

P

Prefazione

*La cosa più bella che possiamo sperimentare
è il mistero; esso è la fonte della vera arte
e della vera scienza.*

Albert Einstein, *What I Believe*, 1930

Guida al libro

Crediamo che lo studio dell'informatica e dell'ingegneria informatica debba non solo riguardare i principi su cui è fondata l'elaborazione, ma anche riflettere lo stato delle conoscenze attuali in questi campi. Crediamo anche che i lettori, qualunque sia la branca dell'informatica nella quale lavorano, possano apprezzare i paradigmi secondo i quali sono organizzati i sistemi di elaborazione, che determinano le loro funzionalità e prestazioni, e ne decretano in ultima analisi il successo.

La tecnologia richiede oggi che i professionisti di tutte le branche dell'informatica conoscano sia il software sia l'hardware, la cui interazione a tutti i livelli è la chiave per capire i principi fondamentali dell'elaborazione. Inoltre, le idee che stanno alla base dell'organizzazione e della progettazione dei calcolatori valgono sia nell'ambito informatico sia in quello dell'ingegneria elettronica e sono le stesse sia che il vostro interesse principale sia il software sia l'hardware. Per questo motivo, nel libro l'enfasi viene posta sulla relazione tra hardware e software e vengono approfonditi i concetti che stanno alla base dei calcolatori delle ultime generazioni.

Il passaggio recente dalle architetture uniprocessore ai multiprocessori multicore ha confermato quanto sia corretta questa prospettiva, che abbiamo adottato fin dalla prima edizione di questo libro. Fino a poco tempo fa i programmatori potevano fare affidamento sul lavoro dei progettisti delle architetture e dei compilatori e dei produttori dei chip, per rendere più veloci o più efficienti dal punto di vista energetico i propri programmi senza il bisogno di apportare alcuna modifica. Questa era è finita: affinché un programma possa essere eseguito più velocemente, deve diventare un programma parallelo.

Anche se l'obiettivo di molti ricercatori è fare sì che i programmatori non si accorgano della natura parallela dell'hardware per il quale scrivono i loro programmi, ci vorranno molti anni prima che ciò divenga effettivamente possibile. Crediamo che nel prossimo decennio la maggior parte dei programmatori dovrà capire a fondo il legame tra hardware e software perché i programmi vengano eseguiti in modo efficiente sui calcolatori paralleli.

Questo libro è rivolto principalmente a coloro che, pur avendo scarse conoscenze del linguaggio assembler e della logica digitale, vogliono capire i concetti di base dell'organizzazione degli elaboratori, e a quei lettori che sono interessati a capire il modo in cui si progetta un elaboratore, come funziona e perché si ottengano determinate prestazioni.

L'altro testo degli stessi autori sulle architetture

Alcuni lettori conosceranno il testo degli stessi autori *Computer Architecture: A Quantitative Approach*, chiamato anche «Hennessy Patterson», mentre questo libro viene solitamente chiamato «Patterson Hennessy». Avevamo scritto quel libro con l'obiettivo di descrivere i principi su cui sono basate le architetture degli elaboratori utilizzando un robusto approccio ingegneristico per illustrare tutti i compromessi tra costi e prestazioni che si rendono necessari. In quel libro avevamo utilizzato un metodo basato su esempi e misure, effettuate su architetture disponibili sul mercato, per consentire al lettore di fare esperienza con la progettazione di sistemi realistici: l'obiettivo era dimostrare che le architetture degli elaboratori possono essere studiate utilizzando metodologie quantitative invece di un approccio descrittivo. Quel libro era stato pensato per i professionisti coscienti che volevano approfondire nei dettagli il funzionamento dei calcolatori.

La maggior parte dei lettori di questo libro non ha intenzione di diventare un progettista di calcolatori. Tuttavia, le prestazioni e l'efficienza energetica dei sistemi software prodotti nel prossimo futuro saranno enormemente influenzati da quanto bene i progettisti software avranno capito le strutture hardware di base che sono presenti in un calcolatore. Perciò, i progettisti dei compilatori e dei sistemi operativi, così come i programmatori di database e della maggior parte delle applicazioni hanno bisogno di conoscere bene i principi fondamentali in base ai quali funziona un calcolatore, presentati in questo libro. Analogamente, i progettisti hardware devono capire a fondo come i loro progetti andranno a influenzare le applicazioni software.

Ci siamo quindi resi conto che questo libro doveva essere molto di più di una semplice estensione del materiale contenuto nell'altro testo, per cui abbiamo riesaminato a fondo il contenuto e lo abbiamo modificato adattandolo ai lettori di questo libro. Il risultato ha avuto così tanto successo che abbiamo eliminato tutto il materiale introduttivo dalle versioni successive di *Computer Architecture* e, quindi, ora la sovrapposizione dei contenuti tra i due libri è minima.

Le novità di questa edizione

Nella stesura di questa edizione di *Struttura e progetto dei calcolatori* abbiamo perseguito **sei obiettivi principali**: dimostrare con esempi reali quanto sia importante comprendere il funzionamento dell'hardware; evidenziare i temi principali di ogni argomento inserendo a bordo pagina le icone associate, che vengono introdotte nelle pagine iniziali; proporre nuovi esempi per rispecchiare i cambiamenti occorsi nel passaggio dall'era dei PC all'era post-PC; distribuire il materiale relativo all'I/O su tutto il libro invece di racchiuderlo in un unico capitolo; aggiornare il contenuto tecnico per rispecchiare i cambiamenti nell'industria negli anni successivi alla pubblicazione della precedente edizione e spo-

stare online le appendici e gli altri capitoli invece di includerli in un CD, per contenere il prezzo e rendere questa edizione disponibile anche in forma elettronica.

Prima di descrivere più in dettaglio questi obiettivi, analizziamo capitolo per capitolo l'approccio adottato nella descrizione sia dell'hardware sia del software (si veda la tabella alla fine della prefazione). I Capitoli 1, 4, 5 e 6 coprono entrambe le aree, indipendentemente dall'argomento trattato. Il Capitolo 1 spiega perché sia diventato importante il consumo di energia e perché questo abbia causato il passaggio dai microprocessori a processore singolo ai microprocessori multicore. In questo capitolo vengono anche introdotte le otto grandi idee sulla progettazione delle architetture degli elaboratori. Il Capitolo 2 è principalmente un capitolo introduttivo all'hardware, ma contiene anche una descrizione dei compilatori e dei linguaggi di programmazione a oggetti, materiale fondamentale per i lettori più interessati agli aspetti software. Il Capitolo 3 è dedicato all'aritmetica in virgola mobile e all'unità di elaborazione dati. I lettori possono tralasciare le parti di questo capitolo che non interessano o che contengono materiale introduttivo, ma non i Paragrafi da 3.6 a 3.8. Questi descrivono una procedura che calcola il prodotto di due matrici, mostrando come il parallelismo a livello di parola consenta di migliorare le prestazioni di quattro volte. Il Capitolo 4 descrive i processori dotati di pipeline. I Paragrafi 4.1, 4.5 e 4.10 contengono un riassunto degli argomenti trattati nel capitolo e nel Paragrafo 4.12 vengono mostrate le modifiche alla procedura che esegue il prodotto di due matrici, che consentono un ulteriore aumento delle prestazioni. Anche i lettori interessati all'hardware troveranno del materiale fondamentale in questo capitolo; se le conoscenze sui circuiti logici non sono sufficienti, si può leggere l'Appendice C sulla progettazione dei circuiti logici, disponibile online, prima di avventurarsi nella lettura di questo capitolo. L'ultimo capitolo, su multicore, multiprocessori e cluster, contiene principalmente nuovo materiale e dovrebbe essere letto da tutti. Il capitolo è stato riorganizzato per rendere più naturale la successione degli argomenti e contiene una descrizione più approfondita di GPU, grandi centri di calcolo e delle interfacce hardware/software delle schede di rete, fondamentali per i cluster.

Il **primo obiettivo** di questa edizione è quello di *utilizzare un esempio concreto per dimostrare quanto sia importante comprendere il funzionamento dell'hardware per ottenere buone prestazioni ed elevata efficienza energetica*. Come già accennato, inizieremo descrivendo il parallelismo a livello di parola nel Capitolo 3, che consente di ottenere un miglioramento delle prestazioni di un fattore 4 sulla moltiplicazione di due matrici. Miglioreremo ulteriormente le prestazioni nel Capitolo 4, attraverso l'espansione dei cicli, dimostrando così l'importanza del parallelismo a livello di istruzioni. Nel Capitolo 5 raddoppieremo ancora le prestazioni, ottimizzando l'utilizzo della cache mediante l'accesso a blocchi. Infine, nel Capitolo 6 mostreremo come ottenere un miglioramento di 14 volte utilizzando 16 processori e il parallelismo a livello di thread. Per ottenere tutte queste ottimizzazioni aggiungeremo solamente 24 linee di codice C alla procedura iniziale.

Il **secondo obiettivo** è aiutare i lettori a *distinguere le idee fondamentali identificando all'inizio otto grandi idee nella progettazione delle architetture* che vengono richiamate nel resto del libro. Abbiamo inserito le relative icone a bordo pagina ed evidenziato nel testo le parole corrispondenti per ricordare ai lettori questi otto argomenti. Questo libro contiene quasi 100 citazioni. Tutti i capitoli contengono almeno sette esempi di grandi idee, e ciascuna idea viene citata almeno cinque volte. Le prestazioni attraverso il parallelismo, la pipeline e la predizione sono tre delle idee più utilizzate, seguite da vicino dalla Legge di Moore. Il Capitolo 4, riservato al processore, è quello che contiene più esempi; ciò non deve sorprendere, poiché è il capitolo che ha probabilmente riscosso più successo presso i progettisti delle architetture digitali. La grande idea ri-

chiamata in tutti i capitoli è quella delle prestazioni attraverso il parallelismo, ben allineata all'enfasi recente sul parallelismo.

Il **terzo obiettivo** è *evidenziare il passaggio generazionale nel campo dell'elaborazione dalla generazione dei PC a quella post-PC*, illustrato da esempi e commenti: il Capitolo 1 analizza in dettaglio un calcolatore tablet invece di un PC e il Capitolo 6 descrive l'infrastruttura di calcolo di un cloud. In questo libro trattiamo anche l'ARM, che è l'insieme di istruzioni utilizzato nei dispositivi mobili personali dell'era post-PC, esattamente come l'insieme di istruzioni x86 ha dominato l'era dei PC e (fino a oggi) domina il mondo del cloud computing.

Il **quarto obiettivo** consiste nel *distribuire il materiale relativo all'I/O su tutto il libro* invece di concentrarlo in un unico capitolo, come abbiamo fatto per il parallelismo nell'edizione precedente: potete trovare il materiale sull'I/O nei Paragrafi 1.4, 4.9, 5.2, 5.5, 5.11 e 6.9. Pensiamo che i lettori (e i professori) leggeranno più volentieri il materiale sull'I/O se non è contenuto in un capitolo a sé.

Il mondo degli elaboratori è un mondo in evoluzione molto veloce e quindi, come succede sempre per le nuove edizioni, uno degli obiettivi più importanti è *utilizzare contenuto tecnico aggiornato*. Relativamente al **quinto obiettivo**, come esempio di architetture in questo libro abbiamo utilizzato il Cortex-A8 ARM e il Core i7 Intel, esempi tipici dei calcolatori dell'era post-PC. Altri contenuti importanti sono una panoramica del nuovo insieme di istruzioni a 64 bit ARMv8, un'introduzione sulla GPU che spiega anche la sua terminologia, una trattazione più approfondita dei calcolatori dei grandi centri di calcolo che sono alla base del cloud e una descrizione dettagliata delle schede Ethernet a 10 Gigabyte.

Rispetto al **sesto obiettivo**, per *contenere il prezzo e mantenere il libro di dimensioni ragionevoli e compatibile con una versione elettronica*, abbiamo reso disponibili online il materiale complementare e le appendici invece di inserirli in un CD allegato, come avveniva nelle edizioni precedenti.

Abbiamo conservato le caratteristiche delle edizioni precedenti rivelatesi più utili: abbiamo mantenuto la definizione delle parole chiave a margine del testo la prima volta che compaiono, le sezioni *Capire le prestazioni dei programmi* (dedicate alle prestazioni e a come migliorarle), le sezioni *Interfaccia hardware/software* (sui compromessi da adottare a livello di questa interfaccia), le sezioni *Quadro d'insieme* (che riepilogano i concetti principali espressi nel testo) e le sezioni *Autovalutazione*, che aiutano il lettore a valutare la comprensione degli argomenti trattati (con le relative risposte esatte alla fine di ogni capitolo). Infine, abbiamo aggiornato tutti gli esercizi. Anche questa edizione contiene nell'ultima pagina una scheda tecnica riassuntiva del MIPS. Il contenuto della scheda è stato aggiornato e costituisce un riferimento immediato per coloro che scrivono programmi nell'assembler del MIPS.

Risorse multimediali per gli studenti e per i docenti

All'indirizzo online.universita.zanichelli.it/patterson-4e sono disponibili le seguenti risorse multimediali:

per lo studente

- le soluzioni degli esercizi di fine capitolo,
- le appendici e i paragrafi aggiuntivi,
- gli approfondimenti,
- alcuni software e modelli,
- i tutorial.

Alcune di queste risorse sono richiamate nel libro con l'icona .

per il docente

- le illustrazioni contenute nel libro di testo suddivise per capitolo.

Chi acquista il libro può inoltre scaricare gratuitamente l'**ebook multimediale**, seguendo le istruzioni presenti nel sito sopra indicato. L'ebook si legge con l'applicazione *Booktab*, che si scarica gratis da App Store (sistemi operativi Apple) o da Google Play (sistemi operativi Android). Per accedere alle risorse protette è necessario registrarsi su **myzanicelli.it** inserendo la chiave di attivazione personale contenuta nel libro.

Considerazioni conclusive

Se leggerete il successivo paragrafo di ringraziamenti, vi renderete conto che abbiamo corretto moltissimi errori. E quando un libro passa attraverso diverse ristampe, si ha la possibilità di correggere un numero ancora maggiore di errori. Se doveste trovare altri errori, per cortesia, contattate direttamente l'editore attraverso la posta elettronica o la posta ordinaria, utilizzando gli indirizzi riportati nella pagina del copyright.

Questa edizione è la seconda dopo l'interruzione della lunga collaborazione tra Hennessy e Patterson, iniziata nel 1989. Gli impegni richiesti per dirigere una delle più importanti università del mondo hanno tolto a Hennessy il tempo necessario per lavorare alle nuove edizioni: il suo coautore, Patterson, si è sentito ancora una volta come un acrobata che si esibisce senza rete. Per questo motivo, le persone elencate nel paragrafo dei ringraziamenti e i colleghi di Berkeley hanno avuto un ruolo ancora maggiore nel dare forma al contenuto di questo libro. Ciò nonostante, uno solo è l'autore responsabile del nuovo materiale che vi apprestate a leggere.

Ringraziamenti per questa edizione

Siamo stati davvero fortunati a ricevere diversi contributi da molti lettori, redattori e ricercatori. Ciascuno di loro ha contribuito a rendere migliore il libro. Il Capitolo 6 è stato rivisto a fondo e abbiamo analizzato separatamente le idee e i contenuti; infine ho apportato le modifiche in base ai suggerimenti di ogni revisore. Vorrei ringraziare Christos Kozyrakis dell'Università di Stanford per il suggerimento di utilizzare l'interfaccia di rete dei cluster per illustrare l'interfaccia hardware/software dell'I/O e per i suggerimenti su come organizzare il capitolo; Mario Flajsilk dell'Università di Stanford, che ha fornito dettagli, diagrammi e misure delle prestazioni del NIC NetFPGA; e i seguenti ricercatori per i loro suggerimenti su come migliorare il capitolo: David Kaeli della Northwestern University, Partha Ranganathan degli HP Labs, David Wood dell'Università del Wisconsin e i miei colleghi di Berkeley Siamak Faridani, Shoaib Kamil, Yunsup Lee, Zhangxi Tan e Andrew Waterman.

Un particolare ringraziamento va a Rimas Avizensis dell'Università di Berkeley, che ha sviluppato le diverse versioni della procedura di moltiplicazione di matrici e fornito la misura delle prestazioni. Avendo lavorato con suo padre quando ero studente di dottorato a UCLA, è stato particolarmente piacevole lavorare con Rimas a Berkeley.

Vorrei ringraziare anche il mio collaboratore di lunga data Randy Katz dell'Università di Berkeley, che mi ha aiutato a sviluppare il concetto delle grandi idee nelle architetture degli elaboratori all'interno della preparazione del corso della laurea di primo livello che abbiamo tenuto assieme.

Desidero ringraziare David Kirk, John Nickolls e i loro colleghi di NVIDIA (Micheal Garland, John Montrym, Doug Voorhies, Lars Nyland, Erik Londholm, Laulus Micikevicius, Massimiliano Fatica, Stuart Oberman e Vasily Volkov) per avere scritto l'Appendice C, che tratta approfonditamente le GPU. Vorrei rinnovare il mio apprezzamento a Jim Larus, recentemente nominato direttore della Facoltà di «Computer and Communication Science» di EPFL, per il suo desiderio di contribuire con la sua esperienza sulla program-

mazione in linguaggio assembler e per avere messo a disposizione dei lettori il simulatore MIPS da lui sviluppato e costantemente aggiornato. Sono anche molto grato a Jason Bakos dell'Università del South Carolina, che ha aggiornato gli esercizi dell'edizione precedente e ne ha creati di nuovi per questa edizione, a partire dagli esercizi sviluppati per la quarta edizione da: Perry Alexander (Università del Kansas), Javier Bruguera (Università di Santiago di Compostela), Matthew Farrens (Università di California, Davis), David Kaeli (Northeastern University), Nicole Kaiyan (Università di Adelaide); John Oliver (Cal Poly, San Luis Obispo), Milos Prvulovic (Georgia Tech) e Jichuan Chang, Jacob Leverich, Kevin Lim, e Partha Ranganathan (tutti di Hewlett-Packard). Un ulteriore ringraziamento va a Jason Bakos per avere sviluppato nuove slide per le lezioni.

Sono particolarmente grato ai diversi docenti che hanno risposto ai questionari fatti circolare dall'editore Morgan Kaufmann, hanno commentato la nostra proposta editoriale e hanno partecipato ai gruppi di discussione per analizzare e fornire suggerimenti per questa edizione. Questi docenti sono (Gruppo di Discussione del 2012): Bruce Barton (Suffolk County Community College), Jeff Braun (Montana Tech), Ed Gehringer (North Carolina State), Michael Goldweber (Xavier University), Ed Harcourt (St. Lawrence University), Mark Hill (University of Wisconsin, Madison), Patrick Homer (University of Arizona), Norm Jouppi (HP Labs), Dave Kaeli (Northeastern University), Christos Kozyrakis (Stanford University), Zachary Kurmas (Grand Valley State University), Jae C. Oh (Syracuse University), Lu Peng (LSU), Milos Prvulovic (Georgia Tech), Partha Ranganathan (HP Labs), David Wood (University of Wisconsin), Craig Zilles (University of Illinois at Urbana-Champaign). Questionari e revisioni: 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 (St. Lawrence University), Justin E. Harlow (University of South Florida), Paul F. Hemler (Hampden-Sydney College), Martin Herboldt (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), E.J. Kim (Texas A&M University), Jihong Kim (Seoul National University), 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), Zachary Kurmas (Grand Valley State University), 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 (Claflin University), Alvin Moser (Seattle University), Walid Najjar (University of California, Riverside), Danial J. Neebel (Loras College), John Nestor (Lafayette College), Jae C. Oh (Syracuse University), Joe Oldham (Centre College), Timour Paltashev, James Parkerson (University of Arkansas), Shaunak Pawagi (SUNY at Stony Brook), Steve Pearce, Ted Pedersen (University of Minnesota), Lu Peng (Louisiana State University), Gregory D Peterson (The University of Tennessee), Milos Prvulovic (Georgia Tech), Partha Ranganathan (HP Labs), Dejan Raskovic (University of Alaska, Fairbanks) Brad Richards (University of Puget Sound), Roman Rozanov, Louis Rubinfeld (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), Mark Smotherman (Clemson University), 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), Dean Tullsen (UC San Diego), Rama Viswanathan (Beloit College), Ken Vollmar (Missouri State University), 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), Ki Hwan Yum (University of Texas, San Antonio), 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).

Desidero ringraziare in particolare modo Mark Smotherman per le successive revisioni della parte tecnica e per avere fornito spunti che hanno migliorato significativamente la qualità di questa edizione.

Desideriamo ringraziare la grande famiglia di Morgan Kaufmann per avere accettato di pubblicare questa nuova edizione del libro sotto la guida capace di Todd Green e Nate McFadden: non sarei riuscito a terminare questo libro senza il loro aiuto. Vogliamo ringraziare anche Lisa Jones che ha gestito il processo di produzione del libro e Russell Purdy che ha disegnato la copertina dell'edizione americana, che congiunge in modo intelligente il contenuto dell'era post-PC con quello della prima edizione.

Il contributo delle quasi 150 persone elencate qui ci ha consentito di realizzare questa edizione, che spero risulti l'edizione migliore. Buona lettura!


David A. Patterson

Capitolo o Appendice	Paragrafi	Focalizzazione sul software	Focalizzazione sull'hardware
1. Il calcolatore: astrazioni e tecnologia	Da 1.1 a 1.11		
	 1.12 (Storia)		
2. Le istruzioni: il linguaggio dei calcolatori	Da 2.1 a 2.14		
	 2.15 (Compilatori e Java)		
	Da 2.16 a 2.20		
	 2.21 (Storia)		
E. A Survey of RISC Architectures for Desktop, Server and Embedded Computers	 Da E.1 a E.17		
3. L'aritmetica dei calcolatori	Da 3.1 a 3.5		
	Da 3.6 a 3.8 (Parallelismo a livello di parola)		
	Da 3.9 a 3.10 (Errori e trabocchetti)		
	 3.11 (Storia)		
B. The Basics of Logic Design	 Da B.1 a B.13		
4. Il processore	4.1 (Panoramica)		
	4.2 (Convenzioni logiche)		
	Da 4.3 a 4.4 (Una semplice implementazione)		
	4.5 (Introduzione alla pipeline)		
	4.6 (Unità di elaborazione dotata di pipeline)		
	Da 4.7 a 4.9 (Hazard ed eccezioni)		
	Da 4.10 a 4.12 (Parallelismo, esempi reali)		
	 4.13 (Unità di controllo della pipeline in Verilog)		
	Da 4.14 a 4.15 (Errori e trabocchetti)		
	 4.16 (Storia)		
D. Mapping Control to Hardware	 Da D.1 a D.6		
5. Grande e veloce: la gerarchia delle memorie	Da 5.1 a 5.10		
	 5.11 Dischi RAID		
	 5.12 (Controllori della cache in Verilog)		
	Da 5.13 a 5.16		
	 5.17 (Storia)		
6. Processori paralleli: dai client al cloud	Da 6.1 a 6.8		
	 6.9 (Le reti)		
	Da 6.10 a 6.14		
	 6.15 (Storia)		
A. Gli assembleri, i linker e il simulatore SPIM	 Da A.1 a A.11		
C. La grafica e il calcolo con la GPU	 Da C.1 a C.13		

Leggere con attenzione 

Riguardare o leggere 

Leggere se si ha tempo 

Leggere per cultura personale 

Riferimento 