Introducción al curso

MSc. Stefano Romero

Motivación

- Las ciencias de la computación están más enfocadas en **la resolución de algoritmos complejos que en el desarrollo mismo del hardware**
- Para comprender por qué un algoritmo es computacionalmente irrealizable, o entender porqué es tan lento, se debe ver el problema **desde el punto de vista de la computadora.**
- Intentar optimizar un sistema informático sin entenderlo primero es como intentar sintonizar tu automóvil vertiendo un elixir en el tanque de gasolina: tendrás suerte si funciona por completo cuando hayas terminado.
- Optimización del programa y ajuste del sistema son quizás las motivaciones más importantes para aprender cómo funcionan las computadoras.

Otras grandes motivaciones

- Desarrollo de compiladores: Entendimiento del entorno del hardware del compilador.
- Modelamiento de grandes y complejos sistemas: Entendimiento de la aritmética de punto flotante y cómo funciona en la práctica.
- Diseño de periféricos: Conocimiento de cómo la computadora interactúa con su entrada y salida.
- Desarrollo de sistemas embebidos: Limitación de recursos, compensación de tiempo, espacio y precio.
- Finalmente, es el entendimiento de la interacción del hardware con el software y a través del mismo.

Conceptos básicos

Organización de computadoras:

- Es la forma en que varios circuitos y componentes se unen para crear sistemas informáticos funcionales. Está relacionado a cómo se controla la computadora y su estudio ayuda a responder cómo funciona una computadora.

Arquitectura de computadoras:

- Se enfoca en la estructura y comportamiento del sistema de la computadora. Se centra en la lógica y los aspectos abstractos de la implementación del sistema visto por el programador. Su estudio ayuda a responder cómo diseñar una computadora.

Conceptos básicos - Arquitectura de computadoras

Es la combinación del hardware más el set de instrucciones de arquitectura (ISA, por sus siglas en inglés). El ISA es el interfaz acordada entre todo el software que se ejecuta en la máquina y el hardware que lo ejecuta.

Las características de ISA se pueden enlistar como las siguientes:

- Clases de instrucciones: Transferencia de data, ALU (ADD, SUB, AND, OR), punto flotante, etc.
- Operaciones y acceso en memoria (Register, Immediate, Displacement, etc.).
- Tamaño y tipos de datos que maneja (Binary Integer (8-bit,…,-64-bit), BCD, punto flotante, etc.).

Componentes principales de un computador

- Procesador (Cerebro):
  - Permite interpretar y ejecutar programas.

- Memoria (Apuntes):
  - Para grabar datos y programas.

- Dispositivos de E/S (Lapicero).
  - Transferencia de datos de y para el mundo.

Desarrollo histórico

- Generación cero: Máquinas de cálculo mecánico 1642-1945
- 1era generación: 1945-1953 → Tubos de vacío
- 2da generación: 1954-1962 → Transistores
- 3era generación: 1963-1970 → Circuitos Integrados
- 4ta generación: 1970 - ? → VLSI (very large-scale integration): +10,000 componentes

Ver desarrollo histórico

Visión por niveles de abstracción del computador

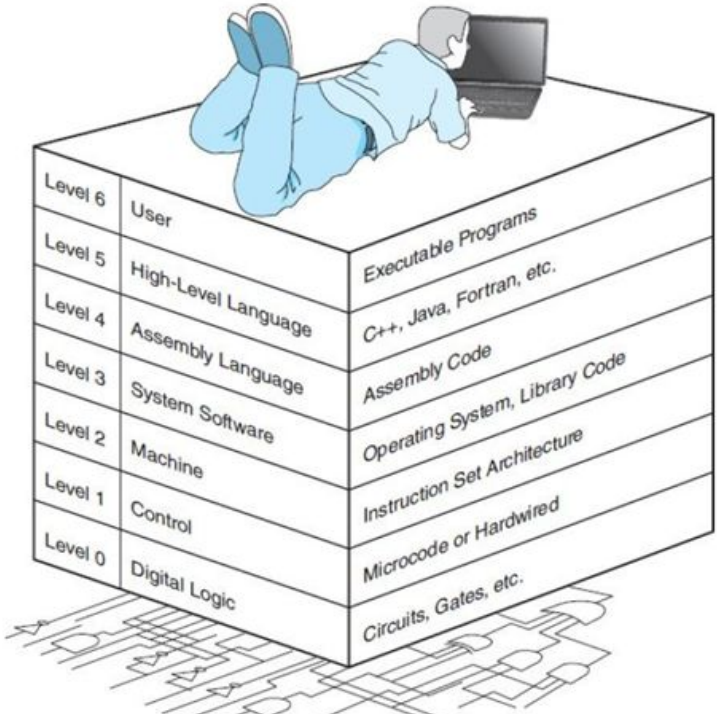

Figura 1. Niveles de abstracción del computador[1].

Modelo de Von Neumann

- Publicado y publicitado por John von Neumann. Propuesto por John W. Mauchly y J. Presper Eckert.
- Fue el primer diseño de computadora en utilizar el concepto de un **programa almacenado en memoria.**
- Principales características:
  - Tres sistemas de hardware:
    - CPU con unidad de control, ALU, registros y contador de programa.
    - Memoria principal, con programas que controlan la operación de la computadora.
    - Sistema de I/O.
  - Capacidad para llevar a cabo el proceso de instrucción secuencial.
  - Contiene una sola ruta, física o lógica, entre el sistema de memoria principal y la unidad de control de la CPU, lo que obliga a la alternancia de instrucciones y ciclos de ejecución **(cuello de botella de von Neumann)**.

Modelo de Von Neumann

- La idea de Von Neumann ha sido extendida a que los programas y los datos se almacenen en un medio de almacenamiento de acceso lento, como un disco duro, y puedan copiarse a un medio de almacenamiento volátil y de acceso rápido como la RAM antes de su ejecución.

- Von Neumann se ha simplificado utilizando el modelo de buses del sistema: bus de datos e instrucciones, bus de direcciones y bus de control.

Modelo de Von Neumann

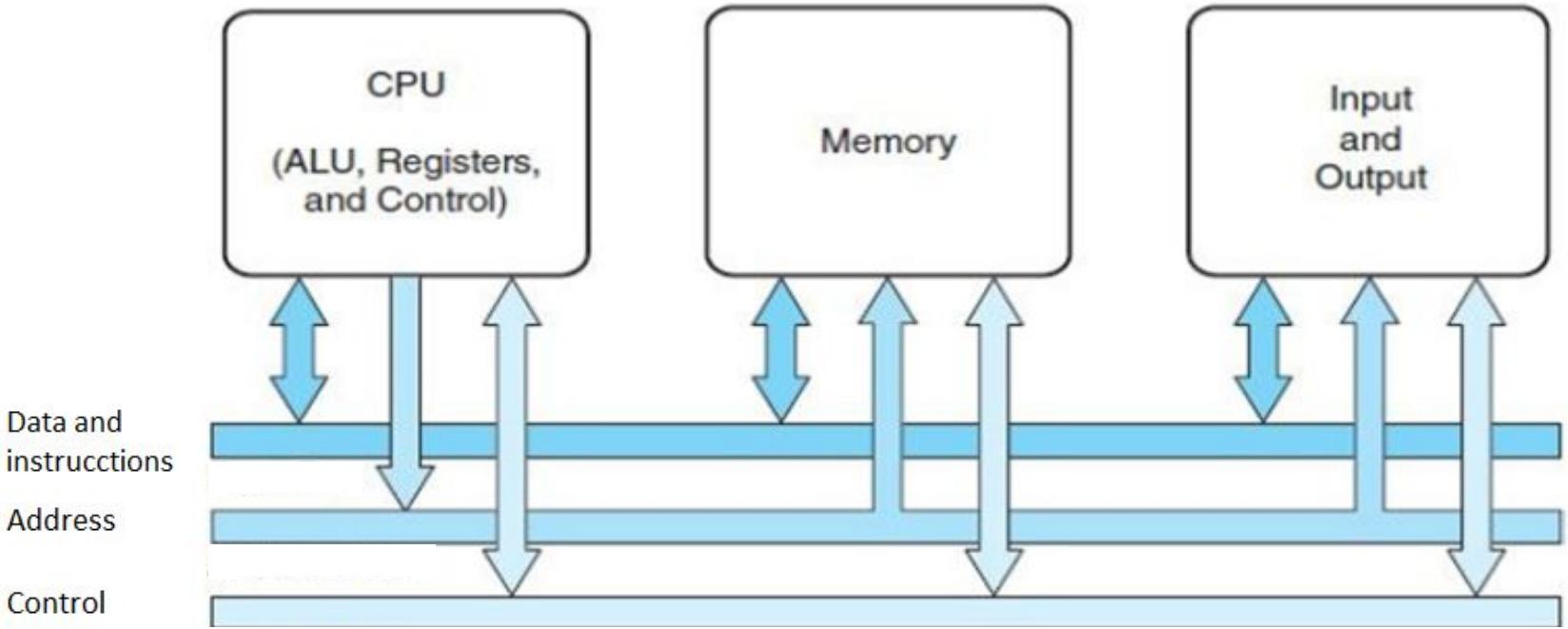

Figura 2. Modelo de Von Neumann [1].

Modelos alternativos a Von Neumann: Arquitectura Harvard

- Basada en la computadora Mark Harvard I.
- **Datos e instrucciones tienen su propio bus y su propio almacenamiento separado**, por lo que la transferencia paralela es posible.
- Ventaja: No se **presenta el cuello de botella** de von Neumann ya que el procesamiento es en paralelo y hay mayor ancho de banda.
- Desventaja: Se deben proporcionar mecanismos para cargar por separado el programa que se ejecutará en la memoria de instrucciones y cualquier información que se operará en la memoria de datos.
- Comúnmente, se utilizan en microcontroladores, sistemas embebidos, entre otros.

Modelos alternativos a Von Neumann: Arquitectura Harvard

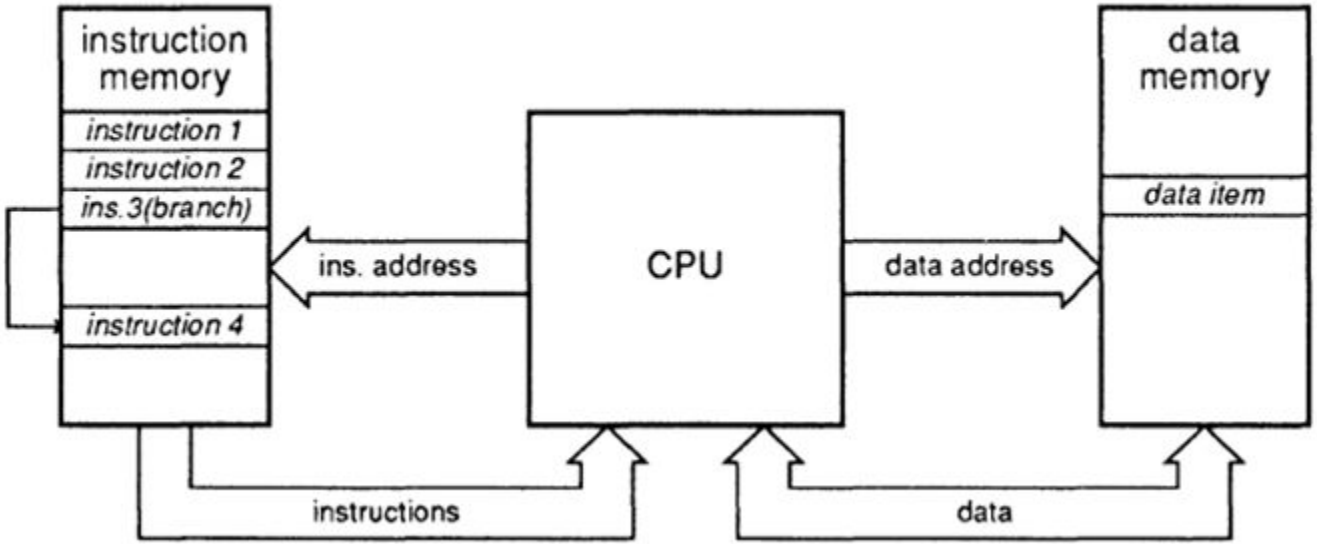

Figura 3. Modelo de Harvard [3].

Comparación entre arquitecturas

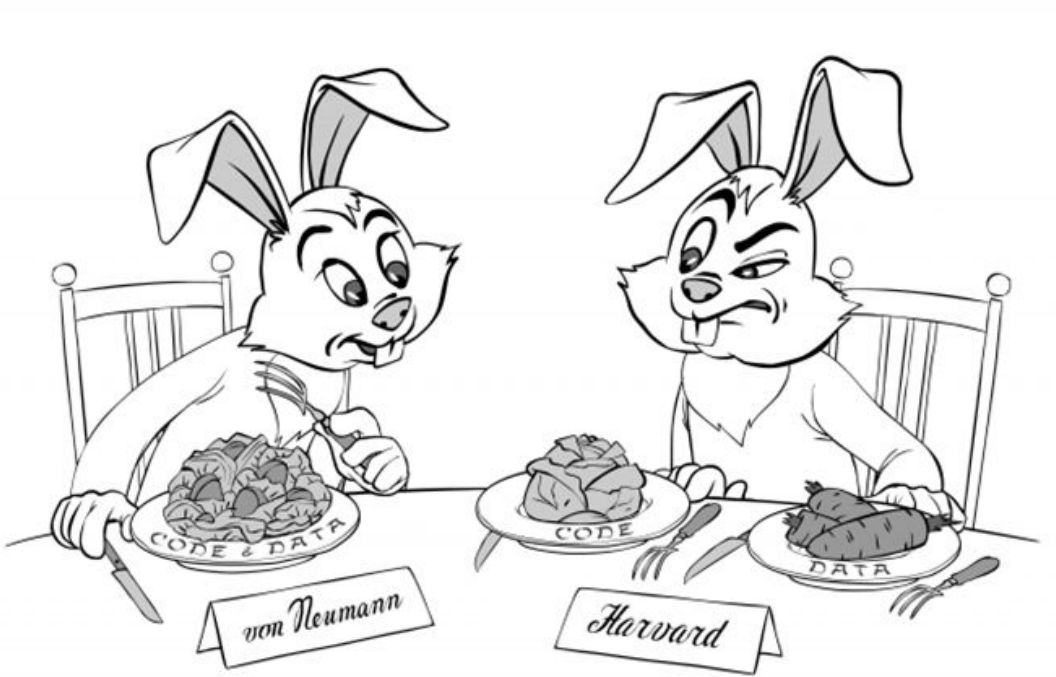

Figura 4. Imagen extraída de “Modified Harvard Architecture: Clarifying Confusion” [4]

Modelos alternativos a Von Neumann:Arquitectura Harvard modificada:

- Principales características:
  - Memoria caché dividida.
  - Acceso a la memoria de instrucciones como data.
  - Leer instrucciones de la memoria de la datos.
- Para lectura, se pueden colocar datos constantes en el espacio de instrucciones.
- Para escritura, se puede escribir espacio de instrucciones durante la ejecución.
- Esta arquitectura también está diseñada para diferentes tipos de sistemas embebidos (Famila AVR) o microcontroladores (COP8).

Modelos alternativos a Von Neumann: Arquitectura Harvard modificada:

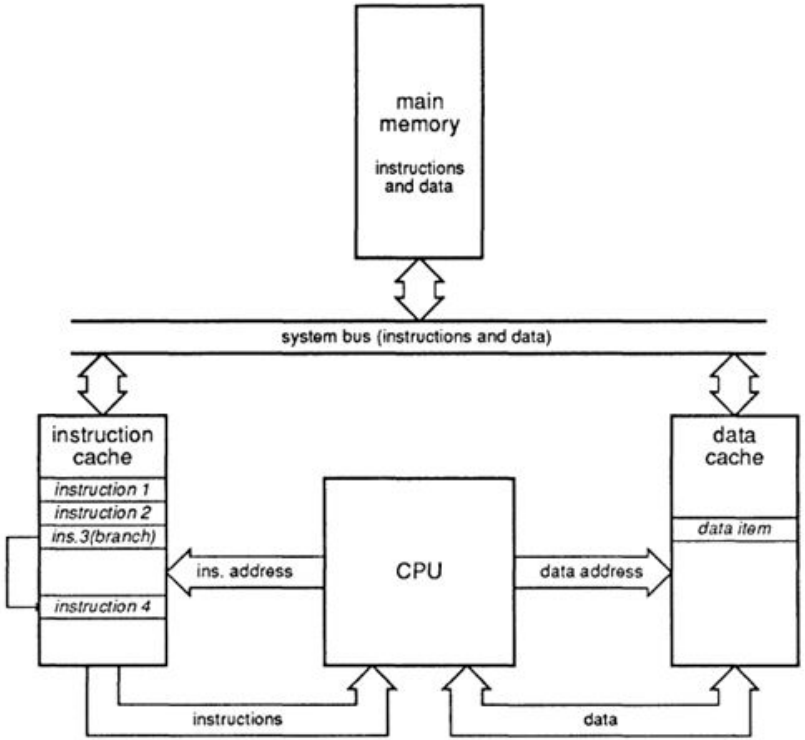

Figura 5. Arquitectura Harvard Modificada [3].

Instruction Set Architecture[5]

- Es parte abstracta del modelo de la computadora. Define cómo está controlada la computadora mediante el software.

- Es básicamente el manual del programador debido a que es la parte de la máquina que es visible para el programador de lenguaje ensamblador, el escritor del compilador y el programador de aplicaciones.

- El ISA define qué tipos de datos se admiten, qué registros, cómo el hardware administra la memoria principal, qué instrucciones puede ejecutar un micro procesador y el modelo de I/O.

Arquitectura CISC (Complex Instruction Set Computer)

- Es un diseño filosófico de arquitectura usado por cada miembro de la familia x86 de Intel.

- Principales características: gran número de instrucciones de longitud variable y diseños complejos.

- Mantenía las instrucciones cortas; sin embargo, también almacena grandes y complejas instrucciones.

- Contenía un gran número de instrucciones que accedían directamente de la memoria, siendo las complejas las que requieren miles de ciclos, añadiendo tiempo de ejecución innecesariamente.

Arquitectura CISC (Complex Instruction Set Computer)

- La labor del escritor de compiladores es generar una secuencia de instrucciones máquina para cada sentencia de alto nivel; sin embargo, las instrucciones complejas son difíciles de aprovechar ya que no se sabe cuales se ajustan perfectamente a la construcción del programa.

- Su objetivo era que se produjeran programas pequeños y rápidos; sin embargo, con el tiempo, las ventajas de estos objetivos no terminaban siendo tan importantes.

Arquitectura RISC (Reduced Instruction Set Computer)

- Busca reducir el número de instrucciones; sin embargo, su real objetivo es simplificar las instrucciones para que el programa se pueda ejecutar más rápido, por lo que solo load y store pueden acceder a memoria.

- Cada instrucción realiza una sola operación, todas tienen el mismo tamaño, solo tienen algunos diseños diferentes y todas las operaciones aritméticas deben realizarse entre registros.

- Ofrece un set de instrucciones menor que CISC.

CISC vs RISC

RISC	CISC
Multiples set de registros (más de 256)	Simple set de registro (6 a 16)
1 ciclo por instrucción (excepto load y store)	Múltiples ciclos por instrucción
Unidad de control cableado	Unidad de control micro-programada
Pocas instrucciones simples	Muchas instrucciones complejas
Pocos modos de direccionamiento	Muchos modos de direccionamiento
Solo load y store acceden a memoria	Muchas instrucciones acceden a memoria

Ejemplo:

A = B+C

CISC:

ADD mem(B) , mem(C), mem (A)

RISC:

load mem (B), reg(1)
load mem (C), reg(2)
add reg(1),reg(2), reg(3)
store reg(3), mem(A)

Referencias

[1] Null, L, et al. *The essentials of computer organization and architecture*. Jones & Bartlett Publishers, 2014.

[2] FURBER, S. *VLSI RISC architecture and organization*. Routledge, 2017.

[3] MITx: 6.004.2x Computation Structures 2: Computer Architecture

[4] http://ithare.com/modified-harvard-architecture-clarifying-confusion/

[5]https://www.arm.com/glossary/isa#:~:text=An%20Instruction%20Set%20Architecture%20(ISA,as%20how%20it%20gets%20done.

Q&A