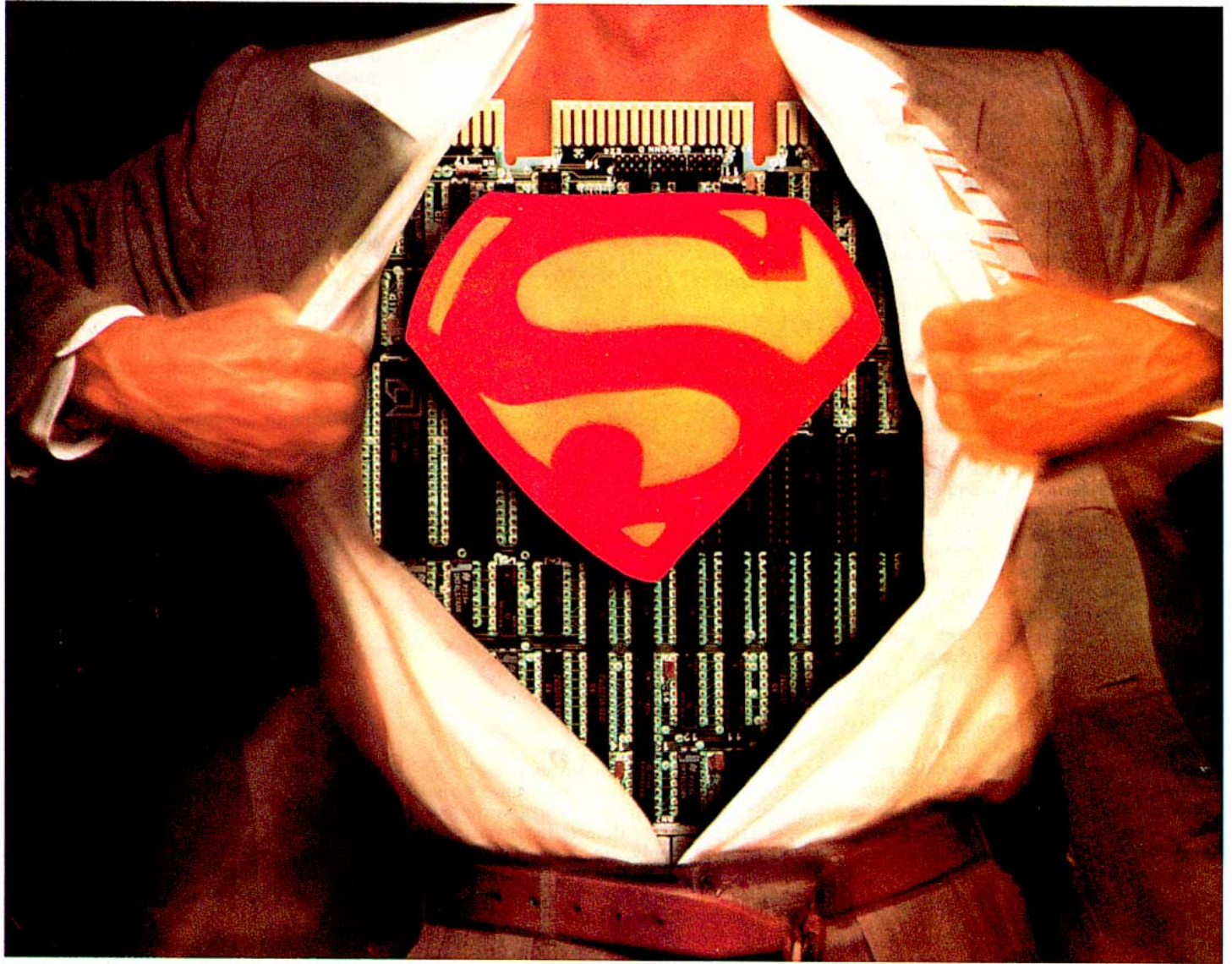




---

## Arquitecturas de los microprocesadores de 32 bits

---

La irrupción en el mercado de componentes electrónicos de las CPUs de 32 bits, junto con una gama aún limitada de productos adicionales como coprocesadores de punto flotante, MMUs, controladores de bus, de E/S, ... es un fenómeno que promete ser transcendental por su incidencia en los precios de los sistemas comúnmente denominados "superminis" (con el VAX 11/780 como modelo de referencia, quizás por sus aproximadamente 1 MIPS, frente a los 3 ó 4 MIPS de un 80386). Estos "Supermicros" reforzados, integrados y abaratados pueden ser los futuros equivalentes de los actuales ordenadores personales. Escribe sobre el tema DARIO MARAVALL.

Para describir las características más destacadas de las arquitecturas de la nueva generación de microprocesadores, los denominados 32 bits (referencia al ancho del bus de datos, aunque la tendencia mayoritaria es hacia un ancho igual para las direcciones) convendría establecer una definición de arquitectura de computador, un término muy utilizado pero que no siempre se interpreta de la misma forma.

Cuando se analiza un sistema de computador pueden adoptarse dos puntos de vista:

- El punto de vista funcional (qué es lo que hace el sistema).
- El punto de vista estructural (cómo lo hace; es decir, la particular forma de implementación del sistema).

Para algunos autores la arquitectura de un ordenador se reduce, exclusivamente, al aspecto funcional (en este caso la arquitectura correspondería a la máquina "vista" por el programador en lenguaje ensamblador). En el extremo opuesto se encontrarían aquellos que tienden a limitar la arquitectura a la estructura o implementación del sistema, aunque reconocen la importancia, y su incidencia sobre la estructura, de la funcionalidad del sistema. Una postura intermedia sostiene que la arquitectura engloba tanto a la función como a su implementación.

Personalmente opino, en relación a esta polémica, que es necesario, antes que nada, delimitar los diferentes niveles del aspecto funcional de un sistema de computador, teniendo en cuenta su organización modular tal como se muestra en la figura 1, en donde se representan los diferentes niveles de un sistema de computador.

En relación a esta figura, una función incluida en el sistema sería, por ejemplo, el manejo de ficheros en memoria secundaria. La implementación de esta función implica una larga cadena que incluye otras funciones, en distinto nivel, del sistema: memoria virtual (MV) y operaciones de entrada/salida (E/S), principalmente. Nadie habla del manejo de ficheros como una componente de la arquitectura, aunque sí de la memoria virtual y de la organización E/S.

La explicación está en el hecho de que el manejo de ficheros es una función de muy alto nivel, implementada como un conjunto de programas. Por supuesto que estos programas (instrucciones y datos), al ejecutarse, ponen en funcionamiento activo diferentes elementos del sistema; pero su funcionalidad es lo suficientemente abstracta e independiente de la forma específica de la organización interna del correspondiente computador como para no incluirla entre los elementos constitutivos de su arquitectura.

Por tanto aquí tenemos un punto de partida para nuestra particular definición de arquitectura: "Arquitectura de un computador es todo lo concerniente a la funcionalidad e implementación que le diferencia de cualquier otro computador (es decir, arquitectura".

“

## *Las máquinas RISC han surgido como una respuesta natural a los sistemas distribuidos y paralelos.*

”

Así, para fijar ideas, un minicomputador convencional (digamos un VAX 11/780), trabajando con un sistema operativo estándar y provisto de un dialecto del lenguaje LISP no incluiría en su arquitectura las primitivas de este lenguaje (abs, end, append, apply, etc., etc.). Sin embargo, estas funciones si formarían parte de la arquitectura de las denominadas máquinas LISP, cuya organización interna está especialmente diseñada para la implementación directa de tales primitivas, que pueden considerarse como parte del repertorio de instrucciones del procesador central o CPU.

Bajo esta perspectiva los microprocesadores, incluidos los más avanzados de 32 bits, son estrictamente hablando procesadores centrales altamente integrados (en una sola pastilla o chip) del tipo convencional de Von Neumann (ejecución secuencial de instrucciones); por tanto constituyen una parte, fundamental, de la arquitectura de un sistema de ordenador... pero sólo una parte. En este sentido, para poner un ejemplo, la arquitectura del miniordenador de 16 bits PDP-11 (mejor sería hablar de una familia) dispone de varias versiones de la CPU integrados en un chip (la más potente, el Micro J-11, incluye en el chip la gestión de memoria virtual). Análogamente, la arquitectura del VAX 11/780 también dispone de un microprocesador (éste ya de 32 bits), el Micro VAX II, que implementa en microcódigo el 80 por ciento del repertorio de instrucciones de la CPU materna (el resto se implementa con interrupciones software mediante una tabla vectorial). Se podrían poner muchos más ejemplos parecidos.

No obstante, a pesar de constituir una parte de la arquitectura, los microprocesadores o CPUs altamente integradas configuran de manera completa el resto de la arquitectura: gestión de la MV (cuando no está integrada directamente en el micro), procesamiento aritmético en punto flotante, E/S ... de forma que se puede utilizar correctamente el término de "arquitectura de un microprocesador". Esto es particularmente cierto en el caso de los microprocesadores que han surgido (sean de 8, 16 ó 32 bits) como elementos originarios y centrales de una arquitectura de computador, a diferencia de los Micro PDP-11, Micro VAX II, Micro IBM/370, Micro Eclipse ... que han aparecido como una versión integrada de CPUs de arquitecturas ya existentes.

Por tanto, los aspectos a considerar en la arquitectura de los microprocesadores son los típicos de una arquitectura convencional:

- Repertorio de instrucciones.
- Modos de direccionamiento.
- Estructura interna de la CPU (aspecto de implementación).
- Soporte material de la memoria virtual.
- Unidades especializadas de procesamiento, incluyendo los denominados coprocesadores.
- Organización de la entrada/salida.
- Sistema de buses, en particular el bus central.
- Organización de la jerarquía de memorias.

Dependiendo de la implementación, estos elementos se integran en uno o varios chips, que constituyen el microprocesador propiamente dicho y su correspondiente familia de componentes: coprocesadores, unidad de gestión de memoria virtual, controladores de E/S, ...

En los anteriores aspectos de una arquitectura, la función y su implementación están íntimamente ligadas. Así, respecto al repertorio de instrucciones, que es una parte fundamental de la funcionalidad de una arquitectura, su implementación configura radicalmente la organización interna del sistema: registros de la CPU, buses internos, organización de la E/S... De hecho, como señalábamos al comienzo del artículo, la importancia individual y las implicaciones mutuas de la funcionalidad y la implementación de una arquitectura constituyen temas polémicos, con posiciones encontradas.

Posiblemente una de las "confrontaciones dialécticas" más interesantes en esta problemática de la funcionalidad/implementación de las arquitectura sea la que enfrenta a los defensores de las CPUs RISC (Reduced Instruction Set Computer) frente a las CPUs "convencionales o CISC (la primera C por Complex)".

(Continúa en pág. 76)

## CUADRO 1

### Principios de la gestión de memoria virtual

El concepto de espacio de memoria virtual, o memoria virtual (MV) a secas, es bien conocido y se refiere al espacio de direccionamiento de un programa (instrucciones + datos), en oposición a la memoria principal (MP) entendida como el espacio de direccionamiento físico accesible directamente por la CPU. Puesto que los programas y datos deben estar almacenados, en toda su plenitud, en un soporte físico que es, precisamente, la memoria secundaria (MS), a veces se confunde erróneamente la MV con la MS, cuando ésta es exclusivamente su soporte de almacenamiento permanente.

La MV tiene su origen en la necesidad de permitir un tamaño de programas y datos superior a la capacidad de la MP de un sistema de ordenador (aunque existen algunas máquinas como el PDP-11, que disponen de una memoria física o principal muy superior a su memoria virtual), junto con la necesidad de repartir el uso del sistema entre varios programas. En resumen, es la respuesta a la necesidad de separar el espacio abstracto de direccionamiento de los programas respecto de su almacenamiento físico en MP.

Los dos procedimientos de implementación de MV, como es bien sabido, son (a) la paginación y (b) la segmentación, que pueden simultanearse. La paginación consiste en dividir en unidades fijas equivalentes, las denominadas páginas, la memoria física y la memoria virtual. Su objetivo es fundamentalmente optimizar el uso de la memoria principal por varios programas y tareas. La segmentación tiene una orientación menos "física", ya que sus objetivos son la re-ubicación de programas en MP, la protección de estos programas ante accesos indebidos y, en general, facilitar la organización lógica de programas y datos por parte de las memorias.

El precio que se paga por las prestaciones de una gestión de memoria virtual es la disminución de la velocidad de ejecución de los programas, al requerirse accesos adicionales a MP, como veremos con un ejemplo; aparte de la parafernalia software y hardware necesaria para su implementación y funcionamiento.

En la figura 11 se presenta un diagrama de bloques en donde se pretende ilustrar la relación entre MV y MF. Nos hemos permitido utilizar la terminología habitual sin traducir (MMU viene de "memory management unit"). DV y DF significan, respectivamente, direcciones virtuales y direcciones físicas.

En la citada figura se ha supuesto que el soporte físico de la gestión de la MV se encuentra fuera de la CPU (o microprocesador, en el contexto del artículo), aunque algunos micros (muy pocos) de 16 y de 32 bits la implementan en su propio seno.

La explicación de la figura 11 es la siguiente. El procesador central o CPU deposita una dirección virtual (instrucción o dato), que es transformada por la MMU en una dirección física, dependiendo de la particular implementación de MV del sistema. Esta DF dirección en memoria física (aquí se ha considerado una configuración general con memoria cache, MC, y memoria principal a secas) la correspondiente información (si está presente, por supuesto; en caso contrario actuaría una parte de la parafernalia soft/hard de que hablábamos antes).

Para fijar ideas en relación a cómo se realiza esta traducción de MV en MF, vamos a considerar el caso más general de memoria virtual con segmentación y paginación, tal como se representa en la figura 12.

La dirección virtual generada por la CPU está dividida en tres partes: segmento, página y desplazamiento dentro de la página. Cada parte con una longitud específica de bits, que restringe el número de segmentos ( $2^n$ ), de páginas por segmento ( $2^m$ ) y la longitud de una página, normalmente en bytes ( $2^r$ ).

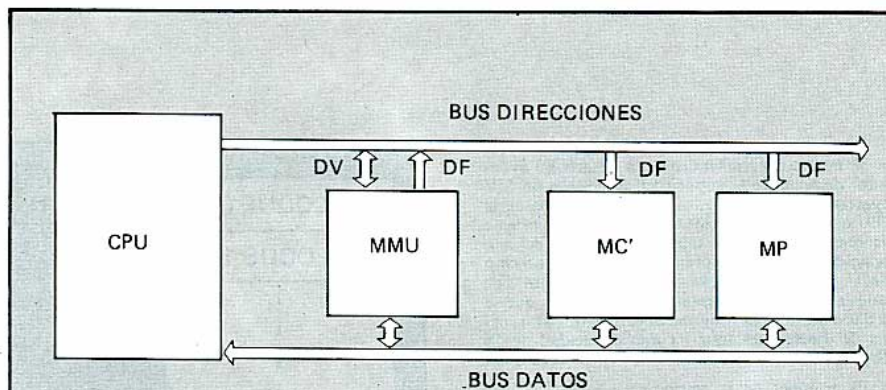


Fig. 1.1.— Esquema de bloques representando las direcciones físicas y direcciones virtuales.

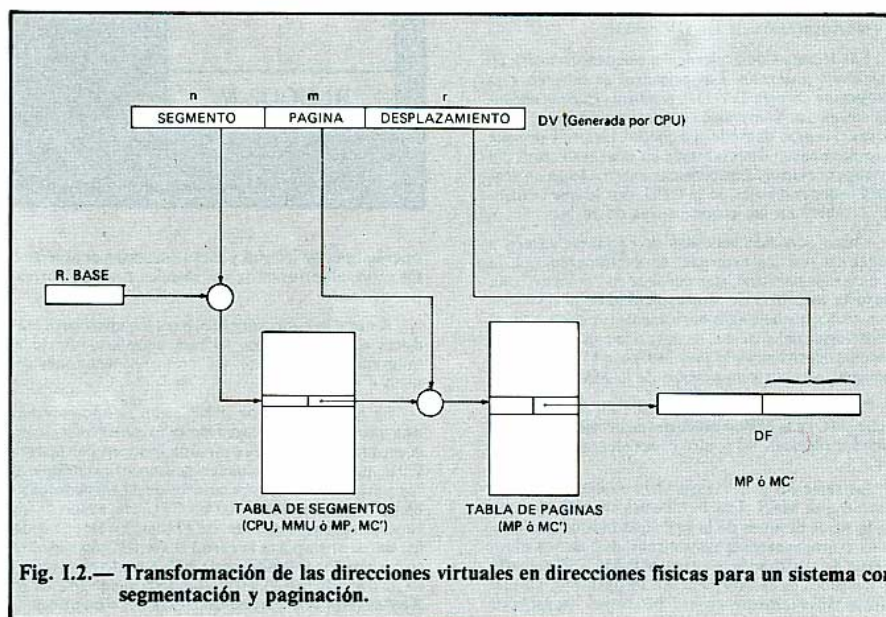


Fig. 1.2.— Transformación de las direcciones virtuales en direcciones físicas para un sistema con segmentación y paginación.

La tabla de segmentos (o banco de registros de segmentos cuando está implementada en hardware) contiene por cada segmento una entrada con la información concerniente a los tipos de acceso, uso y el estatus del segmento, así como la dirección física en MP (y en su caso, en la MC) del comienzo de la tabla de páginas del segmento. El número de la página referenciada dentro del segmento viene determinado por la correspondiente parte de la DV. Finalmente, aparte del estatus de la página referenciada, en la tabla de páginas se encuentra su dirección de comienzo en MP (si está cargada) de modo que junto con el campo de desplazamiento de la DV compondría la dirección física final.

Nótese que para longitudes habituales de página (un compromiso delicado y dependiente de los programas en ejecución) de 512 bytes -1 Kbyte- 2 kbytes, el número de páginas es tan elevado que la tabla de páginas de un segmento está implementada necesariamente en MP. De hecho, para fijar ideas, en un micro de 32 bits como el Micro VAX II trabajando con la versión estándar del sistema operativo VMS, al disponer de un espacio virtual por segmento, de 230 bytes y una longitud de 512

bytes por página, se tendría la friolera de 2 M páginas. Como cada entrada o descriptor de una tabla de páginas ocupa 32 bits, el almacenamiento de una tabla completa ocuparía 8 Mbytes, que es muy superior a la MP disponible en una configuración normal. Por tanto, las referencias intermedias a tablas de páginas pueden producir situaciones de "falta de página", un evento temible que ralentiza aún más la gestión de MV.

La conclusión lógica es que una arquitectura con soporte de MV debe incluir técnicas de aceleración de la traducción de DV en DF, que se comentan en otro recuadro del artículo.

Otro aspecto importante en relación a la MV, en el caso de los microprocesadores, es la integración o no de la MMU en la propia CPU. Su integración en el procesador central elimina los retardos de bus entre CPU-MMU, que pueden ser del orden de 100 ns, pero tiene el inconveniente de su falta de flexibilidad en la gestión de la MV, frente a su implementación en un chip exterior. Otro aspecto a tener en cuenta, evidentemente, es el compromiso entre la complejidad del resto de la CPU y la integración o no de la MMU en el mismo chip.

## CUADRO 2

### Técnicas de aceleración de direcciones

La práctica totalidad de los microprocesadores de 32 bits, comercializados o en desarrollo avanzado, dispone de la capacidad de gestión de MV segmentada y paginada, ya sea directamente en la CPU o con un chip adicional. Aunque la mayoría de los micros de 16 bits también disponen de esta capacidad (eso sí, siempre con el auxilio de un chip MMU, excepto el 80286 que integra una MMU, aunque sólo con segmentación), lo cierto es que se ha dado un salto de longitud del orden de 24 bits del bus de direcciones (que da una capacidad de 16 Mbytes de MV) a una longitud de 32 bits (4G bytes).

El disponer de la gestión de una MV de esa dimensión es uno de los factores que convierte a estos microprocesadores en auténticos supermini-computadores, capaces de soportar sistemas operativos de una considerable potencia. Como aconteció en su día con sus predecesores los superminis, es necesario que incorporen técnicas de aceleración de la transformación de las DV en DF.

Las técnicas de aceleración pueden dividirse en internas y externas. Las primeras se refieren a la traducción directa DV-DF, mientras que las segundas son exclusivamente de manejo de las direcciones físicas. Hemos llamado a aquellas internas porque se implementan directamente en la arquitectura del microprocesador. Las técnicas externas son en realidad independientes de la CPU, por lo que se utilizan también en las arquitecturas de 16 bits.

Como son más sencillas de exponer, vamos a comenzar por las externas; concretamente, con la técnica más habitual, que consiste en "colocar" una pequeña memoria de alta velocidad de acceso (denominado tradicionalmente memoria cache) en el bus exterior del sistema, en paralelo con la memoria principal convencional (ver la figura II correspondiente al recuadro de gestión de la MV).

La idea básica estriba en dividir la memoria cache (MC) y la MP en unidades equivalentes, denominadas bloques, tal como se representa en la figura II.1.

En relación a la figura II.1, evidentemente se cumple que  $M=N$ . Los  $N$  bloques en MC son una copia de  $N$  bloques de la MP (qué bloques en concreto es precisamente responsabilidad de los algoritmos de sustitución y carga MP-MC). El tamaño de los bloques es una elección importante, ya que influye directamente en el "hit ratio" (tanto por ciento de referencias de MP que se encuentran duplicadas en MC), pero al mismo tiempo incide negativamente en el tiempo de sustitución de los bloques.

Los tipos de correspondencia entre las direcciones de la MP y de la MC (Nótese que ambas toman la misma dirección física entregada por la MMU) pueden ser asociativa pura (cualquier bloque de MP puede colocarse en cualquier bloque de la MC), intermedia (los bloques de MP disponen de una asignación flexible dentro de un conjunto fijo de bloques de la MC) y, por último, directa, en la que los bloques de MP tienen una asignación absolutamente rígida en la MC.

La correspondencia asociativa pura produce los mejores rendimientos o hit ratios, como es fácil imaginar; pero su precio es mayor, al requerir una memoria asociativa quien a su vez aumenta algo el tiempo total de la MC. Otro posible inconveniente es la complicación adicional de los algoritmos de sustitución de los bloques en MC (Decimos posibles porque en la mera capacidad de poder aplicarse algoritmos de sustitución reside su interés). Por falta de espacio no comentamos ninguna de las importantes problemáticas relacionadas con las memorias cache: tamaño de la MC y de los bloques,

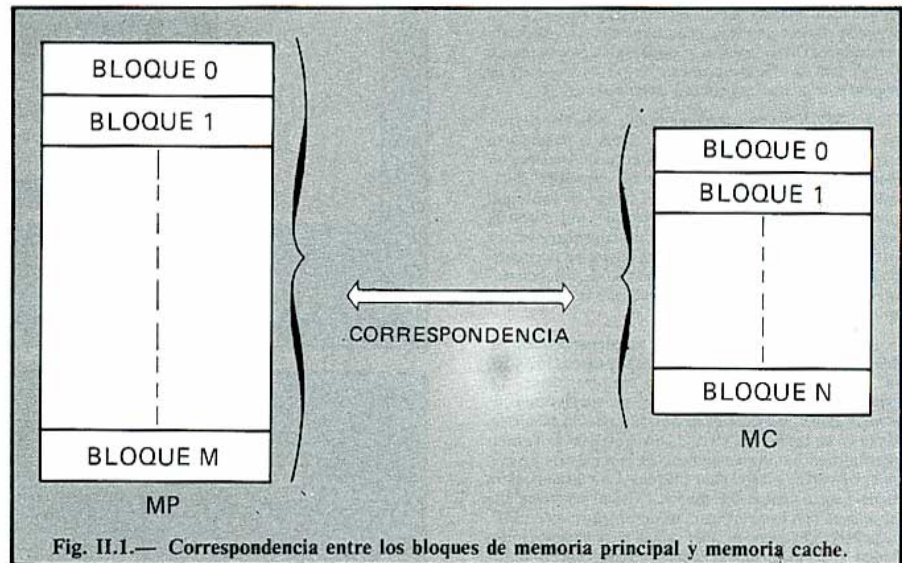


Fig. II.1.— Correspondencia entre los bloques de memoria principal y memoria cache.

operaciones de lectura y escritura, tasas de acierto o hit ratio, algoritmos de sustitución, tipos de correspondencia...

A nuestro conocimiento, para los microprocesadores de 32 bits sólo se han acoplado, hasta el momento, sistemas de MC con organizaciones directa e intermedia.

La organización de la MC, aún siendo como es una problemática propia de la arquitectura de un computador, es en gran medida independiente de la CPU, por lo que no pueden descartarse rendimientos para los microprocesadores de 32 bits compatibles a los conseguidos con las CPUs del rango de los supermini-computadores de 32 bits. En este sentido los datos que aporta la casa Hewlett-Packard sobre su ordenador HP 9000 serie 300, basado en el micro de 32 bits MC68020 y trabajando con una MC de 16 Kbytes (con una correspondencia del tipo intermedia), consistentes en alcanzar un hit ratio entre el 70 y el 75 por ciento parecen poco elevados para una MC de 16 Kbytes.

La aceleración que se consigue en la ejecución de programas merced a un sistema de memoria cache es muy considerable, ya que sus tiempos de acceso pueden moverse en márgenes de entre 4 y 10 veces más rápidos que los de las memorias principales convencionales. Incluso teniendo en cuenta que únicamente son "útiles" las referencias con lectura, puesto que las referencias con escritura (afortunadamente suelen ser menos frecuentes) obliga a acceder también a MP, salvo que se opte por un algoritmo de actualización en MP de los bloques de MC retirados, lo cual incidiría en un coste mayor y para ciertas aplicaciones en una degradación del sistema en los tiempos de ejecución.

Aún teniendo en cuenta la tremenda mejora del rendimiento de los tiempos de acceso a memoria que se consigue con la MC, conviene recordar que se trata de direcciones físicas (por esta razón la denominábamos técnica externa de aceleración). Para un sistema de memoria virtual con segmentación y paginación, cada referencia a memoria (dirección virtual) creada por la CPU puede implicar, como

mínimo, si la tabla de segmentos no está implementada en hardware, tres accesos a memoria física: (1) tabla de segmentos, (2) tabla de páginas y (3) dirección efectiva. Decimos como mínimo porque algunos sistemas, como el VAX 11/780 que sólo utiliza paginación, a fin de proteger los accesos tiene implementadas las tablas de usuarios (en cierto sentido pueden verse como segmentos) en el espacio de direcciones del sistema, por lo que todas las referencias de un programa de usuario implican dos accesos adicionales antes del acceso final o dirección efectiva.

Para evitar los accesos adicionales creados por la paginación, una solución generalizada consiste en implementar en el interior de la CPU (o de la MMU) una tabla hardware de traducciones DV-DF de, normalmente, las últimas referencias a memoria. Este mecanismo de traducción, denominado TLB ("translation look-aside buffer") se materializa con técnicas de memoria asociativa para acelerar el posible "matching". Por tratarse de una arquitectura de 32 bits bien conocida (y disponible como chip o microprocesador) vamos a comentar brevemente, a partir de la figura II.2, la TLB del VAX 11/780.

En la figura II.2 se aprecia que la TLB posee una memoria asociativa de correspondencia fija, utilizándose los 5 bits menos significativos de la página virtual como selector de uno de los 32 registros disponibles en modo usuario y en modo sistema. El resto de los bits actúa como tag o clave de acoplamiento (matching). Supuesto que la página virtual se encuentra referenciada en la TLB, se extrae su correspondiente dirección física que junto con el desplazamiento (9 bits menos significativos de la DV) constituye la DF para acceder a MC y MP.

Todos los micros de 32 bits disponen de este mecanismo interno de aceleración de traducción de las direcciones virtuales indispensable para compensar los accesos intermedios que necesariamente requiere un sistema con memoria virtual.

Así, para poner un par de ejemplos, el MC68020 dispone de una TLB exclusivamente para instruc-

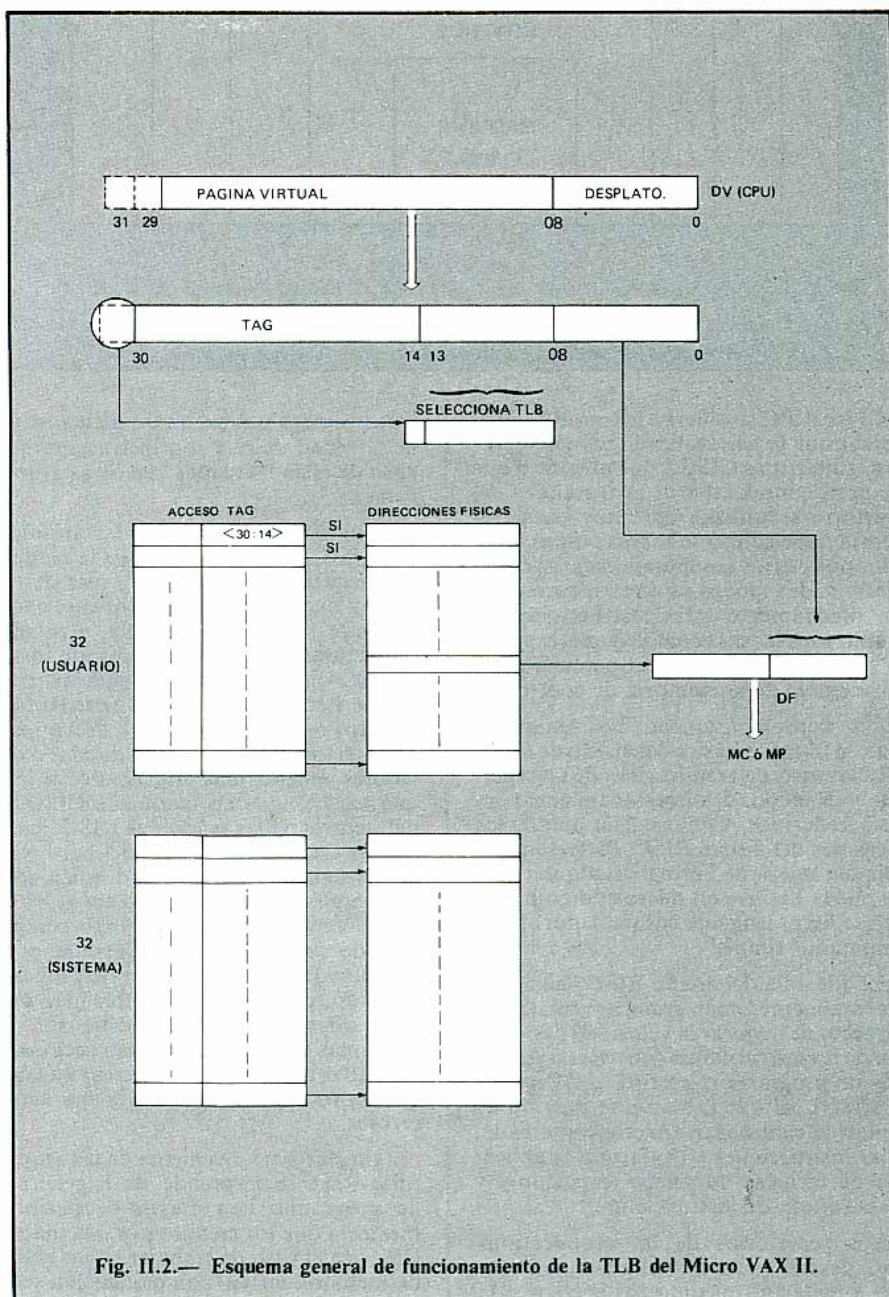


Fig. II.2.— Esquema general de funcionamiento de la TLB del Micro VAX II.

ciones, con una capacidad de 64 registros de direcciones. Análogamente, el 80386 tiene una TLB de 32 registros, válidos para direcciones de instrucciones y datos indistintamente. Tanto el MC68020 como el 80386 trabajan con segmentación y paginación, a diferencia del VAX 11/780 que no dispone de segmentación (salvo los cuatro espacios virtuales, dos para el sistema y dos para usuarios, que pueden

considerarse como segmentos fijos). No obstante, el "cacheado" de los segmentos es muchísimo menos exigente que el de las páginas, al ser unidades por lo general, de tamaño muy superior, siendo suficiente la existencia de un pequeño banco de registros de descriptores de segmentos en el interior de la CPU (o de la MMU, en su caso).

(Viene de pág. 72)

## LA DOBLE EMIGRACION HARDWARE-SOFTWARE O LA POLEMICA RISC-CISC

Todo el mundo ha oído hablar de la denominada crisis del software, que refleja el estado de inferioridad de la programación de computadores respecto al avance de la complejidad y potencia del hardware.

El diseño de las arquitecturas de los años 70 ha estado influido por el estado de ánimo creado por la crisis del software, de manera que la consecuencia ha sido el diseño de arquitecturas con repertorios de instrucciones cada vez más amplios y complejos (es decir, una emigración del software hacia el hardware).

En esos años 70 la implementación del repertorio de instrucciones se basaba casi universalmente en las denominadas memorias de control que almacenaban los microprogramas y que actuaban como "subrutinas" (en una memoria más pequeña y más rápida que la memoria principal) de las instrucciones del repertorio (\*). Merced a la evolución de la tecnología de las memorias, los diseñadores de una arquitectura podían complicar, sin problemas importantes de coste y de tiempos de ejecución, el repertorio de instrucciones.

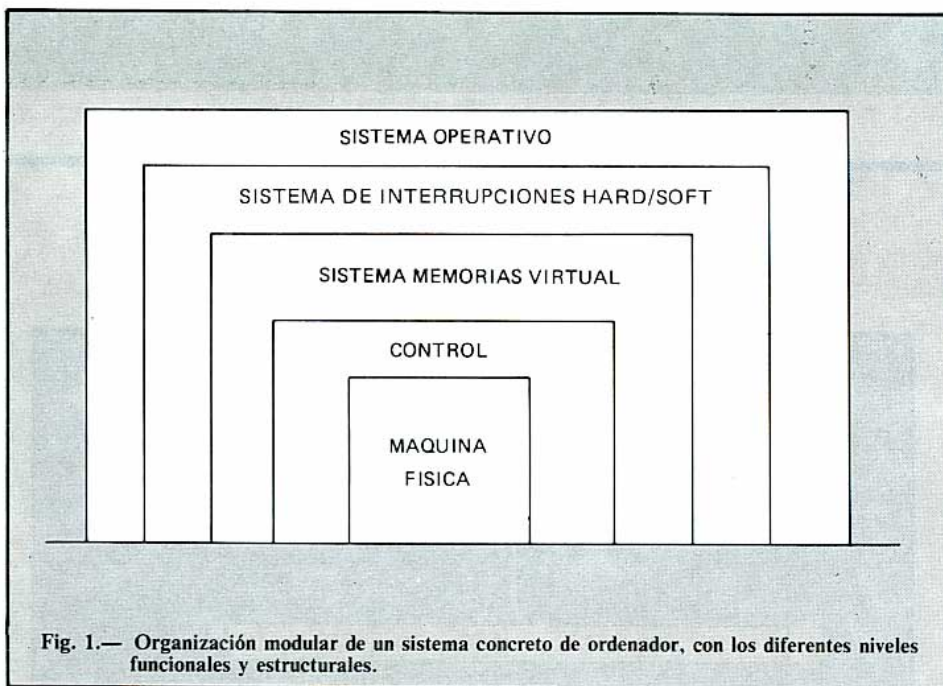
Por otra parte, al ser las micro-instrucciones mucho más rápidas que las instrucciones, esta emigración del software hacia el hardware constituía no sólo un procedimiento para contrarrestar la crisis del software (poniendo en manos de los programadores en ensamblador y de los compiladores una herramienta más versátil y potente), sino que además se disminuían los tiempos de ejecución de los programas re-escritos en un repertorio más amplio, al disminuirse el espacio de ocupación de los programas (aunque, como es sabido, no existe siempre una relación directa entre tamaño de un programa y tiempo de ejecución, ya que influye la velocidad de ejecución de las respectivas instrucciones).

Con la generalización del uso de las denominadas memorias cache, en cuyo seno se espera esté almacenada la información requerida por la CPU en cada momento, la hipótesis de que la velocidad de búsqueda de una microinstrucción sea varias veces superior a la de una instrucción comienza a tambalearse. Con tiempos de acceso (en realidad, con tecnologías comunes) equivalentes a las memorias de control, las memorias cache consiguen que los ciclos de las micro-instrucciones y de las instrucciones sean equivalentes (al menos, en el margen del hit ratio de una cache, que puede ser superior al 90 por ciento de las lecturas).

Por tanto aquella emigración pierde gran parte de su atractivo en lo que a tiempos de ejecución se refiere.

Otro punto de apoyo de los defensores

(\*) En puridad, debería hablarse de una emigración del software hacia el "micro-software" que, por otra parte, es más simple de depurar y no contiene los efectos laterales tan perniciosos en los programas de ordenador.



de CPUs RISC se refiere a los estudios estadísticos que se han realizado para numerosas arquitecturas CISC y que muestran que una parte considerable de instrucciones del repertorio se utilizan con muy poca frecuencia (especialmente en los códigos generados por ciertos computadores), así como determinados modos de direccionamiento. Son precisamente estas instrucciones infrutilizadas las más complejas del repertorio y las que, por tanto, ocupan una parte considerable de la memoria de control.

En consonancia con las anteriores ideas, la filosofía básica del diseño de CPUs RISC consiste en la utilización de un repertorio y de modos de direccionamiento simples y reducidos. Como señala uno de los defensores del diseño RISC, Patterson, parodiando la tesis de Turing: "cualquier cosa que pueda hacerse en micro-código puede hacerse en el lenguaje ensamblador de un computador simple".

La filosofía del diseño RISC tiene, fundamentalmente, unas grandes ventajas en el aspecto de mejorar la velocidad de ejecución de las instrucciones (no necesariamente de un programa re-escrito en el repertorio RISC), ya que (a) permite una implementación cableada, no microprogramada, de las instrucciones y (b) facilita la aplicación de técnicas de solape (pipelining) y pre-búsqueda de instrucciones.

Los repertorios de las arquitecturas RISC desarrolladas actualmente han tendido a evitar al máximo los accesos a la memoria principal (sólo existen dos instrucciones de este tipo: carga y almacenamiento), por lo que se basan en el manejo de un gran número de registros (una de las CPUs RISC tiene más de 500), todo ello con el objetivo de mantener la mayor parte del repertorio dentro de un ciclo de reloj (en el orden de los 10 MHz, por lo que son máquinas próximas a los 10 MIPS o millones de

instrucciones por segundo; dicho esto con la salvedad de que son instrucciones muchas de ellas "sencillas" en otras arquitecturas).

Como resulta evidente, las arquitecturas RISC, con su inversión de la tendencia mayoritaria derivada de la crisis del software y su emigración del hardware hacia el software (significativamente, algunas arquitecturas RISC han introducido instrucciones adicionales, más complejas, por la vía de microcódigo o, más sencillamente, de traps o subrutinas); como decíamos, las arquitecturas RISC obligan a diseñar compiladores mucho más sofisticados y lentos que los basados en máquinas CISC. Los detractores de las máquinas RISC insisten en que esta simplificación del hardware y la consiguiente complejidad adicional del software es su principal desventaja. No obstante, conviene matizar que (a) la complejidad de los compiladores creada por la máquina (no por el lenguaje de programación) no redunda en la complejidad de los programas y (b) precisamente son las máquinas complejas, con instrucciones sofisticadas las que tienden a crear un código ejecutable más susceptible a la aparición de errores.

Un claro inconveniente de las arquitecturas RISC se desprende de la generación de programas con mayor ocupación de memoria que los creados por una máquina CISC en el caso de trabajar en un entorno de memoria virtual con paginación ya que su ejecución se ve sensiblemente degradada.

No obstante, las máquinas RISC han surgido como una respuesta natural a la tendencia, cada vez más acusada, hacia el diseño y construcción de sistemas distribuidos y paralelos, en los que la principal problemática no es tanto la complejidad individual de los procesadores, o parte de ellos,

(Continúa en pág. 78)

## CUADRO 3

### Coprocesadores

Aunque los microprocesadores de 32 bits son CPUs de una considerable potencia, existe la posibilidad de aumentar su funcionamiento mediante los denominados coprocesadores.

Los coprocesadores son chips adicionales que están íntimamente ligados a la CPU, tanto en las líneas de control como de datos y direcciones. En esto se diferencia de las tarjetas compatibles (por ejemplo, un procesador vectorial diseñado ex-pro-peso para ampliar la capacidad de cálculo de una determinada CPU) que se comunican con el procesador central a través de un bus exterior normalizado. Las funciones tradicionales que han cubierto los coprocesadores son, básicamente, de tres tipos: cálculo numérico en formato de punto flotante, entrada/salida con acceso directo a memoria (DMA) y, por último, gestión de memoria virtual (MMU).

Los coprocesadores de DMA, tanto en las arquitecturas de 16 como en las de 32 bits, son imprescindibles para aprovechar convenientemente la potencia de la CPU y descargarla en las transferencias rápidas y voluminosas de E/S. En cuanto a los coprocesadores MMU, permiten implementar una gestión de memoria virtual, normalmente combinando segmentación y paginación. Muy pocos microprocesadores de 16 bits, cual es el caso del 80286, y hasta el momento dos de los micros de 32 bits (el 80386 y el Z80000, todavía en desarrollo) incorporan en su seno la gestión de memoria virtual. No obstante, National ha anunciado que, previsiblemente, su próxima CPU de 32 bits de la familia NS32000, concretamente el NS32C532, pudiera integrar la gestión de memoria virtual.

Si siguiendo con nombres, un microprocesador de 32 bits, el Clipper de Fairchild (diseñado por ingenieros con experiencia y mentalidad de "main-frame") posee dos coprocesadores MMU para instrucciones y datos por separado, cada uno de ellos con 4Kbytes de memoria cache. Curiosamente, el Clipper es el único micro que incorpora instrucciones de punto flotante en la propia CPU. Más sorprendente aún si cabe teniendo en cuenta que es una CPU RISC, con 101 instrucciones cableadas y 67 en microcódigo.

Los coprocesadores numéricos, de punto flotante trabajan directamente en hardware con datos en este formato. El control del flujo de instrucciones es llevado, lógicamente, por el procesador central o CPU; de manera que cuando aparece una instrucción de punto flotante se activa, concurrentemente con la propia CPU, el coprocesador numérico.

Este paralelismo permite una mayor velocidad de ejecución de los programas (aparte, por supuesto, del incremento de velocidad que proporciona esta "emigración" del software al hardware), aunque exige una mayor atención en la programación, para evitar errores de sincronización.

En la figura III.1 se representa un esquema de bloques muy simplificado del coprocesador matemático de punto flotante 80287, compatible con el 80386.

En la figura III.1, la línea unidireccional BUSY-TEST permite sincronizar la CPU con el coprocesador, ya que éste mantiene BUSY en nivel alto mientras está ejecutando una instrucción propia, desactivándola al finalizar. La línea bidireccional RQ/GT sirve para sincronizar el acceso al bus de datos y direcciones por parte de los dos procesadores.

La estructura interna de este coprocesador está basada en una pila de 8 registros de 80 bits (16 de exponente y 64 de mantisa) directamente manejables desde programa. Todas las instrucciones del coprocesador, 74 en total, tienen un subcódigo común de operación que las diferencia del repertorio de la CPU. Estas instrucciones son muy completas, ya que incluyen los siguientes aspectos: transferencia de datos entre MP y la pila del coprocesador, instrucciones aritméticas, instrucciones de com-

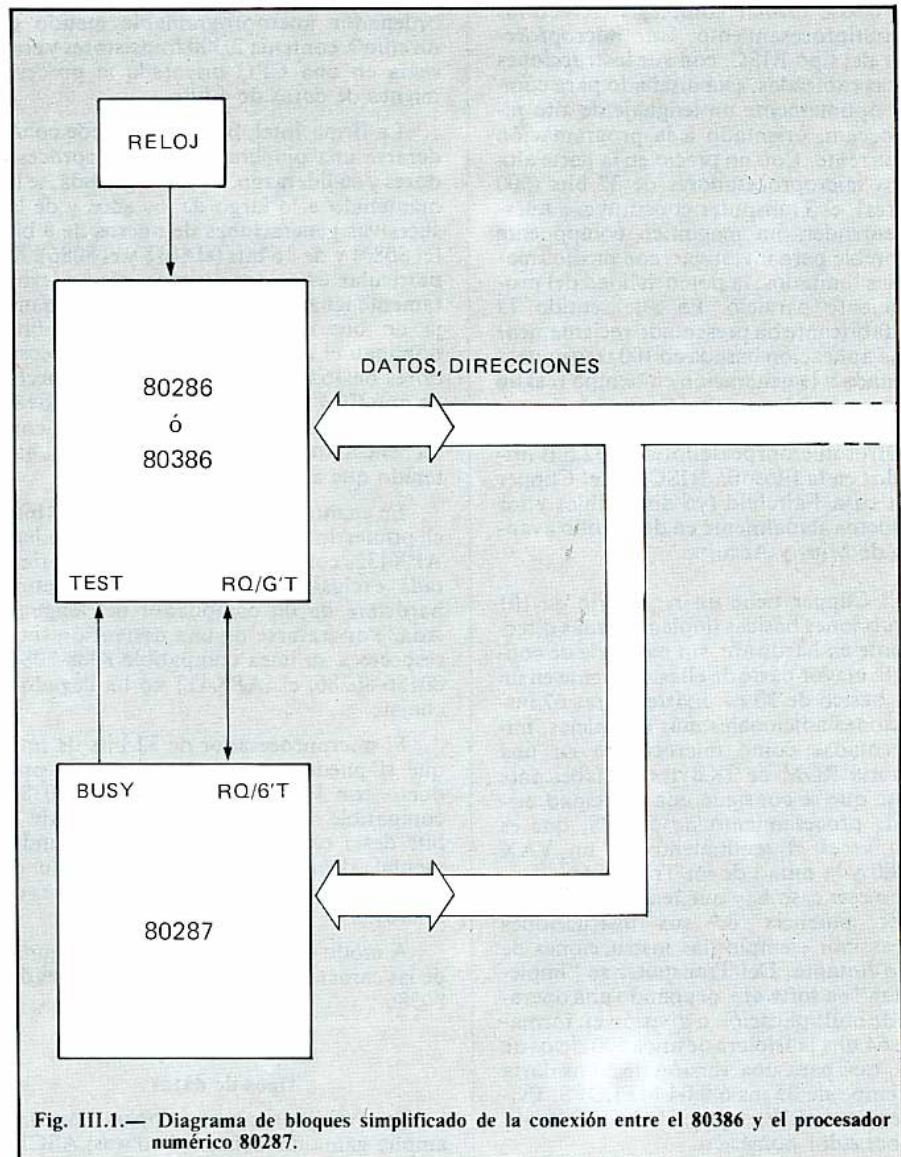


Fig. III.1.— Diagrama de bloques simplificado de la conexión entre el 80386 y el procesador numérico 80287.

ración, instrucciones de constantes, instrucciones de control y, finalmente, instrucciones trascendentes (operaciones trigonométricas y logarítmicas que, como es sabido, consumen un tiempo tremendo cuando se implementan sin coprocesador).

Todos los microprocesadores de 32 bits, ya comercializados o en desarrollo avanzado, se han diseñado con la perspectiva de incorporar un coprocesador numérico. Por ejemplo, el Micro VAX II (un chip que, según su fabricante, proporciona el ochenta por ciento de la potencia del supermini-computador VAX 11/780) dispone en su familia de componentes del Micro VAX 78132, que actúa como su acelerador de punto flotante o coprocesador numérico.

Por último, señalaremos que una de esas pequeñas empresas de la costa oeste de los EE.UU., Weitek Corporation, ha causado un tremendo impacto en el mercado de los coprocesadores numéricos de 64 bits, merced a sus chips WTL 1164 y 1165. Estos coprocesadores ofrecen una potencia de 4

MFLOPS (millones de operaciones de punto flotante por segundo), frente a los 0.7 MFLOPS del acelerador de punto flotante del VAX 11/780 o los aproximadamente 0.2 MFLOPS del 80287. No es de extrañar que National e Intel estén desarrollando chips de conexión de sus CPUs de 72 bits con estos supercoprocesadores numéricos.

(Viene de pág. 76)

sino la interrelación mutua de los procesadores. Un ejemplo lo constituye el Transputer, un microprocesador de 32 bits con un diseño no convencional basado en cuatro enlaces serie full duplex de alta velocidad (10 Mbits por segundo) que le permite conectarse con otros cuatro Transputers, pudiéndose formar complejas estructuras de multiprocesamiento. Este microprocesador del tipo RISC, con sus instrucciones básicas cableadas, está diseñado para compilar óptimamente un lenguaje de alto nivel, *occam*, orientado a la programación concurrente. Con un precio en la parte alta de los microprocesadores de 32 bits (500 dólares), el Transputer constituye, a nuestro entender, un magnífico componente disponible para investigar, con medios materiales limitados, la potencialidad del procesamiento paralelo. En este sentido, la casa fabricante ha presentado recientemente una aplicación basada en 100 transputers orientada a la generación en tiempo real de imágenes de alta resolución.

Otros microprocesadores de 32 bits inspirados en la filosofía RISC son el Clipper de la casa Fairchild (ya disponible) y los productos actualmente en desarrollo avanzado de Mips y Accorn.

El Clipper tiene un repertorio de 101 instrucciones básicas implementadas directamente en hardware, sin memoria de control; la mayor parte de ellas se ejecuta en un ciclo básico de 30 ns. Existen otras 67 instrucciones adicionales más complejas, implementadas como microcódigo en una memoria ROM de 2KBytes. El fabricante afirma que se consigue una velocidad media de procesamiento de 5MIPS, que es cinco veces el rendimiento de un VAX 11/780 y la mitad de un Transputer, aunque en este caso hay que tener en cuenta la menor potencia de sus instrucciones básicas: por ejemplo, las instrucciones de punto flotante. Del Transputer se "implementan" en software, ocupando una operación de multiplicación o división en formato de 64 bits la friolera de unos 500 ciclos de CPU, que para una versión de 50 ns daría un tiempo de 25 ms ó 0.04 MFLOPS. Evidentemente al Transputer le hace falta un coprocesador numérico.

El tiempo dirá si la filosofía RISC se extiende o no en la línea de los microprocesadores de 32 bits. Lo que sí es cierto es el interés de los fabricantes, particularmente los constructores de sistemas completos, hacia las máquinas RISC, como lo demuestran los diferentes proyectos (algunos ya comercializados, como es el caso del ordenador personal IBM RT) de arquitecturas RISC.

Con el objetivo de fijar algunas ideas mínimas en relación a las arquitecturas de micros de 32 bits, aún siendo conscientes de las dificultades y limitaciones del empeño, vamos a descubrir muy superficialmente algunos de los principales rasgos del 80386, un microprocesador de 32 bits ya disponible y sobre el que existe una buena documentación.

### EL MICROPROCESADOR 80386

El año 1971 puede considerarse como el comienzo, al menos en su vertiente comercial, de los microprocesadores con la aparición en el mercado de los componentes electrónicos del Intel 4004. Este circuito integrado, definido en su día como "un ordenador microprogramable metido en un chip", contenía 2.300 transistores y consistía en una CPU orientada al procesamiento de datos de 4 bits.

La firma Intel, por tanto, puede considerarse una pionera de los microprocesadores y su liderazgo, en cierta medida, se ha mantenido a lo largo de los años y de las sucesivas generaciones de micros de 8 bits (el 8080) y de 16 bits (el 8088 y el 8086). En particular estos dos últimos micros (exactamente iguales, excepto que el 8088 maneja un bus de datos exterior de 8 bits) dominan el mercado de los microprocesadores de 16 bits, en parte merced al hecho de que IBM lo ha elegido como CPU de su línea de ordenadores personales (y en consecuencia, toda la masa de compatibles han tenido que adoptar estos micros).

En cuanto a una arquitectura de 32 bits, el primer intento de Intel fue la familia iAPX432, compuesta por tres chips, orientada exclusivamente a servir de soporte hardware de un compilador del lenguaje Ada. Por tratarse de una desviación total respecto a su línea compatible 8088-8086-80186-80286, el iAPX432 no ha llegado a cuajar.

El microprocesador de 32 bits de Intel que sí puede considerarse como un producto con futuro comercial es el 80386, compatible con toda la línea anterior de 16 bits de la casa. Esta propiedad es fundamental, debido a la ingente cantidad de software en activo escrito para esta línea de micros.

A modo de pinceladas, veamos algunas de las características más sobresalientes del 80386.

#### Tipos de datos

Este microprocesador, opera con una amplia gama de formato de datos: ASCII, BCD, cadenas, enteros (con varias posibilidades, que incluyen hasta 32 bits) y reales de varios formatos, con un máximo de 80 bits). En este último caso se necesita el coprocesador 80287 para la implementación hardware directa de operaciones en punto flotante (ver el recuadro de coprocesadores).

#### Repertorio de instrucciones

El 80386 se ha diseñado bajo la idea de mantener una total compatibilidad con el 8086 y sus prolongaciones: el 80186 (que añade al 8086 un subconjunto de instrucciones de E/S, un reloj, un controlador programable de interrupciones y un doble canal para acceso directo a memoria) y el 80286 (una ampliación del 80186 orientada a la implementación de memoria virtual).

Su repertorio de instrucciones, por tanto, es prácticamente el mismo que el del 80286, excepto extensiones debidas a la ampliación en los tipos de datos.

#### Modos de direccionamiento

Los modos de direccionamiento efectivo del 80386 son los mismos que en las arquitecturas CISC estándar. Es decir, directo e indirecto en registro, directo e indirecto en memoria, relativo a base, inmediato, relativo a contador de programa, absoluto y varias modalidades de indexación.

#### Registros

El 80386 posee 8 registros de 32 bits de propósito general (datos y direcciones), manejables por programa, aparte de un registro de indicadores o flags, también de 32 bits. Cada uno de estos registros incluye en sus 16 bits menos significativos los correspondientes registros del 8086.

Existen 6 registros de 16 bits para el manejo de segmentos de 64 KBytes. El registro denominado CS direcciona el segmento de "código" o instrucciones, el SS direcciona el stack o pila de datos y los DS, ES, FS y GS direccionan sendos segmentos de datos. Estos 6 registros de segmentos trabajan en la modalidad de direccionamiento real y están pensados para mantener la compatibilidad con la línea de 16 bits. Cuando la CPU trabaja con direcciones virtuales, existen 6 registros descriptores de segmentos asociados, de manera análoga, con el código o programa, la pila y cuatro zonas de datos.

#### Organización de E/S

Este microprocesador soporta los métodos habituales de programación de E/S basada en interrupciones, tanto de origen software como hardware. Dispone de 64 KBytes para trabajar con E/S mapeada en memoria, así como de tablas en memoria principal para el uso de vectores de interrupción software y hardware.

También soporta, como es lógico, la programación de E/S mediante las técnicas de DMA, o modo canal, según la terminología del fabricante. Estas operaciones de E/S están completamente integradas en el 80386.

#### Organización del bus exterior

El 80386 tiene buses independientes de datos y de direcciones, cada uno de ellos con un ancho de 32 bits. Por razones de compatibilidad, se pueden manejar datos de 8, 16 y 32 bits en cada ciclo de bus. La unidad de conexión del bus, integrada en el micro, permite controlar las transferencias entre la CPU y otros dispositivos exteriores: coprocesadores, memorias, controladores de E/S...

Para aumentar el ancho de banda del bus existe un modo de funcionamiento de

(Continúa en pág. 80)

## CUADRO 4

### La importancia de un bus

Es innecesario resaltar lo vital que es, para cualquier nuevo microprocesador y en general para cualquier CPU con afán de mercado, el contar con un bus estandarizado, ya que es una condición necesaria (pero no suficiente) para su introducción y expansión comerciales. La existencia de un bus posibilita la aparición de tarjetas compatibles que refuerzan extraordinariamente la funcionalidad de la CPU. Ahí están para demostrarlo el Q-bus, el Unibus, el PC-bus de los ordenadores personales y, en concreto, el Multibus I, predecesor del Multibus II.

Este bus de 32 bits, anunciado en 1983 y con una versión disponible desde 1985, está diseñado con el objetivo de jugar el mismo papel que el Multibus I (actual líder del mercado de buses), pero en una línea de mucha mayor potencia, cual es la propia de CPUs de 32 bits. En las especificaciones del Multibus II han participado diecisiete empresas norteamericanas y europeas, estando actualmente en estudio su estandarización por el IEEE, lo que le coloca en competición directa con los otros buses de 32 bits en vías de estandarización: VMEbus, Futuribus y Nubus (actualmente sólo existe un bus de 32 bits plenamente estandarizado por el IEEE: el Fastbus). Por supuesto, la participación de aquellas empresas en las especificaciones del Multibus II no significa un apoyo incondicional, sino únicamente una forma de tantear su viabilidad e incidir en su gestión.

Una idea clave en el diseño de este bus ha sido su total compatibilidad con el Multibus I, por un lado, y la independencia del bus respecto al procesador o procesadores (ya que es un bus orientado a soportar multiprocesamiento) presentes en una configuración específica.

La arquitectura del Multibus II se basa en la existencia de cinco buses independientes tal como se recoge en la figura IV.1. en donde puede notarse que se ha independizado el acceso a memoria por parte de la CPU (bus iLBX II) y del controlador DMA.

La "separación" del Multibus II en varios buses independientes conlleva ventajas muy importantes y evidentes: el ancho de banda ideal del bus se mantiene sin dificultad (y en el Multibus II es de 40 Mbytes/s, trabajando con un reloj de 10 MHz); cada una de las funciones independientes del bus general se pueden implementar en buses individuales, con lo cual se abarata el coste de una configuración específica y, por último, se facilita el paralelismo de los diferentes procesadores presentes en una configuración.

El bus individual fundamental es el iPSB (de Intel Parallel System Bus), que está diseñado para servir de comunicación entre varias CPUs. En la anterior figura aparece una sola CPU así como un controlador DMA, pero pueden conectarse hasta 256, incluyendo procesadores DMA: existiendo un arbitraje de bus con un máximo de 20 peticiones simultáneas.

El iPSB es sincrónico, con un reloj de 10 MHz, que con 32 bits de datos permite el ancho de banda de 40 Mbytes/s que citábamos anteriormente. No obstante, en aras de la flexibilidad de uso, se pueden transmitir datos de 8, 16, 24 y 32 bits. La gestión del iPSB está centralizada en un módulo único de servicio, susceptible de integrarse en una tarjeta (el número de slots o ranuras previstas es de 20).

El bus iLBX (Local Bus Extension II) está pensado para evitar la degradación del rendimiento del bus que se produciría en un sistema multiprocesador en el que cada CPU utilizase un mismo bus para acceder a MP (aunque ésta se encontrase distribuida en módulos independientes para cada procesador). Así, con el iLBX II se consigue independizar los accesos a MP por parte de cada procesador. El ancho de direcciones del iLBX II es de 26 bits, lo que permite una capacidad de hasta 64 Mbytes de memoria local para cada CPU. El ancho de datos es

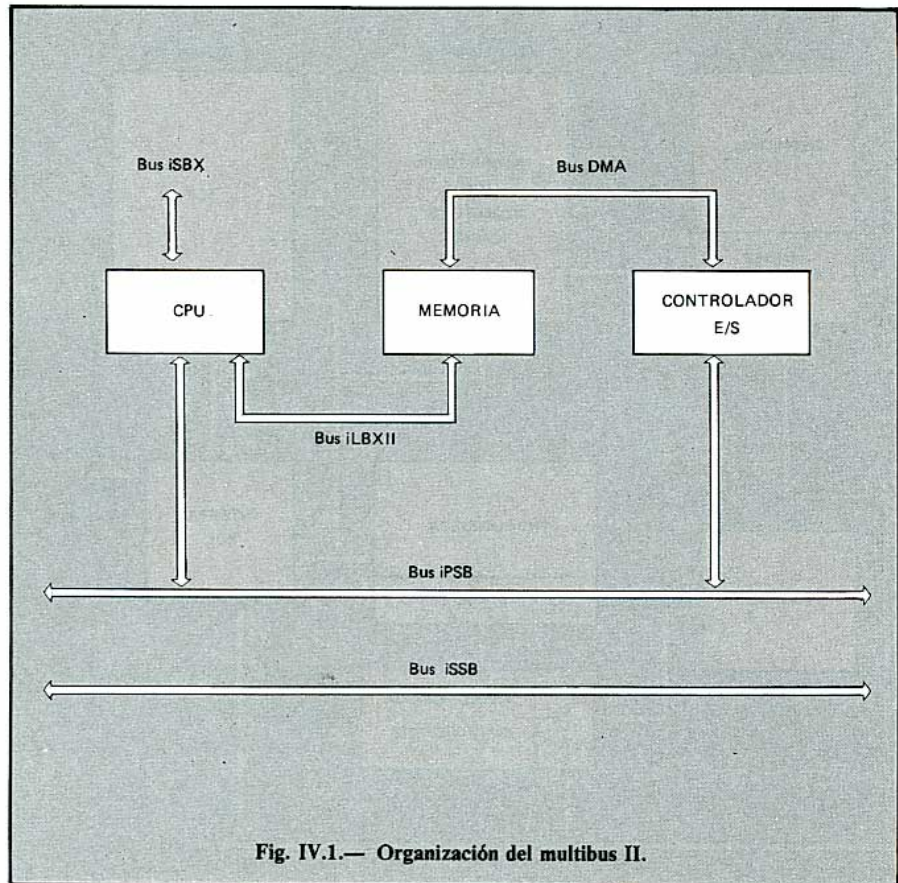


Fig. IV.1.— Organización del multibus II.

de 32 bits (aunque se pueden transferir datos de 8, 16, 24 y 32 bits), con una frecuencia de 12 MHz, que proporciona un ancho de banda de 48 Mbytes/s, difícilmente aprovechable con memorias RAM dinámicas de precio razonable, cuyos tiempos de acceso son superiores a los 100 ns, pero que encajan en los tiempos de acceso de memorias estáticas. La posibilidad de transferencia de bloques en modalidad de ráfaga, así como la superposición de los ciclos de direccionamiento y de datos, permite lograr un alto rendimiento del sistema CPU-MP a través de este bus local.

El tercer bus individual, el iSSB (Serial System Bus), es el más simple, ya que se trata de una conexión serie, con una frecuencia de 2 MHz (lo que proporciona una velocidad ideal de transferencia de 250 Kbytes/s). Está pensado como vehículo de comunicación de bajo coste entre varias tarjetas.

Los otros dos buses son una reproducción de los ya existentes en la arquitectura del Multibus I. El bus de E/S con DMA permite las transferencias de E/S con acceso directo a memoria (nótese su conexión directa con la MP), tales como discos, cintas magnéticas o periféricas de alta velocidad (por ejemplo, un digitalizador de vídeo trabajando en tiempo real). La velocidad de transferencia es de 8 Mbytes/s.

El bus iSBX (System Bus Expansion) conecta la CPU con tarjetas estándar de entrada/salida, especialmente orientadas a transferencias de datos en serie.

Finalmente, algunos comentarios de tipo comercial. Está claro que del mismo modo que los micros de 16 bits y correspondientes buses desplaza-

ron el interés de los usuarios y de los fabricantes respecto a la línea de los 8 bits, con la aparición de los primeros microprocesadores de 32 bits está produciéndose un efecto similar respecto a los productos de 16 bits. Por supuesto, el proceso de diseño, desarrollo y comercialización de buses de 32 bits con "vocación universalista" comenzó antes que apareciesen los primeros micros de 32 bits, pero con ello y el consiguiente abaratamiento de los sistemas de computador en la línea de los 32 bits está claro que estos buses (o los más capacitados) van a desarrollarse de manera espectacular.

El rey de los buses de 16 bits ha sido (desde que en 1982 desbancase al Q-bus) y sigue siendo el Multibus I, con aproximadamente 205 fabricantes de productos compatibles de los que existen más de 1.300. En la gama de los buses de 32 bits el líder es el VMEbus, orientado a la familia de micros de 16/32 bits de Motorola (68000, 68010 y 68020), con aproximadamente 147 fabricantes y 1051 productos (todos estos datos son de Electronics), con un crecimiento excepcional a lo largo de sus tres años de existencia.

Aunque algunos estudios predicen la "emigración" de los usuarios y fabricantes del Multibus I hacia el Multibus II, lo cierto es que nace con un fortísimo competidor. No obstante, Intel asegura que cuenta ya con 21 fabricantes y con 43 tarjetas para Multibus II.

Sólo el tiempo dirá cual de estos dos buses dominará el mercado de los 32 bits (con el de los 16 en estado marginal), ya que el resto de la gama apenas si cuenta, hoy por hoy, con apoyo industrial y comercial.

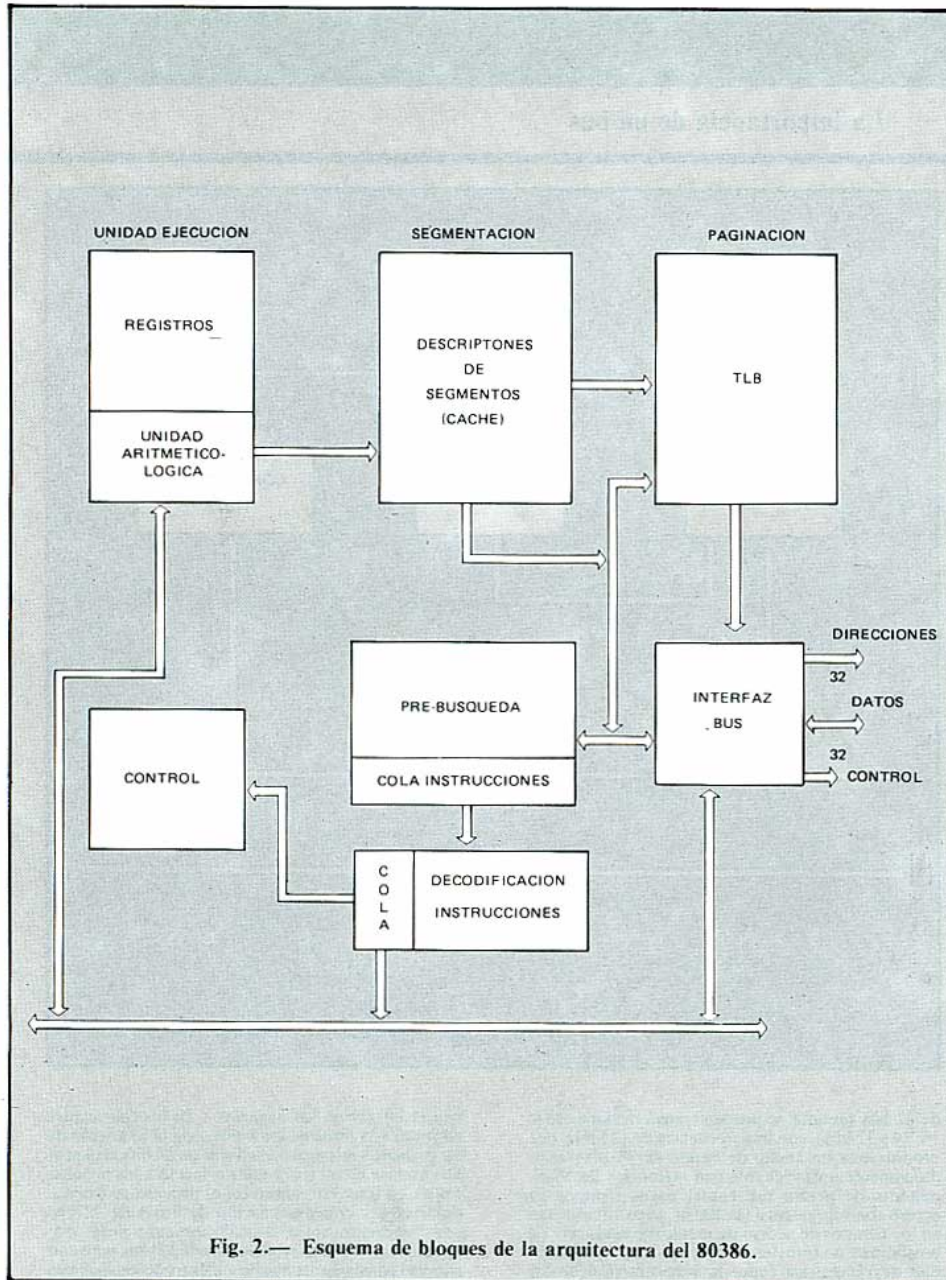


Fig. 2.— Esquema de bloques de la arquitectura del 80386.

(Viene de pág. 78)

tipo pipeline, mediante el cual se puede generar la siguiente dirección de memoria antes de finalizar la transferencia de datos en curso.

Por su importancia estratégica en la arquitectura de un sistema completo de ordenador, especialmente desde un punto de vista comercial, Intel comenzó en 1983 un proceso de diseño y estandarización de un bus de 32 bits: el Multibus II (véase el correspondiente recuadro).

## Gestión de memoria virtual

La gestión de memoria virtual (MV) del 80386 se basa en la segmentación y paginación, aunque se pueden eliminar cualquiera de los dos mecanismos de traducción de direcciones virtuales a físicas. De esta ma-

nera se puede trabajar con cuatro modalidades.

1. Memoria sin segmentación ni paginación.
2. Memoria paginada.
3. Memoria segmentada.
4. Memoria segmentada y paginada.

La tabla de descriptores de segmento y de directorios de páginas están implementadas en software, es decir, residen en memoria principal (o secundaria, si no están activas). No obstante, para acelerar las referencias a memoria, la CPU tiene "cacheados" los descriptores del segmento de programa, de la pila de datos y de hasta

cuatro segmentos de datos, en sendos registros internos. Análogamente, en la unidad de paginación de la CPU existe un buffer de referencias de páginas (TLB, en la terminología habitual) con 32 traducciones de dirección virtual a física.

El 80386, además de esta potente gestión de MV típica de arquitecturas de 32 bits, incluye una característica adicional pensada para soportar sistemas operativos propios de este segmento de CPUs, cual es la conmutación de tareas. A tal fin, almacena en MP el "contexto" completo de una tarea: contenido de los registros de propósito general, contador de programas, registros y descriptores de segmento... todo esto mediante una instrucción de su repertorio.

## Detalles de la implementación

El esquema de bloqueo de la arquitectura del 80386 se muestra en la figura 2. Esta implementación está orientada a lograr el máximo paralelismo (dentro del esquema secuencial clásico) en la ejecución de instrucciones.

En un procesador como el 80386 la principal causa de incremento del tiempo de ejecución de programas estriba en los accesos intermedios a MP que requiere el proceso de traducción de las direcciones virtuales en direcciones físicas, ya que se precisan 3 accesos a MP por cada dirección virtual, que con los tiempos actuales de acceso a una memoria RAM de coste medio constituye una clara ralentización de la CPU. Para reducir este efecto se ha implementado en el chip un banco de registros de conversión dirección virtual-dirección física (un TLB o "translation look-aside buffer") con 32 elementos de conversión. Para una referencia presente en la TLB (un "hit") la conversión se realiza en medio ciclo de reloj. Esta conversión sólo afecta a la paginación, no a la segmentación.

Otro incremento de la velocidad se consigue con el solape de las fases independientes de búsqueda y ejecución de instrucciones, existiendo una cola de instrucciones a tal fin. Para aumentar la velocidad de la etapa de búsqueda de una instrucción, existe un registro de puntero de la siguiente instrucción que contiene la "actualización lineal" (es decir, de dirección física) del contador de programa, con lo que se evita pasar por las fases de conversión de direcciones virtuales salvo cuando exista un salto de página.

Con estas técnicas de aceleración, el fabricante afirma que se pueden conseguir, para una tasa media de 4 ciclos de reloj por instrucción, unos rendimientos de entre 3 y 4 MIPS (millones de instrucciones por segundo), ya que existen dos versiones de 12 y 16 MHz.

## REFERENCIAS BIBLIOGRAFICAS

A nuestro conocimiento, actualmente existen siete familias de microprocesadores de 32 bits (por orden

alfabético de su denominación) disponibles comercialmente.

- Clipper, de Fairchild, del que sólo existen referencias genéricas de presentaciones y reseñas en revistas especializadas.

- iAPX-80386, de Intel, del que existe un manual de referencia del fabricante.

- MC68020, de Motorola, con un manual de usuario editado por su fabricante, así como de su coprocesador numérico, el MC78881. La MMU no está disponible aún comercialmente.

- Micro VAX II, de Digital. Por tratarse de una versión casi completa de la CPU del VAX 11/780, existe una bibliografía muy amplia disponible. Quizás la referencia más didáctica sea el libro: H.M. Levy, R.H. Eckhouse, Jr., "Computer Programming and Architecture. The VAX 11". Digital Press, 1980.

- NS 32000, de National Semiconductor. En realidad se trata de una familia de CPUs ascendentes, siendo la última versión el NS 32332. Existe un manual de referencia del fabricante, así como de su MMU, la NS 32082, y de su coprocesador numérico, el NS 32081.

- Traspouter, de Immos. Su fabricante ha editado unos manuales introductorios muy didácticos de la CPU y del lenguaje de alto nivel occam.

- WE 32100, de ATT. Aunque su fabricante no tiene tradición en la línea de los microprocesadores, actualmente es el que, posiblemente, ofrezca con este micro la más completa familia de componentes: MMU (WE 32101), controlador de memoria dinámica RAM (WE 32103), controlador DMA (WE 32104) y coprocesador numérico (WE 32106). Todos ellos y en particular la CPU poseen su correspondiente documentación.

El Z80000, de Zilog, está en fase muy avanzada de pre-comercialización, existiendo un manual de referencia preliminar editado por Zilog. En el IEEE Micro de Diciembre de 1985 apareció un artículo descriptivo de los rasgos generales de su arquitectura.

De los numerosos productos en desarrollo avanzado (Accorn, AMD, Data General, Fujitsu, Hewlett-Packard, Hitachi, IBM, NCR, NEC, Sperry..., incluso McDonnell Douglas y TRW), no se dispone más que de referencias generales no convalidadas oficialmente.

## RESUMEN DE LA SITUACION ACTUAL DE LOS MICROS DE 32 BITS

La irrupción en el mercado de componentes electrónicos de las CPUs de 32 bits, junto con una gama aún limitada de productos adicionales como procesadores numéricos, MMUs, controladores de bus, de E/S... es un fenómeno que promete ser trascendental por incidencia en los precios de los sistemas comúnmente denominados de la gama "superminicomputador" (con el VAX 11/ 780 como el modelo de referencia, quizá por sus aproximadamente 1 MIPS). Estas CPUs de 32 bits (las cuatro o cinco ya comercializadas) se mueven todavía en precios altos (300-500 dólares, frente a los 20 dólares de una típica de 16 bits), pero estos precios van a caer en cuanto se lancen al ruedo la docena larga de los otros micros de 32 bits en desarrollo y, por otra parte, aumente la demanda de estos componentes. ■



**Dario Maravall Gómez-Allende**, es Doctor Ingeniero de Telecomunicación y Profesor Titular de Universidad. Ha publicado más de 50 trabajos técnicos y profesionales en revistas y congresos nacionales e internacionales.

Actualmente co-dirige un proyecto interdisciplinario financiado por la Fundación Ramón Areces en el campo de la automatización industrial avanzada (percepción visual en robots).

Sus áreas de trabajo y experiencia son: teoría y práctica de la ingeniería de control, máquinas inteligentes y lenguajes y sistemas informáticos.

## INDICE DE PUBLICIDAD

|             |            |                 |                 |
|-------------|------------|-----------------|-----------------|
| INTELSA     | 1          | ENSA            | 32              |
| AMPER       | 2          | PACISA          | 38              |
| SESA        | 4          | TELETTTRA       | 42              |
| REDISLOGAR  | 7, 20 y 25 | TELEVES         | 44              |
| CESELTA     | 8          | SINTEL          | 54              |
| LOBER       | 10 y 94    | DGEI            | 60 y 61         |
| BIT         | 12 y 57    | SITRE           | 62              |
| EUROTELECOM | 13         | AMITRON         | 2ª y 3ª Portada |
| SELCO       | 16         | INISEL          | 4ª Portada      |
| EUROTRONICA | 22         | HEWLETT PACKARD | Entre 16 y 17   |
| ITECESA     | 30         |                 | Entre 64 y 65   |