



Tema 2: Instrucciones. Lenguaje del computador

Unidad 3: Filosofías CISC y RISC

Rafael Casado González
Rosa María García Muñoz
María Teresa López Bonal
Universidad de Castilla–La Mancha



DEPARTAMENTO
DE SISTEMAS
INFORMÁTICOS

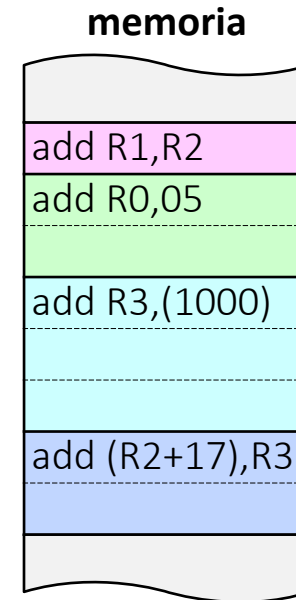


Instituto de Investigación
en informática de Albacete

Filosofía CISC

(Complex Instruction Set Computer)

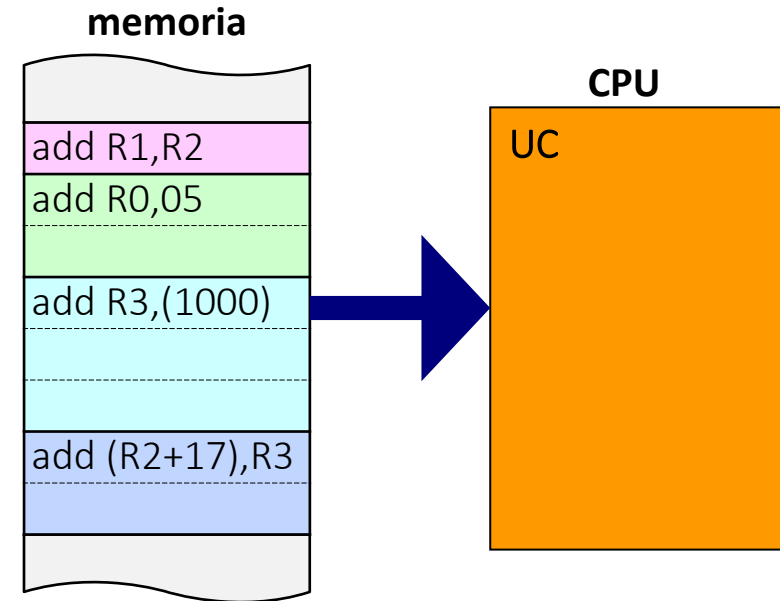
- En los años 60, se programaba fundamentalmente en lenguaje ensamblador
- Se favorecían los juegos de instrucciones ortonormales
 - Donde todas las instrucciones admiten todos los modos de direccionamiento
 - Sin casos particulares
- ¡Fáciles de aprender!



Filosofía CISC

(Complex Instruction Set Computer)

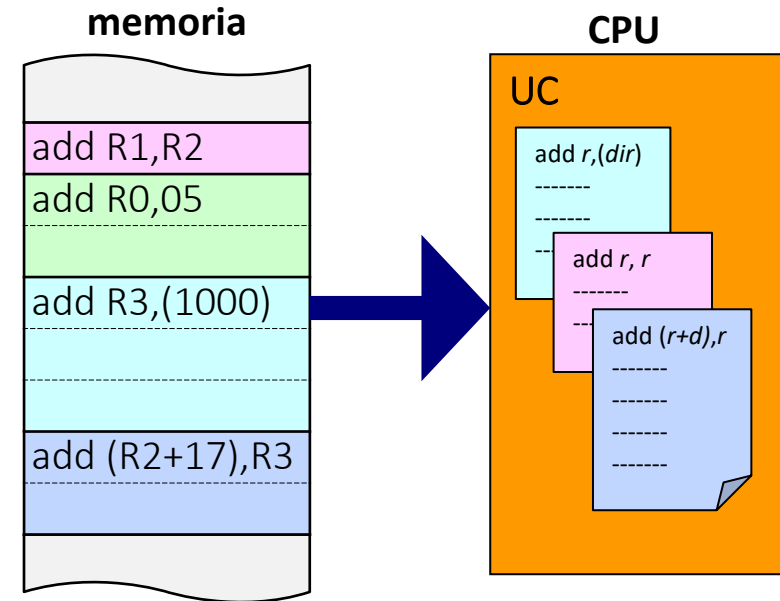
- Además, las memorias eran extremadamente lentas
- Se intentó reducir la cantidad de accesos a memoria con instrucciones muy potentes



Filosofía CISC

(Complex Instruction Set Computer)

- La CPU creció en complejidad
 - La unidad de control ejecutaba cada instrucción leída de memoria descompuesta en una serie de pasos (denominados microinstrucciones)
 - Empleando en ello múltiples ciclos de reloj

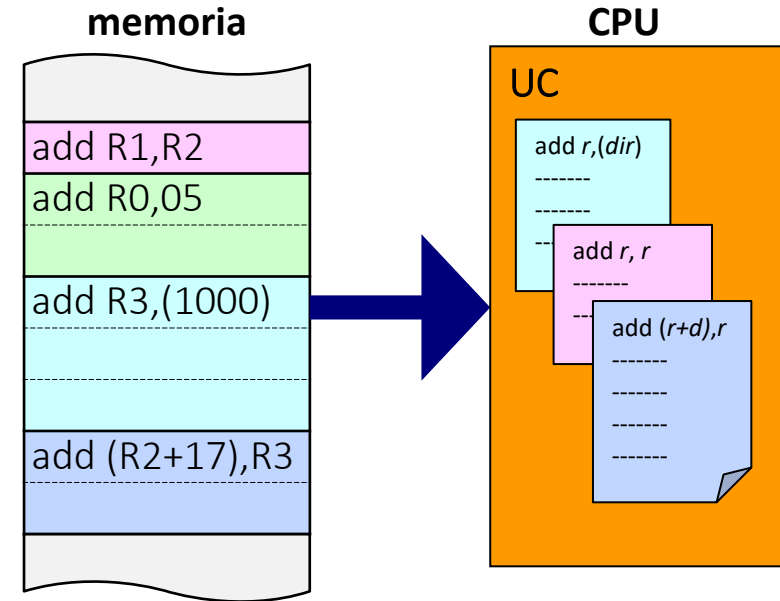


Filosofía RISC

(Reduced Instruction Set Computer)

■ Compiladores

- Posteriormente se empezó a utilizar compiladores de lenguajes de alto nivel
- Se puso de manifiesto que muchas combinaciones de instrucción / direccionamiento no se usaban casi nunca
- Pero estaban igualmente implementadas en el hardware, encareciéndolo

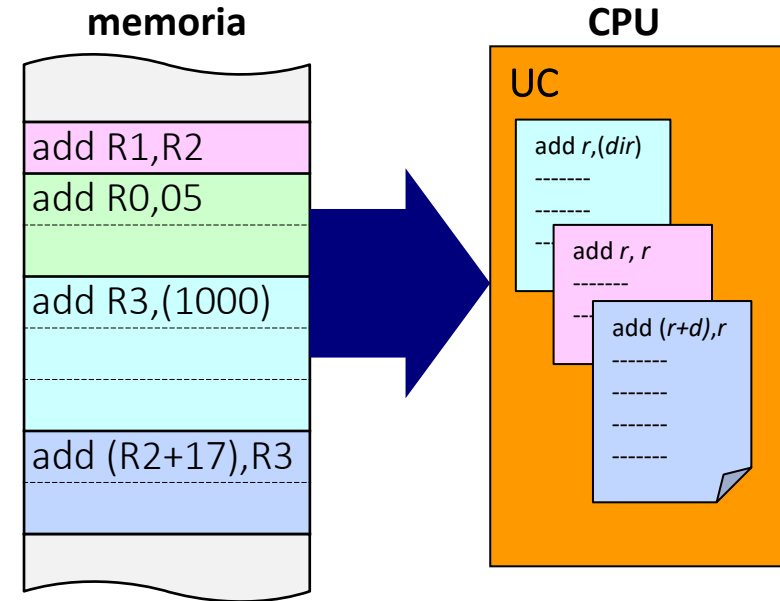


	r_1, r_2	r, dato	r, dir
add	55%	40%	5%
mult	40%	40%	20%
move	30%	30%	40%
shift	90%	8%	2%

Filosofía RISC

(Reduced Instruction Set Computer)

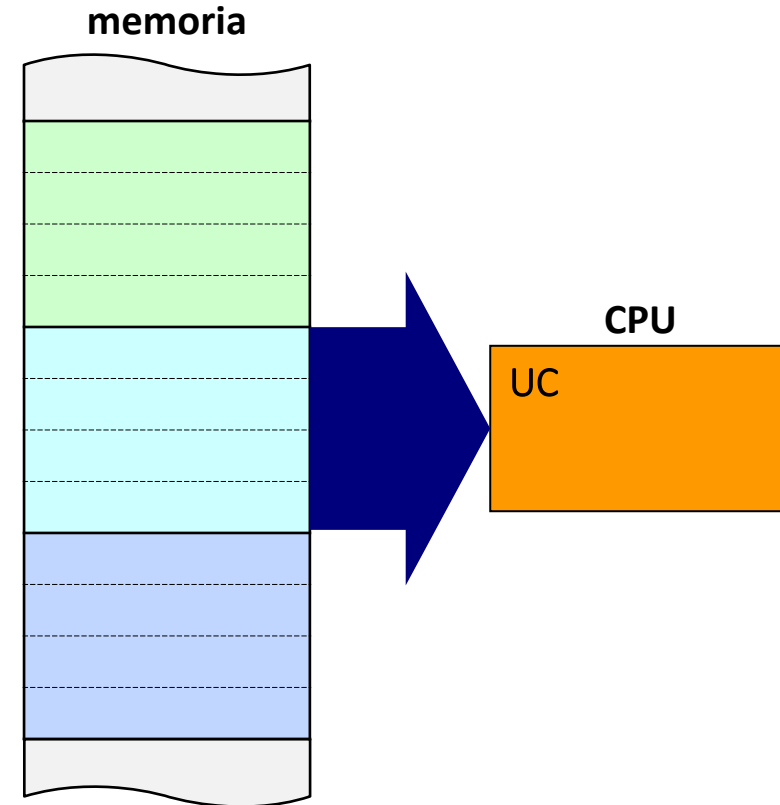
- Compiladores
- Memorias rápidas
 - Los avances tecnológicos proporcionaron memorias tan rápidas como la propia CPU



Filosofía RISC

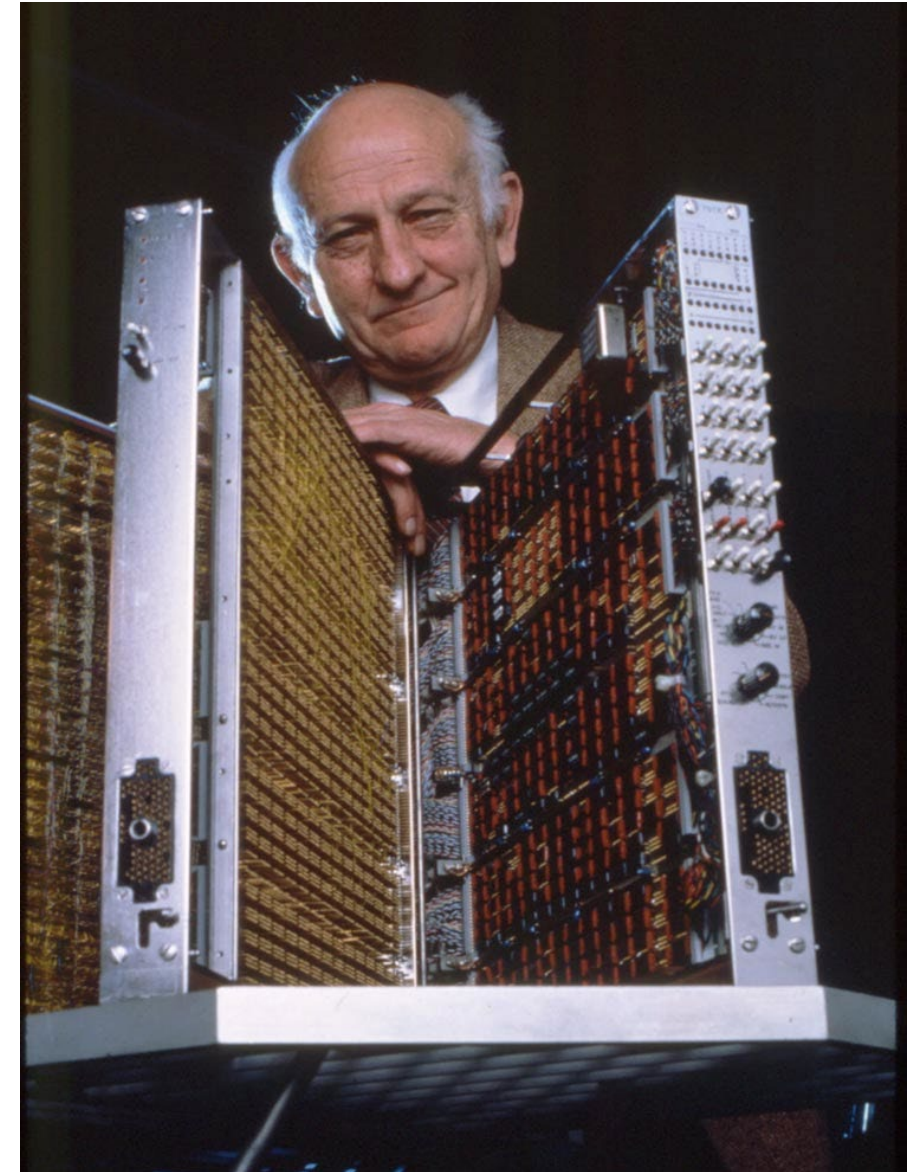
(Reduced Instruction Set Computer)

- Compiladores
- Memorias rápidas
- La evolución lógica fue
 - Guardar en la memoria
 - no la instrucción completa
 - sino directamente los pasos en los que se descompone
 - Eliminar la unidad de control microprogramada
 - Eliminar la ortogonalidad del juego de instrucciones



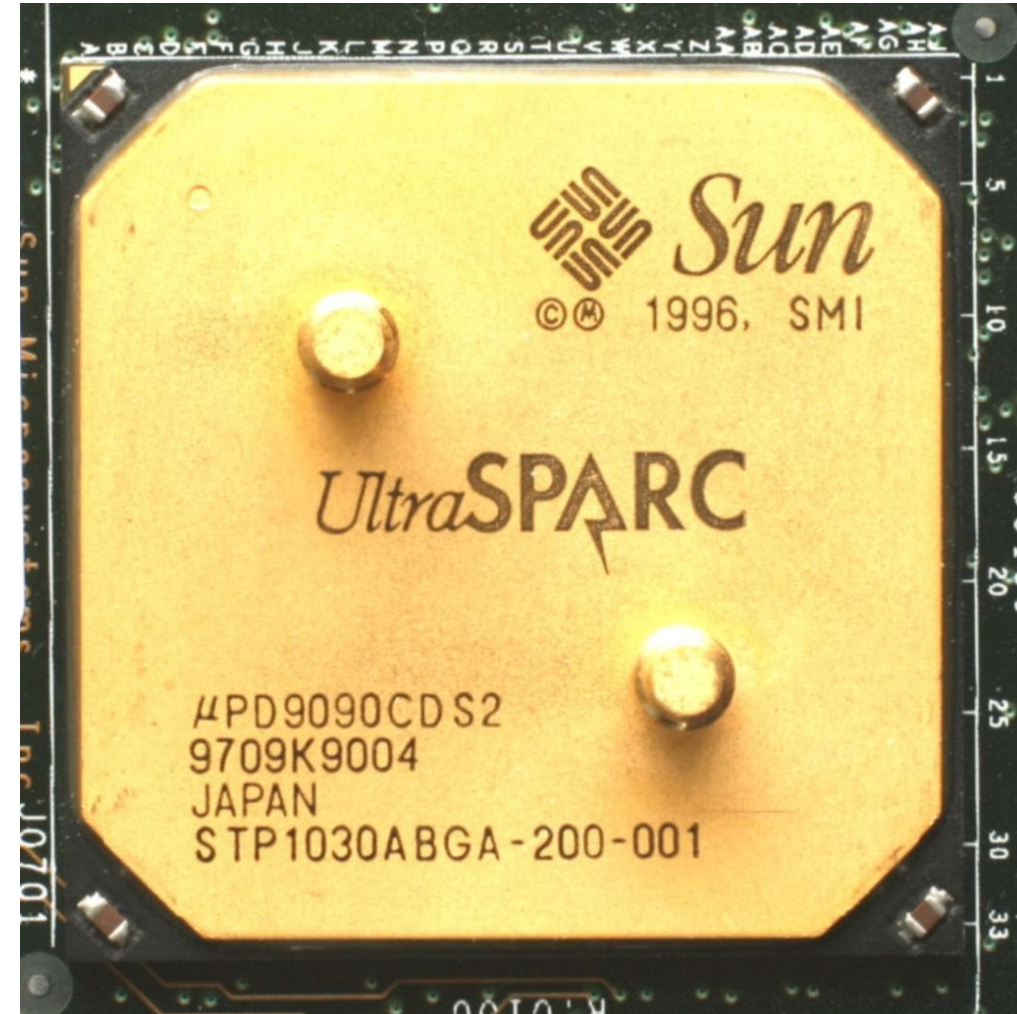
Procesadores RISC

- 1975: IBM 801
 - primer procesador RISC
- 1981: Proyecto MIPS
 - Universidad de Stanford
 - John Hennesy



Procesadores RISC

- 1982: Proyecto RISC
 - Universidad de Berkeley
 - David Patterson
- 1995: UltraSPARC
 - Sun Microsystems y Fujitsu
 - se encuentra en sus últimos modelos de servidores



Procesadores RISC

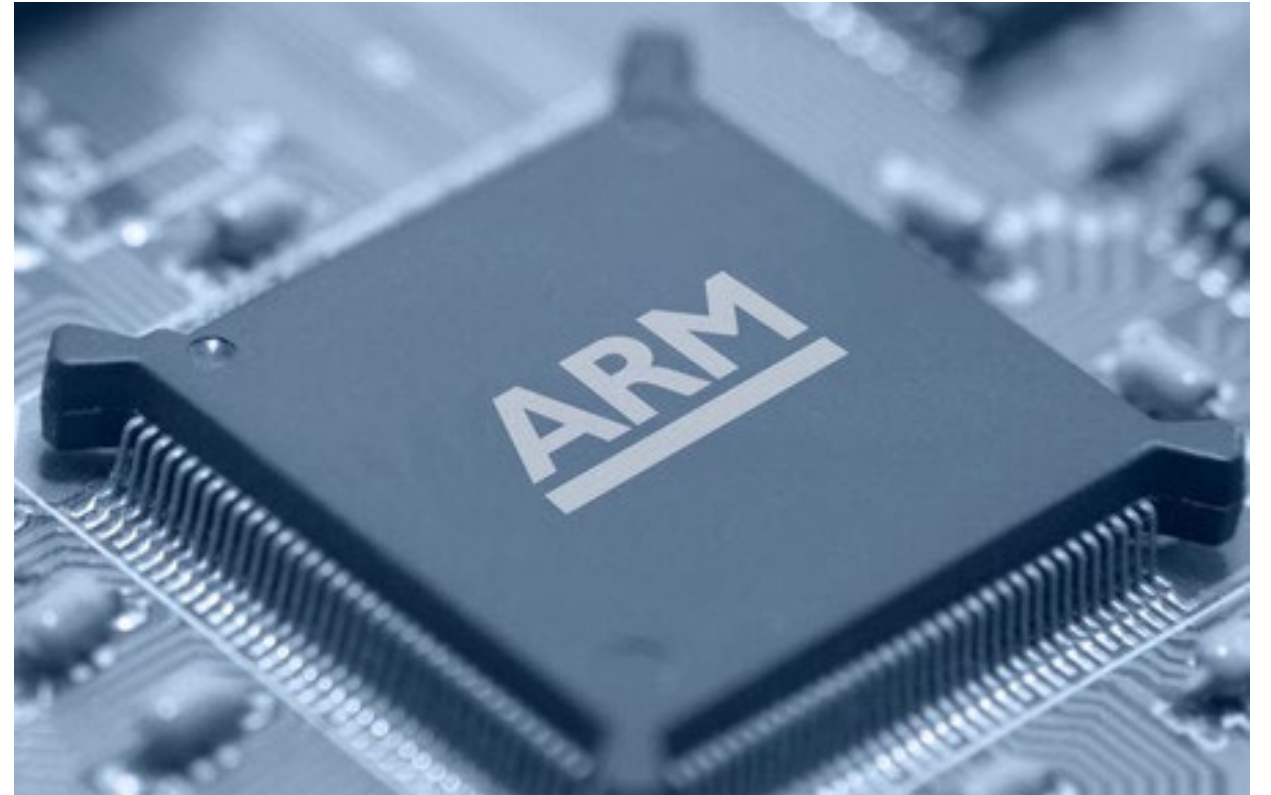
■ IBM POWER

- (Performance Optimization With Enhanced RISC)
- PowerPC (Mac)
- Gekko (Nintendo GameCube)
- Cell (PlayStation3)
- Broadway (Wii)
- Xenon (Xbox 360)



Procesadores RISC

- 1987: ARM
 - (Advanced RISC Machines)
 - Desarrollado por Acorn Computers (Acorn RISC Machines)
 - Arquitectura licenciable
 - Dispositivos PALM, Nintendo DS, Game Boy Advance, PDAs, 98% del mercado de teléfonos móviles y reproductores multimedia



Comparativa CISC / RISC

CISC	RISC
Instrucciones de longitud variable	Instrucciones de longitud fija
Cantidad variable de ciclos de reloj por instrucción	Instrucciones ejecutadas en un único ciclo de reloj
Lenguaje potente, rico en instrucciones y direccionamientos, muy ortogonal	Lenguaje restringido, con pocas instrucciones y direccionamientos, poco ortogonal (muchos casos particulares)
Contiene operaciones con modelo de ejecución memoria-memoria	Modelo de ejecución de carga y almacenamiento
Unidad de control microprogramada	Unidad de control cableada
Fácil de aprender y utilizar	Requiere el uso de compiladores sofisticados

Comparativa CISC / RISC

	CISC			RISC				
Máquina	IBM 370/168	VAX 11/780	Intel 80486	IBM 801	Berkeley RISC1	Stanford MIPS	Xerox SPARC	MIPS R4000
Año	1973	1978	1989	1975	1981	1983	1987	1991
# instrucciones	208	303	235	120	39	55	69	94
Tamaño de instrucción (bytes)	2-6	2-57	1-11	4	4	4	4	4
Modos de direccionamiento	4	22	11	1	1	1	1	1
# registros	16	16	8			32	40-520	32
Microcódigo	54KB	61KB	30KB	0	0	0	0	0
Cache (Kbytes)	64	64	8				32	128

¿CISC o RISC?

¡La eterna discusión!



La gran batalla actual es la de sus dos grandes exponentes

ARM

- ☐ Apuesta por la eficiencia energética

X86

- ☐ Alto rendimiento
- ☐ A costa de consumir bastante más energía