

Arquitectura e Ingeniería de Computadores

valentin.puente@unican.es

pablo.abad@unican.es

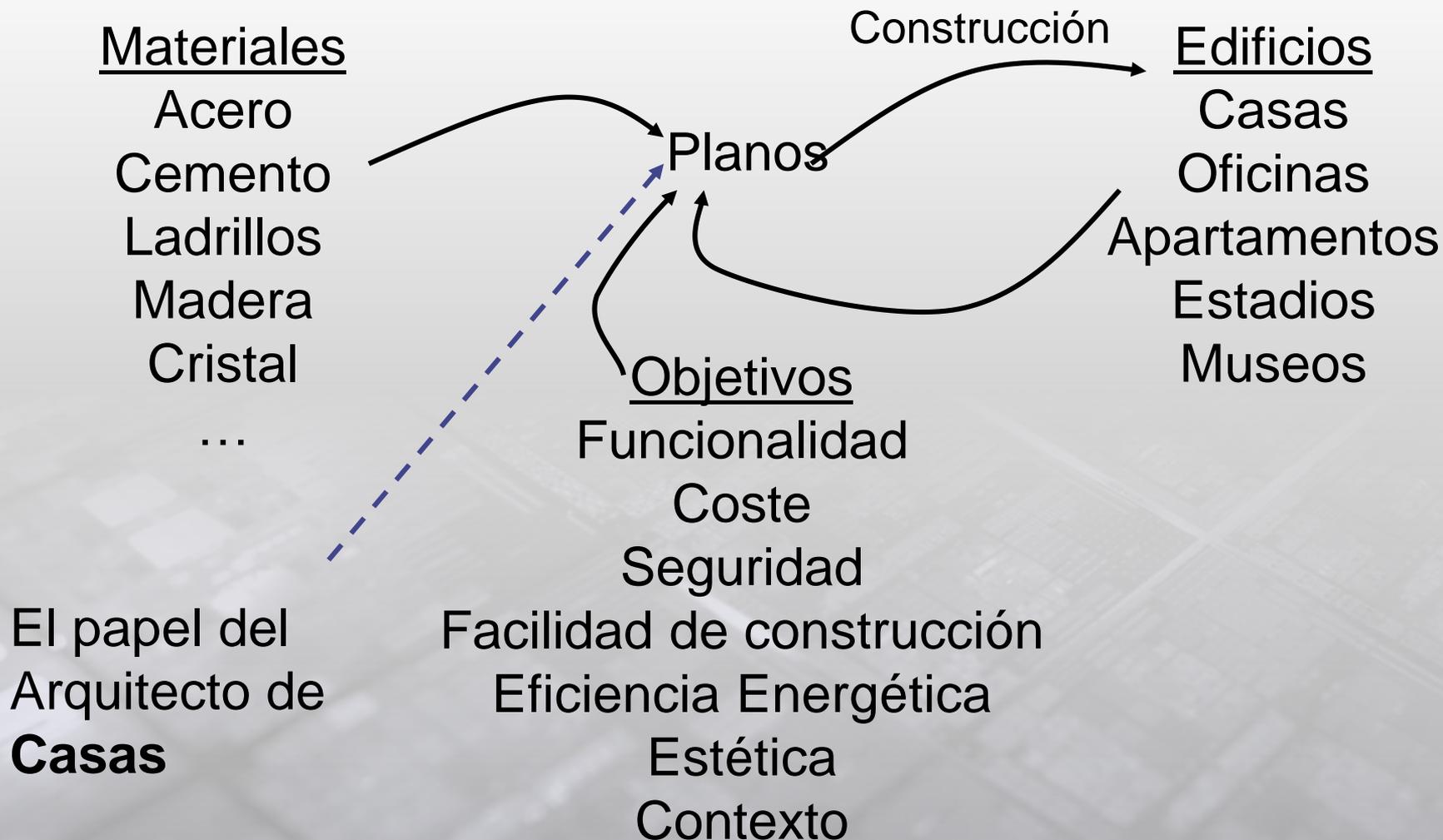
Curso 1011

Revisión 10/02/2011

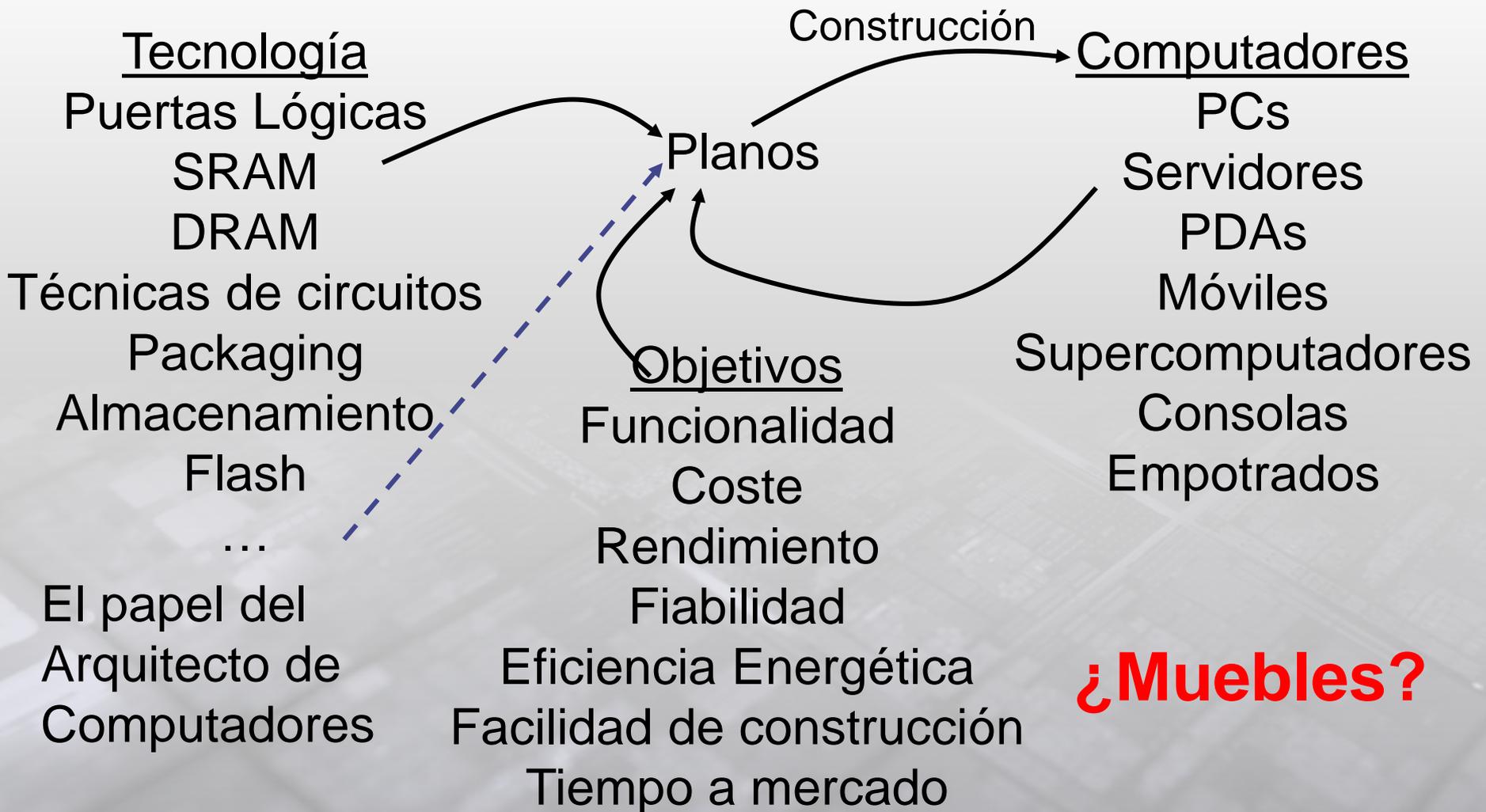
¿Qué es Arquitectura de Computadores?

- “*La arquitectura de Computadores es la ciencia y arte de seleccionar y interconectar componentes hardware para crear computadores que cumplan determinados objetivos funcionales, de rendimiento o coste*” – WWW Computer Architecture Page
- Veamos una analogía con la arquitectura “normal”...

¿Qué es Arquitectura de Casas?



¿Qué es Arquitectura de Computadores?



Tres importantes diferencias

- Bagaje histórico:
 - ~60 años frente a milenios
- Procesos de fabricación en **masa**:
 - magnífica efectos del diseño
- Ritmo de cambio:
 - constante y **muy** rápido

Objetivos de Diseño

- **Funcional**

- Necesita ser correcto
- ¿Que funcionalidad debe incluir?

- **Fiable**

- ¿Durante cuanto tiempo seguirá operando correctamente?
- Fallos permanentes vs fallos transitorios

- **Alto rendimiento**

- No es sólo cuestión “Gigaherzios”
- Objetivo imposible: el más rápido posible en todos los programas posibles

Objetivos de Diseño

- **Bajo coste**

- Coste de fabricación por unidad (coste de la oblea de silicio)
- Coste de hacer el primer chip (coste de las máscaras)
- Coste del diseño (Equipos de diseño muy grandes, por que? ...)

- **Bajo consumo energético**

- Energía de entrada (tiempo de uso con batería, coste de la electricidad)
- Energía de salida (enfriamiento y costes relacionados)
- Estática vs dinámica, sleep, pico vs promedio
- Problema cíclico, (muy serio hoy en día)

- **Reto: balancear la importancia relativa de todos estos objetivos**

- Y el balance esta constantemente cambiando

Cambio constante



- Mejores computadores ayudan a diseñar mejores computadores en la siguiente generación
- +apps

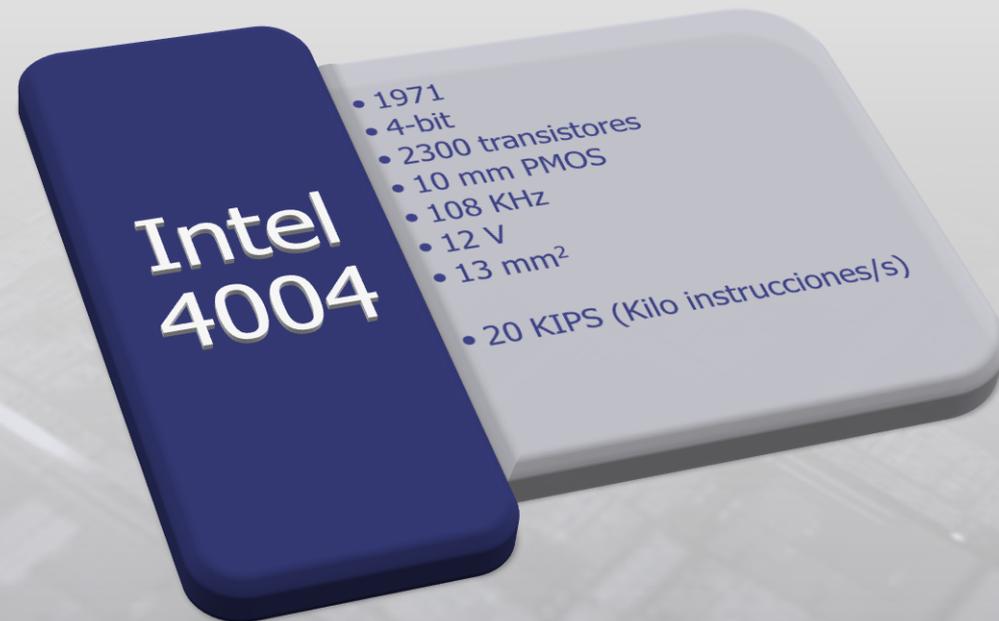
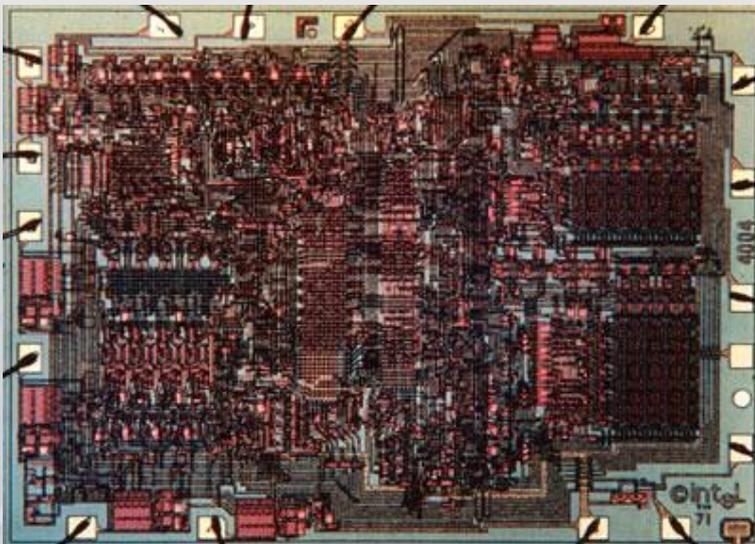
Cambio rápido

	1971– 1980	1981– 1990	1991– 2000	2004	2010
Transistores (Millones)	0.01–0.1	0.1–1	1–100	250	2000
Clock (MHz)	0.2–2	2–20	20–1000	3500	5000
MIPS	<0.2	0.2–20	20–2000	7000	500000

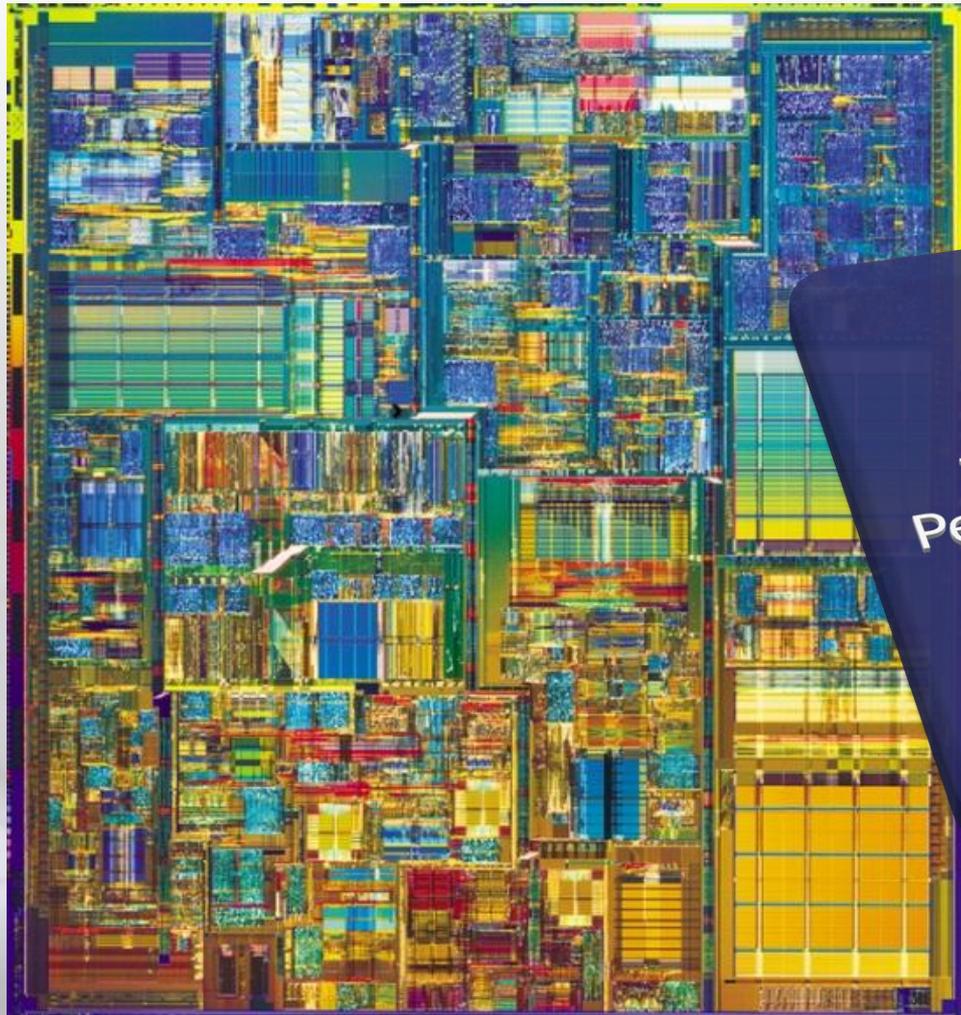
- Quizás el campo del conocimiento humano con más rápida evolución ...
 - Procesadores vs. coches
 - 1985: procesadores = 1 MIPS, coches = 100 km/h
 - 2000: procesadores = 500 MIPS, coches = 50,000 km/h?
- Otro campo excitante? Biotecnología
 - Curiosamente, muchos de los avances biotecnológicos están posibilitados por los computadores

Primer microprocesador

- Conectaron unos pocos transistores juntos para hacer...



Algo más reciente



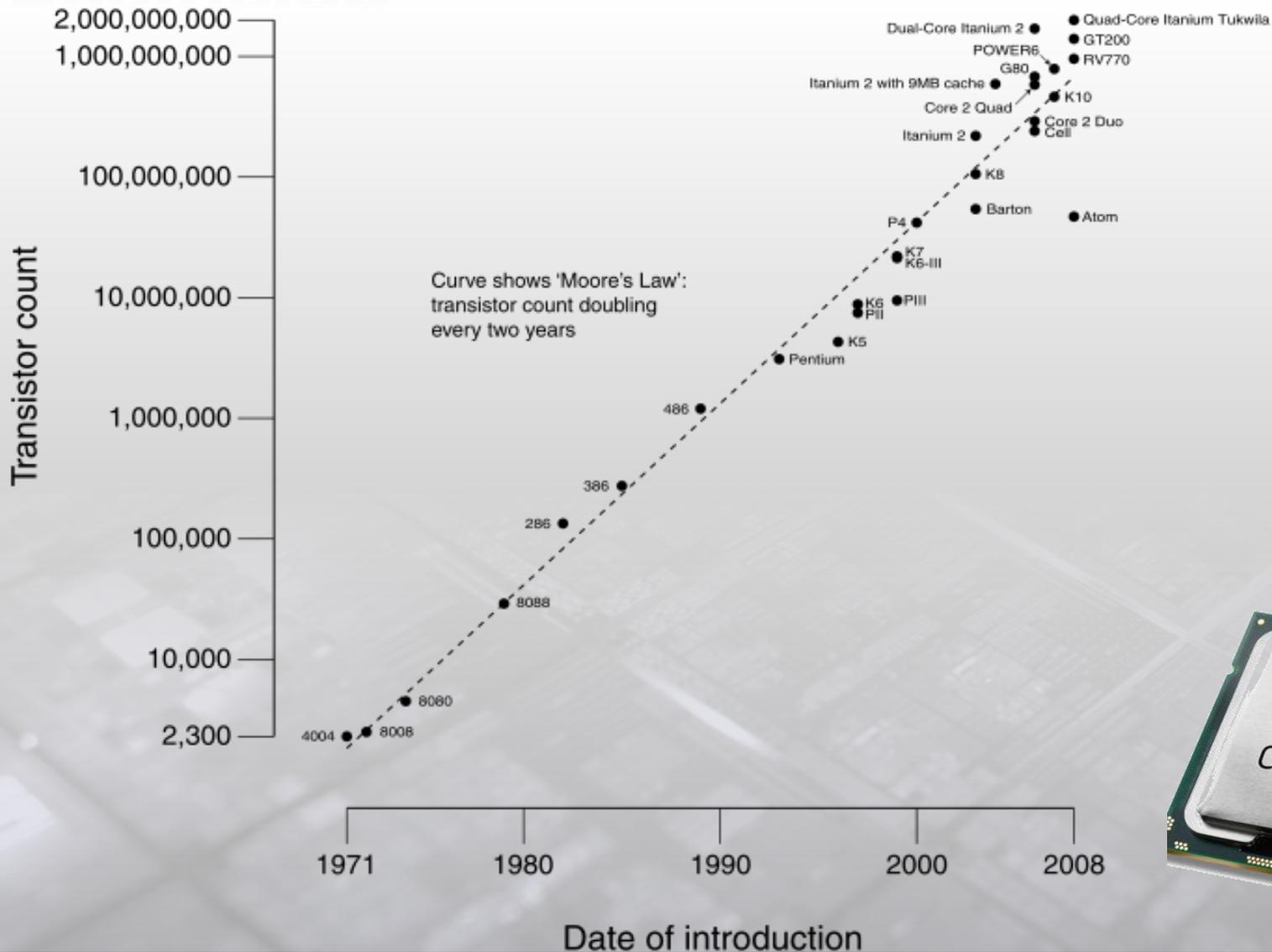
Intel
Pentium4

- 2003
- 32/64-bit data
- 55M transistores
- 0.90 nm CMOS
- 3.4 GHz
- 1.2 V
- 101 mm²
- 8000 MIPS

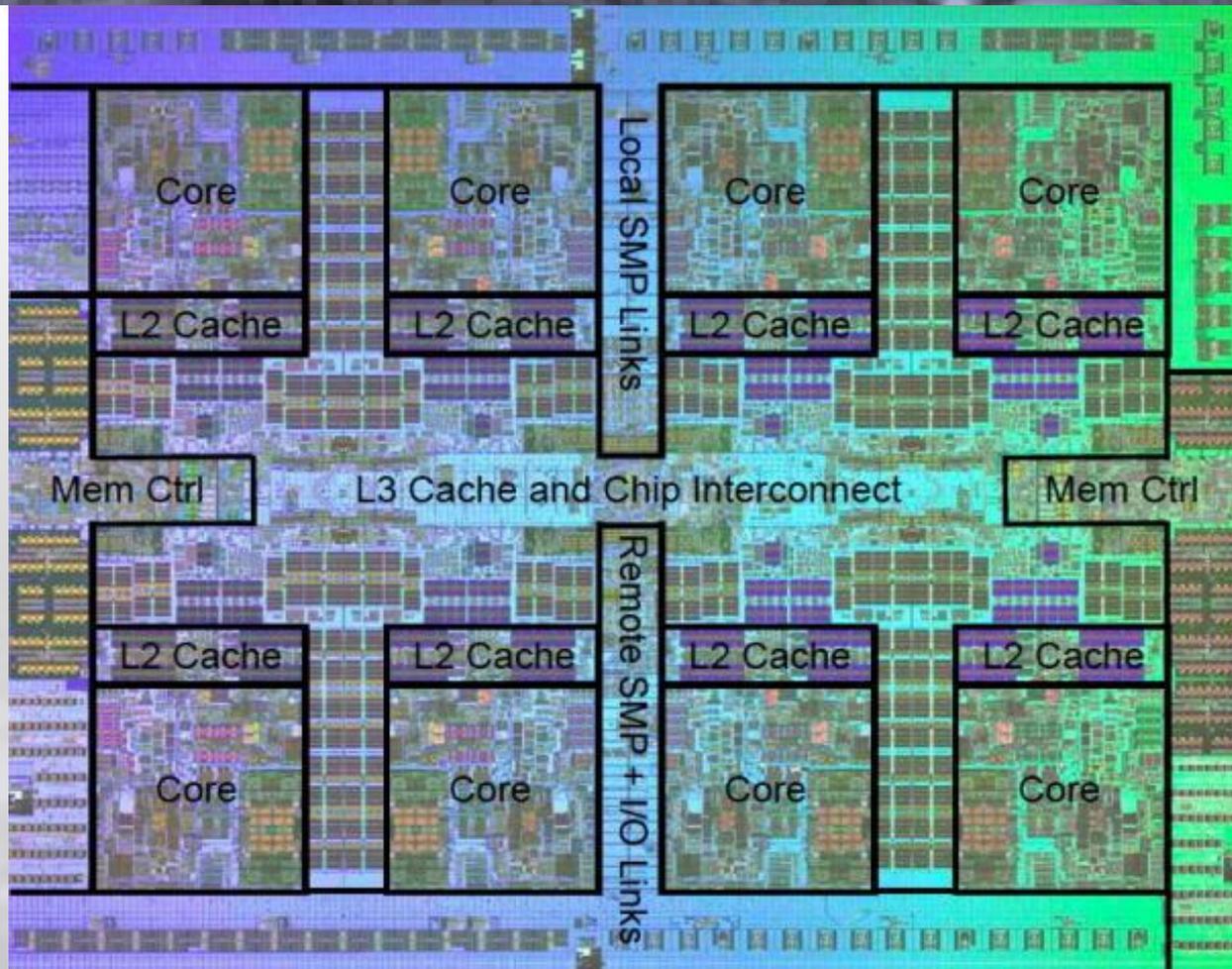
Al final del curso debemos saber...

- **... qué diablos significa todo esto:**
- Especificaciones del Pentium 4:
 - Tecnología
 - 55M transistores, 0.90 μm CMOS, 101 mm², 3.4 GHz, 1.2 V
 - Rendimiento
 - 1705 SPECint, 2200 SPECfp
 - ISA
 - X86+MMX/SSE/SSE2/SSE3 (X86 internamente traducido a uops RISC)
 - Jerarquía de memoria
 - 64KB 2-way cache de instrucciones, 16KB D\$, 512KB–2MB L2
 - Controlador de coherencia usando MESI, modelo de consistencia PC
 - Segmentación
 - 22 etapas, superescalar 4-way, con planificación dinámica
 - 1K BTB, 8Kb DP, 16 entradas RAS
 - 2-way SMT
- Repetir con Core 2 Duo, Core 7i, etc...

Evolución



Formul1 procesadores (IBM Power7)



•1200M transistores, 0.045 μm (45nm)
CMOS, 567mm², 3-4.14GHz,
~300GFLOPS, ~1024 GIPS

Tendencias Tecnológicas

- Procesador (SRAM)
 - Densidad: ~30%, Velocidad: ~20%
- Memoria (DRAM)
 - Densidad : ~60%, Velocidad : ~4%
- Disco
 - Densidad : ~60%, Velocidad : ~10%
- **Cambian rápidamente y lo hacen unas con otras!!**
 - Diferentes contrapartidas en el diseño
 - Lo correcto hoy puede completamente erróneo mañana
- **Requiere constante re-evaluación y re-diseño**

Aplicaciones / Dominios I

- Otra fuerza moldeadora: **aplicaciones**
 - Aplicaciones y dominios de aplicaciones tiene diferentes requerimientos
 - Dominio: grupo con características similares
 - Habitualmente tienden a dar lugar a diferentes diseños
- **Científicas**: predicción del tiempo, cálculo de estructuras,...
 - Primer domino de aplicación de los computadores: cálculo de tablas de tiro
 - Necesitan: muchísima memoria, operaciones en punto flotante
 - Ejemplos: CRAY T3E, IBM BlueGene (www.top500.org)
- **Comercial**: database/web serving, e-commerce, cloud
 - Necesitan: movimiento de datos, memory + I/O bandwidth
 - Ejemplos: Sun Enterprise Server, AMD Opteron, Intel Xeon

Aplicaciones / Dominios II

- **Desktop**: home, office, multimedia, juegos
 - Necesitan: enteros, bandwidth a memoria, aceleración gráfica,..
 - Ejemplo: Intel Core 7i, AMD Bulldozer, IBM Cell (PS3), IBM Xenon (X360)
-
- **Mobile**: portátiles, netbooks, tablets, Smartphones
 - Necesitan: **bajo consumo**, enteros, wireless,...
 - Ejemplos: Intel Atom, ARM A8, Apple A4
-
- **Empotrados**: móviles “tontos”, coches, lavadoras, ...
 - Necesitan: **bajo consumo, bajo coste, SoC**
 - Ejemplos: ARM, X-Scale, MIPS
-
- **Sensores**: disponible “smart dust”
 - Necesitan: **extremo** bajo consumo y coste

Diseños específicos para la aplicación

- Este curso se centra en **arquitecturas de propósito general**
 - Procesadores que pueden hacer cualquier cosa (vía software), específicamente ejecutar un SO
 - E.g., Intel Core i7, IBM Power7, Intel Itanium 2,...
- Existe un mercado (mucho más amplio) de **arquitecturas de propósito específico**
 - Implementan alguna funcionalidad crítica y específica del dominio en hardware
 - Mucho más efectivo (en rendimiento y **consumo**) que el software
 - Ejemplos:
 - Protocolos GSM, HDSPA, h.264, etc....
 - Comienzan a usar técnicas de propósito general

Prerequisitos del curso



¿Para qué estudiar AC?

- Para entender a donde van los computadores
 - Las capacidades futuras guían el resto de la informática
 - Impacto en el mundo real: **Sin arquitectos → No hay computadores**
- Entender los conceptos de alto nivel que intervienen en el diseño de computadores
 - Los mejores diseñadores son aquellos que tienen la visión vertical del computador completamente clara
 - Del transistor a la aplicación
 - No más “*magia*”
- Comprender mejor que acciones hay detrás de vuestro trabajo
 - Los mejores diseñadores de software entienden el hardware
- **Para trabajar como arquitecto**

Objetivos de este curso

- Conocer las **ideas centrales** sobre las que se apoya la arquitectura de computadores
- Entender que es el **rendimiento** de los computadores y las **métricas** que lo determinan
 - Evaluación experimental
 - Obtener experiencia con simuladores (herramienta fundamental de la arquitectura de computadores)
 - Mejorar la comprensión de la interacción hardware-software
- **No es objetivo:** “cómo funcionan los computadores hasta el más ínfimo detalle”: no perder la **perspectiva** y ser **realistas** con el tiempo disponible y que solo es una asignatura introductoria

Perspectiva Histórica I

- Centrado en extraer de modo implícito **paralelismo a nivel de instrucción (ILP)**
 - Podemos disponer de mas recursos: replicación y uso óptimo de ellos
 - Para el software es prácticamente transparente
- Inicialmente usando segmentación ...
 - También posibilita incrementar la frecuencia de reloj
- ... caches ...
 - Se convierte en una necesidad de primer orden a medida que la frecuencia sube
- ... punto flotante integrado
- ... pipelines más profundos y especulación de saltos
- ... ejecución múltiple (superscalar)
- ... y planificación dinámica (out-of-order)

Perspectiva Histórica II

- Las ganancias del ILP se saturan a principio de 00's. No queda mas remedio que optar por el **paralelismo explicito a nivel de thread (TLP)** o **de datos (DLP)**
 - Hardware provee de los recursos necesarios
 - El software debe especificar como se usan
- Desde lo más “directo”: instrucciones vectoriales ..., Intel's SSE
 - Una sola instrucción hace cuatro multiplicaciones simultáneamente (en subwords)
- Hasta lo más “radical”: soporte para programas multi-thread
 - Caches coherentes, soporte hw para primitivas de sincronización...
- ... soporte para hacer todo esto en un chip (**REVOLUCIÓN**)

Temario

- **Fundamentos del diseño de computadores**
- **El repertorio de Instrucciones**
- **Jerarquía de Memoria**
 - Cache
 - Memoria virtual
- **Paralelismo a nivel de instrucción (ILP)**
 - Segmentación y predicción de saltos
 - Arquitecturas super-escalares
 - Ejecución fuera de orden
- **Paralelismo a nivel de *Thread* (TLP)**
 - Multiprocesadores (SMP)
 - Multicores (CMP)
 - Multithreading (SMT)
- **Paralelismo a nivel de datos (DLP)**

Prácticas

- Objetivo
 - **Apoyar** los conocimientos de teoría
 - Una específicamente diseñada para **cada tema**
- Dos partes por práctica
 - Obligatoria
 - **Guiada**
 - Las horas de **dedicación** no deben exceder las horas de laboratorio preestablecidas (3 créditos)
 - Opcional
 - Objetivos abiertos
 - **Trabajo en grupo**
 - **Listas disponibles esta semana**

Material

- Bibliografía Básica
 - **John Hennessy, David Patterson**, "Computer Architecture A Quantitative Approach", **4 Edition**, ISBN: 1-55860-596-7, (**H&P**)
- Bibliografía Complementaria
 - **J.P. Shen and M. Lipasti**, "Modern Processor Design: Fundamentals of Superscalar Processors", 1st edition, McGraw-Hill, ISBN-10: 0070570647, (**S&L**)
- Transparencias (**Inglés**)

Material Prácticas

- Simulador
 - www.simics.net
 - Entorno basado en máquina virtual
- Las prácticas se pueden apoyar con trabajo en casa(aunque no recomendable hacerlas íntegramente así)
 - Experimentación con partes abiertas, puede ser favorable

Acceso al material de apoyo

- Solo en página Web-CT de la asignatura
 - Transparencias teoría
 - Problemas
 - Exámenes
 - Guiones de prácticas
 - Máquina virtual prácticas
 - Planificación de los temas
 - Otra información de utilidad...
- Acceso personalizado
- Comunicación con el profesor y/o compañeros
 - Preferiblemente correo valentin.puente@unican.es o pablo.abad@unican.es
 - asig5411@unican.es

Evaluación

- 60% Teoría
 - **Examen** final en junio
- 40 % Prácticas
 - Parte obligatoria calificable mediante **examen** final en junio (~6/10)
 - Parte opcional calificable por **presentación + memoria**(~4/10)