Sistemas de Representación

Organización del Computador 1
1er Cuatrimestre 2018

0s y 1s

- Los computadores comprenden el lenguaje de los números
- La organización de un computador depende (entre otros factores) del sistema de representación numérica adoptado
- Se trabaja con el sistema binario, de donde proviene el término **bit** como contracción de "**<u>bi</u>**nary digi**<u>t</u>**"

Sistemas de Numeración

- Un ***sistema de numeración*** es un conjunto de símbolos y un conjunto de reglas de combinación de dichos símbolos que permiten representar los números enteros y/o fraccionarios.

Sistemas posicionales

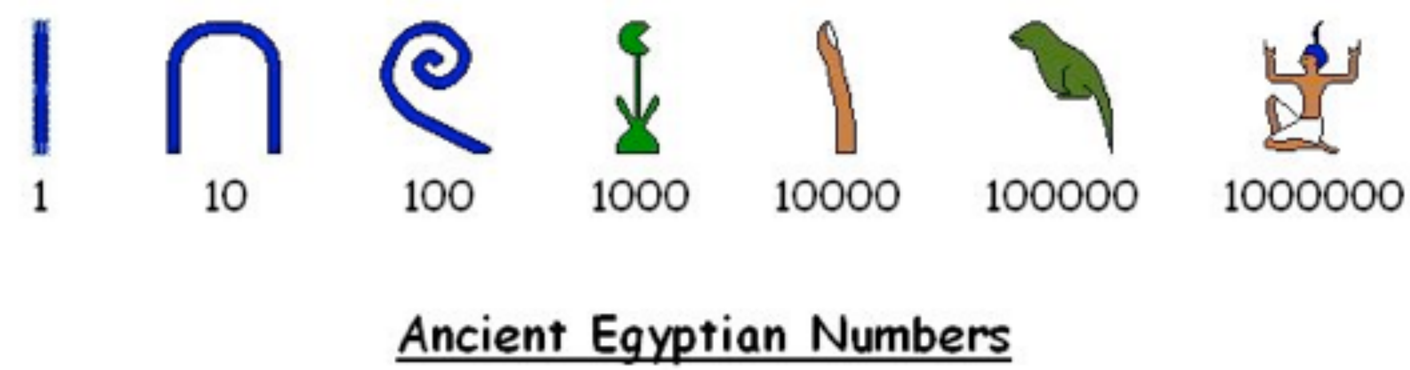
- Dentro de los sistemas de numeración posibles un conjunto importante, destacado, es el constituido por los sistemas de numeración **posicionales** .

- Numeración posicional: cada dígito posee un valor diferente que depende de su posición relativa

Un poco de historia

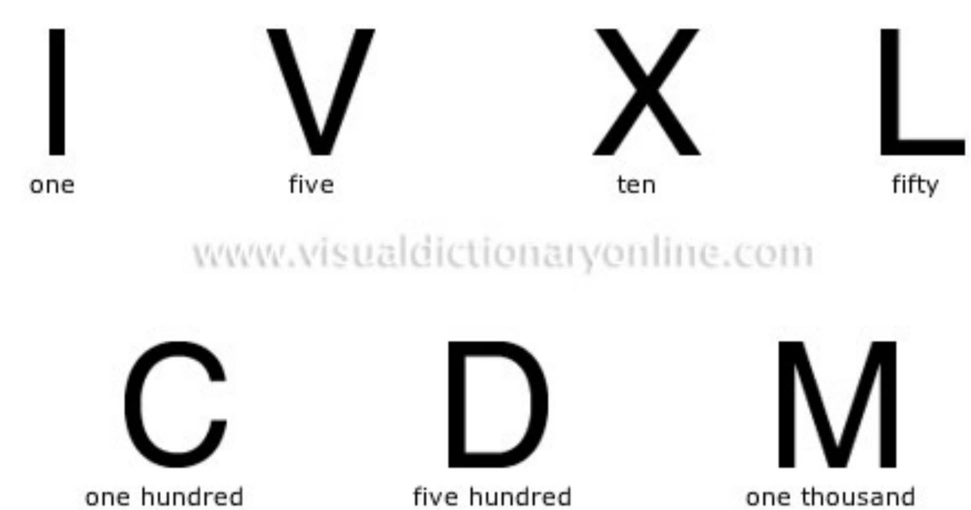
En muchas culturas se adoptó la base 10. Sin embargo, existen otras bases utilizadas

- Los Egipcios (4000-3000 B.C.)
  - No posicional
  - Base 12
  - Base 10:
    - 1 al 9: palitos
    - 10: U invertida
    - 100: soga enrollada
    - 1000: una flor (lottus blossom)



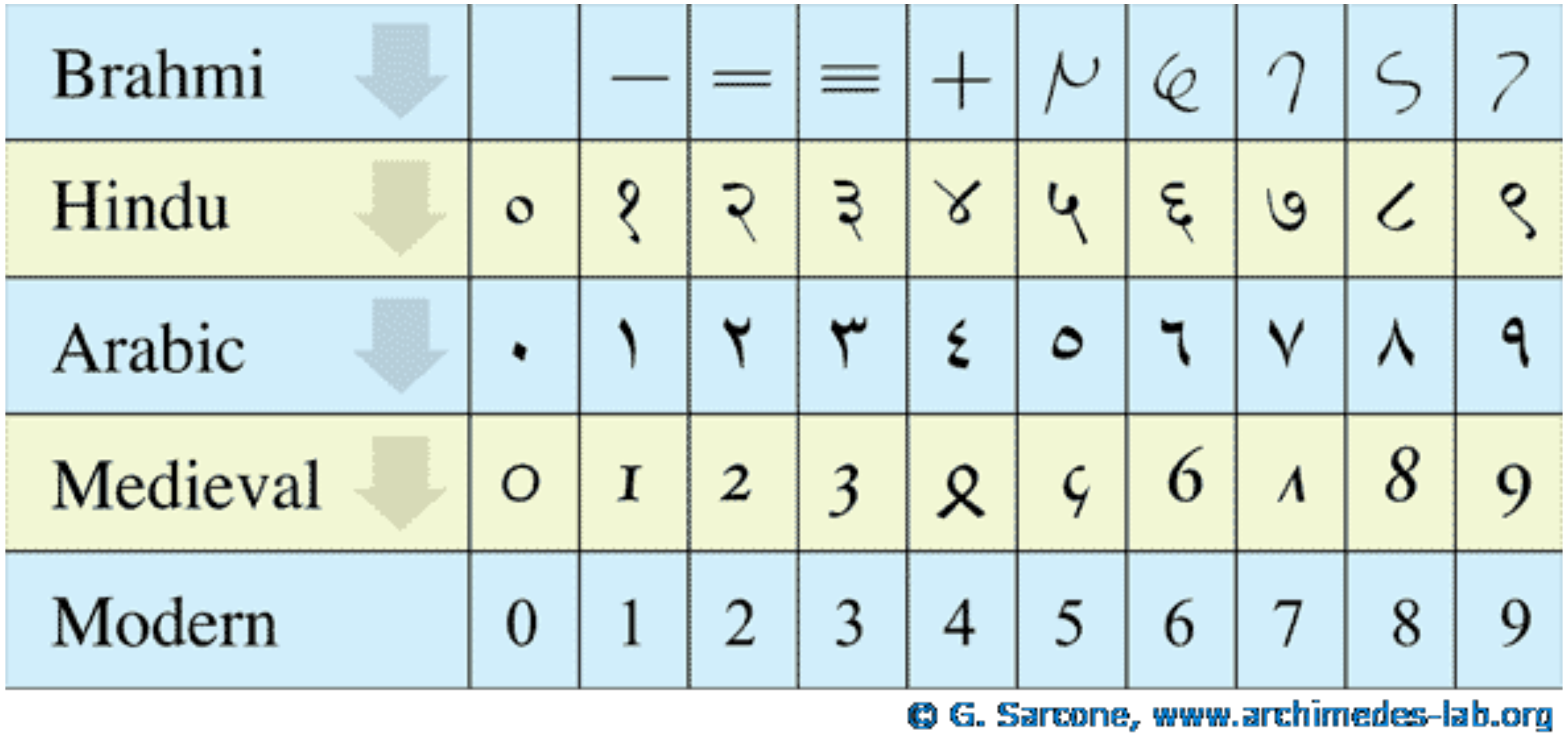
Un poco de historia

- Los Babilonios (1800-1900 B.C.)
  - Primer sistema posicional
  - Base 60!!!
  - No tenían cero.
- Los Romanos (100 B.C. - 400 A.D.)
  - No posicional
  - Base 10
  - Sin cero



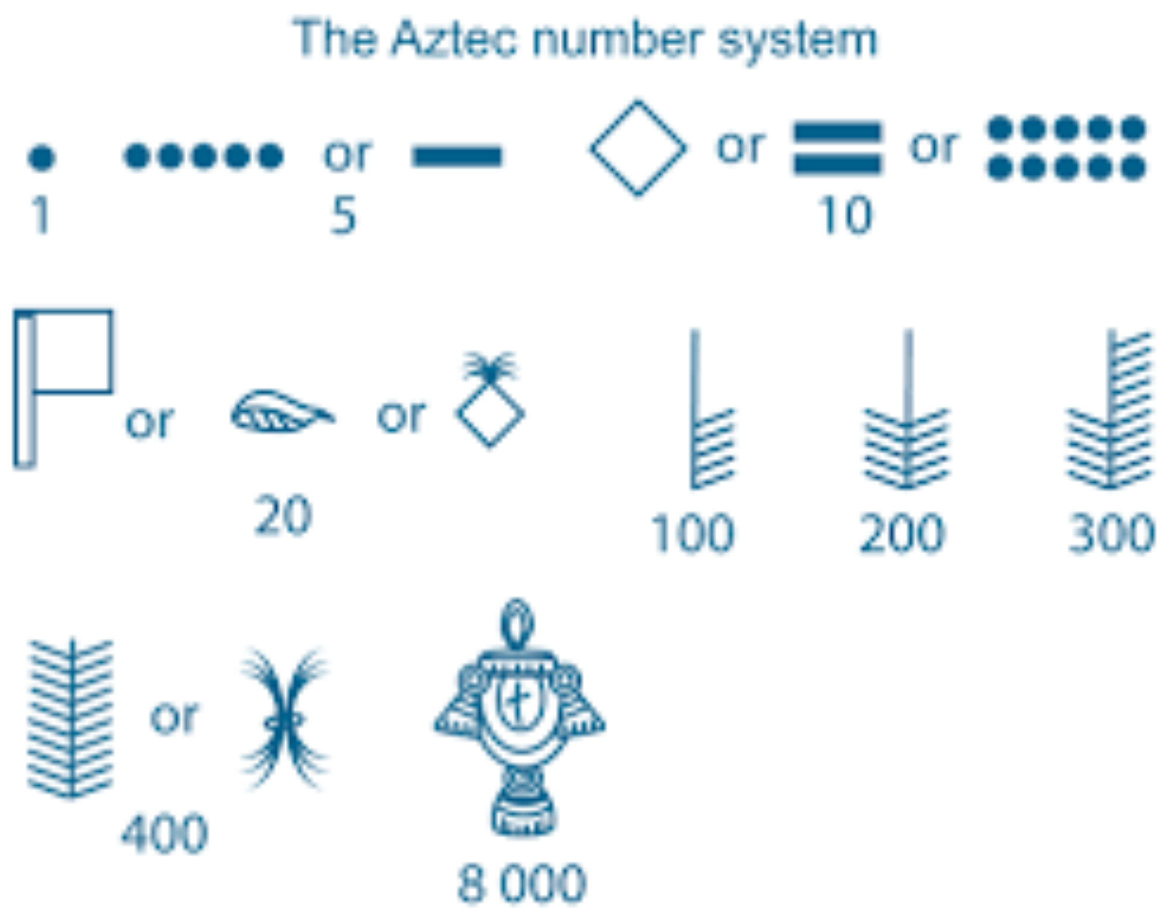
Un poco de historia

- Sistema Arábigo (India y Arabia, 1500 B.C.)
    - Sistema posicional
    - Base 10!!!
    - Tenían cero!



Cruzando el charco

- Cruzando el charco:
  - Los Aztecas (1200 A.D.)

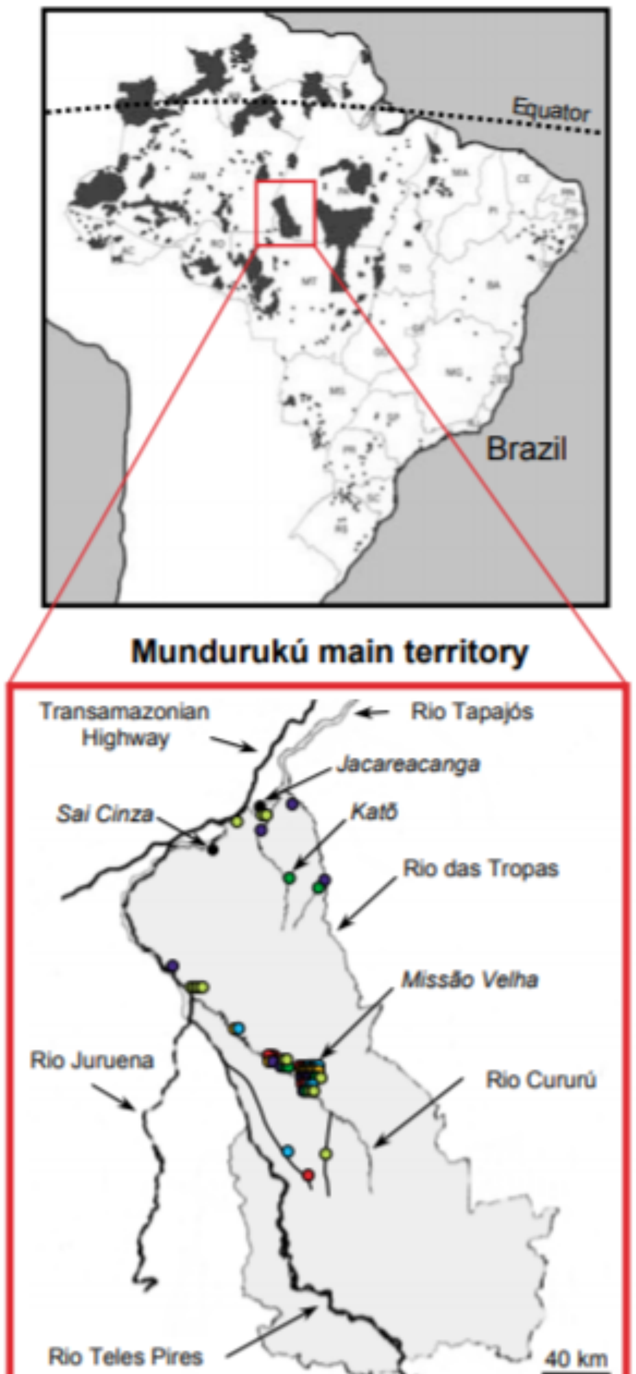


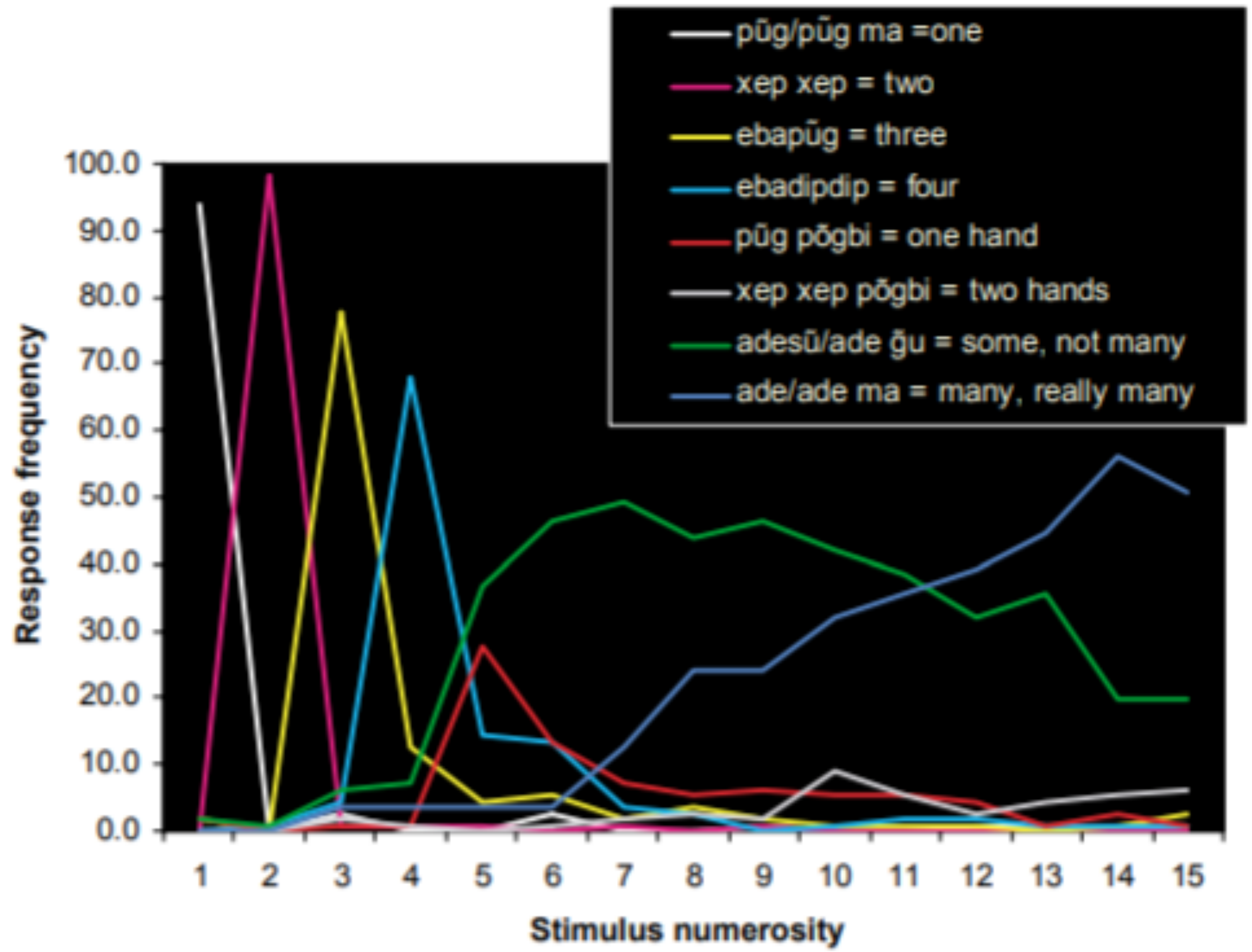
Muchos sistemas, un soporte

Sistemas de numeración en humanos

Science
AAAS

Exact and Approximate Arithmetic in an Amazonian Indigene Group
Pierre Pica *et al.*
Science **306**, 499 (2004);
DOI: 10.1126/science.1102085





Formalizando

- Queremos estudiar los sistemas de representación y su soporte
  - Un **número**: una cantidad
  - Un **numeral**: una representación de un número
  - Un **dígito**: símbolos del numeral

Formalizando

- En un sistema de numeración posicional de base ***b***, la representación de un número se define a partir de la regla:

$(\ldots a_3\, a_2\, a_1\, a_0\, .\, a_{-1}\, a_{-2}\, a_{-3}\, \ldots)_b =$

$$\ldots + a_2 b^2 + a_1 b^1 + a_0 b^0 + a_{-1} b^{-1} + a_{-2} b^{-2} + a_{-3} b^{-3} + \ldots$$

- Donde b es un entero no negativo mayor a 1 y cuando los ai pertenecen al conjunto de enteros en el rango $0 \leq a_i < b$

- El punto que aparece entre los dígitos a_0 y a_{-1} se denomina "*punto fraccionario*".

- Cuando b = 10 se lo llama *punto decimal*

- Cuando b = 2, se lo llama *punto binario*.

Ejemplos de sistemas de numeración

- **Sistema Decimal**: Es el sistema de numeración utilizado en la vida cotidiana, cuya base es diez, utilizando los símbolos 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9 .

- **Sistema Binario:** los dos símbolos utilizados son el 0 y el 1, los que reciben el nombre de bit (**bi**nary digi**t**).

- **Sistema Octal**: de base 8, los símbolos utilizados son 0, 1, 2, 3, 4, 5, 6,7.

- **Sistema Hexadecimal**: de base 16, los símbolos utilizados son 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Ejemplos

- $(243.51)_{10} =$

  $2 * 10^2 + 4 * 10^1 + 3 * 10^0 + 5 * 10^{-1} + 1 * 10^{-2}$

- $(212)_3 =$

  $2 * 3^2 + 1 * 3^1 + 2 * 3^0 = 23_{10}$

- $(10110)_2 =$

  $1 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0 = (22)_{10}$

Cambio de Base

Representemos $(104)_{10}$ en base 3

- Método de restas sucesivas

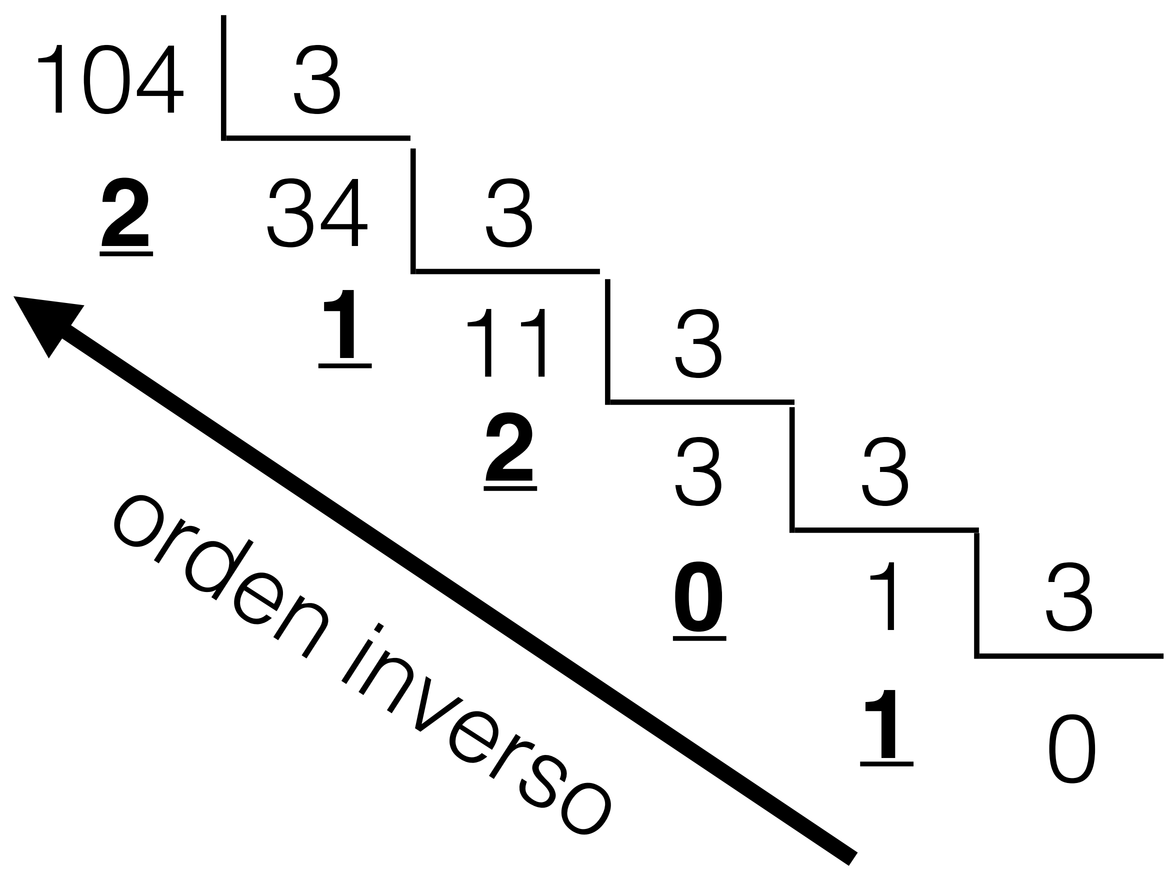
$$
\begin{array}{r l}
104 & \\
\underline{-81} & = 3^4 \times 1 \\
23 & \\
\\
\underline{-0} & = 3^3 \times 0 \\
23 & \\
\\
\underline{-18} & = 3^2 \times 2 \\
5 & \\
\\
\underline{-3} & = 3^1 \times 1 \\
2 & \\
\\
\underline{-2} & = 3^0 \times 2 \\
0 &
\end{array}
$$

$(104)_{10}=(10212)_3$

Cambio de Base

Representemos $(104)_{10}$ en base 3

- Método del resto de los cocientes



$(104)_{10}=(10212)_3$

Bajando a la compu..

- Stallings: "*…máquina digital…*"
  - Sistema **Binario**
- Precisión fija:
  - La computadora opera con sistemas de numeración de **tamaño fijo**
  - No podemos representar todos los números
  - El orden de las operaciones altera el resultado

Enteros en la compu…

- Tenemos que elegir cuáles números queremos representar y con qué sistema
    - Sin signo (sólo positivos y cero)
    - Magnitud con Signo (signed magnitude)
    - Complemento a 2
    - Exceso a M

Magnitud con signo

- **Signed magnitude**: El bit más a la izquierda es usado como indicador de signo.
  - 1 implica que el número es negativo,
  - 0 que es positivo.
- Si usamos **8 bits**, podemos representar el intervalo cerrado $[-(2^7-1), 2^7-1]$.
  - En general, con **n bits** podremos representar el intervalo cerrado $[-(2^{(n-1)}-1), 2^{(n-1)}-1]$.

Magnitud con signo

- La suma es igual que en el sistema decimal, incluyendo el concepto de acarreo.

- En caso de llegar con un acarreo al bit 8, nos encontramos en una situación de **overflow**.

Last carry	1 ←	1 1 1 1	⇐ carries
overflows and is	0	1 0 0 1 1 1 1	(79)
discarded.	0 +	1 1 0 0 0 1 1	+ (99)
	0	0 1 1 0 0 1 0	(50)

Sistemas de Complemento

- **Sistemas de complemento**: El *complemento de un número* se obtiene restando dicho número al número más grande que puede representarse con el tamaño de numeral con que contamos.

- La idea es usar los números "más altos" como números negativos.

- ¿Cuál es el complemento de $(-52)_{10}$ si tenemos 3 dígitos base 10?

  - Máximo numeral = $(999)_{10}$

  - Complemento = $(999)_{10} + (-52)_{10} = (947)_{10}$

Complemento a 2

- "Complemento a 2” es por "radix complement" en base 2.
  - Dado M numerales
    - M/2 se utilizan para representar números negativos
    - M/2 se utilizan para representar números positivos y el número cero

Complemento a 2

- Con n-bits,
  - ¿cuántos numerales distintos tengo?
  - ¿cuántos números positivos puedo representar?
  - ¿cuántos números negativos puedo representar?

Complemento a 2

- Con n-bits,
  - ¿cuántos numerales distintos tengo?
    - Rta: 2^n
  - ¿cuántos números positivos puedo representar?
    - Rta: $2^{(n-1)}-1$
  - ¿cuántos números negativos puedo representar?
    - Rta: $2^{(n-1)}$

Complemento a 2 de 8 bits

- ¿Puedo representar al número 0? ¿Cuál es el numeral binario que lo representa?
- ¿Cuál es el número más grande que puedo representar?
  - ¿Cuál es el numeral binario que lo representa?
- ¿Cuál es el número más negativo que puedo representar?
  - ¿Cuál es el numeral binario que lo representa?

Complemento a 2 de 8 bits

- ¿Puedo representar al número 0? **Rta**: Sí
  - ¿Cuál es el numeral binario que lo representa? **Rta**: $(00000000)_2$
- ¿Cuál es el número más grande que puedo representar?
  - **Rta**: $2^7-1 = (128)_{10} -1 = (127)_{10}$
  - ¿Cuál es el numeral binario que lo representa? **Rta**: $(0111\ 1111)_2$
- ¿Cuál es el número más negativo que puedo representar?
  - **Rta**: $-2^7=-(128)_{10}$
  - ¿Cuál es el numeral binario que lo representa? **Rta**: $(1000\ 0000)_2$

Complemento a 2 de 8 bits

- ¿Puedo representar el $(-1)_{10}$?
  - ¿Qué numeral lo representa?

Complemento a 2 de 8 bits

- ¿Puedo representar el $(-1)_{10}$?
- **Rta**: dado que el menor número es $-(128)_{10}$ y el mayor $(127)_{10}$, el $(-1)_{10}$ se puede representar.
  - ¿Qué numeral binario lo representa? Para calcular el *complemento a 2* de un número negativo debo restarlo al mayor número representable y sumarle 1.
    - **Rta**: $(255)_{10} - 1_{10} + 1 = (255)_{10}$
    - El numeral binario de $(255)_{10}$ es $(1111\ 1111)_2$

Notación exceso

- Dado un número *N*, en notación exceso a *M* se representa como *N*+*M*

- Por ejemplo, en notación exceso a 64 de 8 bits,

  - El número $(35)_{10}$ se representa como $(64)_{10}+(35)_{10}=(99)_{10}$

  - El número $(-35)_{10}$ se representa como $(64)_{10}-(35)_{10}=(29)_{10}$

Representaciones

- En gral., podemos pensar a cada representación de n-bits como una función matemática **parcial**
    - Dominio: Números Enteros
    - Imagen: Números positivos y el cero

Representaciones de n-bits

- sin_signo(x) = x si $0 \leq x \leq 2^n-1$
- complemento_a_2(x) =
  - x si $0 \leq x \leq 2^{(n-1)}-1$
  - $2^n - |x|$ si $-2^{(n-1)} \leq x < 0$
- exceso_M(x) = x+M si $-M \leq x \leq 2^n - M$
- signo_magnitud(x) = … Ejercicio

Suma/Resta Complemento a 2

- La suma se opera **exactamente igual** a la suma de números sin signo.

- La resta, al igual que vimos antes se reduce a la suma del complemento del sustraendo.

- Detectando overflow: Si el acarreo sobre el bit de signo es igual al acarreo fuera de dicho bit, no hay overflow. En caso que sean distintos sí lo hay.

	0←	1	1	1	1					⇐ carries
Discard last		0	1	1	1	1	1	1	0	(126)
carry.	+	0	0	0	0	1	0	0	0	+(8)
		1	0	0	0	0	1	1	0	(–122???)

Ejercicios

- Realizar la suma binaria de los números $(5)_{10}$ y $(-1)_{10}$ representados en complemento a 2 de 4 bits.
    - ¿El resultado es representable?
- Realizar la suma binaria de los números $(5)_{10}$ y $(3)_{10}$ representados en complemento a 2 de 4 bits.
    - ¿El resultado es representable?

Ejercicios

- Realizar la suma binaria de los números $(5)_{10}$ y $(-1)_{10}$ representados en complemento a 2 de 4 bits.
  - $(5)_{10}$ se representa con el numeral binario $(0101)_2$ en notación complemento a 2 de 4 bits.
  - $(-1)_{10}$ se representa con el numeral binario $(1111)_2$ en notación complemento a 2 de 4 bits.
  - La suma binaria es $(0101)_2 + (1111)_2 = (10100)_2$
  - Como el sistema tiene 4 bits, el bit más significativo se descarta, quedando el numeral $(0100)_2 = (4)_{10}$
  - ¿El resultado es representable? **Rta:** Sí

Ejercicios

- Realizar la suma binaria de los números $(5)_{10}$ y $(3)_{10}$ representados en complemento a 2 de 4 bits.
    - ¿El resultado es representable?
    - TAREA!

¿Qué vimos?

- Arquitectura vs. Organización
- ¿Por qué esta materia?
- Funciones principales (almacenamiento, transferencia, procesamiento, control)
- Estructura de una computadora
- Representación de enteros
    - Número, numeral, dígito, base
    - Sistema binario de precisión fija
    - Representaciones: sin signo, complemento a 2, con signo y exceso

Bibliografía

- Null, L. and J. Lobur. The Essentials of Computer Organization and Architecture, Jones and Bartlett Publishers, Feb. 2003