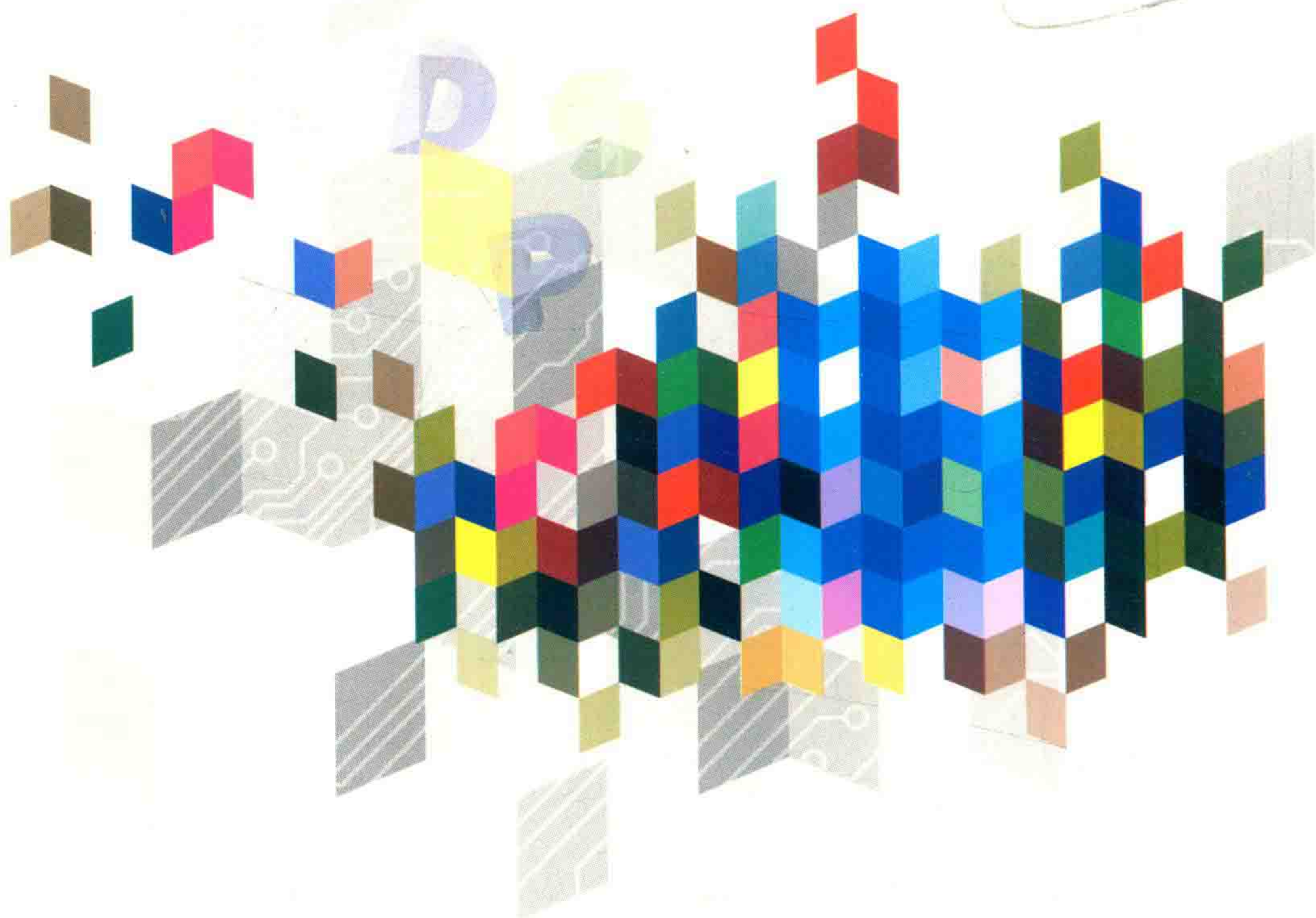


TMS320C6748 DSP

原理与实践

王 斌 熊谷辉 曹琳峰 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

TMS320C6748 DSP

原理与实践

王 斌 熊谷辉 曹琳峰 编著

北京航空航天大学出版社

内 容 简 介

本书以 TMS320C6748 DSP 处理器的使用为主线,结合大量实例分析,由浅入深,从基础的入门操作到复杂的外设操作进行了详细介绍。本书在内容结构上分为快速入门篇、硬件概述篇、C674x 详解篇、外设开发篇和程序优化入门 5 个篇,内容涵盖 DSP 开发流程、中断及缓存的使用、常用及特色外设接口开发详解等。

本书配套资料包含开发工具、开发例程、开发视频、TMS320C6748 中文手册等资料,读者可在广州创龙提供的技术论坛 www.51ele.net 或 www.tronlong.com 免费获取。

本书可作为 DSP 开发应用的参考教材,也可作为其他层次 DSP 开发应用人员的参考手册。

图书在版编目(CIP)数据

TMS320C6748 DSP 原理与实践 / 王斌,熊谷辉,曹琳峰编著. -- 北京:北京航空航天大学出版社,2018.9

ISBN 978-7-5124-2801-0

I. ①T… II. ①王… ②熊… ③曹… III. ①数字信号处理 IV. ①TN911.72

中国版本图书馆 CIP 数据核字(2018)第 166238 号

版权所有,侵权必究。

TMS320C6748 DSP 原理与实践

王 斌 熊谷辉 曹琳峰 编著

责任编辑 董立娟

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编:100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@buaacm.com.cn 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:27 字数:575 千字

2018 年 9 月第 1 版 2018 年 9 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-2801-0 定价:79.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

序

近三十多年来,DSP 技术在中国得到迅速的应用,特别是 TI C2000、C5000、C6000 架构的 DSP 处理器在高校教学、工业控制等领域应用的普及。随着信息技术时代的高速发展,工业领域对高速数据处理的需求越来越大。TI C674x、C66x 等新框架 DSP 处理器的出现,满足了当前工业高速数据处理产品的开发需求,但是,其技术解决方案和开发资料严重匮乏。

本书作者是长期从事 DSP 开发的工程师,积累了许多实际项目的开发经验。为了让更多初学者能快速入门并掌握 DSP 开发技术,于是依据多年的项目设计经验写出了本书。

全书以“如何点亮一个 LED”开篇,从 DSP 工程的建立、编写、调试等方面讲解了集成开发环境的使用,让初学者快速了解 DSP 开发流程;随后,介绍了 TMS320C6748 的硬件设计思路,并详细讲解了 TMS320C6748 外设接口的驱动原理;最后,根据实际应用需要,阐述了 DSP 程序优化的基础知识。

作者从一个初学者的角度,立足于具体实际项目开发需求,讲授了入门学习的开发方法和步骤,对初学者快速了解并掌握 TMS320C6748 的基础开发内容有很大帮助。如果您想入门学习 DSP 技术开发,这本书将是一个很好的选择。

广州创龙研发部经理 梁权荣

2018 年 7 月于广州

前言

随着嵌入式技术的发展,需要用到 C6000 DSP 的开发人员越来越多,但是一本称手的入门资料却很少,于是本书应运而生。

起初规划大纲的时候,计划将很多相关内容编写进去。例如,DSP 开发入门、主要外设详细使用指南、RTSC 实时软件组件及 SYS/BIOS 实时操作系统、ARM 与 DSP 双核开发、ARM、DSP 及 FPGA 三核开发以及 DSP 程序优化等内容。但是,若把这么多内容全部编进去,则这本书的体量就会过于庞大而且不利于初学者使用。于是,决定将内容规划为系列丛书,第一本书,也就是本书以 C6748 为重点来编写,主要设计开发入门、CCS 集成开发环境使用、外设使用、中断及缓存以及程序优化入门等内容。

作者在编写过程中以初学者的视角来编写,方便读者更好、更快地入门。本书在内容结构上分为几个部分:

快速入门篇——指导读者熟悉 DSP 开发流程、生成 DSP 程序需要用到的文件和工具以及使用方法。通过这部分内容,读者可以初步完成 DSP 的简单开发操作。

硬件概述篇——简单介绍了两款 DSP 处理器硬件设计方法及思路。

C674x 详解篇——主要介绍中断及缓存的使用。

外设开发篇——详细介绍了常用以及比较有特色的外设,比如 GPIO、UART、EDMA3、EMIFA、uPP 以及 PRU。

程序优化入门篇——介绍 DSP 程序优化常用方式,并介绍了 C++、线性汇编以及汇编开发 DSP 程序的方法。

本书的集成开发环境是最新的 CCSv7。

本书第 1~3 章、第 5~6 章、第 14~15 章以及附录由王斌编写,第 8~14 章的外设相关章节由熊谷辉编写,第 4 章的硬件部分由曹琳峰编写。

本书能够出版得到很多朋友的帮助。特别感谢哈尔滨工程大学刘淞佐教授细致

严谨的校审工作,也特别感谢陈汇照先生的协调工作。黄继豪先生及朱雅先生对本书的出版也提供了不少帮助,这里一并表示感谢。

由于作者水平有限,本书难免有不足之处,欢迎读者发送邮件进行讨论,邮箱: support@tronlong.com。

作者

广州创龙电子科技有限公司

2018年8月于广州

本书配套资料包含开发工具、开发例程、开发视频、TMS320C6748 中文手册等资料,读者可在广州创龙提供的技术论坛 www.51ele.net 或 www.tronlong.com 免费获取。读者也可以扫描公众号与作者交流互动:



目 录

快速入门篇

第 1 章 如何点亮一个 LED	2
1.1 总体流程	2
1.2 安装及配置 CGT 工具	11
1.2.1 安 装	12
1.2.2 系统环境变量配置	13
1.3 编写源文件	16
1.4 编写 CMD 文件	18
1.4.1 基本原理	18
1.4.2 C6748 CMD 文件	21
1.5 编 译	24
1.6 生成 AIS 文件	28
1.6.1 AIS 文件	28
1.6.2 使用 AISgen 生成 AIS 文件	30
1.6.3 使用 HexAIS 生成 AIS 文件	42
1.7 烧 写	46
第 2 章 Makefile 文件	50
2.1 概 述	50
2.2 配置环境变量	52
2.3 使用 Makefile 生成 DSP 应用程序	54
2.4 使用 Makefile 生成 DSP 静态库	59
第 3 章 CCS 集成开发环境	67
3.1 为什么要使用集成开发环境	67
3.1.1 创建普通 CCS 工程	68
3.1.2 工程导入/导出	77
3.1.3 更改编译选项	81
3.1.4 更改链接选项	85

目 录

3.1.5	生成可执行文件.....	86
3.1.6	创建 CCS 静态库工程	88
3.1.7	工作集	89
3.2	CCS 调试	92
3.2.1	建立仿真器与开发板连接.....	93
3.2.2	GEL 文件	101
3.2.3	调试工程	104
3.2.4	加载程序或符号	106
3.2.5	查看变量、表达式及寄存器值.....	107
3.2.6	查看、导入/导出内存数据	110
3.2.7	设置/配置断点.....	114
3.2.8	单步调试	117
3.2.9	软件仿真模式	117
3.2.10	图形分析.....	119
3.2.11	图像分析.....	121
3.2.12	串口终端.....	124
3.2.13	测量代码执行时间.....	126
3.3	版本管理	127
3.3.1	Git 概述	127
3.3.2	CCS Git 插件使用	127

硬件概述篇

第 4 章	硬件设计.....	146
4.1	核心板硬件设计	146
4.2	底板硬件设计	152

C674x 详解篇

第 5 章	中 断.....	186
5.1	C674x 中断控制器	186
5.2	中断向量表	188
5.3	事 件	189
5.4	配置中断	192
5.5	中断相关寄存器	198

第 6 章 缓 存..... 200

6.1 原 理 200

6.2 C674x 缓存架构 202

6.3 缓存配置 204

6.4 缓存一致性问题 208

外设开发篇

第 7 章 外设驱动库..... 211

7.1 StarterWare 概述 211

7.2 目录结构 212

7.3 上位机需求 215

7.4 USB 协议栈 216

7.5 网络协议栈 217

7.6 在 RTOS(SYS/BIOS)中集成 StarterWare 驱动库..... 219

7.6.1 迁移 StarterWare 程序到 SYS/BIOS 的注意事项 219

7.6.2 GPIO_KEY 例程从 StarterWare 到 SYS/BIOS 221

第 8 章 GPIO 227

8.1 GPIO 结构 227

8.2 引脚复用 228

8.3 GPIO 使用方法 231

8.4 实例分析 232

第 9 章 UART 238

9.1 概 述 238

9.2 串口结构 239

9.3 串口波特率产生 241

9.4 串口使用方法 242

9.5 串口 EDMA 事件支持 243

9.6 实例分析 243

第 10 章 EDMA3 250

10.1 概 述..... 250

10.2 EDMA3 结构 251

10.3 EDMA3 传输类型 254

10.3.1 EDMA3 传输数据块的定义 254

10.3.2 A 同步传输 255

10.3.3 AB 同步传输 256

目 录

10.3.4	A 同步传输与 AB 同步传输的区别	257
10.4	EDMA3 触发方式	258
10.5	参数 RAM(PaRAM)	260
10.5.1	PaRAM 的内容	260
10.5.2	PaRAM 的更新	267
10.5.3	连接传输	269
10.6	事件、通道与 PaRAM 之间的映射关系	272
10.7	事件队列	273
10.8	EDMA3 通道控制器区域与中断	273
10.8.1	EDMA3 通道控制器区域	273
10.8.2	EDMA3 中断	275
10.9	实例分析	277
第 11 章	EMIFA	288
11.1	EMIFA 结构	288
11.2	EMIFA 时钟控制	291
11.3	SDRAM 控制器	291
11.3.1	SDRAM 寻址	292
11.3.2	SDRAM 常见命令	292
11.3.3	SDRAM 读操作	294
11.3.4	SDRAM 写操作	294
11.3.5	逻辑地址与 EMIFA 引脚之间的映射	295
11.4	异步控制器	296
11.4.1	EMIFA 异步接口与异步设备的连接方式	296
11.4.2	EMIFA 异步接口读操作	297
11.4.3	EMIFA 异步接口写操作	299
11.5	EMIFA 地址映射	300
11.6	实例分析	301
第 12 章	uPP	311
12.1	uPP 结构	311
12.2	传输协议分析	312
12.3	实例分析	314
第 13 章	PRU	328
13.1	PRU 结构	328
13.2	PRU 内存映射	329
13.2.1	本地地址空间映射	329
13.2.2	全局地址空间映射	330

13.3	控制/状态寄存器	331
13.4	PRU 中断控制器	335
13.4.1	中断映射	336
13.4.2	PRUSS 系统事件	338
13.4.3	中断控制器配置	340
13.4.4	中断控制器寄存器	340
13.5	C 语言开发 PRU	343
13.5.1	新建 PRU 工程	343
13.5.2	使用仿真器加载并运行程序	351
13.5.3	使用 DSP 加载并运行程序	355

程序优化入门篇

第 14 章	程序优化	358
14.1	基本方法	358
14.2	优化算法函数	361
14.2.1	使用 Intrinsics 函数	361
14.2.2	使用针对 C6000 优化算法函数库	371
第 15 章	混合编程语言开发	377
15.1	使用 C++ 语言	377
15.2	使用线性汇编语言	386
15.3	使用汇编语言	395
15.4	C++/C/线性汇编/汇编混合编程	406
附录	CCS 安装及配置	408
参考文献		419

快速入门篇

本书侧重实践,所以第一部分先带领读者快速入门 DSP 的开发。第一章将以一种最简单的方式完成嵌入式开发的“Hello World”,即点亮一个 LED。后续的章节会对相关内容做出更加详细的诠释。



第 1 章 如何点亮一个 LED

第 2 章 Makefile 文件

第 3 章 CCS 集成开发环境

第 1 章

如何点亮一个 LED

1.1 总体流程

点亮发光二极管 LED(以下同)之前,需要确认 DSP 驱动 LED 的方式。核心板上由 GPIO6[12]及 GPIO6[13]直接驱动两个 LED(这样设计不合适,但 DSP 的 I/O 引脚驱动能力还是比较强,所以也可以这样使用),底板上使用 GPIO0[0]、GPIO0[1]、GPIO0[2]和 GPIO0[5]直接驱动。如若需要这几个 LED 亮或灭,则只需要在相应的 GPIO 端口输出高电平(输出 1)或输出低电平(输出 0)即可。

核心板 LED 硬件连接如图 1-1 所示。底板 LED 硬件连接如图 1-2 所示。

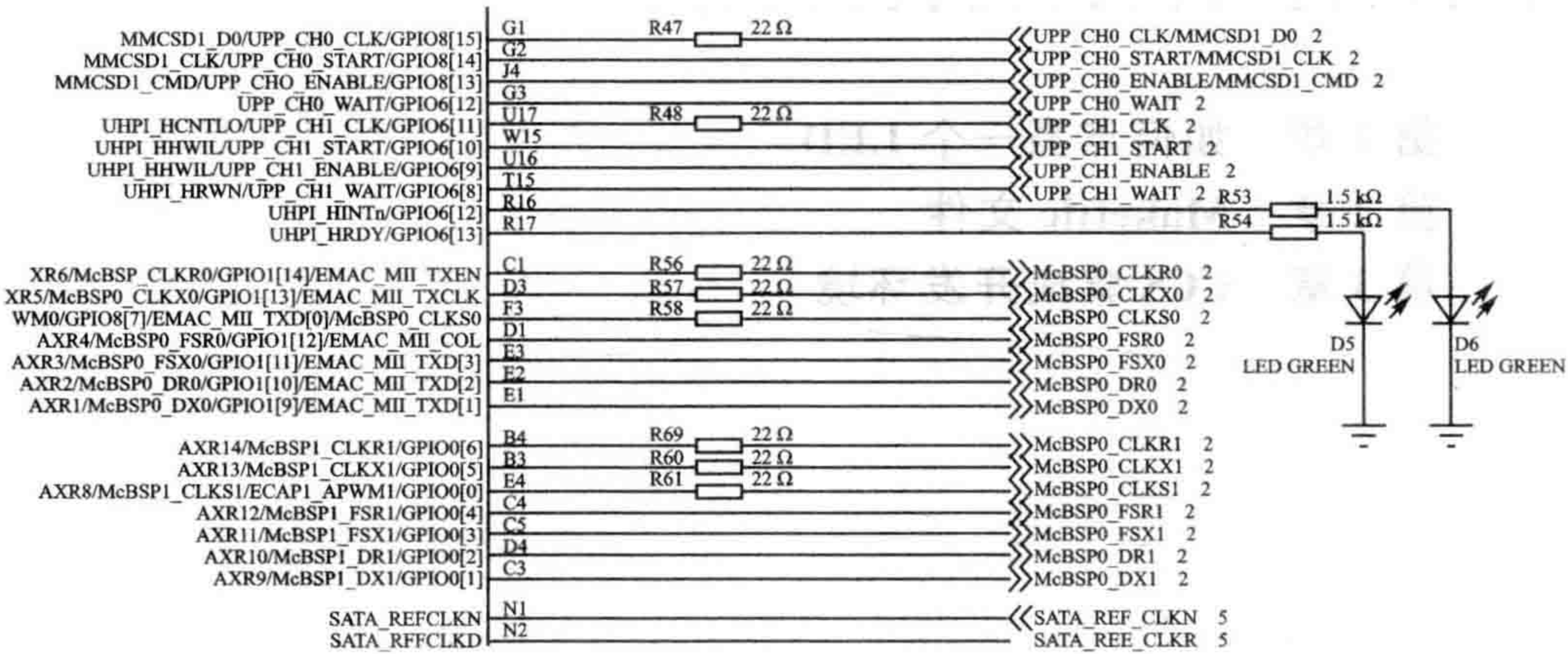


图 1-1 核心板 LED

如何在 TMS320C6748 的 GPIO 口输出高电平及低电平呢?首先要配置的就是引脚复用。一般来说,在外设比较多的嵌入式处理器中都会存在引脚复用。引脚复用可以使处理器在比较少的引脚上为更多的外设提供输入/输出引脚,但这存在引脚冲突,从而导致部分外设不能同时使用。通常,高速接口的输入/输出引脚是不会存在复用功能的,因为它们对速度和稳定性要求很高。C6748 处理器上一共有 361 个引脚,其中 144 个引脚是可以功能复用的。

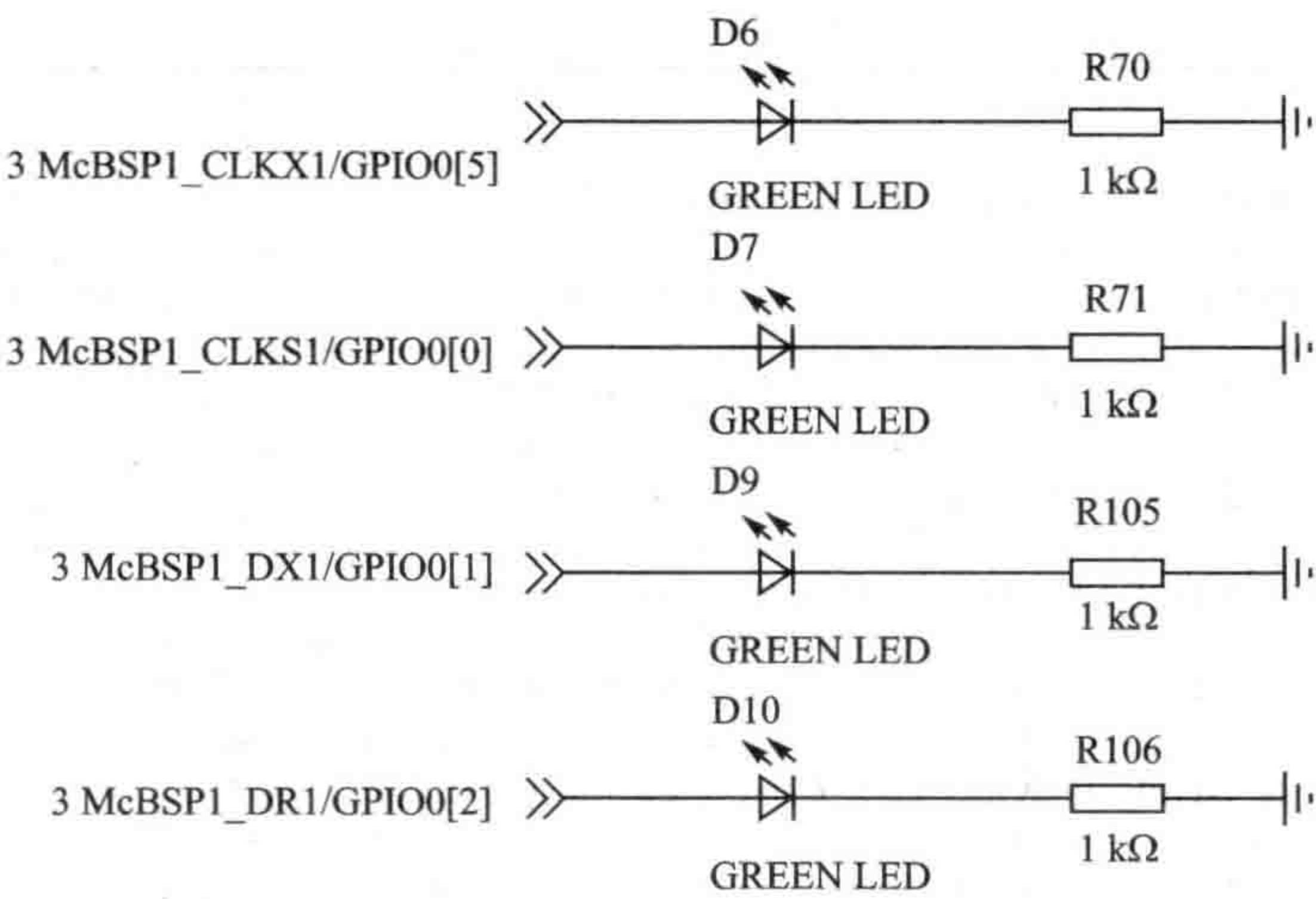


图 1-2 底板 LED

C6748 的 GPIO 部分引脚复用情况如表 1-1 所列。

表 1-1 引脚复用

信号名称	类 型	描 述
ACLKR / PRU0_R30[20] /GP0[15] / PRU0_R31[22]	I/O	GPIO Bank 0
ACLKX / PRU0_R30[19] /GP0[14] / PRU0_R31[21]	I/O	
AFSR /GP0[13] / PRU0_R31[20]	I/O	
AFSX /GP0[12] / PRU0_R31[19]	I/O	
AHCLKR / PRU0_R30[18] / UART1_RTS /GP0[11] /PRU0_R31[18]	I/O	
AHCLKX / USB_REFCLKIN / UART1_CTS /GP0[10] /PRU0_R31[17]	I/O	
AMUTE / PRU0_R30[16] / UART2_RTS /GP0[9] / PRU0_R31[16]	I/O	
RTC_ALARM / UART2_CTS /GP0[8] / DEEPSLEEP	I/O	
AXR15 / EPWM0TZ[0] / ECAP2_APWM2 /GP0[7]	I/O	
AXR14 / CLKR1 /GP0[6]	I/O	
AXR13 / CLKX1 /GP0[5]	I/O	
AXR12 / FSR1 /GP0[4]	I/O	
AXR11 / FSX1 /GP0[3]	I/O	
AXR10 / DR1 /GP0[2]	I/O	
AXR9 / DX1 /GP0[1]	I/O	
AXR8 / CLKS1 / ECAP1_APWM1 / GP0[0] / PRU0_R31[8]	I/O	

第1章 如何点亮一个 LED

续表 1-1

信号名称	类 型	描 述
RESETOUT / UHPI_HAS / PRU1_R30[14] / GP6[15]	I/O	GPIO Bank 6
CLKOUT / UHPI_HDS2 / PRU1_R30[13] / GP6[14]	I/O	
PRU0_R30[31] / UHPI_HRDY / PRU1_R30[12] / GP6[13]	I/O	
PRU0_R30[30] / UHPI_HINT / PRU1_R30[11] / GP6[12]	I/O	
PRU0_R30[29] / UHPI_HCNTL0 / UPP_CHA_CLOCK / GP6[11]	I/O	
PRU0_R30[28] / UHPI_HCNTL1 / UPP_CHA_START / GP6[10]	I/O	
PRU0_R30[27] / UHPI_HHWIL / UPP_CHA_ENABLE / GP6[9]	I/O	
PRU0_R30[26] / UHPI_HRW / UPP_CHA_WAIT / GP6[8] / PRU1_R31[17]	I/O	
VP_CLKIN0 / UHPI_HCS / PRU1_R30[10] / GP6[7] / UPP_2xTXCLK	I/O	
VP_CLKIN1 / UHPI_HDS1 / PRU1_R30[9] / GP6[6] / PRU1_R31[16]	I/O	
VP_DIN[8] / UHPI_HD[0] / UPP_D[0] / GP6[5] / PRU1_R31[0]	I/O	
VP_CLKIN2 / MMCSD1_DAT[3] / PRU1_R30[3] / GP6[4] / PRU1_R31[4]	I/O	
VP_CLKOUT2 / MMCSD1_DAT[2] / PRU1_R30[2] / GP6[3] / PRU1_R31[3]	I/O	
VP_CLKIN3 / MMCSD1_DAT[1] / PRU1_R30[1] / GP6[2] / PRU1_R31[2]	I/O	
VP_CLKOUT3 / PRU1_R30[0] / GP6[1] / PRU1_R31[1]	I/O	
LCD_AC_ENB_CS / GP6[0] / PRU1_R31[28]	I/O	

这里用到的引脚 PRU0_R30[30] / UHPI_HINT / PRU1_R30[11] / GP6[12]、PRU0_R30[31] / UHPI_HRDY / PRU1_R30[12] / GP6[13]、AXR8 / CLKS1 / ECAP1_APWM1 / GP0[0] / PRU0_R31[8]、AXR9 / DX1 / GP0[1]、AXR10 / DR1 / GP0[2] 和 AXR13 / CLKX1 / GP0[5] 都需要配置成普通输入/输出。那么,在软件上该怎么配置呢?需要通过系统配置模块(System Configuration Module)中的引脚复用寄存器来实现。注意,芯片版本 2.1 以前的 C6748 在修改 SYSCFG 寄存器时是需要解锁 KICK 的。KICK 是一种保护机制,用于保护芯片寄存器不被意外改写。解锁 KICK 只需要在 KICK 寄存器写入特定的值即可。最新版本的芯片改写 SYSCFG 寄存器已经不需要解锁 KICK 了。

系统配置模块(SYSCFG)引脚复用配置寄存器名称及地址如表 1-2 所列。

表 1-2 系统配置模块寄存器

地 址	缩 写	名 称	权 限
0x01C14120	PINMUX0	引脚复用配置寄存器 0	特权模式
0x01C14124	PINMUX1	引脚复用配置寄存器 1	特权模式
0x01C14128	PINMUX2	引脚复用配置寄存器 2	特权模式
0x01C1412C	PINMUX3	引脚复用配置寄存器 3	特权模式
0x01C14130	PINMUX4	引脚复用配置寄存器 4	特权模式
0x01C14134	PINMUX5	引脚复用配置寄存器 5	特权模式
0x01C14138	PINMUX6	引脚复用配置寄存器 6	特权模式
0x01C1413C	PINMUX7	引脚复用配置寄存器 7	特权模式
0x01C14140	PINMUX8	引脚复用配置寄存器 8	特权模式
0x01C14144	PINMUX9	引脚复用配置寄存器 9	特权模式
0x01C14148	PINMUX10	引脚复用配置寄存器 10	特权模式
0x01C1414C	PINMUX11	引脚复用配置寄存器 11	特权模式
0x01C14150	PINMUX12	引脚复用配置寄存器 12	特权模式
0x01C14154	PINMUX13	引脚复用配置寄存器 13	特权模式
0x01C14158	PINMUX14	引脚复用配置寄存器 14	特权模式
0x01C1415C	PINMUX15	引脚复用配置寄存器 15	特权模式
0x01C14160	PINMUX16	引脚复用配置寄存器 16	特权模式
0x01C14164	PINMUX17	引脚复用配置寄存器 17	特权模式
0x01C14168	PINMUX18	引脚复用配置寄存器 18	特权模式
0x01C1416C	PINMUX19	引脚复用配置寄存器 19	特权模式

这里需要用的引脚复用配置位于寄存器 PINMUX1 和 PINMUX13。这两个寄存器的具体位域描述,如表 1-3 所列。

表 1-3 引脚复用配置寄存器位域描述

位	域	值	描 述	类 型
31~28	PINMUX1_31_28		AXR8/CLKS1/ECAP1_APWM1/GP0[0]/PRU0_R31[8] 控制	I
		0	PRU0_R31[8]	I/O
		1h	AXR8	I
		2h	CLKS1	X
		3h	保留	I/O
		4h	ECAP1_APWM1	X
		5h~7h	保留	I/O
		8h	GP0[0]	X
		9h~Fh	保留	
27~24	PINMUX1_27_24		AXR9/DX1/GP0[1]控制	Z
		0	三态	
		1h	AXR9	I/O
		2h	DX1	O
		3~7h	保留	X
		8h	GP0[1]	I/O
23~20	PINMUX1_23_20		AXR10/DR1/GP0[2]控制	Z
		0	三态	
		1h	AXR10	I/O
		2h	DR1	I/O
		3~7h	保留	X
		8h	GP0[2]	I/O
19~16	PINMUX1_19_16		AXR10/DR1/GP0[2]控制	Z
		0	三态	
		1h	AXR11	I/O
		2h	RSX1	I
		3~7h	保留	X
		8h	GP0[3]	I/O
15~12	PINMUX1_15_12		AXR12/FSR1/GP0[4]控制	Z
		0	三态	
		1h	AXR12	I/O
		2h	FSR1	I/O
		3~7h	保留	X
		8h	GP0[4]	I/O
				X
		9h~Fh	保留	