

ORDENADORES. FILOSOFIAS Y DESARROLLOS  
ACTUALES Y FUTUROS

Por F. Sáez Vacas  
Catedrático de Ordenadores  
E.T.S.I.T.

*Hewlett Packard 30 Mayo 1977*

INDICE

0. INTRODUCCION .....	1
I. HARDWARE .....	1
I.1. Esquema funcional .....	1
I.2. Dimensiones del espacio del ordenador .....	6
I.3. Tecnología y estructura .....	8
I.4. Memorias .....	11
I.4.1. Memorias asociativas y aplicaciones .....	17
I.4.1.1. Memoria "Cache" .....	19
I.4.1.2. Memoria Virtual .....	21
I.4.2. Microprogramación .....	26
I.5. Propositiones de Sáez (1.973) sobre condiciones de viabilidad de innovaciones en técnica o tecnología .....	30
I.6. Arquitecturas de ordenador .....	31
I.6.1. Sistemas multiprocesadores y paralelos ...	32
I.6.2. Sistemas minis y micros .....	46
I.6.2.1. Estructura-robot y aplicaciones de los minis .....	47
I.7. Ordenador promedio típico del año 2.000 .....	55
II. SOFTWARE .....	56
II.1. Clasificación y algunas definiciones .....	56
II.2. Problemas del software .....	60
II.3. Estructura y costes del proceso de desarrollo y mantenimiento del software .....	63
II.4. Las técnicas de programación estructurada ....	66
II.5. El futuro del software. Resumen de problemas y recomendaciones .....	70

III. INFORMATICA .....	74
III.1. ¿Cómo y en qué utilizar el ordenador? ....	74
III.2. Sistemas centrados en una base de datos ..	78
III.3. La Informática repartida .....	81
REFERENCIAS .....	83

## 0. INTRODUCCION

Esta conferencia pretende presentar una panorámica de las actuales estructuras de ordenador, de las principales corrientes tecnológicas ya concretadas a nivel de producto tangible, de algunos aspectos y problemas de software, y de presente discusión acerca de las informáticas concentrada o repartida.

Se supone que los asistentes son de muy dispar función profesional, pero todos saben ya qué es y cómo funciona un ordenador (ver conferencia sobre Microprocesadores).

Es de observar que nuestro principal énfasis se centra en la arquitectura de los ordenadores, ya de por sí tan compleja que se suele desarrollar durante uno o varios cursos completos. Los aspectos de software se mencionan y se recorren muy superficialmente puesto que, en definitiva, deberían apoyarse en una profundización de los de la arquitectura.

Por último, se destaca que estos apuntes no son más que un soporte escrito de la conferencia, que puede ser aprovechado por los asistentes más interesados para hacer sobreanotaciones.

## I. HARDWARE

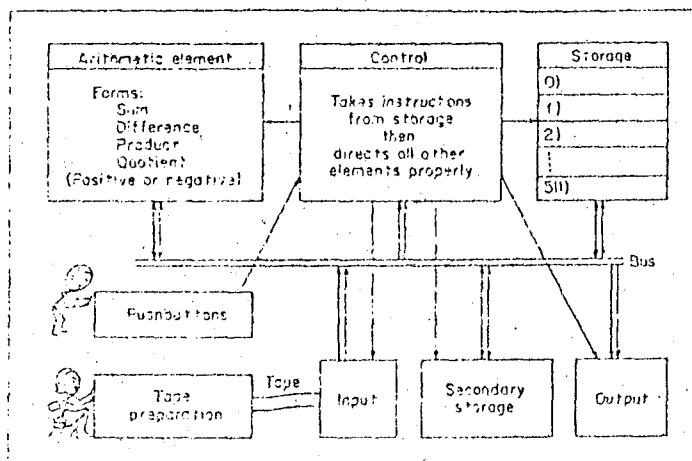
### I.1. Esquema funcional

Suele representar las principales funciones de un ordenador, que son:

Unidad de Memoria	}	Unidad Central
Unidad de Control		} Procesador Central
Unidad Aritmética-Lógica		

Unidad de Entrada/Salida.

Ver esquema del ordenador Whirlwind:



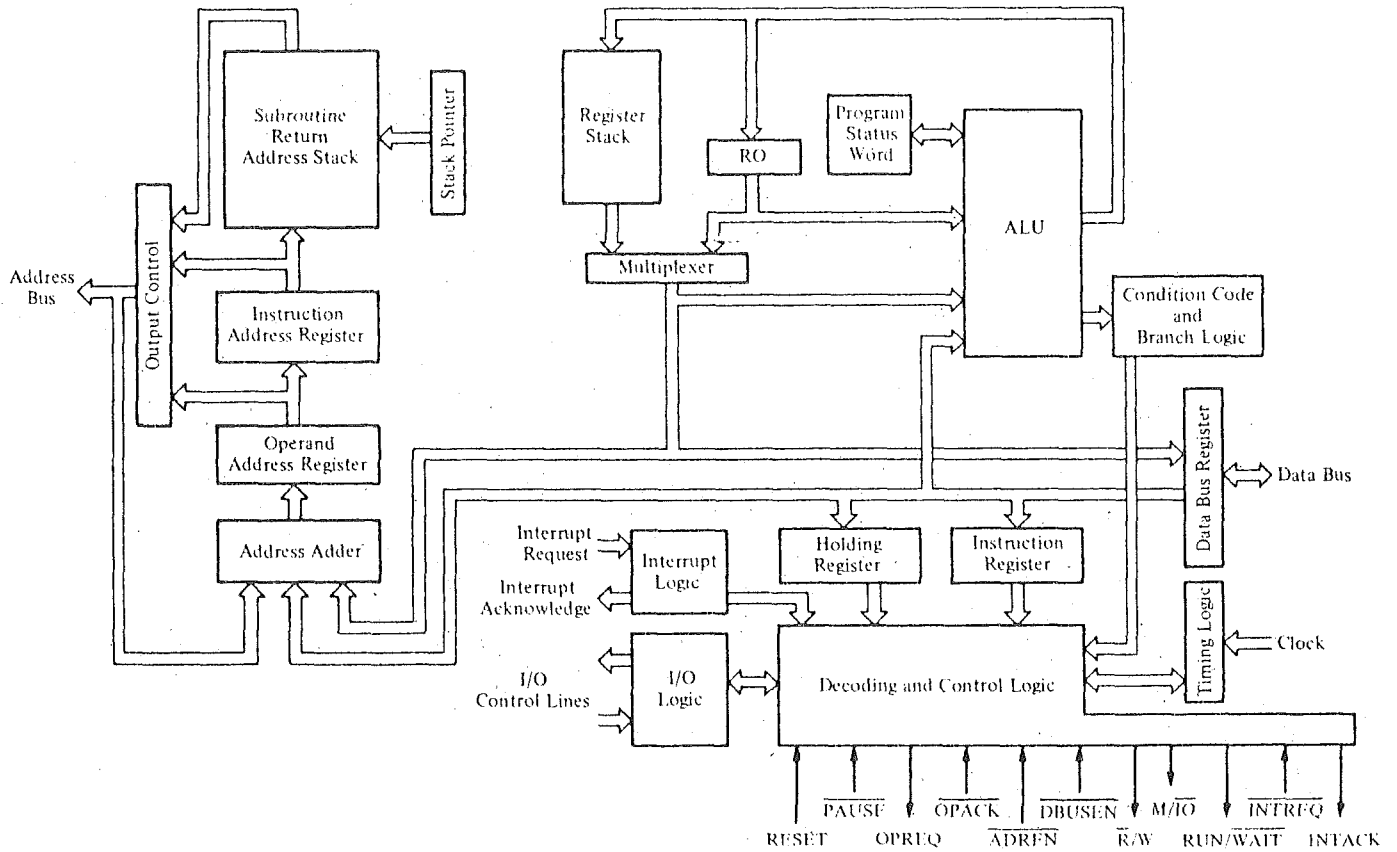
Whirlwind.

Estas funciones siguen existiendo, pero se han escindido ya en numerosas unidades interrelacionadas y jerarquizadas, lo que hace que un esquema como el anterior no represente más - que muy superficialmente la estructura real de un ordenador.

Yendo a mayor detalle hay que utilizar esquemas como - el siguiente (ver pág. 3).

El problema es ¿hasta donde puede llegarse en el detalle de representación?. Profundizando el detalle se podría ir - hasta el circuito lógico y esto, para los especialistas en estructura de ordenadores, sería demasiado. Por otro lado, la figuración mediante cajas y flechas es ambigua, depende mucho del individuo que la emplea.

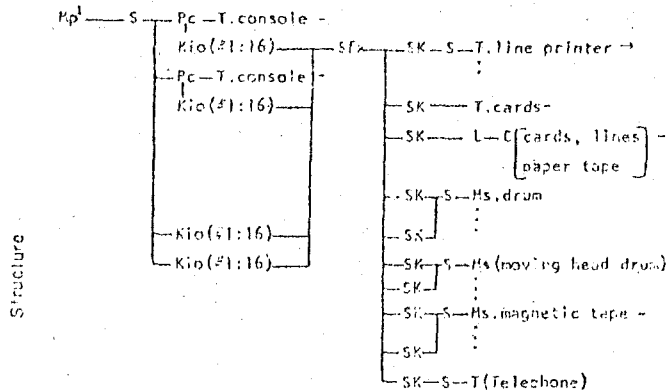
Bell y Newell han atacado este problema y han elaborado unos sistemas de representación, con unos elementos primitivos perfectamente definidos y una notación simbólica a varios -



ESQUEMA DE BLOQUES DEL SIGNETICS 2650 (FUENTE: FOSTER)

niveles o capas de detalle. El libro donde han descrito esta - aproximación es hoy un clásico en el género y absolutamente indispensable en la biblioteca de los especialistas.

A título de ejemplo, recogemos aquí un ordenador des-- crito al nivel PMS:



where:  
 rp/primary memory; Ps/secondary memory;  
 Pc/central processor; T/terminal; and L/link  
 S/switch; K/control; Ki/control for i/o equipment;  
 1Mp(#0:7: core; 32768 word)

Estructura representada en PMS.

Otros niveles reconocidos por Bell/Newell son los siguientes:

- . Nivel Programación
  - . Nivel Diseño Lógico
  - . Nivel circuito
- { Subnivel RT (Register Transfer)
- { Subnivel circuitos conmutación:
- . secuencial
  - . combinacional

A continuación transcribimos la definición de los componentes que intervienen en el nivel PMS (Processor Memory Switch).

MEMORIA, M. Es un componente que retiene o almacena información (es decir, unidades de información, unidades *i*) a lo largo del tiempo. Sus operaciones son leer unidades *i* de la memoria y escribir unidades *i* en la memoria. Toda memoria que almacene más de una unidad *i* debe comportar, asociadamente, un sistema de direccionamiento, por medio del cual pueda seleccionar una unidad particular. Puede también considerarse a una memoria como un conmutador a un cierto número de submemorias. Las unidades *i* no se modifican en modo alguno al ser almacenadas en memoria.

INTERCONEXION, (Link; L). Es un componente que transfiere información (unidades *i*) de un lugar a otro en el sistema ordenador. Tiene puertas fijas. Su operación consiste en transmitir una unidad *i* (o una secuencia de unidades) desde una puerta a otra del componente. Excepto el cambio de posición, no se produce ninguna otra modificación en las unidades *i*.

CONTROL, K. Es un componente que activa las operaciones de otros componentes en el sistema. Todos los demás componentes son considerados como consistentes en un conjunto de operaciones discretas, cada una de las cuales, al ser activada,

realiza alguna transformación discreta de estado. Con la excepción de un procesador, P, el resto de componentes es esencialmente pasivo, por lo que requiere algún agente activo (un K) para entrar en período de actividad.

CONMUTADOR, (Switch, S). Es un componente que construye un enlace entre otros componentes. A cada conmutador va asociado un conjunto de enlaces posibles y sus operaciones consisten en tender algunos de estos enlaces y cortar otros.

TRANSDUCTOR, T. Es un componente que modifica la unidad i utilizada para codificar un determinado significado. El cambio puede afectar al soporte (p. ej. niveles de tensión a flujo magnético, o niveles de tensión a perforaciones en una tarjeta de cartulina), o a la misma estructura de la unidad i (p. ej. bits en serie a bits en paralelo). Los componentes T preservan el significado, pero no necesariamente la información (en el sentido del número de bits).

OPERADOR DE DATOS, D. Es un componente que produce unidades i con nuevos significados. Este tipo de componente realiza todas las operaciones con datos, p. ej. operaciones aritméticas, lógicas, de desplazamiento, etc.

PROCESADOR, P. Es un componente capaz de interpretar un programa en orden a la ejecución de una secuencia de operaciones. Consiste en un conjunto de operaciones de los tipos ya mencionados -M, L, K, S, T y D- más el control necesario para obtener instrucciones de una memoria e interpretarlas como operaciones a ejecutar.

### Comentarios

- Complejidad

Dificultad de representación

Dificultad de comprensión



- Falta de normalización
- Más dificultad de representación, comprensión y comunicación.
- Dificultad de comparación, comunicación y de evaluación.

## I.2. Dimensiones del espacio del ordenador

El cuadro puede dar una idea de un espacio multidimensional, en cada uno de cuyos ejes se identifican distintos valores posibles. Un ordenador concreto quedaría situado en ese espacio por la especificación de los valores que le corresponden (uno o más valores por cada dimensión-eje). Su situación en el espacio mencionado le indica al especialista cuales son sus características funcionales y operativas, tanto en valor absoluto como relativamente a la constelación de ordenadores producidos.

FUNCION DEL ORDENADOR	
Centrifugado	
Control	
Comunicaciones	
(comunicación concentrada)	
Control de líneas	
Terminal	
Tiempo compartido	

TECNOLOGIA DE LA LOGICA	GENERACION	FECHA HISTORICA	VELOCIDAD P <sub>0</sub> (seg)	COSTE OPERACION (S/bit s)
Mecanica				
Electromecanica		1930	10 <sup>-1</sup>	1000
Fluorocata		(1930)	10 <sup>-2</sup>	
Tubo de vacio	Primera	1945	10 <sup>-3</sup>	10
Transistor	Segunda	1958	10 <sup>-8</sup>	1
Integrado		1964	10 <sup>-9</sup>	
Circuito integrado (IC)	Tercera	1966	10 <sup>-7</sup>	0.1
Integrado de mediana gran escala (MSI-LSI)	(Cuarta?)	(1970?)	10 <sup>-6</sup>	0.01

LONGITUD PALABRA	BASE	TIPOS DE INFORMACION
8 b	binaria	palabra
12 b	decimal	entera (direccion central)
16 b		entera (vector de bits)
24 b		entera (n)
32 b		entera (bitolante)
48 b		vector
64 b	caracter (6 b)	cadena de caracteres
	caracter (8 b)	vector de palabras
		vector
		matriz
		arrays
		listas, pilas

DIRECCIONES INSTRUCCION	ESTADO DE MEMORIA DEL PROCESADOR (excluyendo contador programático)
0 direcciones (pila)	pila
1 direccion	1 acumulador
1 + n (incluye) direcciones	acumulador y registros indices
1 + n (registros generales) direcciones	
2 direcciones	
3 direcciones	
n + 1 direcciones	estado no explicito
determinación de lenguaje	
compuesto	
microprogramado	

ESTRUCTURA PMS	CONMUTACION	FUNCION PROCESADOR
1 P <sub>0</sub>	1 n (duplex)	P microprograma
1 P <sub>0</sub> (n <sub>0</sub> (componente))		P <sub>0</sub>
1 P <sub>0</sub> (n P <sub>0</sub> )	n m (multiplex de tiempo)	P <sub>0</sub> (n <sub>0</sub> )
1 P <sub>0</sub> (n P <sub>0</sub> - P (display))		P <sub>0</sub>
2 C (duplex)	2 n (dual duplex)	P display
n P <sub>0</sub> (n (multiprocesamiento))	n m punto de cruce	
n P <sub>0</sub> - P (array algoritmo especial)		P array
n P <sub>0</sub> (procesamiento paralelo)		P mov. vectores
C (n <sub>0</sub> )		P algoritmo
Red	n-2 n/2 (sin jerarquia)	P lenguaje

ALGORITMO ACCESO	CAPACIDAD MP	CAPACIDAD MS	VELOCIDAD MP (b/s)	VELOCIDAD MS (b/s)
lineal (pila)				
lineal (cola)				
binario		esta muy grande		10 <sup>5</sup>
Circulo aleatorio		esta muy grande		
Ciclo	campo grande	esta muy grande		
Arbolario	nucleos (media)	nucleos (menor)	> 10 <sup>4</sup>	> 10 <sup>7</sup>
Por contenido	pequeña (pequeña)		> 10 <sup>4</sup>	
Asociativo	circuito integrado		> 10 <sup>7</sup>	

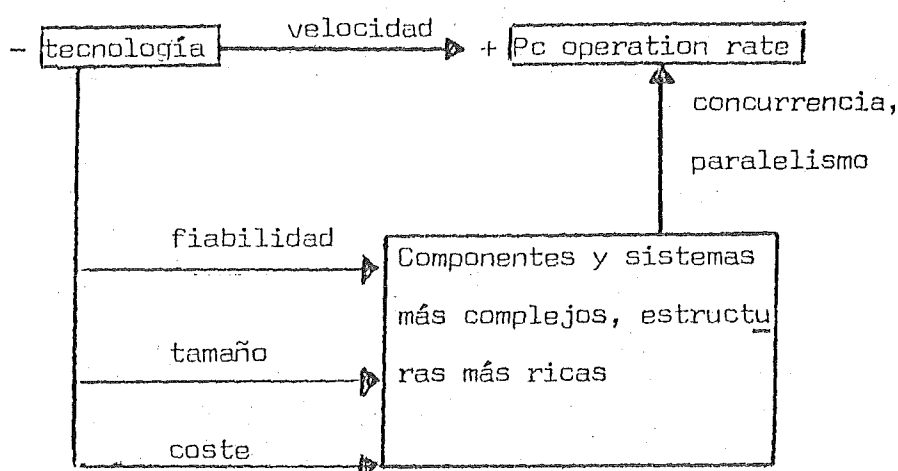
CONCURRENCIA M <sub>0</sub>	COMUNICACION INTERPROCESOS
1 Programa	subrutinas
1 Programa con interrupciones	interrupciones
1 Programa con múltiples subprogramas	interrupciones interprocesador
Concurrentes (p. ej. 1 P <sub>0</sub> n P <sub>0</sub> )	
Monitor o Prog. tipo (M) + 1 programa	extradatos (operadores programados para llamadas monitor)
m + n programas conjetables	
m + n programas (multiprogramación)	
Pila reubicación	
1 segmento	
2 segmentos (para, impuro)	
2 segmentos	
Pilas	
m + n segmentos con progr. compartidos	comunicación intersegmentos
segmentos programados, long. fija	
segmentos programados, long. multiple	
segmentos temporales, variable	
segmentos nominales	

CONCURRENCIA PROCESADOR
Serie bit a bit
Paralelo por palabras
Múltiples corrientes instrucciones 1 P <sub>0</sub>
Múltiples corrientes datos (array)
1 buffer p. instrucción
n buffers p. instrucción
memorias anticipatorias
Procesamiento pipe-line

## DIMENSIONES DEL ESPACIO DEL ORDENADOR

### I.3. Tecnología y estructura

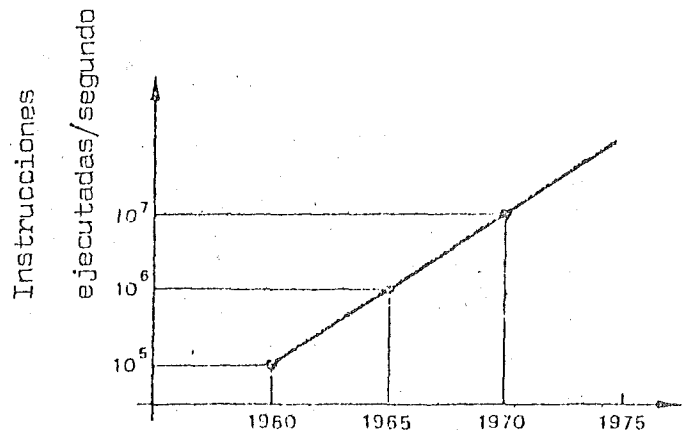
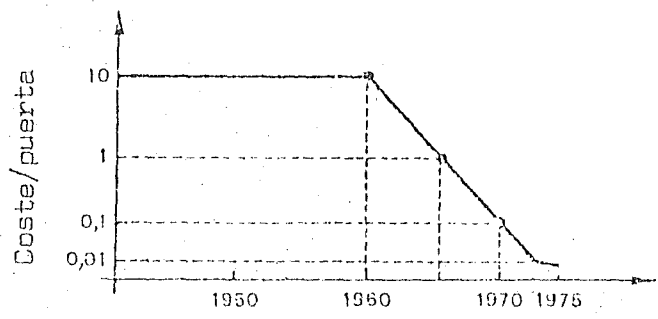
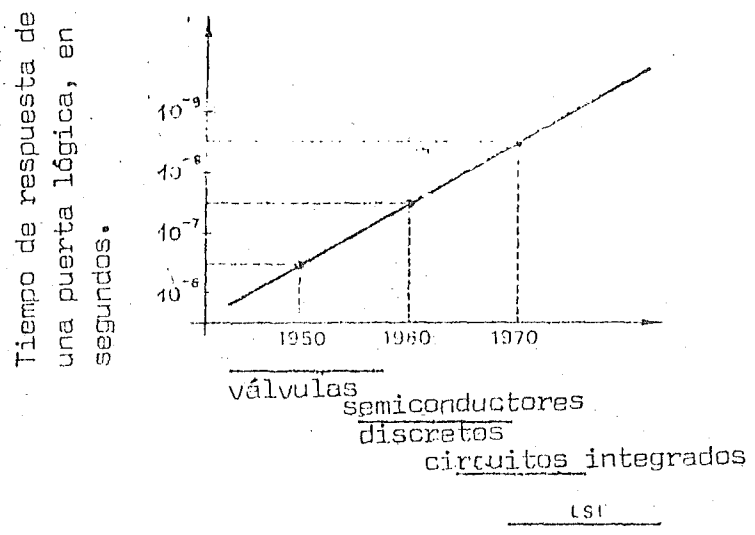
La tecnología es la dimensión más significativa o influyente de todo el espacio, ya que prácticamente condiciona los valores de todas las demás dimensiones. Aunque existen otros tipos de tecnología es la electrónica la que reviste más espectacularidad y trascendencia. Vemos en la figura cómo, por dos caminos diferentes, influye la tecnología en la velocidad de operación del Procesador central (Pc-operation rate).



VELOCIDAD DEL PROCESADOR CENTRAL

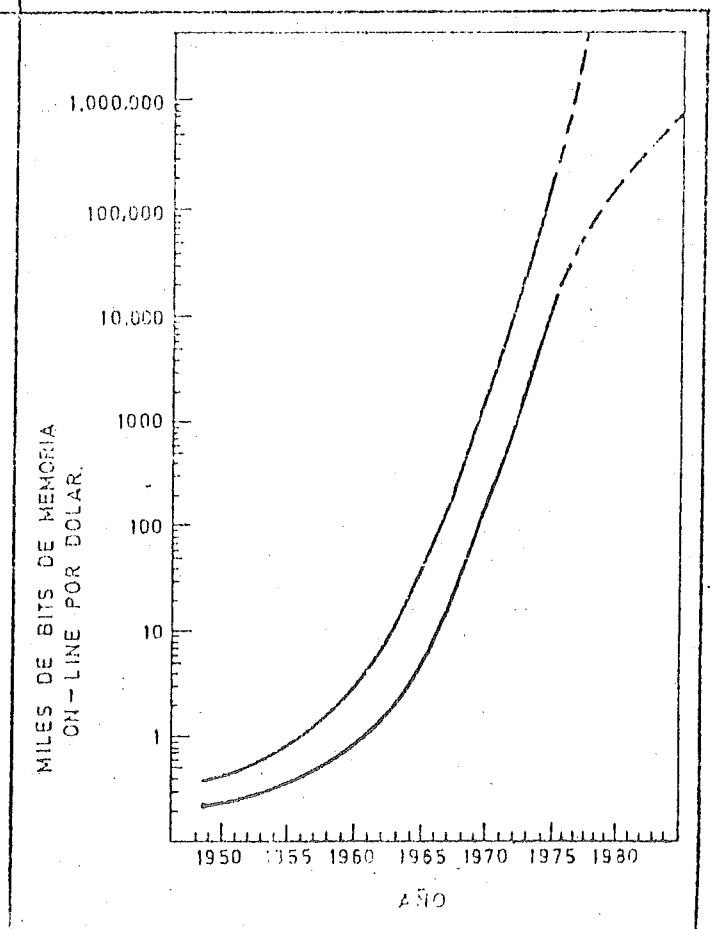
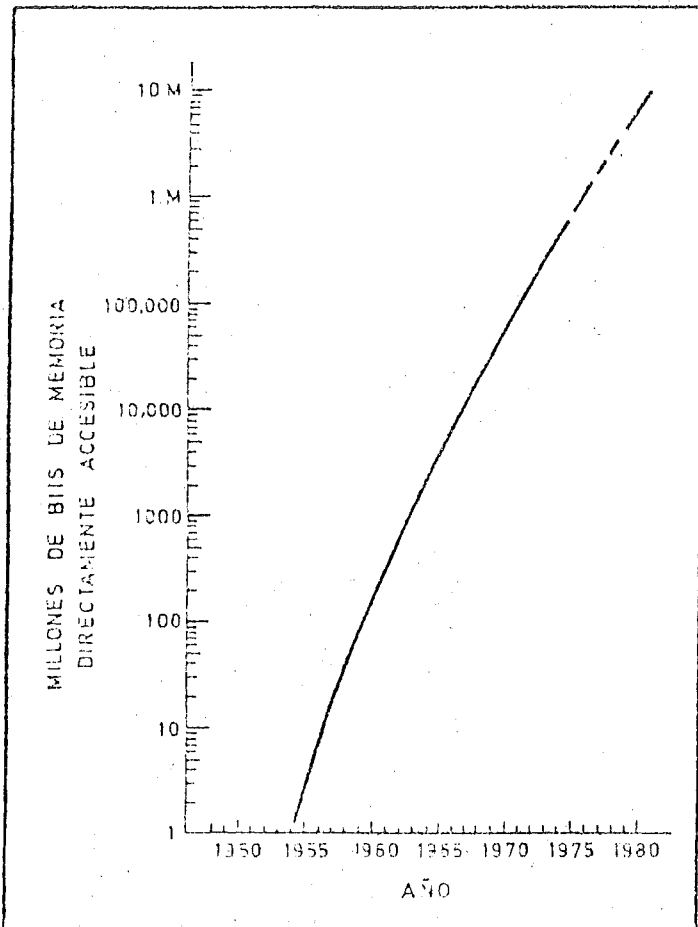
En particular, es muy importante el esfuerzo emprendido en una mejor estructuración y organización de los circuitos y unidades funcionales, lo que conduce a un mayor grado de conurrencia o paralelismo. Este punto ha dado lugar a unas familias de ordenadores que, por su nivel técnico, marcan el estado tecnológico actual del "arte" de la arquitectura de los ordenadores. Por esta razón y por el desconocimiento general -- acerca del tema, dedicaremos un amplio apartado a estas familias.

Las siguientes figuras señalan el efecto de la evolución de la tecnología en los costes y en la velocidad del procesador.



Contrastando la primera y tercera gráficas, se comprueba cómo la velocidad del procesador (instr/seg.) ha crecido relativamente más deprisa que la velocidad de una puerta lógica, a lo largo del tiempo. Esto se debe básicamente a los avances técnicos de diseño e implementación.

Las próximas curvas dan una idea de la misma evolución en el terreno de las memorias.



#### I.4. Memorias

Los circuitos integrados y ultimamente los microprocesadores representan puntos culminantes en la evolución tecnológica de los ordenadores y de manera general de los sistemas digitales. Estos temas han sido exhaustivamente presentados por conferenciantes anteriores.

Interesa resaltar ahora que los circuitos a base de semiconductores e integración a gran escala suponen que, por primera vez, es posible construir toda la estructura de la unidad central de un ordenador a base de la misma tecnología electrónica. Esto es un hito en la historia de los computadores.

Veamos algunos elementos acerca de las memorias centrales de un ordenador. (El cuadro siguiente presenta una panorámica general de las memorias más conocidas, no solamente centrales, al año 1.971 (Bell/Newell)).

Tecnológicamente, la memoria central de un ordenador puede ser de:

- a) núcleos de ferrita
- b) película delgada magnética
- c) semiconductores.

La memoria de núcleos es demasiado conocida, precisamente fué introducida por vez primera en el computador Whirlwind (ver I.1.), y la memoria de semiconductores se ha tratado en otra conferencia.

Por consiguiente, demos algunos datos de la tecnología de película delgada magnética en la técnica de hilo recubierto.

Se basa en una película muy fina de material magnético anisotrópico con el que se recubre un hilo de cobre-berilio, -

CARACTERISTICAS DE LAS MEMORIAS

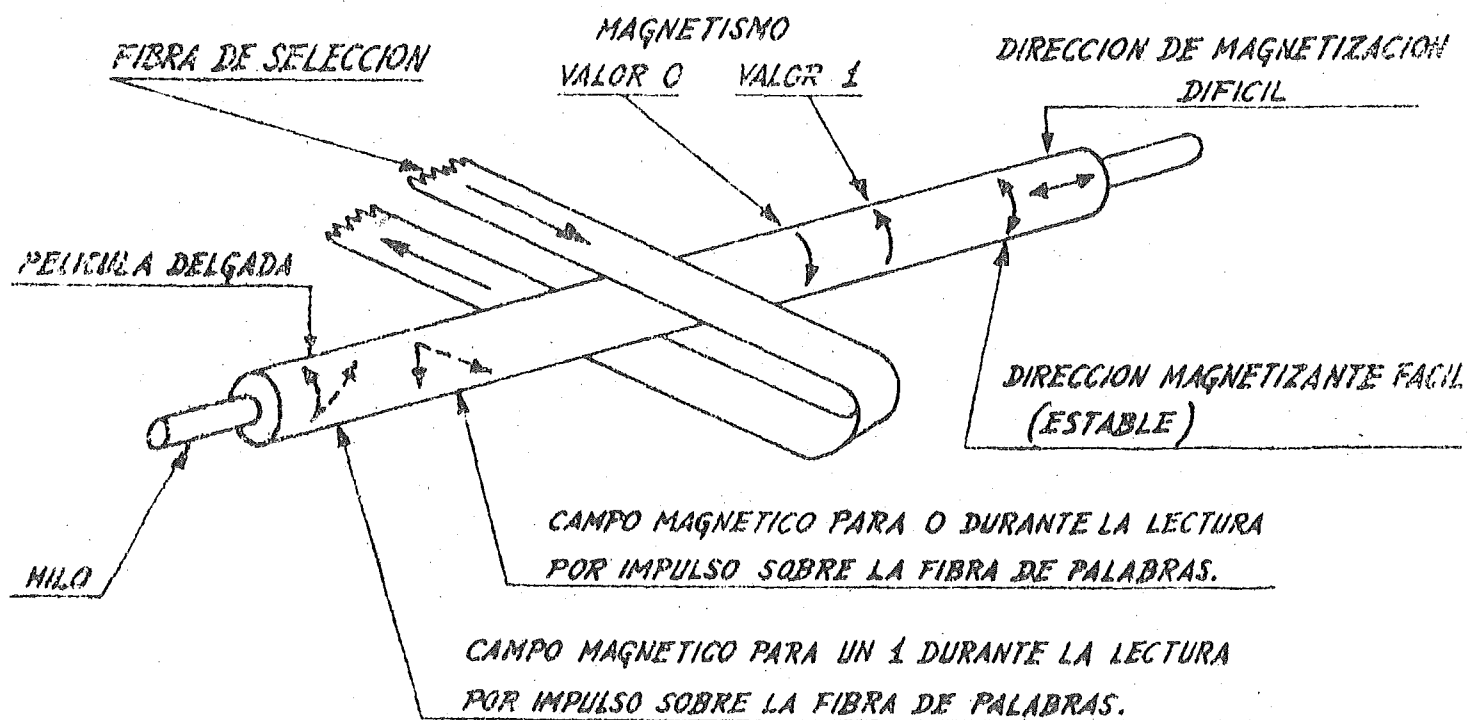
MODULO MEMORIA	FUNCION	MODO DE ACCESO	DIMENSION		RENDIMIENTO		COSTE/BIT (\$) (Soporte + transductor)
			CAPACIDAD DEL MODULO (bits)	MODULOS/ COMPUTADOR	TIEMPO DE ACCESO (Seg.)	CAUDAL (bits/seg.)	
Tarjeta perforada	permanente, fichero	random + secuencial	(500~1000)/tarjeta; ~1000 tarj/ unidad	1 ~ 2	$10^0 \sim 10^3$	$10^4$	$2 \times 10^{-6} + 2 \times 10^{-1}$
Tarjeta magnética	secundaria, fichero	secuencial + constante + cíclico	$3 \times 10^9$	1 ~ 4	$10^{-1} \sim 10^0$	$0,4 \times 10^6$	$1,5 \times 10^{-8} + 5 \times 10^{-5}$
Cinta magnética	secundaria, fichero	secuencial	$2 \times 10^8$	1 ~ 16	$10^0 \sim 10^2$	$0,4 \sim 4 \times 10^6$	$2 \times 10^{-7} + 10^{-4}$
Disk-pack de cabezas móviles	secundaria, "swapping" de ficheros	secuencial + cíclico	$2 \times 10^8$	1 ~ 16	$10^{-1} \sim 10^0$	$2,5 \times 10^6$	$3 \times 10^{-6} + 10^{-4}$
Disco cabezas fijas	secundaria "swapping" de ficheros	cíclico	$5 \times 10^7$	1 ~ 40	$\sim 10^{-2}$	$10^6 \sim 10^7$	$10^{-3}$
Tambor	secundaria "swapping"	cíclico	$(1 \sim 5) \times 10^7$	1 ~ 10	$(5 \sim 30) \times 10^{-3}$	$10^6 \sim 10^7$	$10^{-3}$
Memoria extendida	primaria y/o secundaria, "swapping"	random	$10^7$	1 ~ 8	$(2 \sim 10) \times 10^{-6}$	$10^6 \sim 10^8$	$0,02 \sim 0,05$
Núcleos alta velocidad o película delgada	primaria	random	$10^5 \sim 10^6$	1 ~ 16	$(0,2 \sim 2) \times 10^{-6}$	$10^7 \sim 10^8$	$0,05 \sim 0,25$

CARACTERISTICAS DE LAS MEMORIAS

MÓDULO MEMORIA	FUNCION	MODO DE ACCESO	DIMENSION		RENDIMIENTO		
			CAPACIDAD DEL MODULO (bits)	MODULOS/ COMPUTADOR	TIEMPO DE ACCESO (Seg.)	CAUDAL (bits/seg.)	COSTE/BIT (\$) (Soporte + transductor)
Circuito integrado (scratch-pad)	primaria, estado procesador	random	$10^3 \sim 10^5$	1	$\sim 10^{-7}$	$10^9$	0,25 ~ 1,0
Circuito integrado (asociativa)	primaria tampón	por contenido random	$2 \times 10^5$	1 ~ 2	$\sim 10^{-7}$	$10^9$	1 ~ 3
Real only (capacitor, inductor)	procesador	random	$(1 \sim 5) \times 10^5$	1	$10^{-6} \sim 10^{-7}$	$10^8 \sim 10^9$	$10^{-3} \sim 10^{-2}$

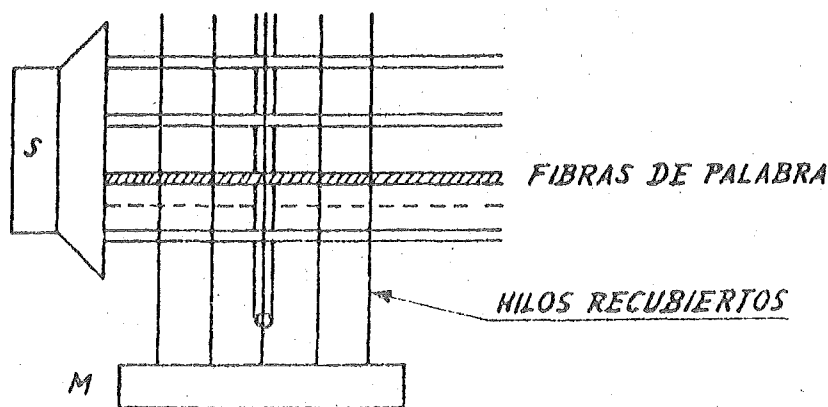


que actúa a la vez como hilo de lectura y de escritura. El hilo puede tener de 2 a 5 milésimas de pulgada de diámetro recu-



bierto por una película de unas 30 micropulgadas, compuesta en un 81,5% de níquel y en un 18,5% de hierro. (Sáez, 1.975).

La organización de la memoria es del tipo 2D como se representa en la figura.



Para almacenar una información se aplica una corriente por la fibra selectora (fibra de palabras) y al tiempo una corriente por el hilo de bit

Leer información se consigue enviando un impulso de corriente por la fibra de palabra, provocando con ello un giro de la magnetización del eje fácil (circular) al difícil (longitudinal). Durante el tiempo de subida del impulso de corriente, la rotación de la magnetización induce una tensión en el hilo de bit, cuya polaridad será positiva o negativa según el bit estuviera imantado en un sentido o en otro. Al no sobrepasar el campo de palabra la región de "no conmutación", normalmente un giro de 45° del momento magnético, la imantación vuelve a su estado inicial una vez desaparecido el impulso de corriente, con lo que se consigue una lectura no destructiva.

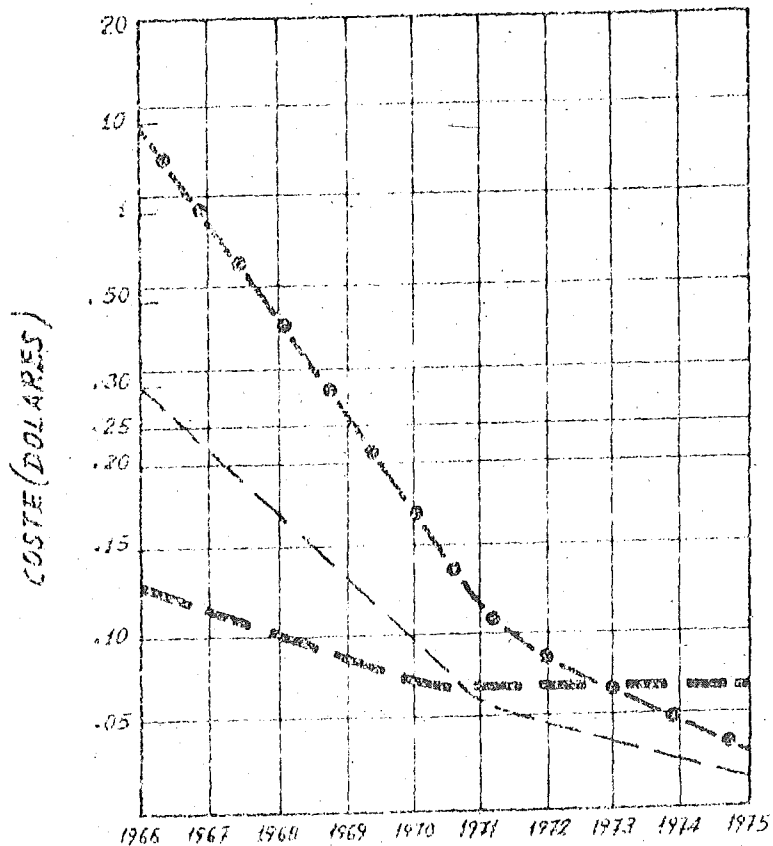
La memoria de hilo recubierto tiene numerosas interesantes características de sencillez de operación, velocidad, bajo consumo, densidad de información, tolerancia a los ambientes especialmente duros. Parecía que podría haber conseguido un puesto privilegiado en el mercado y no ha sido así, lo que puede comentarse en la conferencia, como muestran las siguientes gráficas.

Tecnologías basadas en el fenómeno de las burbujas magnéticas o en el laser no han pasado al estado de comercialización y las dejaremos de momento.

Hay, sin embargo, distintas funciones o refinamientos en las memorias actuales que, más o menos extendidos, empiezan a ser corrientes y con gran probabilidad se extenderán aún más. Se trata de técnicas de organización de las unidades de memoria, como la memoria asociativa, o de técnicas de utilización de ciertas memorias, como la memoria cache, la memoria virtual y por último la microprogramación.

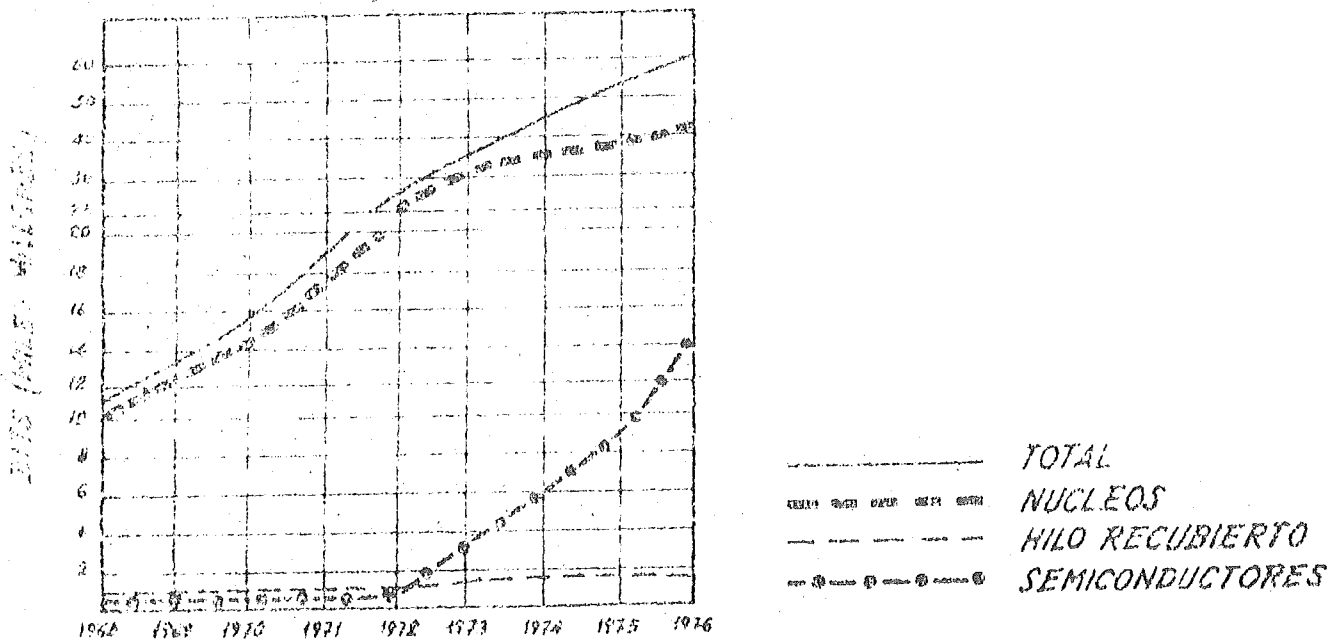
COSTES POR BIT

1966-1975



CONSUMO MEMORIA (BITS)

1968-1976



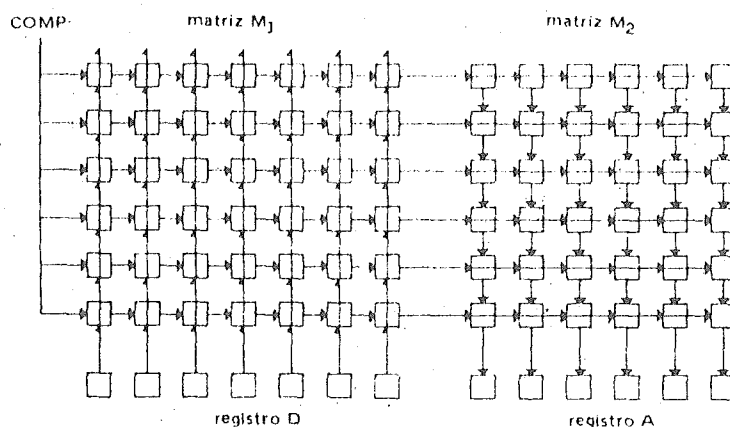
TOTAL  
 NUCLEOS  
 HILO RECUBIERTO  
 SEMICONDUCTORES

### I.4.1. Memorias asociativas y aplicaciones

Reproducimos del libro de Meinadier la muy esquemática presentación que éste hace de la memoria asociativa, también - llamada en la literatura C.A.M. (Content Addressable Memory). Como podrá comprobarse, el esquema de funcionamiento corresponde a unos circuitos especializados en la operación de lectura. Sólo la lectura es asociativa. La fase de escritura se realiza de la manera corriente, R.A.M. (Random Access Memory) que no representan las próximas figuras.

Se conciben estas memorias, también llamadas memorias direccionables por el contenido, no para buscar una información cuya dirección de almacenamiento se conoce, sino para investigar si una información, bautizada descriptor, está en la memoria y, en caso afirmativo, proporcionar una información asociada. Una memoria asociativa se divide en dos partes: una primera parte  $M_1$ , en que todas las palabras son comparadas en paralelo al descriptor  $D$ , y una segunda parte  $M_2$ , que entrega eventualmente a un registro la información asociada al descriptor  $D$ . La memoria asociativa se nos presenta, entonces, como la realización cableada de la estructura de informaciones llamada tabla.

A manera de ejemplo, vamos a mostrar cómo puede realizarse una memoria asociativa en lógica celular.

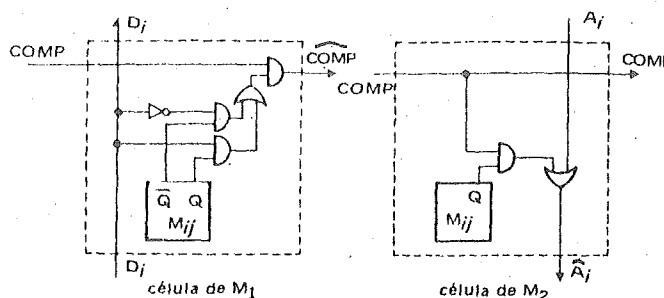


Las memorias  $M_1$  y  $M_2$  están organizadas por palabras, correspondiendo las células de una misma horizontal a los diferentes bits de una palabra y las células de una vertical a los bits de un peso determinado de las diferentes palabras. Cada célula representa un punto de memoria con un biestable y las lógicas de escritura y de comparación para la matriz  $M_1$ , de escritura y de lectura para la matriz  $M_2$ . Aquí nos desinteresaremos de las operaciones de escritura que permiten la carga inicial de la memoria.

La búsqueda en memoria implica la carga previa del descriptor en el registro D. A continuación, se ejecutarán todas las operaciones en paralelo con todas las palabras, bajo el efecto de un solo impulso COMP. Una célula de  $M_1$  transmitirá el impulso COMP si el contenido del punto de memoria es idéntico al contenido del biestable del registro D situado en la misma vertical, y lo bloqueará en caso contrario. Si se supone que hay, como máximo, una información idéntica al descriptor en  $M_1$ , se tendrá como máximo, un impulso COMP a la salida de  $M_1$  que designará a la palabra de  $M_2$  para enviar al registro A.

La próxima figura esquematiza una célula de  $M_1$  y una célula de  $M_2$ . Sus ecuaciones son las siguientes.

para  $M_1$        $\widehat{COMP} = COMP \cdot (D_i \cdot Q_{ij} + \overline{D_i} \cdot \overline{Q_{ij}})$   
 para  $M_2$        $\widehat{A}_i = A_i + (COMP \cdot Q_{ij})$   
 ( $i$  designa el índice de columna y  $j$  el índice de fila).

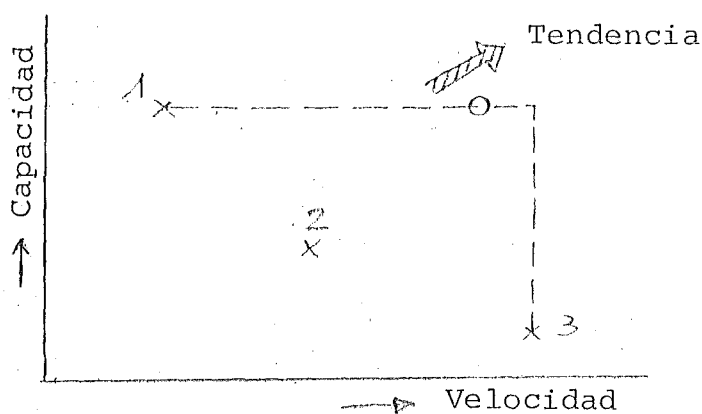


Pueden realizarse pequeñas memorias asociativas (16 palabras de

unos treinta dígitos) con tiempos de búsqueda del orden de 100 a 200 ns. Se utilizan las memorias asociativas para las búsquedas en tabla o para ciertas transcodificaciones, en las que el factor tiempo es demasiado importante como para perderlo en búsquedas programadas sobre tablas en memoria central.

#### I.4.1.1. Memoria "Cache"

Se inscribe dentro del concepto y de las técnicas de jerarquía de memorias. Este concepto, en síntesis, viene a considerar todas las unidades de memoria como componentes de un sistema con distintos niveles. El objetivo ideal del sistema es conseguir un conjunto que se comporte con el máximo de velocidad y el máximo de capacidad, a un mínimo coste por bit.



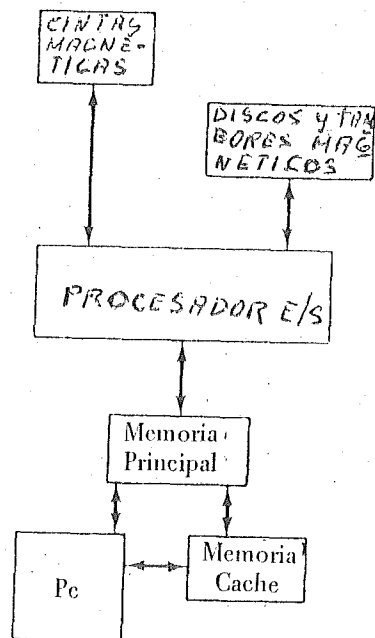
Visto en abstracto, supóngase que se cuenta con tres unidades de memoria, cuyas características de capacidad y velocidad vienen representadas por los puntos numerados 1, 2 y 3. Con ellas puede organizarse un sistema jerarquizado, cuyo comportamiento operativo en estas coordenadas equivaliera estadísticamente al circuitito representado en el mismo gráfico.

La memoria Cache y la memoria Virtual son ejemplos de Jerarquía de Memorias, utilizados en principio en grandes ordenadores, transferidos cada vez más a ordenadores de tipo medio y en un plazo más o menos corto a los pequeños ordenadores. De

hecho la técnica de Memoria Cache ya ha sido utilizada en algún miniordenador.

La memoria cache consiste en una memoria de pequeña capacidad de consulta asociativa y de gran velocidad (realizada con tecnología más rápida que la memoria principal). En esta memoria se cargan dinámicamente informaciones (instrucciones y datos) cuyas direcciones tienen la mayor probabilidad de ser generadas por el Procesador central. Ocurre así:

- a) El procesador central busca la dirección en la memoria asociativa. Si está allí, lee o escribe. Si no está, y se trata de una lectura, lee en memoria principal y aprovecha para cargar en memoria Cache un entorno de la información leída. Este entorno se carga en aquellas posiciones de la memoria cache que hace más tiempo no han sido utilizadas. Para ello hay un cableado, que mantiene actualizada una tabla de uso, mediante un algoritmo LRU (Least Recently Used).
- b) La operación de escribir lleva aparejado siempre la operación de escribir en la memoria principal. Existe comunicación entre Memoria cache y Memoria principal.



- c) Para que el sistema sea eficiente debe haber alguna forma de que la memoria principal pueda transferir varias unidades de información de golpe.

La probabilidad de que la información necesaria esté disponible en la memoria cache depende de:

- 1°. la capacidad de la memoria cache
- 2°. la adecuación del algoritmo de sustitución de informaciones.
- 3°. la dinámica del programa en ejecución.

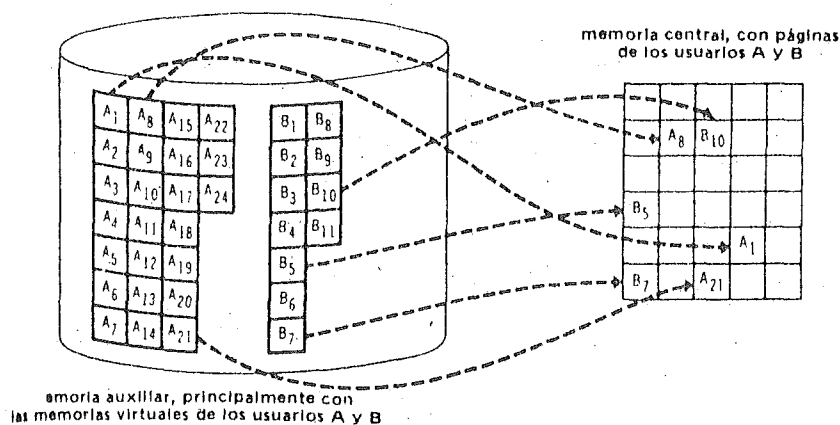
Un sistema cache típico consigue del 64 al 96% del rendimiento teórico que tendría una memoria con la velocidad de la cache y con la capacidad de la principal. Por fijar ideas, una memoria cache, la más conocida de todas, implementada en el ordenador IBM 360/85, tenía 80 ns frente a aproximadamente 1  $\mu$ s la memoria central.

#### I.4.1.2. Memoria Virtual

La memoria virtual es un compendio bastante complejo de técnicas de hardware y software que pretenden desembarazar al programador de las limitaciones asociadas a la capacidad física de la memoria central. Para ello, sustituye el espacio real por un espacio lógico, llamado espacio de direccionamiento, que es considerablemente mayor que el espacio real. El trabajo del sistema virtual se reduce a transformar continuamente el espacio de direccionamiento (virtual) en el espacio real de la memoria central, y viceversa.

Desde el punto de vista de hardware se cuenta con dos unidades de memoria, un disco y la memoria central. A esto hay que añadir los circuitos adecuados para manejar la transformación antes citada.



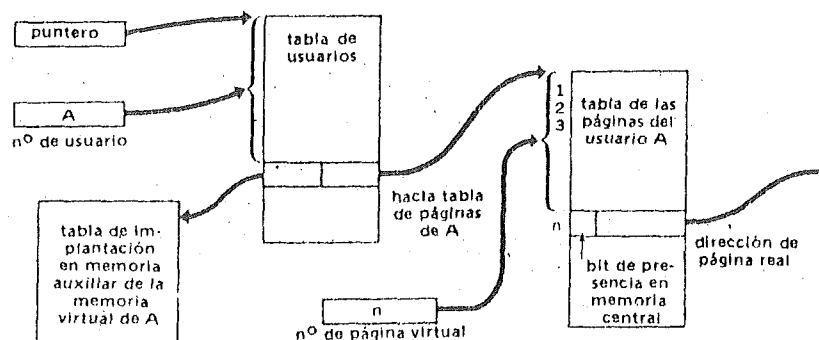


Dinámicamente, la situación que se produce es la que se representa en la figura (Meinadier).

Las memorias virtuales de los distintos usuarios, las memorias auxiliares y la memoria central son divididas en páginas de idéntica longitud (en general de 128, 256, 512 ó 1.024 palabras). Se escribe cada programa en relación con su propia memoria virtual, es decir, que todas las direcciones que contiene son relativas a su primera instrucción. Las llamaremos direcciones virtuales y, en principio, se componen de: un número de página dentro de la memoria virtual y una dirección de palabra dentro de la página. Los programas susceptibles de ser activados son implantados en la memoria auxiliar. En un determinado instante, sólo un reducido número de páginas de cada programa está presente en la memoria central.

El sistema operativo se ocupa de mantener actualizada, mediante una tabla, la correspondencia entre las páginas en memoria auxiliar y las páginas en memoria central. Dedicamos a cada usuario una tabla de páginas virtuales, indicando si la página ha sido cargada en memoria real y, si así es, especificando la dirección de la primera palabra de la correspondiente página real. En el curso de la ejecución de un programa, la transformación de la dirección virtual en dirección real implicaría

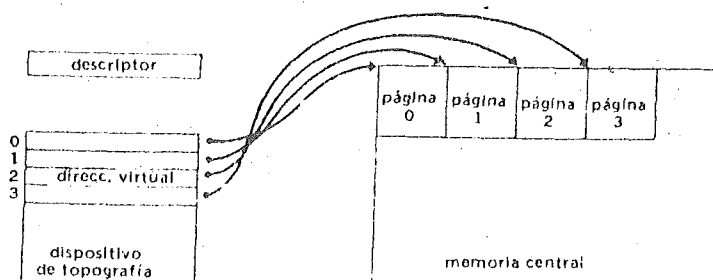
una búsqueda dentro de estas tablas, por cada acceso a memoria. Se evita tal búsqueda en virtud del dispositivo de topografía de memoria.



Tablas del sistema de explotación de un sistema paginado

El dispositivo de topografía de la memoria central. -

Es un conjunto de registros montados como memoria asociativa. En nuestro modelo, cada registro está vinculado a una página de la memoria central y contiene dos informaciones que definen perfectamente su afectación: el número de programa dado por el sistema operativo, o, si se prefiere, el número de memoria virtual, y el número de página virtual en esta memoria.



Dispositivos de topografía de memoria

El dispositivo de topografía de memoria permite el direccionamiento en la memoria real, sabiendo que las direcciones contenidas en las instrucciones se refieren a la memoria virtual del programa en curso. Las informaciones básicas para obtener la dirección son:

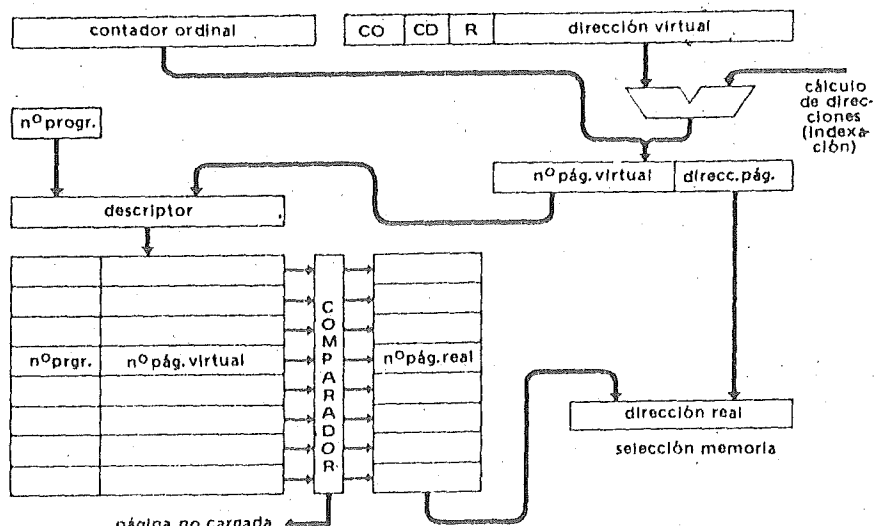
- 1) el número de programa, contenido en un registro de

la unidad central, cargado por el sistema operativo antes de ceder el control al programa.

- 2) la dirección virtual, conocida del programador. La dividimos lógicamente en dos partes: la dirección dentro de la página (pesos inferiores) y el número de página virtual (pesos superiores).

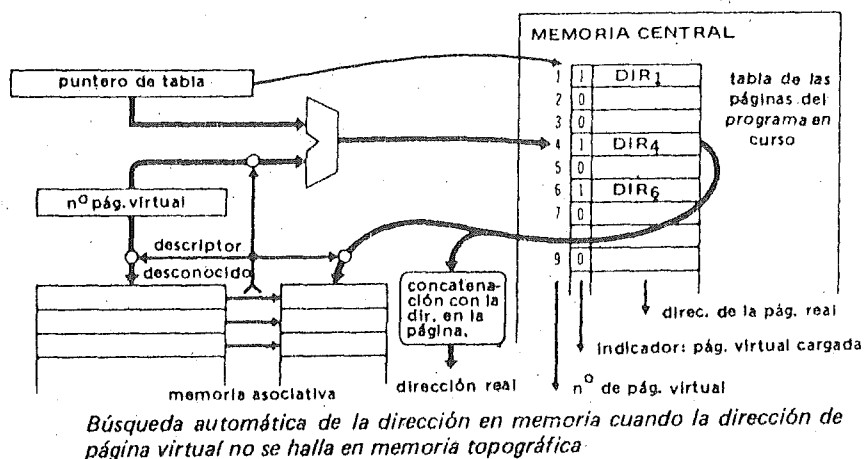
La yuxtaposición del número de programa y del número de página virtual proporciona un descriptor, que es comparado en paralelo con los contenidos de los registros asociativos.

Si se produce una coincidencia es que la página virtual referenciada está presente en memoria central y el número de registro asociativo da el número de la página real correspondiente. Si, por el contrario, no hubiera coincidencia entre el descriptor y el contenido de uno de los registros asociativos, es que la página de referencia no ha sido cargada en la memoria central. En este último caso, se producirá una desviación hacia el sistema operativo, que cargará la página y actualizará el registro asociativo pertinente.



Principio de utilización de un dispositivo de topografía de memoria para el direccionamiento en un contexto paginado

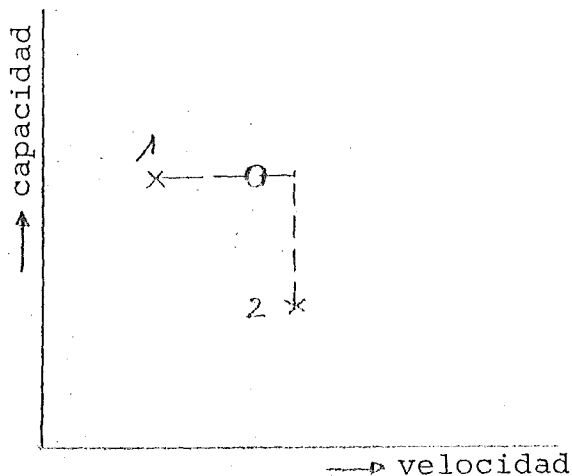
En la práctica no es posible normalmente contar con una memoria asociativa que recoja la topografía completa de la memoria central. Se dispone entonces de un sistema incompleto que, lógicamente, es menos rápido.



De manera similar a la memoria cache, se emplean ciertos algoritmos para decidir qué páginas de memoria van a ser sustituidas.

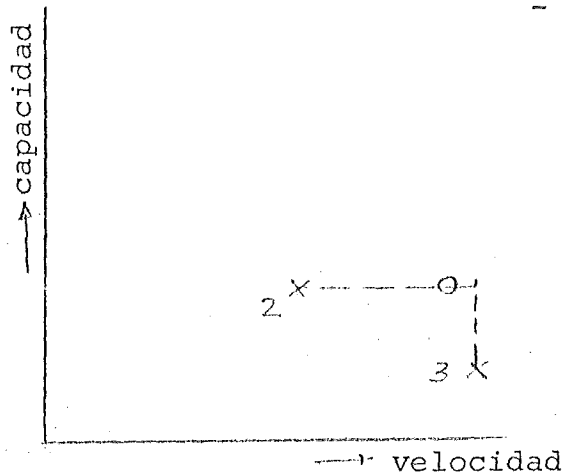
Si recogemos la descripción realizada y la comparamos con la memoria cache, podremos establecer diferencias de objetivos que, a simple vista, no resaltan fácilmente. La organización con la memoria cache es una jerarquía de memorias cuyo objetivo particular es acelerar la velocidad de la memoria central sin modificar el trabajo del programador.

La organización con la memoria virtual es una jerarquía de memorias cuyo objetivo particular es ampliar la capacidad de la memoria central sin modificar el trabajo del programador.



MEMORIA VIRTUAL

- 1. Disco
- 2. M. central
- 0. Resultante



MEMORIA CACHE

- 2. M. central
- 3. M. asociativa (cache, buffer).
- 0. Resultante.

#### I.4.2. Microprogramación

La microprogramación es básicamente una técnica de diseño de unidades de control, a cualquier nivel. Si se introduce en el apartado de memorias es porque su materialización práctica es sobre soporte de memoria, y en una mayoría de casos, sobre memoria ROM.

Como el asistente a la conferencia ya conoce este tipo de memoria y su utilización en el terreno de los microprocesadores, sólo haremos aquí unas reflexiones acerca de la incidencia de la microprogramación en el diseño de las unidades secuenciadoras de microórdenes.

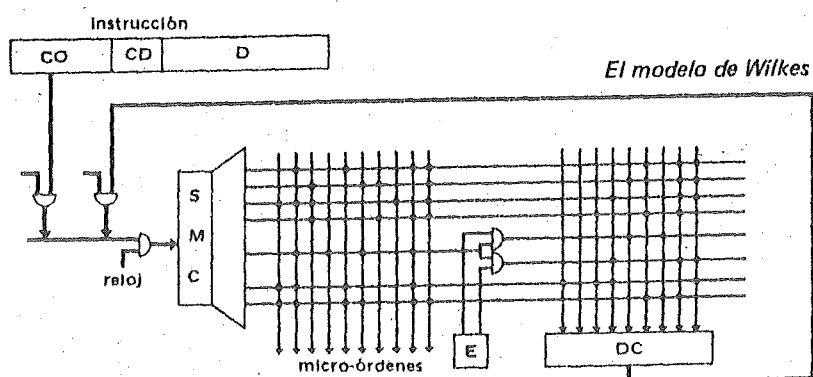
Al secuenciador llegan informaciones, directamente procedentes de los campos de condiciones de direccionamiento del registro de instrucción o a través del decodificador de instrucciones, del distribuidor de fases, del reloj interno y de ciertos biestables de estado, y con ellas genera automáticamente secuencias de señales, llamadas microórdenes, que se desparraman

y actúan sobre cada uno de los elementos de la ruta de datos.

El secuenciador, cuyas ecuaciones lógicas se calculan, es un autómeta de múltiples salidas, normalmente una sola entrada activa y un conjunto de estados representativos de la circunstancia real de la máquina. Produce un microprograma.

microprograma = secuencia de microórdenes/instrucción  
microorden = f (instrucción, estado).

La microprogramación consiste en reemplazar el secuenciador cableado por un secuenciador programado, residente en una memoria especial, llamada memoria de control, que normalmente es accesible sólo en lectura (R.O.M.). Es ésta la aplicación más corriente de la técnica de microprogramación y fue propuesta por Wilkes en el año 1.951, en un modelo cuya finalidad principal era la de sistematizar y simplificar los procedimientos de diseño de la unidad central de control.



El código de operación de una instrucción cualquiera es interpretado como la dirección de una posición precisa de la memoria de control, que contiene una microinstrucción. Se llama microinstrucción porque está registrada en una memoria y porque forma parte de un conjunto asociado a la particular instrucción mencionada, conjunto denominado microprograma. Una mi

croinstrucción consta de un código, que da directamente las microórdenes a distribuir al conjunto del ordenador, y de los elementos de información necesarios al desarrollo del resto de las microinstrucciones del mismo microprograma (normalmente la dirección de la próxima microinstrucción).

"La microprogramación es análoga a la programación convencional. El programador le dice al sistema o al dispositivo lo que tiene que hacer por medio de instrucciones en la memoria central. El microprogramador le dice cómo lo tiene que hacer, controlando qué elementos lógicos y de memoria deben usarse y cómo para cada operación. Así, la instrucción de máquina, que para el programador representaba el nivel más profundo de comunicación con el sistema, puede contemplarse ahora como una subrutina cerrada desglosable en secuencia de funciones más elementales llamadas microinstrucciones" (Husson, 1.970).

Es evidente que la microprogramación plantea de forma llamativa lo convencional de la línea divisoria entre los dominios del hardware y del software. Pero esta técnica está contribuyendo, además, a la gestación de un enfoque distinto en la concepción de los ordenadores. La concepción de los ordenadores se ha iniciado siempre de abajo arriba, o sea, el circuito se ha hecho algoritmo, por expresarlo de manera metafórica. En términos generales, se ha ido del hardware al software. Sin embargo, la verdadera esencia de los ordenadores es el software, nivel todavía sin formular. Por ello, es de esperar que la concepción futura se llevará de arriba abajo, o sea, el algoritmo se hará circuito. Esta es nuestra interpretación personal de los esfuerzos emprendidos para formular lenguajes y compiladores de hardware. En este mismo sentido, la microprogramación pone de manifiesto la auténtica naturaleza algorítmica de las operaciones de control, más parecida en su estructura a la programación que al tradicional enfoque tomado en préstamo de los dos subniveles inferiores de diseño lógico.

Las aplicaciones actuales y las que se prevén para un futuro inmediato son muchas y variadas. Husson las clasifica en las siguientes categorías:

1. Diseño de un control eficaz para una arquitectura bien especificada de un sistema. Se basa en que ya no es cierto, que, microscópicamente, cada instrucción sea un estímulo que provoque necesariamente en el ordenador siempre la misma respuesta, ya que, de un diseño a otro -dentro de la misma arquitectura- es posible cambiar fácilmente el microprograma.
2. Modificación de una arquitectura existente, con o sin modificaciones del flujo de datos existente. (se refiere a los conceptos de repertorio de instrucciones a la medida o de extensibilidad del mismo a gusto del consumidor. El ordenador universal es un hecho, pero una arquitectura universal no es una arquitectura óptima para muchas clases de aplicaciones).
3. Emulación de algunas estructuras de hardware. (Se refiere a dotar a algún sistema de la capacidad de interpretar y ejecutar otras arquitecturas o lenguajes de alto nivel).
4. Integración del hardware y de diferentes niveles de software. (Se refiere al firmware).
5. Aplicaciones especializadas.

Según Dolotta y otros autores, que han hecho un estudio prospectivo del Procesamiento de Datos para el quinquenio de 1.980-85, el trinomio hardware/firmware/software evolucionará así:



"Existe una tendencia clara hacia un uso progresivamente creciente del firmware (microprogramación) para instrumentar funciones que antes eran cableadas o, en ciertos casos, realizadas por software. Así la complejidad de la lógica cableada es aún creciente, pero no tan deprisa como la del lenguaje de máquina, que es contemplado por el software; éste está haciéndose más y más sofisticado y de mayor nivel. Se espera que cada vez más funciones implementadas por software hasta ahora lo sean en adelante por microcódigo (p. ej. partes de paginación, acceso de datos, mecanismos de interrupción, intérpretes de lenguajes de alto nivel, etc.).

Debemos subrayar sin embargo, que no hay magia en el firmware. Participa de muchos de los problemas del hard y del soft, aunque tenga muchas y variadas ventajas propias: es más fácil de modificar (pero es más lento) que el hard, y más rápido (pero más difícil de modificar) que el soft. Añade flexibilidad a la tarea del diseño y por ello se ha convertido en otra herramienta para el diseñador de computadores. Programas implementados por firmware no son inherentemente más fiables que sus contrapartidas en soft. Especialmente, el advenimiento del firmware "escribible" ha eliminado parte del incentivo de hacer programas muy bien depurados.

"Pensamos que se usará más cada vez el firmware pero que, en la mayoría de los casos, los usuarios tendrán poco -si es que tienen alguno-, acceso a él a causa de los fuertes inconvenientes derivados de los problemas de fiabilidad y mantenimiento asociados a dicho acceso".

#### I.5. Propositiones de Sáez (1.973) sobre condiciones de viabilidad de innovaciones en técnica o tecnología.

1<sup>a</sup>. Toda innovación en estructura es posible gracias

a un avance tecnológico que, puesto al servicio de una idea o de una necesidad, se justifica técnicamente si la mejora producida en algún factor de rendimiento es muy superior relativamente al aumento de coste del sistema.

2<sup>a</sup>. Una técnica o una tecnología es, o debe ser, introducida a nivel industrial cuando viene a cubrir una necesidad o produce sustanciales mejoras de rendimiento, siempre que no atente contra las inversiones\* de la comunidad de constructores-usuarios.

Aplicar en discusión a los siguientes temas:

- Memoria Cache.
- Topografía Memoria, en Memoria Virtual
- Memoria película delgada NDRO
- Microprogramación
- Minicomputadores
- Microprocesadores y microordenadores

#### I.6. Arquitecturas de ordenador

Se experimenta hoy día una tendencia, cada vez más acu

---

\* Por "inversión" debe entenderse no sólo la puramente financiera, sino la inversión en trabajo, en formación, en ideas, en molestias, en tiempo, en modificaciones de organización, etc. y - al decir "atentar contra" no se quiere significar inmovilismo - sino dosificación de los cambios para que éstos sean absorbidos por un sistema cuya estructura y relaciones tienden a un equilibrio dinámico, con un crecimiento sostenido.

sada, hacia una polarización hacia el uso de grandes y pequeños sistemas en el mundo de la informática. Esto es cierto por lo que se refiere a ordenadores convencionales. Pero también lo es en lo concerniente a ordenadores no convencionales. De estos podemos distinguir a los que tienen características peculiares, como los sistemas multiprocesadores y paralelos, como la gama de minis y micros que, en lo referente a su estructura han sido tratados en conferencias anteriores, y como la clase de sistemas especiales cuya estructura no refleja los esquemas de Von Neumann.

Por ser menos conocidos, recogeremos con alguna mayor extensión, aunque un tanto superficialmente, las arquitecturas de los sistemas multiprocesadores y paralelos, siguiendo el libro de Enslow.

#### I.6.1. Sistemas multiprocesadores y paralelos

La tabla siguiente lista una cronología resumida de estos sistemas. (Ver pág. 33 y siguientes).

La definición que da Enslow de un sistema multiprocesador es ésta; un sistema con:

Dos o más unidades de procesamiento

Memoria compartida.

E/S compartida.

Sistema Operativo único integrado.

Interacción del hardware y del software a todos los niveles.

Es de destacar que gran parte de estos sistemas han sido diseñados para aplicaciones de tipo más bien poco corriente en los terrenos nuclear, meteorológico, espacial, de teleco

Date	System Manufacturer and Model Number	Remarks
1958	National Bureau of Standards PILOT	Three independently operating computers that could work in cooperation.
1958 (circa)	IBM, AN/FSQ-31 and 32	Solid-state SAGE computer; not multiprocessors; merely duplexed systems.
1960	Ramo Wooldridge TRW-400	"Polymorphic system"; for USAF command and control. Some construction done, not completed. Important for early concepts.
May 1960	UNIVAC LARC	One I/O processor and one computational processor capable of operating in parallel. One delivered to Livermore AEC Laboratories. Not a "true" multiprocessor.
May 1961	IBM Stretch(7030)	Original design called for separate character-oriented processor and binary arithmetic processor. These were dropped from final design; therefore, final product was not a multiprocessor. It did contain look-ahead. Only seven delivered.
Nov. 1962	Burroughs, D-825 (This system carried various military model designations depending on the major system of which it was part.)	First modular system with identical processors. Total memory shared by all processors. Up to four processors, 16 memory modules, 10 I/O controllers, and 64 devices. Important feature was one of the earliest examples of a modern operating system—the Automatic Operating and Scheduling Program (ASOP).
Feb. 1963	Burroughs B-5000	One or two processors. Up to eight memory modules. Programs independent of addresses. Supervisor was the Master Control Program (MCP). Utilized virtual memory concepts and hardware. Machine code based on Polish notation. Users programmed only in ALGO or COBOL. Became the B-5500 in Nov. 1964.

Date	System Manufacturer and Model Number	Remarks
1963	IBM 704X/709X (7040 or 44 and 7090 or 94)	"Direct Coupled System"
1963	Bendix G-21 (later CDC)	A multiprocessor version of the G-20 developed for Carnegie Institute of Technology. A crossbar system.
Sep. 1964	CDC 6600	Contained multiple arithmetic and logic units each of which can execute only a small fraction of the total instruction repertoire. Ten peripheral processors were an integral part of the system. (Number of PPU's increased to 20 in 1969.) The PPU's do constitute a multiprocessor system. Overall system an example of an asymmetric multiprocessor.
Nov. 1964	Burroughs B-5500	An upgrade of the B-5000 (see Feb. 1963).
1964	GE 645 (now Honeywell)	Ordered by Project MAC at MIT.
May 1965	GE 645 (now HIS-645)	Delivered to Project MAC at MIT. Hardware not a standard product; however MULTICS operating system is being released.
Dec. 1965 1965	UNIVAC 1108 SOLOMON I	Design only. First large-scale array processor.
Mar. 1966	IBM S/360 Model 67	Special dual-processor time-sharing system.
Apr. 1966	CDC 6500	Dual 6400's
Dec. 1966	XDS Sigma 7 SOLOMON II	Design only.
Jun. 1967	CDC 6700	Dual CDC 6600's
Aug. 1967 1968	XDS Sigma 5 CDC 7600	Very similar to 6600, but higher speed and included hierarchy of main memory as standard feature.
Apr. 1969	IBM S/360 Model 65 MP	Dual-processor version of standard model 65. A true multiprocessor.
Jun. 1970	XDS Sigma 6	

Date	System Manufacturer and Model Number	Remarks
Oct. 1970	Burroughs B-5700	Similar to B-5500 with capability for increased memory. Capability for four B-5700 systems to share disk storage.
Feb. 1971	Honeywell 6050, 6060, 6080	
Jun. 1971	Burroughs B-6700	
Sep. 1971	Digital Equipment System 10/1055, 10/1077	
Sep. 1971	XDS Sigma 8, 9	
Nov. 1971	UNIVAC 1110	
1971	SDC, PEPE (Parallel Element Processing Ensemble)	Prototype for processing of radar data for ballistic missile defense system.
Jan. 1972	Honeywell 2088	
Sep. 1972	ILLIAC IV	Array processor. 64 processor elements. Driven by a conventional multiprocessor used as a front-end control processor.
Feb. 1972	Burroughs B-7700	
1972	CDC, Cyber 72, 73, 74, 76	
1972	Goodyear STARAN	Parallel associative system.
1972	Texas Instruments ASC (Advanced Scientific Computer)	Embodies both multiprocessing and pipelining.
1973	CDC STAR-100	Pipeline system.
1974	IBM S/370, Models 158 MP & 168 MP	Shared real and virtual storage

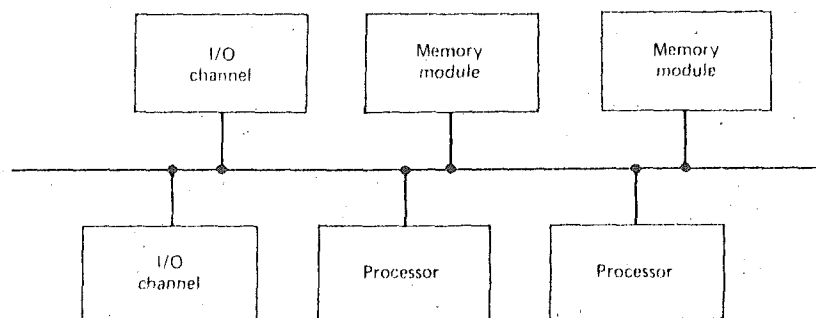
municación, etc.

Se clasifican en siete clases:

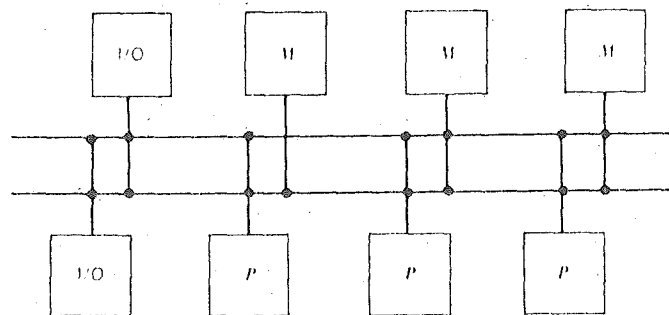
- . De bus común o en tiempo compartido.
- . Crossbar.
- . De bus múltiple/multipuerta
- . Asimétricos o no-homogéneos
- . Procesador de vectores u ordenaciones
- . Procesador pipe-line

. Sistemas antifallo.

La próxima figura presenta un esquema funcional de la primera clase, en donde todos los módulos se conectan en paralelo al bus. El bus puede ser bidireccional o estar dividido en dos buses unidireccionales con un modificador de bits. Es



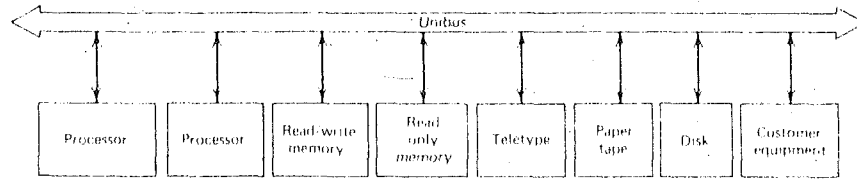
posible contar con más de un bus bidireccional, como muestra la siguiente figura.



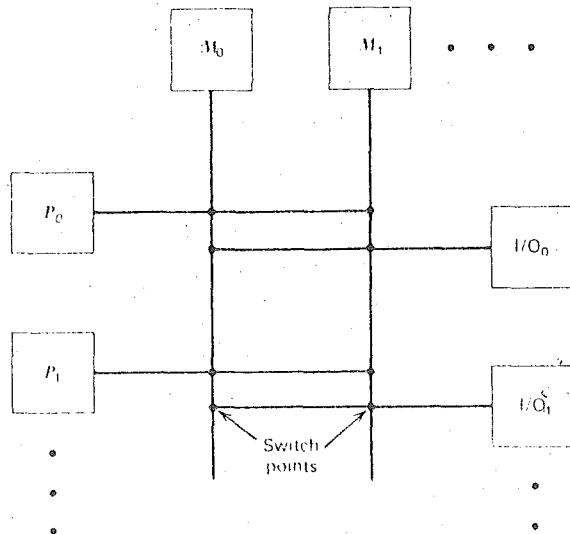
La ventaja de estos sistemas es la simplicidad de su control, que se paga con su bajo grado de paralelismo (prácticamente sólo una transferencia a la vez), sus posiblemente largos tiempos de respuesta, la pequeña fiabilidad en función de un fallo en el bus.

Un ejemplo clásico y bien conocido es el PDP-11 con su

sistema Unibus.



En las estructuras tipo conmutación crossbar cualquier módulo de memoria puede conectarse a cualquier procesador o unidad de E/S. Se establece una conexión real permanente entre dos unidades por toda la duración de la transferencia, de manera - que se sigue el método de división en el espacio.

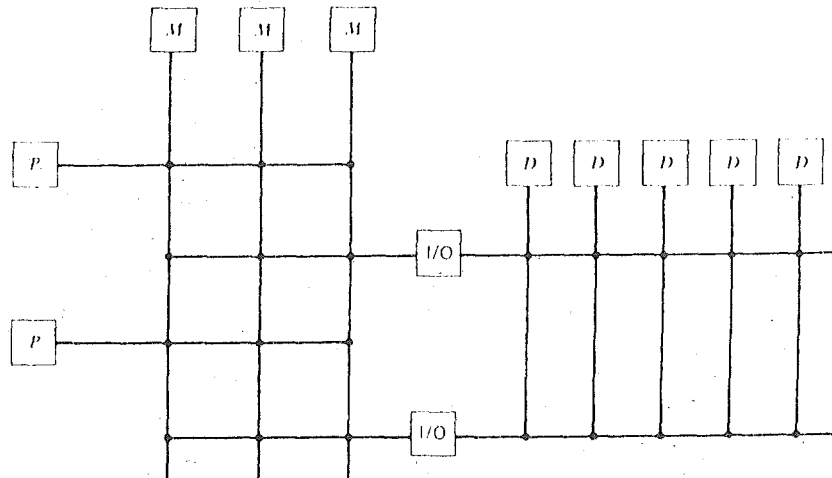


La matriz crossbar o matriz de conmutación puede llegar a ser muy complicada. El H4400 (ocho procesadores centrales o de E/S y 16 módulos de memoria) contiene circuitos y componentes equivalentes a 2,5 procesadores.

Es posible utilizar una matriz de conmutación separada para controlar la conexión de las unidades periféricas con las

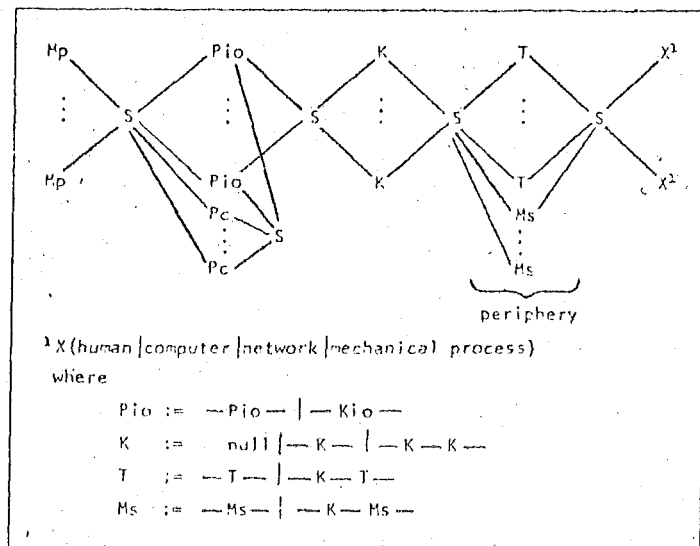


unidades de E/S.



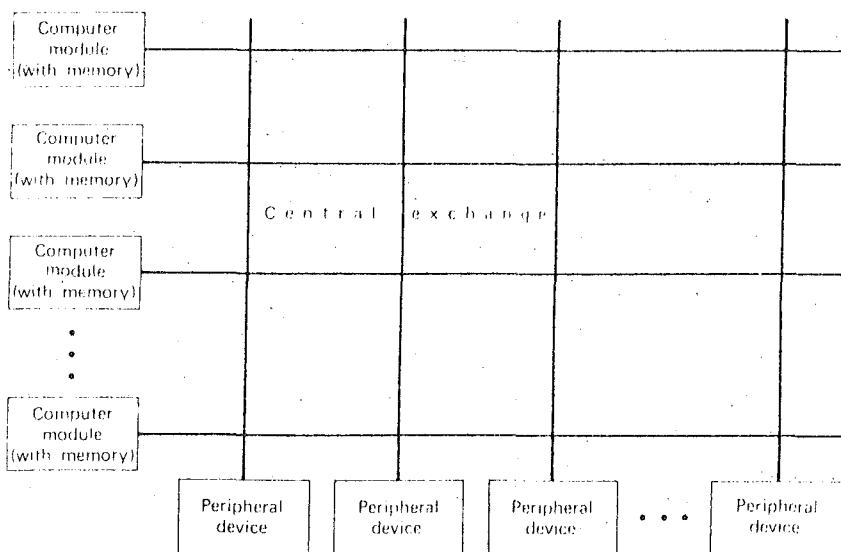
Crossbar switch system organization with separate I/O crossbar switch matrix.

Según la nomenclatura de Bell/Newell ésta es una estructura en celosía, representada así en su ya citado libro



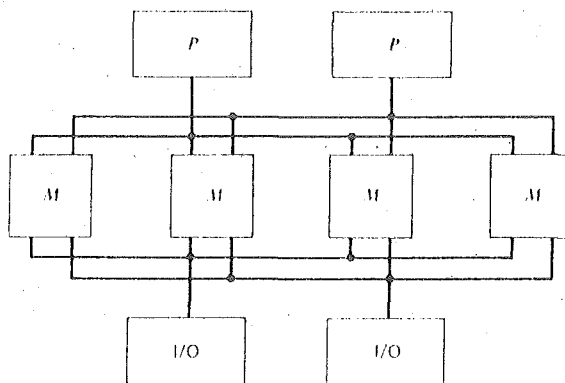
Uno de los primeros multiprocesadores, importante por los conceptos estructurales que incorporó, fué el TRW-400 o computador polimórfico, que posee una sola matriz de conmutación a

los cuales se conectan computadores completos y dispositivos periféricos.



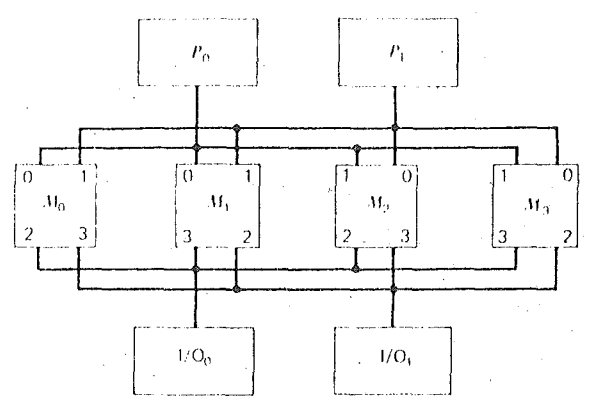
Ramo-Wooldridge, RW-400 system—the "Polymorphic Computer."

Los sistemas multibus/multipuerta proporcionan también más de una vía de transferencia simultánea, aunque, para ello es esencial que los módulos de memoria tengan más de una puerta de acceso y los circuitos adecuados para resolver los conflictos que ocurren si dos o más procesadores o unidades de E/S solicitan acceso.



Multiport-memory/multibus system organization, basic form

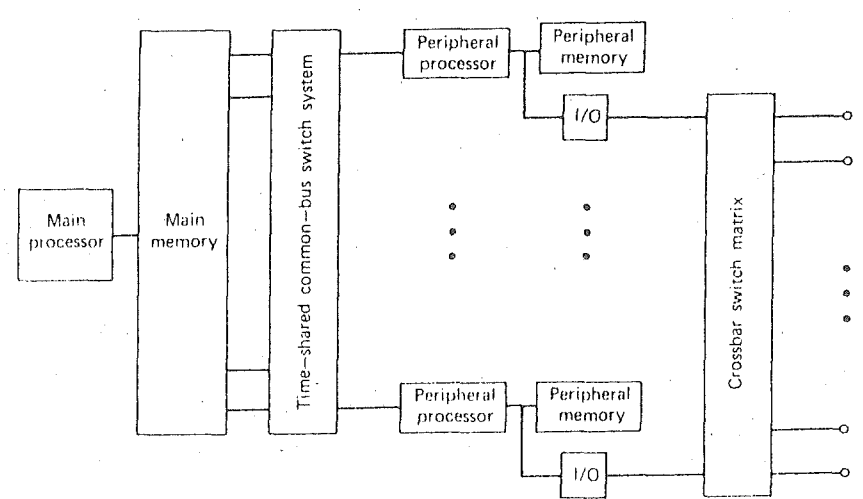
Lo que suele hacerse es asociar una prioridad a cada puerta y punto de conexión, así en caso de conflicto se da preferencia de acceso, por ejemplo, al procesador asociado a su módulo primario de memoria.



Assignment of memory access priorities—multiport system organization.

Ejemplos de estructuras de esta naturaleza los tenemos en las series Univac 1100 y Honeywell 600 ó 6000, en el CDC - - 3600, XDS- Sigma 7 e IBM 360 modelo 67.

La figura siguiente, que esquematiza al CDC 6600, es un ejemplo de sistema asimétrico, que es aquel que tiene componentes de un mismo tipo muy diferenciados: procesadores, módulos de memoria, etc.



Example of asymmetrical/nonhomogeneous system organization—CDC 6600.

Este sistema concreto cuenta con procesadores periféricos (PP) cuya memoria llega hasta 4 K con palabras de 12 bits. Se ocupan de las tareas de contabilidad, control de jobs y operaciones de E/S. Uno de los PP's se ocupa de la consola principal del sistema y otro actúa de monitor de todo el conjunto - asignando tareas, tanto al procesador principal como a los demás PP's.

La técnica de los ordenadores pipe-line se basa en la repetición de la misma operación sobre varios items de una sola corriente de datos. El ejemplo típico es el de la suma de dos números representados en coma flotante. Esta es una operación que puede realizarse en cinco etapas:

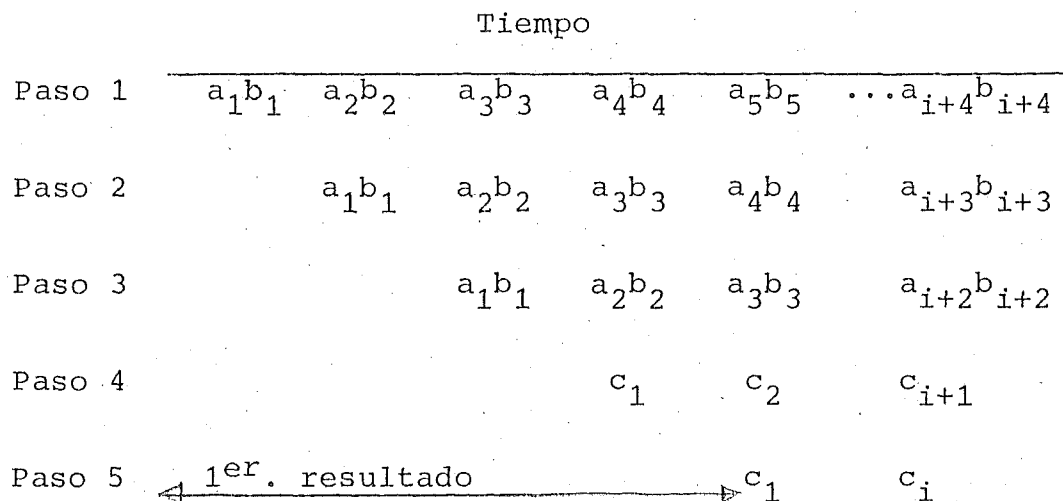
1. Normalización de los números y ajuste de sus exponentes.
2. Comparación de exponentes.
3. Desplazamiento de la característica del número de menor exponente para alinearlo con el de mayor exponente.
4. Suma de las características
5. Normalización de resultados.

Supongamos que los tiempos para realizar cada una de estas etapas son:

1.	100 ns	}	480 ns
2.	60 ns		
3.	100 ns		
4.	120 ns		
5.	100 ns		

En una máquina pipe-line cada paso será ejecutado por unos circuitos propios y el resultado transferido a los siguientes circuitos, insertándose nuevos números en los circuitos que

han culminado su trabajo



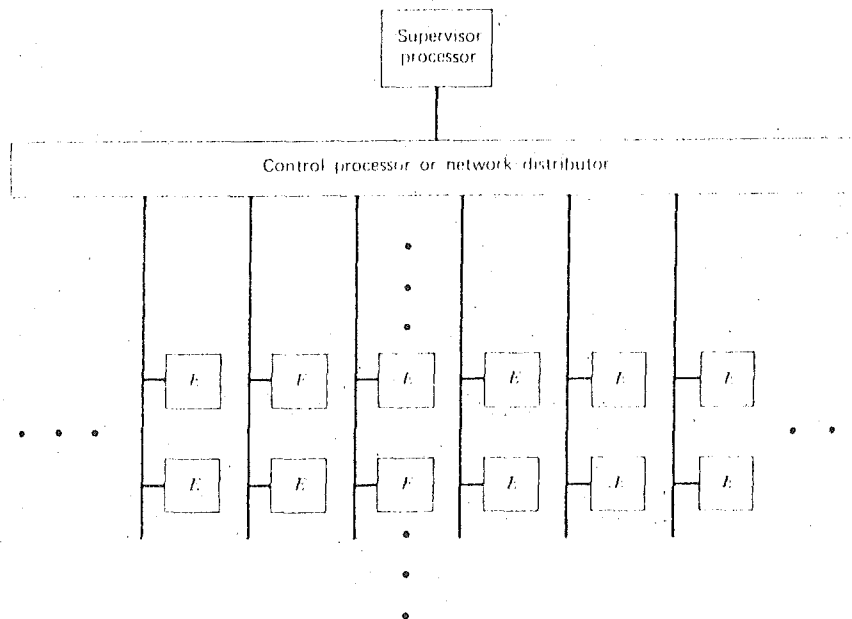
Supongamos que se ajusta el reloj para sincronizar los pasos a 120 ns para dar tiempo al paso 4. Entonces, el primer resultado se tiene a los  $5 \times 120 = 600$  ns, en lugar de a los 480 ns. Pero cada nuevo resultado se tiene a los 120 ns del anterior.

Máquinas que encajen dentro de esta categoría pueden citarse a IBM 360/91, CDC 7600, CDC STAR-100 y Texas Instruments Advanced Scientific Computer (ASC).

La característica más notable de los ordenadores de procesamiento de vectores es que la misma operación ha de ser ejecutada sobre una amplia colección de elementos relacionados de alguna manera y que a menudo es posible que esta operación sea ejecutada sobre todos los elementos a la vez. Vease la configuración básica de uno de estos sistemas. (Pág. 43).

Una configuración así corresponde a un sistema paralelo no multiprocesador, ya que los elementos de procesamiento E no son capaces de operar independientemente.

El ejemplar más conocido de esta categoría es el ILLIAC IV sucesor de los sistemas SOLOMON. Consta de 256 elementos de procesamiento (PE) organizados en cuatro cuadrantes de 64 (orde



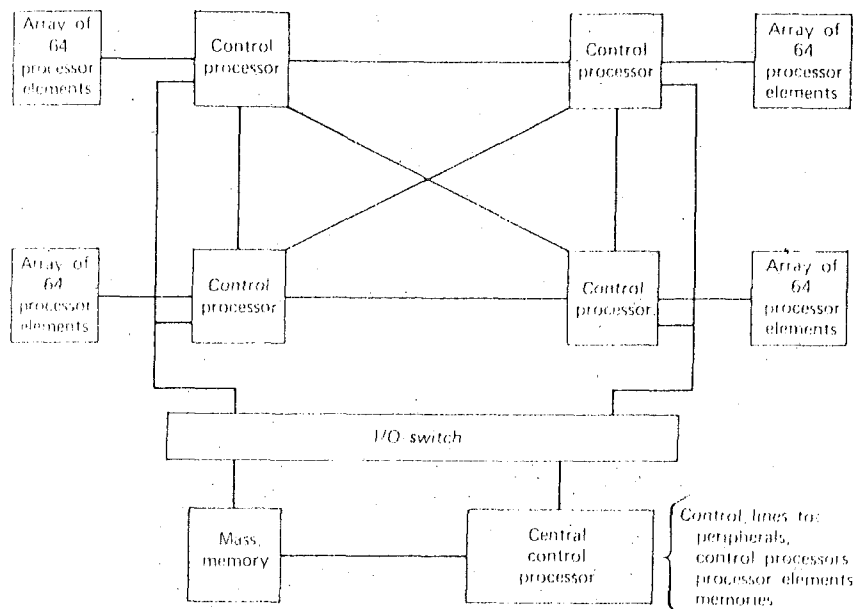
Array processor—basic configuration. *Noté.* Each *E* is an execution or processing element which has most of the capabilities of a stand-alone CPU plus its own private memory

nación 8 x8) con una unidad de control para cada cuadrante. Cada PE tiene una importante memoria propia, acceso directo a un conmutador de E/S y comunicación por cuatro vías con sus PE vecinos. El ordenador de control es un Burroughs B6700, que ejecuta programas ejecutivos y compilación.

El ILLIAC IV puede invertir una matriz de 700 x 700, - adecuadamente preparada, en 1 segundo.

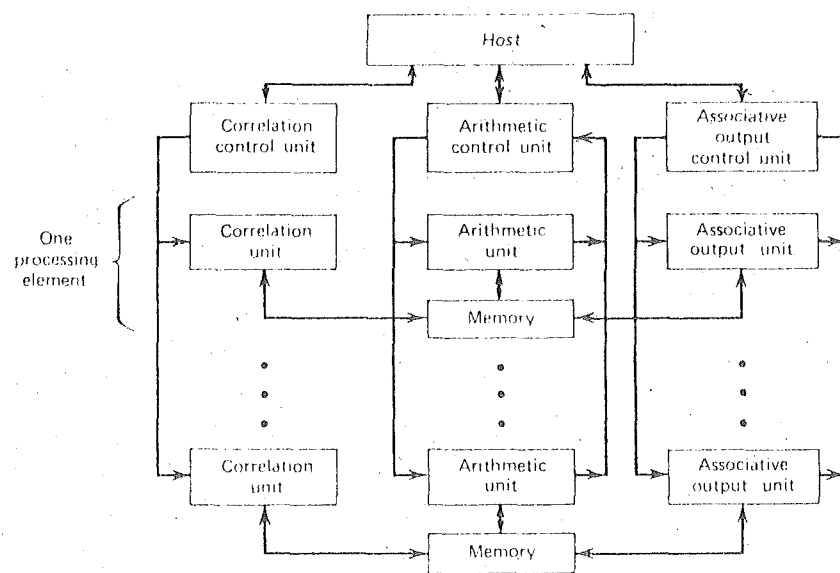
El mayor problema con este tipo de máquina es la preparación de programas que puedan explotar sus capacidades y el - desarrollo de traductores que puedan analizar automáticamente - programas y producir código para su ejecución en paralelo.

Para terminar con esta categoría citemos el sistema PE PE, (Parallel Ensemble of Processing Elements) desarrollado por entidades tan importantes como los Bell Telephone Laboratories,



Organization of the complete (four-quadrant) ILLIAC IV.

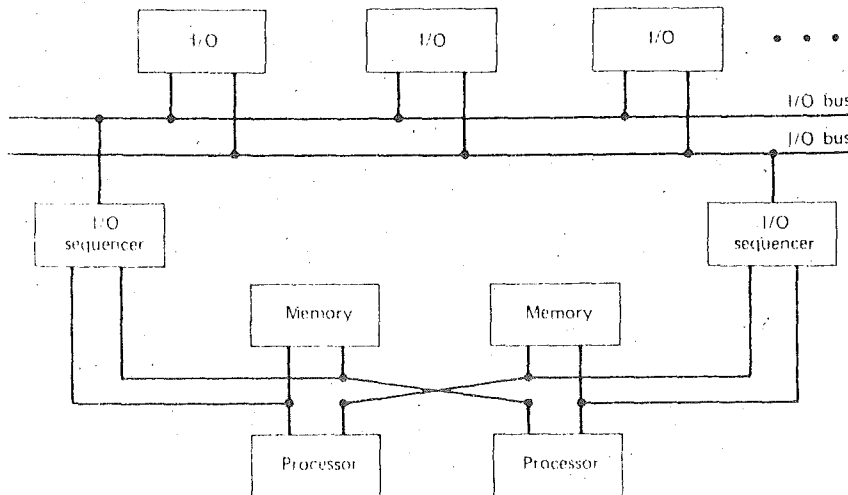
la System Development Corporation y Honeywell, y destinado al procesamiento de informaciones radar en un sistema de defensa con misiles de la U.S.Army.



PEPE system organization.

Puede uno hacerse idea del orden de magnitud del sistema cuando se sabe que el control del mismo recae sobre un ordenador (host) CDC 7600.

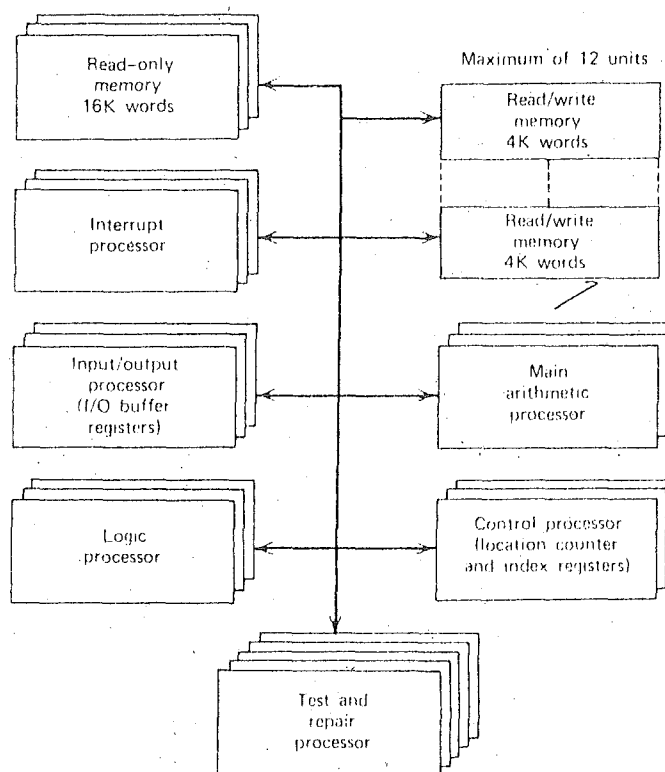
Los sistemas antifallo se llaman así porque están especialmente diseñados y organizados como sistemas para detectar, localizar y minimizar los fallos y sus efectos. Por ejemplo, en el sistema MIT/IL SIRU sus dos procesadores trabajan sobre el mismo problema, ejecutando la misma corriente de instrucciones y la misma corriente de datos síncronamente. Sólo un procesador está "activo" en el sentido de generar órdenes de "escritura". Ambas memorias poseen idéntico contenido, ya que responden a las órdenes de escritura; pero solamente una responde a las peticiones de lectura. El sistema de interconexión está diseñado para permitir un chequeo continuo de la certeza de los distin-



Fault tolerant design--MIT/IL SIRU computer.

tos módulos e inmediata reconfiguración. El sistema JPL STAR - (Self-Testing and Repair) conmuta un módulo en fallo por otro, pero no es un multiprocesador. Su finalidad es lógica y redundante.





JPL STAR system organization [Weitzman].

### I.6.2. Sistemas minis y micros

Los miniordenadores surgen como consecuencia de varios factores que coinciden en un momento dado. En primer lugar, el impulso arrebatador de la tecnología electrónica. Se juntan a esto la necesidad de atacar aplicaciones especializadas, para las que los sistemas que pudieramos llamar convencionales eran demasiado caros o sobreabundantes, y la inercia natural de lo que hemos llamado comunidad de constructores-usuarios.

Todo esto explica que los grandes constructores de ordenadores ha sido pionero en la rama de los minis. Estos han sido, además un campo de experimentación de tecnología (R.O.M., memoria central con Circuitos Integrados, circuitos LSI en el

procesador, etc...) y la puerta de entrada de muchas industrias de electrónica en el mercado de ordenadores.

I.6.2.1. Estructura-robot y aplicaciones de los minis.

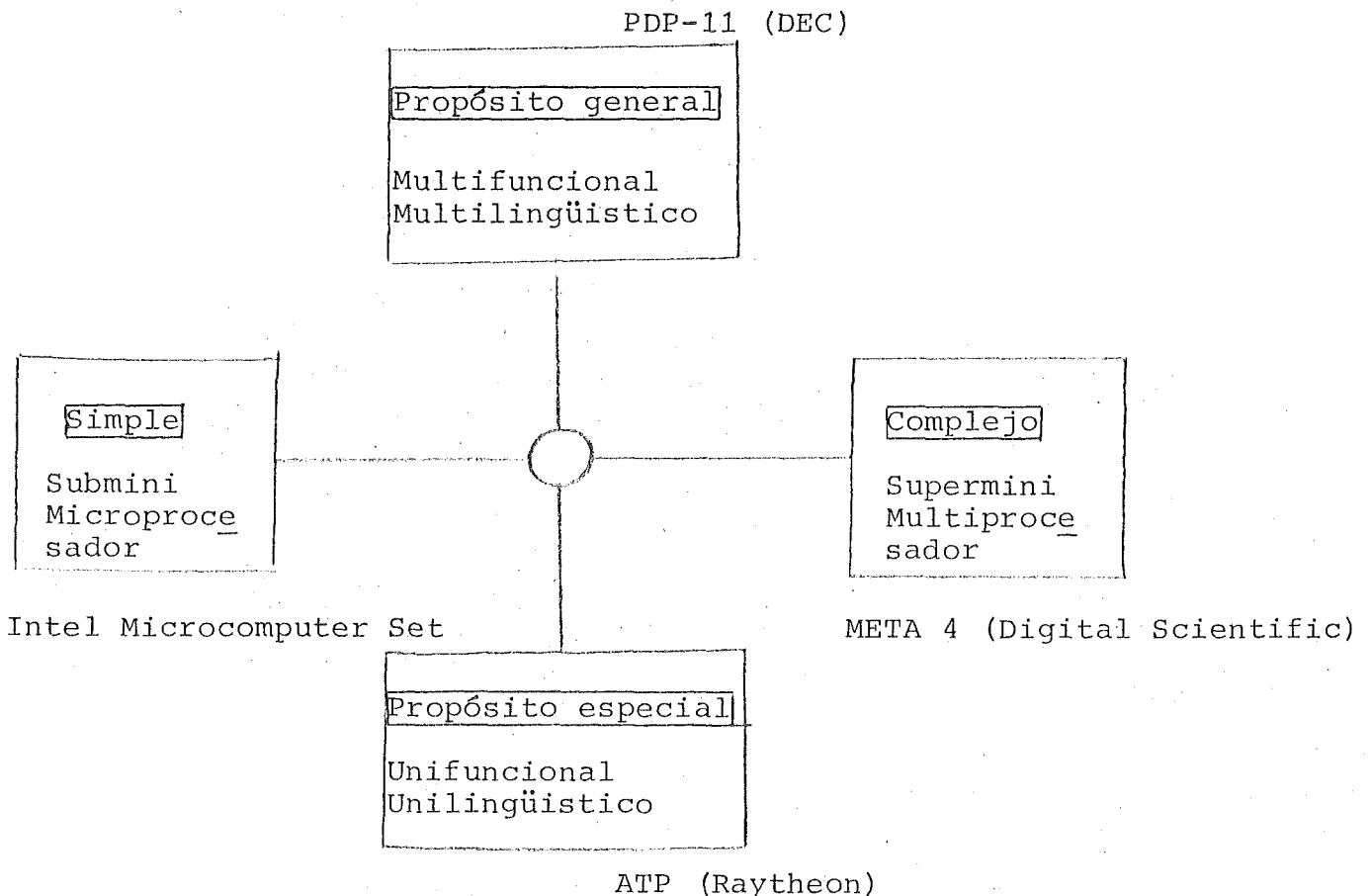
Hay como un retrato-robot de miniordenador (nos referimos sólo a los minis científicos o industriales), que es el siguiente:

16 bits, longitud de información  
palabra-dato en coma fija  
aritmética paralela  
datos negativos en complemento a dos  
direccionamiento por paginación (página cero y corriente), indexación e indirección  
instrucciones de referencia a memoria de una sola dirección  
repertorio de 70 o más instrucciones  
memoria máxima entre 32 y 64 k palabras  
ciclo base, alrededor de 1 microsegundo  
tecnología en parte o totalmente MSI, incluyendo a veces a la memoria.  
interrupción de programa a varios niveles (muchas veces, hasta 64), reloj de tiempo real y acceso directo a memoria; macroensamblador, compilador Fortran, frecuentemente Intérprete de Basic  
sistema operativo elemental  
bajo precio  $\leq$  20.000 \$  
pequeño tamaño, normalmente de sobremesa  
desnudez en cuanto a periféricos propios

El miniordenador es, pues, el resultante de un proceso de diferenciación de los ordenadores que ha producido no pocas variantes, estabilizándose hacia 1.970 mas o menos en torno a una estructura y características, como las listadas en el retrato robot. La estabilización se produce como consecuencia de un proceso de integración.

Mas a un proceso de integración sigue uno de diferenciación, y así sucesivamente. Partiendo del miniordenador tipo se ha producido, según Koudela 1.973, una diferenciación de los minis en cuatro direcciones principales, que representa el esquema siguiente.

Los que llamaríamos de "propósito general" y "complejo" llegan a no distinguirse de los ordenadores del tronco básico más que en su pequeña longitud de información y en la escasa dotación de periferia específica.



Koudela clasifica las aplicaciones de los minis en cuatro amplias categorías de

- a) sólo procesamiento
- b) adquisición y procesamiento
- c) adquisición, procesamiento y control
- d) multiprocesamiento

La 2<sup>a</sup>. y 3<sup>a</sup>. categoría implican la existencia de fuentes exteriores de medida o señalización de las que el ordenador adquiere automáticamente datos. En la 3<sup>a</sup>. se realiza también el control de dichas fuentes en base al procesamiento de los datos de ella adquiridos.

En cuanto a la 1<sup>a</sup>. categoría, en ella entran básicamente las aplicaciones científicas.

En la 2<sup>a</sup>. incluimos todas aquellas relacionadas con las comunicaciones:

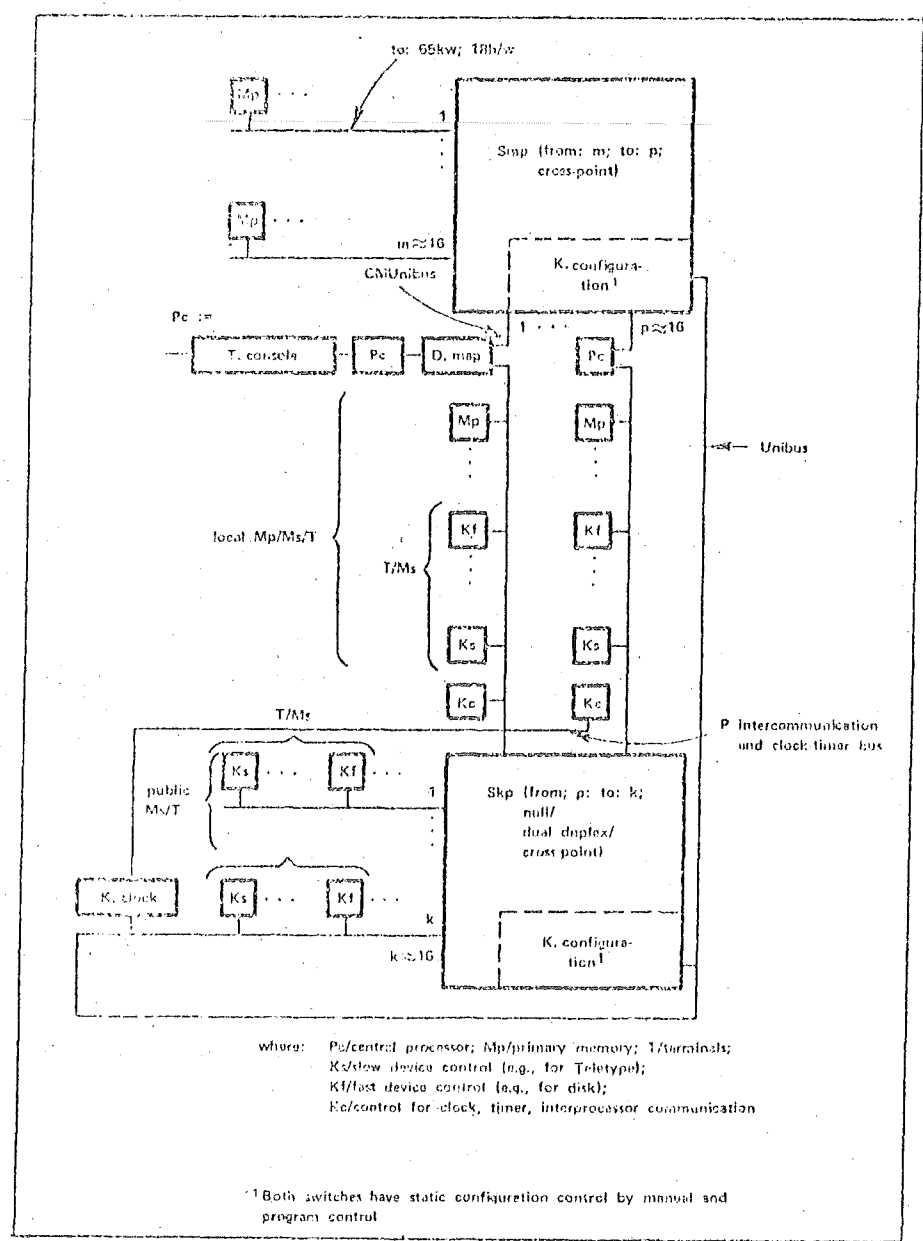
- realizar interfaces entre entradas diferentes y una sola instalación de comunicaciones.
- empaquetar o desempaquetar caracteres
- localizar errores en los datos
- sincronizar datos asíncronos
- formatar
- transcodificar
- multiplicar muchos terminales de baja velocidad sobre una línea de alta
- coordinar las entradas de comunicaciones en los grandes ordenadores.

En la última categoría debemos mencionar el proyecto de la Universidad Carnegie Mellon llamado Ordenador Multimini-

procesador que, basado en el PDP-11, se propone estudiar los siguientes tipos de problemas:

- Procesamiento paralelo.
- Procesamiento pipe-line
- Procesamiento en red
- Procesamiento especializados
- Multiprogramación convencional con multiprocesadores.
- Computadorización individualizada.

La siguiente figura presenta la estructura general de este sistema (Bell, Chen, Rege, 1.972). Este sistema podíamos



haberlo encuadrado en el apartado I.6.1., pero su origen le sitúa mucho mejor en relación con el movimiento de los minis.

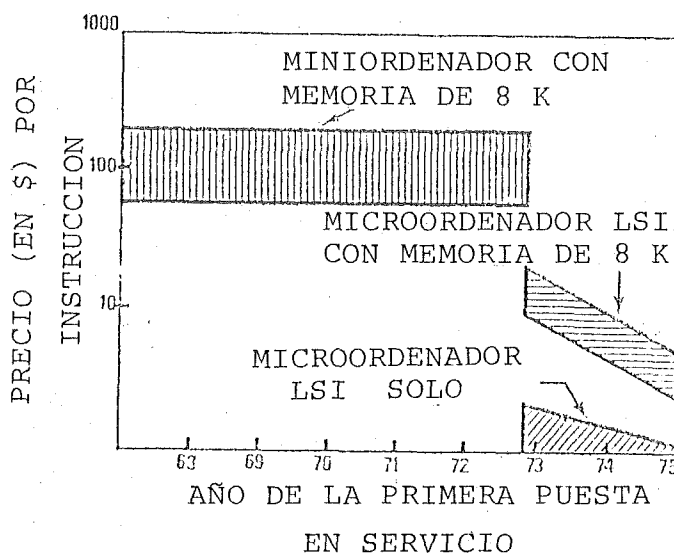
Volviendo a la diferenciación mencionada que se produce en la rama de los minis, recordemos la que hace referencia a los micros. Aquí se produce un nuevo núcleo que puede llegar a ser centro de una nueva integración. Puede contribuir a ello, entre otros factores, un movimiento de pensamiento de aplicación de la informática que se llama micro-informática o informática repartida (o distribuida).

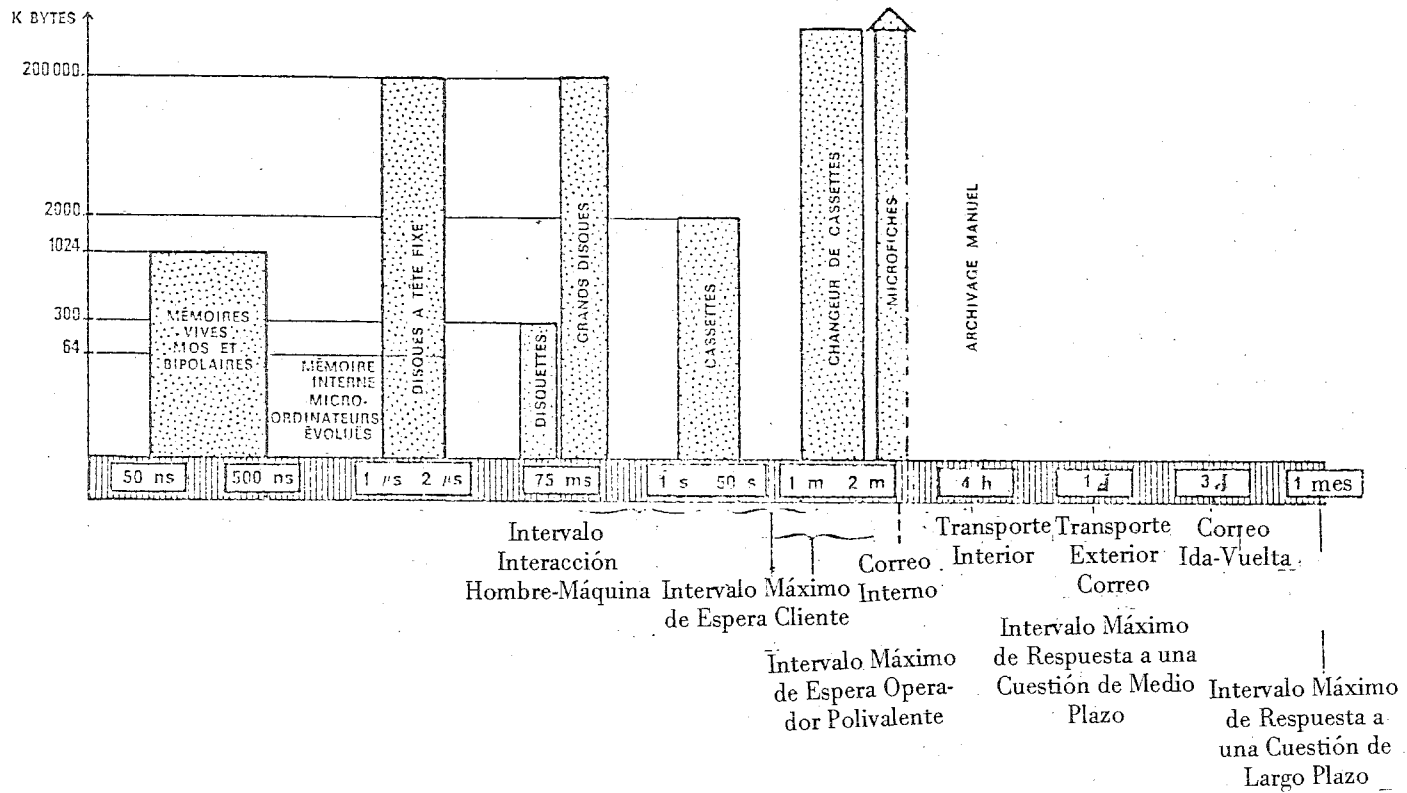
En conferencia anterior se ha presentado con gran minuciosidad la estructura, funcionamiento y programación de los microprocesadores. Añadamos aquí algunos datos curiosos o interesantes relativos a esa nueva tecnología, que se comentarán.

En primer lugar, una figura que representa el tiempo de respuesta y la capacidad de algunas memorias, entre las que se contabilizan algunas típicas de la informática distribuida.

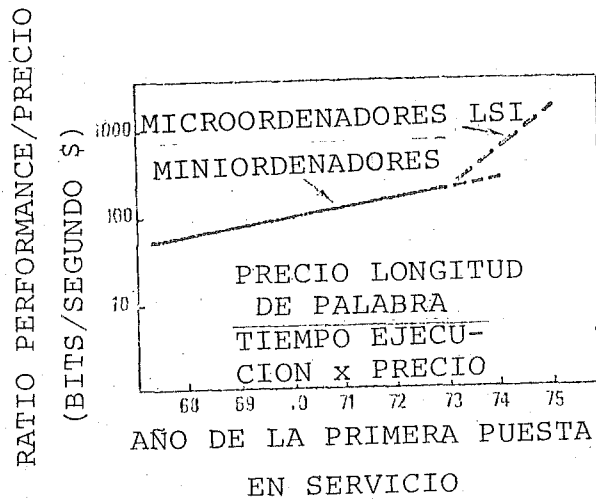
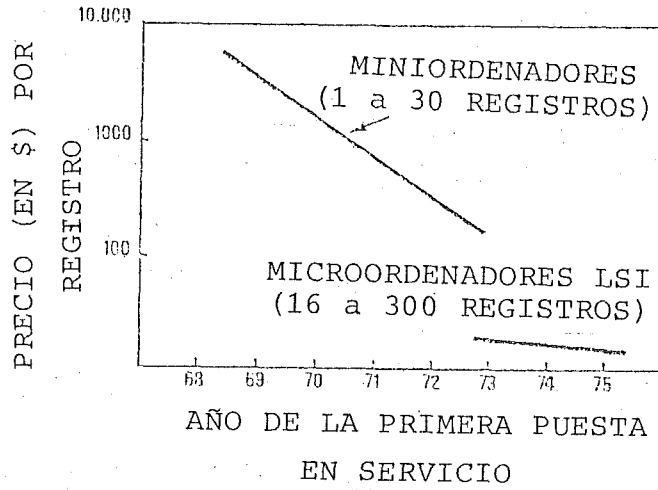
Los siguientes gráficos presentan, respectivamente, - los ratios (Lussato et al.):

- precio/instrucción
- precio/registro
- performance/precio



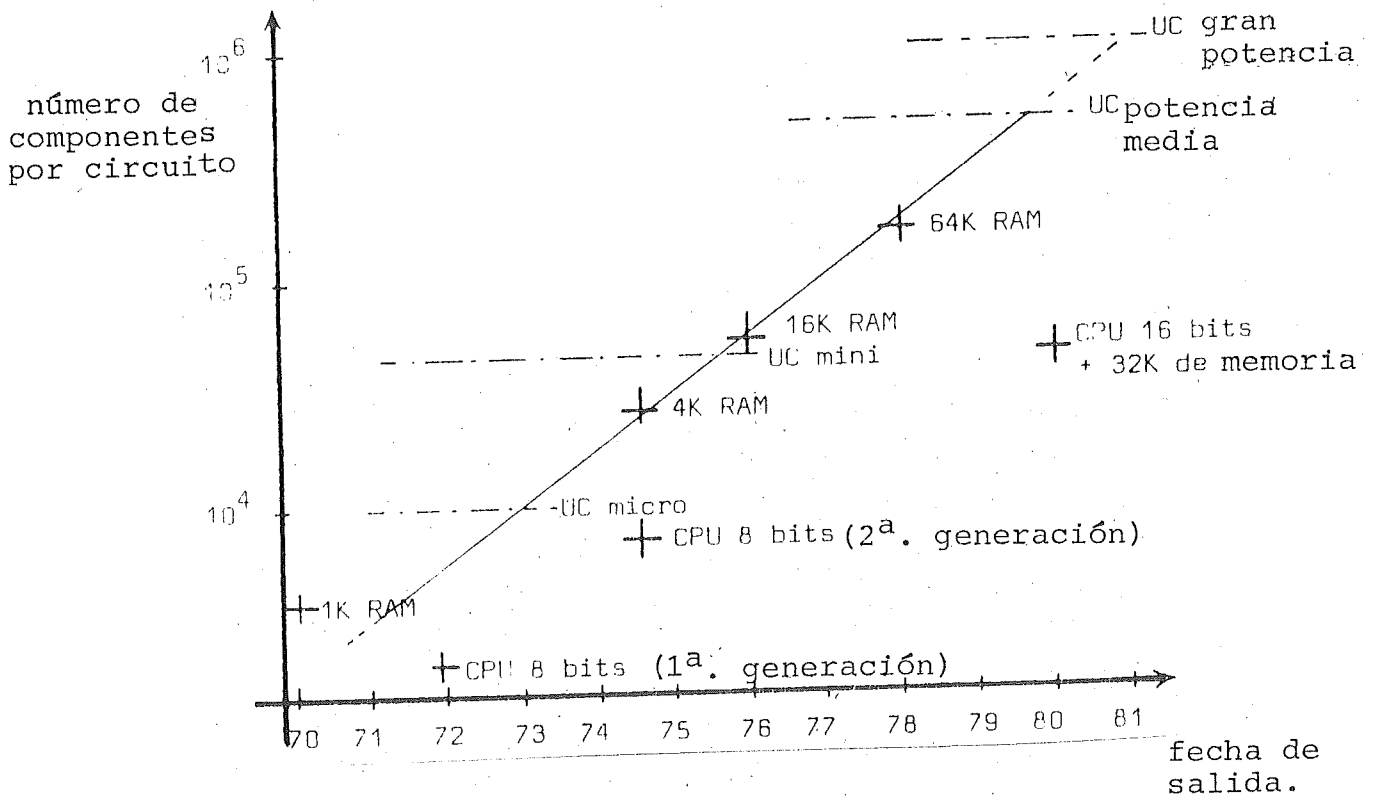


Tiempo de Respuesta Respecto a la Capacidad de la Memoria Prevista

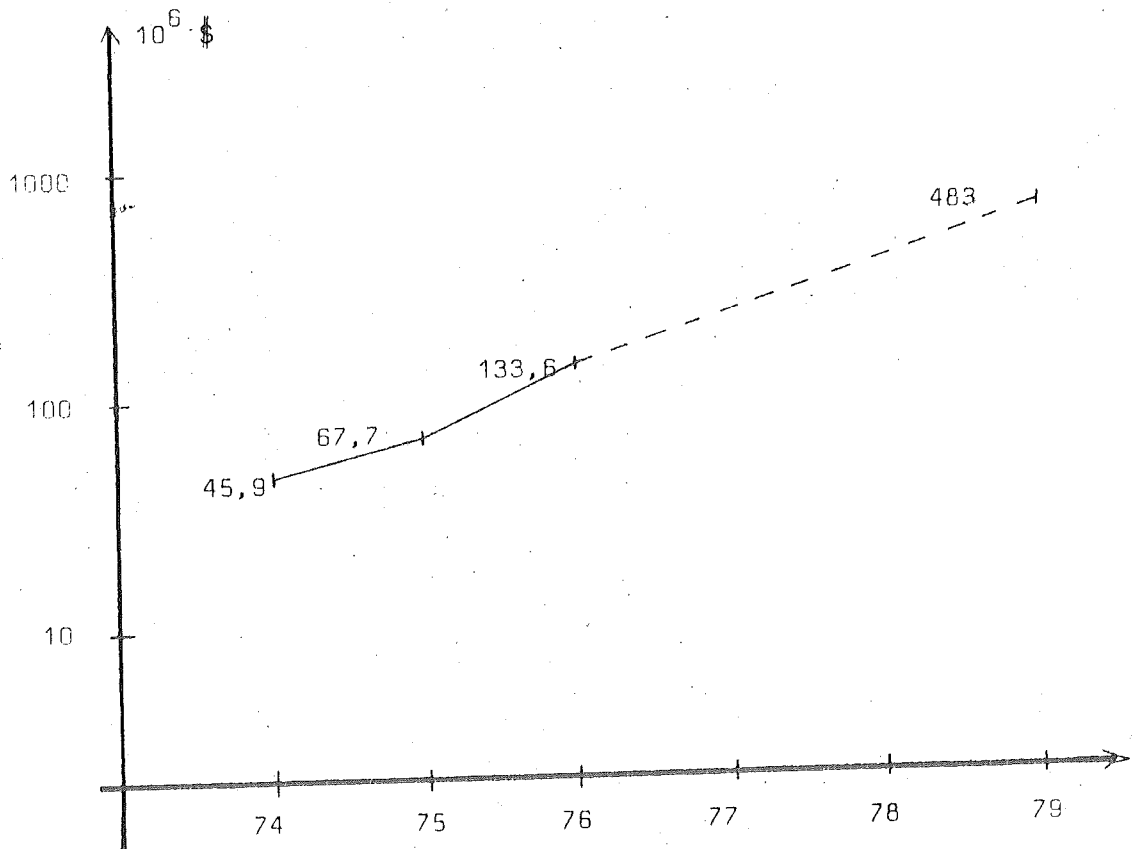


La curva que presento a continuación da idea, sin reflejar el precio, de la evolución de la complejidad de los circuitos. Se prevé para 1.980 un circuito integrado equivalente a la unidad central completa de un mini actual (Anceau, 1.976) y para hacerse un orden de magnitud del mercado mundial de micro-





procesadores tomamos la siguiente curva del mismo estudio (Anceau, 1.976).



### I.7. Ordenador promedio típico del año 2.000

Como se ha visto, las estructuras de los ordenadores - obedecen a sucesivos refinamientos que hacen brotar nuevas ramas de un árbol cuyo tronco sigue alimentado por los conceptos clásicos de Von Neumann. Se han producido ramas que han adquirido una especial robustez y dan lugar a nuevas ramas de sí mismas. No todas ellas son lógicas y acaso el tiempo irá mustiéndolas, porque el campo del diseño del hardware y del software (y por extensión, el firmware) de los ordenadores es extraordinariamente joven e inestable.

¿Qué ordenadores tendremos en el futuro?. Foster, reputado profesor del M.I.T. en Arquitectura de Ordenadores, se - - atreve a profetizar un retrato-robot del ordenador promedio en el año 2.000:

- máquina intérprete capaz de ejecutar directamente - uno o más lenguajes de alto nivel.
- palabras más largas que las máquinas de hoy, posiblemente con tantos como seis campos de dirección por - instrucción.
- predominantemente máquina autónoma con provisión para ocasional acceso remoto de datos.
- sin registros centrales
- probablemente máquina decimal
- con un pequeño sistema operativo cableado (¿microprogramado?)
- con protección de cada palabra y descripción de datos
- monoprocesador que hace sus propias E/S
- lo más probablemente de propiedad privada y monoprogramada.

## II. SOFTWARE

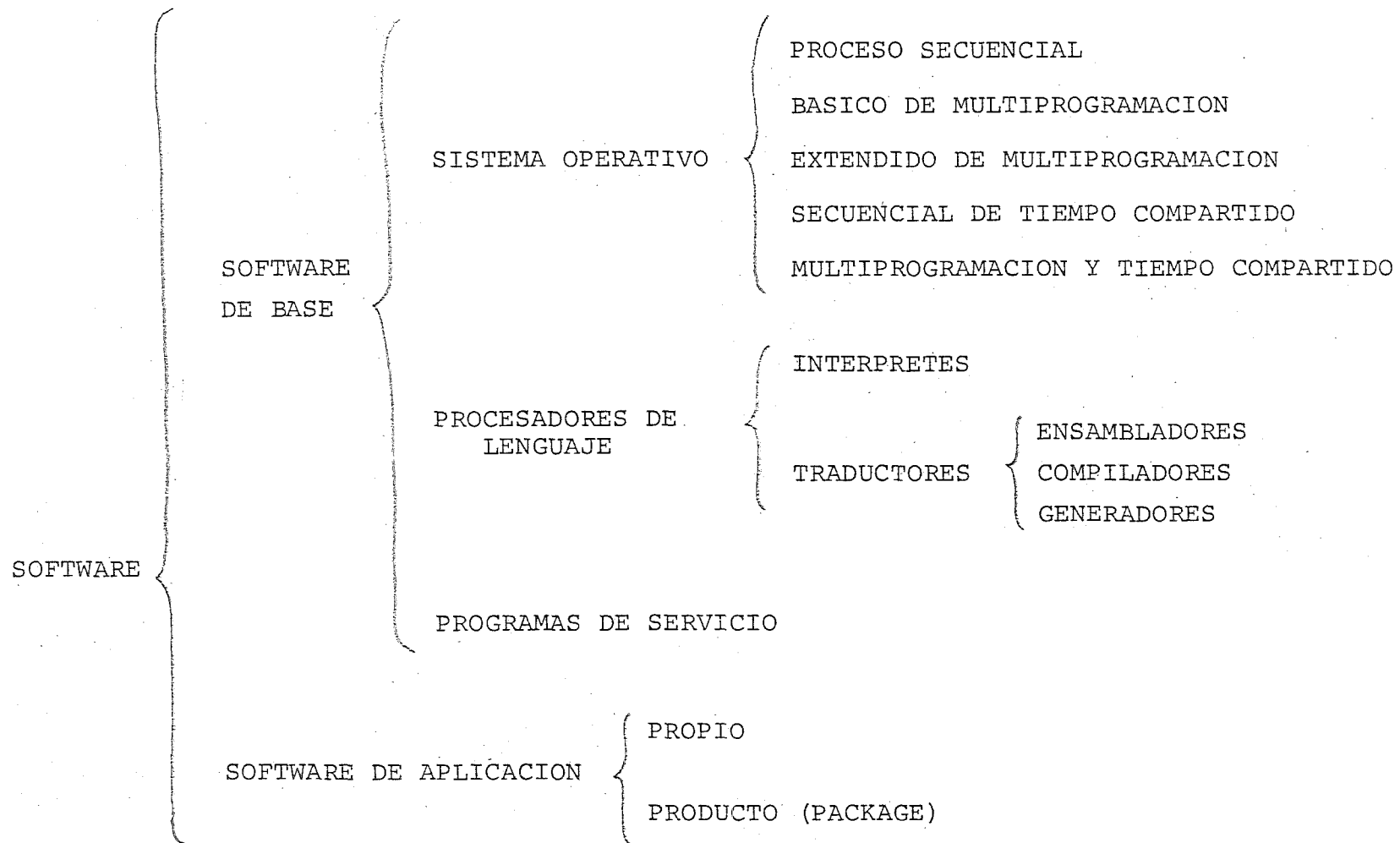
### II.1. Clasificación y algunas definiciones

Al software, que es la programación de los ordenadores, se le empieza a llamar ahora en España "logical" o "logicial". Se puede establecer una relativamente buena clasificación del software, que damos en la página siguiente.

El software de base viene realizado por el constructor y entregado normalmente con la máquina, primero gratuitamente y ahora cada vez más en tarifa aparte (procedimiento "unbundling") Es tanto más abundante y complejo cuanto más complejo y caro es el equipo material.

Sin entrar en detalles, puede definirse un sistema operativo como lo hace la IFIP (International Federation of Information Processing): "Un sistema operativo es una colección ordenada de rutinas y procedimientos que acompañan al ordenador. - Normalmente realizan alguna o todas las funciones siguientes: 1) Planificación, carga, iniciación y supervisión de ejecución de programas; 2) Asignación de memoria, unidades de entrada/salida y otros dispositivos del sistema; 3) Iniciar y controlar operaciones de E/S; 4) Manejar errores y reiniciaciones; 5) Coordinar comunicaciones entre el operador y el sistema; 6) Mantener un diario de las operaciones del sistema; 7) Controlar operaciones en los modos multiprogramación, multiprocesamiento y "time-sharing". Nota: entre las facilidades incluidas frecuentemente en el sistema operativo figuran un supervisor, un planificador, un control de E/S, rutinas de servicio y rutinas monitoras".

Hay tantas categorías de sistemas operativos como modos, entornos o ambientes de explotación.



#### SISTEMAS DE PROCESO SECUENCIAL

- Todos los recursos asignados a un job
- Un solo job en memoria
- ejecución en el mismo orden de la corriente de entrada

#### SISTEMAS BASICOS DE MULTIPROGRAMACION

- Varios jobs (jobsteps, tasks) coexisten en memoria
- A cada corriente de entrada se le asigna una participación de memoria
- Dentro de cada corriente de entrada, los jobs se ejecutan en orden secuencial
- Planificación únicamente externa (usuario)

#### SISTEMAS EXTENDIDOS DE MULTIPROGRAMACION

- Varios jobs (jobsteps, tasks) coexisten en memoria
- No existen particiones fijas en memoria
- Unica cola de entrada para todas las corrientes de jobs
- Asignación de recursos según las necesidades
- Planificación externa (usuario) e interna (sistema)

#### SISTEMAS SECUENCIALES DE TIEMPO COMPARTIDO

- Un solo job (jobstep, task) en memoria
- Cola cíclica de asignación de procesador
- Asignación de "quantum" de tiempo a cada job
- Fenómenos de swapping (canjeo de información entre memorias central y auxiliar).

#### SISTEMAS DE MULTIPROGRAMACION Y TIEMPO COMPARTIDO

- Permiten distintos tipos de proceso
- Tienen todas las características de los sistemas extendidos de multiprogramación
- Tienen también características de los sistemas de tiempo compartido
- Suelen distinguir dos zonas en memoria
  - . Foreground (tiempo compartido)
  - . Background (batch)

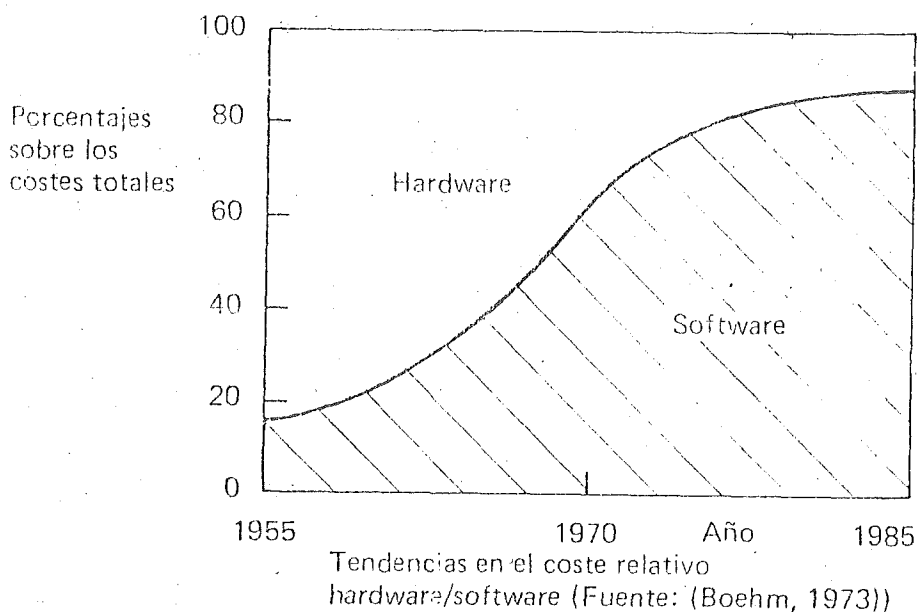
Los procesadores de lenguaje son programas que transforman el programa objeto escrito en lenguaje fuente en programa objeto escrito en lenguaje final. Hoy está bastante avanzada la teoría de los compiladores que, contra lo que muchos piensan, es solo una categoría -si bien la más importante- entre los pro

cesadores. (Algunos autores incluyen los procesadores de lenguaje en el sistema operativo, y también algunas casas constructoras).

El software de aplicación es el conjunto de programas escritos para resolver problemas específicos de los usuarios. Por ello mismo es enormemente variado, y, al tiempo, enormemente redundante.

### II.2. Problemas del software

El primero de todos es el coste. Al revés que sucede con el hardware, el coste del software es creciente. La situación relativa viene representada en la gráfica. Pero no solo el coste sino que todas las características inherentes a la tecnología electrónica: velocidad, fiabilidad, tamaño y coste, evolucionan en el software en dirección contraria. A fin de cuentas, todo podría evaluarse a más corto o largo plazo y cifrarse en coste.



Así:

Velocidad: decrece, por utilizarse cada vez más frecuentemente lenguajes evolucionados. También, por incorporarse progresivamente nuevas funciones de control. Dialéctica hardware-software (y compensación por el hard).

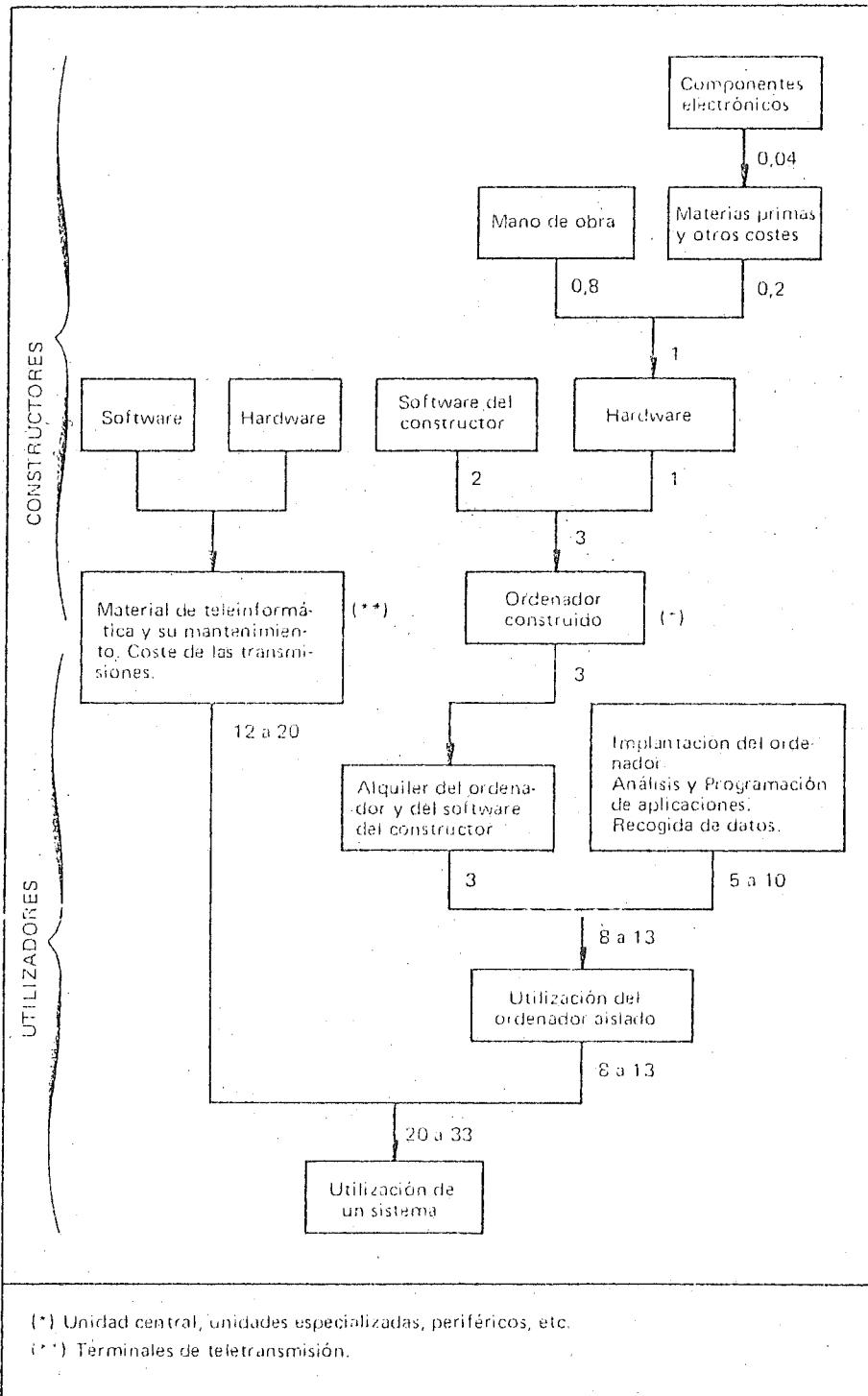
Fiabilidad: La falta de metodología de diseño, unida a la creciente complejidad de los programas, influye negativamente en la fiabilidad. Tal influencia no se percibe a corto sino a medio y largo plazo. Problema del mantenimiento del software.

Tamaño: Al incluir nuevas funciones o amplificar las ya existentes crece sin tasa la dimensión de los programas. Esto implica, además de consumo de tiempo (y por consiguiente disminución relativa de velocidad) consumo de memoria. Nuevamente, dialéctica hard-soft.

Coste: Ya visto. Este punto es siempre creciente ya que no está nada automatizada la producción de soft.

El cuadro siguiente (fuente: Armand, recogida en mi Memoria de Cátedra) merece reflexión, porque aparecen destacadas las capas de coste desde el componente electrónico hasta el sistema en funcionamiento. Aunque este cuadro data del año 1.970, y desde entonces los costes referidos a actividades típicamente humanas han experimentado crecimientos muy fuertes, debe observarse el volumen económico relativo del software de base ("soft. del constructor" en la figura), más lo que hemos llamado software de aplicación, asociado en la misma figura al cuadro donde se refleja el "Análisis y Programación de aplicaciones".





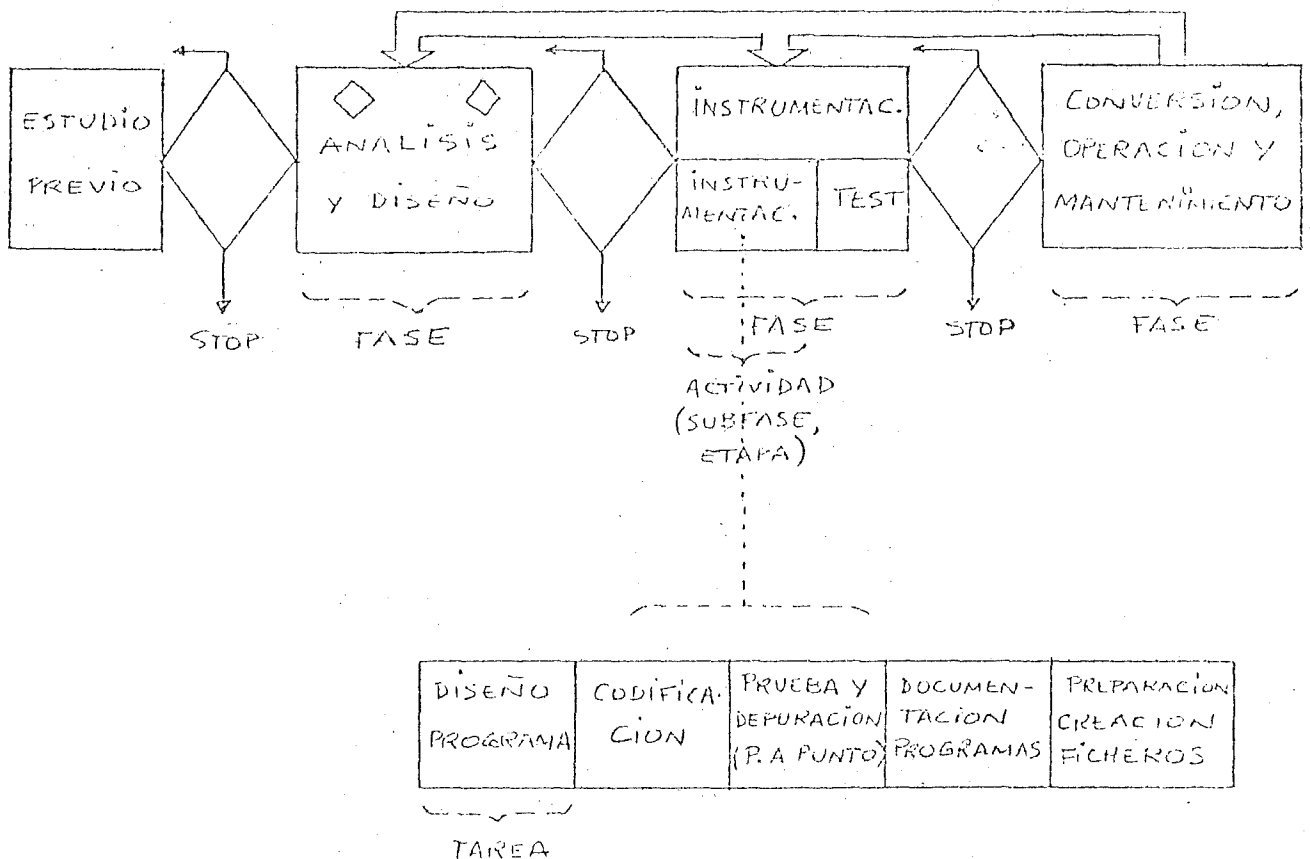
Toda esta problemática preocupa y por ello desde hace algún tiempo se viene analizando como una actividad típica y fundamental de la informática en general y, muy en particular, de la industria de los ordenadores.

Recogeremos aquí algunos datos que cuantifican las etapas y los costes característicos del proceso de desarrollo y mantenimiento de software.

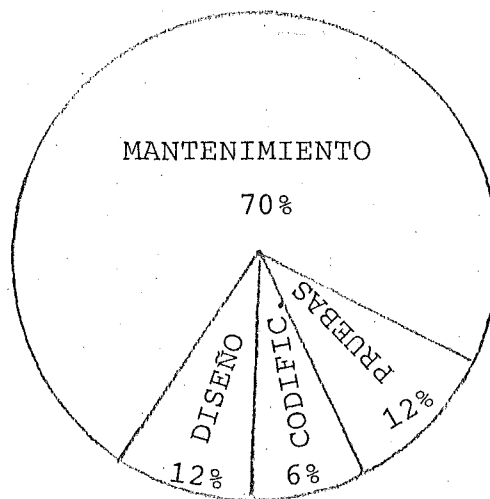
II.3. Estructura y costes del proceso de desarrollo y mantenimiento del software.

El desarrollo, y correspondiente mantenimiento, de un sistema de información y, en general, de un software específico sigue unas grandes líneas que han sido tipificadas como describiendo un ciclo, el "ciclo de vida de un sistema".

CICLO DE VIDA DE UN SISTEMA



mantenimiento (70%). Nuestros estudios y datos personales en relación con el software de aplicación en Informática de Ges--



ción, arroja alrededor de un 45%.

El mantenimiento hay que entenderlo como la repetición de algunas (o de todas) de las tareas de algunas (o de todas) de las actividades de algunas (o de todas) de las fases que componen el ciclo de un sistema. El mantenimiento hay que realizarlo por dos grandes motivos:

- 1º) El entorno del sistema ha cambiado
- 2º) El sistema no satisface todas las condiciones de contorno y se presentan fallas. Este es el problema de la fiabilidad que es, en particular, uno de los mayores azotes de la programación.

El problema de la fiabilidad muy sentido en las estructuras de hardware, es especialmente grave en el software, y es que, aunque parezca paradójico los programas para ordenador se diseñan y realizan casi siempre sin seguir una lógica.

#### II.4. Las técnicas de programación estructurada

Hoy se está extendiendo por el mundo especializado de la informática un amplio movimiento encuadrado bajo el nombre de programación estructurada, que busca hacer sistemática y lóg<sub>u</sub>ica la actividad de programación. Sus efectos son:

- aumentar la fiabilidad de los programas
- mejorar la comunicabilidad de los programas
- reducir el tiempo y costes de pruebas de los programas, etc.

Se basa en tres grandes principios:

- 1°. Diseño del programa por procedimiento "top-down" (de lo general a lo particular).
- 2°. Empleo de solamente tres tipos de estructuras básicas (secuencial, repetitiva, alternativa)
- 3°. Empleo del recurso abstracto.

Dado que todo método de programación que respete estos tres principios es programación estructurada pueden encontrarse varios métodos de P.E. Se conocen los de Warnier, Bertini y Jackson. En el Anexo se recoge un artículo de Ferrán que, de manera muy simplificada, presenta (y nos ahorra describir) las ideas fundamentales de los métodos de P.E.

Para hacerse una idea de los orígenes, alcance e implicaciones de la Programación Estructurada puede consultarse (Sáez, 1.975 Inforprim).

Lo cierto es que la programación estructurada se utiliza aún poco en el mundo y nos consta que, desgraciadamente, bastante menos, relativamente, al nivel del diseño.

Razones de la dificultad de difusión:

- Falta de formación
- Resistencia psicológica al cambio de método y a la disciplina de trabajo.
- Interpretación errónea e individualista del concepto de eficiencia de un programa y de la profesión de programador.
- Encaje imperfecto Programación estructurada-lenguajes habituales de programación.

Y hablando de fiabilidad, creemos que al lector le resultarán instructivas las "leyes de la infiability" de Gilb.

1<sup>a</sup>. Los computadores son infiables, pero los seres humanos lo son aún más.

Corolario:

Como origen de cada error que se le imputa al computador, se encontrarán por lo menos dos errores humanos, incluyendo el error de imputárselo al computador.

2<sup>a</sup>. Cualquier sistema que dependa de la fiabilidad humana es un sistema infiable.

3<sup>a</sup>. Un sistema tiende a crecer en complejidad más que en simplicidad hasta que la resultante infiability se hace intolerable.

4<sup>a</sup>. Los sistemas autocomprobantes tienden a una complejidad proporcional a la inherente falta de fiabilidad del sistema en que se emplean.

5<sup>a</sup>. Las capacidades de corrección y detección de error de cualquier sistema servirán como claves para com

prender el tipo de error que no pueden manejar.

- 6<sup>a</sup>. Los errores indetectables son infinitos en su variedad, contrastando con los errores detectables que son, por definición, limitados.
- 7<sup>a</sup>. Todos los programas reales contienen errores mientras no se demuestre lo contrario-lo cual es imposible.
- 8<sup>a</sup>. La inversión en fiabilidad crecerá hasta que supere al coste probable de los errores, o hasta que alguien insista en que todo el trabajo útil ya está hecho.

Un freno relativo, pero un freno al fin, es el constituido por los lenguajes que se emplean realmente. Prácticamente ninguno de los usados se acomoda perfectamente a la descripción de todas y cada una de las estructuras. De los lenguajes más utilizados en la práctica (ver el cuadro siguiente, donde el índice de uso = número de usuarios/respuestas totales x uso medio, según sondeo de Philippakis para los U.S.A.) el que menos mal se adapta es el PL/I.

<u>Lenguaje</u>	<u>Índice de uso</u>
COBOL	59
ENSAMBLADOR	20
RPG	6
FORTRAN	5
PL/I	4
OTROS	3
BASIC	1
APL	0

Un lenguaje totalmente estructurado, porque se ha creado después que la P.E., es el PASCAL, del que para que el lector se haga una idea visual, reproducimos un programa del último libro de Wirth.

```

var ch: char;
    sym: symbol;
    num: integer;
    id: record
        k: 0..maxk;
        a: array [1..maxk] of char
    end;
procedure scanner;
    var ch1: char;
begin {skip blanks}
    while ch = ' ' do read(ch);
    if ch in ['A'.. 'Z'] then
        with id do
            begin sym := identifier; k := 0;
                repeat if k < maxk then
                    begin k := k+1; a[k] := ch
                        end;
                    read(ch)
                until ¬(ch in ['A'.. 'Z', '0'.. '9'])
            end else
            if ch in ['0'.. '9'] then
                begin sym := number; num := 0;
                    repeat num := 10*num+ord(ch)-ord('0');
                        read(ch)
                    until ¬(ch in ['0'.. '9'])
                end else
            if ch in ['<', ':', '>'] then
                begin ch1 := ch; read(ch);
                    if ch = ':' then
                        begin
                            if ch1 = '<' then sym := leq else
                                if ch1 = '>' then sym := geq else sym := becomes;
                                    read(ch)
                                end
                            else sym := S[ch1]
                        end else
                begin {other symbols}
                    sym := S[ch]; read(ch)
                end
            end
end {scanner}

```

PROGRAMA: UN SCANNER

## II.5. El futuro del software. Resumen de problemas y recomendaciones.

Por su interés, vamos a reproducir lo que, sobre este punto, especifica el reciente estudio, ya citado, de Dolotta y otros.

La producción de software está aún en su infancia. Que los profesionales del software hablan raras veces de sus fallos y aprenden poco, si aprenden algo, de ellos, es un indicador. - Que tienden a sobrevalorar lo que tiene la apariencia de éxito, es otro. La industria puede, ciertamente, continuar dejando que "la naturaleza siga su curso" hacia un futuro indeterminado, pero ese futuro no tiene probablemente que ser un futuro deseable. No todos los problemas planteados estarán resueltos para 1.985, ni las soluciones alcanzadas serán de uso general para la misma época. Por ello, se deben asignar prioridades al conjunto total de los problemas y comenzar a buscar soluciones a los más importantes ya mismo. Si el entorno del procesamiento de datos ha de ser más confortable para entonces habrá que influir antes sobre las casas vendedoras para que provean hardware y SCP que se adapten mejor a estas necesidades claramente establecidas. (SCP = System Control Program).

La industria tiene que empezar ahora a intentar conducir la ocupación de desarrollo del software tal como existe, influyéndola y orientándola hacia la posición de aceptar y adoptar los resultados de estudios, investigación y desarrollos. Individual y colectivamente, profesionales y managers tienen que empezar a atacar los problemas más urgentes y a compartir los resultados de sus esfuerzos. Tienen que apreciar la necesidad de un conjunto de estándares útiles y aplicables a la industria y trabajar para elaborarlos.

Tanto los productores como los usuarios de software tienen que aceptar el hecho de que una parte muy grande (si no



la mayor) del coste de un sistema de software sobre su ciclo de vida se debe al cambio, y que se tienen que desarrollar métodos para diseñar e implementar software que se preste bien a este estado de cambio constante (en requerimientos, en estructuras de datos, en interfaces, en el hardware subyacente, en SCP, etc.), minimizando así los costes del cambio.

Aunque algunas de las herramientas y técnicas necesarias están ya disponibles, y otras existen en estado experimental o de prototipo, hay una cantidad de problemas que requerirán el inicio de nuevos proyectos de investigación y otros que requerirán una amplia aplicación de sentido común y buenas prácticas de buena gestión. Más abajo los citamos.

Ya que todo dentro del entorno del procesamiento de datos afecta de alguna forma a los programas de aplicación o a su implementación, no se hará nada para confinar la lista siguiente a las aplicaciones en sí mismas; por el contrario, lo siguiente es una delineación completa de todos los problemas en el área del software:

- La necesidad de mejorar la productividad de los programadores de aplicación y del proceso de desarrollo de aplicaciones es un problema muy difícil y urgente. Todo esfuerzo emprendido hacia la mejora del software contribuirá a su solución. Relacionada directamente a este problema está la falta de una adecuada medida de la productividad de los programadores.

- Nuevos lenguajes de implementación, en sí mismos, ofrecen poco incremento en productividad. Sin embargo, diseñados como parte integrante de un sistema total de producción de software, podrían tener suficiente impacto para justificar invertir en su creación. Se dice esto en el supuesto de que el diseño de un tal sistema de producción implique o contenga restricciones acerca de cómo deben expresarse o estructurarse los programas.

- Minimización de la fracción del nuevo código necesario para cada nueva aplicación puede tener un efecto marcado. - Se definirán y seleccionarán módulos reutilizables y los necesarios subsistemas de soporte.

- Tienen que encontrarse mejores, más rápidos, más eficientes formas de transferir nuevas capacidades, tan pronto estén disponibles, de aquellos que las desarrollan a aquellos que las necesitan. Este es un problema extraordinariamente complejo que involucra gestión, economía y -demasiado frecuentemente- personalidades.

- Se tiene que mejorar la calidad de los productos de software desde los puntos de vista de fiabilidad, utilizabilidad y flexibilidad. Entre otras cosas, esto implica formas mejoradas de diseñar y probar programas y también meter a los usuarios finales desde el principio en el ciclo de desarrollo haciéndoles partícipes. Significa asimismo hacer a ambos, a los que desarrollan y a los que usan, responsables.

- Tiene que adoptarse una visión integrada del sistema total de procesamiento de datos, incluyendo al usuario final y a su entorno de trabajo tanto como al computador que constituye el corazón del sistema. Los beneficios y desventajas del concepto SCP-ICP, potencia de procesamiento de datos distribuida, sistemas jerárquicos de memoria, etc. tienen que ser evaluados como un todo integrado.

- Es cuestionable si podrán ser resueltos para 1.985 la mayoría de los siguientes problemas:

- fiabilidad del sistema global de procesamiento de datos.
- instalabilidad de grandes sistemas on-line
- desarrollo de un SCP que no sea violado
- metodología para el uso de bases de datos distribuidas

das y redes de ordenador

- formación y educación de los profesionales del procesamiento de datos.
- desarrollo de la programación como una tecnología
- medidas del impacto del procesamiento de datos sobre la productividad total de la empresa.
- mejora de la, en general, mala imagen pública de la industria del procesamiento de datos.

### III. INFORMATICA

#### III.1. ¿Cómo y en qué utilizar el ordenador?.

Se ha hablado del hard y del soft y no se puede dejar de esbozar unas ideas sobre algunos de los problemas y tendencias de la informática de hoy.

Recordemos que la informática se ha definido como la "ciencia del tratamiento racional, principalmente por máquinas automáticas, de la información, considerada como el soporte de los conocimientos y comunicaciones en los terrenos técnico, económico y social".

Dreyfus distingue hasta cinco grandes ramas en la Informática:

- Informática formal y analítica
- Informática física y tecnológica
- Informática metodológica
- Informática sistemática y lógica
- Informática aplicada.

Lo cierto es que en la práctica se confunden informática e industria de los ordenadores. Sáez (1.973) distingue tres etapas fundamentales en la historia de los ordenadores.

El cuadro acentúa el hecho de que han surgido suficientes síntomas para pensar que se está empezando a poner énfasis en los problemas de utilización del ordenador; su coste, sus configuraciones, las metodologías de su uso, etc. Quizá ocurra esto porque el conjunto de usuarios es ya muy cuantioso y se empieza a tomar conciencia de que el ordenador, como herramienta increíblemente potente, no se sabe aún utilizar en una mínima parte de sus posibilidades. Sucede también que, impulsado por

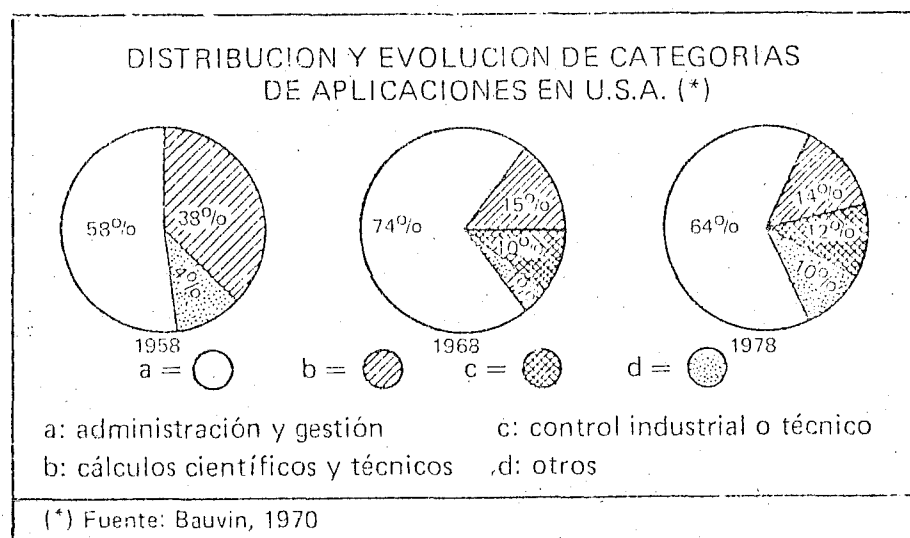
EPOCA (años aprox.)	SIGNO	CARACTERÍSTICAS MUY SOBRESALIENTES	GENERACIONES	CENTRO DE INTERESES EN CUANTO AL ORDENADOR	OBJETIVOS	LOCALIZACIÓN
1935-50	CREACION	INTELIGENCIA 1ª LINEA MATERIALIZACION 1ª ORDENADORES	GENERACION BASE, SIN NUMERO	GESTACION	RESPONDER A UN DESAFIO INTELLECTUAL	UNIVERSIDADES
1950-70	EXPANSION, EUFORIA	3ª INDUSTRIA MUNDIAL AVANCES TECNOLOGIA EVOLUCION TECNICA	1ª, 2ª, 3ª	DESARROLLO	NEGOCIO	EMPRESAS FABRICANTES
1970-	EXAMEN DE CONCIENCIA, RACIONALIZACION	PREGUNTAS: ¿sustituira el ordenador al hombre? ¿no se habrá mitificado excesivam. al orden.?		UTILIZACION	RACIONALIZAR EL USO DEL ORDENADOR	USUARIOS

el avance de la tecnología han aparecido máquinas tan poderosas o más que los primeros ordenadores y que caben, por así decirlo, en una cartera de mano. Máquinas que están al alcance de cualquiera. Las estadísticas de ordenadores no las recogen, ya que el mismo concepto de ordenador se ha quedado obsoleto en su uso cotidiano. Las estadísticas de minis y micros se han empezado a hacer por separado.

EVOLUCION PARQUE ORDENADORES (DIEBOLD computer census) (*)		
	1969	1975 (estimado)
Estados Unidos	56.151	110.000 - 170.000
R.F. Alemana	5.007	19.000 - 31.000
Benelux	1.810	7.000 - 11.000
Francia	3.580	14.000 - 30.000
Gran Bretaña	3.880	15.000 - 25.000
Italia	2.480	9.000 - 15.000
Resto Europa Occidental	2.280	8.000 - 16.000
España (**)	515	2.500 (***)
Total Europa Occidental	19.037	72.000 - 128.000
Japón	4.900	18.000 - 40.000
U.R.S.S.	6.000	?

(\*) Fuente: Margulici, 1970.  
 (\*\*) Fuente: Rodríguez, 1971.  
 (\*\*\*) Fuente: Servicio Central de Informática. Presidencia del Gobierno (citado por Rodríguez)

El cuadro siguiente hace ver cómo evoluciona relativamente el uso de los ordenadores en cuatro grandes categorías - de aplicaciones en los U.S.A. Observese de qué manera crecen, para volver luego a una posición anterior, las aplicaciones -



llamadas de gestión y cómo crecen definitivamente las aplicaciones no tipificadas. Para conocer en detalle los datos referentes a nuestro país, consúltese el estudio, publicado en - - Abril de 1.977 por la Presidencia de Gobierno, con el título - de la Informática en España 1.976 y en el que ha colaborado el autor de esta conferencia.

Anteriormente hemos presentado un cuadro-organigrama con los costes del ordenador desde el de los componentes hasta el sistema funcionando. Ahora veamos concretamente lo que le - cuesta, en promedio, a una empresa su informática (Sáez 1.973).

Coste Informática del 0,5 al 1,5% cifra de negocios

Distribuido:

Alquiler/Compra Equipos	44%
Gastos Explotación	31%
Gastos Programación/Análisis	25%

El estudio de la Presidencia de Gobierno distribuye - de otro modo los costes, que en promedio son éstos:

Personal	44,4%
Equipo (con Mtmo.)	42,9%
Servicios Informáticos	
del Exterior	2,8%
Varios	9,9%

Cualquiera que sea la forma de distribuirlos, las empresas experimentan en sus presupuestos unos costes muy elevados y. muy generalmente, no aprecian rendimientos congruentes y sí, en cambio y demasiado a menudo, problemas considerables: tiempo de respuesta elevado, fallos, inadecuación de las soluciones, malestar psicológico, etc.

Los próximos años parece que serán de encrucijada, - presionando por un lado las industrias de ordenadores, electrónica, comunicaciones y de servicios y por los usuarios. Hemos visto cómo Foster predice para el año 2.000 una estructura de ordenador que no se parece nada a la habitual hoy día. Esto es prueba de un diagnóstico de una cierta inadecuación de las actuales. Sin embargo, en lo inmediato se prevé un auge de la teleinformática y de los sistemas organizados sobre una base de datos. La teleinformática favorece a los equipos grandes y a los pequeños y las bases de datos a los grandes.

Por la parte del interés de los usuarios se debe esperar un empuje a la elaboración y difusión de metodologías de - análisis, diseño y programación de sistemas de información y - de explotación. El movimiento de la microinformática, que tiene algunos rasgos dudosos, seguirá, siempre que sea capaz de plas- marse asimismo en metodologías coherentes con sus postulados.

A continuación presentaremos unos conceptos elementales de los sistemas con bases de datos y de la microinformáti-

ca.

### III.2. Sistemas centrados en una base de datos

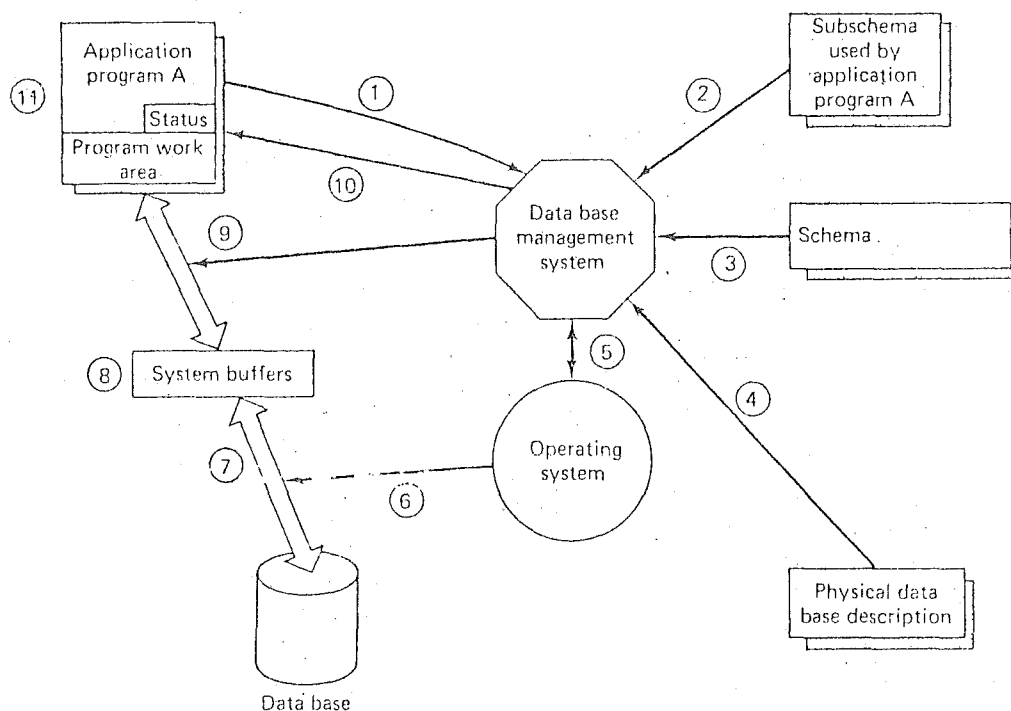
El abaratamiento y aumento de capacidad de memoria on-line con los ordenadores sugiere el empleo de grandes conjuntos organizados de datos, sobre los que se conectarían la mayoría de aplicaciones de una empresa. Es fácil de imaginar que las técnicas de bases de datos y las de transmisión de datos y teleproceso (teleinformática) son complementarias y promueven una información centralizada desconcentrada.

Una base de datos es "una colección de datos interrelacionados almacenados juntos con redundancia tan pequeña como sea posible para servir a una o más aplicaciones en una forma óptima; los datos son almacenados de manera que sean independientes de los programas que los usan; un método común y controlado se usa para adicionar nuevos datos y para modificar y recuperar datos ya existentes en la base". (Martín, 1.975).

Esta técnica acaba generando un nuevo paquete de software llamado sistema de gestión de base de datos. La próxima figura esquematiza los pasos que se suceden cuando un programa de aplicación lee un registro a través de uno de estos sistemas.

1. El programa A genera llamada al SGBD (Sistema de Gestión de la Base de Datos) para leer un registro. El programa da el nombre del programador para el tipo de dato y el valor de la clave del segmento de registro en cuestión.
2. El SGBD obtiene el subesquema (o descripción de datos del programa) que usa el programa A y mira la descripción de los datos cuestionados.



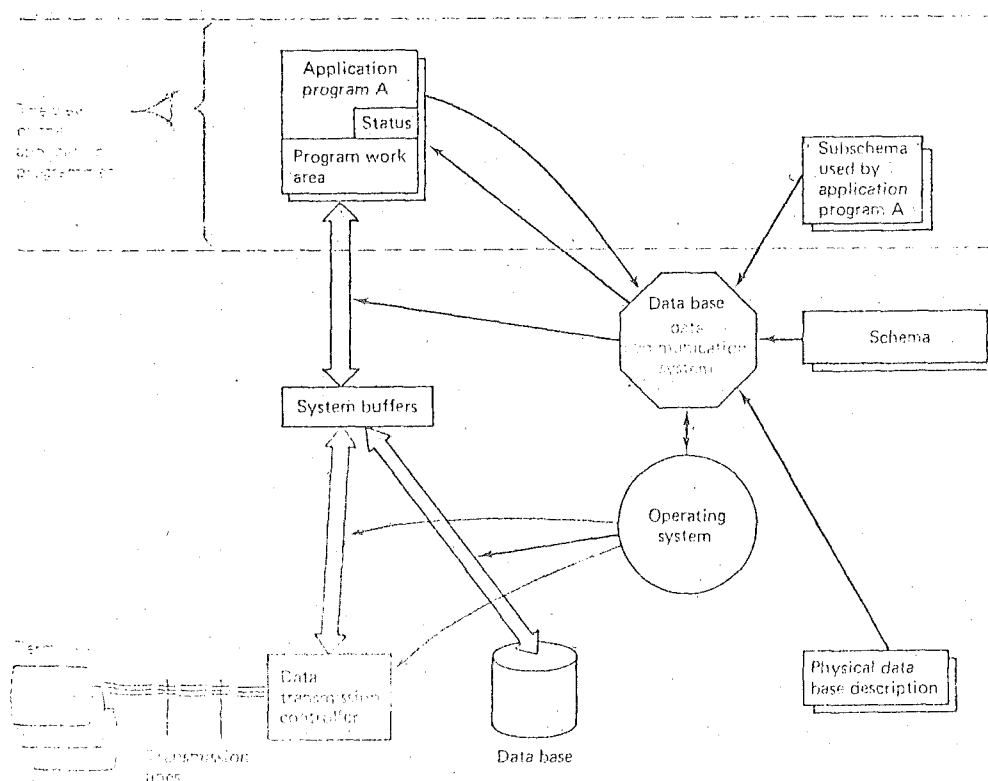


3. El SGBD obtiene el esquema (o descripción lógica global de los datos) y determina qué tipos lógicos de datos se necesitan.
4. El SGBD examina la descripción física de la base y determina qué registros o registros físicos leer.
5. El SGBD genera una orden al sistema operativo indicándole cómo leer el registro requerido.
6. El Sistema Operativo interactúa con el almacenamiento físico donde está el dato.
7. El dato o datos requeridos se transfieren entre el almacenamiento y los buffers.
8. Comparando el subesquema y el esquema, el SGBD deriva de los datos el registro lógico que necesita el pro--

grama. Cualesquiera transformaciones de datos como vienen declarados en el subesquema y los datos como lo vienen en el esquema son realizados por el SGBD.

9. El SGBD transfiere los datos desde los buffers al área de trabajo del programa A.
10. El SGBD proporciona información de estado al programa e indicaciones de error.
11. El programa puede ya operar con los datos en su área de trabajo.

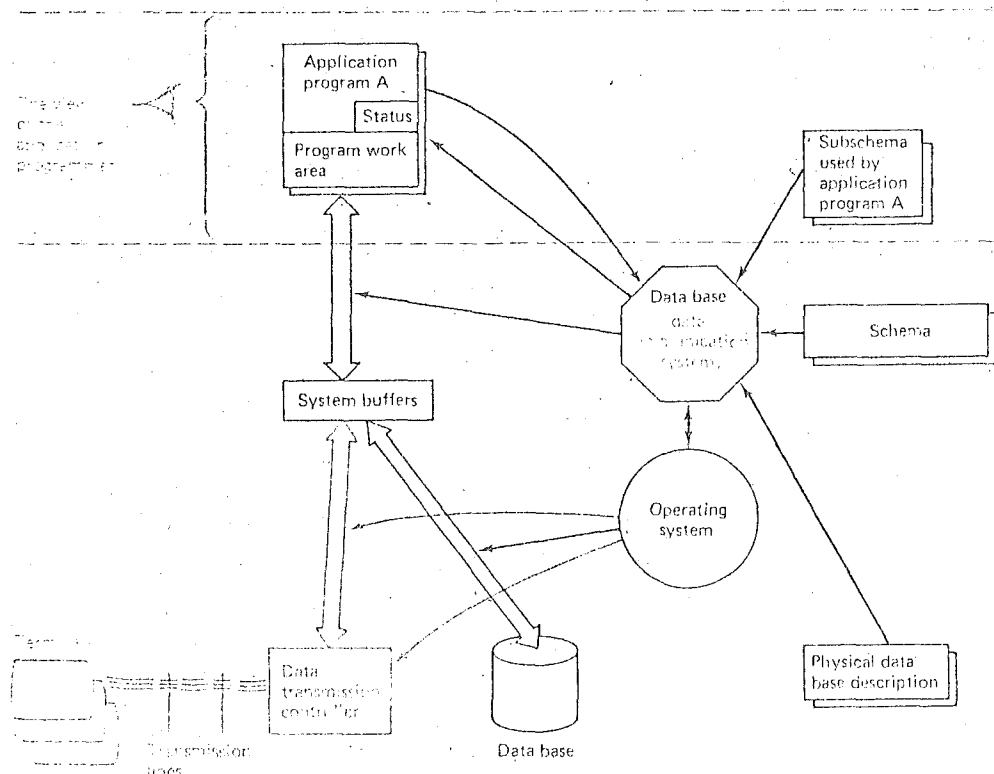
En fin, como se decía antes, los sistemas tienden a usarse en un ambiente de telecomunicación y ello hace que, a veces, se creen sistemas para manejar tanto las operaciones de bases de datos como de transmisión de datos, llamándose en tal caso sistemas DBDC (Data-base data communications, terminología IBM). Ver próxima figura.



grama. Cualesquiera transformaciones de datos como vienen declarados en el subesquema y los datos como lo vienen en el esquema son realizados por el SGBD.

9. El SGBD transfiere los datos desde los buffers al área de trabajo del programa A.
10. El SGBD proporciona información de estado al programa e indicaciones de error.
11. El programa puede ya operar con los datos en su área de trabajo.

En fin, como se decía antes, los sistemas tienden a usarse en un ambiente de telecomunicación y ello hace que, a veces, se creen sistemas para manejar tanto las operaciones de bases de datos como de transmisión de datos, llamándose en tal caso sistemas DBDC (Data-base data communications, terminología IBM). Ver próxima figura.



como puede verse, se parecen bastante a las de los sistemas re  
partidos:

- 1°. Volúmenes pequeños
- 2°. Tiempos de acceso razonables
- 3°. Tratamientos simples
- 4°. Herramientas convivenciales
- 5°. Bajo costo
- 6°. Niveles jerárquicos
- 7°. Cohesión de las células

REFERENCIAS

J.P. Meinadier

Estructura y Funcionamiento de los Computadores digitales  
AC, 1.973

R.W. Wolverton

The Cost of Developing Large-Scale Software  
IEEE Transactions on Computers, Jun. 1.974

T. Gilb

Laws of Unreliability  
Datamation, Marzo 1.975

C.C. Foster

Computer Architecture  
Van Nostrand Reinhold, 1.976

A.S. Philippakis

Programming Language Usage  
Datamation, Oct. 1.973

N. Wirth

Algorithms + Data Structures = Programs  
Prentice Hall, 1.976.

S.S. Husson

Microprogramming: Principles and Practices  
Prentice-Hall, 1.970

T.A. Dolotta et al

Data Processing in 1.980-85. A study of potential limitations  
to Progress  
Wiley, 1.976

Ph. H. Enslow

Multiprocessors and Parallel Processing

Wiley, 1.974

C.G. Bell, R. Chen, S. Rege

Effect of Techonology on near term Computer Structures

Computer, Marzo 1.972

B. Lussato, B. France, J.P. Bouhot

La microinformatique. Introduction aux systèmes répartis

Editions d'Informatique, 1.974

F. Anceau

Le Monde des microprocesseurs.

Rapport de recherche n° 46. Institut National Polytechnique  
de Grenoble, Octobre 1.976.

C.C. Foster

The next three generations

Computer, Marzo 1.972.

La Informática en España 1.976

Servicio Central de Informática. Presidencia de Gobierno, 1.977.

J. Martin

Computer data-base organization

Prentice-Hall, 1.975

G. Bell, A. Newell

Computer Structures: Readings and Examples

McGraw-Hill, 1.971

B.W. Boehm

Software and its impact: a quantitative assessment

Datamation, Mayo 1.973

F. Sáez Vacas

Principios de las memorias de película magnética

Cátedra de Ordenadores, E.T.S.I.T. 1.973 Ref. CO.73.002.01

F. Sáez Vacas

Memoria de la Cátedra de Ordenadores Electrónicos

F. Sáez Vacas, Dic. 1.973

F. Sáez Vacas

Introducción a la arquitectura de ordenadores

ERIA, 1.975

F. Sáez Vacas

Guía para un análisis estructurado de la programación estructu  
rada

INFORPRIM, 1.975