



# Computación Paralela y Distribuida

---

John Corredor, PhD  
ECEI - IS

# Objetivos

- Comprender la Arquitectura de Von Neumann
- Conocer los procesadores modernos

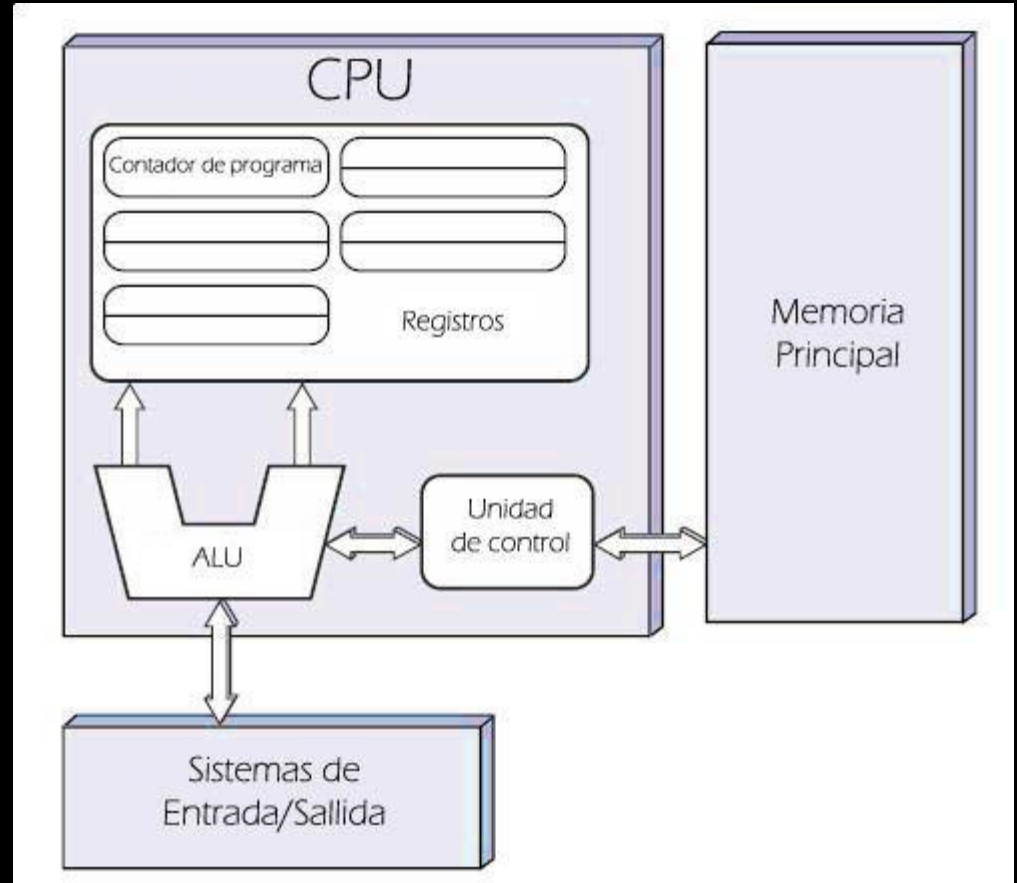
## Arquitectura de Von Neumann

Describe una arquitectura de diseño para un computador digital electrónico con partes que constan de una unidad de procesamiento que contiene una unidad aritmético lógica y registros del procesador, una unidad de control que contiene un registro de instrucciones y un contador de programa, una memoria para almacenar tanto datos como instrucciones, almacenamiento masivo externo, y mecanismos de entrada y salida.



## Características:

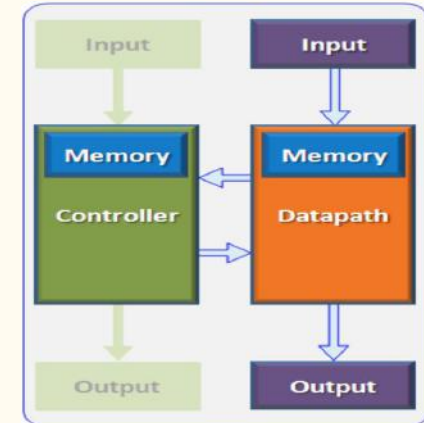
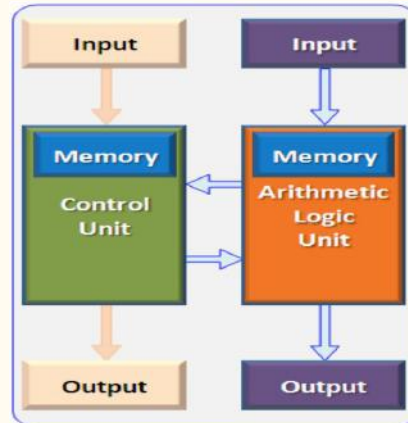
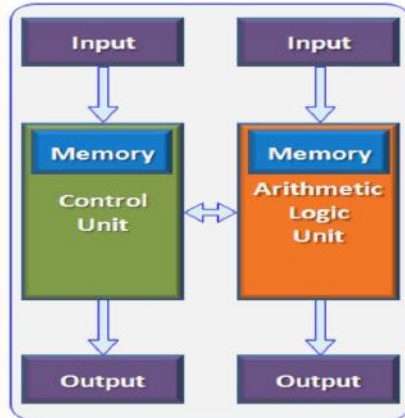
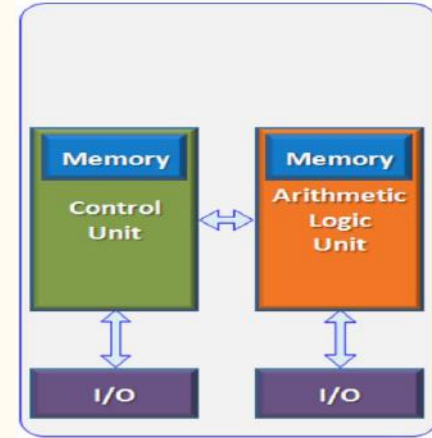
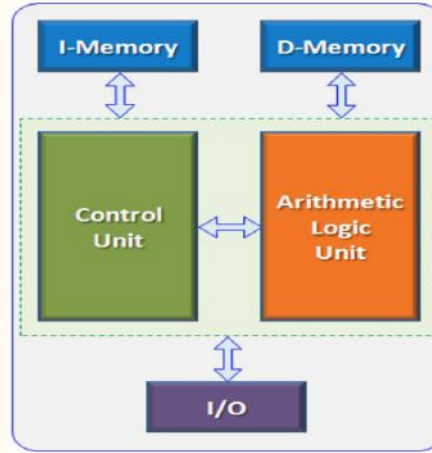
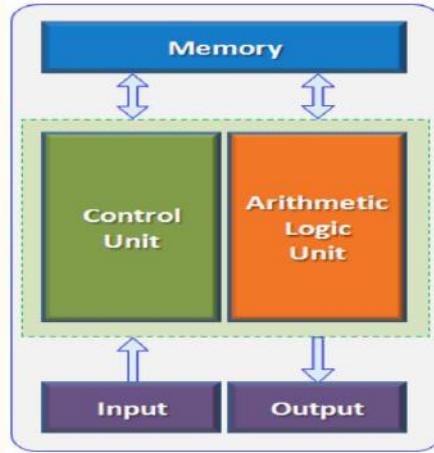
- Memoria
  - Unidad de Control
  - Unidad Aritmética Lógica
  - Interface I/O
- 



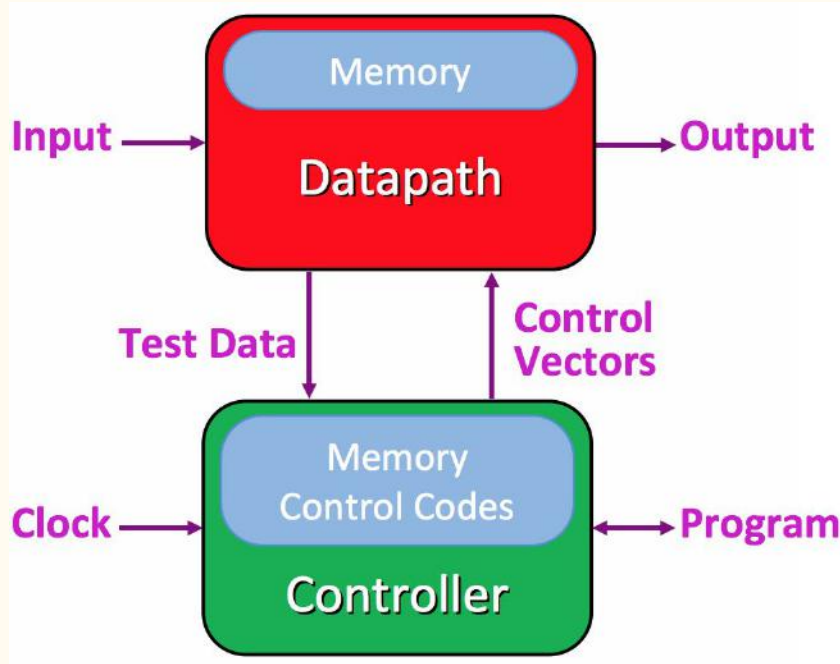
La arquitectura de Von Neumann es un modelo de diseño para una computadora digital de programas almacenados que utiliza una unidad central de procesamiento (CPU) y una única estructura de almacenamiento separada (memoria) para guardar tanto instrucciones como datos.

Desventaja: Memoria compartida para instrucciones y datos con un bus de datos y un bus de direcciones entre el procesador y la memoria. Las instrucciones y los datos tienen que ser recuperados en orden secuencial (conocido como el cuello de botella de Von Neumann), limitando el ancho de banda de la operación.

# Evolución de la Arquitectura

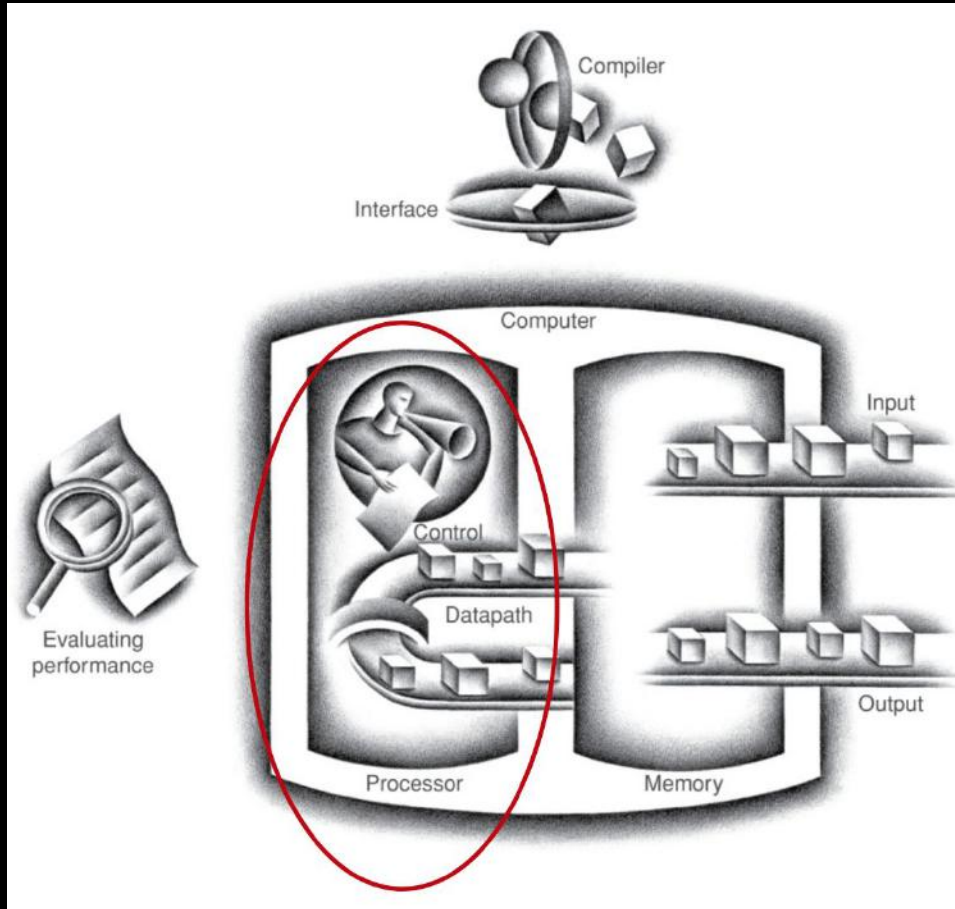


# Arquitectura Universal



Esta arquitectura universal conduce a diferentes estructuras para la ASIC y el procesador de propósito general.

# Componentes del computador



## Datapath:

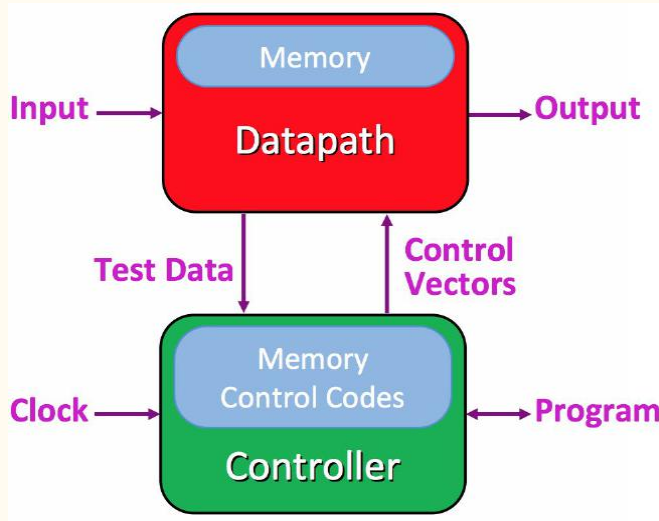
- Componentes del procesador que realizan operaciones aritméticas y guardan datos.

## Control:

- Componente del procesador que gestiona el datapath, la memoria, los dispositivos de E/S según las instrucciones de la memoria.

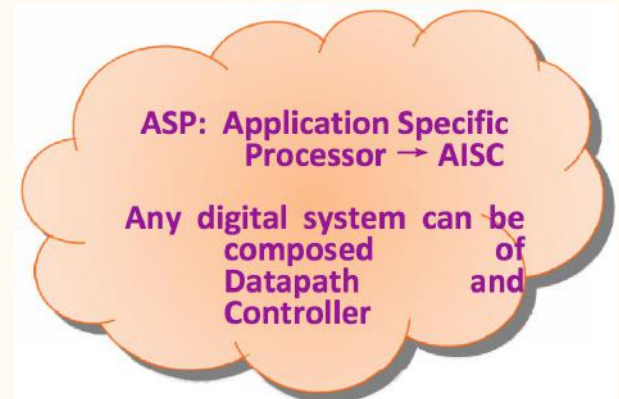


## ASP: Datapath y controlador

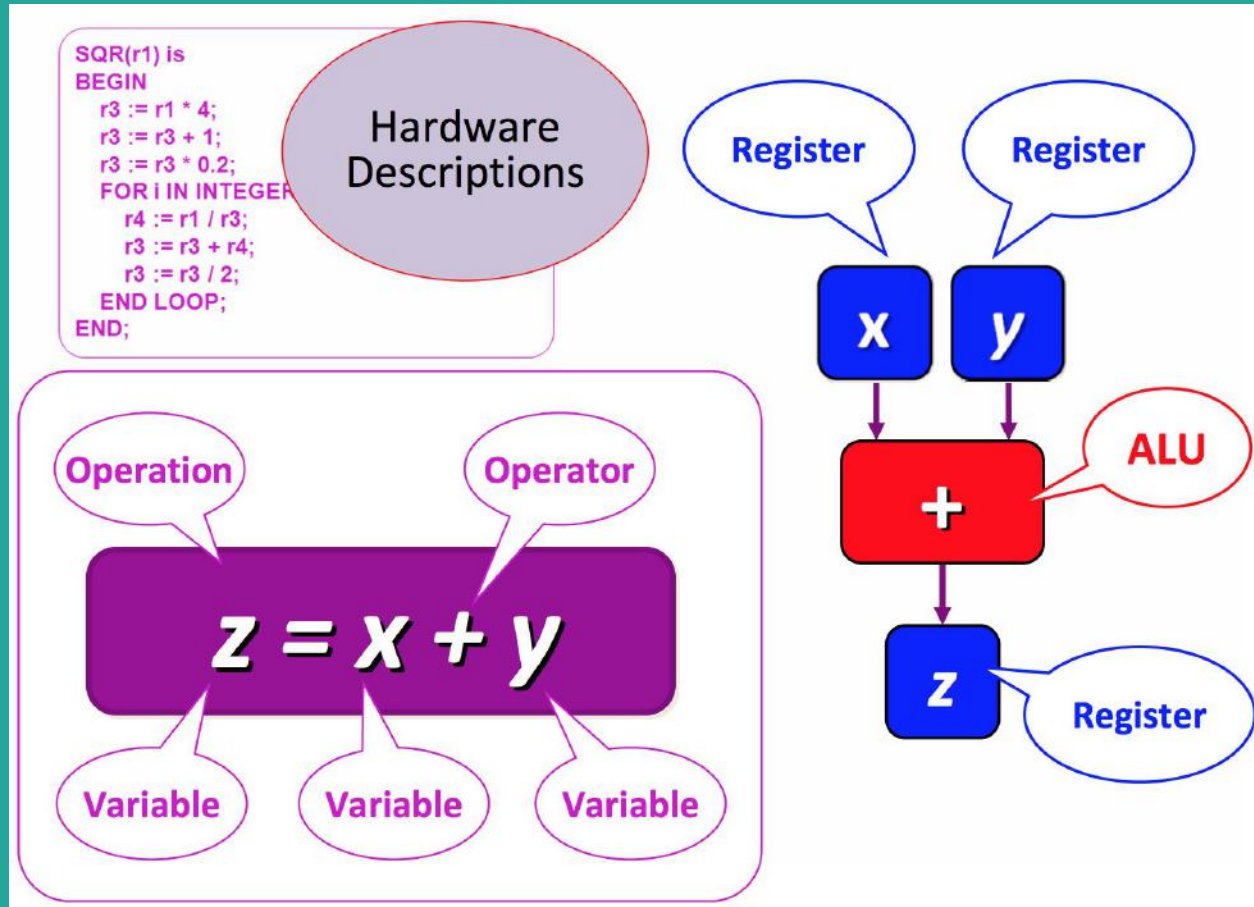


- El controlador consiste en un generador de estado y un generador de control-vector.
- El controlador genera los vectores de control según los datos de prueba y el flujo de datos.
- El controlador funciona según el reloj del sistema.

- Datapath consiste en recursos, memorias e interconexiones.
- Datapath realiza una función cálculo según las entradas.
- Datapath funciona según vectores de control desde el controlador.
- Datapath no es directamente controlado por el reloj del sistema.
- Datapath proporciona la información necesaria datos de prueba al controlador.



# ASCI: Arquitectura Uniforme



## Resumen

- La arquitectura de Von Neumann describe una máquina de computación universal, pero con baja eficiencia.
- Una arquitectura universal para el sistema digital puede ser generada a partir de la arquitectura de Von Neumann que consiste en un datapath y un controlador.
- La metodología de la Síntesis de Alto Nivel es una forma de diseñar el ASIC, que apunta al diseño del hardware.

# Preguntas

## Los años 60 y 70

- ❑ Las instrucciones tomaron múltiples ciclos
- ❑ Sólo una instrucción de vuelo a la vez
- ❑ La optimización significaba minimizar el número de instrucciones ejecutadas
- ❑ A veces se sustituyen las instrucciones de uso general por secuencias especializadas de otras más baratas.

## Los años 1980

- ❑ Las CPUs se convirtieron en pipelines
- ❑ La optimización significaba minimizar las paradas de los pipeline
- ❑ La dependencia se ordena de tal manera que los resultados no se necesitan en la siguiente instrucción
- ❑ Los saltos de cómputo se volvieron muy caros cuando no había predicción correcta

## A principios de los años 1990

- ❑ Las CPU se volvieron mucho más rápidas que la memoria
- ❑ Caches escondían algo de latencia
- ❑ La optimización significaba maximizar la localidad de referencia, prefetching
- ❑ A veces, recalcular los resultados es más rápido que sacarlos de la memoria
- ❑ Nota: ¡Grandes caches consumen mucha energía!

## A mediados de los años 1990

- Las CPU se convirtieron en superescalares
  - Instrucciones independientes ejecutadas en paralelo
- Las CPU se han quedado fuera de servicio
  - Instrucciones re-ordenadas para reducir las dependencias
- La optimización significaba estructurar el código para el mayor ILP posible
- El desenrollado del bucle ya no es una gran victoria



## A finales de los años 1990

- SIMD se convirtió en la corriente principal
- Factor de  $2-4\times$  de velocidad cuando se usa correctamente
- La optimización significaba asegurar el paralelismo de los datos
- El desenrollado de bucle comienza a ganar de nuevo, ya que expone oportunidades de optimización posteriores.

## A principios de los años 2000

- (Homogéneo) El multinúcleo se convirtió en la corriente principal
- La eficiencia energética se volvió importante
- El paralelismo proporciona tanto un mejor rendimiento como una menor potencia
- La optimización significaba explotar el paralelismo de grano fino

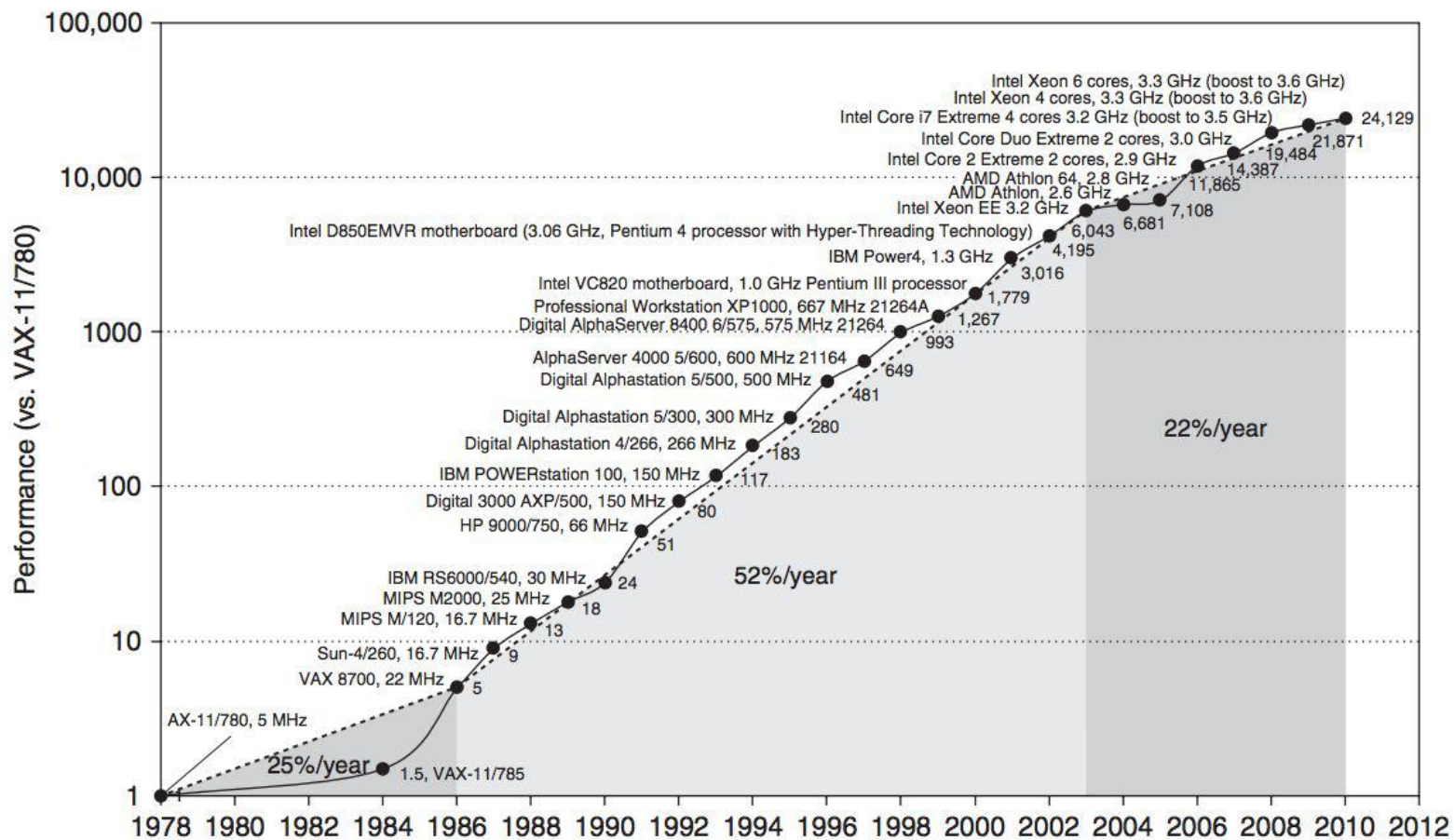
## A finales de los años 2000

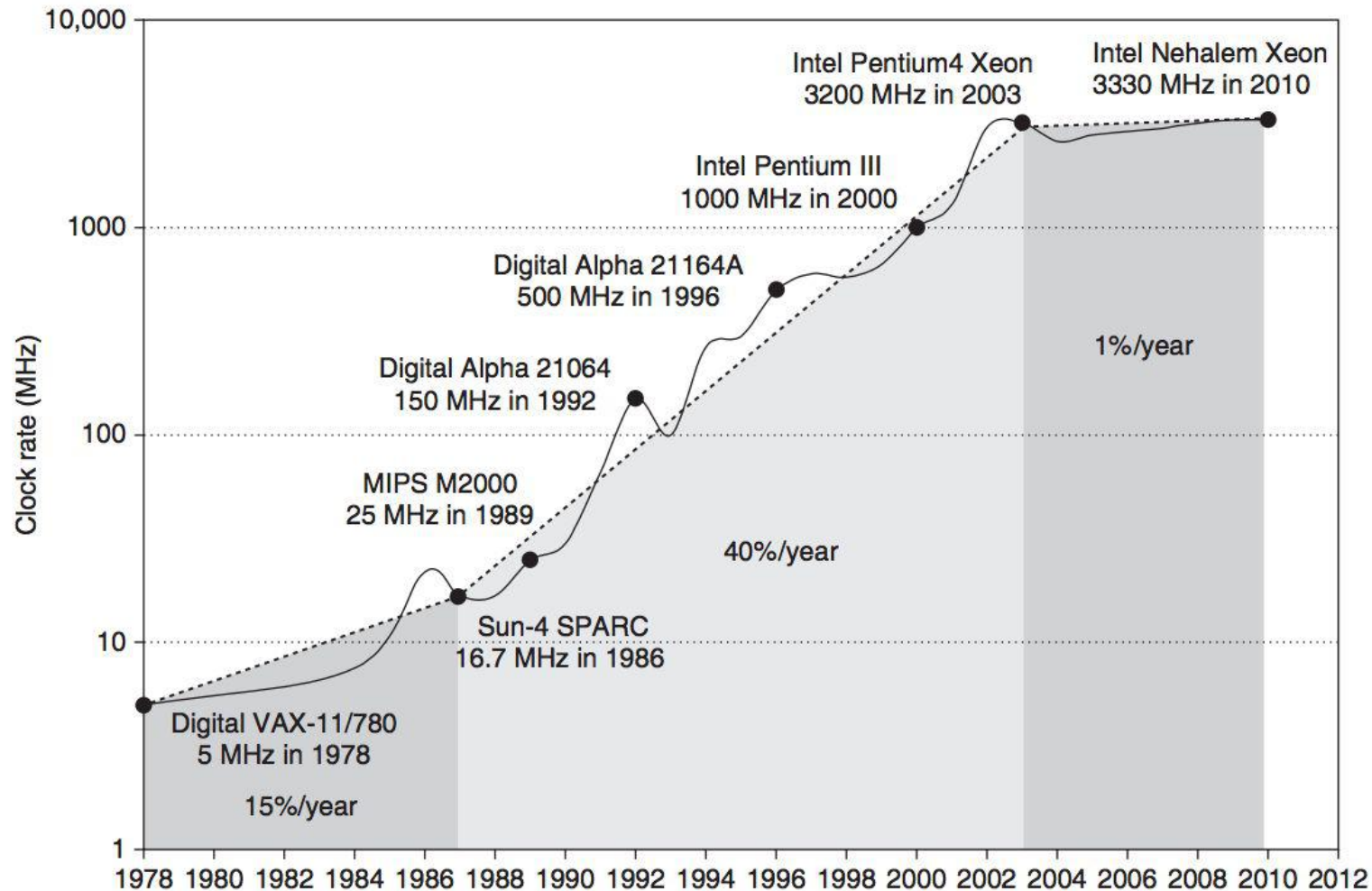
- Las GPU programables se convirtieron en la corriente principal
- Hardware optimizado para el procesamiento de corrientes en paralelo
- Muy rápido para operaciones de punto flotante en paralelo masivo
- El costo de mover datos entre el CPU y la GPU es alto
- La optimización significaba descargar las operaciones a la GPU

## Los años 2010

- Los procesadores modernos vienen con múltiples núcleos de CPU y GPU
- Todos los núcleos detrás de la misma interfaz de memoria, el costo de mover los datos entre ellos son bajos
- Contienen cada vez más aceleradores especializados
- A menudo contienen núcleos de propósito general (programables) para tipos de carga de trabajo especializados (por ejemplo, DSP)
- La optimización es difícil.
- ¡Mucho trabajo para los escritores de compilación!

# Evolución de la Arquitectura





# Referencias

- ★ Hennessy J L and Patterson D. 2011. Computer Architecture, Fifth Edition: A Quantitative Approach 5th ed (San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.) ISBN 012383872X, 9780123838728.
- ★ David B. Kirk and Wen-mei W. Hwu. 2010. Programming Massively Parallel Processors: A Hands-on Approach, Third Edition. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA ©2010 ISBN:0123814723 9780123814722.
- ★ Michael J. Quinn. 2003. Parallel Programming in C with MPI and OpenMP. McGraw-Hill Education Group.
- ★ Victor Eijkhout. 2012. Introduction to High Performance Scientific Computing. Lulu.com.
- ★ Michael McCool, James Reinders, and Arch Robison. 2012. Structured Parallel Programming: Patterns for Efficient Computation (1st ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- ★ Tutorial online:  
[https://www.tutorialspoint.com/parallel\\_computer\\_architecture/index.htm](https://www.tutorialspoint.com/parallel_computer_architecture/index.htm)