

<b>20.1. – Características generales .....</b>	<b>2</b>
20.1.1.- Introducción.....	2
20.1.2.- Arquitectura Básica.....	3
<b>20.2. – Aportaciones y nuevos recursos arquitectónico .....</b>	<b>4</b>
20.2.1.- Registros de Propósito General .....	4
20.2.1.1.- Eliminación de Dependencia de Falsos Registros .....	5
20.2.2.- Caché L2 .....	5
20.2.3.- Aplicación de la técnica RISC.....	7
20.2.4.- Supersegmentación.....	8
20.2.5.- Arquitectura superescalar .....	9
<b>20.3.- Nuevas Instrucciones.....</b>	<b>11</b>
<b>20.4.- Análisis del rendimiento.....</b>	<b>11</b>
20.4.1.- Ejecución dinámica .....	11
20.4.2.- Orientación del software .....	13

## 20.1- CARACTERÍSTICAS GENERALES.

### 20.1.1- Introducción.

La aparición, a finales de marzo de 1995, del procesador Pentium Pro supuso para los servidores de red y las estaciones de trabajo un importante avance, tal y como ocurriera con el Pentium en el ámbito doméstico. Este nuevo dispositivo constituía la sexta generación de procesadores de la familia x86.

La potencia de este procesador había aumentado de forma notable, gracias a la arquitectura de 64 bits y el empleo de una tecnología revolucionaria como es la de 0,6 micras, lo que permitía la inclusión de cinco millones y medio de transistores en su interior. Para la fabricación de dicho procesador se había usado la tecnología BICMOS de 4 niveles.

Una de las novedades que aportaba el pentium pro era su aspecto físico, que se presentaba en un encapsulado con 387 patitas, bastante grande debido a que incorpora dos chips: uno de ellos contiene el procesador y el otro una caché de segundo nivel, que se encargaba de mejorar la velocidad de la memoria caché, lo que resultaba en un incremento del rendimiento sustancioso. Dicha caché se comunicaba con la CPU a la misma velocidad del procesador. Las frecuencias de reloj se mantenían como límite superior en 200 MHz, partiendo de un mínimo de 150 MHz.

En la arquitectura de este procesador se encuentra un bus de direcciones de 36 líneas, por lo que es capaz de direccionar hasta 64 Gigabytes.

Como desventajas tecnológicas, es reseñable el alto precio de fabricación del conjunto multichip. La microarquitectura utilizada está optimizada para software de 32 bits, que tienen rendimiento pobre con código de 16 bits. Se aprecia un consumo de energía y disipación de calor totalmente inapropiadas para ordenadores portátiles

Todas estas características se encuentran resumidas en la tabla 20.1. que corresponden al primer modelo que fue comercializado.

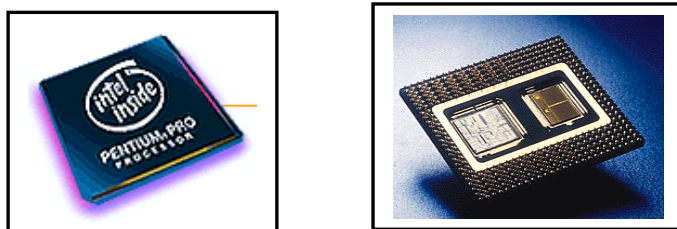


Figura 20.1 . Fotografía de la cápsula del Pentium-Pro

	CARACTERÍSTICAS GENERALES
AÑO	1995
MILLONES DE TRANSISTORES	5,5 (CPU) 15,5 (L2)
TECNOLOGÍA (MICRÓMETROS)	0,6
FRECUENCIA (MHZ)	150
PRECIO (\$)	974
RENDIMIENTO	220 SPEC <sub>INT92</sub>
CAPSULA	PGA 273
ALIMENTACIÓN (VOLTIOS)	2,9
POTENCIA (WATIOS)	20
CACHE L1 (KBYTES)	8 (INSTRUCCIONES) 8 (DATOS)
CACHE L2	256 Kbytes (INTERNA)

Tabla 20.1. Resumen Características

En el Pentium Pro los siguientes puntos de su arquitectura, son de vital relevancia:

1. Integración total de una caché de segundo nivel con el procesador.
2. Potenciación del paralelismo mediante la ejecución dinámica de las instrucciones, es decir, fuera de orden.
3. Potenciación del paralelismo mediante la incorporación de más unidades de ejecución que trabajen simultáneamente, lo que recibe el nombre de superescalar.
4. Adaptación a la línea de los procesadores RISC, intentando traducir las instrucciones complejas de la familia x86, con un formato muy irregular y operandos inmediatos de longitud variable, a microprocesadores RISC con simetría absoluta y posibilidad de ejecución fuera de orden.
5. Incremento de la velocidad mediante la disminución del ciclo de reloj, aumentando así el número de etapas del cauce, que se eleva a 14 y da lugar a la supersegmentación.

### 20.1.1- Arquitectura Interna Básica.

El microprocesador del Pentium Pro está constituido principalmente por los siguientes elementos principales:

1. **Unidad de Bus Externo:** Realiza transacciones de bus cuando es requerido para ello por la caché L2 o núcleo de microprocesador.
2. **Unidad de Bus Trasero:** Interfaz entre el núcleo y la caché L2.
3. **Caché L2 Unificada:** Actúa cuando la caché L1 falla, tanto en datos como en código. Cuando sea necesario hará peticiones a la Unidad de Bus Externo.
4. **Caché de Datos L1:** Da servicio a las peticiones de carga y almacenamiento hechas por las unidades de carga y almacenamiento. Cuando no pueda dar servicio, pasará la petición a la caché L2.
5. **Caché de Código L1:** Da servicio a las peticiones de búsqueda de instrucciones formuladas por el prebuscador de instrucciones.
6. **Microprocesador:** Responsable de las siguientes operaciones primordiales:
  - Búsqueda de instrucciones.
  - Predicción de saltos.
  - Análisis del flujo de instrucciones.
  - Traducción de las instrucciones CISC a RISC.
  - Despacho, ejecución y retirada de microoperaciones.

7. **Unidad Local Apic:** Responsable de recibir las peticiones de interrupción de otros procesadores, de los pines locales de interrupción, del temporizador APIC. Estas peticiones prioritarias, son enviadas al microprocesador para su ejecución.

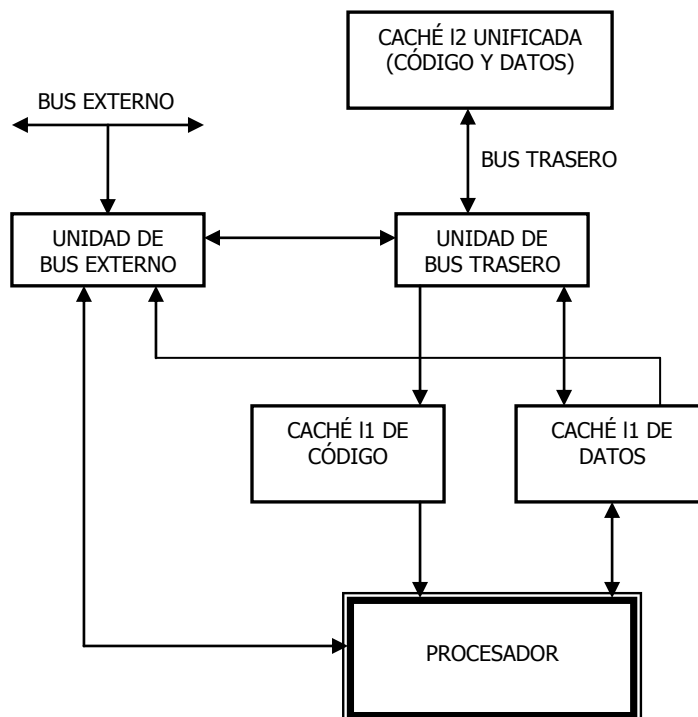


Figura 20.2. Diagrama simplificado de bloques del Pentium Pro

## 20.2- APORTACIONES Y NUEVOS RECURSOS ARQUITECTÓNICOS.

### 20.2.1- Registros de Propósito General

Los registros de los anteriores microprocesadores x86 eran demasiado pequeños. Esta restricción solo permitía al microprocesador mantener un pequeño número de datos fáciles de acceder por las unidades de ejecución. A menudo, el programador se veía obligado a guardar en la memoria los contenidos de uno o más registros, cuando necesitaba leer nuevos operandos para operar con ellos. Más adelante puede necesitar el registro original, que guardó previamente en la memoria, por lo que tendrá que volver a leerlo de la memoria, lo cual es una pérdida de tiempo e introduce una penalización en el rendimiento del programa.

El Pentium Pro soluciona este problema con 40 registros nuevos. Las instrucciones son traducidas a microoperaciones antes de ser ejecutadas. Cuando se ejecutan se pueden dar las siguientes circunstancias.

1. La microoperación tiene que colocar un valor en uno de los registros de propósito general.
2. La microoperación tiene que leer un valor que fue colocado en el registro por otra instrucción ejecutada con anterioridad.
3. La microoperación cuando es ejecutada, cambia los contenidos del registro EFLAGS.

Menos mal que el microprocesador permite la ejecución fuera de orden, ya que si el resultado de la ejecución fuera inmediatamente reflejado en el set de registros del microprocesador, los valores en los registros se cambiarían, los bits de condición de los registros de estado serían actualizados, probablemente de forma incorrecta y en un orden inadecuado.

En vez de una carga inmediata de los resultados de la instrucción en los registros reales, el microprocesador almacena el resultado. Si una vez ejecutada, otra microoperación requiere el resultado producido por las microoperaciones que le preceden, los resultados son enviados a la microoperación peticionaria. A esto se le conoce como Feed Forwarding.

Esta técnica de almacenamiento de registros es necesaria para poder realizar la ejecución fuera de orden, y es conocida también como Aliasing de Registros.

### 20.2.1.1- Eliminación de Dependencia de Falsos Registros.

Si examinamos el ejemplo que tenemos a continuación:

---

#### EJEMPLO 1

**Supongamos que se está ejecutando el siguiente programa:**

```
mov eax,17  
add mem,eax  
mov eax,3  
add eax,ebx
```

En el antiguo microprocesador x86, estas instrucciones hubieran tenido que ser ejecutadas de una en una para proporcionar el resultado correcto. El microprocesador no podría ejecutar las instrucciones 3ª y 4ª antes de haber ejecutado la 1ª la 2ª. La 2ª instrucción debe usar el valor de EAX antes de que la 3ª instrucción coloque un nuevo valor en EAX.

En el Pentium Pro el microprocesador es capaz de darse cuenta de que el registro EAX no tiene porque ser cargado con el valor 17. Se puede obtener el mismo resultado sumando 17 a la dirección de memoria. Del mismo modo, EAX no necesita ser cargado con el valor 3. El valor 3 se puede sumar a EBX y el resultado se carga en EAX. El microprocesador puede ejecutar de forma simultanea las instrucciones 1ª y 3ª, disponiendo de los valores 17 y 3 para cada una de las microoperaciones que lo necesiten. De igual manera las instrucciones 2 y 4 pueden ser ejecutadas en paralelo, utilizando los valores 17 y 3 asociados a las instrucciones 1 y 3.

---

### 20.2.2- Caché L2 y L1

La caché interna de primer nivel L1 del Pentium Pro dispone de una zona de 8kb destinada a contener instrucciones y otra del mismo tamaño para contener datos. La caché L1 de instrucciones es del tipo de asociativa de dos vías y la de datos es de 4 vías.

A un procesador que pase de los 100Mhz, no le basta con una caché L1 de capacidad reducida, ya que el porcentaje de fallos es lo suficientemente importante como para que afecte al rendimiento de la CPU al tener que acceder a la Memoria Principal DRAM (dinamic RAM). Se hace imprescindible potenciar la caché SRAM (static RAM), lo que motivó a INTEL a integrar junto con

el procesador una caché de segundo nivel de capacidad notable, con la que se redujesen drásticamente los accesos a la DRAM.

La caché de segundo nivel L2 del Pentium Pro tiene un tamaño de 256 kb y una estructura asociativa de 4 vías, siendo lo más importante de esta memoria la conexión mediante el bus trasero con el procesador caracterizándose por funcionar a la misma frecuencia que el procesador pudiendo transferir 64 bits por el bus de datos en un ciclo de reloj.

El procesador se comunica con la Memoria Principal con los módulos de E/S mediante otro bus, llamado frontal (figura 20.2.).

El procesador se empaqueta conjuntamente con la caché L2 por dos razones:

1. Construcción de sistemas de altas prestaciones a los fabricantes de equipos.
2. Mejorar el interfaz directo entre el procesador y la caché L2.

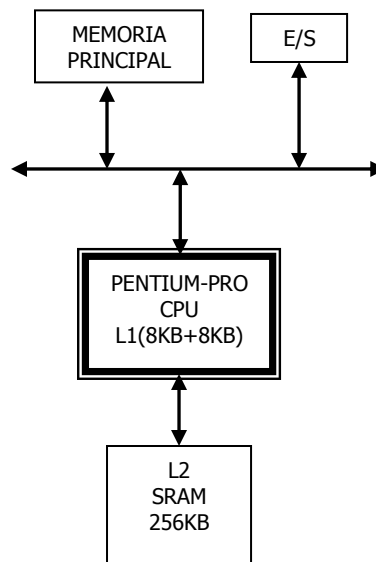


Figura 20.3. Arquitectura de la caché L2 en el Pentium-Pro

Cuando la CPU no encuentra las instrucciones ó datos en las caché L1 pasa a consultar la caché L2, lo que supone un retraso en el ciclo de reloj. Si tampoco se encuentra lo que se busca en la caché de 2º nivel se pasa a buscar en la Memoria Principal, lo que significa una gran penalización de tiempo y una degradación de rendimiento.

Para mejorar lo anteriormente comentado el Pentium Pro opera de forma independiente y simultanea con el bus frontal y trasero. Mientras se espera que se complete un acceso a la memoria, se puede iniciar otro. Para realizar dicho fin existe un Buffer de Ordenamiento de Accesos a Memoria (MOB), que guarda hasta 8 accesos o “transacciones”, pudiendo iniciarse un acceso aunque existan 8 pendientes de realizarse (figura 20.3.)

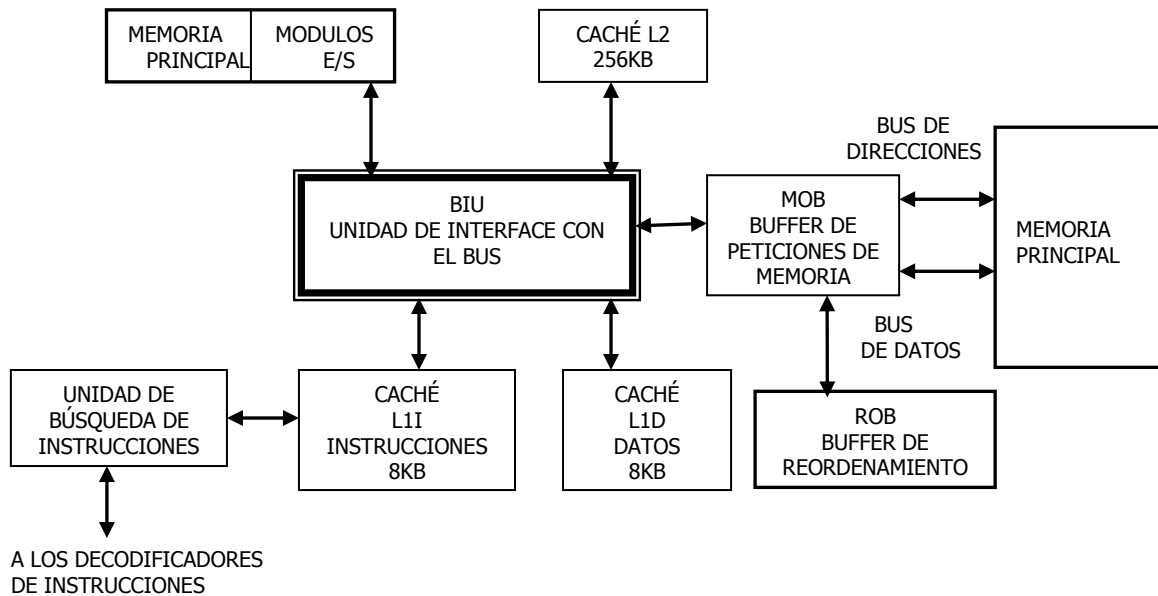


Figura 20.4 Arquitectura de la unidad de interface con el bus (BIU).

El MOB actúa como depósito de reserva y regulación, que permite a determinados accesos de lectura de memoria anticiparse a otras lecturas y escrituras, consiguiendo así elevar notablemente el rendimiento del sistema. El MOB actúa en coordinación con el ROB (Buffer de reordenamiento) que almacena las peticiones de lectura y escritura pendientes de ejecutarse hasta que desaparecen las causas que las impiden, bien procedan de una dependencia de datos o bien de la utilización de un recurso necesario.

Para evitar problemas entre las caches y la Memoria Principal, este procesador, al igual que sucedía con el 486, emplea el protocolo MESI, que está diseñado para evitar los errores generados cuando se usa código automodificable, típico en algunos programas de la familia x86.

### 20.2.3- Aplicación de la técnica RISC

Una de las características que INTEL arrastra desde sus comienzos, es que el software de los nuevos procesadores, es compatible con el software de todas las versiones anteriores. Por lo que el set de instrucciones de la familia x86 de tipo CISC ( Computadores de juego de instrucciones complejo).

Muchas de estas instrucciones son complejas, requiriendo una extensiva decodificación y lógica de ejecución. Algunas de ellas necesitan un numero de recursos elevados para completarse. Una instrucción puede suponer, una o más lecturas de memoria, una o más o escrituras de memoria. Además de estos inconvenientes, una instrucción puede ser desde 1 hasta 15 bytes de longitud. Lo cual es un gran obstáculo con respecto a la ejecución de más de una instrucción simultáneamente. El microprocesador tiene que saber donde acaba una instrucción y donde comienza la siguiente. Solamente entonces la lógica del procesador decodifica la instrucción y determina si pueden ser o no ejecutadas al mismo tiempo.

Debido a todo esto, los fabricantes de los procesadores se deciden por la técnica Risc (computadores de juego de instrucciones reducido), que manejan un grupo pequeño de instrucciones muy sencillas que se ejecutan por hardware en un solo ciclo de reloj.

La conversión de instrucciones Cisc a microprocesadores Risc se lleva a cabo en el Pentium Pro en la etapa **Unidad de decodificación** que consta de 4 bloques de los cuales dos son decodificadores básicos, encargados de la mayoría de las instrucciones.

Las instrucciones son prebuscadas en la memoria y almacenadas en la caché L2 y en la caché de código L1. Como de la caché se obtienen bloques de instrucciones, el núcleo del microprocesador los analiza para distinguir los límites entre las instrucciones. Entonces decodifica las instrucciones, variables en longitud, en instrucciones RISC, de longitud fija, llamadas microoperaciones. Estas se almacenan y esperan a ser despachadas y ejecutadas.

Se puede obtener hasta un máximo de seis microoperaciones por ciclo de reloj, dado que los decodificadores trabajan de forma independiente y en paralelo.

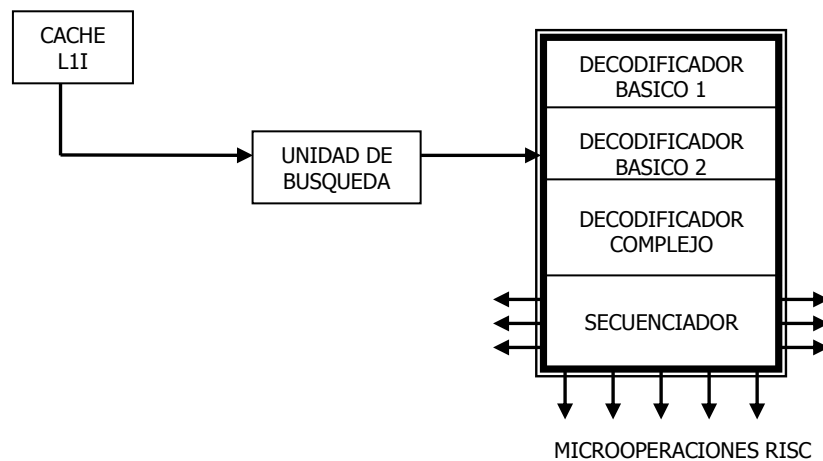


Figura 20.5 Los decodificadores que convierten las instrucciones de la familia x86 en microoperaciones RISC.

## 20.2.4- Supersegmentación

Se dice que el cauce de procesamiento de instrucciones del Pentium Pro es supersegmentado dado que consta de 14 etapas y cuantas más etapas tenga el cauce más elementales son las funciones que realiza. Sin embargo la existencia de tantas etapas complica los riesgos debidos a las dependencias de datos y a las bifurcaciones condicionales.

La primera etapa del cauce: calcula el valor que hay que cargar en el contador de programa (EIP) para apuntar a la dirección de la siguiente instrucción. Como tenemos que tener en cuenta las instrucciones de bifurcación, se dispone de una memoria CAM de 512 entradas, llamada BTB (Buffer de Destinos de Bifurcación), que almacena las direcciones que se han utilizado en las bifurcaciones anteriormente.

Las penalizaciones son bastante significativas, oscilando entre 4 y 5 ciclos de reloj, para mejorar los aciertos se emplean los métodos clásicos de predicción: estático y dinámico.



El método estático para predecir si existe ó no bifurcación se ayuda de un compilador que analiza el comportamiento del programa, sin embargo el comportamiento dinámico evalúa las ramificaciones. Para ello la BTB dispone de bits de historia de la ramificación a dos niveles: a nivel de ramificación individual y a nivel de grupo de ramificaciones.

Las tres siguientes etapas del cauce se destinan a la búsqueda de las instrucciones. Estas proceden de la caché L1 de Instrucciones, de donde son leídas a razón de 2 líneas de 32 bytes cada una por ciclo de reloj. La delimitación de las instrucciones x86 es necesaria al tener un formato variable comprendido entre 8 y 120 bits. Para delimitar las instrucciones se toman los 16 bytes al que apunta el EIP, los cuales se introducen a uno de los decodificadores. Es posible que el byte apuntado por el EIP se encuentre al final de una línea; por dicho motivo se leen dos líneas, evitando tener que realizar otro acceso para completar los 16 bytes que delimitan cada instrucción.

Una vez se hayan delimitadas las instrucciones x86 se introducen al decodificador de instrucciones que transforma los códigos binarios en secuencias de microprocesadores Risc.

La séptima etapa se encarga de realizar un renombrado de los registros de propósito general que emplean las instrucciones para los numero enteros y de coma flotante. 8 de estos registros se utilizan para número enteros y otros 8 para números en coma flotante. La existencia de pocos registros da lugar a muchos problemas de dependencias de datos, ya que todas las instrucciones se disputan los pocos registros disponibles. Como hay que mantener la compatibilidad y admitir dichos registros x86, el Pentium Pro dispone en esta etapa de una Tabla de Alias de Registros (RAT), que permite representar a cualquiera de los registros que emplean las instrucciones por medio de un registro físico del banco de registros ampliado. Esta ampliación alcanza a 40 registros implementados en el Buffer de Reordenamiento.

La octava etapa, se encuentra el ROB (Buffer de Reordenamiento), uno de los elementos clave en la estructura de este microprocesador. Se trata de una memoria CAM con 40 entradas de 254 bits que puede contener cada una una microoperación, los operandos que maneja y los bits de estado que señala. En esta etapa interviene la Unidad de Predicción de destinos de las bifurcaciones (BTB). Un fallo de la BTB significa una penalización promedio de 8 ciclos, necesarios para la recuperación y recomposición de todos los registros que han afectado a la predicción errónea.

Tras el ROB se encuentra (RS) que es la estación de reserva. Tiene como objetivo planificar el orden con que se van entregando las microoperaciones a las Unidades de Ejecución, que comprenden las etapas 10ª y 11ª. Admite hasta 20 microoperaciones.

### **20.2.5- Arquitectura superescalar**

Este concepto corresponde a la existencia de 5 unidades de ejecución las cuales trabajan de forma simultanea é independiente, lo que significa que se puede alcanzar un flujo máximo de hasta 5 microoperaciones ejecutadas por cada ciclo de reloj. Habitualmente se reduce a 3 por ciclo de reloj.

La ejecución dinámica permite la ejecución fuera de orden y con varias microoperaciones en paralelo. Es posible gracias a la intervención de la ROB y de la RS. Dos de las unidades de ejecución (IU1 e IU2) trabajan sobre microoperaciones que manejan número enteros, otras dos (AGU1 y AGU2) calculan las direcciones de los operandos y las instrucciones de carga/almacenamiento y, además, hay una quinta (FPU) que trata las instrucciones con números en coma flotante. (figura 20.5).

En el caso que los bits de estado de una determinada microoperación no detecten dependencias, si se dispone de los operandos y existe una Unidad de Ejecución libre, se lanza dicha microoperación a la fase de ejecución.

Por la compleja red de interconexión y realimentación, los resultados producidos sobre las microoperaciones por las unidades de ejecución se devuelven a la estación de reserva y al ROB. De esta forma, la Estación de Reserva puede introducir los resultados de algunas microoperaciones en otra Unidad de Ejecución que los puede necesitar como operandos, evitando retrasos. Con los resultados recibidos en la ROB se determina cuando se ha completado una microoperación y está lista para ser retirada. La máxima velocidad de retirada de microoperaciones es de 3 ciclos de reloj.

Ejecutada la microoperación, se actualizan los bits de estado y se describe en el ROB para dar información a la estación de reserva y a la unidad de Extracción o Estación de retiro, que almacena los resultados en el Banco de Registros de Retiro (RRF) cuando dicha microoperación se halle completamente terminada. La Unidad de Extracción puede tratar en cada ciclo de reloj hasta la terminación de 3 microoperaciones, por lo que se dice que el Pentium Pro es un “procesador superescalar de 3 vías”.

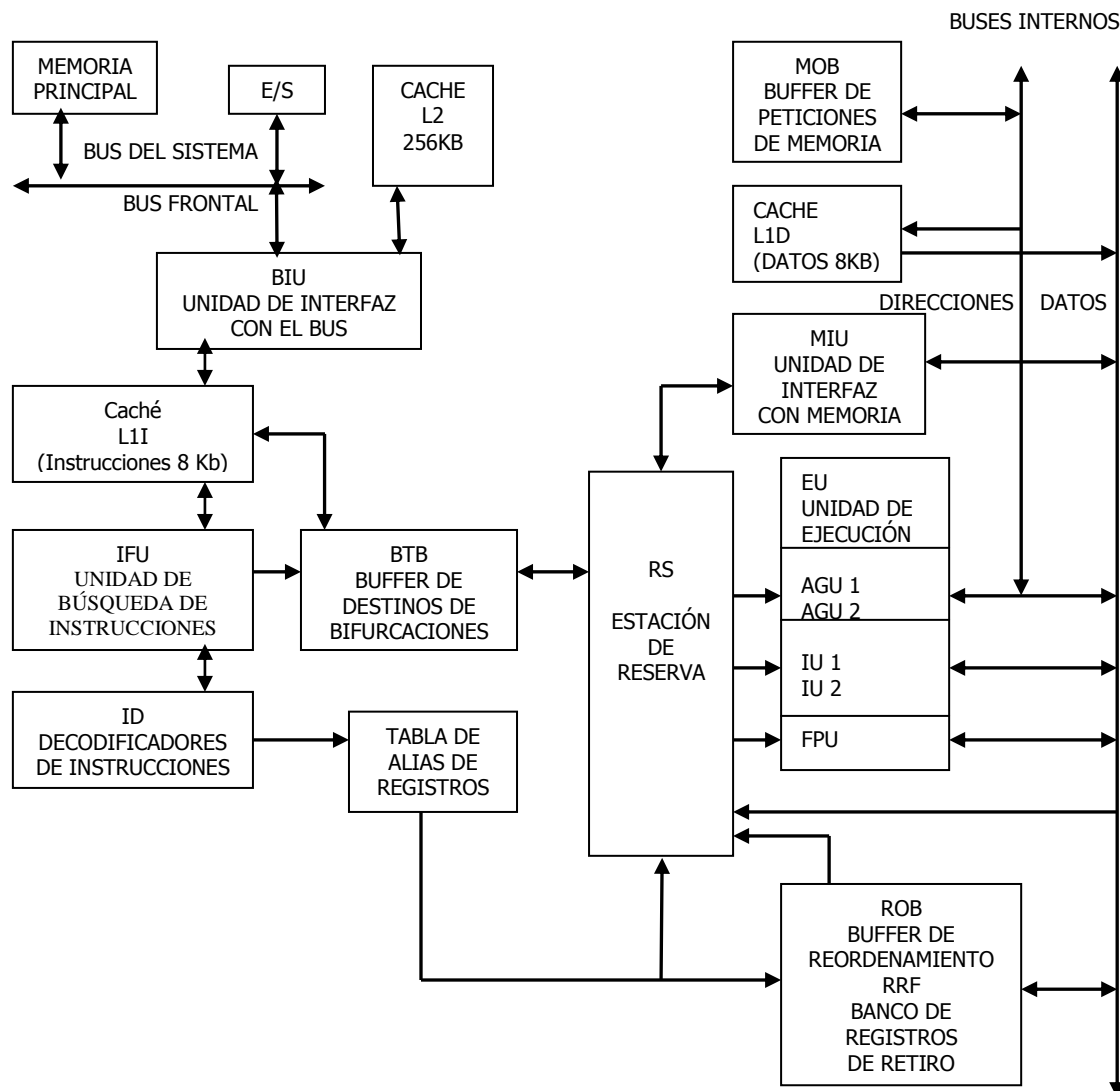


Figura 20.6 Arquitectura interna del Pentium Pro

### 20.3-.NUEVAS INSTRUCCIONES.

El repertorio de instrucciones de Pentium Pro solo se diferencia del Pentium, en una instrucción nueva, la de mover datos condicionalmente que supone una mayor predicción de ramificaciones en la ejecución.

La nueva instrucción es una instrucción de movimiento condicional que puede ser usada por los escritores de compilación como alternativa a una construcción "test and set" (testear y poner a 1). Esto permite que los bifurcaciones de datos dependientes sean eliminados. El código de resultado será más predecible por el procesador y tendrá por tanto mayor rendimiento.

### 20.4- ANÁLISIS DEL RENDIMIENTO.

Con la llegada del Pentium Pro se produjo un gran salto en el rendimiento de los computadores.

Para observar la evolución en el rendimiento del Pentium Pro se ofrece el siguiente gráfico con la comparativa del rendimiento en la medida SPECint92.

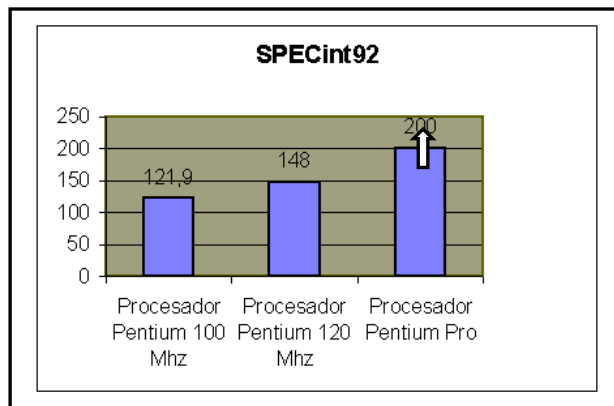


Figura 20.7 Comparativa de rendimiento con SPECint92.

La mejora en el rendimiento se dió debido a la incorporación nuevas mejoras, de las cuales destacaron la ejecución dinámica y la inclusión de una memoria cache secundaria integrada en el encapsulado del chip.

#### 20.4.1- Ejecución dinámica

Para incrementar el rendimiento del Pentium Pro, se intentó aprovechar los tiempos improductivos que se generan en los accesos a la Memoria Principal, es decir cuando se producen fallos en las caches; y usando esos fallos para ejecutar otras instrucciones posteriores que no dependan del resultado de la que se está ejecutando. El Pentium Pro ejecuta las instrucciones fuera del orden, que establece el programa y decidiendo si es posible realizarlas ó no.

## EJEMPLO 2

**Supongamos que se está ejecutando el siguiente programa:**

```
mov r1,(r0)
add r2,r1
inc r6
sub r7,r4
```

La 1ª instrucción carga en r1 el contenido de la posición de memoria apuntada por r0. Dicha posición de memoria, por ser totalmente aleatoria, causará fallo en las caches L1 y L2, por lo que se deberá acceder a la Memoria Principal durante vario ciclos de reloj. En este procesador, para evitar que quede inactivo durante este proceso, se exploran las siguientes instrucciones del programa. La 2ª instrucción no se puede ejecutar porque depende del resultado de la 1ª, que se está llevando a cabo. Pero la 3ª y la 4ª instrucción son independientes de las dos anteriores y pasa a ejecutarlas la CPU originando resultados temporales, que no se transforman en permanentes, hasta que no se realicen las anteriores y se restablezca el orden del programa. Los resultados de la 3ª y la 4ª instrucción se guardan en el Buffer o Depósito de Reordenamiento (ROB).

El Pentium Pro ejecuta las instrucciones fuera de orden. Lo consigue explorando de 20 a 30 instrucciones por delante de la que se halla en curso, y la apunta el contador de programa (CS:EIP). Esta labor es efectuada por La Unidad de Búsqueda, que recoge las instrucciones de L1 y las envía a la unidad de decodificación, la cual convierte las instrucciones x86 en microoperaciones tipo a Risc, que se envían al ROB. En la (figura 20.6) se puede apreciar como la Unidad de Búsqueda y Decodificación introduce al ROB las microoperaciones generadas.

El Buffer de Reordenamiento o ROB comunica con las tres unidades operativas del procesador. La Unidad de selección y Ejecución explora en las microoperaciones del ROB las dependencias de datos y las necesidades de recursos que precisan y seleccionan aquellas que son posibles de ejecutar, las ejecuta y los resultados temporales obtenidos, vuelve a depositarlos en el ROB. Una vez los resultados temporales pasan a ser definitivos al restablecerse el orden original del programa, la Unidad de Extracción, también llamada Unidad de Retiro, los recoge y los almacena.

El comportamiento de la ejecución dinámica de las instrucciones en el Pentium Pro se implementa mediante el uso conjunto de estos tres recursos:

1. Predicción de ramificaciones múltiples.
2. Análisis del flujo de dato.
3. Ejecución especulativa.

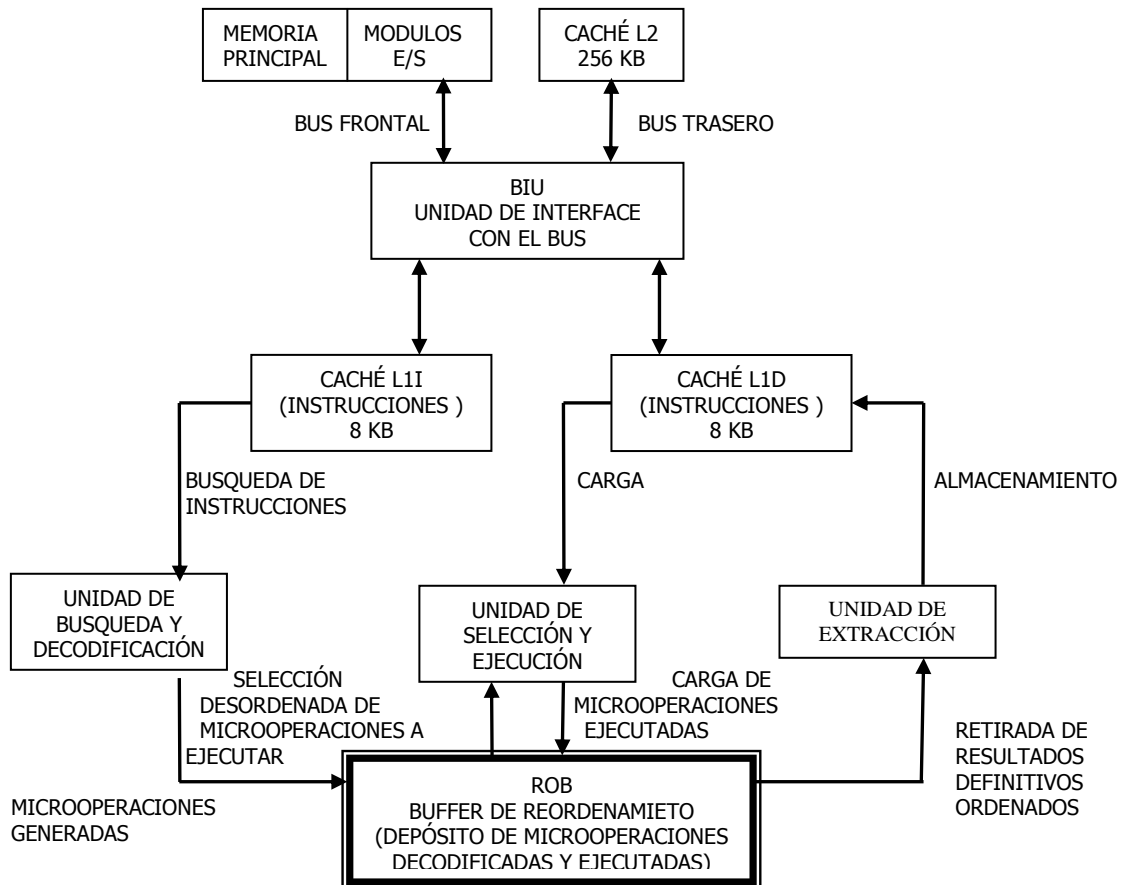


Figura 20.8 El ROB actúa como depósito de microoperaciones y de coordinación entre los tres bloques fundamentales

## 20.4.2- Orientación software del Pentium Pro

Como ya hemos comentado anteriormente, el Pentium Pro está diseñado con una arquitectura de 32 bits, por lo que su rendimiento es inferior en programas escritos en código de 16 bits. Debido a esto no se eliminaron los bloques de registros parciales, muy comunes en el código de 16 bits.

### EJEMPLO 3

Para estudiar esta característica se estudia el siguiente programa:

```
mov a1,c1
mov bx,ax
```

Como la primera instrucción se escribe a1 y con la segunda se lee ax, que es un registro ampliado que contiene a1. Esta secuencia de instrucciones, típica en el código de 16 bits, puede producir un bloqueo de muchos ciclos de reloj en el Pentium Pro, que serán los necesarios hasta que no se retire del ROB la microoperación que provocó la escritura de a1. En el diseño de código de 32 bits debe tenerse cuidado de evitar esta secuencia de instrucciones.

El Pentium Pro para incrementar fuertemente su rendimiento ejecutando las instrucciones fuera de orden, para tener ocupados al máximo todos sus recursos. Las situaciones de bloqueo de los registros parciales implican importantes retrasos, que no están contemplados en el código de 16 bits.

Este microprocesador está orientado a Sistemas Operativos de 32 bits como: Microsoft Windows NT, OS/2 y UNIX. Windows 95 sólo mejora de un 20 a un 30% sobre el Pentium.

Las normas más importantes que se deben seguir en la programación del Pentium Pro:

1. Evitar leer un registro extendido después de haber escrito una parte del mismo.
2. Evitar las bifurcaciones condicionales, usando lo máximo posible las instrucciones JMP, CALL y RET.
3. Alinear los datos.
4. Evitar el uso de código automodificable, que ocasionaría la eliminación del código residente en las caches cuando se mezclasen los canales de ejecución del procesador.