龙芯中科
LOONGSON TECHNOLOGY

LOONGSON

龙芯 1C101 处理器用户手册

2018 年 9 月 29 日

龙芯中科技术有限公司

自主决定命运，创新成就未来

北京市海淀区中关村环保科技示范园龙芯产业园 100095
Loongson Industrial Park. Zhongguancun Environmental Protection Park.
Haidian District, Beijing 100095. P.R.China

龙芯中科技术有限公司

Loongson Technology Corporation Limited

地址：北京市海淀区中关村环保科技示范园龙芯产业园 2 号楼

Building No.2, Loongson Industrial Park, Zhongguancun Environmental Protection Park

电话 (Tel)： 010-62546668

传真 (Fax)： 010-62600826

阅读指南

《龙芯 1C101 处理器用户手册》主要介绍龙芯 1C101 的架构与寄存器描述。龙芯 1C101 处理器所集成的 LS132R 处理器核遵循 MIPS32 Release1 标准，可参阅相关架构文档。

修订历史

序号	更新日期	版本号	更新内容
1	2018-3-26	V0.1	初稿，内部评估版本
2	2018-9-29	V1.0	根据样片验证情况更新

目　　录

目录 ····· **i**
第一章　概述 ····· **1**
1.1　特性 ····· 1
1.2　结构框图 ····· 2
1.3　文档约定 ····· 3
1.3.1　信号命名 ····· 3
1.3.2　信号类型 ····· 3
1.3.3　数值表示 ····· 3
1.3.4　寄存器域 ····· 3
第二章　地址空间 ····· **5**
第三章　电源与顶层控制 ····· **7**
3.1　电源管理 ····· 7
3.2　寄存器定义 ····· 7
3.2.1　芯片全局配置 (ChipCtrl) ····· 8
3.2.2　命令与状态 (CmdSts) ····· 10
3.2.3　时间计数器 (Count) ····· 11
3.2.4　唤醒时间配置 (Compare) ····· 11
3.2.5　引脚复用选择 (IOSEL) ····· 12
3.2.6　外部中断使能 (ExintEn) ····· 12
3.2.7　外部中断极性 (ExintPol) ····· 12
3.2.8　外部中断边沿 (ExintEdge) ····· 13
3.2.9　外部中断状态 (ExintSrc) ····· 13
3.2.10　看门狗配置寄存器 (WdtCfg) ····· 13
3.2.11　看门狗重置寄存器 (WdtFeed) ····· 13
3.2.12　电源配置 (PowerCfg) ····· 14
3.2.13　GPIOA 输出使能 (GPIOA_OE) ····· 14
3.2.14　GPIOA 输出电平 (GPIOA_O) ····· 14
3.2.15　GPIOA 输入电平 (GPIOA_I) ····· 15

3.2.16 GPIOB 输出使能 (GPIOB_OE) ····· 15
3.2.17 GPIOB 输出电平 (GPIOB_O) ····· 15
3.2.18 GPIOB 输入电平 (GPIOB_I) ····· 15
3.2.19 脉冲输出配置 (Pulse0/1) ····· 16
3.2.20 用户数据 (UserDat) ····· 16
3.2.21 ADC 控制 (AdcCtrl) ····· 16
3.2.22 ADC 数据寄存器 (AdcDat) ····· 17
3.2.23 GPIO 位访问端口 (GPIOBit) ····· 17
第四章 中断 ····· 19
4.1 中断结构 ····· 19
4.2 中断处理 ····· 20
4.3 寄存器定义 ····· 20
4.3.1 中断使能寄存器 (INT_EN) ····· 21
4.3.2 中断边沿寄存器 (INT_EDGE) ····· 21
4.3.3 中断极性寄存器 (INT_POL) ····· 21
4.3.4 中断清除寄存器 (INT_CLR) ····· 22
4.3.5 中断置位寄存器 (INT_SET) ····· 22
4.3.6 中断输出寄存器 (INT_OUT) ····· 22
4.3.7 运行状态及保护寄存器 (SRPROT) ····· 22
第五章 Flash ····· 25
5.1 概述 ····· 25
5.2 存储空间 ····· 25
5.3 寄存器空间 ····· 25
5.3.1 命令寄存器 (CMD) ····· 26
5.3.2 加密地址上界寄存器 (CAH) ····· 26
5.3.3 加密地址下界寄存器 (CAL) ····· 26
5.3.4 校验数据寄存器 (VRF) ····· 27
5.3.5 状态寄存器 (STS) ····· 27
5.3.6 擦写时间寄存器 (PET) ····· 27
5.4 使用说明 ····· 28
5.4.1 加密支持 ····· 28

5.4.2 代码保护 ····· 28
5.4.3 中断 ····· 28
5.4.4 编程指南 ····· 29
5.4.5 OTP 功能 ····· 29
第六章 定时器 ····· 31
6.1 概述 ····· 31
6.2 寄存器空间 ····· 31
6.2.1 配置寄存器 (CFG) ····· 31
6.2.2 计数值寄存器 (CNT) ····· 32
6.2.3 比较值寄存器 (CMP) ····· 32
6.2.4 步进值寄存器 (STP) ····· 32
6.3 使用说明 ····· 32
第七章 I2C 控制器 ····· 35
7.1 概述 ····· 35
7.2 寄存器定义 ····· 35
7.2.1 分频值低字节寄存器 (PRERL) ····· 35
7.2.2 分频值高字节寄存器 (PRERH) ····· 36
7.2.3 控制寄存器 (CTR) ····· 36
7.2.4 数据寄存器 (DR) ····· 37
7.2.5 命令寄存器 (CR) ····· 37
7.2.6 状态寄存器 (SR) ····· 37
7.2.7 总线死锁时间寄存器 (BLTOP) ····· 38
7.2.8 从设备地址寄存器 (SADDR) ····· 38
第八章 SPI 控制器 ····· 39
8.1 概述 ····· 39
8.2 寄存器定义 ····· 39
8.2.1 控制寄存器 (SPCR) ····· 40
8.2.2 状态寄存器 (SPSR) ····· 40
8.2.3 数据寄存器 (DATA) ····· 41
8.2.4 外部寄存器 (SPER) ····· 41
8.2.5 参数控制寄存器 (PARAM) ····· 41

8.2.6 片选控制寄存器 (SOFTCS) ····· 42
8.2.7 时序控制寄存器 (TIMING) ····· 42
8.3 接口时序 ····· 42
8.3.1 SPI 主控制器接口时序 ····· 42
8.3.2 SPI Flash 访问时序 ····· 43
8.4 使用指南 ····· 43
8.4.1 SPI 主控制器的读写操作 ····· 43
8.4.2 硬件 SPI Flash 读 ····· 44
8.4.3 混合访问 SPI Flash 和 SPI 主控制器 ····· 45
8.4.4 SPI 从模式操作 ····· 45
8.4.5 安装模式 ····· 45
第九章 UART 控制器 ····· 47
9.1 概述 ····· 47
9.2 寄存器定义 ····· 47
9.2.1 数据寄存器 (DAT) ····· 47
9.2.2 中断使能寄存器 (IER) ····· 47
9.2.3 中断状态寄存器 (IIR) ····· 48
9.2.4 FIFO 控制寄存器 (FCR) ····· 48
9.2.5 线路控制寄存器 (LCR) ····· 49
9.2.6 bit 窗口划分和采样控制寄存器 (sample_ctrl) ····· 49
9.2.7 线路状态寄存器 (LSR) ····· 50
9.2.8 发送队列中待发送的数据量 (TF_CNT) ····· 50
9.2.9 状态寄存器寄存器 (STATUS) ····· 51
9.2.10 分频值低字节寄存器 (DL_L) ····· 51
9.2.11 分频值高字节寄存器 (DL_H) ····· 51
9.2.12 分频值小数寄存器 (DL_D) ····· 52
9.3 配置流程 ····· 52
9.3.1 典型例子 ····· 52
第十章 实时时钟 ····· 55
10.1 概述 ····· 55
10.2 寄存器定义 ····· 55

10.2.1 分频值寄存器 (FREQ) ····· 55
10.2.2 配置寄存器 (CFG) ····· 55
10.2.3 时间值寄存器 0 (RTC0) ····· 56
10.2.4 时间值寄存器 1 (RTC1) ····· 56
10.3 说明 ····· 57
第十一章 DMA 控制器 ····· 59
11.1 概述 ····· 59
11.2 寄存器定义 ····· 59
11.2.1 DMA 命令源地址读写端口 (DMA_SOURCE) ····· 59
11.2.2 DMA 命令数据长度读写端口 (DMA_COUNT) ····· 59
11.2.3 命令和状态寄存器 (CMD&STATUS) ····· 60
11.2.4 中断和状态寄存器 (INT&STATUS) ····· 60
11.2.5 命令队列项 0 的源地址参数 (source0) ····· 61
11.2.6 命令队列项 1 的源地址参数 (source1) ····· 61
11.2.7 命令队列项 0 的 DMA 长度参数 (count0) ····· 62
11.2.8 命令队列项 1 的 DMA 长度参数 (count1) ····· 62
11.3 配置流程 ····· 62
11.3.1 典型例子 ····· 62
第十二章 VPWM 模块 ····· 65
12.1 概述 ····· 65
12.2 寄存器定义 ····· 65
12.2.1 算法配置 (VpwmCfg) ····· 65
12.2.2 数据写端口状态 (WPortSts) ····· 66
12.2.3 数据写端口 (WPort) ····· 66
12.2.4 参数配置公式 ····· 67
12.3 输入数据与描述 ····· 67
第十三章 触摸按键控制器 ····· 69
13.1 概述 ····· 69
13.2 寄存器定义 ····· 70
13.2.1 控制寄存器 (TsCtrl) ····· 70
13.2.2 状态寄存器 (TsStat) ····· 71

13.2.3 环振配置寄存器 (OscCfg) ····· 72
13.2.4 扫描时序寄存器 (PollTim) ····· 72
13.2.5 差异阈值寄存器 (DiffThres) ····· 72
13.2.6 最大计数 (CntMax) ····· 73
13.2.7 最小计数 (CntMin) ····· 73
13.2.8 第二小计数 (CntLow) ····· 73
13.2.9 修正寄存器 (CntAdj) ····· 74
13.2.10 计数结果寄存器 (CntRes) ····· 74

表 目 录

1.1 信号类型约定 ····· 3

2.1 地址空间分布 ····· 5
2.2 小地址模式空间分布 ····· 5

3.1 PMU 寄存器列表 ····· 8
3.2 芯片全局配置 ····· 8
3.3 命令与状态 ····· 10
3.4 时间计数器 ····· 11
3.5 唤醒时间配置 ····· 12
3.6 引脚复用选择 ····· 12
3.7 外部中断使能 ····· 12
3.8 外部中断极性 ····· 12
3.9 外部中断边沿 ····· 13
3.10 外部中断状态 ····· 13
3.11 看门狗配置寄存器 ····· 13
3.12 看门狗重置寄存器 ····· 14
3.13 电源配置 ····· 14
3.14 GPIOA 输出使能 ····· 14
3.15 GPIOA 输出电平 ····· 14
3.16 GPIOA 输入电平 ····· 15
3.17 GPIOB 输出使能 ····· 15
3.18 GPIOB 输出电平 ····· 15
3.19 GPIOB 输入电平 ····· 15
3.20 脉冲输出配置 ····· 16
3.21 用户数据 ····· 16
3.22 ADC 控制 ····· 16
3.23 ADC 数据寄存器 ····· 17

3.24 GPIO 位访问端口 ····· 17

4.1 中断对应关系 ····· 19
4.2 Conf 寄存器列表 ····· 20
4.3 中断使能寄存器 ····· 21
4.4 中断边沿寄存器 ····· 21
4.5 中断极性寄存器 ····· 21
4.6 中断清除寄存器 ····· 22
4.7 中断置位寄存器 ····· 22
4.8 中断输出寄存器 ····· 22
4.9 运行状态及保护寄存器 ····· 22

5.1 Flash 控制器寄存器列表 ····· 26
5.2 命令寄存器 ····· 26
5.3 加密地址上界寄存器 ····· 26
5.4 加密地址下界寄存器 ····· 27
5.5 校验数据寄存器 ····· 27
5.6 状态寄存器 ····· 27
5.7 擦写时间寄存器 ····· 27

6.1 HPET 控制器寄存器列表 ····· 31
6.2 配置寄存器 ····· 31
6.3 计数值寄存器 ····· 32
6.4 比较值寄存器 ····· 32
6.5 步进值寄存器 ····· 32

7.1 I2C 控制器寄存器列表 ····· 35
7.2 分频值低字节寄存器 ····· 36
7.3 分频值高字节寄存器 ····· 36
7.4 控制寄存器 ····· 36
7.5 数据寄存器 ····· 37
7.6 命令寄存器 ····· 37
7.7 状态寄存器 ····· 37

7.8 总线死锁时间寄存器 ····· 38
7.9 从设备地址寄存器 ····· 38

8.1 SPI 控制器寄存器列表 ····· 39
8.2 控制寄存器 ····· 40
8.3 状态寄存器 ····· 40
8.4 数据寄存器 ····· 41
8.5 外部寄存器 ····· 41
8.6 SPI 分频系数 ····· 41
8.7 参数控制寄存器 ····· 42
8.8 片选控制寄存器 ····· 42
8.9 时序控制寄存器 ····· 42

9.1 UART 寄存器列表 ····· 47
9.2 数据寄存器 ····· 47
9.3 中断使能寄存器 ····· 47
9.4 中断状态寄存器 ····· 48
9.5 中断控制器功能表 ····· 48
9.6 FIFO 控制寄存器 ····· 49
9.7 线路控制寄存器 ····· 49
9.8 bit 窗口划分和采样控制寄存器 ····· 50
9.9 线路状态寄存器 ····· 50
9.10 发送队列中待发送的数据量 ····· 51
9.11 状态寄存器寄存器 ····· 51
9.12 分频值低字节寄存器 ····· 51
9.13 分频值高字节寄存器 ····· 51
9.14 分频值小数寄存器 ····· 52

10.1 实时时钟寄存器列表 ····· 55
10.2 分频值寄存器 ····· 55
10.3 配置寄存器 ····· 55
10.4 时间值寄存器 0 ····· 56
10.5 时间值寄存器 1 ····· 56

11.1 DMA 寄存器列表 ····· 59
11.2 DMA 命令源地址读写端口 ····· 59
11.3 DMA 命令数据长度读写端口 ····· 60
11.4 命令和状态寄存器 ····· 60
11.5 中断和状态寄存器 ····· 61
11.6 命令队列项 0 的源地址参数 ····· 61
11.7 命令队列项 1 的源地址参数 ····· 62
11.8 命令队列项 0 的 DMA 长度参数 ····· 62
11.9 命令队列项 1 的 DMA 长度参数 ····· 62

12.1 VPWM 寄存器列表 ····· 65
12.2 算法配置 ····· 65
12.3 数据写端口状态 ····· 66
12.4 数据写端口 ····· 67

13.1 触摸按键模块寄存器列表 ····· 70
13.2 控制寄存器 ····· 70
13.3 状态寄存器 ····· 71
13.4 环振配置寄存器 ····· 72
13.5 扫描时序寄存器 ····· 72
13.6 差异阈值寄存器 ····· 73
13.7 最大计数 ····· 73
13.8 最小计数 ····· 73
13.9 第二小计数 ····· 74
13.10 修正寄存器 ····· 74
13.11 计数结果寄存器 ····· 74

图 目 录

1.1 龙芯 1C101 结构图 ····· 2

3.1 电源架构 ····· 7

4.1 中断连接示意 ····· 19

8.1 SPI 模块结构 ····· 39
8.2 SPI 主控制器接口时序 ····· 43
8.3 SPI Flash 标准读时序 ····· 43
8.4 SPI Flash 快速读时序 ····· 43
8.5 SPI Flash 双向 I/O 读时序 ····· 44

13.1 触摸测量结构 ····· 69

此页留空

第一章　概述

龙芯 1C101 是在龙芯 LS1C100 基础上针对门锁应用而优化设计的单片机芯片。该芯片集成 CPU、Flash、SPI、UART、I2C、RTC、TSENSOR、VPWM、ADC 等功能模块，在满足低功耗要求的同时，可以大幅减少板级成本。

1.1　特性

龙芯 1C101 具有以下关键特性：

- LS132R 处理器核
  - 32 位单发射
  - 顺序执行、三级流水
  - 无 cache、MMU
  - EJTAG 调试接口支持断点、单步
  - 4KB 指令 SRAM、4KB 数据 SRAM
  - 最高主频 10MHz
- 片上 Flash
  - 128KB 容量
  - 每页 128 字节
  - 支持代码加密
- SPI 控制器
  - 3 个片选
  - 独立的 Flash 接口，支持启动
- UART 控制器
  - 3 路两线串口
  - 1 路支持唤醒
- I2C 控制器
  - 1 路
  - 支持主从模式
  - 速率 100/400Kbps
- VPWM 控制器
  - 1 路
  - 支持 6K 采样率
  - 支持 ADPCM 压缩

- ADC
  - 6 路输入
  - 12 位分辨率
- 看门狗
  - 上电默认开启
  - 调试模式下暂停
- 定时器
  - 1 路
  - 支持单次、循环模式
  - 调试模式下暂停
- GPIO
  - 64 路复用 GPIO
  - 上电默认为 GPIO 功能，高阻态

1.2 结构框图

芯片以龙芯 LS132R 处理器为计算核心，采用 32 位 AXI+APB 两级总线连接片上资源和外围接口。芯片的结构如图1.1所示。

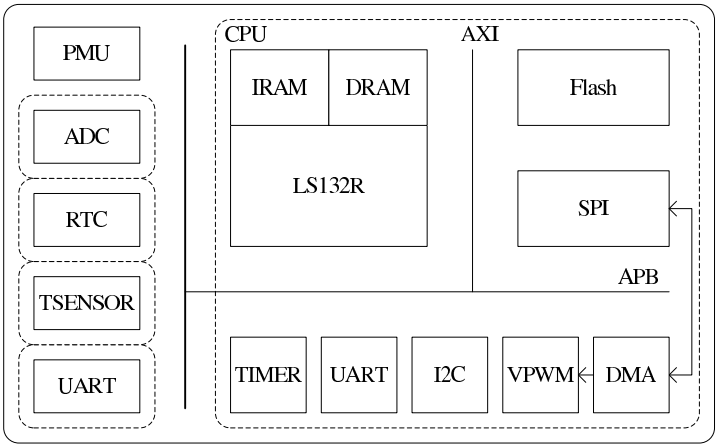

图 1.1: 龙芯 1C101 结构图

1.3 文档约定

1.3.1 信号命名

信号名的选取以方便记忆和明确标识功能为原则。低有效信号以 n 结尾，高有效信号则不带 n。

1.3.2 信号类型

表 1.1: 信号类型约定

代码	描述
A	模拟
DIFF I/O	双向差分
DIFF I	差分输入
DIFF O	差分输出
I	输入
I/O	双向
O	输出
OD	开漏输出
P	电源
G	地

1.3.3 数值表示

16 进制数表示为 'hxxx，2 进制数表示为 'bxx，其他数字为 10 进制数。

功能相同但标号有别的引脚（如 TS00,TS01,...）使用方括号加数字范围的形式简写（如 TS[15:0]）。类似地，寄存器域也采用这种表示方式。

1.3.4 寄存器域

寄存器域以 [寄存器名].[域名] 的形式加以引用。如 ChipCtrl.dram_pd 指芯片配置寄存器（ChipCtrl）的 dram_pd 域。

此页留空

第二章　地址空间

表 2.1: 地址空间分布

地址空间	模块	说明	访问
0xa000_0000 - 0xa000_0fff	IRAM	4KB，可取指	BHW
0xa000_1000 - 0xa000_1fff	DRAM	4KB，不可取指	BHW
0xbe00_0000 - 0xbeff_ffff	SPI	SPI Flash	BHW
0xbf00_0000 - 0xbf01_ffff	Flash	On-chip Flash	BHW
0xbfc0_0000 - 0xbfcf_ffff	Boot	SPI Flash or On-chip Flash	BHW
0xbfe6_0000 - 0xbfe6_0033	Flash	Flash regs	W
0xbfe7_0000 - 0xbfe7_0007	SPI	SPI regs	B
0xbfe8_0000 - 0xbfe8_0007	UART0		B
0xbfe8_8000 - 0xbfe8_8007	UART1		B
0xbfe8_c000 - 0xbfe8_c007	UART2		B
0xbfe9_0000 - 0xbfe9_0007	I2C		B
0xbfea_0000 - 0xbfea_0007	INT		B
0xbfeb_0000 - 0xbfeb_007c	PMU		BW
0xbfeb_4000 - 0xbfeb_40bc	TSENSOR		W
0xbfeb_8000 - 0xbfeb_800c	RTC		W
0xbfec_0000 - 0xbfec_001c	DMA		W
0xbfec_0020 - 0xbfec_002c	VPWM		W
0xbfed_0000 - 0xbfed_000c	TIMER		W

注 1 ：访问类型包括字节（B）、半字（H）和字（W），访存地址必须对齐。

表 2.2: 小地址模式空间分布

地址空间	模块	说明	访问
0x0000_0000 - 0x0000_0fff	IRAM	4KB，可取指	BHW
0x0000_1000 - 0x0000_1fff	DRAM	4KB，不可取指	BHW
0xbe00_0000 - 0xbeff_ffff	SPI	SPI Flash	BHW
0xbf00_0000 - 0xbf01_ffff	Flash	On-chip Flash	BHW
0xbfc0_0000 - 0xbfcf_ffff	Boot	SPI Flash or On-chip Flash	BHW
0x0000_3000 - 0x0000_3033	Flash	Flash regs	W
0x0000_3800 - 0x0000_3807	SPI	SPI regs	B
0x0000_4000 - 0x0000_4007	UART0		B
0x0000_4400 - 0x0000_4407	UART1		B
0x0000_4600 - 0x0000_4607	UART2		B
0x0000_4800 - 0x0000_4807	I2C		B
0x0000_5000 - 0x0000_5007	INT		B
0x0000_5800 - 0x0000_587c	PMU		BW
0x0000_5a00 - 0x0000_5abc	TSENSOR		W
0x0000_5c00 - 0x0000_5c0c	RTC		W

地址空间	模块	说明	访问
0x0000_6000 - 0x0000_601c	DMA		W
0x0000_6020 - 0x0000_602c	VPWM		W
0x0000_6800 - 0x0000_680c	TIMER		W

小地址空间模式默认关闭，如需使用应将 ChipCtrl.compact_mem（见表3.2）置1。

第三章　电源与顶层控制

3.1　电源管理

龙芯 1C101 实现了完善的电源管理，通过有效的软件控制，可大大延长电池寿命。按电源状态分组，龙芯 1C101 包括：电源管理模块（PMU）、实时时钟模块（RTC）、触摸按键模块（TSENSOR）、待机串口模块（UART）、处理器模块（CPU），如图3.1所示。

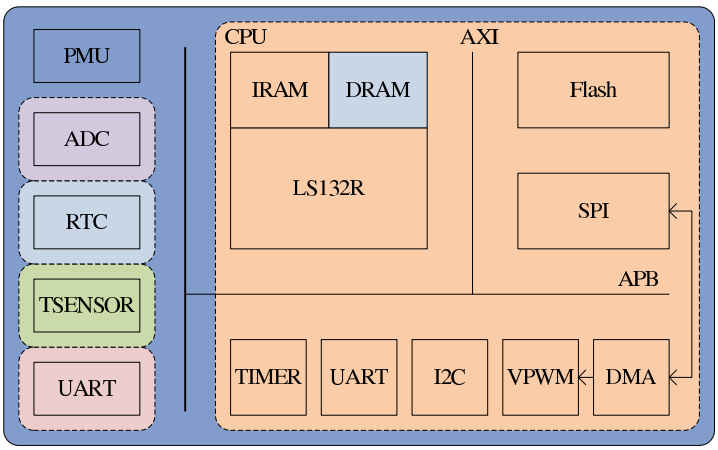

图 3.1: 电源架构

除 PMU 外，其它模块均实现了低功耗关断模式。

CPU 模块的关断由 CmdSts.Sleep 控制，软件写 1 后进入关断模式。当 PMU 看到中断时，CPU 模块将被唤醒。唤醒后 CPU 模块内所有内部寄存器都会被复位，处理器的执行与复位一致，软件需要判断是一次系统复位，还是一次待机唤醒。一个简单的方法是通过 CmdSts.RstSrc 判断，待机唤醒将会是 2'b11。另外还可以在不掉电的 PMU 模块 ChipCtrl.soft_flag 寄存器中存储状态信息。

如果 DRAM 中没有需要休眠保持的数据，设置 ChipCtrl.dram_pd 可以降待机功耗。

其它模块的关断由软件静态配置，没有硬件自动开关。可根据应用需要一次性配好。

3.2　寄存器定义

电源管理模块的基地址为　0xbfeb0000 。除 GPIOBit 使用字节访问外，其它寄存器应当使用字访问。

表 3.1: PMU 寄存器列表

名称	偏移	描述
ChipCtrl	0x00	全局配置
CmdSts	0x04	命令与状态
Count	0x08	时间计数器
Compare	0x0c	唤醒时间配置
IOSEL0	0x10	IO 复用选择 0
IOSEL1	0x14	IO 复用选择 1
IOSEL2	0x18	IO 复用选择 2
IOSEL3	0x1c	IO 复用选择 3
ExintEn	0x20	外部中断使能
ExintPol	0x24	外部中断极性
ExintEdge	0x28	外部中断边沿
ExintSrc	0x2c	外部中断状态
WdtCfg	0x30	看门狗配置
WdtFeed	0x34	看门狗重置
PowerCfg	0x38	电源配置
GPIOA_OE	0x40	GPIOA 输出使能
GPIOA_O	0x44	GPIOA 输出电平
GPIOA_I	0x48	GPIOA 输入电平
GPIOB_OE	0x50	GPIOB 输出使能
GPIOB_O	0x54	GPIOB 输出电平
GPIOB_I	0x58	GPIOB 输入电平
Pulse0	0x60	脉冲输出配置 0
Pulse1	0x64	脉冲输出配置 1
UserDat	0x68	用户数据
AdcCtrl	0x6c	ADC 控制
AdcDat	0x70	ADC 数据
GPIOBit	0x80 ∼0xbf	GPIO 位访问

3.2.1 芯片全局配置 (ChipCtrl)

偏　移：　0x00

复位值：　32'h0000_0008

表 3.2: 芯片全局配置

位域	名称	访问	描述
31:28	soft_flag	RW	**软件标志** 上电复位和看门狗复位初始化为 0，待机唤醒值不变的软件可写寄存器。

位域	名称	访问	描述
27	compact_mem	RW	**小地址空间模式** 0: 常规 32 位空间 1: 64K 小空间 小空间模式下，除内外 Flash 空间位置不变以外，其它空间可由 0 地址开始的 64K 映射过去到大空间 IO 的换算方式为 addr_o = 12'h1fe, addr_i[14:9], 2'b0, 8'b0, addr_i[7:0] 低 8K 同样映射到 RAM，与 0xa0000000 起始的访问一致。
26	spi_start	RW	**SPI 启动速率选择** 发唤醒命令后到 SPI 可用的时间 0: 256us 1: 8us
25:24	batdet_sel	RW	**掉电检测信号选择** 0/1: ADC_I0 2: GPIO00 3: GPIO01
23:20	adc_en	RW	**ADC_I[7:4] 模拟输入使能** 0: 数字输入，可复用为 GPIO 1: 模拟输入，数字输入电路部分关闭
19	adci0_pu	RW	**ADC_I0 400K 上拉** 0: 关闭 1: 打开
18	adci0_pd	RW	**ADC_I0 400K 下拉** 0: 关闭 1: 打开
17	adci0_ien	RW	**ADC_I0 数字输入使能** 0: 关闭，引脚上可以为模拟信号 1: 打开，引脚上是数字信号
16	adc_on	RW	**ADC 电源常开** 0: 自动控制 1: 常开
15	dram_pd	RW	**休眠时数据 RAM 关断** 0: 常开，休眠时有数据需要保存 1: CPU 同时上下电
14	uart2_off	RW	**UART2 关断** 写 1 关断
13	rtc_off	RW	**RTC 关断** 写 1 关断
12	tsensor_off	RW	**触摸按键关断** 写 1 关断
11	fast_ram	RW	**RAM 访问加速** 写 1 使能
10	input_hold	RW	**输入保持** 对处于输入状态的可复用为 GPIO 的引脚，施加与输入值相同的上下拉，写 1 使能。有引脚悬空时，打开此功能可避免漏电。GPIO 方向配置为输入，且 GPIO 输出值配置为 1 的那些引脚才会有此功能。未复用成 GPIO 的引脚同样受 GPIO 配置控制。

位域	名称	访问	描述
9:8	clkup_dly	RW	**高速晶振开启到可以使用的延迟。** 0: 5.14ms 1: 480us 2: 1.46ms 3: 2.44ms
7	c8m_sel	RW	**8M 时钟选择** 0: 内部时钟 1: 外部时钟 变化沿起作用。当外部时钟失效时，将自动切换到内部时钟，但此位保持不变
6	osc8m_en	RW	**高速晶体振荡器使能**
5	c32k_sel	RW	**32K 时钟选择** 0: 内部时钟 1: 外部时钟 变化沿起作用。当外部时钟失效时，将自动切换到内部时钟，但此位保持不变
4	c32k_speed	RW	**内部 32K OSC 速度** 1: 1K 0: 32K
3:0	c32k_trim	RW	**内部 32K OSC Trimming 值**

3.2.2 命令与状态 (CmdSts)

偏　移：　0x04

复位值：　32'h00000000

该寄存器有一些写 1 有效的只写位，这些位在偏移为 0x3c 处有另一个写端口 (CommandW)。往该端口写 1 等价于将 Command 读出，或上要写 1 的位再写回。

表 3.3: 命令与状态

位域	名称	访问	描述
31	clk8m_fail	RO	**8M 外部时钟失效** 1 表示失效
30	clk8m_sel	RO	**8M 时钟选择** 1 表示选外部时钟
29	clk32k_fail	RO	**32K 外部时钟失效** 1 表示失效
28	clk32k_sel	RO	**32K 时钟选择** 1 表示选外部时钟
27:26	RstSrc	RO	**复位来源** 00: 外部复位 01/10: 看门狗复位, 每次看门狗复位均切换 11: 休眠唤醒
25	ExtIntEn	RW	**外部中断使能** 1 有效

位域	名称	访问	描述
24:16	IntSrc	RO	**中断状态** [8]: e_ExtInt [7]: e_ADC [6]: e_RTC [5]: e_C8MFail [4]: e_C32KFail [3]: e_BatFail [2]: e_Uart2 [1]: e_Touch [0]: e_Wake 发生中断后保持为 1，需往 CommandW 寄存器的对应位写 1 以清除中断状态
15:8	IntEn	RW	**中断使能，每一位对应一个中断源** [7]: e_ADC [6]: e_RTC [5]: e_C8MFail [4]: e_C32KFail [3]: e_BatFail [2]: e_Uart2 [1]: e_Touch [0]: e_Wake
7	WakeEn	RW	**定时唤醒使能** 0: 关闭定时唤醒 1: 打开定时唤醒
6:1	reserved	-	保留
0	SleepEn	RO	**进入休眠状态** 当读出值为 1 表示可休眠，此时往 CommandW[0] 写 1 则关闭处理器系统

3.2.3　时间计数器 (Count)

偏　移：　0x8
复位值：　32'h0

表 3.4: 时间计数器

位域	名称	访问	描述
19:0	RTC	RO	**时间计数器** 每 1/256 秒加 1

3.2.4　唤醒时间配置 (Compare)

偏　移：　0x0c
复位值：　32'h0

表 3.5: 唤醒时间配置

位域	名称	访问	描述
19:0	WakeCmp	RW	**唤醒时间配置** 当该值与 Count 寄存器相等且 WakeEn 为 1 时产生唤醒事件

3.2.5 引脚复用选择 (IOSEL)

偏　移：　0x10 ~0x1c
复位值：　32'h0

4 个 32 位寄存器，组成 128 位配置位，由低到高每两位控制一个 GPIO 引脚的复用状态。复位期间所有引脚配置为 GPIO 输入；复位后视封装形式选择，如果选择兼容模式则全部选第二复用，否则为主功能。

表 3.6: 引脚复用选择

位域	名称	访问	描述
1:0	sel	RW	**复用选择** 0: GPIO 1: 主功能 2: 第一复用 3: 第二复用

3.2.6 外部中断使能 (ExintEn)

偏　移：　0x20
复位值：　32'h0

表 3.7: 外部中断使能

位域	名称	访问	描述
31:0	exint_en	RW	**外部中断使能** 0 到 7 位分别对应 GPIO0 到 GPIO7 8 到 15 位分别对应 GPIO16 到 GPIO23 16 到 23 位分别对应 GPIO32 到 GPIO39 24 到 31 位分别对应 GPIO48 到 GPIO55

3.2.7 外部中断极性 (ExintPol)

偏　移：　0x24
复位值：　32'h0

表 3.8: 外部中断极性

位域	名称	访问	描述
31:0	exint_pol	RW	**外部中断极性** 对应关系同上，写 0 为高电平/上升沿有效

3.2.8　外部中断边沿 (ExintEdge)

偏　移：　0x28
复位值：　32'h0

表 3.9: 外部中断边沿

位域	名称	访问	描述
31:0	exint_edge	RW	**外部中断边沿模式选择** 0: 电平模式 1: 边沿模式，对电平宽度无要求

3.2.9　外部中断状态 (ExintSrc)

偏　移：　0x2c
复位值：　32'h0

表 3.10: 外部中断状态

位域	名称	访问	描述
31:0	exint_src	RO	**外部中断状态** 对应关系同上，写 1 为高电平/上升沿有效
31:0	exint_clr	WO	**边沿模式中断清除** 写 1 清除，对电平模式无效

3.2.10　看门狗配置寄存器 (WdtCfg)

偏　移：　0x30
复位值：　32'hfffb0004

看门狗配置必须满足低 16 位奇校验，且高 16 位与低 16 位相反的要求，否则会立即复位。复位后默认的等待延迟为 4 秒。看门狗无法关闭，只有调试模式会暂停计数。

表 3.11: 看门狗配置寄存器

位域	名称	访问	描述
31:16	wdtcfg_hi	RW	**看门狗配置高位** 应当为 wdtcfg_lo 的反
15:0	wdtcfg_lo	RW	**看门狗配置低位** 看门狗复位等待时间，以 1 秒为单位。最高位为奇偶校验位，采用奇校验

3.2.11　看门狗重置寄存器 (WdtFeed)

偏　移：　0x34
复位值：　32'h0

表 3.12: 看门狗重置寄存器

位域	名称	访问	描述
31:0	wdtfeed	WO	**看门狗重置** 写入 0xa55a55aa 将看门狗计数器重置为初始值，写入其它值无效

3.2.12 电源配置 (PowerCfg)

偏　移：　0x38
复位值：　32'h0

表 3.13: 电源配置

位域	名称	访问	描述
31:29	tctrim	RW	**电压调节参数，硬件自动配置，不建议更改**
28:24	abstrim	RW	**电压调节参数，硬件自动配置，不建议更改**
23:0	reserved	-	保留

3.2.13 GPIOA 输出使能 (GPIOA_OE)

偏　移：　0x40
复位值：　32'h0

表 3.14: GPIOA 输出使能

位域	名称	访问	描述
31:0	gpioa_oe[i]	RW	**GPIOA 引脚的输出使能** 0: 输入 1: 输出 管脚对应关系参见数据手册

3.2.14 GPIOA 输出电平 (GPIOA_O)

偏　移：　0x44
复位值：　32'h0

表 3.15: GPIOA 输出电平

位域	名称	访问	描述
31:0	gpioa_out[i]	RW	**GPIOA 引脚的输出电平** 0: 低电平 1: 高电平 输入状态下写 1 表示该引脚需要输入保持功能。参见表 3.2 input_hold。

3.2.15　GPIOA 输入电平 (GPIOA_I)

偏　移：　0x48
复位值：　32'h0

表 3.16: GPIOA 输入电平

位域	名称	访问	描述
31:0	gpioa_in[i]	RO	**GPIOA 引脚的输入** 0: 低电平 1: 高电平

3.2.16　GPIOB 输出使能 (GPIOB_OE)

偏　移：　0x50
复位值：　32'h0

表 3.17: GPIOB 输出使能

位域	名称	访问	描述
31:0	gpiob_oe[i]	RW	**GPIOB 引脚的输出使能** 0: 输入 1: 输出 管脚对应关系参见数据手册

3.2.17　GPIOB 输出电平 (GPIOB_O)

偏　移：　0x54
复位值：　32'h0

表 3.18: GPIOB 输出电平

位域	名称	访问	描述
31:0	gpiob_out[i]	RW	**GPIOB 引脚的输出电平** 0: 低电平 1: 高电平 输入状态下写 1 表示该引脚需要输入保持功能。参见表 3.2 input_hold。

3.2.18　GPIOB 输入电平 (GPIOB_I)

偏　移：　0x58
复位值：　32'h0

表 3.19: GPIOB 输入电平

位域	名称	访问	描述
31:0	gpiob_in[i]	RO	**GPIOB 引脚的输入** 0: 低电平 1: 高电平

3.2.19 脉冲输出配置 (Pulse0/1)

偏　移：0x60/0x64
复位值：32'h0

表 3.20: 脉冲输出配置

位域	名称	访问	描述
17	enable	RW	**脉冲输出使能** 0: 关闭，PULSE* 引脚为 GPIO 模式 1: 打开，根据配置输出占空比 50% 的脉冲信号 在改变时钟选择和 (或) 分频系数前应将使能关闭，完成配置后再打开
16	clk_sel	RW	**时钟源选择** 0: 32K 1: 8M
15:0	pulse_div	RW	**脉冲分频系数** 1~65535: 1~65535 分频

3.2.20 用户数据 (UserDat)

偏　移：0x68
复位值：无

表 3.21: 用户数据

位域	名称	访问	描述
31:0	user_dat	RW	**用户数据** 复位不会清除的数据

3.2.21 ADC 控制 (AdcCtrl)

偏　移：0x6c
复位值：32'h0

表 3.22: ADC 控制

位域	名称	访问	描述
8	run	RWC	**启动单次测量** 写 1 启动，结束后自动清零。如果使能 ADC 中断，还会收到中断
4	div	RW	**时钟分频选择** 0: 2 分频 1: 4 分频

位域	名称	访问	描述
2:0	sel	RW	**通道选择** ADC 测量端口的选择 0: ADC_I0 1: ADC_I1 2: VCORE 3: 1.0V 4: ADC_I4 5: ADC_I5 6: ADC_I6 7: ADC_I7

3.2.22　ADC 数据寄存器 (AdcDat)

偏　移：　0x70
复位值：　-

表 3.23: ADC 数据寄存器

位域	名称	访问	描述
11:0	dout_11_0	RO	**ADC 输出的 12 位数据**

3.2.23　GPIO 位访问端口 (GPIOBit)

偏　移：　0x80 ∼0xbf
复位值：　无

此处有 64 个字节的空间，分别对应 64 个 GPIO。需用字节访问，每个字节内的定义如下

表 3.24: GPIO 位访问端口

位域	名称	访问	描述
1	gpio_oe	WO	**GPIO 方向** 1 为输出
0	gpio_i	RO	**GPIO 输入**
0	gpio_o	WO	**GPIO 输出**

此页留空

第四章　中断

4.1　中断结构

龙芯 1C101 的 CPU 支持 8 个中断事件，其中 6 个来自片内的模块和中断控制器相连。中断的具体连接如图4.1所示。由常开域 PMU 汇总的中断可以起到唤醒系统的作用，由 CPU 域的中断控制器 (INTC) 汇总的中断只能在 CPU 域上电时使用。

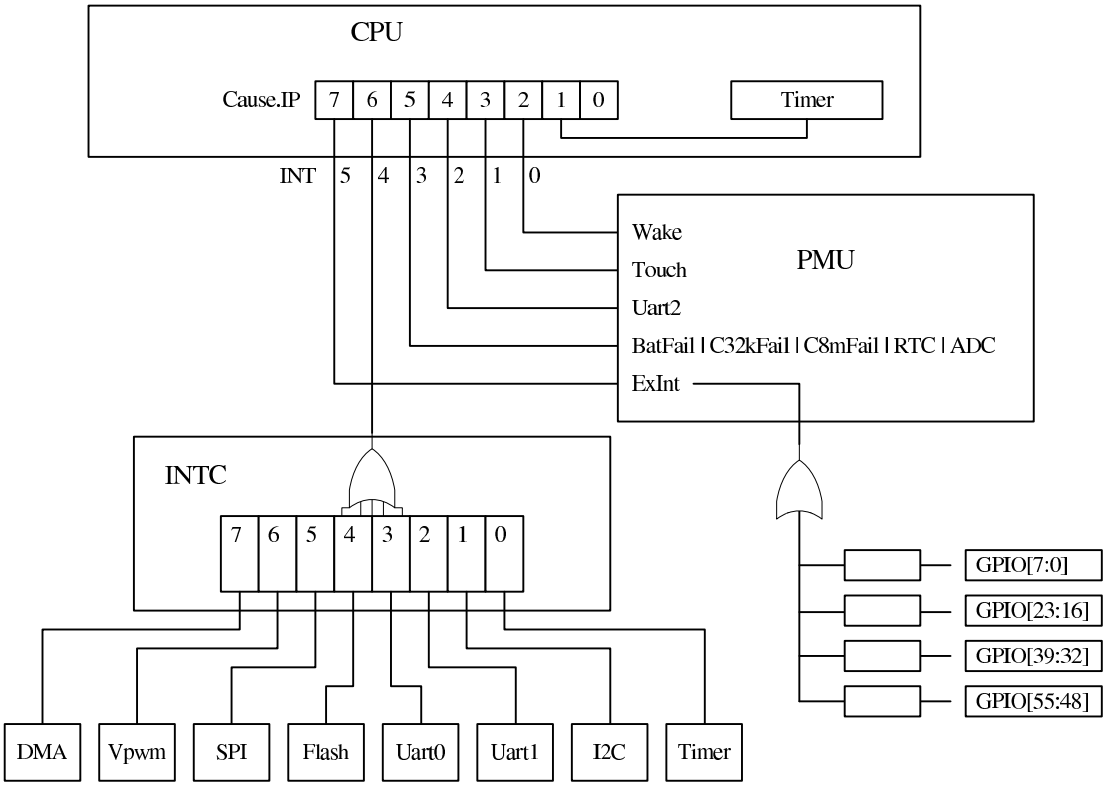

图 4.1: 中断连接示意

表4.1为处理器可见的每个中断源给出了详细的说明

表 4.1: 中断对应关系

中断号	中断名	说明
IP7	ExInt	来自 GPIO 的中断，参见表 3.7~3.10寄存器定义。
IP6	INTC	详见本章寄存器定义。
IP5	PEvent	PMU 中断事件，包括电池掉电、外部时钟失效、RTC 和 ADC 中断。详见第三章。
IP4	Uart2	串口中断，具体见第九章。
IP3	Touch	触摸按键中断，具体见第十三章。

中断号	中断名	说明
IP2	Wake	PMU 唤醒中断，参见 PMU.Count（表3.4）和 PMU.Compare 寄存器。
IP1	Timer	CPU 定时器中断。 禁止计数控制位 (CP0_CAUSE.DC) 复位后为 1，软件使用前需要将其清零。此中断占用 MIPS 定义的软件中断位，不可写。清中断通过写 CP0_COMPARE 寄存器来实现。
IP0	SoftInt	软件中断，写 1 置中断状态。具体用法参见处理器核手册。

4.2 中断处理

当某控制器需要使用中断方式时，需打开整个中断连接路径上的所有中断使能，包括处理器 CP0_STATUS 寄存器的中断使能位，PMU 或者 INTC 中的中断使能位，对应控制器的中断使能位（如果存在）。

CPU 的中断处理入口为 0xbfc00380，需要由软件区分中断源并进行相应的中断处理。中断源为脉冲型的由中断控制寄存器保存状态，清中断时只需清除中断控制寄存器对应位即可。中断源为电平型的则根据中断控制器的配置有所不同，若配置中断控制器配成电平模式，则只需服务中断源对应的模块，清除中断原因；若中断控制器配成边沿模式，则除了服务中断源对应模块外，还应清除中断控制寄存器的对应位。ExInt 的中断类型可配置，其它 PMU 汇总的中断均默认为边沿模式。

中断处理的基本流程如下：

- 将寄存器值保存到 SRAM 中。
- 读取处理器 CP0_CAUSE 寄存器，获取 IP7-IP0 的中断状态，结合 CP0_STATUS 寄存器中的中断使能位，获知处理器中断源。
- 根据表4.1继续查询中断控制器和模块相关寄存器，获知中断发生原因。
- 依据不同中断源进行相关处理，包括事件处理和清中断。
- 处理完成后恢复寄存器值。
- 返回正常执行流。

4.3 寄存器定义

中断相关寄存器存放于 conf 模块，其基地址为 0xbfea0000 。

表 4.2: Conf 寄存器列表

名称	偏移	描述
INT_EN	0x00	中断使能寄存器
INT_EDGE	0x01	中断边沿寄存器
INT_POL	0x02	中断极性寄存器
INT_CLR	0x03	中断清除寄存器
INT_SET	0x04	中断置位寄存器
INT_OUT	0x05	中断输出寄存器
SRPROT	0x06	运行状态及保护寄存器

4.3.1 中断使能寄存器 (INT_EN)

偏 移： 0x00

复位值： 8'h0

表 4.3: 中断使能寄存器

位域	名称	访问	描述
7	dma_int_en	RW	**DMA 中断使能位** 置 1 使能
6	vpwm_int_en	RW	**VPWM 中断使能位** 置 1 使能，表示 VPWM 出现缺数的情况
5	spi_int_en	RW	**SPI 中断使能位** 置 1 使能
4	flash_int_en	RW	**Flash 中断使能位** 置 1 使能
3	uart0_int_en	RW	**UART0 中断使能位** 置 1 使能
2	uart1_int_en	RW	**UART1 中断使能位** 置 1 使能
1	i2c_int_en	RW	**I2C 中断使能位** 置 1 使能
0	timer_int_en	RW	**定时器中断使能位** 置 1 使能

4.3.2 中断边沿寄存器 (INT_EDGE)

偏 移： 0x01

复位值： 8'h40

除外部中断外，其他中断的边沿位由硬件设计决定，软件不应对其进行修改

表 4.4: 中断边沿寄存器

位域	名称	访问	描述
7:0	int_edge	RW	**中断边沿位，对应于中断使能寄存器的各位** 置 1 表示边沿触发，置 0 表示电平触发

4.3.3 中断极性寄存器 (INT_POL)

偏 移： 0x02

复位值： 8'hff

除外部中断外，其他中断的极性位由硬件设计决定，软件不应对其进行修改

表 4.5: 中断极性寄存器

位域	名称	访问	描述
7:0	int_pol	RW	**中断极性位，对应于中断使能寄存器的各位** 置 1 表示高电平/上升沿触发

4.3.4 中断清除寄存器 (INT_CLR)

偏　移：0x03
复位值：8'h0
写 1 后该位自动清零，无需手动清零

表 4.6: 中断清除寄存器

位域	名称	访问	描述
7:0	int_clr	WO	**中断清除位，对应于中断使能寄存器的各位** 置 1 清除内部中断状态

4.3.5 中断置位寄存器 (INT_SET)

偏　移：0x04
复位值：8'h0
写 1 后该位自动清零，通常不使用，仅作为测试中断

表 4.7: 中断置位寄存器

位域	名称	访问	描述
7:0	int_set	WO	**中断置位位，对应于中断使能寄存器的各位** 置 1 置位边沿触发模式的内部中断状态

4.3.6 中断输出寄存器 (INT_OUT)

偏　移：0x05
复位值：8'h0

表 4.8: 中断输出寄存器

位域	名称	访问	描述
7:0	int_out	RO	**中断输出位，对应于中断使能寄存器的各位** 值为 1 表示中断触发，仅在对应中断使能后触发

4.3.7 运行状态及保护寄存器 (SRPROT)

偏　移：0x6
复位值：8'h0

表 4.9: 运行状态及保护寄存器

位域	名称	访问	描述
7	addr_check_en	RW	**地址检查使能** 1 表示进行地址检查，当软件发出未定义的地址时触发 NMI 中断 往此寄存器连续写入 0x00, 0x5a, 0xa5 打开此位写使能
6	reserved	-	保留
5	ejtag_lock	RO	**EJTAG 锁定** 1 表示 EJTAG 接口被禁用

位域	名称	访问	描述
4	otp_lock	RO	**OTP 锁定** 1 表示 OTP 区域被禁写
3	new_pkg	RO	**新封装模式** 表示上电配置非封装兼容模式
2	ejtag_func	RO	**EJTAG 复用** 1 表示 EJTAG 接口可复用为 GPIO
1	install_mode	RO	**安装模式** 1 表示当前为安装模式
0	boot_spi	RO	**SPI 启动** 1 表示当前从 SPI 启动

此页留空

第五章　Flash

5.1　概述

龙芯 1C101 提供了大小为 128KB 的片内 Flash ，可用于程序和数据的存储，Flash 包含 1024 个页，每个页 128 个字节。

Flash 的地址空间包含两个部分：存储空间和寄存器空间。存储空间的基地址为 0xbf000000 ，大小为 128KB。当系统从 Flash 启动时，0xbfc00000 地址也可访问存储空间。寄存器空间的基地址为 0xbfe60000 ，包含 6 个 32 位寄存器。

Flash 还包含一个特殊的 OTP 页用于保存特殊配置值，其访问地址在 128KB 以上，即从 0xbf020000 开始的一个页。该页的最高 4 字节为控制两个配置位，分别表示 OTP 锁定和 EJTAG 锁定。

5.2　存储空间

存储空间与 Flash 的 128KB 存储空间直接对应，可直接读取。

当需要修改 Flash 的内容时，需要进行擦写操作，一次完整的修改流程如下：

1. 清 page_latch
2. 写有效值至目标地址
3. 配置擦除时间
4. 擦除目标页
5. 配置编程时间
6. 编程目标页

上述流程中提及的 page_latch 是 Flash 内部的一个暂存空间，大小与一个页相同。每次清空后，只能对其中每个字节写入一次，否则可能导致数据错误。对存储空间的写操作将直接修改 page_latch ，故写操作的有效范围为一个页。一次擦写前的存储空间写入值应都在一个页中，否则将导致写入错误。对存储空间的写操作只支持 32 位写。

当执行擦除命令时，会将目标页全部擦除；当执行编程命令时，会将 page_latch 中的值编程至 Flash 中。

5.3　寄存器空间

表 5.1: Flash 控制器寄存器列表

名称	偏移	描述
CMD	0x00	命令寄存器
CAH	0x04	加密地址上界寄存器
CAL	0x08	加密地址下界寄存器
VRF	0x10	数据校验寄存器
STS	0x14	状态寄存器
PET	0x18	擦写时间寄存器

5.3.1 命令寄存器 (CMD)

偏　移：0x00
复位值：32'h0

表 5.2: 命令寄存器

位域	名称	访问	描述
31:28	command	RW	**命令** 命令完成后自动清零。 4'b0001 ：数据校验 4'b0011 ：清中断 4'b0100 ：清 page_latch 4'b1001 ：更新区域保护 4'b1010 ：擦除目标页 4'b1100 ：进入休眠模式 4'b1110 ：编程目标页 4'b1111 ：更新密钥
27:18	reserved	-	保留
17:0	pageaddr	RW	**目标页地址** 擦除或编程的目标页地址，范围为 128KB

5.3.2 加密地址上界寄存器 (CAH)

偏　移：0x04
复位值：32'h0

表 5.3: 加密地址上界寄存器

位域	名称	访问	描述
31:18	reserved	-	保留
17:0	caddrh	RW	**加密地址上界** 表示加密范围的上界地址

5.3.3 加密地址下界寄存器 (CAL)

偏　移：0x08
复位值：32'h0

表 5.4: 加密地址下界寄存器

位域	名称	访问	描述
31:18	reserved	-	保留
17:0	caddrl	RW	**加密地址下界** 表示加密范围的下界地址

5.3.4 校验数据寄存器 (VRF)

偏 移： 0x10
复位值： 32'h0

表 5.5: 校验数据寄存器

位域	名称	访问	描述
31:0	verif_data	RW	**校验数据** 欲校验的数据

5.3.5 状态寄存器 (STS)

偏 移： 0x14
复位值： 32'h0

表 5.6: 状态寄存器

位域	名称	访问	描述
31:4	reserved	-	保留
3	no_permission	RO	**无权限** 中断状态为无权限，表示上一次命令无操作权限
2	pe_end	RO	**擦写结束** 中断状态为擦写结束，表示擦写命令完成
1	verif_end	RO	**校验结束** 校验数据的命令执行结束
0	verif_correct	RO	**校验正确** 校验数据的结果为正确

5.3.6 擦写时间寄存器 (PET)

偏 移： 0x18
复位值： 32'h00010

表 5.7: 擦写时间寄存器

位域	名称	访问	描述
31:18	reserved	-	保留
17:16	int_en	RW	**中断使能** 由高位至低位对应于 no_permission、pe_end
15:6	reserved	-	保留

位域	名称	访问	描述
5:3	etime	RW	**擦除时间** 0:1.5ms 1:2.0ms 2:2.5ms 3:3.0ms 4:3.5ms 5:4.0ms 6:4.5ms 7:5.0ms 以 8MHz 时钟计算，默认的擦除时间为 2.5ms
2:0	ptime	RW	**编程时间** 0:1.5ms 1:2.0ms 2:2.5ms 3:3.0ms 4:3.5ms 5:4.0ms 6:4.5ms 7:5.0ms 以 8MHz 时钟计算，默认的编程时间为 1.5ms

5.4 使用说明

5.4.1 加密支持

加解密使用对称算法，采用真随机数作为密钥。当读写地址大于等于加密地址下界且小于加密地址上界时，控制器会进行加密和解密。

新密钥通过更新密钥命令产生并与上下界地址一同存入 flash 的特殊页中。更新密钥命令会自动进行写操作的相关步骤，所以加密地址上下界寄存器应在更新密钥命令之前写好。新写入的加密地址上下界只在使用更新密钥命令后才会生效。

对加密区域的读操作只支持取指令，无法取数据。需要注意的是，一旦重新进行更新密钥命令，老的加密区域内容将被全部擦除。对加密地址的擦写在必须更新密钥后的 5 分钟内完成，超时将无法进行擦写，并产生 no_permission 中断。

加密区域的数据无法读出，但可以通过校验数据命令进行校验。校验前先将待校验数据写入 verif 寄存器，再将需校验的地址和校验数据命令写于命令寄存器，查询状态寄存器，当校验结束时检查校验是否正确。

由于加密区域的数据无法读出，使用时应特别注意要将静态数据段与代码段分开，只针对代码段加密。

5.4.2 代码保护

为了防止误擦写带来的风险，Flash 实现了一种代码保护的机制。

Flash 的存储空间分为三段： 0 ~ 4K 为 ISP 固化代码段， 4K ~ bound 为代码段， bound ~ 128K 为数据段。 ISP 段只能由 ejtag 和 SPI 启动模式改写，代码段只能由 ejtag 、 SPI 启动和 ISP 程序改写，数据段没有限制。

不符合上述规则的擦写将无法完成并产生 no_permission 中断。 bound 地址可通过更新区域保护命令进行配置，该地址值将存入特殊页并自动读出， bound 无法通过软件读取。

5.4.3 中断

Flash 控制器中包含两种中断： pe_end 和 no_permission，中断的状态可通过状态寄存器查询获得。 Pe_end 表示擦写结束， no_permission 表示该次擦写无权限。

在中断处理结束后，需要进行清中断。命令寄存器中的清除中断命令将同时清除两种中断。

5.4.4 编程指南

读取 Flash 中内容

```
temp = *(volatile unsigned int*)(0xbf000000);
```

擦写 Flash 中偏移为 0x100 的页

```
PE_TIME = 0x30019;                          // 配置擦写时间为 3/2ms
CMD     = 0xa0000100;                       // 擦除命令
wait();                                     // 实现 wait 指令，等待中断唤醒
CMD     = 0x40000000;                       // 清除 page_latch
for(i=0xbf000100; i<0xbf000180; i=i+4)  // 128 字节
    *(volatile unsigned int*)(i) = 0x12345678;  // 写入 page_latch
CMD     = 0xe0000100;                       // 编程至 0x100 偏移地址
wait();
```

5.4.5 OTP 功能

OTP 页的最高两字节为 0x5aa5 时表示 OTP 锁定，锁定后 OTP 页只能读出，不可擦除或编程。OTP 页的次高两字节为 0xa55a 时表示 EJTAG 锁定，锁定后芯片的 EJTAG 功能只能读出 IDCODE，并且只能从片内 Flash 启动。OTP 页的其它区域可用于存储用户自定义的产品序列号信息。

此页留空

第六章　定时器

6.1　概述

龙芯 1C101 提供了与 HPET 工作方式类似的 32 位定时器，支持单次定时和周期触发两种模式。

定时器包括 count、compare、step 三个寄存器，当时钟计数器 count 的计数值与 compare 相同时触发中断。在周期触发模式下，触发中断时 compare 自增 step 所存值。定时器工作在主时钟下，定时时间应根据主时钟频率进行设置。

HPET 的基地址为 0xbfed0000 ，寄存器定义见下节。

6.2　寄存器空间

表 6.1: HPET 控制器寄存器列表

名称	偏移	描述
CFG	0x00	配置寄存器
CNT	0x04	计数值寄存器
CMP	0x08	比较值寄存器
STP	0x0c	步进值寄存器

6.2.1　配置寄存器 (CFG)

偏　移：　0x00
复位值：　32'h0

表 6.2: 配置寄存器

位域	名称	访问	描述
31:9	reserved	-	保留
8	int	RW	**中断状态/清中断** 读时表示中断状态，1 表示中断触发；写 1 时清中断状态
7:3	reserved	-	保留
2	periodic	RW	**周期触发** 置 1 使能周期触发，只在开始计数时有效
1	int_en	RW	**中断使能** 置 1 使能中断；不使能中断时中断状态和清除也有效，但处理器不会收到中断
0	start	RW	**计数使能** 置 1 开始计数，使能时无法修改 CNT/CMP/STP 寄存器的值

6.2.2 计数值寄存器 (CNT)

偏　移：0x04
复位值：32'h0

表 6.3: 计数值寄存器

位域	名称	访问	描述
31:0	count	RW	**计数值** 只能在未计数使能时修改；触发中断时不会停止计数；只在计数使能且非处理器调试模式下计数，否则暂停

6.2.3 比较值寄存器 (CMP)

偏　移：0x08
复位值：32'hFFFFFFFF

表 6.4: 比较值寄存器

位域	名称	访问	描述
31:0	compare	RW	**比较值** 只能在未计数使能时修改；周期中断触发时自增步进值

6.2.4 步进值寄存器 (STP)

偏　移：0x0c
复位值：32'h0

表 6.5: 步进值寄存器

位域	名称	访问	描述
31:0	step	RW	**步进值** 只能在未计数使能时修改

6.3 使用说明

例如，在主时钟为 10MHz 时，将定时器配置为首次计时 2 秒、步进 5 秒并打开中断（不包含处理器和中断控制器相关寄存器的配置）。其寄存器配置和中断处理句柄如下所示。

配置 HPET 寄存器

```
CFG = 0;                          // 清零配置寄存器
/* 使能 Confreg 中相关中断位 */
CNT = 0;                          // 清零计数值
CMP = 20000000;                   // 比较值设为 2 秒
STP = 50000000;                   // 步进值设为 5 秒
CFG = 0x7;                        // 开始计数，中断使能，周期触发
```

中断处理句柄

```
CFG = CFG;                          // 读出中断状态并写回，用于清除中断
/* 清除 Confreg 中相关中断位 */
/* 进行定时相关工作，如喂看门狗 */
return 0;
```

此页留空

第七章　I2C 控制器

7.1　概述

龙芯 1C101 芯片集成了一个 I2C 控制器，该控制器可作为 I2C 总线主设备或从设备进行工作。当作为主设备时，控制器可通过轮询或中断方式工作；当作为从设备时，控制器可通过中断方式工作。

龙芯 1C101 中的 I2C 控制器支持的特性包括：

- Standard-Mode, Fast-Mode
- Clock stretch as slave
- Clock synchronization as master

不支持的特性包括：

- Hs-Mode
- 10-bit addressing
- 同时作为主设备和从设备

控制器的基地址为 0xbfe90000 。

7.2　寄存器定义

表 7.1: I2C 控制器寄存器列表

名称	偏移	描述
PRERL	0x00	分频值低字节寄存器
PRERH	0x01	分频值高字节寄存器
CTR	0x02	控制寄存器
DR	0x03	数据寄存器
CR	0x04	命令寄存器
SR	0x04	状态寄存器
BLTOP	0x05	总线死锁时间寄存器
SADDR	0x07	从模式地址寄存器

7.2.1　分频值低字节寄存器 (PRERL)

偏　移：　0x00
复位值：　8'hff

表 7.2: 分频值低字节寄存器

位域	名称	访问	描述
7:0	prer_low	RW	**分频值低字节** 只在作为主设备时有效

7.2.2 分频值高字节寄存器 (PRERH)

偏　移：　0x01
复位值：　8'hff

表 7.3: 分频值高字节寄存器

位域	名称	访问	描述
7:0	prer_high	RW	**分频值高字节** 只在作为主设备时有效

PRERH 和 PRERL 共同组成分频值 PRER，则输出的 SCL 频率为 $\frac{clk_in}{4\times(PRER+1)}$。

7.2.3 控制寄存器 (CTR)

偏　移：　0x02
复位值：　8'h00

表 7.4: 控制寄存器

位域	名称	访问	描述
7	en	RW	**模块工作使能** 1: 正常工作模式 0: 对分频值寄存器进行操作
6	ien	RW	**中断使能** 置 1 使能中断
5	ms	RW	**主从模式选择** 0: 从设备模式 1: 主设备模式
4	txrok	RW	**从设备发送数据准备好** 从设备模式时，当要发送的数据已写入 DR 时，将此位写为 1 ，此位自动清零
3	rxrok	RW	**从设备接收数据已读出** 从设备模式时，当 DR 收到的数据已经被读出时，将此位写为 1 ，此位自动清零
2	reserved	-	保留
1	buslock_check_en	RW	**总线死锁状态检查使能** 使能后，依据 buslock_top 寄存器规定的时间检查总线是否死锁，死锁状态持续周期达到 {buslock_top,16'b0} 认定为产生死锁
0	slv_autoreset_en	RW	**总线死锁时从设备自动复位状态机使能** 使能时产生死锁后从设备会复位自身状态机，从而解除死锁，需要 buslock_check_en 使能

7.2.4 数据寄存器 (DR)

偏 移： 0x03

复位值： 8'h00

表 7.5: 数据寄存器

位域	名称	访问	描述
7:0	TXD/RXD	RW	**数据** 写入时为待发送的数据，读出时为收到的数据

7.2.5 命令寄存器 (CR)

偏 移： 0x04

复位值： 8'h00

控制总线发送、接收等行为，在操作完成或主设备失去仲裁时自动清零

表 7.6: 命令寄存器

位域	名称	访问	描述
7	STA	WO	**开始** 置 1 时，作为主设备时，产生传输开始波形
6	STO	WO	**结束** 置 1 时，作为主设备时，产生传输结束波形
5	RD	WO	**读** 置 1 时，作为主设备时，下一次传输为总线读请求
4	WR	WO	**写** 置 1 时，作为主设备时，下一次传输为总线写请求
3	ACK	WO	**主设备应答** 写 1 表示下一次读数据返回时应答 NACK ，此时连续读请求结束
2	RECOVER	WO	**总线死锁恢复命令** 置 1 时，作为主设备，看到总线死锁状态后，执行此命令解除死锁
1	reserved	-	保留
0	IACK	WO	**中断应答** 写 1 清中断

7.2.6 状态寄存器 (SR)

偏 移： 0x04

复位值： 8'h00

表 7.7: 状态寄存器

位域	名称	访问	描述
7	RxACK	RO	**收到的应答位** 0: 表示收到应答 1: 表示收到 NACK
6	BUSY	RO	**总线忙状态**
5	AL	RO	**失去仲裁** 1：表示主设备失去了总线控制权

位域	名称	访问	描述
4	Slave_addressed	RO	**被寻址** 1：作为从设备时，已被寻址成功
3	Slave_rw	RO	**从设备读写** 0: 表示被读 1: 表示被写
2	buslock	RO	**总线死锁** 1：表示出现总线死锁
1	TIP	RO	**传输进行** 1：主设备有效，表示正在传输
0	IF	RO	**中断标志位** 当传输完一个字节或主设备丢失仲裁时，中断标志位为 1

7.2.7 总线死锁时间寄存器 (BLTOP)

偏　移：0x05
复位值：8'hff

表 7.8: 总线死锁时间寄存器

位域	名称	访问	描述
7:0	buslock_top	RW	**总线死锁时间** 检测使能，死锁状态持续周期达到 {buslock_top,16'b0} 认定为产生死锁

7.2.8 从设备地址寄存器 (SADDR)

偏　移：0x07
复位值：8'h00

表 7.9: 从设备地址寄存器

位域	名称	访问	描述
7	reserved	-	保留
6:0	addr	RW	**从设备地址** 作为从设备时，存放总线地址

第八章　SPI 控制器

8.1　概述

串行外围设备接口 SPI 总线技术是 Motorola 公司推出的多种微处理器、微控制器以及外围设备之间的一种全双工、同步、串行数据接口标准。

龙芯 1C101 中的 SPI 控制器只可作为主控端。对于软件而言，SPI 控制器除了有若干 IO 寄存器外还有一段映射到 SPI Flash 的只读 memory 空间。如果将这段 memory 空间分配在 0xbfc00000 ，复位后不需要软件干预就可以直接访问，从而支持处理器从 SPI Flash 启动。SPI 的 IO 寄存器的基地址为 0xbfe70000 ，外部存储地址空间是 0xbe00,00000xbeff,ffff 共 16MB。

SPI 模块结构如图8.1所示。根据访问类型，来自内部总线的读写请求被分发到 SPI 主控制器和 Flash 读引擎两个子模块。这两个子模块有各自的对外接口，两者可并行访问。Flash 读引擎只支持读取 Flash 内容，擦写操作需要由 SPI 主控制器完成，在内部复用了 SPI_CSn0。PARAM.memory_en 为 0 时表示 SPI 主控制器接到 Flash 接口。

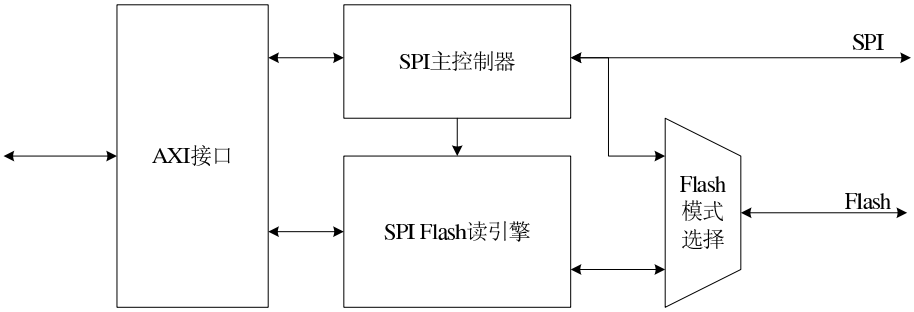

图 8.1: SPI 模块结构

8.2　寄存器定义

SPI 控制器的基地址为 0xbfe70000 。

表 8.1: SPI 控制器寄存器列表

名称	偏移	描述
SPCR	0x00	控制寄存器
SPSR	0x01	状态寄存器
DATA	0x02	数据寄存器
SPER	0x03	外部寄存器
PARAM	0x04	参数控制寄存器

名称	偏移	描述
SOFTCS	0x05	片选控制寄存器
TIMING	0x06	时序控制寄存器

8.2.1 控制寄存器 (SPCR)

偏　移：　0x00
复位值：　8'h12

表 8.2: 控制寄存器

位域	名称	访问	描述
7	spie	RW	**中断数据使能** 高有效
6	spe	RW	**系统工作使能** 高有效
5	reserved	-	保留
4	mstr	RW	**master 模式** 1 为 master 模式，0 为 slave 模式
3	cpol	RW	**时钟极性** 表示无时钟时 SPI_CLK 的电平，1 表示高电平，0 表示低电平
2	cpha	RW	**时钟相位** 0: 相位相同 1: 相位相反
1:0	spr	RW	**时钟分频位** 需与 SPER 的 spre 位一起使用

8.2.2 状态寄存器 (SPSR)

偏　移：　0x01
复位值：　8'h05

表 8.3: 状态寄存器

位域	名称	访问	描述
7	spif	RW	**中断标志位** 值为 1 表示有中断，写 1 清零
6	wcol	RW	**写寄存器溢出标志位** 值为 1 表示溢出，写 1 清零
5	reserved	-	保留
4	busy	R	**表示控制器忙** 主模式下表示 FIFO 非空或状态未完成，从模式下表示 FIFO 非空或片选有效
3	wffull	R	**写寄存器满标志** 1 表示满
2	wfempty	R	**写寄存器空标志** 1 表示空
1	rffull	R	**读寄存器满标志** 1 表示满
0	rfempty	R	**读寄存器空标志** 1 表示空

8.2.3 数据寄存器 (DATA)

偏 移： 0x02
复位值： 8'h00

表 8.4: 数据寄存器

位域	名称	访问	描述
7:0	data	RW	**数据** 写入则为发送数据，读出则为接收数据

8.2.4 外部寄存器 (SPER)

偏 移： 0x03
复位值： 8'h00

表 8.5: 外部寄存器

位域	名称	访问	描述
7:6	icnt	RW	**传输多少字节后发中断** 0:1 1:2 2:3 3:4
5:3	reserved	-	保留
2	mode	RW	**接口模式** 0: 采样与发送时机同时 1: 采样与发送时机错开半周期
1:0	spre	RW	**时钟分频位** 于 spr 一起设定分频比率

表 8.6: SPI 分频系数

spre,spr	分频系数
4'b0000	2
4'b0001	4
4'b0010	16
4'b0011	32
4'b0100	8
4'b0101	64
4'b0110	128
4'b0111	256
4'b1000	512
4'b1001	1024
4'b1010	2048
4'b1011	4096

8.2.5 参数控制寄存器 (PARAM)

偏 移： 0x04
复位值： 8'h21

表 8.7: 参数控制寄存器

位域	名称	访问	描述
7:4	clk_div	RW	**时钟分频数选择** 分频系数与 {spre,spr} 组合相同
3	dual_io	RW	**双 IO 模式** 优先级高于快速读
2	fast_read	RW	**快速读模式**
1	burst_en	RW	**SPI Flash 支持连续地址读模式**
0	memory_en	RW	**SPI Flash 读使能** 0: Flash 编程模式，SPI 控制器通过 CSN0 接入 Flash，其它 SPI 端口不可使用 1: Flash 读出模式，可直接执行指令

8.2.6 片选控制寄存器 (SOFTCS)

偏　移：0x05
复位值：8'hf0

表 8.8: 片选控制寄存器

位域	名称	访问	描述
7:4	csn	RW	**片选** 对应使能有效时，控制 SPI 的四个片选信号。片选 0 对应 flash_csn，其它对应 spi_csn[3:1]。
3:0	csen	RW	**片选使能** 高有效，对应片选

8.2.7 时序控制寄存器 (TIMING)

偏　移：0x06
复位值：8'h03

表 8.9: 时序控制寄存器

位域	名称	访问	描述
7:3	reserved	-	保留
2	tFAST	RW	**SPI flash 读采样模式** 0: 上沿采样，间隔半个 SPI 周期 1: 上沿采样，间隔一个 SPI 周期
1:0	tCSH	RW	**SPI flash 片选信号最短无效时间** 以分频后时钟周期 T 计算 0:1T 1:2T 2:4T 3:8T

8.3 接口时序

8.3.1 SPI 主控制器接口时序

接口时序如图8.2所示。

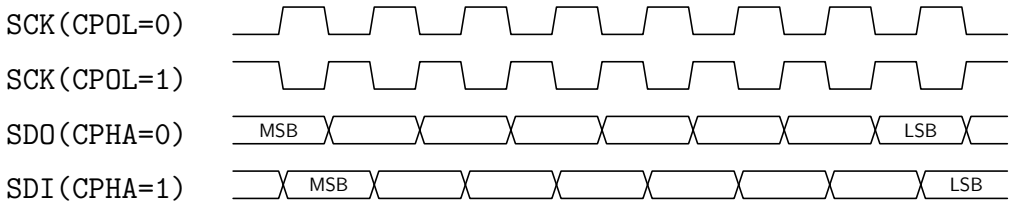

图 8.2: SPI 主控制器接口时序

8.3.2 SPI Flash 访问时序

SPI Flash 的访问时序如图8.3-8.5所示。

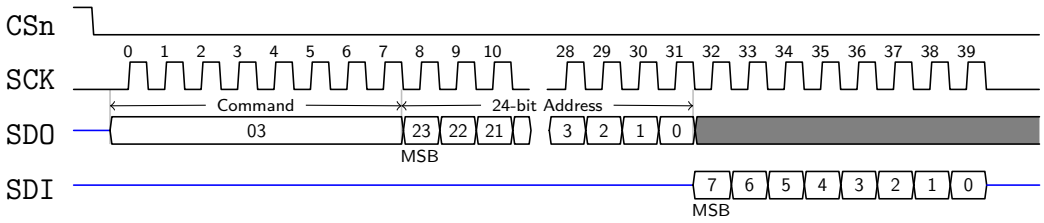

图 8.3: SPI Flash 标准读时序

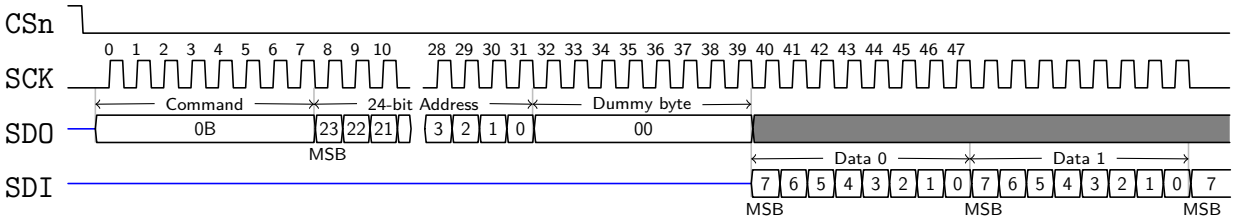

图 8.4: SPI Flash 快速读时序

8.4 使用指南

8.4.1 SPI 主控制器的读写操作

8.4.1.1 模块初始化

- 停止 SPI 控制器工作，对控制寄存器 spcr 的 spe 位写 0
- 重置状态寄存器 spsr，对寄存器写入 8'b1100_0000
- 设置外部寄存器 sper，包括中断申请条件 sper[7:6] 和分频系数 sper[1:0]，具体参考寄存器说明
- 配置 SPI 时序，包括 spcr 的 cpol、cpha 和 sper 的 mode 位。mode 为 1 时是标准 SPI 实现，为 0 时为兼容模式。

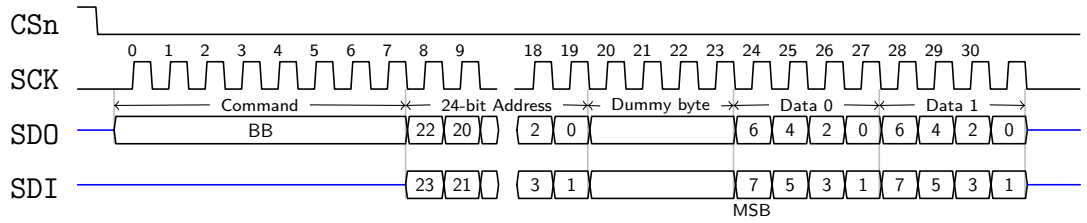

图 8.5: SPI Flash 双向 I/O 读时序

- 配置中断使能，spcr 的 spie 位启动 SPI 控制器，对控制寄存器 spcr 的 spe 位写 1

8.4.1.2 模块的发送/传输操作

- 往数据传输寄存器写入数据
- 传输完成后从数据传输寄存器读出数据。由于发送和接收同时进行，即使 SPI 从设备没有发送有效数据也必须进行读出操作。

8.4.1.3 中断处理

- 接收到中断申请
- 读状态寄存器 spsr 的值，若 spsr[2] 为 1 则表示数据发送完成，若 spsr[0] 为 1 则表示已经接收数据
- 读或写数据传输寄存器
- 往状态寄存器 spsr 的 spif 位写 1，清除控制器的中断申请

8.4.2 硬件 SPI Flash 读

8.4.2.1 初始化

- 将 SFC_PARAM 的 memory_en 位写 1。当 SPI 被选为启动设备时此位复位为 1。
- 设置读参数 (时钟分频、连续地址读、快速读、双 I/O、tCSH 等)。这些参数复位值均为最保守的值。

8.4.2.2 更改参数

如果所使用的 SPI Flash 支持更高的频率或者提供增强功能，修改相应参数可以大大加快 Flash 的访问速度。参数的修改不需要关闭 SPI Flash 读使能 (memory_en)。具体参考寄存器说明。

8.4.3 混合访问 SPI Flash 和 SPI 主控制器

将 SPI Flash 读使能关闭后，软件就可直接控制 csn[0]，并通过 SPI 主控制器访问 SPI 总线。这意味着在进行此操作时，不能从 SPI Flash 中取指。

除了读以外，SPI Flash 还实现了很多命令 (如擦除、写入)，具体参见相关 Flash 的文档。

8.4.4 SPI 从模式操作

- 将 SOFTCS 寄存器最低位设为 0，将片选配置为输入
- 将 PARAM 寄存器的 memory_en 位清零
- 使能 SPI 控制器并设置为从模式
- 根据读写 FIFO 状态及 busy 状态写入或读出数据

8.4.5 安装模式

安装模式（BS1 上拉并且 FLASH_CSNB 上拉）用于简化出厂时的固件烧写。在该模式下芯片会从安装卡上的 SPI Flash 启动。安装卡上放置一片 SPI Flash，其连接与板上 Flash 的区别仅在于片选使用 CSB，并且 CSB 上有上拉电阻。

安装模式下处理器执行安装卡上的软件，烧写内、外 Flash。此时板上的 Flash 只能由普通 SPI 控制。烧写外部 Flash 期间，无法执行安装卡上 Flash 的指令，因此需要把相应的烧写代码拷贝到内部 RAM 执行。

值得注意的是，当芯片被锁定后将无法进入安装模式。

此页留空

第九章　UART 控制器

9.1　概述

龙芯 1C101 有 3 个 UART 控制器，其中 UART0/1 使用总线接口时钟作为波特率时钟，UART2 使用 32K 时钟。

9.2　寄存器定义

UART 寄存器基地址为 0xbfe80000 、 0xbfe88000 和 0xbfe8c000 。

表 9.1: UART 寄存器列表

名称	偏移	描述
DAT/DL_L	0x00	数据寄存器/分频值低字节寄存器
IER/DL_H	0x01	中断使能寄存器/分频值高字节寄存器
IIR	0x02	中断状态寄存器
FCR/DL_D	0x02	FIFO 控制寄存器/分频值小数寄存器
LCR	0x03	线路控制寄存器
sample_ctrl	0x04	bit 窗口划分和采样控制寄存器
LSR	0x05	线路状态寄存器
TF_CNT	0x06	发送队列数据存量
STATUS	0x07	状态寄存器寄存器

9.2.1　数据寄存器 (DAT)

偏　移：　0x00
复位值：　8'h0

表 9.2: 数据寄存器

位域	名称	访问	描述
7:0	data	RW	**数据** 读此寄存器时为收到的数据，写此寄存器将待发送的数据写入发送 FIFO

9.2.2　中断使能寄存器 (IER)

偏　移：　0x01
复位值：　8'h00

表 9.3: 中断使能寄存器

位域	名称	访问	描述
7:4	reserved	-	保留

位域	名称	访问	描述
3	IME	RW	**Modem 状态中断使能** 值为 1 使能
2	ILE	RW	**线路状态中断使能** 值为 1 使能
1	ITE	RW	**发送状态中断使能** 值为 1 使能
0	IRE	RW	**接收状态中断使能** 值为 1 使能

9.2.3 中断状态寄存器 (IIR)

偏　移：　0x02
复位值：　8'h01

表 9.4: 中断状态寄存器

位域	名称	访问	描述
7:4	reserved	-	保留
3:1	II	RO	**中断源** 中断源，详见下表。
0	INTPn	RO	**中断未决状态** 低有效，表示存在未处理的中断

表 9.5: 中断控制器功能表

II	优先级	中断类型	中断源	中断复位控制
3'b011	1	线路状态	奇偶、溢出或帧错误，或打断中断	读 LSR
3'b010	2	接收状态	接收到的数据数量达到了 trigger 值	读数据寄存器
3'b110	2	接收状态	接收超时，接收缓冲中有字符数据且在后续两个字符时间内无操作	读数据寄存器
3'b001	3	发送状态	发送 FIFO 为空	写数据寄存器或读中断状态寄存器
3'b000	4	Modem 状态	reserved，两线串口实现无此中断源	读 Modem 状态寄存器

9.2.4 FIFO 控制寄存器 (FCR)

偏　移：　0x02
复位值：　8'h80

表 9.6: FIFO 控制寄存器

位域	名称	访问	描述
7:3	trigger	WO	**接收中断状态所需 trigger** 单位为字节 0x0, 0x1: 1 字节 0x2: 2 字节 ... 0x10: 16 字节 其它值：保留
2	txreset	WO	**复位发送 FIFO** 此位置位后自动清零
1	rxreset	WO	**复位接收 FIFO** 此位置位后自动清零
0	reserved	-	保留

9.2.5 线路控制寄存器 (LCR)

偏 移： 0x03
复位值： 8'h03

表 9.7: 线路控制寄存器

位域	名称	访问	描述
7	dlab	RW	**分频器模式** 0: 访问正常寄存器 1: 访问分频值寄存器
6	bcb	RW	**打断控制位** 0: 正常操作 1: 串口输出置为 0 (打断状态)
5	spd	RW	**指定奇偶校验位** 0: 不用指定奇偶校验位 1: 如果 eps 为 1 则传输和检查奇偶校验位为 0 ; 如果 eps 为 0 则传输和奇偶校验位为 1
4	eps	RW	**奇偶校验位选择** 0: 奇校验 1: 偶校验
3	pe	RW	**奇偶校验位使能** 0: 无奇偶校验位 1: 使能，输出校验位，输入判断校验位
2	sb	RW	**生成停止位位数** 0: 1 个停止位 1: bec 为 5 时 1.5 个停止位，其他值时 2 个停止位
1:0	bec	RW	**字符位数** 0: 5 位 1: 6 位 2: 7 位 3: 8 位

9.2.6 bit 窗口划分和采样控制寄存器 (sample_ctrl)

偏 移： 0x04
复位值： 8'h70

表 9.8: bit 窗口划分和采样控制寄存器

位域	名称	访问	描述
7:4	sample_point	RW	**bit 采样点位置** bit 采样点位于 bit 划分窗口中的位置 其允许值为 win_size -1 到 1 当 win_size 为 0 时，其值的允许范围为：15 ~ 1
3:0	win_size	RW	**bit 采样窗口长度** 将 1bit 的时间划分为多少份 0x0: 16 份 0x1, 0x2: 保留值 0x3: 3 份 ... 0xf: 15 份

9.2.7 线路状态寄存器 (LSR)

偏　移：　0x05

复位值：　8'h00

对此寄存器进行读操作时，LSR[4:1] 和 LSR[7] 被清零，LSR[6:5] 在给传输 FIFO 写数据时清零，LSR[0] 则对接收 FIFO 进行判断。

表 9.9: 线路状态寄存器

位域	名称	访问	描述
7	error	RO	**错误表示位** 0: 无错误 1: 有奇偶校验错误、帧错误或打断中断
6	TE	RO	**传输为空表示位** 0: 有数据 1: 传输 FIFO 和传输移位寄存器都为空。给传输 FIFO 写数据时清零
5	TFE	RO	**传输 FIFO 为空表示位** 0: 有数据 1: 当前传输 FIFO 为空，给传输 FIFO 写数据时清零
4	BI	RO	**打断中断表示位** 0: 没有中断 1: 接收到起始位+数据+奇偶位+停止位都是 0 ，即有打断中断
3	FE	RO	**帧错误表示位** 0: 没有错误 1: 接收的数据没有停止位
2	PE	RO	**奇偶校验位错误表示位** 0: 没有奇偶错误 1: 当前接收数据有奇偶错误
1	OE	RO	**数据溢出表示位** 0: 无溢出 1: 有数据溢出
0	DR	RO	**接收数据有效表示位** 0: 在 FIFO 中无数据 1: 在 FIFO 中有数据

9.2.8 发送队列中待发送的数据量 (TF_CNT)

偏　移：　0x06

复位值：　8'h00

表 9.10: 发送队列中待发送的数据量

位域	名称	访问	描述
7	loopback	RW	**自回环模式控制位** 0：正常工作模式；1：自回环模式
6:5	reserved	-	保留
4:0	Tf_count	RO	**发送队列中待发送的数据量** 单位为字节

在自回环模式下，原 TX 的输出被直接在内部引入 RX 的输入， TX 对外输出的管脚输出高电平。

9.2.9 状态寄存器寄存器 (STATUS)

偏 移： 0x07
复位值： 8'h00

表 9.11: 状态寄存器寄存器

位域	名称	访问	描述
7	RX_RST	RO	**接收数据通路中 32K 时钟域的复位状态** 0：不在复位状态; 1：在复位状态
6	CLK32K_RST	RO	**控制逻辑 32K 时钟域的复位状态** 0：不在复位状态; 1：在复位状态
5	flush_wait	RO	**接收数据通路中数据丢弃等待标识** 此位为 1，表示 RX 数据通路正处于 flush 数据的过程中
4:0	Rf_count	RO	**接收队列中的数据量** 单位为字节

9.2.10 分频值低字节寄存器 (DL_L)

偏 移： 0x00
复位值： 8'h0

表 9.12: 分频值低字节寄存器

位域	名称	访问	描述
7:0	low	RW	**分频值整数部分低字节**

9.2.11 分频值高字节寄存器 (DL_H)

偏 移： 0x01
复位值： 8'h0

表 9.13: 分频值高字节寄存器

位域	名称	访问	描述
7:0	high	RW	**分频值整数部分高字节**

9.2.12 分频值小数寄存器 (DL_D)

偏　移：0x02
复位值：8'h0

表 9.14: 分频值小数寄存器

位域	名称	访问	描述
7:0	deci	RW	**分频值小数部分的二进制表示，如 0xc0 表示 0.75**

由 DL_H、DL_L、DL_D 组成分频值寄存器 DL，则波特率为 $\frac{32768}{win\_size \times DL}$。UART 控制器的总线收发时钟的频率为 32768 Hz 。如果目标波特率为 2400，则 sample_ctrl 使用缺省设置（ win_size = 0x8， sample_point = 0x3 ）， DL = 1.7046875，故 DL_H = 0x0, DL_L = 0x1, DL_D = 0xb6。如果目标的波特率为 9600，则 sample_ctrl = 0x23（ win_size = 0x3， sample_point = 0x2 ）， DL = 1.1377778，故 DL_H = 0x0, DL_L = 0x1, DL_D = 0x23。波特率的配置值与期望值的误差应在 3% 以内，否则无法识别所有数据位，将导致乱码。

9.3 配置流程

9.3.1 典型例子

假设我们的 UART 总线收发时钟为 32768Hz，目标波特率为 9600，我们需要执行下列初始化操作：

1. 配置 bit 窗口
   - 置 sample_ctrl 的值为 0x23
2. 打开分频器
   - 置 LCR 的 dlab 位为 1
   - 置 DL_D 的值为 0x23
   - 置 DL_H 的值为 0x0
   - 置 DL_L 的值为 0x1
   - 置 LCR 的 dlab 位为 0
3. 配置 LCR
   - 根据帧的格式配置 LCR 的值
4. 配置 FCR
   - 置 FCR 为 0x86 RX 接收的 trigger 值为 16，同时复位发送和接收数据通路
5. 查询复位结束
   - 查询寄存器 0x7 的值，直到其值的 [6] 为 0
6. 开中断使能

- 置 IER 为 0x1 打开接收通路中断使能

此页留空

第十章　实时时钟

10.1　概述

实时时钟模块提供年、月、日、时、分、秒、1/16 秒计数和一个定时中断。定时中断可将芯片从休眠状态唤醒。计时相关逻辑无复位电路，不会因为芯片复位而丢失时间信息。实时时钟的精度依赖于时钟源的精度，当时钟源的频率出现固定偏斜时，可通过修改分频系数来进行修正。

10.2　寄存器定义

实时时钟寄存器基址为 0xbfeb8000 。

表 10.1: 实时时钟寄存器列表

名称	偏移	描述
FREQ	0x00	分频值寄存器
CFG	0x04	配置寄存器
RTC0	0x08	时间值寄存器 0
RTC1	0x0c	时间值寄存器 1

10.2.1　分频值寄存器 (FREQ)

偏　移：　0x00

复位值：　无复位值

内部需要产生 1/16 秒的事件，通过本寄存器的配置分频得到 16Hz 时钟

表 10.2: 分频值寄存器

位域	名称	访问	描述
27:6	freqscale	RW	**分频系数** 27:16 位为整数部分， 15:6 位为小数部分，$freqscale = \frac{freq\_in}{16}$，freq_in 为输入时钟频率，以 HZ 为单位

10.2.2　配置寄存器 (CFG)

偏　移：　0x04

复位值：　32'h0

表 10.3: 配置寄存器

位域	名称	访问	描述
31	state	RW	**操作进行状态** 为 1 表示原子读写序列执行中，写 1 强制清零。仅供硬件调试使用

位域	名称	访问	描述
30	timer_en	RW	**定时器使能** 写 1 使能定时器，时间到后自动清此位
29:26	timer_month	RW	**定时器月** 表示对应单位的时间，用于定时器比较
25:21	timer_day	RW	**定时器日** 表示对应单位的时间，用于定时器比较
20:16	timer_hour	RW	**定时器小时** 表示对应单位的时间，用于定时器比较
15:10	timer_minute	RW	**定时器分钟** 表示对应单位的时间，用于定时器比较
9:4	timer_second	RW	**定时器秒** 表示对应单位的时间，用于定时器比较
3:0	timer_sixteenth	RW	**定时器十六分之一秒** 表示对应单位的时间，用于定时器比较

10.2.3 时间值寄存器 0 (RTC0)

偏　移：0x08

复位值：32'h0

表 10.4: 时间值寄存器 0

位域	名称	访问	描述
31	bad_time	RO	**无效数值** 表示所读出的时间无效
30:21	reserved	-	保留
20:16	hour	RW	**小时** 表示对应单位的时间，写时为待更新时间，读时为当前实时时钟时间
15:10	minute	RW	**分钟** 表示对应单位的时间，写时为待更新时间，读时为当前实时时钟时间
9:4	second	RW	**秒** 表示对应单位的时间，写时为待更新时间，读时为当前实时时钟时间
3:0	sixteenth	RW	**十六分之一秒** 表示对应单位的时间，写时为待更新时间，读时为当前实时时钟时间

10.2.4 时间值寄存器 1 (RTC1)

偏　移：0x0c

复位值：32'h0

读出此寄存器后必须接着读出 RTC0。

表 10.5: 时间值寄存器 1

位域	名称	访问	描述
31	bad_time	RO	**无效数值** 表示所读出的时间无效
30:16	reserved	-	保留

位域	名称	访问	描述
15:9	year	RW	**年** 表示对应单位的时间，写时为待更新时间，读时为当前实时时钟时间，年计数自公元 2000 年起
8:5	month	RW	**月** 表示对应单位的时间，写时为待更新时间，读时为当前实时时钟时间
4:0	day	RW	**日** 表示对应单位的时间，写时为待更新时间，读时为当前实时时钟时间

10.3 说明

实时时钟的功能包括：

- 时间配置
  配置时间时先后写 RTC0 和 RTC1 寄存器，中间不允许插入其它寄存器访问。
- 实时时间读取
  直接读 RTC0 取得时分秒；或者先读 RTC1 后读 RTC0，中间不允许插入其它寄存器访问。内部采样寄存器周期性地从实时时钟中取得时间，当读出 bad_time 为 1 时说明采样未完成，应当再次读出。
- 定时中断
  配置目标时间，然后使能定时器。定时器与实时时钟进行比较，相等时产生中断，使能信号自动清零。

此页留空

第十一章 DMA 控制器

11.1 概述

龙芯 1C101 的 DMA 控制器是一个将 SPI flash 中存储的音频数据搬运到专用输出接口上的 DMA 控制器。其有 1 个可以存储 2 项待执行命令的 DMA 命令队列和一个可以缓冲 2 项 32 位 DMA 数据的数据缓冲。DMA 控制器支持软复位功能。软复位将清空 DMA 控制器的命令队列和数据缓冲。

11.2 寄存器定义

DMA 控制器寄存器基地址为 0xbfec0000 。

表 11.1: DMA 寄存器列表

名称	偏移	描述
DMA_SOURCE	0x00	DMA 命令源地址读写端口
DMA_COUNT	0x04	DMA 命令数据长度读写端口
CMD&STATUS	0x08	命令和状态寄存器
INT&STATUS	0x0c	中断和状态寄存器
source0	0x10	命令队列项 0 的源地址参数
source1	0x14	命令队列项 1 的源地址参数
count0	0x18	命令队列项 0 的 DMA 长度参数
count1	0x1c	命令队列项 1 的 DMA 长度参数

11.2.1 DMA 命令源地址读写端口 (DMA_SOURCE)

偏 移： 0x00
复位值： 32'h0

表 11.2: DMA 命令源地址读写端口

位域	名称	访问	描述
31:0	source	RW	**DMA 命令的源地址参数** 读：DMA 命令队列中当前执行命令项的源地址参数 写：DMA 命令队列当前写入项的源地址参数

从这个寄存器地址读到的 DMA 命令队列当前执行项的源地址参数的值会随着当前命令项 DMA 操作的执行而发生变化。

11.2.2 DMA 命令数据长度读写端口 (DMA_COUNT)

偏 移： 0x04

复位值： 32'h0

表 11.3: DMA 命令数据长度读写端口

位域	名称	访问	描述
31:0	count	RW	**DMA 命令的数据长度参数** 读：DMA 命令队列中当前执行命令项的数据长度参数 写：DMA 命令队列当前写入项的数据长度参数

从这个寄存器地址读到的 DMA 命令队列当前执行项的数据搬运个数参数的值会随着当前命令项 DMA 操作的执行而发生变化。这个参数的单位为数据搬运的个数，而不是数据搬运的字节数。 DMA 控制器搬运的 1 个数据为一个 32 位 word 。

11.2.3 命令和状态寄存器 (CMD&STATUS)

偏 移： 0x08
复位值： 32'h01

表 11.4: 命令和状态寄存器

位域	名称	访问	描述
31	soft_rst	RW	**DMA 控制器软复位** 写：向此位写入 1 将使 DMA 控制器复位 读：此位为 1 表示 DMA 控制器软复位正在进行；此位为 0 表示 DMA 的软复位已经完成
30:2	reserved	-	保留
1	indicate_first	WO	**标识 DMA 的第一个数据** 写：在向 [0] 位写入 1 的同时向此位写入 1, 则要求该 DMA 命令在执行时标识返回的第一个 DMA 数据 在向 [0] 位写入 1 的同时向此位写入 0, 则不要求该 DMA 命令在执行时标识返回的第一个 DMA 数据
0	cmd_en	RW	**DMA 命令生效** 写：向此位写入 1, 将之前配置的源地址和数据长度参数送入 DMA 命令队列 读：此位为 1 表示 DMA 命令队列可写；此位为 0 表示 DMA 命令队列不可写

在配置 DMA_SOURCE 和 DMA_COUNT 前必须先查询 CMD&STATUS 的 [0]，确保 DMA 的命令队列可写。当命令队列不可写时，对 DMA_SOURCE 和 DMA_COUNT 的写操作无效。

11.2.4 中断和状态寄存器 (INT&STATUS)

偏 移： 0x0c
复位值： 32'h00

表 11.5: 中断和状态寄存器

位域	名称	访问	描述
31:20	reserved	-	保留
19	buf_rptr	RO	**DMA 数据缓冲读指针** DMA 数据缓冲的读指针值
18	buf_wptr	RO	**DMA 数据缓冲写指针** DMA 数据缓冲的写指针值
17:16	buf_cnt	RO	**DMA 数据缓冲计数器** DMA 数据缓冲存储的有效数据项数
15:14	reserved	-	保留
13	cmd_rptr	RO	**DMA 命令队列读指针** DMA 命令队列的读指针值
12	cmd_wptr	RO	**DMA 命令队列写指针** DMA 命令队列的写指针值
11:10	cmd1	RO	**DMA 命令队列 1 的命令值** 0x0：命令无效 0x1：命令有效，执行时不需要标识第一个数据 0x3: 命令有效，执行时需要标识第一个数据
9:8	cmd0	RO	**DMA 命令队列 0 的命令值** 0x0：命令无效 0x1：命令有效，执行时不需要标识第一个数据 0x3: 命令有效，执行时需要标识第一个数据
7:2	reserved	-	保留
1:0	int_cnt	R	**中断计数器** 中断计数器的值 中断计数器为饱和计数器，其值计到 0x3 就停止增加
0	int_cnt_dec	W	**中断清除端口** 向此位写入一次 1, 将使得 int_cnt 的值减 1 中断计数器为饱和计数器，其值计到 0x0 就停止减少 当中断计数器的值为 0 时，中断被清除

11.2.5 命令队列项 0 的源地址参数 (source0)

偏 移： 0x10
复位值： 32'h0

表 11.6: 命令队列项 0 的源地址参数

位域	名称	访问	描述
31:0	source0	RO	**DMA 命令队列项 0 的源地址参数** DMA 命令队列项 0 中的源地址参数 如果队列 0 项为正在执行的命令项 其值随 DMA 命令的执行而发生变化

11.2.6 命令队列项 1 的源地址参数 (source1)

偏 移： 0x14
复位值： 32'h0

表 11.7: 命令队列项 1 的源地址参数

位域	名称	访问	描述
31:0	source1	RO	**DMA 命令队列项 1 的源地址参数** DMA 命令队列项 1 中的源地址参数 如果队列项 1 为正在执行的命令项 其值随 DMA 命令的执行而发生变化

11.2.7 命令队列项 0 的 DMA 长度参数 (count0)

偏　移：0x18

复位值：32'h0

表 11.8: 命令队列项 0 的 DMA 长度参数

位域	名称	访问	描述
31:0	count0	RO	**DMA 命令队列项 0 的长度参数** DMA 命令队列项 0 中的长度参数 如果队列项 0 为正在执行的命令项 其值随 DMA 命令的执行而发生变化

11.2.8 命令队列项 1 的 DMA 长度参数 (count1)

偏　移：0x1c

复位值：32'h0

表 11.9: 命令队列项 1 的 DMA 长度参数

位域	名称	访问	描述
31:0	count1	RO	**DMA 命令队列项 1 的长度参数** DMA 命令队列项 1 中的长度参数 如果队列项 1 为正在执行的命令项 其值随 DMA 命令的执行而发生变化

11.3 配置流程

11.3.1 典型例子

假设我们要启动一次 DMA 传输，我们需要执行下列初始化操作：

1. 查询 DMA 命令队列是否可写
   - 读寄存器 0x8，直到读到的值的最低位（[0]）为 1
2. 配置 DMA 命令项的参数
   - 向 DMA_SOURCE 写入源地址参数
   - 向 DMA_COUNT 写入传输长度参数
3. 让 DMA 命令参数进入命令队列生效

- 根据 DMA 命令是否需要标识第一个 DMA 数据向 CMD&STATUS 写入 0x3 或者 0x1

此页留空

第十二章　VPWM 模块

12.1　概述

VPWM 模块解码 PCM 音频数据，将采样幅度转换 PWM 占空比信号，驱动扬声器发声。VPWM 可以通过 DMA/PIO 两种方式获取存放在 SPI FLASH 上音频数据。

VPWM 包括 ADPCM 解压缩，线性插值，PWM 解码三个子模块。

输入的音频数据为一个经过 ADPCM 压缩算法压缩的数据，该算法压缩率为 16:4，即将 16bit 的采样数据压缩为 4bit。VPWM 模块中 ADPCM 解压缩子模块负责将 4bit 的采样数据解压缩为 16bit 数据。

线性插值模块在每两个采样之间线性插值若干个新的采样数据。该插值数量 N 为软件可配值项。插值 N 个新数据，即将原始音频的采样率提升为 $N+1$ 倍。

PWM 解码模块最终将前述两个子模块处理过后的数据解码为扬声器的输入信号，播放存放在 SPI FLASH 的音频数据。PWM 解码模块实现了两种解码算法，其中算法 1 只需要一个输出信号即可驱动扬声器，算法 2 需要同时使用两个输出信号来驱动扬声器。但其音质与音量根据所选算法而有所区别。

12.2　寄存器定义

表 12.1: VPWM 寄存器列表

名称	偏移	描述
VpwmCfg	0x00	算法配置
WPortSts	0x08	数据写端口状态
WPort	0x0c	数据写端口

12.2.1　算法配置 (VpwmCfg)

偏　移：　0x00
复位值：　32'h00000000

表 12.2: 算法配置

位域	名称	访问	描述
31	reserved	-	保留
30	alg_module	RW	**算法选择** 0 表示 PWM 解码模块使用算法 1，1 表示使用算法 2
29	neg_one	RW	**输出反转** 0 表示输出信号与所选解码算法的输出信号保持相同，1 表示输出信号为解码信号的的取反信号

位域	名称	访问	描述
28	reserved	-	保留
27	frq_cfg_selc	RW	**vpwm 时钟选择** 0 表示 VPWM 模块使用 32M 时钟，1 表示使用 8M 时钟
26	frq_cfg_rst	RW	**vpwm 时钟复位** 1 表示复位，0 表示不复位
25:24	Iterate_Factor	RW	**插值系数** 无符号数，表示每两个采样之间，插值出 N 个新的采样，满足 $N = 2^{Iterate\_Factor}$，配合 Iterate_Enable 与 Interpol_Enable 寄存器一起使用
23:20	Available_Bit	RW	**有效位数** 无符号数，表示最终有效的音频编码位数，通常情况下为 10 位有效位数，即将源数据的高 10 位进行解码播放
19	DMA_Enable	RW	**DMA 使能** 1 表示 VPWM 模块通过 DMA 获取数据，0 表示通过寄存器获取数据
18	Interpol_Enable	RW	**数据插值使能** 1 表示开启数据插值功能，0 表示不开启
17	Iterate_Enable	RW	**数据重复使能** 1 表示开启数据重复功能，0 表示不开启
16	ADPCM_Enable	RW	**ADPCM 压缩使能** 1 表示输入数据为经过 ADPCM 算法压缩数据，0 表示原始 PCM 数据
15:4	Sum_Width	RW	**总信号宽度** 无符号数，每个采样的总信号宽度
3:0	reserved	-	保留

12.2.2 数据写端口状态 (WPortSts)

偏　移：0x08

复位值：32'h00000000

表 12.3: 数据写端口状态

位域	名称	访问	描述
31:2	reserved	-	保留
1	First_Data	RW	**第一个数据** 1 表示下一个发送给 VPWM 模块的音频数据是该段数据的第一个数据，0 表示下一个数据不是第一个数据
0	Ready	R	**读请求** 1 表示 VPWM 模块向处理器发送读数据请求，0 表示没有读请求

12.2.3 数据写端口 (WPort)

偏　移：0x0c

复位值：32'h00000000

位域	名称	访问	描述

表 12.4: 数据写端口

位域	名称	访问	描述
31:0	Audio_Data	RW	**音频数据** 当配置为通过寄存器接口获取数据时，软件将音频数据写入到该寄存器

12.2.4 参数配置公式

总信号宽度 Sum_Width，输入数据采样率 Sample_Rate，插值系数 Iterate_Factor，VPWM 模块时钟主频 Clock_Rate 需要满足下面公式

$$Clock\_Rate = Sum\_Width * Sample\_Rate * (2^{Iterate\_Factor})$$

有效位数 Available_Bit 与总信号宽度 Sum_Width 需要满足

$$Sum\_Width >= 2^{Available\_Bit-1}$$

例如 VPWM 模块时钟主频 Clock_Rate 为 32M，需要播放的音频数据为 6K 采样率，插值系数为 3, 则

$$\begin{array}{rcccccc} Sum\_Width & = & \frac{Clock\_Rate}{Sample\_Rate*(2^{Iterate\_Factor})} & = & \frac{32M}{6K*2^3} & = & 666 \\ Available\_Bit & \leq & (log_2 Sum\_Width)+1 & = & (log_2 666)+1 & = & 10 \end{array}$$

则此时该配置表示，使用 32M 的 VPWM 模块时钟，将 6k 采样率 16bit 编码率的音频源数据，播放为 48k 采样率 10bit 编码率的音乐。

12.3 输入数据与描述

本模块标准输入数据为 16bit PCM 格式的音频数据。PCM 数据编码中，0x0000 与 0xffff 表示静音数据，满足 0xffff = 0x0000 - 1。0x7fff 与 0x8000 表示最大声音，并且满足 0x8000 = 0x7fff + 1。

本模块包含 ADPCM 模块，即上述输入数据可以为经过 ADPCM 算法压缩过的 4bit 数据，也可以为 16bit 的未压缩数据。通过配置寄存器 ADPCM_Enable 进行配置。

此页留空

第十三章　触摸按键控制器

13.1　概述

触摸控制控制器 (TSENSOR) 采用 RC 振荡的方式测量按键的电容变化，主要测量结构如图13.1所示。

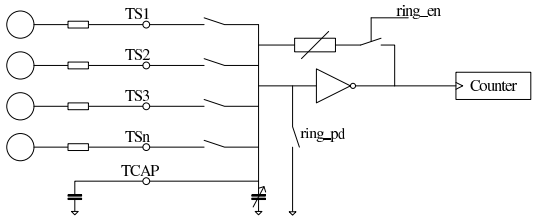

图 13.1: 触摸测量结构

触摸控制控制器可工作在纯硬件模式下，只有检测到按键动作才发出中断唤醒 CPU。其内部主要由时序控制和扫描控制两部分组成。

扫描控制部分负责测量每个按键的电容。开始扫描后，触摸按键传感器将逐个选通，RC 振荡电路使能。在配置好的计数长度内计数时钟周期数。得到 16 个按键值对应的计数，记录到 CntRes 寄存器中。如果配置了热身测量，则在正式测量前加指定数量的热身测量。热身时不选通传感器，测量计数值也会记录到 CntRes 寄存器，并且可能被后续的测量结果所覆盖。

扫描结束后，计数值根据配置进行调整，并排序得到最大值、最小值以及第二小值。判断是否有合理事件发生的条件是最小值与第二小值之间的差大于预设的阈值。判断是否有按键的条件是有最小值与最大值之间的差大于预设的阈值，并且此时有合理事件发生。

时序控制逻辑确定扫描的时机和电源状态。分为待机、激活与去抖三种模式。平时处于待机模式，发现新按键动作进入去抖模式。

去抖模式下，以去抖间隔扫描，如果去抖测量中始终检测到同一键按下，则发中断；否则退出到激活模式。激活模式下，以激活模式周期扫描，如果检测到不同按键则进入去抖模式，如果超过 N 个待机周期未检测到有效按键则退回到待机模式

软件需配合外部电容配置，选择一个恰当的电阻值和计数周期，使得一次测量的计数值足够大 (比如 1000)，但又不会溢出。小的电阻值可以加快计数速度，但最高速度应低于物理实现的限制。如果由于电路实现的原因，按键通道之间存在固定偏差，则应当进行修正。

修正有微调电容和后处理两种方式。通过增加每通道可配置的微调电容，可以直接改变环振频率。后处理修正是把计数值减去通道相关的量。由于计数存在噪声，软件可以在初始化时做多次扫描，如果是未触摸的合理状态，则多个结果平均后得到参考值，然后想办法把通道间偏差补齐。尽量用微调电容修正。值得注意的是，受温漂影响，修正值可能需要周期性地更新。

只读寄存器读出时未经同步。软件应当选择恰当的时机读出，比如停止扫描后。状态寄存器读出时建议连续读，直到两次读出结果相同才使用。

13.2 寄存器定义

表 13.1: 触摸按键模块寄存器列表

名称	偏移	描述
TsCtrl	0x00	控制寄存器
TsStat	0x04	状态寄存器
OscCfg	0x08	环振配置寄存器
PollTim	0x0c	扫描时序寄存器
DiffThres	0x10	差异阈值寄存器
CntMax	0x14	最大计数值
CntMin	0x18	最小计数值
CntLow	0x1c	第二小计数值
CntAdj0 ~15	0x40 ~0x7c	计数修正值 0 ~15
CntRes0 ~15	0x80 ~0xbc	计数结果 0 ~15

13.2.1 控制寄存器 (TsCtrl)

偏　移：　0x00
复位值：　0

表 13.2: 控制寄存器

位域	名称	访问	描述
31:16	scan_mask	RW	**扫描掩码** 为 1 的比特对应的通道被使能，会进行扫描 如果全 0 则只做热身测量，并且结果记录到 CntRes。
15	dbc_en	RW	**去抖使能** 使用专门的去抖间隔扫描
14:12	dbc_num	RW	**去抖计数** 连续检测到几次按键才发中断 0: 关闭 1 ~7: 1 ~7 次
11	eos_ov	RW	**扫描结束强制溢出** 在溢出中断使能时，软件可得知扫描状态。溢出中断状态未清除前循环扫描将暂停

位域	名称	访问	描述
10:8	warmup	RW	**热身测量数** 在真正扫描前进行热身测量的次数 0 ~7: 0 ~7 次
7:4	int_en	RW	**中断使能** [0]: 按下中断使能 [1]: 抬起中断使能 [2]: 激活中断使能 [3]: 溢出中断使能
3	test_en	RW	**测试模式** 使用总线时钟作为环振时钟。可用于测量 32K 时钟与总线时钟的关系。 使用该模式时 CPU 不得休眠
2	adj_en	RW	**计数修正使能** 为 1 时使能
1	poll_en	RW	**循环扫描使能** 为 1 时启动
0	scan_en	RWC	**单次扫描使能** 写 1 启动，完成后清零。循环扫描激活时写 1 可以立即发起一次扫描。 单次扫描不会进行去抖

13.2.2　状态寄存器 (TsStat)

偏　移：　0x04
复位值：　0

表 13.3: 状态寄存器

位域	名称	访问	描述
27:24	scan_state	RO	**扫描状态** [0]: IDLE [1]: SEL [2]: CNT [3]: REC
19:16	poll_state	RO	**循环状态** 0: IDLE 1: ONCE 2: STBW 3: STBR 4: ACTW 5：ACTR 6：DBCW 7：DBCR
7:4	code	RO	**按键编码** 0~15
3	ov	RW1C	**溢出** 测量过程中计数溢出，需要检查是否配置出错。 如果打开溢出中断，发生溢出后循环扫描将暂停，直至中断清除。从计数结果寄存器 CntRes 中可以看到溢出方向

位域	名称	访问	描述
2	trig	RW1C	**激活** 检测到变化，进入激活模式
1	up	RW1C	**抬起** 检测到按键抬起动作。新的按下事件发生时将自动清除该位
0	down	RW1C	**按下** 检测到按键按下动作

13.2.3 环振配置寄存器 (OscCfg)

偏　移：0x08

复位值：0

表 13.4: 环振配置寄存器

位域	名称	访问	描述
16	cnt_neg	RW	**使用双沿计数**
12:8	cnt_prd	RW	**计数长度** 检测某一按键通道时连续计数的时间，以 32k 时钟周期为单位
3:0	rsel	RW	**电阻选择** 0 ~14: 1k ~14k

13.2.4 扫描时序寄存器 (PollTim)

偏　移：0x0c

复位值：0x00080c00

两次扫描开始时刻的间隔为一个扫描周期

表 13.5: 扫描时序寄存器

位域	名称	访问	描述
31:24	dbc_dly	RW	**去抖测量间隔** 前一扫描结束到后一去抖扫描开始的间隔。以 32k 时钟为单位，0 为 256
23:16	stb_prd	RW	**待机模式周期** 以激活模式扫描周期为单位，0 为 256
13:12	act_num	RW	**激活模式扫描时间** 以待机模式周期为单位，0 为 4
11:8	act_prd	RW	**激活模式周期** 以 256 个 32k 时钟为单位，0 为 2^{12}

13.2.5 差异阈值寄存器 (DiffThres)

偏　移：0x10

复位值：0x80400200

表 13.6: 差异阈值寄存器

位域	名称	访问	描述
31:24	press_h	RW	**按键阈值** 未按键时，判断按下的条件是最小值与第大值之间的差大于此域并且按键有效
23:16	press_l	RW	**按键阈值** 有按键后，判断抬起的条件是最小值与第大值之间的差小于此域或者按键无效
15:8	press_d	RW	**按键阈值** 如果最小值与第二小值之间的差大于此域，视作按键有效
7	touch_en	RW	**触摸检测使能** 高有效
6:0	touch	RW	**触摸阈值** 如果最小值与最大值之间的差大于此域，则认为有触摸动作

13.2.6　最大计数 (CntMax)

偏　移：　0x14
复位值：　-
一次扫描得到的最大计数值

表 13.7: 最大计数

位域	名称	访问	描述
19:16	pos	RO	**位置**
11:0	max	RO	**值**

13.2.7　最小计数 (CntMin)

偏　移：　0x18
复位值：　-
一次扫描得到的最小计数值

表 13.8: 最小计数

位域	名称	访问	描述
19:16	pos	RO	**位置**
11:0	min	RO	**值**

13.2.8　第二小计数 (CntLow)

偏　移：　0x1c
复位值：　-

一次扫描得到的第二小计数值

表 13.9: 第二小计数

位域	名称	访问	描述
19:16	pos	RO	**位置**
11:0	low	RO	**值**

13.2.9 修正寄存器 (CntAdj)

偏　移：　0x40 ∼0x7c
复位值：　-

表 13.10: 修正寄存器

位域	名称	访问	描述
23:16	ctune	RW	**电容微调量** 每通道独立的微调电容，1 档约为 5fF
7:0	adj	RW	**后调整量** 在原始计数值基础上减去此域的值，得到用于判断的计数。 8 位无符号数

13.2.10 计数结果寄存器 (CntRes)

偏　移：　0x80 ∼0xbc
复位值：　-

在一个计数周期中出现的上升沿个数

表 13.11: 计数结果寄存器

位域	名称	访问	描述
31	ov	RO	**计数溢出** 为 1 时表明计数值超出 12 位表示范围。如果发生溢出，val 采取饱和计数的方式处理，即下溢记为 0，上溢记为最大值
11:0	val	RO	**计数值** 当计数修正使能时为修正后的计数值，否则为原始值