

# ESTRUCTURA Y DISEÑO DE COMPUTADORES

LA INTERFAZ HARDWARE / SOFTWARE

DAVID A. PATTERSON  
JOHN L. HENNESSY

Cuarta edición original

EDITORIAL REVERTÉ



# Datos de referencia de MIPS

## NÚCLEO DEL REPERTORIO DE INSTRUCCIONES

NOMBRE	MNE-MO-TÉC-NICO	FOR-MA-TO	OPERACIÓN (en verilog)	COD OP/FUNC (Hex)
Suma	add	R	$R[rd] = R[rs] + R[rt]$	1 0/20hex
Suma inmediata	addi	I	$R[rt] = R[rs] + \text{ExtSignInm}$	1,2 8hex
Suma inm. sin signo	addiu	I	$R[rt] = R[rs] + \text{ExtSignInm}$	2 9hex
Suma sin signo	addu	R	$R[rd] = R[rs] + R[rt]$	0/21hex
And	and	R	$R[rd] = R[rs] \& R[rt]$	0/24hex
And inmediato	andi	I	$R[rt] = R[rs] \& \text{ExtSignInm}$	3 chex
Salto si igual	beq	I	if( $R[rs] = R[rt]$ ) $PC = PC + 4 + \text{DirSalto}$	4 4hex
Salto si distinto	bne	I	if( $R[rs] \neq R[rt]$ ) $PC = PC + 4 + \text{DirSalto}$	4 5hex
Salto incondicional	j	J	$PC = \text{DirSaltoInc}$	5 2hex
Saltar y enlazar	jal	J	$R[31] = PC + 8; PC = \text{DirSaltoInc}$	5 3hex
Salto con registro	jr	R	$PC = R[rs]$	0/08hex
Carga de un byte sin signo	lbu	I	$R[rt] = \{24'b0, M[R[rs] + \text{ExtSignInm}(7:0)]\}$	2 24hex
Carga de media palabra sin signo	lhu	I	$R[rt] = \{16'b0, M[R[rs] + \text{ExtSignInm}(15:0)]\}$	2 25hex
Carga enlazada	ll	I	$R[rt] = M[R[rs] + \text{ExtSignInm}]$	2,7 30hex
Carga superior inm.	lui	I	$R[rt] = \{\text{inm}, 16'b0\}$	fhex
Carga de una palabra	lw	I	$R[rt] = M[R[rs] + \text{ExtSignInm}]$	2 23hex
Nor	nor	R	$R[rd] = \sim(R[rs]   R[rt])$	0/27hex
Or	or	R	$R[rd] = R[rs]   R[rt]$	0/25hex
Or inmediato	ori	I	$R[rd] = R[rs]   \text{ExtCeroInm}$	3 dhex
Fijar si menor que	slt	R	$R[rd] = (R[rs] < R[rt]) ? 1:0$	0/2ahex
Fijar si menor que inm.	slti	I	$R[rd] = (R[rs] < \text{ExtSignInm}) ? 1:0$	2 ahex
Fijar si menor que inm. sin signo	sltiu	I	$R[rd] = (R[rs] < \text{ExtSignInm}) ? 1:0$	2,6 bhex
Fijar si menor que sin signo	sltu	R	$R[rd] = (R[rs] < R[rt]) ? 1:0$	6 0/2bhex
Desplazamiento lógico a la izquierda	sll	R	$R[rd] = R[rt] \ll \text{desplaz}$	0/00hex
Desplazamiento a lógico a la derecha	srl	R	$R[rd] = R[rt] \gg \text{desplaz}$	0/02hex
Almacenamiento de un byte	sb	I	$M[R[rs] + \text{ExtSignInm}(7:0)] = R[rt](7:0)$	2 28hex
Almacenamiento condicional	sc	I	$M[R[rs] + \text{ExtSignInm}] = R[rt]; R[rt] = (\text{atomic}) ? 1:0$	2,7 38hex
Almacenamiento de media palabra	sh	I	$M[R[rs] + \text{ExtSignInm}(15:0)] = R[rt](15:0)$	2 29hex
Almacenamiento de una palabra	sw	I	$M[R[rs] + \text{ExtSignInm}] = R[rt]$	2 2bhex
Resta	sub	R	$R[rd] = R[rs] - R[rt]$	1 0/22hex
Resta sin signo	subu	R	$R[rd] = R[rs] - R[rt]$	0/23hex

- (1) Puede producirse una excepción de desbordamiento
- (2)  $\text{ExtSignInm} = \{16\{\text{inmediato}[15], \text{inmediato}\}$
- (3)  $\text{ExtCeroInm} = \{16\{1b'0\}, \text{inmediato}\}$
- (4)  $\text{DirSalto} = \{14\{\text{inmediato}[15], \text{inmediato}, 2'b0\}$
- (5)  $\text{DirSaltoCond} = \{PC+4[31:28], \text{direccion}, 2'b0\}$
- (6) Los operandos se consideran números sin signo
- (7) Pareja atómica comprobar y fijar;  $R[rt]=1$  si pareja atómica 0 si no atómica

## FORMATOS BÁSICOS DE INSTRUCCIÓN

<b>R</b>	cod oper	rs	rt	rd	desplaz	func
	31	26 25	21 20	16 15	11 10	6 5 0
<b>I</b>	cod oper	rs	rt	inmediato		
	31	26 25	21 20	16 15	0	
<b>R</b>	cod oper	dirección				
	31	26 25	0			

## NÚCLEO ARITMÉTICO DEL REPERTORIO DE INSTRUCCIONES

NOMBRE	MNE-MO-TÉC-NICO	FOR-MA-TO	OPERACIÓN	COD OP/FMT/FT/FUNC (Hex)
Salto si FP cierto	bclt	FI	If (FPcond) $PC = PC + 4 + \text{DirSalto}$	4 11/8/1/--
Salto si FP falso	bclf	FI	If (FPcond) $PC = PC + 4 + \text{DirSalto}$	4 11/8/0/--
División	div	R	$Lo = R[rs]/R[rt]; Hi = R[rs]\%R[rt]$	0/--/--1a
División sin signo	divu	R	$Lo = R[rs]/R[rt]; Hi = R[rs]\%R[rt]$	6 11/--/--1b
Suma FP simple	add.s	FR	$F[fd] = F[fs] + F[ft]$	11/10/--/0
Suma FP doble	add.d	FR	$\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} + \{F[ft], F[ft+1]\}$	11/11/--/0/
Comparación FP simple	c.x.s*	FR	$FPcond = (F[fs] op F[ft]) ? 1:0$	11/10/--/y
Comparación FP doble	c.x.d*	FR	$Fpcond = (\{F[fs], F[fs+1]\} op \{F[ft], F[ft+1]\}) ? 1:0$	11/11/--/y
*(x es eq, lt o le) (op es ==, < o <=) (y es 32, 3c o 3e)				
División FP simple	div.s	FR	$F[fd] = F[fs]/F[ft]$	11/10/--/3
División FP doble	div.d	FR	$\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} / \{F[ft], F[ft+1]\}$	11/11/--/3
Multiplicación FP simple	mul.s	FR	$F[fd] = F[fs] * F[ft]$	11/10/--/2
Multiplicación FP doble	mul.d	FR	$\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} * \{F[ft], F[ft+1]\}$	11/11/--/2
Resta FP simple	sub.s	FR	$F[fd] = F[fs] - F[ft]$	11/10/--/1
Resta FP doble	sub.d	FR	$\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} - \{F[ft], F[ft+1]\}$	11/11/--/1
Carga FP simple	lwc1	I	$F[rt] = M[R[rs] + \text{ExtSignInm}]$	2 31/--/--
Carga FP doble	ldc1	I	$F[rt] = M[R[rs] + \text{ExtSignInm}]$ $F[rt+1] = M[R[rs] + \text{ExtSignInm} + 4]$	2 35/--/--
Mover de parte alta	mghi	R	$R[rd] = Hi$	0/--/--10
Mover de parte baja	mflo	R	$R[rd] = Lo$	0/--/--12
Mover de control	mfc0	R	$R[rd] = CR[rs]$	10/0/--/0
Multiplicación	mult	R	$\{Hi, Lo\} = R[rs] * R[rt]$	0/--/--18
Multiplicación sin signo	multu	R	$\{Hi, Lo\} = R[rs] * R[rt]$	6 0/--/--19
Desplazamiento aritmético a la derecha	sra	R	$R[rd] = R[rt] \gg \gg \text{desplaz}$	0/--/--3
Almacenamiento de FP simple	swc1	I	$M[R[rs] + \text{ExtSignInm}] = F[rt]$	2 39/--/--
Almacenamiento de FP doble	sdc1	I	$M[R[rs] + \text{ExtSignInm}] = F[rt];$ $M[R[rs] + \text{ExtSignInm} + 4] = F[rt+1]$	2 3d/--/--

## FORMATOS DE INSTRUCCIÓN PUNTO FLOTANTE

<b>FR</b>	cod oper	fnt	ft	fs	fd	func
	31	26 25	21 20	16 15	11 10	6 5 0
<b>FI</b>	cod oper	fnt	ft	inmediato		
	31	26 25	21 20	16 15	0	

## REPERTORIO DE PSEUDOINSTRUCCIONES

NOMBRE	MNEMONOTÉCNICO	OPERACIÓN
Salto si menor que	blt	If ( $R[rs] < R[rt]$ ) $PC = \text{etiqueta}$
Salto si mayor que	bgt	If ( $R[rs] > R[rt]$ ) $PC = \text{etiqueta}$
Salto si menor que o igual	ble	If ( $R[rs] \leq R[rt]$ ) $PC = \text{etiqueta}$
Salto si mayor que o igual	bge	If ( $R[rs] \geq R[rt]$ ) $PC = \text{etiqueta}$
Carga inmediata	li	$R[rd] = \text{inmediato}$
Mover	move	$R[rd] = R[rs]$

## NOMBRE DE REGISTRO, NÚMERO, USO Y CONVENIO DE LLAMADA

NOMBRE	NÚMERO	USO	¿SE CONSERVA EN UNA LLAMADA?
\$zero	0	Valor constante 0	No disponible
\$at	1	Ensamblador temporal	No
\$v0-\$v1	2 - 3	Valores de resultado de funciones y evaluación de expresiones	No
\$a0-\$a3	4 - 7	Argumentos	No
\$t0-\$t7	8 - 15	Temporales	No
\$s0-\$s7	16 - 23	Temporales almacenados	Sí
\$t8-\$t9	24 - 25	Temporales	No
\$k0-\$k1	26 - 27	Reservados para el núcleo del Sistema Operativo	No
\$gp	28	Puntero global	Sí
\$sp	29	Puntero de pila	Sí
\$fp	30	Puntero de marco	Sí
\$ra	31	Dirección de retorno	Sí

**CÓDIGOS DE OPERACIÓN, CONVERSIÓN DE BASE, SÍMBOLOS ASCII** ③

MIPS cod oper (31:26)	MIPS cod oper (31:26)	(2)MIPS func (5:0)	Binario	Decimal	Hexadecimal	Carácter ASCII	Decimal	Hexadecimal	Carácter ASCII
(1) sil	add.f		00 0000	0	0	NUL	64	40	@
j	sub.f		00 0001	1	1	SOH	65	41	A
jal	mul.f		00 0010	2	2	STX	66	42	B
	div.f		00 0011	3	3	ETX	67	43	C
oq	sllv	sqr.f	00 0100	4	4	EOT	68	44	D
bne	abs.f		00 0101	5	5	BNQ	69	45	E
blez	srlv	mov.f	00 0110	6	6	ACK	70	46	F
bgtz	srav	neg.f	00 0111	7	7	BEL	71	47	G
addi	jl		00 1000	8	8	BS	72	48	H
addiu	jslr		00 1001	9	9	HT	73	49	I
slli	movz		00 1010	10	a	LF	74	4a	J
slliu	movn		00 1011	11	b	VT	75	4b	K
andi	syscall	round.w.f	00 1100	12	c	FF	76	4c	L
ori	break	trunc.w.f	00 1101	13	d	CR	77	4d	M
xori		cell.v.f	00 1110	14	e	SO	78	4e	N
lui	sync	floor.w.f	00 1111	15	f	SI	79	4f	O
(2) rafhi			01 0000	10	10	DLE	80	50	P
mthi			01 0001	17	11	DC1	81	51	Q
mflo	movz.f		01 0010	18	12	DC2	82	52	R
mtlo	movn.f		01 0011	19	13	DC3	83	53	S
			01 0100	20	14	DC4	84	54	T
			01 0101	21	15	NAK	85	55	U
			01 0110	22	16	SYN	86	56	V
			01 0111	23	17	ETB	87	57	W
mult			01 1000	24	18	CAN	88	58	X
multu			01 1001	25	19	EM	89	59	Y
div			01 1010	26	1a	SUB	90	5a	Z
divu			01 1011	27	1b	ESC	91	5b	[
			01 1100	28	1e	FS	92	5c	\
			01 1101	29	1d	GS	93	5d	]
			01 1110	30	1e	RS	94	5e	^
			01 1111	31	1f	US	95	5f	-
Ib	add	cvt.s.f	10 0000	32	20	Space	96	60	'
lh	addu	cvt.d.f	10 0001	33	21	!	97	61	a
lwl	sub		10 0010	34	22		98	62	b
lw	subu		10 0011	35	23	#	99	63	c
lbu	and	cvt.w.f	10 0100	36	24	\$	100	64	d
lhu	or		10 0101	37	25	%	101	65	e
lwr	xor		10 0110	38	26	&	102	66	f
	nor		10 0111	39	27	.	103	67	g
sb			10 1000	40	28	(	104	68	h
sh			10 1001	41	29	)	105	69	i
			10 1010	42	2a	*	106	6a	j
sw	sllt		10 1011	43	2b	+	107	6b	k
			10 1100	44	2c	,	108	6c	l
			10 1101	45	2d	-	109	6d	m
swr			10 1110	46	2e	.	110	6e	n
cache			10 1111	47	2f	/	111	6f	o
ll	tge	c.f.f	11 0000	48	30	0	112	70	p
lwc1	tgeu	c.un.f	11 0001	49	31	1	113	71	q
lwc2	tlt	c.eq.f	11 0010	50	32	2	114	72	r
pref	tltu	c.ueq.f	11 0011	51	33	3	115	73	s
	teq	c.olt.f	11 0100	52	34	4	116	74	t
	tne	c.ult.f	11 0101	53	35	5	117	75	u
ldc1		c.ole.f	11 0110	54	36	6	118	76	v
ldc2		c.uie.f	11 0111	55	37	7	119	77	w
sc		c.sf.f	11 1000	56	38	8	120	78	x
swc1		c.ngle.f	11 1001	57	39	9	121	79	y
swc2		c.seq.f	11 1010	58	3a	:	122	7a	z
		c.ngl.f	11 1011	59	3b	;	123	7b	{
		c.lt.f	11 1100	60	3c	<	124	7c	}
sdc1		c.nge.f	11 1101	61	3d	=	125	7d	}
sdc2		c.le.f	11 1110	62	3e	>	126	7e	}
		c.ngt.f	11 1111	63	3f	?	127	7f	DEL

(1) cod oper(31:26) == 0  
 (2) cod oper(31:26) == 17<sub>dec</sub> (11<sub>hex</sub>); si fnt(25:21) == 16<sub>dec</sub> (10<sub>hex</sub>) f=s (simple)  
 si fnt(52:21) == 17<sub>dec</sub> (11<sub>hex</sub>) f=d (doble)

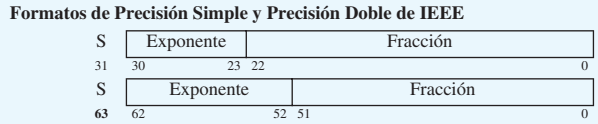
**ESTÁNDAR DE PUNTO FLOTANTE IEEE 754** ④

**Símbolos IEEE 754**

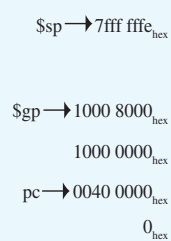
Exponente	Fracción	Objeto
0	0	±0
0	≠0	±Denormal
1 a MAX - 1	Cualquiera	±Número punto flotante
MAX	=0	±∞
MAX	≠0	NaN

donde Sesgo de Precisión Simple = 127  
 Sesgo de Precisión Doble = 1023

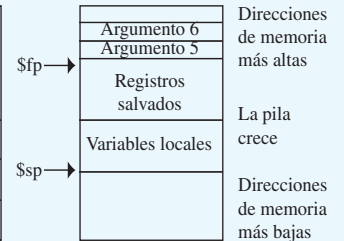
S.P.MAX = 255. D.P.MAX = 2047



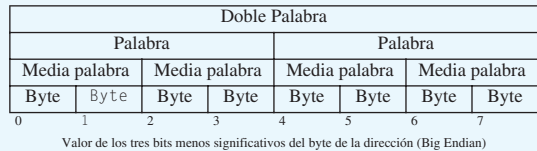
**ASIGNACIÓN DE MEMORIA**



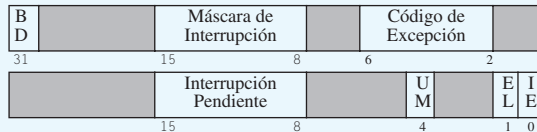
**CUADRO DE PILA**



**ALINEAMIENTO DE DATOS**



**REGISTROS DE CONTROL DE EXCEPCIONES: CAUSA Y ESTADO**



BD: Retraso de Salto (Branch Delay); UM: Modo Usuario (User Mode); EL: Nivel de Excepción (Excepción Level); IE: Habilitación de Interrupción (Interrupt Enable)

**CÓDIGOS DE EXCEPCIÓN**

Nº	Nombre	Causa de la excepción	Nº	Nombre	Causa de la excepción
0	Int	Interrupción (hardware)	9	Bp	Excepción de Punto de Ruptura
4	AdEL	Excepción de Error de Dirección (carga o búsqueda de instrucción)	10	RI	Instrucción de Excepción Reservada
5	AdES	Excepción de Error de Dirección (almacenamiento)	11	CpU	Coprocador no Implementado
6	IBE	Error de Bus en Búsqueda de Instrucción	12	Ov	Excepción de Desbordamiento Aritmético
7	DBE	Error de Bus en Carga o Almacenamiento	13	Tr	Trap
8	Sys	Excepción de Llamada al Sistema	15	FPE	Excepción de Punto Flotante

**TAMAÑOS DE PREFIJOS (10x para discos y comunicaciones; 2x para memoria)**

TAMAÑO	PREFIJO	TAMAÑO	PREFIJO	TAMAÑO	PREFIJO	TAMAÑO	PREFIJO
10 <sup>3</sup> , 2 <sup>10</sup>	Kilo-	10 <sup>15</sup> , 2 <sup>50</sup>	Peta-	10 <sup>-3</sup>	mili-	10 <sup>-15</sup>	femto-
10 <sup>6</sup> , 2 <sup>20</sup>	Mega-	10 <sup>18</sup> , 2 <sup>60</sup>	Exa-	10 <sup>-6</sup>	micro-	10 <sup>-18</sup>	atto-
10 <sup>9</sup> , 2 <sup>30</sup>	Giga-	10 <sup>21</sup> , 2 <sup>70</sup>	Zetta-	10 <sup>-9</sup>	nano-	10 <sup>-21</sup>	zepto-
10 <sup>12</sup> , 2 <sup>40</sup>	Tera-	10 <sup>24</sup> , 2 <sup>80</sup>	Yotta-	10 <sup>-12</sup>	pico-	10 <sup>-24</sup>	yocto-

El símbolo de cada prefijo es justamente su primera letra, excepto µ que se utiliza para micro.

# **Estructura y diseño de computadores**

**LA INTERFAZ SOFTWARE/HARDWARE**

**TRADUCCIÓN DE LA CUARTA EDICIÓN EN LENGUA INGLESA**



# Estructura y diseño de computadores

LA INTERFAZ SOFTWARE / HARDWARE

TRADUCCIÓN DE LA CUARTA EDICIÓN EN LENGUA INGLESA

## David A. Patterson

University of California, Berkeley

## John L. Hennessy

Stanford University

*Con contribuciones de*

Perry Alexander  
The University of Kansas

Peter J. Ashenden  
Ashenden Designs Pty Ltd

Javier Bruguera  
Universidade de Santiago de Compostela

Jichuan Chang  
Hewlett-Packard

Matthew Farrens  
University of California, Davis

David Kaeli  
Northeastern University

Nicole Kaiyan  
University of Adelaide

David Kirk  
NVIDIA

James R. Larus  
Microsoft Research

Jacob Leverich  
Hewlett-Packard

Kevin Lim  
Hewlett-Packard

John Nickolls  
NVIDIA

John Oliver  
Cal Poly, San Luis Obispo

Milos Prvulovic  
Georgia Tech

Partha Ranganathan  
Hewlett-Packard



EDITORIAL  
REVERTÉ

Barcelona - Bogotá - Buenos Aires - Caracas - México

*Título de la obra original:*

**Computer Organization and Design. The Hardware / Software Interface. Fourth Edition**

*Edición original en lengua inglesa publicada por:*

**ELSEVIER INC** of 200 Wheeler Road, 6<sup>th</sup> floor, Burlington, MA 01803, USA

Copyright © 2009 by Elsevier Inc. *All Rights Reserved*

Edición en papel

© Editorial Reverté, S. A., 2011

ISBN: 978-84-291-2620-4

Edición e-book (PDF)

© Editorial Reverté, S. A., 2018

ISBN: 978-84-291-9418-0

*Versión española por:*

**Prof. Dr. Javier Díaz Bruguera**

Catedrático de Universidad en el área de arquitectura y tecnología de computadores

Universidad de Santiago de Compostela

*Maquetación:* REVERTÉ-AGUILAR, SL

**Propiedad de:**

**EDITORIAL REVERTÉ, S. A.**

**Loreto, 13-15, Local B**

**08029 Barcelona – España**

Tel: (34) 93 419 33 36

reverte@reverte.com

www.reverte.com

Reservados todos los derechos. La reproducción total o parcial de esta obra, por cualquier medio o procedimiento, comprendidos la reprografía y el tratamiento informático, queda rigurosamente prohibida, salvo excepción prevista en la ley. Asimismo queda prohibida la distribución de ejemplares mediante alquiler o préstamo públicos, la comunicación pública y la transformación de cualquier parte de esta publicación (incluido el diseño de la cubierta) sin la previa autorización de los titulares de la propiedad intelectual y de la Editorial. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (arts. 270 y siguientes del Código Penal). El Centro Español de Derechos Reprográficos (CEDRO) vela por el respeto a los citados derechos.

*A Linda,  
que ha sido, es y siempre será el amor de mi vida*



## Elogios a *Computer Organization and Design. The Hardware / Software Interface, fourth edition*

“Patterson y Hennessy no sólo mejoran la pedagogía del material tradicional sobre procesadores segmentados y jerarquía de memoria, sino que también extienden enormemente la cobertura de los sistemas multiprocesador para incluir arquitecturas emergentes como los procesadores multinúcleo y las GPUs. La cuarta edición de *Estructura y Diseño de Computadores* es un nuevo punto de referencia con el cual deben compararse los restantes libros sobre arquitectura de computadores.”

—David A. Wood, *University of Wisconsin-Madison*

“Patterson y Hennessy han mejorado enormemente lo que ya era el estándar de referencia de los libros de texto. En un campo en continua y rápida evolución como el de la arquitectura de los computadores, han entrelazado un número impresionante de casos reales y cuestiones actuales en un marco de fundamentos suficientemente comprobados.”

—Fer Chong, *University of California at Santa Barbara*

“Desde la publicación de la primera edición en 1994, *Estructura y diseño de computadores* ha iniciado en el campo de la arquitectura de los computadores a una generación de estudiantes de informática e ingeniería. Ahora, muchos de aquellos estudiantes son los líderes en este campo. La tradición continúa en la universidad, porque los profesores utilizan las últimas ediciones del libro que los inspiró para captar a la siguiente generación. Con la cuarta edición, los lectores se preparan para la próxima era de la computación.”

—David I. August, *Princeton University*

“La nueva cobertura de los sistemas multiprocesador y el paralelismo está a la altura de este excelente clásico. Los nuevos temas se introducen de forma progresiva y bien argumentada, y además se proporcionan muchos ejemplos y detalles extraídos del hardware actual.”

—John Greiner, *Rice University*

“A medida que la arquitectura de los computadores se mueve de los monoprocesadores a los sistemas multinúcleo, los entornos de programación paralela usados para aprovecharlos serán un punto clave para el éxito de estos nuevos sistemas. En los sistemas multinúcleo, la interfaz entre el hardware y el software adquiere una importancia especial. Esta nueva edición de *Estructura y diseño de computadores* es obligatoria para cualquier estudiante que quiera comprender la arquitectura de los multinúcleo, incluida la interfaz entre la programación y la arquitectura.”

—Jesse Fang, *Director of Programming System Lab at Intel*

“La cuarta edición de *Estructura y diseño de computadores* sigue mejorando los altos estándares de calidad de las ediciones anteriores. Los nuevos contenidos, tendencias que están cambiando los ordenadores incluyendo los multinúcleo, memorias flash, GPUs, etc., hacen de esta edición un libro que todos aquellos que hemos crecido con las anteriores ediciones debemos leer.”

—Parthasarathy Ranganathan, *Principal Research Scientist, HP Labs*

## AGRADECIMIENTOS

Figuras 1.7 y 1.8. Gentileza de Other World Computing ([www.macsales.com](http://www.macsales.com)).

Figuras 1.9, 1.19 y 5.37. Gentileza de AMD.

Figura 1.10. Gentileza de Storage Technology Corp.

Figuras 1.10.1, 1.10.2 y 4.15.2. Gentileza del Charles Babbage Institute, University of Minnesota Libraries, Minneapolis.

Figuras 1.10.3, 4.15.1, 4.15.3, 5.12.3 y 6.14.2. Gentileza de IBM.

Figura 1.10.4. Gentileza de Cray Inc.

Figura 1.10.5. Gentileza de Apple Computer Inc.

Figura 1.10.6. Gentileza del Computer History Museum.

Figuras 5.12.1 y 5.12.2. Gentileza de Museum of Science, Boston.

Figura 5.12.4. Gentileza de MIPS Technologies Inc.

Figuras 6.15, 6.16 y 6.17. Gentileza de Sun Microsystems Inc.

Figura 6.4. © Peg Skorpinski.

Figura 6.14.1. Gentileza de Computer Museum of America.

Figura 6.14.3. Gentileza de Commercial Computing Museum.


Figuras 7.13.1. Gentileza de NASA Ames Research Center.

# Contenido

Prefacio xiii

## CAPÍTULOS

### **1 Abstracciones y tecnología de los computadores 2**

- 1.1 Introducción 3
- 1.2 Bajo los programas 10
- 1.3 Bajo la cubierta 13
- 1.4 Prestaciones 26
- 1.5 El muro de la potencia 39
- 1.6 El gran cambio: el paso de monoprocesadores a multiprocesadores 41
- 1.7 Casos reales: fabricación y evaluación del AMD Opteron x4 44
- 1.8 Falacias y errores habituales 51
- 1.9 Conclusiones finales 54
-  1.10 Perspectiva histórica y lecturas recomendadas 55
- 1.11 Ejercicios 56

### **2 Instrucciones: el lenguaje del computador 74**

- 2.1 Introducción 76
- 2.2 Operaciones del hardware del computador 77
- 2.3 Operandos del hardware del computador 80
- 2.4 Números con signo y sin signo 87
- 2.5 Representación de instrucciones en el computador 94
- 2.6 Operaciones lógicas 102
- 2.7 Instrucciones para la toma de decisiones 105
- 2.8 Apoyo a los procedimientos en el hardware del computador 112
- 2.9 Comunicarse con la gente 122
- 2.10 Direcciones y direccionamiento inmediato MIPS para 32 bits 128
- 2.11 Paralelismo e instrucciones: sincronización 137
- 2.12 Traducción e inicio de un programa 139
- 2.13 Un ejemplo de ordenamiento en C para verlo todo junto 149

**Nota importante:** En la presente edición en castellano, los contenidos del CD incluidos en la edición original son accesibles (en lengua inglesa) a través de la página web [www.reverte.com/microsites/pattersonhennessy](http://www.reverte.com/microsites/pattersonhennessy). Aunque en la presente edición no se proporciona un CD-ROM físico, a lo largo de todo el texto se menciona el CD y se utiliza el icono que lo representa para hacer referencia a su contenido.

- 2.14 Tablas frente a punteros 157
- 2.15 Perspectiva histórica y lecturas recomendadas 161
- 2.16 Caso real: instrucciones ARM 161
- 2.17 Casos reales: instrucciones x86 165
- 2.18 Falacias y errores habituales 174
- 2.19 Conclusiones finales 176
- 2.20 Perspectiva histórica y lecturas recomendadas 179
- 2.21 Ejercicios 179



### **3 Aritmética para computadores 222**

- 3.1 Introducción 224
- 3.2 Suma y resta 224
- 3.3 Multiplicación 230
- 3.4 División 236
- 3.5 Punto flotante 242
- 3.6 Paralelismo y aritmética del computador: asociatividad 270
- 3.7 Caso real: punto flotante en el x86 272
- 3.8 Falacias y errores habituales 275
- 3.9 Conclusiones finales 280
- 3.10 Perspectiva histórica y lecturas recomendadas 283
- 3.11 Ejercicios 283



### **4 El procesador 298**

- 4.1 Introducción 300
- 4.2 Convenios de diseño lógico 303
- 4.3 Construcción de un camino de datos 307
- 4.4 Esquema de una implementación simple 316
- 4.5 Descripción general de la segmentación 330
- 4.6 Camino de datos segmentados y control de la segmentación 344
- 4.7 Riesgos de datos: anticipación frente a bloqueos 363
- 4.8 Riesgos de control 375
- 4.9 Excepciones 384
- 4.10 Paralelismo y paralelismo a nivel de instrucciones avanzado 391
- 4.11 Casos reales: El pipeline del AMD Opteron X4 (Barcelona) 404
- 4.12 Tema avanzado: una introducción al diseño digital utilizando un lenguaje de descripción hardware para describir y modelar un pipeline y más figuras sobre segmentación 406
- 4.13 Falacias y errores habituales 407
- 4.14 Conclusiones finales 408
- 4.15 Perspectiva histórica y lecturas recomendadas 409
- 4.16 Ejercicios 409

**5****Grande y rápida: aprovechamiento de la jerarquía de memoria 450**

- 5.1 Introducción 452
- 5.2 Principios básicos de las caches 457
- 5.3 Evaluación y mejora de las prestaciones de la cache 457
- 5.4 Memoria virtual 492
- 5.5 Un marco común para las jerarquías de memoria 518
- 5.6 Máquinas virtuales 525
- 5.7 Utilización de una máquina de estados finitos para el control de una cache sencilla 529
- 5.8 Paralelismo y jerarquías de memoria: coherencia de cache 534
-  5.9 Material avanzado: implementación de controladores de cache 538
- 5.10 Casos reales: las jerarquías de memoria del AMD Opteron X4 (Barcelona) y del Intel Nehalem 539
- 5.11 Falacias y errores habituales 543
- 5.12 Conclusiones finales 547
-  5.13 Perspectiva histórica y lecturas recomendadas 548
- 5.14 Ejercicios 548

**6****Almacenamiento y otros aspectos de la E/S 568**

- 6.1 Introducción 570
- 6.2 Confiabilidad, fiabilidad y disponibilidad 573
- 6.3 Almacenamiento en disco 575
- 6.4 Almacenamiento Flash 580
- 6.5 Conexión entre procesadores, memoria y dispositivos de E/S 582
- 6.6 Interfaz de los dispositivos de E/S al procesador, memoria y sistema operativo 586
- 6.7 Medidas de las prestaciones de la E/S: ejemplos de discos y sistema de ficheros 596
- 6.8 Diseño de un sistema de E/S 598
- 6.9 Paralelismo y E/S: conjuntos redundantes de discos económicos 599
- 6.10 Casos reales: servidor Sun Fire x4150 606
-  6.11 Aspectos avanzados: redes 612
- 6.12 Falacias y errores habituales 613
- 6.13 Conclusiones finales 617
-  6.14 Perspectiva histórica y lecturas recomendadas 618
- 6.15 Ejercicios 619

**7****Multinúcleos, multiprocesadores y clústeres 630**

- 7.1 Introducción 632
- 7.2 La dificultad de crear programas de procesamiento paralelo 634
- 7.3 Multiprocesadores de memoria compartida 638
- 7.4 Clústeres y otros multiprocesadores de paso de mensajes 641

- 7.5 Ejecución multihilo en hardware 645
- 7.6 SISD, MIMD, SIMD, SPMD y procesamiento vectorial 648
- 7.7 Introducción a las unidades de procesamiento gráfico 654
- 7.8 Introducción a las topologías de redes para multiprocesadores 660
- 7.9 Programas de prueba para multiprocesadores 664
- 7.10 Roofline: un modelo de prestaciones sencillo 667
- 7.11 Casos reales: evaluación de cuatro multinúcleos con el modelo Roofline 675
- 7.12 Falacias y errores habituales 684
- 7.13 Conclusiones finales 686
- 7.14 Perspectiva histórica y lecturas recomendadas 688
- 7.15 Ejercicios 688

## APÉNDICES

### A

#### GPUs para gráficos y cálculo A-2

- A.1 Introducción A-3
- A.2 Arquitecturas del sistema de la GPU A-7
- A.3 Programación de las GPU A-12
- A.4 Arquitectura multiprocesador con ejecución multihilo A-25
- A.5 Sistema de memoria paralelo A-36
- A.6 Aritmética punto flotante A-41
- A.7 Casos reales: NVIDIA GeForce 8800 A-46
- A.8 Casos reales: Implementación de aplicaciones en la GPU A-55
- A.9 Falacias y errores habituales A-72
- A.10 Conclusiones finales A-76
- A.11 Perspectiva histórica y lecturas recomendadas A-77



### B

#### Ensambladores, enlazadores y el simulador SPIM B-2

- B.1 Introducción B-3
- B.2 Ensambladores B-10
- B.3 Enlazadores B-18
- B.4 Cargador B-19
- B.5 Utilización de la memoria B-20
- B.6 Convenio de llamada a procedimiento B-22
- B.7 Excepciones e interrupciones B-33
- B.8 Entrada y salida B-38
- B.9 SPIM B-40
- B.10 Lenguaje ensamblador MIPS R2000 B-45
- B.11 Conclusiones finales B-81
- B.12 Ejercicios B-82

## Índice I-1

**C O N T E N I D O S   E N   E L   C D   \*****Conceptos clásicos de diseño lógico C-2**

- C.1 Introducción C-3
- C.2 Puertas, tablas de verdad y ecuaciones lógicas C-4
- C.3 Lógica combinacional C-9
- C.4 Lenguajes de descripción hardware C-20
- C.5 Una unidad aritmético-lógica básica C-26
- C.6 Sumas más rápidas: acarreo adelantado C-38
- C.7 Relojes C-48
- C.8 Elementos de memoria: biestables, cerrojos y registros C-50
- C.9 Elementos de memoria: SRAM y DRAM C-58
- C.10 Máquinas de estados finitos C-67
- C.11 Metodologías de temporización C-72
- C.12 Dispositivos programables por campos C-78
- C.13 Conclusiones finales C-79
- C.14 Ejercicios C-80



**Implementación del control en hardware D-2**

- D.1 Introducción D-3
- D.2 Implementación de unidades de control combinacionales D-4
- D.3 Implementación de un control basado en máquinas de estados finitos D-8
- D.4 Implementación de la función Estado-siguiente con un secuenciador D-22
- D.5 Traducción de un microprograma a hardware D-28
- D.6 Conclusiones finales D-32
- D.7 Ejercicios D-33

**Estudio de arquitecturas RISC para ordenadores de sobremesa, servidores y sistemas empotrados E-2**

- E.1 Introducción E-3
- E.2 Modos de direccionamiento y formatos de instrucciones E-5
- E.3 Instrucciones: El subconjunto del núcleo MIPS E-9
- E.4 Instrucciones: Extensiones multimedia de los servidores y ordenadores de sobremesa RISC E-16

\* **Nota importante:** En la presente edición en castellano, los contenidos del CD incluido en la edición original (en inglés) son accesibles a través de la página web [www.reverte.com/microsites/pattersonhennessy](http://www.reverte.com/microsites/pattersonhennessy). Aunque en la presente edición no se proporciona un CD-ROM físico, a lo largo de todo el texto se menciona el CD y se utiliza el icono que lo representa para hacer referencia a su contenido.

- E.5 Instrucciones: Extensiones para procesado digital de señales de sistemas RISC empotrados E-19
  - E.6 Instrucciones: Extensiones habituales del núcleo MIPS E-20
  - E.7 Instrucciones específicas del MIPS-64 E-25
  - E.8 Instrucciones específicas del Alpha E-27
  - E.9 Instrucciones específicas del Sparc v.9 E-29
  - E.10 Instrucciones específicas del PowerPC E-32
  - E.11 Instrucciones específicas del PA-RISC 2.0 E-34
  - E.12 Instrucciones específicas del ARM E-36
  - E.13 Instrucciones específicas del Thumb E-38
  - E.14 Instrucciones específicas del SuperH E-39
  - E.15 Instrucciones específicas del M32R E-40
  - E.16 Instrucciones específicas del MIPS-16 E-40
  - E.17 Conclusiones finales E-43
-  Glosario G-1
  -  Lecturas recomendadas FR-1

# Prefacio

*Lo más bello que podemos experimentar es el misterio.  
Es la fuente de todo el arte verdadero y la ciencia.*

**Albert Einstein, *What I Believe*, 1930**

## Sobre este libro

Creemos que la enseñanza de la ciencia e ingeniería de computadores debería reflejar el estado actual de este campo, así como introducir los principios que dan forma a la computación. También opinamos que los lectores de cualquier especialidad de la computación necesitan conocer los paradigmas de organización que determinan las capacidades, prestaciones y, en definitiva, el éxito de los sistemas informáticos.

La tecnología de los computadores moderna necesita que los profesionales de todas las especialidades de la informática conozcan el hardware y el software. La interacción entre estos dos aspectos a diferentes niveles ofrece, al mismo tiempo, un entorno para la comprensión de los fundamentos de la computación. Independientemente de que su interés principal sea el hardware o el software, la informática o la electrónica, las ideas centrales de la estructura y el diseño del computador son las mismas. Por lo tanto, hemos centrado este libro en mostrar la relación entre el hardware y el software y en los conceptos básicos de los computadores actuales.

El paso reciente de los monoprocesadores a los microprocesadores multinúcleo ha confirmado la solidez de esta perspectiva, establecida ya en la primera edición. El tiempo en el que los programadores podían ignorar este aviso y confiar en los arquitectos de ordenadores, diseñadores de compiladores e ingenieros electrónicos para que sus programas se ejecutasen cada vez más rápido sin introducir cambio alguno, ha pasado. Para que los programas se ejecuten más rápido debe introducirse paralelismo. El objetivo de muchos investigadores de introducir el paralelismo sin que los programadores tengan que preocuparse de la naturaleza paralela del hardware que están programando, todavía tardará muchos años en hacerse realidad. Según nuestra visión, durante al menos la próxima década, la mayoría de los programadores van a necesitar conocer la interfaz entre el hardware y el software para que sus programas se ejecuten eficientemente en los computadores paralelos.

La audiencia de este libro incluye tanto a aquellos que, con poca experiencia en lenguaje ensamblador o diseño lógico, necesitan entender la organización básica de un computador, como a los que, con conocimientos de lenguaje ensamblador y/o diseño lógico, quieren aprender como se diseña un ordenador o entender como trabaja un sistema y por qué se comporta como lo hace.



## Sobre el otro libro

Algunos lectores seguramente están familiarizados con el libro *Arquitectura de Computadores: un Enfoque Cuantitativo*, conocido popularmente como Hennessy y Patterson. (Este libro, por el contrario, a menudo se llama Patterson y Hennessy). Nuestra motivación al escribir ese libro era describir los principios de la arquitectura de computadores haciendo uso de fundamentos sólidos de ingeniería y compromisos coste/prestaciones cuantitativos. Utilizamos un enfoque que combinaba ejemplos y medidas, basados en sistemas comerciales, para crear experiencias de diseño realistas. Nuestro objetivo fue demostrar que la arquitectura de los computadores se podía aprender con una metodología cuantitativa en lugar de un enfoque descriptivo. El libro estaba dirigido a los profesionales de la informática que querían adquirir un conocimiento detallado de los computadores.








































































La mayoría de los lectores de este libro no tienen previsto convertirse en arquitectos de computadores. Sin embargo, las prestaciones y la eficiencia energética de los sistemas software, en el futuro, dependerán drásticamente de la adecuada comprensión de las técnicas básicas del hardware por parte de los diseñadores de software. De este modo, los diseñadores de compiladores, los diseñadores de sistemas operativos, programadores de bases de datos y la mayor parte de los ingenieros del software necesitan un conocimiento firme de los principios presentados en este libro. De manera similar, los diseñadores del hardware deben comprender claramente los efectos de su trabajo sobre las aplicaciones software.






Por lo tanto, sabíamos que este libro tenía que ser mucho más que un subconjunto del material incluido en el libro *Arquitectura de Computadores*, y el material se revisó ampliamente para adecuarse a una audiencia diferente. Quedamos tan satisfechos del resultado que se revisaron las siguientes ediciones de *Arquitectura de Computadores* para eliminar la mayor parte del material de introducción; así, hay mucho menos solape entre los dos libros ahora que en las primeras ediciones.

## Cambios en la cuarta edición

Teníamos cinco objetivos principales para la cuarta edición de *Estructura y diseño de computadores*: dada la revolución multinúcleo de los microprocesadores, resaltar los aspectos paralelos del hardware y el software a lo largo del libro; racionalizar el material existente para hacer hueco a los aspectos relacionados con el paralelismo; mejorar la pedagogía en general; actualizar el contenido técnico para reflejar los cambios producidos en la industria desde la publicación de la tercera edición en 2004; restablecer la utilidad de los ejercicios en la era de internet.

Antes de discutir estos objetivos con más detalle, echemos un vistazo a la tabla de la página siguiente. Muestra los caminos del hardware y el software a lo largo del libro. Los capítulos 1, 4, 5 y 7 están en ambos caminos, sin importar la experiencia o el enfoque. El capítulo 1 es una introducción nueva que incluye una discusión sobre la importancia de la potencia y como ha alentado el paso de los sistemas con un núcleo a los microprocesadores multinúcleo. Incluye también material sobre prestaciones y evaluación mediante programas de prueba, material que en la tercera edición estaba incluido en un capítulo aparte. El capítulo 2 es más probable que esté orientado hacia el hardware, pero es una lectura esencial para los lectores orientados al software, especialmente para

Capítulo o apéndice	Secciones	Enfoque software	Enfoque hardware
1. Abstracciones y tecnología de los computadores	1.1 a 1.9		
	 1.10 (Historia)		
2. Instrucciones: el lenguaje del computador	2.1 a 2.14		
	 2.15 (Compiladores y java)		
	2.16 a 2.19		
	 2.20 (Historia)		
E. Arquitecturas de repertorio de instrucciones RISC	 E.1 a E.9		
3. Aritmética para computadores	3.1 a 3.9		
	 3.10 (Historia)		
C. Conceptos básicos de diseño lógico	 C.1 a C.13		
4. El procesador	4.1 (Visión general)		
	4.2 (Convenciones lógicas)		
	4.3 a 4.4 (Implementación simple)		
	4.5 (Introducción a segmentación)		
	4.6 (Camino de datos segmentado)		
	4.7 a 4.9 (Riesgos, excepciones)		
	4.10 a 4.11 (Paralelo, caso real)		
	 4.12 (Control segmentado en Verilog)		
	4.13 a 4.14 (Falacias)		
	 4.15 (Historia)		
D. Implementación del control en hardware	 D.1 a D.6		
5. Grande y rápida: aprovechamiento de la jerarquía de memoria	5.1 a 5.8		
	 5.9 (Controlador de cache en Verilog)		
	5.10 a 5.12		
	 5.13 (Historia)		
6. Almacenamiento y otros aspectos de la E/S	6.1 a 6.10		
	 6.11 (Redes)		
	6.12 a 6.13		
	 6.14 (Historia)		
7. Multinúcleo, multiprocesadores y clústeres	7.1 a 7.3		
	 7.14 (Historia)		
A. Unidades de procesamiento gráfico	A.1 a A.12		
B. Ensambladores, enlazadores y el simulador SPIM	B.1 a B.12		

Leer detenidamente    
 Leer si se dispone de tiempo    
 Referencia   
 Revisar o leer    
 Leer por cultura 

aqueños interesados en aprender más sobre compiladores y lenguajes de programación orientada a objetos. Incluye material del capítulo 3 de la tercera edición, de forma que toda la arquitectura MIPS, excepto las instrucciones de punto flotante, se describe ahora en un único capítulo. El capítulo 3 está dirigido a los lectores interesados en el diseño de un camino de datos o en aprender algo más sobre aritmética de punto flotante. Sin embargo, algunos lectores podrán saltarse este capítulo, bien porque no lo necesitan, bien porque es una revisión. El capítulo 4, donde se explica el procesador segmentado, es una combinación de dos capítulos de la tercera edición. Las secciones 4.1, 4.5 y 4.10 proporcionan una visión general para aquellos lectores interesados en los aspectos software. Sin embargo, constituye el material central para los interesados en los aspectos hardware; dependiendo de sus conocimientos previos, estos lectores pueden necesitar o querer leer antes el apéndice C sobre diseño lógico. El capítulo 6, sobre sistemas de almacenamiento, es de importancia crucial para los lectores con un enfoque software y los restantes lectores deberían leerlo también si disponen de tiempo suficiente. El último capítulo sobre multinúcleos, multiprocesadores y clústeres es mayoritariamente material nuevo y debería ser leído por todos.

El primer objetivo fue hacer del paralelismo un ciudadano de primera clase en esta edición, cuando en la anterior edición era un capítulo aparte en el CD. El ejemplo más evidente es el capítulo 7. En este capítulo se introduce el modelo Roofline para la evaluación de las prestaciones y se muestra su validez con la evaluación de cuatro microprocesadores multinúcleo recientes con dos núcleos computacionales. Podría demostrarse que este modelo puede ser tan intuitivo para la evaluación de los multinúcleos como el modelo de las 3Cs para las caches.

Dada la importancia del paralelismo, no sería inteligente esperar hasta el último capítulo para abordarlo, por lo tanto hay una sección sobre paralelismo en cada uno de los seis primeros capítulos:

- *Capítulo 1: Paralelismo y potencia.* Se muestra como los límites impuestos por el consumo de potencia han forzado a los principales fabricantes a mirar hacia el paralelismo, y como el paralelismo ayuda en este problema.
- *Capítulo 2: Paralelismo e instrucciones: Sincronización.* Esta sección analiza los bloqueos para variables compartidas, de forma específica las instrucciones MIPS carga enlazada y almacenamiento condicional.
- *Capítulo 3. Paralelismo y aritmética del computador: Asociatividad punto flotante.* En esta sección se analizan los retos en precisión numérica y en las operaciones punto flotante.
- *Capítulo 4. Paralelismo y paralelismo a nivel de instrucciones avanzado.* Trata el paralelismo a nivel de instrucciones (ILP) avanzado —superescalares, especulación, VLIW, desenrollamiento de lazos y OOO— así como la relación entre la profundidad del procesador segmentado y el consumo de potencia.
- *Capítulo 5. Paralelismo y jerarquías de memoria: Coherencia de cache.* Introduce los protocolos de coherencia, consistencia y fisgoneo (*snooping*) de cache.
- *Capítulo 6. Paralelismo y E/S: Conjuntos redundantes de discos económicos (RAID).* Describe los sistemas RAID como un sistema paralelo de E/S, así como un sistema ICO de alta disponibilidad.

El capítulo 7 concluye con varias razones para ser optimista con respecto a que esta incursión en el paralelismo debería tener más éxito que las que se realizaron anteriormente.

Particularmente, estoy entusiasmado con la incorporación de un apéndice sobre Unidades de Procesamiento Gráfico escrito por el científico jefe de NVIDIA, David Kirk, y el líder del grupo de arquitectos, John Nicolls. En este sentido, el apéndice A es la primera descripción detallada de una GPU, un nuevo e interesante empuje para la arquitectura de los computadores. El apéndice articula los temas paralelos de esta edición para presentar un estilo de computación que permite al programador pensar en un sistema MIMD aunque el hardware ejecute los programas como un SIMD cuando es posible. Como las GPUs son baratas y están disponibles en casi cualquier sistema—incluso en computadores portátiles— y sus entornos de programación son de acceso libre, proporciona una plataforma hardware paralela con la que experimentar.

El segundo objetivo ha sido racionalizar el libro para hacer sitio a nuevo material sobre paralelismo. El primer paso fue mirar con lupa todos los párrafos presentes en las tres ediciones anteriores para determinar si seguían siendo necesarios. Los cambios de grano grueso fueron la combinación de capítulos y la eliminación de temas. Mark Hill sugirió eliminar la implementación multiciclo de los procesadores y añadir, en su lugar, un controlador de cache multiciclo en el capítulo dedicado a la jerarquía de memoria. Este cambio permitió presentar el procesador en un único capítulo en lugar de en los dos capítulos de ediciones anteriores, mejorando el material sobre el procesador por omisión. Los contenidos del capítulo sobre prestaciones de la tercera edición se añadieron al primer capítulo.

El tercer objetivo fue mejorar los aspectos pedagógicos del libro. El capítulo 1 es ahora más jugoso, e incluye material sobre prestaciones, circuitos integrados y consumo de potencia, y crea el marco para el resto del libro. Los capítulos 2 y 3 originalmente estaban escritos con un estilo evolutivo, comenzando con una arquitectura sencilla y terminando con la arquitectura completa del MIPS al final del capítulo 3. Sin embargo, este estilo pausado no le gusta a los lectores modernos. Por lo tanto, en esta edición se junta toda la descripción del repertorio de instrucciones enteras en el capítulo 2—haciendo el capítulo 3 opcional para muchos lectores— y ahora cada sección es independiente, de modo que el lector no necesita ya leer todas las secciones anteriores. Así, el capítulo 2 es mejor como referencia ahora que en ediciones anteriores. El capítulo 4 está mejor organizado ahora porque se dedica un único capítulo al procesador, ya que la implementación multiciclo no es de utilidad actualmente. El capítulo 5 tiene una nueva sección sobre diseño de controladores de cache y el código Verilog para este controlador se incluye en una sección del CD.<sup>1</sup>

El CD-ROM que acompaña al libro, que se introdujo en la tercera edición, nos permitió reducir el coste del libro, que tenía menos páginas, y profundizar en algunos temas que eran de interés sólo para algunos lectores, pero no para todos. Desafortunadamente,

---

<sup>1</sup>**Nota importante:** En la presente edición en castellano, los contenidos del CD incluido en la edición original (en inglés) son accesibles a través de la página web [www.reverte.com/microsites/pattersonhennessy](http://www.reverte.com/microsites/pattersonhennessy). Aunque en la presente edición no se proporciona un CD-ROM físico, a lo largo de todo el texto se menciona el CD y se utiliza el icono que lo representa para hacer referencia a su contenido.

como consecuencia de nuestro entusiasmo por reducir el número de páginas del libro, el lector tenía que ir del libro al CD y del CD al libro más veces de las que le gustaría. Esto no ocurre en esta edición. Cada capítulo tiene la sección de Perspectivas históricas en el CD y, además, todos los ejercicios están en el libro impreso. Por lo tanto, en esta edición será muy raro que el lector tenga que ir del libro al CD y del CD al libro.

Para aquellos que se sorprenden de que incluyamos un CD-ROM con el libro, la respuesta es sencilla: el CD contiene material que hemos considerado que debería ser de acceso fácil e inmediato sin importar el lugar en el que se encuentre el lector. Si usted está interesado en los contenidos avanzados o le gustaría leer una guía práctica de VHDL (por ejemplo), está en el CD, lista para que la utilice. El CD-ROM incluye también una característica que debería mejorar significativamente el acceso al material: un motor de búsqueda que permite hacer búsquedas de cualquier cadena de texto en el CD o en el libro. Si está buscando algún contenido que podría no estar incluido en el índice del libro, simplemente indicando el texto que se quiere buscar el motor devuelve el número de la página en la que aparece ese texto. Esta característica es muy útil y esperamos que usted haga uso de ella frecuentemente.

Este es un campo en continua evolución, y como ocurre siempre que se lanza una nueva edición, uno de los objetivos prioritarios es actualizar los contenidos técnicos. El procesador Opteron X4 2356 (nombre en clave “Barcelona”) de AMD se utiliza como ejemplo guía a lo largo de todo el libro, concretamente para los capítulos 1, 4, 5 y 7. Los capítulos 1 y 6 añaden los resultados obtenidos con los programa de prueba para la evaluación del consumo de potencia de SPEC. El capítulo 2 incorpora una sección sobre la arquitectura ARM, que actualmente es la ISA (arquitectura del repertorio de instrucciones) de 32 bits más popular en todo el mundo. El capítulo 5 tiene una sección sobre Máquinas Virtuales, que están adquiriendo una importancia creciente. En el capítulo 5 se analizan detalladamente las medidas de rendimiento de la cache en el microprocesador multinúcleo Opteron X4 y se proporcionan algunos datos, menos detallados, sobre su rival, el procesador Nehalem de Intel, que no será anunciando antes de que se publique esta edición. En el capítulo 6 se describen por primera vez las memorias Flash así como un destacado servidor de Sun, que empaqueta 8 núcleos, 16 DIMMs y 8 discos en una única unidad 1U. También se incluyen resultados recientes sobre fallos de discos a largo plazo. El capítulo 7 cubre un gran número de temas relacionados con el paralelismo —ejecución multihilo, SIMD, procesamiento vectorial, modelos de prestaciones, programas de prueba, redes de multiprocesadores, entre otros— y describe tres microprocesadores multinúcleo, además del Opteron X4: Intel Xeon e5345 (Clovertown), IBM Cell QS20 y Sun Microsystems T2 5120 (Niagara).

El objetivo final fue intentar que los ejercicios fuesen útiles a los profesores en esta era de Internet, porque los ejercicios que los estudiantes tienen que hacer en su propia casa han sido desde hace mucho tiempo una forma importante de aprender el material contenido en el libro. Desafortunadamente, hoy en día las soluciones se cuelgan en la web casi al mismo tiempo que el libro sale a la venta. Tenemos una propuesta en dos partes. Primero, colaboradores expertos han desarrollado ejercicios totalmente nuevos para todos los capítulos del libro. Segundo, la mayoría de los ejercicios tiene una descripción cualitativa apoyada en una tabla que proporciona varios parámetros cuantitativos alternativos que son necesarios para resolver las preguntas. El elevado número de ejercicios

unidos a la flexibilidad que tiene el profesor para introducir variaciones hará difícil que el estudiante pueda encontrar las soluciones en la red. Los profesores podrán cambiar estos parámetros cuantitativos como quieran, haciendo que aquellos estudiantes que confiaban en internet para encontrar las soluciones de un conjunto de ejercicios estático y fijo se sientan frustrados. Creemos que este enfoque es un nuevo valor añadido al libro; por favor, le pedimos que nos haga saber si este enfoque funciona correctamente, ¡tanto como estudiante o como profesor!

Hemos mantenido elementos útiles de las anteriores ediciones del libro. Para conseguir que el libro se mejore como libro de referencia, mantenemos las definiciones de los nuevos términos en los márgenes de las páginas en las que aparecen por primera vez. Los elementos del libro que hemos llamado secciones “Comprender las prestaciones de los programas” ayudan al lector a comprender las prestaciones de sus programas y como mejorarlas, del mismo modo que los elementos llamados “Interfaz Hardware/Software” ayudan a entender las soluciones de compromiso que se adoptan en esta interfaz. Las secciones “Idea clave” se han mantenido para que el lector vea el bosque a pesar de los árboles. Las secciones “Autoevaluación” ayudan a confirmar la comprensión del material estudiado, ya que las respuestas se incluyen al final de cada capítulo. Esta edición incluye también la tarjeta de referencia MIPS, inspirada en la “Tarjeta Verde” del IBM System/360. Esta tarjeta ha sido actualizada y debería ser una referencia accesible al escribir programas en lenguaje ensamblador MIPS.

## Apoyo a los profesores

Hemos recopilado una gran cantidad de material de ayuda a los profesores que utilicen este libro en sus cursos. Tienen a su disposición soluciones a los ejercicios, puntos críticos de cada capítulo, figuras del libro, notas, dispositivas, etc., en la web de los editores

[www.reverte.com/microsites/pattersonhennessy](http://www.reverte.com/microsites/pattersonhennessy)

## Conclusiones finales

Si usted lee la sección de agradecimientos que viene a continuación, verá que hemos hecho un gran esfuerzo para corregir los posibles errores. Dado que un libro pasa por muchas etapas, hemos tenido la oportunidad de hacer aún más correcciones. Si usted descubre algún error que se nos ha resistido, por favor contacte con el editor a través de correo electrónico en la dirección [produccion@reverte.com](mailto:produccion@reverte.com) o por correo ordinario a la dirección de la página de *copyright*.

Esta edición marca una ruptura en la colaboración entre Hennessy y Patterson, que empezó en 1989 y se mantuvo durante muchos años. La oportunidad de dirigir una de las mayores universidades del mundo implica que el Rector Hennessy no podrá asumir en adelante la responsabilidad de crear una nueva edición. El otro autor se siente como un malabarista que siempre ha trabajado con un socio y de repente es empujado a actuar en solitario. En consecuencia, todos aquellos incluidos en los agradecimientos y los colegas de Berkeley han jugado un papel aun más importante en darle forma a los contenidos de este libro. Sin embargo, esta vez hay un único autor al que culpar por el nuevo material que va a leer.

## Agradecimientos de la cuarta edición

Me gustaría dar las gracias a **David Kirk**, **John Nicholls** y sus colegas de NVIDIA (Michael Garland, John Montrym, Dough Voorhies, Lars Nyland, Erik Lindholm, Paulius Micikevicius, Massimiliano Fatica, Stuart Oberman y Vasily Volkov) por escribir el primer apéndice que analiza en profundidad las GPUs. Quiero expresar otra vez mi agradecimiento a **Jim Larus** de Microsoft Research por su buena disposición a contribuir como experto en programación en lenguaje ensamblador y por alentar a los lectores de este libro a utilizar el simulador que él mismo desarrolló y mantiene.

También estoy muy agradecido por las contribuciones de muchos expertos que desarrollaron los nuevos ejercicios de esta edición. Escribir buenos ejercicios no es una tarea fácil y todos los colaboradores tuvieron que trabajar mucho y duro para plantear unos problemas exigentes y atractivos.

- *Capítulo 1*: **Javier Bruguera** (Universidade de Santiago de Compostela)
- *Capítulo 2*: **John Oliver** (Cal Poly, San Luis Obispo), con contribuciones de **Nicole Kaiyan** (University of Adelaide) y **Milos Prvulovic** (Georgia Tech)
- *Capítulo 3*: **Matthew Farrens** (University of California, Davis)
- *Capítulo 4*: **Milos Prvulovic** (Georgia Tech)
- *Capítulo 5*: **Jichuan Chang**, **Jacob Leverich**, **Kevin Lim** y **Partha Ranganathan** (todos de Hewlett-Packard), con contribuciones de Nicole Kaiyan (University of Adelaide)
- *Capítulo 6*: **Perry Alexander** (The University of Kansas)
- *Capítulo 7*: **David Kaeli** (Northeastern University)

**Peter Asheden** llevó a cabo la hercúlea tarea de editar y evaluar *todos* los nuevos ejercicios. Además, incluso asumió la elaboración del CD que acompaña al libro y de las diapositivas para las clases.

Gracias a **David August** y **Prakash Prabhu** de Princeton University por su trabajo sobre los concursos de los capítulos que están disponibles para los profesores en la Web de los editores.

Conté con el apoyo de mis colegas del Silicon Valley para la elaboración de gran parte del material técnico del libro:

- **AMD**, por los detalles y medidas del Opteron X4 (Barcelona): **William Brantley**, **Vasileios Liaskovitis**, **Chuck Moore** y **Brian Waldecker**.
- **Intel**, por la información prelanzamiento del Intel Nehalem: **Faye Briggs**.
- **Micron**, por la información sobre memorias Flash del capítulo 6: **Dean Klein**.
- **Sun Microsystems**, por las medidas de mezclas de instrucciones de los programas de prueba SPEC2006 en el capítulo 2 y los detalles y medidas del Sun Server x4150 en el capítulo 6: **Yan Fisher**, **John Fowler**, **Darryl Gove**, **Paul Joyce**, **Shenik Mehta**, **Pierre Reynes**, **Dimitry Stuve**, **Durgam Vahia** y **David Weaver**.

- **U.C. Berkeley, Krste Asanovic** (que proporcionó la idea para la concurrencia software frente a paralelismo hardware del capítulo 7), **James Demmel** y **Velvel Kahan** (con sus comentarios sobre paralelismo y cálculos punto flotante), **Zhangxi Tan** (que diseñó en controlador de cache y escribió el código Verilog para el mismo en el capítulo 5), **Sam Williams** (que proporcionó el modelo Roofline y las medidas para los multinúcleos en el capítulo 7) y el resto de mis colegas en el **Par Lab** que hicieron sugerencias y comentarios para los distintos aspectos del paralelismo que se pueden encontrar en este libro.

Estoy agradecido a los muchos profesores que contestaron los cuestionarios enviados por los editores, revisaron nuestras propuestas y asistieron a los grupos de discusión para analizar y responder a nuestros planes para esta edición. Esto incluye a las siguientes personas: *Grupo de discusión*: Mark Hill (University of Wisconsin, Madison), E.J. Kim (Texas A&M University), Jihong Kim (Seoul National University), Lu Peng (Louisiana State University), Dean Tullsen (UC San Diego), Ken Vollmar (Missouri State University), David Wood (University of Wisconsin, Madison), Ki Hwan Yum (University of Texas, San Antonio); *Revisiones*: Mahmoud Abou-Nasr (Wayne State University), Perry Alexander (The University of Kansas), Hakan Aydin (George Mason University), Hussein Badr (State University of New York at Stony Brook), Mac Baker (Virginia Military Institute), Ron Barnes (George Mason University), Douglas Blough (Georgia Institute of Technology), Kevin Bolding (Seattle Pacific University), Miodrag Bolic (University of Ottawa), John Bonomo (Westminster College), Jeff Braun (Montana Tech), Tom Briggs (Shippensburg University), Scott Burgess (Humboldt State University), Fazli Can (Bilkent University), Warren R. Carithers (Rochester Institute of Technology), Bruce Carlton (Mesa Community College), Nicholas Carter (University of Illinois at Urbana-Champaign), Anthony Cocchi (The City University of New York), Don Cooley (Utah State University), Robert D. Cupper (Allegheny College), Edward W. Davis (North Carolina State University), Nathaniel J. Davis (Air Force Institute of Technology), Molisa Derk (Oklahoma City University), Derek Eager (University of Saskatchewan), Ernest Ferguson (Northwest Missouri State University), Rhonda Kay Gaede (The University of Alabama), Etienne M. Gagnon (UQAM), Costa Gerousis (Christopher Newport University), Paul Gillard (Memorial University of Newfoundland), Michael Goldweber (Xavier University), Georgia Grant (College of San Mateo), Merrill Hall (The Master's College), Tyson Hall (Southern Adventist University), Ed Harcourt (Lawrence University), Justin E. Harlow (University of South Florida), Paul F. Hemler (Hampden-Sydney College), Martin Herbordt (Boston University), Steve J. Hodges (Cabrillo College), Kenneth Hopkinson (Cornell University), Dalton Hunkins (St. Bonaventure University), Baback Izadi (State University of New York—New Paltz), Reza Jafari, Robert W. Johnson (Colorado Technical University), Bharat Joshi (University of North Carolina, Charlotte), Nagarajan Kandasamy (Drexel University), Rajiv Kapadia, Ryan Kastner (University of California, Santa Barbara), Jim Kirk (Union University), Geoffrey S. Knauth (Lycoming College), Manish M. Kochhal (Wayne State), Suzan Koknar-Tezel (Saint Joseph's University), Angkul Kongmunvattana (Columbus State University), April Kontostathis (Ursinus



College), Christos Kozyrakis (Stanford University), Danny Krizanc (Wesleyan University), Ashok Kumar, S. Kumar (The University of Texas), Robert N. Lea (University of Houston), Baoxin Li (Arizona State University), Li Liao (University of Delaware), Gary Livingston (University of Massachusetts), Michael Lyle, Douglas W. Lynn (Oregon Institute of Technology), Yashwant K Malaiya (Colorado State University), Bill Mark (University of Texas at Austin), Ananda Mondal (Claf in University), Alvin Moser (Seattle University), Walid Najjar (University of California, Riverside), Danial J. Neebel (Loras College), John Nestor (Lafayette College), Joe Oldham (Centre College), Timour Paltashev, James Parkerson (University of Arkansas), Shaunak Pawagi (SUNY at Stony Brook), Steve Pearce, Ted Pedersen (University of Minnesota), Gregory D Peterson (The University of Tennessee), Dejan Raskovic (University of Alaska, Fairbanks) Brad Richards (University of Puget Sound), Roman Rozanov, Louis Rubinf eld (Villanova University), Md Abdus Salam (Southern University), Augustine Samba (Kent State University), Robert Schaefer (Daniel Webster College), Carolyn J. C. Schauble (Colorado State University), Keith Schubert (CSU San Bernardino), William L. Schultz, Kelly Shaw (University of Richmond), Shahram Shirani (McMaster University), Scott Sigman (Drury University), Bruce Smith, David Smith, Jeff W. Smith (University of Georgia, Athens), Philip Snyder (Johns Hopkins University), Alex Sprintson (Texas A&M), Timothy D. Stanley (Brigham Young University), Dean Stevens (Morningside College), Nozar Tabrizi (Kettering University), Yuval Tamir (UCLA), Alexander Taubin (Boston University), Will Thacker (Winthrop University), Mithuna Thottethodi (Purdue University), Manghui Tu (Southern Utah University), Rama Viswanathan (Beloit College), Guoping Wang (Indiana-Purdue University), Patricia Wenner (Bucknell University), Kent Wilken (University of California, Davis), David Wolfe (Gustavus Adolphus College), David Wood (University of Wisconsin, Madison), Mohamed Zahran (City College of New York), Gerald D. Zarnett (Ryerson University), Nian Zhang (South Dakota School of Mines & Technology), Jiling Zhong (Troy University), Huiyang Zhou (The University of Central Florida), Weiyu Zhu (Illinois Wesleyan University).

Me gustaría dar las gracias especialmente a la gente de Berkeley que dio una retroalimentación clave para el capítulo 7 y el apéndice A, que fueron las dos partes de este libro más complicadas de escribir: **Krste Asanovic, Christopher Batten, Rastilav Bodik, Bryan Catanzaro, Jike Chong, Kaushik Data, Greg Giebling, Anik Jain, Jae Lee, Vasily Volkov y Samuel Williams.**

Un agradecimiento especial también para **Mark Smotherman** por sus múltiples revisiones para encontrar errores técnicos y de escritura, que contribuyeron a mejorar significativamente la calidad de esta edición. Su papel ha sido incluso más importante esta vez puesto que esta edición se hizo en solitario.

Queremos agradecer a la familia de Morgan Kaufmann por estar de acuerdo en la publicación de este libro otra vez bajo la dirección de **Denise Penrose. Nathaniel McFadden** fue el editor de desarrollo (developmental editor) para esta edición y trabajó codo con codo conmigo en los contenidos del libro. **Kimberlee Honjo** coordinó las encuestas de los usuarios y sus respuestas.

**Dawnmarie Simpson** llevó a cabo la supervisión del proceso de producción del libro. También queremos dar las gracias a muchos agentes por cuenta propia (*fre-*

*elance vendors*) que contribuyeron a que este volumen vea la luz, especialmente a Alan Rose de Multiscience Press y diacriTech, nuestro corrector (compositor).

Las contribuciones de las casi 200 personas mencionadas aquí han ayudado a hacer de esta cuarta edición lo que espero sea nuestro mejor libro hasta el momento. ¡Gracias a todos!

*David A. Patterson*


# 1

*La civilización avanza  
extendiendo el  
número de operaciones  
importantes que se  
pueden hacer sin  
pensar en ellas*

**Alfred North Whitehead**  
*An Introduction to Mathematics*, 1911

## **Abstracciones y tecnología de los computadores**

- 1.1**    **Introducción**    3
- 1.2**    **Bajo los programas**    10
- 1.3**    **Bajo la cubierta**    13
- 1.4**    **Prestaciones**    26
- 1.5**    **El muro de la potencia**    39
- 1.6**    **El gran cambio: el paso de monoprocesadores a multiprocesadores**    41

<b>1.7</b>	<b>Casos reales: fabricación y evaluación del AMD Opteron x4</b>	44
<b>1.8</b>	<b>Falacias y errores habituales</b>	51
<b>1.9</b>	<b>Conclusiones finales</b>	54
 <b>1.10</b>	<b>Perspectiva histórica y lecturas recomendadas</b>	55
<b>1.11</b>	<b>Ejercicios</b>	56

**Nota importante:** En la presente edición en castellano, los contenidos del CD incluido en la edición original (en inglés) son accesibles a través de la página web [www.reverte.com/microsites/pattersonhennessy](http://www.reverte.com/microsites/pattersonhennessy). Aunque en la presente edición no se proporciona un CD-ROM físico, a lo largo de todo el texto se menciona el CD y se utiliza el icono que lo representa para hacer referencia a su contenido.

## 1.1

## Introducción

¡Bienvenido a este libro! Estamos encantados de tener esta oportunidad de transmitir el entusiasmo del mundo de los computadores. Éste no es un campo árido y aburrido, donde el progreso es glacial y donde las nuevas ideas se atrofian por negligencia. ¡No! Los computadores son el producto de la increíblemente vibrante industria de las tecnologías de la información, que en su conjunto es responsable como mínimo del 10% del producto nacional bruto de los Estados Unidos y cuya economía se ha vuelto, en parte, dependiente de las rápidas mejoras en tecnologías de la información prometidas por la ley de Moore. Esta insólita industria impulsa la innovación a una velocidad asombrosa. En los últimos 25 años aparecieron varios computadores nuevos cuya introducción parecía que iba a revolucionar la industria de la computación; estas revoluciones duraban poco tiempo simplemente porque alguien construía un computador aun mejor.

Esta carrera por innovar condujo a un progreso sin precedentes desde el inicio de la computación electrónica en los últimos años de la década de 1940. Si los medios de transporte hubiesen ido a la par con la industria de la computación, por ejemplo, hoy en día se podría viajar de Nueva York a Londres en aproximadamente un segundo por unos pocos céntimos. Pensemos un momento cómo tal adelanto habría cambiado la sociedad (vivir en Tahití y trabajar en San Francisco, ir a Moscú para ver el Ballet Bolshoi) y podremos apreciar las implicaciones de tal cambio.

Los computadores nos han llevado a una tercera revolución de la civilización, la revolución de la información, que se sitúa a la par de las revoluciones agrícola e industrial. El resultado de la multiplicación de la potencia y el alcance intelectual de la humanidad ha afectado profundamente a nuestras vidas cotidianas y también ha cambiado la manera de obtener nuevos conocimientos. Hay ahora una nueva forma de investigación científica, en la cual científicos informáticos trabajan junto a científicos teóricos y experimentales en la exploración de nuevas fronteras en astronomía, biología, química y física entre otras.

La revolución de los computadores continúa. Cada vez que el coste de la computación se mejora en un factor 10, las oportunidades para los computadores se multiplican. Aplicaciones que eran económicamente inviables repentinamente se convierten en factibles. Hasta hace muy poco, las siguientes aplicaciones eran “ciencia ficción computacional”.

- *Computadores en los coches*: hasta que los microprocesadores mejoraron drásticamente en precio y prestaciones a principios de la década de 1980, el control por computador en los coches era risible. Hoy en día los computadores reducen la contaminación y mejoran las prestaciones del combustible vía controles en el motor, y también mejoran la seguridad al prevenir peligrosos patinazos e inflando los *air bags* para proteger a los ocupantes en caso de colisión.
- *Teléfonos portátiles*: ¿quién habría podido soñar que los avances en sistemas de computación llevarían al desarrollo de teléfonos móviles, permitiendo una comunicación persona-a-persona casi en cualquier parte del mundo?
- *Proyecto genoma humano*: el coste del equipamiento informático para cartografiar y analizar las secuencias del ADN humano es de centenares de millones de dólares. Es improbable que alguien hubiese considerado este proyecto si los costes de los computadores hubiesen sido entre 10 y 100 veces mayores, tal y como eran hace 10 o 20 años. Además, el coste continúa bajando; podríamos ser capaces de adquirir nuestro propio genoma, permitiendo que los cuidados médicos se adapten a nosotros.
- *World Wide Web (La telaraña mundial)*: la World Wide Web, que no existía en la primera edición de este libro, ha cambiado nuestra sociedad. Para muchos, la *www* ha reemplazado a las bibliotecas.
- *Motores de búsqueda*: Dado que el contenido de la *WWW* ha crecido en tamaño y valor, encontrar información relevante es cada vez más importante. En la actualidad, mucha gente confía tanto en los motores de búsqueda que se verían en apuros si no pudiesen utilizarlos.

Claramente, los avances en esta tecnología influyen hoy en día en casi todos los aspectos de nuestra sociedad. Los avances en el hardware han permitido a los programadores crear programas maravillosamente útiles, y explican por qué los computadores son omnipresentes. Lo que hoy es ciencia ficción, serán las aplicaciones normales en el futuro: ya hay mundos virtuales, reconocimiento de voz y asistencia médica personalizada.

## Tipos de aplicaciones de computador y sus características

Aunque se usan un conjunto común de tecnologías hardware (presentadas en las secciones 1.3 y 1.4) que van desde los electrodomésticos caseros inteligentes a los teléfonos móviles o celulares o los mayores supercomputadores, estas aplicaciones diferentes tienen diversos requerimientos de diseño y utilizan las tecnologías hardware de manera diferente. Grosso modo, los computadores se utilizan en tres clases diferentes de aplicaciones.

Los **computadores de sobremesa** son posiblemente la forma más conocida de computación y están representados por el computador personal, que muchos lectores de este libro habrán usado extensamente. Los computadores de sobremesa se caracterizan por dar buenas prestaciones a bajo coste a único usuario, y a menudo se usan para ejecutar programas de terceros, también llamado software estándar. La evolución de muchas tecnologías es impulsada por este tipo de computadores, ¡que sólo tiene 30 años de antigüedad!

Los **servidores** son la versión moderna de lo que fueron los computadores centrales, los minicomputadores y los supercomputadores, y generalmente solo se accede a ellos vía una red. Los servidores están pensados para soportar grandes cargas de trabajo, que pueden consistir en una única aplicación compleja, generalmente científica o de ingeniería, o en muchos trabajos pequeños, como ocurre en un servidor Web. Estas aplicaciones están basadas en programas de otras fuentes (tales como una base de datos o un sistema de simulación), pero frecuentemente se modifican o adaptan a una función concreta. Los servidores se construyen con la misma tecnología básica que los computadores de sobremesa, pero permiten una mayor ampliación de su capacidad tanto de computación como de entrada/salida. En general, los servidores también ponen gran énfasis en la confiabilidad, puesto que un fallo es generalmente más costoso que en un computador de sobremesa de un único usuario.

Los servidores abarcan la mayor gama de costes y capacidades. En el extremo más bajo, un servidor puede ser un poco más que una máquina de sobremesa sin pantalla ni teclado y con un coste un poco mayor. Estos servidores de gama baja se usan típicamente para almacenar archivos o ejecutar pequeñas aplicaciones de empresas o un servicio web sencillo (véase la sección 6.10). En el otro extremo están los **supercomputadores**, que en la actualidad disponen de cientos de miles de procesadores, y generalmente **terabytes** de memoria y **petabytes** de almacenamiento, y cuestan de millones a cientos de millones de dólares. Los supercomputadores se utilizan para cálculos científicos y de ingeniería de alta calidad, tales como predicción del clima, prospección petrolífera, determinación de la estructura de proteínas, y otros problemas de gran envergadura. Aunque estos supercomputadores representan el pico de la capacidad de computación, en términos relativos constituyen una pequeña fracción del número total de servidores y también del total del mercado de los computadores en términos de facturación.

Aunque no se les llama supercomputadores, los **centros de datos** de internet utilizados por compañías como eBay y Google disponen también de miles de procesadores, terabytes de memoria y petabytes de almacenamiento. Habitualmente se consideran como grandes clústeres de computadores (véase capítulo 7).

**Computador de sobremesa:** computador diseñado para un único usuario, y que incorpora una pantalla, un teclado y un ratón.

**Servidor:** computador que se utiliza para ejecutar grandes programas para muchos usuarios, a menudo simultáneamente; generalmente sólo se accede a él vía un red.

**Supercomputador:** computador con la capacidad de computación y coste más altos; se configuran como servidores y generalmente su coste es de millones de dólares.

**Terabyte:** originalmente son 1 099 511 627 776 ( $2^{40}$ ) bytes, aunque algunos sistemas de comunicaciones y de almacenamiento secundario lo han redefinido como 1 000 000 000 000 ( $10^{12}$ ) bytes.

**Petabyte:** 1000 o 1024 terabytes.

**Centro de datos:** una habitación o edificio diseñado con todo lo que se necesita para un número elevado de servidores: alimentación y potencia eléctrica, aire acondicionado y red.