
MODELO DEL PENTIUM PARA EL PROGRAMADOR DE SISTEMAS

11

11.1.- Registros del sistema	1
11.2.- Registros de segmento	3
11.3.- Registros de tablas	5
11.4.- Registro de tarea	6
11.5.- Registro de señalizadores	8
11.6.- Registro de control	9
11.6.1.- CR0: Doble palabra de estado de la máquina	9
11.6.2.- CR2: Dirección lineal del fallo de página	11
11.6.3.- CR3: Base del directorio de las tablas de páginas	11
11.6.4.- CR4: Extensiones de la arquitectura	11
11.7.- Registros de depuración	12
11.8.- Registros de prueba de la TLB	14
11.9.- Registros de específicos	16

11.1- REGISTROS DEL SISTEMA

En un entorno protegido y multitarea como son el Modo Protegido y el Modo Virtual 86, es indispensable que el programador del sistema posea un conocimiento profundo de todos los mecanismos de protección de la memoria. Además de saber manejar los recursos de la CPU destinados al desarrollo de aplicaciones, dicho programador debe controlar los que tienen un carácter especial y permiten obtener el máximo rendimiento del Pentium en el Modo Protegido. Dichos recursos son destinados a manejar:

- Memoria Virtual (Segmentación y Paginación).
- Entorno Protegido (Reglas de acceso).
- Multitarea (Conmutación de tarea).
- Control del flujo de instrucciones.
- Manejo de interrupciones y excepciones.
- Gestión de la caché.
- Pruebas y autochequeos.

En la figura 11.1 se muestra un esquema que abarca el conjunto de registros del sistema, usados por el programador de sistemas. Obsérvese en dicha figura que aparecen los registros de segmento (CS, DS, SS, ES, FS y GS), cuya misión también se describió al analizar los registros del programador de aplicaciones. El motivo de su presencia en ambos conjuntos de registros se debe a que, si bien el programador de aplicaciones puede modificar en sus programas el contenido de los mismos, cuando la CPU opera en Modo Protegido, se accede a través de ellos a las tablas de descriptores de segmento y se obtienen los parámetros que se cargan, de forma automática y transparente, en los registros caché ultrarápidos e invisibles, sirviendo de base para el direccionamiento y gestión de la memoria virtual.

Igualmente, se ha incluido en la figura 11.1 otro registro presentado dentro del conjunto que maneja el programador de aplicaciones. Se trata del registro de señalizadores EFLAGS también llamado registro de estado. La razón se debe a que en él hay varios bits que son propios en el control del sistema:

- **IOPL:** Nivel de privilegio de E/S. Controla el acceso al espacio de E/S.
- **NT:** Tarea anidada.
- **RF:** Flag de reanudación. Invalida temporalmente los fallos de depuración.
- **VM:** Modo Virtual-86.
- **AC:** Chequeo de alineamiento.
- **VIF:** Flag de interrupción en modo virtual.
- **VIP:** Flag de interrupción pendiente en modo virtual.
- **ID:** Flag de permiso de identificación del procesador.

Los restantes bits del registro de estado (EFLAGS) se usan en el desarrollo de las aplicaciones de usuario.

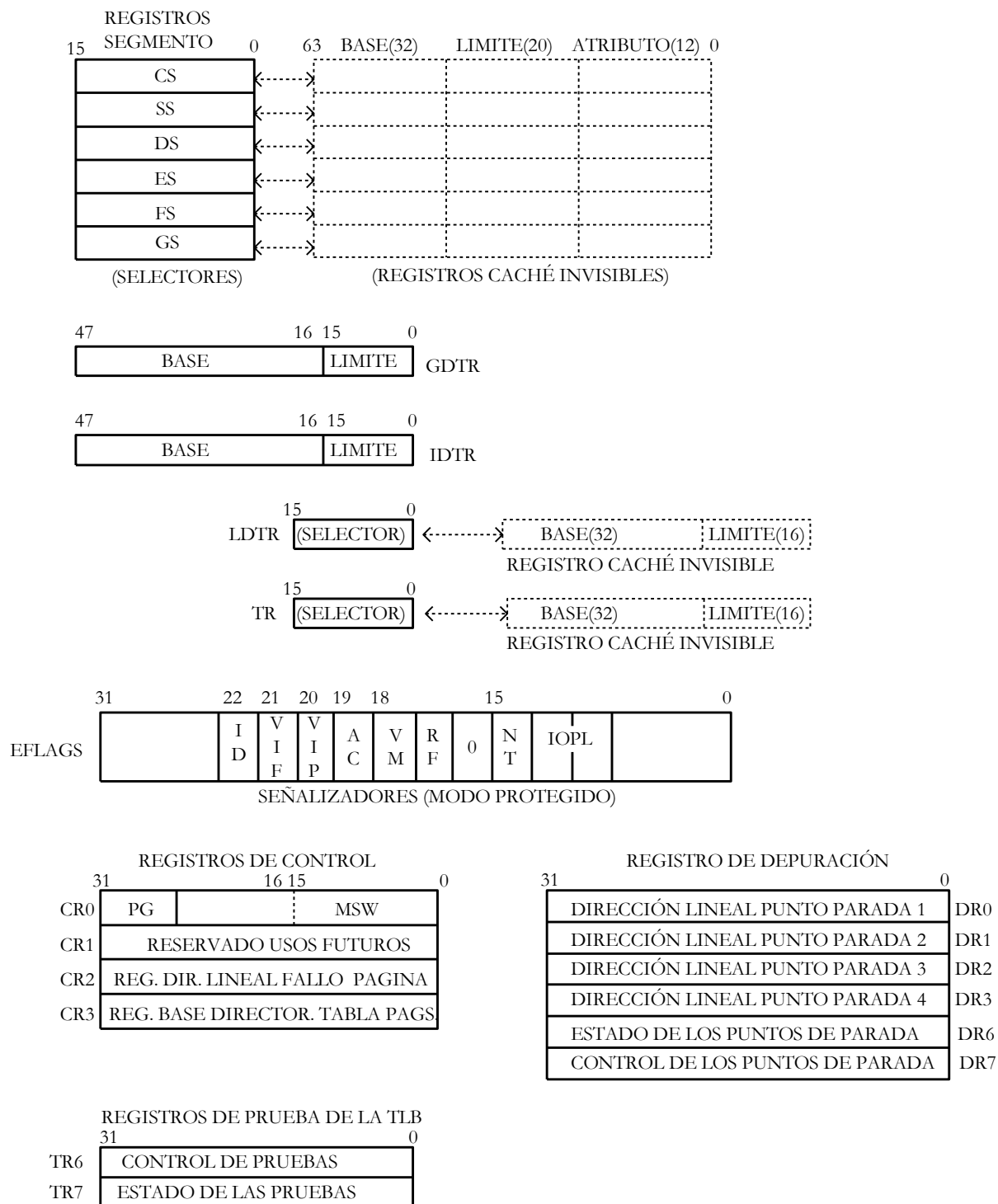


Fig. 11.1. Registros del sistema.

11.2- REGISTROS DE SEGMENTO

Los procesadores Intel utilizan la segmentación como método principal y obligatorio en la gestión de la memoria, mientras que la paginación es optativa.

Los segmentos quedan definidos por su dirección de inicio o base, su tamaño o límite y sus atributos o derechos de acceso.

En Modo Real, la base se calcula multiplicando por 16 (en binario multiplicar por 16 significa añadirle 4 ceros por la derecha) el contenido del registro correspondiente, ya que los registros de segmento son de 16 bits y dado que en Modo Real solo se utiliza el primer MB de la memoria, solo se necesitan 20 bits para direccionar la memoria ($2^{20} = 1$ MB). El límite es fijo y de 64 KB de valor, y los atributos no se utilizan en este modo de trabajo, porque en Modo Real no hay sistema de protección y todos los accesos están permitidos.

En Modo Protegido, los segmentos se especifican mediante estructuras de datos de 8 bytes (64 bits), denominadas descriptores, que están agrupados en tablas. Cada descriptor consta de los siguientes campos:

- Base del segmento (32 bits).
- Límite del segmento (20 bits).
- Atributos (12 bits).

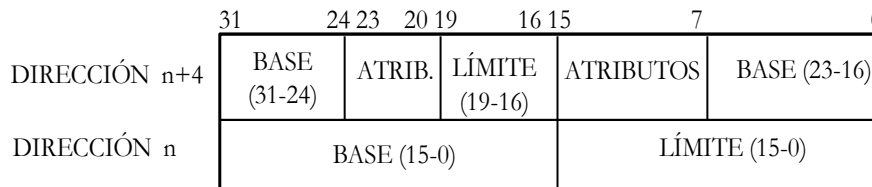


Fig. 11.2. Formato de un descriptor de segmento compuesto por ocho bytes, que ocupan dos posiciones de memoria cada una.

En cada instante, la CPU maneja los segmentos propios de la tarea en curso, que se hallan situados en el área local de dicha tarea. También puede manejar los segmentos compartidos por todas las tareas que se ubican en el área global.

La tabla GDT (Tabla de Descriptores Globales) dispone de tantas entradas como descriptores de segmentos existan en el área global. Dicha tabla tiene un tamaño de 64 KB, por lo que admite hasta 8 K descriptores. Cada descriptor de dicha tabla tiene 64 bits.

Los descriptores de los segmentos específicos de la tarea en curso están reunidos en otra tabla, llamada LDT, Tabla de Descriptores Locales, de la misma capacidad que la GDT, y que al igual que esta última, debe residir en la memoria principal desde el preciso instante en que se pasa al Modo Protegido, para que la CPU pueda localizar el descriptor del segmento a ejecutar.

Entre la GDT y LDT, la CPU puede controlar una memoria virtual de 16 K descriptores de segmento. Puesto que cada segmento puede alcanzar un tamaño máximo de 4 GB, ya que el límite de los segmentos tiene 32 bits y $2^{32} = 4$ GB, teóricamente la capacidad máxima de la memoria virtual que puede manipular un Pentium llegaría a los 64 TB.

En la figura 11.3 se recuerda, de forma gráfica, la actuación del contenido de los registros de segmento, esto es, es el selector del descriptor de un segmento en la GDT o en la LDT.

En el campo RPL del registro de segmento viene el valor del nivel de privilegio del segmento que ha solicitado la intervención del segmento bajo análisis, es decir, es el nivel de privilegio (PL) del segmento que se encuentra en el registro CS. TI, es el Indicador de Tabla, de forma que si TI=0, el descriptor se encuentra en la GDT, y si TI=1, en la LDT.

Los 13 bits de mas peso del selector actúan como índice que referencia una entrada en la tabla seleccionada por TI. Como cada descriptor consta de 8 bytes, el valor del índice se multiplica por 8 para obtener la dirección de inicio del descriptor seleccionado.

Una vez que se ha seleccionado la entrada en la tabla de descriptors, la información contenida en esta se carga en un registro caché oculto ultrarápido que esta asociado al registro de segmento, esta información constará de los 64 bits que hacen falta para direccionar en Modo Protegido, 32 bits para la base, 20 bits para el límite y 12 bits para los atributos, con lo que el registro de segmento correspondiente ya podrá direccionar la memoria.

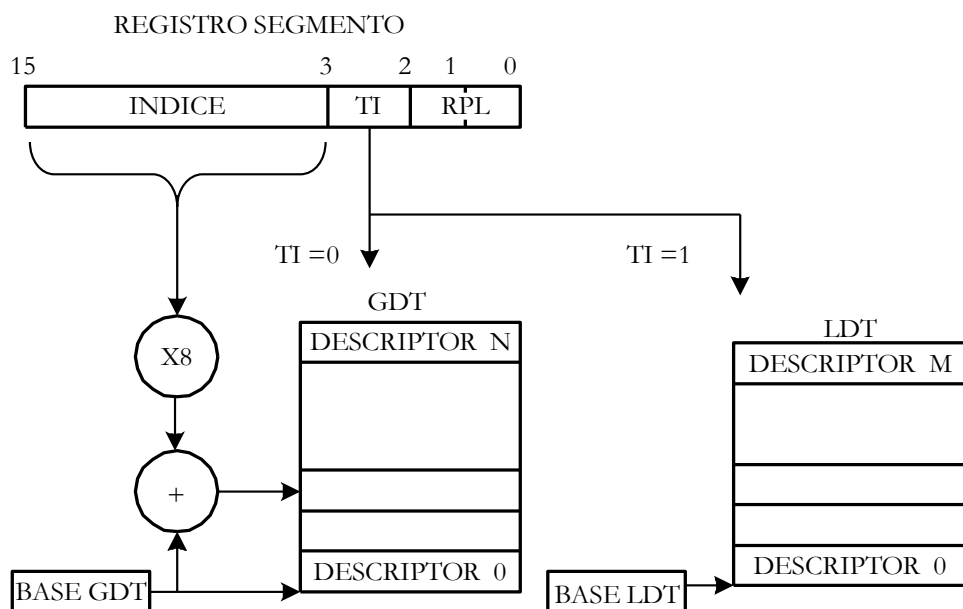


Fig. 11.3. El contenido del registro segmento actúa, en Modo Protegido, como un selector de una entrada en la GDT o en la LDT.

11.3- REGISTROS DE TABLAS

Existen tres registros dedicados a la manipulación de las tablas de descriptores:

1. **GDTR:** Registro de la base de la GDT.
2. **IDTR:** Registro de la base de la IDT (Tabla de Descriptores de Interrupción).
3. **LDTR:** Registro del selector del descriptor de la LDT.

Los registros GDTR e IDTR tienen un tamaño de 48 bits, 32 de ellos están destinados a contener la dirección inicial de la tabla correspondiente (base), y los restantes 16, los de menos peso, para indicar el tamaño, que puede alcanzar un máximo de 64 KB. ($2^{16}=64$ KB)

El GDTR contiene la dirección de la base de la GDT y el tamaño de la misma. Las direcciones de las entradas de las tablas de descriptores deben ser múltiplos de 8. Para manipular este registro se utilizan las instrucciones LGDT y SGDT, para cargarlo y almacenarlo, respectivamente.

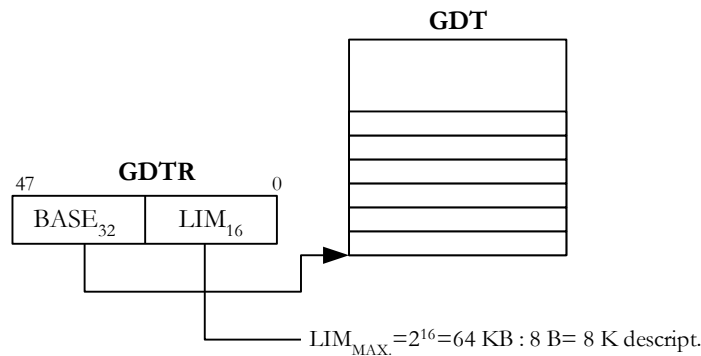


Fig. 11.4. Funcionamiento del registro GDTR.

El IDTR almacena el valor de la base de la IDT, así como su límite. Los descriptores contenidos en la tabla IDT se utilizan cuando se producen interrupciones y excepciones. La tabla IDT, como se estudiará mas adelante, puede contener hasta 256 descriptores de 8 bytes, por lo que su tamaño en Modo Protegido, alcanza los 2 KB. Las entradas de esta tabla en Modo Real no contienen lo mismo que en Modo Protegido.

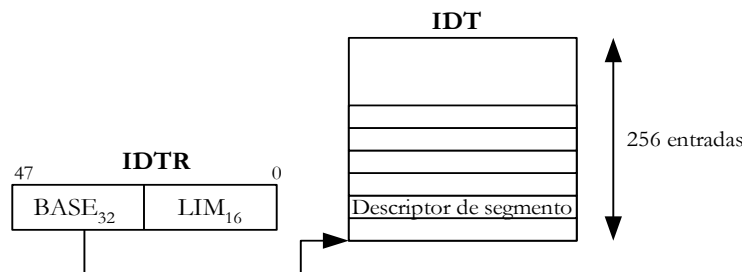


Fig. 11.5. Funcionamiento del registro IDTR.

El registro LDTR solo consta de 16 bits y actúa como un selector de un descriptor de segmento de la GDT. El segmento al que referencia el descriptor contiene la Tabla de Descriptores Locales, LDT, que guarda los descriptores de los segmentos de una tarea. Habrá, por tanto, tantas tablas de descriptores LDT como tareas haya ejecutándose en el sistema. Sin embargo, solo hay un registro LDTR, el cual apuntará siempre a la base de la LDT de la tarea en curso. Para cargar y almacenar este registro se usan las instrucciones LLDT y SLDT respectivamente.

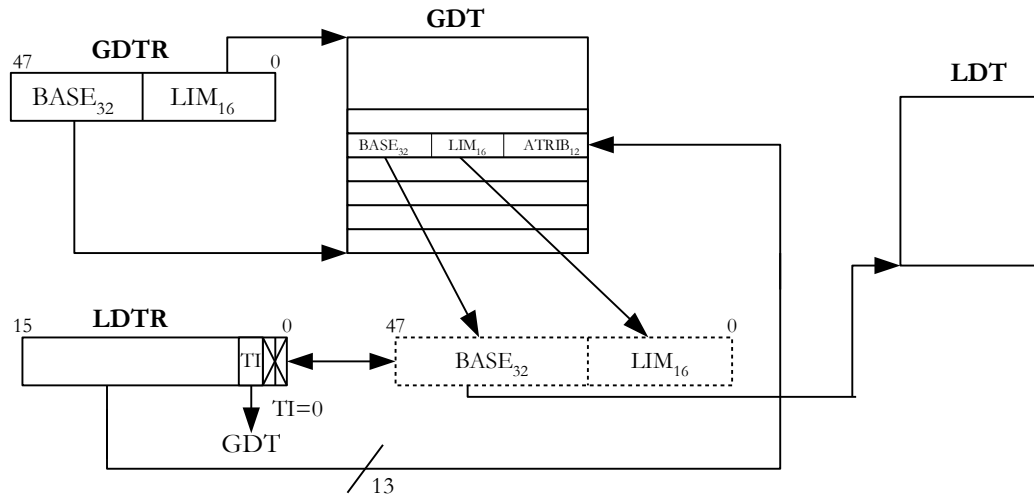


Fig. 11.6. Funcionamiento del registro LDTR.

Con el valor que hay en LDTR, tratándolo como un selector, se localiza uno de los descriptors de la GDT, cuya información (base, límite y atributos), carga la CPU automáticamente en el registro caché invisible asociado a LDTR, que se encarga de determinar la LDT de la tarea en curso. Para que esto se pueda llevar a cabo, en el bit TI del registro LDTR, el cual es el índice de tabla hay un cero, con lo que se indica que el selector que contiene ese registro en sus 13 bits de mayor peso apunta a la GDT.

Los segmentos que contienen las LDT de cada tarea deben tener un descriptor de segmento, cada uno, en la GDT.

Antes de entrar al Modo Protegido hay que asegurarse de que en la memoria principal residen la GDT y la IDT, para que la CPU pueda localizar los segmentos que precisa y atender las excepciones que se produzcan.

11.4- REGISTRO DE TAREA

El registro de tarea (TR), posee 16 bits y actúa, al igual que el LDTR, como selector de un descriptor de un segmento en la GDT. Dicho segmento recibe el nombre de Segmento de Estado de la Tarea (TSS) y guarda el contexto del procesador necesario para reanudar esa tarea.

Se entiende por contexto de la CPU para una tarea, la información que existe en todos los registros del procesador y que se precisa para poder reanudar la tarea en la misma situación en que se abandonó cuando se salvo el contexto.

El contexto de una tarea del Pentium está formado por el valor de los registros de propósito general, el del registro de estado (EFLAGS), EIP, registros de segmento, etc. En la figura 11.4 se muestra la actuación de TR y el contenido del segmento TSS, al que referencia indirectamente TR, a través de la GDT. Apréciase como TR dispone de un registro caché invisible donde la CPU carga los parámetros que definen al segmento TSS.

Para que un sistema pueda llevar a cabo la multitarea, debe poder ejecutar una operación llamada conmutación de tareas, la cual consiste en salvar el contexto de la tarea en curso, guardándola para poder ejecutarla posteriormente, y cargar el contexto de la nueva tarea que se quiera ejecutar, esto es, cada vez que la CPU abandona una tarea para iniciar la ejecución de otra, se salva automáticamente, el contexto actual del procesador en el TSS de la tarea saliente. Después el contenido del TSS de la tarea entrante se carga en los registros de la CPU, conformando el nuevo contexto de trabajo.

Los segmentos que contienen los TSS de cada tarea en curso tienen que tener un descriptor, cada uno, en la GDT.

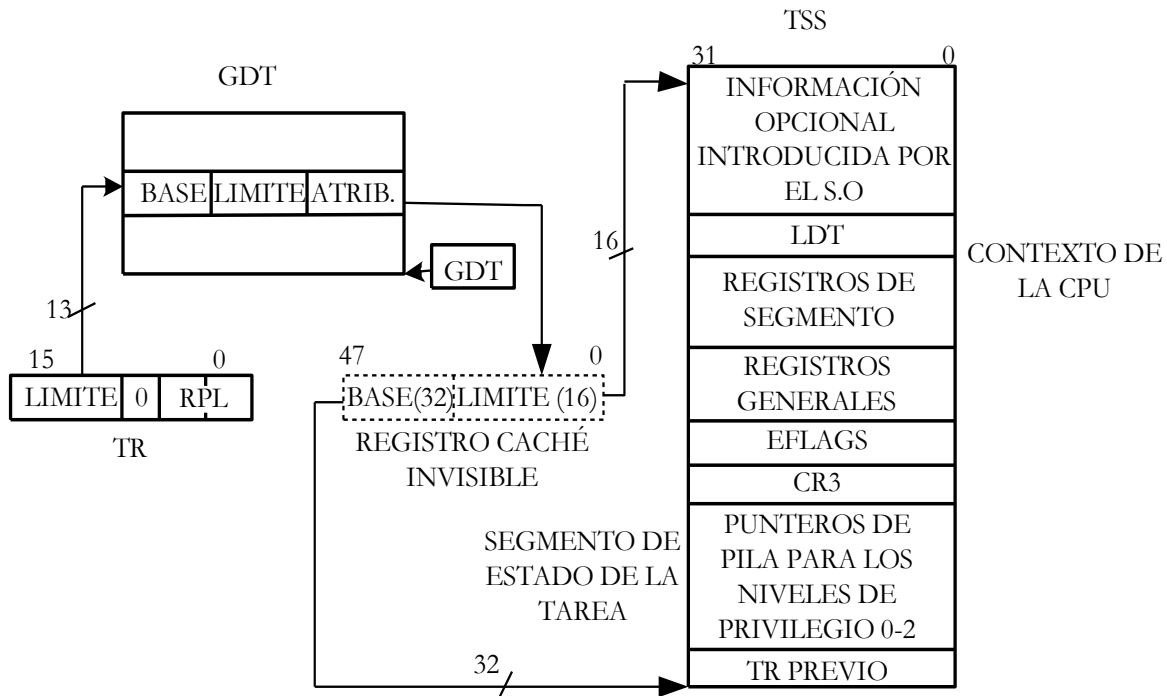


Fig. 11.7. El registro TR actúa como un selector de un descriptor de segmento de la GDT, que apunta al segmento TSS, que es el Segmento de Estado de la Tarea. Tiene igual comportamiento que el registro LDTR.

11.5- REGISTRO DE SEÑALIZADORES (EFLAGS)

El registro EFLAGS, como ya se ha visto anteriormente, se compone de 32 bits, de los cuales 21 son significativos. No obstante, los que a continuación se explicaran son específicos para el programador de sistemas, siendo el resto usados por el programador de aplicaciones:

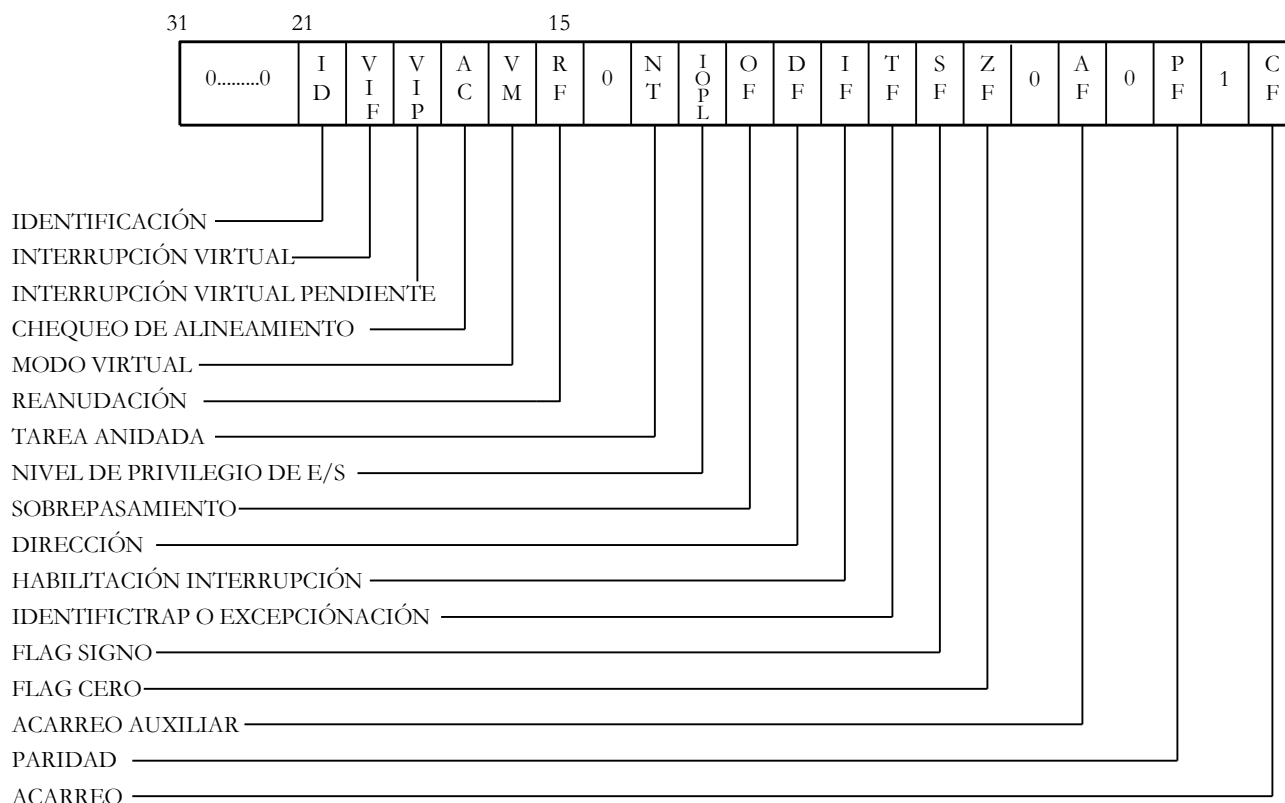


Figura 11.8. Distribución de los bits señalizadores del registro EFLAGS.

- **VM: Modo virtual-86:** Sirve para pasar desde el Modo Protegido al Modo Virtual 86, para lo cual basta con poner a 1 este bit, dentro del Modo Protegido. En este modo de funcionamiento, se pueden ejecutar programas del 8086 en un entorno protegido, esto es, con sistemas de protección, multitarea...
- **RF: Flag de reanudación del punto de parada:** El Pentium dispone de un conjunto de registros de depuración para programar puntos de parada. Poniendo RF=1, se permite ignorar los puntos de parada.
- **IOPL: Nivel de privilegio de las E/S:** Tiene 2 bits. Contiene el menor nivel de privilegio que deben tener los segmentos de código, en Modo Protegido, para poder ejecutar las instrucciones protegidas relacionadas con las E/S.

- **NT: Tarea anidada:** La tarea en fase de ejecución ha sido llamada por la otra a la que hay que retornar en el caso de que el bit NT=1. Controla el encadenamiento de las interrupciones y las llamadas de las tareas. El procesador pone este flag a 1 cuando se hace una llamada a una tarea mediante una instrucción CALL, una interrupción o una excepción, y examina y modifica este flag cuando se retorna de una tarea mediante la instrucción IRET. De cualquier modo, cambiar el valor de este flag puede generar excepciones inesperadas en las aplicaciones.
- **AC: Chequeo de alineamiento:** Si AC=1 y AM=1 en el registro de control CR0 la CPU genera una excepción cuando localiza un operando cuya dirección no sea múltiplo de 4. Normalmente las excepciones provocadas por el chequeo de alineamiento se producen solo en el modo usuario (PL=3). Las excepciones por chequeo de alineamiento pueden ser usadas para comprobar el alineamiento de los datos. Esto es útil cuando se intercambia información con otros procesadores, lo que requiere que los datos estén alineados.
- **VIF: Flag de interrupción en modo virtual:** Es un bit equivalente al bit IF del modo protegido, que se utiliza cuando trabajamos en modo virtual-86, es decir, es la imagen virtual del flag IF. El procesador solo reconoce este flag cuando el bit VME o el bit PVI en el registro CR4 están activados (1) y el nivel de privilegio (PL) es inferior a 3.
- **VIP: Flag de interrupción pendiente en modo virtual:** Este bit nos indica que una interrupción esta pendiente cuando esta a 1, por el contrario no esta pendiente cuando es 0. El software activa y desactiva este flag, mientras que el procesador solamente lo lee, nunca lo modifica. Al igual que pasa en el flag anteriormente visto, el procesador solo reconoce este flag cuando el bit VME o el bit PVI en el registro CR4 están activados (1) y el nivel de privilegio (PL) es inferior a 3.
- **ID: Identificación del procesador:** Si se activa este flag se habilita el uso de la instrucción CPUID, la cual identifica el procesador devolviendo muchas de las características del mismo.

11.6- REGISTROS DE CONTROL

Son 5 registros de 32 bits cada uno, llamados CR0-CR4, de los cuales el CR1 no esta definido en el funcionamiento del Pentium y su uso está previsto desarrollar en nuevos procesadores.

CR0, CR2, CR3 y CR4 son accesibles al programador de sistemas y pueden ser leídos y escritos mediante instrucciones del tipo MOV desde los registros de propósito general. Estos registros determinan el modo de operación del procesador y las características de la tarea en ejecución.

11.6.1- CR0: Doble palabra de estado de la máquina

Como puede apreciarse en la figura 11.6, este registro solo tiene asignadas funciones a once de sus bits.

Los bits definidos en CR0 son:

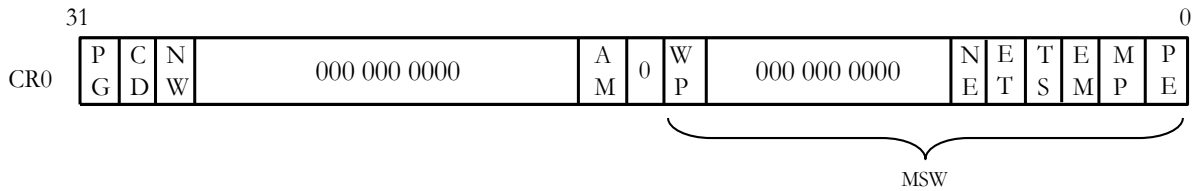


Figura 11.9. Estructura del registro CR0, cuyos 16 bits de menos peso forman la “palabra de estado de la máquina” (MSW).

- **PG: Paginación:** Cuando este bit vale 1 significa que está habilitada la Unidad de Paginación, que traduce las direcciones lineales a físicas. Cuando la paginación esta desactivada, todas las direcciones lineales son tratadas como direcciones físicas. El flag PG no tiene efecto si el flag PE (bit 0 en CR0) no esta tambien activo. De hecho, la activación de PG estando PE inactivo provocaría que se generase una excepción de protección general.
- **CD: Caché desactivada:** Cuando CD=0 activa la caché interna. Si CD=1 la desactiva.
- **NW: No escritura obligada:** Si NW=0 pone en marcha la escritura obligada y los ciclos de invalidación de la caché. Si NW=1 los desactiva.
- **ET: Tarea de coprocesador:** En el Pentium no se utiliza. Se usaba en los procesadores X86, 286 y 386 para indicar si el coprocesador externo era el 287 ó el 387. Cuando valía 1 el coprocesador era el 387.
- **TS: Tarea conmutada:** La CPU pone a 1 este bit al producirse una conmutación de tarea. Luego, durante la ejecución de instrucciones, lo examina cuando encuentra alguna instrucción para el coprocesador, en cuyo caso salva el contexto de la anterior tarea que tenía el coprocesador y carga el nuevo poniendo TS=0.
- **EM: Emulación:** Cuando EM=1 la ejecución de una instrucción numérica genera una excepción, por lo que debe tener dicho valor en caso de que el procesador no disponga de FPU (Unidad de Coma Flotante). Es decir, este bit se usa para indicar si se utiliza el emulador del coprocesador. Poner este bit a 1 fuerza a todas las instrucciones en coma flotante a ser ejecutadas por emulación software.
- **PE: Habilidad del modo protegido:** Cuando PE=1, el Pentium opera en Modo Protegido, mientras que si PE=0, lo hace en Modo Real.
- **AM: Mascara de alineamiento:** Cuando AM=1 y el bit AC(EFLAGS)=1 si el nivel de privilegio (PL)=3, estando en modo protegido o modo virtual-86, comprueba si la dirección de los operandos es múltiplo de 4 y si no lo es genera una excepción.
- **WP: Protección de escritura:** Si WP=1 impide escribir una pagina de nivel de usuario definida de solo lectura o ejecución, incluso desde el modo supervisor.
- **NE: Error del coprocesador:** Cuando NE=1 se activa el mecanismo normal de devolver errores en la x87 FPU. Si NE=0 y IGNE#=0 se ignoran los errores numéricos. Si NE=0 y IGNE#=1 un error numérico causa la parada del procesador y espera una interrupción. Dicha interrupción llega por la patita FERR# que provoca una entrada al controlador de interrupciones y detiene inmediatamente la ejecución de instrucciones antes de ejecutar la siguiente instrucción en coma flotante o las instrucciones WAIT/FWAIT.
- **MP: Monitor de coprocesador:** En el Pentium MP=1. Se usaba en procesadores anteriores para controlar la función de la instrucción WAIT, que se usa para sincronizar el procesador con el coprocesador.

11.6.2- CR2: Dirección lineal de fallo de página

En este registro se almacena la dirección lineal que se introdujo en la Unidad de Paginación para traducirla a dirección física, y que ocasionó un error o fallo de página.

El código de error que especifica la causa de fallo se almacena en la pila del manipulador de los fallos de página.



Figura 11.10. Estructura del registro CR2.

11.6.3- CR3: Base del directorio de las tablas de páginas

Guarda la dirección física en la que comienza el Directorio de las Tablas de Páginas de la tarea en curso. Como dicho Directorio tiene formato de una página de 4 KB, los 12 bits de menos peso de CR3 (como deberían ser todo 0) se ignoran en la escritura y solo dos tienen significado en el Pentium: el PCD y el PWT. Se le conoce como Registro Base del Directorio de Páginas (PDBR).



Figura 11.11. Estructura del registro CR3.

El bit PCD tiene el valor de la patita del Pentium PCD en los ciclos que no hay paginación. Se usa para controlar la caché. Para que un acceso a memoria pueda ser llenado con una línea de la caché, la patita KEN# debe estar a cero. Es decir, para permitirlo PCE y KEN# deben valer cero. El bit PWT toma el valor de la patita del Pentium del mismo nombre en los ciclos que no hay paginación.

Cuando hay paginación, estos dos flags indican lo siguiente:

- PWT: Pagina de escritura obligada.
- PCD: Pagina cacheable.

Indican lo mismo que los flags del mismo nombre en las entradas de las tablas de páginas.

Tanto la conmutación de tareas, como la carga de CR3, invalidan todas las entradas de la TLB.

11.6.4- CR4: Extensiones de la arquitectura

Es un nuevo registro que se incorporó en el Pentium y que contiene varios bits que soportan diversas extensiones de la arquitectura.

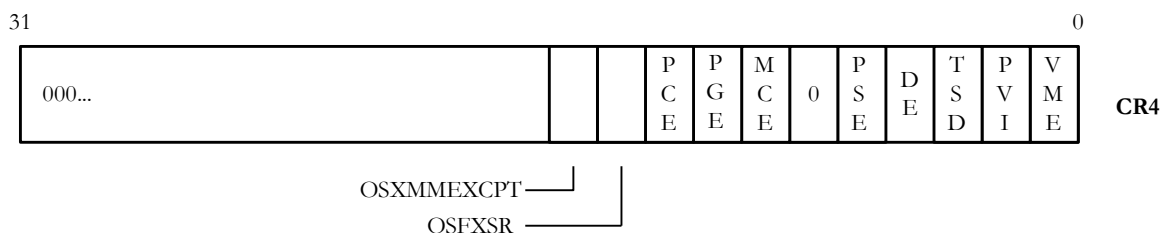


Figura 11.12. Distribución de los bits mas significativos del registro CR4.

- **VME: Extensión del modo virtual 8086:** Si VME=1 se activa el soporte para que funcione el flag VIF para las interrupciones en Modo Virtual.
- **PVI: Interrupciones virtuales en modo protegido:** Si PVI=1 se activa el soporte para que funcione el flag VIF para las interrupciones en Modo Protegido. Así algunos programas diseñados para ejecutarse con PL=0, se pueden ejecutar con PL=3.
- **TSD: Desactivación del “time-stamp”:** Si TSD=1 la instrucción RDTSC pasa a ser de tipo privilegiada. Esto es, no se puede ejecutar para los niveles de prioridad distintos a 0 (PL != 0).
- **DE: Depuración de extensiones:** Con DE=1 se activan los puntos de ruptura en el espacio de E/S.
- **PSE: Extensión del tamaño de página:** Si PSE=1, se activan páginas de 4 MB. Por el contrario si PSE=0 las páginas son de 4 KB.
- **MCE: Activación del chequeo de la máquina:** Con MCE=1 se activa la excepción correspondiente al chequeo de la máquina.
- **PGE:** Cuando este bit esta a 0, las paginas globales están deshabilitadas, mientras que si esta a 1 están habilitadas.
- **PCE:** Este bit permite, cuando esta activado (PCE=1), ejecutar la instrucción RDPMC. Sin embargo, esta instrucción solo se podrá ejecutar desde el nivel de prioridad 0 (RPL=0) cuando este bit valga 0.

11.7- REGISTROS DE DEPURACION

Forman un conjunto de ocho registros de 32 bits, dos de los cuales (DR4-DR5) son de uso reservado por el fabricante. Son accesibles al programador y constituyen una potente herramienta para la puesta a punto del sistema lógico, puesto que permiten especificar hasta cuatro puntos de parada o ruptura, que se controlan de forma muy flexible. (Figura 11.9).

Cuando se produce un punto de parada o se da una condición especial en la depuración, el procesador genera un tipo concreto de interrupción cuya rutina de servicio se encarga de atender el origen que la ocasionó.

El señalizador de reanudación, RF, si esta a 1, permite volver a ejecutar una instrucción después de producirse una excepción de depurado, de manera que no se reproduzca la misma excepción a causa de las mismas condiciones.

Los registros DR0-DR3 contienen las direcciones asociadas a una de las cuatro condiciones de punto de ruptura, definidas por ciertos bits en el registro DR7. Estos cuatro registros contienen direcciones lineales, que son direcciones físicas cuando se halla inhabilitada la paginación, o bien hay que traducirlas a físicas si está habilitada. En este último caso, en DR7 residen los bits de habilitación local y global, que determinan que direcciones de los registros DR0-DR3 son relevantes para cada tarea.

A DR7 se le llama registro de control, pues define y habilita o no, selectivamente, las condiciones de depuración. Cada registro de direcciones de depuración, DR0-DR3, tiene en DR7 sus propios campos de dos bits para la habilitación de puntos de ruptura de Lectura/Escritura (RWx), Longitud (LENx) y Local/Global (Lx y Gx).

En DR7, los bits 16:17, 20:21 y 24:25 contienen los campos R/W. Estos campos R/W se interpretan de la siguiente forma:

- 00: Ruptura en ejecución de una instrucción.
- 01: Ruptura en escritura de datos.
- 10: Operaciones L/E (DE debe ser 1).
- 11: Ruptura en escritura o lectura de datos, pero no en la búsqueda de instrucciones.



Figura 11.13. Formato de los seis registros de depuración que tienen definidas funciones en el Pentium.

Cada registro de direcciones DR0-DR3, tiene asociado en DR7 un campo LENx que especifica la longitud de los datos que se están vigilando, para esto los datos deben estar alineados. Estos campos LEN se encuentran en los bits 18:19, 22:23, 26:27, 30:31 de DR7. LENx se interpreta:

- 00: Longitud de un byte.
- 01: Longitud de dos bytes.
- 10: No utilizado.
- 11: Longitud de cuatro bytes.

Cuando el campo R/W n esta a cero, el campo Ln también debe estar a cero.

Cada registro de direcciones también tiene asociados dos campos, llamados Lx y Gx. Los Lx, que se encuentran en los bits 0, 2, 4 y 6 de DR7, representan las habilitaciones locales de las condiciones x de los puntos de ruptura. Los Gx, sin embargo, se encuentran en los bits 1, 3, 5 y 7 de DR7, y representan las habilitaciones globales de las condiciones x de los puntos de ruptura. En los dos casos anteriores, un 1 habilita local o globalmente la condición x del punto de ruptura al que este asociado (x), mientras que un 0 la deshabilita. La diferencia entre ellos es que Ln habilita su puesta a cero (desactivación) por el microprocesador en cada conmutación de tarea para evitar condiciones indeseables de puntos de ruptura en la nueva tarea.

El registro DR7, además de todos los anteriores, tiene tres bits llamados LE, GE y GD, los cuales se encuentran en los bits 8, 9 y 13, respectivamente. Los dos primeros se ignoran en los procesadores superiores al 486. Sin embargo, GD sí tiene un significado en el Pentium: si GD vale 0 se deshabilita la protección de los registros de depuración, mientras que si vale 1 se habilita esta protección. Este bit es puesto a 0 cuando se entra al manipulador de excepciones.

DR6 contiene varios señalizadores de condición de depuración, que permiten determinar al depurador, las condiciones que se han producido con el error. Cuando se detecta una excepción de depuración habilitada, el bit asociado Bn se pone a 1. El campo BT de DR6 funciona con un bit de excepción en depuración que se guarda en la primera posición del TSS. BT se pone a 1 antes de introducir al manipulador de depuraciones, si se ha producido una excepción de depuración por una conmutación de tareas y el bit de excepción del TSS está a 1. El bit señalizador BS funciona con el bit del señalizador de excepción TF, del registro EFLAGS. BS se pone a 1 cuando el manipulador de depuraciones actúa como resultado de una ejecución paso a paso. El señalizador BD indica si la siguiente instrucción leerá o escribirá uno de los ocho registros de depuración.

El microprocesador nunca pone a cero los señalizadores de DR6. Los ceros se desplazan a DR6 antes de intentar identificar la siguiente excepción de depuración.

Los registros de depuración mejoran considerablemente las características y el control sobre puntos de ruptura propios de los microprocesadores convencionales. Así, por ejemplo, añaden la posibilidad de forzar puntos de ruptura sobre acceso a datos, de forma que si una variable se intenta rescribir accidentalmente, se puede activar un punto de ruptura que detenga la ejecución, siempre que se vaya a modificar el contenido de la variable.

Otras herramientas de las que dispone el Pentium, complementarias a los registros de depuración son:

1. **Puntos de ruptura por software:** Usando la instrucción de un byte INT3, se provoca una excepción al ser ejecutada, la cual es tratada mediante el recurso que halla para su atención en la IDT.
2. **Paso a paso:** Cuando el señalizador TF=1, se produce una excepción cada vez que se ejecuta una instrucción. La rutina que atiende esta excepción puede usarse para visualizar el estado de la CPU, o bien, examinar alguna característica del procesador.

11.8- REGISTROS DE PRUEBA DE LA TLB

Son dos registros de 32 bits, TR6 y TR7, con los que se puede leer y escribir el contenido de una entrada de la TLB, que es una pequeña memoria caché ultrarápida que contiene la traducción de dirección lineal a dirección física de las 32 páginas que se han usado últimamente.

11.9- REGISTROS ESPECÍFICOS

El Pentium incorporó algunos registros específicos como el registro de chequeo de dirección de la máquina, el registro de chequeo del tipo de máquina, los registros de chequeo del TR1 al TR12, el contador de “time-stamp”, el registro de selección de control/eventos y los dos contadores 0 y 1.

Paralelamente se incorporaron al repertorio del Pentium las instrucciones que permitían leer (RDMSR) y escribir (WRMSR) los registros específicos.

- **RDMSR:** (Read Model-Specific Register): El valor en ECX especifica uno de los registros de 64 bits específicos del modelo del procesador. El contenido de ese registro se carga en EDX:EAX. EDX se carga con los 32 bits más significativos, mientras que EAX se carga con los 32 bits menos significativos.
- **WRMSR:** (Write Model-Specific Register): El valor en ECX especifica uno de los registros de 64 bits específicos del modelo del procesador. El contenido de EDX:EAX se carga en ese registro. EDX debe contener los 32 bits más significativos, mientras que EAX debe contener los 32 bits menos significativos.

Para referenciar a estos registros se les asigna un número. Así el 00 es para el registro de chequeo de direcciones de máquina y se usa para conocer las direcciones físicas que provocan un ciclo de bus erróneo. El número 01 es para el registro de chequeo del tipo de máquina y se usa para conocer el tipo de ciclo de bus erróneo. El número 0E Hex es para TR12 y actúa como un registro de control para salto en ciclos especiales. Finalmente el número 10 Hex se asigna al contador “time-stamp” que se usa para leer o escribir el contador interno de 64 bits.

La instrucción RDMSR sólo es ejecutable en Modo Protegido con nivel de privilegio 0.