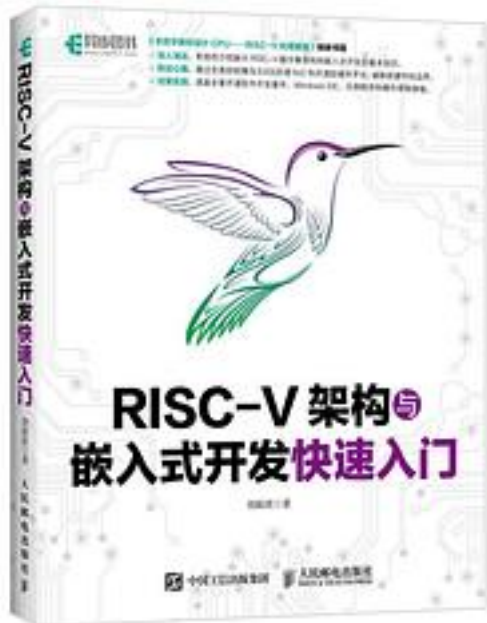


RISC-V架构与嵌入式开发快速入门



[RISC-V架构与嵌入式开发快速入门 下载链接1](#)

著者:胡振波

出版者:人民邮电出版社

出版时间:2019-1

装帧:平装-胶订

isbn:9787115494139

本书是一本介绍RISC-V架构嵌入式开发的入门书籍，以通俗的语言系统介绍了嵌入式开发的基础知识和RISC-V架构的内容，力求帮助读者快速掌握RISC-V架构的嵌入式开发技术。
本书共分为两部分。*部分为第1~14章，基本涵盖了使用RISC-V架构进行嵌入式开发所需的所有关键知识。第二部分为附录部分，详细介绍了RISC-V指令集架构，辅以作者加入的背景知识解读和注解，以便于读者理解。

作者介绍:

胡振波，本科毕业于上海交通大学电子工程系，硕士毕业于上海交通大学微电子学院。拥有多年ASIC和CPU设计与验证经验，曾在Marvell公司任ARM架构CPU设计高级工程师，在Synopsys公司任研发经理，在比特大陆公司任IC设计总监，在武汉聚芯微电子公司任架构师，现致力于RISC-V架构在国内的传播和发展。他开发并开源了蜂鸟E203超低功耗RISC-V处理器内核。

目录: 第1章 进入32位时代，谁能成为下一个80511

1.1 磨刀不误砍柴工—CPU基础知识介绍1

1.1.1 ISA—CPU的灵魂2

1.1.2 CISC与RISC3

1.1.3 32位与64位架构4

1.1.4 ISA众生相5

1.1.5 CPU的领域之分9

1.1.6 8位时代的传奇“前辈”—805110

1.1.7 IoT的崛起—32位时代的到来11

1.2 无敌是多么寂寞—ARM统治着的世界11

1.2.1 独乐乐与众乐乐—ARM公司的盈利模式12

1.2.2 小个子有大力量—无处不在的Cortex-M系列14

1.2.3 移动王者—Cortex-A系列在手持设备领域的巨大成功16

1.2.4 进击的巨人—ARM进军PC与服务器领域的雄心18

1.2.5 游戏终结者之ARM19

1.3 东边日出西边雨，道是无晴却有晴—RISC-V登场19

1.4 RISC-V和其他开放架构有何不同21

1.4.1 “平民英雄”—OpenRISC22

1.4.2 “豪门显贵”—SPARC22

1.4.3 “名校优生”—RISC-V23

1.5 结语：进入32位时代，谁能成为深嵌入式领域的下一个8051? 23

第2章 开源蜂鸟E203超低功耗RISC-V Core与SoC25

2.1 乱花渐欲迷人眼25

2.2 与众不同的蜂鸟E203处理器25

2.3 蜂鸟虽小，五脏俱全—蜂鸟E203简介26

2.4 蜂鸟E203性能指标27

2.5 蜂鸟E203配套SoC28

第3章 大道至简—RISC-V架构之魂29

3.1 简单就是美—RISC-V架构的设计哲学29

3.1.1 无病一身轻—架构的篇幅30

3.1.2 能屈能伸—模块化的指令集31

3.1.3 浓缩的都是精华—指令的数量31

3.2 RISC-V指令集架构简介32

3.2.1 模块化的指令子集32

3.2.2 可配置的通用寄存器组33

3.2.3 规整的指令编码33

3.2.4 简洁的存储器访问指令34

3.2.5 高效的分支跳转指令35

3.2.6 简洁的子程序调用36

3.2.7 无条件码执行36

3.2.8 无分支延迟槽37

3.2.9 零开销硬件循环37

3.2.10 简洁的运算指令38

3.2.11 优雅的压缩指令子集38

3.2.12 特权模式39

3.2.13 CSR寄存器40

3.2.14 中断和异常40

3.2.15 矢量指令子集	40
3.2.16 自定义指令扩展	40
3.2.17 总结与比较	41
第4章 RISC-V架构的中断和异常	43
4.1 中断和异常概述	43
4.1.1 中断概述	43
4.1.2 异常概述	44
4.1.3 广义上的异常	44
4.2 RISC-V架构异常处理机制	46
4.2.1 进入异常	47
4.2.2 退出异常	50
4.2.3 异常服务程序	50
4.3 RISC-V架构中断定义	51
4.3.1 中断类型	51
4.3.2 中断屏蔽	54
4.3.3 中断等待	55
4.3.4 中断优先级与仲裁	55
4.3.5 中断嵌套	56
4.3.6 总结比较	57
4.4 RISC-V架构异常相关CSR寄存器	57
4.5 蜂鸟E203的中断和异常实现	58
第5章 开源蜂鸟E203 MCU SoC总体介绍	59
5.1 Freedom E310 SoC简介	59
5.2 蜂鸟E203 MCU SoC简介	60
5.3 蜂鸟E203 MCU SoC框图	60
5.4 蜂鸟E203 MCU SoC存储资源	61
5.4.1 片上存储资源	61
5.4.2 片外Flash存储资源	61
5.5 蜂鸟E203 MCU SoC外设资源	62
5.6 蜂鸟E203 MCU SoC地址分配	62
5.7 蜂鸟E203 MCU SoC时钟域划分	63
5.8 蜂鸟E203 MCU SoC电源域划分	64
5.9 蜂鸟E203 MCU SoC低功耗模式	64
5.10 蜂鸟E203 MCU SoC的全局复位	65
5.11 蜂鸟E203 MCU SoC的上电流程控制	66
5.12 蜂鸟E203 MCU SoC芯片引脚表	67
5.13 蜂鸟E203 MCU SoC的GPIO引脚分配	68
5.14 蜂鸟E203 MCU SoC的中断处理	70
5.14.1 蜂鸟E203处理器核的异常和中断处理	70
5.14.2 蜂鸟E203处理器的中断接口	71
5.14.3 CLINT模块生成计时器中断和软件中断	72
5.14.4 PLIC管理多个外部中断	73
第6章 开源蜂鸟E203 MCU SoC外设介绍	77
6.1 蜂鸟E203 MCU SoC外设总述	77
6.2 PLIC	78
6.3 CLINT	78
6.4 LCLKGEN	78
6.4.1 LCLKGEN简介	78
6.4.2 LCLKGEN寄存器列表	78
6.5 HCLKGEN	79
6.5.1 HCLKGEN简介	79
6.5.2 HCLKGEN寄存器列表	79
6.6 GPIO	79
6.6.1 GPIO特性	79

- 6.6.2 GPIO寄存器列表80
- 6.6.3 IO结构和IOF模式80
- 6.6.4 SoC各外设复用GPIO引脚83
- 6.6.5 GPIO中断83
- 6.6.6 GPIO_VALUE寄存器84
- 6.6.7 GPIO_INPUT_EN寄存器84
- 6.6.8 GPIO_OUTPUT_EN寄存器85
- 6.6.9 GPIO_PORT寄存器85
- 6.6.10 GPIO_PUE寄存器85
- 6.6.11 GPIO_DS寄存器85
- 6.6.12 GPIO_OUTPUT_XOR寄存器85
- 6.6.13 GPIO_RISE_IE、GPIO_RISE_IP等寄存器85
- 6.7 SPI86
 - 6.7.1 SPI背景知识简介86
 - 6.7.2 SPI特性88
 - 6.7.3 SPI寄存器列表89
 - 6.7.4 SPI接口数据线90
 - 6.7.5 通过SPI_SCKDIV寄存器配置SCK时钟频率90
 - 6.7.6 通过SPI_SCKMODE寄存器配置SCK的极性与相位90
 - 6.7.7 通过SPI_CSID寄存器配置SPI使能信号92
 - 6.7.8 通过SPI_CSDEF寄存器配置使能信号的空闲值92
 - 6.7.9 通过SPI_CSMODE寄存器配置使能信号的行为92
 - 6.7.10 通过SPI_DELAY0和SPI_DELAY1寄存器配置使能信号的行为93
 - 6.7.11 通过SPI_FCTRL寄存器使能QSPI0的Flash XiP模式94
 - 6.7.12 通过SPI_FFMT寄存器控制QSPI0读取外部Flash95
 - 6.7.13 通过SPI_FMT寄存器配置传输参数97
 - 6.7.14 通过SPI_TXDATA寄存器发送数据97
 - 6.7.15 通过SPI_RXDATA寄存器接收数据98
 - 6.7.16 通过SPI_TXMARK寄存器配置发送中断阈值100
 - 6.7.17 通过SPI_RXMARK寄存器配置接收中断阈值100
 - 6.7.18 通过SPI_IE和SPI_IP寄存器控制中断101
- 6.8 I2C102
 - 6.8.1 I2C背景知识简介102
 - 6.8.2 I2C特性103
 - 6.8.3 I2C寄存器列表103
 - 6.8.4 I2C接口数据线104
 - 6.8.5 通过I2C_PRERLo和I2C_PRERHi寄存器配置SCL时钟频率104
 - 6.8.6 通过I2C_CTR寄存器配置功能和中断使能104
 - 6.8.7 I2C模块产生中断105
 - 6.8.8 通过I2C_TXR和I2C_RXR寄存器发送和接收数据105
 - 6.8.9 通过I2C_CR和I2C_SR寄存器发起命令和查看状态106
 - 6.8.10 初始化I2C模块的序列107
 - 6.8.11 通过I2C模块向外部从设备写数据的常用序列107
 - 6.8.12 通过I2C模块从外部从设备读数据的常用序列108
- 6.9 UART109
 - 6.9.1 UART背景知识简介109
 - 6.9.2 UART特性110
 - 6.9.3 UART寄存器列表110
 - 6.9.4 UART接口数据线111
 - 6.9.5 通过UART_TXDATA寄存器发送数据111
 - 6.9.6 通过UART_RXDATA寄存器接收数据112
 - 6.9.7 通过UART_TXCTRL寄存器进行发送控制113
 - 6.9.8 通过UART_RXCTRL寄存器进行接收控制113
 - 6.9.9 通过UART_IE和UART_IP寄存器控制中断114

- 6.9.10 通过UART_DIV寄存器配置波特率115
- 6.10 PWM116
 - 6.10.1 PWM背景知识简介116
 - 6.10.2 PWM特性和结构图116
 - 6.10.3 PWM寄存器列表117
 - 6.10.4 通过PWMCFG寄存器进行配置118
 - 6.10.5 计数器计数值PWMCOUNT寄存器和PWM周期119
 - 6.10.6 计数器比较值PWMS寄存器120
 - 6.10.7 PWM接口数据线121
 - 6.10.8 产生左对齐或者右对齐的脉冲信号121
 - 6.10.9 产生居中对齐的脉冲信号122
 - 6.10.10 配置pwmcmpgang结连产生任意形状的脉冲信号124
 - 6.10.11 配置pwmdeglitch防止输出毛刺124
 - 6.10.12 PWM产生中断124
- 6.11 WDT125
 - 6.11.1 WDT背景知识简介125
 - 6.11.2 WDT特性和结构图125
 - 6.11.3 WDT寄存器列表126
 - 6.11.4 通过WDOGCFG寄存器进行配置127
 - 6.11.5 计数器计数值WDOGCOUNT寄存器128
 - 6.11.6 通过WDOGKEY寄存器解锁128
 - 6.11.7 通过WDOGFEED寄存器喂狗129
 - 6.11.8 计数器比较值WDOGS寄存器129
 - 6.11.9 通过WDOGCMP寄存器配置阈值130
 - 6.11.10 WDT产生全局复位130
 - 6.11.11 WDT产生中断130
- 6.12 RTC131
 - 6.12.1 RTC背景知识简介131
 - 6.12.2 RTC特性和结构图131
 - 6.12.3 RTC寄存器列表131
 - 6.12.4 通过RTCCFG寄存器进行配置132
 - 6.12.5 计数器计数值RTCHI RTCLO寄存器133
 - 6.12.6 计数器比较值RTCS寄存器133
 - 6.12.7 通过RTCCMP寄存器配置阈值134
 - 6.12.8 RTC产生中断134
- 6.13 PMU134
 - 6.13.1 PMU背景知识简介134
 - 6.13.2 PMU特性和结构图135
 - 6.13.3 PMU寄存器列表136
 - 6.13.4 通过PMUKEY寄存器解锁136
 - 6.13.5 通过PMUSLEEP寄存器进入休眠模式137
 - 6.13.6 通过PMUSLEEP寄存器配置休眠指令序列137
 - 6.13.7 通过PMUBACKUP寄存器保存关键信息139
 - 6.13.8 通过PMUIE寄存器配置唤醒条件139
 - 6.13.9 通过PMUWAKEUPI寄存器配置唤醒指令序列140
 - 6.13.10 通过PMUCAUSE寄存器查看唤醒原因141
- 第7章 开源蜂鸟E203 MCU开发板与调试器143
 - 7.1 蜂鸟E203 MCU开发板143
 - 7.2 蜂鸟E203 JTAG调试器143
- 第8章 编译过程简介145
 - 8.1 GCC工具链介绍145
 - 8.1.1 GCC工具链概述145
 - 8.1.2 Binutils146
 - 8.1.3 C运行库147

8.1.4 GCC命令行选项	148
8.2 准备工作	148
8.2.1 Linux安装	148
8.2.2 准备Hello World程序	148
8.3 编译过程	149
8.3.1 预处理	149
8.3.2 编译	150
8.3.3 汇编	150
8.3.4 链接	151
8.3.5 一步到位的编译	153
8.4 分析ELF文件	153
8.4.1 ELF文件介绍	153
8.4.2 ELF文件的段	154
8.4.3 查看ELF文件	154
8.4.4 反汇编	155
8.5 嵌入式系统编译的特殊性	156
8.6 本章小结	156
第9章 嵌入式开发特点与RISC-V GCC工具链	158
9.1 嵌入式系统开发特点	158
9.1.1 交叉编译和远程调试	158
9.1.2 移植newlib或newlib-nano作为C运行库	159
9.1.3 嵌入式引导程序和中断异常处理	160
9.1.4 嵌入式系统链接脚本	160
9.1.5 减少代码体积	161
9.1.6 支持printf函数	161
9.1.7 提供板级支持包	162
9.2 RISC-V GCC工具链简介	162
9.2.1 RISC-V GCC工具链种类	162
9.2.2 riscv-none-embed工具链下载	163
9.2.3 RISC-V GCC工具链的 (?Cmarch=) 和 (?Cmabi=) 选项	164
9.2.4 RISC-V GCC工具链的 (?Cmcmode=) 选项	168
9.2.5 RISC-V GCC工具链的其他选项	169
9.2.6 RISC-V GCC工具链的预定义宏	170
9.2.7 RISC-V GCC工具链使用实例	170
第 10章 RISC-V汇编语言程序设计	171
10.1 汇编语言简介	171
10.2 RISC-V汇编程序概述	172
10.3 RISC-V汇编指令	173
10.4 RISC-V汇编程序伪操作	173
10.5 RISC-V汇编程序示例	177
10.5.1 定义标签	177
10.5.2 定义宏	178
10.5.3 定义常数	178
10.5.4 立即数赋值	178
10.5.5 标签地址赋值	179
10.5.6 设置浮点舍入模式	179
10.5.7 完整实例	180
10.6 在C C 程序中嵌入汇编	181
10.6.1 GCC内联汇编简述	181
10.6.2 GCC内联汇编 “输出操作数” 和 “输入操作数” 部分	182
10.6.3 GCC内联汇编 “可能影响的寄存器或存储器” 部分	183
10.6.4 GCC内联汇编参考实例一	184
10.6.5 GCC内联汇编参考实例二	185
10.6.6 小结	186

10.7 在汇编中调用C C 函数	186
10.8 本章小结	187
第 11章 基于HBird-E-SDK平台的软件开发与运行	188
11.1 HBird-E-SDK平台简介	188
11.2 HBird-E-SDK平台代码结构	189
11.3 HBird-E-SDK板级支持包解析	190
11.3.1 移植了Newlib桩函数	190
11.3.2 支持了printf函数	192
11.3.3 提供系统链接脚本	193
11.3.4 系统启动引导程序	198
11.3.5 系统异常和中断处理	202
11.3.6 减少代码体积	206
11.4 使用HBird-E-SDK开发和编译程序	208
11.4.1 在HBird-E-SDK环境中安装工具链	208
11.4.2 在HBird-E-SDK环境中开发程序	210
11.4.3 编译使得程序从Flash直接运行	211
11.4.4 编译使得程序从ITCM中运行	212
11.4.5 编译使得程序从Flash上载至ITCM中运行	213
11.5 使用HBird-E-SDK下载程序	213
11.5.1 JTAG调试器与MCU原型开发板的连接	213
11.5.2 设置JTAG调试器在Linux系统中的USB权限	214
11.5.3 将程序下载至MCU原型开发板	216
11.6 在MCU原型开发板上运行程序	216
11.6.1 程序从Flash直接运行	217
11.6.2 程序从ITCM中运行	217
11.6.3 程序从Flash上载至ITCM中运行	218
11.7 使用GDB远程调试程序	218
11.7.1 调试器工作原理	218
11.7.2 GDB常用操作示例	220
11.7.3 使用GDB调试Hello World示例	221
第 12章 开源蜂鸟E203 MCU的更多示例程序	226
12.1 Dhrystone示例程序	226
12.1.1 Dhrystone示例程序功能简介	226
12.1.2 Dhrystone示例程序代码结构	229
12.1.3 运行Dhrystone	229
12.2 CoreMark示例程序	231
12.2.1 CoreMark示例程序功能简介	231
12.2.2 CoreMark示例程序代码结构	232
12.2.3 运行CoreMark	233
12.3 Demo_IASM示例程序	234
12.3.1 Demo_IASM示例程序功能简介	234
12.3.2 Demo_IASM示例程序代码结构	234
12.3.3 Demo_IASM示例程序源码解析	235
12.3.4 运行Demo_IASM	235
12.4 Demo_GPIO示例程序	236
12.4.1 Demo_GPIO示例程序功能简介	236
12.4.2 Demo_GPIO示例程序代码结构	237
12.4.3 Demo_GPIO示例程序源码分析	237
12.4.4 运行Demo_GPIO	243
12.5 中断嵌套	244
第 13章 Windows IDE集成开发调试环境	245
13.1 MCU Eclipse IDE简介与安装	245
13.1.1 MCU Eclipse IDE简介	245
13.1.2 RISC-V MCU Eclipse下载	246

- 13.1.3 RISC-V MCU Eclipse安装247
- 13.2 启动Eclipse248
- 13.3 创建Hello World项目249
- 13.4 配置Hello World项目253
 - 13.4.1 配置工具链路径253
 - 13.4.2 配置项目的编译和链接选项256
 - 13.4.3 配置项目的BSP262
 - 13.4.4 配置项目的包含路径和文件265
- 13.5 编译Hello World项目267
- 13.6 运行Hello World项目269
 - 13.6.1 安装JTAG调试器在Windows系统中的USB驱动269
 - 13.6.2 通过Eclipse下载程序至MCU开发板270
 - 13.6.3 在MCU开发板上运行程序274
- 13.7 调试Hello World项目276
- 13.8 拓展一：基于MCU Eclipse运行调试demo_gpio示例280
- 13.9 拓展二：基于MCU Eclipse运行调试dhrystone示例281
- 第 14章 开源蜂鸟E203 MCU开发板移植RTOS283
 - 14.1 RTOS简述283
 - 14.2 常用实时操作系统概述284
 - 14.3 FreeRTOS简介285
 - 14.4 蜂鸟E203 MCU移植RTOS286
- 附录A RISC-V架构指令集介绍287
 - A.1 RV32GC架构概述287
 - A.2 RV32E架构概述288
 - A.3 蜂鸟E203支持的指令列表288
 - A.4 寄存器组288
 - A.4.1 通用寄存器组289
 - A.4.2 CSR寄存器290
 - A.5 指令PC290
 - A.6 寻址空间划分290
 - A.7 大端格式或小端格式290
 - A.8 工作模式290
 - A.9 Hart概念291
 - A.10 复位状态291
 - A.11 中断和异常292
 - A.12 存储器地址管理292
 - A.13 存储器模型292
 - A.14 指令类型293
 - A.14.1 RV32IMAFDC指令列表293
 - A.14.2 基本整数指令（RV32I） 293
 - A.14.3 整数乘法和除法指令（RV32M指令子集） 302
 - A.14.4 浮点指令（RV32F，RV32D指令子集） 304
 - A.14.5 存储器原子操作指令（RV32A指令子集） 315
 - A.14.6 16位压缩指令（RV32C指令子集） 318
 - A.15 伪指令320
 - A.16 指令编码320
- 附录B RISC-V架构CSR寄存器介绍321
 - B.1 蜂鸟E203支持的CSR寄存器列表321
 - B.2 RISC-V标准CSR322
 - B.2.1 misa322
 - B.2.2 mvendorid323
 - B.2.3 marchid323
 - B.2.4 mimpid323
 - B.2.5 mhartid324

B.2.6	fflags	324
B.2.7	frm	324
B.2.8	fcsr	324
B.2.9	mstatus	324
B.2.10	mtvec	326
B.2.11	mepc	326
B.2.12	mcause	327
B.2.13	mtval (mbadaddr)	327
B.2.14	mie	328
B.2.15	mip	328
B.2.16	mscratch	328
B.2.17	mcycle和mcycleh	329
B.2.18	minstret和minstreth	329
B.2.19	mtime、mtimecmp和msip	329
B.3	蜂鸟E203自定义CSR	330
附录C	RISC-V架构的PLIC介绍	331
C.1	概述	331
C.2	PLIC中断目标	332
C.3	PLIC中断源	333
C.3.1	PLIC中断源之闸口 (Gateway) 和IP	334
C.3.2	PLIC中断源之编号 (ID)	334
C.3.3	PLIC中断源之优先级 (Priority)	334
C.3.4	PLIC中断源之中断使能 (Enable)	334
C.4	PLIC中断处理机制	335
C.4.1	PLIC中断通知机制 (Notification)	335
C.4.2	PLIC中断响应机制 (Claim)	335
C.4.3	PLIC中断完成机制 (Completion)	336
C.4.4	PLIC中断完整流程	336
C.5	PLIC寄存器总结	337
C.6	总结与比较	339
附录D	存储器模型背景介绍	340
D.1	为何要有存储器模型的概念	340
D.2	存储器模型定义了什么	341
D.2.1	按序一致性模型	341
D.2.2	松散一致性模型	342
D.2.3	释放一致性模型	342
D.2.4	存储器模型总结	342
D.3	存储器模型应用实例	343
D.4	RISC-V架构的存储器模型	344
附录E	存储器原子操作指令背景介绍	345
E.1	什么是“上锁”问题	345
E.2	通过原子操作解决“上锁”问题	346
E.3	通过互斥操作解决“上锁”问题	346
E.4	RISC-V架构的相关指令	348
附录F	RISC-V指令编码列表	349
F.1	RV32I指令编码	349
F.2	RV32M指令编码	350
F.3	RV32A指令编码	350
F.4	RV32F指令编码	350
F.5	RV32D指令编码	351
F.6	RVC指令编码	351
附录G	RISC-V伪指令列表	353
• • • • •	(收起)	

标签

CPU

计算机

RISC-V

嵌入式

YTYBBY

#JX

#

评论

广告书，精简不见得是优势

可以用来做基础科普

工科生写一本书着实不易，这也是国内riscv嵌入式的首本教材。前面介绍性的内容可以看看了解一下，外设和异常部分翻译地不错，后面就看不下去了。

特权架构只讲了机器模式，没有其它模式。也没有这些模式之间转换的知识点。其它还行。

内容还行。排版槽点太多，有些图片直接高斯模糊了，还有一堆黑底白字的代码图片，墨水全用来印这几个图了

[RISC-V架构与嵌入式开发快速入门_下载链接1](#)

书评

[RISC-V架构与嵌入式开发快速入门_下载链接1](#)