Interrupciones en Modo Protegido

Organizacion del Computador II

Javier Pimás

Departamento de Computacion
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Primer Cuatrimestre 2014

Refresco de Memoria

- Soporta 256 tipos de interrupciones
- Se utiliza una tabla denominada IDT
- La IDT almacena descriptores de interrupcion
- El registro IDTR almacena la direccion de la IDT

IDTR and IDT

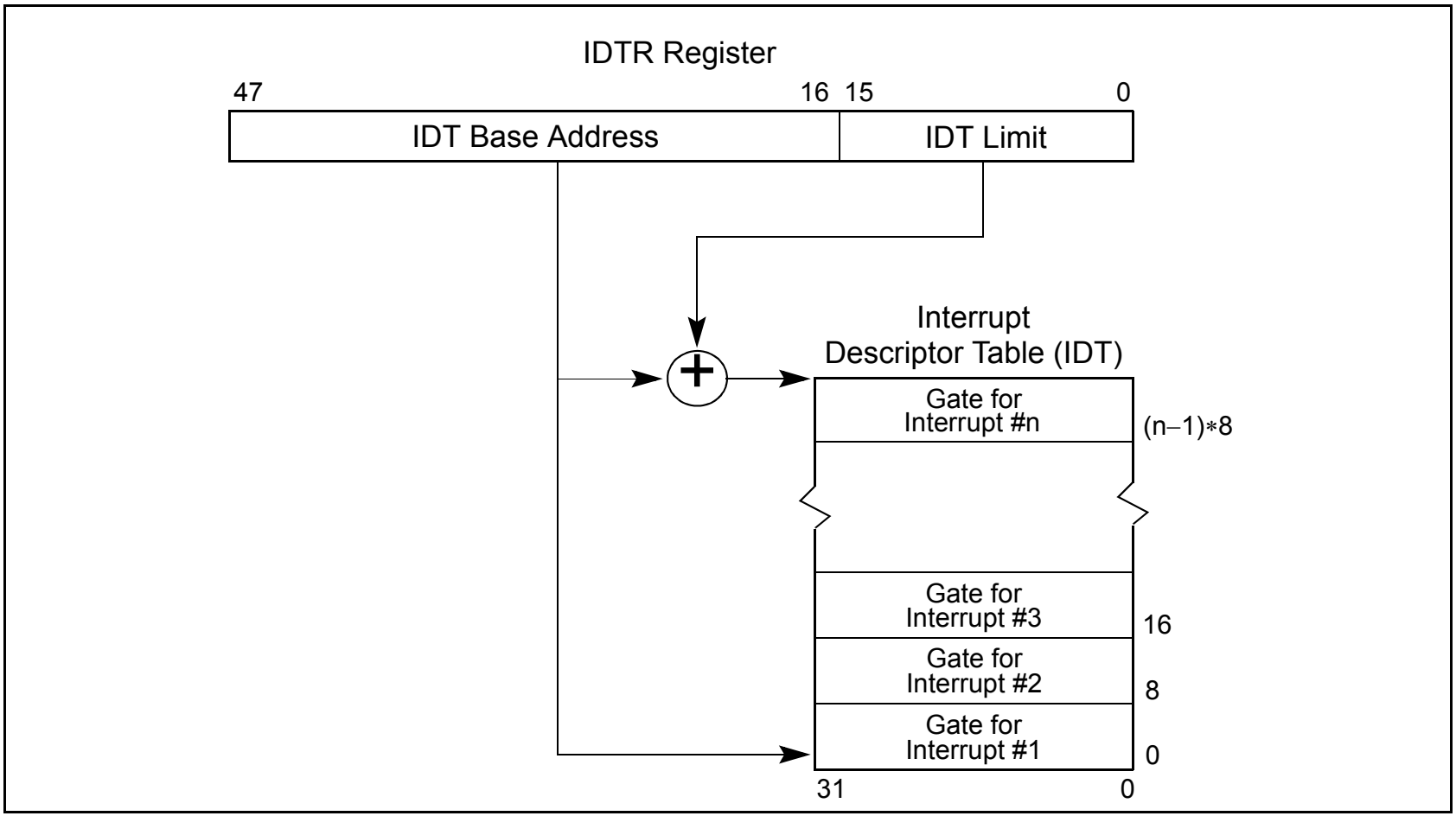
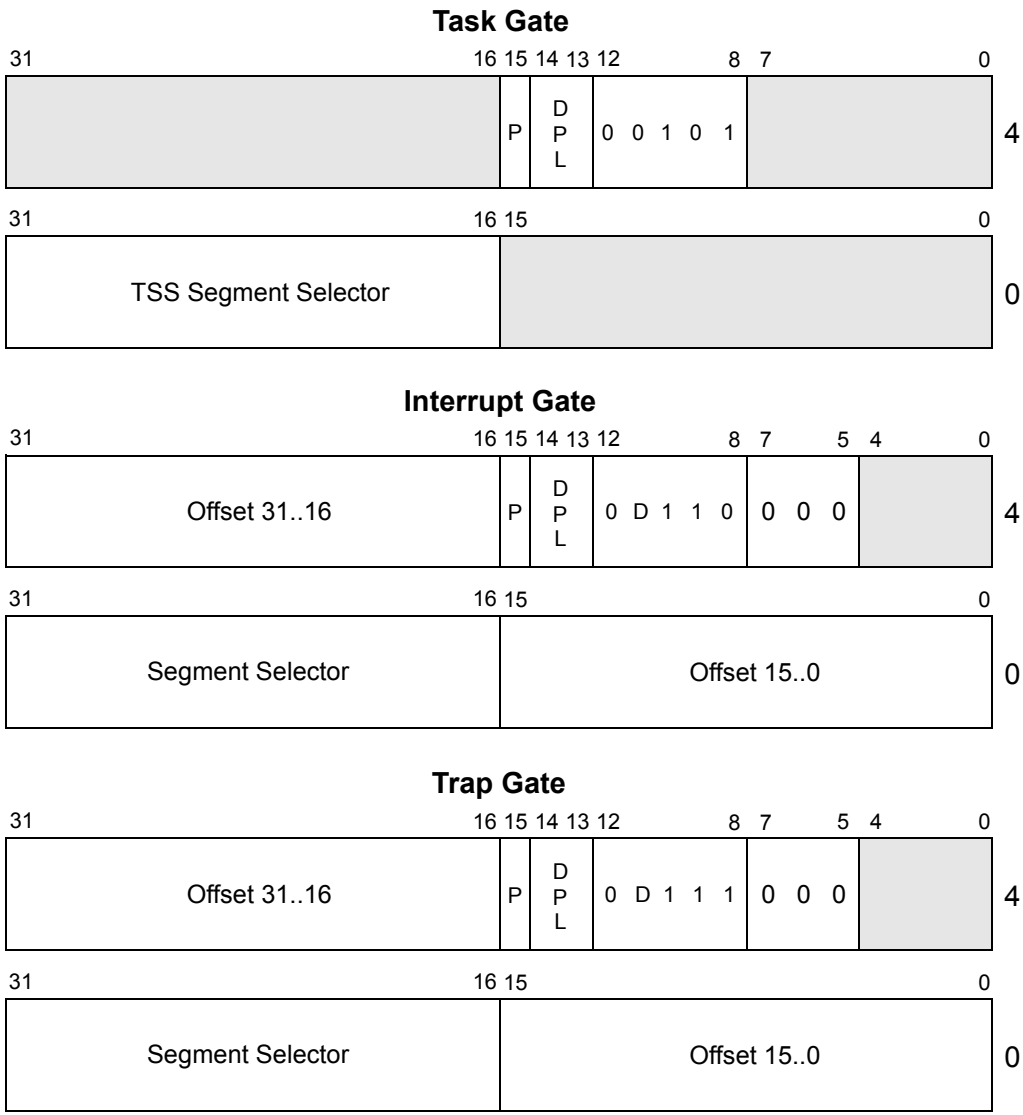


Figure 5-1. Relationship of the IDTR and IDT

Gate Descriptors



DPL	Descriptor Privilege Level
Offset	Offset to procedure entry point
P	Segment Present flag
Selector	Segment Selector for destination code segment
D	Size of gate: 1 = 32 bits; 0 = 16 bits

Reserved

Figure 5-2. IDT Gate Descriptors

Interrupt Procedure Call

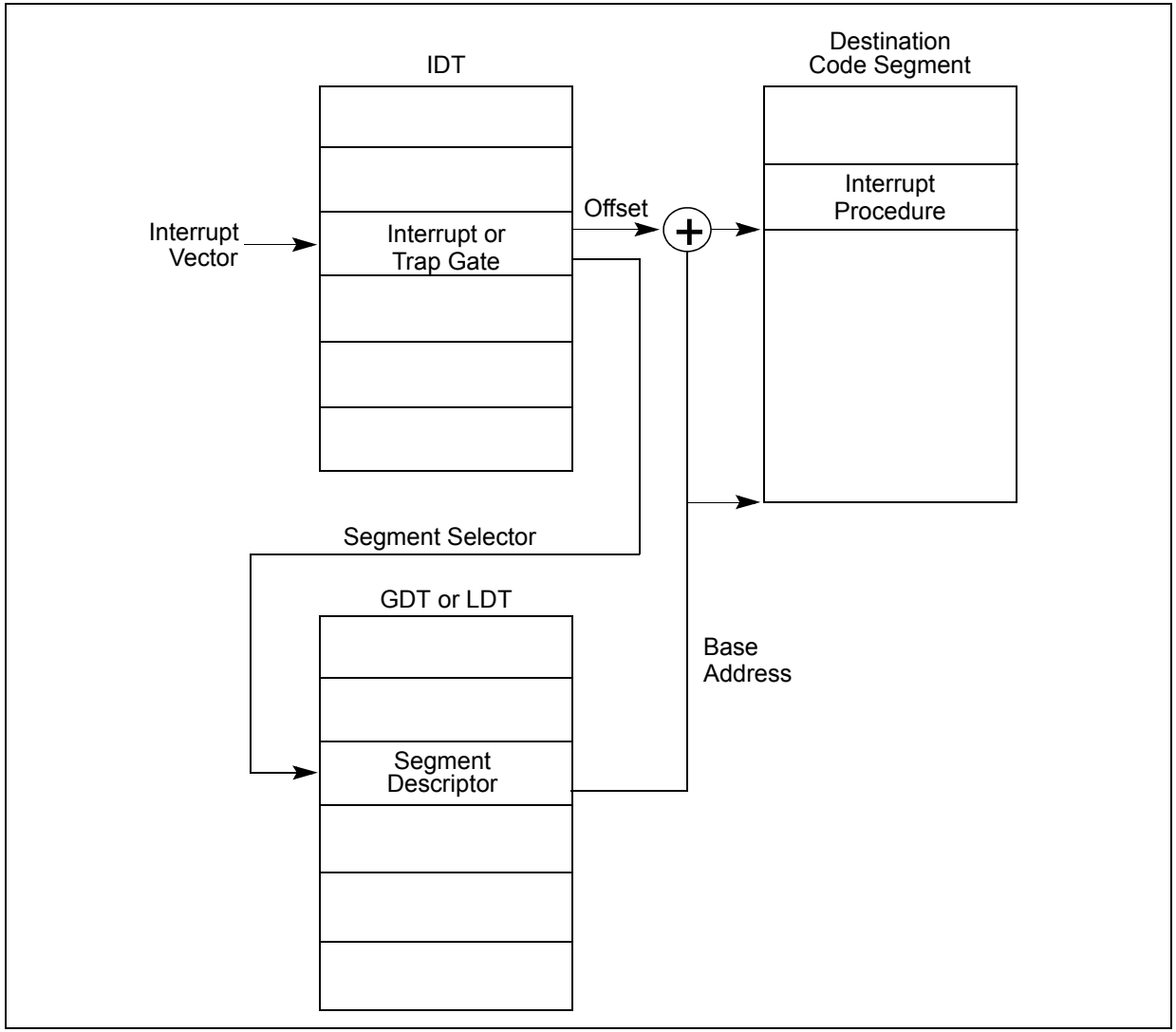


Figure 5-3. Interrupt Procedure Call

Interrupt Task Switch

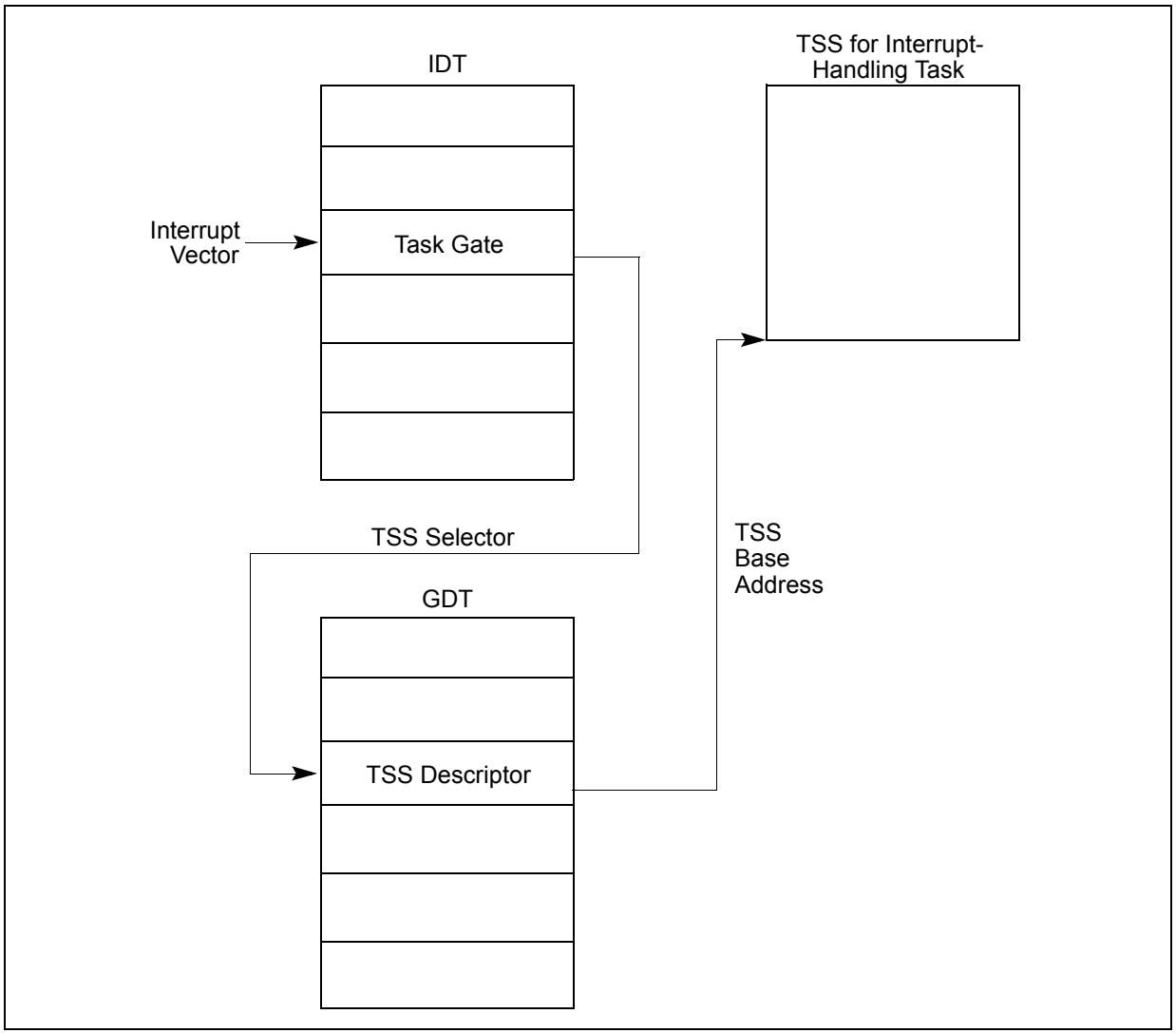


Figure 5-5. Interrupt Task Switch

Tipos de Interrupciones

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

Interrupt Table

Table 5-1. Protected-Mode Exceptions and Interrupts

Vector No.	Mnemonic	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.

Interrupt Table

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]

Table 5-1. Protected-Mode Exceptions and Interrupts (Contd.)

18	#MC	Machine Check	Abort	No	Error codes (if any) and source are model dependent.[4]
19	#XM	SIMD Floating-Point Exception	Fault	No	SSE/SSE2/SSE3 floating-point instructions[5]
20-31	—	Intel reserved. Do not use.			
32-255	—	User Defined (Non-reserved) Interrupts	Interrupt		External interrupt or INT *n* instruction.

NOTES:

1. The UD2 instruction was introduced in the Pentium Pro processor.
2. Processors after the Intel386 processor do not generate this exception.
3. This exception was introduced in the Intel486 processor.
4. This exception was introduced in the Pentium processor and enhanced in the P6 family processors.
5. This exception was introduced in the Pentium III processor.

Stack

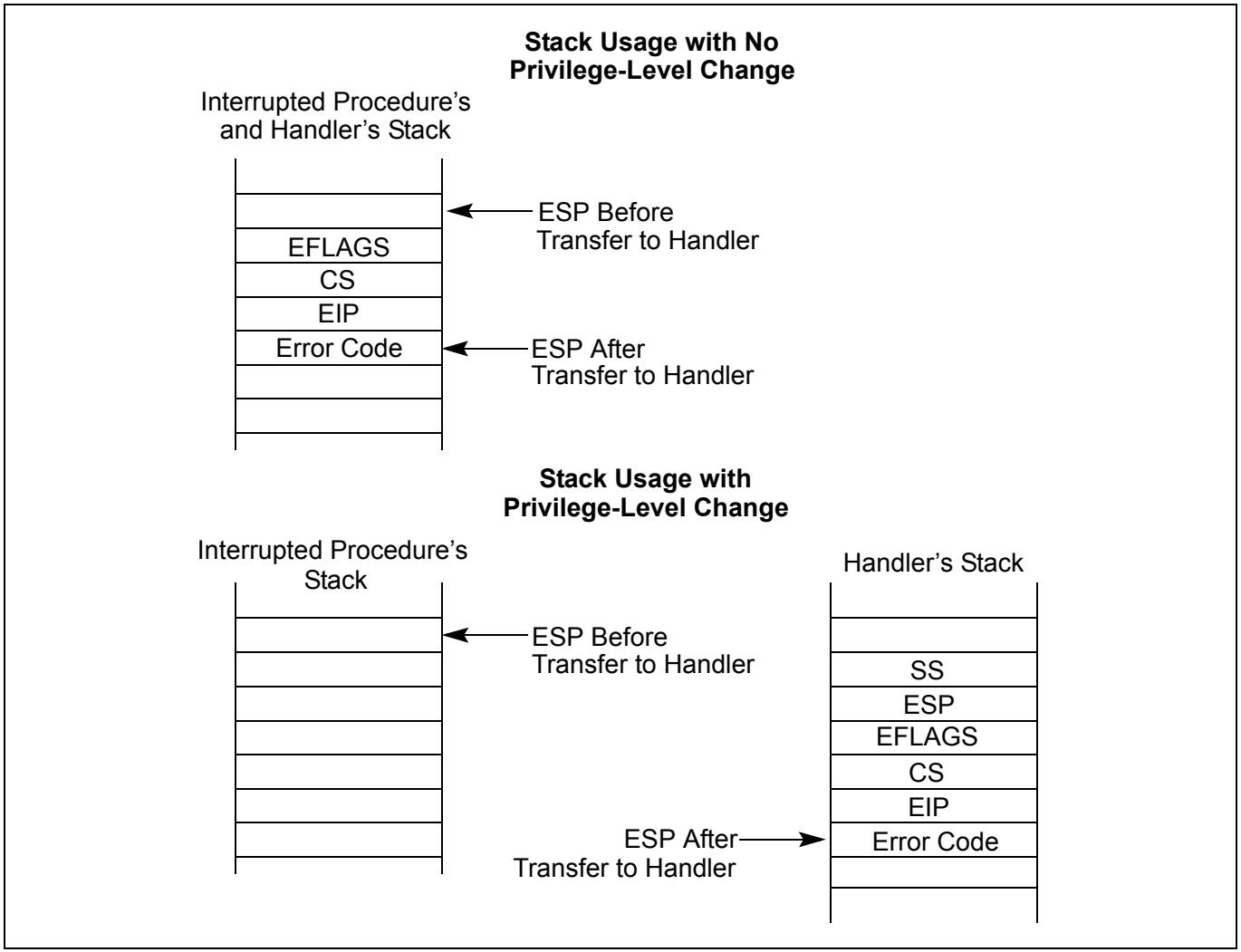


Figure 5-4. Stack Usage on Transfers to Interrupt and Exception-Handling Routines

Error Code

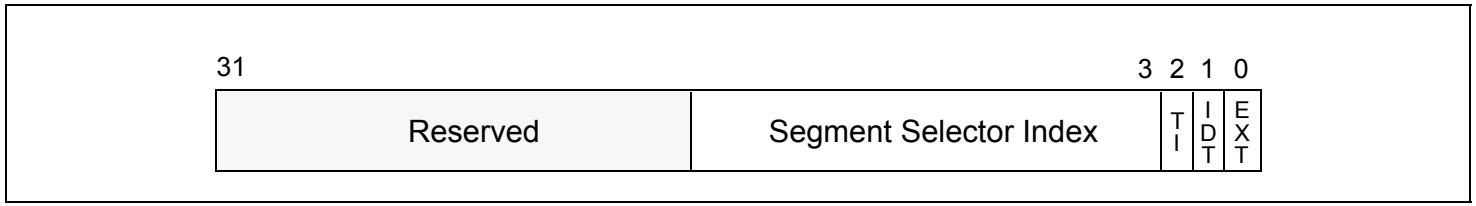


Figure 5-6. Error Code

- **EXT**: (External Event) Se setea para indicar que la excepcion ha sido causada por un evento externo al procesador
- **IDT**: (Descriptor Location) Cuando esta seteado indica que el campo Segment Selector Index se refiere a un descriptor de puerta en la IDT. Cuando esta 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) Tiene significado cuando el bit anterior esta en cero. Indica a que tabla de descriptores corresponde el selector del campo Indice. (GDT=0 , LDT=1)

Ayudas

1. `idt.h`: Descripcion de las estructuras.
2. `idt.c`: Estructura de la IDT con cada una de sus entradas
3. `isr.h`: Descripcion de las funciones handlers
4. `isr.asm`: Codigo de los handlers

- `IDT_ENTRY(numero)`: Permite declarar una entrada en la IDT, para la utilizar el handler de nombre `_isr`X
- `void inicializar_idt()`: Funcion llamada desde el kernel para inicializar las entradas en la idt

IDT

- Struct de descriptor de IDT

```
typedef struct str_idt_descriptor {
unsigned short idt_length;
unsigned int idt_addr;
} __attribute__((__packed__)) idt_descriptor;
```

- Struct de una entrada de la IDT

```
typedef struct str_idt_entry_fld {
unsigned short offset_0_15;
unsigned short segsel;
unsigned short attr;
unsigned short offset_16_31;
} __attribute__((__packed__, aligned (8))) idt_entry;
```

Gracias!!!

Preguntas?

Ahora a darle átomos al TP3!

Ref: Intel Software developer's manual (vol. 3)
capítulo 6, interrupciones (tan sólo 10 pags)