

Computers as Components

Principles of Embedded Computing System Design

Fourth Edition

Marilyn Wolf



ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Morgan Kaufmann is an imprint of Elsevier



Contents

Foreword to the First Edition	xvii
Preface to the First Edition	xix
Preface to the Second Edition	xxiii
Preface to the Third Edition	xxv
Preface to the Fourth Edition	xxvii

CHAPTER 1 Embedded Computing	1
1.1 Introduction	1
1.2 Complex systems and microprocessors	1
1.2.1 Embedding computers	2
1.2.2 Characteristics of embedded computing applications	4
1.2.3 Why use microprocessors?	5
1.2.4 Cyber-physical systems	7
1.2.5 Safety and security	8
1.2.6 Challenges in embedded computing system design ...	10
1.2.7 Performance of embedded computing systems	11
1.3 The embedded system design process	12
1.3.1 Requirements	14
1.3.2 Specification	18
1.3.3 Architecture design	19
1.3.4 Designing hardware and software components	21
1.3.5 System integration	22
1.3.6 Formalisms for system design	22
1.3.7 Structural description	23
1.3.8 Behavioral description	28
1.4 Design example: model train controller	31
1.4.1 Requirements	32
1.4.2 DCC	33
1.4.3 Conceptual specification	35
1.4.4 Detailed specification	38
1.4.5 Lessons learned	45
1.5 A guided tour of this book	46
1.5.1 Chapter 2: Instruction sets	46
1.5.2 Chapter 3: CPUs	47
1.5.3 Chapter 4: Computing platforms	47
1.5.4 Chapter 5: Program design and analysis	48

1.5.5	Chapter 6: Processes and operating systems	49
1.5.6	Chapter 7: System design techniques	50
1.5.7	Chapter 8: Internet-of-Things	50
1.5.8	Chapter 9: Automotive and aerospace systems	50
1.5.9	Chapter 10: Embedded multiprocessors	50
1.6	Summary	51
	What we learned	51
	Further reading	51
	Questions.....	52
	Lab exercises	53
CHAPTER 2	Instruction Sets	55
2.1	Introduction.....	55
2.2	Preliminaries	55
2.2.1	Computer architecture taxonomy.....	56
2.2.2	Assembly languages	58
2.2.3	VLIW processors.....	60
2.3	ARM processor.....	62
2.3.1	Processor and memory organization.....	62
2.3.2	Data operations.....	63
2.3.3	Flow of control.....	70
2.3.4	Advanced ARM features.....	76
2.4	PICmicro midrange family.....	77
2.4.1	Processor and memory organization.....	77
2.4.2	Data operations.....	78
2.4.3	Flow of control.....	81
2.5	TI C55x DSP	82
2.5.1	Processor and memory organization.....	82
2.5.2	Addressing modes	84
2.5.3	Data operations.....	88
2.5.4	Flow of control.....	89
2.5.5	C coding guidelines.....	91
2.6	TI C64x.....	92
2.7	Summary	95
	What we learned.....	96
	Further reading	96
	Questions.....	96
	Lab exercises	98

CHAPTER 3	CPUs	99
3.1	Introduction	99
3.2	Programming input and output	99
3.2.1	Input and output devices	100
3.2.2	Input and output primitives	102
3.2.3	Busy-wait I/O	103
3.2.4	Interrupts	104
3.3	Supervisor mode, exceptions, and traps	117
3.3.1	Supervisor mode	117
3.3.2	Exceptions	118
3.3.3	Traps	118
3.4	Coprocessors	119
3.5	Memory system mechanisms	119
3.5.1	Caches	120
3.5.2	Memory management units and address translation	126
3.6	CPU performance	131
3.6.1	Pipelining	131
3.6.2	Cache performance	136
3.7	CPU power consumption	137
3.7.1	CMOS power consumption	137
3.7.2	Power management modes	138
3.7.3	Program-level power management	141
3.8	Safety and security	141
3.9	Design example: data compressor	143
3.9.1	Requirements and algorithm	143
3.9.2	Specification	145
3.9.3	Program design	147
3.9.4	Testing	153
3.10	Summary	154
	What we learned	155
	Further reading	155
	Questions	155
	Lab exercises	159
CHAPTER 4	Computing Platforms	161
4.1	Introduction	161
4.2	Basic computing platforms	161
4.2.1	Platform hardware components	162
4.2.2	Platform software components	164

4.3	The CPU bus	165
4.3.1	Bus organization and protocol	165
4.3.2	DMA	171
4.3.3	System bus configurations.....	175
4.4	Memory devices and systems.....	177
4.4.1	Memory system organization.....	179
4.5	Designing with computing platforms.....	181
4.5.1	Example platforms	181
4.5.2	Choosing a platform.....	183
4.5.3	Intellectual property	184
4.5.4	Development environments.....	185
4.5.5	Watchdog timers.....	186
4.5.6	Debugging techniques	187
4.5.7	Debugging challenges	189
4.6	Consumer electronics architecture	191
4.6.1	Consumer electronics use cases and requirements	191
4.6.2	File systems	193
4.7	Platform-level performance analysis.....	194
4.8	Platform-level power management	200
4.9	Design example: alarm clock.....	201
4.9.1	Requirements.....	201
4.9.2	Specification	202
4.9.3	System architecture	205
4.9.4	Component design and testing.....	208
4.9.5	System integration and testing.....	208
4.10	Design example: audio player.....	208
4.10.1	Theory of operation and requirements	208
4.10.2	Specification	211
4.10.3	System architecture.....	211
4.10.4	Component design and testing.....	214
4.10.5	System integration and debugging.....	215
4.11	Summary.....	215
	What we learned.....	215
	Further reading	215
	Questions.....	215
	Lab exercises	219
CHAPTER 5	Program Design and Analysis	221
5.1	Introduction	221
5.2	Components for embedded programs	222

5.2.1	State machines.....	222
5.2.2	Circular buffers and stream-oriented programming	224
5.2.3	Queues and producer/consumer systems	229
5.3	Models of programs	231
5.3.1	Data flow graphs.....	232
5.3.2	Control/data flow graphs	234
5.4	Assembly, linking, and loading.....	236
5.4.1	Assemblers.....	238
5.4.2	Linking	242
5.4.3	Object code design.....	243
5.5	Compilation techniques.....	244
5.5.1	The compilation process	244
5.5.2	Basic compilation methods	246
5.5.3	Compiler optimizations.....	254
5.6	Program-level performance analysis.....	262
5.6.1	Analysis of program performance	264
5.6.2	Measurement-driven performance analysis	268
5.7	Software performance optimization.....	271
5.7.1	Basic loop optimizations.....	271
5.7.2	Cache-oriented loop optimizations	273
5.7.3	Performance optimization strategies.....	275
5.8	Program-level energy and power analysis and optimization	276
5.9	Analysis and optimization of program size	280
5.10	Program validation and testing	281
5.10.1	Clear-box testing	282
5.10.2	Black-box testing	289
5.10.3	Evaluating functional tests.....	290
5.11	Safety and security	291
5.12	Design example: software modem.....	292
5.12.1	Theory of operation and requirements	292
5.12.2	Specification	295
5.12.3	System architecture.....	295
5.12.4	Component design and testing.....	296
5.12.5	System integration and testing.....	297
5.13	Design example: digital still camera.....	297
5.13.1	Theory of operation and requirements	297
5.13.2	Specification	301
5.13.3	System architecture	305

5.13.4	Component design and testing	308
5.13.5	System integration and testing	308
5.14	Summary	308
	What we learned	308
	Further reading	309
	Questions.....	309
	Lab exercises	318
CHAPTER 6	Processes and Operating Systems.....	321
6.1	Introduction	321
6.2	Multiple tasks and multiple processes	322
6.3	Multirate systems	324
6.3.1	Timing requirements on processes.....	326
6.3.2	CPU usage metrics	330
6.3.3	Process state and scheduling.....	331
6.3.4	Running periodic processes.....	332
6.4	Preemptive real-time operating systems	334
6.4.1	Two basic concepts	335
6.4.2	Processes and context.....	336
6.4.3	Processes and object-oriented design	339
6.5	Priority-based scheduling	339
6.5.1	Rate-monotonic scheduling.....	341
6.5.2	Earliest-deadline-first scheduling.....	345
6.5.3	RMS versus EDF.....	349
6.5.4	Shared resources.....	349
6.5.5	Priority inversion	351
6.5.6	Scheduling for low power	352
6.5.7	A closer look at our modeling assumptions	352
6.6	Interprocess communication mechanisms.....	355
6.6.1	Shared memory communication	355
6.6.2	Message passing	356
6.6.3	Signals	357
6.6.4	Mailboxes	358
6.7	Evaluating operating system performance.....	359
6.8	Example real-time operating systems	363
6.9	Design example: telephone answering machine.....	369
6.9.1	Theory of operation and requirements.....	369
6.9.2	Specification	372
6.9.3	System architecture	374
6.9.4	Component design and testing	377
6.9.5	System integration and testing	377

6.10	Design example: engine control unit	378
6.10.1	Theory of operation and requirements	378
6.10.2	Specification	379
6.10.3	System architecture	380
6.10.4	Component design and testing	382
6.10.5	System integration and testing	382
6.11	Summary	383
	What we learned	383
	Further reading	383
	Questions	383
	Lab exercises	389
CHAPTER 7	System Design Techniques	391
7.1	Introduction	391
7.2	Design methodologies	391
7.2.1	Why design methodologies?	391
7.2.2	Design flows	393
7.3	Requirements analysis	400
7.4	Specifications	401
7.4.1	Control-oriented specification languages	401
7.4.2	Advanced specifications	404
7.5	System analysis and architecture design	407
7.5.1	CRC cards	407
7.6	Dependability, safety, and security	410
7.6.1	Examples	411
7.6.2	Quality assurance techniques	413
7.6.3	Verifying the specification	415
7.6.4	Design reviews	417
7.6.5	Safety-oriented methodologies	418
7.7	Summary	421
	What we learned	421
	Further reading	421
	Questions	421
	Lab exercises	422
CHAPTER 8	Internet-of-Things Systems	423
8.1	Introduction	423
8.2	IoT system applications	423
8.3	IoT system architectures	425

8.4	Networks for IoT	427
8.4.1	The OSI model	427
8.4.2	Internet Protocol	428
8.4.3	IoT networking concepts	430
8.4.4	Bluetooth and Bluetooth Low Energy	434
8.4.5	802.15.4 and ZigBee	437
8.4.6	Wi-Fi	438
8.5	Databases and timewheels	440
8.5.1	Databases	440
8.5.2	Timewheels	443
8.6	Example: smart home	444
8.7	Summary	446
	What we learned	447
	Further reading	447
	Questions	447
	Lab exercises	448
CHAPTER 9	Automotive and Aerospace Systems	449
9.1	Introduction	449
9.2	Networked control systems in cars and airplanes	449
9.3	Vehicular networks	453
9.3.1	CAN bus	453
9.3.2	Other automotive networks	455
9.4	Safety and security	457
9.5	Summary	459
	What we learned	459
	Further reading	459
	Questions	459
	Lab exercises	460
CHAPTER 10	Embedded Multiprocessors	461
10.1	Introduction	461
10.2	Why multiprocessors?	461
10.3	Categories of multiprocessors	464
10.4	MPSOCs and shared memory multiprocessors	466
10.4.1	Heterogeneous shared memory multiprocessors	466
10.4.2	Accelerators	467
10.4.3	Accelerator performance analysis	469
10.4.4	Scheduling and allocation	474
10.4.5	System integration	476
10.4.6	Debugging	480

10.5	Design example: video accelerator	481
10.5.1	Video compression	481
10.5.2	Algorithm and requirements	483
10.5.3	Specification	485
10.5.4	Architecture	486
10.5.5	Component design	488
10.5.6	System testing	489
10.6	Application example: optical disk	489
10.7	Summary	494
	What we learned	494
	Further reading	494
	Questions	495
	Lab exercises	496
	Glossary	497
	References	517
	Index	527