



杭州万高科技股份有限公司

V85XX 数据手册



目录

1	电气特性	38
1.1	绝对最大额定值	38
1.2	正常工作电压	38
1.3	驱动特性	38
1.4	电源开关导通电阻	38
1.5	功耗	39
1.6	复位和电源控制模块特性	40
1.7	GPIO 特性	40
1.8	ADC 特性	41
1.9	模拟比较器特性	41
1.10	时钟和锁相环特性	42
1.11	FLASH 和 SRAM 特性	43
1.12	晶振的 ESR 特性	43
1.13	时钟输出稳定时间及唤醒时间	43
1.14	ADC 转换时间	44
1.15	芯片可靠性分析结果	45
2	引脚分布图	46
2.1	V8500 引脚分布图	46
2.2	V8531 引脚分布图	47
2.3	V8510 引脚分布图	48
2.4	V85XX 管脚描述	49
3	功能框图	56
3.1	系统功能框图	56
3.2	电源功能框图	57
3.3	时钟功能框图	58
4	内存映射	59
4.1	寄存器地址	60
4.1.1	PMU 寄存器地址	60
4.1.2	ANA 寄存器地址	60
4.1.3	RTC 寄存器地址	62

4.1.4	FLASH 寄存器地址	63
4.1.5	GPIO 寄存器地址.....	64
4.1.6	DMA 寄存器地址.....	65
4.1.7	UART 寄存器地址.....	66
4.1.8	UART32K 寄存器地址.....	67
4.1.9	ISO7816 寄存器地址.....	67
4.1.10	TIMER/PWM 寄存器地址	67
4.1.11	LCD 寄存器	69
4.1.12	SPI 寄存器地址.....	70
4.1.13	I2C 寄存器地址.....	71
4.1.14	MISC 寄存器地址.....	71
4.1.15	CRYPT 寄存器地址.....	72
4.1.16	EPU 寄存器地址	72
4.1.17	RNG 寄存器地址.....	73
4.1.18	Info 参数寄存器.....	73
5	电源系统.....	82
5.1	寄存器地址	84
5.2	寄存器定义	84
5.2.1	ANA_REG5 寄存器.....	84
5.2.2	ANA_REG6 寄存器.....	84
5.2.3	ANA_REG7 寄存器.....	85
5.2.4	ANA_REG8 寄存器.....	85
5.2.5	ANA_REG9 寄存器.....	85
5.2.6	ANA_REGA 寄存器	85
5.2.7	ANA_REGF 寄存器.....	86
5.2.8	ANA_CMPOUT 寄存器	86
5.2.9	ANA_INSTS 寄存器.....	86
5.2.10	ANA_INTEN 寄存器.....	87
5.3	AVCC.....	88
5.4	DVCC	89
5.5	电源监测	90
5.5.1	VDCIN 电源检测	90

5.5.2	VDD 电源监测	91
5.5.3	AVCC 电源监测	91
5.6	电源切换开关	91
5.6.1	电源切换开关 2.....	91
6	工作模式	92
6.1	简介	92
6.2	特点	92
6.3	功能框图	92
6.4	寄存器地址	93
6.5	寄存器定义	93
6.5.1	PMU_CONTROL 寄存器.....	93
6.5.2	PMU_STS 寄存器.....	94
6.5.3	PMU_WDTSTS 寄存器.....	95
6.6	复位	95
6.6.1	片外复位.....	96
6.6.2	看门狗复位	97
6.6.3	上电复位.....	97
6.6.4	M0 软复位	97
6.6.5	从睡眠/IDLE 下唤醒	97
6.7	工作模式的转换	101
6.8	应用笔记	102
6.8.1	外部 IO 唤醒	103
6.8.2	进入睡眠的流程	104
6.8.3	进入 IDLE 的流程.....	105
7	时钟系统	106
7.1	简介	106
7.2	特点	107
7.3	功能框图	107
7.4	寄存器地址	108
7.5	寄存器定义	109
7.5.1	ANA_REG3 寄存器.....	109
7.5.2	ANA_REG4 寄存器.....	109

7.5.3	ANA_REG9 寄存器.....	110
7.5.4	ANA_REGB 寄存器	110
7.5.5	ANA_REGC 寄存器	111
7.5.6	ANA_REG10 寄存器.....	111
7.5.7	ANA_REG11 寄存器.....	111
7.5.8	ANA_CMPOUT 寄存器	112
7.5.9	PMU_CONTROL 寄存器.....	112
7.5.10	PMU_STS 寄存器.....	112
7.5.11	MISC2_CLKSEL 寄存器	113
7.5.12	MISC2_CLKDIVH 寄存器.....	113
7.5.13	MISC2_CLKDIVP 寄存器.....	114
7.5.14	MISC2_HCLKEN 寄存器.....	114
7.5.15	MISC2_PCLKEN 寄存器.....	114
7.5.16	MISC2_EPUCLKCTRL 寄存器.....	115
8	模拟控制器.....	117
8.1	简介	117
8.2	特点	117
8.3	功能框图	117
8.4	寄存器地址	117
8.5	寄存器定义.....	118
8.5.1	ANA_REGx 寄存器	119
8.5.2	ANA_REG0 寄存器.....	120
8.5.3	ANA_REG1 寄存器.....	120
8.5.4	ANA_REG2 寄存器.....	120
8.5.5	ANA_REG3 寄存器.....	121
8.5.6	ANA_REG4 寄存器.....	121
8.5.7	ANA_REG5 寄存器.....	122
8.5.8	ANA_REG6 寄存器.....	122
8.5.9	ANA_REG7 寄存器.....	122
8.5.10	ANA_REG8 寄存器.....	123
8.5.11	ANA_REG9 寄存器.....	123
8.5.12	ANA_REGA 寄存器	124

	8.5.13	ANA_REGB 寄存器	124
	8.5.14	ANA_REGC 寄存器	125
	8.5.15	ANA_REGD 寄存器	125
	8.5.16	ANA_REGE 寄存器	125
	8.5.17	ANA_REGF 寄存器	126
	8.5.18	ANA_REG10 寄存器	126
	8.5.19	ANA_REG11 寄存器	126
	8.5.20	ANA_CTRL 寄存器	127
	8.5.21	ANA_CMPOUT 寄存器	128
	8.5.22	ANA_INSTS 寄存器	129
	8.5.23	ANA_INTEN 寄存器	130
	8.5.24	ANA_ADCCTRL 寄存器	131
	8.5.25	ANA_ADCDATAx 寄存器	133
	8.5.26	ANA_CMPCNTx 寄存器	133
9	ADC	134	
	9.1	简介	134
	9.2	特点	134
	9.3	ADC 功能框图	134
	9.4	寄存器地址	134
	9.5	寄存器定义	135
	9.5.1	ANA_REG1 寄存器	135
	9.5.2	ANA_REG3 寄存器	135
	9.5.3	ANA_INSTS 寄存器	136
	9.5.4	ANA_INTEN 寄存器	136
	9.5.5	ANA_ADCCTRL 寄存器	137
	9.5.6	ANA_ADCDATAx 寄存器	139
	9.6	电池电压测量和外部电压测量	139
	9.7	温度测量	141
	9.8	ADC 注意事项	142
10	比较器控制器	143	
	10.1	简介	143
	10.2	特点	143

10.3	功能框图	143
10.4	寄存器地址	144
10.5	寄存器定义	144
10.5.1	ANA_REG2 寄存器	144
10.5.2	ANA_REG3 寄存器	145
10.5.3	ANA_REG5 寄存器	145
10.5.4	ANA_REGF 寄存器	145
10.5.5	ANA_CTRL 寄存器	146
10.5.6	ANA_CMPOUT 寄存器	147
10.5.7	ANA_INSTS 寄存器	147
10.5.8	ANA_INTEN 寄存器	147
10.5.9	ANA_CMPCNTx 寄存器	147
11	看门狗	149
11.1	简介	149
11.2	特点	149
11.3	看门狗在不同模式下的状态	149
11.4	寄存器地址	149
11.5	寄存器定义	150
11.5.1	PMU_WDTPASS 寄存器	150
11.5.2	PMU_WDTEN 寄存器	150
11.5.3	PMU_WDTCLR 寄存器	150
11.5.4	PMU_WDTSTS 寄存器	151
12	RTC 控制器	152
12.1	简介	152
12.2	特点	152
12.3	功能框图	152
12.4	寄存器读写	152
12.4.1	RTC 寄存器写操作	153
12.4.2	RTC 寄存器读操作	153
12.5	寄存器地址	153
12.6	寄存器定义	155
12.6.1	RTC_SEC/MIN/DAY/WEEK/MONTH/YEAR 寄存器	155

12.6.2	RTC_WKUSEC 寄存器	155
12.6.3	RTC_WKUMIN 寄存器	156
12.6.4	RTC_WKUHOURL 寄存器	156
12.6.5	RTC_WKUCNT 寄存器	156
12.6.6	RTC_CAL 寄存器	157
12.6.7	RTC_DIV 寄存器	157
12.6.8	RTC_CTL 寄存器	158
12.6.9	RTC_PWD 寄存器	158
12.6.10	RTC_CE 寄存器	158
12.6.11	RTC_LOAD 寄存器	159
12.6.12	RTC_INTSTS 寄存器	159
12.6.13	RTC_INTEN 寄存器	160
12.6.14	RTC_PSCA 寄存器	160
12.6.15	RTC_ACTI 寄存器	161
12.6.16	RTC_ACF200 寄存器	161
12.6.17	RTC_ACADCW 寄存器	161
12.6.18	RTC_ACP0 寄存器	161
12.6.19	RTC_ACP1 寄存器	162
12.6.20	RTC_ACP2 寄存器	162
12.6.21	RTC_ACP3 寄存器	162
12.6.22	RTC_ACP4 寄存器	162
12.6.23	RTC_ACP5 寄存器	163
12.6.24	RTC_ACP6 寄存器	163
12.6.25	RTC_ACP7 寄存器	163
12.6.26	RTC_ACKx 寄存器	163
12.6.27	RTC_ACKTEMP 寄存器	164
12.6.28	RTC_ACTEMP 寄存器	164
12.6.29	RTC_ACPPM 寄存器	165
12.6.30	RTC_WKUCNTR 寄存器	165
12.7	Info 参数寄存器（与 RTC 温度补偿有关）	165
12.8	应用笔记	167
12.8.1	温度校正	168

	12.8.2	频率误差校正.....	168
	12.8.3	RTC 手动温度补偿.....	168
	12.8.4	RTC 校正注意事项.....	169
13		FLASH 控制器.....	170
	13.1	简介	170
	13.2	特点	170
	13.3	功能框图	170
	13.4	寄存器地址	171
	13.5	寄存器定义	172
	13.5.1	FLASH_PASS 寄存器.....	172
	13.5.2	FLASH_CTRL 寄存器	172
	13.5.3	FLASH_PGADDR 寄存器	172
	13.5.4	FLASH_PGDATA 寄存器.....	173
	13.5.5	FLASH_PGBx 寄存器	173
	13.5.6	FLASH_PGHwX 寄存器.....	174
	13.5.7	FLASH_SERASE 寄存器	174
	13.5.8	FLASH_CERASE 寄存器.....	174
	13.5.9	FLASH_DSTB 寄存器.....	174
	13.5.10	FLASH_INT 寄存器.....	175
	13.5.11	FLASH_CSSADDR 寄存器	175
	13.5.12	FLASH_CSEADDR 寄存器	175
	13.5.13	FLASH_CSVALUE 寄存器	176
	13.5.14	FLASH_CSCVALUE 寄存器	176
	13.5.15	FLASH_STS 寄存器	176
	13.5.16	MISC2_FLASHWC 寄存器.....	176
14		GPIO 控制器.....	178
	14.1	简介	178
	14.2	特点	178
	14.3	功能框图	178
	14.4	寄存器地址	178
	14.5	寄存器定义	180
	14.5.1	PMU_IOAOEN 寄存器	180

14.5.2	PMU_IOAIE 寄存器	180
14.5.3	PMU_IOADAT 寄存器.....	180
14.5.4	PMU_IOAATT 寄存器	180
14.5.5	PMU_IOAWKUEN 寄存器.....	181
14.5.6	PMU_IOASTS 寄存器.....	182
14.5.7	PMU_IOAINTSTS 寄存器	182
14.5.8	PMU_IOASEL 寄存器	182
14.5.9	PMU_IOANODEG 寄存器.....	183
14.5.10	IOX_OEN 寄存器.....	183
14.5.11	IOX_IE 寄存器.....	183
14.5.12	IOX_DAT 寄存器	183
14.5.13	IOX_ATT 寄存器.....	184
14.5.14	IOX_STS 寄存器.....	184
14.5.15	IOB_SEL 寄存器.....	184
14.5.16	IOE_SEL 寄存器.....	185
14.5.17	IO_MISC 寄存器.....	185
14.6	特殊功能 IO.....	186
14.6.1	特殊功能 IOA.....	186
14.6.2	特殊功能 IOB.....	186
14.6.3	特殊功能 IOC.....	187
14.6.4	特殊功能 IOD.....	188
14.6.5	特殊功能 IOE.....	188
14.6.6	特殊功能 IOF	189
15	DMA 控制器.....	190
15.1	简介	190
15.2	特点	190
15.3	功能框图	190
15.4	寄存器地址	191
15.5	寄存器定义.....	192
15.5.1	DMA_IE 寄存器	192
15.5.2	DMA_STS 寄存器.....	193
15.5.3	DMA_CxCTL 寄存器.....	194

	15.5.4	DMA_CxSRC 寄存器	196
	15.5.5	DMA_CxDST 寄存器	196
	15.5.6	DMA_CxLEN 寄存器.....	197
	15.5.7	DMA_AESCTL 寄存器	197
	15.5.8	DMA_AESKEYx 寄存器	197
16	UART 控制器	199	
	16.1	简介	199
	16.2	特点	199
	16.3	功能框图	199
	16.4	寄存器地址	200
	16.5	寄存器定义	201
	16.5.1	UARTx_DATA 寄存器	201
	16.5.2	UARTx_STATE 寄存器	201
	16.5.3	UARTx_CTRL 寄存器	202
	16.5.4	UARTx_INTSTS 寄存器	202
	16.5.5	UARTx_BAUDDIV 寄存器	203
	16.5.6	UARTx_CTRL2 寄存器	203
	16.5.7	MISC_IREN 寄存器	203
	16.5.8	MISC_DUTYL 寄存器	204
	16.5.9	MISC_DUTYH 寄存器	204
17	UART32K 控制器	205	
	17.1	简介	205
	17.2	特点	205
	17.3	功能框图	205
	17.4	寄存器地址	206
	17.5	寄存器定义	206
	17.5.1	U32Kx_CTRL0 寄存器	206
	17.5.2	U32Kx_CTRL1 寄存器	207
	17.5.3	U32Kx_PHASE 寄存器	207
	17.5.4	U32Kx_DATA 寄存器	208
	17.5.5	U32Kx_STS 寄存器	208
18	ISO7816 控制器	209	
	18.1	简介	209

18.2	特点	209
18.3	功能框图	209
18.4	寄存器地址	209
18.5	寄存器定义	210
18.5.1	ISO7816x_BAUDDIVL 寄存器	210
18.5.2	ISO7816x_BAUDDIVH 寄存器	210
18.5.3	ISO7816x_DATA 寄存器	210
18.5.4	ISO7816x_INFO 寄存器	211
18.5.5	ISO7816x_CFG 寄存器	211
18.5.6	ISO7816x_CLK 寄存器	212
19	Timer/PWM 控制器	214
19.1	简介	214
19.2	特点	214
19.3	功能框图	214
19.4	寄存器地址	215
19.5	寄存器定义	216
19.5.1	TMRx_CTRL 寄存器	217
19.5.2	TMRx_VALUE 寄存器	217
19.5.3	TMRx_RELOAD 寄存器	217
19.5.4	TMRx_INT 寄存器	217
19.5.5	PWMx_CTL 寄存器	218
19.5.6	PWMx_TAR 寄存器	220
19.5.7	PWMx_CCTLx 寄存器	220
19.5.8	PWMx_CCRx 寄存器	223
19.5.9	PWM_O_SEL 寄存器	223
20	LCD 控制器	225
20.1	简介	225
20.2	特点	225
20.3	功能框图	225
20.4	LCD 时序	226
20.5	LCD 波形电压	226
20.6	LCD 工作电流	226

20.7	LCD 驱动波形.....	226
20.8	寄存器地址.....	231
20.9	寄存器定义.....	232
20.9.1	LCD_FBx 寄存器.....	232
20.9.2	LCD_CTRL 寄存器.....	233
20.9.3	LCD_CTRL2 寄存器.....	233
20.9.4	LCD_SEGCTRL0 寄存器.....	234
20.9.5	LCD_SEGCTRL1 寄存器.....	235
20.9.6	LCD_SEGCTRL2 寄存器.....	235
20.9.7	ANA_REG6 寄存器.....	235
21	SPI 控制器.....	236
21.1	简介.....	236
21.2	特点.....	236
21.3	功能框图.....	236
21.4	寄存器地址.....	237
21.5	寄存器定义.....	238
21.5.1	SPIx_CTRL 寄存器.....	238
21.5.2	SPIx_TXSTS 寄存器.....	239
21.5.3	SPIx_TXDAT 寄存器.....	241
21.5.4	SPIx_RXSTS 寄存器.....	241
21.5.5	SPIx_RXDAT 寄存器.....	242
21.5.6	SPIx_MISC 寄存器.....	243
21.6	应用笔记.....	244
21.6.1	主机模式.....	244
21.6.2	从机模式.....	246
21.6.3	连续字节传输.....	246
22	I2C 控制器.....	247
22.1	简介.....	247
22.2	特点.....	247
22.3	功能框图.....	247
22.4	寄存器地址.....	248
22.5	寄存器定义.....	248

22.5.1	I2C_DATA 寄存器.....	248
22.5.2	I2C_ADDR 寄存器.....	249
22.5.3	I2C_CTRL 寄存器.....	249
22.5.4	I2C_STS 寄存器.....	250
22.5.5	I2C_CTRL2 寄存器.....	256
22.6	应用笔记	256
22.6.1	I2C 总线协议：启动/停止的发生.....	256
22.6.2	数据传输格式.....	257
22.6.3	ACK 信号传输.....	258
22.6.4	读写操作.....	258
22.6.5	接口时序.....	259
22.6.6	I2C SDA 管脚死锁的解决方法	261
22.7	工作流程	261
22.7.1	主机发送模式.....	262
22.7.2	主机接收模式.....	263
22.7.3	从机发送模式.....	264
22.7.4	从机接收模式.....	265
23	中断控制器.....	266
23.1	简介	266
23.2	中断源.....	266
24	MISC 控制器	270
24.1	简介	270
24.2	特点	270
24.3	功能框图	270
24.4	寄存器地址.....	270
24.5	寄存器定义.....	271
24.5.1	MISC_SRAMINT 寄存器.....	271
24.5.2	MISC_SRAMINIT 寄存器.....	272
24.5.3	MISC_PARERR 寄存器.....	272
24.5.4	MISC_IREN 寄存器	272
24.5.5	MISC_DUTYL 寄存器	272
24.5.6	MISC_DUTYH 寄存器.....	273

	24.5.7	MISC_IRQLAT 寄存器	273
	24.5.8	MISC_HIADDR 寄存器	273
	24.5.9	MISC_PIADDR 寄存器	274
	24.5.10	MISC2_FLASHWC 寄存器	274
	24.5.11	MISC2_CLKSEL 寄存器	274
	24.5.12	MISC2_CLKDIVH 寄存器	275
	24.5.13	MISC2_CLKDIVP 寄存器	275
	24.5.14	MISC2_HCLKEN 寄存器	275
	24.5.15	MISC2_PCLKEN 寄存器	276
	24.5.16	MISC2_EPUCLKCTRL 寄存器	277
	24.5.17	MISC2_RSTEPU 寄存器	277
25		CRYPT 控制器	278
	25.1	简介	278
	25.2	特点	278
	25.3	功能框图	278
	25.4	寄存器地址	278
	25.5	寄存器定义	279
	25.5.1	CRYPT_CTRL 寄存器	279
	25.5.2	CRYPT_PTRA 寄存器	280
	25.5.3	CRYPT_PTRB 寄存器	280
	25.5.4	CRYPT_PTRO 寄存器	281
	25.5.5	CRYPT_CARRY 寄存器	281
	25.5.6	CRYPT_PRIMEn (n=0~7) 寄存器	281
	25.6	使用时的注意事项	282
	25.6.1	数据格式	282
	25.6.2	运算细节	282
26		RNG 控制器	284
	26.1	简介	284
	26.2	特点	284
	26.3	寄存器地址	284
	26.4	寄存器定义	284
	26.4.1	RNG_IMR 寄存器	284

	26.4.2	RNG_ISR 寄存器.....	285
	26.4.3	RNG_ICR 寄存器.....	285
	26.4.4	RNG_CFG 寄存器	286
	26.4.5	RNG_DATA0 寄存器	286
	26.4.6	RNG_DATA1 寄存器	286
	26.4.7	RNG_DATA2 寄存器	286
	26.4.8	RNG_DATA3 寄存器	287
	26.4.9	RNG_DATA4 寄存器	287
	26.4.10	RNG_DATA5 寄存器	287
	26.4.11	RNG_CMD 寄存器.....	287
	26.4.12	RNG_CNT 寄存器	287
	26.4.13	RNG_BYBASS 寄存器.....	287
27		EPU 控制器.....	289
	27.1	简介	289
	27.2	特点	289
	27.3	功能框图	289
	27.4	寄存器地址.....	290
	27.5	寄存器定义.....	291
	27.5.1	EPU_CFCTRL 寄存器	291
	27.5.2	EPU_HSCTRL 寄存器	292
	27.5.3	EPU_LSCTRL 寄存器.....	293
	27.5.4	EPU_CLR 寄存器	294
	27.5.5	EGY_CRPTH 寄存器	295
	27.5.6	EGY_PWRTH0 寄存器	295
	27.5.7	EGY_PWRTH1 寄存器	295
	27.5.8	EPU_INTEN 寄存器.....	295
	27.5.9	EPU_INTSTS 寄存器.....	296
	27.5.10	EPU_STS 寄存器	296
	27.5.11	EPU_CONST0 寄存器	296
	27.5.12	EPU_OUTL0 寄存器.....	296
	27.5.13	EPU_OUTH0 寄存器	297
	27.5.14	EPU_CFCNT0 寄存器	297

	27.5.15 EPU_CONST1 寄存器	297
	27.5.16 EPU_OUTL1 寄存器	297
	27.5.17 EPU_OUTH1 寄存器	297
	27.5.18 EPU_CFCNT1 寄存器	297
	27.5.19 EPU_CONSTx (x=2-7) 寄存器.....	298
	27.5.20 EPU_OUTx (x=2-7) 寄存器.....	298
	27.5.21 EPU_CFCNTx (x=2-7) 寄存器	298
28	调试特点	299
	28.1 特点	299
	28.2 SWD 引脚	299
29	Cortex-M0 Core 简要描述	300
	29.1 CMSIS 函数说明.....	300
30	封装尺寸图.....	301
	30.1 V8500 封装尺寸图	301
	30.2 V8531 封装尺寸图	302
	30.3 V8510 封装尺寸图	303

表目录

表 1.	文档版本历史	34
表 2.	绝对最大额定值.....	38
表 3.	供电电压	38
表 4.	电源管脚的驱动条件.....	38
表 5.	电源开关导通电阻	39
表 6.	常用工作条件	39
表 7.	电源和复位控制模块特性.....	40
表 8.	电压回差特性	40
表 9.	GPIO 特性.....	40
表 10.	ADC 特性.....	41
表 11.	比较器特性.....	41
表 12.	时钟和锁相环特性	42
表 13.	FLASH 和 SRAM 特性	43
表 14.	晶振的 ESR 特性.....	43
表 15.	时钟输出稳定时间及唤醒时间.....	43
表 16.	ADC 转换时间（CIC 滤波器降采样率与 ADC 时钟频率）	44
表 17.	MADC 转换时间（CIC 滤波器忽略采样点数）	44
表 18.	芯片可靠性分析结果.....	45
表 19.	V85XX 管脚描述.....	49
表 20.	PMU 寄存器地址（PMU 基地址：0x40014000）	60
表 21.	PMU_RAM 寄存器地址（PMU_RAM 基地址：0x40014400）	60
表 22.	模拟控制寄存器地址（ANA 基地址：0x40014200）	60
表 23.	RTC 控制寄存器地址（RTC 基地址：0x40014800）	62
表 24.	FLASH 控制寄存器地址（FLASH 控制寄存器基地址：0x00000000）	63
表 25.	GPIO 控制寄存器地址（GPIO 基地址：0x40000000）	64
表 26.	DMA 控制寄存器地址（DMA 基地址：0x40010000）	65

表 27.	UART 寄存器地址 (UART 基地址: 0x40011800)	66
表 28.	UART32K 寄存器地址 (UART32K 基地址: 0x40014100)	67
表 29.	ISO7816 寄存器地址 (ISO7816 基地址: 0x40012000)	67
表 30.	TIMER 寄存器地址 (32 位 TIMER 基地址: 0x40012800)	67
表 31.	PWM TIMER 寄存器地址 (16 位 PWM TIMER 基地址: 0x40012900)	68
表 32.	LCD 寄存器地址 (LCD 基地址: 0x40002000)	69
表 33.	SPI1 寄存器地址 (SPI1 基地址: 0x40011000)	70
表 34.	SPI2 寄存器地址 (SPI2 基地址: 0x40015800)	70
表 35.	SPI3 寄存器地址 (SPI3 基地址: 0x40016000)	70
表 36.	SPI4 寄存器地址 (SPI4 基地址: 0x40017000)	71
表 37.	I2C 寄存器地址 (I2C 基地址: 0x40010800)	71
表 38.	MISC 寄存器地址 (MISC 基地址: 0x40013000)	71
表 39.	MISC2 寄存器地址 (MISC2 基地址: 0x40013E00)	71
表 40.	CRYPT 寄存器地址 (CRYPT 基地址: 0x40006000)	72
表 41.	EPU 寄存器地址 (EPU 基地址: 0x40015000)	72
表 42.	RNG 寄存器地址 (RNG 基地址: 0x40016800)	73
表 43.	Info 信息寄存器	74
表 44.	ANA 寄存器地址 (ANA 基地址: 0x40014200)	84
表 45.	ANA_REG5 寄存器	84
表 46.	ANA_REG6 寄存器	84
表 47.	ANA_REG7 寄存器	85
表 48.	ANA_REG8 寄存器	85
表 49.	ANA_REG9 寄存器	85
表 50.	ANA_REGA 寄存器	85
表 51.	ANA_REGF 寄存器	86
表 52.	ANA_CMPOUT 寄存器	86
表 53.	ANA_INTSTS 寄存器	86
表 54.	ANA_INTEN 寄存器	87

表 55.	PMU 工作模式寄存器地址（PMU 基地址：0x40014000）	93
表 56.	PMU_RAM 寄存器地址（PMU_RAM 基地址：0x40014400）	93
表 57.	ANA 控制器寄存器地址（基地址：0x40014200）	93
表 58.	PMU_CONTROL 寄存器	93
表 59.	PMU_STS 寄存器	94
表 60.	PMU_WDTSTS 寄存器	95
表 61.	各个模块在不同复位及睡眠唤醒后的复位情况	95
表 62.	在不同模式下的各个时钟的工作状态	96
表 63.	中断源.....	97
表 64.	每一种工作模式下的唤醒源	102
表 65.	不同工作模式下的时钟模块开关	106
表 66.	ANA 控制寄存器（ANA 基地址：0x40014200）	108
表 67.	PMU 寄存器地址（PMU 基地址：0x40014000）	109
表 68.	MISC2 控制寄存器（MISC2 基地址：0x40013E00）	109
表 69.	ANA_REG3 控制寄存器.....	109
表 70.	ANA_REG4 控制寄存器.....	109
表 71.	ANA_REG9 控制寄存器.....	110
表 72.	ANA_REGB 控制寄存器	110
表 73.	ANA_REGC 控制寄存器	111
表 74.	ANA_REG10 控制寄存器.....	111
表 75.	ANA_REG11 控制寄存器.....	111
表 76.	ANA_CMPOUT 寄存器	112
表 77.	PMU_CONTROL 寄存器.....	112
表 78.	PMU_STS 寄存器	112
表 79.	MISC2_CLKSEL 寄存器	113
表 80.	MISC2_CLKDIVH 寄存器.....	113
表 81.	MISC2_CLKDIVP 寄存器.....	114
表 82.	MISC2_HCLKEN 寄存器.....	114

表 83.	HCLK 时钟使能.....	114
表 84.	MISC2_PCLKEN 寄存器	114
表 85.	各个模块的 PCLK 时钟使能.....	115
表 86.	MISC2_EPUCLKCTRL 寄存器.....	115
表 87.	ANA 控制寄存器（ANA 基地址：0x40014200）	117
表 88.	ANA_REGx	119
表 89.	ANA_REG0 各个位的功能描述	120
表 90.	ANA_REG1 各个位的功能描述	120
表 91.	ANA_REG2 各个位的功能描述	120
表 92.	ANA_REG3 各个位的功能描述	121
表 93.	ANA_REG4 各个位的功能描述	121
表 94.	ANA_REG5 各个位的功能描述	122
表 95.	ANA_REG6 各个位的功能描述	122
表 96.	ANA_REG7 各个位的功能描述	122
表 97.	ANA_REG8 各个位的功能描述	123
表 98.	ANA_REG9 各个位的功能描述	123
表 99.	ANA_REGA 各个位的功能描述.....	124
表 100.	ANA_REGB 各个位的功能描述.....	124
表 101.	ANA_REGC 各个位的功能描述.....	125
表 102.	ANA_REGD 各个位的功能描述.....	125
表 103.	ANA_REGE 各个位的功能描述.....	125
表 104.	ANA_REGF 各个位的功能描述	126
表 105.	ANA_REG10 各个位的功能描述.....	126
表 106.	ANA_REG11 各个位的功能描述.....	126
表 107.	ANA_CTRL 寄存器	127
表 108.	ANA_CMPOUT 寄存器	128
表 109.	ANA_INTSTS 寄存器.....	129
表 110.	ANA_INTEN 寄存器.....	130

表 111.	ANA_ADCCTRL 寄存器.....	131
表 112.	ANA_ADCDATAx 寄存器	133
表 113.	ANA_CMPCNTx 寄存器.....	133
表 114.	ANA 控制寄存器（ANA 基地址：0x40014200）	135
表 115.	ANA_REG1 寄存器.....	135
表 116.	ANA_REG3 各个位的功能描述	135
表 117.	ANA_INTSTS 寄存器.....	136
表 118.	ANA_INTEN 寄存器.....	136
表 119.	ANA_ADCCTRL 寄存器.....	137
表 120.	ADC 采样通道.....	138
表 121.	ANA_ADCDATAx 寄存器	139
表 122.	电压测量量程与公式.....	140
表 123.	ANA 控制寄存器（ANA 基地址：0x40014200）	144
表 124.	ANA_REG2 寄存器.....	144
表 125.	ANA_REG3 寄存器.....	145
表 126.	ANA_REG5 寄存器.....	145
表 127.	ANA_REGF 寄存器.....	145
表 128.	ANA_CTRL 寄存器	146
表 129.	ANA_CMPOUT 寄存器	147
表 130.	ANA_INTSTS 寄存器.....	147
表 131.	ANA_INTEN 寄存器.....	147
表 132.	ANA_CMPCNTx 寄存器.....	147
表 133.	看门狗在不同模式下状态（MODE=1）	149
表 134.	PMU_WDT 控制器（PMU 基地址：0x40014000）	149
表 135.	PMU_WDTPASS 寄存器.....	150
表 136.	PMU_WDTEN 寄存器	150
表 137.	PMU_WDTCLR 寄存器	150
表 138.	PMU_WDTSTS 寄存器	151

表 139.	RTC 控制器 (RTC 基地址: 0x40014800)	153
表 140.	RTC_SEC/MIN/DAY/WEEK/MONTH/YEAR 寄存器	155
表 141.	RTC_WKUSEC 寄存器	155
表 142.	RTC_WKUMIN 寄存器	156
表 143.	RTC_WKU 时寄存器	156
表 144.	RTC_WKUCNT 寄存器	156
表 145.	RTC_CAL 寄存器	157
表 146.	RTC_DIV 寄存器	157
表 147.	RTC_CTL 寄存器	158
表 148.	RTC_PWD 寄存器	158
表 149.	RTC_CE 寄存器	158
表 150.	RTC_LOAD 寄存器	159
表 151.	RTC_INTSTS 寄存器	159
表 152.	RTC_INTEN 寄存器	160
表 153.	RTC_PSCA 寄存器	160
表 154.	RTC_ACTI 寄存器	161
表 155.	RTC_ACF200 寄存器	161
表 156.	RTC_ACADCW 寄存器	161
表 157.	RTC_ACP0 寄存器	161
表 158.	RTC_ACP1 寄存器	162
表 159.	RTC_ACP2 寄存器	162
表 160.	RTC_ACP3 寄存器	162
表 161.	RTC_ACP4 寄存器	162
表 162.	RTC_ACP5 寄存器	163
表 163.	RTC_ACP6 寄存器	163
表 164.	RTC_ACP7 寄存器	163
表 165.	RTC_ACKx 寄存器	163
表 166.	RTC_ACKTEMP 寄存器	164

表 167.	RTC_ACTEMP 寄存器	164
表 168.	RTC_ACPPM 寄存器	165
表 169.	RTC_WKUCNTR 寄存器	165
表 170.	Info 信息寄存器（与 RTC 温度补偿有关）	165
表 171.	FLASH 控制器地址（FLASH 控制器基地址：0x00000000）	171
表 172.	MISC2 控制器 for FLASH（MISC2 基地址：0x40013E00）	172
表 173.	FLASH_PASS 寄存器	172
表 174.	FLASH_CTRL 寄存器	172
表 175.	FLASH_PGADDR 寄存器	172
表 176.	PGADDR 及写数据寄存器	173
表 177.	FLASH_PGDATA 寄存器	173
表 178.	FLASH_PGBx 寄存器	173
表 179.	FLASH_PGHwX 寄存器	174
表 180.	FLASH_SERASE 寄存器	174
表 181.	FLASH_CERASE 寄存器	174
表 182.	FLASH_DSTB 寄存器	174
表 183.	FLASH_INT 寄存器	175
表 184.	FLASH_CSSADDR 寄存器	175
表 185.	FLASH_CSEADDR 寄存器	175
表 186.	FLASH_CSVALUE 寄存器	176
表 187.	FLASH_CSCVALUE 寄存器	176
表 188.	FLASH_STS 寄存器	176
表 189.	MISC2_FLASHWC 寄存器	176
表 190.	PMU_IOA 控制器（PMU 基地址：0x40014000）	179
表 191.	GPIO 控制器（GPIO 基地址：0x40000000）	179
表 192.	PMU_IOAOEN 寄存器	180
表 193.	PMU_IOAIE 寄存器	180
表 194.	PMU_IOADAT 寄存器	180

表 195.	PMU_IOAATT 寄存器	180
表 196.	不同设置下的 IO 状态	181
表 197.	PMU_IOAWKUEN 寄存器	181
表 198.	IO 唤醒模式	181
表 199.	PMU_IOASTS 寄存器	182
表 200.	PMU_IOAINTSTS 寄存器	182
表 201.	PMU_IOASEL 寄存器	182
表 202.	PMU_IOANODEGL 寄存器	183
表 203.	IOX_OEN 寄存器	183
表 204.	IOX_IE 寄存器	183
表 205.	IOX_DAT 寄存器	183
表 206.	IOX_ATT 寄存器	184
表 207.	IOX_STS 寄存器	184
表 208.	IOB_SEL 寄存器	184
表 209.	IOE_SEL 寄存器	185
表 210.	IO_MISC 寄存器	185
表 211.	特殊功能 IOA	186
表 212.	特殊功能 IOB	186
表 213.	特殊功能 IOC	187
表 214.	特殊功能 IOD	188
表 215.	特殊功能 IOE	188
表 216.	特殊功能 IOF	189
表 217.	DMA 控制器 (DMA 基地址: 0x40010000)	191
表 218.	DMA_IE 寄存器	192
表 219.	DMA_STS 寄存器	193
表 220.	DMA_CxCTL 寄存器	194
表 221.	DMA 请求源选择	195
表 222.	DMA_CxSRC 寄存器	196

表 223.	DMA_CxDST 寄存器.....	196
表 224.	DMA_CxLEN 寄存器.....	197
表 225.	DMA_AESCTL 寄存器	197
表 226.	DMA_AESKEYx 寄存器.....	197
表 227.	UART 控制器（UART 基地址：0x40011800）	200
表 228.	MISC 控制器（MISC 基地址：0x40013000）	201
表 229.	UARTx_数据寄存器	201
表 230.	UARTx_STATE 寄存器.....	201
表 231.	UARTx_CTRL 寄存器	202
表 232.	UARTx_INTSTS 寄存器	202
表 233.	UARTx_BAUDDIV 寄存器	203
表 234.	UARTx_CTRL2 寄存器	203
表 235.	MISC_IREN 寄存器.....	203
表 236.	MISC_DUTYL 寄存器.....	204
表 237.	MISC_DUTYH 寄存器	204
表 238.	UART32K 控制器（UART32K0 基地址：0x40014100）	206
表 239.	UART32K 控制器（UART32K1 基地址：0x40014180）	206
表 240.	U32Kx_CTRL0 寄存器.....	206
表 241.	U32Kx_CTRL1 寄存器.....	207
表 242.	U32Kx_PHASE 寄存器	207
表 243.	U32Kx_数据寄存器	208
表 244.	U32Kx_STS 寄存器	208
表 245.	ISO7816 控制器地址（ISO7816 基地址：0x40012000）	209
表 246.	ISO7816x_BAUDDIVL 寄存器	210
表 247.	ISO7816x_BAUDDIVH 寄存器.....	210
表 248.	ISO7816x_数据寄存器	210
表 249.	ISO7816x_INFO 寄存器.....	211
表 250.	ISO7816x_CFG 寄存器.....	211

表 251.	ISO7816x_CLK 寄存器	212
表 252.	ISO7816 数据包格式	212
表 253.	32 位 TIMER 控制器（32 位 TIMER 基地址：0x40012800）	215
表 254.	16 位 PWM TIMER 控制器（16bPWM TIMER 基地址：0x40012900）	216
表 255.	TMRx_CTRL 寄存器	217
表 256.	TMRx_VALUE 寄存器	217
表 257.	TMRx_RELOAD 寄存器	217
表 258.	TMRx_INT 寄存器	217
表 259.	PWMx_CTL 寄存器	218
表 260.	PWMx_TAR 寄存器	220
表 261.	PWMx_CCTLx 寄存器	220
表 262.	PWMx_CCRx 寄存器	223
表 263.	PWM_O_SEL 寄存器	223
表 264.	与 LCD 相关的 ANA 控制器（ANA 基地址：0x40014200）	231
表 265.	LCD 控制器（LCD 基地址：0x40002000）	231
表 266.	LCD_FBx 寄存器	232
表 267.	LCD_FBx 寄存器的数据排布	232
表 268.	LCD_CTRL 寄存器	233
表 269.	LCD_CTRL2 寄存器	233
表 270.	LCD_SEGCTRL0 寄存器	234
表 271.	LCD_SEGCTRL1 寄存器	235
表 272.	LCD_SEGCTRL2 寄存器	235
表 273.	ANA_REG6 寄存器	235
表 274.	SPI1 控制器（SPI1 基地址：0x40011000）	237
表 275.	SPI2 控制器（SPI2 基地址：0x40015800）	238
表 276.	SPI3 控制器（SPI3 基地址：0x40016000）	238
表 277.	SPI4 控制器（SPI4 基地址：0x40017000）	238
表 278.	SPIx_CTRL 寄存器	238

表 279.	SPIx_TXSTS 寄存器	239
表 280.	SPIx_TXDAT 寄存器.....	241
表 281.	SPIx_RXSTS 寄存器.....	241
表 282.	SPIx_RXDAT 寄存器	242
表 283.	SPIx_MISC 寄存器.....	243
表 284.	I2C 控制器（I2C 基地址：0x40010800）	248
表 285.	I2C_数据寄存器	248
表 286.	I2C_ADDR 寄存器	249
表 287.	I2C_CTRL 寄存器	249
表 288.	SCL 频率设置	250
表 289.	I2C_STS 寄存器.....	250
表 290.	主机发送模式下的 I2C 状态	250
表 291.	主机接收模式下的 I2C 状态	252
表 292.	从机接收模式下的 I2C 状态	252
表 293.	从机发送模式下的 I2C 状态	254
表 294.	I2C 状态.....	255
表 295.	I2C_CTRL2 寄存器	256
表 296.	I2C AC 特性.....	260
表 297.	中断源.....	266
表 298.	MISC 控制器（MISC 基地址：0x40013000）	270
表 299.	MISC2 控制器（MISC2 基地址：0x40013E00）	271
表 300.	MISC_SRAMINT 寄存器	271
表 301.	MISC_SRAMINIT 寄存器	272
表 302.	MISC_PARERR 寄存器.....	272
表 303.	MISC_IREN 寄存器.....	272
表 304.	MISC_DUTYL 寄存器.....	272
表 305.	MISC_DUTYH 寄存器	273
表 306.	MISC_IRQLAT 寄存器	273

表 307.	MISC_HIADDR 寄存器.....	273
表 308.	MISC_PIADDR 寄存器.....	274
表 309.	MISC2_FLASHWC 寄存器.....	274
表 310.	MISC2_CLKSEL 寄存器	274
表 311.	MISC2_CLKDIVH 寄存器.....	275
表 312.	MISC2_CLKDIVP 寄存器.....	275
表 313.	MISC2_HCLKEN 寄存器.....	275
表 314.	HCLK 时钟使能.....	275
表 315.	MISC2_PCLKEN 寄存器.....	276
表 316.	PCLK 时钟使能.....	276
表 317.	MISC2_EPUCLKCTRL 寄存器.....	277
表 318.	MISC2_RSTGPU 寄存器	277
表 319.	CRYPT 控制器（MISC 基地址：0x40006000）	279
表 320.	CRYPT_CTRL 寄存器.....	279
表 321.	CRYPT_PTRA 寄存器.....	280
表 322.	CRYPT_PTRB 寄存器.....	280
表 323.	CRYPT_PTRO 寄存器	281
表 324.	CRYPT_CARRY 寄存器.....	281
表 325.	CRYPT_PRIMEn 寄存器.....	281
表 326.	RNG 控制器（RNG 基地址：0x40016800）	284
表 327.	RNG_IMR 寄存器.....	284
表 328.	RNG_ISR 寄存器	285
表 329.	RNG_ICR 寄存器	285
表 330.	RNG_CFG 寄存器.....	286
表 331.	RNG_DATA0 寄存器.....	286
表 332.	RNG_DATA1 寄存器.....	286
表 333.	RNG_DATA2 寄存器.....	286
表 334.	RNG_DATA3 寄存器.....	287

表 335.	RNG_DATA4 寄存器.....	287
表 336.	RNG_DATA5 寄存器.....	287
表 337.	RNG_CMD 寄存器	287
表 338.	RNG_CNT 寄存器	287
表 339.	RNG_BYBASS 寄存器.....	287
表 340.	EPU 控制器（EPU 基地址：0x40015000）	290
表 341.	EPU_CFCTRL 寄存器.....	291
表 342.	EPU_HSCTRL 寄存器.....	292
表 343.	EPU_LSCTRL 寄存器	293
表 344.	EPU_CLR 寄存器.....	294
表 345.	EGY_CRPTH 寄存器.....	295
表 346.	EGY_PWRTH0 寄存器	295
表 347.	EGY_PWRTH1 寄存器	295
表 348.	EPU_INTEN 寄存器	295
表 349.	EPU_INTSTS 寄存器	296
表 350.	EPU_STS 寄存器	296
表 351.	EPU_CONST0 寄存器.....	296
表 352.	EPU_OUTL0 寄存器	296
表 353.	EPU_OUTH0 寄存器	297
表 354.	EPU_CFCNT0 寄存器	297
表 355.	EPU_CONST1 寄存器.....	297
表 356.	EPU_OUTL1 寄存器	297
表 357.	EPU_OUTH1 寄存器	297
表 358.	EPU_CFCNT1 寄存器	297
表 359.	EPU_CONSTx （x=2-7）寄存器.....	298
表 360.	EPU_OUTx （x=2-7）寄存器.....	298
表 361.	EPU_CFCNTx （x=2-7）寄存器	298
表 362.	SWD 引脚的特殊功能.....	299

表 363.	CMSIS 函数	300

图目录

图 1.	V85XX 系统功能框图.....	56
图 2.	V85XX 电源系统功能框图.....	57
图 3.	V85XX 时钟功能框图.....	58
图 4.	V85XX 内存映射.....	59
图 5.	V85XX 电源系统框图.....	83
图 6.	AVCCLDO 输出电压与 VDD 输入电压之间的关系	88
图 7.	AVCCLDO 输出电压与负载电流之间的关系	88
图 8.	DVCCLDO 输出电压与 VDD 输入电压之间的关系	89
图 9.	DVCCLDO 输出电压与负载电流之间的关系	89
图 10.	VDCIN 管脚输入信号与位 VDCINDROP 的关系.....	90
图 11.	PMU 功能框图.....	92
图 12.	片外复位去毛刺时序	97
图 13.	V85XX 工作模式转换.....	101
图 14.	外部唤醒示意图.....	103
图 15.	进入睡眠的流程.....	104
图 16.	进入 IDLE 的流程.....	105
图 17.	V85XX 时钟系统功能框图.....	107
图 18.	模拟控制器的功能框图	117
图 19.	ADC 功能框图.....	134
图 20.	比较器的迟滞窗口	143
图 21.	比较器功能框图.....	143
图 22.	RTC 控制器功能框图.....	152
图 23.	RTC 手动温度补偿流程图.....	168
图 24.	FLASH 控制器的功能框图.....	170
图 25.	GPIO 控制器的功能框图.....	178

图 26.	DMA 控制器功能框图	190
图 27.	UART 控制器的功能框图.....	199
图 28.	UART32K 控制器的功能框图	205
图 29.	ISO7816 控制器功能框图	209
图 30.	定时器功能框图.....	214
图 31.	PWM 定时器功能框图	214
图 32.	PWM 定时器模式	218
图 33.	连续模式示例	219
图 34.	向上计数模式下的 PWM 输出	221
图 35.	连续计数模式下的 PWM 输出	222
图 36.	向上/向下计数模式下的 PWM 输出	223
图 37.	LCD 控制器的功能框图	225
图 38.	LCD 控制器结构图	226
图 39.	LCD 波形[1/4 Duty, 1/3 Bias].....	226
图 40.	LCD 波形[1/6 Duty, 1/3 Bias].....	228
图 41.	LCD 波形[1/8 Duty, 1/3 Bias].....	229
图 42.	LCD 波形[1/8 Duty , 1/4 Bias].....	230
图 43.	不同 FBMODE 下的帧缓存器操作模式.....	234
图 44.	SPI 控制器的功能框图.....	236
图 45.	主机模式, SPO=0, SPH=0.....	244
图 46.	主机模式, SPO=0, SPH=1.....	244
图 47.	主机模式, SPO=1, SPH=0.....	245
图 48.	主机模式, SPO=1, SPH=1.....	246
图 49.	I2C 控制器功能框图	247
图 50.	启动和停止条件.....	256
图 51.	7 位地址写模式	257
图 52.	7 位地址读模式	257
图 53.	数据传输	257

图 54.	I2C 总线的应答机制	258
图 55.	I2C 总线连接例	259
图 56.	SCL 和 SDA 时序	259
图 57.	I2C 控制器的 AC 时序	260
图 58.	主机发送模式的操作流程	262
图 59.	主机接收模式的操作流程	263
图 60.	从机发送模式的操作流程	264
图 61.	从机接收模式的操作流程	265
图 62.	MISC 控制器的功能框图	270
图 63.	CRYPT 控制器的功能框图	278
图 64.	EPU 控制器的功能框图	289
图 65.	高速能量累加器的功能框图	289
图 66.	低速能量累加器的功能框图	290

版本历史

表1. 文档版本历史

日期	版本	描述
2023.05.25	V1.0	初版
2023.07.12	V1.1	修正 GPIO 控制器的功能框图。 修正 MISC2_EPUCLOCKCTRL 寄存器名字。 修改 CRYPT_CTRL 寄存器中 MODE 的描述。 修改 CAN 章节描述。
2024.01.25	V2.0	修改 EXTRSTN 管脚的滤波电阻为 51K 欧。 修改 CRYPT_CTRL 寄存器中 ACT 的描述。 删除 CAN。 修正 CRYPT_PRIME 寄存器的默认值。 修正 PMU_IOAINT 为 PMU_IOAINTSTS。 修正 FLASH_CSEADDR 寄存器的默认值。 修改 RTC_ACTI 寄存器的位宽。 修正 MISC_HIADDR 寄存器的描述。 修正 MISC_PIADDR 寄存器的描述。 增加 SPI3 和 SPI4。 修改 MISC2_PCLKEN 寄存器的默认值。 增加 ANA_REG10 和 ANA_REG11 寄存器。 修改 ANA_REG4 寄存器的描述。 修改功耗参数。
2024.03.22	V3.0	增加 V8520 型号。
2024.05.31	V3.1	删除 V8520 型号。
2024.07.23	V3.2	更新 GPIO 章节排版。
2024.11.07	V3.3	删除深睡眠功能； 删除 RTC 自动温补功能； 更正电气特性参数；

V85XX 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，256KB FLASH，32KB SRAM，UART/SPI/I2C 接口，EPU，RNG，LCD，WDT 和 RTC。V85XX 系列支持多种低功耗工作模式。

特点

- 工作电压：
 - V85XX: 2.2V~5.5V,
- 工作电流：
 - 正常模式: 1.64mA@6.5536MHz
 - IDLE 模式: 0.28mA@6.5536MHz
 - 睡眠模式 (LCD 开启): 8.76μA
 - 睡眠模式 (LCD 关闭): 2.89μA
- 封装：
 - LQFP100 (V8500)
 - LQFP80 (V8531)
 - LQFP64 (V8510)
- 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 256KB FLASH 存储器，具有写保护和加密功能，支持 ISP 和 IAP
 - 32K 字节的 SRAM，带奇偶校验，在睡眠模式下可数据保留
 - 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 4 个 SPI 控制器 (主/从机制)
 - 最多 1 个 I2C 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，2.7~3.6V可调节，调节步长是0.06V

- 周期可编程的看门狗定时器
- 支持多唤醒源
- 最多 82 个 GPIO
- 最多 16 个 GPIO 外部中断唤醒口
- 最多 1 个 RNG 控制器
- 1 个能量累加控制器（EPU）带 2 个高速能量累加器、6 个低速能量累加器、支持 2 路 CF 输出
- 模拟控制器
 - 16 位 ADC，具有 10Ksps 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 6.5MHzRC 时钟
 - 内置 2 个 PLL
 - 支持外部 32.768 KHz 晶体或 6.5536 MHz 晶体（可选）
 - 内置 32.768 KHz 和 6.5536 MHz 晶体的监测电路
 - 每个时钟都可选作为系统时钟
 - 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 VDD、VDCIN、VDD、BATRTC、AVCC、DVCC 的电压监测

产品系列

外设	V8500	V8531	V8510
FLASH	256KB	256KB	256KB
SRAM	32KB	32KB	32KB
ADC 输入通道	8	8	4
UART	6	6	5
UART32K	2	2	2
ISO7816	2	2	2
SPI	4	4	4
I2C	1	1	1
比较器	2	2	1
GPIO	82	71	53
外部中断 IO	16	16	15
32 位定时器	4	4	4
16 位 PWM 定时器	4	4	4
PWM_Out	4	4	4
DMA channel	4	4	4
RNG	1	1	1
EPU	1	1	1
LCD	76x4, 74x6, 72x8	65x4, 63x6, 61x8	49x4, 47x6, 45x8
封装	100-LQFP	80-LQFP	64-LQFP

1 电气特性

除非另有说明，所有电压均以 V_{SS} 为参考电位。

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2. 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}	VDD 上电速度	5V/s	1V/μs	--
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	°C
T _S	储存温度范围	-55	+150	°C
T _J	PN 结结温	-40	+125	°C

1.2 正常工作电压

表3. 供电电压

符号	参数	条件	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

1.3 驱动特性

表4. 电源管脚的驱动条件

符号	参数	条件	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

注1: VDD 的驱动电流包括外设、AVCC、DVCC 和 AVCC_OUT 上的负载。

注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5. 电源开关导通电阻

符号	参数	条件	最小	典型	最大	单位
R_{on} , VDD2AVCC	关闭 AVCC LDO 时, VDD、AVCC 之间的 开关导通电阻	VDD=3.3V, $I_L^{(1)} < 30mA$	6.5	7.3	9.0	ohm

注1: AVCC 的负载电流。.

1.5 功耗

表6. 常用工作条件

符号	参数	条件 (CRPTY 时钟关闭, AVCC LDO 关闭, VDD=3.3V (特殊说明除外))	最小	典型	最大	单位
I_{ACTIVE}	工作电流	系统时钟: 26.2144MHz		4750		μA
		系统时钟: 13.1072MHz		2640		
		系统时钟: 6.5536MHz		1510		
		系统时钟: 3.2768MHz		960		
		系统时钟: 1.6384MHz		680		
		系统时钟: 819.2kHz		530		
		系统时钟: 409.6kHz		440		
		系统时钟: 204.8kHz		400		
		系统时钟: 32K RC (FLASH deep-standby)		14.9		
		系统时钟: 32K RC (FLASH 正常)		85.0		
I_{IDLE}	IDLE 电流	系统时钟: 26.2144MHz		779		μA
		系统时钟: 13.1072MHz		470		
		系统时钟: 6.5536MHz		286		
		系统时钟: 3.2768MHz		223		
		系统时钟: 1.6384MHz		182		
		系统时钟: 819.2kHz		162		
		系统时钟: 409.6kHz		152		
		系统时钟: 204.8kHz		127		
		系统时钟: 32K RC		3.65		
I_{SLP1}	LCD 关闭 时的睡眠模 式电流	RTCCLK 不分频		2.89		μA
		RTCCLK 4 分频		2.7		
I_{SLP2}	LCD 开启 时的睡眠模 式电流	RTCCLK 不分频		8.76		μA
		RTCCLK 4 分频		8.5		

1.6 复位和电源控制模块特性

表7. 电源和复位控制模块特性

符号	参数	条件	最小	典型	最大	单位
t_{RST}	复位消抖时间			20		μs
V_{PORH}	PORH 监测电压 (AVCCCLDO)		1.9	2.1	2.2	V
V_{VDCIN}	V_{TH+}	工作温度	1.5			V
	V_{TH-}	工作温度			1.19	
V_{AVCCLV}	AVCC 低电压监测电平		2.2	2.5	2.7	V
$V_{VDDALRAM}$	VDD 电源监测电压	V_{TH} 可配置, $V_{TH}=2.9V$; 参考电压为 V_{REF}	2.73	2.9	3.02	V
AVCC	AVCC 电压		3.1	3.3	3.4	V
LCDLDO	LCDLDO 电压		3.2	3.3	3.4	V

表8. 电压回差特性

符号	参数	条件	最小	典型	最大	单位
V_{PORH_HTRES}	PORH 监测电压回差			66.5		mV
V_{VDCIN_HTRES}	VDCIN 监测电压回差			42.0		mV
V_{AVCCLV_HTRES}	AVCCLV 监测电压回差			42.0		mV
$V_{VDDALRAM\_HTRES}$	VDD 电源监测电压回差	$V_{TH\_VDDALARM}=4.5\text{ V}$		142.5		mV
		$V_{TH\_VDDALARM}=4.2\text{ V}$		133.0		mV
		$V_{TH\_VDDALARM}=3.9\text{ V}$		123.5		mV
		$V_{TH\_VDDALARM}=3.6\text{ V}$		114.0		mV
		$V_{TH\_VDDALARM}=3.2\text{ V}$		101.4		mV
		$V_{TH\_VDDALARM}=2.9\text{ V}$		91.9		mV
		$V_{TH\_VDDALARM}=2.6\text{ V}$		82.4		mV
		$V_{TH\_VDDALARM}=2.3\text{ V}$		72.9		mV

1.7 GPIO 特性

表9. GPIO 特性

符号	参数	V_{VDD}	条件	最小	最大	单位
V_{IH}	输入高电平	5V		$0.7 \cdot V_{VDD}$		V

		3.3V		2		
V _{IL}	输入低电平	5V			0.3*V _{VD} _D	V
		3.3V			0.3*V _{VD} _D	V
V _{HSYS}	施密特触发迟滞	5V		0.1*V _{VDD}		V
		3.3V				
I _{IH}	输入高电平时电流	5V			+2	μA
		3.3V				
I _{IL}	输入低电平时电流	5V		-2		μA
		3.3V				
V _{OH}	输出高电平	5V	驱动电流 10mA	V _{VDD} -0.8	V _{VDD}	V
		3.3V	驱动电流 5mA	2.4	V _{VDD}	V
V _{OL}	输出低电平	5V	驱动电流 10mA		0.5	V
		3.3V	驱动电流 5mA		0.4	V
C _{IN}	输入电容	5V			10	pF
		3.3V				

1.8 ADC 特性

表10. ADC 特性

符号	参数	条件	最小	典型	最大	单位
BGPREF	BGP 电压		1.18	1.2	1.22	V
PSRR	BGP 的电源抑制比			-55		dB
V _{ADC}	ADC 工作电压		2.7	3.3	3.6	V
I _{ADC}	ADC 工作电流		170	230	350	μA
f _{ADCLK}	ADC 采样时钟			1.6384		MHz
C _{ADC}	内部采样和保持电容			1		pF
INL	积分非线性			2		LSB
DNL	微分非线性			1		LSB
Offset	偏移误差			5		mV
V _{WITHSTAND}	管脚耐压值（ADC 输入电压范围）		-0.3		VDD	V

1.9 模拟比较器特性

表11. 比较器特性

符号	参数	条件	最小	典型	最大	单位
V _{CMP}	比较器工作		2.2	3.3	3.6	V



	电压 (AVCC)					
I _{CMP}	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
t _d	传播迟滞	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 _{CMPIN}	比较器输入 电压范围		0.8		VDD-0.3	V
V _{CMPREF}	比较器参考 电压	参考电压为 VREF (工作温 度)	1.1	1.3	1.5	V
		参考电压为 BGPREF	1.18	1.2		V
V _{HTRES}	比较器回差		20.0	25.0	30.0	mV

1.10 时钟和锁相环特性

表12. 时钟和锁相环特性

符号	参数	条件	最小	典型	最大	单位
V _{DDPLL}	PLL 工作电压 (DVCC)		1.35	1.5	1.65	V
I _{VDDPLL}	PLL 工作电流			30		μA
V _{DDPLLH}	PLLH 工作电压 (DVCC)		1.35	1.5	1.65	V
I _{VDDPLLH}	PLLH 工作电流			40		μA
V _{DDRCL}	RCL 工作电压 (AVCC)		2.2	3.3	3.6	V
I _{VDDRCL}	RCL 工作电流			0.2		μA
f _{RCL}	RCL 频率		26	32	35.5	kHz
V _{DDRCH}	RCH 工作电压 (AVCC)		2.2	3.3	3.6	V
I _{VDDRCH}	RCH 工作电流			45		μA
f _{RCH}	RCH 频率		6.3	6.5	6.8	MHz
V _{DDXOH}	XOH 工作电压 (AVCC)		2.2	3.3	3.6	V

I _{VDDXOH}	XOH 工作电流			150		μA
f _{XOH}	XOH 频率			6.5536		MHz

1.11 FLASH 和 SRAM 特性

表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 512 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14. 晶振的 ESR 特性

参数	条件	最小	典型	最大	单位
6.5536M 晶振的 ESR ⁽¹⁾				100	Ω
32768K 晶振的 ESR				50	KΩ

注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15. 时钟输出稳定时间及唤醒时间

参数	条件	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				10	μs
RCH 为系统时钟时的睡眠唤醒时间			18.4		μs
PLLL 为系统时钟时的睡眠唤醒时间			1.03		ms
PLLH 为系统时钟时的睡眠唤醒时间			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 ADC 转换时间

ADC 时钟频率、CIC 滤波器降采样率、CIC 滤波器忽略采样点数的配置不同，MADC 转换时间不同。详情请参考下表。

表16. ADC 转换时间（CIC 滤波器降采样率与 ADC 时钟频率）

CIC 滤波器 降采样率	ADC 时钟频率/MHz				
	6.5536	3.2768	1.6384	0.8192	0.4096
1/512	0.937	0.937	1.875	3.750	7.500
1/256	0.468	0.468	0.937	1.875	3.750
1/128	0.234	0.234	0.468	0.937	1.875
1/64	0.117	0.117	0.234	0.468	0.937
其他参数配置：CICSKIP=6，CIC 滤波器忽略前 2 个采样点。 注 1：ADC 为 3 阶 CIC，忽略前两个采样点，数据稳定。 注 2：ADC 转换时间 = $\frac{1}{\left(\text{CIC 降采样率} \times \frac{\text{ADC 时钟频率}}{2}\right)} \times \text{采样点}$（ADC 时钟频率为 6.5536MHz 时除外），如：CIC 降采样率为 1/512，ADC 时钟频率为 3.2768MHz，忽略前 2 个采样点， ADC 转换时间 = $\frac{1}{\left(\frac{1}{512} \times \frac{3.2768\text{MHz}}{2}\right)} \times 3 = 0.937\text{ms}$					

表17. MADC 转换时间（CIC 滤波器忽略采样点数）



参数	条件（忽略采样点数）	典型	单位
MADC 转换时间	2	1.875	ms
	3	2.520	ms
	4	3.150	ms
	5	3.780	ms
	6	4.410	ms
	7	5.040	ms
其他参数配置：降采样率为 1/512，ADC 时钟频率为 1.6384MHz。			

1.15 芯片可靠性分析结果

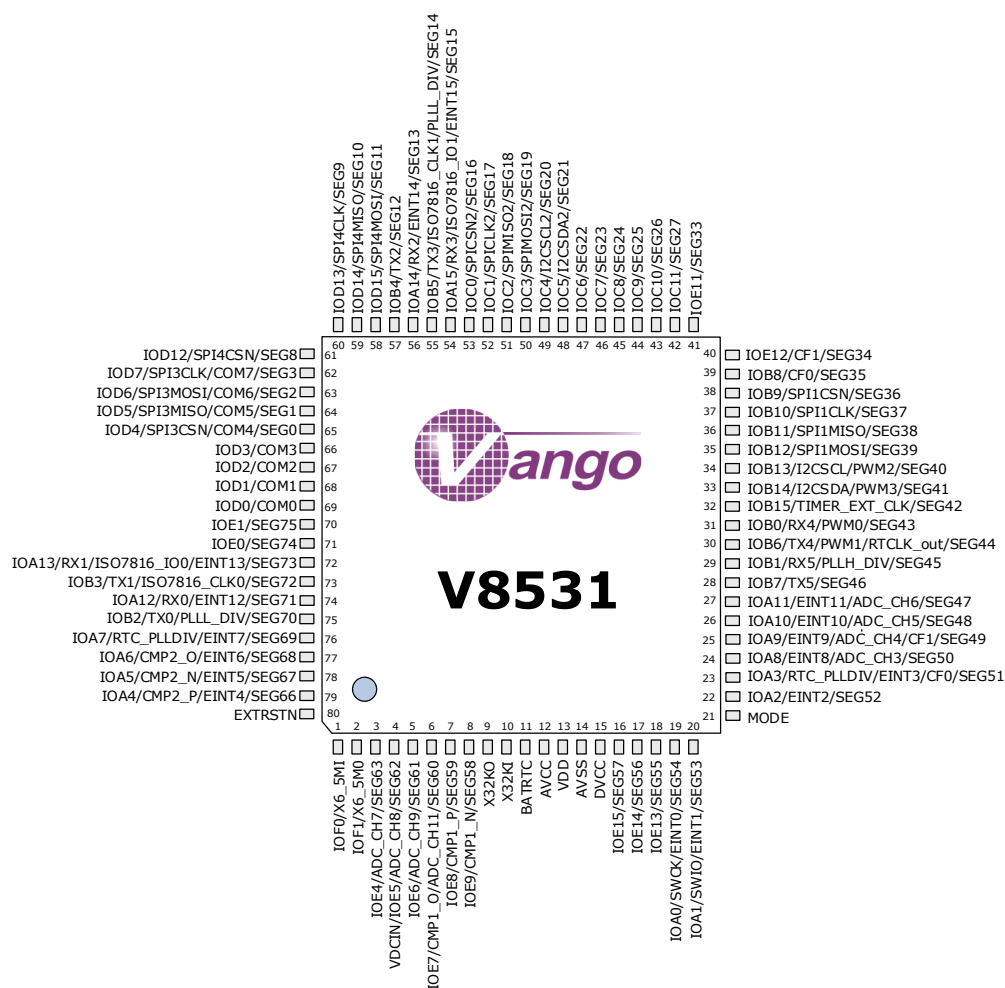
表18. 芯片可靠性分析结果

项目	规范	最小值	最大值
HBM	Mil-Std-883J Method 3015.9	-4KV	+4KV
MM	EDEC EIA/JESD22-A115	-300V	+300V
LATCH-UP	JEDEC EIA/JESD78E	-200mA	+200mA

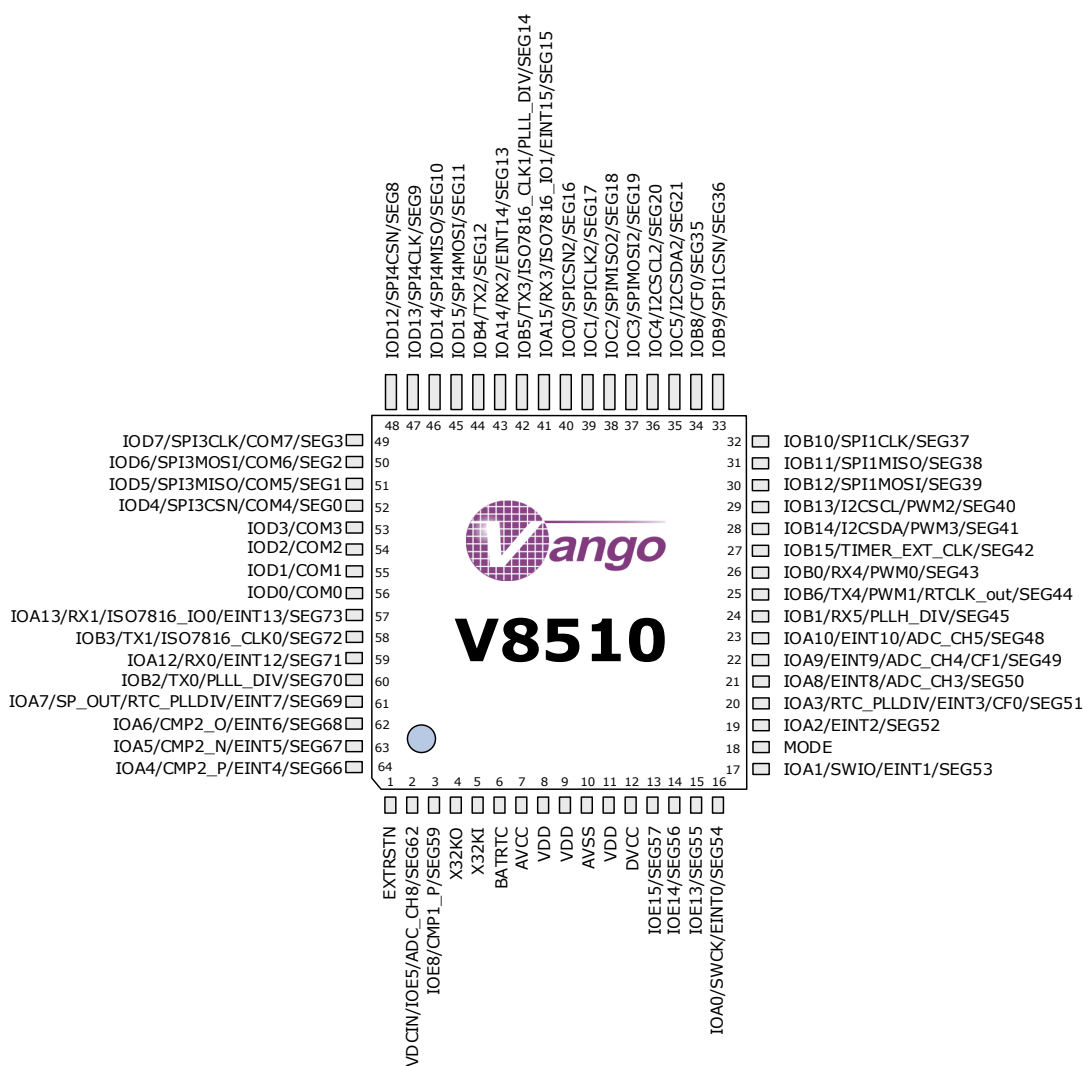
2.1 V8500 引脚分布图



2.2 V8531 引脚分布图



2.3 V8510 引脚分布图



2.4 V85XX 管脚描述

表19. V85XX 管脚描述

引脚编号			引脚名称	类型	描述
V8500	V8531	V8510			
1	1		IOF0	I/O	默认: IOF0 功能 1: 6.5536M 晶体输入
2	2		IOF1	I/O	默认: IOF1 功能 1: 6.5536M 晶体输出
3	3		IOE4	I/O	默认: IOE4 功能 1: ADC_CH7 输入 功能 2: SEG63
4	4	2	IOE5	I/O	默认: VDCIN 输入 功能 1: ADC_CH8 输入 功能 2: IOE5 功能 3: SEG62
5	5		IOE6	I/O	默认: IOE6 功能 1: ADC_CH9 输入 功能 2: SEG61
6	6		IOE7	I/O	默认: IOE7 功能 1: 比较器 1 输出 功能 2: ADC_CH11 输入 功能 3: SEG60
7	7	3	IOE8	I/O	默认: IOE8 功能 1: 比较器 1 P 输入 功能 2: SEG59
8	8		IOE9	I/O	默认: IOE9 功能 1: 比较器 1 N 输入 功能 2: SEG58
9			NC		NC
10	9	4	X32KO	O	32768hz 晶体输出引脚, 内部匹配电容为 12pF
11	10	5	X32KI	I	32768hz 晶体输入引脚
12	11	6	BATRTC	I	默认: ADC 通道 2 的输入, 用于电池电压测量。 比较器正端输入, 用于电池电压检测。
13	12	7	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。
14	13	8	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。 在 3.3V 系统应用中, VDD 应与 AVCC 需

					要连一起。
15		9	VDD	P	主电源输入引脚。
16	14	10	AVSS	G	模拟地
17		11	VDD	I	默认: ADC 通道 1 的输入, 用于电池电压测量。 比较器正端输入, 用于电池电压检测。
18			AVCC_OUT	O	默认: 高阻 功能: AVCC 电压输出引脚, 该电平与 AVCC 管脚保持一致。用户可以用来驱动小功率模块。
19	15	12	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。
20	16	13	IOE15	I/O	默认: IOE15 功能 1: SEG57
21	17	14	IOE14	I/O	默认: IOE14 功能 1: SEG56
22	18	15	IOE13	I/O	默认: IOE13 功能 2: SEG55
23	19	16	IOA0	I/O	默认: IOA0 (MODE=1), SWCK (MODE=0) 功能 1: EINT0 功能 2: SEG54
24	20	17	IOA1	I/O	默认: IOA1 (MODE=1), SWIO (MODE=0) 功能 1: EINT1 功能 2: SEG53
25	21	18	MODE	I	调试模式或正常模式选择。 0: 调试模式。 1: 正常模式。 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。
26	22	19	IOA2	I/O	默认: IOA2 功能 1: EINT2 功能 2: SEG52
27	23	20	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: CF0 功能 4: SEG51
28	24	21	IOA8	I/O	默认: IOA8

					功能 1: EINT8 功能 2: ADC_CH3 输入 功能 3: SEG50
29	25	22	IOA9	I/O	默认: IOA9 功能 1: EINT9 功能 2: ADC_CH4 输入 功能 3: CF1 功能 4: SEG49
30	26	23	IOA10	I/O	默认: IOA10 功能 1: EINT10 功能 2: ADC_CH5 输入 功能 3: SEG48
31	27		IOA11	I/O	默认: IOA11 功能 1: EINT11 功能 2: ADC_CH6 输入 功能 3: SEG47
32	28		IOB7	I/O	默认: IOB7 功能 1: UARTTXD5 功能 2: SEG46
33	29	24	IOB1	I/O	默认: IOB1 功能 1: UARTRXD5 功能 2: PLLH 经过分频器输出 功能 3: SEG45
34	30	25	IOB6	I/O	默认: IOB6 功能 1: UARTTXD4 功能 2: PWM1 输出 功能 3: RTCCLK 输出 功能 4: SEG44
35	31	26	IOB0	I/O	默认: IOB0 功能 1: UARTRXD4 功能 2: PWM0 输出 功能 3: SEG43
36	32	27	IOB15	I/O	默认: IOB15 功能 1: 定时器外部时钟输入. 功能 2: SEG42
37	33	28	IOB14	I/O	默认: IOB14 功能 1: I2CSDA 功能 2: PWM3 输出 功能 3: SEG41
38	34	29	IOB13	I/O	默认: IOB13 功能 1: I2CSCL 功能 2: PWM2 输出

					功能 3: SEG40
39	35	30	IOB12	I/O	默认: IOB12 功能 1: SPI1MOSI 功能 2: SEG39
40	36	31	IOB11	I/O	默认: IOB11 功能 1: SPI1MISO 功能 2: SEG38
41			AVSS	G	模拟地
42	37	32	IOB10	I/O	默认: IOB10 功能 1: SPI1CLK 功能 2: SEG37
43	38	33	IOB9	I/O	默认: IOB9 功能 1: SPI1CSN 功能 2: SEG36
44	39	34	IOB8	I/O	默认: IOB8 功能 1: CF0 功能 2: SEG35
45	40		IOE12	I/O	默认: IOE12 功能 1: CF1 功能 2: SEG34
46	41		IOE11	I/O	默认: IOE11 功能 1: SEG33
47			IOE10	I/O	默认: IOE10 功能 1: SEG32
48			IOC15	I/O	默认: IOC15 功能 1: SEG31
49			IOC14	I/O	默认: IOC14 功能 1: SEG30
50			IOC13	I/O	默认: IOC13 功能 1: SEG29
51			IOC12	I/O	默认: IOC12 功能 1: SEG28
52	42		IOC11	I/O	默认: IOC11 功能 1: SEG27
53	43		IOC10	I/O	默认: IOC10 功能 1: SEG26
54	44		IOC9	I/O	默认: IOC9 功能 1: SEG25
55	45		IOC8	I/O	默认: IOC8 功能 1: SEG24
56	46		IOC7	I/O	默认: IOC7 功能 1: SEG23

57	47		IOC6	I/O	默认: IOC6 功能 1: SEG22
58	48	35	IOC5	I/O	默认: IOC5 功能 1: I2CSDA 功能 2: SEG21
59			NC		NC
60	49	36	IOC4	I/O	默认: IOC4 功能 1: I2CSCL 功能 2: SEG20
61	50	37	IOC3	I/O	默认: IOC3 功能 1: SPI2MOSI 功能 2: SEG19
62	51	38	IOC2	I/O	默认: IOC2 功能 1: SPI2MISO 功能 2: SEG18
63	52	39	IOC1	I/O	默认: IOC1 功能 1: SPI2CLK 功能 2: SEG17
64	53	40	IOC0	I/O	默认: IOC0 功能 1: SPI2CSN 功能 2: SEG16
65	54	41	IOA15	I/O	默认: IOA15 功能 1: UARTRXD3 功能 2: ISO7816I/O1 功能 3: EINT15 功能 4: SEG15
66	55	42	IOB5	I/O	默认: IOB5 功能 1: UARTRXD3 功能 2: ISO7816CLK1 功能 3: PLLL_DIV 功能 4: SEG14
67			AVSS	G	模拟地
68	56	43	IOA14	I/O	默认: IOA14 功能 1: UARTRXD2 功能 2: EINT14 功能 3: SEG13
69	57	44	IOB4	I/O	默认: IOB4 功能 1: UARTRXD2 功能 2: SEG12
70	58	45	IOD15	I/O	默认: IOD15 功能 1: SEG11 功能 2: SPI4MOSI

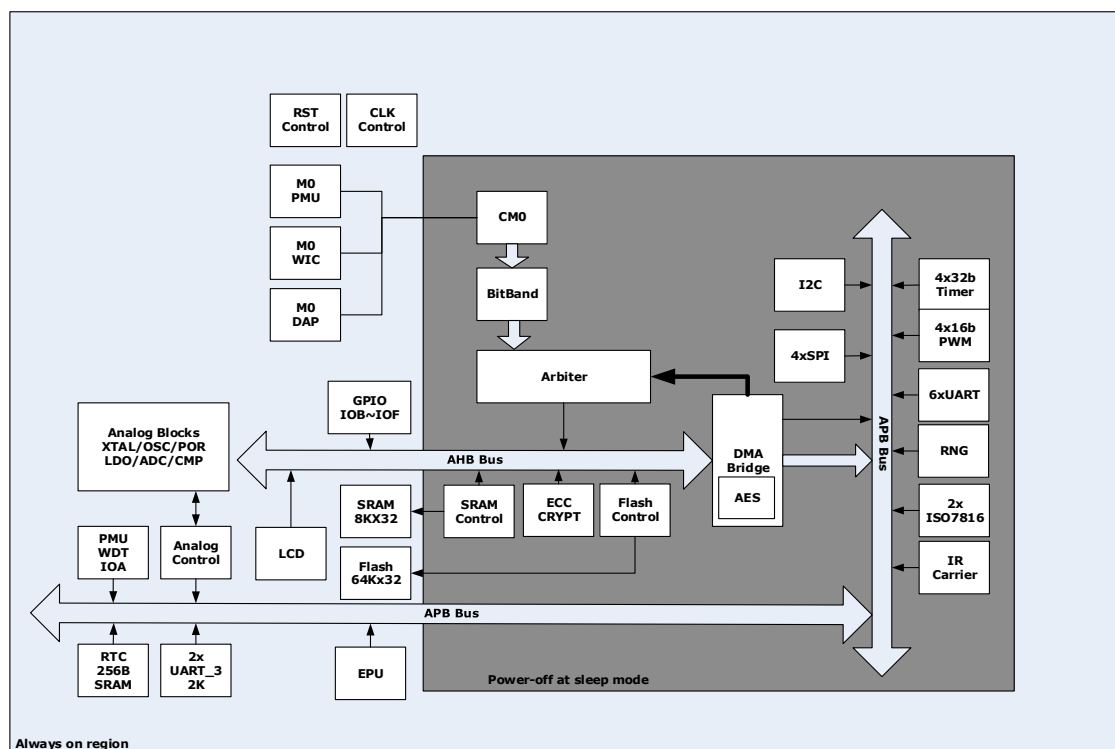
71	59	46	IOD14	I/O	默认: IOD14 功能 1: SEG10 功能 2: SPI4MISO
72	60	47	IOD13	I/O	默认: IOD13 功能 1: SEG9 功能 2: SPI4CLK
73	61	48	IOD12	I/O	默认: IOD12 功能 1: SEG8 功能 2: SPI4CSN
74			IOD11	I/O	默认: IOD11 功能 1: SEG7
75			VDDCAP	P	主电源去耦引脚, 推荐外接 0.1uF 电容。
76			IOD10	I/O	默认: IOD10 功能 1: SEG6
77			IOD9	I/O	默认: IOD9 功能 1: SEG5
78			IOD8	I/O	默认: IOD8 功能 1: SEG4
79	62	49	IOD7	I/O	默认: IOD7 功能 1: COM7/SEG3 功能 2: SPI3CLK
80	63	50	IOD6	I/O	默认: IOD6 功能 1: COM6/SEG2 功能 2: SPI3MOSI
81	64	51	IOD5	I/O	默认: IOD5 功能 1: COM5/SEG1 功能 2: SPI3MISO
82	65	52	IOD4	I/O	默认: IOD4 功能 1: COM4/SEG0 功能 2: SPI3CSN
83	66	53	IOD3	I/O	默认: IOD3 功能 1: COM3
84	67	54	IOD2	I/O	默认: IOD2 功能 1: COM2
85	68	55	IOD1	I/O	默认: IOD1 功能 1: COM1
86	69	56	IOD0	I/O	默认: IOD0 功能 1: COM0
87			IOE3	I/O	默认: IOE3 功能 1: SEG77
88			NC		NC
89			IOE2	I/O	默认: IOE2

					功能 1: SEG76
90	70		IOE1	I/O	默认: IOE1 功能 2: SEG75
91	71		IOE0	I/O	默认: IOE0 功能 2: SEG74
92	72	57	IOA13	I/O	默认: IOA13 功能 1: UARTRXD1 功能 2: ISO7816I/O0 功能 3: EINT13 功能 4: SEG73
93	73	58	IOB3	I/O	默认: IOB3 功能 1: UARTTXD1 功能 2: ISO7816CLK0 功能 3: SEG72
94	74	59	IOA12	I/O	默认: IOA12 功能 1: UARTRXD0 功能 2: EINT12 功能 3: SEG71
95	75	60	IOB2	I/O	默认: IOB2 功能 1: UARTTXD0 功能 2: PLLL_DIV 功能 3: SEG70
96	76	61	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
97	77	62	IOA6	I/O	默认: IOA6 功能 1: 比较器 2 输出 功能 2: EINT6 功能 3: SEG68
98	78	63	IOA5	I/O	默认: IOA5 功能 1: 比较器 2 N 输入 功能 2: EINT5 功能 3: SEG67
99	79	64	IOA4	I/O	默认: IOA4 功能 1: 比较器 2 P 输入 功能 2: EINT4 功能 3: SEG66
100	80	1	EXTRSTN	I	外部复位引脚, 低电平有效, RC 滤波电路推荐使用 51 K 欧电阻和 1 uF 电容。

3 功能框图

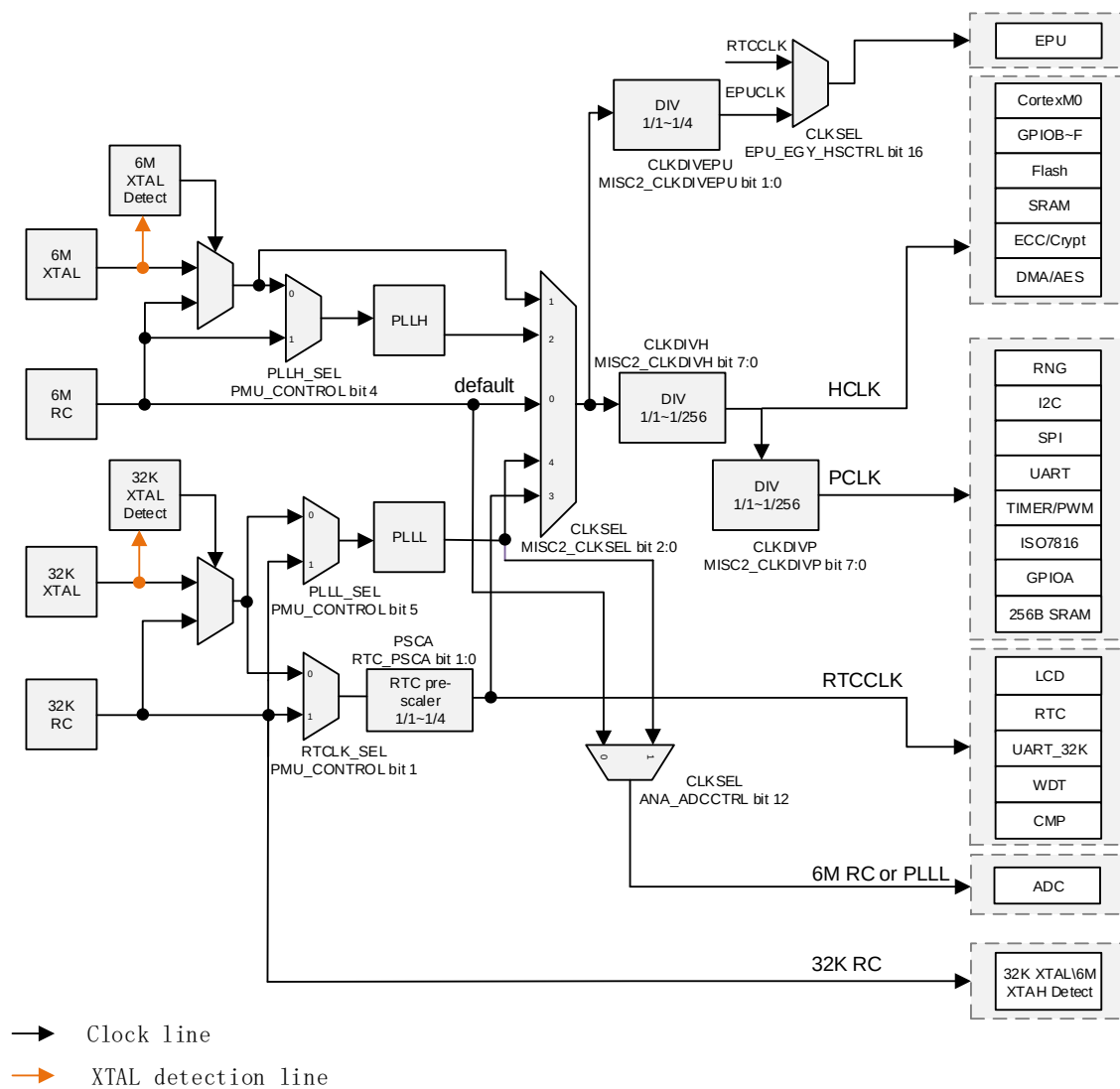
3.1 系统功能框图

图1. V85XX 系统功能框图



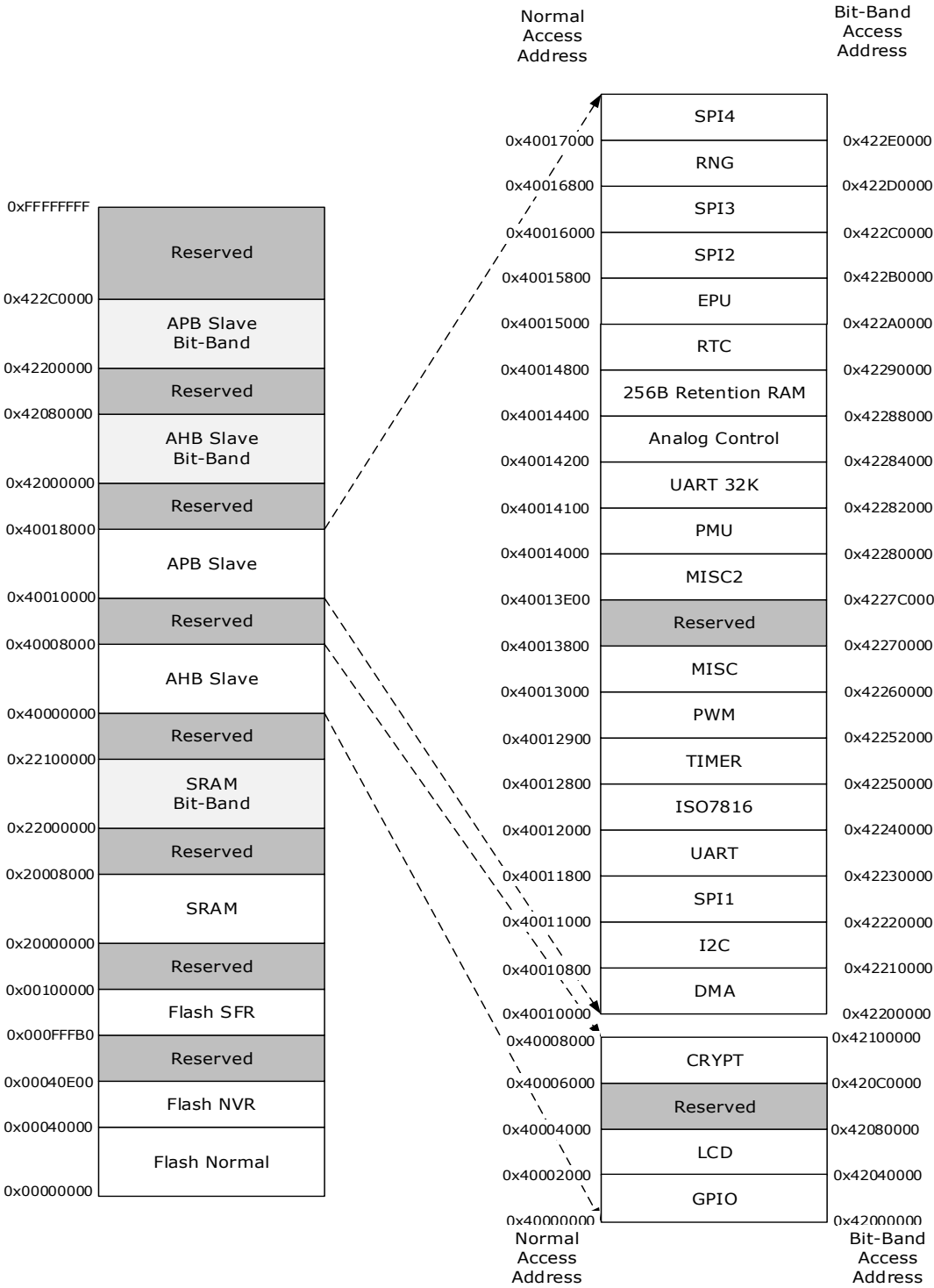
3.3 时钟功能框图

图3. V85XX 时钟功能框图



4 内存映射

图4. V85XX 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20. PMU 寄存器地址（PMU 基地址：0x40014000）

名称	类型	地址	描述	默认值
PMU_CONTROL	rw	0x0008	PMU 控制寄存器	0x0000
PMU_STS	rc_w1	0x000C	PMU 状态寄存器	0x0000074
PMU_IOAOEN	rw	0x0010	IOA 输出使能寄存器	0xFFFF
PMU_IOAIE	rw	0x0014	IOA 输入使能寄存器	0xFFFF
PMU_IOADAT	rw	0x0018	IOA 输出数据寄存器	0x0000
PMU_IOAATT	rw	0x001C	IOA 属性寄存器	0x0000
PMU_IOAWKUEN	rw	0x0020	IOA 唤醒使能寄存器	0x00000000
PMU_IOASTS	r	0x0024	IOA 输入状态寄存器	--
PMU_IOAINTSTS	rc_w1	0x0028	IOA 中断状态寄存器	0x0000
PMU_IOASEL	rw	0x0038	IOA 特殊功能选择寄存器	0x0000
VERSIONID	r	0x003C	V85XX 版本号	--
PMU_WDTPASS	rw	0x0040	看门狗解锁寄存器	0x00000000
PMU_WDTEN	rw	0x0044	看门狗定时器使能寄存器	0x1
PMU_WDTCCLR	w	0x0048	看门狗定时器清零寄存器	0x0000
PMU_IOANODEG	rw	0x0050	IOA 去抖动电路控制寄存器	0x0000
PMU_WDTSTS	rc_w1	0x0060	看门狗定时器状态寄存器	0x00

表21. PMU_RAM 寄存器地址（PMU_RAM 基地址：0x40014400）

名称	类型	地址	描述	默认值
PMU_RAM0	rw	0x0000	PMU 32 位 RAM0	--
PMU_RAM1	rw	0x0004	PMU 32 位 RAM1	--
PMU_RAM2	rw	0x0008	PMU 32 位 RAM2	--
...	...	...	...	
PMU_RAM63	rw	0x00FC	PMU 32 位 RAM63	--

4.1.2 ANA 寄存器地址

表22. 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	rw	0x0000	模拟寄存器 0	0x00
ANA_REG1	rw	0x0004	模拟寄存器 1	0x00

ANA_REG2	rw	0x0008	模拟寄存器 2	0x00
ANA_REG3	rw	0x000C	模拟寄存器 3	0x00
ANA_REG4	rw	0x0010	模拟寄存器 4	0x00
ANA_REG5	rw	0x0014	模拟寄存器 5	0x00
ANA_REG6	rw	0x0018	模拟寄存器 6	0x00
ANA_REG7	rw	0x001C	模拟寄存器 7	0x00
ANA_REG8	rw	0x0020	模拟寄存器 8	0x00
ANA_REG9	rw	0x0024	模拟寄存器 9	0x00
ANA_REGA	rw	0x0028	模拟寄存器 10	0x00
ANA_REGB	rw	0x002C	模拟寄存器 11	来自 FLASH，用户不得修改
ANA_REGC	rw	0x0030	模拟寄存器 12	来自 FLASH，用户不得修改
ANA_REGD	rw	0x0034	模拟寄存器 13	来自 FLASH，用户不得修改
ANA_REGE	rw	0x0038	模拟寄存器 14	来自 FLASH，用户不得修改
ANA_REGF	rw	0x003C	模拟寄存器 15	0x00
ANA_REG10	rw	0x0040	模拟寄存器 16	来自 FLASH
ANA_REG11	rw	0x0044	模拟寄存器 17	来自 FLASH
ANA_CTRL	rw	0x0050	模拟控制寄存器	0x00000000
ANA_CMPOUT	r	0x0054	比较器结果寄存器	0x0030
ANA_INTSTS	rc_w1	0x0060	模拟中断状态寄存器	0x0000
ANA_INTEN	rw	0x0064	模拟中断使能寄存器	0x0000
ANA_ADCCTRL	rw	0x0068	ADC 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_ADCDATAA	r	0x0098	ADC 通道 10 数据寄存器	--
ANA_ADCDATAB	r	0x009C	ADC 通道 11 数据寄存器	--
ANA_CMPCNT1	rc_w1	0x00B0	比较器 1 计数器	0x00000000
ANA_CMPCNT2	rc_w1	0x00B4	比较器 2 计数器	0x00000000

4.1.3 RTC 寄存器地址

表23. RTC 控制寄存器地址（RTC 基地址：0x40014800）

名称	类型	地址	描述	默认值	Write Protect	Read Protect
RTC_SEC	rw	0x0000	RTC 秒钟寄存器	--	V	V
RTC_MIN	rw	0x0004	RTC 分钟寄存器	--	V	V
RTC_HOUR	rw	0x0008	RTC 时寄存器	--	V	V
RTC_DAY	rw	0x000C	RTC 天寄存器	--	V	V
RTC_WEEK	rw	0x0010	RTC 周寄存器	--	V	V
RTC_MON	rw	0x0014	RTC 月寄存器	--	V	V
RTC_YEAR	rw	0x0018	RTC 年寄存器	--	V	V
RTC_WKUSEC	rw	0x0020	RTC 唤醒秒钟寄存器	0x00	V	
RTC_WKUMIN	rw	0x0024	RTC 唤醒分钟寄存器	0x00	V	
RTC_WKUHOURL	rw	0x0028	RTC 唤醒时寄存器	0x00	V	
RTC_WKUCNT	rw	0x002C	RTC 唤醒计数器寄存器	0x00000000	V	
RTC_CAL	rw	0x0030	RTC 校正寄存器	--	V	
RTC_DIV	rw	0x0034	RTC_PLLDIV 分频寄存器	0x00000000		
RTC_CTL	rw	0x0038	RTC_PLLDIV 分频控制寄存器	0x0		
RTC_PWD	rw	0x0044	RTC 密码控制寄存器	0x00000000		
RTC_CE	rw	0x0048	RTC 写使能控制寄存器	0x0		
RTC_LOAD	rw	0x004C	RTC 读使能控制寄存器	--		
RTC_INTSTS	rw	0x0050	RTC 中断状态控制寄存器	0x000		
RTC_INTEN	rw	0x0054	RTC 中断使能控制寄存器	0x000		
RTC_PSCA	rw	0x0058	RTC 时钟预分频控制寄存器	0x0	V	
RTC_ACTI	rw	0x0084	RTC 晶体顶点温度寄存器	0x1800	V	
RTC_ACF200	rw	0x0088	RTC200 倍频率校正控制寄存器	0x640000	V	
RTC_ACADCW	rw	0x008C	RTC 手动 ADC 值校正控制寄存器	0x0000	V	
RTC_ACP0	rw	0x0090	RTC 校正参数 0 控制寄存器	0x0000	V	

RTC_ACP1	rw	0x0094	RTC 校正参数 1 控制寄存器	0x0000	V	
RTC_ACP2	rw	0x0098	RTC 校正参数 2 控制寄存器	0x0000	V	
RTC_ACP3	r	0x009C	RTC 校正参数 3 控制寄存器	0x0000		
RTC_ACP4	rw	0x00A0	RTC 校正参数 4 控制寄存器	0x0000	V	
RTC_ACP5	rw	0x00A4	RTC 校正参数 5 控制寄存器	0x0000	V	
RTC_ACP6	rw	0x00A8	RTC 校正参数 6 控制寄存器	0x0000	V	
RTC_ACP7	rw	0x00AC	RTC 校正参数 7 控制寄存器	0x0000	V	
RTC_ACK1	rw	0x00B0	RTC 校正参数 k1 控制寄存器	0x0000	V	
RTC_ACK2	rw	0x00B4	RTC 校正参数 k2 控制寄存器	0x0000	V	
RTC_ACK3	rw	0x00B8	RTC 校正参数 k3 控制寄存器	0x0000	V	
RTC_ACK4	rw	0x00BC	RTC 校正参数 k4 控制寄存器	0x0000	V	
RTC_ACK5	rw	0x00C0	RTC 校正参数 k5 控制寄存器	0x0000	V	
RTC_ACTEMP	r	0x00C4	记录 RTC 校正温度计算值	0x0000		
RTC_ACPPM	r	0x00C8	记录 RTC 校正 PPM 计算值	0x0000		
RTC_WKUCNTR	r	0x00CC	记录当前 WKUCNT 值	0x000000		
RTC_ACKTEMP	rw	0x00D0	RTC 校正温度分段区间寄存器	0x3C2800EC	V	

4.1.4 FLASH 寄存器地址

表24. 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_STS	r	0xFFFFBC	FLASH 编程状态寄存器	0x00
FLASH_INT	rc_w1	0xFFFFCC	FLASH 校验和中断状态	0x0
FLASH_CSSADDR	rw	0xFFFFD0	FLASH 校验和开始地址	0x00000
FLASH_CSEADDR	rw	0xFFFFD4	FLASH 校验和结束地址	0x3FFFC

FLASH_CSVALUE	r	0xFFFD8	FLASH 校验和值读取寄存器	--
FLASH_CSCVALUE	rw	0xFFFD8	FLASH 校验和比较值设置寄存器	0x00000000
FLASH_PASS	rw	0xFFFE0	FLASH 密码寄存器	0x00000000
FLASH_CTRL	rw	0xFFFE4	FLASH 控制寄存器	0x0
FLASH_PGADDR	rw	0xFFFE8	FLASH 程序地址寄存器	0x00000
FLASH_PGDATA	rw	0xFFFE8	FLASH 程序字数据寄存器	--
FLASH_PGB0	rw	0xFFFE8	FLASH 程序字节数据寄存器 0	--
FLASH_PGB1	rw	0xFFFE8	FLASH 程序字节数据寄存器 1	--
FLASH_PGB2	rw	0xFFFE8	FLASH 程序字节数据寄存器 2.	--
FLASH_PGB3	rw	0xFFFE8	FLASH 程序字节数据寄存器 3	--
FLASH_PGHW0	rw	0xFFFE8	FLASH 程序半字数据寄存器 0	--
FLASH_PGHW1	rw	0xFFFE8	FLASH 程序半字数据寄存器 1	--
FLASH_SERASE	rw	0xFFFF4	FLASH 扇区擦除控制寄存器	0x00000000
FLASH_CERASE	rw	0xFFFF8	FLASH 芯片擦除控制寄存器	0x00000000
FLASH_DSTB	rw	0xFFFFC	FLASH deep standby 控制寄存器	0x00000000

4.1.5 GPIO 寄存器地址

表25. 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	rw	0x0020	IOB 输出使能寄存器	0xFFFF
IOB_IE	rw	0x0024	IOB 输入使能寄存器	0xFFFF
IOB_DAT	rw	0x0028	IOB 输出数据寄存器	0x0000
IOB_ATT	rw	0x002C	IOB 属性寄存器	0x0000
IOB_STS	r	0x0030	IOB 输入状态寄存器	--
IOC_OEN	rw	0x0040	IOC 输出使能寄存器	0xFFFF
IOC_IE	rw	0x0044	IOC 输入使能寄存器	0xFFFF
IOC_DAT	rw	0x0048	IOC 输出数据寄存器	0x0000
IOC_ATT	rw	0x004C	IOC 属性寄存器	0x0000
IOC_STS	r	0x0050	IOC 输入状态寄存器	--
IOD_OEN	rw	0x0060	IOD 输出使能寄存器	0xFFFF
IOD_IE	rw	0x0064	IOD 输入使能寄存器	0xFFFF
IOD_DAT	rw	0x0068	IOD 输出数据寄存器	0x0000
IOD_ATT	rw	0x006C	IOD 属性寄存器	0x0000
IOD_STS	r	0x0070	IOD 输入状态寄存器	--
IOE_OEN	rw	0x0080	IOE 输出使能寄存器	0xFFFF
IOE_IE	rw	0x0084	IOE 输入使能寄存器	0xFFFF
IOE_DAT	rw	0x0088	IOE 输出数据寄存器	0x0000
IOE_ATT	rw	0x008C	IOE 属性寄存器	0x0000
IOE_STS	r	0x0090	IOE 输入状态寄存器	--
IOF_OEN	rw	0x00A0	IOF 输出使能寄存器	0x1F

IOF_IE	rw	0x00A4	IOF 输入使能寄存器	0x1F
IOF_DAT	rw	0x00A8	IOF 输出数据寄存器	0x0
IOF_ATT	rw	0x00AC	IOF 属性寄存器	0x0
IOF_STS	r	0x00B0	IOF 输入状态寄存器	--
IOB_SEL	rw	0x00C0	IOB 特殊功能选择寄存器	0x00
IOE_SEL	rw	0x00CC	IOE 特殊功能选择寄存器	0x00
IO_MISC	rw	0x00E0	IO MISC 控制寄存器	0x00

4.1.6 DMA 寄存器地址

表26. 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	rw	0x0000	DMA 中断使能寄存器	0x000
DMA_STS	rw	0x0004	DMA 状态寄存器	0x0000
DMA_C0CTL	rw	0x0010	DMA 通道 0 控制寄存器	0x00000000
DMA_C0SRC	rw	0x0014	DMA 通道 0 源寄存器	0x00000000
DMA_C0DST	rw	0x0018	DMA 通道 0 目的寄存器	0x00000000
DMA_C0LEN	r	0x001C	DMA 通道 0 传输长度寄存器	0x0000
DMA_C1CTL	rw	0x0020	DMA 通道 1 控制寄存器	0x00000000
DMA_C1SRC	rw	0x0024	DMA 通道 1 源寄存器	0x00000000
DMA_C1DST	rw	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	rw	0x0030	DMA 通道 2 控制寄存器	0x00000000
DMA_C2SRC	rw	0x0034	DMA 通道 2 源寄存器	0x00000000
DMA_C2DST	rw	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	rw	0x0040	DMA 通道 3 控制寄存器	0x00000000
DMA_C3SRC	rw	0x0044	DMA 通道 3 源寄存器	0x00000000
DMA_C3DST	rw	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	rw	0x0050	DMA AES 控制寄存器	0x00000000
DMA_AESKEY0	rw	0x0060	DMA AES 密钥 0 寄存器	0x00000000
DMA_AESKEY1	rw	0x0064	DMA AES 密钥 1 寄存器	0x00000000
DMA_AESKEY2	rw	0x0068	DMA AES 密钥 2 寄存器	0x00000000
DMA_AESKEY3	rw	0x006C	DMA AES 密钥 3 寄存器	0x00000000
DMA_AESKEY4	rw	0x0070	DMA AES 密钥 4 寄存器	0x00000000
DMA_AESKEY5	rw	0x0074	DMA AES 密钥 5 寄存器	0x00000000
DMA_AESKEY6	rw	0x0078	DMA AES 密钥 6 寄存器	0x00000000
DMA_AESKEY7	rw	0x007C	DMA AES 密钥 7 寄存器	0x00000000

4.1.7 UART 寄存器地址

表27. UART 寄存器地址（UART 基地址：0x40011800）

名称	类型	地址	描述	默认值
UART0_DATA	rw	0x0000	UART0 数据寄存器	0x00
UART0_STATE	rc_w1	0x0004	UART0 状态寄存器	0x00
UART0_CTRL	rw	0x0008	UART0 控制寄存器	0x000
UART0_INTSTS	rc_w1	0x000C	UART0 中断状态寄存器	0x00
UART0_BAUDDIV	rw	0x0010	UART0 波特率分频寄存器	0x00000
UART0_CTRL2	rw	0x0014	UART0 控制寄存器 2	0x0
UART1_DATA	rw	0x0020	UART1 数据寄存器	0x00
UART1_STATE	rc_w1	0x0024	UART1 状态寄存器	0x00
UART1_CTRL	rw	0x0028	UART1 控制寄存器	0x000
UART1_INTSTS	rc_w1	0x002C	UART1 中断状态寄存器	0x00
UART1_BAUDDIV	rw	0x0030	UART1 波特率分频寄存器	0x00000
UART1_CTRL2	rw	0x0034	UART1 控制寄存器 2	0x0
UART2_DATA	rw	0x0040	UART2 数据寄存器	0x00
UART2_STATE	rc_w1	0x0044	UART2 状态寄存器	0x00
UART2_CTRL	rw	0x0048	UART2 控制寄存器	0x000
UART2_INTSTS	rc_w1	0x004C	UART2 中断状态寄存器	0x00
UART2_BAUDDIV	rw	0x0050	UART2 波特率分频寄存器	0x00000
UART2_CTRL2	rw	0x0054	UART2 控制寄存器 2	0x0
UART3_DATA	rw	0x0060	UART3 数据寄存器	0x00
UART3_STATE	rc_w1	0x0064	UART3 状态寄存器	0x00
UART3_CTRL	rw	0x0068	UART3 控制寄存器	0x000
UART3_INTSTS	rc_w1	0x006C	UART3 中断状态寄存器	0x00
UART3_BAUDDIV	rw	0x0070	UART3 波特率分频寄存器	0x00000
UART3_CTRL2	rw	0x0074	UART3 控制寄存器 2	0x0
UART4_DATA	rw	0x0080	UART4 数据寄存器	0x00
UART4_STATE	rc_w1	0x0084	UART4 状态寄存器	0x00
UART4_CTRL	rw	0x0088	UART4 控制寄存器	0x000
UART4_INTSTS	rc_w1	0x008C	UART4 中断状态寄存器	0x00
UART4_BAUDDIV	rw	0x0090	UART4 波特率分频寄存器	0x00000
UART4_CTRL2	rw	0x0094	UART4 控制寄存器 2	0x0
UART5_DATA	rw	0x00A0	UART5 数据寄存器	0x00
UART5_STATE	rc_w1	0x00A4	UART5 状态寄存器	0x00
UART5_CTRL	rw	0x00A8	UART5 控制寄存器	0x000
UART5_INTSTS	rc_w1	0x00AC	UART5 中断状态寄存器	0x00
UART5_BAUDDIV	rw	0x00B0	UART5 波特率分频寄存器	0x00000
UART5_CTRL2	rw	0x00B4	UART5 控制寄存器 2	0x0

4.1.8 UART32K 寄存器地址

表28. UART32K 寄存器地址（UART32K 基地址：0x40014100）

名称	类型	地址	描述	默认值
U32K0_CTRL0	rw	0x0000	UART32K0 控制寄存器 0	0x000
U32K0_CTRL1	rw	0x0004	UART32K0 控制寄存器 1	0x00
U32K0_PHASE	rw	0x0008	UART32K0 波特率控制寄存器	0x4B00
U32K0_DATA	r	0x000C	UART32K0 接收数据缓存区	--
U32K0_STS	rc_w1	0x0010	UART32K0 中断状态寄存器	0x0
U32K1_CTRL0	rw	0x0080	UART32K1 控制寄存器 0	0x000
U32K1_CTRL1	rw	0x0084	UART32K1 控制寄存器 1	0x00
U32K1_PHASE	rw	0x0088	UART32K1 波特率控制寄存器	0x4B00
U32K1_DATA	r	0x008C	UART32K1 接收数据缓存区	--
U32K1_STS	rc_w1	0x0090	UART32K1 中断状态寄存器	0x0

4.1.9 ISO7816 寄存器地址

表29. ISO7816 寄存器地址（ISO7816 基地址：0x40012000）

名称	类型	地址	描述	默认值
ISO78160_BAUDDIVL	rw	0x0004	ISO78160 波特率低字节寄存器	0x00
ISO78160_BAUDDIVH	rw	0x0008	ISO78160 波特率高字节寄存器	0x00
ISO78160_DATA	rw	0x000C	ISO78160 数据寄存器	0x00
ISO78160_INFO	rc_w1	0x0010	ISO78160 信息寄存器	0x00
ISO78160_CFG	rw	0x0014	ISO78160 控制寄存器	0x00
ISO78160_CLK	rw	0x0018	ISO78160 时钟分频器控制寄存器	0x00
ISO78161_BAUDDIVL	rw	0x0044	ISO78161 波特率低字节寄存器	0x00
ISO78161_BAUDDIVH	rw	0x0048	ISO78161 波特率高字节寄存器	0x00
ISO78161_DATA	rw	0x004C	ISO78161 数据寄存器	0x00
ISO78161_INFO	rc_w1	0x0050	ISO78161 信息寄存器	0x00
ISO78161_CFG	rw	0x0054	ISO78161 控制寄存器	0x00
ISO78161_CLK	rw	0x0058	ISO78161 时钟分频器控制寄存器	0x00

4.1.10 TIMER/PWM 寄存器地址

表30. 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	rw	0x0000	Timer0 控制寄存器	0x0
TMR0_VALUE	rw	0x0004	Timer0 实时计数寄存器	0x00000000
TMR0_RELOAD	rw	0x0008	Timer0 重载寄存器	0x00000000
TMR0_INT	rc_w1	0x000C	Timer0 中断状态寄存器	0x0

TMR1_CTRL	rw	0x0020	Timer1 控制寄存器	0x0
TMR1_VALUE	rw	0x0024	Timer1 实时计数寄存器	0x00000000
TMR1_RELOAD	rw	0x0028	Timer1 重载寄存器	0x00000000
TMR1_INT	rc_w1	0x002C	Timer1 中断状态寄存器	0x0
TMR2_CTRL	rw	0x0040	Timer2 控制寄存器	0x0
TMR2_VALUE	rw	0x0044	Timer2 实时计数寄存器	0x00000000
TMR2_RELOAD	rw	0x0048	Timer2 重载寄存器	0x00000000
TMR2_INT	rc_w1	0x004C	Timer2 中断状态寄存器	0x0
TMR3_CTRL	rw	0x0060	Timer3 控制寄存器	0x0
TMR3_VALUE	rw	0x0064	Timer3 实时计数寄存器	0x00000000
TMR3_RELOAD	rw	0x0068	Timer3 重载寄存器	0x00000000
TMR3_INT	rc_w1	0x006C	Timer3 中断状态寄存器	0x0

表31. 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	rw	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	rw	0x0008	PWM Timer0 比较寄存器 0	0x000
PWM0_CCTL1	rw	0x000C	PWM Timer0 比较寄存器 1	0x000
PWM0_CCTL2	rw	0x0010	PWM Timer0 比较寄存器 2	0x000
PWM0_CCR0	rw	0x0014	PWM Timer0 比较数据寄存器 0	0x0000
PWM0_CCR1	rw	0x0018	PWM Timer0 比较数据寄存器 1	0x0000
PWM0_CCR2	rw	0x001C	PWM Timer0 比较数据寄存器 2	0x0000
PWM1_CTL	rw	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	rw	0x0028	PWM Timer1 比较数据寄存器 0	0x000
PWM1_CCTL1	rw	0x002C	PWM Timer1 比较数据寄存器 1	0x000
PWM1_CCTL2	rw	0x0030	PWM Timer1 比较数据寄存器 2	0x000
PWM1_CCR0	rw	0x0034	PWM Timer1 比较数据寄存器 0	0x0000
PWM1_CCR1	rw	0x0038	PWM Timer1 比较数据寄存器 1	0x0000
PWM1_CCR2	rw	0x003C	PWM Timer1 比较数据寄存器 2	0x0000
PWM2_CTL	rw	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	rw	0x0048	PWM Timer2 比较控制寄存器 0	0x000
PWM2_CCTL1	rw	0x004C	PWM Timer2 比较控制寄存器 1	0x000
PWM2_CCTL2	rw	0x0050	PWM Timer2 比较控制寄存器 2	0x000
PWM2_CCR0	rw	0x0054	PWM Timer2 比较数据寄存器 0	0x0000
PWM2_CCR1	rw	0x0058	PWM Timer2 比较数据寄存器 1	0x0000
PWM2_CCR2	rw	0x005C	PWM Timer2 比较数据寄存器 2	0x0000
PWM3_CTL	rw	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	rw	0x0068	PWM Timer3 比较控制寄存器 0	0x000
PWM3_CCTL1	rw	0x006C	PWM Timer3 比较控制寄存器 1	0x000
PWM3_CCTL2	rw	0x0070	PWM Timer3 比较控制寄存器 2	0x000
PWM3_CCR0	rw	0x0074	PWM Timer3 比较数据寄存器 0	0x0000
PWM3_CCR1	rw	0x0078	PWM Timer3 比较数据寄存器 1	0x0000
PWM3_CCR2	rw	0x007C	PWM Timer3 比较数据寄存器 2	0x0000
PWM_O_SEL	rw	0x00F0	PWM 输出选择寄存器	0xDB51

4.1.11 LCD 寄存器

表32. LCD 寄存器地址（LCD 基地址：0x40002000）

名称	类型	地址	描述	默认值
LCD_FB00	rw	0x0000	LCD 帧缓存 0 寄存器	--
LCD_FB01	rw	0x0004	LCD 帧缓存 1 寄存器	--
LCD_FB02	rw	0x0008	LCD 帧缓存 2 寄存器	--
LCD_FB03	rw	0x000C	LCD 帧缓存 3 寄存器	--
LCD_FB04	rw	0x0010	LCD 帧缓存 4 寄存器	--
LCD_FB05	rw	0x0014	LCD 帧缓存 5 寄存器	--
LCD_FB06	rw	0x0018	LCD 帧缓存 6 寄存器	--
LCD_FB07	rw	0x001C	LCD 帧缓存 7 寄存器	--
LCD_FB08	rw	0x0020	LCD 帧缓存 8 寄存器	--
LCD_FB09	rw	0x0024	LCD 帧缓存 9 寄存器	--
LCD_FB0A	rw	0x0028	LCD 帧缓存 10 寄存器	--
LCD_FB0B	rw	0x002C	LCD 帧缓存 11 寄存器	--
LCD_FB0C	rw	0x0030	LCD 帧缓存 12 寄存器	--
LCD_FB0D	rw	0x0034	LCD 帧缓存 13 寄存器	--
LCD_FB0E	rw	0x0038	LCD 帧缓存 14 寄存器	--
LCD_FB0F	rw	0x003C	LCD 帧缓存 15 寄存器	--
LCD_FB10	rw	0x0040	LCD 帧缓存 16 寄存器	--
LCD_FB11	rw	0x0044	LCD 帧缓存 17 寄存器	--
LCD_FB12	rw	0x0048	LCD 帧缓存 18 寄存器	--
LCD_FB13	rw	0x004C	LCD 帧缓存 19 寄存器	--
LCD_FB14	rw	0x0050	LCD 帧缓存 20 寄存器	--
LCD_FB15	rw	0x0054	LCD 帧缓存 21 寄存器	--
LCD_FB16	rw	0x0058	LCD 帧缓存 22 寄存器	--
LCD_FB17	rw	0x005C	LCD 帧缓存 23 寄存器	--
LCD_FB18	rw	0x0060	LCD 帧缓存 24 寄存器	--
LCD_FB19	rw	0x0064	LCD 帧缓存 25 寄存器	--
LCD_FB1A	rw	0x0068	LCD 帧缓存 26 寄存器	--
LCD_FB1B	rw	0x006C	LCD 帧缓存 27 寄存器	--
LCD_FB1C	rw	0x0070	LCD 帧缓存 28 寄存器	--

LCD_FB1D	rw	0x0074	LCD 帧缓存 29 寄存器	--
LCD_FB1E	rw	0x0078	LCD 帧缓存 30 寄存器	--
LCD_FB1F	rw	0x007C	LCD 帧缓存 31 寄存器	--
LCD_FB20	rw	0x0080	LCD 帧缓存 32 寄存器	--
LCD_FB21	rw	0x0084	LCD 帧缓存 33 寄存器	--
LCD_FB22	rw	0x0088	LCD 帧缓存 34 寄存器	--
LCD_FB23	rw	0x008C	LCD 帧缓存 35 寄存器	--
LCD_FB24	rw	0x0090	LCD 帧缓存 36 寄存器	--
LCD_FB25	rw	0x0094	LCD 帧缓存 37 寄存器	--
LCD_FB26	rw	0x0098	LCD 帧缓存 38 寄存器	--
LCD_FB27	rw	0x009C	LCD 帧缓存 39 寄存器	--
LCD_CTRL	rw	0x0100	LCD 控制寄存器	0x00
LCD_CTRL2	rw	0x0104	LCD 控制寄存器 2	0x0000
LCD_SEGCTRL0	rw	0x0108	LCDSEG 使能控制寄存器 0	0x00000000
LCD_SEGCTRL1	rw	0x010C	LCDSEG 使能控制寄存器 1	0x00000000
LCD_SEGCTRL2	rw	0x0110	LCDSEG 使能控制寄存器 2	0x00000000

4.1.12 SPI 寄存器地址

表33. SPI1 寄存器地址（SPI1 基地址：0x40011000）

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

表34. SPI2 寄存器地址（SPI2 基地址：0x40015800）

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

表35. SPI3 寄存器地址（SPI3 基地址：0x40016000）

名称	类型	地址	描述	默认值
SPI3_CTRL	rw	0x0000	SPI3 控制寄存器	0x0000

SPI3_TXSTS	rw	0x0004	SPI3 发送状态寄存器	0x8200
SPI3_TXDAT	rw	0x0008	SPI3 发送 FIFO 寄存器	--
SPI3_RXSTS	rw	0x000C	SPI3 接收状态寄存器	0x0000
SPI3_RXDAT	rw	0x0010	SPI3 接收 FIFO 寄存器	--
SPI3_MISC	rw	0x0014	SPI3 Misc 控制寄存器	0x0003

表36. SPI4 寄存器地址（SPI4 基地址：0x40017000）

名称	类型	地址	描述	默认值
SPI4_CTRL	rw	0x0000	SPI4 控制寄存器	0x0000
SPI4_TXSTS	rw	0x0004	SPI4 发送状态寄存器	0x8200
SPI4_TXDAT	rw	0x0008	SPI4 发送 FIFO 寄存器	--
SPI4_RXSTS	rw	0x000C	SPI4 接收状态寄存器	0x0000
SPI4_RXDAT	rw	0x0010	SPI4 接收 FIFO 寄存器	--
SPI4_MISC	rw	0x0014	SPI4 Misc 控制寄存器	0x0003

4.1.13 I2C 寄存器地址

表37. I2C 寄存器地址（I2C 基地址：0x40010800）

名称	类型	地址	描述	默认值
I2C_DATA	rw	0x0000	I2C 数据寄存器	0x00
I2C_ADDR	rw	0x0004	I2C 地址寄存器	0x00
I2C_CTRL	rw	0x0008	I2C 控制/状态寄存器	0x00
I2C_STS	rw	0x000c	I2C 状态寄存器	0xF8
I2C_CTRL2	rw	0x0018	I2C 中断使能寄存器	0x0

4.1.14 MISC 寄存器地址

表38. MISC 寄存器地址（MISC 基地址：0x40013000）

名称	类型	地址	描述	默认值
MISC_SRAMINT	rc_w1	0x0000	SRAM 奇偶校验错误中断状态寄存器	0x00
MISC_SRAMINIT	rw	0x0004	SRAM 初始化寄存器	0x01
MISC_PARERR	r	0x0008	SRAM 奇偶校验错误地址寄存器	0x000
MISC_IREN	rw	0x000C	IR 使能控制寄存器	0x00
MISC_DUTYL	rw	0x0010	IR 调制波低电平宽度控制寄存器	0x0000
MISC_DUTYH	rw	0x0014	IR 调制波高电平宽度控制寄存器	0x0000
MISC_IRQLAT	rw	0x0018	Cortex-M0 IRQ 延迟控制寄存器	0x000
MISC_HIADDR	r	0x0020	AHB 无效访问地址	--
MISC_PIADDR	r	0x0024	APB 无效访问地址	--

表39. MISC2 寄存器地址（MISC2 基地址：0x40013E00）

名称	类型	地址	描述	默认值
MISC2_FLASHWC	rw	0x0000	FLASH 等待周期寄存器	0x2100
MISC2_CLKSEL	rw	0x0004	时钟选择寄存器	0x0
MISC2_CLKDIVH	rw	0x0008	AHB 时钟分频控制寄存器	0x00
MISC2_CLKDIVP	rw	0x000C	APB 时钟分频控制寄存器	0x01
MISC2_HCLKEN	rw	0x0010	AHB 时钟使能控制寄存器	0x1FF
MISC2_PCLKEN	rw	0x0014	APB 时钟使能控制寄存器	0xF83FFFFFFF
MISC2_EPUCLKCTRL	rw	0x0020	EPU 时钟使能控制寄存器	0x0
MISC2_RSTGPU	rw	0x0024	EPU 复位控制寄存器	0x0

4.1.15 CRYPT 寄存器地址

表40. CRYPT 寄存器地址（CRYPT 基地址：0x40006000）

名称	类型	地址	描述	默认值
CRYPT_CTRL	rw	0x0000	CRYPT 控制寄存器	0x0000
CRYPT_PTRA	rw	0x0004	CRYPT 指针 A	0x0000
CRYPT_PTRB	rw	0x0008	CRYPT 指针 B	0x0000
CRYPT_PTRO	rw	0x000C	CRYPT 指针 O	0x0000
CRYPT_CARRY	r	0x0010	CRYPT 借位/进位寄存器	0x0
CRYPT_PRIME0	rw	0x0020	模运算的素数寄存器 0	0xFFFFFFFF
CRYPT_PRIME1	rw	0x0024	模运算的素数寄存器 0	0xFFFFFFFF
CRYPT_PRIME2	rw	0x0028	模运算的素数寄存器 0	0xFFFFFFFF
CRYPT_PRIME3	rw	0x002C	模运算的素数寄存器 0	0x0000
CRYPT_PRIME4	rw	0x0030	模运算的素数寄存器 0	0x0000
CRYPT_PRIME5	rw	0x0034	模运算的素数寄存器 0	0x0000
CRYPT_PRIME6	rw	0x0038	模运算的素数寄存器 0	0x00000001
CRYPT_PRIME7	rw	0x003C	模运算的素数寄存器 0	0xFFFFFFFF

4.1.16 EPU 寄存器地址

表41. EPU 寄存器地址（EPU 基地址：0x40015000）

名称	类型	地址	描述	默认值
EGY_CFCTRL	rw	0x0008	CF 输出控制寄存器	0x0
EGY_HSCTRL	rw	0x000C	高速能量累加控制寄存器	0x0606
EGY_LSCTRL	rw	0x0010	低速能量累加控制寄存器	0x6666
EGY_CLR	rw	0x0014	清除寄存器	0x0
EGY_CRPTH	rw	0x0018	防潜门限值寄存器	0x0
EGY_PWRTH0	rw	0x001C	能量累加器门限值 0	0x0
EGY_PWRTH1	rw	0x0020	能量累加器门限值 1	0x0
EPU_INTEN	rw	0x0024	中断使能寄存器	0x0

EPU_INTSTS	rc_w1	0x0028	中断状态寄存器	0x0
EPU_STS	r	0x002C	EPU 状态寄存器	0x0
EPU_CONST0	rw	0x0034	能量累加器 0 常数寄存器（32 位无符号数）	0x0
EPU_OUTL0	r	0x0038	能量累加器 0 累加值低位	0x0
EPU_OUTH0	r	0x003C	能量累加器 0 累加值高位	0x0
EPU_CFCNT0	r	0x0040	能量累加器 0 脉冲计数器	0x0
EPU_CONST1	rw	0x0048	能量累加器 1 常数寄存器（32 位无符号数）	0x0
EPU_OUTL1	r	0x004C	能量累加器 1 累加值低位	0x0
EPU_OUTH1	r	0x0050	能量累加器 1 累加值高位	0x0
EPU_CFCNT1	r	0x0054	能量累加器 1 脉冲计数器	0x0
EPU_CONSTx (x=2-7)	rw	0x0070+ (x-2)*12	能量累加器 x 的常数寄存器（32 位无符号数）	0x0
EPU_OUTx (x=2-7)	r	0x0074+ (x-2)*12	能量累加器 x 的累加值	0x0
EPU_CFCNTx (x=2-7)	r	0x0078+ (x-2)*12	能量累加器 x 的脉冲计数器	0x0

4.1.17 RNG 寄存器地址

表42. RNG 寄存器地址（RNG 基地址：0x40016800）

名称	类型	地址	描述	默认值
RNG_IMR	rw	0x0100	中断掩码寄存器	0x0000
RNG_ISR	rw	0x0104	中断状态寄存器	0x0000
RNG_ICR	rw	0x0108	中断清除寄存器	0x0000
RNG_CFG	rw	0x010C	配置寄存器	0x0000
RNG_DATA0	rw	0x0114	RNG 数据寄存器 0	0x0000
RNG_DATA1	rw	0x0118	RNG 数据寄存器 1	0x0000
RNG_DATA2	rw	0x011C	RNG 数据寄存器 2	0x0000
RNG_DATA3	rw	0x0120	RNG 数据寄存器 3	0x0000
RNG_DATA4	rw	0x0124	RNG 数据寄存器 4	0x0000
RNG_DATA5	rw	0x0128	RNG 数据寄存器 5	0x0000
RNG_CMD	rw	0x012C	RNG 源启用寄存器	0x0000
RNG_CNT	rw	0x0130	采样计数器寄存器	0x0000
RNG_BYPASS	rw	0x0138	RNG 测试旁路寄存器	0x0000

4.1.18 Info 参数寄存器

Info 信息存储在以 0x00040800 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 0x00040800~0x0004085C 数据，通过烧写工具（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43. Info 信息寄存器

地址	符号	信息	描述
0x00040800	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。
0x00040804		Checksum1	INV (SUM (0x40800, 0x40800))
0x00040808	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。
0x0004080C		Checksum2	INV (SUM (0x40808, 0x40808))
0x00040810	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 20827。
0x00040814	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 21496。
0x00040818	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 22020。
0x0004081C	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 24517。
0x00040820	K5	晶体 K5 系数 1	加载到 RTC_ACK5 寄存器，K5 计算公式如下： $K5=B5/1000000*65536$ ，B5 为晶振曲线二次函数的分段系数。K5 值比如 25257。
0x00040824		Checksum1	INV (SUM (0x40810, 0x40820))
0x00040828	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 20827。
0x0004082C	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 21496。
0x00040830	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 22020。
0x00040834	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 24517。
0x00040838	K5	晶体 K5 系数 2	加载到 RTC_ACK5 寄存器，K5 计算公式如

			下: $K5=B5/1000000*65536$, B5 为晶振曲线二次函数的分段系数。K5 值比如 25257。
0x0004083C		Checksum2	INV (SUM (0x40828, 0x40838))
0x00040840	ACTI	晶体顶点温度 1	加载到 RTC_ACTI 寄存器, 比如: 0x1800。
0x00040844		Checksum1	INV (SUM (0x40840, 0x40840))
0x00040848	ACTI	晶体顶点温度 2	加载到 RTC_ACTI 寄存器, 比如: 0x1800。
0x0004084C		Checksum2	INV (SUM (0x40848, 0x40848))
0x00040850	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。
0x00040854	—	Checksum1	INV (SUM (0x40850, 0x40850))
0x00040858	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。
0x0004085C		Checksum2	INV (SUM (0x40858, 0x40858))
...			保留
0x00040CE0		BAT_R offset 1	(3.6-BAT 测量值)*1000, 电阻分压, BAT 包括 VDD 通道和 BATRTC 通道。
0x00040CE4		BAT_C offset 1	(3.6-BAT 测量值)*1000, 电容分压
0x00040CE8		Checksum1	INV (SUM (0x40CE0, 0x40CE4))
0x00040CEC			保留
0x00040CF0		BAT_R offset 2	(3.6-BAT 测量值)*1000, 电阻分压
0x00040CF4		BAT_C offset 2	(3.6-BAT 测量值)*1000, 电容分压
0x00040CF8		Checksum2	INV (SUM (0x40CF0, 0x40CF4))
0x00040CFC			保留
0x00040D00	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。
0x00040D04	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。
0x00040D08	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。
0x00040D0C	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, 其中 $a=PCLK/6553600$ 。
0x00040D10		Checksum1	INV (SUM (0x40D00, 0x40D0C))
0x00040D14	P1/P0	RTC_ACP1/0 设置 2	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。
0x00040D18	P2'	RTC_ACP2 设置 2	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。
0x00040D1C	P5	RTC_ACP5 设置 2	高 16 位加载到 RTC_ACP5 寄存器, 比如: 6444; 低 16 位舍弃。
0x00040D20	P7/P6'	RTC_ACP7/6 设置 2	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, 其中 $a=PCLK/6553600$ 。
0x00040D24		Checksum2	INV (SUM (0x40D14, 0x40D20))
0x00040D28		AVCC gain 1	Pre-trim result/3.3*10000
0x00040D2C		DVCC gain 1	Pre-trim result/1.5*10000
0x00040D30		BGP gain 1	Pre-trim result/1.2*10000
0x00040D34		RCL gain 1	Pre-trim result/32768*10000
0x00040D38		RCH gain 1	Pre-trim result/6553600*10000
0x00040D3C		Check sum 1	INV (SUM (0x40D28, 0x40D38))
0x00040D40		AVCC gain 2	Pre-trim result/3.3*10000
0x00040D44		DVCC gain 2	Pre-trim result/1.5*10000
0x00040D48		BGP gain 2	Pre-trim result/1.2*10000
0x00040D4C		RCL gain 2	Pre-trim result/32768*10000
0x00040D50		RCH gain 2	Pre-trim result/6553600*10000
0x00040D54		Check sum 2	INV (SUM (0x40D40, 0x40D50))
0x00040D58		ID word 0, Backup 1	
0x00040D5C		ID word 1, Backup 1	
0x00040D60		ID Check sum 1	INV (SUM (0x40D58, 0x40D5C))
0x00040D64		ID word 0, Backup 2	
0x00040D68		ID word 1, Backup 2	
0x00040D6C		ID Check sum 2	INV (SUM (0x40D64, 0x40D68))
0x00040D70	Tr	实际温度 1 (来自 tmp275)	根据公式: $P2=P2'+(Tr-Tm)*256$, 将 P2 加载到 RTC_ACP2 寄存器。
0x00040D74	Tm	测量温度 1 (来自 ADC)	
0x00040D78		Temp Check sum 1	INV (SUM (0x40D70, 0x40D74))
0x00040D7C	Tr	实际温度 2 (来自 tmp275)	根据公式: $P2=P2'+(Tr-Tm)*256$, 将 P2 加载到 RTC_ACP2 寄存器。
0x00040D80	Tm	测量温度 2 (来	

		自 ADC)	
0x00040D84		Temp Check sum 2	INV (SUM (0x40D7C, 0x40D80))
0x00040D88			保留
0x00040D8C			保留
0x00040D90		Measure LCD LDO gain 1	Pre-trim result/3.3*10000
0x00040D94		VLCD setting 1	
0x00040D98		LCD LDO Check sum 1	INV (SUM (0x40D90, 0x40D94))
0x00040D9C		Measure LCD LDO gain 2	Pre-trim result/3.3*10000
0x00040DA0		VLCD setting 2	
0x00040DA4		LCD LDO Check sum 2	INV (SUM (0x40D9C, 0x40DA0))
0x00040DC0		Analog trim data1	CP 测试时自动加载到 ANA_REGx (x=B~E) 的值
0x00040DC4		0xFFFFFFFF	
0x00040DC8		0xFFFFFFFF	
0x00040DCC		Checksum1	INV (SUM (0x40DC0, 0x40DC8)), 其中 0x00040DC8 地址的数据不可读, 使用 0xFFFFFFFF 代替。
0x00040DD0		Analog trim data2	CP 测试时自动加载到 ANA_REGx (x=B~E) 的值
0x00040DD4		0xFFFFFFFF	
0x00040DD8		0xFFFFFFFF	
0x00040DDC		Checksum2	INV (SUM (0x40DD0, 0x40DD8)), 其中 0x00040DD8 地址的数据不可读, 使用 0xFFFFFFFF 代替。
...			
0x00040400	a1	ADC_CHx 通道在 3.3V 系统下不分压标定 1	Vdc=a1/100000000*X+b1/100000000 (用补码方式存放负值)
0x00040404	b1	ADC_CHx 通道在 3.3V 系统下不分压标定 1	
0x00040408	a2	ADC_CHx 通道在 3.3V 系统下电阻分压标定 1	Vdc=a2/100000000*X+b2/100000000 (用补码方式存放负值)
0x0004040C	b2	ADC_CHx 通道在 3.3V 系统下电阻分压标定 1	

0x00040410	a3	ADC_CHx 通道在 3.3V 系统下电容分压标定 1	Vdc=a3/100000000*X+b3/100000000 (用补码方式存放负值)
0x00040414	b3	ADC_CHx 通道在 3.3V 系统下电容分压标定 1	
0x00040418	a4	VDD 通道在 3.3V 系统下电阻分压标定 1	Vdc=a4/100000000*X+b4/100000000 (用补码方式存放负值)
0x0004041C	b4	VDD 通道在 3.3V 系统下电阻分压标定 1	
0x00040420	a5	VDD 通道在 3.3V 系统下电容分压标定 1	Vdc=a5/100000000*X+b5/100000000 (用补码方式存放负值)
0x00040424	b5	VDD 通道在 3.3V 系统下电容分压标定 1	
0x00040428	a6	BATRTC 通道在 3.3V 系统下电阻分压标定 1	Vdc=a6/100000000*X+b6/100000000 (用补码方式存放负值)
0x0004042C	b6	BATRTC 通道在 3.3V 系统下电阻分压标定 1	
0x00040430	a7	BATRTC 通道在 3.3V 系统下电容分压标定 1	Vdc=a7/100000000*X+b7/100000000 (用补码方式存放负值)
0x00040434	b7	BATRTC 通道在 3.3V 系统下电容分压标定 1	
0x00040438		Checksum1	INV (SUM (0x00040400, 0x00040434))
0x0004043C			保留
0x00040440	a1	ADC_CHx 通道在 3.3V 系统下不分压标定 2	Vdc=a1/100000000*X+b1/100000000 (用补码方式存放负值)
0x00040444	b1	ADC_CHx 通道在 3.3V 系统下不分压标定 2	
0x00040448	a2	ADC_CHx 通道在 3.3V 系统下电阻分压标定 2	Vdc=a2/100000000*X+b2/100000000 (用补码方式存放负值)
0x0004044C	b2	ADC_CHx 通道	

		在 3.3V 系统下 电阻分压标定 2	
0x00040450	a3	ADC_CHx 通道 在 3.3V 系统下 电容分压标定 2	Vdc=a3/100000000*X+b3/100000000 (用补 码方式存放负值)
0x00040454	b3	ADC_CHx 通道 在 3.3V 系统下 电容分压标定 2	
0x00040458	a4	VDD 通道在 3.3V 系统下电阻 分压标定 2	Vdc=a4/100000000*X+b4/100000000 (用补 码方式存放负值)
0x0004045C	b4	VDD 通道在 3.3V 系统下电阻 分压标定 2	
0x00040460	a5	VDD 通道在 3.3V 系统下电容 分压标定 2	Vdc=a5/100000000*X+b5/100000000 (用补 码方式存放负值)
0x00040464	b5	VDD 通道在 3.3V 系统下电容 分压标定 2	
0x00040468	a6	BATRTC 通道在 3.3V 系统下电阻 分压标定 2	Vdc=a6/100000000*X+b6/100000000 (用补 码方式存放负值)
0x0004046C	b6	BATRTC 通道在 3.3V 系统下电阻 分压标定 2	
0x00040470	a7	BATRTC 通道在 3.3V 系统下电容 分压标定 2	Vdc=a7/100000000*X+b7/100000000 (用补 码方式存放负值)
0x00040474	b7	BATRTC 通道在 3.3V 系统下电容 分压标定 2	
0x00040478		Checksum2	INV (SUM (0x00040440, 0x00040474))
0x00040480	A1	ADC_CHx 通道 在 5V 系统下不 分压标定 1	Vdc=A1/100000000*X+B1/100000000 (用补 码方式存放负值)
0x00040484	B1	ADC_CHx 通道 在 5V 系统下不 分压标定 1	
0x00040488	A2	ADC_CHx 通道 在 5V 系统下电 阻分压标定 1	Vdc=A2/100000000*X+B2/100000000 (用补 码方式存放负值)

0x0004048C	B2	ADC_CHx 通道在 5V 系统下电阻分压标定 1	
0x00040490	A3	ADC_CHx 通道在 5V 系统下电容分压标定 1	Vdc=A3/100000000*X+B3/100000000 (用补码方式存放负值)
0x00040494	B3	ADC_CHx 通道在 5V 系统下电容分压标定 1	
0x00040498	A4	VDD 通道在 5V 系统下电阻分压标定 1	
0x0004049C	B4	VDD 通道在 5V 系统下电阻分压标定 1	Vdc=A4/100000000*X+B4/100000000 (用补码方式存放负值)
0x000404A0	A5	VDD 通道在 5V 系统下电容分压标定 1	
0x000404A4	B5	VDD 通道在 5V 系统下电容分压标定 1	Vdc=A5/100000000*X+B5/100000000 (用补码方式存放负值)
0x000404A8	A6	BATRTC 通道在 5V 系统下电阻分压标定 1	
0x000404AC	B6	BATRTC 通道在 5V 系统下电阻分压标定 1	Vdc=A6/100000000*X+B6/100000000 (用补码方式存放负值)
0x000404B0	A7	BATRTC 通道在 5V 系统下电容分压标定 1	
0x000404B4	B7	BATRTC 通道在 5V 系统下电容分压标定 1	
0x000404B8		Checksum1	INV (SUM (0x00040480, 0x000404B4))
0x000404BC			保留
0x000404C0	A1	ADC_CHx 通道在 5V 系统下不分压标定 2	Vdc=A1/100000000*X+B1/100000000 (用补码方式存放负值)
0x000404C4	B1	ADC_CHx 通道在 5V 系统下不分压标定 2	
0x000404C8	A2	ADC_CHx 通道	Vdc=A2/100000000*X+B2/100000000 (用补

		在 5V 系统下电阻分压标定 2	码方式存放负值)
0x000404CC	B2	ADC_CHx 通道在 5V 系统下电阻分压标定 2	
0x000404D0	A3	ADC_CHx 通道在 5V 系统下电容分压标定 2	Vdc=A3/100000000*X+B3/100000000 (用补码方式存放负值)
0x000404D4	B3	ADC_CHx 通道在 5V 系统下电容分压标定 2	
0x000404D8	A4	VDD 通道在 5V 系统下电阻分压标定 2	Vdc=A4/100000000*X+B4/100000000 (用补码方式存放负值)
0x000404DC	B4	VDD 通道在 5V 系统下电阻分压标定 2	
0x000404E0	A5	VDD 通道在 5V 系统下电容分压标定 2	Vdc=A5/100000000*X+B5/100000000 (用补码方式存放负值)
0x000404E4	B5	VDD 通道在 5V 系统下电容分压标定 2	
0x000404E8	A6	BATRTC 通道在 5V 系统下电阻分压标定 2	Vdc=A6/100000000*X+B6/100000000 (用补码方式存放负值)
0x000404EC	B6	BATRTC 通道在 5V 系统下电阻分压标定 2	
0x000404F0	A7	BATRTC 通道在 5V 系统下电容分压标定 2	Vdc=A7/100000000*X+B7/100000000 (用补码方式存放负值)
0x000404F4	B7	BATRTC 通道在 5V 系统下电容分压标定 2	
0x000404F8		Checksum2	INV (SUM (0x000404C0, 0x000404F4))

5 电源系统

V85XX 电源控制器有以下特点：

- 输入电压范围：2.2V~5.5V
- 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

5.1 寄存器地址

表44. ANA 寄存器地址（ANA 基地址：0x40014200）

寄存器名	类型	地址	描述	默认值
ANA_REG5	rw	0x0014	模拟寄存器 5	0x00
ANA_REG6	rw	0x0018	模拟寄存器 6	0x00
ANA_REG7	rw	0x001C	模拟寄存器 7	0x00
ANA_REG8	rw	0x0020	模拟寄存器 8	0x00
ANA_REG9	rw	0x0024	模拟寄存器 9	0x00
ANA_REGA	rw	0x0028	模拟寄存器 10	0x00
ANA_REGF	rw	0x003C	模拟寄存器 15	0x00
ANA_CTRL	rw	0x0050	模拟控制寄存器	0x00000000
ANA_CMPOUT	r	0x0054	比较器结果寄存器	0x0030
ANA_INTSTS	rc_w1	0x0060	模拟中断状态寄存器	0x0000
ANA_INTEN	rw	0x0064	模拟中断使能寄存器	0x0000

5.2 寄存器定义

5.2.1 ANA_REG5 寄存器

表45. ANA_REG5 寄存器

位	名称	功能	描述
6	PD_AVCCDET	关闭 AVCC 低电压监测模块。	0: 开启。 1: 关闭。

5.2.2 ANA_REG6 寄存器

表46. ANA_REG6 寄存器

位	名称	功能	描述
6	VDDDISC	给 BAT 电池放电。 放电电阻是 1.7k，放电电流是 $V_{VDD}/1.7k$ 。	0: 禁用从 VDD 到 GND 的 1.7k 欧姆电阻。 1: 使能从 VDD 到 GND 的 1.7k 欧姆电阻。
7	BATRCDISC	给 BATRTC 电池放电。 放电电阻是 1.7k，放电电流是 $V_{BATRTC}/1.7k$ 。	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。

5.2.3 ANA_REG7 寄存器

表47. ANA_REG7 寄存器

位	名称	功能	描述
7: 0	Rsvd	保留	默认值为 0，必须配置为 0。

5.2.4 ANA_REG8 寄存器

表48. ANA_REG8 寄存器

位	名称	功能	描述
3: 0	Rsvd	保留	0
6: 4	VDDPVDSEL[2: 0]	VDD 电压报警阈值设置寄存器，当 VDD 电压低于该设置值时，将产生 VDDALARM 中断，并置相应中断标志位为 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
7	PD_AVCCCLDO	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。 AVCCCLDO 只能在供电电压低于 3.6V 时关闭。

5.2.5 ANA_REG9 寄存器

表49. ANA_REG9 寄存器

位	名称	功能	描述
7	PD_VDDDET	关闭 VDD 电压监测模块，该模块由 VDD 供电。	0: 开启。 1: 关闭。

5.2.6 ANA_REGA 寄存器

表50. ANA_REGA 寄存器

位	名称	功能	描述
6: 0	Rsvd	保留	默认值是 0x0，必须配置为 0x0A。
7	PD_VDCINDET	关闭 VDCIN 监测模块。	0: 开启 1: 关闭

5.2.7 ANA_REGF 寄存器

表51. ANA_REGF 寄存器

位	名称	功能	描述
2	AVCCO_EN	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。

5.2.8 ANA_CMPOUT 寄存器

表52. ANA_CMPOUT 寄存器

位	名称	类型	描述	默认值
10	AVCCCLV	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 VDDPVDSSEL 设置的电压 1: VDD 电压低于 VDDPVDSSEL 设置的电压	0x0
6: 0	Rsvd	-	保留	0

5.2.9 ANA_INSTS 寄存器

表53. ANA_INTSTS 寄存器

位	名称	类型	描述	默认值
11	INTSTS11	rc_w1	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)，使得系统可以从睡眠模式唤醒 (VDCINDROP 为 0)。	0x0

			Read0: 无进入睡眠事件发生 (VDCINDROP=0) Read1: 发生进入睡眠事件 (VDCINDROP=0) Write0: 无影响 Write1: 清除该位	
10	INTSTS10	rc_w1	AVCCLV 的中断标志, 该中断会在 AVCCLV 上升和下降时产生。 Read0: 无 AVCCLV 中断 Read1: 发生 AVCCLV 中断 Write0: 无效 Write1: 清零	0x0
8	INTSTS8	rc_w1	VDCINDROP 的中断标志, 该中断会在 VDCINDROP 上升和下降时产生。 Read0: 无 VDCIN 中断 Read1: 发生 VDCIN 中断 Write0: 无效 Write1: 清零	0x0
7	INTSTS7	rc_w1	VDDALARM 的中断标志, 该中断会在 VDDALARM 上升和下降时产生。 Read0: 无 VDDALARM 中断 Read1: 发生 VDDALARM 中断 Write0: 无效 Write1: 清零	0x0
6: 0	Rsvd	-	保留	0

5.2.10 ANA_INTEN 寄存器

表54. ANA_INTEN 寄存器

位	名称	类型	描述	默认值
11	INTEN11	rw	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	rw	上升和下降时, AVCCLV 中断或唤醒使能控制。 0: 禁止 AVCCLV 中断或唤醒 1: 使能 AVCCLV 中断或唤醒	0x0
9	Rsvd	-	保留	0
8	INTEN8	rw	上升和下降时, 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	rw	上升和下降时，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
6: 0	Rsvd	-	保留	0

5.3 AVCC

V85XX 中的模拟电路由 AVCC 供电，AVCC 状态由控制位‘PD_AVCCCLDO’（ANA_REG8 位 7）控制。AVCCCLDO 只有在供电电压低于 3.6V 时可关闭。

AVCC 的驱动能力是 30mA。

AVCC_OUT 管脚是否输出 AVCC 电平可控制，通过控制位‘AVCCO_EN’（ANA_REGF）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 AVCCCLV 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6. AVCCCLDO 输出电压与 VDD 输入电压之间的关系

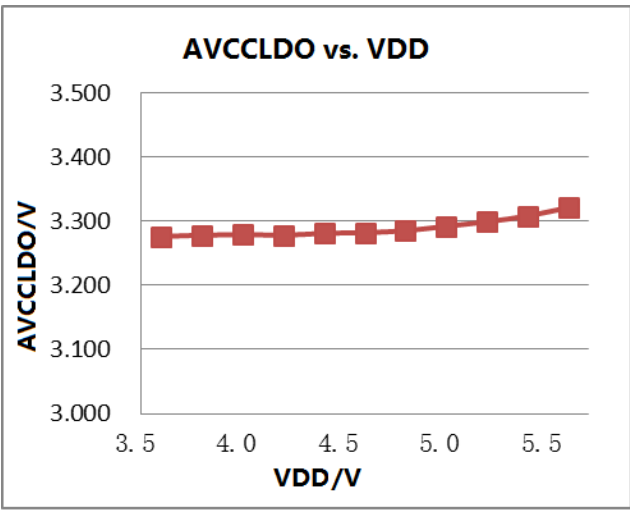
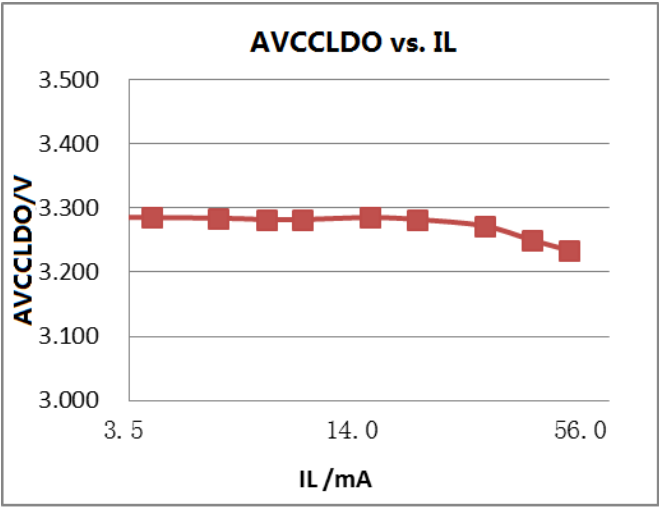


图7. AVCCCLDO 输出电压与负载电流之间的关系



5.4 DVCC

V85XX 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8. DVCCCLDO 输出电压与 VDD 输入电压之间的关系

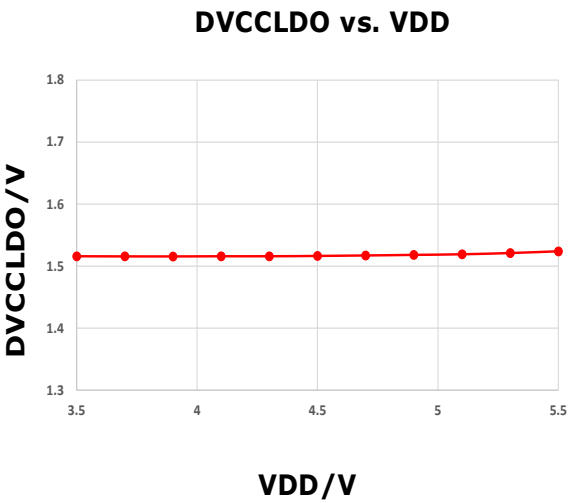
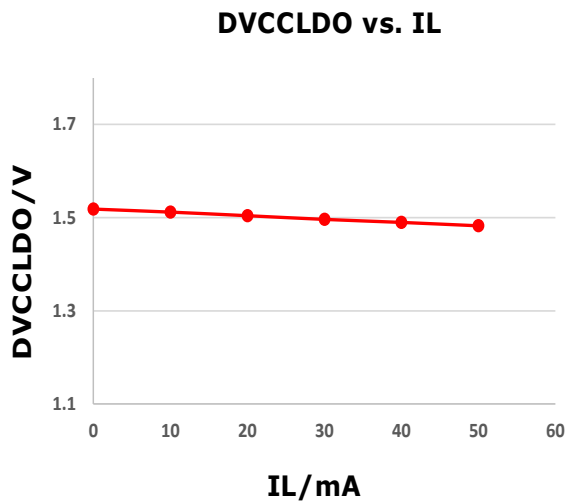


图9. DVCCCLDO 输出电压与负载电流之间的关系

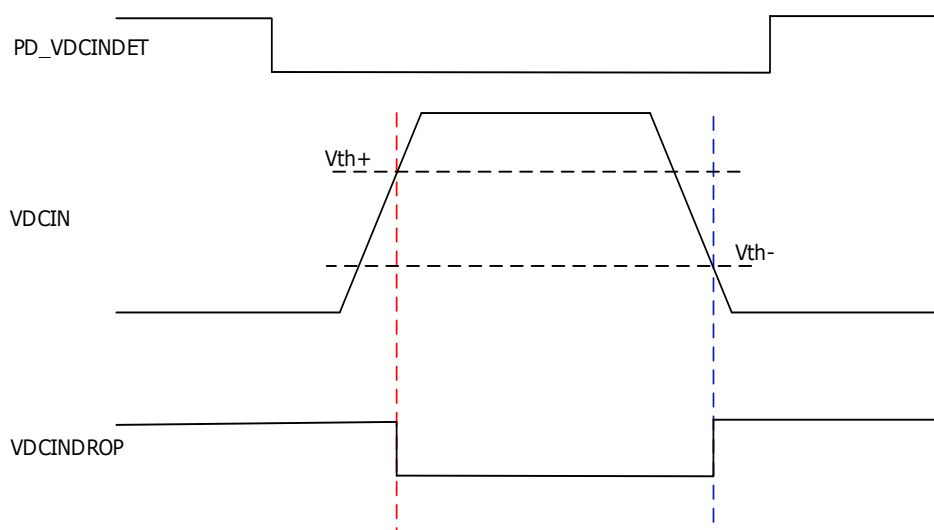


5.5 电源监测

5.5.1 VDCIN 电源检测

在 V85XX 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 电压降低，低于 V_{th-} 时，标志位 VDCINDROP（ANACMP_OUT 位 8）将会被置 1 并产生掉电中断。当 VDCIN 电压上升，高于 V_{th+} 时，标志位 VDCINDROP（ANACMP_OUT 位 8）将会被清 0 并产生来电中断。VDCIN 电压监测功能默认开启，不用的时候可通过对 ANA_REGA 寄存器的 BIT7（PD_VDCINDET）配置 1 关闭该功能。

图10. 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 2

电源开关 2 用于选择 AVCC 的输入源。当 PD_AVCCCLDO（ANA_REG8 位 7）为 0，AVCC 输入源为 AVCCCLDO 的输出，电压为 3.3V；当 PD_AVCCCLDO（ANA_REG8 位 7）为 1，AVCC 引脚通过一个开关连接到 VDD，开关内阻请参考表 5。

AVCCCLDO 只有在供电电压低于 3.6V 时可关闭。

6 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 V85XX 的睡眠模式。PMU 控制器内支持最多 16 个外部 IO 唤醒源，可以将芯片从睡眠模式唤醒。

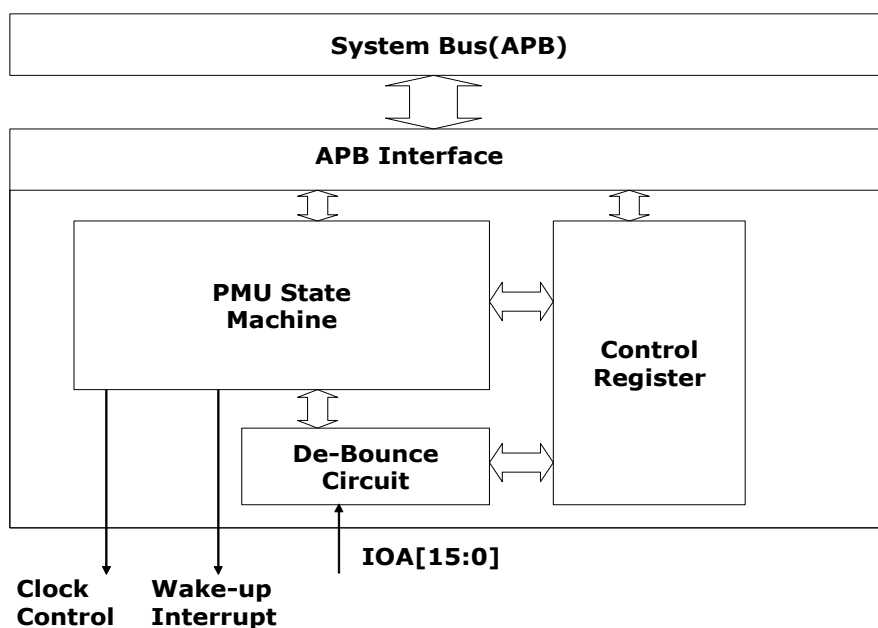
睡眠模式将掉电部分外设，只有 CPU、SRAM、GPIO、RTC、模拟控制器、PMU、EPU 和 UART32K 等模块在此模式下会工作。

6.2 特点

- 睡眠模式进出可控，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。
- 16 个具有唤醒和中断功能的 GPIO。
- 进入睡眠时会自动关闭时钟，睡眠唤醒后会自动打开睡眠前的系统时钟；
- 晶体缺失监测和中断。
- 睡眠唤醒后自动打开看门狗。

6.3 功能框图

图11. PMU 功能框图



6.4 寄存器地址

表55. 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_CONTROL	rw	0x0008	PMU 控制寄存器	0x0000
PMU_STS	rc_w1	0x000C	PMU 状态寄存器	0x0000074
VERSIONID	r	0x003C	V85XX 版本号	--

表56. PMU_RAM 寄存器地址（PMU_RAM 基地址：0x40014400）

名称	类型	地址	描述	默认值
PMU_RAM0	rw	0x0000	PMU 32 位 RAM0	--
PMU_RAM1	rw	0x0004	PMU 32 位 RAM1	--
PMU_RAM2	rw	0x0008	PMU 32 位 RAM2	--
.....				--
PMU_RAM63	rw	0x00FC	PMU 32 位 RAM63	--

表57. 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	rw	0x0050	模拟控制寄存器	0x0000000

6.5 寄存器定义

6.5.1 PMU_CONTROL 寄存器

表58. PMU_CONTROL 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 8	PWUPCYC	rw	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	Rsvd	-	保留	0
5	PLLL_SEL	rw	低速 PLL 输入时钟选择。 0: 32K XTAL。 1: 32K RC。	0x0
4	PLLH_SEL	rw	高速 PLL 输入时钟选择。 0: 6.5MHz RC。 1: 6.5536MHz XTAL。	0x0
3	INT_6M_EN	rw	6.5536M XTAL 失效中断使能控制位。该位用于控制	0x0

			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。	
2	INT_32K_EN	rw	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	rw	RTC 时钟选择。 0: 32K XTAL。 1: 32K RC。	0x0
0	INT_IOA_EN	rw	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

6.5.2 PMU_STS 寄存器

表59. PMU_STS 寄存器

位	名称	类型	描述	默认值
31: 25	Rsvd	-	保留	0
24	MODE	r	该位 MODE 引脚当前状态。 0: Debug 模式。 1: Normal 模式。	--
23: 7	Rsvd	-	保留	0
6	DPORST	rc_w1	该位表示上一次复位是否由内部数字上电复位信号引起，只有第一次上电才会置 1。写 1 清零该位。	0x1
5	PORST	rc_w1	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	rc_w1	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 的位 7 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	rc_w1	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	0x0

			1。 当该位为 1 时，表示 6.5536M 晶振被去除或损坏。 向此位写入 1 可将该标志清零。	
0	INT_32K	rc_w1	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。 向此位写入 1 可将该标志清零。	0x0

6.5.3 PMU_WDTSTS 寄存器

表60. PMU_WDTSTS 寄存器

位	名称	类型	描述	默认值
31: 1	Rsvd	-	保留	0
0	WDTSTS	rc_w1	该位指示上一次复位是否由看门狗复位信号引起。写 1 清零该位。	0x0

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61. 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	唤醒前模式	
					睡眠	IDLE
Cortex-M0	√	√	√	√	-	-
系统 SRAM	-	-	-	-	-	-
掉电保持 SRAM	-	-	-	-	-	-
PMU (IOA)	√	√	√	-	-	-
WDT	√	√	√	-	-	-
RTC	√*	√*	√*	-	-	-
UART32K0	√	√	√	-	-	-
UART32K1	√	√	√	-	-	-
模拟控制器	√	√	√	-	-	-
LCD	√	√	√	√	-	-
GPIO (IOB~IOF)	√	√	√	√	-	-
MISC2	√	√	√	√	-	-
MISC	√	√	√	√	√	-
I2C	√	√	√	√	√	-
SPI	√	√	√	√	√	√
UART	√	√	√	√	√	√

ISO7816	√	√	√	√	√	√	-
TIMER	√	√	√	√	√	√	-
PWM	√	√	√	√	√	√	-
DMA	√	√	√	√	√	√	-
RNG	√	√	√	√	√	√	-
EPU	√	√	√	√	√	-	-
SRAM 控制器	√	√	√	√	√	√	-
FLASH 控制器	√	√	√	√	√	√	-

注1: (√*): RTC 不带写保护的寄存器会被上电复位、外部复位、看门狗复位这三种复位源复位。RTC 带写保护寄存器不会被外部复位、看门狗复位这两种复位源复位，除时间外的带写保护的寄存器会被上电复位这种复位源复位。不会被上电复位的 RTC 寄存器是: RTC_SEC、RTC_MIN、RTC_HOUR、RTC_DAY、RTC_WEEK、RTC_MON、RTC_YEAR、RTC_CAL。

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62. 在不同模式下的各个时钟的工作状态

时钟源	工作模式		
	睡眠	IDLE	工作
6.5536M RC	OFF	由 RCHPD 控制	
6.5536M XTAL	OFF	由 XOHPDN 控制	
PLLH	OFF	由 PLLHPDN 控制	
PLLL	OFF	由 PLLLPDN 控制	
32K RC	ON	ON	
32K XTAL	ON	ON	

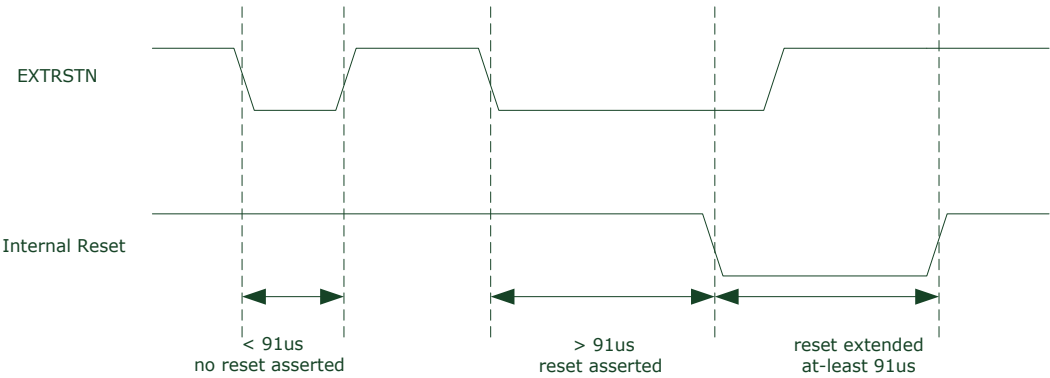
时钟源	不同模式唤醒	
	睡眠	IDLE
6.5536M RC	由 RCHPD 控制	
6.5536M XTAL	由 XOHPDN 控制	
PLLH	由 PLLHPDN 控制	
PLLL	由 PLLLPDN 控制	
32K RC	ON	
32K XTAL	ON	

6.6.1 片外复位

片外复位引脚（EXTRSTN）能复位 V85XX 的大部分模块，具体可参考表 61。为避免外部噪声耦合进此脚，V85XX 集成了一个去毛刺电路。下图显示去毛刺这个电路的工作原理。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 $91 \times 4 = 364\mu s$ 。

图12. 片外复位的去毛刺时序



6.6.2 看门狗复位

看门狗复位与外部复位等级相同，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 V85XX 芯片的大多数模块，详情请参考表 61。

6.6.5 从睡眠/IDLE 下唤醒

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

表63. 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
						睡眠
NMI	00000008h	-14	NMI			
HardFault	0000000Ch	-13	HardFault			
SVCall	0000002Ch	-5	SVCall			
PendSV	00000038h	-2	PendSV			
SysTick	0000003C	-1	SysTick			

	h					
PMU	00000040h	0	IOA0~15	PMU_CONTROL.0andPMU_IOAWKUEN.0~15	PMU_IOAINTSTS.0~15	V
			32K 晶振失效	PMU_CONTROL.2	PMU_STS.0	V
			6M 晶振失效	PMU_CONTROL.3	PMU_STS.1	
RTC	00000044h	1	保留	RTC_INTEN.0	RTC_INTSTS.0	V
			非法时间格式	RTC_INTEN.1	RTC_INTSTS.1	V
			多秒中断	RTC_INTEN.2	RTC_INTSTS.2	V
			多分钟中断	RTC_INTEN.3	RTC_INTSTS.3	V
			多小时中断	RTC_INTEN.4	RTC_INTSTS.4	V
			午夜（00:00）中断	RTC_INTEN.5	RTC_INTSTS.5	V
			32K 计数器中断	RTC_INTEN.6	RTC_INTSTS.6	V
			-	-	-	-
U32K0~1	00000048h 0000004Ch	2\3	接收到数据	U32Kx_CTRL1.0	U32Kx_STS.0	V
			接收奇偶校验错误	U32Kx_CTRL1.1	U32Kx_STS.1	V
			接收缓存溢出	U32Kx_CTRL1.2	U32Kx_STS.2	V
I2C	00000050h	4	I2C 串行中断	I2C_CTRL2.0	I2C_CTRL.3	
SPI1~4	00000054h \000000AC h\000000A 8h\000000 A4h	5\27 \26\ 25	SPI 发送	SPIx_TXSTS.14	SPIx_TXSTS.15	
			SPI 接收	SPIx_RXSTS.14	SPIx_RXSTS.15	
UART0~5	00000058h ~0000006 Ch	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	
ISO78160~1	00000070h ~00000074	12\13	接收中断	ISO7816x_CFG.5	ISO7816x_INFO.5	
			发送中断	ISO7816x_CFG.6	ISO7816x_INFO.6	

	h		接收溢出	ISO7816x_CFG.7	ISO7816x_INFO.7	
Timer0~3	00000078h ~00000084h	14\15\16\17	定时器溢出	TMRx_CTRL.3	TMRx_INT.0	
PWM0~3	00000088h ~00000094h	18\19\20\21	PWM 定时器溢出	PWMx_CTL.1	PWMx_CTL.0	
			比较 0	PWMx_CCTL0.4	PWMx_CCTL0.0	
			比较 1	PWMx_CCTL1.4	PWMx_CCTL1.0	
			比较 2	PWMx_CCTL2.4	PWMx_CCTL2.0	
DMA	00000098h	22	通道 0 包结束	DMA_IE.0	DMA_STS.0	
			通道 1 包结束	DMA_IE.1	DMA_STS.1	
			通道 2 包结束	DMA_IE.2	DMA_STS.2	
			通道 3 包结束	DMA_IE.3	DMA_STS.3	
			通道 0 帧结束	DMA_IE.4	DMA_STS.4	
			通道 1 帧结束	DMA_IE.5	DMA_STS.5	
			通道 2 帧结束	DMA_IE.6	DMA_STS.6	
			通道 3 帧结束	DMA_IE.7	DMA_STS.7	
			通道 0 数据丢弃	DMA_IE.8	DMA_STS.8	
			通道 1 数据丢弃	DMA_IE.9	DMA_STS.9	
			通道 2 数据丢弃	DMA_IE.10	DMA_STS.10	
			通道 3 数据丢弃	DMA_IE.11	DMA_STS.11	
FLASH	0000009Ch	23	校验和错误	FLASH_CTRL.2	FLASH_INT.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	
			COMP1 的上升和下降沿	ANA_INTEN.2	ANA_INTSTS.2	V
			COMP2 的上	ANA_INTEN.3	ANA_INTSTS.3	V

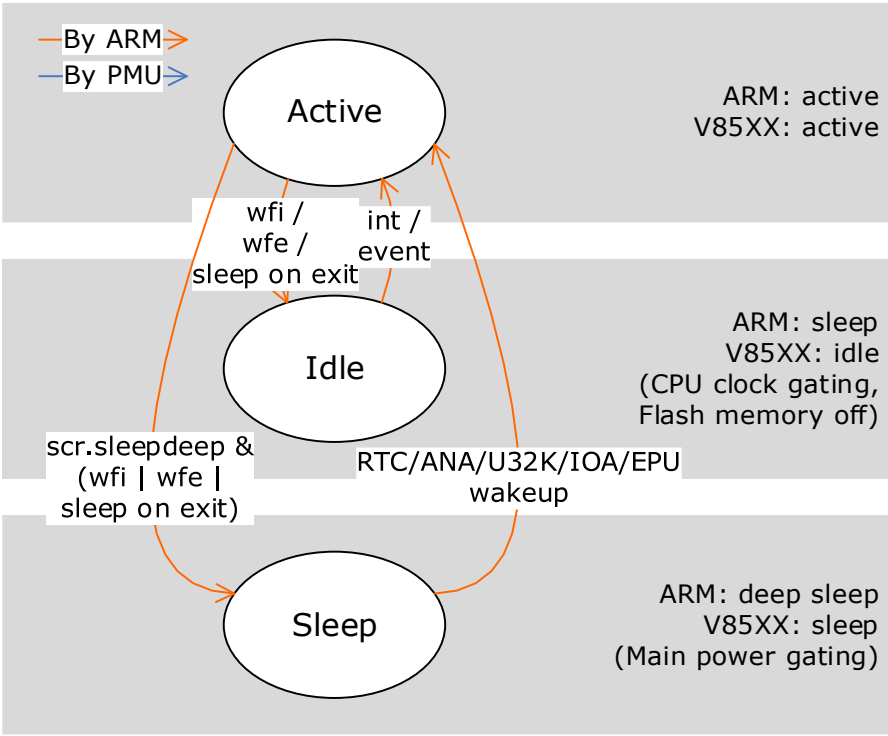
			升和下降沿			
			VDDALARM 的上升和下降 沿	ANA_INTEN.7	ANA_INTSTS.7	V
			VDCIN 的上升 和下降沿	ANA_INTEN.8	ANA_INTSTS.8	V
			AVCCLV 的上 升和下降沿	ANA_INTEN.10	ANA_INTSTS.10	V
			VDCINDROP =0 同时进入了 睡眠状态	ANA_INTEN.11	ANA_INTSTS.11	V
			ANA_REGx 错误	ANA_INTEN.12	ANA_INTSTS.12	V
			TADC 变化超 过阈值	ANA_INTEN.13	ANA_INTSTS.13	V
EPU	000000B0 h	28	正确接收 0xA5 帧	EPU_INTEN.12	EPU_INTSTS.12	
			接收字节超时	EPU_INTEN.11	EPU_INTSTS.11	
			接收数据 checksum 错 误	EPU_INTEN.10	EPU_INTSTS.10	
			接收数据奇偶 校验错误	EPU_INTEN.9	EPU_INTSTS.9	
			正确接收帧数 据	EPU_INTEN.8	EPU_INTSTS.8	
			能量累加器 7 溢出	EPU_INTEN.7	EPU_INTSTS.7	
			能量累加器 6 溢出	EPU_INTEN.6	EPU_INTSTS.6	
			能量累加器 5 溢出	EPU_INTEN.5	EPU_INTSTS.5	
			能量累加器 4 溢出	EPU_INTEN.4	EPU_INTSTS.4	
			能量累加器 3 溢出	EPU_INTEN.3	EPU_INTSTS.3	
			能量累加器 2 溢出	EPU_INTEN.2	EPU_INTSTS.2	
			能量累加器 1 溢出	EPU_INTEN.1	EPU_INTSTS.1	V
			能量累加器 0 溢出	EPU_INTEN.0	EPU_INTSTS.0	V

RNG	000000BC h	31	冯诺依曼错误	RNG_IMR.3	RNG_ISR.3	
			连续随机数生成错误	RNG_IMR.2	RNG_ISR.2	
			自相关错误	RNG_IMR.1	RNG_ISR.1	
			数据有效	RNG_IMR.0	RNG_ISR.0	

6.7 工作模式的转换

下图描述了 V85XX 各种工作模式的转换条件和状态。

图13. V85XX 工作模式转换



根据上图，IDLE 模式或睡眠模式由 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。

注1: 当 MODE=0 时，MCU 执行进入睡眠的指令后，会进入 IDLE 模式；
注2: 当 MCU 从睡眠唤醒后，MCU 会自动切换系统时钟到进入睡眠之前的设置。例如，睡眠之前系统时钟为 PLLL，睡眠唤醒后，硬件会自动切换系统时钟到 PLLL。IDLE 模式前后，硬件不会自动改变时钟状态。
注3: 在进入睡眠模式时不能选用 RTCCLK 为系统时钟。系统时钟源由 CLKSEL（MISC2_CTRL 位 2: 0）控制。
注4: 在睡眠唤醒时，PLLL 锁定时间约为 1ms，PLLH 锁定时间为 15μs。

表64. 每一种工作模式下的唤醒源

序号	唤醒源	唤醒前模式	
		睡眠	IDLE
Event	-	-	-
NMI	SRAM 奇偶校验错误	-	V
0	PMU (ExtInt)	V	V
1	RTC	V	V
2	UART32K0	V	V
3	UART32K1	V	V
4	I2C	-	V
5	SPI1	-	V
6	UART0	-	V
7	UART1	-	V
8	UART2	-	V
9	UART3	-	V
10	UART4	-	V
11	UART5	-	V
12	ISO78160	-	V
13	ISO78161	-	V
14	TIMER0	-	V
15	TIMER1	-	V
16	TIMER2	-	V
17	TIMER3	-	V
18	PWM0	-	V
19	PWM1	-	V
20	PWM2	-	V
21	PWM3	-	V
22	DMA	-	V
23	FLASH	-	V
24	ANA	V	V
25	SPI4	-	V
26	SPI3	-	V
27	SPI2	-	V
28	EPU	V	V
31	RNG	-	V

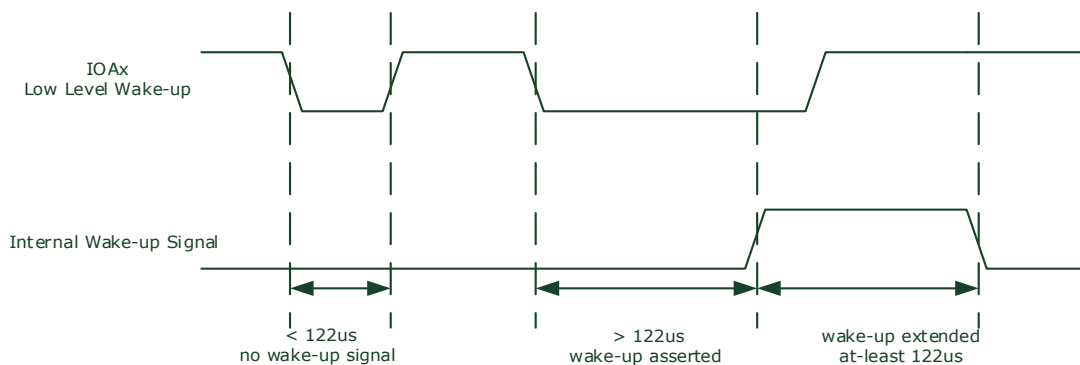
6.8 应用笔记

6.8.1 外部 IO 唤醒

所有 16 个 IOA 都可以将系统从睡眠模式唤醒。V85XX 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 μ s。当 RTCCLK 4 分频时（RTC_PSCA = 1），去毛刺时间会增加到 122*4=488 μ s。去毛刺电路仅对睡眠模式的 IOA 边沿唤醒信号有效；对正常中断和睡眠 IOA 电平唤醒信号无效。

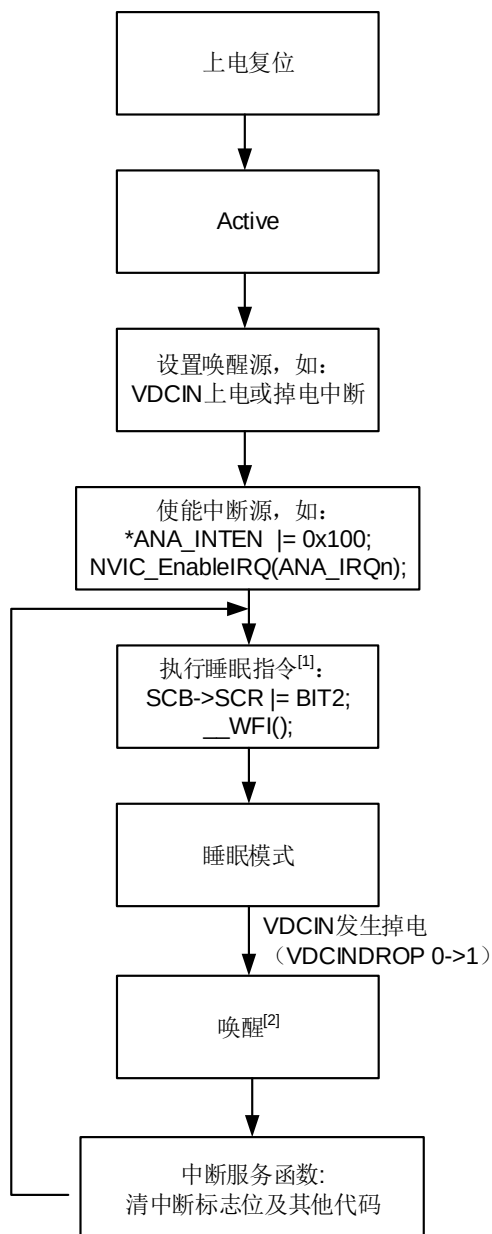
图14. 外部唤醒示意图



6.8.2 进入睡眠的流程

下图描述了一个进入睡眠的步骤示例。

图15. 进入睡眠的流程



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

[2]: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65。

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

睡眠及睡眠唤醒后的时钟状态, 请参考表67。

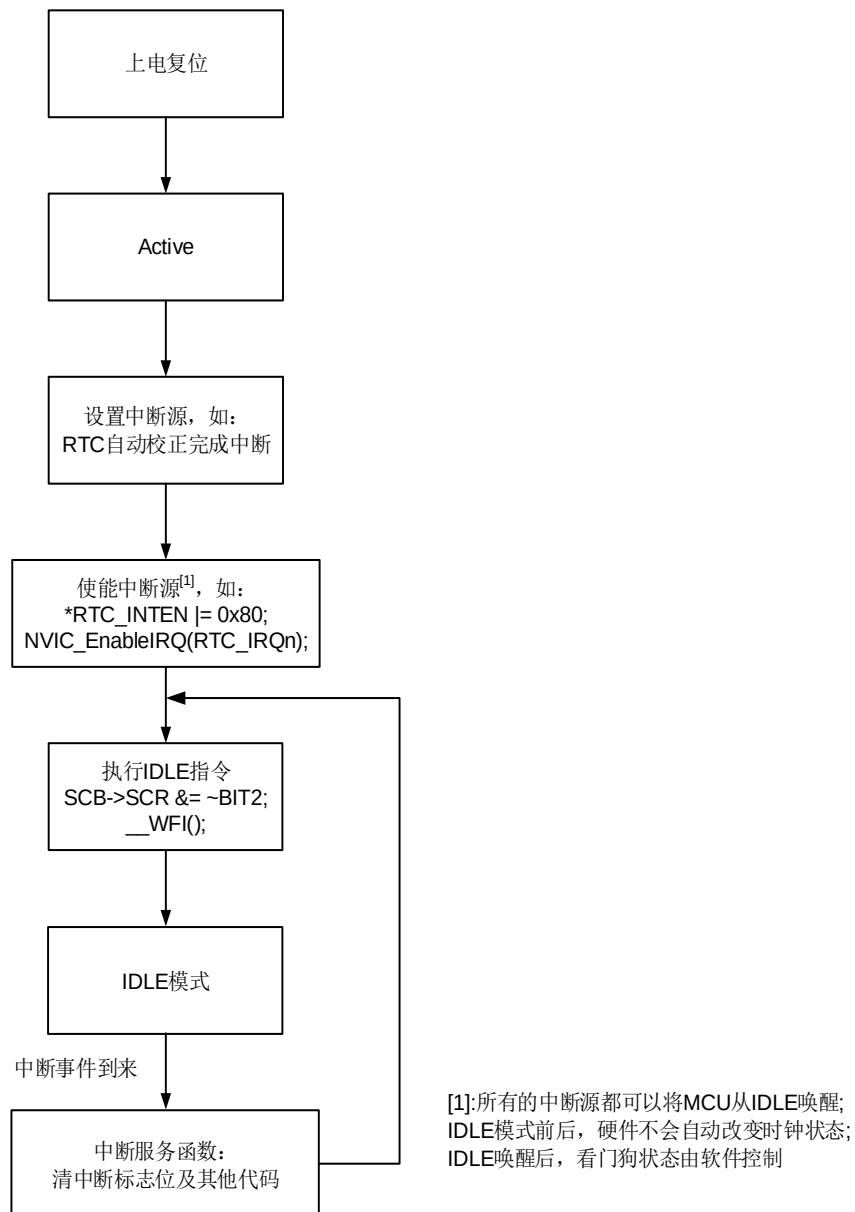
睡眠唤醒会自动打开看门狗。

注意: 进入睡眠是不能选用RTCCLK为系统时钟。系统时钟源由CLKSEL(MISC2_CTRL bit2:0)控制。

6.8.3 进入 IDLE 的流程

下图描述了进入 IDLE 模式的流程。

图16. 进入 IDLE 的流程



7 时钟系统

7.1 简介

时钟系统的相关设置在系统复位后会被复位。

在 V85XX 中，有四个时钟源：

- 32K RC 振荡电路，用于产生 32.768K 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 6M 晶振时钟，用户需手动开启 6M 晶振时钟输出。

在 V85XX 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 32.768K 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 32.768K RC 时钟产生，为 RTC\WDT\UART32K\LCD 提供时钟。

表65. 不同工作模式下的时钟模块开关

时钟源	电源模式		
	睡眠	IDLE	工作
6.5536MRC	OFF	由 RCHPD 控制	
6.5536MXTAL	OFF	由 XOHPDN 控制	
PLLH	OFF	由 PLLHPDN 控制	
PLLL	OFF	由 PLLLPDN 控制	
32KRC	ON	ON	
32KXTAL	ON	ON	
时钟源	不同模式唤醒		
	睡眠	IDLE	
6.5536MRC	由 RCHPD 控制		
6.5536MXTAL	由 XOHPDN 控制		
PLLH	由 PLLHPDN 控制		
PLLL	由 PLLLPDN 控制		

32K RC	ON	ON	ON
32K XTAL	ON	ON	ON

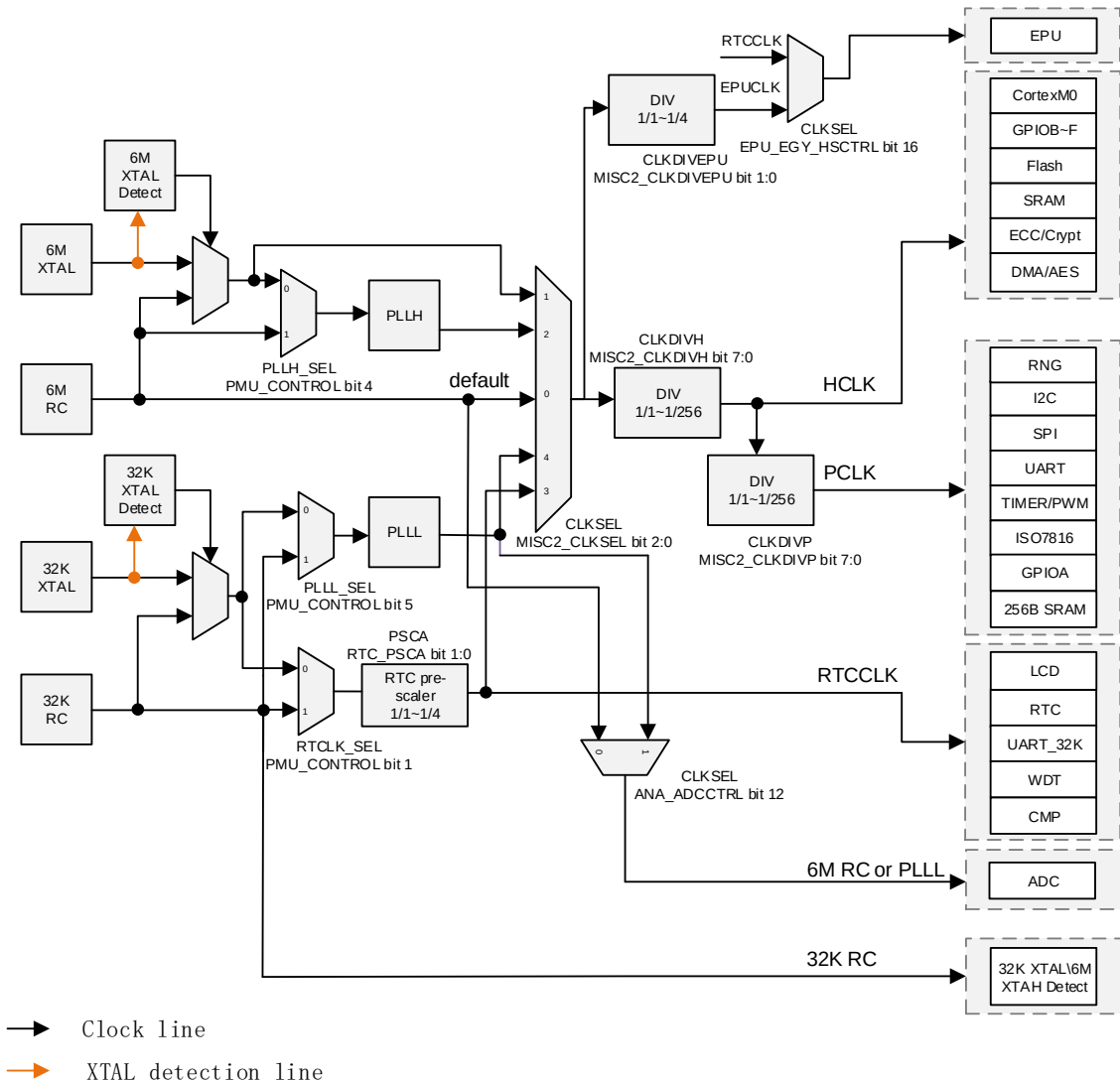
7.2 特点

- 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 32.768 XTAL 停止振荡，32K RC 将自动替换 32K XTAL，并且在 32.768K XTAL 恢复之后，自动使用 32.768K XTAL。

注1: HCLK 指的是 AHB 时钟，也叫做 AHBCLK；同样，PCLK 指的是 APB 时钟，也叫做 APBCLK。

7.3 功能框图

图17. V85XX 时钟系统功能框图



7.4 寄存器地址

表66. ANA 控制寄存器（ANA 基地址：0x40014200）

名称	类型	地址	描述	默认值
ANA_REG2	rw	0x0008	模拟寄存器 2	0x00
ANA_REG3	rw	0x000C	模拟寄存器 3	0x00
ANA_REG4	rw	0x0010	模拟寄存器 4	0x00
ANA_REG9	rw	0x0024	模拟寄存器 9	0x00
ANA_REGB	rw	0x002C	模拟寄存器 11	来自 FLASH
ANA_REGC	rw	0x0030	模拟寄存器 12	来自 FLASH
ANA_REG10	rw	0x0040	模拟寄存器 16	来自 FLASH
ANA_REG11	rw	0x0044	模拟寄存器 17	来自 FLASH
ANA_CMPOUT	r	0x0054	比较器结果寄存器	0x0030

表67. PMU 寄存器地址（PMU 基地址：0x40014000）

名称	类型	地址	描述	默认值
PMU_CONTROL	rw	0x0008	PMU 控制寄存器	0x0000
PMU_STS	rc_w1	0x000C	PMU 状态寄存器	0x0000074

表68. MISC2 控制寄存器（MISC2 基地址：0x40013E00）

名称	类型	地址	描述	默认值
MISC2_CLKSEL	rw	0x0004	时钟选择寄存器	0x0
MISC2_CLKDIVH	rw	0x0008	AHB 时钟分频控制寄存器	0x00
MISC2_CLKDIVP	rw	0x000C	APB 时钟分频控制寄存器	0x01
MISC2_HCLKEN	rw	0x0010	AHB 时钟使能控制寄存器	0x1FF
MISC2_PCLKEN	rw	0x0014	APB 时钟使能控制寄存器	0xF83FFFFFF
MISC2_EPUCLKCTRL	rw	0x0020	EPU 时钟控制寄存器	0x0

7.5 寄存器定义

7.5.1 ANA_REG3 寄存器

表69. ANA_REG3 控制寄存器

位	名称	功能	注意事项
3	BGPPD*	BGP 掉电控制。	0: 开启 BGP 1: 关闭 BGP
4	RCHPD	高频 RC 时钟 RCH（6.5536MRC）开关控制信号。	0: 开启 RCH 1: 关闭 RCH
5	PLLLPDN	PLLL（32768Hz 输入 PLL）开关控制信号。	0: 关闭 PLLL 1: 开启 PLLL
6	PLLHPDN	PLLH（6.5536MHz 输入 PLL）开关控制信号。	0: 关闭 PLLH 1: 开启 PLLH
7	XOHPDN	6.5536M 晶振时钟开关控制信号。	0: 关闭 XOH 1: 开启 XOH
*: 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。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70. ANA_REG4 控制寄存器

位	名称	功能	注意事项
7: 2	Rsvd	保留	默认值是 0x00，必须配置为 0x00。
1: 0	XRSEL[1: 0]	XOL 晶振下拉电阻	XOL 晶振下拉电阻选择。 00: 使能下拉电阻 210K 01: 使能下拉电阻 105K 10/11: 旁路下拉电阻 XOL 晶体上拉电阻选择，详见 ANA_REG10。

7.5.3 ANA_REG9 寄存器

表71. ANA_REG9 控制寄存器

位	名称	功能	注意事项
2: 0	PLLLSEL[2: 0]	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
6: 3	PLLHSEL[3: 0]	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

7.5.4 ANA_REGB 寄存器

表72. ANA_REGB 控制寄存器

位	名称	功能	注意事项
4: 0	RCLTRIM[4: 0]	32kHz RC 时钟频率微调。	00000~01111: 增加 4%/步长; 10000~11111: 减少 4%/步长
注意: 用户不要对寄存器 ANA_REGx (x=B~E) 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73. ANA_REGC 控制寄存器

位	名称	功能	注意事项
5: 0	RCHTRIM[5: 0]	6.5536MHzRC 时钟频率微调。	000000~011111: 增加 1.25%/步长; 100000~111111: 减少 1.25%/步长。
注意: 用户不要对寄存器 ANA_REGx (x=B~E) 进行配置, 这几个寄存器的值会从 FLASH 自动加载。			

7.5.6 ANA_REG10 寄存器

表74. ANA_REG10 控制寄存器

位	名称	功能	注意事项
6: 0	Rsvd	保留	0
7	XOL_LP	XOL 晶振上拉电阻选择	XOL 晶振上拉电阻选择。 0: 5.3MΩ 1: 450KΩ XOL 晶振下拉电阻选择, 详见 ANA_REG4
注意: ANA_REGx (x=10~11), 这两个寄存器的值会从 FLASH 自动加载。			

7.5.7 ANA_REG11 寄存器

表75. ANA_REG11 控制寄存器

位	名称	功能	注意事项
6: 0	Rsvd	保留	0
7	PLLL_X_SEL	PLLL 倍频选择	PLLL 倍频选择, 配置 1 时倍频 1.2 倍 0: x 1 1: x 1.2
注意: ANA_REGx (x=10~11), 这两个寄存器的值会从 FLASH 自动加载。			

7.5.8 ANA_CMPOUT 寄存器

表76. 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 15μs。 0: PLLH 未锁定。 1: PLLH 已锁定。	0x0

7.5.9 PMU_CONTROL 寄存器

表77. PMU_CONTROL 寄存器

位	名称	类型	描述	默认值
5	PLLL_SEL	rw	低速 PLL 输入时钟源选择。 0: 32K XTAL 1: 32K RC	0x0
4	PLLH_SEL	rw	高速 PLL 输入时钟源选择。 0: 6.5MHz RC 1: 6.5536MHz XTAL	0x0
3	INT_6M_EN	rw	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	rw	32K 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 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	rw	RTC 时钟选择。 0: 32K XTAL 1: 32K RC	0x0

7.5.10 PMU_STS 寄存器

表78. 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 的位 7 配置为 1）后，该状态位才会刷新，否则保持之前状态。	0x0

			0: 6.5536M 晶体不存在。 1: 6.5536M 晶体存在。	
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	rc_w1	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。 向此位写入 1 可将该标志清零。	0x0
0	INT_32K	rc_w1	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。 向此位写入 1 可将该标志清零。	0x0

7.5.11 MISC2_CLKSEL 寄存器

表79. MISC2_CLKSEL 寄存器

位	名称	类型	描述	默认值
31: 3	Rsvd	-	保留	0
2: 0	CLKSEL	rw	该寄存器用于控制 AHB 时钟源。 0: RCH (6.5M RC)。 1: XOH (6.5M XTAL)。 2: PLLH。 3: RTCCLK (由 PMU_CONTROL 寄存器中的 RTCCLK_SEL 控制)。 4: PLLL。 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

7.5.12 MISC2_CLKDIVH 寄存器

表80. MISC2_CLKDIVH 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7: 0	CLKDIVH	rw	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 ...	0x00

			255: 时钟源除以 256	

7.5.13 MISC2_CLKDIVP 寄存器

表81. MISC2_CLKDIVP 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7: 0	CLKDIVP	rw	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

7.5.14 MISC2_HCLKEN 寄存器

表82. MISC2_HCLKEN 寄存器

位	名称	类型	描述	默认值
31: 9	Rsvd	-	保留	0
8: 0	HCLKEN	rw	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83. HCLK 时钟使能

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

7.5.15 MISC2_PCLKEN 寄存器

表84. MISC2_PCLKEN 寄存器

位	名称	类型	描述	默认值
31: 0	PCLKEN	rw	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。	0xF83FFFFFF

表85. 各个模块的 PCLK 时钟使能

位	Module	Note
0	AHB2APB Bridge	禁止写 0
1	DMA 控制器	
2	I2C	
3	SPI1	
4	UART0	
5	UART1	
6	UART2	
7	UART3	
8	UART4	
9	UART5	
10	ISO78160	
11	ISO78161	
12	Timer	
13	MISC	
14	MISC2	
15	PMU	
16	RTC	
17	ANA	
18	U32K0	
19	U32K1	
20	保留	
21	SPI2	
22	EPU	
23	RNG	
24	SPI4	
25	保留	
26	SPI3	
31: 27	保留	

7.5.16 MISC2_EPUCLKCTRL 寄存器

表86. MISC2_EPUCLKCTRL 寄存器



位	名称	类型	描述	默认值
31: 5	Rsvd	-	保留	0
4	EPUCLKEN	rw	EPU 时钟控制 0: 关闭; 1: 开启。	0x0
3: 2	Rsvd	-	保留	0
1: 0	CLKDIVEPU	rw	EPU 时钟分频控制 0: 时钟源除以 1 1: 时钟源除以 2 2: 时钟源除以 3 3: 时钟源除以 4 注意: EPU 工作时钟是 6.5536MHz。	0x0

8 模拟控制器

8.1 简介

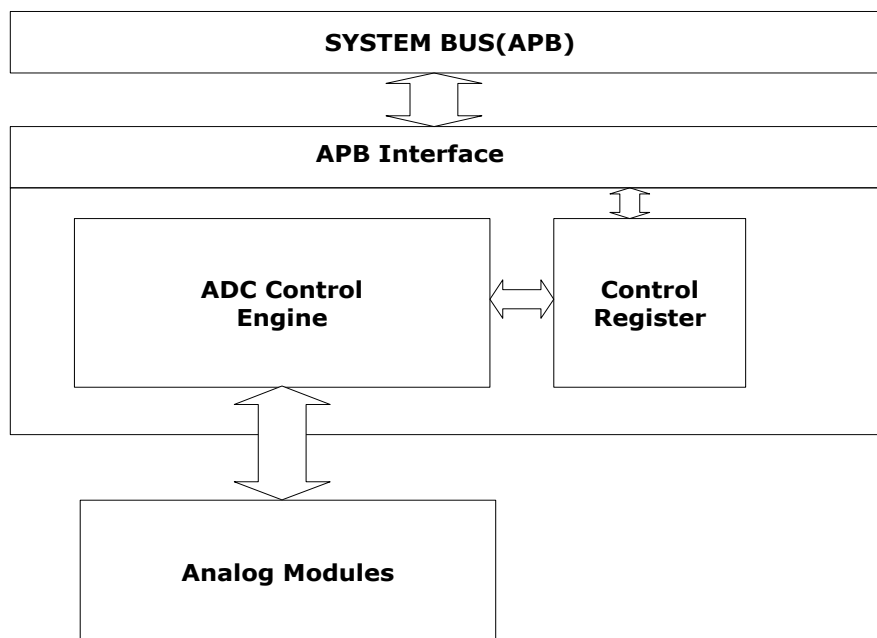
模拟控制器用于控制 V85XX 的模拟功能。所有模拟相关控制，如 ADC、比较器和其他模拟监测都由本模块控制。

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18. 模拟控制器的功能框图



8.4 寄存器地址

表87. ANA 控制寄存器（ANA 基地址：0x40014200）

名称	类型	地址	描述	默认值
ANA_REG0	rw	0x0000	模拟寄存器 0	0x00
ANA_REG1	rw	0x0004	模拟寄存器 1	0x00
ANA_REG2	rw	0x0008	模拟寄存器 2	0x00
ANA_REG3	rw	0x000C	模拟寄存器 3	0x00
ANA_REG4	rw	0x0010	模拟寄存器 4	0x00
ANA_REG5	rw	0x0014	模拟寄存器 5	0x00
ANA_REG6	rw	0x0018	模拟寄存器 6	0x00
ANA_REG7	rw	0x001C	模拟寄存器 7	0x00
ANA_REG8	rw	0x0020	模拟寄存器 8	0x00
ANA_REG9	rw	0x0024	模拟寄存器 9	0x00
ANA_REGA	rw	0x0028	模拟寄存器 10	0x00
ANA_REGB	rw	0x002C	模拟寄存器 11	From FLASH
ANA_REGC	rw	0x0030	模拟寄存器 12	From FLASH
ANA_REGD	rw	0x0034	模拟寄存器 13	From FLASH
ANA_REGE	rw	0x0038	模拟寄存器 14	From FLASH
ANA_REGF	rw	0x003C	模拟寄存器 15	0x00
ANA_REG10	rw	0x0040	模拟寄存器 16	From FLASH
ANA_REG11	rw	0x0044	模拟寄存器 17	From FLASH
ANA_CTRL	rw	0x0050	模拟控制寄存器	0x00000000
ANA_CMPOUT	r	0x0054	比较器结果寄存器	0x0030
ANA_INTSTS	rc_w1	0x0060	模拟中断状态寄存器	0x0000
ANA_INTEN	rw	0x0064	模拟中断使能寄存器	0x0000
ANA_ADCCTRL	rw	0x0068	ADC 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_ADCDATAA	r	0x0098	ADC 通道 10 数据寄存器	--
ANA_ADCDATAB	r	0x009C	ADC 通道 11 数据寄存器	--
ANA_CMPCNT1	rc_w1	0x00B0	比较器 1 计数器	0x00000000
ANA_CMPCNT2	rc_w1	0x00B4	比较器 2 计数器	0x00000000

8.5 寄存器定义

8.5.1 ANA_REGx 寄存器

表88. ANA_REGx

	位								
	7	6	5	4	3	2	1	0	地址
ANA_REG0	-	-							0x0000
ANA_REG1	-	-	GDE4	RESDIV					0x0004
ANA_REG2	-	-	REFSEL_CMP2	REFSEL_CMP1	CMP2_SEL[1: 0]		CMP1_SEL[1: 0]		0x0008
ANA_REG3	XOH_PDN	PLLHPDN	PLLLPDN	RCHPD	BGPPD	CMP2PDN	CMP1PDN	ADCPDN	0x000C
ANA_REG4							XRSEL		0x0010
ANA_REG5	-	PD_AVCDET			IT_CMP2[1: 0]		IT_CMP1[1: 0]		0x0014
ANA_REG6	BATRTCDISC	VDDDISC		VLCD[3: 0]				LCD_BMOD	0x0018
ANA_REG7	-	-	-	-	-	-	-	-	0x001C
ANA_REG8	PD_AVCDET	VDDPVDSSEL[2: 0]			-	-	-		0x0020
ANA_REG9	PD_VDDDET	PLLHSEL[3: 0]				PLLLSEL[2: 0]			0x0024
ANA_REGA	PD_VDCINDET	-	-	-	-	-	-	-	0x0028
ANA_REGB		-	-	RCLTRIM[4: 0]					0x002C
ANA_REGC	-	-	RCHTRIM[5: 0]						0x0030
ANA_REGD									0x0034
ANAREGE									0x0038
ANAREGF	COMP2P_SEL	COMP1P_SEL	BATVOL_SEL		BAT_SEL	AVCCO_EN	-		0x003C

	SEL								
ANA_REG10	XOL_L P		-	-	-	-	-	-	
ANA_REG11	PLLL _X_S EL	-	-	-	-	-	-	-	

8.5.2 ANA_REG0 寄存器

表89. ANA_REG0 各个位的功能描述

位	名称	功能	描述
7: 0	REG0_RSVD	保留	0

8.5.3 ANA_REG1 寄存器

表90. ANA_REG1 各个位的功能描述

位	名称	功能	描述
3: 0	Rsvd	保留	0
4	RESDIV	使能 ADC 输入信号的电阻分压	0: 不使能 1: 使能 1/4 电阻分压
5	GDE4	使能 ADC 输入信号的电容分压	0: 不使能 1: 使能 1/4 电容分压
7: 6	Rsvd	保留	0

8.5.4 ANA_REG2 寄存器

表91. ANA_REG2 各个位的功能描述

位	名称	功能	描述
1: 0	CMP1_SEL[1: 0]	CMP1 信号输入源选择	00: CMP1_P 与 REF 01: CMP1_N 与 REF 1*: CMP1_P 与 CMP1_N
3: 2	CMP2_SEL[1: 0]	CMP2 信号输入源选择	00: CMP2_P 与 REF 01: CMP2_N 与 REF 1*: CMP2_P 与 CMP2_N
4	REFSEL_CMP1	CMP1 的 REF 选择	0: VREF 1: BGPREF
5	REFSEL_CMP2	CMP2 的 REF 选择	0: VREF 1: BGPREF
7: 6	Rsvd	保留	0

8.5.5 ANA_REG3 寄存器

表92. ANA_REG3 各个位的功能描述

位	名称	功能	描述
0	ADCPDN	ADC 开关控制信号。ADC 关闭之前需要先禁止 ADC 自动转换。	0: 关闭 ADC 1: 开启 ADC
1	CMP1PDN	CMP1 开关控制信号	0: 关闭 CMP1 1: 开启 CMP1
2	CMP2PDN	CMP2 开关控制信号	0: 关闭 CMP2 1: 开启 CMP2
3	BGPPD*	BGP 开关控制信号	0: 开启 BGP 1: 关闭 BGP
4	RCHPD	高频 RC 时钟电路 RCH (6.5536MRC) 开关控制信号	0: 开启 RCH 1: 关闭 RCH
5	PLLLPDN	PLLL (32768Hz 输入 PLL) 开关控制信号	0: 关闭 PLLL 1: 开启 PLLL
6	PLLHPDN	PLLH (6.5536MHz 输入 PLL) 开关控制信号	0: 关闭 PLLH 1: 开启 PLLH
7	XOHPDN	6.5536M 晶体振荡电路开关控制信号	0: 关闭 XOH 1: 开启 XOH
*: 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。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93. ANA_REG4 各个位的功能描述

位	名称	功能	描述
1: 0	XRSEL[1: 0]	XOL 晶振下拉电阻选择	XOL 晶振下拉电阻选择。 00: 使能下拉电阻 210K 01: 使能下拉电阻 105K 10/11: 旁路下拉电阻 XOL 晶振上拉电阻选择, 详见 ANA_REG10。
7: 2	Rsvd	保留	默认值是 0x00, 必须配置为 0x00。

8.5.7 ANA_REG5 寄存器

表94. ANA_REG5 各个位的功能描述

位	名称	功能	描述
1: 0	IT_CMP1[1: 0]	CMP1 偏置电流选择	00: 20nA; 01: 100nA; 1*: 500nA;
3: 2	IT_CMP2[1: 0]	CMP2 偏置电流选择	00: 20nA; 01: 100nA; 1*: 500nA;
5: 4	Rsvd	保留	0
6	PD_AVCCDET	关闭 AVCC 低电压监测	0: 开启 AVCC 低电压监测 1: 关闭 AVCC 低电压监测
7	Rsvd	保留	0

8.5.8 ANA_REG6 寄存器

表95. ANA_REG6 各个位的功能描述

位	名称	功能	描述
0	LCD_BMODE	LCD 偏置模式选择	0: 1/3 偏置 1: 1/4 偏置
4: 1	VLCD[3: 0]	LCD 驱动电压设置	当 VLCD=0: Default 当 VLCD=0~5: 调整幅度 = $+60\text{mV} \times \text{VLCD}$ 当 VLCD=6~15: 调整幅度= $-60\text{mV} \times (\text{VLCD}-5)$
5	Rsvd	保留	0
6	VDDDISC	给 BAT 电池放电。 放电电阻是 1.7k, 放电电流是 $V_{\text{VDD}}/1.7\text{k}$ 。	0: 禁用从 VDD 到 GND 的 1.7k 欧姆电阻。 1: 使能从 VDD 到 GND 的 1.7k 欧姆电阻。
7	BATRCDISC	给 BATRTC 电池放电 放电电阻是 1.7k, 放电电流是 $V_{\text{BATRTC}}/1.7\text{k}$ 。	1: 使能从 BATRTC 到 GND 的 1.7k 电阻

8.5.9 ANA_REG7 寄存器

表96. ANA_REG7 各个位的功能描述

位	名称	功能	描述
7: 0	Rsvd	保留	默认值为 0, 必须配置为 0。

8.5.10 ANA_REG8 寄存器

表97. ANA_REG8 各个位的功能描述

位	名称	功能	描述
3: 0	Rsvd	保留	0
6: 4	VDDPVDSEL[2: 0]	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 111: 2.3V
7	PD_AVCCCLDO	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。 AVCCCLDO 只能在供电电压低于 3.6V 时关闭。

8.5.11 ANA_REG9 寄存器

表98. ANA_REG9 各个位的功能描述

位	名称	功能	描述
2: 0	PLLLSEL[2: 0]	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
6: 3	PLLHSEL[3: 0]	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
7	PD_VDDDET	关闭 VDD 电压监测模块, 该模块由 VDD 供电。	0: 开启。 1: 关闭。

8.5.12 ANA_REGA 寄存器

表99. ANA_REGA 各个位的功能描述

位	名称	功能	描述
6: 0	Rsvd	保留	默认值是 0x0, 必须配置为 0x0A。
7	PD_VDCINDET	关闭 VDCIN 监测模块	0: 开启 1: 关闭

8.5.13 ANA_REGB 寄存器

表100. ANA_REGB 各个位的功能描述

位	名称	功能	描述
4: 0	RCLTRIM[4: 0]	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
7: 5	VREFTRIM[2: 0]	VREF 电压调节, 它将同步影响 DVCC 和 AVCC 的输出电压值	011: +9%; 010: +6%; 001: +3%; 000: 0%; 100: -3%; 101: -6%; 110: -9%; 111: -12%;
注意: 用户不要对寄存器 ANA_REGx (x=B~E) 进行配置, 这几个寄存器的值会从 FLASH 自动加载。			

8.5.14 ANA_REGC 寄存器

表101. ANA_REGC 各个位的功能描述

位	名称	功能	描述
5: 0	RCHTRIM[5: 0]	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
7: 6	Rsvd	保留	0
注意: 用户不要对寄存器 ANA_REGx (x=B~E) 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102. ANA_REGD 各个位的功能描述

位	名称	功能	描述
7: 0	REGD_RSVD	保留	0
注意: 用户不要对寄存器 ANA_REGx (x=B~E) 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103. ANA_REGE 各个位的功能描述

位	名称	功能	描述
7: 0	REGE_RSVD	保留	0
注意: 用户不要对寄存器 ANA_REGx (x=B~E) 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104. ANA_REGF 各个位的功能描述

位	名称	功能	描述
1: 0	Rsvd	保留	0
2	AVCCO_EN	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
3	BAT_SEL	选择检测的 BAT	0: VDD 1: BATRTC
5: 4	BATVOL_SEL	BAT 电压选择	00: 2.6V 01: 2.4V 10: 2.2V 11: 2.0V
6	COMP1P_SEL	COMP1 正端来源选择	0: 由 CMP1_SEL 决定 1: 由 BAT_SEL 决定
7	COMP2P_SEL	COMP2 正端来源选择	0: 由 CMP2_SEL 决定 1: 由 BAT_SEL 决定

8.5.18 ANA_REG10 寄存器

表105. ANA_REG10 各个位的功能描述

位	名称	功能	描述
6: 0	Rsvd	保留	0
7	XOL_LP	XOL 晶振上拉电阻选择	XOL 晶振上拉电阻选择。 0: 5.3MΩ 1: 450KΩ XOL 晶体下拉电阻选择，详见 ANA_REG4。

注意：ANA_REGx (x=10~11)，这两个寄存器的值会从 FLASH 自动加载。

8.5.19 ANA_REG11 寄存器

表106. ANA_REG11 各个位的功能描述

位	名称	功能	描述
6: 0	Rsvd	保留	0
7	PLLL_X_SEL	PLLL 倍频选择	PLLL 倍频选择, 配置 1 时倍频 1.2 倍 0: x 1 1: x 1.2
注意: ANA_REGx (x=10~11), 这两个寄存器的值会从 FLASH 自动加载。			

8.5.20 ANA_CTRL 寄存器

表107. ANA_CTRL 寄存器

位	名称	类型	描述	默认值
31: 26	Rsvd	-	保留	0
25: 24	VDCINDEB	rw	VDCIN 去抖控制寄存器。 0: 无去抖 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。	0x0
23: 22	CMP2DEB	rw	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。	0x0
21: 20	CMP1DEB	rw	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。	0x0
19: 16	Rsvd	-	保留	0

15: 8	RCHTGT	rw	高频时钟 RCH 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 那么用户需要在此寄存器中填入 $6553600/32768=200$ 。	0x0
7: 4	Rsvd	-	保留	0
3: 2	COMP2_SEL	rw	该寄存器用于控制比较器 COMP2 的中断和唤醒。 0: 关闭 1: 上升沿 2: 下降沿 3: 上下沿	0x0
1: 0	COMP1_SEL	rw	该寄存器用于控制比较器 COMP1 的中断和唤醒。 0: 关闭 1: 上升沿 2: 下降沿 3: 上下沿	0x0

8.5.21 ANA_CMPOUT 寄存器

表108. ANA_CMPOUT 寄存器

位	名称	类型	描述	默认值
31: 11	Rsvd	-	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	Rsvd	-	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	Rsvd	-	保留	0
3	COMP2	r	比较器 2 输出。	0x0
2	COMP1	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。	0x0

			1: PLLH 已锁定。	

8.5.22 ANA_INSTS 寄存器

表109. ANA_INTSTS 寄存器

位	名称	类型	描述	默认值
31: 13	Rsvd	-	保留	0
12	INTSTS12	rc_w1	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。 Read0: 未发生中断 Read1: 发生中断 Write0: 无影响; Write1: 清除该位。	0x0
11	INTSTS11	rc_w1	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），使得系统可以从睡眠模式唤醒（VDCINDROP 为 0）。 Read0: 无进入睡眠事件发生（VDCINDROP=0） Read1: 发生进入睡眠事件（VDCINDROP=0） Write0: 无影响; Write1: 清除该位。	0x0
10	INTSTS10	rc_w1	AVCCLV 低电压中断标记位。 Read0: 未发生中断 Read1: 发生中断 Write0: 无影响; Write1: 清除该位。	0x0
9	Rsvd	-	保留	0
8	INTSTS8	rc_w1	VDCINDROP 的中断标志，该中断会在 VDCINDROP 上升和下降时产生。 Read0: 无 VDCIN 中断 Read1: 发生 VDCIN 中断 Write0: 无效 Write1: 清零	0x0
7	INTSTS7	rc_w1	VDDALARM 的中断标志，该中断会在 VDDALARM 上升和下降时产生。 Read0: 无 VDDALARM 中断 Read1: 发生 VDDALARM 中断 Write0: 无效 Write1: 清零	0x0

6: 4	Rsvd	-	保留	0
3	INTSTS3	rc_w1	COMP2 的中断标志, 中断产生条件受 COMP2_SEL 控制。读 0: 无 COMP2 中断。 读 1: 发生 COMP2 中断。 写 0: 无效。 写 1: 清零。	0x0
2	INTSTS2	rc_w1	COMP1 的中断标志, 中断产生条件受 COMP1_SEL 控制。读 0: 无 COMP1 中断。 读 1: 发生 COMP1 中断。 写 0: 无效。 写 1: 清零。	0x0
1	INTSTS1	rc_w1	ADC 自动转换完成中断标记位。 Read0: 未发生中断 Read1: 发生中断 Write0: 无影响; Write1: 清除该位。	0x0
0	INTSTS0	rc_w1	手动设置 ADC 转换完成中断标记位。 Read0: 未发生中断 Read1: 发生中断 Write0: 无影响; Write1: 清除该位。	0x0

8.5.23 ANA_INTEN 寄存器

表110. ANA_INTEN 寄存器

位	名称	类型	描述	默认值
31: 13	Rsvd	-	保留	0
12	INTEN12	rw	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	rw	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	rw	上升和下降时, AVCCCLV 中断或唤醒使能控制。 0: 禁止 AVCCCLV 中断或唤醒 1: 使能 AVCCCLV 中断或唤醒	0x0
9	Rsvd	-	保留	0
8	INTEN8	rw	上升和下降时, VDCINDROP 中断或唤醒使能控	0x0

			制。 0: 禁止 VDCIN 中断或唤醒 1: 使能 VDCIN 中断或唤醒	
7	INTEN7	rw	上升和下降时, 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
6: 4	Rsvd	-	保留	0
3	INTEN3	rw	COMP2 的中断和唤醒使能控制。 0: 禁止 COMP2 中断和唤醒。 1: 使能 COMP2 中断和唤醒。	0x0
2	INTEN2	rw	COMP1 的中断和唤醒使能控制。 0: 禁止 COMP1 中断和唤醒。 1: 使能 COMP1 中断和唤醒。	0x0
1	INTEN1	rw	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	rw	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

8.5.24 ANA_ADCCTRL 寄存器

表111. ANA_ADCCTRL 寄存器

位	名称	类型	描述	默认值
31	MTRIG	rw	手动控制 ADC 转换。 Write0: 无影响 Write1: 使能一次手动 ADC 转换 Read0: 当前手动设置的 ADC 转换已完成 Read1: 当前手动设置的 ADC 转换未完成	0x0
30	Rsvd	-	保留	0
29	CICAON	rw	CIC 滤波器开启控制寄存器 0: 只有在 ADC 转换进行时开启 CICI 滤波器 1: CIC 滤波器始终开启	0x0
28	CICINV	rw	CIC 滤波器输入反向控制 0: 无需反向 1: 将 CIC 输入信号反向	0x0
27	CICSCA	rw	CIC 输出值缩小 2 倍选择 0: 输出值不变 1: 输出值缩小 2 倍	0x0
26: 24	CICSKIP	rw	鉴于 ADC 和 CIC 滤波器需要一定的稳定时间, 该寄存器用来选择忽略最开始的 CIC 输出采样点的数	0x0

			量。如果 CICAON=1 同时 ADC 采样通道没有变过，那么该功能将失效。 0: 忽略最开始的 4 个采样点 1: 忽略最开始的 5 个采样点 2: 忽略最开始的 6 个采样点 3: 忽略最开始的 7 个采样点 4: 忽略最开始的 0 个采样点 5: 忽略最开始的 1 个采样点 6: 忽略最开始的 2 个采样点 7: 忽略最开始的 3 个采样点	
23: 22	DSRSEL	rw	CIC 降采样率控制寄存器。降采样率越高，数据输出结果越稳定，收敛时间越长。 0: 1/512 1: 1/256 2: 1/128 3: 1/64	0x0
21	AMODE	rw	ADC 自动转换模式控制。 0: 根据 ACH 的设置做单通道转换 1: 12 通道转换	0x0
20	MMODE	rw	手动设置 ADC 转换模式控制。 0: 根据 MCH 的设置做单通道转换 1: 12 通道转换	0x0
19	Rsvd	-	保留。该寄存器位必须保持为 0。	0x0
18: 16	AEN	rw	ADC 自动转换使能控制寄存器。 0: 禁止 ADC 自动转换 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	Rsvd	-	保留	0
12	CLKSEL	rw	ADC 时钟源选择 0: 6.5M RCH 1: PLLL	0x0
11: 8	CLKDIV	rw	ADC 时钟分频设置。典型工作时钟为 1.6384MHz。 ADC 时钟频率=ADC 时钟源频率/(CLKDIV+1)。 比如: ADC 时钟源为内部高频 RCH (6.5536MHz)，此时，应该设置寄存器 CLKDIV=3，以使得 ADC 的时钟频率 1.6384MHz。	0x0
7: 4	ACH	rw	ADC 自动转换通道控制 0: 通道 0 1: 通道 1 ...	0x0

			11: 通道 11 12~15: 保留 只有当 AMODE=0 时该寄存器有效	
3: 0	MCH	rw	ADC 手动转换通道控制 0: 通道 0 1: 通道 1 ... 11: 通道 11 12~15: 保留 只有当 MMODE=0 时该寄存器有效	0x0

8.5.25 ANA_ADCDATAx 寄存器

表112. ANA_ADCDATAx 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	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。 ... DATA11: 存储 ADC 通道 11 转换结果。 单通道转换 (MMODE=0 或 AMODE=0) 时, 只有被 MCH 或 ACH 选择通道对应的 DATA 值更新。多通道转换 (MMODE=1 或 AMODE=1), 更新 12 个采样通道的转换值。	0x--

8.5.26 ANA_CMPCNTx 寄存器

表113. ANA_CMPCNTx 寄存器

位	名称	类型	描述	默认值
31: 0	CNTx	rc_w1	该寄存器用于存储 COMPx_SEL 设置的比较器 x 的事件次数的计数器。比如 COMPx_SEL 被设置为 1, 当 COMPx 发生上升沿时, 该计数器加一。通过写 0 可以将该寄存器清零, 但这个操作只有在 CMPPDNx=1 时有效。	0x0000000

9 ADC

9.1 简介

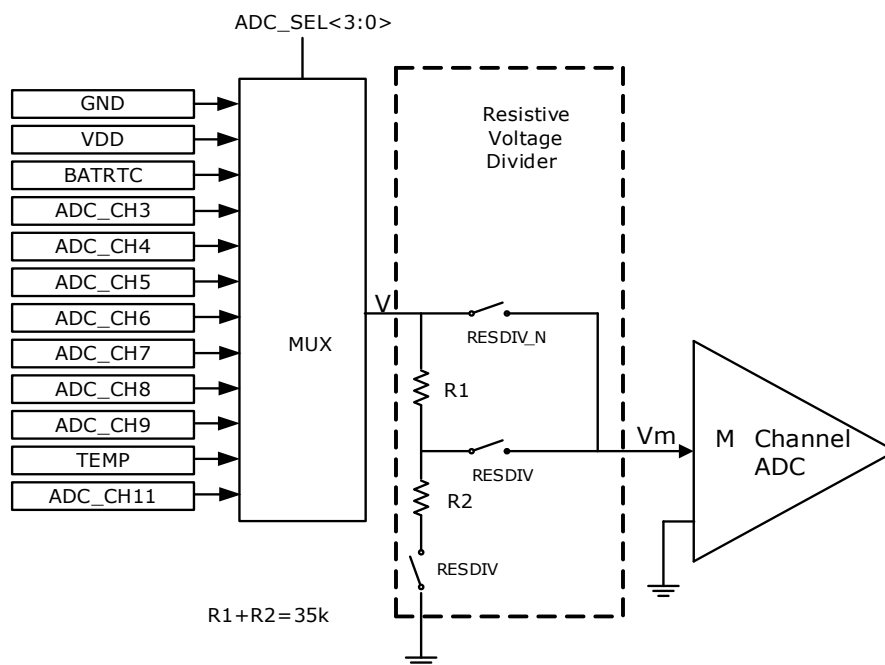
芯片中集成了一个二阶 $\Sigma\Delta$ ADC，支持 12 通道的信号输入。所有通道的采样量程在不分压情况下均为 0~1.2V。分压情况下，3.3V 供电时，各通道量程为 0~3.6V。5V 供电时，采用电容分压，各通道量程为 0~4.6V；采用电阻分压，各通道量程为 0~5.2V。ADC 采样通道包含 GND、温度、VDD、BATRTC 电压以及外部输入信号。ADC 的时钟来自 6.5536M RC 或 PLLL。

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 12 通道全扫描功能。

9.3 ADC 功能框图

图19. ADC 功能框图



9.4 寄存器地址

表114. ANA 控制寄存器（ANA 基地址：0x40014200）

名称	类型	地址	描述	默认值
ANA_REG0	rw	0x0000	模拟寄存器 0	0x00
ANA_REG1	rw	0x0004	模拟寄存器 1	0x00
ANA_REG3	rw	0x000C	模拟寄存器 3	0x00
ANA_INTSTS	rc_w1	0x0060	模拟中断状态寄存器	0x0000
ANA_INTEN	rw	0x0064	模拟中断使能寄存器	0x0000
ANA_ADCCTRL	rw	0x0068	ADC 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_ADCDATAA	r	0x0098	ADC 通道 10 数据寄存器	--
ANADCATAB	r	0x009C	ADC 通道 11 数据寄存器	--

9.5 寄存器定义

9.5.1 ANA_REG1 寄存器

表115. ANA_REG1 寄存器

位	名称	功能	描述
3: 0	Rsvd	保留	0
4	RESDIV	使能 ADC 输入信号的电阻分压	0: 不使能 1: 使能 1/4 电阻分压
5	GDE4	使能 ADC 输入信号的电容分压	0: 不使能 1: 使能 1/4 电容分压
7: 6	Rsvd	保留	0

9.5.2 ANA_REG3 寄存器

表116. ANA_REG3 各个位的功能描述

位	名称	功能	描述
0	ADCPDN	ADC 开关控制信号。ADC 关闭之前需要先禁止 ADC 自动转换。	0: 关闭 1: 开启
3	BGPPD	BGP 掉电控制	0: 开启 BGP 1: 关闭 BGP
4	RCHPD	高频 RC 时钟 RCH (6.5536MRC) 开关控制信号	0: 开启 RCH 1: 关闭 RCH
*: 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。睡眠模式下, 不管 BGP 是否有被用到, 都可以关闭 BGP。			

9.5.3 ANA_INSTS 寄存器

表117. ANA_INTSTS 寄存器

位	名称	类型	描述	默认值
1	INTSTS1	rc_w1	ADC 自动转换完成中断标志。 读 0: ADC 自动转换未完成 读 1: ADC 自动转换完成 写 0: 无效 写 1: 清零	0x0
0	INTSTS0	rc_w1	ADC 手动转换完成中断标志。 读 0: ADC 手动转换未完成 读 1: ADC 手动转换完成 写 0: 无效 写 1: 清零	0x0

9.5.4 ANA_INTEN 寄存器

表118. ANA_INTEN 寄存器

位	名称	类型	描述	默认值
1	INTEN1	rw	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	rw	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

9.5.5 ANA_ADCCTRL 寄存器

表119. ANA_ADCCTRL 寄存器

位	名称	类型	描述	默认值
31	MTRIG	rw	触发 ADC 手动转换 Write0: 无影响 Write1: 开始 ADC 手动转换 Read0: 当前 ADC 手动转换已完成 Read1: 当前 ADC 手动转换未完成	0x0
30	Rsvd	-	保留	0
29	CICAON	rw	CIC 滤波器始终开启控制寄存器。 0: 只有在 ADC 转换进行时开启 CICI 滤波器 1: CIC 滤波器始终开启	0x0
28	CICINV	rw	CIC 滤波器反向输入控制 0: 无需反向 1: 将 CIC 输入信号反向	0x0
27	CICSCA	rw	CIC 输出值缩小 2 倍选择 0: 输出值不变 1: 输出值缩小为 1/2	0x0
26: 24	CICSKIP	rw	鉴于 ADC 和 CIC 滤波器需要一定的稳定时间, 该寄存器用来选择忽略最开始的 CIC 输出采样点的数量。如果 CICAON=1 同时 ADC 采样通道没有变过, 那么该功能将失效。 0: 忽略起始的 4 个采样点 1: 忽略起始的 5 个采样点 2: 忽略起始的 6 个采样点 3: 忽略起始的 7 个采样点 4: 忽略起始的 0 个采样点 5: 忽略起始的 1 个采样点 6: 忽略起始的 2 个采样点 7: 忽略起始的 3 个采样点	0x0
23: 22	DSRSEL	rw	CIC 降采样率控制寄存器。降采样率越高, 数据输出越稳定, 采样率越低。 0: 1/512 1: 1/256 2: 1/128 3: 1/64	0x0
21	AMODE	rw	ADC 自动模式控制 0: 根据 ACH 的设置做单通道转换 1: 12 通道转换	0x0
20	MMODE	rw	手动设置 ADC 转换模式控制 0: 根据 MCH 的设置做单通道转换	0x0

			1: 12 通道转换	
19	Rsvd	-	保留。该寄存器位必须保持为 0。	0x0
18: 16	AEN	rw	ADC 自动转换使能控制 0: 禁止 ADC 自动转换 4: 定时器 0 溢出时触发 ADC 自动转换 5: 定时器 1 溢出时触发 ADC 自动转换 6: 定时器 2 溢出时触发 ADC 自动转换 7: 定时器 3 溢出时触发 ADC 自动转换	0x0
15: 13	Rsvd	-	保留	0
12	CLKSEL	rw	ADC 时钟源选择 0: 6.5M RCH 1: PLLL	0x0
11: 8	CLKDIV	rw	ADC 时钟分频设置。典型工作时钟为 1.6384MHz。 ADC 时钟频率=ADC 时钟源频率/(CLKDIV+1)。 比如: ADC 时钟源为内部高频 RCH (6.5536MHz), 此时, 应该设置寄存器 CLKDIV=3, 以使得 ADC 的时钟频率 1.6384MHz。	0x0
7: 4	ACH	rw	ADC 自动转换通道控制 0: 通道 0 1: 通道 1 ... 11: 通道 11 12~15: 保留 只有当 AMODE=0 时该寄存器有效 ADC 采样通道参见下面的表格	0x0
3: 0	MCH	rw	ADC 手动转换通道控制 0: 通道 0 1: 通道 1 ... 11: 通道 11 12~15: 保留 只有当 MMODE=0 时该寄存器有效 ADC 采样通道参见下面的表格	0x0

表120. ADC 采样通道

位	名称	通道
7: 4	ACH	0000: GND; 0001: VDD 0010: BATRTC; 0011: ADC_CH3
3: 0	MCH	0100: ADC_CH4; 0101: ADC_CH5

		0110: ADC_CH6; 0111: ADC_CH7 1000: ADC_CH8; 1001: ADC_CH9 1010: TEMP; 1011: ADC_CH11 1100~1111: 保留 当选择 TEMP 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。

9.5.6 ANA_ADCDATAx 寄存器

表121. ANA_ADCDATAx 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	-
15: 0	ADCDATAx	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。 ... DATA11: 存储 ADC 通道 11 转换结果。 单通道转换（MMODE=0 或 AMODE=0）时，只有 被 MCH 或 ACH 选择通道对应的 DATA 值更新。多 通道转换（MMODE=1 或 AMODE=1），更新 12 个 采样通道的转换值。	0x--

9.6 电池电压测量和外部电压测量

在 V85XX 中，ADC 能通过外部输入管脚测量外部信号，也能测量 VDD、BATRTC 等信号的电压。

ADC 工作模式：定时器自动触发 ADC 转换、手动触发 ADC 转换。

ADC 采样通道选择：单通道、12 通道。

ADC 分压模式：不分压、电阻分压、电容分压。电阻分压有额外功耗。因为内部分压电路有一个 35kΩ 的电阻，这个电路消耗的功率：

$$P = U_D \times I_D = V \times \frac{V}{R_1 + R_2} = \frac{V^2}{35000}$$

Equation 9-1

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 VDD、BATRTC 测量（通道 1、2）。

用户需要检查用于存储 ADC 系数的 a、b（A、B）存储区域参数是否合法，如果合法，采用该存储区域数据做运算，如果非法，ADC 系数采用固定参数做运算，运算公式见表 122。ADC 系数的



a、b（A、B）见表 43。

表122. 电压测量量程与公式

Power	Channel	Divider Mode	Signal range (V)	Formula
3.3V	ADC_CHx	No divider	-0.2~1.2	如果 ADC 的 a、b 参数合法， $V_{DC} = a1 \div 100000000 * X + b1 \div 100000000$ 否则 $V_{DC} = 0.00003680 * X + 0.00205011$
		Resistive	-0.2~3.6	如果 ADC 的 a、b 参数合法， $V_{DC} = a2 \div 100000000 * X + b2 \div 100000000$ 否则 $V_{DC} = 0.00016425 * X + 0.03739179$
		Capacitive	-0.2~3.6	如果 ADC 的 a、b 参数合法， $V_{DC} = a3 \div 100000000 * X + b3 \div 100000000$ 否则 $V_{DC} = 0.00014051 * X - 0.00023322$
	VDD channels	Resistive	2.5~4.8	如果 ADC 的 a、b 参数合法， $V_{DC} = a4 \div 100000000 * X + b4 \div 100000000$ 否则 $V_{DC} = 0.00015392 * X + 0.06667986 + \text{OffsetBatR} \div 1000$
		Capacitive	2.5~4.6	如果 ADC 的 a、b 参数合法， $V_{DC} = a5 \div 100000000 * X + b5 \div 100000000$ 否则 $V_{DC} = 0.00014107 * X - 0.00699515 + \text{OffsetBatC} \div 1000$
	BATRTC channels	Resistive	0.7~4.8	如果 ADC 的 a、b 参数合法， $V_{DC} = a6 \div 100000000 * X + b6 \div 100000000$ 否则 $V_{DC} = 0.00015392 * X + 0.06667986 + \text{OffsetBatR} \div 1000$
		Capacitive	0.7~4.6	如果 ADC 的 a、b 参数合法， $V_{DC} = a7 \div 100000000 * X + b7 \div 100000000$ 否则 $V_{DC} = 0.00014107 * X - 0.00699515 + \text{OffsetBatC} \div 1000$
5V	ADC_CHx	No divider	-0.2~1.2	如果 ADC 的 A、B 参数合法， $V_{DC} = A1 \div 100000000 * X + B1 \div 100000000$ 否则 $V_{DC} = 0.00003678 * X + 0.00235783$
		Resistive	-0.2~4.8	如果 ADC 的 A、B 参数合法， $V_{DC} = A2 \div 100000000 * X + B2 \div 100000000$ 否则 $V_{DC} = 0.00016129 * X + 0.00673599$

	VDD channels	Capacitive	0.2~4.6	如果 ADC 的 A、B 参数合法， $V_{DC} = A3 \div 100000000 * X + B3 \div 100000000$ 否则 $V_{DC} = 0.00014076 * X - 0.00753319$
		Resistive	2.5~4.8	如果 ADC 的 A、B 参数合法， $V_{DC} = A4 \div 100000000 * X + B4 \div 100000000$ 否则 $V_{DC} = 0.00015392 * X + 0.06667986 + \text{OffsetBatR} \div 1000$
		Capacitive	2.5~4.6	如果 ADC 的 A、B 参数合法， $V_{DC} = A5 \div 100000000 * X + B5 \div 100000000$ 否则 $V_{DC} = 0.00014107 * X - 0.00699515 + \text{OffsetBatC} \div 1000$
	BATRTC channels	Resistive	0.7~4.8	如果 ADC 的 A、B 参数合法， $V_{DC} = A6 \div 100000000 * X + B6 \div 100000000$ 否则 $V_{DC} = 0.00015392 * X + 0.06667986 + \text{OffsetBatR} \div 1000$
		Capacitive	0.7~4.6	如果 ADC 的 A、B 参数合法， $V_{DC} = A7 \div 100000000 * X + B7 \div 100000000$ 否则 $V_{DC} = 0.00014107 * X - 0.00699515 + \text{OffsetBatC} \div 1000$

上述公式中的 X 值计算公式如下：

$$X = \text{ADC\_DATA}_x$$

$$\text{Equation 9-2}$$

其中：

ADC_DATA_x 是地址（0x40014270+4x）（16 位）中的值，x 是对应的通道号 0-11。

9.7 温度测量

CH10 为温度测量通道，温度测量范围是-40~+85℃。

测量温度的步骤如下：

1. 设置ADC
2. ADC的时钟频率必须设置为1.6384Mhz，同时设置CIC的降采样率DSRSEL为1/512，并且不分压或电阻分压（不支持电容分压模式下测温）。设置手动或自动触发ADC。
3. 等待ANAINSTS（0x40014260）寄存器的位ANAINSTS0（手动触发）/ANAINSTS1（自动触发）值为1时，读寄存器ADC_DATA_A（0x40014298）的值。
4. 计算温度T：（单位：1/256℃），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：

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

X 是从寄存器 ADC_DATAA (16 位) 读出来的值

P0 是地址 0x40D00 (16 位) 中的值

P1 是地址 0x40D02 (16 位) 中的值

P2 是由以下公式计算出的值:

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

P2' 是地址 0x40D04 (32 位) 中的值

Tr 是地址 0x40D70 (32 位) 中的值

Tm 是地址 0x40D74 (32 位) 中的值

注1: 对于通道 10, 硬件内部做了 4 个点求平均, 内部用于求平均的四个缓存是个先入先出队列, 会被 DPOR 复位为 0。用户每做一次 ADC 转换, 硬件就往缓存写入一个数据, 并将内部四个缓存的数据的平均值计算出来, 放在 ADC 通道 10 寄存器。因此, 为了得到某个温度点准确的温度值, 需要用户连续转换 ADC 四次, 取最后一次 ADC 通道 10 寄存器数据作为 ADC 数据。

9.8 ADC 注意事项

1. ADC 自动触发转换流程

开启自动触发 ADC 转换步骤: 打开 ADC, 配置并开启定时器。

关闭自动触发 ADC 转换步骤: 关闭定时器, 再关闭 ADC。

2. ADC 相关的 ANA_ADCCTRL 寄存器写操作

对 ANA_ADCCTRL 寄存器写操作前, 需要保证位 31 (MTRIG) 读回来的值为 0, 即当前 ADC 手动转换已经完成。

10 比较器控制器

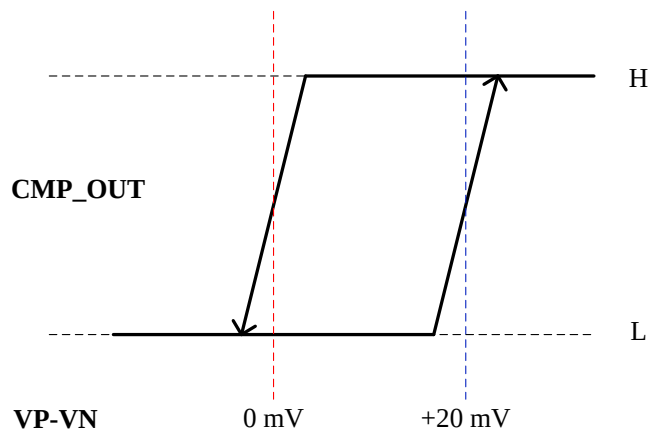
10.1 简介

V85XX 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 **VDD** 或者 **BATRTC**，负端信号接 **BGPREF** 或 **VREF**。

每个比较器有 20mv 的电压回差。

图20. 比较器的迟滞窗口

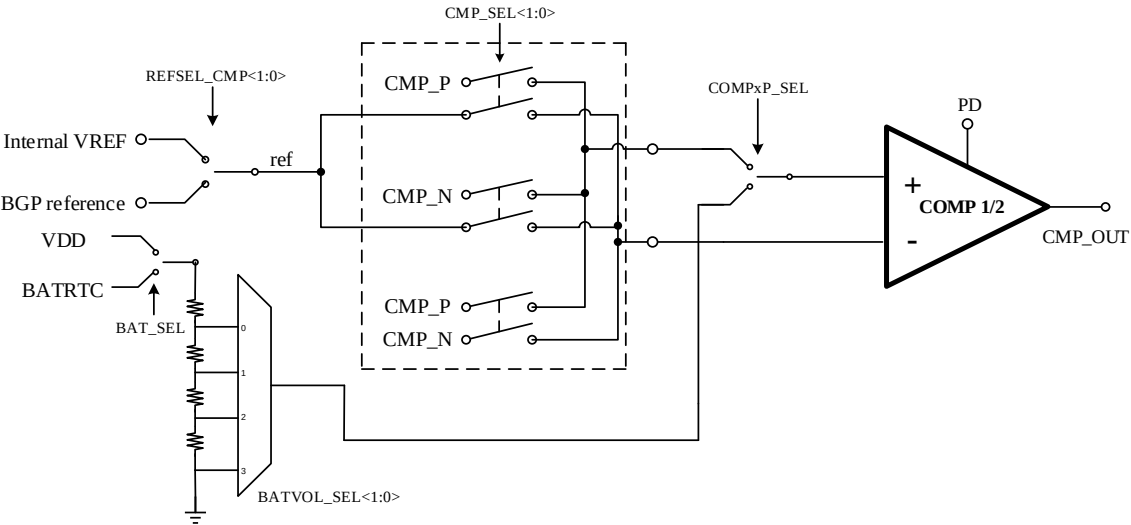


10.2 特点

- 比较器有中断/唤醒功能
- 有多种比较模式
- 有比较器计数器
- 在睡眠下也能工作

10.3 功能框图

图21. 比较器功能框图



10.4 寄存器地址

表123. ANA 控制寄存器（ANA 基地址：0x40014200）

名称	类型	地址	描述	默认值
ANA_REG2	rw	0x0008	模拟寄存器 2	0x00
ANA_REG3	rw	0x000C	模拟寄存器 3	0x00
ANA_REG5	rw	0x0014	模拟寄存器 5	0x00
ANA_REGF	rw	0x003C	模拟寄存器 15	0x00
ANA_CTRL	rw	0x0050	模拟控制寄存器	0x00000000
ANA_CMPOUT	r	0x0054	比较结果寄存器	0x0030
ANA_INTSTS	rc_w1	0x0060	模拟中断状态寄存器	0x00
ANA_INTEN	rw	0x0064	模拟中断使能寄存器	0x00
ANA_CMPCNT1	rc_w1	0x00B0	比较器 1 计数器	0x00000000
ANA_CMPCNT2	rc_w1	0x00B4	比较器 2 计数器	0x00000000

10.5 寄存器定义

10.5.1 ANA_REG2 寄存器

表124. ANA_REG2 寄存器

位	名称	功能	描述
1: 0	CMP1_SEL[1: 0]	CMP1 信号源选择	00: CMP1_P 与 REF 01: CMP1_N 与 REF 1*: CMP1_P 与 CMP1_N
3: 2	CMP2_SEL[1: 0]	CMP2 信号源选择	00: CMP2_P 与 REF 01: CMP2_N 与 REF

			1*: CMP2_P 与 CMP2_N
4	REFSEL_CMP1	CMP1 的参考电压	0: VREF 1: BGPREF
5	REFSEL_CMP2	CMP2 的参考电压	0: VREF 1: BGPREF

10.5.2 ANA_REG3 寄存器

表125. ANA_REG3 寄存器

位	名称	功能	描述
1	CMP1PDN	CMP1 掉电控制信号	0: 关闭 CMP1 1: 开启 CMP1
2	CMP2PDN	CMP2 掉电控制信号	0: 关闭 CMP2 1: 开启 CMP2
3	BGPPD	BGP 掉电控制信号	0: 上电 BGP 1: 掉电 BGP
*: 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。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126. ANA_REG5 寄存器

位	名称	功能	描述
1: 0	IT_CMP1[1: 0]	CMP1 的偏置电流选择。	00: 20nA; 01: 100nA; 1*: 500nA;
3: 2	IT_CMP2[1: 0]	CMP2 的偏置电流选择	00: 20nA; 01: 100nA; 1*: 500nA;
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127. ANA_REGF 寄存器

位	名称	功能	描述
3	BAT_SEL	BAT 选择	0: VDD 1: BATRTC
5: 4	BATVOL_SEL	BAT 电压选择	00: 2.6V 01: 2.4V 10: 2.2V 11: 2.0V
6	COMP1P_SEL	COMP1 正端来源选择	0: 由 COMP1_SEL 决定 1: 由 BAT_SEL 决定
7	COMP2P_SEL	COMP2 正端来源选择	0: 由 COMP2_SEL 决定 1: 由 BAT_SEL 决定

10.5.5 ANA_CTRL 寄存器

表128. ANA_CTRL 寄存器

位	名称	类型	描述	默认值
23: 22	CMP2DEB	rw	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, 包括睡眠模式。	0x0
21: 20	CMP1DEB	rw	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, 包括睡眠模式。	0x0
19: 18	Rsvd	-	保留	0
3: 2	COMP2_SEL	rw	该寄存器用来控制比较器 2 中断和唤醒信号。 0: 关闭 1: 比较器 2 的上升沿。 2: 比较器 2 的下降沿。 3: 比较器 2 的双边沿。	0x0
1: 0	COMP1_SEL	rw	该寄存器用来控制比较器 1 的中断和唤醒信号。	0x0

			0: 关闭 1: 比较器 1 的上升沿。 2: 比较器 1 的下降沿。 3: 比较器 1 的双边沿。	

10.5.6 ANA_CMPOUT 寄存器

表129. ANA_CMPOUT 寄存器

位	名称	类型	描述	默认值
3	COMP2	r	该位显示比较器 2 的结果。	0x0
2	COMP1	r	该位显示比较器 1 的结果。	0x0

10.5.7 ANA_INSTS 寄存器

表130. ANA_INTSTS 寄存器

位	名称	类型	描述	默认值
3	INTSTS3	rc_w1	COMP2 的中断标志，中断产生条件受 COMP2_SEL 控制。读 0: 无 COMP2 中断。 读 1: 发生 COMP2 中断。 写 0: 无效。 写 1: 清零。	0x0
2	INTSTS2	rc_w1	COMP1 的中断标志，中断产生条件受 COMP1_SEL 控制。读 0: 无 COMP1 中断。 读 1: 发生 COMP1 中断。 写 0: 无效。 写 1: 清零。	0x0

10.5.8 ANA_INTEN 寄存器

表131. ANA_INTEN 寄存器

位	名称	类型	描述	默认值
3	INTEN3	rw	COMP2 的中断和唤醒使能控制。 0: 禁止 COMP2 中断和唤醒。 1: 使能 COMP2 中断和唤醒。	0x0
2	INTEN2	rw	COMP1 的中断和唤醒使能控制。 0: 禁止 COMP1 中断和唤醒。 1: 使能 COMP1 中断和唤醒。	0x0

10.5.9 ANA_CMPCNTx 寄存器

表132. ANA_CMPCNTx 寄存器



位	名称	类型	描述	默认值
31: 0	CNTx	rc_w1	根据 COMPx_SEL 的设置，该寄存器储存比较器 X 的发生次数。例如，当 COMPx_SEL 设置为 1，当 COMPx 上升，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

11 看门狗

11.1 简介

V85XX 内部集成 24 位的看门狗（WDT）电路，WDT 时钟源是 32kHz 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。WDT 复位等级与 POR 复位等级相同，可提供芯片内部的复位。

11.2 特点

- WDT 的复位周期可选为：512s\256s\128s\64s\2s\1s\0.5s\0.25s。
- WDT 具有密码保护机制，防止对 WDT 的非法访问。

11.3 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 位 24）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133. 看门狗在不同模式下状态（MODE=1）

看门狗	电源模式		
	睡眠	IDLE	工作
开/关	ON（WDTEN=1） OFF（WDTEN=0）	ON（WDTEN=1） OFF（WDTEN=0）	ON（WDTEN=1） OFF（WDTEN=0）
看门狗	不同模式唤醒		
	睡眠	IDLE	
开/关	ON	ON（WDTEN=1） OFF（WDTEN=0）	

11.4 寄存器地址

表134. PMU_WDT 控制器（PMU 基地址：0x40014000）

名称	类型	地址	描述	默认值
PMU_WDTPASS	rw	0x0040	看门狗解锁寄存器	0x00000000
PMU_WDTEN	rw	0x0044	看门狗定时器使能寄存器	0x1
PMU_WDTCCLR	w	0x0048	清看门狗寄存器	0x0000
PMU_WDTSTS	rc_w1	0x0060	看门狗定时器状态寄存器	0x00



11.5 寄存器定义

11.5.1 PMU_WDTPASS 寄存器

表135. PMU_WDTPASS 寄存器

位	名称	类型	描述	默认值
31: 1	Rsvd	-	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

11.5.2 PMU_WDTEN 寄存器

表136. PMU_WDTEN 寄存器

位	名称	类型	描述	默认值
31: 5	Rsvd	-	保留	0
4: 2	WDTSEL	rw	设置 WDT 的计数周期。 0: 2 秒 1: 1 秒 2: 0.5 秒 3: 0.25 秒 4: 512 秒 5: 256 秒 6: 128 秒 7: 64 秒 在设置该寄存器之前，首先应将 PMU_WDTPASS 的 UNLOCK 位置 1。	0x0
1	Rsvd	-	保留	0
0	WDTEN	rw	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

11.5.3 PMU_WDTCLR 寄存器

表137. PMU_WDTCLR 寄存器

位	名称	类型	描述	默认值
31: 24	Rsvd	-	保留	0
23: 0	WDTCNT	rc_w1	该寄存器表示看门狗定时器的当前计数值。当该定时器	0x0

			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。	

11.5.4 PMU_WDTSTS 寄存器

表138. PMU_WDTSTS 寄存器

位	名称	类型	描述	默认值
31: 1	Rsvd	-	保留	0
0	WDTSTS	rc_w1	该位指示上一次复位是否由看门狗复位信号引起。写 1 清零该位。	0x0

12 RTC 控制器

12.1 简介

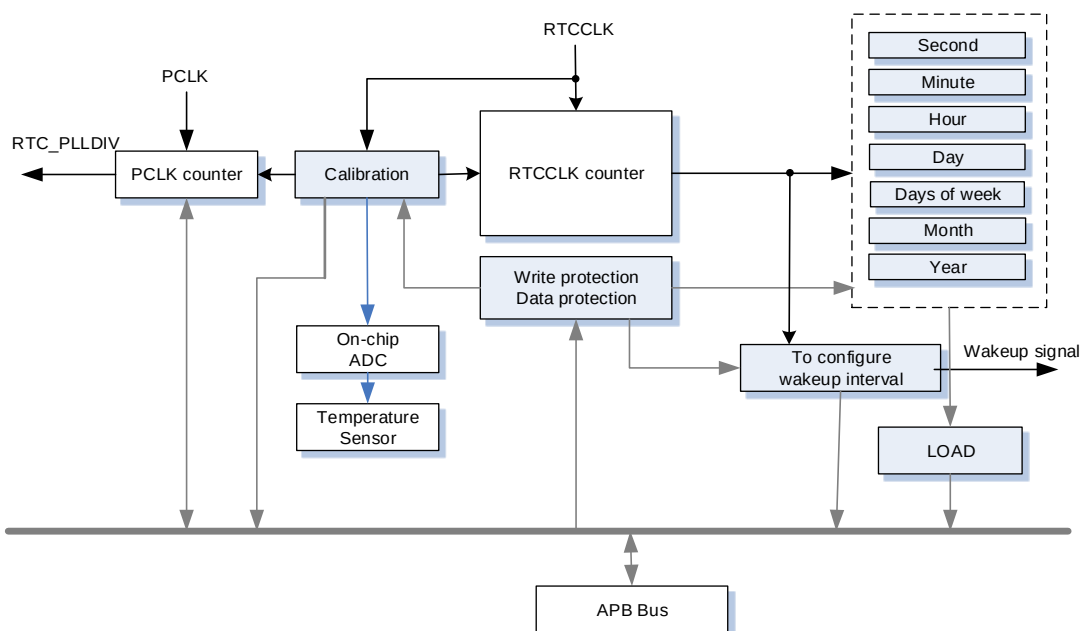
RTC 控制器用来控制时间的计算。时间计算功能可以实现年、月、星期、日、小时、分、秒自动计算功能和闰年监测。RTC 引擎内置多秒、多分、多小时和 RTC 计数器自动唤醒定时器。

12.2 特点

- 支持从秒到年的 BCD 格式的时间计算。
- 自动闰年监测。
- 自动温度监测。
- 晶体时钟频率补偿。
- 可编程的多唤醒源。
- 为厂家提供校正功能。
- RTC 分频在睡眠模式下实现低功耗。

12.3 功能框图

图22. RTC 控制器功能框图



12.4 寄存器读写

12.4.1 RTC 寄存器写操作

V85XX 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 139。

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寄存器的位1）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寄存器的位0）标志。当CE从1清为0时，BSY（RTC_CE寄存器的位1）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寄存器的位1）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。

12.4.2 RTC 寄存器读操作

RTC 时间寄存器受读保护，详情可参考表 139。

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寄存器的位1）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寄存器的位1）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100μs；
- 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。

12.5 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139. 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	rw	0x0000	RTC 秒寄存器	--	V	V



RTC_MIN	rw	0x0004	RTC 分寄存器	--	V	V
RTC_HOUR	rw	0x0008	RTC 时寄存器	--	V	V
RTC_DAY	rw	0x000C	RTC 天寄存器	--	V	V
RTC_WEEK	rw	0x0010	RTC 周寄存器	--	V	V
RTC_MON	rw	0x0014	RTC 月寄存器	--	V	V
RTC_YEAR	rw	0x0018	RTC 年寄存器	--	V	V
RTC_WKUSEC	rw	0x0020	RTC 秒唤醒寄存器	0x00	V	
RTC_WKUMIN	rw	0x0024	RTC 分唤醒寄存器	0x00	V	
RTC_WKUHOURL	rw	0x0028	RTC 时唤醒寄存器	0x00	V	
RTC_WKUCNT	rw	0x002C	RTC 计数唤醒寄存器	0x00000000	V	
RTC_CAL	rw	0x0030	RTC 校正寄存器	--	V	
RTC_DIV	rw	0x0034	RTC_PLLDIV 分频寄存器	0x00000000		
RTC_CTL	rw	0x0038	RTC_PLLDIV 分频器控制寄存器	0x0		
RTC_PWD	rw	0x0044	RTC 密码控制寄存器	0x00000000		
RTC_CE	rw	0x0048	RTC 写使能控制寄存器	0x0		
RTC_LOAD	rw	0x004C	RTC 读使能控制寄存器	--		
RTC_INTSTS	rw	0x0050	RTC 中断状态控制寄存器	0x000		
RTC_INTEN	rw	0x0054	RTC 中断使能控制寄存器	0x000		
RTC_PSCA	rw	0x0058	RTC 时钟预分频控制寄存器	0x0	V	
RTC_ACTI	rw	0x0084	晶体顶点温度寄存器	0x1800	V	
RTC_ACF200	rw	0x0088	RTC200 倍频率寄存器	0x640000	V	
RTC_ACADCW	rw	0x008C	RTC 手动写入 ADC 值寄存器	0x0000	V	
RTC_ACP0	rw	0x0090	RTC 参数 0 寄存器	0x0000	V	
RTC_ACP1	rw	0x0094	RTC 参数 1 寄存器	0x0000	V	
RTC_ACP2	rw	0x0098	RTC 参数 2 寄存器	0x0000	V	
RTC_ACP3	r	0x009C	RTC 参数 3 寄存器	0x0000		
RTC_ACP4	rw	0x00A0	RTC 参数 4 寄存器	0x0000	V	
RTC_ACP5	rw	0x00A4	RTC 参数 5 寄存器	0x0000	V	
RTC_ACP6	rw	0x00A8	RTC 参数 6 寄存器	0x0000	V	
RTC_ACP7	rw	0x00AC	RTC 参数 7 寄存器	0x0000	V	
RTC_ACK1	rw	0x00B0	RTC 参数 k1 寄存器	0x0000	V	
RTC_ACK2	rw	0x00B4	RTC 参数 k2 寄存器	0x0000	V	
RTC_ACK3	rw	0x00B8	RTC 参数 k3 寄存器	0x0000	V	
RTC_ACK4	rw	0x00BC	RTC 参数 k4 寄存器	0x0000	V	
RTC_ACK5	rw	0x00C0	RTC 参数 k5 寄存器	0x0000	V	
RTC_ACTEMP	r	0x00C4	记录 RTC 校正得到的温度值	0x0000		
RTC_ACPPM	r	0x00C8	记录 RTC 校正得到的 PPM 值	0x0000		
RTC_WKUCNTR	r	0x00CC	记录当前 WKUCNT 值	0x00000000		
RTC_ACKTEMP	rw	0x00D0	RTCK 系数温度分段点控制寄存器	0x3C2800EC	V	

RTC 不带写保护的寄存器会被上电复位、外部复位、看门狗复位这三种复位源复位。RTC 带写保护寄存器不会被外部复位、看门狗复位这两种复位源复位，除时间外的带写保护的寄存器会被上电复位这种复位源复位。不会被上电复位的 RTC 寄存器是：RTC_SEC、RTC_MIN、RTC_HOUR、RTC_DAY、RTC_WEEK、RTC_MON、RTC_YEAR、RTC_CAL。

12.6 寄存器定义

12.6.1 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，否则该寄存器值无效。

表140. RTC_SEC/MIN/DAY/WEEK/MONTH/YEAR 寄存器

寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
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
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。								

12.6.2 RTC_WKUSEC 寄存器

表141. RTC_WKUSEC 寄存器

位	名称	类型	描述	默认值
31: 6	Rsvd	-	保留	0
5: 0	WKUSEC	rw	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	0x00

			题，设置一个替代的值（如 0）到该寄存器和更新它的 RTC_CE 操作，然后设置正确的值。	

12.6.3 RTC_WKUMIN 寄存器

表142. RTC_WKUMIN 寄存器

位	名称	类型	描述	默认值
31: 6	Rsvd	-	保留	0
5: 0	WKUMIN	rw	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 $0 \sim (WKUMIN + 1)$ 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

12.6.4 RTC_WKUHOUR 寄存器

表143. RTC_WKU 时寄存器

位	名称	类型	描述	默认值
31: 5	Rsvd	-	保留	0
4: 0	WKUHOUR	rw	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

12.6.5 RTC_WKUCNT 寄存器

表144. RTC_WKUCNT 寄存器

位	名称	类型	描述	默认值
31: 26	Rsvd	-	保留	0

25: 24	CNTSEL	rw	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	rw	该寄存器是用来控制 32K 计数器唤醒功能。唤醒时间为: $(WKUCNT+1) \times \text{计数器时钟周期}$, 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

12.6.6 RTC_CAL 寄存器

表145. RTC_CAL 寄存器

位	名称	类型	描述	默认值
31: 14	Rsvd	-	保留	0
13: 0	CAL	rw	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 $\pm 8332.3\text{ppm}$ (± 12 分钟/天)。手动校正时, 需手动更新该寄存器的值。	--

12.6.7 RTC_DIV 寄存器

表146. RTC_DIV 寄存器

位	名称	类型	描述	默认值
31: 26	Rsvd	-	保留	0
25: 0	RTCDIV	rw	该寄存器用来配置 RTC_PLLDIV 输出频率。设置 IOA3 (PMU_IOASEL 位 3 置 1) 或 IOA7 (PMU_IOASEL 位 7 置 1) 为 RTC_PLLDIV 输出功能, 同时设置 RTC_PLLDIV 输出使能 (RTC_CTL 位 2), 则对应 IOA 输出分频脉冲。输出频率为 $PCLK / (2 * (RTCDIV + 1))$ 。RTC 手动	0x00000

			校正时，该寄存器需手动更新。	

12.6.8 RTC_CTL 寄存器

表147. RTC_CTL 寄存器

位	名称	类型	描述	默认值
31: 3	Rsvd	-	保留	0
2	RTCPLLOE	rw	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 位 3 置 1）或 IOA7（PMU_IOASEL 位 7 置 1）为 RTC_PLLDIV 输出使能。	0x0
1: 0	Rsvd	-	保留	0x0

12.6.9 RTC_PWD 寄存器

表148. RTC_PWD 寄存器

位	名称	类型	描述	默认值
31: 1	Rsvd	-	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

12.6.10 RTC_CE 寄存器

表149. RTC_CE 寄存器

位	名称	类型	描述	默认值
31: 2	Rsvd	-	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 100μs。	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

12.6.11 RTC_LOAD 寄存器

表150. RTC_LOAD 寄存器

位	名称	类型	描述	默认值
31: 0	LOAD	rw	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 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 位 1）位判断读取过程是否完成。从该寄存器读取的数据无效。	--

12.6.12 RTC_INTSTS 寄存器

表151. RTC_INTSTS 寄存器

位	名称	类型	描述	默认值
31:9	Rsvd	-	保留	0
8	INTSTS8	rc_w1	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 位 0）被置 1 或读取 RTC_LOAD 寄存器。写 1 清零该位。	0x0
7	Rsvd	-	保留	0x0
6	INTSTS6	rc_w1	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	rc_w1	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	rc_w1	中断状态 4，当到达由 WKU HOUR 设置的多小时中断周期时，该位置 1。写 1 清零该位。	0x0
3	INTSTS3	rc_w1	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	rc_w1	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	rc_w1	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	Rsvd	-	保留	0

12.6.13 RTC_INTEN 寄存器

表152. RTC_INTEN 寄存器

位	名称	类型	描述	默认值
31: 9	Rsvd	-	保留	0
8	INTEN8	rw	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	Rsvd	-	保留	0x0
6	INTEN6	rw	中断使能 6，当该位为 1 且 INSTS6 置 1 时，将产生 WKUCNT 设置周期计数中断或唤醒信号。	0x0
5	INTEN5	rw	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	rw	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	rw	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	rw	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	rw	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	Rsvd	-	保留	0

12.6.14 RTC_PSCA 寄存器

表153. RTC_PSCA 寄存器

位	名称	类型	描述	默认值
31:2	Rsvd	-	保留	0
1:0	PSCA	rw	该寄存器用于控制 RTCCLK 预分频器。RTCCLK 时钟速率慢时，可以显著降低睡眠模式下的功耗。 0：无预分频器，RTCCLK 为 32768 Hz。 1：1/4 预分频器，RTCCLK 为 8192 Hz。 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

12.6.15 RTC_ACTI 寄存器

表154. RTC_ACTI 寄存器

位	名称	类型	描述	默认值
31: 14	Rsvd	-	保留	0
13: 0	ACTI	rw	ACTI 表示 RTC 校正期间晶体中心温度的 Ti 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：~0x1880+1=0xE780。 该寄存器受 RTC 写保护，当 ACEN 为 1 时应为固定值。	0x1800

12.6.16 RTC_ACF200 寄存器

表155. RTC_ACF200 寄存器

位	名称	类型	描述	默认值
31: 26	Rsvd	-	保留	0
25: 0	F200	rw	该寄存器用于存储用于计算 PLLDIV 的值，即 PCLK/2。例如，当 PCLK 为 13107200， PCLK/2=6553600=0x640000。该寄存器受 RTC 写保护，当 ACEN 为 1 时应为固定值。	0x640000

12.6.17 RTC_ACADCW 寄存器

表156. RTC_ACADCW 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	符号扩展位	-
15: 0	ADCW	rw	该寄存器存储手动输入的 ADC 温度值。该寄存器为 16 位补码。	0x0000

12.6.18 RTC_ACP0 寄存器

表157. RTC_ACP0 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	符号扩展位	-
15: 0	P0	rw	P0 是 16 位的补码，用于温度校正。需从 Info 区加载数据，详情参考表 170。该寄存器受 RTC 写保护，并且在 ACEN 为 1 时应为固定值。	0x0000

12.6.19 RTC_ACP1 寄存器

表158. RTC_ACP1 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	符号扩展位	-
15: 0	P1	rw	P1 是 16 位的补码，用于温度校正。需从 Info 区加载数据，详情参考表 170。该寄存器受 RTC 写保护，并且在 ACEN 为 1 时应为固定值。	0x0000

12.6.20 RTC_ACP2 寄存器

表159. RTC_ACP2 寄存器

位	名称	类型	描述	默认值
31: 0	P2	rw	P2 是 32 位的补码，用于温度校正。需从 Info 区加载数据，详情参考表 170。该寄存器的值计算公式如下： $P2 = P2' + (Tr - Tm) * 256$ P2' 为 Info 区 0x40D04 地址存储值。 Tr 为 Info 区 0x40D70 地址存储值。 Tm 为 Info 区 0x40D74 地址存储值。 该寄存器受 RTC 写保护，并且在 ACEN 为 1 时应为固定值。	0x00000000

12.6.21 RTC_ACP3 寄存器

表160. 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 170。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 Kx (x=1~5) 并将系数存储至 RTC_ACP3 寄存器。该寄存器受 RTC 写保护，并且在 ACEN 为 1 时应为固定值。	0x00000000

12.6.22 RTC_ACP4 寄存器

表161. RTC_ACP4 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	符号扩展位	-
15: 0	P4	rw	P4 是晶体常温偏移量，是 16 位补码。需从 Info 区	0x0000

			加载数据，详情参考表 170。该寄存器受 RTC 写保护，并且在 ACEN 为 1 时应为固定值。	

12.6.23 RTC_ACP5 寄存器

表162. RTC_ACP5 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	符号扩展位	-
15: 0	P5	rw	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 170。该寄存器受 RTC 写保护，并且在 ACEN 为 1 时应为固定值。	0x0000

12.6.24 RTC_ACP6 寄存器

表163. RTC_ACP6 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	符号扩展位	-
15: 0	P6	rw	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： $P6=a*P6'$ $a=PCLK/6553600$ 。P6'为 Info 区加载值，详情参考表 170。 该寄存器受 RTC 写保护，并且在 ACEN 为 1 时应为固定值。	0x0000

12.6.25 RTC_ACP7 寄存器

表164. RTC_ACP7 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	符号扩展位	-
15: 0	P7	rw	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 170。该寄存器受 RTC 写保护，并且在 ACEN 为 1 时应为固定值。	0x0000

12.6.26 RTC_ACKx 寄存器

表165. RTC_ACKx 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	符号扩展位	-
15: 0	K1~K5	rw	K1~K5 是 16 位补码，用于频率校正。需从 Info 区	0x0000

			加载数据，详情参考表 170。RTC 手动温度补偿时，用户需根据计算的温度 T 手动匹配 Kx ($x=1\sim5$)，并根据 Kx 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，并且在 ACEN 为 1 时应为固定值。 K1: $T < KTEMP1$。 K2: $KTEMP1 \leq T < KTEMP2$。 K3: $KTEMP2 \leq T < KTEMP3$。 K4: $KTEMP3 \leq T < KTEMP4$。 K5: $T \geq KTEMP4$。	

12.6.27 RTC_ACKTEMP 寄存器

表166. RTC_ACKTEMP 寄存器

位	名称	类型	描述	默认值
31: 24	KTEMP4	rw	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 170。该寄存器受 RTC 写保护，并且在 ACEN 为 1 时应为固定值。	0x3C
23: 16	KTEMP3	rw	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 170。该寄存器受 RTC 写保护，并且在 ACEN 为 1 时应为固定值。	0x28
15: 8	KTEMP2	rw	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 170。该寄存器受 RTC 写保护，并且在 ACEN 为 1 时应为固定值。	0x00
7: 0	KTEMP1	rw	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 170。该寄存器受 RTC 写保护，并且在 ACEN 为 1 时应为固定值。	0xEC

12.6.28 RTC_ACTEMP 寄存器

表167. RTC_ACTEMP 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	符号扩展位	-
15: 0	TEMP	r	该寄存器用于存储当前温度的计算结果。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。实际温度放大 256 倍的结果。例如，	-

			0x1880 表示 24.5 度: $24.5 \times 256 = 6272 = 0x1880$; 0xE780 表示 -24.5 度: $\sim 0x1880 + 1 = 0xE780$ 。	

12.6.29 RTC_ACPPM 寄存器

表168. RTC_ACPPM 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	符号扩展位	-
15: 0	PPM	r	该寄存器用于存储低频晶振当前温度的计算结果偏差。该寄存器是一个 16 位补码, 该寄存器的单位为 0.1ppm。	-

12.6.30 RTC_WKUCNTR 寄存器

表169. RTC_WKUCNTR 寄存器

位	名称	类型	描述	默认值
31: 24	Rsvd	-	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。	-

12.7 Info 参数寄存器（与 RTC 温度补偿有关）

Info 信息存储在以 0x00040800 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 0x00040800~0x0004085C 数据, 通过烧写工具 (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170. Info 信息寄存器（与 RTC 温度补偿有关）

地址	符号	信息	描述
0x00040800	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。
0x00040804		Checksum1	INV (SUM (0x40800, 0x40800))
0x00040808	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。
0x0004080C		Checksum1	INV (SUM (0x40808, 0x40808))
0x00040810	K1	晶体 K1 系数 1	加载到 RTC_ACK1 寄存器, K1 计算公式如下: $K1 = B1 / 1000000 \times 65536$, B1 为晶振曲线二次函数的分段系数。K1 值比如 20827。

0x00040814	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 21496。
0x00040818	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 22020。
0x0004081C	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 24517。
0x00040820	K5	晶体 K5 系数 1	加载到 RTC_ACK5 寄存器, K5 计算公式如下: $K5=B5/1000000*65536$, B5 为晶振曲线二次函数的分段系数。K5 值比如 25257。
0x00040824		Checksum1	INV (SUM (0x40810, 0x40820))
0x00040828	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 20827。
0x0004082C	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 21496。
0x00040830	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 22020。
0x00040834	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 24517。
0x00040838	K5	晶体 K5 系数 2	加载到 RTC_ACK5 寄存器, K5 计算公式如下: $K5=B5/1000000*65536$, B5 为晶振曲线二次函数的分段系数。K5 值比如 25257。
0x0004083C		Checksum2	INV (SUM (0x40828, 0x40838))
0x00040840	ACTI	晶体顶点温度 1	加载到 RTC_ACTI 寄存器, 比如: 0x1800。
0x00040844		Checksum1	INV (SUM (0x40840, 0x40840))
0x00040848	ACTI	晶体顶点温度 2	加载到 RTC_ACTI 寄存器, 比如: 0x1800。
0x0004084C		Checksum2	INV (SUM (0x40848, 0x40848))
0x00040850	KTEMPx (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。
0x00040854		Checksum1	INV (SUM (0x40850, 0x40850))
0x00040858	KTEMPx (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。
0x0004085C		Checksum2	INV (SUM (0x40858, 0x40858))
0x00040D00	P1/P0	RTC_ACP1/0 设	高 16 位加载到 RTC_ACP1 寄存器, 比如:

		置 1	1060; 低 16 位加载到 RTC_ACP0 寄存器, 比如: -214。
0x00040D04	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。
0x00040D08	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。
0x00040D0C	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, 其中 $a=PCLK/6553600$ 。
0x00040D10		Checksum1	INV (SUM (0x40D00, 0x40D0C))
0x00040D14	P1/P0	RTC_ACP1/0 设置 2	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。
0x00040D18	P2'	RTC_ACP2 设置 2	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。
0x00040D1C	P5	RTC_ACP5 设置 2	高 16 位加载到 RTC_ACP5 寄存器, 比如: 6444; 低 16 位舍弃。
0x00040D20	P7/P6'	RTC_ACP7/6 设置 2	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, 其中 $a=PCLK/6553600$ 。
0x00040D24		Checksum2	INV (SUM (0x40D14, 0x40D20))
0x00040D70	Tr	实际温度 1 (来自 tmp275)	根据公式: $P2=P2'+(Tr-Tm)*256$, 将 P2 加载到 RTC_ACP2 寄存器。
0x00040D74	Tm	测量温度 1 (来自 ADC)	
0x00040D78		Temp Check sum 1	INV (SUM (0x40D70, 0x40D74))
0x00040D7C	Tr	实际温度 2 (来自 tmp275)	根据公式: $P2=P2'+(Tr-Tm)*256$, 将 P2 加载到 RTC_ACP2 寄存器。
0x00040D80	Tm	测量温度 2 (来自 ADC)	
0x00040D84		Temp Check sum 2	INV (SUM (0x40D7C, 0x40D80))

12.8 应用笔记

12.8.1 温度校正

根据下式计算温度 T ， T 为实际温度的 256 倍。

$$T = \left((P0 * ((X * X) >> 16)) + P1 * X + P2 \right) >> 8$$

其中 X 为 ADC 温度采样值，为 16 位补码。

$P0$ ~ $P2$ 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： $T' = T/256$

12.8.2 频率误差校正

1. 芯片频率误差计算公式如下：

$$\Delta = (P3 * ((T - Ti)^2 >> 16)) >> 16 + P4$$

其中， Δ 是 32K 晶体的频率偏差率（单位 0.1ppm）。

T = 温度计算值。

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。

$P4$ 位晶体常温偏差值。

$P3$ 是当前温度 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 Kx ($x=1\sim5$)，并根据 Kx 进行频率校正。不需存储值到 RTC_ACP3 寄存器。

2. 频率校正值计算公式如下：

$$RTC\_CAL = (\Delta * P5) >> 16$$

$P5$ 是 RTC_ACP5 寄存器的值，或根据公式：

$P5 = 65536 / 10 / (1 / 30 / 32768 / RTCCLK\text{分频值})$ 计算，即 RTC_CAL 值为 1 时 $P5$ 的值，式中 $(1 / 30 / 32768 / RTCCLK\text{分频值})$ 为寄存器 RTC_CAL 的 PPM 分辨率。

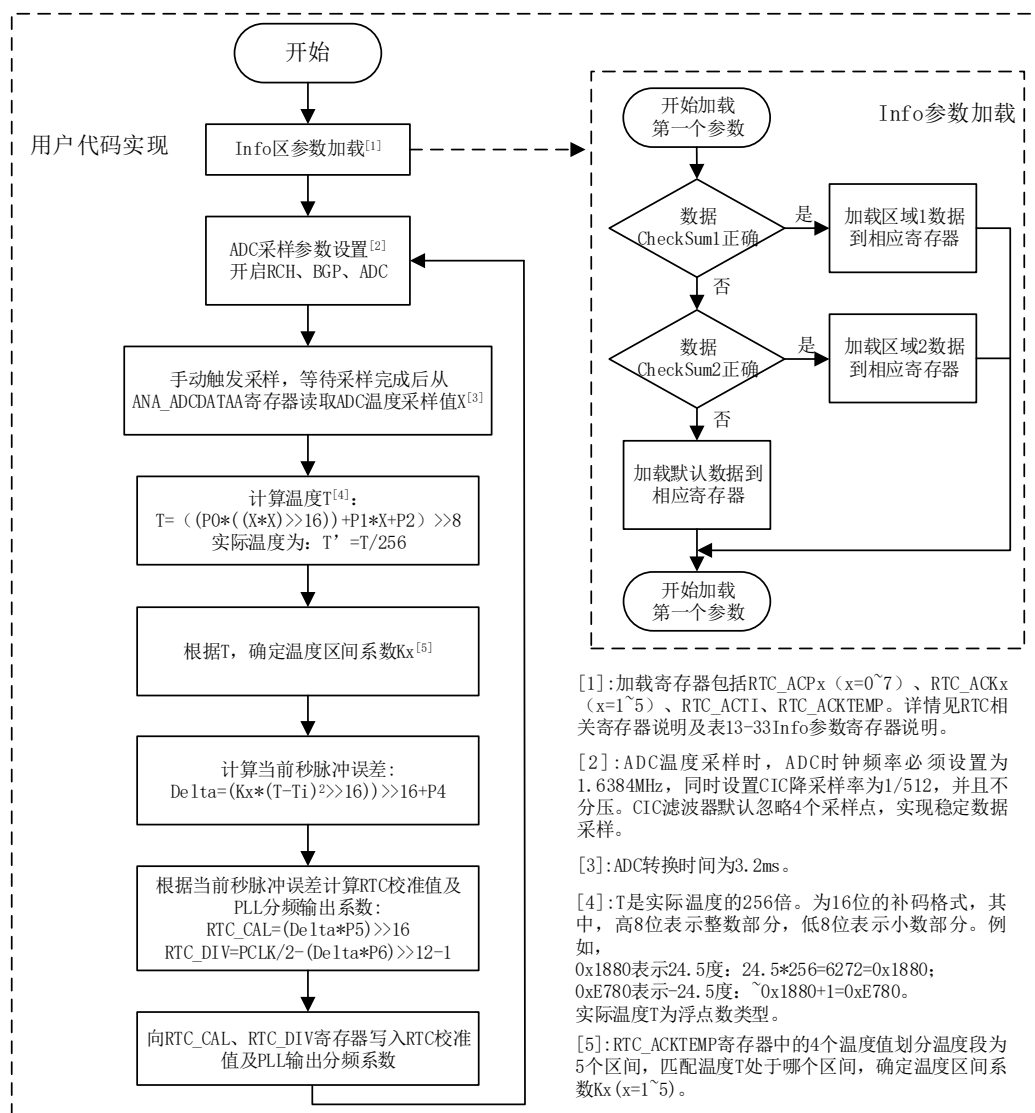
3. PLL分频秒脉冲输出计算公式如下：

$$RTC\_DIV = PCLK/2 - (\Delta * P6) >> 12 - 1$$

$P6$ 是 RTC_ACP6 寄存器的值，或根据公式： $P6 = PCLK * 2.048 * 10^{-4}$ 计算。

12.8.3 RTC 手动温度补偿

图23. RTC 手动温度补偿流程图



12.8.4 RTC 校正注意事项

RTC寄存器读写操作

对 RTC 寄存器的读操作和写操作要严格按照手册中的操作步骤进行。

13 FLASH 控制器

13.1 简介

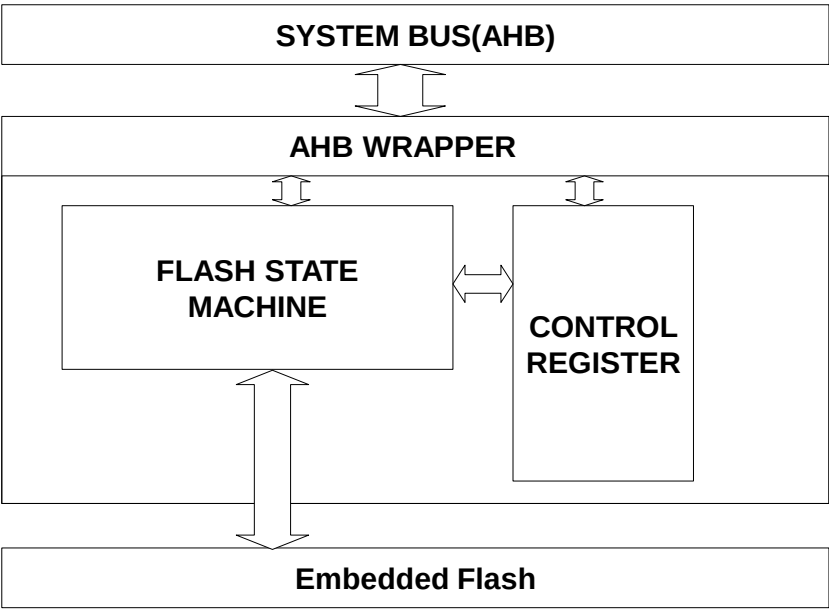
FLASH 控制器用于控制内置 FLASH 的读/写程序。它支持字节/半字/字程序和扇区/芯片擦除。可编程时序可由 MISC 控制器中的 1USCYCLE 控制。FLASH 控制器中的设置将在从睡眠模式唤醒后复位，用户应在从睡眠状态唤醒后手动恢复设置。

13.2 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，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 1μstick 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。

13.3 功能框图

图24. FLASH 控制器的功能框图



13.4 寄存器地址

表171. 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_STS	r	0xFFFFBC	FLASH 编程状态寄存器	0x00
FLASH_INT	rc_w1	0xFFFFCC	FLASH 校验和中断状态	0x0
FLASH_CSSADDR	rw	0xFFFFD0	FLASH 校验和开始地址	0x00000
FLASH_CSEADDR	rw	0xFFFFD4	FLASH 校验和结束地址	0x3FFFC
FLASH_CSVALUE	r	0xFFFFD8	FLASH 校验和计算值寄存器	--
FLASH_CSCVALUE	rw	0xFFFFDC	FLASH 校验和预期值设置寄存器	0x00000000
FLASH_PASS	rw	0xFFFE0	FLASH 密码寄存器	0x00000000
FLASH_CTRL	rw	0xFFFE4	FLASH 控制寄存器	0x0
FLASH_PGADDR	rw	0xFFFE8	FLASH 程序地址寄存器	0x00000
FLASH_PGDATA	rw	0xFFFE0	FLASH 程序字数据寄存器	--
FLASH_PGB0	rw	0xFFFE0	FLASH 程序字节数据寄存器 0	--
FLASH_PGB1	rw	0xFFFE4	FLASH 程序字节数据寄存器 1	--
FLASH_PGB2	rw	0xFFFE8	FLASH 程序字节数据寄存器 2	--
FLASH_PGB3	rw	0xFFFE0	FLASH 程序字节数据寄存器 3	--
FLASH_PGH0	rw	0xFFFE0	FLASH 程序半字数据寄存器 0	--
FLASH_PGH1	rw	0xFFFE4	FLASH 程序半字数据寄存器 1	--
FLASH_SERASE	rw	0xFFFF4	FLASH 扇区擦除控制寄存器	0x00000000
FLASH_CERASE	rw	0xFFFF8	FLASH 芯片擦除控制寄存器	0x00000000
FLASH_DSTB	rw	0xFFFFC	FLASH deep-standby 控制寄存器	0x00000000

表172. MISC2 控制器 for FLASH (MISC2 基地址: 0x40013E00)

名称	类型	地址	描述	默认值
MISC2_FLASHWC	rw	0x0000	FLASH 等待周期寄存器	0x2100

13.5 寄存器定义

13.5.1 FLASH_PASS 寄存器

表173. FLASH_PASS 寄存器

位	名称	类型	描述	默认值
31: 1	Rsvd	-	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

13.5.2 FLASH_CTRL 寄存器

表174. FLASH_CTRL 寄存器

位	名称	类型	描述	默认值
31: 4	Rsvd	-	保留	0
2	CSINTEN	rw	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	rw	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

13.5.3 FLASH_PGADDR 寄存器

表175. FLASH_PGADDR 寄存器

位	名称	类型	描述	默认值
31: 18	Rsvd	-	保留	0
17: 0	PGADDR	rw	flash 编程地址设置。这是 FLASH IP 的字节地址,	0x000000

			有关每种模式下的地址的详细信息，请参见表 176。当编程成功完成后，该地址将自动递增。	

表176. 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

13.5.4 FLASH_PGDATA 寄存器

表177. FLASH_PGDATA 寄存器

位	名称	类型	描述	默认值
31: 0	PGDATA	rw	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 程序的速度会变慢，因为 FLASH IP 需要更多的启动时间。建议将您的编程程序放在 SRAM 中，这样可以实现最大程序速度。	

13.5.5 FLASH_PGBx 寄存器

表178. FLASH_PGBx 寄存器

位	名称	类型	描述	默认值
7: 0	PGBx	rw	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。在此期间对 FLASH IP 的任何访问将封闭，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 程序的速度会变慢，因为 FLASH IP 需要更多的启动时间。建议将您的编程程	



			序放在 SRAM 中，这样可以实现最大程序速度。	

13.5.6 FLASH_PGHwX 寄存器

表179. FLASH_PGHwX 寄存器

位	名称	类型	描述	默认值
15: 0	PGHWx	rw	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。在此期间对 FLASH IP 的任何访问将封闭，直到程序完成。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 程序的速度会变慢，因为 FLASH IP 需要更多的启动时间。建议将您的编程程序放在 SRAM 中，这样可以实现最大程序速度。	

13.5.7 FLASH_SERASE 寄存器

表180. FLASH_SERASE 寄存器

位	名称	类型	描述	默认值
31: 1	Rsvd	-	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 512Byte。当 UNLOCK 为 1 且用户将 0xAA5555AA 写入该寄存器时，该位可以置 1，并且在扇区擦除完成后它将自动清零。扇区擦除的地址由 PGADDR[17: 2]控制。扇区擦除时间为 4000*1μs。	0x0

13.5.8 FLASH_CERASE 寄存器

表181. FLASH_CERASE 寄存器

位	名称	类型	描述	默认值
31: 1	Rsvd	-	保留	0
0	CERASE	r	该位用来表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，该位可以被置 1，芯片擦除完成后该位将自动清除。芯片擦除时间为 30000*1μs。	0x0

13.5.9 FLASH_DSTB 寄存器

表182. FLASH_DSTB 寄存器

位	名称	类型	描述	默认值
31: 1	Rsvd	-	保留	0
0	DSTB	r	该位用于指示 FLASH IP 是否正在进入 deep-standby。 当 UNLOCK 为 1 且用户将 0xAA5555AA 写入该寄存器时，该位可以置 1，并且在 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。进入 deep-standby 后功耗可以降低 80μA。 睡眠模式下，FLASH 自动进入 deep-standby。 IDLE 模式下，用户需要在 SRAM 中运行代码，手动配置该寄存器。	0x0

13.5.10 FLASH_INT 寄存器

表183. FLASH_INT 寄存器

位	名称	类型	描述	默认值
31: 1	Rsvd	-	保留	0
0	CSERR	rc_w1	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 可清除该标志。	0x0

13.5.11 FLASH_CSSADDR 寄存器

表184. FLASH_CSSADDR 寄存器

位	名称	类型	描述	默认值
31: 18	Rsvd	-	保留	0
17: 0	CSSADDR	rw	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

13.5.12 FLASH_CSEADDR 寄存器

表185. FLASH_CSEADDR 寄存器

位	名称	类型	描述	默认值
31: 18	Rsvd	-	保留	0

17: 0	CSEADDR	rw	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）。	0x3FFFC

13.5.13 FLASH_CSVALUE 寄存器

表186. 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

13.5.14 FLASH_CSCVALUE 寄存器

表187. FLASH_CSCVALUE 寄存器

位	名称	类型	描述	默认值
31: 0	CSCVALUE	rw	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

13.5.15 FLASH_STS 寄存器

表188. FLASH_STS 寄存器

位	名称	类型	描述	默认值
31: 5	Rsvd	-	保留	0
4: 0	STS	r	FLASH 控制器状态寄存器。 1: FLASH 操作完成。 其他: 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

13.5.16 MISC2_FLASHWC 寄存器

表189. MISC2_FLASHWC 寄存器

位	名称	类型	描述	默认值

31: 14	Rsvd	-	保留	0
13: 8	1USCYCLE	rw	这个寄存器用于 FLASH 控制器以 AHBClock 计算 1μs 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=1μs 的滴答时间*10。必须满足 1μstick>=1μs。 $1\mu stick = (AHB \text{ 时钟周期}) * (1USCYCLE + 1)$。 比如 AHB 时钟频率为 26.2144M，为了保证 1μs 的滴答时间，以及最小的唤醒时间，1USCYCLE 应该设置为 26。此时 FLASH 的唤醒时间为 $27 / 26214400 * 10$，大约为 10μs。 比如 AHB 时钟频率为 32.768k，为了保证 1μs 的滴答时间，以及最小的唤醒时间，1USCYCLE 应该设置为 0。此时 FLASH 的唤醒时间是 $1 / 32768 * 10$，大约为 305μs。	0x21
7: 0	Rsvd	-	保留	0

14 GPIO 控制器

14.1 简介

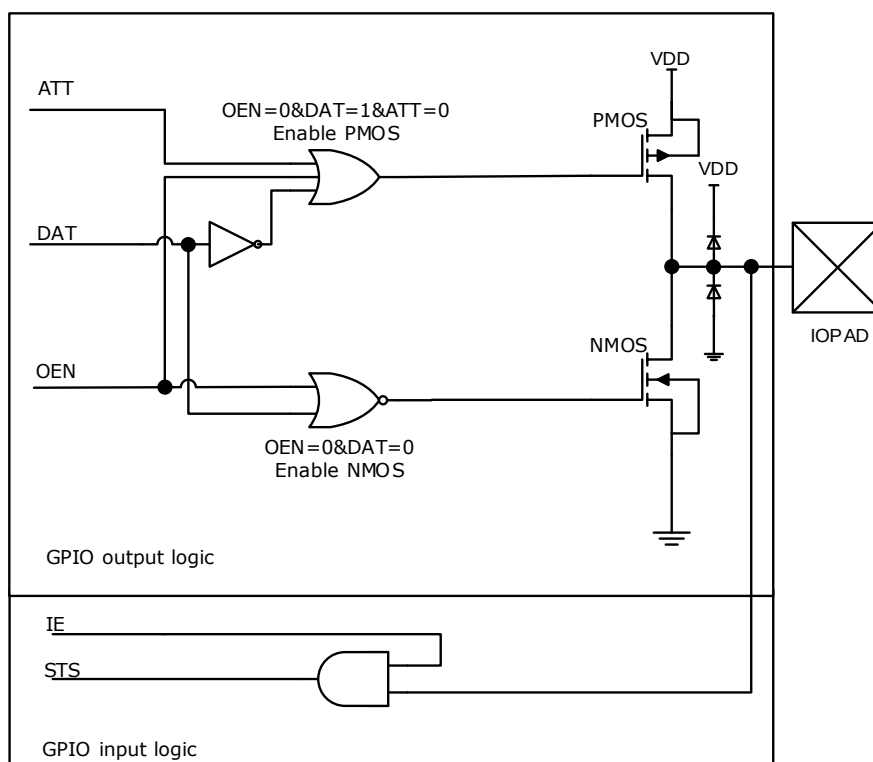
V85XX 中最多有 82 个 IO，其中 16 个 IO（GPIOA）由 PMU 控制器控制，可以唤醒睡眠。其余 IO（GPIOB~GPIOF）由 GPIO 控制器控制，不可以唤醒睡眠；睡眠下状态保持。

14.2 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 可以唤醒睡眠

14.3 功能框图

图25. GPIO 控制器的功能框图



14.4 寄存器地址

表190. PMU_IOA 控制器 (PMU 基地址: 0x40014000)

名称	类型	地址	描述	默认值
PMU_IOAOEN	rw	0x0010	IOA 输出使能寄存器	0xFFFF
PMU_IOAIE	rw	0x0014	IOA 输入使能寄存器	0xFFFF
PMU_IOADAT	rw	0x0018	IOA 数据寄存器	0x0000
PMU_IOAATT	rw	0x001C	IOA 属性寄存器	0x0000
PMU_IOAWKUEN	rw	0x0020	IOA 唤醒使能寄存器	0x00000000
PMU_IOASTS	r	0x0024	IOA 输入状态寄存器	--
PMU_IOAINTSTS	rc_w1	0x0028	IOA 中断状态寄存器	0x0000
PMU_IOASEL	rw	0x0038	IOA 特殊功能选择寄存器	0x0000
PMU_IOANODEG	rw	0x0050	IOA 去抖动电路控制	0x0000

表191. GPIO 控制器 (GPIO 基地址: 0x40000000)

名称	类型	地址	描述	默认值
IOB_OEN	rw	0x0020	IOB 输出使能寄存器	0xFFFF
IOB_IE	rw	0x0024	IOB 输入使能寄存器	0xFFFF
IOB_DAT	rw	0x0028	IOB 输出数据寄存器	0x0000
IOB_ATT	rw	0x002C	IOB 属性寄存器	0x0000
IOB_STS	r	0x0030	IOB 输入状态寄存器	--
IOC_OEN	rw	0x0040	IOC 输出使能寄存器	0xFFFF
IOC_IE	rw	0x0044	IOC 输入使能寄存器	0xFFFF
IOC_DAT	rw	0x0048	IOC 输出数据寄存器	0x0000
IOC_ATT	rw	0x004C	IOC 属性寄存器	0x0000
IOC_STS	r	0x0050	IOC 输入状态寄存器	--
IOD_OEN	rw	0x0060	IOD 输出使能寄存器	0xFFFF
IOD_IE	rw	0x0064	IOD 输入使能寄存器	0xFFFF
IOD_DAT	rw	0x0068	IOD 输出数据寄存器	0x0000
IOD_ATT	rw	0x006C	IOD 属性寄存器	0x0000
IOD_STS	r	0x0070	IOD 输入使能寄存器	--
IOE_OEN	rw	0x0080	IOE 输出使能寄存器	0xFFFF
IOE_IE	rw	0x0084	IOE 输入使能寄存器	0xFFFF
IOE_DAT	rw	0x0088	IOE 输出数据寄存器	0x0000
IOE_ATT	rw	0x008C	IOE 属性寄存器	0x0000
IOE_STS	r	0x0090	IOE 输入使能寄存器	--
IOF_OEN	rw	0x00A0	IOF 输出使能寄存器	0x1F
IOF_IE	rw	0x00A4	IOF 输入使能寄存器	0x1F
IOF_DAT	rw	0x00A8	IOF 输出数据寄存器	0x0
IOF_ATT	rw	0x00AC	IOF 属性寄存器	0x0
IOF_STS	r	0x00B0	IOF 输入使能寄存器	--
IOB_SEL	rw	0x00C0	IOB 特殊功能选择寄存器	0x00

IOE_SEL	rw	0x00CC	IOE 特殊功能选择寄存器	0x00
IO_MISC	rw	0x00E0	lomisc.控制寄存器	0x00

14.5 寄存器定义

14.5.1 PMU_IOAOEN 寄存器

表192. PMU_IOAOEN 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 0	IOAOEN	rw	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 196。 0: 使能 IO 输出模式。 1: 禁止 IO 输出模式。	0xFFFF

14.5.2 PMU_IOAIE 寄存器

表193. PMU_IOAIE 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 0	IOAIE	rw	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 196。 0: 禁止 IO 输入功能。 1: 使能 IO 输入功能。	0xFFFF

14.5.3 PMU_IOADAT 寄存器

表194. PMU_IOADAT 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 0	IOADAT	rw	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 196。	0x0000

14.5.4 PMU_IOAATT 寄存器

表195. PMU_IOAATT 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 0	IOAATT	rw	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	0x0000

			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）。	

表196. 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	输出低电平
0	1	0	输出高电平
0	0	1	开漏输出低电平
0	1	1	开漏拉高

14.5.5 PMU_IOAWKUEN 寄存器

表197. PMU_IOAWKUEN 寄存器

位	名称	类型	描述	默认值
31: 0	IOAWKUEN	rw	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 198。	0x00000000

表198. 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，表示某个 IO 口。

硬件在进入睡眠模式前锁存 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，包括正常中断和睡眠的 IOA 电平唤醒。去毛刺电路仅对睡眠模式的 IOA 边沿唤醒信号有效，去毛刺时间约为 4 个 RTCCLK 时钟周期。

14.5.6 PMU_IOASTS 寄存器

表199. PMU_IOASTS 寄存器

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

14.5.7 PMU_IOAINTSTS 寄存器

表200. PMU_IOAINTSTS 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 0	IOAINTSTS	rc_w1	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 会清零对应的标志位。	0x0000

14.5.8 PMU_IOASEL 寄存器

表201. PMU_IOASEL 寄存器

位	名称	类型	描述	默认值
31: 10	Rsvd	-	保留	0
9	IOA_SEL9	rw	IOA9 特殊功能选择寄存器。 0: GPIO。 1: 特殊功能 1 (CF1)。	0x0
8	Rsvd	-	保留	0
7	IOA_SEL7	rw	IOA7 特殊功能选择寄存器。 0: GPIO。 1: 特殊功能 1 (RTC_PLLDIV)。	0x0
6	IOA_SEL6	rw	IOA6 特殊功能选择寄存器。 0: GPIO。 1: 特殊功能 2 (CMP2_O)。	0x0
5: 4	Rsvd	-	保留	0
3	IOA_SEL3	rw	IOA3 特殊功能选择寄存器。 0: GPIO。 1: 特殊功能 1 (RTC_PLLDIV) 或者 2 (CF0)。	0x0

2: 0	Rsvd	-	保留	0

14.5.9 PMU_IOANODEG 寄存器

表202. PMU_IOANODEGL 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	
15: 0	IOANODEG	rw	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

14.5.10 IOX_OEN 寄存器

表203. IOX_OEN 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 0	IOXOEN	rw	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 196。 0: 使能 IO 输出模式。 1: 禁止 IO 输出模式。	0xFFFF

14.5.11 IOX_IE 寄存器

表204. IOX_IE 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 0	IOXIE	rw	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 196。 0: 禁止 IO 输入功能。 1: 使能 IO 输入功能。	0xFFFF

14.5.12 IOX_DAT 寄存器

表205. IOX_DAT 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 0	IOXDAT	rw	每个位控制 IOX 的输出数据和拉低/高功能，有关	0x0000

			IO 状态的详细信息，请参见表 196。	

14.5.13 IOX_ATT 寄存器

表206. IOX_ATT 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 0	IOXATT	rw	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

14.5.14 IOX_STS 寄存器

表207. IOX_STS 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	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。	--

14.5.15 IOB_SEL 寄存器

表208. IOB_SEL 寄存器

位	名称	类型	描述	默认值
31: 9	Rsvd	-	保留	0
8	IOB_SEL8	rw	IOB8 特殊功能选择寄存器。 0: GPIO 或特殊功能 1。 1: 特殊功能 2 (CF0 输出)。	0x0
7	Rsvd	-	保留	0
6	IOB_SEL6	rw	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	IOB_SEL5	rw	IOB5 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 (PLLL 经过分频器输出)。	0x0
4: 3	Rsvd	-	保留	0
2	IOB_SEL2	rw	IOB2 特殊功能选择寄存器。	0x0

			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 (PLLL 经过分频器输出)。	
1	IOB_SEL1	rw	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 经过分频器输出)。	0x0
0	Rsvd	-	保留	0

14.5.16 IOE_SEL 寄存器

表209. IOE_SEL 寄存器

位	名称	类型	描述	默认值
31: 13	Rsvd	-	保留	0
12	IOE_SEL12	rw	IOE12 特殊功能选择寄存器。 0: GPIO 或高阻。 1: 特殊功能 1 (CF1)。	0x0
11: 8	Rsvd	-	保留	0
7	IOE_SEL7	rw	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	Rsvd	-	保留	0

14.5.17 IO_MISC 寄存器

表210. IO_MISC 寄存器

位	名称	类型	描述	默认值
31: 6	Rsvd	-	保留	0
5	I2CIOC	rw	该寄存器用于控制 IOB 或 IOC 的 I ² C 功能。 0: I ² C 在 IOB13~IOB14。 1: I ² C 在 IOC4~IOC5。	0x0
4: 3	Rsvd	-	保留	0
2: 0	PLLHDIV	rw	当 IOB1 被选择为特殊功能 3 时、IOB2 被选择为特殊功能 3 或 IOB5 被选择为特殊功能 3 时，该寄存器用于控制 PLLH、PLLL 输出的分频比。 0: /1 1: /2 2: /4 3: /8 4: /16 其他: 保留。	0x0

14.6 特殊功能 IO

14.6.1 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211. 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 (When MODE is 0)		LCDSEG54	
IOA1	SWDIO (When MODE is 0)		LCDSEG53	
IOA2			LCDSEG52	
IOA3	RTC_PLLDIV	CF0	LCDSEG51	
IOA4			LCDSEG66	CMP2_P
IOA5			LCDSEG67	CMP2_N
IOA6	CMP2_O		LCDSEG68	
IOA7	RTC_PLLDIV		LCDSEG69	
IOA8			LCDSEG50	ADC_CH3
IOA9	CF1		LCDSEG49	ADC_CH4
IOA10			LCDSEG48	ADC_CH5
IOA11			LCDSEG47	ADC_CH6
IOA12	UARTRX0		LCDSEG71	
IOA13	UARTRX1	ISO7816_IO0	LCDSEG73	
IOA14	UARTRX2		LCDSEG13	
IOA15	UARTRX3	ISO7816_IO1	LCDSEG15	

14.6.2 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212. 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	UARTRX4	PWM0		LCDSEG43
IOB1	UARTRX5		PLLH divider	LCDSEG45
IOB2	UARTTX0		PLLL divider	LCDSEG70

IOB3	UARTTX1	ISO7816_CLK0		LCDSEG72
IOB4	UARTTX2			LCDSEG12
IOB5	UARTTX3	ISO7816_CLK1	PLLL divider	LCDSEG14
IOB6	UARTTX4	PWM1	RTCCLK	LCDSEG44
IOB7	UARTTX5			LCDSEG46
IOB8		CF0		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

14.6.3 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213. 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

14.6.4 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214. 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	SPI4CSN	
IOD13	LCDSEG9	SPI4CLK	
IOD14	LCDSEG10	SPI4MISO	
IOD15	LCDSEG11	SPI4MOSI	

14.6.5 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215. 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
IOE7	CMP1_O	LCDSEG60	ADC_CH11

IOE8		LCDSEG59	CMP1_P
IOE9		LCDSEG58	CMP1_N
IOE10		LCDSEG32	
IOE11		LCDSEG33	
IOE12	CF1	LCDSEG34	
IOE13		LCDSEG55	
IOE14		LCDSEG56	
IOE15		LCDSEG57	

14.6.6 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216. 特殊功能 IOF

IO	特殊功能 1
IOF0	X6_5MI
IOF1	X6_5MO

15 DMA 控制器

15.1 简介

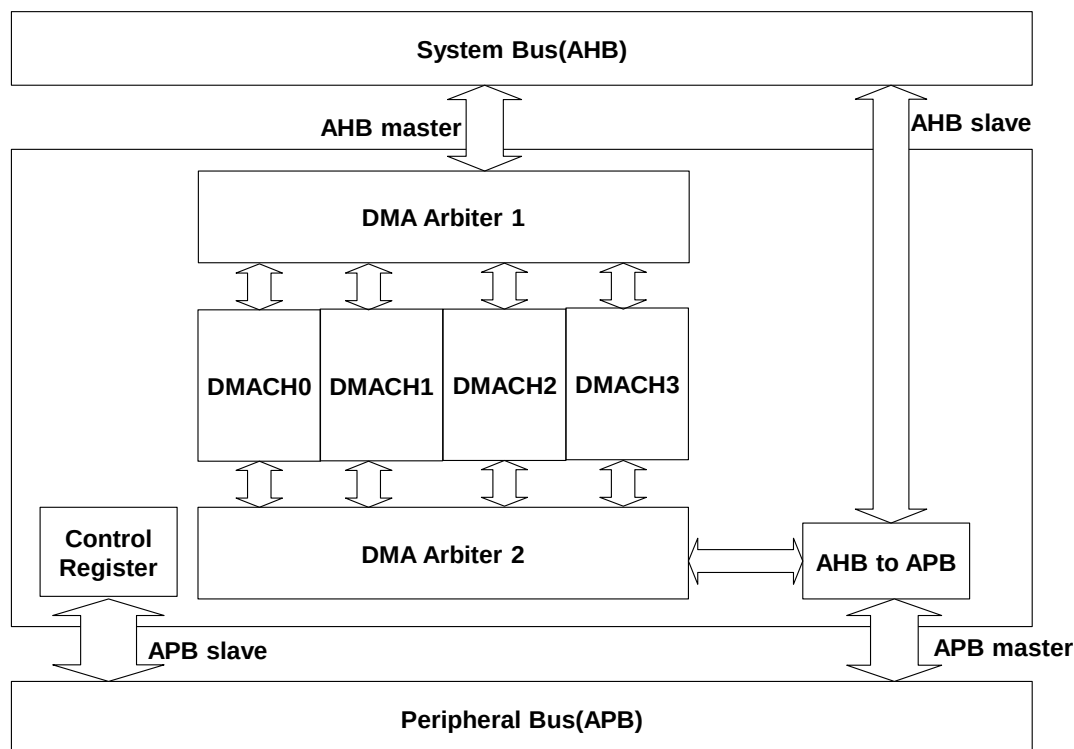
直接存储器存取（DMA）用于提供存储器到存储器或存储器到外设之间的高速数据传输。DMA 控制器集成了 AHB 到 APB 桥接器，因此当 DMA 进行低速外设数据传输时，高速外设总线（AHB）性能不会受到低速外设的影响。在 V85XX 中集成了 4 个 DMA 通道，4 个通道的优先级是循环的。当系统从睡眠唤醒时，DMA 控制器的设置会被复位，所以从这睡眠状态唤醒后应该手动恢复相关设置。

15.2 特点

- 支持存储器间的传输；
- 支持存储器到外设传输；
- 支持外设到存储器传输；
- 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。

15.3 功能框图

图26. DMA 控制器功能框图



15.4 寄存器地址

表217. DMA 控制器（DMA 基地址：0x40010000）

名称	类型	地址	描述	默认值
DMA_IE	rw	0x0000	DMA 中断使能寄存器	0x000
DMA_STS	rw	0x0004	DMA 状态寄存器	0x0000
DMA_C0CTL	rw	0x0010	DMA 通道 0 控制寄存器	0x00000000
DMA_C0SRC	rw	0x0014	DMA 通道 0 源寄存器	0x00000000
DMA_C0DST	rw	0x0018	DMA 通道 0 目的寄存器	0x00000000
DMA_C0LEN	r	0x001C	DMA 通道 0 传输长度寄存器	0x0000
DMA_C1CTL	rw	0x0020	DMA 通道 1 控制寄存器	0x00000000
DMA_C1SRC	rw	0x0024	DMA 通道 1 源寄存器	0x00000000
DMA_C1DST	rw	0x0028	DMA 通道 1 目标寄存器	0x00000000
DMA_C1LEN	r	0x002C	DMA 通道 1 传输长度寄存器	0x0000
DMA_C2CTL	rw	0x0030	DMA 通道 2 控制寄存器	0x00000000
DMA_C2SRC	rw	0x0034	DMA 通道 2 源寄存器	0x00000000
DMA_C2DST	rw	0x0038	DMA 通道 2 目标寄存器	0x00000000
DMA_C2LEN	r	0x003C	DMA 通道 2 传输长度寄存器。	0x0000
DMA_C3CTL	rw	0x0040	DMA 通道 3 控制寄存器	0x00000000
DMA_C3SRC	rw	0x0044	DMA 通道 3 源寄存器	0x00000000

DMA_C3DST	rw	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	rw	0x0050	DMAAES 控制寄存器	0x00000000
DMA_AESKEY0	rw	0x0060	DMAAES 密钥 0 寄存器	0x00000000
DMA_AESKEY1	rw	0x0064	DMAAES 密钥 1 寄存器	0x00000000
DMA_AESKEY2	rw	0x0068	DMAAES 密钥 2 寄存器	0x00000000
DMA_AESKEY3	rw	0x006C	DMAAES 密钥 3 寄存器	0x00000000
DMA_AESKEY4	rw	0x0070	DMAAES 密钥 4 寄存器	0x00000000
DMA_AESKEY5	rw	0x0074	DMAAES 密钥 5 寄存器	0x00000000
DMA_AESKEY6	rw	0x0078	DMAAES 密钥 6 寄存器	0x00000000
DMA_AESKEY7	rw	0x007C	DMAAES 密钥 7 寄存器	0x00000000

15.5 寄存器定义

15.5.1 DMA_IE 寄存器

表218. DMA_IE 寄存器

位	名称	类型	描述	默认值
31: 12	Rsvd	-	保留	0
11	C3DAIE	rw	通道 3 数据中止中断使能。 0: 禁用 1: 启用	0x0
10	C2DAIE	rw	通道 2 数据中止中断使能。 0: 禁用 1: 启用	0x0
9	C1DAIE	rw	通道 1 数据中止中断使能。 0: 禁用 1: 启用	0x0
8	C0DAIE	rw	通道 0 数据中止中断使能。 0: 禁用 1: 启用	0x0
7	C3FEIE	rw	通道 3 所有包传输完成中断使能。 0: 禁用 1: 启用	0x0
6	C2FEIE	rw	通道 2 所有包传输完成中断使能。 0: 禁用 1: 启用	0x0
5	C1FEIE	rw	通道 1 所有包传输完成中断使能。 0: 禁用 1: 启用	0x0
4	C0FEIE	rw	通道 0 所有包传输完成中断使能。	0x0

			0: 禁用 1: 启用	
3	C3PEIE	rw	通道 3 单个包传输完成中断使能。 0: 禁用 1: 启用	0x0
2	C2PEIE	rw	通道 2 单个包传输完成中断使能。 0: 禁用 1: 启用	0x0
1	C1PEIE	rw	通道 1 单个包传输完成中断使能。 0: 禁用 1: 启用	0x0
0	C0PEIE	rw	通道 0 单个包传输完成中断使能。 0: 禁用 1: 启用	0x0

15.5.2 DMA_STS 寄存器

表219. DMA_STS 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15	C3DA	rc_w1	通道 3 数据中止中断标志, 写 1 清除该标志。	0x0
14	C2DA	rc_w1	通道 2 数据中止中断标志, 写 1 清除该标志。	0x0
13	C1DA	rc_w1	通道 1 数据中止中断标志, 写 1 清除该标志。	0x0
12	C0DA	rc_w1	通道 0 数据中止中断标志, 写 1 清除该标志。	0x0
11	C3FE	rc_w1	通道 3 所有包传输完成中断标志, 写 1 清除该标志。	0x0
10	C2FE	rc_w1	通道 2 所有包传输完成中断标志, 写 1 清除该标志。	0x0
9	C1FE	rc_w1	通道 1 所有包传输完成中断标志, 写 1 清除该标志。	0x0
8	C0FE	rc_w1	通道 0 所有包传输完成中断标志, 写 1 清除该标志。	0x0
7	C3PE	rc_w1	通道 3 单个包传输完成中断标志, 写 1 清除该标志。	0x0
6	C2PE	rc_w1	通道 2 单个包传输完成中断标志, 写 1 清除该标志。	0x0
5	C1PE	rc_w1	通道 1 单个包传输完成中断标志, 写 1 清除该标志。	0x0
4	C0PE	rc_w1	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	0x0

			1: 工作	
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

15.5.3 DMA_CxCTL 寄存器

表 220 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220. DMA_CxCTL 寄存器

位	名称	类型	描述	默认值
31: 24	FLEN	rw	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	rw	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	rw	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	rw	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 应配置为 4 的整数倍。 0: 禁止 1: 使能	0x0
13	CONT	rw	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	rw	传送模式选择寄存器。	0x0

			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。	
11: 7	DMASEL	rw	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	rw	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	rw	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	rw	传输长度模式。 0: 字节 (8 位) 1: 半字 (16 位) 2: 全字 (32 位) 3: 保留。	0x0
0	EN	rw	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221. DMA 请求源选择

DMASEL	请求源
0	Software
1	-
2	UART0 TX
3	UART0 RX
4	UART1 TX
5	UART1 RX
6	UART2 TX
7	UART2 RX
8	UART3 TX
9	UART3 RX
10	UART4 TX

11	UART4 RX
12	UART5 TX
13	UART5 RX
14	ISO78160 TX
15	ISO78160 RX
16	ISO78161 TX
17	ISO78161 RX
18	TIMER 0
19	TIMER 1
20	TIMER 2
21	TIMER 3
22	SPI1 TX
23	SPI1 RX
24	UART 32K 0
25	UART 32K 1
26	CMP1
27	CMP2
28	SPI3 TX
29	SPI3 RX
30	SPI2 TX
31	SPI2 RX

15.5.4 DMA_CxSRC 寄存器

表222. DMA_CxSRC 寄存器

位	名称	类型	描述	默认值
31: 0	SRC	rw	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，该寄存器必须全字对齐。当 SRC 设置为 0x4001xxxx 时，它将自动切换到 IO 读取。	0x00000000

15.5.5 DMA_CxDST 寄存器

表223. DMA_CxDST 寄存器

位	名称	类型	描述	默认值
31: 0	DST	rw	DMA 目标地址寄存器。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，该寄存器必须全字对齐。当 DST 设置为 0x4001xxxx 时，它将自动切换到 IO 写入。	0x00000000

15.5.6 DMA_CxLEN 寄存器

表224. DMA_CxLEN 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	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 完成所有包传输完成后自动清零。	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 完成所有包传输完成后自动清零。	0x00

15.5.7 DMA_AESCTL 寄存器

表225. DMA_AESCTL 寄存器

位	名称	类型	描述	默认值
31: 4	Rsvd	-	保留	0
3: 2	MODE	rw	AES 模式选择寄存器。 0: AES128 1: AES192 2: AES256 3: 保留。	0x0
1	Rsvd	-	保留	0
0	ENC	rw	AES 编码/解码选择寄存器。 0: 解码 1: 编码	0x0

15.5.8 DMA_AESKEYx 寄存器

表226. DMA_AESKEYx 寄存器

位	名称	类型	描述	默认值
31: 0	KEYx	rw	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	0x00000000

			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。	

16 UART 控制器

16.1 简介

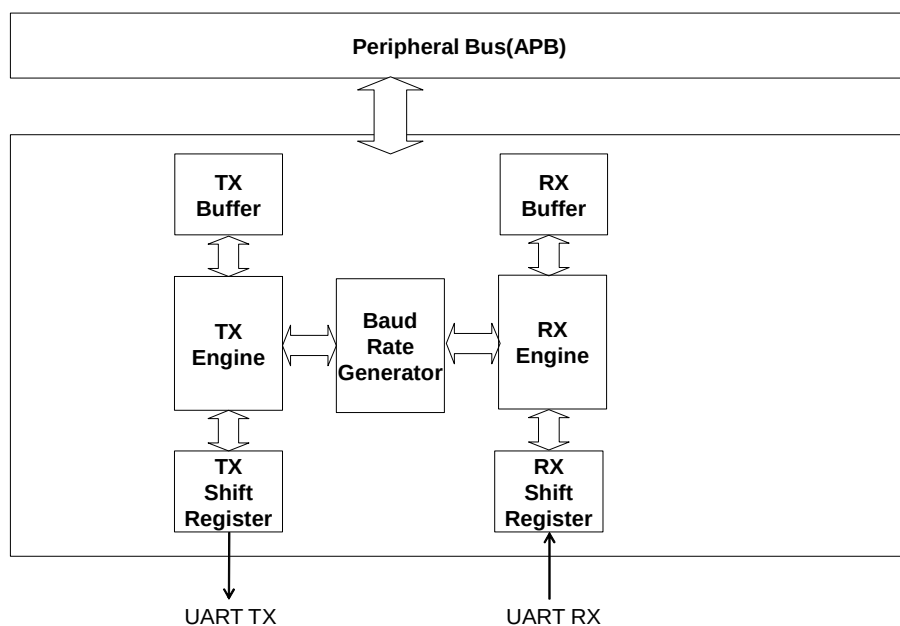
UART 控制器用于通过 UART 协议发送/接收数据。V85XX 最多有 6 个 UART 控制器，每个 UART 控制器可以单独编程，并有自己的 CPU 中断。UART 控制器可支持无奇偶校验的 8 位数据或有奇偶校验的 9 位数据传输。发送机和接收机都有一个数据缓存器和一个移位寄存器。UART 控制器中的设置将在从睡眠模式唤醒后复位，用户应在从睡眠状态唤醒后手动恢复设置。UART 的波特率支持范围 300~819200bps。

16.2 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。

16.3 功能框图

图27. UART 控制器的功能框图



16.4 寄存器地址

表227. UART 控制器 (UART 基地址: 0x40011800)

名称	类型	地址	描述	默认值
UART0_DATA	rw	0x0000	UART0 数据寄存器	0x00
UART0_STATE	rc_w1	0x0004	UART0 状态寄存器	0x00
UART0_CTRL	rw	0x0008	UART0 控制寄存器	0x000
UART0_INTSTS	rc_w1	0x000C	UART0 中断状态寄存器	0x00
UART0_BAUDDIV	rw	0x0010	UART0 波特率分频寄存器	0x00000
UART0_CTRL2	rw	0x0014	UART0 控制寄存器 2	0x0
UART1_DATA	rw	0x0020	UART1 数据寄存器	0x00
UART1_STATE	rc_w1	0x0024	UART1 状态寄存器	0x00
UART1_CTRL	rw	0x0028	UART1 控制寄存器	0x000
UART1_INTSTS	rc_w1	0x002C	UART1 中断状态寄存器	0x00
UART1_BAUDDIV	rw	0x0030	UART1 波特率分频寄存器	0x00000
UART1_CTRL2	rw	0x0034	UART1 控制寄存器 2	0x0
UART2_DATA	rw	0x0040	UART2 数据寄存器	0x00
UART2_STATE	rc_w1	0x0044	UART2 状态寄存器	0x00
UART2_CTRL	rw	0x0048	UART2 控制寄存器	0x000
UART2_INTSTS	rc_w1	0x004C	UART2 中断状态寄存器	0x00
UART2_BAUDDIV	rw	0x0050	UART2 波特率分频寄存器	0x00000
UART2_CTRL2	rw	0x0054	UART2 控制寄存器 2	0x0
UART3_DATA	rw	0x0060	UART3 数据寄存器	0x00
UART3_STATE	rc_w1	0x0064	UART3 状态寄存器	0x00
UART3_CTRL	rw	0x0068	UART3 控制寄存器	0x000
UART3_INTSTS	rc_w1	0x006C	UART3 中断状态寄存器	0x00
UART3_BAUDDIV	rw	0x0070	UART3 波特率分频寄存器	0x00000
UART3_CTRL2	rw	0x0074	UART3 控制寄存器 2	0x0
UART4_DATA	rw	0x0080	UART4 数据寄存器	0x00
UART4_STATE	rc_w1	0x0084	UART4 状态寄存器	0x00
UART4_CTRL	rw	0x0088	UART4 控制寄存器	0x000
UART4_INTSTS	rc_w1	0x008C	UART4 中断状态寄存器	0x00
UART4_BAUDDIV	rw	0x0090	UART4 波特率分频寄存器	0x00000
UART4_CTRL2	rw	0x0094	UART4 控制寄存器 2	0x0
UART5_DATA	rw	0x00A0	UART5 数据寄存器	0x00
UART5_STATE	rc_w1	0x00A4	UART5 的状态寄存器	0x00
UART5_CTRL	rw	0x00A8	UART5 控制寄存器	0x000
UART5_INTSTS	rc_w1	0x00AC	UART5 中断状态寄存器	0x00
UART5_BAUDDIV	rw	0x00B0	UART5 波特率分频寄存器	0x00000
UART5_CTRL2	rw	0x00B4	UART5 控制寄存器 2	0x0

表228. MISC 控制器（MISC 基地址：0x40013000）

名称	类型	地址	描述	默认值
MISC_IREN	rw	0x000C	IR 使能控制寄存器	0x00
MISC_DUTYL	rw	0x0010	IR 调制波低电平宽度控制寄存器	0x0000
MISC_DUTYH	rw	0x0014	IR 调制波高电平宽度控制寄存器	0x0000

16.5 寄存器定义

16.5.1 UARTx_DATA 寄存器

表229. UARTx_数据寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7: 0	DATA	rw	读：接收数据。 写：发送数据。	0x0

16.5.2 UARTx_STATE 寄存器

表230. UARTx_STATE 寄存器

位	名称	类型	描述	默认值
31: 7	Rsvd	-	保留	0
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	rc_w1	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	rc_w1	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	rc_w1	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	rc_w1	发送缓存区溢出标志。当 TXFULL 为 1 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。	0x0

			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。	
0	Rsvd	-	保留	0

16.5.3 UARTx_CTRL 寄存器

表231. UARTx_CTRL 寄存器

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

16.5.4 UARTx_INTSTS 寄存器

表232. UARTx_INTSTS 寄存器

位	名称	类型	描述	默认值
31: 6	Rsvd	-	保留	0
5	TXDONEIF	rc_w1	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	rc_w1	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	rc_w1	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	rc_w1	发送缓存区溢出标志。当 TXOVIE 为 1、TXFULL 为 1 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	0x0

			寄存器中的 TXOV 位写 1，该位将被清零。	
1	RXIF	rc_w1	接收中断标志，当 UARTx_DATA 寄存器接收到数据时，在数据结束位之前该位置 1。向该位写 1 即可清零。	0x0
0	Rsvd	-	保留。	0

16.5.5 UARTx_BAUDDIV 寄存器

表233. UARTx_BAUDDIV 寄存器

位	名称	类型	描述	默认值
31: 20	Rsvd	-	保留	0
19: 0	BAUDDIV	rw	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。 例如当 APBCLK=6.5536MHz，波特率是 9600bps 时，该寄存器值等于 6553600/9600=682。	0x0

16.5.6 UARTx_CTRL2 寄存器

表234. UARTx_CTRL2 寄存器

位	名称	类型	描述	默认值
31: 4	Rsvd	-	保留	0
3: 2	PMODE	rw	奇偶校验模式选择寄存器。该寄存器仅在 MODE 设置为 1 时有效。 0: 偶校验。 1: 奇校验。 2: 奇偶校验位始终为 0。 3: 奇偶校验位始终为 1。	0x0
1	MODE	rw	UART 数据模式控制寄存器。 0: 无校验，1+8+1 模式。 1: 奇偶校验，1+8+1+1 模式，奇偶校验位由 PMODE 寄存器控制。	0x0
0	MSB	rw	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

16.5.7 MISC_IREN 寄存器

表235. MISC_IREN 寄存器

位	名称	类型	描述	默认值
31: 6	Rsvd	-	保留	0

5: 0	IREN	rw	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

16.5.8 MISC_DUTYL 寄存器

表236. MISC_DUTYL 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 0	DUTYL	rw	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

16.5.9 MISC_DUTYH 寄存器

表237. MISC_DUTYH 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 0	DUTYH	rw	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

17 UART32K 控制器

17.1 简介

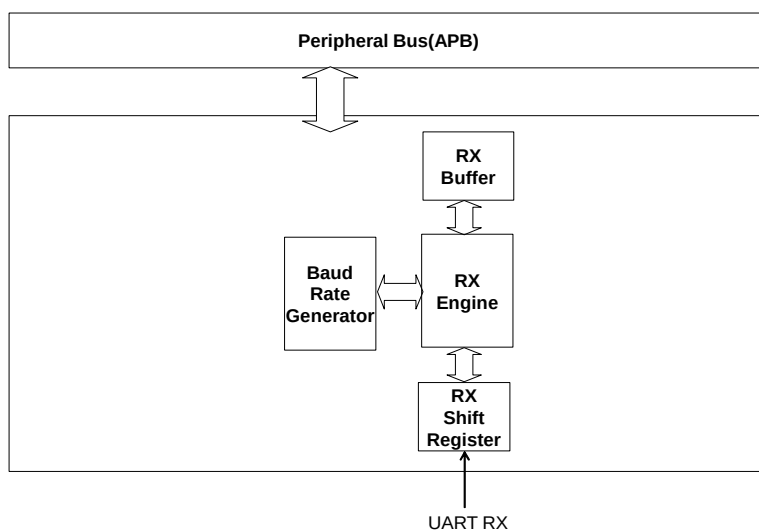
UART32K 控制器使用 UART 协议接收数据。V85XX 中有两个独立的 UART32K 控制器，其输入引脚从普通串口 UART0~UART3 的输入脚（rx0~rx3）中选择。UART32K 控制器的工作频率是 RTCCLK，因此可以在睡眠模式下工作。UART32K 支持的典型波特率为 9600，支持数据长度为 8 位，支持无奇偶校验的 8 位数据和有奇偶校验的 9 位数据传输，接收机带一个数据缓冲器和一个移位寄存器。UART32K 的波特率支持范围 300~14400bps。

17.2 特点

- 可在睡眠模式下工作。
- 具有睡眠唤醒功能。
- 可编程波特率发生器。
- 支持 8 位数据传输。
- 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。

17.3 功能框图

图28. UART32K 控制器的功能框图



17.4 寄存器地址

表238. UART32K 控制器（UART32K0 基地址：0x40014100）

名称	类型	地址	描述	默认值
U32K0_CTRL0	rw	0x0000	UART32K0 控制寄存器 0	0x000
U32K0_CTRL1	rw	0x0004	UART32K0 控制寄存器 1	0x00
U32K0_PHASE	rw	0x0008	UART32K0 波特率控制寄存器	0x4B00
U32K0_DATA	r	0x000C	UART32K0 接收数据缓存区	--
U32K0_STS	rc_w1	0x0010	UART32K0 中断状态寄存器	0x0

表239. UART32K 控制器（UART32K1 基地址：0x40014180）

名称	类型	地址	描述	默认值
U32K1_CTRL0	rw	0x0000	UART32K1 控制寄存器 0	0x00
U32K1_CTRL1	rw	0x0004	UART32K1 控制寄存器 1	0x00
U32K1_PHASE	rw	0x0008	UART32K1 波特率控制寄存器	0x4B00
U32K1_DATA	r	0x000C	UART32K1 接收数据缓存区	--
U32K1_STS	rc_w1	0x0010	UART32K1 中断状态寄存器	0x0

17.5 寄存器定义

17.5.1 U32Kx_CTRL0 寄存器

表240. U32Kx_CTRL0 寄存器

位	名称	类型	描述	默认值
31: 9	Rsvd	-	保留	0
8	WKUMODE	rw	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	rw	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: 4	PMODE	rw	奇偶校验模式控制，仅在 MODE 设置为 1 时有效。 0: 偶校验。	0x0

			1: 奇校验。 2: 奇偶校验位始终为 0。 3: 奇偶校验位始终为 1。	
3	MODE	rw	UART 数据模式控制寄存器。 0: 无校验。1+8+1 模式。 1: 有奇偶校验。1+8+1+1 模式, 奇偶校验位由 PMODE 控制。	0x0
2	MSB	rw	数据位接收顺序控制。 0: 低位先收 (LSB)。 1: 高位先收 (MSB)。	0x0
1	Rsvd	-	保留	0
0	EN	rw	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

17.5.2 U32Kx_CTRL1 寄存器

表241. U32Kx_CTRL1 寄存器

位	名称	类型	描述	默认值
31: 6	Rsvd	-	保留	0
5: 4	RXSEL	rw	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	Rsvd	-	保留	0
2	RXOVIE	rw	接收溢出中断/唤醒使能。	0x0
1	RXPEIE	rw	接收奇偶校验错误中断/唤醒使能。	0x0
0	RXIE	rw	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

17.5.3 U32Kx_PHASE 寄存器

表242. U32Kx_PHASE 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0

15: 0	PHASE	rw	波特率分频寄存器，在使能 UART32K 之前必须设置这个寄存器。 PHASE=65536*Baud-rate/RTCCLK。 例如，当波特率是 9600bps，RTCCLK 频率为 32768，65536*9600/32768=19200。UART32K 支持的最大波特率是 9600bps。 当 RTC_PSCA 寄存器的 PSCA=1，PHASE 应该使用 RTCCLK 的分频后的频率进行计算。	0x4B00

17.5.4 U32Kx_DATA 寄存器

表243. U32Kx_数据寄存器

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

17.5.5 U32Kx_STS 寄存器

表244. U32Kx_STS 寄存器

位	名称	类型	描述	默认值
31: 3	Rsvd	-	保留	0
2	RXOV	rc_w1	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。向该位写 1 可将标志位清零。	0x0
1	RXPE	rc_w1	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。向该位写 1 可将标志位清零。	0x0
0	RXIF	rc_w1	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。向该位写 1 可将标志位清零。	0x0

18 ISO7816 控制器

18.1 简介

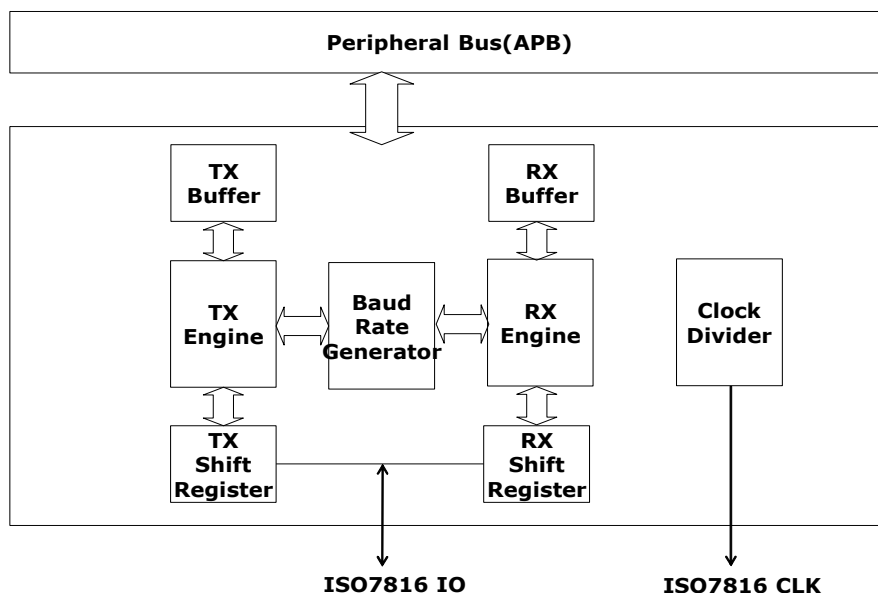
ISO7816 是一种增强型 UART 协议，能够在 2 线总线上进行半双工通信。V85XX 内部集成 2 个可独立编程的 ISO7816 控制器，并拥有独立的 CPU 中断。IP 中内置一个时钟分频器，能够为 ISO7816 器件提供 1~5MHz 的时钟。从睡眠模式唤醒后，ISO7816 控制器的设置会被复位，所以，在程序唤醒后应该手动恢复相关的设置。ISO7816 波特率支持范围：200~2625000bps。

18.2 特点

- 可编程波特率发生器
- 支持半双工发送/接收。
- 支持时钟分频。

18.3 功能框图

图29. ISO7816 控制器功能框图



18.4 寄存器地址

表245. ISO7816 控制器地址（ISO7816 基地址：0x40012000）

名称	类型	地址	描述	默认值
ISO78160_BAUDDIVL	rw	0x0004	ISO78160 波特率低字节寄存器	0x00
ISO78160_BAUDDIVH	rw	0x0008	ISO78160 波特率高字节寄存器	0x00
ISO78160_DATA	rw	0x000C	ISO78160 数据寄存器	0x00
ISO78160_INFO	rc_w1	0x0010	ISO78160 信息寄存器	0x00
ISO78160_CFG	rw	0x0014	ISO78160 控制寄存器	0x00
ISO78160_CLK	rw	0x0018	ISO78160 时钟分频器控制寄存器	0x00
ISO78161_BAUDDIVL	rw	0x0044	ISO78161 波特率低字节寄存器	0x00
ISO78161_BAUDDIVH	rw	0x0048	ISO78161 波特率高字节寄存器	0x00
ISO78161_DATA	rw	0x004C	ISO78161 数据寄存器	0x00
ISO78161_INFO	rc_w1	0x0050	ISO78161 信息寄存器	0x00
ISO78161_CFG	rw	0x0054	ISO78161 控制寄存器	0x00
ISO78161_CLK	rw	0x0058	ISO78161 时钟分频器控制寄存器	0x00

18.5 寄存器定义

18.5.1 ISO7816x_BAUDDIVL 寄存器

表246. ISO7816x_BAUDDIVL 寄存器

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

18.5.2 ISO7816x_BAUDDIVH 寄存器

表247. ISO7816x_BAUDDIVH 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7: 0	BAUDDIVH	rw	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： BAUDDIV=0x10000-（APBCLK/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

18.5.3 ISO7816x_DATA 寄存器

表248. ISO7816x_数据寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7: 0	DATA	rw	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

18.5.4 ISO7816x_INFO 寄存器

表249. ISO7816x_INFO 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7	OVIF	rc_w1	接收溢出标志。当已写入数据未被 CPU 读取时，接收到另一个数据，该位将置 1。向该位写 1 可将标志位清零。	0x0
6	SDIF	rc_w1	发送中断标志。当成功发送一字节数据并收到应答（ACK），该位将置 1。这里，不关心接收应答是 0 还是 1。向该位写 1 可将标志位清零。	0x0
5	RCIF	rc_w1	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。向该位写 1 可将标志位清零。	0x0
4	LSB	rw	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
3	SDERR	rc_w1	发送模式下接收到的应答（ACK）为 0 时，该位将被置 1。向该位写 1 可将标志位清零。	0x0
2	RCERR	rc_w1	接收数据有校验和错误时，该位将被置为 1。向该位写 1 可将标志位清零。	0x0
1	CHKSUM	rw	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	RCACK	rw	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

18.5.5 ISO7816x_CFG 寄存器

表250. ISO7816x_CFG 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7	OVIE	rw	接收溢出中断使能寄存器。	0x0
6	SDIE	rw	发送中断使能寄存器。	0x0
5	RCIE	rw	接收中断使能寄存器。	0x0

4	ACKLEN	rw	接收错误数据时 ACK 低电平长度。 0: 1 位 1: 2 位	0x0
3: 2	Rsvd	-	保留	0x0
1	CHKP	rw	奇偶校验模式控制寄存器。 0: 偶校验 1: 奇校验。	0x0
0	EN	rw	ISO7816 使能寄存器。 0: 禁用 ISO7816 功能。 1: 使能 ISO7816 功能。	0x0

18.5.6 ISO7816x_CLK 寄存器

表251. ISO7816x_CLK 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7	CLKEN	rw	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	rw	ISO7816 时钟分频值。 0: APBCLK/1 1: APBCLK/2 2: APBCLK/3 ... 127: APBCLK/128	0x0

下表显示了 ISO7816 的数据包格式。

表252. ISO7816 数据包格式

位	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 位	Data	Parity 位	ACK*		Wait	Start 位	...
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。

发送包含总共 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。

19 Timer/PWM 控制器

19.1 简介

V85XX 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 比较功能。每个定时器可以产生 CPU 中断。每个 PWM 可以输出具有不同输出波形的 3 个输出。定时器/PWM 控制器中的设置将在从睡眠模式唤醒后复位，用户应在从这睡眠状态唤醒后手动恢复设置。

19.2 特点

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

19.3 功能框图

图30. 定时器功能框图

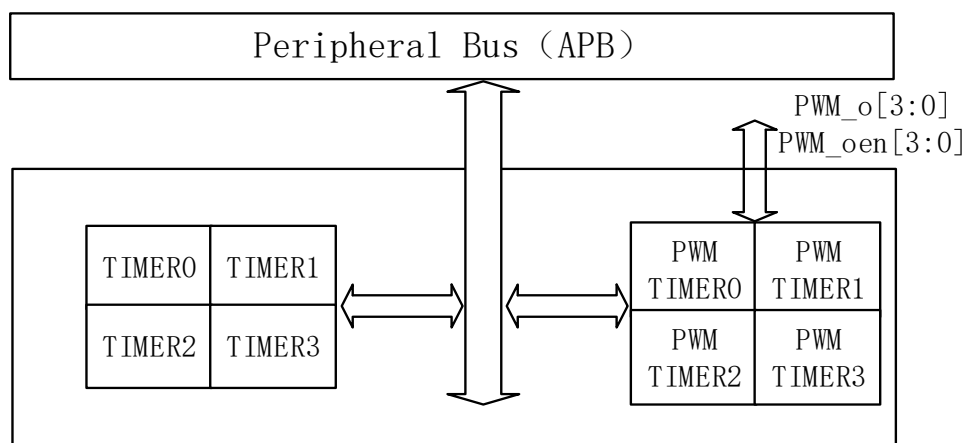
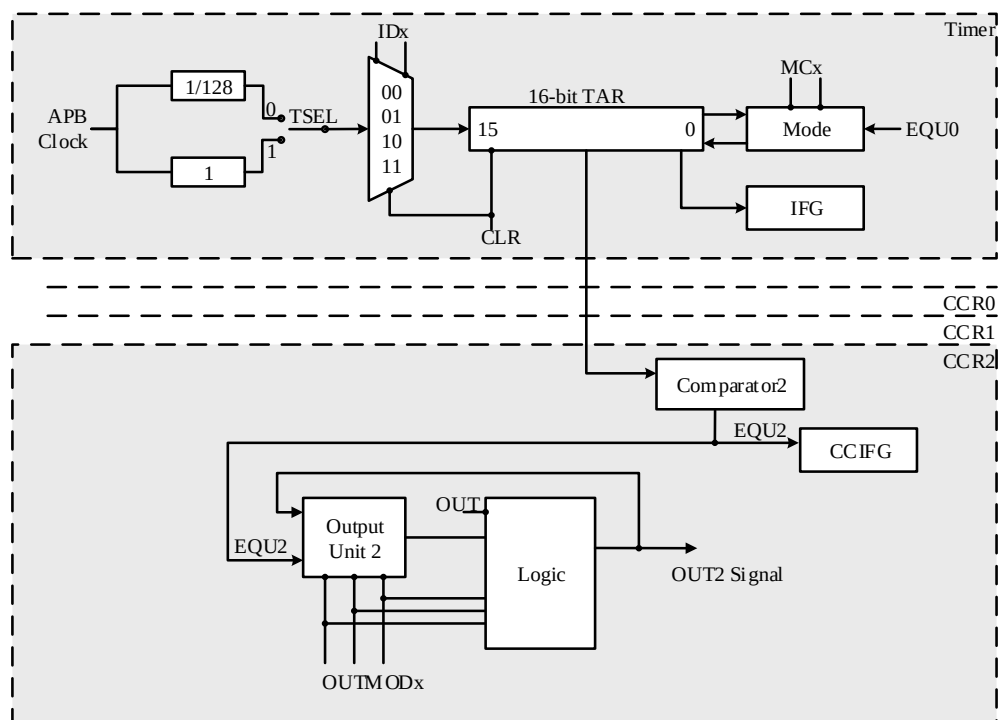


图31. PWM 定时器功能框图



19.4 寄存器地址

表253. 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	rw	0x0000	Timer0 控制寄存器	0x0
TMR0_VALUE	rw	0x0004	Timer0 实时计数寄存器	0x00000000
TMR0_RELOAD	rw	0x0008	Timer0 重载寄存器	0x00000000
TMR0_INT	rc_w1	0x000C	Timer0 中断状态寄存器	0x0
TMR1_CTRL	rw	0x0020	Timer1 控制寄存器	0x0
TMR1_VALUE	rw	0x0024	Timer1 实时计数寄存器	0x00000000
TMR1_RELOAD	rw	0x0028	Timer1 重载寄存器	0x00000000
TMR1_INT	rc_w1	0x002C	Timer1 中断状态寄存器	0x0
TMR2_CTRL	rw	0x0040	Timer2 控制寄存器	0x0
TMR2_VALUE	rw	0x0044	Timer2 实时计数寄存器	0x00000000
TMR2_RELOAD	rw	0x0048	Timer2 重载寄存器	0x00000000
TMR2_INT	rc_w1	0x004C	Timer2 中断状态寄存器	0x0
TMR3_CTRL	rw	0x0060	Timer3 控制寄存器	0x0
TMR3_VALUE	rw	0x0064	Timer3 实时计数寄存器	0x00000000
TMR3_RELOAD	rw	0x0068	Timer3 重载寄存器	0x00000000

TMR3_INT	rc_w1	0x006C	Timer3 中断状态寄存器	0x0

表254. 16 位 PWM TIMER 控制器（16bPWM TIMER 基地址：0x40012900）

名称	类型	地址	描述	默认值
PWM0_CTL	rw	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	rw	0x0008	PWM Timer0 比较寄存器 0	0x0000
PWM0_CCTL1	rw	0x000C	PWM Timer0 比较寄存器 1	0x0000
PWM0_CCTL2	rw	0x0010	PWM Timer0 比较寄存器 2	0x0000
PWM0_CCR0	rw	0x0014	PWM Timer0 比较数据寄存器 0	0x0000
PWM0_CCR1	rw	0x0018	PWM Timer0 比较数据寄存器 1	0x0000
PWM0_CCR2	rw	0x001C	PWM Timer0 比较数据寄存器 2	0x0000
PWM1_CTL	rw	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	rw	0x0028	PWM Timer1 比较数据寄存器 0	0x0000
PWM1_CCTL1	rw	0x002C	PWM Timer1 比较数据寄存器 1	0x0000
PWM1_CCTL2	rw	0x0030	PWM Timer1 比较数据寄存器 2	0x0000
PWM1_CCR0	rw	0x0034	PWM Timer1 比较数据寄存器 0	0x0000
PWM1_CCR1	rw	0x0038	PWM Timer1 比较数据寄存器 1	0x0000
PWM1_CCR2	rw	0x003C	PWM Timer1 比较数据寄存器 2	0x0000
PWM2_CTL	rw	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	rw	0x0048	PWM Timer2 比较控制寄存器 0	0x0000
PWM2_CCTL1	rw	0x004C	PWM Timer2 比较控制寄存器 1	0x0000
PWM2_CCTL2	rw	0x0050	PWM Timer2 比较控制寄存器 2	0x0000
PWM2_CCR0	rw	0x0054	PWM Timer2 比较数据寄存器 0	0x0000
PWM2_CCR1	rw	0x0058	PWM Timer2 比较数据寄存器 1	0x0000
PWM2_CCR2	rw	0x005C	PWM Timer2 比较数据寄存器 2	0x0000
PWM3_CTL	rw	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	rw	0x0068	PWM Timer3 比较控制寄存器 0	0x0000
PWM3_CCTL1	rw	0x006C	PWM Timer3 比较控制寄存器 1	0x0000
PWM3_CCTL2	rw	0x0070	PWM Timer3 比较控制寄存器 2	0x0000
PWM3_CCR0	rw	0x0074	PWM Timer3 比较数据寄存器 0	0x0000
PWM3_CCR1	rw	0x0078	PWM Timer3 比较数据寄存器 1	0x0000
PWM3_CCR2	rw	0x007C	PWM Timer3 比较数据寄存器 2	0x0000
PWM_O_SEL	rw	0x00F0	PWM 输出选择寄存器	0xDB51

19.5 寄存器定义

19.5.1 TMRx_CTRL 寄存器

表255. TMRx_CTRL 寄存器

位	名称	类型	描述	默认值
31: 4	Rsvd	-	保留	0
3	INTEN	rw	Timerx 中断使能寄存器。	0x0
2	EXTCLK	rw	选择外部时钟信号 <code>ext_clk</code> (IOB15) *作为时钟源。 <code>Ext_clk</code> 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 <code>ext_clk</code> 上升沿, 内部计数器减 1。该模式下, 外部时钟信号 <code>ext_clk</code> 的频率应低于 $PCLK/6$ 。	0x0
1	EXTEN	rw	选择 <code>ext_clk</code> (IOB15) 管脚作为时钟使能 IO 口。当 <code>EXTEN</code> 为 1, 并且 <code>ext_clk</code> 为 1 时, 内部计数器将以 $PCLK$ 的时钟速率递减 1。当 <code>EXTEN</code> 为 1, 并且 <code>ext_clk</code> 为 0, 内部计数器不会变化。 <code>Ext_clk</code> 会有两个周期的同步时钟, 以避免 <code>ext_clk</code> 上的毛刺。当 <code>EXTCLK</code> 和 <code>EXTEN</code> 同时使能, <code>EXTCLK</code> 优先级更高。	0x0
0	EN	rw	Timerx 使能控制寄存器。	0x0

注1: (*) `ext_clk` (IOB15) 是名为 `TIMEREXT_CLK` (IOB15) 的引脚

19.5.2 TMRx_VALUE 寄存器

表256. TMRx_VALUE 寄存器

位	名称	类型	描述	默认值
31: 0	VALUE	rw	Timerx 实时值寄存器。	0x00000000

19.5.3 TMRx_RELOAD 寄存器

表257. TMRx_RELOAD 寄存器

位	名称	类型	描述	默认值
31: 0	RELOAD	rw	Timerx 重载值寄存器。对该寄存器写值即设置当前值。每当定时器向下计数到 0, <code>RELOAD</code> 的值将被重载到计数器 <code>TMRx_VALUE</code> 中, 并重新开始向下计数。同时, <code>INT</code> 位将被置 1。 定时器 <code>RELOAD</code> 计算公式如下: $RELOAD = Period * APBCLK - 1$ 。	0x00000000

19.5.4 TMRx_INT 寄存器

表258. TMRx_INT 寄存器



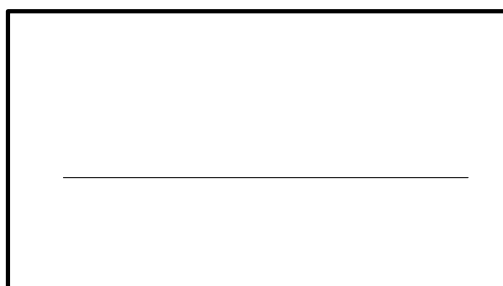
位	名称	类型	描述	默认值
31: 1	Rsvd	-	保留	0
0	INT	rc_w1	Timerx 中断状态寄存器，写 1 清零。当内部向下计数器计数到 0 时，该位被置 1。	0x0

19.5.5 PWMx_CTL 寄存器

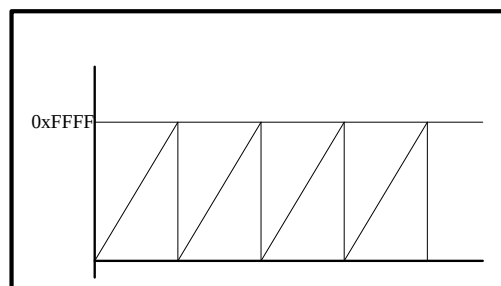
表259. PWMx_CTL 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7: 6	ID	rw	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	rw	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	rw	时钟源选择。 0: APB 时钟/128。 1: APB 时钟。	0x0
2	CLR	rw	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	rw	PWM Timerx 中断使能寄存器。	0x0
0	IFG	rc_w1	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

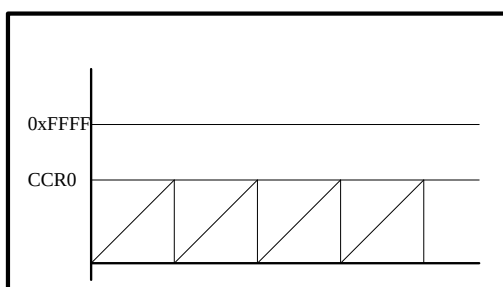
图32. PWM 定时器模式



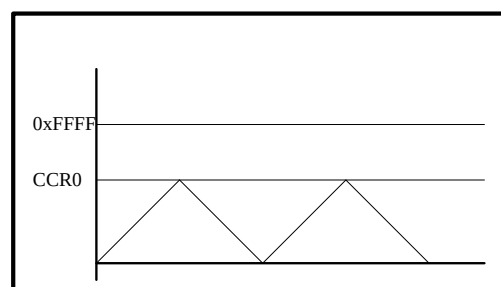
Stop Mode
Timer stop.



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.

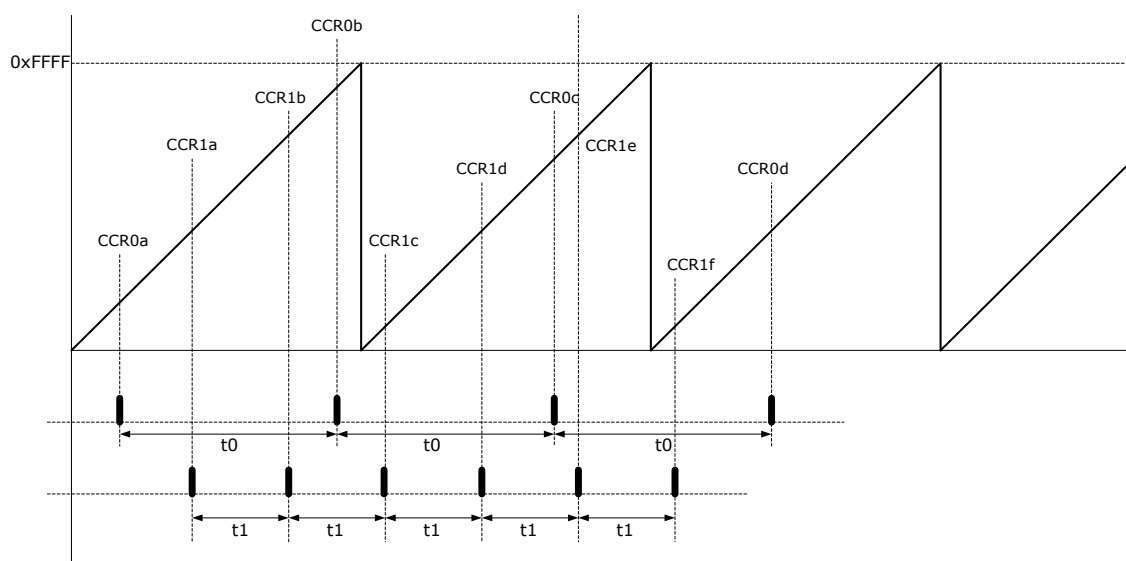
MC 设置为非零值时，定时器将开始工作。PWM Timer 工作模式为向上计数模式和向上/向下计数模式时，若设置 PWMx_CCR0 为 0，定时器 x 将停止计数。如果重新向 PWMx_CCR0 写入非零值，定时器将重新开始计数。

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33. 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，用户可以使用 CCIFG（CCTLx 的位 0）和 CCIE（CCTLx 的位 4），并设置 CCR0a 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。

19.5.6 PWMx_TAR 寄存器

表260. PWMx_TAR 寄存器

位	名称	类型	描述	默认值
31: 0	TAR	rw	PWM Timerx 实时计数寄存器	0x00000000

19.5.7 PWMx_CCTLx 寄存器

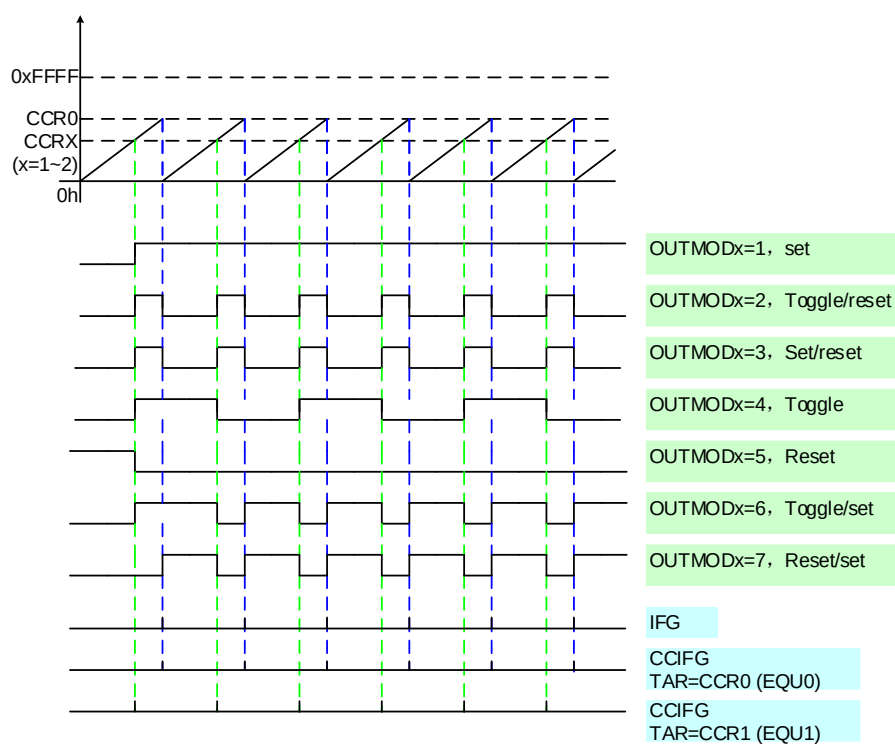
表261. PWMx_CCTLx 寄存器

位	名称	类型	描述	默认值
31: 10	Rsvd	-	保留	0
9	OUTEN	rw	输出 OUTx 使能控制寄存器。 0: OUTx 输出禁止。 1: OUTx 输出使能。	0x0
8	Rsvd	-	保留	0
7: 5	OUTMOD	rw	输出模式选择。 0: 常数模式，OUTx 输出 OUT 位的值。 1: 置位模式，当 $TAR = CCRx$ ($x=0\sim2$) 时，OUTx	0x0

			将置 1。 2: 翻转/复位模式, 当 $TAR=CCR_x$ ($x=1\sim2$) 时, OUT_x 翻转, 当 $TAR=CCR_0$ 时, OUT_x 复位为 0。此模式不能用于 OUT_0。 3: 置位/复位模式, 当 $TAR=CCR_x$ ($x=1\sim2$) 时, OUT_x 置 1, 当 $TAR=CCR_0$ 时, OUT_x 复位为 0。此模式不能用于 OUT_0。 4: 翻转模式, 当 $TAR=CCR_x$ ($x=1\sim2$) 时, OUT_x 将会被翻转。 5: 复位模式, 当 $TAR=CCR_x$ ($x=1\sim2$) 时, OUT_x 复位为 0。 6: 翻转/置位模式, 当 $TAR=CCR_x$ ($x=1\sim2$) 时, OUT_x 被翻转, 当 $TAR=CCR_0$ 时, OUT_x 置为 1。此模式不能用于 OUT_0。 7: 复位/置位模式, 当 $TAR=CCR_x$ ($x=1\sim2$) 时, OUT_x 复位为 0, 当 $TAR=CCR_0$ 时, OUT_x 将置为 1。此模式不能用于 OUT_0。	
4	CCIE	rw	比较中断使能寄存器。	0x0
3	Rsvd	-	保留	0
2	OUT	rw	当 $OUTMOD$ 设置为 0 时, 该位用于控制 OUT_x 的输出值。	0x0
1	Rsvd	-	保留	0
0	CCIFG	rw	比较模式下, 当 $TAR=CCR_x$ 时, 该位将被置 1。向该位写 1 可将标志位清零。	0

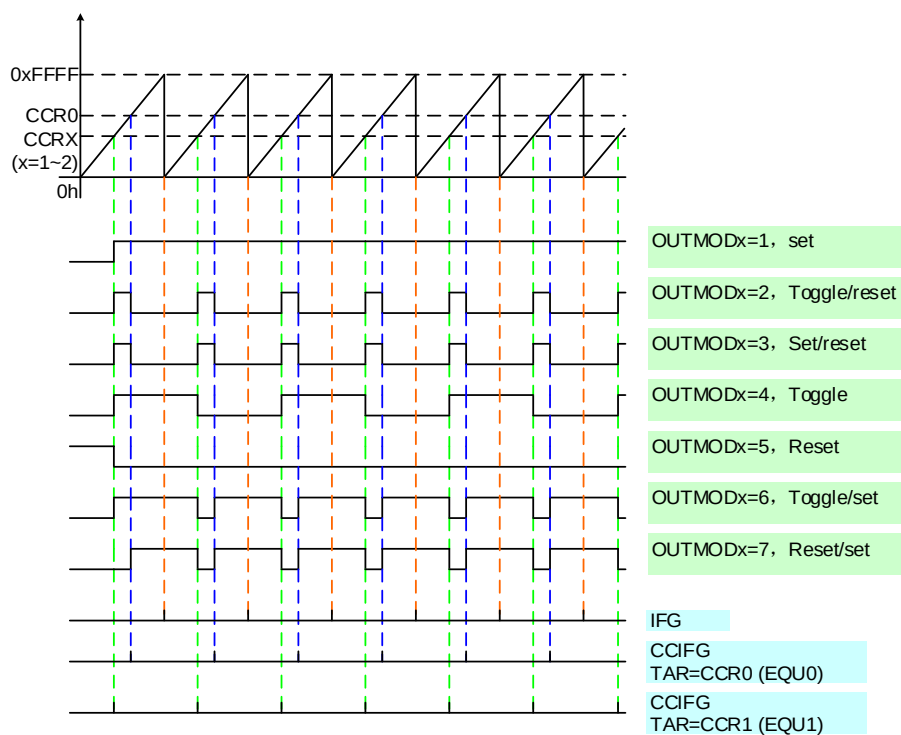
下图是在向上计数模式下的 PWM 输出:

图34. 向上计数模式下的 PWM 输出



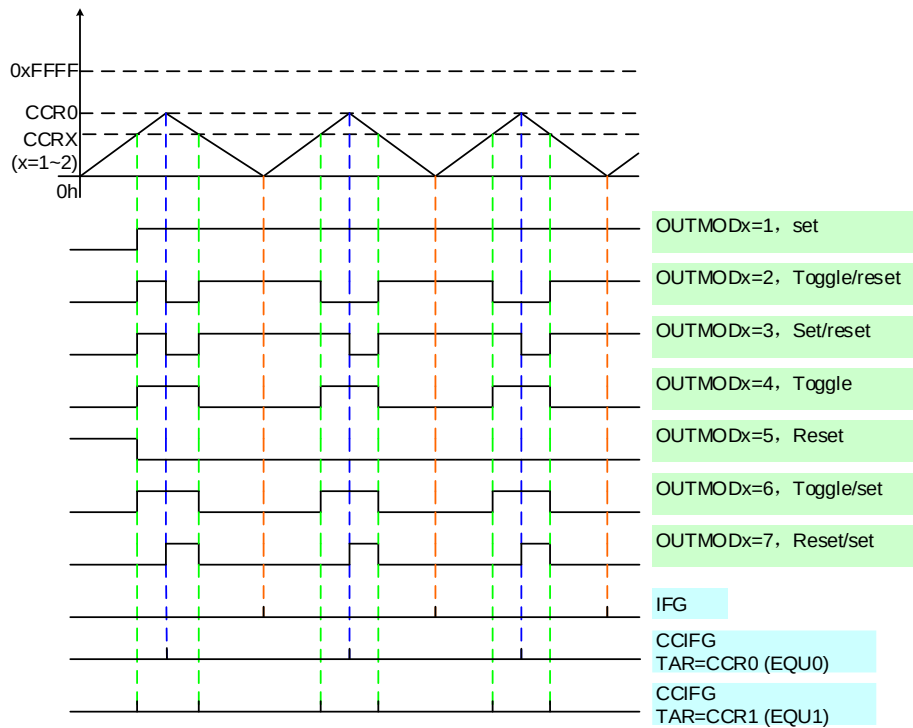
下图是在连续计数模式下的 PWM 输出：

图35. 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36. 向上/向下计数模式下的 PWM 输出



19.5.8 PWMx_CCRx 寄存器

表262. PWMx_CCRx 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 0	CCR _x	rw	比较数据寄存器。该寄存器用于与 TAR 比较以生成 PWM 输出。	0x0

19.5.9 PWM_O_SEL 寄存器

表263. PWM_O_SEL 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 12	O_SEL3	rw	外部引脚输出 PWM3 的输出选择寄存器。与 O_SEL0 的定义相同。	0xD
11: 8	O_SEL2	rw	外部引脚输出 PWM2 的输出选择寄存器。与 O_SEL0 的定义相同。	0xB
7: 4	O_SEL1	rw	外部引脚输出 PWM1 的输出选择寄存器。与	0x5



			O_SEL0 的定义相同。	
3: 0	O_SEL0	rw	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 LCD 控制器

20.1 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。

20.2 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 VLDO 产生，在 3.6V-2.7V 间以 60mV/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。

20.3 功能框图

图37. LCD 控制器的功能框图

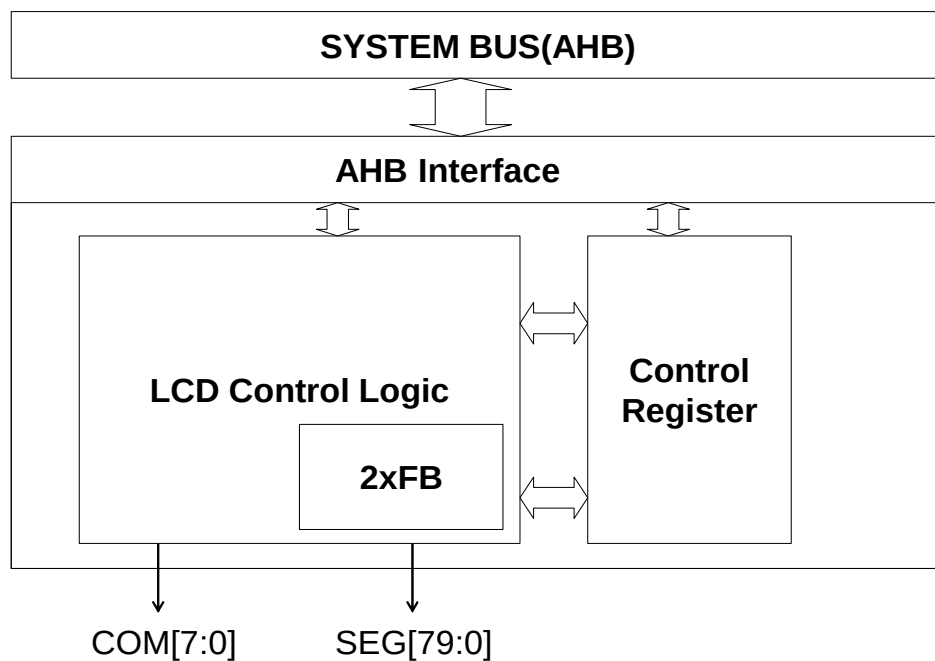
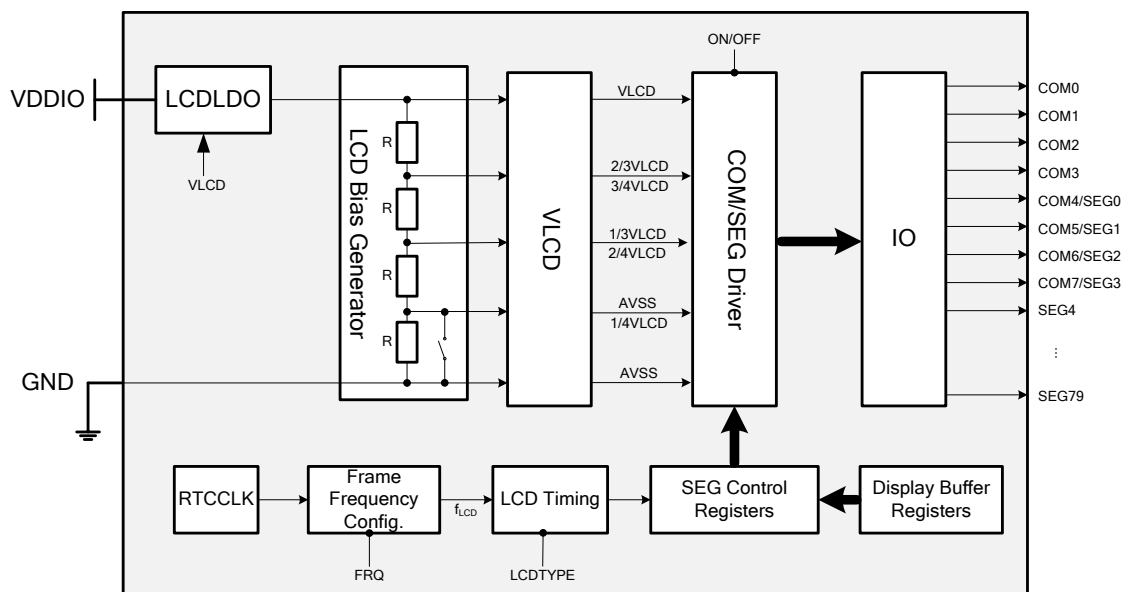


图38. LCD 控制器结构图



20.4 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 78 段。LCD 控制器内有两个具有自动切换功能的帧缓存器。

20.5 LCD 波形电压

V85XX 中，LCD 驱动电路的偏置电压由独立的 AVCC 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。

20.6 LCD 工作电流

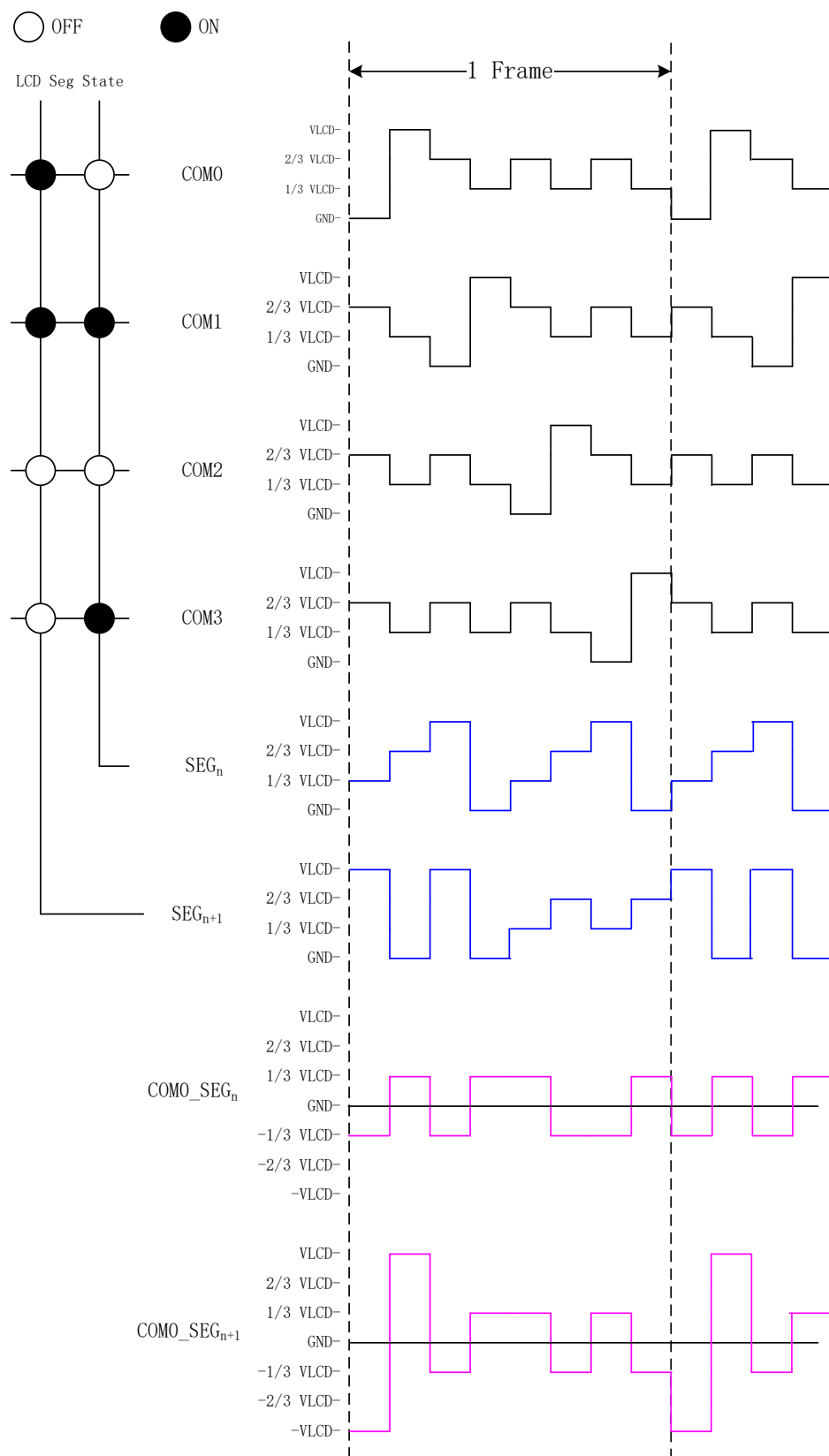
用户可以改变寄存器 LCD_CTRL 中的控制位 DRV，配置偏置电压产生电路中的各分压电阻的阻值，改变电路电流大小，从而调整液晶屏显示亮度。默认情况下各分压电阻的阻值为 300kΩ。

20.7 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 LCD_BMOD 位（ANA_REG6 的位 0）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。

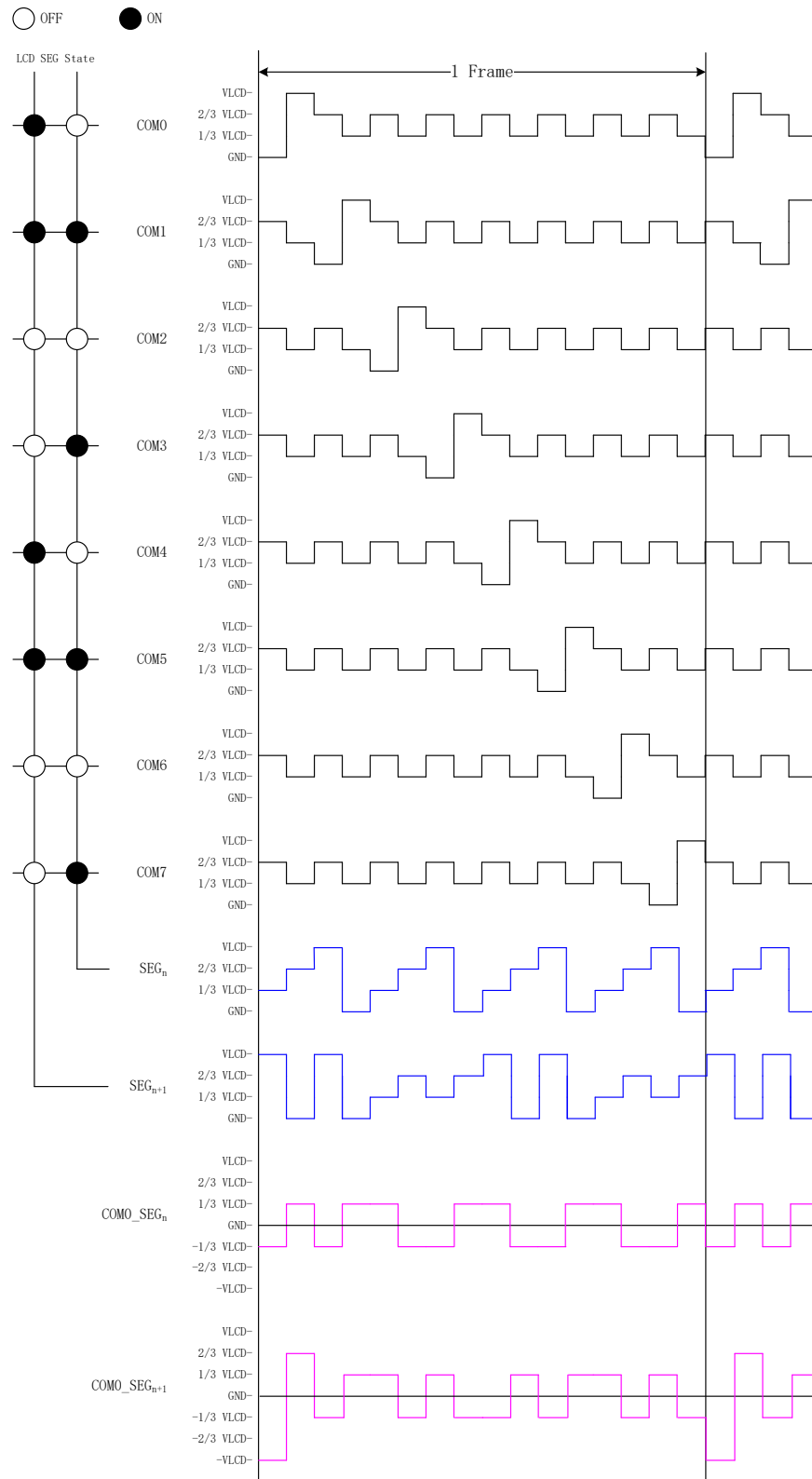
下图为[1/4Duty, 1/3Bias]的 LCD 驱动波形（示例）。

图39. LCD 波形[1/4 Duty, 1/3 Bias]



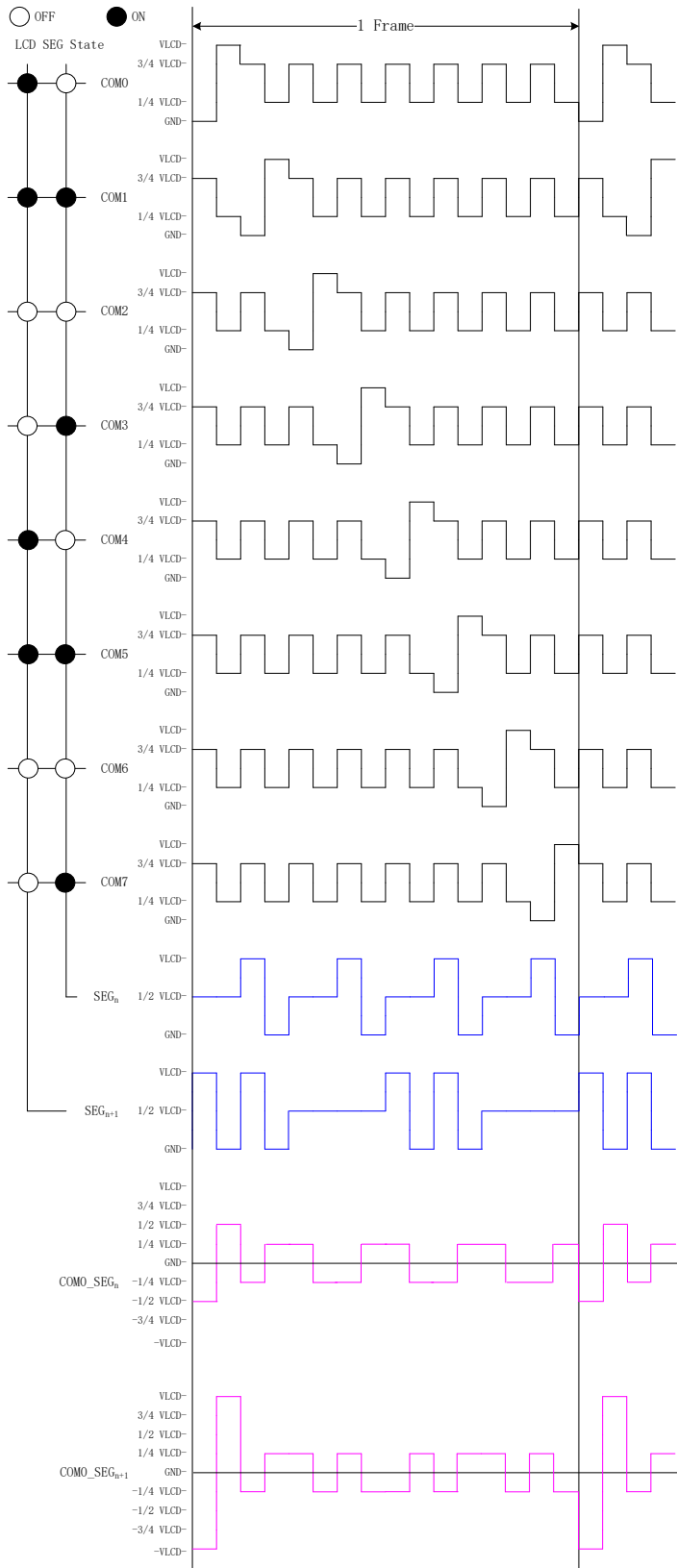
下图为[1/8Duty, 1/3Bias]的 LCD 驱动波形（示例）

图41. 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42. LCD 波形[1/8 Duty , 1/4 Bias]



20.8 寄存器地址

表264. 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	rw	0x0018	模拟寄存器 6	0x00

表265. LCD 控制器 (LCD 基地址: 0x40002000)

名称	类型	地址	描述	默认值
LCD_FB00	rw	0x0000	LCD 帧缓存 0 寄存器	--
LCD_FB01	rw	0x0004	LCD 帧缓存 1 寄存器	--
LCD_FB02	rw	0x0008	LCD 帧缓存 2 寄存器	--
LCD_FB03	rw	0x000C	LCD 帧缓存 3 寄存器	--
LCD_FB04	rw	0x0010	LCD 帧缓存 4 寄存器	--
LCD_FB05	rw	0x0014	LCD 帧缓存 5 寄存器	--
LCD_FB06	rw	0x0018	LCD 帧缓存 6 寄存器	--
LCD_FB07	rw	0x001C	LCD 帧缓存 7 寄存器	--
LCD_FB08	rw	0x0020	LCD 帧缓存 8 寄存器	--
LCD_FB09	rw	0x0024	LCD 帧缓存 9 寄存器	--
LCD_FB0A	rw	0x0028	LCD 帧缓存 10 寄存器	--
LCD_FB0B	rw	0x002C	LCD 帧缓存 11 寄存器	--
LCD_FB0C	rw	0x0030	LCD 帧缓存 12 寄存器	--
LCD_FB0D	rw	0x0034	LCD 帧缓存 13 寄存器	--
LCD_FB0E	rw	0x0038	LCD 帧缓存 14 寄存器	--
LCD_FB0F	rw	0x003C	LCD 帧缓存 15 寄存器	--
LCD_FB10	rw	0x0040	LCD 帧缓存 16 寄存器	--
LCD_FB11	rw	0x0044	LCD 帧缓存 17 寄存器	--
LCD_FB12	rw	0x0048	LCD 帧缓存 18 寄存器	--
LCD_FB13	rw	0x004C	LCD 帧缓存 19 寄存器	--
LCD_FB14	rw	0x0050	LCD 帧缓存 20 寄存器	--
LCD_FB15	rw	0x0054	LCD 帧缓存 21 寄存器	--
LCD_FB16	rw	0x0058	LCD 帧缓存 22 寄存器	--
LCD_FB17	rw	0x005C	LCD 帧缓存 23 寄存器	--
LCD_FB18	rw	0x0060	LCD 帧缓存 24 寄存器	--
LCD_FB19	rw	0x0064	LCD 帧缓存 25 寄存器	--
LCD_FB1A	rw	0x0068	LCD 帧缓存 26 寄存器	--
LCD_FB1B	rw	0x006C	LCD 帧缓存 27 寄存器	--
LCD_FB1C	rw	0x0070	LCD 帧缓存 28 寄存器	--
LCD_FB1D	rw	0x0074	LCD 帧缓存 29 寄存器	--
LCD_FB1E	rw	0x0078	LCD 帧缓存 30 寄存器	--

LCD_FB1F	rw	0x007C	LCD 帧缓存 31 寄存器	--
LCD_FB20	rw	0x0080	LCD 帧缓存 32 寄存器	--
LCD_FB21	rw	0x0084	LCD 帧缓存 33 寄存器	--
LCD_FB22	rw	0x0088	LCD 帧缓存 34 寄存器	--
LCD_FB23	rw	0x008C	LCD 帧缓存 35 寄存器	--
LCD_FB24	rw	0x0090	LCD 帧缓存 36 寄存器	--
LCD_FB25	rw	0x0094	LCD 帧缓存 37 寄存器	--
LCD_FB26	rw	0x0098	LCD 帧缓存 38 寄存器	--
LCD_FB27	rw	0x009C	LCD 帧缓存 39 寄存器	--
LCD_CTRL	rw	0x0100	LCD 控制寄存器	0x00
LCD_CTRL2	rw	0x0104	LCD 控制寄存器 2	0x0000
LCD_SEGCTRL0	rw	0x0108	LCDSEG 使能控制寄存器 0	0x00000000
LCD_SEGCTRL1	rw	0x010C	LCDSEG 使能控制寄存器 1	0x00000000
LCD_SEGCTRL2	rw	0x0110	LCDSEG 使能控制寄存器 2	0x00000000

20.9 寄存器定义

20.9.1 LCD_FBx 寄存器

表266. LCD_FBx 寄存器

位	名称	类型	描述	默认值
31: 0	DATAx	rw	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 3~6。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267. LCD_FBx 寄存器的数据排布

寄存器	数据位			
	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	SEG78 对应的	SEG77 对应的	SEG76 对应的

	COM[7: 0]	COM[7: 0]	COM[7: 0]	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。

20.9.2 LCD_CTRL 寄存器

表268. LCD_CTRL 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7	EN	rw	LCD 使能寄存器。 0: 禁用 1: 启用	0x0
6	Rsvd	-	保留	0
5: 4	TYPE	rw	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	rw	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	rw	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1: 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：
 $T_{Delay}=1/（64/4）=62.5ms$

20.9.3 LCD_CTRL2 寄存器

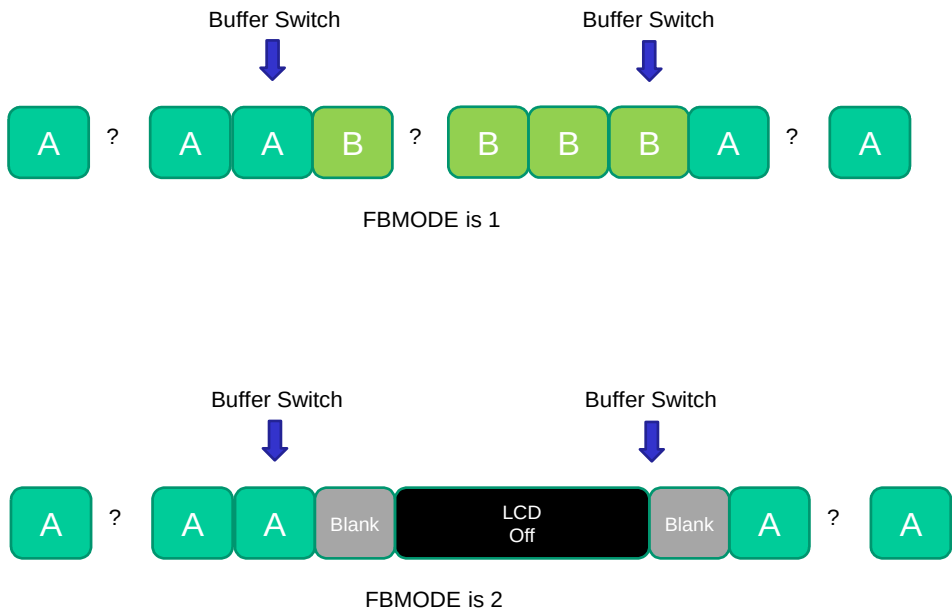
表269. LCD_CTRL2 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 8	SWPR	rw	帧缓存器转换周期。转换周期计算公式如下：	0x0



			0.5 秒* (SWPR+1)	
7: 6	FBMODE	rw	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	Rsvd	-	保留	0
4	BKFILL	rw	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	Rsvd	-	保留	0

图43. 不同 FBMODE 下的帧缓存器操作模式



20.9.4 LCD_SEGCTRL0 寄存器

表270. LCD_SEGCTRL0 寄存器

位	名称	类型	描述	默认值
31: 0	SEGCTRL	rw	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。	0x00000000

			1: 使能 SEG 输出。	

20.9.5 LCD_SEGCTRL1 寄存器

表271. LCD_SEGCTRL1 寄存器

位	名称	类型	描述	默认值
31: 0	SEGCTRL	rw	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

20.9.6 LCD_SEGCTRL2 寄存器

表272. LCD_SEGCTRL2 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 0	SEGCTRL	rw	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

20.9.7 ANA_REG6 寄存器

表273. ANA_REG6 寄存器

位	名称	功能	注释
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=0~5: 调整幅度 =+60mV*VLCD 当 VLCD=6~15: 调整幅度=- 60mV* (VLCD-5)

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。

21 SPI 控制器

21.1 简介

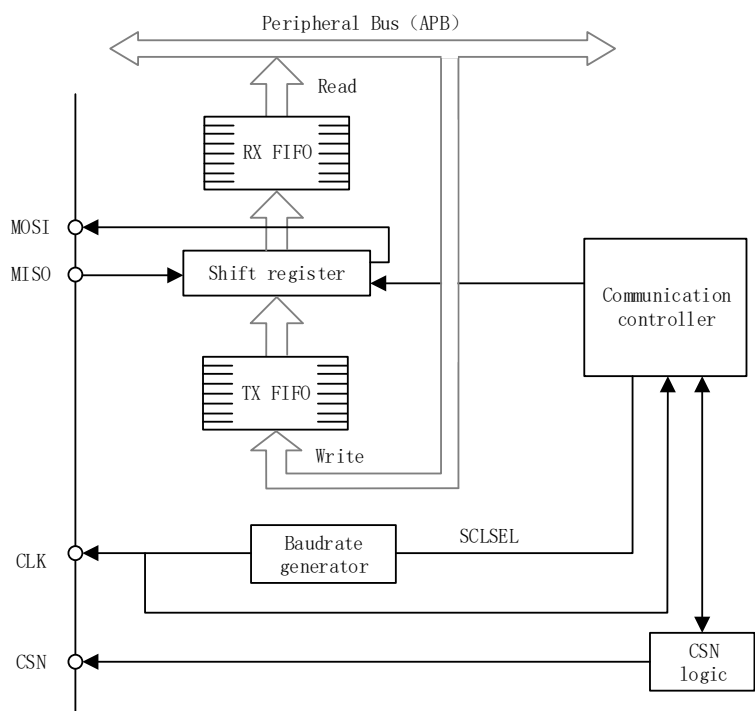
V85XX 中集成了 4 个串行外设接口（SPI）控制器，以便与其他设备或组件通信。SPI 控制器中的设置在从睡眠模式唤醒后会被复位，用户应在睡眠模式唤醒后手动恢复设置。SPI 主机模式支持的 SPICLK 时钟：APBCLK/2/4/8/16/32/64/128。SPI 从机模式支持的最大 SPICLK 时钟为 APBCLK/2。

21.2 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
- 支持接收溢出错误指示
- 支持发送/接收中断请求
- 主机模式下相位和极性可编程
- 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 SCK 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。

21.3 功能框图

图44. 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1: SPI 从机模式支持的最大 SPICLK 时钟为 1/2 APBCLK。

21.4 寄存器地址

表274. SPI1 控制器（SPI1 基地址：0x40011000）

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



表275. SPI2 控制器（SPI2 基地址：0x40015800）

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

表276. SPI3 控制器（SPI3 基地址：0x40016000）

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

表277. SPI4 控制器（SPI4 基地址：0x40017000）

名称	类型	地址	描述	默认值
SPI4_CTRL	rw	0x0000	SPI4 控制寄存器	0x0000
SPI4_TXSTS	rw	0x0004	SPI4 发送状态寄存器	0x8200
SPI4_TXDAT	rw	0x0008	SPI4 发送 FIFO 寄存器	--
SPI4_RXSTS	rw	0x000C	SPI4 接收状态寄存器	0x0000
SPI4_RXDAT	rw	0x0010	SPI4 接收 FIFO 寄存器	--
SPI4_MISC	rw	0x0014	SPI4 Misc 控制寄存器	0x0003

21.5 寄存器定义

21.5.1 SPIx_CTRL 寄存器

SPIx_CTRL 寄存器用来控制 SPIx 控制器的行为。

表278. SPIx_CTRL 寄存器

位	名称	类型	描述	默认值
15	SPIEN	rw	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。 0：禁用 SPI 功能	0x0

			1: 启用 SPI 功能	
14: 12	Rsvd	-	保留	0
11	SPIRST	w	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。 0: 无效 1: 复位 SPI 控制器	0x0
10	CSGPIO	rw	SPICS 管脚由 GPIO 或 H/W 控制。 主控模式: 0: SPICS 管脚由硬件 H/W 控制 1: SPICS 管脚由软件 GPIO 控制 从机模式: 0: SPICS 管脚由外部主机控制 1: SPICS 管脚为逻辑 0	0x0
9	SWAP	rw	SPI MISO/MOSI 引脚互换控制寄存器。 0: 不交换 MISO/MOSI 1: 交换 MISO/MOSI	0x0
8	MOD	rw	SPI 模式选择寄存器。 0: 主机模式 1: 从机模式	0x0
7: 6	Rsvd	-	保留	0
5	SCKPHA	rw	SPI 时钟相位, 请参考时序原理图。	0x0
4	SCKPOL	rw	SPI 时钟极性, 请参考时序原理图。	0x0
3	Rsvd	-	保留	0
2: 0	SCKSEL	rw	主机模式时钟选择。 000: APBCLK/2 001: APBCLK/4 010: APBCLK/8 011: APBCLK/16 100: APBCLK/32 101: APBCLK/64 110: APBCLK/128 111: 保留。	0x0

21.5.2 SPIx_TXSTS 寄存器

SPIx_TXSTS 寄存器控制 SPIx 发送 FIFO 和相关中断。

表279. SPIx_TXSTS 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15	SPITXIF	rc_w1	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	0x1

			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：未发生中断 读 1：发生中断 写 0：无效 写 1：清除标志	
14	SPITXIEN	rw	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	Rsvd	-	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	Rsvd	-	保留	0
6: 4	TXFLEV	rw	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。 010：FIFO 中的数据<3，允许写入 6 个数据。 011：FIFO 中的数据<4，允许写入 5 个数据。 100：FIFO 中的数据<5，允许写入 4 个数据。 101：FIFO 中的数据<6，允许写入 3 个数据。 110：FIFO 中的数据<7，允许写入 2 个数据。 111：FIFO 中的数据<8，允许写入 1 个数据。	0x0
3	Rsvd	-	保留	0
2: 0	TXFFLAG	r	发送 FIFO 中数据个数。 该寄存器用于指示 FIFO 中的数据数量。 0000：FIFO 中没有数据或 FIFO 中有 8 个字节。 0001：FIFO 中有 1 个字节。	0x0

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

21.5.3 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280. SPIx_TXDAT 寄存器

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

21.5.4 SPIx_RXSTS 寄存器

SPI_RXSTS 寄存器用于控制 SPI 接收 FIFO 及相关中断。

表281. SPIx_RXSTS 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15	SPIRXIF	rc_w1	SPI 接收中断标志。 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”。 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。 读 0: 未发生中断 读 1: 发生中断 写 0: 无效 写 1: 清除标志	0x0
14	SPIRXIEN	rw	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	Rsvd	-	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 未满足时，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 会将该位清零。	0x0
7	Rsvd	-	保留	0
6: 4	RXFLEV	rw	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	Rsvd	-	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。 0110: FIFO 中有 6 个字节。 0111: FIFO 中有 7 个字节。	0x0

21.5.5 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282. SPIx_RXDAT 寄存器

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

21.5.6 SPIx_MISC 寄存器

SPI_MISC 寄存器用于配置 SPI 接收发送特殊功能。

表283. SPIx_MISC 寄存器

位	名称	类型	描述	默认值
31: 10	Rsvd	-	保留	0
9	OVER	rw	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	rw	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 SMART 模式，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	Rsvd	-	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 未滿 1: 接收 FIFO 已滿	0x0
2	RNE	r	接收 FIFO 非空标记。 该位显示接收 FIFO 是否有数据。 0: 接收 FIFO 为空 1: 接收 FIFO 非空	0x0
1	TNF	r	发送 FIFO 未滿标记。 该位显示发送 FIFO 是否未滿。 0: 发送 FIFO 满，不能写入 1: 发送 FIFO 未滿	0x1
0	TFE	r	发送 FIFO 为空标记 该位显示发送 FIFO 是否为空。 0: 发送 FIFO 非空	0x1

			1: 发送 FIFO 空	

21.6 应用笔记

21.6.1 主机模式

主控模式下，移位时钟（SPICLK）由 V85XX 产生。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重新开始发送下一个数据。

SPI 从 SDO 引脚发送数据时，先发高位，再发低位。移出 8 位数据需要 8 个 SCK 周期。同时，数据也通过从器件 SDI 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 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 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”）。

图45. 主机模式，SPO=0，SPH=0

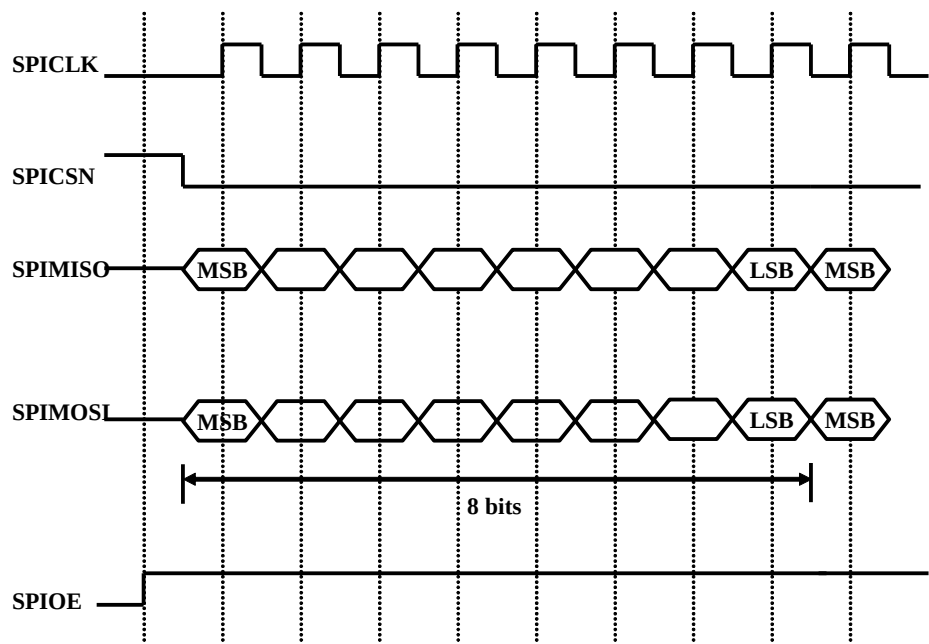


图46. 主机模式，SPO=0，SPH=1

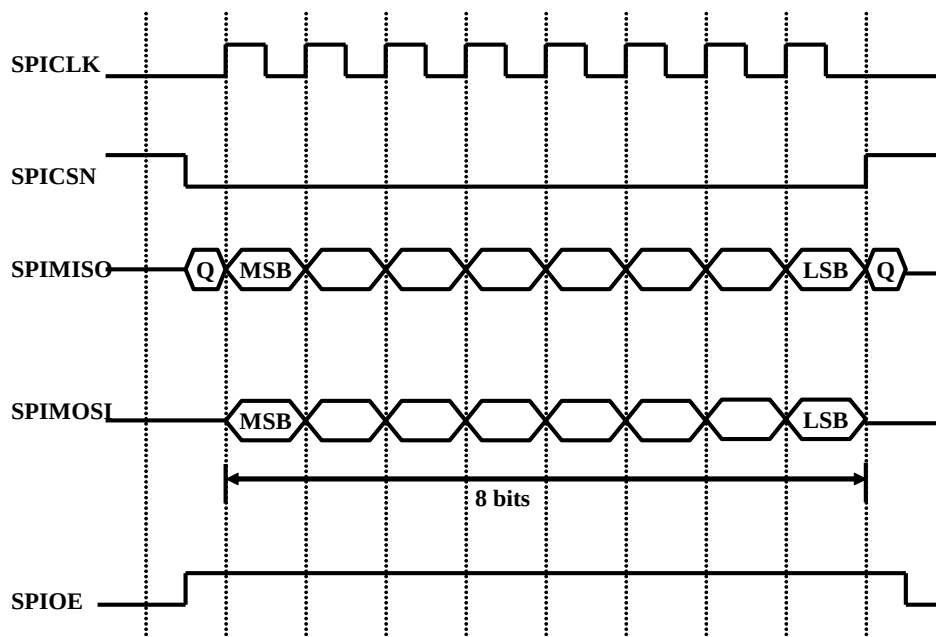


图47. 主机模式, SPO=1, SPH=0

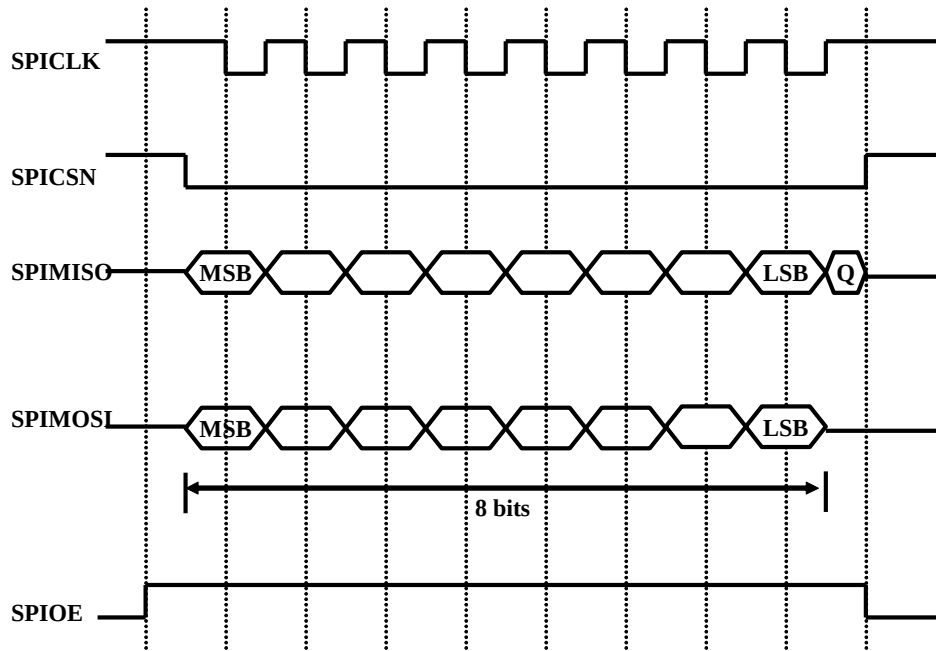
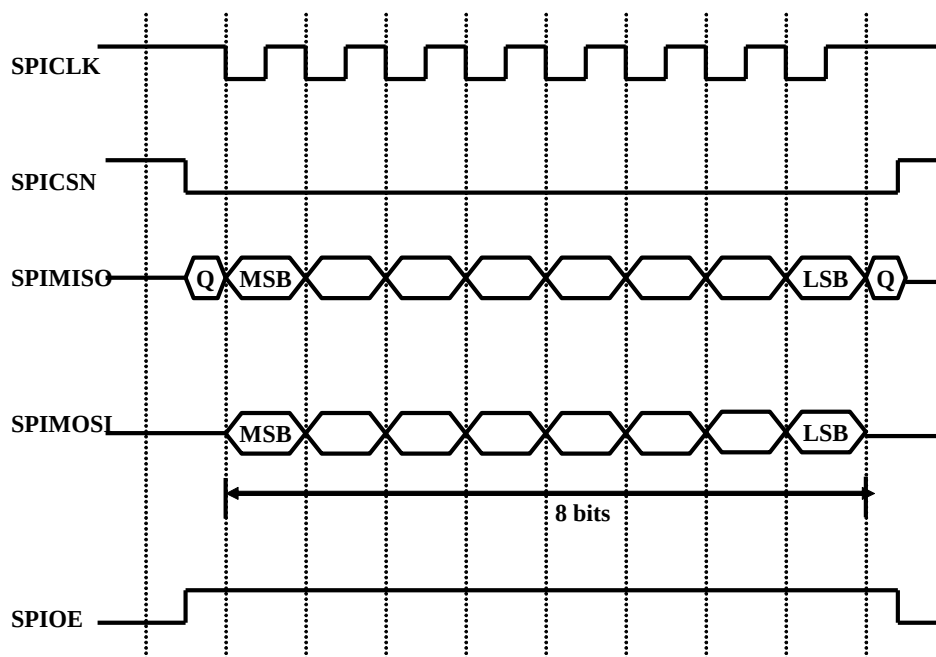


图48. 主机模式，SPO=1，SPH=1



21.6.2 从机模式

SPI 从机模式时，SPICLK 和 SPICSN 由外部主机产生。SPI 从机模式仅支持时钟极性和相位都为 0（SPO=0 和 SPH=0）。SPI_TXDAT 和 SPI_RXDAT 寄存器都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 1/2APBCLK。当接收时钟速率较快时，建议使用 DMA 传输。

21.6.3 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，软件会检查溢出标志，以监测是否由于轮询速率太低而丢失数据。

22 I2C 控制器

22.1 简介

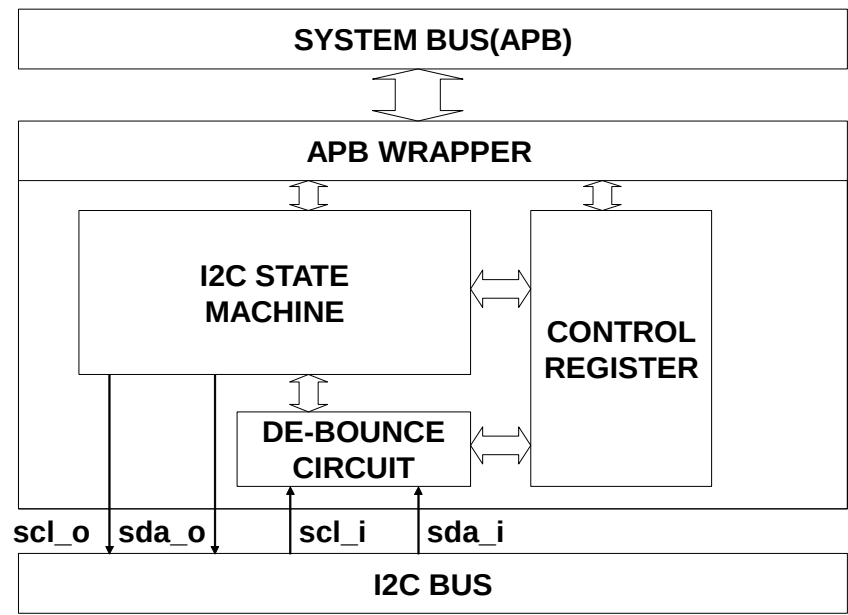
I2C 子组件是 I2C 总线控制器，它提供了一个满足 Philips I2C 总线规范的接口，并支持 I2C 总线双向传输。I2C 总线使用两条线在连接到总线的设备之间传输信息：“scl”（串行时钟线）和“sda”（串行数据线）。I2C 支持单字节自动传输，支持串行传输，状态寄存器反映 I2C 总线控制器和 I2C 总线的状态。I2C 控制器中的设置在从睡眠模式唤醒后会被复位，用户应在睡眠模式唤醒后手动恢复设置。I2C 总线最大支持 $f_{SCL} = 1\text{MHz}$ （周期： $1\mu\text{s}$ ），可以通过设置定时器 3 溢出频率实现。

22.2 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。

22.3 功能框图

图49. I2C 控制器功能框图



22.4 寄存器地址

表284. I2C 控制器（I2C 基地址：0x40010800）

名称	类型	地址	描述	默认值
I2C_DATA	rw	0x0000	I2C 数据寄存器	0x00
I2C_ADDR	rw	0x0004	I2C 地址寄存器	0x00
I2C_CTRL	rw	0x0008	I2C 控制/状态寄存器	0x00
I2C_STS	rw	0x000c	I2C 状态寄存器	0xF8
I2C_CTRL2	rw	0x0018	I2C 中断使能寄存器	0x0

22.5 寄存器定义

22.5.1 I2C_DATA 寄存器

表285. I2C_数据寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7: 0	DATA	rw	I2C_DATA 寄存器包含一个要通过 I2C 总线传输的字节或刚通过 I2C 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 I2C 中断时读取 I2C_DATA。	0x00

22.5.2 I2C_ADDR 寄存器

表286. I2C_ADDR 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7: 1	SLA	rw	自有 I2C 从地址（7 位）	0x00
0	GC	rw	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

22.5.3 I2C_CTRL 寄存器

表287. I2C_CTRL 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7	CR2	rw	时钟速率控制位 2	0x0
6	EN	rw	I2C 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'，I2C 组件被使能，相应 IO 将自动设置为 I2C 功能。	0x0
5	STA	rw	START 标志 当 STA='1'时，I2C 组件检查 I ² C 总线状态，如果总线空闲，则产生 START 状态。	0x0
4	STO	rw	STOP 标志 当 STO='1'且 I ² C 接口处于主机模式时，会向 I2C 总线发送一个 STOP 状态。	0x0
3	SI	rc_w1	串行中断标志 I2C 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	rw	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 ² C 地址寄存器中的 GC 位置 1 时接收到广播呼	0x0

			叫地址 -当 I2C 处于主接收器模式时，已接收到数据字节 -当 I2C 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 I2C 处于主接收器模式时，已接收到数据字节 -当 I2C 处于从接收器模式时，已接收到数据字节	
1	CR1	rw	时钟速率控制位 1	0x0
0	CR0	rw	时钟速率控制位 0	0x0

SCL 频率设置显示在不同 CR2~CR0 设置下，I2C 作为主机的主 SCL 时钟频率。

表288. SCL 频率设置

CR2	CR1	CR0	SCL 频率（KHz）				时钟分频数
			PCLK= 6.5536 MHz	PCLK= 13.1072 MHz	PCLK= 19.6608 MHz	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				

22.5.4 I2C_STS 寄存器

表289. I2C_STS 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7: 3	STS	r	I2C 状态码	0x1F
2: 0	Rsvd	-	保留	0

表290. 主机发送模式下的 I2C 状态

STS	I2C 状态	应用软件响应				I2C 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, I2C 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	I2C 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	I2C 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], 0 表示写。

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

表291. 主机接收模式下的 I2C 状态

STS	I2C 状态	应用软件响应					I2C 的下一步动作
		读/写	I2C_CTRL				
			I2C_DATA	STA	STO	SI	
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，I2C 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	SLA+R 已发送，ACK 未接收。	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 标志将重置。
		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	I2C 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	I2C 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292. 从机接收模式下的 I2C 状态

STS	I2C 状态	应用软件响应					I2C 的下一步动作
		读/写	I2C_CTRL				
			I2C_DATA	STA	STO	SI	
0x60	接收到自有 SLA+W, 已	无操作	X	0	0	0	将接收数据字节, 并且不返回 ACK。

	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293. 从机发送模式下的 I2C 状态

STS	I2C	应用软件响应					I2C 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	写数据字节	X	0	0	0	将发送上一个数据字节, 并接收

	SLA+R/W 作为主机时 丢失。已接 收自有 SLA+R, ACK 已返 回。						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294. I2C 状态

STS	I2C 状态	应用软件响应				I2C 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。

22.5.5 I2C_CTRL2 寄存器

表295. I2C_CTRL2 寄存器

位	名称	类型	描述	默认值
31: 1	Rsvd	-	保留	0
0	INTEN	rw	I2C 控制器的中断使能控制。 0: 禁止。 1: 使能。	0x0

22.6 应用笔记

22.6.1 I2C 总线协议：启动/停止的发生

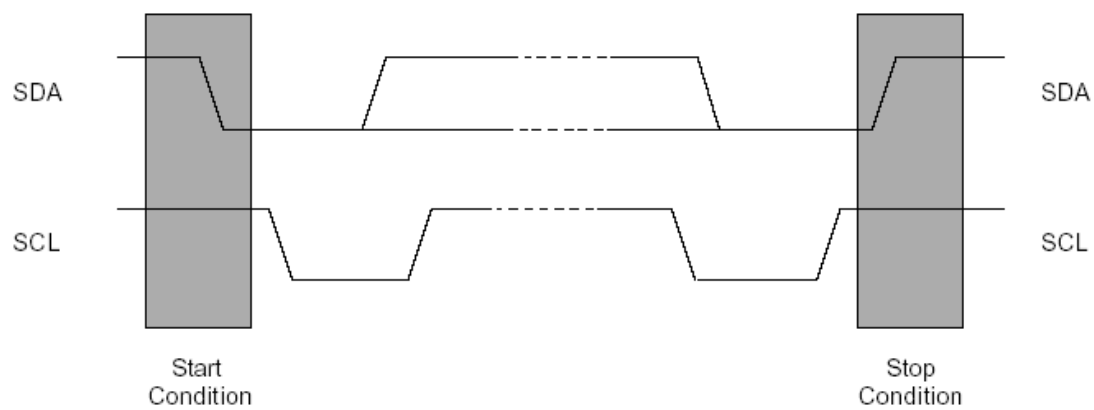
启动状态可以通过 SDA 线传输一个字节的串行数据，停止状态可以终止数据传输。停止状态是 SCL 为高电平时 SDA 线的低电平向高电平跳变。开始和停止状态始终由主器件生成。当产生启动状态时，I2C 总线忙。停止状态后几个时钟，I2C 总线将再次被释放。

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，则它表示数据读取请求（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 I2C 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50. 启动和停止条件





22.6.2 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 I2C 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51. 7 位地址写模式

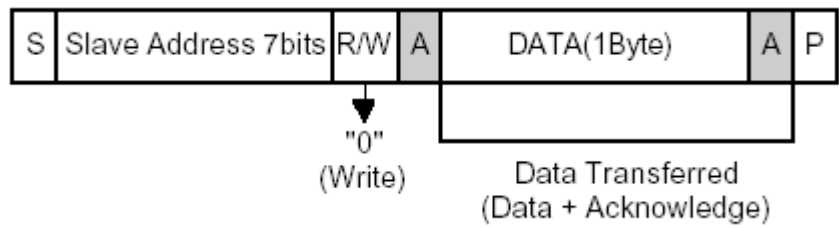


图52. 7 位地址读模式

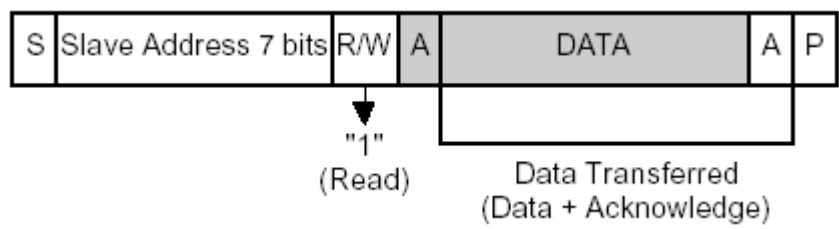
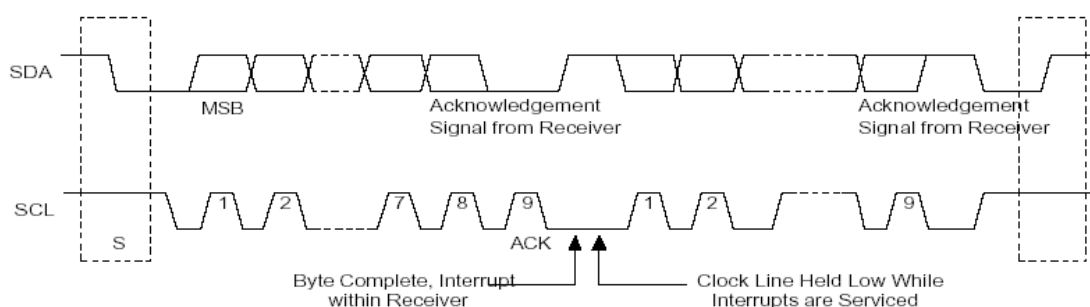


图53. 数据传输



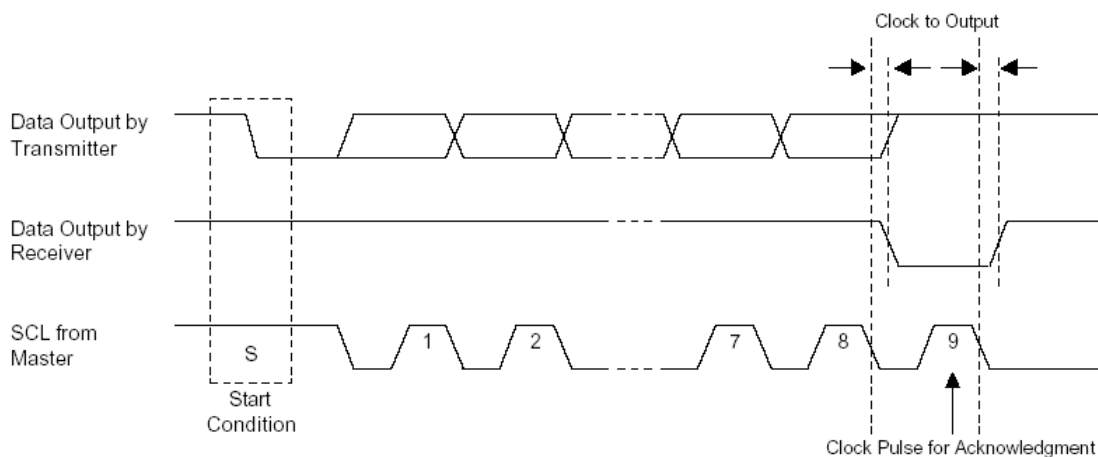
22.6.3 ACK 信号传输

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（AA）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54. I2C 总线的应答机制



22.6.4 读写操作

发送模式下，数据传输后，I2C 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，I2C 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，当主机的 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线路上保持高电平，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

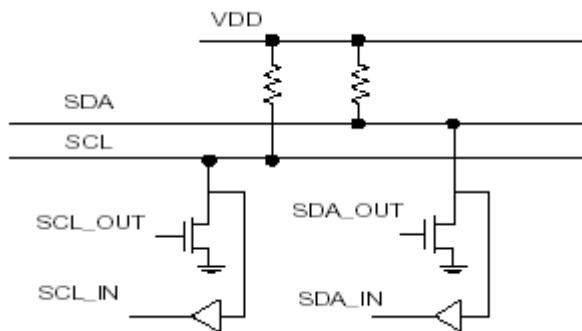
如果从接收器不应答从机地址，它应该保持 SDA 线为高电平。在这种情况下，主机应产生停止状态并中止传输。

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。

22.6.5 接口时序

下图显示了 I²C 总线的接口连接。

图55. I2C 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 μs ）

-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56. SCL 和 SDA 时序

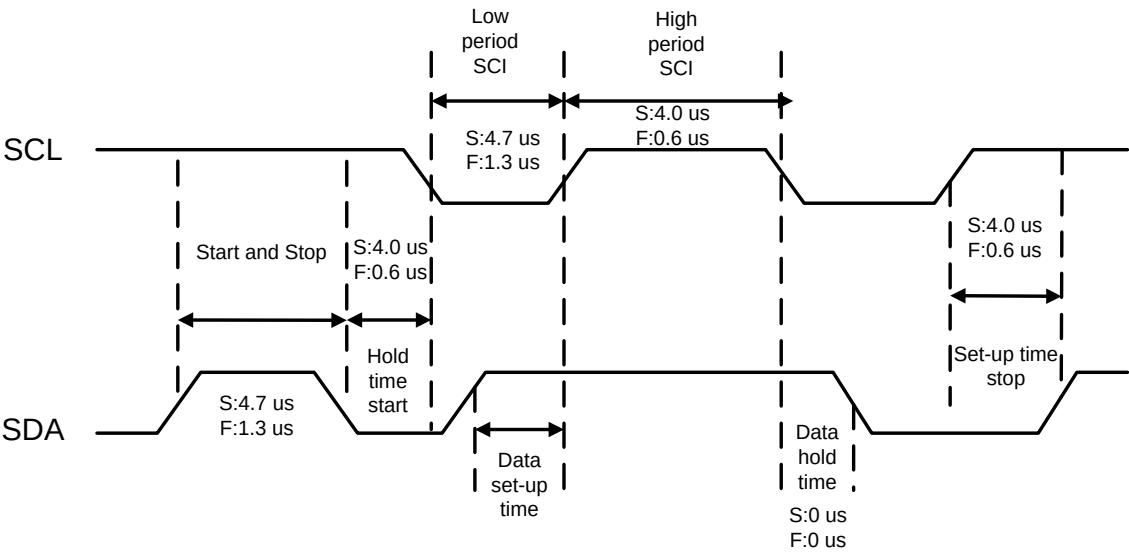


表296. I2C AC 特性

Time	I2C-mastermode (CR<7)		I2C-mastermode (CR=7)	
t _{HD} : STA	scl/4	min (15clk)	scl/4	(2bclk)
t _{LOW}	scl/2	min (30clk)	scl/2	(4bclk)
t _{HIGH}	scl/2	min (30clk)	scl/2	(4bclk)
t _{SU} : STA	scl/4	min (15clk)	scl/2-if (scl/4<7clk)	
			scl/4-others	
t _{HD} : DAT	6*clk	(6clk)	6*clk	(6clk)
t _{SU} : DAT	1*clk (-forfirstbit) *	(1clk)	1*clk (-forfirstbit) *	(1clk)
	scl/2-6*clk	min (24clk)	scl/2-6*clk	min (1clk)
t _{SU} : STO	scl/4+7*clk	min (22clk)	min (bclk+9)	
t _{BUF}	(3/4scl) +9*clk	min (54clk)	min (3/4scl)	

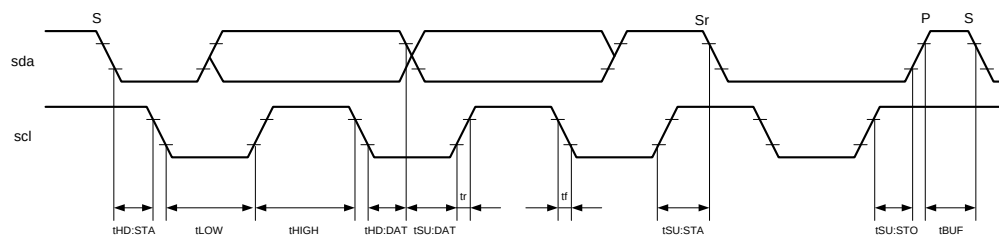
*-第一个字节数据的第一个位建立时间会短一点，只有一个 clk，而这种状况只会发生在写入数据寄存器时，刚好很接近内部 CSL 的上升沿。

Clk: APB 时钟周期

bclk: 定时器 3 的溢出周期

scl: I2C 时钟周期

图57. I2C 控制器的 AC 时序



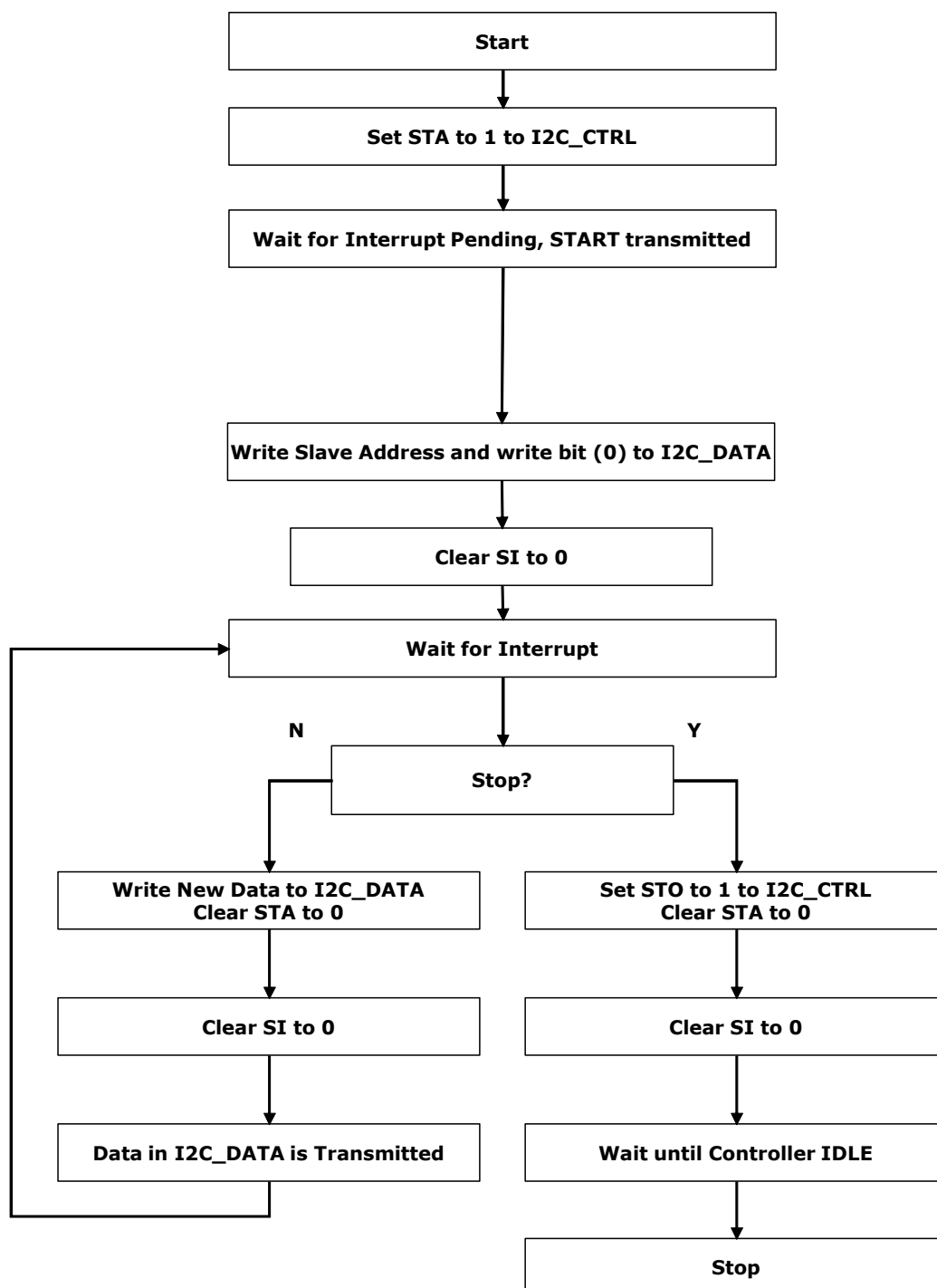
22.6.6 I2C SDA 管脚死锁的解决方法

I2C 通讯过程中，当芯片异常复位（软复位、看门狗复位、外部 EXRST 管脚复位）发生时，有可能导致 I2C 总线死锁（SDA 死锁在低电平状态）。规避方法：在开启 IIC 之前，用户增加 I2C 总线恢复时序。即每次 I2C 主设备复位后，软件模拟控制 I2C 中的 SCL 时钟线产生 9 个时钟脉冲，这样 I2C 的 SDA 管脚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 I2C 使用。

22.7 工作流程

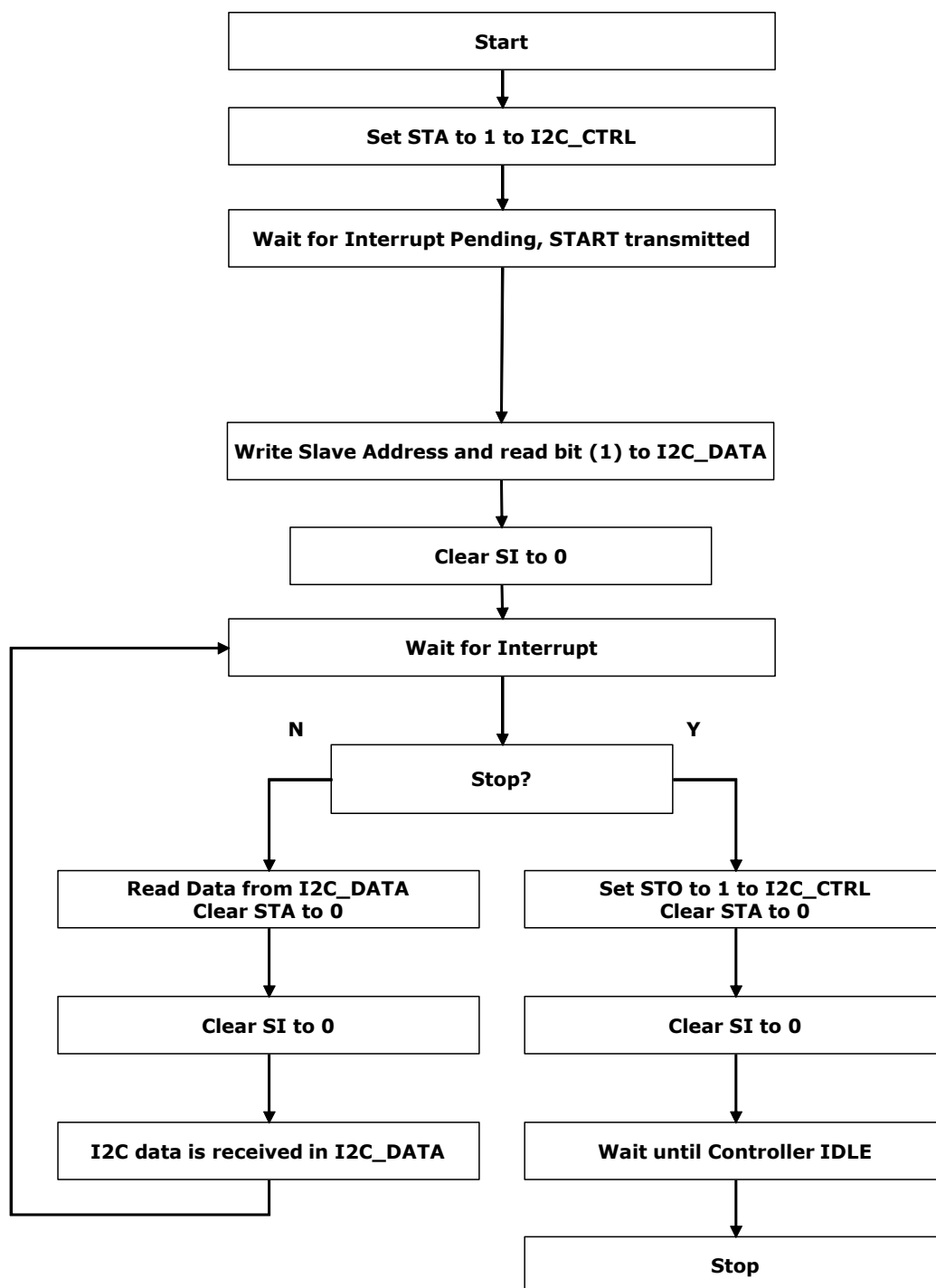
22.7.1 主机发送模式

图58. 主机发送模式的操作流程



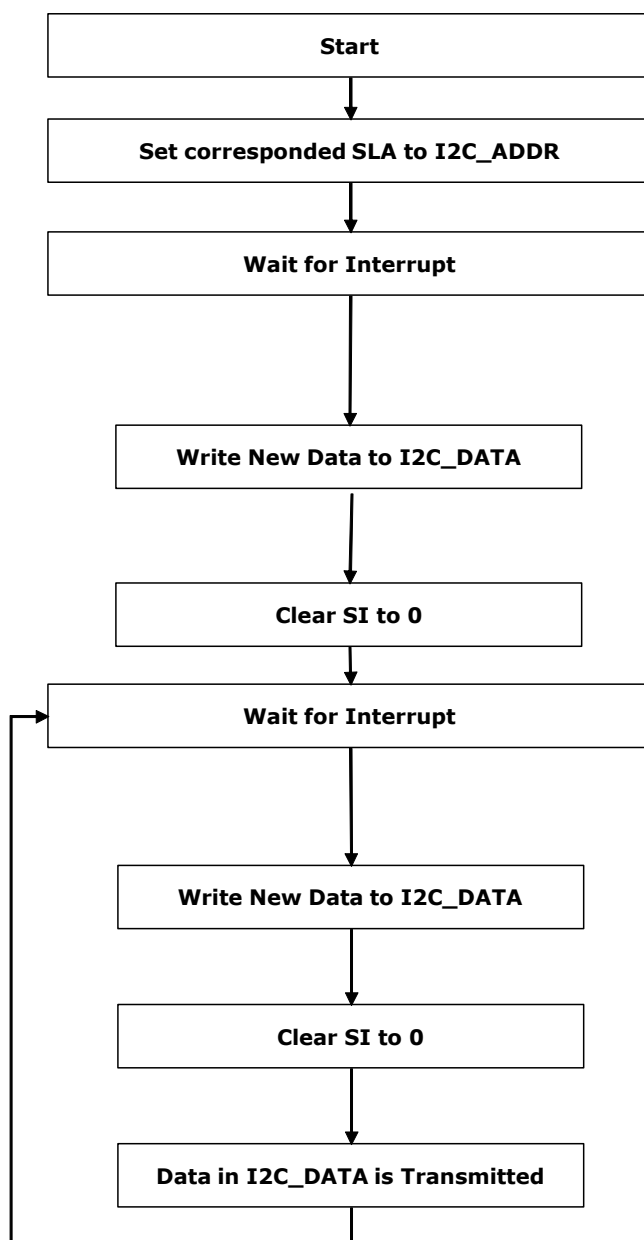
22.7.2 主机接收模式

图59. 主机接收模式的操作流程



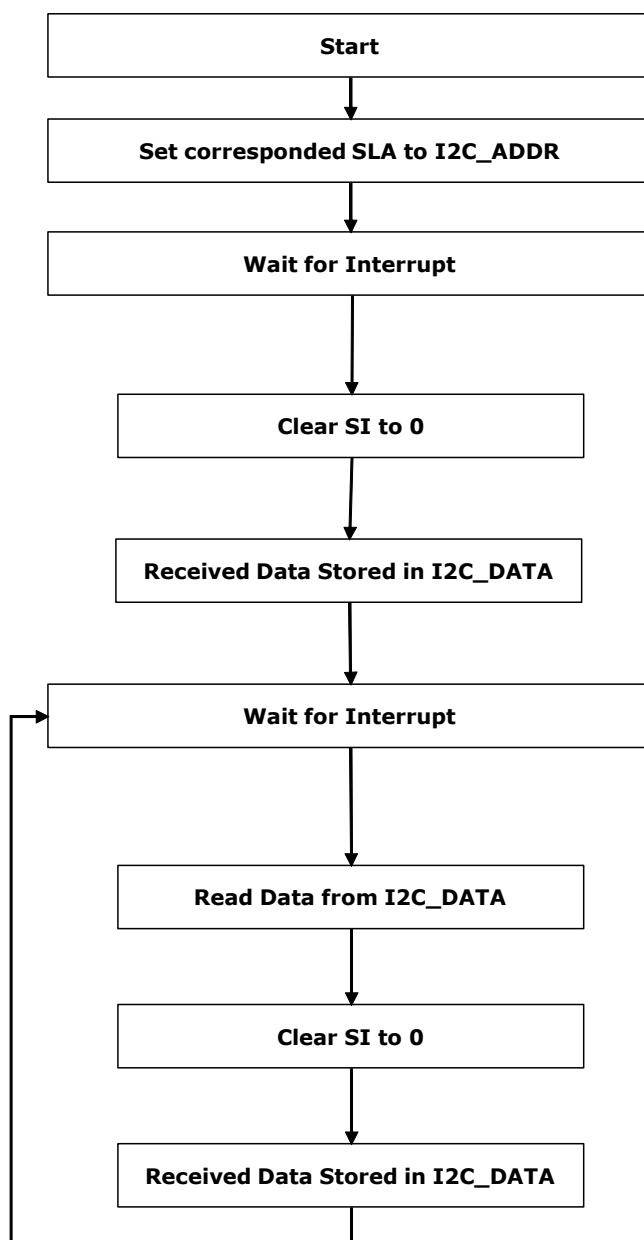
22.7.3 从机发送模式

图60. 从机发送模式的操作流程



22.7.4 从机接收模式

图61. 从机接收模式的操作流程



23 中断控制器

23.1 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睡眠（SLEEP）中唤醒。

23.2 中断源

表297. 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
						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 and PMU_IOAWKUEN.0~15	PMU_IOAINTSTS.0~15	V
			32K 晶振失效	PMU_CONTROL.2	PMU_STS.0	V
			6M 晶振失效	PMU_CONTROL.3	PMU_STS.1	
RTC	00000044h	1	保留	RTC_INTEN.0	RTC_INTSTS.0	V
			非法时间格式	RTC_INTEN.1	RTC_INTSTS.1	V
			多秒中断	RTC_INTEN.2	RTC_INTSTS.2	V
			多分钟中断	RTC_INTEN.3	RTC_INTSTS.3	V
			多小时中断	RTC_INTEN.4	RTC_INTSTS.4	V
			午夜（00: 00）中断	RTC_INTEN.5	RTC_INTSTS.5	V

			32K 计数器中断	RTC_INTEN.6	RTC_INTSTS.6	V	V
			-	-	-	-	-
			对 CE 寄存器的非法写入	RTC_INTEN.8	RTC_INTSTS.8		
U32K0~1	00000048h 0000004Ch	2\3	接收到数据	U32Kx_CTRL1.0	U32Kx_STS.0	V	V
			接收奇偶校验错误	U32Kx_CTRL1.1	U32Kx_STS.1	V	V
			接收缓存溢出	U32Kx_CTRL1.2	U32Kx_STS.2	V	V
I2C	00000050h	4	I2C 串行中断	I2C_CTRL2.0	I2C_CTRL.3		
SPI1~4	00000054h \000000ACh \000000A8h \000000A4h	5\27\26\25	SPI 发送	SPIx_TXSTS.14	SPIx_TXSTS.15		
			SPI 接收	SPIx_RXSTS.14	SPIx_RXSTS.15		
UART0~5	00000058h ~00000006Ch	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		
ISO78160~1	00000070h ~000000074h	12\13	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		
Timer0~3	00000078h ~000000084h	14\15\16\17	定时器溢出	TMRx_CTRL.3	TMRx_INT.0		
PWM0~3	00000088h	18\19\20\21	PWM 定时器溢出	PWMx_CTL.1	PWMx_CTL.0		

	~000094h		比较 0	PWMx_CCTL0.4	PWMx_CCTL0.0		
			比较 1	PWMx_CCTL1.4	PWMx_CCTL1.0		
			比较 2	PWMx_CCTL2.4	PWMx_CCTL2.0		
DMA	00000098h	22	通道 0 包结束	DMA_IE.0	DMA_STS.0		
			通道 1 包结束	DMA_IE.1	DMA_STS.1		
			通道 2 包结束	DMA_IE.2	DMA_STS.2		
			通道 3 包结束	DMA_IE.3	DMA_STS.3		
			通道 0 帧结束	DMA_IE.4	DMA_STS.4		
			通道 1 帧结束	DMA_IE.5	DMA_STS.5		
			通道 2 帧结束	DMA_IE.6	DMA_STS.6		
			通道 3 帧结束	DMA_IE.7	DMA_STS.7		
			通道 0 数据丢弃	DMA_IE.8	DMA_STS.8		
			通道 1 数据丢弃	DMA_IE.9	DMA_STS.9		
			通道 2 数据丢弃	DMA_IE.10	DMA_STS.10		
			通道 3 数据丢弃	DMA_IE.11	DMA_STS.11		
FLASH	0000009Ch	23	校验和错误	FLASH_CTRL.2	FLASH_INT.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		
			COMP1 的上升和下降沿	ANA_INTEN.2	ANA_INTSTS.2	V	V
			COMP2 的上升和下降沿	ANA_INTEN.3	ANA_INTSTS.3	V	V
			VDDALARM 的上升和下降沿	ANA_INTEN.7	ANA_INTSTS.7	V	V
			VDCIN 的上升和下降沿	ANA_INTEN.8	ANA_INTSTS.8	V	V
			AVCCLV 的上升和下降沿	ANA_INTEN.10	ANA_INTSTS.10	V	V
			VDCINDROP=0 同时进入了睡眠状态	ANA_INTEN.11	ANA_INTSTS.11	V	V

			ANA_REGx 错误	ANA_INTEN.12	ANA_INTSTS.12	V	V
			TADC 变化超过阈值	ANA_INTEN.13	ANA_INTSTS.13	V	V
EPU	000000 B0h	28	正确接收 0xA5 帧	EPU_INTEN.12	EPU_INTSTS.12		
			接收字节超时	EPU_INTEN.11	EPU_INTSTS.11		
			接收数据 checksum 错误	EPU_INTEN.10	EPU_INTSTS.10		
			接收数据奇偶校验错误	EPU_INTEN.9	EPU_INTSTS.9		
			正确接收帧数据	EPU_INTEN.8	EPU_INTSTS.8		
			能量累加器 7 溢出	EPU_INTEN.7	EPU_INTSTS.7		
			能量累加器 6 溢出	EPU_INTEN.6	EPU_INTSTS.6		
			能量累加器 5 溢出	EPU_INTEN.5	EPU_INTSTS.5		
			能量累加器 4 溢出	EPU_INTEN.4	EPU_INTSTS.4		
			能量累加器 3 溢出	EPU_INTEN.3	EPU_INTSTS.3		
			能量累加器 2 溢出	EPU_INTEN.2	EPU_INTSTS.2		
			能量累加器 1 溢出	EPU_INTEN.1	EPU_INTSTS.1		V
			能量累加器 0 溢出	EPU_INTEN.0	EPU_INTSTS.0		V
RNG	000000 BCh	31	冯诺依曼错误	RNG_IMR.3	RNG_ISR.3		
			连续随机数生成错误	RNG_IMR.2	RNG_ISR.2		
			自相关错误	RNG_IMR.1	RNG_ISR.1		
			数据有效	RNG_IMR.0	RNG_ISR.0		

24 MISC 控制器

24.1 简介

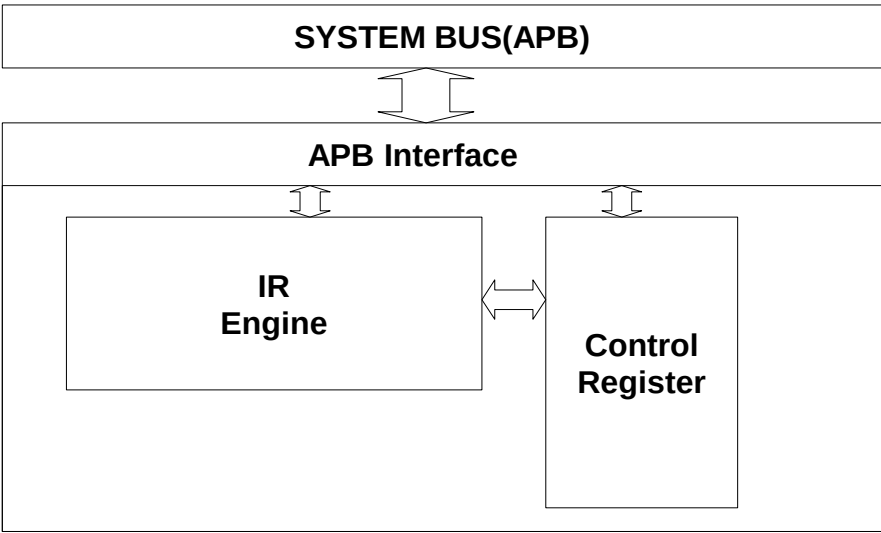
MISC 控制器用于控制 V85XX 的某些特殊功能。V85XX 内置两个 MISC 控制器，一个在核心域（MISC 控制器），在睡眠模式下会掉电。从睡眠模式唤醒后，MISC 控制器的设置会被复位，在从睡眠状态唤醒后，用户应手动恢复设置。MISC2 从睡眠模式唤醒后不会被复位。

24.2 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。

24.3 功能框图

图62. MISC 控制器的功能框图



24.4 寄存器地址

表298. MISC 控制器（MISC 基地址：0x40013000）

名称	类型	地址	描述	默认值



MISC_SRAMINT	rc_w1	0x0000	SRAM 奇偶校验错误中断	0x00
MISC_SRAMINIT	rw	0x0004	SRAM 初始化寄存器	0x01
MISC_PARERR	r	0x0008	SRAM 奇偶校验错误地址寄存器	0x000
MISC_IREN	rw	0x000C	IR 控制寄存器	0x00
MISC_DUTYL	rw	0x0010	IR 调制波低电平宽度控制寄存器	0x0000
MISC_DUTYH	rw	0x0014	IR 调制波高电平宽度控制寄存器	0x0000
MISC_IRQLAT	rw	0x0018	Cortex-M0 IRQ 延迟控制寄存器	0x000
MISC_HIADDR	r	0x0020	AHB 无效访问地址	--
MISC_PIADDR	r	0x0024	APB 无效访问地址	--

表299. MISC2 控制器（MISC2 基地址：0x40013E00）

名称	类型	地址	描述	默认值
MISC2_FLASHWC	rw	0x0000	FLASH 等待周期寄存器	0x2100
MISC2_CLKSEL	rw	0x0004	时钟选择寄存器	0x0
MISC2_CLKDIVH	rw	0x0008	AHB 时钟分频控制寄存器	0x00
MISC2_CLKDIVP	rw	0x000C	APB 时钟分频控制寄存器	0x01
MISC2_HCLKEN	rw	0x0010	AHB 时钟使能控制寄存器	0x1FF
MISC2_PCLKEN	rw	0x0014	APB 时钟使能控制寄存器	0xF83FFFFFFF
MISC2_EPUCLKCTRL	rw	0x0020	EPU 时钟控制寄存器	0x0
MISC2_RSTGPU	rw	0x0024	EPU 复位寄存器	0x0

24.5 寄存器定义

24.5.1 MISC_SRAMINT 寄存器

表300. MISC_SRAMINT 寄存器

位	名称	类型	描述	默认值
31: 5	Rsvd	-	保留	0
4	LOCKUP	rc_w1	该位表示发生了 CM0 锁定，写 1 清除该标志	0x0
3	PIAC	rc_w1	该位表示进行了 APB 总线上的无效地址访问。同时，无效访问地址会被锁存到 MISC_PIADDR。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	rc_w1	该位表示进行了 AHB 总线上的无效地址访问。同时，无效访问地址会被锁存到 MISC_HIADDR。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	Rsvd	-	保留	0
0	PERR	rc_w1	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

24.5.2 MISC_SRAMINIT 寄存器

表301. MISC_SRAMINIT 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7	LOCKIE	rw	CM0 发生锁定时，使能 NMI	0x0
6	PIACIE	rw	发生 APB 无效地址访问，使能 NMI	0x0
5	HIACIE	rw	发生 AHB 无效地址访问，使能 NMI	0x0
4: 2	Rsvd	-	保留	0
1	PERRIE	rw	SRAM 奇偶校验错误时，使能 NMI	0x0
0	Rsvd	-	保留	0x1

24.5.3 MISC_PARERR 寄存器

表302. MISC_PARERR 寄存器

位	名称	类型	描述	默认值
31: 13	Rsvd	-	保留	0
12: 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	--

24.5.4 MISC_IREN 寄存器

表303. MISC_IREN 寄存器

位	名称	类型	描述	默认值
31: 6	Rsvd	-	保留	0
5: 0	IREN	rw	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

24.5.5 MISC_DUTYL 寄存器

表304. MISC_DUTYL 寄存器

位	名称	类型	描述	默认值

31: 16	Rsvd	-	保留	0
15: 0	DUTYL	rw	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 (BUTYL+1) *APBCLK 周期。	0x0000

24.5.6 MISC_DUTYH 寄存器

表305. MISC_DUTYH 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15: 0	DUTYH	rw	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

24.5.7 MISC_IRQLAT 寄存器

表306. MISC_IRQLAT 寄存器

位	名称	类型	描述	默认值
31: 10	Rsvd	-	保留	0
9	NOHARDFULT	rw	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	rw	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	rw	该寄存器用于控制 Cortex-M0 IRQ 响应延迟。	0x00

24.5.8 MISC_HIADDR 寄存器

表307. MISC_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= 0x40000000 + HIADDR	--

24.5.9 MISC_PIADDR 寄存器

表308. MISC_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= 0x40010000 + PIADDR	--

24.5.10 MISC2_FLASHWC 寄存器

表309. MISC2_FLASHWC 寄存器

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

24.5.11 MISC2_CLKSEL 寄存器

表310. MISC2_CLKSEL 寄存器

位	名称	类型	描述	默认值
31: 3	Rsvd	-	保留	0
2: 0	CLKSEL	rw	该寄存器用于控制 AHB 时钟源。 0: RCH (6.5MRC)。 1: XOH (6.5MXTAL)。 2: PLLH。 3: RTCCLK (由 PMU_CONTROL 寄存器中的 RTCCLK_SEL 控制)。	0x0

			4: PLLL。 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。	

24.5.12 MISC2_CLKDIVH 寄存器

表311. MISC2_CLKDIVH 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7: 0	CLKDIVH	rw	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

24.5.13 MISC2_CLKDIVP 寄存器

表312. MISC2_CLKDIVP 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7: 0	CLKDIVP	rw	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

24.5.14 MISC2_HCLKEN 寄存器

表313. MISC2_HCLKEN 寄存器

位	名称	类型	描述	默认值
31: 9	Rsvd	-	保留	0
8: 0	HCLKEN	rw	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314. HCLK 时钟使能

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

24.5.15 MISC2_PCLKEN 寄存器

表315. MISC2_PCLKEN 寄存器

位	名称	类型	描述	默认值
31: 0	PCLKEN	rw	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。	0xF83FFFFFF

表316. PCLK 时钟使能

位	模块	注意事项
0	AHB2APB Bridge	禁止写 0
1	DMA 控制器	
2	I2C	
3	SPI1	
4	UART0	
5	UART1	
6	UART2	
7	UART3	
8	UART4	
9	UART5	
10	ISO78160	
11	ISO78161	
12	Timer	
13	MISC	
14	MISC2	
15	PMU	
16	RTC	

17	ANA	
18	U32K0	
19	U32K1	
20	保留	
21	SPI2	
22	EPU	
23	RNG	
24	SPI4	
25	保留	
26	SPI3	
31: 27	保留	

24.5.16 MISC2_EPUCLKCTRL 寄存器

表317. MISC2_EPUCLKCTRL 寄存器

位	名称	类型	描述	默认值
31: 5	Rsvd	-	保留	0
4	EPUCLKEN	rw	EPU 时钟控制 0: 关闭; 1: 开启。	0x0
3: 2	Rsvd	-	保留	0
1: 0	CLKDIVEPU	rw	EPU 时钟分频控制 0: 时钟源除以 1 1: 时钟源除以 2 2: 时钟源除以 3 3: 时钟源除以 4 注意: EPU 工作时钟是 6.5536MHz。	0x0

24.5.17 MISC2_RSTGPU 寄存器

表318. MISC2_RSTGPU 寄存器

位	名称	类型	描述	默认值
31: 23	Rsvd	-	保留	0
22	EPURST	rw	EPU 复位控制 由软件置 1 和清零。 1: 复位 EPU 0: 不复位 EPU	0x0
21: 0	Rsvd	-	保留	0

25 CRYPT 控制器

25.1 简介

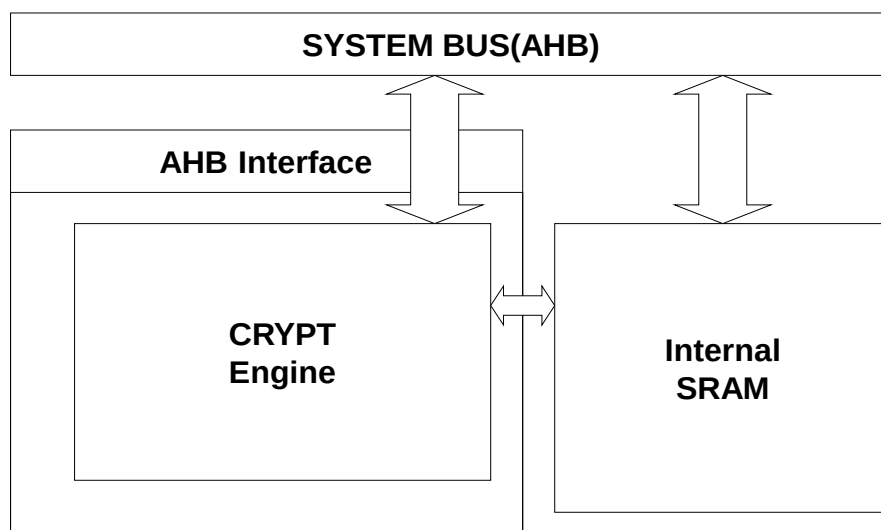
CRYPT 控制器用来给 ECC 加速。CRYPT 控制器的主要特征是实现 VLI 可变长度整数乘法、加法（带进位）、减法（带借位）、乘法加法减法取模和移位。它比纯软件实现的处理能力提高 12 倍。凡是用到 VLI 的加密算法都可以用该硬件加速处理，提升处理速度。CRYPT 控制器能直接访问 SRAM，而不会影响 M0 处理器核的访问。

25.2 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
- 支持乘法、加法和减法的取模计算
- 支持 32 字节的数据搬运

25.3 功能框图

图63. CRYPT 控制器的功能框图



25.4 寄存器地址

表319. CRYPT 控制器 (MISC 基地址: 0x40006000)

名称	类型	地址	描述	默认值
CRYPT_CTRL	rw	0x0000	CRYPT 控制寄存器	0x0000
CRYPT_PTRA	rw	0x0004	CRYPT 指针 A	0x0000
CRYPT_PTRB	rw	0x0008	CRYPT 指针 B	0x0000
CRYPT_PTRO	rw	0x000C	CRYPT 指针 O	0x0000
CRYPT_CARRY	r	0x0010	CRYPT 借位/进位寄存器	0x0000
CRYPT_PRIME0	rw	0x0020	模运算的素数寄存器 0	0xFFFFFFFF
CRYPT_PRIME1	rw	0x0024	模运算的素数寄存器 0	0xFFFFFFFF
CRYPT_PRIME2	rw	0x0028	模运算的素数寄存器 0	0xFFFFFFFF
CRYPT_PRIME3	rw	0x002C	模运算的素数寄存器 0	0x0000
CRYPT_PRIME4	rw	0x0030	模运算的素数寄存器 0	0x0000
CRYPT_PRIME5	rw	0x0034	模运算的素数寄存器 0	0x0000
CRYPT_PRIME6	rw	0x0038	模运算的素数寄存器 0	0x00000001
CRYPT_PRIME7	rw	0x003C	模运算的素数寄存器 0	0xFFFFFFFF

25.5 寄存器定义

25.5.1 CRYPT_CTRL 寄存器

表320. CRYPT_CTRL 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15	NOSTOP	rw	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	Rsvd	-	保留	0
11: 8	LENGTH	rw	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	0x0

			15: 512 位	
7	Rsvd	-	保留	0
6: 4	MODE	rw	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, $*PTRO = *PTRAx * PTRB$ 1: 加法模式, $*PTRO = *PTRAx + PTRB$ 2: 减法模式, $*PTRO = *PTRAx - PTRB$ 3: 逻辑右移一位, $*PTRO = (*PTRAx >> 1)$ 4: 乘法取模模式, $*PTRO = (*PTRAx * PTRB) \text{ MOD } (PRIME_n)$ 5: 加法取模模式, $*PTRO = (*PTRAx + PTRB) \text{ MOD } (PRIME_n)$ 6: 减法取模模式, $*PTRO = (*PTRAx - PTRB) \text{ MOD } (PRIME_n)$ 7: 移动数据模式, $*PTRO = *PTRAx$	0x0
3: 1	Rsvd	-	保留	0
0	ACT	rw	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。 注意：该位写 1 前，关闭所有中断，等该位变成 0 后，软件恢复中断。	0x0

25.5.2 CRYPT_PTRA 寄存器

表321. CRYPT_PTRA 寄存器

位	名称	类型	描述	默认值
31: 15	Rsvd	-	保留	0
14: 0	PTRA	rw	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

25.5.3 CRYPT_PTRB 寄存器

表322. CRYPT_PTRB 寄存器

位	名称	类型	描述	默认值
31: 15	Rsvd	-	保留	0
14: 0	PtrB	rw	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

		0]应该始终是 0。	

25.5.4 CRYPT PTRO 寄存器

表323. CRYPT_PTR0 寄存器

位	名称	类型	描述	默认值
31: 15	Rsvd	-	保留	0
14: 0	PTRO	rw	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

25.5.5 CRYPT CARRY 寄存器

表324. CRYPT_CARRY 寄存器

位	名称	类型	描述	默认值
31: 1	Rsvd	-	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

25.5.6 CRYPT PRIMEn (n=0~7) 寄存器

表325. CRYPT PRIMEn 寄存器

位	名称	类型	描述	默认值
31: 0	NUM	rw	取模运算	0x0

例如：

P = 0xFFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF

寄存器配置如下:

Name	Offset	Description
PRIME0	0x20	0xFFFFFFFF
PRIME1	0x24	0xFFFFFFFF
PRIME2	0x28	0xFFFFFFFF
PRIME3	0x2c	0x00000000
PRIME4	0x30	0x00000000
PRIME5	0x34	0x00000000
PRIME6	0x38	0x00000001
PRIME7	0x3c	0xFFFFFFFF

25.6 使用时的注意事项

25.6.1 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	Data
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]

25.6.2 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。

	MULT	ADD	SUB	RSHIFT1
Operation	*PTRO= *PTRAx*PTRB	*PTRO= *PTRAx+*PTRB	*PTRO= *PTRAx-*PTRB	*PTRO= (*PTRAx>>1)
Outputlength	2*input length	Input length	Input length	Input length
CARRYbit	X	0: 输出没有进位 1: 输出有进位	0: *PTRAx>*PTRB 1: *PTRAx<*PTRB	X
LENGTH	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。

26 RNG 控制器

26.1 简介

RNG 能够从数字逻辑中产生和收集真正随机的比特流。RNG 的典型应用是使用批准的确定性随机数进行密钥生成。

26.2 特点

- 包括一个基于数字逆变器链的内部熵源。
- 内置硬件测试，用于自相关和连续随机数生成测试。

26.3 寄存器地址

表326. RNG 控制器（RNG 基地址：0x40016800）

名称	类型	地址	描述	默认值
RNG_IMR	rw	0x0100	中断掩码寄存器	0x0000
RNG_ISR	rw	0x0104	中断状态寄存器	0x0000
RNG_ICR	rw	0x0108	中断清除寄存器	0x0000
RNG_CFG	rw	0x010C	配置寄存器	0x0000
RNG_DATA0	rw	0x0114	RNG 数据寄存器 0	0x0000
RNG_DATA1	rw	0x0118	RNG 数据寄存器 1	0x0000
RNG_DATA2	rw	0x011C	RNG 数据寄存器 2	0x0000
RNG_DATA3	rw	0x0120	RNG 数据寄存器 3	0x0000
RNG_DATA4	rw	0x0124	RNG 数据寄存器 4	0x0000
RNG_DATA5	rw	0x0128	RNG 数据寄存器 5	0x0000
RNG_CMD	rw	0x012C	RNG 源启用寄存器	0x0000
RNG_CNT	rw	0x0130	采样计数器寄存器	0x0000
RNG_BYPASS	rw	0x0138	RNG 测试旁路寄存器	0x0000

26.4 寄存器定义

26.4.1 RNG_IMR 寄存器

表327. RNG_IMR 寄存器

位	名称	类型	描述	默认值
31: 4	Rsvd	-	保留	0
3	VN_ERR_MASK	rw	此位屏蔽来自 RNG_ISR 寄存器的 VN_ERR	0x0

			中断信号。 0: VN_ERR 中断被解除屏蔽。 1: VN_ERR 中断被屏蔽。	
2	CRNGT_ERR_MASK	rw	此位屏蔽来自 RNG_ISR 寄存器的 CRNGT_ERR 中断信号。 0: CRNGT_ERR 中断被解除屏蔽。 1: CRNGT_ERR 中断被屏蔽。	0x0
1	AUTOCORR_ERR_MASK	rw	此位屏蔽来自 RNG_ISR 寄存器的 AUTOCORR_ERR 中断信号。 0: AUTOCORR_ERR 中断被解除屏蔽。 1: AUTOCORR_ERR 中断被屏蔽。	0x0
0	DATA_VALID_MASK	rw	此位屏蔽来自 RNG_ISR 寄存器的 DATA_VALID 中断信号。 0: DATA_VALID 中断被解除屏蔽。 1: DATA_VALID 中断被屏蔽。	0x0

26.4.2 RNG_ISR 寄存器

表328. RNG_ISR 寄存器

位	名称	类型	描述	默认值
31: 4	Rsvd	-	保留	0
3	VN_ERR	r	如果 32 个连续收集的比特相同，即 32 个零或 32 个 1，则发生冯诺依曼错误。 1: 冯诺依曼错误发生。 0: 没有冯诺依曼错误发生。	0x0
2	CRNGT_ERR	r	当设置为 1 时，表示 RNG 测试中的连续随机数生成测试（CRNGT）错误失败。当两个连续的 16 个收集比特的块相等时，就会发生失败。 1: 连续发生随机数生成测试错误。 0: 没有连续的随机数生成测试错误发生。	0x0
1	AUTOCORR_ERR	r	当设置为 1 时，表示自相关测试连续四次失败。设置后，RNG 将停止工作，直到下一次复位。 1: 发生自相关误差。 0: 没有自相关错误发生。	0x0
0	DATA_VALID	r	当 RNG 中收集了 192 位，并且 DATA0~5 寄存器准备好读取时，设置为 1。 1: DATA0~5 寄存器中的数据准备就绪。 0: DATA0~5 寄存器中的数据尚未就绪。	0x0

26.4.3 RNG_ICR 寄存器

表329. RNG_ICR 寄存器

位	名称	类型	描述	默认值
31: 4	Rsvd	-	保留	0
3	VN_ERR	w	写入 1 以清除冯诺依曼错误。	0x0
2	CRNGT_ERR	w	写入 1 以清除连续随机数生成测试 (CRNGT) 错误。	0x0
1	Rsvd	-	保留	0x0
0	DATA_VALID	w	在读取 DATA0~5 寄存器后写入 1, 以清除 DATA_VALID 状态。	0x0

26.4.4 RNG_CFG 寄存器

表330. RNG_CFG 寄存器

位	名称	类型	描述	默认值
31: 2	Rsvd	-	保留	0
1: 0	CHAIN_SEL	rw	选择环形振荡器中反相器的长度。 00: 选择最短的逆变器链长。 01: 选择短逆变器链长。 10: 选择长逆变器链长。 11: 选择最长的逆变器链长。	0x0

26.4.5 RNG_DATA0 寄存器

表331. RNG_DATA0 寄存器

位	名称	类型	描述	默认值
31: 0	DATA	r	随机数据[31: 0]位。	0x0

26.4.6 RNG_DATA1 寄存器

表332. RNG_DATA1 寄存器

位	名称	类型	描述	默认值
31: 0	DATA	r	随机数据[63: 32]位。	0x0

26.4.7 RNG_DATA2 寄存器

表333. RNG_DATA2 寄存器

位	名称	类型	描述	默认值
31: 0	DATA	r	随机数据[95: 64]位。	0x0

26.4.8 RNG_DATA3 寄存器

表334. RNG_DATA3 寄存器

位	名称	类型	描述	默认值
31: 0	DATA	r	随机数据[127: 96]位。	0x0

26.4.9 RNG_DATA4 寄存器

表335. RNG_DATA4 寄存器

位	名称	类型	描述	默认值
31: 0	DATA	r	随机数据[159: 128]位。	0x0

26.4.10 RNG_DATA5 寄存器

表336. RNG_DATA5 寄存器

位	名称	类型	描述	默认值
31: 0	DATA	r	随机数据[191: 160]位。	0x0

26.4.11 RNG_CMD 寄存器

表337. RNG_CMD 寄存器

位	名称	类型	描述	默认值
31: 1	Rsvd	-	保留	0
0	EN	rw	1: 启用 RNG。 0: RNG 被禁用。	0x0

26.4.12 RNG_CNT 寄存器

表338. RNG_CNT 寄存器

位	名称	类型	描述	默认值
31: 0	CNT	rw	设置两个连续环形振荡器采样之间的时钟周期数。	0

26.4.13 RNG_BYBASS 寄存器

表339. RNG_BYBASS 寄存器

位	名称	类型	描述	默认值
31: 4	Rsvd	-	保留	0
3	AUTOCORR_BYPASS	rw	当设置为 1 时，RNG 模块中的自相关测试被绕	0x0

			过。	
2	CRNGT_BYPASS	rw	当设置为 1 时，RNG 模块中的 CRNGT 测试被绕过。	0x0
1	VN_BYPASS	rw	当设置为 1 时，RNG 模块中的冯·诺依曼均衡器被旁路（包括 32 连续位测试）。	0x0
0	Rsvd	-	保留	0

27 EPU 控制器

27.1 简介

EPU 控制器用于常数累加计算。共有 8 个能量累加器，包括两个高速能量累加器和六个低速能量累加器。低速能量累加器累加频率是 50Hz，工作频率为 6.5536MHz。高速能量累加器累加频率是 204.8 KHz 或 32.768 KHz，工作频率为 6.5536 MHz 或者 32.768 KHz。高速能量累加器和低速能量累加器累加源来自常数寄存器。

27.2 特点

- 支持能量累加器防潜
- 支持门限值可配置
- 支持在 sleep 模式下高速能量累加器工作
- 支持高速能量累加器输出 CF
- 支持清除计数器

27.3 功能框图

图64. EPU 控制器的功能框图

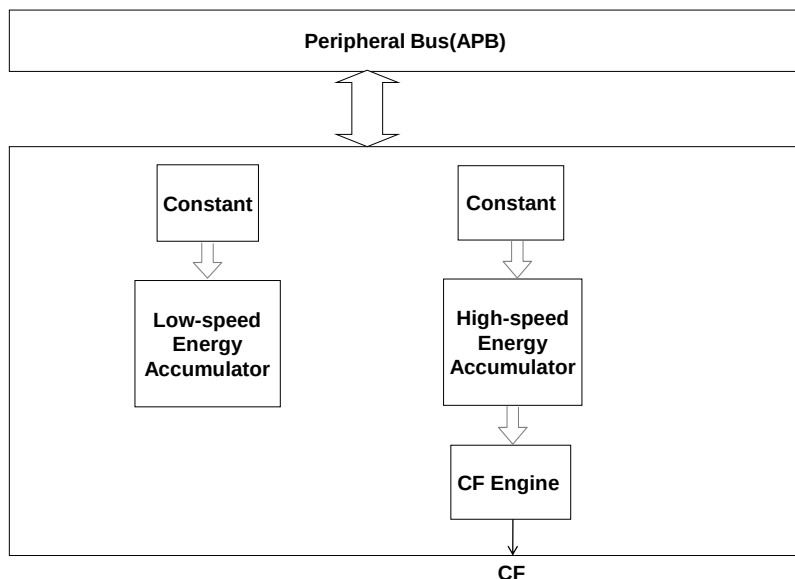


图65. 高速能量累加器的功能框图

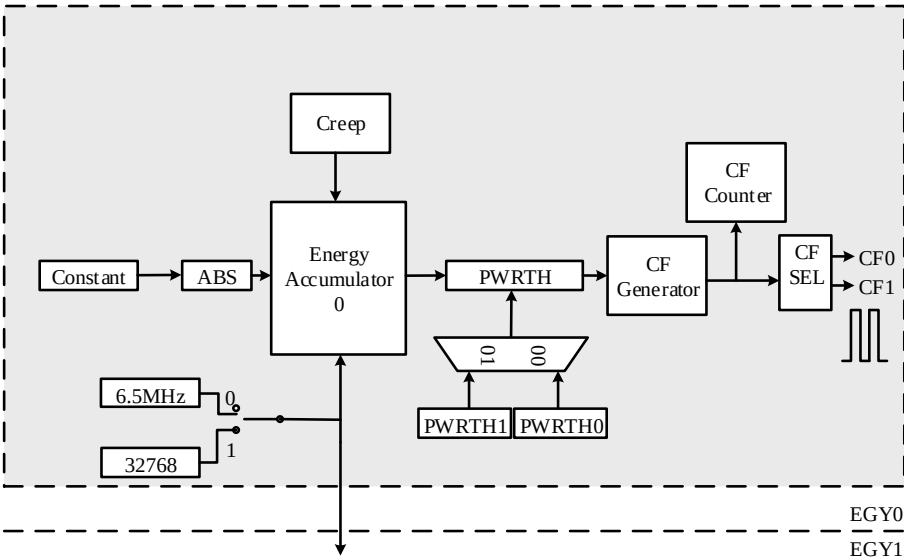
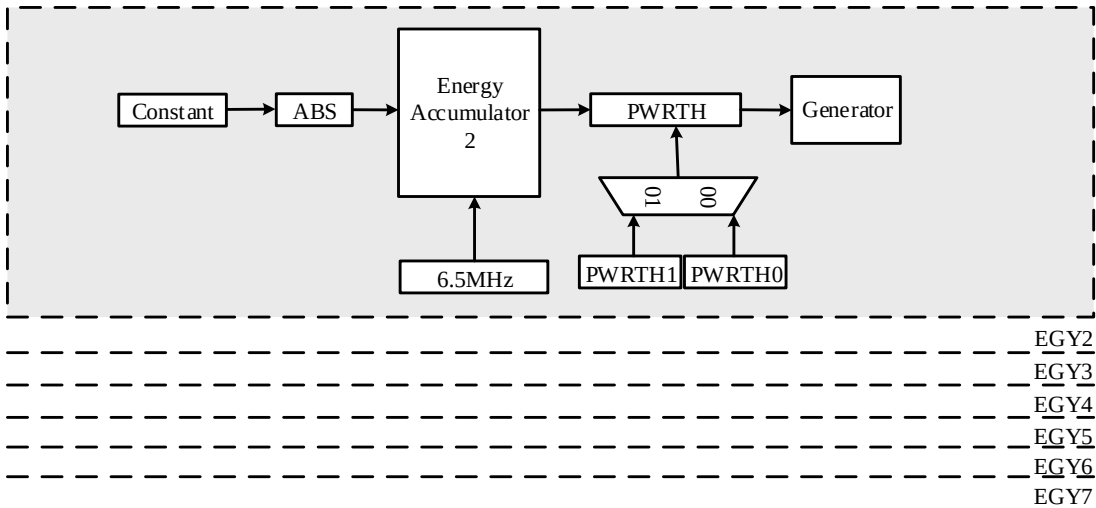


图66. 低速能量累加器的功能框图



27.4 寄存器地址

表340. EPU 控制器（EPU 基地址：0x40015000）

名称	类型	地址	描述	默认值
EGY_CFCTRL	rw	0x0008	CF 输出控制寄存器	0x0
EGY_HSCTRL	rw	0x000C	高速能量累加控制寄存器	0x0606
EGY_LSCTRL	rw	0x 0010	低速能量累加控制寄存器	0x6666
EGY_CLR	rw	0x0014	清除寄存器	0x0
EGY_CRPTH	rw	0x0018	防潜门限值寄存器	0x0
EGY_PWRTH0	rw	0x001C	能量累加器门限值 0	0x0

EGY_PWRTHT1	rw	0x0020	能量累加器门限值 1	0x0
EPU_INTEN	rw	0x0024	中断使能寄存器	0x0
EPU_INTSTS	rc_w1	0x0028	中断状态寄存器	0x0
EPU_STS	r	0x002C	EPU 状态寄存器	0x0
EPU_CONST0	rw	0x0034	能量累加器 0 常数寄存器 (32 位无符号数)	0x0
EPU_OUTL0	r	0x0038	能量累加器 0 累加值低位	0x0
EPU_OUTH0	r	0x003C	能量累加器 0 累加值高位	0x0
EPU_CFCNT0	r	0x0040	能量累加器 0 脉冲计数器	0x0
EPU_CONST1	rw	0x0048	能量累加器 1 常数寄存器 (32 位无符号数)	0x0
EPU_OUTL1	r	0x004C	能量累加器 1 累加值低位	0x0
EPU_OUTH1	r	0x0050	能量累加器 1 累加值高位	0x0
EPU_CFCNT1	r	0x0054	能量累加器 1 脉冲计数器	0x0
EPU_CONSTx (x=2-7)	rw	0x0070+ (x-2)*12	能量累加器 x 的常数寄存器 (32 位无符号数)	0x0
EPU_OUTx (x=2-7)	r	0x0074+ (x-2)*12	能量累加器 x 的累加值	0x0
EPU_CFCNTx (x=2-7)	r	0x0078+ (x-2)*12	能量累加器 x 的脉冲计数器	0x0

27.5 寄存器定义

27.5.1 EPU_CFCTRL 寄存器

表341. EPU_CFCTRL 寄存器

位	名称	类型	描述	默认值
31: 11	Rsvd	-	保留	0
10	CF1INV	rw	CF1 极性控制位: 0: 原始极性。 1: 反向极性。	0x0
9	CF1EN	rw	CF1 输出使能控制: 0: 关闭。 1: 开启。	0x0
8	CF1SEL	rw	CF1 输入源选择: 0: 来源于能量累加器 0。 1: 来源于能量累加器 1。	0x0
7	Rsvd	-	保留	0
6	CF0INV	rw	CF0 极性控制位: 0: 原始极性。 1: 反向极性。	0x0

5	CF0EN	rw	CF0 输出使能位： 0：关闭。 1：开启。	0x0
4	CF0SEL	rw	CF0 输入源选择： 0：来源于能量累加器 0。 1：来源于能量累加器 1。	0x0
3: 2	CFPLUSE	rw	CF 脉冲宽度选择： 0：80 ms 1：40 ms 2：20 ms 3：10 ms	0x0
1: 0	CFFASTEN	rw	CF 脉冲加速控制： 0：1 倍 1：4 倍 2：8 倍 3：16 倍	0x0

27.5.2 EPU_HSCTRL 寄存器

表342. EPU_HSCTRL 寄存器

位	名称	类型	描述	默认值
31: 18	Rsvd	-	保留	0
17	CRPEN	rw	高速能量累加器的防潜开关： 0：关闭 1：打开 32768 时钟下不支持潜动。	0x0
16	CLKSEL	rw	高速能量累加器时钟源选择： 0：6.5 MHz 1：32768 Hz 时钟切换流程为：关闭能量累加器开关—>等待 CF 输出完成—>切换时钟源—>打开能量累加器开关。	0x0
15: 13	Rsvd	-	保留	0
12	EGY1DATASEL	rw	能量累加器 1 的数据源开关： 0：关闭 1：开启	0x0
11	EGY1PWRTSEL	rw	能量累加器 1 的门限值选择： 0：EGY_PWRT0 1：EGY_PWRT1	0x0
10: 9	Rsvd	-	保留，必须配置 3。	3
8	EGY1EN	rw	能量累加器 1 开关 0：关闭	0x0

			1: 开启	
7: 5	Rsvd	-	保留	0
4	EGY0DATASEL	rw	能量累加器 0 的数据源开关: 0: 关闭 1: 开启	0x0
3	EGY0PWRTTHSEL	rw	能量累加器 0 门限值选择: 0: EGY_PWRTTH0 1: EGY_PWRTTH1	0x0
2: 1	Rsvd	-	保留, 必须配置 3。	3
0	EGY0EN	rw	能量累加器 0 开关: 0: 关闭 1: 开启	0x0

27.5.3 EPU_LSCTRL 寄存器

表343. EPU_LSCTRL 寄存器

位	名称	类型	描述	默认值
31: 24	Rsvd	-	保留	0
23	EGY7PWRTTHSEL	rw	能量累加器 7 门限值选择: 0: EGY_PWRTTH0。 1: EGY_PWRTTH1。	0x0
22: 21	Rsvd	-	保留, 必须配置 3。	3
20	EGY7EN	rw	能量累加器 7 使能: 0: 关闭。 1: 开启。	0x0
19	EGY6PWRTTHSEL	rw	能量累加器 6 门限值选择: 0: EGY_PWRTTH0。 1: EGY_PWRTTH1。	0x0
18: 17	Rsvd	-	保留, 必须配置 3。	3
16	EGY6EN	rw	能量累加器 6 使能: 0: 关闭。 1: 开启。	0x0
15	EGY5PWRTTHSEL	rw	能量累加器 5 门限值选择: 0: EGY_PWRTTH0。 1: EGY_PWRTTH1。	0x0
14: 13	Rsvd	-	保留, 必须配置 3。	3
12	EGY5EN	rw	能量累加器 5 使能: 0: 关闭。 1: 开启。	0x0
11	EGY4PWRTTHSEL	rw	能量累加器 4 门限值选择: 0: EGY_PWRTTH0。 1: EGY_PWRTTH1。	0x0

10: 9	Rsvd	-	保留, 必须配置 3。	3
8	EGY4EN	rw	能量累加器 4 使能: 0: 关闭。 1: 开启。	0x0
7	EGY3PWRTSEL	rw	能量累加器 3 门限值选择: 0: EGY_PWRTSEL0。 1: EGY_PWRTSEL1。	0x0
6: 5	Rsvd	-	保留, 必须配置 3。	3
4	EGY3EN	rw	能量累加器 3 使能: 0: 关闭。 1: 开启。	0x0
3	EGY2PWRTSEL	rw	能量累加器 2 门限值选择: 0: EGY_PWRTSEL0。 1: EGY_PWRTSEL1。	0x0
2: 1	Rsvd	-	保留, 必须配置 3。	3
0	EGY2EN	rw	能量累加器 2 使能: 0: 关闭。 1: 开启。	0x0

27.5.4 EPU_CLR 寄存器

表344. EPU_CLR 寄存器

位	名称	类型	描述	默认值
31: 16	Rsvd	-	保留	0
15	EGY7OUTCLR	rc_w1	写 1 清除 EPU_OUT7。	0x0
14	EGY6OUTCLR	rc_w1	写 1 清除 EPU_OUT6。	0x0
13	EGY5OUTCLR	rc_w1	写 1 清除 EPU_OUT5。	0x0
12	EGY4OUTCLR	rc_w1	写 1 清除 EPU_OUT4。	0x0
11	EGY3OUTCLR	rc_w1	写 1 清除 EPU_OUT3。	0x0
10	EGY2OUTCLR	rc_w1	写 1 清除 EPU_OUT2。	0x0
9	EGY1OUTCLR	rc_w1	写 1 清除 EPU_OUT1 和 EPU_OUTL1。	0x0
8	EGY0OUTCLR	rc_w1	写 1 清除 EPU_OUT0 和 EPU_OUTL0。	0x0
7	EGY7CFCNTCLR	rc_w1	写 1 清除 EPU_CNT7。	0x0
6	EGY6CFCNTCLR	rc_w1	写 1 清除 EPU_CNT6。	0x0
5	EGY5CFCNTCLR	rc_w1	写 1 清除 EPU_CNT5。	0x0
4	EGY4CFCNTCLR	rc_w1	写 1 清除 EPU_CNT4。	0x0
3	EGY3CFCNTCLR	rc_w1	写 1 清除 EPU_CNT3。	0x0
2	EGY2CFCNTCLR	rc_w1	写 1 清除 EPU_CNT2。	0x0
1	EGY1CFCNTCLR	rc_w1	写 1 清除 EPU_CNT1。	0x0
0	EGY0CFCNTCLR	rc_w1	写 1 清除 EPU_CNT0。	0x0

27.5.5 EGY_CRPTH 寄存器

表345. EGY_CRPTH 寄存器

位	名称	类型	描述	默认值
31: 0	CRPTH	rw	能量累加器潜动门限值。(用于判断能量累加器是否处于潜动状态。当防潜累加值超过 EGY_CRPTH 并且高速能量累加器累加值未超过 EGY_PWRTH 时, 高速能量累加器的累加值会被清掉; 反之, 防潜累加值被清掉。)	0x0

27.5.6 EGY_PWRTH0 寄存器

表346. EGY_PWRTH0 寄存器

位	名称	类型	描述	默认值
31: 0	PWRTH	rw	能量累加器溢出门限值 0。由于能量累加器为 44 位, 实际高速能量累加器累加阈值为该阈值 *4096; 低速能量累加器累加阈值为该阈值*1。	0x0

27.5.7 EGY_PWRTH1 寄存器

表347. EGY_PWRTH1 寄存器

位	名称	类型	描述	默认值
31: 0	PWRTH		能量累加器溢出门限值 1。由于能量累加器为 44 位, 实际高速能量累加器累加阈值为该阈值 *4096; 低速能量累加器累加阈值为该阈值*1。	0

27.5.8 EPU_INTEN 寄存器

表348. EPU_INTEN 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7	EGY7OVIE	rw	能量累加器 7 溢出中断使能。	0x0
6	EGY6OVIE	rw	能量累加器 6 溢出中断使能。	0x0
5	EGY5OVIE	rw	能量累加器 5 溢出中断使能。	0x0
4	EGY4OVIE	rw	能量累加器 4 溢出中断使能。	0x0
3	EGY3OVIE	rw	能量累加器 3 溢出中断使能。	0x0
2	EGY2OVIE	rw	能量累加器 2 溢出中断使能。	0x0
1	EGY1OVIE	rw	能量累加器 1 溢出中断使能。	0x0
0	EGY0OVIE	rw	能量累加器 0 溢出中断使能。	0x0

27.5.9 EPU_INTSTS 寄存器

表349. EPU_INTSTS 寄存器

位	名称	类型	描述	默认值
31: 8	Rsvd	-	保留	0
7	EGY7OVIF	rc_w1	能量累加器 7 溢出中断标志, 写 1 清 0。	0x0
6	EGY6OVIF	rc_w1	能量累加器 6 溢出中断标志, 写 1 清 0。	0x0
5	EGY5OVIF	rc_w1	能量累加器 5 溢出中断标志, 写 1 清 0。	0x0
4	EGY4OVIF	rc_w1	能量累加器 4 溢出中断标志, 写 1 清 0。	0x0
3	EGY3OVIF	rc_w1	能量累加器 3 溢出中断标志, 写 1 清 0。	0x0
2	EGY2OVIF	rc_w1	能量累加器 2 溢出中断标志, 写 1 清 0。	0x0
1	EGY1OVIF	rc_w1	能量累加器 1 溢出中断标志, 写 1 清 0。	0x0
0	EGY0OVIF	rc_w1	能量累加器 0 溢出中断标志, 写 1 清 0。	0x0

27.5.10 EPU_STS 寄存器

表350. EPU_STS 寄存器

位	名称	类型	描述	默认值
31: 2	Rsvd	-	保留	0
1	EGY1CRP	r	能量累加器 1 防潜状态 0: 启动状态。 1: 潜动状态。	0x0
0	EGY0CRP	r	能量累加器 0 防潜状态 0: 启动状态。 1: 潜动状态。	0x0

27.5.11 EPU_CONST0 寄存器

表351. EPU_CONST0 寄存器

位	名称	类型	描述	默认值
31: 0	CONST0	rw	能量累加器 0 累加常数。当 EGY0DATASEL=1 时, 能量累加器 0 累加源来自于 CONST0。	0x0

27.5.12 EPU_OUTL0 寄存器

表352. EPU_OUTL0 寄存器

位	名称	类型	描述	默认值
31: 0	OUTL0	r	能量累加器 0 累加值低位。	0x0

27.5.13 EPU_OUTH0 寄存器

表353. EPU_OUTH0 寄存器

位	名称	类型	描述	默认值
31: 0	OUTH0	r	能量累加器 0 累加值高位，低 13 位有效。	0x0

27.5.14 EPU_CFCNT0 寄存器

表354. EPU_CFCNT0 寄存器

位	名称	类型	描述	默认值
31: 0	CFCNT0	r	能量累加器 0 脉冲计数器。	0x0

27.5.15 EPU_CONST1 寄存器

表355. EPU_CONST1 寄存器

位	名称	类型	描述	默认值
31: 0	CONST1	rw	能量累加器 1 累加常数。当 EGY1DATASEL=1 时，能量累加器 1 累加源来自于 CONST1。	0x0

27.5.16 EPU_OUTL1 寄存器

表356. EPU_OUTL1 寄存器

位	名称	类型	描述	默认值
31: 0	OUTL1	r	能量累加器 1 累加值低位。	0x0

27.5.17 EPU_OUTH1 寄存器

表357. EPU_OUTH1 寄存器

位	名称	类型	描述	默认值
31: 0	OUTH1	r	能量累加器 1 累加值高位，低 13 位有效。	0x0

27.5.18 EPU_CFCNT1 寄存器

表358. EPU_CFCNT1 寄存器

位	名称	类型	描述	默认值
31: 0	CFCNT1	r	能量累加器 1 脉冲计数器。	0x0

27.5.19 EPU_CONSTx (x=2-7) 寄存器

表359. EPU_CONSTx (x=2-7) 寄存器

位	名称	类型	描述	默认值
31: 0	CONSTx	rw	能量累加器 x 的累加常数。	0x0

27.5.20 EPU_OUTx (x=2-7) 寄存器

表360. EPU_OUTx (x=2-7) 寄存器

位	名称	类型	描述	默认值
31: 0	OUTx	r	能量累加器 x 的累加值。	0

27.5.21 EPU_CFCNTx (x=2-7) 寄存器

表361. EPU_CFCNTx (x=2-7) 寄存器

位	名称	类型	描述	默认值
31: 0	CFCNTx	r	能量累加器 x 的脉冲计数器。	0x0

28 调试特点

V85XX 使用的是 Cortex-M0 内核，包含硬件调试模块。

28.1 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

28.2 SWD 引脚

表362. 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
正常模式 MODE 脚是高电平	SEG54/IOA0	SEG53/IOA1

29 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。

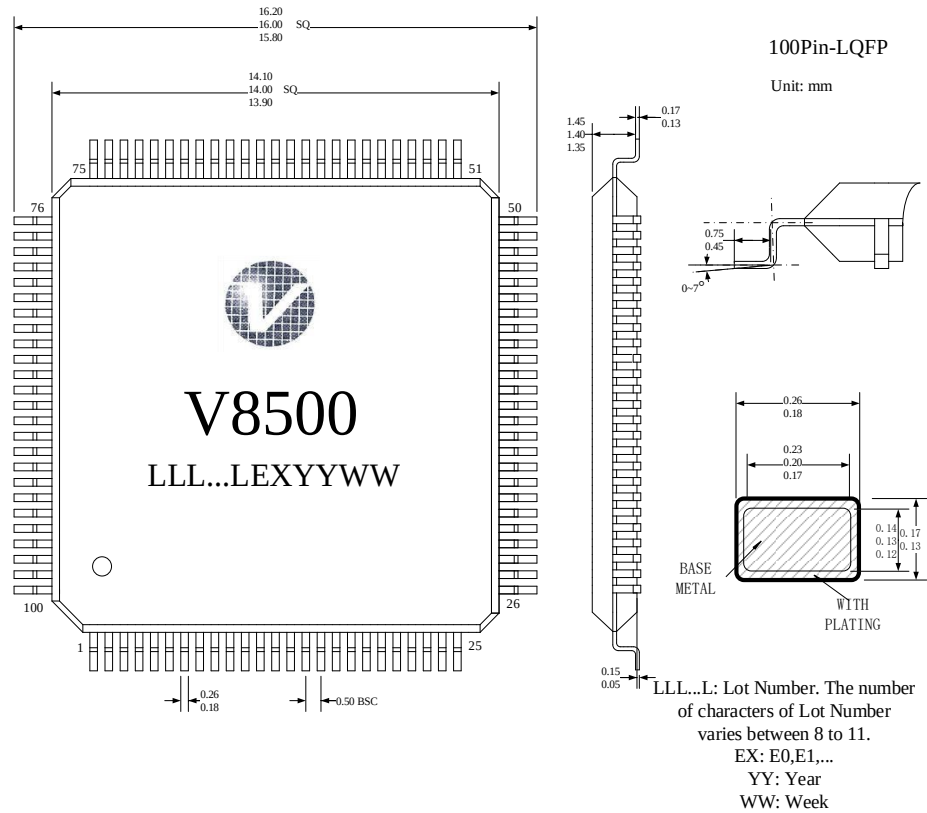
29.1 CMSIS 函数说明

表363. CMSIS 函数

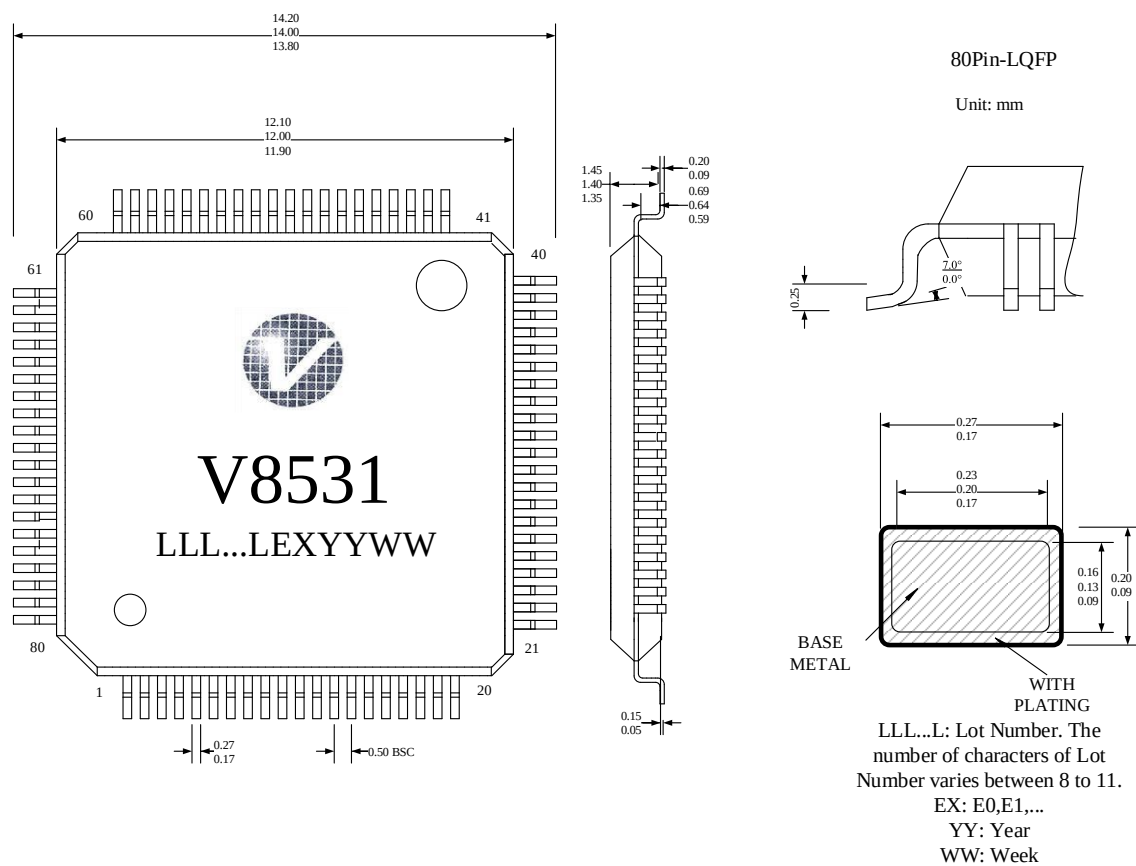
函数	说明
void NVIC_EnableIRQ (IRQn_Type IRQn) ;	使能中断号为 IRQn 的中断，该功能不适用于系统异常的情况。
Void NVIC_DisableIRQ (IRQn_Type IRQn) ;	屏蔽中断号为 IRQn 的中断，该功能不适用于系统异常的情况。
Void VIC_SetPendingIRQ (IRQn_Type IRQn) ;	设置中断号为 IRQn 中断的挂起状态，该功能不适用于系统异常的情况。
Void NVIC_ClearPendingIRQ (IRQn_Type IRQn) ;	清除中断号为 IRQn 中断的挂起状态，该功能不适用于系统异常的情况。
UInt32_t NVIC_GetPendingIRQ (IRQn_Type IRQn) ;	获取中断号为 IRQn 中断的挂起状态
void NVIC_SetPriority (IRQn_Type IRQn, uint32_t priority) ;	设置中断号为 IRQn 中断的优先级
uint32_t NVIC_GetPriority (IRQn_Type IRQn) ;	获取中断号为 IRQn 的中断优先级。
Void __enable_irq (void) ;	使能全局中断。
Void __disable_irq (void) ;	屏蔽全局中断（除了 hardfault 和 NMI）。
UInt32_t SysTick_Config (uint32_t ticks) ;	初始化并启动 SysTick 计数器和它的中断。
Void NVIC_SystemReset (void) ;	M0 内核软复位
SCB->SCR = SCB_SCR_SLEEPDEEP_Msk; void __WFI (void) ;	等待中断（进入睡眠模式）
SCB->SCR &= (uint32_t) ~((uint32_t) SCB_SCR_SLEEPDEEP_Msk) ; void __WFI (void) ;	等待中断（进入 IDLE 模式）

30 封装尺寸图

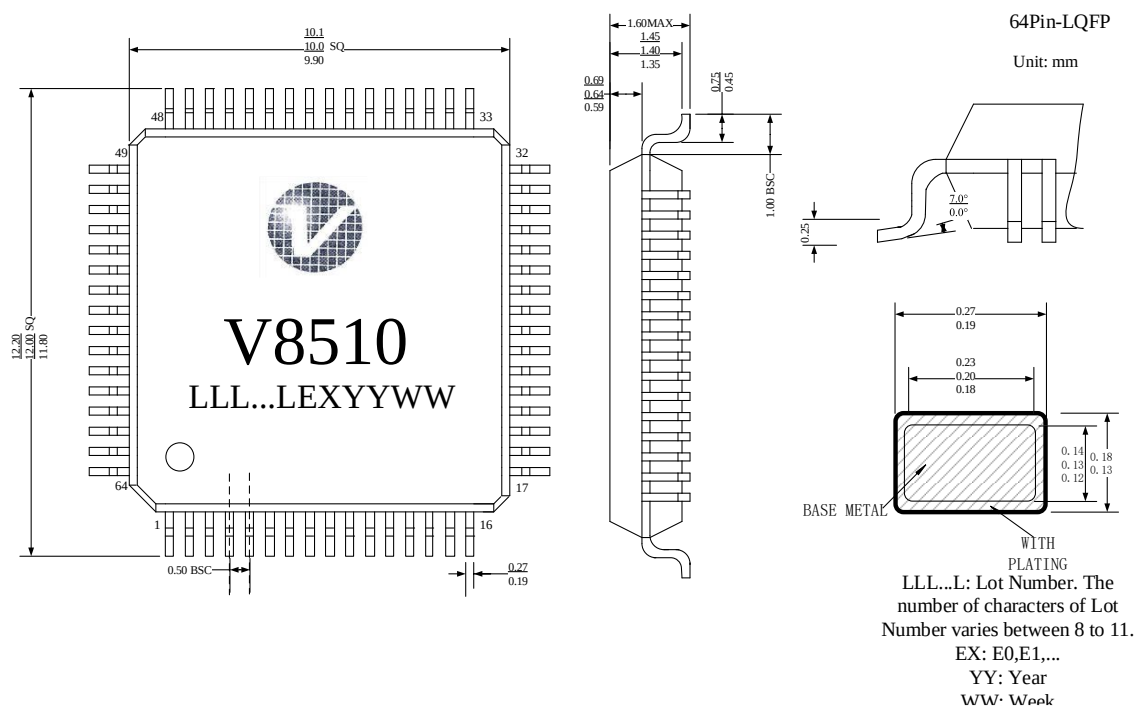
30.1 V8500 封装尺寸图



30.2 V8531 封装尺寸图



30.3 V8510 封装尺寸图



Innovates for the Internet of Energy

杭 州 万 高 科 技 股 份 有 限 公 司

地址:杭州市滨江区六和路 368 号海创基地北楼 A4070室

电话: +86-571-88157065

传真: +86-571-88157067

邮箱:service@vangotech.com

网址:<http://www.vangotech.com>

微 信 公 众 号

