

*Alle nostre famiglie*



# Indice generale

<i>Prefazione degli autori</i> . . . . .	XIII
<i>Ringraziamenti</i> . . . . .	XV

<b>1.9</b> ■ <b>Riassunto e anticipazione</b> . . . . .	27
Esercizi . . . . .	29
Domande di valutazione. . . . .	34

## Capitolo 1 Da zero a uno

<b>1.1</b> ■ <b>La pianificazione del gioco</b> . . . . .	1
<b>1.2</b> ■ <b>L'arte di gestire la complessità</b> . . . . .	2
1.2.1 L'astrazione. . . . .	2
1.2.2 La disciplina. . . . .	3
1.2.3 Le tre -Y. . . . .	3
<b>1.3</b> ■ <b>L'astrazione digitale</b> . . . . .	5
<b>1.4</b> ■ <b>I sistemi numerici</b> . . . . .	6
1.4.1 Numeri decimali . . . . .	6
1.4.2 Numeri binari . . . . .	7
1.4.3 Numeri esadecimali . . . . .	8
1.4.4 Byte, nibble, word . . . . .	9
1.4.5 Somma binaria . . . . .	10
1.4.6 Numeri binari relativi . . . . .	11
<b>1.5</b> ■ <b>Le porte logiche</b> . . . . .	14
1.5.1 La porta NOT. . . . .	15
1.5.2 Buffer. . . . .	15
1.5.3 La porta AND . . . . .	15
1.5.4 La porta OR. . . . .	15
1.5.5 Altre porte logiche a due ingressi . . . . .	15
1.5.6 Porte a ingressi multipli . . . . .	16
<b>1.6</b> ■ <b>Oltre l'astrazione digitale</b> . . . . .	16
1.6.1 Tensione . . . . .	17
1.6.2 Livelli logici . . . . .	17
1.6.3 Margini di rumore . . . . .	17
1.6.4 Caratteristica di trasferimento DC. . . . .	18
1.6.5 La disciplina statica. . . . .	19
<b>1.7</b> ■ <b>I transistori CMOS*</b> . . . . .	20
1.7.1 Semiconduttori . . . . .	20
1.7.2 I diodi. . . . .	21
1.7.3 I condensatori . . . . .	21
1.7.4 Transistori nMOS e pMOS . . . . .	22
1.7.5 Porta NOT CMOS. . . . .	24
1.7.6 Altre porte logiche CMOS . . . . .	24
1.7.7 Porte di trasmissione . . . . .	25
1.7.8 Logica pseudo-nMOS . . . . .	26
<b>1.8</b> ■ <b>Consumo di potenza*</b> . . . . .	26

## Capitolo 2 Progetto di reti logiche combinatorie

<b>2.1</b> ■ <b>Introduzione</b> . . . . .	35
<b>2.2</b> ■ <b>Espressioni booleane</b> . . . . .	37
2.2.1 Terminologia. . . . .	38
2.2.2 Forma somma di prodotti . . . . .	38
2.2.3 Forma prodotto di somme . . . . .	39
<b>2.3</b> ■ <b>Algebra booleana</b> . . . . .	40
2.3.1 Postulati. . . . .	40
2.3.2 Teoremi a una variabile . . . . .	41
2.3.3 Teoremi di più variabili . . . . .	41
2.3.4 Saranno veri i teoremi booleani? . . . . .	43
2.3.5 Semplificare le espressioni . . . . .	43
<b>2.4</b> ■ <b>Dalla logica alle porte</b> . . . . .	45
<b>2.5</b> ■ <b>Logica combinatoria su più di due livelli</b> . . . . .	47
2.5.1 Riduzione dell'hardware . . . . .	48
2.5.2 Spingere le bolle. . . . .	49
<b>2.6</b> ■ <b>Non solo 0 e 1, anche X e Z</b> . . . . .	50
2.6.1 Il valore illegale: X . . . . .	50
2.6.2 Il valore fluttuante: Z . . . . .	51
<b>2.7</b> ■ <b>Le mappe di Karnaugh</b> . . . . .	52
2.7.1 Pensare in cerchi. . . . .	53
2.7.2 Minimizzazione logica con le mappe di Karnaugh . . . . .	54
2.7.3 Indifferenze. . . . .	57
2.7.4 Il quadro generale . . . . .	57
<b>2.8</b> ■ <b>Blocchi costitutivi combinatori</b> . . . . .	58
2.8.1 Multiplexer . . . . .	58
2.8.2 Decoder. . . . .	61
<b>2.9</b> ■ <b>Temporizzazioni</b> . . . . .	62
2.9.1 Ritardi di propagazione e di contaminazione . . . . .	62
2.9.2 Alee . . . . .	65
<b>2.10</b> ■ <b>Riassunto</b> . . . . .	67
Esercizi . . . . .	69
Domande di valutazione. . . . .	72

## Capitolo 3 Progetto di logica sequenziale

<b>3.1</b>	<b>■ Introduzione</b> .....	73
<b>3.2</b>	<b>■ Latch e flip-flop</b> .....	73
3.2.1	Latch SR .....	75
3.2.2	Latch D .....	76
3.2.3	Flip-flop D .....	77
3.2.4	Registro .....	78
3.2.5	Flip-flop con abilitazione .....	78
3.2.6	Flip-flop resettabile .....	78
3.2.7	Progetto di latch e flip-flop a livello di transistori .....	79
3.2.8	Per riassumere .....	80
<b>3.3</b>	<b>■ Progetto di reti logiche sincrone</b> ..	81
3.3.1	Alcune reti problematiche .....	81
3.3.2	Reti sequenziali sincrone .....	82
3.3.3	Reti sincrone e asincrone .....	84
<b>3.4</b>	<b>■ Macchine a stati finiti</b> .....	84
3.4.1	Esempio di progettazione di una FSM .....	85
3.4.2	Codifica degli stati .....	89
3.4.3	Macchine alla Moore e macchine alla Mealy .....	92
3.4.4	Fattorizzazione delle macchine a stati ..	94
3.4.5	Derivare una FSM da uno schema circuitale .....	95
3.4.6	Riassunto sulle FSM .....	98
<b>3.5</b>	<b>■ Temporizzazione della logica sequenziale</b> .....	98
3.5.1	La disciplina dinamica .....	99
3.5.2	Temporizzazione del sistema .....	100
3.5.3	Sfasamento del clock* .....	104
3.5.4	Metastabilità .....	107
3.5.5	Sincronizzatori .....	108
3.5.6	Formulazione del tempo di risoluzione* .....	110
<b>3.6</b>	<b>■ Parallelismo</b> .....	112
<b>3.7</b>	<b>■ Riassunto</b> .....	115
	Esercizi .....	116
	Domande di valutazione .....	120

## Capitolo 4 Linguaggi di descrizione dell'hardware

<b>4.1</b>	<b>■ Introduzione</b> .....	121
4.1.1	Moduli .....	121

4.1.2	Origini dei linguaggi .....	122
4.1.3	Simulazione e sintesi .....	123
<b>4.2</b>	<b>■ Logica combinatoria</b> .....	125
4.2.1	Operatori a singolo bit .....	125
4.2.2	Commenti e spazio vuoto .....	127
4.2.3	Operatori di riduzione .....	127
4.2.4	Assegnamento condizionale .....	128
4.2.5	Variabili interne .....	129
4.2.6	Precedenza .....	131
4.2.7	Numeri .....	132
4.2.8	Z e X .....	132
4.2.9	Concatenazione di bit .....	134
4.2.10	Ritardi .....	134
<b>4.3</b>	<b>■ Modellazione strutturale</b> .....	135
<b>4.4</b>	<b>■ Logica sequenziale</b> .....	138
4.4.1	Registri .....	138
4.4.2	Registri resettabili .....	140
4.4.3	Registri con abilitazione .....	141
4.4.4	Registri multipli .....	142
4.4.5	Latch .....	143
<b>4.5</b>	<b>■ Ancora logica combinatoria</b> .....	143
4.5.1	Istruzione case .....	145
4.5.2	Istruzione if .....	145
4.5.3	Tabelle delle verità con indifferenze .....	147
4.5.4	Assegnamenti bloccanti e non bloccanti .....	147
<b>4.6</b>	<b>■ Macchine a stati finiti</b> .....	153
<b>4.7</b>	<b>■ Tipi di dati*</b> .....	157
4.7.1	SystemVerilog .....	157
4.7.2	VHDL .....	158
<b>4.8</b>	<b>■ Moduli parametrici*</b> .....	160
<b>4.9</b>	<b>■ Testbench</b> .....	163
<b>4.10</b>	<b>■ Riassunto</b> .....	167
	Esercizi .....	168
	Domande di valutazione .....	174

## Capitolo 5 Blocchi costruttivi digitali

<b>5.1</b>	<b>■ Introduzione</b> .....	175
<b>5.2</b>	<b>■ Circuiti aritmetici</b> .....	175
5.2.1	Addizione .....	175
5.2.2	Sottrazione .....	181
5.2.3	Comparatori .....	182
5.2.4	ALU .....	183
5.2.5	Traslatori e rotatori .....	185
5.2.6	Moltiplicazione* .....	186

5.2.7	Divisione*	188
5.2.8	Lecture aggiuntive	189
<b>5.3</b>	<b>■ Sistemi di numerazione</b>	189
5.3.1	Numeri in virgola fissa	189
5.3.2	Numeri in virgola mobile*	190
<b>5.4</b>	<b>■ Blocchi costruttivi sequenziali</b>	193
5.4.1	Contatori	193
5.4.2	Registri a scorrimento	194
<b>5.5</b>	<b>■ Componenti di memoria</b>	196
5.5.1	Panoramica	197
5.5.2	Memoria ad accesso casuale dinamica	199
5.5.3	Memoria ad accesso casuale statica	199
5.5.4	Area e ritardo	199
5.5.5	Banchi di registri	200
5.5.6	Memorie a sola lettura	200
5.5.7	Reti logiche realizzate con componenti di memoria	202
5.5.8	Descrizione HDL delle memorie	202
<b>5.6</b>	<b>■ Matrici logiche</b>	204
5.6.1	Matrici logiche programmabili	204
5.6.2	Matrici di porte logiche programmabili sul campo	205
5.6.3	Realizzazione delle matrici di memoria*	209
<b>5.7</b>	<b>■ Riassunto</b>	210
	Esercizi	212
	Domande di valutazione	216

## Capitolo 6 Architettura

<b>6.1</b>	<b>■ Introduzione</b>	217
<b>6.2</b>	<b>■ Il linguaggio <i>assembly</i></b>	218
6.2.1	Istruzioni	218
6.2.2	Operandi: registri, memoria e costanti	220
<b>6.3</b>	<b>■ Programmare</b>	224
6.3.1	Istruzioni di elaborazione dati	224
6.3.2	<i>Flag</i> di condizione	226
6.3.3	Salti	227
6.3.4	Costrutti di selezione	229
6.3.5	Cicli	230
6.3.6	La memoria	232
6.3.7	Chiamate a sottoprogrammi	235
<b>6.4</b>	<b>■ Linguaggio macchina</b>	244
6.4.1	Istruzioni di elaborazione dati	244
6.4.2	Istruzioni di accesso a memoria	247

6.4.3	Istruzioni di salto	248
6.4.4	Modi di indirizzamento	250
6.4.5	Interpretare il linguaggio macchina	250
6.4.6	La potenza di un programma scritto in memoria	251

<b>6.5</b>	<b>■ Compilare, assemblare e caricare*</b>	252
6.5.1	La mappa di memoria	252
6.5.2	Compilazione	253
6.5.3	Assemblaggio	254
6.5.4	Collegamento	255
6.5.5	Caricamento	256
<b>6.6</b>	<b>■ Qualche dettaglio</b>	257
6.6.1	Caricamento di <i>literal</i>	257
6.6.2	NOP	258
6.6.3	Eccezioni	258
<b>6.7</b>	<b>■ Evoluzione dell'architettura ARM</b>	261
6.7.1	Set di istruzione <i>Thumb</i>	262
6.7.2	Istruzioni DSP	263
6.7.3	Istruzioni in virgola mobile	266
6.7.4	Istruzioni per il risparmio di potenza e per la sicurezza	267
6.7.5	Istruzioni SIMD	268
6.7.6	Architettura a 64 bit	269
<b>6.8</b>	<b>■ Un'altra prospettiva: l'architettura x86</b>	269
6.8.1	Registri x86	270
6.8.2	Operandi x86	271
6.8.3	Flag di stato	272
6.8.4	Istruzioni x86	272
6.8.5	Codifica delle istruzioni x86	272
6.8.6	Altre particolarità di x86	275
6.8.7	Il quadro generale	275
<b>6.9</b>	<b>■ Riassunto</b>	275
	Esercizi	276
	Domande di valutazione	282

## Capitolo 7 Microarchitettura

<b>7.1</b>	<b>■ Introduzione</b>	283
7.1.1	Stato architetturale e set di istruzioni	283
7.1.2	Progettazione	284
7.1.3	Microarchitetture	285
<b>7.2</b>	<b>■ Analisi delle prestazioni</b>	286

<b>7.3</b>	<b>■ Processore a ciclo singolo</b> . . . . .	287
7.3.1	Percorso dati a ciclo singolo. . . . .	287
7.3.2	Unità di controllo a ciclo singolo. . . . .	293
7.3.3	Istruzioni aggiuntive. . . . .	296
7.3.4	Analisi delle prestazioni. . . . .	298
<b>7.4</b>	<b>■ Processore multi ciclo</b> . . . . .	300
7.4.1	Percorso dati multi ciclo. . . . .	300
7.4.2	Unità di controllo multi ciclo. . . . .	306
7.4.3	Analisi delle prestazioni. . . . .	313
<b>7.5</b>	<b>■ Processore pipeline</b> . . . . .	316
7.5.1	Percorso dati pipeline. . . . .	318
7.5.2	Unità di controllo della pipeline. . . . .	319
7.5.3	Dipendenze. . . . .	321
7.5.4	Analisi delle prestazioni. . . . .	328
<b>7.6</b>	<b>■ Rappresentazione HDL*</b> . . . . .	330
7.6.1	Processore a ciclo singolo. . . . .	331
7.6.2	Altri blocchi costruttivi. . . . .	337
7.6.3	Testbench . . . . .	339
<b>7.7</b>	<b>■ Microarchitetture avanzate*</b> . . . . .	343
7.7.1	Pipeline lunghe. . . . .	344
7.7.2	Micro operazioni. . . . .	344
7.7.3	Previsione dei salti . . . . .	345
7.7.4	Processori superscalari . . . . .	347
7.7.5	Processore <i>out-of-order</i> . . . . .	349
7.7.6	Ridenominazione dei registri. . . . .	351
7.7.7	<i>Multithreading</i> . . . . .	352
7.7.8	Multiprocessori . . . . .	353
<b>7.8</b>	<b>■ Uno sguardo al mondo reale: evoluzione dell'architettura ARM*</b> . . . . .	355
<b>7.9</b>	<b>■ Riassunto</b> . . . . .	359
	Esercizi . . . . .	361
	Domande di valutazione. . . . .	364

## Capitolo 8 Sistemi di memoria

<b>8.1</b>	<b>■ Introduzione</b> . . . . .	365
<b>8.2</b>	<b>■ Analisi delle prestazioni del sistema di memoria</b> . . . . .	368
<b>8.3</b>	<b>■ Memoria cache</b> . . . . .	370
8.3.1	Quali dati devono essere memorizzati nella cache? . . . . .	370
8.3.2	Come si verifica se un dato è in cache? . . . . .	371
8.3.3	Quale dato viene sostituito? . . . . .	378
8.3.4	Progetto di cache avanzate* . . . . .	378
8.3.5	Evoluzione delle cache di ARM . . . . .	382

<b>8.4</b>	<b>■ Memoria virtuale</b> . . . . .	382
8.4.1	Traduzione dell'indirizzo . . . . .	384
8.4.2	La tabella delle pagine . . . . .	386
8.4.3	Il <i>Translation Lookaside Buffer</i> (TLB). . . . .	387
8.4.4	Protezione della memoria . . . . .	388
8.4.5	Politiche di sostituzione* . . . . .	389
8.4.6	Tabelle delle pagine multi livello* . . . . .	389
<b>8.5</b>	<b>■ Riassunto</b> . . . . .	391
	<b>Epilogo</b> . . . . .	391
	Esercizi . . . . .	392
	Domande di valutazione. . . . .	396

## Capitolo 9 Sistemi di ingresso/uscita



Capitolo disponibile online

<b>9.1</b>	<b>■ Introduzione</b> . . . . .	1
<b>9.2</b>	<b>■ I/O mappato in memoria</b> . . . . .	1
<b>9.3</b>	<b>■ I/O nei sistemi embedded</b> . . . . .	3
9.3.1	Il sistema a singolo chip BCM2835. . . . .	3
9.3.2	Driver di dispositivo . . . . .	4
9.3.3	I/O digitali di uso generale. . . . .	7
9.3.4	I/O seriale. . . . .	9
9.3.5	Timer . . . . .	18
9.3.6	I/O analogici . . . . .	19
9.3.7	Interrupt . . . . .	25
<b>9.4</b>	<b>■ Altre periferiche di microcontrollori</b> . . . . .	26
9.4.1	Display di caratteri a cristalli liquidi. . . . .	26
9.4.2	Monitor VGA. . . . .	29
9.4.3	Comunicazioni wireless Bluetooth . . . . .	33
9.4.4	Controllo di motori. . . . .	35
<b>9.5</b>	<b>■ Interfacce a bus</b> . . . . .	43
9.5.1	AHB-Lite . . . . .	43
9.5.2	Esempio di interfaccia a memoria e periferica . . . . .	44
<b>9.6</b>	<b>■ Sistemi di I/O del PC</b> . . . . .	47
9.6.1	USB. . . . .	48
9.6.2	PCI e PCI Express . . . . .	48
9.6.3	Memoria DDR3 . . . . .	49
9.6.4	Interconnessione in rete. . . . .	49
9.6.5	SATA . . . . .	50
9.6.6	Interfacciamento a un PC. . . . .	50
<b>9.7</b>	<b>■ Riassunto</b> . . . . .	52

## Appendice A Realizzazione dei sistemi digitali

A.1	■ Introduzione	397
A.2	■ La logica 74xx	397
A.2.1	Porte logiche	398
A.2.2	Altre funzioni	398
A.3	■ Logica programmabile	398
A.3.1	PROM	398
A.3.2	PLA	402
A.3.3	FPGA	402
A.4	■ Circuiti integrati specifici per un'applicazione	404
A.5	■ Data sheet	404
A.6	■ Famiglie logiche	408
A.7	■ Packaging e assemblaggio	411
A.8	■ Linee di trasmissione	414
A.8.1	Terminazione adattata	416
A.8.2	Terminazione aperta	417
A.8.3	Terminazione cortocircuitata	418
A.8.4	Terminazione non adattata	418
A.8.5	Quando serve usare i modelli delle linee di trasmissione	420
A.8.6	Corrette terminazioni delle linee di trasmissione	421
A.8.7	Espressione di $Z_0^*$	422
A.8.8	Espressione del coefficiente di riflessione*	423
A.8.9	Riassumendo	424
A.9	■ Aspetti economici	425

## Appendice B Istruzioni ARM

B.1	■ Istruzioni di elaborazione dati	427
B.1.1	Istruzioni di moltiplicazione	427
B.2	■ Istruzioni di accesso a memoria	429
B.3	■ Istruzioni di salto	429
B.4	■ Istruzioni varie	430
B.5	■ Flag di condizione	430

## Appendice C Programmazione in C

Capitolo disponibile online

C.1	■ Introduzione	1
C.2	■ Benvenuti al linguaggio C	2
C.2.1	Struttura di un programma C	3
C.2.2	Esecuzione di un programma C	3
C.3	■ Compilazione	4
C.3.1	Commenti	4
C.3.2	#define	5
C.3.3	#include	5
C.4	■ Variabili	6
C.4.1	Tipi di dati primitivi	7
C.4.2	Variabili globali e locali	8
C.4.3	Inizializzazione delle variabili	9
C.5	■ Operatori	10
C.6	■ Chiamate di funzione	12
C.7	■ Istruzioni di controllo del flusso di esecuzione	13
C.7.1	Istruzioni condizionali	13
C.7.2	Cicli	15
C.8	■ Altri tipi di dati	17
C.8.1	Puntatori	17
C.8.2	Array	18
C.8.3	Caratteri	22
C.8.4	Stringhe	23
C.8.5	Strutture	24
C.8.6	typedef	25
C.8.7	Allocazione dinamica della memoria*	26
C.8.8	Liste collegate*	27
C.9	■ Librerie standard	29
C.9.1	stdio	29
C.9.2	stdlib	33
C.9.3	math	34
C.9.4	string	35
C.10	■ Opzioni del compilatore e argomenti nella riga di comando	35
C.10.1	Compilare più file sorgente C	35
C.10.2	Opzioni del compilatore	35
C.10.3	Argomenti nella riga di comando	35
C.11	■ Errori frequenti	36
	Indice analitico	431





# 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 espone 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](http://online.universita.zanichelli.it/harris-arm)).

### LE RISORSE MULTIMEDIALI

All'indirizzo [online.universita.zanichelli.it/harris-arm](http://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.

