

Tema 1 : Introducción a la estructura de computadores



Índice

1. Objetivos de la asignatura.
2. Niveles de descripción de un computador.
3. Estructura básica de un computador convencional.
4. Evolución histórica: tecnología y arquitectura.
5. Lenguajes de descripción hardware.

1. Objetivos de la asignatura



Arquitectura del Computador:

- Conjuntos de atributos de un computador que son visibles a:
 - El programador en lenguaje máquina
 - El sistema operativo
 - El compilador
- Engloba los siguientes elementos:**
 - Conjunto de registros visibles al programador
 - Conjunto de instrucciones máquina
 - Tipos básicos de datos soportados por las instrucciones máquina
 - Modos de direccionamiento
 - Mecanismos de E/S
- Propiedades**
 - Permanencia con el tiempo / tecnología (portabilidad)
 - Proporciona funcionalidad eficaz a los niveles superiores
 - Permite implementación eficiente en los niveles inferiores

1. Objetivos de la asignatura



Estructura de un computador:

- Organización internas de los componentes hardware de un computador con el objetivo de implementar una arquitectura determinada.
- Una misma arquitectura se puede conseguir con diferentes estructuras, diferenciándose unas de otras fundamentalmente en el coste y rendimiento.

Tecnología de un computador:

- Implementación física de la estructura interna de un computador.
- Una estructura se puede implementar con diferentes tecnologías.

1. Objetivos de la asignatura



Abordar el ***estudio de la organización o estructura interna de un computador.***

Para situar con mayor precisión el objeto de estudio de la asignatura analizaremos los diferentes niveles de descripción que se contemplan en el estudio de un computador digital.

2. Niveles de descripción de un computador



Estrategia: especificación a diferentes niveles de abstracción.

Cada nivel se caracteriza por:

- Unos elementos de entrada que proceden del nivel inmediato inferior.
- Unos elementos de salida destinados al nivel inmediato superior.
- Una metodología de análisis y síntesis de los elementos de salida en términos de los de entrada.

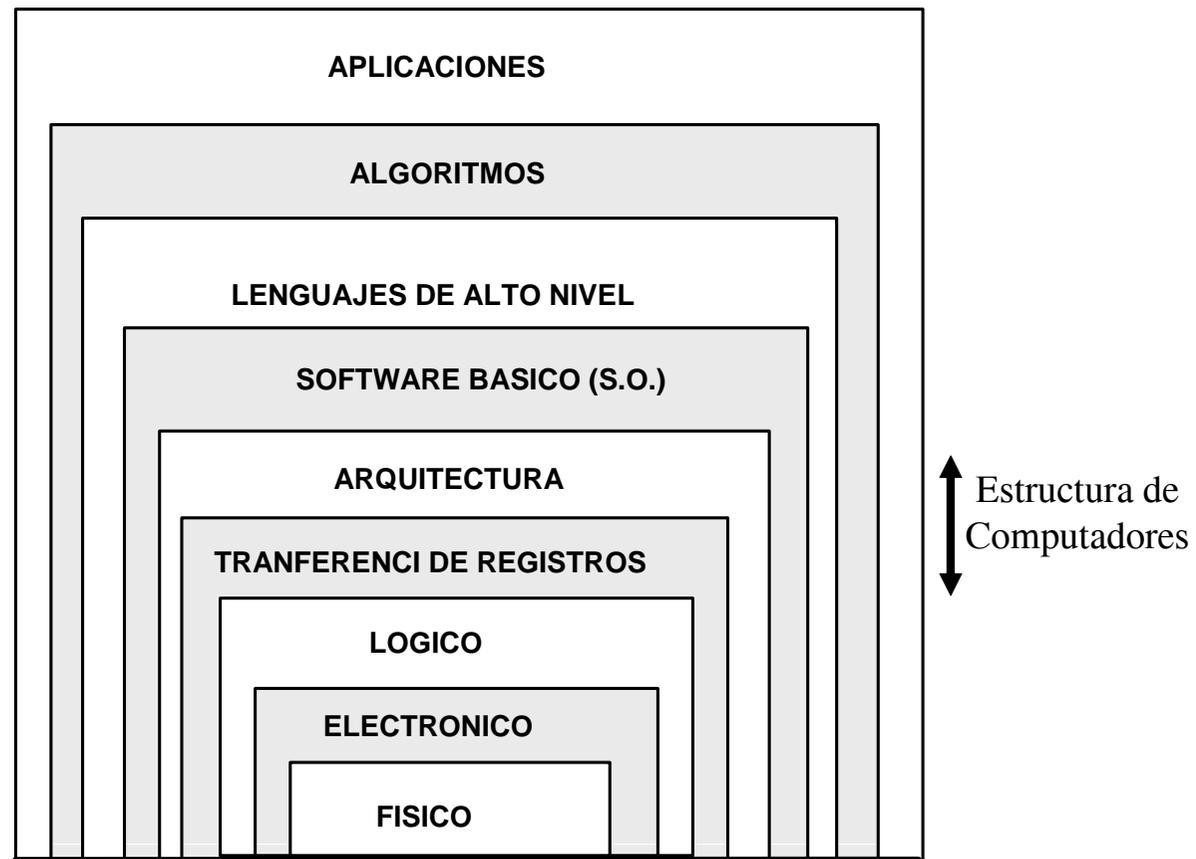
Resultado:

- La complejidad del sistema queda dividida, acotada y organizada en las complejidades parciales de cada nivel
- Dentro de cada nivel se puede aplicar una metodología propia de estudio.

2. Niveles de descripción de un computador



Niveles de abstracción en el estudio de un computador digital



2. Niveles de descripción de un computador



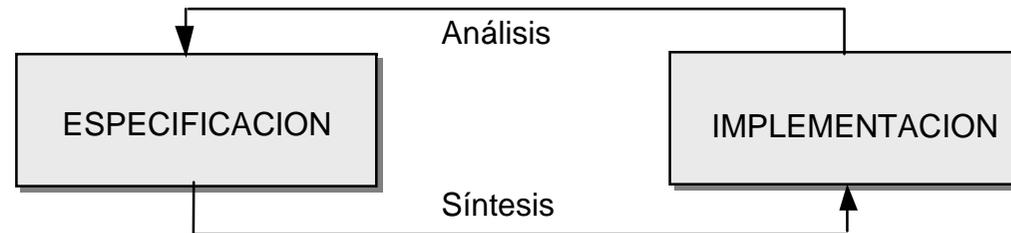
Procesos de estudio: *análisis* y *síntesis*

Análisis

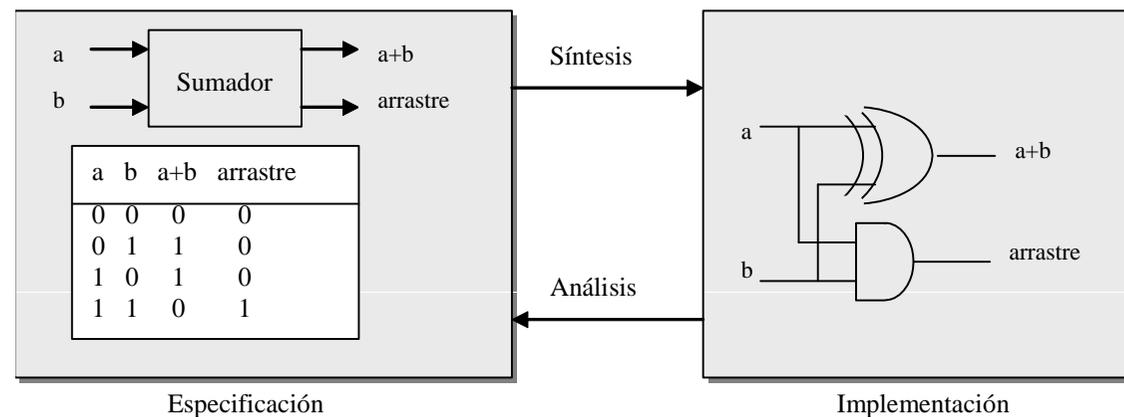
Parte de la implementación del sistema a un nivel en términos de elementos del nivel inferior y determina la función del sistema, es decir, su *especificación*.

Síntesis

Parte de la especificación de un sistema y obtiene su implementación en función de los elementos básicos del nivel inferior.



Ejemplo:

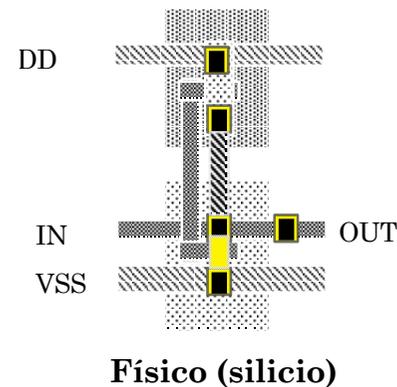


2. Niveles de descripción de un computador



Nivel Físico

Elementos de entrada: formas geométricas que se corresponden con las máscaras de difusión utilizadas en el proceso de fabricación de los circuitos integrados del computador.



Elementos de salida: dispositivos electrónicos: transistores, resistencias, etc.

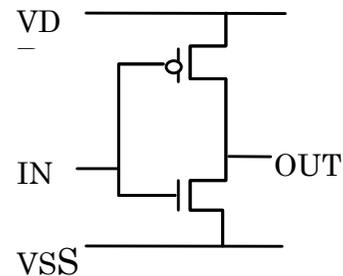
Metodología de estudio: software de manipulación gráfica con restricciones.



2. Niveles de descripción de un computador

Nivel Electrónico

Elementos de entrada: dispositivos electrónicos (resistencias, transistores, etc.)



Circuito eléctrico

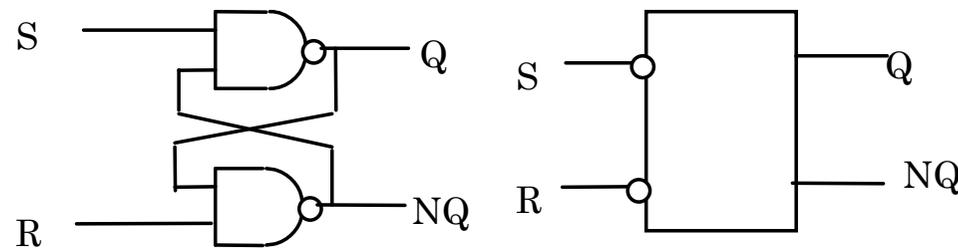
Elementos de salida: biestables y puertas lógicas

Metodología de estudio: técnicas cuantitativas de análisis en el plano eléctrico-temporal, fundamentalmente ecuaciones algebraicas y diferenciales.

2. Niveles de descripción de un computador

Nivel Lógico

Elementos de entrada: biestables y puertas lógicas



Circuito lógico

Elementos de salida: módulos combinacionales y secuenciales tales como multiplexores, codificadores, sumadores, registros, contadores, etc.

Metodología de estudio: dispone de sus propias técnicas de análisis y síntesis.

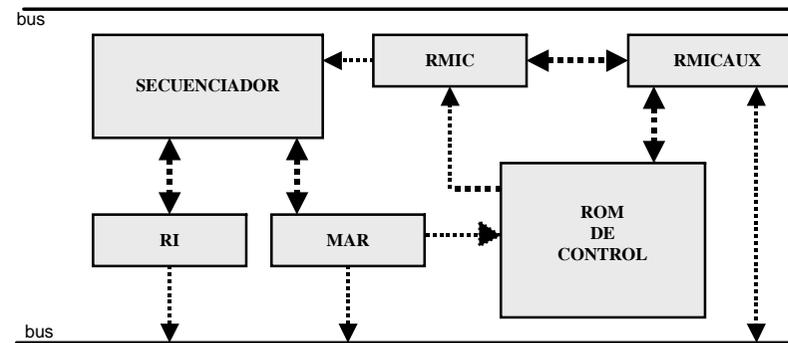
- Circuitos combinacionales: el álgebra de conmutación (un álgebra de Boole).
- Circuitos secuenciales: teoría de las máquinas de estados finitos.

2. Niveles de descripción de un computador



Transferencia de Registros (RT)

Elementos de entrada: registros, módulos combinacionales y elementos de interconexión (buses y/o multiplexores).



Transferencia de registros

Elementos de salida: ruta de datos (construida con los tres tipos de elementos de entrada) + conjunto de transferencias elementales posibles sobre la ruta de datos.

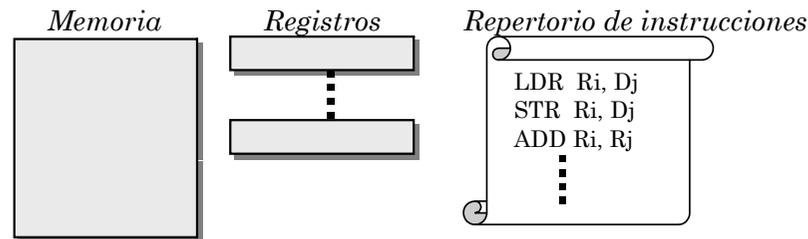
Metodología de estudio: no dispone de una herramienta propia de análisis y síntesis, aunque sí es posible utilizar los modernos lenguajes de descripción hardware (por ejemplo VHDL)



2. Niveles de descripción de un computador

Arquitectura (lenguaje máquina)

Elementos de entrada: transferencias y transformaciones posibles de información en la ruta de datos.



Elementos de salida: lenguaje máquina definido por:

- *Repertorio de instrucciones*: formatos, direccionamiento, etc.
- *Memoria* y conjunto de *registros* referenciables por las instrucciones.

Metodología de estudio: tecnología e ingeniería del software.

Dos planteamientos para el nivel máquina de un computador:

CISC (*Complex Instruction Set Computers*): define un repertorio de instrucciones complejo y numeroso, con muchos tipos de direccionamiento y modos de control.

RISC (*Reduced Instruction Set Computer*): simplifica la complejidad y el número de instrucciones, dejándolo reducido a un conjunto pequeño y rápido

2. Niveles de descripción de un computador



Software básico (sistema operativo)

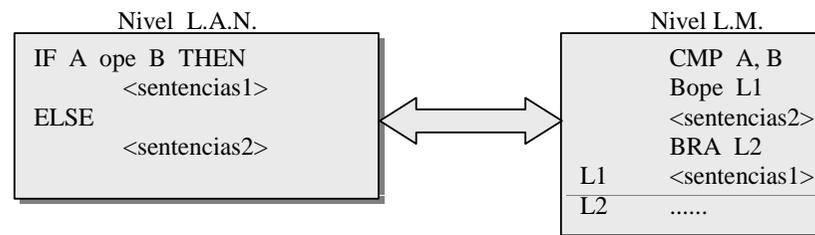
- El Sistema Operativo (SO) no constituye un nivel del mismo tipo que los demás.
 - Se trata de un gestor de determinados recursos del nivel máquina que por la frecuencia y complejidad de uso resulta más eficiente utilizarlos de modo conjunto.
- SO en los primeros computadores
 - las funciones del SO eran escasas, limitadas básicamente a la carga del programa y a la entrada salida.
- SO en las máquinas modernas
 - Las competencias han ido aumentando del que funcionan en entornos multiusuario y multitarea y que requieren una gestión de todos los recursos de la máquina:
 - CPU
 - Jerarquía de memoria,
 - Tratamiento de las excepciones
 - Mecanismos de protección
 - Gestión de archivos, etc.
- SO en un computador actual
 - **Bastante protagonista: cuando se diseña su arquitectura hay que tener en cuenta sus funciones.**



2. Niveles de descripción de un computador

Lenguajes de alto nivel

- En este nivel se utilizan lenguajes de programación con una sintaxis y una semántica más complejas que las del lenguaje ensamblador.
- Facilitan el trabajo del programador aportando recursos expresivos más próximos a los problemas que se van a resolver.
- **Se requiere un proceso de traducción al nivel máquina que es realizado por un programa denominado *compilador*.**



- Existen dos paradigmas de lenguajes de alto nivel:
 - **Lenguajes imperativos** (Pascal, Fortran, C, etc.) con una semántica operacional que obliga al programador a expresar la secuencia de órdenes cuya ejecución resuelve el problema.
 - **Lenguajes declarativos** (Prolog, Miranda o Lisp) con una semántica que permite que el programador tan solo declare las relaciones lógicas o funcionales del problema.

2. Niveles de descripción de un computador



Algoritmos

- En el nivel algorítmico se expresa la resolución de un problema mediante un conjunto de reglas aplicadas de forma sistemática y ordenada.
- Los procedimientos que define un algoritmo son independientes de cualquier lenguaje de programación y de cualquier máquina particular.

Aplicaciones

- Las aplicaciones se corresponden con dominios de actividad que pueden automatizarse con el uso de un computador digital.
- Del análisis del dominio se extraen unas especificaciones funcionales que son expresadas mediante algoritmos.
- Codificados estos algoritmos en un lenguaje de programación y previa compilación, se ejecutan en la máquina.

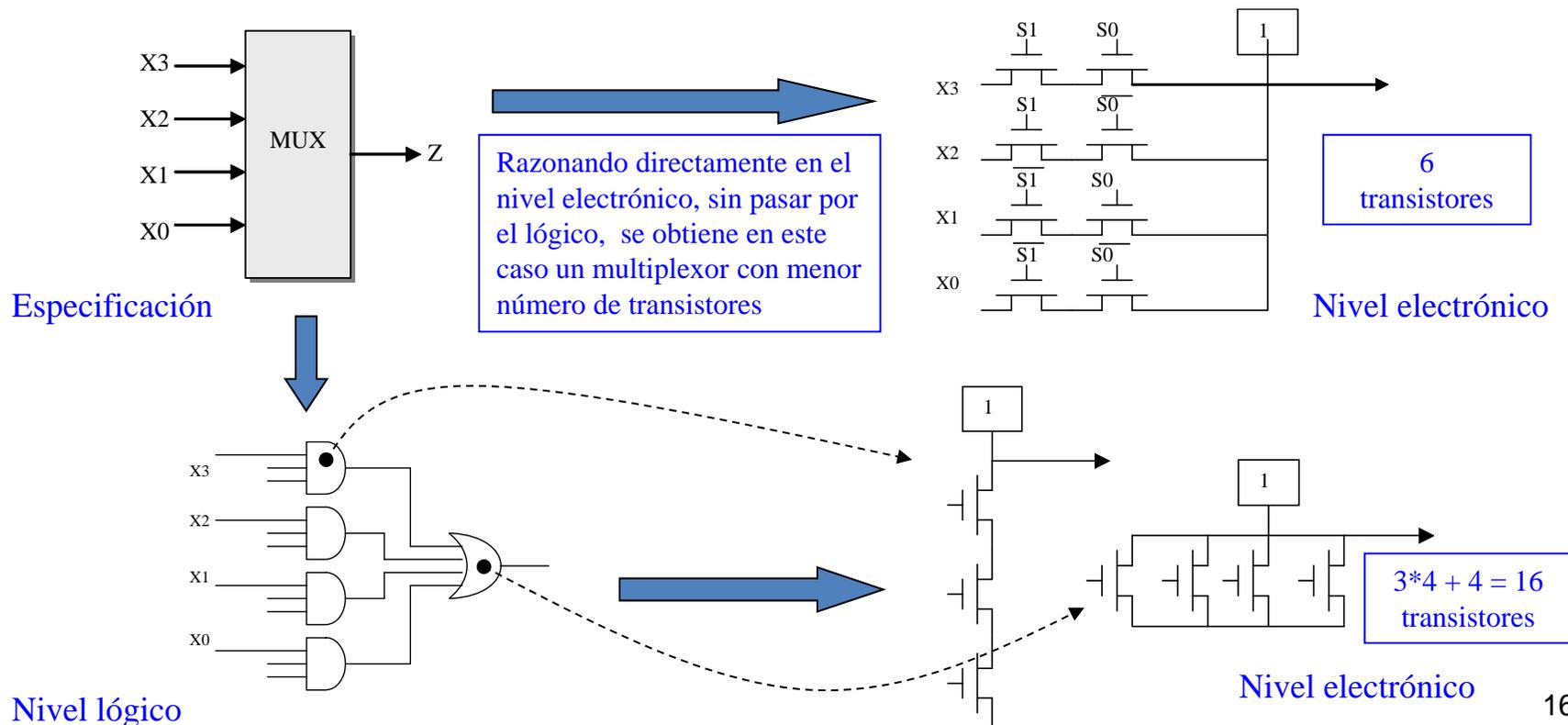
2. Niveles de descripción de un computador



Inconvenientes de la división entre niveles

- Cuando se implementa una especificación deben cumplirse los requerimientos funcionales y optimizar una función de calidad relacionada con la velocidad (*max*) y/o el costo (*min*).
- En ocasiones contemplar tan solo los niveles frontera de un nivel en el que se plantea un problema de diseño puede dar lugar a la imposibilidad de optimizar la implementación.

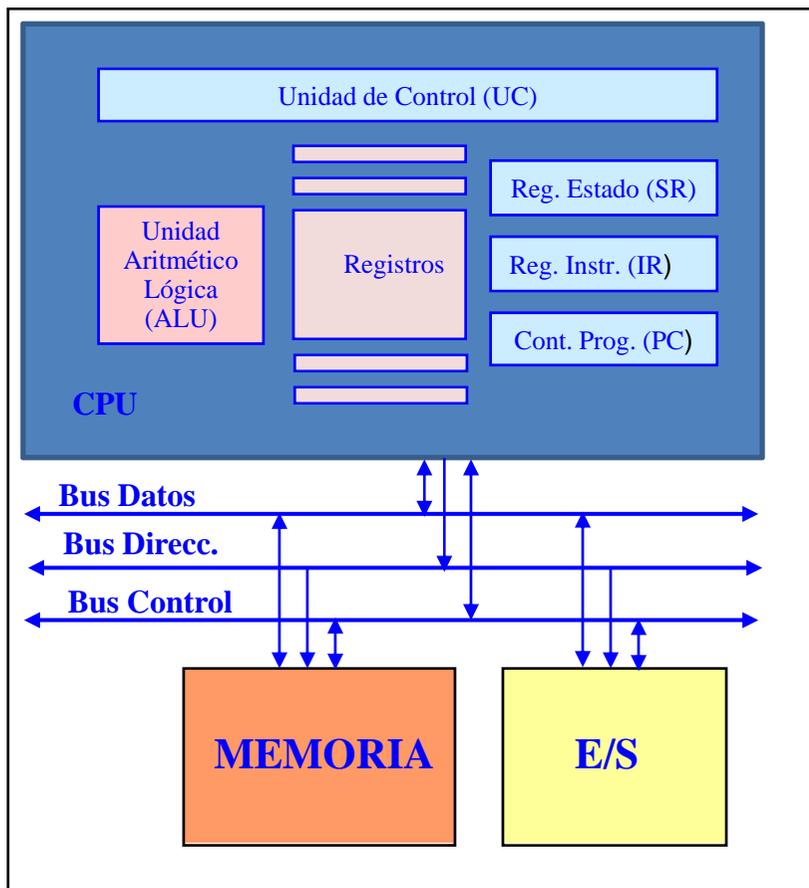
Ejemplo: diseño de un multiplexor con conmutadores bidireccionales





3. Estructura básica de un computador convencional

La estructura básica de un computador actual sigue siendo la original de von Neumann:



CPU (Unidad Central de Proceso)

- Realiza la ejecución de las instrucciones

Unidad de Memoria

- Almacena las instrucciones y los datos

Unidad de E/S

- Transfiere información entre el computador y los dispositivos periféricos

Bus de datos

- Transfiere datos entre CPU, memoria y E/S

Bus de direcciones

- Especifica la dirección de memoria o la dirección del registro de E/S

Bus de control

- Controlan las transferencias (reloj, etc.)

3. Estructura básica de un computador convencional



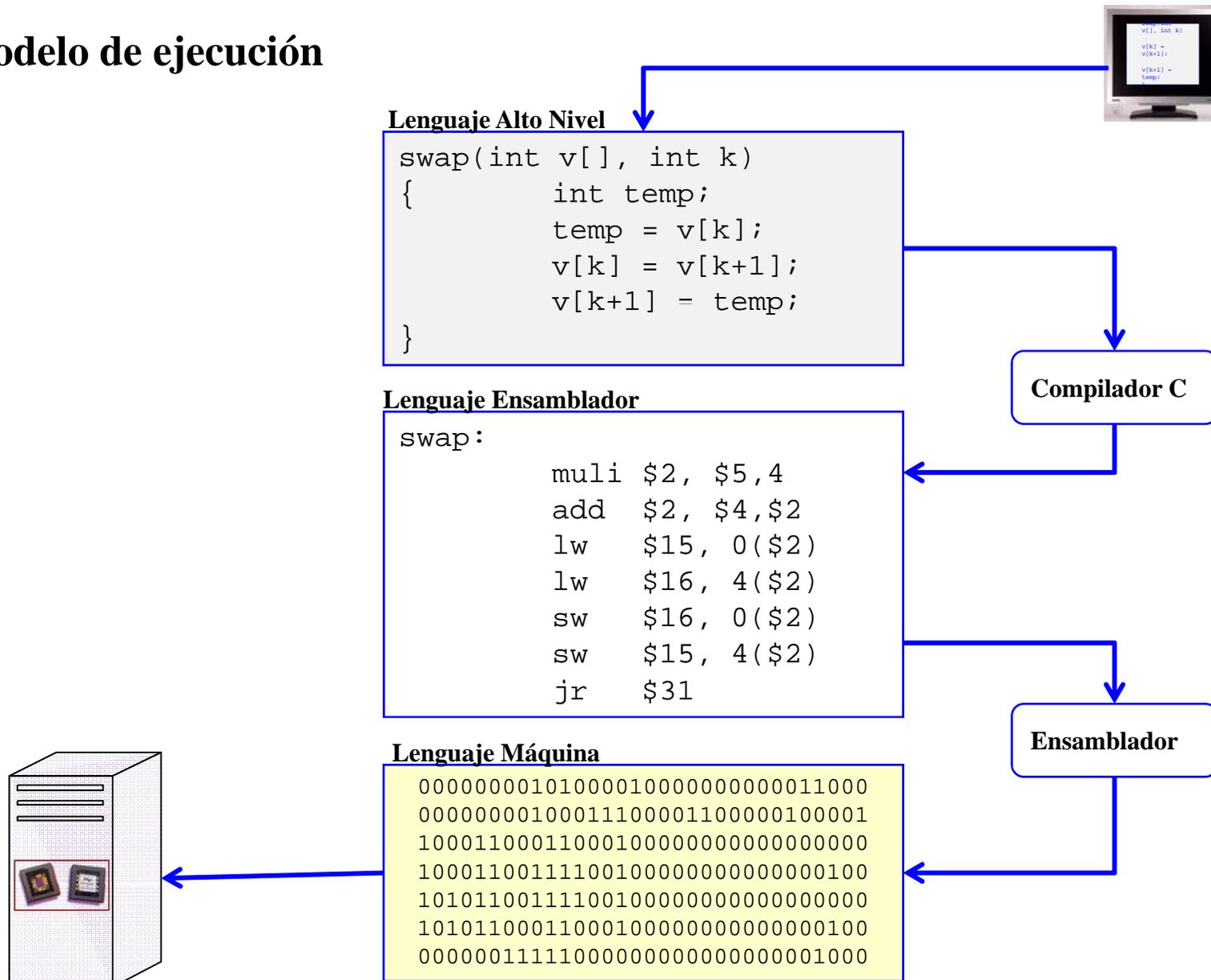
Principios de funcionamiento de la **arquitectura von Neumann**

- Se basa en el concepto de **programa almacenado** en memoria. La memoria almacena:
 - Instrucciones: programa que controla el funcionamiento del computador.
 - Datos: datos que procesa y genera dicho programa.
- Las palabras en memoria siguen una **organización lineal**
- La **ejecución secuencial** de las instrucciones
 - El secuenciamiento de las instrucciones es implícito, y viene determinado por el orden en que han sido almacenadas en la memoria.
 - Este secuenciamiento sólo puede ser modificado por instrucciones de salto.
 - El *contador de programa* indica en cada instante la siguiente instrucción a ejecutar.
- Las fases de ejecución de una instrucción son:
 - **Búsqueda** de la instrucción en memoria y cálculo de la instrucción siguiente
 - **Descodificación** de la instrucción por parte de la CPU
 - **Búsqueda de los operandos**
 - **Ejecución**
 - **Escritura** del resultado

3. Estructura básica de un computador convencional



Modelo de ejecución

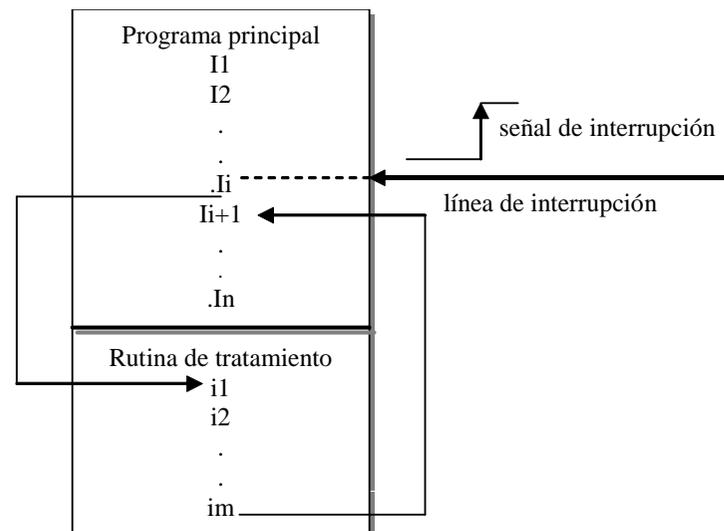




3. Estructura básica de un computador convencional

Aportaciones sobre la arquitectura von Neumann: **Sistema de interrupciones**

- Produce la interrupción de un programa en ejecución cuando aparece una señal externa a la máquina.
- El sistema de interrupciones permite una mejor sincronización de la Entrad/Salida con el exterior .
- Posibilita compartir la CPU por más de un programa.

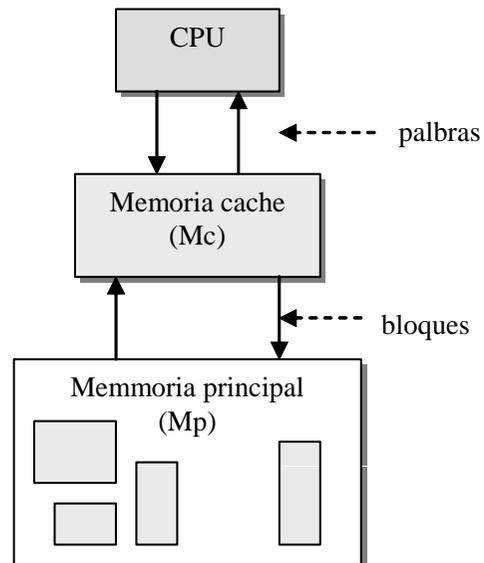




3. Estructura básica de un computador convencional

Aportaciones sobre la arquitectura von Neumann: **Sistema de memoria caché**

- **Permite disminuir el tiempo de acceso a la memoria principal (Mp)** ubicando una memoria de menor tamaño y mayor velocidad (memoria cache, Mc) entre la CPU y Mp
- El sistema explota la localidad de referencia de los programas haciendo que Mc contenga en cada momento los bloques de Mp más referenciados, y evitando así que la CPU tenga que acceder a Mp

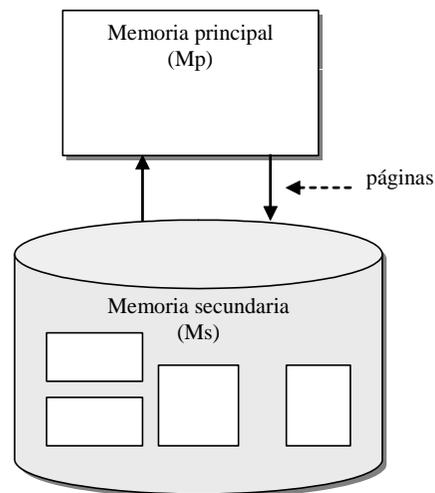




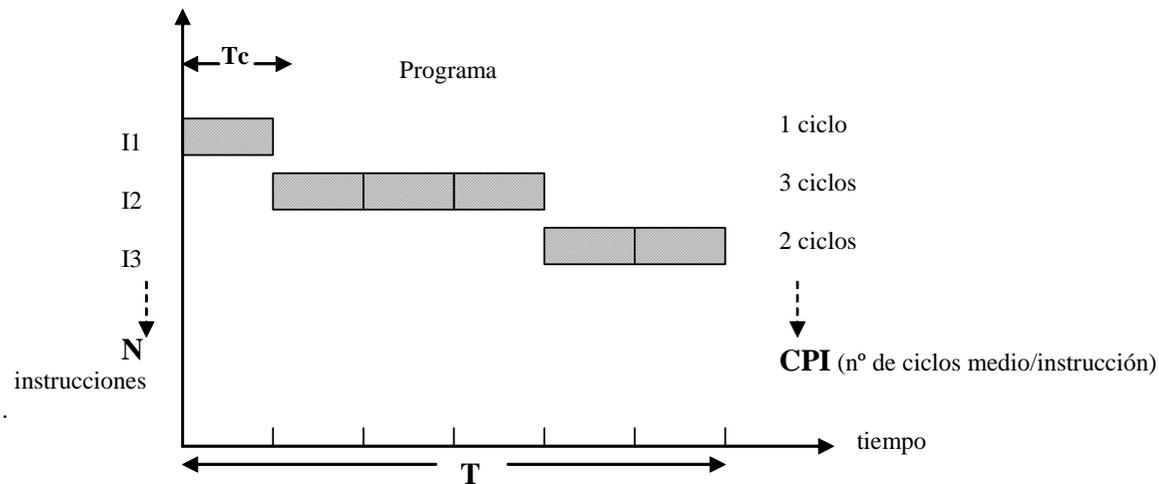
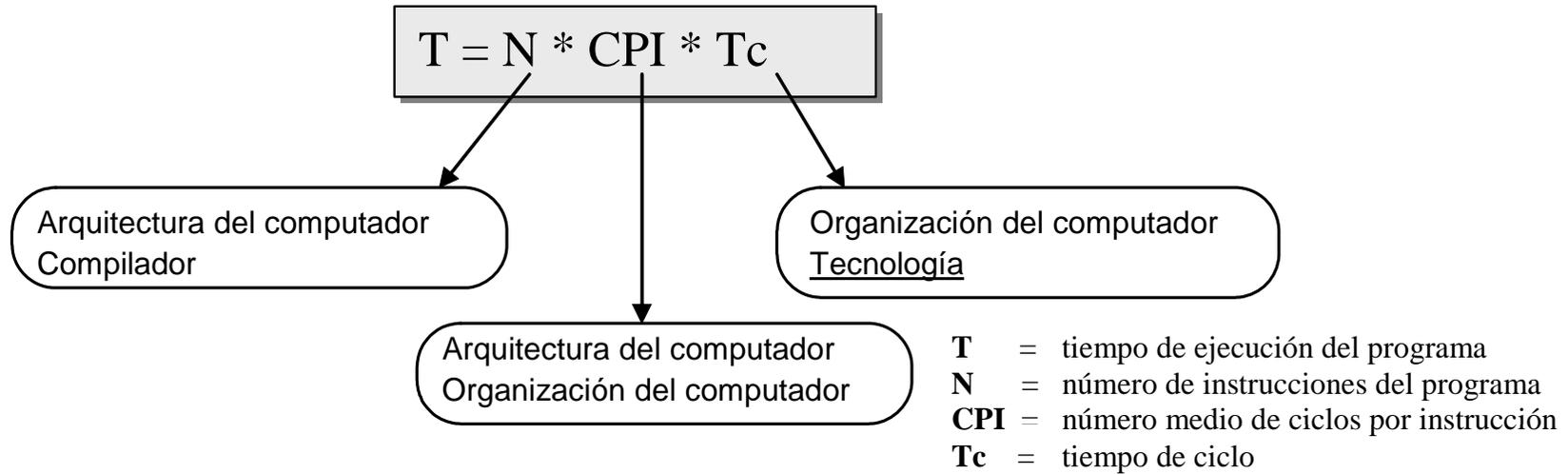
3. Estructura básica de un computador convencional

Aportaciones sobre la arquitectura von Neumann: **Sistema de memoria virtual**

- **Permite la ejecución de programas cuyo tamaño supere el de la Mp**
- Para ello el sistema mantiene en Mp sólo el conjunto de páginas activas (con mayor probabilidad de ser referenciadas) del programa en ejecución
- Las restantes páginas residen en la memoria secundaria, hasta que son referenciadas, en cuyo caso el sistema las activa llevándolas a Mp



4. Evolución histórica: tecnología, estructura y arquitectura



4. Evolución histórica: tecnología, estructura y arquitectura



Generaciones de computadores

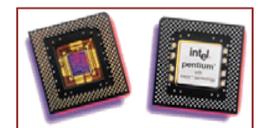
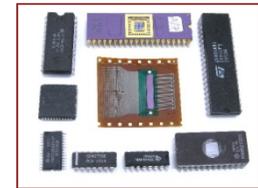
- Primera generación (1946-1957): Válvulas de vacío
 - ENIAC, UNIVAC I y II (Eckert & Mauchly), IBM Serie 700
 - Modelo Von Neumann, programación en lenguaje máquina y ensamblador

- Segunda generación (1958-1964): Transistores
 - DEC PDP-1, IBM Serie 7000, UNIVAC 1100
 - Lenguajes de alto nivel (Fortran, Cobol, ...), canales de E/S, m. virtual, interrupciones

- Tercera generación (1965-1971): Circuitos integrados
 - DEC PDP-8, IBM Series 360 y 370, UNIVAC 1108, CDC series 6600, 7800 y Cyber
 - Microprogramación, caché, DMA, lenguajes (Basic, Pascal), SO (MVS, VMS)

- Cuarta generación (1972-1988): Microprocesador
 - Primer microprocesador: Intel 4004: 4 bits, 2250 transistores en un chip (1971)

- Quinta generación (1988-...): Sistemas basados en micro



4. Evolución histórica: tecnología, estructura y arquitectura



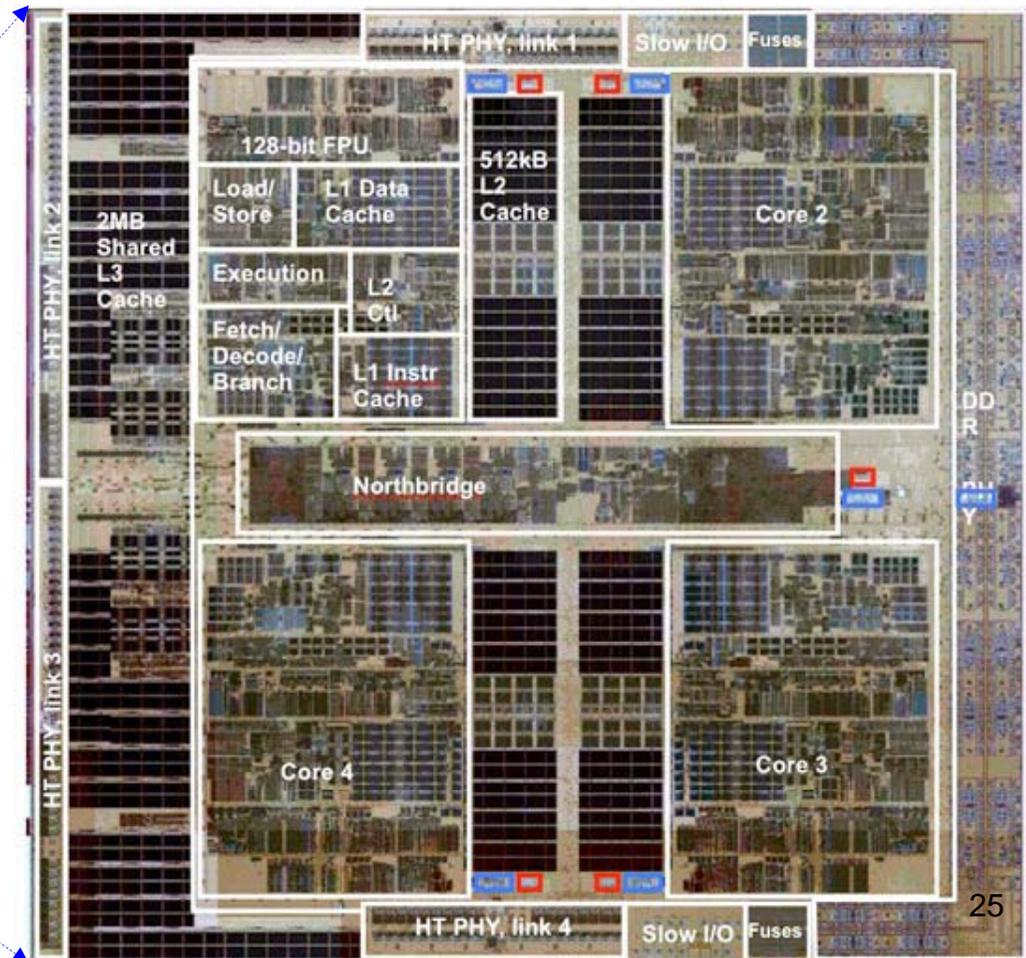
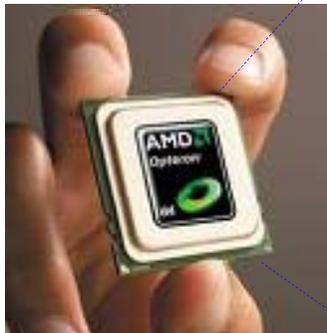
Ley Moore

- Incremento del rendimiento: x 1.54 anual
- Incremento en la frecuencia del reloj: x 1.25 anual

Ejemplo de procesador actual

AMD Quad-Core (K10)

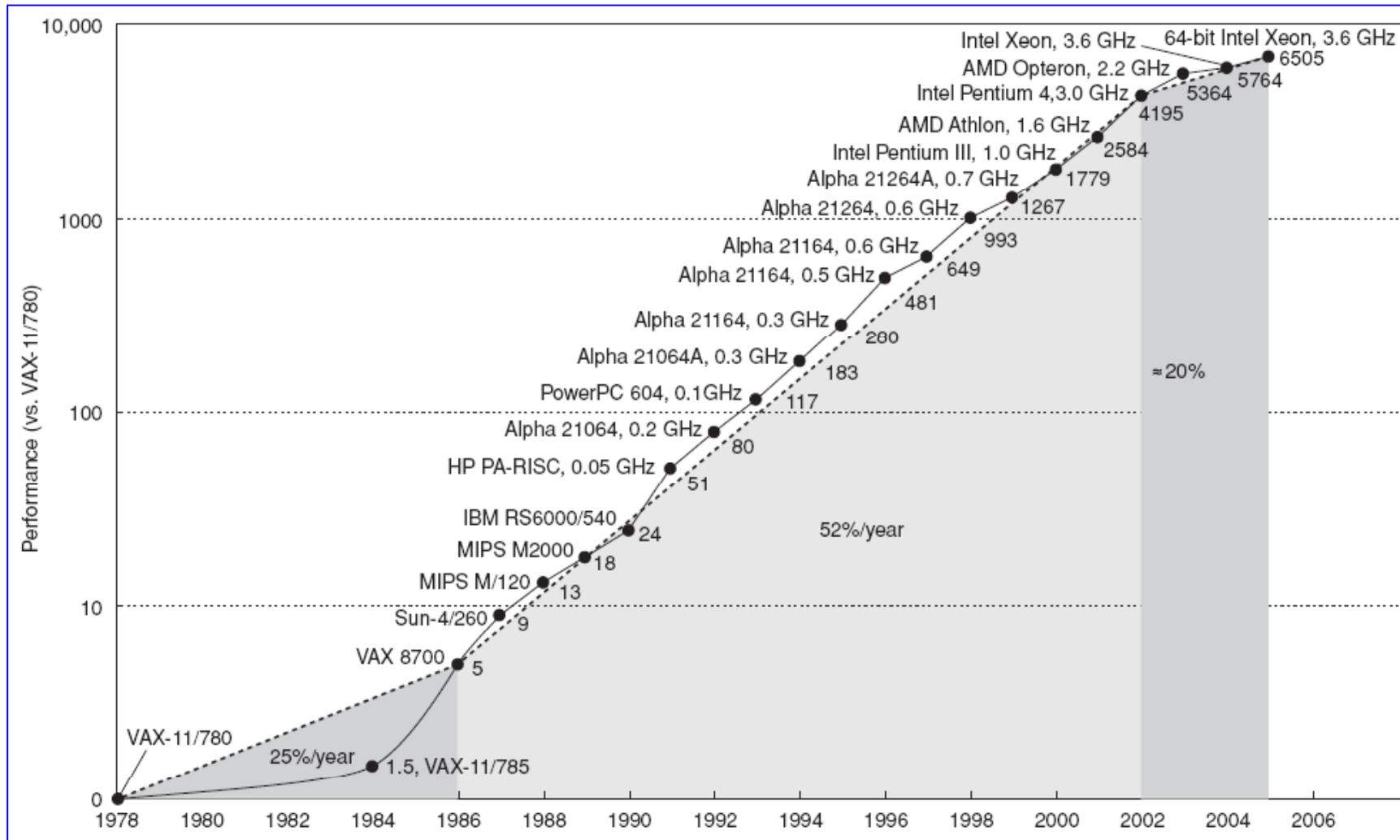
- Tecnología 65nm
- Tres niveles de caché:
 - L1: 256KB
 - L2: 512KB
 - L3: 2MB



4. Evolución histórica: tecnología, estructura y arquitectura



Crecimiento del rendimiento de los procesadores desde 1978 medido en SPECint



4. Evolución histórica: tecnología, estructura y arquitectura



Aumento de rendimiento a través de la organización y arquitectura

Paralelismo y Segmentación

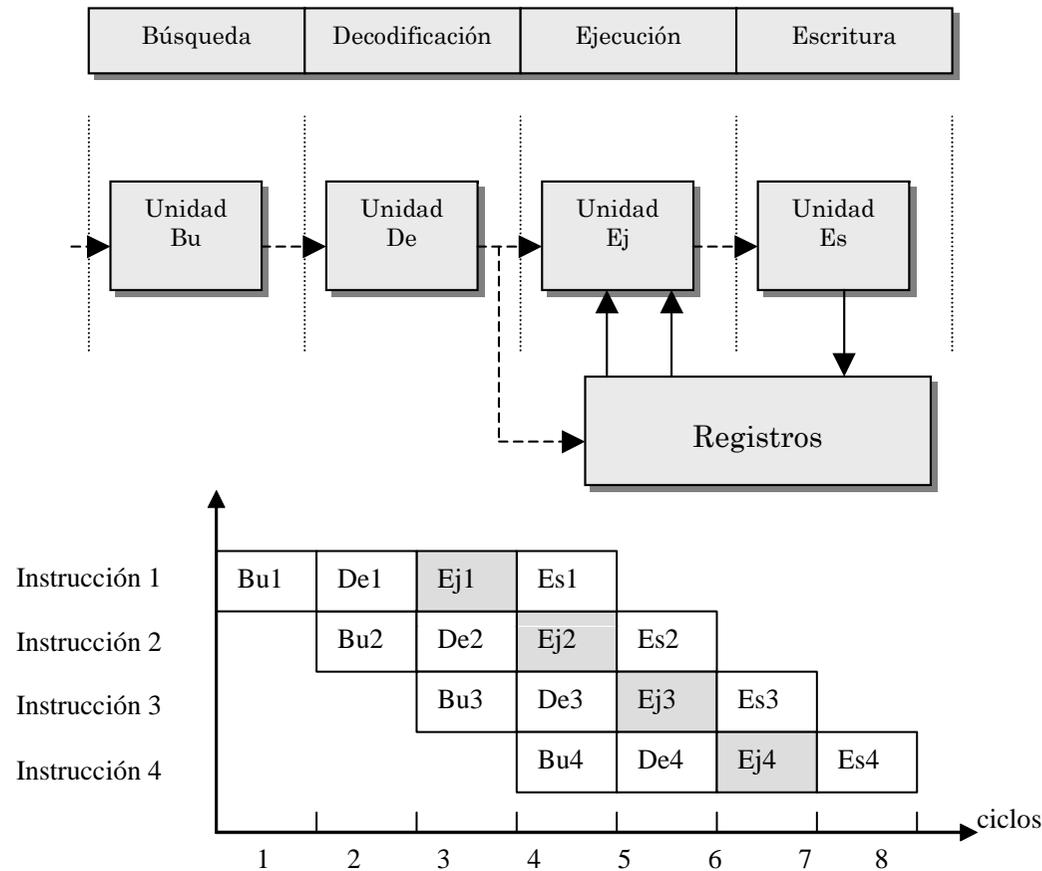
- Las organizaciones y arquitecturas paralelas consiguen que en ciertos instantes de tiempo el computador procese **simultáneamente** más de una operación básica.
- La simultaneidad temporal se consigue fundamentalmente con dos técnicas: el paralelismo y la segmentación.
- **Paralelismo:** ejecuta simultáneamente varias operaciones independientes replicando el número de operadores hardware.
- **Segmentación:** descompone el operador y la operación correspondiente en etapas secuenciales y autónomas, de manera que simultáneamente se puedan ejecutar etapas diferentes de varias operaciones.
- Ambas técnicas se consideran como dos formas del paralelismo: el **paralelismo espacial** o replicación la primera, y el **paralelismo temporal** la segunda.

4. Evolución histórica: tecnología, estructura y arquitectura



Procesadores Segmentados

- Aplican el principio de la fabricación en cadena a la ejecución de instrucciones
- Pueden conseguir un aumento de rendimiento igual al número de fases utilizadas



Limitaciones:

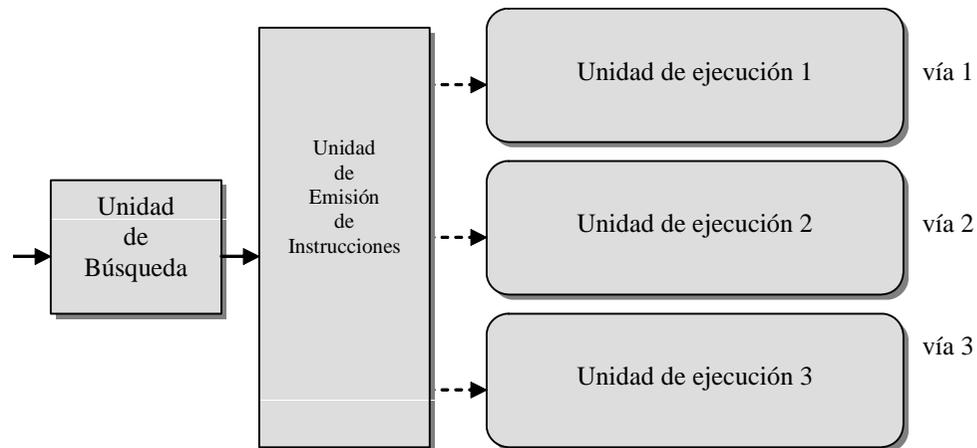
- Dependencias de datos
- Dependencias de recursos
- Bifurcaciones
- Interrupciones

4. Evolución histórica: tecnología, estructura y arquitectura



Procesadores Superescalares

- Un *procesador superescalar* de grado m emite m instrucciones por ciclo.
- **Dispone de m cauces segmentados** operando concurrentemente, si bien en algunas etapas los cauces pueden compartir algunas unidades funcionales.
- Los **conflictos por dependencias** de datos, de control y estructurales de los segmentados siguen existiendo en los superescalares con mayor complejidad.
- Las máquinas superescalares proporcionan **compatibilidad** a nivel del código objeto con las máquinas escalares

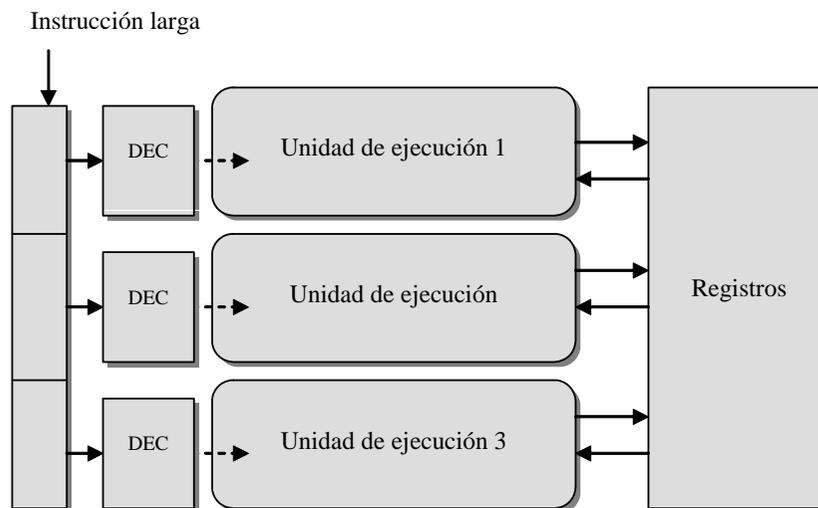


Superescalar de 4 vías y de 64 bits

4. Evolución histórica: tecnología, estructura y arquitectura

Procesadores VLIW

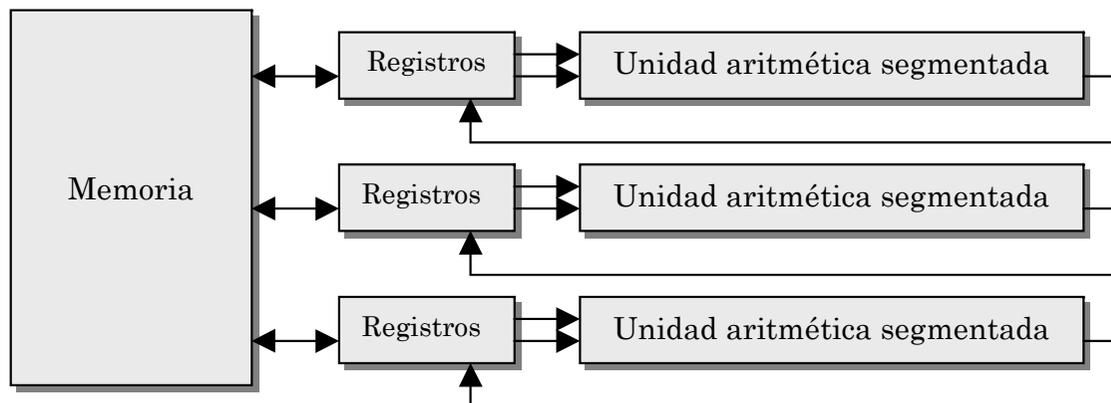
- En un procesador VLIW (*Very Long Instruction Word*) una **única instrucción especifica más de una operación** concurrente.
- Los procesadores VLIW extendieron el concepto de **microcodificación horizontal** utilizado en procesadores dedicados al procesamiento de señales digitales.
- Extraen el paralelismo de **grano fino** de un amplio rango de aplicaciones científicas y de propósito general utilizando técnicas avanzadas de compilación.
- Los **conflictos por dependencias** de datos y estructurales se resuelven antes de la ejecución, y son explícitamente controlados por las instrucciones.



4. Evolución histórica: tecnología, estructura y arquitectura

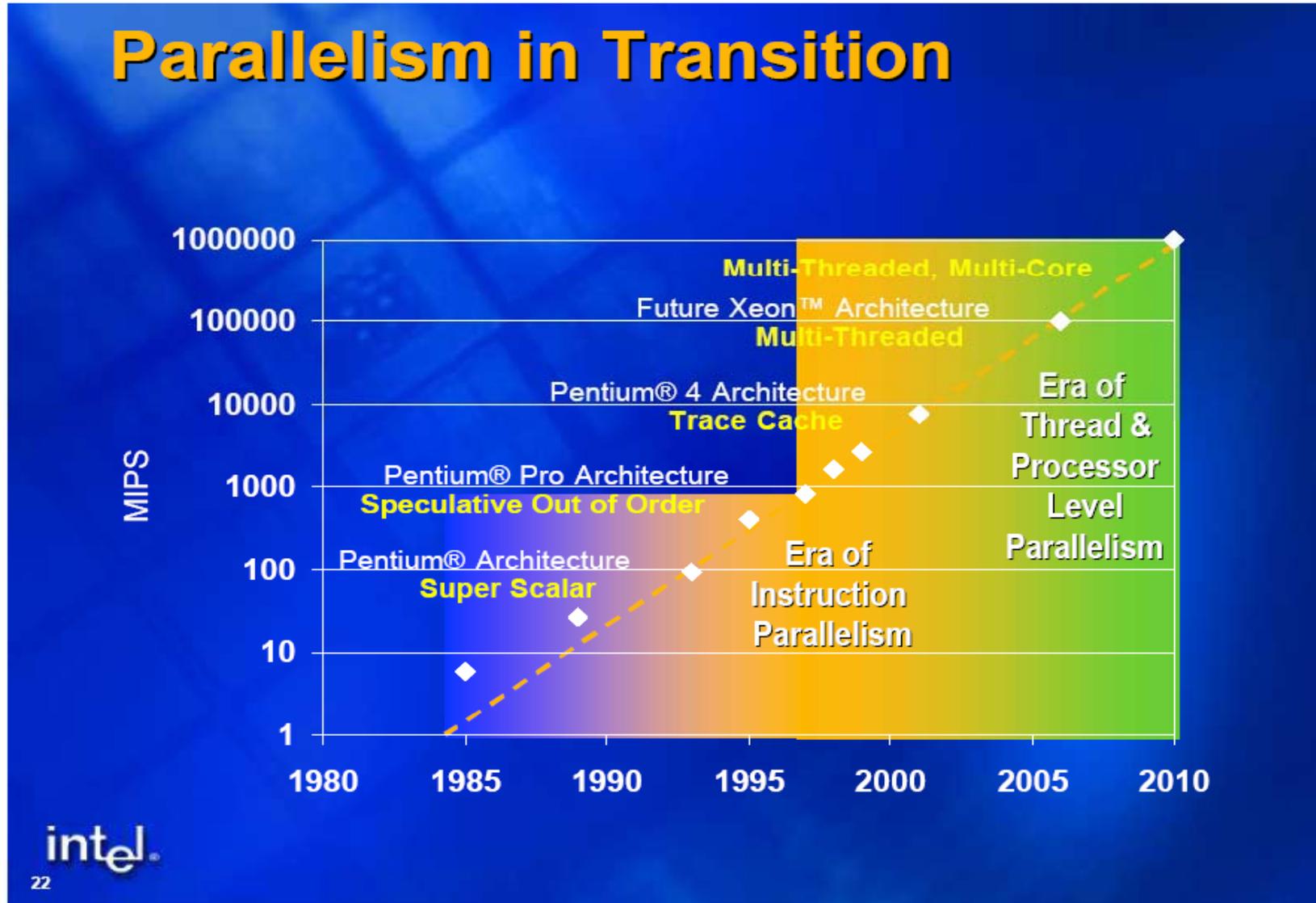
Procesadores Vectoriales

- Los procesadores vectoriales disponen de instrucciones que **operan sobre vectores**.
- Una instrucción vectorial es **equivalente a un bucle completo**.
- Las instrucciones vectoriales **reducen el ancho de banda** necesario para su lectura en comparación con las instrucciones escalares equivalentes.
- Desde el punto de vista arquitectónico **son procesadores segmentados** con instrucciones máquina vectoriales.
- **No existe dependencias entre las operaciones** de una instrucción vectorial
→ se explota eficientemente la segmentación en las unidades aritméticas.



Cray X-MP

4. Evolución histórica: tecnología, estructura y arquitectura



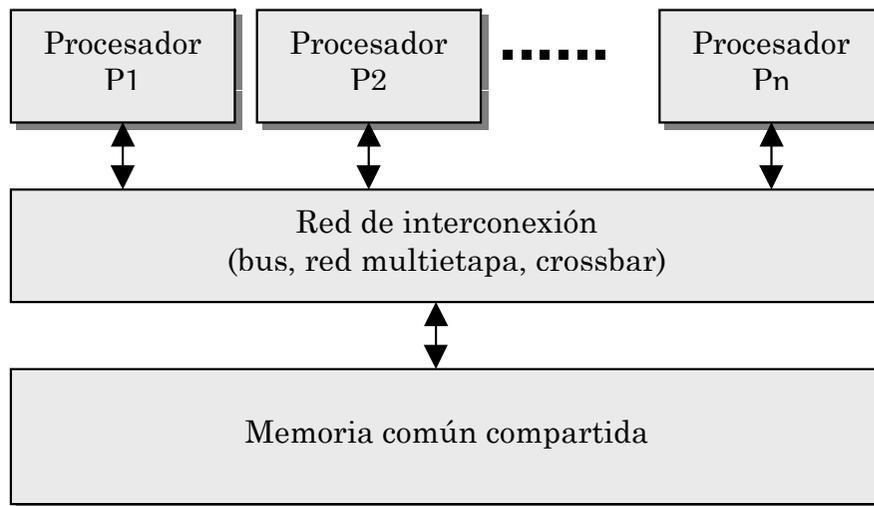
an International Academic Forum, April 20-21, 2014 in Barcelona, an Intel Innovation Acceleration Event



4. Evolución histórica: tecnología, estructura y arquitectura

Multiprocesadores de Memoria Compartida

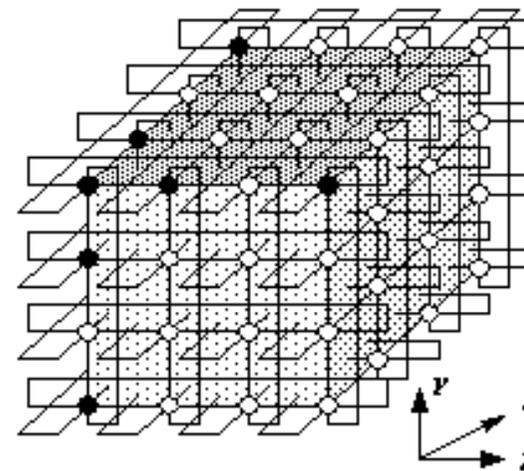
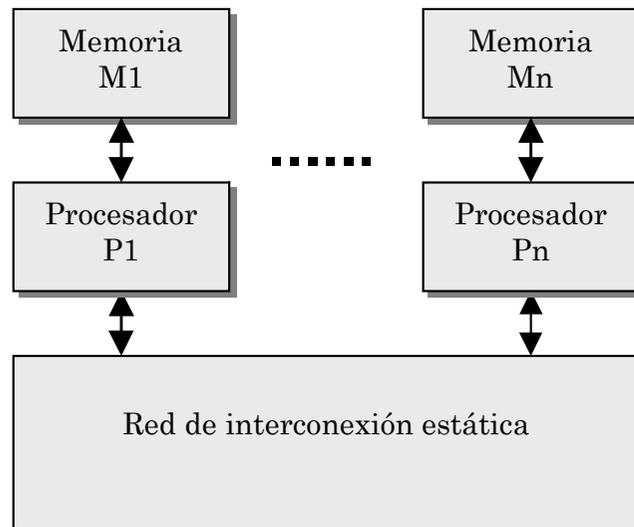
- Son arquitecturas compuestas por un **conjunto de procesadores** que acceden a una **única memoria común** a través de una red de interconexión.
- Utilizan memorias **caché locales** para las que hay que resolver el problema de su coherencia con respecto a la memoria principal y entre sí.
- La **sincronización** de los diferentes procesadores cuando participan en una tarea común se resuelve con instrucciones del tipo TEST&SET, TEST&AND, etc.



4. Evolución histórica: tecnología, estructura y arquitectura

Multicomputadores

- Son multiprocesadores de **memoria distribuida** donde cada procesador tiene un **espacio privado de direcciones**.
- Se comunican y sincronizan mediante **paso de mensajes** a través de una red de interconexión.
- Las **topologías de red** más utilizadas son la malla y el hipercubo.

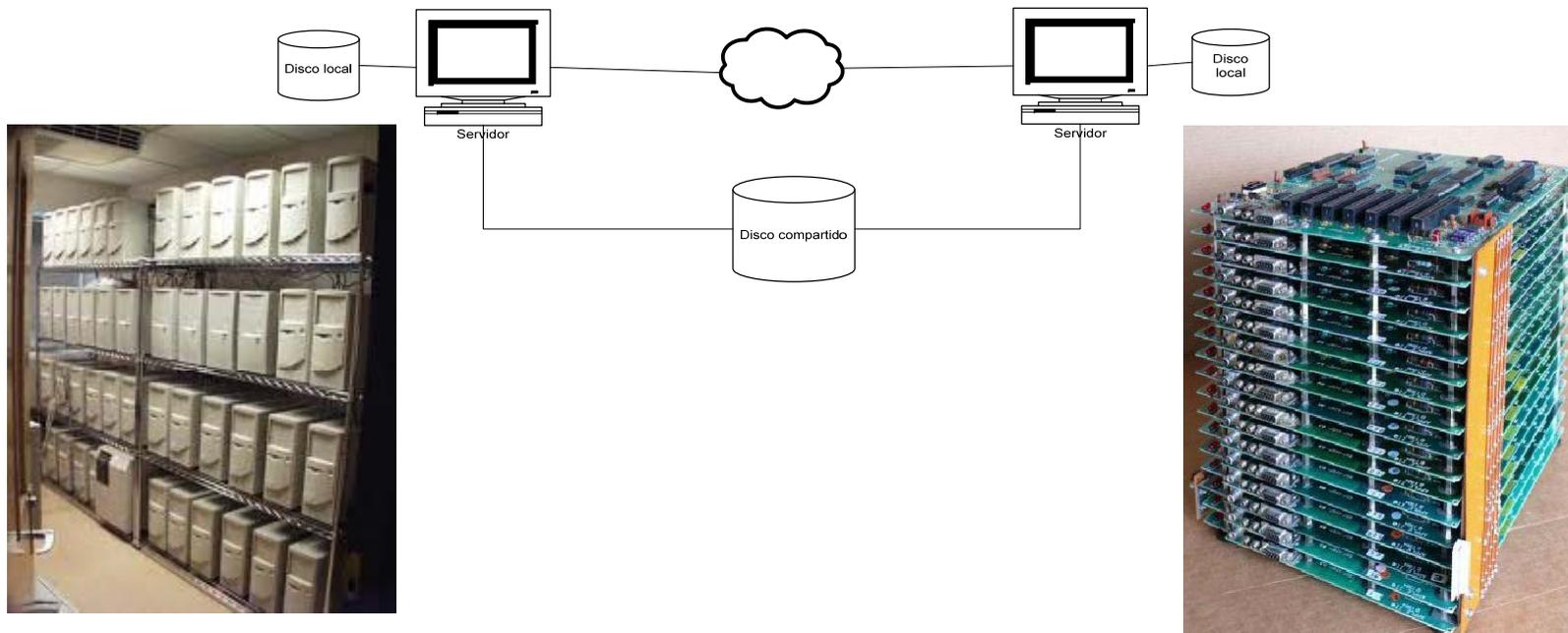


Cray T3E

4. Evolución histórica: tecnología, estructura y arquitectura

Clusters de computadores

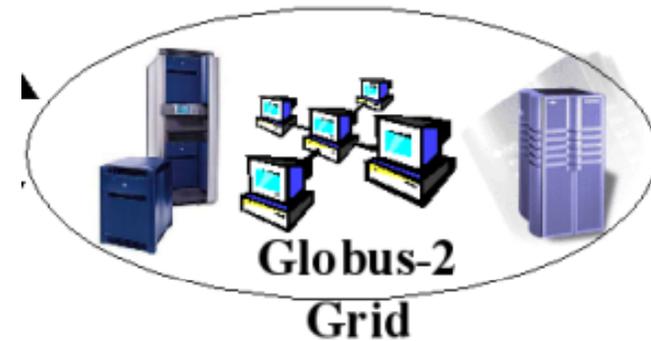
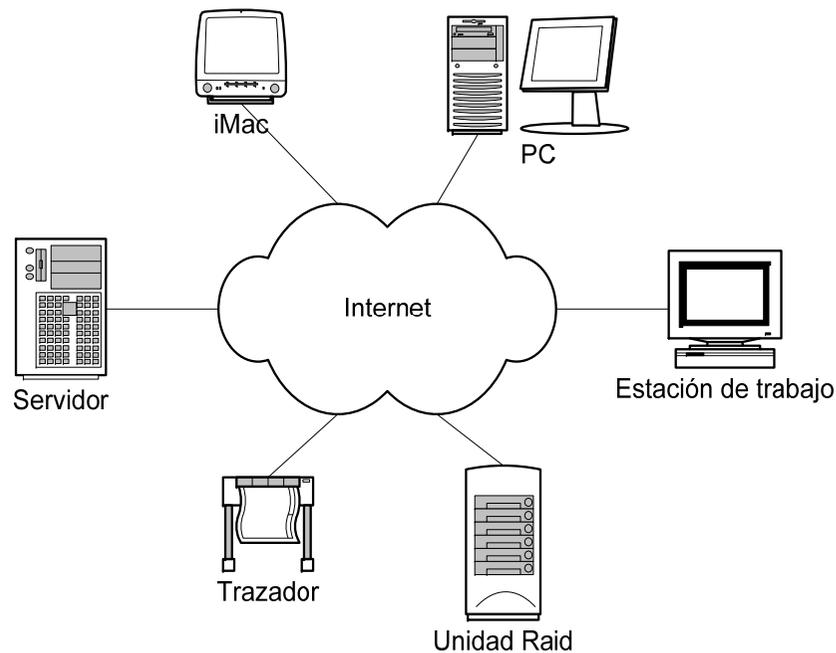
- Un cluster es un conjunto de computadores independientes que comparten una red de interconexión rápida y unidades de almacenamiento.
- Ejecutan una serie de aplicaciones de forma conjunta y aparecen ante clientes y aplicaciones como un solo sistema.
- Los clusters permiten aumentar la escalabilidad, disponibilidad y fiabilidad de los servidores.



4. Evolución histórica: tecnología, estructura y arquitectura

Computación en red

- Se coordinan un conjunto de computadores a través de una red para realizar una tarea
- La red puede ser local o global (Internet)
- Se organizan como servicios y se pueden distinguir entre dos planteamientos:
 - Servicios Grid
 - Servicios Web (XML, WSDL, SOAP, UDDI)
- En la actualidad se da una convergencia entre ambos planteamientos



4. Evolución histórica: tecnología, estructura y arquitectura

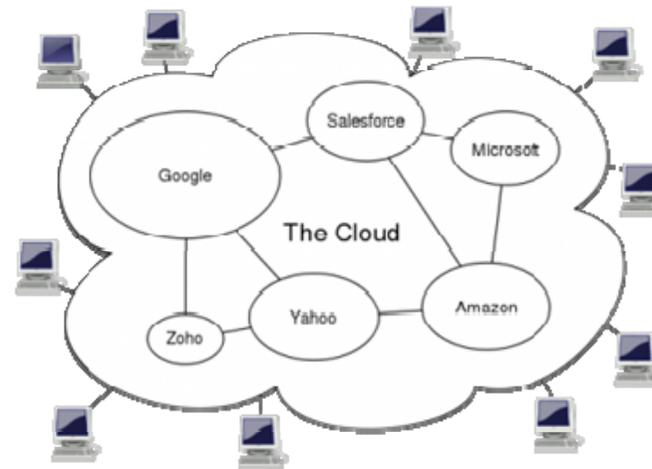
Computación en nube (Cloud computing)

- Sistema informático basado en Internet y centros de datos remotos.
- Gestiona servicios de información y aplicaciones.
- No necesita instalar las aplicaciones en la computadora.
- Solo necesita acceso a Internet.
- Ofrece un uso más eficiente de recursos: memoria, CPU, ancho de banda, etc.
- Se utilizan sólo los recursos necesarios en cada momento.



4. Evolución histórica: tecnología, estructura y arquitectura

Computación en nube: **Centros de Datos** (Data Centers)



5. Lenguajes de descripción hardware (HDL)



- Son **lenguajes de alto nivel** con sintaxis similar a los de programación (C, ADA, Pascal, ...)
- Su semántica permite el **modelado y simulación** de dispositivos hardware a diferentes niveles
- Los **primeros HDLs**
 - Sólo pretendían servir de **vehículo de comunicación** del diseño
 - Se trataba de **formalismos de especificación** de dispositivos hardware
 - Fueron **desarrollados por instituciones universitarias** o por la industria electrónica
 - Alcanzaron **escasa difusión**
- Los **actuales HDLs**
 - Han adquirido un alto grado de **estandarización**
 - Han adoptado los nuevos conceptos de la **ingeniería software**
 - Permiten la **verificación** de una **especificación** del diseño mediante **simulación**
 - Se utilizan como vehículo de **entrada a muchas herramientas de diseño** automático

5. Lenguajes de descripción hardware (HDL)



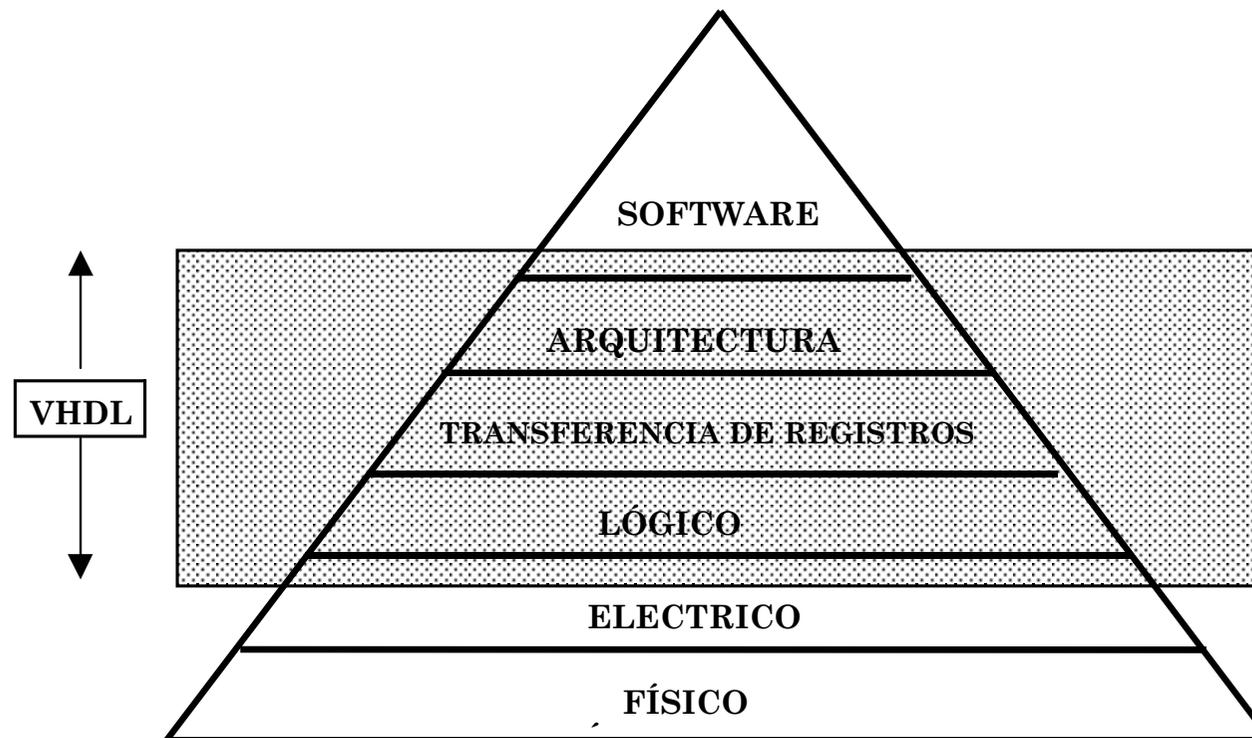
Lenguajes actuales

- **Verilog**
 - Es un lenguaje de descripción hardware diseñado por la compañía Cadence Design Systems Inc., que se ha venido utilizando como lenguaje de su simulador digital.
 - En Verilog la **unidad de diseño fundamental es el módulo**, que describe un componente hardware con su interfaz y contenido.
- **VHDL** (*VHSIC Hardware Description Language*)
 - Es un lenguaje impulsado por el Departamento de Defensa de los Estados Unidos y estandarizado por IEEE Computer Society.
 - Con VHDL se puede estudiar un sistema digital a diferentes niveles de abstracción dentro de un único lenguaje de programación.
 - Es un lenguaje con una semántica orientada a la *simulación*.

5. Lenguajes de descripción hardware (HDL)



Alcance de la capacidad de simulación de VHDL



5. Lenguajes de descripción hardware (HDL)



Áreas de aplicación de VHDL

