

INGENIERÍA INFORMÁTICA

LABORATORIO DE COMPUTADORAS

ARQUITECTURA X86

TEMA: ARQUITECTURA
80286

Novedades

Las principales novedades de 80286 respecto al 8086 son:

1. Memoria Virtual.
2. Multitarea.
3. Asignación de diferentes niveles de privilegio en los programas y en los segmentos.

El 80286 puede emplear o no estas innovaciones.

Dispone de dos posibles modos de funcionamiento:

a. Modo Real de Direcciones.

Maneja solo direcciones reales como el 8086.

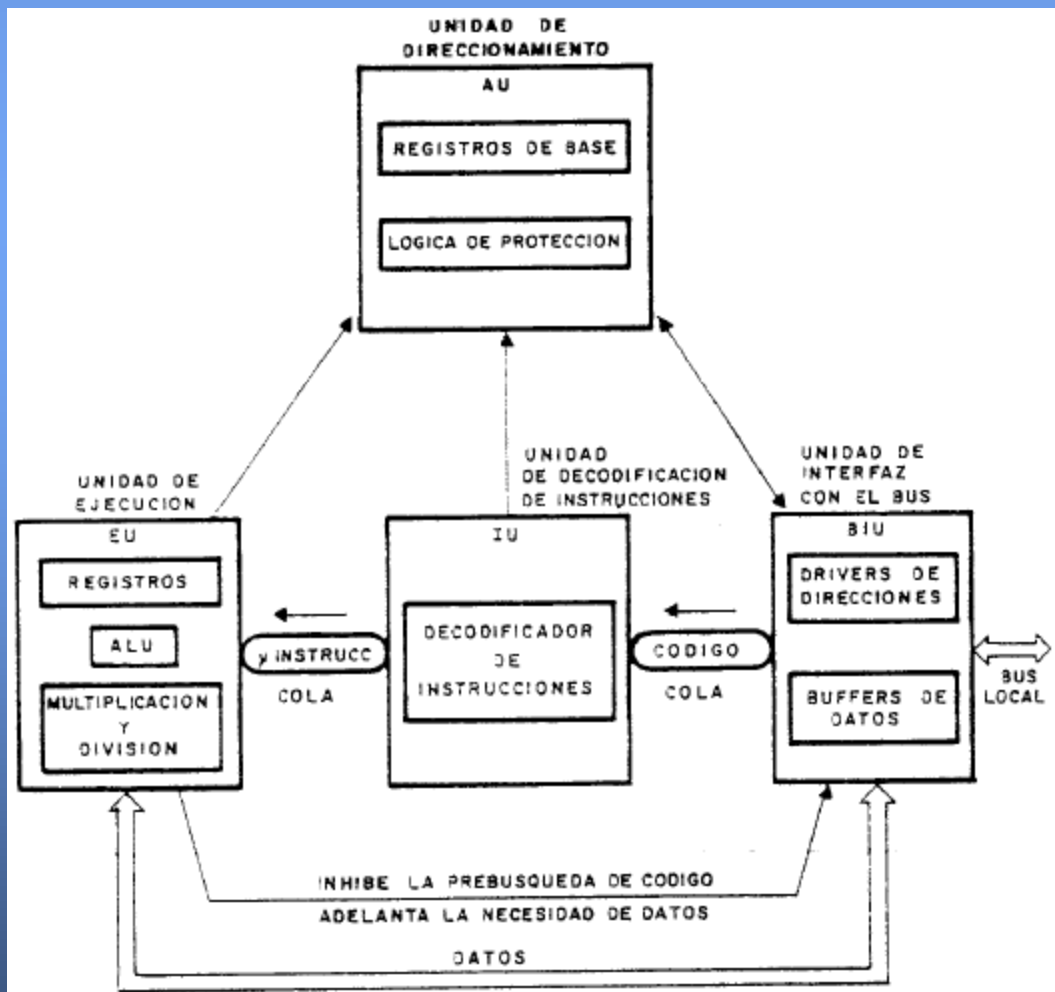
b. Modo de Direccionamiento Virtual Protegido.

Utiliza toda su potencia y características: memoria virtual, mecanismos de protección y nuevas instrucciones.

Estructura Interna

Consta de 4 bloques fundamentales:

1. Unidad de Decodificación de Instrucciones.
2. Unidad de Ejecución.
3. Unidad de Direccionamiento.
4. Unidad de Interfaz con el Bus.



Ejecución de Instrucciones

La constitución interna del 80286 descompone a cada instrucción en varias operaciones elementales:

- a. Búsqueda de la instrucción (código OP).
- b. Decodificación.
- c. Generación de la dirección del operando.
- d. Búsqueda del operando.
- e. Ejecución de la instrucción.
- f. Almacenamiento del resultado.

Modo Real

El 286 trabaja como un 8086 pero en modo máximo. Cada vez que se inicializa el micro mediante RESET comienza funcionando en modo Real.

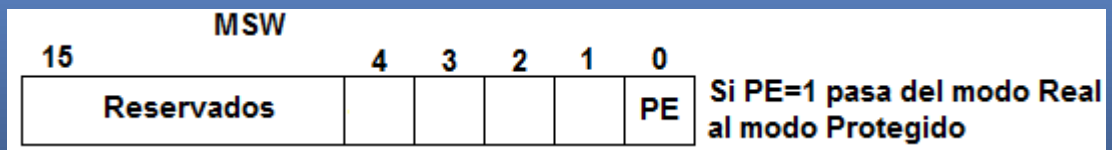
La única particularidad del modo Real es que el 80286 puede usar los nuevos registros que dispone, y las nuevas instrucciones para manipularlos.

Los nuevos registros en modo Real, sirven para pasar al modo Protegido.

Los registros son:

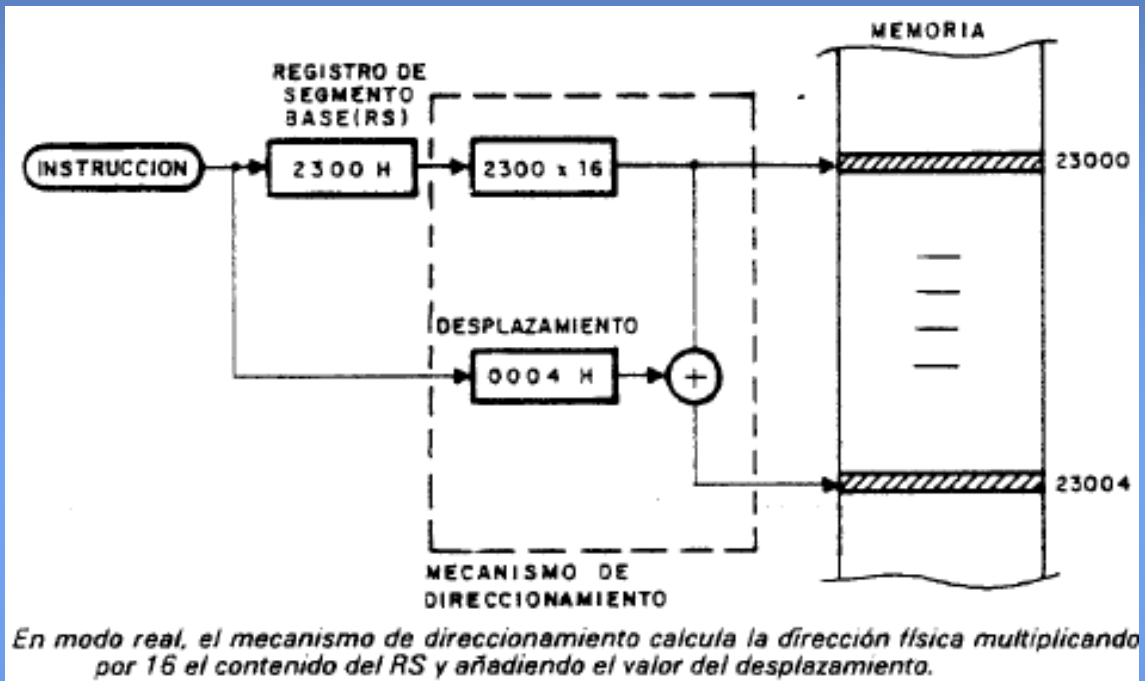
- MSW: Palabra de Estado de la Máquina.
- GDTR: Registro de la Tabla Global de Descriptores.
- IDTR: Registro de la Tabla de Descriptores de Interrupción.

Ejemplo:



Modo Real

El 80286 trabaja, en modo real, con direcciones físicas de la misma forma que el 8086.



Modo Protegido

Se entra en modo Protegido desde el Modo Real, poniendo a 1 el bit PE de la MSW. El mecanismo de direccionamiento se transforma radicalmente y el mapa de memoria virtual direccionable se amplía desde $2^{24}=16$ Mb hasta $2^{30}=1$ Gb.

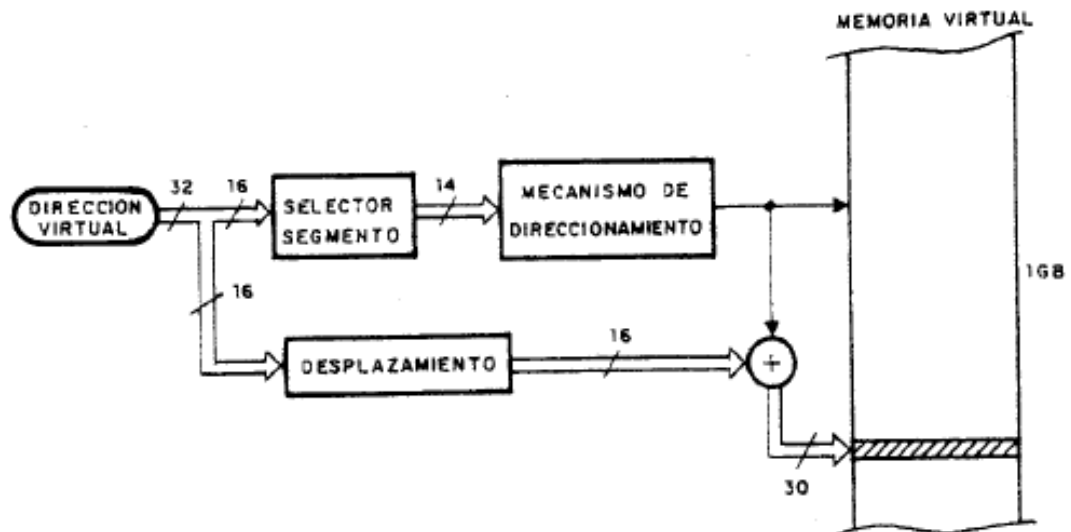
En memoria virtual hay que distinguir entre dirección virtual y dirección física. El programador dispone de una gran memoria, que alcanza 1 Gb, aunque el procesador solo controla, directamente, una memoria real (físicamente implementada) más pequeña (16 Mb).

Dirección virtual \rightarrow 30 bits Dirección física \rightarrow 24 bits

El mecanismo de direccionamiento es la “Unidad de Manejo de la Memoria (MMU)”. Transforma una dirección lógica en una dirección física.

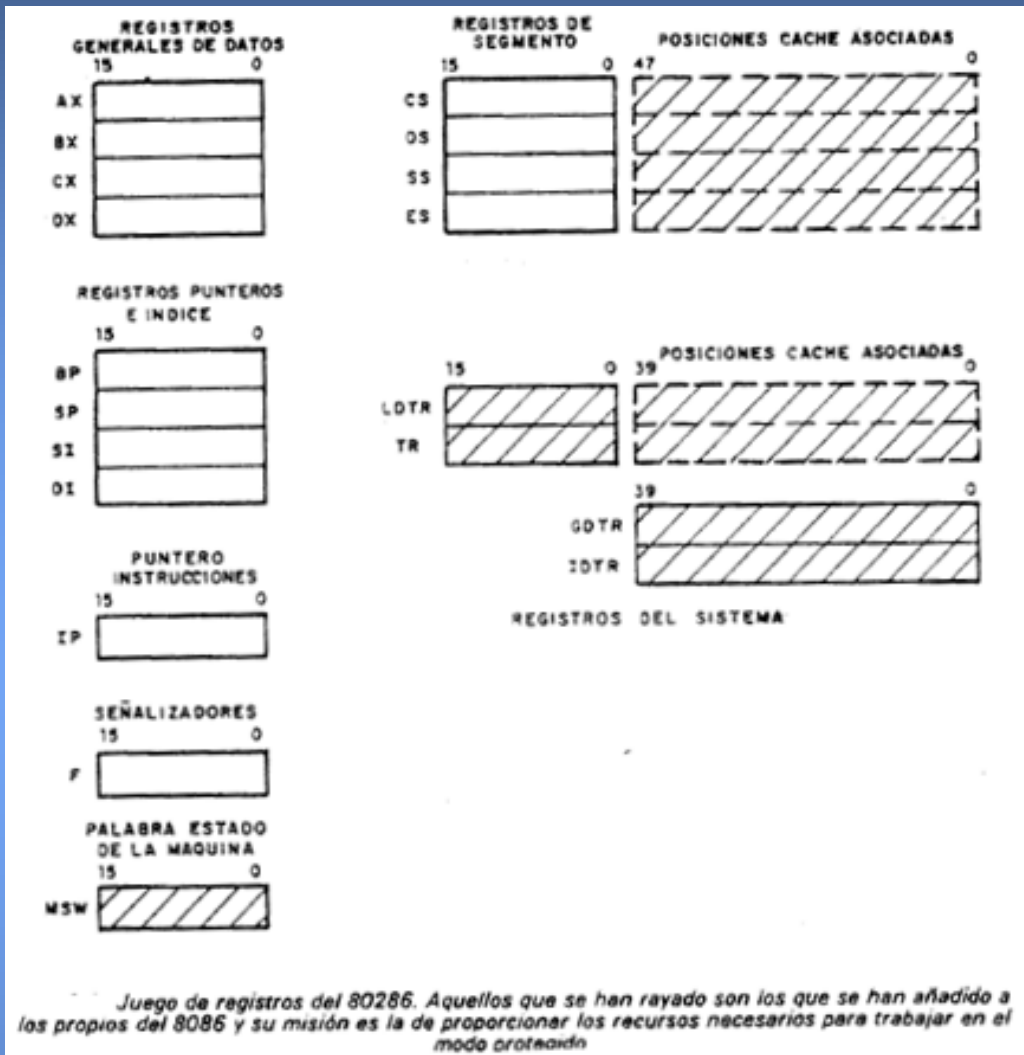
Los contenidos de la memoria virtual suelen situarse sobre una memoria de masa, por ej. disco, y solo una pequeña parte de la misma está depositada en la memoria principal (RAM), directamente controlada por el bus de direcciones de la CPU. También hay que determinar si dicha posición está cargada en la memoria principal, para que en caso contrario se produzca una “excepción” de fallo en la memoria y se ejecute una rutina que se encargue de transferir el bloque donde se encuentra dicha posición desde el disco

MMU



En modo protegido, la dirección virtual consta de 32 bits, que se descompone en dos campos: selector de segmento (14 bits) y desplazamiento (16 bits). Los dos bits restantes no intervienen en el direccionamiento de la memoria virtual.

Juego de Registros

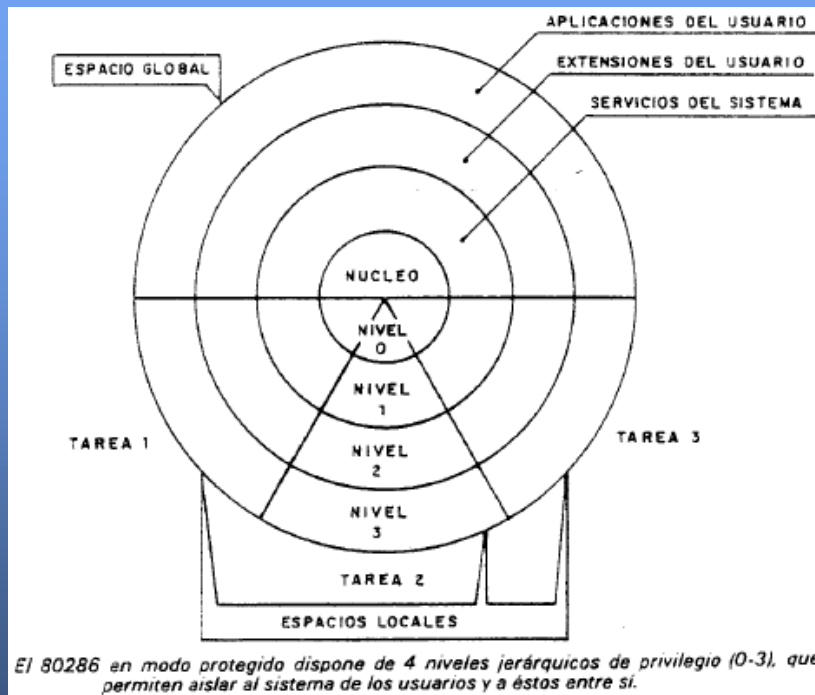


Los registros del sistema: GDTR, IDTR, LDTR y TR se emplean en el manejo de la memoria virtual y el soporte de la multitarea.

Niveles de Privilegio

Cuando se entra en el modo protegido, el 80286 proporciona, al sistema lógico, cuatro niveles jerárquicos de protección, que se designan del 0 al 3. El nivel más privilegiado es el 0 y suele dedicarse al núcleo del sistema operativo; el menos privilegiado es el 3 y lo emplean las aplicaciones del usuario.

Mediante la correcta aplicación de los niveles de privilegio se puede aislar al sistema respecto a las aplicaciones del usuario en ambientes multitarea, a los usuarios entre sí y se puede comprobar, en todo momento, el tipo de información que se maneja.



Direccionamiento en Modo Protegido

Tanto en modo real como en modo protegido, la memoria que maneja el 80286 es del tipo segmentado. Esto significa que cada tarea agrupa, en segmentos de memoria, la información del mismo tipo.

Así, existen 1 o varios segmentos de código, de datos y de pila.

Un segmento es un trozo de memoria de tamaño variable que contiene el mismo tipo de información.

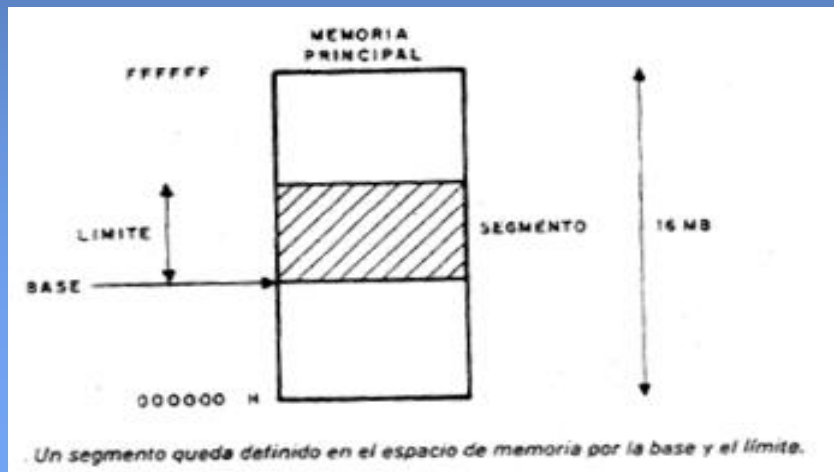
En el 80286, un segmento de memoria alcanza la longitud máxima de 64 kb, igual que el 8086.

Un segmento queda perfectamente definido mediante tres informaciones básicas de él:

- 1) Dirección real de la base del segmento (24 bits).
- 2) Límite o longitud máxima en bytes (16 bits).
- 3) Derechos de acceso (8 bits): definen el tipo de segmento (código, datos, etc.) y su accesibilidad (solo lectura, lectura-escritura, etc.).

Descriptores

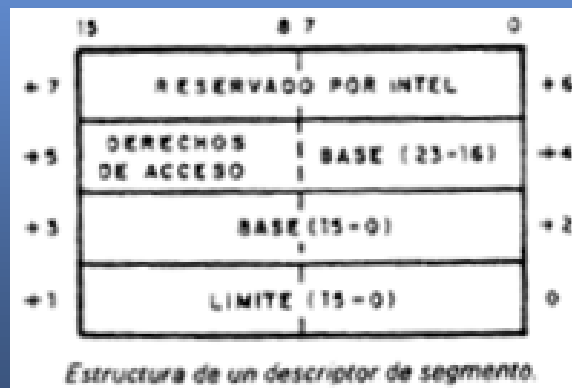
Recibe el nombre de descriptor de segmento una estructura de datos formada por un conjunto de 8 bytes que describen a un segmento en particular.



Contiene:

- La dirección real de la base.
- El límite.
- Los derechos de acceso.

Estructura de un descriptor de segmento



Tablas de Descriptores

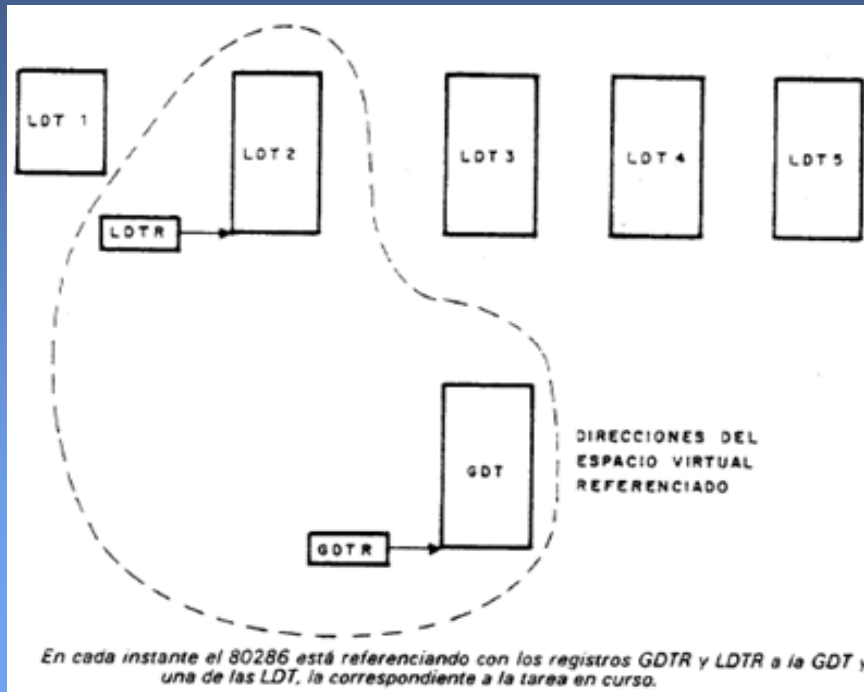
Dentro de la memoria principal existen unas tablas que contienen a los descriptores de segmento. Los descriptores que son usados por todas las tareas están depositadas en la tabla denominada GLOBAL o GDT (Tabla Global de Descriptores).

Las restantes tablas guardan los descriptores de cada tarea, es decir, para cada tarea hay una tabla de descriptores propios de la misma.

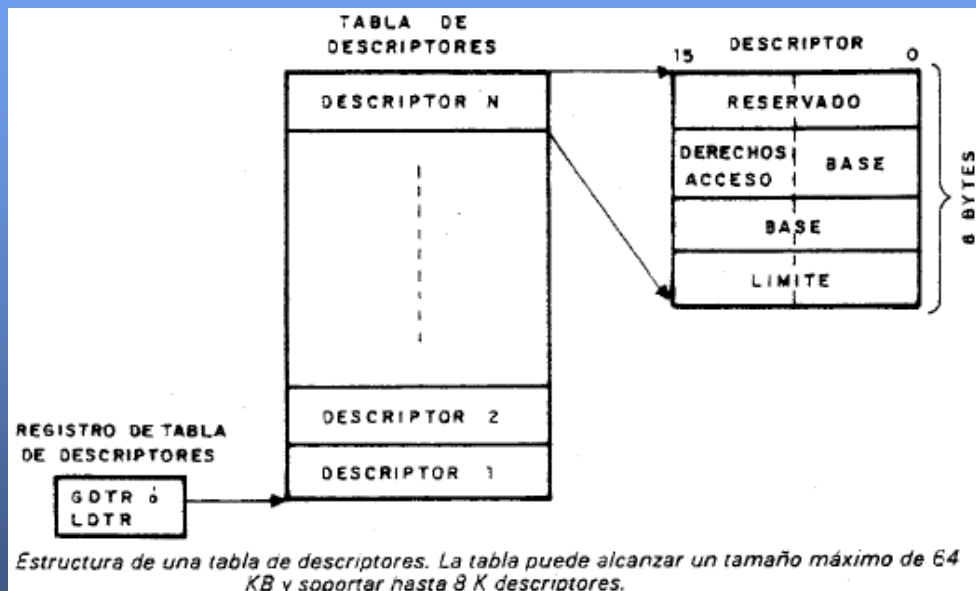
Este último tipo de tabla se llama LDT (Tabla Local de Descriptores) y existen tantas como tareas soporte el sistema.

Cuando el 286 trabaja en modo protegido siempre está haciendo referencia a la GDT y a una de las LDT, y sólo puede acceder a los segmentos referidos por los descriptores de dichas tablas, Dos registros internos de la CPU se encargan de especificar las tablas GDT y LDT, son el GDTR (Registro de la Tabla Global de Descriptores) y el LDTR (Registro de la Tabla Local de Descriptores).

Tabla de Descriptores



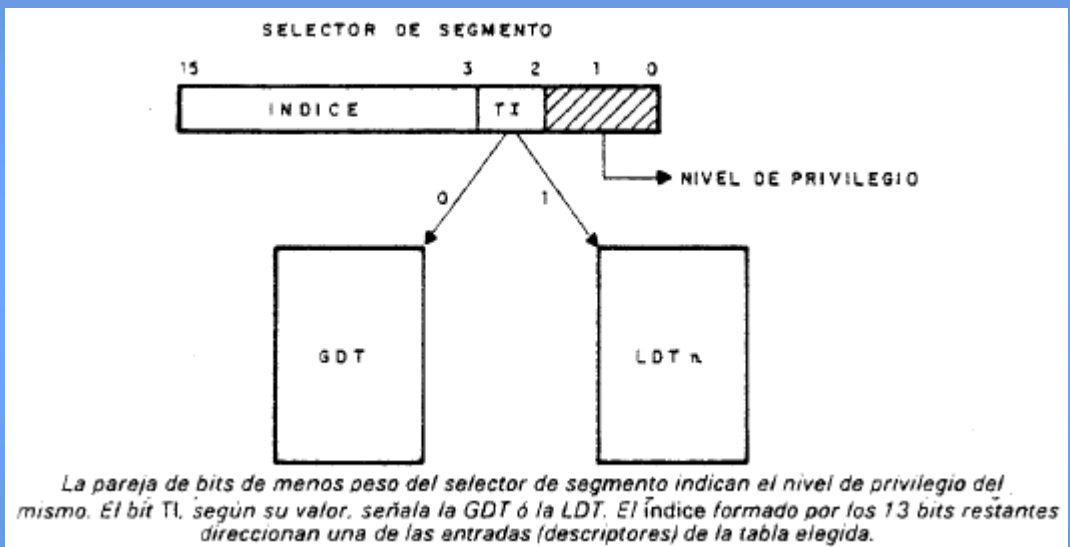
Estructura interna de una Tabla de Descriptores



Mecanismo de Direccionamiento

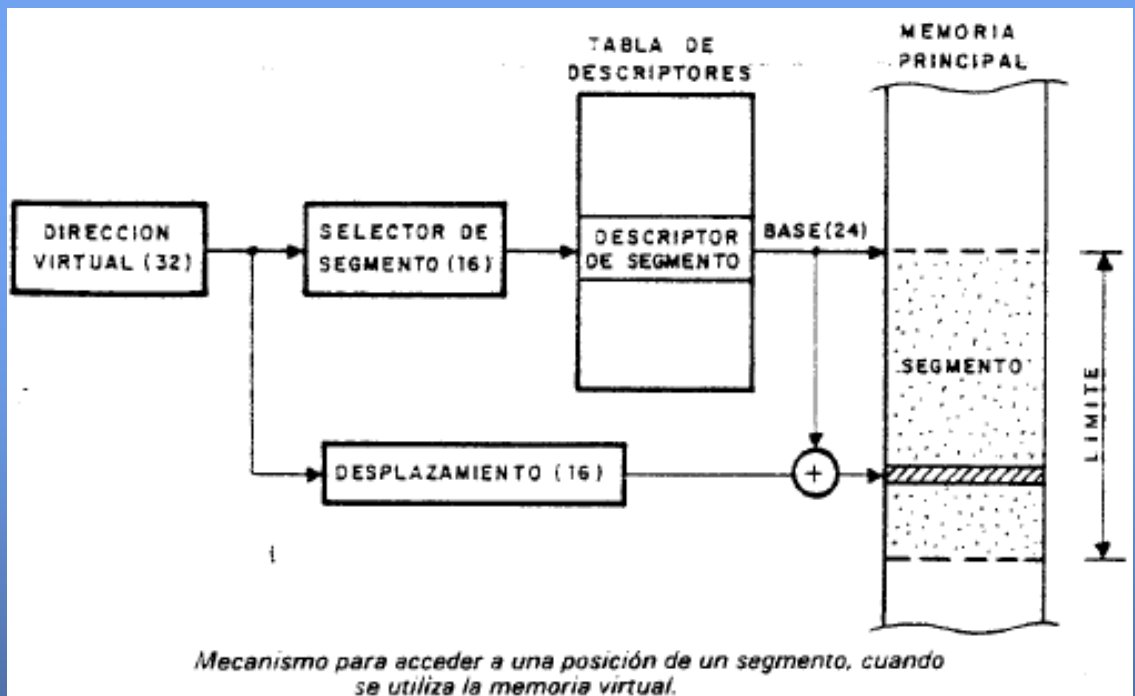
Una dirección virtual está formada por dos campos de 16 bits cada uno:

Selector de Segmento : Desplazamiento

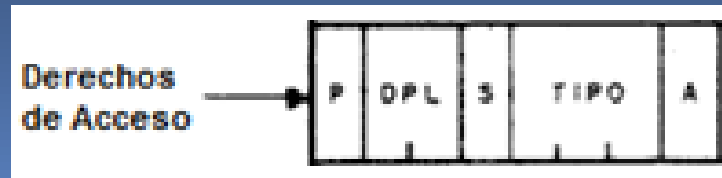


Mecanismo de Direccionamiento

Una vez seleccionado un descriptor de segmento mediante el selector, se usa el otro campo de la dirección virtual, que también consta de 16 bits, para determinar dentro del segmento referenciado por el descriptor, una posición concreta.



Derechos de Acceso



P (Presencia): este bit es manejado por el sistema operativo cuando trabaja en modo virtual para saber si el segmento a que referencia el descriptor está contenido en la memoria principal. Si se halla en la memoria principal es $P=1$, caso contrario $P=0$.

DPL (Nivel de Privilegio): el campo DPL consta de 2 bits y contiene el nivel de privilegio del descriptor. Se usa para determinar la accesibilidad a ese segmento desde otras partes del sistema lógico.

S (Descriptor de Segmento): si $S=1$ el descriptor hace referencia a un segmento de código o de datos. Si $S=0$, el descriptor hace referencia a otro elemento distinto de un segmento normal.

Tipo (Tipo del Segmento): dentro del byte de derechos de acceso, el campo Tipo consta de tres bits.

Derechos de Acceso

Con este campo, el procesador distingue la clase de segmento al que se refiere el descriptor. El 80286 distingue entre dos grandes clases de segmentos: de datos y de código.

Si es un segmento de datos, los bits del campo Tipo puntualizan sus características relevantes:



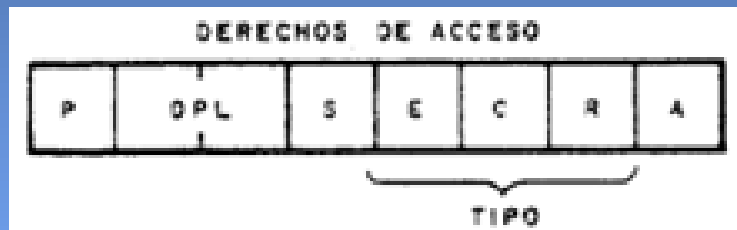
E (Ejecutable): indica si la información que contiene el segmento es ejecutable o no. En un segmento de datos $E = 0$.

ED (Dirección de Expansión): cuando $ED = 0$ se trata de un segmento de datos con direccionamiento normal, es decir, el crecimiento de la dirección es ascendente. Si $ED = 1$ se trata de un segmento de pila, con crecimiento descendente de las direcciones.

W (Escritura): si $W = 0$ se trata de un segmento que sólo se puede leer, mientras que si $W = 1$, se puede leer y escribir.

Derechos de Acceso

Si es un segmento de código, los bits del campo Tipo puntualizan sus características relevantes:



E (Ejecutable): indica si la información que contiene el segmento es ejecutable o no. En un segmento de código $E = 1$.

C (Ajustable): cuando $C = 0$ el segmento sigue las reglas generales existentes sobre los privilegios para ser accedido. Si $C = 1$ el acceso al segmento se regula por normas especiales (segmento “ajustable”).

R (Lectura): si $R = 0$ el segmento no es accesible a la lectura. Si $R = 1$ el acceso está permitido a la búsqueda del código OP y a la lectura en memoria.

A (Accedido): este bit se pone a 1 cada vez que se accede a dicho segmento. Lo usa el S.O. para aplicar los algoritmos de intercambio de segmentos entre la memoria virtual y la principal.

Registros de Segmento

Los registros de segmento (RS) tienen, en modo protegido, un comportamiento completamente diferente que en modo real; contienen el campo selector de segmento de la dirección virtual. En este modo (virtual) a cada RS la CPU le asocia un registro caché oculto de 48 bits, caracterizado por su alta velocidad de acceso. Una vez que se selecciona un descriptor con el contenido de un RS, la CPU carga automáticamente los tres parámetros que contiene el descriptor en el registro caché.

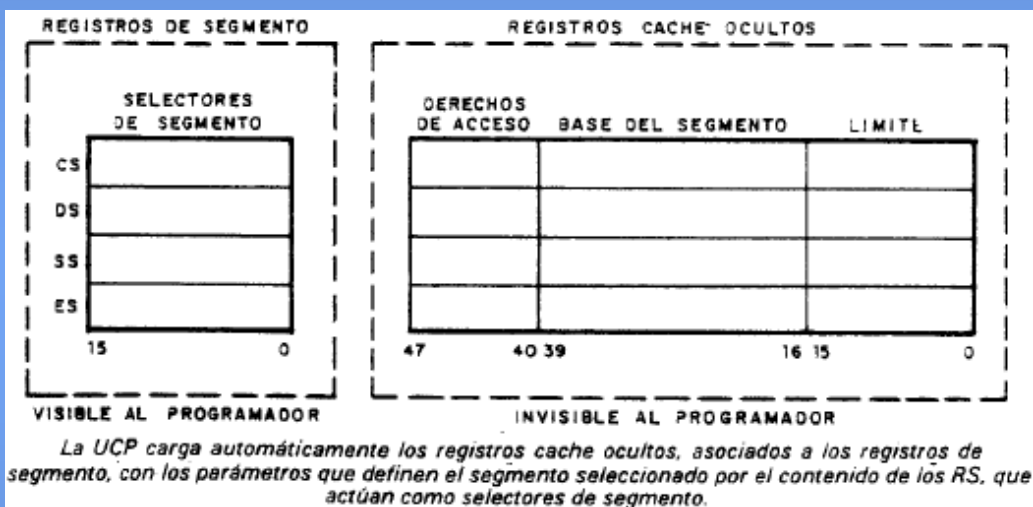
Estos parámetros son:

- Base (24 bits).
- Límite (16 bits).
- Derechos de acceso (8 bits).

Esto significa que el movimiento de información a estos registros caché lo hace la CPU y son invisibles para el programador.

Registros de Segmento

De esta forma el procesador alcanza una enorme velocidad de proceso, puesto que sólo tiene que acceder una vez a la tabla de descriptores para buscar los parámetros que definen al segmento necesario. Todo ello mientras no se modifique el valor del registro de segmento.



Tanto la tabla global como la local residen en la memoria principal y ocupan un máximo de 64 Kb, por lo que se puede almacenar hasta 8 k descriptores, en cada una. Como cada segmento puede definir un espacio máximo de 64 Kb; con cada una de las dos tablas se hace referencia a un máximo de:

$$8192 * 64 \text{ Kb} = 0,5 \text{ Gb}$$

Registros de Segmento

Los 13 bits del campo Índice del selector de Registro de Segmento seleccionan el descriptor de la GDT o la LDT en curso ($2^{13} = 8192$).

En resumen:

Dirección de Descriptor Global = $8 * \text{Índice (RS)} + \text{Base (GDTR)}$ ($T_i = 0$)

Dirección de Descriptor Local = $8 * \text{Índice (RS)} + \text{Base (LDTR)}$ ($T_i = 1$)

Por lo general el valor de GDTR no se altera puesto que los programas y los datos del S.O. suelen ser constantes. Por el contrario, el valor del LDTR cambia automáticamente por acción directa de la Unidad de Direccionamiento, cada vez que se produce una conmutación de tarea. En consecuencia, GDTR se carga en la inicialización del sistema, mientras que LDTR lo hace cada vez que se pasa de una tarea a otra.

GDTR y LDTR

Los registros que direccionan la base de las tablas global y local de descriptores son diferentes.

GDTR consta de 40 bit, 24 de los cuales contienen la base de la GDT y los restantes 16 el límite de dicha tabla.

El LDTR dispone de un registro visible y accesible por el programador de 16 bits, que tiene el formato y la aplicación de un selector de segmento. Consta de 2 bits que contienen el nivel de privilegio, luego el bit Ti, que siempre vale 0 y hace referencia a la tabla GDT, y finalmente los 13 bits restantes actúan, como índice que apunta a una entrada de la GDT. Dicha entrada contiene un descriptor de la tabla de descriptores donde se guarda la dirección de la base y el tamaño de la LDT seleccionada. Estos dos últimos valores son depositados en el registro caché asociado a LDTR.