

# Arquitectura de Computadores

Yadran Eterovic S. (yadran@ing.puc.cl)

# La inimaginablemente vital industria de la tecnología de la información

En los últimos 40 años, el desempeño de los computadores ha aumentado unas 50 mil veces:

- comparando el VAX 11/780 de 5 MHz con el Intel Core i7 de 4 núcleos de 4.2 GHz

Un teléfono celular que hoy cuesta menos de 500 dólares tiene el mismo desempeño que el computador más rápido del mundo de hace 25 años y que costaba 50 millones de dólares

Si la industria del transporte hubiera progresado a la misma tasa que la industria de la computación, hoy podríamos viajar de Nueva York a Londres en un segundo y por 10 pesos

# La tercera revolución de la civilización

## **La tercera revolución de la civilización:**

- la revolución de la agricultura (circa 10,000 a. C.)
- la revolución industrial (1760–1840)
- la “revolución de la información”, liderada por los computadores (actualmente)

## **La investigación científica:**

- científicos teóricos
- científicos experimentales
- científicos computacionales

## **En los últimos 25 – 30 años:**

- la *Web*
- los motores de búsqueda
- los teléfonos celulares/smartphones

## **... y estamos “ahí” de:**

- anteojos para realidad aumentada
- autos que se conducen a sí mismos
- la desaparición del dinero en efectivo

Los  
computadores  
son usados en  
varias clases  
disímiles de  
aplicaciones

Computadores personales (PCs)

Servidores

Computadores embebidos

Dispositivos móviles personales (PMDs: smart phones, tablets)

Computación “en la nube”

El computador personal (PC)  
—*notebook* o *laptop*— es el tipo de computador más comúnmente reconocible









# Sistemas embebidos

## Examples of Embedded Systems



Many Different Products Depend on Embedded Systems

*Cloud  
computing*  
(computación  
en la nube):

- Amazon
- Google
- Microsoft





# El hardware de todo computador ejecuta cuatro funciones básicas

Ingresar o recibir (input de) datos

Emitir o enviar (output de) datos

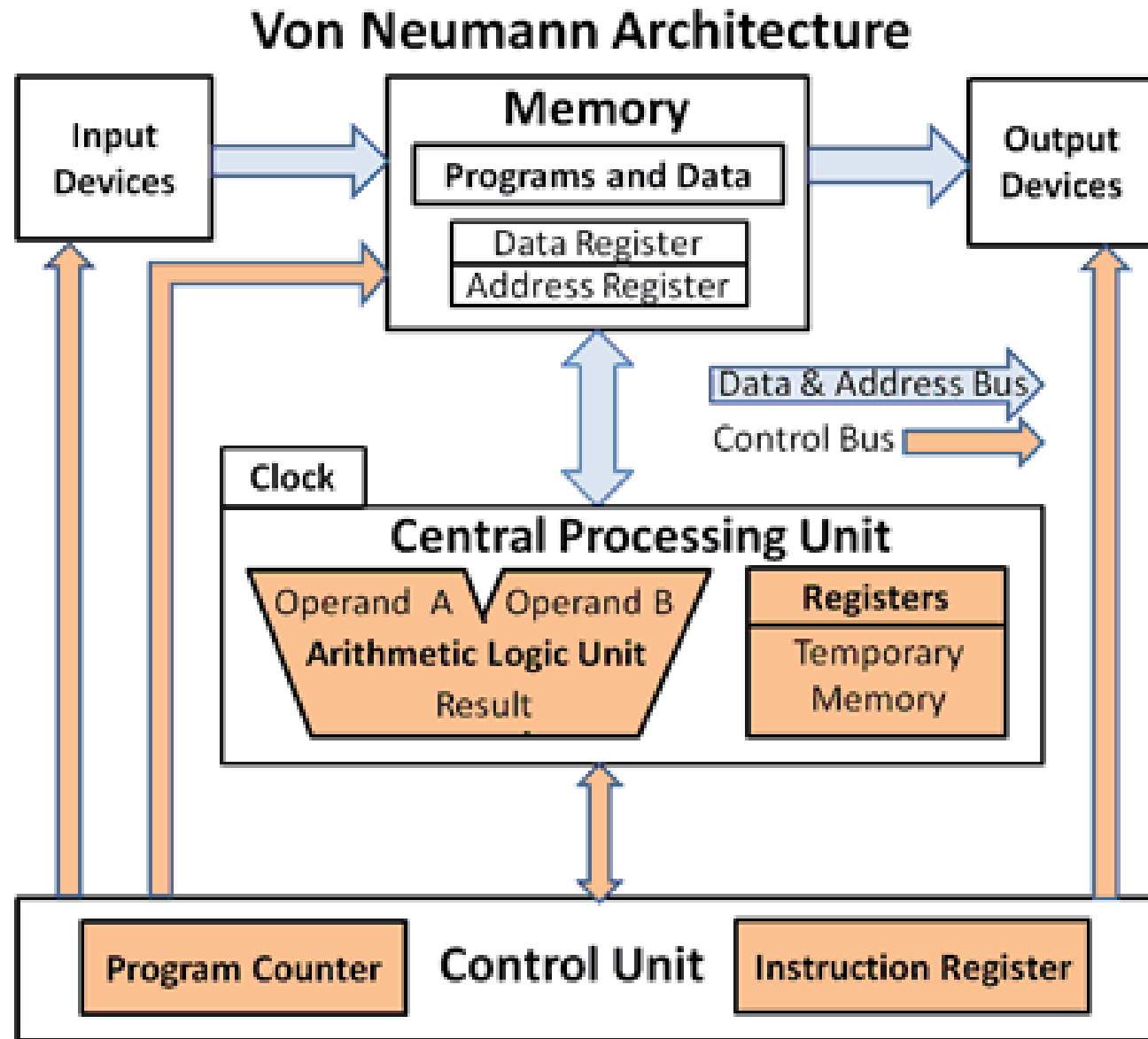
Procesar datos

Almacenar datos

**En este curso estudiaremos cómo se llevan a cabo estas funciones**

# Componentes clásicas de un computador:

- CPU
- control
- *datapath*
- memoria
- *input/output*



Una aplicación típica:

- millones de líneas de código
- + librerías de software que implementan funciones complejas

Pero ... el hardware de un computador sólo puede ejecutar instrucciones de “bajo nivel” muy simples:

- sumar dos números
- revisar un número para ver si es cero
- copiar datos desde una parte de la memoria a otra



Para “hablarle”  
al hardware  
electrónico,  
hay que enviar-  
le señales  
eléctricas

Las señales más simples y fáciles de distinguir son *on* y *off* :

... → el alfabeto del computador tiene solo dos letras, cuyos símbolos son los números 0 y 1

... → cada letra es un **dígito binario**, o **bit**

Los comandos —*instrucciones de máquina*— son secuencias de bits que el computador entiende y obedece:

- p.ej., 1000110010100000 le dice al computador que suma dos números

Los primeros programadores se comunicaban con los computadores usando directamente estas secuencias de dígitos binarios

El lenguaje  
simbólico es el  
*lenguaje de  
ensamble*, o  
**assembly**

El lenguaje bina-  
rio es el **lenguaje  
de máquina**

Luego, los programadores inventaron notaciones más cercanas a nuestra forma de pensar; p.ej.:

- **add A,B** podría ser la representación simbólica de la instrucción anterior

... e inventaron programas —el ensamblador o *assembler*— para traducir de esta notación simbólica a la binaria:

- el assembler traduce **add A,B** a 1000110010100000

Es decir, los programadores usaron el computador para que los ayudara a programar el propio computador

El compilador traduce un programa escrito en un lenguaje de alto nivel a instrucciones que el hardware puede ejecutar

```
swap(size_t v[], size_t k):  
    size_t temp  
    temp = v[k]  
    v[k] = v[k+1]  
    v[k+1] = temp
```

programa  
en lenguaje  
de alto nivel

programa en  
lenguaje de  
máquina binario

```
0000 ... 1000  
0000 ... 0001  
1000 ... 0000  
1000 ... 0100  
1010 ... 0000  
1010 ... 0100  
0000 ... 1000
```

programa en  
lenguaje  
assembly RISC-V

compilador

swap:

```
slli x6,x11,3  
add x6,x10,x6  
ld x5,0(x6)  
ld x7,8(x6)  
sd x7,0(x6)  
sd x5,8(x6)  
jalr x0,0(x1)
```

assembler



En la actualidad, varias capas de software organizadas jerárquicamente interpretan o traducen las operaciones de “alto nivel” a las instrucciones simples del computador:

- las capas de más “arriba”: aplicaciones, directamente disponibles para nosotros los usuarios
- **las capas intermedias: software de sistemas, en particular, sistemas operativos, compiladores, *loaders*, y *assemblers***
- la capa de más “abajo”: hardware

# El sistema operativo

Es la interfaz entre los programas de los usuarios y el hardware:

- p.ej., Linux, iOS, Windows

Proporciona servicios y funciones de supervisión:

- manejo de operaciones básicas de i/o
- asignación de almacenamiento y memoria
- compartimiento protegido del computador entre muchas aplicaciones que lo usan simultáneamente

# Contenido

Números enteros y de punto flotante: representación y operaciones

Lógica digital, circuitos combinacionales y la ALU

Circuitos secuenciales, registros y memoria

El computador básico: componentes, instrucciones y el *datapath*

La arquitectura del set de instrucciones (ISA)

*Pipelining*

Memoria principal, caches y memoria virtual

Input/output



# Las clases

Para terminar la clase puntualmente, avísenme cuando falten 3 – 5 minutos (para las 9.40)

Todo el material de clases (y la clase misma) será subido a la página del curso en Github

Estamos aprendiendo a hacer clases híbridas: cualquier sugerencia (herramienta, metodología, etc.) es bienvenida

# Evaluación

Tres pruebas, cada una vale 20% :

- I1: martes 28 sept., desde las 8.30 am
- I2: martes 16 nov., desde las 8.30 am
- I3: viernes 10 dic. (**presencial**), de 8.30 am a 11.30 am
- no hay exmen

Dos tareas, cada una vale 20% :

- los enunciados se publicarán las semanas del 20 sept. y del 8 nov., respectivamente
- cada una tendrá un plazo de 2 a 3 semanas

Este curso suscribe el **Código de Honor** de la universidad

<http://www.uc.cl/codigo-de-honor/>

Copias y otras faltas a la honestidad académica serán sancionados con nota final **NF = 1.1** en el curso



# Bibliografía

“Apuntes”, de Alejandro Echeverría y Hans Löbel, “Clases” y otros, disponibles en el sitio del curso

D.A. Patterson, J.L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface* (RISC-V ed.), Morgan Kaufmann (Elsevier) 2018

A.S. Tanenbaum, T. Austin, *Structured Computer Organization* (6th ed.), Pearson Education Limited 2013