Unidad 8

MICROPROCESADORES Y MICROCONTROLADORES

Chip: Circuito electrónico diseñado sobre una estructura de silicio donde se incluyen millones de componentes electrónicos. Es de bajo costo y gran capacidad lógica.

TIPOS DE CHIP:

1) MICROCONTROLADORES

Los microcontroladores (µC), circuito integrado capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales. Incluye en su interior las tres unidades funcionales principales de una computadora: unidad central de procesamiento (CPU), memoria y periféricos de entrada y salida.

Principales características:

- Unidad de Procesamiento Central (CPU):
- Memoria de Programa
- Memoria de Datos
- Generador del Reloj
- Interfaz de Entrada/Salida

Proceso de Desarrollo:

- Desarrollo de software: Esta etapa corresponde a la escritura y compilación/ensamblaje del programa que rige las acciones del µC y los sistemas periféricos conectados a este.
- Programación del µC: el código de máquina correspondiente al programa desarrollado en la etapa anterior se descarga en la memoria del µC.
- Prueba y verificación: Por último, el µC debe conectarse al circuito base y someterse a pruebas para verificar el funcionamiento correcto del programa.

La motherboard → es el sustento físico del microprocesador, además de ser el elemento de comunicación entre los distintos componentes que se encuentran en el entorno del procesador.

2) MICROPROCESADORES µP

circuito integrado formado por transistores. Su arquitectura interna contiene la unidad central de proceso (CPU).

Casi todos los microprocesadores contienen como mínimo lo siguiente:

- UAL
- Varios registros
- Contador de programa
- Circuitería de decodificación de instrucciones
- Sección de control y temporización
- Cerrojos y buffers de datos
- Líneas de control y buses internos

- Varias entradas y salidas de control

SE PUEDEN DIFERENCIAR DIVERSAS PARTES:

- El encapsulado: es lo que rodea a la oblea de silicio en sí, para darle consistencia, impedir su deterioro y permitir el enlace con los conectores externos
- La memoria caché: es una memoria ultrarrápida que emplea el µP para tener a mano ciertos datos que predeciblemente serán utilizados en las siguientes operaciones sin tener que acudir a la memoria RAM reduciendo el tiempo de espera.
- Coprocesador Matemático: Es la parte del micro especializada en esa clase de cálculos matemáticos.
- Los registros: memoria pequeña con fines especiales que el µP tiene disponible para algunos usos particulares. En total son 32 registros.
- La memoria: lugar donde el procesador encuentra sus instrucciones de programa y sus datos.
- Puertos: es la manera en que el µP se comunica con el mundo externo.

Las funciones de la mayoría de las unidades de un microprocesador son:

- Registro de instrucción: Es un registro que contiene la primer combinación binaria de una instrucción
- Decodificador de instrucciones: Esta unidad interpreta el contenido del registro de instrucción, determina el microprograma exacto que se debe seguir para ejecutar la instrucción completa y dirige adecuadamente la sección de control
- UAL: Realiza las operaciones aritméticas, lógicas y de desplazamiento.
- Acumulador: Es un registro de propósito general asociado a las operaciones de la ALU
- Contador de programa: Es un área de almacenamiento de bits que siempre apunta a la siguiente instrucción que se va a ejecutar.
- Unidad de control y temporización: Recibe señales del decodificador de instrucciones para determinar la naturaleza de la instrucción que se va a ejecutar. Luego, las señales de temporización y control son enviadas a todo el microprocesador para coordinar la ejecución de las instrucciones
- Registro de estado: Contiene señalizadores de cero y arrastre
- Reloj interno: Genera una señal de reloj para utilizarla en el interior de la MPU que sirven para sincronizar acciones en el sistema completo.

Algunas de las características principales compartidas por casi todos los microprocesadores:

- Conexiones de alimentación
- Tamaño en bits (Longitud de palabra): Los microprocesadores se clasifican normalmente en unidades de 4, 8, 16, 32 o 64 bits.
- Líneas de datos: Los microprocesadores transfieren datos e instrucciones entre la MPU y memoria vía un bus de datos bidireccional.
- Líneas de dirección: Los microprocesadores utilizan buses de dirección de n líneas, a través de los cuales pueden direccionar 2n bits de memoria

- Líneas de control: La mayoría de los microprocesadores se caracterizan porque tienen todas o algunas de las siguientes líneas de control:
  o Líneas de reloj
  o Líneas de lectura/escritura o Líneas de entrada/salida
  o Líneas de interrupción
  o Líneas de re inicialización
  o Líneas de control del bus
  o Líneas de status del ciclo
- Registros internos
- Modos de direccionamiento: técnica utilizada para buscar el operando deseado durante la ejecución de una instrucción.

Arquitectura abierta vs Arquitectura cerrada

▪ El microprocesador tiene una arquitectura abierta: a) porque el computador que implementa es configurable por el usuario y puede realizar diversas tareas; b) puede construirse un computador con las características que deseen porque el sistema tiene a disposición los buses (datos, control y dirección) de forma que su configuración será variable de acuerdo con la aplicación a la que se destine.

▪ El microcontrolador tiene una arquitectura cerrada: a) porque el computador que implementa no es configurable por el usuario; b) es un sistema cerrado que contiene un computador completo y se destina a realizar una sola tarea ya que sus prestaciones son limitadas y no se pueden modificar; c) los microcontroladores suelen ir incrustados o embebidos en el propio dispositivo que gobierna (ejemplo: microcontroladores en electrodomésticos)

Concepto de familia

ALTAIR – INTEL- SIEMENS- MOTOROLA- MICROCHIP.

▪ Una familia de microprocesadores es un conjunto de modelos ligados por algunas características comunes.

▪ Familia x86 → los procesadores de esta familia son de Intel; su arquitectura responde al nombre IA-32.

▪ El núcleo de esta arquitectura es común para todos los microprocesadores y cada modelo agrega extensiones y recursos a dicho modelo.

▪ La familia x86 esta compuesta por seis generaciones. Dentro de cada generación hay diferentes modelos que varían en la relación de prestaciones y el consumo. Cada modelo esta orientado a cubrir el sector de mercado donde ese factor sea critico. Todos son compatibles entre sí (leer del libro).

Arquitecturas de diseño de computadores

▪ CISC: Complex Instruction Set Computer, las instrucciones son numerosas complejas y largas, necesitando múltiples ciclos de reloj para su ejecución. Una instrucción compleja → varias microinstrucciones, ejemplo: familia x86, motorola, AMD, etc.

▪ RISC: Reduced Instruction Set Computer; la idea es que cada instrucción tenga la mayor cantidad de microoperaciones solapadas posible, de modo que la mayoría de ellas se ejecuten en un ciclo de instrucción. Todas las instrucciones tienen el mismo tamaño, ejemplo: PowerPC, ARM, SPARC, etc.

▪ EPIC: Explicitly Parallel Instruction Computing o computación de instrucciones paralelas explícitas; su característica más importante es el permitir agrupar instrucciones para ejecutarlas de forma paralela exponencialmente; ejemplo: usado por HP e Intel en arquitecturas de 64 bits.

Longitud de palabra

La palabra del microprocesador está asociada a la cantidad de bits que un microprocesador "procesa" simultáneamente como grupo. El procesamiento involucra la obtención de memoria, su operación y el almacenamiento del resultado de nuevo en memoria. En la actualidad muchos de los microprocesadores son de palabra de 64 bits. Palabra también puede utilizarse para hablar de estructuras de datos y para expresar la cantidad de bits a los que se puede acceder por vez en memoria (palabra de memoria).

Palabra de memoria: Cantidad de bits a los que se puede acceder por vez.

Palabra: grupo de 16 bits

Capacidad de direccionamiento

▪ Tiene relación con el acceso a las líneas que transfieren direcciones físicas a memoria (mm) .

▪ Cada dirección de memoria permite individualizar un bloque físico de la misma, donde se va a leer o escribir.

▪ Las líneas de acceso a la memoria constituyen el bus de direcciones

▪ Con n bits, se puede direccionar 2 n direcciones de memoria distintas.

Número de instrucciones

La cantidad de instrucciones que un microprocesador puede decodificar y por lo tanto ejecutar, implica su mejor capacidad para hacer cosas distintas. Cada instrucción tiene un número predeterminado de microoperaciones y si se las agrupa como ladrillos o bloques de construcción de un programa, serán consideradas paquetes de microoperaciones.

Número de registros internos

Se refiere a la cantidad de registros con la que cuenta el microprocesador, cuya función es brindar almacenamiento temporal durante la ejecución. Existen dos tipos de registros:

Los llamados registros para el programador de aplicaciones o registros visibles que pueden ser actualizados por las aplicaciones.

Los llamados registros para el programador de sistemas o registros invisibles que solo pueden ser accedidos por el Sistema operativo.

Pipeline: técnica que consiste en dividir el procesamiento de cada instrucción en etapas y que estas operen en paralelo

El SW compilado para sistemas de 16 bit sólo “visualizará” los registros de 16 bits aun cuando la arquitectura sea de 32.

El x86 contaba con dos unidades funcionales:

- Unidad de ejecución (EU): decodificaba y ejecutaba las instrucciones
- Unidad de interfaz (BIU): Mediante su bus, se encargaba de obtener y almacenar los datos en memoria principal y también de obtener las instrucciones del programa

Registros de cálculo de 16 bits (del IA-16)

15 87 0

AH	AL	Ax Registro Acumulador (operación de E/S y de cadena)
BH	BL	BX Registro Base (registro base para direccionamiento)
CH	CL	Cx Registro Contador (para bucles, iteraciones, desplazamientos y rotaciones)
DH	DL	Dx Registro para datos (almacenado de datos, direcciones de punteros, extensión de Ax en multiplicación y división)

Todos los registros se pueden utilizar en operaciones de cálculo aritmético o lógico

Es factible emplearlos con instrucciones que los afecten en su totalidad, AX, o que afecten la parte alta, AH, o la baja, AL.

REGISTROS DE PROPOSITO GENERAL

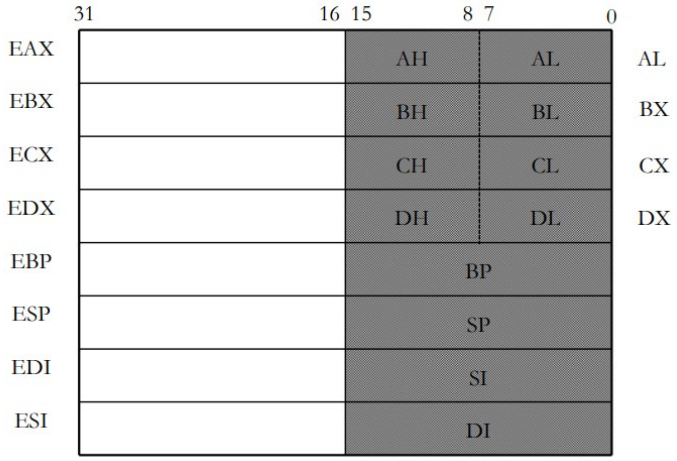
Son los mismos registros que tenía el 8086 de 16 bits pero ampliados a 32 bits. Este grupo consta de ocho registros capaces de trabajar con información de 32 bits cuando utilizan todo su tamaño

Los nombres de cada uno de estos ocho registros son:

- EAX: Acumulador
- EBX: Base
- ECX: Contador
- EDX: Datos

- ESP: Puntero de pila
- EBP: Puntero de base (Base Pointer)
- ESI: Índice fuente (Snack Pointer)
- EDI: Índice destino

Cuando se hace referencia a uno de los registros que trabaja con 32 bits, se pone la letra “E” de extendido precediendo al nombre habitual del registro de 16 bits.



Los registros EAX, EBX, ECX y EDX, se emplean fundamentalmente en operaciones generales como las lógicas. Por lo general el registro EAX, muy utilizado en los procesadores Intel, se emplea como Acumulador en instrucciones lógico-aritméticas

El registro EBX contiene la base de la dirección donde empieza una estructura datos

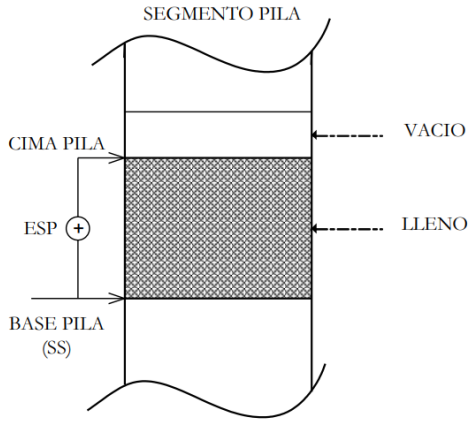
El registro ECX es habitual utilizarlo como contador en instrucciones que se repiten distintas veces.

El registro EDX se utiliza como apoyo al EAX. También para operaciones de E/S, posiciones de E/S que van periféricos o vienen de periféricos.

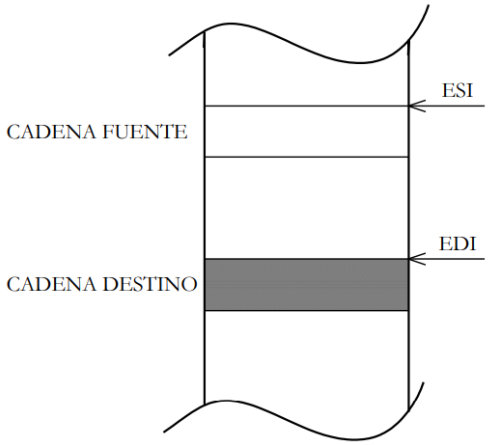
Los registros apuntadores ESP y EBP, sirven para controlar el direccionamiento de la pila y almacenar desplazamientos relativos a la pila en curso.

Las operaciones en la pila las soportan tres registros diferentes:

- Registro de segmento de pila (SS). Especifica las características del segmento de pila que reside en memoria. Una pila puede tener hasta 4 GBytes de longitud que es el máximo tamaño de un segmento.
- Registro puntero de pila (ESP). Contiene el desplazamiento de la cima de la pila en el segmento de la pila actual. Lo usan las operaciones PUSH y POP
- Registro puntero base de la pila (EBP). Se usa normalmente par a acceder a estructuras de datos pasadas en la pila



Por último, los registros ESI y EDI, contienen valores índice usados en la exploración de grandes conjuntos de datos y admiten la posibilidad de incremento y decremento automático de su valor para relaciones fuente y destino



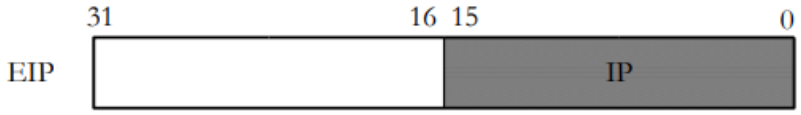
REGISTROS PUNTEROS

Son registros utilizados para desplazarse dentro de un bloque o zona de memoria. Si el bloque es de código "IP", si es un bloque de pila "SP" y "BP" y si es un bloque de dato "SI"

- IP: Registro puntero de instrucción
- SP: Registro puntero de pila
- BP: Registro base para la pila
- SI: Registro índice fuente
- DI: Registro índice destino

EIP: REGISTRO PUNTERO DE INSTUCCION

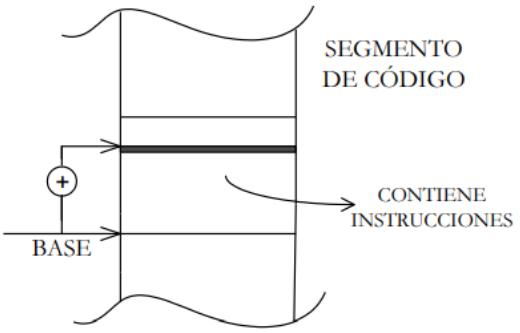
registro de 32 bits que almacena el desplazamiento que hay que añadir a la base del segmento de código para obtener la dirección donde está la siguiente instrucción.



Puede trabajar en dos modos:

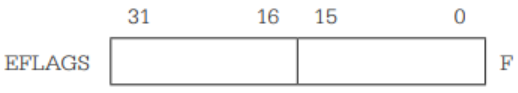
- Modo nativo: recibe el nombre de EIP y posee 32 bits
- Modo real: sólo se precisan de 16 bits para especificar el desplazamiento, que son los dos bytes de menos peso de EIP, y que se denominan IP.

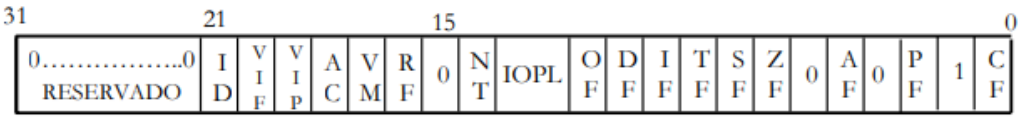
En la memoria segmentada la dirección de la instrucción en curso se halla sumando el desplazamiento a la base donde comienza el segmento del código



REGISTRO DE ESTADO

consta de 32 bits de los cuales la mayoría son señalizadotes de estado,actuando los restantes como señalizadotes del sistema. (En este registro se alojan, por nombrar algunas, todas las banderas aritméticas, banderas de modo de trabajo del microprocesador, banderas asociadas a interrupciones, etc)





Bits que forman parte del registro EFLAGS:

1) CF: Señalizador de acarreo en el bit más significativo.
2) PF: Bit de paridad impar
3) AF: Señalizador de acarreo auxiliar
4) ZF: Señalizador de cero
5) SF: Señalizador de signo
6) TF: Excepción al terminar la ejecución de la instrucción
7) IF: Flag de habilitación de interrupciones
8) DF: Flag de dirección de explotación de las cadenas de caracteres
9) OF: Flag de overflow
10) IOPL: Nivel de privilegio de las entradas y salidas (input/output)
11) NT: Tarea anidada
12) RF: Flag de reanudación.
13) VM: Modo virtual.
14) AC: Bit de chequeo de alineamiento.
15) VIP: Interrupción virtual pendiente.
16) VIF: Interrupción virtual. El procesador sólo reconoce el flan VIF
17) ID: Bit de identificación

REGISTRO DE SEGMENTO

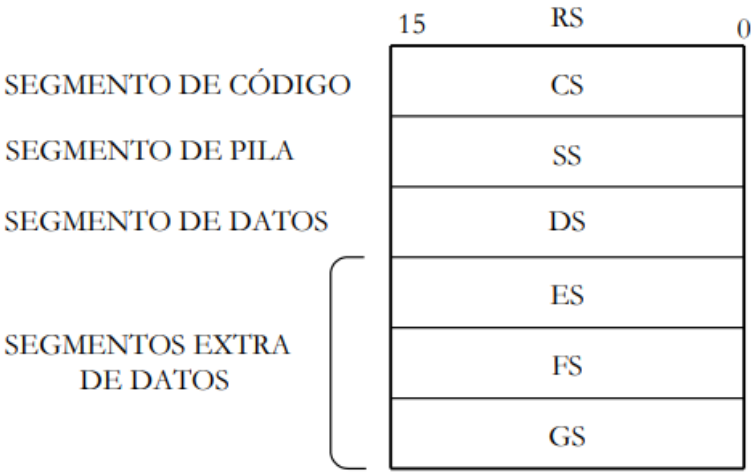
Son registros concebidos para brindar soporte a aquellos sistemas operativos que administran la memoria como una agrupación de segmentos. Almacenan la referencia binaria a la base de un segmento en memoria. Es decir, bloques de tamaño ajustado al objeto que contienen.

Cuál es la función y para que se utiliza un Segmento:

Los segmentos son zonas de la memoria de tamaño variable que contienen el mismo tipo de información.

El registro de segmento cumple la función de almacenar la referencia binaria a la base de un segmento en memoria, se utiliza para crear un programa en Assembler en el que se divide lógicamente en 3 partes: Código, Datos y Pila.

Objeto: código de un programa, los datos que utiliza (variables locales + globales) y su pila.



Relación entre los registros y el modo de direccionamiento de datos

La siguiente formula muestra la menor cantidad de modos de direccionamiento indicada en los manuales del microprocesador INTEL de 16 bits:

direc. de segmento+direc. base+índice+desplazamiento

La base de un segmento se encuentra en los registros de segmento DS, ES o SS y se multiplica por 16 antes de sumarse al resto de los argumentos enunciados en la fórmula.

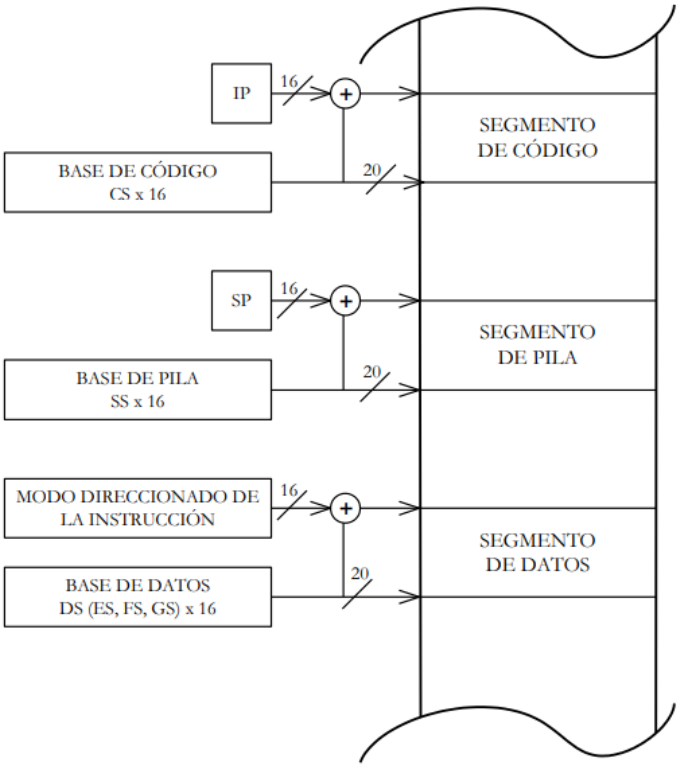
El valor de una base para direccionamiento base “dentro” del segmento se almacena en cualquiera de los registros BX o BP. el valor índice que recorre una estructura de dato se encuentra en uno de los registros índice SI o DI y el desplazamiento puede ser una constante de 8, de 16
La base y el índice son valores que se pueden modificar durante el procesamiento pero el desplazamiento no porque es una constante

Velocidad de microprocesador

Cada microprocesador tiene su propio reloj interno, cuya frecuencia indica con qué velocidad puede procesar bits; ésta se expresa en Hz

MODO REAL

Un segmento queda definido por dos elementos. El primero es la base de 20 bits y el segundo es el desplazamiento de 16 bits.



En modo real todo segmento de código, datos o pila está especificado por una dirección lógica, compuesta por dos campos de 16 bits cada uno. Estos son:

- El Selector: referencia la base del segmento, esto se deduce a partir del valor contenido en el registro de segmento apropiado. Como el 8086 sólo maneja una memoria de 1 MByte de capacidad máxima, para obtener la base del segmento se multiplica dicho valor binario por 16.
- El Desplazamiento: solo se necesitan 16 bits para expresar el desplazamiento que hay que añadir a la base ya que el tamaño máximo del segmento en 8086 es de 64 KBytes.

Por tanto una dirección quedaría: Dirección efectiva = RS x 16 + Desplazamiento.

DESPLAZAMIENTO EN MODO PROTEGIDO

(multitarea), un segmento queda caracterizado por tres parámetros fundamentales:

- la base: direccion lineal donde comienza el segmento. Esta fomada por 32 bits (2^{32} = 4 GBytes).
- el limite: consta de 20 bits que determinan con exactitud el tamaño del segmento usado por el programador y en el que residen informaciones válidas.
- los atributos: campo de 12 bits que proporciona las características relevantes del segmento.

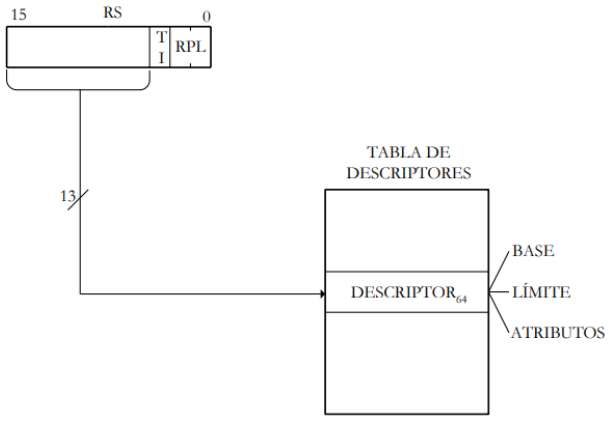
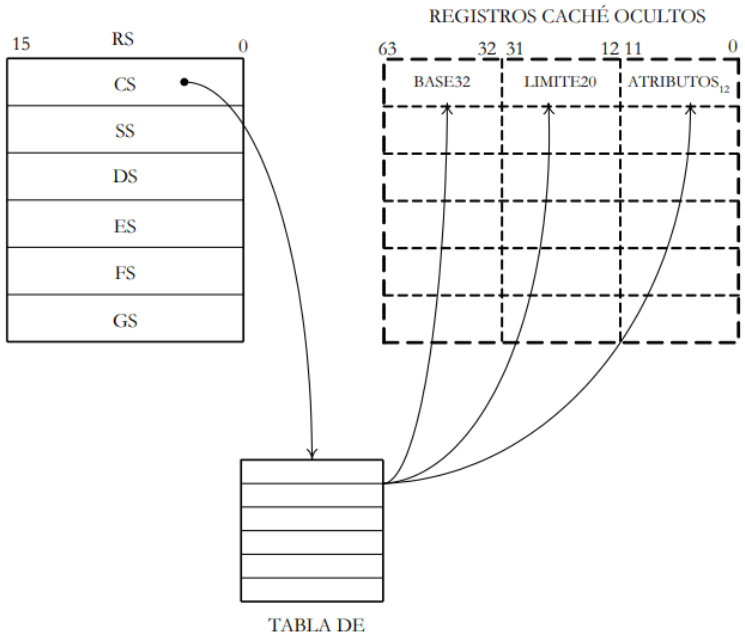


Figura 7.14 – Se accede a la tabla de descriptores consultado el selector y comprobando el índice de tabla.

Descriptor de segmento: (64 bits) conjunto de base, limite y atributo

En modo protegido para obtener el descriptor de segmento, el Pentium utiliza el valor de registro de segmento, para acceder a unas tablas residentes en memoria principal. Como necesita disponer directamente de estas características, a cada segmento se le asocia un registro caché ultrarrápido.
Cuando se carga un registro segmento, la CPU busca automáticamente en la tabla de descriptores residentes en memoria principal, la base, el limite y los atributos del segmento referenciado y los carga en el registro caché asociado. A partir de aquí, todo acceso a dicho segmento, se realizará utilizando los datos que se encuentran en el registro caché. Cuando se modifica el valor de algún registro de segmento, surge una penalización en el tiempo por eso hay cuatro registros de datos.
Cuando se modifica el valor de algún registro de segmento, surge una penalización en el tiempo ya que se tiene que actualizar la caché. Por eso hay cuatro registros de datos (ES, DS, FS, y GS) ya que lo normal es cambiar los segmentos de datos y no los de código

En modo real, también se usan los registros caché invisibles, siendo el valor de la base el del registro de segmento con cuatro ceros añadidos y todos los derechos de acceso están permitidos.



JUEGO DE REGISTROS DE LA UNIDAD EN COMA FLOTANTE (¿)

Familia

Una familia de microprocesadores es un conjunto de modelos ligados por algunas características comunes.

Familia X86

Los procesadores de esta familia son del fabricante Intel, y su arquitectura responde al nombre IA32. El juego de instrucciones de esta arquitectura es de tipo CISC (repertorio amplio de instrucciones),con lo que cada instrucción es de tipo complejo y se ejecuta en varios ciclos de reloj.

Generaciones

Son 6. Todos son compatibles entre sí.

Primera generación: 8086 y 8088

(1978-79) los procesadores 8086 y 8088. Implementan el pipe line o solapamiento de procesos. Representa una mejora en la cantidad de instrucciones que se procesan por segundo. El pipe line permite procesar simultáneamente diversas etapas de distintas instrucciones, completándose en cada etapa una parte de la ejecución de cada instrucción. El 8088 se diseña con un bus de 8 bits en vez de 16 bits como el que tiene el 8086. Para resolver esta diferencia Intel divide cada procesador 8088 y 8086 en dos Sub-procesadores:

- Unidad de Ejecución (EU): realiza todas las operaciones.
- Unidad de Interfaz con el Bus (BIU): accede a datos e instrucciones del mundo exterior.

En ambos procesadores las Unidades de Ejecución son idénticas pero varia la Unidad de Interfaz con el bus. Así se consiguió un ahorro de esfuerzo para producir el 8088

Estos procesadores tienen 27 modos de direccionamiento para localizar un operando de una instrucción.

Las principales aportaciones son:

- Gestión de memoria
- Segmentación de 2 etapas: Buscar instrucción y ejecutar.
- Interrupciones sectorizadas multinivel.

La segunda generación: 80286

1982. Se caracterizan por poseer dos modos de funcionamiento completamente diferenciados:

- Modo Real: Se comporta igual que un 8086 pero con mayor velocidad, al ser conectado a la alimentación arranca en este modo.
- Modo Protegido: funciona con capacidad de proceso multitarea y memoria virtual. Este modo es propio del 286, por lo que pierde la compatibilidad con los procesadores anteriores. Cuando la CPU esta en modo protegido, los programas de usuario tienen un acceso limitado al juego de instrucciones; solo el proceso supervisor esta capacitado para realizar ciertas tareas. Así se evitan posibles conflictos entre los distintos programas de usuario, con lo que el fallo de un proceso no afecta al resto.

Las principales aportaciones son:

- Memoria Virtual hasta 1 GB y Memoria Física hasta 16 MB.
- Admitía multitarea e introdujo sistemas de protección.
- Cuatro niveles de privilegio.
- Aumenta la segmentación a cuatro capas.

Tercera generación: 80386

Es el primer procesador de 32 bits del mundo.Tiene tres modos posibles de funcionamiento:

- Modo Real: Compatible con el 8086.
- Modo Protegido: propio
- Modo Virtual 86: puede emular el funcionamiento simultáneo de varios 8086.

El sistema de paginación es transparente a la segmentación y permite el manejo de direcciones físicas. Cada segmento se divide en una o más páginas de cuatro kilobytes. La unidad de segmentación provee cuatro niveles de protección para aislar y proteger aplicaciones y el sistema operativo

Para facilitar diseños de hardware de alto rendimiento, la interfaz con el bus (BIU) del 80386 ofrece pipelining de direcciones, tamaño dinámico del ancho del bus de datos (16 o 32 bits) y señales de habilitación de bytes por cada byte del bus de datos.

Cuarta generación: 80486

El 80486 es una versión mejorada del 80386 que además tiene integrada una cache de 8 Kbytes y un coprocesador matemático 80387, con lo que se consigue que casi la mitad de las

instrucciones del 486 se ejecuten en un periodo de reloj en vez de los dos periodos que requiere el 386.

Los principales bloques que están en el interior de un procesador 486 y sus funciones son:

- Los registros de direcciones (RDI) y de datos (RDA), pertenecientes a la unidad de interconexión con el bus (BIU) encargada de la comunicación con el exterior a través de las 32 lineas de datos y 32 lineas de direcciones del bus.
- La unidad de caché de 8KB guarda las instrucciones y datos que seguramente serán requeridos próximamente. A través de un bus de 128 líneas se pueden leer del caché 16 bytes (128/8) que pasan a un buffer de la unidad de pre-carga de instrucciones.
- La unidad de pre-carga proporciona las direcciones de las próximas instrucciones al ejecutar y guarda las mismas en orden en dos buffers de 16 bytes, para que luego cada una sea decodificada.
- La unidad de decodificación realiza dos decodificaciones de cada instrucción.
- La unidad de control mediante líneas que salen de ella activa las operaciones que con cada pulso de reloj deben realizar los distintos bloques de la UCP (U de pre-carga, decodificadora, UAL, UPF, UC) conforme lo establecen micrócodigos de la ROM de control
- La unidad de manejo de memoria (MMU) se encarga de proporcionar las direcciones físicas de memoria que utiliza un programa y de la protección contra escrituras no permitidas en zonas reservadas de memoria.

Todas estas unidades participan del pipe line de instrucciones que en el 486 consta de 5 etapas:

1. Pre-carga: Consiste en la llegada de los códigos de las próximas instrucciones que entrarán al pipe-line a dos buffers de la unidad de pre-carga, para formar una cola.

2. Primera decodificación: A la unidad de decodificación llegan los primeros 3 bytes de cada instrucción, para separar su código de operación del número que hace referencia la dirección del dato.

3. Segunda decodificación: El código de operación es decodificado, lo que permite determinar la secuencia del micrócodigos contenida en la ROM de control. La UC generará las señales de control que enviará por líneas que salen de ella, para que cada unidad que controla, ejecute una parte de la instrucción con cada pulso reloj.

4. Ejecución de cada instrucción en la unidad correspondiente.

5. Almacenamiento de resultados: llega la primer instrucción completándose su ejecución. En caso de generar datos, estos se almacenan en los registros correspondientes. En este momento la instrucción 2 (I2), ingresa en etapa de ejecución, I3 entra a la segunda decodificación y los bytes de I4 son sometidos a la primera.

Quinta generación: Pentium

Las ventajas que aporta son:

- Supersegmentación con 14 etapas. Técnicas de predicción de saltos condicionales para evitar introducir demasiadas burbujas.

- Arquitectura superescalar. Dos cauces de datos, en un ciclo se ejecutan más de una instrucción.
- Aumento de cache. Caché de primer nivel (L1) y cache de segundo nivel (L2). La caché L1 se divide en dos partes independientes para datos para instrucciones de 8 KB cada una,con lo que es posible acceder a un dato y una instrucción en paralelo.
- El chip se empaqueta en formato PGA (Pin Gris Array) de 273 pines.

Las 7 unidades funcionales que aportan características específicas e innovadoras al procesador consiguiendo altas prestaciones, compatibilidad y mantenimiento de la integridad de los datos son:

- Unidad de enteros Superescalar: consiste en dos unidades de enteros de 32 bits que operan en paralelo.
- Unidades de memoria caché: están subdivididas en dos memorias caché independientes, una para datos y otra para instrucciones.
- Unidad de interconexión con el bus: presenta un bus de datos de 64 bits con lo que se obtiene una velocidad de transferencia de 538 MB/s.
- Monitor de prestaciones: consta de una serie de controladores internos y unidades de rastreo para evitar que se pierda gran cantidad de tiempo en ciertas rutinas o secciones de código.
- Unidad de redundancia funcional: consiste en una serie de técnicas para asegurar la integridad de los datos.
- Unidad de predicción de bifurcaciones: consta de una caché específica encargada de hacer una predicción dinámica de los saltos condicionales.
- Unidad de coma flotante: ha sido mejorada respecto del 486 incorporando un cauce segmentado de instrucciones de 8 etapas.
- Vías de acceso múltiple: proporcionan una arquitectura superescalar que tiene la habilidad de ejecutar más de una instrucción por cada ciclo de reloj

PROCESADOR	AÑO DE PRESENTACION	VELOCIDAD DE RELOJ	ARQUITECTURA	NRO. DE TRANSISTORES	DESCRIPCION
4004	1971	108 KHz	4 bits	2300	Primer chip con manipulación aritmética
8008	1972	108 KHz	8 bits	3500	Manipulación Datos/texto
8080	1974	2 MHz	8 bits	6000	10 veces las prestaciones del 8008
Intel 386 DX	1985	16 MHz a 33 MHz	32 bits	275 mil	Primer chip capaz de manejar juegos de datos de 32 bits
Pentium II	1997	233 MHz a 300 MHz	64 bits	7.5 millones	Doble Bus, Ejecución Dinámica
Intel Core Duo	2006	1,06 GHz a 2,50 GHz	32 bits	151 millones	Sexta generación de microprocesadores
Intel Core i7	2008	2,66 GHz a 4,2 GHz	64 bits	1.170 millones	Septima generación de microprocesadores

ARQUITECTURA DEL PENTIUM (CAPITULO 6)

El microprocesador Pentium sigue teniendo una arquitectura interna IA-32 de 32 bits, similar a la de sus predecesores 80386 y 80486.

Evolución de las primeras versiones del Pentium:

PROCESADOR	FRECUENCIA	TECNOLOGÍA	VOLTAJE	BUS	MULTIPLICADOR
P60	60 MHz	0,8 µ	5 V	60 MHz	-
P66	66 MHz	0,8 µ	5 V	66 MHz	-
P75	75 MHz	0,6 µ	3,52 V	50 MHz	1,5
P90	90 MHz	0,6 µ	3,52 V	60 MHz	1,5
P100	100 MHz	0,6 µ	3,52 V	66 MHz	1,5
P120	120 MHz	0,35 µ	3,52 V	60 MHz	2
P133	133 MHz	0,35 µ	3,52 V	66 MHz	2
P150	150 MHz	0,35 µ	3,52 V	60 MHz	2,5
P166	166 MHz	0,35 µ	3,52 V	66 MHz	2,5
P200	200 MHz	0,35 µ	3,52 V	66 MHz	3

Segmentos del mercado en para los cuales es aconsejable un ordenador con procesador de la familia Pentium:

- ordenadores personales con altas prestaciones
- el uso como servidores de redes de área local y sistemas multiprocesador

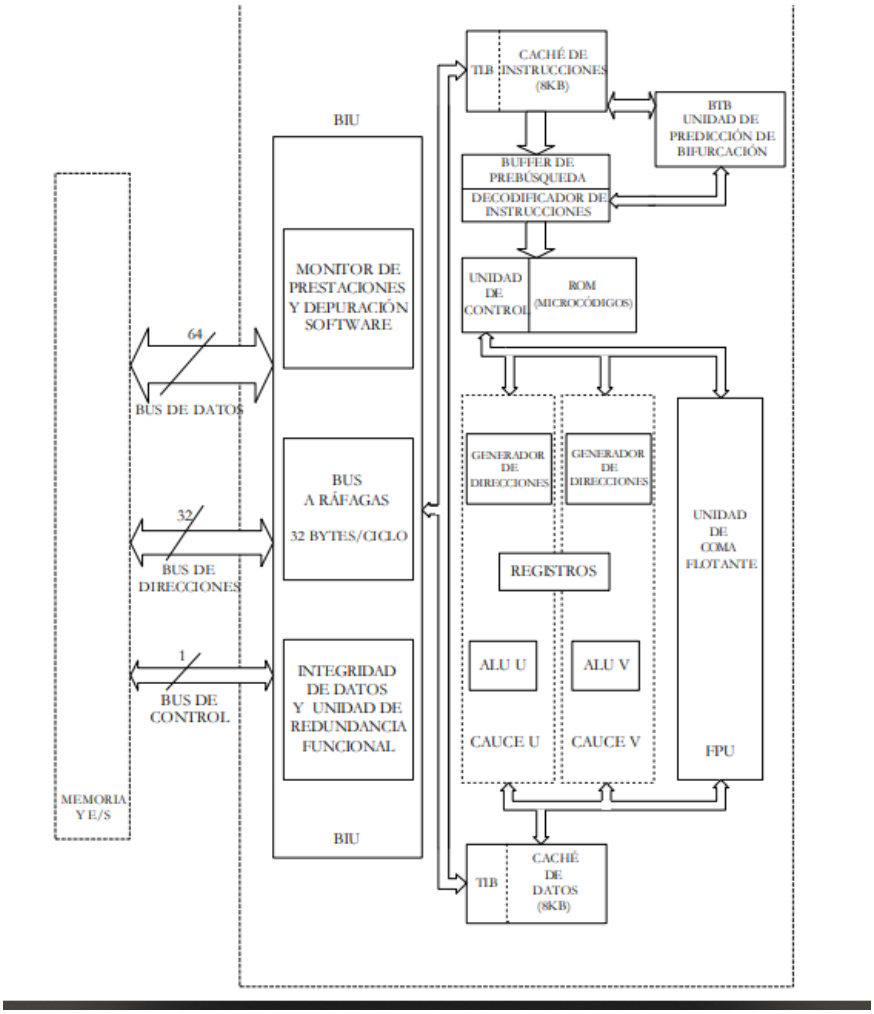
una novedad que incorporó el Pentium fue la división de la memoria caché de primer nivel L1 en dos secciones independientes de la misma capacidad (8 KB), una dedicada a las instrucciones y otra a los datos. También destaca la ampliación del bus de datos externos a 64 líneas bidireccionales y la potenciación de la Unidad de Coma Flotante

Arquitectura interna del Pentium

su rango de direccionamiento es de 4 GBytes de memoria física o principal.
Direcciones físicas: 32 bits (232 = 4GB).
Memoria virtual: 64 TBytes
Direcciones virtuales: 46 bits (246 = 64 TB)
Bus de datos: 64 bits.
Bus de direcciones: 32 bits.
Bus de control: 1 bit.

BLOQUES EN LOS QUE SE DIVIDE LA ARQ. INT DEL PENTIUM:

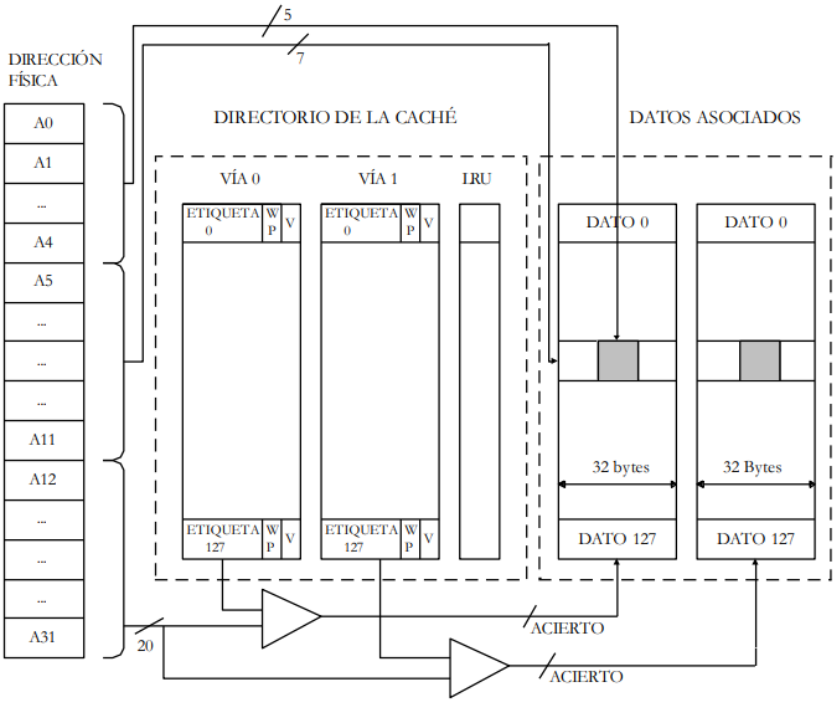
- Subsistema de memoria caché
- Unidad de enteros superescalar
- FPU: coprocesador matemático
- Sistema de predicción de saltos condicionales
- BIU: unidad de interfaz con el bus
    - Monitor de presentaciones
    - Bus a ráfagas
    - Redundancia funcional



Subsistema de memoria cache:

El subsistema de memoria caché se compone de dos memorias caché independientes de 8KB cada una: una para almacenar instrucciones (código) y otra para almacenar datos.

Las dos memorias cachés son memorias asociativas de dos vías que utilizan como unidad de información una línea que es de 32 Bytes, ya que el bus externo del Pentium es de 64 bits



EL bus que parte de la cache de datos es de 64 bits.
El bus que conecta la cache de instrucciones con los registros de prebusqueda de inst es de 256 bits.
La caché de datos utiliza el protocolo MESI. Puede haber varias cachés y puede darse que dos CPU´s estén empleando la misma posición de la memoria principal. Así dicha posición se encontrará en ambas cachés. Este protocolo asegura que se lea el dato que este más actualizado. Cada línea puede tener 4 estados:
M (modificado), E (exclusivo), S (simultaneo), I (invalida).
Cuando se precisa almacenar instrucciones o datos en la caché correspondiente y está totalmente ocupada con valores válidos, se usa el algoritmo de sustitución de líneas LRU (Last Recently Used). Se reemplaza la línea que hace más tiempo que no se usa.

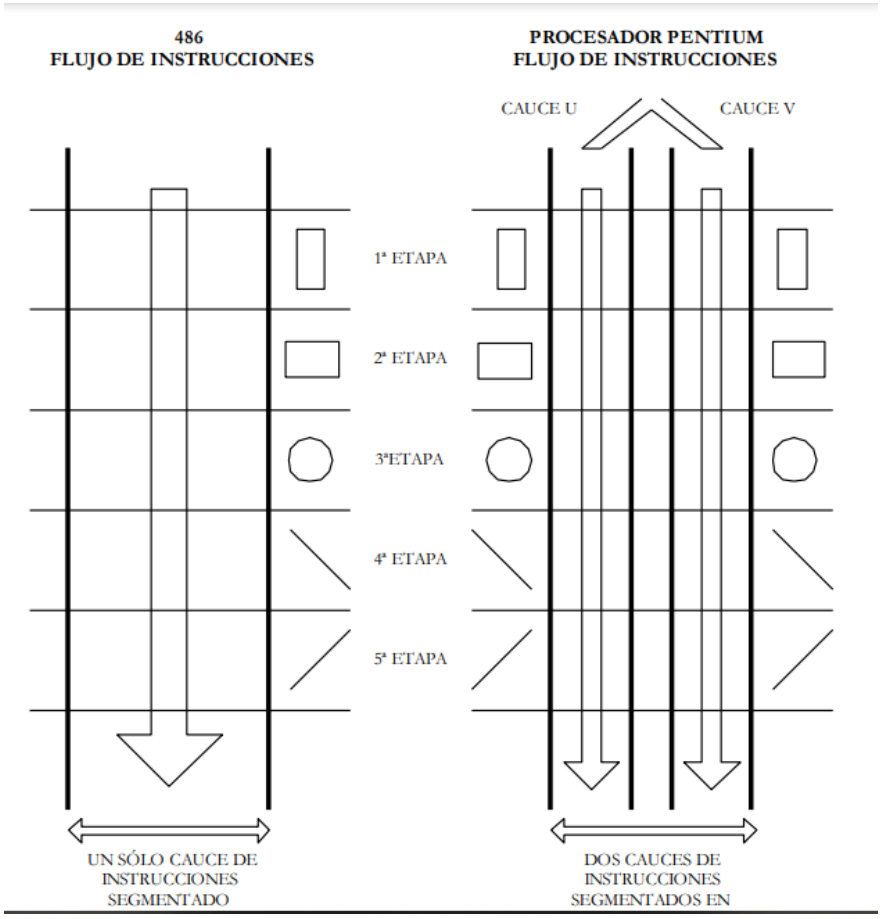
Las caches son de escritura obligada, implica que los resultados de las operaciones no se transfieren a memoria principal sino que se quedan dentro de la caché hasta que sea preciso actualizarla.

Existen dos situaciones que obligan a actualizar la memoria principal:

- Cuando se va a machacar una línea de la caché porque está llena y ésta no ha sido transferida a memoria principal.
- Cuando algún otro procesador, por ejemplo DMA intenta acceder a una posición de memoria principal cuyo dato está en la caché y ha sido modificado por la CPU

Unidad de enteros superescalar

Superescalar: en su interior existe más de una unidad de ejecución dedicadas a realizar las mismas funciones



existen dos cauces (U y V) que operan en paralelo para ejecutar las instrucciones de números enteros. Son independientes entre sí.

Cada unidad de enteros tiene un cauce segmentado de instrucciones de cinco etapas:

- Prebúsqueda de instrucciones
- Decodificación
- Cálculo de la dirección efectiva (búsqueda de operandos)
- Ejecución
- Escritura de los resultados

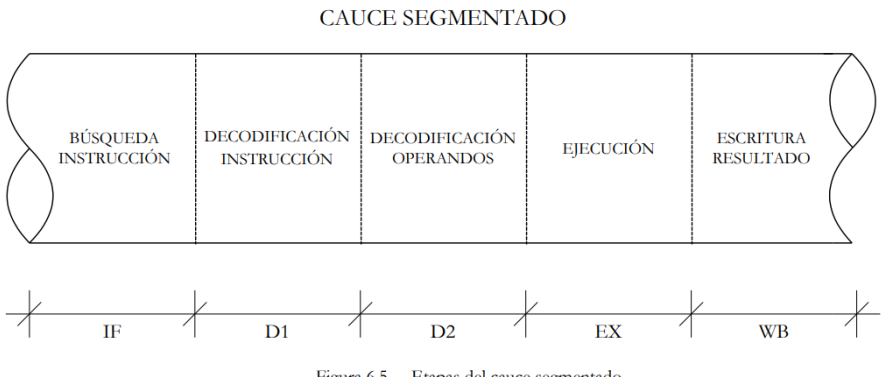


Figura 6.5. Etapas del cauce segmentado

Cada unidad de proceso interno tiene su propia unidad aritmético lógica (ALU) con un circuito de generación de direcciones exclusivo y un interfaz específico con la memoria caché de datos. Los resultados de las operaciones se almacenan en la caché interna y no se transfieren a la memoria principal a no ser que sea necesario

La Pentium es capaz de ejecutar 1,3 instrucciones por cada ciclo de reloj.

Funcionamiento de la segmentación:
La unidad de prebúsqueda manda una dirección a la caché de instrucciones. Si la caché tiene dicha dirección manda una línea de información (32 bytes) a uno de los buffer de prebúsqueda que a su vez pasará la dirección en cuestión a la unidad decodificadora donde decodificará la información. Inicialmente las instrucciones son decodificadas para ver si pueden ser ejecutadas a la vez. En caso afirmativo, una instrucción irá al cauce U y otra al V para realizar simultáneamente. Si existen dependencia entre ellas, la primera deberá completar su ejecución antes de que comience la segunda.
Cuando se predice un salto, la dirección de esta instrucción es demandada por la caché de instrucciones. Si se encuentra allí, una línea de código se manda al otro buffer de prebúsqueda de tal manera que se impide cualquier retraso cuando la instrucción branch se ejecute. Si no hay instrucciones de este tipo ambos cauces son tratados conjuntamente, realizando las prebúsquedas linealmente.

Pairing/ emparejamiento: se le denomina a la introducion y obtención de dos instrucciones en cada etapa del cause.
EN EL PENTIUM SOLO SE PUEDEN EJECUTAR DOS INTRUCCIONES COMO MAXIMO.

FPU: Coprocesador matemático

La FPU es un coprocesador que opera con unidades enteras de otros procesadores, de los cuales coge sus instrucciones desde el mismo decodificador y secuenciador que la unidad de enteros, compartiendo con esta última el bus del sistema
La FPU consta de un cauce segmentado de instrucciones de 8 etapas que permite obtener resultados partiendo de instrucciones de coma flotante en cada ciclo de reloj.
Etapas:

1. Prebúsqueda de instrucciones
2. Decodificación
3. Cálculos de la dirección efectiva
4. Ejecución
5. Ejecución de las instrucciones de coma flotante
6. Ejecución de las instrucciones de coma flotante
7. Escritura de los resultados
8. Informe de posibles errores

Para llevar a cabo estas instrucciones el coprocesador matemático internamente posee registros:

- Registros de datos
- Registros de estado
- Registros de control
- Registros de palabra

- Registro puntero de instrucciones
- Registro puntero al último operando/ puntero de datos
- Registro de código

El coprocesador puede obtener y escribir datos en memoria de los siguientes tipos:

- Entero: Words de 16 bits, Dword de 32 bits y Qwords de 64 bits.
- Real: Words de 16 bits, Dword de 32 bits, Qwords de 64 bits y Twords de 80 bits.
- Simple precisión en coma flotante.
- Doble precisión en coma flotante.
- Doble precisión expandida en coma flotante.
- Entero con signo.
- BCD.

Sistema de predicción de saltos condicionales

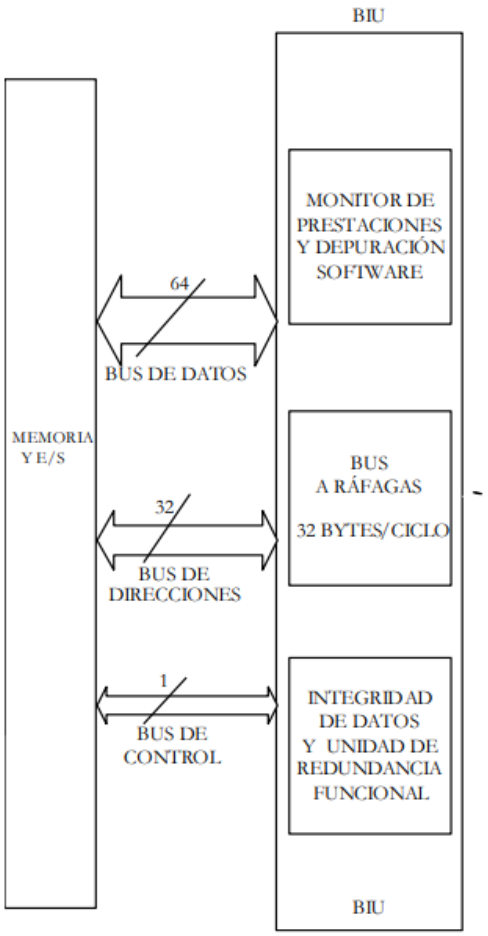
Una de las razones que más influyen en el bajo rendimiento del procesador son los saltos condicionales, que obligatoriamente introducen tres burbujas en el cauce, ya que no se sabe la siguiente instrucción a ejecutar.
Intel elimina las burbujas prediciéndolo en un solo ciclo. Para eso utiliza:

- Software: Consiste en un potente algoritmo estadístico.
- Hardware: (BTB) Buffer de destino de las bifurcaciones

Cuando una instrucción supone un salto la BTB recuerda dicha instrucción y la dirección de salto efectuada y predice, aplicando ciertos algoritmos en qué dirección se va a producir el salto la próxima vez que se ejecute.

BIU: Unidad de interfaz con el bus

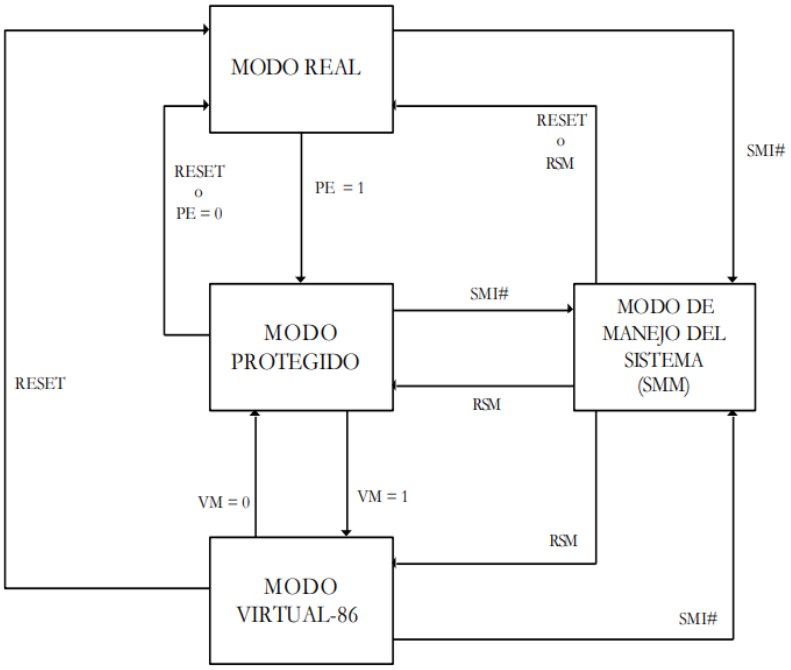
La BIU (“Bus Interface Unit”) es el bloque encargado de soportar todas las transferencias con el mundo exterior. Controla los ciclos del bus que acceden a la memoria y a las E/S.



La BIU consta de tres partes:

- Monitor de prestaciones : Desarrollar aplicaciones es cada vez más complejo y precisa de una cuidadosa realización para evitar que la mayor parte del tiempo se pierdan ciertas rutinas o selecciones del código que no son excesivamente importantes. Para facilitar el trabajo de los desarrolladores de software, el procesador Pentium incorpora un monitor de prestaciones y una unidad de depuración software
- Bus a ráfagas: Es una circuitería de silicio que permite cargar 256 bits (32 Bytes que es igual a la línea de caché) en la caché de datos en un ciclo.
  los valores medios corresponden a:
  - 36%: Prebúsqueda de instrucciones
  - 21%: Lecturas de datos
  - 36%: Escritura de datos
  - 7%: Escrituras obligadas de datos (L1)
- Unidad de redundancia funcional: Es un recurso que emplea diferentes técnicas para la detección de errores tanto externa como internamente, para asegurar la integridad de los datos

Modo de funcionamiento del Pentium



Modo real:

El Pentium solo trabaja con el primer MByte de la memoria, siendo 64 KBytes la capacidad máxima que puede ocupar un segmento.

El bus de direcciones utiliza los 20 bits de menos peso. Este modo es monotarea y los registros de propósito general (AX, BX, CX, DX, SI, DI, SP, BP), IP y flags son de 16 bits. La tabla IDT es 256 entradas y en cada una de ellas hay un vector de interrupciones que apunta al inicio de la rutina que atiende esa entrada.

Se empieza a trabajar con un reset.

Modo protegido:

Cuando el Pentium trabaja en modo protegido (multitarea), un segmento queda caracterizado por tres parámetros fundamentales que son comprobados automáticamente por el sistema de protección cada vez que se utiliza. Dichos parámetros son:

1) Base, es la dirección lineal donde comienza el segmento. Está formada por 32 bits, que es la longitud de la dirección de la memoria física que puede alcanzar un tamaño máximo de 232 = 4 GB.

2) Límite, consta de 20 bits que determinan con exactitud el tamaño del segmento usado por el programador y en el que residen informaciones válidas. Si está expresado en bytes, el límite máximo sería de 2 elevado a 20 = 1 MB y si está expresado en páginas de 4 KB, un segmento puede ser tan grande como la memoria principal, es decir 4 GB.

3) Atributos o derechos de acceso, se trata de un campo de 12 bits, que proporciona las características relevantes del segmento como:

- Tipo de segmento, admitiendo las variantes de legible, escribible, ejecutable o una combinación de estos.

• Nivel de privilegio, que oscila entre 0 y 3. Es el grado de seguridad que tiene el contenido del segmento en el sistema.

• Indicadores sobre aspectos relacionados con la gestión de la memoria virtual, como el que indica si el segmento se halla cargado o no en la memoria física.

Modo virtual:

Es una mezcla de modo real y modo protegido. Trabajamos en un ambiente igual que en modo protegido, en un ambiente multitarea y con sistema de protección entre las tareas, pero se permiten ejecutar tareas del 8086 (del modo real).
Para pasar del modo protegido al modo virtual, basta con poner el bit VM a 1 del registro de estado, en caso contrario habrá que poner el bit VM a 0.

Modo de manejo del sistema (SMM):

En este modo el Pentium proporciona un sistema operativo que es transparente para el programador y que implementa dos funciones muy importantes: la primera función está relacionada con la seguridad de todo el sistema y mejora dicha seguridad; la otra función es un sistema de control de la alimentación que controla el consumo del procesador del sistema y lo mejora.
Para pasar de cualquiera de los otros tres modos a SSM hay que activar por hardware una patita del Pentium. Se trata de la patita SMI# que se activa por nivel bajo.
Para pasar a modo real desde este modo, basta con que se produzca un reset.

Segmentación en modo real

Cuando el Pentium funciona en modo real (monotarea), compatible con el 8086, y sin ningún tipo de protección ni posibilidad de manejo de memoria virtual, un segmento queda definido básicamente por los siguientes elementos:

1. Base o dirección de comienzo de 20 bits.

2. Desplazamiento o tamaño de 16 bits.

El tamaño máximo que se admite en este modo para mantener la compatibilidad con el 8086, es de 64 KB y la capacidad máxima de la memoria principal sólo es de 1 MB. En modo real todo segmento de código, datos o pila está especificado por una dirección lógica, compuesta por dos campos de 16 bits cada uno.

Segmentación en modo Protegido

Cuando el Pentium trabaja en modo protegido (multitarea), un segmento queda caracterizado por tres parámetros fundamentales que son comprobados automáticamente por el sistema de protección cada vez que se utiliza. Dichos parámetros son:

1) Base, es la dirección lineal donde comienza el segmento. Está formada por 32 bits, que es la longitud de la dirección de la memoria física que puede alcanzar un tamaño máximo de 232 = 4 GB.

2) Límite, consta de 20 bits que determinan con exactitud el tamaño del segmento usado por el programador y en el que residen informaciones válidas. Si está expresado en bytes, el límite máximo sería de 2 elevado a 20 = 1 MB y si está expresado en páginas de 4 KB, un segmento puede ser tan grande como la memoria principal, es decir 4 GB.

3) Atributos o derechos de acceso, se trata de un campo de 12 bits, que proporciona las características relevantes del segmento como:

- Tipo de segmento, admitiendo las variantes de legible, escribible, ejecutable o una combinación de estos.
- Nivel de privilegio, que oscila entre 0 y 3. Es el grado de seguridad que tiene el contenido del segmento en el sistema.
- Indicadores sobre aspectos relacionados con la gestión de la memoria virtual, como el que indica si el segmento se halla cargado o no en la memoria física.

Capítulo 14: Interrupciones y Excepciones

Una interrupción es generalmente un acontecimiento externo que desvía el flujo de control de la CPU. Un ejemplo puede ser la activación mediante un flanco ascendente/descendente de alguna patita del procesador, o por la ejecución de alguna de las instrucciones específicas que el procesador dispone para generarlas.

Una excepción es una desviación del flujo de control provocada automáticamente como consecuencia de alguna anomalía en la CPU, producida y detectada en el desarrollo del programa en curso de ejecución. Un ejemplo es la ejecución de la instrucción de la división por cero.

Cada interrupción está asociada con un número que la identifica; este permite convocar al servicio que la atiende.

Un Pentium maneja las interrupciones y excepciones en base a:

El Pentium tiene dispone de una Tabla de Interrupciones llamada IDT para manejar interrupciones y excepciones. Esta tabla tiene 256 entradas, cada una de las cuales atiende a un tipo de interrupción o excepción diferente. Mediante un mecanismo, cada entrada especifica la dirección de comienzo del procedimiento que atiende la causa que la ha provocado

Las interrupciones internas:

Se tratan de señales eléctricas activadas por componentes hardware externos, que provocan la activación de una las patitas del Pentium. Esta activación es detectada por el Controlador de Interrupciones Programable Avanzado Local (APIC).

Cuando el APIC no está habilitado, las patitas de este se configuran como las señales INTR y NMI del procesador. Cuando está habilitado, las patitas se pueden configurar a través de la tabla de vectores para asociarlo con cualquier vector de interrupción o excepción del procesador.

La diferencia entre interrupción externa enmascarable e interrupción externa no enmascarable:

• NMI: Es una interrupción no enmascarable (no puede ser ignorada). Puesto que siempre es atendida, suele ser el resultado de un problema hardware serio. Un ejemplo de este tipo de interrupción es la caída de tensión de la alimentación.

• INTR: Es una interrupción enmascarable (puede ser atendida o ignorada por la CPU), que se origina por la activación de su patita INTR con un flanco activo. Un ejemplo es cuando se necesita la atención al periférico de entrada.

Los tipos de interrupciones:

Interrupciones

- *Externas o Hardware*: Son convocadas asincrónicamente en cualquier momento. No se encuentra bajo el control del programa.
    - *No Enmascarables*: Siempre es atendida. Se considera de máxima importancia. Se avisa al CPU por la señal NMI.
    - *Enmascarables*: Se produce según una condición. Ej.: Si EI esta activada, es atendida por el CPU. De ser así, un circuito externo especial, llamado Controlador Programable de Interrupciones, decide que prioridad tiene sobre las demás peticiones de interrupción posibles y le da curso a la consulta del vector de interrupciones correspondiente. INTR
- *Internas o Software*: Se convocan en forma sincrónica en el programa. Se pueden producir por la ejecución de una instrucción específica dentro de un programa, para solicitar una interrupción que cumpla con alguna función determinada. INTO / INT n.

Los tipos de excepciones:

Las excepciones son provocadas automáticamente por el procesador al detectar alguna anomalía en el flujo de control.

Los tipos de excepciones son las siguientes:

• **Excepciones faltas o errores:** son aquellas excepciones que se encargan de corregir el error o la falta al intentar ejecutar una instrucción, retornando al lugar donde la CPU la dejó, tras la finalización de la excepción. De esta forma la instrucción que la había provocado se puede realizar. Un ejemplo de este tipo de excepción es cuando la CPU ejecuta una operación matemática y aún no tiene todos los operandos que están involucrados en la instrucción.

• **Trampa:** son aquellas excepciones que se generan tras la finalización de la instrucción. Un ejemplo de este tipo de excepción son las interrupciones definidas por el usuario e incluidas en el programa.

• **Aborto:** son aquellas excepciones que no permiten la localización exacta de la instrucción que la origino. Se suele usar para indicar errores muy graves, como los que se originan del comportamiento del equipo físico o valores en las tablas que manipulan el sistema.

Resguardode contexto: Cuando se produce una interrupción y esta no provoca la finalización del programa en ejecución, se debe resguardar la información que se aloja en los registros del micro. Toda esta información asociada con la ejecución se la almacena en la pila de MP.

Restauración de contexto: Cuando el servicio se ejecuto por completo, el programa interrumpido debe continuar su ejecución, pero los registros contienen información inútil, es entonces el momento de restaurar el entorno del CPU "rescatando" la información que tenia en los registro desde la pila.

Ósea, todo el proceso de interrupción se lleva a cabo en los siguientes pasos:

- Programa en ejecución
- Presentación de la interrupción
    - Resguardo de contexto de CPU en la pila
    - Ejecución del servicio de atención de interrupción
    - Restauración del contexto de CPU
- Programa nuevamente en ejecución.

Fases de atención a una interrupción o excepción

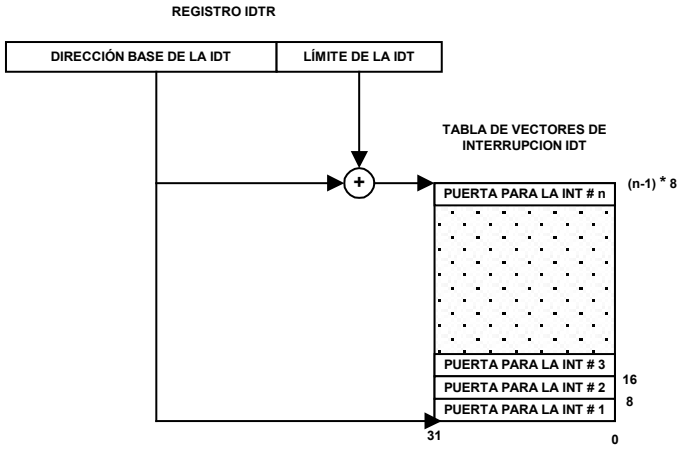
Cuando se produce una interrupción o una excepción. Los pasos que hace la CPU para tratarla son los siguientes:

1. Analizar si hay más de una interrupción pendiente. En caso de haberla, se selecciona aquella solicitud con mayor prioridad. Las prioridades son las siguientes (de mayor prioridad a menor prioridad):
    - ***a.*** Reset del hardware y comprobación de la máquina.
    - ***b.*** Depuración en el cambio de tarea.
    - ***c.*** Depuración de la instrucción previa.
    - ***d.*** Interrupciones externas.
    - ***e.*** Faltas procedentes de la captura de las instrucciones siguientes.
    - ***f.*** Falta procedente de la codificación de las instrucciones siguientes:
    - ***g.*** Longitud de instrucción > 15 bytes.
    - ***h.*** Código OP ilegal.
    - ***i.*** Coprocesador no válido.
    - ***j.*** Faltas en la ejecución de una instrucción
2. Se salva en la pila el contenido del CS, IP y el registro de estado E-FLAGS. Además, se pone a 0 los bits TF e IF.
3. Busca el vector predefinido, mirando en la IDT. Si no viene predefinido, viene dado por la instrucción INTR que está en los bits de menos peso del bus de datos (D0 - D7).
4. Al finalizar la rutina de la interrupción, con la instrucción IRET, se saca de la pila los datos antes salvados, CS, IP y E-FLAGS para continuar con lo que antes estaba haciendo el CPU.

La tabla de descriptores de interrupción (IDT)

Es la tabla que usa el procesador para manejar las interrupciones y excepciones. IDT es el bit que indica que el error se ha producido en la tabla IDT, si vale 1 y en caso de que valga 0, el error se ha producido en la GDT o LDT.

La tabla IDT contiene 256 entradas para el tratamiento de las posibles interrupciones. Ésta está cargada por descriptores de puertas, que ocupan 8 Bytes cada uno, haciendo que el tamaño máximo de la IDT sea de 2K.



Vector	Descripción	Flag	Causa	Código de error	Clase
0	Error de división	#DE	DIV IDIV	No	Falta
1	Excepción de depuración	#DB	Cualquier código/dato de referencia/INT1	No	Falta/trampa
2	Interrupción NMI	-	Interrupción externa no enmascarable	No	Interrupción
3	Punto de ruptura	#BP	INT 3	No	Trampa
4	Desbordamiento	#OF	INT 0	No	Trampa
5	Comprobación de limites	#BR	BOUND	No	Falta
6	Código OP no valido	#UD	UD2 o código OP reservado	No	Falta
7	Coprocesador matemático no disponible	#NM	WAIT/FWAIT o coma flotante	No	Falta
8	Doble falta	#DF	Instrucciones que originen una excepción, NMI o INTR	Si (cero)	Aborto
9	Desbordamiento del segmento del coprocesador	-	Instrucciones de coma flotante	No	Falta
10	TSS no valido	#TS	Acceso TSS o conmutación de tareas	Si	Falta
11	Segmento no presente	#NP	Carga de registros de segmento o acceso de segmentos	Si	Falta
12	Excepción en la pila	#SS	Operaciones de pila y carga registros SS	Si	Falta
13	Protección general	#GP	Referencia a memoria y comprobación	Si	Falta
14	Fallo de pagina	#PF	Referencia a memoria	Si	Falta
15	Reservados por Intel	-	-	No	-
16	Error de cina flotante FPU	#MF	WAIT/FWAIT o coma flotante	No	Falta
17	Comprobación de alimentación	#AC	Cualquier dato referenciado en memoria	Si (cero)	Falta
18	Comprobación de la maquina	#MC	Códigos de error y fuentes son modelos dependientes	No	Aborto

19	Excepción de coma flotante SIMD	#XF	SEE SSE2	No	Falta
20-31	Reservados por Intel	-	-	-	-
32-255	Interrupción definida por el usuario	-	Interrupción externa o INT n	-	Interrupción

Cómo está compuesta y cuál es la principal diferencia en la estructura IDT en Modo Real y en Modo Protegido

La Tabla de Descriptores de Interrupciones (IDT), está compuesta por dos instrucciones la LIDT y SIDT. La instrucción LIDT carga en la IDTR la dirección base y el límite. Esta instrucción puede ser ejecutada únicamente cuando el CPL=0.

Como en Modo Protegido, la IDT del Modo Real dispone de 256 entradas para las interrupciones.

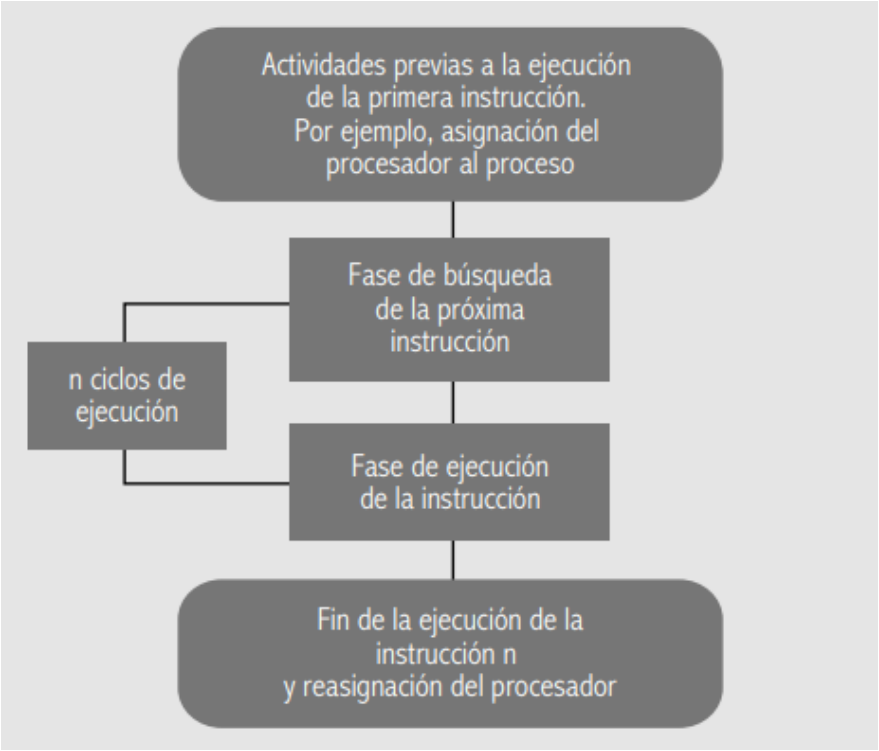
La diferencia entre el Modo Protegido y el Modo Real, reside en el tipo de entradas que dispone cada uno. En Modo Protegido, la tabla contiene descriptores, que son de 64 bits. En cambio, las entradas en Modo Real son de 32 bits cada una.

Los descriptores de la IDT responden a tres tipos de puertas:

- *Puertas de tarea*: Las puertas de tarea son un tipo especial de descriptor del sistema ubicados en la GDT, en la LDT o en la IDT.
  En la entrada de la IDT se pueden encontrar puertas de tareas. Esto quiere decir que la rutina donde está localizada la interrupción está en otra tarea distinta a la tarea en curso. Esto tiene la ventaja de que hay una independencia entre tareas, pero como inconveniente hay que decir que una conmutación de tarea consume mucho tiempo en realizarla.
  Esto quiere decir que posee una ventaja ya que cada tarea sirve para realizar una conmutación de tareas, pero posee la desventaja de una lentitud mayor. La puerta de tarea realiza una indirección sobre el TSS, comportándose de forma similar a la puerta de llamada.
- *Puertas de interrupciones y excepciones*: No hay conmutación de tareas como con las puertas de tarea. En este caso lo que se hace es conmutar dentro de la misma tarea a un segmento de código donde resida la rutina de interrupción.
  En la primera fase de la interrupción se detecta la interrupción o excepción, y en la segunda fase se guardan los valores del estado actual de la pila para poder retornar al finalizar la interrupción.

  En caso de encontrarse en el mismo nivel de privilegio, las puertas de interrupción tienen que salvar en la pila los registros CS, EIP y EFLAGS. Si la puerta es de excepción entonces IF = 1 (permitiendo atender las interrupciones enmascarables). Mientras que, si la puerta es de interrupción, IF=0 prohibiendo las interrupciones enmascarables.

Capítulo 7 Quiroga



Ejemplo de un programa (en assembler y código de máquina) para introducir el concepto de pipeline o segmentación de instrucciones:

1531:0100	8B1E0002	MOV	BX[200]
1531:0104	0307	ADD	AX,[BX]
1531:0106	D1E0	SHL	AX,1
1531:0108	83C302	ADD	BX,+02

Este programa se halla almacenado en un segmento de código, cuya base es 1531

1. MOOV BX ,[200]: Es una instrucción de transferencia y está ubicada a partir del desplazamiento 0100 respecto de 15310 (dirección física).
   La instrucción ocupa 4 bits: 0100, 0101 (corresponde al código de operación 0B1E), 0102 y 0103 (corresponde al campo DATA)
   Es una instrucción que requiere la lectura de un dato de memoria alojado en el desplazamiento 0200 y su carga posterior en el registro general BX. El modo de direccionamiento es directo
2. ADD AX,[BX]: Es una instrucción de aritmética que comienza con el byte 0104 y ocupa 2 bytes: el 0104 y el 0105,, que corresponde al código de operación 0307
   Su ejecución suma al registro general AX el valor de BX (los corchetes indicas "contenido de") y no accede a memoria para buscar dato. El modo de direccionamiento será directo de registro

3. SHL AX,1: Es una instrucción de desplazamiento que comienza en el byte 0106 y ocupa 2 bytes: el 0106 y el 0107, correspondientes al código de operación D1E0 que permite el desplazamiento de 1 bit a la izquierda (SHIFT LEFT) del registro AX y, por lo tanto, no accede a la búsqueda de un dato en memoria principal. El modo de direccionamiento será inmediato al registro
4. ADD BX,+02: Es una instrucción aritmética en el byte 0108 y ocupa 3 bytes: el 0108, 0109, 010A. Los dos primeros correspondientes al código de ocupación 83C3, que efectua la suma del valor 02 al registro BX.
   No accede a memoria para obtener el dato y permanece al modo de direccionamiento inmediato al registro

Secuencia de llenado de cola

La BIU: se encarga de controlar la transferencia entre el entorno y el microprocesador; también entrega las instrucciones a la unidad de decodificación a medida que se necesiten.

Unidad de decodificación: asociada a la BIU por un externo y a la EU, es la que interpreta el código de operación, reconoce el verbo de la instrucción, cuantos bytes mide en total para solicitarlos si es necesario y como se obtiene el dato según el modo de direccionamiento especificado en el código de operación.

De esta manera genera la demanda de cola y, como es independiente de la unidad de ejecución, puede darse el caso de que mientras se decodifica una instrucción todavía se este ejecutando la anterior, generando un palalelismo

La tendencia "ideal" es la de ejecutar una instrucción por ciclo de reloj.

El objetivo es extraer una instruccion de memoria, decodificarla y redirigirla a unidades funcionales de la EU. Mientras estas unidades "ejecutan" la instrucción, deberá extraerse y decodificarse otra instrucción para redirigirla a otras unidades de la EU no comprometidas en la ejecución de la anterior. Así, cuantas más unidades formen parte de la EU mayor será la posibilidad de "ejecución en paralelo"

La EU debe contar con registros epeciales e invisibles al programador de aplicaciones para resguardar el entorno de ejecución al momento que se produzca el salto.

Pipeline o tubería (iii)

▪ Secuencia de llenado de la cola

▪ BIU → se encarga de controlar la transferencia entre la MP y el Microprocesador → entrega las instrucciones a la unidad de decodificación

▪ Unidad de decodificación → por un extremo asociado a la BIU de la que recibe las instrucciones → por el otro extremo asociado a la EU que ejecuta las instrucciones → interpreta el código de operación → cuantos bytes mide la instrucción → si hay datos

involucrados y que tipo de direccionamiento se utiliza para poder solicitarlos (en base a esto se genera la demanda de la cola)

- Secuencia de llenado de la cola

Tabla 8-4. Secuencia de llenado y vaciado de la cola de 6 bytes de a 2 bytes.

	XX vacía	XX vacía	XX vacía	XX vacía	XX vacía	XX vacía	Estado inicial
Hacia unidad de decodificación	XX vacía	XX vacía	XX vacía	XX vacía	8B	1E	Desde memoria
	XX vacía	XX vacía	8B	1E	00	02	
	8B	1E	00	02	03	07	
	00	02	03	07	D1	E0	
	03	07	D1	E0	83	C3	
	D1	E0	83	C3	02	XX vacía	
	83	C3	02	XX vacía	XX vacía	XX vacía	
	02	XX vacía	XX vacía	XX vacía	XX vacía	XX vacía	
	XX vacía	XX vacía	XX vacía	XX vacía	XX vacía	Estado final	

Tabla 8-5. Etapas de ejecución de la rutina ejemplo.

Ciclo 1	*Ciclo 2*	*Ciclo 3*	*Ciclo 4*	*Ciclo 5*	*Ciclo 6*	*Ciclo 7*	*Ciclo 8*	*Ciclo 9*	*Ciclo 10*	*Ciclo 11*
FetchI1 8B1E		FetchI1 *0002*	FetchI2 *0307*		FetchI3 D1E0		FetchI4 83C3		FetchI4 02XX	
	DECO I1			DECO I2		DECO I3		DECO I4		
			EU I1	EU I1	EU I1	EU I2	EU I2	EU I3		EU I4

Capacidad de interrupción

Las **interrupciones** y las **excepciones** son acontecimientos causados tanto por los dispositivos E/S como por el programa que se ejecuta en el microprocesador y su efecto produce una suspensión de la actividad actual del micro, para pasar a ejecutar un servicio que "interprete el manejo de esa interrupción".
Los dispositivos externos utilizan interrupciones para informa su estado o solicitar la ejecución de actividades que le son necesarias. Los programas a su vez solicitan información de los dispositivos de E/S.
Cada interrupción está asociada a un número que la identifica

Interrupciones internas: interrupciones programadas que causan la suspensión momentánea del programa que las convoca para bifurcar el servicio solicitado, este se ejecuta y retorna el programa interrumpido.

Interrupciones externas: servicios residentes en memoria principal que son bifurcados luego de que el microprocesador reciba una señal de interrupción y deje la ejecución del programa actual

Diferencia entre los distintos tipos de interrupciones

	No enmascarablas (NMI)	
	La CPU es alertada por la señal de control NMI	Siempre son atendidas
Externas o hardware	**Enmascarables**	
		Hay una bandera que autoriza estas interrupciones.
Son convocadas		Flag F=1, la CPU suspende de manera nmomentanea
de forma	La CPU es avisada por una señal distinta a la MMI	la ejecucion de programa activo y ejecuta el servicio
asincronica		de int.
Internas o externas	-	La forma de convocarla es INT #, siendo 3 el numero de la interrupcion.
Son convocadas por el programa		
	Faltas o errores	Son las que pueden detectar y corregir antes de que se produzca
Excepciones		la ejecucion de una instrucción determinada.
Son provocadas como	**Trampas**	Se detectan una vez ejecutada la instrucción que las provoca.
consecuencia que		
se producen y	**Abortos**	Se detectan sin localizar la instrucción que las procova, abortado
detectan durante		la ejecucion del programa.
la ejecucion del prog.		

Cuando se produce una interrupción y cuando esta no provoca la finalización del programa en ejecución, se debe resguardar la información que se aloja en todos los registros del micro y que se relaciona con la ejecución del programa interrumpido.

Esta información se almacena en MP, en una estructura de dato denominado pila asociada al programa. Este procedimiento permite **resguardar el entorno de** CPU para reanudar la ejecución a partir del momento que se produjo la interrupción, "rescatando" la información que tenían los registros internos de la pila. Este último procedimiento se conoce como **restauración de contexto** de CPU.

Pasos que pasan cuando se detecta una interrupción y esta puede ser atendida:

1) Programa de ejecución
2) Presentacion de la interrupción :
    1. **Resguardo del contexto en la Pila.**
    2. **Ejecución del servicio asociado a la interrupción.**
    3. **Restauración del contexto**.
3) Programa nuevamente en ejecución

Los servicios que atienden a las diversas interrupciones y excepciones se hallan en MP. Para conocer donde se aloja un servicio, se mantiene en memoria una tabla de vectores de interrupción de n entradas. Cada uno de estos se corresponde con el número de interrupción; así, la INT 10, encuentra su vector en la entrada 10 hexadecimal. El término vector debe asumirse como señalador o puntero. Estos vectores señalan zonas de memoria RAM o ROM. Hay tantas entradas como servicios definidos, cada vector contiene la posición del servicio.

Al conjunto de servicios se lo denomina **manejadores de interrupciones** (Interruption Drivers).

Cada vector usa 4 bytes, para encontrar la dirección de memoria que corresponda, Interrupción Driver, deberá seguir los pasos siguientes:

- Multiplicar el número de interrupción por 4.
- Tomar los 4 bytes que se encuentran en esa localidad, que están invertidos, y asumirlos como dos entidades separadas de 2 bytes.

- Convertirlos a segmento: desplazamiento.
- Invertir los bytes de cada palabra, ya que en memoria el almacenamiento se encuentra en orden inverso.

Concepto de pila

La pila es una estructura de dato en memoria de acceso LIFO. El registro de segmento SS o segmento de pila, se accede con criterio LIFO. La que se encarga del acceso a la pila es la CPU, ejecutando instrucciones PUSH y POP.

La CPU utiliza la pila para:

- Almacenar la dirección de retorno IP.
- Almacenar el estado del procesador cuando se produce una interrupción. Los registros que apila son el CS y el IP y estado de Flags.
- Pasar parámetros entre procedimientos.

El acceso a la pila se realiza mediante los registros punteros SP y BP. El SP es el registro que contiene la dirección del próximo elemento de la pila vacío
La carga o extracción de datos de la pila es un procedimiento software.
Estas operaciones se llevan a cabo incrementando o decrementando el registro SP.

- **CALL** y **RET** son instrucciones que sirven para invocar y dar retorno a un procedimiento o subrutina, mientras que **INT** e **IRET** cumplen la misma función cunado se invoca una subrutina de interrupción.
- **PUSH** pone una palabra en la pila y luego decrementa el **SP.**
- **POP** saca palabra de la pila y incrementa el **SP.**

Alimentación

La alimentación de los distintos componentes proviene de una fuente

Fuente: dispositivo que transforma la corriente de la red eléctrica para que sea aceptable para los circuitos electrónicos

Tecnología

Cisc:

- Se pueden ejecutar instrucciones simples o complejas (estas ultimas utilizan microinstrucciones
- Aumenta el numero de instrucciones del set cuando este admite tanta variedad de modo de direccionamiento

Risc:

- Cada instrucción tiene la mayor cantidad de microopercaiones solapadas posibles, de modo que la mayoría de ellas se ejecute en un ciclo de reloj
- Las instrucciones tienen el mismo

- Las instrucciones que tienen que realizar operaciones aritméticas son de referencia a registro y, en general, tienen especificados 2 para los operandos y 1 para el resultado
- El formato típico de una instrucción aritmética RISC es

Código de operación	Registro 1	Registro 2	Registro 3

Epic:

- Permite agrupar instrucciones para ejecutarlas de forma paralela en forma especifica
- Predicación: método para manejar saltos condicionales. La idea es que el compilador planifique ambos caminos posibles de la ramificación, para que se ejecuten en el procesador simultáneamente
- Carga especulativa: la especulación trata de aprovechar el microprocesador cuando está en periodo de latencia. En ese momento especula sobre las instrucciones y los datos que va a necesitar más adelante y los carga.

¿Cuál es la función y para que se utiliza EFLAGS?

El registro EFLAGS o también conocido como registro de estado o señaladores, consta de 32 bits de los cuales la mayoría son señalizadores de estado, controlados por la ALU (acarreo, paridad, acarreo auxiliar, cero, signo y sobre pasamiento), actuando los restantes como señalizadores del sistema, ligados al mecanismo de protección y a otros recursos de que dispone el sistema de explotación cuya misión será mejor interpretada a medida que se profundice en el estudio del procesador. A su vez, también permite ejecutar instrucciones protegidas, siempre y cuando sean determinadas por el programador en sistemas.

¿Cuál es la función y para que se utiliza el registro EIP?

El registro de puntero de instrucciones EIP se encarga de almacenar el desplazamiento que se añade a la base del segmento de código para obtener la dirección de donde está la siguiente instrucción. Este registro es capaz de trabajar de 2 maneras distintas:

Por una parte, se encuentra el modo nativo el cual recibe el nombre de EIP y posee 32 bits.

Luego se encuentra el modo real que se emplea en un direccionamiento reducido y es compatible con los procesadores 8086 y 80286, que solo precisan 16 bits para especificar el desplazamiento, hace referencia a los 2 bytes de menor peso de EIP que se denominan IP.

¿Qué es el código de error y qué información contiene?

Hace referencia a cuando el procesador está atendiendo una interrupción excepción y se detecta una nueva interrupción o excepción. Dentro del código de error hay 4 indicadores que proporcionan la información sobre este mismo:

• EXT: Es un bit que indica el origen del error. Si vale 1, indica que el error viene desde el exterior y por lo tanto y por lo tanto es una interrupción. Si por lo contrario es 0, indica que el procesador ha detectado una anomalía y por lo tanto se trata de una excepción.

• IDT: Es el bit que indica que el error se ha producido en la tabla IDT, si vale 1 y en caso de que valga 0, el error se ha producido en la GDT o LDT.

• TI: Es el bit que indica si el error proviene de la GDT, si tiene valor '0' o de la LDT si tiene valor '1'.

• ÍNDICE: Indica el selector donde se produjo el error.

El acceso a las interrupciones y excepciones en modo protegido funciona de la siguiente manera:

Al producirse la interrupción en modo protegido, se direcciona una entrada de la IDT, en la que reside el descriptor de una puerta de tarea que la va a atender. Este tipo de descriptor da lugar a una conmutación de tarea, que implica un total aislamiento de la nueva tarea en curso con respecto a la tarea que estaba ejecutando el procesador antes de la interrupción, dejando el señalizador NT con valor a 1. La tarea de la interrupción finalizará con la ejecución de la instrucción IRET, que devolverá el control a la tarea previa.

Si durante la ejecución de la tarea de la interrupción, se producen nuevas excepciones, se introducirán en la pila de la tarea de la interrupción y antes de terminar la tarea de la interrupción hay que eliminar las excepciones apiladas de la tarea.

¿Como se pasa del modo real al modo protegido en un Pentium y cuál es la consideración previa que se debe producir y cuál es su implicancia?

Para pasar a Modo Protegido sólo hay que cambiar el valor del bit PE de registro de estados CR=0 a 1.

La consideración previa que se debe producir es que el sistema operativo debe haber creado la tabla IDT en memoria principal nada más empezar el Modo Protegido.

¿Cómo opera el IDTR y el IDT al arrancar el Pentium en modo real?

Los Pentium como los demás procesadores de Intel, cuando arrancan por primera vez, trabajan en Modo Real y el sistema operativo debe tener ya en la memoria principal la tabla IDT, para funcionar. 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 IDTR con el valor 0000 0000H.

¿Cuál es la diferencia entre el programador de sistemas y el de aplicaciones?

La diferencia entre el programador de sistemas y el de aplicaciones es que el de sistemas tiene como misión construir un sistema de explotación óptimo que sea capaz de soportar todas las aplicaciones previstas. Entre sus funciones más destacadas están:

• Organizar el sistema para el correcto tratamiento de las tareas pertenecientes a los diferentes usuarios.

• Confección de objetos para sistemas operativos, depuradores, compiladores.

• Asignar a cada tarea su nivel de privilegio y un sistema de protección adecuado.

• Organizar toda la memoria y el procesador para que de esta forma las tareas consigan un mejor rendimiento.

El programador de sistemas a su vez, es el encargado de desarrollar programas y utilidades del SO; necesita mayor nivel de conocimientos ya que controla a los programas de aplicaciones. Debe conocer (de forma indispensable) profundamente la arquitectura detallada de la CPU para así optimizar todos los recursos obteniendo en su funcionamiento la máxima potencia, seguridad y rendimiento. Debe conocer también las prestaciones de la memoria virtual, mecanismos de interrupciones, excepciones, etc.

Mientras que, por otro lado, el programador de aplicaciones se encarga de crear el sistema lógico que soportan las aplicaciones del usuario, el CPU es fundamental para llevar a cabo la tarea de aplicación junto con otras distintas de acuerdo con un mecanismo de protección que controla los accesos de las mismas. Los conocimientos que el programador de aplicaciones debe tener sobre la máquina son imprescindibles para obtener el máximo rendimiento de las instrucciones usadas para resolver las aplicaciones.

• Encargado de codificar programas para el usuario final.

• Realiza aplicaciones en lenguaje de alto nivel.

• Tiene una visión limitada a los recursos del procesador.

Unidad 9

Capitulo 5

La memoria cache

Necesidad de la cache

los nuevos microprocesadores se apoyan en tres recursos fundamentales:

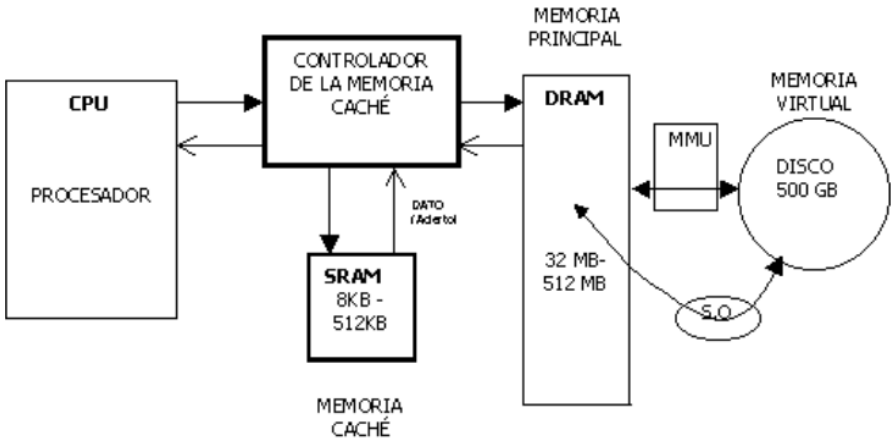
- Arquitectura superescalar
- Supersegmentacion
- Potencionamiento de la memoria cache

Un procesador segmentado ejecuta cada instrucción en cinco etapas

1. Busqueda de la instrucción (fetch)
2. Decodificación
3. Búsqueda de operandos
4. Ejecución de la instrucción
5. Escritura del resultado

1,3 y 5: consisten en un acceso a la memoria principal
2y4: Propias del procesador

Para que el rendimiento sea óptimo, el objetivo es que todas las etapas duren lo mismo. Las cachés son ultras rápidas, pero de poca capacidad y muy caras, por lo que no se pueden sustituir las DRAMs por cachés. Por lo que debemos emplear la jerarquía de memoria, la cuál consiste en interponer entre la CPU y la memoria DRAM una memoria ultrarápida (caché).



La memoria caché es una SRAM (RAM estática) que tiene un tamaño comprendido entre 8 KB y 512 KB mientras que la Memoria Principal puede alcanzar cientos de MB.
La CPU se relaciona con la Memoria Caché y si ésta contiene lo solicitado se tardan unos pocos ns en el acceso. Si hay fallo se debe acceder a la Memoria Principal.
El movimiento de datos se genera de forma que produciéndose una ausencia, la caché recibe de la Memoria Principal el dato pedido y otros contiguos, que previsiblemente va a pedir la CPU

- si empleamos una memoria caché con un tiempo de acceso tc , con una tasa de acierto del 90% y una memoria principal con un tiempo de acceso tm, la formula que refleja el tiempo medio de acceso al sistema de memoria es:

$t = 0.9 \times tc + 0.1 \times (tc + tm)$

- Si suponemos que el tiempo de acceso a la memoria caché es de 5ns y el tiempo de acceso a la Memoria Principal es de 5 y 50 ns respectivamente, el resultado es:

$t = 0.9 \times 5ns + 0.1 \times (5ns + 50ns) = 10$ ns

<u>Hay dos factores destacables en la memoria que proporciona la cache:</u>

1. **Factor de velocidad:** relación entre el tiempo de acceso a la Memoria Principal y el tiempo de acceso a la Memoria Caché:

$$\gamma = \frac{tp}{tc}$$

2. **Factor de eficacia**: La eficacia depende en gran medida, del programa que se esté ejecutando

$$\text{Eficacia} = \frac{tc}{t}$$

Principio de funcionamiento de la cache:

La cache esta estructurada en 3 bloques:

- **Bloque de etiquetas:** RAM-CAM: memoria de acceso por contenido.
- **Bloque de datos asociados:** SRAM: conjunto de datos de forma que a cada dato le corresponde una etiqueta.
- **Logica de control:** Comparadores de “n" bits, tantos como tenga la etiqueta.

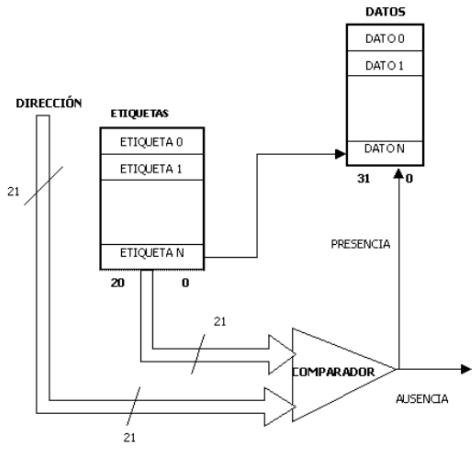


Figura 5.2. Principio de funcionamiento de la caché

El bus de direcciones va a la caché, que tiene dos partes: la de la etiquetas y la de datos. Hay N posiciones de etiquetas y cada etiqueta tiene 21 bits. Los 21 bits de la etiqueta se corresponden con los 21 bits de más peso de la dirección, por eso, el comparador coge las N etiquetas y las compara con los 21 bits de más peso de la dirección. Si alguna coincide, el

comparador devuelve un uno. Si no coincide ninguna, el comparador devuelve un cero lo que implica ausencia

Si el comparador devuelve un uno (presencia), el dato correspondiente a la etiqueta es transferido a la CPU reflejando los bits restantes, la posición en la que se encuentra el dato.

Si el comparador devuelve un cero (ausencia) se accede a la memoria principal.

Tipo de conexionado de las memorias cache:

- Conexión en serie: La CPU sólo se conecta con la caché por lo que todas las peticiones que hace la CPU al bus del sistema son a través de la memoria caché.
- Conexión en paralelo: todo depende del bus del sistema. La caché es opcional. Cada vez que la CPU realiza una petición, la envía simultáneamente a la caché y a la Memoria Principal

Arquitectura del subsistema de memoria cache:

Caracteriasticas que determinan un subsistema:

- Tamaño de la cache: suele oscilar entre 8KB y 512KB. La mejor capacidad de la caché está entre 32K y256K.
- Organización:
  1) Totalmente asociativa: cualquier posición de la memoria principal se puede ubicar en cualquier posición de la memoria caché. Como hay flexibilidad total, la etiqueta ha de contener TODOS los bits de la dirección de la memoria principal a los que corresponden los datos.
  2) Asociativa de una vía: La memoria principal se divide en bloques. Cada posición de 1 bloque de la memoria principal sólo puede ir a la misma posición de la caché. Solo es necesario precisar cuál es el bloque, porque sabiendo el bloque ya se conoce la posición de la caché. Como sólo cabe una dirección de cada bloque en una posición, tendremos que machacar la dirección anterior, cada vez que queramos modificarla.
  3) Asociativa de "n" vías: es similar al de una vía, pero la caché se va a descomponer en varias vías, no en una sola. La memoria principal se divide en fragmentos iguales a cada uno le corresponde una de las vías funcionando de la misma manera que en el caso anterior, incorporando la ventaja de que no es necesario machacar las posiciones de memoria inmediatamente que surja una modificación.
- Estructura física de una cache: La sustitución de direcciones se realiza mediante el algoritmo LRU que es aquel que selecciona la dirección que menos se ha utilizado. Cuando dicha dirección se va a incorporar en una línea y estas están ocupadas, se consultan los dos bits del LRU, y la dirección menos utilizada será la que se machaque.
- Actualizacion de una cache: Si fuera totalmente asociativa, cualquier línea se podría almacenar en cualquier posición de la caché. Si se necesita introducir una posición y hay alguna vía vacía, ésta se ocupa. Si hay una posición que las cuatro vías tengan ocupada, se aplica uno de estos dos algoritmos para extraer una de ellas y machacarla:
  - RANDOM: Aleatoriamente se elige y se machaca una de las posiciones ocupadas de una de las cuatro vías.
  - LRU: Se elimina la posición de la vía que menos se haya empleado últimamente. Hay doy alternativas para actualizarla:

A. El dato pedido va en ultimo lugar
B. El dato pedido va en primer lugar

- Actualización de la memoria principal: La memoria principal se actualiza mediante uno de estos tres métodos:
  - Actualización por escritura inmediata: Cada vez que la CPU modifica la caché, ésta última manda una orden al bus del sistema y se transfiere la información a la CPU, consiguiendo que no haya errores de coherencia y actualizando así la memoria principal
  - Actualización por escritura diferida: La caché dispone de registros intermedios donde cargas temporalmente las modificaciones que ha habido en la caché. Actualiza la memoria principal cuando el bus del sistema está libre
  - Actualización por escritura obligada: La actualización de memoria principal se produce cuando no queda otro remedio

Protocolo Mesi

En los sistemas multiprocesador puede haber varias cachés, por tanto, puede suceder que una misma posición de la memoria principal la están empleando dos CPU´s y como consecuencia, permanecer en las dos cachés. surge la necesidad de asegurar que cualquier acceso a la memoria lea el dato más actualizado. Por eso Intel desarrollo el protocolo Mesi.
asigna cuatro estados diferentes a cada línea, que definen si una línea es válida. Si está disponible para otras cachés. Estos estados pueden ser modificados bien por el propio procesador, o bien por unidades lógicas externas.

Los posibles estados son:

- **M: MODIFICADO**
- **E: Exclusiva**
- **S: Simultaneo**
- **I: Invalido**

Conexionado de caches de varios niveles:

- **Conexionado en paralelo:**
  Si la caché L1 da fallo, no importa porque a través del bus del sistema se envía la petición a la caché L2 y a la memoria principal
- Conexionado en serie:
  Desventajas:
  - Hay penalización de tiempo: Si la caché L2 da fallo transcurrirá tiempo hasta que la caché L1 reciba la petición.
  - La caché L2 es obligatoria porque la caché L1 no se puede conectar directamente al bus del sistema

  Ventajas:

  - El tráfico de peticiones a la memoria principal disminuye considerablemente y por lo tanto el bus del sistema está desocupado la mayor parte del tiempo y puede hacer frente a los sistemas adheridos.

Explique el criterio de vecindad espacial y temporal en una memoria caché.

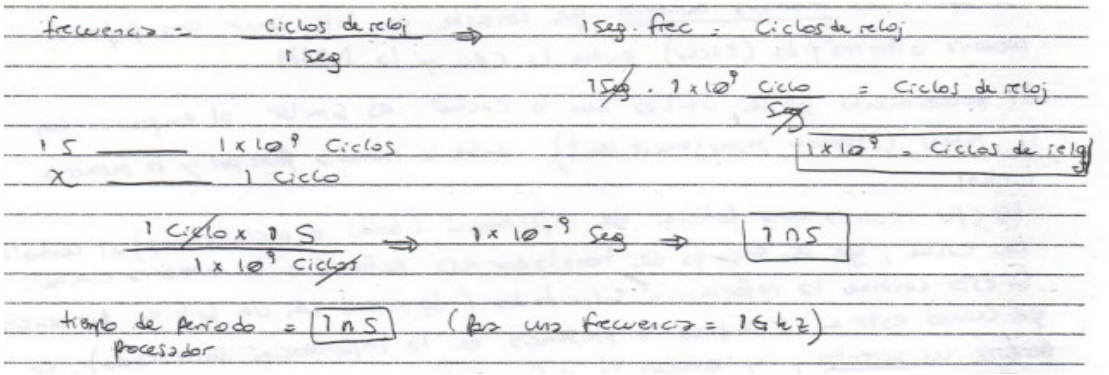
El criterio de *vecindad espacial* en una memoria caché es que, si una localización de memoria es referenciada en un momento concreto, es probable que las localizaciones cercanas a ella sean también referenciadas pronto. Existe vecindad espacial entre las posiciones de memoria que son referenciadas en momentos cercanos. En este caso es común estimar las posiciones cercanas para que estas tengan un acceso más rápido.
Por otro lado, el criterio de *vecindad temporal* en una memoria caché es que, si en un momento una posición de memoria particular es referenciada, entonces es muy probable que la misma ubicación vuelva a ser referenciada en un futuro cercano. Existe proximidad temporal entre las referencias adyacentes a la misma posición de memoria. En este caso es común almacenar una copia de los datos referenciados en caché para lograr un acceso más rápido a ellos.
Memoria cache

Una de las limitaciones mas importantes a la hora de diseñar computadoras, es la velocidad.

CPU-> Frecuencia de CK muy alta – Ejecuta gran numero de instrucciones por ciclo de CK.

Memoria, buses de datos, periféricos, etc -> Mas lentos. La CPU debe esperar para tener disponibles los datos.

EJ: Si un microprocesador trabaja a una frecuencia de 1GHZ (1x 10^9hz)



SI por otra parte la memoria DRAM, trabaja con un tiempo de acceso de 50 ns, y de acuerdo con las etapas definidas para un procesador (pentium) el tiempo total para ejecutar una instrucción seria:

ETAPAS:

1. Fetch -> acceso a mm **50ns.**
2. Decodificación de la instrucción -> la hace la CPU **1ns.**
3. Búsqueda de operandos-> acceso a mm para lectura **50ns.**
4. Ejecución de la instrucción -> EU de la CPU **1ns.**
5. Escritura de resultados - > Se almacenan en la Memoria Principal (se escribe en la misma) **50ns.**

Tiempo total = 152ns

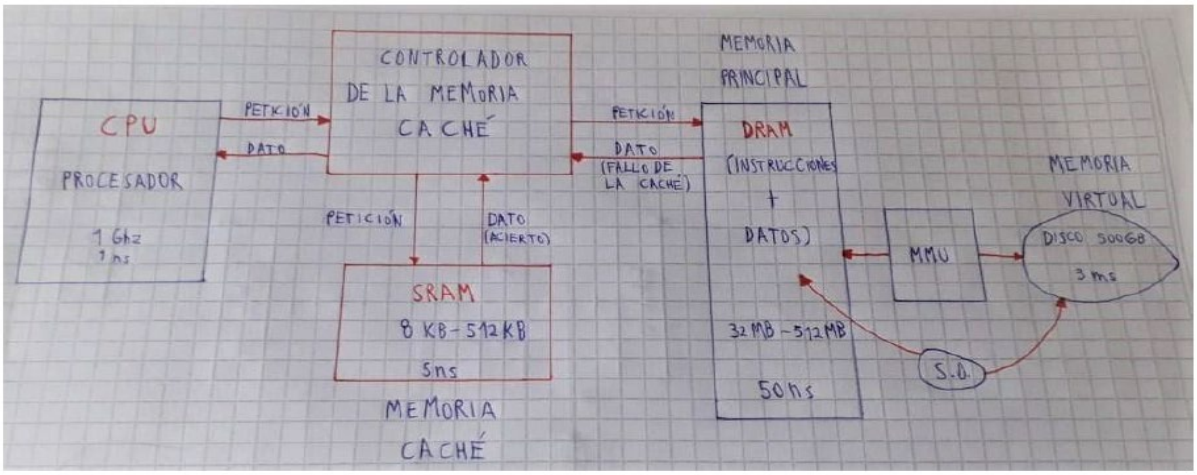
De los cuales el tiempo total de acceso a la mm(Memoria) es de 150 ns y el tiempo total de la CPU es de 2ns.

- De estos resultados, se desprende un notable desequilibrio (es el tiempo entre etapas): Para un optimo rendimiento todas las etapas deberían durar lo mismo.

- Una solución es el uso de la MEMORIA (MM) caché

  Las MM ultrarapidas pero muy caras, por lo que no reemplazan a la DRAM (tienen poca capacidad).
  Se aplica la jerarquía de MM que consiste en interponer una pequeña memoria ultrarapida (Cache) entre la CPU y la DRAM.

  El procedimiento que se utiliza con la caché es similar al empleado con la MMU (Memory Management Unit) entre la memoria principal y la memoria virtual.
  La CPU realiza una petición de información (datos o instrucciones) al controlador de cache, que se encarga de trasladar esa petición a la memoria caché.
  Si esta contiene la información solicitada (lo indica con un bit de presencia que cuando está en 1, indica la presencia de la información solicitada), se produce un acierto, y entrega la información a la CPU. En caso de no encontrarse la información (bit de presencia= 0, es decir Ausencia), se produce un fallo. Entonces se debe obtener la información de la Memoria Principal.
  Si tampoco se encuentra en la MP, la MMU activara el mecanismo para traerla desde la Memoria Virtual.



JERARQUIA DE MEMORIA

La petición a la MP puede darse, bien simultáneamente a la petición que se hace a la memoria cache (sin esperar a conocer el resultado), o bien después de esa petición y una vez que se conoce el resultado de esta ha sido un fallo. Esto depende del tipo de conexionado de la caché e influye en los tiempos de retardo y en el rendimiento.

- Si la cache tiene el dato, solo realiza el tiempo de acceso a la misma.
  Sin embargo, cuando la cache no dispone del dato solicitado, el tiempo empleado se incrementa, debido al acceso a la Memoria Principal.
- **La memoria cache** permite mejorar la productividad de los sistemas informáticos sin elevar el costo en forma significativa.

- **El movimiento de datos** se realiza de forma que, al producirse un fallo por ausencia, la memoria cache recibe de la MP no solo el dato pedido, sino también otros contiguos que previsiblemente va a pedir la CPU. De este modo, se consigue optimizar la transferencia de bloques, siempre que la programación cumpla las reglas clásicas de la vecindad espacial y temporal (simplemente poner a mano las cosas de uso más frecuente).

- **La localidad espacial:** se refiere a las direcciones físicas de MM (Memoria) en que se alojan las instrucciones o datos. Es decir, al instante en que se van a necesitar esas instrucciones o datos. Es fácil poner la siguiente información que va a solicitar la CPU, debido a que tiende a requerir datos que estén en posiciones cercanas físicamente en instantes próximos en el tiempo, y a utilizar las mismas instrucciones repetidamente.

- Hay 2 factores destacables en la memoria que proporciona la cache:

    - **Factor de velocidad:** Es la relación entre el tiempo de acceso a la MP y el tiempo de acceso a la memoria cache.

      $$r = \frac{Tp}{Tc}$$

    - **Factor de eficacia:** Depende en gran medida, del programa que se esta ejecutando, es decir, de como esta escrito y estructurado el SW.

      $$\text{Eficacia} = \frac{Tc}{T}$$

Donde:

- TP: Tiempo de acceso a Memoria Principal.
- TC: Tiempo de acceso a Cache.
- T: Tiempo promedio de acceso.

Ejemplo: * Si el tiempo de acceso a la MP es de 50ns y el tiempo de acceso a la memoria cache es de 5ns.

El factor de velocidad se calcula como -> $r = \frac{Tp}{Tc} = \frac{50}{5} = 10$ El factor de la memoria cache respecto a la MP es de 10 lo que significa que el tiempo de acceso a cache es 10 veces memoria al de la MP.

- Si además el tiempo medio de acceso, para un programa dado es de 20ns. ¿Qué eficiencia tiene?

  $$\text{Eficacia} = \frac{Tc}{T} = \frac{5}{20} = 0,25$$

- El tiempo medio de acceso depende en gran medida, de los tiempos de acceso a las memorias empleadas y del tipo de interconexión, si emplea la jerarquía de niveles, etc.

- Fórmula para calcular el tiempo medio de acceso:

$$\overline{t} = \alpha * Tc + (1-\alpha) * (Tc + Tp)$$

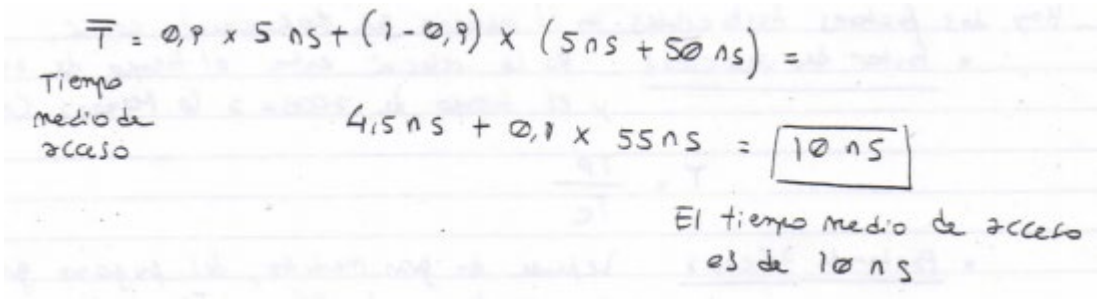
Donde:

TC: tiempo de acceso MM Cache

TP: Tiempo de acceso MP (memoria principal).

α: tasa de acierto.

Ejemplo * Si el TC= 5ns ; el TP = 50ns y la tasa de acierto (α) es del 90%.



Principio de funcionamiento de la Cache:

- La memoria cache es de tipo SRAM (RAM estático), reside muy cerca de la CPU.
- Los tiempos de acceso de la Memoria cache se encuentran entre los 3 y los 10ns aproximadamente.
- Esta diseñada para proporcionar a la CPU los datos e instrucciones que se solicitan con más frecuencia.
- Se fundamenta en la regla ”80/20” que establece que aproximadamente el 20% de todos los programas y datos en la computadora se utilizan el 80% del tiempo. Del mismo modo, el 80% restante de datos e información se utiliza el 20% del tiempo.
- El proceso de gestión de MM cache es transparente para la CPU. La única diferencia que debe encontrar una CPU con MM cache respecto a un MM RAM es la velocidad con que recibe los datos.

Componentes mas importantes de la cache:

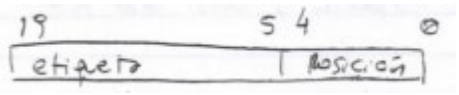
- **Controlador de Cache:** Encargado de gobernar cada uno de los elementos de los que consta la MM cache y controla los movimientos de información entre los dispositivos.
- **Directorio cache o bloque de etiquetas RAM – CAM:** Es una memoria RAM de acceso por contenido no por direcciones de la MM principal, cuyos datos están almacenados en la MM cache en lugar de utilizar una dirección especifica para acceder a una posición de la MM cache, se emplea parte de la dirección referida a la MP. Este valor se compara con cada etiqueta. Si se produce una coincidencia, entonces el dato buscado está en la cache.
- **La memoria cache o bloque de datos asociados SRAM:** es una MM pequeña y rápida usado para almacenar réplicas de instrucciones y datos. La información se agrupa en conjuntos de datos (línea).
- **Lógica de control:** son comparadores de distintos bits como tenga la etiqueta.

- **Organización de la MM cache (traducción de la dirección física):** como el tamaño de la DRAM no coincide con el de la cache, sus respectivos espacios de direccionamiento son distintos, por lo tanto, una dirección física desea ser traducida o mapeada por el controlador, que comprueba que exista tal referencia en la cache.

- **Clasificación según la organización:**
    1. Totalmente asociativa
    2. Asociativa de 1 vía (o de correspondencia directa).
    3. Asociativa de conjunto o de N vías.

MAPEO TOTALMENTE ASOCIATIVO

En este caso a cada línea le corresponde una etiqueta.

Esquema de cómo se analiza la dirección física: 

- Se la considera la mejor organización.
- Ante un fracaso la palabra se obtiene de la MM principal, siguiendo 2 caminos. Hacia la CPU y hacia la cache.
- Se puede organizar como un anillo.

Desventaja: debido a que hay una gran cantidad de etiquetas, la lógica de comparación es cara.

MAPEO ASOCIATIVO DE UNA VIA

- Es la organización interna menos utilizada actualmente.
- Es muy simple.

Ejemplo 1: Suponiendo una MP de 1MB y una cache de 4kb la cache se puede organizar en: 256 líneas de 16 bytes cada una.

(256L*16b = 4096 bytes = 4kb)

A su vez para representar a las “Mega Posiciones” se necesitan 20 bits (1 mega = 2^{20}) -> representados en hexadecimal

XXYYZ

Z: Identifica al número de byte. Sale de 16 bytes (va desde 0 a 2^4-1).

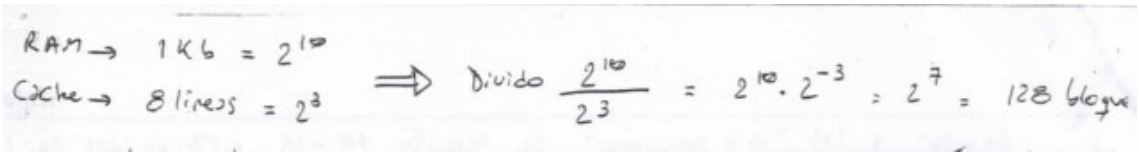
YY: Identifican al nro de bloque o líneas de cache, que equivalen al nro de sector de RAM. Sale de 256 líneas (desde 0 a 2^8-1).

XX: Etiqueta. El resto de los bits de orden superior (del total de 20 bits) se asignan a la etiqueta)

- De este modo hay una etiqueta por numero sector RAM o línea cache (en el ejemplo son 256 etiquetas).
- YYZ: se considera como la dirección de acceso (para el ejemplo 12 bits) 2^{12} =4k) -> la información buscada se encuentra en la cache – Si para YYZ la etiqueta asociada coincide con XX.

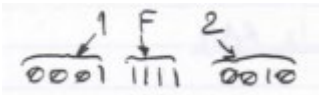
EQUIVALENCIA CON LA DIRECCION FISICA DE LA RAM

- Los 3 bits correspondientes a “línea” identifican a la línea de la cache y al sector de la RAM. EJ: si los 3 bits tienen valor 000 identifican a la línea 0 de la cache y al sector 0 de la RAM.
- Los 3 bits correspondientes a “posición” identifican al numero de byte dentro de la cache (para este ejemplo van desde 0 a 7). Pero este valor no identifica al numero de grupo o bloque de la RAM.



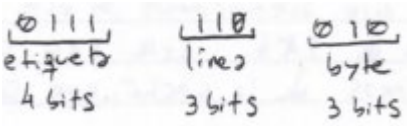
Ejemplo: se quiere acceder al dato correspondiente a la dirección 1 F2h. Se desea saber si el dato esta guardado o no en la cache, y en caso de que no lo este, actualizar la cache para que lo contenga.

Paso 1: Se pasa la dirección dada en hexadecimal a binario.

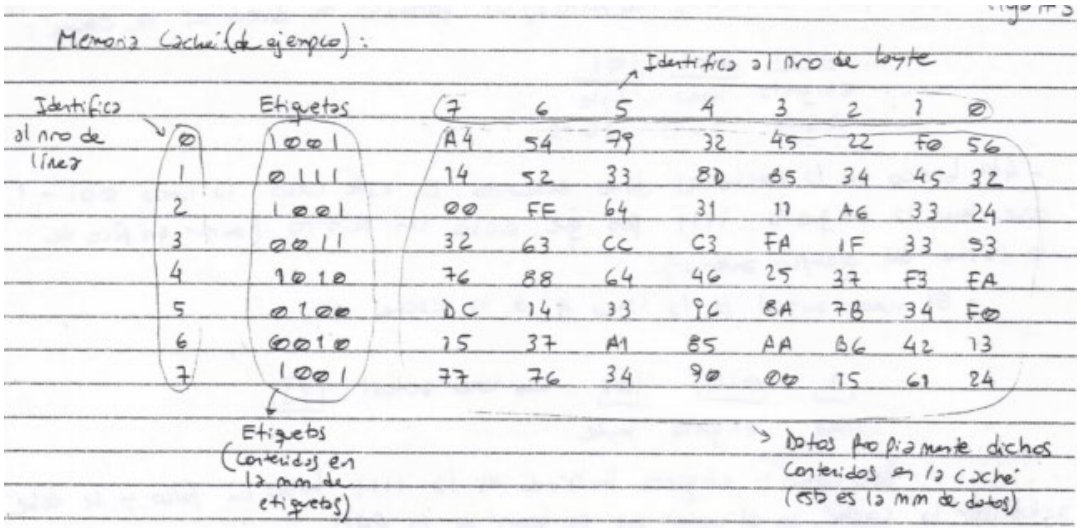


Paso 2: En el paso anterior se obtuvieron 12 bits, pero en este ejemplo las direcciones de cache son de 10 bits. Por lo tanto, se toman los 10 bits de orden inferior. Es decir: 0111110010.

Paso 3: Se mapean los 10 bits obtenidos en el punto anterior, al formato de dirección de la cache de la siguiente forma:



Paso 4: Se procede a buscar en la cache el dato deseado. Para esto se mira el valor correspondiente a la línea obtenida en el paso anterior (110=6). Para esta línea el valor de la etiqueta debe coincidir con el de la etiqueta de la cache (para que exista un acierto). Supóngase para ello la siguiente cache:



- En esta cache de ejemplo para la línea 6, la etiqueta es 0010 que no coincide con la etiqueta de la dirección (que es 0111).
- En este caso hubo un fallo y se debe acceder a la RAM a buscar el dato solicitado y a su vez actualizar la cache.
    - El valor actual de la cache en la línea 6 es:
      6 0010 010 -> valor actual: B6
      Línea etiqueta byte
    - Después de la sustitución por el valor leído de la RAM (suponiendo que sea A1), quedan así:
      6 0111 010 -> valor actualizado: A1
      Línea Etiqueta byte
      Actualizada

  Es decir que para la línea 6 se actualiza la etiqueta 7 el valor contenido en la Memoria para el byte 2.

MAPEO ASOCIATIVO DE N VIAS O N CONJUNTOS

- Es similar al mapeo directo, pero cada línea admite n etiquetas, matrices de datos. Esto implica una cache n veces mas grande y menor posibilidad de fallos (al buscar un dato en cache).
- Las N etiquetas y las n matrices de una misma línea constituyen un conjunto.
- Para buscar un dato en cache, dada una dirección, primero se busca la línea correspondiente, y luego se verifica la etiqueta. Que en este caso será n etiquetas para una misma línea.

PRACTICAS DE SUSTITUCION

Cuando hay un fallo, es decir, la palabra buscada no esta en la cache, se la debe incorporar a la misma. Para esto existen diferentes métodos para determinar el elemento que se reemplazara con el nuevo valor a incorporar.

- LRU (Least Recently Used): se reemplaza la de uso menos reciente.
- FIFO (First IN First Out): Se reemplaza la palabra que primero se escribió.
- RMD (Random): se reemplaza una palabra en forma aleatoria.

Niveles de cache:

- Cache de nivel 1 (primaria, L1): Es una memoria integrada a la CPU. Es la MM más rápida de un ordenador, suele funcionar a la mima velocidad que el microprocesador y es de tamaño pequeño.
- Cache nivel 2 (secundaria, L2): Es ligeramente más lenta que la cache L1 y de mayor tamaño. Se encarga, generalmente, de almacenar aquellos datos e instrucciones muy usados recientemente, pero que no han sido guardadas por la cache L1. Se la puede encontrar en la placa base o integrada en la CPU. Aun en este caso se la considera diferente a la L1 de la que permanece separada.
- Cache nivel 3 (L3): Como muchos procesadores integran parte de la cache L2 en la CPU, a la parte que permanece en la placa base a menudo se la llama.

CAPITULO 8

MEMORIA SEGMENTADA

Organización de la memoria

La memoria que controla en Pentium está organizada en:

- Bytes
- Palabras: se almacenan 2 bytes
- Dobles palabras: ocupan 4 bytes consecutivos
- Cuadruples palabras: 8 bytes

las palabras se encuentran en dirección par y las dobles palabras en direcciones múltiplos de cuatro.

Además de los tipos básicos de estructuras de datos mencionadas, el Pentium maneja otros dos mucho más complejos:

- Segmentos: bloques de memoria de tamaño variable, que contienen información de la misma clase y constituyen el objeto principal sobre el que se basa el mecanismo de protección
- Paginas: La paginación divide el espacio de memoria en trozos de longitud fija, llamados páginas, que, en el caso del Pentium, tienen un tamaño de 4 KB ó 4 MB.

El Pentium es capaz de combinar ambos métodos cuando funciona con segmentación paginada. De esta forma, el programador de aplicaciones estructura la memoria lógica en segmentos que soportan la multitarea, mientras el programador de sistemas emplea la paginación en el manejo y transferencia de bloques en la memoria física.

La memoria en modo real:

Se encuentran:

- 8 registros de propósito general de 16 bits (AX,BX,DX,SP,BP,SI,DI)

- Registros extendidos de 32 bits (EAX, EBX, ECX, EDX, ESP, EBP, ESI y EDI)
- Registros de segmento (CS, DS, SS, ES, FS y GS)

En este modo se contempla un espacio de direcciones directamente accesibles por la CPU de 1MB, en el que sólo es posible aplicar la técnica de la segmentación. Se multiplica el contenido del registro segmento por 16 para hallar la base del mismo y luego para calcular la dirección a acceder, se suma al resultado obtenido

DIRECCIÓN EFECTIVA = RS x 16 + DESPLAZAMIENTO

Para direccionar una instrucción en el segmento de código, se usa como registro de segmento a CS y como desplazamiento el contenido de IP

Aunque en Modo Real el Pentium manipula, por defecto, operandos y desplazamientos de 16 bits, es posible usar elementos de 32 bits de longitud.

La memoria en modo protegido:

Dentro de la memoria contemplada por el Pentium, se pueden distinguir tres espacios:

- Espacio virtual o lógico
- Espacio lineal
- Espacio físico

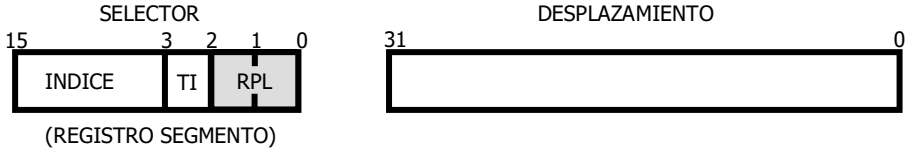
El espacio virtual abarca toda la dimensión de la memoria virtual y es el que maneja el programador de aplicaciones. Como la dirección virtual es de 46 bits el tamaño del espacio virtual (memoria de masa o disco) es de 246 =64 TB.
La Unidad de Segmentación, cuya activación siempre es obligada, traduce las direcciones virtuales a lineales, que reciben este nombre porque hacen referencia a segmentos que, al situarse sobre la memoria física, tienen dispuestas todas sus posiciones en orden consecutivo o lineal.
Cuando la Unidad de Paginación, optativa, no está activada, la dirección lineal coincide con la dirección física, es decir, la de acceso a la memoria principal ligada directamente a la CPU. Si la Unidad de Paginación está activada, cada segmento se descompone en un número variable de páginas del mismo tamaño (4KB ó 4MB), y la Unidad de Paginación deposita a dichas páginas sobre la memoria física en los huecos que encuentra libres, no una detrás de otra, lo que significa que la dirección lineal se debe traducir a física de acuerdo con esta distribución aleatoria de las páginas.

Los programas de aplicaciones que procesa el computador sólo hacen referencia a direcciones virtuales y es la MMU la que se encarga de traducirlas a direcciones físicas. Como la memoria física es mucho más pequeña que la virtual, la MMU también deberá detectar la ausencia de los elementos que no estén cargados en la memoria física

El espacio virtual o lógico:

La dirección lógica que usan los programadores consta de:

- *Selector*: Campo de 14 bits que selecciona un determinado segmento del espacio virtual. Este campo está contenido en los 14 bits de más peso del registro de segmento, sirviendo los 2 restantes para referenciar al nivel de privilegio.
- *Desplazamiento*: Campo de 32 bits que determina una posición del segmento (tamaño máximo 4 GB). También puede ser de 16 bits para mantener compatibilidad (máx.: 64 KB).



  El campo TI (Indicador de tabla) indica si la tabla de descriptores a acceder es la local (TI=1) o global (TI=0).

El desplazamiento se suma a la base del segmento para hallar la dirección a acceder.

- Cuando se accede a código: El CS actúa como selector y el EIP como desplazamiento.
- Si se accede a la pila: El SS hace de selector y el ESP de desplazamiento.
- Cuando se accede a datos: El DS, ES, FS o GS actúa como selector y el desplazamiento se calcula de acuerdo con el modo de direccionamiento utilizando la instrucción en curso.

La MMU recoge una dirección virtual de 46 bits y la introduce en la Unidad de Segmentación que contiene las Tablas de Descriptores de Segmentos que determinan los segmentos que están en MP y su posición. Si el segmento solicitado esta en MP, se traduce la dir. virtual a dirección lineal de 32 bits; Si no se encuentra en MP, el SO inicia una excepción que traslada el segmento de MV a MP y actualiza las Tablas de Descriptores.
Si la paginación esta activada, la dirección lineal pasa a la Unidad de Paginación que contiene la Tabla de Páginas que indica si las páginas esta en MP y su ubicación. Si la pág. esta en MP, la dir. lineal se traduce a dirección física de 32 bits. Si no está en MP, se produce una excepción que atiende el SO realizando la transferencia de la correspondiente pág. a MP actualizando las tablas.

El espacio lineal

La Unidad de Segmentación siempre se halla activada en el Pentium y se encarga de traducir la dirección lógica de 46 bits en dirección lineal de 32 bits. La dirección lineal coincide con la física correspondiente a la memoria principal si la Unidad de Paginación no está activada.

Los objetos con los que opera la Unidad de Segmentación son los segmentos, por lo tanto en la memoria principal sitúa y mueve segmentos completos. De ahí proviene el nombre de dirección lineal, que significa que la segmentación referencia a bloques (segmentos) que tienen todas sus posiciones ordenadas consecutiva o linealmente.

Cuando la dirección virtual hace referencia a un segmento que no está cargado en la memoria principal la Unidad de Segmentación detecta su ausencia. En esta situación, provoca una excepción, que pone en marcha una rutina del Sistema Operativo que se encarga de trasladar dicho segmento desde la memoria virtual a la memoria física

Descriptores de segmento

En el modo protegido un segmento queda especificado por tres parámetros: Base de 32 bits, Límite o tamaño de 20 bits y Atributos de 12 bits. Al conjunto de estas informaciones se llama "Descriptor" que es una estructura de 8 bytes.

- **Base (32 bits):** contiene la dirección lineal donde comienza el segmento
- **Limite (20 bits):** Expresa el tamaño del segmento. Su forma de expresión depende de un campo de los atributos, donde el tamaño máximo puede ser 1 MB o 4 GB.
- **Atributo o derecho de acceso (12 bits):** proporciona las características relevantes del segmento

función de diversos bits que componen el campo de atributos:

- **Bit de presencia (P):** Indica si el segmento al que referencia el descriptor está cargado, o sea, se halla presente en la memoria principal (P=1), o bien, está ausente (P=0).
- **Nivel de privilegio (DPL**): Compuesto por 2 bits, puede variar entre 0 y 3.
- **Tipo de segmento (S):** S=1 Segmento normal (código, pila o datos), S=0 segmento del sistema (manejados por el programador de sistemas o el SO).
- **Tipo:** Distingue entre los segmentos normales si es de código, datos o pila y además determina el acceso permitido (lectura/escritura/ejecución). Campo de 3 bits
- **Accedido (A):** Se pone automáticamente en 1 cada vez que el procesador accede al segmento. El SO lleva la cuenta de veces que es accedido cada segmento para poder implementar algoritmos como el LRU.
- **Granularidad (G):** G=0 indica que el limite (tamaño del segmento) esta expresado en bytes (máx. tamaño de segmento: 1 MB), G=1 indica que esta expreso en paginas de 4 KB (máx. tamaño de segmento: 4 GB).
- **Defecto/Grande (D/B):** Permite distinguir los segmentos nativos de 32 bits para el Pentium (para compatibilidad). D=1 direcciones efectivas y operandos de 32 bits. D=0 de 16 bits.
- **Disponible (AVL):** Bit a disposición del usuario para poder diferenciar ciertos segmentos que contengan un tipo determinado de información o que cubran alguna función específica.

Tipos de segmentos normales:

- Segmento de código:
  1. Solo ejecutables
  2. Ejecutables y leíble
  3. Ajustable o conforming
- Segmento de datos:
  1. Se puede leer y escribir
  2. Solo de puede leer
  3. Es un segmento de pila

De los tres bits que componen el campo TIPO. El bit de más peso E (Ejecutable), diferencia los segmentos de código (E=1), de los segmentos de datos que no se pueden ejecutar (E=0).

En caso de que E=1. Los otros dos bits de este campo tienen el siguiente cometido:

- Ajustable o conforming (C): C=0, no cambia su nivel de privilegio. C=1, se llama segmento ajustable.
- Leible (R)

En caso de E=0. Los otros dos bits del campo TIPO tienen el siguiente significado:

- Expansión decresiente (ED)
- Escribible (W)

Manejo de los descriptores

A los descriptores de los segmentos sólo los maneja el procesador automáticamente.
Los descriptores están agrupados en Tablas disponibles en la memoria principal. Cuando en un programa se desea acceder a un nuevo segmento se ejecuta una instrucción.

Por ejemplo si se quiere cambiar de segmento de código se ejecuta una " jmp CS':DESPLAZAMIENTO ". Dicha instrucción carga el valor de CS' en el registro de segmento CS. Inmediatamente que se modifica un valor de cualquiera de los seis registros de segmento (CS, SS, DS, ES, FS y GS) el procesador toma como puntero los 14 bits de más peso de dicho registro y direcciona una entrada de la Tabla de descriptores de segmentos. El contenido de dicha entrada que son los 8 bytes de un descriptor los carga en un registro caché ultrarápido de 64 bits asociado al registro de segmento modificado y a partir de ese momento el procesador toma los valores de la Base, Límite y Atributos del descriptor cargado en el registro oculto para direccionar el segmento. En el caso de ser un segmento de código el desplazamiento se carga en el puntero de instrucciones EIP y su valor se suma al de la Base para determinar la siguiente instrucción a ejecutar

Tabla de descriptores

En los Pentium se trabaja en un ambiente multitarea. Un sistema multitarea se compone de un área global, en la que residen los segmentos comunes a todas las tareas y de un área local exclusiva de cada tarea
Cada segmento del área global está definido por un descriptor, existiendo una tabla, llamada **Tabla de descriptores globales (GDT)**
Tabla de descriptores locales (LDT): tabla para cada tarea, que recoge todos los descriptores de los segmentos de cada una de ellas
En un momento determinado el Pentium estará ejecutando una tarea concreta y tendrá activas a la GDT y a la LDT correspondiente a la tarea en curso.
Las Tablas de Descriptores tienen un tamaño máximo de 64 KB y contiene un máximo de 8 K descriptores.
Bit TI: Como la CPU tiene activadas dos tablas de descriptores este bit indica a cual de ellas se refiere. Cuando es 1, selecciona a la LDT y cuando es 0 a la GDT.
Indice: 13 bits más significativos del selector el valor de ÍNDICE se multiplica por ocho para apuntar la dirección concreta de inicio del descriptor, puesto que cada uno consta de ocho bytes

El modelo plano

El mecanismo de segmentación es intrínseco al Pentium y no puede desactivarse. En aquellas aplicaciones y sistemas en los que no se use la segmentación se usa el procedimiento de modelo plano para simular su inhabilitación.
Se cargan todos los registros de segmento con selectores que apuntan en las tablas a

descriptores caracterizados porque el valor de su base es 00000000H y el límite FFFFFFFFH. De esta manera, la CPU sólo maneja un único segmento, que abarca todo el espacio lineal

CAPITULO 9

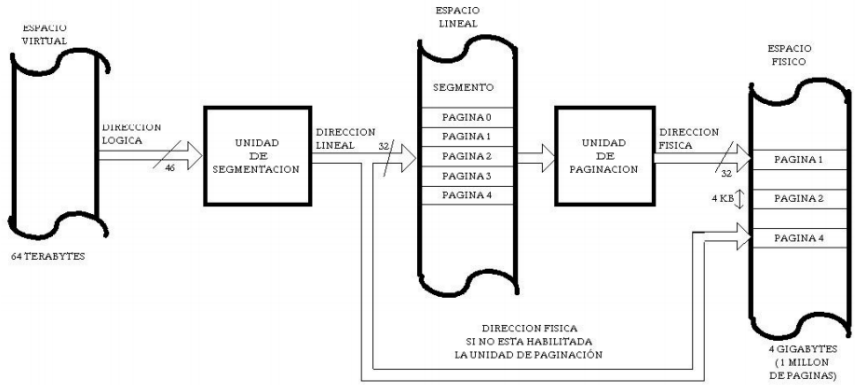
Mecanismo de paginación

Paginación: Divide y manipula los programas y los datos en trozos de tamaño fijo, llamados páginas. El procesador Pentium siempre trabaja con segmentación y optativamente puede trabajar además con paginación.

El funcionamiento de la paginación es optativo y para su habilitación basta con poner a 1 un bit (PG) de uno de los registros de control (CR0). se utiliza la instrucción: MOV CR0, FFFF. Como a dicho bit solo se le puede modificar en Modo Protegido, la paginación solo opera en dicho modo

se divide a cada segmento del espacio lineal en páginas sucesivas de 4 KB. Luego, la Unidad de Paginación carga y distribuye, de forma aleatoria, las páginas que se precisan en cada momento, sobre el espacio de la memoria física.

Los algoritmos usados en la transferencia de bloques desde/hacia la memoria principal, son mucho más sencillos y efectivos que en la segmentación, puesto que manipular bloques de tamaño fijo y reducido, optimiza el aprovechamiento del espacio de memoria
El Pentium utiliza un buffer (BTB) de predicción de bifurcaciones que evita la necesidad de utilizar instrucciones para eliminar las instrucciones que ya han sido buscadas y decodificadas como ocurría en los procesadores anteriores.



Para referenciar la base de la página bastan 20 bits
la Unidad de Paginación manejaría una tabla con un millón de entradas, conteniendo cada una la base (20 bits) y los derechos de acceso (12 bits) de cada página de la memoria principal. Luego si cada entrada consta de 32 bits de Tabla de Páginas tendría un tamaño de 1 M x 4 bytes = 4 MB

Tablas de páginas.

Como cada página pesa 4KB y la memoria principal tiene una capacidad máxima de 4 GB, caben en total **un millón de páginas** en la memoria principal. Cada entrada ocupa 4 bytes (32bits) por lo que la tabla de páginas pesa 4 MB. Debido a que para modelos anteriores como 386 4 MB en memoria era totalmente inaccesible física y comercialmente, Intel recurrió a una **traducción en 2 niveles**. El **primer nivel** estaba compuesto con un **directorio de tablas de páginas**, que contaba con 1024 entradas de **32 bits**, cada una apunta a una tabla de páginas, siendo estas la que soportan el **segundo nivel**, teniendo estas un 1024 entradas de 32 bits cada una.

Las entradas de tabla de páginas se ve caracterizada por:

- **Base**: los 20 bits de más peso.
- **Atributos**: los 12 bits de menos peso.

31-12	11-9	8	7	6	5	4	3	2	1	0
Dirección tabla de páginas (31-12)	—	0	0	D	A	0	0	U/S	R/W	P

Figura 9.3. *Distribución de los bits de una entrada del Directorio.*

El directorio es fijo para cada tarea, estando la base almacenada en el registro de control **CR3**. El acceso a una entrada del directorio de páginas se calcula así:

CR3 + (10 bits de más peso de la dirección lineal)*4

Las entradas de páginas tiene el mismo formato que las entradas de la tabla de página.

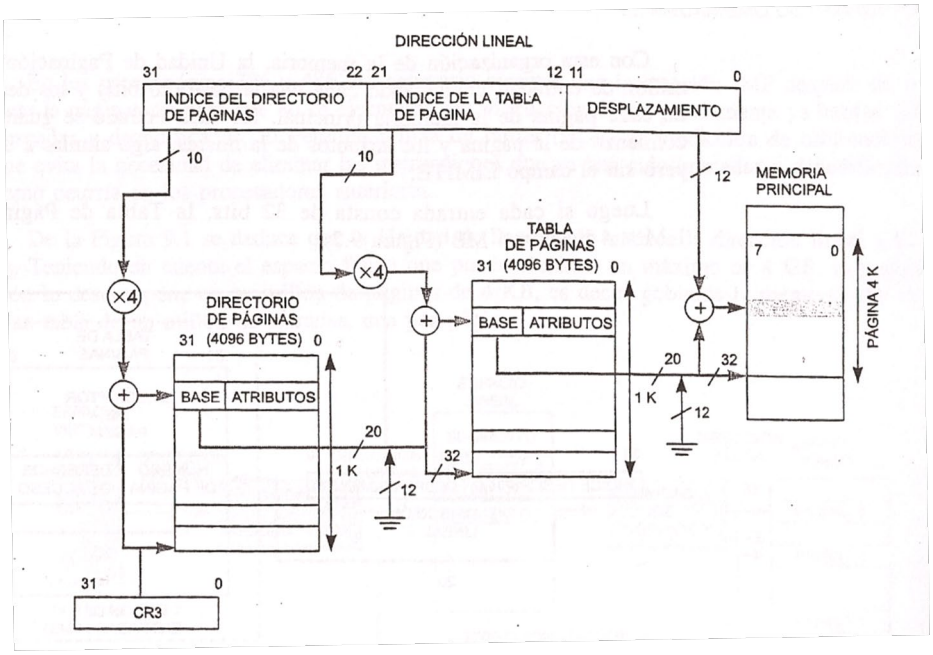
31-12	11-9	8	7	6	5	4	3	2	1	0
Dirección física de la página (31-12)	—	0	0	D	A	0	0	U/S	R/W	P

Figura 9.4. *Formato de una entrada de la Tabla de Páginas.*

Para poder situarse en una entrada de la tabla de páginas se calcula así:

Base + (10 bits del medio de la dirección lineal)*4

De esta manera accediendo a las entradas de tabla de páginas se puede **obtener la base y atributos** de los segmentos. Sumándole el **desplazamiento**, es decir los **12 bits** de menos peso de la dirección lineal, nos podemos posicionar sobre los elementos de la página.

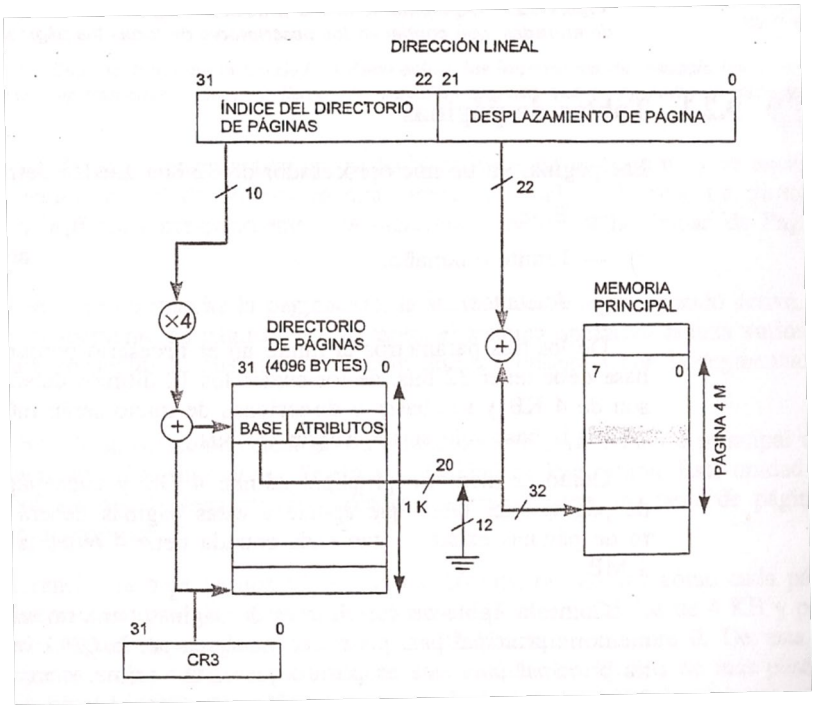


Cuando el Pentium maneja páginas de **4 MB** en la memoria solo residen **1024 páginas**, siendo la capacidad física de la memoria principal de 4 GB. El Pentium utiliza la traducción de un solo nivel, siendo soportado **por el directorio de páginas,** que posee la misma estructura que el directorio de tablas de páginas y de las tablas de páginas.

CR3 sigue siendo la base del directorio y para poder situarse en las entradas de la tabla se calcula así:

CR3 + (10 bits de mayor peso de la dirección lineal)*4

Obteniendo de la entradas la base y atributos de la pagina. Sumándole el **desplazamiento**, es decir los **22 bits** de menor peso de la dirección lineal podemos acceder a los elementos de la página de 4 MB.



Fallo de página.

Cada vez que la unidad de paginación detecta que la pagina no está en memoria, ocurrió un **fallo de pagina**, origina una excepción que atiende el SO, este se encarga de llevar dicha pagina a memoria principal.

Formato de las entradas del directorio y de las talas de páginas.

Como dijimos anteriormente son de 32 bits, de los cuales la base está compuesta de los 20 bits más significativos y los 12 bits de menor peso son atributos.

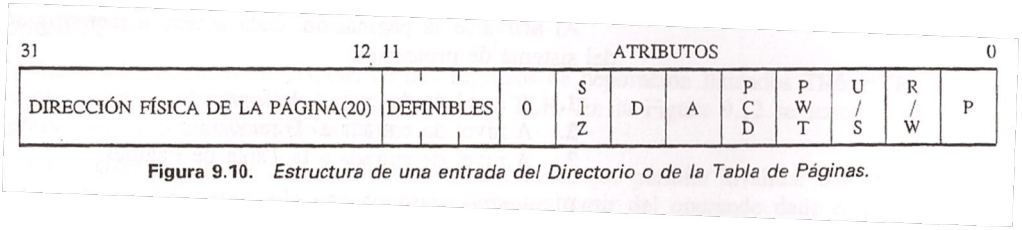


Figura 9.10. *Estructura de una entrada del Directorio o de la Tabla de Páginas.*

Atributos:

- **Bit de presencia (P):** indica con 1 que la página está cargada en memoria, con 0 que la página no reside en memoria.
- **Bit accedido(A):** se pone en 1 cada vez que la página es accedida. El SO lo utiliza para utilizar el algoritmo LRU.
- **SIZ:** "**PSE**", indica si trabaja con paginas de 4 KB (si esta en 1) o paginas de 4 MB (si esta en 0).

- **Bit sucio (D):** indica si la página fue escrita. El SO la utiliza para actualizar la memoria virtual en caso de que la página haya sido escrita y ahora se deba sobrescribir.
- **Bit de escritura/lectura(R/W):** indica que puede ser leída y escrita (si vale 1) o solo leída (si vale 0).
- **Bit usuario/supervisor (U/S):** indica el nivel de privilegio.
- **Bit de aceptación de la cache (PDC):** indica si la página puede ser cacheable, es decir metida en cache.
- **Bit de escritura obligada (PWT):** indica si la página además de ser cacheable funciona en modo de escritura obligada.
- **Definibles:** Son tres bits a disposición del S.O. que pueden usarse para guardar información auxiliar sobre la página.

Protección a nivel de páginas.

Solo existen 2 niveles de protección: **usuario y supervisor**. Siendo el nivel usuario en modo paginación equivalente al nivel 3 de la segmentación y el nivel supervisor equivalente a nivel 0 respectivamente.

Cada acceso a memoria soporta 3 niveles de protección:

- A nivel de descriptor de segmentos.
- A nivel de entrada al directorio.
- A nivel de entrada a la tabla de páginas.

Tabla de traducción de dirección lineal (TLB)
El gran inconveniente de la traducción 2 niveles es la lentitud, por este motivo Intel desarrollo para acelerar este proceso una cache especial ultrarrápida llamada **TLB**. La TLB es una cache de **acceso por contenido**, que contiene etiquetas con sus respectivos datos asociados. Las **direcciones lineales** hacen de **etiquetas**, mientras que los **datos asociados** son la **dirección física** correspondiente.

Estructura y funcionamiento de la TLB.

La TLB cuenta con **32 entradas**, en las cuales se guardan las **últimas 32 páginas** que ha manejado la CPU. La TLB ocupa un total de 128 KB. Cuando se quiere obtener información se consulta la TLB suministrando la dirección lineal, que se compara con las etiquetas respectivamente. Sabemos que como toda cache la TLB puede dar **acierto o fallo** cuando vamos a buscar un dato, el fallo se puede dar por dos condiciones:

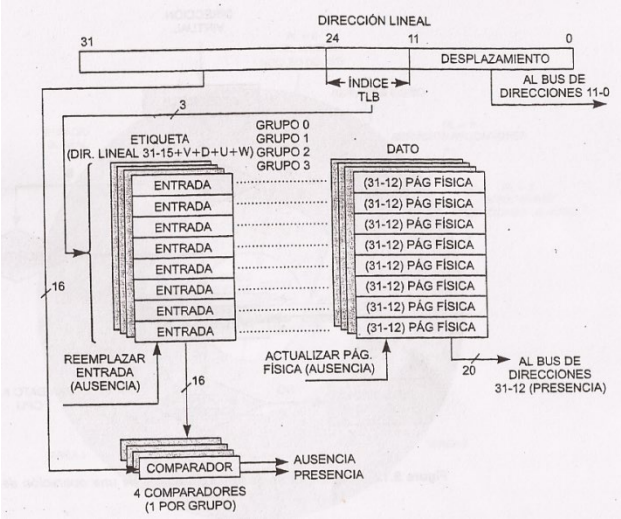
- La pagina esta en memoria pero no está referenciada en la TLB,
- La página no está en memoria. Se carga dicha página en memoria y luego se asocia a la entrada de la TLB.

En el **386** la TLB consta de de **4 grupos de 8** entradas cada uno, es decir es una **cache de 4 vías**. Con los bits **12, 13 y 14** de la dirección lineal se selecciona una de las 8 entradas en los 4 grupos de la TLB. La comparación de etiquetas se hace de forma paralela, dichas **entradas consta de 20 bits**:

- 16 bits de más peso de la dirección líneas.
- 1 bit de validez.

- 3 bits de atributos (U: usuario, D: sucio, W: escritura).

En el caso de que haya acierto, la CPU utiliza los 20 bits de más peso de los datos asociados a la etiqueta que acertó para sumarle el desplazamiento que son los 12 bits de menor speso de la dirección lineal.



Describa brevemente los componentes más importantes que constituyen la caché.

- Tamaño de la caché.
- Organización.
- Estructura física de una caché.
- Actualización de la caché.
- Actualización de la memoria principal.

CAPITULO 10

Mecanismo de protección

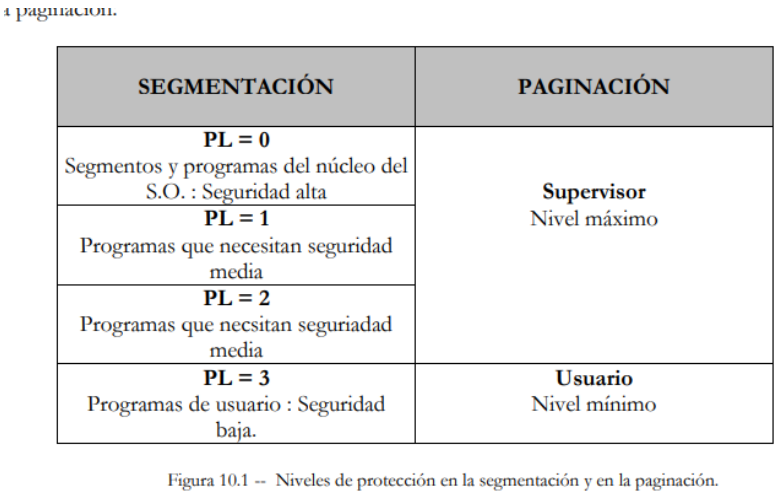
Necesidad de protección

en Modo Protegido, el procesador atiende a varias tareas simultáneamente. El Pentium dispone de un hardware auxiliar, integrado en el chip, que se encarga de comprobar el cumplimiento de unas reglas que conforman el llamado mecanismo de protección y así permitir la ejecución de todas las tareas sin interferencias.

El objetivo fundamental del mecanismo de protección es caracterizar y defender las funciones vitales del sistema de explotación ante las posibles intrusiones procedentes de las aplicaciones, pero sin excluir una comunicación controlada.

Cuando el mecanismo de protección detecta una violación de las reglas escritas en el silicio, genera una excepción, deteniendo el procesamiento normal de la CPU

En la siguiente tabla se refleja la relación entre los niveles de protección en la segmentación y en la paginación:



SEGMENTACIÓN	PAGINACIÓN
PL = 0 Segmentos y programas del núcleo del S.O. : Seguridad alta	**Supervisor** Nivel máximo
PL = 1 Programas que necesitan seguridad media	
PL = 2 Programas que necsitan seguriadad media	
PL = 3 Programas de usuario : Seguridad baja.	**Usuario** Nivel mínimo

Figura 10.1 -- Niveles de protección en la segmentación y en la paginación.

Unidad de segmentación: evalúa el cumplimiento de las reglas de acceso y manejo de los segmentos en primer lugar y si está habilitada la paginación, son examinadas las reglas que afectan a las páginas en la Unidad de Paginación seguidamente.

El mecanismo ideal de protección es el proporcionado conjuntamente por la segmentación y la paginación: segmentación para la protección y paginación para manejar la memoria.

El mecanismo de protección del Pentium cubre los siguientes niveles:
1. Protección de tareas
2. Protección de los segmentos
3. Protección de las páginas
4. Protección de las instrucciones

Activacion y desactivación de la protección de segmentos y paginas

Para que el procesador cambie a Modo Protegido y éste active el mecanismo de protección de segmentos, se debe activar el flag EP (enable protection) del registro de control CR0.
Asignando el nivel de protección 0 (mayor privilegio) a todos los segmentos y descriptores de segmento se pueden desactivar las reglas de protección de segmentos basadas en los niveles de privilegio
Si se quiere desactivar el mecanismo de protección en la paginación, se debe poner el flag WP del registro CR0 a 0. Posteriormente, se ponen a 1 los flags R/W (read/write) y U/S (usuario/supervisor) en el Directorio de Páginas y en la Tabla de Páginas.

Campo y flags usados para la protección entre segmentos y paginas

Campos que usa el mecanismo de protección para controlar el acceso a las paginas y segmentos:

- Flag tipo de segmento (S): Determina si el descriptor de segmento describe un segmento del sistema o un segmento normal de código o datos.
- Campo TIPO: Determina si el segmento es de código, de datos o del sistema.

- Campo del limite: Determina el tamaño del segmento, estando relacionado con el flag G y el flag E
- Flag G: Define el tamaño del segmento, estando relacionado con el límite del segmento y el flag E
- Flag E: Define el tamaño del segmento, estando relacionado con el límite del segmento y el flag G. Si esta en 1 es un segmento de código, si es 0 es un segmento de dato
- Campo descriptor del nivel de privilegio (DPL): Determina el nivel de privilegio del segmento
- Campo del nivel de privilegio del peticionario (RPL): bits 0 y 1 de cualquier selector de segmento.
- Campo de nivel de privilegio de la tarea en curso (CPL): Se refiere al procedimiento o programa que se está ejecutando en ese momento
- Bit de tamaño (SIZ): Si es 0 se trata de paginas de 4KB, si es 1 se trata de paginas de 4MB
- Bit sucio (D): Sirve para avisar al Sistema Operativo que antes de machacar la página hay que actualizarla en la Memoria Virtual.
- Bit A (Accedido): Cada vez que se accede a esa página el bit A se pone automáticamente a 1. Al final cuando se llena la memoria con páginas el S.O sustituye el que menos tiene en su contador.
- Bit PCD: Indica si está a 1 que se trata e una página cacheable
- Bit PWT: Indica si está a 1 que la página es de escritura obligada y cacheable.
- Usuario/supervisor (R/W): Indica el tipo de acceso permitido a esta página: sólo lectura o lectura y escritura
- Bit de presencia (P): Indica si está a 1 que la página está presente en la memoria principal y si está a 0 que no lo está.

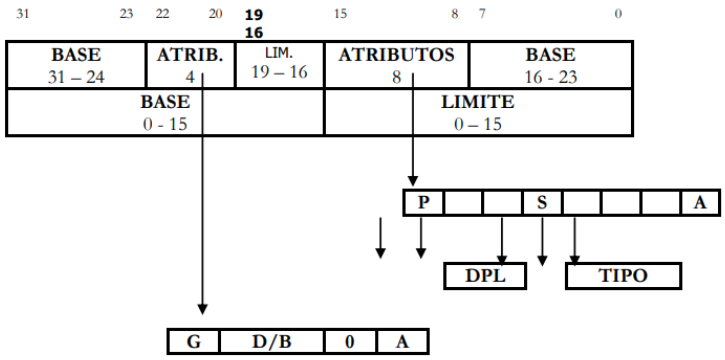


Figura 10.4 Estructura del descriptor de segmento

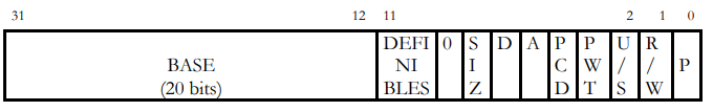


Figura 10.5 Formato de las entradas a las Tablas de Páginas

NIVELES DE PROTECCION

Proteccion entre tareas

Descriptores de segmento de la GDT: hacen referencia a objetos del espacio global
Descriptores de segmento de la LDT: hacen referencia a los segmentos de la tarea en curso

Proteccion de segmentos

Cuando se carga el contenido de un descriptor en el registro caché ultrarrápido asociado a un registro de segmento, se almacena la base (32 bits), el límite (20 bits) y los atributos (12 bits). Cada vez que se accede a memoria, la Unidad de Segmentación es la encargada de comprobar las normas de protección y en caso de fallo, se genera una excepción.

Proteccion del limite

Para llevar a cabo la comprobación del límite de los segmentos se debe tener en cuenta el flag de granularidad (G). Para segmentos de datos, también depende del flag ED (dirección de expansión descendente) y del B (tamaño por defecto del puntero a pila).

Para todos los segmentos, la última dirección a la que se puede acceder, es el tamaño del segmento menos 1. Cada vez que se intente acceder a las siguientes direcciones en un segmento, el procesador genera una excepción de protección general:

1. Un byte más arriba del límite efectivo
2. Una palabra más arriba del límite efectivo (-1)
3. Una doble palabra más arriba del límite efectivo (-3)
4. Una cuádruple palabra más arriba del límite efectivo (-7)

Para segmentos de Pila, el limite depende del valor de B. Si es 0 el rango va de a 4 KB y si es 1 el rango va de a 4 GB.

Proteccion del tipo

Para evitar el uso de un segmento o puerta de forma incorrecta se tiene en cuenta el tipo de los descriptores de segmento que viene determinado por el campo tipo y el flag S

<table>
<tr><td rowspan="2">S = 1</td><td>E = 1</td><td>C</td><td>R</td><td rowspan="2">A</td></tr>
<tr><td>E = 0</td><td>ED</td><td>W</td></tr>
</table>

S:

- Si es 1: Se trata de un segmento normal de código, datos o pila
- Si es 0: es un segmento especial creado por el programador de sistemas

E:

- Si es 1: El segmento es ejecutable o de código:
    - Los bits restantes componen el campo tipo C. Si C es 1, es ajustable.

    - El bit R indica si el segmento es leible
- Si es 0: Se trata de un segmento o ejecutable:
    - ED y W forman el campo tipo
    - ED: indica si el segmento tiene o no expansio de direcciones decrecientes
    - W: indica si el segmento es escribible

Las diferentes excepciones generadas por el mecanismo de protección, son las siguientes:

1. Si se intenta escribir en un segmento de código (E=1)
2. Si se intenta leer un segmento de código con R=0 (prohibición de lectura)
3. Si se intenta cargar CS (registro de segmento de código) con el valor de un selector que corresponda a un descriptor con E=0.
4. Si se intenta escribir un segmento de datos con W=0 (prohibición de escritura)
5. Si se intenta cargar SS (registro del segmento de pila) con el valor de un selector, cuyo descriptor asociado esté definido como no escribible (W=0).

Comprobación del selector del segmento nulo: Si se intenta cargar un selector de segmento nulo en CS o SS se provoca una excepción.

Proteccion según el nivel de privilegio

Los niveles de privilegio son usados por el procesador para evitar accesos indebidos a los mismos, como puede ser que un programa de un nivel bajo de seguridad acceda a los segmentos de niveles superiores. En caso de que esto ocurra, el procesador genera una excepción de protección general (GP).
Siempre que se quiera acceder a un segmento debe ser a través de una instrucción ejecutada en el segmento de código en curso, su nivel de privilegio se denomina Nivel de Privilegio en Curso (CPL).

Mediante la ejecución de una instrucción se realiza el acceso a cualquier segmento. Para realizar la selección del segmento en curso se hace mediante el selector cargado en CS: sus 13 bits de más peso actúan como índice de la tabla GDT o LDT, pudiendo así encontrar el descriptor del segmento de código cuyo nivel de privilegio es definido por el campo DPL.

El nivel de privilegio del segmento de código en curso recibe el nombre de Nivel de Privilegio en Curso (CPL) y a partir del valor del CPL se determinan las reglas de acceso a otros segmentos

Para controlar el acceso a los distintos tipos de segmentos existen unas reglas básicas:

1. **Acceso a segmentos de código:** Sólo se puede acceder a segmentos de código que tengan el mismo PL que el segmento de curso peticionario. Las instrucciones que permiten estas llamadas o saltos directos son JMP, CALL y RET. El CPL del segmento de código peticionado debe ser igual al DPL del segmento a acceder
2. **Regla de acceso a segmentos de datos:** Sólo está permitido el acceso desde segmentos de código con un PL a otros de datos de igual o menor PL. Para ello se usan instrucciones tipo MOV o similares
3. **Regla de acceso a segmento de pila:** Sólo se permite acceder a segmentos de pila con el mismo PL que el del segmento de código que los solicita.

El procesador debe tener en cuenta los siguientes tipos de niveles de privilegio para llevar a cabo todas las comprobaciones:

- **Nivel de privilegio actual (CPL)**
- **Nivel de privilegio del descriptor (DPL)**
- **Nivel de privilegio del peticionario (RPL)**

Acceso a segmentos con el bit ajustables a cero

Para accesos a segmentos de código con C = 0, el CPL del segmento que quiere hacer el acceso, tiene que ser igual del DPL del segmento de código destino, sino se daría excepción de protección general (GP).

Acceso a segmento con el bit ajustables a uno

Cuando se accede a segmentos de código con C = 0, el CPL de la rutina que hace la llamada, puede ser igual o menor que el DPL del segmento de código destino, sino se daría excepción de protección general (GP).

En el caso de C = 1, no se tiene en cuenta el RPL del selector de segmento del destino. En estos casos el DPL representa numéricamente el de menor PL que puede tener una rutina para poder hacer una llamada

Proteccion de las paginas

Las reglas de paginación se aplican tras aplicar las reglas en segmentación.
Para definir el acceso el único bit que se usa es el W/R. En caso que éste sea 0, puede leerse, si es 1 puede tanto leerse como escribirse.

Los niveles son: Usuario y supervisor

Puede establecerse una relación entre estos niveles y los de la segmentación:

Supervisor: equivale al los niveles 0, 1 o 2 en los que se realizan la ejecución de instrucciones y programador del sistema.

Usuario: equivale al nivel 3 se realizan las aplicaciones de usuario y del programador de aplicaciones. En este tipo de protección se tienen en cuenta dos reglas:

1. Desde una página ubicada en nivel Usuario sólo se puede acceder a páginas de dicho nivel.
2. Desde una página del nivel de Supervisor se puede acceder a todas.

Instrucciones protegidas

Son aquéllas que sólo pueden ejecutarse desde objetos de código situados en un nivel de privilegio igual o mayor que el indicado en el campo IOPL (Nivel de privilegio de E/S) del registro EFLAGS, éste es determinado por el programador del sistema.

Se incluyen las siguientes instrucciones para manejar las operaciones del espacio de E/S periféricos:

- IN: puerta por la que toma un valor para cargar el acumulador
- OUT: puerta en la que se deposita el valor del acumulador
- INS, OUTS: manejan una cadena de caracteres
- CLI: pone el flan de interrupción IF del registro de señalizadotes a 0

- STI: pone el flan de interrupción IF del registro de señalizadotes a 1

Instrucciones privilegiadas

Estas instrucciones controlan las funciones del sistema.

Sólo se pueden ejecutar desde el nivel de mayor de privilegio (CPL=0), sino se da una excepción de protección general (GP).

1. Instrucciones que pueden modificar el IOPL
   POPF: carga los 32 bits de la cima de la pila en el registro de flags afectando al IOPL
   RET: retorno de interrupción
2. Instrucciones que escriben los registros que controlan las tablas del sistema al operar en modo protegido
3. Instrucciones que afectan al contenido de la palabra de estado
4. Instrucción de paro HTL.

Lista de instrucciones privilegiadas:

- LGDT
- LLDT
- LTR
- LIDT
- MOV
- LMSW
- CLTS
- ……

MEMORIAS

Clasificación de memorias (es lo que va a tomar)

Tipos:

- EL modo de acceso a la unidad de información
- Las operaciones que aceptan por cada acceso
- La duración de la información en el soporte

1. Clasificacion según el modo de acceso a la unidad de info:

- Acceso aleatorio: un **componente de selección** habilita una palabra e inhabilita a las demás. Tiempo de acceso independiente del lugar fisico
- Acceso secuencial: para acceder a una unidad de información se establece un posicion de referencia, a partir de la cual comienza un rastreo de la unidad de información que consiste en la lectura de todas las unidades que la precedan, hasta lograr la búsqueda. El tiempo de acceso depende de la distancia entre la posición inicial y la unidad de información

- Acceso asociativo: la búsqueda de la unidad de información implica la comparación de un grupo de bits de la unidad de información con el contenido de una posición de memoria

2. Clasificacion según las operaciones que aceptan por cada acceso
   - Lectura/escritura (“vivas”)
   - Solo de escritura (“muertas”)

3. Clasificacion según la duración de la informacion

- Volátiles: pierden su información con el corte de suministros de corrientes
- Perennes, permanentes, no volátiles

Dimension de memoria:

Capacidad de memoria: cantidad de info que se puede almacenar en ella

- Bit
- BYTE
- KB: 1024 bytes 1KB= 2^{10} bytes
- MB: 1024K 1MB= 2^{20} bytes
- GB: 2024 MB 1gb= 2^{30} bytes
- TB: 1024 gb 1TB= 2^{40} bytes

RAM estatica y dinámica

Las SRAM son memorias vivas, volátiles y estáticas. Cada celda es un elemento biestable

diseñado con compuertas.

Las DRAM son memorias vivas, volátiles y dinámicas. Estas degradan su información con el transcurso del tiempo

Cada celda almacena un 1 que se representa con la carga de un condensador, antes de que la información se pierda hay que restablecer la carga, se denomina ciclo de refresco (refresh cycle), esto debe estar cargo del controlador de memoria. Son memorias mas lentas que las SRAM, pero tienen mayor capacidad.

RAM de acceso directo

Acceso a la información en forma random o al azar una memoria se organiza de manera matricial en filas y columnas. El número que identifica la palabra en un acceso random o al azar se denomina dirección física, y representa en realidad el número ordinal que le corresponde dentro de la matriz, comienzo 0 hasta p-1.

<u>En el caso ejemplo de una RAM estática lo mas importante es:</u>

- La línea WE indica que con 1 en esta línea se da una orden de escritura, si es 0 es una orden de lectura.
- La línea EN indica con un 1 que este chip se habilito para su acceso.

Biestable asociada a una matriz:

La memoria estática está constituida por biestables.

Cada uno de estos tiene 2 salidas, una para el valor normal del bit almacenado, que

llamaremos Q y la otra que es el complemento no Q.

Una celda SRAM tiene 3 estados posibles:

- Reposo
- Lectura
- Escritura

Ram con acceso directo: Las memorias asociativas son accesibles por contenido, el contenido buscado se denomina rotulo, descriptor o segmento. La posibilidad de asociación requiere que todas las celdas de almacenamiento se relacionen con circuitos que permitan la comparación, razón por la que se vuelven más caras y su uso se justifica en aplicaciones en las que sea imprescindible la búsqueda por contenido

Jerarquia de memorias

La determinación de una jerarquía de memoria está dada básicamente por tres atributos:

- Velocidad de acceso
- Costo de la celda
- Capacidad de almacenamiento

De acuerdo a la jerarquía las podemos clasificar en:

- Primer Nivel: Registros internos del procesador, denominados registros de propósito general.
- Segundo y Tercer Nivel: Soportes de almacenamiento temporal de instrucciones y datos intercambiables, a los que accede el microprocesador en forma directa.

Estas se clasifican en 2 dos tipos:

- Memoria cache: Es una memoria de semi conductores, más rápida que la DRAM, de mayor complejidad por lo tanto de menor capacidad, su velocidad de respuesta se adapta a las exigencias del procesador.
- Memoria DRAM: Es una memoria de semi conductores lenta, menor complejidad, mayor capacidad de memoria.
- Ultimo Nivel: Memorias auxiliares.

MEMORIA CACHE

Es de tipo estático, su velocidad de respuesta se ajusta a los tiempos del procesador. La cache se usa como memoria intermedia entre el procesador y la DRAM, almacena en forma temporal la info a la que se accede con mayor frecuencia en esta última.

En la memoria de etiquetas o tags se almacenan las referencias de memoria principal

asociadas a cada bloque.

El cerebro de una memoria cache es el controlador de cache.

Una MC está constituida por una memoria de etiquetas, una memoria de datos y un

controlador.

El controlador se utiliza para gestionar su actividad.

Principios de funcionamiento

La comunicación entre el procesador-RAM es de forma continua, ya que el micro busca en el MP la instrucción para ejecutarla o cuando busca un dato que requiera la ejecución de una instrucción.

La comunicación entre el procesador-RAM se estable por medio del bus de direcciones y el bus de datos.

En cualquiera de los tipos de conexión enunciadas, el controlador de cache debe capturar la dirección para verificar si se puede ofrecer al procesador su contenido. La forma en que se captura la dirección depende del tipo de organización.

Caching

Procedimiento que gestionado por el controlador, anticipa las necesidades de posiciones de memoria principal de acuerdo con cierto cálculo de probabilidad de uso y utilizando criterios que consideran los principios de vecindad espacial y temporal.

El rendimiento depende tanto de la efectividad de la gestión de caching como de su tamaño.

Unidad 10

Instrucciones

Formato de instrucción

Manera en que deben interpretarse los bits que constituyen el código de máquina de la instrucción

Instrucciones sin dirección

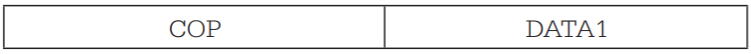
Está constituida por un grupo de bits que deben interpretarse como una unidad. Ésta representa únicamente al código de operación, esto es, cómo va a ser ejecutada después de haberse leído de memoria y colocada en la unidad de decodificación y secuenciamiento de la CPU.

Ejemplos: HLT, RET “C3”

Cuando una instrucción afecta un dato, éste puede estar implícito en el código de operación.
Ejemplo: LAHF

Instrucciones de una sola dirección

todas las instrucciones hacen referencia implícita, en el código de operación, al registro acumulador; por lo tanto, DATA1 hace referencia al dato afectado

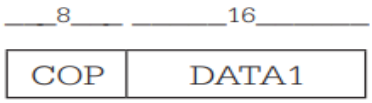


- Dato incluido en una instrucción

Cuando una instrucción afecta un dato y éste se encuentra incluido en el código de la instrucción, el formato cambia, o sea que debe incluir un grupo de bits que referencia al dato, que llamaremos en forma genérica campo DATA1.

Ejemplo:
MOV AX, 000h (transferencia del dato cuyo valor en Hexa es 000 al registro acum AX)

MOV BX, 000h



- Dato referido por la instrucción

Instrucción en la que el dato debe buscarse en la memoria principal

Ejemplo:

MOV AX, [FFFF]

esta vez el dato se encuentra en la memoria y el código de la instrucción hace referencia a su posición en ella.

Instrucciones de dos direcciones

Se agrega el campo DATA2, de modo tal que se utilice para referenciar el segundo operando. A su vez, referencia el lugar donde se almacena el resultado.

Ejemplo:
SUME 100 A02
Equivale a :
LDA 100
ADA A02
STA A02

Instrucciones de tres direcciones

se agrega el campo DATA3 para referenciar el lugar donde se almacena el resultado.

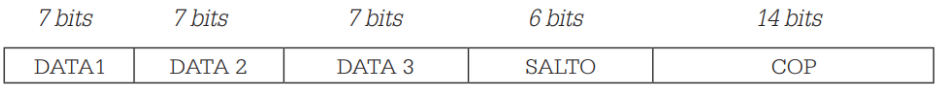
Instrucciones de cuatro direcciones

DATA1 y DATA2: datos afectados por el código de operación
DATA3: referencia el resultado
DATA4: referencia la próxima instrucción por efectuarse

- ## Instrucción de formato fijo en silabas

  Silaba: agrupación de 128 bits que el procesador puede capturar por vez. Cada una de estas estructuras contiene tres instrucciones de 41 bits y un campo de 5 bits denominado plantilla o template

  EPIC: las tres instrucciones podrían ejecutarse en forma simultánea, utilizando una unidad de ejecución diferente



- ## Interpretacion de un código de maquina

  Las instrucciones se implementan en lenguaje de máquina identificando con "campos de bits" los siguientes parámetros:

  - Dónde se ubica la identificación del registro fuente y el registro destino.
  - El tamaño de los datos
  - El tamaño del desplazamiento
  - Los registros de base e índice que se utilizan

Modo de direccionamientos

El campo de referencia a dato, que hasta ahora hemos llamado DATA, no va a ser considerado de la misma forma, sino que su interpretación depende del valor en bits que determine una de varias técnicas para calcular la dirección efectiva (o física) del dato. A cada una de estas técnicas se la conoce como modos de direccionamiento.

- Direccionamiento directo de memoria

la referencia al dato que especifica el campo DATA queda sin alteraciones y el tiempo de captación del dato depende sólo del tiempo de un acceso a memoria.

Ejemplo: MOV AH, [0200] (los corchetes indican que el dato se encuentra "contenido" en la referencia 0200 de memoria y debe transferirse al registro AH)

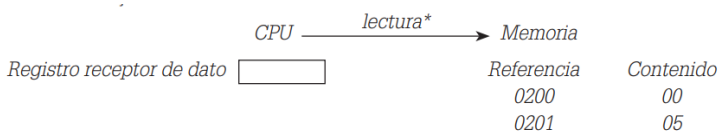
hay dos modos que determinan que no se acceda a memoria para obtener el dato:

1) Direccionamiento implicito: el dato queda determinado por el mismo verbo y, en consecuencia, en el campo código de operación. Todas las instrucciones que asignen un valor por medio del COP pertenecen a esta categoría
2) Direccionamiento inmediato: involucran el dato en la instrucción en sí, pero ahora no en el campo código de operación sino en el campo DATA. Ejemplo: MOV AH,05 (el numero 5 forma pare de la instrucción)

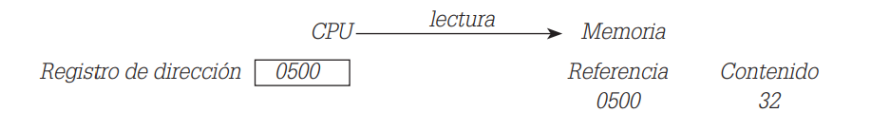
- Direccionamiento indirecto de memoria

En este modo el campo DATA contiene una dirección de una posición de memoria que, a su vez, contiene la referencia al dato. Para acceder al dato, se requieren dos accesos a memoria.

Ejemplo:

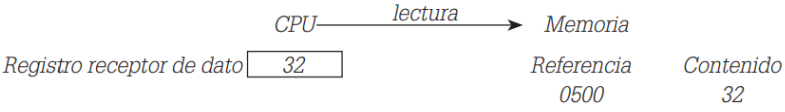


Como la referencia es de dos bytes, la CPU considera el contenido de 0200 y del siguiente byte 0201; para "construir" la nueva referencia, lee el primer byte lo acomoda a la derecha del registro de dirección y luego el segundo byte a su izquierda:



Con esta nueva referencia ordena una nueva lectura y obtiene el dato, que es el valor 32.

Después de la ejecución



- Direccionamiento de la CPU asociado a registros

la CPU cuenta con varios registros internos. Las instrucciones pueden hacer referencia a estas memorias locales que son los registros y optimizar así el tiempo de ejecución de los programas (REGISTROS GENERALES).
Si los registros generales están conectados entre si por conexiones directas con compuertas de habilitación entonces se denominan Registros independientes

- Direccionamiento directo por registro

no hay acceso a memoria y se observa una ganancia de tiempo en el acceso al dato. Ejemplo: MOV CX, DX

- Direccionamiento indexado

se utilizan algoritmos que involucran índices. En CPU, se aloja un registro que lleva el control del índice, ganando velocidad y simplificando las tareas. Este registro se llama Idex Register

- Direccionamiento relativo a la base

Se usan dos registros: BX (base) y BP (base pointer)

- Direccionamiento a una pila

La CPU puede contar entre sus registros internos con un puntero a una pila de datos en memoria. Una pila es una estructura de datos a los que se accede según el criterio LIFO. El que lleva el control de las direcciones en la pila es el SP (stack pointer). Su contenido direcciona la primera posición vacía de la pila y se actualiza por cada dato agregado y cada dato extraído.

La pila es el lugar utilizado para almacenar en forma temporal el contenido de los registros de la CPU.

Es aconsejable utilizar el BP para acceder a los datos en la pila. Por ejemplo, si se quiere armar una pila de datos en memoria, es aconsejable copiar el SP al BP y luego para direccionar la pila usar el BP más un desplazamiento para acceder a cualquier palabra de la pila. De esta manera, el SP siempre estará apuntando a la última palabra utilizada del segmento de pila, sin modificar su criterio LIFO. Ejemplos de modos de direccionamiento:

- Referencia a registro: el operando esta en la direccion del registro especificado en la instrucción
- Referencia a memoria: para acceder al operando se deben sumar “hipotéticamente” las cantidades: “direcc. De segmento + direcc base + índice + desplazamiento”
  Donde:
  1) La dirección del segmento se encuentra en el registro de segmento y se multiplica por 16 antes de sumar
  2) La dirección base almacena el registro base (BX o BP).
  3) El índice se encuentra en el registro índice (SI o DI).
  4) El desplazamiento puede ser de 16 bits, 8 bits o 0 bit.

La base y el índice son datos que se pueden modificar durante el procesamiento, ya que se encuentran en registros de propósito general de la CPU

Tipos validos de instrucciones

Los tipos de transferencias válidas son:

- Constante a registro MOV AH, 1234h
- Constante a memoria MOV [0200], 6789h
- Registro a registro MOV AX, BX
- Registro a memoria MOV [012A], AX
- Memoria a registro MOV AX, [0203]

  - Los operandos fuente y destino deben ser del mismo tamaño.
  - Las constantes y los registros de segmento no pueden moverse a registros de segmento.
  - Los operandos fuente y destino no puede ser ambas posiciones de memoria.

Unidad 11

Software del sistema

Clasificacion del software de sistemas

El software de sistemas es el nexo entre las necesidades del usurario y las capacidades del hardware. Esta integrado por:
Software de base:

- Controla y respalda el software de otras categorías y todas ellas están íntimamente relacionadas en mayor o menor grado con el diseño del software
- Su nucleo se denomina Sistema operativo
- La interfaz grafica de usuario forma parte de el y su objetivo principal es crear un entorno organizado para el usuario

Sistemas operativos

Es una colección de programas que administran la operación de una (o varias) computadora/as con el fin de obtener un comportamiento eficiente. Es una plataforma software que asigna recursos y supervisa al resto de los programas que se ejecutan en la computadora.

Hay dos grandes componentes software en el:

- Residentes: componentes que residen permanentemente en memoria principal durante todo el procesamiento
- Transitorios: residen solo cuando se los necesitan y están almacenados en memorias secundarias cuando no están en memoria principal

Niveles de administración del sistema operativo

- Administracion del procesador y los procesos

- Administracion de memoria
- Administracion de dispositivos de E/S
- Administracion de archivos:
    - Supervisa la gestión de archivos para su creación, acceso y eliminación
    - Define la política que determina de que forma serán almacenadas físicamente

Tipos de sistemas operativos

- Multitarea:
    - Son capaces de administrar procesos concurrentes
    - Permiten que tanto las instrucciones como los datos procedentes de varios programas residan al mismo tiempo en la memoria principal y eventualmente en el disco
- Tiempo compartido: tratan de administrar los recursos repartiéndolos equitativamente
- Multiusuario: permiten el acceso de varios usuarios desde distintas terminales administradas por el mismo SO
- Tiempo real:
    - Tienen como objetivo proporcionar tiempos mas rapidos de respuesta
    - Sus acciones se deben ejecutar en intervalos de tiempo determinados por la dinámica de los sistemas físicos que controlan

Clasificacion de los sistemas operativos (ANGULO)

1) Según el punto de vista del usuario
    - Sistemas monousuarios: la máquina virtual tiene un solo usuario y generalmente está dedicada a una sola función
    - Sistemas de tiempo real: tienen la necesidad de dar respuesta a unas entradas con un tiempo de proceso informático limitado
    - Sistemas transaccionales: sus funciones son gestionar un gran volumen de info desde distintos y numerosos puntos de acceso y de transacciones desarrollándose simultáneamente
    - Sistemas TS/ multiprogramados:
      Time sharing: se caracterizan por prestar servicio a un conjunto de usuarios simultáneamente
      Multiprogramados: ejecución simultanea de varios programas
    - Sistemas multiprocesados: se caracterizan por la existencia de varios procesadores centrales compartiendo a veces memoria y periféricos en la ejecución de instrucciones en paralelo
2) Según la arquitectura del computador
    - Sistemas monolíticos: están diseñados como un conjunto de rutinas compiladas por separado, que pueden llamarse entre si y se montan para formar el S.O
    - Sistemas por niveles o capas: los sistemas operativos están diseñados como una jerarquía de niveles, en la que cada nivel hace uso de las facilidades que le proporcionan el hard y los niveles inferiores a él
    - Sistemas de maquinas virtual: se separa la parte del S.O. que proporciona la máquina virtual de la parte que posibilita la compartición de recursos.

- Sistemas basados en microkernels: incluyen en el núcleo sólo las funciones esenciales del S.O, estando el resto de funciones implementadas en módulos a nivel de usuario sobre el núcleo

Memoria virtual (angulo)

conjunto de programas que tienen el sistema operativo que hacen creer a las CPU que pueden manejar directamente los discos aunque en la realidad sólo pueda acceder a la memoria electrónica.

Para utilizar la memoria virtual el procesador deberá realizar una serie de comprobaciones y pasos:

1. Genera la dirección del objeto que necesita, (MMU) comprueba si el objeto se encuentra en memoria principal
2. Si está en memoria principal accede a él normalmente.
3. En caso contrario, comunica el hecho al sistema operativo que pone en marcha la rutina encargada de localizar el objeto en memoria virtual (disco) y transferirlo a la memoria principal para que la CPU acceda normalmente a él.

La memoria virtual permite que las aplicaciones ocupen mucho más espacio que el disponible en la memoria principal

Formas de organizar la memoria virtual:

1. Segmentacion
2. Paginacion

Tipos de memorias virtuales

Los distintos modelos de memoria virtual se diferencian por sus políticas de solape y por métodos que emplean en la organización de la memoria

El sistema operativo debe tener en cuenta la fracción de memoria principal que se va a sustituir o cargar en disco así como las modificaciones que existan de lectura y escritura. Los criterios usados son los siguientes

- Regla FIFO: Se sustituye la fracción que más tiempo lleva en la memoria principal para dejar hueco a otra.
- Regla LRU: La porción que lleva en la memoria más tiempo sin haber sido usada.
- Regla LIFO: Se sustituye la fracción que menos tiempo lleva en la memoria principal para dejar hueco a otra.
- Regla LFU: La porción que se accedido menos veces desde que se inició el proceso.
- Regla RAND: se elige una porción al azar
- Regla CLOCK: Cuando se coloca un bit de uso en cada entrada de una cola FIFO y se establece un puntero que se convierte en circular. Es una aproximación al algoritmo LRU con una simple cola FIFO.

Traductores de lenguajes

Son programas que convierten los programas escritos por el usuario en lenguajes simbólicos o lenguajes de maquinas

Ensambladores:

- Software que traduce un archivo fuente escrito en un lenguaje assembler a un archivo cuyas instrucciones esten en código maquina, que es ejecutable directamente por el procesador para el que se creo
- Los traductores se dividen en dos grupos en función de la relación entre lenguaje fuente y lenguaje maquina. El primer grupo traduce una instrucción de un lenguaje y genera una única instrucción de maquina, mientras que el segundo grupo lo componen lenguajes de alto nivel en los que una sentencia se traduce a varias instrucciones en código de maquina

Interpretes:

- Traductor de lenguaje que traduce una instrucción en lenguaje de alto nivel a lenguaje de maquina y, de ser correcto, la ejecuta de inmediato. Si hay un error semántico, lo señala e interrumpe la ejecución
- El programa se va probando a medida que se confecciona
- Debe traducirse cada vez que se quiera ejecutar, aun cuando no haya habido modificaciones

Compiladores:

- Traductor de lenguaje que traduce un programa escrito en lenguaje de alto nivel a lenguaje de maquina
- Separa la traducción de la ejecución del programa, agilizando tanto una tarea como la otra
- La ejecución del programa solo se realiza cuando la compilación termino satisfactoriamente
- Permite obtener el código de maquina del programa compilado
- Permite hacer un resguardo del programa compilado en una memoria externa

Unidad 13

Transferencia de información

Buses

Conjunto de conductores que transfieren señales eléctricas en forma pasiva, asociado con un hardware que regula su actividad, denominado controlador de bus.

Cada dato transferido por un bus se conoce como **transferencia elemental** y se produce en un tiempo determinado, regulado por el controlador, y **denominado ciclo de bus.**

Las señales que representan bits sobre el bus son digitales y pueden transferir el dato en **serie** o en **paralelo**

Caudal del bus: grado de paralelismo del bus unido a la velocidad que admite para lograr la transferencia

Jerarquia de buses

- Buses internos al chip
- Buses que conectan chips sobre una placa
- Buses que conectan distintas placas
- Buses de entrada/salida: determinan una multiplicidad de estructuras que se denominan **arquitectura de buses**.
  Arquitectura de buses: permite definir normas de comportamiento para la transferencia de datos desde o hacia los dispositivos de entrada/salida

Dispositivos de entrada/salida

Se denominan dispositivos de E/S tanto las unidades periféricas en sí como aquellas "intermediarias", que se encargan de efectivizar una transferencia entre la memoria interna y la memoria externa en los periféricos

En toda transferencia se utilizan señales de control, dato y dirección, esta vez sobre los buses de E/S.

Cuando la actividad del bus de E/S es sincrónica se utilizan señales del clock que regulan la transferencia

Cuando la actividad del bus es asincrónica, la velocidad depende de los tiempos de los dispositivos conectados a ellos

señales diferentes comunes a la mayoría de los buses de E/S para "controlar" la transferencia:

- Señal de clock
- Señal que puede habilitar un tiempo de espera
- Señal de lectura o escritura a un disp. De E/S
- Señales de interrupción a la CPU
- Señales de reconocimiento que permiten el dialogo de los dispositivos durante la transferencia
- Señal de bus cedido u ocupado

Las señales de dirección permiten representar la dirección del emisor y el receptor. Cada dispositivo de E/S tiene asociada una combinación binaria

Las señales de dato representan los bits del mensaje que se ha de transferir

Las unidades de comunicación con el bus son unidades hardware, que actúan de intermediarias en la comunicación CPU - memoria interna →controlador → periférico. Se agrupan en:

- Interfaces paralelo
- Interfaces serie
- Controlador DMA
- Canal de E/S

Controladores

Controlador de periféricos: dispositivo asociado en forma directa al periférico y esta construido por:

- Un buffer interno: que permite el almacenamiento de la información que "viaja" desde o hacia el soporte.
- Una lógica de control: que interpreta comandos de periférico, genera señales para su ejecución y gobierna así la unidad.

Sus funciones son:

- Aislar el software de servicio de E/S, que se "ocupa" de la transferencia de los detalles específicos del hardware del periférico y los convierte en invisibles.
- Compatibilizar la velocidad del periferico respecto de la del resto del sistema

Adaptadores

provee una función para conectar y lograr la operación de un componente conectado a un bus. Un adaptador puede residir en una tarjeta o en la placa del sistema

todo dispositivo que no cumpla la función por sí solo necesita su propio adaptador

Puertos de E/S

es un área de almacenamiento alojada en una interface, que permite la comunicación de un periférico con la memoria para enviar o recibir una secuencia de bits. (ejem: USB)

Interfaces

hardware que actúa de nexo entre un periférico o un adaptador y el bus. Sirven para adecuar las señales y preparar la transferencia elemental basada en un protocolo

- Interfaz paralela: dispositivo hardware que permite el control de la transferencia en paralelo entre el bus de sistema y un periférico. La interfaz cuenta con registros denominados ports. Un port está dividido en partes (registro de datos y registro de control), su función es lograr la transferencia elemental.
- Interfaz serie: dispositivo hardware que permite el control de la transferencia de bits en serie entre el bus y un dispositivo de E/S. Está asociado con una lógica de direccionamiento que permite establecer que esa interfaz fue seleccionada por el procesador para la transferencia. Los registros de la interfaz constituyen el denominado puerto serie

<u>Canales o procesadores de E/S:</u>

Procesadores "delicados" o "específicos" para controlas las transferencias de E/S sin intervención de la CPU en la ejecución del software de E/S.

<u>Transferencia de E/S</u>

Aspectos fundamentales para resolver:

- Sincronizar los tiempos de transferencia entre la CPU-memoria y el periferico.
- Decodificar los bits que identifican al dispositivo
- Convertir, si es necesario, un mensaje serie a paralelo, o al reves
- Convertir, si es necesario, el mensaje enviado de un formato a otro.
- Convertir, si en necesario, el mensaje enviado de un código a otro

- Controlar, si es posible, que el mensaje enviado se reciba en forma correcta.
- Decodificar un comando para el dispositivo.
- Controlar las banderas de estado

Maestros:

- Dispositivos que tienen el control del bus en un momento determinado
- puede enviar las señales de control, dirección y dato sobre el bus
- conoce la dirección del emisor del mensaje y la dirección del receptor

Esclavos:

- Los dispositivos restantes, conectados al bus pero que no lo controlan
- puede pedir un servicio de transferencia (request) pero no inicializarla

La estrategia de control del bus depende de la arquitectura diseñada para el sistema y se denomina “arbitraje” del bus.

Drivers

Aquellos programas que “conocen” el dispositivo periférico. Son programas en cuya codificación se hace referencia a los comandos propios para cada periférico.

Cada driver actúa como un receptor de requerimientos de otros programas, que pertenecen a otro nivel y desconocen las peculiaridades de cada uno de los distintos dispositivos externos.

Los comandos de los periféricos se clasifican en:

- Comando de verificación. Evaluan si el periferico esta prendido o apagado, ocupado, no operable, en error de operación
- Comandos de control. Ordenan al periferico a prenderse o apagarse, saltar de pagina en una impresora, leer, escribir

Modalidades de E/S

El sistema operativo cuenta con programas que gestionan las transferencias de entrada/salida pero esta vez en un nivel superior. Realizan funciones comunes a todos los dispositivos periféricos.
Desde el punto de vista de un programa de aplicación, un proceso que ejecuta la CPU actualmente puede ir escribiendo datos en un “área de almacenamiento” hasta completar el bloque, momento en el que se llama al servicio para realizar la trasferencia del bloque completo

Los programas de aplicación llaman a los programas de E/S del sistema operativo mediante una “llamada al sistema”. La relación entre el programa de aplicación en estado de ejecución como peticionario del servicio de transferencia y el hardware de entrada/salida como su proveedor se muestra asi:
Aplicación → Programa de E/S del nivel abstracto → Driver → Hardware

Para llevar a cabo la transferencia completa se pueden identificar modos:

- Transferencia controlada por programa
- Transferencia iniciada por interrupción
- Transferencia con acceso directo a memoria

- Transferencia a traves de un procesador IOP

Transferencia controlada por programa:

- la que ejecuta el programa de E/S es la CPU, que para comunicarse utiliza un bloque hardware denominado interfaz. Es la CPU la que controla el acceso a memoria para ubicar el dato
- La CPU debe verificar el estado de la interfaz a través del puerto de control en forma continua, a través de sus propios registros internos y los de la interfaz
- Esta modalidad de entrada/salida también se conoce como "por sondeo" que significa que es la CPU quien se ocupa de verificar el estado del periférico y no el periférico el que interrumpe a la CPU

Transferencia iniciada por interrupción:

- La CPU no verifica de manera continua el estado de un dato. La interfaz asociada genera un aviso que indica que esta preparada para transferir (provoca una interrupción), se indica con un bit de interrupción

Transferencia con acceso directo a memoria (DMA)

- Las interfaces pueden conectarse a la memoria a través del controlador de acceso directo a memoria. Los DMA están asociados a dispositivos rápidos y que transfieren la información en grupos de bytes.
- la CPU sólo interviene indicándole a la DMA la cantidad de palabras a transferir, la posición inicial de la palabra en memoria interna y la dirección del dispositivo como parámetros
- La técnica consiste en relacionar la funcionalidad DMA con el bus que conecta la CPU con la memoria principal o interna, y que ese controlador establezca la relación memoria-interfaz sin intervención de la CPU

Unidad 14

Procesadores a nivel avanzados

Paralelismo a nivel de instrucción

consiste en la simultaneidad de ejecución por etapas. La cantidad de etapas definidas se denomina grado de paralelismo.

Dependencia de datos: Es necesario concluir la instrucción n para ejecutar la instrucción n+1

Ejecución fuera de orden: Técnica que produce un reordenamiento de las instrucciones sin afectar la lógica del programa

Diseño superescalar: si por cada etapa se duplica la cantidad de unidades de ejecución. Se crean múltiples vías de procesamiento paralelo a nivel instrucción

Durante la ejecución, las múltiples instrucciones se leen y pasan a un panificador, que determina cuales de ellas se pueden ejecutar en paralelo, la mayor complejidad de diseño de CPU se relación con este. Mientras más paralelismo haya mayor es la posibilidad de dependencias o saltos inesperados. La predicción de bifurcación procura predecir qué camino o bifurcación tomara una instrucción de salto condicional, Intel la llama rama de ejecución.

El paralelismo a nivel hilo de ejecución (TLP) tiene como objetivo incrementar el número de programas individuales que una CPU pueda ejecutar en forma simultánea, los hilos son rutinas concurrentes que comparten variables globales y el mismo espacio de direccionamiento, su mejora en el rendimiento se apoya en la habilidad de solapar cálculos con operación de entrada/salida.

Para lograr el TLP se usa el multiprocesamiento a nivel instrucción a nivel de chip y el multihilado simultaneo.

Paralelismo a nivel de arquitectura

Para lograr un alto rendimiento, los microprocesadores no sólo deben ejecutar las instrucciones de una manera más rápida, sino que también es preciso que ejecuten más instrucciones por ciclo de reloj. La ejecución en paralelo requiere técnicas de predicción de saltos o especulación de datos; en este último caso, para aliviar el retrieve de memoria. Cuando todo esto no alcanza para lograr el rendimiento deseado, se debe llevar el "paralelismo a nivel procesador" a un grado mayor que se alcanza en las arquitecturas denominadas paralelas

Una computadora con más de un procesador ejecutando en paralelo y de forma coordinada puede pertenecer a una de las siguientes categorías determinadas por la taxonomía de Michael J. Flynn:

1. SISD: la CPU recibe una sola secuencia de instrucciones que operan en una única secuencia a los datos y no hay paralelismo;
2. MISD: se utilizan para ejecutar distintos procesos sobre un único flujo de datos. Cada proceso se desarrolla en un procesador con su propia unidad de control y accediendo a una memoria común a todos
3. SIMD: Cada procesador opera en sincronismo con los demás, ejecuta la misma secuencia de instrucciones sobre diferentes flujos de dato
4. MIMD: Este tipo de computadora es de procesamiento paralelo asincrónico. Para cada procesador se asigna una secuencia de instrucciones y datos.

Tabla 14-7. Resumen de sistemas paralelos.

COMPUTADORAS PARALELAS	MIMD	MULTIPROCESADOR Información compartida en memoria compartida	DSM-NUMA	DSM-SVM	
				DSM-COMA	
				DSM-nccNUMA	
				DSM-ccNUMA	Servidores Origin Silicon Graphics
			UMA	SMP Escalable	
				SMP bus	HP-UX SMP
		MULTICOMPUTADORA Información compartida por paso de mensajes			
	SIMD	Computadoras vectoriales			NEC SX-9
		Máquina de conexión			
	MISD				
REDES DE COMPUTADORAS*	NoW				
	Malla				
	Cluster	Cluster simple			
		Constelación			

Microprocesadores avanzados

Descripcion de la arquitectura Itanium

fue diseñada con el fin primario de obtener un mayor paralelismo a nivel instrucción, pero con técnicas de predicción, especulación y predicación, su arquitectura se constituye en lo que se denomina una arquitectura de la bifurcación. Estas características se apoyan de aplicaciones de alto nivel como datawarehousing y datamining.

Esta provista de mecanismos que le permiten al código compilado gestionar la forma de uso del procesador a nivel hardware corroborando información del entorno en tiempo de ejecución que permite salvar fallas.

Modos de operación

Cuando se ejecutan instrucciones IA-32 se conmuta a Itanium con la instrucción jmpe, cuando se ejecutan instrucciones Itanium se conmuta a IA-32 con la instrucción br.ia.m. Si se produce una interrupción, siempre se operan en entorno Itanium y se retorna del servicio con rfi.

Intel Italium arquitectura EPIC

Epic esta tecnología permite la ejecución en paralelo de la mayoría de las instrucciones Isa-64bits que definen la arquitectura, además de contar con gran cantidad de registros internos.

El paralelismo explicito significa que las dependencias de instrucciones se evalúan a nivel compilación, mientras que a nivel ejecución las instrucciones de distintas ramas de un salto condicional son marcadas por registros para ejecutarse en forma simultánea.

Paralelismo explicito

la tarea de lograr el paralelismo está a cargo del compilador que organiza de manera eficiente el código para su procesamiento y hace explícito el pedido para que de este modo el procesador pueda enfocarse en la ejecución simultánea de instrucciones de la forma más efectiva

- Tablas de registros de propósito general

Son 128 de 64bits cada uno, y se denominan GR0-GR127, cuando se ejecutan aplicaciones sobre datos enteros se utilizan los registros GR0-GR31, están incluidos los registros de segmento. Los primeros 32 son estáticos, los restantes se llaman stacked general registers, las aplicaciones los pueden utilizar para almacenar el marco de pila de registros.

- Registros de coma flotante

Son 128 se identifican FR0 a FR127.

- Los registros predicado a PRs

Utilizados para el tratamiento de saltos, son 64 registros de 1 bit y se idéntican como PR0-PR063. Los predicados se utilizan en instrucciones de salto condicionado, hay 2 caminos de ejecución.

Una vez que se ejecuta la condición, el procesador guarda un 1 en un registro de predicado que corresponde a destino verdadero y 0 en los otros.

- Los registros rama o BRs

Son 8 registros de 64 bits cada uno y se identifican como BR7-BR0

- Especulacion

  Es una técnica que aplicada a instrucciones consiste en ejecutar una instrucción antes de tener la seguridad de que su resultado se vaya a utilizar.

  Control Speculation es el término aplicado y una de las formas de llevar a cabo esto es el uso de predicados.

  Especular es a nivel datos, la técnica consiste en leer anticipadamente un dato, suponiendo que una escritura pueda modificarla. Data Speculation es el término.

  Los compiladores capaces de utilizar la técnica de especulación deben transformar una lógica que ha sido programada en forma secuencial en una cierta cantidad de hilos de ejecución en paralelo.

- Predicacion

Ejecución de secuencia de instrucciones que dependen de una condición se cumpla. Todas las instrucciones se ejecutan aun cuando dependen del valor de su predicado. Si se pueden ejecutar ambas ramas en forma simultánea, el salto puede evitarse utilizando instrucciones denominadas código de predicado.

- Prediccion de saltos

Hay dos clases de predicciones:

- Estáticas: La predicción se produce a nivel compilación, el compilador estima una dirección para cada una de las instrucciones de salto, en los saltos incondicionales siempre se conoce la dirección, no es estimativa.

Dinámica: Ocurre durante la ejecución, se construye una tabla de saltos y se guarda en un registro su comportamiento con el fin de decidir cuál es la dirección más probable de la próxima instrucción.