



# DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

## Organización del Computador 1 Introducción e Historia

Dr. Marcelo Risk

16 de agosto 2022

# Índice

Introducción

Historia

# Recursos

- ▶ Libros:
  - ▶ Andrew S. Tanenbaum, Structured Computer Organization, Prentice Hall, 4ta edición (en adelante, actual 6ta).
  - ▶ William Stallings, Computer Organization and Architecture, Prentice Hall, 5ta edición (en adelante, actual 10ma).
  - ▶ Linda Null, Julia Lobur, The Essentials of Computer Organization and Architecture, Jones and Bartlett, 2da edición (en adelante, actual 4ta).
- ▶ Otros recursos:
  - ▶ <https://campus.exactas.uba.ar> → Courses → Dto de Computación → 2022
  - ▶ ¡Internet!

# Introducción

- ▶ ¿Qué es una computadora?

# Introducción

- ▶ ¿Qué es una computadora?
- ▶ Stallings:

*Máquina digital electrónica programable para el tratamiento automático de la información, capaz de recibirla, operar sobre ella mediante procesos determinados y suministrar los resultados de tales operaciones.*

# Introducción

- ▶ ¿Por qué estudiar organización y arquitectura de computadoras?
  - ▶ Diseñar mejores programas de base:
    - ▶ compiladores, sistemas operativos, y drivers
  - ▶ Optimizar programas
  - ▶ Construir computadoras
  - ▶ Evaluar su desempeño
  - ▶ Entender los **compromisos** entre poder de cómputo, espacio y costos

# Introducción

- ▶ ¿Por qué estudiar organización y arquitectura de computadoras?
  - ▶ Diseñar mejores programas de base:
    - ▶ compiladores, sistemas operativos, y drivers
  - ▶ Optimizar programas
  - ▶ Construir computadoras
  - ▶ Evaluar su desempeño
  - ▶ Entender los **compromisos** entre poder de cómputo, espacio y costos

*El hardware es lo que hace a una máquina rápida, el software es lo que hace que una máquina rápida se vuelva lenta.*

Craig Bruce

# Arquitectura vs Organización

***Arquitectura: atributos visibles al programador***

- ▶ Set de instrucciones, bits utilizados para representar los datos, mecanismos de direccionamiento, entrada y salida, etc.

# Arquitectura vs Organización

*Arquitectura: atributos visibles al programador*

- ▶ Set de instrucciones, bits utilizados para representar los datos, mecanismos de direccionamiento, entrada y salida, etc.

*Organización: cómo se implementan*

- ▶ Señales de control, tecnología de la memoria
- ▶ Ejemplos:
  - ▶ ¿Las instrucciones las ejecuta directo el hardware o son interpretadas por microprogramas?
  - ▶ ¿La multiplicación es realizada directamente por un componente o se realizan muchas sumas?

## Arquitectura vs Organización

- ▶ Toda la familia x86 de Intel comparte la misma arquitectura básica
- ▶ Esto asegura la compatibilidad de código
  - ▶ Al menos la de programas antiguos
- ▶ La organización cambia entre diferentes versiones de una misma familia

# Componentes

- ▶ No hay una clara distinción entre asuntos relacionados con la organización y los relevantes con la arquitectura
- ▶ **Principio de equivalencia Hardware-Software:**  
*Cualquier cosa que puede ser hecha por software puede ser hecha en hardware y cualquier cosa que puede ser hecha con hardware puede ser hecha con software*

## Estructura vs. Función

- ▶ La **Estructura** es la forma en que los componentes se relacionan entre sí.

## Estructura vs. Función

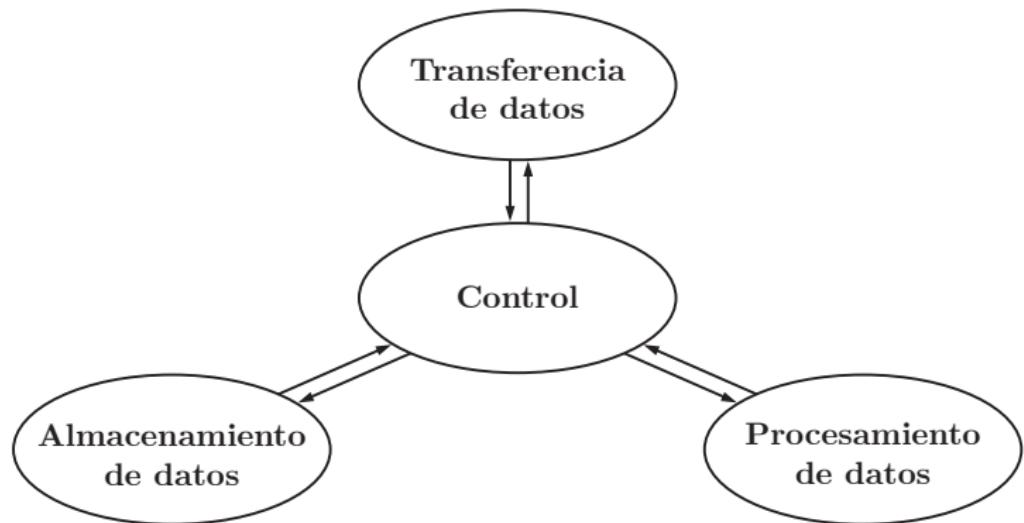
- ▶ La **Estructura** es la forma en que los componentes se relacionan entre sí.
- ▶ La **Función** es la operación que realizan los componentes individuales como parte de una estructura.

# Funciones

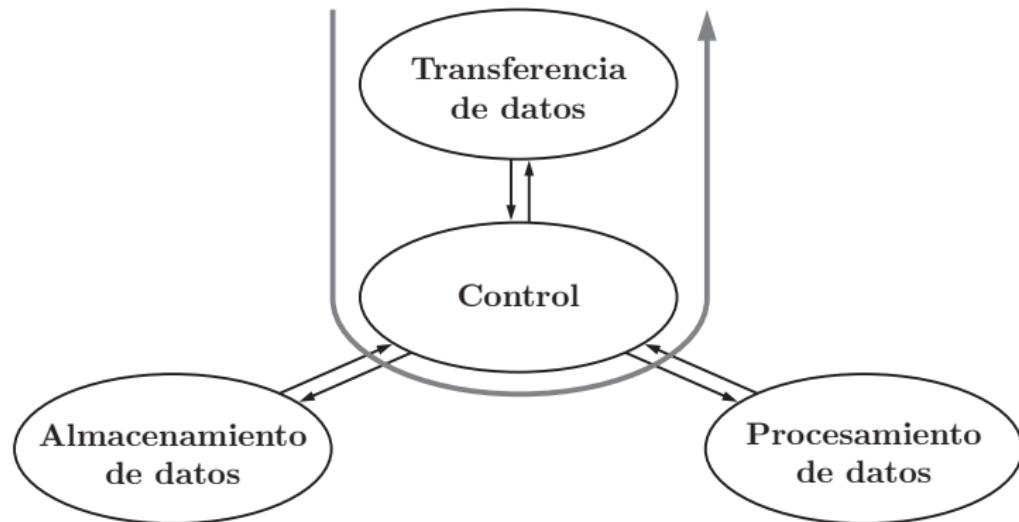
Las funciones básicas de una computadora son:

- ▶ Procesamiento de datos
- ▶ Almacenamiento de datos
- ▶ Transferencia de datos
- ▶ Control

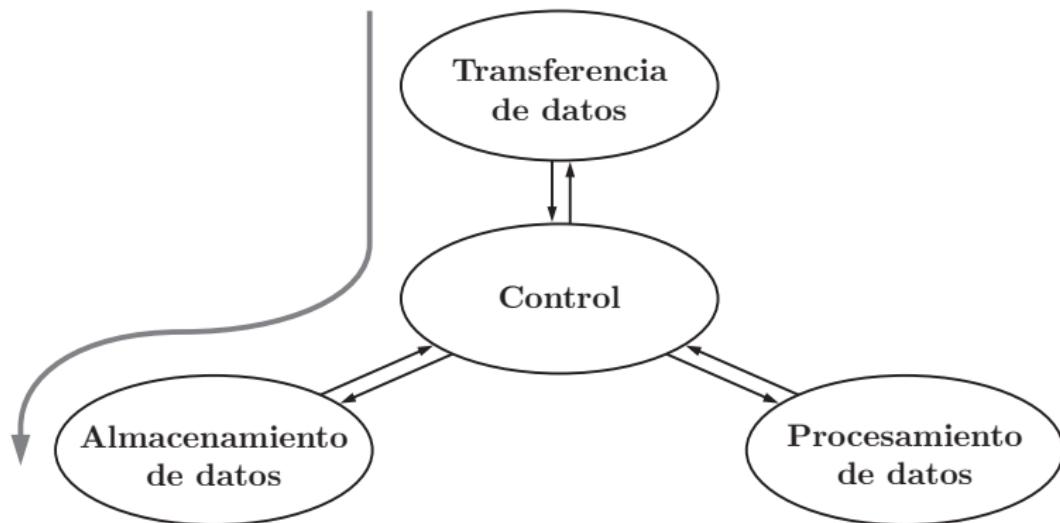
# Visión funcional



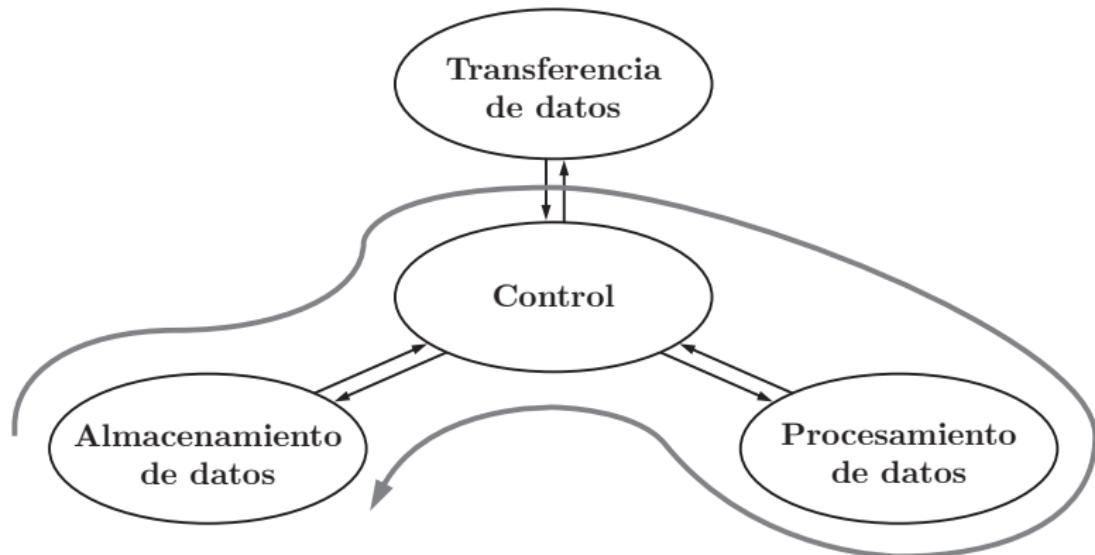
# Operaciones: Transferencia de datos



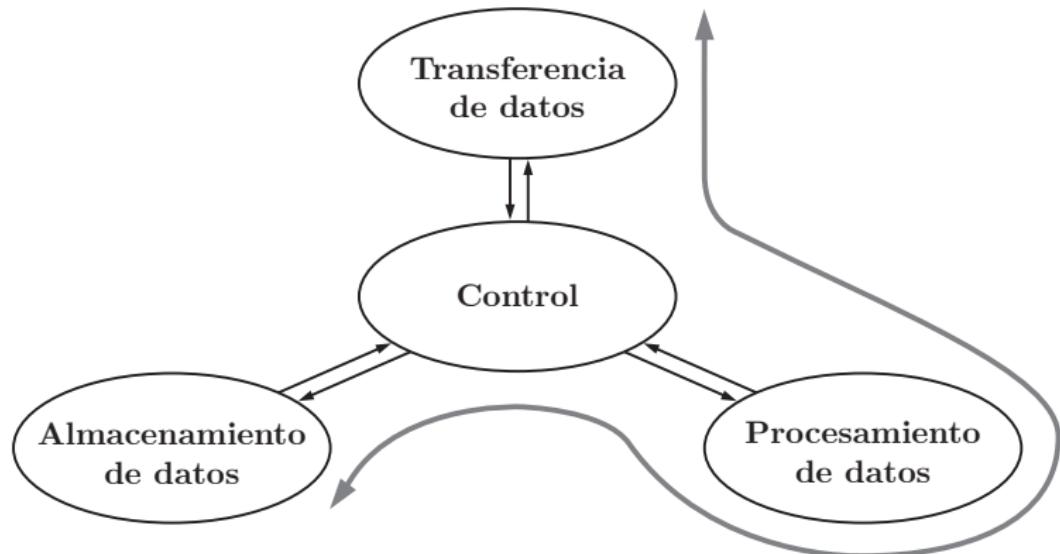
# Operaciones: Almacenamiento



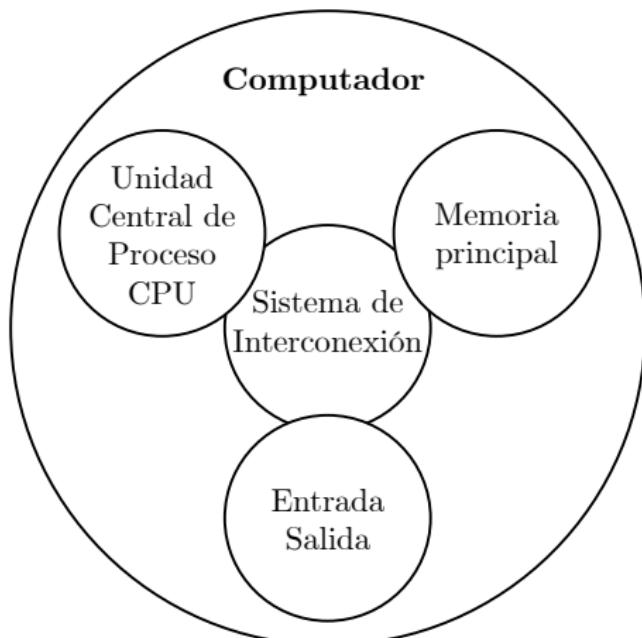
## Operaciones: procesamiento desde/hasta almacenamiento



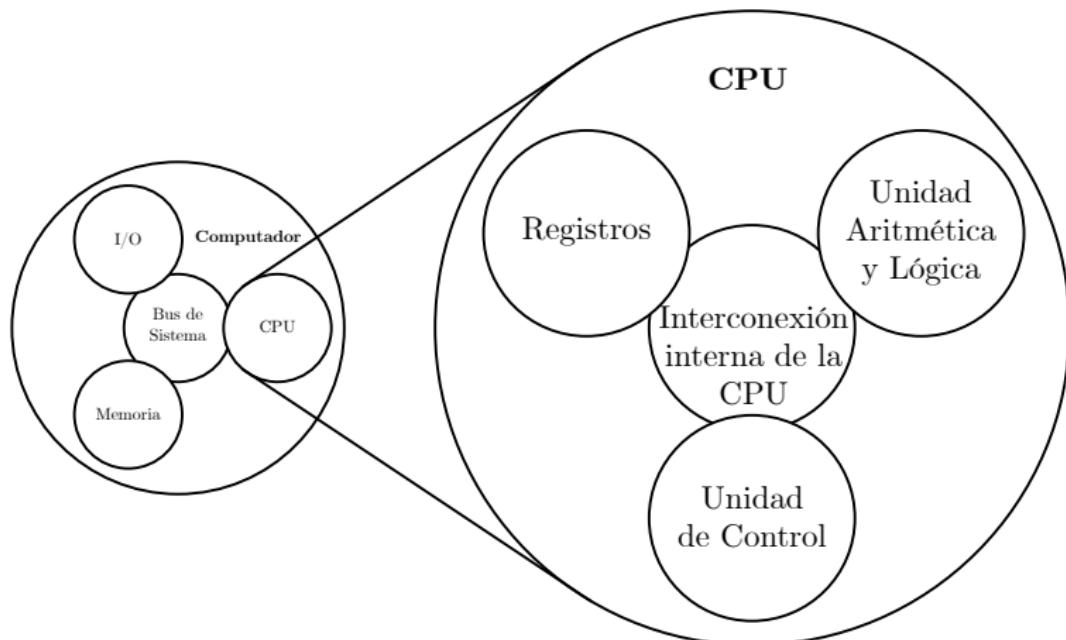
# Operaciones: procesamiento desde almacenamiento a E/S



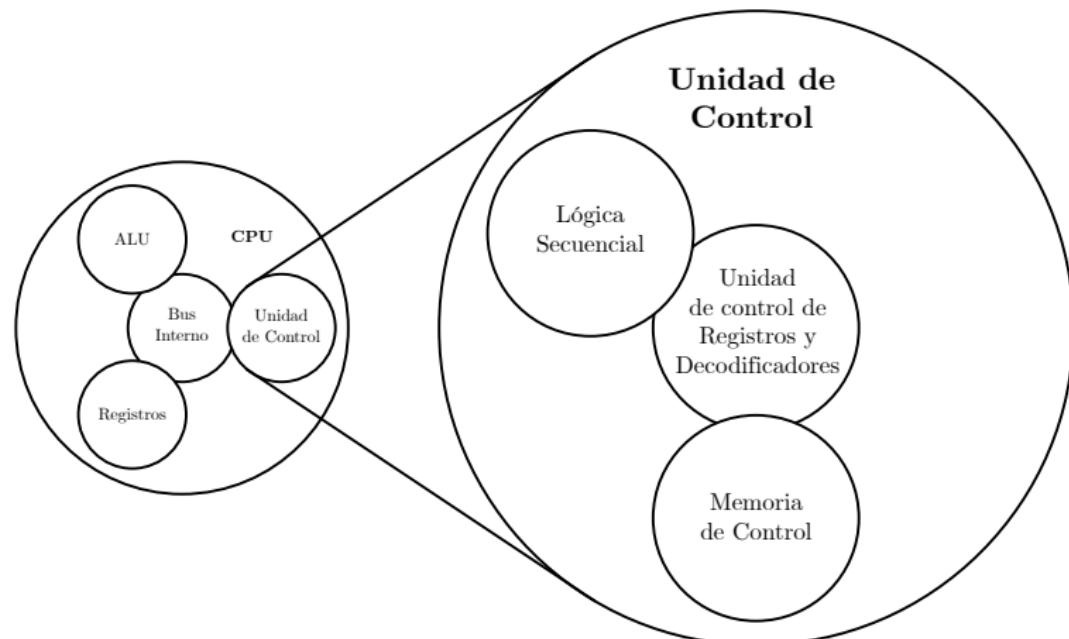
## Estructura: computadora



# Estructura: CPU



## Estructura: UC - Unidad de Control



## Algunas abreviaturas

- ▶ Medidas de **capacidad y velocidad**:
  - ▶ Kilo- (k) = mil =  $10^3$  y  $2^{10}$
  - ▶ Mega- (M) = 1 millón =  $10^6$  y  $2^{20}$
  - ▶ Giga- (G) = 1000 millones =  $10^9$  y  $2^{30}$
  - ▶ Tera- (T) = 1 billón =  $10^{12}$  y  $2^{40}$
  - ▶ Peta- (P) = 1000 millones =  $10^{15}$  y  $2^{50}$

Que una potencia corresponda a medidas de 10 ó 2 depende de la magnitud a medir.

## Algunas abreviaturas

- ▶ **Hertz** = ciclos de reloj por segundo (frecuencia)
  - ▶  $1 \text{ MHz} = 1000000 \text{ Hz}$
  - ▶  $1 \text{ GHz} = 1000 \text{ MHz}$
  - ▶ La velocidad del procesador actual se mide en MHz o GHz.
- ▶ **Byte** = unidad de almacenamiento
  - ▶  $1 \text{ kB} = 2^{10} = 1024 \text{ B}$
  - ▶  $1 \text{ MB} = 2^{20} = 1048576 \text{ B}$
  - ▶ La memoria principal (RAM) se mide en MB
  - ▶ El almacenamiento en disco se mide en GB para sistemas chicos, en TB para sistemas más grandes.
- ▶ **Word** (palabra) = unidad de transferencia: cantidad de bits que pueden moverse simultáneamente dentro de la CPU
  - ▶ 8 bit, 16 bit, 32 bit, 64 bit

# Algunas abreviaturas

- ▶ Medidas de **tiempo** y **espacio**:
  - ▶ Mili- (m) = milésima =  $10^{-3}$
  - ▶ Micro- ( $\mu$ ) = millonésima =  $10^{-6}$
  - ▶ Nano- (n) = mil millonésima =  $10^{-9}$
  - ▶ Pico- (p) = billonésima =  $10^{-12}$
  - ▶ Femto- (f) = mil billonésima =  $10^{-15}$

## Un ejemplo

- ▶ Milisegundo = milésima de segundo
  - ▶ El tiempo de acceso de los HD suele ser de 10 a 20 milisegundos.
- ▶ Nanosegundo = mil millonésima de segundo
  - ▶ El tiempo de acceso a RAM suele ser de 50 a 70 nanosegundos.
- ▶ Micrón (micrómetro) = millonésima de un metro
  - ▶ Los circuitos en los chips de una computadora suelen medirse en micrones.

## Un ejemplo

- ▶ Notar que el tiempo de un ciclo es inversamente proporcional a la frecuencia del reloj.
- ▶ Un bus operando a 133 MHz tiene un tiempo de ciclo de 7.52 ns ( $T = 1/F$ ):

## Un ejemplo

- ▶ Notar que el tiempo de un ciclo es inversamente proporcional a la frecuencia del reloj.
- ▶ Un bus operando a 133 MHz tiene un tiempo de ciclo de 7.52 ns ( $T = 1/F$ ):

133000000 ciclos/s  $\Rightarrow$  7.52 ns/ciclo

## Un ejemplo

- ▶ Las computadoras con mucha memoria principal pueden ejecutar programas más grandes con mayor velocidad que las computadoras que tienen poca memoria.

## Un ejemplo

- ▶ Las computadoras con mucha memoria principal pueden ejecutar programas más grandes con mayor velocidad que las computadoras que tienen poca memoria.
- ▶ RAM es la sigla para nombrar a la **memoria de acceso aleatorio**. Esto significa que los contenidos pueden ser accedidos directamente en base a su ubicación y que el acceso a cualquier posición toma tiempos similares.

## Un ejemplo

- ▶ Las computadoras con mucha memoria principal pueden ejecutar programas más grandes con mayor velocidad que las computadoras que tienen poca memoria.
- ▶ RAM es la sigla para nombrar a la **memoria de acceso aleatorio**. Esto significa que los contenidos pueden ser accedidos directamente en base a su ubicación y que el acceso a cualquier posición toma tiempos similares.
- ▶ La **cache** es un tipo de memoria temporal que puede ser accedida más rápidamente que la RAM.

## Un ejemplo

- ▶ Los puertos seriales envían datos como una serie de pulsos sobre 1 ó 2 líneas de datos, también denominados puertos RS-232.

## Un ejemplo

- ▶ Los puertos seriales envían datos como una serie de pulsos sobre 1 ó 2 líneas de datos, también denominados puertos RS-232.
- ▶ Los puertos paralelos envían datos como un solo pulso sobre varias líneas de datos.

## Un ejemplo

- ▶ Los puertos seriales envían datos como una serie de pulsos sobre 1 ó 2 líneas de datos, también denominados puertos RS-232.
- ▶ Los puertos paralelos envían datos como un solo pulso sobre varias líneas de datos.
- ▶ USB (Universal Serial Bus), es una interface serie inteligente que se “auto-configura” (*plug and play*).

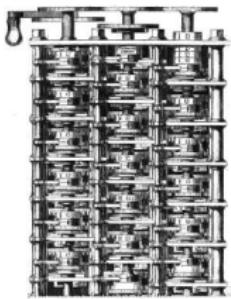
# Historia

Generación	Años	Características
0	hasta 1945	Sistemas mecánicos y electromecánicos
1	1945 a 1954	Tubos al vacío, tableros
2	1955 a 1965	Transistores y sistemas por lotes
3	1965 a 1980	Circuitos integrados
4	desde 1980	VLSI - Computadoras personales y super computadoras

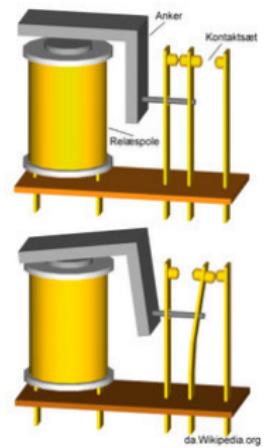
# Primeras computadoras



Ábaco



Calculadora mecánica

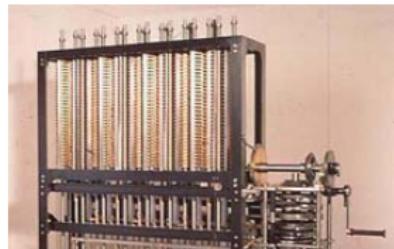


Sistemas basados en relés

# Máquinas diferenciales de Babbage



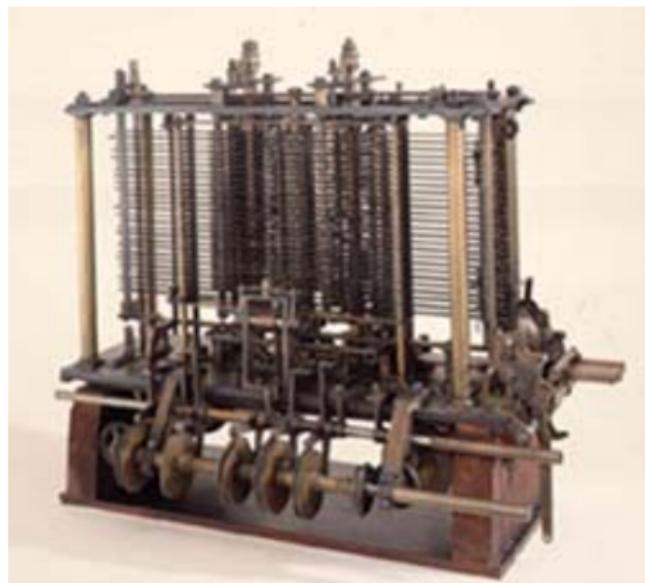
1822: Primera computadora mecánica. Usaba el método de las diferencias finitas para el cálculo de polinomios de 2do grado. Requería aprox. 25000 partes. Fracasó en el intento.



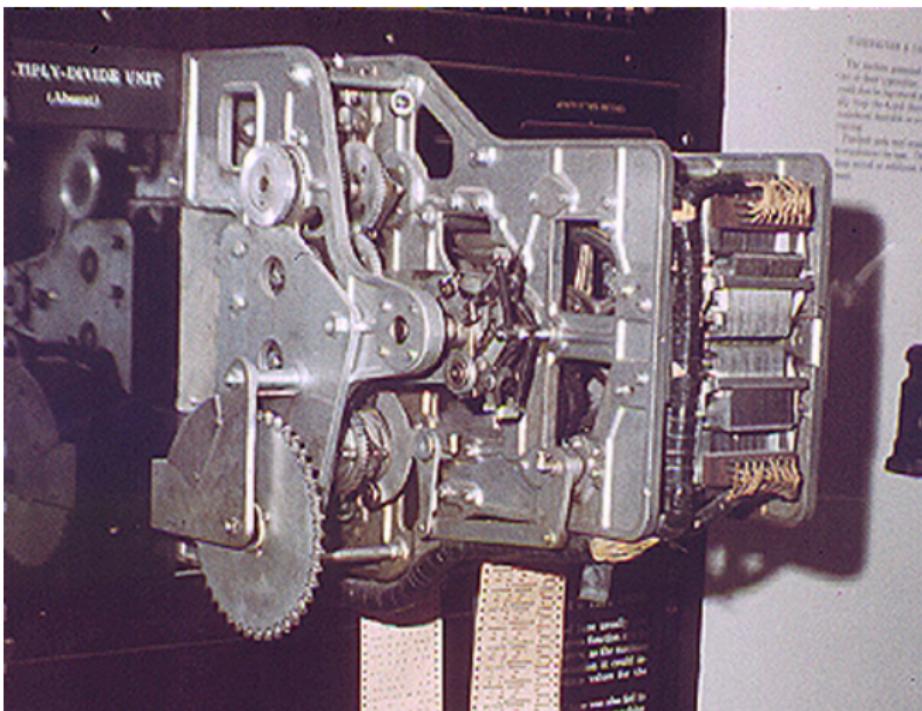
1847: Otra versión más pequeña. No llegó a construirse. Fue reproducida por el Museo de Ciencia en 1985.

## Máquina analítica (1834)

- ▶ Primera Computadora Digital (mecánica)?
- ▶ Calculaba cualquier función algebraica y almacenaba números.
- ▶ Se programaba con tarjetas.
- ▶ Charles Babbage y Ada Lovelace.
- ▶ Fracasó en el intento, pero porque las piezas no podían producirse con la precisión requerida...



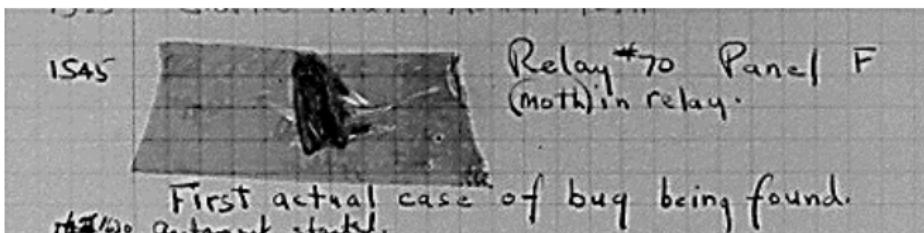
# Harvard Mark I (1939 a 1944)



# Harvard Mark I (1939 a 1944)



## Harvard Mark I (1939 a 1944)



Rear Admiral Grace Murray Hopper:

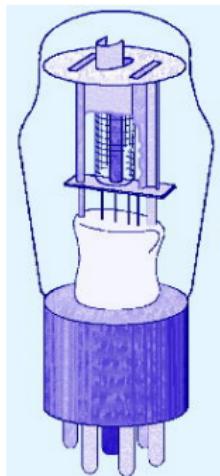
*First actual case of bug being found.*

- ▶ Hopper fue la primer persona en identificar un **bug** en una computadora.
- ▶ Los usuarios venían reportando errores en los resultados de los programas.
- ▶ Cuando abrieron la máquina, encontraron una polilla en el circuito (funcionaba como aislante eléctrico).

# Primera Generación

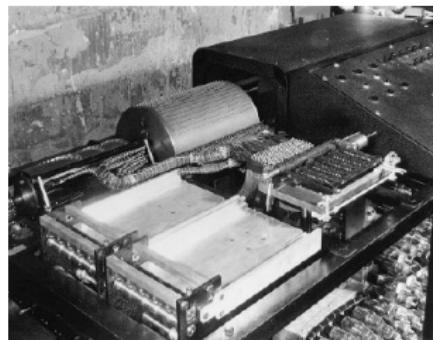
1940 a 1955

- ▶ Utilizan **tubos al vacío o válvulas**
- ▶ Enormes (20000 tubos) y lentas (¿un ciclo? 1 s.)
- ▶ Un solo grupo diseñaba, construía, programaba, operaba y mantenía cada máquina.
- ▶ Toda la programación se hacía en lenguaje máquina (conectando cables en un tablero por ejemplo).
- ▶ No existían los sistemas operativos.
- ▶ En 1950 se introducen las tarjetas perforadas.



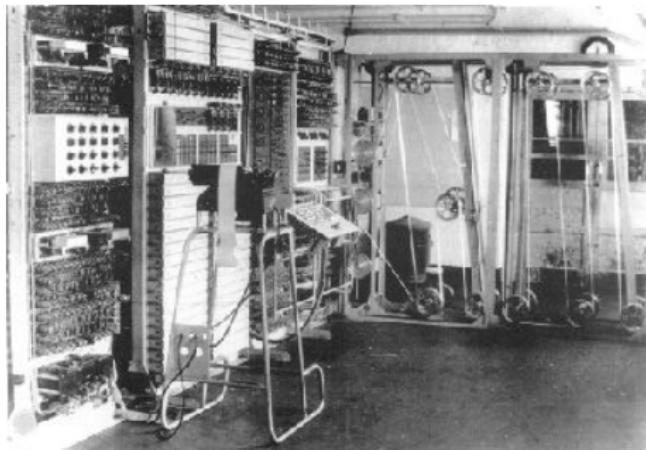
# Atanasoff Berry Computer (1939 a 1942)

- ▶ Primera computadora digital (binaria)
- ▶ **No era de propósito general**
- ▶ Resolvía sistemas de ecuaciones lineales.
- ▶ John Atanasoff y Clifford Berry de la Iowa State University.



# Colossus (1943)

- ▶ Desarrollo Británico.
- ▶ Diseñada para descifrar los mensajes encriptados por los alemanes.
- ▶ Participó Turing.
- ▶ No se conoció hasta los 80 (Top Secret).



# ENIAC (1946)

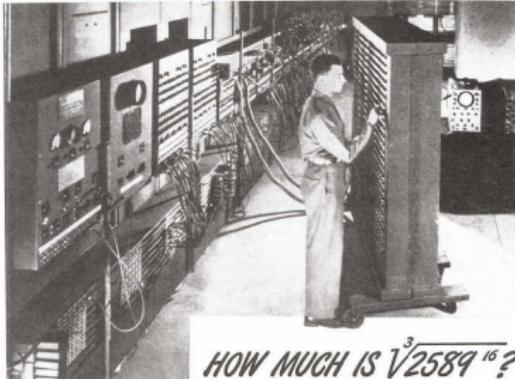
- ▶ Electronic Numerical Integrator and Computer
  - ▶ John Mauchly and J. Presper Eckert (Pennsylvania)
- ▶ Primera computadora de propósito general
- ▶ Se programaba **cableando**
- ▶ Construida entre 1943 y 1946 para calcular trayectoria de artillería
- ▶ Pero se terminó tarde...
- ▶ Von Neumann participó de las últimas etapas del proyecto
- ▶ Se usó hasta 1955



## ENIAC - Detalles

- ▶ Decimal (no binaria)
- ▶ 20 acumuladores de 10 dígitos
- ▶ Programada manualmente usando switches
- ▶ 18000 válvulas
- ▶ ¡30 toneladas!
- ▶ ¡2.40 m ancho × 30 m largo!
- ▶ 140 kW de consumo
- ▶ 5000 adiciones por segundo
- ▶ 500 FLOPS

# ENIAC (1946)



**The Army's ENIAC can give you the answer in a fraction of a second!**

Think that's a stumper? You should see some of the ENIAC's problems! Brain twisters that if put to paper would run off this page and feel beyond . . . addition, subtraction, multiplication, division — square root, cube root, any root. Solved by an incredibly complex system of circuits operating 18,000 electronic tubes and tipping the scales at 30 tons!

The ENIAC is symbolic of many amazing Army devices with a brilliant future for you! The new Regular Army needs men with aptitude for scientific work, and as one of the first trained in the post-war era, you stand to get in on the ground floor of important jobs

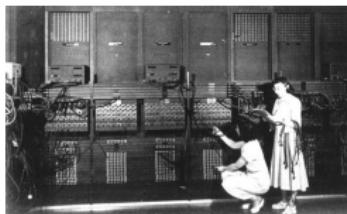
**YOUR REGULAR ARMY SERVES THE NATION  
AND MANKIND IN WAR AND PEACE**

which have never before existed. You'll find that an Army career pays off.

The most attractive fields are filling quickly. Get into the swim while the getting's good! 1½, 2 and 3 year enlistments are open in the Regular Army to ambitious young men 18 to 34 (17 with parents' consent) who are otherwise qualified. If you enlist for 3 years, you may choose your own branch of the service, of those still open. Get full details at your nearest Army Recruiting Station.

**A GOOD JOB FOR YOU**  
**U. S. Army**  
**CHOOSE THIS**  
**FINE PROFESSION NOW!**

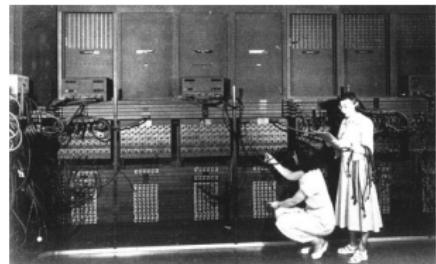
# El modelo de von Neumann



- ▶ Antes: programar era conectar cables...
- ▶ Hacer programas era más una cuestión de ingeniería electrónica.
- ▶ Cada vez que había que calcular algo distinto había que reconectar todo.
- ▶ Mauchly and Eckert (ENIAC) documentaron la idea de almacenar programas como base de la EDVAC.
- ▶ Pero no lo publicaron...

# John Von Neumann

- ▶ 1903 (Hungría) - 1957
- ▶ Doctor en matemática y química
- ▶ Publicó y publicitó la idea de **programa almacenado en memoria**
- ▶ No está claro que se le haya ocurrido a él...

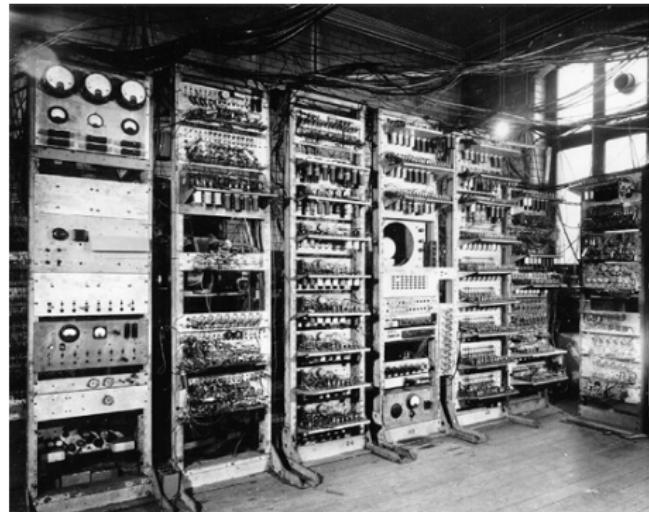


## von Neumann/Turing

- ▶ Los datos y programas se almacenan en una misma **memoria** de lectura-escritura
- ▶ Los contenidos de esta memoria se **direccinan** indicando su posición sin importar su tipo
- ▶ Ejecución en **secuencia** (salvo que se indique lo contrario)

# Manchester Mark I (1948)

- ▶ También llamada **Baby**. Usada para demostrar el concepto de programa almacenado.
- ▶ En 1948 se contrató a Turing para el desarrollo de un lenguaje de programación para la máquina.



# Primer programa almacenado

000  $CI = S$

001  $A = A - S$

010  $A = -S$

011 If  $A < 0$ ,  $CI = CI + 1$

100  $CI = CI + S$

101  $A = A - S$

110  $S = A$

111 HALT

- Obtenía el máximo factor propio de A.
- Tardaba 52 minutos en ejecutarse.

18/7/49

Kilburn Highest Factor Routine (anecdote)

function.	C	25	26	27	line	012345	13456
-24 to C	$-G_1$	-	-	-	1	00011	010
$\leftarrow 26$		$-G_1$			2	01011	110
-26 to C	$b_1$		$-G_1$	$b_n$	3	01011	010
$\leftarrow 27$			$-G_1$	$b_n$	4	11011	110
-23 to C	a	$T_{n+1}$	$-G_n$	$b_n$	5	11101	010
dub. 27	$a = b_1$				6	11011	001
test					7	-	011
add 20 to b6					8	00101	100
dub. 26	$r_n$				9	01011	001
$\leftarrow 25$	$r_n$				10	10011	110
-25 to C					11	10011	010
test					12	-	011
stop	0	0	$-G_N$	$b_N$	13		111
-26 to C	$b_n$	$r_n$	$-G_n$	$b_n$	14	01011	010
dub. 21	$(b_n)^{-1}$		$-G_n$		15	10101	001
$\leftarrow 27$	$b_n^{-1}$			$b_{n+1}$	16	11011	110
-27 to C	$(b_n)^{-1}$			$b_{n+1}$	17	11011	010
$\leftarrow 26$		$-G_n$	$b_n$	$b_{n+1}$	18	01011	110
22 to b6		$r_n$	$-G_n$	$b_n$	19	01101	000

init.	final
20	$r_n(20)$
21	$-G_N$
22	$b_N$

or 10100

# UNIVAC (1949)

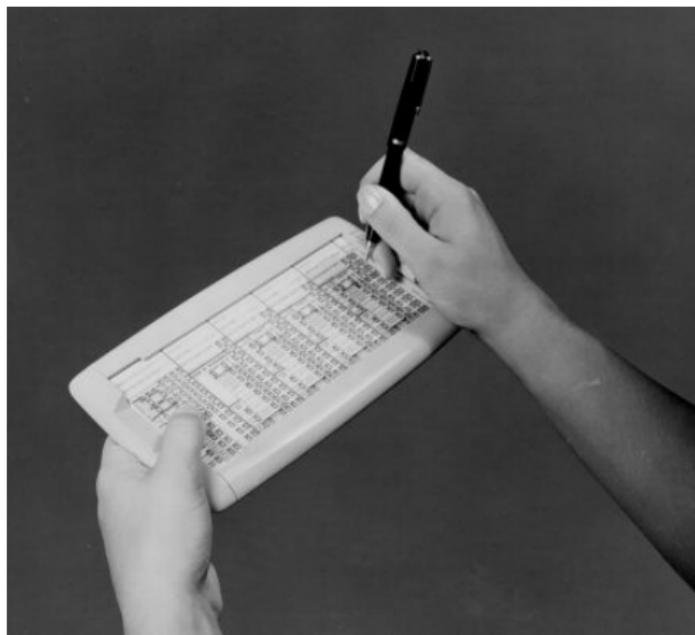
- ▶ Primera computadora comercial
- ▶ Eckert-Mauchly Computer Corporation
- ▶ (Universal Automatic Computer)
  - ▶ Incorpora el uso de cintas magnéticas
  - ▶ Cálculos para el censo de USA
  - ▶ Fin de los 50 - UNIVAC II (más rápida y con más memoria)



# UNIVAC



## Tarjetas perforadas



# JOHNNIAC (1954)



Clon de las IAS. Funcionaba con tarjetas.

# IBM 650 (1955)

- ▶ Primera computadora producida en masa
- ▶ Fuera de circulación en 1969



# IBM 704 (1955)



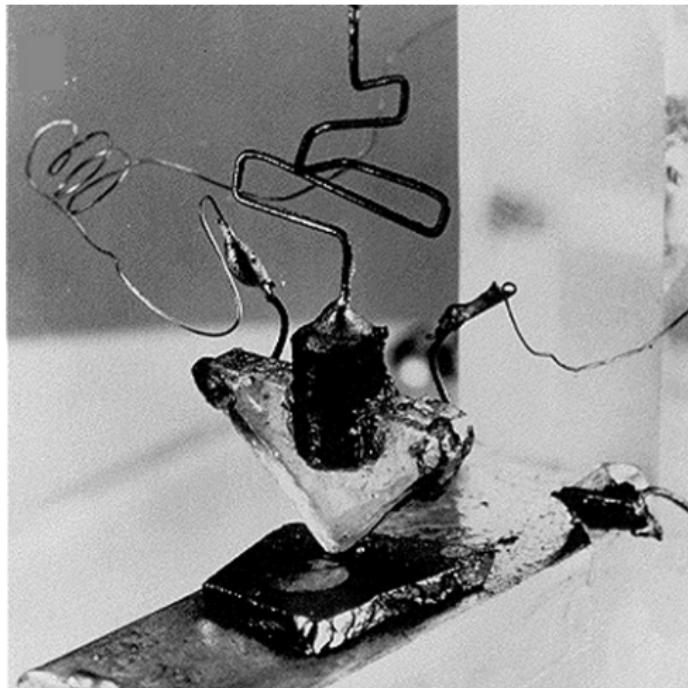
- ▶ Primera máquina comercial con hardware de punto flotante.
- ▶ 5 KFLOPS.

## Segunda Generación

### 1955 a 1966

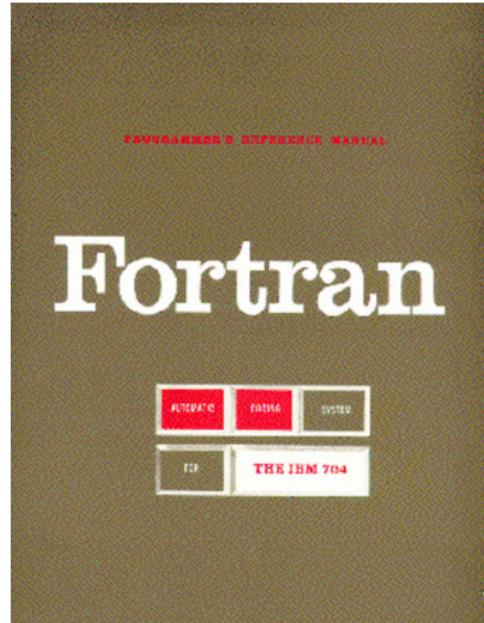
- ▶ Se introducen los **transistores**.
  - ▶ Más baratos.
  - ▶ Más chicos.
  - ▶ Menos disipación de calor.
  - ▶ Silicio (arena).
- ▶ **Mainframes** en salas acondicionadas.
  - ▶ Se escribían los programas en papel, luego se perforaban las tarjetas.
  - ▶ Los operadores toman las tarjetas del programa y colocan también los del compilador.
  - ▶ Se crea el proceso por lotes que agrupa trabajos.
- ▶ Nace la **microprogramación**.

# Transistor (1947)



# FORTRAN (1957)

- ▶ Primer compilador FORTRAN para IBM 704
- ▶ (Formula Translator)



## IBM 1401 (1959)

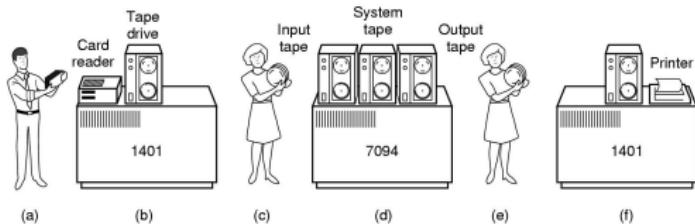
- ▶ 4 kB de memoria expandible a 16 kB.
- ▶ Buena para leer tarjetas, copiar cintas e imprimir resultados.
- ▶ Mala para cálculos numéricos.
- ▶ Se utilizaba con fines comerciales (bancos, etc.)

# IBM 7094 (1962)

- ▶ Buena para hacer cómputos.
- ▶ Se utilizaba con fines científicos.



# IBM 7094 (1962)

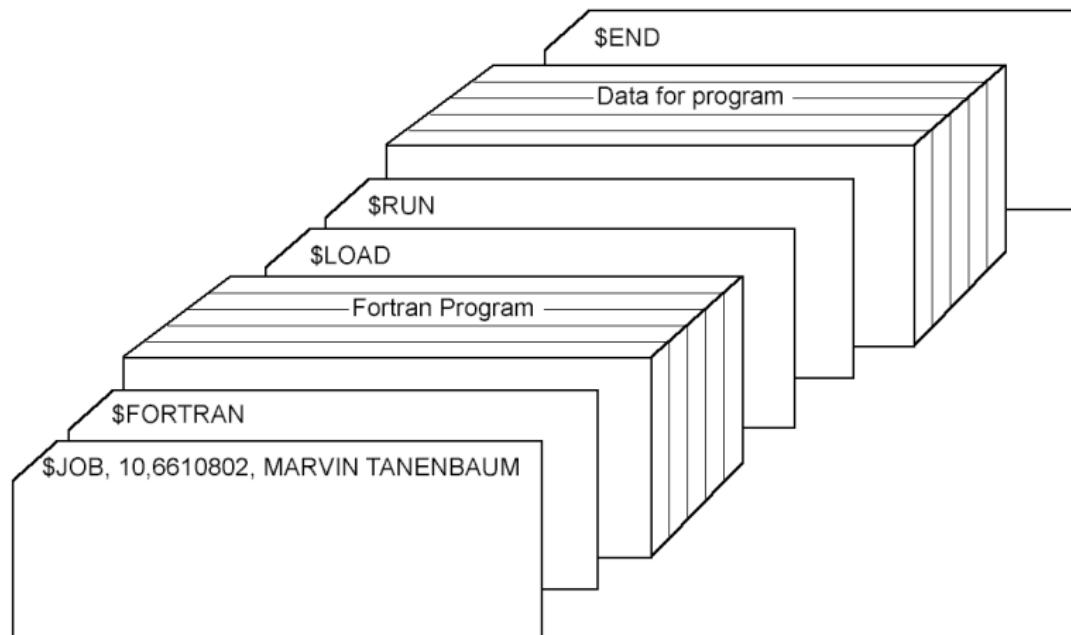


IBM 1401 - IBM 7094:

- a) Los programadores llevan tarjetas
- b) La 1401 lee un lote de tarjetas y los graba en la cinta
- c) Un operador lleva la cinta a la 7094
- d) La 7094 realiza los cómputos
- e) Un operador lleva la cinta a una 1401
- f) La 1401 imprime las salidas

# Trabajo en FORTRAN

- ▶ Fortran Monitor System, comienzo de los sistemas operativos.

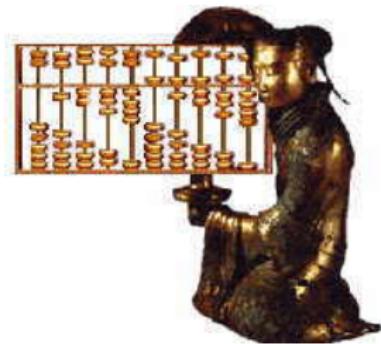


# DEC PDP-1 (1961)

- ▶ 4K de palabras de 18 bits.
- ▶ U\$S 120000.
- ▶ 5% del precio de la IBM 7094.



# Invención del Mouse (1964)

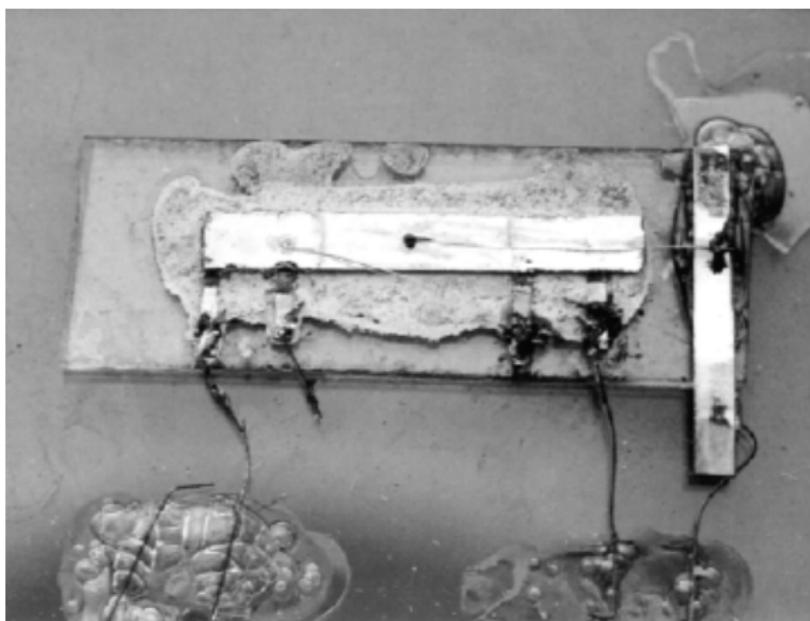


## Tercera Generación

### 1965 a 1980

- ▶ Se introducen los **circuitos integrados**.
  - ▶ Bajan los costos.
  - ▶ Sube su desempeño.
- ▶ Se introduce la **multiprogramación**.
  - ▶ Tiempo compartido entre usuarios.
- ▶ Se introducen los **discos duros**.

# Circuitos integrados



# IBM 360 (1964)

- ▶ Multiprogramación
- ▶ Terminales bobas
- ▶ Software compatible con IBM 7094, 1401 entre otros
- ▶ Aparece el byte de 8bits



# DEC PDP-8 (1964)

- ▶ Primer minicomputador
- ▶ No necesita una habitación con aire acondicionado
- ▶ Lo bastante pequeño para colocarlo en una mesa de laboratorio
- ▶ US\$ 16000



## Inicio de una era: Fairchild Semiconductors

*Everytime we came up with a new idea, we spawned two or three companies trying to exploit it.*

Gordon Moore, co-fundador de Fairchild Semiconductor.

- ▶ Fundada en 1957 por ocho empleados descontentos del Shockley Semiconductor Laboratory (dirigido por el co-inventor del transistor) en Mountain View, California.

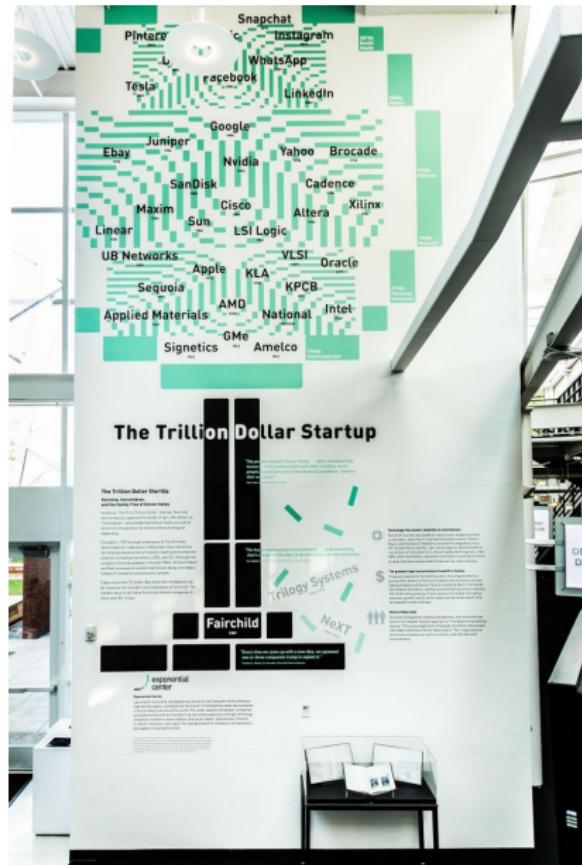
# Inicio de una era: Fairchild Semiconductors

*Everytime we came up with a new idea, we spawned two or three companies trying to exploit it.*

Gordon Moore, co-fundador de Fairchild Semiconductor.

- ▶ Fundada en 1957 por ocho empleados descontentos del Shockley Semiconductor Laboratory (dirigido por el co-inventor del transistor) en Mountain View, California.
- ▶ Un mes luego de fundada, USSR puso el **Sputnik** en órbita: la respuesta de USA fue una gran demanda de semiconductores.
- ▶ De un día para el otro, Fairchild se convirtió en un éxito técnico y financiero volviendo a esta industria la de mayor crecimiento en la región. Surge el término **Silicon Valley**.

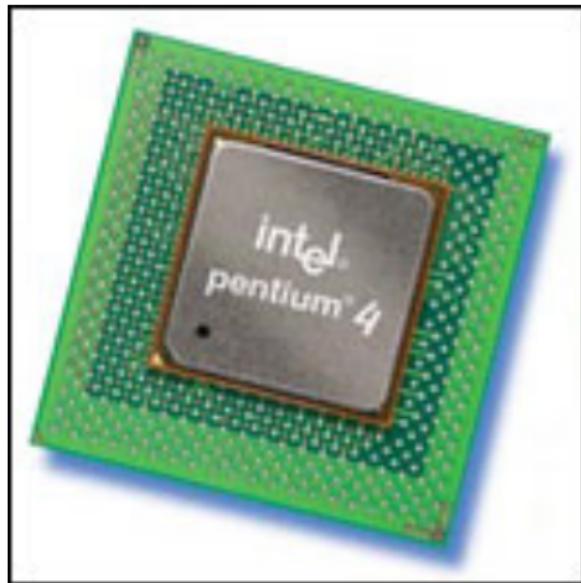
# Fairchildren: Lobby del Computer History Museum



- ▶ A partir de Fairchild surgió la cultura del *spin-off* que hoy en día sigue marcando la dinámica de Silicon Valley.
- ▶ Muchas empresas conocidas hoy en día fueron fundadas por empleados de Fairchild.
- ▶ También existió una gran influencia de Standford University y de Xerox PARC.

## Fundación de Intel (1968)

- ▶ Andy Grove, Robert Noyce y Gordon Moore

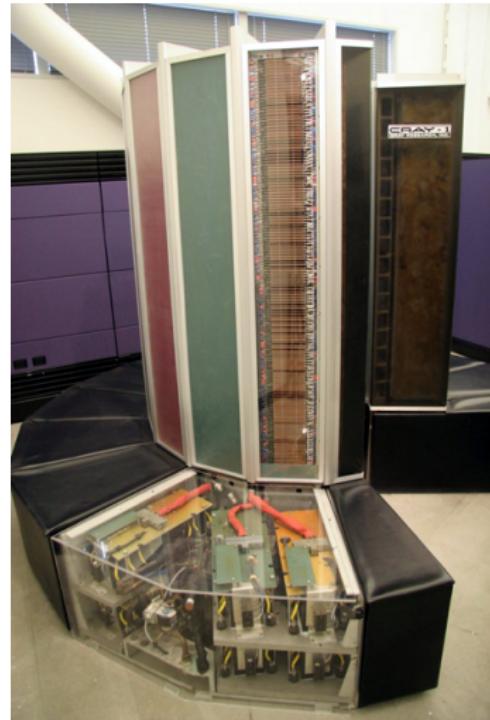


## Lenguaje C (1972)

- ▶ Dennis Ritchie desarrolla el lenguaje C en Laboratorios Bell.
- ▶ Por muchos años, el standard para el lenguaje C era la versión que venía con UNIX.
- ▶ Recién en 1983 se comienza a trabajar en un standard (ANSI C), que salió en 1990 (C89).
- ▶ C89 es la versión más adoptada y base de lo que después fue C++.

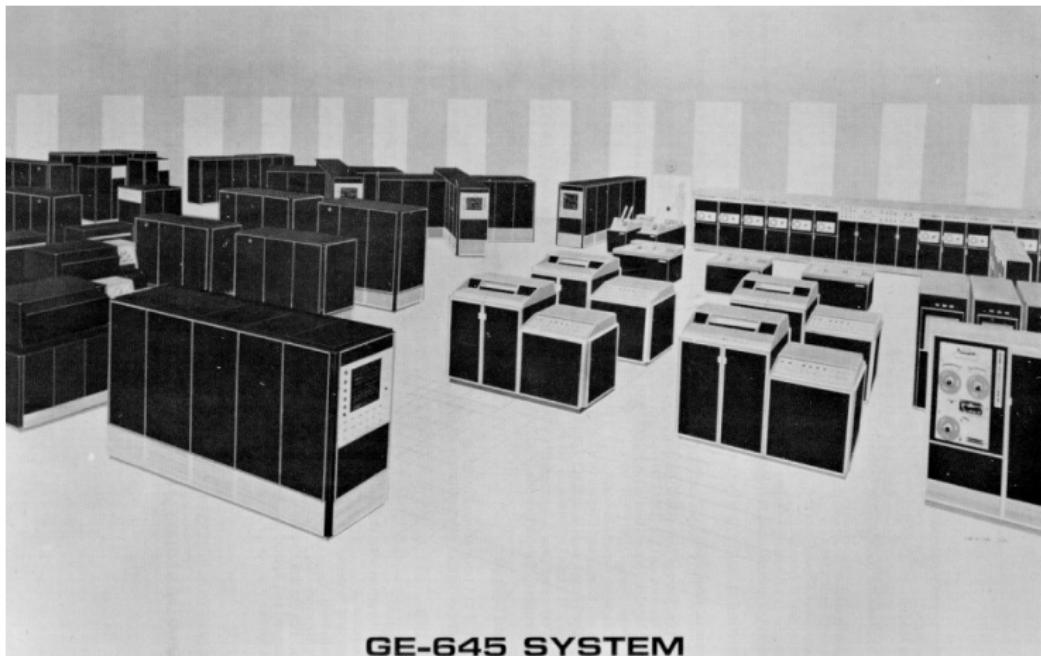
# Cray 1 (1976)

- ▶ Seymour Cray
- ▶ Primer instalación en Los Alamos National Laboratory
- ▶ Costo: US\$ 7,9 millones en 1977 (equivalente a 33,3 millones in 2019)
- ▶ Primera supercomputadora
- ▶ **Procesamiento vectorial**
  - ▶ 12 unidades procesando en paralelo
- ▶ Aprox. 120 MFlops



# MULTICS (1976)

- ▶ Impulso en el desarrollo de Sistema Operativo **timesharing**



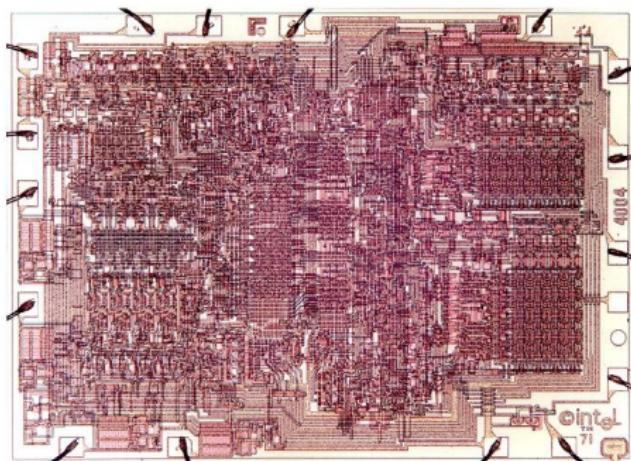
# Primer microprocesador en un chip Intel

## Intel 4004 (1971)

- ▶ CPU de 4 bits
- ▶ 2300 transistores
- ▶ Usado para calculadoras
- ▶ Dispositivos de control

## Intel 8080 (1974)

- ▶ 8 bits datos
- ▶ 16 bits direcciones



# ALTAIR 8800 (1975)

- ▶ Primera computadora personal
- ▶ Tenía un Intel 8080



# Apple 1 (1976)

Steve Jobs y Steve Wosniak



## Apple 2 (1978)

- ▶ Se podía aumentar la RAM
- ▶ Tenía 8 slots de expansión



# Microsoft (1978)

- ▶ 1975: Basic para la Altair
- ▶ 1981: acuerdan con IBM el desarrollo de DOS

Would you have invested?



Microsoft Corporation, 1978

# Cuarta Generación

## Desde 1980

- ▶ Usan VLSI (large scale integration)
  - ▶ Más de 100000 componentes por chip
  - ▶ Facilita la creación de microprocesadores
- ▶ Intel 8080 (8 bits)
  - ▶ IBM PC (1981) con DOS.
  - ▶ Intel 80286, 80386 y 80486.
- ▶ Aparecen las terminales gráficas (GUI)
  - ▶ Macintosh
  - ▶ Microsoft **adopta** GUI y desarrolla Windows (sobre DOS)
- ▶ Aparecen la filosofía **RISC**

# IBM PC (1981)

- ▶ Usa el Intel 8088
- ▶ Sistema DOS (Microsoft)
- ▶ 1983: XT, con disco rígido



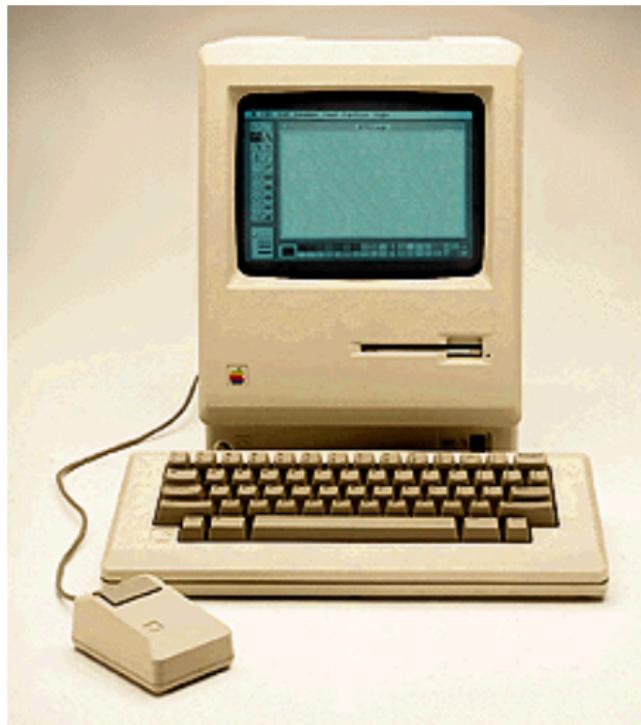
# Commodore 64 (1982)



## Sony introduce el CD (1984)



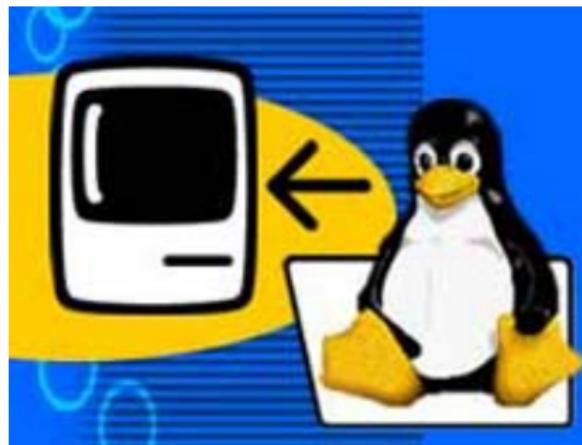
# Macintosh (1984)



## Linux (1991)

*Estoy construyendo un sistema operativo gratuito (no es más que un hobby, no será una cosa grande y profesional como GNU) para clones AT (con un 386 o 486).*

Linus Torvalds, Helsinki, Oct. 1991



# Pentium (1993)

- ▶ Incorpora ideas de máquinas RISC
- ▶ 1994: Pentium Bug
  - ▶  $5505001 / 294911 = 18.66600093$   
(Pentium)
  - ▶  $5505001 / 294911 = 18.666651973$   
(Powerpc)
  - ▶  $X = 5505001, Y = 294911$
  - ▶  $Z = (X/Y)*Y - X$  (debería dar 0)
  - ▶ Pentium con Bug: -256.00000



## Resumen

- ▶ Tubos de vacío: 1946 a 1957
- ▶ Transistores: 1958 a 1964
- ▶ Small scale integration (SSI): hasta 1965
  - ▶ Hasta 100 dispositivos en un chip
- ▶ Medium scale integration (MSI): hasta 1971
  - ▶ 100 a 3000 de dispositivos en un chip
- ▶ Large scale integration (LSI): 1971 a 1977
  - ▶ 3000 a 100000 de dispositivos en un chip
- ▶ Very large scale integration (VSLI): 1978 a 1991
  - ▶ 100000 a 100000000 de dispositivos en un chip
- ▶ Ultra large scale integration (ULSI): 1991 a la actualidad
  - ▶ Más de 100000000 de dispositivos en un chip

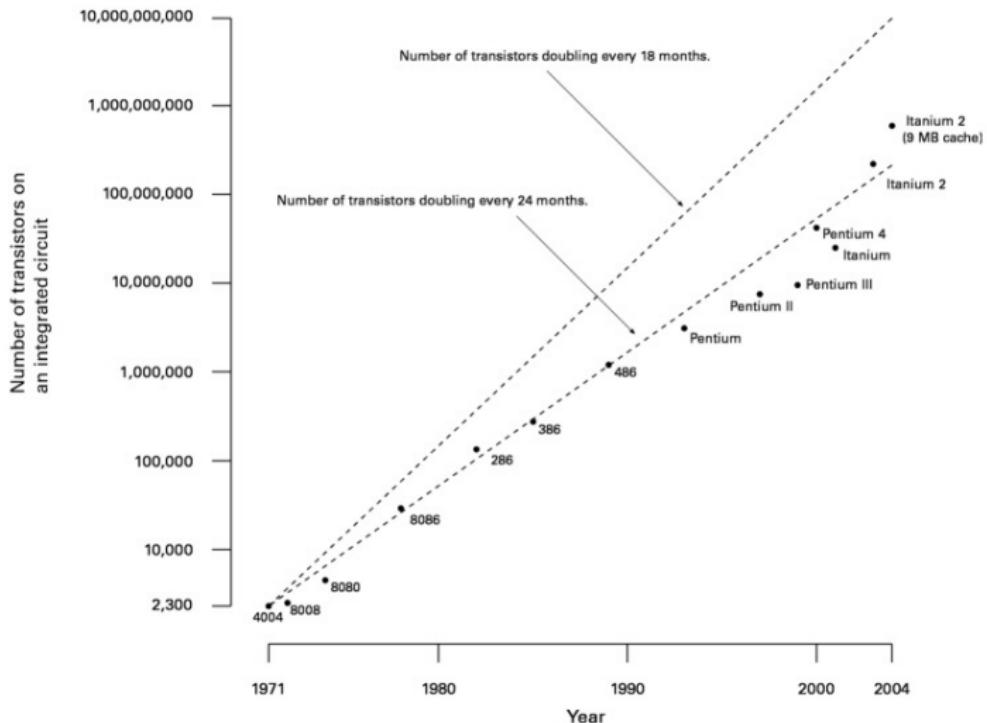
# Desarrollo

- ▶ Ley de Moore (1965)
  - ▶ Gordon Moore, fundador de Intel
  - ▶ **La densidad de transistores en un circuito integrado se duplicará cada año**
- ▶ Versión contemporánea:
  - ▶ **La densidad de chips de silicio se duplica cada 18 meses**

Pero esta ley no puede durar por siempre...

# Desarrollo

Moore's Law



## Desarrollo

- ▶ Ley de Rock
  - ▶ Arthur Rock, ejecutivo de finanzas de Intel
  - ▶ **El costo de equipamiento necesario para construir semiconductores se duplicará cada cuatro años**
- ▶ En 1968, construir una planta para chips costaba alrededor de US\$ 12000

Más o menos lo que salía una casa linda en la periferia de la ciudad.

Un muy buen sueldo anual de un ejecutivo.

## Desarrollo histórico

- ▶ Ley de Rock
  - ▶ En 2003, una fábrica de chips costaba aprox. US\$ 2500 millones.
  - ▶ En 1968, construir una planta para chips costaba alrededor de US\$ 12000.

Esto es más que el producto bruto de algunos países chicos como Belize y la República de Sierra Leona.

# Intel (1)

- ▶ 8080
  - ▶ Primer microprocesador de uso general.
  - ▶ 8 bits.
  - ▶ Usado en la primer PC - Altair.
- ▶ 8086
  - ▶ Mucho más poderoso.
  - ▶ 16 bits.
  - ▶ Cache de instrucciones, prefetch de instrucciones.
  - ▶ 088 (bus externo de 8 bits) - Primera PC de IBM.
- ▶ 80286
  - ▶ Direcciona 16 MB de memoria.

## Intel (2)

- ▶ 80386
  - ▶ 32 bits.
  - ▶ Soporte para multitarea.
- ▶ 80486
  - ▶ Cache y pipeline de instrucciones.
  - ▶ Co-procesador matemático.
- ▶ Pentium
  - ▶ Superscalar.
  - ▶ Varias instrucciones ejecutando en paralelo.
- ▶ Pentium Pro
  - ▶ Predicción de saltos.
  - ▶ Ejecución especulativa.
- ▶ Pentium II
  - ▶ MMX, procesamiento de gráficos, videos y audio.

## Intel (3)

- ▶ Pentium III
  - ▶ Más instrucciones de punto flotante para gráficos.
- ▶ Pentium 4
  - ▶ Mejoras en punto flotante y multimedia.
  - ▶ Hiperthreading.
- ▶ Itanium
  - ▶ 64 bits.
- ▶ Itanium 2
  - ▶ Mejoras en hardware para aumentar la velocidad.
- ▶ ¡Mirar páginas de Intel!

## Algunos Links

- ▶ <http://www.computerhistory.org/>
- ▶ <http://www.intel.com/>
  - ▶ Intel Museum.
- ▶ <http://www.ibm.com/ibm/history>
- ▶ <http://www.dec.com>
- ▶ [www.cbi.umn.edu](http://www.cbi.umn.edu): Charles Babbage Institute