contenido del registro IDTR. Generalmente, para el programador este hecho carece de importancia, ya que, después de un reset, el registro IDTR presenta una dirección base de 0 y un límite de 3FFH. Y esto coincide totalmente con las del procesador 8086. En modo real, sin embargo, se puede utilizar la instrucción LIDT para modificar los valores de la dirección base y del límite en el registro LDTR. En caso de producirse una interrupción y el contenido en la tabla de interrupciones indicara a una posición más allá de los límites que están almacenados en el registro IDTR, esto provocaría entonces un fallo doble.

6.2 MODO PROTEGIDO

Desde el punto de vista de un programador, la diferencia entre el modo protegido y el modo real está en el espacio de direccionamiento inmensamente ampliado y los mecanismos de direccionamiento distintos.

Modo protegido del 80286

El procesador 80286 permite dos modos de funcionamiento: el modo real de direccionamiento y el modo protegido. Trabajando en modo real 8086, el procesador 80286 puede utilizar hasta 1 Mbyte de espacio direccionable, memoria física, al emplear una dirección de 20 bits. En modo protegido, los programas que corren con el 80286 pueden utilizar incluso hasta 1 Gigabyte de espacio direccionable. Éste es automáticamente asignado al espacio direccionable de 16 Mbytes por el 80286, al emplear una dirección física de 24 bits. Desde el punto de vista del hardware, ambos modos se diferencian sólo en las cuatro líneas superiores de dirección (A23 hasta A20). Éstas son, en modo real, simplemente ignoradas o, en modo protegido, decodificadas junto a las otras líneas de dirección.

Como se apuntó anteriormente, el procesador 80286 puede direccionar una memoria física de hasta 1 Mbyte en modo real y en modo protegido hasta 16 Mbytes. Aparte de los diferentes tamaños de memoria, la organización de la memoria y de las E/S es, en el modo real y el modo protegido, idéntica.

Adicionalmente a las posibilidades de direccionamiento de memoria vistos, el 80286 puede direccionar directamente hasta 65536 puertos de E/S de 8 bits o hasta 32768 puertos de E/S de 16 bits, que estén asignados en una zona direccionable de E/S separada.

El programador ve el espacio direccionable de memoria del 80286 como una secuencia continua de bytes (8 bits), en donde cada byte con-

tiene un dato de 8 bits y 2 bytes seguidos un dato de 16 bits (palabra). Las informaciones byte o palabra pueden ser asignadas a direcciones pares o impares. No existe ninguna limitación para límites de palabras.

Modo protegido del 80386/80486

Cuando se realiza un reset el 80386/80486 arranca en modo real. Como se comentó anteriormente, el hacer trabajar a estos procesadores en modo real, es conveniente en aplicaciones que exigen un «rápido» 8086. La mayoría de los programas de aplicación, sin embargo, pueden correr de una manera más eficaz, cuando todos los recursos del procesador pueden ser agotados. Para ello el procesador debe ser llevado del modo real al modo protegido. El modo protegido permite el uso de instrucciones adicionales que optimizan especialmente los sistemas operativos multitarea, así como una paginación y segmentación de la memoria.

Como en el modo real, también en el modo protegido del 80386 se usan dos componentes para formar la dirección lógica: Se utiliza un selector de 16 bits, para determinar la dirección base lineal de un segmento. La dirección base, a su vez, es sumada a la dirección efectiva de 32 bits y forma una dirección lineal de 32 bits.

La dirección lineal de 32 bits puede ser utilizada como dirección física de 24 bits o, en caso de ser posible/permitido el acceso a páginas, el mecanismo de paginación convierte la dirección de 32 bits en una dirección física de 24 bits. La diferencia de ambos modos está en el cálculo de la dirección base.

El modo protegido amplía el espacio direccionable lineal hasta 4 Gbytes (2^{32}) y permite la ejecución de programas de memoria virtual con tamaños casi ilimitados (64 Tbytes, 2^{46}). Adicionalmente, el modo protegido permite a los microprocesadores 80386 y 80486 la ejecución de cualquier software, que fue escrito para las CPU 80386 DX, 80286 y 8086. Sin embargo, por ejemplo al utilizar el software de una CPU 80386 DX, sólo son utilizados 16 Mbytes de la memoria física.

Modo protegido del procesador Pentium

Cuando se realiza un reset, el procesador Pentium también arranca en modo real. Adicionalmente a las propiedades citadas, que pueden ser utilizadas en los procesadores 80386 y 80486 en modo protegido, las posibilidades del procesador Pentium en modo protegido fueron ampliadas con los nuevos señalizadores VIP y VIF en el registro EFLAG. VIP (*Virtual Interrupt Pending Flag* - señalizador de interrupciones

pendientes virtual) y VIF (*Virtual Interrupt Flag* - señalizador de interrupciones virtual) permiten que los programas de aplicación en un entorno de multitarea tengan acceso a una variante virtual del IF (*Interrupt Flag*). Con ello se evita, en ciertas operaciones, la ejecución paso a paso a un monitor del modo virtual 8086 o del modo protegido.

La inicialización del modo protegido

La inicialización es una secuencia de diferentes instrucciones al sistema operativo, antes que la primera tarea sea ejecutada, es decir, el código de inicialización del sistema operativo debe ser modificado adecuadamente. La inicialización genera estructuras de datos del sistema operativo y es independiente del 80386. Los siguientes procedimientos deben ser ejecutados, para llegar al modo protegido:

- Apertura del modo protegido
- Apertura de las páginas (a elección)
- Conmutación a las tareas originales

La mayoría de los sistemas operativos deben transferir casi simultáneamente los controles a un segmento de 32 bits, para adaptar los operandos estándar (por defecto) y el tamaño de dirección a 32 bits. Generalmente un sistema operativo cambia automáticamente del modo real al modo protegido y de un segmento de 16 bits a uno de 32 bits después de realizarse un RESET.

La tabla 6.2 muestra el contenido de los registros, después de realizarse un RESET.

Un RESET pone las líneas de dirección A31 hasta A20 en estado «*high*», para que, de acuerdo a ello, puedan ser llamados diferentes códigos. Estas líneas de dirección permanecen en estado «*high*» hasta que se produzca un salto entre segmentos o una petición equivalente. A31 hasta A20 permanecen en estado «*low*», mientras el procesador está en el modo protegido.

Las referencias a datos, que siguen a un RESET, son dirigidas a los primeros 64 Kbytes del espacio direccionable lineal (y físico). En consecuencia, después de un RESET los 64 Kbytes superiores del espacio direccionable lineal del 80286 están reservados para código y los 64 Kbytes inferiores para datos. Una simple manera de implementar una rutina RESET consiste en posicionar tanto los códigos como los datos en los 64 Kbytes superiores y utilizar un prefijo de sustitución de segmento CS para las referencias a datos. Con ello las direcciones de datos quedan en la parte superior del espacio direccionable. Esta sencilla rutina debe permanecer alejada de una transferencia entre segmentos,

Tabla 6.2 Contenido de los registros después de un RESET

Registro	*Valor*
EFLAGS	*Los bits definidos contienen 0; los bits no definidos contienen valores no definidos*
CR0	*Los bits definidos contienen 0, excepto ET (bit de tipo de extensión), bits no definidos contienen valores no definidos*
CS base	*FFFF0000H*
CS límite	*FFFFH*
EIP	*0000FFF0H*
DS–GS base	*00000000H*
DS–GS límite	*FFFFH*
EAX	*El resultado de autoprueba permanece indefinido*
EDX	*Número de componente y revisión*
El resto	*Indefinido*

mientras el procesador no haya pasado del modo real al modo protegido.

Al recibir RESET los valores de CS, EIP y las líneas de dirección A31-20, el 80386 toma su primera instrucción de la dirección lineal FFFFFFF0H. Puesto que la dirección de RESET está muy cerca del límite del segmento de código, la instrucción debería ser un salto entre segmentos a un *offset* inferior, en los 64 Kbytes del segmento de código. El intento de tomar una instrucción fuera del límite del segmento de código de 64 Kbytes, provoca una excepción de protección general.

6.3 MODO VIRTUAL

Desde la introducción del 80386, los procesadores soportan la ejecución de programas 8086, 8088, 80186 u 80188, también en un entorno de modo protegido. Esto hasta entonces no era posible. Por ello debía renunciarse, al ejecutar programas 8086, a las prestaciones que brinda el modo protegido (paginación, ampliación del espacio direccionable y, sobre todo, a la variedad de protecciones). Por este motivo, con el 386, el modo protegido fue ampliado con el modo virtual 8086. Un programa del 8086 corre en modo virtual, como parte de una tarea virtual. Una tarea es un programa que ha sido cargado, el cual es controlado con ayuda del sistema operativo y tiene su propio entorno. Las tareas virtua-

les utilizan la ventaja ofrecida por el modo protegido del soporte de hardware a la multitarea. Por tanto, no sólo pueden correr varias tareas virtuales, en donde en cada tarea puede correr un programa 8086, sino que éstas pueden correr simultáneamente con otras tareas (multitarea).

Una tarea virtual 8086 consta de un programa 8086, que debe ejecutarse, y de un código especial (de un software especial) que sirve como monitor de máquina virtual. La tarea debe estar representada por medio de un TSS (TSS del 386, 486 o Pentium). El procesador entra al modo virtual, ejecuta el programa del 8086 y, posteriormente, retorna al modo protegido. De esta forma, el monitor puede continuar trabajando o, por ejemplo, realizar otras tareas (las de un 80386 u 80486).

Para poder trabajar en modo virtual, el programa 80386 en curso necesita software de modo virtual 8086 y los servicios del sistema operativo. El software de modo virtual es un código de modo protegido que corre en el nivel de prioridad 0 (más privilegiado). El software consta, en gran parte, de procedimientos de inicialización y gestión de excepciones. Igual que para otros programas, los descriptores de segmentos para el software deben estar en las tablas de descriptores locales o globales (GTD o LTD). Las direcciones lineales por encima de 10FFEFH están disponibles para el software virtual 8086, el sistema operativo y otros softwares de sistema. Además, el software virtual necesita descriptores de segmento de datos para que la tabla de vectores de interrupción, u otras partes del programa 8086, puedan ser controladas en el primer megabyte del espacio direccionable.

Generalmente existen dos posibilidades de implementar el sistema operativo 8086:

1. El sistema operativo 8086 corre como parte del programa 8086. Esta manera es recomendable puesto que, de lo contrario, el código de aplicación 8086 modificaría el sistema operativo. Mayoritariamente tampoco no hay tiempo de desarrollo disponible para reimplementar el sistema operativo 8086 como un sistema operativo 386, 486 o Pentium.

2. El sistema operativo 8086 puede ser implementado o emulado en el software virtual 8086. Esta manera es igualmente recomendable, puesto que entonces las funciones del sistema operativo se dejan coordinar más fácilmente con todas las tareas virtuales 8086 y, también, las funciones del sistema operativo 8086 pueden ser mayoritariamente emuladas sin dificultad por llamadas, por ejemplo, a un sistema operativo 486.

En las posibilidades de implementación, hay que tener en cuenta que el sistema operativo 8086 puede presentar distintas tareas virtuales 8086, puesto que también se utilizan distintos sistemas operativos 8086.

El hardware brinda un *set* virtual de registros (a través de TSS), un espacio de memoria virtual (el primer megabyte del espacio direccionable lineal de las tareas) y ejecuta directamente todas las instrucciones que tienen que ver con estos registros y espacios de dirección.

Software virtual

El software controla las interfaces externas de la máquina virtual (E/S, interrupciones y excepciones) de tal manera, que el gran entorno en el que corre sea consistente. En caso de E/S, el software puede elegir si las instrucciones de E/S son emuladas o si el hardware las ejecuta directamente sin intervención del software. Al software que soporta máquinas virtuales se le llama monitor virtual 8086.

El procesador se encuentra en modo virtual en el momento en que el bit VM *(Virtual Machine)* en el registro EFLAG es activado. El procesador supervisa este señalizador bajo dos condiciones:

1. En el momento en que los registros de segmento son cargados. Con esto, el procesador determina si se utilizan formaciones de direcciones de tipo 8086.
2. En el momento en que se decodifican instrucciones. Por medio de esto, el procesador determina si las instrucciones reaccionan a IOPL.

Aparte de estos dos cambios de las operaciones normales, los procesadores que soportan el modo virtual trabajan igual en modo virtual que en modo protegido.

Registros en modo virtual

Los registros disponibles en modo virtual son los mismos de un microprocesador 8086, incluyendo los nuevos registros de la familia de microprocesadores Intel 80XXX. Éstos son los registros FS, GS, de depuración, de control y de prueba. Igualmente están disponibles las nuevas instrucciones que trabajan con FS y GS. Además, los prefijos de sustitución de segmento pueden ser utilizados para que los registros FS y GS puedan ser empleados en el cálculo de dirección. Las instrucciones pueden utilizar prefijos de operandos de 32 bits, al ser empleado el tamaño del operando.

Formación de dirección en modo virtual

Trabajando en modo virtual, los procesadores 80386, 80486 y Pentium no interpretan los selectores 8086 como si éstos se refirieran a descriptores, sino que forman las direcciones de igual manera como lo

Tabla 6.3 La utilización de registros

Registro	**Uso en modo real**		**Uso en modo protegido**		**Uso en modo virtual 8086**	
	Carga	*Almacena*	*Carga*	*Almacena*	*Carga*	*Almacena*
Registro general	*sí*	*sí*	*sí*	*sí*	*sí*	*sí*
Registro de segmentos	*sí*	*sí*	*sí*	*sí*	*sí*	*sí*
Registro de señalizadores	*sí*	*sí*	*sí*	*sí*	*IOPL**	*IOPL**
Registro de control	*sí*	*sí*	*PL = 0*	*PL = 0*	*no*	*sí*
GDTR	*sí*	*sí*	*PL = 0*	*sí*	*no*	*sí*
IDTR	*sí*	*sí*	*PL = 0*	*sí*	*no*	*sí*
LDTR	*no*	*no*	*PL = 0*	*sí*	*no*	*no*
TR	*no*	*no*	*PL = 0*	*sí*	*no*	*no*
Registro de control de depurado	*sí*	*sí*	*PL = 0*	*PL = 0*	*no*	*no*
Registro de prueba	*sí*	*sí*	*PL = 0*	*PL = 0*	*no*	*no*

haría el 8086. Desplazan el selector en 4 bits hacia la izquierda, para formar una dirección base de 20 bits. La dirección efectiva es entonces ampliada en 4 bits, por la parte de los bits de mayor peso y sumada a la dirección base. De esta manera se crea una dirección lineal.

Puesto que existe la posibilidad de un acarreo, la dirección resultante puede presentar 21 bits significativos. Un programa 8086 puede crear direcciones en cualquier lugar del espacio direccionable, en la zona de 0 hasta 10FFEFH (1 Mbyte más, aproximadamente, 64 Kbytes).

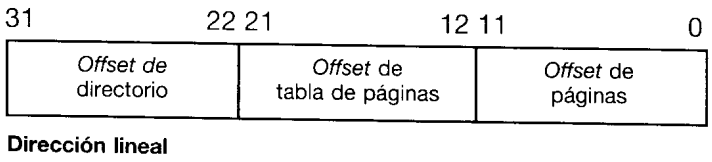

Fig. 6.2 Formato de una dirección lineal para una página de 4 Kbytes

Las tareas virtuales 8086 crean direcciones lineales de 32 bits. Sin embargo, ya que un programa 8086 sólo puede utilizar los 21 bits de menor peso de una dirección lineal, la dirección lineal puede ser asignada a cualquier dirección física utilizando el *paging* (paginación). A diferencia del procesador 8086 y 80286, desde la aparición del procesador 80386 pueden ser creadas direcciones efectivas de 32 bits utilizando un prefijo de sustitución de dirección. Sin embargo, en modo real una dirección de 32 bits no puede sobrepasar el valor de 65535 sin provocar una excepción. Para garantizar una total compatibilidad con el modo real 80286, se originan los llamados seudofallos (interrupción 12 o 13 sin código de error), cuando se presenta una dirección efectiva fuera de la zona limitada entre 0 y 65535.

Paginación en tareas virtuales 8086

Una paginación para tareas virtuales 8086 aisladas no es necesaria; sin embargo, según las siguientes razones no sólo puede ser práctica, sino incluso necesaria:

En la creación de varias tareas virtuales 8086.

Cada tarea debe asignar el megabyte inferior de una dirección lineal a diferentes posiciones físicas.

En la emulación de una dirección que queda cerca al megabyte.

Con los procesadores de la familia 8086 es posible especificar direcciones más grandes que 1 Mbyte. Ejemplo: Con un selector de 0FFFFH y un *offset* de 0FFFFH, la dirección efectiva es 10FFEFH (1 Mbyte más 65519 bytes). El procesador 8086, sin embargo, sólo puede formar direcciones hasta una longitud de 20 bits. Por este motivo, simplemente ignora el bit de mayor peso y «mutila» la dirección a 0FFEFH. Sin embargo, el procesador 80386, por ejemplo, no puede mutilar así una dirección. En caso de depender un programa 8086 de una formación de dirección semejante, se puede conseguir el mismo efecto asignando una memoria lineal entre 100000H y 110000H y una memoria lineal entre 0 y 100000H, a la misma memoria física.

En la creación de un espacio de direccionamiento virtual más grande que el espacio direccionable físico.

En caso de ser utilizado en común por el sistema operativo o la ROM, el código similar de los distintos programas 8086, ejecutables en multitarea.

En caso de presentarse una bifurcación o una ejecución paso a paso, referente a dispositivos de E/S asignados a la memoria.

Protección dentro de una tarea virtual

No se ha suministrado una protección entre los segmentos de un programa 8086. Para proteger el software de sistema que corre dentro de una tarea virtual 8086 del programa de aplicación 8086, el desarrollador de software debería tener en cuenta lo siguiente:

- Los primeros megabytes (más 64 Kbytes) del espacio direccionable lineal de cada tarea serán reservados para el programa 8086. Una tarea 8086 no puede generar direcciones fuera de esta zona.
- Los bits U/S *(User/Supervisor)* de las entradas a las tablas de página serán utilizados para que el software virtual y otros software de sistema estén protegidos en cada espacio de tarea virtual 8086. En el momento en que el procesador se encuentre en el modo virtual 8086, CPL será 3 (es decir, la menor prioridad). Por esta razón, un programa 8086 tiene sólo la prioridad de un usuario. En caso de que las páginas del software virtual tengan prioridad de supervisor, el programa 8086 no podrá acceder más a ellas.

Entrada y salida del modo virtual

Se entra al modo virtual, activando el señalizador VM. Para ello existen dos posibilidades, que serán explicadas brevemente tomando como ejemplo al 80386:

1. Una conmutación de tarea a una tarea 80386, carga el contenido de los EFLAGS de la nueva TSS. Pero la TSS de la nueva tarea debe ser la de un microprocesador 80386 y en ningún caso la de un 80286. El motivo para ello es que la TSS del 80286 no carga las palabras de mayor peso de los EFLAGS; sin embargo, son éstas las que contienen el señalizador VM necesario.

Un señalizador VM activo en el nuevo contenido de los registros EFLAG indica que la nueva tarea ejecuta instrucciones 8086. En el momento en que los registros de segmento son cargados por la TSS, el microprocesador 80386 genera las direcciones base, de la misma manera que un 8086.

2. Una instrucción IRET de una subrutina de una tarea 80386 DX carga los registros EFLAG desde la pila. Un señalizador VM activo indica a la subrutina que, al retornar al programa interrumpido, debe volver a un programa 8086. En el instante en que la instrucción IRET es ejecutada, CPL debe ser 0. De lo contrario el procesador no cambia el estado del señalizador VM.

En el momento en que se utiliza una conmutación de tarea, para entrar al modo virtual, los registros de segmento son cargados desde

una TSS. Sin embargo, si se utiliza una instrucción IRET para activar el señalizador VM, entonces los registros de segmento conservan el contenido, que fue cargado durante el modo protegido. El software debería, por ello, cargar estos registros con el contenido de los selectores de segmento, que corresponden al modo virtual.

El procesador deja el modo virtual en el momento en que se produce una interrupción o una excepción. Aquí existen igualmente dos posibilidades. También éstas serán brevemente comentadas, tomando como ejemplo al procesador 80386:

1. La interrupción o la excepción provocan una conmutación de tarea. Una conmutación de tarea, de una tarea virtual a otra tarea, provoca que los registros EFLAG sean cargados por la TSS de la nueva tarea. En caso de que la nueva TSS sea una TSS 80386 DX, y de que el señalizador VM esté borrado en el nuevo contenido de los EFLAGS, entonces el procesador borra el señalizador VM de los registros EFLAG. Lo mismo pasa cuando la nueva TSS es una TSS 80286. Los registros de segmento son cargados por la nueva TSS con un formato de dirección 80386 y la ejecución de la nueva tarea comienza en el modo protegido 80386.

2. La interrupción o la excepción llaman a un procedimiento (una subrutina) con el nivel de prioridad 0. El procesador carga entonces el contenido actual de los registros EFLAG en la pila y borra el señalizador VM. El gestor de interrupción o excepción corre como código «nativo» del microprocesador 80386 en modo protegido. En caso que una interrupción o una excepción llame a un procedimiento en un segmento equivalente, o a un segmento con un nivel de prioridad diferente a 0. entonces el procesador genera una excepción de protección general. El código de error es el selector del segmento de código que debía ser llamado.

El software de sistema no cambia directamente el estado del señalizador VM. Cambia más bien el estado en la copia de los registros EFLAG, que está almacenado en la pila o en la TSS. El software virtual 8086 activa el señalizador VM en la copia del EFLAG, la cual está en la pila o en la TSS, en el momento en que se crea una tarea virtual. El gestor de interrupción o excepción puede supervisar al señalizador VM en la pila. En caso de que la subrutina de interrupción se ejecutara en modo virtual, entonces el gestor se deberá llamar al software virtual.