



# Presentación

Arquitectura de Computadores – IIC2343

Yadran Eterovic S. ( [yadran@uc.cl](mailto:yadran@uc.cl) )

2025-2

Entre 1978 y 2018, el desempeño de los computadores aumentó 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 500 dólares tiene el mismo desempeño que el computador más rápido del mundo de hace 30 años

... y que costaba 50 millones de dólares

Si otras tecnologías hubieran progresado a la misma tasa, podríamos:

- volar de Nueva York a Londres en un segundo y por 10 pesos
- tener un auto capaz de correr a 30,000 km/h

... de rendir 400,000 km por litro

... y que cuesta 1,000 pesos

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

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

Componentes típicas de un computador:

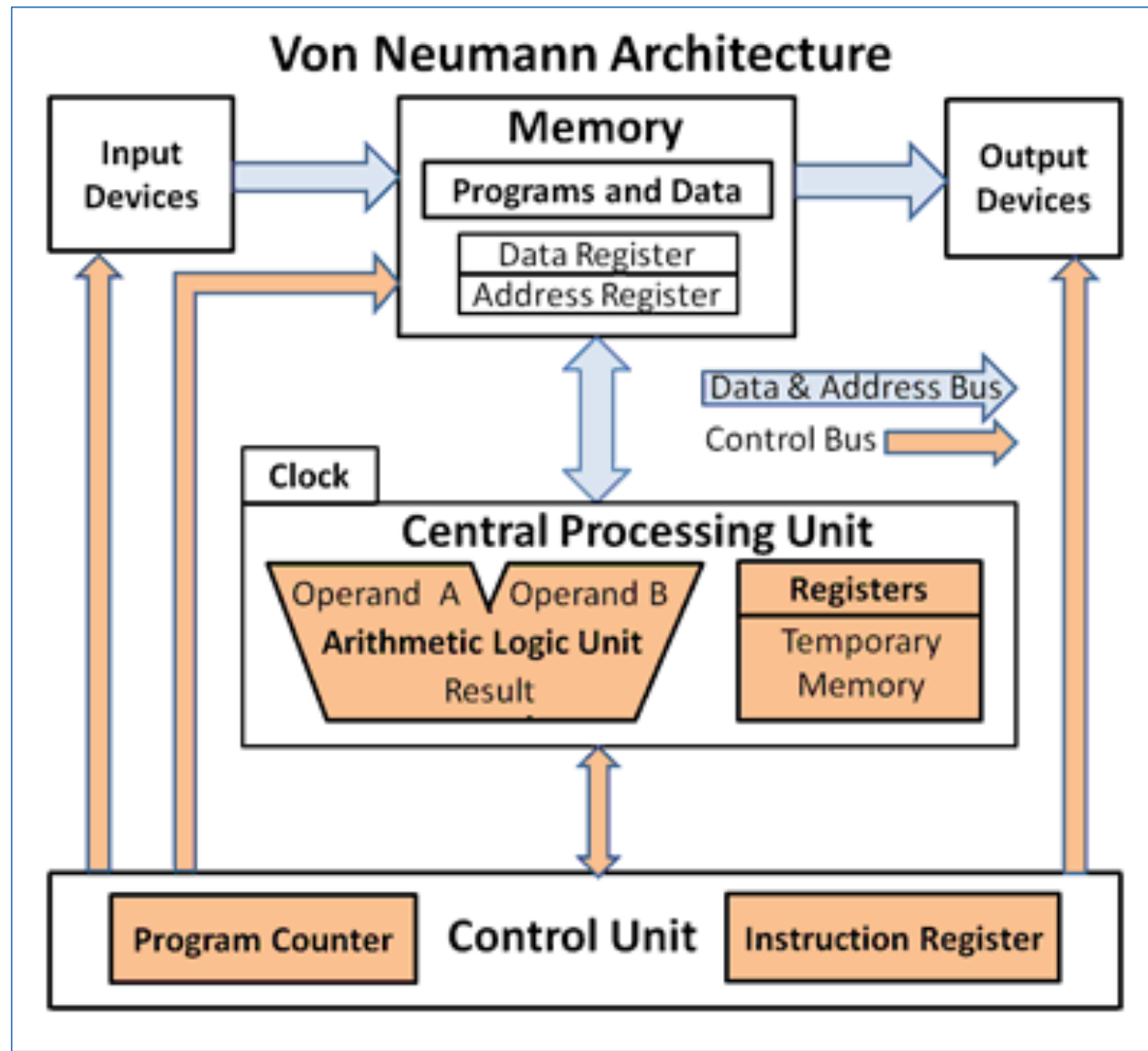
- *CPU*
- *control unit*
- memoria
- *input / output*
- *datapath*

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

Componentes típicos de un computador:

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



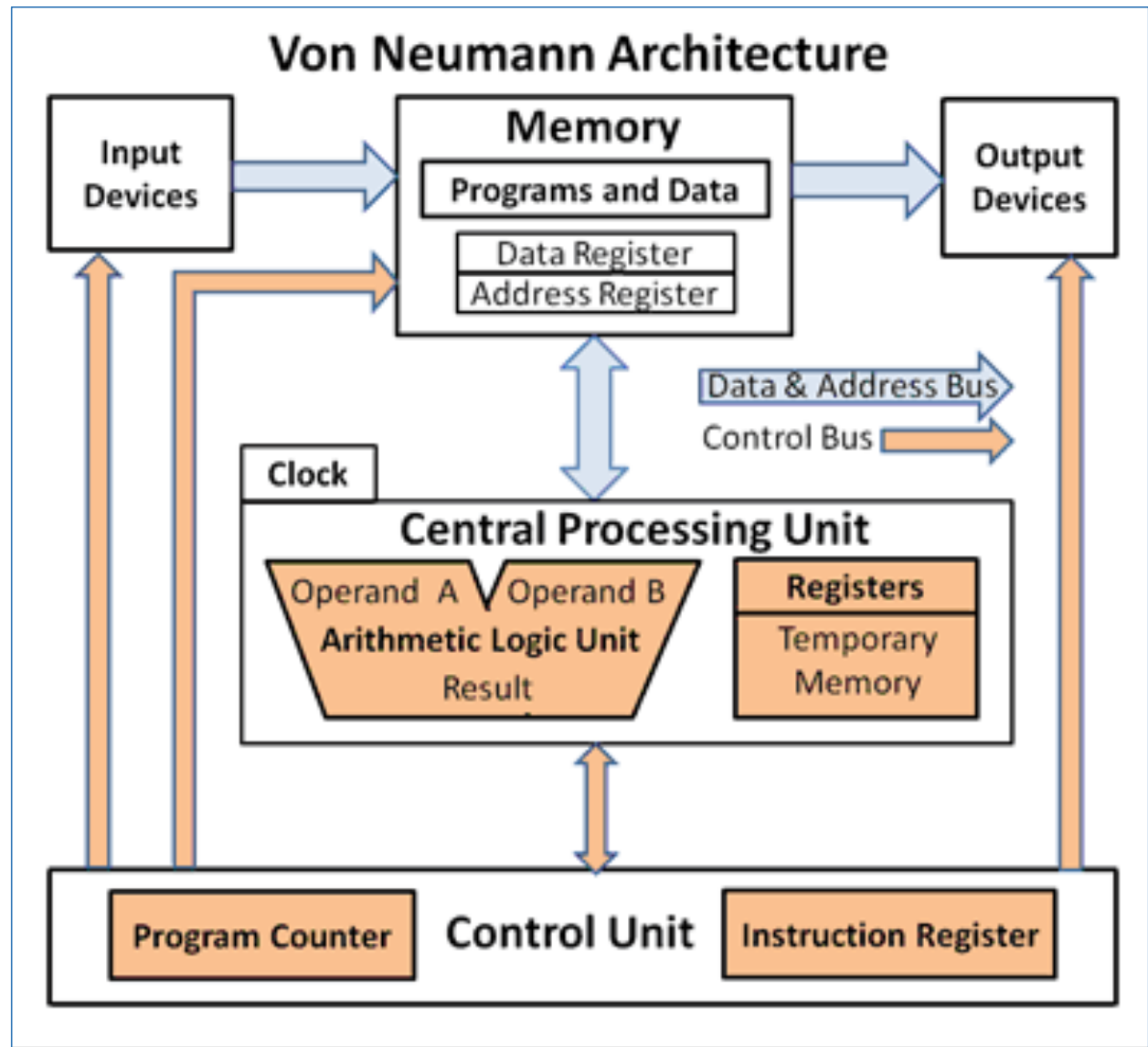
El *hardware* de **todo** computador ejecuta cuatro funciones básicas:

- ingresar o recibir (*input* de) datos
- emitir o enviar (*output* de) datos
- procesar datos
- almacenar datos

Estudiaremos cómo se llevan a cabo las cuatro funciones básicas ... y cómo funciona cada una de las componentes típicas

Componentes típicas de un computador:

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



## Temario

Números enteros y de punto flotante: representación, almacenamiento y operaciones

Lógica digital, circuitos combinacionales y la *ALU*

Circuitos secuenciales, registros y memorias

El computador básico: componentes, instrucciones y su ejecución, el *datapath*, control de flujo, y subrutinas

Control de flujo, subrutinas

La arquitectura del set de instrucciones (*ISA*)

*Pipelining*

Memoria principal, caches ( y memoria virtual )

*Input/output*: dispositivos, comunicación con *CPU* y memoria

Paralelismo (más allá de *pipelining*)

## Evaluación: Interrogaciones + proyecto + otros

*I*1: viernes 12 sept., 5.30 pm

*I*2: miércoles 29 oct., 5.30 pm + AP jueves 30/5, 9.20 am

*I*3: jueves 11 dic., 5.30 pm (fecha Examen)

Sea  $I = 0.3 \times I_1 + 0.3 \times I_2 + 0.3 \times I_3 + 0.1 \times AP$

$P = 3$  entregas de proyecto (22/sept., 20/oct., 17/nov.; c/u 30%)

+ 2 laboratorios (11/ago., 18/ago.; c/u 5%)

**Nota Final** =  $(I + P)/2$ , si  $I \geq 4.0$  y  $P \geq 4.0$

mín{ 3.9,  $(I + P)/2$  }, en otro caso



## Evaluación: Interrogaciones + proyecto + otros

*I*1: viernes 12 sept., 5.30 pm

*I*2: miércoles 29 oct., 5.30 pm + AP jueves 30/5, 9.20 am

*I*3: jueves 11 dic., 5.30 pm (fecha Examen)

Sea  $I = 0.3 \times I_1 + 0.3 \times I_2 + 0.3 \times I_3 + 0.1 \times AP$

$P = 3$  entregas de proyecto (22/sept., 20/oct., 17/nov.; c/u 30%)

+ 2 laboratorios (11/ago., 18/ago.; c/u 5%)

**Nota Final** =  $(I + P)/2$ , si  $I \geq 4.0$  y  $P \geq 4.0$   
mín{ 3.9,  $(I + P)/2$  }, en otro caso

Se puede faltar a sólo una interrogación  
La ausencia ser justificada ante la DiPre  
Se reemplaza con preguntas adicionales  
en la I3  
(si la ausencia justificada es a la I3,  
queda con Nota P)

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 ***Nota Final = 1.1*** en el curso

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 ***Nota Final = 1.1*** en el curso

Uso de IA

## Ayudantes

- **Coordinador**

Joaquín Peralta [jperaltaperez@uc.cl](mailto:jperaltaperez@uc.cl)

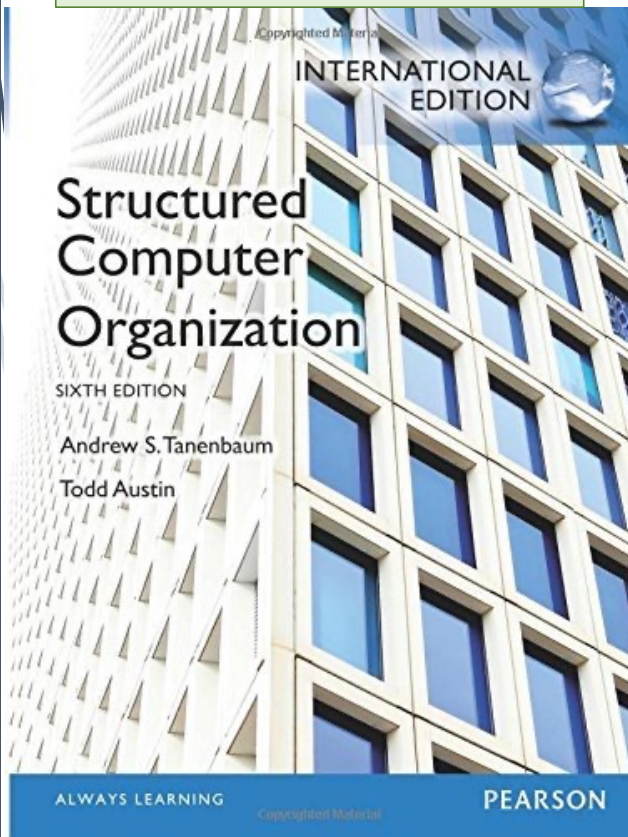
- **De bienestar**

Francisca Quijada [franciscaquijada@uc.cl](mailto:franciscaquijada@uc.cl)

- **Coordinadora de laboratorio**

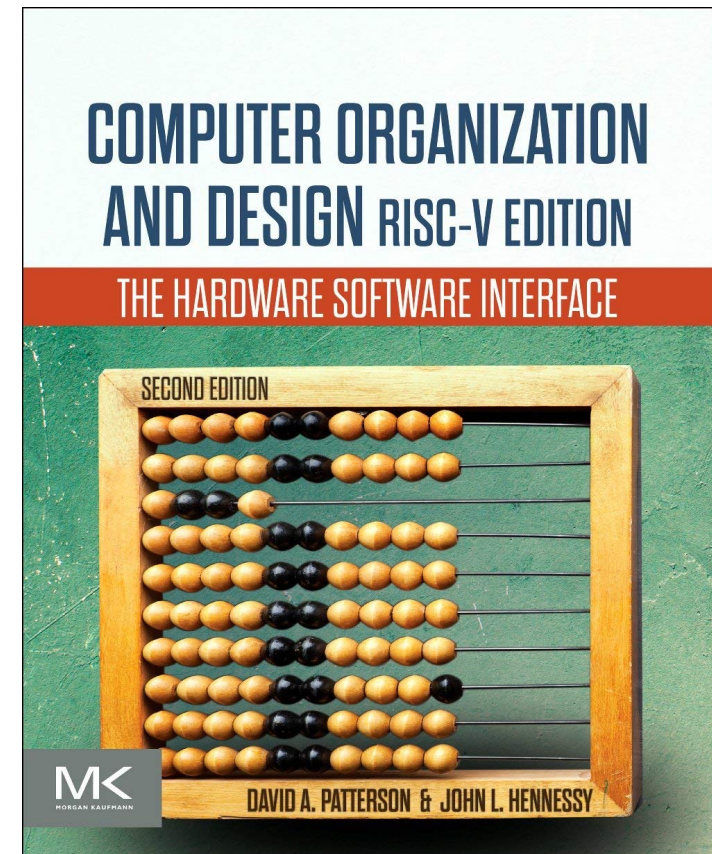
Javiera Pinto [jpints@uc.cl](mailto:jpints@uc.cl)

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



“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 RISC-V Edition:  
The Hardware Software Interface* (2nd  
ed.), Morgan Kaufmann (Elsevier) 2021



Los computadores son usados en varios tipos de aplicaciones

Computadores personales (PCs)

Servidores

Computadores embebidos

Dispositivos móviles personales (PMDs)

Computación “en la nube”

... pero todas comparten un conjunto común de tecnologías de hardware —que vamos a estudiar en este curso



El **computador personal (PC)** —notebook o laptop— es el tipo de computador más comúnmente reconocible: énfasis en buen desempeño para usuarios individuales a bajo costo



Los **servidores** están orientados a ejecutar altas cargas de trabajo—p.ej., una aplicación científica compleja, o muchos trabajos pequeños, en el caso de un servidor web—y en general son accesibles sólo a través de una red: misma tecnología que los PCs, pero mucha mayor capacidad de computación, almacenamiento e I/O + énfasis en confiabilidad



Van desde (un poco más que) un computador de escritorio (sin pantalla ni teclado), de mil dólares ... hasta un **supercomputador**, con cientos de miles de procesadores y muchos terabytes de memoria, de cientos de millones de dólares



Los **computadores embebidos** están dentro de otro dispositivo y ejecutan una aplicación predeterminada

Son la clase más grande de computadores y abarcan el rango más amplio de tipos de aplicaciones y de desempeños

## Examples of Embedded Systems



**Many Different Products Depend on Embedded Systems**

Los **PMDs** son dispositivos inalámbricos pequeños, sin teclado ni mouse —p.ej., smart phones y tablets— que se conectan a la Internet, usan baterías, y se les instala software mediante la descarga de apps



By Yutaka Tsutano from Lincoln, United States - iPad & iPhoneUploaded by Mewtu, CC BY 2.0, <https://commons.wikimedia.org/w/index.php?curid=10109134>



**Cloud computing** (computación “en la nube”) es la versión moderna de los servidores convencionales, consistente en datacenters gigantes, llamados Warehouse Scale Computers (WSCs).

P.ej., Amazon, Google, Microsoft construyen WSCs con decenas de miles de servidores, y arriendan parte de esta capacidad a compañías que a su vez ofrecen software como servicio (SaaS) a PMDs (una parte de la aplicación corre en la nube y la otra en el PMD)



Primero

... lo más básico sobre la representación de  
datos en un computador

⇒ la **representación de números** (enteros)

Cuando se trata de números, usamos  
*base 10*: **números decimales**

Cuando se trata de números, usamos  
*base 10: números decimales*

Usamos diez símbolos diferentes:

- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 —*dígitos decimales*, o *dígitos*

... y usamos *notación posicional*:

- el valor de un dígito en un número depende tanto del dígito mismo  
... como también de su posición dentro del número

Cuando se trata de números, usamos  
*base 10: números decimales*

Usamos diez símbolos diferentes:

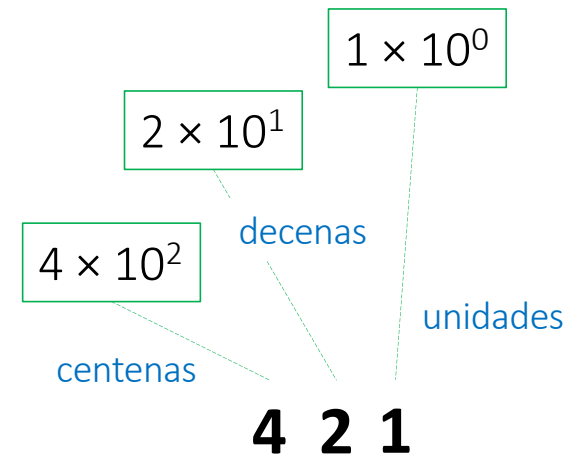
- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 —*dígitos decimales*, o *dígitos*

... y usamos *notación posicional*:

- el valor de un dígito en un número depende tanto del dígito mismo  
... como también de su posición dentro del número

P.ej., cuando escribimos **421** hablamos del número (o valor numérico) que resulta de la siguiente operación:

$$4 \times 10^2 + 2 \times 10^1 + 1 \times 10^0$$



Los números también pueden ser representados en *base 2*: **números binarios**



Los números también pueden ser representados en *base 2*: **números binarios**

Usamos sólo dos símbolos diferentes:

- 0 y 1 —*dígitos binarios* o **bits**

... y la misma notación posicional

Los números también pueden ser representados en *base 2*: **números binarios**

Usamos sólo dos símbolos diferentes:

- 0 y 1 —*dígitos binarios* o **bits**

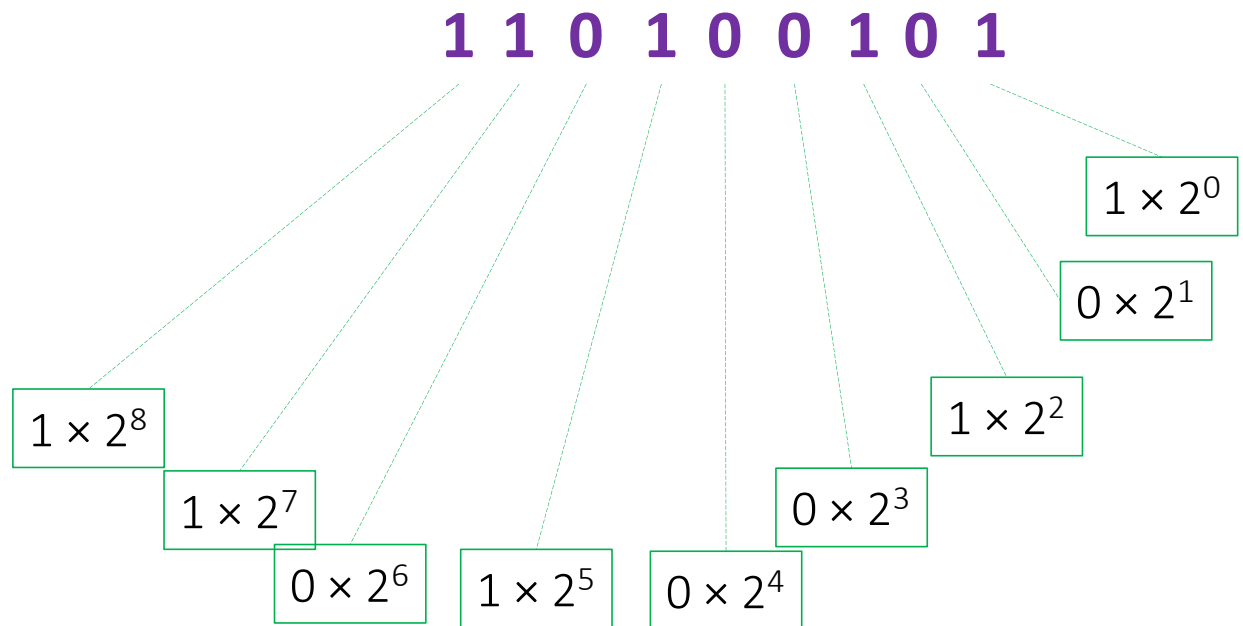
... y la misma notación posicional

P.ej., **421** en base 2 se representa así

**1 1 0 1 0 0 1 0 1**

... ya que

$$\begin{aligned} &1 \times 2^8 + 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 \\ &+ 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 421 \end{aligned}$$



Los **bits** son los *átomos* de la computación

Toda información en formato “computacional” se compone sólo de bits

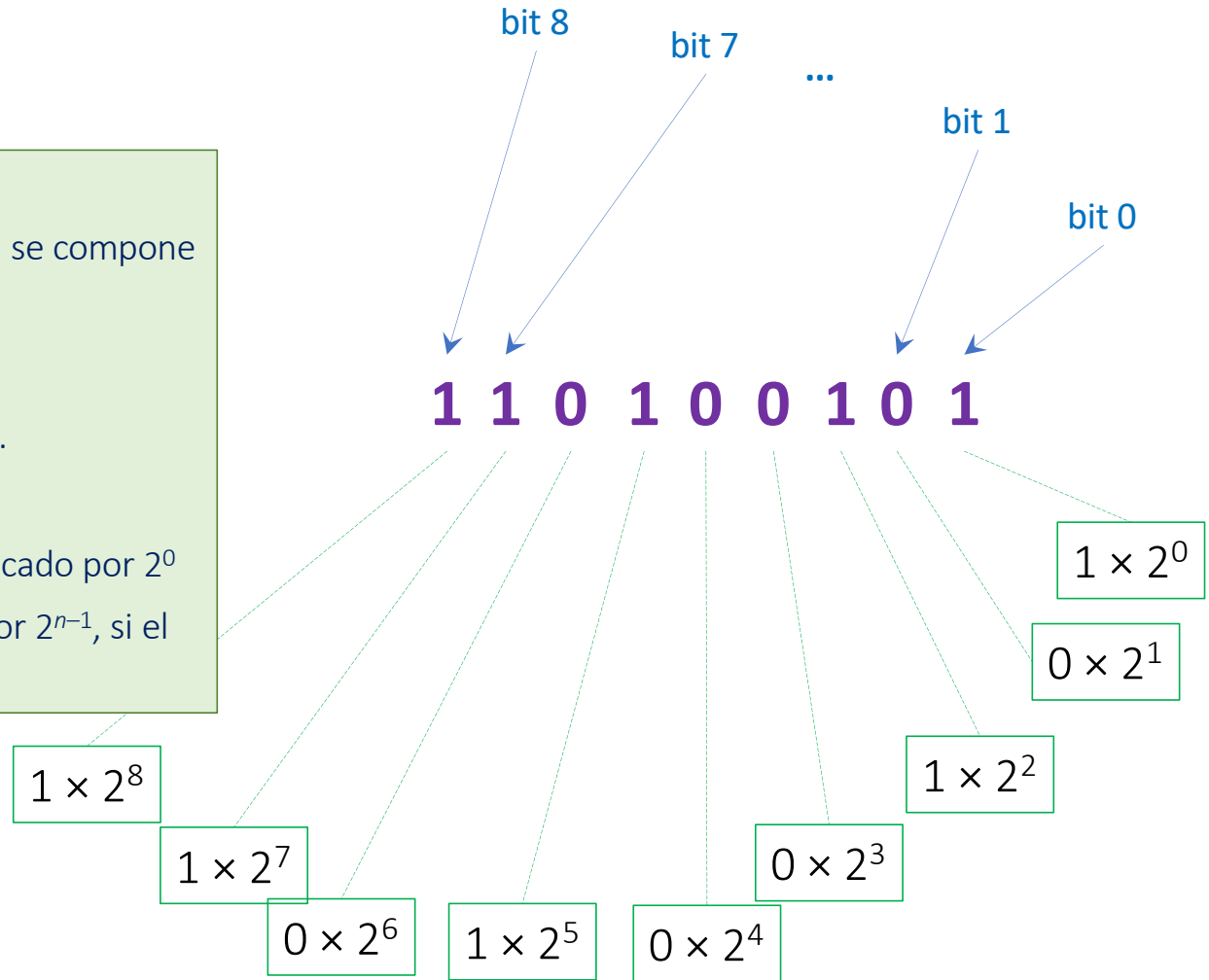
Numeramos los bits de un número binario

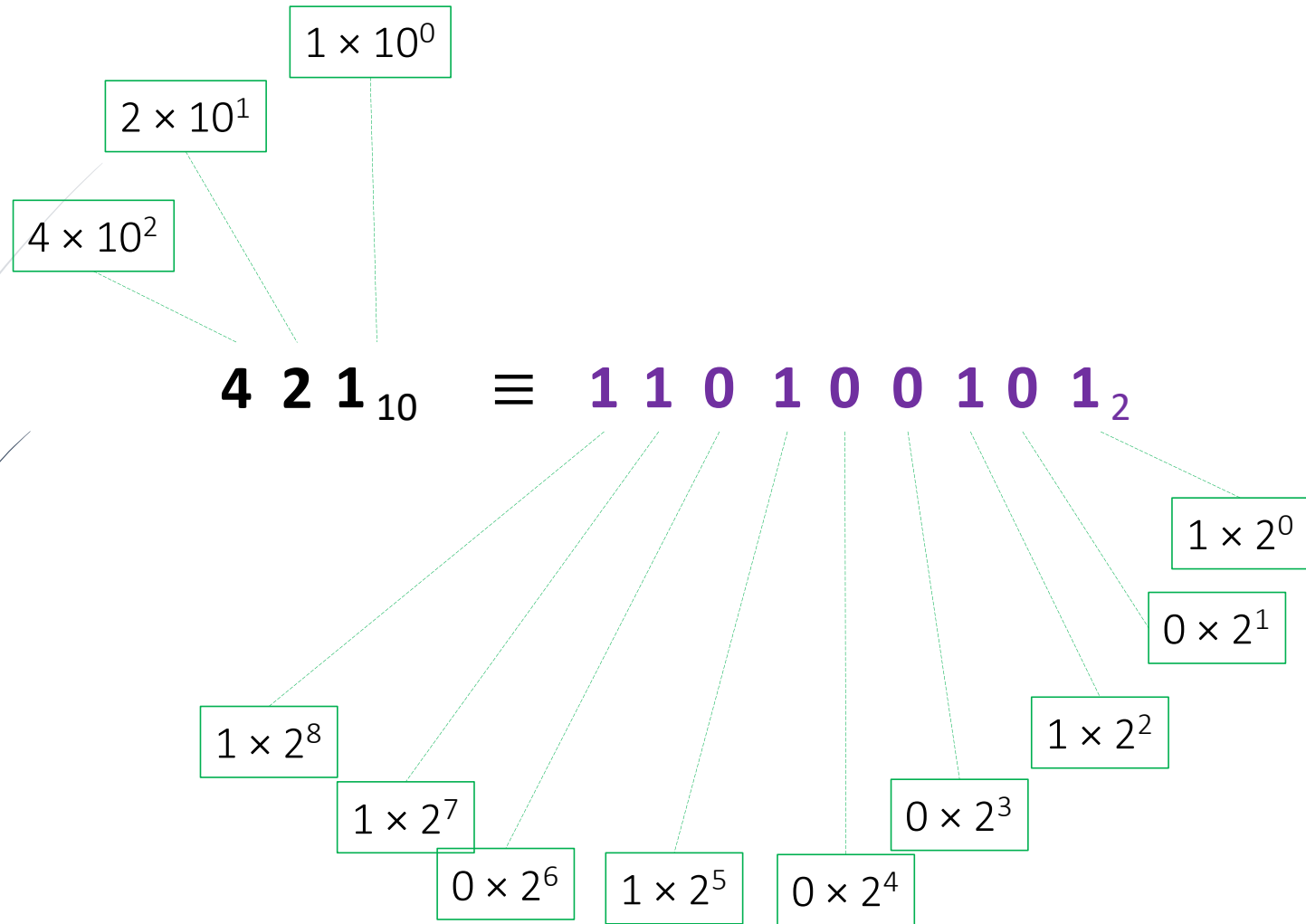
el bit 0, el bit 1, el bit 2, el bit 3, ...

... de derecha a izquierda:

⇒ desde el **bit menos significativo** —multiplicado por  $2^0$

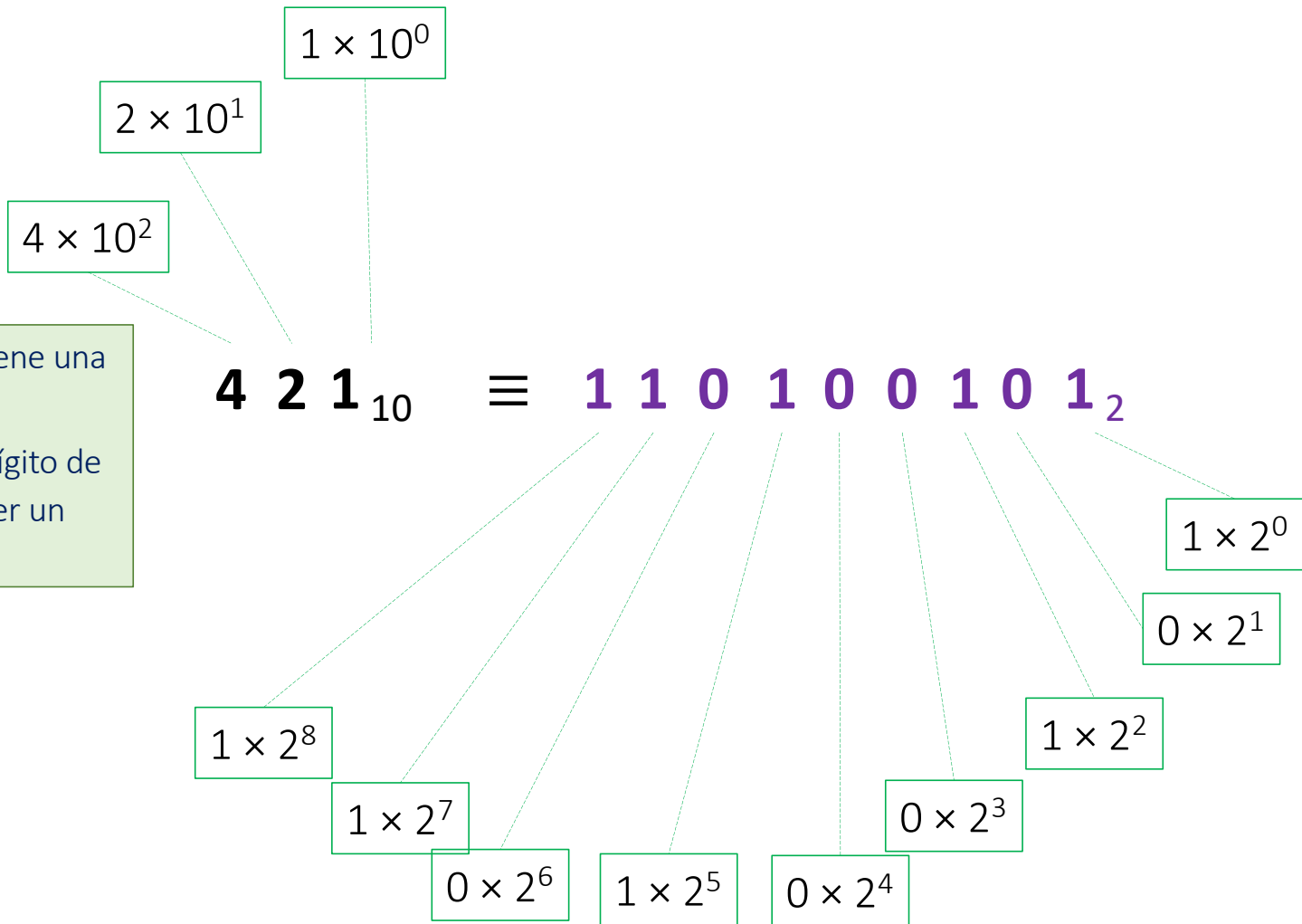
... al **bit más significativo** —multiplicado por  $2^{n-1}$ , si el número tiene  $n$  bits





Un número (un valor numérico) tiene una única representación en base 10:

- p.ej., si cambiamos cualquier dígito de 421, el nuevo número va a tener un valor numérico distinto de 421



Un número (un valor numérico) tiene una única representación en base 10:

- p.ej., si cambiamos cualquier dígito de 421, el nuevo número va a tener un valor numérico distinto de 421

$$1 \times 10^0$$

$$2 \times 10^1$$

$$4 \times 10^2$$

**4 2 1**<sub>10</sub>

≡

**1 1 0 1 0 0 1 0 1**<sub>2</sub>

... y una única representación en base 2:

- p.ej., si en 1 1 0 1 0 0 1 0 1 cambiamos cualquiera de los 1's por 0's o cualquiera de los 0's por 1's, el valor numérico del número binario resultante va a ser distinto de 421

$$1 \times 2^0$$

$$0 \times 2^1$$

$$1 \times 2^2$$

$$0 \times 2^3$$

$$0 \times 2^4$$

$$1 \times 2^5$$

$$0 \times 2^6$$

$$1 \times 2^7$$

$$1 \times 2^8$$

Los números pueden ser representados *en cualquier base*, usando la misma notación posicional