

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-Albert Löbel

¿DE QUÉ ME SIRVE APRENDER ESTO?

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

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

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

¿QUÉ ES UN COMPUTADOR?



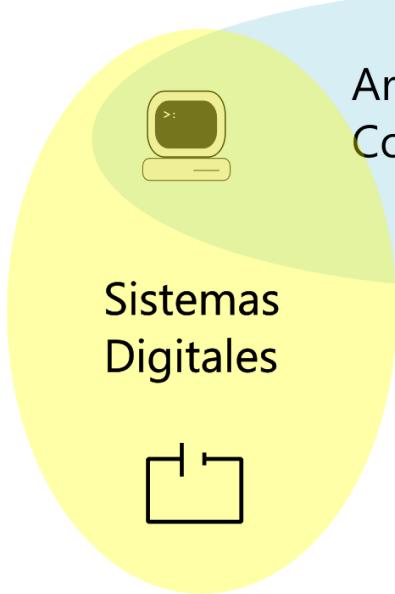
Muchas posibles respuestas

- Lo más importante: es una **máquina programable**.
- Ejecuta programas (secuencia de instrucciones).
- La pregunta que no interesa ahora (y en el curso de arquitectura) es la siguiente:

¿cómo construimos una máquina
programable que ejecuta programas?



usuario



Arquitectura de
Computadores

```
010110  
101100  
111101
```

Sistemas Operativos

```
010110  
101100  
111101
```

```
010110  
101100  
111101
```

Compiladores

```
class A  
{  
    int a;  
}
```

Introducción a
la Programación

```
class A  
{  
    int a;  
}
```

Programación
Avanzada

```
class A  
{  
    int a;  
}
```



desarrollador

De qué trata esta clase

- Empezaremos 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.

Veamos un poco de historia

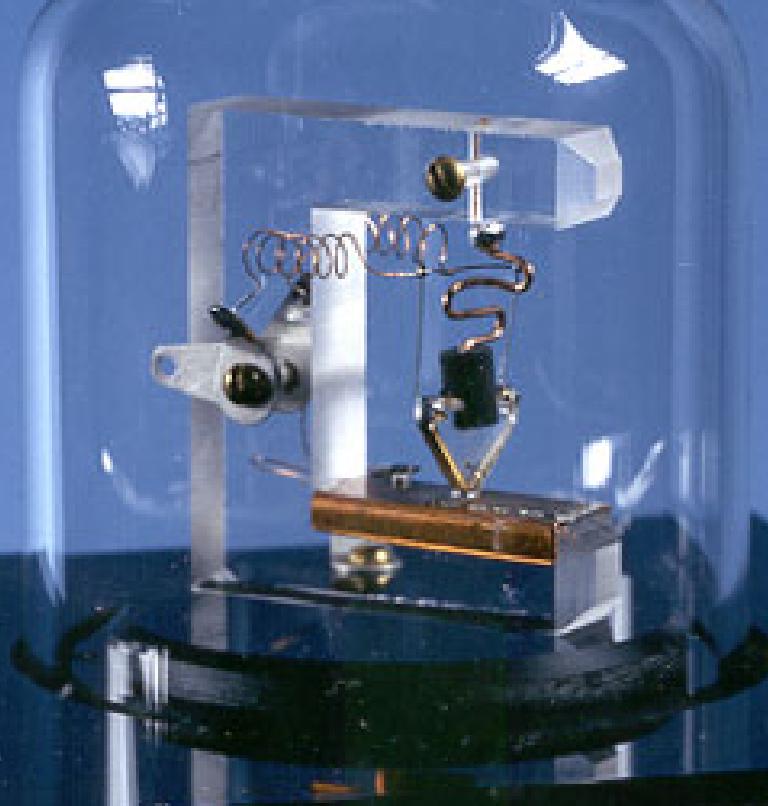


GENERACIÓN 0 (≈ 1830)



GENERACIÓN 1 (\approx 1900)

GENERACIÓN 2 (1947)

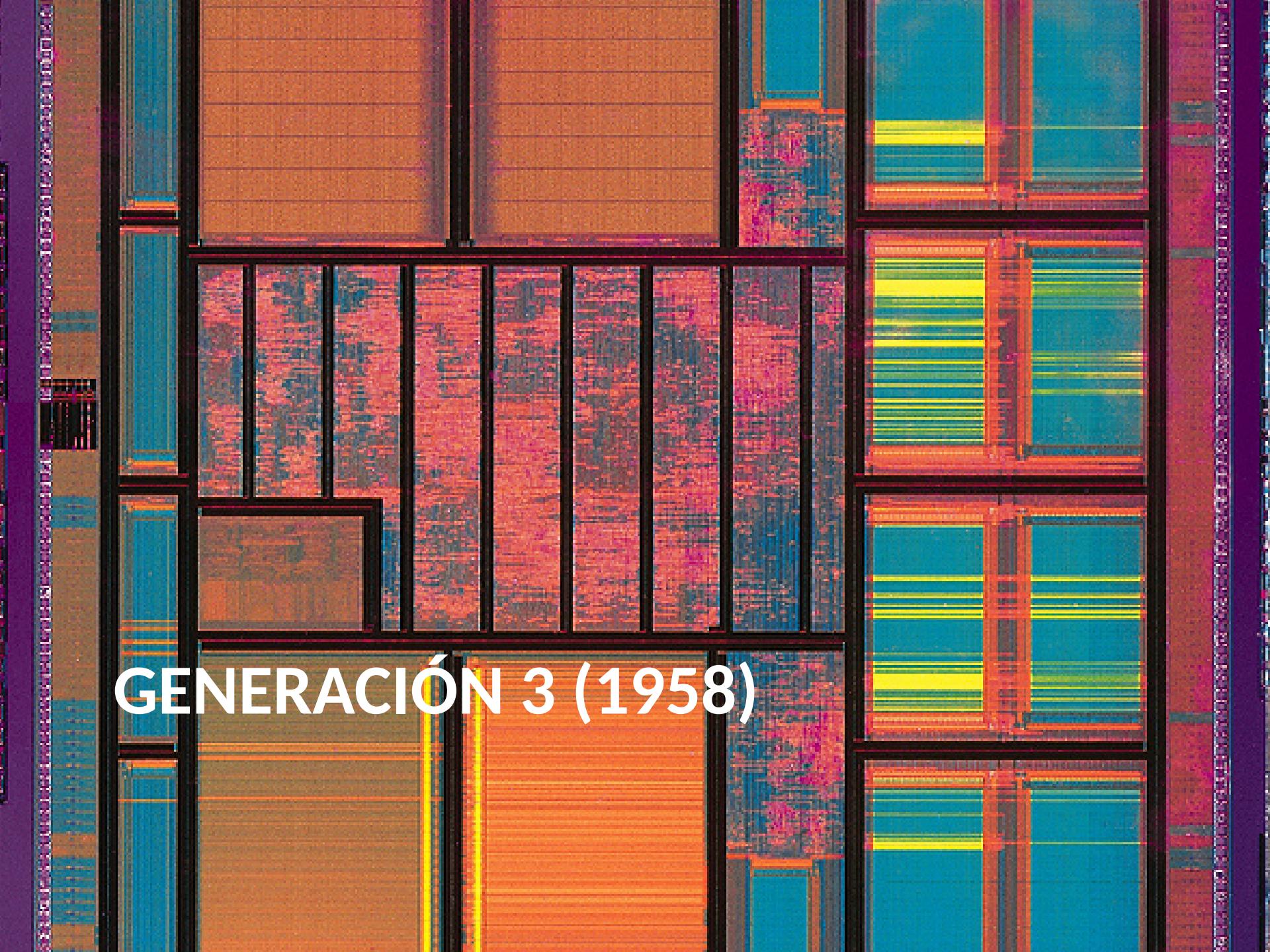


microelectronics group

Lucent Technologies
Bell Labs Innovations

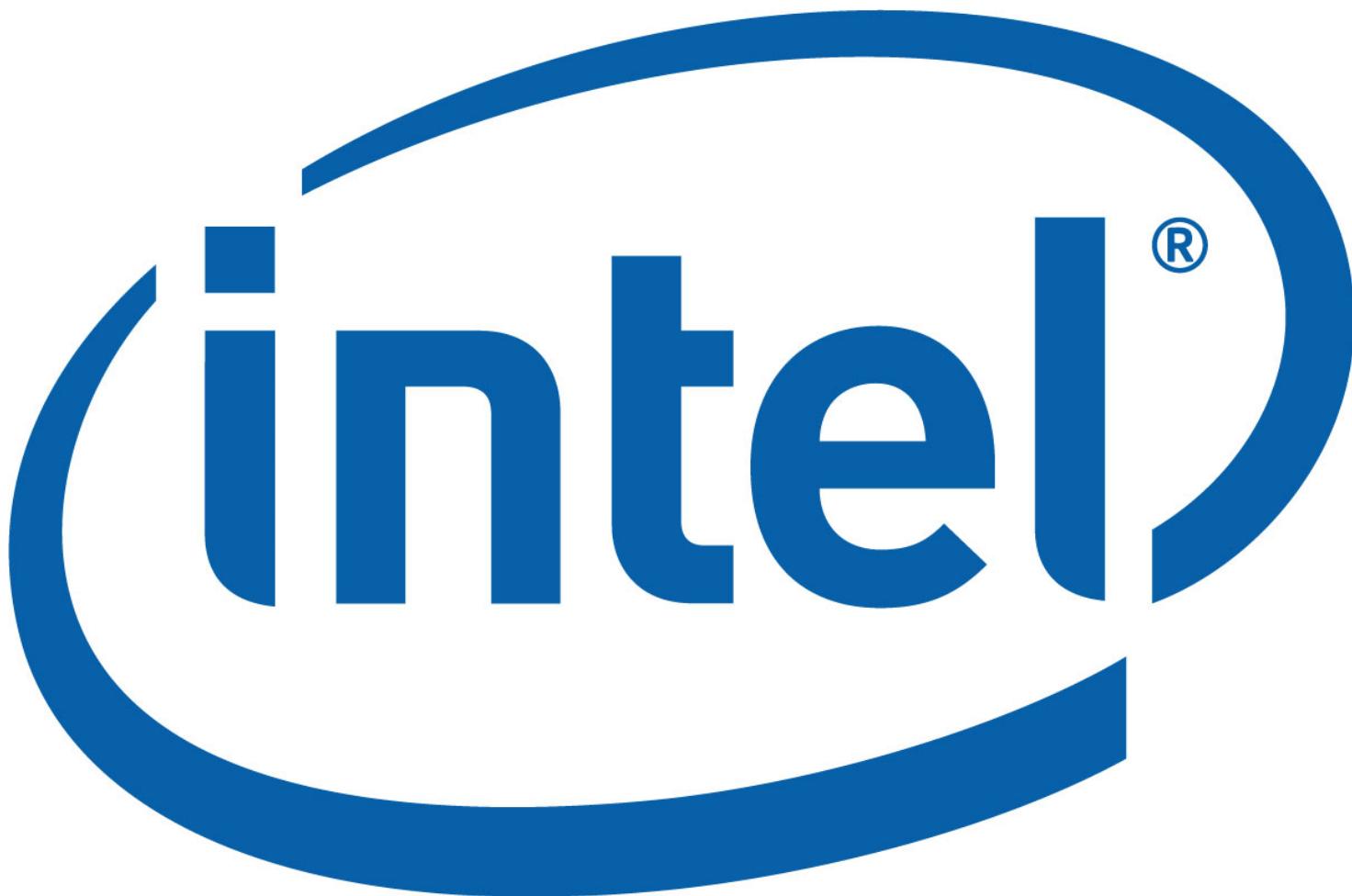
A replica of the first transistor,
invented at Bell Labs,
December 23, 1947

50 Years and Counting...



GENERACIÓN 3 (1958)

Intel, 1968

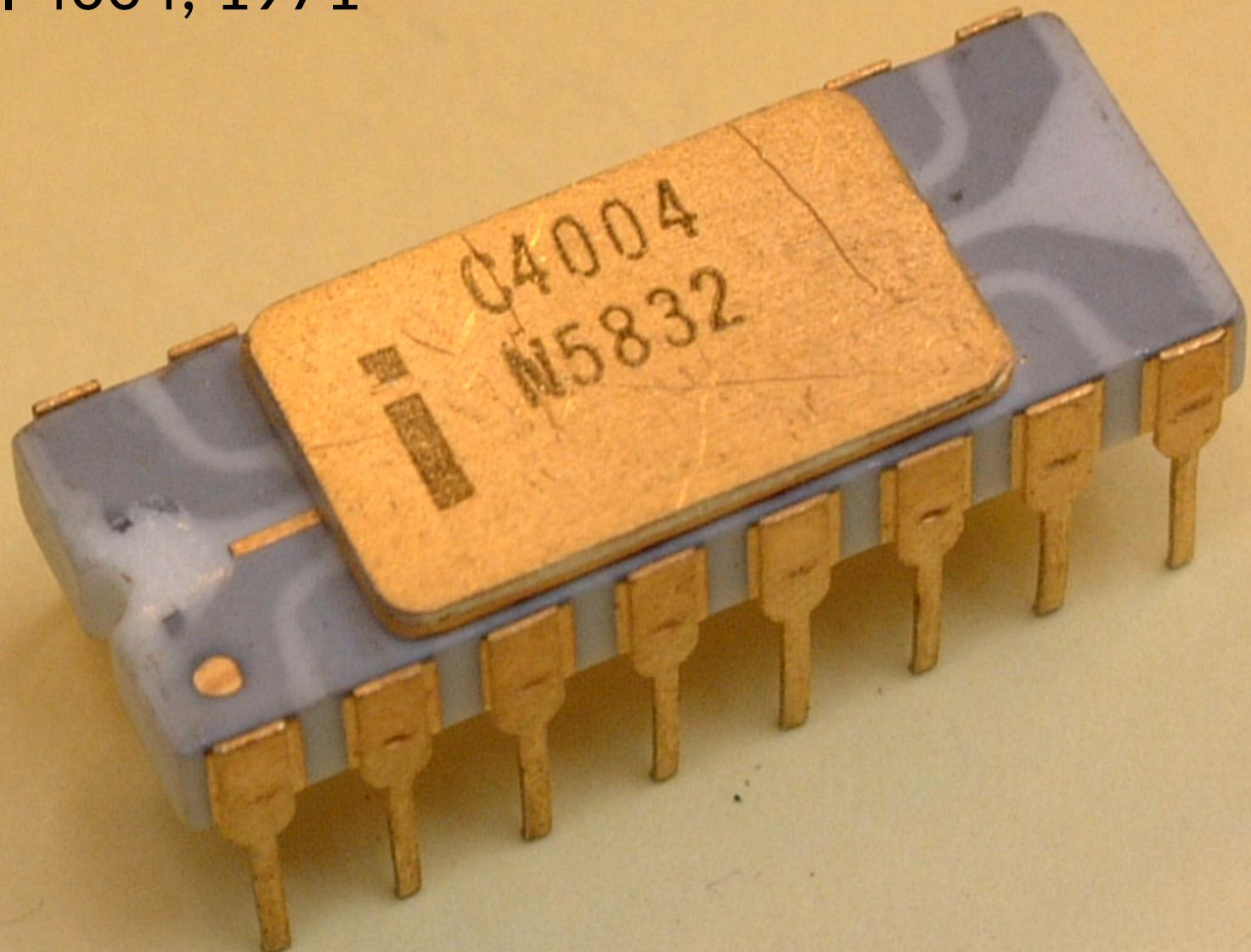




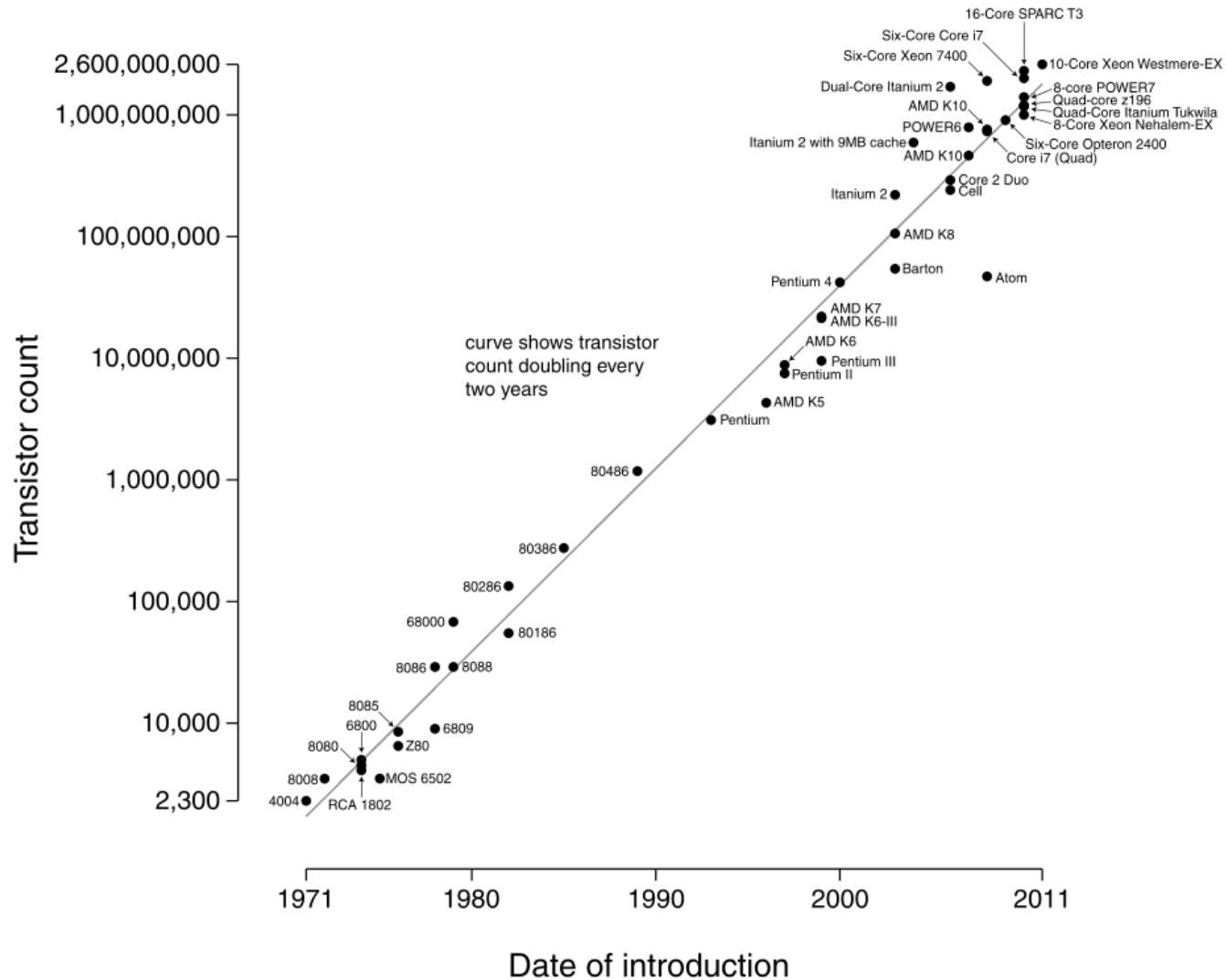
Gordon
Moore

Robert
Noyce

Intel 4004, 1971

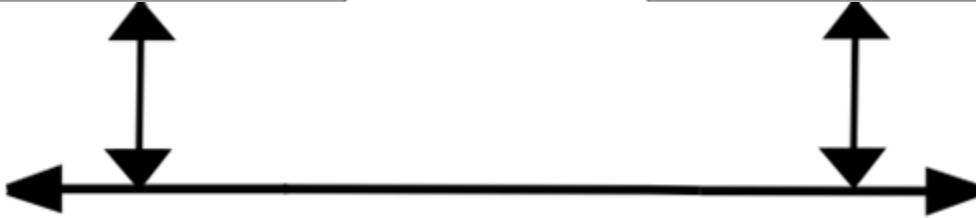


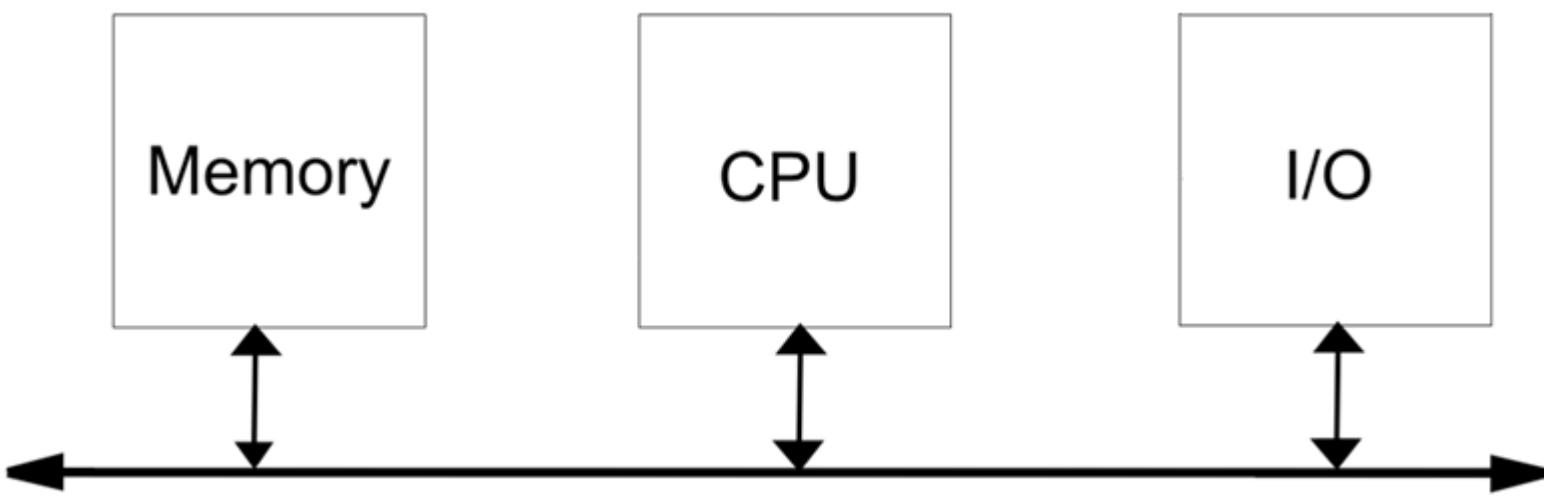
Microprocessor Transistor Counts 1971-2011 & Moore's Law



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





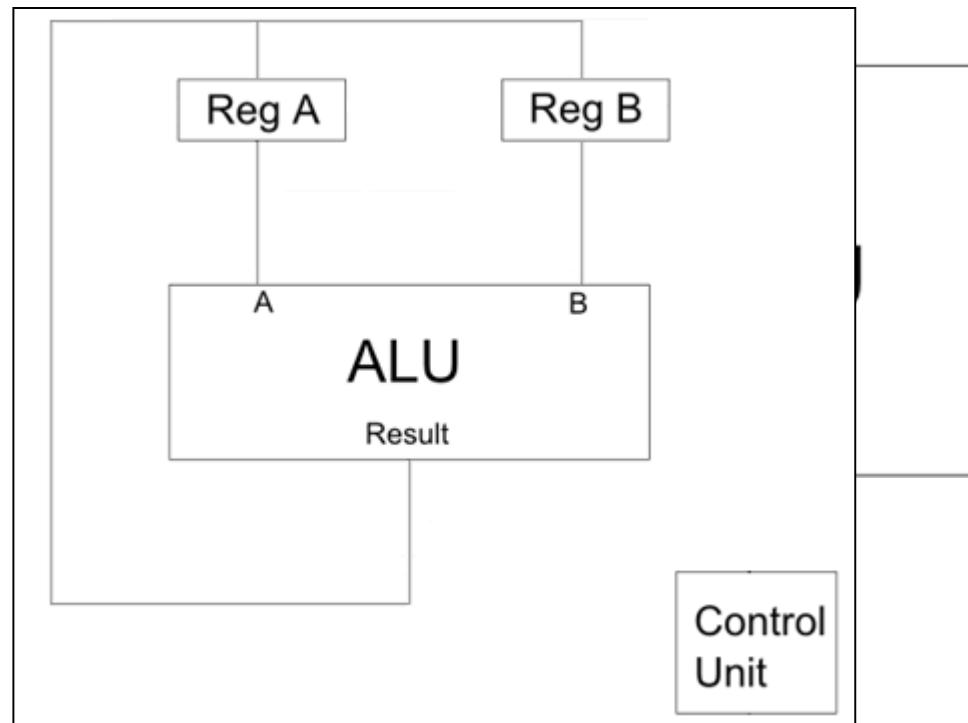
Muy bien por el hardware, pero ¿qué pasa con las instrucciones?

- 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 por la unidad de control, que decide que elementos deben actuar.
- Cada posible instrucción del computador está asociada a **sólo una** secuencia de 0s y 1s.

Memory

CPU





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**)

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.

Microarquitectura e ISA definen la arquitectura de un computador

La arquitectura de un computador caracteriza en base a dos elementos:

- Microarquitectura: se refiere a los distintos componentes de hardware que están presentes en el computador.
- Arquitectura del set de instrucciones (ISA): se refiere al tipo, formato, características, etc., de las instrucciones soportadas por el computador.

Arquitecturas de von Neumann y Harvard se utilizan en distintos casos

La memoria presenta una división entre **2 grandes paradigmas** dentro de la arquitectura de los computadores:

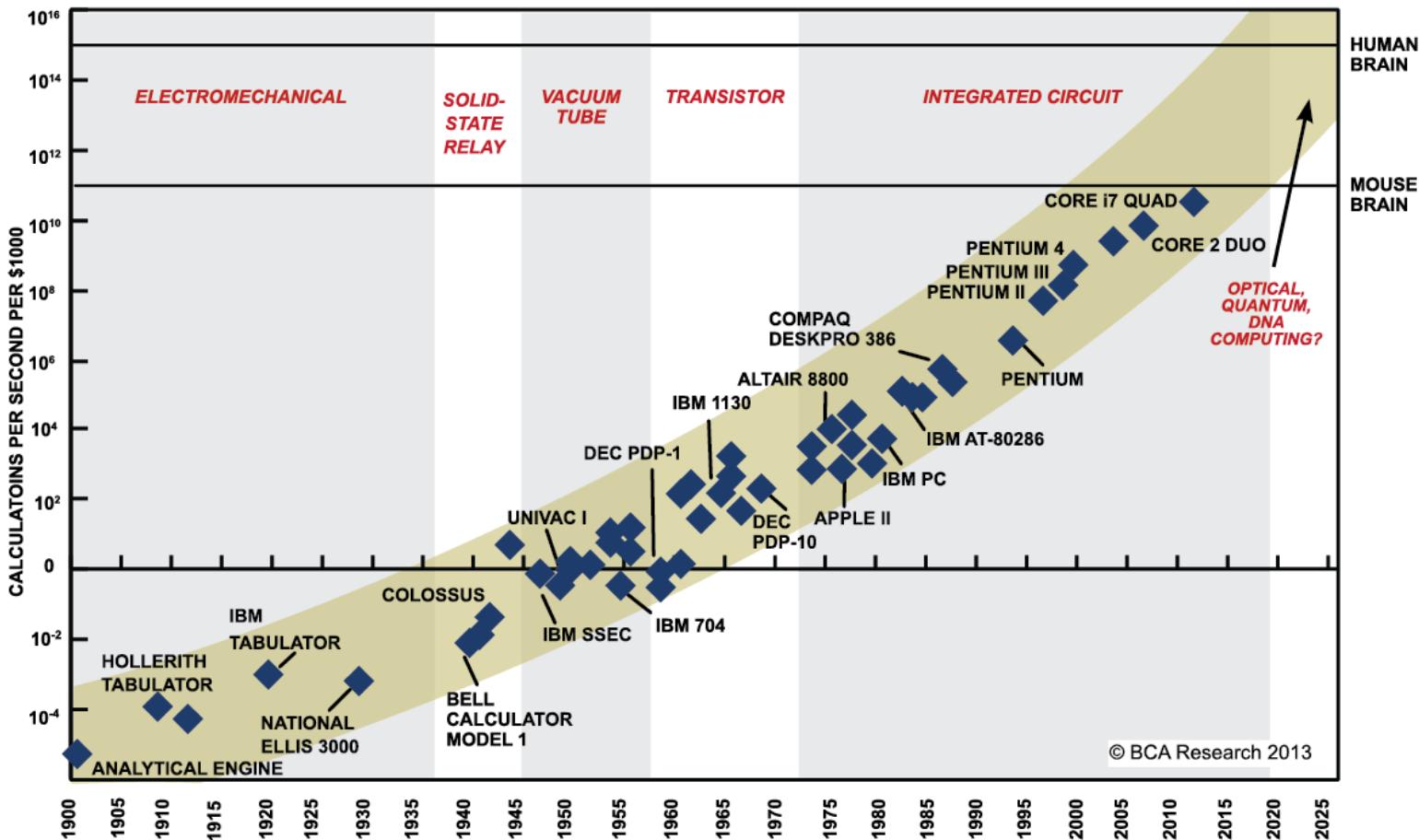
- **Arquitectura Harvard:** presenta memorias independientes para instrucciones y para datos.
- **Arquitectura von Neumann:** utiliza una sola memoria compartida entre instrucciones y datos. Permite **AUTOPROGRAMABILIDAD**.

RISC y CISC presentan soluciones con distinto foco para un mismo problema

Implementación de ISA responde generalmente a uno de dos paradigmas:

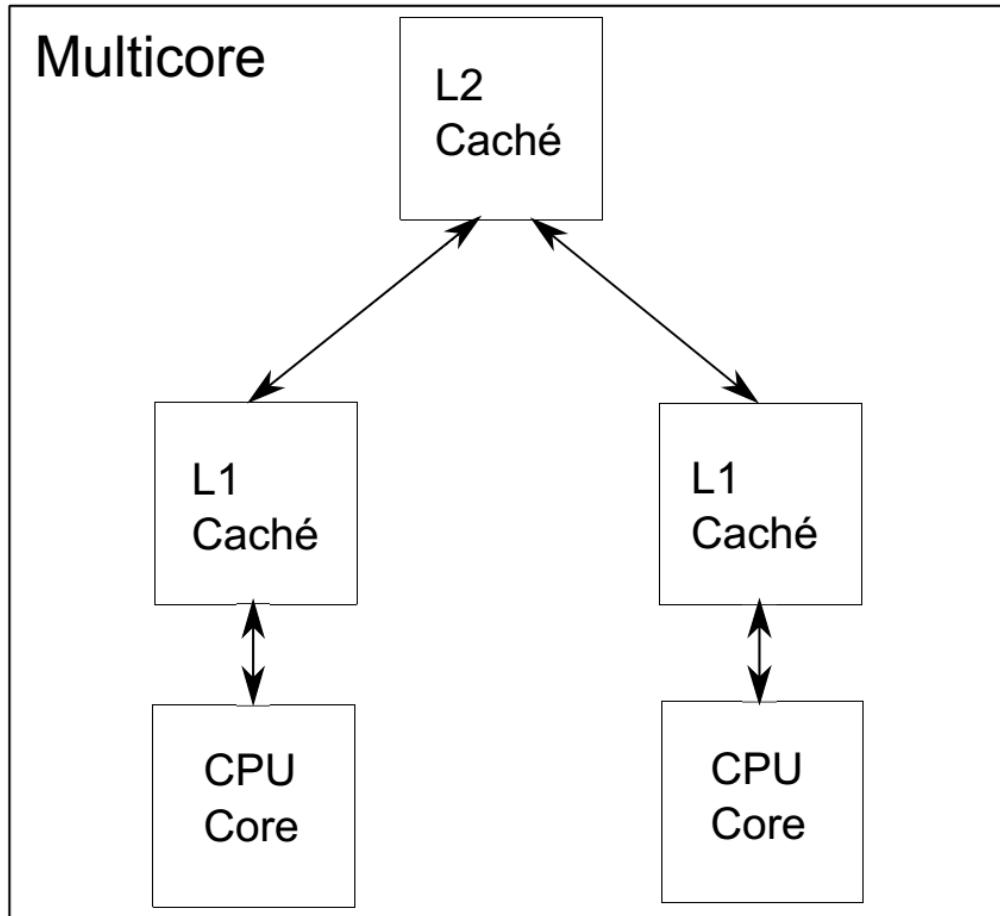
- **RISC:** Instrucciones pequeñas y simples. Diseñado para minimizar complejidad del hardware. Énfasis en el software.
- **CISC:** Muchas instrucciones y de alta complejidad. Énfasis en el hardware.

El futuro y la importancia de las arquitecturas avanzadas

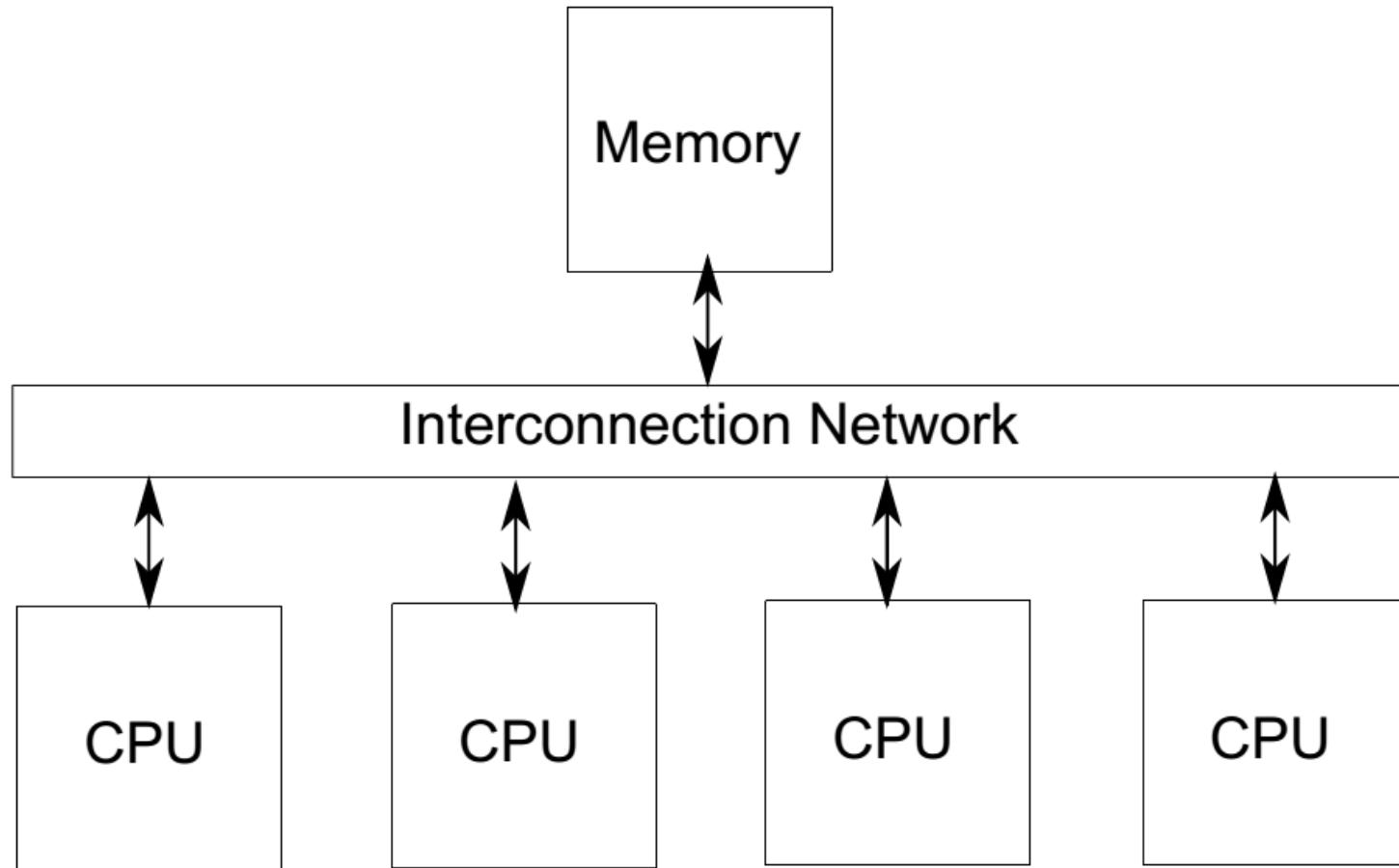


SOURCE: RAY KURZWEIL, "THE SINGULARITY IS NEAR: WHEN HUMANS TRANSCEND BIOLOGY", P.67, THE VIKING PRESS, 2006. DATAPoints BETWEEN 2000 AND 2012 REPRESENT BCA ESTIMATES.

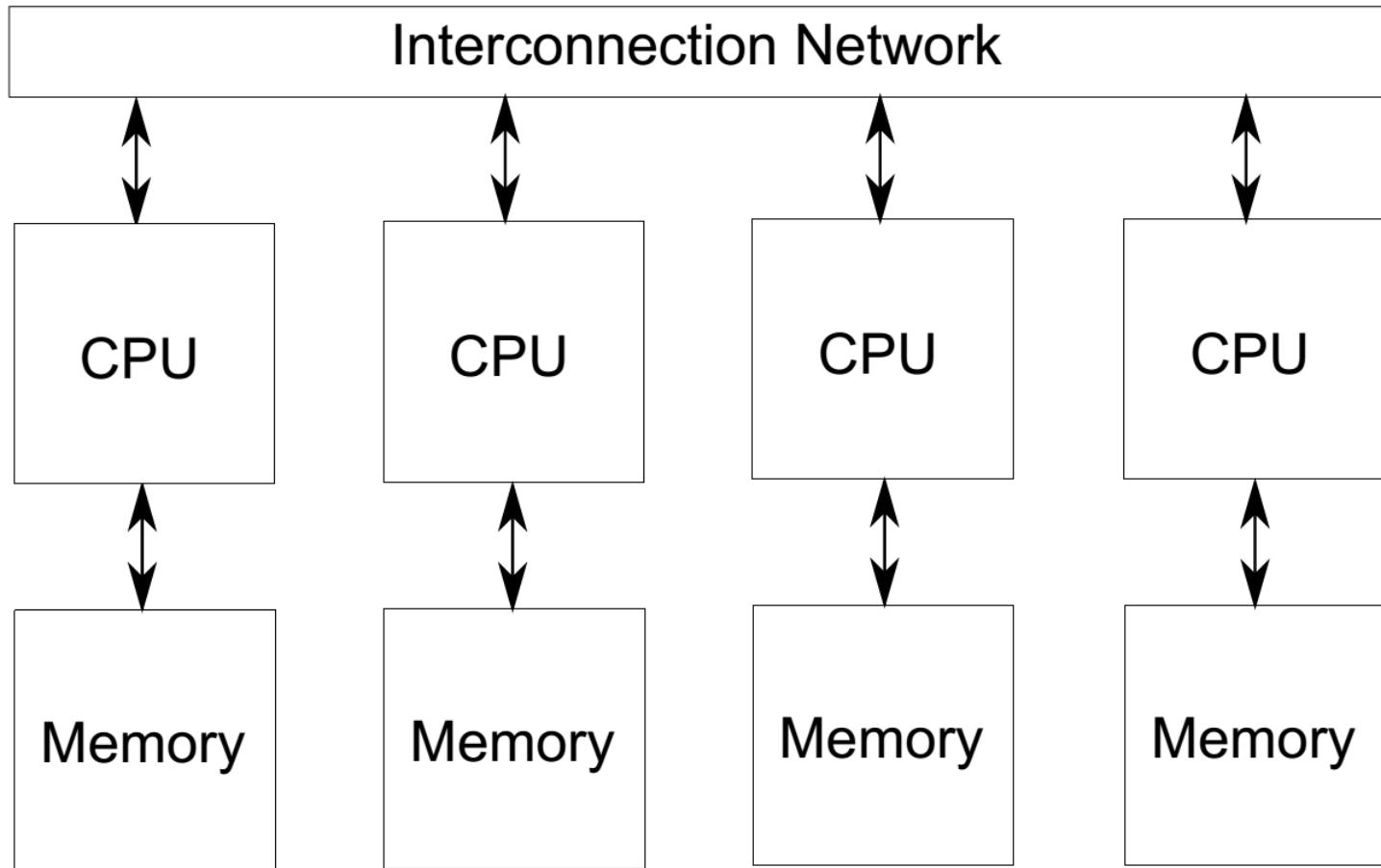
Procesadores *multicore* entregan un mecanismo para tener multiprocesamiento en un solo chip



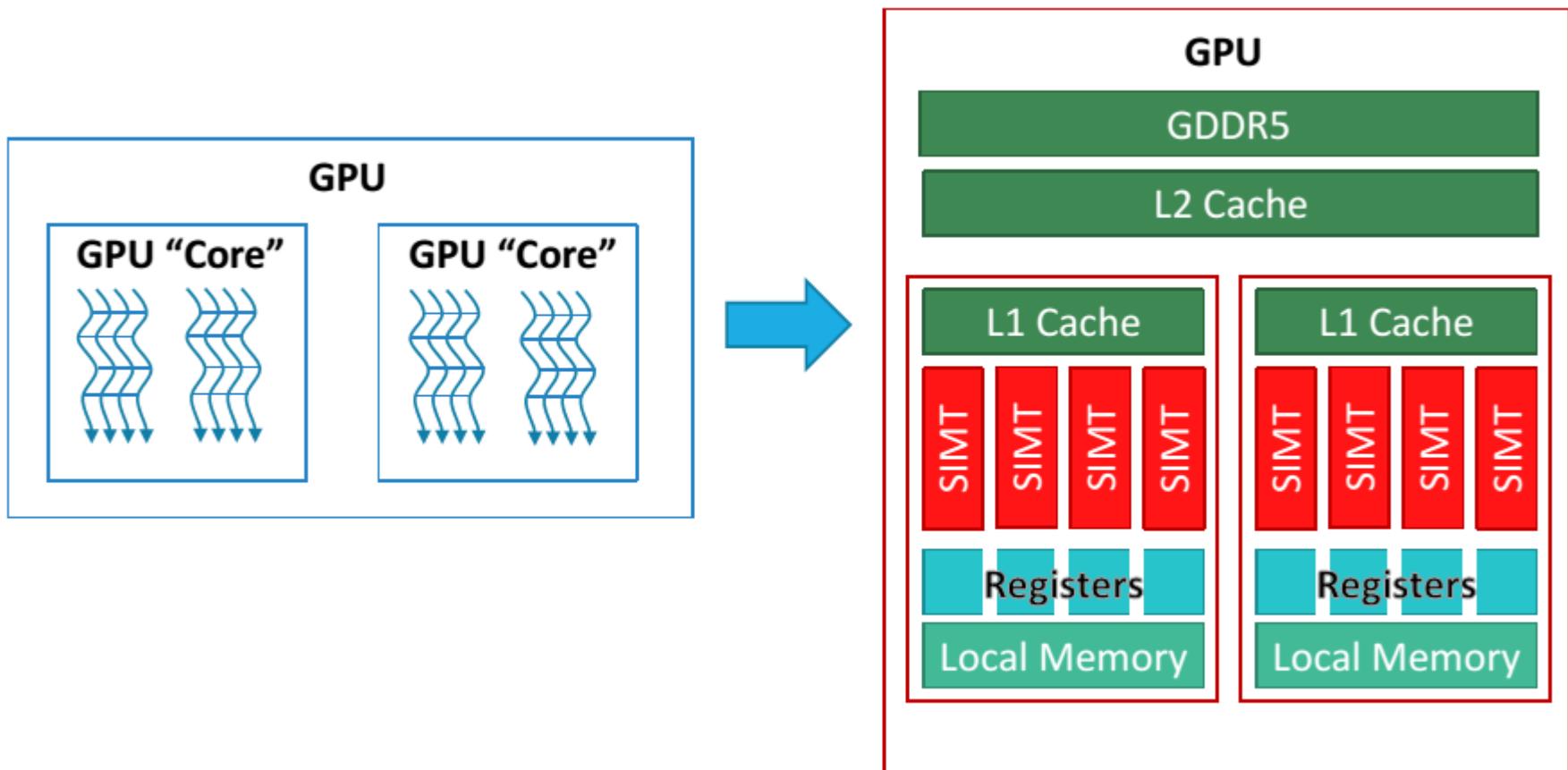
Multiprocesador con memoria compartida necesita mecanismos para mantener coherencia en memoria



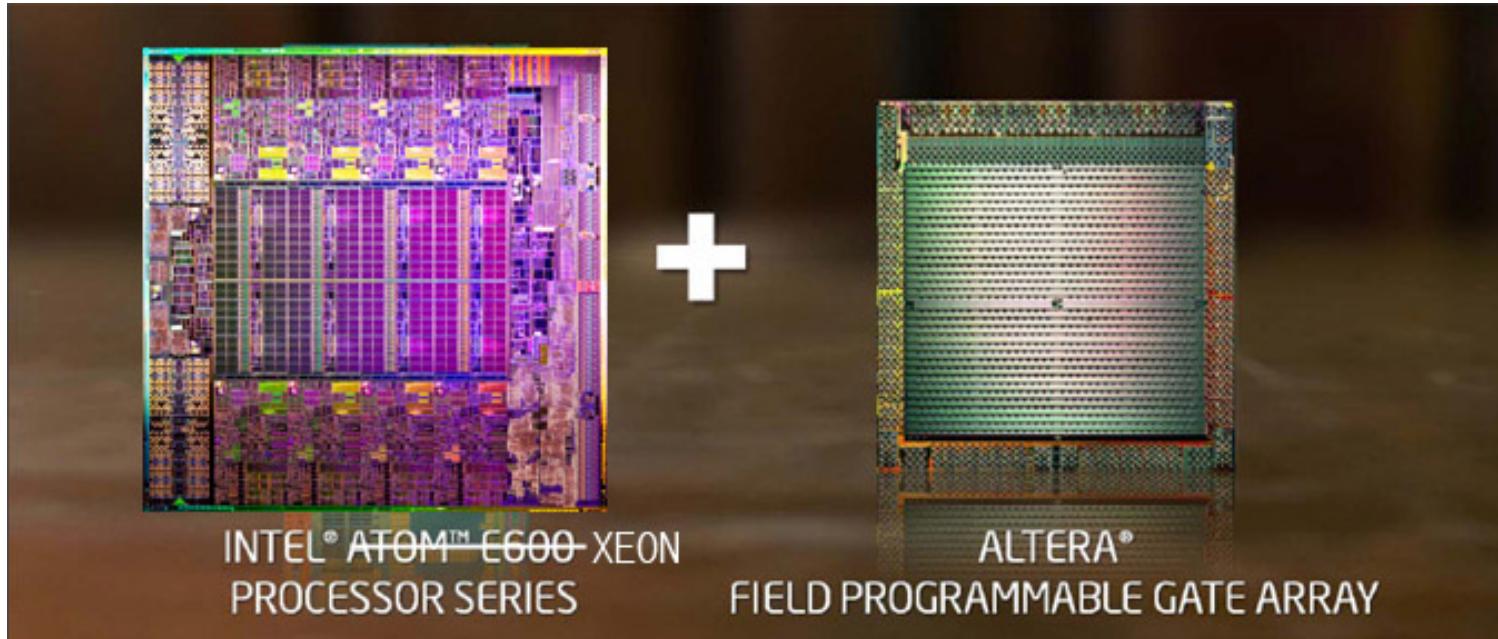
Multiprocesador por paso de mensajes
puede ser un **cluster** o un **sistema distribuido**



GPUs contienen cientos de pequeños procesadores



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-Albert Löbel