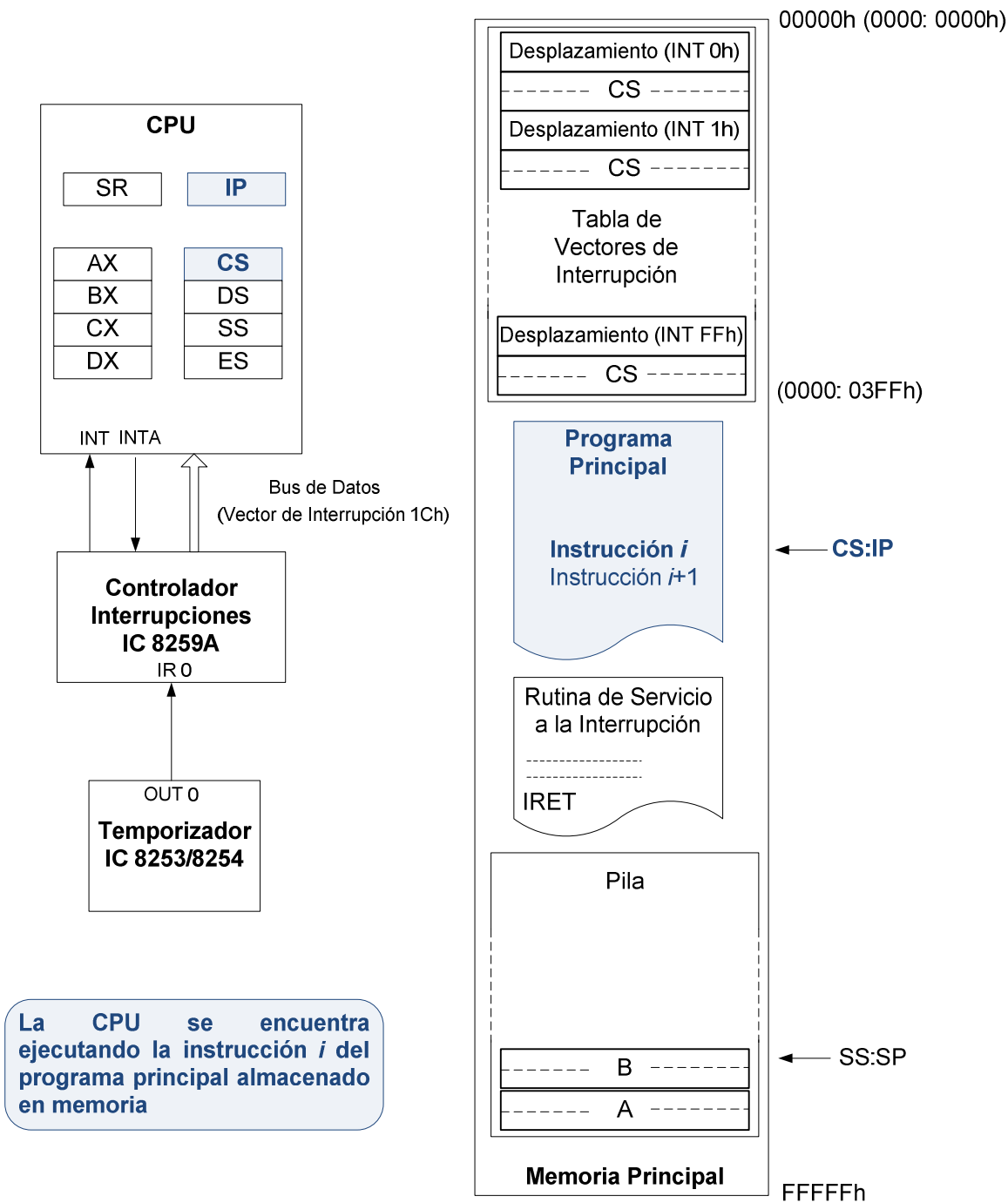


Figura 2.1. Situación previa a la generación de una interrupción.

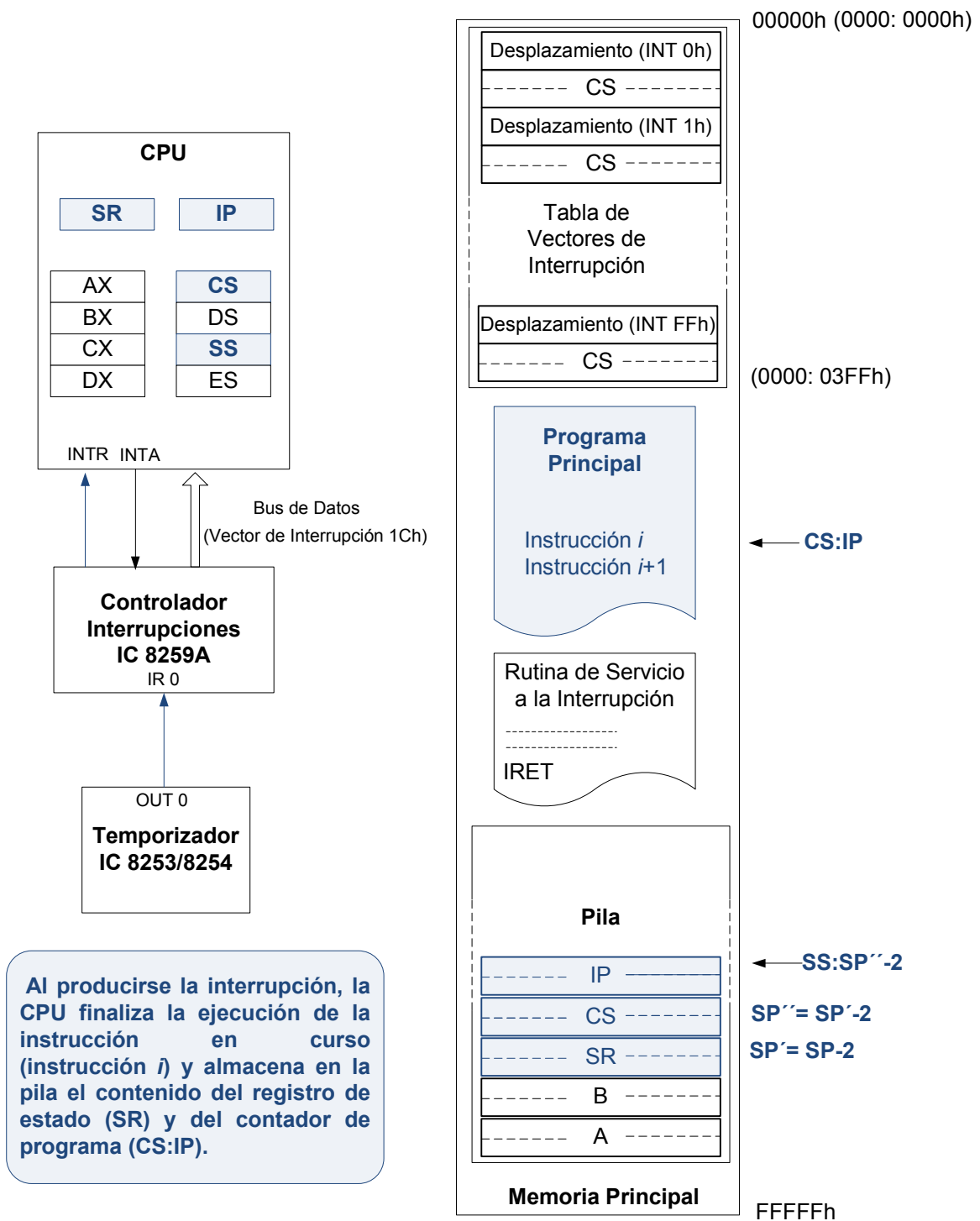


Figura 2.2. Generación de interrupción.

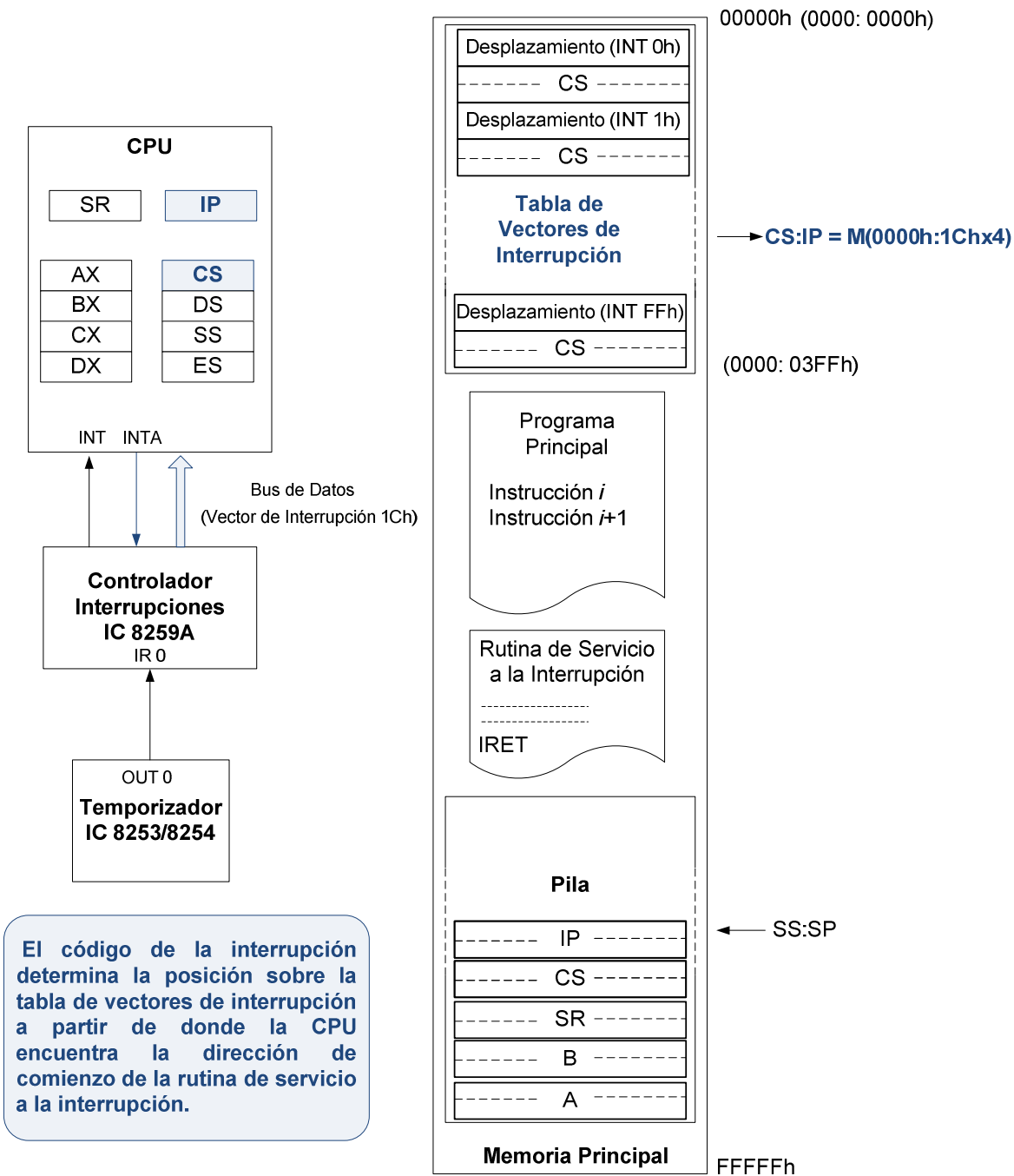


Figura 2.3. Aceptación de la interrupción y búsqueda de la dirección de comienzo de la RSI en la tabla de vectores de interrupción.

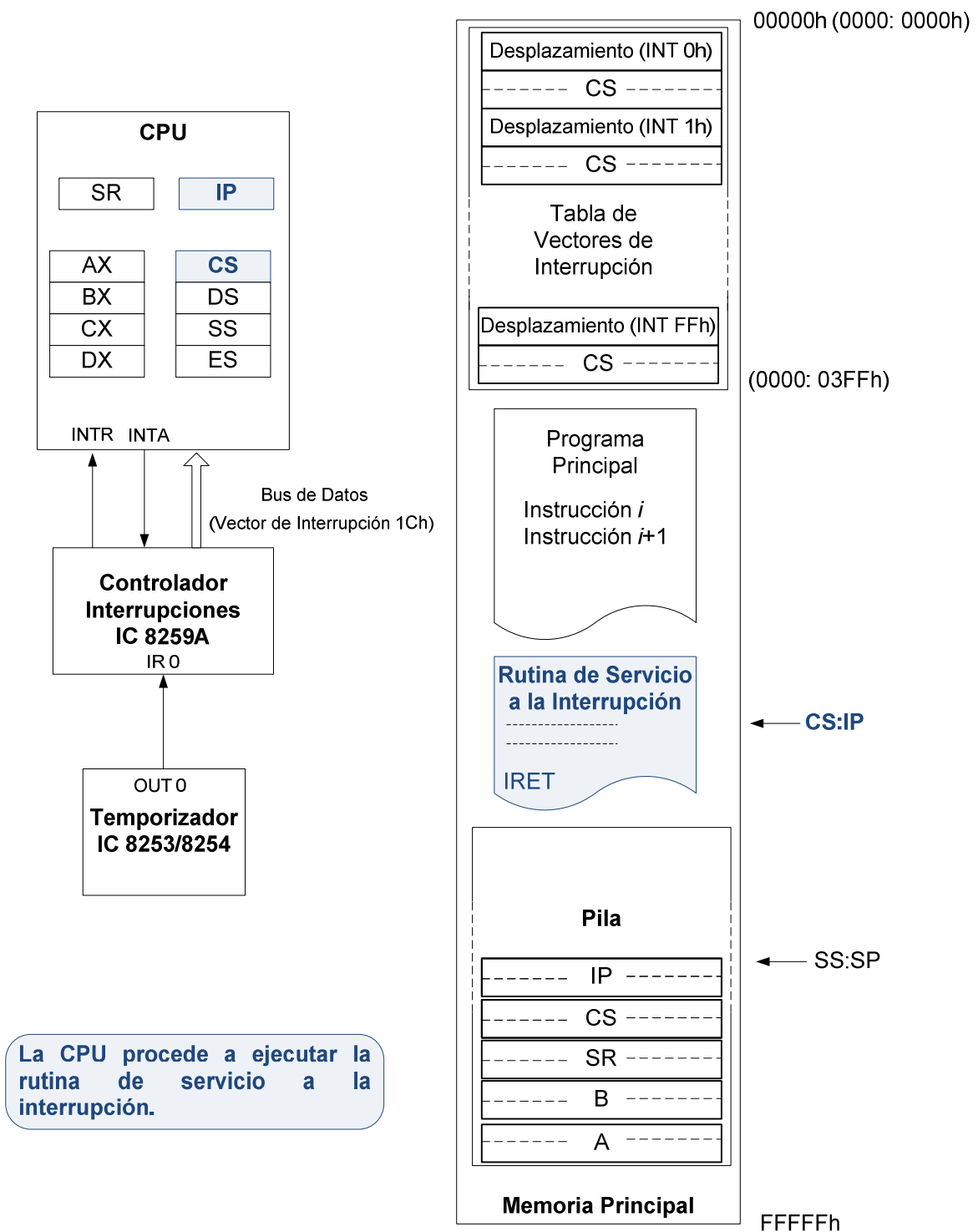


Figura 2.4. Bifurcación y posterior ejecución de la RSI.

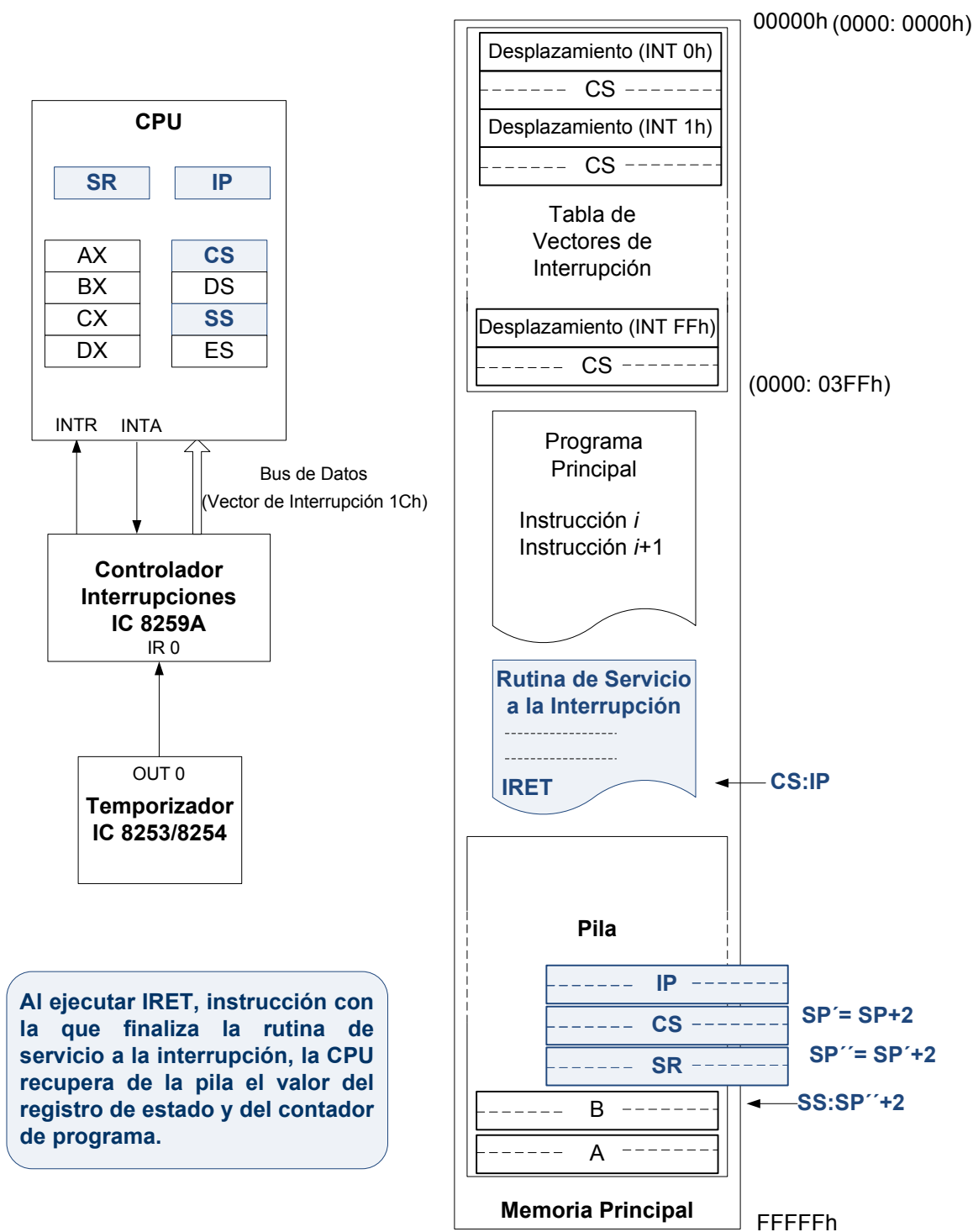


Figura 2.5. Finalización de la RSI y bifurcación al programa principal.

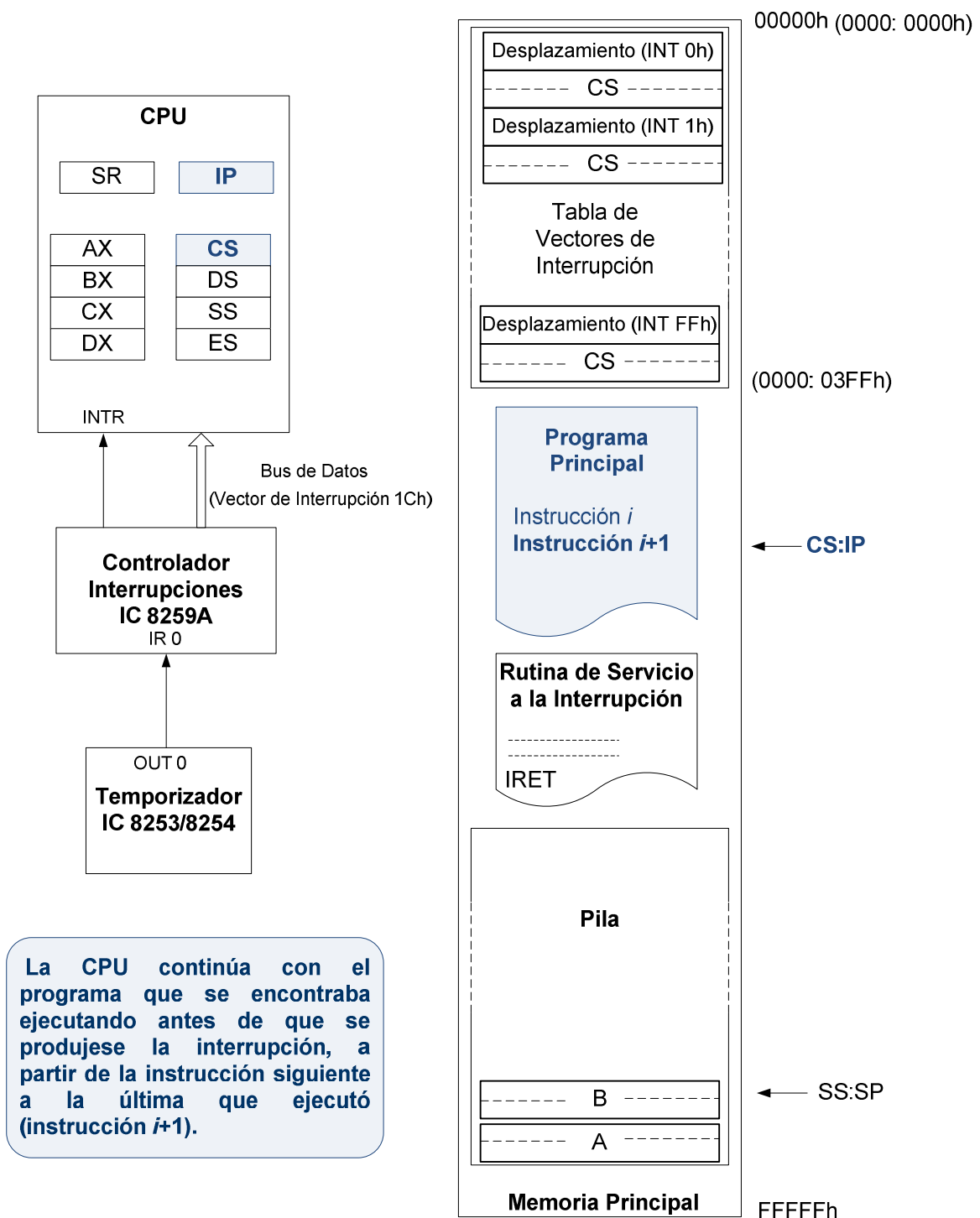


Figura 2.6. Vuelta a la ejecución del programa principal.

Ejercicio a desarrollar

La interrupción con código **1Ch** (generada a través de la patilla INTR del microprocesor), es una interrupción de tipo hardware provocada por el dispositivo externo temporizador IC 8253/8254. Este dispositivo dispone de un contador que varía de 0-65535 con una frecuencia de 1193180 ciclos/seg, provocando 18,2 interrupciones por segundo ($1193180/65536 = 18,2$). La interrupción provocada por el temporizador llega al controlador de interrupciones IC 8259, que activa la señal de petición de interrupción (INTR) para reclamar la atención de la CPU. El vector de interrupción (1Ch) es enviado a la CPU a través del bus de datos.

Realizar un programa en ensamblador del μP 8086 que se encargue de atender a la interrupción 1Ch provocada por el temporizador 8253 del μP 8086. Este programa será la Rutina de Servicio a la Interrupción (RSI) y se encargará de contar el tiempo, de forma que cada vez que transcurra 1 segundo, se muestre sobre la pantalla un cronómetro en formato de mm:ss (minutos:segundos) que se vaya incrementando a medida que avanza el tiempo. El mensaje mostrado en pantalla asociado al cronómetro será con texto blanco y fondo negro. Para ello se escribirá directamente sobre la memoria de vídeo.

Para hacer que la RSI esté ejecutándose permanentemente, el código correspondiente a la RSI debe quedar *residente* en memoria. Para hacer que un programa o parte de éste quede residente, se puede emplear la interrupción software INT 27h (ver apartado [2.2 Información útil](#)).

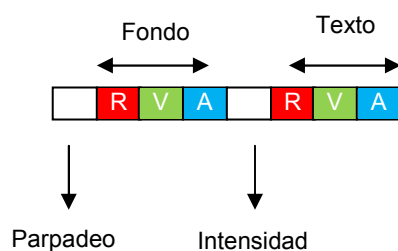
El proceso de linkado del programa residente debe hacerse para que el fichero resultante sea de tipo .COM (hasta ahora, después del proceso de linkado, todos los ficheros ejecutables generados han sido de tipo .EXE). Cuando se vayan a ejecutar los dos programas (programa principal tipo .EXE y programa residente tipo .COM), se ejecutará primero el programa residente y a continuación el programa principal. Justificar los cambios que aparecen durante la ejecución del programa principal (Apartado 1.2 - Práctica 1) cuando no estaba presente el programa residente.

Tomando como referencia las Figuras 2.1 a 2.6, comentar lo que sucede cuando se están ejecutando el programa principal y se provoca la interrupción 1Ch.

2.2. Información útil

- Direcciones de memoria principal:
 - Memoria de video en modo gráfico: dirección absoluta A0000h.
 - Memoria de video en modo texto: dirección absoluta B8000h.

- Organización de la pantalla en modo texto: 25/50 filas x 80 columnas.
- Interrupción software para terminar y dejar residente un código:
DX = dirección (desplazamiento respecto al segmento de código) de la instrucción siguiente a la última que debe quedar residente.
INT 27h
- Interrupción software para establecer el modo de pantalla
AL = modo (texto → 03h, gráfico → 13h)
AH = 0
INT 10h
- Interrupción software para capturar una tecla pulsada en teclado:
AH = 0
INT 16h
Se almacena en AX el código de la tecla pulsada
Código tecla ESC: 011Bh
- Proceso de linkado para que un programa sea de tipo .COM:
tlink /t nombre_fichero
- Asignación de atributos a caracteres en memoria de video:



- Estructura de un programa .COM, destinado a modificar el vector de interrupciones y a dejar residente una rutina de tratamiento: ver Figura 2.7.

```

.MODEL SMALL
.CODE
    org 100h

Programa_Int:

    JMP Reside
    ;;;; Declaración de variables

Rutina_Servicio PROC
    CLI
    ; Salvar el contenido de los registros implicados en la rutina
    .....
    .....
    ; Recupera el valor de los registros implicados en la rutina
    STI
    IRET
ENDP
Reside: ;etiqueta para determinar la dirección siguiente a la última de la rutina
        que debe quedar residente

        MOV DX, offset Rutina_Servicio
        MOV AX, 0
        MOV ES, AX
        MOV SI, Código_Int*4
        CLI
        MOV ES:[SI], DX
        MOV ES:[SI+2], CS
        STI
        MOV DX, offset Reside
        INT 27h

END Programa_Int
    
```

Figura 2.7. Estructura de un fichero que contiene una RSI y deja residente el código asociado a la misma.

2.3. Evaluación

Al finalizar la práctica el alumno debe entender y dominar los siguientes conceptos.

- Los adquiridos en las prácticas anteriores.

En relación a las rutinas de servicio a la interrupción:

- Interrupciones. Definición de interrupción. Definición y organización de la tabla de vectores de interrupción. Definición de una rutina de interrupción. Número de interrupciones diferentes que pueden ser atendidas según la tabla de vectores de interrupción. Número máximo de posiciones de la tabla de vectores de interrupción.

En relación a los programas residentes:

- Programas residentes: Definición de programa residente. Estructura de un programa residente.
- Ficheros .COM: Definición de fichero .COM. Estructura de un fichero .ASM para que pueda ser ensamblado como .COM.

En relación a la memoria de vídeo:

- ¿Qué es la memoria de vídeo?; ¿Qué es el modo gráfico?; ¿Qué es el modo texto?.
- ¿Cómo se accede a la memoria de vídeo en modo gráfico?.
- ¿Cómo está organizada la memoria de vídeo en modo gráfico (relación entre lo que se escribe en memoria y lo que aparece en pantalla)?.
- ¿Cómo se accede a la memoria de vídeo en modo texto?.
- ¿Cómo está organizada la memoria de vídeo en modo texto (relación entre lo que se escribe en memoria y lo que aparece en pantalla)?.

2.4. Referencias bibliográficas

- Morgan C. L., Waite M. (1988). *Introducción al microprocesador 8086/8088 (16 Bit)*, Madrid, McGraw-Hill.
- Charte Ojeda F. (2003). *Programación en ensamblador*, Madrid, Anaya.
- García de Celis, Ciriaco. *El universo digital del IBM PC, AT y PS/2*
<http://meltingpot.fortunecity.com/uruguay/978/>