 UTN.BA <small>UNIVERSIDAD TECNOLÓGICA NACIONAL FACULTAD REGIONAL BUENOS AIRES</small>	Arquitectura de Computadores	Curso:	K1092
	UNIDAD 9	MICROPROCESADORES- INTERRUPCIONES	

TRABAJO PRÁCTICO - UNIDAD N° 9


MICROPROCESADORES - INTERRUPCIONES

GRUPO N° 5

Arquitectura de Computadoras

Comision K1092


ALUMNO	LEGAJO
BUONAMICO, Leandro Elían	1776721
DI NAPOLI, Franco Antú	2047901
PERALTA, Roque Damián	1447580
RODRIGUEZ, Iván Ángel	1782630

 UTN.BA <small>UNIVERSIDAD TECNOLÓGICA NACIONAL FACULTAD REGIONAL BUENOS AIRES</small>	Arquitectura de Computadores	Curso:	K1092
	UNIDAD 9	MICROPROCESADORES- INTERRUPCIONES	

TP Unidad 9: Microprocesadores – Interrupciones

b) Responder: (utilizando como bibliografía el Cáp. 14 de Angulo y las clases teóricas dictadas)

1. ¿Qué es una interrupción?
2. ¿Qué es una excepción?
3. ¿En base a que maneja las interrupciones y excepciones un Pentium? Breve concepto.
4. ¿Qué son las interrupciones externas?
5. ¿Cuál es la diferencia entre interrupción externa enmascarable e interrupción externa no enmascarable? De un ejemplo de cada caso.
6. Indique y explique los tipos de excepciones.
7. ¿Qué es una Tabla de Descriptores de Interrupción (IDT) y que contiene?
8. ¿Cómo está compuesta y cuál es la principal diferencia en la estructura IDT en Modo Real y en Modo Protegido?
9. ¿Para qué sirve el IDTR? Realice un esquema de cómo se vincula con la IDT
10. Describa brevemente los vectores de interrupción #0 #2 #3 #6 #7 #12 #14 #16 #19
11. ¿Qué es el código de error y qué información contiene?
12. Describa cómo opera el IDTR y el IDT al arrancar el Pentium en modo real.
13. Esquematice el acceso a las interrupciones y excepciones en modo real.
14. Describa brevemente cuáles son las fases de atención de una interrupción o excepción.

 UTN.BA <small>UNIVERSIDAD TECNOLÓGICA NACIONAL FACULTAD REGIONAL BUENOS AIRES</small>	Arquitectura de Computadores	Curso:	K1092
	UNIDAD 9	MICROPROCESADORES- INTERRUPCIONES	

1) Las **interrupciones** son acontecimientos que provocan un desvío en el flujo del control de la CPU, por ejemplo, averías en el hardware.

2) Las **excepciones** son acontecimientos que provocan un desvío en el flujo del control de la CPU, estas son internas y se producen como consecuencia de una anomalía dentro de la CPU (por ejemplo, una división por cero) durante la ejecución de un programa.

3) Para el manejo de interrupciones y excepciones, el modelo dispone de una tabla o vector de interrupciones llamada **tabla de descriptores de interrupciones y excepciones (IDT)**.

La IDT está formada por 256 entradas, cada uno de los diferentes tipos de interrupciones está asociado a una de esas entradas. La misma ocupa un segmento, cuya base y límite están contenidos en el Registro de Tablas de descriptores de Interrupciones (IDTR).

Tanto en modo protegido como en modo real se utiliza la IDT, aunque la forma de operar es distinta en cada modo, el objetivo es el mismo.


4) Cuando el microprocesador “recibe” una señal de interrupción “desde afuera”, deja la ejecución del programa actual y bifurca al servicio residente en la memoria principal. Estos servicios se denominan servicios de interrupción o de dispositivo, estas interrupciones se clasifican como **externas o hardware**.

Las interrupciones **externas** son convocadas en forma asincrónica, no dependen del programa en ejecución, estas pueden ser enmascarables o no enmascarables.

5) Una **interrupción no enmascarable (NMI)** causa que la CPU deje lo que está haciendo, cambie el puntero de instrucción para que apunte a una dirección particular y continúe ejecutando el código de esa dirección. Se diferencia de los otros tipos de interrupción en que los programadores no pueden hacer que la CPU las ignore. Están ligadas a tareas en el sistema que no se pueden retrasar como son las rutinas de bajada del sistema por corte de energía, chequeo de la integridad de memoria, etc. Un ejemplo de este tipo de interrupciones puede ser en los teléfonos móviles cuando se está terminando la batería, el sistema lanza una interrupción avisando al usuario del problema.

Las **interrupciones enmascarables (INTR)** se ejecutan a solicitud del periférico correspondiente siempre que en el registro de status (FLAGS REGISTER) del procesador las interrupciones estén habilitadas. En el caso de no estarlo la solicitud será ignorada y la CPU continuará con lo que estaba realizando. Un ejemplo de una interrupción enmascarable es cuando el usuario presiona una tecla en el teclado, el procesador no deja de atender a la tarea para procesar esta nueva acción, sino que la interrupción va a ser procesada a medida que se vaya liberando.

La diferencia más significativa entre estos tipos de interrupciones es que en una interrupción no enmascarable, el CPU deja de procesar la tarea que esté haciendo para atender y darle lugar a la interrupción, mientras que las enmascarables se atienden cuando el periférico lo requiera y hasta pueden ser ignoradas.

 UTN.BA <small>UNIVERSIDAD TECNOLÓGICA NACIONAL FACULTAD REGIONAL BUENOS AIRES</small>	Arquitectura de Computadores	Curso:	K1092
	UNIDAD 9	MICROPROCESADORES- INTERRUPCIONES	

6) Las excepciones son un tipo de interrupción sincrónica, típicamente causada por una condición de error en un programa, como por ejemplo una división entre 0 o un acceso inválido a memoria en un proceso de usuario.

Normalmente genera un cambio de contexto a modo supervisor para que el sistema operativo atienda el error.

Tipos de excepciones:

- **Faltas o errores.** Se pueden detectar y corregir antes de que se produzca la ejecución de una instrucción determinada.
- **Trampas.** Se detectan una vez ejecutada la instrucción que las provoca.
- **Abortos.** Se detectan sin localizar la instrucción que las provoca, produciendo que se aborte o cancele la ejecución del programa.

Cabe destacar que, cuando el sistema operativo detecta una excepción durante la ejecución de un programa, procede a terminar con el proceso en sí.

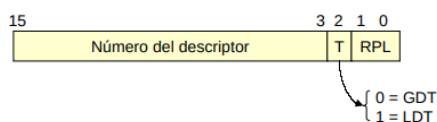
7) La tabla de descriptores de interrupción (IDT) asocia cada excepción o vector de interrupción con sus procedimientos de procesamiento. Similar a la tabla GDT, IDT también es una matriz compuesta de descriptores largos de 8 bytes. A diferencia de GDT, el primer elemento de la tabla puede contener descriptores. Para formar un valor de índice en la tabla IDT, el procesador multiplica el número de vector de excepción o de interrupción por 8. Debido a que hay como máximo 256 vectores de interrupción o excepción, IDT no necesita contener más de 256 descriptores. El IDT puede contener menos de 256 descriptores, porque solo las posibles excepciones o interrupciones requieren descriptores. Sin embargo, todas las entradas de descriptor vacías en IDT deben establecer sus bits de existencia (banderas) en 0. La tabla IDT puede residir en cualquier lugar del espacio de direcciones lineales, y el procesador usa el registro IDTR para ubicar la tabla IDT. Este registro contiene la dirección base de 32 bits y el valor de longitud de 16 bits (longitud límite) de la tabla IDT. La dirección base de la tabla IDT debe alinearse en un límite de 8 bytes para mejorar la eficiencia de acceso del procesador. El valor límite de longitud es la longitud de la tabla IDT en bytes.


TABLAS DE DESCRIPTORES

- ♦ Descriptores agrupados en tablas de descriptores
- ♦ Descriptor = 8 bytes
- ♦ Tablas limitadas a 64 Kbytes
- ♦ Total = 8192 descriptores por tabla

Tabla de descriptores	Registro
GDT (<i>Global Descriptor Table</i>)	GDTR
LDT (<i>Local Descriptor Table</i>)	LDTR
IDT (<i>Interrupt Descriptor Table</i>)	IDTR

Selector de segmento (CS,DS,ES,...)



 UTN.BA <small>UNIVERSIDAD TECNOLÓGICA NACIONAL FACULTAD REGIONAL BUENOS AIRES</small>	Arquitectura de Computadores	Curso:	K1092
	UNIDAD 9	MICROPROCESADORES- INTERRUPCIONES	

8) En modo protegido la tabla IDT está compuesta por entradas que contienen descriptores de puertas que permiten acceder al segmento donde se encuentra la rutina. El tamaño máx. de la IDT es 2KBS (2256 entradas x 8 bytes/ entrada).

Estos descriptores IDT responden a 3 tipos de puertas:

*** Puertas de tarea:**

En la entrada de la IDT hay un descriptor de una puerta de tarea (PT), que sirve para realizar una conmutación de tareas.

Ventaja: aislamiento entre tareas.


Desventaja: lentitud.

*** Puertas de interrupciones y excepciones:** Actúan de la misma forma que las puertas de llamadas. Se conmuta dentro de la misma tarea un segmento de código donde reside la rutina de interrupción. El bit "x" representa el tipo de puerta. En la primera fase de atención a la interrupción se detecta la interrupción o excepción y en la segunda fase se guardan los valores del estado actual en la pila para poder retornar al finalizar la interrupción.

IDT modo real :

Para el manejo de interrupciones y excepciones, el modelo dispone de una tabla o vector de interrupciones llamada tabla de descriptores de interrupciones y excepciones (IDT).

- La IDT está formada por 256 entradas, cada uno de los diferentes tipos de interrupciones está asociado a una de esas entradas.
- La IDT ocupa un segmento, cuya base y límite están contenidos en el Registro de Tablas de descriptores de Interrupciones (IDTR).
- Tanto en modo protegido como en modo real se utiliza la IDT, aunque la forma de operar es distinta en cada modo, el objetivo es el mismo.
- En la inicialización o reset del modelo (modo real), el SO carga la base del registro IDTR con el valor 0000 0000H , ya que solo se utiliza el primer MB de la memoria.
- Se trabaja con segmentos (no existe la paginación) y máximo tamaño del registro es de 64 KB.
- En cada entrada de la IDT, se apunta al segmento de código donde reside la rutina que atiende a la interrupción.
- El tamaño máximo de la IDT es de 1 KB, o sea, 256 entradas x 4 bytes/entrada, por lo tanto el límite es 03FFH

 UTN.BA <small>UNIVERSIDAD TECNOLÓGICA NACIONAL FACULTAD REGIONAL BUENOS AIRES</small>	Arquitectura de Computadores	Curso:	K1092
	UNIDAD 9	MICROPROCESADORES- INTERRUPCIONES	

9) IDTR: Almacena el valor de la base de la IDT y su límite, sus registros se usan cuando se produce una interrupción o excepción. Su tamaño alcanza los 2KB.

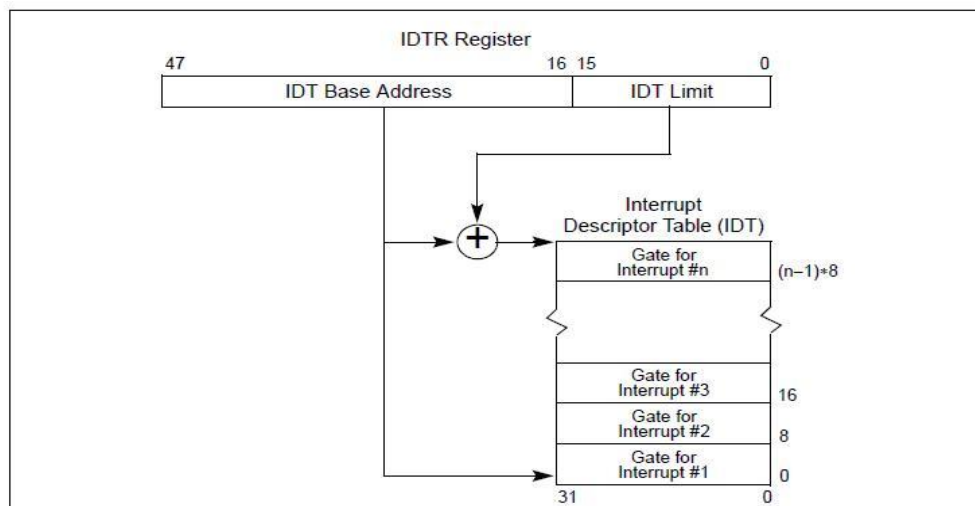


Figure 5-1. Relationship of the IDTR and IDT

10)

Vector 0:

Error de división (#DE): Lo generan DIV o IDIV cuando el divisor vale 0 o el resultado no se puede representar con los bits especificados.

Vector 2:

NMI: la provoca la patita NMI del procesador.

Vector 3:

Punto de ruptura (#BP): se ejecutó una INT 3, produciendo una trampa.

Vector 6:

Excepción por código de operación no válido (#UD): el procesador intentó ejecutar un código de operación no válido o reservado, o una interrupción con un tipo no válido de operando para el tipo de código, etc.

Vector 7:

Excepción por coprocesador no disponible (#NM): procesador no detecta coprocesador.


Vector 12:

Excepción por falta de pila: surge cuando ocurre algún problema con la pila.

Vector 14:

Excepción de falta de página: Indica que con la paginación habilitada, el procesador detecta una de las siguientes condiciones mientras se emplea el mecanismo de traducción de página para traducir la dirección lineal a física:

El señalizador P (presencia) en una dirección de página o en la entrada de la tabla de páginas accedida para la traducción de la dirección está puesto a 0 indicando que una tabla de páginas o la página que contiene

 UTN.BA <small>UNIVERSIDAD TECNOLÓGICA NACIONAL FACULTAD REGIONAL BUENOS AIRES</small>	Arquitectura de Computadores	Curso:	K1092
	UNIDAD 9	MICROPROCESADORES- INTERRUPCIONES	

el operando no está en memoria física.

El proceso no tiene suficiente privilegio para acceder a la página indicada.

El código de ejecución en modo usuario intenta escribir en una página de solo lectura.

Uno o más bits reservados en la entidad de directorio páginas son activados a 1.

11)

Cuando se produce una interrupción o excepción el procesador coloca en la cima de la pila del gestor de interrupciones un código de error. Mediante este código de error la cpu sabe cual es el origen de la interrupción o excepción y a donde tiene que acceder para tratarla (LDT, GDT o IDT). EXT es el bit que indica el origen del error, IDT indica que el error se produjo en la tabla IDT y TI es el que indica que el error desde GDT.

12)

El Pentium al arrancar trabaja por defecto en modo Real y el SO debe tener cargado en la memoria principal la tabla IDT. En la inicialización o Reset del Pentium, hay que ubicar la tabla de vectores de interrupción en el mismo sitio que el 8086 y para ello se carga la base del registro IDTR con el valor 0000 0000H.

En modo Real sólo se utiliza el primer MB de la memoria. Se trabaja con segmentos (no existe la paginación) y cada segmento como máximo tiene 64 KB.

La tabla IDT es referenciada mediante el IDTR (base y límite). En cada entrada se apunta al segmento de código donde reside la rutina que atiende a la interrupción. Cada vector es de 32 bits, y guarda los registros CS e IP. Añadiendo cuatro ceros (multiplicar por 16) a los 16 bits del CS se obtiene la dirección de la base del segmento, y con los 16 bits del IP se dispone del desplazamiento.

13)

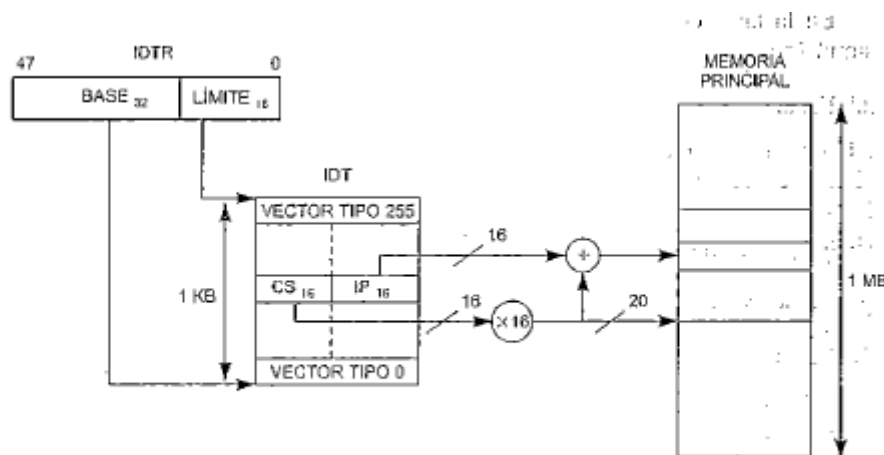



Figura 14.8. Acceso a las interrupciones y excepciones en modo Real.

 UTN.BA <small>UNIVERSIDAD TECNOLÓGICA NACIONAL FACULTAD REGIONAL BUENOS AIRES</small>	Arquitectura de Computadores	Curso:	K1092
	UNIDAD 9	MICROPROCESADORES- INTERRUPCIONES	

14)

- **Fase 1**

Se comprueba si hay interrupciones pendientes para ser atendidas. Si hay varias peticiones de interrupciones pendientes, se atenderán por orden de prioridad.

- **Fase 2**

Al comenzar la interrupción se **resguarda el entorno actual de la CPU** (contextswitch) para que al finalizar la interrupción o excepción se retorne al mismo punto, es decir, se guardan el contenido de los registros del procesador en la pila.

Por último $IF = 0$ prohíbe las interrupciones mascarables, evitando que una interrupción de menor importancia interrumpa a la rutina en ejecución. (ejemplo, periférico)

- **Fase 3**

Se busca la entrada correspondiente en la IDT y se cargan los registros CS e IP con el valor contenido en dicha entrada, de esta forma, se obtiene el inicio de la rutina y comienza la ejecución de la misma.

- **Fase 4**

La rutina de interrupción finaliza con la instrucción IRET, **se restaura el contexto de la CPU** (contextswitch). De esa manera, el programa continua su ejecución en la siguiente instrucción a la que se produjo la interrupción o excepción.

El $flagIF=1$, es decir, se vuelven a activar las interrupciones mascarables.