



# V85XXP 数据手册

Specifications are subject to change without notice.

© 2023 Vango Technologies, Inc.

This document contains information that is proprietary to Vango Technologies, Inc.

Unauthorized reproduction of this information in whole or in part is strictly prohibited.

V85XXP 是一个高度集成的 32 位 MCU 系列产品, 适合于多功能、低功耗的物联网应用。它集成了 Cortex-M0 核, 512KB FLASH, 64KB SRAM, UART/SPI/I<sup>2</sup>C 接口, LCD, WDT 和 RTC。V85XXP 系列支持多种低功耗工作模式, 支持 RTC 电源独立供电。

## 特点

- V85XXP 工作电压: 2.2V~5.5V
- 工作电流:
  - 正常模式: 2.1mA@6.5536MHz
  - IDLE 模式: 0.5mA@6.5536MHz
  - 浅睡眠模式 (5V 供电, LCD 开启): 12μA
  - 浅睡眠模式 (5V 供电, LCD 关闭): 5.4μA
  - 深睡眠 (5V 供电, LCD 关闭): 4.6μA
  - RTC only 模式: 1.3μA
- 封装:
  - LQFP100(V8503P)
  - LQFP100(V8500P)
  - LQFP80(V8530P/V8531P)
  - LQFP64(V8510P)
- 工作温度: -40~+85°C
- 储存温度: -55~+150°C
- MCU
  - 32 位 Cortex-M0 核, 最高速度高达 26.2144MHz
  - 单指令周期乘法
  - 标准两线 SWD 调试接口
  - 内置 512KB FLASH 存储器, 具有写保护和加密功能, 支持 ISP 和 IAP
  - 64K 字节的 SRAM, 带奇偶校验, 在浅睡眠模式下可数据保留
  - 256 字节的 SRAM, 在深睡眠模式下也有数据保留
  - 可实时监测异常, 包括 FLASH 校验和错误, SRAM 奇偶校验错误, 内存地址错误和内存对齐错误
- 外设
  - 最多 6 个 UART 控制器, 具有奇偶校验功能
  - 每个串口发送通道可以配置带红外载波以进行红外传输
  - 最多 2 个 ISO7816 控制器
  - 最多 3 个 SPI 控制器 (主/从机制)
  - 最多 1 个 I<sup>2</sup>C 控制器 (主/从机制)
  - 4 个 32 位定时器
  - 4 个 16 位 PWM 定时器
  - 4 通道 DMA 控制器
  - 128/192/256 位 AES 编解码器
  - ECC 加速算法引擎
  - 双帧缓存 LCD 控制器

- ✓ 4COM/6COM/8COM
- ✓ 1/3 或 1/4 偏置
- ✓ 支持多类型的扫描频率
- ✓ LCD 电压：默认输出电压是 3.3V 左右，2.85~3.6V 可调节，调节步长是 60mV
- 周期可编程的看门狗定时器
- 支持多唤醒源
- 最多 86 个 GPIO
- 最多 16 个 GPIO 外部中断唤醒口
- 模拟控制器
  - 12 位 ADC，具有最高 240Ksps 及最多 8 个外部输入
    - ✓ ADC 支持手动采样模式或自动采样模式
  - 2 路比较器，单端输入或差分输入
  - 内置 32KHz 及 6.5MHz RC 时钟
  - 内置 2 个 PLL
  - 支持外部 32.768KHz 晶体或 6.5536MHz 晶体（可选）
  - 内置 32.768KHz 和 6.5536MHz 晶体的监测电路
  - 每个时钟都可选做为系统时钟
  - 支持数字时钟分频器，最高可达 1/256
  - 支持可编程的低电压监测
  - 支持 AVCC 和 DVCC 的上电复位
  - 支持 RTC 电池供电
  - 支持每个输入的电压监测
- RTC
  - 支持 RTC 电源独立
  - 支持 RTC 1ppm 级别的 RTC 校正

## V85XXP 产品系列

外设	V8503P	V8500P	V8530P	V8531P	V8510P
ADC 外部通道	8	8	7	8	4
Tiny ADC 通道	2	2	2	2	无
UART	6	6	6	6	5
ISO7816	2	2	2	2	2
SPI	3	3	3	3	3
I <sup>2</sup> C	1	1	1	1	1
比较器	2	2	2	2	1
GPIO	86	82	69	71	53
外部 IO 中断和唤醒	16	16	15	16	15
32-bit 定时器	4	4	4	4	4
16-bit PWM 定时器	4	4	4	4	4
PWM 输出和捕获输入	4	4	4	4	4
DMA 通道	4	4	4	4	4
LCD	76x4, 74x6, 72x8	76x4, 74x6, 72x8	63x4, 61x6, 59x88	65x4, 63x6, 61x8	49x4, 47x6, 45x8
FLASH	512KB	512KB	512KB	512KB	512KB
SRAM	64KB	64KB	64KB	64KB	64KB
封装类型	LQFP100	LQFP100	LQFP80	LQFP80	LQFP64

## 声明

杭州万高科技股份有限公司保留对本手册所涉及的产品及相关的技术信息进行补正或更新的权利。使用本手册时，请您从我们的销售渠道或登录公司网站 <http://www.vangotech.com> 获取最新信息。

# 目录

特点 .....	2
V85XXP 产品系列 .....	4
声明 .....	5
目录 .....	6
<b>第 1 章 电气特性 .....</b>	<b>16</b>
1.1 绝对最大额定值 .....	16
1.2 正常工作电压 .....	16
1.3 驱动特性 .....	16
1.4 电源开关导通电阻 .....	16
1.5 功耗 .....	17
1.6 复位和电源控制模块特性 .....	18
1.7 GPIO 特性 .....	19
1.8 ADC 特性 .....	20
1.9 模拟比较器特性 .....	20
1.10 时钟和锁相环特性 .....	20
1.11 FLASH 和 SRAM 特性 .....	21
1.12 晶振的 ESR 特性 .....	21
1.13 时钟输出稳定时间及唤醒时间 .....	22
1.14 TinyADC 转换时间 .....	22
1.15 芯片可靠性分析结果 .....	22
<b>第 2 章 引脚分布图 .....</b>	<b>23</b>
2.1 V8503P 引脚分布图 .....	23
2.2 V8500P 引脚分布图 .....	24
2.3 V8530P 引脚分布图 .....	25
2.4 V8531P 引脚分布图 .....	26
2.5 V8510P 引脚分布图 .....	27
2.6 V85XXP 管脚描述 .....	28
<b>第 3 章 功能框图 .....</b>	<b>35</b>
3.1 系统功能框图 .....	35
3.2 电源功能框图 .....	36
3.3 时钟功能框图 .....	37
<b>第 4 章 内存映射 .....</b>	<b>38</b>
4.1 寄存器地址 .....	38
4.1.1 PMU 寄存器地址 .....	38
4.1.2 ANA 寄存器地址 .....	39
4.1.3 RTC 寄存器地址 .....	41
4.1.4 FLASH 寄存器地址 .....	42
4.1.5 GPIO 寄存器地址 .....	43
4.1.6 DMA 寄存器地址 .....	44
4.1.7 UART 寄存器地址 .....	44
4.1.8 UART32K 寄存器地址 .....	45
4.1.9 ISO7816 寄存器地址 .....	46
4.1.10 TIMER/PWM 寄存器地址 .....	46
4.1.11 LCD 寄存器 .....	48
4.1.12 SPI 寄存器地址 .....	49
4.1.13 I <sup>2</sup> C 寄存器地址 .....	50
4.1.14 MISC 寄存器地址 .....	50
4.1.15 CRYPT 寄存器地址 .....	51
4.1.16 Info 参数寄存器 .....	51
<b>第 5 章 电源系统 .....</b>	<b>60</b>
5.1 寄存器地址 .....	62

5.2	寄存器定义 .....	62
5.2.1	ANA_REG5 寄存器 .....	62
5.2.2	ANA_REG6 寄存器 .....	62
5.2.3	ANA_REG7 寄存器 .....	62
5.2.4	ANA_REG8 寄存器 .....	63
5.2.5	ANA_REG9 寄存器 .....	63
5.2.6	ANA_REGA 寄存器 .....	63
5.2.7	ANA_REGF 寄存器 .....	64
5.2.8	ANA_CTRL 寄存器 .....	64
5.2.9	ANA_CMPOUT 寄存器 .....	64
5.2.10	ANA_INTSTS 寄存器 .....	65
5.2.11	ANA_INTEN 寄存器 .....	65
5.2.12	ANA_CMPCTL 寄存器 .....	66
5.3	AVCC .....	66
5.4	DVCC .....	67
5.5	电源监测 .....	68
5.5.1	VDCIN 电源监测 .....	68
5.5.2	VDD 电源监测 .....	69
5.5.3	AVCC 电源监测 .....	69
5.6	电源切换开关 .....	69
5.6.1	电源切换开关 1 .....	69
5.6.2	电源切换开关 2 .....	69
5.7	应用笔记 .....	70
第 6 章	工作模式 .....	71
6.1	简介 .....	71
6.2	特点 .....	71
6.3	功能框图 .....	72
6.4	寄存器地址 .....	72
6.5	寄存器定义 .....	72
6.5.1	PMU_DSLEEPEN 寄存器 .....	73
6.5.2	PMU_DSLEPPASS 寄存器 .....	73
6.5.3	PMU_CONTROL 寄存器 .....	73
6.5.4	PMU_STS 寄存器 .....	74
6.5.5	ANA_CTRL 寄存器 .....	75
6.5.6	PMU_RAMx 寄存器 .....	75
6.6	复位 .....	76
6.6.1	片外复位 .....	77
6.6.2	看门狗复位 .....	77
6.6.3	上电复位 .....	77
6.6.4	MO 软复位 .....	77
6.6.5	从浅睡眠/深睡眠/IDLE 下唤醒 .....	78
6.7	工作模式的转换 .....	80
6.8	应用笔记 .....	82
6.8.1	外部 IO 唤醒 .....	82
6.8.2	进入深睡眠模式的流程 .....	83
6.8.3	进入浅睡眠的流程 .....	83
6.8.4	进入 IDLE 的流程 .....	84
第 7 章	时钟系统 .....	86
7.1	简介 .....	86
7.2	特点 .....	86
7.3	功能框图 .....	88
7.4	寄存器地址 .....	88
7.5	寄存器定义 .....	89
7.5.1	ANA_REG3 寄存器 .....	89
7.5.2	ANA_REG4 寄存器 .....	89
7.5.3	ANA_REG9 寄存器 .....	89
7.5.4	ANA_REGB 寄存器 .....	90

7.5.5	ANA_REGC 寄存器.....	90
7.5.6	ANA_CMPOUT 寄存器.....	91
7.5.7	PMU_CONTROL 寄存器.....	91
7.5.8	PMU_STS 寄存器.....	91
7.5.9	MISC2_CLKSEL 寄存器.....	92
7.5.10	MISC2_CLKDIVH 寄存器.....	92
7.5.11	MISC2_CLKDIVP 寄存器.....	92
7.5.12	MISC2_HCLKEN 寄存器.....	93
7.5.13	MISC2_PCLKEN 寄存器.....	93
第 8 章	模拟控制器.....	95
8.1	简介.....	95
8.2	特点.....	95
8.3	功能框图.....	95
8.4	寄存器地址.....	95
8.5	寄存器定义.....	98
8.5.1	ANA_REGx 寄存器.....	98
8.5.2	ANA_REG0 寄存器.....	98
8.5.3	ANA_REG1 寄存器.....	98
8.5.4	ANA_REG2 寄存器.....	99
8.5.5	ANA_REG3 寄存器.....	99
8.5.6	ANA_REG4 寄存器.....	100
8.5.7	ANA_REG5 寄存器.....	100
8.5.8	ANA_REG6 寄存器.....	100
8.5.9	ANA_REG7 寄存器.....	101
8.5.10	ANA_REG8 寄存器.....	101
8.5.11	ANA_REG9 寄存器.....	102
8.5.12	ANA_REGA 寄存器.....	102
8.5.13	ANA_REGB 寄存器.....	103
8.5.14	ANA_REGC 寄存器.....	103
8.5.15	ANA_REGD 寄存器.....	103
8.5.16	ANA_REGE 寄存器.....	103
8.5.17	ANA_REGF 寄存器.....	103
8.5.18	ANA_REG10 寄存器.....	104
8.5.19	ANA_REG11 寄存器.....	104
8.5.20	ANA_CTRL 寄存器.....	104
8.5.21	ANA_CMPOUT 寄存器.....	106
8.5.22	ANA_ADCSTATE 寄存器.....	106
8.5.23	ANA_INTSTS 寄存器.....	107
8.5.24	ANA_INTEN 寄存器.....	110
8.5.25	ANA_ADCCTRL0 寄存器.....	112
8.5.26	ANA_ADCDATAx 寄存器.....	112
8.5.27	ANA_CMPCNTx 寄存器.....	113
8.5.28	ANA_MISC 寄存器.....	113
8.5.29	ANA_ADCDOS 寄存器.....	113
8.5.30	ANA_ADCDCPN 寄存器.....	113
8.5.31	ANA_ADCDCNM0 寄存器.....	113
8.5.32	ANA_ADCDATADMA 寄存器.....	114
8.5.33	ANA_ADCCTRL1 寄存器.....	114
8.5.34	ANA_ADCCTRL2 寄存器.....	115
8.5.35	ANA_ADCDATATHD1_0 寄存器.....	116
8.5.36	ANA_ADCDATATHD3_2 寄存器.....	116
8.5.37	ANA_ADCDATATHD_CH 寄存器.....	116
8.5.38	ANA_CMPCTL 寄存器.....	117
8.5.39	ANA_CMPTHR 寄存器.....	119
第 9 章	ADC 控制器.....	121
9.1	简介.....	121

9.2	特点 .....	121
9.3	ADC 功能框图 .....	121
9.4	寄存器地址 .....	122
9.5	寄存器定义 .....	123
9.5.1	ANA_REG0 寄存器 .....	123
9.5.2	ANA_REG1 寄存器 .....	123
9.5.3	ANA_REG3 寄存器 .....	123
9.5.4	ANA_ADCSTATE 寄存器 .....	123
9.5.5	ANA_INTSTS 寄存器 .....	124
9.5.6	ANA_INTEN 寄存器 .....	125
9.5.7	ANA_ADCCTRL0 寄存器 .....	126
9.5.8	ANA_ADCCTRL1 寄存器 .....	127
9.5.9	ANA_ADCCTRL2 寄存器 .....	128
9.5.10	ANA_ADCDATATHD1_0 寄存器 .....	129
9.5.11	ANA_ADCDATATHD3_2 寄存器 .....	130
9.5.12	ANA_ADCDATATHD_CH 寄存器 .....	130
9.5.13	ANA_ADCDATAx 寄存器 .....	131
9.5.14	ANA_ADCDOS 寄存器 .....	132
9.5.15	ANA_ADCDCPN 寄存器 .....	132
9.5.16	ANA_ADCDCNMO 寄存器 .....	132
9.5.17	ANA_ADCDATADMA 寄存器 .....	132
9.5.18	RTC_ADCUCALK 寄存器 .....	132
9.5.19	RTC_ADCMACTL 寄存器 .....	132
9.5.20	RTC_ADCDTCTL 寄存器 .....	133
9.6	ADC 自校准流程 .....	133
9.7	电池电压测量和外部电压测量 .....	135
9.8	温度测量 .....	136
第 10 章	比较器控制器 .....	138
10.1	简介 .....	138
10.2	特点 .....	138
10.3	功能框图 .....	139
10.4	寄存器地址 .....	139
10.5	寄存器定义 .....	140
10.5.1	ANA_REG2 寄存器 .....	140
10.5.2	ANA_REG3 寄存器 .....	140
10.5.3	ANA_REG5 寄存器 .....	141
10.5.4	ANA_REGF 寄存器 .....	141
10.5.5	ANA_CTRL 寄存器 .....	141
10.5.6	ANA_CMPOUT 寄存器 .....	142
10.5.7	ANA_INTSTS 寄存器 .....	142
10.5.8	ANA_INTEN 寄存器 .....	143
10.5.9	ANA_CMPCNTx 寄存器 .....	143
10.5.10	ANA_CMPCTL 寄存器 .....	143
10.5.11	ANA_CMPTHR 寄存器 .....	144
第 11 章	TinyADC 控制器 .....	146
11.1	简介 .....	146
11.2	特点 .....	146
11.3	功能框图 .....	146
11.4	寄存器地址 .....	146
11.5	寄存器定义 .....	146
11.5.1	ANA_REGF 寄存器 .....	147
11.5.2	ANA_CMPOUT 寄存器 .....	147
11.5.3	ANA_INTSTS 寄存器 .....	147
11.5.4	ANA_INTEN 寄存器 .....	148
11.5.5	ANA_MISC 寄存器 .....	148
第 12 章	看门狗 .....	149

12.1	简介 .....	149
12.2	特点 .....	149
12.3	看门狗在不同模式下的状态 .....	149
12.4	寄存器地址 .....	149
12.5	寄存器定义 .....	149
12.5.1	PMU_WDTPASS 寄存器 .....	149
12.5.2	PMU_WDTEN 寄存器 .....	150
12.5.3	PMU_WDTCLR 寄存器 .....	150
第 13 章	RTC 控制器 .....	151
13.1	简介 .....	151
13.2	特点 .....	151
13.3	功能框图 .....	152
13.4	寄存器读写 .....	152
13.4.1	RTC 寄存器写操作 .....	152
13.4.2	RTC 寄存器读操作 .....	153
13.5	寄存器地址 .....	153
13.6	寄存器定义 .....	154
13.6.1	RTC_SEC/MIN/DAY/WEEK/MONTH/YEAR 寄存器 .....	154
13.6.2	RTC_TIME 寄存器 .....	155
13.6.3	RTC_WKUSEC 寄存器 .....	155
13.6.4	RTC_WKUMIN 寄存器 .....	155
13.6.5	RTC_WKUHOUR 寄存器 .....	156
13.6.6	RTC_WKUCNT 寄存器 .....	156
13.6.7	RTC_CAL 寄存器 .....	156
13.6.8	RTC_DIV 寄存器 .....	157
13.6.9	RTC_CTL 寄存器 .....	157
13.6.10	RTC_ITV 寄存器 .....	157
13.6.11	RTC_SITV 寄存器 .....	158
13.6.12	RTC_PWD 寄存器 .....	158
13.6.13	RTC_CE 寄存器 .....	158
13.6.14	RTC_LOAD 寄存器 .....	158
13.6.15	RTC_INTSTS 寄存器 .....	159
13.6.16	RTC_INTEN 寄存器 .....	159
13.6.17	RTC_PSCA 寄存器 .....	160
13.6.18	RTC_ACTI 寄存器 .....	160
13.6.19	RTC_ACF200 寄存器 .....	160
13.6.20	RTC_ACP0 寄存器 .....	161
13.6.21	RTC_ACP1 寄存器 .....	161
13.6.22	RTC_ACP2 寄存器 .....	161
13.6.23	RTC_ACP3 寄存器 .....	161
13.6.24	RTC_ACP4 寄存器 .....	161
13.6.25	RTC_ACP5 寄存器 .....	162
13.6.26	RTC_ACP6 寄存器 .....	162
13.6.27	RTC_ACP7 寄存器 .....	162
13.6.28	RTC_ACKx 寄存器 .....	162
13.6.29	RTC_WKUCNTR 寄存器 .....	163
13.6.30	RTC_ACKTEMP 寄存器 .....	163
13.6.31	RTC_ALARMSEC 寄存器 .....	163
13.6.32	RTC_ALARMMIN 寄存器 .....	163
13.6.33	RTC_ALARMHOUR 寄存器 .....	164
13.6.34	RTC_ALARMCTL 寄存器 .....	164
13.6.35	RTC_ADCUCALK 寄存器 .....	164
13.6.36	RTC_ADCMACTL 寄存器 .....	164
13.6.37	RTC_ADCDCTL 寄存器 .....	165
13.7	Info 参数寄存器(与 RTC 温度补偿有关) .....	165
13.8	校正流程 .....	168
13.8.1	温度校正 .....	168

13.8.2	频率误差校正 .....	168
13.8.3	RTC 手动温度补偿 .....	169
第 14 章	FLASH 控制器 .....	170
14.1	简介 .....	170
14.2	特点 .....	170
14.3	功能框图 .....	171
14.4	寄存器地址 .....	171
14.5	寄存器定义 .....	172
14.5.1	FLASH_PASS 寄存器 .....	172
14.5.2	FLASH_CTRL 寄存器 .....	172
14.5.3	FLASH_PGADDR 寄存器 .....	172
14.5.4	FLASH_PGDATA 寄存器 .....	173
14.5.5	FLASH_PGBx 寄存器 .....	173
14.5.6	FLASH_PGHwX 寄存器 .....	173
14.5.7	FLASH_SERASE 寄存器 .....	174
14.5.8	FLASH_CERASE 寄存器 .....	174
14.5.9	FLASH_DSTB 寄存器 .....	174
14.5.10	FLASH_INTSTS 寄存器 .....	175
14.5.11	FLASH_CSSADDR 寄存器 .....	175
14.5.12	FLASH_CSEADDR 寄存器 .....	175
14.5.13	FLASH_CSVALUE 寄存器 .....	175
14.5.14	FLASH_CSCVALUE 寄存器 .....	175
14.5.15	FLASH_STS 寄存器 .....	176
14.5.16	MISC2_FLASHWC 寄存器 .....	176
14.5.17	FLASH_RDPROT 寄存器 .....	176
14.5.18	FLASH_WRPROT 寄存器 .....	177
14.5.19	FLASH_ICEPROT 寄存器 .....	177
14.6	FLASH 保护说明 .....	177
第 15 章	GPIO 控制器 .....	178
15.1	简介 .....	178
15.2	特点 .....	178
15.3	功能框图 .....	178
15.4	寄存器地址 .....	178
15.5	寄存器定义 .....	179
15.5.1	PMU_IOAOEN 寄存器 .....	180
15.5.2	PMU_IOAIE 寄存器 .....	180
15.5.3	PMU_IOADAT 寄存器 .....	180
15.5.4	PMU_IOAATT 寄存器 .....	180
15.5.5	PMU_IOAWKUEN 寄存器 .....	181
15.5.6	PMU_IOASTS 寄存器 .....	181
15.5.7	PMU_IOAINTSTS 寄存器 .....	182
15.5.8	PMU_IOASEL 寄存器 .....	182
15.5.9	PMU_IOANODEG 寄存器 .....	182
15.5.10	IOX_OEN 寄存器 .....	182
15.5.11	IOX_IE 寄存器 .....	183
15.5.12	IOX_DAT 寄存器 .....	183
15.5.13	IOX_ATT 寄存器 .....	183
15.5.14	IOX_STS 寄存器 .....	183
15.5.15	IOB_SEL 寄存器 .....	184
15.5.16	IOE_SEL 寄存器 .....	184
15.5.17	IO_MISC 寄存器 .....	184
15.6	特殊功能 IO .....	185
15.6.1	特殊功能 IOA .....	185
15.6.2	特殊功能 IOB .....	185
15.6.3	特殊功能 IOC .....	186
15.6.4	特殊功能 IOD .....	187

15.6.5	特殊功能 IOE .....	187
15.6.6	特殊功能 IOF .....	188
第 16 章	DMA 控制器 .....	189
16.1	简介 .....	189
16.2	特点 .....	189
16.3	功能框图 .....	189
16.4	寄存器地址 .....	189
16.5	寄存器定义 .....	190
16.5.1	DMA_IE 寄存器 .....	190
16.5.2	DMA_STS 寄存器 .....	191
16.5.3	DMA_CxCTL 寄存器 .....	192
16.5.4	DMA_CxSRC 寄存器 .....	194
16.5.5	DMA_CxDST 寄存器 .....	194
16.5.6	DMA_CxLEN 寄存器 .....	194
16.5.7	DMA_AESCTL 寄存器 .....	194
16.5.8	DMA_AESKEYx 寄存器 .....	195
第 17 章	UART 控制器 .....	196
17.1	简介 .....	196
17.2	特点 .....	196
17.3	功能框图 .....	196
17.4	寄存器地址 .....	196
17.5	寄存器定义 .....	198
17.5.1	UARTx_DATA 寄存器 .....	198
17.5.2	UARTx_STATE 寄存器 .....	198
17.5.3	UARTx_CTRL 寄存器 .....	199
17.5.4	UARTx_INTSTS 寄存器 .....	199
17.5.5	UARTx_BAUDDIV 寄存器 .....	199
17.5.6	UARTx_CTRL2 寄存器 .....	200
17.5.7	MISC1_IREN 寄存器 .....	200
17.5.8	MISC1_DUTYL 寄存器 .....	200
17.5.9	MISC1_DUTYH 寄存器 .....	200
第 18 章	UART32K 控制器 .....	202
18.1	简介 .....	202
18.2	特点 .....	202
18.3	功能框图 .....	202
18.4	寄存器地址 .....	202
18.5	寄存器定义 .....	203
18.5.1	U32Kx_CTRL0 寄存器 .....	203
18.5.2	U32Kx_CTRL1 寄存器 .....	204
18.5.3	U32Kx_BAUDDIV 寄存器 .....	204
18.5.4	U32Kx_DATA 寄存器 .....	205
18.5.5	U32Kx_STS 寄存器 .....	205
第 19 章	ISO7816 控制器 .....	206
19.1	简介 .....	206
19.2	特点 .....	206
19.3	功能框图 .....	206
19.4	寄存器地址 .....	206
19.5	寄存器定义 .....	207
19.5.1	ISO7816x_BAUDDIVL 寄存器 .....	207
19.5.2	ISO7816x_BAUDDIVH 寄存器 .....	207
19.5.3	ISO7816x_DATA 寄存器 .....	207
19.5.4	ISO7816x_INFO 寄存器 .....	208
19.5.5	ISO7816x_CFG 寄存器 .....	208
19.5.6	ISO7816x_CLK 寄存器 .....	209
第 20 章	Timer/PWM 控制器 .....	211

20.1	简介 .....	211
20.2	特点 .....	211
20.3	功能框图 .....	211
20.4	寄存器地址 .....	212
20.5	寄存器定义 .....	213
20.5.1	TMRx_CTRL 寄存器 .....	214
20.5.2	TMRx_VALUE 寄存器 .....	214
20.5.3	TMRx_RELOAD 寄存器 .....	214
20.5.4	TMRx_INTSTS 寄存器 .....	214
20.5.5	PWMx_CTL 寄存器 .....	214
20.5.6	PWMx_TAR 寄存器 .....	217
20.5.7	PWMx_CCTLx 寄存器 .....	217
20.5.8	PWMx_CCRx 寄存器 .....	219
20.5.9	PWM_O_SEL 寄存器 .....	220
20.5.10	PWM_I_SEL01 寄存器 .....	220
20.5.11	PWM_I_SEL23 寄存器 .....	221
第 21 章	LCD 控制器 .....	223
21.1	简介 .....	223
21.2	特点 .....	223
21.3	功能框图 .....	223
21.4	LCD 时序 .....	224
21.5	LCD 波形电压 .....	224
21.6	LCD 工作电流 .....	224
21.7	LCD 驱动波形 .....	224
21.8	寄存器地址 .....	228
21.9	寄存器定义 .....	230
21.9.1	LCD_FBx 寄存器 .....	230
21.9.2	LCD_CTRL 寄存器 .....	230
21.9.3	LCD_CTRL2 寄存器 .....	231
21.9.4	LCD_SEGCTRL0 寄存器 .....	232
21.9.5	LCD_SEGCTRL1 寄存器 .....	232
21.9.6	LCD_SEGCTRL2 寄存器 .....	233
21.9.7	ANA_REG6 寄存器 .....	233
第 22 章	SPI 控制器 .....	234
22.1	简介 .....	234
22.2	特点 .....	234
22.3	功能框图 .....	234
22.4	寄存器地址 .....	235
22.5	寄存器定义 .....	235
22.5.1	SPIx_CTRL 寄存器 .....	235
22.5.2	SPIx_TXSTS 寄存器 .....	237
22.5.3	SPIx_TXDAT 寄存器 .....	238
22.5.4	SPIx_RXSTS 寄存器 .....	238
22.5.5	SPIx_RXDAT 寄存器 .....	240
22.5.6	SPIx_MISC 寄存器 .....	240
22.6	应用笔记 .....	241
22.6.1	主机模式 .....	241
22.6.2	从机模式 .....	243
22.6.3	连续字节传输 .....	243
第 23 章	I <sup>2</sup> C 控制器 .....	244
23.1	简介 .....	244
23.2	特点 .....	244
23.3	功能框图 .....	244
23.4	寄存器地址 .....	244
23.5	寄存器定义 .....	246
23.5.1	I2C_DATA 寄存器 .....	246

23.5.2	I2C_ADDR 寄存器 .....	246
23.5.3	I2C_CTRL 寄存器 .....	246
23.5.4	I2C_STS 寄存器 .....	247
23.5.5	I2C_CTRL2 寄存器 .....	251
23.6	应用笔记 .....	253
23.6.1	I <sup>2</sup> C 总线协议：启动/停止条件 .....	253
23.6.2	数据传输格式 .....	253
23.6.3	ACK 信号传输 .....	254
23.6.4	读写操作 .....	254
23.6.5	接口时序 .....	255
23.6.6	I <sup>2</sup> C SDA 管脚死锁的解决方法 .....	256
23.7	工作流程 .....	256
23.7.1	主机发送模式 .....	256
23.7.2	主机接收模式 .....	258
23.7.3	从机发送模式 .....	259
23.7.4	从机接收模式 .....	260
第 24 章	中断控制器 .....	261
24.1	简介 .....	261
24.2	中断源 .....	261
第 25 章	MISC 控制器 .....	264
25.1	简介 .....	264
25.2	特点 .....	264
25.3	功能框图 .....	264
25.4	寄存器地址 .....	264
25.5	寄存器定义 .....	265
25.5.1	MISC1_SRAMINT 寄存器 .....	265
25.5.2	MISC1_SRAMINIT 寄存器 .....	265
25.5.3	MISC1_PARERR 寄存器 .....	266
25.5.4	MISC1_IREN 寄存器 .....	266
25.5.5	MISC1_DUTYL 寄存器 .....	266
25.5.6	MISC1_DUTYH 寄存器 .....	266
25.5.7	MISC1_IRQLAT 寄存器 .....	267
25.5.8	MISC1_HIADDR 寄存器 .....	267
25.5.9	MISC1_PIADDR 寄存器 .....	267
25.5.10	MISC2_FLASHWC 寄存器 .....	267
25.5.11	MISC2_CLKSEL 寄存器 .....	268
25.5.12	MISC2_CLKDIVH 寄存器 .....	268
25.5.13	MISC2_CLKDIVP 寄存器 .....	269
25.5.14	MISC2_HCLKEN 寄存器 .....	269
25.5.15	MISC2_PCLKEN 寄存器 .....	269
第 26 章	CRYPT 控制器 .....	271
26.1	简介 .....	271
26.2	特点 .....	271
26.3	功能框图 .....	271
26.4	寄存器地址 .....	271
26.5	寄存器定义 .....	271
26.5.1	CRYPT_CTRL 寄存器 .....	271
26.5.2	CRYPT_PTRA 寄存器 .....	272
26.5.3	CRYPT_PTRB 寄存器 .....	272
26.5.4	CRYPT_PTRO 寄存器 .....	273
26.5.5	CRYPT_CARRY 寄存器 .....	273
26.6	使用时的注意事项 .....	273
26.6.1	数据格式 .....	273
26.6.2	运算细节 .....	274
第 27 章	调试特点 .....	275
27.1	特点 .....	275

27.2	SWD 引脚 .....	275
第 28 章	Cortex-M0 Core 简要描述 .....	276
28.1	CMSIS 函数说明 .....	276
第 29 章	封装尺寸图 .....	277
29.1	V8503P 封装尺寸图 .....	277
29.2	V8500P 封装尺寸图 .....	278
29.3	V8530P/V8531P 封装尺寸图 .....	279
29.4	V8510P 封装尺寸图 .....	280
图索引	.....	281
表索引	.....	283
版本更新说明	.....	293

## 第1章 电气特性

### 1.1 绝对最大额定值

超出下列最大值或低于下列最小值的工作条件可能会造成芯片的永久性损伤。长时间工作在最大值条件下可能会影响芯片的可靠性。

表 1-1 绝对最大额定值

符号	参数	最小	最大	单位
$V_{VDD-VSS}$	外部电源电压	-0.3	+5.8	V
$V_{IN-VSS}$	输入到 GPIO 引脚的外部信号	-0.3	+5.8	V
$S_{VDD}$	IO 上电斜率	5V/s	1V/ $\mu$ s	--
$I_{INJ\_PAD}$	单引脚输入电流	-10	+10	mA
$I_{INJ\_SUM}$	所有引脚输入电流的总和	-50	+40	mA
$T_W$	工作温度范围	-40	+85	$^{\circ}$ C
$T_S$	储存温度范围	-55	+150	$^{\circ}$ C
$T_J$	PN 结结温	-40	+125	$^{\circ}$ C

### 1.2 正常工作电压

表 1-2 供电电压

符号	参数	条件（全温度范围）	最小	典型	最大	单位
$V_{IN.VDD}$	VDD 的输入电压		2.2	3.3/5	5.5	V
$V_{IN.BATRTC}$	BATRTC 的输入电压		2.2	3.3/5	5.5	V

### 1.3 驱动特性

表 1-3 电源管脚的驱动条件

符号	参数	条件（全温度范围）	最小	典型	最大	单位
$I_{DRV.VDD}$	VDD 的驱动能力				40*[1]	mA
$I_{DRV.AVCC}$	AVCC 的驱动能力				30	mA
$I_{DRV.DVCC}$	DVCC 的驱动能力				35	mA
$I_{DRV.AVCCOUT}$	AVCC_OUT 的驱动能力*[2]				20*[2]	mA

\*[1]: VDD 的驱动电流包括外设、AVCC、DVCC 和 AVCC\_OUT 上的负载。

\*[2]: AVCC、AVCC\_OUT 两个管脚同时有驱动的话，驱动电流之和不能超过 30mA。

### 1.4 电源开关导通电阻

表 1-4 电源开关导通电阻

符号	参数	条件（全温度范围）	最小	典型	最大	单位	
$R_{on.BATRTC2RTC}$	BATRTC、RTC 模块之间的开关导通电阻	RTC 模块由 BATRTC 供电	3.3V	325	500	675	ohm
		$I_L=10\mu$ A	5V	250	400	475	ohm

R <sub>on</sub> -VDD2RTC	VDD、RTC 模块之间的开关导通电阻	RTC 模块由 VDD 供电 IL=10μA	3.3V	325	500	675	ohm
			5V	250	400	475	ohm
R <sub>on</sub> -VDD2AVCC	关闭 AVCC 时，VDD、AVCC 之间的开关导通电阻	VDD=3.3V, IL <sup>[1]</sup> <30mA		6.5	7.3	9.0	ohm

\*[1]: AVCC 的负载电流。

## 1.5 功耗

表 1-5 常用工作条件

符号	参数	条件 (常温下测量, AVCC 关闭, VDD=3.3V (特殊说明除外))	最小	典型	最大	单位
I <sub>ACTIVE</sub>	工作电流	系统时钟: 26.2144MHz		6882		μA
		系统时钟: 13.1072MHz		3852		
		系统时钟: 6.5536MHz		2118		
		系统时钟: 3.2768MHz		1175		
		系统时钟: 1.6384MHz		668		
		系统时钟: 819.2kHz		432		
		系统时钟: 409.6kHz		304		
		系统时钟: 204.8kHz		238		
		系统时钟: 32K RC (FLASH deep-standby)		20		
		系统时钟: 32K RC (FLASH 正常)		24		
I <sub>IDLE</sub>	IDLE 电流	系统时钟: 26.2144MHz		1735		μA
		系统时钟: 13.1072MHz		900		
		系统时钟: 6.5536MHz		527		
		系统时钟: 3.2768MHz		343		
		系统时钟: 1.6384MHz		253		
		系统时钟: 819.2kHz		203		
		系统时钟: 409.6kHz		181		
		系统时钟: 204.8kHz		171		
		系统时钟: 32K RC		14		
I <sub>SLP1</sub>	LCD 关闭时的浅睡眠模式电流	RTCCLK 不分频		3.1		μA
		RTCCLK 4 分频		6.3@80 °C		
I <sub>SLP2</sub>	LCD 开启时的浅睡眠模式电流	RTCCLK 不分频		12		μA
		RTCCLK 4 分频		11		
I <sub>DSP</sub>	深睡眠模式电流, VDD=3.3V BATRTC=3.3V	RTCCLK 不分频		2.9		μA
		RTCCLK 4 分频		2.6		
I <sub>RTCONLY</sub>		RTCCLK 不分频		1.3		μA

RTC ONLY 模式电 流, VDD=0V BATRTC=3.3V	RTCCLK 4 分频	1.1		
--	-------------	-----	--	--

表 1-6 各个模块的功耗

模块	电流 ( $\mu\text{A}$ ) (常温下测量)	状态
CMP1/2	0.08	IT_CMP=00 (偏置电流=20nA)
	0.4	IT_CMP=01 (偏置电流=100nA)
	2	IT_CMP=1* (偏置电流=500nA)
BGP	108	
ADC	380	
TinyADC	0.7	
Tempsensor	80	
Input Buffer	60	
Ref Buffer	60	
LCD	5	LCD 驱动电阻=600k
	9.9	LCD 驱动电阻=300k
	14.5	LCD 驱动电阻=200k
	19	LCD 驱动电阻=150k
32768 XTAL	0.6	
32K RC	0.2	
6.5M RC	80	
6.5M XTAL	120	
PLL_L	20	
PLL_H	40	
AVCCLDO	1	

## 1.6 复位和电源控制模块特性

表 1-7 电源和复位控制模块特性

符号	参数	条件 (常温下测量)	最小	典型	最大	单位
$t_{\text{RST}}$	复位消抖时间			20		$\mu\text{s}$
VREF	参考电压 VREF		1.222	1.3	1.352	V
V <sub>PORH</sub>	PORH 监测电压 (AVCCLDO)		1.955	2.08	2.2	V
V <sub>PORL</sub>	PORL 监测电压 (DVCCLDO)		1.222	1.3	1.352	V
V <sub>VDCIN</sub>	VDCIN 监测电平	上升沿			1.062	V
		下降沿	0.868			V
V <sub>VDCINHYS</sub>	VDCIN 迟滞	迟滞		100		mV
V <sub>AVCCLV</sub>	AVCC 低电压监测电平		2.35	2.5	2.6	V
V <sub>VDDALRAM</sub>	VDD 电源监测电压	V <sub>TH</sub> 可配置, V <sub>TH</sub> =2.9V; 参考电压为 VREF	2.726	2.9	3.016	V

AVCC	AVCC 电压		3.2	3.3	3.4	V
V <sub>AVCC_Drop</sub>	AVCC 压降	I <sub>OUT</sub> =30mA			200	mV

表 1-8 电压回差特性

符号	参数	条件 (常温下测量)	最小	典型	最大	单位
V <sub>PORH_HTRES</sub>	PORH 监测电压回差		53.2	66.5	79.8	mV
V <sub>PORL_HTRES</sub>	PORL 监测电压回差		33.6	42.0	50.4	mV
V <sub>VDCIN_HTRES</sub>	VDCIN 监测电压回差		94/188	100/200	106/212	mV
V <sub>AVCCLV_HTRES</sub>	AVCCLV 监测电压回差		33.6	42.0	50.4	mV
V <sub>VDDALRAM_HTRES</sub>	VDD 电源监测电压回差	V <sub>TH_VDDALARM</sub> = 4.5 V	114.0	142.5	171.0	mV
		V <sub>TH_VDDALARM</sub> = 4.2 V	106.4	133.0	159.6	mV
		V <sub>TH_VDDALARM</sub> = 3.9 V	98.8	123.5	148.2	mV
		V <sub>TH_VDDALARM</sub> = 3.6 V	91.2	114.0	136.8	mV
		V <sub>TH_VDDALARM</sub> = 3.2 V	81.1	101.4	121.6	mV
		V <sub>TH_VDDALARM</sub> = 2.9 V	73.5	91.9	110.2	mV
		V <sub>TH_VDDALARM</sub> = 2.6 V	65.9	82.4	98.8	mV
		V <sub>TH_VDDALARM</sub> = 2.3 V	58.3	72.9	87.4	mV

## 1.7 GPIO 特性

表 1-9 GPIO 特性

符号	参数	V <sub>VDD</sub>	条件 (全温度范围)	最小	最大	单位
V <sub>IH</sub>	输入高电平	5V		0.7*V <sub>VDD</sub>		V
		3.3V		2		
V <sub>IL</sub>	输入低电平	5V			0.3*V <sub>VDD</sub>	V
		3.3V				
V <sub>HSYS</sub>	施密特触发迟滞	5V		0.1*V <sub>VDD</sub>		V
		3.3V				
I <sub>IH</sub>	输入高电平时电流	5V			+1	μA
		3.3V				
I <sub>IL</sub>	输入低电平时电流	5V		-1		μA
		3.3V				
V <sub>OH</sub>	输出高电平	5V	驱动电流 11.2mA	V <sub>VDD</sub> -0.8	V <sub>VDD</sub>	V
		3.3V	驱动电流 5.6mA	2.4	V <sub>VDD</sub>	V
V <sub>OL</sub>	输出低电平	5V	驱动电流 11.2mA		0.5	V
		3.3V	驱动电流 5.6mA		0.4	V
C <sub>IN</sub>	输入电容	5V			10	pF
		3.3V				

## 1.8 ADC 特性

表 1-10 ADC 特性

符号	参数	条件	最小	典型	最大	单位
工作电压、基准和功耗						
BGPREF	BGP 电压		1.203	1.22	1.237	V
PSRR	BGP 的电源抑制比			-55		dB
V <sub>ADC</sub>	ADC 工作电压由 AVCC 供电		2.5	3.3	3.6	V
I <sub>ADC</sub>	ADC 工作电流	ADCCLK=6.5MHz		400		μA
ADC 时间参数						
F <sub>mclk</sub>	ADC 工作时钟			6.5		MHz
F <sub>samp</sub>	ADC 采样速率			240		KHz
C <sub>ADC</sub>	内部采样和保持电容			2.6		pF
T <sub>on</sub>	ADC 上电时间		5			T <sub>mclk</sub>
T <sub>CAL</sub>	ADC 校正时间			4096		T <sub>mclk</sub>
ADC 动态特性						
SNDR	信噪失真比	自校准后		70		dB
SFDR	无杂散动态范围			87		dB
THD	总谐波失真			-85		dB
ADC 静态特性						
RES	精度	自校准后		12		bits
ED	差分非线性误差		-1	-0.7/+0.9	1	LSB
EL	积分非线性误差		-2	±0.9	2	LSB
EO	偏移误差			-0.1		
EG	增益误差			-1.5		
ADC 数据刷新时间						
(求平均点数+skip 忽略点数) * (19+8) * ADC 工作时钟周期						

## 1.9 模拟比较器特性

表 1-11 比较器特性

符号	参数	条件 (常温)	最小	典型	最大	单位
V <sub>DDCMP</sub>	比较器工作电压(AVCC)		2.2	3.3	3.6	V
I <sub>COMP</sub>	比较器工作电流	输入偏置电流 20nA, 输入 50kHz 方波		0.08		μA
		输入偏置电流 100nA, 输入 50kHz 方波		0.4		μA
		输入偏置电流 500nA, 输入 50kHz 方波		2		μA
td	传播迟滞 (不加滤波)	输入偏置电流 20nA, 输入 50kHz 方波		1.6		μs
		输入偏置电流 100nA, 输入 50kHz 方波		0.63		μs
		输入偏置电流 500nA, 输入 50kHz 方波		0.27		μs
V <sub>COMPIN</sub>	比较器输入电压范围		0.8		VDD-0.3	V
V <sub>COMPREF</sub>	比较器参考电压	参考电压为 VREF	1.222	1.3	1.352	V
		参考电压为 BGPREF		1.22		V
V <sub>HTRES</sub>	比较器回差		20.0	25.0	30.0	mV

## 1.10 时钟和锁相环特性

表 1-12 时钟和锁相环特性

符号	参数	条件 (全温度范围)	最小	典型	最大	单位
V <sub>VDDPLL</sub>	PLL 工作电压 (DVCC)		1.35	1.5	1.65	V
I <sub>VDDPLL</sub>	PLL 工作电流			30		μA
V <sub>VDDPLLH</sub>	PLLH 工作电压 (DVCC)		1.35	1.5	1.65	V
I <sub>VDDPLLH</sub>	PLLH 工作电流			40		μA
V <sub>VDDRCL</sub>	RCL 工作电压 (AVCC)		2.2	3.3	3.6	V
I <sub>VDDRCL</sub>	RCL 工作电流			0.2		μA
f <sub>RCL</sub>	RCL 频率		29.7	32	35.5	kHz
V <sub>VDDRCH</sub>	RCH 工作电压 (AVCC)		2.2	3.3	3.6	V
I <sub>VDDRCH</sub>	RCH 工作电流			45		μA
f <sub>RCH</sub>	RCH 频率		6.357	6.5	6.75	MHz
V <sub>VDDXOH</sub>	XOH 工作电压 (AVCC)		2.2	3.3	3.6	V
I <sub>VDDXOH</sub>	XOH 工作电流			150		μA
f <sub>XOH</sub>	XOH 频率			6.5536		MHz

## 1.11 FLASH 和 SRAM 特性

表 1-13 FLASH 和 SRAM 特性

参数	条件	最小	典型	最大	单位
FLASH 读一个字的时间		38			ns
FLASH 擦写次数	-40~85°C	20000			time
FLASH 数据保留时间	-40~85°C	20			year
FLASH 字节编程时间		6		7.5	μs
FLASH 扇区擦除时间 (一个扇区有 1024 字节)		4		5	ms
FLASH 芯片全擦除时间		30		40	ms
FLASH 读操作电流	26MHz 访问		2.5	3.5	mA
FLASH 编程操作电流				3.5	mA
FLASH 擦除操作电流				2	mA
FLASH standby 电流			80	150	μA
FLASH deep standby 电流			0.1	6	μA
FLASH 工作电压 (DVCC)		1.35	1.5	1.65	V
SRAM 数据保持电压 (DVCC)		1.35	1.5	1.65	V
SRAM 工作温度		-40	+25	+85	°C

## 1.12 晶振的 ESR 特性

表 1-14 晶振的 ESR 特性

参数	条件 (全温度范围)	最小	典型	最大	单位
6.5536M 晶振的 ESR <sup>[1]</sup>				100	Ω

32768K 晶振的 ESR				50	KΩ
----------------	--	--	--	----	----

\*[1]: ESR(Equivalent series resistance): 等效串联电阻

## 1.13 时钟输出稳定时间及唤醒时间

表 1-15 时钟输出稳定时间及唤醒时间

参数	条件 (全温度范围)	最小	典型	最大	单位
PLLL 锁定时间				1	ms
PLLH 锁定时间				15	μs
RCL 输出稳定时间				200	μs
RCH 输出稳定时间				5	μs
BGP 输出稳定时间				100	μs
RCH 为系统时钟时的浅睡眠唤醒时间			18.4		μs
PLLL 为系统时钟时的浅睡眠唤醒时间			1.03		ms
PLLH 为系统时钟时的浅睡眠唤醒时间			22.8		μs
RCH 为系统时钟时的 IDLE 中断响应时间			6		μs
PLLL 为系统时钟时的 IDLE 中断响应时间			1.6		μs
PLLH 为系统时钟时的 IDLE 中断响应时间			1.6		μs

## 1.14 TinyADC 转换时间

表 1-16 TinyADC 转换时间

参数	条件	最小	典型	最大	单位
TinyADC 转换时间			40		μs

## 1.15 芯片可靠性分析结果

表 1-17 芯片可靠性分析结果

项目	规范	最大值
HBM	Mil-Std-883J Method 3015.9	±4000V
CDM	JS-002-2018	±2000V
LATCH-UP	JEDEC EIA/JESD78E	±200mA

## 第2章 引脚分布图

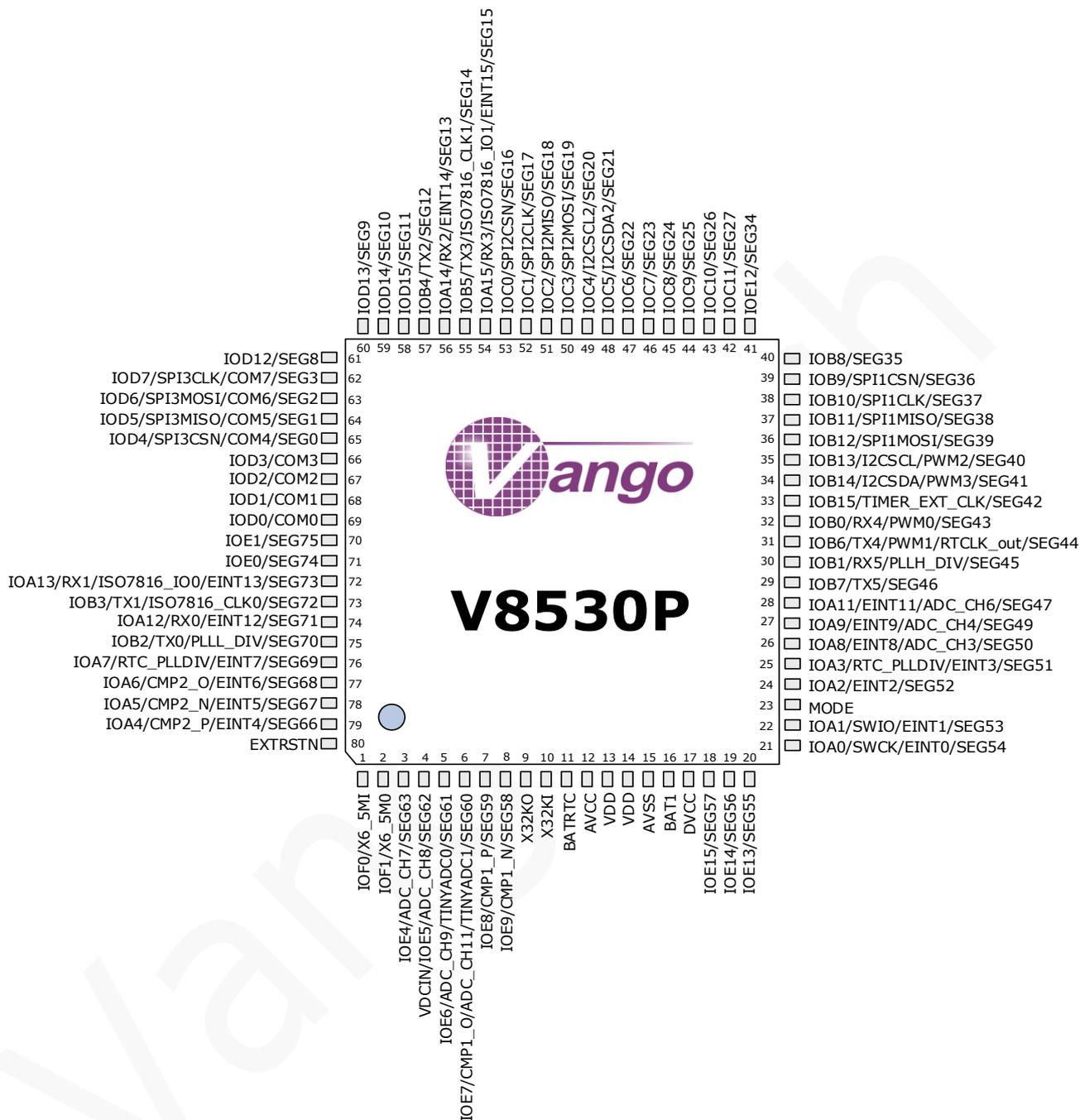
### 2.1 V8503P 引脚分布图



## 2.2 V8500P 引脚分布图



### 2.3 V8530P 引脚分布图



## 2.4 V8531P 引脚分布图



## 2.5 V8510P 引脚分布图



## 2.6 V85XXP 管脚描述

表 2-1 V85XXP 管脚描述

编号	引脚编号					引脚名称	类型	描述
	V8503P	V8500P	V8530P	V8510P	V8531P			
1	1	2	2		2	IOF1	I/O	默认: IOF1 功能 1: 6.5536M 晶体输出
2	2	3	3		3	IOE4	I/O	默认: IOE4 功能 1: ADC_CH7 输入 功能 2: SEG63
3	3	4	4	2	4	IOE5	I/O	默认: VDCIN 输入 功能 1: ADC_CH8 输入 功能 2: IOE5 功能 3: SEG62
4	4	5	5		5	IOE6	I/O	默认: IOE6 功能 1: ADC_CH9 输入和 TinyADC 通道 0 输入 功能 2: SEG61
5	5	6	6		6	IOE7	I/O	默认: IOE7 功能 1: 比较器 1 输出 功能 2: ADC_CH11 输入和 TinyADC 通道 1 输入 功能 3: SEG60
6	6	7	7	3	7	IOE8	I/O	默认: IOE8 功能 1: 比较器 1 P 端输入 功能 2: SEG59
7	7	8	8		8	IOE9	I/O	默认: IOE9 功能 1: 比较器 1 N 端输入 功能 2: SEG58
8	8					IOF7	I/O	默认: IOF7
9	9					IOF11	I/O	默认: IOF11
10	10					IOF12	I/O	默认: IOF12
11	11	9				NC		
12	12	10	9	4	9	X32KO	O	32768Hz 晶体输出引脚, 需要外加电容到地
13	13	11	10	5	10	X32KI	I	32768Hz 晶体输入引脚, 需要外加电容到地
14	14	12	11	6	11	BATRTC	P	RTC 电池输入引脚。
15	15	13	12	7	12	AVCC	O/P	内部模拟电源输出, 用户应该在此引脚连接一个 0.1uF 和一个 10uF 的去耦电容。
16	16	14	13	8	13	VDD	P	主电源输入引脚。用户应该在此引脚连接一个 0.1uF 的去耦电容。
17		15	14	9		VDD	P	主电源输入引脚。用户应该在此引脚连接一个 0.1uF 的去耦电容。

18	17	16	15	10	14	AVSS	G	模拟地
19	18	17	16	11		BAT1	P	BAT1, 电池检测输入引脚。如果需要兼容 V85XXP 系列, 该管腿不接电池, 可以直连 VDD。
20		18				AVCC_OUT	O	默认: 高阻 功能 1: AVCC 电压输出引脚, 该电平与 AVCC 管腿保持一致。用户可以用来驱动小功率模块。
21	19	19	17	12	15	DVCC	O/P	内部数字电源输出引脚, 用户应该在此引脚连接一个 0.1uF 和一个 10uF 的去耦电容。
22	20	20	18	13	16	IOE15	I/O	默认: IOE15 功能 1: SEG57
23	21	21	19	14	17	IOE14	I/O	默认: IOE14 功能 1: SEG56
24	22	22	20	15	18	IOE13	I/O	默认: IOE13 功能 1: SEG55
25	23	23	21	16	19	IOA0	I/O	默认: IOA0(MODE=1), SWCK(MODE=0) 功能 1: EINT0 功能 2: SEG54
26	24	24	22	17	20	IOA1	I/O	默认: IOA1(MODE=1), SWIO(MODE=0) 功能 1: EINT1 功能 2: SEG53
27	25	25	23	18	21	MODE	I	调试模式或非调试模式选择。 0: 调试模式。 1: 非调试模式。 这个引脚应该使用 VDD 上拉, 正常或调试操作过程中该 IO 的状态不应该改变。
28	26	26	24	19	22	IOA2	I/O	默认: IOA2 功能 1: EINT2 功能 2: SEG52
29	27	27	25	20	23	IOA3	I/O	默认: IOA3 功能 1: 秒脉冲输出(RTC_PLLDIV 输出) 功能 2: EINT3 功能 3: SEG51
30	28	28	26	21	24	IOA8	I/O	默认: IOA8 功能 1: EINT8 功能 2: ADC_CH3 输入 功能 3: SEG50
31	29	29	27	22	25	IOA9	I/O	默认: IOA9 功能 1: EINT9 功能 2: ADC_CH4 输入 功能 3: SEG49
32	30	30		23	26	IOA10	I/O	默认: IOA10

								功能 1: EINT10 功能 2: ADC_CH5 输入 功能 3: SEG48
33	31	31	28		27	IOA11	I/O	默认: IOA11 功能 1: EINT11 功能 2: ADC_CH6 输入 功能 3: SEG47
34	32	32	29		28	IOB7	I/O	默认: IOB7 功能 1: UART5_TX 功能 2: SEG46
35	33	33	30	24	29	IOB1	I/O	默认: IOB1 功能 1: UART5_RX 功能 2: PLLH 分频器输出 功能 3: SEG45
36	34	34	31	25	30	IOB6	I/O	默认: IOB6 功能 1: UART4_TX 功能 2: PWM 输出或捕获输入 1 功能 3: RTCCLK 输出 功能 4: SEG44
37	35	35	32	26	31	IOB0	I/O	默认: IOB0 功能 1: UART4_RX 功能 2: PWM 输出或捕获输入 0 功能 3: SEG43
38	36	36	33	27	32	IOB15	I/O	默认: IOB15 功能 1: 定时器外部时钟输入。 功能 2: SEG42
39	37	37	34	28	33	IOB14	I/O	默认: IOB14 功能 1: I2CSDA1 功能 2: PWM 输出或捕获输入 3 功能 3: SEG41
40	38	38	35	29	34	IOB13	I/O	默认: IOB13 功能 1: I2CSCL1 功能 2: PWM 输出或捕获输入 2 功能 3: SEG40
41	39	39	36	30	35	IOB12	I/O	默认: IOB12 功能 1: SPI1MOSI 功能 2: SEG39
42	40	40	37	31	36	IOB11	I/O	默认: IOB11 功能 1: SPI1MISO 功能 2: SEG38
43		41				AVSS	G	模拟地
44	41	42	38	32	37	IOB10	I/O	默认: IOB10

								功能 1: SPI1CLK 功能 2: SEG37
45	42	43	39	33	38	IOB9	I/O	默认: IOB9 功能 1: SPI1CSN 功能 2: SEG36
46	43	44	40	34	39	IOB8	I/O	默认: IOB8 功能 1: SEG35
47	44	45	41		40	IOE12	I/O	默认: IOE12 功能 1: SEG34
48	45	46			41	IOE11	I/O	默认: IOE11 功能 1: SEG33
49	46	47				IOE10	I/O	默认: IOE10 功能 1: SEG32
50	47	48				IOC15	I/O	默认: IOC15 功能 1: SEG31
51	48	49				IOC14	I/O	默认: IOC14 功能 1: SEG30
52	49					AVSS	G	模拟地
53	50	50				IOC13	I/O	默认: IOC13 功能 1: SEG29
54	51	51				IOC12	I/O	默认: IOC12 功能 1: SEG28
55	52	52	42		42	IOC11	I/O	默认: IOC11 功能 1: SEG27
56	53	53	43		43	IOC10	I/O	默认: IOC10 功能 1: SEG26
57	54	54	44		44	IOC9	I/O	默认: IOC9 功能 1: SEG25
58	55	55	45		45	IOC8	I/O	默认: IOC8 功能 1: SEG24
59	56	56	46		46	IOC7	I/O	默认: IOC7 功能 1: SEG23
60	57	57	47		47	IOC6	I/O	默认: IOC6 功能 1: SEG22
61	58	58	48	35	48	IOC5	I/O	默认: IOC5 功能 1: I2CSDA2 功能 2: SEG21
62		59				NC		
63	59	60	49	36	49	IOC4	I/O	默认: IOC4 功能 1: I2CSCL2 功能 2: SEG20
64	60	61	50	37	50	IOC3	I/O	默认: IOC3

								功能 1: SPI2MOSI 功能 2: SEG19
65	61	62	51	38	51	IOC2	I/O	默认: IOC2 功能 1: SPI2MISO 功能 2: SEG18
66	62	63	52	39	52	IOC1	I/O	默认: IOC1 功能 1: SPI2CLK 功能 2: SEG17
67	63	64	53	40	53	IOC0	I/O	默认: IOC0 功能 1: SPI2CSN 功能 2: SEG16
68	64	65	54	41	54	IOA15	I/O	默认: IOA15 功能 1: UART3_RX 功能 2: ISO7816IO1 功能 3: EINT15 功能 4: SEG15
69	65	66	55	42	55	IOB5	I/O	默认: IOB5 功能 1: UART3_TX 功能 2: ISO7816CLK1 功能 3: SEG14
70		67				AVSS	G	模拟地
71	66	68	56	43	56	IOA14	I/O	默认: IOA14 功能 1: UART2_RX 功能 2: EINT14 功能 3: SEG13
72	67	69	57	44	57	IOB4	I/O	默认: IOB4 功能 1: UART2_TX 功能 2: SEG12
73	68	70	58	45	58	IOD15	I/O	默认: IOD15 功能 1: SEG11
74	69	71	59	46	59	IOD14	I/O	默认: IOD14 功能 1: SEG10
75	70	72	60	47	60	IOD13	I/O	默认: IOD13 功能 1: SEG9
76	71	73	61	48	61	IOD12	I/O	默认: IOD12 功能 1: SEG8
77	72	74				IOD11	I/O	默认: IOD11 功能 1: SEG7
78		75				VDDCAP	P	主电源去耦引脚, 推荐外接 0.1uF 电容。
79	73					IOF8	I/O	默认: IOF8
80	74	76				IOD10	I/O	默认: IOD10 功能 1: SEG6

81	75	77				IOD9	I/O	默认: IOD9 功能 1: SEG5
82	76	78				IOD8	I/O	默认: IOD8 功能 1: SEG4
83						AVCC_OUT	P	默认: AVCC_OUT
84	78	79	62	49	62	IOD7	I/O	默认: IOD7 功能 1: COM7/SEG3 功能 2: SPI3CLK
85	79					AVSS	G	模拟地
86	80	80	63	50	63	IOD6	I/O	默认: IOD6 功能 1: COM6/SEG2 功能 2: SPI3MOSI
87	81	81	64	51	64	IOD5	I/O	默认: IOD5 功能 1: COM5/SEG1 功能 2: SPI3MISO
88	82	82	65	52	65	IOD4	I/O	默认: IOD4 功能 1: COM4/SEG0 功能 2: SPI3CSN
89	83	83	66	53	66	IOD3	I/O	默认: IOD3 功能 1: COM3
90	84	84	67	54	67	IOD2	I/O	默认: IOD2 功能 1: COM2
91	85	85	68	55	68	IOD1	I/O	默认: IOD1 功能 1: COM1
92	86	86	69	56	69	IOD0	I/O	默认: IOD0 功能 1: COM0
93	87	87				IOE3	I/O	默认: IOE3 功能 1: SEG77
94		88				NC		
95	88	89				IOE2	I/O	默认: IOE2 功能 1: SEG76
96	89	90	70		70	IOE1	I/O	默认: IOE1 功能 1: SEG75
97	90	91	71		71	IOE0	I/O	默认: IOE0 功能 1: SEG74
98	91	92	72	57	72	IOA13	I/O	默认: IOA13 功能 1: UART1_RX 功能 2: ISO7816I00 功能 3: EINT13 功能 4: SEG73
99	92	93	73	58	73	IOB3	I/O	默认: IOB3

								功能 1: UART1_TX 功能 2: ISO7816CLK0 功能 3: SEG72
100	93	94	74	59	74	IOA12	I/O	默认: IOA12 功能 1: UART0_RX 功能 2: EINT12 功能 3: SEG71
101	94	95	75	60	75	IOB2	I/O	默认: IOB2 功能 1: UART0_TX 功能 2: SEG70
102	95	96	76	61	76	IOA7	I/O	默认: IOA7 功能 1: 秒脉冲输出(RTC_PLLDIV 输出) 功能 2: EINT7 功能 3: SEG69
103	96	97	77	62	77	IOA6	I/O	默认: IOA6 功能 1: 比较器 2 输出 功能 2: EINT6 功能 3: SEG68
104	97	98	78	63	78	IOA5	I/O	默认: IOA5 功能 1: 比较器 2 N 端输入 功能 2: EINT5 功能 3: SEG67
105	98	99	79	64	79	IOA4	I/O	默认: IOA4 功能 1: 比较器 2 P 端输入 功能 2: EINT4 功能 3: SEG66
106	99	100	80	1	80	EXTRST N	I	外部复位引脚, 低电平有效, RC 滤波电路推荐使用 510K 欧电阻和 0.1uF 电容。
107	100	1	1		1	IOF0	I/O	默认: IOF0 功能 1: 6.5536M 晶体输入

## 第3章 功能框图

### 3.1 系统功能框图

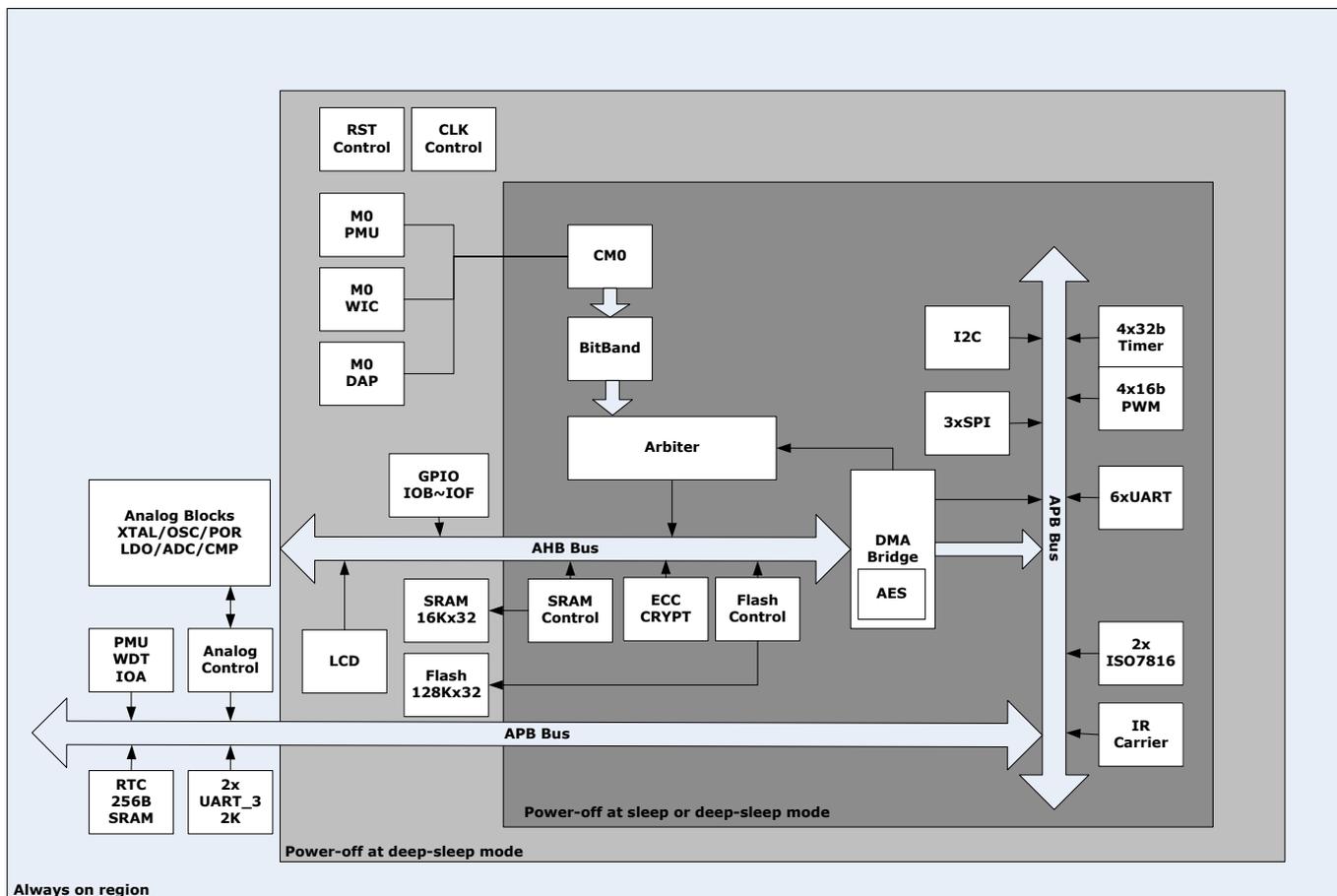


图 3-1 V85XXP 系统功能框图

### 3.2 电源功能框图

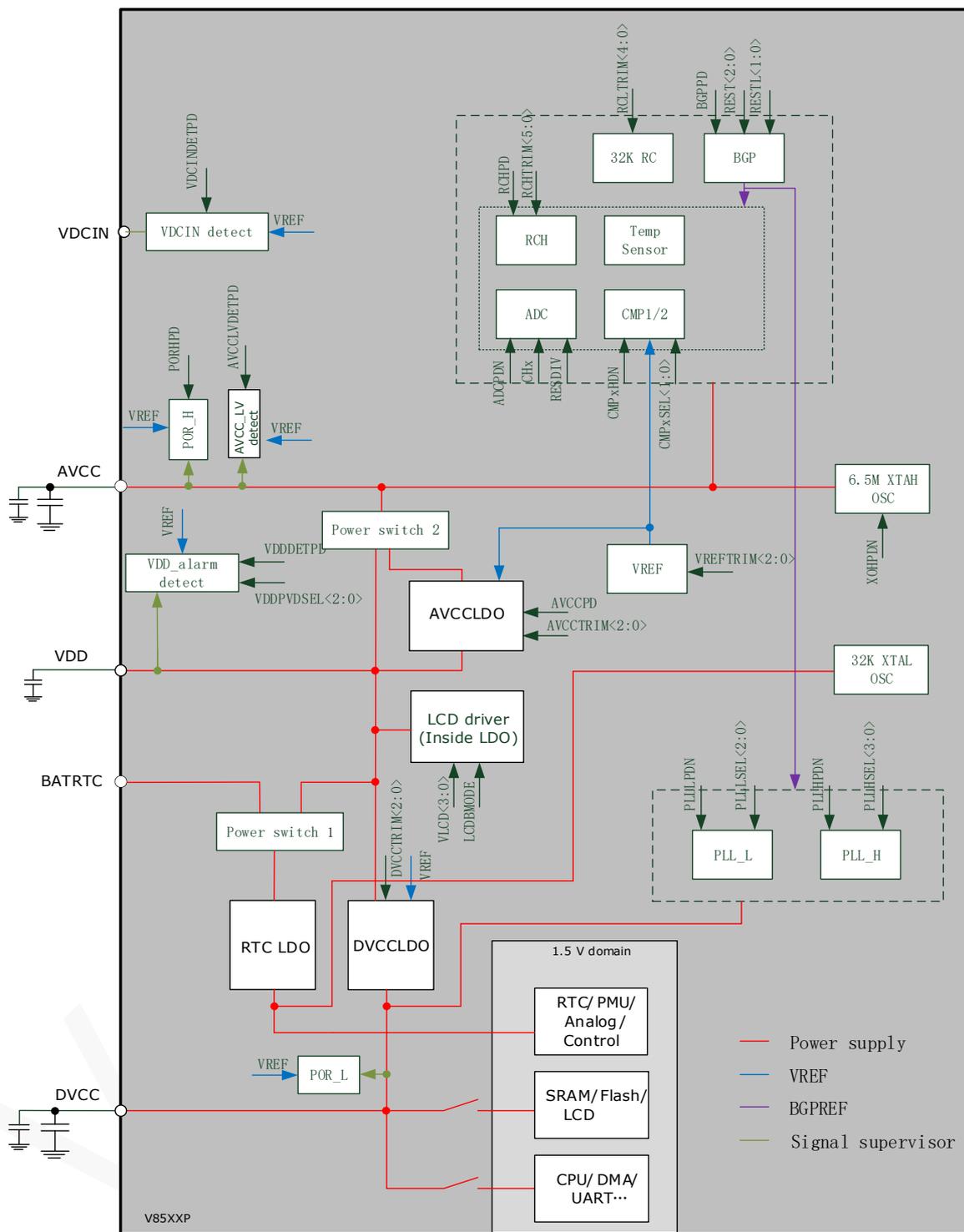


图 3-2 V85XXP 电源系统功能框图

### 3.3 时钟功能框图

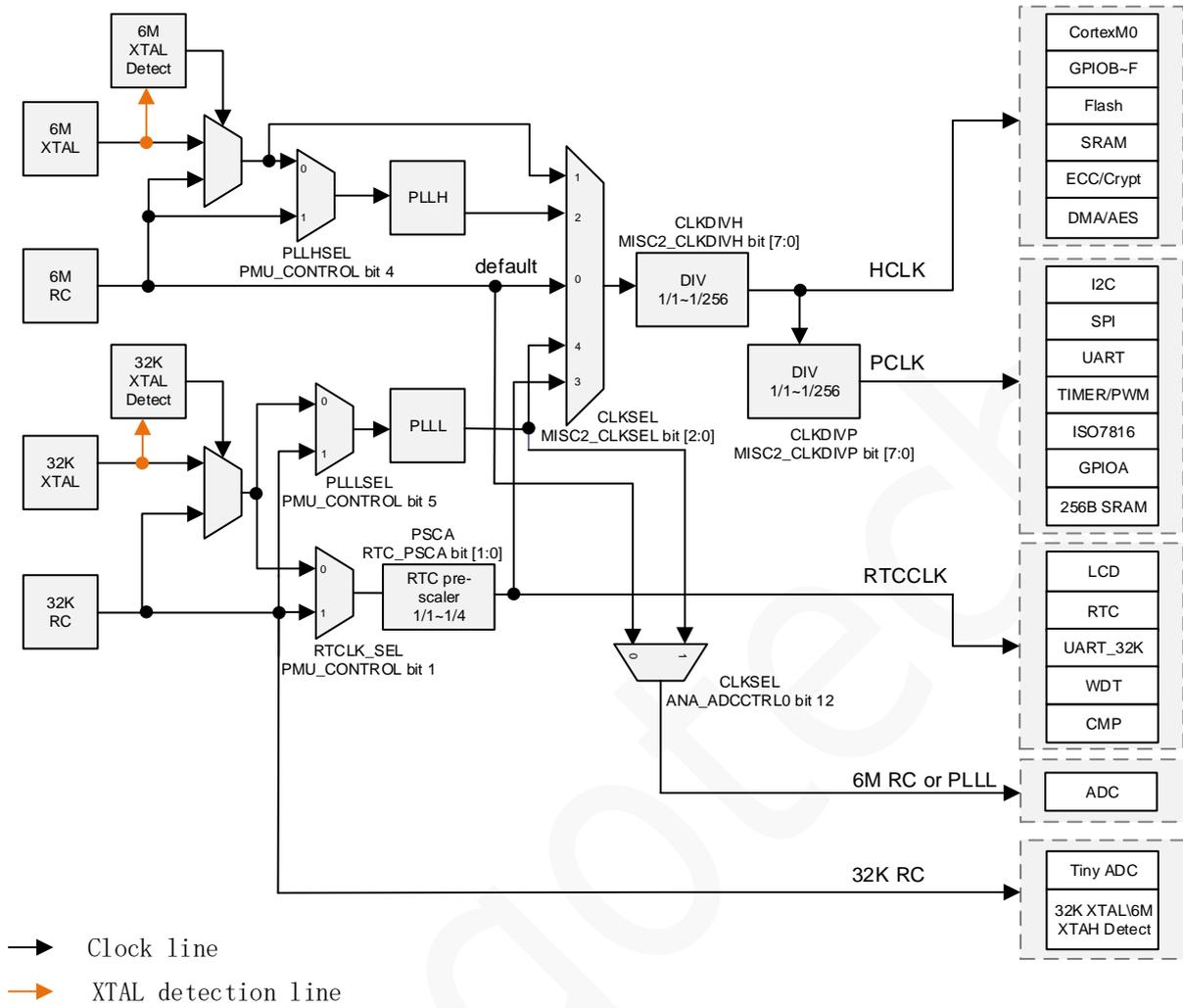


图 3-3 V85XXP 时钟功能框图

## 第4章 内存映射

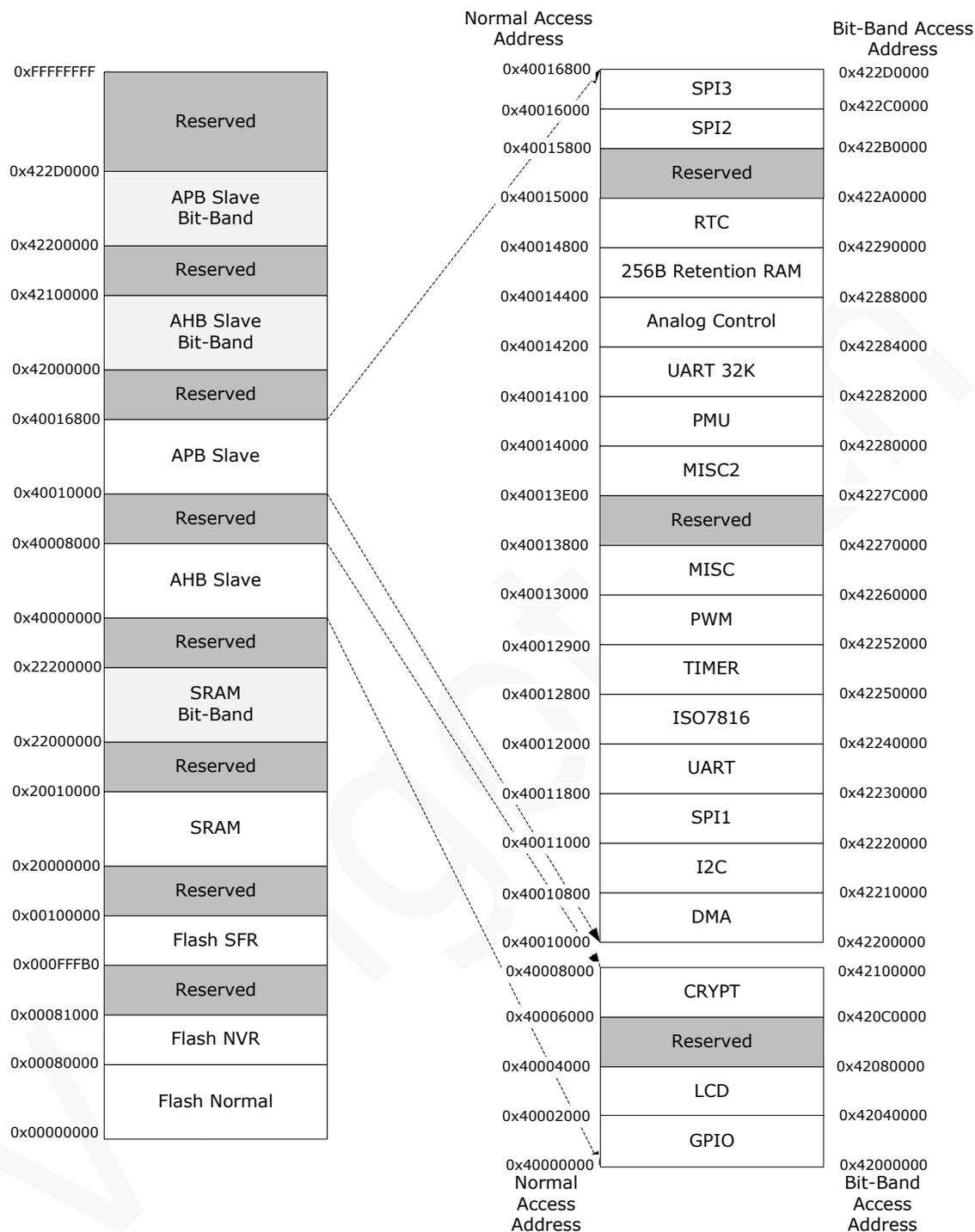


图 4-1 V85XXP 内存映射

Bit\_band 区域仅能由 Cortex-M0 访问，而 DMA 控制器不能。除此以外的其他区域，Cortex-M0 和 DMA 控制器都可以访问。

### 4.1 寄存器地址

#### 4.1.1 PMU 寄存器地址

表 4-1 PMU 寄存器地址(PMU 基地址: 0x40014000)

名称	类型	地址	描述	默认值
PMU_DSLEEPEN	R/W	0x0000	PMU 深睡眠使能寄存器	0x00000000
PMU_DSLEEPPASS	R/W	0x0004	PMU 深睡眠密码寄存器	0x00000000
PMU_CONTROL	R/W	0x0008	PMU 控制寄存器	0x0000
PMU_STS	R/C	0x000C	PMU 状态寄存器	0x0000074
PMU_IOAOEN	R/W	0x0010	IOA 输出使能寄存器	0xFFFF
PMU_IOAIE	R/W	0x0014	IOA 输入使能寄存器	0xFFFF
PMU_IOADAT	R/W	0x0018	IOA 输出数据寄存器	0x0000
PMU_IOAATT	R/W	0x001C	IOA 属性寄存器	0x0000
PMU_IOAWKUEN	R/W	0x0020	IOA 唤醒使能寄存器	0x00000000
PMU_IOASTS	R	0x0024	IOA 输入状态寄存器	--
PMU_IOAINT	R/C	0x0028	IOA 中断状态寄存器	0x0000
PMU_IOASEL	R/W	0x0038	IOA 特殊功能选择寄存器	0x0000
PMU_WDTPASS	R/W	0x0040	看门狗解锁寄存器	0x00000000
PMU_WDTEN	R/W	0x0044	看门狗定时器使能寄存器	0x1
PMU_WDTCLR	R/C	0x0048	看门狗定时器清零寄存器	0x0000
PMU_IOANODEG	R/W	0x0050	IOA 去抖动电路控制	0x0000

表 4-2 PMU 寄存器地址(PMU 掉电保持 RAM 基地址: 0x40014400)

名称	类型	地址	描述	默认值
PMU_RAM0	R/W	0x0000	PMU 32 位掉电保持 RAM0	--
PMU_RAM1	R/W	0x0004	PMU 32 位掉电保持 RAM1	--
PMU_RAM2	R/W	0x0008	PMU 32 位掉电保持 RAM2	--
.....			.....	
PMU_RAM63	R/W	0x00FC	PMU 32 位掉电保持 RAM63	--

#### 4.1.2 ANA 寄存器地址

表 4-3 模拟控制寄存器地址(ANA 基地址: 0x40014200)

名称	类型	地址	描述	默认值
ANA_REG0	R/W	0x0000	模拟寄存器 0	0x00
ANA_REG1	R/W	0x0004	模拟寄存器 1	0x00
ANA_REG2	R/W	0x0008	模拟寄存器 2	0x00
ANA_REG3	R/W	0x000C	模拟寄存器 3	0x00
ANA_REG4	R/W	0x0010	模拟寄存器 4	0x00
ANA_REG5	R/W	0x0014	模拟寄存器 5	0x00
ANA_REG6	R/W	0x0018	模拟寄存器 6	0x00
ANA_REG7	R/W	0x001C	模拟寄存器 7	0x00
ANA_REG8	R/W	0x0020	模拟寄存器 8	0x00
ANA_REG9	R/W	0x0024	模拟寄存器 9	0x00

ANA_REGA	R/W	0x0028	模拟寄存器 10	0x00
ANA_REGB	R/W	0x002C	模拟寄存器 11	来自 FLASH, 用户不得修改
ANA_REGC	R/W	0x0030	模拟寄存器 12	来自 FLASH, 用户不得修改
ANA_REGD	R/W	0x0034	模拟寄存器 13	来自 FLASH, 用户不得修改
ANA_REGE	R/W	0x0038	模拟寄存器 14	来自 FLASH, 用户不得修改
ANA_REGF	R/W	0x003C	模拟寄存器 15	0x00
ANA_REG10	R/W	0x0040	模拟寄存器 16	来自 FLASH
ANA_REG11	W	0x0044	模拟寄存器 17	--
ANA_CTRL	R/W	0x0050	模拟控制寄存器	0x0000000
ANA_CMPOUT	R	0x0054	比较器结果寄存器	0x0030
ANA_ADCSTATE	R	0x005C	ADC 状态寄存器	--
ANA_INTSTS	R/C	0x0060	模拟中断状态寄存器	0x0000
ANA_INTEN	R/W	0x0064	模拟中断使能寄存器	0x0000
ANA_ADCCTRL0	R/W	0x0068	ADC 控制寄存器 0	0x00000000
ANA_CMPCTL	R/W	0x006C	比较器控制寄存器	0x00000000
ANA_ADCDATA0	R	0x0070	ADC 通道 0 数据寄存器	--
ANA_ADCDATA1	R	0x0074	ADC 通道 1 数据寄存器	--
ANA_ADCDATA2	R	0x0078	ADC 通道 2 数据寄存器	--
ANA_ADCDATA3	R	0x007C	ADC 通道 3 数据寄存器	--
ANA_ADCDATA4	R	0x0080	ADC 通道 4 数据寄存器	--
ANA_ADCDATA5	R	0x0084	ADC 通道 5 数据寄存器	--
ANA_ADCDATA6	R	0x0088	ADC 通道 6 数据寄存器	--
ANA_ADCDATA7	R	0x008C	ADC 通道 7 数据寄存器	--
ANA_ADCDATA8	R	0x0090	ADC 通道 8 数据寄存器	--
ANA_ADCDATA9	R	0x0094	ADC 通道 9 数据寄存器	--
ANA_ADCDATA10	R	0x0098	ADC 通道 10 数据寄存器	--
ANA_ADCDATA11	R	0x009C	ADC 通道 11 数据寄存器	--
ANA_ADCDATA12	R	0x00A0	ADC 通道 12 数据寄存器	--
ANA_ADCDATA13	R	0x00A4	ADC 通道 13 数据寄存器	--
ANA_ADCDATA14	R	0x00A8	ADC 通道 14 数据寄存器	--
ANA_ADCDATA15	R	0x00AC	ADC 通道 15 数据寄存器	--
ANA_CMPCNT1	R/C	0x00B0	比较器 1 计数器	0x00000000
ANA_CMPCNT2	R/C	0x00B4	比较器 2 计数器	0x00000000
ANA_MISC	R/W	0x00B8	模拟 misc 控制寄存器	0x00
ANA_ADCDOS	R	0x00C0	ADC 自校准寄存器 DOS	0
ANA_ADCDCPN	R	0x00CC	ADC 自校准寄存器 DCPN	0
ANA_ADCDCNM0	R	0x00D8	ADC 自校准寄存器 DCNM0	0
ANA_ADCDATADMA	R/W	0x00E0	DMA 的 ADC 数据源	0x0000

ANA_CMPTHR	R/W	0x00E4	比较器 1/比较器 2 阈值寄存器	0x0000
ANA_ADCCTRL1	R/W	0x00E8	ADC 控制寄存器 1	0x00000000
ANA_ADCCTRL2	R/W	0x00EC	ADC 控制寄存器 2	0x00000000
ANA_ADCDATATHD1_0	R/W	0x00F4	ADC 阈值控制寄存器 0	0x00000000
ANA_ADCDATATHD3_2	R/W	0x00F8	ADC 阈值控制寄存器 1	0x00000000
ANA_ADCDATATHD_CH	R/W	0x00FC	ADC 阈值状态寄存器	0x00000000

### 4.1.3 RTC 寄存器地址

表 4-4 RTC 控制寄存器地址(RTC 基地址: 0x40014800)

名称	类型	地址	描述	默认值	写保护	读保护
RTC_SEC	R/W	0x0000	RTC 秒寄存器	--	√	√
RTC_MIN	R/W	0x0004	RTC 分寄存器	--	√	√
RTC_HOUR	R/W	0x0008	RTC 时寄存器	--	√	√
RTC_DAY	R/W	0x000C	RTC 天寄存器	--	√	√
RTC_WEEK	R/W	0x0010	RTC 周寄存器	--	√	√
RTC_MON	R/W	0x0014	RTC 月寄存器	--	√	√
RTC_YEAR	R/W	0x0018	RTC 年寄存器	--	√	√
RTC_TIME	R/W	0x001C	RTC 亚秒计数器寄存器	--	√	√
RTC_WKUSEC	R/W	0x0020	RTC 秒唤醒寄存器	0x00	√	
RTC_WKUMIN	R/W	0x0024	RTC 分唤醒寄存器	0x00	√	
RTC_WKUHOURL	R/W	0x0028	RTC 时唤醒寄存器	0x00	√	
RTC_WKUCNT	R/W	0x002C	RTC 计数唤醒寄存器	0x00000000	√	
RTC_CAL	R/W	0x0030	RTC 校正寄存器	--	√	
RTC_DIV	R/W	0x0034	RTC_PLLDIV 分频寄存器	0x00000000		
RTC_CTL	R/W	0x0038	RTC_PLLDIV 分频器控制寄存器	0x0		
RTC_ITV	R/W	0x003C	RTC ITV 唤醒寄存器	0x0	√	
RTC_SITV	R/W	0x0040	RTC SITV 唤醒寄存器	0x00	√	
RTC_PWD	R/W	0x0044	RTC 密码控制寄存器	0x00000000		
RTC_CE	R/W	0x0048	RTC 写使能控制寄存器	0x0		
RTC_LOAD	R/W	0x004C	RTC 读使能控制寄存器	--		
RTC_INTSTS	R/W	0x0050	RTC 中断状态控制寄存器	0x000		
RTC_INTEN	R/W	0x0054	RTC 中断使能控制寄存器	0x000		
RTC_PSCA	R/W	0x0058	RTC 时钟预分频控制寄存器	0x0	√	
RTC_ACTI	R/W	0x0084	晶体顶点温度寄存器	0x1800	√	
RTC_ACF200	R/W	0x0088	RTC 200 倍频率寄存器	0x640000	√	
RTC_ACP0	R/W	0x0090	RTC 参数 0 寄存器	0x0000	√	
RTC_ACP1	R/W	0x0094	RTC 参数 1 寄存器	0x0000	√	

RTC_ACP2	R/W	0x0098	RTC 参数 2 寄存器	0x0000	√	
RTC_ACP3	R	0x009C	RTC 参数 3 寄存器	0x0000		
RTC_ACP4	R/W	0x00A0	RTC 参数 4 寄存器	0x0000	√	
RTC_ACP5	R/W	0x00A4	RTC 参数 5 寄存器	0x0000	√	
RTC_ACP6	R/W	0x00A8	RTC 参数 6 寄存器	0x0000	√	
RTC_ACP7	R/W	0x00AC	RTC 参数 7 寄存器	0x0000	√	
RTC_ACK0	R/W	0x00B0	RTC 参数 k0 寄存器	0x0000	√	
RTC_ACK1	R/W	0x00B4	RTC 参数 k1 寄存器	0x0000	√	
RTC_ACK2	R/W	0x00B8	RTC 参数 k2 寄存器	0x0000	√	
RTC_ACK3	R/W	0x00BC	RTC 参数 k3 寄存器	0x0000	√	
RTC_ACK4	R/W	0x00C0	RTC 参数 k4 寄存器	0x0000	√	
RTC_WKUCNTR	R	0x00CC	记录当前 WKUCNT 值	0x000000		
RTC_ACKTEMP	R/W	0x00D0	RTC K 系数温度分段点控制寄存器	0x3C2800EC	√	
RTC_ALARMSEC	R/W	0x00D8	RTC 闹钟功能的秒设置寄存器	0x0000	√	
RTC_ALARMMIN	R/W	0x00DC	RTC 闹钟功能的分钟设置寄存器	0x0000	√	
RTC_ALARMHOUR	R/W	0x00E0	RTC 闹钟功能的小时设置寄存器	0x0000	√	
RTC_ALARMCTL	R/W	0x00E4	RTC 闹钟控制寄存器	0x0000	√	
RTC_ADCUCALK	R/W	0x00E8	ADC Ucal K 系数寄存器	0x546A546A	√	
RTC_ADCMACTL	R/W	0x00EC	ADC 求平均控制寄存器	0x70000000	√	
RTC_ADCDTCTL	R/W	0x00F0	ADC 数据控制	0x80000000	√	

#### 4.1.4 FLASH 寄存器地址

表 4-5 FLASH 控制寄存器地址(FLASH 控制寄存器基地址: 0x00000000)

名称	类型	地址	描述	默认值
FLASH_ICEPROT	R/W	0xFFFFA8	ICE 保护寄存器	--
FLASH_RDPROT	R	0xFFFFAC	Flash 读保护状态寄存器	0x00000000
FLASH_WRPROT	R/W	0xFFFFB0	Flash 写保护控制寄存器	0x00000000
FLASH_STS	R	0xFFFFBC	FLASH 编程状态寄存器	0x00
FLASH_INTSTS	R/C	0xFFFFCC	FLASH 校验和中断状态寄存器	0x0
FLASH_CSSADDR	R/W	0xFFFFD0	FLASH 校验和开始地址寄存器	0x00000
FLASH_CSEADDR	R/W	0xFFFFD4	FLASH 校验和结束地址寄存器	0xFFFFF
FLASH_CSVALUE	R	0xFFFFD8	FLASH 校验和值读取寄存器	--
FLASH_CSCVALUE	R/W	0xFFFFDC	FLASH 校验和比较值设置寄存器	0x00000000
FLASH_PASS	R/W	0xFFFFE0	FLASH 密码寄存器	0x00000000
FLASH_CTRL	R/W	0xFFFFE4	FLASH 控制寄存器	0x0
FLASH_PGADDR	R/W	0xFFFFE8	FLASH 编程地址寄存器	0x00000
FLASH_PGDATA	R/W	0xFFFFEC	FLASH 编程字数据寄存器	--
FLASH_PGB0	R/W	0xFFFFEC	FLASH 编程字节数据寄存器 0	--

FLASH_PGB1	R/W	0xFFFD	FLASH 编程字节数据寄存器 1	--
FLASH_PGB2	R/W	0xFFFE	FLASH 编程字节数据寄存器 2.	--
FLASH_PGB3	R/W	0xFFEF	FLASH 编程字节数据寄存器 3	--
FLASH_PGHWO	R/W	0xFFEC	FLASH 编程半字数据寄存器 0	--
FLASH_PGHW1	R/W	0xFFEE	FLASH 编程半字数据寄存器 1	--
FLASH_SERASE	R/W	0xFFFF4	FLASH 扇区擦除控制寄存器	0x00000000
FLASH_CERASE	R/W	0xFFFF8	FLASH 芯片擦除控制寄存器	0x00000000
FLASH_DSTB	R/W	0xFFFFC	FLASH deep standby 控制寄存器	0x00000000

### 4.1.5 GPIO 寄存器地址

表 4-6 GPIO 控制寄存器地址(GPIO 基地址: 0x40000000)

名称	类型	地址	描述	默认值
IOB_OEN	R/W	0x0020	IOB 输出使能寄存器	0xFFFF
IOB_IE	R/W	0x0024	IOB 输入使能寄存器	0xFFFF
IOB_DAT	R/W	0x0028	IOB 输出数据寄存器	0x0000
IOB_ATT	R/W	0x002C	IOB 属性寄存器	0x0000
IOB_STS	R	0x0030	IOB 输入状态寄存器	--
IOC_OEN	R/W	0x0040	IOC 输出使能寄存器	0xFFFF
IOC_IE	R/W	0x0044	IOC 输入使能寄存器	0xFFFF
IOC_DAT	R/W	0x0048	IOC 输出数据寄存器	0x0000
IOC_ATT	R/W	0x004C	IOC 属性寄存器	0x0000
IOC_STS	R	0x0050	IOC 输入状态寄存器	--
IOD_OEN	R/W	0x0060	IOD 输出使能寄存器	0xFFFF
IOD_IE	R/W	0x0064	IOD 输入使能寄存器	0xFFFF
IOD_DAT	R/W	0x0068	IOD 输出数据寄存器	0x0000
IOD_ATT	R/W	0x006C	IOD 属性寄存器	0x0000
IOD_STS	R	0x0070	IOD 输入状态寄存器	--
IOE_OEN	R/W	0x0080	IOE 输出使能寄存器	0xFFFF
IOE_IE	R/W	0x0084	IOE 输入使能寄存器	0xFFFF
IOE_DAT	R/W	0x0088	IOE 输出数据寄存器	0x0000
IOE_ATT	R/W	0x008C	IOE 属性寄存器	0x0000
IOE_STS	R	0x0090	IOE 输入状态寄存器	--
IOF_OEN	R/W	0x00A0	IOF 输出使能寄存器	0x7FFF
IOF_IE	R/W	0x00A4	IOF 输入使能寄存器	0x7FFF
IOF_DAT	R/W	0x00A8	IOF 输出数据寄存器	0x0
IOF_ATT	R/W	0x00AC	IOF 属性寄存器	0x0
IOF_STS	R	0x00B0	IOF 输入状态寄存器	--
IOB_SEL	R/W	0x00C0	IOB 特殊功能选择寄存器	0x00

IOE_SEL	R/W	0x00CC	IOE 特殊功能选择寄存器	0x00
IO_MISC	R/W	0x00E0	IOmisc 控制寄存器	0x00

#### 4.1.6 DMA 寄存器地址

表 4-7 DMA 控制寄存器地址(DMA 基地址: 0x40010000)

名称	类型	地址	描述	默认值
DMA_IE	R/W	0x0000	DMA 中断使能寄存器	0x0000
DMA_STS	R/W	0x0004	DMA 状态寄存器	0x0000
DMA_C0CTL	R/W	0x0010	DMA 通道 0 控制寄存器	0x00000000
DMA_C0SRC	R/W	0x0014	DMA 通道 0 源地址寄存器	0x00000000
DMA_C0DST	R/W	0x0018	DMA 通道 0 目的地址寄存器	0x00000000
DMA_C0LEN	R	0x001C	DMA 通道 0 传输长度寄存器	0x0000
DMA_C1CTL	R/W	0x0020	DMA 通道 1 控制寄存器	0x00000000
DMA_C1SRC	R/W	0x0024	DMA 通道 1 源寄存器	0x00000000
DMA_C1DST	R/W	0x0028	DMA 通道 1 目的寄存器	0x00000000
DMA_C1LEN	R	0x002C	DMA 通道 1 传输长度寄存器	0x0000
DMA_C2CTL	R/W	0x0030	DMA 通道 2 控制寄存器	0x00000000
DMA_C2SRC	R/W	0x0034	DMA 通道 2 源寄存器	0x00000000
DMA_C2DST	R/W	0x0038	DMA 通道 2 目的寄存器	0x00000000
DMA_C2LEN	R	0x003C	DMA 通道 2 传输长度寄存器	0x0000
DMA_C3CTL	R/W	0x0040	DMA 通道 3 控制寄存器	0x00000000
DMA_C3SRC	R/W	0x0044	DMA 通道 3 源寄存器	0x00000000
DMA_C3DST	R/W	0x0048	DMA 通道 3 目的寄存器	0x00000000
DMA_C3LEN	R	0x004C	DMA 通道 3 传输长度寄存器	0x0000
DMA_AESCTL	R/W	0x0050	DMA AES 控制寄存器	0x00000000
DMA_AESKEY0	R/W	0x0060	DMA AES 密钥 0 寄存器	0x00000000
DMA_AESKEY1	R/W	0x0064	DMA AES 密钥 1 寄存器	0x00000000
DMA_AESKEY2	R/W	0x0068	DMA AES 密钥 2 寄存器	0x00000000
DMA_AESKEY3	R/W	0x006C	DMA AES 密钥 3 寄存器	0x00000000
DMA_AESKEY4	R/W	0x0070	DMA AES 密钥 4 寄存器	0x00000000
DMA_AESKEY5	R/W	0x0074	DMA AES 密钥 5 寄存器	0x00000000
DMA_AESKEY6	R/W	0x0078	DMA AES 密钥 6 寄存器	0x00000000
DMA_AESKEY7	R/W	0x007C	DMA AES 密钥 7 寄存器	0x00000000

#### 4.1.7 UART 寄存器地址

表 4-8 UART 寄存器地址(UART 基地址: 0x40011800)

名称	类型	地址	描述	默认值
UART0_DATA	R/W	0x0000	UART0 数据寄存器	0x00

UART0_STATE	R/C	0x0004	UART0 状态寄存器	0x00
UART0_CTRL	R/W	0x0008	UART0 控制寄存器	0x000
UART0_INTSTS	R/C	0x000C	UART0 中断状态寄存器	0x00
UART0_BAUDDIV	R/W	0x0010	UART0 波特率分频寄存器	0x00000
UART0_CTRL2	R/W	0x0014	UART0 控制寄存器 2	0x0
UART1_DATA	R/W	0x0020	UART1 数据寄存器	0x00
UART1_STATE	R/C	0x0024	UART1 状态寄存器	0x00
UART1_CTRL	R/W	0x0028	UART1 控制寄存器	0x000
UART1_INTSTS	R/C	0x002C	UART1 中断状态寄存器	0x00
UART1_BAUDDIV	R/W	0x0030	UART1 波特率分频寄存器	0x00000
UART1_CTRL2	R/W	0x0034	UART1 控制寄存器 2	0x0
UART2_DATA	R/W	0x0040	UART2 数据寄存器	0x00
UART2_STATE	R/C	0x0044	UART2 状态寄存器	0x00
UART2_CTRL	R/W	0x0048	UART2 控制寄存器	0x000
UART2_INTSTS	R/C	0x004C	UART2 中断状态寄存器	0x00
UART2_BAUDDIV	R/W	0x0050	UART2 波特率分频寄存器	0x00000
UART2_CTRL2	R/W	0x0054	UART2 控制寄存器 2	0x0
UART3_DATA	R/W	0x0060	UART3 数据寄存器	0x00
UART3_STATE	R/C	0x0064	UART3 状态寄存器	0x00
UART3_CTRL	R/W	0x0068	UART3 控制寄存器	0x000
UART3_INTSTS	R/C	0x006C	UART3 中断状态寄存器	0x00
UART3_BAUDDIV	R/W	0x0070	UART3 波特率分频寄存器	0x00000
UART3_CTRL2	R/W	0x0074	UART3 控制寄存器 2	0x0
UART4_DATA	R/W	0x0080	UART4 数据寄存器	0x00
UART4_STATE	R/C	0x0084	UART4 状态寄存器	0x00
UART4_CTRL	R/W	0x0088	UART4 控制寄存器	0x000
UART4_INTSTS	R/C	0x008C	UART4 中断状态寄存器	0x00
UART4_BAUDDIV	R/W	0x0090	UART4 波特率分频寄存器	0x00000
UART4_CTRL2	R/W	0x0094	UART4 控制寄存器 2	0x0
UART5_DATA	R/W	0x00A0	UART5 数据寄存器	0x00
UART5_STATE	R/C	0x00A4	UART5 状态寄存器	0x00
UART5_CTRL	R/W	0x00A8	UART5 控制寄存器	0x000
UART5_INTSTS	R/C	0x00AC	UART5 中断状态寄存器	0x00
UART5_BAUDDIV	R/W	0x00B0	UART5 波特率分频寄存器	0x00000
UART5_CTRL2	R/W	0x00B4	UART5 控制寄存器 2	0x0

#### 4.1.8 UART32K 寄存器地址

表 4-9 UART32K 寄存器地址(UART32K 基地址: 0x40014100)

名称	类型	地址	描述	默认值
U32K0_CTRL0	R/W	0x0000	UART32K0 控制寄存器 0	0x000
U32K0_CTRL1	R/W	0x0004	UART32K0 控制寄存器 1	0x00
U32K0_BAUDDIV	R/W	0x0008	UART32K0 波特率分频寄存器	0x4B00
U32K0_DATA	R	0x000C	UART32K0 接收数据寄存器	--
U32K0_STS	R/C	0x0010	UART32K0 中断状态寄存器	0x0
U32K1_CTRL0	R/W	0x0080	UART32K1 控制寄存器 0	0x000
U32K1_CTRL1	R/W	0x0084	UART32K1 控制寄存器 1	0x00
U32K1_BAUDDIV	R/W	0x0088	UART32K1 波特率分频寄存器	0x4B00
U32K1_DATA	R	0x008C	UART32K1 接收数据寄存器	--
U32K1_STS	R/C	0x0090	UART32K1 中断状态寄存器	0x0

### 4.1.9 ISO7816 寄存器地址

表 4-10 ISO7816 寄存器地址(ISO7816 基地址: 0x40012000)

名称	类型	地址	描述	默认值
ISO78160_BAUDDIVL	R/W	0x0004	ISO78160 波特率低字节寄存器	0x00
ISO78160_BAUDDIVH	R/W	0x0008	ISO78160 波特率高字节寄存器	0x00
ISO78160_DATA	R/W	0x000C	ISO78160 数据寄存器	0x00
ISO78160_INFO	R/C	0x0010	ISO78160 状态寄存器	0x00
ISO78160_CFG	R/W	0x0014	ISO78160 控制寄存器	0x00
ISO78160_CLK	R/W	0x0018	ISO78160 时钟分频器控制寄存器	0x00
ISO78161_BAUDDIVL	R/W	0x0044	ISO78161 波特率低字节寄存器	0x00
ISO78161_BAUDDIVH	R/W	0x0048	ISO78161 波特率高字节寄存器	0x00
ISO78161_DATA	R/W	0x004C	ISO78161 数据寄存器	0x00
ISO78161_INFO	R/C	0x0050	ISO78161 状态寄存器	0x00
ISO78161_CFG	R/W	0x0054	ISO78161 控制寄存器	0x00
ISO78161_CLK	R/W	0x0058	ISO78161 时钟分频器控制寄存器	0x00

### 4.1.10 TIMER/PWM 寄存器地址

表 4-11 TIMER 寄存器地址(32 位 TIMER 基地址: 0x40012800)

名称	类型	地址	描述	默认值
TMR0_CTRL	R/W	0x0000	Timer0 控制寄存器	0x0
TMR0_VALUE	R/W	0x0004	Timer0 实时计数寄存器	0x00000000
TMR0_RELOAD	R/W	0x0008	Timer0 重载寄存器	0x00000000
TMR0_INTSTS	R/C	0x000C	Timer0 中断状态寄存器	0x0
TMR1_CTRL	R/W	0x0020	Timer1 控制寄存器	0x0
TMR1_VALUE	R/W	0x0024	Timer1 实时计数寄存器	0x00000000

TMR1_RELOAD	R/W	0x0028	Timer1 重载寄存器	0x00000000
TMR1_INTSTS	R/C	0x002C	Timer1 中断状态寄存器	0x0
TMR2_CTRL	R/W	0x0040	Timer2 控制寄存器	0x0
TMR2_VALUE	R/W	0x0044	Timer2 实时计数寄存器	0x00000000
TMR2_RELOAD	R/W	0x0048	Timer2 重载寄存器	0x00000000
TMR2_INTSTS	R/C	0x004C	Timer2 中断状态寄存器	0x0
TMR3_CTRL	R/W	0x0060	Timer3 控制寄存器	0x0
TMR3_VALUE	R/W	0x0064	Timer3 实时计数寄存器	0x00000000
TMR3_RELOAD	R/W	0x0068	Timer3 重载寄存器	0x00000000
TMR3_INTSTS	R/C	0x006C	Timer3 中断状态寄存器	0x0

表 4-12 PWM TIMER 寄存器地址(16 位 PWM TIMER 基地址: 0x40012900)

名称	类型	地址	描述	默认值
PWM0_CTL	R/W	0x0000	PWM Timer0 控制寄存器	0x00
PWM0_TAR	R	0x0004	PWM Timer0 实时计数寄存器	0x0000
PWM0_CCTL0	R/W	0x0008	PWM Timer0 比较/捕获控制寄存器 0	0x000
PWM0_CCTL1	R/W	0x000C	PWM Timer0 比较/捕获控制寄存器 1	0x000
PWM0_CCTL2	R/W	0x0010	PWM Timer0 比较/捕获控制寄存器 2	0x000
PWM0_CCR0	R/W	0x0014	PWM Timer0 比较/捕获数据寄存器 0	0x0000
PWM0_CCR1	R/W	0x0018	PWM Timer0 比较/捕获数据寄存器 1	0x0000
PWM0_CCR2	R/W	0x001C	PWM Timer0 比较/捕获数据寄存器 2	0x0000
PWM1_CTL	R/W	0x0020	PWM Timer1 控制寄存器	0x00
PWM1_TAR	R	0x0024	PWM Timer1 实时计数寄存器	0x0000
PWM1_CCTL0	R/W	0x0028	PWM Timer1 比较/捕获数据寄存器 0	0x000
PWM1_CCTL1	R/W	0x002C	PWM Timer1 比较/捕获数据寄存器 1	0x000
PWM1_CCTL2	R/W	0x0030	PWM Timer1 比较/捕获数据寄存器 2	0x000
PWM1_CCR0	R/W	0x0034	PWM Timer1 比较/捕获数据寄存器 0	0x0000
PWM1_CCR1	R/W	0x0038	PWM Timer1 比较/捕获数据寄存器 1	0x0000
PWM1_CCR2	R/W	0x003C	PWM Timer1 比较/捕获数据寄存器 2	0x0000
PWM2_CTL	R/W	0x0040	PWM Timer2 控制寄存器	0x00
PWM2_TAR	R	0x0044	PWM Timer2 实时计数寄存器	0x0000
PWM2_CCTL0	R/W	0x0048	PWM Timer2 比较/捕获控制寄存器 0	0x000
PWM2_CCTL1	R/W	0x004C	PWM Timer2 比较/捕获控制寄存器 1	0x000
PWM2_CCTL2	R/W	0x0050	PWM Timer2 比较/捕获控制寄存器 2	0x000
PWM2_CCR0	R/W	0x0054	PWM Timer2 比较/捕获数据寄存器 0	0x0000
PWM2_CCR1	R/W	0x0058	PWM Timer2 比较/捕获数据寄存器 1	0x0000

PWM2_CCR2	R/W	0x005C	PWM Timer2 比较/捕获数据寄存器 2	0x0000
PWM3_CTL	R/W	0x0060	PWM Timer3 控制寄存器	0x00
PWM3_TAR	R	0x0064	PWM Timer3 实时计数寄存器	0x0000
PWM3_CCTL0	R/W	0x0068	PWM Timer3 比较/捕获控制寄存器 0	0x000
PWM3_CCTL1	R/W	0x006C	PWM Timer3 比较/捕获控制寄存器 1	0x000
PWM3_CCTL2	R/W	0x0070	PWM Timer3 比较/捕获控制寄存器 2	0x000
PWM3_CCR0	R/W	0x0074	PWM Timer3 比较/捕获数据寄存器 0	0x0000
PWM3_CCR1	R/W	0x0078	PWM Timer3 比较/捕获数据寄存器 1	0x0000
PWM3_CCR2	R/W	0x007C	PWM Timer3 比较/捕获数据寄存器 2	0x0000
PWM_O_SEL	R/W	0x00F0	PWM 输出选择寄存器	0xDB51
PWM_I_SEL01	R/W	0x00F4	PWM0 PWM1 输入选择寄存器	0x00390024
PWM_I_SEL23	R/W	0x00F8	PWM2 PWM3 输入选择寄存器	0x0013000E

#### 4.1.11 LCD 寄存器

表 4-13 LCD 寄存器地址(LCD 基地址: 0x40002000)

名称	类型	地址	描述	默认值
LCD_FB00	R/W	0x0000	LCD 帧缓存 0 寄存器	--
LCD_FB01	R/W	0x0004	LCD 帧缓存 1 寄存器	--
LCD_FB02	R/W	0x0008	LCD 帧缓存 2 寄存器	--
LCD_FB03	R/W	0x000C	LCD 帧缓存 3 寄存器	--
LCD_FB04	R/W	0x0010	LCD 帧缓存 4 寄存器	--
LCD_FB05	R/W	0x0014	LCD 帧缓存 5 寄存器	--
LCD_FB06	R/W	0x0018	LCD 帧缓存 6 寄存器	--
LCD_FB07	R/W	0x001C	LCD 帧缓存 7 寄存器	--
LCD_FB08	R/W	0x0020	LCD 帧缓存 8 寄存器	--
LCD_FB09	R/W	0x0024	LCD 帧缓存 9 寄存器	--
LCD_FB0A	R/W	0x0028	LCD 帧缓存 10 寄存器	--
LCD_FB0B	R/W	0x002C	LCD 帧缓存 11 寄存器	--
LCD_FB0C	R/W	0x0030	LCD 帧缓存 12 寄存器	--
LCD_FB0D	R/W	0x0034	LCD 帧缓存 13 寄存器	--
LCD_FB0E	R/W	0x0038	LCD 帧缓存 14 寄存器	--
LCD_FB0F	R/W	0x003C	LCD 帧缓存 15 寄存器	--
LCD_FB10	R/W	0x0040	LCD 帧缓存 16 寄存器	--
LCD_FB11	R/W	0x0044	LCD 帧缓存 17 寄存器	--
LCD_FB12	R/W	0x0048	LCD 帧缓存 18 寄存器	--

LCD_FB13	R/W	0x004C	LCD 帧缓存 19 寄存器	--
LCD_FB14	R/W	0x0050	LCD 帧缓存 20 寄存器	--
LCD_FB15	R/W	0x0054	LCD 帧缓存 21 寄存器	--
LCD_FB16	R/W	0x0058	LCD 帧缓存 22 寄存器	--
LCD_FB17	R/W	0x005C	LCD 帧缓存 23 寄存器	--
LCD_FB18	R/W	0x0060	LCD 帧缓存 24 寄存器	--
LCD_FB19	R/W	0x0064	LCD 帧缓存 25 寄存器	--
LCD_FB1A	R/W	0x0068	LCD 帧缓存 26 寄存器	--
LCD_FB1B	R/W	0x006C	LCD 帧缓存 27 寄存器	--
LCD_FB1C	R/W	0x0070	LCD 帧缓存 28 寄存器	--
LCD_FB1D	R/W	0x0074	LCD 帧缓存 29 寄存器	--
LCD_FB1E	R/W	0x0078	LCD 帧缓存 30 寄存器	--
LCD_FB1F	R/W	0x007C	LCD 帧缓存 31 寄存器	--
LCD_FB20	R/W	0x0080	LCD 帧缓存 32 寄存器	--
LCD_FB21	R/W	0x0084	LCD 帧缓存 33 寄存器	--
LCD_FB22	R/W	0x0088	LCD 帧缓存 34 寄存器	--
LCD_FB23	R/W	0x008C	LCD 帧缓存 35 寄存器	--
LCD_FB24	R/W	0x0090	LCD 帧缓存 36 寄存器	--
LCD_FB25	R/W	0x0094	LCD 帧缓存 37 寄存器	--
LCD_FB26	R/W	0x0098	LCD 帧缓存 38 寄存器	--
LCD_FB27	R/W	0x009C	LCD 帧缓存 39 寄存器	--
LCD_CTRL	R/W	0x0100	LCD 控制寄存器	0x00
LCD_CTRL2	R/W	0x0104	LCD 控制寄存器 2	0x0000
LCD_SEGCTRL0	R/W	0x0108	LCDSEG 使能控制寄存器 0	0x00000000
LCD_SEGCTRL1	R/W	0x010C	LCDSEG 使能控制寄存器 1	0x00000000
LCD_SEGCTRL2	R/W	0x0110	LCDSEG 使能控制寄存器 2	0x00000000

#### 4.1.12 SPI 寄存器地址

表 4-14 SPI1 寄存器地址(SPI1 基地址: 0x40011000)

名称	类型	地址	描述	默认值
SPI1_CTRL	R/W	0x0000	SPI1 控制寄存器	0x0000
SPI1_TXSTS	R/W	0x0004	SPI1 发送状态寄存器	0x8200
SPI1_TXDAT	R/W	0x0008	SPI1 发送 FIFO 寄存器	--
SPI1_RXSTS	R/W	0x000C	SPI1 接收状态寄存器	0x0000
SPI1_RXDAT	R/W	0x0010	SPI1 接收 FIFO 寄存器	--
SPI1_MISC	R/W	0x0014	SPI1 Misc 控制寄存器	0x0003

表 4-15 SPI2 寄存器地址(SPI2 基地址: 0x40015800)

名称	类型	地址	描述	默认值
SPI2_CTRL	R/W	0x0000	SPI2 控制寄存器	0x0000
SPI2_TXSTS	R/W	0x0004	SPI2 发送状态寄存器	0x8200
SPI2_TXDAT	R/W	0x0008	SPI2 发送 FIFO 寄存器	--
SPI2_RXSTS	R/W	0x000C	SPI2 接收状态寄存器	0x0000
SPI2_RXDAT	R/W	0x0010	SPI2 接收 FIFO 寄存器	--
SPI2_MISC	R/W	0x0014	SPI2 Misc 控制寄存器	0x0003

表 4-16 SPI3 寄存器地址(SPI3 基地址: 0x40016000)

名称	类型	地址	描述	默认值
SPI3_CTRL	R/W	0x0000	SPI3 控制寄存器	0x0000
SPI3_TXSTS	R/W	0x0004	SPI3 发送状态寄存器	0x8200
SPI3_TXDAT	R/W	0x0008	SPI3 发送 FIFO 寄存器	--
SPI3_RXSTS	R/W	0x000C	SPI3 接收状态寄存器	0x0000
SPI3_RXDAT	R/W	0x0010	SPI3 接收 FIFO 寄存器	--
SPI3_MISC	R/W	0x0014	SPI3 Misc 控制寄存器	0x0003

### 4.1.13 I<sup>2</sup>C 寄存器地址

表 4-17 I<sup>2</sup>C 寄存器地址(I<sup>2</sup>C 基地址: 0x40010800)

名称	类型	地址	描述	默认值
I2C_DATA	R/W	0x0000	I <sup>2</sup> C 数据寄存器	0x00
I2C_ADDR	R/W	0x0004	I <sup>2</sup> C 地址寄存器	0x00
I2C_CTRL	R/W	0x0008	I <sup>2</sup> C 控制/状态寄存器	0x00
I2C_STS	R/W	0x000C	I <sup>2</sup> C 状态寄存器	0xF8
I2C_CTRL2	R/W	0x0018	I <sup>2</sup> C 中断使能寄存器	0x0

### 4.1.14 MISC 寄存器地址

表 4-18 MISC 寄存器地址(MISC 基地址: 0x40013000)

名称	类型	地址	描述	默认值
MISC1_SRAMINT	R/C	0x0000	SRAM 奇偶校验错误中断状态寄存器	0x00
MISC1_SRAMINIT	R/W	0x0004	SRAM 初始化寄存器	0x01
MISC1_PARERR	R	0x0008	SRAM 奇偶校验错误地址寄存器	0x000
MISC1_IREN	R/W	0x000C	IR 使能控制寄存器	0x00
MISC1_DUTYL	R/W	0x0010	IR 调制波低电平宽度控制寄存器	0x0000
MISC1_DUTYH	R/W	0x0014	IR 调制波高电平宽度控制寄存器	0x0000
MISC1_IRQLAT	R/W	0x0018	Cortex-M0 IRQ 延迟控制寄存器	0x000
MISC1_HIADDR	R	0x0020	记录的非法访问 AHB 地址	--
MISC1_PIADDR	R	0x0024	记录的非法访问 APB 地址	--

表 4-19 MISC2 寄存器地址(MISC2 基地址: 0x40013E00)

名称	类型	地址	描述	默认值
MISC2_FLASHWC	R/W	0x0000	FLASH 等待周期寄存器	0x2100
MISC2_CLKSEL	R/W	0x0004	系统时钟选择寄存器	0x0
MISC2_CLKDIVH	R/W	0x0008	AHB 时钟分频控制寄存器	0x00
MISC2_CLKDIVP	R/W	0x000C	APB 时钟分频控制寄存器	0x01
MISC2_HCLKEN	R/W	0x0010	AHB 时钟使能控制寄存器	0x1FF
MISC2_PCLKEN	R/W	0x0014	APB 时钟使能控制寄存器	0xFFFFFFFF

### 4.1.15 CRYPT 寄存器地址

表 4-20 CRYPT 寄存器地址(CRYPT 基地址: 0x40006000)

名称	类型	地址	描述	默认值
CRYPT_CTRL	R/W	0x0000	CRYPT 控制寄存器	0x0000
CRYPT_PTRA	R/W	0x0004	CRYPT 指针 A	0x0000
CRYPT_PTRB	R/W	0x0008	CRYPT 指针 B	0x0000
CRYPT_PTRO	R/W	0x000C	CRYPT 指针 O	0x0000
CRYPT_CARRY	R	0x0010	CRYPT 借位/进位寄存器	0x0

### 4.1.16 Info 参数寄存器

Info 信息存储在以 0x00080400 为起始地址的区域, 只能读不能写。所有信息都有双备份, 表格中第一份数据用 1 表示, 第二份用 2 表示。每份数据都带校验 (Checksum), 校验算法: 各份中的数据项的数据相加, 对该相加后的结果取反。

地址 0x00080400~0x000805FF 的数据是需要软件加载芯片寄存器的默认值, 数据在出厂前已写入。

地址 0x00080800~0x0008085C 数据, 通过专用烧写工具 (脱机烧写并开启 RTC 校正功能) 写入。其它数据出厂前已写入。

表 4-21 Info 信息寄存器

地址	符号	信息	描述
0x00080400		Count	配置值需要软件从 infor 区加载的寄存器的个数
0x00080404		Check_sum1	INV(SUM(0x80400, 0x80400))
0x00080408		address0	ANA_REG0, 0x40014200
0x0008040C		value0	0x30
0x00080410		Check_sum1	INV(SUM(0x80408, 0x8040C))
0x00080414		address1	ANA_REG4, 0x40014210
0x00080418		value1	0x01
0x0008041C		Check_sum1	INV(SUM(0x80414, 0x80418))
0x00080420		address2	ANA_REGA, 0x40014228
0x00080424		value2	0x02
0x00080428		Check_sum1	INV(SUM(0x80420, 0x80424))
0x0008042C		address3	ANA_ADCCTRL0, 0x40014268

0x00080430		value3	0x300000
0x00080434		Check_sum1	INV(SUM(0x8042C, 0x80430))
0x00080438		address4	ANA_ADCCTRL1, 0x400142E8
0x0008043C		value4	0x02
0x00080440		Check_sum1	INV(SUM(0x80438, 0x8043C))
0x00080444		address5	ANA_ADCCTRL2, 0x400142EC
0x00080448		value5	0x8010
0x0008044C		Check_sum1	INV(SUM(0x80444, 0x80448))
0x00080450		address6	RTC_ADCMACTL, 0x400148EC
0x00080454		value6	0x70000000
0x00080458		Check_sum1	INV(SUM(0x80450, 0x80454))
0x0008045C		address7	RTC_ADCDTCTL, 0x400148F0
0x00080460		value7	0x80000000
0x00080464		Check_sum1	INV(SUM(0x8045C, 0x80460))
0x00080468		address8	RTC_ADCUCALK, 0x400148E8
0x0008046C		value8	0x599A599A
0x00080470		Check_sum1	INV(SUM(0x80468, 0x8046C))
0x00080474		address9	GPIOA_IE
0x00080478		value9	未封装出来的管脚相应的 bit 位写 0, 其他封装出来的相应 bit 位写 1
0x0008047C		Check_sum1	
0x00080480		address10	GPIOB_IE
0x00080484		value10	未封装出来的管脚相应的 bit 位写 0, 其他封装出来的相应 bit 位写 1
0x00080488		Check_sum1	
0x0008048C		address11	GPIOC_IE
0x00080490		value11	未封装出来的管脚相应的 bit 位写 0, 其他封装出来的相应 bit 位写 1
0x00080494		Check_sum1	
0x00080498		address12	GPIOD_IE
0x0008049C		value12	未封装出来的管脚相应的 bit 位写 0, 其他封装出来的相应 bit 位写 1
0x000804A0		Check_sum1	
0x000804A4		address13	GPIOE_IE
0x000804A8		value13	未封装出来的管脚相应的 bit 位写 0, 其他封装出来的相应 bit 位写 1
0x000804AC		Check_sum1	
0x000804B0		address14	GPIOF_IE
0x000804B4		value14	未封装出来的管脚相应的 bit 位写 0, 其他封装出来的相应 bit 位写 1
0x000804B8		Check_sum1	

...		...	...
0x00080500		Count	配置值需要软件从 infor 区加载的寄存器的个数
0x00080504		Check_sum2	INV(SUM(0x80500, 0x80500))
0x00080508		address0	ANA_REG0, 0x40014200
0x0008050C		value0	0x30
0x00080510		Check_sum2	INV(SUM(0x80508, 0x8050C))
0x00080514		address1	ANA_REG4, 0x40014210
0x00080518		value1	0x01
0x0008051C		Check_sum2	INV(SUM(0x80514, 0x80518))
0x00080520		address2	ANA_REGA, 0x40014228
0x00080524		value2	0x02
0x00080528		Check_sum2	INV(SUM(0x80520, 0x80524))
0x0008052C		address3	ANA_ADCCTRL0, 0x40014268
0x00080530		value3	0x300000
0x00080534		Check_sum2	INV(SUM(0x8052C, 0x80530))
0x00080538		address4	ANA_ADCCTRL1, 0x400142E8
0x0008053C		value4	0x02
0x00080540		Check_sum2	INV(SUM(0x80538, 0x8053C))
0x00080544		address5	ANA_ADCCTRL2, 0x400142EC
0x00080548		value5	0x8010
0x0008054C		Check_sum2	INV(SUM(0x80544, 0x80548))
0x00080550		address6	RTC_ADCMACTL, 0x400148EC
0x00080554		value6	0x70000000
0x00080558		Check_sum2	INV(SUM(0x80550, 0x80554))
0x0008055C		address7	RTC_ADCDTCTL, 0x400148F0
0x00080560		value7	0x80000000
0x00080564		Check_sum2	INV(SUM(0x8055C, 0x80560))
0x00080568		address8	RTC_ADCUCALK, 0x400148E8
0x0008056C		value8	0x599A599A
0x00080570		Check_sum2	INV(SUM(0x80568, 0x8056C))
0x00080574		address9	GPIOA_IE
0x00080578		value9	未封装出来的管脚相应的 bit 位写 0, 其他封装出来的相应 bit 位写 1
0x0008057C		Check_sum2	
0x00080580		address10	GPIOB_IE
0x00080584		value10	未封装出来的管脚相应的 bit 位写 0, 其他封装出来的相应 bit 位写 1
0x00080588		Check_sum2	
0x0008058C		address11	GPIOC_IE

0x00080590		value11	未封装出来的管脚相应的 bit 位写 0，其他封装出来的相应 bit 位写 1
0x00080594		Check_sum2	
0x00080598		address12	GPIOD_IE
0x0008059C		value12	未封装出来的管脚相应的 bit 位写 0，其他封装出来的相应 bit 位写 1
0x000805A0		Check_sum2	
0x000805A4		address13	GPIOE_IE
0x000805A8		value13	未封装出来的管脚相应的 bit 位写 0，其他封装出来的相应 bit 位写 1
0x000804AC		Check_sum2	
0x000805B0		address14	GPIOF_IE
0x000805B4		value14	未封装出来的管脚相应的 bit 位写 0，其他封装出来的相应 bit 位写 1
0x000805B8		Check_sum2	
...	...	...	...
0x00080800	P4	晶体常温偏移 1	unit(0.1ppm)，低 16 位加载到 RTC_ACP4 寄存器，比如 0。
0x00080804		Checksum1	INV(SUM(0x80800, 0x80800))
0x00080808	P4	晶体常温偏移 2	unit(0.1ppm)，低 16 位加载到 RTC_ACP4 寄存器，比如 0。
0x0008080C		Checksum2	INV(SUM(0x80808, 0x80808))
0x00080810	K0	晶体 K0 系数 1	加载到 RTC_ACK0 寄存器，K0 计算公式如下： $K0=B0/1000000*65536$ ，B0 为晶振曲线二次函数的分段系数。 K0 值比如 20827。
0x00080814	K1	晶体 K1 系数 1	加载到 RTC_ACK1 寄存器，K1 计算公式如下： $K1=B1/1000000*65536$ ，B1 为晶振曲线二次函数的分段系数。 K1 值比如 21496。
0x00080818	K2	晶体 K2 系数 1	加载到 RTC_ACK2 寄存器，K2 计算公式如下： $K2=B2/1000000*65536$ ，B2 为晶振曲线二次函数的分段系数。 K2 值比如 22020。
0x0008081C	K3	晶体 K3 系数 1	加载到 RTC_ACK3 寄存器，K3 计算公式如下： $K3=B3/1000000*65536$ ，B3 为晶振曲线二次函数的分段系数。 K3 值比如 24517。
0x00080820	K4	晶体 K4 系数 1	加载到 RTC_ACK4 寄存器，K4 计算公式如下： $K4=B4/1000000*65536$ ，B4 为晶振曲线二次函数的分段系数。 K4 值比如 25257。
0x00080824		Checksum1	INV(SUM(0x80810, 0x80820))
0x00080828	K0	晶体 K0 系数 2	加载到 RTC_ACK0 寄存器，K0 计算公式如下： $K0=B0/1000000*65536$ ，B0 为晶振曲线二次函数的分段系数。 K0 值比如 20827。
0x0008082C	K1	晶体 K1 系数 2	加载到 RTC_ACK1 寄存器，K1 计算公式如下： $K1=B1/1000000*65536$ ，B1 为晶振曲线二次函数的分段系数。 K1 值比如 21496。
0x00080830	K2	晶体 K2 系数 2	加载到 RTC_ACK2 寄存器，K2 计算公式如下： $K2=B2/1000000*65536$ ，B2 为晶振曲线二次函数的分段系数。 K2 值比如 22020。

0x00080834	K3	晶体 K3 系数 2	加载到 RTC_ACK3 寄存器，K3 计算公式如下： $K3=B3/1000000*65536$ ，B3 为晶振曲线二次函数的分段系数。 K3 值比如 24517。
0x00080838	K4	晶体 K4 系数 2	加载到 RTC_ACK4 寄存器，K4 计算公式如下： $K4=B4/1000000*65536$ ，B4 为晶振曲线二次函数的分段系数。 K4 值比如 25257。
0x0008083C		Checksum2	INV(SUM(0x80828, 0x80838))
0x00080840	ACTI	晶体顶点温度 1	加载到 RTC_ACTI 寄存器，比如：0x1800。
0x00080844		Checksum1	INV(SUM(0x80840, 0x80840))
0x00080848	ACTI	晶体顶点温度 2	加载到 RTC_ACTI 寄存器，比如：0x1800。
0x0008084C		Checksum2	INV(SUM(0x80848, 0x80848))
0x00080850	KTEMP x(x=4 ~1)	晶体 K 系数温度分段 区间设置 1	加载到 RTC_ACKTEMP 寄存器，比如 0x3C2800EC。
0x00080854	—	Checksum1	INV(SUM(0x80850, 0x80850))
0x00080858	KTEMP x(x=4 ~1)	晶体 K 系数温度分段 区间设置 2	加载到 RTC_ACKTEMP 寄存器，比如 0x3C2800EC。
0x0008085C		Checksum2	INV(SUM(0x80858, 0x80858))
...			保留
0x00080C00	a1	a1(ADC_CH4 不分 压标定 1)(5V 系 统)	$Vdc=a1/100000000*X+b1/100000000+offset1$ (用补码方式 存放负值)
0x00080C04	b1	b1(ADC_CH4 不分 压标定 1)(5V 系 统)	
0x00080C08	a2	a2(ADC_CH4 电阻 分压标定 1)(5V 系 统)	$Vdc=a2/100000000*X+b2/100000000+offset2$ (用补码方式 存放负值)
0x00080C0C	b2	b2(ADC_CH4 电阻 分压标定 1)(5V 系 统)	
0x00080C10	a3	a3(BAT1 电阻分压 标定 1)(5V 系统)	$Vdc=a3/100000000*X+b3/100000000+offset3$ (用补码方式 存放负值)
0x00080C14	b3	b3(BAT1 电阻分压 标定 1)(5V 系统)	
0x00080C18	a4	a4(BATRTC 电阻分 压标定 1)(5V 系 统)	$Vdc=a4/100000000*X+b4/100000000+offset4$ (用补码方式 存放负值)
0x00080C1C	b4	b4(BATRTC 电阻分 压标定 1)(5V 系 统)	
0x00080C20		Checksum1	INV(SUM(0x00080C00, 0x00080C1C))
0x00080C24	a1	a1(ADC_CH4 不分 压标定 2)(5V 系 统)	同备份 1

0x00080C28	b1	b1(ADC_CH4 不分压标定 2)(5V 系统)	
0x00080C2C	a2	a2(ADC_CH4 电阻分压标定 2)(5V 系统)	
0x00080C30	b2	b2(ADC_CH4 电阻分压标定 2)(5V 系统)	
0x00080C34	a3	a3(BAT1 电阻分压标定 2)(5V 系统)	
0x00080C38	b3	b3(BAT1 电阻分压标定 2)(5V 系统)	
0x00080C3C	a4	a4(BATRTC 电阻分压标定 2)(5V 系统)	
0x00080C40	b4	b4(BATRTC 电阻分压标定 2)(5V 系统)	
0x00080C44		Checksum2	
0x00080C48	a1	a1(ADC_CH4 不分压标定 1)(3.3V 系统)	原始值放大 100000000 倍, 保存 32 位值 (用补码方式存放负值)
0x00080C4C	b1	b1(ADC_CH4 不分压标定 1)(3.3V 系统)	原始值放大 100000000 倍, 保存 32 位值 (用补码方式存放负值)
0x00080C50	a2	a2(ADC_CH4 电阻分压标定 1)(3.3V 系统)	原始值放大 100000000 倍, 保存 32 位值 (用补码方式存放负值)
0x00080C54	b2	b2(ADC_CH4 电阻分压标定 1)(3.3V 系统)	原始值放大 100000000 倍, 保存 32 位值 (用补码方式存放负值)
0x00080C58	a3	a3(BAT1 电阻分压标定 1)(3.3V 系统)	原始值放大 100000000 倍, 保存 32 位值 (用补码方式存放负值)
0x00080C5C	b3	b3(BAT1 电阻分压标定 1)(3.3V 系统)	原始值放大 100000000 倍, 保存 32 位值 (用补码方式存放负值)
0x00080C60	a4	a4(BATRTC 电阻分压标定 1)(3.3V 系统)	原始值放大 100000000 倍, 保存 32 位值 (用补码方式存放负值)
0x00080C64	b4	b4(BATRTC 电阻分压标定 1)(3.3V 系统)	原始值放大 100000000 倍, 保存 32 位值 (用补码方式存放负值)
0x00080C68		Checksum1	INV(SUM(0x80C48, 0x80C64))
0x00080C6C	a1	a1(ADC_CH4 不分压标定 2)(3.3V 系统)	同备份 1

0x00080C70	b1	b1(ADC_CH4 不分压标定 2)(3.3V 系统)	
0x00080C74	a2	a2(ADC_CH4 电阻分压标定 2)(3.3V 系统)	
0x00080C78	b2	b2(ADC_CH4 电阻分压标定 2)(3.3V 系统)	
0x00080C7C	a3	a3(BAT1 电阻分压标定 2)(3.3V 系统)	
0x00080C80	b3	b3(BAT1 电阻分压标定 2)(3.3V 系统)	
0x00080C84	a4	a4(BATRTC 电阻分压标定 2)(3.3V 系统)	
0x00080C88	b4	b4(BATRTC 电阻分压标定 2)(3.3V 系统)	
0x00080C8C		Checksum2	
0x00080C90	Offset1	ADC_CH4 不分压测试偏差 1 (5V 系统)	(测试值-理论值) *1000
0x00080C94	Offset2	ADC_CH4 电阻分压测试偏差 1 (5V 系统)	(测试值-理论值) *1000
0x00080C98	Offset3	BAT1 电阻分压偏差 1 (5V 系统)	(测试值-理论值) *1000
0x00080C9C	Offset4	BATRTC 电阻分压偏差 1 (5V 系统)	(测试值-理论值) *1000
0x00080CA0		Checksum1	INV(SUM(0x80C90, 0x80C9C))
0x00080CA4	Offset1	ADC_CH4 不分压测试偏差 2 (5V 系统)	同备份 1
0x00080CA8	Offset2	ADC_CH4 电阻分压测试偏差 2 (5V 系统)	
0x00080CAC	Offset3	BAT1 电阻分压偏差 2 (5V 系统)	
0x00080CB0	Offset4	BATRTC 电阻分压偏差 2 (5V 系统)	
0x00080CB4		Checksum2	
0x00080CB8	Offset1	ADC_CH4 不分压测试偏差 1 (3.3V 系统)	(测试值-理论值) *1000

0x00080CBC	Offset2	ADC_CH4 电阻分压测试偏差 1 (3.3V 系统)	(测试值-理论值) *1000
0x00080CC0	Offset3	BAT1 电阻分压偏差 1 (3.3V 系统)	(测试值-理论值) *1000
0x00080CC4	Offset4	BATRTC 电阻分压偏差 1 (3.3V 系统)	(测试值-理论值) *1000
0x00080CC8		Checksum1	INV(SUM(0x80CB8, 0x80CC4))
0x00080CCC	Offset1	ADC_CH4 不分压测试偏差 2 (3.3V 系统)	同备份 1
0x00080CD0	Offset2	ADC_CH4 电阻分压测试偏差 2 (3.3V 系统)	
0x00080CD4	Offset3	BAT1 电阻分压偏差 2 (3.3V 系统)	
0x00080CD8	Offset4	BATRTC 电阻分压偏差 2 (3.3V 系统)	
0x00080CDC		Checksum2	
0x00080D10	P1/P0	RTC_ACP1/0 设置 1	高 16 位加载到 RTC_ACP1 寄存器, 比如: 1060; 低 16 位加载到 RTC_ACP0 寄存器, 比如: -214。
0x00080D14	P2'	RTC_ACP2 设置 1	该地址存储值记为 P2', 比如: -19746971。根据公式: $P2 = P2' + (Tr - Tm) * 256$ , 将 P2 加载到 RTC_ACP2 寄存器。
0x00080D18	P5	RTC_ACP5 设置 1	高 16 位加载到 RTC_ACP5 寄存器, 比如: 6444; 低 16 位舍弃。
0x00080D1C	P7/P6'	RTC_ACP7/6 设置 1	高 16 位加载到 RTC_ACP7 寄存器, 比如: 0; 低 16 位加载到 P6'。比如: 1342。 根据公式 $P6 = a * P6'$ , 将 P6 加载至 RTC_ACP6 寄存器。
0x00080D20		Checksum1	INV(SUM(0x80D10, 0x80D1C))
0x00080D24	P1/P0	RTC_ACP1/0 设置 2	同备份 1
0x00080D28	P2'	RTC_ACP2 设置 2	
0x00080D2C	P5	RTC_ACP5 设置 2	
0x00080D30	P7/P6'	RTC_ACP7/6 设置 2	
0x00080D34		Checksum2	
0x00080D38		AVCC gain 1	Pre-trim result/3.3 * 10000
0x00080D3C		DVCC gain 1	Pre-trim result/1.5 * 10000
0x00080D40		BGP gain 1	Pre-trim result/1.2 * 10000
0x00080D44		RCL gain 1	Pre-trim result/32768 * 10000
0x00080D48		RCH gain 1	Pre-trim result/6553600 * 10000
0x00080D4C		Checksum1	INV(SUM(0x80D38, 0x80D48))
0x00080D50		AVCC gain 2	同备份 1
0x00080D54		DVCC gain 2	
0x00080D58		BGP gain 2	

0x00080D5C		RCL gain 2	
0x00080D60		RCH gain 2	
0x00080D64		Checksum2	
0x00080D68		ID word 0, Backup 1	
0x00080D6C		ID word 1, Backup 1	
0x00080D70		Checksum1	INV(SUM(0x80D68, 0x80D6C))
0x00080D74		ID word 0, Backup 2	同备份 1
0x00080D78		ID word 1, Backup 2	
0x00080D7C		Checksum2	
0x00080D80	Tr	实际温度 Tr 1 (来自 tmp275)	根据公式: $P2 = P2' + (Tr - Tm) * 256$ , 将 P2 加载到 RTC_ACP2 寄存器
0x00080D84	Tm	测量温度 Tm 1 (来自 ADC)	根据公式: $P2 = P2' + (Tr - Tm) * 256$ , 将 P2 加载到 RTC_ACP2 寄存器
0x00080D88		Checksum1	INV(SUM(0x80D80, 0x80D84))
0x00080D8C	Tr	实际温度 Tr 2 (来自 tmp275)	同备份 1
0x00080D90	Tm	测量温度 Tm 2 (来自 ADC)	
0x00080D94		Checksum2	
...			保留
0x00080DC0		ANA_REGBCDE trim data1	芯片复位后自动加载到 ANA_REGx (x=B~E) 的值。
0x00080DC4		0xFFFFFFFF	
0x00080DC8		0xFFFFFFFF	
0x00080DCC		Checksum1	INV(SUM(0x80DC0, 0x80DC8)), 其中 0x00080DC8 地址不支持读操作, 数据使用 0xFFFFFFFF 代替。
0x00080DD0		ANA_REGBCDE trim data2	芯片复位后自动加载到 ANA_REGx (x=B~E) 的值。
0x00080DD4		0xFFFFFFFF	
0x00080DD8		0xFFFFFFFF	
0x00080DDC		Checksum2	INV(SUM(0x80DD0, 0x80DD8)), 其中 0x00080DD8 地址不支持读操作, 数据使用 0xFFFFFFFF 代替。
0x00080DE0		ANA_REG10 trim data1	芯片复位后自动加载到 ANA_REG10 的值
0x00080DE4		Checksum1	INV(SUM(0x80DE0, 0x80DE0))
0x00080DE8		ANA_REG10 trim data2	芯片复位后自动加载到 ANA_REG10 的值
0x00080DEC		Checksum2	INV(SUM(0x80DE8, 0x80DE8))

## 第5章 电源系统

V85XXP 电源控制器有以下特点：

- 输入电压范围：2.2V~5.5V
- 支持 RTC 电源独立供电
- GPIO 电源为 VDD
- 模拟电路由 AVCC 供电
- 数字电路和 PLL 电路由 DVCC 供电
- 支持低电压监测和实时电池监测

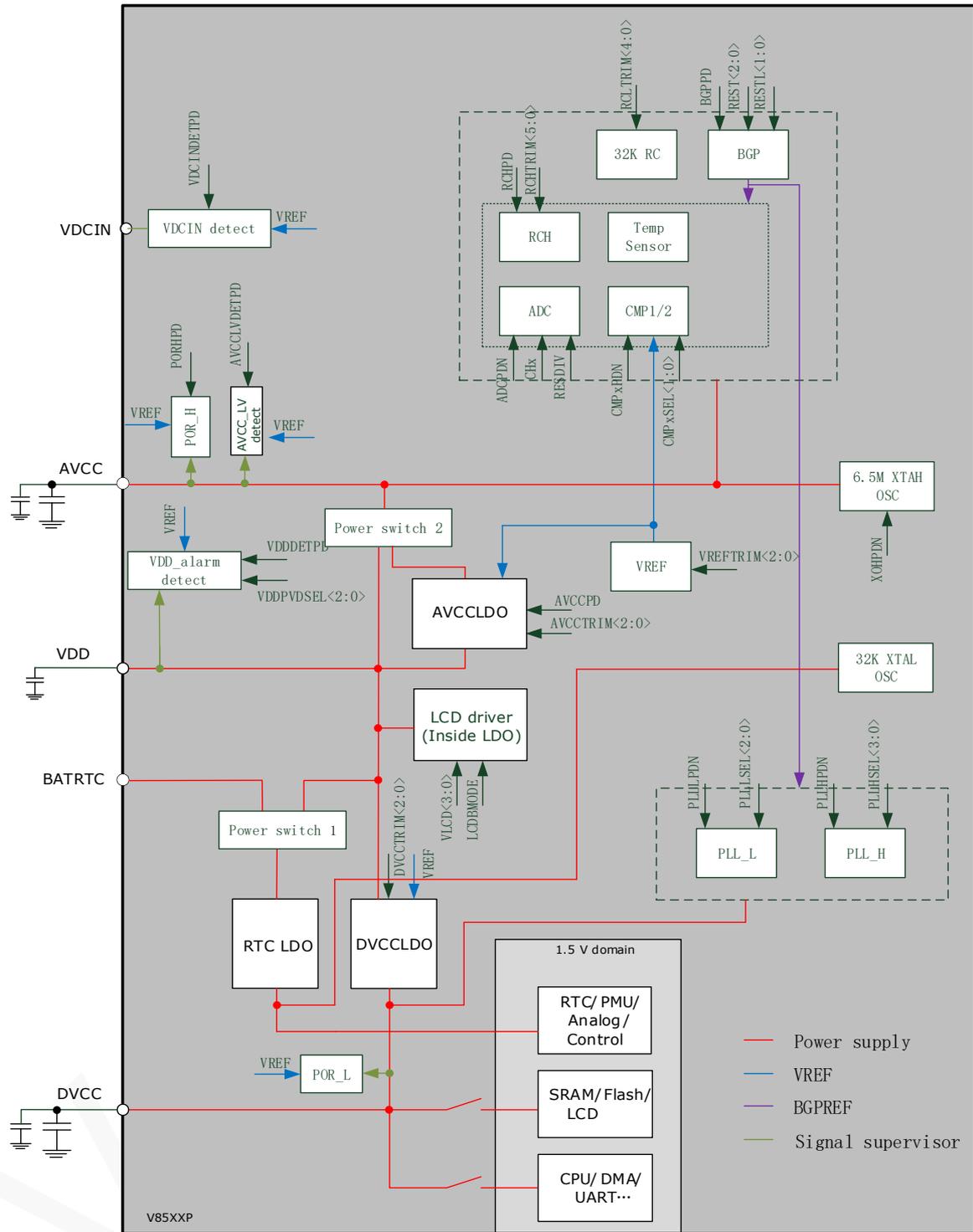


图 5-1 V85XXP 电源系统框图

## 5.1 寄存器地址

表 5-1 ANA 寄存器地址(ANA 基地址: 0x40014200)

命名	类型	地址	描述	默认值
ANA_REG5	R/W	0x0014	模拟寄存器 5	0x00
ANA_REG6	R/W	0x0018	模拟寄存器 6	0x00
ANA_REG7	R/W	0x001C	模拟寄存器 7	0x00
ANA_REG8	R/W	0x0020	模拟寄存器 8	0x00
ANA_REG9	R/W	0x0024	模拟寄存器 9	0x00
ANA_REGA	R/W	0x0028	模拟寄存器 10	0x00
ANA_REGF	R/W	0x003C	模拟寄存器 15	0x00
ANA_CTRL	R/W	0x0050	模拟控制寄存器	0x0000000
ANA_CMPOUT	R	0x0054	比较器结果寄存器	0x0030
ANA_INTSTS	R/C	0x0060	模拟中断状态寄存器	0x0000
ANA_INTEN	R/W	0x0064	模拟中断使能寄存器	0x0000
ANA_CMPCTL	R/W	0x006C	比较器控制寄存器	0x00000000

## 5.2 寄存器定义

### 5.2.1 ANA\_REG5 寄存器

表 5-2 ANA\_REG5 寄存器

位	名称	类型	描述	默认值
6	PD_AVCCDET	R/W	关闭 AVCC 低电压监测模块。 0: 开启。 1: 关闭。	0

### 5.2.2 ANA\_REG6 寄存器

表 5-3 ANA\_REG6 寄存器

位	名称	类型	描述	默认值
7	BATRCDISC	R/W	给 RTC 电池 BATRTC 放电。放电电阻是 1.7k，放电电流是 Vbatrtc/1.7k。 0: 禁用从 BATRTC 到 GND 的 1.7k 欧姆电阻。 1: 使能从 BATRTC 到 GND 的 1.7k 欧姆电阻。	0

### 5.2.3 ANA\_REG7 寄存器

表 5-4 ANA\_REG7 寄存器

位	名称	类型	描述	默认值
7: 3		R/W	保留	0
2	VDCINHYSSEL	R/W	VDCIN 管脚电压检测回差选择 0: 100mV 1: 200mV	0
1: 0		R/W	保留	0

## 5.2.4 ANA\_REG8 寄存器

表 5-5 ANA\_REG8 寄存器

位	名称	类型	描述	默认值
3: 0			保留	0
6: 4	VDDPVDSEL[2: 0]	R/W	VDD 电压报警阈值设置寄存器，当 VDD 电压低于该设置值时，VDDALARM 中断标志位会置 1。  000: 4.5V 001: 4.2V 010: 3.9V 011: 3.6V 100: 3.2V 101: 2.9V 110: 2.6V 111: 2.3V	0
7	AVCCLDOPD	R/W	0: 开启 LDO，AVCC 引脚电压为 3.3V。 1: 关闭 LDO，AVCC 引脚通过一个开关连接到 VDD，开关内阻请参考表 1-4。  只有当 VDD 管脚输入电压低于 3.6V 时，才可以关闭 AVCCLDO。	0

## 5.2.5 ANA\_REG9 寄存器

表 5-6 ANA\_REG9 寄存器

位	名称	类型	描述	默认值
7	VDDDETPD	R/W	关闭 VDD 电压监测模块 (VDDALARM)，该模块由 VDD 供电。 0: 开启，该模块不可以由硬件控制。 1: 关闭，该模块可以由硬件开启，详见 ANA_CMPCTL 的 bit22~23。	0

## 5.2.6 ANA\_REGA 寄存器

表 5-7 ANA\_REGA 寄存器

位	名称	类型	描述	默认值
6: 0	-	R/W	保留	默认值是 0x0，必须配置为 0x02。
7	VDCINDETPD	R/W	VDCIN 监测模块开关。 0: 开启 VDCIN 监测 1: 关闭 VDCIN 监测	0

## 5.2.7 ANA\_REGF 寄存器

表 5-8 ANA\_REGF 寄存器

位	名称	类型	描述	默认值
2	AVCCO_EN	R/W	AVCC_OUT 引脚输出使能控制。 0: 高阻。 1: AVCC_OUT 输出 AVCC 电平，可以用来驱动小功率模块。	0

## 5.2.8 ANA\_CTRL 寄存器

表 5-9 ANA\_CTRL 寄存器

位	名称	类型	描述	默认值
26	PDNS2	R/W	该位控制 VDDALARM 为 0（同时考虑 PDNS 的设置）时进入深睡眠的行为。 0: 当 VDDALARM 为 0 时不会进入深睡眠。当 VDDALARM 为 1 时，系统可以进入深睡眠模式，并且当 VDDALARM 变为 0 时，系统自动从深睡眠唤醒。 1: VDDALARM 不作为深睡眠条件。	0x0
6	PDNS	R/W	该位控制 VDCINDROP 为 0（同时考虑 PDNS2 的设置）时进入深睡眠的行为。 0: VDCINDROP 为 0 时不会进入深睡眠。当 VDCINDROP 为 1 时，系统可以进入深睡眠，并且当 VDCINDROP 变为 0 时，系统自动从深睡眠唤醒。 1: VDCINDROP 不作为深睡眠条件。	0x0

## 5.2.9 ANA\_CMPOUT 寄存器

表 5-10 ANA\_CMPOUT 寄存器

位	名称	类型	描述	默认值
10	AVCCLV	R	AVCC 低电压状态位。电压回差是 20mV~30mV。 0: AVCC 高于 2.5V 1: AVCC 低于 2.5V	0x0
8	VDCINDROP	R	VDCIN 掉电状态位 0: VDCIN 未掉电（VDCIN 大于阈值） 1: VDCIN 掉电（VDCIN 小于阈值）	0x0
7	VDDALARM	R	VDD 电压报警状态位 0: VDD 电压高于 VDDPVDSEL 设置的电压	0x0

1: VDD 电压低于 VDDPVDSEL 设置的电压

## 5.2.10 ANA\_INTSTS 寄存器

表 5-11 ANA\_INTSTS 寄存器

位	名称	类型	描述	默认值
11	INTSTS11	R/C	当 VDCINDROP 为 0，进入睡眠模式中断标志位（VDCIN 大于阈值）。当 VDCINDROP 为 0，并且检测到系统进入睡眠的动作，将置位中断标志位。用户可以使能中断使能位（INTEN11），使得系统可以从浅睡眠或深睡眠模式唤醒（VDCINDROP 为 0）。 读 0：无进入睡眠事件发生（VDCINDROP=0） 读 1：发生进入睡眠事件（VDCINDROP=0） 写 0：无影响 写 1：清除该位	0x0
10	INTSTS10	R/C	AVCCLV 的中断标志，该中断会在 AVCCLV 上升和下降时产生。 读 0：无 AVCCLV 中断 读 1：发生 AVCCLV 中断 写 0：无效 写 1：清除该位	0x0
8	INTSTS8	R/C	VDCINDROP 的中断标志，该中断会在 VDCINDROP 上升和下降时产生。 读 0：无 VDCIN 中断 读 1：发生 VDCIN 中断 写 0：无效 写 1：清除该位	0x0
7	INTSTS7	R/C	VDDALARM 的中断标志，该中断会在 VDDALARM 上升和下降时产生。 读 0：无 VDDALARM 中断 读 1：发生 VDDALARM 中断 写 0：无效 写 1：清除该位	0x0

## 5.2.11 ANA\_INTEN 寄存器

表 5-12 ANA\_INTEN 寄存器

位	名称	类型	描述	默认值
11	INTEN11	R/W	当 VDCINDROP 为 0，检测到进入睡眠模式的动作，该事件的中断和唤醒使能控制。 0：当 VDCINDROP 为 0 时，检测到进入睡眠，禁止该中断或唤醒 1：当 VDCINDROP 为 0 时，检测到进入睡眠，使能该中断或唤醒	0x0
10	INTEN10	R/W	AVCCLV 中断或唤醒使能控制。	0x0

			0: 禁止 AVCCLV 中断或唤醒 1: 使能 AVCCLV 中断或唤醒	
8	INTEN8	R/W	VDCINDROP 中断或唤醒使能控制。 0: 禁止 VDCIN 中断或唤醒 1: 使能 VDCIN 中断或唤醒	0x0
7	INTEN7	R/W	VDDALARM 中断或唤醒使能控制。 0: 禁止 VDDALARM 中断或唤醒 1: 使能 VDDALARM 中断或唤醒	0x0

## 5.2.12 ANA\_CMPCTL 寄存器

表 5-13 ANA\_CMPCTL 寄存器

位	名称	类型	描述	默认值
31 : 24	PWR_DEB_SEL	R/W	VDDALARM、VDCIN 和 AVCCDET 的数字滤波控制寄存器。 0: 无滤波。 1: 2 个 RTCCLK 时钟滤波 2: 3 个 RTCCLK 时钟滤波 3: 4 个 RTCCLK 时钟滤波 4: 5 个 RTCCLK 时钟滤波 ... 255: 256 个 RTCCLK 时钟滤波	0x00
23 : 22	VDDALARM_CHK_FRQ_SEL	R/W	VDDALARM 检测硬件间歇开启控制。当 ANA_REG9 的 bit7 配置为 1 时，该硬件控制生效。 0: 一直不开启，不检测 1: 以 RTCCLK 时钟频率，一直开启检测 2: 每 7.8125ms 间隔开启检测一次，检测完毕后，自动关闭。 3: 保留	0x0

## 5.3 AVCC

V85XXP 中的模拟电路由 AVCC 供电，AVCC 状态由控制位 AVCCLDOPD (ANA\_REG8 bit7) 控制。只有当 VDD 管脚输入电压低于 3.6V 时，才可以关闭 AVCCLDOPD。

AVCC 的驱动能力是 30mA。

AVCC\_OUT 为可控制的 AVCC 电源输出管脚，通过控制位 AVCCOEN (ANA\_REGF bit2) 可关闭管脚输出，默认不输出。AVCC\_OUT 管脚的驱动能力是 20mA，AVCC、AVCC\_OUT 两个管脚同时有驱动的话，驱动电流不能超过 30mA。

芯片内设计了 AVCC 的电压监测模块，AVCC 电压低于 2.5V 时，将会产生 AVCCLV 中断。

AVCC 管脚外部需要连接去耦电容，建议采用一个 10uF 电容与一个 0.1uF 电容并联。

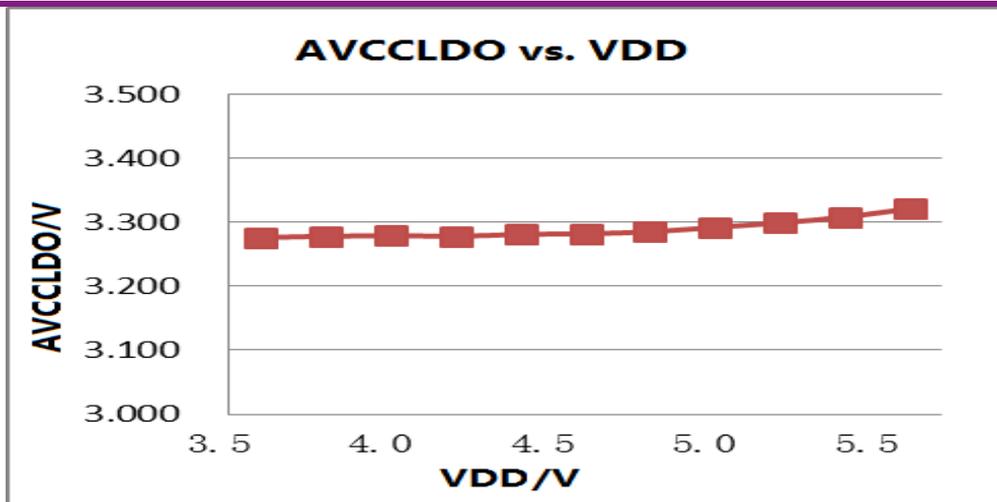


图 5-2 AVCCCLDO 输出电压与 VDD 输入电压之间的关系

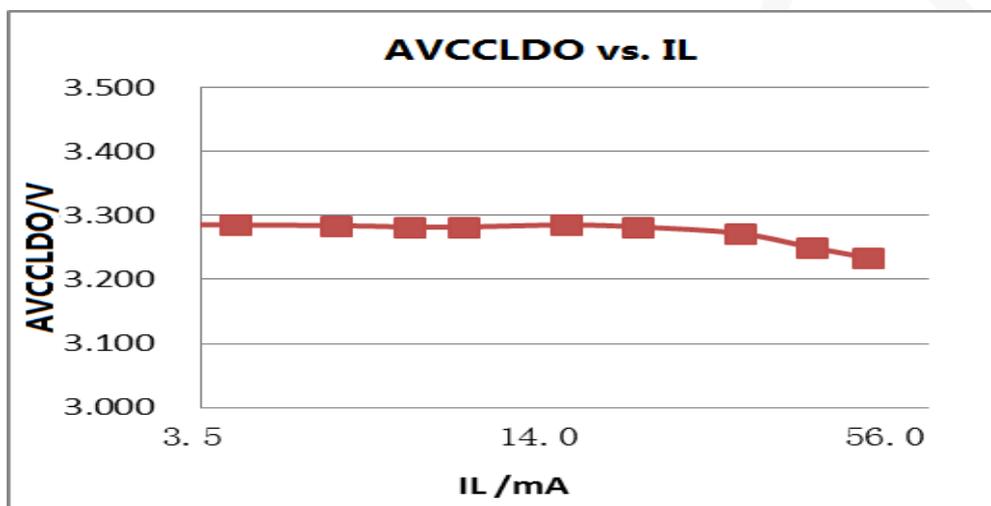


图 5-3 AVCCCLDO 输出电压与负载电流之间的关系

## 5.4 DVCC

V85XXP 的 PLL 电路和数字电路都由数字电路电源 DVCC 供电。当 DVCC 的输入电压大于 1.7V 时，DVCC 会输出稳定的 1.5V 电压。

数字电路电源的驱动能力为 35mA，当负载电流小于等于 35mA 时，电源输出电压稳定；负载电流大于 35mA 时，负载电流越高，则数字供电越低。

该电源电路始终工作，除非整个芯片系统掉电。

DVCC 管脚外部需要连接去耦电容，建议采用一个 10F 电容与一个 0.1uF 电容并联。

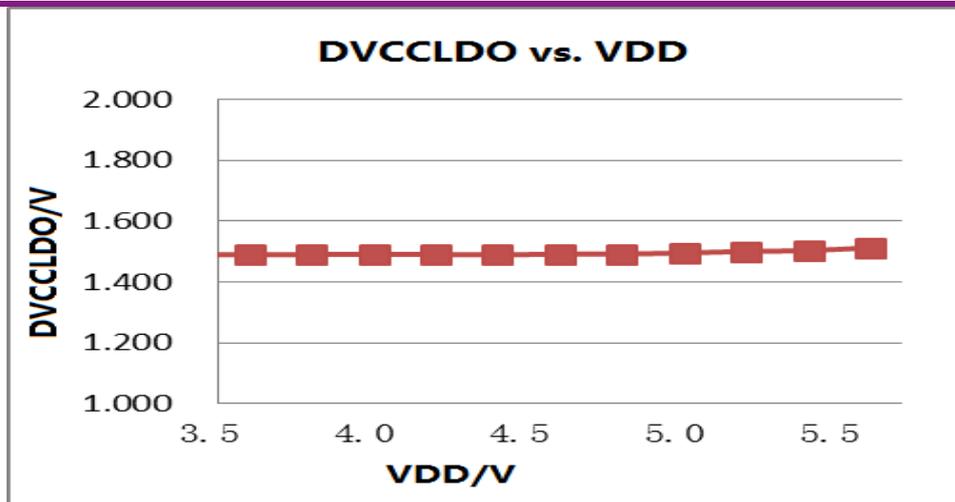


图 5-4 DVCCLDO 输出电压与 VDD 输入电压之间的关系

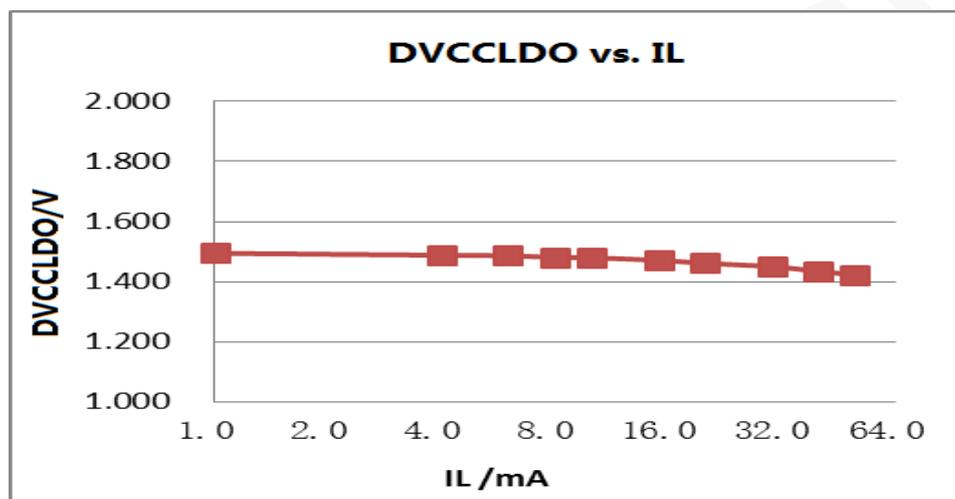


图 5-5 DVCCLDO 输出电压与负载电流之间的关系

## 5.5 电源监测

### 5.5.1 VDCIN 电源监测

在 V85XXP 中，主电源电阻分压后送入 VDCIN 管脚。电源检测系统会持续监视 VDCIN 管脚的电压。当 VDCIN 电压低于 1.0V 时，标志位 VDCINDROP (ANACMP\_OUT bit8) 将会被置 1 并产生掉电中断。VDCINDROP 将持续为 1，直至电压恢复到 1.1V/1.2V 以上，回差可以通过 VDCINHYSSEL(ANA\_REG7 bit2) 进行配置。

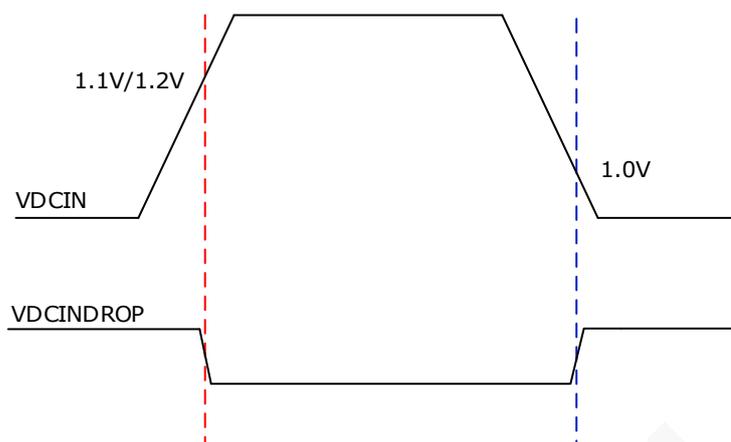


图 5-6 VDCIN 管脚输入信号与位 VDCINDROP 的关系

## 5.5.2 VDD 电源监测

该电源监测模块监测 VDD 的输入电压，当 VDD 的电压低于阈值（寄存器 ANA\_REG8 的位 VDDPVDSEL<2:0>设置的电压）时，检测模块将向 MCU 发出一个警告中断。

VDDPVDSEL<2:0>	电压报警阈值设置	000: 4.5V 001: 4.2V 010: 3.9V 011: 3.6V 100: 3.2V 101: 2.9V 110: 2.6V 111: 2.3V
----------------	----------	--

## 5.5.3 AVCC 电源监测

AVCC 低压检测模块监视 AVCC 的低压状态，当 AVCC 电压低于 2.5V 时，AVCC 低压监测模块将会为 MCU 产生一个低压中断信号。此时 ADC 的精度会明显下降，所以不要使用 ADC 采样任何信号。

## 5.6 电源切换开关

### 5.6.1 电源切换开关 1

电源开关 1 用于在 VDD、BATRTC 之间选择 RTC 的输入源。当 VDD 高于 RTC 电池 BATRTC 的电压时，电源开关将选择 VDD 为 RTC 的供电源，否则，选择 BATRTC 为 RTC 的供电源。

### 5.6.2 电源切换开关 2

电源开关 2 用于选择 AVCC 的输入源。当 AVCCLDOPD(ANA\_REG8 bit 7)为 0，AVCC 输入源为 AVCCLD0 的输出，电压为 3.3V；当 AVCCLDOPD(ANA\_REG8 bit 7)为 1，AVCC 引脚通过一个开关连接到 VDD，开关内阻请参考表 1-4。

只有当 VDD 管脚输入电压低于 3.6V 时，才可以关闭 AVCCLD0。

## 5.7 应用笔记

双电源供电：VDD 由一个电源供电，BATRTC 由另一个电源供电。

单电源供电：VDD、BATRTC 脚连接，统一连到单电源供电上。

## 第6章 工作模式

### 6.1 简介

PMU 控制器（电源控制器 Power Management Unit，简称 PMU）用于控制 V85XXP 的浅睡眠和深睡眠模式。PMU 控制器内支持最多 16 个外部 IO 唤醒源，可以将芯片从浅睡眠或深睡眠模式唤醒。

浅睡眠模式将掉电部分外设，只有 CPU、SRAM、GPIO、RTC、模拟控制器、PMU 和 UART32K 等模块在此模式下会工作。

深睡眠模式将掉电 CPU 和所有外设（包括 FLASH、SRAM、LCD 控制器和 GPIOB~GPIOF）。只有 GPIOA、RTC、模拟控制器、PMU 和 UART32K 模块在此模式下会工作。

RTC ONLY 模式将掉电 CPU 和所有外设，包括 FLASH、SRAM、LCD 控制器、GPIO、模拟控制器、PMU 和 UART32K 模块。只有 RTC 在此模式下会工作。

### 6.2 特点

- 深睡眠模式进出可控，并且深睡眠唤醒后复位相应模块，程序从 0 地址开始运行；
- 浅睡眠模式进出可控，浅睡眠唤醒后，MCU 进相应唤醒源中断服务函数，中断事件响应完成后，回到主程序继续往下运行；
- 密码保护，以避免在意外事件中进入深睡眠模式；
- 16 个具有唤醒和中断功能的 GPIO；
- 进入浅睡眠和深睡眠时会自动关闭时钟，浅睡眠唤醒后会自动打开睡眠前的系统时钟；深睡眠唤醒后会自动打开 RCH 时钟；
- 晶体缺失监测和中断；
- 内置的 256 字节 SRAM 在深睡眠模式下可保存必要的数；
- 深睡眠和浅睡眠唤醒后自动打开看门狗。

## 6.3 功能框图

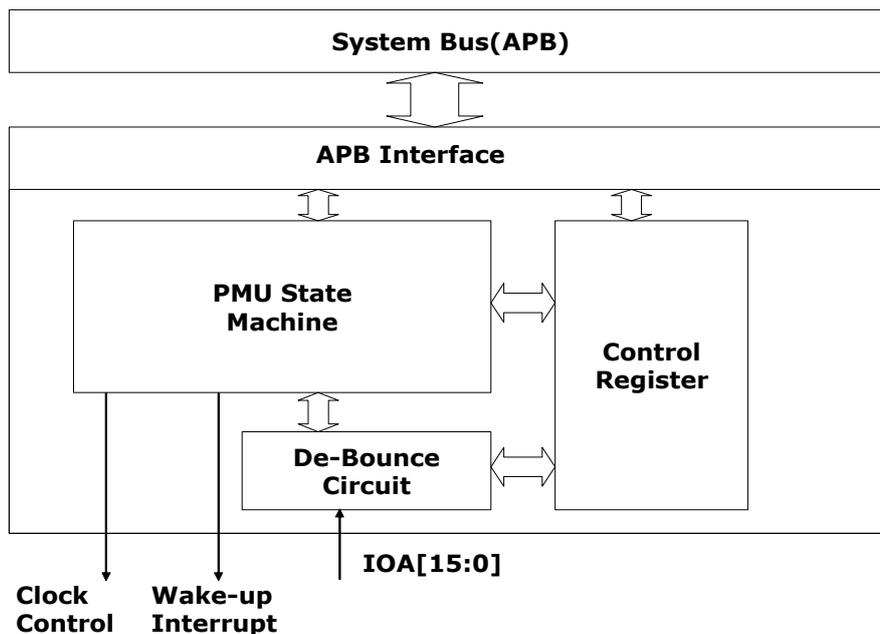


图 6-1 PMU 功能框图

## 6.4 寄存器地址

表 6-1 PMU 工作模式寄存器地址(PMU 基地址: 0x40014000)

名称	类型	地址	描述	默认值
PMU_DSLEEPEN	R/W	0x0000	PMU 深睡眠使能寄存器	0x00000000
PMU_DSLEEPPASS	R/W	0x0004	PMU 深睡眠密码寄存器	0x00000000
PMU_CONTROL	R/W	0x0008	PMU 控制寄存器	0x0000
PMU_STS	R/C	0x000C	PMU 状态寄存器	0x0000074

表 6-2 PMU RAM Retention 寄存器地址(PMU 掉电保持 RAM 基地址: 0x40014400)

名称	类型	地址	描述	默认值
PMU_RAM0	R/W	0x0000	PMU32 掉电保持 RAM0	--
PMU_RAM1	R/W	0x0004	PMU32 掉电保持 RAM1	--
PMU_RAM2	R/W	0x0008	PMU32 掉电保持 RAM2	--
.....	.....	.....	.....	--
PMU_RAM63	R/W	0x00FC	PMU32 掉电保持 RAM63	--

表 6-3 ANA 控制器寄存器地址(基地址: 0x40014200)

名称	类型	地址	描述	默认值
ANA_CTRL	R/W	0x0050	模拟控制寄存器	0x00000000

## 6.5 寄存器定义

### 6.5.1 PMU\_DSLEEPEN 寄存器

为使能深睡眠模式，用户应该先将密码 0xAA5555AA 写入 PMU\_DSLEPPASS 寄存器，然后写 0x55AAAA55 到 PMU\_DSLEEPEN 寄存器。在调试模式（模式为 0）时，不允许进入深睡眠和浅睡眠模式，所有写入到 PMU\_DSLEEPEN 寄存器的值将被丢弃。如果芯片在深睡眠模式且 MODE 管脚由 1 变化为 0，它会自动唤醒且复位，使能 ICE 接口的访问。

表 6-4 PMU\_DSLEEPEN 寄存器

位	名称	类型	描述	默认值
31	WKU	R	当前唤醒信号状态。如果是 1，表明有唤醒信号没有清除，用户必须清除相应中断标志位，确保进入深睡眠模式之前该位为 0，否则系统将因未清除唤醒事件而立即唤醒。	0x0
30: 0			保留	0

### 6.5.2 PMU\_DSLEPPASS 寄存器

表 6-5 PMU\_DSLEPPASS 寄存器

位	名称	类型	描述	默认值
31: 1			保留	0
0	UNLOCK	R	该位表明深睡眠模式入口已解锁，并准备进入深睡眠模式。为解锁深睡眠模式，用户应该写 0xAA5555AA 到该寄存器。该位会在任意寄存器读或 PMU 寄存器写后立即清除，包括 ICE 的读/写。因而用户应立即在 PMU_DSLEEPEN 中设置正确的密码。	0x0

### 6.5.3 PMU\_CONTROL 寄存器

表 6-6 PMU\_CONTROL 寄存器

位	名称	类型	描述	默认值
31: 21			保留	0
20	FORCE_CLKSEL_RCH	R/W	空闲模式的唤醒时钟选择 0: 与空闲前相同 1: RCH	0x0
19: 16	-	-	保留	0
15: 8	PWUPCYC	R/W	上电周期计数，该寄存器控制收到唤醒时的上电等待时间。单位为 32K 时钟周期。 建议配置为 0。	0x0
7: 6	-	-	保留	0
5	PLLL_SEL	R/W	低速 PLL 输入时钟选择。 0: 32K XTAL 1: 32K RC	0x0
4	PLLH_SEL	R/W	高速 PLL 输入时钟选择。 0: 6.5MHz RC 1: 6.5536MHz XTAL	0x0

3	INT_6M_EN	R/W	6.5536M XTAL 失效中断使能控制位。该位用于控制输出到 CPU 的中断信号。该位设置为 1 后，如果 6.5536M 晶振被移除或损坏，则会向 CPU 发出中断。	0x0
2	INT_32K_EN	R/W	32K XTAL 失效中断使能控制位。该位用于控制输出到 CPU 的中断信号。该位设置为 1 后，如果 32K 晶振被移除或损坏，则会向 CPU 发出中断。如果这个事件发生在睡眠期间，CPU 将被唤醒。	0x0
1	RTCCLK_SEL	R/W	RTC 时钟选择。 0: 32K XTAL 1: 32K RC	0x0
0	INT_IOA_EN	R/W	IOA0~15 中断使能控制位，用来设置外部中断唤醒 IO 的使能控制。	0x0

### 6.5.4 PMU\_STS 寄存器

表 6-7 PMU\_STS 寄存器

位	名称	类型	描述	默认值
31: 25	-	-	保留	0
24	MODE	R	该位 MODE 引脚当前状态。 0: 调试模式 1: 非调试模式	--
23	-	-	保留	0
22	WKUMODE	R	此寄存器用于指示上一个从深睡眠模式唤醒的源是否来自 MODE 管脚从 1 更改为 0。只要进入深睡眠模式，此位将清除为 0。	--
21	-	-	保留	--
20	WKUXTAL	R	此寄存器用于指示上一个从深睡眠模式唤醒的源是否来自晶体停振检测模块。只要进入深睡眠模式，此位将清除为 0。	--
19	WKUU32K	R	此寄存器用于指示上一个从深睡眠模式唤醒的源是否来自 UART32K 模块。只要进入深睡眠模式，此位将清除为 0。	--
18	WKUANA	R	此寄存器用于指示上一个从深睡眠模式唤醒的源是否来自模拟模块。只要进入深睡眠模式，此位将清除为 0。	--
17	WKURTC	R	此寄存器用于指示上一个从深睡眠模式唤醒的源是否来自 RTC 模块。只要进入深睡眠模式，此位将清除为 0。	--
16	WKUIOA	R	此寄存器用于指示上一个从深睡眠模式唤醒的源是否来自 IOA 模块。只要进入深睡眠模式，此位将清除为 0。	--
15: 11	-	-	保留	
10	MODERST	R/C	此寄存器表示睡眠模式或深睡眠模式下，由于发生了 MODE 管脚从高到低的变化，导致 MCU 被唤醒且发生复位。用户可以读取该位，以确定复位源是否为 MODE 唤醒复位。写 1 清零该位。	--
9	-	-	保留	
8	SFTRST	R/C	此寄存器表示发生了软复位。用户可以读取这个位，以确定复位源是否是软复位。写 1 清零该位。	--
7	WDTRST	R/C	此寄存器表示发生了看门狗复位。用户可以读取这个位，以确定复位源是否是看门狗复位。写 1 清零该位。	--

6	DPORST	R/C	该位表示上一次复位是否由内部数字上电复位信号引起，只有第一次上电（BATRTC 也要放电干净）才会置 1。写 1 清零该位。	0x1
5	PORST	R/C	该位指示上一次复位是 PORH 复位或 PORL 复位，PORH 是 AVCCCLDO 电压低于 2.08V 时发生的复位；PORL 是 DVCCCLDO 电压低于 1.3V 时发生的复位。写 1 清零该位。	0x1
4	EXTRST	R/C	该位指示上一次复位是否由外部复位信号引起。写 1 清零该位。	0x1
3	EXIST_6M	R	6.5536M XTAL 存在状态寄存器。该位表示 6.5536M XTAL 存在或不存在。开启 6.5536M XTAL（ANA_REG3 的 BIT7 配置为 1）后，该状态位才会刷新，否则保持之前状态。 0: 6.5536M 晶体不存在 1: 6.5536M 晶体存在	0x0
2	EXIST_32K	R	32K XTAL 存在状态寄存器。该位表示 32K XTAL 存在或不存在。 0: 32K 晶体不存在 1: 32K 晶体存在	0x1
1	INT_6M	R/C	该位表示 6.5536M 晶振失效中断状态。当 EXIST_6M 状态位从 1 变成 0 的时候，该状态位置 1。 当该位为 1 时，表示 6.5536M 晶振被去除或损坏。 写 1 清零该位。	0x0
0	INT_32K	R/C	该位表示 32K 晶振失效中断状态。当 EXIST_32K 状态位从 1 变成 0 的时候，该状态位置 1。 当该位为 1 时，表示 32K 晶体被去除或损坏。 写 1 清零该位。	0x0

### 6.5.5 ANA\_CTRL 寄存器

表 6-8 ANA\_CTRL 寄存器

位	名称	类型	描述	默认值
26	PDNS2	R/W	该位控制 VDDALARM 为 0（同时考虑 PDNS 的设置）时进入深睡眠的行为。 0: 当 VDDALARM 为 0 时不会进入深睡眠。当 VDDALARM 为 1 时，系统可以进入深睡眠模式，并且当 VDDALARM 变为 0 时，系统自动从深睡眠唤醒。 1: VDDALARM 不作为深睡眠条件。	0x0
6	PDNS	R/W	该位控制 VDCINDROP 为 0（同时考虑 PDNS2 的设置）时进入深睡眠的行为。 0: VDCINDROP 为 0 时不会进入深睡眠。当 VDCINDROP 为 1 时，系统可以进入深睡眠，并且当 VDCINDROP 变为 0 时，系统自动从深睡眠唤醒。 1: VDCINDROP 的值，不作为进入深睡眠条件。	0x0

### 6.5.6 PMU\_RAMx 寄存器

表 6-9 PMU\_RAMx 寄存器

位	Name	Type	Description	Default
---	------	------	-------------	---------

31:0	RAM	R/W	PMU 控制器中嵌入了一个 256 字节(64x32)的 SRAM。该 RAM 可在深睡眠模式下保留数据。这些端口仅允许字访问。	--
------	-----	-----	--	----

## 6.6 复位

芯片由一种工作模式转为另一种工作模式时，某些硬件模块将自动复位，下表显示每个模块的复位详情。

表 6-10 各个模块在不同复位及睡眠唤醒后的复位情况

模块	外部复位	上电复位	看门狗复位	M0 软复位		退出 RTC only 模式	唤醒前模式		
				mode	管脚		深睡眠	浅睡眠	IDLE
Cortex-M0	√	√	√	√	√	√	√	-	-
系统 SRAM	-	-	-	-	-	数据丢失	数据丢失	-	-
掉电保持 SRAM	-	-	-	-	-	数据丢失	-	-	-
PMU(IOA)	√	√	√	-	√	√	-	-	-
WDT	√	√	√	-	√	√	-	-	-
RTC	√*	√*	√*	-	√*	√*	-	-	-
UART32K0	√	√	√	-	√	√	-	-	-
UART32K1	√	√	√	-	√	√	-	-	-
模拟控制器	√	√	√	-	√	√	-	-	-
LCD	√	√	√	√	√	√	√	-	-
GPIO(IOB~IOF)	√	√	√	√	√	√	√	-	-
MISC2	√	√	√	√	√	√	√	-	-
MISC	√	√	√	√	√	√	√	√	-
I <sup>2</sup> C	√	√	√	√	√	√	√	√	-
SPI	√	√	√	√	√	√	√	√	-
UART	√	√	√	√	√	√	√	√	-
ISO7816	√	√	√	√	√	√	√	√	-
TIMER	√	√	√	√	√	√	√	√	-
PWM	√	√	√	√	√	√	√	√	-
DMA	√	√	√	√	√	√	√	√	-
SRAM 控制器	√	√	√	√	√	√	√	√	-
FLASH 控制器	√	√	√	√	√	√	√	√	-

注意 (√\*): RTC 带写保护寄存器不会被上电复位、外部复位、看门狗复位这三种复位源复位，其他的 RTC 寄存器均可以被这三种唤醒源复位。

不同模式下，时钟产生模块的使能或禁止由硬件或软件控制，下表显示不同模式下的时钟状态。

表 6-11 在不同模式下的各个时钟的工作状态

时钟源	电源模式				
	RTC only	深睡眠	浅睡眠	IDLE	工作

6.5536M RC	OFF	OFF	OFF	由 RCHPD 控制
6.5536M XTAL	OFF	OFF	OFF	由 XOHPDN 控制
PLLH	OFF	OFF	OFF	由 PLLHPDN 控制
PLLL	OFF	OFF	OFF	由 PLLLPDN 控制
32K RC	ON	ON	ON	ON
32K XTAL	ON	ON	ON	ON

时钟源	退出 RTC only 模式	不同模式唤醒		
		深睡眠	浅睡眠	IDLE
6.5536M RC	OFF	ON	由 RCHPD 控制	
6.5536M XTAL	OFF	OFF	由 XOHPDN 控制	
PLLH	OFF	OFF	由 PLLHPDN 控制	
PLLL	OFF	OFF	由 PLLLPDN 控制	
32K RC	ON	ON	ON	
32K XTAL	ON	ON	ON	

### 6.6.1 片外复位

片外复位引脚(EXTRSTN)能复位 V85XXP 的大部分模块，具体可参考表 6-10。为避免外部噪声耦合进此脚，V85XXP 集成了一个去毛刺电路。下图显示去毛刺电路的工作原理。当 RTCCLK 预分频器设置为非零值时，抗尖峰时间将相应增加。例如，当 RTCCLK 预分频器设置为 1/4 时，去毛刺时间将增加到  $91 \times 4 = 364\mu\text{s}$ 。

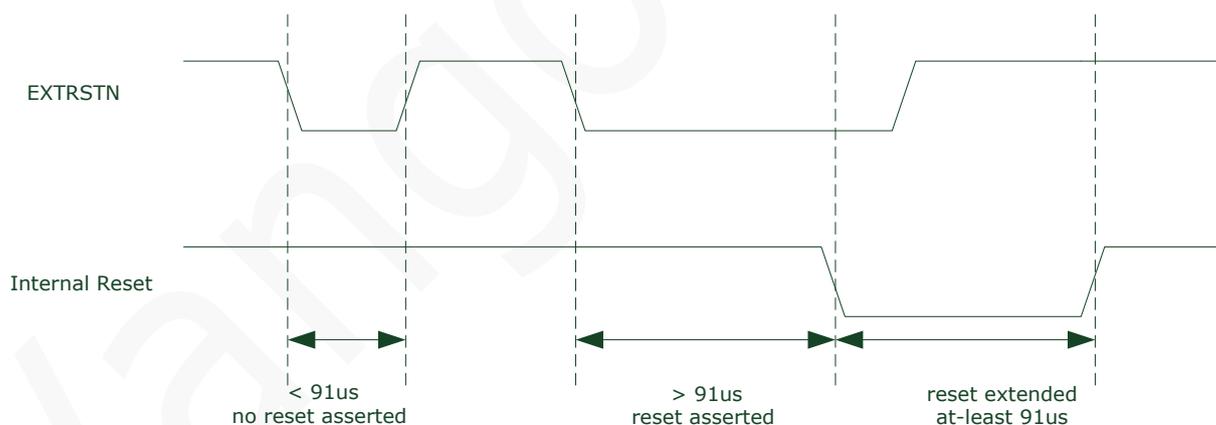


图 6-2 片外复位的去毛刺时序

### 6.6.2 看门狗复位

看门狗复位与上电复位等级相同，提供了芯片的内部复位机制。

### 6.6.3 上电复位

上电复位 (POR) 模块提供了芯片的内部复位机制，它与片外复位配合工作。

### 6.6.4 M0 软复位

M0 软复位指令可以复位 V85XXP 芯片的大多数模块，详情请参考表 6-10。

## 6.6.5 从浅睡眠/深睡眠/IDLE 下唤醒

任何的中断都能将芯片从 IDLE 状态唤醒，其中一些中断能从浅睡眠状态唤醒芯片，一些中断能从深睡眠状态唤醒芯片。

表 6-12 中断源

序列	向量地址	中断序号	描述	外设事件使能位	外设事件标记位	唤醒源	
						深睡眠	浅睡眠
NMI	00000008h	-14	NMI				
HardFault	0000000Ch	-13	HardFault				
SVCall	0000002Ch	-5	SVCall				
PendSV	00000038h	-2	PendSV				
SysTick	0000003Ch	-1	SysTick				
PMU	00000040h	0	IOA0~15	PMU_CONTROL.0a andPMU_IOAWKUEN .0~15	PMU_IOAINTSTS.0~ 15	√	√
			32K 晶振失效	PMU_CONTROL.2	PMU_STS.0	√	√
			6M 晶振失效	PMU_CONTROL.3	PMU_STS.1		
RTC	00000044h	1	ITV 和 SITV 中断	RTC_INTEN.0	RTC_INTSTS.0	√	√
			非法时间格式	RTC_INTEN.1	RTC_INTSTS.1	√	√
			多秒中断	RTC_INTEN.2	RTC_INTSTS.2	√	√
			多分钟中断	RTC_INTEN.3	RTC_INTSTS.3	√	√
			多小时中断	RTC_INTEN.4	RTC_INTSTS.4	√	√
			午夜（00:00） 中断	RTC_INTEN.5	RTC_INTSTS.5	√	√
			32K 计数器中断	RTC_INTEN.6	RTC_INTSTS.6	√	√
			对 CE 寄存器的 非法写入	RTC_INTEN.8	RTC_INTSTS.8		
			闹钟中断	RTC_INTEN.10	RTC_INTSTS.10	√	√
U32K0~ 1	00000048h 0000004Ch	2\3	接收到数据	U32Kx_CTRL1.0	U32Kx_STS.0	√	√
			接收奇偶校验错 误	U32Kx_CTRL1.1	U32Kx_STS.1	√	√
			接收缓存溢出	U32Kx_CTRL1.2	U32Kx_STS.2	√	√
I <sup>2</sup> C	00000050h	4	I <sup>2</sup> C 串行中断	I2C_CTRL2.0	I2C_CTRL.3		
SPI1~3	00000054h \000000ACh \000000B0h	5\27\ 28	SPI 发送	SPIx_TXSTS.14	SPIx_TXSTS.15		
			SPI 接收	SPIx_RXSTS.14	SPIx_RXSTS.15		
UART0~	00000058h	6\7\8	接收中断	UARTx_CTRL.3	UARTx_INTSTS.1		

5	~0000006Ch	\9\10 \11	发送溢出	UARTx_CTRL.4	UARTx_INTSTS.2		
			接收溢出	UARTx_CTRL.5	UARTx_INTSTS.3		
			接收奇偶校验错误	UARTx_CTRL.7	UARTx_INTSTS.4		
			发送完成	UARTx_CTRL.8	UARTx_INTSTS.5		
ISO781 60~1	00000070h ~00000074h	12\13	接收错误中断	ISO7816x_CFG.2	ISO7816x_INFO.2		
			接收中断	ISO7816x_CFG.5	ISO7816x_INFO.5		
			发送中断	ISO7816x_CFG.6	ISO7816x_INFO.6		
			接收溢出	ISO7816x_CFG.7	ISO7816x_INFO.7		
			传输重试错误中断	ISO7816x_CFG.8	ISO7816x_INFO.8		
Timer0 ~3	00000078h ~00000084h	14\15 \16\17	定时器溢出	TMRx_CTRL.3	TMRx_INTSTS.0		
PWM0~ 3	00000088h ~00000094h	18\19 \20\21	PWM 定时器溢出	PWMx_CTL.1	PWMx_CTL.0		
			捕获/比较 0	PWMx_CCTL0.4	PWMx_CCTL0.0		
			捕获/比较 1	PWMx_CCTL1.4	PWMx_CCTL1.0		
			捕获/比较 2	PWMx_CCTL2.4	PWMx_CCTL2.0		
DMA	00000098h	22	通道 0 包结束	DMA_IE.0	DMA_STS.4		
			通道 1 包结束	DMA_IE.1	DMA_STS.5		
			通道 2 包结束	DMA_IE.2	DMA_STS.6		
			通道 3 包结束	DMA_IE.3	DMA_STS.7		
			通道 0 帧结束	DMA_IE.4	DMA_STS.8		
			通道 1 帧结束	DMA_IE.5	DMA_STS.9		
			通道 2 帧结束	DMA_IE.6	DMA_STS.10		
			通道 3 帧结束	DMA_IE.7	DMA_STS.11		
			通道 0 数据丢弃	DMA_IE.8	DMA_STS.12		
			通道 1 数据丢弃	DMA_IE.9	DMA_STS.13		
			通道 2 数据丢弃	DMA_IE.10	DMA_STS.14		
			通道 3 数据丢弃	DMA_IE.11	DMA_STS.15		
FLASH	0000009Ch	23	校验和错误	FLASH_CTRL.2	FLASH_INTSTS.0		
ANA	000000A0h	24	手动 ADC 转换完成	ANA_INTEN.0	ANA_INTSTS.0		
			自动 ADC 转换完成	ANA_INTEN.1	ANA_INTSTS.1		

		CMP1 的上升和下降沿	ANA_INTEN.2	ANA_INTSTS.2	√	√
		CMP2 的上升和下降沿	ANA_INTEN.3	ANA_INTSTS.3	√	√
		VDDALARM 的上升和下降沿	ANA_INTEN.7	ANA_INTSTS.7	√	√
		VDCIN 的上升和下降沿	ANA_INTEN.8	ANA_INTSTS.8	√	√
		AVCCLV 的上升和下降沿	ANA_INTEN.10	ANA_INTSTS.10	√	√
		VDCINDROP=0 同时进入了浅睡眠/深睡眠状态	ANA_INTEN.11	ANA_INTSTS.11	√	√
		ANA_REGx 错误	ANA_INTEN.12	ANA_INTSTS.12	√	√
		TADC 变化超过阈值	ANA_INTEN.13	ANA_INTSTS.13	√	√
		ADC 值超过阈值	ANA_INTEN.14~21	ANA_INTSTS.14~21		

## 6.7 工作模式的转换

下图描述了 V85XXP 各种工作模式的转换条件和状态。

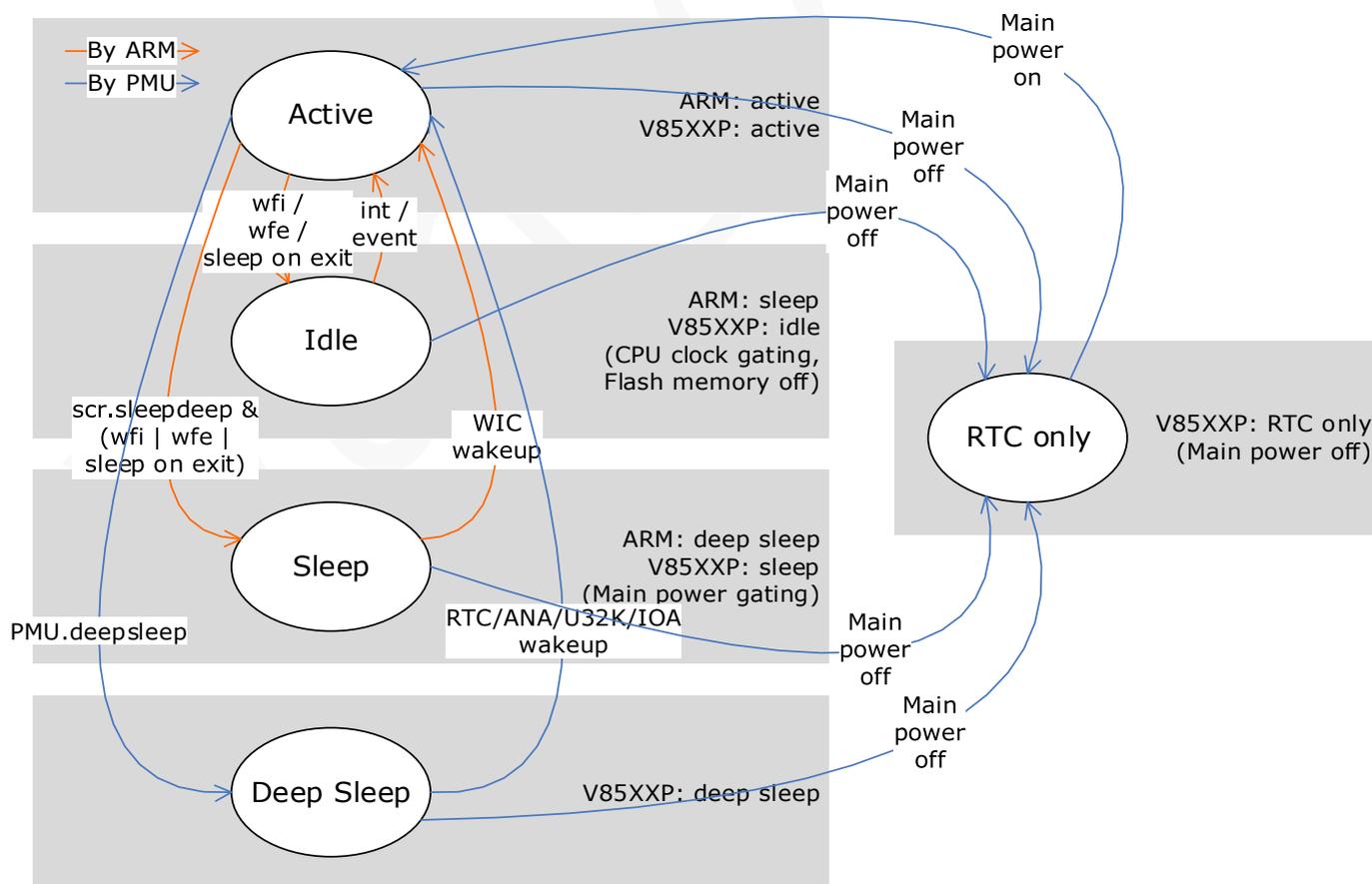


图 6-3 V85XXP 工作模式转换

根据上图，IDLE 模式或浅睡眠模式由 CPU 直接控制，用户只需要执行相应的指令（WFI 或 WFE）即可进入这两种模式。PMU 控制器控制深睡眠模式的进出，并使该模式在唤醒后总能返回到工作模式。进入深睡眠模式之前必须设置相应的唤醒源，这部分须由软件细致处理。以下显示了每种模式下的唤醒源。

**注意：**

- (1) 只有在 MODE=1 时，MCU 才可以进入深睡眠或浅睡眠模式；当 MODE=0 时，MCU 执行进入浅睡眠的指令后，会进入 IDLE 模式；进入深睡眠之前，应该首先配置 PDNS (ANA\_CTRL bit6)、PDNS2(ANA\_CTRL bit24)，或者检查 VDCINDROP 和 VDDALARM 状态，详情请参考 ANA\_CTRL 寄存器。
- (2) 当 MCU 从浅睡眠唤醒后，MCU 会自动切换系统时钟到进入浅睡眠之前的设置。例如，浅睡眠之前系统时钟为 PLLL，浅睡眠唤醒后，硬件会自动切换系统时钟到 PLLL。当 MCU 从深睡眠唤醒后，硬件会自动开启 RCH 作为系统时钟。IDLE 模式前后，硬件不会自动改变时钟状态。
- (3) 在浅睡眠唤醒时，PLLL 锁定时间约为 1ms，PLLH 锁定时间为 15 $\mu$ s。

**表 6-13 每一种工作模式下的唤醒源**

序号	唤醒源	唤醒前模式		
		深睡眠	浅睡眠	IDLE
Event	-	-	-	-
NMI	SRAM 奇偶校验错误	-	-	√
0	PMU(ExtInt)	√	√	√
1	RTC	√	√	√
2	UART32K0	√	√	√
3	UART32K1	√	√	√
4	I <sup>2</sup> C	-	-	√
5	SPI1	-	-	√
6	UART0	-	-	√
7	UART1	-	-	√
8	UART2	-	-	√
9	UART3	-	-	√
10	UART4	-	-	√
11	UART5	-	-	√
12	ISO78160	-	-	√
13	ISO78161	-	-	√
14	TIMER0	-	-	√
15	TIMER1	-	-	√
16	TIMER2	-	-	√
17	TIMER3	-	-	√
18	PWM0	-	-	√
19	PWM1	-	-	√
20	PWM2	-	-	√
21	PWM3	-	-	√
22	DMA	-	-	√
23	FLASH	-	-	√
24	ANA	√	√	√
27	SPI2	-	-	√

28	SPI3	-	-	√
----	------	---	---	---

## 6.8 应用笔记

### 6.8.1 外部 IO 唤醒

所有 16 个 IOA 都可以将系统从浅睡眠或深睡眠模式唤醒。V85XXP 中有一个去毛刺电路，防止信号上的毛刺意外唤醒系统。

下图是去毛刺电路示意图。去毛刺时间为 4 个 RTCCLK 时钟周期，RTCCLK 不分频时 (RTC\_PSCA = 0)，抗干扰时间约为 122 $\mu$ s。当 RTCCLK 4 分频时 (RTC\_PSCA = 1)，去毛刺时间会增加到 122\*4=488 $\mu$ s。去毛刺电路仅对浅睡眠、深睡眠模式的 IOA 边沿唤醒信号有效；对正常中断和浅睡眠、深睡眠的 IOA 电平唤醒信号无效。

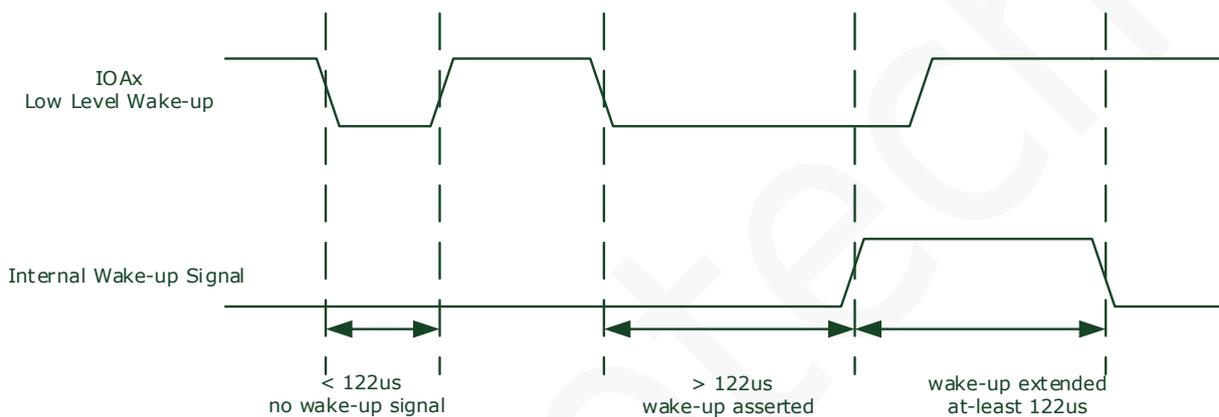


图 6-4 外部唤醒示意图

## 6.8.2 进入深睡眠模式的流程

下图显示了进入深睡眠模式的步骤。

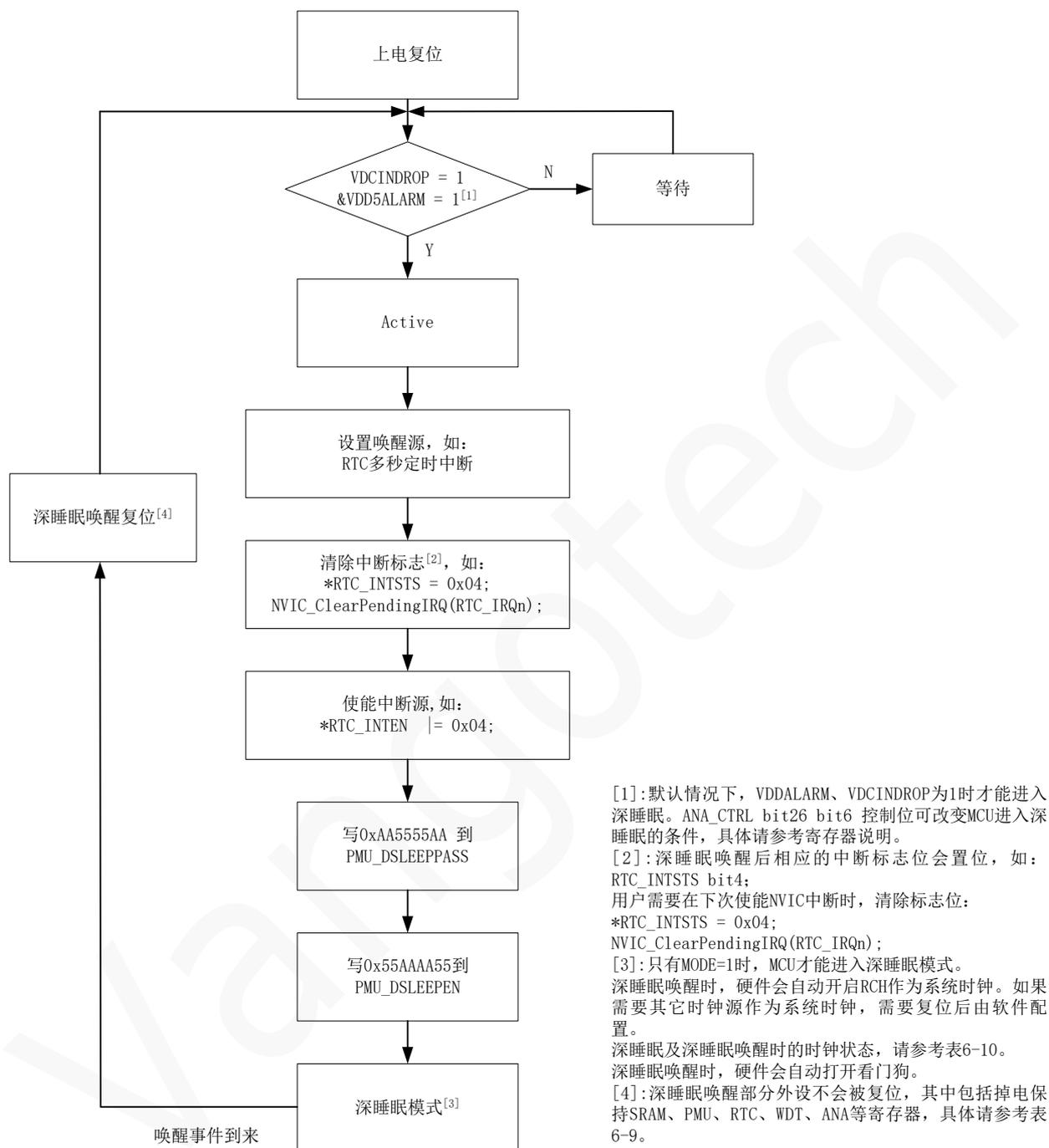
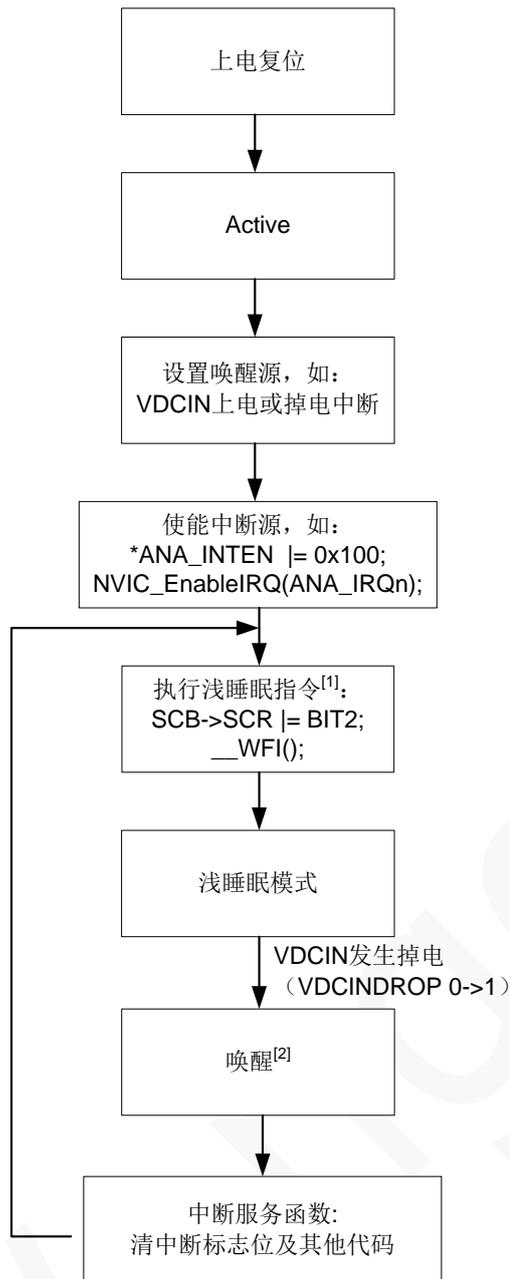


图 6-5 进入深睡眠的流程

## 6.8.3 进入浅睡眠的流程

下图描述了一个进入浅睡眠的步骤示例。



[1]:只有在MODE=1时，MCU才可以浅睡眠模式；当MODE=0时，MCU执行浅睡眠的指令后，会进入Idle模式。

[2]:浅睡眠唤醒后会复位部分外设，包括UART、SPI、I2C等；LCD、GPIO、ANA等寄存器不会被复位，具体请参考表6-9。

浅睡眠唤醒后，MCU会自动切换系统时钟到进入浅睡眠之前的设置。例如，浅睡眠之前系统时钟为PLLL，浅睡眠唤醒后，硬件会自动切换系统时钟到PLLL。

浅睡眠及浅睡眠唤醒后的时钟状态，请参考表6-10。浅睡眠唤醒会自动打开看门狗。

图 6-6 进入浅睡眠的流程

#### 6.8.4 进入 IDLE 的流程

下图描述了进入 IDLE 模式的流程。

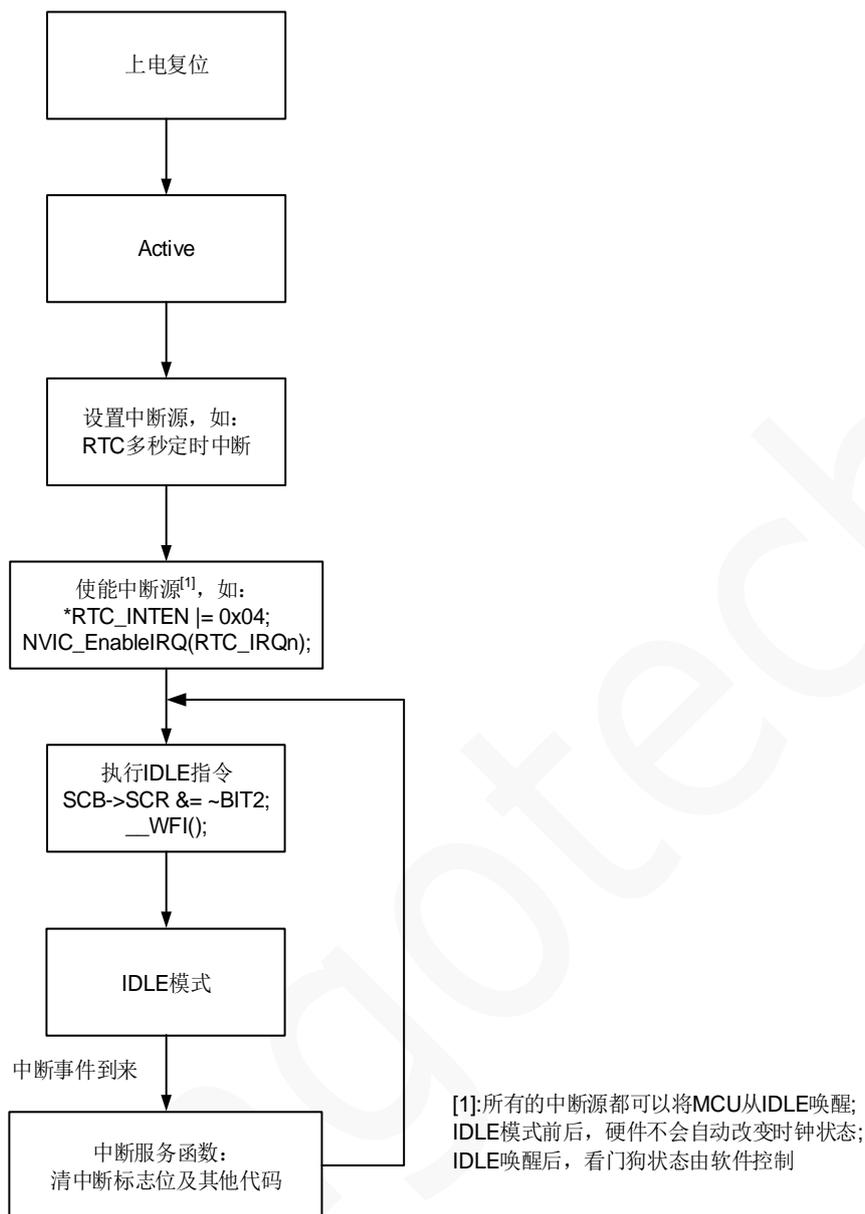


图 6-7 进入 IDLE 的流程

## 第7章 时钟系统

### 7.1 简介

时钟系统的相关设置在系统复位或深睡眠唤醒后会被复位，从深睡眠唤醒后用户需要手动恢复与时钟相关的寄存器设置。

在 V85XXP 中，有四个时钟源：

--32K RC 振荡电路，用于产生 32K RC 时钟。该电路始终运行，且频率可以细调。

--32K 晶体振荡电路，用于产生 32.768K 的晶振时钟。32.768K 晶振时钟始终开启。32.768K 晶体振荡时钟由 32K RC 时钟监测，当其停止振荡时，32K RC 时钟将自动替换它，并且在晶振恢复工作之前，有检测电路始终监测晶振电路。

--6M RC 时钟振荡电路，用于产生 6.5536M RC 时钟。用户可以通过寄存器设置停止该电路的运行，该电路在芯片复位后自动开始运行。

--6M 晶体振荡电路，用于产生 6.5536M 晶振时钟。上电时默认关闭 6.5536M 晶振时钟，用户需手动开启 6.5536M 晶振时钟输出。

在 V85XXP 中，有三个时钟域：

--HCLK 由 6.5536M 晶振时钟、6.5536M RC 时钟、32.768K 晶振时钟或 32K RC 时钟产生或它们的倍频产生，HCLK 为 CPU\FLASH\SRAM\DMA\GPIO\LCD\CRYPT 提供时钟。

--PCLK 由 HCLK 分频产生的，PCLK 为低速外设提供时钟。

--RTCCLK 由 32.768K 晶振时钟或 32K RC 时钟产生，为 RTC\WDT\UART32K\LCD 提供时钟。

表 7-1 不同工作模式下的时钟模块开关

时钟源	电源模式				
	RTC only	深睡眠	浅睡眠	IDLE	工作
6.5536MRC	OFF	OFF	OFF	由 RCHPD 控制	
6.5536MXTAL	OFF	OFF	OFF	由 XOHPDN 控制	
PLLH	OFF	OFF	OFF	由 PLLHPDN 控制	
PLLL	OFF	OFF	OFF	由 PLLLPDN 控制	
32K RC	ON	ON	ON	ON	
32K XTAL	ON	ON	ON	ON	
时钟源	退出 RTC only 模式	不同模式唤醒			IDLE
		深睡眠	浅睡眠		
6.5536MRC	OFF	ON	由 RCHPD 控制		
6.5536MXTAL	OFF	OFF	由 XOHPDN 控制		
PLLH	OFF	OFF	由 PLLHPDN 控制		
PLLL	OFF	OFF	由 PLLLPDN 控制		
32K RC	ON	ON	ON		
32K XTAL	ON	ON	ON		

### 7.2 特点

--子模块的时钟控制；

--AHBCLK 和 APBCLK 的时钟分频；

--实时晶体检测:

晶振检测电路工作时钟为 32K RC。实时检测高频晶体（默认关闭，需要手动开启）与低频晶体状态。当 6.5536M XTAL 停止振荡，6M RC 将自动替换 6.5536M XTAL，并且在 6.5536M XTAL 恢复之后，自动使用 6.5536M XTAL；当 32.768K XTAL 停止振荡，32K RC 将自动替换 32.768K XTAL，并且在 32.768K XTAL 恢复之后，自动使用 32.768K XTAL。

注意：HCLK 指的是 AHB 时钟，也叫做 AHCLK；同样，PCLK 指的是 APB 时钟，也叫做 APCLK。

## 7.3 功能框图

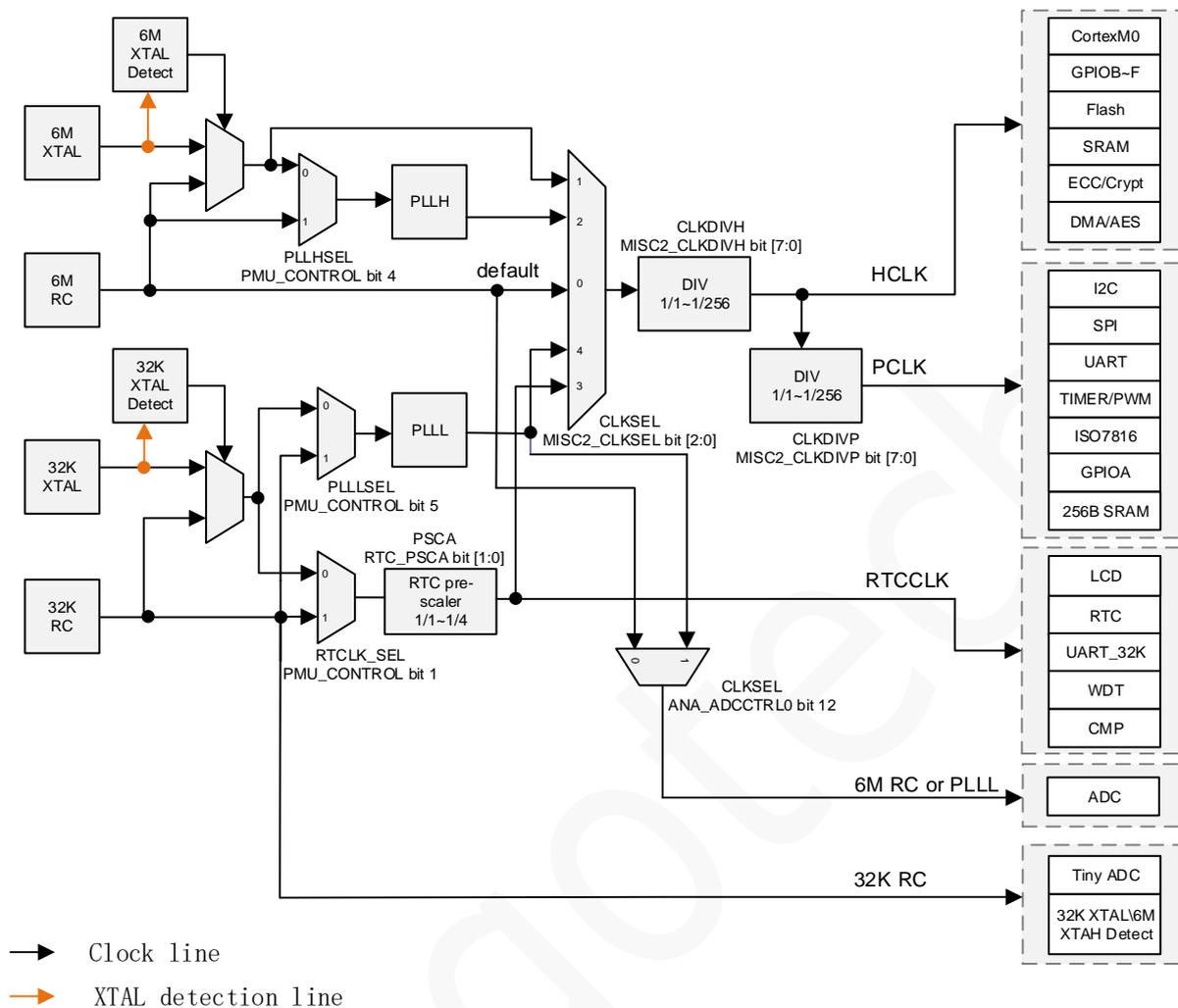


图 7-1 V85XXP 时钟系统功能框图

## 7.4 寄存器地址

表 7-2 ANA 控制寄存器(ANA 基地址: 0x40014200)

名称	类型	地址	描述	默认值
ANA_REG3	R/W	0x000C	模拟寄存器 3	0x00
ANA_REG4	R/W	0x0010	模拟寄存器 4	0x00
ANA_REG9	R/W	0x0024	模拟寄存器 9	0x00
ANA_REGB	R/W	0x002C	模拟寄存器 11	来自 FLASH
ANA_REGC	R/W	0x0030	模拟寄存器 12	来自 FLASH
ANA_CMPOUT	R	0x0054	比较器结果寄存器	0x0030

表 7-3 PMU 寄存器地址(PMU 基地址: 0x40014000)

名称	类型	地址	描述	默认值
PMU_CONTROL	R/W	0x0008	PMU 控制寄存器	0x0000
PMU_STS	R/C	0x000C	PMU 状态寄存器	0x0000074

表 7-4 MISC2 控制寄存器(MISC2 基地址: 0x40013E00)

名称	类型	地址	描述	默认值
MISC2_CLKSEL	R/W	0x0004	时钟选择寄存器	0x0
MISC2_CLKDIVH	R/W	0x0008	AHB 时钟分频控制寄存器	0x00
MISC2_CLKDIVP	R/W	0x000C	APB 时钟分频控制寄存器	0x01
MISC2_HCLKEN	R/W	0x0010	AHB 时钟使能控制寄存器	0x1FF
MISC2_PCLKEN	R/W	0x0014	APB 时钟使能控制寄存器	0xFFFFFFFF

## 7.5 寄存器定义

### 7.5.1 ANA\_REG3 寄存器

表 7-5 ANA\_REG3 控制寄存器

位	名称	类型	描述	默认值
3	BGPPD <sup>*[1]</sup>	R/W	BGP 掉电控制。 0: 开启 BGP 1: 关闭 BGP	0
4	RCHPD	R/W	高频 RC 时钟 RCH(6.5536MRC)开关控制信号。 0: 开启 RCH 1: 关闭 RCH	0
5	PLLLPDN	R/W	PLLL(32768Hz 输入 PLL)开关控制信号。 0: 关闭 PLLL 1: 开启 PLLL	0
6	PLLHPDN	R/W	PLLH(6.5536MHz 输入 PLL)开关控制信号。 0: 关闭 PLLH 1: 开启 PLLH	0
7	XOHPDN	R/W	6.5536M 晶振时钟开关控制信号。 0: 关闭 XOH 1: 开启 XOH	0

<sup>\*[1]</sup>: RCH、PLL、ADC 与 BGP 有关系, 在开 RCH、PLL、ADC 之前, 必须开启 BGP。如果系统中没有使用 BGP, 可通过 BGPPD 位关闭 BGP; 如果 BGP 有被用到, 此时, 通过 BGPPD 不能关闭 BGP, 因为系统具有保护功能, 会重新开启 BGP。浅睡眠或深睡眠模式下, 不管 BGP 是否有被用到, 都可以关闭 BGP。

### 7.5.2 ANA\_REG4 寄存器

表 7-6 ANA\_REG4 控制寄存器

位	名称	类型	描述	默认值
7: 0	-	R/W	保留	默认值是 0x00, 必须配置为 0x01。

### 7.5.3 ANA\_REG9 寄存器

表 7-7 ANA\_REG9 控制寄存器



位	名称	类型	描述	默认值
2: 0	PLLLSEL[2: 0]	R/W	选择 PLLL 的时钟频率。 000: 26.2144MHz 001: 13.1072MHz 010: 6.5536MHz 011: 3.2768MHz 100: 1.6384MHz 101: 0.8192MHz 110: 0.4096MHz 111: 0.2048MHz	0
6: 3	PLLHSEL[3: 0]	R/W	PLLH 的时钟频率控制位。PLLH 时钟频率是外部 XOH 或内部 RCH 的倍数。 1000~1011: 保留 1100: 2x 输入时钟频率 1101: 2.5x 输入时钟频率 1110: 3x 输入时钟频率 1111: 3.5x 输入时钟频率 0000: 4x 输入时钟频率 0001: 4.5x 输入时钟频率 0010: 5x 输入时钟频率 0011: 5.5x 输入时钟频率 0100: 6x 输入时钟频率 0101: 6.5x 输入时钟频率 0110: 7x 输入时钟频率 0111: 7.5x 输入时钟频率	0

### 7.5.4 ANA\_REGB 寄存器

表 7-8 ANA\_REGB 控制寄存器

位	名称	类型	描述	默认值
4: 0	RCLTRIM[4:0]	R	32kHz RC 时钟频率微调。 00000~01111: 增加 4%/步长; 10000~11111: 减少 4%/步长	0

注意: 用户不要对寄存器 ANA\_REGx (x=B~E、10) 进行配置, 这几个寄存器的值会从 FLASH 自动加载。

### 7.5.5 ANA\_REGC 寄存器

表 7-9 ANA\_REGC 控制寄存器

位	名称	类型	描述	默认值
5: 0	RCHTRIM[5:0]	R	6.5536MHzRC 时钟频率微调。 000000~011111: 增加 1.25%/步长; 100000~111111: 减少 1.25%/步长。	0

注意: 用户不要对寄存器 ANA\_REGx (x=B~E、10) 进行配置, 这几个寄存器的值会从 FLASH 自动加载。

## 7.5.6 ANA\_CMPOUT 寄存器

表 7-10 ANA\_CMPOUT 寄存器

位	名称	类型	描述	默认值
1	LOCKL	R	PLLL 锁定状态，锁定时间大约是 1ms。 0: PLLL 未锁定。 1: PLLL 已锁定。	0x0
0	LOCKH	R	PLLH 锁定状态，锁定时间大约是 15 $\mu$ s。 0: PLLH 未锁定。 1: PLLH 已锁定。	0x0

## 7.5.7 PMU\_CONTROL 寄存器

表 7-11 PMU\_CONTROL 寄存器

位	名称	类型	描述	默认值
5	PLLL_SEL	R/W	低速 PLL 输入时钟源选择。 0: 32.768K XTAL 1: 32K RC	0x0
4	PLLH_SEL	R/W	高速 PLL 输入时钟源选择。 0: 6.5MHz RC 1: 6.5536MHz XTAL	0x0
3	INT_6M_EN	R/W	6.5536M XTAL 失效中断使能寄存器。该位用于控制输出到 CPU 的中断信号。该位设置为 1 后，如果 6.5536M 晶振被移除或损坏，则会向 CPU 发出中断。	0x0
2	INT_32K_EN	R/W	32.768K XTAL 失效中断使能寄存器。该位用于控制输出到 CPU 的中断信号。该位设置为 1 后，如果 32.768K 晶振被移除或损坏，则会向 CPU 发出中断。如果这个事件发生在睡眠期间，CPU 将被唤醒。	0x0
1	RTCCLK_SEL	R/W	RTC 时钟选择。 0: 32.768K XTAL 1: 32K RC	0x0

## 7.5.8 PMU\_STS 寄存器

表 7-12 PMU\_STS 寄存器

位	名称	类型	描述	默认值
3	EXIST_6M	R	6.5536M XTAL 存在状态寄存器。该位表示 6.5536M XTAL 存在或不存在。开启 6.5536M XTAL (ANA_REG3 的 BIT7 配置为 1) 后，该状态位才会刷新，否则保持之前状态。 0: 6.5536M 晶体不存在。 1: 6.5536M 晶体存在。	0x0
2	EXIST_32K	R	32.768K XTAL 存在状态寄存器。该位表示 32.768K XTAL 存在或不存在。 0: 32K 晶体不存在。	0x1

			1: 32K 晶体存在。	
1	INT_6M	R/C	该位表示 6.5536M XTAL 失效中断状态。当 EXIST_6M 状态位从 1 变成 0 的时候, 该状态位置 1。 当该位为 1 时, 表示 6.5536M 晶振被去除或损坏。 写 1 清零该位。	0x0
0	INT_32K	R/C	该位表示 32.768K XTAL 失效中断状态。当 EXIST_32K 状态位从 1 变成 0 的时候, 该状态位置 1。 当该位为 1 时, 表示 32.768K 晶体被去除或损坏。 写 1 清零该位。	0x0

## 7.5.9 MISC2\_CLKSEL 寄存器

表 7-13 MISC2\_CLKSEL 寄存器

位	名称	类型	描述	默认值
31: 3	-	-	保留	0
2: 0	CLKSEL	R/W	该寄存器用于控制 AHB 时钟源。 0: RCH (6.5M RC) 1: XOH (6.5536M XTAL) 2: PLLH 3: RTCCLK (由 PMU_CONTROL 寄存器中的 RTCCLK_SEL 控制) 4: PLLL 其它: 保留 在时钟选择到一个时钟源之前, 用户应首先通过设置 PMU_CONTROL 寄存器来使能相应的模块。	0x0

## 7.5.10 MISC2\_CLKDIVH 寄存器

表 7-14 MISC2\_CLKDIVH 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7: 0	CLKDIVH	R/W	该寄存器用于控制 AHB 时钟分频。 0: 时钟源除以 1 1: 时钟源除以 2 2: 时钟源除以 3 ... 255: 时钟源除以 256	0x00

## 7.5.11 MISC2\_CLKDIVP 寄存器

表 7-15 MISC2\_CLKDIVP 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7: 0	CLKDIVP	R/W	该寄存器用于控制 APB 时钟分频。 0: AHB 时钟除以 1 1: AHB 时钟除以 2 2: AHB 时钟除以 3 ... 255: AHB 时钟除以 256	0x01

## 7.5.12 MISC2\_HCLKEN 寄存器

表 7-16 MISC2\_HCLKEN 寄存器

位	名称	类型	描述	默认值
31: 9	-	-	保留	0
8: 0	HCLKEN	R/W	该寄存器用于控制每个 AHB 模块的时钟使能。对应的模块只有在其功能未被使用时才能被关闭。有关每个模块的详细信息，请参见下表。 0: 关闭 1: 开启	0x1FF

表 7-17 HCLK 时钟使能

位	模块	注意事项
0	--	--
1	Arbiter&BusMatrix	保留
2	FLASH 控制器	保留
3	SRAM 控制器	保留
4	DMA 控制器	保留
5	GPIO 控制器	保留
6	LCD 控制器	保留
7	--	--
8	CRYPT 控制器	保留

## 7.5.13 MISC2\_PCLKEN 寄存器

表 7-18 MISC2\_PCLKEN 寄存器

位	名称	类型	描述	默认值
31: 0	PCLKEN	R/W	该寄存器用于控制每个 APB 模块的时钟使能。对应的模块只有在其功能未被使用时才能被关闭。有关每个模块的详细信息，请参见下表。 0: 关闭 1: 开启	0xFFFFFFFF

表 7-19 各个模块的 PCLK 时钟使能

Bit	Module	Note
0	AHB2APB Bridge	保留
1	DMA 控制器	
2	I <sup>2</sup> C	
3	SPI1	
4	UART0	
5	UART1	
6	UART2	
7	UART3	

8	UART4	
9	UART5	
10	ISO78160	
11	ISO78161	
12	TIMER	
13	MISC1	
14	MISC2	
15	PMU	
16	RTC	
17	ANA	
18	U32K0	
19	U32K1	
20	保留	
21	SPI2	
22	SPI3	
31: 23	保留	

## 第8章 模拟控制器

### 8.1 简介

模拟控制器用于控制 V85XXP 的模拟功能。所有模拟相关控制，如 ADC、比较器和其他模拟监测都由本模块控制。模拟控制器总是处于上电状态，因此所有模拟设置在深睡眠模式下仍会保留，不会被复位。

### 8.2 特点

- 产生模拟监测中断/唤醒信号；
- 产生比较器中断；
- 比较器计数器；
- ADC 手动和自动采样模式；
- 产生 ADC 中断。

### 8.3 功能框图

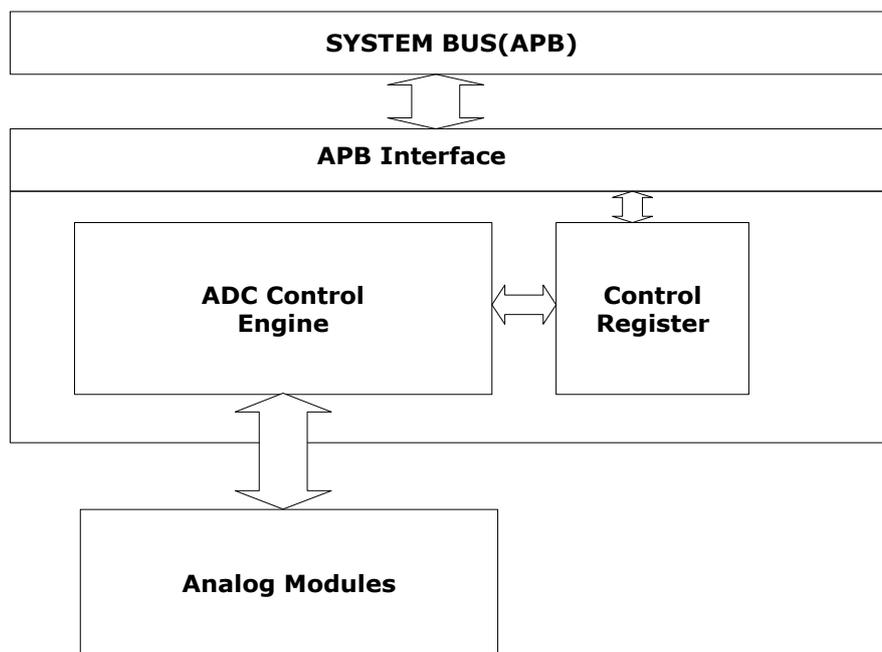


图 8-1 模拟控制器的功能框图

### 8.4 寄存器地址

表 8-1 ANA 控制寄存器(ANA 基地址: 0x40014200)

名称	类型	地址	描述	默认值
ANA_REG0	R/W	0x0000	模拟寄存器 0	0x00
ANA_REG1	R/W	0x0004	模拟寄存器 1	0x00
ANA_REG2	R/W	0x0008	模拟寄存器 2	0x00
ANA_REG3	R/W	0x000C	模拟寄存器 3	0x00
ANA_REG4	R/W	0x0010	模拟寄存器 4	0x00

ANA_REG5	R/W	0x0014	模拟寄存器 5	0x00
ANA_REG6	R/W	0x0018	模拟寄存器 6	0x00
ANA_REG7	R/W	0x001C	模拟寄存器 7	0x00
ANA_REG8	R/W	0x0020	模拟寄存器 8	0x00
ANA_REG9	R/W	0x0024	模拟寄存器 9	0x00
ANA_REGA	R/W	0x0028	模拟寄存器 10	0x00
ANA_REGB	R/W	0x002C	模拟寄存器 11	来自 FLASH
ANA_REGC	R/W	0x0030	模拟寄存器 12	来自 FLASH
ANA_REGD	R/W	0x0034	模拟寄存器 13	来自 FLASH
ANA_REGE	R/W	0x0038	模拟寄存器 14	来自 FLASH
ANA_REGF	R/W	0x003C	模拟寄存器 15	0x00
ANA_REG10	R/W	0x0040	模拟寄存器 16	来自 FLASH
ANA_REG11	W	0x0044	模拟寄存器 17	--
ANA_CTRL	R/W	0x0050	模拟控制寄存器	0x0000000
ANA_CMPOUT	R	0x0054	比较器结果寄存器	0x0030
ANA_ADCSTATE	R	0x005C	ADC 状态寄存器	--
ANA_INTSTS	R/C	0x0060	模拟中断状态寄存器	0x0000
ANA_INTEN	R/W	0x0064	模拟中断使能寄存器	0x0000
ANA_ADCCTRL0	R/W	0x0068	ADC 控制寄存器 0	0x00000000
ANA_CMPCTL	R/W	0x006C	比较器控制寄存器	0x00000000
ANA_ADCDATA0	R	0x0070	ADC 通道 0 数据寄存器	--
ANA_ADCDATA1	R	0x0074	ADC 通道 1 数据寄存器	--
ANA_ADCDATA2	R	0x0078	ADC 通道 2 数据寄存器	--
ANA_ADCDATA3	R	0x007C	ADC 通道 3 数据寄存器	--
ANA_ADCDATA4	R	0x0080	ADC 通道 4 数据寄存器	--
ANA_ADCDATA5	R	0x0084	ADC 通道 5 数据寄存器	--
ANA_ADCDATA6	R	0x0088	ADC 通道 6 数据寄存器	--
ANA_ADCDATA7	R	0x008C	ADC 通道 7 数据寄存器	--
ANA_ADCDATA8	R	0x0090	ADC 通道 8 数据寄存器	--
ANA_ADCDATA9	R	0x0094	ADC 通道 9 数据寄存器	--
ANA_ADCDATA10	R	0x0098	ADC 通道 10 数据寄存器	--
ANA_ADCDATA11	R	0x009C	ADC 通道 11 数据寄存器	--
ANA_ADCDATA12	R	0x009C	ADC 通道 12 数据寄存器	--
ANA_ADCDATA13	R	0x009C	ADC 通道 13 数据寄存器	--
ANA_ADCDATA14	R	0x009C	ADC 通道 14 数据寄存器	--
ANA_ADCDATA15	R	0x009C	ADC 通道 15 数据寄存器	--
ANA_CMPCNT1	R/C	0x00B0	比较器 1 计数器	0x00000000
ANA_CMPCNT2	R/C	0x00B4	比较器 2 计数器	0x00000000

ANA_MISC	R/W	0x00B8	模拟 misc 控制寄存器	0x00
ANA_ADCDOS	R	0x00C0	ADC 自校准寄存器 DOS	0
ANA_ADCDCPN	R	0x00CC	ADC 自校准寄存器 DCPN	0
ANA_ADCDCNM0	R	0x00D8	ADC 自校准寄存器 DCNM0	0
ANA_ADCDATADMA	R/W	0x00E0	DMA 的 ADC 数据源	--
ANA_CMPTHR	R/W	0x00E4	CMP1/CMP2 阈值寄存器	0x0000
ANA_ADCCTRL1	R/W	0x00E8	ADC 控制寄存器 1	0x000000C0
ANA_ADCCTRL2	R/W	0x00EC	ADC 控制寄存器 2	0x00000000
ANA_ADCDATATHD1_0	R/W	0x00F4	ADC 阈值控制寄存器	0x00000000
ANA_ADCDATATHD3_2	R/W	0x00F8	ADC 阈值控制寄存器	0x00000000
ANA_ADCDATATHD_CH	R/W	0x00FC	ADC 阈值状态寄存器	0x00000000

## 8.5 寄存器定义

### 8.5.1 ANA\_REGx 寄存器

表 8-2 ANA\_REGx 表

名称	Bit								地址
	7	6	5	4	3	2	1	0	
ANA_REG0	-	-	必须写 0x3		-	-	-	-	0x0000
ANA_REG1	ADCMODESEL	-	-	-	ADCSEL[3:0]				0x0004
ANA_REG2	-	-	CMP2REFSEL	CMP1REFSEL	CMP2SEL[1:0]		CMP1SEL[1:0]		0x0008
ANA_REG3	XOHPDN	PLLHPDN	PLLLPDN	RCHPD	BGPPD	CMP2PDN	CMP1PDN	-	0x000C
ANA_REG4	-		-		-	必须写 0x1			0x0010
ANA_REG5	-	AVCCLVDETPD	-	-	CMP2IT[1:0]		CMP1IT[1:0]		0x0014
ANA_REG6	BATRCDISC	BAT1DISC	-	VLCD<3:0>				LCDBMODE	0x0018
ANA_REG7	-	-	-	-	-	VDCINHYSSEL	-	-	0x001C
ANA_REG8	AVCCLDOPD	VDDPVDSSEL[2: 0]			-	-	DVCCSEL[1: 0]		0x0020
ANA_REG9	VDDDETPD	PLLHSEL[3: 0]			PLLSEL[2: 0]				0x0024
ANA_REGA	VDCINDETPD	-	-	-	-	-	必须写 1	-	0x0028
ANA_REGB <sup>[1]</sup>	-		RCLTRIM<4:0>					0x002C	
ANA_REGC <sup>[1]</sup>	-		RCHTRIM[5:0]					0x0030	
ANA_REGD <sup>[1]</sup>	-	-	-			-			0x0034
ANAREGE <sup>[1]</sup>	BKPWREN	-	-		必须写 1	-			0x0038
ANAREGF	ADTREF3SEL	ADTREF2SEL	ADTREF1SEL	ADTSEL	ADTPDN	AVCCOEN	BATRCDETEN	BAT1DETEN	0x003C
ANA_REG10 <sup>[1]</sup>	-	-			-	-			0x0040
ANA_REG11 <sup>[2]</sup>	-	-	-	-	-	-	-	-	0x0044

[1]这些寄存器(REGB~REGE 和 REG10) 是模拟模块的参数调节寄存器。当芯片发生复位后, 自动从 flash 中加载, 用户不要修改。

[2]ANA\_REG11 是只写寄存器, 不可读, 读出来的数据不确定。写寄存器后无法确认是否写成功。默认值是随机值, 用户操作的时候不可使用与操作和或操作。

### 8.5.2 ANA\_REG0 寄存器

表 8-3 ANA\_REG0 各个 bit 的功能描述

位	名称	类型	描述	默认值
7: 0	-	R/W	保留	默认值是 0, 必须配置为 0x30

### 8.5.3 ANA\_REG1 寄存器

表 8-4 ANA\_REG1 各个 bit 的功能描述

位	名称	类型	描述	默认值
6: 0	-	-	保留	0
7	ADCMODESEL	R/W	ADC 模式选择 当 RESDIV_CHx=1 时, 该控制位有效。 0: 当测试通道输入非交流周波 1: 当测试通道输入交流周波 (如±300mV 信号)	0

## 8.5.4 ANA\_REG2 寄存器

表 8-5 ANA\_REG2 各个 bit 的功能描述

位	名称	类型	描述	默认值
1: 0	CMP1SEL[1: 0]	R/W	CMP1 信号输入源选择 00: CMP1_P 与 REF 01: CMP1_N 与 REF 1*: CMP1_P 与 CMP1_N	0
3: 2	CMP2SEL[1: 0]	R/W	CMP2 信号输入源选择 00: CMP2_P 与 REF 01: CMP2_N 与 REF 1*: CMP2_P 与 CMP2_N	0
4	CMP1REFSEL	R/W	CMP1 的 REF 选择 0: VREF 1: BGPREF	0
5	CMP2REFSEL	R/W	CMP2 的 REF 选择 0: VREF 1: BGPREF	0
7: 6	-	-	保留	0

## 8.5.5 ANA\_REG3 寄存器

表 8-6 ANA\_REG3 各个 bit 的功能描述

位	名称	类型	描述	默认值
0	-	-	保留	0
1	CMP1PDN	R/W	CMP1 开关控制信号 0: 关闭 CMP1 1: 开启 CMP1	0
2	CMP2PDN	R/W	CMP2 开关控制信号 0: 关闭 CMP2 1: 开启 CMP2	0
3	BGPPD <sup>[1]</sup>	R/W	BGP 开关控制信号 0: 开启 BGP 1: 关闭 BGP	0
4	RCHPD	R/W	高频 RC 时钟电路 RCH (6.5536MRC) 开关控制信号 0: 开启 RCH 1: 关闭 RCH	0
5	PLLLPDN	R/W	PLLL (32768Hz 输入 PLL) 开关控制信号 0: 关闭 PLLL 1: 开启 PLLL	0

6	PLLHPDN	R/W	PLLH (6.5536MHz 输入 PLL) 开关控制信号 0: 关闭 PLLH 1: 开启 PLLH	0
7	XOHPDN	R/W	6.5536M 晶体振荡电路开关控制信号 0: 关闭 XOH 1: 开启 XOH	0

\*[1]: RCH、PLL、ADC 与 BGP 有关系, 在开 RCH、PLL、ADC 之前, 必须开启 BGP。如果系统中没有使用 BGP, 可通过 BGPPD 位关闭 BGP; 如果 BGP 有被用到, 此时, 通过 BGPPD 不能关闭 BGP, 因为系统具有保护功能, 会重新开启 BGP。浅睡眠或深睡眠模式下, 不管 BGP 是否有被用到, 都可以关闭 BGP。

## 8.5.6 ANA\_REG4 寄存器

表 8-7 ANA\_REG4 各个 bit 的功能描述

位	名称	类型	描述	默认值
7: 0	-	R/W	保留	默认值是 0x00, 必须配置为 0x01。

## 8.5.7 ANA\_REG5 寄存器

表 8-8 ANA\_REG5 各个 bit 的功能描述

位	名称	类型	描述	默认值
1: 0	CMP1IT[1: 0]	R/W	CMP1 偏置电流选择 00: 20nA; 01: 100nA; 1*: 500nA;	0
3: 2	CMP2IT[1: 0]	R/W	CMP2 偏置电流选择 00: 20nA; 01: 100nA; 1*: 500nA;	0
5: 4	-	-	保留	0
6	AVCCLVDETPD	R/W	AVCC 低电压检测开关控制 0: 开启; 1: 关闭	0
7	-	-	保留	0

## 8.5.8 ANA\_REG6 寄存器

表 8-9 ANA\_REG6 各个 bit 的功能描述

位	名称	类型	描述	默认值
0	LCDBMODE	R/W	LCD 偏置模式选择 0: 1/3 偏置 1: 1/4 偏置	0

4: 1	VLCD[3: 0]	R/W	当 VLCD=0: Default 当 VLCD=0~6: 调整幅度=+50mV*VLCD 当 VLCD=7~15: 调整幅度=-50mV*(VLCD-6)	
5	-	-	保留	0
6	BAT1DISC	R/W	给 BAT1 电池放电, 放电电阻是 1.7k, 放电电流是 Vbat1/1.7k。 0: 禁止 1: 使能从 BAT1 到 GND 的 1.7k 电阻	0
7	BATRTCDISC	R/W	给 BATRTC 电池放电, 放电电阻是 1.7k, 放电电流是 Vbatrtc/1.7k。 0: 禁止 1: 使能从 BATRTC 到 GND 的 1.7k 电阻	0

### 8.5.9 ANA\_REG7 寄存器

表 8-10 ANA\_REG7 各个 bit 的功能描述

位	名称	类型	描述	默认值
7: 3	-	R/W	保留	默认值为 0, 必须配置为 0x10。
2	VDCINHYSSEL	R/W	VDCIN 管脚电压检测回差选择 0: 100mV 1: 200mV	0
1: 0	-	R/W	保留	0

### 8.5.10 ANA\_REG8 寄存器

表 8-11 ANA\_REG8 各个 bit 的功能描述

位	名称	类型	描述	默认值
1: 0	DVCCSEL[1: 0]	R/W	DVCC 电压调节。 00: +0.0V 01: +0.1V 10: -0.1V 11: +0.2V	0
3: 2	-	-	保留	0
6: 4	VDDPVDSEL[2:0]	R/W	VDD 电源的低电压监测阈值设置寄存器, 设置将影响 VDDALARM 的状态。 000: 4.5V 001: 4.2V 010: 3.9V 011: 3.6V 100: 3.2V 101: 2.9V 110: 2.6V	0

			111: 2.3V	
7	AVCCLDOPD	R/W	AVCC LDO 控制，控制 LDO 的开启和关闭。 0: 开启 LDO，AVCC 引脚电压为 3.3V。 1: 关闭 LDO，AVCC 引脚通过一个开关连接到 VDD，开关内阻请参考表 1-4。AVCCLD0 只能在供电电压低于 3.6V 时关闭。	0

### 8.5.11 ANA\_REG9 寄存器

表 8-12 ANA\_REG9 各个 bit 的功能描述

位	名称	类型	描述	默认值
2: 0	PLLLSEL[2: 0]	R/W	PLLL 频率选择。 000: 26.2144MHz 001: 13.1072MHz 010: 6.5536MHz 011: 3.2768MHz 100: 1.6384MHz 101: 0.8192MHz 110: 0.4096MHz 111: 0.2048MHz	0
6: 3	PL LHSEL[3: 0]	R/W	PLLH 频率选择。PLLH 的频率是外部 XOH 或内部 RCH 的倍数。 1000~1011: 保留 1100: 2x 输入时钟频率 1101: 2.5x 输入时钟频率 1110: 3x 输入时钟频率 1111: 3.5x 输入时钟频率 0000: 4x 输入时钟频率 0001: 4.5x 输入时钟频率 0010: 5x 输入时钟频率 0011: 5.5x 输入时钟频率 0100: 6x 输入时钟频率 0101: 6.5x 输入时钟频率 0110: 7x 输入时钟频率 0111: 7.5x 输入时钟频率	0
7	VDDDETPD	R/W	关闭 VDD 电压监测模块，该模块由 VDD 供电。 0: 开启，不可以由硬件控制。 1: 关闭，该模块可以由硬件开启，详见 ANA_CMPCTL 的 bit22~23。	0

### 8.5.12 ANA\_REGA 寄存器

表 8-13 ANA\_REGA 各个 bit 的功能描述

位	名称	类型	描述	默认值
6: 0	-	R/W	保留，必须配置为 0x02	0x0
7	VDCINDETPD	R/W	VDCIN 监测模块开关。 0: 开启 VDCIN 监测 1: 关闭 VDCIN 监测	0

### 8.5.13 ANA\_REGB 寄存器

表 8-14 ANA\_REGB 各个 bit 的功能描述

位	名称	类型	描述	默认值
4: 0	RCLTRIM[4: 0]	R	32kHz RC 频率调节 00000~01111: 频率以 4%的步长增加; 10000~11111: 频率以 4%的步长减小	0
7: 5	-	-	保留	0

### 8.5.14 ANA\_REGC 寄存器

注意: 用户不要对寄存器 ANA\_REGx (x=B~E 和 10) 进行配置, 这几个寄存器的值会从 FLASH 自动加载。

表 8-15 ANA\_REGC 各个 bit 的功能描述

位	名称	类型	描述	默认值
5: 0	RCHTRIM[5: 0]	R	6.5536MHz RC 频率调节 000000~011111: 频率以 1.25%的步长增加; 100000~111111: 频率以 1.25%的步长减小	0
7: 6	-	-	保留	0

注意: 用户不要对寄存器 ANA\_REGx (x=B~E 和 10) 进行配置, 这几个寄存器的值会从 FLASH 自动加载。

### 8.5.15 ANA\_REGD 寄存器

表 8-16 ANA\_REGD 各个 bit 的功能描述

位	名称	类型	描述	默认值
7: 0	-	-	保留	0

注意: 用户不要对寄存器 ANA\_REGx (x=B~E 和 10) 进行配置, 这几个寄存器的值会从 FLASH 自动加载。

### 8.5.16 ANA\_REGE 寄存器

表 8-17 ANA\_REGE 各个 bit 的功能描述

位	名称	类型	描述	默认值
6: 0	-	-	保留	0
7	BKPWREN	R	使能低功耗备用电源 (在进休眠时将 AVCC 模块关闭, 开启低功耗备用电源) 0: 关闭; (休眠时仍使用 AVCC) 1: 开启	0

注意: 用户不要对寄存器 ANA\_REGx (x=B~E 和 10) 进行配置, 这几个寄存器的值会从 FLASH 自动加载。

### 8.5.17 ANA\_REGF 寄存器

表 8-18 ANA\_REGF 各个 bit 的功能描述

位	名称	类型	描述	默认值
0	BAT1DETEN	R/W	使能 BAT1 连接到比较器 1 的 P 端输入	0

			0: 禁止 1: 使能	
1	BATRTCDETEN	R/W	使能 BATRTC 连接到比较器 2 的 P 端输入 0: 禁止 1: 使能	0
2	AVCCOEN	R/W	AVCC_OUT 引脚输出使能控制 0: 输出高阻 1: 允许 AVCC_OUT 管脚输出内部的 AVCC 模块的电源电压	0
3	ADTPDN	R/W	Tiny ADC 开关控制 0: 关闭 1: 开启	0
4	ADTSEL	R/W	Tiny ADC 信号输入源选择 0: 连接到 IOE6 1: 连接到 IOE7	0
5	ADTREF1SEL	R/W	ADTREF1 电压选择 0: 0.9V 1: 0.7V	0
6	ADTREF2SEL	R/W	ADTREF2 电压选择 0: 1.8V 1: 1.6V	0
7	ADTREF3SEL	R/W	ADTREF3 电压选择 0: 2.7V 1: 2.5V	0

### 8.5.18 ANA\_REG10 寄存器

表 8-19 ANA\_REG10 各个 bit 的功能描述

位	名称	类型	描述	默认值
7:0	-	-	保留	0

注意：用户不要对寄存器 ANA\_REGx (x=B~E 和 10) 进行配置，这几个寄存器的值会从 FLASH 自动加载。

### 8.5.19 ANA\_REG11 寄存器

表 8-20 ANA\_REG11 各个 bit 的功能描述

位	名称	类型	描述	默认值
7:0	-	-	保留，用户只能进行写 0 操作。	-

注意 ANA\_REG11 寄存器只能写 0 操作，读取内容和写入内容不相关，用户不可用或操作和与操作。

### 8.5.20 ANA\_CTRL 寄存器

表 8-21 ANA\_CTRL 寄存器

位	名称	类型	描述	默认值
31: 27	-	-	保留	0
26	PDNS2	R/W	该位控制 VDDALARM 为 0（同时考虑 PDNS 的设置）时进入深睡眠的行为。 0: 当 VDDALARM 为 0 时不会进入深睡眠。当 VDDALARM 为 1 时，系统可以进入深睡眠模式，并且当 VDDALARM 变为 0 时，系统自动从深睡眠唤醒。 1: VDDALARM 不作为深睡眠条件。	0x0
25: 24	-	-	保留	0
23: 22	CMP2DEB	R/W	比较器 2 消抖控制寄存器。 0: 不消抖。 1: 2 个 RTCCLK 时钟周期消抖。 2: 3 个 RTCCLK 时钟周期消抖。 3: 4 个 RTCCLK 时钟周期消抖。 当使能消抖时，只有当信号在 RTCCLK 时钟的多周期内没有变化时，输入信号才有效。此外，唤醒和中断的响应时间将被延迟，直到信号有效。该电路可以在所有模式下工作，包括浅睡眠和深睡眠模式。	0x0
21: 20	CMP1DEB	R/W	比较器 1 消抖控制寄存器。 0: 不消抖。 1: 2 个 RTCCLK 时钟周期消抖。 2: 3 个 RTCCLK 时钟周期消抖。 3: 4 个 RTCCLK 时钟周期消抖。 当使能消抖时，只有当信号在 RTCCLK 时钟的多周期内没有变化时，输入信号才有效。此外，唤醒和中断的响应时间将被延迟，直到信号有效。该电路可以在所有模式下工作，包括浅睡眠和深睡眠模式。	0x0
19: 16	-	-	保留	0
15: 8	RCHTGT	R/W	高频时钟 RCH 自动校正频率设置。这个寄存器用于存储用户设置的高频时钟目标频率值，如果用户要设置的目标频率值是 6.5536MHz 那么用户需要 在该寄存器中填入 $6553600/32768=200$ 。	0x0
7	-	-	保留	0
6	PDNS	R/W	该位控制 VDCINDROP 为 0（同时考虑 PDNS2 的设置）时进入深睡眠的行为。 0: VDCINDROP 为 0 时不会进入深睡眠。当 VDCINDROP 为 1 时，系统可以进入深睡眠，并且当 VDCINDROP 变为 0 时，系统自动从深睡眠唤醒。 1: 不管 VDCINDROP 的值，都能进入深睡眠	0x0
5: 4	-	-	保留	0
3: 2	CMP2SEL	R/W	该寄存器用于控制比较器 2 的计数器累加条件。 0: 关闭 1: 比较器 2 输出的上升沿	0x0

			2: 比较器 2 输出的下降沿 3: 比较器 2 输出的双边沿	
1: 0	CMP1SEL	R/W	该寄存器用于控制比较器 1 的计数器累加条件。 0: 关闭 1: 比较器 1 输出的上升沿 2: 比较器 1 输出的下降沿 3: 比较器 1 输出的双边沿	0x0

### 8.5.21 ANA\_CMPOUT 寄存器

表 8-22 ANA\_CMPOUT 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 14	TADCO	R	TinyADC 输出。	-
13: 11	-	-	保留	0
10	AVCCLV	R	AVCC 低电压状态位。电压回差是 20mV~30mV。 0: AVCC 高于 2.5V 1: AVCC 低于 2.5V	0x0
9	-	-	保留	0
8	VDCINDROP	R	VDCIN 掉电状态。 0: VDCIN 高于阈值电压，没有掉电。 1: VDCIN 低于阈值电压，发生掉电。	0x0
7	VDDALARM	R	电源 VDD 电压预警信号 VDDALARM 输出。 0: 电源 VDD 电压高于 VDDPVDSEL 设置的值。 1: 电源 VDD 电压低于 VDDPVDSEL 设置的值。	0x0
6: 4	-	-	保留	0
3	CMP2	R	比较器 2 输出。	0x0
2	CMP1	R	比较器 1 输出。	0x0
1	LOCKL	R	PLLL 锁定状态，锁定时间大约是 1ms。 0: PLLL 未锁定。 1: PLLL 已锁定。	0x0
0	LOCKH	R	PLLH 锁定状态，锁定时间大约是 15μs。 0: PLLH 未锁定。 1: PLLH 已锁定。	0x0

### 8.5.22 ANA\_ADCSTATE 寄存器

表 8-23 ANA\_ADCSTATE 寄存器

位	名称	类型	描述	默认值
31:5	-	-	保留	0x0

4	RESET	R	该位显示 ADC 接口中 RESET 的状态。	0x0
3	ADC_EN	R	该位显示 ADC 接口中 ADC_EN 的状态。	0x0
2:0	ADCSTATE	R	这些位显示 ADC 控制器状态机的状态。 0: 空闲 2: ADC 多通道自动转换 4: ADC 多通道手动转换 其它: 保留	0x0

### 8.5.23 ANA\_INTSTS 寄存器

表 8-24 ANA\_INTSTS 寄存器

位	名称	类型	描述	默认值
31: 22	-	-	保留	0
21	INTSTS21	R/C	ADC_THD3 数据大于 UPPER_THD3 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
20	INTSTS20	R/C	ADC_THD3 数据小于 LOWER_THD3 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
19	INTSTS19	R/C	ADC_THD2 数据大于 UPPER_THD2 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
18	INTSTS18	R/C	ADC_THD2 数据小于 LOWER_THD2 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
17	INTSTS17	R/C	ADC_THD1 数据大于 UPPER_THD1 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0

16	INTSTS16	R/C	ADC_THD1 数据小于 LOWER_THD1 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
15	INTSTS15	R/C	ADC_THD0 数据大于 UPPER_THD0 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
14	INTSTS14	R/C	ADC_THD0 数据小于 LOWER_THD0 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
13	INTSTS13	R/C	TADC 过阈值中断标志位。TADC 在上升或下降时, 并且相比上一周期的变化值大于阈值 (TADCTH), 中断标志位将会置位。 读 0: 未发生 读 1: 发生 写 0: 无效 写 1: 清除该位	0x0
12	INTSTS12	R/C	模拟控制寄存器 ANA_REGx 错误标记位。芯片内部自动对所有模拟寄存器 ANA_REGx 进行校验和校验和奇偶校验, 当外部干扰引起模拟寄存器 ANA_REGx 被错误改写时, 该标记位将置 1。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
11	INTSTS11	R/C	当 VDCINDROP 为 0, 进入睡眠模式中断标志位 (VDCIN 大于阈值)。当 VDCINDROP 为 0, 并且检测到系统进入睡眠的动作, 将置位中断标志位。用户可以使能中断使能位 (INTEN11), 使得系统可以从浅睡眠或深睡眠模式唤醒 (VDCINDROP 为 0)。 读 0: 未发生进入睡眠事件 (VDCINDROP=0) 读 1: 发生进入睡眠事件 (VDCINDROP=0) 写 0: 无影响 写 1: 清除该位	0x0
10	INTSTS10	R/C	AVCCLV 低电压中断标记位。 读 0: 未发生 读 1: 发生 写 0: 无影响	0x0

			写 1: 清除该位	
9	-	-	保留	0
8	INTSTS8	R/C	VDCINDROP 的中断标志, 该中断会在 VDCINDROP 上升和下降时产生。 读 0: 未发生 读 1: 发生 写 0: 无效 写 1: 清除该位	0x0
7	INTSTS7	R/C	VDDALARM 的中断标志, 该中断会在 VDDALARM 上升和下降时产生。 读 0: 未发生 读 1: 发生 写 0: 无效 写 1: 清除该位	0x0
6: 4	-	-	保留	0
3	INTSTS3	R/C	CMP2 的中断标志, 中断产生条件受 CMP2SEL 控制。 读 0: 未发生 读 1: 发生 写 0: 无效 写 1: 清除该位	0x0
2	INTSTS2	R/C	CMP1 的中断标志, 中断产生条件受 CMP1SEL 控制。 读 0: 未发生 读 1: 发生 CMP1 中断 写 0: 无效 写 1: 清除该位	0x0
1	INTSTS1	R/C	ADC 自动转换完成中断标记位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
0	INTSTS0	R/C	手动设置 ADC 转换完成中断标记位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0

## 8.5.24 ANA\_INTEN 寄存器

表 8-25 ANA\_INTEN 寄存器

位	名称	类型	描述	默认值
31: 22	-	-	保留	0
21	INTEN21	R/W	ADC_THD3 数据大于 UPPER_THD3 阈值中断使能控制。 0: 禁止 1: 使能	0x0
20	INTEN20	R/W	ADC_THD3 数据小于 LOWER_THD3 阈值中断使能控制。 0: 禁止 1: 使能	0x0
19	INTEN19	R/W	ADC_THD2 数据大于 UPPER_THD2 阈值中断使能控制。 0: 禁止 1: 使能	0x0
18	INTEN18	R/W	ADC_THD2 数据小于 LOWER_THD2 阈值中断使能控制。 0: 禁止 1: 使能	0x0
17	INTEN17	R/W	ADC_THD1 数据大于 UPPER_THD1 阈值中断使能控制。 0: 禁止 1: 使能	0x0
16	INTEN16	R/W	ADC_THD1 数据小于 LOWER_THD1 阈值中断使能控制。 0: 禁止 1: 使能	0x0
15	INTEN15	R/W	ADC_THD0 数据大于 UPPER_THD0 阈值中断使能控制。 0: 禁止 1: 使能	0x0
14	INTEN14	R/W	ADC_THD0 数据小于 LOWER_THD0 阈值中断使能控制。 0: 禁止 1: 使能	0x0
13	INTEN13	R/W	TADC 的中断和唤醒使能 (TADC 输出值的变化大于等于阈值)。 0: 禁止 TADC 中断和唤醒 1: 使能 TADC 中断和唤醒	0x0
12	INTEN12	R/W	模拟寄存器 ANA_REGx 状态错误中断唤醒使能控制位。 0: 禁止 1: 使能	0x0
11	INTEN11	R/W	当 VDCINDROP 为 0, 检测到进入睡眠模式的动作, 该事件的中断和唤醒使能控制。	0x0

			0: 当 VDCINDROP 为 0 时, 检测到进入睡眠, 禁止该中断或唤醒 1: 当 VDCINDROP 为 0 时, 检测到进入睡眠, 使能该中断或唤醒	
10	INTEN10	R/W	AVCCLV 中断和唤醒使能控制。 0: 禁止 AVCCLV 中断和唤醒 1: 使能 AVCCLV 中断和唤醒	0x0
9	-	-	保留	0
8	INTEN8	R/W	VDCINDROP 中断和唤醒使能控制。 0: 禁止 VDCIN 中断和唤醒 1: 使能 VDCIN 中断和唤醒	0x0
7	INTEN7	R/W	VDDALARM 中断和唤醒使能控制。 0: 禁止 VDDALARM 中断和唤醒 1: 使能 VDDALARM 中断和唤醒	0x0
6: 4	-	-	保留	0
3	INTEN3	R/W	CMP2 的中断和唤醒使能控制。 0: 禁止 CMP2 中断和唤醒 1: 使能 CMP2 中断和唤醒	0x0
2	INTEN2	R/W	CMP1 的中断和唤醒使能控制。 0: 禁止 CMP1 中断和唤醒 1: 使能 CMP1 中断和唤醒	0x0
1	INTEN1	R/W	ADC 自动转换完成的中断使能控制。 0: 禁止 ADC 自动转换中断 1: 使能 ADC 自动转换中断	0x0
0	INTEN0	R/W	ADC 手动转换完成的中断使能控制。 0: 禁止 ADC 手动转换中断 1: 使能 ADC 手动转换中断	0x0

## 8.5.25 ANA\_ADCCTRL0 寄存器

表 8-26 ANA\_ADCCTRL0 寄存器

位	名称	类型	描述	默认值
31	MTRIG	R/W	手动控制 ADC 转换。 写 0: 无影响 写 1: 使能一次手动 ADC 转换 读 0: 当前手动设置的 ADC 转换已完成 读 1: 当前手动设置的 ADC 转换未完成	0x0
30: 22	-	-	保留	0
21: 20	-	R/W	保留, 该寄存器位必须配置为 0x03。	0
19	STOP	R/W	强制停止 ADC 转换流程。向该位写入 1 并保持 1 个 ADCCLK 时间, 可停止当前的 ADC 转换。在下次 ADC 转换开始之前向该寄存器写入 0。	0x0
18: 16	AEN	R/W	ADC 自动转换使能控制寄存器。 0: 禁止 ADC 自动转换 1: ADC 自动转换由 RTC_ITV 和 RTC_SITV 触发 2: ADC 自动转换由 RTC_WKUSEC 触发 3: ADC 自动转换由 RTC_ALARM 触发 4: ADC 自动转换由定时器 0 的溢出触发 5: ADC 自动转换由定时器 1 的溢出触发 6: ADC 自动转换由定时器 2 的溢出触发 7: ADC 自动转换由定时器 3 的溢出触发	0x0
15: 13	-	-	保留	0
12	CLKSRCSEL	R/W	ADC 时钟源选择。当更改该寄存器时, 硬件逻辑会停止, 停止时间为 3 个 ADC 时钟, 此时任何 ADC 操作都会被硬件忽略。所以在此期间不要操作 ADC。 0: 6.5M RCH 1: 6.5M PLLL	0x0
11: 0	-	-	保留	0

## 8.5.26 ANA\_ADCDATAx 寄存器

表 8-27 ANA\_ADCDATAx 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	ADCDATAx	R	ADC 的转换结果数据寄存器。 DATA0: 存储 ADC 通道 0 转换结果 DATA1: 存储 ADC 通道 1 转换结果 ... DATA15: 存储 ADC 通道 15 转换结果	0x--

## 8.5.27 ANA\_CMPCNTx 寄存器

表 8-28 ANA\_CMPCNTx 寄存器

位	名称	类型	描述	默认值
31: 0	CNT	R/C	该寄存器用于存储比较器 x 的事件发生次数的计数器。 通过写 0 可以将该寄存器清零，但这个操作只有在 CMPxPDN=1 时有效 (x=1,2)。	0x0000000

## 8.5.28 ANA\_MISC 寄存器

表 8-29 ANA\_MISC 寄存器

位	名称	类型	描述	默认值
31: 6	-	-	保留	0x0
5: 4	TADCTH	R/W	TADC 阈值设置。该寄存器控制 TADC 产生中断的阈值。TADC 时钟选用内部 32K RC 作为时钟源采样外部信号。当 TADC 的中断使能，并且连续两个周期采样值的差值大于阈值，INTSTS13 就会置 1 并产生中断。	0x0
3: 0	-	-	保留	0x0

## 8.5.29 ANA\_ADCDOS 寄存器

表 8-30 ANA\_ADCDOS 寄存器

位	名称	类型	描述	默认值
31:9	-	-	保留	0
8:0	DOS	R	ADC 自校准寄存器 DOS	0

## 8.5.30 ANA\_ADCDCPN 寄存器

表 8-31 ANA\_ADCDCPN 寄存器

位	名称	类型	描述	默认值
31:27	DCPN4	R	ADC 自校准寄存器 DCPN4	
26:18	DCPN3	R	ADC 自校准寄存器 DCPN3	
17:9	DCPN2	R	ADC 自校准寄存器 DCPN2	
8:0	DCPN1	R	ADC 自校准寄存器 DCPN1	

## 8.5.31 ANA\_ADCDCNM0 寄存器

表 8-32 ANA\_ADCDCNM0 寄存器

位	名称	类型	描述	默认值
31:9	-	-	保留	0
8:0	NM0	R	ADC 自校准寄存器 NM0	0

### 8.5.32 ANA\_ADCDATADMA 寄存器

表 8-33 ANA\_ADCDATADMA 寄存器

位	名称	类型	描述	默认值
31 : 16	-	-	保留	0x0
15 : 0	ADCDATA_DMA	R	DMA 的 ADC 数据源	0x0

### 8.5.33 ANA\_ADCCTRL1 寄存器

表 8-34 ANA\_ADCCTRL1 寄存器

位	名称	类型	描述	默认值
31: 16	RESDIV_CHx	R/W	每一位控制对应 ADC 通道的电阻分压使能与禁止。 0: 禁止分压 1: 使能分压 RESDIV_CH15~0 分别对应的 ADC 通道 15~0。	0x0
15	UPPER_THD3_EN	R/W	UPPER_THD3 阈值检测控制 0: 禁止 1: 使能	0x0
14	LOWER_THD3_EN	R/W	LOWER_THD3 阈值检测控制 0: 禁止 1: 使能	0x0
13	UPPER_THD2_EN	R/W	UPPER_THD2 阈值检测控制 0: 禁止 1: 使能	0x0
12	LOWER_THD2_EN	R/W	LOWER_THD2 阈值检测控制 0: 禁止 1: 使能	0x0
11	UPPER_THD1_EN	R/W	UPPER_THD1 阈值检测控制 0: 禁止 1: 使能	0x0
10	LOWER_THD1_EN	R/W	LOWER_THD1 阈值检测控制 0: 禁止 1: 使能	0x0
9	UPPER_THD0_EN	R/W	UPPER_THD0 阈值检测控制 0: 禁止 1: 使能	0x0
8	LOWER_THD0_EN	R/W	LOWER_THD0 阈值检测控制 0: 禁止 1: 使能	0x0
7: 0	-	R/W	保留，用户必须配置成 0x02。	0xC0

## 8.5.34 ANA\_ADCCTRL2 寄存器

表 8-35 ANA\_ADCCTRL2 寄存器

位	名称	类型	描述	默认值
31: 16	SCAN_CHx	R/W	每个位表示多通道 ADC 扫描模式下对应通道的使能或禁止。 0: 禁止 1: 使能 SCAN_CH15~0 分别对应的 ADC 通道 15~0。 SCAN_CHx 全部为 0 时候, 多通道 ADC 扫描通道为前 12 通道全扫描。	0x0
15	-	R/W	保留, 用户必须配置为 1。	0x0
14: 12	-	R/W	保留, 用户必须配置为 0。	0x0
11	CONV_ERR	R	ADC 转换错误标志位。表示转换期间是否发生错误。 0: 转换期间没有错误 1: 转换期间发生错误	0x0
10	CAL_ERR	R	ADC 自校准错误标志位。表示自校准期间是否发生错误。 0: 自校准期间没有发生错误 1: 自校准期间发生错误	0x0
9	CONV_ERR_CLR	R/C	向该位写 1 将 CONV_ERR 重置为 0。 向该位写 0, 清零此位	0x0
8	CAL_ERR_CLR	R/C	向该位写 1 将 CAL_ERR 重置为 0。 向该位写 0, 清零此位	0x0
7	ADC_CAL_DONE	R	ADC 自校准完成标志位 1: ADC 自校准完成 0: ADC 自校准未完成。 当发生 DPORST(PMU_STS[6]) 复位或 ADC RESET(ANA_ADCCTRL2[1])后, 该位清 0。	0x0
6	ADC_EN_TRG_CAL	R/W	ADC 自校准使能控制位 1: ADC_EN(ANA_ADCCTRL2[0])写 1 将触发 ADC 自校准流程。 0: 写 ADC_EN(ANA_ADCCTRL2[0])不会触发 ADC 自校准流程。	0x0
5	BUSY	R	ADC 是否忙碌标志位。 1: ADC 忙碌, ADC 在做自校准、取样或转换操作。 0: ADC 不忙碌。ADC 自校准、取样或转换是空闲的。	0x0
4	-	R/W	保留, 用户必须配置为 1。	0x0
3: 2	-	R/W	保留, 用户必须配置为 0。	0x0
1	RESET	R/W	复位除 ADC 自校准寄存器外的 ADC 数字电路。 0: 正常工作 1: 复位	0x0

0	ADC_EN	R/W	ADC 启用，由 MCLK 下降沿采样。 0：禁用 ADC，ADC 处于低功耗关机状态，自校准寄存器不复位。 1：启用 ADC。 ADC_EN 使能过 4 个 ADC 时钟后，ADC 操作才能使能。	0x0
---	--------	-----	--	-----

### 8.5.35 ANA\_ADCDATATHD1\_0 寄存器

表 8-36 ANA\_ADCDATATHD1\_0 寄存器

位	名称	类型	描述	默认值
31 : 24	UPPER_THD1	R/W	ADC 数据带符号右移 8 位后的数据高于这个设置值时，UPPER_THD1_TRGED 状态更新。BIT7 是符号位。	0x0
23 : 16	LOWER_THD1	R/W	ADC 数据带符号右移 8 位后的数据低于这个设置值时，LOWER_THD1_TRGED 状态更新。BIT7 是符号位。	0x0
15 : 8	UPPER_THD0	R/W	ADC 数据带符号右移 8 位后的数据高于这个设置值时，UPPER_THD0_TRGED 状态更新。BIT7 是符号位。	0x0
7 : 0	LOWER_THD0	R/W	ADC 数据带符号右移 8 位后的数据低于这个设置值时，LOWER_THD0_TRGED 状态更新。BIT7 是符号位。	0x0

### 8.5.36 ANA\_ADCDATATHD3\_2 寄存器

表 8-37 ANA\_ADCDATATHD3\_2 寄存器

位	名称	类型	描述	默认值
31 : 24	UPPER_THD3	R/W	ADC 数据带符号右移 8 位后的数据高于这个设置值时，UPPER_THD3_TRGED 状态更新。BIT7 是符号位。	0x0
23 : 16	LOWER_THD3	R/W	ADC 数据带符号右移 8 位后的数据低于这个设置值时，LOWER_THD3_TRGED 状态更新。BIT7 是符号位。	0x0
15 : 8	UPPER_THD2	R/W	ADC 数据带符号右移 8 位后的数据高于这个设置值时，UPPER_THD2_TRGED 状态更新。BIT7 是符号位。	0x0
7 : 0	LOWER_THD2	R/W	ADC 数据带符号右移 8 位后的数据低于这个设置值时，LOWER_THD2_TRGED 状态更新。BIT7 是符号位。	0x0

### 8.5.37 ANA\_ADCDATATHD\_CH 寄存器

表 8-38 ANA\_ADCDATATHD\_CH 寄存器

位	名称	类型	描述	默认值
31	UPPER_THD3_TRGED	R	指示 THD3_CH 通道的高 8 位 ADC 数据是否高于 UPPER_THD3 设置阈值 0：否 1：是	0x0
30	LOWER_THD3_TRGED	R	指示 THD3_CH 通道的高 8 位 ADC 数据是否低于 LOWER_THD3 设置阈值 0：否 1：是	0x0

29	UPPER_THD2_TRGED	R	指示 THD2_CH 通道的高 8 位 ADC 数据是否高于 UPPER_THD2 设置阈值 0: 否 1: 是	0x0
28	LOWER_THD2_TRGED	R	指示 THD2_CH 通道的高 8 位 ADC 数据是否低于 LOWER_THD2 设置阈值 0: 否 1: 是	0x0
27	UPPER_THD1_TRGED	R	指示 THD1_CH 通道的高 8 位 ADC 数据是否高于 UPPER_THD1 设置阈值 0: 否 1: 是	0x0
26	LOWER_THD1_TRGED	R	指示 THD1_CH 通道的高 8 位 ADC 数据是否低于 LOWER_THD1 设置阈值 0: 否 1: 是	0x0
25	UPPER_THD0_TRGED	R	指示 THD0_CH 通道的高 8 位 ADC 数据是否高于 UPPER_THD0 设置阈值 0: 否 1: 是	0x0
24	LOWER_THD0_TRGED	R	指示 THD0_CH 通道的高 8 位 ADC 数据是否低于 LOWER_THD0 设置阈值 0: 否 1: 是	0x0
23 : 22	THD3_SEL	R/W	设置 UPPER_THD3 和 LOWER_THD3 的触发条件 0: THD 标志位为 1 1: THD 标志位从 0 变为 1 2: THD 标志位从 1 变为 0 3: THD 标志位发生变化	0x0
21 : 20	THD2_SEL	R/W	同 THD3_SEL	0x0
19 : 18	THD1_SEL	R/W	同 THD3_SEL	0x0
17 : 16	THD0_SEL	R/W	同 THD3_SEL	0x0
15 : 12	THD3_CH	R/W	LOWER_THD3 和 UPPER_THD3 的通道选择数值 0~15 分别表示 ADC 通道 0~通道 15。	0x0
11 : 8	THD2_CH	R/W	同 THD3_CH	0x0
7 : 4	THD1_CH	R/W	同 THD3_CH	0x0
3 : 0	THD0_CH	R/W	同 THD3_CH	0x0

### 8.5.38 ANA\_CMPCTL 寄存器

表 8-39 ANA\_CMPCTL 寄存器

位	名称	类型	描述	默认值
31 : 24	PWR_DEB_SEL	R/W	VDDALARM、VDCIN 和 AVCCDET 的数字滤波控制寄存器。 0: 无滤波 1: 2 个 RTCCLK 时钟滤波 2: 3 个 RTCCLK 时钟滤波 3: 4 个 RTCCLK 时钟滤波 4: 5 个 RTCCLK 时钟滤波 ... 255: 256 个 RTCCLK 时钟滤波	0x00
23 : 22	VDDALARM_CHK_FRQ_SEL	R/W	VDDALARM 检测硬件间歇开启控制。当 ANA_REG9 的 bit7 配置为 1 时，该硬件控制生效。 0: 一直不开启，不检测 1: 以 RTCCLK 时钟频率，一直开启检测 2: 每 7.8125ms 间隔开启检测一次，检测完毕后，自动关闭。 3: 保留	0x0
21	CMP2_IO_NODEB	R/W	比较器 2 输出 IO (CMP2_O) 是否输出经过滤波控制的信号。 0: 经过滤波处理后的比较器输出 1: 原始信号输出	0x0
20	CMP2_INT_MASK_EN	R/W	当 CMP2_CNT 值超过 CMP2_THR 阈值，并且 CMP2_THR_EN = 1，产生一个中断还是多个中断选择。 0: 每次减法发生都产生一次中断 1: 一个中断	0x0
18 : 19	-	-	保留	-
17	CMP1_IO_NODEB	R/W	比较器 1 输出 IO (CMP1_O) 是否输出经过滤波控制的信号。 0: 经过滤波处理后的比较器输出 1: 原始信号输出	0x0
16	CMP1_INT_MASK_EN	R/W	当 CMP1_CNT 值超过 CMP1_THR 阈值，并且 CMP1_THR_EN = 1，产生一个中断还是多个中断选择。 0: 每次减法发生都产生一次中断 1: 一个中断	0x0
15 : 12	CMP2_CHK_NUM	R/W	CMP2 输出结果判定窗口设置 CMP2 原始输出连续 CMP2_CHK_NUM 次输出为 1，比较器输出结果为 1。 CMP2 原始输出连续 CMP2_CHK_NUM 次输出为 0，比较器输出结果为 0。 否则，比较器输出结果保持不变	0x0

			0~15: 1次~16次	
11	CMP2_THR_EN	R/W	CMP2 计数器减 CMP2_THR 使能 0: 比较器 2 保持向上累加 1: 如果比较器 2 的计数器值超过 CMP2_THR, 计数器值等于计数器值减去 CMP2_THR。直到计数器值小于 CMP2_THR。	0x0
10 : 8	CMP2_CHK_FRQ	R/W	CMP2 数据检测频率 0: RTCCLK 时钟周期间隔检测一次 1: 7.8125ms 间隔检测一次 2: 125ms 间隔检测一次 3: 250ms 间隔检测一次 4: 500ms 间隔检测一次 5: 保留 6: 保留 7: 保留	0x0
7 : 4	CMP1_CHK_NUM	R/W	CMP1 输出结果判定窗口设置 CMP1 原始输出连续 CMP1_CHK_NUM 次输出为 1, 比较器输出结果为 1。 CMP1 原始输出连续 CMP1_CHK_NUM 次输出为 0, 比较器输出结果为 0。 否则, 比较器输出结果保持不变 0~15: 1次~16次	0x0
3	CMP1_THR_EN	R/W	CMP1 计数器减 CMP1_THR 使能 0: 比较器 1 保持向上累加 1: 如果比较器 1 的计数器值超过 CMP1_THR, 计数器值等于计数器值减去 CMP1_THR。直到计数器值小于 CMP1_THR。	0x0
2 : 0	CMP1_CHK_FRQ	R/W	CMP1 数据检测频率 0: RTCCLK 时钟周期间隔检测一次 1: 7.8125ms 间隔检测一次 2: 125ms 间隔检测一次 3: 250ms 间隔检测一次 4: 500ms 间隔检测一次 5: 保留 6: 保留 7: 保留	0x0

### 8.5.39 ANA\_CMPTHR 寄存器

表 8-40 ANA\_CMPTHR 寄存器

位	名称	类型	描述	默认值
---	----	----	----	-----

31 : 16	CMP2_THR	R/W	<p>CMP2 中断阈值设置</p> <p>0: 每次比较器 2 计数器值加 1, 中断标志位都会置位。</p> <p>1~65535: 当比较器 2 计数器值大于该阈值时, 中断标志位会置位</p>	0x0000
15 : 0	CMP1_THR	R/W	<p>CMP1 中断阈值设置</p> <p>0: 每次比较器 1 计数器值加 1, 中断标志位都会置位。</p> <p>1~65535: 当比较器 1 计数器值大于该阈值时, 中断标志位会置位</p>	0x0000

## 第9章 ADC 控制器

### 9.1 简介

芯片中集成了一个 12bit SAR ADC, 支持 16 通道的信号输入。所有通道的采样量程在不分压情况下均为 0~1.2V。分压情况下, 普通通道测量范围 1.2V~3.3V, BAT 通道测量范围 1.2V~5.2V。ADC 采样通道包含 GND、温度、BAT1、BATRTC 电压以及外部输入信号。ADC 的时钟来自 6.5536M RCH 或 PLLL, 支持 6.5MHz 时钟频率, 转换速率为 ADC 时钟频率/(19+8)。ADC 控制器工作在主域, 因此与 ADC 相关的寄存器设置在深睡眠下可保持, 不会被复位。

注意: ADC 时钟和系统时钟非同步, 可独立配置, 在操作 ADC 相关寄存器时, 需要等待至少一个 ADC clock 时间才能生效。

### 9.2 特点

- ADC 支持手动转换和自动转换模式;
- 支持 ADC 转换完成中断;
- 支持 16 全通道或多通道全扫描功能。

### 9.3 ADC 功能框图

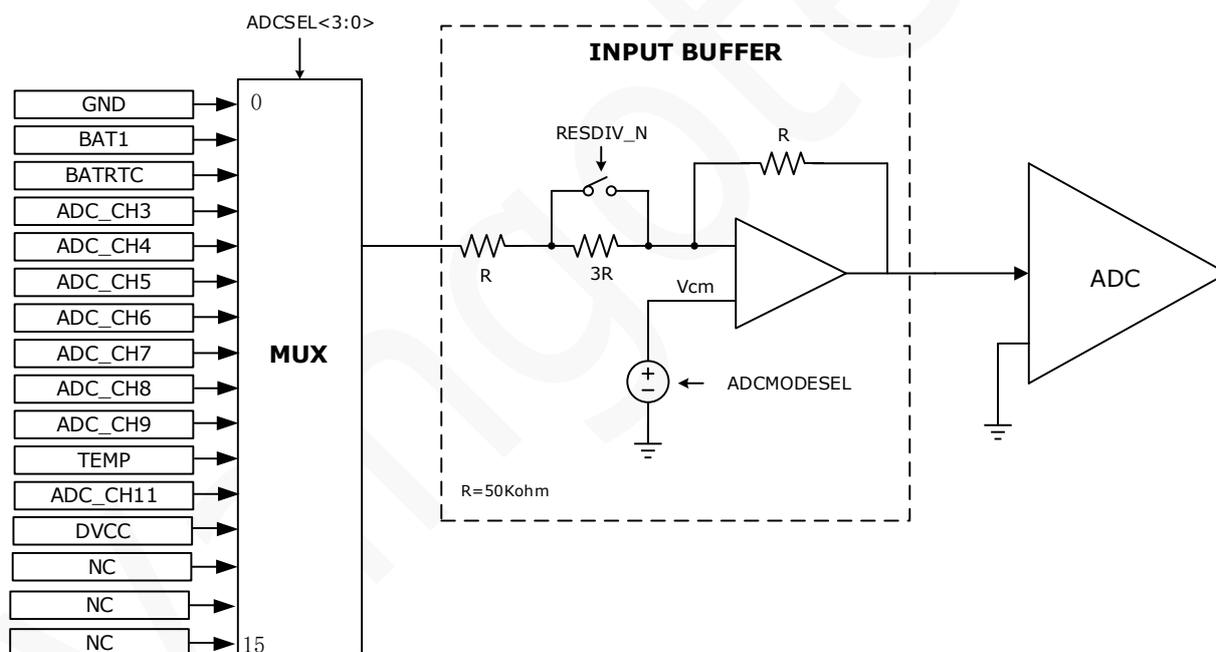


图 9-1 ADC 功能框图

## 9.4 寄存器地址

表 9-1 ANA 控制寄存器(ANA 基地址: 0x40014200)

名称	类型	地址	描述	默认值
ANA_REG0	R/W	0x0000	模拟寄存器 0	0x00
ANA_REG1	R/W	0x0004	模拟寄存器 1	0x00
ANA_REG3	R/W	0x000C	模拟寄存器 3	0x00
ANA_REG11	W	0x0044	模拟寄存器 11	--
ANA_ADCSTATE	R	0x005C	ADC 状态寄存器	--
ANA_INTSTS	R/C	0x0060	模拟中断状态寄存器	0x0000
ANA_INTEN	R/W	0x0064	模拟中断使能寄存器	0x0000
ANA_ADCCTRL0	R/W	0x0068	ADC 控制寄存器 0	0x00000000
ANA_ADCDATA0	R	0x0070	ADC 通道 0 数据寄存器	--
ANA_ADCDATA1	R	0x0074	ADC 通道 1 数据寄存器	--
ANA_ADCDATA2	R	0x0078	ADC 通道 2 数据寄存器	--
ANA_ADCDATA3	R	0x007C	ADC 通道 3 数据寄存器	--
ANA_ADCDATA4	R	0x0080	ADC 通道 4 数据寄存器	--
ANA_ADCDATA5	R	0x0084	ADC 通道 5 数据寄存器	--
ANA_ADCDATA6	R	0x0088	ADC 通道 6 数据寄存器	--
ANA_ADCDATA7	R	0x008C	ADC 通道 7 数据寄存器	--
ANA_ADCDATA8	R	0x0090	ADC 通道 8 数据寄存器	--
ANA_ADCDATA9	R	0x0094	ADC 通道 9 数据寄存器	--
ANA_ADCDATA10	R	0x0098	ADC 通道 10 数据寄存器	--
ANA_ADCDATA11	R	0x009C	ADC 通道 11 数据寄存器	--
ANA_ADCDATA12	R	0x00A0	ADC 通道 12 数据寄存器	--
ANA_ADCDATA13	R	0x00A4	ADC 通道 13 数据寄存器	--
ANA_ADCDATA14	R	0x00A8	ADC 通道 14 数据寄存器	--
ANA_ADCDATA15	R	0x00AC	ADC 通道 15 数据寄存器	--
ANA_ADCDOS	R	0x00C0	ADC 自校准寄存器 DOS	0
ANA_ADCDCPN	R	0x00CC	ADC 自校准寄存器 DCPN	0
ANA_ADCDCNM0	R	0x00D8	ADC 自校准寄存器 DCNM0	0
ANA_ADCDATADMA	R/W	0x00E0	DMA 的 ADC 数据源	--
ANA_ADCCTRL1	R/W	0x00E8	ADC 控制寄存器 1	0x000000C0
ANA_ADCCTRL2	R/W	0x00EC	ADC 控制寄存器 2	0x00000000
ANA_ADCDATATHD1_0	R/W	0x00F4	ADC 阈值控制寄存器	0x00000000
ANA_ADCDATATHD3_2	R/W	0x00F8	ADC 阈值控制寄存器	0x00000000
ANA_ADCDATATHD_CH	R/W	0x00FC	ADC 阈值状态寄存器	0x00000000

表 9-2 RTC 控制器(RTC 基地址: 0x40014800)

名称	类型	地址	描述	默认值
RTC_ADCUCALK	R/W	0x00E8	ADC Ucal K 系数寄存器，有写保护	0x546A546A
RTC_ADCMACTL	R/W	0x00EC	ADC 求平均控制寄存器，有写保护	0x70000000
RTC_ADCDTCTL	R/W	0x00F0	ADC 数据控制	0x80000000

## 9.5 寄存器定义

### 9.5.1 ANA\_REG0 寄存器

表 9-3 ANA\_REG0 寄存器

位	名称	类型	描述	默认值
7:0	-	R/W	保留，必须配置为 0x30。	0

### 9.5.2 ANA\_REG1 寄存器

表 9-4 ANA\_REG1 寄存器

位	名称	类型	描述	默认值
7	ADCMODESEL	R/W	ADC 模式选择 当 RESDIV_CHx=1 时，该控制位有效。 0: 测试通道输入非交流周波 1: 测试通道输入交流周波（如±300mV 信号）	0

### 9.5.3 ANA\_REG3 寄存器

表 9-5 ANA\_REG3 各个 bit 的功能描述

位	名称	类型	描述	默认值
3	BGPPD <sup>*[1]</sup>	R/W	BGP 掉电控制 0: 开启 BGP 1: 关闭 BGP	0
4	RCHPD	R/W	高频 RC 时钟 RCH(6.5536MRC)开关控制信号 0: 开启 RCH 1: 关闭 RCH	0

<sup>\*[1]</sup>: RCH、PLL、ADC 与 BGP 有关系，在开 RCH、PLL、ADC 之前，必须开启 BGP。如果系统中没有使用 BGP，可通过 BGPPD 位关闭 BGP；如果 BGP 有被用到，此时，通过 BGPPD 不能关闭 BGP，因为系统具有保护功能，会重新开启 BGP。浅睡眠或深睡眠模式下，不管 BGP 是否有被用到，都可以关闭 BGP。

### 9.5.4 ANA\_ADCSTATE 寄存器

表 9-6 ANA\_ADCSTATE 寄存器

位	名称	类型	描述	默认值
31:5	-	-	保留	0x0
4	RESET	R	该位显示 ADC 接口中 RESET 的状态。	0x0

3	ADC_EN	R	该位显示 ADC 接口中 ADC_EN 的状态。	0x0
2:0	ADCSTATE	R	这些位显示 ADC 控制器状态机的状态。 0: 空闲 2: ADC 多通道自动转换 4: ADC 多通道手动转换 其它: 保留	0x0

### 9.5.5 ANA\_INTSTS 寄存器

表 9-7 ANA\_INTSTS 寄存器

位	名称	类型	描述	默认值
21	INTSTS21	R/C	ADC_THD3 数据大于 UPPER_THD3 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
20	INTSTS20	R/C	ADC_THD3 数据小于 LOWER_THD3 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
19	INTSTS19	R/C	ADC_THD2 数据大于 UPPER_THD2 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
18	INTSTS18	R/C	ADC_THD2 数据小于 LOWER_THD2 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
17	INTSTS17	R/C	ADC_THD1 数据大于 UPPER_THD1 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
16	INTSTS16	R/C	ADC_THD1 数据小于 LOWER_THD1 阈值中断标志位。 读 0: 未发生	0x0

			读 1: 发生 写 0: 无影响 写 1: 清除该位	
15	INTSTS15	R/C	ADC_THD0 数据大于 UPPER_THD0 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
14	INTSTS14	R/C	ADC_THD0 数据小于 LOWER_THD0 阈值中断标志位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
1	INTSTS1	R/C	ADC 自动转换完成中断标记位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0
0	INTSTS0	R/C	手动设置 ADC 转换完成中断标记位。 读 0: 未发生 读 1: 发生 写 0: 无影响 写 1: 清除该位	0x0

### 9.5.6 ANA\_INTEN 寄存器

表 9-8 ANA\_INTEN 寄存器

位	名称	类型	描述	默认值
21	INTEN21	R/W	ADC_THD3 数据大于 UPPER_THD3 阈值中断使能控制。 0: 禁止 1: 使能	0x0
20	INTEN20	R/W	ADC_THD3 数据小于 LOWER_THD3 阈值中断使能控制。 0: 禁止 1: 使能	0x0
19	INTEN19	R/W	ADC_THD2 数据大于 UPPER_THD2 阈值中断使能控制。 0: 禁止 1: 使能	0x0

18	INTEN18	R/W	ADC_THD2 数据小于 LOWER_THD2 阈值中断使能控制。 0: 禁止 1: 使能	0x0
17	INTEN17	R/W	ADC_THD1 数据大于 UPPER_THD1 阈值中断使能控制。 0: 禁止 1: 使能	0x0
16	INTEN16	R/W	ADC_THD1 数据小于 LOWER_THD1 阈值中断使能控制。 0: 禁止 1: 使能	0x0
15	INTEN15	R/W	ADC_THD0 数据大于 UPPER_THD0 阈值中断使能控制。 0: 禁止 1: 使能	0x0
14	INTEN14	R/W	ADC_THD0 数据小于 LOWER_THD0 阈值中断使能控制。 0: 禁止 1: 使能	0x0
1	INTEN1	R/W	ADC 自动转换完成的中断使能控制。 0: 禁止 ADC 自动转换中断。 1: 使能 ADC 自动转换中断。	0x0
0	INTEN0	R/W	ADC 手动转换完成的中断使能控制。 0: 禁止 ADC 手动转换中断。 1: 使能 ADC 手动转换中断。	0x0

### 9.5.7 ANA\_ADCCTRL0 寄存器

表 9-9 ANA\_ADCCTRL0 寄存器

位	名称	类型	描述	默认值
31	MTRIG	R/W	手动控制 ADC 转换。 写 0: 无影响 写 1: 使能一次手动 ADC 转换 读 0: 当前手动设置的 ADC 转换已完成 读 1: 当前手动设置的 ADC 转换未完成	0x0
30: 22	-	-	保留	0x0
21: 20	-	R/W	保留，用户必须配置为 0x03。	0x0
19	STOP	R/W	强制停止 ADC 转换流程。向该位写入 1 并保持 1 个 ADCCLK 时间，可停止当前的 ADC 转换。在下次 ADC 转换开始之前向该寄存器写入 0。	0x0
18: 16	AEN	R/W	ADC 自动转换使能控制寄存器。 0: 禁止 ADC 自动转换 1: ADC 自动转换由 RTC_ITV 和 RTC_SITV 触发	0x0

			2: ADC 自动转换由 RTC WKUSEC 触发 3: ADC 自动转换由 RTC ALARM 触发 4: ADC 自动转换由定时器 0 的溢出触发 5: ADC 自动转换由定时器 1 的溢出触发 6: ADC 自动转换由定时器 2 的溢出触发 7: ADC 自动转换由定时器 3 的溢出触发	
15: 13	-	-	保留	0x0
12	CLKSRCSEL	R/W	ADC 时钟源选择。当更改该寄存器时，硬件逻辑会停止，停止时间为 3 个 ADC 时钟，此时任何 ADC 操作都会被硬件忽略。所以在此期间不要操作 ADC。 0: 6.5M RCH 1: PLLL	0x0
11: 0			保留	0x0

表 9-10 ADC 采样通道

通道号	测量信号
0	GND
1	BAT1
2	BATRTC
3~9	ADC_CHx
10	温度
11	ADC_CH11
12	DVCC
13~15	NC

说明：当选择温度通道时，温度传感器自动打开；当选择其它通道时，温度传感器自动关闭。通过 SCAN\_CHx 配置通道，ADC 支持单通道测量和多通道同时扫描。

ADC 测温通道建立时间需要 2us，在多个通道转换 ADC 时，不能包含测温通道，测温通道必须单独操作。

### 9.5.8 ANA\_ADCCTRL1 寄存器

表 9-11 ANA\_ADCCTRL1 寄存器

位	名称	类型	描述	默认值
31: 16	RESDIV_CHx	R/W	每一位控制对应 ADC 通道的电阻分压使能与禁止。 0: 禁止分压 1: 使能分压 RESDIV_CH15~0 分别对应的 ADC 通道 15~0	0x0
15	UPPER_THD3_EN	R/W	UPPER_THD3 阈值检测控制 0: 禁止 1: 使能	0x0
14	LOWER_THD3_EN	R/W	LOWER_THD3 阈值检测控制	0x0

			0: 禁止 1: 使能	
13	UPPER_THD2_EN	R/W	UPPER_THD2 阈值检测控制 0: 禁止 1: 使能	0x0
12	LOWER_THD2_EN	R/W	LOWER_THD2 阈值检测控制 0: 禁止 1: 使能	0x0
11	UPPER_THD1_EN	R/W	UPPER_THD1 阈值检测控制 0: 禁止 1: 使能	0x0
10	LOWER_THD1_EN	R/W	LOWER_THD1 阈值检测控制 0: 禁止 1: 使能	0x0
9	UPPER_THD0_EN	R/W	UPPER_THD0 阈值检测控制 0: 禁止 1: 使能	0x0
8	LOWER_THD0_EN	R/W	LOWER_THD0 阈值检测控制 0: 禁止 1: 使能	0x0
7: 0	-	R/W	保留, 用户必须配置成 0x02。	0xC0

### 9.5.9 ANA\_ADCCTRL2 寄存器

表 9-12 ANA\_ADCCTRL2 寄存器

位	名称	类型	描述	默认值
31: 16	SCAN_CHx	R/W	每个位表示多通道 ADC 扫描模式下对应通道的使能或禁止。 0: 禁止 1: 使能 SCAN_CH15~0 分别对应的 ADC 通道 15~0。 SCAN_CHx 全部为 0 时候, 多通道 ADC 扫描通道为前 12 通道全扫描。	0x0
15	-	R/W	保留, 用户必须配置为 1。	0x0
14: 12	-	R/W	保留, 用户必须配置为 0。	0x0
11	CONV_ERR	R	ADC 转换错误标志位。表示转换期间是否发生错误。 0: 转换期间没有错误 1: 转换期间发生错误	0x0
10	CAL_ERR	R	ADC 自校准错误标志位。表示自校准期间是否发生错误。	0x0

			0: 自校准期间没有发生错误 1: 自校准期间发生错误	
9	CONV_ERR_CLR	R/C	向该位写 1 将 CONV_ERR 重置为 0。 向该位写 0, 清零此位	0x0
8	CAL_ERR_CLR	R/C	向该位写 1 将 CAL_ERR 重置为 0。 向该位写 0, 清零此位	0x0
7	ADC_CAL_DONE	R	ADC 自校准完成标志位 1: ADC 自校准完成 0: ADC 自校准未完成。 当发生 DPORST(PMU_STS[6]) 复位或 ADC RESET(ANA_ADCCTRL2[1])后, 该位清 0。 ADC 自校准开启后, 需等待至少一个 ADC clock 才能查询该标志位, 保证有足够的时间刷新该位。	0x0
6	ADC_EN_TRG_CAL	R/W	ADC 自校准使能控制位 1: ADC_EN(ANA_ADCCTRL2[0])写 1 将触发 ADC 自校准流程。 0: 写 ADC_EN(ANA_ADCCTRL2[0])不会触发 ADC 自校准流程。	0x0
5	BUSY	R	ADC 是否忙碌标志位。 1: ADC 忙碌, ADC 在做自校准、取样或转换操作。 0: ADC 不忙碌。ADC 自校准、取样或转换是空闲的。	0x0
4	-	R/W	保留, 用户必须配置为 1。	0x0
3: 2	-	R/W	保留, 用户必须配置为 0。	0x0
1	RESET	R/W	复位除 ADC 自校准寄存器外的 ADC 数字电路。 0: 正常工作 1: 复位	0x0
0	ADC_EN	R/W	ADC 启用, 由 MCLK 下降沿采样。 0: 禁用 ADC, ADC 处于低功耗关机状态, 自校准寄存器不复位。 1: 启用 ADC。 ADC 配置后, 因为模拟 buff 建立需要时间, 所以启用 ADC 后, 需软件等待 100us, 再做 ADC 转换。  禁用 ADC 即 ADC_EN 写 0 需按以下流程进行操作 a) 在 ADC_EN 写 0 前, 需要确保 ADC_State 为 idle。 b) ADC_State 为 idle 后, ADC_EN 写 0。 c) 确保 ADC_State 寄存器 ADC_EN 状态为 0。	0x0

### 9.5.10 ANA\_ADCDATATHD1\_0 寄存器

表 9-13 ANA\_ADCDATATHD1\_0 寄存器

位	名称	类型	描述	默认值
---	----	----	----	-----

31 : 24	UPPER_THD1	R/W	ADC 数据带符号右移 8 位后的数据高于这个设置值时, UPPER_THD1_TRGED 状态更新。BIT7 是符号位。	0x0
23 : 16	LOWER_THD1	R/W	ADC 数据带符号右移 8 位后的数据低于这个设置值时, LOWER_THD1_TRGED 状态更新。BIT7 是符号位。	0x0
15 : 8	UPPER_THD0	R/W	ADC 数据带符号右移 8 位后的数据高于这个设置值时, UPPER_THD0_TRGED 状态更新。BIT7 是符号位。	0x0
7 : 0	LOWER_THD0	R/W	ADC 数据带符号右移 8 位后的数据低于这个设置值时, LOWER_THD0_TRGED 状态更新。BIT7 是符号位。	0x0

### 9.5.11 ANA\_ADCDATATHD3\_2 寄存器

表 9-14 ANA\_ADCDATATHD3\_2 寄存器

位	名称	类型	描述	默认值
31 : 24	UPPER_THD3	R/W	ADC 数据带符号右移 8 位后的数据高于这个设置值时, UPPER_THD3_TRGED 状态更新。BIT7 是符号位。	0x0
23 : 16	LOWER_THD3	R/W	ADC 数据带符号右移 8 位后的数据低于这个设置值时, LOWER_THD3_TRGED 状态更新。BIT7 是符号位。	0x0
15 : 8	UPPER_THD2	R/W	ADC 数据带符号右移 8 位后的数据高于这个设置值时, UPPER_THD2_TRGED 状态更新。BIT7 是符号位。	0x0
7 : 0	LOWER_THD2	R/W	ADC 数据带符号右移 8 位后的数据低于这个设置值时, LOWER_THD2_TRGED 状态更新。BIT7 是符号位。	0x0

### 9.5.12 ANA\_ADCDATATHD\_CH 寄存器

表 9-15 ANA\_ADCDATATHD\_CH 寄存器

位	名称	类型	描述	默认值
31	UPPER_THD3_TRGED	R	指示 THD3_CH 通道的 ADC 数据是否高于 UPPER_THD3 设置阈值 0: 否 1: 是	0x0
30	LOWER_THD3_TRGED	R	指示 THD3_CH 通道的 ADC 数据是否低于 LOWER_THD3 设置阈值 0: 否 1: 是	0x0
29	UPPER_THD2_TRGED	R	指示 THD2_CH 通道的 ADC 数据是否高于 UPPER_THD2 设置阈值 0: 否 1: 是	0x0
28	LOWER_THD2_TRGED	R	指示 THD2_CH 通道的 ADC 数据是否低于 LOWER_THD2 设置阈值 0: 否 1: 是	0x0
27	UPPER_THD1_TRGED	R	指示 THD1_CH 通道的 ADC 数据是否高于 UPPER_THD1 设置阈值	0x0

			0: 否 1: 是	
26	LOWER_THD1_TRGED	R	指示 THD1_CH 通道的 ADC 数据是否低于 LOWER_THD1 设置阈值 0: 否 1: 是	0x0
25	UPPER_THD0_TRGED	R	指示 THD0_CH 通道的 ADC 数据是否高于 UPPER_THD0 设置阈值 0: 否 1: 是	0x0
24	LOWER_THD0_TRGED	R	指示 THD0_CH 通道的 ADC 数据是否低于 LOWER_THD0 设置阈值 0: 否 1: 是	0x0
23 : 22	THD3_SEL	R/W	设置 UPPER_THD3 和 LOWER_THD3 的触发条件 0: THD 标志位为 1 1: THD 标志位从 0 变为 1 2: THD 标志位从 1 变为 0 3: THD 标志位发生变化	0x0
21 : 20	THD2_SEL	R/W	同 THD3_SEL	0x0
19 : 18	THD1_SEL	R/W	同 THD3_SEL	0x0
17 : 16	THD0_SEL	R/W	同 THD3_SEL	0x0
15 : 12	THD3_CH	R/W	LOWER_THD3 和 UPPER_THD3 的通道选择 数值 0~15 分别表示 ADC 通道 0~通道 15。	0x0
11 : 8	THD2_CH	R/W	同 THD3_CH	0x0
7 : 4	THD1_CH	R/W	同 THD3_CH	0x0
3 : 0	THD0_CH	R/W	同 THD3_CH	0x0

### 9.5.13 ANA\_ADCDATAx 寄存器

表 9-16 ANA\_ADCDATAx 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	-
15: 0	ADCDATA	R	ADC 的转换结果数据寄存器，是 16 位补码。 DATA0: 存储 ADC 通道 0 转换结果。 DATA1: 存储 ADC 通道 1 转换结果。 ... DATA15: 存储 ADC 通道 15 转换结果。	0x--

### 9.5.14 ANA\_ADCDOS 寄存器

表 9-17 ANA\_ADCDOS 寄存器

位	名称	类型	描述	默认值
31:9	-	-	保留	0
8:0	DOS	R	ADC 自校准寄存器 DOS	

### 9.5.15 ANA\_ADCDCPN 寄存器

表 9-18 ANA\_ADCDCPN 寄存器

位	名称	类型	描述	默认值
31:27	DCPN4	R	ADC 自校准寄存器 DCPN4	
26:18	DCPN3	R	ADC 自校准寄存器 DCPN3	
17:9	DCPN2	R	ADC 自校准寄存器 DCPN2	
8:0	DCPN1	R	ADC 自校准寄存器 DCPN1	

### 9.5.16 ANA\_ADCDCNM0 寄存器

表 9-19 ANA\_ADCDCNM0 寄存器

位	名称	类型	描述	默认值
31:9	-	-	保留	0
8:0	NM0	R	ADC 自校准寄存器 NM0	0

### 9.5.17 ANA\_ADCDATADMA 寄存器

表 9-20 ANA\_ADCDATADMA 寄存器

位	名称	类型	描述	默认值
31:16	-	-	保留	0
15:0	ADCDATA_DMA	R	DMA 的 ADC 数据源	0

### 9.5.18 RTC\_ADCUCALK 寄存器

表 9-21 RTC\_ADCUCALK 寄存器

位	名称	类型	描述	默认值
31:16	UCAL_K3	R/W	测温通道 UCAL K3, 用户必须配置为 0x599A	0x546A
15:0	UCAL_K1	R/W	测温通道 UCAL K1, 用户必须配置为 0x599A	0x546A

### 9.5.19 RTC\_ADCMACTL 寄存器

表 9-22 RTC\_ADCMACTL 寄存器

位	名称	类型	描述	默认值
31: 28	-	-	保留	0x7
27	-	R/W	保留, 用户必须设置为 0	0x0

26: 24	-	-	保留	00
23 : 20	SKIP_SAMPLE	R/W	该寄存器用来选择忽略最开始的 ADC 输出采样点的数量。 SKIP_SAMPLE [3: 0] = x: 忽略最开始的 x 个采样点。 0: 忽略 0 个采样点 4: 忽略 4 个采样点 8: 忽略 8 个采样点 12: 忽略 12 个采样点 其它: 保留	0x0
19	-	-	保留	0x0
18 : 16	AVERAGE_SAMPLE	R/W	求平均的采样点的数量 0: 2 点求平均 1: 4 点求平均 2: 8 点求平均 3: 16 点求平均 4: 32 点求平均 5: 64 点求平均 6: 保留 7: 保留	0x0
15:0	AVERAGE_CHx	R/W	每个位分别控制对应通道的求平均使能 1: 使能 0: 禁止 AVERAGE_CH15~0 分别对应 ADC 通道 15~0。	0x0

## 9.5.20 RTC\_ADCDTCTL 寄存器

表 9-23 RTC\_ADCDTCTL 寄存器

位	名称	类型	描述	默认值
31:30	ENDED_IN2ADC_CONVERT	R/W	必须配置为 2	0x2
29:0	-	-	保留	-

## 9.6 ADC 自校准流程

➤ ADC 自校准说明:

- 1、ADC 自校准需要花费 4096 个 ADC clock 时钟。
- 2、发生 DPORST(PMU\_STS[6])复位、ADC RESET(ANA\_ADCCTRL2[1])以及 AVCC 电压低于 2.2V 时，自校准的数据会丢失，即 ANA\_ADCDOS 变为 0 或 ADC\_CAL\_DONE 变为 0，恢复正常后需重新做一次自校准。
- 3、ADC 由 AVCC 供电，正常工作电压不得低于 2.2V，建议在自校准前对 AVCC 电压进行查询 (ANA\_CMPOUT[10])，判断是否满足条件。
- 4、开始 ADC 自校准工作前，建议做一次 ADC 外设寄存器的推荐默认值填写

➤ ADC 自校准流程流程:

1、设置 ADC 时钟

- a) 选择 6.5M ADC 时钟
- b) 保持至少一个 ADC clock

2、关闭 ADC 功能

- a) 设置  $ADC\_EN = 0$  并保持至少一个 ADC clock
- b) 设置  $CAL\_ERR\_CLR=1$  并保持至少一个 ADC clock
- c) 设置  $CAL\_ERR\_CLR=0$  并保持至少一个 ADC clock

3、停止当前 ADC 转换

- a) 等待  $MTRIG = 0$ ，即等待当前手动设置的 ADC 转换完成
- b) 设置  $STOP = 1$  并保持至少一个 ADC clock
- c) 设置  $STOP = 0$  并保持至少一个 ADC clock

4、复位 ADC

- a) 设置  $RESET = 1$  并保持至少一个 ADC clock
- b) 设置  $RESET = 0$  并保持至少一个 ADC clock

5、使能 ADC 自校准

- a) 设置  $ADC\_EN\_TRG\_CAL = 1$  并保持至少一个 ADC clock
- b) 设置  $ADC\_EN = 1$  并保持至少一个 ADC clock

6、等待 ADC 自校准完成

- a) 等待至少一个 ADC clock
- b) 等待  $ADC\_CAL\_DONE$  为 1

7、关闭 ADC 自校准

- a) 设置  $ADC\_EN\_TRG\_CAL = 0$  并保持至少一个 ADC clock
- b) 设置  $ADC\_EN = 0$  并保持至少一个 ADC clock

8、检测  $CAL\_ERR$

- a) 为 0，自校准成功
- b) 为 1，自校准失败:

设置  $CAL\_ERR\_CLR=1$  将  $CAL\_ERR$  重置为 0，并保持至少一个 ADC clock

设置  $CAL\_ERR\_CLR=0$  并保持至少一个 ADC clock

从步骤 3 开始重新进行自校准

相关寄存器说明

$ADC\_EN$  即  $ANA\_ADCCTRL2[0]$

$CAL\_ERR\_CLR$  即  $ANA\_ADCCTRL2[8]$

$MTRIG$  即  $ADCCTRL0[31]$

$STOP$  即  $ADCCTRL0[19]$

$RESET$  即  $ADCCTRL2[1]$

$ADC\_EN\_TRG\_CAL$  即  $ANA\_ADCCTRL2[6]$

$ADC\_CAL\_DONE$  即  $ANA\_ADCCTRL2[7]$

CAL\_ERR 即 ANA\_ADCCTRL2[10]

## 9.7 电池电压测量和外部电压测量

在 V85XXP 中，ADC 能通过外部输入管脚测量外部信号，也能测量 BAT1、BATRTC 等信号的电压。

ADC 工作模式：自动触发 ADC 转换、手动触发 ADC 转换。

ADC 采样通道选择：单通道、多通道。

ADC 分压模式：不分压、电阻分压。电阻分压有额外功耗，这个电路消耗的功率：

$$P = U_D \times I_D = (V_{IN} - V_{CM}) \times \frac{V_{IN} - V_{CM}}{R_1 + 3R_1} \quad \text{Equation 9-1}$$

其中， $V_{CM}=1.2V$ ， $R_1=50Kohm$

测量外部电压（ADC\_CHx），包括通道 3、4、5、6、7、8、9、11。

测量电池电压通道，即 BAT1、BATRTC 测量（通道 1、2）。

表 9-24 不同量程 ADC 输入信号的配置范围

配置	输入信号		说明
	最小	最大	
小于 1.2V 的通道电压	0.0	1.2	小于 1.2V 的 ADC_CHx 通道
大于 1.2V 的通道电压	1.2	3.3	大于 1.2V 的 ADC_CHx 通道
电池（BAT1/BATRTC）	1.2	5.2	电池（BAT1/BATRTC）
通道输入交流周波（如±300mV 信号）	-0.3	0.4	通道输入交流周波（如±300mV 信号）

表 9-25 直流电压、电池电压配置

寄存器	配置	说明
ADCMODESEL	0	是否为交流模式
SKIP_SAMPLE		忽略点数，建议忽略 4 个点数
AVERAGE_SAMPLE		求平均点数
RESDIV_CHx		是否电阻分压
SCAN_CHx		选择扫描通道

表 9-26 测交流电压配置

寄存器	配置	说明
ADCMODESEL	1	是否为交流模式
SKIP_SAMPLE		忽略的点数，建议忽略 4 个点数
AVERAGE_SAMPLE		求平均点数
RESDIV_CHx	1	是否电阻分压
SCAN_CHx		选择扫描通道

表 9-27 电压测量量程与公式

Channel	Divider Mode	Signal range (V)	Formula
ADC_CHx	No divider	-0.2~1.2	$V_{DC} = a1 * X \div 100000000 + b1 \div 100000000 + offset1 \div 1000$
	Resistive	1.2~3.3	$V_{DC} = a2 * X \div 100000000 + b2 \div 100000000 + offset2 \div 1000$
BAT1 channels	Resistive	2.5~5.2	$V_{DC} = a3 * X \div 100000000 + b3 \div 100000000 + offset3 \div 1000$
BATRTC channels	Resistive	0.7~5.2	$V_{DC} = a4 * X \div 100000000 + b4 \div 100000000 + offset4 \div 1000$

ADC系数的a、b存储在Info信息区，有双备份。用户需要检查用于存储ADC系数的a、b、offset存储区域参数是否合法。运算公式见表9-27 电压测量量程与公式。ADC系数的a、b、offset根据不同的电源系统选择不同位置的数据。见表4-21 Info信息寄存器，地址范围0x00080C00~0x00080CD8。

上述公式中的X值计算公式如下：

$$X = ADC\_DATA_x \quad \text{Equation 9-2}$$

其中：ADCDATA<sub>x</sub>是地址(0x40014270+4x)(16bit)中的值，x是对应的通道号0-15。

## 9.8 温度测量

CH10为温度测量通道，温度测量范围是-40~+85°C。

表 9-28 测温

寄存器	配置	说明
ADCMODESEL	0	是否为交流模式
SKIP_SAMPLE		忽略的点数，建议忽略4个点数
AVERAGE_SAMPLE	>0	求平均点数：4点以及4点以上
RESDIV_CH10	0	通道10是否电阻分压

测量温度的步骤如下：

1. 确保ADC至少成功完成过一次校正。设置手动或自动触发ADC。
2. 设置ADC  
ADC时钟选择为6.5M RCH，内部平均选择至少4点，如需达到测温跳差0.1°C以内，建议设置内部64点平均，再使用软件做64次平均。选择通道10，打开ADC。手动触发/自动触发ADC转换。
3. 等待ANA\_INTSTS(0x40014260)寄存器的位INTSTS0(手动触发)/INTSTS1(自动触发)值为1时，读寄存器ANA\_ADCDATA10(0x40014298)的值。
4. 计算温度T：(单位：1/256°C)，T是一个16位的补码，实际温度值等于T/256.0。例如0x1880代表24.5摄氏度。

计算公式：

$$T = (P0 * ((X_m * X_m) \gg 16) + P1 * X_m + P2) \gg 8 \quad \text{Equation 9-3}$$

$X_m$ 为校准点(如25°C)下ADC数值

P0是地址0x80D10(16bit)中的值

P1是地址0x80D12(16bit)中的值

P2是由以下公式计算出的值：

$$P2 = P2' + (T_r - T_m) * 256 \quad \text{Equation 9-4}$$

P2'是地址0x80D14(32bit)中的值

$T_r$  是地址 0x80D80(32bit)中的值

$T_m$  是地址 0x80D84(32bit)中的值

## 第10章 比较器控制器

### 10.1 简介

V85XXP 集成了两个比较器，用来比较模拟信号的大小，有以下几种比较模式：

- 比较器正端接管脚 CMP\_P，负端信号接管脚 CMP\_N；
- 比较器正端接管脚 CMP\_P，负端信号接 BGPREF 或 VREF；
- 比较器正端接管脚 CMP\_N，负端信号接 BGPREF 或 VREF；
- 比较器正端接管脚 BAT1/BATRTC，负端信号接 BGPREF 或 VREF。

每个比较器有 20mV 的电压回差。

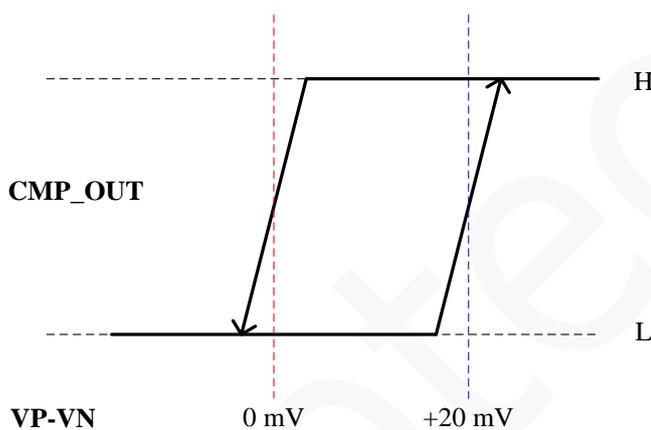


图 10-1 比较器的迟滞窗口

### 10.2 特点

- 中断/唤醒功能；
- 多种比较模式；
- 比较器计数器；
- 睡眠模式下也能工作。

## 10.3 功能框图

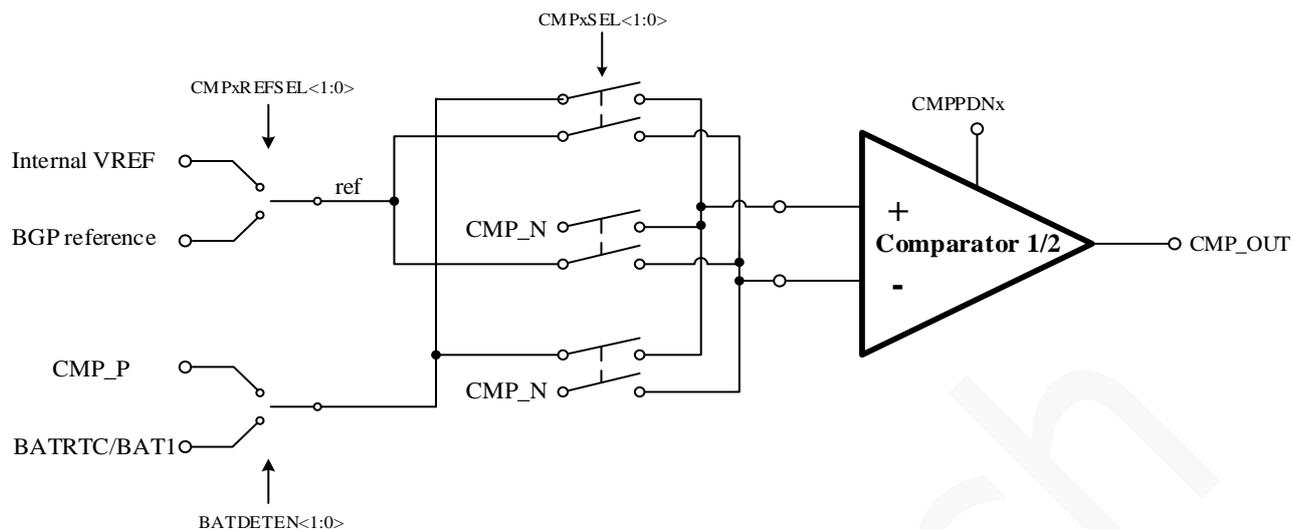


图 10-2 比较器功能框图

## 10.4 寄存器地址

表 10-1 ANA 控制寄存器(ANA 基地址: 0x40014200)

名称	类型	地址	描述	默认值
ANA_REG2	R/W	0x0008	模拟寄存器 2	0x00
ANA_REG3	R/W	0x000C	模拟寄存器 3	0x00
ANA_REG5	R/W	0x0014	模拟寄存器 5	0x00
ANA_REGF	R/W	0x003C	模拟寄存器 F	0x00
ANA_CTRL	R/W	0x0050	模拟控制寄存器	0x0000000
ANA_CMPOUT	R	0x0054	比较结果寄存器	0x0030
ANA_INTSTS	R/C	0x0060	模拟中断状态寄存器	0x00
ANA_INTEN	R/W	0x0064	模拟中断使能寄存器	0x00
ANA_CMPCTL	R/W	0x006C	比较器控制寄存器	0x00000000
ANA_CMPCNT1	R/C	0x00B0	比较器 1 计数器	0x00000000
ANA_CMPCNT2	R/C	0x00B4	比较器 2 计数器	0x00000000
ANA_CMPTHR	R/W	0x00E4	CMP1/CMP2 阈值寄存器	0x0000

## 10.5 寄存器定义

### 10.5.1 ANA\_REG2 寄存器

表 10-2 ANA\_REG2 寄存器

位	名称	类型	描述	默认值
1: 0	CMP1SEL[1 : 0]	R/W	CMP1 信号源选择 00: CMP1_P 与 REF 01: CMP1_N 与 REF 1*: CMP1_P 与 CMP1_N	0
3: 2	CMP2SEL[1 : 0]	R/W	CMP2 信号源选择 00: CMP2_P 与 REF 01: CMP2_N 与 REF 1*: CMP2_P 与 CMP2_N	0
4	CMP1REFSEL	R/W	CMP1 的 REF 选择 0: VREF 1: BGPREF	0
5	CMP2REFSEL	R/W	CMP2 的 REF 选择 0: VREF 1: BGPREF	0

### 10.5.2 ANA\_REG3 寄存器

表 10-3 ANA\_REG3 寄存器

位	名称	类型	描述	默认值
1	CMP1PDN	R/W	CMP1 掉电控制信号 0: 关闭 CMP1 1: 开启 CMP1	0
2	CMP2PDN	R/W	CMP2 掉电控制信号 0: 关闭 CMP2 1: 开启 CMP2	0
3	BGPPD <sup>[1]</sup>	R/W	BGP 掉电控制信号 0: 开启 BGP 1: 关闭 BGP	0

<sup>[1]</sup>: RCH、PLL、ADC 与 BGP 有关系，在开 RCH、PLL、ADC 之前，必须开启 BGP。如果系统中没有使用 BGP，可通过 BGPPD 位关闭 BGP；如果 BGP 有被用到，此时，通过 BGPPD 不能关闭 BGP，因为系统具有保护功能，会重新开启 BGP。浅睡眠或深睡眠模式下，不管 BGP 是否有被用到，都可以关闭 BGP。

### 10.5.3 ANA\_REG5 寄存器

表 10-4 ANA\_REG5 寄存器

位	名称	类型	描述	默认值
1: 0	CMP1IT[1: 0]	R/W	CMP1 的偏置电流选择。 00: 20nA; 01: 100nA; 1*: 500nA;	0
3: 2	CMP2IT[1: 0]	R/W	CMP2 的偏置电流选择 00: 20nA; 01: 100nA; 1*: 500nA;	0

注意: 当 CMP 偏置电流越小时, 功耗越低以及传播延迟时间越长; 当 CMP 偏置电流越大时, 功耗越高以及传播延迟时间越短。

### 10.5.4 ANA\_REGF 寄存器

表 10-5 ANA\_REGF 寄存器

位	名称	类型	描述	默认值
1	BATRTCDETEN	R/W	BATRTC 接入到比较器 2 的正输入端 0: 不接入; 1: 接入。	0
0	BAT1DETEN	R/W	BAT1 接入到比较器 1 的正输入端 0: 不接入; 1: 接入。	0

### 10.5.5 ANA\_CTRL 寄存器

表 10-6 ANA\_CTRL 寄存器

位	名称	类型	描述	默认值
23: 22	CMP2DEB	R/W	比较器 2 消抖控制寄存器。 0: 无消抖。 1: 2 个 RTCCLK 时钟周期消抖。 2: 3 个 RTCCLK 时钟周期消抖。 3: 4 个 RTCCLK 时钟周期消抖。 当使能消抖时, 只有当信号在 RTCCLK 时钟的多周期内没有改变时, 输入信号才有效。此外, 唤醒和中断的响应时间将被延迟, 直到信号有效。该电路可以在所有模式下工作, 包括浅睡眠和深睡眠模式。	0x0
21: 20	CMP1DEB	R/W	比较器 1 消抖控制寄存器。 0: 不消抖。	0x0

			<p>1: 2 个 RTCCLK 时钟周期消抖。</p> <p>2: 3 个 RTCCLK 时钟周期消抖。</p> <p>3: 4 个 RTCCLK 时钟周期消抖。</p> <p>当使能消抖时, 只有当信号在 RTCCLK 时钟的多周期内没有变化时, 输入信号才有效。此外, 唤醒和中断的响应时间将被延迟, 直到信号有效。该电路可以在所有模式下工作, 包括浅睡眠和深睡眠模式。</p>	
19: 18	-	-	保留	0x0
3: 2	CMP2SEL	R/W	<p>该寄存器用于控制比较器 2 的计数器累加条件。</p> <p>0: 关闭</p> <p>1: 比较器 2 输出的上升沿。</p> <p>2: 比较器 2 输出的下降沿。</p> <p>3: 比较器 2 输出的双边沿。</p>	0x0
1: 0	CMP1SEL	R/W	<p>该寄存器用于控制比较器 1 的计数器累加条件。</p> <p>0: 关闭</p> <p>1: 比较器 1 输出的上升沿。</p> <p>2: 比较器 1 输出的下降沿。</p> <p>3: 比较器 1 输出的双边沿。</p>	0x0

## 10.5.6 ANA\_CMPOUT 寄存器

表 10-7 ANA\_CMPOUT 寄存器

位	名称	类型	描述	默认值
3	CMP2	R	该位显示比较器 2 的结果。	0x0
2	CMP1	R	该位显示比较器 1 的结果。	0x0

## 10.5.7 ANA\_INTSTS 寄存器

表 10-8 ANA\_INTSTS 寄存器

位	名称	类型	描述	默认值
3	INTSTS3	R/C	<p>CMP2 的中断标志, 中断产生条件受 ANA_CTRL、ANA_CMPCTL 和 ANA_CMPTHR 控制。</p> <p>读 0: 未发生</p> <p>读 1: 发生</p> <p>写 0: 无效</p> <p>写 1: 清除该位</p>	0x0
2	INTSTS2	R/C	<p>CMP1 的中断标志, 中断产生条件受 ANA_CTRL、ANA_CMPCTL 和 ANA_CMPTHR 控制。</p> <p>读 0: 未发生</p> <p>读 1: 发生</p> <p>写 0: 无效</p> <p>写 1: 清除该位</p>	0x0

## 10.5.8 ANA\_INTEN 寄存器

表 10-9 ANA\_INTEN 寄存器

位	名称	类型	描述	默认值
3	INTEN3	R/W	CMP2 的中断和唤醒使能控制。 0: 禁止 CMP2 中断和唤醒。 1: 使能 CMP2 中断和唤醒。	0x0
2	INTEN2	R/W	CMP1 的中断和唤醒使能控制。 0: 禁止 CMP1 中断和唤醒。 1: 使能 CMP1 中断和唤醒。	0x0

## 10.5.9 ANA\_CMPCNTx 寄存器

表 10-10 ANA\_CMPCNTx 寄存器

位	名称	类型	描述	默认值
31: 0	CNT	R/C	根据 CMPxSEL 的设置, 该寄存器储存比较器 X 的事件发生次数。例如, 当 CMPxSEL 设置为 1, 当 CMPx 输出从 0 变成 1, 那么这个计数器将增加 1。该寄存器可以写 0 清零, 但仅当相应的 CMPPDNx 设置为 1 时有效 (即相应的比较器也被使能时)。	0x0000000

## 10.5.10 ANA\_CMPCTL 寄存器

表 10-11 ANA\_CMPCTL 寄存器

位	名称	类型	描述	默认值
21	CMP2_IO_NODEB	R/W	比较器 2 输出 IO (CMP2_O) 是否输出经过滤波控制的信号。 0: 经过滤波处理后的比较器输出 1: 原始信号输出	0x0
20	CMP2_INT_MASK_EN	R/W	当 CMP2_CNT 值超过 CMP2_THR 阈值, 并且 CMP2_THR_EN = 1, 产生一个中断还是多个中断选择。 0: 每次减法发生都产生一次中断 1: 一个中断	0x0
18: 19	-	-	保留	-
17	CMP1_IO_NODEB	R/W	比较器 1 输出 IO (CMP1_O) 是否输出经过滤波控制的信号。 0: 经过滤波处理后的比较器输出 1: 原始信号输出	0x0
16	CMP1_INT_MASK_EN	R/W	当 CMP1_CNT 值超过 CMP1_THR 阈值, 并且 CMP1_THR_EN = 1, 产生一个中断还是多个中断选择。 0: 每次减法发生都产生一次中断 1: 一个中断	0x0

15 : 12	CMP2_CHK_NUM	R/W	<p>CMP2 输出结果判定窗口设置</p> <p>CMP2 原始输出连续 CMP2_CHK_NUM 次输出为 1，比较器输出结果为 1。</p> <p>CMP2 原始输出连续 CMP2_CHK_NUM 次输出为 0，比较器输出结果为 0。</p> <p>否则，比较器输出结果保持不变</p> <p>0~15: 1 次~16 次</p>	0x0
11	CMP2_THR_EN	R/W	<p>CMP2 计数器减 CMP2_THR 使能</p> <p>0: 比较器 2 保持向上累加</p> <p>1: 如果比较器 2 的计数器值超过 CMP2_THR，计数器值等于计数器值减去 CMP2_THR。直到计数器值小于 CMP2_THR。</p>	0x0
10 : 8	CMP2_CHK_FRQ	R/W	<p>CMP2 数据检测频率</p> <p>0: RTCCLK 时钟周期间隔检测一次</p> <p>1: 7.8125ms 间隔检测一次</p> <p>2: 125ms 间隔检测一次</p> <p>3: 250ms 间隔检测一次</p> <p>4: 500ms 间隔检测一次</p>	0x0
7 : 4	CMP1_CHK_NUM	R/W	<p>CMP1 输出结果判定窗口设置</p> <p>CMP1 原始输出连续 CMP1_CHK_NUM 次输出为 1，比较器输出结果为 1。</p> <p>CMP1 原始输出连续 CMP1_CHK_NUM 次输出为 0，比较器输出结果为 0。</p> <p>否则，比较器输出结果保持不变</p> <p>0~15: 1 次~16 次</p>	0x0
3	CMP1_THR_EN	R/W	<p>CMP1 计数器减 CMP1_THR 使能</p> <p>0: 比较器 1 保持向上累加</p> <p>1: 如果比较器 1 的计数器值超过 CMP1_THR，计数器值等于计数器值减去 CMP1_THR。直到计数器值小于 CMP1_THR。</p>	0x0
2 : 0	CMP1_CHK_FRQ	R/W	<p>CMP1 数据检测频率</p> <p>0: RTCCLK 时钟周期间隔检测一次</p> <p>1: 7.8125ms 间隔检测一次</p> <p>2: 125ms 间隔检测一次</p> <p>3: 250ms 间隔检测一次</p> <p>4: 500ms 间隔检测一次</p>	0x0

### 10.5.11 ANA\_CMPTHR 寄存器

表 10-12 ANA\_CMPTHR 寄存器

位	名称	类型	描述	默认值
---	----	----	----	-----

31 : 16	CMP2_THR	R/W	<p>CMP2 中断阈值设置</p> <p>0: 每次比较器 2 计数器值加 1, 中断标志位都会置位。</p> <p>1~65535: 当比较器 2 计数器值大于该阈值时, 中断标志位会置位</p>	0x0000
15 : 0	CMP1_THR	R/W	<p>CMP1 中断阈值设置</p> <p>0: 每次比较器 1 计数器值加 1, 中断标志位都会置位。</p> <p>1~65535: 当比较器 1 计数器值大于该阈值时, 中断标志位会置位</p>	0x0000

## 第11章 TinyADC 控制器

### 11.1 简介

TinyADC（2 位的 ADC）控制器用来控制 V85XXP 的 TinyADC 功能，TinyADC 是低精度模数转换器。TinyADC 由 AVCC 供电。TinyADC 控制器位于电源主域，所以它在深睡眠下也可以正常工作。

### 11.2 特点

--产生 TinyADC 中断。

### 11.3 功能框图

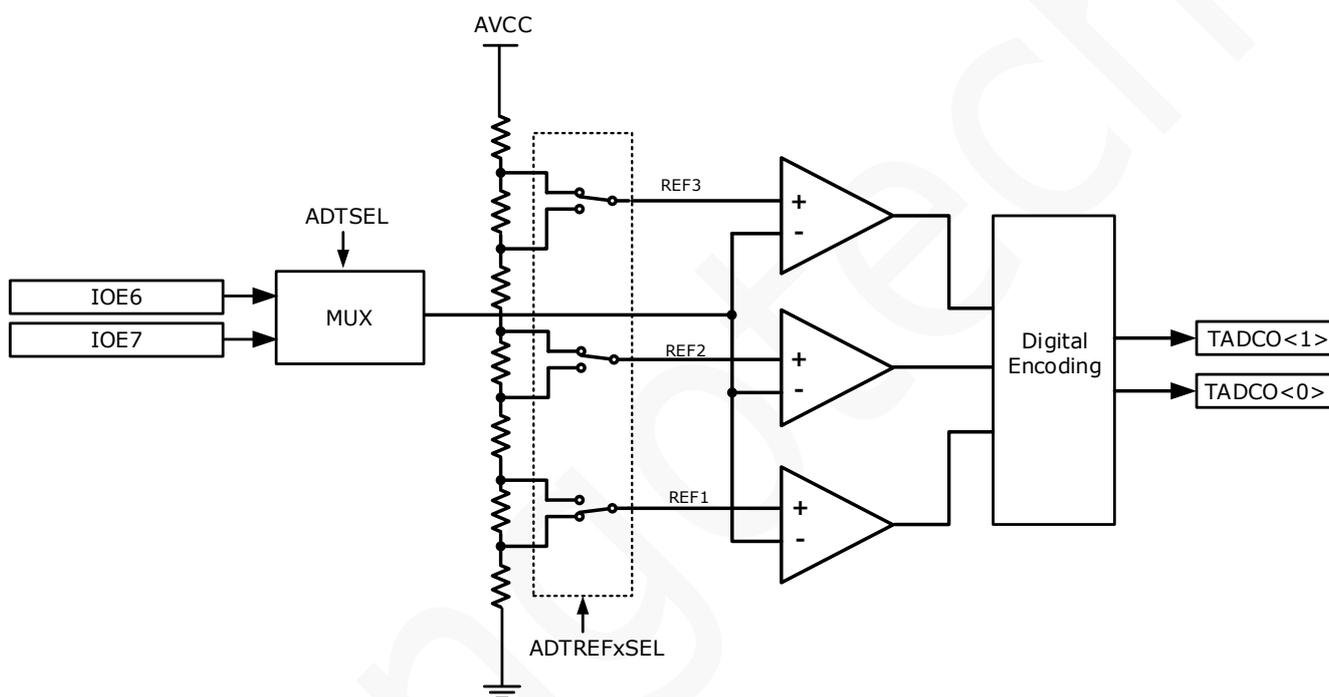


图 11-1 模拟控制器功能框图

### 11.4 寄存器地址

表 11-1 ANA 控制器(ANA 基地址: 0x40014200)

名称	类型	地址	描述	默认值
ANA_REGF	R/W	0x003C	模拟控制寄存器 15	0x00
ANA_CMPOUT	R	0x0054	比较器结果寄存器	0x0030
ANA_INTSTS	R/C	0x0060	模拟中断状态寄存器	0x0000
ANA_INTEN	R/W	0x0064	模拟中断使能寄存器	0x0000
ANA_MISC	R/W	0x00B8	模拟 misc 控制寄存器	0x00

### 11.5 寄存器定义

### 11.5.1 ANA\_REGF 寄存器

表 11-2 ANA\_REGF 寄存器

位	名称	类型	描述	默认值
3	ADTPDN	R/W	Tiny ADC 开关控制 0: 掉电 1: 上电	0
4	ADTSEL	R/W	Tiny ADC 信号输入源选择 0: 从 IOE6 输入 1: 从 IOE7 输入	0
5	ADTREF1SEL	R/W	Tiny ADC 参考电压 1 的选择 0: 0.9V 1: 0.7V	0
6	ADTREF2SEL	R/W	Tiny ADC 参考电压 2 的选择 0: 1.8V 1: 1.6V	0
7	ADTREF3SEL	R/W	Tiny ADC 参考电压 3 的选择 0: 2.7V 1: 2.5V	0

### 11.5.2 ANA\_CMPOUT 寄存器

表 11-3 有关 TADC 的 ANA\_CMPOUT 寄存器

位	名称	类型	描述	默认值
15: 14	TADCO	R	TinyADC 输出。 例如: 如果 ADTREFxSEL 三个位配置为 0, 那么: 信号 < 0.9V, TADCO 输出 0。 信号 > 0.9V && < 1.8V, TADCO 输出 1。 信号 > 1.8V && < 2.7V, TADCO 输出 2。 信号 > 2.7V, TADCO 输出 3。	-

### 11.5.3 ANA\_INTSTS 寄存器

表 11-4 有关 TADC 的 ANA\_INTSTS 寄存器

位	名称	类型	描述	默认值
31: 14	-	-	保留	0
13	INTSTS13	R/C	TADC 过阈值中断标志位。TADC 在上升或下降时, 并且相比上一周期的变化值大于阈值 (TADCTH), 中断标志位将会置位。 读 0: 未发生 读 1: 发生 写 0: 无效	0x0

			写 1：清除该位	
--	--	--	----------	--

### 11.5.4 ANA\_INTEN 寄存器

表 11-5 ANA\_INTEN 寄存器

位	名称	类型	描述	默认值
31: 14	-	-	保留	0
13	INTEN13	R/W	TADC 的中断和唤醒使能（TADC 的变化超过阈值）。 0：禁止 TADC 中断和唤醒。 1：使能 TADC 中断和唤醒。	0x0

### 11.5.5 ANA\_MISC 寄存器

表 11-6 ANA\_MISC 寄存器

位	名称	类型	描述	默认值
31: 6	-	-	保留	0
5: 4	TADCTH	R/W	TADC 阈值设置。该寄存器控制 TADC 产生中断的阈值。TADC 时钟选用内部 32K RC 作为时钟源采样外部信号。当 TADC 的中断使能，并且两个连续周期采样值的差值大于阈值，INTSTS13 就会置 1 并产生中断。	0x0
3: 0	*[1]	-	保留	0

\*[1]：ANA\_MISC 寄存器的[3: 0]位必须写 0。

## 第12章 看门狗

### 12.1 简介

V85XXP内部集成16位的看门狗（WDT）电路，WDT时钟源是RTCCLK。当发生程序跑飞的情况，该定时器将会溢出并产生系统复位，复位标志位通过PMU\_STS寄存器的BIT7查看，见表6-7。WDT复位等级与POR复位等级相同，可提供芯片内部的复位。

### 12.2 特点

--WDT 的复位周期可选为：2 秒\4 秒\8 秒\16 秒；

--WDT 具有密码保护机制，防止对 WDT 的非法访问。

### 12.3 看门狗在不同模式下的状态

看门狗的使能或禁止在不同模式下分别由硬件或是软件控制，下表为 MODE=1（PMU\_STS bit24）时看门狗的状态控制。当 MODE=0 时，看门狗在任何模式下都是无效的。

表 12-1 看门狗在不同模式下状态(MODE=1)

看门狗	电源模式				
	RTC only	深睡眠	浅睡眠	IDLE	工作
开/关	无效	ON(WDTEN=1) OFF(WDTEN=0)	ON(WDTEN=1) OFF(WDTEN=0)	ON(WDTEN=1) OFF(WDTEN=0)	ON(WDTEN=1) OFF(WDTEN=0)
看门狗	退出 RTC only 模式	不同模式唤醒			
		深睡眠	浅睡眠	IDLE	
开/关	ON	ON	ON	ON(WDTEN=1) OFF(WDTEN=0)	

### 12.4 寄存器地址

表 12-2 PMU\_WDT 控制器(PMU 基地址：0x40014000)

名称	类型	地址	描述	默认值
PMU_WDTPASS	R/W	0x0040	看门狗解锁寄存器	0x00000000
PMU_WDTEN	R/W	0x0044	看门狗定时器使能寄存器	0x1
PMU_WDTCLR	R/C	0x0048	清看门狗寄存器	0x0000

### 12.5 寄存器定义

#### 12.5.1 PMU\_WDTPASS 寄存器

表 12-3 PMU\_WDTPASS 寄存器

位	名称	类型	描述	默认值
31: 1	-	-	保留	0

0	UNLOCK	R	该位表示看门狗使能寄存器已被解锁，并准备设置看门狗使能控制寄存器。向该寄存器写入 0xAA5555AA，会将标志位 UNLOCK 置为 1。任何对 PMU 控制寄存器（包括 ICE）的写操作，都会清零该位，因此用户应在 UNLOCK 位置 1 后，立即置位 PMU_WDTEN，否则只能重新继续解锁步骤。	0x0
---	--------	---	--	-----

## 12.5.2 PMU\_WDTEN 寄存器

表 12-4 PMU\_WDTEN 寄存器

位	名称	类型	描述	默认值
31: 4	-	-	保留	0
3: 2	WDTSEL	R/W	设置 WDT 的计数周期。 0: 2 秒 1: 4 秒 2: 8 秒 3: 16 秒 在设置该寄存器之前，首先应将 PMU_WDTPASS 的 UNLOCK 位置 1。	0x0
1	-	-	保留	0
0	WDTEN	R/W	该位表示看门狗定时器使能。在设置该寄存器之前，首先应将 PMU_WDTPASS 的 UNLOCK 位置 1。	0x1

## 12.5.3 PMU\_WDTCLR 寄存器

表 12-5 PMU\_WDTCLR 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	WDTCNT	R/C	该寄存器表示看门狗定时器的当前计数值。当该定时器计数到达 WDTSEL 设置的极限值时，将会发生看门狗系统复位。因此，用户应定时清空此定时器，防止看门狗复位。向该寄存器写入 0x55AAAA55 即可清空该寄存器。该寄存器在调试模式下会自动清零，即调试模式下不会发生看门狗复位。	0x0

## 第13章 RTC 控制器

### 13.1 简介

RTC 控制器用来控制时间的计算、手动校准和实时时钟定时唤醒、闹钟唤醒等功能。时间计算功能可以实现年、月、星期、日、小时、分、秒自动计算功能和闰年监测。手动校正功能，可在通过软件监测温度和调整秒时钟周期以补偿由温度变化引起的时钟偏移。RTC 引擎内置多秒、多分、多小时和 RTC 计数器自动唤醒定时器。

### 13.2 特点

- 支持从秒到年的 BCD 格式的时间计算；
- 自动闰年监测；
- 支持晶体时钟频率补偿；
- 可编程的多唤醒源；
- 为厂家提供手动校正功能；
- RTC 分频在浅睡眠/深睡眠模式下实现低功耗；
- 支持闹钟；
- 支持亚秒。

## 13.3 功能框图

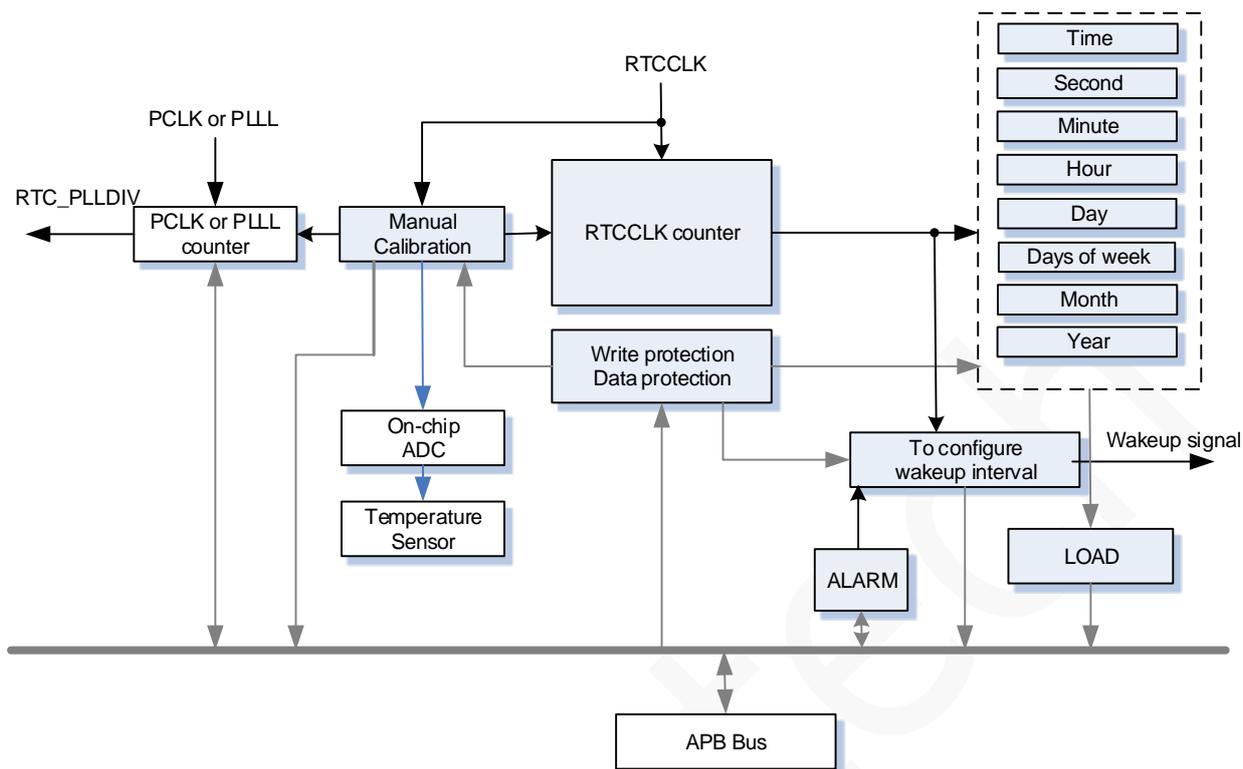


图 13-1 RTC 控制器功能框图

## 13.4 寄存器读写

### 13.4.1 RTC 寄存器写操作

V85XXP 芯片中，部分 RTC 寄存器受写保护，包括 RTC 时间寄存器、RTC 校正寄存器和另外几个寄存器，详情可参考表 13-1。

MCU 应按以下步骤对写保护寄存器进行写操作：

1. 首先用户必须等待 BSY（RTC\_CE 寄存器的 bit1）为 0，以保证 RTC 处于空闲状态；
2. 对 RTC\_PWD 寄存器写 0x5AA55AA5，使能对 RTC\_CE 寄存器的访问；
3. 对 RTC\_CE 寄存器写 0xA55AA55B，使能对带写保护寄存器的写操作；
4. 配置写保护寄存器，其中时间寄存器 RTC\_SEC~RTC\_YEAR，RTC\_TIME 应该同时进行配置；
5. 对 RTC\_PWD 寄存器写 0x5AA55AA5，使能对 RTC\_CE 寄存器的访问；
6. 对 RTC\_CE 寄存器写 0xA55AA55A，清除 CE(RTC\_CE 寄存器的 bit0)标志。当 CE 从 1 清为 0 时，BSY（RTC\_CE 寄存器的 bit1）置 1。当 CE 清为 0 时，写保护寄存器的值将开始更新到 RTC 内核，更新步骤需要 3 个 32K 周期，大约 100μs；
7. 步骤 6 之后，用户需要等待 BSY（RTC\_CE 寄存器的 bit1）自动清 0，以保证 RTC 的更新完成。

不受写保护的 RTC 寄存器，MCU 可以直接进行写操作。

## 13.4.2 RTC 寄存器读操作

RTC 时间寄存器受读保护，详情可参考表 13-1。

MCU 应按以下步骤完成 RTC 时间寄存器的读：

1. 首先用户必须等待 BSY (RTC\_CE 寄存器的 bit1) 为 0，以保证 RTC 处于空闲状态；
2. MCU 读 RTC\_LOAD 寄存器，RTC\_LOAD 寄存器的读操作开始，BSY (RTC\_CE 寄存器的 bit1) 会立马置 1。BSY 在读操作后会自动清 0，读操作需要 3 个 32K 周期，大约 100 $\mu$ s；
3. RTC\_LOAD 寄存器的读操作结束后，当前时间会被锁定，用户可以读取 RTC\_SEC~RTC\_YEAR, RTC\_TIME 寄存器获得时间值。

不受读保护的 RTC 寄存器，MCU 可以直接进行读操作。

## 13.5 寄存器地址

下表列出了每个寄存器的位置。“写保护”是指该寄存器只当 CE 设置为 1 时可写，当 CE 清零时值才会更新到 RTC 寄存器。“读保护”是指仅当 RTC\_LOAD 读操作发生后且 BSY 位清 0，这些寄存器的值会被更新。

表 13-1 RTC 控制器(RTC 基地址：0x40014800)

名称	类型	地址	描述	默认值	写保护	读保护
RTC_SEC	R/W	0x0000	RTC 秒寄存器	--	√	√
RTC_MIN	R/W	0x0004	RTC 分寄存器	--	√	√
RTC_HOUR	R/W	0x0008	RTC 时寄存器	--	√	√
RTC_DAY	R/W	0x000C	RTC 天寄存器	--	√	√
RTC_WEEK	R/W	0x0010	RTC 周寄存器	--	√	√
RTC_MON	R/W	0x0014	RTC 月寄存器	--	√	√
RTC_YEAR	R/W	0x0018	RTC 年寄存器	--	√	√
RTC_TIME	R/W	0x001C	RTC 亚秒计数器寄存器	--	√	√
RTC_WKUSEC	R/W	0x0020	RTC 秒唤醒寄存器	0x00	√	
RTC_WKUMIN	R/W	0x0024	RTC 分唤醒寄存器	0x00	√	
RTC_WKUHOURL	R/W	0x0028	RTC 时唤醒寄存器	0x00	√	
RTC_WKUCNT	R/W	0x002C	RTC 计数唤醒寄存器	0x00000000	√	
RTC_CAL	R/W	0x0030	RTC 校正寄存器	--	√	
RTC_DIV	R/W	0x0034	RTC_PLLDIV 分频寄存器	0x00000000		
RTC_CTL	R/W	0x0038	RTC_PLLDIV 分频器控制寄存器	0x0		
RTC_ITV	R/W	0x003C	RTC ITV 唤醒寄存器	0x0	√	
RTC_SITV	R/W	0x0040	RTC SITV 唤醒寄存器	0x00	√	
RTC_PWD	R/W	0x0044	RTC 密码控制寄存器	0x00000000		
RTC_CE	R/W	0x0048	RTC 写使能控制寄存器	0x0		
RTC_LOAD	R/W	0x004C	RTC 读使能控制寄存器	--		
RTC_INTSTS	R/W	0x0050	RTC 中断状态寄存器	0x000		
RTC_INTEN	R/W	0x0054	RTC 中断使能寄存器	0x000		
RTC_PSCA	R/W	0x0058	RTC 时钟预分频控制寄存器	0x0	√	

RTC_ACTI	R/W	0x0084	晶体顶点温度寄存器	0x1800	√	
RTC_ACF200	R/W	0x0088	RTC200 倍频率寄存器	0x640000	√	
RTC_ACP0	R/W	0x0090	RTC 参数 0 寄存器	0x0000	√	
RTC_ACP1	R/W	0x0094	RTC 参数 1 寄存器	0x0000	√	
RTC_ACP2	R/W	0x0098	RTC 参数 2 寄存器	0x0000	√	
RTC_ACP3	R	0x009C	RTC 参数 3 寄存器	0x0000		
RTC_ACP4	R/W	0x00A0	RTC 参数 4 寄存器	0x0000	√	
RTC_ACP5	R/W	0x00A4	RTC 参数 5 寄存器	0x0000	√	
RTC_ACP6	R/W	0x00A8	RTC 参数 6 寄存器	0x0000	√	
RTC_ACP7	R/W	0x00AC	RTC 参数 7 寄存器	0x0000	√	
RTC_ACK0	R/W	0x00B0	RTC 参数 k0 寄存器	0x0000	√	
RTC_ACK1	R/W	0x00B4	RTC 参数 k1 寄存器	0x0000	√	
RTC_ACK2	R/W	0x00B8	RTC 参数 k2 寄存器	0x0000	√	
RTC_ACK3	R/W	0x00BC	RTC 参数 k3 寄存器	0x0000	√	
RTC_ACK4	R/W	0x00C0	RTC 参数 k4 寄存器	0x0000	√	
RTC_WKUCNTR	R	0x00CC	记录当前 WKUCNT 值	0x000000		
RTC_ACKTEMP	R/W	0x00D0	RTC K 系数温度分段点控制寄存器	0x3C2800EC	√	
RTC_ALARMSEC	R/W	0x00D8	RTC 闹钟功能的秒设置寄存器	0x0000	√	
RTC_ALARMMIN	R/W	0x00DC	RTC 闹钟功能的分钟设置寄存器	0x0000	√	
RTC_ALARMHOUR	R/W	0x00E0	RTC 闹钟功能的小时设置寄存器	0x0000	√	
RTC_ALARMCTL	R/W	0x00E4	RTC 闹钟控制寄存器	0x0000	√	
RTC_ADCUCALK	R/W	0x00E8	ADC Ucal K 系数寄存器	0x546A546A	√	
RTC_ADCMACTL	R/W	0x00EC	ADC 求平均控制寄存器	0x70000000	√	
RTC_ADCDTCTL	R/W	0x00F0	ADC 数据控制	0x80000000	√	

## 13.6 寄存器定义

### 13.6.1 RTC\_SEC/MIN/DAY/WEEK/MONTH/YEAR 寄存器

时间寄存器为 BCD 编码，其中 7~4 位代表十位数，3~0 位为个位数。这些寄存器的值只有当 CE 为高时可以设置，仅当 CE 从 1 清为 0 时，才更新到 RTC 核心。为读取当前时间，用户应该读操作 RTC\_LOAD 寄存器并等待 BSY 清零，否则读到的时间无效值无效。

表 13-2 RTC\_SEC/MIN/DAY/WEEK/MONTH/YEAR 寄存器

寄存器		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0000	RTC_SEC, 0~59	-	S40	S20	S10	S8	S4	S2	S1
0x0004	RTC_MIN, 0~59	-	M40	M20	M10	M8	M4	M2	M1
0x0008	RTC_HOUR, 0~23	-	-	H20	H10	H8	H4	H2	H1
0x000C	RTC_DAY, 1~31	-	-	D20	D10	D8	D4	D2	D1
0x0010	RTC_WEEK, 0~6	-	-	-	-	-	W4	W2	W1

寄存器		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0014	RTC_MON, 1~12	-	-	-	Mo10	Mo8	Mo4	Mo2	Mo1
0x0018	RTC_YEAR, 00~99	Y80	Y40	Y20	Y10	Y8	Y4	Y2	Y1
默认值		X	X	X	X	X	X	X	X

“周”只能由用户设置，RTC 不会自动监测。初始化设置后，RTC 将增加周自动计数。例如，当用户设置 2010/1/1 星期五，RTC 将监测到 2010/01/02 为星期六。W4/W2/W1: 000, 星期日; 001, 星期一; 010、星期二; 011、星期三; 100、星期四; 101、星期五; 110、星期六。

系统只能设置年的最后两位数字，例如 2010 年，RTC\_YEAR 应设置为 0b00010000。

## 13.6.2 RTC\_TIME 寄存器

表 13-3 RTC\_TIME 寄存器

位	名称	类型	描述	默认值
31: 22	-	-	保留	
21: 0	TIME	R/W	亚秒计数器值	0x0

## 13.6.3 RTC\_WKUSEC 寄存器

表 13-4 RTC\_WKUSEC 寄存器

位	名称	类型	描述	默认值
31: 6	-	-	保留	0
5: 0	WKUSEC	R/W	该寄存器用于控制多秒唤醒功能。唤醒周期为(WKUSEC+1)*1秒。当 INTEN[2]为 1 且内部唤醒秒计数达到目标值，INTSTS[2]将被置 1，而唤醒信号也会生效到 PMU 控制器。由 WKUSEC 产生的第一次中断，如果新的 WKUSEC 值不等于当前 WKUSEC 数将会有 <1 秒的误差。如果新的 WKUSEC 等于当前 WKUSEC，第一次中断时间可能有 0~(WKUSEC+1)的变化。为了避免这个问题，向这个寄存器设置一个替代的值（如 0）并通过 RTC_CE 操作更新它的值，然后设置正确的值。	0x00

## 13.6.4 RTC\_WKUMIN 寄存器

表 13-5 RTC\_WKUMIN 寄存器

位	名称	类型	描述	默认值
31: 6	-	-	保留	0
5: 0	WKUMIN	R/W	该寄存器用于控制多分唤醒功能。唤醒周期为(WKUMIN+1)*1分。当 INTEN[3]为 1 且内部唤醒分计数达到目标值，INTSTS[3]将被置 1，而唤醒信号也会生效到 PMU 控制器。由 WKUMIN 产生的第一次中断，如果新的 WKUMIN 值不等于当前 WKUMIN 数，将会有 <1 分钟的误差。如果新的 WKUMIN 等于当前 WKUMIN，第一次中断时间可能有 0~(WKUMIN+1)的变化。为了避免这个问题，向这个寄存器设置一个替代的值（如 0）并通过 RTC_CE 操作更新它的值，然后设置正确的值。	0x00

### 13.6.5 RTC\_WKUHOUR 寄存器

表 13-6 RTC\_WKUHOUR 寄存器

位	名称	类型	描述	默认值
31: 5	-	-	保留	0
4: 0	WKUHOUR	R/W	该寄存器用于控制多小时唤醒功能。唤醒周期为 $(WKUHOUR+1)*1$ 小时。当 INTEN[4] 为 1 且内部唤醒小时计数达到目标值, INTSTS[4] 将被置 1, 而唤醒信号也会生效到 PMU 控制器。由 WKUHOUR 产生的第一次中断, 如果新的 WKUHOUR 值不等于当前 WKUHOUR 数将会有 $<1$ 小时的误差。如果新的 WKUHOUR 等于当前 WKUHOUR, 第一次中断时间可能有 $0\sim(WKUHOUR + 1)$ 的变化。为了避免这个问题, 向这个寄存器设置一个替代的值(如 0)并通过 RTC_CE 操作更新它的值, 然后设置正确的值。	0x00

### 13.6.6 RTC\_WKUCNT 寄存器

表 13-7 RTC\_WKUCNT 寄存器

位	名称	类型	描述	默认值
31: 26	-	-	保留	0
25: 24	CNTSEL	R/W	该寄存器是用来设置 WKUCNT 计数器时钟。 当 PSCA 是 0: 0: 计数器时钟为 32768Hz。 1: 计数器的时钟 2048Hz。 2: 计数器的时钟 512Hz。 3: 计数器的时钟 128Hz。 当 PSCA 是 1: 0: 计数器的时钟 8192Hz。 1: 计数器的时钟 2048Hz。 2: 计数器的时钟 512Hz。 3: 计数器的时钟 128Hz。	0x0
23: 0	WKUCNT	R/W	该寄存器是用来控制 32K 计数器唤醒功能。唤醒时间为: $(WKUCNT+1) * \text{计数器时钟周期}$ , 计数器时钟由 CNTSEL 控制。当 INTEN[6] 为 1 且内部计数器达到 WKUCNT 设定值, 将产生 WKUCNT 周期计数中断或唤醒信号。	0x000000

### 13.6.7 RTC\_CAL 寄存器

表 13-8 RTC\_CAL 寄存器

位	名称	类型	描述	默认值
31: 14	-	-	保留	0
13: 0	CAL	R/W	RTC 32768 校正寄存器, 该寄存器是一个 14 位的补码。RTC 会每 30 秒进行一次校正, 内部计数器会在 1~29 秒计数, 在第 30 秒计为 $[32768-(CAL-1)]$ , 这样方可让 30 秒内的平均 1 秒脉冲成为精确的 1 秒脉冲。CAL 寄存器的 PPM 分辨率为 1.02ppm, 可调范围为 $\pm 8332.3\text{ppm}$ ( $\pm 12$ 分钟/天)。当手	--

		动校正时，需手动更新该寄存器的值。	
--	--	-------------------	--

### 13.6.8 RTC\_DIV 寄存器

表 13-9 RTC\_DIV 寄存器

位	名称	类型	描述	默认值
31: 26	-	-	保留	0
25: 0	RTCDIV	R/W	该寄存器用来配置 RTC_PLLDIV 输出频率。设置 IOA3 (PMU_IOASEL bit3 置 1) 或 IOA7 (PMU_IOASEL bit7 置 1) 为 RTC_PLLDIV 输出功能，同时设置 RTC_PLLDIV 输出使能 (RTC_CTL bit2)，则对应 IOA 输出分频脉冲。输出频率为 $CLK/(2*(RTCDIV+1))$ 。RTC 手动校正时，该寄存器需手动更新。CLK 来源有 PCLK 或 PLL，通过 RTCPLLCLKSEL (RTC_CTL bit4) 配置。	0x0000000

### 13.6.9 RTC\_CTL 寄存器

表 13-10 RTC\_CTL 寄存器

位	名称	类型	描述	默认值
31: 5	-	-	保留	0
4	RTCPLLCLKSEL	R/W	RTC_PLLDIV 时钟源选择 0: PCLK 1: PLLL	0x0
3	-	-	保留	0
2	RTCPLLOE	R/W	RTC_PLLDIV 输出使能，该寄存器用于控制 RTC_PLLDIV 分频器输出。 0: 禁止 RTC_PLLDIV 输出。 1: 使能 RTC_PLLDIV 输出。 RTC_PLLDIV 输出引脚为 IOA[3]或 IOA[7]，应该设置 IOA3 (PMU_IOASEL bit3 置 1) 或 IOA7 (PMU_IOASEL bit7 置 1) 为 RTC_PLLDIV 输出使能。	0x0
1: 0	-	-	保留	0x0

### 13.6.10 RTC\_ITV 寄存器

表 13-11 RTC\_ITV 寄存器

位	名称	类型	描述	默认值
31: 3	-	-	保留	
2: 0	ITV	R/W	该寄存器用于控制 RTC 的唤醒和中断间隔。该寄存器与 RTC_WKUSEC、RTC_WKUMIN、RTC_WKUHOURL 和 RTC_WKUCNT 独立运行，时间间隔的设置都是整秒、整分、整小时，整天触发，所以第一次中断触发可能会有一定误差，例如 ITV 设置为 2，即每 1 小时触发一次，芯片当前时间为 10: 58，则芯片将会在两分钟后即 11: 00 整点触发（不是 11:58 分触发），后续将在 12: 00, 13: 00...每一个整点触发一次。 000: 1 秒; 001: 1 分; 010: 1 小时; 011: 1 天 (00: 00);	0x0

			100: 500 毫秒; 101: 250 毫秒; 110: 125 毫秒; 111: 62.5 毫秒 (当 RTC_SITV 寄存器的 SITVEN 为 0)。	
--	--	--	--	--

### 13.6.11 RTC\_SITV 寄存器

表 13-12 RTC\_SITV 寄存器

位	名称	类型	描述	默认值
31: 7	-	-	保留	
6	SITVEN	R/W	必须先将寄存器 RTC_ITV 设置为 0x07, 再将该位置 1, 才能通过该寄存器的 bit[5: 0]设置秒唤醒间隔时间。	0x0
5: 0	SITV	R/W	设置秒唤醒间隔时间, 实际秒唤醒间隔时间为 (bit[5: 0]+1) 秒。	0x00

### 13.6.12 RTC\_PWD 寄存器

表 13-13 RTC\_PWD 寄存器

位	名称	类型	描述	默认值
31: 1	-	-	保留	0
0	PWDEN	R	该寄存器用于保护 RTC_CE 寄存器的访问。在访问 RTC_CE 之前, 应写密码 0x5AA55AA5 到 RTC_PWD 寄存器, 此时 PWDEN 将置为 1。任何对 RTC_CE 寄存器的写操作, 都会使 PWDEN 自动清零, 这意味着用户在下次访问 RTC_CE 寄存器之前应该再次向该寄存器写 0x5AA55AA5。	0x0

### 13.6.13 RTC\_CE 寄存器

表 13-14 RTC\_CE 寄存器

位	名称	类型	描述	默认值
31: 2	-	-	保留	0
1	BSY	R	该位在 CE 从 1 清零或当 RTC_LOAD 寄存器被 CPU 读取后立即置 1, 标志着 RTC 更新过程或 RTC 读取过程正在进行, 该位在读或写过程完成后自动清零。用户可以查询该位检查 RTC 更新是否完成。更新或读取过程需要大约 3 个 32K 周期, 大约 100 $\mu$ s。	0x0
0	CE	R	该寄存器用于解锁对 RTC 寄存器的访问。该位只能在 PWDEN 置为 1 且将 0xA55AA55B 写入 RTC_CE 寄存器后置 1。该位置为 1 后, 可以对 RTC 寄存器进行编程, 但当该位清零后寄存器值才会真正进行更新。要清除该位, 应将 PWDEN 设置为 1, 并向 RTC_CE 寄存器写入 0xA55AA55A。	0x0

### 13.6.14 RTC\_LOAD 寄存器

表 13-15 RTC\_LOAD 寄存器

位	名称	类型	描述	默认值
31: 0	LOAD	R	用户读该寄存器后, 当前时间将被锁存到时间寄存器, 用户可以	--

			从 RTC_SEC~RTC_YEAR, RTC_TIME 寄存器读取数据。读取过程将需要 3 个 32K 周期, 用户可以检查 BSY (RTC_CE bit1) 位判断读取过程是否完成。从该寄存器读取的数据无效。	
--	--	--	---	--

### 13.6.15 RTC\_INTSTS 寄存器

表 13-16 RTC\_INTSTS 寄存器

位	名称	类型	描述	默认值
31: 11	-	-	保留	--
10	INTSTS10	R/C	闹钟中断标志位	0x0
9	-	-	保留	0x0
8	INTSTS8	R/C	中断状态 8, 发生非法写 RTC_CE 寄存器时, 该位置 1。非法写入是指 BSY 位为 1 时, CE (RTC_CE bit0) 被置 1 或读取 RTC_LOAD 寄存器。写 1 清零该位。	0x0
7	-	-	保留	0x0
6	INTSTS6	R/C	中断状态 6, 当 32K 计数器到达由 WKUCNT 设置的周期时, 该位置 1。写 1 清零该位。	0x0
5	INTSTS5	R/C	中断状态 5, 该时间到达午夜 (00: 00) 时, 该位置 1。写 1 清零该位。	0x0
4	INTSTS4	R/C	中断状态 4, 当到达由 WKUHOURL 设置的多小时中断周期时, 该位置 1。写 1 清零该位。	0x0
3	INTSTS3	R/C	中断状态 3, 当到达由 WKUMIN 设置的多分钟中断周期时, 该位置 1。写 1 清零该位。	0x0
2	INTSTS2	R/C	中断状态 2, 当到达由 WKUSEC 设置的多秒中断周期时, 该位置 1。写 1 清零该位。	0x0
1	INTSTS1	R/C	中断状态 1, 当非法格式的时间写入 RTC 寄存器时, 该位置 1。写 1 清零该位。	0x0
0	INTSTS0	R/C	中断状态 0, 当到达由 ITV 和 SITV 设置的中断周期时, 该位置 1。写 1 清零该位。	0x00

### 13.6.16 RTC\_INTEN 寄存器

表 13-17 RTC\_INTEN 寄存器

位	名称	类型	描述	默认值
31: 11	-	-	保留	0
10	INTEN10	R/W	中断使能 10, 当该位为 1 且 INTSTS10 置 1 时, 将产生闹钟中断。	0x0
9	-	-	保留	--
8	INTEN8	R/W	中断使能 8, 当该位为 1 且 INTSTS8 置 1 时, 将产生 CE 非法访问中断。	0x0
7	-	-	保留	0x0
6	INTEN6	R/W	中断使能 6, 当该位为 1 且 INTSTS6 置 1 时, 将产生 WKUCNT 设置周期计数中断或唤醒信号。	0x0
5	INTEN5	R/W	中断使能 5, 当该位为 1 且 INTSTS5 置 1 时, 将产生午夜计时中断或唤醒信号。	0x0

4	INTEN4	R/W	中断使能 4, 当该位为 1 且 INTSTS4 置 1 时, 将产生多小时中断或唤醒信号。	0x0
3	INTEN3	R/W	中断使能 3, 当该位为 1 且 INTSTS3 置 1 时, 将产生多分钟中断或唤醒信号。	0x0
2	INTEN2	R/W	中断使能 2, 当该位为 1 且 INTSTS2 置 1 时, 将产生多秒中断或唤醒信号。	0x0
1	INTEN1	R/W	中断使能 1, 当该位为 1 且 INTSTS1 置 1 时, 将产生非法格式时间写入中断或唤醒信号。	0x0
0	INTEN0	R/W	中断使能 0, 当该位为 1 且 INTSTS0 置 1 时, 将产生 ITV 和 SITV 设定的中断或唤醒信号。	0x00

### 13.6.17 RTC\_PSCA 寄存器

表 13-18 RTC\_PSCA 寄存器

位	名称	类型	描述	默认值
31: 2	-	-	保留	0
1: 0	PSCA	R/W	该寄存器用于控制 RTCCLK 预分频器。RTCCLK 时钟速率慢时, 可以显著降低浅睡眠或深睡眠模式下的功耗。 0: 无预分频器, RTCCLK 为 32768Hz。 1: 1/4 预分频器, RTCCLK 为 8192Hz。 2~3: 保留 当该寄存器配置为 1 时, 所有使用 RTCCLK 的模块都将受到影响。硬件会自动调整 RTC 计数器时钟和 LCD 帧速率, 因此不需手动修改设置。但是 RTC_WKUCNT 的输入时钟将会改变, 用户需要修改相应的配置; UART32K 波特率寄存器需要根据新的 RTCCLK 重新计算。	0x0

### 13.6.18 RTC\_ACTI 寄存器

表 13-19 RTC\_ACTI 寄存器

位	名称	类型	描述	默认值
31: 14	-	-	保留	0
13: 0	ACTI	R/W	ACTI 表示 RTC 校正期间晶体中心温度的 $T_i$ 值, 其值需要从 Info 区加载。该寄存器是 16 位的补码, 其中, 高 8 位表示整数部分, 低 8 位表示小数部分。例如, 0x1880 表示 24.5 度: $24.5 * 256 = 6272 = 0x1880$ ; 0xE780 表示 -24.5 度: $\sim 0x1880 + 1 = 0xE780$ 。 该寄存器受 RTC 写保护, 当 ACEN 为 1 时应为固定值。	0x1800

### 13.6.19 RTC\_ACF200 寄存器

表 13-20 RTC\_ACF200 寄存器

位	名称	类型	描述	默认值
31: 26	-	-	保留	0
25: 0	F200	R/W	该寄存器存储用于计算 PLLDIV 的值。 假如使用 PCLK 作为 RTCPLLDIV 的时钟源, 即 $PCLK/2$ 。当 PCLK 为 13107200, $PCLK/2 = 6553600 = 0x640000$ 。	0x640000

			假如使用 PLLL 作为 RTCPLLDIV 的时钟源,即 PLLL/2。当 PLLL 为 26214400, PLLL/2=13107200=0xC80000。	
--	--	--	---	--

### 13.6.20 RTC\_ACP0 寄存器

表 13-21 RTC\_ACP0 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	符号扩展位	-
15: 0	P0	R/W	P0 是 16 位的补码,用于温度计算。需从 Info 区加载数据,详情参考表 13-39。该寄存器受 RTC 写保护。	0x0000

### 13.6.21 RTC\_ACP1 寄存器

表 13-22 RTC\_ACP1 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	符号扩展位	-
15: 0	P1	R/W	P1 是 16 位的补码,用于温度计算。需从 Info 区加载数据,详情参考表 13-39。该寄存器受 RTC 写保护。	0x0000

### 13.6.22 RTC\_ACP2 寄存器

表 13-23 RTC\_ACP2 寄存器

位	名称	类型	描述	默认值
31: 0	P2	R/W	P2 是 32 位的补码,用于温度计算。需从 Info 区加载数据,详情参考表 13-39。该寄存器的值计算公式如下: $P2 = P2' + (Tr - Tm) * 256$ P2'为 Info 区 0x80D14 地址存储值。 Tr 为 Info 区 0x80D80 地址存储值。 Tm 为 Info 区 0x80D84 地址存储值。 该寄存器受 RTC 写保护。	0x00000000

### 13.6.23 RTC\_ACP3 寄存器

表 13-24 RTC\_ACP3 寄存器

位	名称	类型	描述	默认值
31: 0	P3	R	P3 是 16 位补码,用于频率校正。需从 Info 区加载数据,详情参考表 13-39。RTC_ACKTEMP 寄存器中的 4 个温度值划分温度段为 5 个区间,匹配校正后的温度 T 处于哪个区间,确定温度区间系数 RTC_ACKx(x=0~4)并将系数存储至 RTC_ACP3 寄存器。该寄存器受 RTC 写保护。	0x00000000

### 13.6.24 RTC\_ACP4 寄存器

表 13-25 RTC\_ACP4 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	符号扩展位	-

15: 0	P4	R/W	P4 是晶体常温偏移量，是 16 位补码。需从 Info 区加载数据，详情参考表 13-39。该寄存器受 RTC 写保护。	0x0000
-------	----	-----	---	--------

### 13.6.25 RTC\_ACP5 寄存器

表 13-26 RTC\_ACP5 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	符号扩展位	-
15: 0	P5	R/W	P5 是 16 位补码，用于 RTC_CAL 寄存器校正值计算。需从 Info 区加载数据，详情参考表 13-39。该寄存器受 RTC 写保护。	0x0000

### 13.6.26 RTC\_ACP6 寄存器

表 13-27 RTC\_ACP6 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	符号扩展位	-
15: 0	P6	R/W	P6 是 16 位补码，用于 RTC_DIV 寄存器存储值计算。P6 计算公式如下： $P6=a*P6'$ ， $P6'$ 为 Info 区加载值，详情参考表 13-39。 假如使用 PCLK 作为 RTCPLLDIV 的时钟源， $a = PCLK/6553600$ 。 假如使用 PLLL 作为 RTCPLLDIV 的时钟源， $a = PLLL/6553600$ 。 该寄存器受 RTC 写保护。	0x0000

### 13.6.27 RTC\_ACP7 寄存器

表 13-28 RTC\_ACP7 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	符号扩展位	-
15: 0	P7	R/W	P7 是 16 位补码，用于频率校正。需从 Info 区加载数据，详情参考表 13-39。该寄存器受 RTC 写保护。	0x0000

### 13.6.28 RTC\_ACKx 寄存器

表 13-29 RTC\_ACKx 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	符号扩展位	-
15: 0	ACK	R/W	RTC_ACK0~RTC_ACK4 是 16 位补码，用于频率校正。需从 Info 区加载数据，详情参考表 13-39。RTC 手动温度补偿时，用户需根据计算的温度 T 手动匹配 RTC_ACKx ( $x=0\sim4$ )，并根据 RTC_ACKx 进行频率校正。不需存储值到 RTC_ACP3 寄存器。该寄存器受 RTC 写保护。  RTC_ACK0: $T < KTEMP1$ 。 RTC_ACK1: $KTEMP1 \leq T < KTEMP2$ 。 RTC_ACK2: $KTEMP2 \leq T < KTEMP3$ 。	0x0000

			RTC_ACK3: $KTEMP3 \leq T < KTEMP4$ 。	
			RTC_ACK4: $T \geq KTEMP4$ 。	

### 13.6.29 RTC\_WKUCNTR 寄存器

表 13-30 RTC\_WKUCNTR 寄存器

位	名称	类型	描述	默认值
31: 24	-	-	保留	0
23: 0	WKUCNTR	R	记录当前 WKUCNT 值。数值递增，数据范围：0 ~ WKUCNT。从 0 开始累加，到达 WKUCNT 设定值后，从 0 开始继续累加。	--

### 13.6.30 RTC\_ACKTEMP 寄存器

表 13-31 RTC\_ACKTEMP 寄存器

位	名称	类型	描述	默认值
31: 24	KTEMP4	R/W	KTEMP4 是一个 8 位的补码，用来控制温度分段节点 4。需从 Info 区加载数据，详情参考表 13-39。该寄存器受 RTC 写保护。	0x3C
23: 16	KTEMP3	R/W	KTEMP3 是一个 8 位的补码，用来控制温度分段节点 3。需从 Info 区加载数据，详情参考表 13-39。该寄存器受 RTC 写保护。	0x28
15: 8	KTEMP2	R/W	KTEMP2 是一个 8 位的补码，用来控制温度分段节点 2。需从 Info 区加载数据，详情参考表 13-39。该寄存器受 RTC 写保护。	0x00
7: 0	KTEMP1	R/W	KTEMP1 是一个 8 位的补码，用来控制温度分段节点 1。需从 Info 区加载数据，详情参考表 13-39。该寄存器受 RTC 写保护。	0xEC

### 13.6.31 RTC\_ALARMSEC 寄存器

表 13-32 RTC\_ALARMSEC 寄存器

位	名称	类型	描述	默认值
31: 7	-	-	保留	
6: 0	ALARMSEC	R/W	此寄存器用于闹钟功能的秒设置	0x0

### 13.6.32 RTC\_ALARMMIN 寄存器

表 13-33 RTC\_ALARMMIN 寄存器

位	名称	类型	描述	默认值
31: 7	-	-	保留	
6: 0	ALARMMIN	R/W	此寄存器用于闹钟功能的分设置	0x0

### 13.6.33 RTC\_ALARMHOUR 寄存器

表 13-34 RTC\_ALARMHOU 寄存器

位	名称	类型	描述	默认值
31: 6	-	-	保留	
5: 0	ALARMHOUR	R/W	此寄存器用于闹钟功能的小时设置	0x0

### 13.6.34 RTC\_ALARMCTL 寄存器

表 13-35 RTC\_ALARMCTL 寄存器

位	名称	类型	描述	默认值
31: 3	-	-	保留	
2	TIME_CNT_EN	R/W	此寄存器用于控制亚秒计数器是否启用 0: 不启用亚秒计数器 1: 启用亚秒计数器	0x0
1	-	R/W	保留, 用户必须配置为 1	0x0
0	ALARM_EN	R/W	此寄存器用于控制闹钟功能是否启用 0: 禁用闹钟 1: 启用闹钟	0x0

### 13.6.35 RTC\_ADCUCALK 寄存器

表 13-36 RTC\_ADCUCALK 寄存器

位	名称	类型	描述	默认值
31: 16	UCAL_K3	R/W	测温通道 UCAL K3, 用户必须配置为 0x599A	0x546A
15: 0	UCAL_K1	R/W	测温通道 UCAL K1, 用户必须配置为 0x599A	0x546A

### 13.6.36 RTC\_ADCMACTL 寄存器

表 13-37 RTC\_ADCMACTL 寄存器

位	名称	类型	描述	默认值
31: 28	-	-	保留	0x7
27	-	R/W	保留, 用户必须配置为 0	0
26: 24	-	-	保留	0
23: 20	SKIP_SAMPLE	R/W	该寄存器用来选择忽略最开始的 ADC 输出采样点的数量。 SKIP_SAMPLE[3: 0] = x: 忽略最开始的 x 个采样点。 0: 忽略 0 个采样点 4: 忽略 4 个采样点 8: 忽略 8 个采样点 12: 忽略 12 个采样点	0x0

			其它：保留	
19	-	-	保留	0x0
18 : 16	AVERAGE_SAMPLE	R/W	求平均的采样点的数量 0: 2 点求平均 1: 4 点求平均 2: 8 点求平均 3: 16 点求平均 4: 32 点求平均 5: 64 点求平均 6: 保留 7: 保留	0x0
15 : 0	AVERAGE_CHx	R/W	每个位分别控制对应通道的求平均使能 1: 使能 0: 禁止 AVERAGE_CH15~0 分别对应 ADC 通道 15~0。	0x0

### 13.6.37 RTC\_ADCDTCTL 寄存器

表 13-38 RTC\_ADCDTCTL 寄存器

位	名称	类型	描述	默认值
31:30	ENDED_IN2ADC_CONVERT	R/W	必须配置为 2	0x2
29:0	-	-	保留	-

## 13.7 Info 参数寄存器(与 RTC 温度补偿有关)

Info 信息存储在以 0x00080800 为起始地址的区域，只能读不能写。所有信息都有双备份，表格中第一份数据用 1 表示，第二份用 2 表示。每份数据都带校验（Checksum），校验算法：各份中的数据项的数据相加，对该相加后的结果取反。

其中地址 0x00080800~0x0008085C 数据，通过专用烧写工具（脱机烧写并开启 RTC 校正功能）写入。其它数据出厂前已写入。

表 13-39 Info 信息寄存器(与 RTC 温度补偿有关)

地址	符号	信息	描述
0x00080800	P4	晶体常温偏移 1	unit(0.1ppm), 低 16 位加载到 RTC_ACP4 寄存器, 比如 0。
0x00080804		Checksum1	INV(SUM(0x80800, 0x80800))
0x00080808	P4	晶体常温偏移 2	unit(0.1ppm), 低 16 位加载到 RTC_ACP4 寄存器, 比如 0。
0x0008080C		Checksum2	INV(SUM(0x80808, 0x80808))
0x00080810	K0	晶体 K0 系数 1	加载到 RTC_ACK0 寄存器, K0 计算公式如下: $K0=B0/1000000*65536$ , B0 为晶振曲线二次函数的分

			段系数。K0 值比如 20827。
0x00080814	K1	晶体 K1 系数 1	加载到 RTC_ACK1 寄存器，K1 计算公式如下： $K1=B1/1000000*65536$ ，B1 为晶振曲线二次函数的分段系数。K1 值比如 21496。
0x00080818	K2	晶体 K2 系数 1	加载到 RTC_ACK2 寄存器，K2 计算公式如下： $K2=B2/1000000*65536$ ，B2 为晶振曲线二次函数的分段系数。K2 值比如 22020。
0x0008081C	K3	晶体 K3 系数 1	加载到 RTC_ACK3 寄存器，K3 计算公式如下： $K3=B3/1000000*65536$ ，B3 为晶振曲线二次函数的分段系数。K3 值比如 24517。
0x00080820	K4	晶体 K4 系数 1	加载到 RTC_ACK4 寄存器，K4 计算公式如下： $K4=B4/1000000*65536$ ，B4 为晶振曲线二次函数的分段系数。K4 值比如 25257。
0x00080824	-	Checksum1	INV(SUM(0x80810, 0x80820))
0x00080828	K0	晶体 K0 系数 2	加载到 RTC_ACK0 寄存器，K0 计算公式如下： $K0=B0/1000000*65536$ ，B0 为晶振曲线二次函数的分段系数。K0 值比如 20827。
0x0008082C	K1	晶体 K1 系数 2	加载到 RTC_ACK1 寄存器，K1 计算公式如下： $K1=B1/1000000*65536$ ，B1 为晶振曲线二次函数的分段系数。K1 值比如 21496。
0x00080830	K2	晶体 K2 系数 2	加载到 RTC_ACK2 寄存器，K2 计算公式如下： $K2=B2/1000000*65536$ ，B2 为晶振曲线二次函数的分段系数。K2 值比如 22020。
0x00080834	K3	晶体 K3 系数 2	加载到 RTC_ACK3 寄存器，K3 计算公式如下： $K3=B3/1000000*65536$ ，B3 为晶振曲线二次函数的分段系数。K3 值比如 24517。
0x00080838	K4	晶体 K4 系数 2	加载到 RTC_ACK4 寄存器，K4 计算公式如下： $K4=B4/1000000*65536$ ，B4 为晶振曲线二次函数的分段系数。K4 值比如 25257。
0x0008083C	-	Checksum2	INV(SUM(0x80828, 0x80838))
0x00080840	ACTI	晶体顶点温度 1	加载到 RTC_ACTI 寄存器，比如：0x1800。
0x00080844	-	Checksum1	INV(SUM(0x80840, 0x80840))
0x00080848	ACTI	晶体顶点温度 2	加载到 RTC_ACTI 寄存器，比如：0x1800。
0x0008084C	-	Checksum2	INV(SUM(0x80848, 0x80848))
0x00080850	KTEMP x(x=4 ~1)	晶体 K 系数温度分段 区间设置 1	加载到 RTC_ACKTEMP 寄存器，比如 0x3C2800EC。
0x00080854	-	Checksum1	INV(SUM(0x80850, 0x80850))
0x00080858	KTEMP x(x=4 ~1)	晶体 K 系数温度分段 区间设置 2	加载到 RTC_ACKTEMP 寄存器，比如 0x3C2800EC。
0x0008085C	-	Checksum2	INV(SUM(0x80858, 0x80858))
...	-	-	保留
0x00080D10	P1/P0	RTC_ACP1/0 设置 1	高 16 位加载到 RTC_ACP1 寄存器，比如：1060；低 16 位加载到 RTC_ACP0 寄存器，比如：-214。
0x00080D14	P2'	RTC_ACP2 设置 1	该地址存储值记为 P2'，比如：-19746971。根据公式： $P2=P2'+(Tr-Tm)*256$ ，将 P2 加载到 RTC_ACP2 寄存

			器。
0x00080D18	P5	RTC_ACP5 设置 1	高 16 位加载到 RTC_ACP5 寄存器, 比如: 6444; 低 16 位舍弃。
0x00080D1C	P7/P6'	RTC_ACP7/6 设置 1	高 16 位加载到 RTC_ACP7 寄存器, 比如: 0; 低 16 位加载到 P6'。比如: 1342。 根据公式 $P6 = a * P6'$ , 将 P6 加载至 RTC_ACP6 寄存器。
0x00080D20	-	Checksum1	INV(SUM(0x80D10, 0x80D1C))
0x00080D24	P1/P0	RTC_ACP1/0 设置 2	同备份 1
0x00080D28	P2'	RTC_ACP2 设置 2	
0x00080D2C	P5	RTC_ACP5 设置 2	
0x00080D30	P7/P6'	RTC_ACP7/6 设置 2	
0x00080D34	-	Checksum2	
0x00080D38	-	AVCC gain 1	Pre-trim result/3.3 * 10000
0x00080D3C	-	DVCC gain 1	Pre-trim result/1.5 * 10000
0x00080D40	-	BGP gain 1	Pre-trim result/1.2 * 10000
0x00080D44	-	RCL gain 1	Pre-trim result/32768 * 10000
0x00080D48	-	RCH gain 1	Pre-trim result/6553600 * 10000
0x00080D4C	-	Checksum1	INV(SUM(0x80D38, 0x80D48))
0x00080D50	-	AVCC gain 2	同备份 1
0x00080D54	-	DVCC gain 2	
0x00080D58	-	BGP gain 2	
0x00080D5C	-	RCL gain 2	
0x00080D60	-	RCH gain 2	
0x00080D64	-	Checksum2	
0x00080D68	-	ID word 0, Backup 1	
0x00080D6C	-	ID word 1, Backup 1	
0x00080D70	-	Checksum1	INV(SUM(0x80D68, 0x80D6C))
0x00080D74	-	ID word 0, Backup 2	同备份 1
0x00080D78	-	ID word 1, Backup 2	
0x00080D7C	-	Checksum2	
0x00080D80	-	实际温度 Tr 1 (来自 tmp275)	根据公式: $P2 = P2' + (Tr - Tm) * 256$ , 将 P2 加载到 RTC_ACP2 寄存器
0x00080D84	-	测量温度 Tm 1 (来自 ADC)	根据公式: $P2 = P2' + (Tr - Tm) * 256$ , 将 P2 加载到 RTC_ACP2 寄存器
0x00080D88	-	Checksum1	INV(SUM(0x80D80, 0x80D84))
0x00080D8C	-	实际温度 Tr 2 (来自 tmp275)	同备份 1
0x00080D90	-	测量温度 Tm 2 (来自 ADC)	

0x00080D94	-	Checksum2	
...	-	-	保留

## 13.8 校正流程

### 13.8.1 温度校正

根据下式计算温度  $T$ ， $T$  为实际温度的 256 倍。

$$T = ((P0 * ((X * X) >> 16)) + P1 * X + P2) >> 8 \quad \text{Equation 13-1}$$

其中  $X$  为 ADC 温度采样值，为 16 位补码。

$P0 \sim P2$  是寄存器 RTC\_ACP0~RTC\_ACP2 的存储值，被用作温度校正。

实际温度计算结果为： $T' = T / 256$

### 13.8.2 频率误差校正

(1) 芯片频率误差计算公式如下：

$$\Delta = (P3 * ((T - Ti)^2 >> 16)) >> 16 + P4 \quad \text{Equation 13-2}$$

其中， $\Delta$  是 32K 晶体的频率偏差率（单位 0.1ppm）。

$T$  为温度计算值，根据公式 Equation 13-1 计算得到。

$Ti$  为 RTC 校正期间晶体中心温度的  $Ti$  值，即 RTC\_ACTI 寄存器存储值。

$P4$  为晶体常温偏差值。

$P3$  为当前温度  $T$  所在温度区间的  $K$  系数，RTC 手动温度补偿时，用户需根据计算的温度  $T$  手动匹配 ACKx ( $x=0 \sim 4$ )，并根据 ACKx 进行频率校正。不需将值存储到 RTC\_ACP3 寄存器中。

(2) 频率校正值计算公式如下：

$$\text{RTC\_CAL} = (\Delta * P5) >> 16 + 1 \quad \text{Equation 13-3}$$

$$P5 = 65536 / 10 / P_x$$

其中  $P_x$  为寄存器 RTC\_CAL 的 PPM 分辨率。

$P_x = \frac{1}{30} * \frac{\text{CLK}}{32768}$ 。CLK 为 RTCCLK 分频，有 1 分频和 4 分频两种。

(3) PLL 分频秒脉冲输出计算公式如下：

$$\text{RTC\_DIV} = \frac{\text{CLK}}{2} - (\Delta * P6) >> 12 - 1 \quad \text{Equation 13-4}$$

$$P6 = \text{CLK} * 2.048 * 10^{-4} \quad \text{Equation 13-5}$$

CLK 为 RTCPLLDIV 分频前的频率值，来源有 PCLK 或 PLLL，通过 RTCPLLCLKSEL 配置。

## 13.8.3 RTC 手动温度补偿

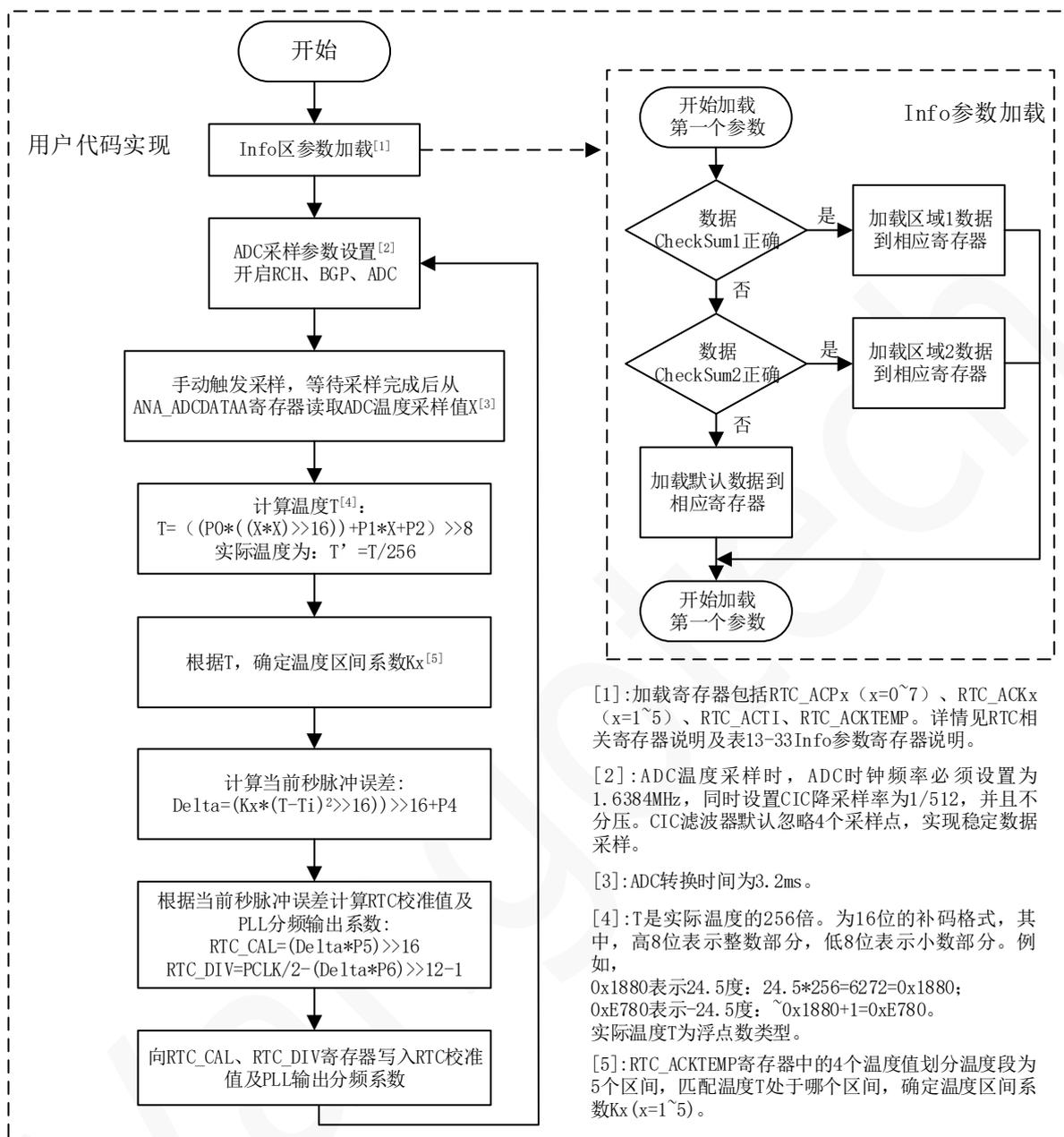


图 13-2 RTC 手动温度补偿流程图

## 第14章 FLASH 控制器

### 14.1 简介

FLASH 控制器用于控制内置 FLASH 的读/写程序。它支持字节/半字/字程序和扇区/芯片擦除。可编程时序可由 MISC2 控制器中的 CYCLE\_1US 控制。FLASH 控制器中的设置将在从浅睡眠或深睡眠模式唤醒后复位，用户应在从这两种状态唤醒后手动恢复设置。

### 14.2 特点

- 支持字节/半字/字编程;
- 支持扇区/芯片擦除;
- 支持 standby (普通待机) 模式, 当 MCU 没有访问 flash, flash 自动进入 standby 模式。任何对 flash 的访问操作, 都会让 flash 从 standby 唤醒, 唤醒时间为 0;
- 支持 deep-standby (深度待机) 模式, 当 MCU 进入 sleep 模式或 deepsleep 模式, flash 自动进入 deep-standby 模式。同时支持手动配置 flash 的 FLASH\_DSTB 寄存器进入 deep-standby 模式。任何对 flash 的访问操作, 都会让 flash 从 deep-standby 唤醒, 唤醒时间为 10 个 1 $\mu$ stick 时间;
- 当程序在 FLASH 中运行, 并且 FLASH 未准备就绪时, MCU 会自动挂起。当 FLASH 准备就绪, MCU 自动从挂起中恢复;
- 支持 Info 扇区读;
- 支持 FLASH 配置寄存器读/写;
- 可编程的编程速度;
- 支持硬件自动校验和比对功能, 支持校验和错误中断;
- 支持读保护、写保护、ICE 保护。

### 14.3 功能框图

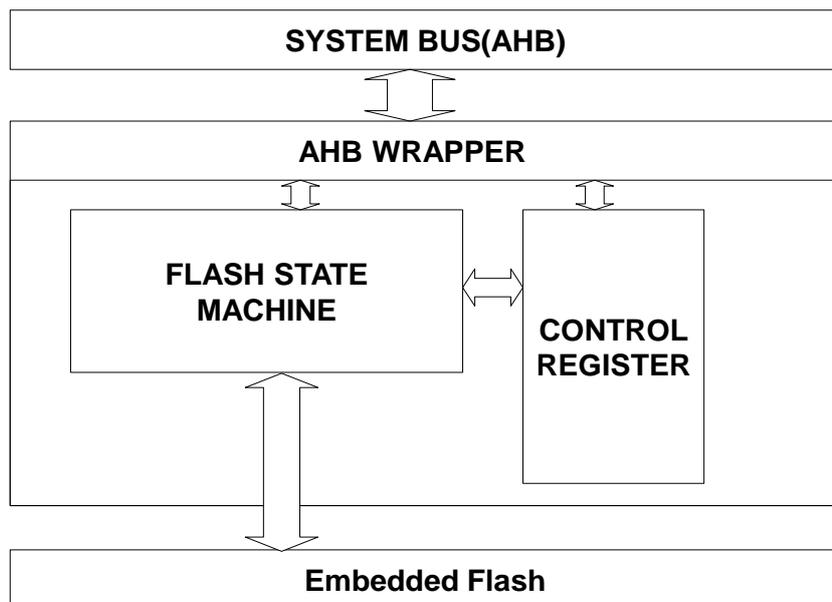


图 14-1 FLASH 控制器的功能框图

### 14.4 寄存器地址

表 14-1 FLASH 控制器地址(FLASH 控制器基地址: 0x00000000)

名称	类型	地址	描述	默认值
FLASH_ICEPROT	R/W	0xFFFA8	ICE 保护寄存器	--
FLASH_RDPROT	R	0xFFAC	Flash 读保护状态寄存器	0x00000000
FLASH_WRPROT	R/W	0xFFB0	Flash 写保护控制寄存器	0x00000000
FLASH_STS	R	0xFFBC	FLASH 编程状态寄存器	0x00
FLASH_INTSTS	R/C	0xFFCC	FLASH 校验和中断状态	0x0
FLASH_CSSADDR	R/W	0xFFD0	FLASH 校验和开始地址	0x00000
FLASH_CSEADDR	R/W	0xFFD4	FLASH 校验和结束地址	0xFFFF
FLASH_CSVALUE	R	0xFFD8	FLASH 校验和计算值寄存器	--
FLASH_CSCVALUE	R/W	0xFFDC	FLASH 校验和预期值设置寄存器	0x00000000
FLASH_PASS	R/W	0xFFE0	FLASH 密码寄存器	0x00
FLASH_CTRL	R/W	0xFFE4	FLASH 控制寄存器	0x00
FLASH_PGADDR	R/W	0xFFE8	FLASH 编程地址寄存器	0x00000
FLASH_PGDATA	R/W	0xFFEC	FLASH 编程字节数据寄存器	--
FLASH_PGB0	R/W	0xFFEC	FLASH 编程字节数据寄存器 0	--
FLASH_PGB1	R/W	0xFFED	FLASH 编程字节数据寄存器 1	--
FLASH_PGB2	R/W	0xFFEE	FLASH 编程字节数据寄存器 2	--
FLASH_PGB3	R/W	0xFFEF	FLASH 编程字节数据寄存器 3	--
FLASH_PGHW0	R/W	0xFFEC	FLASH 编程半字节数据寄存器 0	--
FLASH_PGHW1	R/W	0xFFEE	FLASH 编程半字节数据寄存器 1	--

FLASH_SERASE	R/W	0xFFFF4	FLASH 扇区擦除控制寄存器	0x00
FLASH_CERASE	R/W	0xFFFF8	FLASH 芯片擦除控制寄存器	0x00
FLASH_DSTB	R/W	0xFFFFC	FLASH deep-standby 控制寄存器	0x00

表 14-2 MISC2 控制器(MISC2 基地址: 0x40013E00)

INTSTS	INTSTS	INTSTS	INTSTS	INTSTS
MISC2_FLASHWC	R/W	0x0000	FLASH 等待周期寄存器	0x2100

## 14.5 寄存器定义

### 14.5.1 FLASH\_PASS 寄存器

表 14-3 FLASH\_PASS 寄存器

位	名称	类型	描述	默认值
31: 1	-	-	保留	0
0	UNLOCK	R	UNLOCK 位用于指示 FLASH 是否已解锁。要设置此位, 用户应将 0x55AAAA55 写入此寄存器。该位将保持高电平, 直到用户将任何其他值写入该寄存器。如果 UNLOCK 位为 0, 则将禁止所有 FLASH 编程或擦除或写配置操作。	0x0

### 14.5.2 FLASH\_CTRL 寄存器

表 14-4 FLASH\_CTRL 寄存器

位	名称	类型	描述	默认值
31: 3	-	-	保留	0
2	CSINTEN	R/W	该寄存器用于控制校验和错误的中断使能。 0: 禁止校验和错误中断。 1: 使能校验和错误中断。	0x0
1: 0	CSMODE	R/W	该寄存器用于控制校验和工作模式, 用于控制校验和计算以及比对。 0: 关闭校验和模式。 1: 校验和计算和比对工作始终进行。 2: 每一次定时器 2 溢出时做一次校验和计算和比对。 3: 每个 RTC 秒脉冲的上升沿做一次校验和计算和比对。	0x0

### 14.5.3 FLASH\_PGADDR 寄存器

表 14-5 FLASH\_PGADDR 寄存器

位	名称	类型	描述	默认值
31: 18	-	-	保留	0
17: 0	PGADDR	R/W	flash 编程地址设置。这是 FLASH IP 的字节地址, 有关每种模式下的地址的详细信息, 请参见表 14-6。当编程成功完成后, 该地址将自动递增。	0x000000

表 14-6 PGADDR 及写数据寄存器

模式	PGADDR[17: 2]	PGADDR[1: 0]	写数据寄存器
字编程模式	0x0000~0xFFFF	0	FLASH_PGDATA
字节编程模式	0x0000~0xFFFF	0	FLASH_PGB0
		1	FLASH_PGB1
		2	FLASH_PGB2
		3	FLASH_PGB3
半字编程模式	0x0000~0xFFFF	0	FLASH_PGHW0
		2	FLASH_PGHW1
页擦除	0x0000~0xFFFF	0	FLASH_SERASE

#### 14.5.4 FLASH\_PGDATA 寄存器

表 14-7 FLASH\_PGDATA 寄存器

位	名称	类型	描述	默认值
31: 0	PGDATA	R/W	该寄存器用于控制字编程数据。当 UNLOCK 为 1 时，写入该寄存器将自动触发一个 4 字节的编程。在此期间禁止对 FLASH IP 的任何访问，直到编程完成。FLASH 控制器内部有两个 32 位 FIFO，因此用户可以在没有中断的情况下进行连续编程。但是如果用户需要访问 FLASH IP，比如执行程序，在这段时间内 FLASH 编程的速度会变慢，因为 FLASH IP 需要更多的启动时间。建议将编程程序放在 SRAM 中，这样可以实现最大编程速度。	

#### 14.5.5 FLASH\_PGBx 寄存器

表 14-8 FLASH\_PGBx 寄存器

位	名称	类型	描述	默认值
7: 0	PGBx	R/W	该寄存器用于控制字节编程数据。当 UNLOCK 为 1 时，写入该寄存器将自动触发一个 1 字节的编程。在在此期间禁止对 FLASH IP 的任何访问，直到编程完成。FLASH 控制器内部有一个双 8 位 FIFO，因此用户可以在没有中断的情况下进行连续编程。但是如果用户需要访问 FLASH IP，比如执行程序，在这段时间内 FLASH 编程的速度会变慢，因为 FLASH IP 需要更多的启动时间。建议将编程程序放在 SRAM 中，这样可以实现最大编程速度。	

#### 14.5.6 FLASH\_PGHwx 寄存器

表 14-9 FLASH\_PGHwx 寄存器

位	名称	类型	描述	默认值
---	----	----	----	-----

15: 0	PGHWx	R/W	该寄存器用于控制半字编程数据。当 UNLOCK 为 1 时，写入该寄存器将自动触发一个 2 字节的编程。在此期间禁止对 FLASH IP 的任何访问，直到编程完成。FLASH 控制器内部有一个双 16 位 FIFO，因此用户可以在没有中断的情况下进行连续编程。但是如果用户需要访问 FLASH IP，比如执行程序，在这段时间内 FLASH 编程的速度会变慢，因为 FLASH IP 需要更多的启动时间。建议将编程程序放在 SRAM 中，这样可以实现最大编程速度。	
-------	-------	-----	---	--

### 14.5.7 FLASH\_SERASE 寄存器

表 14-10 FLASH\_SERASE 寄存器

位	名称	类型	描述	默认值
31: 1	-	-	保留	0
0	SERASE	R	该位用于指示扇区擦除是否正在进行。扇区大小是 1024Byte。当 UNLOCK 为 1 且用户将 0xAA5555AA 写入该寄存器时，该位置 1，并且在扇区擦除完成后将自动清零。扇区擦除的地址由 PGADDR[17: 2]控制。扇区擦除时间为 4000*1μs。	0x0

### 14.5.8 FLASH\_CERASE 寄存器

表 14-11 FLASH\_CERASE 寄存器

位	名称	类型	描述	默认值
31: 1	-	-	保留	0
0	CERASE	R	该位用于指示芯片全擦除是否正在进行。当 UNLOCK 为 1 并且用户将 0xAA5555AA 写入寄存器，该位置 1，芯片擦除完成后该位将自动清零。芯片擦除时间为 30000*1μs。	0x0

### 14.5.9 FLASH\_DSTB 寄存器

表 14-12 FLASH\_DSTB 寄存器

位	名称	类型	描述	默认值
31: 1	-	-	保留	0
0	DSTB	R	<p>该位用于指示 FLASH IP 是否正在进入 deep-standby。</p> <p>当 UNLOCK 为 1 且用户将 0xAA5555AA 写入该寄存器时，该位置 1，FLASH IP 进入 deep-standby 模式。</p> <p>当 FLASH IP 从 deep-standby 模式唤醒后，该位自动清零。deep-standby 模式可以通过对 FLASH IP 的任何访问退出，但是需要 10 个 1μs 滴答的时间才能使 FLASH IP 恢复正常状态。在此期间，对 FLASH 的任何访问将被过滤，直到 FLASH 准备就绪。</p> <p>深睡眠和浅睡眠模式下，FLASH 自动进入 deep-standby。</p> <p>IDLE 模式下，用户需要在 SRAM 中运行代码，手动配置该寄存器。</p>	0x0

### 14.5.10 FLASH\_INTSTS 寄存器

表 14-13 FLASH\_INTSTS 寄存器

位	名称	类型	描述	默认值
31: 1	-	-	保留	0
0	CSERR	R/C	校验和错误状态位。该标志用于指示以前的校验和错误，也就是说与 CSVALUE/CSCVALUE 不一致。当 CSINTEN 为 1 时，将产生一个中断给 CPU。写 1 清零该位。	0x0

### 14.5.11 FLASH\_CSSADDR 寄存器

表 14-14 FLASH\_CSSADDR 寄存器

位	名称	类型	描述	默认值
31: 19	-	-	保留	0
18: 0	CSSADDR	R/W	校验和起始地址寄存器。该寄存器用于控制校验和的起始地址。该值为该寄存器的字节地址，但最低 2 位始终为 0。	0x00000

### 14.5.12 FLASH\_CSEADDR 寄存器

表 14-15 FLASH\_CSEADDR 寄存器

位	名称	类型	描述	默认值
31: 19	-	-	保留	0
18: 0	CSEADDR	R/W	校验和结束地址寄存器。该寄存器用于控制校验和的结束地址。该值是校验和结束地址的字节地址，但最低 2 位始终为 0。校验和范围为 (CSSADDR=<ADDR=<CSEADDR)。	0x7FFFC

### 14.5.13 FLASH\_CSVALUE 寄存器

表 14-16 FLASH\_CSVALUE 寄存器

位	名称	类型	描述	默认值
31: 0	CSVALUE	R	校验和计算值寄存器。该寄存器用于表示当前 MCU 自动计算的校验和结果。	0x00000000

### 14.5.14 FLASH\_CSCVALUE 寄存器

表 14-17 FLASH\_CSCVALUE 寄存器

位	名称	类型	描述	默认值
31: 0	CSCVALUE	R/W	校验和预期值设置寄存器。该寄存器用于设置预期的校验和结果。当校验和完成，MCU 自动计算的校验和值 CSVALUE 与用户设置的预期校验值 CSCVALUE 不匹配时，CSERR 标志将被置 1。校验和预期值的计算方法：将该范围 (CSSADDR=<ADDR=<CSEADDR) 内的值 (无符号 32 位数据类型) 全部累加，累加结果保留低 32 位。	0x00000000

### 14.5.15 FLASH\_STS 寄存器

表 14-18 FLASH\_STS 寄存器

位	名称	类型	描述	默认值
31: 5	-	-	保留	0
4: 0	STS	R	FLASH 控制器状态寄存器。 1: FLASH 操作完成。 其它: FLASH 控制器忙碌状态或处于 IDLE 状态。 用户可以查看这个寄存器, 确保在禁用 FLASH_PASS 寄存器之前, 所有的 FLASH 操作已完成。	0x0

### 14.5.16 MISC2\_FLASHWC 寄存器

表 14-19 MISC2\_FLASHWC 寄存器

位	名称	类型	描述	默认值
31: 14	-	-	保留	0
13: 8	CYCLE_1US	R/W	这个寄存器用于 FLASH 控制器以 AHBClock 计算 1 $\mu$ s 的滴答。该设置与 FLASH 的唤醒时间以及 FLASH 的擦除时间、FLASH 的编程时间有关。FLASH 唤醒时间=1 $\mu$ s 的滴答时间*10。必须满足 1 $\mu$ stick $\geq$ 1 $\mu$ s。 1 $\mu$ stick=(AHB 时钟周期)*(CYCLE_1US+1)。 比如 AHB 时钟频率为 26.2144M, 为了保证 1 $\mu$ s 的滴答时间, 以及最小的唤醒时间, CYCLE_1US 应该设置为 26。此时 FLASH 的唤醒时间为 27 / 26214400 * 10, 大约为 10 $\mu$ s。 比如 AHB 时钟频率为 32.768k, 为了保证 1 $\mu$ s 的滴答时间, 以及最小的唤醒时间, CYCLE_1US 应该设置为 0。此时 FLASH 的唤醒时间是 1 / 32768 * 10, 大约为 305 $\mu$ s。	0x21
7: 0			保留	0

### 14.5.17 FLASH\_RDPROT 寄存器

表 14-20 FLASH\_RDPROT 寄存器

位	名称	类型	描述	默认值
31: 0	RDPORT	R	此寄存器用于指示特定区域在 ICE 模式 (MODE 为 0) 期间是否已被读保护。对于该寄存器中的每个位, 它指示 FLASH 中 16Kbyte 区域的保护状态。位 0 是第一个 16Kbyte 区域 (0~0x3FFF), 位 31 是最后一个 16Kbyte 区域 (0x7C000~0x7FFFF)。当 RDPORT 寄存器的对应位设置为 1 时, 在 ICE 模式下将忽略对此区域的读取操作。 0: 没有读保护。 1: 读保护, 禁用读操作。 通电后或每次复位后, FLASH 控制器会读取 0x7FC00 值并取反, 然后存储在此寄存器中。FLASH 的 0x7FC00 地址被读取后, 该寄存器也将被更新。例如, 当 0x7FC00 中的数据为 0xFFFFFFFF 时, 该寄存器的值为 0x00000000, 表示没有读	0x00000000

		保护功能。如果 RDPORT 的任何位置为 1，则禁止对相应扇区执行扇区擦除操作，只有芯片全擦除可以清除读保护。 当指令区的读取操作受到保护时，意味着不允许进行 ICE 调试。对于所有受保护区域，在 ICE 模式下，读取值将变为 0	
--	--	---	--

### 14.5.18 FLASH\_WRPROT 寄存器

表 14-21 FLASH\_WRPROT 寄存器

位	名称	类型	描述	默认值
31: 0	WRPORT	R/W	此寄存器用于保护特定的 FLASH 区域不受擦除或写操作的影响。对于这个寄存器中的每个位，它将保护 FLASH 中的 16Kbyte 区域。位 0 是第一个 16Kbyte 区域（0~0x3FFF），位 31 是最后一个 16Kbyte 区域（0x7C000~0x7FFFF）。当对应的位设置为 1 时，将忽略对此区域的擦除或编程操作。 0: 没有擦除/写保护。 1: 擦除/写保护，擦除或编程操作无效。	0x00000000

### 14.5.19 FLASH\_ICEPROT 寄存器

表 14-22 FLASH\_ICEPROT 寄存器

位	名称	类型	描述	默认值
31: 0	ICEPORT	R/W	此寄存器用于指示是否设置了防 ICE 模式。通电后或每次复位后，FLASH 控制器将从地址 0x7FC08 读取数据。如果数据的低 4 位是 0xA，则 ICE 调试模式无效。即芯片无法进入调试模式。	0x00000000

## 14.6 FLASH 保护说明

表 14-23 FLASH 保护说明

模式	功能描述	打开方法及说明	解除方法
读保护	在 ICE Mode 下，保护指定代码区不被读取。	向 flash 地址 0x7FC00 写入 0，然后再读取一次后，读保护生效。 32 个 bit 位，每个 bit 位保护 16KByte 区域的 flash。 Bit0 保护 0~0x3FFF； Bit31 保护 0x7C000~0x7FFFF。 MODE 为 0 时，被保护区域的读出值为 0。 MODE 为 1 时，读保护无效。	只有芯片擦除可以解除读保护模式。
写保护	防止指定代码区被篡改。	向寄存器 FLASH_WRPROT 写 1，设置写保护。 32 个 bit 位，每个 bit 位保护 16KByte 区域的 flash。 Bit0 保护 0~0x3FFF； Bit31 保护 0x7C000~0x7FFFF。 设置写保护后，无法对被保护区域进行编程和扇区擦除操作。	向寄存器 FLASH_WRPROT 写 0，解除写保护； 芯片复位后寄存器 FLASH_WRPROT 被复位为 0，也会解除写保护功能。
ICE 保护	保护芯片，无法进入 ICE Mode。	向 flash 地址 0x7FC08 低 4 位写入 0xA，在芯片复位后，进入 ICE 保护模式。该模式下，无法进行调试操作。	对最后一个扇区进行扇区擦除操作，或进行芯片擦除操作，可以解除 ICE 保护。

## 第15章 GPIO 控制器

### 15.1 简介

V85XXP 中最多有 86 个 IO，其中 16 个 IO（GPIOA）由 PMU 控制器控制，可以唤醒浅睡眠与深睡眠。其余 IO（GPIOB~GPIOF）由 GPIO 控制器控制，不可以唤醒浅睡眠与深睡眠；寄存器配置在浅睡眠下状态保持，深睡眠下状态丢失，深睡眠模式唤醒后用户应手动恢复设置。

### 15.2 特点

- 每个 IO 都可以为输入或输出模式；
- 每个 IO 都可以为开漏模式；
- 所有 IO 无上拉、下拉电阻；
- GPIOA 可以唤醒浅睡眠与深睡眠。

### 15.3 功能框图

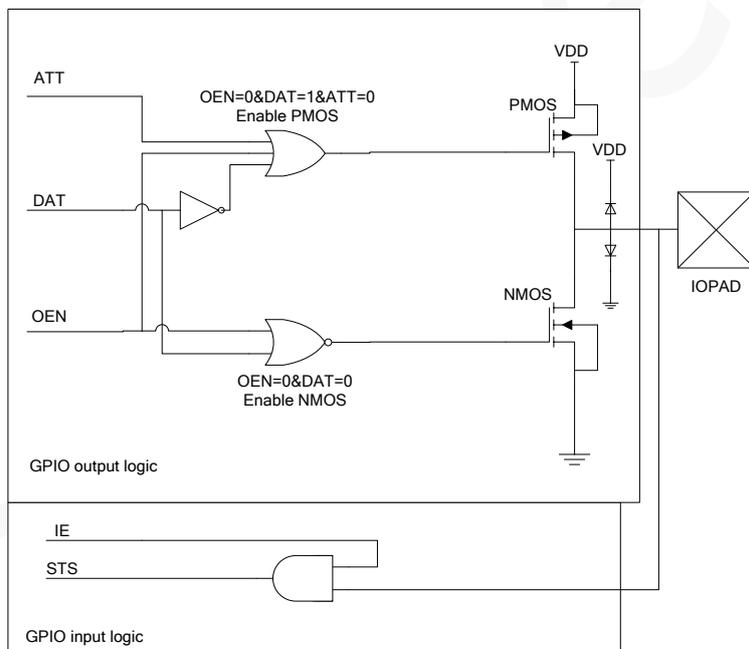


图 15-1 GPIO 控制器的功能框图

### 15.4 寄存器地址

表 15-1 PMU\_IOA 控制器(PMU 基地址: 0x40014000)

名称	类型	地址	描述	默认值
PMU_IOAOEN	R/W	0x0010	IOA 输出使能寄存器	0xFFFF
PMU_IOAIE	R/W	0x0014	IOA 输入使能寄存器	0xFFFF
PMU_IOADAT	R/W	0x0018	IOA 数据寄存器	0x0000
PMU_IOAATT	R/W	0x001C	IOA 属性寄存器	0x0000
PMU_IOAWKUEN	R/W	0x0020	IOA 唤醒使能寄存器	0x00000000
PMU_IOASTS	R	0x0024	IOA 输入状态寄存器	--

PMU_IOAINTSTS	R/C	0x0028	IOA 中断状态寄存器	0x0000
PMU_IOASEL	R/W	0x0038	IOA 特殊功能选择寄存器	0x0000
PMU_IOANODEG	R/W	0x0050	IOA 去抖动电路控制	0x0000

表 15-2 GPIO 控制器(GPIO 基地址: 0x4000000)

名称	类型	地址	描述	默认值
IOB_OEN	R/W	0x0020	IOB 输出使能寄存器	0xFFFF
IOB_IE	R/W	0x0024	IOB 输入使能寄存器	0xFFFF
IOB_DAT	R/W	0x0028	IOB 输出数据寄存器	0x0000
IOB_ATT	R/W	0x002C	IOB 属性寄存器	0x0000
IOB_STS	R	0x0030	IOB 输入状态寄存器	--
IOC_OEN	R/W	0x0040	IOC 输出使能寄存器	0xFFFF
IOC_IE	R/W	0x0044	IOC 输入使能寄存器	0xFFFF
IOC_DAT	R/W	0x0048	IOC 输出数据寄存器	0x0000
IOC_ATT	R/W	0x004C	IOC 属性寄存器	0x0000
IOC_STS	R	0x0050	IOC 输入状态寄存器	--
IOD_OEN	R/W	0x0060	IOD 输出使能寄存器	0xFFFF
IOD_IE	R/W	0x0064	IOD 输入使能寄存器	0xFFFF
IOD_DAT	R/W	0x0068	IOD 输出数据寄存器	0x0000
IOD_ATT	R/W	0x006C	IOD 属性寄存器	0x0000
IOD_STS	R	0x0070	IOD 输入状态寄存器	--
IOE_OEN	R/W	0x0080	IOE 输出使能寄存器	0xFFFF
IOE_IE	R/W	0x0084	IOE 输入使能寄存器	0xFFFF
IOE_DAT	R/W	0x0088	IOE 输出数据寄存器	0x0000
IOE_ATT	R/W	0x008C	IOE 属性寄存器	0x0000
IOE_STS	R	0x0090	IOE 输入状态寄存器	--
IOF_OEN	R/W	0x00A0	IOF 输出使能寄存器	0x7FFF
IOF_IE	R/W	0x00A4	IOF 输入使能寄存器	0x7FFF
IOF_DAT	R/W	0x00A8	IOF 输出数据寄存器	0x0000
IOF_ATT	R/W	0x00AC	IOF 属性寄存器	0x0000
IOF_STS	R	0x00B0	IOF 输入状态寄存器	--
IOB_SEL	R/W	0x00C0	IOB 特殊功能选择寄存器	0x00
IOE_SEL	R/W	0x00CC	IOE 特殊功能选择寄存器	0x00
IO_MISC	R/W	0x00E0	IOmisc 控制寄存器	0x00

## 15.5 寄存器定义

### 15.5.1 PMU\_IOAOEN 寄存器

表 15-3 PMU\_IOAOEN 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	IOAOEN	R/W	每个位控制 IOA 的输出使能信号，有关 IO 状态的详细信息，请参见表 15-7。 0: 使能 IOA 输出模式。 1: 禁止 IOA 输出模式。	0xFFFF

### 15.5.2 PMU\_IOAIE 寄存器

表 15-4 PMU\_IOAIE 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	IOAIE	R/W	每个位控制 IOA 的输入使能信号，有关 IO 状态的详细信息，请参见表 15-7。 0: 禁止 IOA 输入功能。 1: 使能 IOA 输入功能。	0xFFFF

### 15.5.3 PMU\_IOADAT 寄存器

表 15-5 PMU\_IOADAT 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	IOADAT	R/W	每个位控制 IOA 的输出数据和拉低/高功能，有关 IO 状态的详细信息，请参见表 15-7。	0x0000

### 15.5.4 PMU\_IOAATT 寄存器

表 15-6 PMU\_IOAATT 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	IOAATT	R/W	每一位控制 IOA 的属性，下表是不同设置下的 IO 状态的详细信息。当一个 IO 口设置为输出模式（GPIO 或特殊功能），该位用来控制 IO 为 CMOS 或开漏模式。 0: CMOS 模式。 1: 开漏模式(禁止 PMOS 输出)。	0x0000

表 15-7 不同设置下的 IO 状态

IOx 设置			IO 状态
IOxOEN	IOxDAT	IOxATT	
1	0	0	禁止输出
1	1	0	禁止输出

1	0	1	禁止输出
1	1	1	禁止输出
0	0	0	输出寄存器上的'0'激活 N-MOS, 而输出寄存器上的'1'将激活 P-MOS。
0	1	0	
0	0	1	输出寄存器上的'0'激活 N-MOS, 而输出寄存器上的'1'将端口置于高阻状态(P-MOS 从不被激活)。
0	1	1	

### 15.5.5 PMU\_IOAWKUEN 寄存器

表 15-8 PMU\_IOAWKUEN 寄存器

位	名称	类型	描述	默认值
31: 0	IOAWKUEN	R/W	每 2 位控制 IOA 的中断和唤醒功能。 位[1: 0]: IOA0WKUEN[1: 0] 位[3: 2]: IOA1WKUEN[1: 0] ..... 位[31: 30]: IOA15WKUEN[1: 0] 有关每种唤醒模式的详细信息, 请参见表 15-9。	0x00000000

表 15-9 IO 唤醒模式

IOAyWKUEN[1: 0]	IOAyDAT	唤醒事件
0	X	禁用唤醒功能
1	0	上升沿唤醒
	1	下降沿唤醒
2	0	高电平唤醒
	1	低电平唤醒
3	X	双边沿唤醒

其中  $y=0...15$ , 表示 IOA0~15。

硬件在进入浅睡眠或深睡眠模式前锁存 IOA 最后时刻的状态, 当选择上升沿唤醒时, 用户一定要保证在进入睡眠前该 IOA 为低电平, 确保正确检测到 IOA 的上升沿, 否则不能正常唤醒。当选择下降沿唤醒模式时, 用户应保证进入睡眠前该 IOA 状态为高电平, 确保正确检测到 IOA 的下降沿, 否则不能正常唤醒。

电平中断没有去毛刺电路, 包括正常中断和浅睡眠、深睡眠的 IOA 电平唤醒。去毛刺电路仅对浅睡眠、深睡眠模式的 IOA 边沿唤醒信号有效, 去毛刺时间约为 4 个 RTCCLK 时钟周期。

### 15.5.6 PMU\_IOASTS 寄存器

表 15-10 PMU\_IOASTS 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	IOASTS	R	每个位表示当前 IO 的输入数据值。	--

### 15.5.7 PMU\_IOAINTSTS 寄存器

表 15-11 PMU\_IOAINTSTS 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	INTSTS	R/C	IO 中断状态寄存器中的每一位代表 IOA 的中断状态。当检测到相应 IOA 唤醒事件时，对应的中断位会置 1。写 1 清零对应的标志位。	0x0000

### 15.5.8 PMU\_IOASEL 寄存器

表 15-12 PMU\_IOASEL 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7	IOA_SEL7	R/W	IOA7 特殊功能选择寄存器。 0: GPIO。 1: 特殊功能 1 (RTC_PLLDIV)。	0x0
6	IOA_SEL6	R/W	IOA6 特殊功能选择寄存器。 0: GPIO。 1: 特殊功能 1 (CMP2_O)。	0x0
5: 4	-	-	保留	0
3	IOA_SEL3	R/W	IOA3 特殊功能选择寄存器。 0: GPIO。 1: 特殊功能 1 (RTC_PLLDIV)。	0x0
2: 0	-	-	保留	0

### 15.5.9 PMU\_IOANODEG 寄存器

表 15-13 PMU\_IOANODEGL 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	
15: 0	IOANODEG	R/W	每一位控制 IOA 边沿唤醒信号是否有去抖动电路。该位设置为 1，相应 IOA 引脚的去抖动电路就失效，唤醒速度变快。去抖动电路影响浅睡眠、深睡眠模式下的边沿唤醒信号。 0: IOA 唤醒信号有去抖动电路。 1: IOA 唤醒信号无去抖动电路。	0x0000

### 15.5.10 IOX\_OEN 寄存器

表 15-14 IOX\_OEN 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0

15: 0	IOXOEN	R/W	每个位控制 IOX 的输出使能信号，有关 IO 状态的详细信息，请参见表 15-7。 0: 使能 IO 输出模式。 1: 禁止 IO 输出模式。	0xFFFF
-------	--------	-----	--	--------

### 15.5.11 IOX\_IE 寄存器

表 15-15 IOX\_IE 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	IOXIE	R/W	每个位控制 IOX 的输入使能信号，有关 IO 状态的详细信息，请参见表 15-7。 0: 禁止 IO 输入功能。 1: 使能 IO 输入功能。	0xFFFF

### 15.5.12 IOX\_DAT 寄存器

表 15-16 IOX\_DAT 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	IOXDAT	R/W	每个位控制 IOX 的输出数据和拉低/高功能，有关 IO 状态的详细信息，请参见表 15-7。	0x0000

### 15.5.13 IOX\_ATT 寄存器

表 15-17 IOX\_ATT 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	IOXATT	R/W	该寄存器的每一位用来控制相应 IO 的属性。 0: CMOS 输出 1: 开漏输出 当某一个 IO 被配置为一个特殊功能，比如 UART 或 SPI，用户可以设置相应的 IOXATT 为 1，让它处于开漏模式，通过外部上拉输出高。	0x0000

### 15.5.14 IOX\_STS 寄存器

表 15-18 IOX\_STS 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	IOXSTS	R	每个位显示当前 IOX 输入数据值。必须使能输入，才能读到有效的外部电平状态。	--

### 15.5.15 IOB\_SEL 寄存器

表 15-19 IOB\_SEL 寄存器

位	名称	类型	描述	默认值
31: 7	-	-	保留	0
6	IOB_SEL6	R/W	IOB6 特殊功能选择寄存器。 0: GPIO 或特殊功能 1/2。 1: 特殊功能 1/2 未使能时的特殊功能 3 (RTCCLK 输出)。	0x0
5: 3	-	-	保留	0
2	IOB_SEL2	R/W	IOB2 特殊功能选择寄存器。 0: GPIO 或特殊功能 1/2。 1: 特殊功能 1/2 未使能时的特殊功能 3 (PLL 输出)。	0x0
1	IOB_SEL1	R/W	IOB1 特殊功能选择寄存器。 0: GPIO 或特殊功能 1/2。 1: 特殊功能 1/2 未使能时的特殊功能 3 (PLLH 分频器输出)。	0x0
0	-	-	保留	0

### 15.5.16 IOE\_SEL 寄存器

表 15-20 IOE\_SEL 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7	IOE_SEL7	R/W	IOE7 特殊功能选择寄存器。 0: GPIO 或高阻。 1: 特殊功能 1 (CMP1_O)。	0x0
6: 0	-	-	保留	0

### 15.5.17 IO\_MISC 寄存器

表 15-21 IO\_MISC 寄存器

位	名称	类型	描述	默认值
31: 6	-	-	保留	0
5	I2CIOC	R/W	该寄存器用于控制 IOB 或 IOC 的 I <sup>2</sup> C 功能。 0: I <sup>2</sup> C 在 IOB13~IOB14。 1: I <sup>2</sup> C 在 IOC4~IOC5。	0x0
4: 3	-	-	保留	0
2: 0	PLLHDIV	R/W	当 IOB1 被选择为特殊功能 3 时, 该寄存器用于控制 PLLH 输出的分频比。 0: /1 1: /2	0x0

			2: /4 3: /8 4: /16 其他: 保留	
--	--	--	------------------------------------	--

## 15.6 特殊功能 IO

### 15.6.1 特殊功能 IOA

下表显示的特殊功能 IOA 的特殊功能将开启时，对应的模块功能。

表 15-22 特殊功能 IOA

IO	特殊功能 1	特殊功能 2	特殊功能 3	特殊功能 4
IOA0	SWDCLK (MODE = 0)	-	LCDSEG54	-
IOA1	SWDIO (MODE = 0)	-	LCDSEG53	-
IOA2	--	-	LCDSEG52	-
IOA3	RTC_PLLDIV	-	LCDSEG51	-
IOA4	-	-	LCDSEG66	CMP2_P
IOA5	-	-	LCDSEG67	CMP2_N
IOA6	CMP2_O	-	LCDSEG68	-
IOA7	RTC_PLLDIV	-	LCDSEG69	-
IOA8	-	-	LCDSEG50	ADC_CH3
IOA9	-	-	LCDSEG49	ADC_CH4
IOA10	-	-	LCDSEG48	ADC_CH5
IOA11	-	-	LCDSEG47	ADC_CH6
IOA12	UART0_RX	-	LCDSEG71	-
IOA13	UART1_RX	ISO7816_IO0	LCDSEG73	-
IOA14	UART2_RX	-	LCDSEG13	-
IOA15	UART3_RX	ISO7816_IO1	LCDSEG15	-

### 15.6.2 特殊功能 IOB

下表显示了 IOB 的特殊功能，特殊功能会在相应模块被使能后打开。但是当特殊功能是 LCD COM/SEG 或 ADC 或 CMP 或 X6\_5M 或 VDCIN 时，用户应该手动禁止相应 GPIO 的输入输出功能，并禁用此 GPIO 上的其他特殊功能，以确保这些特殊功能的正确性。

表 15-23 特殊功能 IOB

IO	特殊功能 1	特殊功能 2	特殊功能 3	特殊功能 4
IOB0	UART4_RX	PWM0	-	LCDSEG43
IOB1	UART5_RX	-	PLLH divider	LCDSEG45

IOB2	UART0_TX	-	PLLL	LCDSEG70
IOB3	UART1_TX	ISO7816_CLK0	-	LCDSEG72
IOB4	UART2_TX	-	-	LCDSEG12
IOB5	UART3_TX	ISO7816_CLK1	-	LCDSEG14
IOB6	UART4_TX	PWM1	RTCCLK	LCDSEG44
IOB7	UART5_TX	-	-	LCDSEG46
IOB8	-	-	-	LCDSEG35
IOB9	SPI1CSN	-	-	LCDSEG36
IOB10	SPI1CLK	-	-	LCDSEG37
IOB11	SPI1MISO	-	-	LCDSEG38
IOB12	SPI1MOSI	-	-	LCDSEG39
IOB13	I2CSCL (I2CIOC=0)	PWM2	-	LCDSEG40
IOB14	I2CSDA (I2CIOC=0)	PWM3	-	LCDSEG41
IOB15	TIMER EXT CLK	-	-	LCDSEG42

### 15.6.3 特殊功能 IOC

下表显示了 IOC 的特殊功能，特殊功能会在相应模块被使能后打开。但是当特殊功能是 LCD COM/SEG 或 ADC 或 CMP 或 X6\_5M 或 VDCIN 时，用户应该手动禁止相应 GPIO 的输入输出功能，并禁用此 GPIO 上的其他特殊功能，以确保这些特殊功能的正确性。

表 15-24 特殊功能 IOC

IO	特殊功能 1	特殊功能 2	特殊功能 3
IOC0	SPI2CSN	-	LCDSEG16
IOC1	SPI2CLK	-	LCDSEG17
IOC2	SPI2MISO	-	LCDSEG18
IOC3	SPI2MOSI	--	LCDSEG19
IOC4	-	I2CSCL(I2CIOC=1)	LCDSEG20
IOC5	-	I2CSDA(I2CIOC=1)	LCDSEG21
IOC6	-	-	LCDSEG22
IOC7	-	-	LCDSEG23
IOC8	-	-	LCDSEG24
IOC9	-	-	LCDSEG25
IOC10	-	-	LCDSEG26
IOC11	-	-	LCDSEG27
IOC12	-	-	LCDSEG28
IOC13	-	-	LCDSEG29
IOC14	-	-	LCDSEG30
IOC15	-	-	LCDSEG31

## 15.6.4 特殊功能 IOD

下表显示了 IOD 的特殊功能，特殊功能会在相应模块被使能后打开。但是当特殊功能是 LCD COM/SEG 或 ADC 或 CMP 或 X6\_5M 或 VDCIN 时，用户应该手动禁止相应 GPIO 的输入输出功能，并禁用此 GPIO 上的其他特殊功能，以确保这些特殊功能的正确性。

表 15-25 特殊功能 IOD

IO	特殊功能 1	特殊功能 2	特殊功能 3
IOD0	LCDCOM0	-	-
IOD1	LCDCOM1	-	-
IOD2	LCDCOM2	-	-
IOD3	LCDCOM3	-	-
IOD4	LCDCOM4	LCDSEG0	SPI3CSN
IOD5	LCDCOM5	LCDSEG1	SPI3MISO
IOD6	LCDCOM6	LCDSEG2	SPI3MOSI
IOD7	LCDCOM7	LCDSEG3	SPI3CLK
IOD8	LCDSEG4	-	-
IOD9	LCDSEG5	-	-
IOD10	LCDSEG6	-	-
IOD11	LCDSEG7	-	-
IOD12	LCDSEG8	-	-
IOD13	LCDSEG9	-	-
IOD14	LCDSEG10	-	-
IOD15	LCDSEG11	-	-

## 15.6.5 特殊功能 IOE

下表显示了 IOE 的特殊功能，特殊功能会在相应模块被使能后打开。但是当特殊功能是 LCD COM/SEG 或 ADC 或 CMP 或 X6\_5M 或 VDCIN 时，用户应该手动禁止相应 GPIO 的输入输出功能，并禁用此 GPIO 上的其他特殊功能，以确保这些特殊功能的正确性。

表 15-26 特殊功能 IOE

IO	特殊功能 1	特殊功能 2	特殊功能 3
IOE0	-	LCDSEG74	-
IOE1	-	LCDSEG75	-
IOE2	-	LCDSEG76	-
IOE3	-	LCDSEG77	-
IOE4	-	LCDSEG63	ADC_CH7
IOE5	-	LCDSEG62	ADC_CH8
IOE6	-	LCDSEG61	ADC_CH9, Tiny ADC 0
IOE7	CMP1_0	LCDSEG60	ADC_CH11, Tiny ADC 1
IOE8	-	LCDSEG59	CMP1_P
IOE9	-	LCDSEG58	CMP1_N

IOE10	-	LCDSEG32	-
IOE11	-	LCDSEG33	-
IOE12	-	LCDSEG34	-
IOE13	-	LCDSEG55	-
IOE14	-	LCDSEG56	-
IOE15	-	LCDSEG57	-

### 15.6.6 特殊功能 IOF

下表显示了 IOF 的特殊功能，特殊功能会在相应模块被使能后打开。但是当特殊功能是 LCD COM/SEG 或 ADC 或 CMP 或 X6\_5M 或 VDCIN 时，用户应该手动禁止相应 GPIO 的输入输出功能，并禁用此 GPIO 上的其他特殊功能，以确保这些特殊功能的正确性。

表 15-27 特殊功能 IOF

IO	特殊功能 1
IOF0	X6_5MI
IOF1	X6_5MO
IOF7	无
IOF8	无
IOF11	无
IOF12	无

## 第16章 DMA 控制器

### 16.1 简介

DMA 模块提供内存到内存、内存到外设、外设到内存之间的高速数据传输。DMA 控制器集成了 AHB 到 APB 桥接器，因此当 DMA 进行低速外设数据传输时，高速外设总线（AHB）性能不会受到低速外设的影响。在 V85XXP 中集成了 4 个 DMA 通道，4 个通道的优先级是循环的。当系统从浅睡眠或深睡眠唤醒时，DMA 控制器的设置会被复位，所以从这两种状态唤醒后应该手动恢复相关设置。

### 16.2 特点

- 支持内存间的传输；
- 支持内存到外设传输；
- 支持外设到内存传输；
- 软件模式或硬件 DMA 请求模式；
- 支持固定地址、单个包传输完成后地址循环或所有包传输完成后地址循环模式；
- 支持单个包传输完成/所有包传输完成/数据传输中止 IRQ 产生；
- 支持总线数据中止检测；
- 集成 AHB 到 APB 桥接；
- 支持 AES128/192/256 编解码。

### 16.3 功能框图

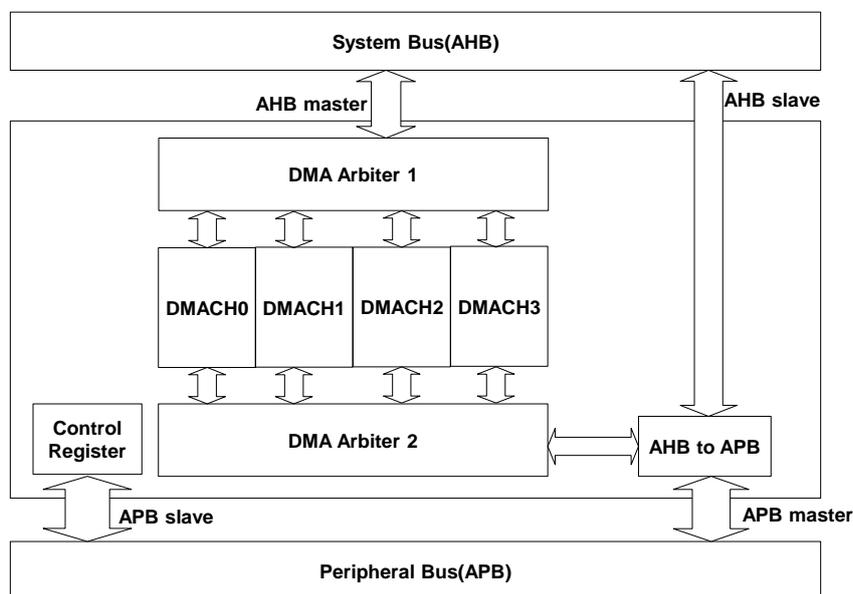


图 16-1 DMA 控制器功能框图

### 16.4 寄存器地址

表 16-1 DMA 控制器(DMA 基地址: 0x40010000)

名称	类型	地址	描述	默认值
DMA_IE	R/W	0x0000	DMA 中断使能寄存器	0x000

DMA_STS	R/W	0x0004	DMA 状态寄存器	0x0000
DMA_C0CTL	R/W	0x0010	DMA 通道 0 控制寄存器	0x00000000
DMA_C0SRC	R/W	0x0014	DMA 通道 0 源地址寄存器	0x00000000
DMA_C0DST	R/W	0x0018	DMA 通道 0 目的地址寄存器	0x00000000
DMA_C0LEN	R	0x001C	DMA 通道 0 传输长度寄存器	0x0000
DMA_C1CTL	R/W	0x0020	DMA 通道 1 控制寄存器	0x00000000
DMA_C1SRC	R/W	0x0024	DMA 通道 1 源地址寄存器	0x00000000
DMA_C1DST	R/W	0x0028	DMA 通道 1 目的地址寄存器	0x00000000
DMA_C1LEN	R	0x002C	DMA 通道 1 传输长度寄存器	0x0000
DMA_C2CTL	R/W	0x0030	DMA 通道 2 控制寄存器	0x00000000
DMA_C2SRC	R/W	0x0034	DMA 通道 2 源地址寄存器	0x00000000
DMA_C2DST	R/W	0x0038	DMA 通道 2 目的地址寄存器	0x00000000
DMA_C2LEN	R	0x003C	DMA 通道 2 传输长度寄存器。	0x0000
DMA_C3CTL	R/W	0x0040	DMA 通道 3 控制寄存器	0x00000000
DMA_C3SRC	R/W	0x0044	DMA 通道 3 源地址寄存器	0x00000000
DMA_C3DST	R/W	0x0048	DMA 通道 3 目的地址寄存器	0x00000000
DMA_C3LEN	R	0x004C	DMA 通道 3 传输长度寄存器	0x0000
DMA_AESCTL	R/W	0x0050	DMA AES 控制寄存器	0x00000000
DMA_AESKEY0	R/W	0x0060	DMA AES 密钥 0 寄存器	0x00000000
DMA_AESKEY1	R/W	0x0064	DMA AES 密钥 1 寄存器	0x00000000
DMA_AESKEY2	R/W	0x0068	DMA AES 密钥 2 寄存器	0x00000000
DMA_AESKEY3	R/W	0x006C	DMA AES 密钥 3 寄存器	0x00000000
DMA_AESKEY4	R/W	0x0070	DMA AES 密钥 4 寄存器	0x00000000
DMA_AESKEY5	R/W	0x0074	DMA AES 密钥 5 寄存器	0x00000000
DMA_AESKEY6	R/W	0x0078	DMA AES 密钥 6 寄存器	0x00000000
DMA_AESKEY7	R/W	0x007C	DMA AES 密钥 7 寄存器	0x00000000

## 16.5 寄存器定义

### 16.5.1 DMA\_IE 寄存器

表 16-2 DMA\_IE 寄存器

位	名称	类型	描述	默认值
31: 12			保留	0
11	C3DAIE	R/W	通道 3 数据中止中断使能。 0: 禁用 1: 启用	0x0

10	C2DAIE	R/W	通道 2 数据中止中断使能。 0: 禁用 1: 启用	0x0
9	C1DAIE	R/W	通道 1 数据中止中断使能。 0: 禁用 1: 启用	0x0
8	C0DAIE	R/W	通道 0 数据中止中断使能。 0: 禁用 1: 启用	0x0
7	C3FEIE	R/W	通道 3 所有包传输完成中断使能。 0: 禁用 1: 启用	0x0
6	C2FEIE	R/W	通道 2 所有包传输完成中断使能。 0: 禁用 1: 启用	0x0
5	C1FEIE	R/W	通道 1 所有包传输完成中断使能。 0: 禁用 1: 启用	0x0
4	C0FEIE	R/W	通道 0 所有包传输完成中断使能。 0: 禁用 1: 启用	0x0
3	C3PEIE	R/W	通道 3 单个包传输完成中断使能。 0: 禁用 1: 启用	0x0
2	C2PEIE	R/W	通道 2 单个包传输完成中断使能。 0: 禁用 1: 启用	0x0
1	C1PEIE	R/W	通道 1 单个包传输完成中断使能。 0: 禁用 1: 启用	0x0
0	C0PEIE	R/W	通道 0 单个包传输完成中断使能。 0: 禁用 1: 启用	0x0

## 16.5.2 DMA\_STS 寄存器

表 16-3 DMA\_STS 寄存器

位	名称	类型	描述	默认值
31: 16			保留	0
15	C3DA	R/C	通道 3 数据中止中断标志，写 1 清除该标志。	0x0

14	C2DA	R/C	通道 2 数据中止中断标志，写 1 清除该标志。	0x0
13	C1DA	R/C	通道 1 数据中止中断标志，写 1 清除该标志。	0x0
12	C0DA	R/C	通道 0 数据中止中断标志，写 1 清除该标志。	0x0
11	C3FE	R/C	通道 3 所有包传输完成中断标志，写 1 清除该标志。	0x0
10	C2FE	R/C	通道 2 所有包传输完成中断标志，写 1 清除该标志。	0x0
9	C1FE	R/C	通道 1 所有包传输完成中断标志，写 1 清除该标志。	0x0
8	C0FE	R/C	通道 0 所有包传输完成中断标志，写 1 清除该标志。	0x0
7	C3PE	R/C	通道 3 单个包传输完成中断标志，写 1 清除该标志。	0x0
6	C2PE	R/C	通道 2 单个包传输完成中断标志，写 1 清除该标志。	0x0
5	C1PE	R/C	通道 1 单个包传输完成中断标志，写 1 清除该标志。	0x0
4	C0PE	R/C	通道 0 单个包传输完成中断标志，写 1 清除该标志。	0x0
3	C3BUSY	R	DMA 通道 3 工作状态寄存器。 0: 空闲 1: 工作	0x0
2	C2BUSY	R	DMA 通道 2 工作状态寄存器。 0: 空闲 1: 工作	0x0
1	C1BUSY	R	DMA 通道 1 工作状态寄存器。 0: 空闲 1: 工作	0x0
0	C0BUSY	R	DMA 通道 0 工作状态寄存器。 0: 空闲 1: 工作	0x0

### 16.5.3 DMA\_CxCTL 寄存器

表 16-4 DMA\_CxCTL 寄存器用于控制 DMA，该寄存器中的数据将在传输开始后锁存到 DMA 通道，因此对该寄存器的任何修改都不会影响正在进行的 DMA 传输。只有 STOP 位可以立即停止 DMA 传输。

表 16-4 DMA\_CxCTL 寄存器

位	名称	类型	描述	默认值
31: 24	FLEN	R/W	包个数寄存器，DMA 传输数据帧中所有包的包个数为 (FLEN+1)。DMA 传输的总长度为 (FLEN+1) * (PLEN+1)。	0x0
23: 16	PLEN	R/W	包长度寄存器，单个包的数据个数为 (PLEN+1)。DMA 传输的总长度为 (FLEN+1) * (PLEN+1)。	0x0
15	STOP	R/W	强制结束 DMA 传输，向此位写入 1 将强制 DMA 通道返回空闲状态，如果另一个 DMA 传输需要分配到同一通道，用户应将此位写零。 0: 默认工作状态 1: 强制结束 DMA 传输	0x0
14	AESEN	R/W	使能 DMA 通道的 AES 加密/解密功能。只有 DMA 通道 3 支持 AES 功能。当启用 AES 功能时，SIZE 应设置为 32 位模式，SMODE 和 DMODE 应设置为 1 或 2。使能 AES 功能时，PLEN	0x0

			应配置为 4 的整数倍。 0: 禁止 1: 使能	
13	CONT	R/W	连续模式, DMA 传输会持续直到 STOP 位设置为 1 才停止。每次 DMA 完成所有包和帧传输后,它将从头开始执行并连续传输。 0: 非连续模式 1: 连续模式	0x0
12	TMODE	R/W	传送模式选择寄存器。 0: DMA 请求源每触发一次只传输一个数据。 1: DMA 请求源每触发一次传输一个包数据。	0x0
11: 7	DMASEL	R/W	DMA 请求源选择。有关每个 DMA 请求源的详细信息,请参见表 16-5	0x0
6: 5	DMODE	R/W	目的地址模式。 0: 固定 1: 自动增加但在单个包传输完成后返回起始值 2: 自动增加但在所有包传输完成后返回起始值 3: 保留	0x0
4: 3	SMODE	R/W	源地址模式 0: 固定 1: 自动增加但在单个包传输完成后返回起始值 2: 自动增加但在所有包传输完成后返回起始值 3: 保留	0x0
2: 1	SIZE	R/W	传输长度模式 0: 字节(8 位) 1: 半字(16 位) 2: 字(32 位) 3: 保留	0x0
0	EN	R/W	DMA 通道使能寄存器 0: 禁止 DMA 通道(当 CONT 为 1 时无效) 1: 使能 DMA 通道 当在 CONT 为 0 时, DMA 传输完毕后, 该位将由硬件自动清零	0x0

表 16-5 DMA 请求源选择

DMASEL	请求源	DMASEL	请求源	DMASEL	请求源	DMASEL	Source
0	Software	8	UART3 TX	16	ISO78161 TX	24	UART 32K 0
1	ADC	9	UART3 RX	17	ISO78161 RX	25	UART 32K 1
2	UART0 TX	10	UART4 TX	18	TIMER 0	26	CMP1
3	UART0 RX	11	UART4 RX	19	TIMER 1	27	CMP2

4	UART1 TX	12	UART5 TX	20	TIMER 2	28	SPI3 TX
5	UART1 RX	13	UART5 RX	21	TIMER 3	29	SPI3 RX
6	UART2 TX	14	ISO78160 TX	22	SPI1 TX	30	SPI2 TX
7	UART2 RX	15	ISO78160 RX	23	SPI1 RX	31	SPI2 RX

### 16.5.4 DMA\_CxSRC 寄存器

表 16-6 DMA\_CxSRC 寄存器

位	名称	类型	描述	默认值
31: 0	SRC	R/W	DMA 源地址寄存器。当 DMA_CxCTL 中的 SIZE 为 1 时，该寄存器必须为半字对齐。当 DMA_CxCTL 中的 SIZE 为 2 时，该寄存器必须为字对齐。当 SRC 设置为 0x4001xxxx 时，它将自动切换到 IO 读取。	0x00000000

### 16.5.5 DMA\_CxDST 寄存器

表 16-7 DMA\_CxDST 寄存器

位	名称	类型	描述	默认值
31: 0	DST	R/W	DMA 目的地址寄存器。当 DMA_CxCTL 中的 SIZE 为 1 时，该寄存器必须为半字对齐。当 DMA_CxCTL 中的 SIZE 为 2 时，该寄存器必须为字对齐。当 DST 设置为 0x4001xxxx 时，它将自动切换到 IO 写入。	0x00000000

### 16.5.6 DMA\_CxLEN 寄存器

表 16-8 DMA\_CxLEN 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 8	CFLEN	R	当前已传输的包个数。 DMA 已传输的总长度为 $CFLEN * (PLEN + 1) + CPLEN$ 。 在 DMA 完成所有包传输后自动清零。	0x00
7: 0	CPLEN	R	当前单个传输包中已传输的数据个数。 DMA 已传输的总长度为 $CFLEN * (PLEN + 1) + CPLEN$ 。 在 DMA 完成单个包传输后自动清零。	0x00

### 16.5.7 DMA\_AESCTL 寄存器

表 16-9 DMA\_AESCTL 寄存器

位	名称	类型	描述	默认值
31: 4	-	-	保留	0
3: 2	MODE	R/W	AES 模式选择寄存器。 0: AES128 1: AES192	0x0

			2: AES256 3: 保留	
1	-	-	保留	0
0	ENC	R/W	AES 编码/解码选择寄存器。 0: 解码 1: 编码	0x0

### 16.5.8 DMA\_AESKEYx 寄存器

表 16-10 DMA\_AESKEYx 寄存器

位	名称	类型	描述	默认值
31: 0	KEYx	R/W	AES 密钥寄存器: KEY0: 位 31~0 KEY1: 位 63~32 KEY2: 位 95~64 KEY3: 位 127~96 KEY4: 位 159~128 KEY5: 位 191~160 KEY6: 位 223~192 KEY7: 位 255~224 当模式为 AES128 时, 仅使用位 127~0。 当模式为 AES192 时, 仅使用位 191~0。 当模式为 AES256 时, 使用位 255~0。	0x00000000

X=0~7

## 第17章 UART 控制器

### 17.1 简介

UART 控制器用于通过 UART 协议发送/接收数据。V85XXP 最多有 6 个 UART 控制器，每个 UART 控制器可以单独编程，并有自己的 CPU 中断。UART 控制器可支持无奇偶校验的 8 位数据或有奇偶校验的 9 位数据传输。发送机和接收机都有一个数据缓存器和一个移位寄存器。UART 控制器中的设置将在从浅睡眠或深睡眠模式唤醒后复位，用户应在从这两种状态唤醒后手动恢复设置。UART 的波特率支持范围 300~819200bps。

### 17.2 特点

- 可编程波特率发生器；
- 数据长度为 8 位；
- 带奇偶校验或不带奇偶校验；
- 奇偶校验错误监测；
- 发送/接收中断标志；
- 发送/接收溢出中断标志。

### 17.3 功能框图

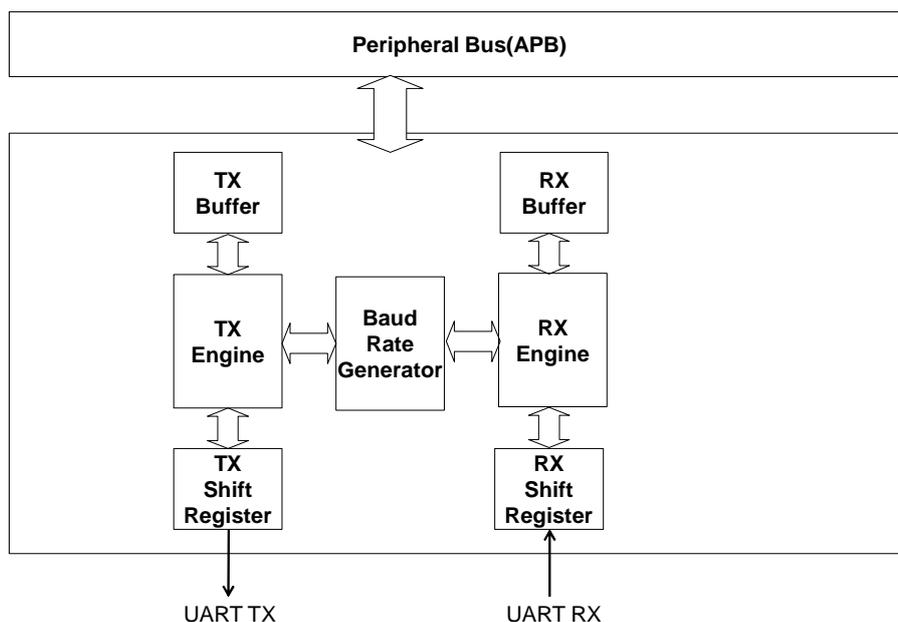


图 17-1 UART 控制器的功能框图

### 17.4 寄存器地址

表 17-1 UART 控制器(UART 基地址: 0x40011800)

名称	类型	地址	描述	默认值
UART0_DATA	R/W	0x0000	UART0 数据寄存器	0x00

UART0_STATE	R/C	0x0004	UART0 状态寄存器	0x00
UART0_CTRL	R/W	0x0008	UART0 控制寄存器	0x000
UART0_INTSTS	R/C	0x000C	UART0 中断状态寄存器	0x00
UART0_BAUDDIV	R/W	0x0010	UART0 波特率分频寄存器	0x00000
UART0_CTRL2	R/W	0x0014	UART0 控制寄存器 2	0x0
UART1_DATA	R/W	0x0020	UART1 数据寄存器	0x00
UART1_STATE	R/C	0x0024	UART1 状态寄存器	0x00
UART1_CTRL	R/W	0x0028	UART1 控制寄存器	0x000
UART1_INTSTS	R/C	0x002C	UART1 中断状态寄存器	0x00
UART1_BAUDDIV	R/W	0x0030	UART1 波特率分频寄存器	0x00000
UART1_CTRL2	R/W	0x0034	UART1 控制寄存器 2	0x0
UART2_DATA	R/W	0x0040	UART2 数据寄存器	0x00
UART2_STATE	R/C	0x0044	UART2 状态寄存器	0x00
UART2_CTRL	R/W	0x0048	UART2 控制寄存器	0x000
UART2_INTSTS	R/C	0x004C	UART2 中断状态寄存器	0x00
UART2_BAUDDIV	R/W	0x0050	UART2 波特率分频寄存器	0x00000
UART2_CTRL2	R/W	0x0054	UART2 控制寄存器 2	0x0
UART3_DATA	R/W	0x0060	UART3 数据寄存器	0x00
UART3_STATE	R/C	0x0064	UART3 状态寄存器	0x00
UART3_CTRL	R/W	0x0068	UART3 控制寄存器	0x000
UART3_INTSTS	R/C	0x006C	UART3 中断状态寄存器	0x00
UART3_BAUDDIV	R/W	0x0070	UART3 波特率分频寄存器	0x00000
UART3_CTRL2	R/W	0x0074	UART3 控制寄存器 2	0x0
UART4_DATA	R/W	0x0080	UART4 数据寄存器	0x00
UART4_STATE	R/C	0x0084	UART4 状态寄存器	0x00
UART4_CTRL	R/W	0x0088	UART4 控制寄存器	0x000
UART4_INTSTS	R/C	0x008C	UART4 中断状态寄存器	0x00
UART4_BAUDDIV	R/W	0x0090	UART4 波特率分频寄存器	0x00000
UART4_CTRL2	R/W	0x0094	UART4 控制寄存器 2	0x0
UART5_DATA	R/W	0x00A0	UART5 数据寄存器	0x00
UART5_STATE	R/C	0x00A4	UART5 状态寄存器	0x00
UART5_CTRL	R/W	0x00A8	UART5 控制寄存器	0x000
UART5_INTSTS	R/C	0x00AC	UART5 中断状态寄存器	0x00
UART5_BAUDDIV	R/W	0x00B0	UART5 波特率分频寄存器	0x00000
UART5_CTRL2	R/W	0x00B4	UART5 控制寄存器 2	0x0

表 17-2 MISC1 控制器(MISC1 基地址: 0x40013000)

名称	类型	地址	描述	默认值
MISC1_IREN	R/W	0x000C	IR 使能控制寄存器	0x00

MISC1_DUTYL	R/W	0x0010	IR 调制波低电平宽度控制寄存器	0x0000
MISC1_DUTYH	R/W	0x0014	IR 调制波高电平宽度控制寄存器	0x0000

## 17.5 寄存器定义

### 17.5.1 UARTx\_DATA 寄存器

表 17-3 UARTx\_DATA 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7: 0	DATA	R/W	读: 接收数据 写: 发送数据	0x0

### 17.5.2 UARTx\_STATE 寄存器

表 17-4 UARTx\_STATE 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7	DMATXDONE	R/C	发送完成标志。当通过 DMA 方式批量数据发送时, 会在全部发送完成时自动置 1。用户必须等到 DMA 发送完成后才能读取此标志。写 1 清零该位。	0x0
6	RXPSTS	R	接收奇偶校验标志。该位表示最后接收到的数据的奇偶校验位。	0x0
5	TXDONE	R/C	发送完成标志。当单字节数据发送完成时, 该标志将被置 1。向该位写 1 或向 UARTx_INTSTS 寄存器中的 TXDONEIF 位写 1, 该位将被清零。	0x0
4	RXPE	R/C	接收数据奇偶校验错误标志位。当接收数据的奇偶校验值与期望不同时, 该标志会被置 1。向该位写 1 或向 UARTx_INTSTS 寄存器中的 RXPEIF 位写 1, 该位将被清零。	0x0
3	RXOV	R/C	接收缓存区溢出标志。当 RXFULL 为 1 并且又从 RX 接收机接收到一个数据时, 该标志将被置 1。向该位写 1 或向 UARTx_INTSTS 寄存器中的 RXOVIF 位写 1, 该位将被清零。	0x0
2	TXOV	R/C	发送缓存区溢出标志。当 UART 发送没有完成并且有一个新数据写入到 UARTx_DATA 寄存器时, 该标志将被置 1。向该位写 1 或向 UARTx_INTSTS 寄存器中的 TXOVIF 位写 1, 该位将被清零。	0x0
1	RXFULL	R	接收缓存器满标志 0: 接收缓存器为空 1: 接收缓存器已满 当 UARTRX 接收机接收到数据时, 该标志位将被置 1。用户从 UARTx_DATA 读取数据后, 该位将被清零。	0x0
0	-	-	保留	0

### 17.5.3 UARTx\_CTRL 寄存器

表 17-5 UARTx\_CTRL 寄存器

位	名称	类型	描述	默认值
31: 9	-	-	保留	0
8	TXDONEIE	R/W	发送完成中断使能	0x0
7	RXPEIE	R/W	接收数据奇偶校验错误中断使能	0x0
6	-	-	保留	0
5	RXOVIE	R/W	接收溢出中断使能	0x0
4	TXOVIE	R/W	发送溢出中断使能	0x0
3	RXIE	R/W	接收中断使能	0
2	-	-	保留	0x0
1	RXEN	R/W	接收使能	0x0
0	TXEN	R/W	发送使能	0x0

### 17.5.4 UARTx\_INTSTS 寄存器

表 17-6 UARTx\_INTSTS 寄存器

位	名称	类型	描述	默认值
31: 5	-	-	保留	0
5	TXDONEIF	R/C	发送完成标志。当发送单个字节并且 TXDONEIE 为 1 时，该标志位将被置 1。向该位写 1 或向 UARTx_STATE 寄存器中的 TXDONE 位写 1，该位将被清零。	0x0
4	RXPEIF	R/C	接收数据奇偶校验错误标志。当接收数据的奇偶校验值与期望不同时，该标志会被置 1。向该位写 1 或向 UARTx_STATE 寄存器的 RXPE 位写入 1 时，该位将被清零。	0x0
3	RXOVIF	R/C	接收缓存区溢出标志。当 RXOVIE 为 1、RXFULL 为 1 并且从 RX 接收机收到一个新数据时，该标志将被置 1。向该位或向 UARTx_STATE 寄存器中的 RXOV 位写 1，该位将被清零。	0x0
2	TXOVIF	R/C	发送缓存区溢出标志。当 TXOVIE 为 1、UART 数据发送没有完成且向 UARTx_DATA 寄存器写入一个新数据，该标志将置 1。向该位写 1 或向 UARTx_STATE 寄存器中的 TXOV 位写 1，该位将被清零。	0x0
1	RXIF	R/C	接收中断标志，当 UARTx_DATA 寄存器接收到数据时，该位置 1。写 1 清零该位。	0x0
0	-	-	保留	0

### 17.5.5 UARTx\_BAUDDIV 寄存器

表 17-7 UARTx\_BAUDDIV 寄存器

位	名称	类型	描述	默认值
31: 20	-	-	保留	0
19: 0	BAUDDIV	R/W	波特率分频寄存器。在使能 UART 之前必须先设置好该寄存器。 BAUDDIV=APBCLK/Baud_rate。	0x0

			例如当 APBCLK=6.5536MHz, 波特率是 9600bps 时, 该寄存器值等于 $6553600/9600=682$ 。	
--	--	--	--	--

## 17.5.6 UARTx\_CTRL2 寄存器

表 17-8 UARTx\_CTRL2 寄存器

位	名称	类型	描述	默认值
31: 4	-	-	保留	0
3: 1	PMODE	R/W	UART 数据校验模式控制寄存器。 0: 无校验位, 1+8+1 模式。 1: 偶校验 3: 奇校验 5: 校验位始终为 0 7: 校验位始终为 1。 其它: 保留	0x0
0	MSB	R/W	数据位传输顺序控制寄存器。 0: 低位先传 (LSB) 1: 高位先传 (MSB)	0x0

## 17.5.7 MISC1\_IREN 寄存器

表 17-9 MISC1\_IREN 寄存器

位	名称	类型	描述	默认值
31: 6	-	-	保留	0
5: 0	IREN	R/W	IR 使能控制寄存器。该寄存器中的每个位对应于 1 个 UARTTX 通道。当 IREN[x]置 1 时, 表示 UARTTX[x]输出为红外载波 (IR) 调制信号。	0x00

## 17.5.8 MISC1\_DUTYL 寄存器

表 17-10 MISC1\_DUTYL 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	DUTYL	R/W	红外载波 (IR) 调制波低电平宽度控制寄存器。低脉冲宽度将是 $(DUTYL+1)*APBCLK$ 周期。	0x0000

## 17.5.9 MISC1\_DUTYH 寄存器

表 17-11 MISC1\_DUTYH 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	DUTYH	R/W	红外载波 (IR) 调制波高电平宽度控制寄存器。高脉冲宽度为	0x0000

			(DUTYH+1) *APBCLK 周期。	
--	--	--	-----------------------	--

## 第18章 UART32K 控制器

### 18.1 简介

UART32K 控制器使用 UART 协议接收数据。V85XXP 中有两个独立的 UART32K 控制器，其输入引脚从普通串口 UART0~UART3 的输入脚（RX0~RX3）中选择。UART32K 控制器的工作频率是 RTCCLK，因此可以在浅睡眠或深睡眠模式下工作。UART32K 支持的典型波特率为 9600bps，支持数据长度为 8 位，支持无奇偶校验的 8 位数据和有奇偶校验的 9 位数据传输，接收机有一个数据缓冲器和一个移位寄存器。UART32K 的波特率支持范围 300~14400bps。

### 18.2 特点

- 可在浅睡眠或深睡眠模式下工作；
- 具有浅睡眠或深睡眠唤醒功能；
- 可编程波特率发生器；
- 支持无奇偶校验的 8 位数据和有奇偶校验的 9 位数据传输；
- 奇偶校验错误检测；
- 接收中断标志；
- 接收奇偶校验错误标志；
- 接收溢出中断标志；

### 18.3 功能框图

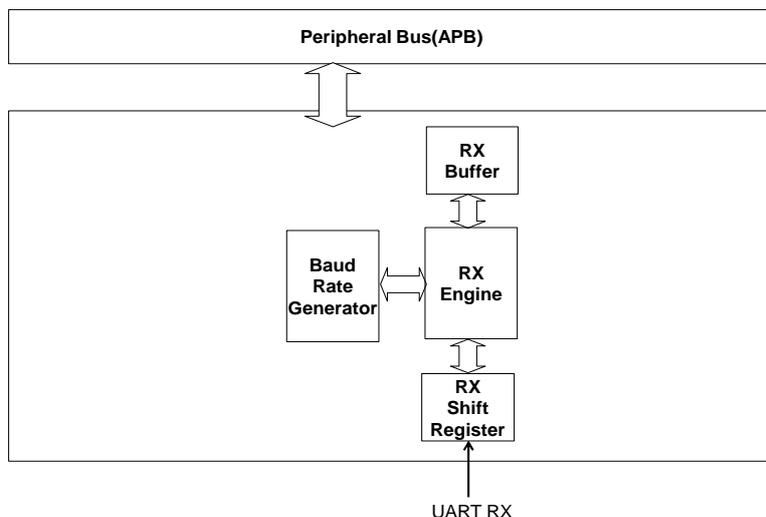


图 18-1 UART32K 控制器的功能框图

### 18.4 寄存器地址

表 18-1 UART32K 控制器(UART32K0 基地址: 0x40014100)

名称	类型	地址	描述	默认值
U32K0_CTRL0	R/W	0x0000	UART32K0 控制寄存器 0	0x000
U32K0_CTRL1	R/W	0x0004	UART32K0 控制寄存器 1	0x00
U32K0_BAUDDIV	R/W	0x0008	UART32K0 波特率分频寄存器	0x4B00

U32K0_DATA	R	0x000C	UART32K0 接收数据寄存器	--
U32K0_STS	R/C	0x0010	UART32K0 中断状态寄存器	0x0

**UART32K1 基地址: 0x40014180**

名称	类型	地址	描述	默认值
U32K1_CTRL0	R/W	0x0000	UART32K1 控制寄存器 0	0x00
U32K1_CTRL1	R/W	0x0004	UART32K1 控制寄存器 1	0x00
U32K1_BAUDDIV	R/W	0x0008	UART32K1 波特率分频寄存器	0x4B00
U32K1_DATA	R	0x000C	UART32K1 接收数据寄存器	--
U32K1_STS	R/C	0x0010	UART32K1 中断状态寄存器	0x0

**18.5 寄存器定义****18.5.1 U32Kx\_CTRL0 寄存器**

表 18-2 U32Kx\_CTRL0 寄存器

位	名称	类型	描述	默认值
31: 9	-	-	保留	0
8	WKUMODE	R/W	唤醒模式控制。RXIE=1 时这个寄存器才会生效。 0: 只要接收到数据就会唤醒（不管校验是否正确） 1: 只有当奇偶校验位和停止位正确才会唤醒	0x0
7: 6	DEBSEL	R/W	消抖控制。 0: 无消抖。 1: 2 个 RTCCLK 时钟周期消抖 2: 3 个 RTCCLK 时钟周期消抖 3: 4 个 RTCCLK 时钟周期消抖 当 UART 总线上有干扰时, 启用消抖功能可在接收数据时降低干扰。但是消抖周期大时可能会使接收数据丢失。	0x0
5: 3	PMODE	R/W	UART 数据校验模式控制寄存器。 0: 无校验位, 1+8+1 模式 1: 偶校验 3: 奇校验 5: 校验位始终为 0 7: 校验位始终为 1 其它: 保留	0x0
2	MSB	R/W	数据位接收顺序控制。 0: 低位先收 (LSB) 1: 高位先收 (MSB)	0x0

1	ACOFF	R/W	自动校正控制。为了在 RTCCLK 时钟下正确接收数据，控制器内部有一个硬件自动校正机制。默认情况下，硬件自动校正机制是打开的，向该位写 1 可以关闭硬件自动校正机制。 0: 开启自动校正 1: 关闭自动校正	0x0
0	EN	R/W	UART32K 控制器使能。将该位置 1，UART32K 控制器将在两个 RTCCLK 时钟周期后开始接收数据。因此在将该位置 1 之前，必须为各个 U32K 控制器设置正确的参数。 0: 禁止 1: 使能	0x0

## 18.5.2 U32Kx\_CTRL1 寄存器

表 18-3 U32Kx\_CTRL1 寄存器

位	名称	类型	描述	默认值
31: 6	-	-	保留	0
5: 4	RXSEL	R/W	接收数据源控制。 0: 来自 UART RX0(IOA12) 1: 来自 UART RX1(IOA13) 2: 来自 UART RX2(IOA14) 3: 来自 UART RX3(IOA15)	0x0
3	-	-	保留	0
2	RXOVIE	R/W	接收溢出中断/唤醒使能	0x0
1	RXPEIE	R/W	接收奇偶校验错误中断/唤醒使能	0x0
0	RXIE	R/W	接收中断/唤醒使能。使能后，当 WKUMODE 为 0 时，只要接收到串口数据就会唤醒；当 WKUMODE 为 1 时，只有当奇偶校验位和停止位正确才会唤醒。	0x0

## 18.5.3 U32Kx\_BAUDDIV 寄存器

表 18-4 U32Kx\_BAUDDIV 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	BAUDDIV	R/W	波特率分频寄存器，在使能 UART32K 之前必须设置这个寄存器。 $BAUDDIV = 65536 * \text{Baud\_rate} / \text{RTCCLK}$ 。 例如，当波特率是 9600bps，RTCCLK 频率为 32768， $65536 * 9600 / 32768 = 19200$ 。 当 RTC_PSCA 寄存器的 PSCA=1，BAUDDIV 应该使用 RTCCLK 的分频后的频率进行计算。	0x4B00

## 18.5.4 U32Kx\_DATA 寄存器

表 18-5 U32Kx\_DATA 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7: 0	DATA	R	读：接收数据	--

## 18.5.5 U32Kx\_STS 寄存器

表 18-6 U32Kx\_STS 寄存器

位	名称	类型	描述	默认值
31: 3	-	-	保留	0
2	RXOV	R/C	接收缓存器溢出标志。当接收 FIFO 满并且从 RX 接收机接收到一个新数据时，该标志位置 1。写 1 清零该位。	0x0
1	RXPE	R/C	接收数据奇偶校验错误标志。当接收数据的奇偶校验值错误时，该标志位置 1。写 1 清零该位。	0x0
0	RXIF	R/C	接收中断标志位，当 WKUMODE 为 0 时，只要接收到串口数据就会置 1；当 WKUMODE 为 1 时，只有当奇偶校验位和停止位正确才会置 1。写 1 清零该位。	0x0

## 第19章 ISO7816 控制器

### 19.1 简介

ISO7816是一种增强型UART协议，能够在2线总线上进行半双工通信。V85XXP内部集成2个可独立编程的ISO7816控制器，并拥有独立的CPU中断。IP中内置一个时钟分频器，能够为ISO7816器件提供1~5MHz的时钟。从浅睡眠或深睡眠模式唤醒后，ISO7816控制器的设置会被复位，所以，在程序唤醒后应该手动恢复相关的设置。ISO7816波特率支持范围：200~2625000bps。

### 19.2 特点

- 可编程波特率发生器；
- 支持半双工发送/接收；
- 支持重发机制，支持重发次数控制；
- 支持时钟分频。

### 19.3 功能框图

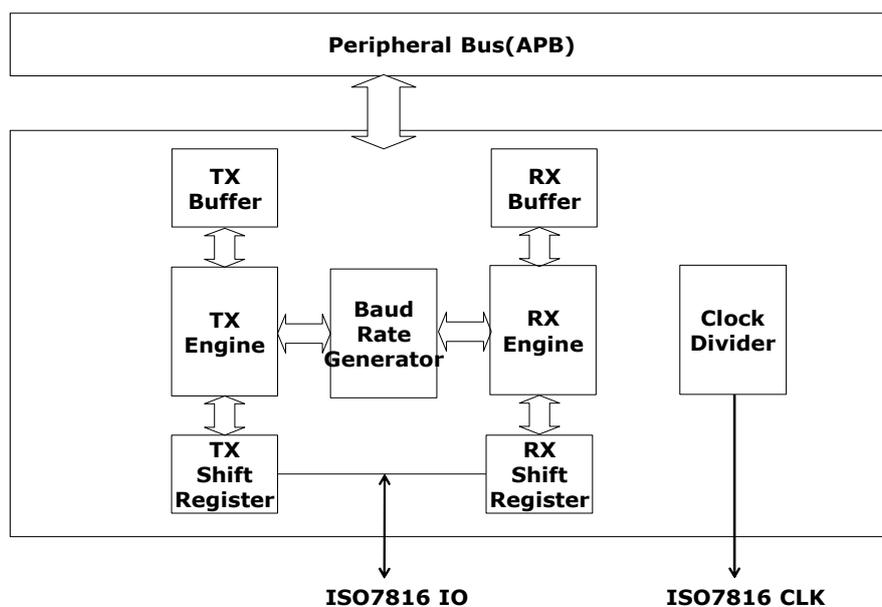


图 19-1 ISO7816 控制器功能框图

### 19.4 寄存器地址

表 19-1 ISO7816 控制器地址(ISO7816 基地址: 0x40012000)

名称	类型	地址	描述	默认值
ISO78160_BAUDDIVL	R/W	0x0004	ISO78160 波特率低字节寄存器	0x00
ISO78160_BAUDDIVH	R/W	0x0008	ISO78160 波特率高字节寄存器	0x00

ISO78160_DATA	R/W	0x000C	ISO78160 数据寄存器	0x00
ISO78160_INFO	R/C	0x0010	ISO78160 信息寄存器	0x00
ISO78160_CFG	R/W	0x0014	ISO78160 控制寄存器	0x00
ISO78160_CLK	R/W	0x0018	ISO78160 时钟分频器控制寄存器	0x00
ISO78161_BAUDDIVL	R/W	0x0044	ISO78161 波特率低字节寄存器	0x00
ISO78161_BAUDDIVH	R/W	0x0048	ISO78161 波特率高字节寄存器	0x00
ISO78161_DATA	R/W	0x004C	ISO78161 数据寄存器	0x00
ISO78161_INFO	R/C	0x0050	ISO78161 信息寄存器	0x00
ISO78161_CFG	R/W	0x0054	ISO78161 控制寄存器	0x00
ISO78161_CLK	R/W	0x0058	ISO78161 时钟分频器控制寄存器	0x00

## 19.5 寄存器定义

### 19.5.1 ISO7816x\_BAUDDIVL 寄存器

表 19-2 ISO7816x\_BAUDDIVL 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7: 0	BAUDDIVL	R/W	波特率分频器的低字节。	0x0

### 19.5.2 ISO7816x\_BAUDDIVH 寄存器

表 19-3 ISO7816x\_BAUDDIVH 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7: 0	BAUDDIVH	R/W	波特率分频的高字节。波特率计数器是从 BAUDDIV 数到 0xFFFF，所以波特率计算公式为： $\text{BAUDDIV} = 0x10000 - (\text{APBCLK} / \text{波特率})$ 例如：当 APBCLK=6.5536MHz，波特率=9600bps，寄存器值应该为： $0x10000 - (6553600 / 9600) = 0xFD56$ 。	0x0

### 19.5.3 ISO7816x\_DATA 寄存器

表 19-4 ISO7816x\_DATA 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7: 0	DATA	R/W	读：接收数据，当接收数据没有被取走，又接收到一个新的数据时，OVIF 标志位将会置 1。 写：发送数据，对该端口写将会触发 ISO7816 总线的一次数据发送。	0x0

## 19.5.4 ISO7816x\_INFO 寄存器

表 19-5 ISO7816x\_INFO 寄存器

位	名称	类型	描述	默认值
31: 10	-	-	保留	0
9	DMATXDONE	R/C	发送完成标志。当通过 DMA 方式批量数据发送时，会在全部发送完成时自动置 1。用户必须等到 DMA 发送完成后才能读取此标志。写 1 清零该位。	0x0
8	TXRTYERRIF	R/C	传输重试计数器到达标志。当重试次数达到 TXRTYCNT 寄存器中的限制设置时，该位将置 1。当读取到该位为 1 时，有必要通过 ISO7816x_CFG 寄存器中的 EN 位来禁用和重新初始化整个 ISO7816 模块。写 1 清零该位。	0x0
7	RXOVIF	R/C	接收溢出标志。当已接收数据未被 CPU 读取时，接收到另一个数据，该位将置 1。写 1 清零该位。	0x0
6	TXDONEIF	R/C	发送完成中断标志。 当收到应答 (ACK) 为 1 或重试次数达到 TXRTYCNT 中的设置时，才会设置该位。写 1 清零该位。	0x0
5	RXIF	R/C	接收中断标志。当成功接收一字节数据并发送应答信号 (ACK)，该位将置 1。写 1 清零该位。	0x0
4: 3	-	-	保留	0x0
2	RXERRIF	R/C	接收校验和错误标志。接收数据有校验和错误时，该位将被置为 1。写 1 清零该位。	0x0
1	CHKSUM	R	发送或接收的数据的校验和位。这个位可以被 CPU 读/写，但原来的信息会丢失。	0x0
0	RXACK	R	在发送结束时收到的 ACK。这个位可以被 CPU 读/写，但原来的信息会丢失。	0x0

## 19.5.5 ISO7816x\_CFG 寄存器

表 19-6 ISO7816x\_CFG 寄存器

位	名称	类型	描述	默认值
31: 17	-	-	保留	0
16	RXACKSET	R/W	接收的 ACK 长度设置。 0: 2 位 1: 1 位	0x0
15 : 12	TXRTYCNT	R/W	发送重试次数设置寄存器。 0: 不重试 (只要接收 ACK 为 0, TXRTYERRIF 立即置为 1, 同时 ISO7816 发送停止。) 1: 再试一次 2: 重试两次 ... 15: 重试 15 次 当达到重试次数但仍收到错误 ACK, TXRTYERRIF 位将置为 1, 同时 ISO7816 发送停止。并且不再向 DMA 控制器做 DMA 请	0x0

			求。这在使用 DMA 传输数据时很有用，用户可以使用 DMATXDONE 来确定传输是否已终止，并通过 TXRTYERRIF 查看传输是否发生错误。当 TXRTYERRIF 为 1 时，用户必须向 DMA_CxCTL 寄存器的 STOP 位写 1 强制停止 DMA。如果该 DMA 传输需要恢复，用户应将 STOP 位写零。	
11	LSB	R/W	数据位发送/接收顺序控制寄存器。 0: 高位先传 (MSB) 1: 低位先传 (LSB)	0x0
10	-	R/W	保留。建议用户配置为 1。	0x0
9	AUTORXACK	R/W	当接收到错误数据时，自动应答 ACK 为 0，让发送端重新发送数据。 0: 禁用自动重新接收模式。 1: 启用自动重新接收模式。	0x0
8	TXRTYERRIE	R/W	传输重试错误中断使能寄存器。 当达到重试次数但仍收到错误 ACK, TXRTYERRIF 位将置为 1。	0x0
7	RXOVIE	R/W	接收溢出中断使能寄存器。	0x0
6	TXDONEIE	R/W	发送完成中断使能寄存器。	0x0
5	RXIE	R/W	接收中断使能寄存器。	0x0
4	ACKLEN	R/W	接收错误数据后，发送 ACK 低电平长度设置。 0: 1 位 1: 2 位	0x0
3	-	R/W	保留	0x0
2	RXERRIE	R/W	接收错误中断使能寄存器。	0x0
1	CHKP	R/W	奇偶校验模式控制寄存器。 0: 偶校验 1: 奇校验	0x0
0	EN	R/W	ISO7816 使能寄存器。 0: 禁用 ISO7816 功能 1: 使能 ISO7816 功能	0x0

### 19.5.6 ISO7816x\_CLK 寄存器

表 19-7 ISO7816x\_CLK 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7	CLKEN	R/W	ISO7816 时钟输出使能。 0: 禁止时钟输出 1: 使能时钟输出	0x0
6: 0	CLKDIV	R/W	ISO7816 时钟分频值。 0: APBCLK/1	0x0

			1: APBCLK/2 2: APBCLK/3 ... 127: APBCLK/128	
--	--	--	--	--

下表显示了 ISO7816 的数据包格式。

表 19-8 ISO7816 数据包格式

bit	0	1~8	9	10	11	12	0	...
发送方向	发送机→接收机			接收机→发送机		--	发送机→接收机	
定义	Start bit	Data	Parity bit	ACK*		Wait	Start bit	...
值	0	MSB→LSB	P	ACK*		1	0	...
TX	0	MSB→LSB	P	1			0	...
RX	1	1	1	ACK*		1	1	...
描述	10 位数据			3 位保护时间		下一帧		

\*如果 ACK 为 0，则 RX 将在第 10 位中间从 1 复位为 0，在第 11 或第 12 位置为 1，由 ISO7816x\_CFG 中的 ACKLEN 控制。

发送包总共 10 位，包括 1 位起始位、8 位数据和 1 个奇偶校验位。有 3 位的保护时间，接收器检查接收机的正确性并发送 1 位或 2 位的响应 ACK。发送器和接收器在位 12 的位置都应该保持在高电平状态，等待下一个数据包。

## 第20章 Timer/PWM 控制器

### 20.1 简介

V85XXP 共有 8 个定时器。其中四个是通用 32 位定时器，用于定时功能。另外 4 个定时器是 16 位定时器，具有 PWM 比较和捕获功能。每个定时器可以单独产生 CPU 中断。每个 PWM 可以输出具有不同输出波形的 3 个输出。Timer/PWM 控制器中的设置将在从浅睡眠或深睡眠模式唤醒后复位，用户应在从这两种状态唤醒后手动恢复设置。

### 20.2 特点

- 4 个 32 位通用定时器；
- 4 个 16 位 PWM 输出和捕获定时器；
- 每个 PWM 定时器最多可以有 3 个输出和 3 个输入；
- 有 8 个输出模式可选择。

### 20.3 功能框图

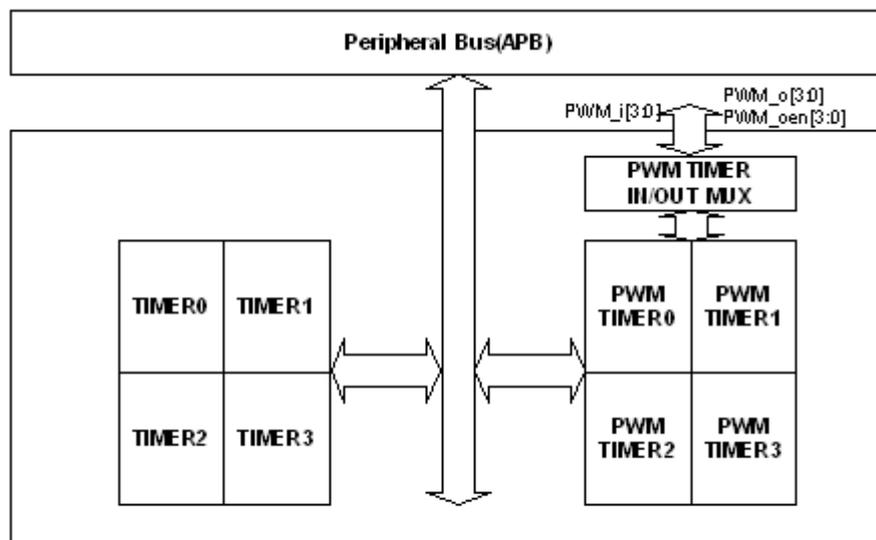


图 20-1 定时器功能框图

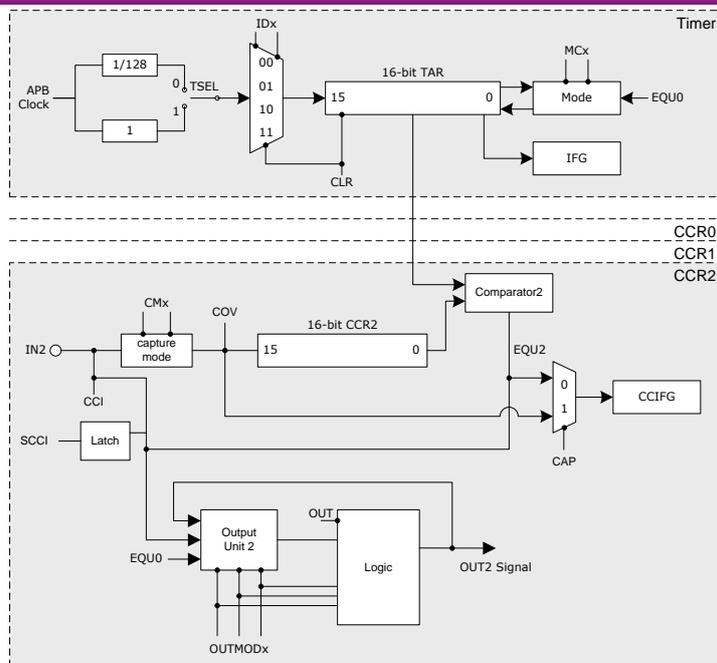


图 20-2 PWM 定时器功能框图

## 20.4 寄存器地址

表 20-1 32 位 TIMER 控制器(32 位 TIMER 基地址: 0x40012800)

名称	类型	地址	描述	默认值
TMR0_CTRL	R/W	0x0000	Timer0 控制寄存器	0x0
TMR0_VALUE	R/W	0x0004	Timer0 实时计数寄存器	0x00000000
TMR0_RELOAD	R/W	0x0008	Timer0 重载寄存器	0x00000000
TMR0_INTSTS	R/C	0x000C	Timer0 中断状态寄存器	0x0
TMR1_CTRL	R/W	0x0020	Timer1 控制寄存器	0x0
TMR1_VALUE	R/W	0x0024	Timer1 实时计数寄存器	0x00000000
TMR1_RELOAD	R/W	0x0028	Timer1 重载寄存器	0x00000000
TMR1_INTSTS	R/C	0x002C	Timer1 中断状态寄存器	0x0
TMR2_CTRL	R/W	0x0040	Timer2 控制寄存器	0x0
TMR2_VALUE	R/W	0x0044	Timer2 实时计数寄存器	0x00000000
TMR2_RELOAD	R/W	0x0048	Timer2 重载寄存器	0x00000000
TMR2_INTSTS	R/C	0x004C	Timer2 中断状态寄存器	0x0
TMR3_CTRL	R/W	0x0060	Timer3 控制寄存器	0x0
TMR3_VALUE	R/W	0x0064	Timer3 实时计数寄存器	0x00000000
TMR3_RELOAD	R/W	0x0068	Timer3 重载寄存器	0x00000000
TMR3_INTSTS	R/C	0x006C	Timer3 中断状态寄存器	0x0

表 20-2 16 位 PWM TIMER 控制器(16 位 PWM TIMER 基地址: 0x40012900)

名称	类型	地址	描述	默认值
PWM0_CTL	R/W	0x0000	PWM Timer0 控制寄存器	0x00

PWM0_TAR	R	0x0004	PWM Timer0 实时计数寄存器	0x0000
PWM0_CCTL0	R/W	0x0008	PWM Timer0 比较/捕获控制寄存器 0	0x000
PWM0_CCTL1	R/W	0x000C	PWM Timer0 比较/捕获控制寄存器 1	0x000
PWM0_CCTL2	R/W	0x0010	PWM Timer0 比较/捕获控制寄存器 2	0x000
PWM0_CCR0	R/W	0x0014	PWM Timer0 比较/捕获数据寄存器 0	0x0000
PWM0_CCR1	R/W	0x0018	PWM Timer0 比较/捕获数据寄存器 1	0x0000
PWM0_CCR2	R/W	0x001C	PWM Timer0 比较/捕获数据寄存器 2	0x0000
PWM1_CTL	R/W	0x0020	PWM Timer1 控制寄存器	0x00
PWM1_TAR	R	0x0024	PWM Timer1 实时计数寄存器	0x0000
PWM1_CCTL0	R/W	0x0028	PWM Timer1 比较/捕获控制寄存器 0	0x000
PWM1_CCTL1	R/W	0x002C	PWM Timer1 比较/捕获控制寄存器 1	0x000
PWM1_CCTL2	R/W	0x0030	PWM Timer1 比较/捕获控制寄存器 2	0x000
PWM1_CCR0	R/W	0x0034	PWM Timer1 比较/捕获数据寄存器 0	0x0000
PWM1_CCR1	R/W	0x0038	PWM Timer1 比较/捕获数据寄存器 1	0x0000
PWM1_CCR2	R/W	0x003C	PWM Timer1 比较/捕获数据寄存器 2	0x0000
PWM2_CTL	R/W	0x0040	PWM Timer2 控制寄存器	0x00
PWM2_TAR	R	0x0044	PWM Timer2 实时计数寄存器	0x0000
PWM2_CCTL0	R/W	0x0048	PWM Timer2 比较/捕获控制寄存器 0	0x000
PWM2_CCTL1	R/W	0x004C	PWM Timer2 比较/捕获控制寄存器 1	0x000
PWM2_CCTL2	R/W	0x0050	PWM Timer2 比较/捕获控制寄存器 2	0x000
PWM2_CCR0	R/W	0x0054	PWM Timer2 比较/捕获数据寄存器 0	0x0000
PWM2_CCR1	R/W	0x0058	PWM Timer2 比较/捕获数据寄存器 1	0x0000
PWM2_CCR2	R/W	0x005C	PWM Timer2 比较/捕获数据寄存器 2	0x0000
PWM3_CTL	R/W	0x0060	PWM Timer3 控制寄存器	0x00
PWM3_TAR	R	0x0064	PWM Timer3 实时计数寄存器	0x0000
PWM3_CCTL0	R/W	0x0068	PWM Timer3 比较/捕获控制寄存器 0	0x000
PWM3_CCTL1	R/W	0x006C	PWM Timer3 比较/捕获控制寄存器 1	0x000
PWM3_CCTL2	R/W	0x0070	PWM Timer3 比较/捕获控制寄存器 2	0x000
PWM3_CCR0	R/W	0x0074	PWM Timer3 比较/捕获数据寄存器 0	0x0000
PWM3_CCR1	R/W	0x0078	PWM Timer3 比较/捕获数据寄存器 1	0x0000
PWM3_CCR2	R/W	0x007C	PWM Timer3 比较/捕获数据寄存器 2	0x0000
PWM_O_SEL	R/W	0x00F0	PWM 输出选择寄存器	0xDB51
PWM_I_SEL01	R/W	0x00F4	PWM0 PWM1 输入选择寄存器	0x00390024
PWM_I_SEL23	R/W	0x00F8	PWM2 PWM3 输入选择寄存器	0x0013000E

## 20.5 寄存器定义

## 20.5.1 TMRx\_CTRL 寄存器

表 20-3 TMRx\_CTRL 寄存器

位	名称	类型	描述	默认值
31: 4	-	-	保留	0
3	INTEN	R/W	Timerx 中断使能寄存器。	0x0
2	EXTCLK	R/W	选择外部时钟信号 ext_clk (IOB15)*作为时钟源。ext_clk 的上升沿将作为内部计数器向下计数的使能信号。每到来一个外部时钟 ext_clk 上升沿，内部计数器减 1。该模式下，外部时钟信号 ext_clk 的频率应低于 PCLK/6。	0x0
1	EXTEN	R/W	选择 ext_clk (IOB15)管腿作为时钟使能 IO 口。当 EXTEN 为 1，并且 ext_clk 为 1 时，内部计数器将以 PCLK 的时钟速率递减 1。当 EXTEN 为 1，并且 ext_clk 为 0，内部计数器不会变化。ext_clk 会有两个周期的同步时钟，以避免 ext_clk 上的毛刺。当 EXTCLK 和 EXTEN 同时使能，EXTCLK 优先级更高。	0x0
0	EN	R/W	Timerx 使能控制寄存器。	0x0

注 (\*): ext\_clk (IOB15)是名为 TIMEREXT\_CLK (IOB15)的引脚

## 20.5.2 TMRx\_VALUE 寄存器

表 20-4 TMRx\_VALUE 寄存器

位	名称	类型	描述	默认值
31: 0	VALUE	R/W	Timerx 实时计数值寄存器。	0x00000000

## 20.5.3 TMRx\_RELOAD 寄存器

表 20-5 TMRx\_RELOAD 寄存器

位	名称	类型	描述	默认值
31: 0	RELOAD	R/W	Timerx 重载寄存器。每当定时器向下计数到 0，RELOAD 的值将被重载到计数器 TMRx_VALUE 中，并重新开始向下计数。同时，INTSTS 位将被置 1。 定时器 RELOAD 计算公式如下： $RELOAD = Period * APBCLK - 1$ 。	0x00000000

## 20.5.4 TMRx\_INTSTS 寄存器

表 20-6 TMRx\_INTSTS 寄存器

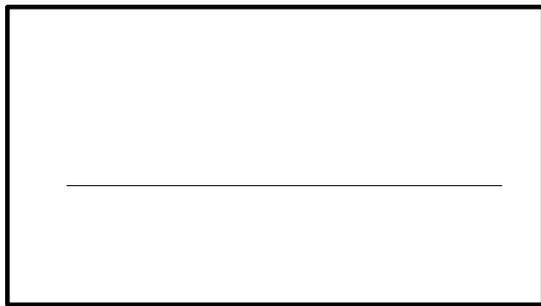
位	名称	类型	描述	默认值
31: 1	-	-	保留	0
0	INTSTS	R/C	Timerx 中断状态寄存器，当计数器向下计数到 0 时，该位被置 1。写 1 清零该位。	0x0

## 20.5.5 PWMx\_CTL 寄存器

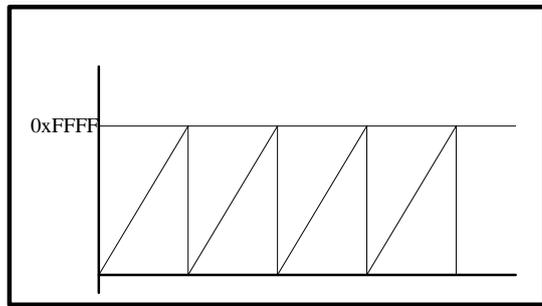
表 20-7 PWMx\_CTL 寄存器

位	名称	类型	描述	默认值
---	----	----	----	-----

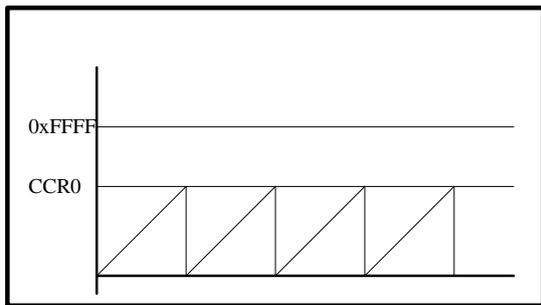
31: 8	-	-	保留	0
7: 6	ID	R/W	PWM Timerx 输入时钟分频控制。 0: 输入时钟 1/2 分频 1: 输入时钟 1/4 分频 2: 输入时钟 1/8 分频 3: 输入时钟 1/16 分频	0x0
5: 4	MC	R/W	PWM Timer 模式控制。 0: 停止计数。 1: 向上计数模式: 计数器向上计数到 CCR0, 并从 0 重新开始。 2: 连续计数模式: 计数器向上计数到 0xFFFF, 并从 0 重新开始。 3: 向上/向下计数模式: 计数器向上计数到 CCR0, 并从 CCR0 向下计数到 0。	0x0
3	TSEL	R/W	时钟源选择。 0: APB 时钟/128 1: APB 时钟	0x0
2	CLR	R/W	TAR 清零寄存器, 当该位设置为 1 时, TAR 将清零。在 TAR 清零后, 该位将自动清零。	0x0
1	IE	R/W	PWM Timerx 中断使能寄存器。	0x0
0	IFG	R/C	PWM Timerx 中断状态标志, 写 1 清零该标志。 向上计数模式: 当计数器从 CCR0 变为 0 时标志位置 1 连续计数模式: 当计数器从 0xFFFF 变为 0 时标志位置 1 向上/向下计数模式: 当计数器从 0x0001 到 0 时标志位置 1	0x0



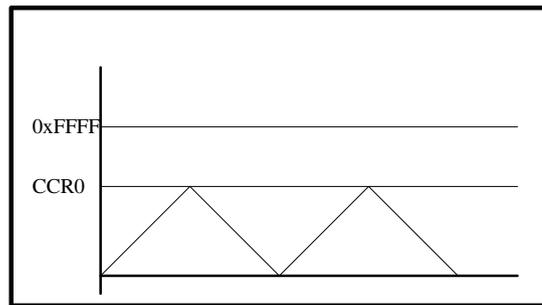
**Stop Mode**  
Timer stop.



**Continuous Mode**  
Timer will count to 0xFFFF, and restart from 0 and count to 0xFFFF.



**Up Mode**  
Timer will count to CCR0, and restart from 0 to CCR0.



**Up/Down Mode**  
Timer will count to CCR0, and then down count to 0x0, and then up\_count to CCR0 again.

图 20-3 PWM 定时器模式

MC 设置为非零值时，定时器将开始工作。PWM Timer 工作模式为向上计数模式和向上/向下计数模式时，若设置 PWMx\_CCR0 为 0，定时器 x 将停止计数。如果重新向 PWMx\_CCR0 写入非零值，定时器将重新开始计数。

向上计数模式下，CCR0 的值在计数期间被改变并且 CCR0 的新值大于当前计数值，则定时器将计数到新的 CCR0，然后返回到 0。如果 CCR0 的新值小于当前计数值，定时器将计数一个节拍后重置计数器值为 0 并重新开始计数。

向上/向下计数模式下，向上计数期间，如果 CCR0 的值在计数期间被改变并且 CCR0 的新值大于当前计数值，则定时器将计数到新的 CCR0，然后递减计数到 0。如果新的 CCR0 小于当前计数值，则定时器将计数一个节拍后开始向下计数到 0。向下计数期间，如果 CCR0 的值在计数期间被改变，定时器将连续完成向下计数过程。

连续模式下，用户可以动态设置 CCR0~CCR2 的值控制输出频率。下图显示了此应用的示例。

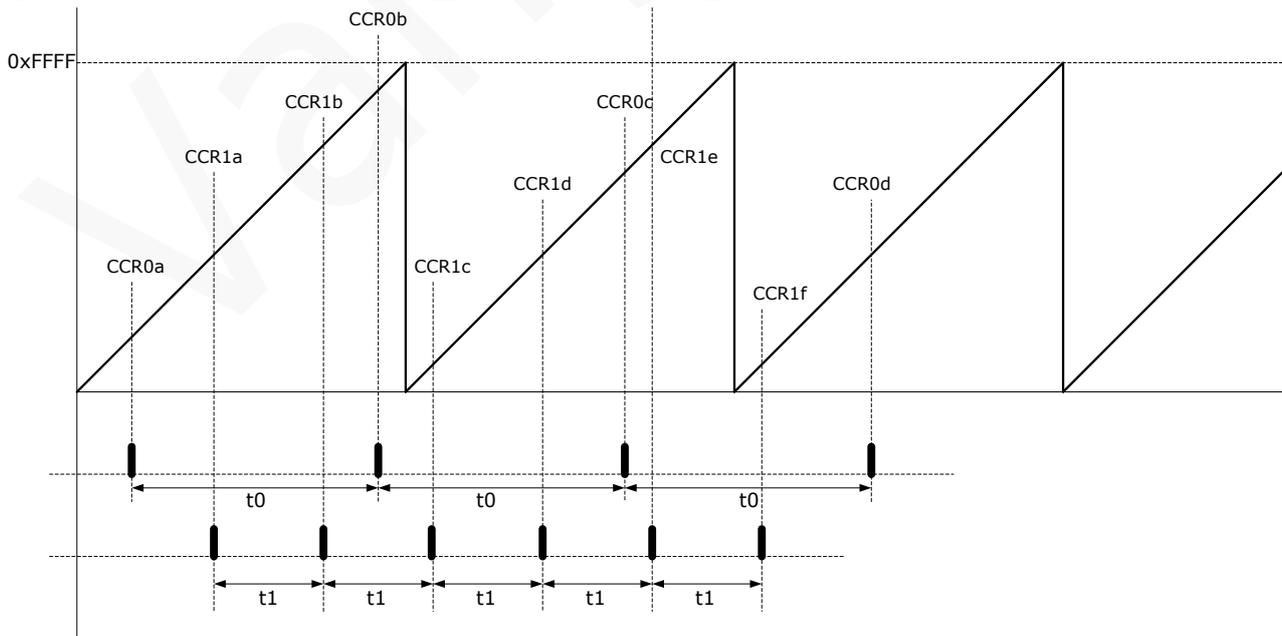


图 20-4 连续模式示例

上图中，CCR0a、CCR1a 分别是 CCR0、CCR1 在时刻 Ta0、Ta1 的值，CCR0b、CCR1b 分别是 CCR0、CCR1 在时刻 Tb0、Tb1 的值（Tb0=Ta0+t0，Tb1=Ta1+t1）。

上图展示在连续模式下，用户可以设置 CCIFG（CCTLx 的位 0）和 CCIE（CCTLx 的位 4），并设置 CCR0 和 CCR1，则当 TAR=CCR0a 或 TAR=CCR1a 时可产生中断信号。当 TAR=CCR0b 或 TAR=CCR1b 时，可以产生另一个具有不同周期的中断。t0 和 t1 完全独立，所以一个 PWM 定时器最多可设置 3 个完全独立的中断。在此模式下，当定时器计数从 0xFFFF 到 0x0 时，IFG 标志将被置 1

## 20.5.6 PWMx\_TAR 寄存器

表 20-8 PWMx\_TAR 寄存器

位	名称	类型	描述	默认值
31: 0	TAR	R/W	PWM Timer0 实时计数寄存器	0x00000000

## 20.5.7 PWMx\_CCTLx 寄存器

表 20-9 PWMx\_CCTLx 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 14	CM	R/W	捕获沿选择 0: 禁止捕获功能 1: 上升沿捕获 2: 下降沿捕获 3: 双边沿捕获	0x0
13: 11	-	-	保留	0
10	SCCI	R	当 TAR 等于 CCRx 时，当前通道输入管脚的电平状态。	0x0
9	OUTEN	R/W	OUTx 输出使能控制寄存器。 0: OUTx 输出禁止 1: OUTx 输出使能	0x0
8	CAP	R/W	捕获/比较模式选择。 0: 比较模式，用于产生 PWM 输出 1: 捕获模式，用于测量外部信号速率或对外部信号进行计数	0x0
7: 5	OUTMOD	R/W	输出模式选择。 0: 常数模式，OUTx 输出 OUT 位的值。 1: 置位模式，当 TAR=CCRx(x=0~2)时，OUTx 将置 1。 2: 翻转/复位模式，当 TAR=CCRx(x=1~2)时，OUTx 翻转，当 TAR=CCR0 时，OUTx 复位为 0。此模式不能用于 OUT0。 3: 置位/复位模式，当 TAR=CCRx(x=1~2)时，OUTx 置 1，当 TAR=CCR0 时，OUTx 复位为 0。此模式不能用于 OUT0。 4: 翻转模式，当 TAR=CCRx(x=0~2)时，OUTx 将会被翻转。 5: 复位模式，当 TAR=CCRx(x=0~2)时，OUTx 复位为 0。	0x0

			<p>6: 翻转/置位模式, 当 <math>TAR=CCR_x(x=1\sim2)</math> 时, <math>OUT_x</math> 被翻转, 当 <math>TAR=CCR_0</math> 时, <math>OUT_x</math> 置为 1。此模式不能用于 <math>OUT_0</math>。</p> <p>7: 复位/置位模式, 当 <math>TAR=CCR_x(x=1\sim2)</math> 时, <math>OUT_x</math> 复位为 0, 当 <math>TAR=CCR_0</math> 时, <math>OUT_x</math> 将置为 1。此模式不能用于 <math>OUT_0</math>。</p>	
4	CCIE	R/W	捕获/比较中断使能寄存器	0x0
3	-	-	保留	0
2	OUT	R/W	当 $OUTMOD$ 设置为 0 时, 该位用于控制 $OUT_x$ 的输出值。	0x0
1	COV	R/C	捕获溢出标志, 当 $CCIFG$ 位为 1 时发生另一个捕获事件, 该位将被置 1。写 1 清零该位。	0x0
0	CCIFG	R/C	<p>捕获模式下, 当捕获事件发生, 该位将被置 1, 并且 <math>TAR</math> 将被锁存到 <math>CCR_x</math> 中。</p> <p>比较模式下, 当 <math>TAR=CCR_x</math> 时, 该位将被置 1。</p> <p>写 1 清零该位。</p>	0

下图是在向上计数模式下的 PWM 输出:

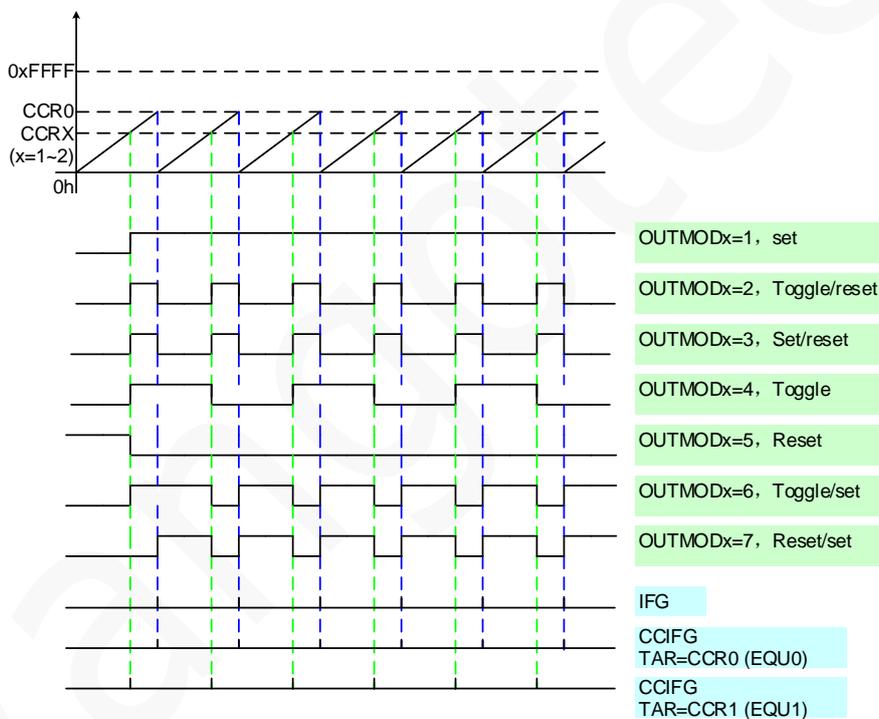


图 20-5 向上计数模式下的 PWM 输出

下图是在连续计数模式下的 PWM 输出:

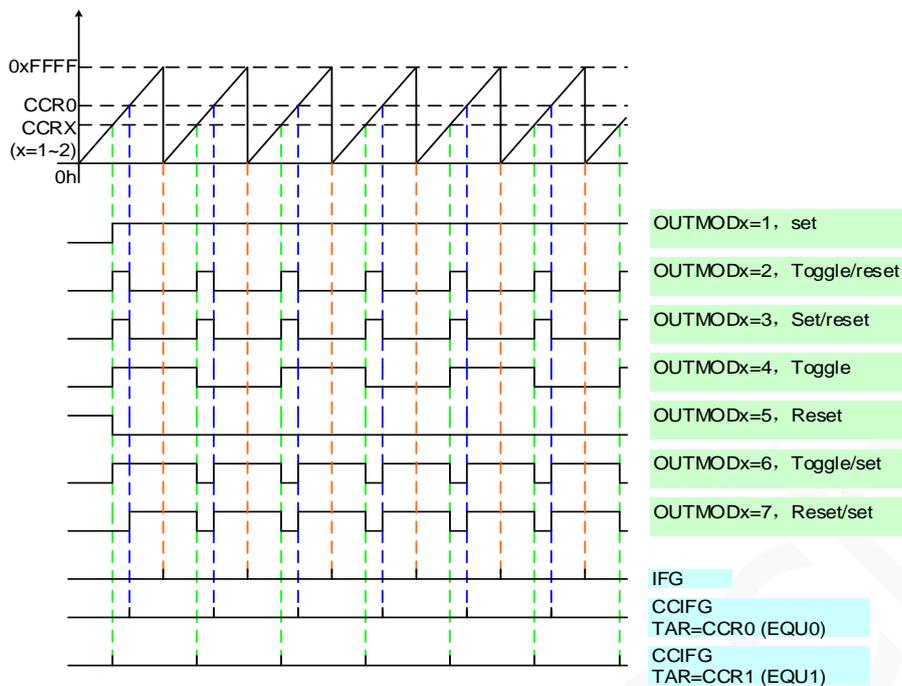


图 20-6 连续计数模式下的 PWM 输出

下图是在向上/向下计数模式下的 PWM 输出：

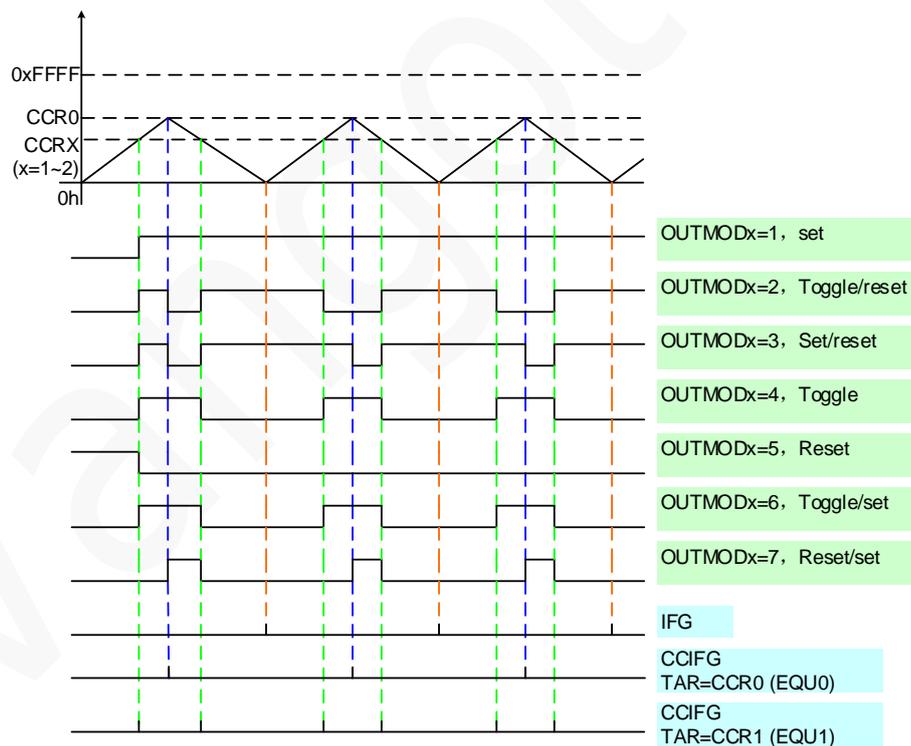


图 20-7 向上/向下计数模式下的 PWM 输出

### 20.5.8 PWMx\_CCRx 寄存器

表 20-10 PWMx\_CCRx 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0

15: 0	CCR <sub>x</sub>	R/W	比较/捕获数据寄存器。 捕获模式下，捕获的 TAR 将存储在此寄存器中。 比较模式下，该寄存器用于与 TAR 比较以生成 PWM 输出	0x0
-------	------------------	-----	---	-----

## 20.5.9 PWM\_O\_SEL 寄存器

表 20-11 PWM\_O\_SEL 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 12	SEL3	R/W	外部引脚输出 PWM3 的输出选择寄存器。与 SEL0 的定义相同。	0xD
11: 8	SEL2	R/W	外部引脚输出 PWM2 的输出选择寄存器。与 SEL0 的定义相同。	0xB
7: 4	SEL1	R/W	外部引脚输出 PWM1 的输出选择寄存器。与 SEL0 的定义相同。	0x5
3: 0	SEL0	R/W	外部输出引脚 PWM0 的输出选择寄存器。 0x0: 来自 PWM0 的 OUT0 0x1: 来自 PWM0 的 OUT1 0x2: 来自 PWM0 的 OUT2 0x4: 来自 PWM1 的 OUT0 0x5: 来自 PWM1 的 OUT1 0x6: 来自 PWM1 的 OUT2 0x8: 来自 PWM2 的 OUT0 0x9: 来自 PWM2 的 OUT1 0xA: 来自 PWM2 的 OUT2 0xC: 来自 PWM3 的 OUT0 0xD: 来自 PWM3 的 OUT1 0xE: 来自 PWM3 的 OUT2 其他: 保留	0x1

## 20.5.10 PWM\_I\_SEL01 寄存器

表 20-12 PWM\_I\_SEL01 寄存器

位	名称	类型	描述	默认值
31: 22	-	-	保留	0
21: 20	SEL12	R/W	PWM1 通道 2 (IN2) 的外部输入控制。 0: 来自 PWM0 管腿 1: 来自 PWM1 管腿 2: 来自 PWM2 管腿 3: 来自 PWM3 管腿	0x3
19: 18	SEL11	R/W	PWM1 通道 1 (IN1) 的外部输入控制。	0x2

			0: 来自 PWM0 管腿 1: 来自 PWM1 管腿 2: 来自 PWM2 管腿 3: 来自 PWM3 管腿	
17: 16	SEL10	R/W	PWM1 通道 0 (IN0) 的外部输入控制。 0: 来自 PWM0 管腿 1: 来自 PWM1 管腿 2: 来自 PWM2 管腿 3: 来自 PWM3 管腿	0x1
15: 6	-	-	保留	0
5: 4	SEL02	R/W	PWM0 通道 2 (IN2) 的外部输入控制。 0: 来自 PWM0 管腿 1: 来自 PWM1 管腿 2: 来自 PWM2 管腿 3: 来自 PWM3 管腿	0x2
3: 2	SEL01	R/W	PWM0 通道 1 (IN1) 的外部输入控制。 0: 来自 PWM0 管腿 1: 来自 PWM1 管腿 2: 来自 PWM2 管腿 3: 来自 PWM3 管腿	0x1
1: 0	SEL00	R/W	PWM0 通道 0 (IN0) 的外部输入控制。 0: 来自 PWM0 管腿 1: 来自 PWM1 管腿 2: 来自 PWM2 管腿 3: 来自 PWM3 管腿	0x0

### 20.5.11 PWM\_I\_SEL23 寄存器

表 20-13 PWM\_I\_SEL23 寄存器

位	名称	类型	描述	默认值
31: 22	-	-	保留	0
21: 20	SEL32	R/W	PWM3 通道 2 (IN2) 的外部输入控制。 0: 来自 PWM0 管腿 1: 来自 PWM1 管腿 2: 来自 PWM2 管腿 3: 来自 PWM3 管腿	0x1
19: 18	SEL31	R/W	PWM3 通道 1 (IN1) 的外部输入控制。	0x0

			0: 来自 PWM0 管腿 1: 来自 PWM1 管腿 2: 来自 PWM2 管腿 3: 来自 PWM3 管腿	
17: 16	SEL30	R/W	PWM3 通道 0 (IN0) 的外部输入控制。 0: 来自 PWM0 管腿 1: 来自 PWM1 管腿 2: 来自 PWM2 管腿 3: 来自 PWM3 管腿	0x3
15: 6	-	-	保留	0
5: 4	SEL22	R/W	PWM2 通道 2 (IN2) 的外部输入控制。 0: 来自 PWM0 管腿 1: 来自 PWM1 管腿 2: 来自 PWM2 管腿 3: 来自 PWM3 管腿	0x0
3: 2	SEL21	R/W	PWM2 通道 1 (IN1) 的外部输入控制。 0: 来自 PWM0 管腿 1: 来自 PWM1 管腿 2: 来自 PWM2 管腿 3: 来自 PWM3 管腿	0x3
1: 0	SEL20	R/W	PWM2 通道 0 (IN0) 的外部输入控制。 0: 来自 PWM0 管腿 1: 来自 PWM1 管腿 2: 来自 PWM2 管腿 3: 来自 PWM3 管腿	0x2

## 第21章 LCD 控制器

### 21.1 简介

LCD控制器用于在LCD面板上显示内容。LCD控制器支持4/6/8 COM模式，最多76段。LCD控制器内有两个具有自动切换功能的帧缓存器。LCD控制器中的设置在从深睡眠模式唤醒后会被复位，用户应在深睡眠模式唤醒后手动恢复设置。

### 21.2 特点

- 4/6/8 COM 模式;
- 支持最多 4COM×76SEG, 6COM×74SEG 或 8COM×72SEG;
- 支持 1/3 Bias 或 1/4 Bias 模式;
- LCD 的偏置电压由一个独立的 3.3 V LDO 产生, 在 2.85~3.6V 间以 50mV/LSB 的步长可调
- 内置电阻分压电路产生 LCD 波形电压;
- LCD 扫描频率由 RTCCLK 时钟产生;
- 两个帧缓存器, 支持自动切换功能;
- 可调帧速率。

### 21.3 功能框图

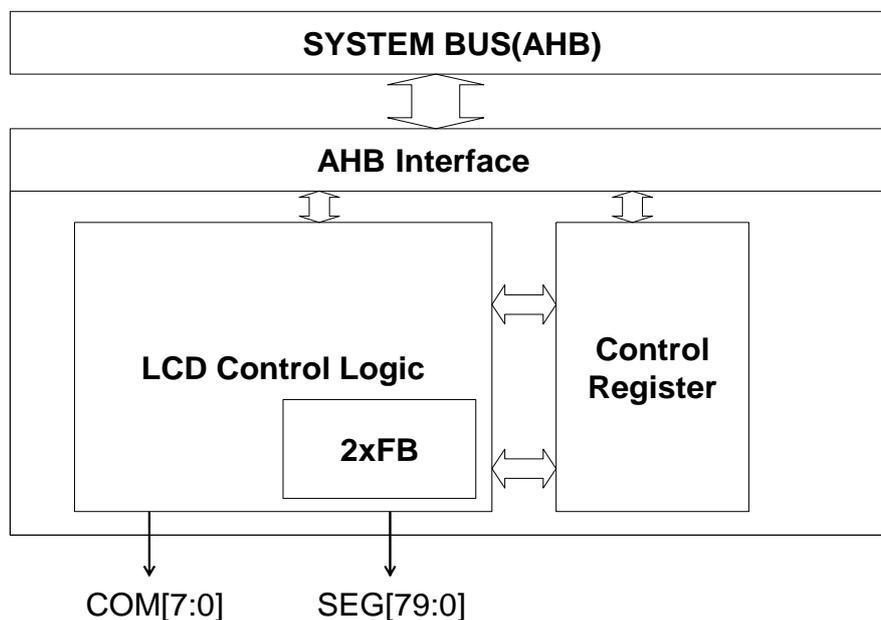


图 21-1 LCD 控制器的功能框图

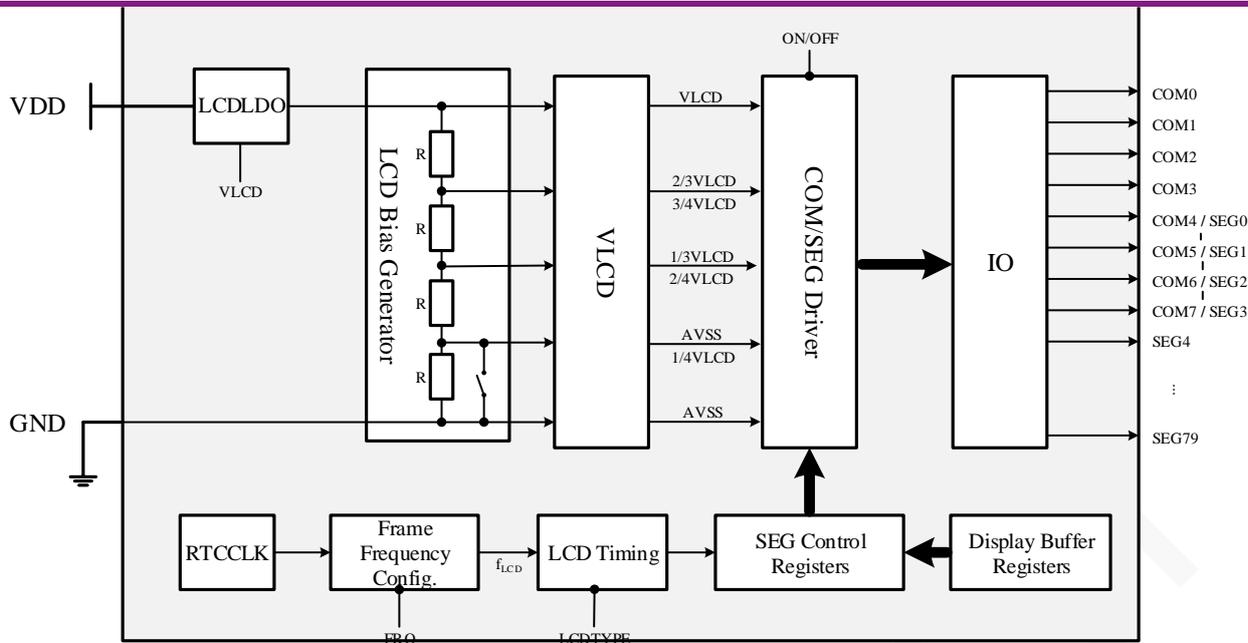


图 21-2 LCD 控制器结构图

## 21.4 LCD 时序

LCD 控制器用于在 LCD 面板上显示内容。LCD 控制器支持 4/6/8COM 模式，最多 76 段。LCD 控制器内有两个具有自动切换功能的帧缓存器。LCD 控制器中的设置在从深睡眠模式唤醒后会被复位，用户应在深睡眠模式唤醒后手动恢复设置。

## 21.5 LCD 波形电压

V85XXP 中，LCD 驱动电路的偏置电压由独立的 LDO 通过内部分压电阻电路产生 LCD 偏置电压 (VLCD)，用于产生 LCD 驱动波形。用户可通过 ANA\_REG6 寄存器中的控制位 VLCD[4: 1] 对波形电压进行调整。

## 21.6 LCD 工作电流

用户可以改变寄存器 LCD\_CTRL 中的控制位 DRV，配置偏置电压产生电路中的各分压电阻的阻值，改变电路电流大小，从而调整液晶屏显示亮度。默认情况下各分压电阻的阻值为 300kΩ。

## 21.7 LCD 驱动波形

LCD 偏置电压产生电路中有 4 个串联电阻，可以配置为 1/3 Bias 模式或 1/4 Bias 模式。用户可通过配置 LCDBMOD 位 (ANA\_REG6 的 bit0) 开启或关闭一个电阻，从而配置 LCD 为 1/3 偏置或 1/4 偏置模式。

下图为[1/4Duty, 1/3Bias]的 LCD 驱动波形 (示例)。

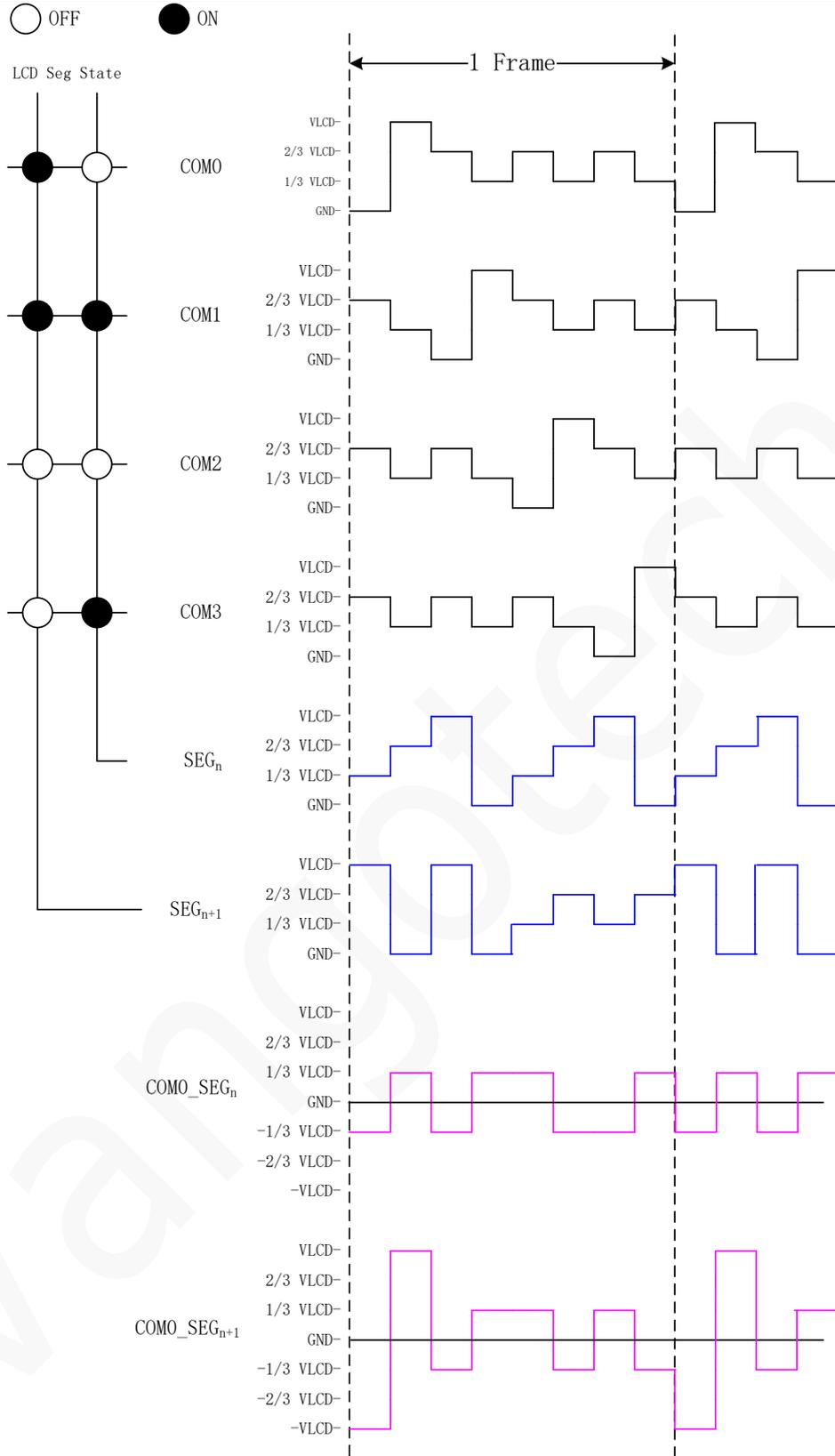


图 21-3 LCD 波形[1/4 Duty, 1/3 Bias]

下图为[1/6Duty, 1/3Bias]的 LCD 驱动波形（示例）。

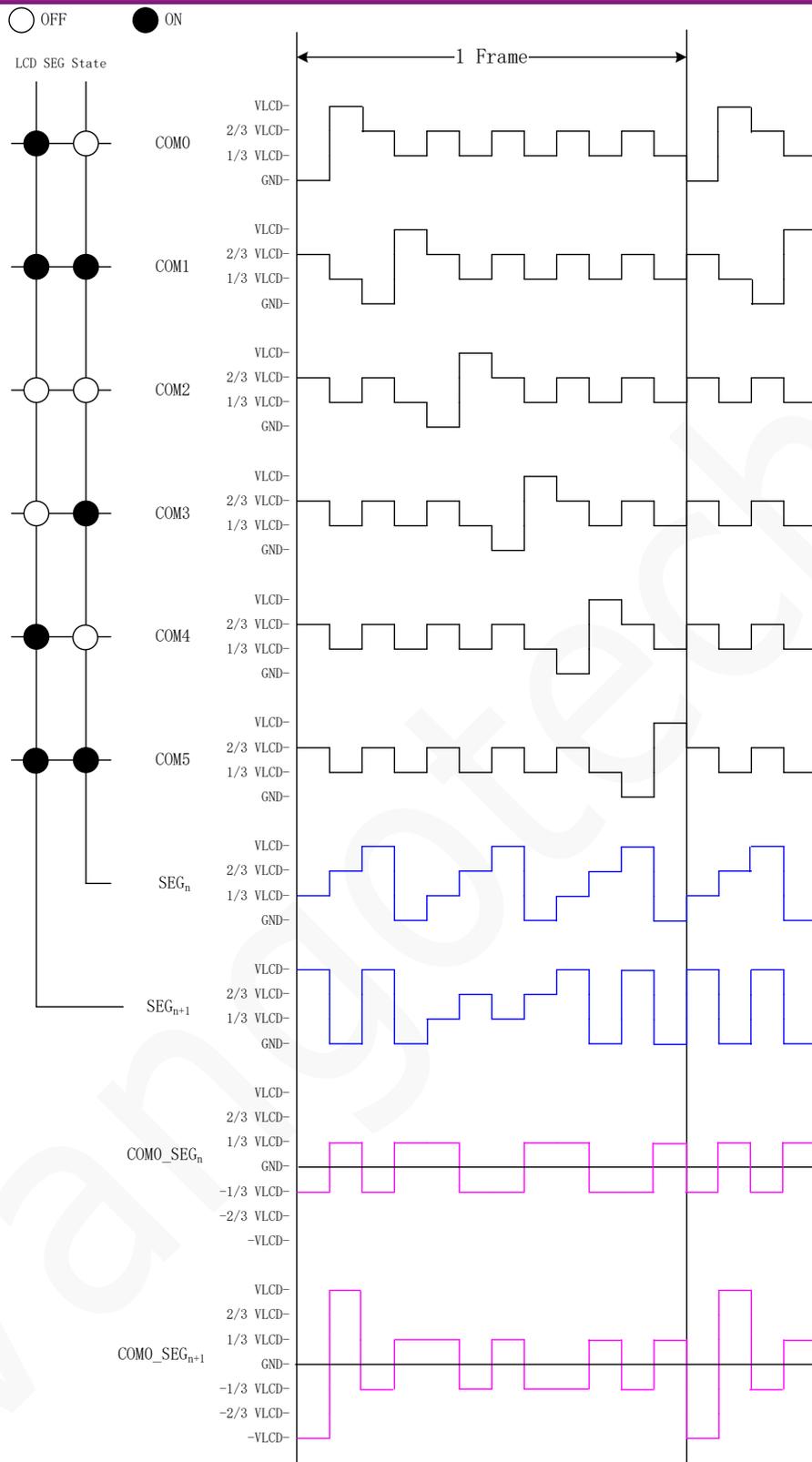


图 21-4 LCD 波形[1/6 Duty, 1/3 Bias]

下图为[1/8Duty, 1/3Bias]的 LCD 驱动波形（示例）

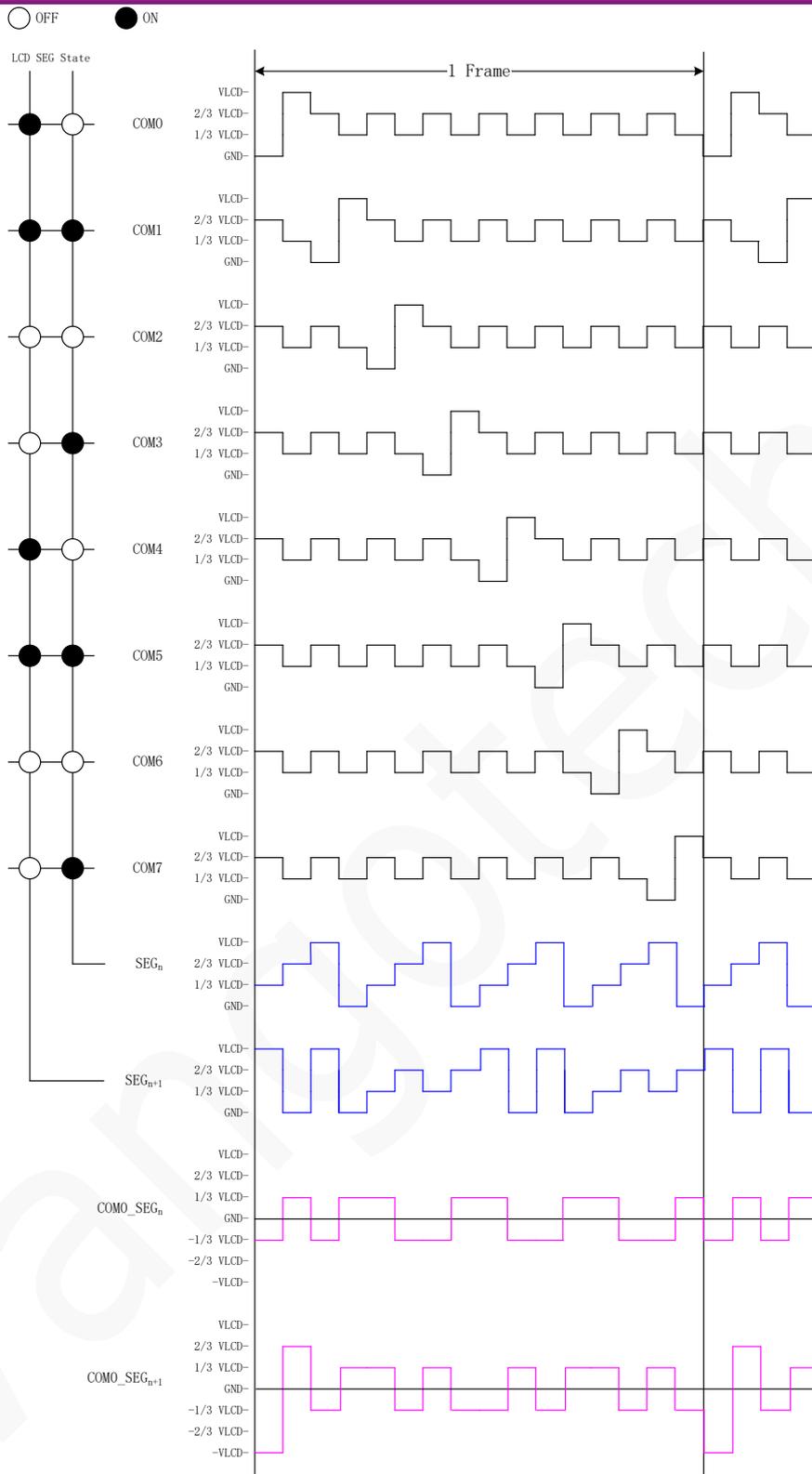


图 21-5 LCD 波形[1/8 Duty, 1/3 Bias]

下图为[1/8Duty, 1/4Bias]的 LCD 驱动波形（示例）

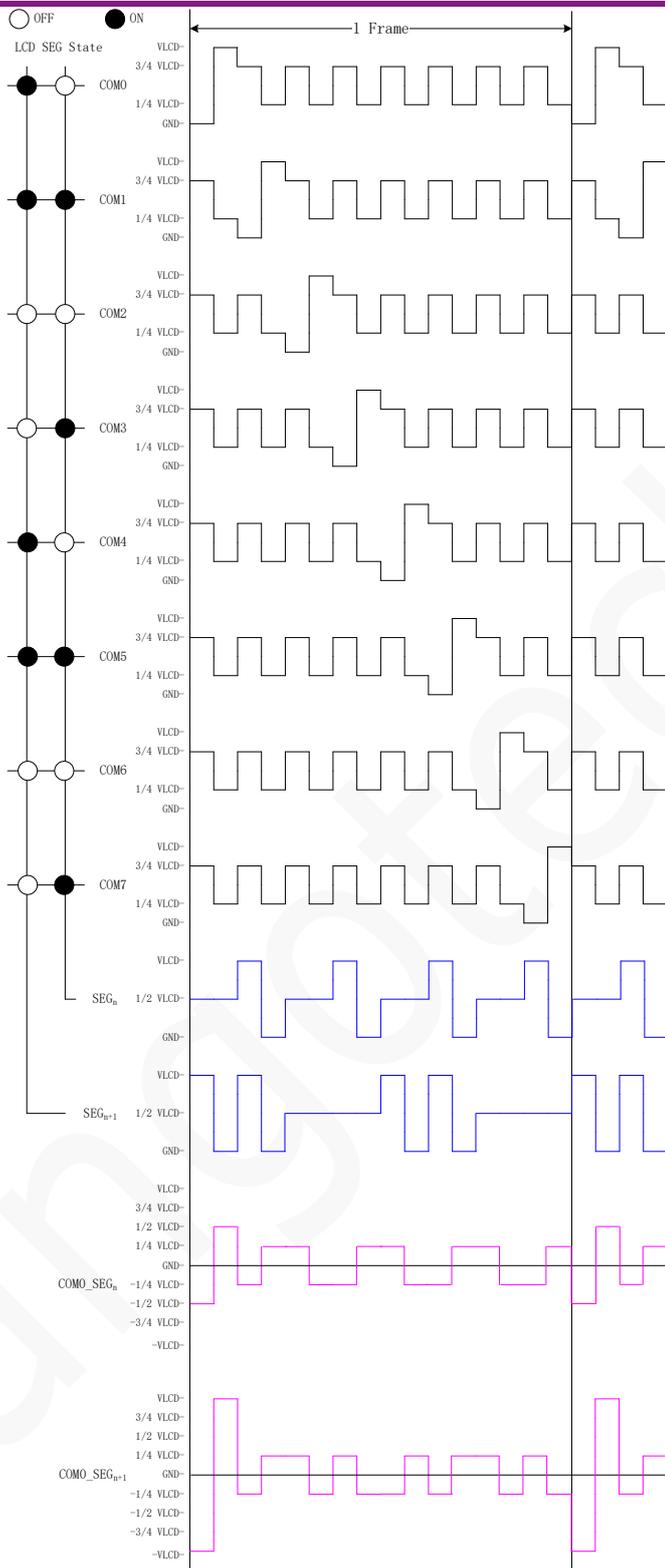


图 21-6 LCD 波形[1/8 Duty , 1/4 Bias]

## 21.8 寄存器地址

表 21-1 与 LCD 相关的 ANA 控制器(ANA 基地址: 0x40014200)

名称	类型	地址	描述	默认值
ANA_REG6	R/W	0x0018	模拟寄存器 6	0x00

表 21-2 LCD 控制器(LCD 基地址: 0x40002000)

名称	类型	地址	描述	默认值
----	----	----	----	-----

LCD_FB00	R/W	0x0000	LCD 帧缓存 0 寄存器	--
LCD_FB01	R/W	0x0004	LCD 帧缓存 1 寄存器	--
LCD_FB02	R/W	0x0008	LCD 帧缓存 2 寄存器	--
LCD_FB03	R/W	0x000C	LCD 帧缓存 3 寄存器	--
LCD_FB04	R/W	0x0010	LCD 帧缓存 4 寄存器	--
LCD_FB05	R/W	0x0014	LCD 帧缓存 5 寄存器	--
LCD_FB06	R/W	0x0018	LCD 帧缓存 6 寄存器	--
LCD_FB07	R/W	0x001C	LCD 帧缓存 7 寄存器	--
LCD_FB08	R/W	0x0020	LCD 帧缓存 8 寄存器	--
LCD_FB09	R/W	0x0024	LCD 帧缓存 9 寄存器	--
LCD_FB0A	R/W	0x0028	LCD 帧缓存 10 寄存器	--
LCD_FB0B	R/W	0x002C	LCD 帧缓存 11 寄存器	--
LCD_FB0C	R/W	0x0030	LCD 帧缓存 12 寄存器	--
LCD_FB0D	R/W	0x0034	LCD 帧缓存 13 寄存器	--
LCD_FB0E	R/W	0x0038	LCD 帧缓存 14 寄存器	--
LCD_FB0F	R/W	0x003C	LCD 帧缓存 15 寄存器	--
LCD_FB10	R/W	0x0040	LCD 帧缓存 16 寄存器	--
LCD_FB11	R/W	0x0044	LCD 帧缓存 17 寄存器	--
LCD_FB12	R/W	0x0048	LCD 帧缓存 18 寄存器	--
LCD_FB13	R/W	0x004C	LCD 帧缓存 19 寄存器	--
LCD_FB14	R/W	0x0050	LCD 帧缓存 20 寄存器	--
LCD_FB15	R/W	0x0054	LCD 帧缓存 21 寄存器	--
LCD_FB16	R/W	0x0058	LCD 帧缓存 22 寄存器	--
LCD_FB17	R/W	0x005C	LCD 帧缓存 23 寄存器	--
LCD_FB18	R/W	0x0060	LCD 帧缓存 24 寄存器	--
LCD_FB19	R/W	0x0064	LCD 帧缓存 25 寄存器	--
LCD_FB1A	R/W	0x0068	LCD 帧缓存 26 寄存器	--
LCD_FB1B	R/W	0x006C	LCD 帧缓存 27 寄存器	--
LCD_FB1C	R/W	0x0070	LCD 帧缓存 28 寄存器	--
LCD_FB1D	R/W	0x0074	LCD 帧缓存 29 寄存器	--
LCD_FB1E	R/W	0x0078	LCD 帧缓存 30 寄存器	--
LCD_FB1F	R/W	0x007C	LCD 帧缓存 31 寄存器	--
LCD_FB20	R/W	0x0080	LCD 帧缓存 32 寄存器	--
LCD_FB21	R/W	0x0084	LCD 帧缓存 33 寄存器	--
LCD_FB22	R/W	0x0088	LCD 帧缓存 34 寄存器	--
LCD_FB23	R/W	0x008C	LCD 帧缓存 35 寄存器	--
LCD_FB24	R/W	0x0090	LCD 帧缓存 36 寄存器	--
LCD_FB25	R/W	0x0094	LCD 帧缓存 37 寄存器	--

LCD_FB26	R/W	0x0098	LCD 帧缓存 38 寄存器	--
LCD_FB27	R/W	0x009C	LCD 帧缓存 39 寄存器	--
LCD_CTRL	R/W	0x0100	LCD 控制寄存器	0x00
LCD_CTRL2	R/W	0x0104	LCD 控制寄存器 2	0x0000
LCD_SEGCTRL0	R/W	0x0108	LCDSEG 使能控制寄存器 0	0x00000000
LCD_SEGCTRL1	R/W	0x010C	LCDSEG 使能控制寄存器 1	0x00000000
LCD_SEGCTRL2	R/W	0x0110	LCDSEG 使能控制寄存器 2	0x00000000

## 21.9 寄存器定义

### 21.9.1 LCD\_FBx 寄存器

表 21-3 LCD\_FBx 寄存器

位	名称	类型	描述	默认值
31: 0	DATAx	R/W	每个位表示显示阵列中的数据，不同模式下数据排列的详细信息，请参见表 21-4。这些寄存器可以进行字节读或写，因此用户可以使用 CPU 或 DMA 来读写这些寄存器。	--

LCD\_FB00~LCD\_FB13 是帧缓存器 A。

LCD\_FB14~LCD\_FB27 是帧缓存器 B。

下表仅显示帧缓存器 A，如果 FBMODE 被选择为 1，则帧缓存器 B 呈现与帧缓存器 A 相同的数据。

表 21-4 LCD\_FBx 寄存器的数据排布

寄存器	数据 bit			
	31: 24	23: 16	15: 8	7: 0
LCD_FB00	SEG3 对应的 COM[7: 0]	SEG2 对应的 COM[7: 0]	SEG1 对应的 COM[7: 0]	SEG0 对应的 COM[7: 0]
LCD_FB01	SEG7 对应的 COM[7: 0]	SEG6 对应的 COM[7: 0]	SEG5 对应的 COM[7: 0]	SEG4 对应的 COM[7: 0]
.....				
LCD_FB13	SEG79 对应的 COM[7: 0]	SEG78 对应的 COM[7: 0]	SEG77 对应的 COM[7: 0]	SEG76 对应的 COM[7: 0]

当选择 4/6 COM 模式时，COM 的高位将被丢弃，因此数据排布格式与 8COM 模式相同。

### 21.9.2 LCD\_CTRL 寄存器

表 21-5 LCD\_CTRL 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7	EN	R/W	LCD 使能寄存器。 0: 禁用 1: 启用	0x0

6	-	-	保留	0
5: 4	TYPE	R/W	LCD 模式控制寄存器。 0: 4 COM 模式 1: 6 COM 模式 2: 8 COM 模式 3: 保留	0x0
3: 2	DRV	R/W	LCD 驱动电阻控制寄存器。 0: 300k ohm 1: 600k ohm 2: 150k ohm 3: 200 kohm	0x0
1: 0	FRQ	R/W	LCD 扫描频率。 0: 64Hz 1: 128Hz 2: 256Hz 3: 512Hz  这里的扫描频率是指每个 COM 的扫描速度，因此如果 COM 总数为 4，则整个帧速率等于该值除以 4。	0x0

**注意：**切换 COM 类型时，首先关闭 LCD 控制器（LCD\_CTRL 寄存器的第 7 位），然后等待一个延时，以保证当前帧传输完成。延时与当前帧速率相关（LCD\_CTRL 寄存器的第 1: 0 位、第 7 位）。

例如：LCD 的当前扫描速率是 64Hz，将 COM 类型从 4COM 切换到 6COM，延时时间计算如下：

$$T_{\text{Delay}} = 1 / (64 / 4) = 62.5\text{ms}$$

### 21.9.3 LCD\_CTRL2 寄存器

表 21-6 LCD\_CTRL2 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 8	SWPR	R/W	帧缓存器转换周期。转换周期计算公式如下： $0.5 \text{ 秒} * (\text{SWPR} + 1)$	0x0
7: 6	FBMODE	R/W	LCD 帧缓存转换模式控制寄存器。 0: 始终显示帧缓存器 A 1: 在帧缓存器 A 和帧缓存器 B 之间切换 2: 在帧缓存器 A 和空白之间切换 3: 保留	0x0
5	-	-	保留	0
4	BKFILL	R/W	空白周期填充值。该寄存器控制空白周期内的填充值。	0x0
3: 0	-	-	保留	0

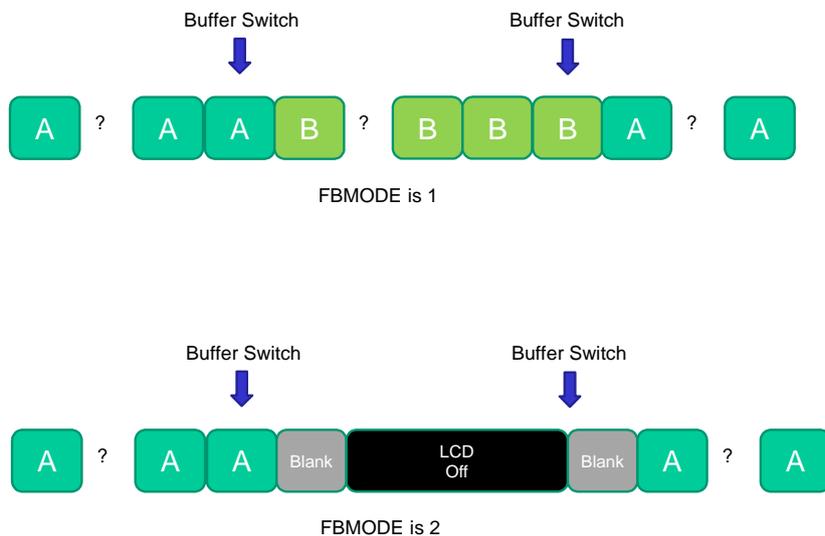


图 21-7 不同 FBMODE 下的帧缓存器操作模式

## 21.9.4 LCD\_SEGCTRL0 寄存器

表 21-7 LCD\_SEGCTRL0 寄存器

位	名称	类型	描述	默认值
31: 0	SEGCTRL	R/W	每个位控制 SEG0~SEG31 的 LCD 信号使能。 位 0: SEG0 的使能控制 位 1: SEG1 的使能控制 .... 位 31: SEG31 的使能控制 0: 禁用 SEG 输出 1: 使能 SEG 输出	0x00000000

## 21.9.5 LCD\_SEGCTRL1 寄存器

表 21-8 LCD\_SEGCTRL1 寄存器

位	名称	类型	描述	默认值
31: 0	SEGCTRL	R/W	每个位控制 SEG32~SEG63 的 LCD 信号使能。 位 0: SEG32 的使能控制 位 1: SEG33 的使能控制 .... 位 31: SEG63 的使能控制 0: 禁用 SEG 输出 1: 使能 SEG 输出	0x00000000

## 21.9.6 LCD\_SEGCTRL2 寄存器

表 21-9 LCD\_SEGCTRL2 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	SEGCTRL	R/W	每个位控制 SEG64~SEG79 的 LCD 信号使能。 位 0: SEG64 的使能控制 位 1: SEG65 的使能控制 ... 位 15: SEG79 的使能控制。 0: 禁用 SEG 的输出 1: 使能 SEG 输出	0x0000

## 21.9.7 ANA\_REG6 寄存器

表 21-10 ANA\_REG6 寄存器

位	名称	Function	Notes
0	LCD_BMODE	LCD 偏置模式选择。	0: 1/3 Bias 1: 1/4 Bias
4: 1	VLCD[3: 0]	LCD 驱动电压。	当 VLCD=0: Default 当 VLCD=0~6 : 调整幅度 = +50mV*VLCD 当 VLCD=7~15 : 调整幅度 = -50mV*(VLCD-6)

对于每个 COM 和 SEG，为了确保 LCD 正常工作，在使能相应引脚的 LCD 功能之前，用户应设置相应的 IO 禁止输入禁止输出。

## 第22章 SPI 控制器

### 22.1 简介

V85XXP 中集成了 3 个串行外设接口 (SPI) 控制器, 以便与其他设备或组件通信。SPI 控制器中的设置在从浅睡眠或深睡眠模式唤醒后会被复位, 用户应在浅睡眠/深睡眠模式唤醒后手动恢复设置。SPI 主机模式支持的 SPICLK 时钟: APBCLK/2/4/8/16/32/64/128。SPI 从机模式支持的最大 SPICLK 时钟为 APBCLK/2。

### 22.2 特点

- 支持主机模式和从机模式;
- 支持单字节和连续字节传输;
- 支持接收溢出错误标志;
- 支持发送/接收中断请求;
- 相位和极性可编程;
- 可选择的数据采样时间 (时钟周期的结束或中间);
- 主机模式下 CLK 时钟频率可编程: APBClock/2/4/8/16/32/64/128;
- 发送和接收方向上内置 8 个 8 位的 FIFO, 两种 FIFO 的中断触发阈值都可编程。

### 22.3 功能框图

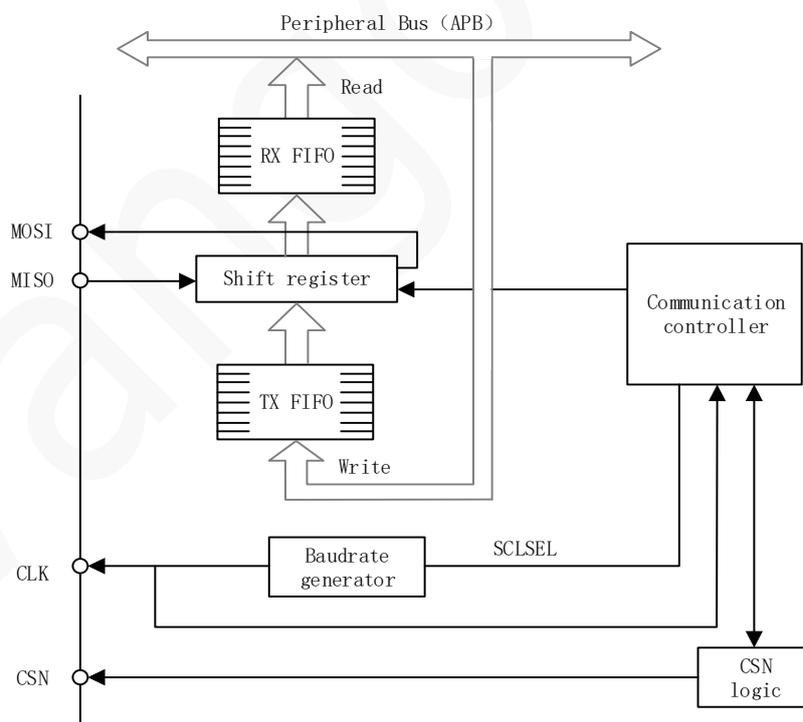


图 22-1 SPI 控制器的功能框图

通常 SPI 通过 4 个引脚与外部设备相连。

**MOSI:** 主设备输出/从设备输入。一般情况下, 该引脚用于主机模式下的数据发送和从机模式下的数据接收。

**MISO:** 主设备输入/从设备输出。一般情况下, 该引脚用于主机模式下的数据接收和从机模式下的数据发送。

**CLK:** SPI 串行时钟输出引脚, 时钟由主机控制。

**CSN:** 从机片选引脚。根据 SPI 和 CSN 的设置, 该引脚可用于: 选择一个从机通信、同步数据帧。

注意：SPI 从机模式支持的最大 SPICLK 时钟为 APBCLK/2。

## 22.4 寄存器地址

表 22-1 SPI1 控制器(SPI1 基地址: 0x40011000)

名称	类型	地址	描述	默认值
SPI1_CTRL	R/W	0x0000	SPI1 控制寄存器	0x0000
SPI1_TXSTS	R/W	0x0004	SPI1 发送状态寄存器	0x8200
SPI1_TXDAT	R/W	0x0008	SPI1 发送 FIFO 寄存器	--
SPI1_RXSTS	R/W	0x000C	SPI1 接收状态寄存器	0x0000
SPI1_RXDAT	R/W	0x0010	SPI1 接收 FIFO 寄存器	--
SPI1_MISC	R/W	0x0014	SPI1Misc 控制寄存器	0x0003

表 22-2 SPI2 控制器(SPI2 基地址: 0x40015800)

名称	类型	地址	描述	默认值
SPI2_CTRL	R/W	0x0000	SPI2 控制寄存器	0x0000
SPI2_TXSTS	R/W	0x0004	SPI2 发送状态寄存器	0x8200
SPI2_TXDAT	R/W	0x0008	SPI2 发送 FIFO 寄存器	--
SPI2_RXSTS	R/W	0x000C	SPI2 接收状态寄存器	0x0000
SPI2_RXDAT	R/W	0x0010	SPI2 接收 FIFO 寄存器	--
SPI2_MISC	R/W	0x0014	SPI2Misc 控制寄存器	0x0003

表 22-3 SPI3 控制器(SPI3 基地址: 0x40016000)

名称	类型	地址	描述	默认值
SPI3_CTRL	R/W	0x0000	SPI3 控制寄存器	0x0000
SPI3_TXSTS	R/W	0x0004	SPI3 发送状态寄存器	0x8200
SPI3_TXDAT	R/W	0x0008	SPI3 发送 FIFO 寄存器	--
SPI3_RXSTS	R/W	0x000C	SPI3 接收状态寄存器	0x0000
SPI3_RXDAT	R/W	0x0010	SPI3 接收 FIFO 寄存器	--
SPI3_MISC	R/W	0x0014	SPI3Misc 控制寄存器	0x0003

## 22.5 寄存器定义

### 22.5.1 SPIx\_CTRL 寄存器

SPIx\_CTRL 寄存器用来控制 SPIx 控制器的行为。

表 22-4 SPIx\_CTRL 寄存器

位	名称	类型	描述	默认值
15	EN	R/W	若该位置 1, SPICLK、SPIMISO、SPIMOSI 这些引脚将不能被用作普通 GPIO, 因此, 任何对 GPIO 的输入输出设置都会无效。	0x0

			<p>对于 SPI1 引擎, 如果该位设置为 1, IOB[12: 10]将成为 SPI1 接口。</p> <p>对于 SPI2 引擎, 如果该位设置为 1, IOC[3: 1]将成为 SPI2 接口。</p> <p>对于 SPI3 引擎, 如果该位设置为 1, IOD[7: 5]将成为 SPI3 接口。</p> <p>SPICSN 管脚由 CSGPIO 控制。</p> <p>0: 禁用 SPI 功能 1: 启用 SPI 功能</p>	
14: 13	-	-	保留	0
12	LSBF	R/W	<p>SPI LSB 控制</p> <p>数据位传输顺序控制寄存器。</p> <p>0: 高位先传 (MSB)。 1: 低位先传 (LSB)。</p>	
11	RST	W	<p>SPI 软件复位。如果该位写 1, SPI 控制器和 FIFO 指针的状态会恢复到初始状态。</p> <p>0: 无效 1: 复位 SPI 控制器</p>	0x0
10	CSGPIO	R/W	<p>SPICSN 管脚由 GPIO 或 H/W 控制。</p> <p>主控模式:</p> <p>0: SPICSN 管脚由硬件 H/W 控制 1: SPICSN 管脚由软件 GPIO 控制</p> <p>从机模式:</p> <p>0: SPICSN 管脚由外部主机控制 1: SPICSN 管脚为逻辑 0</p>	0x0
9	SWAP	R/W	<p>SPI MISO/MOSI 引脚互换控制寄存器。</p> <p>0: 不交换 MISO/MOSI 1: 交换 MISO/MOSI</p>	0x0
8	MOD	R/W	<p>SPI 模式选择寄存器。</p> <p>0: 主机模式 1: 从机模式</p>	0x0
7: 6	-	-	保留	0
5	SCKPHA	R/W	SPI 时钟相位。请参考应用笔记。	0x0
4	SCKPOL	R/W	SPI 时钟极性。请参考应用笔记。	0x0
3	-	-	保留	0
2: 0	SCKSEL	R/W	<p>主机模式时钟选择。</p> <p>000: APBCLK/2 001: APBCLK/4 010: APBCLK/8</p>	0x0

			011: APBCLK/16 100: APBCLK/32 101: APBCLK/64 110: APBCLK/128 111: 保留	
--	--	--	--	--

## 22.5.2 SPIx\_TXSTS 寄存器

SPIx\_TXSTS 寄存器控制 SPIx 发送 FIFO 和相关中断。

表 22-5 SPIx\_TXSTS 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15	TXIF	R/C	SPI 发送中断标志。 当发送 FIFO 中数据数量低于用户设置的值时，该位由硬件置 1。如果 SPI_MISC 寄存器中的 SMART 值 1，只要发送 FIFO 中数据数量高于用户设置的值，该位就会被清零，否则应该对该位写 1 以清除该标志。 读 0: 未发生 读 1: 发生 写 0: 无效 写 1: 清除标志	0x1
14	TXIEN	R/W	SPI 发送中断使能。 如果该位置为 1，且 SPI 中断发生时，硬件将向 CPU 发出一个中断。如果该位清零，该中断将被屏蔽。 0: 禁止发送中断 1: 使能发送中断	0x0
13: 10	-	-	保留	0
9	TXEMPTY	R	发送 FIFO 空标志。 当发送 FIFO 为空时，该位由硬件置 1。当发送 FIFO 不为空时，该位由硬件清除。 0: 发送 FIFO 不为空 1: 发送 FIFO 为空	0x1
8	TXFUR	R	运行中从机发送 FIFO 空标志位。 当发送 FIFO 为空且外部 SPI 主机请求更多数据时，该位将被置 1，该位只在 SPI 从机模式下会置 1。	0x0
7	-	-	保留	0
6: 4	TXFLEV	R/W	发送 FIFO 的中断触发阈值。 该寄存器用于指示在发生中断时，发送 FIFO 中存储的字节数。设置值越小，中断触发越少，因此可以在一个中断中写入更多数据。 000: FIFO 中的数据 < 1，允许写入 8 个数据。 001: FIFO 中的数据 < 2，允许写入 7 个数据。	0x0

			<p>010: FIFO 中的数据&lt;3, 允许写入 6 个数据。</p> <p>011: FIFO 中的数据&lt;4, 允许写入 5 个数据。</p> <p>100: FIFO 中的数据&lt;5, 允许写入 4 个数据。</p> <p>101: FIFO 中的数据&lt;6, 允许写入 3 个数据。</p> <p>110: FIFO 中的数据&lt;7, 允许写入 2 个数据。</p> <p>111: FIFO 中的数据&lt;8, 允许写入 1 个数据。</p>	
3	DMATXDONE	R/C	发送完成标志。当通过 DMA 方式批量数据发送时, 会在全部发送完成时自动置 1。用户必须等到 DMA 发送完成后才能读取此标志。写 1 清零该位。	0x0
2: 0	TXFFLAG	R	<p>发送 FIFO 中数据个数。</p> <p>该寄存器用于指示 FIFO 中的数据数量。</p> <p>0000: FIFO 中没有数据或 FIFO 中有 8 个字节</p> <p>0001: FIFO 中有 1 个字节</p> <p>0010: FIFO 中有 2 个字节</p> <p>0011: FIFO 中有 3 个字节</p> <p>0100: FIFO 中有 4 个字节</p> <p>0101: FIFO 中有 5 个字节</p> <p>0110: FIFO 中有 6 个字节</p> <p>0111: FIFO 中有 7 个字节</p>	0x0

### 22.5.3 SPIx\_TXDAT 寄存器

SPI\_TXDAT 寄存器用于 SPI 发送机将数据写入到 SPI 的发送 FIFO, 并移出到外部主机或从机。

表 22-6 SPIx\_TXDAT 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7: 0	TXD	W	向 SPI 发送 FIFO 写数据。	--

### 22.5.4 SPIx\_RXSTS 寄存器

SPI\_RXSTS 寄存器用于控制 SPI 接收 FIFO 及相关中断。

表 22-7 SPIx\_RXSTS 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15	RXIF	R/C	<p>SPI 接收中断标志。</p> <p>当接收 FIFO 中数据个数高于用户设定的值时, 该位由硬件置 1。</p> <p>如果 SPI_MISC 寄存器中的 SMART 值设置为 1, 只要接收 FIFO 中数据个数低于中断触发阈值, 该位就会被清零; 如果 SMART 值设置为 0, 应该对该位写 1 以清除该标志。</p> <p>读 0: 未发生</p> <p>读 1: 发生</p>	0x0

			写 0: 无效 写 1: 清除标志	
14	RXIEN	R/W	SPI 接收中断使能  如果该位设置为 1, 且 SPI 中断发生时, 硬件将向 CPU 发出一个中断。如果该位清零, 该中断将被屏蔽。  0: 禁止接收中断 1: 使能接收中断	0x0
13: 10	-	-	保留	0
9	RXFULL	R	接收 FIFO 满标志。  当接收 FIFO 已满时, 该位由硬件置 1。当接收 FIFO 未滿时, 硬件将清零该位。  0: 接收 FIFO 不满 1: 接收 FIFO 满	0x0
8	RXFOV	R	接收 FIFO 溢出寄存器。  接收 FIFO 已满并且从 SPI 数据总线上接收到更多数据时, 该位将置 1。向 SPIRXIF 位写 1 清零该位。	0x0
7	-	-	保留	0
6: 4	RXFLEV	R/W	接收 FIFO 的中断触发阈值。  该寄存器表示在发生中断时接收 FIFO 中有多少字节的数据。设置值越大, 中断触发越少, 因此可以在一个中断中读取更多数据。  000: FIFO 中的数据 ≥ 1, 允许 1 次数据读取 001: FIFO 中的数据 ≥ 2, 允许 2 次数据读取 010: FIFO 中的数据 ≥ 3, 允许 3 次数据读取 011: FIFO 中的数据 ≥ 4, 允许 4 次数据读取 100: FIFO 中的数据 ≥ 5, 允许 5 次数据读取 101: FIFO 中的数据 ≥ 6, 允许 6 次数据读取 110: FIFO 中的数据 ≥ 7, 允许 7 次数据读取 111: FIFO 中的数据 ≥ 8, 允许 8 次数据读取	0x0
3	-	-	保留	0
2: 0	RXFFLAG	R	接收 FIFO 数据个数。  该寄存器表示接收 FIFO 的数据数量。  0000: FIFO 中没有数据或 FIFO 中有 8 个字节 0001: FIFO 中有 1 个字节 0010: FIFO 中有 2 个字节 0011: FIFO 中有 3 个字节 0100: FIFO 中有 4 个字节 0101: FIFO 中有 5 个字节	0x0

			0110: FIFO 中有 6 个字节	
			0111: FIFO 中有 7 个字节	

## 22.5.5 SPIx\_RXDAT 寄存器

SPI\_RXDAT 寄存器用于 SPI 接收机从接收 FIFO 中读取数据。

表 22-8 SPIx\_RXDAT 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7: 0	RXD	R	从 SPI 接收 FIFO 中读取数据。	--

## 22.5.6 SPIx\_MISC 寄存器

SPI\_MISC 寄存器用于配置 SPI 接收发送特殊功能。

表 22-9 SPIx\_MISC 寄存器

位	名称	类型	描述	默认值
31: 10	-	-	保留	0
9	OVER	R/W	SPI FIFO 溢出控制。 该寄存器用于控制当 TX/RXFIFO 已满时，数据会被重写或跳过。 0: 将跳过对已满 FIFO 的进一步写操作。 1: 写入已满 FIFO，会覆盖 FIFO 中最先写入的数据。	0x0
8	SMART	R/W	SPI FIFO SMART 模式寄存器。 如果该位设置为 1，用户在 FIFO 状态到达中断等级时不需要手动清除发送/接收中断标志，用户只需向发送 FIFO 写入数据或从接收 FIFO 读取数据，中断标志将自动清零。当选择 DMA 模式时，必须将该位设置为 1，以保证 DMA 操作的正确。 0: 正常中断清零 1: 智能中断清零	0x0
7: 5	-	-	保留	0
4	BSY	R	SPI 控制器忙标志。 该位用于指示 SPI 控制器是否忙。 0: 空闲 1: 忙	0x0
3	RFF	R	接收 FIFO 满标志。 该位指示 SPI 控制器接收 FIFO 是否为满。 0: 接收 FIFO 未满 1: 接收 FIFO 已满	0x0
2	RNE	R	接收 FIFO 非空标志。 该位显示接收 FIFO 是否有数据。 0: 接收 FIFO 为空	0x0

			1: 接收 FIFO 非空	
1	TNF	R	发送 FIFO 未空标志。 该位显示发送 FIFO 是否未空。 0: 发送 FIFO 满 1: 发送 FIFO 未空	0x1
0	TFE	R	发送 FIFO 为空标志 该位显示发送 FIFO 是否为空。 0: 发送 FIFO 非空 1: 发送 FIFO 空	0x1

## 22.6 应用笔记

### 22.6.1 主机模式

主机模式下，移位时钟（SPICLK）由 V85XXP 产生。SCKPHA 和 SCKPOL 两个控制位分别控制时钟相位和极性。向 SPI\_TXDAT 寄存器写入一个数据即可开始数据传输。只要 FIFO 中有数据，在发送完一字节数据后会自动发送下一个数据。

SPI 发送数据时，根据 LSBF 控制位将数据从 MSB 移到 LSB 或从 LSB 移到 MSB。移出 8 位数据需要 8 个 SPICLK 周期。同时，数据也通过从器件输出引脚移入。当发送 FIFO 等级低于中断触发等级时，SPITXIF 标志位将被置 1。同时，如果 SPITXIEN 位置 1，将产生 SPI 发送中断。当接收 FIFO 等级高于中断触发等级时，SPIRXIF 标志位将被置 1。同时，如果 SPIRXIEN 位置 1，将产生 SPI 接收中断。从 SPI\_RXDAT 寄存器可以读取 SPI 数据。

下图展示 SPI 主机模式下不同操作类型配置下的不同时序方案。（极性控制位等于 1 或 0，相位控制位等于 1 或 0）。从机模式同样支持极性控制和相位控制。

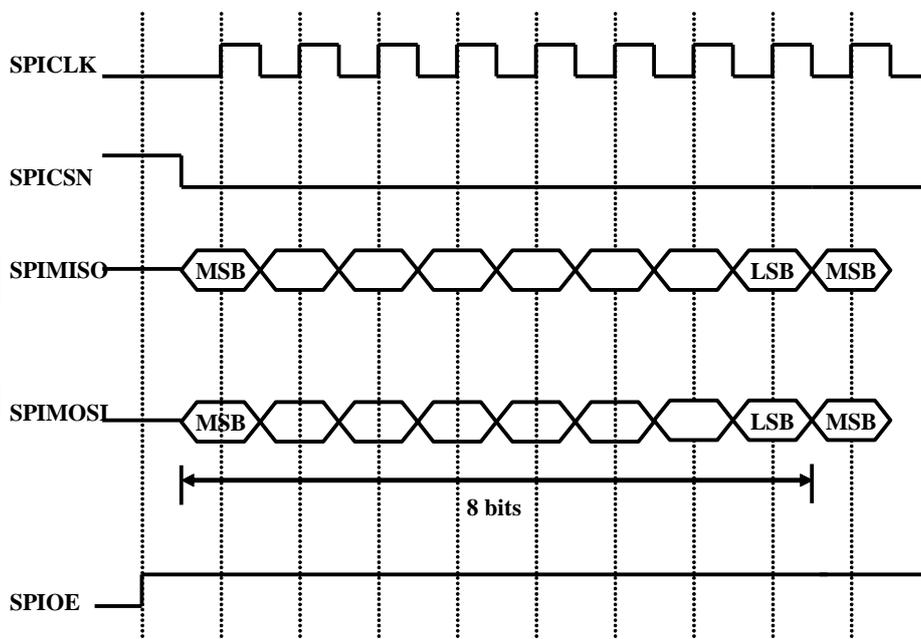


图 22-2 主机模式，SPO=0，SPH=0

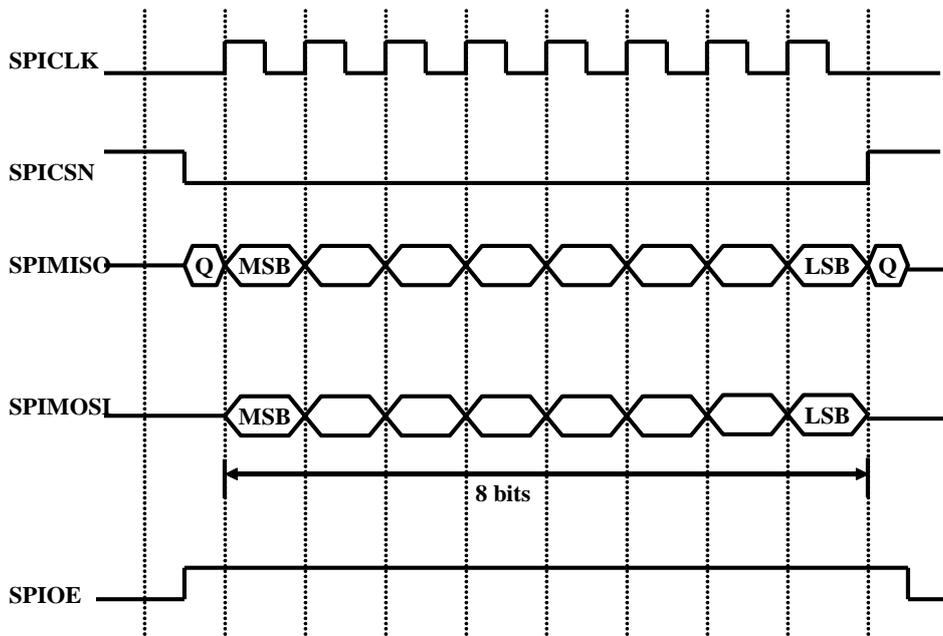


图 22-3 主机模式, SPO=0, SPH=1

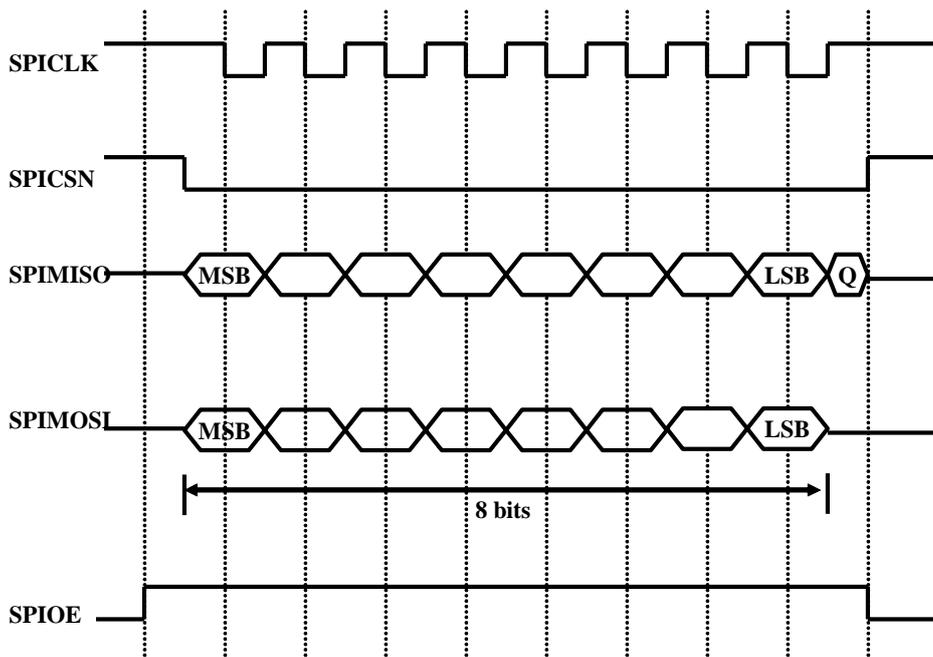


图 22-4 主机模式, SPO=1, SPH=0

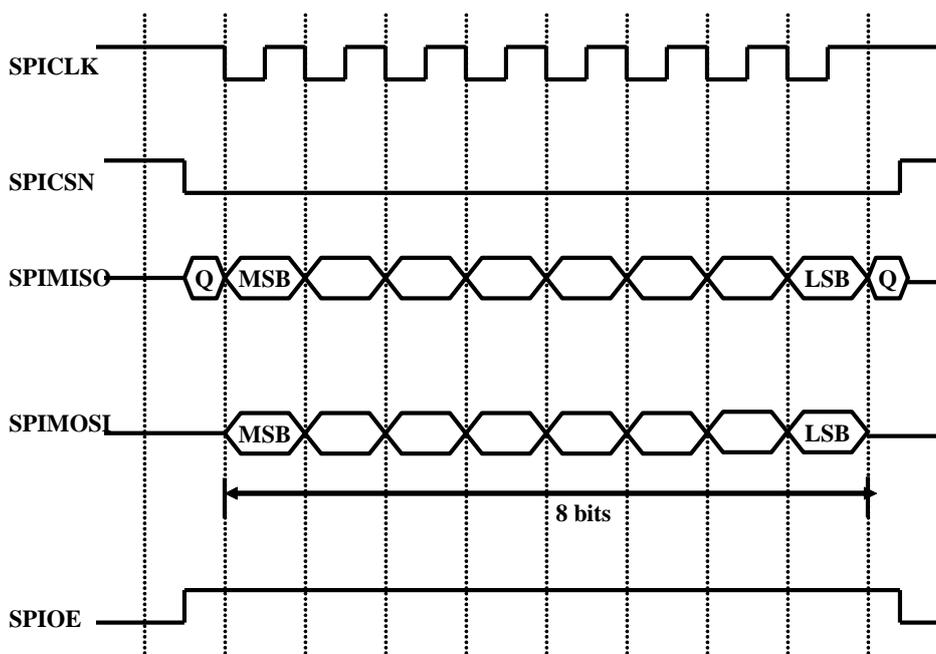


图 22-5 主机模式，SPO=1，SPH=1

## 22.6.2 从机模式

SPI 从机模式时，SPICLK 和 SPICSN 由外部主机产生。SPI 从机模式支持时钟极性和相位配置。SPI\_TXDAT 和 SPI\_RXDAT 寄存器可以用于数据发送或接收。SPI 从机模式支持的最大 SPICLK 时钟为 APBCLK/2。当接收时钟速率较快时，建议使用 DMA 传输。

## 22.6.3 连续字节传输

连续字节传输在主机模式或从机模式都可用。发送数据时，只要 TNF 位为 1，软件就能够连续发送数据。接收数据时，软件需要检查溢出标志，以监测是否由于轮询速率太低而丢失数据。

## 第23章 I<sup>2</sup>C 控制器

### 23.1 简介

I<sup>2</sup>C 控制器提供了一个满足 Philips I<sup>2</sup>C 总线规范的接口，并支持双向传输。I<sup>2</sup>C 总线使用两条线在连接到总线的设备之间传输信息：SCL（串行时钟线）和 SDA（串行数据线）。I<sup>2</sup>C 支持单字节自动传输。状态寄存器反映 I<sup>2</sup>C 总线控制器和 I<sup>2</sup>C 总线的状态。I<sup>2</sup>C 控制器中的设置在从浅睡眠或深睡眠模式唤醒后会被复位，用户应在浅睡眠/深睡眠模式唤醒后手动恢复设置。I<sup>2</sup>C 总线最大支持频率  $f_{SCL}=1\text{MHz}$ （周期：1 $\mu\text{s}$ ），可以通过设置定时器 3 溢出频率实现。

### 23.2 特点

- 支持主机发送、主机接收模式；
- 支持从机发送、从机接收模式；
- 总线仲裁失败监测；
- 支持中断信号；
- 应答（ACK）产生可编程；
- 主机模式时钟速率可编程；
- 输入端具有消抖电路。

### 23.3 功能框图

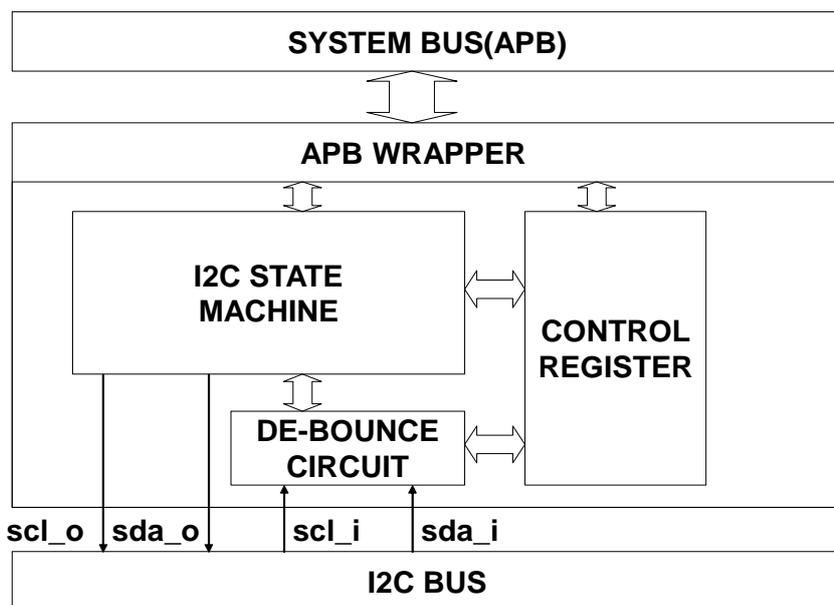


图 23-1 I<sup>2</sup>C 控制器功能框图

### 23.4 寄存器地址

表 23-1 I<sup>2</sup>C 控制器(I<sup>2</sup>C 基地址：0x40010800)

名称	类型	地址	描述	默认值
I2C_DATA	R/W	0x0000	I <sup>2</sup> C 数据寄存器	0x00
I2C_ADDR	R/W	0x0004	I <sup>2</sup> C 地址寄存器	0x00

I2C_CTRL	R/W	0x0008	I <sup>2</sup> C 控制/状态寄存器	0x00
I2C_STS	R/W	0x000c	I <sup>2</sup> C 状态寄存器	0xF8
I2C_CTRL2	R/W	0x0018	I <sup>2</sup> C 中断使能寄存器	0x0

## 23.5 寄存器定义

### 23.5.1 I2C\_DATA 寄存器

表 23-2 I2C\_DATA 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7: 0	DATA	R/W	I2C_DATA 寄存器包含一个要通过 I <sup>2</sup> C 总线传输的字节或刚通过 I <sup>2</sup> C 总线接收的字节。该 8 位寄存器不处于字节移位时, CPU 可以对其进行读写操作。I2C_DATA 寄存器没有影子寄存器或双缓存, 因此用户只能在发生 I <sup>2</sup> C 中断时读取 I2C_DATA。	0x00

### 23.5.2 I2C\_ADDR 寄存器

表 23-3 I2C\_ADDR 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7: 1	SLA	R/W	自有 I <sup>2</sup> C 从地址(7 位)	0x00
0	GC	R/W	广播呼叫地址响应。 如果该位置 1, 则识别广播呼叫地址; 否则会忽略广播呼叫地址。	0x0

### 23.5.3 I2C\_CTRL 寄存器

表 23-4 I2C\_CTRL 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7	CR2	R/W	时钟速率控制位 2	0x0
6	EN	R/W	I <sup>2</sup> C 使能位 当 EN=0, 会将“sdao”和“sclo”输出置为 1, 芯片输出处于高阻状态。此时, “sdai”和“scli”引脚的输入信号会被忽略。 当 EN=1, I <sup>2</sup> C 组件被使能, 相应 IO 将自动设置为 I <sup>2</sup> C 功能。	0x0
5	STA	R/W	START 标志 当 STA=1 时, I <sup>2</sup> C 组件检查 I <sup>2</sup> C 总线状态, 如果总线空闲, 则产生 START 状态。	0x0
4	STO	R/W	STOP 标志 当 STO=1 且 I <sup>2</sup> C 接口处于主机模式时, 会向 I <sup>2</sup> C 总线发送一个 STOP 状态。	0x0
3	SI	R/C	串行中断标志 I <sup>2</sup> C 输入状态为 26 种可能输入的 25 种 (F8h 除外) 时, “SI”由硬件置 1。F8h 是唯一不置位“SI”的状态, 其表示没有相关的可用状态信息。“SI”标志必须由软件清零。为了清除“SI”位, 必须向该位写 0。向该位写 1 不会改变“SI”的值。当发生中断时, 清除 SI 表示下一个动作的触发, 因此在清除 SI 之前, 必须设置好必需的控制寄存器或数据。	0x0

2	AA	R/W	有效应答标志 当 AA=1 时, 将在以下情况下返回“应答”: -已接收到“自有从地址” -在 I <sup>2</sup> C 地址寄存器中的 GC 位置 1 时接收到广播呼叫地址 -当 I <sup>2</sup> C 处于主接收器模式时, 已接收到数据字节 -当 I <sup>2</sup> C 处于从接收器模式时, 已接收到数据字节  当 AA='0'时, 将在以下情况下返回“不应答”: -当 I <sup>2</sup> C 处于主接收器模式时, 已接收到数据字节 -当 I <sup>2</sup> C 处于从接收器模式时, 已接收到数据字节	0x0
1	CR1	R/W	时钟速率控制位 1	0x0
0	CR0	R/W	时钟速率控制位 0	0x0

CR2~CR0 用于控制 I<sup>2</sup>C 作为主机的 SCL 时钟频率。

表 23-5 SCL 频率设置

CR2	CR1	CR0	SCL 频率(KHz)				时钟分频数
			PCLK= 6.5536MHz	PCLK= 13.1072MHz	PCLK= 19.6608MHz	PCLK= 26.2144 MHz	
0	0	0	25.6	51.2	76.8	102.4	256
0	0	1	29.2	58.4	87.6	116.8	224
0	1	0	34.1	68.2	102.3	136.4	192
0	1	1	40.9	81.9	122.9	163.8	160
1	0	0	6.8	13.7	20.5	27.3	960
1	0	1	54.6	109.2	163.8	218.4	120
1	1	0	109.2	218.4	327.7	436.9	60
1	1	1	TIMER3 溢出频率/8				

### 23.5.4 I2C\_STS 寄存器

表 23-6 I2C\_STS 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7: 3	STS	R	I <sup>2</sup> C 状态码	0x1F
2: 0		R	保留	0

表 23-7 主机发送模式下的 I<sup>2</sup>C 状态

STS	I <sup>2</sup> C 状态	应用软件响应				I <sup>2</sup> C 的下一步动作	
		读/写	I2C_CTRL				
			I2C_DATA	STA	STO		SI
0x08	START 状态已发送。	写 SLA+W*	X	0	0	X	SLA+W 将发送, ACK 将被接收。

0x10	重复 START 状态发送。	写 SLA+W	X	0	0	X	SLA+W 将发送, ACK 将被接收。
		写 SLA+R*	X	0	0	X	SLA+R 将发送, I <sup>2</sup> C 将会切换到主机接收模式。
0x18	SLA+W 已发送, ACK 已接收。	写数据字节	0	0	0	X	数据字节将发送, ACK 将被接收。
		无操作	1	0	0	X	重复 START 状态将发送。
		无操作	0	1	0	X	STOP 状态将发送, STO 标志将复位。
		无操作	1	1	0	X	STOP 状态发送后, 紧接着发送一个 START 状态, STO 标志会复位。
0x20	SLA+W 已发送, ACK 未接收到。	写数据字节	0	0	0	X	将传输数据字节, ACK 将被接收。
		无操作	1	0	0	X	重复 START 将发送。
		无操作	0	1	0	X	STOP 状态将发送, STO 标志将复位。
		无操作	1	1	0	X	STOP 状态发送后, 紧接着发送一个 START 状态, STO 标志会复位。
0x28	I <sup>2</sup> C 数据字节已发送, ACK 已接收。	写数据字节	0	0	0	X	数据字节将发送, ACK 将被接收。
		无操作	1	0	0	X	重复 START 会发送。
		无操作	0	1	0	X	STOP 状态将发送, STO 标志将复位。
		无操作	1	1	0	X	STOP 状态发送后, 紧接着发送一个 START 状态, STO 标志会复位。
0x30	I <sup>2</sup> C 数据字节已发送, ACK 未接收。	写数据字节*	0	0	0	X	数据字节将发送, ACK 位将被接收。
		无操作	1	0	0	X	重复 START 状态会发送。
		无操作	0	1	0	X	STOP 状态将发送, STO 标志将复位。
		无操作	1	1	0	X	STOP 状态发送后, 紧接着发送一个 START 状态, STO 标志将复位。

\*: SLA+W: Slave 地址[7: 1]+写操作控制位[0], 0 表示写。

\*: SLA+R: Slave 地址[7: 1]+读操作控制位[0], 1 表示读。

表 23-8 主机接收模式下的 I<sup>2</sup>C 状态

STS	I <sup>2</sup> C 状态	应用软件响应					I <sup>2</sup> C 的下一步动作
		读/写 I2C_DATA	I2C_CTRL				
			STA	STO	SI	AA	
0x08	START 状态已发送。	写 SLA+R	X	0	0	X	将发送 SLA+R, 将接收 ACK。
0x10	重复 START 状态发送。	写 SLA+R	X	0	0	X	将发送 SLA+R, 将接收 ACK。
		写 SLA+W	X	0	0	X	将发送 SLA+W, I <sup>2</sup> C 将切换为主机发送模式。
0x40	SLA+R 已发送, ACK 已接收。	无操作	0	0	0	0	将接收数据字节, 无 ACK 返回。
		无操作	0	0	0	1	将接收数据字节, 将返回 ACK。
0x48		无操作	1	0	0	X	重复 START 状态将发送。
		无操作	0	1	0	X	STOP 状态将发送, STO 标志将重置。

	SLA+R 已发送, ACK 未接收。	无操作	1	1	0	X	STOP 状态发送后, 紧接着发送一个 START 状态, STOP 标志将重置。
0x50	I <sup>2</sup> C 数据的数字字节已接收, 已返回 ACK。	读数据字节	0	0	0	0	将接收数据字节, 无 ACK 返回。
		读数据字节	0	0	0	1	将接收数据字节, 返回 ACK。
0x58	I <sup>2</sup> C 数据的数字字节已发送, 未返回 ACK。	读数据字节	1	0	0	X	重复 START 状态将发送。
		读数据字节	0	1	0	X	STOP 状态将发送, STO 标志将重置。
		读数据字节	1	1	0	X	STOP 状态发送后, 紧接着发送一个 START 状态, STOP 标志将重置。

表 23-9 从机接收模式下的 I<sup>2</sup>C 状态

STS	I <sup>2</sup> C 状态	应用软件响应					I <sup>2</sup> C 的下一步动作
		读/写 I <sup>2</sup> C_DATA	I <sup>2</sup> C_CTRL				
			STA	STO	SI	AA	
0x60	接收到自有 SLA+W, 已返回 ACK。	无操作	X	0	0	0	将接收数据字节, 并且不返回 ACK。
		无操作	X	0	0	1	将接收数据字节, 并且返回 ACK。
0x68	仲裁在 SLA+R/W 作为主机时丢失。 已接收自有 SLA+W, ACK 已返回。	无操作	X	0	0	0	将接收数据字节, 并且不返回 ACK。
		无操作	X	0	0	1	将接收数据字节, 并且返回 ACK。
0x70	已接收到广播地址 (00H), ACK 已返回。	无操作	X	0	0	0	将接收数据字节, 并且不返回 ACK。
		无操作	X	0	0	1	将接收数据字节, 并且返回 ACK。
0x78	仲裁在 SLA+R/W 作为主机时丢失。 已接收到广播地址 (00H), ACK 已返回。	无操作	X	0	0	0	将接收数据字节, 并且不返回 ACK。
		无操作	X	0	0	1	将接收数据字节, 并且返回 ACK。
0x80	之前在使用自己的 SLV 地址寻址。DATA 已收到。ACK 返回。	读数据字节	X	0	0	0	将接收数据字节, 并且不返回 ACK。
		读数据字节	X	0	0	1	将接收数据字节, 并且返回 ACK。
0x88	之前在使用自己的 SLA 地址寻址。DATA 字节已收到。无 ACK 返回。	读数据字节	0	0	0	0	切换为无寻址 SLV 模式。没有识别到自有 SLA 或广播地址
		读数据字节	0	0	0	1	切换为无寻址 SLV 模式。将识别到自有 SLA 或广播地址。
		读数据字节	1	0	0	0	切换为无寻址 SLV 模式。未识别到自有 SLA 或广播地址。START 状态将在总线空闲时发送。
		读数据字节	1	0	0	1	切换为无寻址 SLV 模式。将识别到自有 SLA 或广播地址。START 状态将在总线空闲时发送。

0x90	之前在使用广播地址寻址。DATA 已收到。ACK 已返回。	读数据字节	X	0	0	0	将收到数据字节，无 ACK 返回。
		读数据字节	X	0	0	1	将接收数据字节，并且返回 ACK。
0x98	之前在使用广播呼叫地址寻址。DATA 已收到。无 ACK 返回。	读数据字节	0	0	0	0	切换为无寻址 SLV 模式。未识别到自有 SLA 或广播地址。
		读数据字节	0	0	0	1	切换为无寻址 SLV 模式。未识别到自有 SLA 或广播地址。
		读数据字节	1	0	0	0	切换为无寻址 SLV 模式。未识别到自有 SLA 或广播地址。START 状态将在总线空闲时发送。
		读数据字节	1	0	0	1	切换为无寻址 SLV 模式。将识别到自有 SLA 或广播地址。START 状态将在总线空闲时发送。
0xA0	已接收到 STOP 状态或重复 START 状态，但仍以从接收或从发送模式寻址。	无操作	0	0	0	0	切换为无寻址 SLV 模式。未识别到自有 SLA 或广播地址。
		无操作	0	0	0	1	切换为无寻址 SLV 模式。将识别到自有 SLA 或广播地址。
		无操作	1	0	0	0	切换为无寻址 SLV 模式。未识别到自有 SLA 或广播地址。START 状态将在总线空闲时发送。
		无操作	1	0	0	1	切换为无寻址 SLV 模式。将识别到自有 SLA 或广播地址。START 状态将在总线空闲时发送。

表 23-10 从机发送模式下的 I<sup>2</sup>C 状态

STS	I <sup>2</sup> C	应用软件响应					I <sup>2</sup> C 的下一步动作
		读/写	I2C_CTRL				
			I2C_DATA	STA	STO	SI	
0xA8	已接收自有 SLA+R，已返回 ACK。	写数据字节	X	0	0	0	将发送上一个数据字节，并接收 ACK。
		写数据字节	X	0	0	1	将发送数据字节，并接收 ACK。
0xB0	仲裁在 SLA+R/W 作为主机时丢失。已接收自有 SLA+R，ACK 已返回。	写数据字节	X	0	0	0	将发送上一个数据字节，并接收 ACK。
		写数据字节	X	0	0	1	将发送数据字节，并接收 ACK。
0xB8	数据字节已发送，已接收 ACK。	写数据字节	X	0	0	0	将发送上一个数据字节，并接收 ACK。
		写数据字节	X	0	0	1	将发送数据字节，并接收 ACK。
0xC0	数据字节已发送，未接收 ACK。	无操作	0	0	0	0	切换为无寻址 SLV 模式。未识别到自有 SLA 或广播地址。
		无操作	0	0	0	1	切换为无寻址 SLV 模式。将识别到自有 SLA 或广播地址。

		无操作	1	0	0	0	切换为无寻址 SLV 模式。未识别到自有 SLA 或广播地址。START 状态将在总线空闲时发送。
		无操作	1	0	0	1	切换为无寻址 SLV 模式。将识别到自有 SLA 或广播地址。START 状态将在总线空闲时发送。
0xC8	上一个数据字节已发送，并已接收 ACK。	无操作	0	0	0	0	切换为无寻址 SLV 模式。未识别到自有 SLA 或广播地址。
		无操作	0	0	0	1	切换为无寻址 SLV 模式。将识别到自有 SLA 或广播地址。
		无操作	1	0	0	0	切换为无寻址 SLV 模式。未识别到自有 SLA 或广播地址。START 状态将在总线空闲时发送。
		无操作	1	0	0	1	切换为无寻址 SLV 模式。将识别到自有 SLA 或广播地址。START 状态将在总线空闲时发送。

表 23-11 I<sup>2</sup>C 状态

STS	I <sup>2</sup> C 状态	应用软件响应					I <sup>2</sup> C 硬件下一步动作
		读/写 I2C_DATA	I2C_CTRL				
			STA	STO	SI	AA	
0x38	已接收自有 SLA+R, ACK 已返回。	写数据字节	X	0	0	0	将发送上一个数据字节，并接收 ACK。
		写数据字节	X	0	0	1	将发送数据字节，并接收 ACK。
0xF8	仲裁在 SLA+R/W 作为主机时丢失。已接收自有 SLA+R, ACK 已返回。	写数据字节	X	0	0	0	将发送上一个数据字节，并接收 ACK。
0x00	数据字节已发送。ACK 已接收。	写数据字节	X	0	0	0	将发送上一个数据字节，并接收 ACK。

### 23.5.5 I2C\_CTRL2 寄存器

表 23-12 I2C\_CTRL2 寄存器

位	名称	类型	描述	默认值
31: 1	-	-	保留	0
0	INTEN	R/W	I <sup>2</sup> C 控制器的中断使能控制。 0: 禁止。 1: 使能。	0x0

Vangotech

## 23.6 应用笔记

### 23.6.1 I<sup>2</sup>C 总线协议：启动/停止条件

启动状态可以通过 SDA 线传输一个字节的串行数据，停止状态可以终止数据传输。停止状态是 SCL 为高电平时 SDA 线的低电平向高电平跳变。开始和停止状态始终由主器件生成。当产生启动状态时，I<sup>2</sup>C 总线忙。停止状态后几个时钟，I<sup>2</sup>C 总线将再次被释放。

当主器件发起启动状态时，它应发送从器件地址以确认从器件。一字节的地址包含 7 位地址和 1 位传输方向指示符（即，写或读）组成。如果第 8 位为 0，则表示写操作（发送操作）；如果第 8 位为 1，则表示数据读取请求（接收操作）。

主机通过发送停止状态结束传输操作。如果主机想继续向 I<sup>2</sup>C 总线传输数据，则它应该重新发送一个启动状态和从机地址。通过这种方式，可以进行各种格式的读/写操作。

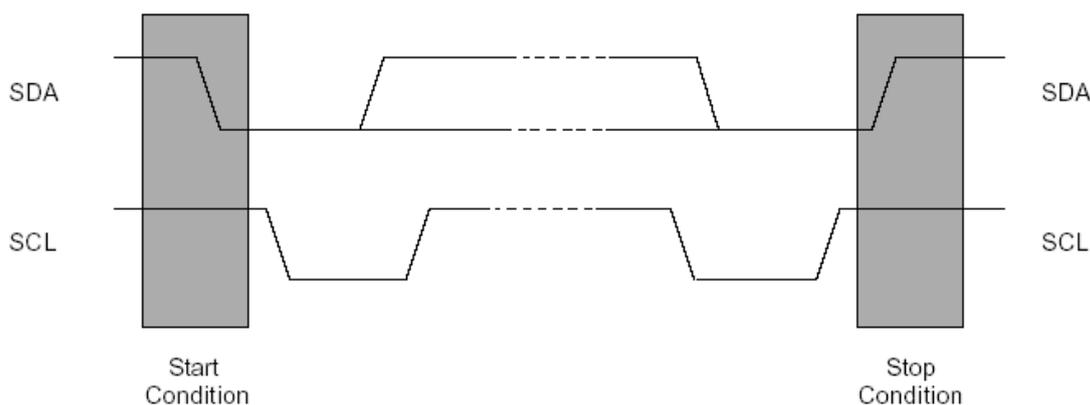


图 23-2 启动和停止条件

### 23.6.2 数据传输格式

位于 SDA 线上的每个字节的长度应为 8 位。每次允许传输的字节数量没有限制。启动状态后的第一个字节应该为地址字段。当 I<sup>2</sup>C 总线工作在主模式时，地址字段可以由主机发送。每个字节后面应跟一个确认（ACK）位。串行数据和地址总是高位（MSB）先发。

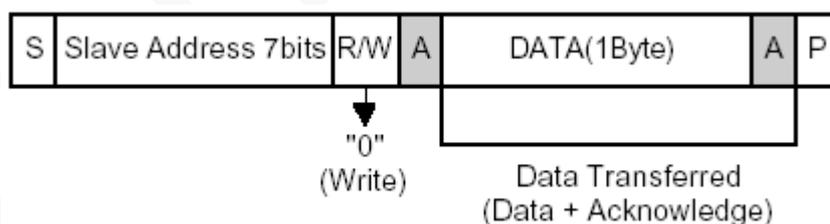


图 23-3 7 位地址写模式

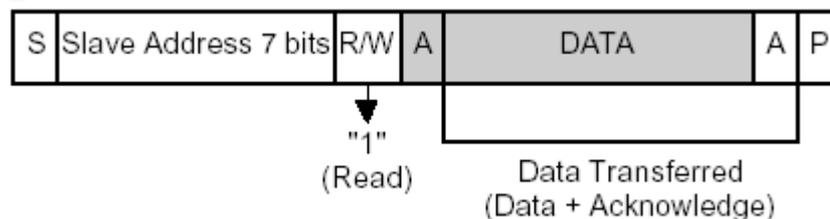


图 23-4 7 位地址读模式

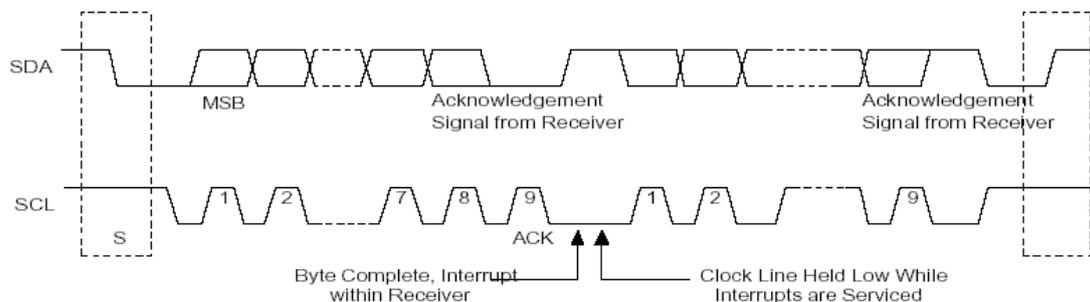


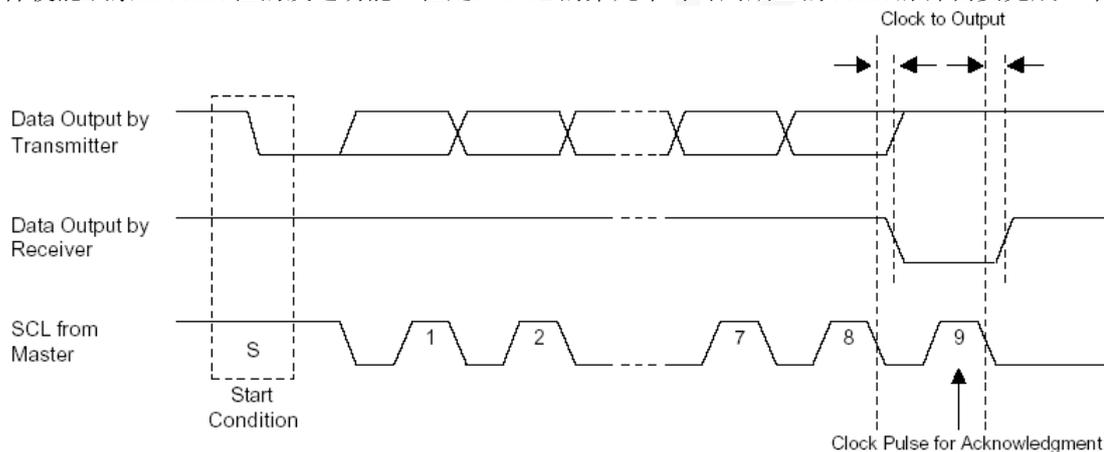
图 23-5 数据传输

### 23.6.3 ACK 信号传输

要完全完成一个字节的传输操作，接收器应向发送器发送一个 ACK 位。ACK 脉冲应在 SCL 线的第 9 个时钟周期发生。一字节数据传输需要八个时钟周期。主机应产生发送 ACK 位所需的时钟脉冲。

当接收到 ACK 时钟脉冲时，发送器应通过使 SDA 线设置为高电平来释放 SDA 线。接收器还应在 ACK 时钟脉冲期间将 SDA 线驱动为低电平，以便在第九个 SCL 脉冲的高电平期间 SDA 为低电平。

可以通过软件使能或禁止 ACK 位的发送功能。但是，SCL 的第九个时钟周期上的 ACK 脉冲需要完成一个字节的



数据传输操作。

图 23-6 I<sup>2</sup>C 总线的应答机制

### 23.6.4 读写操作

发送模式下，数据传输后，I<sup>2</sup>C 总线接口将等待，直到挂起的中断被清除。在中断被清除之前，SCL 线将保持低电平。中断清除后，SCL 线将被释放。在 CPU 接收到中断请求后，它应该在清除挂起中断之前将新数据写入 I2C\_DATA。

接收模式下，在接收到数据后，I<sup>2</sup>C 总线接口将等待，直到挂起的中断被清除。在等待中断被清除之前，SCL 线将保持低电平。等待中断清除后，SCL 线将被释放。在 CPU 接收到中断请求后，它应该在清除挂起中断之前从 I2C\_DATA 中读取数据。

#### 总线仲裁过程

仲裁发生在 SDA 线路上，以防止两个主机之间的总线上的竞争。如果 SDA 为高电平的主机监测到另一个主机的 SDA 为低电平，则它不会启动数据传输，因为总线上的当前电平与其自身不对应。仲裁过程将延长，直到 SDA 线变为高电平。

然而，当主机 SDA 线同时为低电平时，每个主机应该评估控制权是否属于自己。为了进行评估，每个主机应该监测地址位。当每个主机产生从机地址时，它也应该监测 SDA 线上的地址位，因为 SDA 电平为低比 SDA 线为高电平的优先级更高。例如，一个主机的首地址位是低电平，而另一个主机维持高电平，在这种情况下，两个主机将监测总线上的低电平，因为即使第一主机试图在电路上保持高电平，低电平也强于高电平。当这种情况发生时，产生低电平（作为地址的第一位）的主机将获得主机控制权，产生高电平（作为地址的第一位）的主机应该被取消控制权。如果两个主机都在第一位就产生低电平，则应再次对第二地址位进行仲裁。此仲裁将继续到最后一个地址位。

## 中止状态

如果从接收器不应答从机地址，它应该保持 SDA 线为高电平。在这种情况下，主机应产生停止状态并中止传输。

如果主接收器想中止传输，则它应在接收到从机的最后一个数据字节后取消生成 ACK，以通知从机发送操作的结束。从发射器应释放 SDA，以允许主机产生停止状态。

## 23.6.5 接口时序

下图显示了 I<sup>2</sup>C 总线的接口连接。

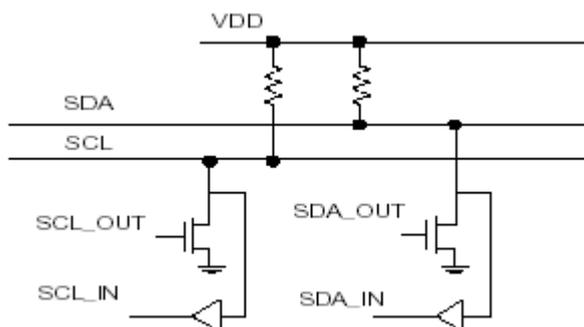


图 23-7 I<sup>2</sup>C 总线连接例

可通过 I2C\_CTRL 寄存器中的 CR2~CR0 实现不同串行时钟（SCL）的频率控制。

最大支持  $f_{SCL}=1\text{MHz}$ （周期：1 $\mu\text{s}$ ）

-SCL 和 SDA 的上升时间： $f_{SCL}=100\text{kHz}$ : 1000ns;  $f_{SCL}=400\text{kHz}$ : 300ns

-SCL 和 SDA 的下降时间： $f_{SCL}=100\text{kHz}$ : 300ns;  $f_{SCL}=400\text{kHz}$ : 300ns

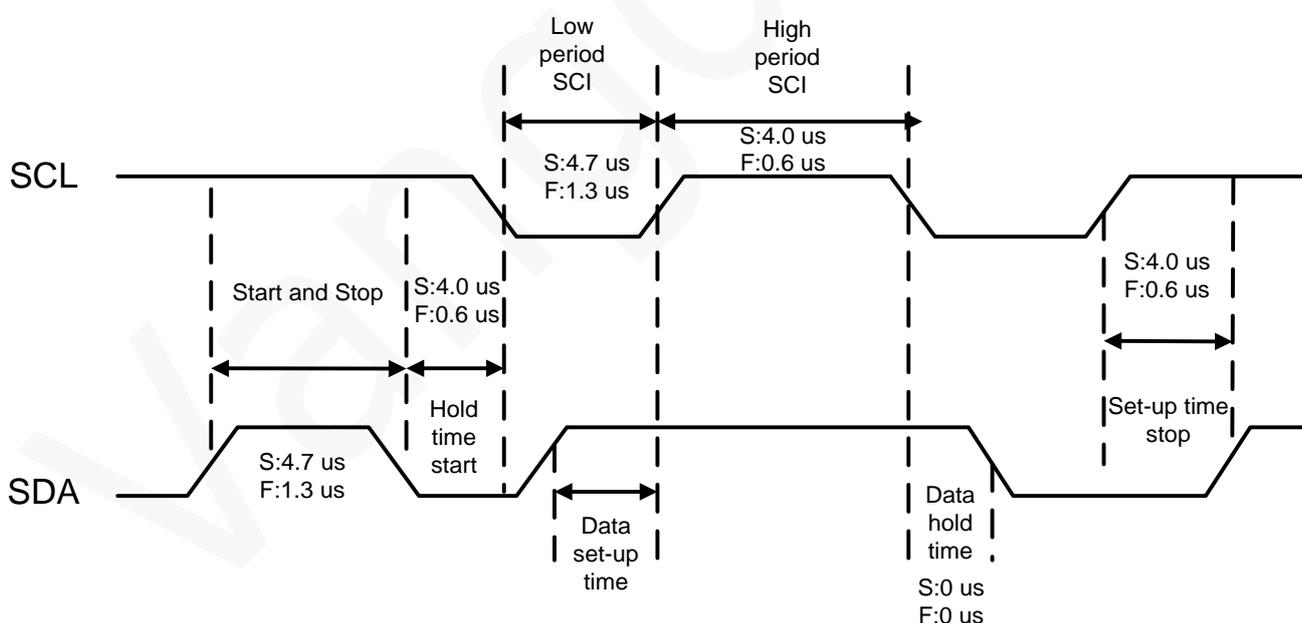


图 23-8 SCL 和 SDA 时序

表 23-13 I<sup>2</sup>C AC 特性

Time	I2C-mastermode(CR<7)		I2C-mastermode(CR=7)	
tHD: STA	scl/4	min(15clk)	scl/4	(2bclk)
tLOW	scl/2	min(30clk)	scl/2	(4bclk)
tHIGH	scl/2	min(30clk)	scl/2	(4bclk)

tSU: STA	scl/4	min(15clk)	当scl/4<7clk: scl/2	-
			其它: scl/4	-
tHD: DAT	6*clk	(6clk)	6*clk	(6clk)
tSU: DAT	1*clk(-forfirstbit)* <sup>[1]</sup>	(1clk)	1*clk(-forfirstbit)* <sup>[1]</sup>	(1clk)
	scl/2-6*clk	min(24clk)	scl/2-6*clk	min(1clk)
tSU: STO	scl/4+7*clk	min(22clk)	min(bclk+9)	-
tBUF	(3/4scl)+9*clk	min(54clk)	min(3/4scl)	-

<sup>[1]</sup>第一个字节数据的第一个位建立时间会短一点，只有一个 clk，而这种状况只会发生在写入数据寄存器时，刚好很接近内部 CSL 的上升沿。

clk: APB 时钟周期

bclk: 定时器 3 的溢出周期

scl: I<sup>2</sup>C 时钟周期

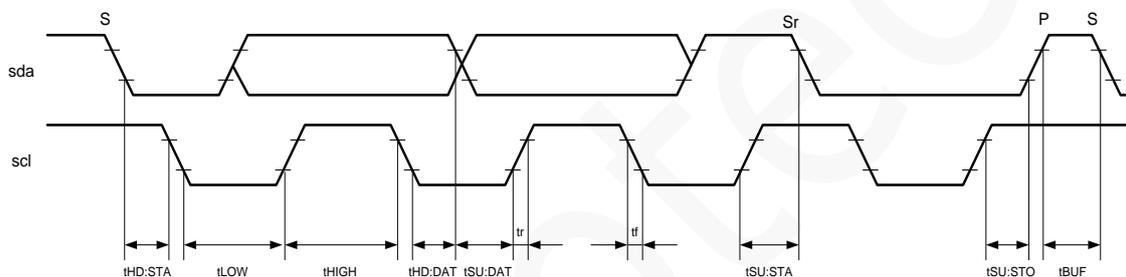


图 23-9 I<sup>2</sup>C 控制器的 AC 时序

## 23.6.6 I<sup>2</sup>C SDA 管脚死锁的解决方法

I<sup>2</sup>C 通讯过程中，当芯片异常复位（软复位、看门狗复位、外部 EXRST 管腿复位）发生时，有可能会导 I<sup>2</sup>C 总线死锁（SDA 死锁在低电平状态）。规避方法：在开启 I<sup>2</sup>C 之前，用户增加 I<sup>2</sup>C 总线恢复时序。即每次 I<sup>2</sup>C 主设备复位后，软件模拟控制 I<sup>2</sup>C 中的 SCL 时钟线产生 9 个时钟脉冲，这样 I<sup>2</sup>C 的 SDA 管腿可以从死锁状态中恢复过来。如果当前不处于死锁状态，增加总线恢复程序也不影响 I<sup>2</sup>C 使用。

## 23.7 工作流程

### 23.7.1 主机发送模式

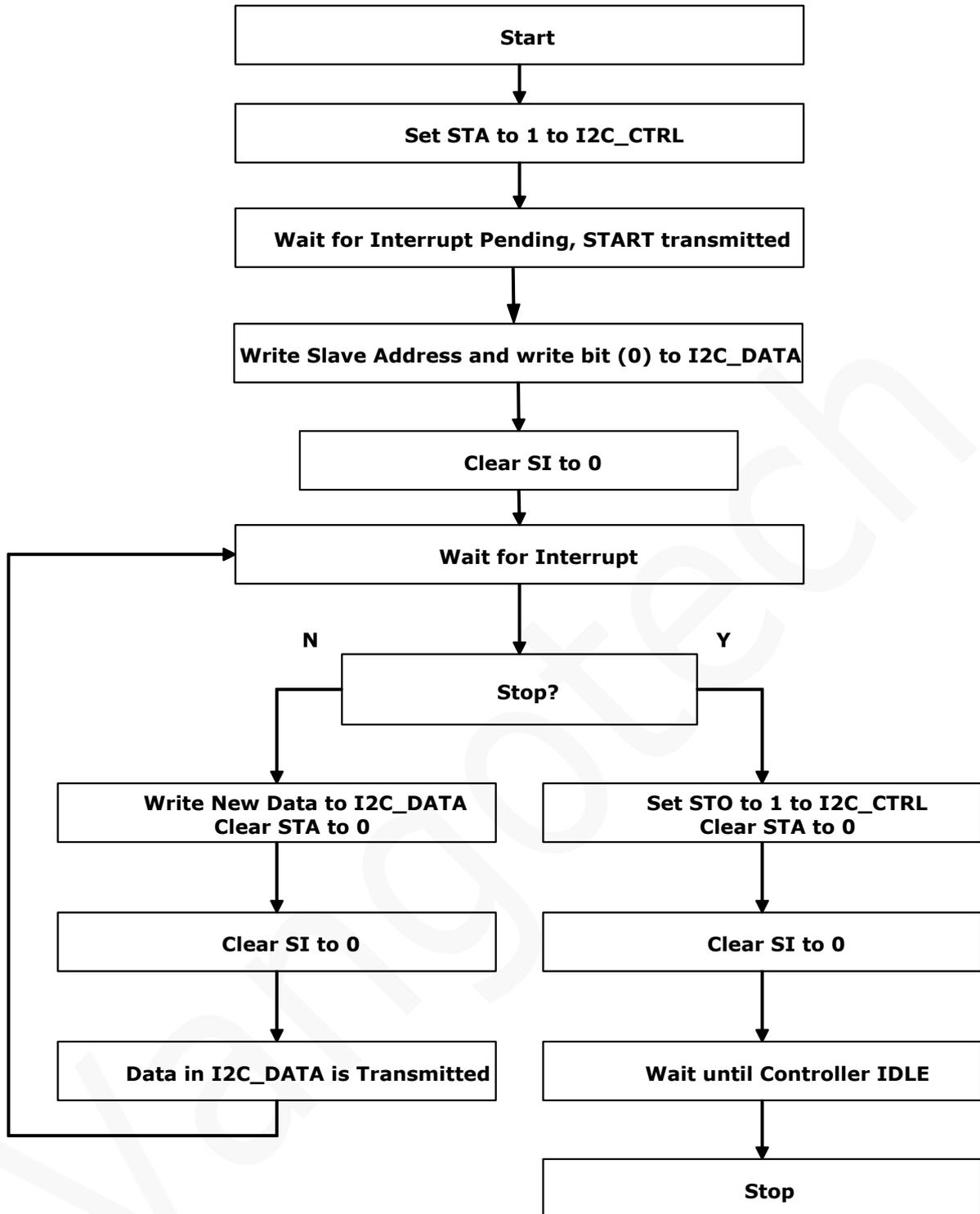


图 23-10 主机发送模式的操作流程

## 23.7.2 主机接收模式

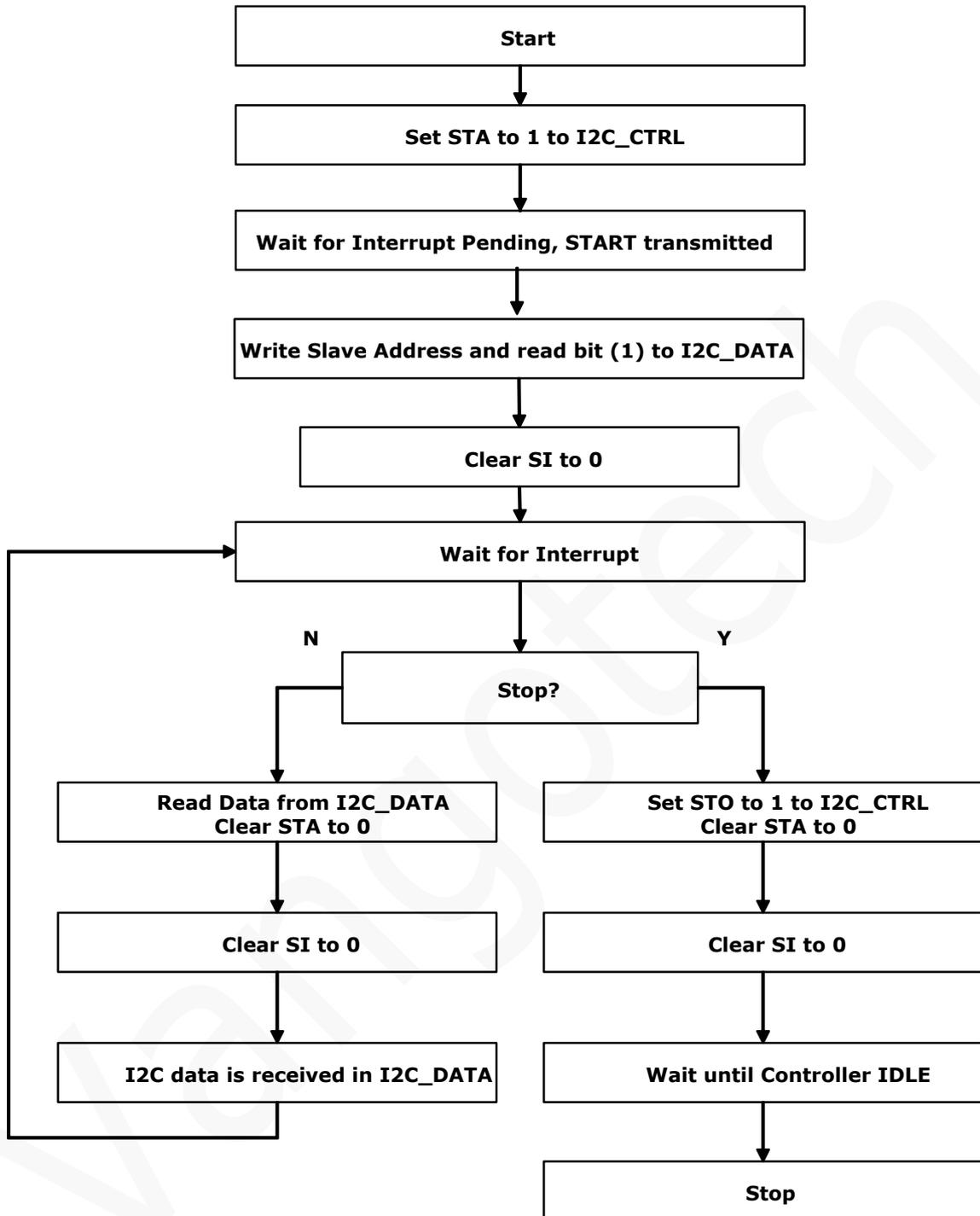


图 23-11 主机接收模式的操作流程

## 23.7.3 从机发送模式

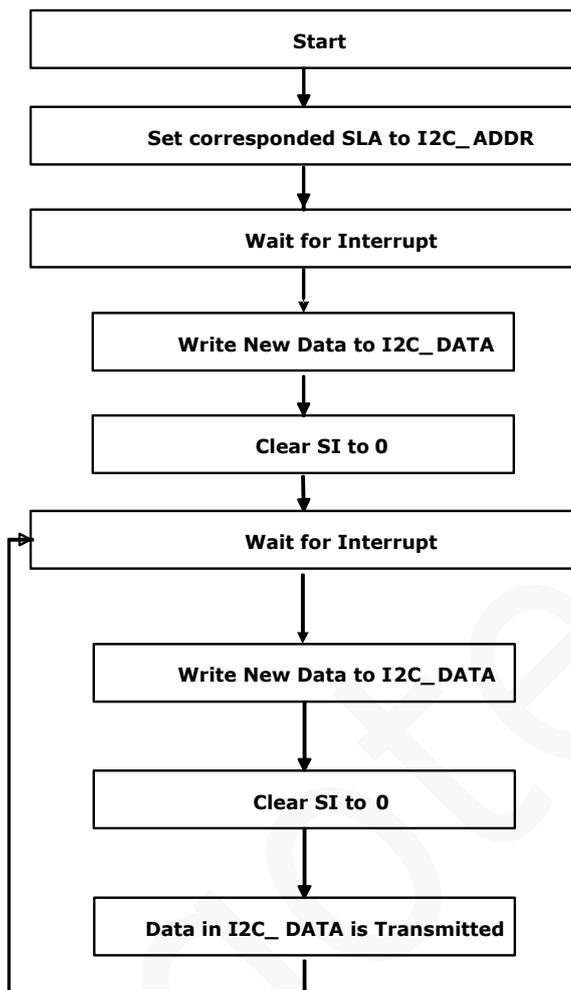


图 23-12 从机发送模式的操作流程

## 23.7.4 从机接收模式

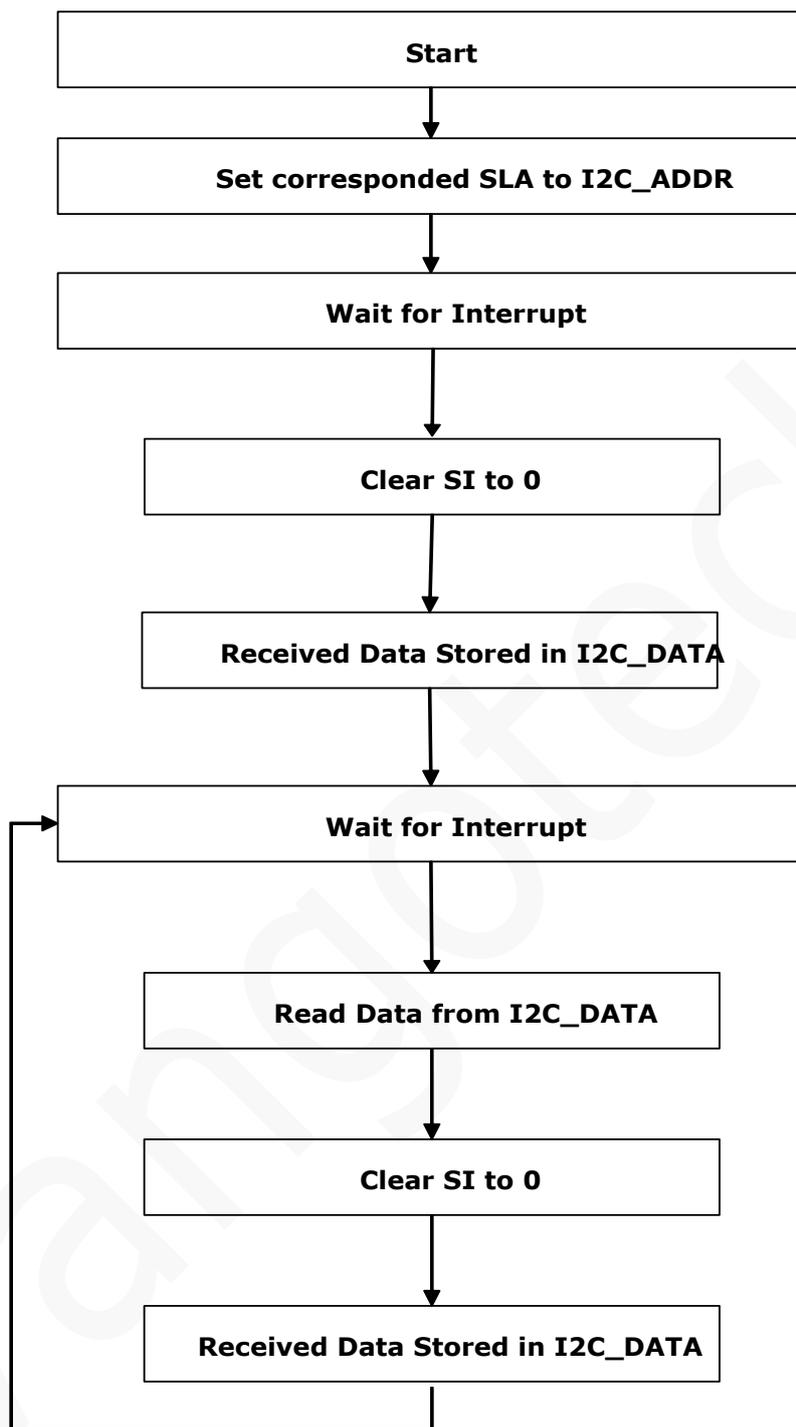


图 23-13 从机接收模式的操作流程

## 第24章 中断控制器

### 24.1 简介

任何中断都能将系统从挂起（IDLE）状态唤醒，其中的少部分可以将系统从浅睡眠（SLEEP）中唤醒，其中更少的部分可以将系统从深睡眠（DEEPSLEEP）中唤醒。

### 24.2 中断源

表 24-1 中断源

序列	向量地址	中断序号	描述	外设事件使能位	外设事件标记位	唤醒源	
						深睡眠	浅睡眠
NMI	00000008h	-14	NMI				
HardFault	0000000Ch	-13	HardFault				
SVCall	0000002Ch	-5	SVCall				
PendSV	00000038h	-2	PendSV				
SysTick	0000003Ch	-1	SysTick				
PMU	00000040h	0	IOA0~15	PMU_CONTROL. 0 和 PMU_IOAWKUEN .0~15	PMU_IOAINTSTS .0~15	√	√
			32K 晶振失效	PMU_CONTROL. 2	PMU_STS.0	√	√
			6M 晶振失效	PMU_CONTROL. 3	PMU_STS.1		
RTC	00000044h	1	ITV 和 SITV 中断	RTC_INTEN.0	RTC_INTSTS.0	√	√
			非法时间格式	RTC_INTEN.1	RTC_INTSTS.1	√	√
			多秒中断	RTC_INTEN.2	RTC_INTSTS.2	√	√
			多分钟中断	RTC_INTEN.3	RTC_INTSTS.3	√	√
			多小时中断	RTC_INTEN.4	RTC_INTSTS.4	√	√
			午夜（00: 00） 中断	RTC_INTEN.5	RTC_INTSTS.5	√	√
			32K 计数器中断	RTC_INTEN.6	RTC_INTSTS.6	√	√
			对 CE 寄存器的非法 写入	RTC_INTEN.8	RTC_INTSTS.8		
			闹钟中断	RTC_INTEN.10	RTC_INTSTS.10	√	√
U32K0~1	00000048h 0000004Ch	2\3	接收到数据	U32Kx_CTRL1.0	U32Kx_STS.0	√	√
			接收奇偶校验错误	U32Kx_CTRL1.1	U32Kx_STS.1	√	√
			接收缓存溢出	U32Kx_CTRL1.2	U32Kx_STS.2	√	√

I <sup>2</sup> C	00000050h	4	I <sup>2</sup> C 串行中断	I2C_CTRL2.0	I2C_CTRL.3		
SPI1~3	00000054h \000000ACh\000 000B0h	5\27\ 28	SPI 发送	SPIx_TXSTS.14	SPIx_TXSTS.15		
			SPI 接收	SPIx_RXSTS.14	SPIx_RXSTS.15		
UART0~5	00000058h ~0000006Ch	6\7\8 \9\10\ 11	接收中断	UARTx_CTRL.3	UARTx_INTSTS. 1		
			发送溢出	UARTx_CTRL.4	UARTx_INTSTS. 2		
			接收溢出	UARTx_CTRL.5	UARTx_INTSTS. 3		
			接收奇偶校验错误	UARTx_CTRL.7	UARTx_INTSTS. 4		
			发送完成	UARTx_CTRL.8	UARTx_INTSTS. 5		
ISO7816 0~1	00000070h ~00000074h	12\13	接收错误中断	ISO7816x_CFG. 2	ISO7816x_INFO .2		
			接收中断	ISO7816x_CFG. 5	ISO7816x_INFO .5		
			发送中断	ISO7816x_CFG. 6	ISO7816x_INFO .6		
			接收溢出	ISO7816x_CFG. 7	ISO7816x_INFO .7		
			传输重试错误中断	ISO7816x_CFG. 8	ISO7816x_INFO .8		
Timer0~ 3	00000078h ~00000084h	14\15 \16\17	定时器溢出	TMRx_CTRL.3	TMRx_INTSTS.0		
PWM0~3	00000088h ~00000094h	18\19 \20\21	PWM 定时器溢出	PWMx_CTL.1	PWMx_CTL.0		
			捕获/比较 0	PWMx_CCTL0.4	PWMx_CCTL0.0		
			捕获/比较 1	PWMx_CCTL1.4	PWMx_CCTL1.0		
			捕获/比较 2	PWMx_CCTL2.4	PWMx_CCTL2.0		
DMA	00000098h	22	通道 0 包结束	DMA_IE.0	DMA_STS.4		
			通道 1 包结束	DMA_IE.1	DMA_STS.5		
			通道 2 包结束	DMA_IE.2	DMA_STS.6		
			通道 3 包结束	DMA_IE.3	DMA_STS.7		
			通道 0 帧结束	DMA_IE.4	DMA_STS.8		
			通道 1 帧结束	DMA_IE.5	DMA_STS.9		
			通道 2 帧结束	DMA_IE.6	DMA_STS.10		
			通道 3 帧结束	DMA_IE.7	DMA_STS.11		
			通道 0 数据丢弃	DMA_IE.8	DMA_STS.12		

			通道 1 数据丢弃	DMA_IE.9	DMA_STS.13		
			通道 2 数据丢弃	DMA_IE.10	DMA_STS.14		
			通道 3 数据丢弃	DMA_IE.11	DMA_STS.15		
FLASH	0000009Ch	23	校验和错误	FLASH_CTRL.2	FLASH_INTSTS.0		
ANA	000000A0h	24	手动 ADC 转换完成	ANA_INTEN.0	ANA_INTSTS.0		
			自动 ADC 转换完成	ANA_INTEN.1	ANA_INTSTS.1		
			CMP1 的上升和下降沿	ANA_INTEN.2	ANA_INTSTS.2	√	√
			CMP2 的上升和下降沿	ANA_INTEN.3	ANA_INTSTS.3	√	√
			VDDALARM 的上升和下降沿	ANA_INTEN.7	ANA_INTSTS.7	√	√
			VDCIN 的上升和下降沿	ANA_INTEN.8	ANA_INTSTS.8	√	√
			AVCCLV 的上升和下降沿	ANA_INTEN.10	ANA_INTSTS.10	√	√
			VDCINDROP=0 同时进入了浅睡眠/深睡眠状态	ANA_INTEN.11	ANA_INTSTS.11	√	√
			ANA_REGx 错误	ANA_INTEN.12	ANA_INTSTS.12	√	√
			TADC 变化超过阈值	ANA_INTEN.13	ANA_INTSTS.13	√	√
			ADC 数据超阈值中断	ANA_INTEN.14~21	ANA_INTSTS.14~21		

## 第25章 MISC 控制器

### 25.1 简介

MISC 控制器用于控制 V85XXP 的某些特殊功能。V85XXP 内置两个 MISC 控制器，一个在核心域（MISC1 控制器），在浅睡眠和深睡眠模式下会掉电。从浅睡眠或深睡眠模式唤醒后，MISC 控制器的设置会被复位，在从这两个状态唤醒后，用户应手动恢复设置。另一个（MISC2 控制器）处于保留域，会在深睡眠模式下掉电。在从深睡眠模式唤醒后，MISC2 控制器的设置会被复位，用户应在深睡眠模式唤醒后手动恢复设置。

### 25.2 特点

- 每个子模块的时钟控制；
- AHBCLK 和 APBCLK 的时钟分频器；
- FLASH 编程滴答控制；
- 串口 IR 占空比控制；
- SRAM 奇偶校验中断控制。

### 25.3 功能框图

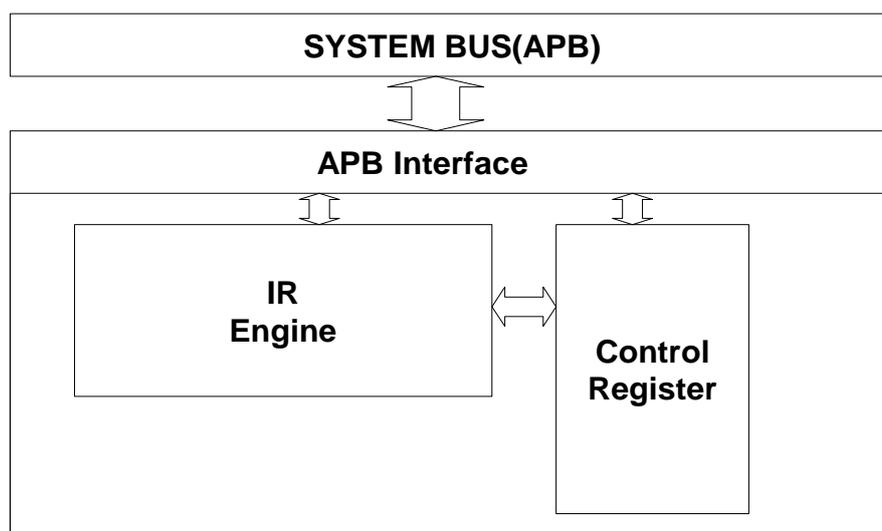


图 25-1 MISC 控制器的功能框图

### 25.4 寄存器地址

表 25-1 MISC1 控制器(MISC1 基地址: 0x40013000)

名称	类型	地址	描述	默认值
MISC1_SRAMINT	R/C	0x0000	SRAM 奇偶校验错误中断状态寄存器	0x00
MISC1_SRAMINIT	R/W	0x0004	SRAM 初始化寄存器	0x01
MISC1_PARERR	R	0x0008	SRAM 奇偶校验错误地址寄存器	0x000
MISC1_IREN	R/W	0x000C	IR 使能控制寄存器	0x00

MISC1_DUTYL	R/W	0x0010	IR 调制波低电平宽度控制寄存器	0x0000
MISC1_DUTYH	R/W	0x0014	IR 调制波高电平宽度控制寄存器	0x0000
MISC1_IRQLAT	R/W	0x0018	Cortex-M0 IRQ 延迟控制寄存器	0x000
MISC1_HIADDR	R	0x0020	记录的非法访问 AHB 地址	--
MISC1_PIADDR	R	0x0024	记录的非法访问 APB 地址	--

表 25-2 MISC2 控制器(MISC2 基地址: 0x40013E00)

名称	类型	地址	描述	默认值
MISC2_FLASHWC	R/W	0x0000	FLASH 等待周期寄存器	0x2100
MISC2_CLKSEL	R/W	0x0004	系统选择寄存器	0x0
MISC2_CLKDIVH	R/W	0x0008	AHB 时钟分频控制寄存器	0x00
MISC2_CLKDIVP	R/W	0x000C	APB 时钟分频控制寄存器	0x01
MISC2_HCLKEN	R/W	0x0010	AHB 时钟使能控制寄存器	0x1FF
MISC2_PCLKEN	R/W	0x0014	APB 时钟使能控制寄存器	0xFFFFFFFF

## 25.5 寄存器定义

### 25.5.1 MISC1\_SRAMINT 寄存器

表 25-3 MISC1\_SRAMINT 寄存器

位	名称	类型	描述	默认值
31: 5	-	-	保留	0
4	LOCKUP	R/C	该位表示发生了 CM0 锁定, 写 1 清除该标志	0x0
3	PIAC	R/C	该位表示进行了 APB 总线上的无效地址访问。无效 APB 地址是不在 0x40010000~0x4001FFFF 区域内的地址。如果 PIACIE 为 1, 则向 CM0 发出 NMI 中断。写 1 清除该标志。	0x0
2	HIAC	R/C	该位表示进行了 AHB 总线上的无效地址访问。无效 AHB 地址是 FLASH、SRAM 或 IO 区域之外的地址。如果 HIACIE 为 1, 则将向 CM0 发出 NMI 中断。写 1 清除该标志。	0x0
1	HIAL	R/C	此位表示在 AHB 总线上发生无效的对齐访问。同时, 故障地址被锁存到 MISC1_HIADDR 中。写 1 清除此标志。	0
0	PERR	R/C	该位表示在 SRAM 读取过程中发生 SRAM 奇偶校验错误。如果 PERRIE 为 1, 则将向 CM0 发出 NMI 中断。写 1 清除该标志。	0x0

### 25.5.2 MISC1\_SRAMINIT 寄存器

表 25-4 MISC1\_SRAMINIT 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7	LOCKIE	R/W	CM0 发生锁定时, 使能 NMI	0x0
6	PIACIE	R/W	发生 APB 无效地址访问, 使能 NMI	0x0
5	HIACIE	R/W	发生 AHB 无效地址访问, 使能 NMI	0x0

4: 3			保留	0
2	INIT	R/W	SRAM 初始化寄存器, 将该寄存器置 1, 会使用奇偶校验正确的全零值初始化 SRAM, 该位将自动清零。在初始化期间, 不允许访问总线上的任何主机 (CPU 或 DMA), 否则访问值可能是错误值	0x0
1	PERRIE	R/W	SRAM 奇偶校验错误时, 使能 NMI	0x0
0	PEN	R/W	奇偶校验使能寄存器。当该位为 1 时, 任何对 SRAM 的写访问, 都会向奇偶校验缓存器写入校验信息。SRAM 读取时, 将进行自动奇偶校验检查, 并在奇偶校验失败时产生奇偶校验错误中断。该功能默认打开, 因此无需在软件端进行特殊操作。但是如果 CPU 或 DMA 访问一些未初始化区域, 可能会发生奇偶校验错误, 因为非初始化区域中不存在奇偶校验信息。此时, 用户可以使用 INIT 位自动进行 SRAM 初始化。	0x1

### 25.5.3 MISC1\_PARERR 寄存器

表 25-5 MISC1\_PARERR 寄存器

位	名称	类型	描述	默认值
31: 14	-	-	保留	0
13: 0	PEADDR	R	奇偶校验错误地址。该寄存器存储奇偶校验错误地址的信息, 当校验错误发生时, 用户可以通过该寄存器定位 SRAM 校验出现错误的地址。 SRAM 校验错误地址=0x20000000+4*PEADDR	--

### 25.5.4 MISC1\_IREN 寄存器

表 25-6 MISC1\_IREN 寄存器

位	名称	类型	描述	默认值
31: 6	-	-	保留	0
5: 0	IREN	R/W	IR 使能控制寄存器。该寄存器中的每个位对应于 1 个 UARTTX 通道。当 IREN[x] 设置为 1 时, 表示 UARTTX[x] 将被 IR 脉冲调制输出。	0x0

### 25.5.5 MISC1\_DUTYL 寄存器

表 25-7 MISC1\_DUTYL 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	DUTYL	R/W	红外载波 (IR) 调制波低电平宽度控制寄存器。低脉冲宽度将是 (DUTYL+1) * APBCLK 周期。	0x0000

### 25.5.6 MISC1\_DUTYH 寄存器

表 25-8 MISC1\_DUTYH 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0

15: 0	DUTYH	R/W	红外载波 (IR) 调制波高电平宽度控制寄存器。高脉冲宽度为 (DUTYH+1) * APBCLK 周期。	0x0000
-------	-------	-----	---	--------

## 25.5.7 MISC1\_IRQLAT 寄存器

表 25-9 MISC1\_IRQLAT 寄存器

位	名称	类型	描述	默认值
31: 10	-	-	保留	0
9	NOHARDFFAULT	R/W	该寄存器用于禁止 CPU 产生 HardFault 中断。 0: 当发生总线错误时, 使能 HardFault 中断。 1: 当发生总线错误时, 禁止 HardFault 中断。 当 CM0 监测发生硬件故障, 将跳转到硬件故障中断服务程序。硬件故障服务程序执行完毕后, 程序将跳到正常代码并再次执行相同的指令。因此, 如果硬故障再次发生, 它将停在这个循环中, 永远不会出来。该寄存器用于 CPU 暂时禁止硬故障产生。	0x0
8	LOCKRESET	R/W	此寄存器用于控制锁定是否发出系统复位。 0: 禁止 CM0 锁定的复位产生。 1: 使能 CM0 锁定的复位产生。	0x0
7: 0	IRQLAT	R/W	该寄存器用于控制 Cortex-M0 IRQ 响应延迟。 如果设置为 0, 则将尽可能快的进中断。对于零等待状态下实现零抖动, 则至少设置为 13 (DEC)。对于非零等待状态, 可以使用更高的值来实现零抖动。	0x00

## 25.5.8 MISC1\_HIADDR 寄存器

表 25-10 MISC1\_HIADDR 寄存器

位	名称	类型	描述	默认值
31: 0	HIADDR	R	AHB 总线错误地址。该寄存器存储访问 AHB 总线错误地址的信息, 当错误发生时, 用户可以通过该寄存器定位 AHB 总线访问出现错误的地址。 错误地址 = HIADDR	--

## 25.5.9 MISC1\_PIADDR 寄存器

表 25-11 MISC1\_PIADDR 寄存器

位	名称	类型	描述	默认值
31: 0	PIADDR	R	APB 总线错误地址。该寄存器存储访问 APB 总线错误地址的信息, 当错误发生时, 用户可以通过该寄存器定位 APB 总线访问出现错误的地址。 错误地址 = 0x40000000 + PIADDR	--

## 25.5.10 MISC2\_FLASHWC 寄存器

表 25-12 MISC2\_FLASHWC 寄存器

位	名称	类型	描述	默认值
31: 14	-	-	保留	0

13: 8	CYCLE_1US	R/W	<p>这个寄存器用于 FLASH 控制器以 AHBClock 计算 1<math>\mu</math>s 的滴答。该设置与 FLASH 的唤醒时间以及 FLASH 的擦除时间、FLASH 的编程时间有关。FLASH 唤醒时间=1<math>\mu</math>s 的滴答时间*10。必须满足 1<math>\mu</math>stick<math>\geq</math>1<math>\mu</math>s。</p> <p>1<math>\mu</math>stick=(AHB 时钟周期)*(CYCLE_1US+1)。</p> <p>比如 AHB 时钟频率为 26.2144M, 为了保证 1<math>\mu</math>s 的滴答时间, 以及最小的唤醒时间, CYCLE_1US 应该设置为 26。此时 FLASH 的唤醒时间为 27 / 26214400 * 10, 大约为 10<math>\mu</math>s。</p> <p>比如 AHB 时钟频率为 32.768k, 为了保证 1<math>\mu</math>s 的滴答时间, 以及最小的唤醒时间, CYCLE_1US 应该设置为 0。此时 FLASH 的唤醒时间是 1 / 32768 * 10, 大约为 305<math>\mu</math>s。</p>	0x21
7: 0	-	-	保留	0

### 25.5.11 MISC2\_CLKSEL 寄存器

表 25-13 MISC2\_CLKSEL 寄存器

位	名称	类型	描述	默认值
31: 3			保留	0
2: 0	CLKSEL	R/W	<p>该寄存器用于控制 AHB 时钟源。</p> <p>0: RCH (6.5MRC)</p> <p>1: XOH (6.5MXTAL)</p> <p>2: PLLH</p> <p>3: RTCCLK (由 PMU_CONTROL 寄存器中的 RTCCLK_SEL 控制)</p> <p>4: PLLL</p> <p>5: 保留</p> <p>6: 保留</p> <p>7: 保留</p> <p>在时钟选择到一个时钟源之前, 用户应首先通过设置 PMU_CONTROL 寄存器来使能相应的模块。</p>	0x0

### 25.5.12 MISC2\_CLKDIVH 寄存器

表 25-14 MISC2\_CLKDIVH 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7: 0	CLKDIVH	R/W	<p>该寄存器用于控制 AHB 时钟分频</p> <p>0: 时钟源除以 1</p> <p>1: 时钟源除以 2</p> <p>2: 时钟源除以 3</p> <p>...</p> <p>255: 时钟源除以 256</p>	0x00

### 25.5.13 MISC2\_CLKDIVP 寄存器

表 25-15 MISC2\_CLKDIVP 寄存器

位	名称	类型	描述	默认值
31: 8	-	-	保留	0
7: 0	CLKDIVP	R/W	该寄存器用于控制 APB 时钟分频 0: AHB 时钟除以 1 1: AHB 时钟除以 2 2: AHB 时钟除以 3 ... 255: AHB 时钟除以 256	0x01

### 25.5.14 MISC2\_HCLKEN 寄存器

表 25-16 MISC2\_HCLKEN 寄存器

位	名称	类型	描述	默认值
31: 9	-	-	保留	0
8: 0	HCLKEN	R/W	该寄存器用于控制每个 AHB 模块的时钟使能。对应的模块只有在其功能未被使用时才能被关闭。有关每个模块的详细信息，请参见下表。 0: 关闭; 1: 开启。	0x1FF

表 25-17 HCLK 时钟使能

位	模块	注意事项
0	--	保留
1	Arbiter 和 BusMatrix	保留
2	FLASH 控制器	保留
3	SRAM 控制器	保留
4	DMA 控制器	
5	GPIO 控制器	
6	LCD 控制器	
7	--	
8	CRYPT 控制器	

### 25.5.15 MISC2\_PCLKEN 寄存器

表 25-18 MISC2\_PCLKEN 寄存器

位	名称	类型	描述	默认值
31: 0	PCLKEN	R/W	该寄存器用于控制每个 APB 模块的时钟使能。对应的模块只有在其功能未被使用时才能被关闭。有关每个模块的详细信息，请参见下表。 0: 关闭;	0xFFFFFFFF

1: 开启。

表 25-19 PCLK 时钟使能

位	模块	注意事项
0	AHB2APB Bridge	保留
1	DMA 控制器	
2	I <sup>2</sup> C	
3	SPI1	
4	UART0	
5	UART1	
6	UART2	
7	UART3	
8	UART4	
9	UART5	
10	ISO78160	
11	ISO78161	
12	TIMER	
13	MISC1	
14	MISC2	
15	PMU	
16	RTC	
17	ANA	
18	U32K0	
19	U32K1	
20	保留	
21	SPI2	
22	SPI3	
31: 23	保留	

## 第26章 CRYPT 控制器

### 26.1 简介

CRYPT 控制器用来给 ECC 加速。CRYPT 控制器的主要特征是实现 VLI 可变长度整数乘法、加法（带进位）、减法（带借位）和移位。它比纯软件实现的处理能力提高 4 倍。凡是用到 VLI 的加密算法都可以用该硬件加速处理，提升处理速度。CRYPT 控制器能直接访问 SRAM，而不会影响 M0 处理器核的访问。

### 26.2 特点

- 支持 32 位到 512 位的可变长度整数；
- VLI 乘法；
- VLI 带进位加法；
- VLI 带借位减法；
- VLI 逻辑右移一位；
- 直接访问内部 SRAM。

### 26.3 功能框图

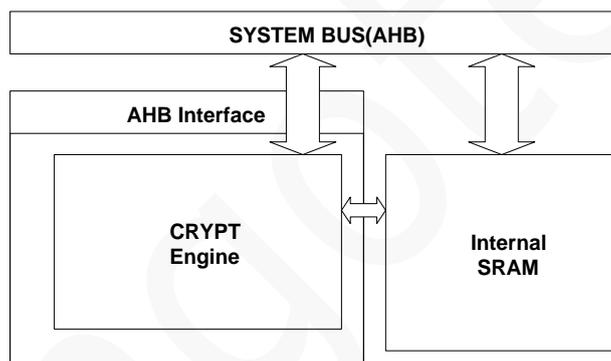


图 26-1 CRYPT 控制器的功能框图

### 26.4 寄存器地址

表 26-1 CRYPT 控制器(CRYPT 基地址: 0x40006000)

名称	类型	地址	描述	默认值
CRYPT_CTRL	R/W	0x0000	CRYPT 控制寄存器	0x0000
CRYPT_PTRA	R/W	0x0004	CRYPT 指针 A	0x0000
CRYPT_PTRB	R/W	0x0008	CRYPT 指针 B	0x0000
CRYPT_PTRO	R/W	0x000C	CRYPT 指针 O	0x0000
CRYPT_CARRY	R	0x0010	CRYPT 借位/进位寄存器	0x0

### 26.5 寄存器定义

#### 26.5.1 CRYPT\_CTRL 寄存器

表 26-2 CRYPT\_CTRL 寄存器

位	名称	类型	描述	默认值
---	----	----	----	-----

31: 16	-	-	保留	0
15	NOSTOP	R/W	该位用来设置当 CRYPT 忙碌时，是否允许 CPU 读写 CRYPT 寄存器。 0: 当 CRYPT 忙碌时禁止 CPU 访问 CRYPT 寄存器 1: 当 CRYPT 忙碌时允许 CPU 访问 CRYPT 寄存器 该位仅在 CPU 要访问 CRYPT 寄存器时起作用，如果 CPU 没有访问 CRYPT 寄存器，该位是无影响的。	0x0
14: 12	-	-	保留	0
11: 8	LENGTH	R/W	该位用来控制当前操作数据 (VLI) 的长度。该长度的设置可以和 ACT 置 1 同时操作。但是当 ACT 已经是 1 时，不能改变该位的设置。 0: 32 位 1: 64 位 2: 96 位 ..... 15: 512 位	0x0
7	-	-	保留	0
6: 4	MODE	R/W	该位用来控制 CRYPT 的操作模式。该位的设置与 ACT 置 1 可以同时操作。但是当 ACT 已经是 1 时，不能改变该位的设置。 0: 乘法模式, $*PTRO=*PTRA*PTRB$ 1: 加法模式, $*PTRO=*PTRA+*PTRB$ 2: 减法模式, $*PTRO=*PTRA-*PTRB$ 3: 逻辑右移一位, $*PTRO>(*PTRA>>1)$ 4~7: 保留	0x0
3: 1	-	-	保留	0
0	ACT	R/W	该位写 1 将启动 MODE 寄存器所指定的运算，在运算完成后，该位会自动清零。对该位写 0 是无效的。	0x0

## 26.5.2 CRYPT\_PTRA 寄存器

表 26-3 CRYPT\_PTRA 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	PTRA	R/W	CRYPT 控制器的指针 A。PTRA 的数值是 SRAM 的一个地址，该地址中的数据用做 CRYPT 计算。由于 CRYPT 只能控制 32 位的访问，所以 PTRA[1: 0]应该始终是 0。	0x0000

## 26.5.3 CRYPT\_PTRB 寄存器

表 26-4 CRYPT\_PTRB 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0

15: 0	PTRB	R/W	CRYPT 控制器的指针 B。PTRB 的数值是 SRAM 的一个地址，该地址中的数据用做 CRYPT 计算。由于 CRYPT 只能控制 32 位的访问，所以 PTRB[1: 0]应该始终是 0。	0x0000
-------	------	-----	--	--------

## 26.5.4 CRYPT\_PTRO 寄存器

表 26-5 CRYPT\_PTRO 寄存器

位	名称	类型	描述	默认值
31: 16	-	-	保留	0
15: 0	PTRO	R/W	CRYPT 控制器的指针 O。PTRO 的数值是 SRAM 的一个地址，该地址用来存储 CRYPT 的计算结果。由于 CRYPT 只能控制 32 位的访问，所以 PTRO[1: 0]应该始终是 0。	0x0000

## 26.5.5 CRYPT\_CARRY 寄存器

表 26-6 CRYPT\_CARRY 寄存器

位	名称	类型	描述	默认值
31: 1	-	-	保留	0
0	CARRY	R	该位代表加法后产生了进位或者减法后产生了借位。用户在运算完成后（ACT 被清 0）直接读取该寄存器位的值。	0x0

## 26.6 使用时的注意事项

### 26.6.1 数据格式

VLI 是一个 32~512 位之间的正整数，所以所有的操作都用的是无符号数计算。下面表格中列出了 SRAM 中的数据顺序。

地址	数据
PTR+0	VLI[31: 0]
PTR+4	VLI[63: 32]
PTR+8	VLI[95: 64]
PTR+12	VLI[127: 96]
.....	.....
PTR+60	VLI[511: 480]

在不同的长度模式（LENGTH 寄存器）下，只有 VLI 定义的数据长度才能作为输入数据。下表是在不同 LENGTH 设置下有效的 VLI 位数。

LENGTH	Data
0	VLI[31: 0]
1	VLI[63: 0]
2	VLI[95: 0]
3	VLI[127: 0]
.....	.....
15	VLI[511: 0]

## 26.6.2 运算细节

对于每一种运算，它们都会表现出多种不同的行为，比如处理的时间周期、输出数据宽度等。下表详细列出了每一种模式的运算细节。

	<b>MULT</b>	<b>ADD</b>	<b>SUB</b>	<b>RSHIFT1</b>
<b>Operation</b>	*PTRO= *PTRA x *PTRB	*PTRO= *PTRA + *PTRB	*PTRO= *PTRA - *PTRB	*PTRO= (*PTRA>>1)
<b>Outputlength</b>	2*input length	Input length	Input length	Input length
<b>CARRYbit</b>	X	0: 输出没有进位 1: 输出有进位	0: *PTRA>*PTRB 1: *PTRA<*PTRB	X
<b>LENGTH</b>	运算周期			
0	8	6	6	6
1	17	8	8	8
2	32	12	12	10
3	53	14	14	12
4	80	18	18	14
5	113	20	20	16
6	152	24	24	18
7	197	26	26	20
8	248	30	30	22
9	305	32	32	24
10	368	36	36	26
11	437	38	38	28
12	512	42	42	30
13	593	44	44	32
14	680	48	48	34
15	773	50	50	36

默认情况下，运算过程中，任何 CPU 对 CRYPT 寄存器的访问都会挂起 CPU，直到运算结束。运算过程中不允许改变配置，所以运算过程中有高优先级事件发生时，都会被阻塞，直到 CPU 被 CRYPT 事件释放。将 NOSTOP 位置为 1 会防止这种情况的发生，但在进行 CRYPT 运算过程中需要确保 CRYPT 配置参数不变。

因为 CRYPT 只支持 SRAM 的数据操作，所以当其中一个操作数不在 SRAM 中时，运算之前，用户需要先将该数据搬移到 SRAM 中。

## 第27章 调试特点

V85XXP 使用的是 Cortex-M0 内核，包含硬件调试模块。

### 27.1 特点

Cortex-M0 的硬件调试包含如下几个特点：

- 程序的暂停、恢复以及单步执行；
- 访问处理器核寄存器和特殊功能寄存器；
- 最多支持 4 个硬件断点；
- 软件断点(BKPT 指令)；
- 数据监测点(最多两个)；
- 动态存储器访问；
- 支持串行线调试 SWD 协议。

### 27.2 SWD 引脚

表 27-1 SWD 引脚的特殊功能

模式	管脚 1	管脚 2
调试模式 MODE 脚是低电平	SWCLK	SWDIO
非调试模式 MODE 脚是高电平	LCDSEG54/IOA0	LCDSEG53/IOA1

## 第28章 Cortex-M0 Core 简要描述

ARM Cortex-M0 处理器主要是满足超低功耗 MCU 和混合信号器件而设计的。它采用三级流水线冯·诺伊曼结构 (VonNeumann architecture)。通过简单、功能强大的指令集以及全面优化的设计 (提供包括一个单周期乘法器在内的高端处理硬件), ARM Cortex-M0 处理器可实现极高的能效。ARM Cortex-M0 处理器采用 ARMv6-M 结构, 基于 16 位的 Thumb 指令集, 并包含 Thumb-2 技术。

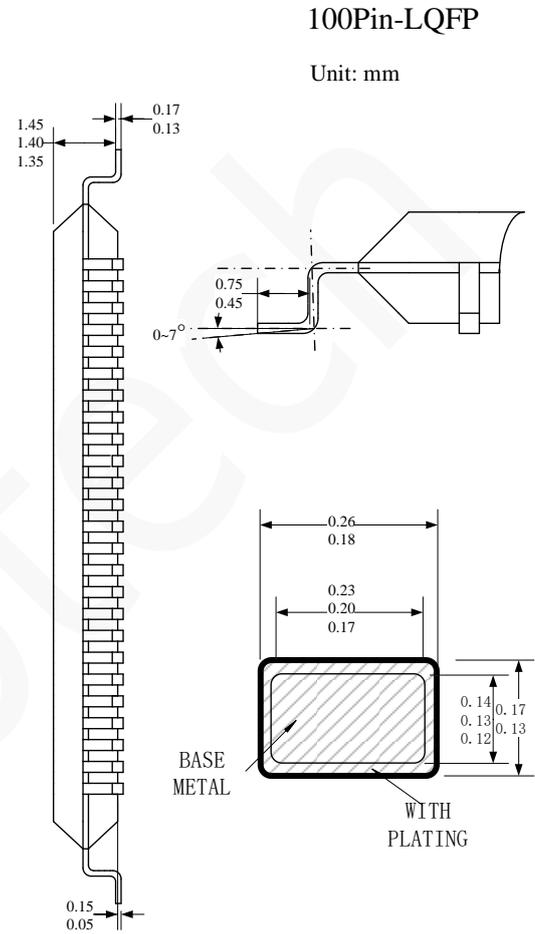
### 28.1 CMSIS 函数说明

表 28-1 CMSIS 函数

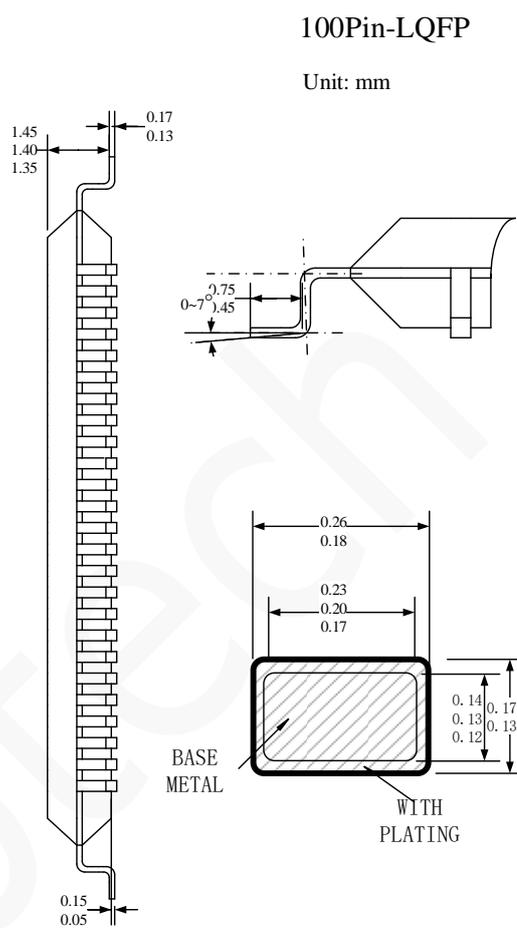
函数	说明
<code>void NVIC_EnableIRQ(IRQn_Type IRQn);</code>	使能中断号为IRQn的中断, 该功能不适用于系统异常的情况
<code>void NVIC_DisableIRQ(IRQn_Type IRQn);</code>	屏蔽中断号为IRQn的中断, 该功能不适用于系统异常的情况
<code>void VIC_SetPendingIRQ(IRQn_Type IRQn);</code>	设置中断号为IRQn中断的挂起状态, 该功能不适用于系统异常的情况
<code>void NVIC_ClearPendingIRQ(IRQn_Type IRQn);</code>	清除中断号为IRQn中断的挂起状态, 该功能不适用于系统异常的情况
<code>uint32_t NVIC_GetPendingIRQ(IRQn_Type IRQn);</code>	获取中断号为IRQn中断的挂起状态
<code>void NVIC_SetPriority(IRQn_Type IRQn, uint32_t priority);</code>	设置中断号为IRQn中断的优先级
<code>uint32_t NVIC_GetPriority(IRQn_Type IRQn);</code>	获取中断号为IRQn的中断优先级
<code>void __enable_irq(void);</code>	使能全局中断
<code>void __disable_irq(void);</code>	屏蔽全局中断(除了hardfault和NMI)
<code>uint32_t SysTick_Config(uint32_t ticks);</code>	初始化并启动SysTick计数器和它的中断
<code>void NVIC_SystemReset(void);</code>	M0内核软复位
<code>SCB-&gt;SCR  = SCB_SCR_SLEEPDEEP_Msk;</code> <code>void __WFI(void);</code>	等待中断 (进入浅睡眠模式)
<code>SCB-&gt;SCR &amp;= (uint32_t)~((uint32_t)SCB_SCR_SLEEPDEEP_Msk);</code> <code>void __WFI(void);</code>	等待中断 (进入IDLE模式)

## 第29章封装尺寸图

### 29.1 V8503P 封装尺寸图

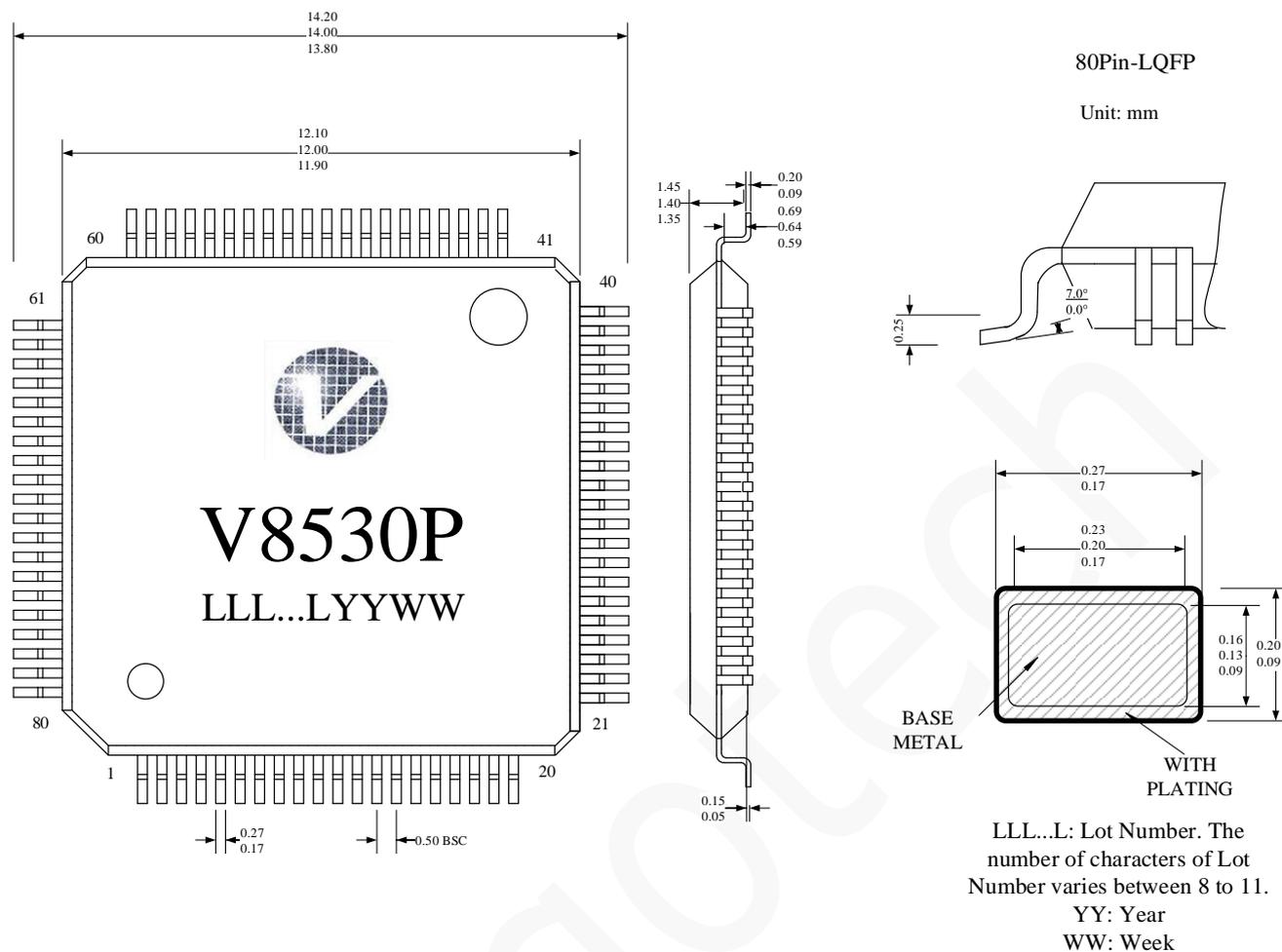


29.2 V8500P 封装尺寸图

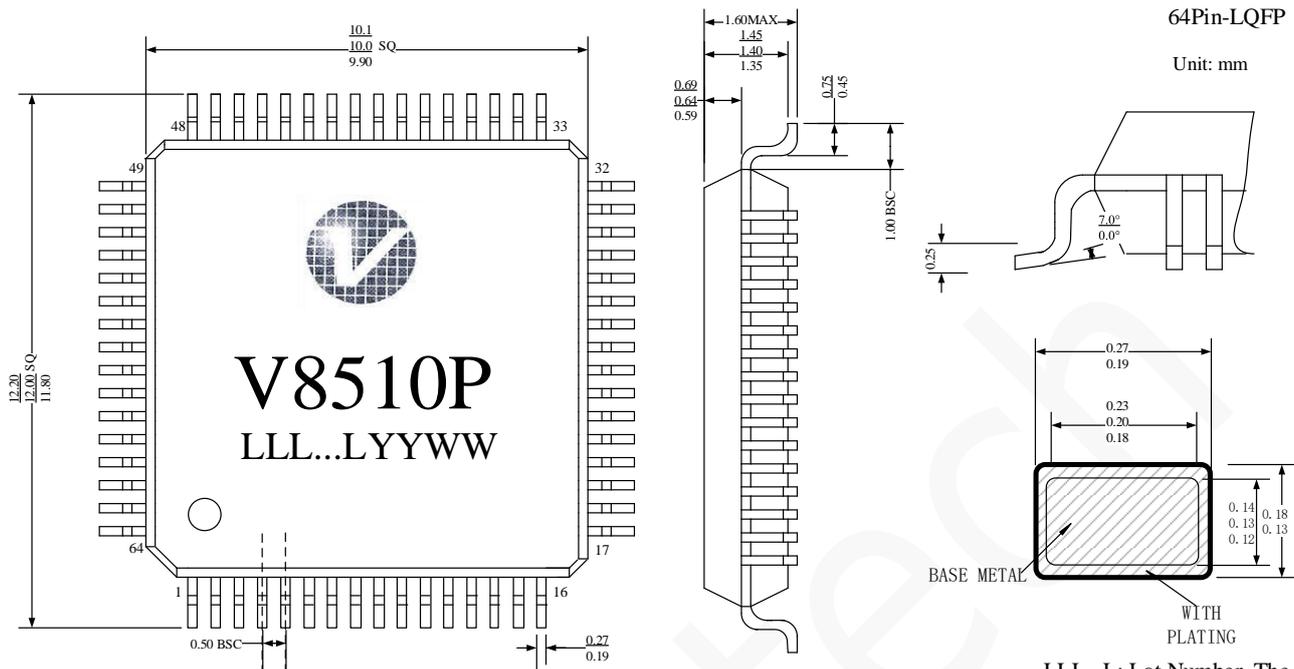


LLL...L: Lot Number. The number of characters of Lot Number varies between 8 to 11.  
YY: Year  
WW: Week

29.3 V8530P/V8531P 封装尺寸图



29.4 V8510P 封装尺寸图



LLL...L: Lot Number. The number of characters of Lot Number varies between 8 to 11.  
 YY: Year  
 WW: Week

## 图索引

图 3-1 V85XXP 系统功能框图 .....	35
图 3-2 V85XXP 电源系统功能框图 .....	36
图 3-3 V85XXP 时钟功能框图 .....	37
图 4-1 V85XXP 内存映射 .....	38
图 5-1 V85XXP 电源系统框图 .....	61
图 5-2 AVCCCLDO 输出电压与 VDD 输入电压之间的关系 .....	67
图 5-3 AVCCCLDO 输出电压与负载电流之间的关系 .....	67
图 5-4 DVCCCLDO 输出电压与 VDD 输入电压之间的关系 .....	68
图 5-5 DVCCCLDO 输出电压与负载电流之间的关系 .....	68
图 5-6 VDCIN 管脚输入信号与位 VDCINDROP 的关系 .....	69
图 6-1 PMU 功能框图 .....	72
图 6-2 片外复位的去毛刺时序 .....	77
图 6-3 V85XXP 工作模式转换 .....	80
图 6-4 外部唤醒示意图 .....	82
图 6-5 进入深睡眠的流程 .....	83
图 6-6 进入浅睡眠的流程 .....	84
图 6-7 进入 IDLE 的流程 .....	85
图 7-1 V85XXP 时钟系统功能框图 .....	88
图 8-1 模拟控制器的功能框图 .....	95
图 9-1 ADC 功能框图 .....	121
图 10-1 比较器的迟滞窗口 .....	138
图 10-2 比较器功能框图 .....	139
图 11-1 模拟控制器功能框图 .....	146
图 13-1 RTC 控制器功能框图 .....	152
图 13-2 RTC 手动温度补偿流程图 .....	169
图 14-1 FLASH 控制器的功能框图 .....	171
图 15-1 GPIO 控制器的功能框图 .....	178
图 16-1 DMA 控制器功能框图 .....	189
图 17-1 UART 控制器的功能框图 .....	196
图 18-1 UART32K 控制器的功能框图 .....	202
图 19-1 ISO7816 控制器功能框图 .....	206
图 20-1 定时器功能框图 .....	211
图 20-2 PWM 定时器功能框图 .....	212
图 20-3 PWM 定时器模式 .....	216
图 20-4 连续模式示例 .....	216
图 20-5 向上计数模式下的 PWM 输出 .....	218
图 20-6 连续计数模式下的 PWM 输出 .....	219
图 20-7 向上/向下计数模式下的 PWM 输出 .....	219

图 21-1 LCD 控制器的功能框图 .....	223
图 21-2 LCD 控制器结构图 .....	224
图 21-3 LCD 波形[1/4 Duty, 1/3 Bias] .....	225
图 21-4 LCD 波形[1/6 Duty, 1/3 Bias] .....	226
图 21-5 LCD 波形[1/8 Duty, 1/3 Bias] .....	227
图 21-6 LCD 波形[1/8 Duty, 1/4 Bias] .....	228
图 21-7 不同 FBMODE 下的帧缓存器操作模式 .....	232
图 22-1 SPI 控制器的功能框图 .....	234
图 22-2 主机模式, SPO=0, SPH=0 .....	241
图 22-3 主机模式, SPO=0, SPH=1 .....	242
图 22-4 主机模式, SPO=1, SPH=0 .....	242
图 22-5 主机模式, SPO=1, SPH=1 .....	243
图 23-1 I <sup>2</sup> C 控制器功能框图 .....	244
图 23-2 启动和停止条件 .....	253
图 23-3 7 位地址写模式 .....	253
图 23-4 7 位地址读模式 .....	253
图 23-5 数据传输 .....	254
图 23-6 I <sup>2</sup> C 总线的应答机制 .....	254
图 23-7 I <sup>2</sup> C 总线连接例 .....	255
图 23-8 SCL 和 SDA 时序 .....	255
图 23-9 I <sup>2</sup> C 控制器的 AC 时序 .....	256
图 23-10 主机发送模式的操作流程 .....	257
图 23-11 主机接收模式的操作流程 .....	258
图 23-12 从机发送模式的操作流程 .....	259
图 23-13 从机接收模式的操作流程 .....	260
图 25-1 MISC 控制器的功能框图 .....	264
图 26-1 CRYPT 控制器的功能框图 .....	271

## 表索引

表 1-1 绝对最大额定值 .....	16
表 1-2 供电电压 .....	16
表 1-3 电源管脚的驱动条件.....	16
表 1-4 电源开关导通电阻 .....	16
表 1-5 常用工作条件 .....	17
表 1-6 各个模块的功耗 .....	18
表 1-7 电源和复位控制模块特性 .....	18
表 1-8 电压回差特性 .....	19
表 1-9 GPIO 特性 .....	19
表 1-10 ADC 特性 .....	20
表 1-11 比较器特性 .....	20
表 1-12 时钟和锁相环特性.....	20
表 1-13 FLASH 和 SRAM 特性 .....	21
表 1-14 晶振的 ESR 特性 .....	21
表 1-15 时钟输出稳定时间及唤醒时间 .....	22
表 1-16 TinyADC 转换时间 .....	22
表 1-17 芯片可靠性分析结果 .....	22
表 2-1 V85XXP 管脚描述.....	28
表 4-1 PMU 寄存器地址(PMU 基地址: 0x40014000).....	38
表 4-2 PMU 寄存器地址(PMU 掉电保持 RAM 基地址: 0x40014400) .....	39
表 4-3 模拟控制寄存器地址(ANA 基地址: 0x40014200).....	39
表 4-4 RTC 控制寄存器地址(RTC 基地址: 0x40014800) .....	41
表 4-5 FLASH 控制寄存器地址(FLASH 控制寄存器基地址: 0x00000000) .....	42
表 4-6 GPIO 控制寄存器地址(GPIO 基地址: 0x40000000).....	43
表 4-7 DMA 控制寄存器地址(DMA 基地址: 0x40010000).....	44
表 4-8 UART 寄存器地址(UART 基地址: 0x40011800).....	44
表 4-9 UART32K 寄存器地址(UART32K 基地址: 0x40014100).....	45
表 4-10 ISO7816 寄存器地址(ISO7816 基地址: 0x40012000) .....	46
表 4-11 TIMER 寄存器地址(32 位 TIMER 基地址: 0x40012800).....	46
表 4-12 PWM TIMER 寄存器地址(16 位 PWM TIMER 基地址: 0x40012900) .....	47
表 4-13 LCD 寄存器地址(LCD 基地址: 0x40002000) .....	48
表 4-14 SPI1 寄存器地址(SPI1 基地址: 0x40011000) .....	49
表 4-15 SPI2 寄存器地址(SPI2 基地址: 0x40015800) .....	50
表 4-16 SPI3 寄存器地址(SPI3 基地址: 0x40016000) .....	50
表 4-17 I <sup>2</sup> C 寄存器地址(I <sup>2</sup> C 基地址: 0x40010800).....	50
表 4-18 MISC 寄存器地址(MISC 基地址: 0x40013000) .....	50
表 4-19 MISC2 寄存器地址(MISC2 基地址: 0x40013E00).....	51

表 4-20 CRYPT 寄存器地址(CRYPT 基地址: 0x40006000) .....	51
表 4-21 Info 信息寄存器 .....	51
表 5-1 ANA 寄存器地址(ANA 基地址: 0x40014200) .....	62
表 5-2 ANA_REG5 寄存器 .....	62
表 5-3 ANA_REG6 寄存器 .....	62
表 5-4 ANA_REG7 寄存器 .....	62
表 5-5 ANA_REG8 寄存器 .....	63
表 5-6 ANA_REG9 寄存器 .....	63
表 5-7 ANA_REGA 寄存器 .....	63
表 5-8 ANA_REGF 寄存器 .....	64
表 5-9 ANA_CTRL 寄存器 .....	64
表 5-10 ANA_CMPOUT 寄存器 .....	64
表 5-11 ANA_INTSTS 寄存器 .....	65
表 5-12 ANA_INTEN 寄存器 .....	65
表 5-13 ANA_CMPCTL 寄存器 .....	66
表 6-1 PMU 工作模式寄存器地址(PMU 基地址: 0x40014000) .....	72
表 6-2 PMU RAM Retention 寄存器地址(PMU 掉电保持 RAM 基地址: 0x40014400) .....	72
表 6-3 ANA 控制器寄存器地址(基地址: 0x40014200) .....	72
表 6-4 PMU_DSLEEPEN 寄存器 .....	73
表 6-5 PMU_DSLEPPASS 寄存器 .....	73
表 6-6 PMU_CONTROL 寄存器 .....	73
表 6-7 PMU_STS 寄存器 .....	74
表 6-8 ANA_CTRL 寄存器 .....	75
表 6-9 PMU_RAMx 寄存器 .....	75
表 6-10 各个模块在不同复位及睡眠唤醒后的复位情况 .....	76
表 6-11 在不同模式下的各个时钟的工作状态 .....	76
表 6-12 中断源 .....	78
表 6-13 每一种工作模式下的唤醒源 .....	81
表 7-1 不同工作模式下的时钟模块开关 .....	86
表 7-2 ANA 控制寄存器(ANA 基地址: 0x40014200) .....	88
表 7-3 PMU 寄存器地址(PMU 基地址: 0x40014000) .....	88
表 7-4 MISC2 控制寄存器(MISC2 基地址: 0x40013E00) .....	89
表 7-5 ANA_REG3 控制寄存器 .....	89
表 7-6 ANA_REG4 控制寄存器 .....	89
表 7-7 ANA_REG9 控制寄存器 .....	89
表 7-8 ANA_REGB 控制寄存器 .....	90
表 7-9 ANA_REGC 控制寄存器 .....	90
表 7-10 ANA_CMPOUT 寄存器 .....	91
表 7-11 PMU_CONTROL 寄存器 .....	91

表 7-12 PMU_STS 寄存器 .....	91
表 7-13 MISC2_CLKSEL 寄存器 .....	92
表 7-14 MISC2_CLKDIVH 寄存器 .....	92
表 7-15 MISC2_CLKDIVP 寄存器 .....	92
表 7-16 MISC2_HCLKEN 寄存器 .....	93
表 7-17 HCLK 时钟使能 .....	93
表 7-18 MISC2_PCLKEN 寄存器 .....	93
表 7-19 各个模块的 PCLK 时钟使能 .....	93
表 8-1 ANA 控制寄存器(ANA 基地址: 0x40014200) .....	95
表 8-2 ANA_REGx 表 .....	98
表 8-3 ANA_REG0 各个 bit 的功能描述 .....	98
表 8-4 ANA_REG1 各个 bit 的功能描述 .....	98
表 8-5 ANA_REG2 各个 bit 的功能描述 .....	99
表 8-6 ANA_REG3 各个 bit 的功能描述 .....	99
表 8-7 ANA_REG4 各个 bit 的功能描述 .....	100
表 8-8 ANA_REG5 各个 bit 的功能描述 .....	100
表 8-9 ANA_REG6 各个 bit 的功能描述 .....	100
表 8-10 ANA_REG7 各个 bit 的功能描述 .....	101
表 8-11 ANA_REG8 各个 bit 的功能描述 .....	101
表 8-12 ANA_REG9 各个 bit 的功能描述 .....	102
表 8-13 ANA_REGA 各个 bit 的功能描述 .....	102
表 8-14 ANA_REGB 各个 bit 的功能描述 .....	103
表 8-15 ANA_REGC 各个 bit 的功能描述 .....	103
表 8-16 ANA_REGD 各个 bit 的功能描述 .....	103
表 8-17 ANA_REGE 各个 bit 的功能描述 .....	103
表 8-18 ANA_REGF 各个 bit 的功能描述 .....	103
表 8-19 ANA_REG10 各个 bit 的功能描述 .....	104
表 8-20 ANA_REG11 各个 bit 的功能描述 .....	104
表 8-21 ANA_CTRL 寄存器 .....	104
表 8-22 ANA_CMPOUT 寄存器 .....	106
表 8-23 ANA_ADCSTATE 寄存器 .....	106
表 8-24 ANA_INTSTS 寄存器 .....	107
表 8-25 ANA_INTEN 寄存器 .....	110
表 8-26 ANA_ADCCTRL0 寄存器 .....	112
表 8-27 ANA_ADCDATAx 寄存器 .....	112
表 8-28 ANA_CMPCNTx 寄存器 .....	113
表 8-29 ANA_MISC 寄存器 .....	113
表 8-30 ANA_ADCDOS 寄存器 .....	113
表 8-31 ANA_ADCDCPN 寄存器 .....	113

表 8-32 ANA_ADCDCNM0 寄存器.....	113
表 8-33 ANA_ADCDATADMA 寄存器 .....	114
表 8-34 ANA_ADCCTRL1 寄存器.....	114
表 8-35 ANA_ADCCTRL2 寄存器.....	115
表 8-36 ANA_ADCDATATHD1_0 寄存器 .....	116
表 8-37 ANA_ADCDATATHD3_2 寄存器 .....	116
表 8-38 ANA_ADCDATATHD_CH 寄存器.....	116
表 8-39 ANA_CMPCTL 寄存器.....	117
表 8-40 ANA_CMPTHR 寄存器 .....	119
表 9-1 ANA 控制寄存器(ANA 基地址: 0x40014200) .....	122
表 9-2 RTC 控制器(RTC 基地址: 0x40014800).....	122
表 9-3 ANA_REG0 寄存器 .....	123
表 9-4 ANA_REG1 寄存器.....	123
表 9-5 ANA_REG3 各个 bit 的功能描述.....	123
表 9-6 ANA_ADCSTATE 寄存器 .....	123
表 9-7 ANA_INTSTS 寄存器 .....	124
表 9-8 ANA_INTEN 寄存器.....	125
表 9-9 ANA_ADCCTRL0 寄存器.....	126
表 9-10 ADC 采样通道.....	127
表 9-11 ANA_ADCCTRL1 寄存器.....	127
表 9-12 ANA_ADCCTRL2 寄存器.....	128
表 9-13 ANA_ADCDATATHD1_0 寄存器 .....	129
表 9-14 ANA_ADCDATATHD3_2 寄存器 .....	130
表 9-15 ANA_ADCDATATHD_CH 寄存器.....	130
表 9-16 ANA_ADCDATAx 寄存器.....	131
表 9-17 ANA_ADCDOS 寄存器.....	132
表 9-18 ANA_ADCDCPN 寄存器 .....	132
表 9-19 ANA_ADCDCNM0 寄存器.....	132
表 9-20 ANA_ADCDATADMA 寄存器 .....	132
表 9-21 RTC_ADCUCALK 寄存器 .....	132
表 9-22 RTC_ADCMACTL 寄存器.....	132
表 9-23 RTC_ADCDTCTL 寄存器 .....	133
表 9-24 不同量程 ADC 输入信号的配置范围 .....	135
表 9-25 直流电压、电池电压配置 .....	135
表 9-26 测交流电压配置 .....	135
表 9-27 电压测量量程与公式 .....	135
表 9-28 测温.....	136
表 10-1 ANA 控制寄存器(ANA 基地址: 0x40014200) .....	139
表 10-2 ANA_REG2 寄存器 .....	140

表 10-3 ANA_REG3 寄存器 .....	140
表 10-4 ANA_REG5 寄存器 .....	141
表 10-5 ANA_REGF 寄存器 .....	141
表 10-6 ANA_CTRL 寄存器 .....	141
表 10-7 ANA_CMPOUT 寄存器 .....	142
表 10-8 ANA_INTSTS 寄存器 .....	142
表 10-9 ANA_INTEN 寄存器 .....	143
表 10-10 ANA_CMPCNTx 寄存器 .....	143
表 10-11 ANA_CMPCTL 寄存器 .....	143
表 10-12 ANA_CMPTHR 寄存器 .....	144
表 11-1 ANA 控制器(ANA 基地址: 0x40014200) .....	146
表 11-2 ANA_REGF 寄存器 .....	147
表 11-3 有关 TADC 的 ANA_CMPOUT 寄存器 .....	147
表 11-4 有关 TADC 的 ANA_INTSTS 寄存器 .....	147
表 11-5 ANA_INTEN 寄存器 .....	148
表 11-6 ANA_MISC 寄存器 .....	148
表 12-1 看门狗在不同模式下状态(MODE=1) .....	149
表 12-2 PMU_WDT 控制器(PMU 基地址: 0x40014000) .....	149
表 12-3 PMU_WDTPASS 寄存器 .....	149
表 12-4 PMU_WDTEN 寄存器 .....	150
表 12-5 PMU_WDTCLR 寄存器 .....	150
表 13-1 RTC 控制器(RTC 基地址: 0x40014800) .....	153
表 13-2 RTC_SEC/MIN/DAY/WEEK/MONTH/YEAR 寄存器 .....	154
表 13-3 RTC_TIME 寄存器 .....	155
表 13-4 RTC_WKUSEC 寄存器 .....	155
表 13-5 RTC_WKUMIN 寄存器 .....	155
表 13-6 RTC_WKUHOUR 寄存器 .....	156
表 13-7 RTC_WKUCNT 寄存器 .....	156
表 13-8 RTC_CAL 寄存器 .....	156
表 13-9 RTC_DIV 寄存器 .....	157
表 13-10 RTC_CTL 寄存器 .....	157
表 13-11 RTC_ITV 寄存器 .....	157
表 13-12 RTC_SITV 寄存器 .....	158
表 13-13 RTC_PWD 寄存器 .....	158
表 13-14 RTC_CE 寄存器 .....	158
表 13-15 RTC_LOAD 寄存器 .....	158
表 13-16 RTC_INTSTS 寄存器 .....	159
表 13-17 RTC_INTEN 寄存器 .....	159
表 13-18 RTC_PSCA 寄存器 .....	160

表 13-19 RTC_ACTI 寄存器.....	160
表 13-20 RTC_ACF200 寄存器.....	160
表 13-21 RTC_ACP0 寄存器.....	161
表 13-22 RTC_ACP1 寄存器.....	161
表 13-23 RTC_ACP2 寄存器.....	161
表 13-24 RTC_ACP3 寄存器.....	161
表 13-25 RTC_ACP4 寄存器.....	161
表 13-26 RTC_ACP5 寄存器.....	162
表 13-27 RTC_ACP6 寄存器.....	162
表 13-28 RTC_ACP7 寄存器.....	162
表 13-29 RTC_ACKx 寄存器.....	162
表 13-30 RTC_WKUCNTR 寄存器.....	163
表 13-31 RTC_ACKTEMP 寄存器.....	163
表 13-32 RTC_ALARMSEC 寄存器.....	163
表 13-33 RTC_ALARMMIN 寄存器.....	163
表 13-34 RTC_ALARMHOU 寄存器.....	164
表 13-35 RTC_ALARMCTL 寄存器.....	164
表 13-36 RTC_ADCUCALK 寄存器.....	164
表 13-37 RTC_ADCMACTL 寄存器.....	164
表 13-38 RTC_ADCDTCTL 寄存器.....	165
表 13-39 Info 信息寄存器(与 RTC 温度补偿有关).....	165
表 14-1 FLASH 控制器地址(FLASH 控制器基地址: 0x00000000).....	171
表 14-2 MISC2 控制器(MISC2 基地址: 0x40013E00).....	172
表 14-3 FLASH_PASS 寄存器.....	172
表 14-4 FLASH_CTRL 寄存器.....	172
表 14-5 FLASH_PGADDR 寄存器.....	172
表 14-6 PGADDR 及写数据寄存器.....	173
表 14-7 FLASH_PGDATA 寄存器.....	173
表 14-8 FLASH_PGBx 寄存器.....	173
表 14-9 FLASH_PGHwX 寄存器.....	173
表 14-10 FLASH_SERASE 寄存器.....	174
表 14-11 FLASH_CERASE 寄存器.....	174
表 14-12 FLASH_DSTB 寄存器.....	174
表 14-13 FLASH_INTSTS 寄存器.....	175
表 14-14 FLASH_CSSADDR 寄存器.....	175
表 14-15 FLASH_CSEADDR 寄存器.....	175
表 14-16 FLASH_CSVALUE 寄存器.....	175
表 14-17 FLASH_CSCVALUE 寄存器.....	175
表 14-18 FLASH_STS 寄存器.....	176

表 14-19 MISC2_FLASHWC 寄存器 .....	176
表 14-20 FLASH_RDPROT 寄存器 .....	176
表 14-21 FLASH_WRPROT 寄存器 .....	177
表 14-22 FLASH_ICEPROT 寄存器 .....	177
表 14-23 FLASH 保护说明 .....	177
表 15-1 PMU_IOA 控制器(PMU 基地址: 0x40014000) .....	178
表 15-2 GPIO 控制器(GPIO 基地址: 0x40000000) .....	179
表 15-3 PMU_IOAOEN 寄存器 .....	180
表 15-4 PMU_IOAIE 寄存器 .....	180
表 15-5 PMU_IOADAT 寄存器 .....	180
表 15-6 PMU_IOAATT 寄存器 .....	180
表 15-7 不同设置下的 IO 状态 .....	180
表 15-8 PMU_IOAWKUEN 寄存器 .....	181
表 15-9 IO 唤醒模式 .....	181
表 15-10 PMU_IOASTS 寄存器 .....	181
表 15-11 PMU_IOAINTSTS 寄存器 .....	182
表 15-12 PMU_IOASEL 寄存器 .....	182
表 15-13 PMU_IOANODEGL 寄存器 .....	182
表 15-14 IOX_OEN 寄存器 .....	182
表 15-15 IOX_IE 寄存器 .....	183
表 15-16 IOX_DAT 寄存器 .....	183
表 15-17 IOX_ATT 寄存器 .....	183
表 15-18 IOX_STS 寄存器 .....	183
表 15-19 IOB_SEL 寄存器 .....	184
表 15-20 IOE_SEL 寄存器 .....	184
表 15-21 IO_MISC 寄存器 .....	184
表 15-22 特殊功能 IOA .....	185
表 15-23 特殊功能 IOB .....	185
表 15-24 特殊功能 IOC .....	186
表 15-25 特殊功能 IOD .....	187
表 15-26 特殊功能 IOE .....	187
表 15-27 特殊功能 IOF .....	188
表 16-1 DMA 控制器(DMA 基地址: 0x40010000) .....	189
表 16-2 DMA_IE 寄存器 .....	190
表 16-3 DMA_STS 寄存器 .....	191
表 16-4 DMA_CxCTL 寄存器 .....	192
表 16-5 DMA 请求源选择 .....	193
表 16-6 DMA_CxSRC 寄存器 .....	194
表 16-7 DMA_CxDST 寄存器 .....	194

表 16-8 DMA_CxLEN 寄存器 .....	194
表 16-9 DMA_AESCTL 寄存器.....	194
表 16-10 DMA_AESKEYx 寄存器.....	195
表 17-1 UART 控制器(UART 基地址: 0x40011800).....	196
表 17-2 MISC1 控制器(MISC1 基地址: 0x40013000) .....	197
表 17-3 UARTx_DATA 寄存器 .....	198
表 17-4 UARTx_STATE 寄存器.....	198
表 17-5 UARTx_CTRL 寄存器 .....	199
表 17-6 UARTx_INTSTS 寄存器 .....	199
表 17-7 UARTx_BAUDDIV 寄存器 .....	199
表 17-8 UARTx_CTRL2 寄存器.....	200
表 17-9 MISC1_IREN 寄存器.....	200
表 17-10 MISC1_DUTYL 寄存器.....	200
表 17-11 MISC1_DUTYH 寄存器 .....	200
表 18-1 UART32K 控制器(UART32K0 基地址: 0x40014100) .....	202
表 18-2 U32Kx_CTRL0 寄存器.....	203
表 18-3 U32Kx_CTRL1 寄存器.....	204
表 18-4 U32Kx_BAUDDIV 寄存器 .....	204
表 18-5 U32Kx_DATA 寄存器 .....	205
表 18-6 U32Kx_STS 寄存器 .....	205
表 19-1 ISO7816 控制器地址(ISO7816 基地址: 0x40012000) .....	206
表 19-2 ISO7816x_BAUDDIVL 寄存器 .....	207
表 19-3 ISO7816x_BAUDDIVH 寄存器.....	207
表 19-4 ISO7816x_DATA 寄存器 .....	207
表 19-5 ISO7816x_INFO 寄存器.....	208
表 19-6 ISO7816x_CFG 寄存器 .....	208
表 19-7 ISO7816x_CLK 寄存器 .....	209
表 19-8 ISO7816 数据包格式.....	210
表 20-1 32 位 TIMER 控制器(32 位 TIMER 基地址: 0x40012800) .....	212
表 20-2 16 位 PWM TIMER 控制器(16 位 PWM TIMER 基地址: 0x40012900).....	212
表 20-3 TMRx_CTRL 寄存器 .....	214
表 20-4 TMRx_VALUE 寄存器 .....	214
表 20-5 TMRx_RELOAD 寄存器.....	214
表 20-6 TMRx_INTSTS 寄存器.....	214
表 20-7 PWMx_CTL 寄存器 .....	214
表 20-8 PWMx_TAR 寄存器.....	217
表 20-9 PWMx_CCTLx 寄存器.....	217
表 20-10 PWMx_CCRx 寄存器 .....	219
表 20-11 PWM_O_SEL 寄存器 .....	220

表 20-12 PWM_I_SEL01 寄存器.....	220
表 20-13 PWM_I_SEL23 寄存器.....	221
表 21-1 与 LCD 相关的 ANA 控制器(ANA 基地址: 0x40014200) .....	228
表 21-2 LCD 控制器(LCD 基地址: 0x40002000) .....	228
表 21-3 LCD_FBx 寄存器 .....	230
表 21-4 LCD_FBx 寄存器的数据排布 .....	230
表 21-5 LCD_CTRL 寄存器 .....	230
表 21-6 LCD_CTRL2 寄存器 .....	231
表 21-7 LCD_SEGCTRL0 寄存器 .....	232
表 21-8 LCD_SEGCTRL1 寄存器 .....	232
表 21-9 LCD_SEGCTRL2 寄存器 .....	233
表 21-10 ANA_REG6 寄存器 .....	233
表 22-1 SPI1 控制器(SPI1 基地址: 0x40011000).....	235
表 22-2 SPI2 控制器(SPI2 基地址: 0x40015800).....	235
表 22-3 SPI3 控制器(SPI3 基地址: 0x40016000).....	235
表 22-4 SPIx_CTRL 寄存器 .....	235
表 22-5 SPIx_TXSTS 寄存器 .....	237
表 22-6 SPIx_TXDAT 寄存器 .....	238
表 22-7 SPIx_RXSTS 寄存器 .....	238
表 22-8 SPIx_RXDAT 寄存器.....	240
表 22-9 SPIx_MISC 寄存器 .....	240
表 23-1 I <sup>2</sup> C 控制器(I <sup>2</sup> C 基地址: 0x40010800) .....	244
表 23-2 I2C_DATA 寄存器.....	246
表 23-3 I2C_ADDR 寄存器.....	246
表 23-4 I2C_CTRL 寄存器.....	246
表 23-5 SCL 频率设置 .....	247
表 23-6 I2C_STS 寄存器 .....	247
表 23-7 主机发送模式下的 I <sup>2</sup> C 状态 .....	247
表 23-8 主机接收模式下的 I <sup>2</sup> C 状态 .....	248
表 23-9 从机接收模式下的 I <sup>2</sup> C 状态 .....	249
表 23-10 从机发送模式下的 I <sup>2</sup> C 状态.....	250
表 23-11 I <sup>2</sup> C 状态 .....	251
表 23-12 I2C_CTRL2 寄存器 .....	251
表 23-13 I <sup>2</sup> C AC 特性 .....	255
表 24-1 中断源 .....	261
表 25-1 MISC1 控制器(MISC1 基地址: 0x40013000) .....	264
表 25-2 MISC2 控制器(MISC2 基地址: 0x40013E00) .....	265
表 25-3 MISC1_SRAMINT 寄存器.....	265
表 25-4 MISC1_SRAMINIT 寄存器 .....	265

表 25-5 MISC1_PARERR 寄存器.....	266
表 25-6 MISC1_IREN 寄存器.....	266
表 25-7 MISC1_DUTYL 寄存器 .....	266
表 25-8 MISC1_DUTYH 寄存器 .....	266
表 25-9 MISC1_IRQLAT 寄存器 .....	267
表 25-10 MISC1_HIADDR 寄存器 .....	267
表 25-11 MISC1_PIADDR 寄存器.....	267
表 25-12 MISC2_FLASHWC 寄存器 .....	267
表 25-13 MISC2_CLKSEL 寄存器 .....	268
表 25-14 MISC2_CLKDIVH 寄存器 .....	268
表 25-15 MISC2_CLKDIVP 寄存器.....	269
表 25-16 MISC2_HCLKEN 寄存器.....	269
表 25-17 HCLK 时钟使能 .....	269
表 25-18 MISC2_PCLKEN 寄存器.....	269
表 25-19 PCLK 时钟使能 .....	270
表 26-1 CRYPT 控制器(CRYPT 基地址: 0x40006000).....	271
表 26-2 CRYPT_CTRL 寄存器.....	271
表 26-3 CRYPT_PTRA 寄存器.....	272
表 26-4 CRYPT_PTRB 寄存器.....	272
表 26-5 CRYPT_PTRO 寄存器 .....	273
表 26-6 CRYPT_CARRY 寄存器 .....	273
表 27-1 SWD 引脚的特殊功能.....	275
表 28-1 CMSIS 函数 .....	276

## 版本更新说明

时间	版本	说明
2020.06.22	V2.0	初版发布。
2020.10.17	V2.1	增加 LQFP100(V8500P)/LQFP80(V8530P)/LQFP64(V8510P)。
2021.07.16	V2.2	修改表 1-10 ADC 特性，工作时间和采样速率典型值。
2022.01.04	V2.3	修改 ANA_REG4<2:0>=001；ANA_REG10 不需要配置；
2022.02.17	V2.4	增加补充 VDCIN 及 VDCINHYST 描述； 修正 V85XXP 时钟功能框图 ADC 时钟源选择寄存器。
2022.04.14	V2.5	修改 RTC_ALARMCTL 寄存器 Bit 1 功能描述； 修改 V85XXP 管脚描述中 X32KI 和 X32KO 描述； 删除 RTC_ALARMTIME 寄存器。
2022.05.09	V2.6	修正 8.5.12 ANA_REGA 寄存器 Bit[6:0]描述；
2022.12.02	V2.7	修改公式 9-1；
2023.03.03	V2.8	修正 IOA4~IOA7 对应 SEG 管脚号； 增加 V8531P；