HZ BOOKS
华章教育

PEARSON

经 典 原 版 书 库

# 嵌入式微控制器与处理器设计

（英文版）

embedded microcontrollers and processor design

GREG OSBORN

（美）Greg Osborn 著

机械工业出版社
China Machine Press

经 典 原 版 书 库

# 嵌入式微控制器与处理器设计

（英文版）

# Embedded Microcontrollers and Processor Design



（美）Greg Osborn 著

机械工业出版社
China Machine Press

**本书版权登记号：图字：01-2009-7321**

**图书在版编目（CIP）数据**

嵌入式微控制器与处理器设计（英文版）/（美）奥斯本（Osborn，G.）著．—北京：机械工业出版社，2010.1

（经典原版书库）

书名原文：Embedded Microcontrollers and Processor Design

ISBN 978-7-111-29250-0

Ⅰ．嵌…　Ⅱ．奥…　Ⅲ．①微控制器－系统设计－英文　②微处理器－系统设计－英文　Ⅳ．TP332

中国版本图书馆 CIP 数据核字（2009）第 228290 号

机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码　100037）

责任编辑：迟振春

北京京师印务有限公司印刷

2010 年 1 月第 1 版第 1 次印刷

150mm × 214mm · 14.125 印张

标准书号：ISBN 978-7-111-29250-0

定价：49.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991；88361066

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

读者信箱：hzjsj@hzbook.com

# 出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章分社较早意识到“出版要为教育服务”。自 1998 年开始，华章分社就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum，Bjarne Stroustrup，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章分社欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www. hzbook. com
电子邮件：hzjsj@hzbook. com
联系电话：(010)88379604
联系地址：北京市西城区百万庄南街1号
邮政编码：100037


华章教育
华章科技图书出版中心

# PREFACE

Microcontrollers have become ubiquitous elements of everyday life. Most of the electronic products we use in daily life have a microcontroller tucked inside. They are used in household appliances, automobiles, copiers, cell phones, and even used to control powerful locomotives. Where electricity is used, you will find a microcontroller!

There are many technical books on the subject of microcontrollers. Why develop a new book? All the popular microcontroller chips and architectures have "how-to" books in print. The focus of this text is a broader introduction to the student of microcontroller processor technology, both in single-chip and intellectual property form.

Many electronic engineering students are required to take a course in embedded system design with microcontrollers. Devices such as the Intel 8051, ZiLOG Z8, or Microchip PIC are most often the microcontroller of choice because of their widespread popularity. Also, they have extensive, inexpensive tools available to provide design support.

A wider variety of choices for an embedded microcontroller-based design face the engineer today, not only the popular single-chip devices, but also intellectual property cores for ASIC system-on-a-chip (SoC) design. Although the computer world has solidified around Intel-based architectures, the microcontroller world continues to evolve with innovative new designs.

This book is organized into three major sections: an introduction to microcontroller architectures, single-chip microcontrollers, and embedded IP cores. Each chapter within the section begins with learning objectives. Questions are provided at the end of the section to monitor progress.

Not only are specific chips and cores covered, but also this book introduces the student to the concepts of microcontroller architectures: for instance, how the concept of the computing devices evolved, and why different types of devices are used in design.

Single-chip microcontrollers as referenced in this book are typical commercial high-volume classical designs. Certainly, myriad parts are also available, particularly from "fabless" design houses. Microcontroller cores are in reference to established system-on-a-chip intellectual property cores and marketed as such. In architectural discussions in this book, the term "processor" incorporates both the "processor" element of "single-chip microcontrollers" and "IP cores."

This book is intended to provide the reader with an introduction to single-chip and embedded microcontroller processor design. The difference between architectures of the CISC- and RISC-based processors is discussed. Single-chip microcontroller design flows and embedded processor design flows are discussed.

The 16-bit Freescale MC9S12X family of single-chip microcontrollers is covered in detail. The RISC-based PIC18F4520 and the ZiLOG Z8 Encore! 8-bit microcontrollers

are also discussed. The peripherals that are available with various members of the families are explained.

The concept of instruction set architecture (ISA) is introduced to develop an understanding of the commonality of the CISC and RISC processor families, respectively. This is expanded to the design of SoC embedded controllers-based core IP using the ISAs of ARM and MIPS. The ARM10TDMI and MIPS32 4KE™ IP cores are presented in some detail.

Configurable processor technology is increasingly important, particularly in the design of higher performing consumer products. It allows customization of the core processor, which can have both performance and power impact on the SoC embedded design. The Tensillica Xtensa LX2 Series configurable processor is covered.

A discussion of derivative RISC application-specific processors is presented. An overview of a digital signal processors (DSP), including the Texas Instruments' TMS320C55 and Analog Devices' ADSP-BR533 Blackfin is given. The methodology of the engineering design flow is also covered. Different tools available to the engineer for the design process are discussed. An example of using an integrated design environment (IDE) for single-chip microcontroller is presented.

Software programming for microcontroller design can be as simple as a program for controlling lawn sprinklers to a complex RTOS for robots. Programming techniques from simple polling loops to multilevel interrupt driven systems are discussed. Many single-chip microcontrollers include functional blocks for serial I/O. They are primarily used to communicate data. The UART, $I^2C$, $I^2S$, CAN/LIN SPI, and USB peripheral functions are discussed.

System-on-a-chip design requires a close relationship with the semiconductor foundry. As a fabless design technology, SoCs need specialized engineering techniques to integrate the functions needed for the chip. Combining IP functional blocks available from the foundry with those from independent companies to achieve a working chip is a complex process.

This book is intended as an introductory understanding of microcontrollers in single-chip and embedded forms. The concept of ISA is developed with the methodology for product design. System-on-a-chip design is introduced through the use of intellectual property.

Microcontroller design, at any level of abstraction, is based on a balancing of available technologies. The primary three technologies this book will focus on are processor, memory, and software: processor technology as it is defined in terms of semiconductor fabrication capability; memory technology as it is implemented in a hierarchical storage structure; and software technology as it is implemented in the form of assembler and optimizing compilers.

Within the scope of this book, generalizations are taken as they relate to characteristics of microcontroller-based design. In general, CISC-based processors have more complex instructions than RISC-based processors. In general, RISC registers sets are orthogonal when compared to CISC. In general, optimizing C compilers are more efficient for RISC than CISC.

RISC and CISC are considered in their global context of instruction set architecture (ISA). New innovations in architecture, such as VLIW and EPIC, are mentioned for comparison. The focus of this book is on microcontroller technology, which is predominately RISC- or CISC-based. This will provide the basic knowledge needed for the student to understand other derivative ISAs.

Microcontrollers have, at their heart, a microprocessor. In this book, the term *processor* is used in a broad sense. Whether implemented as core IP in an SoC, or in traditional single-chip form, the basic concept of the processor is the same. MIPS32 4KE™ IP can be incorporated as single chips from NEC or an SoC in a CISCO router. They are implemented differently but are the same architecturally.

This book is intended as an introduction to the topic of microcontroller technology for college engineering students. It is not a hardware reference manual. It is not intended as a series of application notes. The concepts presented are in general form. This will allow a broad group of engineering students to understand the basic concepts and apply them to real-world situations.

An online instructor's manual is available for instructors using this text for a course. To access supplementary materials online, instructors need to request an instructor access code. Go to **www.pearsonhighered.com/irc,** where you can register for an instructor access code. Within 48 hours after registering, you will receive a confirming e-mail, including an instructor access code. Once you have received your code, go to the site and log on for full instructions on downloading the materials you wish to use.

The author thanks the following reviewers of the manuscript: C. Richard G. Helps, Brigham Young University; James Streib, Illinois College; Chao-Ying Wang, DeVry University—Columbus; and Richard Warren, Vermont Technical College.

**Greg Osborn**

# CONTENTS


**CHAPTER 1** **EMBEDDED PROCESSORS** **1**

**1.0 Microcontrollers 1**
- 1.1 Microcontroller Markets 1
- 1.2 Data Path 2
- 1.3 Commercial Microcontrollers 2
- 1.4 SoC Core Processors 2
- 1.5 Relative SoC Unit Volumes 3
- 1.6 Very-Large-Scale Integration (VLSI) Chip Design Tools 4
- 1.7 Intellectual Property 4
- 1.8 Instruction Set Architecture 6
- 1.9 Return on Investment 6
- 1.10 Semiconductor Technology Developments 7

**CHAPTER 2** **MICROCONTROLLER ARCHITECTURE** **11**

**2.0 Computer on a Chip 11**
- 2.1 John von Neumann 12
  - 2.1.1 von Neumann Architecture 12
- 2.2 Computer Architectures 13
  - 2.2.1 CISC and RISC 13
- 2.3 Semiconductor Technology 14
  - 2.3.1 Small-Scale Integration 14
  - 2.3.2 Hardware Bus 14
  - 2.3.3 Intelligent Peripherals 15
  - 2.3.4 Standardized I/O Interfaces 15
- 2.4 MSI and LSI 16
- 2.5 Electronic Calculator 17
  - 2.5.1 Programmable Calculator 17
- 2.6 Microprocessors 18
  - 2.6.1 Application-Oriented Processing 18
  - 2.6.2 Intel i4004 19
  - 2.6.3 Intel i8080 19
- 2.7 Microprocessor Peripherals 20
  - 2.7.1 Microcomputer 20
- 2.8 i8051 Microcontroller 21

2.9 RISC Introduction 22
2.9.1 RISC Processors 22
2.9.2 RISC Synergy 23
2.9.3 RISC Marketing 24
2.10 Fabless Semiconductor Company 24
2.10.1 RISC as Intellectual Property 25
2.10.2 RISC Technology Curve 25
2.11 Embedded Controller IP 26
2.11.1 CISC IP 27
2.11.2 RISC IP 27
2.11.3 Third-Party IP 27
2.12 Application Specific Processors 27
2.13 Summary 28

**CHAPTER 3 EMBEDDED MICROCONTROLLER TECHNOLOGY 30**

**3.0 Integrated Circuits 30**
3.1 Moore's Law 30
3.1.1 Microprocessor Performance 31
3.1.2 Enabling Technologies 32
3.1.3 Amdahl's Law 33
3.1.4 Technology Convergence 33
3.2 Design Abstraction 34
3.2.1 Instruction Set Architectures 34
3.2.2 Processor Family Tree 35
3.3 RISC and CISC 35
3.3.1 Processor Technology 36
3.3.2 Performance Measurement 36
3.3.3 Program Instructions 36
3.3.4 Cost per Instruction 37
3.3.5 Microcoded Instructions 37
3.4 Memory Technology 38
3.4.1 Locality 39
3.4.2 Memory Hierarchy 39
3.4.3 Cache Memory 40
3.4.4 L1 and L2 Cache 40
3.4.5 Data Registers 41
3.4.6 Instruction Queues 41
3.4.7 Branch Instructions 41
3.4.8 Memory Latency 42
3.4.9 Cache Blocks 42
3.5 Instruction Processing 44
3.5.1 Symbolic Assembly 44
3.5.2 Program Compilers 45
3.5.3 Hard-Coded Instructions 45
3.6 Program Design 45
3.6.1 Program Code Size Creep 46
3.6.2 CISC Instruction Set 46
3.7 Unified Instruction Set 47
3.7.1 Industry Standard Software 47
3.7.2 Instruction Set Extensions 47
3.8 RISC Instruction Set Architecture 48
3.8.1 Microcode 48
3.8.2 Micro Instruction Cycles 48
3.8.3 Application Specific Instructions 48
3.8.4 Single-Cycle Instructions 49

3.9 Processor Logic 49
3.9.1 Synchronous Logic 50
3.9.2 Register Sets 50
3.9.3 Orthogonal Registers 50
3.9.4 Register Optimization 50
3.9.5 Load/Store Data Operations 51
3.10 Processor Functional Partitioning 51
3.10.1 Instruction Pipelining 51
3.10.2 Execution Units 52
3.10.3 Pipeline Stages 52
3.10.4 Pipeline Throughput 53
3.10.5 Sequential Execution 54
3.10.6 Branch Execution 54
3.11 Five-Stage Pipeline 54
3.11.1 Instruction Pipeline Stalls 56
3.11.2 Branch Prediction Table 56
3.11.3 Data Pipeline Stall 56
3.12 Summary 56

**CHAPTER 4 MICROCONTROLLER FUNCTIONS** **58**

**4.0 Device Functions 58**
4.1 Transistor Technology 59
4.1.1 CMOS Transistor 59
4.1.2 CMOS Power Consumption 60
4.1.3 Packaging 60
4.1.4 Operating Temperature Range 61
4.2 Memory Technologies 61
4.2.1 DRAM 62
4.2.2 SRAM 62
4.2.3 NVRWM 63
4.2.4 EEPROM 63
4.2.5 FLASH Technology 64
4.2.6 ROM 64
4.3 Hardware Features 64
4.3.1 Configuration Word 64
4.3.2 Oscillator Types 65
4.3.3 Reset 66
4.3.4 Standby Modes 66
4.3.5 Low-Power Consumption 67
4.3.6 Watchdog Timer 67
4.3.7 In-Circuit Programming 67
4.4 Data Input/Output 68
4.4.1 Parallel I/O 68
4.4.2 Tri-State Bit I/O 69
4.4.3 Memory Mapped I/O 69
4.5 Synchronous Serial Communication 70

**CHAPTER 5 PROGRAM DESIGN** **72**

**5.0 Program Design 72**
5.1 Polling Program 73
5.1.1 Program Flow 73
5.1.2 Program Timing 74
5.1.3 Sequential Tasks 74

5.1.4 Task Timing 75
5.1.5 Multiple Sequential Tasks 76
5.2 Interrupts 76
5.2.1 Asynchronous Timing 77
5.2.2 Interrupt Enable 77
5.2.3 Machine State 78
5.2.4 Latency 78
5.2.5 Context Switch 79
5.2.6 Interrupt Vector 79
5.2.7 Nested Interrupts 80
5.2.8 Critical Code 80
5.2.9 Interrupt Service Routine 82
5.3 Real-Time Operating System 82
5.4 Event-Driven System 83
5.5 Nucleus 83
5.6 System Layering 84
5.7 Risk 84

**CHAPTER 6 HARDWARE/SOFTWARE DEBUG 86**

**6.0 Hardware/Software Debug 86**
6.1 COTS Controller Tools 87
6.2 Embedded Controller Tools 88
6.3 First Silicon 88
6.4 Board-Level Probes 89
6.5 Debug Process Steps 90
6.5.1 Software Editor 90
6.5.2 Compilation 91
6.5.3 Program Build 92
6.5.4 Simulator 92
6.5.5 In-Circuit Emulation 93
6.6 SoC Debug Strategies 94
6.6.1 SoC Software Debug 95
6.6.2 Core-Level Debug 95
6.6.3 JTAG/EJTAG Specification 96
6.7 ARM SoC Debug 96
6.8 MIPS SoC Debug 98
6.8.1 EJTAG Functions 99

**CHAPTER 7 SERIAL DATA COMMUNICATIONS 101**

**7.0 Serial Data Communication 101**
7.1 UART 101
7.1.1 Asynchronous Mode 102
7.1.2 Transmit/Receive Buffers 104
7.2 SPI – Serial Peripheral Interface 105
7.3 $I^2C$ – Inter-IC Bus 108
7.3.1 How the $I^2C$ Bus Works 109
7.3.2 $I^2C$ Bus Terminology 110
7.3.3 Terminology for Bus Transfer 111
7.4 CAN—Controller Area Network 112
7.5 LIN—Local Interconnect Network 115
7.6 $I^2S$—Inter-IC Sound 116
7.6.1 $I^2S$ Serial Data 117
7.6.2 $I^2S$ Word Select 117
7.6.3 $I^2S$ Bus Timing 117

7.7 IrDA – Infrared Data Association 118
7.7.1 IrDA Stack 119
7.8 USB – Universal Peripheral Bus 119
7.8.1 USB Topology 120
7.8.2 USB Architecture 121
7.8.3 USB Physical Connection 122
7.8.4 USB Interface 122
7.8.5 USB 2.0 Specification 122
7.9 Bluetooth 122
7.9.1 Bluetooth Architecture 124
7.9.2 Bluetooth Frequency 124
7.9.3 Bluetooth Network 125

**CHAPTER 8 ANALOG TO DIGITAL CONVERSION** **127**

**8.0 Analog-to-Digital Conversion 127**
8.1 Analog-to-Digital Conversion Overview 127
8.2 Transducers 129
8.3 Low-Pass Filter 130
8.3.1 Active Filter 131
8.4 Sampling 131
8.5 Shannon's Sampling Theorem 132
8.6 What is an ADC? 133
8.6.1 ADC Converter Resolution 134
8.6.2 LSB and MSB Defined 134
8.6.3 Quantization 135
8.6.4 Quantization Error 137
8.6.5 Offset Error 138
8.6.6 Differential Nonlinearity 139
8.6.7 Missing Codes 139
8.6.8 SNR—Signal-to-Noise Ratio 140
8.7 Analog-to-Digital Conversion Algorithms 141
8.7.1 Successive Approximation 142
8.7.2 SAR ADC Architecture 142
8.7.3 Flash ADC 145
8.7.4 Integrating ADCs 146
8.7.4.1 Single-Slope Architecture 146
8.7.4.2 Dual-Slope Architecture 147
8.7.5 Pipeline ADC 148
8.7.6 Sigma-Delta 149
8.8 Oversampling 150

**CHAPTER 9 DIGITAL SIGNAL PROCESSING** **153**

**9.0 Digital Signal Processing 153**
9.1 What is a DSP? 154
9.1.1 Filtering and Synthesis 155
9.1.2 DSP Performance 155
9.1.3 Analog Signal Conversion 156
9.2 DSP Controller Architectures 156
9.3 Analog Filters 159
9.3.1 Filter Performance Measurements 159
9.3.2 Time Domain Response 161
9.3.3 Analog Low-Pass Filter 161
9.3.4 Active Analog Filters 162
9.3.5 Active Filter Comparison 163

9.4 Digital Filters 164
9.4.1 Finite Input Response Filter 164
9.4.2 FIR Filter Implementation 166
9.4.3 Convolution 167
9.4.4 Infinite Impulse Response Filter 169
9.5 Signal Transformation 170
9.5.1 Phasor Model 170
9.5.2 Fourier Series 171
9.5.3 Discrete Fourier Series 171
9.5.4 Fourier Transform 171
9.5.5 Discrete Fourier Transform 172
9.6 Fast Fourier Transform 174
9.6.1 FFT Implementation 174
9.6.2 DFT "Butterfly" 175
9.7 Table Addressing 176

**CHAPTER 10 FUZZY LOGIC 178**

**10.0 Fuzzy Logic 178**
10.1 Fuzzy Logic Method 180
10.2 Fuzzy Perception 180
10.3 Fuzzy Logic Terminology 181
10.4 Fuzzy Expert System 182
10.4.1 The Inference Process 183
10.4.2 Fuzzification 183
10.4.3 Inference 184
10.4.4 Composition 184
10.4.5 Defuzzification 185
10.5 Linguistic Variables 185
10.5.1 Using Linguistic Variables 187
10.5.2 Anatomy of a Fuzzy Rule 188
10.5.3 Logically Combining Linguistic Variables 188
10.6 PID Controller 189
10.6.1 Linguistic Time of Day 189
10.6.2 Linguistic Comparisons 190
10.7 Fuzzy Logic Application 191
10.7.1 How Fuzzy Logic is Used 191
10.8 The Rule Matrix 192
10.8.1 Fuzzy Logic Implementation 193
10.8.2 Membership Functions 194
10.8.3 Input Degree of Membership 197
10.8.4 Inferencing 197
10.9 Defuzzification 198
10.9.1 Fuzzy Centroid Algorithm 198
10.10 Tuning and System Enhancement 199

**CHAPTER 11 8-BIT MICROCONTROLLERS 201**

**11.0 General-Purpose Microcontrollers 201**
11.1 MicroChip PIC18F4520 202
11.1.1 PIC18F4520 Harvard Architecture 202
11.1.2 Instruction Pipeline 204
11.1.3 Special Features 205
11.1.4 Power Management Modes 205
11.1.5 Oscillator Configuration 206

11.1.6 Reset 207
11.1.7 Memory Organization 208
11.1.8 Interrupt Structure 210
11.1.9 Input/Output Ports 211
11.1.10 Timer-Related Functions 211
11.1.11 Timer Modules 212
11.1.12 Capture/Compare/PWM Functions 215
11.1.13 Serial Communication Interface 218
11.1.13.1 MSSP 218
11.1.13.2 SPI 218
11.1.13.3 $I^2C$ 219
11.1.13.4 EUSART 220
11.1.14 Analog-to-Digital Converter 222
11.1.15 Analog Comparator 223
11.1.16 Special Features of the CPU 224
11.1.17 Instruction Set 225
11.1.18 Electrical Characteristics 225
11.2 ZiLOG Z8 Encore! XP F0830 Series 226
11.2.1 eZ8 CPU Description 227
11.2.2 The Z8 Encore! CPU Architecture 228
11.2.2.1 Fetch Unit 228
11.2.2.2 Execution Unit 228
11.2.3 Address Space 229
11.2.3.1 Register File 229
11.2.3.2 Program Memory 230
11.2.3.3 Data Memory 230
11.2.4 Peripherals Overview 231
11.2.5 Reset Controller and Stop Mode Recovery 233
11.2.6 Low-Power Modes 233
11.2.7 General-Purpose Input/Output 234
11.2.7.1 GPIO Architecture 234
11.2.7.2 GPIO Alternate Functions 235
11.2.7.3 GPIO Interrupts 235
11.2.8 Interrupt Controller 235
11.2.8.1 Master Interrupt Enable 236
11.2.8.2 Interrupt Vectors and Priority 236
11.2.9 Timers 237
11.2.9.1 ONE-SHOT Mode 237
11.2.9.2 CONTINUOUS Mode 238
11.2.9.3 COMPARATOR COUNTER Mode 238
11.2.9.4 PWM SINGLE OUTPUT Mode 238
11.2.9.5 PWM DUAL OUTPUT Mode 238
11.2.9.6 CAPTURE Mode 239
11.2.9.7 CAPTURE RESTART Mode 239
11.2.9.8 COMPARE Mode 239
11.2.9.9 GATED Mode 240
11.2.9.10 CAPTURE/COMPARE Mode 240
11.2.10 Watchdog Timer 240
11.2.11 Analog-to-Digital Converter 241
11.2.11.1 ADC Operation 242
11.2.11.2 ADC Timing 242
11.2.12 Comparator 243
11.2.13 Flash Memory 243
11.2.14 Nonvolatile Data Storage 243
11.2.15 On-Chip Debugger 244

11.2.16 Oscillator Control 245
11.2.16.1 Crystal Oscillator 245
11.2.16.2 Internal Precision Oscillator 246
11.2.17 eZ8 CPU Instructions and Programming 247
11.2.17.1 Program Stack 247

**CHAPTER 12 16-BIT MICROCONTROLLER 250**

**12.0 16-bit Processor Overview 250**
12.1 Freescale S12XD Processor Overview 250
12.1.1 XGATE Overview 253
12.1.1.1 XGATE Module 254
12.1.1.2 XGATE RISC Core 255
12.1.1.3 XGATE Programmer's Model 255
12.1.1.4 XGATE Memory Map 256
12.1.1.5 XGATE Semaphores 257
12.1.1.6 XGATE Modes of Operation 257
12.1.2 Clocking 257
12.1.2.1 Clock and Reset Generator (CRG) 258
12.1.2.2 Pierce Oscillator (XOSC) 258
12.1.3 Analog-to-Digital Convertor (ATD) 259
12.1.4 Enhanced Capture Timer (ECT) 261
12.1.4.1 Features 261
12.1.5 Pulse-Width Modulator (PWM) 262
12.1.5.1 Features 263
12.1.6 Interintegrated Circuit (IIC) 263
12.1.6.1 Features 263
12.1.7 Scalable Controller Area Network (CAN) 264
12.1.7.1 Features 264
12.1.7.2 CAN System 265
12.1.8 Serial Communication Interface (SCI) 265
12.1.8.1 Features 265
12.1.8.2 Functional Description 266
12.1.8.3 Data Formats 268
12.1.8.4 Receiver 268
12.1.8.5 Transmitter 268
12.1.8.6 Baud Rate Generator 268
12.1.9 Serial Peripheral Interface (SPI) 269
12.1.9.1 Features 269
12.1.9.2 Functional Description 271
12.1.10 Periodic Interrupt Timer (PIT) 272
12.1.10.1 Features 273
12.1.11 Voltage Regulator (VREG) 273
12.1.11.1 Features 274
12.1.12 Background Debug Module (BDM) 274
12.1.12.1 Features 274
12.1.13 Interrupt Module (XINT) 275
12.1.13.1 Features 275
12.1.13.2 Interrupt Nesting 276
12.1.14 Mapping Memory Control (MMC) 277
12.1.14.1 Features 277
12.1.15 Debug (DBG) 278
12.1.15.1 Features 278
12.1.16 External Bus Interface (XEBI) 280
12.1.16.1 Features 280

12.1.17 Port Integration Module (PIM) 280
12.1.17.1 Features 282
12.1.17.2 Port Pin 282
12.1.17.3 Functional Description 282
12.1.17.4 Data Register 282
12.1.17.5 Input Register 283
12.1.17.6 Data Direction Register 283
12.1.18 2 Kbyte EEPROM (EETX2K) 284
12.1.18.1 Features 284
12.1.18.2 Functional Description 285
12.1.18.3 EEPROM Module Security 286
12.1.19 512 Kbyte Flash Module (FTX512K4) 286
12.1.19.1 Features 286
12.1.20 Security (SEC) 286
12.1.20.1 Features 286
12.1.20.2 Modes of Operation 288
12.1.20.3 Secured Microcontroller 288
12.2 Texas Instruments MSP430™ Family 288
12.2.1 Low Power Design 291
12.2.2 Flexible Clock System 291
12.2.3 MSP430 CPU 292
12.2.4 Operating Modes 293
12.2.5 FLL+ Clock Module 293
12.2.6 Flash Memory Controller 295
12.2.7 Hardware Multiplier 295
12.2.8 DMA Controller 296
12.2.9 Digital I/O 297
12.2.10 Watchdog Timer 297
12.2.11 Timers A and B 298
12.2.12 USART 299
12.2.13 USCI 301
12.2.13.1 UART Mode 301
12.2.13.2 SPI Mode 301
12.2.13.3 $I^2C$ Mode 303
12.2.14 ADC12 Function 304
12.2.15 DAC12 306
12.2.16 Embedded Emulation Module 306
12.2.16.1 Triggers 307

**CHAPTER 13 INTELLECTUAL PROPERTY SoC CORES 309**

**13.0 SoC Overview 309**
13.1 SoC Design Challenges 310
13.1.1 Configurable Processors 312
13.1.2 SoC Integration 314
13.1.3 Extensible Processors 316
13.1.4 Extensible Processors as RTL Alternatives 316
13.1.5 Explicit Control Scheme 317
13.2 The MIPS32 4K Processor Core Family 318
13.2.1 Key Features of the 4KE Family 319
13.2.2 Execution Unit 322
13.2.3 Multiply/Divide Unit (MDU) 323
13.2.4 Memory Manage Unit (MMU) 324
13.2.5 Cache Controller 325
13.2.6 Bus Interface Unit (BIU) 325
13.2.7 Power Management 326