

Season的专栏

目录视图

个人资料



o0溺水的鱼

关注

发私信

访问：24608次

积分：942

等级：

博主

排名：千里之外

原创：68篇

转载：10篇

译文：0篇

评论：4条

文章搜索

文章分类

算法 (9)

《编程之美》 (4)

嵌入式开发——MCU (30)

嵌入式开发——FPGA (5)

光通信 (11)

C语言 (6)

校招 (4)

实验室项目记录 (4)

MATLAB (1)

Android (1)

Linux (2)

cocos2d (1)

[移动信息安全的漏洞和逆向原理](#) [程序员11月书讯，评论得书啦](#) [Get IT技能知识库，50个领域一键直达](#)

无线路灯项目——STM32L151概述

标签: [stm32](#) 2016-05-28 13:41 259人阅读 评论

分类： [嵌入式开发——MCU \(29 \)](#)

版权声明：本文为博主原创文章，未经博主允许不得转载。

STM32L151RCT 256KB 属于Cat.3

在stm32l1xx.h文件中，有下面一幅图，说明了不同容量的MCU应该用哪个startup.s文件

```
/*
The table below provides the list of available part numbers per category, and the corresponding preprocessor
switch to be defined in the StdLib.
*/
```

Memory density level	RPNs	StdLib switch	Cat.1	Cat.2	Cat.3	Cat.4	Cat.5
32KB	STM32L100x6xx, STM32L151x6xx, STM32L152x6xx, STM32L151x6xxA and STM32L152x6xxA	STM32L1XX_MD	X	X			
64KB	STM32L100x8xx, STM32L151x8xx, STM32L152x8xx, STM32L151x8xxA and STM32L152x8xxA	STM32L1XX_MD	X	X			
128KB	STM32L100xBxx, STM32L151xBxx, STM32L152xBxx, STM32L151xBxxA and STM32L152xBxxA	STM32L1XX_MD	X	X			
256KB	STM32L100xCxx, STM32L151xCxx, STM32L152xCxx and STM32L162xCxx	STM32L1XX_MDP			X	X	
384KB	STM32L151xDxx, STM32L152xDxx and STM32L162xDxx	STM32L1XX_HD				X	
512KB	STM32L151xExx, STM32L152xExx and STM32L162xExx	STM32L1XX_XL					X

```
*/
```

有关density和Cat名称的问题，ST官方给出下面的说法：

Replaced “Low density”, “Medium density”, “Medium+ density” and “High density” categories by Cat.1, Cat in all document.

下面是STM32L1系列中Cat.3的系统结构图

阅读排行

阅读排行

评论排行

最新评论

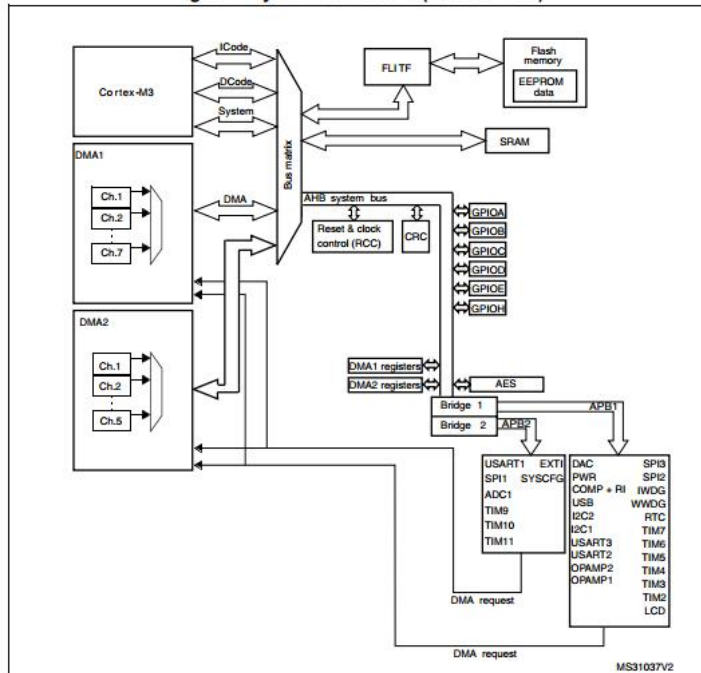
xhr514147982 :非常感谢,解决了我这个初学者花了很长时间都没头绪,并困扰我很久的问题。要是早点发现的话能少走多的弯路...

推荐文章

- * 程序员10月书讯，评论得书
- * Android中Xposed框架篇---修改系统位置信息实现自身隐藏功能
- * Chromium插件（Plugin）模块（Module）加载过程分析
- * Android TV开发总结--构建一个TV app的直播节目实例
- * 架构设计：系统存储--MySQL简单主从方案及暴露的问题



Figure 2. System architecture (Cat.3 devices)



位带区与位带别名区的膨胀关系图

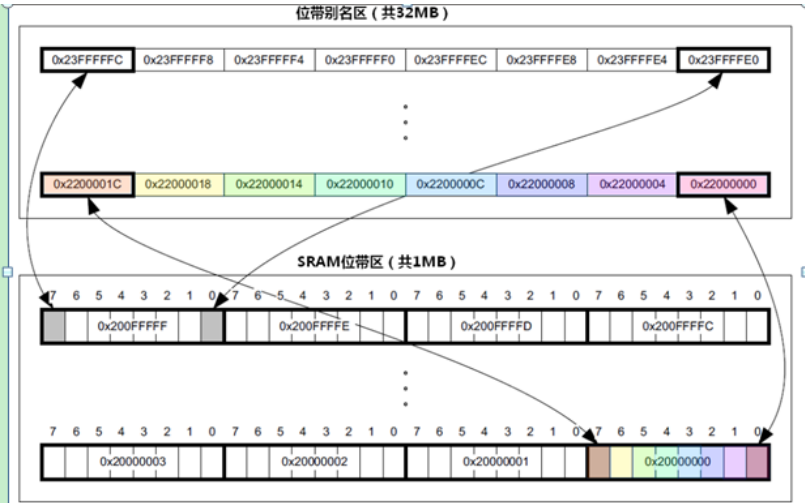
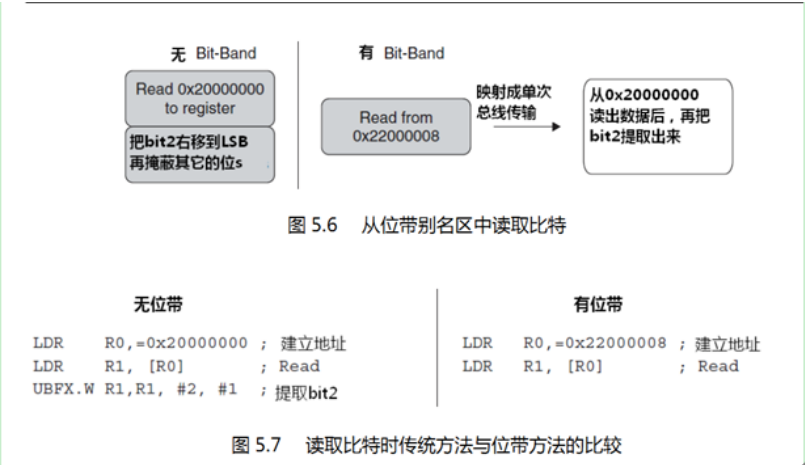
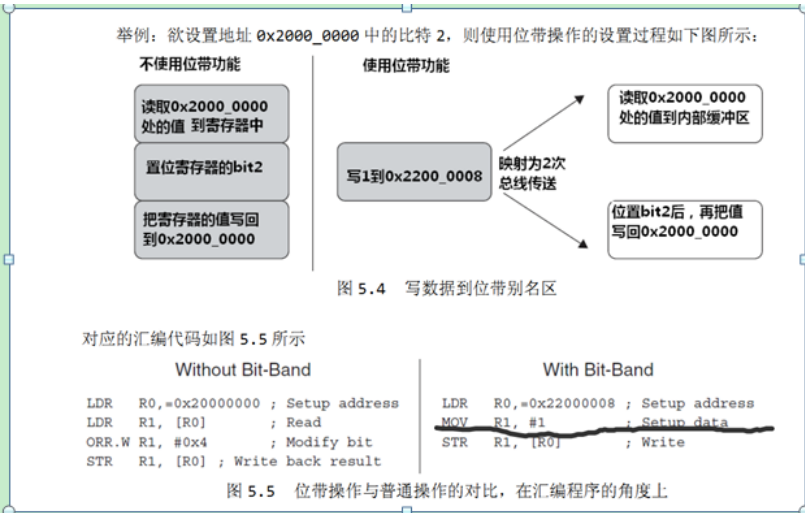


图 5.3B 位带区与位带别名区的膨胀对应关系图 B



位带操作的优越性

- 1.位带操作对于硬件 I/O密集型的底层程序最有用处了
- 2.位带操作还能用来化简跳转的判断。当跳转依据是某个位时，以前必须这样做：

- 读取整个寄存器
- 掩蔽不需要的位
- 比较并跳转

现在只需：

从位带别名区读取状态位

- 比较并跳转

- 3.以前修改一个值需要经过“读 - 改 - 写” 3条指令，导致这中间留有两个能被中断的空档，这在多任务系统中会引起系统紊乱。

作，就可以消灭上述的紊乱。CM3把这个“读 - 改 - 写” 做成一个硬件级别支持的原子操作，不能被中断。

请注意：当使用位带功能时，要访问的变量必须用volatile来定义。因为C编译器并不知道同一个比特可以有两个地址。所以就器每次都如实地把新数值写入存储器，而不再会出于优化的考虑，在中途使用寄存器来操作数据的复本，直到最后才把复本写回式访问同一个位会得到不一致的结果。

对于STM32L151，ST提供的库函数中，大部分还是没有使用bit-band，在stm32l1xx.h中，有下面的宏定义，能进行bit-band部分，他们非位带操作的地址是(uint32_t)0x40000000 和 (uint32_t)0x20000000

```
1 #define SRAM_BB_BASE      ((uint32_t)0x22000000) /*!< SRAM base address in the bit-band region */
2 #define PERIPH_BB_BASE    ((uint32_t)0x42000000) /*!< Peripheral base address in the bit-band regio
```

GPIO

当GPIO配置为input时，无需考虑是推挽（PP）还是开漏（OD）；无需考虑速度，这个速度是描述输出速度；无需考虑输出寄了0或1。

Table 37. Port bit configuration table⁽¹⁾

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [B:A]	PUPDR(i) [1:0]		I/O configuration	
01	0	SPEED [B:A]	0	0	GP output	PP
	0		0	1	GP output	PP + PU
	0		1	0	GP output	PP + PD
	0		1	1	Reserved	
	1		0	0	GP output	OD
	1		0	1	GP output	OD + PU
	1		1	0	GP output	OD + PD
	1		1	1	Reserved (GP output OD)	
10	0	SPEED [B:A]	0	0	AF	PP
	0		0	1	AF	PP + PU
	0		1	0	AF	PP + PD
	0		1	1	Reserved	
	1		0	0	AF	OD
	1		0	1	AF	OD + PU
	1		1	0	AF	OD + PD
	1		1	1	Reserved	

Table 37. Port bit configuration table⁽¹⁾ (continued)

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [B:A]		PUPDR(i) [1:0]		I/O configuration	
00	x	x	x	0	0	Input	Floating
	x	x	x	0	1	Input	PU
	x	x	x	1	0	Input	PD
	x	x	x	1	1	Reserved (input floating)	
11	x	x	x	0	0	Input/output	Analog
	x	x	x	0	1	Reserved	
	x	x	x	1	0		
	x	x	x	1	1		

1. GP = general-purpose, PP = push-pull, PU = pull-up, PD = pull-down, OD = open-drain, AF = alternate function.

【Reference】

《Cortex-M3权威指南》

<http://blog.chinaunix.net/uid-26285146-id-3071387.html>

顶 踩
0 0

- 上一篇 无线路灯项目——SX1278调试
- 下一篇 无线路灯项目——STM32L低功耗相关

我的同类文章

嵌入式开发——MCU (29)

- LWIP之HTTP

2016-05-28

阅读 254

•

无线路灯项目——SIM900A调试

2016-05-28

阅读 422

•

无线路灯项目——SX1278调试

2016-05-28

阅读 1053

•

树莓派入门之装系统

2016-05-28

阅读 455

•

调试uIP出现死机问题

2016-05-28

阅读 105
- 虚拟服务器/DMZ/花生壳

2016-05-

•

无线路灯项目——STM32L低功耗相关

2016-05-

•

STM32 JTAG失效恢复

2016-05-

•

STM32的UART DMA传输总结

2016-05-

•

uIP的ARP协议分析

2016-05-

更多文章

猜你在找

从零写Bootloader及移植uboot、…
Flash动画制作
大数据处理和统一调度系统优化
uboot的硬件驱动部分-2.10. uboo…
Flash实战技能应用从入门到精通

无线路灯项目SIM900A调试
无线路灯项目SX1278调试
KeilMDK422 编译STM32工程报错E…
Error L6218E Undefined symbol…
STM32_TESTaxf Error L6218E Un…



查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

- 全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VP
- IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apa
- HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rai
- Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

S
- Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

An
- Cloud Foundry

Redis

Scala

Django

Bootstrap