Gestión de Memoria e Interrupciones en modo protegido

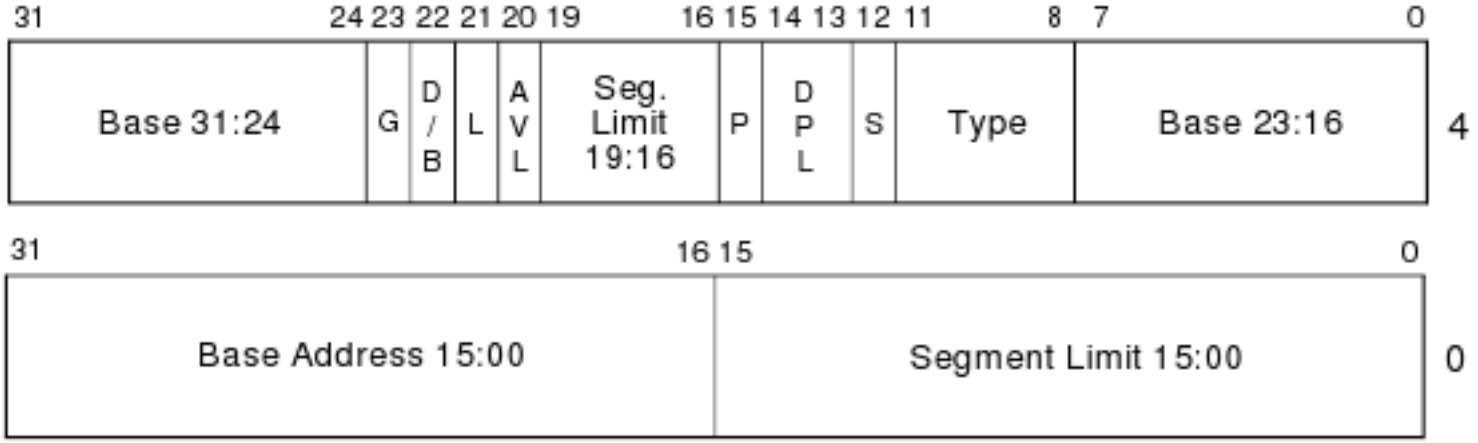
Acceso a memoria en MP (32 bits)

- Para acceder a la memoria los segmentos seguirán trabajando con segmentos, de manera similar a como lo hacen en modo real.
- La diferencia está en la información que se necesita para definir un segmento:
  - Dirección a partir de la cual comienza el segmento. La llamamos **Dirección Base**.
  - Tamaño del segmento. Intel lo denomina **Límite**.
  - Permisos de acceso al segmento, ejemplo: Lectura, Escritura, si es de Código, de Datos o del Sistema; y demás características que ahora serán rigurosamente chequeadas. Los denominaremos **Atributos**.

Descriptores de Segmento en MP

- La información necesaria para describir un segmento se almacena fuera del procesador en la memoria RAM.
- La estructura que define un segmento se denomina descriptor.
- Estos se agrupan en tablas.

Estructura de un descriptor



L — 64-bit code segment (IA-32e mode only)
AVL — Available for use by system software
BASE — Segment base address
D/B — Default operation size (0 = 16-bit segment; 1 = 32-bit segment)
DPL — Descriptor privilege level
G — Granularity
LIMIT — Segment Limit
P — Segment present
S — Descriptor type (0 = system; 1 = code or data)
TYPE — Segment type

Atributos de un segmento en MP

- L: Si es de código de 64 bits (sólo en IA-32e)
- AVL: Bit disponible para el desarrollador del SO
- D/B: Default, para segmentos de código, define si el segmento opera con 16 o 32 bits. Big, para segmentos de pila, define si opera con ESP o SP
- DPL: Nivel de privilegio del segmento
- G: Granularidad, multiplicador por 4K
- P: Presente
- S: Sistema (S=0 del sistema)

De registro de segmento a selector

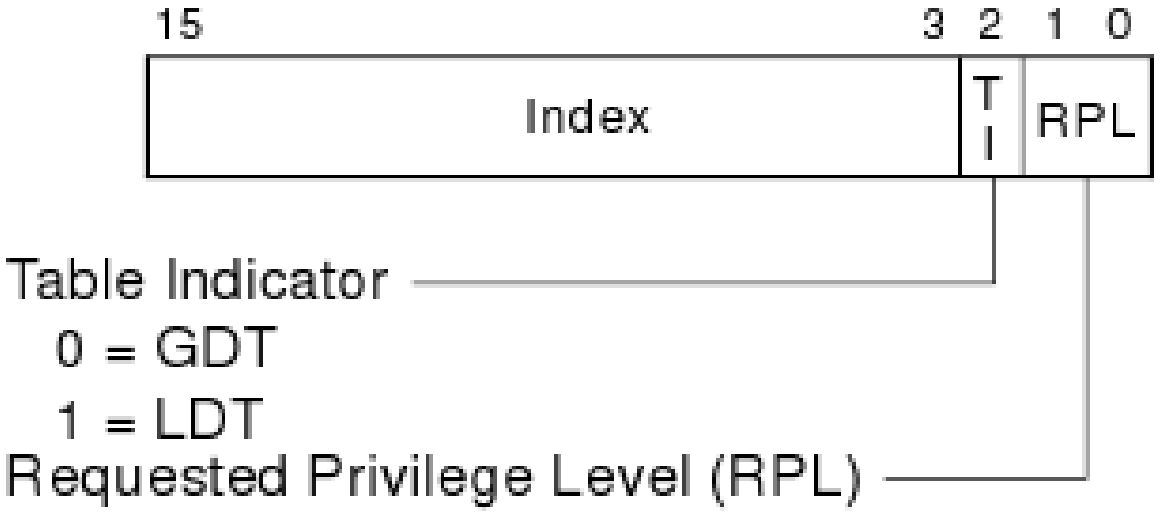
- En modo real los registros de segmento contenían toda la información necesaria para definir el segmento, **la base**. El límite está dado (64KB) y no tiene atributo.
- En Modo Protegido los registros de segmento siguen conteniendo información necesaria para acceder al segmento, pero en forma indirecta apunta a un descriptor en una **tabla de descriptores**.

Tabla de descriptores

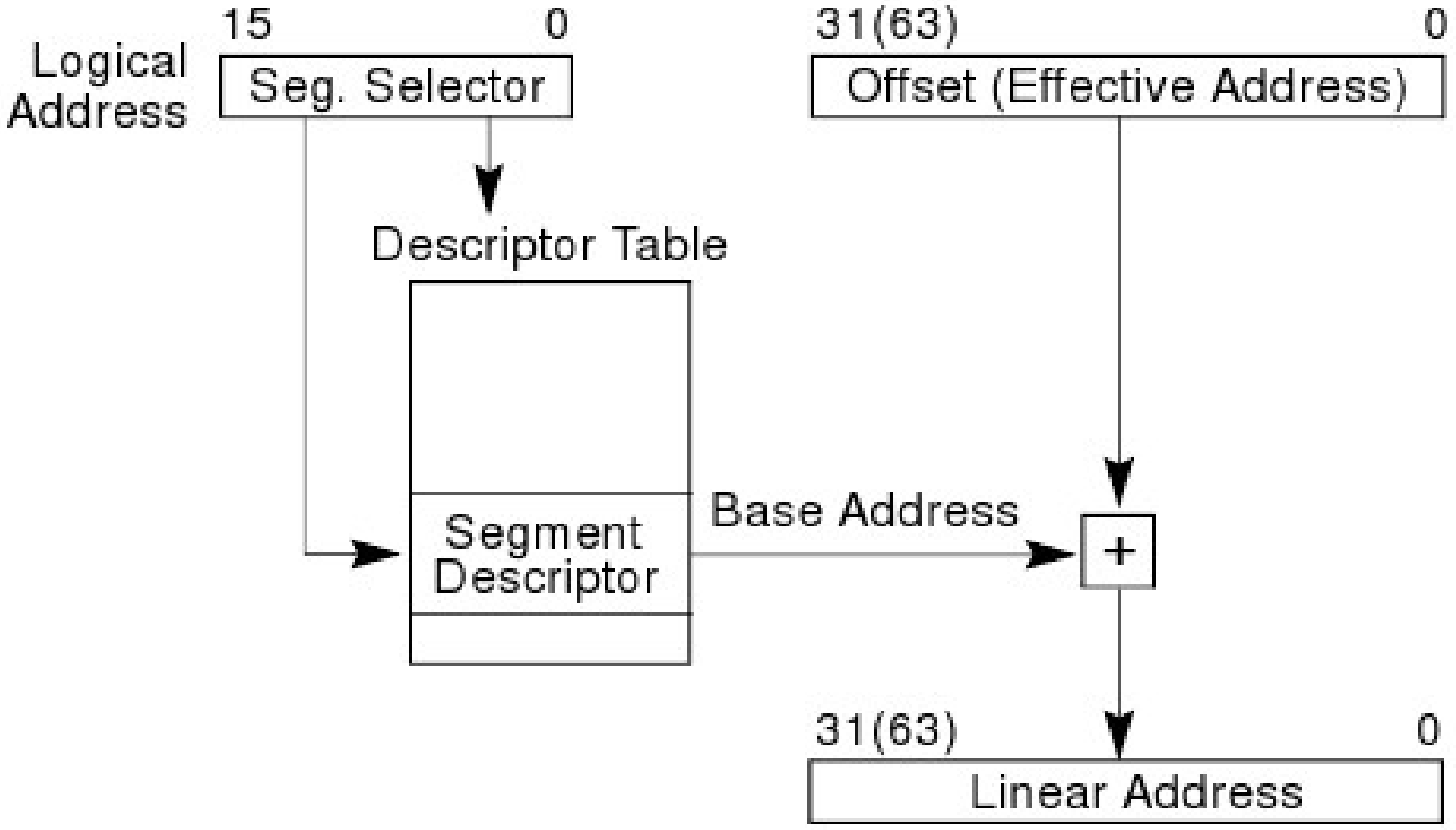
- En el sistema hay dos tipos de tablas: GDT y LDT
- La GDT es única para todo el sistema
- La dirección base de GDT está almacenada en el registro GDTR
- El GDTR tiene dos partes: Dirección Lineal Base y límite
- En el sistema puede haber mas de una LDT, pero sólo una activa
- Las LDT están descriptas en la GDT
- La dirección base de la LDT está almacenada en el registro LDTR
- El LDTR es un índice a la GDT

Selectores

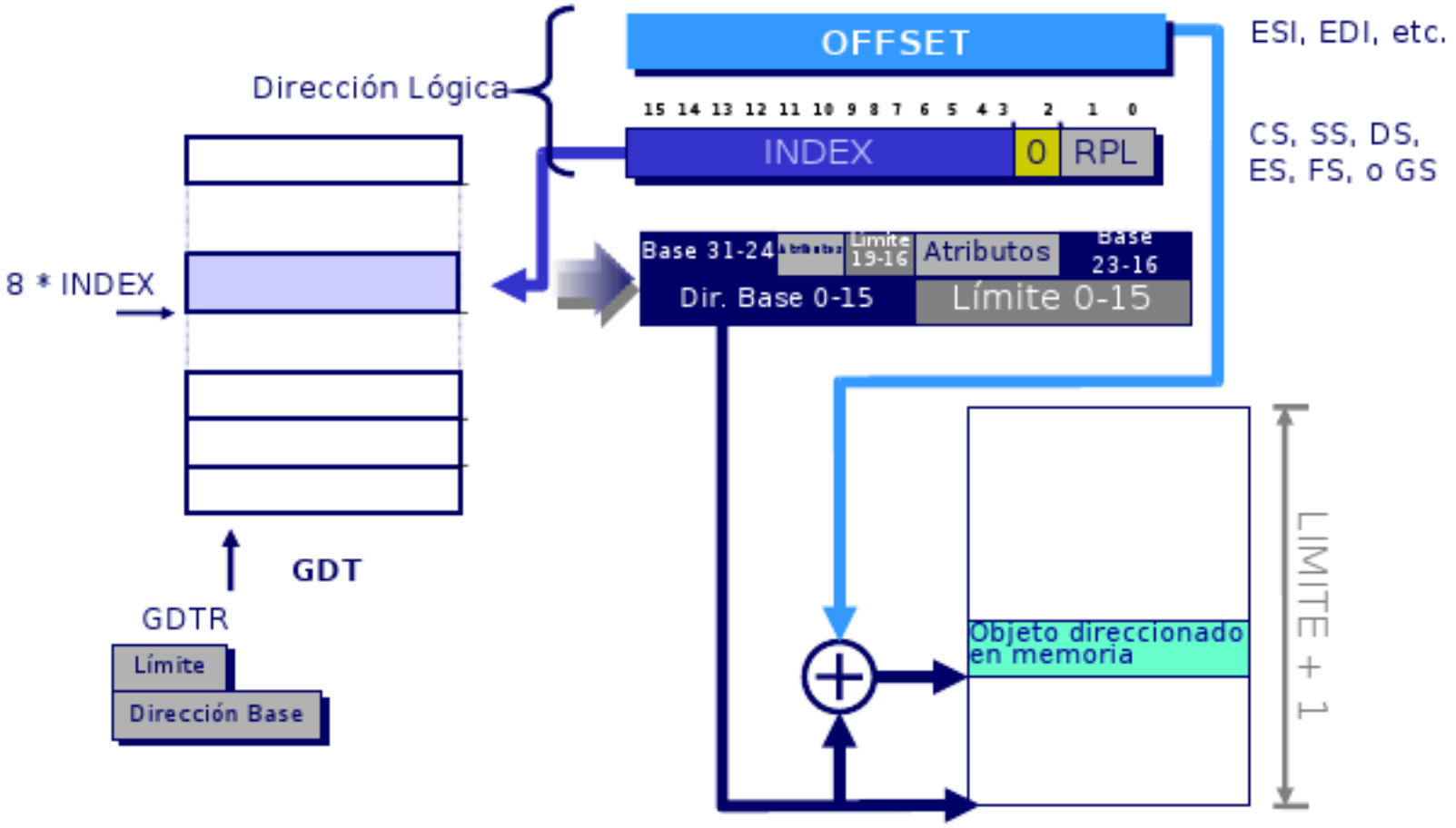
- Por tal motivo al contenido de un registro de segmento, en Modo Protegido se lo denomina **Selector de segmento**.



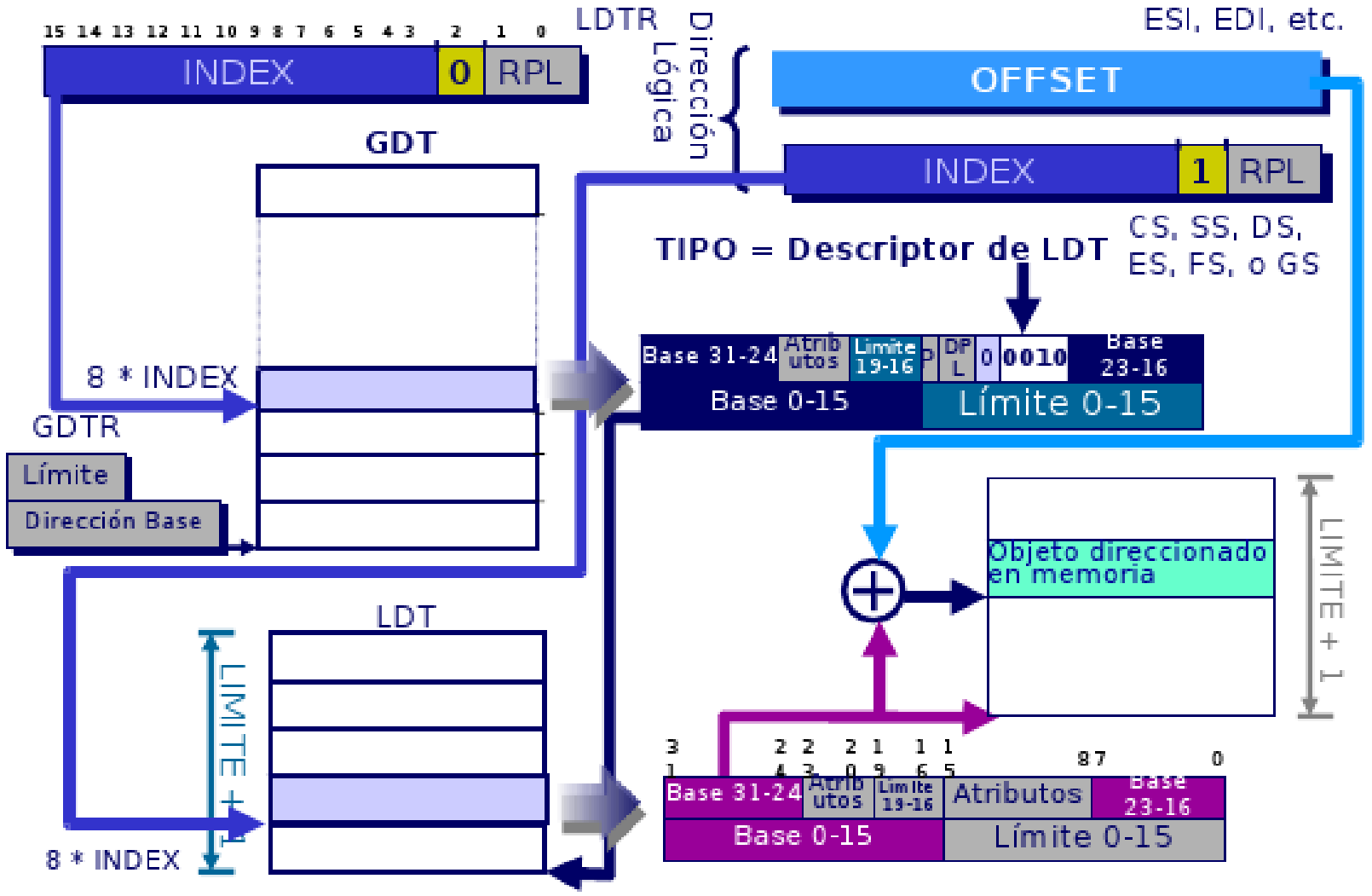
De dirección lógica a lineal en MP



Trabajando con GDT



Trabajando con LDT



Registros ocultos

- Para evitar un acceso a la GDT o LDT, cada vez que necesita un descriptor de segmento el procesador mantiene un registro caché invisible por cada registro cuyo contenido sea un selector.
- Esos registros no son accesibles ni siquiera al programador del S.O.
- Sólo se ejecuta el procedimiento citado cada vez que se altera el valor de algún registro selector de segmento.

Dirección lineal

- La Dirección Lineal recibe ese nombre por ser la salida de la Unidad de Segmentación. Es un espacio contiguo y consecutivo de direcciones de memoria.

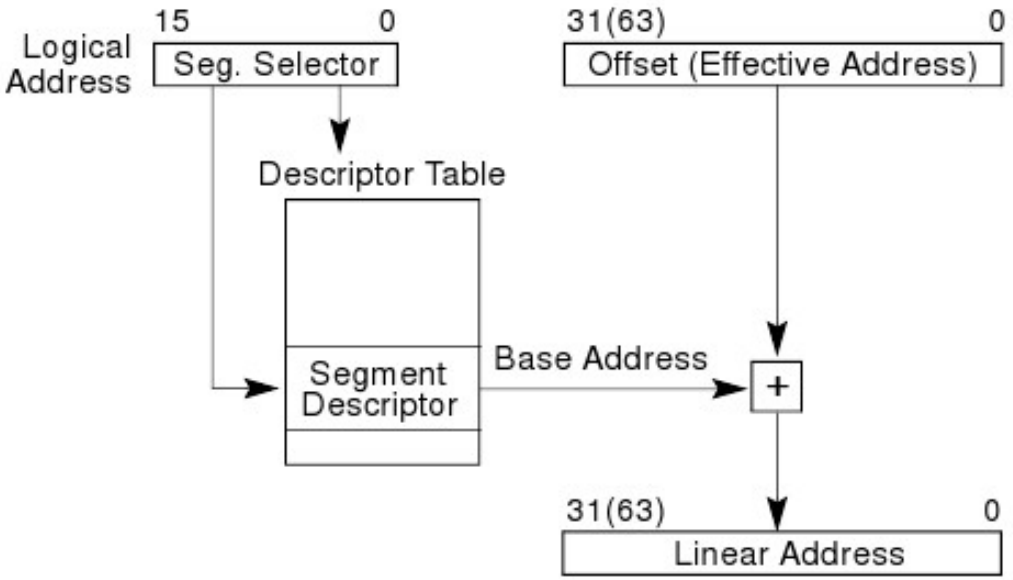


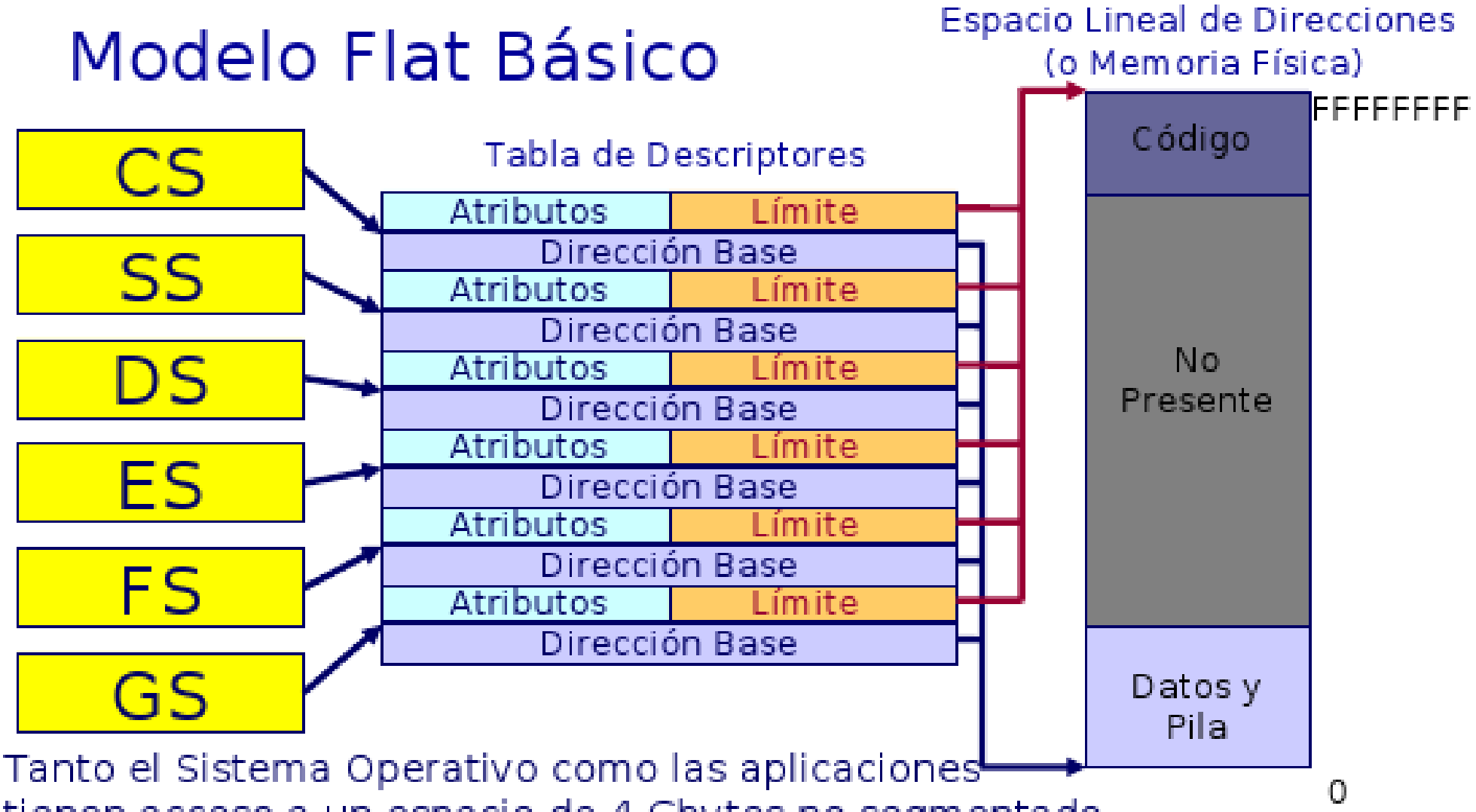
Table 3-2. System-Segment and Gate-Descriptor Types

Type Field					Description	
Decimal	11	10	9	8	32-Bit Mode	IA-32e Mode
0	0	0	0	0	Reserved	Upper 8 byte of an 16-byte descriptor
1	0	0	0	1	16-bit TSS (Available)	Reserved
2	0	0	1	0	LDT	LDT
3	0	0	1	1	16-bit TSS (Busy)	Reserved
4	0	1	0	0	16-bit Call Gate	Reserved
5	0	1	0	1	Task Gate	Reserved
6	0	1	1	0	16-bit Interrupt Gate	Reserved
7	0	1	1	1	16-bit Trap Gate	Reserved
8	1	0	0	0	Reserved	Reserved
9	1	0	0	1	32-bit TSS (Available)	64-bit TSS (Available)
10	1	0	1	0	Reserved	Reserved
11	1	0	1	1	32-bit TSS (Busy)	64-bit TSS (Busy)
12	1	1	0	0	32-bit Call Gate	64-bit Call Gate
13	1	1	0	1	Reserved	Reserved
14	1	1	1	0	32-bit Interrupt Gate	64-bit Interrupt Gate
15	1	1	1	1	32-bit Trap Gate	64-bit Trap Gate

Table 3-1. Code- and Data-Segment Types

Type Field					Descriptor Type	Description
Decimal	11	10 E	9 W	8 A		
0	0	0	0	0	Data	Read-Only
1	0	0	0	1	Data	Read-Only, accessed
2	0	0	1	0	Data	Read/Write
3	0	0	1	1	Data	Read/Write, accessed
4	0	1	0	0	Data	Read-Only, expand-down
5	0	1	0	1	Data	Read-Only, expand-down, accessed
6	0	1	1	0	Data	Read/Write, expand-down
7	0	1	1	1	Data	Read/Write, expand-down, accessed
		C	R	A		
8	1	0	0	0	Code	Execute-Only
9	1	0	0	1	Code	Execute-Only, accessed
10	1	0	1	0	Code	Execute/Read
11	1	0	1	1	Code	Execute/Read, accessed
12	1	1	0	0	Code	Execute-Only, conforming
13	1	1	0	1	Code	Execute-Only, conforming, accessed
14	1	1	1	0	Code	Execute/Read, conforming
15	1	1	1	1	Code	Execute/Read, conforming, accessed

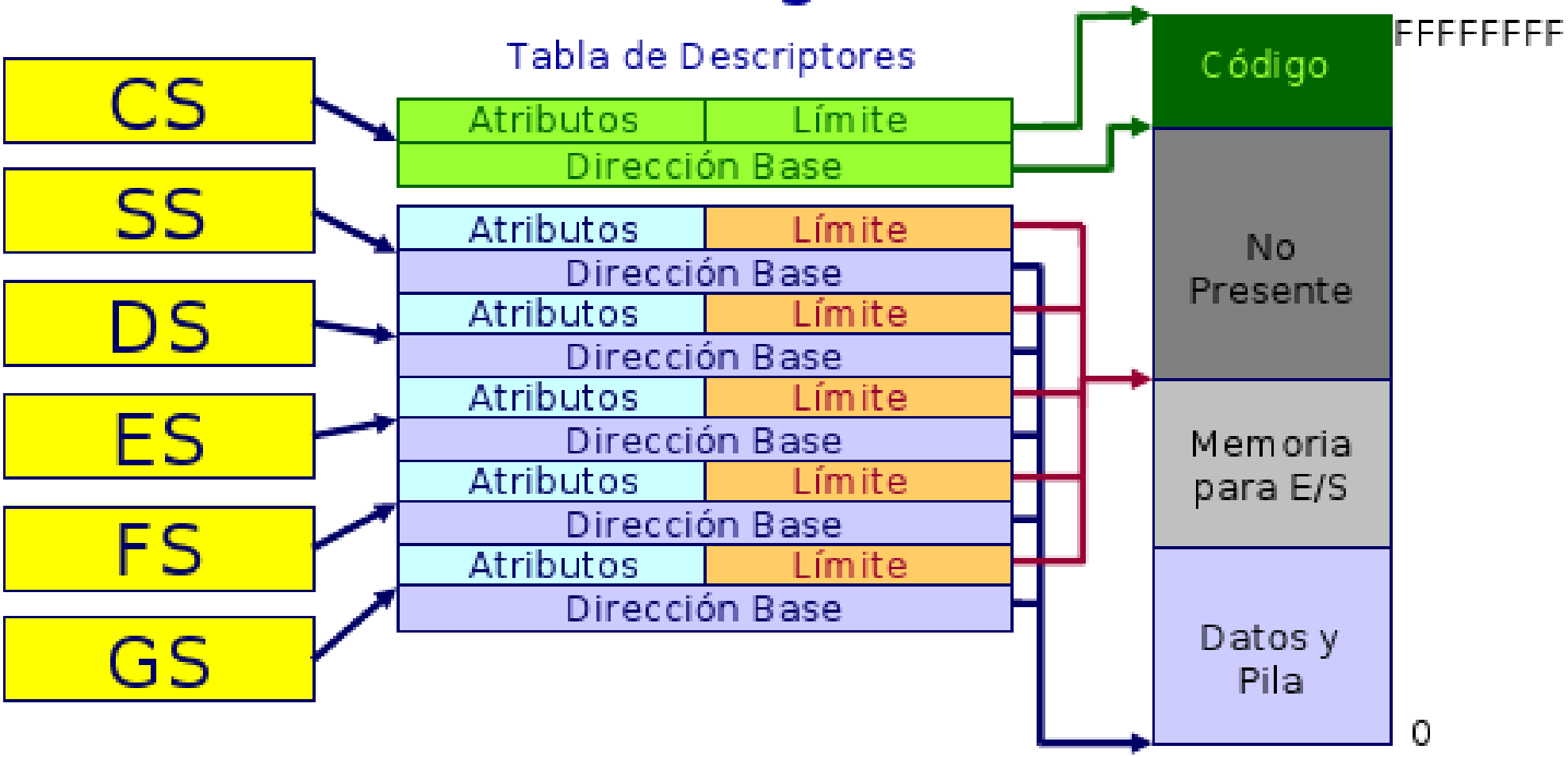
Modelo Flat Básico



Tanto el Sistema Operativo como las aplicaciones tienen acceso a un espacio de 4 Gbytes no segmentado.

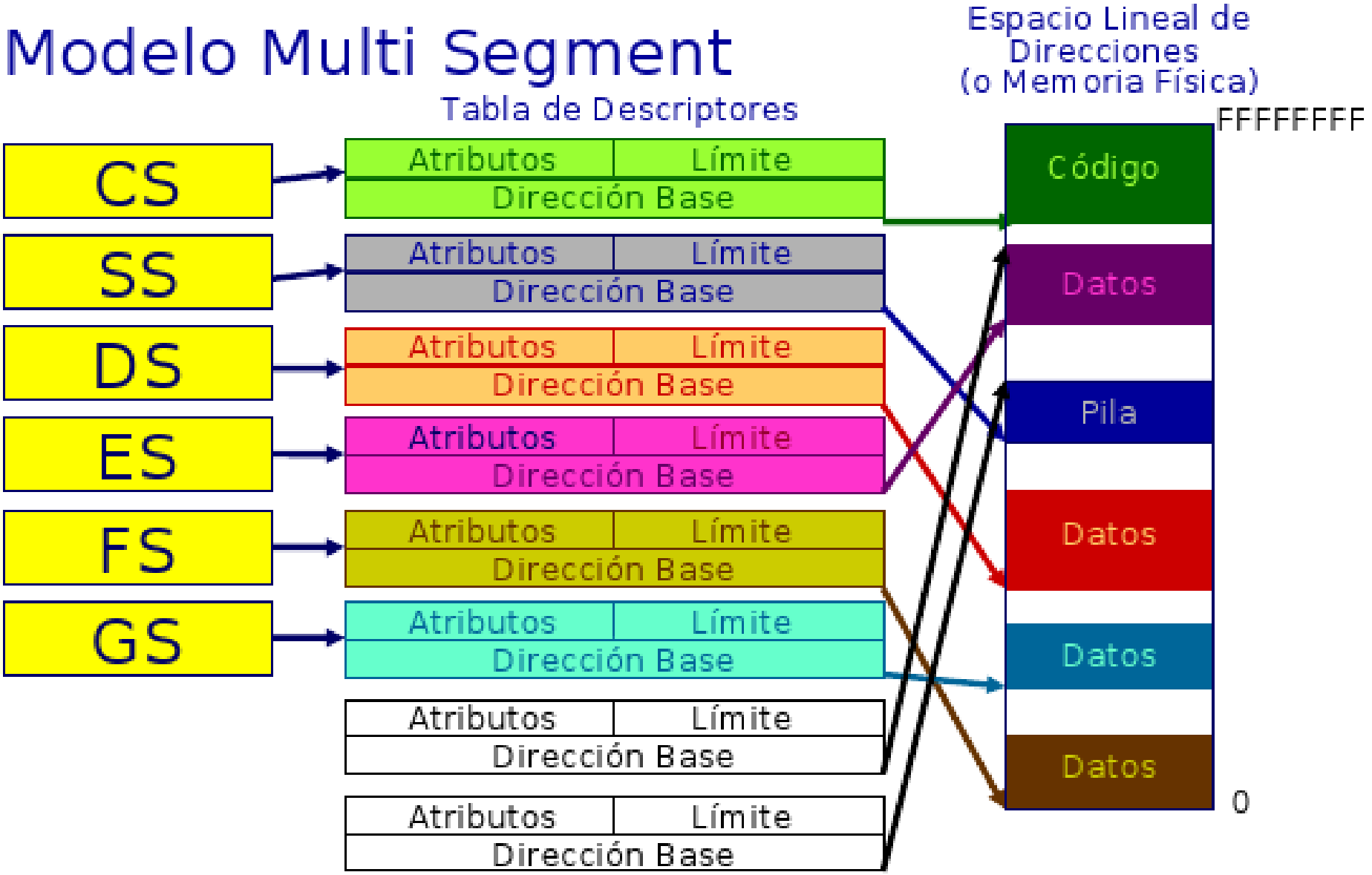
Se evitan las excepciones por exceso en ellímite de memoria ya que el límite de todos los descriptores es FFFFFFFF. Aún si se accede a áreas en las que no existe memoria física.

Modelo Flat Protegido



Los segmentos tienen el límite acorde a la memoria física instalada en el sistema.

Modelo Multi Segment



Acceso a memoria en IA-32e

- En IA32e no utiliza segmentación
- CS, DS, SS, ES asumen segmento FLAT, base 0.
- FS y GS no asumen base 0
- No se chequea el límite de segmento

Interrupciones

- En cualquiera de los modos de trabajo del procesador las interrupciones se identifican mediante un número de un byte llamado **tipo**
- Tiene 256 tipos diferentes de interrupción
- El sistema de interrupciones fundado por el 8086, se mantuvo invariable en sus sucesores

Interrupciones y Excepciones

- Origen de las interrupciones:
  - Externa, por el hardware en interrupciones que ingresan por el **pin INTR** del procesador.
  - Externa, **pin NMI** son de tipo 2.
  - Interna la instrucción **INT <type>**, para el caso de las interrupciones por software, por ejemplo INT 21h.
  - Interna, producto de la detección de algún error, ejemplo división por cero. Se las denomina **Excepciones**

Clasificación de las excepciones

- **Fault**: Excepción que puede corregirse permitiendo al programa retomar la ejecución de esa instrucción sin perder continuidad. El procesador guarda en la pila la dirección de la instrucción que produjo la falla.
- **Traps**: Excepción producida inmediatamente a continuación de la ejecución de una instrucción. Algunas permiten al procesador retomar la ejecución sin perder continuidad. Otras no. El procesador guarda en la pila la dirección de la instrucción a ejecutarse luego de la instrucción trapeada.
- **Aborts**: Excepción que no siempre puede determinar la instrucción que la causó, ni permite recuperar la ejecución de la tarea que la causó. Reporta errores severos de hardware o inconsistencias en tablas del sistema.

Table 6-1. Protected-Mode Exceptions and Interrupts

Vector No.	Mne-monic	Description	Type	Error Code	Source
0	#DE	Divide Error	Fault	No	DIV and IDIV instructions.
1	#DB	RESERVED	Fault/ Trap	No	For Intel use only.
2	—	NMI Interrupt	Interrupt	No	Nonmaskable external interrupt.
3	#BP	Breakpoint	Trap	No	INT 3 instruction.
4	#OF	Overflow	Trap	No	INTO instruction.
5	#BR	BOUND Range Exceeded	Fault	No	BOUND instruction.
6	#UD	Invalid Opcode (Undefined Opcode)	Fault	No	UD2 instruction or reserved opcode.[1]
7	#NM	Device Not Available (No Math Coprocessor)	Fault	No	Floating-point or WAIT/FWAIT instruction.
8	#DF	Double Fault	Abort	Yes (zero)	Any instruction that can generate an exception, an NMI, or an INTR.
9		Coprocessor Segment Overrun (reserved)	Fault	No	Floating-point instruction.[2]
10	#TS	Invalid TSS	Fault	Yes	Task switch or TSS access.
11	#NP	Segment Not Present	Fault	Yes	Loading segment registers or accessing system segments.
12	#SS	Stack-Segment Fault	Fault	Yes	Stack operations and SS register loads.
13	#GP	General Protection	Fault	Yes	Any memory reference and other protection checks.
14	#PF	Page Fault	Fault	Yes	Any memory reference.
15	—	(Intel reserved. Do not use.)		No	
16	#MF	x87 FPU Floating-Point Error (Math Fault)	Fault	No	x87 FPU floating-point or WAIT/FWAIT instruction.
17	#AC	Alignment Check	Fault	Yes (Zero)	Any data reference in memory.[3]

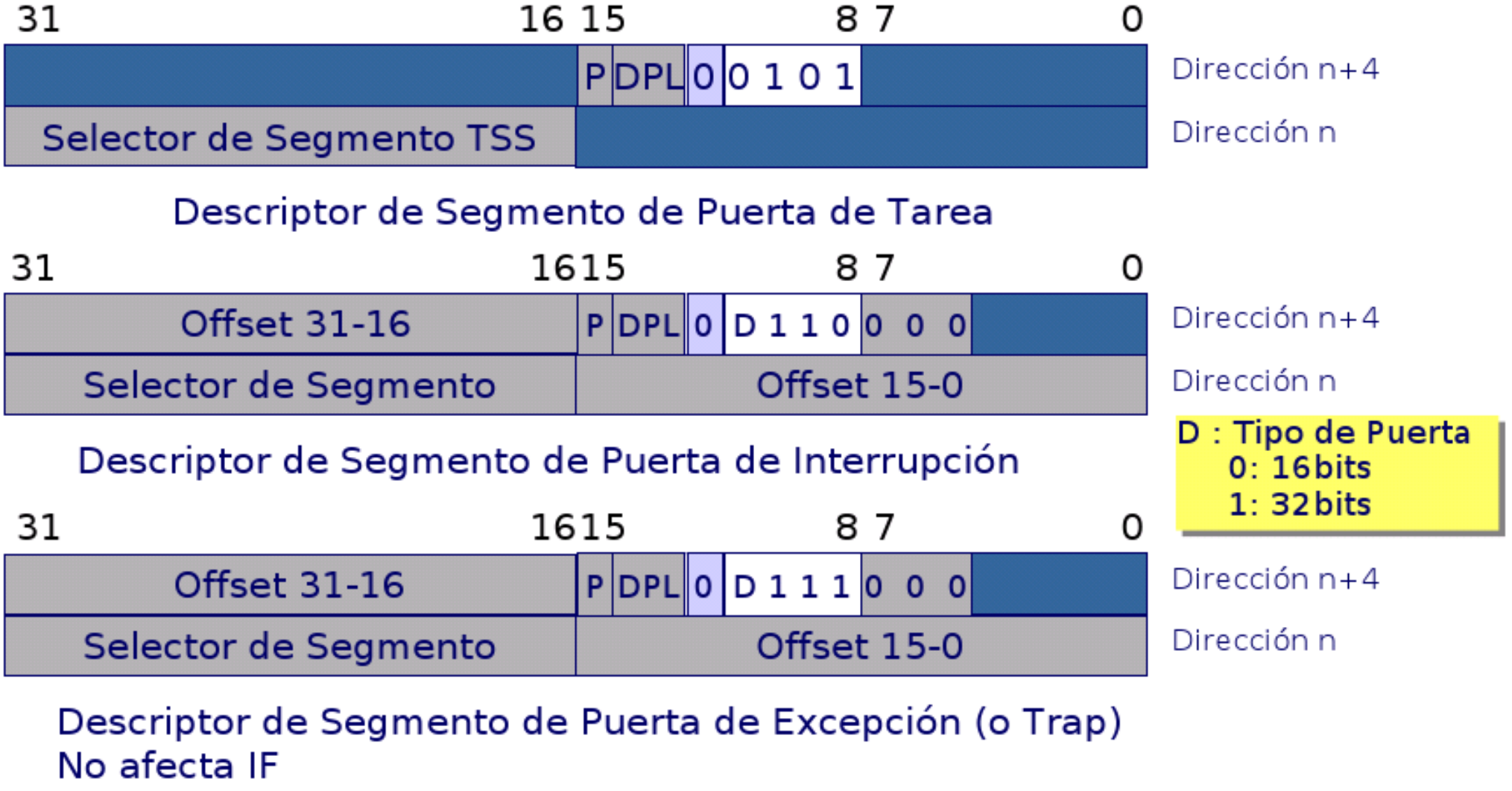
Interrupciones en modo protegido

- IDT (Interrupt Descriptor Table), que almacena Descriptores, similares a los vistos en la GDT o LDT
- Esta tabla tiene únicamente 256 entradas, coincidiendo con la cantidad de tipos de interrupciones diferentes que maneja el microprocesador
- No se puede definir en la IDT un descriptor de segmento de datos **ni de código**.
- Son descriptores del sistema (Bit S=0 en el descriptor).

Descriptores en la IDT en MP

- Interrupt Gate
  - Limpia Interrupt Flag (IF=0), no puede ser interrumpido nuevamente, pero sí por una excepción o una interrupción no enmascarable.
- Trap Gate
  - No afecta Interrupt Flag
- Task Gate
  - Cambia de tarea

Descriptores en la IDT en MP (2)



Registro IDTR

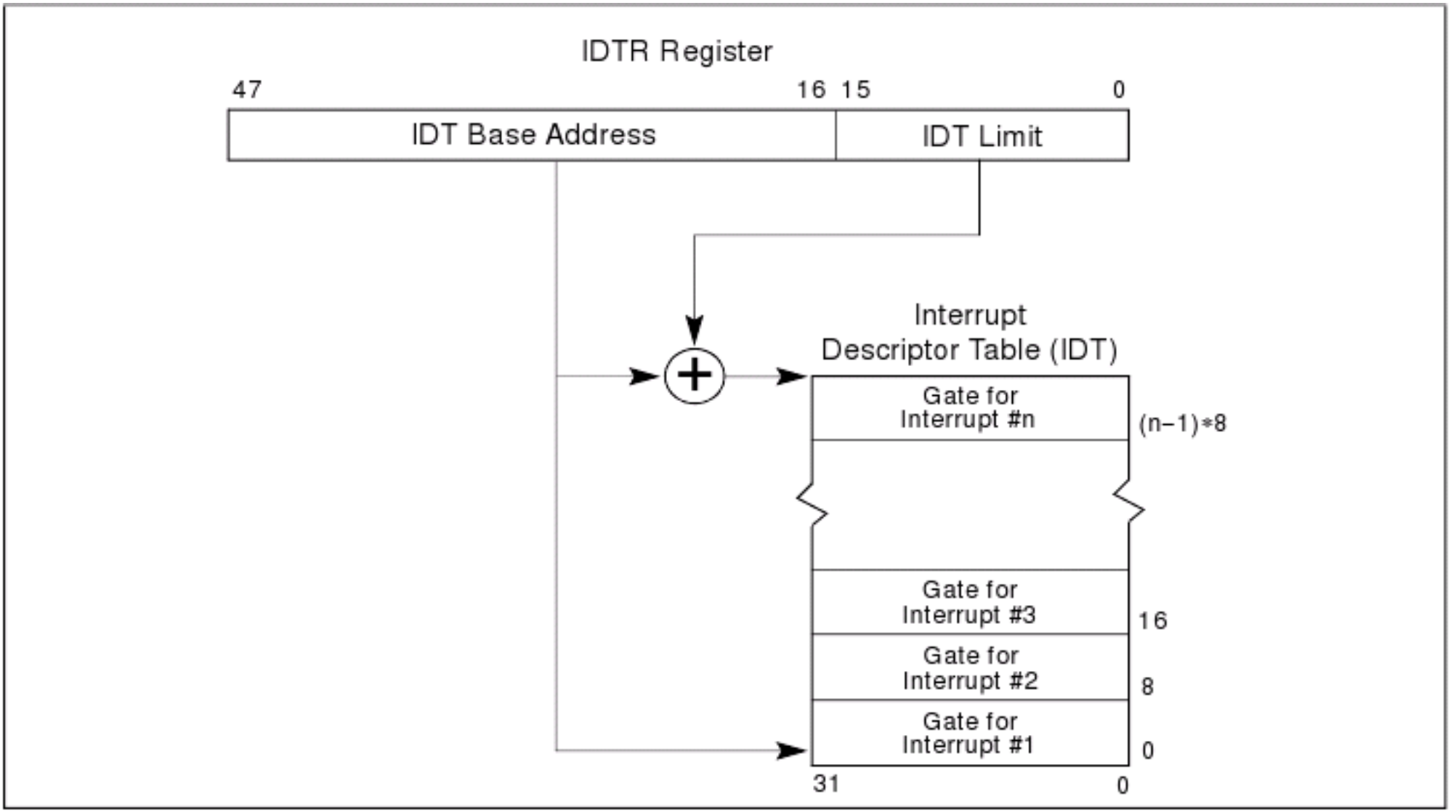
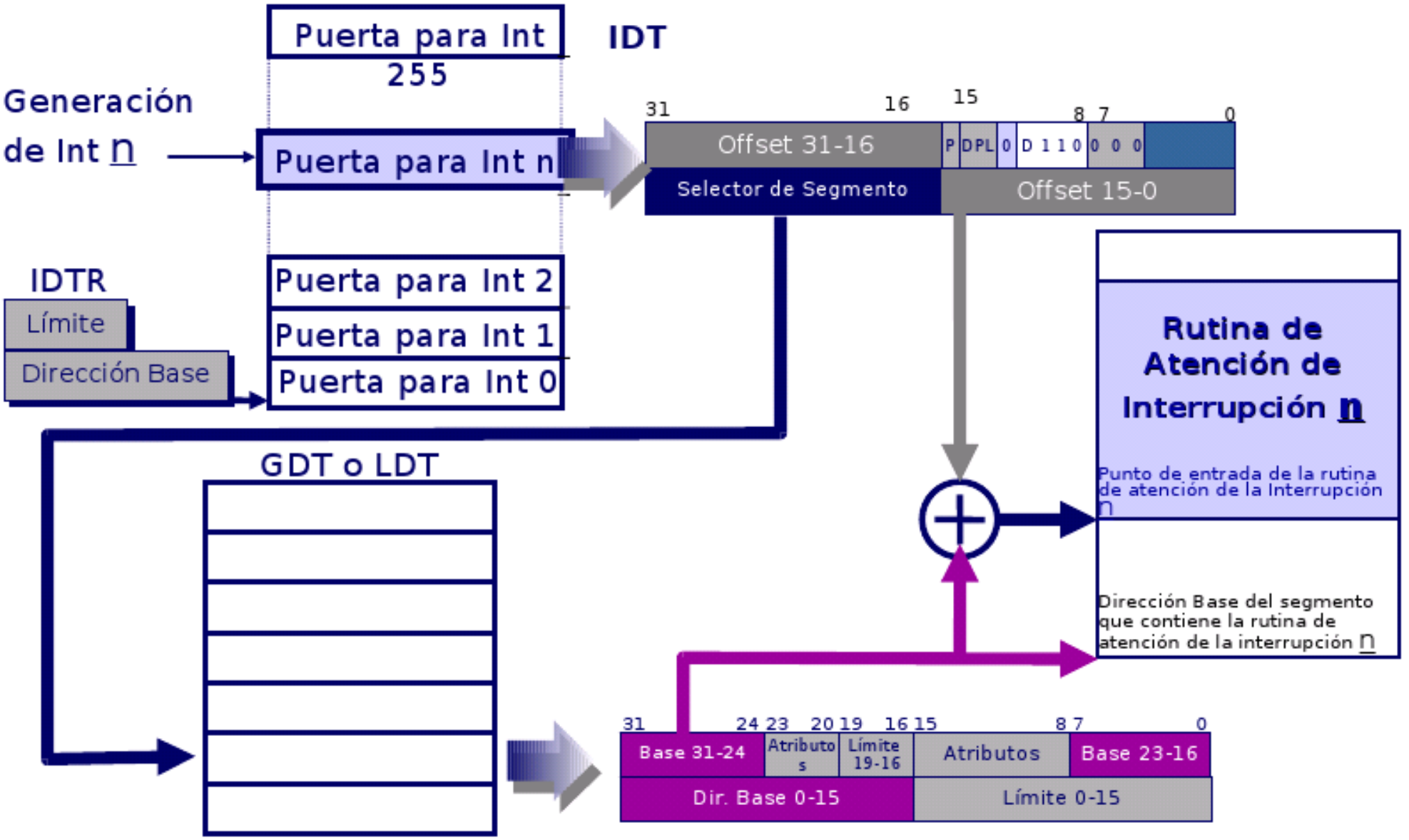
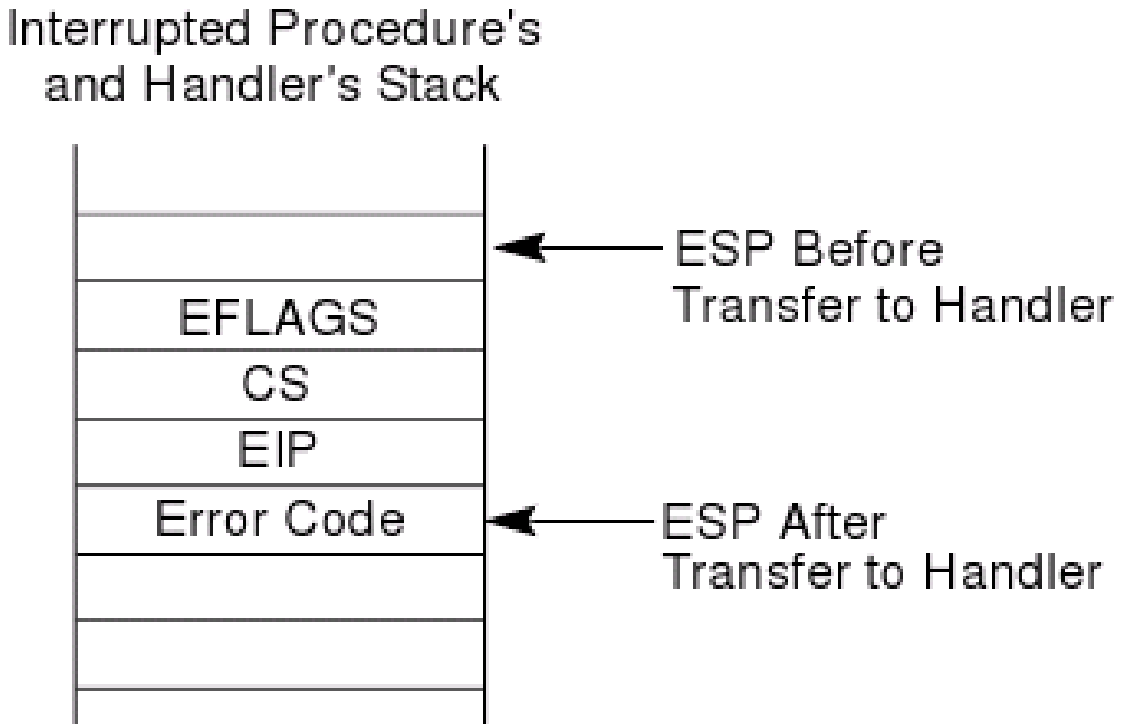


Figure 6-1. Relationship of the IDTR and IDT

Procedimiento de Interrupción



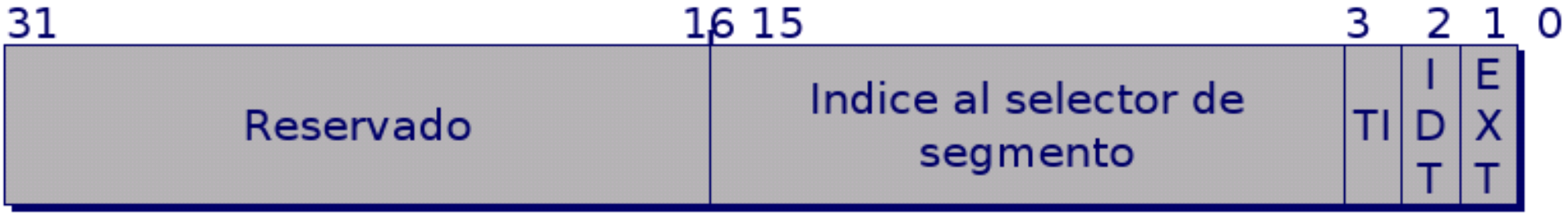
Manejo de la Pila: Sin cambio de nivel de privilegio



SP o ESP apunta al último dato almacenado, estos serán de 16 o 32 bits dependiendo del tipo de segmento de código en operación.

Código de error

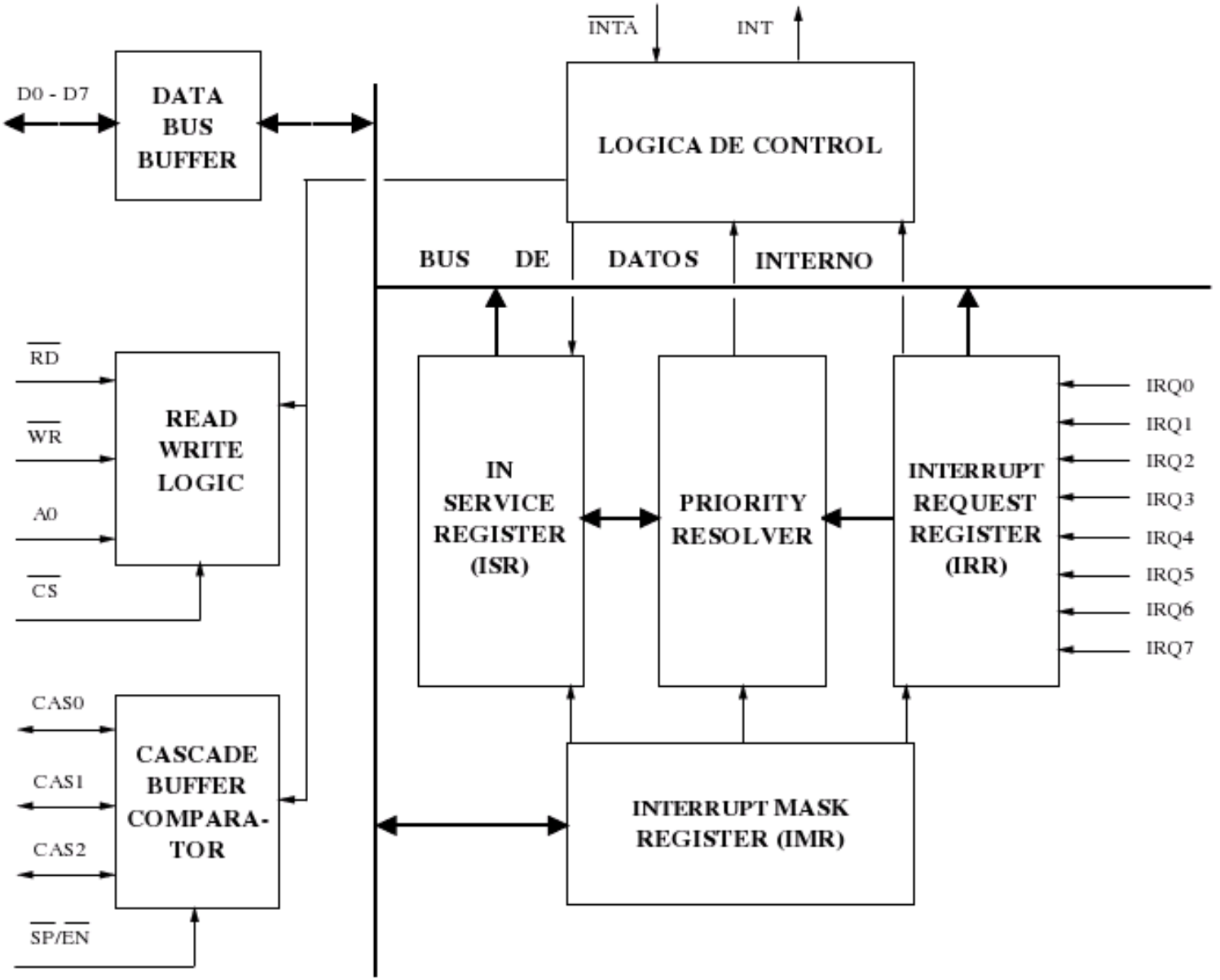
- EXT: External Event (bit 0): Se setea para indicar que la excepción ha sido causada por un evento externo al procesador
- IDT: Descriptor Location (bit 1): Cuando está seteado indica que el campo Segment Selector Index se refiere a un descriptor de puerta en la IDT: Cuando está en cero indica que dicho campo se refiere a un descriptor en la GDT o en la LDT de la tarea actual.
- TI: GDT/LDT (bit 2): Tiene significado cuando el bit anterior está en cero. Indica a que tabla de descriptores corresponde el selector del campo Indice. 0 GDT , 1 LDT (idéntico significado que en el selector de segmento).



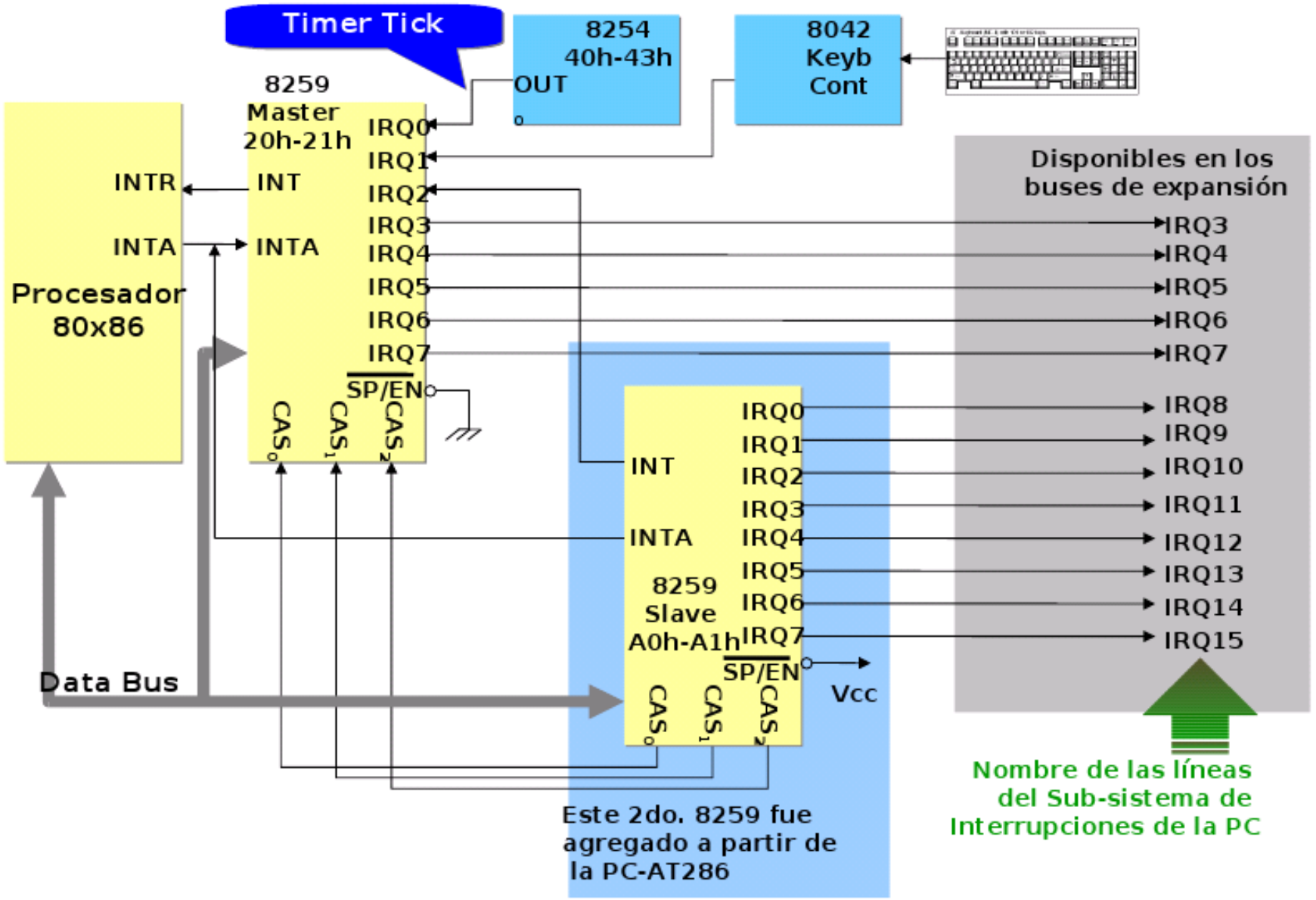
8259: Controlador Programable de Interrupciones

- El procesador sólo dispone de dos líneas de interrupción
- La forma que tiene la familia IA-32 de disponer de mas líneas de interrupción es por medio de un controlador programable de interrupciones
- El controlador recibe la interrupción, la transmite al CPU por la línea INTR y le informa cual de todas sus líneas de interrupción (IRQ's) fue por medio de bus de datos
- Las líneas de interrupción (IRQ's) son físicas
- Los tipos de interrupción (INT's) son lógicas

8259: Controlador Programable de Interrupciones



El PIC 8259 en la PC



Asignación y Tipo de las IRQ's

IRQ	Tipo	Descripción
IRQ0	08h	Timer tick (T=55 mseg.)
IRQ1	09h	Teclado
IRQ2	0Ah	INT desde 8259A esclavo
IRQ8	70h	Servicio de reloj en tiempo real.
IRQ9	71h	Redireccionamiento por soft. a IRQ2
IRQ10	72h	Reservada
IRQ11	73h	Reservada
IRQ12	74h	Reservada.
IRQ13	75h	Coprocesador numérico.
IRQ14	76h	Controlador de disco rígido.
IRQ15	77h	Reservada.
IRQ3	0Bh	COM2
IRQ4	0Ch	COM1
IRQ5	0Dh	LPT2
IRQ6	0Eh	Controlador de disco flexible (Floppy)
IRQ7	0Fh	LPT1

El PIC 8259 en la PC

- Se presenta al procesador una interfaz de programación a través de dos direcciones de E/S.
  - La primer PC tenía un solo PIC en las direcciones de port 20h y 21h.
  - La PC AT 286, incluyó un segundo PIC, ya que la PC original ya había agotado la asignación de IRQ’s, y seguían apareciendo nuevos dispositivos: Placas de red, placas de sonido, etc.
  - Este PIC es accesible en las direcciones de port A0h y A1h.

Palabras de Comando de Inicialización del 8259

Son una secuencia de entre dos y cuatro bytes que se envía al procesador al 8259A antes de comenzar la operación normal, a fin de configurarlo.

La secuencia de Inicialización en el PIC 8259 es una operación atómica, es decir, que no puede dividirse.

El 8259 detecta la secuencia de inicialización cuando recibe en la dirección de port par (A0 = 0), una palabra con el bit D4=1.

Palabras de Comando de Operación del 8259

- Una vez inicializado el 8259A, la palabras de comando le definen al 8259 diversas operaciones a realizar.
- Luego de la inicialización, se pueden enviar en cualquier momento.

Programación del PIC 8259

```
; Inicialización PIC #1
      mov al,11h      ;ICW1: IRQs activas por flanco, Modo cascada, ICW4 Si.
      out  20h,al
      mov al,8        ;ICW2: INT base para el PIC N#1 Tipo 8.
      out  21h,al
      mov al,04h      ;ICW3: PIC N#1 Master, tiene un Slave conectado a IRQ2 (0000 0100b)
      out  21h,al
      mov al,01h      ;ICW4: Modo No Buffered, Fin de Interrupción Normal, procesador 8086
      out  21h,al

; Antes de inicializar el PIC N#2, deshabilitamos las Interrupciones del PIC N#1
      mov al,0FFh   :OCW1: Set o Clear el IMR
      out  21h,al

; Inicialización PIC N #2
      mov al,11h      ;ICW1: IRQs activas por flanco, Modo cascada, ICW4 Si.
      out  0A0h,al
      mov al,070h     ;ICW2: INT base para el PIC N#1 Tipo 070h.
      out  0A1h,al
      mov al,02h      ;ICW3: PIC N#2 Slave, IRQ2 es la línea que envía al Master (010b)
      out  0A1h,al
      mov al,01h       ;ICW4: Modo No Buffered, Fin de Interrupción Normal, procesador 8086
      out  0A1h,al
```

Programación del PIC 8259 (2)

```
; Enmascarar interrupciones del PIC #1
mov al, 11111101b
out 21h, al
; Enmascarar interrupciones del PIC #2
mov al, 11111111b
out 0A1h, al
```

Programación del PIC 8259 (3)

```
; Al final de manejador de interrupción
mov al, 20h
out 20h, al
iret
```

Hardware de Soporte: 8253 / 8254

- Timer 0: Dirección 40h. Base de tiempos del sistema. A la entrada CLK0 se conecta un cristal de 1,193,180 MHz. Se programa para generar por OUT0 un pulso cada 55 mseg. Este pin se conecta a la línea IRQ0 del PIC 8259: así se genera una interrupción a dicho intervalo.
- Timer 1: Dirección 41h. Se programa del mismo modo que Timer 0 pero se lo utiliza para activar el sistema de refresco de memoria DRAM. Para ello, OUT1 va conectado a la entrada DREQ0 del 8237. A partir del modelo AT 286 se utilizará un hardware dedicado al refresco de memoria.
- Timer 2: Dirección 42h. Se programa para generar a su salida una señal cuadrada de 50% de duty cycle y OUT2 se conecta al parlante del sistema.
- Registro de comando y status. Dirección 43h.

Hardware de Soporte: 8255

- Port A: Dirección 60h. La lógica de control de teclado al recibir los códigos de las teclas los almacena en este port.
- Port B: Dirección 61h. Sus líneas trabajan como salidas de control individuales.
  - Envío de la salida del Timer al parlante de la PC
  - Envío de un pulso a la línea IRQ1 del PIC 8259 cada vez que se recibe un código de tecla desde el teclado.
- Port C: Dirección 62h. Cuatro de estas líneas se utilizan como información de configuración del sistema. El sistema de configuración en los modelos originales PC y PC-XT era sumamente rudimentario (trabajaba con dip switches).

Pasaje a modo protegido desde modo real

CLI

Armar GDT, el primer descriptor debe ser nulo

Armar IDT y LDT (opcional)

Palabras de Comando de Operación de los 8259's (opcional)

Habilitar el bit PE

Jmp far

STI (opcional)

Práctica

~/bochs/tp1/ & ~/bochs/

- Comando de consola debug de bochs:
  - Info gdt
  - Info idt
  - r
  - sreg
  - creg
  - vb 0x8:0x21
- Está bien “Copiar – Pegar”, sólo si es código que ustedes hicieron.

Referencias

- Intel® 64 and IA-32 Architectures Software Developer’s Manual Volume 3A: System Programming Guide, Capítulos 2, 3 y 5
- Organización y Arquitectura de Computadores. 5ta. Ed. William Stallins, Capítulos 4, y 7