

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

Los ingenieros entendemos las cosas en profundidad con el fin de aplicarlas correctamente.



# ¿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.
- La pregunta que no interesa ahora (y en la parte inicial del curso de arquitectura) es la siguiente:

**¿cómo construimos una máquina programable que ejecuta programas?**



```
def calcular_promedio():  
    notas = [6, 4, 2, 3, 5]  
    n = 5  
    i = 0  
    promedio = 0  
    while i < n:  
        promedio += notas[i]  
        i += 1  
    promedio = promedio / n  
    print(promedio)
```



## En resumen

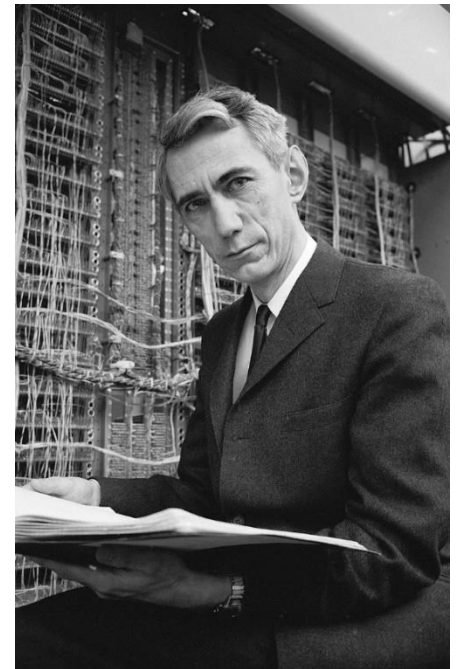
- ¿Qué es un computador?
  - Máquina **programable** que ejecuta programas
- Elementos necesarios para un computador
  - Datos: números enteros y reales, texto, imágenes, etc.
  - Variables: simples, listas
  - Decisiones: comparación, ciclos
  - Operaciones

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.
- Planteo 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.



Lógica booleana permite representar operaciones aritméticas





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

$$S = A \oplus B$$

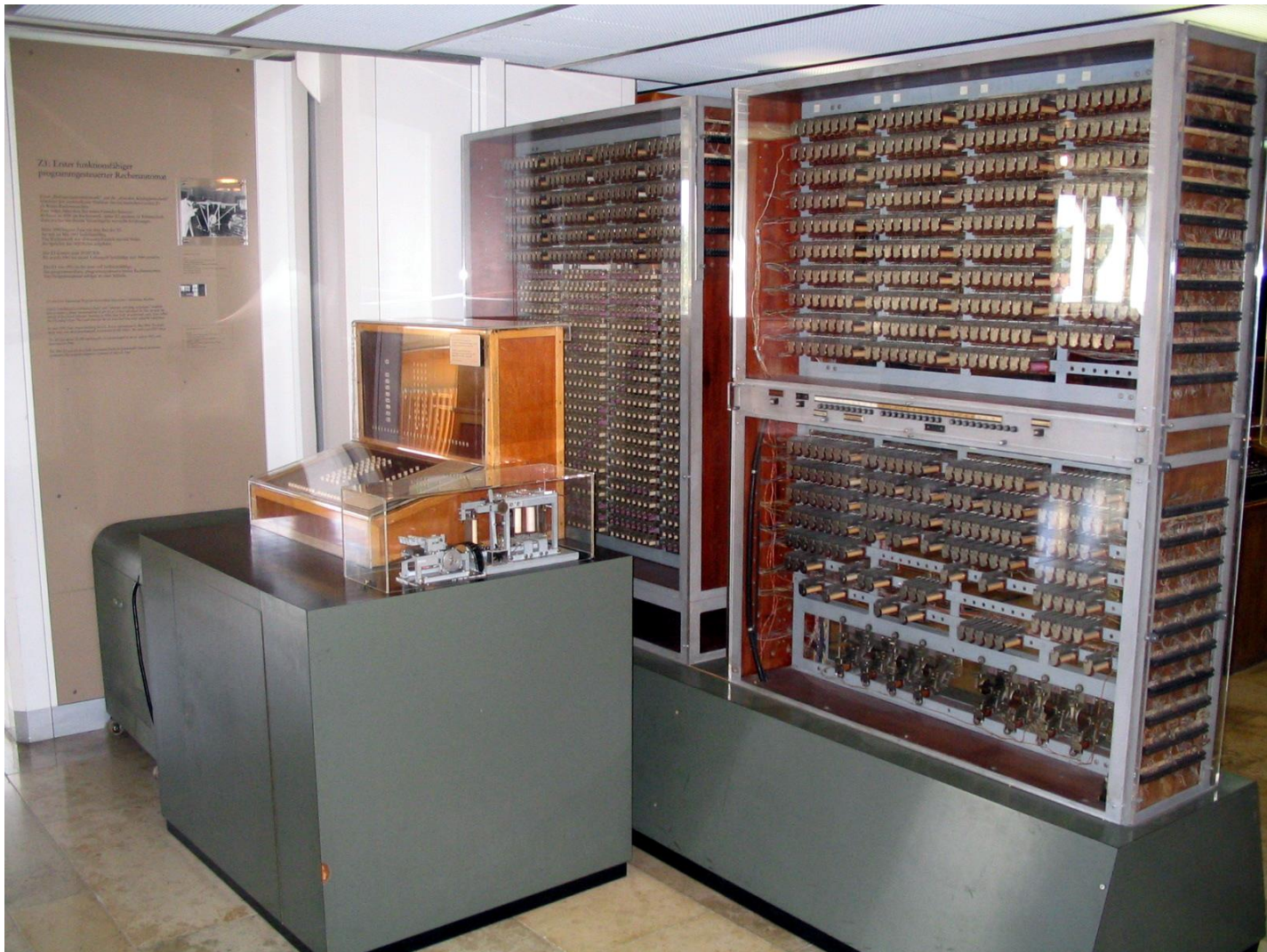
$$C = A \wedge B$$





Relé (Relay en inglés)





Z3 (1941)



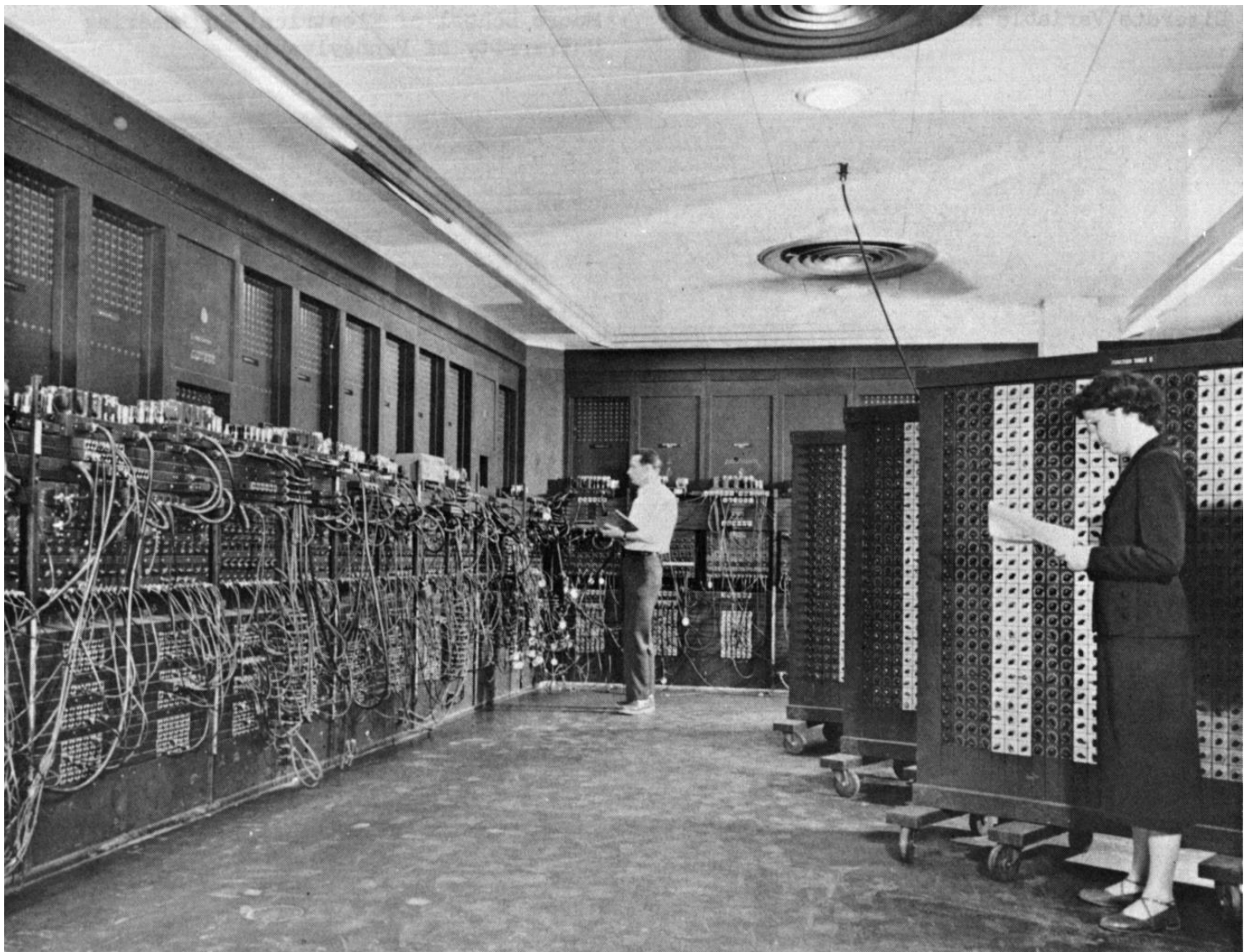


Relé (Relay en inglés)

Tubo de vacío

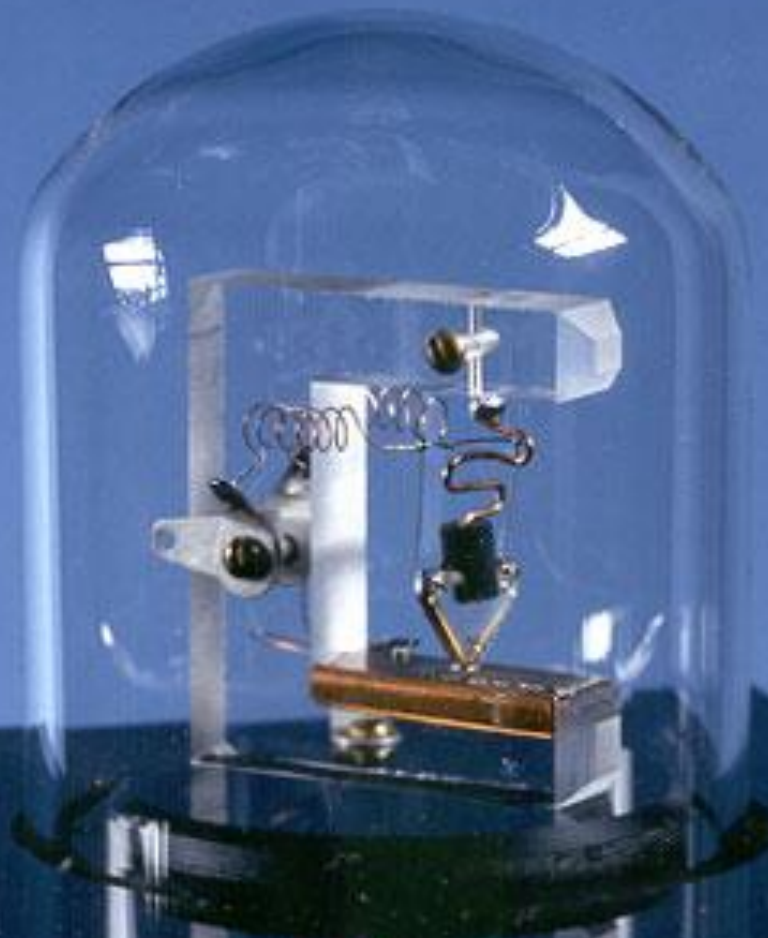






ENIAC (1945)





microelectronics group

Lucent Technologies  
Bell Labs Innovations

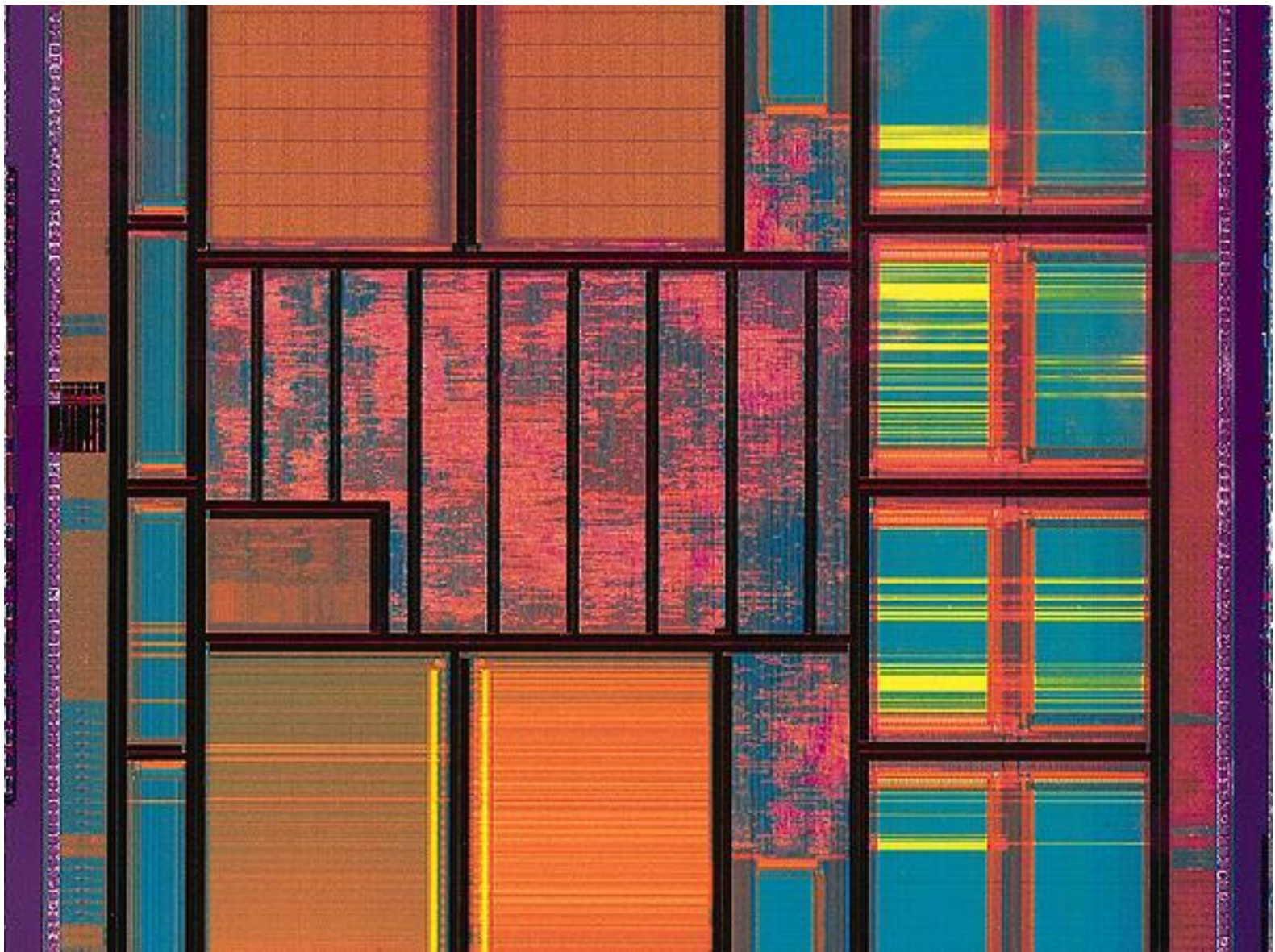


A replica of the first transistor,  
invented at Bell Labs,  
December 23, 1947

50 Years and Counting...



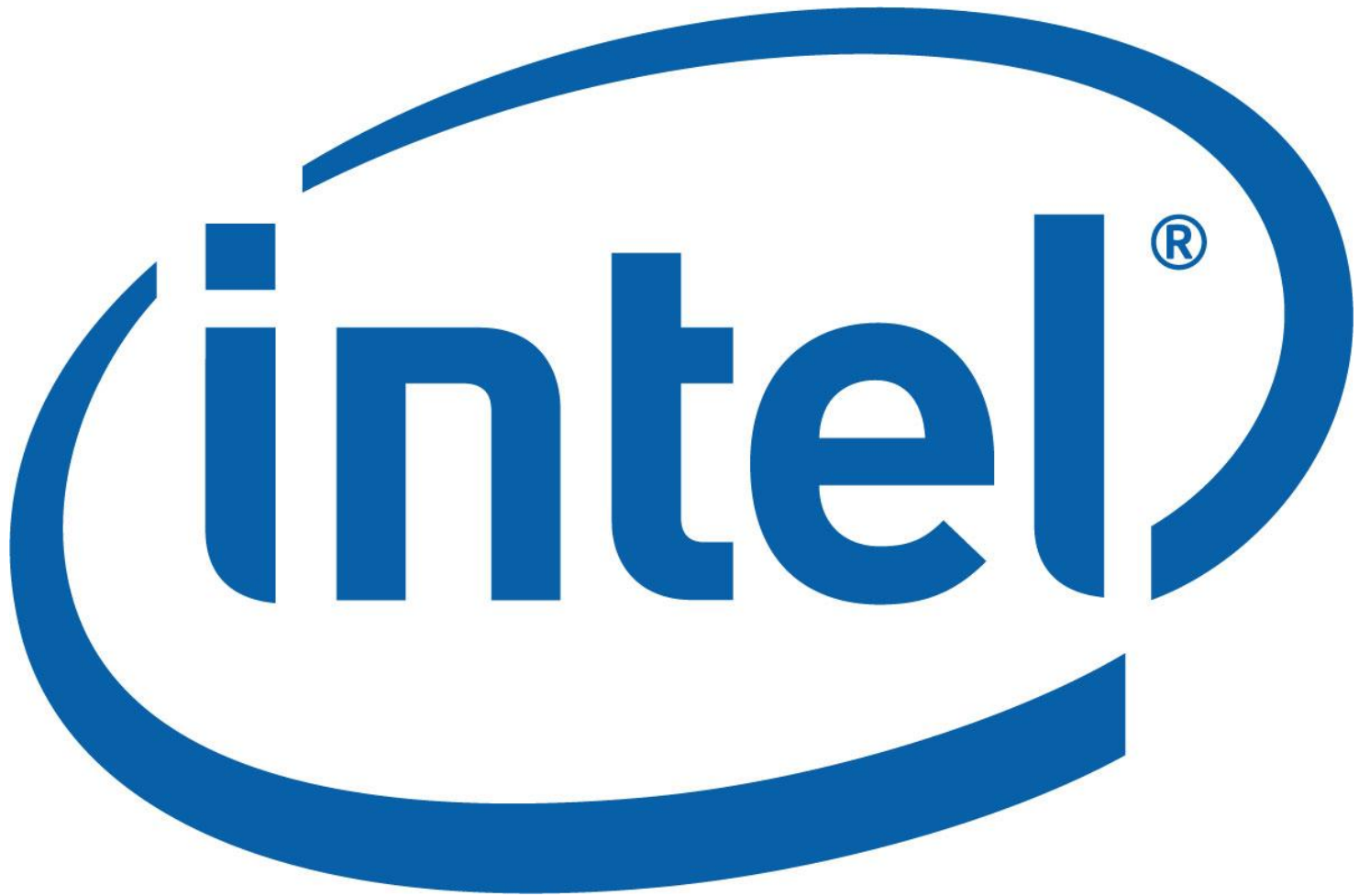




Circuito integrado o chip  
(1958,1959)




Intel, 1968



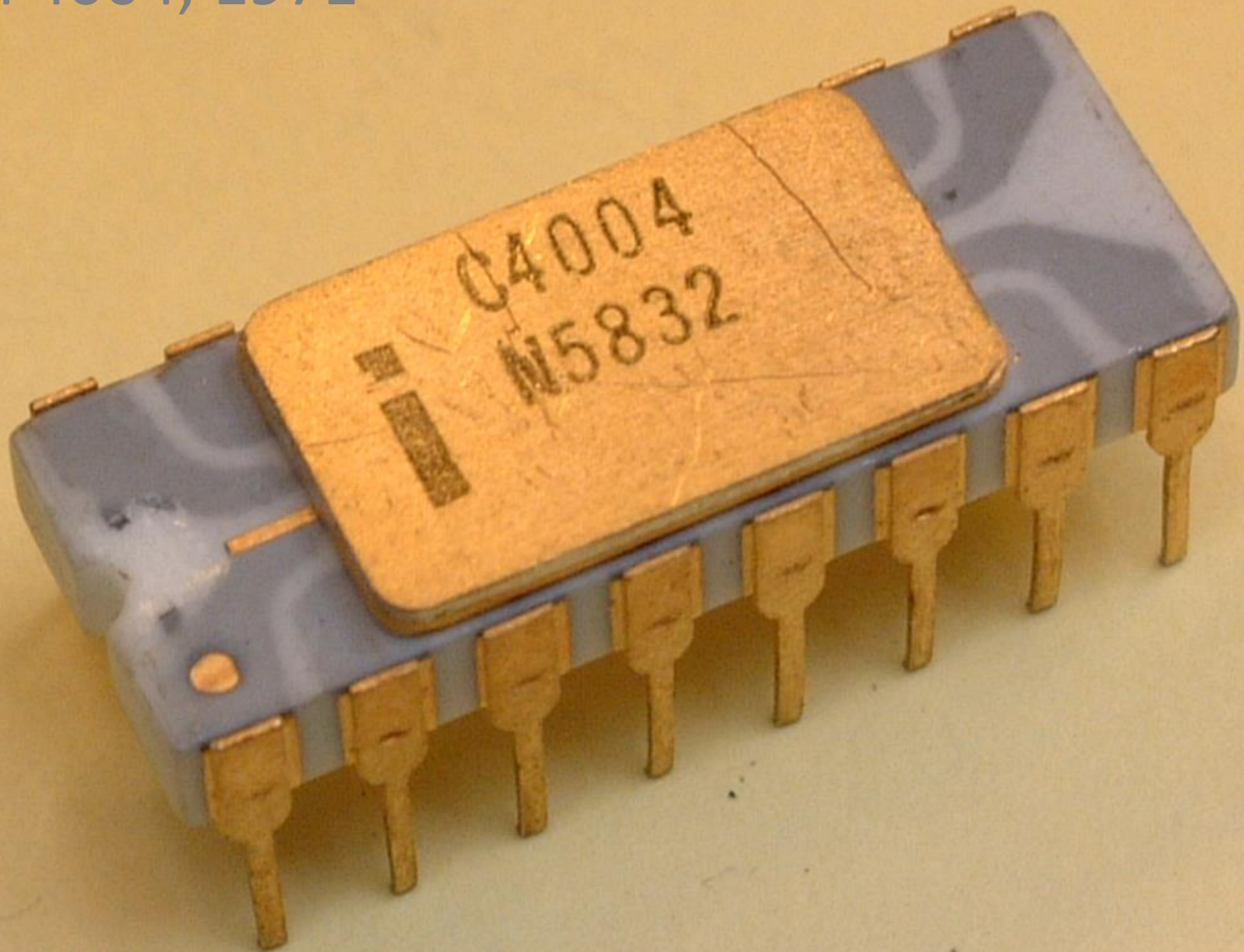




Gordon  
Moore

Robert  
Noyce 

Intel 4004, 1971



Our World  
in Data

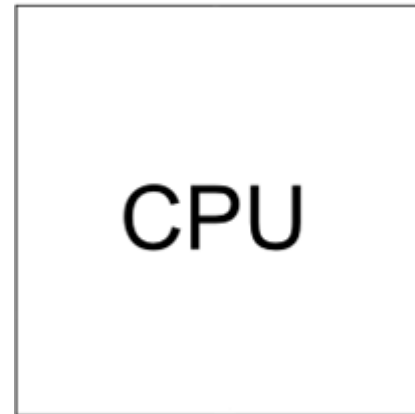
Data source: Wikipedia ([https://en.wikipedia.org/wiki/Transistor\\_count](https://en.wikipedia.org/wiki/Transistor_count))  
The data visualization is available at [OurWorldinData.org](https://ourworldindata.org). There you find more visualizations and research on this topic.

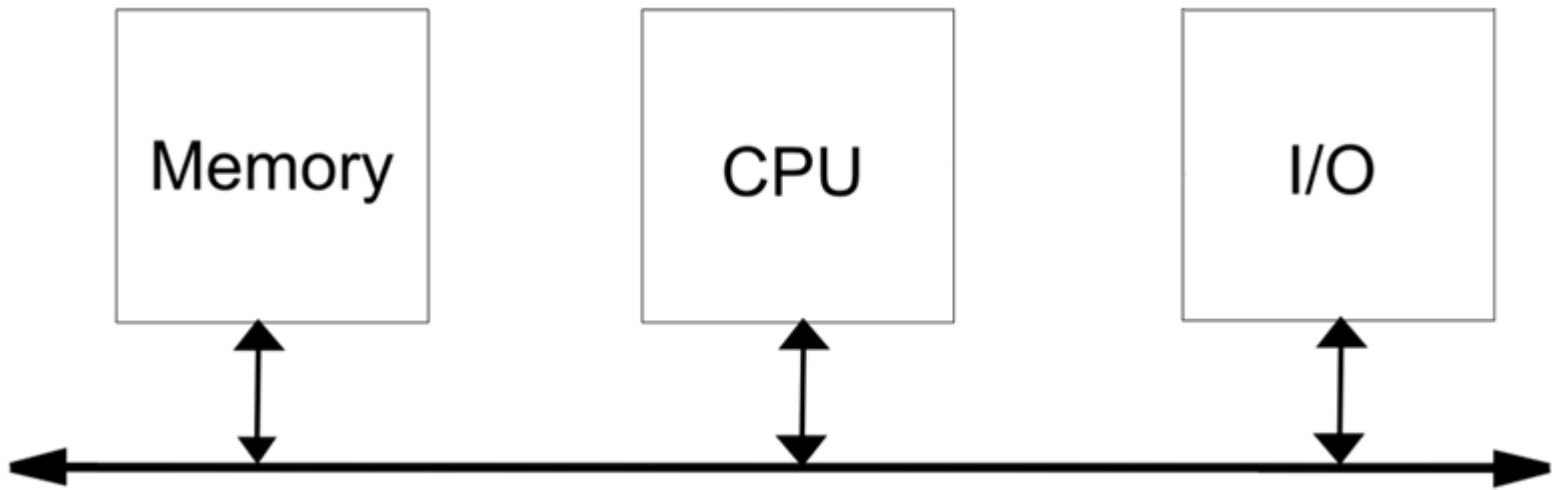


Veamos ahora que tiene por  
dentro un computador



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



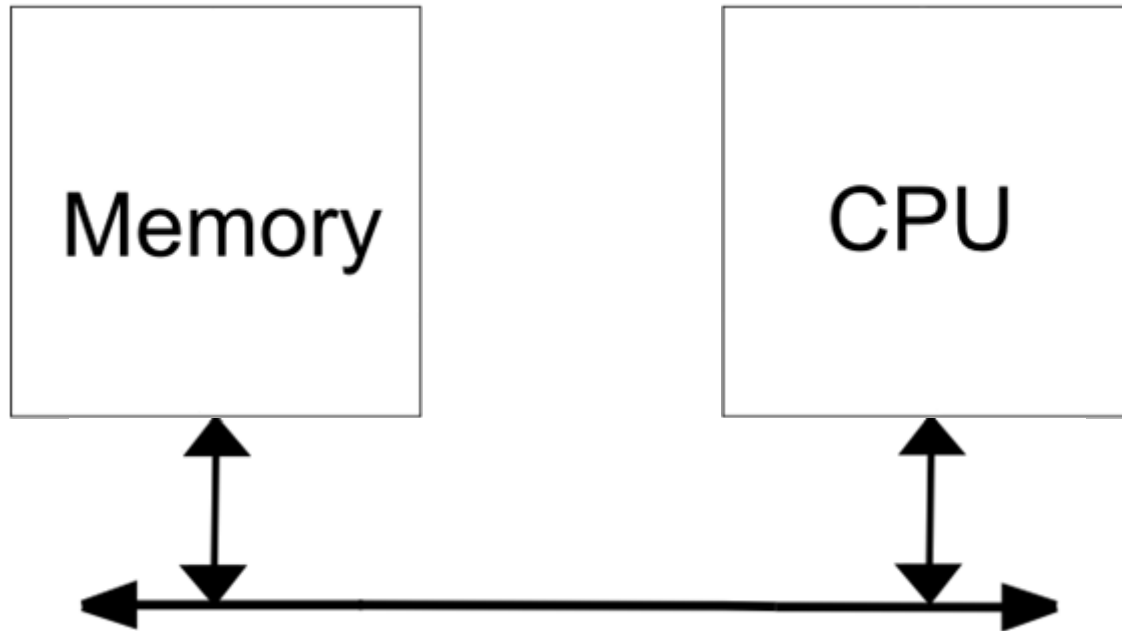




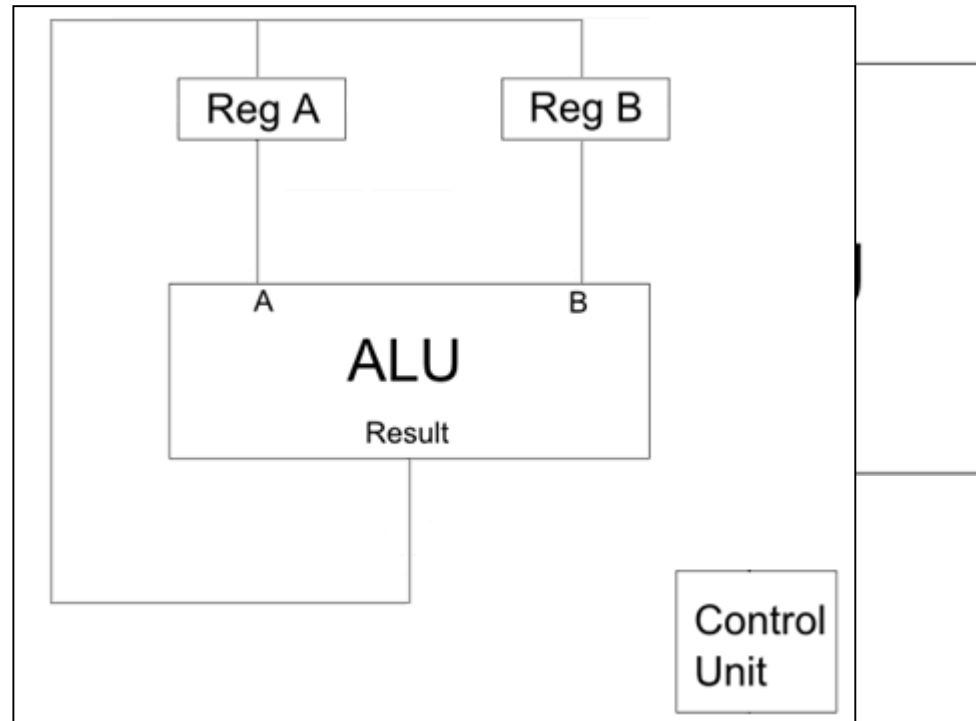
Muy bien por el hardware, pero ¿qué pasa con las instrucciones (programas)?

- 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.



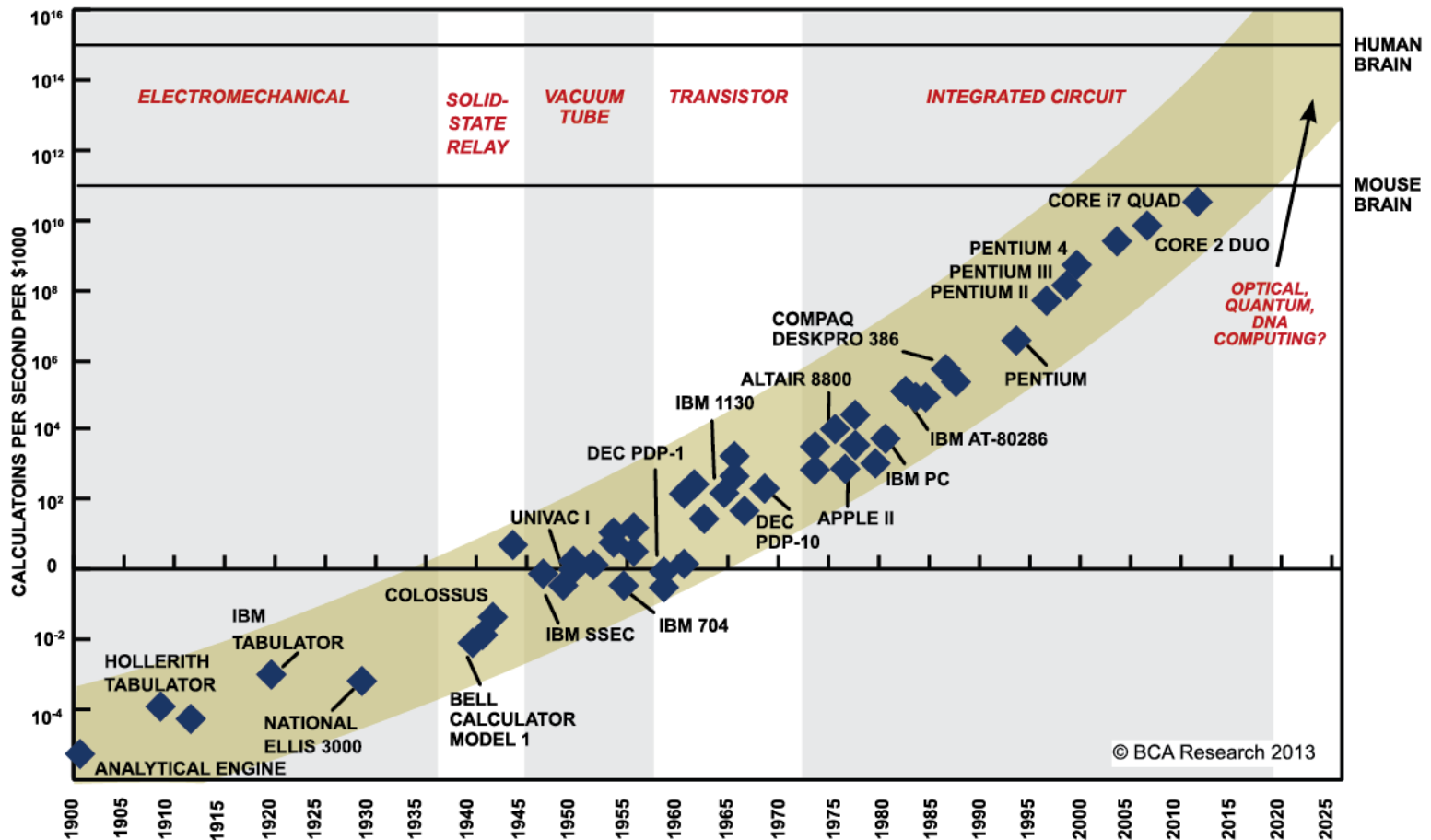
El **flujo completo** de una instrucción puede describirse de la siguiente manera

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

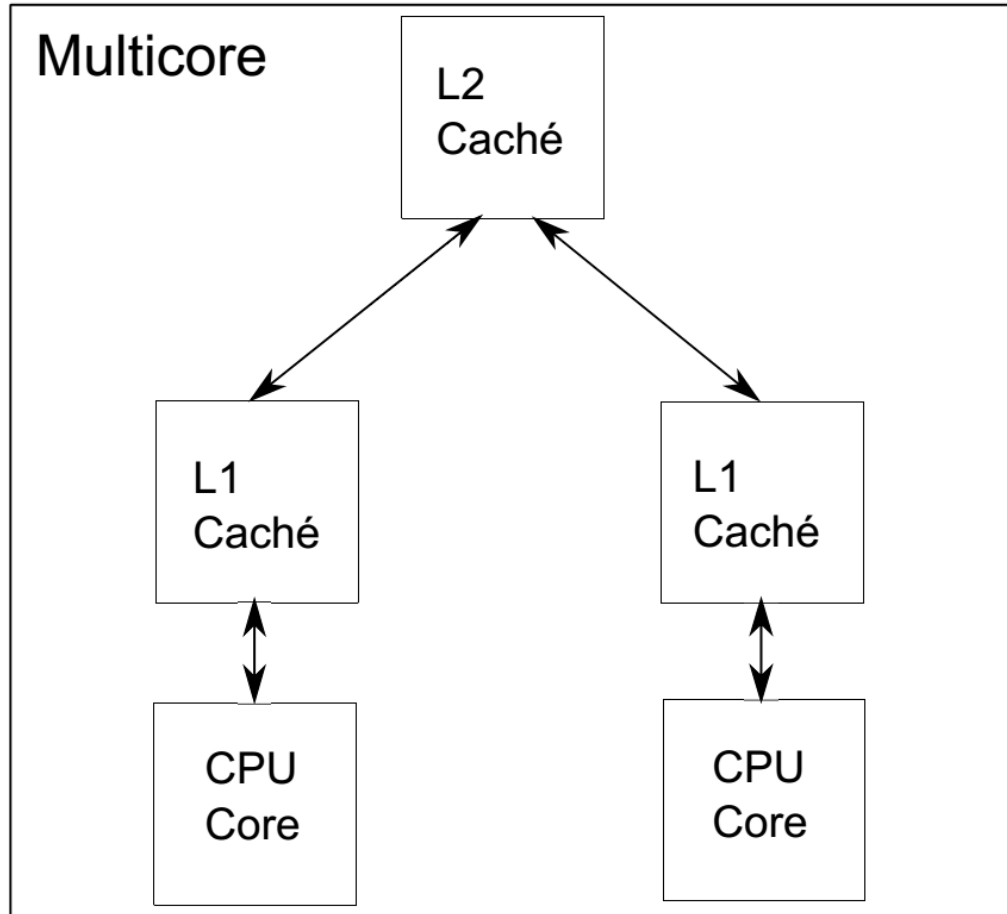




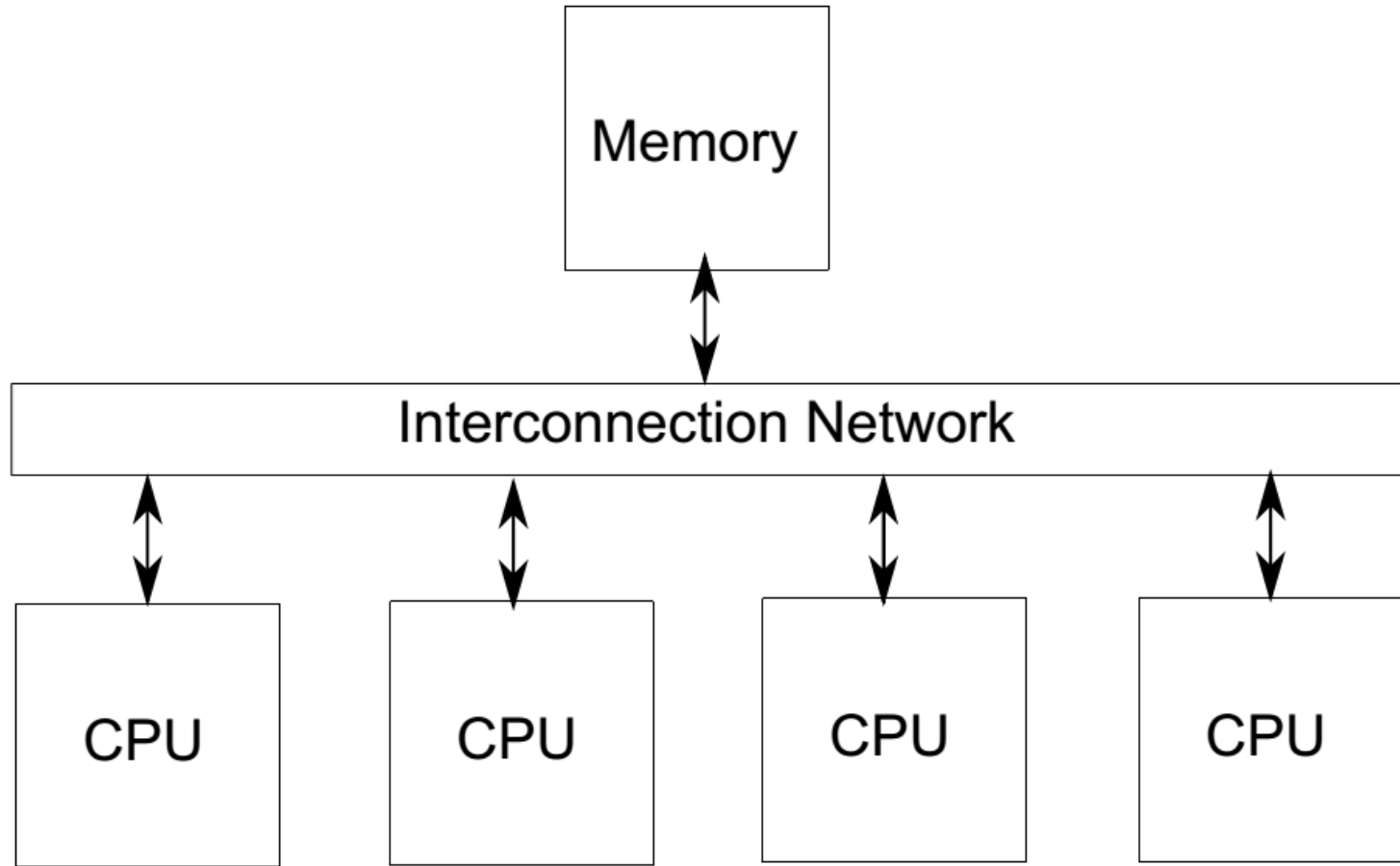
SOURCE: RAY KURZWEIL, "THE SINGULARITY IS NEAR: WHEN HUMANS TRANSCEND BIOLOGY", P.67, THE VIKING PRESS, 2006. DATAPOINTS BETWEEN 2000 AND 2012 REPRESENT BCA ESTIMATES.



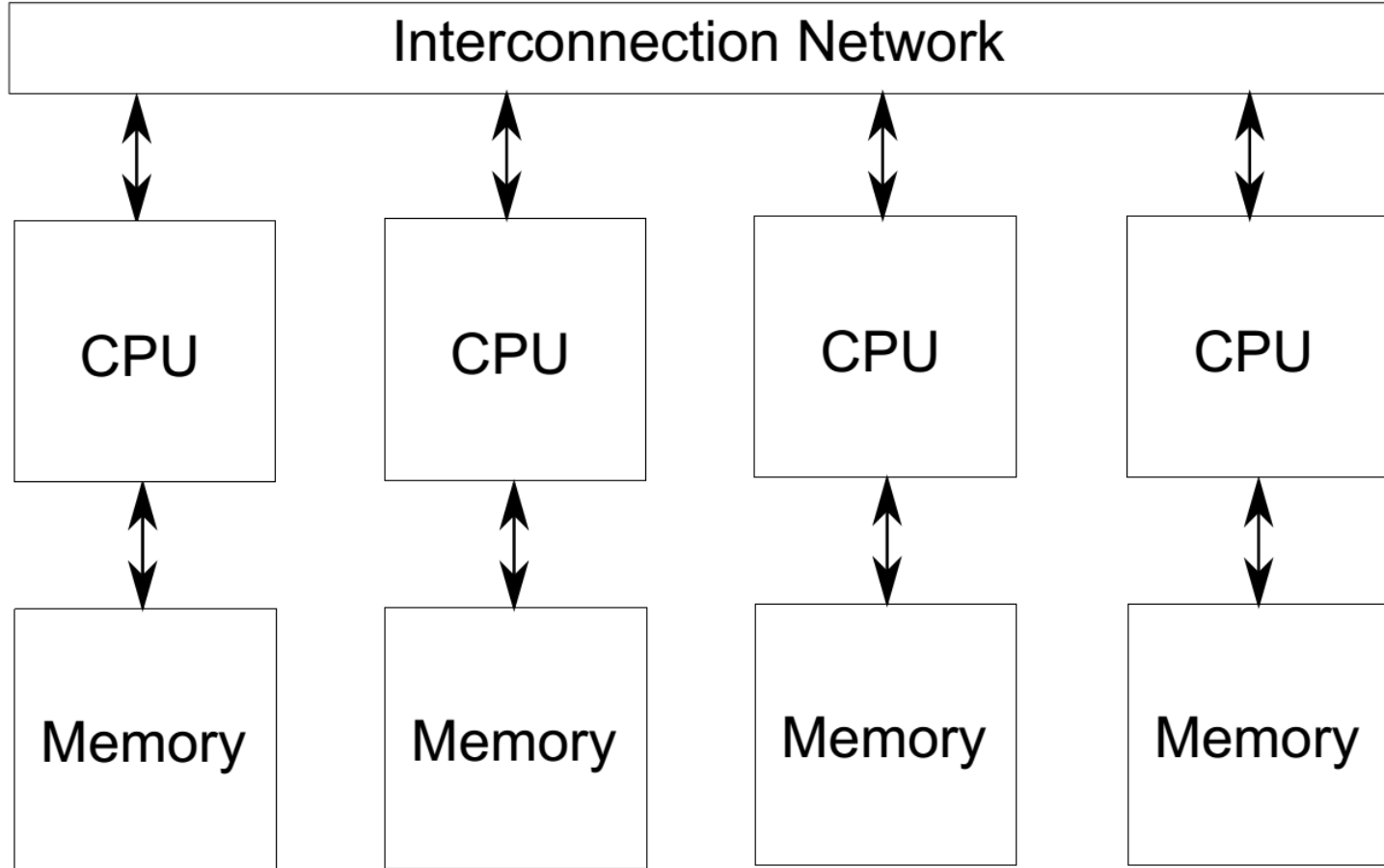
Procesadores *multicore* entregan un mecanismo para tener multiprocesamiento en un solo chip



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

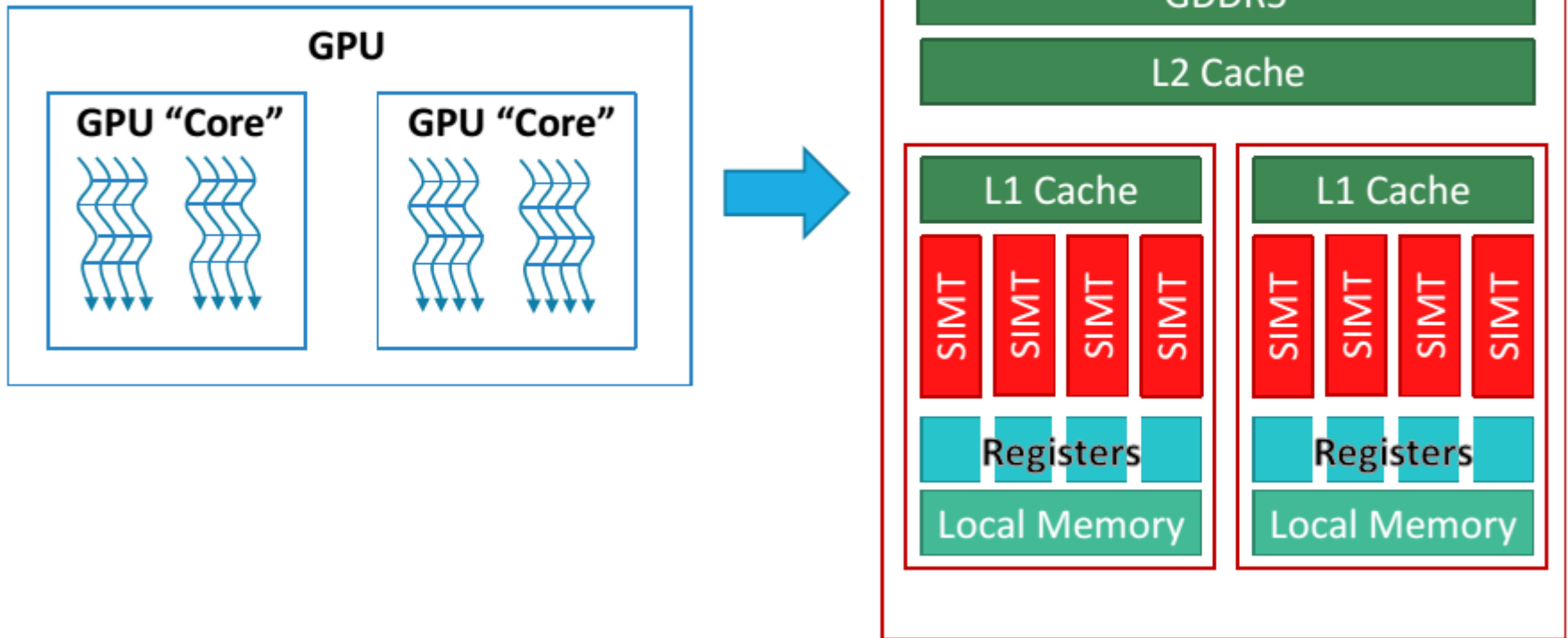


Multiprocesador por paso de mensajes  
puede ser un **cluster** o un **sistema distribuido**





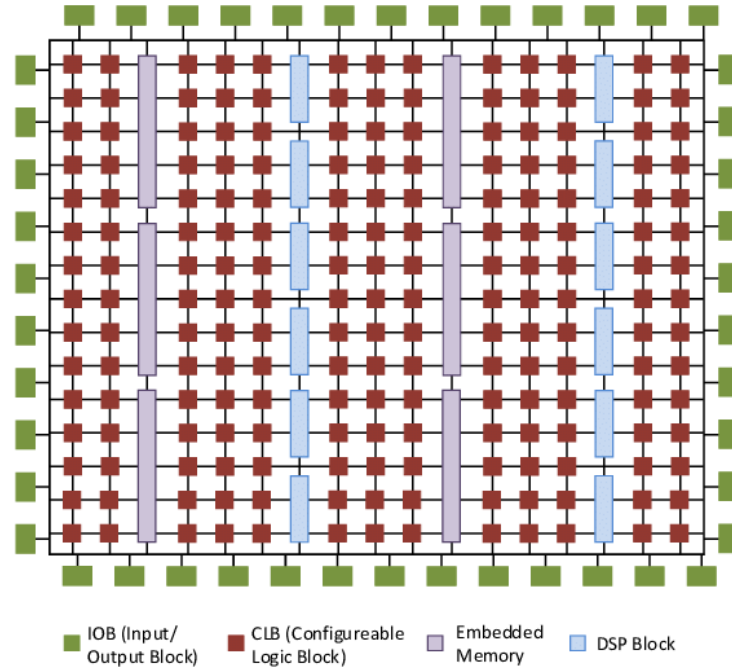
# GPUs contienen cientos de pequeños procesadores



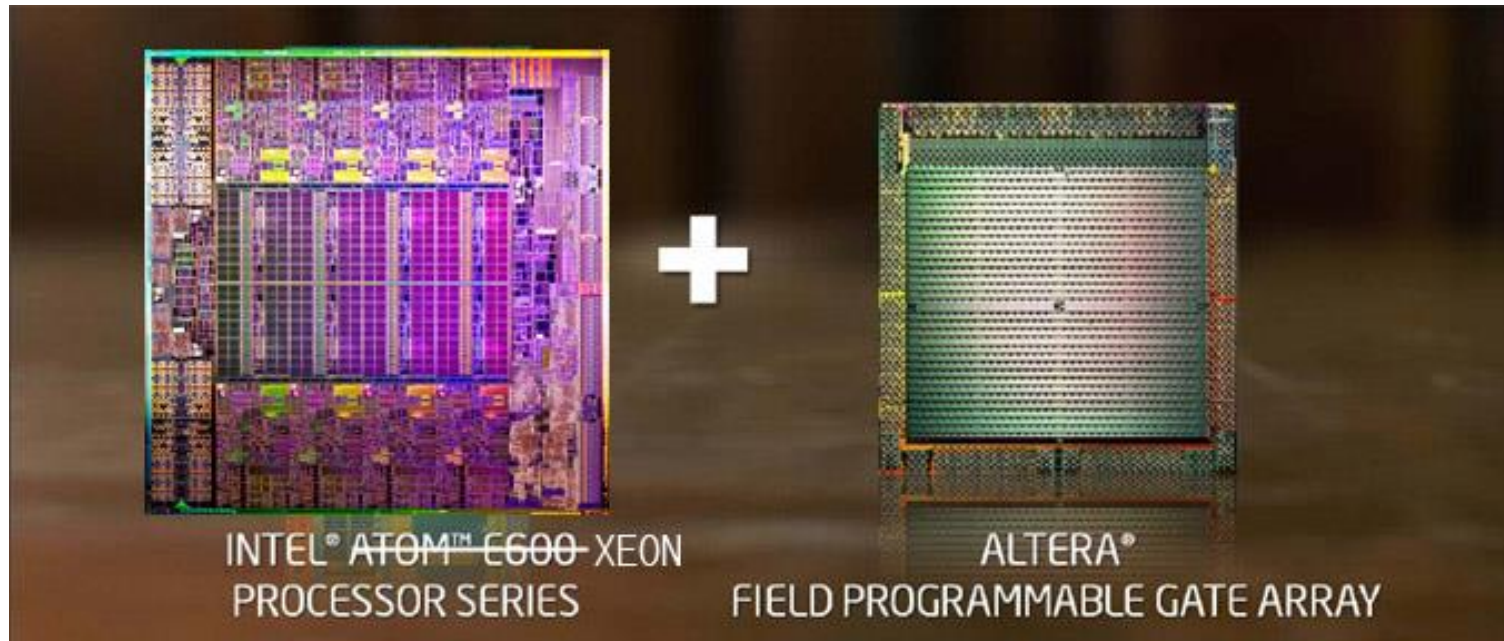
Para terminar, una pequeña muestra de la dimensión de la industria de las arquitectura de computadores



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

