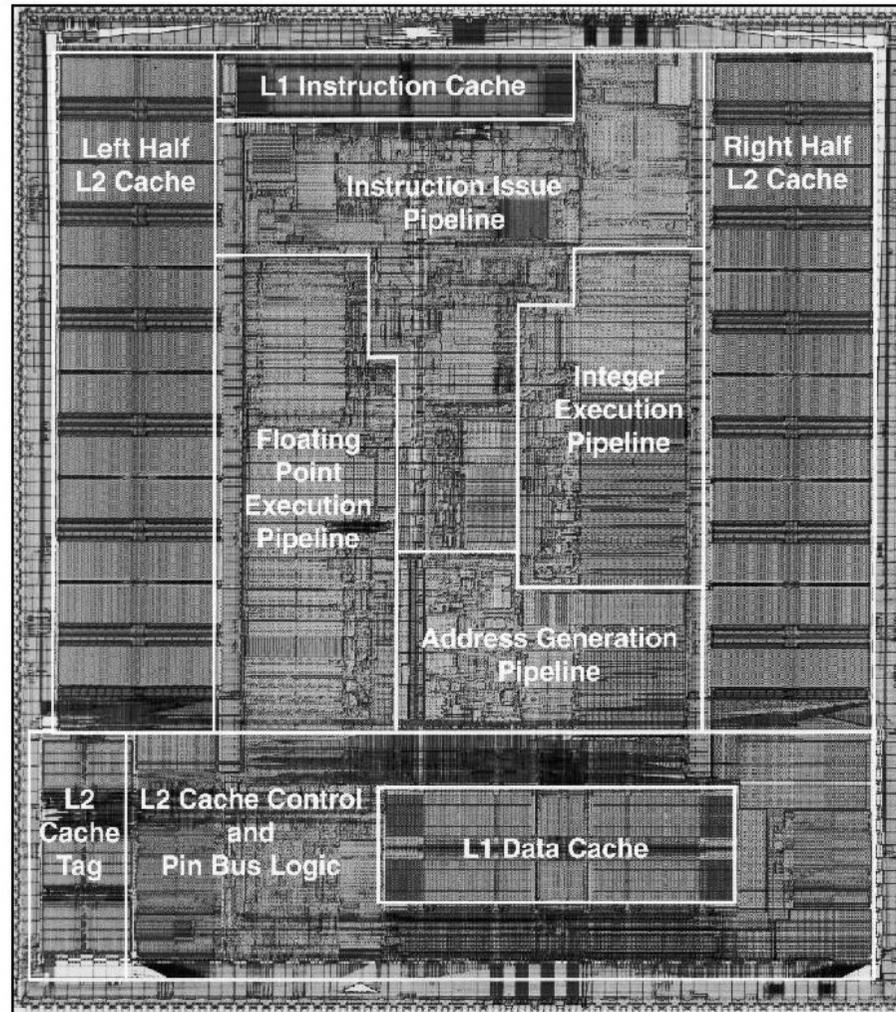


Evolución y tendencias futuras de los microprocesadores, microcontroladores y DSP.



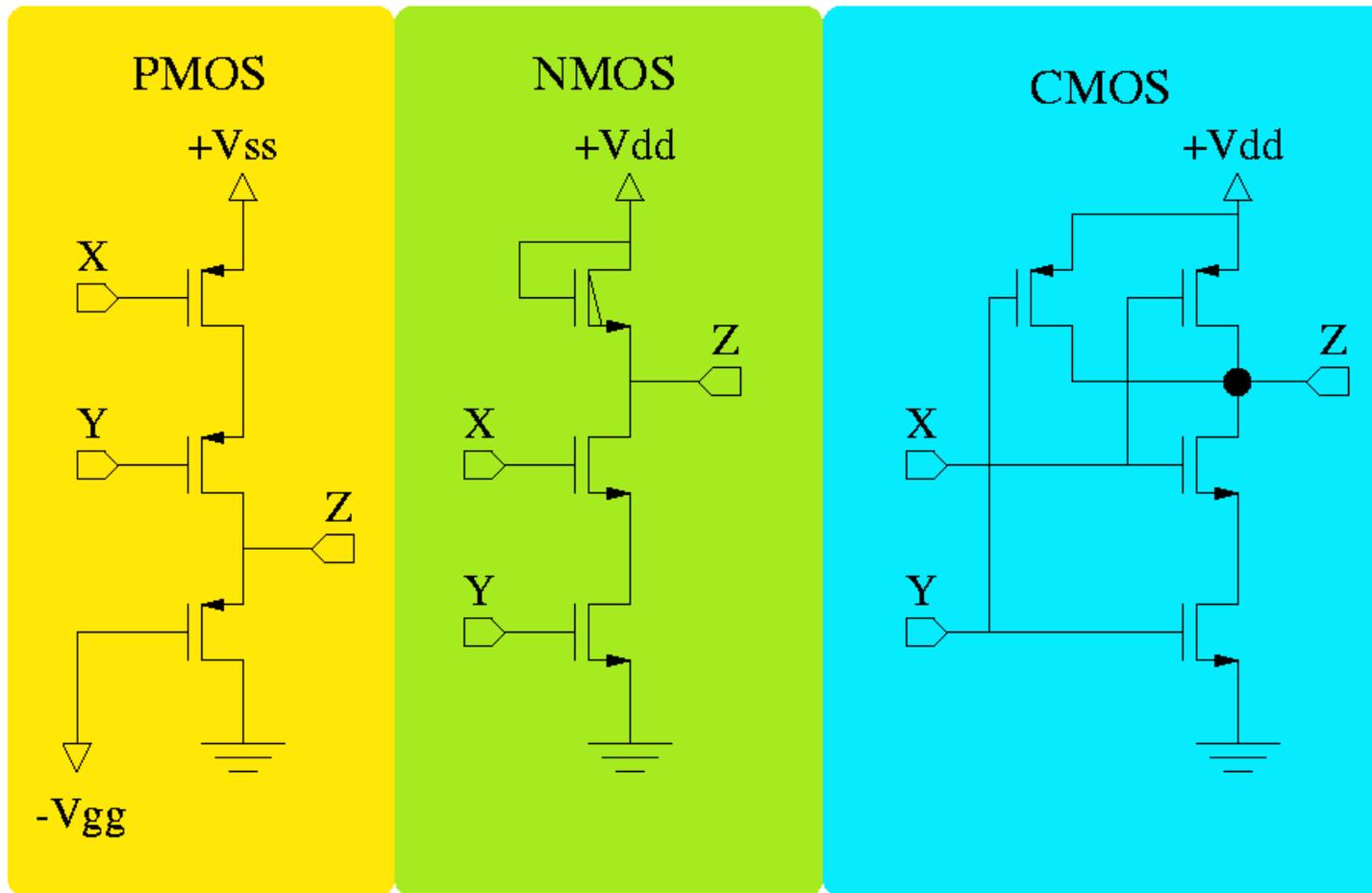
Alpha 21164. Microprocesador RISC superescalar.

Evolución y tendencias futuras de los microprocesadores, microcontroladores y DSP

- Tecnología:
 - Evolución.
 - Nuevos retos.
- Arquitectura:
 - Evolución CISC vs. RISC.
 - Arquitecturas avanzadas. Conceptos:
 - Caché.
 - Ejecución segmentada: Pipelining.
 - Paralelismo.
- Aplicación específica.
 - Microprocesadores.
 - Microcontroladores.
 - DSP.

Tecnología. Evolución: Lógica.

FAMILIAS LOGICAS



Tecnología. Retos.

- Litografía. (Actualidad: 180 nm).
 - Ultravioleta profundo ($\lambda \approx 190$ nm).
 - Límite físico: 100 nm.
 - Alternativas:
 - Litografía de barrido de haz de electrones.
 - Litografía de rayos X.
 - Reducción del coste.
 - Proyección por haz de electrones.

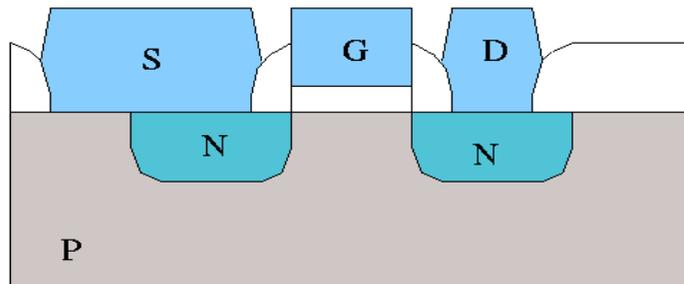
Tecnología. Retos.

- Dispositivos. (Transistores)
 - Menor longitud de canal.
 - Menor área.
 - Inyección en el aislante de puerta.
 - Reducción de la tensión de alimentación.
 - Dieléctricos de alta permitividad. (Nitruro de Silicio)
 - Fluctuación estadística de dopantes.
 - Control de tensión umbral difícil.

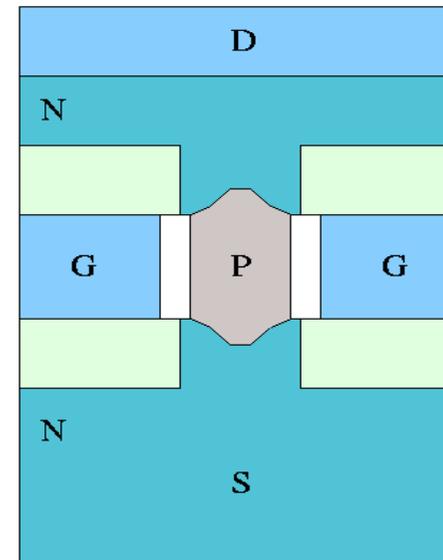
Tecnología. Retos: Dispositivos.

Transistor MOSFET vertical (Bell-Labs)

Transistor convencional



Transistor vertical

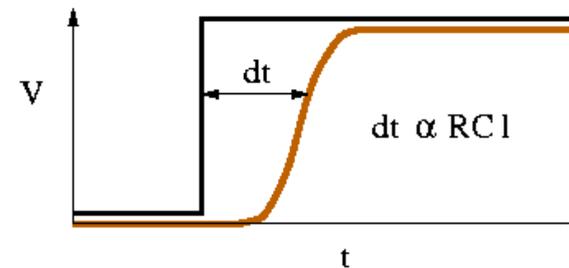
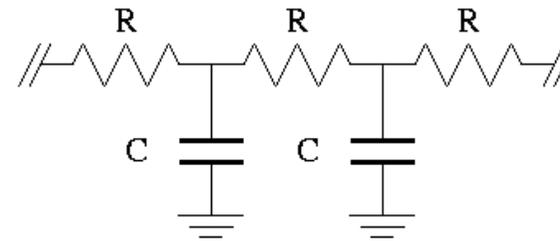
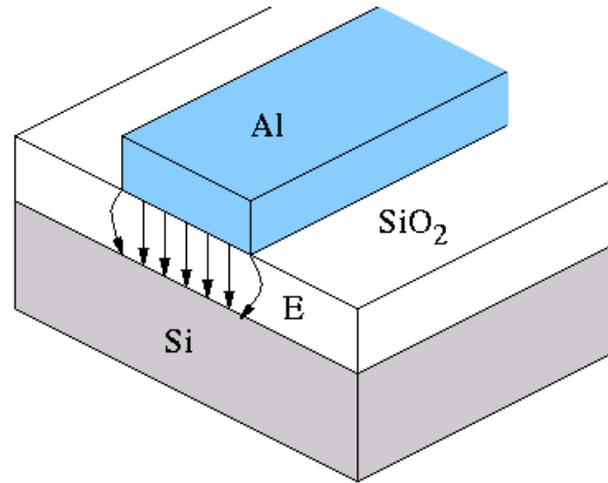


Ventajas del transistor vertical

- Control preciso de la longitud de puerta.
- Menor superficie ocupada
- Menor R_{ON}

Tecnología. Retos:

RETARDO DE INTERCONEXION



Resistividad ($\mu\Omega$ cm)	Constante dieléctrica relativa	Retardo relativo (0.1 μ m)
Al: 3.0	SiO ₂ : 4.0	1
Cu: 1.7	Low-K: 2.0	0.32

Materiales de baja constante dieléctrica

SiO₂ poroso

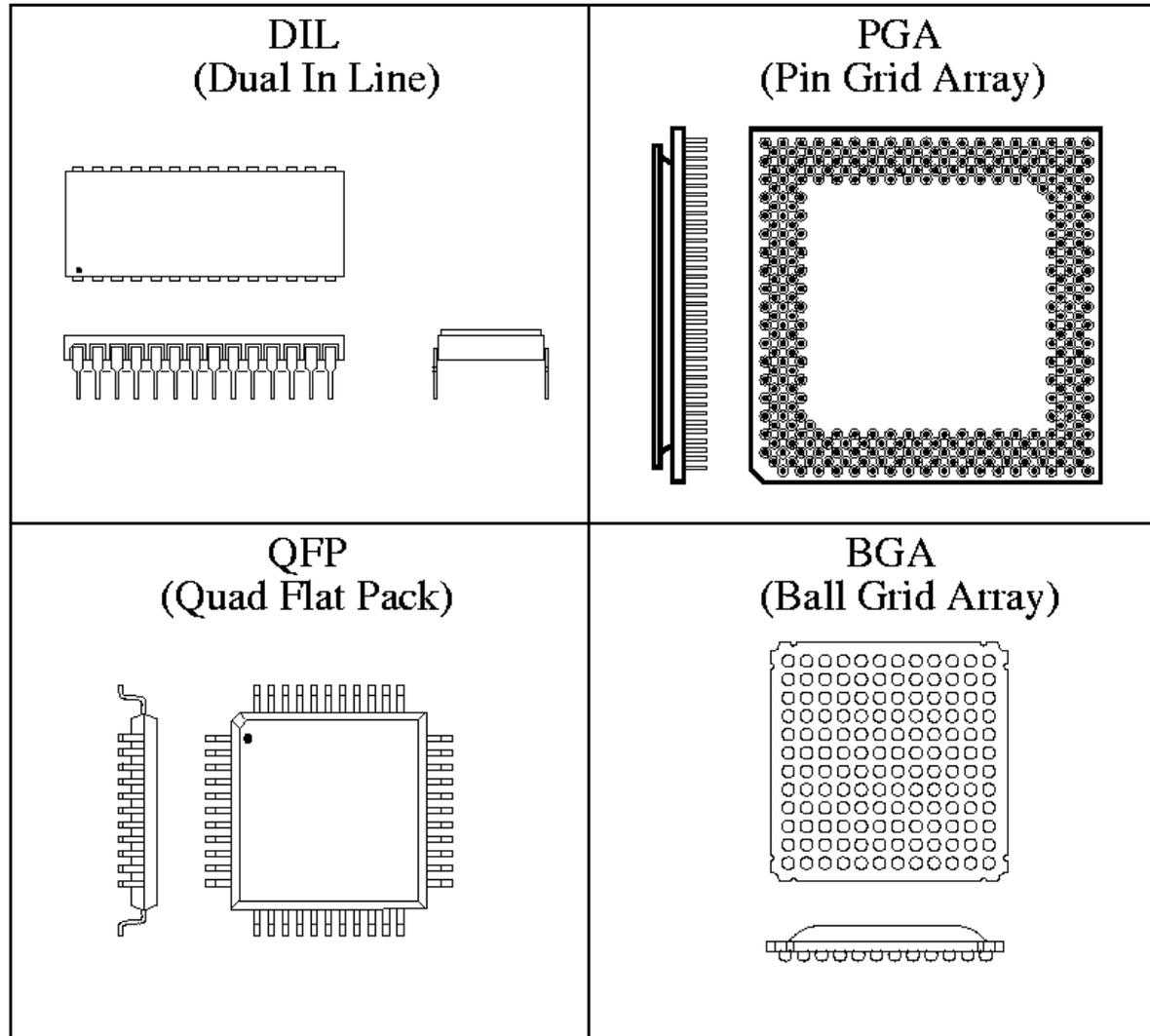
Carbono fluorado

Polímeros

Tecnología. Retos: Encapsulado.

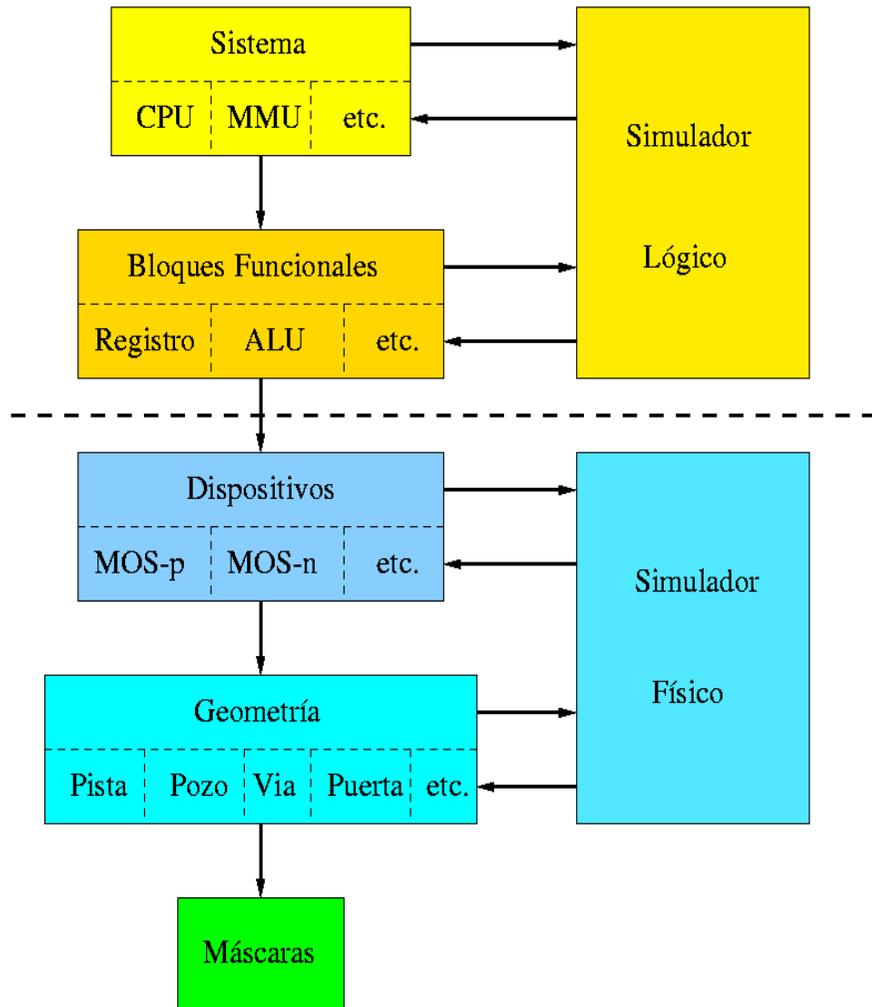
- Mayor número de terminales. (Actualidad ~400).
 - Terminales de menores dimensiones.
 - Terminales distribuidos por todo el integrado (PGA,BGA).
- Elementos de circuito parásitos menores.
 - Mayor frecuencia de funcionamiento del sistema.
 - Módulos Multichip (MCM).
- Baja resistencia térmica.
 - *Flip-Chip*.

Tecnología. Retos. Encapsulado. Evolución.

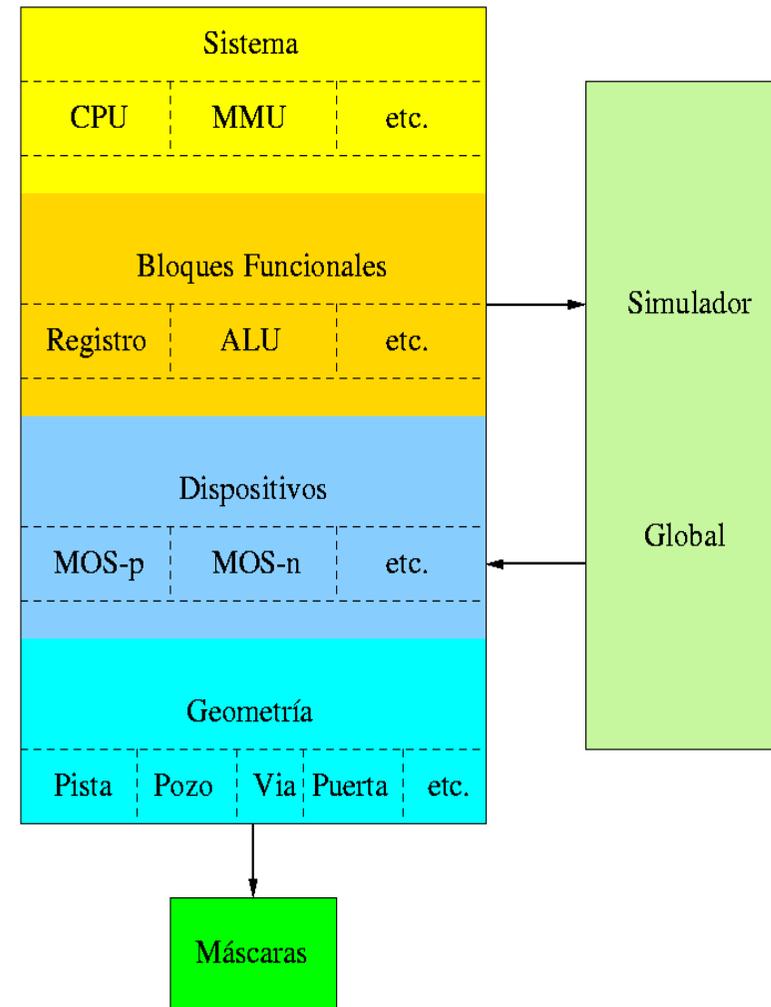


Tecnología. Retos: Diseño.

Diseño Jerarquico



Diseño Unificado



Tecnología. Retos: Test.

- Principales problemas:
 - Coste equipo de test.
 - Sondas: Elementos de circuito parásitos.
- Soluciones: Diseño orientado al test.
 - *Built-In Self Test* (BIST).
 - *Boundary-Scan* (JTAG).

Arquitectura. Evolución CISC vs RISC.

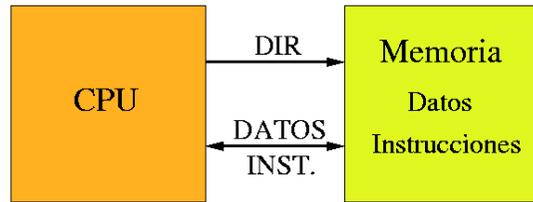
- Años 80: CISC: (68020, I386.)
 - Instrucciones complejas, orientadas a lenguajes de alto nivel.
 - Manejo de cadenas (REP MOVS).
 - Preparación de tramas de pila (LINK, UNLINK, ENTER, LEAVE).
 - Bucles (BDcc, LOOP).
 - Manejo de campos de bit (BSET, BSF).
 - Modos de direccionamiento complejos:
 - Registro-Registro, Registro-Memoria, Memoria-Memoria.
 - $EA = (\text{Base} + \text{Índice} * \text{escala} + \text{desplazamiento})$
 - No todos los direccionamientos son válidos en todas las instrucciones.
 - Códigos de operación de longitud variable (I386: 1 - 15 bytes)
 - Microprogramación.
 - Arquitectura Von-Neumann (Memoria unificada).

Arquitectura. Evolución CISC vs RISC.

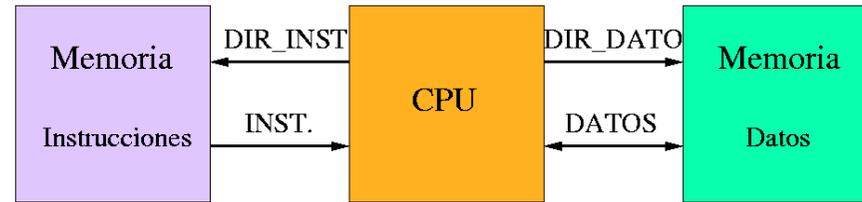
- Años 90: RISC:
 - Procesadores SPARC, PowerPC, Alpha, MIPS, PA-RISC, ARM, etc.
 - Sólo instrucciones simples con operandos en registros.
 - Sólo las instrucciones LOAD y STORE acceden a la memoria.
 - Instrucciones de 3 operandos.
 - Grandes bancos de registros.
 - Ventanas de registros (SPARC).
 - Direccionamientos sencillos en instrucciones LOAD y STORE.
 - Códigos de operación de longitud fija (32 o 64 bits).
 - Ejecución segmentada (*Pipepiling*).
 - Arquitectura Harvard (Memorias de programa y datos separadas).

Arquitectura. Modelos Von-Neumann y Harvard.

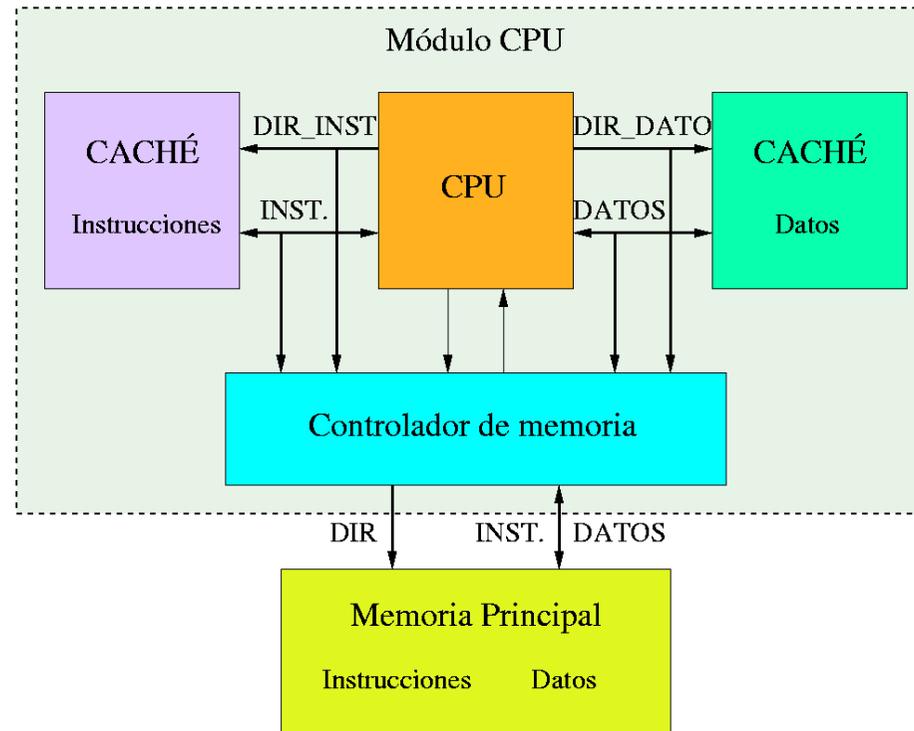
Arquitectura VON-NEWMANN



Arquitectura HARVARD



Arquitectura moderna, HIBRIDA



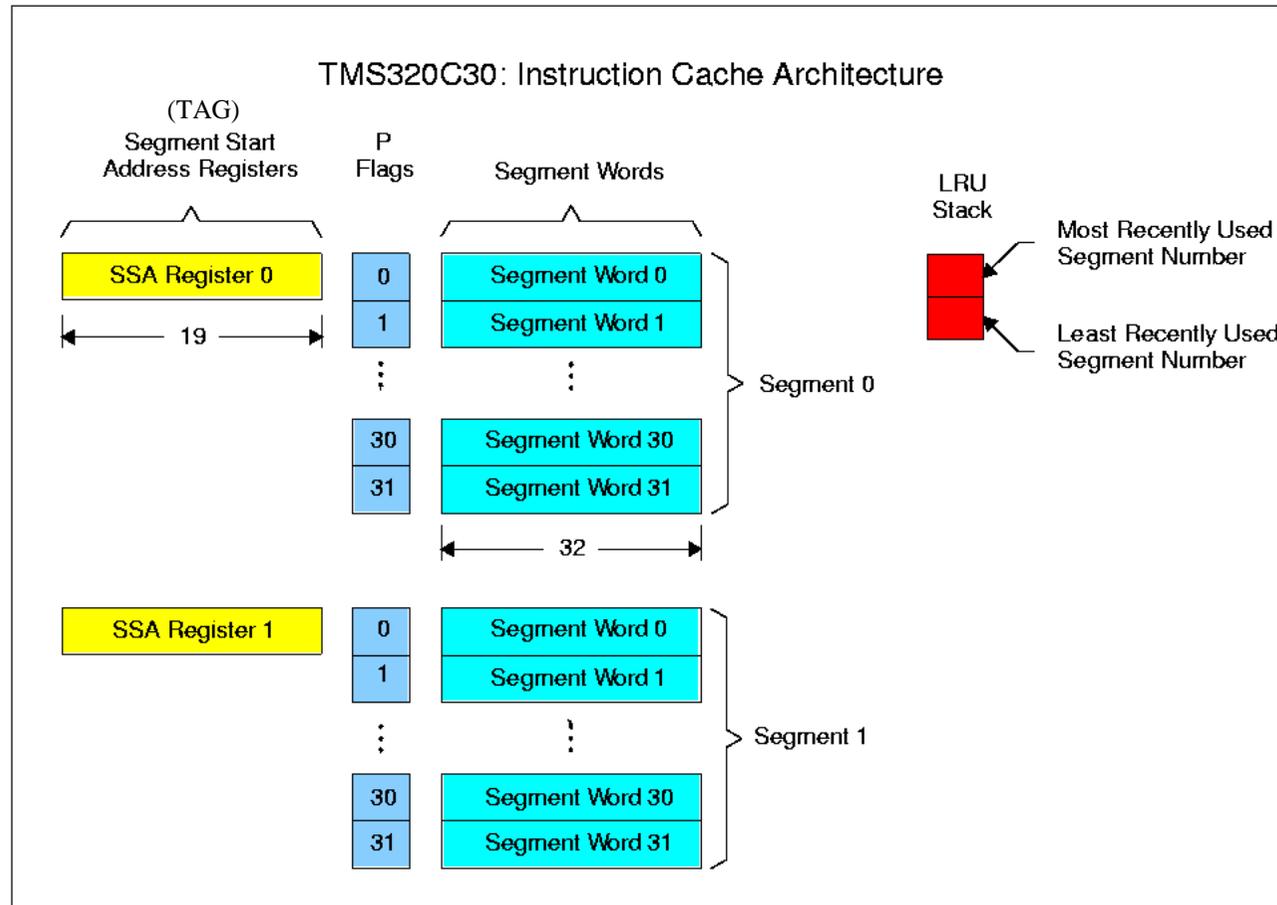
Arquitecturas avanzadas.

- Memoria Caché.
 - Concepto.
 - Escritura en la caché.
 - Caché multinivel.
- Ejecución segmentada. (Pipelining).
 - Concepto.
 - Bloqueo del pipeline.
- Paralelismo.
 - Paralelismo SIMD.
 - Paralelismo MIMD.
 - Sistemas Multiprocesador.

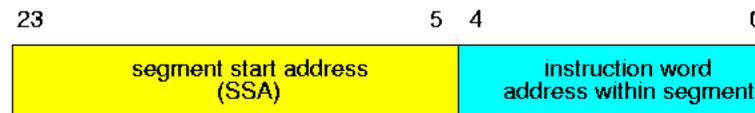
Arquitecturas avanzadas. Caché.

- Problema: La memoria principal, DRAM, es mucho más lenta que el procesador.
 - Ciclo CPU: ~ 2 ns.
 - Acceso DRAM: ~ 60 ns. (10 ns SDRAM).
- Solución:
 - Mantener una copia de los datos más frecuentes en una memoria rápida aunque pequeña: Memoria Caché.
- Implementación:
 - Memorias asociativas.
 - Integración en el propio chip.

Arquitecturas avanzadas. Caché. Ejemplo:



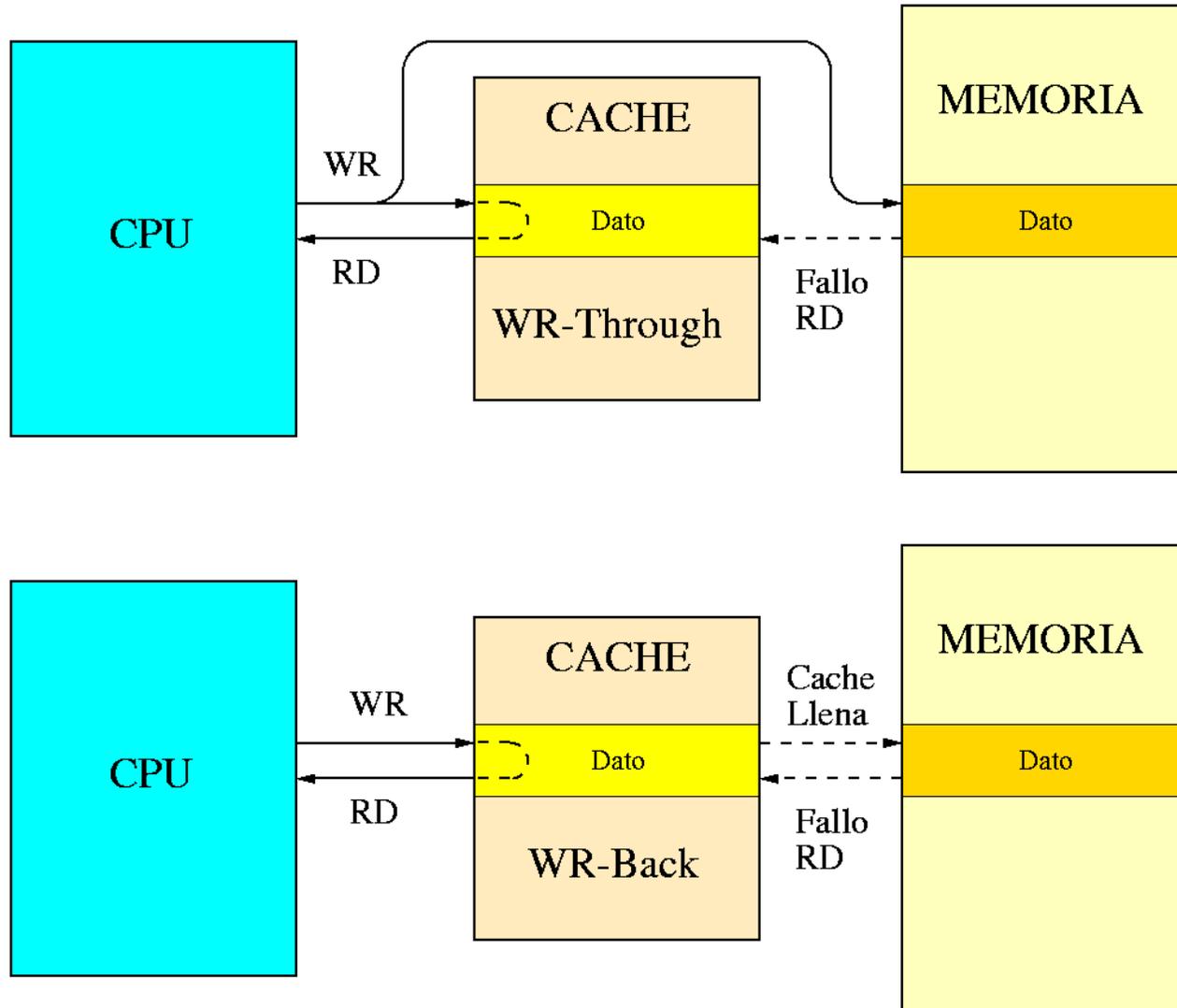
Address Partitioning for Cache Control Algorithm



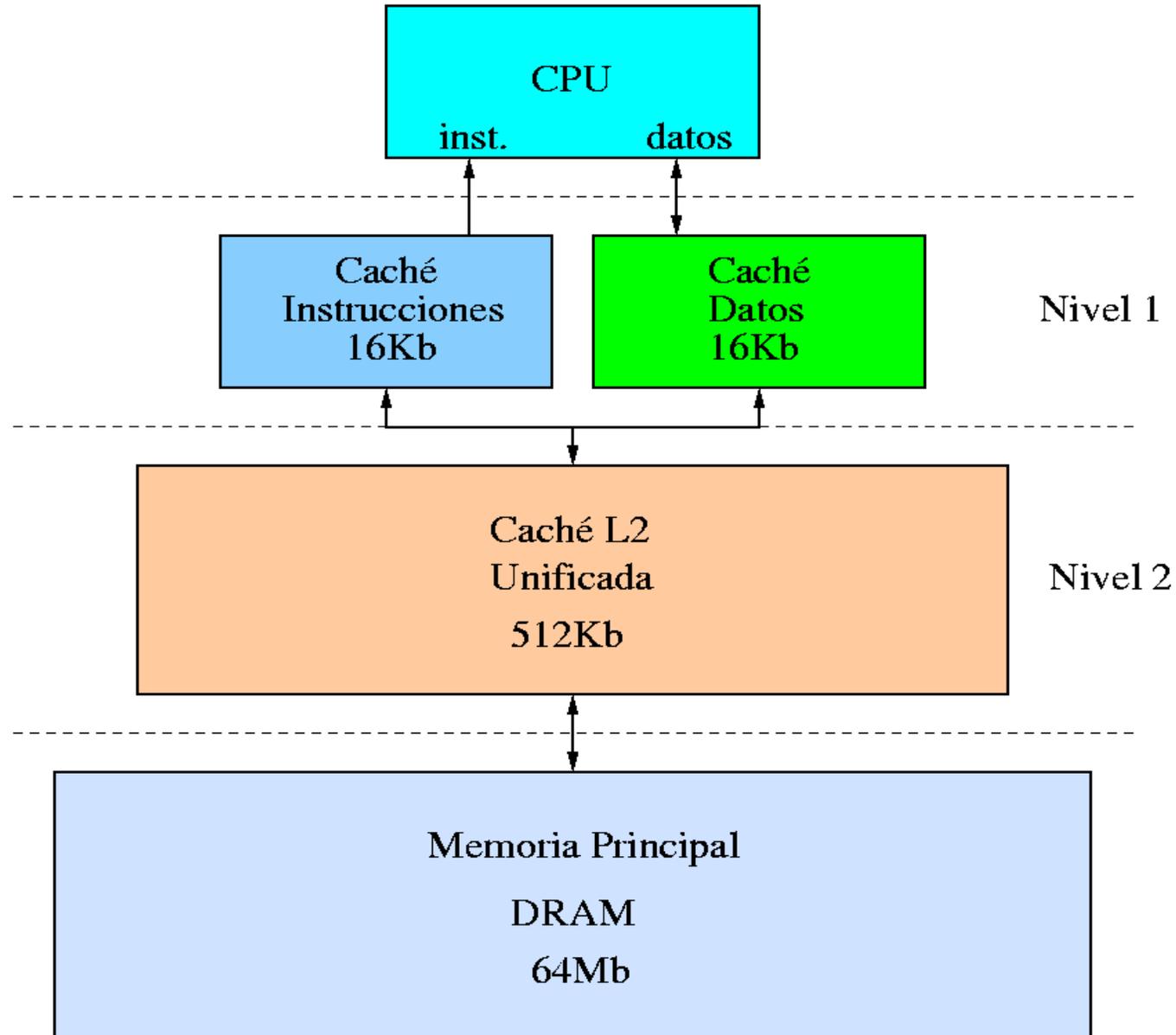
Arquitecturas avanzadas. Caché. Ejemplo:

- Coincidencia (MSB dirección = SSA, bit P = 1):
 - El dato se lee de la memoria caché.
 - El segmento pasa a “más recientemente usado”
- Fallo (bit P = 0):
 - Se lee el dato de la memoria principal y se copia en la caché.
 - Se pone el bit P en 1.
 - El segmento pasa a “más recientemente usado”
- Fallo (MSB dirección \neq SSA):
 - Se borran todos los bits P del segmento menos recientemente usado.
 - Se carga el registro SSA con los bits MSB de la dirección.
 - Se procede igual que en el caso anterior.

Arquitecturas avanzadas. Caché. Escritura.



Arquitecturas avanzadas. Caché. Caché multinivel.

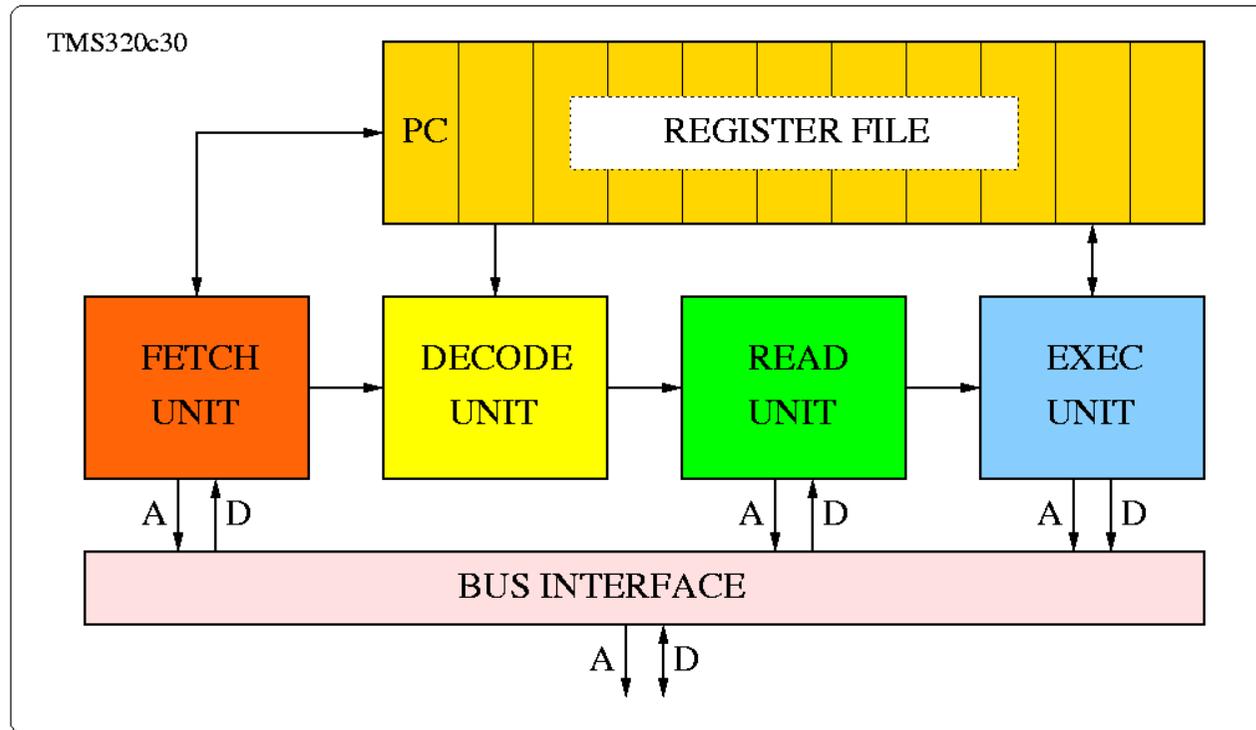


Arquitecturas avanzadas. Pipelining.

- Concepto: Ejecución segmentada:
 - Dividir la ejecución de todas las instrucciones en un número fijo de etapas.
 - Cada etapa se ejecuta en una unidad diferente.
 - Todas las unidades están operando simultáneamente sobre instrucciones distintas.

Arquitecturas avanzadas. Pipelining. Ejemplo:

EJECUCION SEGMENTADA (PIPELINING)



- ◊ FETCH : Trae el código de operación de la memoria. Incrementa PC.
- ◊ DECODE : Decodifica la instrucción. Calcula dirección del operando.
- ◊ READ : Lee el operando de la memoria si es necesario.
- ◊ EXEC : Realiza la operación y escribe el resultado.

Arquitecturas avanzadas. Pipelining. Ejemplo:

Ejecución de programas en el "PIPELINE"

Secuencia de Programa

Ciclo	FETCH	DECODE	READ	EXEC
n	ADDI	***	***	***
n+1	MPYI	ADDI	***	***
n+2	AND	MPYI	ADDI	***
n+3	STF	AND	MPYI	ADDI
n+4	NEGF	STF	AND	MPYI
⋮	⋮	⋮	⋮	⋮

LATENCIA: 4 CICLOS

RENDIMIENTO: 1 INSTRUCCION por CICLO

Arquitecturas Avanzadas. Pipelining. Bloqueo.

- Bloqueo del pipeline (Stall):
 - Alguna o varias unidades del pipeline quedan inactivas (ejecutando NOP's implícitos) durante uno o varios ciclos de reloj.
- Causas del bloqueo del pipeline:
 - Bifurcaciones del programa.
 - Conflictos de Registros.
 - Conflictos de Bus.

Arquitecturas Avanzadas. Pipelining. Bloqueo.

1 Bifurcaciones del programa

- Las instrucciones de salto alteran el PC, pero al final de su fase de ejecución.
- Hay que descartar las instrucciones posteriores y esperar la ejecución del salto.

Ejemplo:

PC → BR Etiq

MPYF

Etiqu: OR

Ciclo	FETCH	DECODE	READ	EXEC
n	BR	***	***	***
n+1	MPYF	BR	***	***
n+2	---	---	BR	***
n+3	---	---	---	BR
n+4	OR	---	---	---

n+1 Se detecta la instrucción BR. Se descarta MPYF.

n+2, n+3 Se vacia el Pipeline. Se ejecuta BR y cambia el PC.

n+4 Se reanuda la ejecución tras el salto. Han transcurrido 4 ciclos.

Arquitecturas avanzadas. Pipelining. Bloqueo.

◦ 2 Conflictos de registros.

- Cuando una instrucción usa un registro que modifica la instrucción anterior se debe bloquear parte del pipeline hasta que el valor del registro sea correcto.

Ciclo	FETCH	DECODE	READ	EXEC
n	ADDI	***	***	***
n+1	LDI	ADDI	***	***
n+2	ANDI	LDI	ADDI	***
n+3	ANDI	LDI	---	ADDI
n+4	ANDI	LDI	---	---
n+5	ORI	ANDI	LDI	---

Ejemplo:

ADDI R1,AR0
LDI *AR0,R2

ANDI
ORI

n+2 La fase DECODE de LDI necesita el valor de AR0, pero AR0 lo modifica ADDI.

n+3, n+4 Las fases FETCH y DECODE siguen bloqueadas. ADDI calcula AR0

n+5 Continúa la ejecución. Se han perdido 2 ciclos.

Arquitecturas Avanzadas. Pipelining. Bloqueo.

- 3 Conflicto de bus.
 - Varias unidades del pipeline intentan acceder a la memoria simultáneamente.
 - La unidad FETCH siempre está leyendo la memoria. Entraría en conflicto con cualquier otro acceso.

Arquitecturas avanzadas. Pipelining. Bloqueo.

Soluciones:

- 1 Bifurcaciones del programa:
 - Reducir el número de saltos en los programas:
 - Optimización de los compiladores.
 - Instrucciones de ejecución condicional (Predicados).
 - Saltos retardados:
 - No se vacía el pipeline, por lo que se ejecutan varias instrucciones tras el salto.
 - Se utiliza la instrucción NOP para rellenar el pipeline si fuera necesario.
 - Predicción de saltos:
 - Se proporciona a la unidad FETCH una dirección para seguir buscando código antes de completarse la ejecución del salto.

Arquitecturas avanzadas. Pipelining. Bloqueo.

Predicción de saltos:

- Estática:
 - Saltos incondicionales.
 - Saltos condicionales ejecutados por primera vez.
 - Predicción de la ejecución o no del salto.
 - Variables: Signo del desplazamiento, Código de condición.
- Dinámica:
 - Tabla histórico de saltos (BHT).
 - Bits histórico de ejecución del salto: Predicen si el salto se va a ejecutar.
 - Caché de direcciones de salto.
 - Pila hardware para el retorno de subrutinas.
 - Predice la dirección de retorno de las subrutinas más anidadas.

Arquitecturas avanzadas. Pipelining. Bloqueo.

Soluciones:

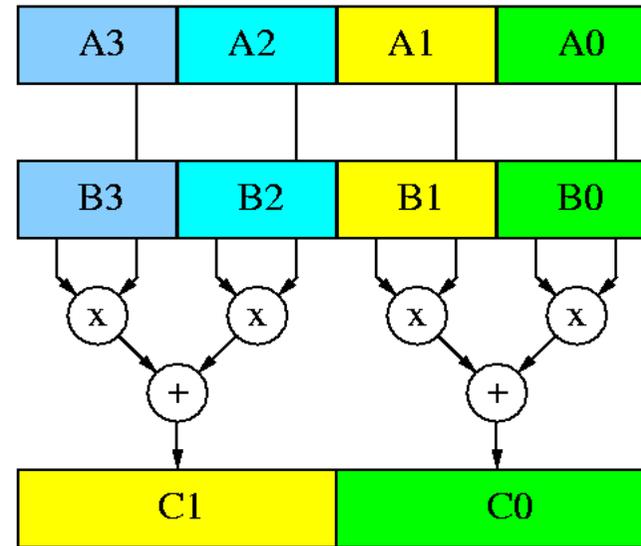
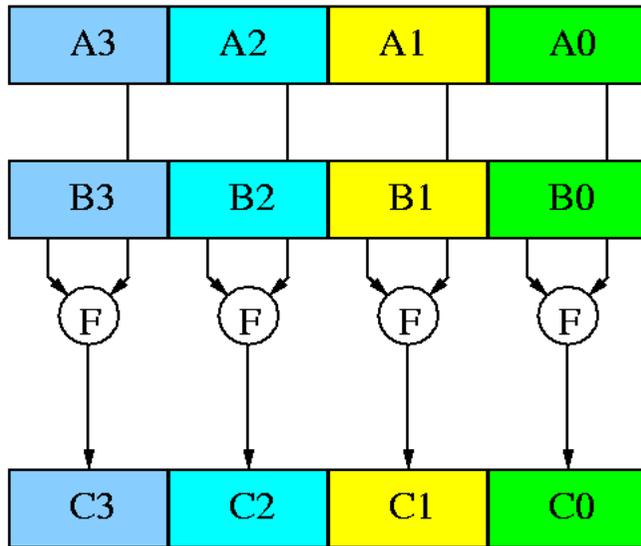
- 2 Conflictos de registros:
 - Optimización de los compiladores.
 - Disponibilidad de muchos registros.
- 3 Conflictos de bus:
 - Arquitectura Harvard:
 - Memoria de programa y de datos separadas.
 - La unidad FETCH lee su propia memoria.
 - Caché multipuerto:
 - Permite varios accesos simultáneos.

Arquitecturas avanzadas. Paralelismo.

- Paralelismo SIMD: Procesamiento vectorial.
 - Extensiones MMX, 3Dnow, Streaming SIMD, AltiVec.
- Paralelismo MIMD: Procesadores superescalares.
 - Procesadores con varias pipelines de ejecución.
- Ordenadores Multiprocesador: SMP.
 - Memoria compartida. Coherencia de las cachés.

Arquitecturas avanzadas. Paralelismo SIMD.

PROCESAMIENTO VECTORIAL: ARITMÉTICA



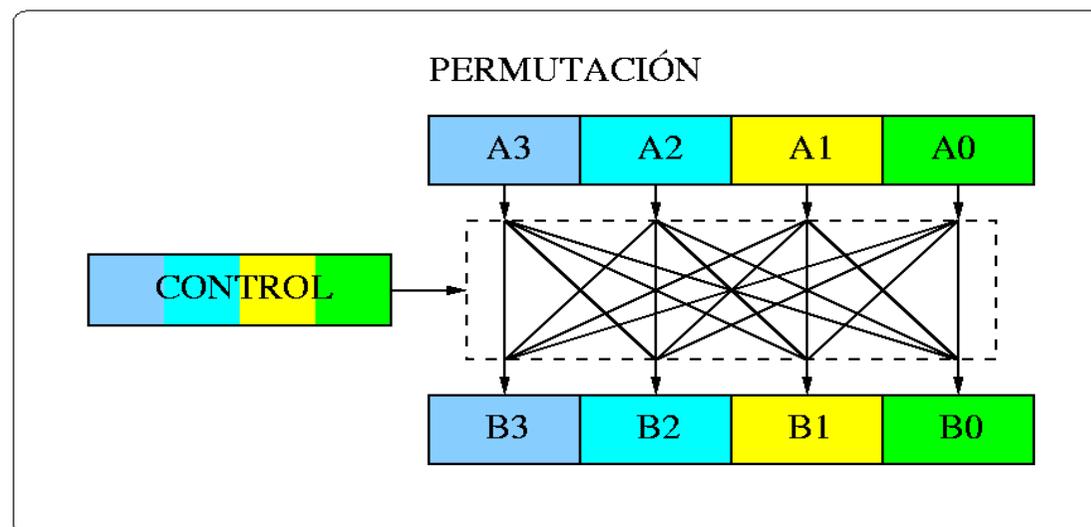
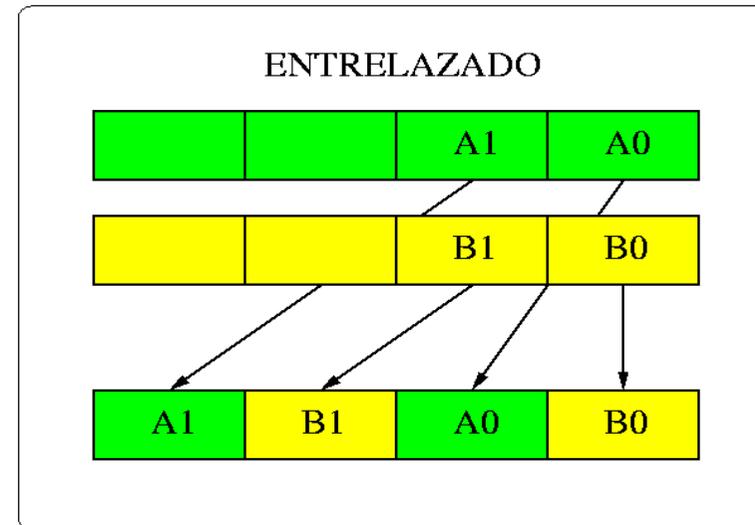
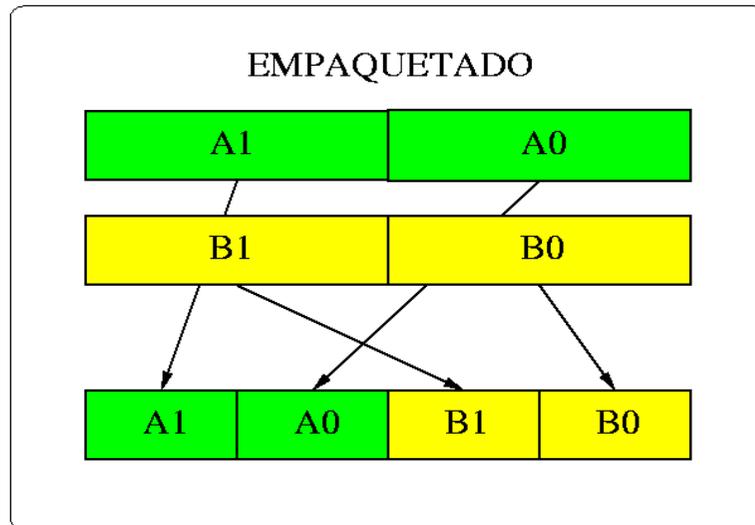
$C=F(A,B)$ F {
Suma, Resta, Multiplicación.
Operaciones lógicas. Desplazamientos.
Comparación.

Instrucción MAC: Convolución.

Desbordamiento {
Truncamiento.
Saturación sin signo.
Saturación con signo.

Arquitecturas avanzadas. Paralelismo SIMD.

PROCESAMIENTO VECTORIAL. MOVIMIENTO DE DATOS



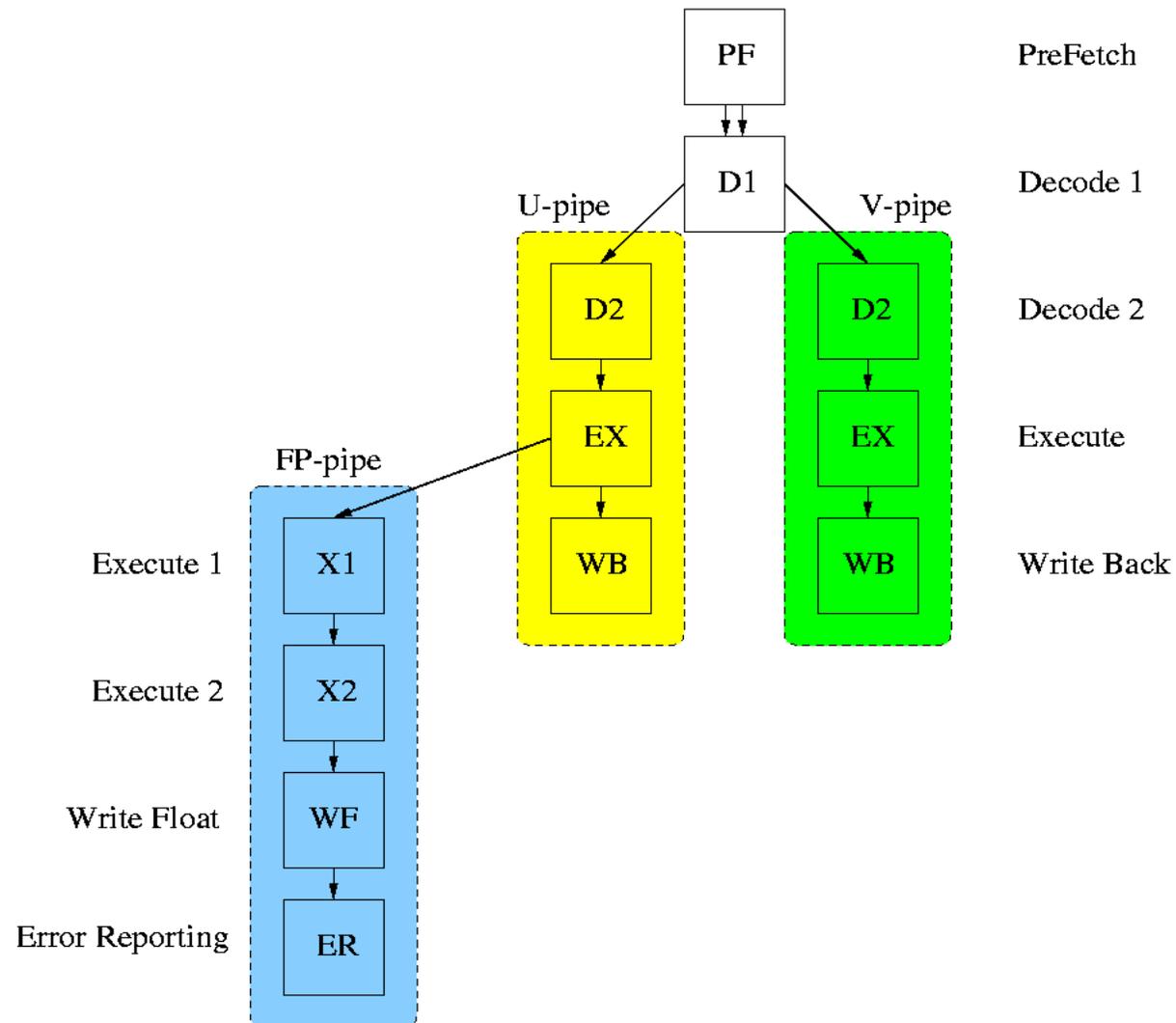
Arquitecturas avanzadas. Paralelismo MIMD.

Procesadores superescalares.

- Paralelismo a nivel de instrucción (ILP):
 - Varias pipelines de ejecución.
 - Más de una instrucción terminada cada ciclo de reloj.
 - Dependencias de datos: pérdida de rendimiento.
 - Flujo de datos vs Flujo de programa: Ejecución fuera de orden.

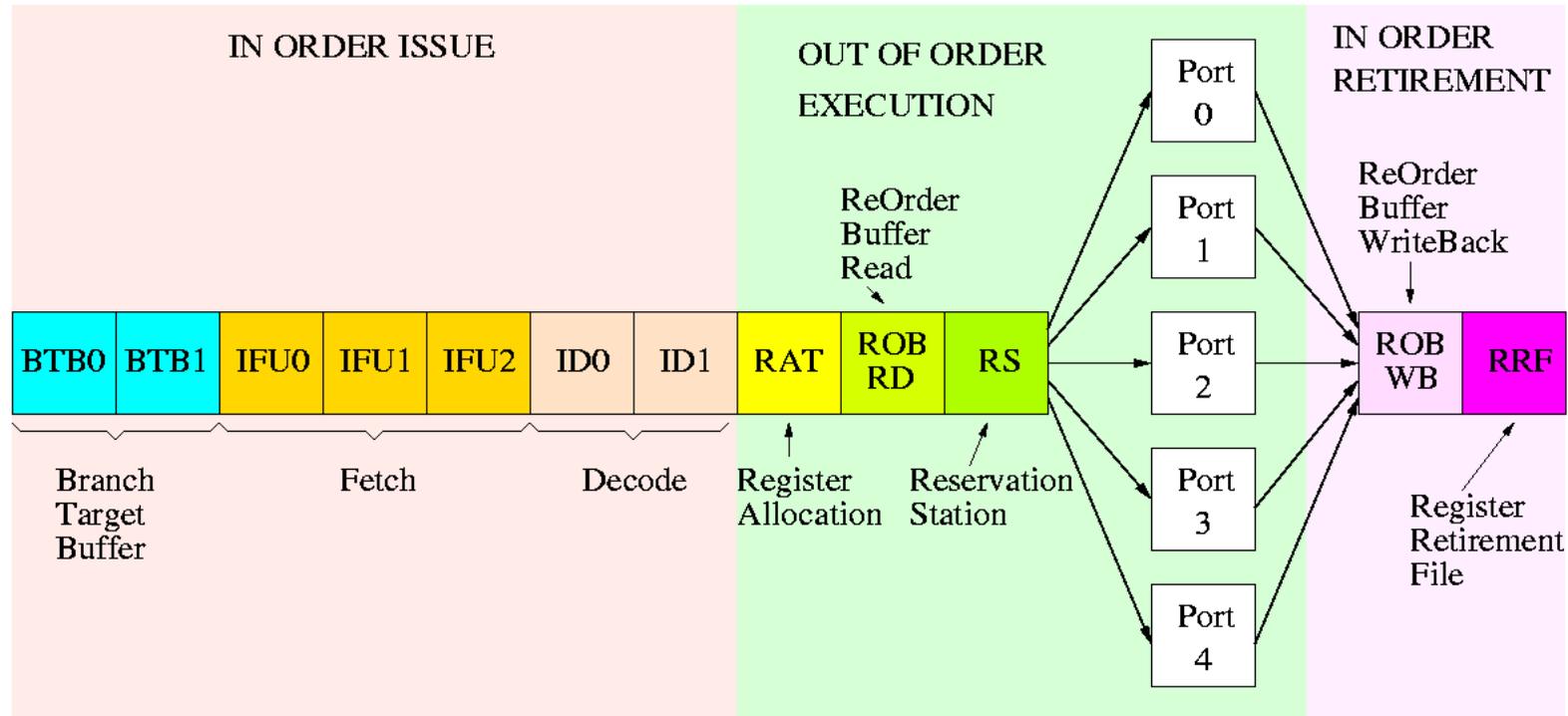
Arquitecturas avanzadas. Paralelismo MIMD.

PROCESADOR SUPERESCALAR: PENTIUM



Arquitecturas avanzadas. Paralelismo MIMD.

EJECUCION FUERA DE ORDEN: PROCESADORES PENTIUM PRO, PENTIUM-II



Port 0: { Integer ALU
Integer MUL
Floating Point
MMX ALU
MMX MUL

Port 1: { Integer ALU
MMX ALU

Port 2: Load Unit

Port 3: Store Address Unit

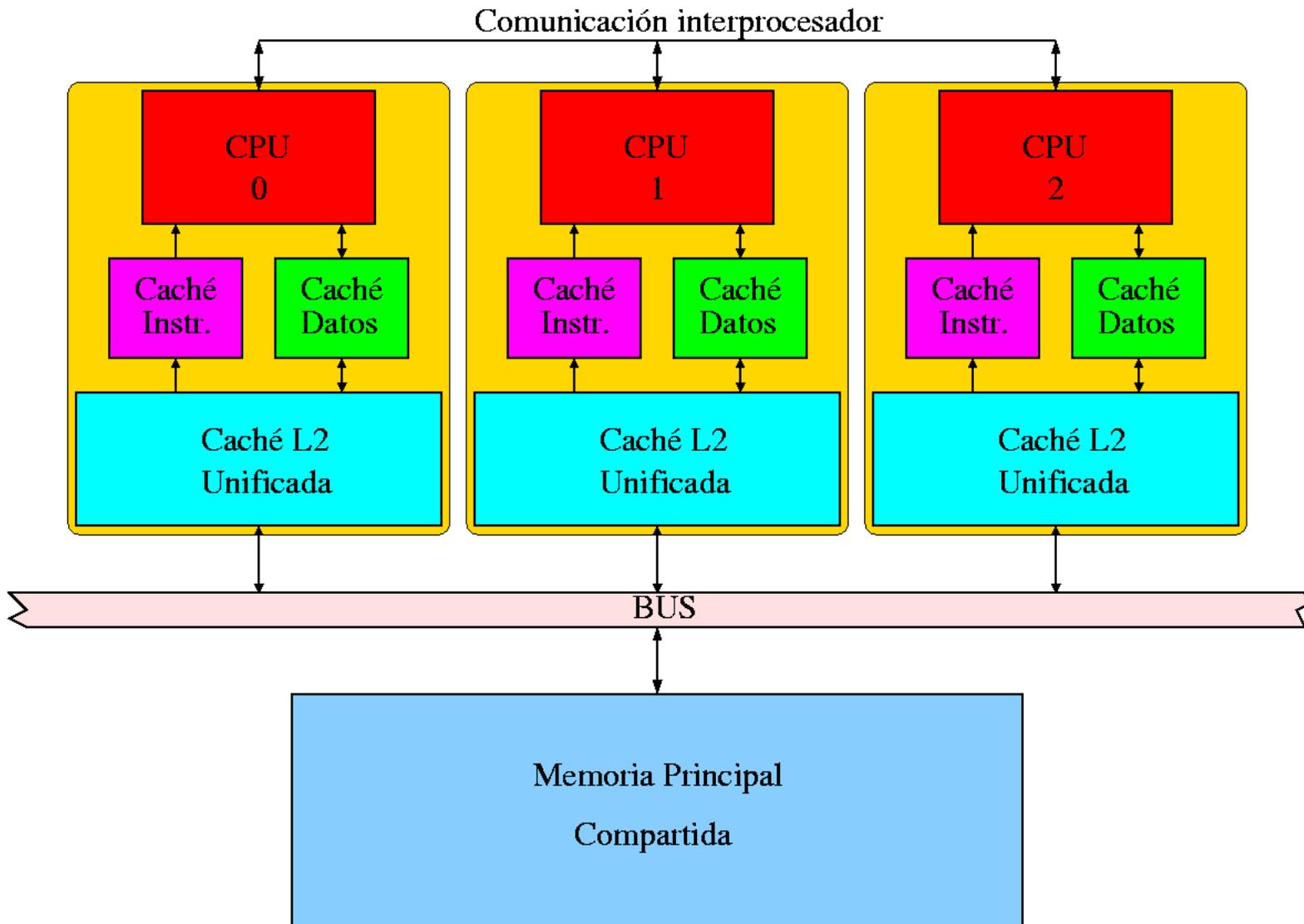
Port 4: Store data Unit.

Arquitecturas avanzadas. Paralelismo MIMD.

Ejecución fuera de orden. Procesadores Pentium-Pro, Pentium-II.

- *In Order Issue.*
 - Lee códigos de operación (Hasta 3 instrucciones simultáneas).
 - Realiza la predicción de saltos (BTB).
 - Convierte las instrucciones x86 (CISC) en μ ops (RISC). (Hasta 6 μ ops simultáneos: 4-1-1)
- *Out of Order Execution.*
 - Asigna registros del *Re-Order Buffer* a las μ ops (RAT).
 - Las μ ops que no tienen listos sus datos quedan a la espera en la etapa *Reservation Station*.
 - Distribuye las μ ops a las cinco pipelines de ejecución (*Ports*).
- *In Order Retirement.*
 - Recupera los datos ya procesados del *Re-Order Buffer* y los asigna a los registros de la arquitectura x86.

Arquitecturas avanzadas. Sistemas multiprocesador.



Arquitecturas avanzadas. Sistemas multiprocesador.

Sistemas Multiprocesador Simétricos (SMP)

- Memoria principal compartida.
 - Limitación de ancho de banda.
- Cachés locales.
 - Menor acceso a la memoria principal.
 - Coherencia de datos entre cachés.
 - Cuando un procesador modifica una línea de su caché lo notifica al resto de los procesadores.
 - Los procesadores que tengan una copia de esa línea en sus respectivas cachés deben invalidarla.
 - Las líneas de caché modificadas se deben escribir en la memoria principal cuando otro procesador lea dichos datos.

Aplicación específica.

- Microprocesadores.
 - Máxima potencia de cálculo.
 - Soporte de Sistemas Operativos avanzados.
- Microcontroladores.
 - Bajo coste.
 - Integración de memoria y periféricos.
 - Bajo consumo eléctrico.
- Procesadores Digitales de Señal (DSP).
 - Orientados a la ejecución de algoritmos de procesamiento de señales.
 - Compromiso entre potencia de cálculo y coste.

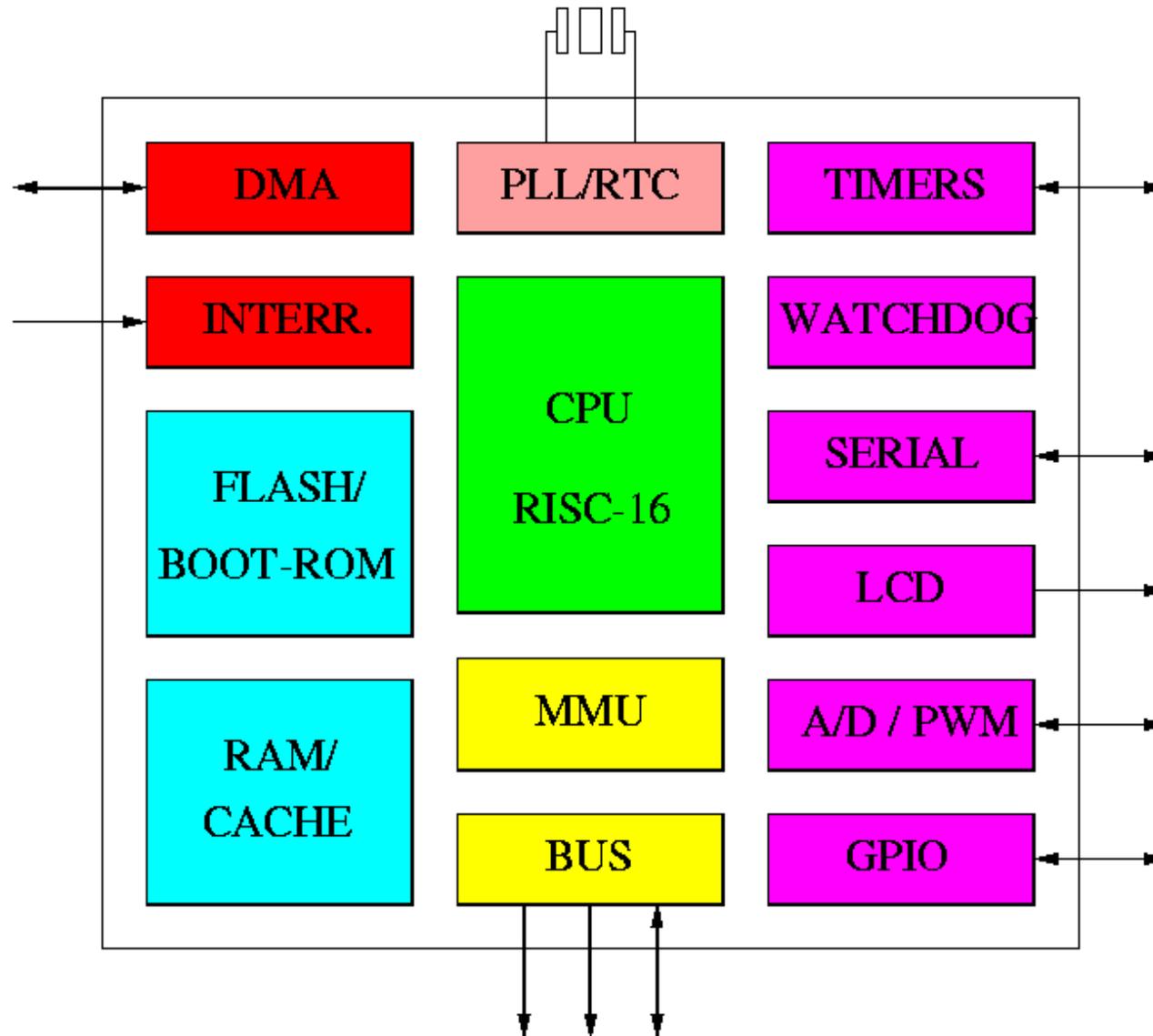
Aplicación específica. Microprocesadores.

- Máxima potencia de cálculo.
 - Buen rendimiento en todo tipo de algoritmos.
 - Direccionamiento de grandes bloques de memoria.
 - Coste elevado.
 - Alto consumo eléctrico. Necesitan refrigeración.
- Soporte de sistemas operativos avanzados.
 - Conmutación automática de tareas.
 - Protección.
 - Modos Usuario y Supervisor.
 - Memoria virtual. Paginación.
- Compatibilidad con procesadores obsoletos.
 - Emulación Hardware (Intel x86).
 - Emulación Software (Motorola 68k).

Aplicación específica. Microcontroladores.

- Bajo coste.
 - Microcontroladores de 8 y 16 bits.
 - Compatibilidad con procesadores antiguos (6502, 6800, Z80, 8086, 68000, etc).
- Sistema en un chip.
 - Integración de memoria y periféricos.
 - Conexión de memoria y periféricos externos sin lógica adicional (*Glue Logic*).
 - Múltiples variantes del microcontrolador.
 - Distintas combinaciones de memoria y/o periféricos.
 - Permiten elegir el chip que mejor se adapte a la aplicación.
- Bajo consumo de potencia.
 - Utilización en equipos portátiles. (teléfonos móviles, PDA, etc).
 - Modos de bajo consumo.
 - Frecuencia de reloj baja.

Aplicación específica. Microcontroladores avanzados.



Aplicación específica. Microcontroladores avanzados.

- Núcleos RISC.
 - SH (Hitachi), ARM-Thumb (Cirrus Logic). V85x (NEC). Colfire (Motorola).
- Código compacto (Memorias pequeñas).
 - Códigos de operación de 16 bits.
 - Los códigos de operación de 32 bits producen programas largos.
 - En 16 bits hay que restringir los campos de dirección de registro.
 - Bancos de registros pequeños. Típicamente 8 o 16 registros.
 - Instrucciones de sólo dos operandos.
 - Códigos de operación de longitud variable.
 - Palabras de extensión 16/32 bits (NEC V85x).
 - Código tipo 68K: Coldfire.
 - Ventajas del código de 16 bits.
 - Tamaño reducido de los programas. (~65% respecto de los 32 bits)
 - Buses de 16 u 8 bits. Bajo coste sin rendimientos malos.
 - Cachés más pequeñas.

Aplicación específica. Microcontroladores avanzados.

- Programación en el propio sistema.
 - Memoria Flash.
 - *Bootloader* serie.
 - Reprogramación / actualización.
 - Test del sistema.
 - Depuración.
- Unidad de gestión de memoria.
 - Paginación.
 - *Chip Selects* programables.
 - Rangos de direcciones programables.
 - Ancho de bus programable: 8 o 16 bits.
 - Estados de espera programables.
- Relojes del microcontrolador programables.
 - Síntesis PLL a partir de un único cristal.
 - Control del consumo eléctrico.

Aplicación específica. DSP.

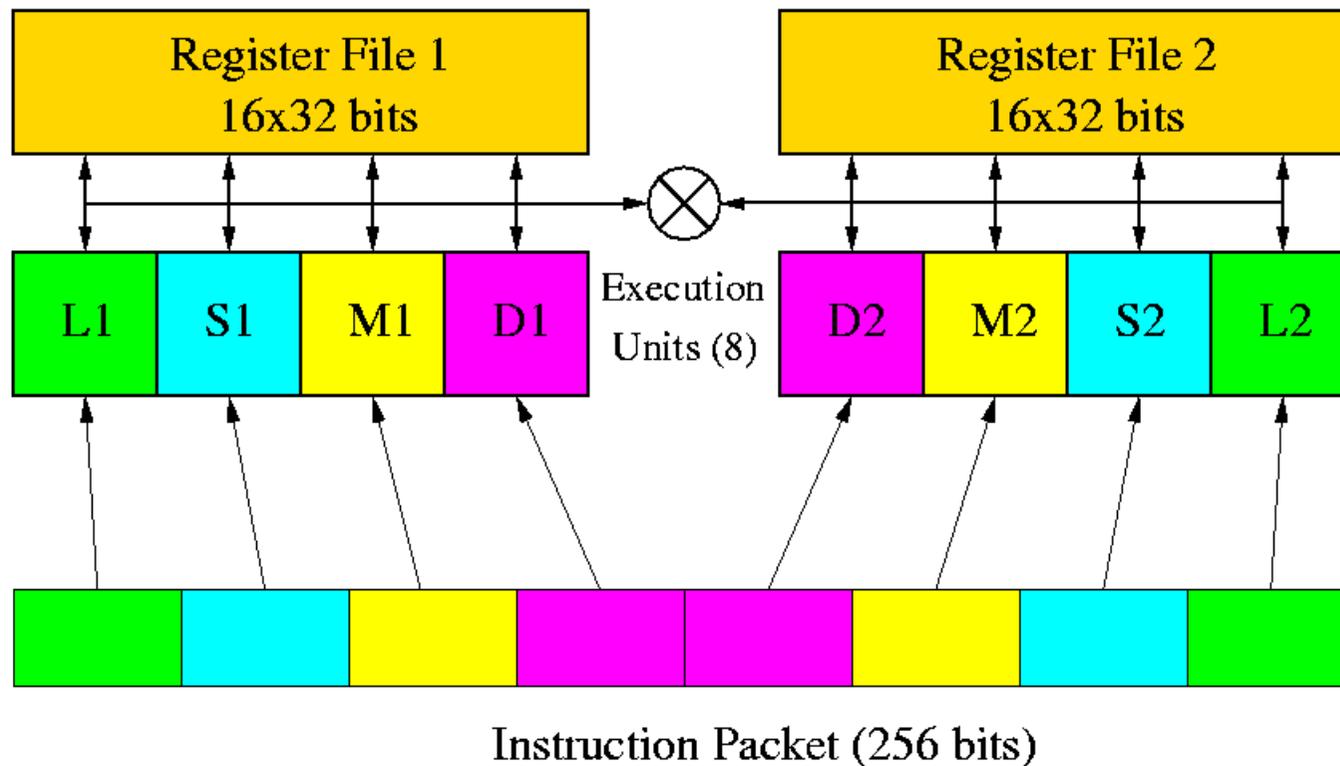
- Procesadores Digitales de Señal (DSP).
 - Algoritmos de procesamiento de señales muestreadas.
 - Operaciones aritméticas: Multiplicación y suma.
 - Paralelismo explícito.
 - Características típicas.
 - Multiplicación rápida.
 - Instrucción MAC. Paralelismo Multiplicación / Suma.
 - Desbordamiento con saturación.
 - Direccionamiento circular: Colas de datos.
 - Direccionamiento con bits permutados: FFT.
 - Arquitectura RISC.
 - Memoria y/o periféricos integrados. (microcontrolador)
 - Bajo coste por MIPS.
 - Programación en lenguaje ensamblador.

Aplicación específica. DSP avanzados.

- *VLIW (Very Long Instruction Word)*
 - TMS320C6x (Texas Instruments), SC-100 (Lucent/Motorola), TigerSHARC (Analog Devices).
 - Ejecución paralela MIMD.
 - Varias unidades de ejecución independientes.
 - Paquetes de instrucciones.
 - Cada unidad de ejecución tiene un campo en el código de operación.
 - Longitud del paquete de instrucciones :128~256 bits.
 - Simplicidad de las unidades de búsqueda y decodificación.

Aplicación específica. DSP avanzados.

TMS320C62xx VLIW DSP.

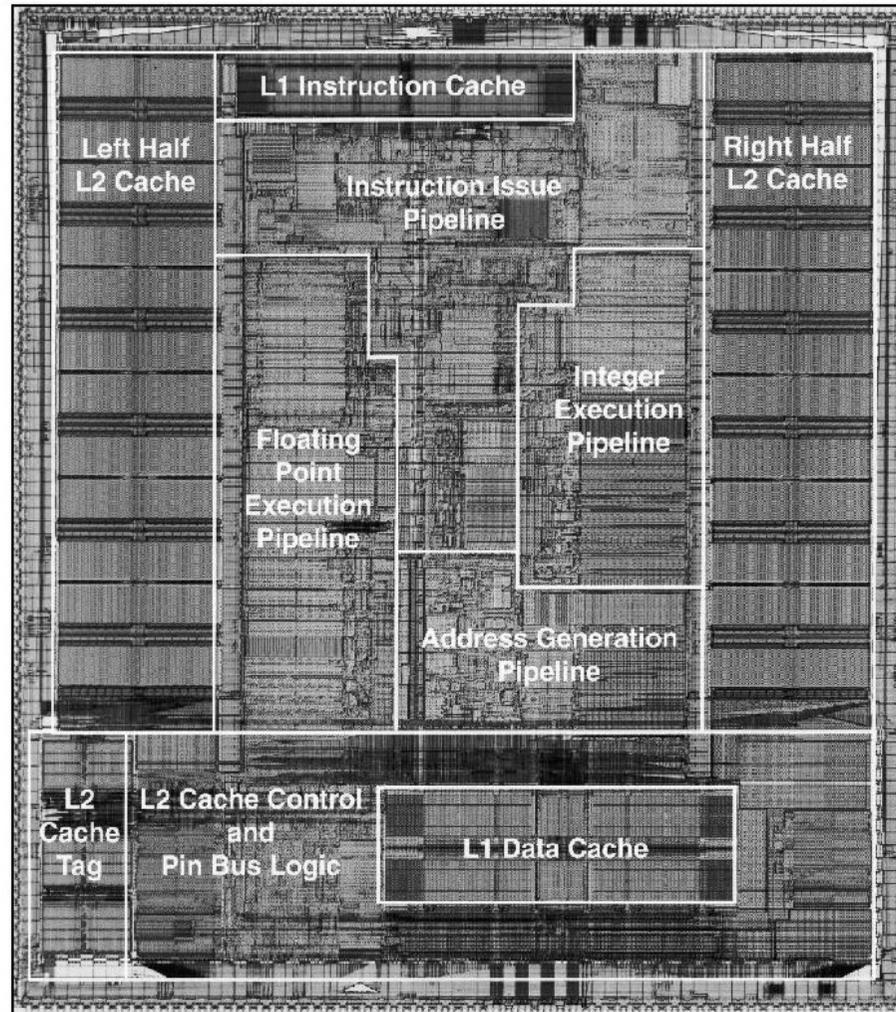


D :Data move M: Multiply. L,S : Arithmetic, Logic, Branch.

Conclusiones.

- La tecnología continúa reduciendo las dimensiones de los dispositivos.
 - Mayor complejidad de los microprocesadores.
 - Frecuencias de reloj mayores. (limitada por los retardos de interconexión)
- Uso generalizado del modelo RISC.
- Memoria caché multinivel.
- Las nuevas arquitecturas explotan el paralelismo para obtener mayor potencia de cálculo.
 - Paralelismo SIMD.
 - Paralelismo MIMD.

Evolución y tendencias futuras de los microprocesadores, microcontroladores y DSP.



Alpha 21164. Microprocesador RISC superescalar.