

Prefazione degli autori

Questo libro è unico nel senso che presenta il progetto della logica digitale dal punto di vista dell'architettura dei calcolatori, partendo dagli 1 e dagli 0, procedendo fino al progetto di un microprocessore.

Riteniamo che costruire un microprocessore sia uno speciale rito di passaggio per gli studenti di ingegneria e di informatica. Il funzionamento interno di un processore può sembrare quasi magico agli occhi degli inesperti, ma diventa chiarissimo se spiegato con cura. Il progetto digitale di per sé è un argomento vasto e interessante. Programmare in linguaggio assembly rivela il linguaggio interno del processore. La microarchitettura è il ponte di collegamento che unisce il tutto.

Le prime due edizioni di questo testo, la cui popolarità va aumentando, trattavano l'architettura MIPS adeguandosi all'impostazione data dai diffusi testi di architettura dei calcolatori di Patterson e Hennessy. Il MIPS – una delle prime architetture RISC (*Reduced Instruction Set Computer*) – è chiaro e facile da comprendere e costruire. MIPS rimane un'architettura importante, e ha ricevuto nuova energia dopo il suo acquisto nel 2013 da parte della Imagination Technologies.

Negli ultimi due decenni, l'architettura ARM è esplosa in popolarità grazie alla sua efficienza e ricchezza. Sono stati prodotti più di 50 miliardi di processori ARM, e più del 75% della popolazione mondiale utilizza prodotti che contengono processori ARM. Al momento della stesura di questo libro, praticamente ogni telefono cellulare e ogni tablet contengono uno o più processori ARM e le previsioni dicono che migliaia di milioni di processori ARM controlleranno in futuro la cosiddetta *Internet of Things* (Internet delle cose). Molte aziende stanno costruendo sistemi ARM ad alte prestazioni per poter concorrere nel mercato dei server contro i processori Intel. Vista l'importanza commerciale e l'interesse da parte degli studenti, abbiamo deciso di sviluppare questa edizione ARM del nostro libro.

Da un punto di vista pedagogico, gli obiettivi di apprendimento dell'edizione MIPS e dell'edizione ARM sono identici. L'architettura ARM possiede un numero di funzioni, inclusi i metodi di indirizzamento e l'esecuzione condizionale, che contribuiscono all'efficacia del processore ma aumentano un po' il livello di complessità. Anche le microarchitetture sono molto simili: le differenze maggiori sono l'esecuzione condizionale e il program counter. Il capitolo sull'ingresso/uscita fornisce numerosi esempi utilizzando il Raspberry Pi, un diffusissimo calcolatore Linux su singola scheda basato su ARM.

Ci aspettiamo di offrire entrambe le edizioni MIPS e ARM finché il mercato lo richiede.

CARATTERISTICHE

Trattazione fianco a fianco di SystemVerilog e VHDL

I linguaggi di descrizione dell'hardware (HDL, *Hardware Description Languages*) sono il punto chiave delle moderne pratiche di progettazione digitale. Ma i progettisti si dividono quasi equamente tra due linguaggi dominanti: SystemVerilog e VHDL. In questo libro, gli HDL sono trattati nel Capitolo 4 dopo la presentazione della logica combinatoria e di quella sequenziale. Gli HDL vengono utilizzati successivamente nei Capitoli 5 e 7 per progettare

blocchi di costruttivi più complessi e interi processori. È comunque possibile tralasciare il Capitolo 4: i capitoli successivi rimangono utilizzabili anche per quei corsi che non affrontano lo studio degli HDL.

Questo libro è unico nella sua presentazione fianco a fianco di SystemVerilog e VHDL, che permette al lettore di apprendere entrambi i linguaggi. Il Capitolo 4 descrive i principi che si applicano a entrambi gli HDL, e successivamente presenta in colonne adiacenti la sintassi e gli esempi d'uso dei due linguaggi specifici. Questa doppia trattazione permette inoltre a un docente di scegliere uno dei due HDL, e al lettore di passare facilmente da uno all'altro, sia in aula sia nella pratica professionale.

Architettura e microarchitettura ARM

I Capitoli 6 e 7 offrono la prima esposizione dettagliata dell'architettura e della microarchitettura di ARM. ARM è un'architettura perfetta per un libro, in quanto è un'architettura reale presente in milioni di prodotti ogni anno, pur restando un'architettura agile e facile da imparare. Inoltre, data la sua popolarità nel mondo commerciale e degli hobbisti, esistono per questa architettura numerosi simulatori e strumenti di sviluppo. Tutti i materiali relativi alla tecnologia ARM sono stati riprodotti con permesso rilasciato dalla ARM Limited.

Prospettive del mondo reale

Oltre alla prospettiva del mondo reale offerta presentando l'architettura ARM, il Capitolo 6 illustra l'architettura dei processori Intel x86 per dare una prospettiva alternativa. Il Capitolo 9 (disponibile come risorsa aggiuntiva online) descrive anche le periferiche relative al calcolatore su scheda singola Raspberry Pi, una piattaforma basata su ARM molto diffusa. Queste prospettive sul mondo reale mostrano come i concetti esposti nei capitoli si traducano nei chip presenti in molti personal computer e in altri prodotti elettronici di largo consumo.

Panoramica accessibile di una microarchitettura avanzata

Il Capitolo 7 include una visione d'insieme sulle caratteristiche architetturali di una moderna microarchitettura ad alte prestazioni, come la predizione dei salti, l'esecuzione superscalare e *out-of-order*, il *multithreading*, i processori *multicore*. La trattazione è accessibile anche agli studenti dei primi anni e mostra come le microarchitetture esposte nel libro possano essere estese ai processori moderni.

Esercizi ricapitolativi e domande per colloqui

Il modo migliore per imparare la progettazione digitale è metterla in pratica. Ogni capitolo termina quindi con degli esercizi ricapitolativi per fare pratica sull'argomento. Gli esercizi sono seguiti da alcune domande di valutazione che i nostri colleghi dell'industria hanno posto a studenti candidatisi per posti di lavoro in questo campo. Queste domande forniscono un'anteprima utile dei tipi di problemi davanti ai quali vengono messi i candidati a un posto di lavoro durante il colloquio di assunzione. Le soluzioni agli esercizi sono disponibili sul sito del libro (online.universita.zanichelli.it/harris-arm).

LE RISORSE MULTIMEDIALI

All'indirizzo online.universita.zanichelli.it/harris-arm sono disponibili le seguenti risorse:

- le soluzioni degli esercizi con numero dispari (in lingua inglese);
- il Capitolo 9, sui sistemi I/O;
- l'appendice C, sulla programmazione in C;
- i materiali per il laboratorio (in lingua inglese);
- il codice HDL del processore ARM (in lingua inglese).

COME USARE GLI STRUMENTI SOFTWARE IN UN CORSO

Altera Quartus II

Quartus II Web Edition è una versione gratuita dello strumento professionale di progettazione di componenti FPGA Quartus II. Questo strumento permette agli studenti di costruire i propri progetti digitali come schemi circuitali o di utilizzare i linguaggi di descrizione hardware (HDL) SystemVerilog e VHDL. Una volta costruito il progetto, gli studenti possono lanciare una simulazione della struttura circuitale utilizzando ModelSim™-Altera Starter Edition, disponibile all'interno di Altera Quartus II Web Edition. Quartus II Web Edition include anche uno strumento di sintesi logica che supporta sia SystemVerilog sia VHDL.

La differenza tra Web Edition e Subscription Edition è che la prima supporta solo un sottoinsieme dei componenti FPGA di Altera più diffusi. La differenza tra ModelSim-Altera Starter Edition e le versioni commerciali di ModelSim è che la Starter Edition offre prestazioni degradate per simulazioni di codici con più di 10 000 righe di HDL.

ARM Microcontroller Development Kit (MDK-ARM) di Keil

MDK-ARM della Keil è uno strumento per sviluppare programmi per il processore ARM, disponibile gratuitamente su web. Include un compilatore ARM C commerciale e un simulatore che permette agli studenti di scrivere programmi sia in linguaggio C sia in linguaggio assembly, di compilarli e di simularli.

RINGRAZIAMENTI

Ringraziamo il duro lavoro di Nate McFadden, Joe Hayton, Punithavathy Govindaradjane e di tutti gli altri componenti del gruppo di lavoro della Morgan Kaufmann, che ha reso possibile la realizzazione di questo volume. Apprezziamo moltissimo il lavoro di Duane Bibby, le cui vignette animano i capitoli.

Ringraziamo Matthew Watkins che ha contribuito al paragrafo sui multiprocessori eterogenei nel Capitolo 7. Abbiamo molto apprezzato il lavoro di Joshua Vasquez che ha sviluppato il codice per il Raspberry Pi nel Capitolo 9. Ringraziamo anche Josef Spjut e Ruye Wang, che hanno collaudato il materiale in classe.

Numerosi revisori hanno migliorato sostanzialmente questo libro. Tra questi ricordiamo Boyang Wang, John Barr, Jack V. Briner, Andrew C. Brown, Carl Baumgaertner, A. Utku Diril, Jim Frenzel, Jaeha Kim, Phillip King, James PinterLucke, Amir Roth, Z. Jerry Shi, James E. Stine, Luke Teysier, Peiyi Zhao, Zach Dodds, Nathaniel Guy, Aswin Krishna, Volnei Pedroni, Karl Wang, Ricardo Jasinski, Josef Spjut, Jörgen Lien, Sameer Sharma, John Nestor, Syed Manzoor, James Hoe, Srinivasa Vemuru, K. Joseph Hass, Jayantha Herath, Robert Mullins, Bruno Quoitin, Subramaniam Ganesan, Braden Phillips, John Oliver, Yahswant K. Malaiya, Mohammad Awedh, Zachary Kurmas, Donald Hung, e un revisore anonimo. Ringraziamo inoltre Khaled Benkrid e i suoi colleghi della ARM per aver esaminato attentamente i materiali relativi ad ARM.

Ringraziamo anche gli studenti dei nostri corsi allo Harvey Mudd College e a UNLV che ci hanno dato le loro utili opinioni sulle bozze di questo libro di testo. Sono degni di menzione Clinton Barnes, Matt Weiner, Carl Walsh, Andrew Carter, Casey Schilling, Alice Clifton, Chris Acon, e Stephen Brawner.

Un ultimo ma non per questo meno importante ringraziamento va alle nostre famiglie per l'amore e il supporto che ci hanno dato.