YC3121-e 芯片数据手册

V1.0

Revision History

Version	Date	Author	Description
V1.0	2019-12-23		Initial version

Confidentiality Level:

Confidential

目录

1 芯片简介

1.1 简介

YC3121-e 芯片使用 Cortex-M0 内核处理器，具有卓越的架构、高性能和超低功耗等特性， 提供高性能的及安全数据处理解决方案。

芯片内置硬件安全加密模块， 支持多种加密安全算法，包括 DES、 TDES、 AES、SM2、SM3、SM4、ECC、 RSA、SHA、国密等主流加密算法。 具有多种攻击检测功能，符合金融安全设备标准。

芯片内部包含安全 BOOT 程序， 支持下载、启动时对固件进行 RSA 签名校验。 芯片内置 1MB 安全 Flash、 64KB SRAM 和 8KB OTP 存储区。 同时集成了丰富的外设资源，所有外设驱动软件兼容目前主流安全芯片软件接口并符合 ARM CMSIS 规范， 用户可在现有方案基础上进行快速开发和移植。

1.2 功能特性

◆32-bit Cortex-M0 内核处理器

- ●MPU 保护单元
- ●最高 96MHZ 主频，支持 2，4，8 分频
- ●1 个受控 JTAG 调试口

◆64K 随机加扰 SRAM

◆1MB 可选安全存储 Flash

◆8KB OTP

◆安全加密算法加速引擎

- ●对称算法：DES、TDES、AES-128/192/256、国密 IV（SM4）
- ●非对称算法：RSA-1024/2048、 国密 II（SM2）、ECC
- ●HASH 校验算法 ：SHA-1/224/256/384/512、 国密 III（SM3）

◆1 个 ISO7816 接口，支持 3V、1.8V 供电

◆1 个三轨磁条卡解码模块，支持 ISO/ABA、AAMVA 及 IBM 等标准卡

◆2 个 UART 接口

◆2 个 SPI 接口，1 个 QSPI 接口

◆8 个 32 位 TIMER（支持 PWM）

◆1 个真随机数发生器

◆1 个 IIC 接口

◆6 个 DMA（SPI0、SPI1、UART0、UART1、IIC、MEMCP）

◆1 个 CRC 模块

◆48 个 GPIO

◆最多支持 8 个静态 Tamper 或 4 组动态 Tamper(4 输出，4 输入)， 动/静态可配

◆1 组内部 Sensor（支持高低电压、高低温、 Mesh、时钟和 voltage glitch 等传感器）

◆1 块密钥存储区（32 X 32bit 支持硬件快速擦除）

◆1 个 USB 接口

◆1 个看门狗模块

◆10 bit ADC 模块

- ●HVIN ADC 电压范围：0-5V
- ●VIN ADC 电压范围：2.2-3.3V
- ●GPIO ADC 电压范围：0-1.5V

1.3 应用领域

金融安全设备、移动安全设备及其他对功耗和成本敏感的安全设备。

2 芯片结构

2.1 系统架构

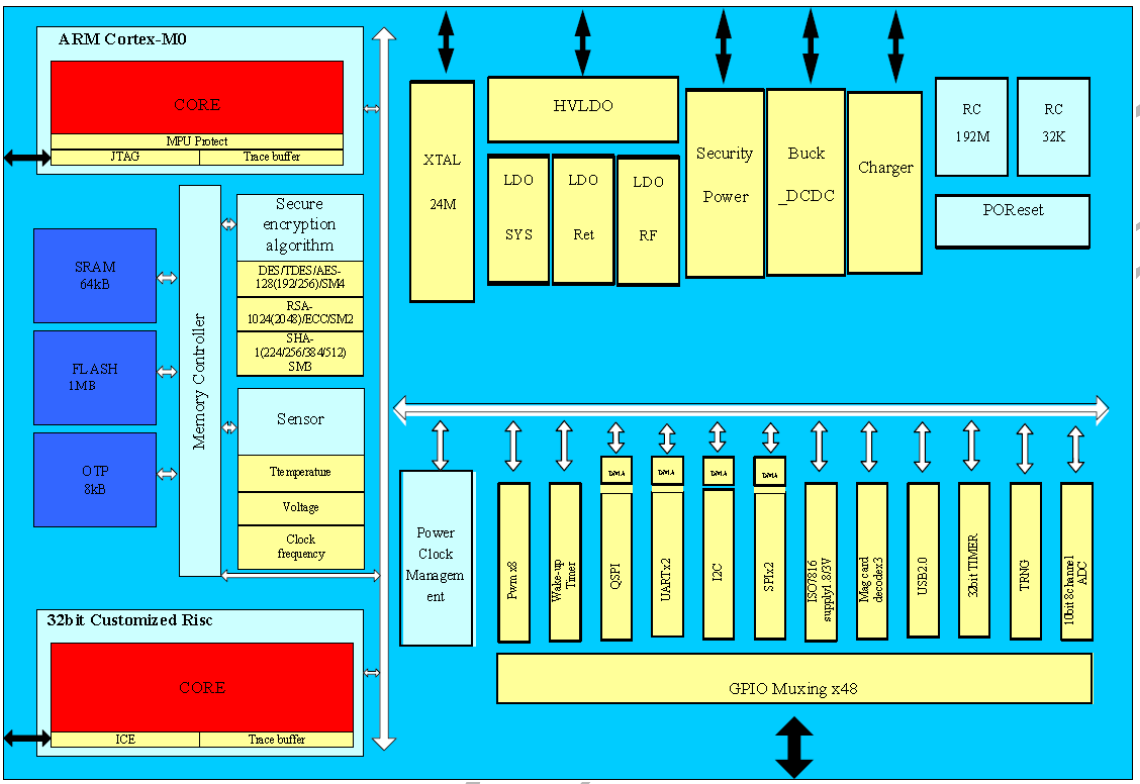


图 1 系统架构图

2.2 存储器映射

2.2.1 存储器地址映射

	0xffffffff
Restricted	0xe00ff000
PPB	0xe0000000
Restricted	0x02000000
Flash(512KB)	0x01000000
Restricted	0x000fa000
Peripherals	0x000f0000
Restricted	0x00030000
RAM(64KB)	0x00020000
Restricted	0x00008000
ROM(32KB)	0x00000000

图 2 存储器映射地址

2.2.2 外设地址映射

下表为总体外设地址映射表，外设地址映射见相应章节。

外设	基地址名称	基地址
WDT	WDT_BASEADDR	0xf0000
SCI7816	SCI7816_BASEADDR	0xf0400
TIMER	TIMER_BASEADDR	0xf0c00
SM4	SM4_BASEADDR	0xf5200
DES	DES_BASEADDR	0xf8000
USB	USB_ BASEADDR	0xf6000
AES	AES_ BASEADDR	0xf8300
CRC	CRC_BASEADDR	0xf8200
RSA	RSA_BASEADDR	0xf5800
SHA	SHA_BASEADDR	0xf8600
TRNG	RNG_BASEADDR	0xf852c
SYSCTRL	SYSCTRL_BASEADDR	0xf8500
MPUCTRL	MPUCTRL_BASEADDR	0xf8580
LPM	LPM_BASEADDR	0xf8400
GPIO	GPIO_BASEADDR	0xf8700
DMA	DMA_BASEADDR	0xf8800
QSPI	QSPI_BASEADDR	0xf8800
SPI	SPI_BASEADDR	0xf891c
UART	UART_ BASEADDR	0xf8b1c
IIC	IIC_ BASEADDR	0xf8d1c
SYSTICK	SYSTICK _BASEADDR	0xE000E010
NVIC	NVIC_BASEADDR	0xe000e100

2.3 嵌入式 RAM

本芯片采用 RAM 数据总线宽度为 36 bits，支持 byte 写操作，RAM 存储区大小为 64K，地址范围为：0x20000 - 0x2FFFF，主要用于存储系统临时数据。

外设	地址范围	容量
RAM	0x0002_0000-0x0002_FFFF	64KB

3 芯片特性

3.1 电气特性

电气特性↓

参数	说明	范围			单位
		Min	Typ	Max	
HVIN	HVLDO 输入	3.5	4.2	5.5	V
VIN	电源电压输入	1.8	3.0	3.6	V
VBAT	纽扣电池输入	1.9	3.0	3.6	V
VIO	GPIO 电源	1.8	3.0	3.6	V
CHGRIN	充电电源输入	4.25	5.0	6.5	V
CHGROUT	charge current be set at 150mA .provides charge to 4.2V				-
IHVLDO	HVLDO 驱动电流	-	-	200	mA
Tamb	工作温度	-40	-	+80	℃
Tstg	储藏温度	-40	-	+125	℃
VSS	地	-0.3	0	+0.3	V
Voh	数字输出高电平	0.7*VIO	VIO	VIO	V
Vol	数字输出低电平	VSS	VSS	0.3*VIO	V
Ioh	拉电流	GPIO[0:7]、GPIO[32:47]:100mA；other:20mA			
Iol	灌电流	GPIO[0:7]、GPIO[32:47]:100mA；other:20mA			
Vih	数字输入高电平	0.7*VIO	VIO	VIO	V
Vil	数字输入低电平	VSS	VSS	0.3*VIO	V

安全特性↓

传感器	说明	范围	单位
温度传感器	高温检测范围	100±10	℃
	低温检测范围	-30~-40	℃
电压传感器	主电源电压高压检测范围	3.7±0.15	V
	主电源电压低压检测范围	1.9±0.15	V
	电池电压高压检测范围	3.7±0.15	V
	电池电压低压检测范围	1.9±0.15	V
外部 Tamper 电阻	Tamper 管脚上拉电阻阻值	1M±10%	Ω

3.2 管脚定义

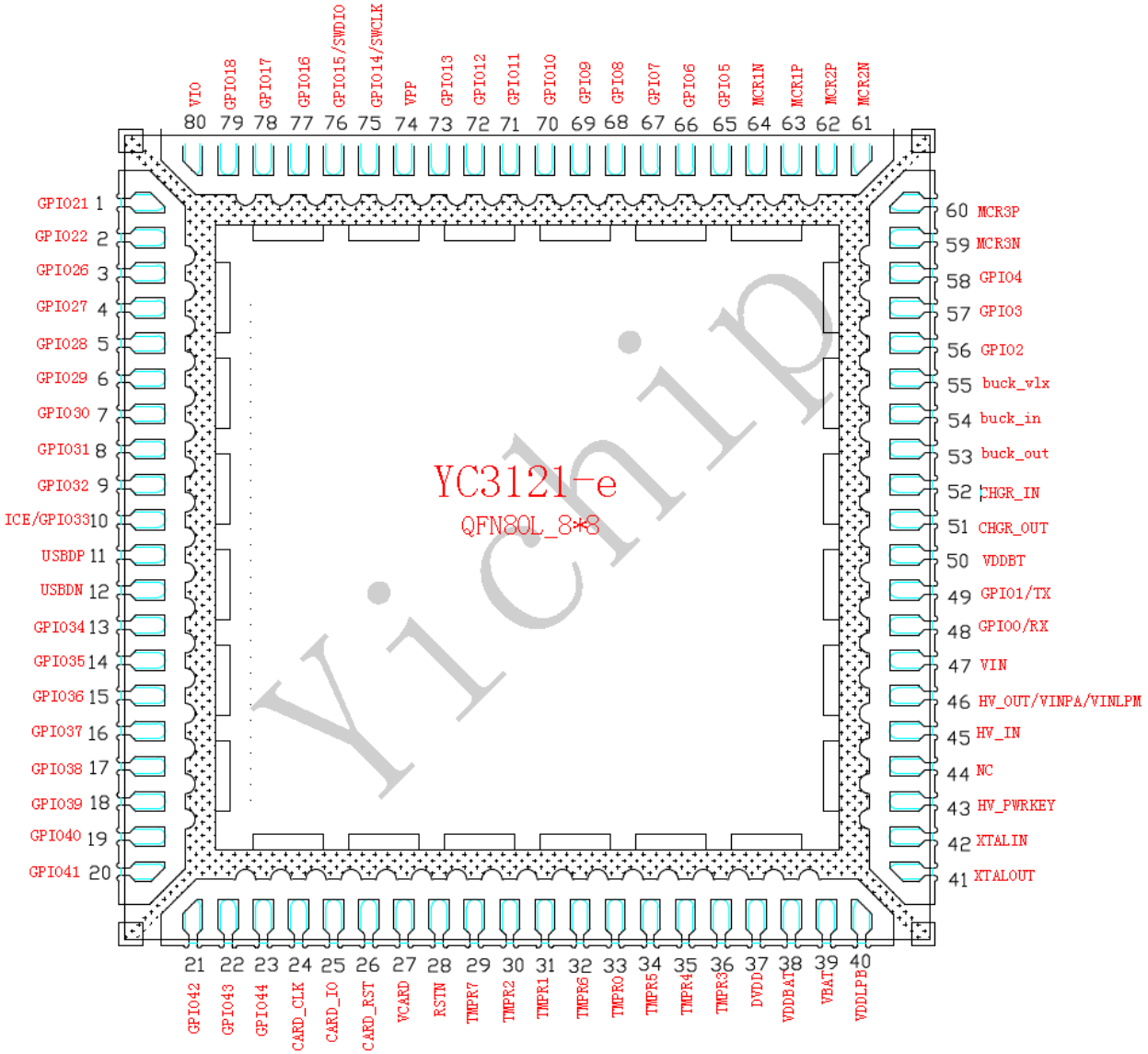


图 3 芯片管脚图

管脚定义列表

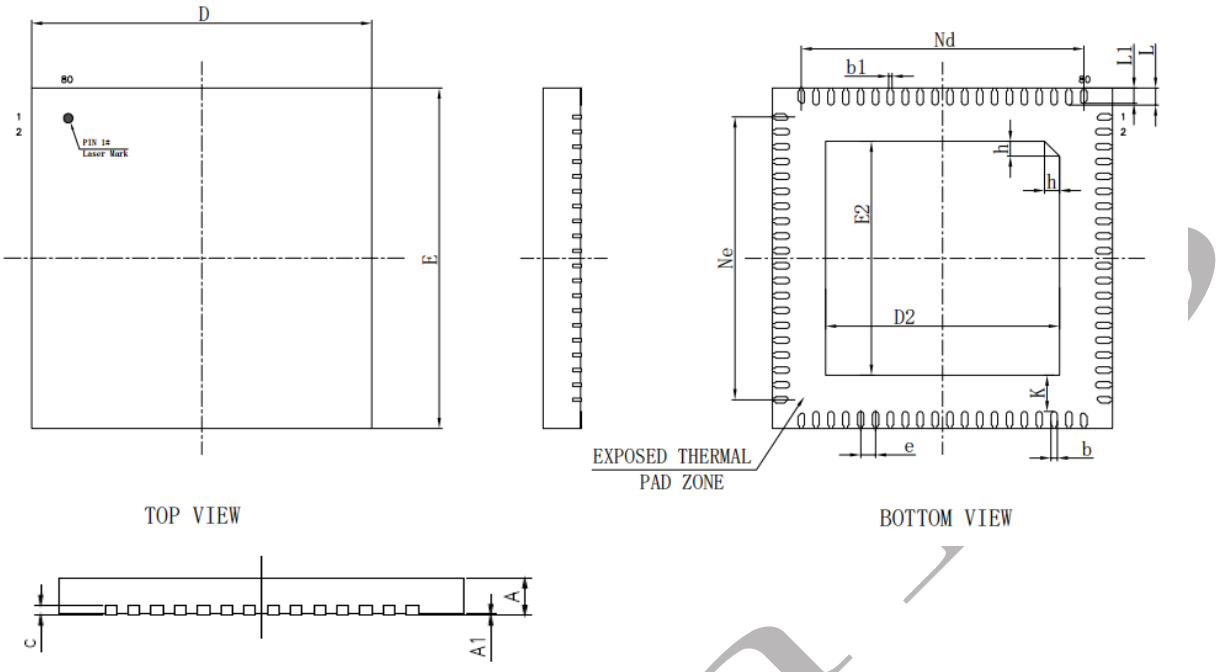
PIN No.	Pad name	说明
1	GPIO21	最大驱动电流 20mA
2	GPIO22	最大驱动电流 20mA
3	GPIO26	最大驱动电流 20mA
4	GPIO27	最大驱动电流 20mA
5	GPIO28	最大驱动电流 20mA
6	GPIO29	最大驱动电流 20mA
7	GPIO30	最大驱动电流 20mA
8	GPIO31	最大驱动电流 20mA
9	GPIO32	最大驱动电流 100mA
10	GPIO33/ICE	默认为 ICE，最大驱动电流 100mA
11	USBDP	

12	USBDN	
13	GPIO34	IO 最大驱动电流 100mA
14	GPIO35	IO 最大驱动电流 100mA
15	GPIO36	IO 最大驱动电流 100mA
16	GPIO37	IO 最大驱动电流 100mA
17	GPIO38	adc_channel2，IO 最大驱动电流 100mA
18	GPIO39	adc_channel3，IO 最大驱动电流 100mA
19	GPIO40	adc_channel4，IO 最大驱动电流 100mA
20	GPIO41	adc_channel5，IO 最大驱动电流 100mA
21	GPIO42	adc_channel6，IO 最大驱动电流 100mA
22	GPIO43	adc_channel7，IO 最大驱动电流 100mA
23	GPIO44	adc_channel8，IO 最大驱动电流 100mA
24	CARD_CLK	7816 clock
25	CARD_IO	7816 IO（data）
26	CARD_RST	7816 reset
27	VCARD	7816 VCC（持 1.8V 和 3.0V）
28	RSTN	芯片 reset，低有效
29	TMPR7	防拆 Tamper 1、 静态，独立使用，外部接地 2、 动态，0-1、2-3、4-5、6-7 直连配对使用
30	TMPR2	
31	TMPR1	
32	TMPR6	
33	TMPR0	
34	TMPR5	
35	TMPR4	
36	TMPR3	
37	DVDD	对应 VIN，主电 LDO 输出接电容
38	VDDBAT	对应 VBAT，纽扣电池 LDO 输出接电容
39	VBAT	纽扣电池供电
40	VDDLPB	对应 VINLPB，蓝牙低功耗 LDO 输出接电容
41	XTALOUT	24MHz 晶体
42	XTALIN	
43	HV_PWRKEY	HVLDO Power Key
44	NC	
45	HV_IN	HVLDO 输入
46	HV_OUT/VINPA/VINLPM	HV_OUT:HVLDO 输出（3.3V）;VINPA:模拟模块电源输入
47	VIN	芯片电源
48	GPIO0/RX	ROM BOOT UART RX，最大驱动电流 100mA
49	GPIO1/TX	ROM BOOT UART TX，最大驱动电流 100mA
50	VDDBT	蓝牙 LDO 输出接电容
51	CHGR_OUT	充电模块输出
52	CHGR_IN	充电模块输入
53	buck_out	
54	buck_in	buck 电源输入（2.2-4.2V）
55	buck_vlx	buck 电源输出（1.5-3V）
56	GPIO2	最大驱动电流 100mA

57	GPIO3	最大驱动电流 100mA
58	GPIO4	最大驱动电流 100mA
59	MCR3N	磁道三 N
60	MCR3P	磁道三 P
61	MCR2N	磁道二 N
62	MCR2P	磁道二 P
63	MCR1P	磁道一 P
64	MCR1N	磁道一 N
65	GPIO5	IO 最大驱动电流 100mA
66	GPIO6	IO 最大驱动电流 100mA
67	GPIO7	IO 最大驱动电流 100mA
68	GPIO8	最大驱动电流 20mA
69	GPIO9	最大驱动电流 20mA
70	GPIO10	最大驱动电流 20mA
71	GPIO11	最大驱动电流 20mA
72	GPIO12	最大驱动电流 20mA
73	GPIO13	最大驱动电流 20mA
74	VPP	写 OTP 电源输入
75	GPIO14/SWCLK	默认 JTAG_SW_CLK，IO 最大驱动电流 20mA
76	GPIO15/SWDIO	默认 JTAG_SW_IO，IO 最大驱动电流 20mA
77	GPIO16	最大驱动电流 20mA
78	GPIO17	最大驱动电流 20mA
79	GPIO18	最大驱动电流 20mA
80	VIO	GPIO 电源

注：所有 GPIO 均可任意配置成 GPIO 模式中的其中一种功能,(ADC channel 除外)。

3.3 封装信息



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.85	0.90	0.95
A1	0	0.02	0.05
b	0.10	0.15	0.20
b1		0.08REF	
c		0.203REF	
D	7.90	8.00	8.10
D2	5.40	5.50	5.60
e		0.35BSC	
Ne		6.65BSC	
Nd		6.65BSC	
E	7.90	8.00	8.10
E2	5.40	5.50	5.60
L	0.35	0.40	0.45
L1	0.31	0.36	0.41
h	0.30	0.35	0.40
K		0.85REF	

图 4 QFN8*8_80L

3.4 系统复位控制

在本芯片中存在以下几种类型的复位源：

1)VBAT 电压域 POR 上电复位；

2) VSEC 电压域 POR 上电复位；

3) Watch dog 溢出复位；

4)外部 RSTN 管脚信号复位；

5)Sensor 检测异常；

6)Software 复位。

说明：各复位功能使能由寄存器（SYSCTRL_RST_EN）控制。

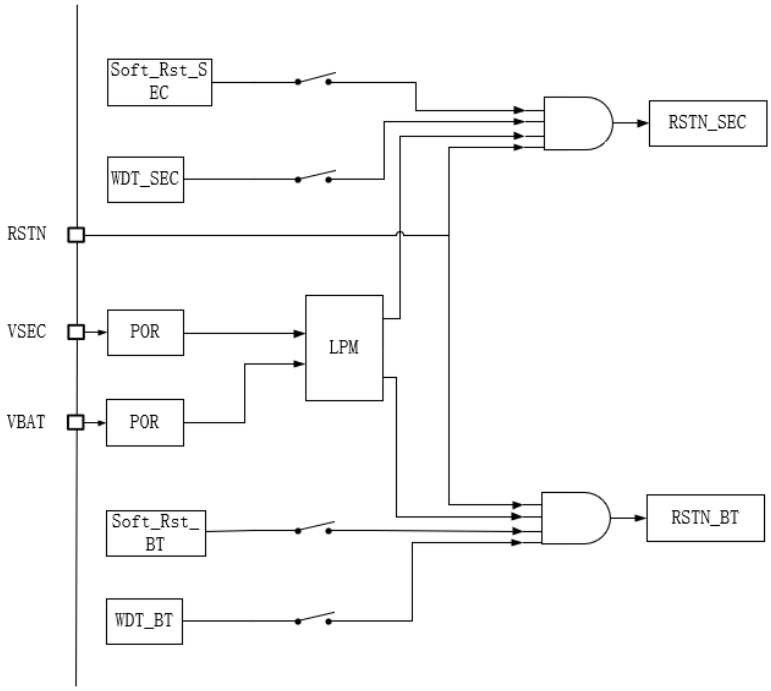


图 5 系统复位控制

3.5 软复位

系统提供软件复位操作，首先要开启软件复位使能（对 SYSCTRL_RST_EN bit0 写 1），然后对软件复位寄存器（SYSCTRL_RESET）写 0x55 实现软件复位。

3.6 时钟

内部 192Mhz OSC 时钟经过 DPLL（/2）后为系统 96Mhz 时钟，为 CPU 以及总线上各个模块提供时钟源。内部 32768HZ 晶体为 RTC，纽扣电池模块，以及蓝牙 Wakeup timer 模块提供时钟源；内部 32K 作为 LPM 域下的时钟，在纽扣电池供电的情况下正常工作。HCLK 可由主时钟 0、2、4、8 分频，再经过分频逻辑单元作为 PCLK 时钟。

内部 40M OSC 只供蓝牙总线上各模块的时钟 。

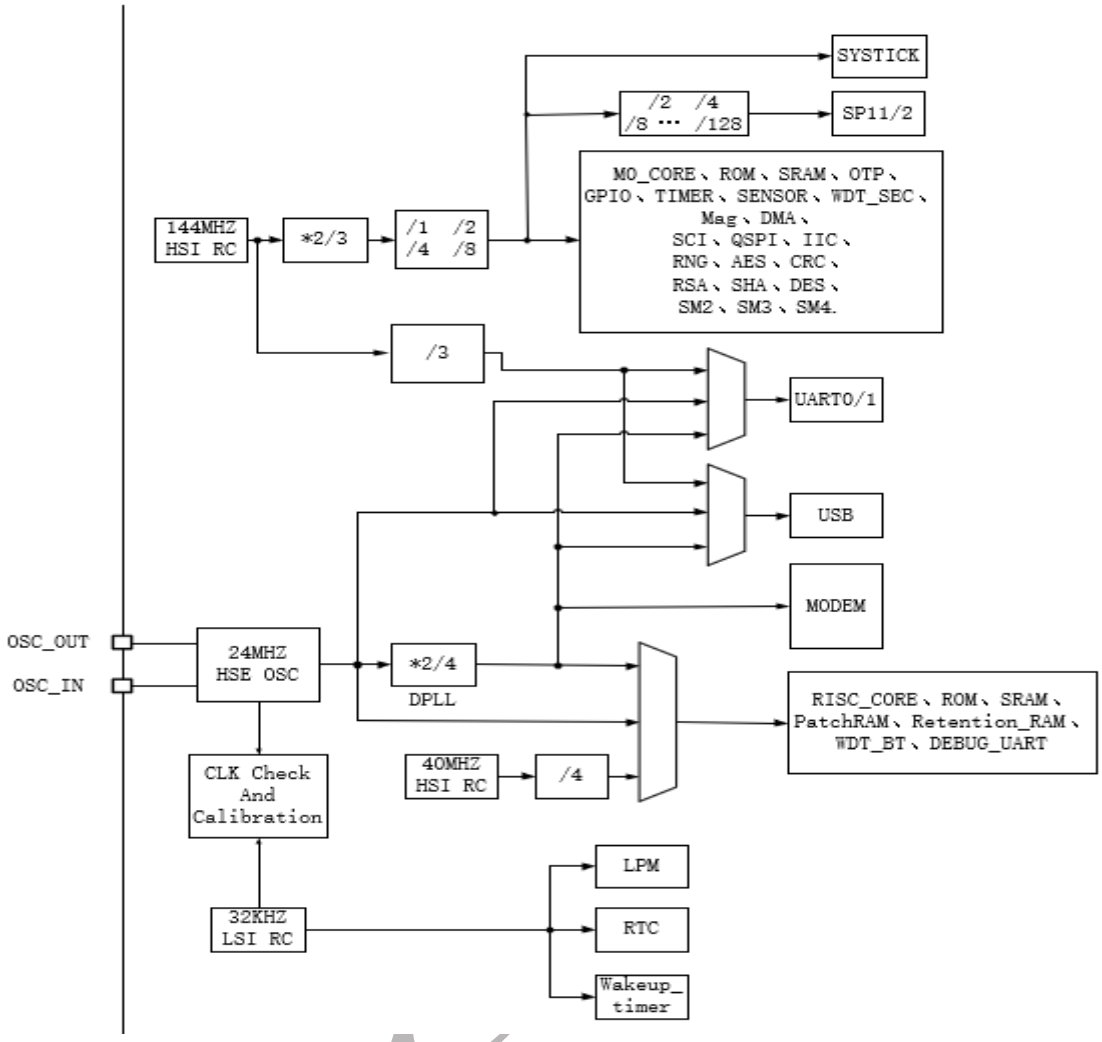


图 6 各模块时钟

3.6.1 外部时钟源

外部 24Mhz 给内部对时间有高要求的模块（UART，USB）提供时钟源，同时也提供对内部 32k 校准的功能。

3.6.2 外设时钟管理

系统提供时钟门控控制寄存器（SYSCTRL_CLK_CLS）管理外设时钟。用户可以通过该寄存器打开和关闭对外设。外设时钟关闭后外设将不在运行，通过该寄存器可以更灵活的控制系统时钟。

3.7 低功耗控制

系统或电源复位后，安全 CPU 处于运行状态。当 CPU 不需要继续运行时，可以利用多种低功耗式来节省能耗，例如等待某个外部事件的产生。

可以通过下几种方式降低功耗：

降低系统 AHB 总线时钟

睡眠模式（CPU 内核停止，所有外设仍在运行）

深度睡眠模式（CPU 内核和外设均停止运行）

3.8 寄存器描述

3.8.1 SYSCTRL_ STATUS

功能：系统控制状态位
地址：SYSCTRL_BASEADDR +0x04（0xf8504）

Bit	功能	说明	R/W	复位值
[13:0]	自检	bist_done	R	0
[27:14]	自检	bist_fail	R	0
[28]	充电标志位	1：CHGR_IND 为高,表示正在充电 0: 充电满	R	0
[29]		Ido_card_oc	R	0
[30]	充电唤醒标志	1：外电插入唤醒（外电插入）CHGIN 为高	R	0
[31]	GPIO 唤醒状态标志位	1： GPIO 唤醒或/power_key(为高)唤醒	R	0

3.8.2 SYSCTRL_LPM_RDATA

功能：读 LPM 域下的寄存器
地址：SYSCTRL_BASEADDR +0x10（0xf8510）

Bit	功能	说明	R/W	复位值
[31:0]		LPM 域下的寄存器，读出来的值不会直接保存在变量中，会保存在此寄存器。所以读 LPM 寄存器，需要读两次，先读本身，再读 SYSCTRL_LPM_RDATA	R	0

3.8.3 SYSCTRL_HWCTRL0

地址：SYSCTRL_BASEADDR +0x18（0xf8518）

Bit	功能	说明	R/W	复位值

[2:0]		rg_ldo_card_vtrim	R/W	101
[3]		Enable of LDO for IC card. 0: off; 1: on	R/W	0
[4]		Output voltage control of LDO for IC card. 0: 1.8V 1: 3V	R/W	0
[5]		Security main LDO output voltage detector enable. 0: off; 1: on	R/W	0
[6]		Security main LDO output voltage detector self-test enable. 0: off; 1: on	R/W	0
[7]		Security main supply voltage detector enable. 0: off; 1: on	R/W	0
[8]		Security main supply voltage detector self-test enable. 0: off; 1: on	R/W	0
[9]		rg_adc_channel_sel_clk_en	R/W	1
[10]		MCR ADC enable. 0: off; 1: on	R/W	0
[11]		MCR ADC reference voltage enable. 0: off; 1: on	R/W	0
[12]		MCR ADC constant Gm bias enable. 0: off; 1: on	R/W	0
[13]		MCR ADC regulator enable. 0: off; 1: on	R/W	0
[15:14]		rg_mcr_adc_clksel	R/W	11
[21:16]		rg_mcr_dcoc_pga0	R/W	10 0000
[23:22]		rg_mcr_adc_refbuf_vref_ctrl	R/W	10
[29:24]		rg_mcr_dcoc_pga1	R/W	10 0000
[31:30]		rg_mcr_adc_rega_vctrl	R/W	10

3.8.4 SYSCTRL_HWCTRL1

地址：SYSCTRL_BASEADDR +0x1c（0xf851c）

Bit	功能	说明	R/W	复位值
[5:0]		rg_mcr_dcoc_pga2	R/W	10 0000
[7:6]		rg_mcr_adc_regd_vctrl	R/W	10
[10:8]		rg_mcr_adc_ibc_refbuf	R/W	111
[11]		rg_mcr_adc_ibc_refbuf2	R/W	0
[14:12]		rg_mcr_adc_vctrl_biasgen	R/W	110
[15]		MCR PGA enable. 0: off; 1: on	R/W	0
[18:16]		rg_mcr_pga_csel	R/W	101
[19]		da_mcr_pga1_en	R/W	0
[22:20]		rg_mcr_pga_ioutsel	R/W	001
[23]		da_mcr_pga2_en	R/W	1
[26:24]		rg_mcr_pga_rlsel	R/W	011
[27]		da_mcr_pga_vcm_gen_en	R/W	0
[30:28]		rg_ts_otc	R/W	000
[31]		rg_mcr_pga_cm_en	R/W	1

3.8.5 SYSCTRL_HWCTRL2

地址：SYSCTRL_BASEADDR +0x20（0xf8520）

Bit	功能	说明	R/W	复位值
[7:0]		rg_mcr_pga_reserve	R/W	00000000
[10:8]		rg_ts_utc	R/W	000
[11]		MCR PGA LDO enable. 0: off; 1: on	R/W	0
[13:12]		rg_adc_test_channel_sel	R/W	00
[14]		rg_adc_test_channel_en	R/W	0
[15]		rg_adc_test_out_en	R/W	0
[16]		rg_mcr_pga_res_bypass	R/W	0
[19:17]		rg_ncs_i_set	R/W	100
[20]		NCS enable. 0: off; 1: enable	R/W	0
[21]		NCS reset. 0: reset; 1: normal operation	R/W	0
[22]		Security TRNG LDO enable. 0: off; 1: on	R/W	1
[23]		TRNG bias enable signal, high active	R/W	1
[24]		TRNGA LFOSC enable signal,high active	R/W	1
[25]		TRNGB LFOSC enable signal,high active	R/W	1

[26]		TRNGC LFOSC enable signal,high active	R/W	1
[27]		TRNGD LFOSC enable signal,high active	R/W	1
[28]		TRNGA sample DFF output clear signal,low active	R/W	1
[29]		TRNGB sample DFF output clear signal,low active	R/W	1
[30]		TRNGC sample DFF output clear signal,low active	R/W	1
[31]		TRNGD sample DFF output clear signal,low active	R/W	1

3.8.6 SYSCTRL_HWCTRL3

地址：SYSCTRL_BASEADDR +0x24（0xf8524）

Bit	功能	说明	R/W	复位值
[1:0]		TRNGA LFOSC frequency control signal	R/W	11
[3:2]		TRNGA LFOSC vref control signal	R/W	01
[6:4]		TRNGA LFOSC jitter control signal	R/W	11
[7]		TRNGA HFOSC enable signal,high active	R/W	10
[9:8]		TRNGB LFOSC frequency control signal	R/W	11
[11:10]		TRNGB LFOSC vref control signal	R/W	01
[14:12]		TRNGB LFOSC jitter control signal	R/W	11
[15]		TRNGB HFOSC enable signal,high active	R/W	10
[17:16]		TRNGC LFOSC frequency control signal	R/W	11
[19:18]		TRNGC LFOSC vref control signal	R/W	01
[22:20]		TRNGC LFOSC jitter control signal	R/W	11
[23]		TRNGC HFOSC enable signal,high active	R/W	10
[25:24]		TRNGD LFOSC frequency control signal	R/W	11
[27:26]		TRNGD LFOSC vref control signal	R/W	01
[30:28]		TRNGD LFOSC jitter control signal	R/W	11
[31]		TRNGD HFOSC enable signal,high active	R/W	10

3.8.7 SYSCTRL_HCLK_CON

功能：配置 AHB 时钟频率和特殊功能
地址：SYSCTRL_BASEADDR +0x60（0xf8560）

Bit	功能	说明	R/W	复位值
[3:0]	Hclk 分频选择	生成 AHB hclk 时的分频值。f(hclk) = clk_osc192m/(hclk_sel+2)	R/W	2’b11
[4]	预留	未使用		
[5]		生成 AHB hclk 时，分频值 hclk_cnt 选择。 0：使用 hclk_sel 1：rng_0data[7:4]>hclk_sel 或 rng_0data[7:4]大于 6 时，使用 rng_0data[7:4]，否则使用（8+rng_0data[7:4]）hclk = clk_osc192m/(hclk_cnt+2)，设置为偶数时，点空比为 50%，否则‘0’比‘1’多一个 clk_osc192m 周期	R/W	0
[6]	预留	0：da_ncs_clk 始终为 0 1：da_ncs_clk 输出，控制频率为 hclk/(2^clkncs_sel)，高电平期间输出 hclk，低电平期间，输出 0	R/W	0
[10:8]		clk_ncs 时钟选择信号。控制频率为 hclk/(2^clkncs_sel)，高电平期间输出 hclk，低电平期间，输出 0	R/W	0
[11]		USB 时钟 clk_usb 选择。 0：osc192m 分频后生成的 48M 时钟，bt_core 外 1：dpll192m 分频后生成的 48M 时钟，bt_core 内	R/W	0
[12]		UART 时钟 clk_uart 选择。 0：osc192m 分频后生成的 48M 时钟，bt_core 外 1：dpll192m 分频后生成的 48M 时钟，bt_core 内	R/W	0
[31:13]	预留			

3.8.8 SYSCTRL_RSA_CLK

功能：配置 RSA 时钟分频
地址：SYSCTRL_BASEADDR +0x68（0xf8568）

Bit	功能	说明	R/W	复位值
[3:0]	Hclk 分频选择	clk_rsa 频率为 hclk，但每 16 个 clk_rsa 中，前 n 个可以强制为 0，此寄存器用于选择 n 值。 1 ：3，每 16 个 hclk 中，	R/W	0000

		前 3 个周期强制为 0，即为 clk_rsa 2，4，5 ：2，每 16 个 hclk 中， 前 2 个周期强制为 0，即为 clk_rsa 3 ：1，每 16 个 hclk 中， 第一个强制为 0，即为 clk_rsa 6，7 ：0，clk_rsa 完全等价于 hclk others ：7，每 16 个 hclk 中，前 7 个周期强制为 0，即为 clk_rsa		
[31:4]	预留	未使用		

3.8.9 SYSCTRL_CLK_CLS

功能：时钟关断寄存器
地址：SYSCTRL_BASEADDR +0x6c（0xf856c）

Bit	功能	说明	R/W	复位值
[0]		reserved	R/W	0
[1]		reserved	R/W	0
[2]	SHA 模块时钟使能	0:打开 SHA 模块时钟 1:关闭 SHA 模块时钟	R/W	0
[3]	CRC 模块时钟控制	0:打开 CRC 模块时钟 1:关闭 CRC 模块时钟	R/W	1
[4]	TIM 模块时钟控制	0:打开 TIM 的模块时钟 1:关闭 TIM 的模块时钟	R/W	0
[5]	看门狗模块时钟控制	0:打开看门狗的时钟 1:关闭看门狗的时钟	R/W	0
[6]	USB 模块时钟控制	0:打开 USB 模块的时钟 1:关闭 USB 模块的时钟	R/W	1
[7]	SPI 模块时钟控制	0:打开 SPI 控制模块时钟 1:关闭 SPI 控制模块时钟	R/W	0
[8]	DES 模块时钟控制	0:打开 DES 模块时钟 1:关闭 DES 模块时钟	R/W	0
[9]	RSA 模块时钟控制	0:打开 RSA 模块时钟 1:关闭 RSA 模块时钟	R/W	1
[10]	AES 模块时钟控制	0:打开 AES 模块时钟 1:关闭 AES 模块时钟	R/W	0
[11]	GPIO 模块时钟控制	0:打开 GPIO 模块时钟 1:关闭 GPIO 模块时钟	R/W	0
[12]	SCI7816 模块时钟控制	0:打开 SCI7816 模块时钟 1:关闭 SCI7816 模块时钟	R/W	0
[13]	蓝牙时钟	0:打开 BT 时钟 1:关闭 BT 时钟	R/W	0
[14]	SM4 模块时钟控制	0：打开 SM4 模块时钟 1：关闭 SM4 模块时钟	R/W	1
[15]	UART 模块时钟控制	0：打开 UART 模块时钟 1：关闭 UART 模块时钟	R/W	0
[16]	7811 磁卡模块时钟	0:打开 7811 模块时钟 1:关闭 7811 模块时钟	R/W	0
[17]	磁卡 adc 时钟控制	0：打开 adc7811 模块时钟 1：关闭 adc7811 模块时钟	R/W	0

[18]	cpclk		R/W	0
[31:13]	预留			

注:

1) 一旦某个模块的时钟被设置成关闭，该模块将停止运作，该模块中的 SFR 将无法被写入。所以，某个模块在使用前必须确认其时钟已经被打开。

2) Bit4(PIT_cls)仅用于控制 Timer1/Timer2/Timer3 的系统时钟；TimerX 的外部时钟（clk_timer1_ext/clk_timer2_ext/clk_timer3_ext）不受此开关控制，而是分别受 Timer1 控制寄存器 Timer1Ctrl[0](Timer1_en)/ Timer2 控制寄存器 Timer2Ctrl[0](Timer2_en)/ Timer3 控制寄存器 Timer3Ctrl[0](Timer3_en)控制。

3) Bit9(RSA_cls)既用于控制 RSA 系统时钟，又用于控制 RSA 外部时钟。

4) 当需要打开 APB 接口类模块（如 GPIO、SCI7816、PIT、WDT、SPI）的时钟，打开该模块的时钟后，如果 CPU 需立刻访问该模块，需要在打开该模块的时钟的指令后加 8 个 NOP 指令，因为 PCLK 可能是 HCLK 的 8 分频，所以需要一段延迟时间来打开 PCLK，PCLK 有效后才能对模块进行访问。

5）因为很多安全功能需要用到 RNG 模块的随机数，所以即使在非接下也不建议关闭 RNG 模块的时钟。

3.8.10SYSCTRL_RST_EN

功能：复位使能寄存器
地址：CLKEN_BASEADDR +0x14（0xf8574）
说明：可读/可写（测试模式）/ 只读（应用模式）

Bit	功能	说明	R/W	复位值
[0]	软复位使能	0: 1:使能软件复位功能	R/W	1
[1]	看门狗复位使能	0: 1:使能看门狗复位功能	R/W	0
[3:2]	预留		R/W	0
[4]	安全域电源 1.2v 输出低压自检复位功能	1: 使能安全域电源 1.2v 输出低压自检复位功能	R/W	0
[5]	锂电池 3.3v 输出高压自检复位功能	1: 使能锂电池 3.3v 输出高压自检复位功能		0
[6]	锂电池 3.3v 输出低压自检复位功能	1: 使能锂电池 3.3v 输出低压自检复位功能	R/W	0
[7]	纽扣电池 1.2v 输出低压自检复位功能	1: 使能纽扣电池 1.2v 输出低压自检复位功能		0
[8]	使能纽扣电池 3.3v	1: 使能纽扣电池 3.3v 输出高压自检复位功能	R/W	0

	输出高压自检复位功能			
[9]	使能纽扣电池 3.3v 输出低压自检复位功能	1: 使能纽扣电池 3.3v 输出低压自检复位功能	R/W	0
[10]	使能高温自检复位功能	1: 使能高温自检复位功能	R/W	0
[11]	使能低温自检复位功能	1: 使能低温自检复位功能	R/W	0
[31:12]	保留			

注：1. 由于上电复位(POR)请求是必须被处理的复位请求，一旦发生必定响应，因此不允许本寄存器来控制是否使能。

2. 本寄存器中定义的某个复位使能一旦被设为允许响应，那么，一旦发生该类型的复位请求，则必然使系统产生规定的复位。反之，某个复位使能一旦被设为不允许响应，那么，相应的复位请求必然不会产生系统复位。

3. SCI7816 PAD(PDR)复位必须响应，所以没有对应的使能位进行控制， PDR 复位只能在接触式时钟存在的情况下才会产生。

3.8.11 SYSCTRL_RST_TYPE

功能：复位类型寄存器，读取复位的类型;软件可读写。
地址：SYSCTRL_BASEADDR +0x18(0xf8578)
说明：若对应的复位信号触发，则硬件自动置 1，清 0 需由软件完成。

Bit	功能	说明	R/W	复位值
[0]	软复位复位状态位	0:本次复位不是软件复位请求 1:本次复位是软件复位请求	R/W	1
[1]	看门狗复位状态位	0:本次复位不是看门狗复位请求 1:本次复位是看门狗复位请求	R/W	0
[3:2]		reserved	R/W	0
[4]	安全域电源 1.2v 输出低压自检复位	1: 本次复是位安全域电源 1.2v 输出低压自检复位	R/W	0
[5]	锂电池 3.3v 输出高压自检复位	1: 本次复位锂电池 3.3v 输出高压自检复位	R/W	0
[6]	锂电池 3.3v 输出低压自检	1: 本次复位锂电池 3.3v 输出低压自检复位	R/W	0

	复位			
[7]	纽扣电池 1.2v 输出低压自检复位	1: 本次复位纽扣电池 1.2v 输出低压自检复位	R/W	0
[8]	纽扣电池 3.3v 输出高压自检复位	1: 本次复位纽扣电池 3.3v 输出高压自检复位	R/W	0
[9]	纽扣电池 3.3v 输出低压自检复位	1: 本次复位是纽扣电池 3.3v 输出低压自检复位	R/W	0
[10]	高温自检复位	1:本次复位是高温自检复位	R/W	0
[11]	低温自检复位	1: 本次复位是低温自检复位	R/W	0
[31:12]		reserved	R	0

注：1. 一旦发生本寄存器中定义的复位请求类型，本寄存器中相应的位将自动由硬件设置为 1。

2. 在复位使能寄存器中被设为不允许响应的复位请求将不会被记录在本寄存器中。

3. 软件应该在每次系统复位后立即查询本寄存器以知晓复位是由哪种（些）请求引起的，以便做出相应处理，并适时清除本寄存器中相应的位。

4. 如果软件同时查询到本寄存器多位为 1，说明在软件查询前确实都发生过这些复位请求。

3.8.12SYSCTRL_RESET

功能:对本寄存器写入 0x55 的值将产生软件复位请求,写 0xAB 触发 SCI 复位；写 0xC3 触发 7811 复位
地址：SYSCTRL_BASEADDR +0x7（0xf857c）

Bit	功能	说明	R/W	复位值
[7:0]	对本寄存器写入 0x55 的值将产生软件复位请求	1） 本寄存器为只写寄存器。 2） 本寄存器并非真正意义上的寄存器。 3）对本寄存器真正有意义的操作就是对本寄存器定义的地址进行写动作,并且数据必须是 0x55，0xAB,0xC3。当硬件观察到该动作时,就会产生软件复位请求。其余针对本寄存器的操作将不产生任何影响。 写入 "0x55" ，触发软件复位，sw_rst 写入 "0xAB“ ,触发 sci 复位,rst_sci 写入 "0xC3" ， 触发 7811 复位，rst_7811	W	0

4 MPU

4.1 简述

MPU 是 YC3121 芯片内部的安全存储保护单元。 MPU 实现内部存储空间的划分及安全管理。 MPU 对芯片的不同应用分配不同的存储空间，同时保护各个应用存储空间的数据不被非法的访问及篡改，并可以指示出存储器及受保护的寄存器非法访问的错误。

MPU 模块的主要功能如下：

1、芯片内部根据地址可以划分为 8 个区，每个区的范围和权限都是可以配置的,每个区的操作可以配置成四种配置。具体如下：

区	寄存器值	含义
Region (0-7)	00	No access
	01	Private Only
	10	Private + user read only
	11	Full Access

其中：

No access ： 没有用到

Private Only：私有权限，只能固件进行读写

Private + user read only： 固件可以进行读写，用户仅能进行读，不行写

Full Access ： 固件和应用都能进行读写

因为权限根据地址划分，可进行划分的区域为整个芯片资源，包括：ROM, RAM, 寄存器, FLASH，划分如下：

	划分	权限设置
ROM	固件	Private Only
寄存器	固件	Private Only
RAM	固件	Private Only
	应用	Full Access
FLASH	固件API	Private + user read only
	固件	Private Only
	应用	Full Access

配置流程：

设备开机后，程序先跑到固件中，固件对 MPU 的权限进行划分，如上图所示。交易所有用到的敏感数据都保存在固件中，应用只能通过固件 API 进行访问，而固件的 API 并不会暴露敏感的数据，这样就能保证敏感数据安全了。 另外一旦权限划分以后，应用无法对 MPU 权限重新分配。

4.2 权限切换

使能 MPU，配置 user_start 寄存器，固件和应用代码空间访问控制如下：

1） 程序运行在固件中，切换为 Private Only 权限，在固件区域内可以进行读写操作。

2） 程序一旦从固件跳转到应用，由用户权限接管程序，则无法对固件再进行读写，但是可以跳转到固件给出的 API 函数 。

3） 程序运行在应用程序中，可以对应用程序进行读写和跳转.

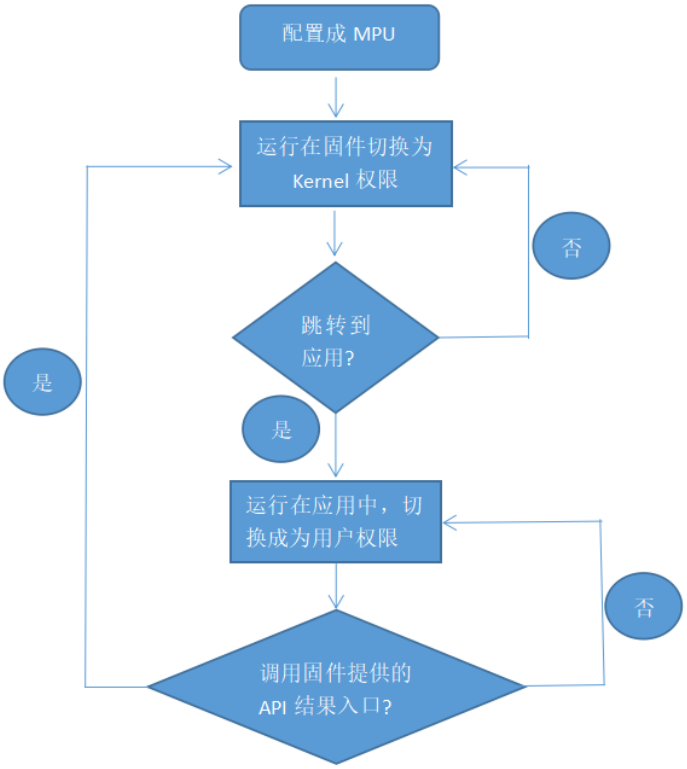


图 7 权限切换

4.3 操作说明

MPU 是 YC3121 芯片内部使用的安全存储保护单元。MPU 实现内部存储空间的划分及安全管理。MPU 会对芯片的固件和应用分配不同的存储空间，同时保护的存储空间的数据不被非法的访问及篡改。MPU 会对内部的 ROM，RAM，Flash，OTP，内部寄存器等地址访问空间进行四种权限的划分。具体使用说明如下

内存的权限设置通过。

固件 code 拥有最高权限，可以访问寄存器，等敏感信息。

应用只能通过固件 API 获取到相关的非敏感信息。例如应用需要加密某些数据，只能通过固件的 API 将待加密的数据输入，来获取加密数据而得不到密钥的相关信息。

程序进入固件，固件 code 可以按照如下的方式配置各段内存的权限。并配置”user_start”，这样就将应用与固件划分开来。应用将不能访问 Private Only 的内存区域，但是可以调用固件 API 来获取相关的非敏感数据。

4.4 寄存器说明

4.4.1 MPUCTRL_CTRL

功能：MPU 控制寄存器
地址： 0xf8584

Bit	功能	说明	R/W	复位值
[0]	使能 MPU	0:mpu enable 1:mpu disable	R/W	0
[31:1]	预留		R/W	0

4.4.2 MPUCTRL_FSR

功能： MPUCTRL_FSR
地址： 0xf859c

Bit	功能	说明	R/W	复位值
[1:0]		Fault status	R/W	0
[4:2]		fault region	R/W	0

4.4.3 MPUCTRL_PROTECTION

功能： MPUCTRL
地址： 0xf8594

Bit	功能	说明	R/W	复位值
[15:0]		protection15：0 00：no access 01：private only 10：private + user read only 11：Full Access	R/W	0

4.4.4 MPUCTRL_USER_START

功能： 设置用户程序的起始地址
地址： 0xf8598

Bit	功能	说明	R/W	复位值
[31:0]		用户程序的起始地址	R/W	0

4.4.5 MPUCTRL_REGION_BASE0

功能：划分首地址寄存器，配置区域起始地址
地址： 0xf85c0

Bit	功能	说明	R/W	复位值
[0]	使能位	1：regionx_enable	R/W	0
[5:1]	预留		R/W	
[20:6]	配置区域起始地址	BASEx_REG	R/W	0
[31:21]	预留			

4.4.6 MPUCTRL_REGION_BASE1

功能：划分首地址寄存器，配置区域起始地址
地址： 0xf85c4

Bit	功能	说明	R/W	复位值
[0]	使能位	1：regionx_enable	R/W	0
[5:1]	预留		R/W	
[20:6]	配置区域起始地址	BASEx_REG	R/W	0
[31:21]	预留			

4.4.7 MPUCTRL_REGION_BASE2

功能：划分首地址寄存器，配置区域起始地址
地址： 0xf85c8

Bit	功能	说明	R/W	复位值
[0]	使能位	1：regionx_enable	R/W	0
[5:1]	预留		R/W	
[20:6]	配置区域起始地址	BASEx_REG	R/W	0
[31:21]	预留			

4.4.8 MPUCTRL_REGION_BASE3

功能：划分首地址寄存器，配置区域起始地址
地址： 0xf85cc

Bit	功能	说明	R/W	复位值

[0]	使能位	1：regionx_enable	R/W	0
[5:1]	预留		R/W	
[20:6]	配置区域起始地址	BASEx_REG	R/W	0
[31:21]	预留			

4.4.9 MPUCTRL_REGION_BASE4

功能： 划分首地址寄存器，配置区域起始地址
地址： 0xf85d0

Bit	功能	说明	R/W	复位值
[0]	使能位	1：regionx_enable	R/W	0
[5:1]	预留		R/W	
[20:6]	配置区域起始地址	BASEx_REG	R/W	0
[31:21]	预留			

4.4.10 MPUCTRL_REGION_BASE5

功能： 划分首地址寄存器，配置区域起始地址
地址： 0xf85d4

Bit	功能	说明	R/W	复位值
[0]	使能位	1：regionx_enable	R/W	0
[5:1]	预留		R/W	
[20:6]	配置区域起始地址	BASEx_REG	R/W	0
[31:21]	预留			

4.4.11 MPUCTRL_REGION_BASE6

功能： 划分首地址寄存器，配置区域起始地址
地址： 0xf85d8

Bit	功能	说明	R/W	复位值
[0]	使能位	1：regionx_enable	R/W	0
[5:1]	预留		R/W	
[20:6]	配置区域起始地址	BASEx_REG	R/W	0
[31:21]	预留			

4.4.12 MPUCTRL_REGION_BASE7

功能：划分首地址寄存器，配置区域起始地址
地址： 0xf85dc

Bit	功能	说明	R/W	复位值
[0]	使能位	1：regionx_enable	R/W	0
[5:1]	预留		R/W	
[20:6]	配置区域起始地址	BASEx_REG	R/W	0
[31:21]	预留			

4.4.13 MPUCTRL_REGION_LIMIT0

功能： 划分首地址寄存器，配置结束地址
地址： 0xf85e0

Bit	功能	说明	R/W	复位值
[5:0]	预留		R/W	0
[20:6]		配置区域结束地址	R/W	0
[31:21]	预留			

4.4.14 MPUCTRL_REGION_LIMIT1

功能：划分首地址寄存器，配置结束地址
地址： 0xf85e4

Bit	功能	说明	R/W	复位值
[5:0]	预留		R/W	0
[20:6]		配置区域结束地址	R/W	0
[31:21]	预留			

4.4.15 MPUCTRL_REGION_LIMIT2

功能：划分首地址寄存器，配置结束地址
地址： 0xf85e8

Bit	功能	说明	R/W	复位值
[5:0]	预留		R/W	0
[20:6]		配置区域结束地址	R/W	0
[31:21]	预留			

4.4.16 MPUCTRL_REGION_LIMIT3

功能：划分首地址寄存器，配置结束地址
地址： 0xf85ec

Bit	功能	说明	R/W	复位值
[5:0]	预留		R/W	0
[20:6]		配置区域结束地址	R/W	0
[31:21]	预留			

4.4.17 MPUCTRL_REGION_LIMIT4

功能：划分首地址寄存器，配置结束地址
地址： 0xf85f0

Bit	功能	说明	R/W	复位值
[5:0]	预留		R/W	0
[20:6]		配置区域结束地址	R/W	0
[31:21]	预留			

4.4.18 MPUCTRL_REGION_LIMIT5

功能：划分首地址寄存器，配置结束地址
地址： 0xf85f4

Bit	功能	说明	R/W	复位值
[5:0]	预留		R/W	0
[20:6]		配置区域结束地址	R/W	0
[31:21]	预留			

4.4.19 MPUCTRL_REGION_LIMIT6

功能：划分首地址寄存器，配置结束地址
地址： 0xf85f8

Bit	功能	说明	R/W	复位值
[5:0]	预留		R/W	0
[20:6]		配置区域结束地址	R/W	0
[31:21]	预留			

4.4.20 MPUCTRL_REGION_LIMIT7

功能：划分首地址寄存器，配置结束地址
地址： 0xf85fc

Bit	功能	说明	R/W	复位值
[5:0]	预留		R/W	0
[20:6]		配置区域结束地址	R/W	0
[31:21]	预留			

5 通用输入输出（GPIO）

5.1 GPIO 功能描述

芯片一共有 48 个 GPIO。每个 GPIO 可以复用为任意外设的输入或者输出脚。GPIO 端口的每个引脚可以配置为多种工作方式。

输入模式（浮空输入、输入上拉、输入下拉）；

推挽输出

模拟输入

5.1.1 通用 I/O（GPIO）

作为输出配置时，写到输出数据寄存器上的值将输出到对于 I/O 上。输入数据寄存器显示 APB 上 捕捉 I/O 上的数据。所有 GPIO 引脚上都有一个内部上拉，可以通过上拉使能寄存器控制是否有效。

5.1.2 专用 I/O（GPIO）

GPIO45、GPIO46、GPIO47 这 3 个 I/O 是 7816 模块所使用的 I/O。

5.1.3 外部中断

通过 GPIO_INTR_EN 寄存器开启中断，GPIO_TRIG_MODE 控制中断响应的类型。

GPIO 外部中断响应类型：

高电平中断；

低电平中断。

5.1.4 外部唤醒事件

芯片所有 GPIO 管脚均支持超低功耗唤醒，GPIO 支持低电平唤醒或高电平唤醒。每个 GPIO 都有独立的 GOIO 低功耗唤醒使能位，由低功耗域下寄存器 LPM_GPIO_WKUP 与 LPM_GPIO_WKHI

5.1.5 I/O 功能复用

外设与 I/O 复用可通过复用控制寄存器（GPIO_CONFIG）进行配置。

根据需要进行I/O 复用。

复用为功能外设后，无需配置 I/O 工作模式，复用配置完成后系统会自动进入对应 I/O 工作模式。

5.2 GPIO 寄存器

5.2.1 配置寄存器 GPIO_CONFIG

功能：GPIO 配置寄存器
地址：GPIO_BASEADDR+0x00
说明： 第 n 个 GPIO 的配置寄存器地址为 0xf8700 +n

Bit	功能	说明	R/W	复位值
[5:0]	IO function	赋值与功能对应关系见下表	R、W	0
[7:6]	IO mode	00:FLOAT 01:PULL UP 10:PULL DOWN 11: ANALOG	R、W	0

GPIO_CONFIG 寄存器值与 IO 功能对应表

寄存器值	功能	寄存器值	功能
0	Input(float)	1	预留
2	QSPI_NCS	3	QSPI_SCK
4	QSPI_IO0	5	QSPI_IO1
6	QSPI_IO2	7	QSPI_IO3
8	UART0_TXD	9	UART0_RXD
10	UART0_RTS	11	UART0_CTS
12	UART1_TXD	13	UART1_RXD
14	UART1_RTS	15	UART1_CTS
16	PWM_OUT0	17	PWM_OUT1
18	PWM_OUT2	19	PWM_OUT3
20	PWM_OUT4	21	PWM_OUT5
22	PWM_OUT6	23	PWM_OUT7

24	SPI0_NCS	25	SPI0_SCK
26	SPI0_MOSI	27	SPI0_SDIO
28	SPI0_MISO	29	GPCFG_SPID0_NCSIN
30	GPCFG_SPID0_SCKIN	31	GPCFG_PWM_OUT8
32	预留	33	预留
34	预留	35	预留
36	预留	37	预留
38	预留	39	预留
40	预留	41	预留
42	预留	43	预留
44	预留	45	预留
46	预留	47	预留
48	SPI1_NCS	49	SPI1_SCK
50	SPI1_MOSI	51	SPI1_SDIO
52	SPI1_MISO	53	GPCFG_SPID1_NCSIN
54	GPCFG_SPID1_SCKIN	55	预留
56	GPCFG_SCI7816_IO	57	GPCFG_ICE
58	IIC_SCL	59	IIC_SDA
60	JTAG_SW_CLOCK	61	JTAG_SW_DATA
62	GPIO_OUTPUT_LOW	63	GPIO_ OUTPUT_HIGH
64	PULL UP	128	PUUL DOWN
192	ANALOG		

5.2.2 中断模式配置寄存器 GPIO_TRIG_MODE

功能：GPIO 中断触发方式配置寄存器
地址：GPIO_BASEADDR+0x36（0xf8736-0xf873b）

Bit	说明	R/W	复位值
[47:0]	0： 高电平触发 GPIO 中断 1： 低电平触发 GPIO 中断 说明： 1、bit0-47 分别控制 gpio0-47 的中断使能 2、当一直处于高或低电平是会一直触发中断(采用首次进中断后将终端触发方式置反来避免一直触发,当置反的后的中断触发时再把中断触发方式再次置反）	R/W	0

5.2.3 中断使能寄存器 GPIO_INTR_EN

功能：input 模式下读取 GPIO 状态寄存器
地址：GPIO_BASEADDR+0x30（f8730-f8735）
说明：每个 bit 分别控制一个 IO 的中断使能

Bit	功能	说明	R/W	复位值
[0]	GPIO0 中断出发模式	0： 1：使能 GPIO 中断	R/W	0

[1]..[46]	分别控制 GPIO1 至 GPIO46 中断出发模式	0： 1： 使能 GPIO 中断	R/W	
[47]	GPIO47 中断出发模式	0： 1： 使能 GPIO 中断	R/W	

5.2.4 状态寄存器 GPIO_IN

功能：input 模式下读取 GPIO 状态寄存器
地址：GPIO_BASEADDR+0x3c（0xf873c-0xf8741）
说明： 从地址 GPIO_BASEADDR+0x3C 第 0bit 开始依次表示 gpio0 到 gpio47，每个 bit 分别表示一个 GPIO 的状态。

Bit	功能	说明	R/W	复位值
0	gpio0 状态位	0： 低电平 1： 高电平	R	0
[1]..[46]	分别控制 GPIO1 至 GPIO46 中表示	0： 低电平 1： 高电平	R	0
[47]	GPIO47 状态位	0： 低电平 1： 高电平	R	0

6 CRC 计算单元

6.1 CRC 简介

循环冗余校验计算单元（CRC）为 16 位的校验，校验的结果由寄存器 CRC_RESULT_REG 读出；校验的初始值配置到 CRC_RESULT_REG 中。CRC 主要用来检测或校验数据传输或者保存后可能出现的错误。CRC 模块时钟复位时关闭，使用该模块需要先打开。

6.2 CRC 主要特性

仅支持 CRC16 校验

数据按字节输入

支持指定 CRC 计算初始值

下图为 CRC 计算单元框图：

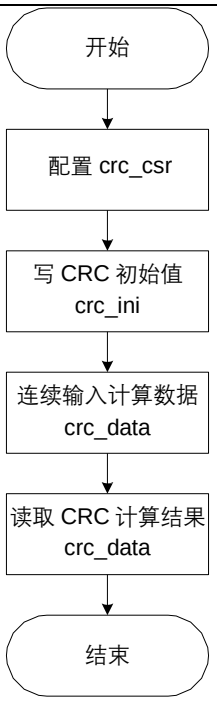


图 8 CRC 计算单元框图

6.3 CRC 寄存器

6.3.1 CRC_RESULT_REG

功能：CRC 结果寄存器
地址：(CRC_BASEADDR+0X04)0xf8204
说明：低十六位保存 CRC 的校验值

Bit	功能	说明	R/W	复位值
[15:0]	CRC 校验值	可写：CRC 计算初始值 可读：CRC 计算结果	R/W	0
[31:16]		reserved		0

6.3.2 CRC_MASK_REG

功能：CRC 掩码寄存器
地址：(CRC_BASEADDR+0X08)0xf8208

Bit	功能	说明	R/W	复位值
[15:0]	CRC 校验值	安全的掩码，对校验结果没有影响	W	0

6.3.3 CRC_DATAB_REG

功能：CRC 数据寄存器
地址：(CRC_BASEADDR+0X80)0xf8280
说明：将要校验的数据写入

Bit	功能	说明	R/W	复位值
[7:0]	写入数据校验	每写入一次，自运算校验值	W	0

7 真随机数发生器（TRNG）

7.1 TRNG 简介

RNG 随机数发生器通过控制和处理物理噪声源中产生的一连串真随机数字节，为芯片在某些应用场景中提供随机数。随机数模拟模块（RNG_SRC）为随机数数字模块(RNG_UNIT)提供 1bit 的随机噪声源，数字电路为模拟电路提供控制信号，并对随机噪声源进行数学后处理，产生符合要求的随机数。

随机数也用于生成 RSA 和 DES 加密密钥。也用于安全模块、时钟模块、系统模块。

7.2 TRNG 主要特性

- 支持 AMBA 2.0 AHB 总线（8 位，16 位，32 位）以小端方式进行访问，也支持系统对本模块进行权限控制。
- 一次操作产生的随机数长度为：128Bits。
- 支持 TOT 检验。

7.3 TRNG 功能描述

RNG 后处理流程如下图所示：

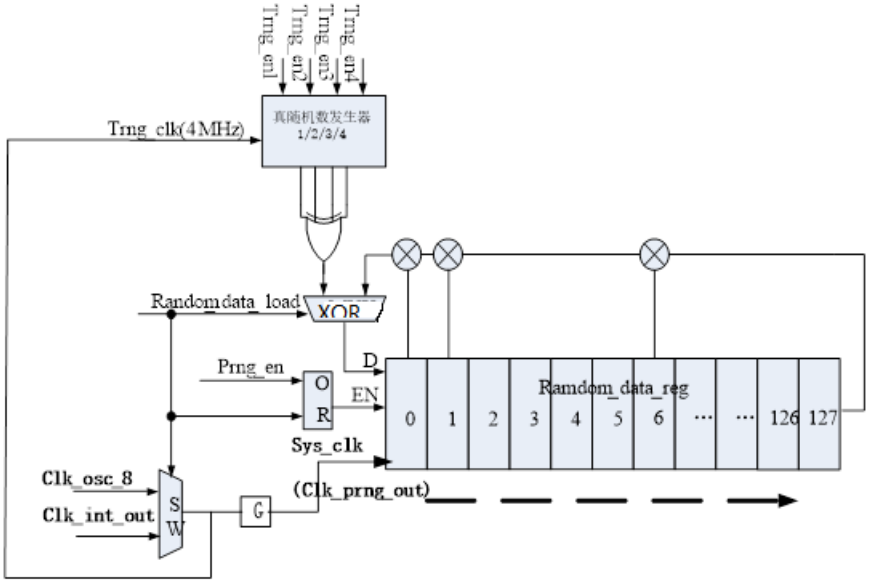


图 9 后处理流程图

说明：

模块使用方法：

1. 使能真随机数发生器 TRNG_en1、2、3、4＝1，使其输出真随机序列。

2. 开启真随机数质量判断功能

3. 真随机数发生器输出装载 M 序列，等待 n 个系统时钟的时间，软件置 Random_data_load＝1，等待硬件装载 128 位 M 序列。

4. 硬件装满后检查是否为全 0 或全 1，若是则重新装载，

硬件装满后检查 128bit 1 的个数是否满足 46<=X<=82，如不满足则重新装载，

硬件装满后检查相邻两比特异或为 1 的个数是否满足 46<=Y<=81，若不满足则重新装载（软件在系统起始需要配置是否进行后两项的检测）；否则硬件拉低 Random_data_load，结束装载。此过程中 M 序列中的值系统不可读。

5. 关闭真随机数发生器 TRNG_en1、2、3、4＝0（逐个关闭）。

6. 使能 M 序列 PRNG_en＝1，开始线性反馈移位。

7. 软件等待 m 个系统时钟的时间

8. 软件可读取 M 序列中的随机数据。

9. 按照上述过程，软件先后读取两次 M 序列中的随机数，进行比较，如果相同就返回错误代码，重取随机数；如果不同，两组随机数，一个做密钥，一个做明文，进行 3DES 运算，得到的密文与密钥再次异或，结果作为最后的输出。

7.4 TRNG 寄存器

7.4.1 SYSCTRL_RNG_CTRL

功能：随机数控制寄存器
地址：SYSCTRL_BASEADDR +0x28（0xf8528）

Bit	功能	说明	R/W	复位值
[0]		rng_gen_en	R/W	1
[1]		rng_tot_test_en	R/W	1
[2]		rng_po_check_en	R/W	1
[3]		rng_soft_seed_on	R/W	0
[6:4]		rng_tot_ctrl	R/W	1
[8:7]		rng_src_sel	R/W	1
[9]		rg_init_tot_alarm_dis	R/W	1
[10]		rng_init_po_alarm_dis	R/W	1
[11]		rng_prng_sel	R/W	1
[15:12]		rng_clk_div_num	R/W	1
[16]		rng_soft_rd_test	R/W	0
[31:17]		reserved	R	0

7.4.1 SYSCTRL_RNG_DATA0

功能：随机数寄存器
地址：SYSCTRL_BASEADDR +0x2c（0xf852c）

Bit	功能	说明	R/W	复位值
[31:0]		rng_0data	R	

7.4.2 SYSCTRL_RNG_DATA1

功能：flash 控制寄存器
地址：SYSCTRL_BASEADDR +0x30（0xf8530）

Bit	功能	说明	R/W	复位值
[31:0]		rng_1data	R	

7.4.3 SYSCTRL_RNG_DATA2

功能：flash 控制寄存器
地址：SYSCTRL_BASEADDR +0x34（0xf8534）

Bit	功能	说明	R/W	复位值
[31:0]		rng_2data	R	

7.4.4 SYSCTRL_RNG_DATA3

功能：flash 控制寄存器
地址：SYSCTRL_BASEADDR +0x38（0xf8538）

Bit	功能	说明	R/W	复位值
[31:0]		rng_3data	R	

8 OTP 控制模块（OTP_CTRL）

8.1 OTP 简介

芯片内置一块 8KBytes 的 OTP，可以配置成三种区域只读区域，隐藏区域（不可读写），用户可操作区域，并通过 otp 锁定寄存器锁定配置（在调试和产品阶段由 ROM 控制）。该 OTP 采用 fuse 的方式，数据只能由 0 写成 1 采用的是 oxide breakdown，确保了数据的安全性。OTP 是 1 块具有单次写操作的特殊存储器，OTP 出厂时内部数据经过初始化后 bit 位均为“0”，OTP 写操作只能将内部 bit 位由“0”写为“1”，而不能由“1”改为“0”。写 OTP 需要 VPP 供 6.5V 直流电压。

8.2 OTP 功能描述

8.2.1 OTP 只读锁定

OTP 提供区域写保护和区域写保护锁定功能。

区域写保护：

OTP 区域写保护 bit 位为“0”时，对应区域可以进行编程操作，为“1”时，对应区域只能进行读操作。

区域写保护锁定：

OTP 区域写保护锁定 bit 位为“0”时，对应区域写保护 bit 位可以修改，为“1”时，对应区域写保护 bit 位保持已有状态不可修改。

8.2.2 OTP 编程操作保护

为防止用户程序对 OTP 的误操作，OTP 在启动编程/擦除操作时，需要进行固定的寄存器操作后， 再启动编程/擦除操作使能。

8.3 寄存器模块

8.3.1 OTP_CTRL

功能：OTP 控制寄存器
地址：SYSCTRL_BASEADDR +0xa（0xf850a）

Bit	功能	说明	R/W	复位值
[15:0]		otp_addr	R/W	
[25:16]		otp_ctrl	R/W	
[31:26]	预留			

8.3.2 OTP_STATUS

功能：OTP 状态寄存器
地址：SYSCTRL_BASEADDR +0x0c（0xf850c）

Bit	功能	说明	R/W	复位值
[7:0]		otp_data	R	
[8]		otp_status	R	
[15:9]	预留		R	
[24:16]		Sar adc data	R	
[31:25]	预留		R	

9 CACHE 模块（CACHE）

9.1 CACHE 简介

Cache 用于提升处理器从低速存储器中取指的效率，为两者的中间媒介。其原理是根据程序局 部性原则，通过小容量速度快的存储器缓存部分指令或数据，以减少处理器对慢速大容量存储器的 访问次数，从而提升处理器效率。

9.2 CACHE 特性

- 处理器通过AHB Code Bus 从Cache 中取指，Cache 访问Flash 的最大空间为 2MB；
- 内含 16KByte 高速缓存（Cache）；
- Cache line 大小为32Bytes；
- 包含1-Way 牺牲缓存；

9.3 CACHE 功能描述

CPU 通过 CodeBus 从 Cache 中取指，Cache 通过 AHB Master 从 Flash Controller 中读取数据。

Cache 访问 Flash 的最大空间为 2MB。

主缓冲区大小为4098x32bit，TAG_RAM大小为64x8x32bit为主缓存单元对应的CodeBus的地址标签，用于识别主缓存的命中或缺失。

Cache 工作流程为 Code Bus 发起读操作，根据 CodeBus 地址 Index 段选中的某一组，然后将该组所有路的 TAG 与 Code Bus 的 TAG 段进行匹配，确定 Cache 命中或缺失；当命中某路，用地址 Offset 段选通数据中的某个 32-bit 数据输出；如果均未命中则挂起总线，通过 AHB 进行 Flash 数据读操作。

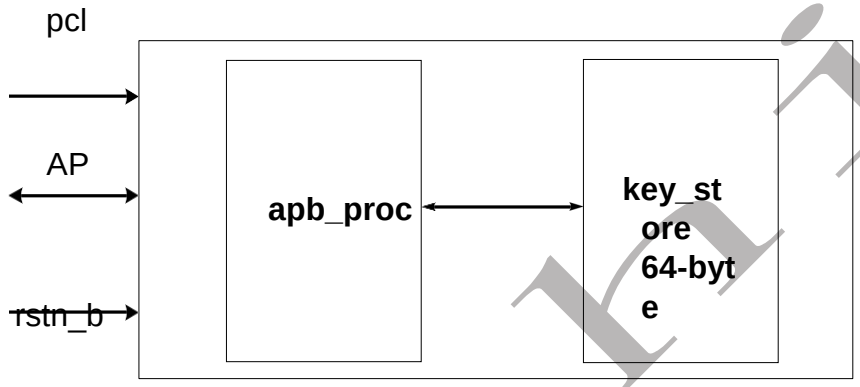
10 备份寄存器 BPK

10.1 BPK 简介

BPK、RTC、SENSOR 都处于电池电源域。电池和主电源同时存在的情况下，由主电源和电池 电源同时给安全区供电，此时两个电源中电压偏高的电源提供较大的电流。

系统提供 32Bx32 的带电存储区域（BPK），其数据在受到攻击后由硬件自动清除。

模块示意图:



10.2 BPK 特性

空间 32Bx32

BPK 内数据在受到攻击后由硬件清除

10.3BPK 寄存器

10.3.1BPK 基地址

BPK 起始地址：0xf8400
BPK 地址范围：0xf8400~0xf887f

10.3.2读写 BPK：LPM_KEY(x)

功能：用于存储密钥
地址：(LPM_BASEADDR + 0x80 + x*4)
说明：0xf4080+x*4 (x 的范围在 0 到 255)；该寄存器属于 LPM 域。读写方式请参考 3.9.2

SYSCTRL_LPM_RDATA

Bit	功能	说明	R/W	复位值
[1023：0]	存储密钥	存储 1024bytes 密钥,每次取 4byte	R/W	0

11 传感器单元（SENSOR）

11.1 SENSOR 简介

BPK、RTC、SENSOR 都处于 VBAT 电池电源域。电池和主电源同时存在的情况下，由主电源和电池 电源同时给安全区供电，此时两个电源中电压偏高的电源提供较大的电流。

11.2 SENSOR 特性

可配外部动态/静态 TAMPER

最多 8 路外部静态 TAMPER

高、低温检测

高、低压检测

Active shielding

探测到攻击后产生中断/复位选择

动态 Tamper/active shielding 翻转频率设定

11.3 外部静态/动态功能说明

外部传感器可配成静态或动态模式，静态传感器的“攻击电平”固定，外部传感器检测到攻击电 平后擦除 BPK 中的数据；

“静态传感器”指正常情况下，传感器的输入端口为固定的 0，当该电平翻转后，则传感器认 为发生攻击。

静态配置时，各端口均为输入，外部传感器对端口的高低电平进行检测。当检测到“攻击电平” 时，激活“BPK 擦除操作”。**静态使用时是成对打开的，只使用一个静态，另一个脚需要接地，**各端口对应传感器如下表。

表 11-1 静态传感器端口表

传感器名	端口（方向均为输入）
静态传感器0	ext_s0(高电平产生攻击)
静态传感器1	ext_s1(高电平产生攻击)
静态传感器2	ext_s2(高电平产生攻击)
静态传感器3	ext_s3(高电平产生攻击)
静态传感器4	ext_s4(高电平产生攻击)
静态传感器5	ext_s5(高电平产生攻击)
静态传感器6	ext_s6(高电平产生攻击)
静态传感器7	ext_s7(高电平产生攻击)

“动态传感器”指传感器的输出输入端口组成回环，输出端口发送随机数，输入端口接收，如果发送和接收数据不相同，则认为发生攻击。

“动态传感器”相邻两端口组成为动态传感器的输出/输入端口,各端口对应传感器如表“动态传感器端口”所示。

表 11-2 动态传感器端口表

传感器名	输出	输入
动态传感器0	ext_s0	ext_s1
动态传感器1	ext_s2	ext_s3
动态传感器2	ext_s4	ext_s5
动态传感器3	ext_s6	ext_s7

11.3.1 SECURE_CTRL

功能： sensor 检测使能。
地址：SECURE_BASEADDR +0x00（0xf8540）

Bit	功能	说明	R/W	复位值
[0]	安全域电源 1.2v 输出低压自检 ad_vddsec_uv/uvb	0： 1:使能	R/W	0
[1]	锂电池 3.3v 输出高压自检，ad_vsec_ov/ovb	0： 1:使能	R/W	0
[2]	锂电池 3.3v 输出低压自检，ad_vsec_uv/uvb	0： 1:使能	R/W	0
[3]	纽扣电池 1.2v 输出低压自检，ad_dvddlpm_uvh/uvhb	0： 1:使能	R/W	0
[4]	纽扣电池 3.3v 输出高压自检，ad_vbat_ovh/ovhb	0： 1:使能	R/W	0
[5]	纽扣电池 3.3v 输出低压自检，ad_vbat_uvh/uvhb	0： 1:使能	R/W	0
[6]	高温自检，ad_ts_oth/othb	0： 1:使能	R/W	0
[7]	低温自检，ad_ts_uth/uthb	0： 1:使能	R/W	0
[11:8]	sensor 检测警报持续时间门限，大于此门限发出警	0： 1:使能	R/W	0

	报，否则不报警。 时间门限值：(2^sensor_delay)*hclk			
[31:12]	预留		R/W	0

11.3.2 SECURE_STATUS

功能：安全警报

地址：SECURE_BASEADDR +0x04（0xf8544）

Bit	功能	说明	R/W	复位值
[0]		1： [8:1]任一为 1；	R	0
[1]	安全域电源 1.2v 输出低压自检警报	1：对应检测项报警。	R	0
[2]	锂电池 3.3v 输出高压自检警报	1： 锂电池 3.3v 输出高压自检警报	R	0
[3]	锂电池 3.3v 输出低压自检警报	1：锂电池 3.3v 输出低压自检警报	R	0
[4]	纽扣电池 1.2v 输出低压自检警报	1：纽扣电池 1.2v 输出低压自检警报	R	0
[5]	纽扣电池 3.3v 输出高压自检警报	1：纽扣电池 3.3v 输出高压自检警报	R	0
[6]	纽扣电池 3.3v 输出低压自检警报	1：纽扣电池 3.3v 输出低压自检警报	R	0
[7]	高温自检警报	1：高温自检警报	R	0
[8]	低温自检警报		R	0
[9]	安全域电源 1.2v 输出低压自检电路结果输出	1：安全域电源 1.2v 输出低压自检电路结果输出	R	0
[10]	安全域电源 1.2v 输出低压自检电路结果输出	1：安全域电源 1.2v 输出低压自检电路结果输出	R	0
[11]	锂电池 3.3v 输出高压自检电路结果输出	锂电池 3.3v 输出高压自检电路结果输出	R	0
[12]	锂电池 3.3v 输出高压自检电路结果输出	锂电池 3.3v 输出高压自检电路结果输出	R	0
[13]	锂电池 3.3v 输出低压自检电路结果输出	锂电池 3.3v 输出低压自检电路结果输出	R	0
[14]	锂电池 3.3v 输出低压自检电路结果输出	锂电池 3.3v 输出低压自检电路结果输出	R	0
[15]	纽扣电池 1.2v 输出低压自检电路结果输出	纽扣电池 1.2v 输出低压自检电路结果输出	R	0
[16]	纽扣电池 1.2v 输出低压自检电路结果输出	纽扣电池 1.2v 输出低压自检电路结果输出	R	0
[17]	纽扣电池 3.3v 输出高压自检电路结果输出	纽扣电池 3.3v 输出高压自检电路结果输出	R	0
[18]	纽扣电池 3.3v 输出高压自检电路结果输出	纽扣电池 3.3v 输出高压自检电路结果输出	R	0
[19]	纽扣电池 3.3v 输出低压自检电路结果输出	纽扣电池 3.3v 输出低压自检电路结果输出	R	0
[20]	纽扣电池 3.3v 输出低压自检电路结果输出	纽扣电池 3.3v 输出低压自检电路结果输出	R	0

[21]	高温自检电路结果输出	高温自检电路结果输出	R	0
[22]	高温自检电路结果输出	高温自检电路结果输出	R	0
[23]	低温自检电路结果输出	低温自检电路结果输出	R	0
[24]	低温自检电路结果输出	低温自检电路结果输出	R	0
[31:25]		reserved	R	0

11.3.3LPM_CTRL

功能：安全电压
地址：0xf8400
说明：LPM 域下的寄存器，

Bit	功能	说明	R/W	复位值
[0]	da_ldo_sec_en	Security core LDO enable. 0: off; 1: on	R/W	0
[1]	da_ldo_osc192m_en	Security RC oscillator LDO enable. 0: off; 1: on	R/W	0
[2]	da_osc192m_en	Security RC oscillator enable. 0: off; 1: on	R/W	0
[3]	da_osc192m_rstn	Security RC oscillator reset. 0: reset; 1: normal operation		0
[4]	da_dvddlpm_vdt_en	1: 模拟自检使能，实际使能还受 sensor_dur 控制。纽扣电池 1.2v 低压检测电路		0
[5]	da_vbat_vdt_en	1: 模拟自检使能，实际使能还受 sensor_dur 控制。纽扣电池 3.3v 高压和低压检测电路		0
[6]	da_ts_en	1: 模拟自检使能，实际使能还受 sensor_dur 控制。高温检测和低温检测电路		0
[7]	da_hvldo_doze_en	HVLDO doze enable. 0: 700nA: 1: 100nA		0
[16:8]	rg_osc192m_vc	rc clock 192m 频率调整		0
[17]	da_ts_test_en	1: 测试用寄存器，强制模拟自检输出警报		0
[19:18]	osc_lpm_ib_ctrl			0
[23:20]	rg_ldo_sec_vtrim	LDO SEC 输出电压调整		0
[26:24]	rg_ts_otc<2:0>			0
[27]	da_vbat_vdt_test_en	1: 测试用寄存器，强制模拟自检输出警报		0
[30:28]	rg_ts_utc<2:0>			0
[31]	da_dvddlpm_vdt_test_en	1: 测试用寄存器，强制模拟自检输出警报		0

11.3.4LPM_SENSOR

功能：总线加密控制
地址：0xf8404
说明：

Bit	功能	说明	R/W	复位值
[0]		传感器检测使能。 [0]：纽扣电池 1.2v 输出低压自检	R/W	0
[1]		[1]：纽扣电池 3.3v 输出高压自检	R/W	0
[2]		[2]：纽扣电池 3.3v 输出低压自检	R/W	0
[3]		[3]：高温自检		0
[4]		[4]：低温自检		0
[6:5]		警报持续时间门限，大于门限的信号，将触发 KEY 擦除操作。用于滤除毛刺，防止虚警。 00：1*(1/32k)=31.25us 01：8*(1/32k)=250us 10：32*(1/32k)=1ms 11：128*(1/32k)=4ms，模拟和软件确认？现在是传感器打开就检测！！！同 1 同 0 未检测		0
[7]		锁定寄存器，配置为 1 后，无法写回 0。 0：寄存器 shield_ctrl/lock_reg/sensor_delay/sensor_en/sensor_dur/enable_krstn 可正常配置 1：寄存器 shield_ctrl/lock_reg/sensor_delay/sensor_en/sensor_dur/enable_krstn 无法再配置		0
[8]		enable sdio0/1		0
[9]		enable sdio2/3		0
[10]		enable sdio4/5		0
[11]		enable sdio6/7		0
[12]		1：sdio0/1 动态模式		0
[13]		1：sdio2/3 动态模式		0
[14]		1：sdio4/5 动态模式		0
[15]		1：sdio6/7 动态模式		0
[23:16]		pullup sdio0-7		0
[25:24]		interval，硬件自动检测周期 0:=sec_inte		0
[26]		lock interval reg		0
[27]		shielding alarm en，tamper 自动检测使能。SDIO 功能使能。		0
[29:28		pu_delay，控制每个 interval 期间，SDIO		0

]		拉高时间。0: always on; 1: 2ms; 2: 8ms; 3: 16ms。		
[31:30]		alarm_delay，警报持续时间门限。0: 31.25us（不推荐使用）; 1: 1ms; 2: 4ms; 3: 8ms。		0

11.3.5LPM_WKUP_TIMER

地址：0xf8408

Bit	功能	说明	R/W	复位值
[0]		传感器检测使能。 [0]: 纽扣电池 1.2v 输出低压自检	R/W	0
[1]		[1]: 纽扣电池 3.3v 输出高压自检	R/W	0
[2]		[2]: 纽扣电池 3.3v 输出低压自检	R/W	0
[3]		[3]: 高温自检		0
[4]		[4]: 低温自检		0
[6:5]		警报持续时间门限，大于门限的信号，将触发 KEY 擦除操作。用于滤除毛刺，防止虚警。 00: 1*(1/32k)=31.25us 01: 8*(1/32k)=250us 10: 32*(1/32k)=1ms 11: 128*(1/32k)=4ms，模拟和软件确认？现在是传感器打开就检测！！！同 1 同 0 未检测		0
[7]		锁定寄存器，配置为 1 后，无法写回 0。 0: 寄存器 shield_ctrl/lock_reg/sensor_delay/sensor_en/sensor_dur/enable_krstn 可正常配置 1: 寄存器 shield_ctrl/lock_reg/sensor_delay/sensor_en/sensor_dur/enable_krstn 无法再配置		0
[8]		enable sdio0/1		0
[9]		enable sdio2/3		0
[10]		enable sdio4/5		0
[11]		enable sdio6/7		0
[12]		1：sdio0/1 动态模式		0
[13]		1：sdio2/3 动态模式		0
[14]		1：sdio4/5 动态模式		0
[15]		1：sdio6/7 动态模式		0
[23:16]		pullup sdio0-7		0
[25:24]		interval，硬件自动检测周期 0:=sec_inte		0
[26]		lock interval reg		0
[27]		shielding alarm en，tamper 自动检测使能。SDIO 功能使能。		0

[29:28]		pu_delay，控制每个 interval 期间，SDIO 拉高时间。0: always on; 1: 2ms; 2: 8ms; 3: 16ms。软件确认？		0
[31:30]		alarm_delay，警报持续时间门限。0: 31.25us; 1: 1ms; 2: 4ms; 3: 8ms。软件确认？		0

11.3.6LPM_GPIO_WKUP

功能：GPIO 高电平唤醒功能使能
地址：0xf8410
说明： 设置 GPIO0 到 GPIO31 的唤醒使能功能。

Bit	功能	说明	R/W	复位值
[0]		1：使能 gpio0 唤醒功能	R/W	0
[1]		1：使能 gpio1 唤醒功能	R/W	0
[2]		1：使能 gpio2 唤醒功能	R/W	0
[3]		1：使能 gpio3 唤醒功能		0
[4]		1：使能 gpio4 唤醒功能		0
[5]		1：使能 gpio5 唤醒功能		0
[6]		1：使能 gpio6 唤醒功能		0
[7]		1：使能 gpio7 唤醒功能		0
				0
				0
[30]		1：使能 gpio30 唤醒功能		0
[31]		1：使能 gpio31 唤醒功能		0

11.3.7LPM_GPIO_WKHI

功能：GPIO 高电平唤醒功能使能说明：
地址：0xf8414
说明： 设置 GPIO32 到 GPIO47 的唤醒使能功能。

Bit	功能	说明	R/W	复位值
[0]		1：使能 gpio32 唤醒功能	R/W	0
[1]		1：使能 gpio33 唤醒功能	R/W	0
[2]		1：使能 gpio34 唤醒功能	R/W	0
[3]		1：使能 gpio35 唤醒功能	R/W	0
[4]		1：使能 gpio36 唤醒功能	R/W	0
[5]		1：使能 gpio37 唤醒功能	R/W	0
[6]		1：使能 gpio38 唤醒功能	R/W	0
[7]		1：使能 gpio39 唤醒功能	R/W	0
[8]		1：使能 gpio40 唤醒功能	R/W	0
[9]		1：使能 gpio41 唤醒功能	R/W	0
[10]		1：使能 gpio42 唤醒功能	R/W	0
[11]		1：使能 gpio43 唤醒功能	R/W	0
[12]		1：使能 gpio44 唤醒功能	R/W	0

[13]		1：使能 gpio45 唤醒功能	R/W	0
[14]		1：使能 gpio46 唤醒功能	R/W	0
[15]		1：使能 gpio47 唤醒功能	R/W	0
[16]				
[17]		0：禁止 rtc timeout，清除 rtc_intr 中断标志 1：rtc timeout 唤醒使能，rtc timeout 中断使能	R/W	
[18]				
[19]				
[20]		0：key 软件可写，硬件警报不自动擦除 key 1：key 软件禁止写入,硬件警报自动擦除 key	R/W	
[21:22]		Sensor 检测时间长度 00：always on 01：2ms 10：8ms 11：16ms	R/W	

11.3.8LPM_SLEEP

功能：总线加密控制
地址：0xf8420

Bit	功能	说明	R/W	复位值
[7：0]		写入"0x5A"后，LPM 进入“SLEEPING”状态。	R/W	0
[31：8]	预留		R/W	0

11.3.9 LPM_CLR_INTR

功能：总线加密控制
地址：0xf8424

Bit	功能	说明	R/W	复位值
[7:0]		写入“0x6C”,清除 lpm_intr。清除 lpm_intr 中断之前，应先清除 lpm_intr 的下级中断，包括 rtc_intr，sensor_alert，shield_alarm，否则会继续进入中断	R/W	0
[31:8]		reserved		

11.3.10 LPM_STATUS

地址：0xf8478

Bit	功能	说明	R/W	复位值
[0:15]	预留		R/W	1
[16]		VDT1.2L 报警		
[17]		VDT3.3H 报警		

[18]		VDT3.3L 报警		
[19]		高温报警		
[20]		低温报警		
[21]		Shielding alarm 0		
[22]		Shielding alarm 1		
[23]		Shielding alarm 2		
[24]		Shielding alarm 3		
[25]		Shielding alarm 4		
[26]		Shielding alarm 5		
[27]		Shielding alarm 6		
[28]		Shielding alarm 7		
[29]		Mesh shielding alarm		
[30]		Sensor alert 高温/低温、3.3L/3.3H/1.2L 有警告并且 sensor dur 期间置 1		
[31]		rtc_timeout 中断		

12 看门狗（WDT）

12.1 看门狗外设时钟

看门狗外设时钟由 HCLK 提供，即看门狗外设时钟频率等于 HLCK 时钟频率。

12.2 计数器（Counter）

看门狗计数器（DWT_CCVR：Watchdog Timer Current Counter Value Register）为递减计数器，即计数器值由预设值递减直至数值为 0。当计数器计数到 0 时，看门狗根据设定模式产生系统复位或中断。

12.3 计数器预设值

看门狗计数器 WD_CONFIG[4:0]预设值由保存，用户可以通 过 WD_STATUS 寄存器设定看门狗计器超时时间。对 WD_KICK 寄存器写 0x5937 将对 DWT_CCVR 寄存器的置 重置为此预设值，完成“喂狗”操作。

12.4启用看门狗

看门狗开启由看门狗控制寄存器 WDT_CONFIG 控制，当 WDT_CONFIG [6] = 1 时看门狗开启。看门狗使能开启后将无法关闭，可通过复位看门狗模块来关闭看门狗。

12.5系统复位中断

看门狗包含 2 种模式：

WDT_CONFIG[5] = 0：系统复位模式

WDT_ CONFIG [5] = 1：中断模式

看门狗计数器计数到 0 后，系统立即产生复位。 中断模式：

在看门狗计数器第 1 次计数到 0 时，会产生看门狗中断（中断源为不可屏蔽中断 NMI），并 重置看门狗计数器到预设值，但不会产生系统复位。此后看门狗计数器会进入下 1 轮递减计数，用户必须在此次计数过程中进行喂狗或清中断处理操作，否则在此次计数至 0 后，系统发生复位。

运行在中断模式中的看门狗，除了使用普通喂狗方式（重置看门狗计数器）外，还可以通过清 除看门狗中断标记完成喂狗。

看门狗中断可以通过以下两种方式清除：

1、重置看门狗计数器（喂狗） 对 WDT_CONFIG 寄存器写 0x5937 后，硬件自动完成“喂狗”操作。

2、读看门狗中断清除寄存器（WDT_ CLEAR） WDT_CLEAR[0] 寄存器进行读操作清除看门狗中断标。

12.6 寄存器

12.6.1看门狗控制寄存器 WDT_CONFIG

功能：控制寄存器
地址：WDT_BASEADDR+0x00（0xf0000）

Bit	功能	说明	R/W	复位值
[4:0]	WDT_load	此 4 个 bit 的中作为 2 的幂计算后的值为 WDT 初始计数值	R/W	0
[5]	WDT 响应方式	0 :计数值溢出后直接产生复位 1:计数值溢出后先产生中断，如果没喂狗则再产生复位	R/W	0
[6]	WDT_EN	WDT 使能寄存器，这一位用于打开或关闭 WDT 功能，当关闭 WDT 功能，看门狗计数器停止计数，这样将不会产生中断或复位。一旦打开 WDT 功能，只能由系统复位关闭	R/W	0

		0 :WDT 功能关闭. 1: WDT 功能打开.		
[31:5]	预留			0

注：BIT[6] (WDT_EN)只能写 1,不能写 0;也即 WDT 功能一旦开启后,只能通过系统复位才能关闭,该寄存器位才能清掉.

12.6.2看门狗中断状态寄存器 WDT_ STATUS

功能：看门狗中断状态寄存器
地址：WDT_BASEADDR+0x04（0xf0004）

Bit	功能	说明	R/W	复位值
[0]	WDT 中断状态	1: 发生 WDT 中断 0 : 没有发生 WDT 中断	R	0
[31:1]	预留			0

12.6.3WDT_ KICK

功能：喂狗寄存器
地址：WDT_BASEADDR+0x08（0xf0008）

Bit	功能	说明	R/W	复位值
[0]	当作为读寄存器的时候：看门狗的中断状态标志	[0]:无 [1]:看门狗中断产生	R	0
[31:1]	预留		R	0
[31:0]	喂狗寄存器 WDT_KICK	此寄存器为喂狗寄存器，只能写固定值 0x5937 实现喂狗	W	0

12.6.4看门狗中断清除寄存器 WDT_ CLEAR

功能：清除 WDT 中断
地址：WDT_BASEADDR+0x0c（0xf000c）

Bit	功能	说明	R/W	复位值
[0]	清除 WDT 中断	向这个寄存器中写 1 清除 WDT 中断	W	0
[31:1]	预留			0

13 SCI7816

13.1 7816 模块简介

SmartCard 接口（支持 EMV Level-1 规范、 ISO7816-3 标准）， 集成 7816 电平转换功能， 可通过寄存器配置输出 3V 和 1.8V， 不支持升压功能，数据通过 IO 口在终端和 IC 卡之间以异步半双工的方式进行双向传输。

- 芯片包含 1 个智能卡接口（Smart Card Interface）
- 支持 ISO7816-3 标准和 EMV Level-1 规范。
- 支持异步 T=0 和 T=1 传输协议;
- 协议时序、 时间参数可配;
- 8 字符深度接收、 发送缓冲;
- 接收、 发送 FIFO 监测中断;
- 中断状态寄存器;

13.2 寄存器描述

13.2.1 SCI7816_MODE

功能：SCI7816 工作模式配置寄存器

地址：SCI7816_BASEADDR +0x00（0xf0400）

Bit	功能	说明	R/W	复位值
[0]	传输协议选择	0: T=0 协议 1：T=1 协议		
[1]	编码选择	0: 正向编码 1：反向编码		
[2]	IO 模式选择	0: 开漏模式 1: 推挽模式(default)		
[4:3]	ETU	00: 无额外 ETU 01: 1 个额外的 ETU 10:2 个额外的 ETU 11:4 个额外的 ETU		
[7:5]	重传次数选择	000: 不重传 001: 1 次 010: 2 次 011: 3 次 100: 4 次 101: 5 次 110: 6 次 111: 7 次		
[8]	重传使能	0:失能	RW	0

	位	1：使能		
[9]	SCI7816 使能位	0：失能 1：使能		
[10]	块保护时间使能位	0：失能 1：使能		
[11]	CWT 计时器使能位	0：失能 1：使能		
[14:12]	CLK 时钟源	PWM(0-7)		
[15]	主机模式使能位	0：失能 1：使能		
[16]	EDC 错误检测使能位	0：失能 1：使能		
[31:17]		预留		

13.2.2 SCI7816_CTRL

功能：SCI7816 控制寄存器

地址：SCI7816_BASEADDR +0x08（0xf0408）

Bit	功能	说明	R/W	复位值
[0]	RX_FIFO 内容清除	写 1 清除 FIFO 内数据	RW	0
[1]	TX_FIFO 内容清除	写 1 清除 FIFO 内数据		
[2]	检测 TS 字节	1:接收到数据为 03 时校验位电平取反 0：不起作用		
[31:3]	预留			

13.2.3 SCI7816_STAT

功能：SCI7816 状态寄存器

地址：SCI7816_BASEADDR +0x0C（0xf040C）

Bit	功能	说明	R/W	复位值
[0]	接收缓冲器状态	0:接收缓冲器空 1:接收缓冲器中不空	R	0
[1]	接收缓冲器状态	0:接收缓冲器不满 1:接收缓冲器满		
[2]	奇偶校验状态	0：奇偶校验正确 1：奇偶校验错误		
[3]	发送缓冲器状态	0：发送缓冲器空 1：发送缓冲器中不空		
[4]	发送缓冲器状态	0：发送缓冲器不满 1：发送缓冲器满		
[5]	重传后奇偶校验状	0：奇偶校验正确 1：奇偶校验错误（在重传功能开		

	态	启时，只有在发送达到重传次数时仍有错误发生，此位才被置位）		
[6]	BGT 超时状态	0：没超时 1：超时（参考：SCI7816_BGT 寄存器说明）		
[7]	CWT 超时状态	0：没超时 1：超时（参考：SCI7816_CWT 寄存器说明）		
[8]	校验位状态	0:检验位正确 1:检验位错误		
[31:9]	预留			

13.2.4 SC7816_INT_IO

功能：SCI7816 中断标志寄存器

地址：SCI7816_BASEADDR +0x10（0xf0410）

Bit	功能	说明	R/W	复位值
[0]	接收完成标志	0:接收未完成 1:接收完成 (中断模式下对应 SCI0_IRQ 中断函数)	RW	0
[1]	发送完成标志	0:发送未完成 1:发送完成 (中断模式下对应 SCI1_IRQ 中断函数)		
[31:2]	预留			

13.2.5 SCI7816_DATA

功能：SCI7816 缓冲寄存器

地址：SCI7816_BASEADDR +0x20（0xf0420）

Bit	功能	说明	R/W	复位值
[7:0]	buffer	在发送或接收模式下分别充当发送或接收 buffer 角色	RW	0
[31:8]	预留			

13.2.6 SCI7816_ETU

功能：SCI7816 ETU 配置寄存器

地址：SCI7816_BASEADDR +0x28（0xf0428）

Bit	功能	说明	R/W	复位值
[12:0]	速率配置	配置 SCI7816 通讯速率	RW	0
[31:13]	预留			

13.2.7 SCI7816_BGT

功能：SCI7816 发送块等待时间配置寄存器

地址：SCI7816_BASEADDR +0x2C（0xf042C）

Bit	功能	说明	R/W	复位值
[4:0]	时间配置	配置 SCI7816 块反向发送时间间隔	RW	0
[31:5]	预留			

13.2.8 SCI7816_CWT

功能：SCI7816 字符等待时间配置寄存器

地址：SCI7816_BASEADDR +0x30（0xf0430）

Bit	功能	说明	R/W	复位值
[23:0]	CWT 重载值	配置 SCI7816CWT 定时值，发送字节完成、接收起始时启动计时。	RW	0
24	使能 CWT	为 1 时 CWT 计时立即生效		
[31:25]	预留			

13.2.9 SCI7816_EDC

功能：SCI7816 错误校验码保存寄存器

地址：SCI7816_BASEADDR +0x34（0xf0434）

Bit	功能	说明	R/W	复位值
[7:0]	LRC 结果	保存 LRC 计算结果。	R	0
[31:8]	预留			

14 定时器(TIMER) 定时器简介

- 1 个Timer 单元，包含9 个独立定时器（Timer0,Timer1，Timer2，Timer3，Timer4，Timer5,Timer6,Timer7 ,Timer8）
  9 个定时器中断源独立，每个定时器单独占1 个中断源
  定时器采用向下计数方式
  每个单元定时器都支持PWM 模式

使用HCLK 时钟频率作为定时器计时钟源

14.1 定时器外设时钟

定时器外设时钟由 HCLK 提供，即定时器时钟频率等于 HCLK 外设时钟频率

14.2 通用定时器

14.2.1通用定时器计数值

当定时器使能后计数值 TIM_CNT 寄存器载入。

14.2.2中断处理

在 TIMER 模式下默认开启中断使能。

14.3 PWM 模式

Timer 单元的 9 个独立定时器均可编程产生 PWM 信号。 当用户设定 PWM_CTRL 中对应比特为“1 后，定时器进入 PWM 工作模式。此时 PWM 由 TIM_PCNT 和 TIM_NCNT 寄存器分别控制高电平及低电平周期翻转输出。

14.4 寄存器描述

14.4.1TIM_PCNT

功能：在 PWM 模式下配置 PWM 高电平持续时间，在定时器模式下作为重复计数的重载值
地址：0xf0c00
说明：1、PCNT 寄存器位宽为 4byte（32bit）
2、第 n 通道的 PCNT 寄存器地址为 0xf0c00+n*8（通道编号从 0 开始）
3、定时器模式下此寄存器的值将作为重复计数的重载值

寄存器	地址
TIM0_PCNT	0xf0c00
TIM0_NCNT	0xf0c04
TIM 1_PCNT	0xf0c08
TIM 1_NCNT	0xf0c0c
TIM 2_PCNT	0xf0c10

TIM 2_NCNT	0xf0c14
TIM 3_PCNT	0xf0c18
TIM 3_NCNT	0xf0c1c
TIM 4_PCNT	0xf0c20
TIM 4_NCNT	0xf0c24
TIM 5_PCNT	0xf0c28
TIM 5_NCNT	0xf0c2c
TIM 6_PCNT	0xf0c30
TIM 6_NCNT	0xf0c34
TIM 7_PCNT	0xf0c38
TIM 7_NCNT	0xf0c3c
TIM 8_PCNT	0xf0c40
TIM 8_NCNT	0xf0c44

Bit	功能	说明	R/W	复位值
[31:0]	配置 PWM 高电平持续时间	如果 TIM_CTRL 模式设为位为 0（PWM 模式），该寄存器功能为配置 PWM 高电平持续时间，与 NCNT 一起决定占空比。如果 PWM_CTRL 模式设为位为 1（定时模式），作为重复技术重载值寄存器。	R、W	0

14.4.2TIM_NCNT

功能：配置 PWM 低电平持续时间
地址： 0xf0c48
说明：1、NCNT 寄存器位宽为 4byte（32bit）
2、第 n 通道的 NCNT 寄存器地址为 0xf0c48+n*8（通道编号从 0 开始），各通道地址见 PWM_PCNT

Bit	功能	说明	R/W	复位值
[31:0]	配置 PWM 低电平持续时间	配置 PWM 低电平持续时间，与 PCNT 一起决定占空比	R、W	0

注：详细使用见 PWM_CNT 注释

14.4.3TIM_CTRL

功能：PWM 控制寄存器
地址：0xf0c30
说明：从第 0 位开始，每四个 Bit 控制一个定时器（TIM0 到 TIM7）。
0-3bit：TIM0
4-7bit：TIM1
8-11bit：TIM2
12-15bit：TIM3
16-19bit：TIM4
20-23bit：TIM 5
24-27bit：TIM6
28-31bit: TIM7
TIM_CTRL 的 0-3bit 含义如下，从第 4bit 开始每 4bit 功能与 0-3bit 对应相同，分别控制对应的 TIM

Bit	功能	说明	R/W	复位值
[0]	模块使能	0:关闭模块 1:使能模块 注：先配置好其他 bit 再使能模块	R、W	0
[1]	初始电平	0:初始为低电平 1:初始为高电平	R、W	0
[2]	定时器模式	0:PWM 模式 1:TIMER 模式	R、W	0
[3]	自动重载	0:.关闭自动重载 1: .开启自动重载 只有在 TIMER 模式才有意义，TIMER 模式下此 bit 为 1 则在计数到 0 时自动从 PCNT 寄存器重载计数值	R、W	0
[31:4]		每四个 bit 控制一个定时器		

14.4.4TIM_CTRL1

功能：PWM 控制寄存器
地址：0xf0c4c
说明： TIM8 配置寄存器

Bit	功能	说明	R/W	复位值
[0]	模块使能	0:关闭模块 1:使能模块 注：先配置好其他 bit 再使能模块	R、W	0
[1]	初始电平	0:初始为低电平 1:初始为高电平	R、W	0
[2]	定时器模式	0:PWM 模式 1:TIMER 模式	R、W	0
[3]	自动重载	0:.关闭自动重载 1: .开启自动重载 只有在 TIMER 模式才有意义，TIMER 模式下此 bit 为 1 则在计数到 0 时自动从 PCNT 寄存器重载计数值	R、W	0

14.4.5TIM_CNT

功能：PWM 模式下配置 PWM 周期;定时模式当前计数器值
地址：PWM_BASEADDR+0x50（0xf0c34）
说明：1、CNT 寄存器位宽为 4byte（32bit）
2、第 n 通道的 CNT 寄存器地址为 0xf0c50+n*4（通道编号从 0 开始）

寄存器	地址
TIM0_CNT	0xf0c50
TIM1_CNT	0xf0c54
TIM2_CNT	0xf0c58
TIM3_CNT	0xf0c5c
TIM4_CNT	0xf0c60
TIM5_CNT	0xf0c64

TIM6_CNT	0xf0c68
TIM7_CNT	0xf0c6c
TIM8_CNT	0xf0c70

Bit	功能	说明	R/W	复位值
[31:0]	PWM 模式下配置 PWM 周期;定时模式当前计数器值	PWM 模式：PCNT、NCNT 与 CNT 需满足 CNT=PCNT+NCNT; 定时器模式：保存当前计数器的值	R、W	0

注：周期 T=（52+PWM_PCNT*28+ PWM_NCNT*28）ns

15 实时时钟（RTC）

15.1 RTC 简介

实时时钟是一个独立的定时器。RTC 模块拥有一组连续计数的计数器。RTC 模块和 RTC 相关配 置寄存器都处于电池电源域，即主电源掉电对 RTC 没有任何影响，RTC 依旧保持正常计数。

15.2 RTC 特性

以秒作为计时单位（通过配置产生秒中断）；

非独立中断源，SEC 部分为同一中断源；

15.3 RTC 寄存器

RTC 的所有寄存器器属于 LPM 模块寄存器，不能将值直接读出，读出后保存在 SYSCTRL_LPM_RDATA(addr: (0xf7040[31:0])中。

读 XREG 寄存器步骤： 1. 先读寄存器 ：int temp =XREG;

2.再读实际的值： int reg_value = SYSCTRL_LPM_RDATA;

15.3.1RTC 使能

RTC 使能位包含在寄存器 LPM_GPIO_WKHI 中

功能：开启 RTC

地址：0xf8414

Bit	功能	说明	R/W	复位值
[16:0]		在 LPM 部分用到此功能，	R/W	0
[17]	使能 RTC	1：使能 RTC（RTC 使能之后默认开启），中断源属于 SEC 部分。	R/W	0
[31:18]		在 LPM 部分用到此功能，	R/W	0

15.3.2RTC 当前计数值寄存器

LPM_RTC_CNT
功能：设置
地址：0xf747c

Bit	功能	说明	R/W	复位值
[31:0]	读写计数值	设置计数器的初始计数值，读取当前的计数值。	R、W	0

15.3.3RTC 计数校准寄存器

LPM_SECMAX
功能:设置 RTC 的计数一次的时间
地址:0xf740c

Bit	功能	说明	R/W	复位值
[15:0]	设置 RTC 计数一次的时间	如果设置RTC计数寄存器一秒钟计数一次，在时钟为 32k 的情况下，则设置该寄存器的值为 0x8000;一秒计数一次	R、W	0

15.3.4 RTC 闹钟设置寄存器

LPM_WKUP_TIMER
功能:超时的计数值，当写入的值等于 计数值（LPM_RTC_CNT）时触发中断。
地址: f8408

Bit	功能	说明	R/W	复位值
[31:0]	设置计数中断值	当设置的值等于RTC寄存器的寄存器的时候产生中断。	R、W	0

15.3.5RTC 中断状态寄存器

LPM_STATUS
功能:读取 RTC 的中断状态。

地址:0xf8478

Bit	功能	说明	R/W	复位值
[30：0]	保留在其他功能	具体的功能请参照 LPM 部分。	R	0
[31]	RTC 中断产生标志	0:未产生中断 1:产生 RTC 的中断	R	0

。

16 DMA 控制器（DMAC）

16.1 DMA 简介

直接存储器存取(DMA)用来提供在外设和存储器之间或者存储器和存储器之间的高速数据传输。无须 CPU 干预，数据可以通过 DMA 快速地移动，这就节省了 CPU 的资源来做其他操作。

注意：用户不可将敏感数据，通过 DMA 的方式发送到芯片外部，在进行密钥运算或敏感操作时应关闭 DMA 防止误操作将敏感信息发送到芯片外部。

16.2DMA 主要特性

支持内存到内存、内存到外设、外设到内存之间的传输。

16.3 DMA 的使用

DMA 模块设立了六个独立的通道，以下为 1 到 6 通道:

DMACH_SPID0

DMACH_SPID1

DMACH_UART0

DMACH_UART1

DMACH_IICD

DMACH_MEMCP

SPI、UART、IIC 这几个外设的任一数据收发场景都需要使用到 DMA 模块。

DMA 的使用：

1. 使用 DMA_SRC_ADDR 寄存器设置发送数据内存的起始地址。
2. 使用 DMA_DEST_ADDR 寄存器设置接收数据的起始地址。
3. 使用 DMA_LEN 寄存器设置接收和发送能存的大小。
4. 使用 DMA_CONFIG 开启 DMA 使能。

完成以上步骤后，数据开始发送。DMA 的状态寄存器将保存 DMA 的状态寄存器:DMA_STATUS bit0 等于 1 时表示 DMA 处于空闲状态，表示数据发送完成。

16.4 DMA 的中断

从内存到内存的通道 DMACH_MEMCP 中断类型：数据 copy 完成进入中断。

16.5 DMA 寄存器描述

16.5.1通道 x 源地址寄存器 DMA_SRC_ADDR

功能: DMA 源地址的指针

地址：0xf8800 + N*0x100

说明：N 为 DMA 通道序号;

Bit	功能	说明	R/W	复位值
[31:0]	配置 DMA 发送始地址	在 DMA 发送数据时，要发送的数据的起始地址写入该寄存器。	R/W	0

16.5.2通道 x 目的地址寄存器 DMA_DEST_ADDR

地址：0xf8804 + N*0x100

说明：N 为 DMA 通道序号;

Bit	功能	说明	R/W	复位值
[31:0]	配置 DMA 接收数据的起始地址	在 DMA 接收数据时，将保存数据的起始地址写入该寄存器	R/W	0

16.5.3DMA 长度陪住寄存器 DMA_LEN

功能：设置 DMA 的长度（byte）

地址：0xf8808 + N*0x100

说明：N 为 DMA 通道序号;

Bit	功能	说明	R/W	复位值
[15:0]	发送数据长度	DMA 发送数据时，目标地址数据的长度	R/W	

[31:16]	接收数据长度	DMA 接收数据时，源地址数据的长度		

16.5.4DMA 控制寄存器 DMA_CONFIG

功能：配置某外设的 DMA 通道

地址：0xf880c + N*0x100

说明：N 为 DMA 通道序号;

Bit	功能	说明	R/W	复位值
[0]	UART RX DMA 回环	1：开启 UART DMA 回环接收数据	R/W	0
[1]	DMA 中断	1:开启 DMA 中断		0
[30]	清除 DMA 中断	1: 清楚 DMA 中断	R/W	0
[31]	DMA start	1: 开启 DMA		0

16.5.5DMA 状态寄存器 DMA_STATUS

功能： DMA 状态寄存器

地址：0xf8810 + N*0x100

说明：N 为 DMA 通道序号;

Bit	功能	说明	R/W	复位值
[0]	DMA 中有无数据	[0]DMA 有数据 [1]DMA 空闲	R/W	0
[31:1]	预留		R/W	0

16.5.6DMA_RPTR

功能：DMA 读指针

地址：0xf8814 + N*0x100

说明：N 为 DMA 通道序号;

Bit	功能	说明	R/W	复位值
[31:0]	预留	存放 DMAx 发送 BUF 读指针	R/W	0

16.5.7DMA_WPTR

功能：DMA 读指针

地址：0xf8818 + N*0x100

说明：N 为 DMA 通道序号;

Bit	功能	说明	R/W	复位值
[31:0]	预留	存放 DMAx 接收 BUF 写指针	R/W	0

17 UART

17.1 UART 简介

通用异步收发器(UART)提供了一种灵活的方法与使用工业标准 NRZ 异步串行数据格式的部设备之间进行全双工数据交换。UART 利用波特率发生器提供宽范围的波特率选择。

17.2 UART 外设时钟

UART 外设时钟由内部的 RC 分频为固定 48M,不会根据系统时钟的变化而改变,时钟默认开启。

17.3 中断

用户可以通过 UART_CTRL 配置中断类型。

UART 外设可产生的中断类型如下：

发送数据完成中断；

接收数据有效中断；

接收数据超时中断；

接收数据中断模式下，触发中断数据的个数是可以通过 UART_CTRL 寄存器配置的。

17.4DMA 支持

UART 外设使用 DMA 功能可以有的效减少系统中断，提高数据传输效率。每个 UART 外设可以 使

用 2 个 DMA 通道，分别用来接收和发送数据。

17.5 UART 控制寄存器 UART_CTRL

功能：UART0 与 UART1 配置寄存器 UART0_CTRL UART1_CTRL

地址：UART0_CTRL: 0xf8b1c UART1_CTRL: 0xf8c1c

Bit	功能	说明	R/W	复位值
[0]	使能 rx	0：失能 Rx 1：使能 Rx (初始化寄存器时先失能再使能)	R/W	0
[1]	设置奇偶校验	0：Parity_Even 1：Parity_Odd	R/W	0
[2]	设置字长	0：8 bits 字长 1：9 bits 字长	R/W	0
[3]	设置停止位停止位	0：一个停止位 1：两个停止位	R/W	0
[4]	设置流控	0：无流控 1：使能流控	R/W	0
[5]	Scard	0：关闭智能卡模式 1：开启智能卡模式	R/W	0
[6]	设置字长	0：8 bits 字长 1：9 bits 字长	R/W	0
[7]	设置重置波特率标志	0： 使用自动波特率 1： 重置波特率	R/W	0
[15:8]	配置 rx 中断触发数据个数	配置 RX 触发中断数据长度（0 为不触发）	R/W	0
[30:16]	设置波特率	写入的值为系统时钟除以要设置的波特率	R/W	0
[31]	使能 tx 中断	0：失能数据发送完成中断中断 1：使能 tx 中断	R/W	0

17.5.1UART_INTR

功能：配置中断超时时间

地址：UART0_INTR: 0xf8b20 UART1_INTR: 0xf8c20

说明：两次接收的时间间隔超过此时间，则产生中断

Bit	功能	说明	R/W	复位值
[15:0]	设置接收超时中断时间	超时时间为 值*48 个 clock	R/W	0
[31:16]		reserved	R	0

17.5.2数据接收寄存器 UART_RDATA

功能：接收数据寄存器

地址：UART0_RDATA：0xf8b24 ；UART1_RDATA：0xf8c24.

说明：接收数据寄存器,

Bit	功能	说明	R/W	复位值
[7：0]	接收数据	读取接收数据。在接收数据之前，要配置好接收数据的内存，即将接收数据的起始地址写入 DMA_DEST_ADDR（DMACH_UARTx）。	R	0

17.5.3状态寄存器 UART_STATUS

功能：UARTx 状态标识寄存器

地址：0xf8b28+ N*0x100

说明：N 为 UART 序号;N=0,则配置 UART0,N=1； N=1,配置 UART1

Bit	功能	说明	R/W	复位值
[31:16]	接收到的数据个数	当前接收数据 buffer 中的数据个数(从 UART_RDATA 中每读出一个数，自减 1)		
[15:3]	保留	默认为 0		
[2]	接收数据 buf 接近 full 标识	[0] [1]rx near full		
[1]	接收数据 buf 满	[0]rx buf 满 [1]rx buf 未满		
[0]	未接收数据	[0]rx buf 有数据 [1]rx buf 为空		

18 SPI 接口

18.1SPI 简介

串行外设接口(SPI)允许芯片与外部设备以半/全双工、同步、串行方式通信。此接口支持主、从模式，并为外部从设备提供通信时钟(SCK)。

18.2 SPI 主要特点

- SPI 时钟由 HCLK 提供，即 SPI_CLK = HCLK；
- Master 模式与 Slave 模式独立地址操作；
- Master 模式支持全双工、单工收、单工发、EEPROM 模式支持协议，多个 Master 冲突探测；
- DMA 支持；

18.3 SPI 功能描述

18.3.1SPI 外设时钟及要求

SPI 时钟由 HCLK 提供，时钟复位状态为开启；

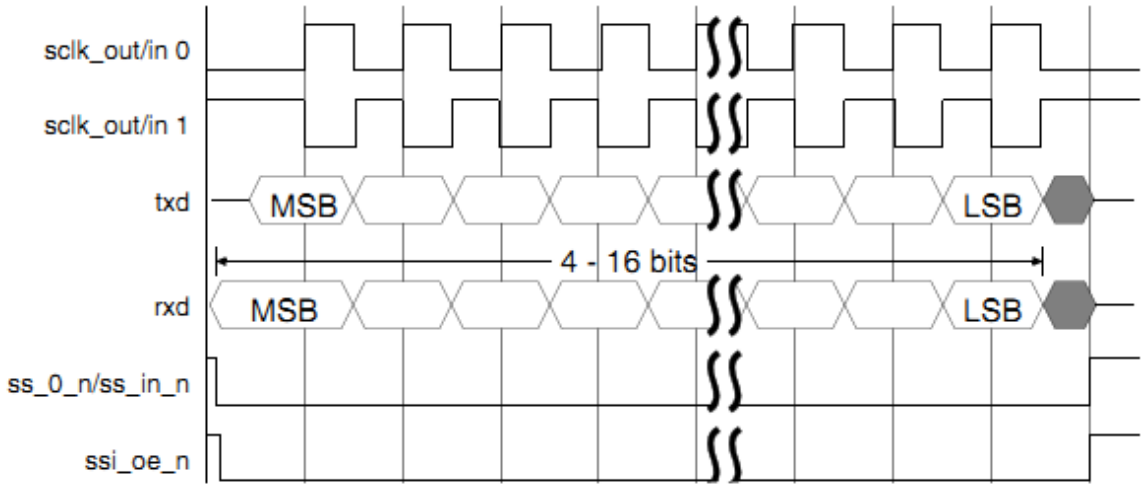
SPI_CLK：SPI 接入时钟频率，为 HCLK ；

SPI_M_CLK：SPI 主模式总线时钟频率,最高为主时钟的一半，最多支持 128 分频；

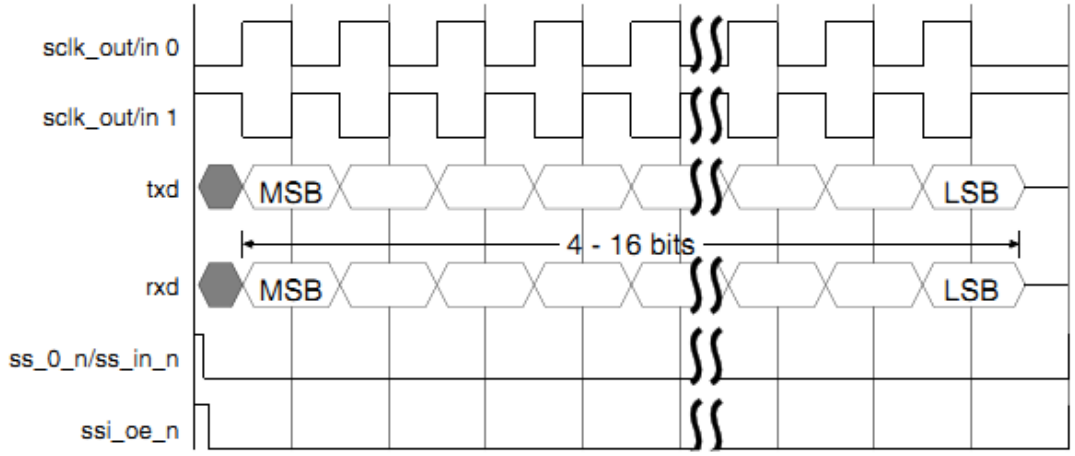
18.3.1.1 MotorolaSPI 通行协议

常用 Motorola SPI 通讯协议支持的四种通讯模式，能够实现全双工通讯。系统上电默认采用模式 0 工作方式。

SCPH = 0:



SCPH = 1:



sclk_out/ in：总线时钟，out：SPI 为主设备输出 CLK。in：SPI 为从设备

输入 CLK sclk_out/ in = 0： （CPHA）=0

ss_0_n/ss_in_n:片选信号,s_0_n:SPI 为主设备时输出片选。s_in_n:SPI 为主设备时输入片选ss_oe_n:SPI 为从模式时输出使能选项。

SPI 协议规定的 4 中通讯格式说明如下：

模式 0：时钟极性（CPOL）=0，时钟相位（CPHA）=0，该模式下串行同步时钟的空闲状 态为低电平，芯片将在串行同步时钟的第一个跳变沿（上升沿）采样，芯片默认为该模式；

模式 1：时钟极性（CPOL）=0，时钟相位（CPHA）=1，该模式下串行同步时钟的空闲状 态为低电平，芯片将在串行同步时钟的第二个跳变沿（下降沿）采样；

模式 2：时钟极性（CPOL）=1，时钟相位（CPHA）=0，该模式下串行同步时钟的空闲状 态为高电平，芯片将在串行同步时钟的第一个跳变沿（下降沿）采样；

模式 3：时钟极性（CPOL）=1，时钟相位（CPHA）=1，该模式下串行同步时钟的空闲状态为高电平，芯片将在串行同步时钟的第二个跳变沿（上升沿）采样。

18.3.1.2 SPI 主模式配置

主模式下使用 SPIx:配置流程：

SPI 时钟默认打开，通过 SPID_CTRL 寄存器配置主模式，时钟极性，时钟相位。

18.3.1.3 主模式数据收发

SPI 只通过 DMA 收发数据，收发数据之前必须手动分配内存，详见 DMA 部分。

18.3.1.4 SPI 寄存器描述

18.3.1.4.1 SPI 控制寄存器 SPID_CTRL

功能：SPIx 控制寄存器

地址：0xf891C+ N*0x100

说明：N 为 SPI 序号;N=0,则配置 SPI0, N=1,配置 SPI1

Bit	功能	说明	R/W	复位值
[2:0]	配置时钟	2*2^Value([2:0])为分频系数 001： 作为 Master 总线时钟频率的 1/2. 48M 主时钟时，SPI 总线时钟为 12M 010： 4 分频 011： 8 分频 100： HCLK/2/16 为时钟频率 .... 111： HCLK/2/128 为时钟频率	R/W	0
[3]	模式	0: Master Mode 1：Slave Mode		
[4]	CPHA	0: 奇数边沿采样 1: 偶数边沿采样		
[5]	CPOL	0: 空闲时时钟 IO 为电平 1: 空闲时时钟 IO 为高电平		
[6]	SPI_reset	1: reset SPI		
[7]		预留		
[14:8]	Delay	发送与接收反向间隔（value*16 个 clock）		
[31:15]	预留			

19 USB

19.1 USB 简介

USB 外设实现了 USB2.0 全速总线和 APB1 总线间的接口 USB 外设支持 USB 挂起/恢复操作，支持低速和全速模式（8 endpoints）。

19.2 USB 主要特点

- 符合 USB2.0 全速设备的技术规范；
- CRC（循环冗余校验）生成/校验，反向不归零（NRZI）编码/解码和位填充；

19.3 USB 功能描述

USB 模块为 PC 主机和微控制器所实现的功能之间提供了符合 USB 规范的通信连接。PC 主机和微控制器之间的数据传输是通过共享一专用的数据缓冲区来完成的，该数据缓冲区能被 USB 外设直接访问。

- 支持 USB2.0
- 支持主机协商协议(HNP)和会话请求协议(SRP)
- 支持 SRP 协议的 USB 全速/低速
- 提供 512 字节的专用 RAM 和高级的 FIFO 管理
- 通过软件为不同的 FIFO 配置不同的 RAM 区域，以便灵活有效的使用 RAM
- 允许动态的分配存储区
- 不限定 FIFO 的长度（不强制 2 的幂次长度， 可以连续的使用存储区）
- 允许相同端点号（IN/OUT 端点共用同一个 FIFO， 更加有效的使用存储区）
- 拥有 EP0-EP3 4 个端点
- USB 状态寄存器清除状态时，往相应的 bit 位写入 1

19.4 USB 存储器描述

19.4.1USB_CONFIG

功能：USB 模块控制寄存器

地址：USB _BASEADDR +0x00（0xf8600）

Bit	功能	说明	R/W
[0]	ISO enable	1:enable ISO for endpoint 2 OUT	RW
[1]	ISO enable	1:enable ISO for endpoint 2 IN	RW
[3:2]	pad bias	USB pad bias control	RW
[4]	enable	0:关闭 USB 模块 1:开启 USB 模块	RW
[5]	Speed	0:低速模式（1.5M） 1:全速模式（12M）	RW
[6]	resume	1：复位 USB 模块	RW

[7]	Wakeup	0:关闭端点唤醒 1:开启端点唤醒	RW

19.4.2USB_IRQ_MASK1

功能：USB 中断控制寄存器 1

地址：USB _BASEADDR +0x01（0xf6001）

Bit	功能	说明	R/W
[0]	EP0 packet 中断使能	0:开启端点 0 out FIFO 接收到 packet 中断功能 1:屏蔽端点 0 out FIFO 接收到 packet 中断功能	RW
[1]	EP1 packet 中断使能	0:开启端点 1 out FIFO 接收到 packet 中断功能 1:屏蔽端点 1 out FIFO 接收到 packet 中断功能	RW
[2]	EP2 packet 中断使能	0:开启端点 2 out FIFO 接收到 packet 中断功能 1:屏蔽端点 2 out FIFO 接收到 packet 中断功能	RW
[3]	EP3 packet 中断使能	0:开启端点 3 out FIFO 接收到 packet 中断功能 1:屏蔽端点 3 out FIFO 接收到 packet 中断功能	RW
[4]	Setup 包中断使能	0:开启接收到 setup 包中断功能 1:屏蔽接收到 setup 包中断功能	RW
[5]	暂停中断使能	0:开启暂停状态中断功能 1:屏蔽暂停状态中断功能	RW
[6]	NAK 中断使能	0:开启 NAK 中断功能 1:屏蔽 NAK 中断功能	RW
[7]	STALL 中断使能	0:开启 STALL 中断功能 1:屏蔽 STALL 中断功能	RW

19.4.3USB_IRQ_MASK2

功能：USB 中断控制寄存器 2

地址：USB _BASEADDR +0x02（0xf6002）

Bit	功能	说明	R/W
[0]	EP0 IN FIFO 中断使能	0:开启端点 0 IN FIFO 空中断功能 1:屏蔽端点 0 IN FIFO 空中断功能	RW
[1]	EP1 IN FIFO 中断使能	0:开启端点 1 IN FIFO 空中断功能 1:屏蔽端点 1 IN FIFO 空中断功能	RW
[2]	EP2 IN FIFO 中断使能	0:开启端点 2 IN FIFO 空中断功能 1:屏蔽端点 2 IN FIFO 空中断功能	RW
[3]	EP3 IN FIFO 中断使能	0:开启端点 3 IN FIFO 空中断功能 1:屏蔽端点 3 IN FIFO 空中断功能	RW
[4]	EP0 OUT FIFO 中断使能	0:开启端点 0 OUT FIFO 空中断功能 1:屏蔽端点 0 OUT FIFO 空中断功能	RW

[5]	EP1 OUT FIFO 中断使能	0:开启端点 1 OUT FIFO 空中断功能 1:屏蔽端点 1 OUT FIFO 空中断功能	RW
[6]	EP2 OUT FIFO 中断使能	0:开启端点 2 OUT FIFO 空中断功能 1:屏蔽端点 2 OUT FIFO 空中断功能	RW
[7]	EP3 OUT FIFO 中断使能	0:开启端点 3 OUT FIFO 空中断功能 1:屏蔽端点 3 OUT FIFO 空中断功能	RW

19.4.4USB_IRQ_MASK3

功能：USB 中断控制寄存器 3

地址：USB _BASEADDR +0x03（0xf6003）

Bit	功能	说明	R/W
[0]	EP0 IN FIFO 中断使能	0:开启端点 0 IN FIFO 满中断功能 1:屏蔽端点 0 IN FIFO 满中断功能	RW
[1]	EP1 IN FIFO 中断使能	0:开启端点 1 IN FIFO 满中断功能 1:屏蔽端点 1 IN FIFO 满中断功能	RW
[2]	EP2 IN FIFO 中断使能	0:开启端点 2 IN FIFO 满中断功能 1:屏蔽端点 2 IN FIFO 满中断功能	RW
[3]	EP3 IN FIFO 中断使能	0:开启端点 3 IN FIFO 满中断功能 1:屏蔽端点 3 IN FIFO 满中断功能	RW
[4]	EP0 OUT FIFO 中断使能	0:开启端点 0 OUT FIFO 满中断功能 1:屏蔽端点 0 OUT FIFO 满中断功能	RW
[5]	EP1 OUT FIFO 中断使能	0:开启端点 1 OUT FIFO 满中断功能 1:屏蔽端点 1 OUT FIFO 满中断功能	RW
[6]	EP2 OUT FIFO 中断使能	0:开启端点 2 OUT FIFO 满中断功能 1:屏蔽端点 2 OUT FIFO 满中断功能	RW
[7]	EP3 OUT FIFO 中断使能	0:开启端点 3 OUT FIFO 满中断功能 1:屏蔽端点 3 OUT FIFO 满中断功能	RW

19.4.5USB_ADDR

功能：usb 地址寄存器

地址：USB _BASEADDR +0x04（0xf6004）

Bit	功能	说明	R/W
[6:0]	Usb addr	存放 sub 设备地址	RW
[7]	-	Disable broadcast （address 0）packet receive	RW

19.4.6USB_TRG

功能：usb 发送控制寄存器

地址：USB _BASEADDR +0x10（0xf6010）

Bit	功能	说明	R/W
[0]	send EP0 data	此位写 1 开始发送端点 0 INFIFO 中的数据	RW
[1]	send EP1 data	此位写 1 开始发送端点 1 INFIFO 中的数据	RW
[2]	send EP2 data	此位写 1 开始发送端点 2 INFIFO 中的数据	RW
[3]	send EP3 data	此位写 1 开始发送端点 3 INFIFO 中的数据	RW
[5:4]	Reply zero packet	端点 0，1 发送空包	RW
[6]	Reply zero packet	端点 2 发送空包	RW
[7]	Reply zero packet	端点 3 发送空包	RW

19.4.7USB_STALL

功能：USB STALL 控制寄存器

地址：USB _BASEADDR +0x11（0xf6011）

Bit	功能	说明	R/W
[0]	EP0 STALL	此位写 1 设置 EP0 为 STALL 状态	RW
[1]	EP1 IN STALL	此位写 1 设置 EP1 IN 为 STALL 状态	RW
[2]	EP1 OUT STALL	此位写 1 设置 EP1 OUT 为 STALL 状态	RW
[3]	EP2 IN STALL	此位写 1 设置 EP2 IN 为 STALL 状态	RW
[4]	EP2 OUT STALL	此位写 1 设置 EP2 OUT 为 STALL 状态	RW
[5]	EP3 IN STALL	此位写 1 设置 EP3 IN 为 STALL 状态	RW
[6]	EP3 OUT STALL	此位写 1 设置 EP3 OUT 为 STALL 状态	RW
[7]	-	预留	-

19.4.8USB_CLEAR

功能：USB FIFO 清除控制寄存器

地址：USB _BASEADDR +0x12（0xf6012）

Bit	功能	说明	R/W

[0]	Clear EP1 IN data	此位写 1 清空 EP1 IN 数据	RW
[1]	Clear EP1 OUT data	此位写 1 清空 EP1 OUT 数据	RW
[2]	Clear EP2 IN data	此位写 1 清空 EP2 IN 数据	RW
[3]	Clear EP2 OUT data	此位写 1 清空 EP2 OUT 数据	RW
[4]	Clear EP3 IN data	此位写 1 清空 EP3 IN 数据	RW
[5]	Clear EP3 OUT data	此位写 1 清空 EP3 OUT 数据	RW
[6]	USB reset	USB 协议 reset	RW
[7]	USB reset1	清 USB 寄存器内存	RW

19.4.9USB_EP

功能：USB 端点收发数据

地址：USB _BASEADDR +0x18（0xf6018+x）

Bit	功能	说明	R/W
[7:0]	端点 0 读写数据	从此寄存器读数据或者写数据;当发送数据的时候写入寄存器,收数据从该寄存器读数据。	R/W
[15:8]	端点 1 读写数据	从此寄存器读数据或者写数据;当发送数据的时候写入寄存器,收数据从该寄存器读数据。	R/W
[23:16]	端点 2 读写数据	从此寄存器读数据或者写数据;当发送数据的时候写入寄存器,收数据从该寄存器读数据。	R/W
[31:17]	端点 3 读写数据	从此寄存器读数据或者写数据;当发送数据的时候写入寄存器,收数据从该寄存器读数据。	R/W

19.4.10 USB_EP_LEN

功能：端点数据长度

地址：USB _BASEADDR +0x20（0xf6020+x）

USB_EP0_LEN 0xf6020+0x20

USB_EP1_LEN 0xf6020+0x21

USB_EP2_LEN 0xf6020+0x22

USB_EP3_LEN 0xf6020+0x23

19.4.11 USB_STATUS

功能：USB 中断控制寄存器 1

地址：USB _BASEADDR +0x01（0xf6026）

Bit	功能	说明	R/W
[0]	EN0 OUT 中断状态	1：EN0 OUT 中断产生。	RW
[1]	EN1 OUT 中断状态	1：EN1 OUT 中断产生。	RW
[2]	EN2 OUT 中断状态	1：EN2 OUT 中断产生。	RW
[3]	EN3 OUT 中断状态	1：EN3 OUT 中断产生。	RW
[4]	Setup 包中断状态	1：接收到 setup 中断产生	RW
[5]	暂停中断状态	1 暂停状态中断产生	RW
[6]	NAK 中断状态	1：NAK 中断产生	RW
[7]	STALL 中断状态	0:开启 STALL 中断功能 1:屏蔽 STALL 中断功能	RW

19.4.12 USB_FIFO_EMPTY

功能：USB FIFO 空寄存器

地址：USB _BASEADDR +0x27（0xf6027）

Bit	功能	说明	R/W
[0]		1:端点 0 IN FIFO 空中断产生	RW
[1]		1:端点 1 IN FIFO 空中断产生	RW
[2]		1:端点 2 IN FIFO 空中断产生	RW
[3]		1:端点 3 IN FIFO 空中断产生	RW
[4]		1:端点 0 OUT FIFO 空中断产生	RW
[5]		1:端点 1 OUT FIFO 空中断产生	RW
[6]		1:端点 2 OUT FIFO 空中断产生	RW
[7]		1:端点 2 OUT FIFO 空中断产生	RW

19.4.13 USB_FIFO_FULL

功能：USB FIFO 满中断状态寄存器

地址：USB _BASEADDR +0x28（0xf6028）

Bit	功能	说明	R/W
[0]		1:端点 0 IN FIFO 满中断产生	RW
[1]		1:端点 1 IN FIFO 满中断产生	RW
[2]		1:端点 2 IN FIFO 满中断产生	RW
[3]		1:端点 3 IN FIFO 满中断产生	RW
[4]		1:端点 0 OUT FIFO 满中断产生	RW
[5]		1:端点 1 OUT FIFO 满中断产生	RW
[6]		1:端点 2 OUT FIFO 满中断产生	RW
[7]		1:端点 2 OUT FIFO 满中断产生	RW

19.5USB 复位

19.5.1外设模式下

当充当外围设备和检测到 USB 其他重置条件时，该装置将执行以下操作：

- 设置FADDR 为0
- 设置索引为 0
- 刷新所有端点的 FIFO
- 清除所有控制/状态寄存器
- 启用所有的端点中断

产生一个复位中断 当应用软件驱动 YC3121 收到一个复位中断，应该关闭所有打开的管道并等待总线枚举的开始。

19.6 连接/断开

相关连接和断开 YC3121 无论是在的特定行为，还是在对等通信外设模式中都可以使用。

在外设模式下操作时，该设备连接到主机上不产生中断。当主机终止会话时产生一个断开中断。

19.7 规划方案

这与以下各部分看，该装置控制所述 YC3121 芯需要执行的操作与在该影响下核心操作的各个方面。在整个讨论中，控制装置被假定为运行某些固件的单片机，但它可以定制硬布线逻辑块。

19.7.1USB 中断处理

当在有一个 USB 中断 CPU 断开时，需要读取中断状态寄存器来确定是哪个端点造成的中断，并跳转到相应的程序。如果是多个端点造成的中断，端点 0 先服务，其次是其他终端。

19.8 VBUS 活动

USB 规范定义了一系列涉及在点对点通信设备需要对应的阈值：

VBUS 有效（要求在4.4 和4.75）；

会话结束（要求在0.2V 和0.8V 之间）；

（在一特定装置中使用的实际阈值需要通过一系列比较器，外部 YC3121 核心。外部 YC3121 核心采取相应的 VBUSVALID，AVALID 和 SESSEND 依据 VBUS 级别设置输入高或低） 其中这些阈值是关键的，其中 CPU 控制 YC3121 需要响应的方式取决于设备是’A’设备或’B’设备和发生其他事件的情况。所需操作总结如下文

19.8.1作为‘B’设备操作

1）VBUS>会话有效值（即 Vbus[1:0](DevCTL[D4:D3])=10B，会话位（DevCtl.D0）设置)。这表明 从“A”设备活动。YC3121 将设置会话位，并采取DPPULLDOWN 输出低电平来断开D+线下拉 电阻。

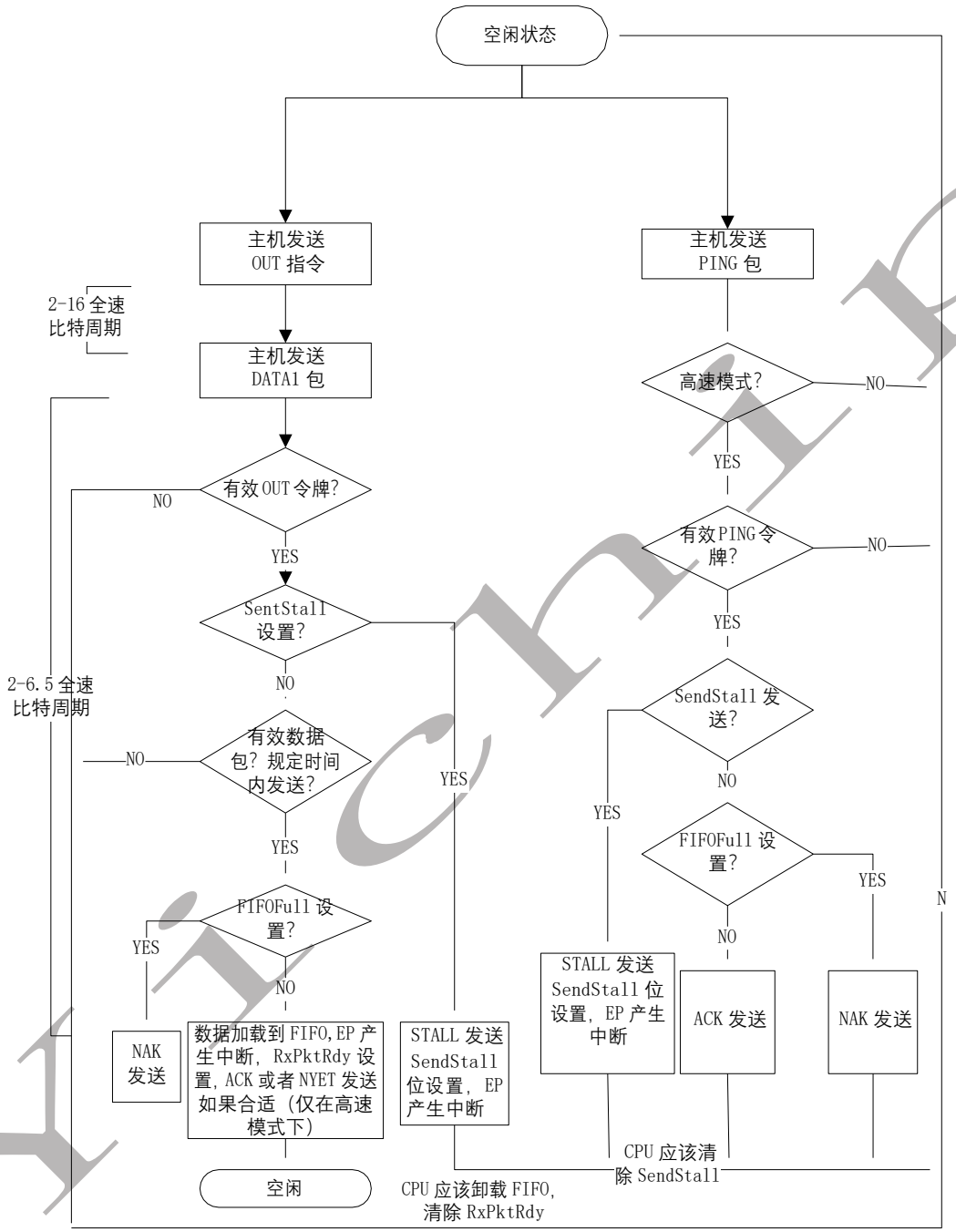
2）VBUS<会话有效值而会话位保持设置即Vbus[1:0](DevCTL[D4:D3])=01B会话位DevCtl.D0） 设置)。这表明“A”设备已经失去权力（或断开连接）YC3121 将清除会话位（DevCtl.D0）并 产生一个断开中断（IntrUSB.D5）。CPU 结束会话。

3） VBUS<会话有效值（即 Vbus[1:0](DevCTL[D4:D3])=00B）。这是下一个“B”设备可以发起会话请 求的条件。如果会话位（DevCtl.D0）设置，SE0 在总线上执行 2 毫秒之后，YC3121 将首先脉 冲数据线，然后脉冲VBUS（采取CHRGVBUS 高点）开始调整计划。

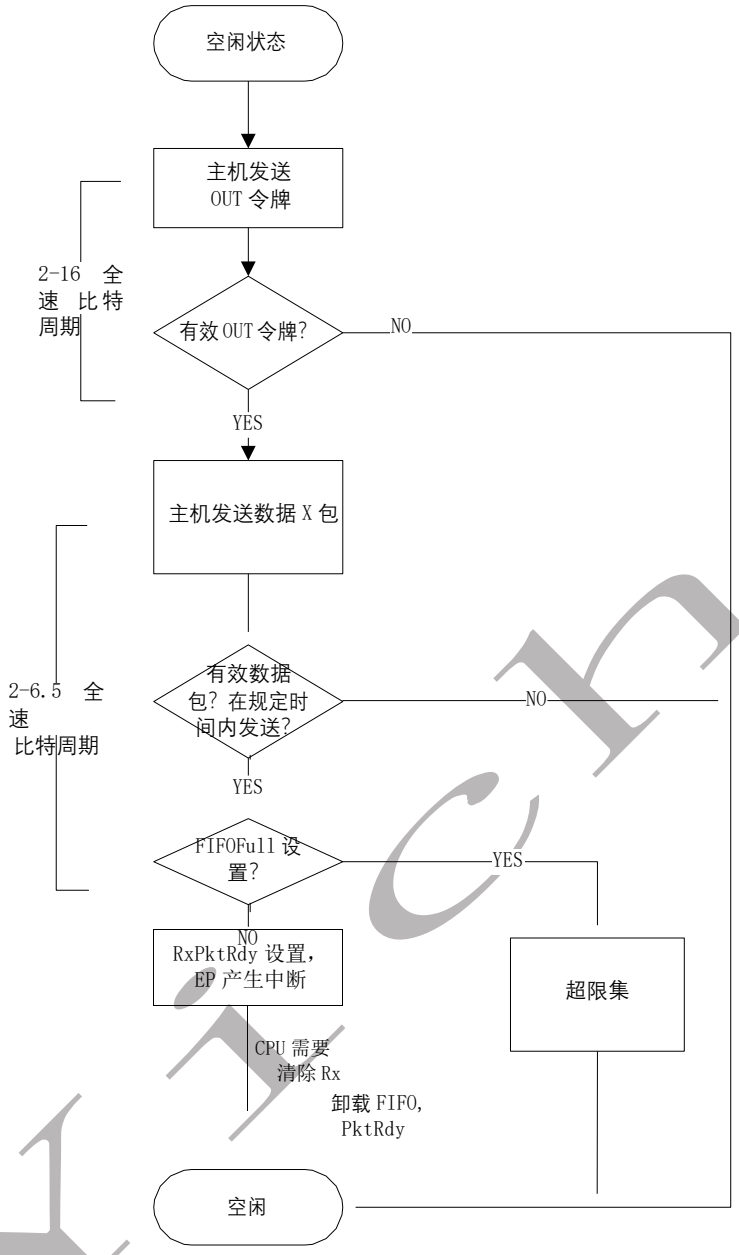
19.9 FIFO

YC3121 对每个端点的 FIFO 的大小都是固定的。端点 0 与端点 1 的大小为 64Byte，端点 2 的大小为 128Byte，端点 3 的大小为 256Byte。

19.10BULK/低带宽中断事务



19.11全速/低带宽等时事务



20 ADC（SAR_ADC）

20.1 ADC 简介

ADC 采样率为 600KHz，采样精度为 10 比特。

20.2 ADC 特性

ADC 有 8 个通道，最高采样率为 600KHz，最高采样精度为 10 比特，AD 的参考电压为 1.2V,校准值写在 OTP 中，电压范围为 0~1.2V。

20.3 ADC 寄存器

20.3.1 ADC_ENABLE

功能：ADC 使能寄存器
地址：0xc8906
描述：使能 ADC

Bit	功能	说明	R/W	复位值
[0]		clkpll_en_clk2dac	R/W	0
[1]		clkpll_en_clk2rccal	R/W	0
[2]		clkpll_en_clk2saradc	R/W	0
[3]		misc_saradc_en	R/W	0
[4]		misc_saradc_en_biasgen	R/W	0
[5]		misc_saradc_en_constgm	R/W	0
[6]		misc_saradc_en_reg	R/W	0
[7]		cic_en	R/W	0

20.3.2 ADC_CTRL0

功能：ADC 控制寄存器
地址：0xc8970
描述：配置 ADC 的模式。

Bit	功能	说明	R/W	复位值
[2:0]		misc_saradc_ibc_ibuf<2:0>	R/W	111

[3]		misc_saradc_ibc_cmbuf	R/W	1
[6:4]		misc_saradc_ibc_refbuf<2:0>	R/W	111
[7]		misc_saradc_ibc_refbuf2	R/W	1

20.3.3ADC_CTRL1

功能：ADC 控制寄存器
地址：0xc8971
描述：ADC 的通道选择

Bit	功能	说明	R/W	复位值
[1:0]		misc_saradc_clksel	R/W	11
[3:2]		misc_saradc_ibuf_bw	R/W	11
[6:4]	ADC 模式选择	misc_saradc_mode: 000: gpio mode 001:gpio diff mode 010:hvin mode 011: vinlpm mode 110: temperature mode	R/W	000
[7]		misc_saradc_ibuf_en_rc	R/W	1

20.3.4ADC_CTRL2

功能：ADC 控制寄存器
地址：0xc8972
描述：ADC 的通道选择

Bit	功能	说明	R/W	复位值
[1:0]		misc_saradc_ibuf_gc	R/W	11
[3:2]		misc_saradc_refbuf_vref_ctrl	R/W	10
[5:4]		misc_saradc_rega_vctrl	R/W	10
[7:6]		misc_saradc_regd_vctrl	R/W	10

20.3.5ADC_CTRL3

功能：ADC 控制寄存器
地址：0xc8973
描述：ADC 的通道选择

Bit	功能	说明	R/W	复位值
[2:0]	ADC 通道选择	misc_saradc_sel_ch_s: 000：通道 1-GPIO37 001：通道 2-GPIO38	R/W	0

		.... 110：通道 7-GPIO43 111：通道 8-GPIO44		
[5:3]		misc_saradc_vctrl_biasgen	R/W	
[6]		pmu_chgpump_en	R/W	
[7]		pmu_chgpump_hv	R/W	

20.3.6ADC 数据寄存器

功能：ADC RDATA
地址：0xf850e
描述：读取 ADC 数据

Bit	功能	说明	R/W	复位值
[0]		预留	R	0
[10：1]		data	R	
[15:11]		预留	R	

21 充电模块（CHARGE）

21.1 充电模块简介

- 用于给 3.7V 锂电池充电；
- 支持最大 100mA 的充电电流；
- 电池充满的电压为 4.15±0.05V；
- 电池充满后电压降到 4.05V 之后将重新给电池充电。

21.2 充电模块特性

当电池电压低于 4.15V 时采用恒流充电（100mA），当电池电压达到 4.15V 时采用恒压充电（4.15V），电流将逐渐减小，当电流小于 10mA 认为充满，将不再给电池充电。寄存器 SYSCTRL_STATUS，可查询充电的状态。

21.3充电模块寄存器

地址：0xC8129

说明：充电模块寄存器属于本田 bt core LPM 域下的寄存器，不能直接读写

Bit	功能	说明	R/W	复位值
[6:0]	Buck control			
[7]	充电使能位	1: enable charger	R/W	1

地址：0xC812A

Bit	功能	说明	R/W	复位值
[7:0]	输出电压细调	Charger CV mode output voltage fine-tune control in 50mV/steps 00000001: coarse voltage-350mV 00000010: coarse voltage-300mV … 00010000: coarse voltage-150mV (default) ... 10000000: coarse voltage		

地址：0xC812C

Bit	功能	说明	R/W	复位值
[3:0]	Chgr_tch<3:0>	Charger test point select		
[6:4]	输出电压粗调	Charger CV mode output voltage coarse-tune control in 180mV/steps 000: 4.19V 100: 4.36V (default) 110: 4.54V 111: 4.74V	R/W	100
[9:7]	chgr_rdn<2:0>输出的	chgr_rup chgr_rdn I_charge 111 000 94.91mA 011 000 78.26mA 001 000 72.47mA 000 000 69.87mA 111 100 120.5mA 111 110 137.4mA 111 111 147.9mA		

		（char_rup:0xC812D[7:5]）		

地址：0xC812D

Bit	功能	说明	R/W	复位值
[4:2]	预留			100
[7:5]	chgr_rup	See description of chgr_rdn		111

22 开关机电路

22.1 开关电路简介

开关机电路：当 HVIN 为 5V 输入时，HVOUT 是否输出由 HV_OPTION（默认为低）,CHGIN,KEYPOWER 共同控制。HVOUT 有输出的条件如下图所示（HVOUT 可输出则为开机状态）。寄存器 SYSCTRL_ STATUS 可查询 CHGTIN 和 KEYPOWER 的状态。