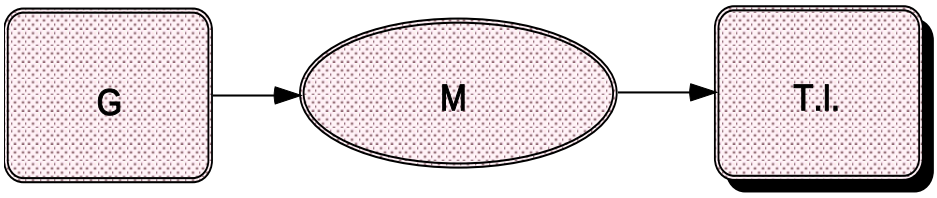
Hardware

La tecnología microelectrónica ha dado un impulso tan espectacular a la estructura y arquitectura de los ordenadores que ha cambiado en pocos años varias veces los órdenes de magnitud de sus prestaciones y su complejidad. Cada uno de los temas que se tocan en este capítulo, y otros que ni siquiera incluímos, constituyen de por sí un dominio especializado. Por esta razón, nos hemos limitado a abordarlos de forma que se resalten los aspectos que tienen que ver con conceptos como ***integración***, ***niveles jerárquicos***, ***capas***, ***clasificaciones***, ***categorías de máquinas***, ***relaciones hardware-software*** *y* ***paralelismo***.



1. Introducción

En los capítulos anteriores hemos ido desarrollando un marco conceptual sobre complejidad y sistemas que vamos a tomar ahora como punto de partida para repasar las tecnologías hardware. Normalmente, cuando se habla de hardware, nos referimos a los ordenadores en su aspecto físico, cómo están construidos, características de diseño, materiales empleados, organización interna, etc.. Aunque a la hora de trabajar con ellos estas características queden ocultas por diversas capas de máquinas virtuales y el software de aplicación, es fundamental para el ingeniero poseer unos conocimientos básicos sobre arquitectura y aspectos físicos del computador. Esto no es un gran descubrimiento pero sí requiere un análisis detallado para evitar aproximaciones simplistas, pues el hardware se ha convertido en una tecnología muy compleja que a fuerza de avanzar a pasos agigantados se ha diversificado de tal manera que es muy dificil obtener una perspectiva global. Hace muy pocos años los ordenadores diferían en muy pocas cosas y casi se podían contar con los dedos de la mano las diferentes aproximaciones prácticas al problema de la computación. Sin embargo, hoy en día existen ordenadores comerciales que funcionan basándose en interpretaciones radicalmente distintas y la creciente tendencia a integrar software y hardware acentúa aún más la diversificación.

Si se intenta dibujar una perspectiva generalista, no hay más remedio que recurrir a las herramientas que hemos ido detallando en capítulos anteriores. No en vano, muchos de los trabajos sobre complejidad y sistemas comentados proceden de investigadores y científicos muy relacionados con los ordenadores. Se puede cerrar un bucle imaginario que va desde el estudio de la complejidad a los ordenadores y de éstos a la complejidad, pues, como se ha dicho, "el ordenador es el instrumento de las ciencias de la complejidad" [Pagels, 1989, p. 36]. En el presente capítulo intentaremos trazar un enfoque sistémico y generalista de las tecnologías de hardware haciendo hincapié en la relación que tienen muchos de los aspectos que trataremos con las nociones de complejidad y sistemas. La referencia a ideas ya mencionadas será obligada e invitamos al lector a encontrar otras nuevas y a interpretar las tecnologías que presentamos a la luz del estudio de la complejidad.

Es una opinión extendida dentro del mundillo informático que las auténticas "revoluciones" tecnológicas siempre vienen y han venido de la mano de avances en el hardware. El resto de la tecnología, incluido el software, se desarrolla y evoluciona según la pauta que marcan los equipos y plantean la innovación en función de éstos. Existen muchos ejemplos que demuestran lo acertado de esta opinión, uno de ellos es el creciente interés en la publicación electrónica dada la aparición de muchos programas que aprovechan los nuevos monitores gráficos (VGA en los pc's, por ejemplo) y

las prestaciones de las impresoras láser, ahora con un precio suficientemente bajo como para pensar en conectarlas a ordenadores personales.

Pero un prueba aún más palpable es que el software va siempre por detrás del hardware, en investigación, desarrollo y productos comerciales. Muchos equipos ven hipotecadas sus prestaciones por la falta de programas adecuados, uno de los pilares básicos de la industria de los pc's es la cantidad de programas disponibles, un hecho que actúa como serio obstáculo para introducir nuevos modelos de ordenador. En ordenadores de la gama alta (superordenadores y máquinas especializadas) el problema es similar y gran parte de la investigación se ve frenada por la falta de lenguajes de programación adecuados y herramientas que puedan sacar partido a las nuevas arquitecturas. Este fenómeno se puede comprobar en los ordenadores paralelos donde existe ya una cantidad apreciable de hardware comercial y, sin embargo, el estado del arte del software paralelo no permite utilizar adecuadamente muchas de ellas.

Y como demuestra el ejemplo que mencionábamos ahora sobre la publicación electrónica, las aportaciones del hardware no se producen sólo en el terreno de la arquitectura de ordenadores, sino también en el de coprocesadores, impresoras, monitores gráficos, redes de comunicación, medios de almacenamiento y todo tipo de periféricos. Gran parte de las ideas sobre hipertexto, por ejemplo, serían impensables si no fuera por el aumento de capacidad de los discos duros y la aparición de los discos ópticos, muchas aplicaciones de diseño asistido no se hubieran desarrollado de no existir los periféricos adecuados, desde monitores de alta resolución hasta *plotters* y tableros gráficos, etc.

2. Hardware-Software

El estudio de los ordenadores se ha dividido tradicionalmente en hardware y software. La primera disciplina se centra en los recursos físicos para la computación y va desde el estudio de los dispositivos electrónicos hasta las arquitecturas de los ordenadores. El software se suele identificar con la programación de esos recursos físicos para conseguir que realicen determinadas tareas. Sin embargo, la frontera no es muy clara en bastantes ocasiones, sobre todo dada la tendencia a integrar software y hardware, un punto sobre el que volveremos más adelante.

2.1 Máquinas virtuales o el hardware fantasma

La separación existente entre el hardware y el software plantea una dificultad evidente en el manejo de los ordenadores. Esta es una de las primeras ideas que se estudia en "fundamentos de ordenadores". En el hardware se manejan, dependiendo del nivel, desde coeficientes de amplificación y capacidades de sustrato a ciclos de bus, velocidades de acceso y capacidad de memoria. En el software, el nivel inferior maneja direcciones de memoria, estado del procesador y "flags" de condiciones, componentes que definen un nivel muy por encima de lo que es el hardware. Existe un nivel intermedio, el de microprograma, que es difícil de encajar tanto en el hardware como en el software y en el que se produce la simbiosis entre ambas vertientes de la computación. A partir de él se empieza a desarrollar el software en sucesivos niveles, según la aproximación clásica. Cada uno de esos niveles se interpreta desde el inmediato superior como una máquina, como si fuera hardware, de ahí el nombre que reciben de máquinas virtuales.

Esta estructura "de cebolla" es la que aparece en la siguiente figura con la distribución en capas de un sistema operativo:

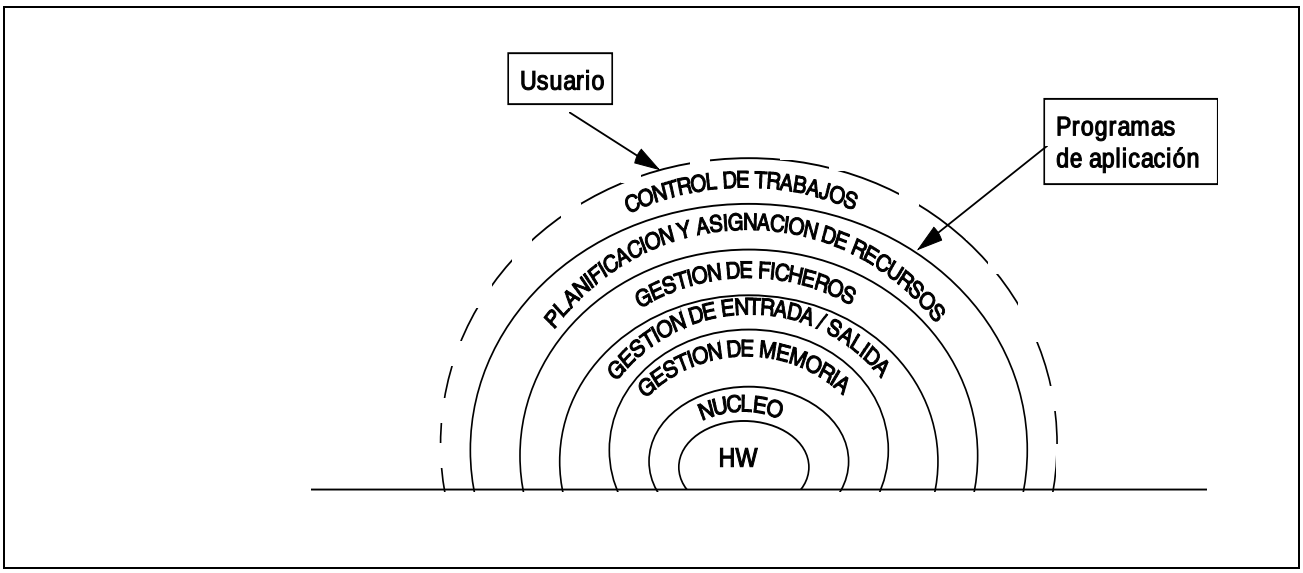


Fig.1 Sistema operativo construido a base de capas, o máquinas, virtuales. [Fernández, Sáez Vacas, 1984, p. 717].

Esta misma estructura jerárquica en capas la vamos a encontrar en el hardware, tal y como recogíamos en un capítulo anterior (ver "Sistemas, visión estructural y visión funcional", en particular la figura 7).

Hardware y niveles de complejidad

En el capítulo dedicado a los marcos conceptuales recogíamos un modelo de complejidad propuesto por Sáez Vacas [1983] que planteaba tres niveles en la informática. Un primer nivel es el de los objetos o elementos aislados, como puede ser un algoritmo, un circuito electrónico, etc. Estos elementos así considerados exhiben una complejidad característica que se denomina complejidad de los elementos aislados. Por encima de este nivel está la complejidad de los sistemas, es decir, la complejidad que aparece cuando esos elementos aislados se combinan para formar un todo del que interesa su comportamiento global. Un ejemplo es un ordenador personal que puede verse como un sistema compuesto de un gran número de elementos. Otro puede ser un chip. El último nivel de este modelo introduce un tipo de complejidad diferente a la que aparece en los otros dos, es la complejidad antropotécnica que aparece como resultado de la interacción entre los sistemas tecnológicos y la sociedad. En este nivel es donde se estudian problemas tan importantes como las interfaces de usuario, los factores humanos en la implementación tecnológica, los cambios organizativos como consecuencia de la utilización de la tecnología, etc.

Cada una de las capas tiene un lenguaje propio y define un dominio de trabajo único para ese nivel. Esta jerarquía es importante porque da una idea muy clara de cómo se orquestan los diferentes niveles y las distintas tecnologías dentro de un mismo campo, la informática, y puede ser muy útil para interpretar los avances y desarrollos que se producen. Por ejemplo, se puede pensar en integrar varias capas en circuitos electrónicos (como de hecho está sucediendo cada vez más), o trasladar la complejidad de una capa a otra, simularlas en un ordenador distinto, etc.

La razón para utilizar este tipo de jerarquías ya la vimos en el capítulo dedicado al tratamiento de la complejidad, pero conviene recordar que estas clasificaciones nos proporcionan un mapa muy útil del terreno por el que nos movemos y nos permiten establecer relaciones y niveles bien diferenciados para poder estudiarlos por separado.

2.2 Integración de Hardware y Software

Dada la semejanza en la estructura adoptada para interpretar el hardware y el software, se puede pensar en cambiar los límites establecidos entre ambos campos. Se podría, por ejemplo, implementar en hardware alguna de las capas que forman el sistema operativo, o incluso ir más lejos e implementar directamente el sistema operativo en forma de microprograma. En la realidad se ha avanzado aún más, existen ordenadores cuya arquitectura está directamente orientada a un lenguaje de programación concreto y las instrucciones se ejecutan sobre la máquina directamente.

De la misma forma, se puede pensar en reproducir mediante programas diferentes niveles de hardware, como de hecho se hace en las máquinas que simulan un ordenador distinto. La estructura en capas permite pensar en un proceso de computación en el que la frontera entre el hardware y el software se deja a elección del diseñador, de ahí la importancia de tener una perspectiva de conjunto.

Más adelante volveremos sobre las arquitecturas dedicadas (construidas para un lenguaje o una aplicación concreta), por ahora mencionemos sólo que cada vez son más los ordenadores comerciales que abandonan las arquitecturas tradicionales y optan por una estructura interna especialmente adaptada para el uso que se va a hacer de ellos.

Como ejemplo de la simulación mediante software de niveles hardware podemos citar las máquinas UNIX que ejecutan el sistema operativo DOS como una subtarea, lo que permite trabajar con todas las aplicaciones diseñadas para este sistema operativo en máquinas en las que, en principio, no se podrían utilizar. Como ejercicio piense el lector cómo y qué funciones deberían simularse para que el funcionamiento fuera idéntico al de una máquina DOS (como pistas mencionaremos la entrada/salida, dispositivos de interacción, y la velocidad de ejecución).

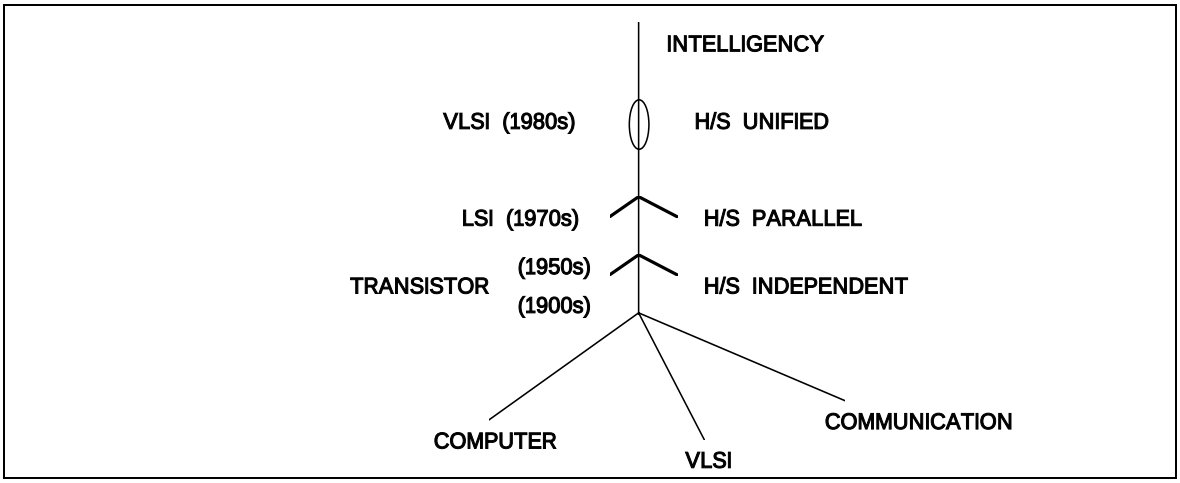


Fig.2 Integración del Software y el Hardware [Matsumura, 1983]

Matsumura recoge esta misma idea de integración entre el hardware y el software y la resume en la figura 2.

Antes de acabar este apartado, conviene mencionar un hecho que, aunque no está directamente relacionado con la integración hardware y software, es muy importante para entender la evolución de los ordenadores. Gran parte de las investigaciones sobre nuevos materiales y dispositivos electrónicos dependen de la capacidad de computación disponible, es decir, cuando se dispone de tecnología para construir un ordenador más rápido, más potente y con más capacidad que los anteriores, se puede utilizar para perfeccionar las técnicas que permitieron su construcción, al mejorar éstas se puede desarrollar otro modelo más potente y así sucesivamente. Este bucle es responsable, al menos en parte, de los grandes avances que se han dado en las tecnologías de la información, se puede decir que se trata de construir una herramienta que nos permita construir una herramienta mejor. Ejemplo palpable de ello es la tecnología VLSI que permite diseñar ordenadores mucho más potentes y que, paradójicamente, es el resultado de poder utilizar computadores para gestionar la complejidad de los diseños de circuitos integrados.

Para muchos, el motor de los avances en las tecnologías de la información es el hardware.

2.3 VLSI

Hablar sobre hardware puede convertirse en una tarea muy ardua si no delimitamos el alcance de la discusión previamente. Aquí nos limitaremos a las arquitecturas de ordenadores, pero antes de entrar a estudiarlas, es conveniente detenerse un momento en los niveles más bajos del hardware, concretamente en el de circuitos.

Existe un consenso casi absoluto a la hora de atribuir el enorme avance de los ordenadores a la creciente capacidad de integrar circuitos. Que no consiste sólo en hacerlos cada vez más pequeños, sino también más rápidos, eficientes, baratos y fiables. Sin esta capacidad de integración, el auge de los pc's hubiera sido impensable (por precio, tamaño y prestaciones) pero tampoco hubieran podido plantearse los ordenadores paralelos (utilizar varios procesadores, a veces en cantidades masivas), el hardware tolerante a fallos (que funciona fundamentalmente por duplicación), los procesadores dedicados (para comunicaciones, para servidores de red), el proceso de señal (con arquitecturas dedicadas gracias a las cuales son posibles muchas aplicaciones de comunicaciones y tratamiento de imágenes), etc.

El poder integrar cada vez más circuitos en un espacio más pequeño no es un capricho. En primer lugar, la mayor rapidez permite construir ordenadores más eficientes y con unas prestaciones elevadas. El poder incluir muchas funciones en un solo chip se traduce en mayor fiabilidad en el proceso de información (un fallo dentro del chip es menos probable que si se divide en varias partes y se comunican a través de un bus) y, lo que es más importante desde nuestro punto de vista, reduce la complejidad a la que se enfrentan los niveles superiores al permitir que parte de esa complejidad se traslade a los circuitos.

Los primeros ordenadores eran máquinas enormes que ocupaban salas enteras y exigían un control absoluto de las condiciones de trabajo (temperatura, humedad, vibraciones, etc.), de ahí hasta los

ordenadores portátiles que conocemos hoy en día se ha recorrido un largo camino. En aquellos ordenadores primitivos no existían muchos de los niveles que hoy consideramos "casi" imprescindibles, el software era prácticamente inexistente tal y como lo entendemos hoy en día, y el hardware era muy simple comparado incluso con el del un ordenador personal actual.

El sucesivo desarrollo de los niveles de la jerarquía se hizo en función de la capacidad para integrar circuitos electrónicos en un espacio cada vez menor. Al mismo tiempo, había que reducir el consumo de potencia, la disipación de calor, flexibilizar las condiciones de trabajo y aumentar la velocidad de computación. Gran parte de los avances que se han producido en las tecnologías de la información se deben a avances previos en el nivel más bajo del hardware, el de circuitos.

Escalas de integración

ULSI es, por ahora, el último escalón de una serie de teconologías de integración de circuitos. Anteriormente, los ordenadores se basaban en tecnología SSI (pequeña escala de integración) o MSI (media escala de integración), con las que se llegaba a integrar hasta 10 puertas lógicas. Posteriormente se alcanzó la LSI (gran escala de integración) con la que se pueden integrar hasta 10.000 puertas lógicas y VLSI que permite integrar cientos de miles de puertas en un chip. ULSI integra millones de puertas.

Hoy en día prácticamente existen ordenadores en un solo chip y la integración de la que hablábamos antes no sería posible sin esta tecnología. Gracias a ella se puede pensar en implementar en hardware cualquiera de los niveles de la jerarquía propuesta por Bell y Newell, incluso los niveles superiores, de aplicaciones, algo impensable hace tan sólo unos años. Gracias a ello, el diseñador de ordenadores ha de enfrentarse con una complejidad distinta ya que cuenta con procesadores y circuitos periféricos de gran potencia que resuelven gran parte de los problemas. Por ejemplo, gran parte de los procesadores avanzados de hoy en día llevan incorporada la Unidad Aritmético-Lógica a la CPU, ya no hay que diseñar la conexión entre ellos, lo mismo sucede con las memorias cache y los gestores y muchos otros elementos.

3. Clasificación de las arquitecturas de ordenadores

Dada la enorme velocidad de desarrollo de los ordenadores es difícil tener una imagen de conjunto de las cosas que existen, las que pueden existir, qué se investiga y qué aproximaciones han quedado superadas. El bombardeo de información, comercial e investigador, al que está sometido cualquiera que trabaje en este campo es suficiente para despistar al más experto. Ordenadores paralelos, procesadores en array, conexionismo, monoprocesadores, sistemas distribuidos, procesadores dedicados, ordenadores vectoriales, micro y macroparalelismo, máquinas de flujo de datos, quinta generación, ordenadores LISP, RISC y CISC, etc., son términos que aparecen frecuentemente no sólo en las publicaciones especializadas sino en cualquier folleto publicitario, los entienda el futuro usuario o no.

Semejante diversidad puede parecer sorprendente si se tiene en cuenta que todas esas arquitecturas se derivan de unos pocos modelos establecidos en los años 40. Pero la variedad es tal que se necesita una clasificación para poder moverse con un mínimo de seguridad en este terreno [Dasgupta, 1990, p. 65].

Por otro lado, las clasificaciones son muy útiles para poder establecer posibles líneas de investigación y evaluar el estado del arte, cada vez más complejo. Existen muchas clasificaciones de las arquitecturas de ordenadores pues los parámetros a seguir son muchísimos, y se necesitan perspectivas muy amplias para que la clasificación no se quede anticuada en muy poco tiempo. Esta característica, la capacidad de predicción, es fundamental en toda taxonomía. Pero la velocidad de desarrollo de las tecnologías de la información en conjunto hace realmente difícil predecir el futuro, por eso muchas de las clasificaciones se hacen intentando extender otras ya existentes y más antiguas, a las que el tiempo ha restado interés y resolución.

3.1 Datos vs. Control, la clasificación tradicional

Una de las clasificaciones más utilizada es la basada en la relación entre datos y control dentro del procesador. Esta clasificación se debe M.J. Flynn y se puede encontrar en cualquier libro sobre ordenadores. A pesar de su simplicidad, es casi la única de uso general.

El criterio de calsificación es el número de instrucciones que se procesan simultáneamente y sobre cuántos datos. De acuerdo con este criterio, se distinguen cuatro grandes grupos de arquitecturas:

SISD (Single Instruction, Single Data), Una Instrucción, Un Dato. Las instrucciones se ejecutan una a una y procesan un único dato cada vez. Ésta es la arquitectura clásica (Von Neumann).

SIMD (Single Instruction, Multiple Data) Una Instrucción, Datos Múltiples. Cada instrucción opera con varios datos al ejecutarse. Esta es la arquitectura de los procesadores en array y de muchas máquinas orientadas al proceso numérico (como las de procesamiento de señales o de imágenes), donde las operaciones a realizar con los datos son relativamente pocas pero hay que efectuarlas sobre muchos datos.

MISD (Multiple Instructions, Single Data) Múltiples Instrucciones, un Único Dato. Supuestamente se ejecutan muchas instrucciones que operan con un único dato. Esta arquitectura no tiene sentido fuera del puramente formal.

MIMD (Multiple Instructions, Multiple Data) Múltiples Instrucciones, Datos Múltiples. Ejecución simultánea de varias instrucciones que operan con varios datos. Esta es la arquitectura de los ordenadores paralelos.

Como es fácil ver, la clasificación es muy simple y permite pocas distinciones entre los ordenadores. Cuando se formuló (1972), todavía no existían ordenadores paralelos (al menos no a gran escala y con paralelismo masivo) y muy pocos procesadores dedicados que son generalmente SIMD). Hoy en día esta clasificación no nos permite distinguir entre todas las clases de ordenadores

paralelos que existen e incluso es muy poco útil para las arquitecturas convencionales que han evolucionado lo suficiente como para que se pueda pensar en distinguir diversas categorías.

3.2 Otros tipos de taxonomía

Como hemos puesto de relieve a lo largo de todos los capítulos anteriores, una estructura muy común que aparece al tratar con sistemas complejos es la jerarquía. Gran parte de las clasificaciones, de arquitecturas de ordenadores o de cualquier otro tipo, se caracterizan por su naturaleza eminentemente jerárquica y se distinguen entre sí, las que operan sobre la misma clase de sistemas, por los niveles que consideren y los que detallan con mayor precisión.

Con los ordenadores sucede exactamente lo mismo. Partiendo de la clasificación de Flynn, que considera dos niveles, datos y control, se pueden detallar aún más esos niveles o se pueden tratar de identificar otros, superiores o inferiores, con lo que se llega a otra clasificación de los mismos sistemas. De hecho, mucha de las propuestas existentes parten del esquema de Flynn. Pero no es la única posibilidad, las clasificaciones tiene un objetivo muy claro, saber qué se ha conseguido y descubrir por dónde se puede avanzar de acuerdo con algun criterio que se convierte en el eje de la clasificación. Se puede pensar en clasificaciones que atienden al número de usuarios, a velocidad de proceso, al número de procesadores que existen, a prestaciones concretas, etc.

Skillicorn [1988, p. 47] recoge varias de estas clasificaciones que resumimos brevemente. T.Y. Feng, en 1972, propuso una clasificación orientada a las prestaciones en cuanto al paralelismo que permite la máquina. El paralelismo se evalúa en función del número de bits que se pueden procesar simultáneamente y se representa como un par de números, el primero la longitud de la palabra con que se trabaja y el segundo el número de palabras que se pueden procesar simultáneamente. Evidentemente, este esquema es muy útil para comparar muy diversas arquitecturas pero no permite resaltar las diferencias ni las similitudes ya que no se hace referencia al tamaño, número de procesadores, precio o aplicación.

Reddi y Feurstel, en 1976, propusieron una clasificación más descriptiva en la que las arquitecturas se clasifican de acuerdo con su organización física, el flujo de información y cómo se representa y transforma ésta. Siendo más genérica que la de Feng, estos parámetros están demasiado orientados a la implementación específica del ordenador pues muchas máquinas resultan ser prácticamente iguales con diferentes organizaciones físicas (por ejemplo, microprogramación vs. cableado). Händler, en un trabajo presentado en 1977, describe las arquitecturas dando el número de procesadores y cómo se interconectan, así como el tamaño de palabra y la profundidad de las unidades aritmético-lógicas. Esta última clasificación está orientada a las máquinas de proceso vectorial y es difícil de generalizar a arquitecturas multiprocesador.

Otras dos clasificaciones más, que detallamos a continuación, son las propuestas por Skillicorn [1988] y Dasgupta [1990], que son un ejemplo muy claro de refinamientos sucesivos de clasificaciones previas y de propuestas de notación, cuando menos curiosas. Skillicorn parte del esquema de Flynn, y Dasgupta desarrolla aún más la propuesta de Skillicorn dotándola de una gran potencia denotacional.

3.3 Una ampliación al trabajo de Flynn

En esta clasificación se opta por una aproximación al problema de la computación y las máquinas a través de varios niveles de abstracción. En el nivel más alto estaría el modelo de computación. Uno de los más conocidos es el de Von Neumann, pero existen muchos más como el de fujo de datos, el de reducción de grafos, los modelos de pilas, paralelos, etc. Un nivel con un grado mayor de refinamiento sería el de máquinas abstractas (sobre las que se implementa el modelo computacional) ya que cada modelo computacional se puede implementar sobre máquinas muy diferentes, aunque se ajuste mejor a determinadas estructuras. Este es el nivel más alto de la clasificación de Skillicorn.

El siguiente nivel sería el de la implementación, la máquina tal y como la ve el programador en lenguaje ensamblador (ésta es una de las definiciones clásicas de arquitectura de ordenadores). Este es el segundo nivel de la clasificación de Skillicorn. Como el mismo reconoce, hay parámetros importantes que deja fuera de la clasificación, uno de ellos es el lenguaje de programación ya que hay "una creciente tendencia a construir conjuntamente el lenguaje y la máquina" (vease qué decíamos anteriormente sobre la integración hardware/software).

Los parámetros que describen el primer nivel de la clasificación son los siguientes:

* El número de procesadores de instrucciones (IP),
* el número de memorias de instrucciones (IM),
* el tipo de conmutadores que conectan las memorias de instrucciones y los procesadores de instrucciones,
* el número de procesadores de datos (DP),
* el número de memorias de datos (DM),
* el tipo de conmutadores que conectan los procesadores de datos y las memorias de datos,
* el tipo de conmutador que conecta los procesadores de instrucciones (IP) y los procesadores de datos (DP),
* el tipo de conmutador que conecta los procesadores de datos (DP) entre sí.

Este primer nivel extiende las clases de Flynn considerando algunas características importantes de las clasificaciones actuales, entre ellas, el tipo de conexión entre los diversos procesadores, un parámetro fundamental en la arquitectura y organización interna del ordenador y decisivo a la hora de considerar sus prestaciones. El segundo nivel considera la posibilidad de "pipeline" en los procesadores y el comportamiento de su diagrama de estados.

La siguiente tabla (figura 3) muestra una posible clasificación atendiendo únicamente al primer nivel y suponiendo que el número de memorias coincide con el número de procesadores:

Class	IPs	DPs	IP-DP	IP-IM	DP-DM	DP-DP	Name
1	0	1	none	none	1-1	none	reduct/dataflow uniprocessor
2	0	n	none	none	n-n	none	separate machines
3	0	n	none	none	n-n	n x n	loosely coupled reduct/dataflow
4	0	n	none	none	n x n	none	tighly coupled reduct/dataflow
5	0	n	none	none	n x n	n x n	
6	1	1	1-1	1-1	1-1	none	von Neumann uniprocessor
7	1	n	1-n	1-1	n-n	none	
8	1	n	1-n	1-1	n-n	n x n	Type 1 array processor
9	1	n	1-n	1-1	n x n	none	Type 2 array processor
10	1	n	1-n	1-1	n x n	n x n	
11	n	1	1-n	n-n	1-1	none	
12	n	1	1-n	n x n	1-1	none	
13	n	n	n-n	n-n	n-n	none	separate von Neumann uniprocessors
14	n	n	n-n	n-n	n-n	n x n	loosely coupled von Neumann
15	n	n	n-n	n-n	n x n	none	tighly coupled von Neumann
16	n	n	n-n	n-n	n x n	n x n	
17	n	n	n-n	n x n	n-n	none	
18	n	n	n-n	n x n	n-n	n x n	
19	n	n	n-n	n x n	n x n	none	Denelcor Heterogeneous Element Processor
20	n	n	n-n	n x n	n x n	n x n	
21	n	n	n x n	n-n	n-n	none	
22	n	n	n x n	n-n	n-n	n x n	
23	n	n	n x n	n-n	n x n	none	
24	n	n	n x n	n-n	n x n	n x n	
25	n	n	n x n	n x n	n-n	none	
26	n	n	n x n	n x n	n-n	n x n	
27	n	n	n x n	n x n	n x n	none	
28	n	n	n x n	n x n	n x n	n x n	

Fig. 3 Cuadro con la clasificación de arquitecturas de Skillicorn.

3.4 Química, arquitectura y ordenadores

Esta clasificación es una mejora explícita de la propuesta de Skillicorn que intenta superar una serie de limitaciones que considera importantes. La primera de ellas es la falta de poder de predicción ya que no permite comparar arquitecturas que pertenecen a clases diferentes o cómo varían diferentes modelos de la misma línea arquitectónica. A pesar de que Skillicorn hace referencia explícita a la necesidad de un modelo jerárquico, Dasgupta sostiene que no lo es y critica que un concepto tan importante como el "pipeline" aparezca en un nivel diferente a las memorias y los procesadores.

El artículo de Dasgupta comienza haciendo un estudio de la ciencia de la taxonomía (clasificación) y enumera las propiedades que debe reunir una buena clasificación. De acuerdo con éstas, considera que la propuesta de Skillicorn carece de poder de explicación (aunque está latente), que no es un esquema jerárquico y que se olvida de aspectos importantes como el ya comentado o la diferencia entre memorias principales y memorias cache.

Para resolver estos problemas, Dasgupta propone una jerarquía basada en los modelos utilizados en la Química, utilizando profusamente términos como radicales atómicos, moléculas, radicales complejos, etc. Las entidades básicas, o átomos, son las siguientes:

* (iM) "interleaved Memory", memorias en las que se puede acceder a varias unidades básicas en un único ciclo de memoria,
* (sM) "single Memory", memorias simples, sean de datos o de instrucciones,
* (C) "cache", memorias "buffer" de acceso rápido,
* (sI) unidad simple (de un sólo paso o de una sola instrucción) de preparación de las instrucciones,
* (pI) unidad con "pipeline" de preparación de instrucciones,
* (sX) unidad simple (de una instrucción o de un solo paso) de ejecución de instrucciones,
* (pX) unidad con "pipeline" de ejecución de instrucciones.

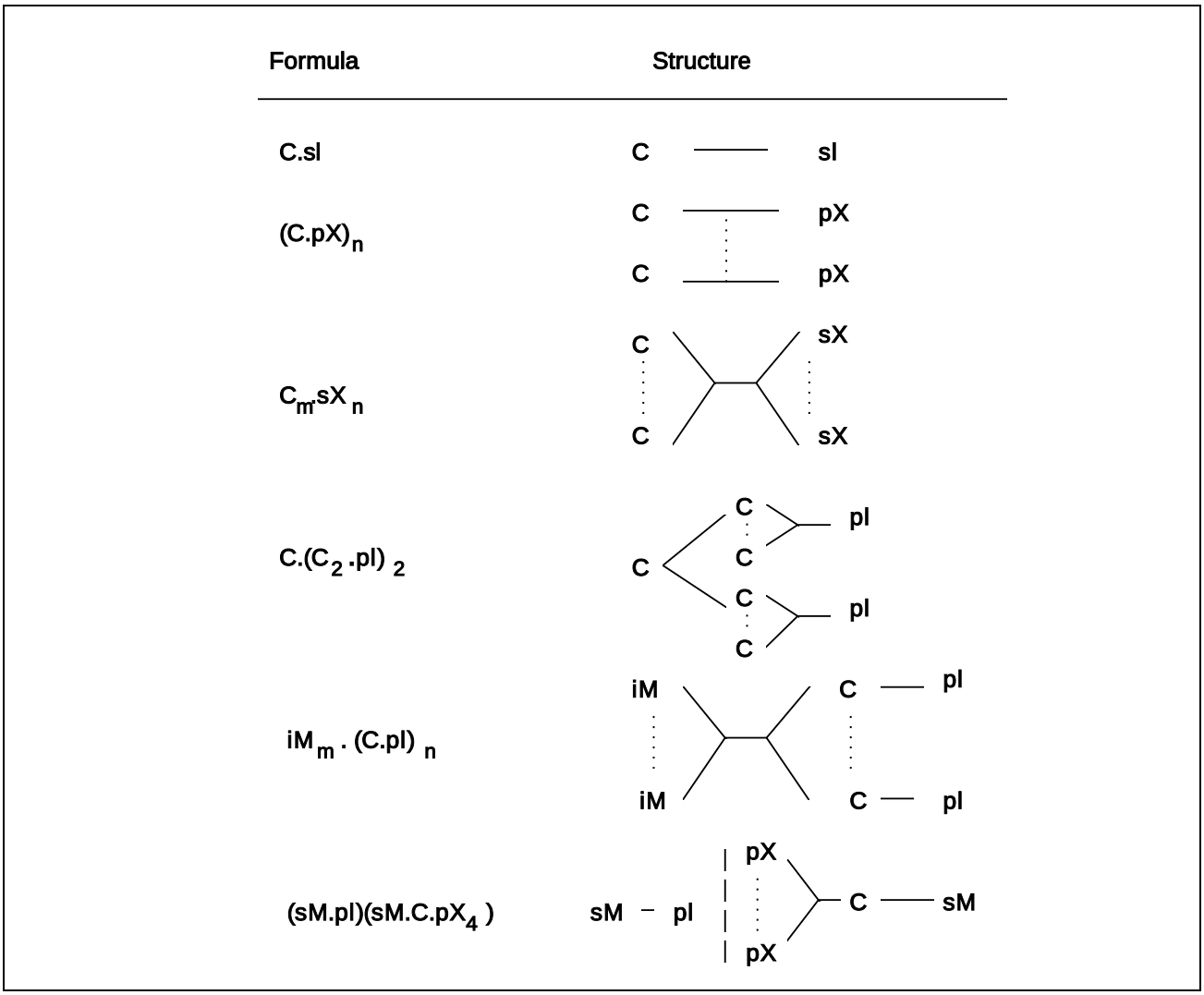


Fig.4 Ejemplos de "formulación" de arquitecturas

Si una máquina tiene varios átomos, se denota con un subíndice añadido a los átomos (por ejemplo, iM_3 o pX_4). También se pueden construir radicales compuestos de varios átomos, por ejemplo: $(iM)_m.(C.pI)_n$, que significa m memorias del tipo iM, asociadas a n procesadores de instrucciones con "pipeline" y dotados, cada uno de ellos, de memoria cache. Una molécula representa un subsistema completo dentro de un ordenador dado, puede ser una molécula X (de ejecución) o I (de preparación de las instrucciones). Una macromolécula combina una molécula I y una molécula X y

representa un ordenador completo. Sobre esta estructura se definen una serie de reglas sintácticas para combinar todos los elementos y formular las expresiones adecuadas a cada caso. En la figura 4 se recogen algunos ejemplos de esta clasificación.

Esta clasificación de Dasgupta es un buen ejemplo de cómo aprovechar notaciones ya aceptadas en otros campos para desarrolar una propia y cómo pueden utilizarse las analogías para facilitar la comprensión y el manejo intuitivo de la taxonomía. Sin embargo, como el propio Dasgupta reconoce, tanto esta clasificación como la de Skillicorn están seriemente limitadas cuando se trata de arquitecturas que no siguen el modelo Von Neumann.

4. Ordenadores Personales y Estaciones de Trabajo

Estudiar las diferentes arquitecturas de ordenador es una tarea muy complicada dada la gran variedad de posibilidades existentes. Pero hay algo aún más difícil, clasificar los ordenadores que están disponibles comercialmente, no según el número de procesadores o de memorias, sino siguiendo la denominación tradicional de Ordenador Personal (pc), Microordenador, Estación de Trabajo (Workstation), Miniordenador, Superordenador, etc.

¿Qué significa cada una de estas categorías?, evidentemente nadie lo sabe con exactitud. En primer lugar, las definiciones de cada una de ellas abundan casi tanto como los productos comerciales y, en segundo lugar, la falta de rigor es casi absoluta. Al observar el panorama existente en las publicaciones especializadas cabe preguntarse si la borrosidad que todo el mundo parece admitir en este sentido no es más que un intento fallido de reducir por las bravas la complejidad de descripción de la oferta informática.

Aquí nos vamos a centrar en lo que parecen ser los dos escalones más bajos, los pc's y las Estaciones de Trabajo (WS, a partir de ahora). Con ellos se puede seguir mejor que con ningún otro el vertiginoso desarrollo de los ordenadores, los cambios que en ellos se han producido y las consecuencias que tuvieron y que tienen a la hora de concebir la informática. Por otro lado, son la referencia más inmediata que tienen la gran mayoría de los usuarios, alejados de los grandes ordenadores y las máquinas especializadas en aplicaciones científicas muy concretas.

4.1 Su majestad el pc

Resumir, aún concisa y brevemente, la historia del Ordenador Personal sería tan sólo repetir lo que otros muchos han hecho en multitud de libros y artículos. Y a pesar de que también existen análisis de todos los tipos, vamos a detenernos brevemente con algunas conclusiones que se pueden sacar de esa historia.

Definir un Ordenador Personal es muy arriesgado, entre otras cosas porque es muy probable que la definición resulte superada al cabo de muy poco tiempo. Sin embargo, parece que existe una serie de características que sin llegar a constituir una definición clara permiten diferenciarlo de alguna manera de otro tipo de ordenadores. Dada la creciente y fuerte tendencia a la normalización, un pc generalmente trabaja bajo un sistema operativo interactivo (el más extendido es el DOS, de

Microsoft), es monousuario, con una memoria que oscila entre los 250 Kb y 4 Mb, con una o dos unidades de disco flexible, y posiblemente con un disco duro de varias decenas de Mb. Es posible conectarlos en red, disponen de varias ranuras de ampliación y el tipo de monitores suele ser de 80x25 columnas con diversas resoluciones en modo gráfico, siempre limitadas por el tamaño del monitor.

Los ordenadores personales y la complejidad

La principal ventaja que ofrecen los ordenadores personales es que ponen al alcance de cualquiera una potencia de cálculo impensable no hace mucho tiempo. La ciencia de hace un siglo se encontraba fuertemente limitada por la falta de instrumentos de cálculo y sólo podía tratar con problemas de una complejidad abordable que, para los estándares de la época, no era muy elevada (recordar el capítulo dedicado a la simplificación y el ejemplo del estudio del sistema solar).

Evidentemente, el H y el O eran exactamente el mismo ahora que hace un siglo pero el I ha cambiado radicalmente y, en consecuencia, también ha cambiado mucho la Imagen del Objeto que nos hemos formado. Al aparecer los primeros ordenadores se pudo empezar a tratar con problemas de una complejidad mucho mayor pero esta posibilidad sólo estaba al alcance de unos pocos privilegiados. Poco a poco fue haciéndose más fácil acceder a los ordenadores hasta que empezaron a aparecer los primeros Ordenadores Personales que brindaron esa posibilidad a todo el mundo. Es decir, permitieron que cambiara la IO de mucha más gente.

La lista de descubrimientos científicos de mayor o menor importancia que se han hecho utilizando un ordenador personal es enorme ya que permite que mucha más gente investigue y estudie un determinado objeto, con lo que las posibilidades de profundizar en el conocimiento sobre ese objeto son mucho mayores. Antes no todo el mundo disponía del tiempo o de la capacidad necesaria para resolver, por ejemplo, un conjunto complicado de ecuaciones, ahora éste es un trabajo que realiza el ordenador y que además lo realiza de forma más precisa y permitiendo ampliar detalles que antes no se podían considerar (de nuevo nos remitimos al ejemplo del sistema solar que estudiamos en el capítulo dedicado al tratamiento de la complejidad). Desde luego siempre están los superordenadores como las herramientas dedicadas a tratar con problemas de una complejidad enorme, pero los Ordenadores Personales, aún disponiendo de una capacidad incomparablemente menor, permiten que cada usuario individual maneje una complejidad considerable con un coste muy reducido.

En este apartado dedicado a los ordenadores personales vamos a estudiarlos desde el punto de vista de su evolución en el tiempo y la creciente expansión en el mercado tanto de los ordenadores personales como de las estaciones de trabajo. Lo que aquí digamos ha de interpretarse de acuerdo con la idea de que los ordenadores son básicamente herramientas para tratar la complejidad y que los Ordenadores Personales son herramientas que nos permiten tratar la complejidad de muchos objetos cotidianos.

Evidentemente todas estas características son discutibles y lo serán cada vez más a medida que pase el tiempo, pero aún así pueden dar una idea bastante clara de lo que es un pc.

La gran ventaja que ofrecen es poner a disposición del usuario no especializado una enorme capacidad de proceso unida a una flexibilidad que no tiene prácticamente ningún otro tipo de ordenador. Si cuando aparecieron ya se consideraron como un hecho "revolucionario" en la forma

de trabajar, esa revolución continúa pues los avances han permitido multiplicar por un factor muy elevado todas las prestaciones que ofrecen.

El ordenador es una herramienta que permite manejar mucha complejidad. Esto es algo que hemos venido repitiendo con frecuencia a lo largo de estos apuntes. El ordenador personal es una herramienta accesible a mucha gente a la que le permite gestionar de forma individual problemas cada vez más complejos. Los primitivos ordenadores personales con 64 Kb de memoria y procesadores de 8 bits permitieron que muchos científicos, por citar algún ejemplo concreto, dispusieran de una herramienta personal (y este adjetivo es lo importante) para abordar los problemas que estudiaban. Muchos de los estudios realizados en la segunda mitad de este siglo son inseparables de la historia del ordenador y no se podrían haber abordado sin éstos. Un ejemplo muy claro es la dinámica de sistemas cuyo potencial sólo es aprovechable si se dispone de un ordenador capaz de ejecutar rápidamente todas las simulaciones que sean necesarias. Y lo mismo sucede con la predicción del tiempo atmosférico, el estudio del caos, los fractales, estudios aerodinámicos, controles de centrales de energía y un largo etcétera.

Pero la mayoría de esas aplicaciones requerían y requieren grandes ordenadores manejados por especialistas, la importancia del pc es que puso una parte de ese poder de computación en manos de los usuarios. Hoy en día existen ordenadores que aún estando dentro de la categoría de los pc permiten realizar complicadas tareas de diseño y análisis que no hace mucho eran competencia exclusiva de los grandes ordenadores. Pero aún dejando de lado esas aplicaciones especializadas, los pc's permiten que cualquiera, cómodamente instalado en su casa, realice complejos análisis financieros, prepare documentos de gran calidad, lleve la contabilidad de pequeñas e incluso medianas empresas, utilice sistemas expertos, programas de análisis de circuitos y un sinfín de tareas que antes eran inabordables.

Además, la enorme presión comercial en este mercado hace que los precios bajen cada vez más y las máquinas sean cada vez más potentes. Cualquiera familiarizado con el mundo de los microprocesadores conoce la saga de los 8086, 8088, 80186, 80286, 80386 y 80486, todos ellos aportando verdaderos saltos cualitativos en prestaciones y concepción de los ordenadores personales. Modelos como el 8 Mb Mac II de Apple, PS/2 Modelo 80 de IBM o el COMPAQ Deskpro 386/40 tienen memorias de alrededor de 10 Mb y procesadores que funcionan a 16,7 MHz con capacidad multitarea, a un precio que en ningún caso llega a los 10.000 $ (y algunos son bastante más baratos).

Por otro lado, la enorme flexibilidad con que están diseñados permiten convertir a un ordenador personal en un auténtico ordenador especializado que pone a disposición de los usuarios un potente centro de cálculo. Personalizar un pc para una aplicación concreta es probablemente más barato que comprar un ordenador grande capaz de abordar ese mismo problema directamente. Existen tarjetas de ampliación con procesadores especializados en proceso numérico, procesado de imágenes, procesado de voz, arrays sistólicos, tarjetas de adquisición de datos, periféricos para programar PLD's y memorias, posibilidades de conectarse a todo tipo de redes, convertir un pc en un banco de pruebas de circuitos integrados, controladores de equipos con pantallas táctiles y multitud de periféricos que permiten hacer de un ordenador personal una potente máquina especializada.

4.2 Estaciones de trabajo

Como su propio nombre indica, las estaciones de trabajo fueron inicialmente concebidas para aplicaciones muy especializadas que requerían gran potencia de cálculo y capacidad de memoria. El desarrollo de este tipo de máquinas es, de alguna manera, paralelo al de los ordenadores personales pero aplicado a la ciencia y a la ingeniería.

Definir una WS es tan complicado como definir un ordenador personal, normalmente se entiende como tal una máquina monousuario, con capacidad de multitarea y, como mínimo, prestaciones de 1 MIPS (Mega Instructions Per Second), conectable mediante red Ethernet, sistema Operativo UNIX, aplicaciones técnicas y de ingeniería y monitor de alta resolución.

Aplicaciones típicas de las estaciones de trabajo son programas de CAE/CAD (Diseño Asistido por Ordenador) y la mayor parte de los usuarios de estos ordenadores son técnicos y especialistas en diversas ramas de la ciencia y la ingeniería. Son, en general, problemas que podrían resolverse con un pc pero a costa de pagar un alto precio en eficiencia y velocidad y con limitaciones importantes (sobre todo en procesamiento numérico y representación gráfica), de hecho muchas de las aplicaciones existentes en el mercado de WS también existen en el mercado de los ordenadores personales aunque en versión reducida. Ya hemos mencionado el diseño asistido por ordenador (que necesita una alta resolución gráfica y unas prestaciones numéricas importantes para procesar las imágenes) pero también conviene mencionar el diseño industrial (de componentes mecánicos y electrónicos y las correspondientes pruebas y análisis), modelado de elementos finitos, animación de vídeo, análisis térmicos y electromagnéticos, estudios de dinámica de fluidos, programación de grandes sistemas, fabricación asistida, etcétera.

Una idea aproximada de la potencia de las actuales estaciones de trabajo es la siguiente: "con los sistemas actuales, los ingenieros pueden diseñar, por ejemplo, una pieza del ala de un avión, con la próxima generación de ordenadores más potentes podrán diseñar el ala entera y con futuras generaciones serán capaces de diseñar el avión completo" [LoPiccolo, 1988]

Sin embargo existe un problema que ya hemos comentado en otros apartados de este capítulo. El hardware parece ir siempre muy por delante del software y, aunque la tecnología y los ordenadores cada vez son más potentes, no se dispone de las herramientas necesarias para sacarle partido. Y hay muchos factores que contribuyen a aumentar aún más ese distanciamiento entre el hardware y el software, las máquinas RISC (ver el apartado dedicado a este tema en este mismo capítulo) han permitido aumentar de forma importante la potencia y prestaciones de las estaciones de trabajo, lo mismo sucede con el paralelismo (especialmente las arquitecturas como los *arrays* sistólicos) que permite realizar simultáneamente muchas operaciones que antes se tenían que hacer de forma secuencial. La tecnología de las estaciones de trabajo es tan fluída que es muy difícil adaptar el software existente a las nuevas máquinas que van saliendo al mercado y más si estas aparecen a la velocidad con que lo hacen.

Al contrario de lo que sucede en los pc's, donde las prestaciones son importantes pero lo es aún más disponer de un programa para aprovechar la flexibilidad que ofrecen, en las estaciones de trabajo el parámetro fundamental es la velocidad y la capacidad, siempre se necesita más potencia de cálculo y mayor resolución gráfica. A estas demandas responden la industria y la investigación hardware

pero la velocidad de desarrollo hace muy difícil disponer del sofware adecuado y más si tenemos en cuenta que las estaciones de trabajo suelen utilizarse de forma muy especializada (para una o dos aplicaciones fundamentales).

Quién es quién en el mercado de las estaciones de trabajo

Hay cinco grandes compañías que compiten por el mercado de las estaciones de trabajo (4 si tenemos en cuenta que Hewlett-Packard compró Apollo). Apollo Computer Inc., ofrecía tres familias de estaciones: 3000, 4000 y 5XX, estas dos últimas de altas prestaciones. Sun Mycrosystems Inc. (el nombre Sun significa Stanford University) ofrece una gama muy amplia de estaciones de trabajo con unas prestaciones muy interesantes gracias a sistemas muy novedosos como la estación Sun 4/200, con arquitectura RISC o el sistema distribuido de ficheros NFS, que permite trabajar con unidades sin disco al compartirlos a través de la red. Hewlett-Packard, conocida como la compañía de ingenieros para los ingenieros, presenta una línea de estaciones en las que la arquitectura RISC también juega un papel muy importante. Digital Equipment Corportaion (DEC) tiene la gran ventaja de ser uno de los líderes mundiales en redes de ordenadores y ofrecer una gran compatibilidad entre máquinas y un parque instalado muy amplio (los conocidos VAX, a los que las estaciones de trabajo, Micro VAX, se pueden conectar de forma sencilla algo que no es cierto en la mayoría de los otros equipos). IBM, el gigante azul, fue la primera compañía que anunció una estación de trabajo con arquitectura RISC, el IBM RT PC, que, sin embargo, no fue un éxito comercial. Los nuevos sistemas PS/2 y el nuevo modelo de estación, 9370, intentan recuperan ese terreno perdido. Pero este panorama está cambiando continuamente, desde luego a mayor velocidad de la que rige el ritmo de una publicación como ésta.

Existen teorías que aseguran que el número de errores que comete un ingeniero al diseñar un dispositivo determinado decrece de una forma muy importante con el aumento de la velocidad del respuesta del ordenador, debido a que no tiene que esperar esta respuesta y no se interrumpen sus procesos mentales [Stenzel, citado por Lopiccolo, 1988]. Evidentemente, si se está realizando un diseño gráfico, representa una gran ventaja ver inmediatamente cuáles son las consecuencias de una acción determinada (esto, dada la complejidad de las tareas involucradas es algo que no se puede conseguir con un pc y quizá sea un tema interesante a estudiar a la hora de diseñar las interfaces de las aplicaciones de los pc's). Lo mismo se puede decir de la resolución gráfica, se puede entender mucho mejor lo que se está haciendo si se ve correctamente y con detalle (algo difícil de conseguir con los pc's).

Algunas medidas de productividad aseguran que la productividad de los ingenieros puede incrementarse en un 30-40 % si utilizan una estación de trabajo. Este dato es muy importante, porque el primer objetivo de las estaciones de trabajo es conseguir importantes beneficios de la inversión, como pueden ser mejoras en el proceso de diseño o ciclos de diseño y fabricación más cortos.

4.3 Ordenadores Personales vs. Estaciones de Trabajo

El principal problema que se le plantea a un usuario es decidir qué es lo que necesita para sus aplicaciones, un ordenador personal o una estación de trabajo. Realmente no es fácil de decir aunque los productores de este tipo de equipos parecen tener muy clara la diferencia entre ambos mercados. No cabe ninguna duda de que los avances en la tecnología de los pc's hacen que éstos sean cada vez más parecidos a las estaciones de trabajo. Los microprocesadores utilizados en modelos de ordenador personal comerciales y ampliamente extendidos ya permiten multitarea, manejan varios megas de memoria y utilizan dispositivos gráficos de alta resolución, las prestaciones son buenas y se pueden mejorar de forma importante si se recurre a la personalización del equipo. También son capaces de funcionar bajo el sistema operativo UNIX intentando aprovechar las aplicaciones escritas para este sistema. Pero las estaciones de trabajo evolucionan a la misma velocidad que los ordenadores personales y sus prestaciones mejoran a gran velocidad. Una forma de interpretar este dilema es considerar que la gama alta (los mejores equipos de los pc's, con mayores prestaciones y más caros) puede llegar a confundirse con la gama baja de las estaciones de trabajo (las no muy especializadas y con prestaciones "normales" dentro de los márgenes que manejan las WS). Algunos llegan a hablar incluso de las pc-Workstations en contraposición a las "Technical Workstations" para diferenciar entre unos y otros.

La figura 5 muestra las prestaciones de tres equipos punteros dentro de la oferta de los ordenadores personales en un momento histórico ya claramente sobrepasado.

Memory Capacity	8	16	10
MB/ board - vendor	1	2	n/a
MB/board - Clearpoint	1	n/a	n/a
Bus Architecture	NuBus	MicroChannel	Modified AT
Expansion Slots (memory)	8	8	n/a
Error Checking	none	Parity	Parity
Processor	16.7 MHz 68020	16.7MHz 80386	16.7 MHz 80386
Math Co-processor	68881	80387(opt)	80387 or Weitek
MIPS	2	1	n/a
Operating System	Apple, A/UX	OS/2(1988)	OS/2 (1988)
***Base System Price**	4300	6995	7999

Fig.5 La gama alta de los ordenadores personales [Engineering Tools, 1988, p. 19]

Si comparamos los precios de estos equipos con los que aparecen el cuadro de las estaciones de trabajo llegamos a una de las conclusiones más importantes a la hora de decidir entre un pc o una WS. Los pc's no admiten utilizar UNIX de forma natural y son más dificiles de conectar en red (un aspecto fundamental en toda instalación) y sus prestaciones no justifican su precio en muchos casos. Aún así, el número de aplicaciones que existen para WS no es comparable a la ingente cantidad de programas escritos para pc, si lo que se desea es flexibilidad y un espectro amplio de utilización la balanza se inclina definitivamente por los pc's (eso sin tener en cuenta la diferencia de precio entre el software para pc y el software para WS).

La figura 6 recoge las prestaciones y precio de algunas estaciones de trabajo agrupadas por compañías y en las diferentes series existentes. Es importante comparar estos datos con los del cuadro anterior para tener una idea más clara de dónde están las diferencias.

Memory Capacity	8	32	32	16	6	32
MB/ board - vendor	1,2	8	8	8	4	8
MB/board - Clearpoint	1,2	8	na	16	16	na
Bus Architecture	PC-AT	PC-AT	VME	Q-Bus	none	Q-Bus
Expansion Slots (memory)	4	4	4	2	none	4
Error Checking	Parity	Parity	EDC	Parity	Parity	EDC
Processor	12 MHz	25 MHz 68020	20 MHz	20 MHz	20 MHz Micro VAX	22 MHz
Math Co-processor		68881		78132		CMOS FPU
MIPS	1.5	4	n/a	9	9	3
Operating System	Domain/IX (UNIX clone)			VMS, Ultrix (UNIX clone)		
***Base System Price ($)**	4990	13,900	46,900	26,000	5400	50,400

Fig.6 Cuadro de prestaciones y precios de algunas estaciones de trabajo

Pero apenas se ha definido una situación, esta cambia con una velocidad sorprendente. Al igual que sucedió con los pc's, las aplicaciones de las WS de carácter más técnico están dando paso a otras mucho más genéricas (existe un estudio en el que se afirma que el 70% de utilización de las estaciones de trabajo se dedica a funciones administrativas). De alguna manera, las estaciones de trabajo están invadiendo el terreno de los pc's. Y como todo lo que pueda ser un mercado es aprovechable, están empezando a aparecer aplicaciones de proceso de texto, hojas electrónicas, publicación electrónica y bases de datos con interfaces mucho más convivenciales que trasladan una tarea clásica del ordenador personal a las más potentes estaciones de trabajo. En la figura 8 se muestran unas expectativas de desarrollo del mercado tanto de las estaciones de trabajo como de los pc's (nótese la agrupación que antes mencionábamos en pc-Workstations y Technical Workstations).

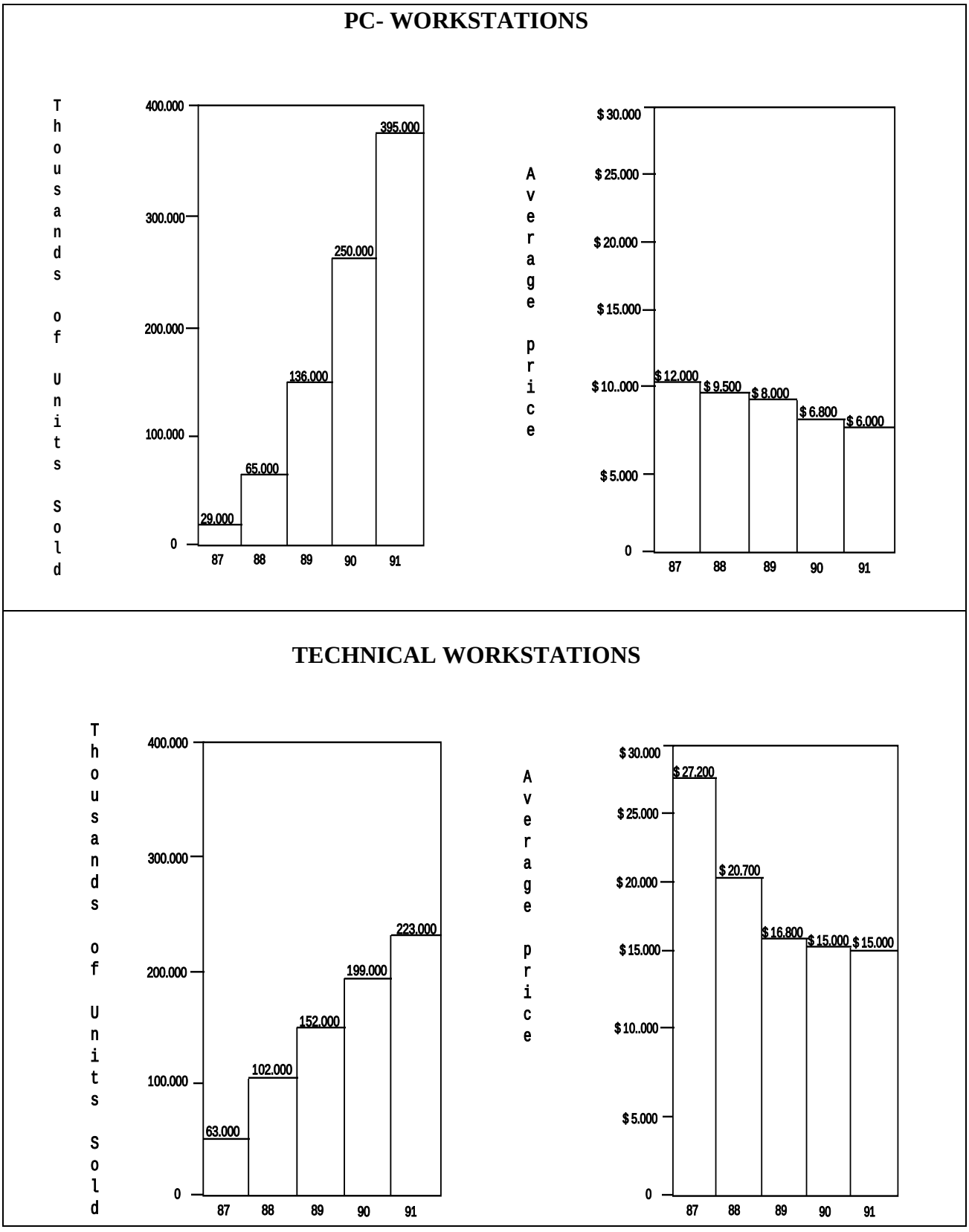


Fig.7 Evolución del mercado de ordenadores personales y estaciones de trabajo [Engineering Tools, 1988, p. 87] 5.

5. RISC, la simplificación del diseño

A partir de la propuesta inicial de Von Neumann y gracias a los avances en electrónica, se fueron construyendo ordenadores cada vez más complejos. Las primeras máquinas no eran más que un conjunto de puertas lógicas con una arquitectura muy simple, que se fue complicando a medida que aumentaba la capacidad de integración de circuitos. Para ganar en velocidad y eficiencia se fueron implementando en hardawre cada vez más niveles de la jerarquía y la potencia de computación del hardware se multiplicó por factores muy elevados.

Esto dió lugar a arquitecturas extremadamente complejas, con un lenguaje máquina de varios cientos de instrucciones que permitían realizar prácticamente cualquier operación imaginable (por ejemplo, el VAX-11 tiene instrucciones máquina directamente orientadas a lenguajes de alto nivel como son INDEX, para utilizar matrices, CASE, para la estructura de alto nivel del mismo nombre o CALL, para llamadas a procedimientos con paso de parámetros). Muchas de las instrucciones de código máquina eran incluso más potentes que algunas instruciones de lenguajes de alto nivel. Como consecuencia, la complejidad organizativa de los procesadores y elementos asociados creció considerablemente. Las razones para construir conjuntos de instrucciones complejos eran dos fundamentalmente [Sáez Vacas, 1987, p. 60]:

- Aproximar los niveles semánticos entre varios lenguajes de alto nivel y una parte de los niveles de lenguaje ensamblador y máquina. Al implementar en hardware instrucciones muy complejas se permite que instrucciones de alto nivel puedan ejecutarse casi sin necesidad de ser traducidas por el compilador. De forma extrema, se puede pensar en un lenguaje máquina que coincide con el lenguaje de alto nivel.

- Mejorar la relación procesador/memoria, al reducirse el número de accesos a memoria. Una instrucción compleja permite operar mucho más sobre un dato determinado sin necesidad de traerlo y llevarlo a memoria.

Esto está en la línea que ya comentábamos al principio del capítulo al hablar sobre integración de hardware y el software.

Sorprendentemente, era ampliamente conocido y estaba suficientemente documentado que los programadores y, en consecuencia, los compiladores (que son los que más frecuentemente acceden al código máquina), utilizan un conjunto bastante reducido de instrucciones que supera el 75 % de frecuencia de utilización de las instrucciones del lenguaje [Sáez Vacas, 1987, p. 59].

Para reducir la complejidad de diseño se inició a finales de los 70 y principios de los 80 una tendencia a diseñar las arquitecturas con un conjunto reducido y simple de instrucciones, comenzando así la singladura de los ordenadores RISC (Reduced Instruction Set Computers).

Tempus y Hora, una metáfora RISC

En el capítulo dedicado al tratamiento de la complejidad estudiabamos la propuesta de simplificación que, en forma de parábola, hace H.A. Simon. El problema que plantean las arquitecturas RISC es muy similar. Los conjuntos complejos de instrucciones consumen muchos ciclos de reloj por instrucción, de forma que cada vez que aparece una interrupción el sistema o salva el estado global (muy complejo, dada la estructura del procesador) o desperdicia mucho trabajo reinicializando la instrucción interrumpida.

La aproximación RISC propone instrucciones mucho más simples (teóricamente de un único ciclo de reloj de duración) de forma que las interrupciones no suponen un gran problema y, cuando se necesitan operaciones complicadas (que como comentabamos antes es menos fecuente de lo que a primera vista pudiera parecer) se pueden construir a partir de las más simples. El esfuerzo se traslada a la compilación, pudiéndose así construir procesadores mucho más simples, eficientes y rápidos.

La filosofía de diseño RISC plantea las siguientes premisas [Gimarc, 1987, p. 59]:

1.- Analizar las aplicaciones para determinar qué operaciones se utilizan con más frecuencia.

2.- Optimizar la ruta de datos para ejecutar esas instrucciones tan rápidamente como sea posible.

3.- Incluir otras instrucciones sólo si se ajustan a la ruta de datos previamente especificada, son relativamente frecuentes o su inclusión no ralentiza la ejecución de las instrucciones más frecuentes.

4.- Aplicar la misma filosofía a todos los recursos del procesador, incluir sólo los más frecuentemente utilizados y sólo si no ralentizan recursos más utilizados.

5.- Trasladar toda la complejidad que sea razonablemente posible del hardware de ejecución al software de compilación.

A partir de ellas, las máquinas RISC ofrecen una serie de características comunes que todas comparten en mayor o menor grado: [Gimarc, 1988, p. 59]:

* Ejecución de la mayoría de las instrucciones en un solo ciclo.
* Conjunto de instrucciones de recuperación y almacenamiento.
* Decodificación cableada de instrucciones.
* Relativamente pocas instrucciones y modos de direccionamiento.
* Formato de instrucciones fijo para facilitar la decodificación.
* La complejidad se traslada a compiladores optimizados.
* Ruta de datos muy segmentada para facilitar concurrencia.
* Conjunto amplio de registros.
* Múltiples niveles de memoria.
* Conjunto de instrucciones diseñado para un grupo determinado de aplicaciones.

La controversia entre ordenadores RISC y CISC puede dividirse en dos categorías: cómo diferenciar un ordenador RISC de uno CISC y cómo hacer comparaciones razonables entre ellos.

MOTOROLA amplía su familia de microprocesadores CISC
(Mundo Electrónico, Abril 1990, p. 51)

"El nuevo miembro de la familia 680X0 de microprocesadores CISC de 32 bits de Motorola dobla las prestaciones del 68030, integra más de 1,2 millones de transistores y consigue una media de ejecución sostenida de 1,3 instrucciones complejas por ciclo de reloj, valor muy próximo al teórico de 1 instrucción por ciclo de los microprocesadores RISC".

Este microprocesador, llamado 68040, integra en un único chip una unidad de cálculo de coma flotante y dos antememorias de 4Kbytes y se prepara en tres versiones, de 25, 33 y 50 MHz. Con técnicas de segmentación, puede realizar hasta 14 tareas simultáneas, la unidad de cálculo de enteros realiza 6 operaciones al tiempo y la unidad de coma flotante 3 operaciones simultáneas. La unidad de cálculo puede trabajar de forma independiente sobre las dos antememorias. Para evitar tener que salvar el estado del procesador cada vez que se produce una interrupción (operación muy compleja dada la cantidad de parámetros involucrados) sólo se almacenan los datos de partida y si se produce alguna interrupción, se reanuda la tarea desde el principio.

Hewlett-Packard anuncia una amplia oferta de productos basados en RISC
(Mundo Electrónico, Mayo 1990, p. 60)

"... con esta presentación de nuevos equipos, la mayor que realiza la compañía en sus 50 años de historia, H-P ofrece ahora la más amplia oferta de odenadores basados en la arquitectura RISC de todo el mercado". Hewlett-Packard es una de las compañías que más firmemente se ha comprometido con la arquitectura RISC con su línea de procesadores Spectrum (HP Precision Architecture) orientada a aplicaciones UNIX.

IBM anuncia la familia RISC System/6000
(Mundo Electrónico, Abril 1990, p. 56)

"IBM España ha presentado recientemente su nueva familia RISC System/6000 ...Estos nuevos equipos, que han supuesto una inversión de 1000 millones de dólares, mejoran la tecnología RISC, al tiempo que incorporan la nueva versión del sistema operativo UNIX de IBM, el AIX V.3".

"Todos estos modelos utilizan la arquitectura POWER (Performance Optimization with Enhanced RISC), con un nuevo procesador super-escalar capaz de ejecutar varias instrucciones en un solo ciclo."

Desde el punto de vista conceptual, los ordenadores RISC son una aproximación simplificadora al diseño de arquitecturas de ordenadores partiendo de una filosofía coherente con lo que se quiere y lo que se necesita. Desde el punto de vista de la complejidad, constituyen un ejemplo muy interesante de reducción de la complejidad de diseño y un punto de partida a considerar al plantearse nuevas aproximaciones a los ordenadores, pero desde luego han de compararse cuidadosamente con arquitecturas tradicionales y no es tarea fácil. Desde el punto de vista tecnológico son una elección cuyas motivaciones y consecuencias cambian a gran velocidad, muchas de las características de los ordenadores RISC provienen de los CISC, como la ruta de datos segmentada o las memorias cache. Y a pesar de que fueron inicialmente pensados para tipos concretos de aplicaciones, hay una clara tendencia a ensanchar el espectro de aplicaciones y hacerlas similares en utilidad a los CISC. Todo ello conduce a una situación de confusión bastante extendida sobre las ventajas de unas y de otras arquitecturas, confusión que aumenta a medida que la tecnología reduce las diferencias que

inicialmente eran más evidentes. Como prueba de ello mostramos el cuadro que recoge diversas noticias referentes a la presentación de diversos equipos informáticos.

6. Paralelismo

La enorme velocidad de evolución de los ordenadores ha permitido abordar problemas cada vez más complejos a través del aumento de la capacidad de computación. Pero aunque esto fuera impensable hace pocos años, se está llegando al límite de esas posibilidades, dada la concepción tradicional de los ordenadores. La arquitectura von Neumann, la más conocida y utilizada, presenta varios problemas de cuellos de botella que hasta ahora se han ido resolviendo mejorando el hardware que la soportaba. Cuando hizo falta más velocidad se construyeron procesadores más rápidos, cuando faltó más capacidad se construyeron procesadores más potentes, para responder a la necesidad de manejar enormes cantidades de datos se perfeccionaron las memorias, su velocidad y su capacidad, pero incluso todas esas mejoras y avances tiene un límite si no se cambia la arquitectura del ordenador.

El límite teórico en arquitecturas clásicas viene impuesto por la velocidad de transmisión de los electrones dentro de los circuitos que componen el ordenador (una fracción de la velocidad de la luz). Una velocidad ciertamente elevada pero insuficiente para muchos problemas que se plantean hoy en día a los ordenadores.

Un ejemplo de este tipo de problemas es la simulación de tiempo atmosférico. Estas simulaciones se hacen dividiendo la zona de la atmósfera a estudiar en una rejilla tridimensional y estudiando la evolución de las condiciones climatológicas en cada una de las divisiones, luego se integran todos los resultados en una solución global. La resolución y la exactitud aumenta disminuyendo el tamaño de las divisiones (ver en el capítulo dedicado a los conceptos relacionados con la complejidad los problemas que plantea el estudio de la atmósfera). Estos cálculos, para que sean útiles en predicción, han de hacerse en tiempo real, es decir, la predicción del tiempo para mañana no puede tardar dos días.

Para solucionar este tipo de problemas se pensó hace ya tiempo en recurrir a arquitecturas paralelas, en las que la idea básica es que si un procesador ejecuta 1000 instrucciones por segundo, 100 procesadores ejecutarán 100.000 instrucciones por segundo. En la práctica esto no es así, como veremos, pero este tipo de arquitecturas no sólo permiten abordar problemas como los mencionados sino que además pueden tratarse de un forma totalmente distinta. El ejemplo del tiempo atmosférico ilustra muy bien este punto. Los cálculos para cada división de la rejilla son independientes entre sí, salvo las condiciones de contorno que se imponen unas a otras y las influencias entre ellas, luego se puede pensar en realizar el análisis de cada una de estas divisiones en un procesador y, utilizando un número suficiente de procesadores, ejecutar todas al mismo tiempo.

6.1 Arquitecturas paralelas

Las diversas categorías de ordenadores paralelos se diferencian entre sí en dos parámetros fundamentales. El primero de ellos es cómo se va a conseguir el paralelismo, es decir, qué

estructuras son las que van a funcionar en paralelo. Una vez decidido este punto hay que precisar el método de interconexión entre los elementos paralelizables. Las comunicaciones son un problema crítico dentro de un ordenador paralelo ya que es necesario coordinar todos los procesos que se ejecutan concurrentemente, el tipo de solución adoptada caracteriza a la máquina tanto o más que la propia arquitectura.

Como se dijo antes, la idea básica del paralelismo es que si un procesador consigue unas prestaciones X, entonces es lógico pensar que n procesadores conseguirán unas prestaciones nX. Esto, por supuesto, en teoría, ya que la coordinación de esos n procesadores supone una sobrecarga muy importante y hay problemas que no se pueden resolver de forma eficiente con un ordenador paralelo. De acuerdo con esto, las interpretaciones que se pueden hacer, y se han hecho, del paralelismo son muy diversas y afectan a todos los niveles de organización del ordenador. Desde las máquinas para proceso numérico con múltiples unidades funcionales a los ordenadores con varios procesadores completos, hay un amplio abanico de posibilidades que prácticamente se ha experimentado por completo, existiendo muchos prototipos y productos comerciales.

Aquí vamos a dar sólo un breve repaso a las diversas arquitecturas básicas y a algunos métodos de conexión, intentando resaltar la complejidad que resuelven y la que generan.

***Arrays* sistólicos.** Se basan en el hecho de que en muchas aplicaciones matemáticas hay un núcleo básico de operaciones que se están utilizando constantemente, por ejemplo, las multiplicaciones de matrices, soluciones de sistemas lineales o transformadas de Fourier. Con un número suficientemente amplio de elementos funcionales (pequeños procesadores o unidades aritmético-lógicas) y las conexiones adecuadas entre ellos se puede conseguir paralelizar una operación matemática compleja reduciendo en mucho los tiempos de cálculo. En la figura 8 aparece un array sistólico para la multiplicación de matrices [A] x [B] = [C]. Los arrays sistólicos ejecutan las operaciones y la Entrada/Salida al mismo tiempo y no necesiatn ningún tipo de control.

Evidentemente, sólo son útiles en determinadas operaciones matemáticas y rentables unicamente en determinadas aplicaciones de proceso numérico y necesitan un ordenador de propósito general que realice el resto de las funciones. La eficiencia conseguida es muy grande pues se diseña el hardware exactamente para el tipo de operación a realizar.

Procesadores en array (vectoriales). El principio en que se basan es practicamente el mismo que el de los arrays sistólicos. Sin embargo, la aplicación es más genérica pues la operación a realizar sobre los datos se envía a todos los procesadores que la ejecutan sobre sus datos locales. Este tipo de arquitecturas se utiliza para problemas muy estructurados que requieren manejar un gran número de datos en forma matricial. Al contrario de lo que sucede con los arrays sistólicos, en los que el programador no necesita saber siquiera que están, la eficiencia de los procesadores en array depende en gran medida de la programación y el diseño de algoritmos adecuados. Existen máquinas con mas de 16.000 elementos de proceso conectados en forma de matriz utilizadas para procesar imágenes de satélites, simulación del tiempo atmosférico, estudios aerodinámicos, procesamiento de imágenes radar, etc. Este tipo de arquitecturas introduce una de las controversias características del paralelismo ya que ofrecen la posibilidad de una velocidad de proceso muy elevada pero con el coste adicional de una mayor complejidad en el diseño de los programas.

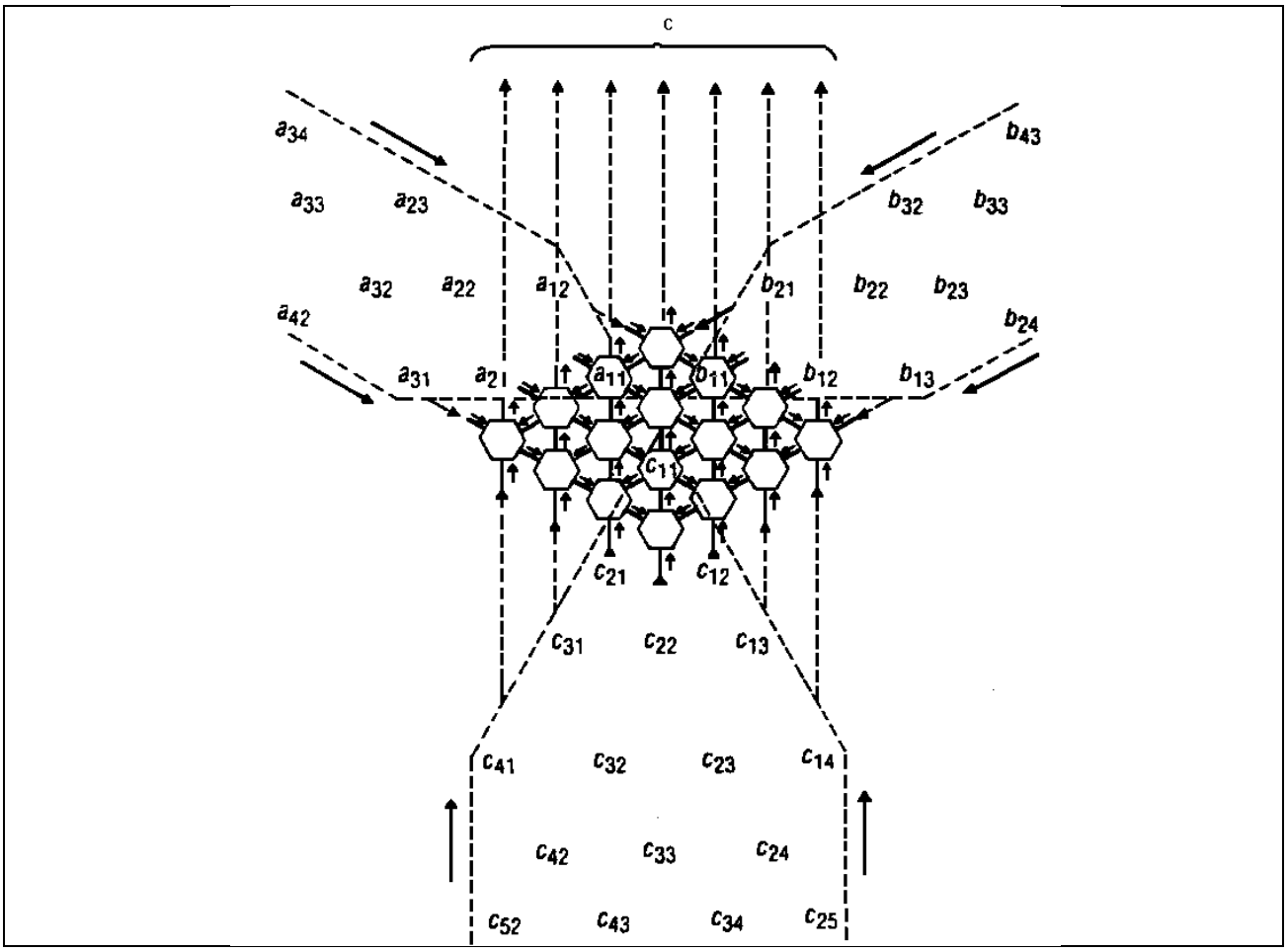


Fig.8 Ejemplo de utilización de un array sistólico

Máquinas de flujo de datos. Este tipo de ordenadores propone una arquitectura radicalmente diferente a la de Von Neumann. En lugar de centrarse en las instrucciones (las instrucciones se ejecutan tras decodificarse y traer los datos correspondientes), en estas máquinas lo relevante son los datos, no existe nada parecido a un contador de programa o a una unidad de control. Para explicar mejor su funcionamiento utilizaremos la siguiente figura:

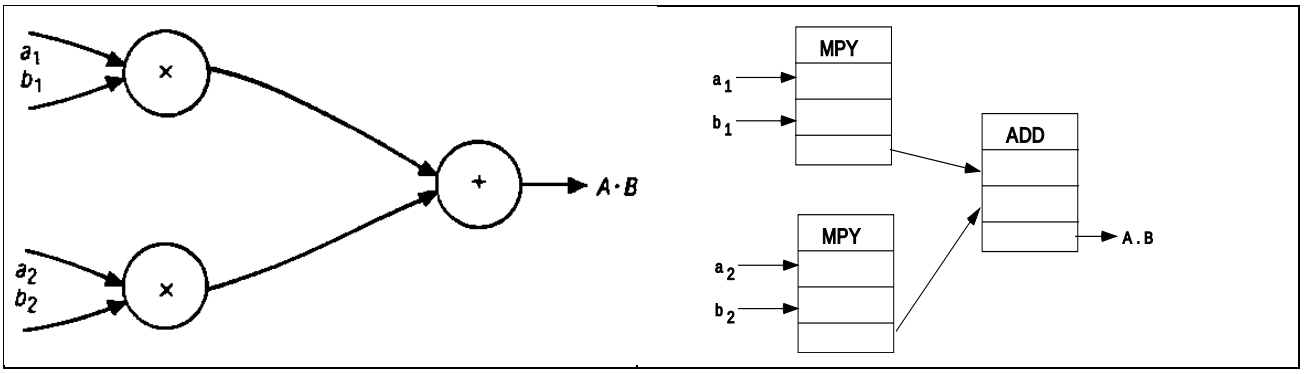


Fig.9 Ejemplo de ejecución en una máquina de flujo de datos

Para realizar la operación $AxB = a_1xb_1 + a_2xb_2$, cada operación a realizar se codifica en un patrón único (MPY y ADD, en este caso), cuando los datos están disponibles se pone la dirección del patrón en la cola de ejecución y se asigna un procesador para realizarla. Esto genera un paquete de resultados que sirve como dato para el siguiente patrón. Este tipo de máquinas plantea dos problemas importantes, los lenguajes de programación (muy diferentes de los tradicionales, complejos y con serias limitaciones) y la red de comunicación necesaria para llevar los datos de un procesador a otro, un aspecto crítico en todos los ordenadores paralelos. Una posible aplicación de estas arquitecturas es utilizarlas como unidades funcionales de un ordenador más general (al igual que sucede con los arrays sistólicos).

Máquinas de programación funcional. Este tipo de máquinas se han diseñado para ejecutar diversos tipos de lenguajes funcionales y aprovechar así su paralelismo inherente. La gran ventaja de estas máquinas es que una vez que el programador está familiarizado con los lenguajes de programación funcional no ha de preocuparse de especificar el paralelismo ya que éste es inherente al lenguaje y no hace falta un análisis profundo de la estructura del programa. Sin embargo, sí es tarea del programador escoger una solución paralela o paralelizable y descomponer el problema de la forma adecuada, tarea nada sencilla y que es objeto de un fuerte debate dentro de los que estudian el paralelismo, ya que muchos proponen que el paralelismo debería ser sólo en la máquina, el programador resuelve el problema como lo ha hecho tradicionalemente y es el compilador el encargado de paralelizar el programa.

Procesadores múltiples. La reducción del precio de los procesadores completos ha producido un interés creciente en las máquinas que incorporan varios procesadores como elementos básicos de bajo coste. En estos sistemas cada procesador se puede programar completamente y ejecutar su propio programa, de esta forma se consigue una flexibilidad mucho mayor que en las estructuras anteriores aunque esta flexibilidad se traduce en una complejidad mucho mayor en el control y la programación. Con este tipo de diseño se pueden conseguir factores de aceleración casi lineales pero existen una serie de factores a tener en cuenta: si los procesadores se han de coordinar cada cierto tiempo, las prestaciones se ven seriamente afectadas, existen muchos procesos que no son paralelizables y sólo se pueden ejecutar de forma secuencial, los algoritmos paralelos requieren más pasos que los secuenciales y una sobrecarga adicional para gestionar ese paralelismo, que varios procesadores compartan recursos puede ralentizar mucho determinados procesos y, finalmente, operaciones como la entrada/salida pueden llegar a consumir un tanto por ciento muy elevado del tiempo de computación reduciendo las ventajas del paralelismo.

Estos son los cinco tipos básicos de arquitecturas paralelas. Por supuesto, se podrían hacer muchas otras clasificaciones (tema que ya tratamos al hablar de las taxonomías) atendiendo a criterios diversos. Aparte de esta clasificación, la otra forma de estudiar los ordenadores paralelos es atendiendo a las conexiones entre los diferentes elementos que los forman:

***Arrays* sistólicos.** Una forma muy eficiente de conectar los elementos de proceso es realizar estas conexiones de acuerdo con la tarea que van a realizar. La solución no es flexible en absoluto (una vez establecidas las conexiones, los procesadores sólo funcionarán eficientemente con la aplicación para la que fueron adaptados) pero se consiguen soluciones muy optimizadas. El problema fundamental que presentan es que esa optimización sólo ocurre en el proceso de los datos haciendo que la Entrada/Salida sea vital para conseguir unas buenas prestaciones.

Redes de Banyan. Para solucionar la falta de flexibilidad de los arrays sistólicos, se pueden sustituir algunos elementos de proceso por conmutadores programables que permitan cambiar la configuración de la red de procesadores ampliando el abanico de posibilidades a las que es aplicable, esto son las redes de Banyan. Tienen la gran ventaja de que como la conmutación es hardware, no hay pérdida de eficiencia en los cálculos. Algunas aplicaciones de este tipo de redes también solucionan en parte el problema de la E/S dejando los datos fijos y realizando operaciones sucesivamente sobre éstos.

Cubos-k. Los cubos k-binarios conectan 2^k procesadores. Las conexiones se hacen entre los vértices (procesadores) de un cubo de k dimensiones. En la figura aparece el caso k=3. Este tipo de conexiones ofrecen muchas rutas de datos posibles y son muy apropiadas para cierto tipo de problemas como las clasificaciones. Tienen el gran problema de que dado un número demasiado elevado de procesadores la conexión se haga imposible por problemas de cableado, es factible llegar a unos 64000 procesadores. Un ejemplo comercial es la famosa Connection Machine.

Cubos de ciclos conectados. Es un refinamiento de los cubos-k que intenta resolver los problemas de conexión que se plantean. En lugar de colocar un único procesador por vértice, se coloca un anillo de procesadores. El gran potencial de este tipo de conexiones está en problemas que se puedan dividir en varias tareas más pequeñas, que, además, lleven a soluciones que puedan combinarse para dar la solución al problema completo. Esta división de las tareas no es, en muchos casos, obvia y queda a la discrección del programador.

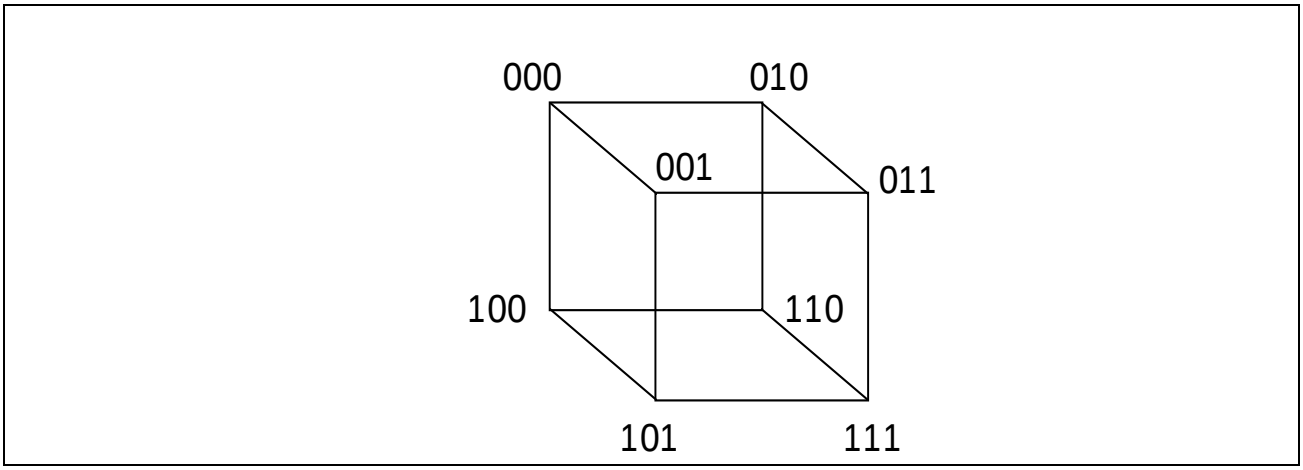


Fig.10 Esquema de conexión de un cubo K

6.2 Problemas paralelos

El paralelismo es más que una simple curiosidad en el mundo de las arquitecturas de ordenadores. No sólo permite resolver problemas ya conocidos de una forma más eficiente y rápida, sino que también permite abordar problemas totalmente nuevos que ni se podían plantear. Pero las grandes ventajas que ofrecen tienen un costo, a veces muy elevado. Ya hemos mencionado algo sobre este tema al enumerar algunas de las arquitecturas existentes y sus formas de conexión.

La posibilidad de paralelizar los diferentes niveles que se pueden considerar en un ordenador tiene distintas consecuencias según el nivel escogido (al hablar de estos niveles nos referimos a la jerarquía de Bell y Newell que veíamos al principio del capítulo). En los niveles más bajos, el paralelismo consigue una mayor velocidad en la ejecución y ciertas ventajas en cuanto a prestaciones del sistema, la duplicación del hardware (para conseguir tolerancia a fallos) o la ejecución segmentada de instrucciones son algunas formas de paralelismo muy utilizadas y con grandes ventajas pero sin que se traduzcan en un aprovechamiento real de lo que es el paralelismo en sí. Otro tipo de paralelismo es el que comentabamos al hablar de los arrays sistólicos, en los que la idea básica es disponer de un número elevado de unidades de cálculo (o de proceso, pero generalmente se utilizan en proceso numérico) que permitan realizar varias operaciones a la vez sobre un conjunto de datos. Según se conecten estas unidades de cálculo se tendrá mayor o menor flexibilidad, tal y como hemos visto anteriormente.

Programación estructurada y paralelismo

Ya quedan lejos los tiempos en que los ordenadores se programaban bit a bit, a través de conmutadores binarios. A la utilización de los lenguajes de ensamblador le siguieron lenguajes como Fortran o PL-1 en los que el grado de abstracción era mucho mayor. Tras éstos aparecieron poco a poco los primeros lenguajes de programación estructurada, orientados a la resolución de problemas por el conocido algoritmo "divide y vencerás". Pero la programación estructurada tiene una serie de problemas que se han ido conociendo con el tiempo. En primer lugar, y al contrario de lo que mantiene la creencia popular, la programación estructurada no facilita el diseño de programas, lo que facilita es su mantenimiento y depuración. Utilizar un lenguaje de este tipo obliga a descomponer el problema de forma coherente y estudiar una jerarquía, a veces muy compleja, de procedimientos y funciones. Hacerlo así facilita la depuración, pues cada procedimiento puede probarse por separado y es más sencillo localizar los posibles fallos. También facilita el manteniemiento pues las modificaciones son generalmente en los procedimientos, no en el programa completo. Pero, al igual que sucede con el paralelismo, y es lo que queremos resaltar aquí, un lenguaje de programación estructurada, al igual que un lenguaje de programación paralelo, traslada gran parte de la complejidad de diseño al programador que ha de resolver el problema de una forma que puede no ser la más adecuada su estructura. El paralelismo hace que esto sea aún más crítico, pues puede llegar a obligar al programador (macroparalelismo) no sólo a idear una solución paralela sino a controlar la ejecución concurrente y simultánea de varios subprogramas.

Pero el paralelismo realmente interesante es el que presentan los ordenadores que disponen de varios elementos de proceso (varias CPU's), capaces de ejecutar cada uno su propio programa. En este punto es donde comienza la controversia entre las diferentes formas de interpretar el paralelismo. La primera alternativa es elegir entre una arquitectura compuesta de muchos elementos de proceso simples, cada uno de ellos capaz de manipular sólo unos pocos datos (en el límite, procesadores de un bit), o utilizar una arquitectura con bastantes menos procesadores, pero éstos mucho más potentes (procesadores completos como los utilizados en máquinas monoprocesador). Y esta primera elección no es en absoluto trivial pues el conjunto de problemas abordables de forma directa con cada una de las arquitecturas es muy diferente. En cierta forma, el problema es similar al que discutíamos al hablar de los ordenadores RISC y CISC.

Otro punto muy importante dentro del paralelismo es optar por el macroparalelismo o hacerlo por el microparalelismo. Este último propone aprovechar la estructura paralela de algunas construcciones de lenguajes de programación tradicionales, sería el compilador el encargado de traducir el programa, escrito sin tener en cuenta la arquitectura paralela en que se va a ejecutar, a un código preparado para sacar partido de la existencia de varios procesadores. El macroparalelismo, en cambio, se plantea la resolución de problemas a través de algoritmos paralelos, es el programador el que debe estructurar la solución de forma que sea ejecutable en un ordenador paralelo y saque partido de las prestaciones que pueda ofrecer. Son dos enfoques radicalmente diferentes y que se basan en una interpretación distinta de la complejidad inherente a la programación.
Los más pesimistas consideran que no es factible trabajar con ordenadores paralelos, dado el gran número de factores a controlar y su complejidad (las comunicaciones entre los diferentes procesadores y procesos que se ejecutan, ejecución no determinista, efectos laterales, descomposición del problema para poder resolverlo con un algoritmo paralelo, etc.), por lo tanto, el esfuerzo de paralelización ha de hacerse en tiempo de compilación trasladando la complejidad del programador al compilador que se encargaría de optimizar el programa y paralelizarlo (microparalelismo). Esto tiene la ventaja de que no hay que cambiar los hábitos de programación y los programas se ejecutarían mucho más eficientemente. Por ejemplo, un bucle "for i := 1 to 100 do" podría repartir la tarea entre cinco procesadores asignándo a cada uno un tramo del recorrido del índice (1-20, 21-40, 41-60, 61-80, 81-100).

Sin embargo, esta solución tampoco está exenta de problemas muy complicados de resolver, entre ellos, la construcción de compiladores capaces de discernir entre estructuras paralelizables y las que no lo son, sin intervención del programador. Por otro lado, existen muchos problemas cuya solución es naturalmente paralela y tener la posibilidad de ejecutarlos de esa manera sería una gran ventaja. Aquí es donde aparecen los partidarios del macroparalelismo, argumentando que es factible manejar la compejidad que genera un ordenador paralelo a través de las herramientas y los lenguajes de programación adecuados. Aplicaciones como la predicción del tiempo atmosférico, los estudios aerodinámicos, las grandes simulaciones y el proceso de imágenes, son ejemplos de problemas donde el paralelismo se puede utilizar de forma natural, e incluso es la única forma factible de resolverlos. Cierto es que, hoy por hoy, existen más ordenadores paralelos que herramientas y lenguajes adecuados para utilizarlos, pero no hay ninguna duda sobre la importancia de este tipo de arquitecturas y el importante papel que van a desempeñar en el futuro de la computacuión.

7. Resumen

A lo largo de este capítulo hemos tratado algunos aspectos destacados de la tecnología hardware intentando resaltar su importancia en la innovación tecnológica y el papel que juegan en el desarrollo de lo que se ha dado en llamar tecnologías de la información. Por supuesto, nos hemos dejado muchos temas e incluso sobre los que se han tocado cabría decir mucho más, ya mencionábamos al principio que el hardware comprende un sinfín de cuestiones que sería demasiado complicado y prolijo desarrollar por completo.

De este capítulo conviene destacar dos cosas. Primero, el hecho que hemos intentado resaltar continuamente de que el hardware es el auténtico motor del desarrollo tecnológico y el que propicia las pocas o muchas "revoluciones" que en este campo se dan. Tanto es así, que es frecuente

encontrar tecnología en busca de aplicaciones en las que se puedan aprovechar las prestaciones que ofrecen los equipos y esto sucede no sólo con ordenadores sino también con los periféricos (impresoras láser, discos ópticos, etc.) y todo tipo de tecnologías hardware, desde los circuitos integrados a las redes de comunicación para ordenadores. Esto tiene un lado contraproducente que es la enorme velocidad de cambio y evolución de este campo, algo que se traduce en una dificultad enorme para comprenderlo totalmente. A la complejidad inherente al hardware se añade la complejidad de descripción de un objeto polifacético y constantemente en transformación. Por ello es necesario aproximarse al problema con una perspectiva global y para ello nada mejor que las herramientas y estructuras conceptuales que hemos ido desarrollando en capítulos anteriores.

En segundo lugar, un factor importante de la tecnología es la complejidad que ella misma genera. Un ordenador permite manejar complejidad pero al mismo tiempo es un generador de complejidad. El hardware no es una excepción. A medida que surgen equipos más y más potentes con innovaciones que permiten aumentar en muchos órdenes sus prestaciones (y por tanto manejar más complejidad) cada vez es más difícil controlar estos ordenadores. El ejemplo del paralelismo es muy claro en este punto. La enorme ventaja que supone tener varios procesadores trabajando al mismo tiempo se traduce en una enorme complejidad de programación, independientemente de quién la gestione el programador o el compilador. Poder integrar circuitos en VLSI implica poder construir ordenadores más potentes y compactos, con menos consumo y más eficientes, pero también implica disponer de las herramientas necesarias para poder diseñar algo tan complejo como un circuito con 100.000 transistores. Esto conduce a una especie de paradoja de la tecnología que puede explicar en parte lo vertiginoso de su avance e incluso determinadas crisis que se producen: la tecnología genera tecnología. Cuando se consigue un equipo con unas prestaciones X el avance que supone respecto a equipos anteriores permite perfeccionar el diseño (y el desarrollo, y las pruebas y las simulaciones, etc.) de nuevos equipos con lo que al tener el equipo X se está en disposición de poder desarrollar un equipo Y aún más potente y así una y otra vez. ¿Cuando se para ese ciclo?, cuando la tecnología ha avanzado tanto que no hay herramientas software capaces de aprovechar la potencia que ofrecen o no existen aplicaciones en las que esa potencia se pueda utilizar (esto último puede parecer paradójico pero las aplicaciones depende mucho del estado del arte de cada campo concreto, de los intereses que existan en el momento y de otras tecnologías que pueden estar menos avanzadas). Un ejemplo de ello es la tecnología de comunicaciones que siempre va muy por delante de los productos comerciales, ofreciendo anchos de banda varios órdenes de magnitud por encima de lo que se está utilizando.

Bibliografía

La dividimos en dos apartados. En el apartado de Notas Bibliográficas se comentarán aquellos trabajos que más profusamente han servido para redactar las páginas anteriores. El apartado de Referencias Bibliográficas contiene todos los trabajos citados.

Notas bibliográficas

Bibliografía sobre clasificaciones de arquitecturas: Los dos artículos que hemos utilizado como referencia para esta parte son dos buenos ejemplos de cómo tratar el problema de las clasificaciones.

Además son complementarios ya que el trabajo de Dasgupta pretende completar el de Skillicorn. El artículo de Dasgupta resulta especialmente interesante por el estudio previo que hace de las taxonomías y las características que toda buena clasificación debe reunir. Skillicorn, David B. "A Taxonomy for Computer Architectures", **IEEE Computer**, Noviembre 1988, pp. 46-57; Dasgupta, Subrata, "A hierarchical taxonomic system for computer architectures", **IEEE Computer**, Marzo 1990, pp. 65-74

Bibliografía sobre Ordenadores Personales: Si hay un tema favorito de los estudiosos del mundo de los ordenadores, es sin duda el de los ordenadores personales. Citar incluso una bibliografía básica resumida nos podría llevar páginas enteras. Bástenos aquí con un par de referencias ineresantes: un estudio completo sobre el mundo de los ordenadores personales se puede encontrar en **Computadores Personales**, de F. Sáez Vacas, editado por Fundesco (Madrid, 1987) que, además, tiene la ventaja de estar en español. Una buena referencia a la historia de los pc´s y su problemática es el número de Mayo de 1986 del **IEEE Spectrum**, dedicado por completo a los ordenadores personales. Los números de Enero de esta misma revista presentan una análisis del estado del arte de la tecnología, durante los últimos años buena parte del interés se ha centrado en los pc's por lo que puede ser una fuente importante de datos.

Bibliografía sobre arquitecturas RISC: Las referencias utilizadas no entran en detalles demasiado técnicos pero presentan una discusión muy interesante de la aportación de este tipo de arquitecturas y la motivación que conduce a plantearse un problema de este tipo. Gimarc, C.E. y Milutinovic, V.M. "A Survey of RISC Processors and Computers of the Mid-1980s", **IEEE Computer**, Septiembre 1987, pp. 59-69, es un buen artículo para estudiar las características básicas de este tipo de ordenadores y cómo se resuelven los problemas que plantean. En Sáez Vacas, F. **Computadores Personales**, Fundesco 1987, pp. 58-62, se hace un estudio sobre las arquitecturas RISC desde el punto de vista de la complejidad, resaltando sobre todo las implicaciones que conlleva el diseño RISC. Tabak, D. "Which system is RISC?", **IEEE Computer**, Octubre 1986, pp. 85-86, es un artículo breve en el que se discute muy acertadamente cómo se deben clasificar los sistemas RISC y cuáles son las características más sobresalientes de estos ordenadores.

Bibliografía sobre paralelismo: Sobre este tema la bibliografía es abundantísima por lo que nos limitaremos a citar algunas referencias fáciles de localizar y que pueden servir muy bien como introducción a los ordenadores paralelos. Existen dos números del **IEEE Computer** dedicados al paralelismo, uno es el de Agosto de 1986 y otro el de Enero de 1982. En esta misma revista existen otros muchos artículos dedicados a tratar aspectos concretos siempre a un nivel más o menos asequible. Para encontrarlos, los números de Dicienbre de todos los años llevan un índice anual en el que la entrada Paralelismo es muy frecuente y con muchos artículos. También hay que citar el **IEEE Spectrum** que en un tono más divulgativo tiene artículos muy interesantes en los que se comenta el estado del arte del paralelismo, en concreto, se puede consultar "A parallel architecture comes of age at last" de P. Wiley, en el número de Junio de 1987, pp. 46-50.

Referencias bibliográficas

Alonso, G. (1989) **"Paralelismo, nuevas proposiciones desde la cibernética"**, premio Antonio Fernández Huertas de la Sección Española del IEEE.

Dasgupta, Subrata (1990) "A hierarchical taxonomic system for computer architectures", **IEEE Computer**, Marzo, pp. 65-74.

Engineering Tools (1988), número 1, volumen 1, dedicado a los ordenadors personales y estaciones de trabajo, Febrero.

Fernández G, y Sáez Vacas, F. (1984) **Fundamentos de los ordenadores**, ETSIT, Madrid

Gelernter, D. (1986) "Domesticating parallelism", **IEEE Computer**, Agosto, pp.12- 16.

Gimarc, C.E. y Milutinovic, V.M. (1987) "A Survey of RISC Processors and Computers of the Mid-1980s", **IEEE Computer**, Septiembre, pp. 59-69.

Haynes, Siewiorek et al. (1982) "A survey on Highly Parallel Computing", **IEEE Computer**, Enero, pp. 9-22.

LoPiccolo, P.J. (1988) "High-End Workstations", **Engineering Tools**, num. 1, vol. 1, Febrero pp. 80-95.

Matsumura, T. (1983) "Future Microprocessor trends", **Congreso Mundial IFIP**, pp. 213- 217.

Murakami et al. (1985) "Research on a Paralell Machine for fifth generation computer systems", **IEEE Computer**, Junio 1986, pp. 76-91.

Pagels, H.R. (1989) **The dreams of reason. The computer and the rise of the sciences of complexity**, Bantam Books, Nueva York.

Sáez Vacas, F. (1983) "Facing informatics via three level complexity views", **X International Congress on Cybernetics**, Namur, Bélgica, pp. 30-40.

Sáez Vacas, F. (1987) **Computadores Personales**, Fundesco, Madrid.

Skillicorn, David B. "A Taxonomy for Computer Architectures", **IEEE Computer**, Noviembre , pp. 46-57.

Tabak, D. (1986) "Which system is RISC?", **IEEE Computer**, Octubre, pp. 85-86.

Wiley, P. (1987) "A parallel architecture comes of age at last", **IEEE Spectrum**, Junio, pp. 46-50.