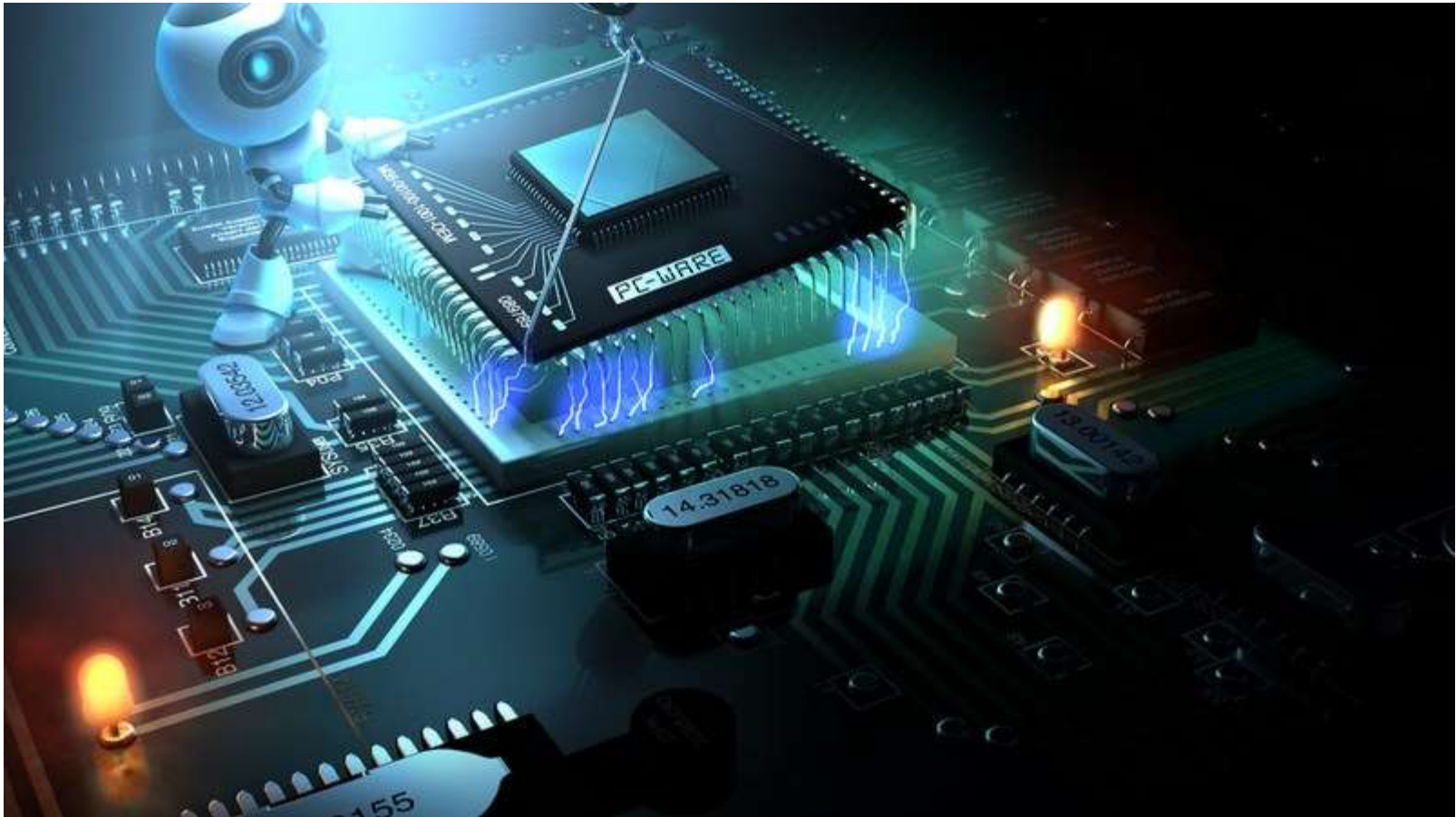


LENGUAJE ENSAMBLADOR

(7°-ICI)

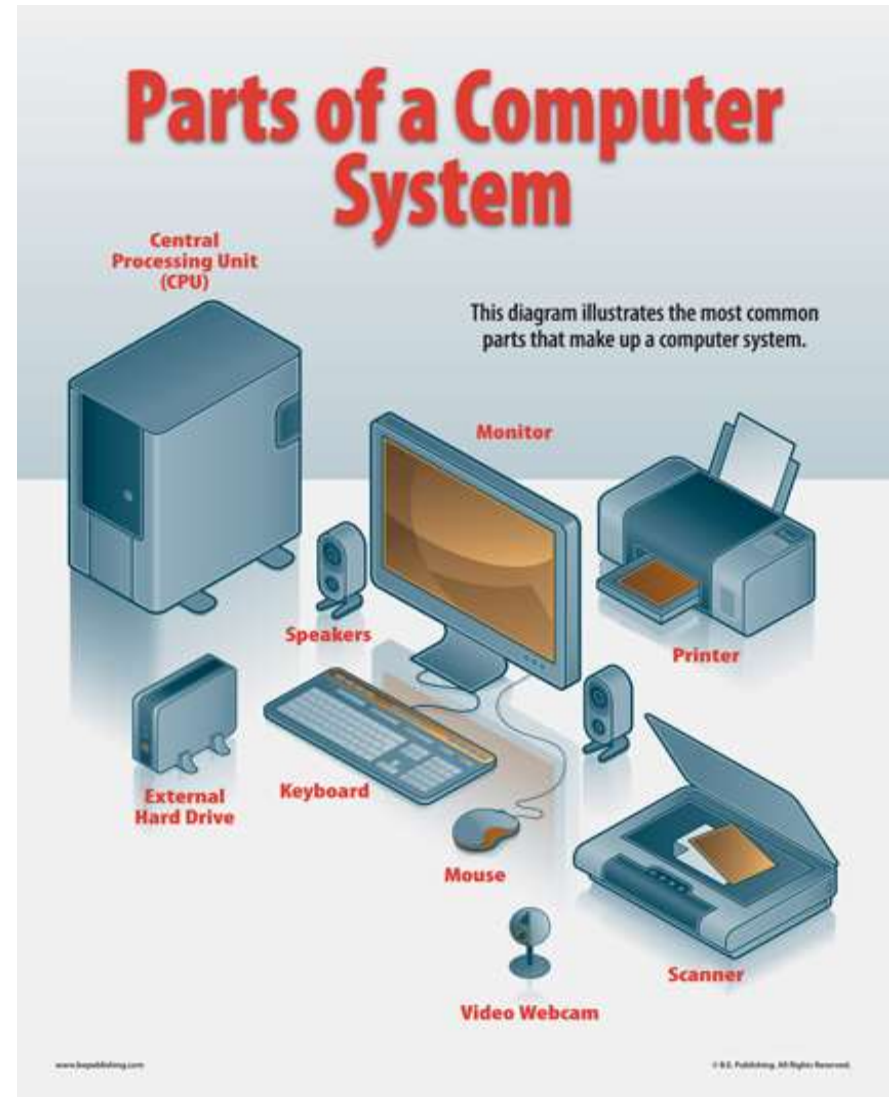


Mtro. en Ing. Armando Álvarez Fdez.
Ago-2024

FUNDAMENTOS DE UN SISTEMA DE COMPUTO

¿Qué es el sistema de Computo?

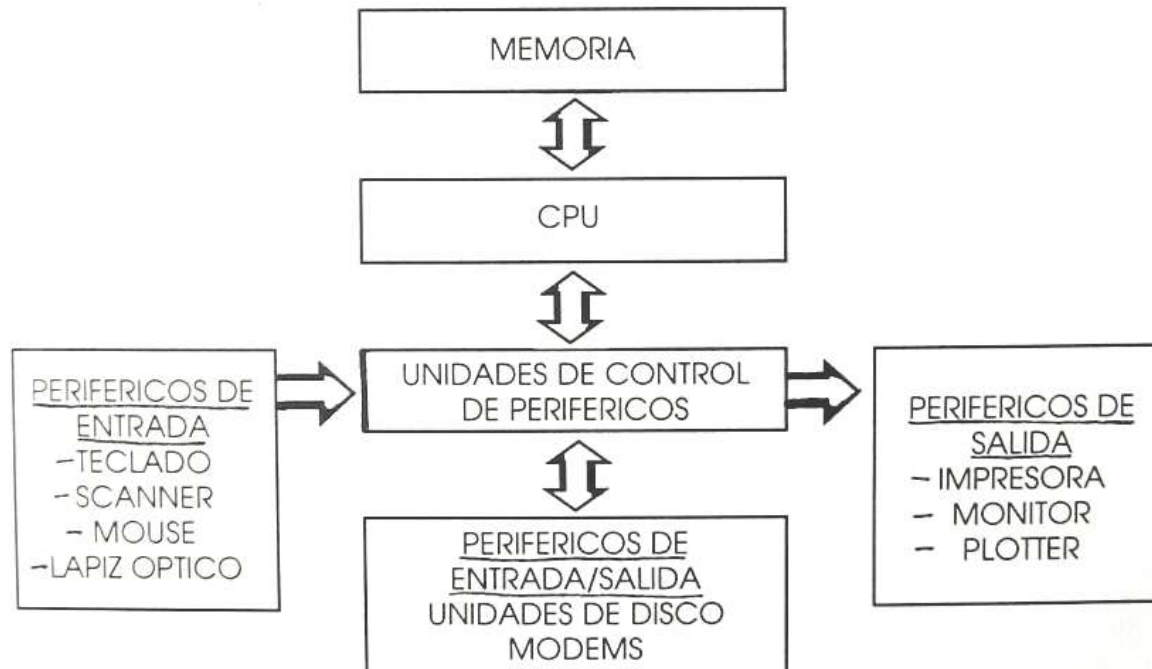
Un sistema de Computo es un conjunto de dispositivos integrados que ingresan, generan, procesan y almacenan datos e información. Los sistemas de Computo se construyen actualmente alrededor de al menos un dispositivo de procesamiento digital (CPU) y todo su hardware de soporte, así como de programas de software y del talento humano que los vuelve productivos.



¿Por qué se llama sistema de Computo?

Se hace referencia a la computadora como un sistema porque está formada por componentes integrados (entrada, salida, almacenamiento y CPU) que trabajan juntos para realizar los pasos necesarios en el programa que se se está ejecutando.

ARQUITECTURA GENERAL DE UNA COMPUTADORA



Como se puede observar, en primer lugar tenemos al CPU (Central Processing Unit) ó unidad central de procesamiento, que es el corazón ó el cerebro de todo sistema y la encargada de procesar los datos de un programa.

El CPU requiere de un lugar de almacenaje de información o memoria, donde serán guardados el programa de computación así como los datos o información generada. Como será visto mas adelante, la memoria se divide en dos tipos: Memoria RAM (Memoria volátil o de acceso aleatorio), generalmente para el usuario y sus programas, así como Memoria ROM (Memoria no volátil o de solo lectura), para el uso del hardware así como funciones elementales y técnicas de operación, generalmente poco accesadas o conocidas por el usuario convencional.

COMPONENTES PRINCIPALES DE HARDWARE EN UN SISTEMA DE COMPUTO

Procesador

Almacenamiento (RAM y ROM)

Unidades de control de dispositivos periféricos

Dispositivos de Entrada,

Dispositivos de Salida

Dispositivos de Entrada/Salida



Partes Internas



Partes Externas



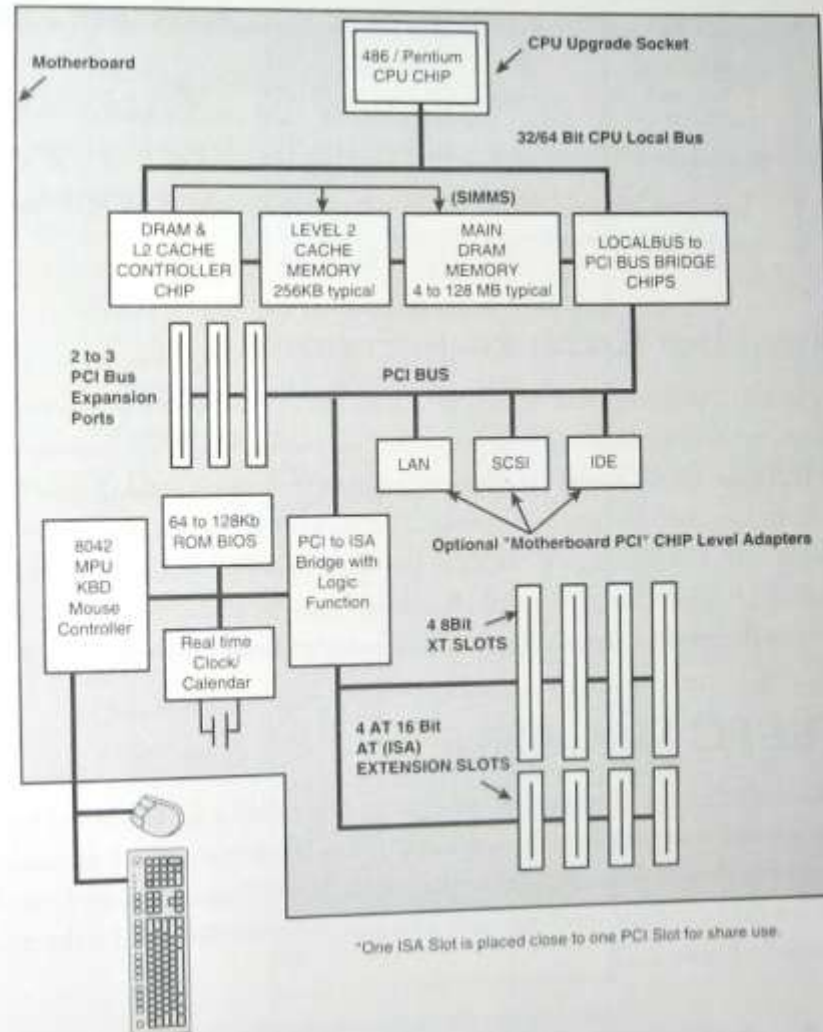
<http://personal.globared.com/computacion-para-el-odontologo>

Hardware

The physical parts of a computer system



TARJETA MADRE (MOTHERBOARD)



MICROPROCESADORES CISC Y RISC

MICROPROCESADORES RISC Y CISC.

Acrónimos: RISC (Reduced Instruction Set Computers)
CISC (Complex Instruction Set Computers)

RISC

Es un tipo de arquitectura especial en el diseño de los CPU's, que se basa en la implementación de un conjunto de instrucciones reducido y optimizadas al máximo, de tal manera que se ejecuten en un ciclo de reloj.

ANTECEDENTES DE LA TECNOLOGIA RISC.

Anteriormente el diseño de CPU's se había centrado en CISC. Se buscaba a ver quien ponía mas cosas en un CPU.

PORCENTAJES DE OCURRENCIA DE INSTRUCCIONES EN DIFERENTES LENGUAJES.

TIPO DE INSTRUCCIONES	% FORTRAN	% C	% PASCAL	% PROMEDIO
-ASIGNACION	51	38	45	47
-CONDICIONALES	10	43	29	23
-LLAMADAS A PROCEDIMIENTOS	5	12	15	15
-ITERACIONES	9	3	5	6
-GOTO	9	3	0	3
-OTRAS	16	1	6	7

CONCLUSION:

"LOS CISC OFRECEN MAS DE LO QUE EN REALIDAD SE REQUIERE"

ARQUITECTURAS ASPECTOS	CISC	RISC
<i>Significado</i>	Computadoras con un conjunto de instrucciones reducido.	Computadoras con un conjunto de instrucciones complejo
<i>Aplicación</i>	Aplicada en ordenadores domésticos	Utilizada para entornos de red
<i>Características</i>	Instrucciones muy amplias.	Instrucciones de tamaño fijo. Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos.
<i>Objetivos</i>	Permitir operaciones complejas entre operandos situados en la memoria o en los registros internos	Posibilitar la segmentación y el paralelismo en la ejecución de instrucciones y reducir los accesos a memoria.
<i>Ventajas</i>	Reduce la dificultad de crear compiladores. Permite reducir el costo total del sistema. Mejora la compactación de código. Facilita la depuración de errores.	La CPU trabaja más rápido al utilizar menos ciclos de reloj. Reduciendo la ejecución de las operaciones. Cada instrucción puede ser ejecutada en un solo ciclo del CPU
<i>Microprocesadores Basados en:</i>	<ul style="list-style-type: none"> • Intel 8086, 8088, 80286, 80386, 80486. • Motorola 68000, 68010, 68020, 68030, 6840 	MIPS Technologies IBM POWER PowerPC de Motorola e IBM SPARC y UltraSPARC

CUADRO COMPARATIVO

PROBLEMAS QUE IMPIDEN LA EFICIENCIA DE UN CISC

MAYOR NUMERO DE INSTRUCCIONES. = MAYOR TIEMPO GASTADO EN DECODIFICAR INSTRUCCIONES.

MAYOR NUM. DE MODOS DE DIRECCIONAMIENTO. = MAYOR TIEMPO GASTADO EN DETERMINAR DIRECCIONES.

SE IMPLEMENTAN FUNCIONES QUE CASI NO SE USAN.

CISC (complex instructions)

```
010010111001001110111111001111
1110001101001100100110011001111
0001101010011101000111100111
```

```
010010111001001110111111001111
1110001101001100100110011001111
0001101010011101000111100111010
001101001100100110011001111000
1101010011101000111100111010001
101001100100110011001111000110
1010011101000111100111010
```

```
10100100
```

RISC (simple instructions)

```
0100101110010011
1011111110011111
```

```
0111100100100111
0111111100111011
```

```
0111100111110100
1011101110100111
```

```
0101110101011111
1100111001001111
```

```
1110101111010111
011111110011101
```

CISC	RISC
Instrucciones multiciclo	Instrucciones de único ciclo
Arquitectura memoria-memoria	Arquitectura registro-registro
Instrucciones largas, Código con menos líneas	Instrucciones cortas, Código con más líneas
Se enfatiza la versatilidad del repertorio de instrucciones	Se añaden instrucciones nuevas sólo si son de uso frecuente y no reducen el rendimiento de las más importantes
Reduce la dificultad de implementar compiladores	Compiladores complejos

HISTORIA.

Se considera que el primer CPU RISC surge cuando John Cocke (entre 1974 y 1975), publica un trabajo dentro de IBM y desarrollan la computadora 801 de IBM en 1975 con técnicas de RISC. (El trabajo se difundió hasta 1982). Por otro lado de 1980 a 1982 en Berkeley David Paterson y Carlo Sequin lanzaron su RISC I y RISC II.

REGLA DE ORO DE RISC:

La regla más importante de RISC es sacrificar todo para reducir el ciclo de tiempo de la trayectoria de datos.

COMPARACION

RISC

- Instrucc. sencillas en un ciclo
- Instrucc. ejecutadas por hardware completamente.
- Pocas instrucc. y modos.
- La complejidad está en el compilador.

CISC

- Instrucc. complejas en varios ciclos.
- Instrucc. interpretadas por el microprograma.
- Muchas instrucc. y modos.
- La complejidad está en el microprograma.

RISC vs. CISC

- En realidad es difícil de comparar ambas arquitecturas puesto que tienen objetivos diferentes, una apuesta por la sencillez del hardware (RISC) y la otra por la aceleración en instrucciones (CISC):

CISC	RISC
Énfasis en el Hardware (Velocidad)	Énfasis en el Software (Sencillez y rapidez en el Pipeline)
Incluye instrucciones complejas multi-reloj	Instrucciones con reloj simple
Instrucciones complejas (Más de 1000)	Instrucciones sencillas (Menos de 100)
Instrucciones que trabajan de memoria a memoria	Instrucciones que trabajan de registro a registro
Carga y almacenamiento incorporados en la misma instrucción	La carga y el almacenamiento son instrucciones independientes
El tamaño del código es pequeño	El tamaño del código es muy grande
Muchos ciclos de reloj por segundo	Pocos ciclos de reloj por segundo
Pocos registros para almacenar los resultados	Muchos registros para almacenar resultados
Alto costo de producción	Bajo costo de producción

POWER PC (CPU RISC)

El 3 de julio de 1991 IBM, Apple, y Motorola firmaron un acuerdo para desarrollar el CPU Power PC.

Por una parte los creadores de Power PC. aseguran que las aplicaciones en DOS, MAC, ALUX, trabajarán tan rápido como los actuales. También se espera que posteriormente los desarrolladores de Software y Hardware desarrollen aplicaciones nativas en el sistema CSF/1 UNIX. Tanto Apple, IBM y Motorola seguirán trabajando en sus sistemas anteriores y aún mejorandolos.

LA FAMILIA POWER PC.

601.- Primer POWER PC; 32 bits; 3 capas superescalares; 50,66 y 80 MHZ.

Desempeño: 85 SPECint 92, 105 SPECfp 92 a 80 MHZ;1993.

602.-32 bits; 3 capas superescalares; 66 y 80 MHZ;Desempeño: 75 SPECint, 85 SPECfp a 80 MHZ;1993

604.- CPU veloz de punto flotante de doble precisión; 32 bits; 4 capas superescalares.

(Esta disponible aproximadamente el segundo semestre de 1994).

620.- CPU de 64 bits; 6 capas superescalares.

(Esta disponible el 1er. semestre de 1995).

ALGUNOS OTROS CPUs RISCs:

Low-end and mobile systems [\[edit\]](#)

By the beginning of the 21st century, the majority of low-end and mobile systems relied on RISC architectures.^[46] Examples include:

- The [ARM architecture](#) dominates the market for low power and low cost embedded systems (typically 200–1800 MHz in 2014). It is used in a number of systems such as most [Android](#)-based systems, the Apple [iPhone](#) and [iPad](#), Microsoft [Windows Phone](#) (former [Windows Mobile](#)), RIM devices, Nintendo [Game Boy Advance](#), [DS](#), [3DS](#) and [Switch](#), [Raspberry Pi](#), etc.
- IBM's [PowerPC](#) was used in the [GameCube](#), [Wii](#), [PlayStation 3](#), [Xbox 360](#) and [Wii U](#) gaming consoles.
- The [MIPS](#) line (at one point used in many [SGI](#) computers) was used in the [PlayStation](#), [PlayStation 2](#), [Nintendo 64](#), [PlayStation Portable](#) game consoles, and [residential gateways](#) like [Linksys WRT54G](#) series.
- [Hitachi's SuperH](#), originally in wide use in the [Sega Super 32X](#), [Saturn](#) and [Dreamcast](#), now developed and sold by [Renesas](#) as the [SH4](#).
- [Atmel AVR](#) used in a variety of products ranging from [Xbox](#) handheld controllers and the [Arduino](#) open-source microcontroller platform to [BMW](#) cars.
- [RISC-V](#), the [open-source](#) fifth Berkeley RISC ISA, with 32- or 64-bit [address spaces](#), a small core integer instruction set, and an experimental "Compressed" ISA for code density and designed for standard and special purpose extensions.

Desktop and laptop computers [\[edit\]](#)

- IBM's [PowerPC](#) architecture was used in Apple's [Macintosh](#) computers from 1994, when they began a switch from [Motorola 68000 family](#) processors, to 2005, when they [transitioned to Intel x86 processors](#).^[47]
- Some [chromebooks](#) use ARM-based platforms since 2012.^[48]
- [Apple](#) uses [self-designed processors](#) based on the ARM architecture for its lineup of desktop and laptop computers since its [transition](#) from [Intel processors](#),^[49] and the first such computers were released in November 2020.^[40]
- [Microsoft](#) uses [Qualcomm](#)^[50] ARM-based processors for its [Surface](#) line.

Workstations, servers, and supercomputers [\[edit\]](#)

- [MIPS](#), by [Silicon Graphics](#) (ceased making MIPS-based systems in 2006).
- [SPARC](#), by [Oracle](#) (previously [Sun Microsystems](#)), and [Fujitsu](#).
- IBM's [IBM POWER instruction set architecture](#), [PowerPC](#), and [Power ISA](#) were and are used in many of IBM's supercomputers, mid-range servers and workstations.
- [Hewlett-Packard's PA-RISC](#), also known as [HP-PA](#) (discontinued at the end of 2008).
- [Alpha](#), used in [single-board computers](#), workstations, servers and supercomputers from [Digital Equipment Corporation](#), then [Compaq](#) and finally [Hewlett-Packard \(HP\)](#) (discontinued as of 2007).
- [RISC-V](#), the [open source](#) fifth Berkeley RISC ISA, with 64- or 128-bit [address spaces](#), and the integer core extended with floating point, [atomics](#) and [vector processing](#), and designed to be extended with instructions for networking, I/O, and data processing. A 64-bit superscalar design, "Rocket", is available for download. It is implemented in the [European Processor Initiative](#) processor.

TAXONOMÍA DE FLYNN

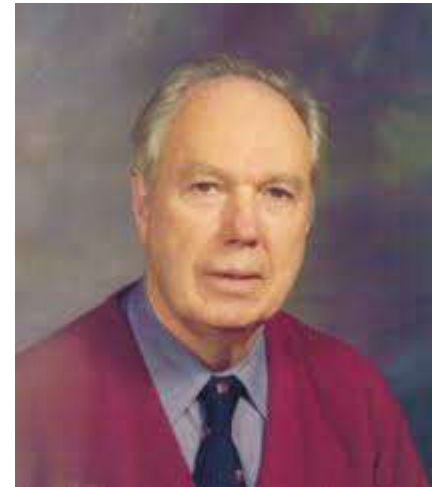
TAXONOMÍA DE FLYNN

La taxonomía de Flynn es una clasificación para las computadoras con arquitectura paralela, propuesta por el profesor emérito de la Universidad de Stanford **Michael J. Flynn**, la cual clasifica a las mismas atendiendo a la **cantidad de instrucciones y flujo de datos concurrentes** en un instante de procesamiento.

Taxonomía de Flynn

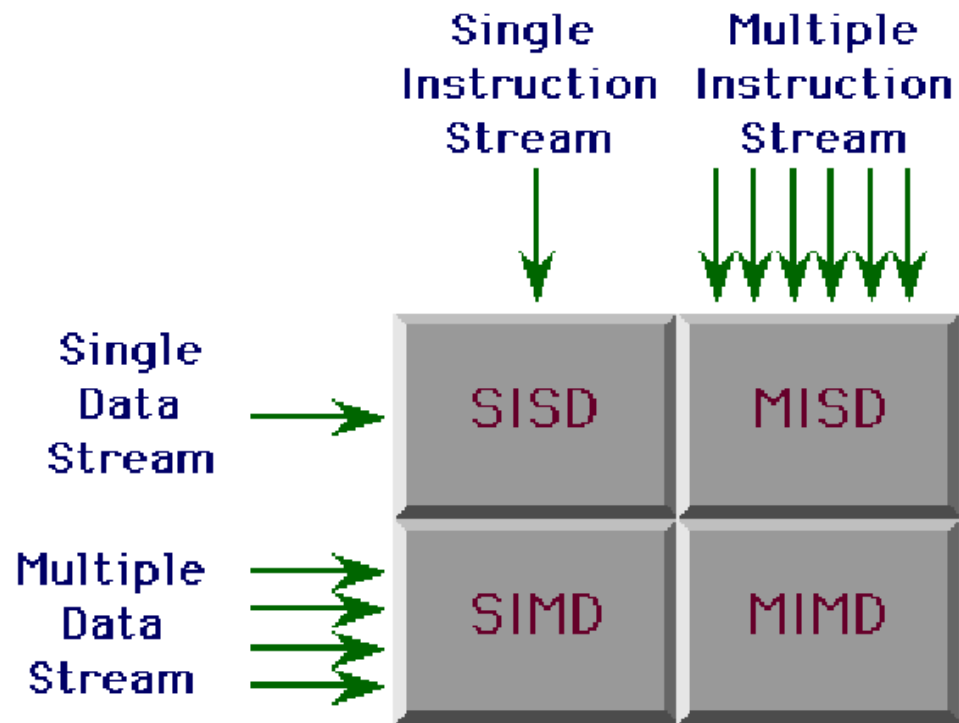
Es la clasificación más extendida del paralelismo:

- Distingue entre instrucciones y datos
- Estos pueden ser simples o múltiples



		Datos	
		Simple	Múltiples
Instrucciones	Simple	SISD	SIMD
	Múltiples	MISD	MIMD

TAXONOMÍA DE FLYNN



SISD:(Single Instruction Single Data)

Esta clasificación se refiere a las **computadoras tradicionales y secuenciales** en las cuales **una** instrucción a la vez se ejecuta sobre **un único** dato cada ciclo de reloj.

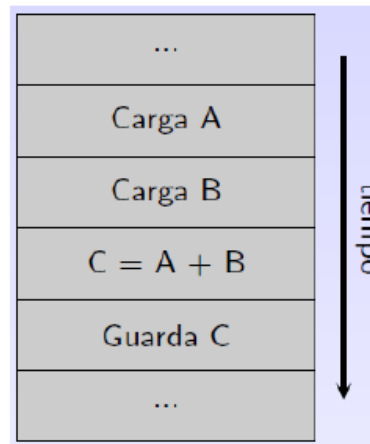
Los datos en cuestión se almacenan en una única memoria en la cual se **usan técnicas como la segmentación** para evitar errores de fragmentación interna. Un ejemplo sencillo de estas computadoras son los antiguos mainframe basados en la arquitectura de Von-Neumann.

SISD: Single Instruction, Single Data

Características del modelo SISD:

- La CPU procesa únicamente una instrucción por cada ciclo de reloj
- Únicamente un dato es procesado en cada ciclo de reloj
- Es el modelo más antiguo de computadora y el más extendido

Ejemplo: la mayoría de las computadoras, servidores y estaciones de trabajo



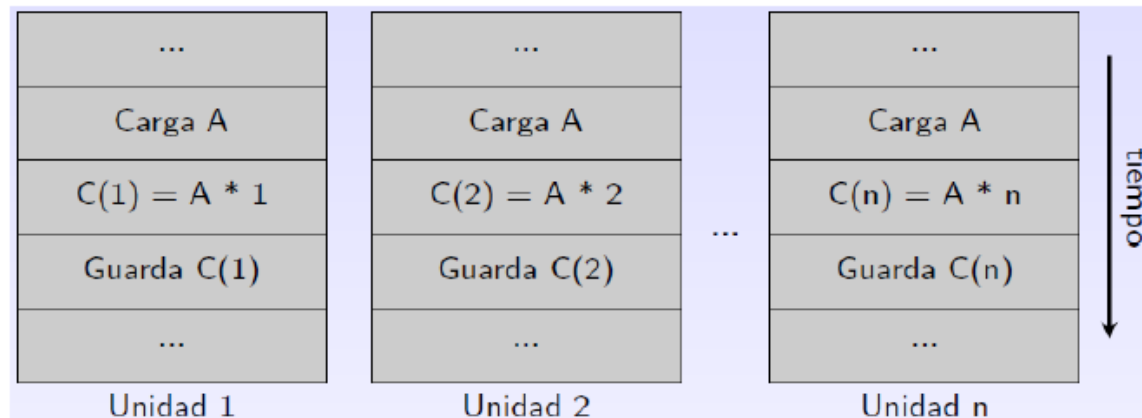
Constan de una única unidad de control de memoria (CU, control unit) y una única unidad aritmético-lógica (ALU), por tanto, sólo ejecuta una única instrucción en un único dato por ciclo de reloj. Corresponde a la arquitectura de von Neumann original.

MISD: (Multiple Instruction Single Data)

Arquitectura que se refiere a **múltiples instrucciones** ejecutándose **sobre un único dato**. Comúnmente se considera esta **arquitectura poco práctica** ya que en tiempo de ejecución la efectividad del **paralelismo requiere un múltiple flujo de datos** y, además, el acceso concurrente a un mismo dato en memoria puede ocasionar que un **CPU tenga que esperar** a que el recurso(**dato**) esté disponible para poder acceder a él.

MISD: Multiple Instruction, Single Data

- Características del modelo MISD:
- Cada unidad ejecuta una instrucción distinta
- Cada unidad procesa el mismo dato
- Aplicación muy limitada en la vida real



Las pocas computadoras de este tipo, que existen generalmente se utilizan para tolerancia a fallos, donde varios chequeos se hacen en paralelo y un resultado general se guarda en una variable en memoria.

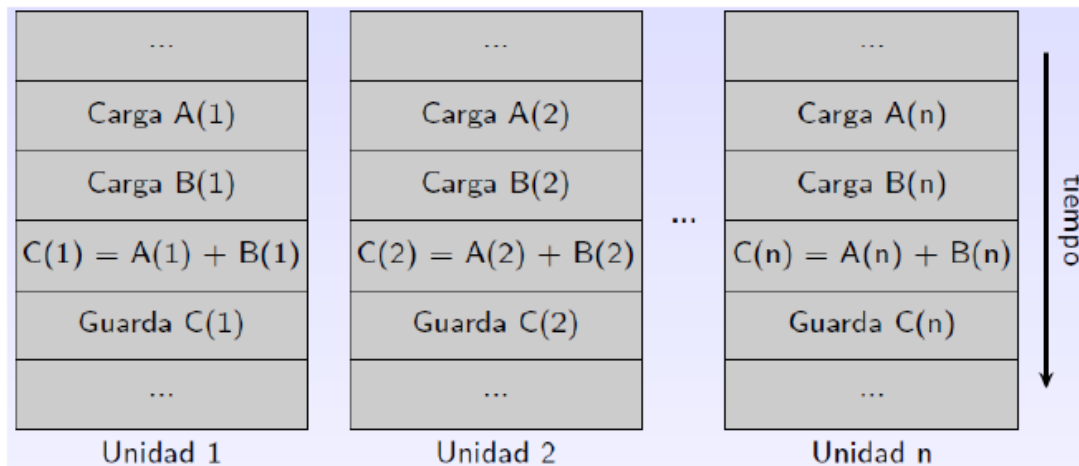
SIMD:(Single Instruction Multiple Data)

Esta arquitectura representa la ejecución de una misma instrucción sobre un conjunto de datos. La misma es comúnmente vista en ciclos de programación que ejecutan una misma instrucción una y otra vez sobre datos de un arreglo o conjunto de datos. En la arquitectura SIMD estos datos son procesados por múltiples CPU que ejecutan la misma instrucción sobre una parte del conjunto o arreglo, cada uno, hasta llegar a procesar la totalidad de los mismos.

SIMD: Single Instruction, Multiple Data

Características del modelo SIMD:

- Todas las unidades ejecutan la misma instrucción
- Cada unidad procesa un dato distinto
- Todas las unidades operan simultáneamente



Dos tipos de arquitecturas SIMD son:

Procesadores vectoriales (vector processors). Son procesadores cuyas instrucciones trabajan no en un dato a la vez (escalar), sino en arreglos de datos (vectores). Por ejemplo, un procesador tradicional puede calcular la suma de dos registros y guardar el resultado en uno de ellos. Un procesador vectorial tiene varias ALU y puede calcular la suma (una instrucción) de un arreglo de números en un ciclo de reloj si el tamaño del arreglo es menor o igual que su cantidad de ALU.

Unidades de procesamiento gráfico (GPU, graphics processing units). Son similares a los procesadores tradicionales con algunas diferencias. Las instrucciones están orientadas al trabajo con gráficas, como suavizar una curva o aplicar efectos a píxeles, puntos, líneas o triángulos. Dado que estos elementos son usualmente independientes, las GPU son arquitecturas altamente paralelas, compuestas de múltiples shaders. Aunque las GPU se suelen clasificar como arquitecturas SIMD, pueden ejecutar instrucciones distintas en distintos datos.

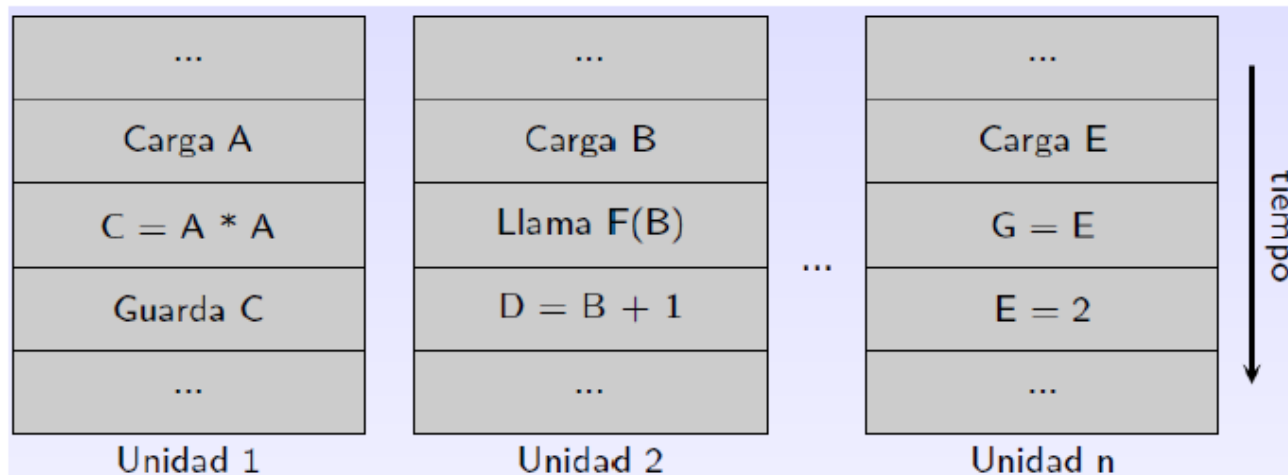
MIMD: (Multiple Instruction Multiple Data)

Esta arquitectura representa a un conjunto de instrucciones que se ejecutan sobre un conjunto múltiple de datos. La misma es muy usada hoy en día para explotar el paralelismo ya sea con memoria distribuida y memoria compartida o híbridos como los clústers de computadoras. Muchos multiprocesadores modernos (como los de la tecnología Core i de Intel) entran en esta clasificación.^[2]

MIMD: Multiple Instruction, Multiple Data

Características del modelo MIMD:

- Cada unidad ejecuta una instrucción distinta
- Cada unidad procesa un dato distinto
- Todas las unidades operan simultáneamente



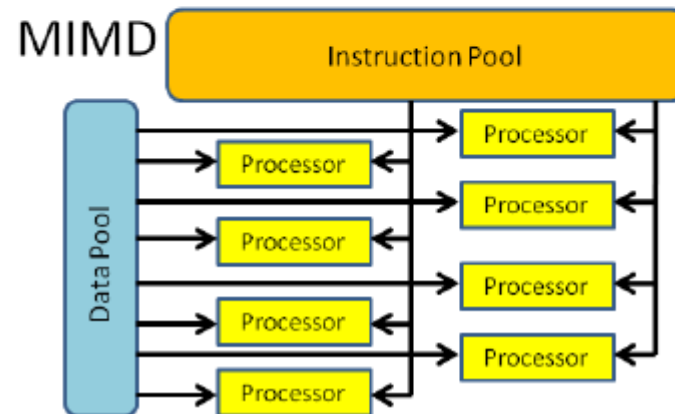
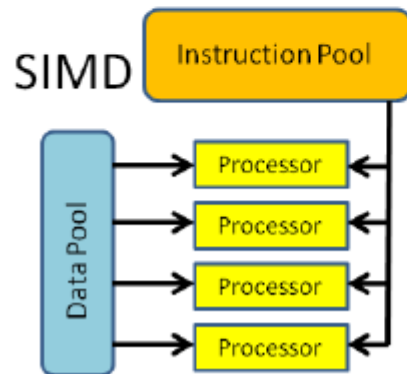
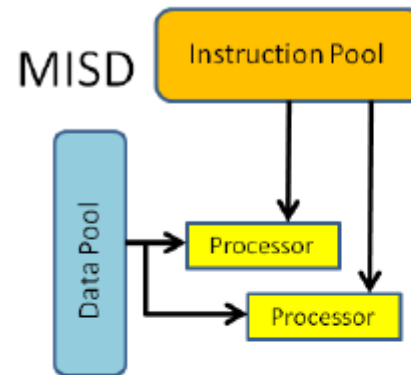
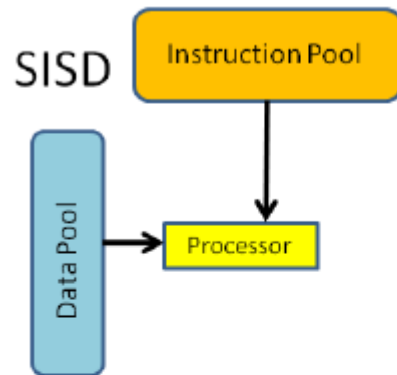
Dos o más núcleos pueden estar simultáneamente ejecutando instrucciones distintas y en datos distintos. Después del 2005 las arquitecturas SISD dejaron de ser las más comunes para dar paso a las arquitecturas MIMD.

Las arquitecturas de múltiples instrucciones y múltiples datos se pueden clasificar en sistemas de memoria compartida y sistemas de memoria distribuida.

Fuentes

- DUNCAN, Ralph. A survey of parallel computer architectures. Computer, 1990, vol. 23, no 2, p. 5-16.
- JOHNSON, Eric E. Completing an MIMD multiprocessor taxonomy. ACM SIGARCH Computer Architecture News, 1988, vol. 16, no 3, p. 44-47.

TAXONOMÍA DE FLYNN DE LA ARQUITECTURA DE COMPUTADORAS



APENDICES

