

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación



IIC1005 – Computación: Ciencia y Tecnología del Mundo Digital

Arquitectura de Computadores

Hans Löbel

De qué trata esta clase

- Empezaremos con algunas definiciones y conceptos básicos sobre los computadores.
- A continuación seguiremos con un poco de historia de los computadores modernos.
- Luego veremos el interior de un computador y como funciona.
- Finalmente discutiremos sobre el futuro y por qué es y será esencial saber cómo funciona un computador.

¿DE QUÉ ME SIRVE APRENDER ESTO?

Tengo que tomar decisiones sobre la arquitectura de TI de mi empresa, y me ofrecen muchas opciones.

¿DE QUÉ ME SIRVE APRENDER ESTO?

Tengo que tomar decisiones sobre la arquitectura de TI de mi empresa, y me ofrecen muchas opciones.

Mi código falla o corre muy lento, a pesar de que lo he revisado y corregido muchas veces.

¿DE QUÉ ME SIRVE APRENDER ESTO?

Tengo que tomar decisiones sobre la arquitectura de TI de mi empresa, y me ofrecen muchas opciones.

Mi código falla o corre muy lento, a pesar de que lo he revisado y corregido muchas veces.

Si nuestra especialidad se centra en los computadores, es fundamental entonces comprender como funcionan.

¿QUÉ ES UN COMPUTADOR?



¿QUÉ ES UN COMPUTADOR?



¿QUÉ ES UN COMPUTADOR?



¿QUÉ ES UN COMPUTADOR?



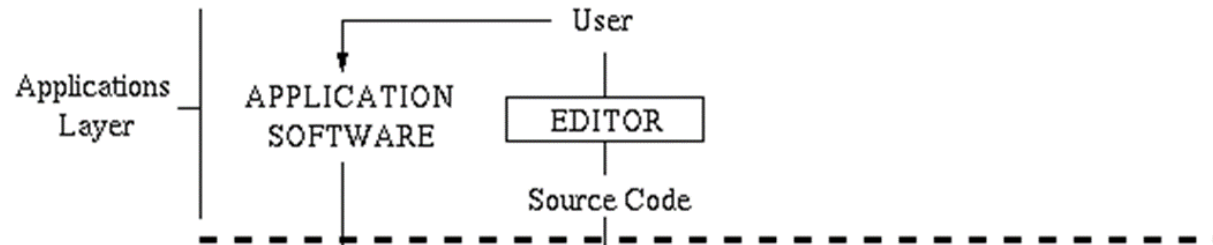
¿QUÉ ES UN COMPUTADOR?



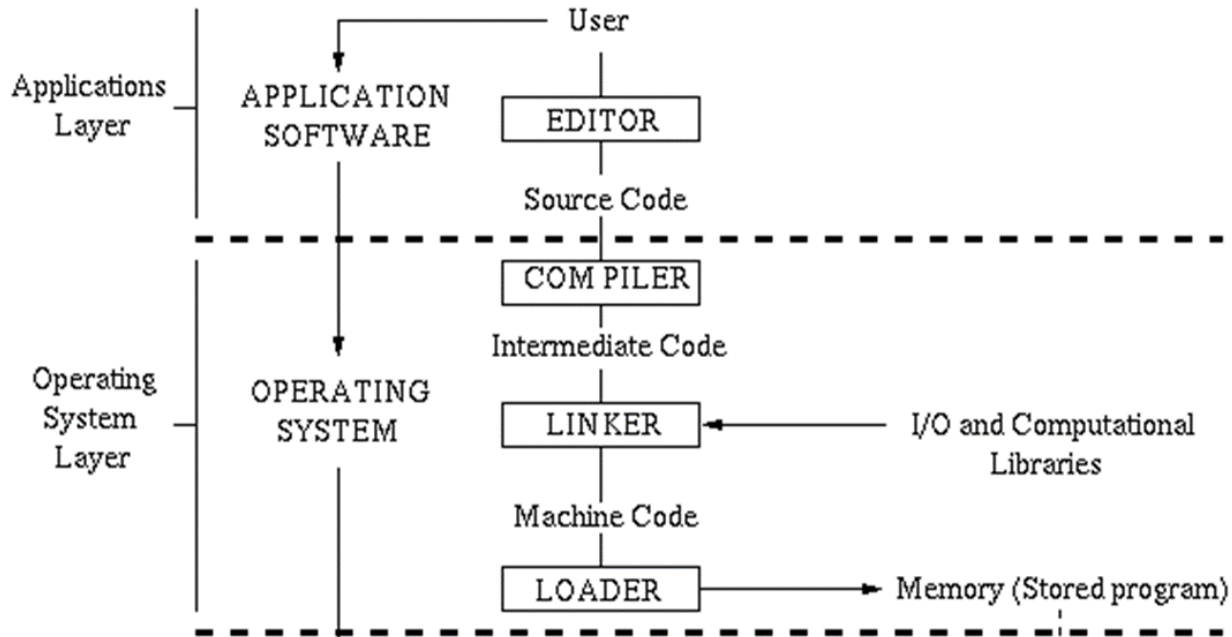
Desde el punto de vista de la arquitectura de computadores, la respuesta es clara

- Lo más importante: es una **máquina programable**.
- Permite escribir programas (secuencias de instrucciones) para ella y ejecutarlos.
- Cómo se construye una máquina de este tipo es la pregunta central del curso de arquitectura de computadores.
- Sin embargo, la arquitectura de computadores cubre bastante más...

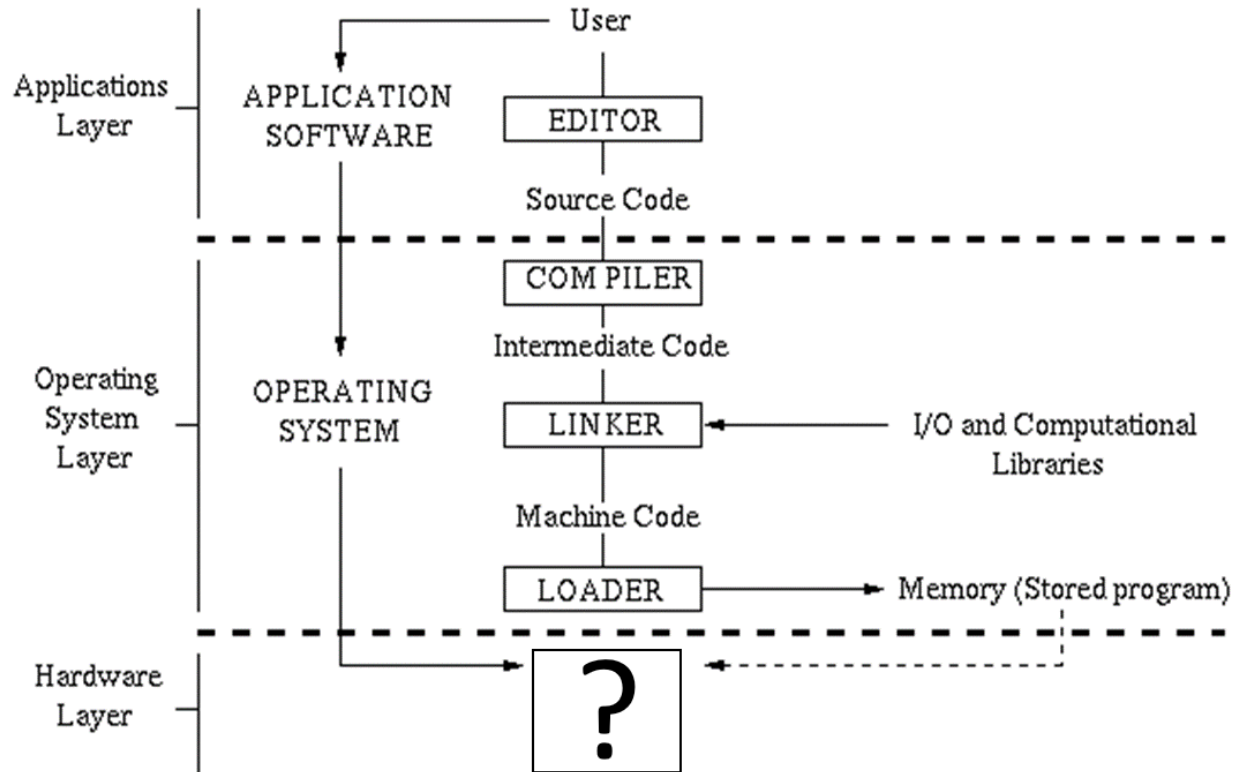
¿Cómo se estructura un sistema computacional (software+hardware)?



¿Cómo se estructura un sistema computacional (software+hardware)?



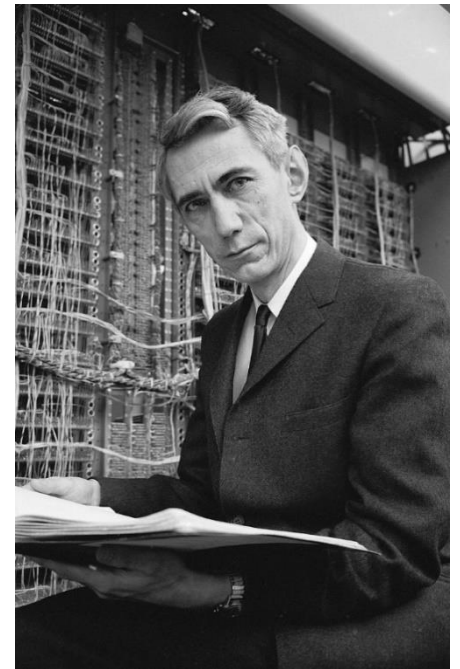
¿Cómo se estructura un sistema computacional (software+hardware)?



Veamos un poco de historia

Durante los 30s, las máquinas analógicas dominaban la computación

- **Claude Shannon** (padre de la teoría de la información) estaba convencido que esta no era la mejor solución.
- Planteó una estrategia de tres pasos para diseñar un computador más eficiente, que finalmente resultaría siendo el computador digital.
- Desarrolló estas ideas en su tesis de magister, a los 19 años.



Una fórmula de lógica booleana tiene 2 componentes principales

- Propositiones lógicas o variables:
verdadero o falso

A = la luz está prendida

B = está nublado

- Conectivos lógicos

Permiten unir las variables, análogo a operadores (+,-,...).

Se definen usando tablas de verdad.

| A | B | A and B |
|---|---|---------|
| F | F | F |
| F | V | F |
| V | F | F |
| V | V | V |

| A | B | A or B |
|---|---|--------|
| F | F | F |
| F | V | V |
| V | F | V |
| V | V | V |

| A | not(A) |
|---|--------|
| F | V |
| V | F |

Álgebra booleana permite definir sentencias lógicas complejas

- Bastan **AND, OR** y **NOT** para representar cualquier tabla de verdad de conectivos binarios.
- Para definir sentencias complejas, basta con conectar múltiples variables y operadores.
- Por ejemplo:

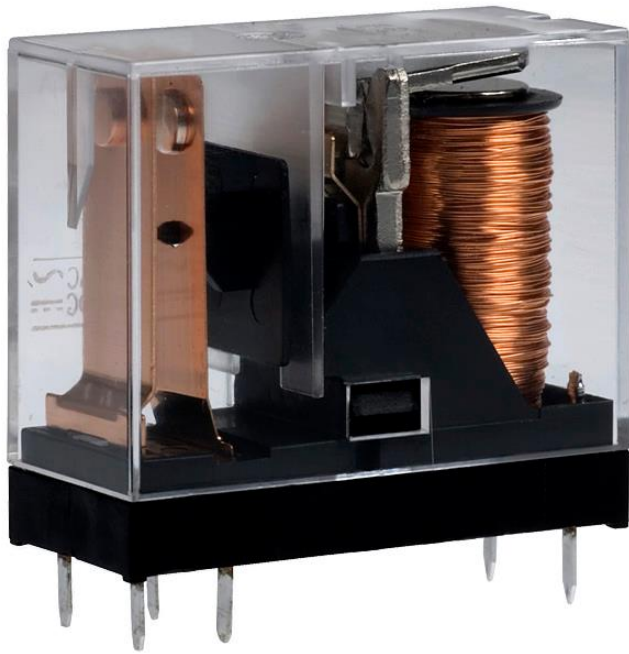
$$A \oplus B = (\neg A \wedge B) \vee (A \wedge \neg B)$$

Lógica booleana permite representar operaciones aritméticas

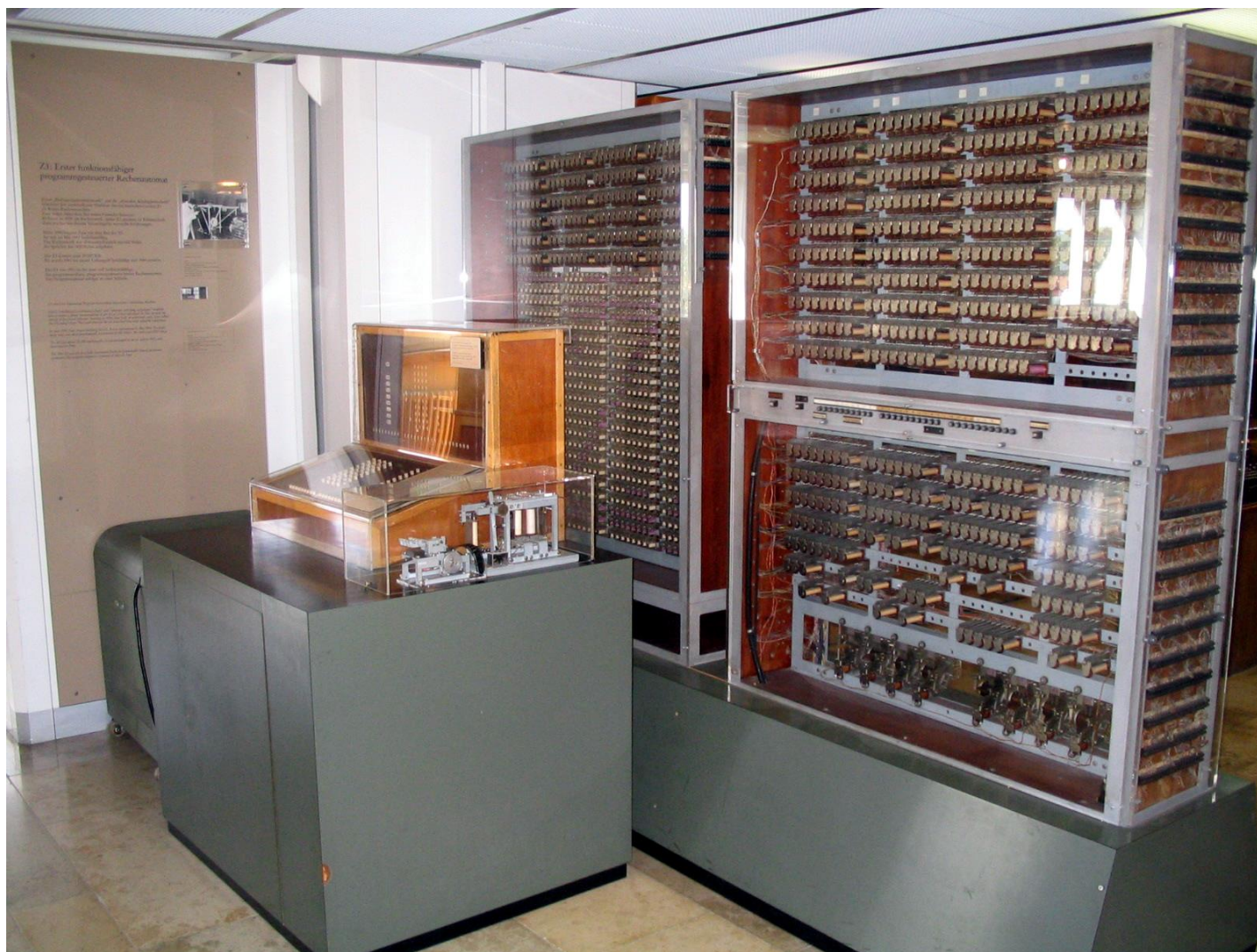
| A | B | C | S |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

$$S = A \oplus B$$

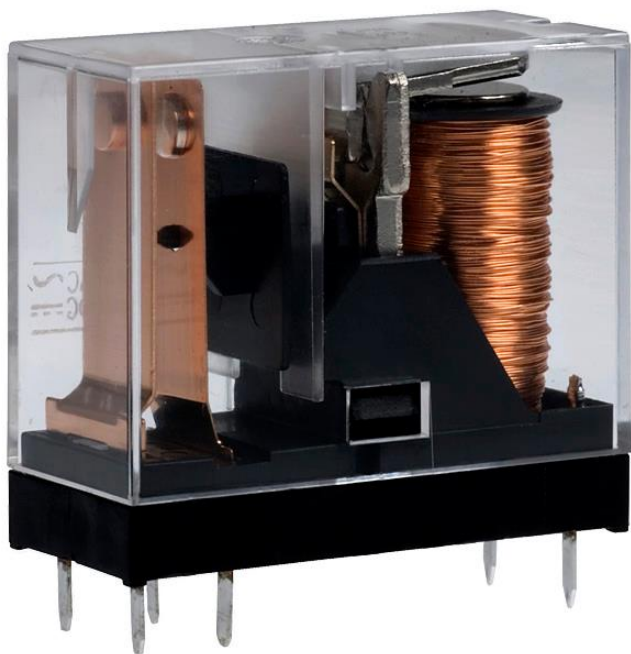
$$C = A \wedge B$$



Relé (Relay en inglés)



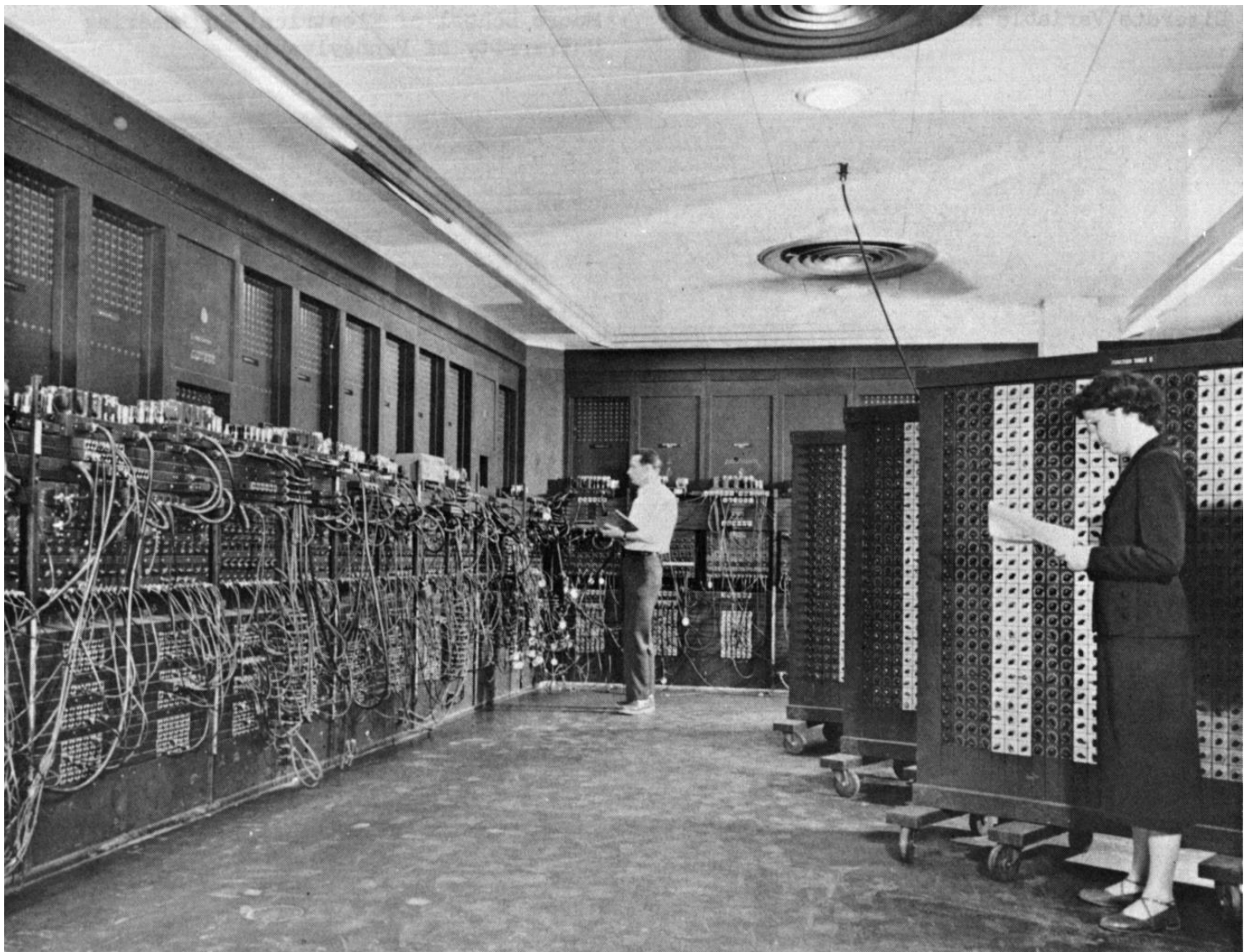
Z3 (1941)



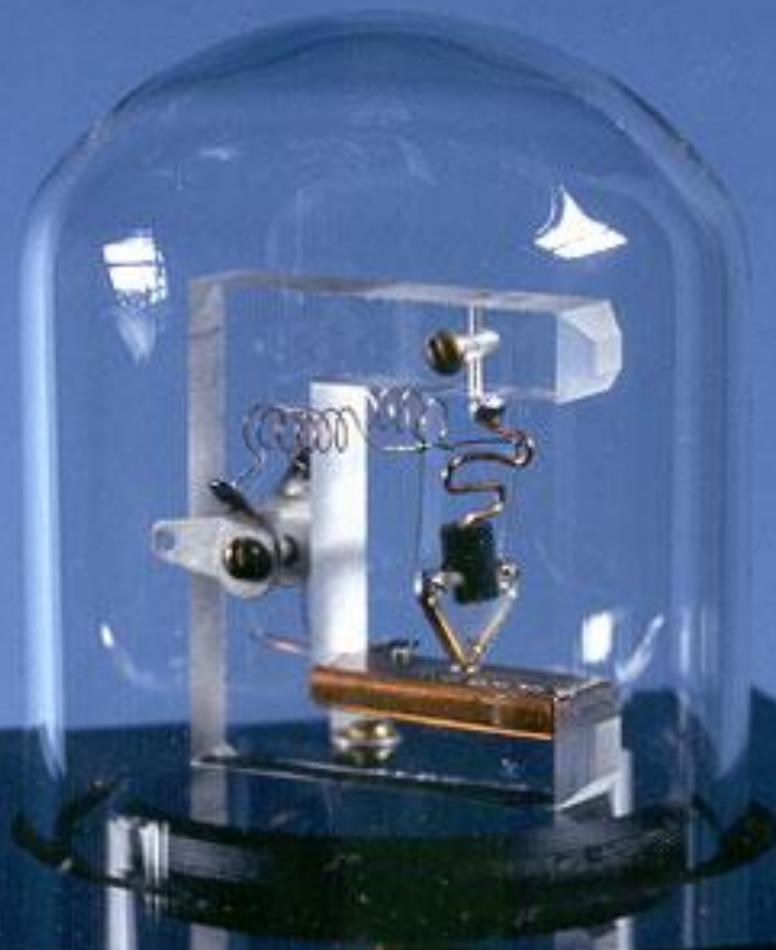
Relé (Relay en inglés)

Tubo de vacío





ENIAC (1945)



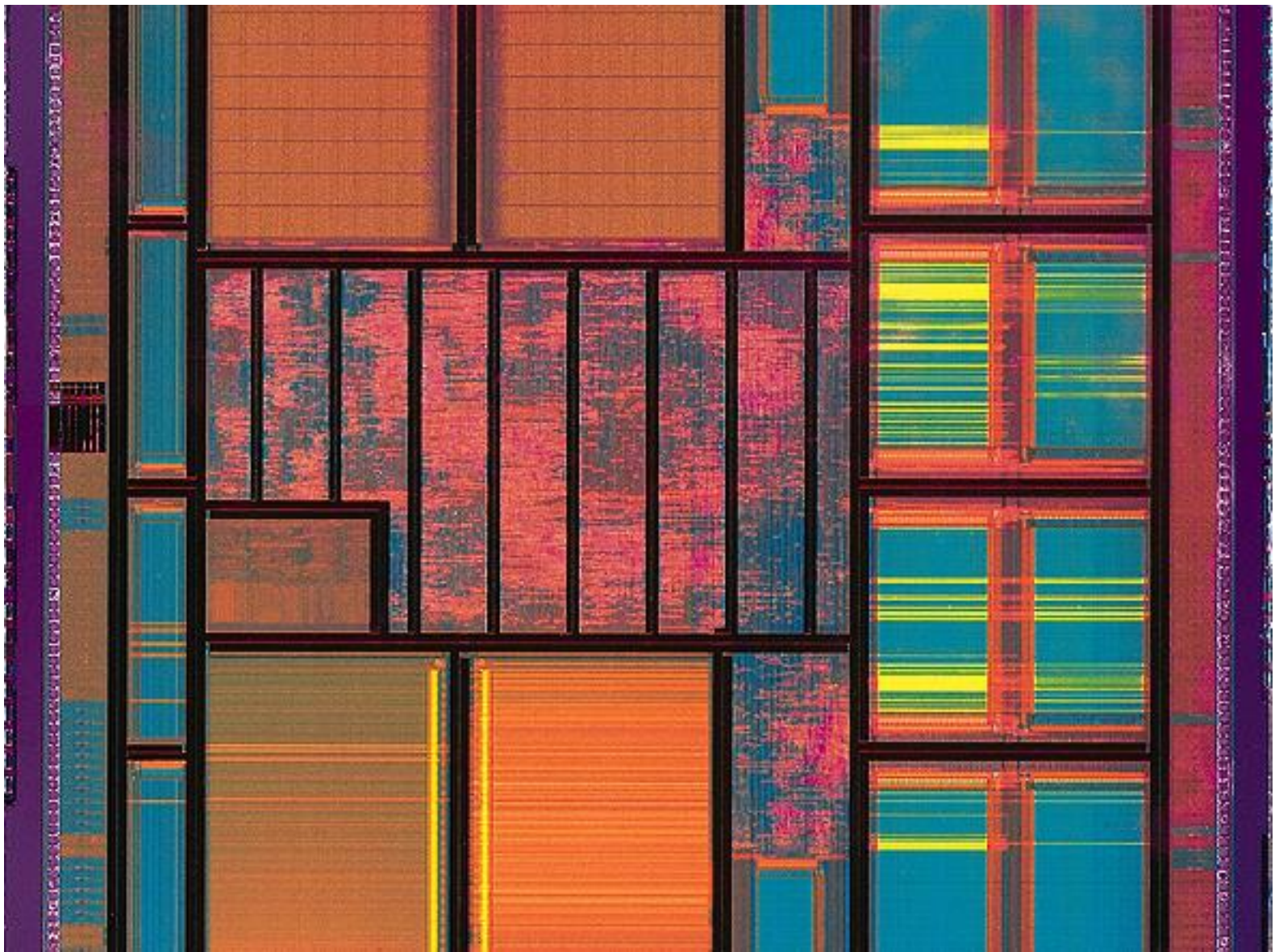
microelectronics group

Lucent Technologies
Bell Labs Innovations



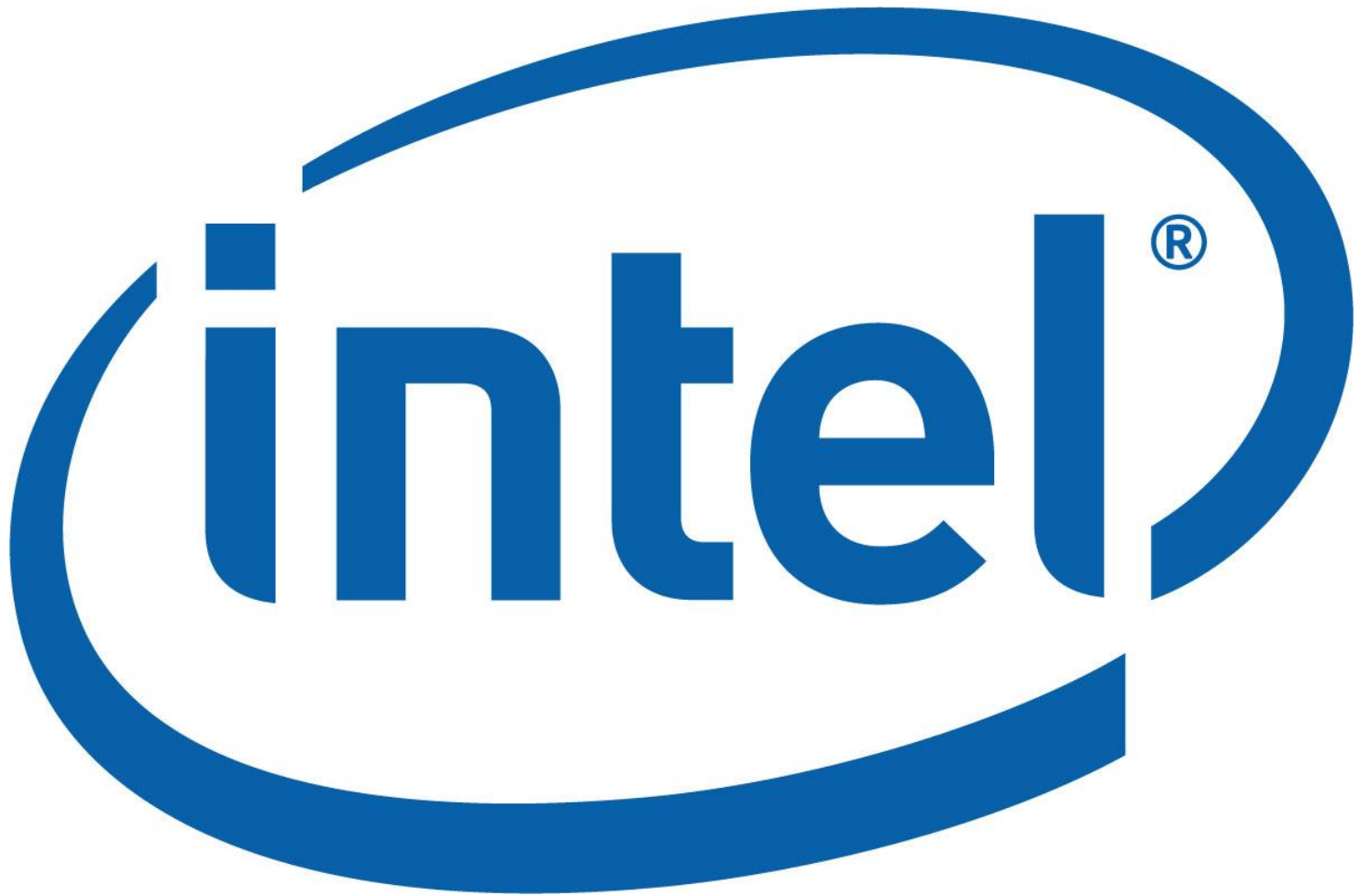
A replica of the first transistor,
invented at Bell Labs,
December 23, 1947

50 Years and Counting...



Circuito integrado o chip
(1958,1959)

Intel, 1968

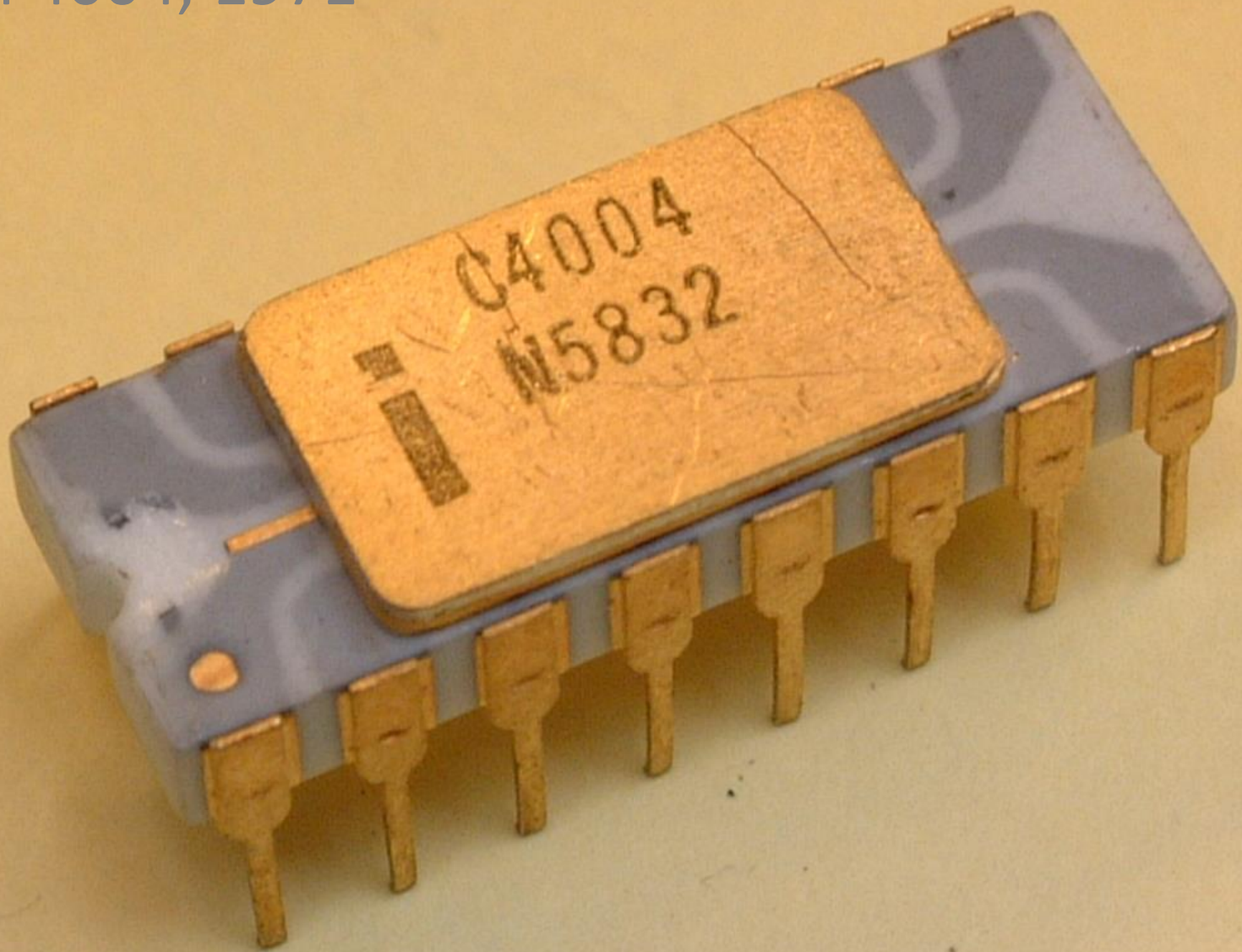




Gordon
Moore

Robert
Noyce

Intel 4004, 1971



Our World
in Data

This scatter plot illustrates the exponential growth of transistor counts in integrated circuits over time, from 1970 to 2018. The y-axis represents the transistor count on a logarithmic scale, ranging from 1,000 to 50,000,000,000. The x-axis represents the year. The data points are labeled with various processor models, showing the rapid increase in complexity over time.

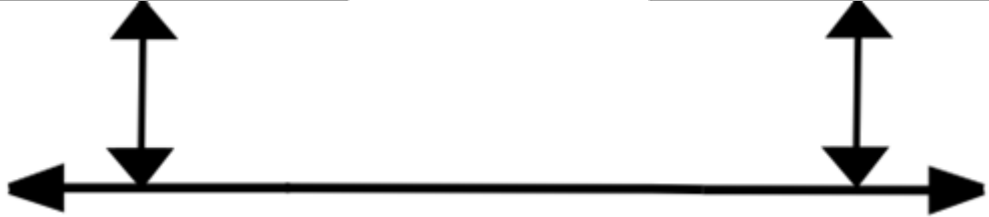
Key labeled data points include:

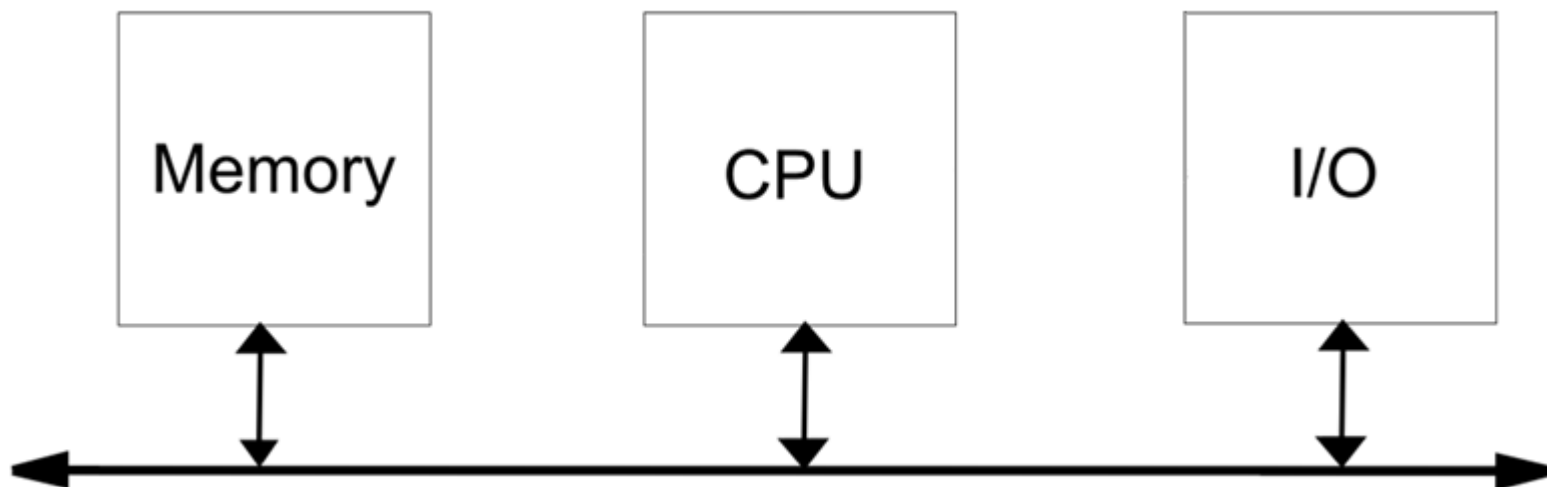
- Intel 4004 (1971)
- Intel 8008 (1972)
- Intel 8080 (1974)
- Intel 8085 (1976)
- Intel 8086 (1978)
- Intel 8088 (1982)
- Intel 80286 (1985)
- Intel 80386 (1985)
- Intel 80486 (1989)
- Intel Pentium (1993)
- Intel Pentium Pro (1995)
- Intel Pentium II (1997)
- Intel Pentium III (1999)
- Intel Pentium 4 (2000)
- Intel Pentium D (2002)
- Intel Pentium E (2004)
- Intel Pentium Xeon (2006)
- Intel Pentium Xeon Phi (2013)
- Intel Pentium Xeon E5 (2016)
- Intel Pentium Xeon Phi 7200 (2017)
- Intel Pentium Xeon Phi 9400 (2018)
- AMD K5 (1994)
- AMD K6 (1995)
- AMD K7 (1999)
- AMD K8 (2005)
- AMD K10 (2008)
- AMD K11 (2011)
- AMD K12 (2016)
- ARM 1 (1985)
- ARM 2 (1987)
- ARM 3 (1989)
- ARM 6 (1991)
- ARM 9TDMI (1997)
- ARM Cortex-A9 (2008)
- ARM Cortex-A15 (2012)
- ARM Cortex-A58 (2015)
- ARM Cortex-A72 (2016)
- ARM Cortex-X1 (2018)
- ARM Cortex-M0 (2004)
- ARM Cortex-M23 (2016)
- ARM Cortex-M55 (2018)
- ARM Cortex-M85 (2018)
- ARM Cortex-M88 (2018)
- ARM Cortex-M89 (2018)
- ARM Cortex-M90 (2018)
- ARM Cortex-M91 (2018)
- ARM Cortex-M92 (2018)
- ARM Cortex-M93 (2018)
- ARM Cortex-M94 (2018)
- ARM Cortex-M95 (2018)
- ARM Cortex-M96 (2018)
- ARM Cortex-M97 (2018)
- ARM Cortex-M98 (2018)
- ARM Cortex-M99 (2018)
- ARM Cortex-M100 (2018)
- ARM Cortex-M101 (2018)
- ARM Cortex-M102 (2018)
- ARM Cortex-M103 (2018)
- ARM Cortex-M104 (2018)
- ARM Cortex-M105 (2018)
- ARM Cortex-M106 (2018)
- ARM Cortex-M107 (2018)
- ARM Cortex-M108 (2018)
- ARM Cortex-M109 (2018)
- ARM Cortex-M110 (2018)
- ARM Cortex-M111 (2018)
- ARM Cortex-M112 (2018)
- ARM Cortex-M113 (2018)
- ARM Cortex-M114 (2018)
- ARM Cortex-M115 (2018)
- ARM Cortex-M116 (2018)
- ARM Cortex-M117 (2018)
- ARM Cortex-M118 (2018)
- ARM Cortex-M119 (2018)
- ARM Cortex-M120 (2018)
- ARM Cortex-M121 (2018)
- ARM Cortex-M122 (2018)
- ARM Cortex-M123 (2018)
- ARM Cortex-M124 (2018)
- ARM Cortex-M125 (2018)
- ARM Cortex-M126 (2018)
- ARM Cortex-M127 (2018)
- ARM Cortex-M128 (2018)
- ARM Cortex-M129 (2018)
- ARM Cortex-M130 (2018)
- ARM Cortex-M131 (2018)
- ARM Cortex-M132 (2018)
- ARM Cortex-M133 (2018)
- ARM Cortex-M134 (2018)
- ARM Cortex-M135 (2018)
- ARM Cortex-M136 (2018)
- ARM Cortex-M137 (2018)
- ARM Cortex-M138 (2018)
- ARM Cortex-M139 (2018)
- ARM Cortex-M140 (2018)
- ARM Cortex-M141 (2018)
- ARM Cortex-M142 (2018)
- ARM Cortex-M143 (2018)
- ARM Cortex-M144 (2018)
- ARM Cortex-M145 (2018)
- ARM Cortex-M146 (2018)
- ARM Cortex-M147 (2018)
- ARM Cortex-M148 (2018)
- ARM Cortex-M149 (2018)
- ARM Cortex-M150 (2018)
- ARM Cortex-M151 (2018)
- ARM Cortex-M152 (2018)
- ARM Cortex-M153 (2018)
- ARM Cortex-M154 (2018)
- ARM Cortex-M155 (2018)
- ARM Cortex-M156 (2018)
- ARM Cortex-M157 (2018)
- ARM Cortex-M158 (2018)
- ARM Cortex-M159 (2018)
- ARM Cortex-M160 (2018)
- ARM Cortex-M161 (2018)
- ARM Cortex-M162 (2018)
- ARM Cortex-M163 (2018)
- ARM Cortex-M164 (2018)
- ARM Cortex-M165 (2018)
- ARM Cortex-M166 (2018)
- ARM Cortex-M167 (2018)
- ARM Cortex-M168 (2018)
- ARM Cortex-M169 (2018)
- ARM Cortex-M170 (2018)
- ARM Cortex-M171 (2018)
- ARM Cortex-M172 (2018)
- ARM Cortex-M173 (2018)
- ARM Cortex-M174 (2018)
- ARM Cortex-M175 (2018)
- ARM Cortex-M176 (2018)
- ARM Cortex-M177 (2018)
- ARM Cortex-M178 (2018)
- ARM Cortex-M179 (2018)
- ARM Cortex-M180 (2018)
- ARM Cortex-M181 (2018)
- ARM Cortex-M182 (2018)
- ARM Cortex-M183 (2018)
- ARM Cortex-M184 (2018)
- ARM Cortex-M185 (2018)
- ARM Cortex-M186 (2018)
- ARM Cortex-M187 (2018)
- ARM Cortex-M188 (2018)
- ARM Cortex-M189 (2018)
- ARM Cortex-M190 (2018)
- ARM Cortex-M191 (2018)
- ARM Cortex-M192 (2018)
- ARM Cortex-M193 (2018)
- ARM Cortex-M194 (2018)
- ARM Cortex-M195 (2018)
- ARM Cortex-M196 (2018)
- ARM Cortex-M197 (2018)
- ARM Cortex-M198 (2018)
- ARM Cortex-M199 (2018)
- ARM Cortex-M200 (2018)
- ARM Cortex-M201 (2018)
- ARM Cortex-M202 (2018)
- ARM Cortex-M203 (2018)
- ARM Cortex-M204 (2018)
- ARM Cortex-M205 (2018)
- ARM Cortex-M206 (2018)
- ARM Cortex-M207 (2018)
- ARM Cortex-M208 (2018)
- ARM Cortex-M209 (2018)
- ARM Cortex-M210 (2018)
- ARM Cortex-M211 (2018)
- ARM Cortex-M212 (2018)
- ARM Cortex-M213 (2018)
- ARM Cortex-M214 (2018)
- ARM Cortex-M215 (2018)
- ARM Cortex-M216 (2018)
- ARM Cortex-M217 (2018)
- ARM Cortex-M218 (2018)
- ARM Cortex-M219 (2018)
- ARM Cortex-M220 (2018)
- ARM Cortex-M221 (2018)
- ARM Cortex-M222 (2018)
- ARM Cortex-M223 (2018)
- ARM Cortex-M224 (2018)
- ARM Cortex-M225 (2018)
- ARM Cortex-M226 (2018)
- ARM Cortex-M227 (2018)
- ARM Cortex-M228 (2018)
- ARM Cortex-M229 (2018)
- ARM Cortex-M230 (2018)
- ARM Cortex-M231 (2018)
- ARM Cortex-M232 (2018)
- ARM Cortex-M233 (2018)
- ARM Cortex-M234 (2018)
- ARM Cortex-M235 (2018)
- ARM Cortex-M236 (2018)
- ARM Cortex-M237 (2018)
- ARM Cortex-M238 (2018)
- ARM Cortex-M239 (2018)
- ARM Cortex-M240 (2018)
- ARM Cortex-M241 (2018)
- ARM Cortex-M242 (2018)
- ARM Cortex-M243 (2018)
- ARM Cortex-M244 (2018)
- ARM Cortex-M245 (2018)
- ARM Cortex-M246 (2018)
- ARM Cortex-M247 (2018)
- ARM Cortex-M248 (2018)
- ARM Cortex-M249 (2018)
- ARM Cortex-M250 (2018)
- ARM Cortex-M251 (2018)
- ARM Cortex-M252 (2018)
- ARM Cortex-M253 (2018)
- ARM Cortex-M254 (2018)
- ARM Cortex-M255 (2018)
- ARM Cortex-M256 (2018)
- ARM Cortex-M257 (2018)
- ARM Cortex-M258 (2018)
- ARM Cortex-M259 (2018)
- ARM Cortex-M260 (2018)
- ARM Cortex-M261 (2018)
- ARM Cortex-M262 (2018)
- ARM Cortex-M263 (2018)
- ARM Cortex-M264 (2018)
- ARM Cortex-M265 (2018)
- ARM Cortex-M266 (2018)
- ARM Cortex-M267 (2018)
- ARM Cortex-M268 (2018)
- ARM Cortex-M269 (2018)
- ARM Cortex-M270 (2018)
- ARM Cortex-M271 (2018)
- ARM Cortex-M272 (2018)
- ARM Cortex-M273 (2018)
- ARM Cortex-M274 (2018

Licensed under [CC-BY-SA](#) by the author Max Roser.

Veamos ahora que tiene por
dentro un computador

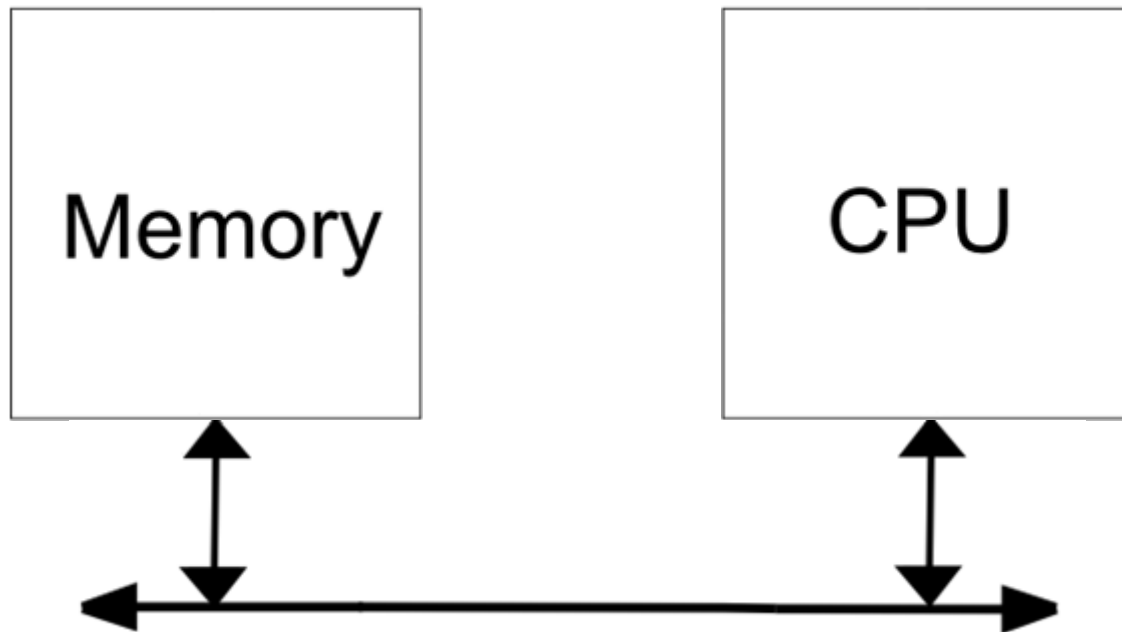
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |

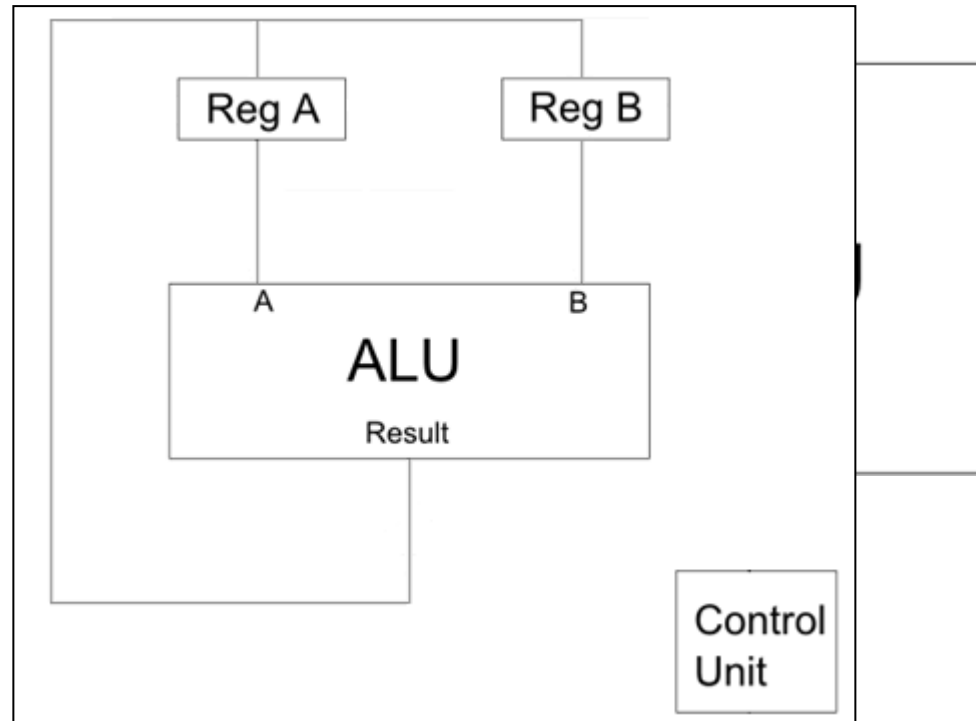




Muy bien por el hardware, pero ¿qué pasa con las instrucciones (programas)?

- Las instrucciones son almacenadas en la memoria (RAM).
- Luego, una instrucción es una secuencia de 0s y 1s.
- Esta secuencia es recibida en la CPU para ser ejecutada.
- Cada posible instrucción del computador está asociada a sólo una secuencia de 0s y 1s.





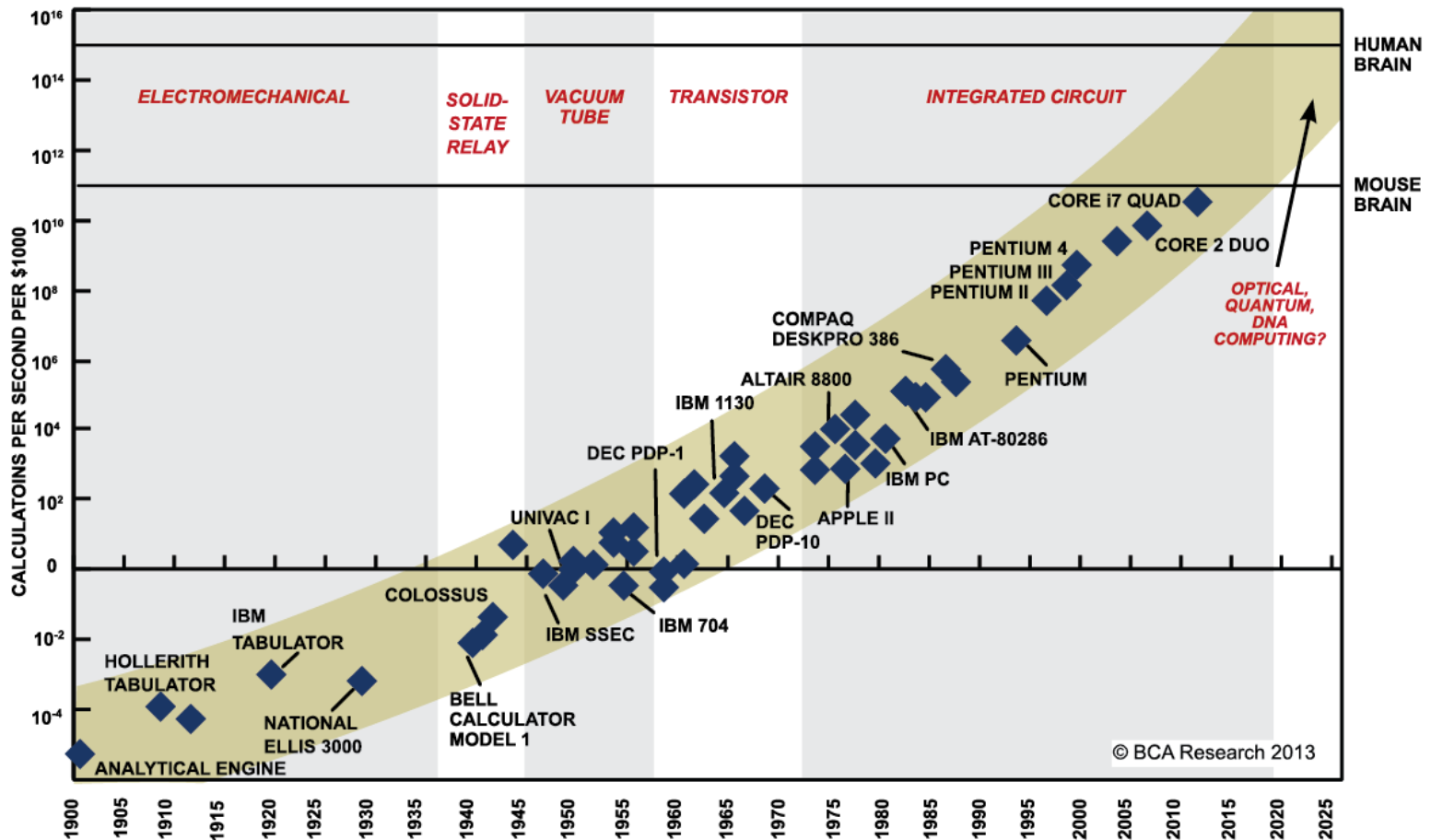
Un ejemplo muy simple

- Supongamos que tenemos la instrucción **01010011**, que corresponde a sumar el contenido de los regs. A y B, para luego almacenarlo nuevamente en A.
- Llamaremos a esta instrucción **ADD A, B**.
- La unidad de control, al recibir 01010011, chequea si es una instrucción válida y luego la decodifica, enviando a cada elemento la orden correspondiente.
- En este caso, le dice a la ALU que sume sus dos entradas, y le dice al reg. A, que almacene el resultado de la ALU.

El **flujo completo** de una instrucción puede describirse de la siguiente manera

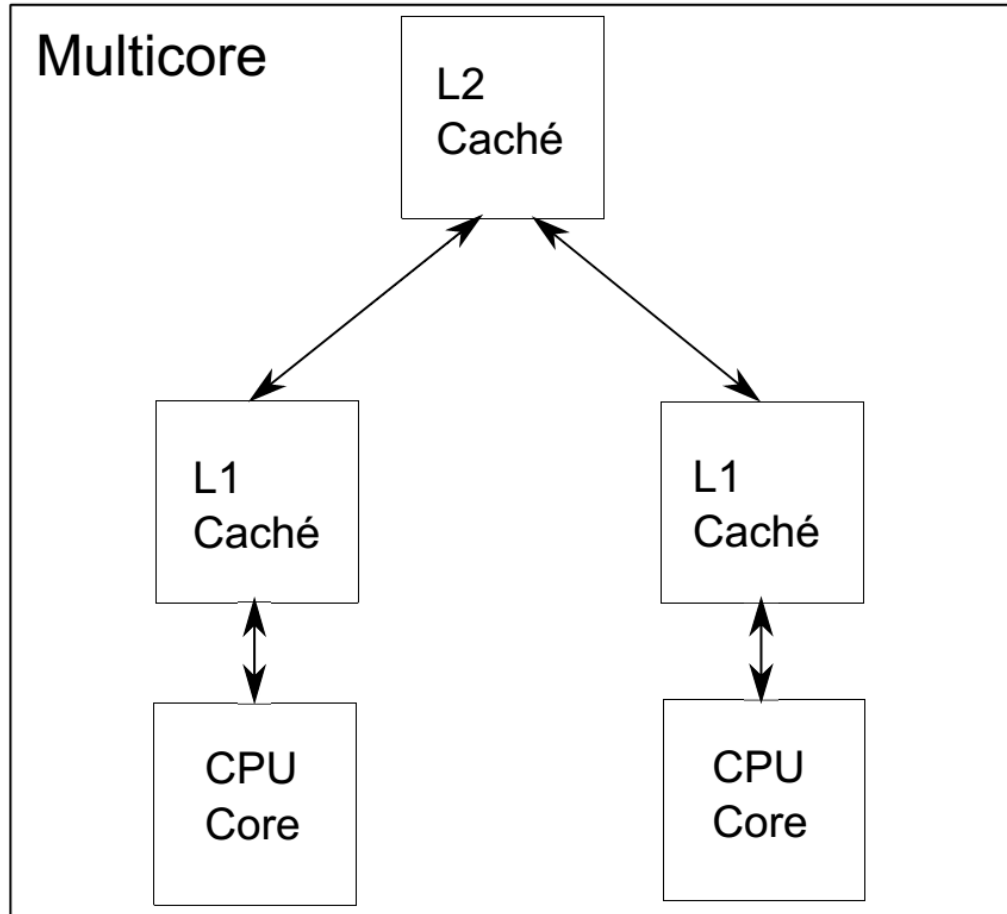
1. Lectura de instrucción desde memoria (**Fetch**)
2. **Decode** (Unidad de Control)
3. Obtener dato de memoria o registros (**Mem**)
4. **Execute** (ALU)
5. Escribir resultado en memoria o registros (**Write Back**)

El futuro y la importancia de las arquitecturas avanzadas

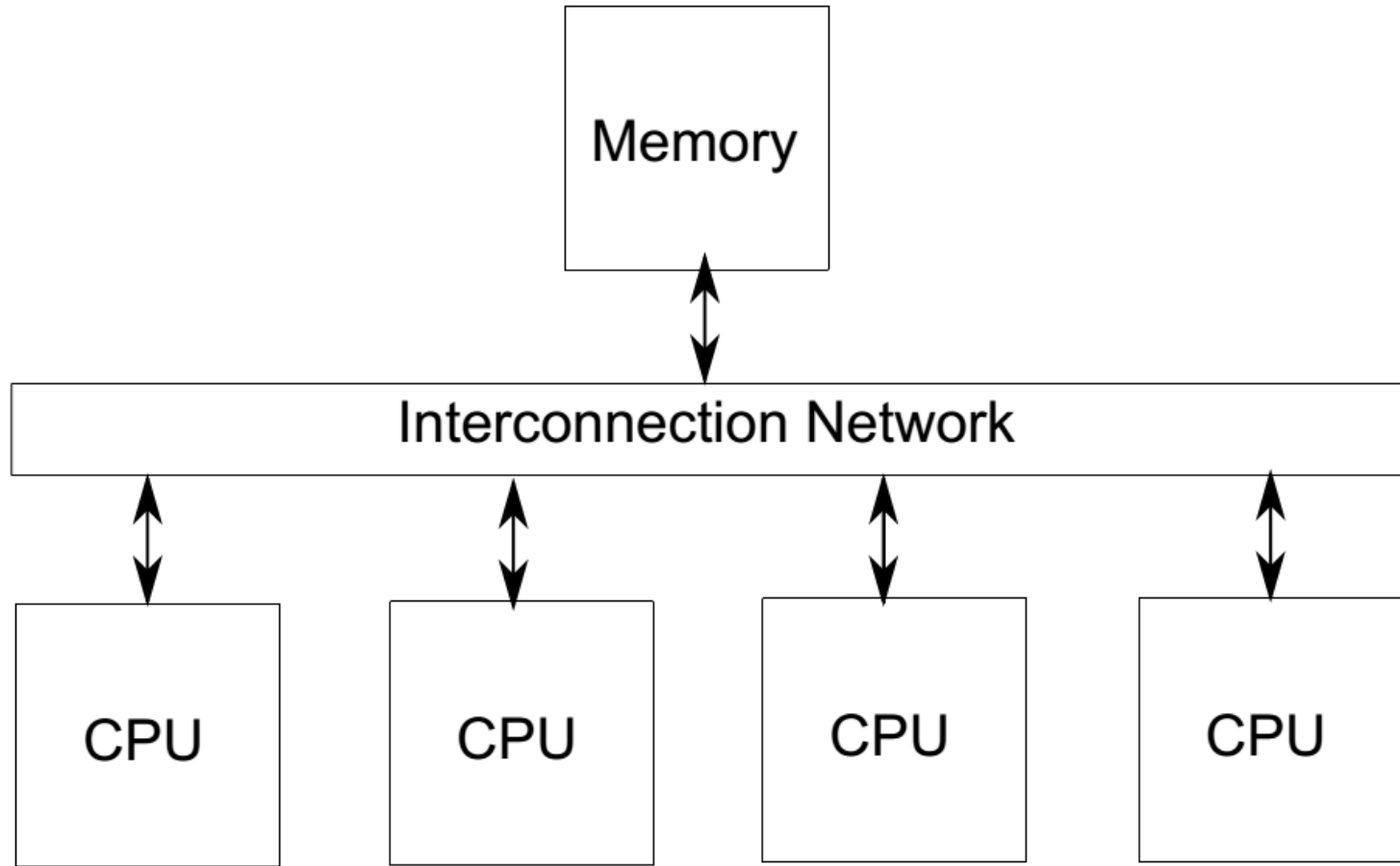


SOURCE: RAY KURZWEIL, "THE SINGULARITY IS NEAR: WHEN HUMANS TRANSCEND BIOLOGY", P.67, THE VIKING PRESS, 2006. DATAPOINTS BETWEEN 2000 AND 2012 REPRESENT BCA ESTIMATES.

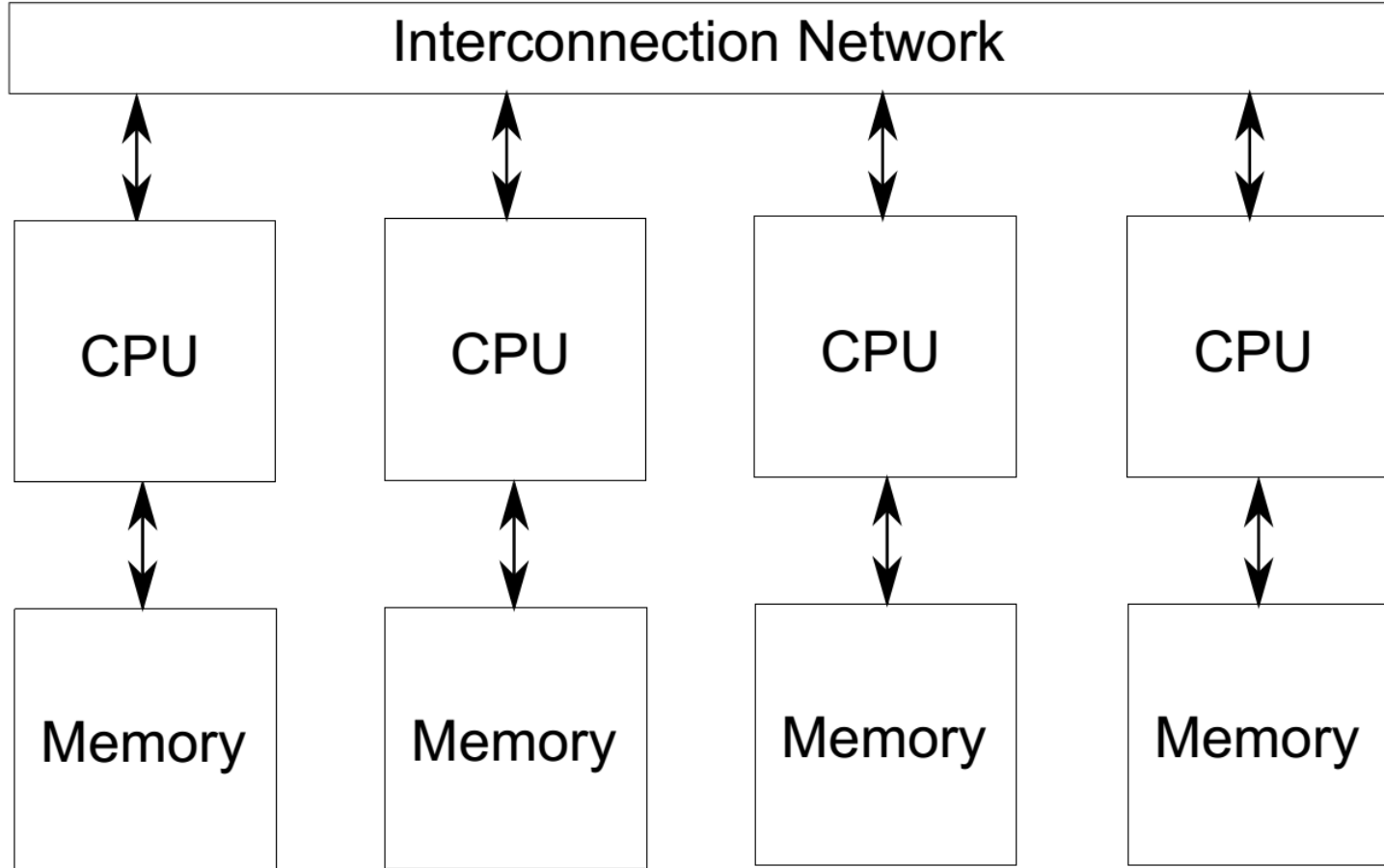
Procesadores *multicore* entregan un mecanismo para tener multiprocesamiento en un solo chip



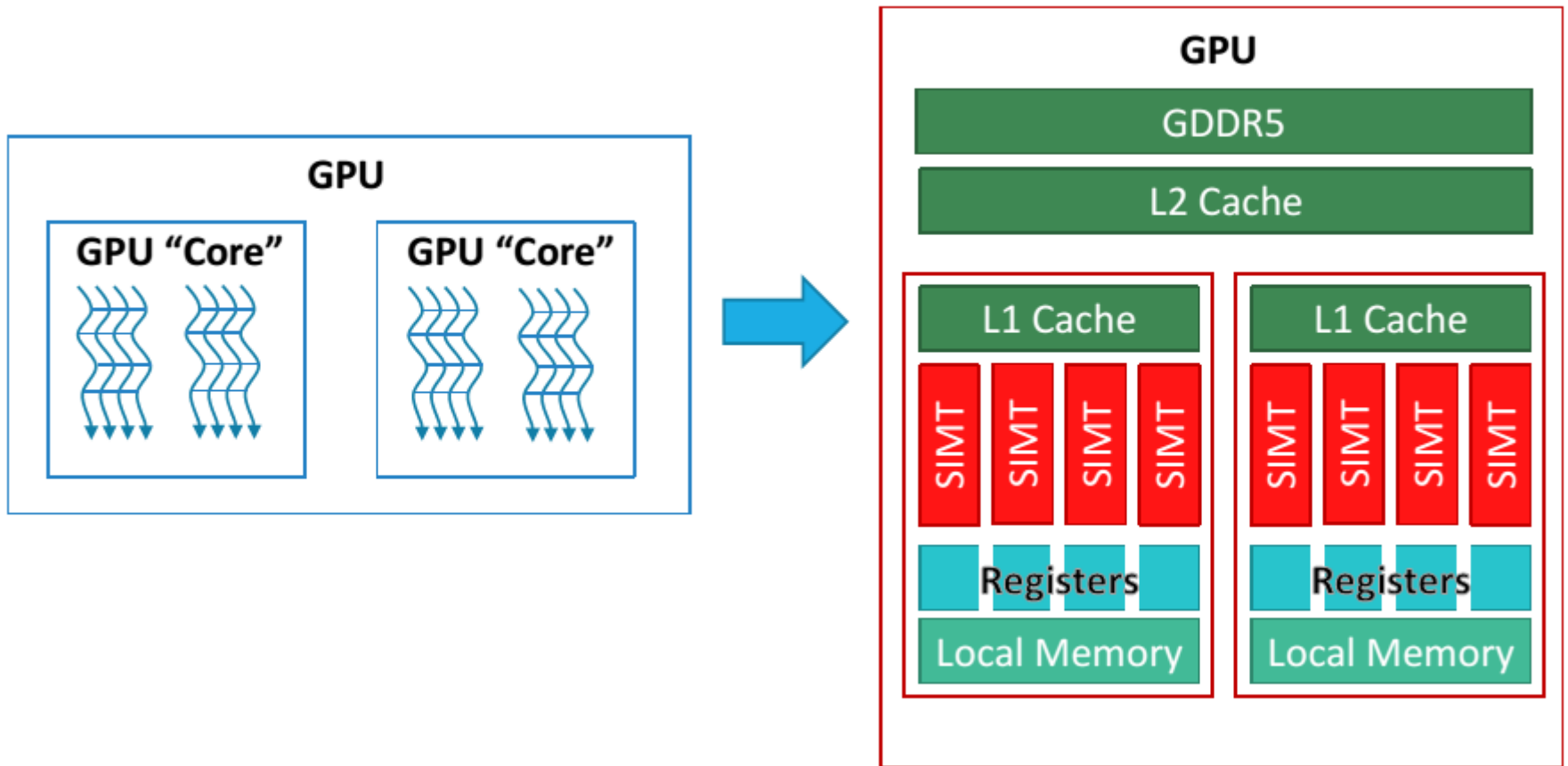
Multiprocesador con **memoria compartida** necesita mecanismos para mantener coherencia en memoria



Multiprocesador por paso de mensajes
puede ser un **cluster** o un **sistema distribuido**

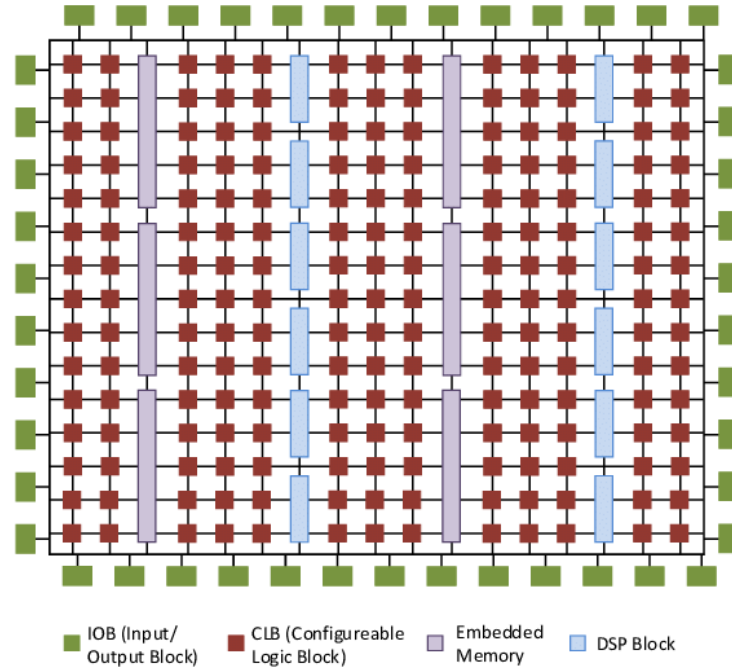


GPUs contienen cientos de pequeños procesadores

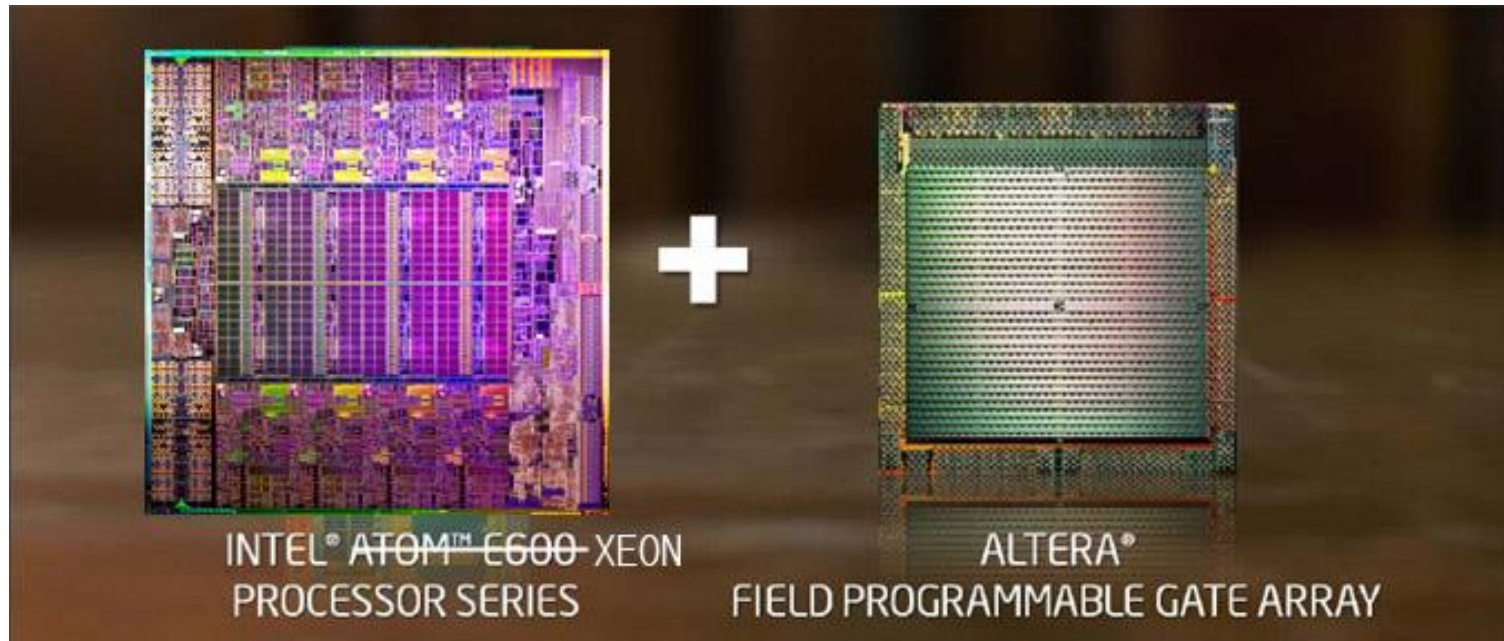


Para terminar, una pequeña muestra de la dimensión de la industria de la arquitectura de computadores

FPGA son tarjetas con compuertas lógicas programables, que permiten construir arquitecturas nuevas fácilmente.



La mejor solución es justamente construir
la arquitectura adecuada al problema



16.7 billones de USD

La mejor solución es justamente construir
la arquitectura adecuada al problema



400 millones de USD (45 personas)

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación



IIC1005 – Computación: Ciencia y Tecnología del Mundo Digital

Arquitectura de Computadores

Hans Löbel