AC636X 用户手册

珠海市杰理科技股份有限公司
Zhuhai Jieli Technologyco.,LTD

2020 年 3 月

修改记录

版本	更新日期	描述
V1.0	2020-3-13	初始版本
V1.1	2020-11-11	更新第 3 章输入/输出（I/O）内容

目 录

序言

AC696X 系列芯片是集成三模蓝牙及 FM 收发的系统级 SOC，支持高性能低功耗的蓝牙及音频应用。

芯片内置蓝牙调制解调器、基带及模拟 RF 模块，支持蓝牙 V2.1/V4.2/V5.1 版本及低功耗蓝牙连接待机，通信距离长，待机时间久。芯片集成 FM 立体声收发，性能与单独的外挂 FM 芯片相仿，支持蓝牙后台及蓝牙与 FM 发送同时工作与 FM 接收分时工作。芯片集成高性能音频 AD/DA，支持 APE、FLAC 等无损音频解码，支持通话回声消除及降噪。AC696X 内置 192Mhz 高性能浮点 DSP，嵌入低延时 cache，方便开发，支持 SDcard、USB、FLASH 等多个外设。

AC696X 主推 BluetoothSmart 和 SmartReady 应用方案，并提供了蓝牙双模的单芯片解决方案，减少设计的工作量让产品更快赢得市场。

AC696X 集成了低成本、超低功耗、2.4G 射频模块，并提供极低的射频活动功耗和 MCU 功耗，支持低功耗模式，出色的电池使用寿命使其适合功耗敏感的应用。

AC696X 提供了完整的 FCC 与 BQB 的认证，使客户在该系列下开发自己的产品时减少了认证的费用。

AC696X 集成了一整套的蓝牙、FM 音频解决方案，为用户开发提供了方便的环境，为产品提供了无限的可能。

第 1 章. 总体介绍

CPU

❖32 位浮点 DSP，最高 192MHz

Memory

❖片上集成了 128K 字节 SRAM

❖16K 4-Way Icache

中断控制器

❖64 个中断源，8 级可编程中断优先级

❖支持 27 个外部 I/O 中断，其中 8 个可低功耗唤醒@@

❖带软中断（虚拟中断）功能，优先级可以配置

数字 I/O

❖最多 27 个可编程数字 I/O 引脚

❖不同的功能引脚有不同的电流选择，最大拉/灌电流为 64 mA（HD IO）24mA（其它 IO）

❖可配置上拉 10K、下拉 10K 功能

复杂设备

❖Full Speed USB OTG 控制器，除 EP0 外带四个 In EP、四个 Out EP

❖2 路立体声音频 DAC，支持直推耳机

❖1 路立体声音频 ADC

❖带有 1 路 MIC 放大电路

❖FM 调频立体声接收系统，覆盖范围 76-108MHz，步进 50/100KHz（与蓝牙分时使用）

❖FM 调频立体声发送系统，覆盖范围 76-108MHz，步进 50/100/200KHz（可与蓝牙同时使用）

❖V2.1/ V4.2/ V5.1 版本蓝牙

数字模块

❖6 个 32 位异步分频可重载定时器，可用作计时、捕捉，PWM 模式 (Timer0/1/2/3)

❖硬件看门狗（Watchdog）

❖UART0/1/2 串口控制器,支持 DMA

❖SPI0/1/2 控制器，支持 1/2/4 线（SPI1/2 只支持 1/2 线），支持 DMA

❖SPI FLASH 控制器，只支持跑代码

❖SD0/1 host 控制器，支持 1/ 4 线，支持 DMA

❖IIC 控制机，支持主从机

❖正交解码器 0/1/2，可同时工作

❖AUDIO IIS 0/1 接口，支持片外音频 DAC/ADC，每个接口最大支持 4 路立体声

❖EMI 控制器，支持 DMA 发送

❖SPDIF 音频输入接口（不包含模拟放大）

❖INPUT channel 及 OUTPUT channel，为部分模块提供任意的 IO 输入输出选择

模拟模块

❖10 位精度 16 通道 ADC（ADC Key/电压检测等）

❖LVD，支持掉电保护

❖PMU，支持软开关功能，蓝牙 sniff 低功耗，支持电池充电

❖PLL，中心频率为 480MHz

❖RC，内部 RC 时钟，频率在 16MHz 左右

❖LRC，内部低温漂 RC 时钟，频率在 32kHz 左右

❖高精度高速 OSC（12/24/40M）

❖64 位 EFUSE

工作范围

❖工作电压 2.2~5.0V

❖工作温度（-20° C 至+85° C）

应用领域

❖蓝牙音箱

❖蓝牙车机

❖蓝牙声霸

❖蓝牙点烟器

封装

❖LQFP48(7*7)

❖QFN20(3*3)

❖SOP16

第 2 章. 中断系统

2.1. 特征

1. 中断源

中断源可分为系统中断源和外设中断源。

中断号	中断类型	说明
63-4	外设中断源	优先级可配，外设中断入口，可扩展到 250 个
3	系统中断源	内核定时器 tick_timer
2	系统中断源	内核调度入口，不受 IE,IP 和 GIE 控制，使用 syscall 进入
1	系统中断源	内核异常入口，不受 IE,IP 和 GIE 控制
0	系统中断源	仿真调试入口，不受 IE,IP 和 GIE 控制，使用 bkpt 进入

2、中断控制寄存器 ICFGx：

每个外设中断源都分配 4bit 中断控制位 ICFGx[3:0]，bit3-1 对应 IP，bit0 对应 IE。具体见中断表。优先级 IP 有 3bit，共 8 级，0 代表最低，7 代表最高。进入中断后，硬件会自动屏蔽比其优先级低的中断入口，例如当优先级为 1 的中断产生后，将被屏蔽优先级小于 1 或等于 1 的中断，直到中断退出。中断使能 IE，只要将相应 IE 的控制位打开即可。

3、总中断：

为了方便程序快速打开和关中断。CPU 增设了 GIE0(用户 GIE)，GIE1(操作系统 CLI_GIE)。当 GIE0 于 GIE1 同时为 1 时，总中断才打开。

4、中断入口：

中断入口从中断 BASE 地址，每 4 个 byte 对应一个中断，存放相应服务程序的地址。中断发生时，CPU 会从相应中断入口取中断服务程序的入口地址，跳到该中断服务程序。

5、清外设模块内部的中断即可清除外设中断源。

6、置软中断：写 ILAT_SET[7:0]或使用 asm(“swi #u3”)分别对应软中断 7-0 清软中断：写 ILAT_CLR[7:0]清相应软中断

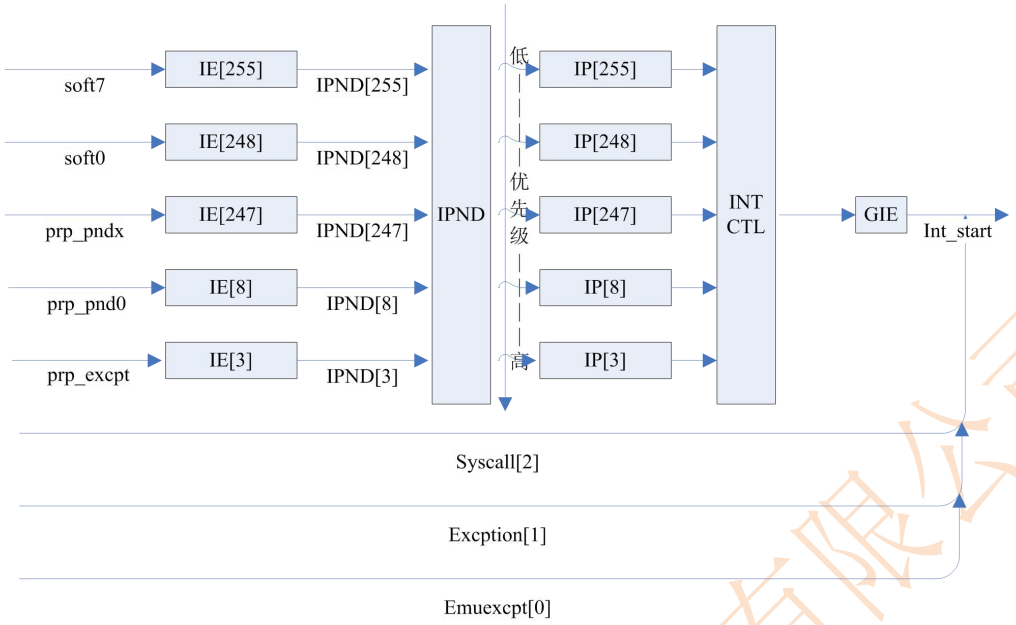


图 2-1 中断系统示意图

2.2. 中断表

中断号	{IP[2:0],IE}	中断入口地址	中断源
63	ICFG7[31:28]	BASE + 中断号×4	soft[3]
62	ICFG7[27:24]	BASE + 中断号×4	soft[2]
61	ICFG7[23:20]	BASE + 中断号×4	soft[1]
60	ICFG7[19:16]	BASE + 中断号×4	soft[0]
59	ICFG7[15:12]	BASE + 中断号×4	
58	ICFG7[11:08]	BASE + 中断号×4	
57	ICFG7[07:04]	BASE + 中断号×4	
56	ICFG7[03:00]	BASE + 中断号×4	
55	ICFG6[31:28]	BASE + 中断号×4	
54	ICFG6[27:24]	BASE + 中断号×4	
53	ICFG6[23:20]	BASE + 中断号×4	CTM
52	ICFG6[19:16]	BASE + 中断号×4	PWM
51	ICFG6[15:12]	BASE + 中断号×4	SPDIF
50	ICFG6[11:08]	BASE + 中断号×4	RDEC2
49	ICFG6[07:04]	BASE + 中断号×4	RDEC1
48	ICFG6[03:00]	BASE + 中断号×4	DCP_INT(DMA_COPY)
47	ICFG5[31:28]	BASE + 中断号×4	FMTX_HWFE
46	ICFG5[27:24]	BASE + 中断号×4	GPC_INT
45	ICFG5[23:20]	BASE + 中断号×4	SBC_INT
44	ICFG5[19:16]	BASE + 中断号×4	SPI2_INT
43	ICFG5[15:12]	BASE + 中断号×4	FMRX_HWFE

42	ICFG5[11:08]	BASE + 中断号×4	CHX_PWM
41	ICFG5[07:04]	BASE + 中断号×4	MCTMRX
40	ICFG5[03:00]	BASE + 中断号×4	AES
39	ICFG4[31:28]	BASE + 中断号×4	BT_EVENT
38	ICFG4[27:24]	BASE + 中断号×4	BT_BLE
37	ICFG4[23:20]	BASE + 中断号×4	OSC_SAFE
36	ICFG4[19:16]	BASE + 中断号×4	ALNK1
35	ICFG4[15:12]	BASE + 中断号×4	PD_TMR1_PND
34	ICFG4[11:08]	BASE + 中断号×4	PD_TMR0_PND
33	ICFG4[07:04]	BASE + 中断号×4	EQ
32	ICFG4[03:00]	BASE + 中断号×4	FFT
31	ICFG3[31:28]	BASE + 中断号×4	SRC_INT
30	ICFG3[27:24]	BASE + 中断号×4	BT_LOFC
29	ICFG3[23:20]	BASE + 中断号×4	BT_DBG
28	ICFG3[19:16]	BASE + 中断号×4	BT_CLKN
27	ICFG3[15:12]	BASE + 中断号×4	BT_BREDR
26	ICFG3[11:08]	BASE + 中断号×4	LRCT
25	ICFG3[07:04]	BASE + 中断号×4	RDEC0
24	ICFG3[03:00]	BASE + 中断号×4	PLNK
23	ICFG2[31:28]	BASE + 中断号×4	SARADC
22	ICFG2[27:24]	BASE + 中断号×4	IIC
21	ICFG2[23:20]	BASE + 中断号×4	PAP
20	ICFG2[19:16]	BASE + 中断号×4	UART2
19	ICFG2[15:12]	BASE + 中断号×4	UART1
18	ICFG2[11:08]	BASE + 中断号×4	UART0
17	ICFG2[07:04]	BASE + 中断号×4	SDC1
16	ICFG2[03:00]	BASE + 中断号×4	SDC0
15	ICFG1[31:28]	BASE + 中断号×4	SPI1
14	ICFG1[27:24]	BASE + 中断号×4	SPI0
13	ICFG1[23:20]	BASE + 中断号×4	PORT
12	ICFG1[19:16]	BASE + 中断号×4	AUDIO
11	ICFG1[15:12]	BASE + 中断号×4	ALNK0
10	ICFG1[11:08]	BASE + 中断号×4	RTC
09	ICFG1[07:04]	BASE + 中断号×4	FUSB_CTL
08	ICFG1[03:00]	BASE + 中断号×4	FUSB_SOF
07	ICFG0[31:28]	BASE + 中断号×4	TMR3
06	ICFG0[27:24]	BASE + 中断号×4	TMR2
05	ICFG0[23:20]	BASE + 中断号×4	TMR1
04	ICFG0[19:16]	BASE + 中断号×4	TMR0
03	ICFG0[15:12]	BASE + 中断号×4	tick_tmr
02	ICFG0[11:08]	BASE + 中断号×4	syscall
01	ICFG0[07:04]	BASE + 中断号×4	exception(misalign/watchdog)
00	ICFG0[03:00]	BASE + 中断号×4	emuexcpt

表 2-1. 中断对应表

2.3. 寄存器说明

1、CPU 内部 SFR：ICFG

icfg 是中断配置寄存器，包括总中断配置及一些中断标志。

Bit	Name	RW/df	Description
31-27	reserved	ro/5'h0	预留,读为 5'h0。
26-24	int_ip	ro/3'h0	上一次中断锁存的优先级
23-16	int_num	ro/8'h0	上一次中断锁存的中断号
15-12	os_flag	rw/4'h0	操作系统标志存储位
11	switch_dis	rw/0	写 1 禁止模式切换时自动切换栈指针。
10	user_mode	ro/0	内部执行模式控制位。
9-8	GIE[1:0]	rw/2'h0	总中断 = GIE[1] & GIE[0]；同时打开才能进入中断。 GIE[1] ：CPU 内部总中断 1。（服务于操作系统，是 CLI_GIE） GIE[0] ：CPU 内部总中断 0。(服务于普通中断，是 RTI_GIE)
7-0	ibit[7:0]	ro/8'h0	中断阻挡标志位。 ibit[7]=1 时，阻挡所有中断。 ibit[6]=1 时，阻挡中断优先级小于 7 的所有中断。 ibit[5]=1 时，阻挡中断优先级小于 6 的所有中断。 ibit[4]=1 时，阻挡中断优先级小于 5 的所有中断。 ibit[3]=1 时，阻挡中断优先级小于 4 的所有中断。 ibit[2]=1 时，阻挡中断优先级小于 3 的所有中断。 ibit[1]=1 时，阻挡中断优先级小于 2 的所有中断。 ibit[0]=1 时，阻挡中断优先级小于 1 的所有中断。

2、CPU 内部 SFR：IPMASK

IPMASK 是中断优先级屏蔽寄存器，用于快速屏蔽优先级小于 IPMASK 的所有中断。

Bit	Name	RW/df	Description
31-3	reserved	rw/29'h0	预留,读为 29'h0。
2-0	ipmask	rw/3'h0	ipmask=0 时，关闭优先级屏蔽功能。 ipmask=1 时，屏蔽中断优先级小于 1 的所有中断。 ipmask=2 时，屏蔽中断优先级小于 2 的所有中断。 …… ipmask=7 时，屏蔽中断优先级小于 7 的所有中断。

3、CPU 执行模式与指针

第一种情况：switch_dis = 0；

CPU 具有两种执行模式，分别为：用户模式，系统模式（中断时进入系统模式）。

1、sys_mode ：(user_mode==0) → 对应指针 SSP

2、usr_mode ：(user_mode==1) → 对应指针 USP

两种执行模式分别对应两支不同的栈指针，当模式切换时硬件会将系统栈指针切换到当前模式的指针。

在超级模式时：读写 SP 相当于读写 SSP，模式切换时会把 SP 备份到 SSP

在用户模式时：读写 SP 相当于读写 USP，模式切换时会把 SP 备份到 USP

第二种情况：switch_dis = 1;

CPU 具有两种执行模式，分别为：用户模式，系统模式（中断时进入系统模式）。

1、sys_mode ：(user_mode==0) → 对应指针 SP

2、usr_mode ：(user_mode==1) → 对应指针 SP

硬件不自动切指针，SSP/USP 可视为临时寄存器，由程序员控制指针切换。

4、CPU 进入 IDLE

CPP 代码里面写： asm violate(“idle”);

ASM 代码里面写： idle;

第 3 章. 输入/输出（I/O）

3. 1. 概述

3.1.1. 引脚功能复用表

端口	可复用功能												
PA0（超强驱动）	P00	ROM 后	SD 卡供电脚	SDPG			SPIDIF_OUT		ALNK_DAT0_A/ALNK_DAT0_B		ADC0	CLKOUT0	UART1TXC
PA1	P01	ROM 后			MIC				ALNK_DAT1_A		ADC1	PWM4	UART1RXC
PA2	P02	ROM 后		SDOCLK_C	MIC_BIAS				ALNK_MCLK_A/ALNK_DAT1_B			CAP3	
PA3	P03	ROM 后		SDODAT_C	AMUX0L	Q-decoder0_0	PLNK_SCLK	BT_Active	ALNK_SCLK_A/ALNK_DAT2_B	UART1_CTS	ADC2	PWM5	UART2TXA
PA4	P04	ROM 后		SDOCMD_CE	AMUX0R	Q-decoder0_1	PLNK_DAT1	Wlan_Active	ALNK_LRCK_A/ALNK_DAT3_B	UART1_RTS	ADC3	TMR4	UART2RXA
PA5	P05	ROM 后					SPDIF_IN_A	BT_priority	ALNK_DAT2_A/ALNK_SCLK_B	IIC_SCL_D		PWM0	UART0TXA
PA6	P06	ROM 后					SPDIF_IN_B	BT_Freq	ALNK_DAT3_A/ALNK_LRCK_B	IIC_SDA_D	ADC4	CAP4	UART0RXA
PB0（高压）	PP0	ROM 后		SDOCLK_D		SDTAP_CLKC	SPDIF_IN_C		SPI1CLKA			TMR5	UART0TXB
PB1（上拉）	PP1	ROM 后	长按 Reset			SDTAP_DATC	SPDIF_IN_D		SPI1DOA	FM_TXA	ADC5	TMR2	UART0RXB
PB2（高压）	PP2	ROM 后		SDOCMD_D		Q-decoder1_0			SPI1DIA			CAP0	UART2TXB
PB3（超强驱动）	PP3	ROM 后	主要用于 FM 发射	SDODAT_D		Q-decoder1_1				FM_TXB	ADC6	PWM2	UART2RXB
PB4	PF1	ROM 中		SDODAT_F	LVD	Q-decoder2_0	spi0_DAT2A(2)	SFC_DAT2A(2)			ADC7	CLKOUT1	UART2TXC/UART2RXC
PB5（高压）	PP5	ROM 中	maskrom 串口升	SDODAT_B		Q-decoder2_1			SPI2DIA			PWM3/CAP1	UART0TXC/UART0RXC
PB6	PP6	ROM 后		SDOCMD_BF	AMUX1L	SDTAP_CLKD			SPI2CLKA	IIC_SCL_C	ADC8	TMR3	UART1TXA
PB7	PP7	ROM 后		SDOCLK_BF	AMUX1R	SDTAP_DATD			SPI2DOA	IIC_SDA_C	ADC9	PWM5	UART1RXA
PC1	PE1	ROM 中		PA_EN			spi0_CSB	SFC_CSB	SPI2DIB			TMR0	UART1TXB
PC2	PE2	ROM 中		LNA_EN			spi0_DIB(1)	SFC_DIB(1)	ALNK_MCLK_B		ADC10	CAP5	UART1RXB
PC3	PE3	ROM 中		SDODAT_A			spi0_DAT2B(2)	SFC_DAT2B(2)	SPI1DIB			CAP2	UART0TXD/UART0RXD
PC4	PF0	ROM 中		SDOCMD_A		SDTAP_CLKA	spi0_DAT3AB(3)	SFC_DAT3AB(3)	SPI1CLKB	IIC_SCL_B	ADC11	PWM1	UART2TXD
PC5	PE5	ROM 后		SDOCLK_AE		SDTAP_DATA			SPI1DOB	IIC_SDA_B	ADC12	TMR1	UART2RXD
PD0	PF4	ROM 中					spi0_CLKAB	SFC_CLKAB					
PD1	PF5	ROM 中					spi0_DOAB(0)	SFC_DOAB(0)					
PD2	PF3	ROM 中					spi0_DIA(1)	SFC_DIA(1)					

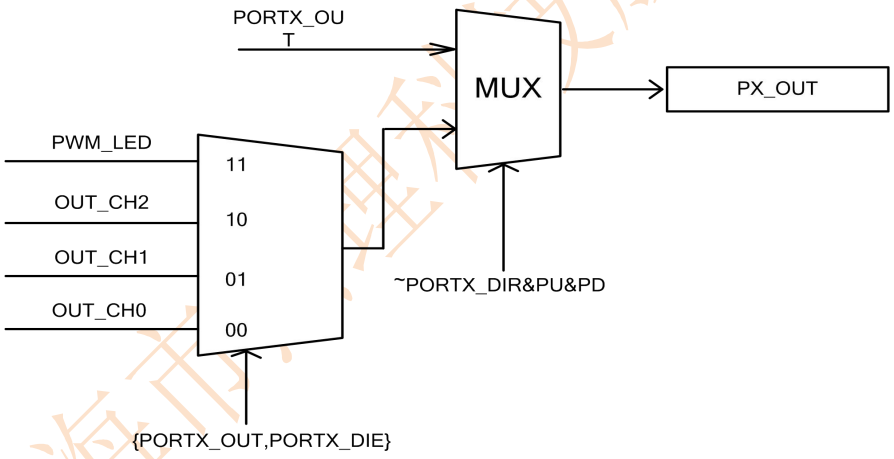
PD3	PF2	ROM 中					spi0_CSA	SFC_CSA					
PD4 （超强驱动）		ROM 中	Flash 供电脚										
USBDP （下拉）		ROM 中				SDTAP_CLKB	ISP_CLK（mode_det0）		SPI2CLKB	IIC_SCL_A	ADC13		UART1TXD
UDBDM （下拉）		ROM 中		SDODAT_E		SDTAP_DATB	ISP_DI（mode_det1）		SPI2DOB	IIC_SDA_A	ADC14		UART1RXD

3.2. 寄存器说明

PAP/SD/SPI/IIC/UART 等外设的端口可支持 remapping，即可配置为使用不同的 IO 端口。**IOMAP_CONx** 寄存器用于设置此类的 IO remapping。

注意：下文中的 IO 不代表实际 IO 数量，请参照 IO_MAPPING 来选择对应的 IO，如果 CHANNEL 选择到了 IO_MAPPING 上不存在的 IO，结果不可预测。

IO 的输出选通如下图：



3.2.1. IO 功能设置寄存器

1.**IOMAP_CON0**: *IO mapping control register0*

Bit	Name	RW	Default	Description
31:28	-	-	-	预留
27	TMR5_CAP_IOS	rw	0	TIMER5 边沿捕抓 IO 选择 0: PC2 1: input channel 2
26	TMR4_CAP_IOS	rw	0	TIMER4 边沿捕抓 IO 选择

				0: PA6 1: input channel 2
25	TMR3_CAP_IOS	rw	0	TIMER3 边沿捕抓 IO 选择 0: PA2 1: input channel 2
24	TMR2_CAP_IOS	rw	0	TIMER2 边沿捕抓 IO 选择 0: PC3 1: input channel 2
23	TMR1_CAP_IOS	rw	0	TIMER1 边沿捕抓 IO 选择 0: PB5 1: input channel 2
22	TMR0_CAP_IOS	rw	0	TIMER0 边沿捕抓 IO 选择 0: PB2 1: input channel 2
21	TMR5_CIN_IOS	rw	0	TIMER5 外部时钟输入 IO 选择 0 :PB0 1: pll_12m
20	TMR4_CIN_IOS	rw	0	TIMER4 外部时钟输入 IO 选择 0 :PA4 1: input channel 4
19	TMR3_CIN_IOS	rw	0	TIMER3 外部时钟输入 IO 选择 0 :PB6 1: pll_12m
18	TMR2_CIN_IOS	rw	0	TIMER2 外部时钟输入 IO 选择 0 :PB1 1: input channel 4
17	TMR1_CIN_IOS	rw	0	TIMER1 外部时钟输入 IO 选择 0 :PC5 1: pll_24m
16	TMR0_CIN_IOS	rw	0	TIMER0 外部时钟输入 IO 选择 0 :PC1 1: input channel 4
15	--			预留
14	SPDIF_IN_D_IOS	rw	0	SPDIF_IN_D 输入选择 0： input channel 9 1： PB1
13	SPDIF_IN_C_IOS	rw	0	SPDIF_IN_C 输入选择 0： input channel 8 1： PB0
12	SPDIF_IN_B_IOS	rw	0	SPDIF_IN_B 输入选择 0： input channel 10 1： PA6

11	SPDIF_IN_A_IOS	rw	0	SPDIF_IN_A 输入选择 0： input channel 11 1： PA5
10:8	SD0_IOS	rw	0	SD0 IO 设置，详见 IO spec 文档 0, 选择组别 A 1, 选择组别 B 2, 选择组别 C 3, 选择组别 D 4, 选择组别 E 5, 选择组别 F
7:5	IRFLT_OS	rw	0	与 IRFLT_EN 一起组成 TIMR 捕获端选择 010：IRFLT 输出至 timer0 的捕获端； 011：IRFLT 输出至 timer1 的捕获端； 100：IRFLT 输出至 timer2 的捕获端； 101：IRFLT 输出至 timer3 的捕获端； 110：IRFLT 输出至 timer4 的捕获端； 111：IRFLT 输出至 timer5 的捕获端； 其他：IRFLT 不输出；
4:3	UT0_IOS	rw	0	UART0 模块 IO 设置，详见 IO spec 文档 0, 选择组别 A 1, 选择组别 B 2, 选择组别 C 3, 选择组别 D
2	SPI0_IOS	rw	0	SPI0 IO 选择（见 IO spec） 0： 选择 A 组； 1： 选择 B 组；
1	SD0_DT_EN	rw	1	允许 SD0 CMD/DAT 占用 IO 0:无论 SD0 有无使能，其 CMD/DAT 不占用相应 IO； 1:当 SD0 使能时，其 CMD/DAT 占用相应 IO
0	SD0_CK_EN	rw	1	允许 SD0 CLK 占用 IO 0:无论 SD0 有无使能，其 CLK 不占用相应 IO； 1:当 SD0 使能时，其 CLK 占用相应 IO

2.**IOMAP_CON1**: *IO mapping control register1*

Bit	Name	RW	Default	Description
31:24	预留			
23	ALNK0_IOS	rw	0	ALNK0 IO 设置，详见 IO spec 文档 0, 选择组别 A 1, 选择组别 B
22:21	SDTAP_IOS	rw	0	SDTAP 模块 IO 设置，详见 IO spec 文档 0, 选择组别 A

<table>
<tr><td></td><td></td><td></td><td></td><td>1，选择组别 B
2，选择组别 C
3，选择组别 D</td></tr>
<tr><td>20</td><td>PLNK_SCKOE</td><td>rw</td><td>0</td><td>PLNK_SCK 从 IO 输出使能</td></tr>
<tr><td>19:18</td><td>IIC_IOS</td><td>rw</td><td>-</td><td>IIC 模块 IO 选择设置，详见 IO spec 文档
0，选择组别 A
1，选择组别 B
2，选择组别 C
3，选择组别 D</td></tr>
<tr><td>17</td><td>-</td><td>-</td><td>0</td><td>预留</td></tr>
<tr><td>16</td><td>SPI2_IOS</td><td>rw</td><td>0</td><td>SPI2 IO 设置，详见 IO spec 文档
0，选择组别 A
1，选择组别 B</td></tr>
<tr><td>15:14</td><td>UT2_IOS</td><td>rw</td><td>0</td><td>UART2 模块 IO 设置，详见 IO spec 文档
0，选择组别 A
1，选择组别 B
2，选择组别 C
3，选择组别 D</td></tr>
<tr><td>13</td><td>RDEC0_SIN1_IOS</td><td>rw</td><td>0</td><td>rdec0_sin1 输入选择
0:input channel 7；
1:对应 IO spec 描述的 IO；</td></tr>
<tr><td>12</td><td>RDEC0_SIN0_IOS</td><td>rw</td><td>0</td><td>rdec0_sin0 输入选择
0:input channel 6；
1:对应 IO spec 描述的 IO；</td></tr>
<tr><td>11:8</td><td>OUTPUT_CH0_SEL</td><td>rw</td><td>0</td><td>选择输出到 IO 的信号
<table>
<tr><td>0</td><td>UT0_TX</td></tr>
<tr><td>1</td><td>UT1_TX</td></tr>
<tr><td>2</td><td>TMR0_PWM_OUT</td></tr>
<tr><td>3</td><td>TMR1_PWM_OUT</td></tr>
<tr><td>4</td><td>RTC_OSCL</td></tr>
<tr><td>5</td><td>BTOSC_CLK</td></tr>
<tr><td>6</td><td>PLL_12M</td></tr>
<tr><td>7</td><td>UT2_TX</td></tr>
<tr><td>8</td><td>-</td></tr>
<tr><td>9</td><td>-</td></tr>
<tr><td>10</td><td>-</td></tr>
<tr><td>11</td><td>-</td></tr>
<tr><td>12</td><td>TMR4_PWM_OUT</td></tr>
<tr><td>13</td><td>TMR5_PWM_OUT</td></tr>
<tr><td>14</td><td>WLC_INT_FREQ</td></tr>
<tr><td>15</td><td>TMR3_PWM_OUT</td></tr>
</table></td></tr>
<tr><td>7</td><td>UT1_CTS_IOS</td><td>rw</td><td>0</td><td>UART1 CTS 选择</td></tr>
</table>

				0:input channel 5; 1:对应 IO spec 描述的 IO;
6	CAP_ES	rw	0	input channel2 边沿选择 0:上升沿; 1:下降沿;
5	SFC_IOS	rw	0	SFC 模块 IO 设置，详见 IO spec 文档 0, 选择组别 A 1, 选择组别 B
4	SPI1_IOS	rw	0	SPI1 模块 IO 设置，详见 IO spec 文档 0, 选择组别 A 1, 选择组别 B
3:2	UT1_IOS	rw	0	UART1 模块 IO 设置，详见 IO spec 文档 0, 选择组别 A 1, 选择组别 B 2, 选择组别 C 3, 选择组别 D
1	SPI0_DIDO_MIX	rw	0	spi0 输入结果产生选项 0:spi0 内部输入为 di 1:spi0 内部输入为 di & do
0	--		0	预留

*3.***IOMAP_CON2**: *IO mapping control register2*

Bit	Name	RW	Default	Description
31:30	-	-	0	预留
29:24	INPUT CHANNEL 3	rw	0	wkup 输入选择 0-15 :选择 PA0-PA15 16-31:选择 PB0-PB15 32-47:选择 PC0-PC15 48-55:选择 PD0-PD7 56-58:无 59:选择 TMR0_PWM_OUT 60:选择 TMR1_PWM_OUT 61:选择 USBDP 62:选择 USBDM 63:无
23:22	-	rw	0	预留
21:16	INPUT CHANNEL 2	rw	0	IRFLT 功能脚选择，同 INPUT CHANNEL3
15	RDEC2_SIN1_IOS	rw	0	rdec2_sin1 输入选择 0:input channel 7 1:对应 IO spec 描述的 IO;
14	RDEC2_SIN0_IOS	rw	0	rdec2_sin0 输入选择

				0:input channel 6 1:对应 IO spec 描述的 IO;
13:8	INPUT CHANNEL 1	rw	0	CAPTURE 功能脚选择,同 INPUT CHANNEL3
7	RDEC1_SIN1_IOS	rw	0	rdec1_sin1 输入选择 0:input channel 7 1:对应 IO spec 描述的 IO;
6	RDEC1_SIN0_IOS	rw	0	rdec1_sin0 输入选择 0:input channel 6 1:对应 IO spec 描述的 IO;
5:0	INPUT CHANNEL 0	rw	0	UART_RX 脚选择,同 INPUT CHANNEL3

4.**IOMAP_CON3**: *IO mapping control register3*

Bit	Name	RW	Default	Description
31:28	-	-	0	预留
27:24	OUTPUT_CH2_SEL	rw	0	0 UT1_RTS 1 UT1_TX 2 WLC_INT_ACTIVE 3 TMR1_PWM_OUT 4 PLNK_SCLK 5 BTOSC_CLK 6 PLL_24M 7 UT2_TX 8 - 9 - 10 - 11 - 12 TMR4_PWM_OUT 13 TMR5_PWM_OUT 14 TMR2_PWM_OUT 15 TMR3_PWM_OUT
23:20	OUTPUT_CH1_SEL	rw	0	0 UT0_TX 1 UT1_TX 2 TMR0_PWM_OUT 3 WLC_INT_STATUS 4 RTC_OSL 5 BTOSC_CLK

				6 SPDIF_DO 7 UT2_TX 8 - 9 - 10 - 11 - 12 TMR4_PWM_OUT 13 TMR5_PWM_OUT 14 TMR2_PWM_OUT 15 TMR3_PWM_OUT
19:16	-	-	0	预留
15	PLNK_D1_IOS		1	pdm link data1 select 0 ： 对应 IO spec 描述的 IO； 1 ： input channel 9
14	-	-	0	预留
13	WLC_EXT_IOS	rw	0	wlc_ext_active IO 选择 0 ： PA6 1 ： input channel 10
12				
11	UT2_IOEN	rw	1	允许 UART2 占用 IO 1:占用相应 IO； 0:不占用 IO，该 IO 用于其它功能；
10:8	UT2_MXS	rw	0	UART2 输入选择 0XX:选择普通 IO 作为输入（UT2_IOS）； 100:选择由 WKUP_MUX（input channel 0）选择的 IO 作为输入； 101:选择由 IRFT_MUX（input channel 1）选择的 IO 作为输入； 110:选择由 CAP_MUX（input channel 2）选择的 IO 作为输入； 111:选择由 UART_RX_MUX（input channel 3）选择的 IO 作为输入；
7	UT1_IOEN	rw	1	允许 UART1 占用 IO 1:占用相应 IO； 0:不占用 IO，该 IO 用于其它功能
6:4	UT1_MXS	rw	0	UART1 输入选择 0XX:选择普通 IO 作为输入（UT1_IOS）； 100:选择由 WKUP_MUX（input channel 0）选择的 IO 作为输入； 101:选择由 IRFT_MUX（input channel 1）选择的 IO

				作为输入； 110:选择由 CAP_MUX（input channel 2）选择的 IO 作为输入； 111:选择由 UART_RX_MUX（input channel 3）选择的 IO 作为输入；
3	UT0_IOEN	rw	1	允许 UART0 占用 IO 1:占用相应 IO； 0:不占用 IO，该 IO 用于其它功能；
2:0	UT0_MXS	rw	0	UART0 输入选择 0XX:选择普通 IO 作为输入（UT0_IOS）； 100:选择由 WKUP_MUX（input channel 0）选择的 IO 作为输入； 101:选择由 IRFT_MUX（input channel 1）选择的 IO 作为输入； 110:选择由 CAP_MUX（input channel 2）选择的 IO 作为输入； 111:选择由 UART_RX_MUX（input channel 3）选择的 IO 作为输入

5.**IOMAP_CON4**: *IO mapping control register4*

Bit	Name	RW	Default	Description
31:30	-	-	0	预留
29:24	INPUT CHANNEL 7		0	RDEC_DI[1] 输入选择 0-15 :选择 PA0-PA15 16-31:选择 PB0-PB15 32-47:选择 PC0-PC15 48-55:选择 PD0-PD7 56-58:无 59:选择 TMR0_PWM_OUT 60:选择 TMR1_PWM_OUT 61:选择 USBDP 62:选择 USBDM 63:无
23:22	-	-	0	预留
21:16	INPUT CHANNEL 6		0	RDEC_DI[0]选择，同 INPUT CHANNEL 7
15:14	-	-	0	预留
13:8	INPUT CHANNEL 5		0	UART1_CTS 选择,同 INPUT CHANNEL 7
7:6	-	-	0	预留
5:0	INPUT CHANNEL 4		0	TIMER CLOCK 选择,同 INPUT CHANNEL 7

*6.***IOMAP_CON5**: *IO mapping control register5*

Bit	Name	RW	Default	Description
31:30	-	-	0	预留
29:24	INPUT CHANNEL 11	rw	0	SPDIF_DI_A_P 输入选择 0-15 :选择 PA0-PA15 16-31:选择 PB0-PB15 32-47:选择 PC0-PC15 48-55:选择 PD0-PD7 56-58:无 59:选择 TMR0_PWM_OUT 60:选择 TMR1_PWM_OUT 61:选择 USBDP 62:选择 USBDM 63:无
23:22	-	-	0	预留
21:16	INPUT CHANNEL 10	rw	0	WLC_EXT_ACTIVE_P 的输入选择,同 INPUT CHANNEL 11
15:14	-	-	0	预留
13:8	INPUT CHANNEL 9	rw	0	PDLINK_IN1 的输入选择,同 INPUT CHANNEL 11
7:6	-	-	0	预留
5:0	INPUT CHANNEL 8	rw	0	PDLINK_IN0 的输入选择,同 INPUT CHANNEL 11

第 4 章. 32 位定时器（TIMER32）

4.1. 概述

Timer32 是一个集合了定时/计数/捕获功能于一体的多动能 32 位定时器。它的驱动源可以选择片内时钟或片外信号。它带有一个可配置的最高达 64 的异步预分频器，用于扩展定时时间或片外信号的最高频率。它还具有上升沿/下降沿捕获功能，可以方便的对片外信号的高电平/低电平宽度进行测量。

4.2. 寄存器说明

1.Tx_CON： timer x control register

Bit	Name	RW	Description
31-16	Reserved		
15	PND	R	中断请求标志，当 timer 溢出或产生捕获动作时会被硬件置 1，需要由软件清 0
14	PCLR	W	软件在此位写入‘1’将清除 PND 中断请求标志
	Reserved		
9	PWM_ INV	RW	PWM 信号输出反向
8	PWM_EN	RW	PWM 信号输出使能。此位置 1 后，相应 IO 口的功能将会被 PWM 信号输出替代。
7-4	PSET	RW	预分频选择位 0000：预分频 1 0001：预分频 4 0010：预分频 16 0011：预分频 64 0100：预分频 1*2 0101：预分频 4*2 0110：预分频 16*2 0111：预分频 64*2 1000：预分频 1*256 1001：预分频 4*256 1010：预分频 16*256

			1011：预分频 64*256 1100：预分频 1*2*256 1101：预分频 4*2*256 1110：预分频 16*2*256 1111：预分频 64*2*256
3-2	SSEL	RW	timer 驱动源选择 00：使用系统时钟作为 timer 的驱动源； 01：使用 IO 口信号作为 timer 的驱动源； 10：使用 OSC 时钟作为 timer 的驱动源； 11：使用 RC 时钟作为 timer 的驱动源。
1-0	MODE	RW	工作模式选择 00： timer 关闭； 01： 定时/计数模式； 10： IO 口上升沿捕获模式（当 IO 上升沿到来时，把 TxCNT 的值捕捉到 TxPR 中）； 11： IO 口下降沿捕获模式（当 IO 下降沿到来时，把 TxCNT 的值捕捉到 TxPR 中）。

2.Tx_CNT：timer x counter register

Bit	Name	RW	Description
31-0	Tx_CNT	RW	Timer32 的计数寄存器

3.Tx_PR： timer x period register

Bit	Name	RW	Description
31-0	Tx_PR	RW	Timer32 的周期寄存器 在定时/计数模式下，当 Tx_CNT == Tx_PR 时，Tx_CNT 会被清 0。 在上升沿/下降沿捕获模式下，TxPR 是作为捕获寄存器使用的，当捕获发生时，Tx_CNT 的值会被复制到 Tx_PR 中。

			而此时 Tx_CNT 自由的由 0-4294967295-0 计数，不会和 Tx_PR 进行比较清 0。

4.Tx_PWM： timer x PWM register

Bit	Name	RW	Description
31-0	Tx_PWM	RW	Timer32 的 PWM 设置寄存器 在 PWM 模式下，此寄存器的值决定 PWM 输出的占空比。占空比 N 的计算公式如下： N = (Tx_PWM / Tx_PR) * 100% 此寄存器不带有缓冲，写此寄存器的动作将可能导致不同步状态产生的 PWM 波形占空比瞬间过大或过小的问题

第 5 章. 集成电路总线（IIC）

5.1. 概述

IIC 接口是一个标准的遵守 IIC 协议的串行通讯接口。在上面传输的数据以 Byte（8bit）为最小单位，且永远是 MSB 在前。

IIC 接口支持主机和从机两种模式

主机：1）IIC 接口时钟由本机产生，提供给片外 IIC 设备使用；

2）IIC 接口的驱动时钟可配置，频率范围为 F = Fsystem/（（1+IIC_BAUD）*2） + 电阻上拉的时间。上拉电阻越大，频率越低。

从机：1）IIC 接口时钟由片外 IIC 设备产生，提供给本机使用；

2）IIC 总线上每一个 start/restart 位后接从机地址，此时当外部 IIC 设备所发的地址与我们匹配时，硬件会自动回应（ack 位上为“0”）；

3）发送时，当前这包的数据传输 ACK 后，不支持接下来直接 restart 或者 end；

4）接收时，当前这包的数据传输 ACK 后，不支持接下来直接 restart；

IIC 接口使用下降沿更新数据，上升沿采集数据。

IIC 接口的发送寄存器和接收寄存器在物理上是分开的，但在逻辑上它们一起称为 IIC_BUF 寄存器，使用相同的 SFR 地址。当写这个 SFR 地址时，写入至发送寄存器。当读这个 SFR 地址时，从接收寄存器读出。

5.2. 寄存器说明

1.IIC_CON0: IIC control register0 (16bit addressing)

Bit	Name	RW	Description
15	PND	ro	PND：中断请求标志，当完成 1Byte 加 1 应答位的传输后会被硬件置“1”；清除此标志用软件方式：向 PCLR 写“1”。
14	PCLR	wo	软件在此位写入‘1’将清除 PND 中断请求标志。
13	END_PND	ro	结束位标志。
12	END_PND_CLR	wo	软件在此位写入‘1’将清除 END_PND 标志
11	reserved	rw	预留
10	END_PND_IE	rw	结束位中断使能
9	IIC_ISEL	rw	iic_cki 和 iic_di 输入选择： 0：选择 IO 直接输入

			1：选择 IO 经过 filter 后再输入
8	IE	rw	IIC 中断使能。 0：禁止 SPI 中断 1：允许中断允许
7	RD_ACK	ro	IIC 中的应答位标志。 0：应答 1：不应答
6	WR_ACK	rw	当作为接收方（主机或者从机模式）对应答位的设置。 0：应答 1：不应答 当作为发送方时，硬件自动置“1”释放总线。
5	M_SET_RSTART	wo	主机重新开始设置位，硬件自动清“0”。 每次传输的第一包数据，只要总线上是上拉的，硬件自动加 start 位，过后的每包数据传输接不接 start 位可配 0：下一包数据传输不接重新开始位 1：下一包数据传输紧接重新开始位
4	M_SET_END	wo	主机结束设置位，硬件自动清“0”。 0：当前这一包数据传输后不接结束位 1：当前这一包数据传输后紧接结束位
3	DAT_DIR	rw	IIC 接口传输方向。 0：发送数据 1：接收数据
2	N_CFG_DONE	wo	每次传输前，传输配置完成标志。（configuration done） 每次传输前，需要对 IIC 接口进行所需的配置（DAT_DIR、M_SET_END、IIC_BUF、WR_ACK 等）之后再将此 bit 置“1”，硬件自动清“0”。 0：配置中 1：配置完成 配置完毕过后才可将此 bit 置“1”，建议配置完所需设置后延时一段时间才将其置“1”。
1	SLAVE	rw	IIC 接口模式选择。 0：主机模式 1：从机模式
0	EN	rw	IIC 接口使能。

			0：关闭 IIC 接口 1：打开 IIC 接口

2.IIC_CON1: IIC control register1 (16bit addressing)

Bit	Name	RW	Description
15	SPND	ro	SPND：起始位标志
14	SPND_CLR	wo	SPND_CLR：起始位清除，写 1 清除
13	SI_MODE	rw	SI_MODE：从机忽略总线活动模式：（1 有效）当 IIC 处于从机模式，访问我们的设备地址与我们设置的地址（IIC_BAUD 高 7 位）不匹配，则硬件不作任何响应，直到访问我们的设备地址匹配，硬件才运作。
12-0	reserved	rw	预留

3.IIC_BAUD:(16bit addressing，write only)

Bit	Name	RW	Description
15-0	IIC_BAUD	rw	

1）IIC 接口为主机时，IIC_BAUD 做为时钟设置寄存器。

2)频率范围为 F = Fsystem/（（1+IIC_BAUD）*2） + 电阻上拉的时间。上拉电阻越大，频率越低。

3）IIC 接口为从机时，IIC_BAUD 做为从机地址(7bits)设置寄存器。

从机地址 = IIC_BAUD[7：1]

当地址匹配时硬件自动应答（ACK 位为“0”）

4.IIC_BUF: (8bit addressing，write and read)

Bit	Name	RW	Description
7-0	IIC_BUF	rw	

发送寄存器和接收寄存器共用此 SFR 地址。写入至发送寄存器，从接收寄存器读出。

第 6 章. 串行通信（UART）

6.1. 概述

UART0、 UART1 支持接收带循环 Buffer 的 DMA 模式和普通模式。

UART2 只支持普通模式，不支持 DMA 模式。

UART0 在 DMA 接收的时候有一个循环 Buffer，UTx_RXSADR 表示它的起始，UTx_RXEADR 表示它的结束。同时，在接收过程中，会有一个超时计数器（UTx_OTCNT），如果在指定的时间里没有收到任何数据，则超时中断就会产生。超时计数器是在收到数据的同时自动清空。

6.2. 控制寄存器

寄存器列表	UART0	UART1	UART2
UTx_CON(0)	UT0_CON0	UT1_CON	UT2_CON
UTx_CON1	UT0_CON1	UT1_CON1	
UTx_BAUD	UT0_BAUD	UT1_BAUD	UT2_BAUD
UTx_BUF	UT0_BUF	UT1_BUF	UT2_BUF
UTx_TXADR	UT0_TXADR	UT1_TXADR	
UTx_TXCNT	UT0_TXCNT	UT1_TXCNT	
UTx_RXCNT	UT0_RXCNT	UT1_RXCNT	
UTx_RXSADR	UT0_RXSADR	UT1_RXSADR	
UTx_RXEADR	UT0_RXEADR	UT1_RXEADR	
UTx_HRXCNT	UT0_HRXCNT	UT1_HRXCNT	
UTx_OTCNT	UT0_OTCNT	UT1_OTCNT	

6.3. 寄存器说明

1.UTx_CON(0): uart x control register (16bit addressing).

Bit	Name	RW	Description
15	TPND	r	TPND:TX Pending
14	RPND	r	RPND:RX Pending& Dma_Wr_Buf_Empty，数据接收不完 Pending 不会为 1
13	CLRTPND	r	CLRTPND:清空 TX Pending
12	CLRRPND	r	CLRRPND:清空 RX Pending
11	OTPND	r	(*) OTPND:OverTime Pending

10	CLR_OTPND	r	（*）CLR_OTPND：清空 OTPND
9	TB8	rw	TB8:9Bit 模式时，TX 发送的第 9 位
8	RB8	r	RB8:9Bit 模式时。RX 接收到的第 9 位
7	RDC	w	RDC（*）:写 1 时，将已经收到的数目写到 UTx_HRXCNT，已收到的数目清零写 0 无效
6	RX_MODE	rw	RXMODE（*）:读模式选择 0:普通模式，不用 DMA； 1:DMA 模式。
5	OT_IE	rw	OTIE（*）:OT 中断允许 0:不允许； 1:允许。
4	DIVS	rw	DIVS:前 3 分频选择，0 为 4 分频，1 为 3 分频
3	RXIE	rw	RXIE:RX 中断允许 当 RX Pending 为 1，而且 RX 中断允许为 1，则会产生中断
2	TXIE	rw	TXIE:TX 中断允许 当 TX Pending 为 1，而且 TX 中断允许为 1，则会产生中断
1	M9EN	rw	M9EN:9bit 模式使能
0	UTEN	rw	UTEN:UART 模块使能

（*）:只有 UART0/1 支持

2.UTx_CON1: uart x control register1 (16bit addressing).

Bit	Name	RW	Description
15	CTSPND	rw	CTSPND:CTS 中断 pending
14	CLR_CTSPND	wo	CLR_CTSPND:清楚 CTS pending
13	CLRRTS	wo	CLRRTS:清除 RTS 0:N/A 1:清空 RTS
12-5	Reserved	rw	预留
4	RX_DISABLE	rw	关闭数据接收 0:开启输入（正常模式） 1:关闭输入（输入固定为 1）
3	CTSIE	rw	CTSIE:CTS 中断使能 0:禁止中断； 1:中断允许 ***注：只有 UART1 有该功能***
2	CTSE	rw	CTSE:CTS 使能

			0:禁止 CTS 硬件流控制 1:允许 CTS 硬件流控制 ***注：只有 UART1 有该功能***
1	RTS_DMAEN	rw	RTS_DMAEN:RTS 接收数据流控制使能 0:禁止 1:允许 ***注：只有 UART1 有该功能***
0	RTSE	rw	RTSE:RTS 使能 0:禁止 RTS 硬件流控制 1:允许 RTS 硬件流控制 ***注：只有 UART1 有该功能***

3.UTx_BAUD: uart x baudrate register (16bit addressing, Write Only).

Bit	Name	RW	Description
15-0	UTx_BAUD	wo	

uart 的 UTx_DIV 的整数部分

串口频率分频器因子(UTx_DIV)的整数部分

当 DIVS=0 时，

Baudrate = Freq_sys / ((UTx_BAUD+1) *4 + BAUD_FRAC)

当 DIVS=1 时，

Baudrate = Freq_sys / ((UTx_BAUD+1) *3 + BAUD_FRAC)

(Freq_sys 是 apb_clk,指慢速设备总线的时钟，非系统时钟)

4.UTx_BUF: uart x data buffer register (8bit addressing).

Bit	Name	RW	Description
15-0	UT0BUF	wo	uart 的收发数据寄存器 写 UTx_BUF 可启动一次发送； 读 UTx_BUF 可获得已接收到的数据。

5.UTx_TXADR:uart x TX DMA address(25bit addressing, Write Only)

Bit	Name	RW	Description
24-0	UTx_TXADR	wo	DMA 发送数据的起始地址

6.UTx_TXCNT:uart x TX DMA count (32bit addressing, Write Only).

Bit	Name	RW	Description
31-0	UTx_TXCNT	wo	写 UTx_TXCNT，控制器产生一次 DMA 的操作，同时清空中断，当 uart 需要发送的数据达到 UTx_TXCNT 的值，控制器会停止发送数据的操作，同时产生中断（UTx_CON[15]）。

7.UTx_RXCNT:uart x receive DMA count(32bit addressing, Write Only).

Bit	Name	RW	Description
31-0	UTx_RXCNT	wo	写 UTx_RXCNT，控制器产生一次 DMA 的操作，同时清空中断，当 uart 需要接收的数据达到 UTx_RXCNT 的值，控制器会停止接收数据的操作，同时产生中断（UTx_CON[14]）。

8.UTx_RXSADR:uart x receive DMA address(25bit addressing, Write Only)

Bit	Name	RW	Description
24-0	UTx_RXSADR	wo	DMA 接收数据时，循环 buffer 的起始地址

9.UTx_RXEADR:uart x receive DMA end address(25bit addressing, Write Only).

Bit	Name	RW	Description
24-0	UTx_RXEADR	wo	DMA 接收数据时，循环 buffer 的结束地址

10.UTx_HRXCNT:uart x have receive DMA count(32bit addressing, Read Only).

Bit	Name	RW	Description
31-0	UTx_HRXCNT	ro	当设这 RDC（UTx_CON[7]）=1 时，串口设备会将当前总共收到的字节数记录到 UTx_HRXCNT 里。

11.UTx_OTCNT:uart x OverTime count(32bit addressing, Write Only).

Bit	Name	RW	Description
31-0	UTx_OTCNT	wo	

设置串口设备在等待多久 RX 下降沿的时间，如果在所设置的时间里没收到 RX 的下降沿，则产生 OT 中断（OT_PND）

Time(ot)= Time(uart_clk) * UTx_OTCNT;

例如：波特率时间为 100ns，UTx_OTCNT = 10,那么，OT 的时间就为 1000ns

第 7 章. 串行外设接口（SPI）

7.1. 特征

- 传输数据以 Byte（8bit）为最小单位；
- 支持主机模式和从机模式操作。
- 单项传输支持 1bit data、2bit data 和 4bit data 模式，双向传输只支持 1bit data 模式
- 支持 CPU 方式操作和 DMA 方式操作。

7.2. 概述

SPI 接口是一个标准的遵守 SPI 协议的串行通讯接口，在上面传输的数据以 Byte（8bit）为最小单位，且永远是 MSB 在前。SPI 接口可独立地选择在 SPI 时钟的上升沿或下降沿更新数据，在 SPI 时钟的上升沿或下降沿采样数据。

（1）主/从机模式：

SPI 接口支持主机和从机两种模式：

主机：SPI 接口时钟由本机产生，提供给片外 SPI 设备使用；在主机模式下，SPI 接口的驱动时钟可配置，范围为 系统时钟~系统时钟/256

从机：SPI 接口时钟由片外 SPI 设备产生，提供给本机使用；在从机模式下，SPI 接口的驱动时钟频率无特殊要求，但数据速率需要进行限制，否则易出现接收缓冲覆盖错误。

（2）传输模式：

SPI 接口支持单向（Unidirection）和双向（Bidirection）模式

单向模式：使用 SPICK 和 SPIDAT 两组连线，其中 SPIDAT 为双向信号线，同一时刻数据只能单方向传输。

双向模式：使用 SPICK，SPIDI 和 SPIDO 三组连线，同一时刻数据双向传输。但 DMA 不支持双向数据传输，当在本模式下使能 DMA 时，也只有一个方向的数据能通过 DMA和 系统进行传输。

（3）数据模式：

SPI 单向模式支持 1bit data、2bit data 和 4bit data 模式，即：

1bit data 模式：串行数据通过一根 DAT 线传输，一个字节数据需 8 个 SPI 时钟。

2bit data 模式：串行数据通过两根 DAT 线传输，一个字节数据需 4 个 SPI 时钟。

4bit data 模式：串行数据通过四根 DAT 线传输，一个字节数据需 2 个 SPI 时钟。

（注：SPI0 支持 1bit，2bit 和 4bit 模式，SPI1/SPI2 只支持 1bit 和 2bit 模式）

SPI 双向模式只支持 1bit data 模式，即：

1bit data 模式：串行数据通过一根 DAT 线传输，一个字节数据需 8 个 SPI 时钟。

（4）数据发送与接收：

SPI 接口在发送方向上为单缓冲，在上一次传输未完成之前，不可开始下一次传输。在接收方向上为双缓冲，如果在下一次传输完成时 CPU 还未取走本次的接收数据，那么本次的接收数据将会丢失。

SPI 接口的发送寄存器和接收寄存器在物理上是分开的，但在逻辑上它们一起称 SPIBUF 寄存器，使用相同的 SFR 地址。当写这个 SFR 地址时，写入至发送寄存器。当读这个 SFR 地址时，从接收寄存器读出。

（5）操作模式：

SPI 传输支持由 CPU 直接驱动，写 SPIBUF 的动作将启动一次 Byte 传输。

SPI 传输也支持 DMA 操作，但 DMA 操作永远是单方向的，即一次 DMA 要么是发送一包数据，要么是接收一包数据，不能同时发送并且接收一包数据，即使在双向模式下也是这样。每次 DMA 操作支持的数据量为 1-65535Byte。写 SPIADR 的动作将启动一次 DMA 传输。

7.3. 传输波形

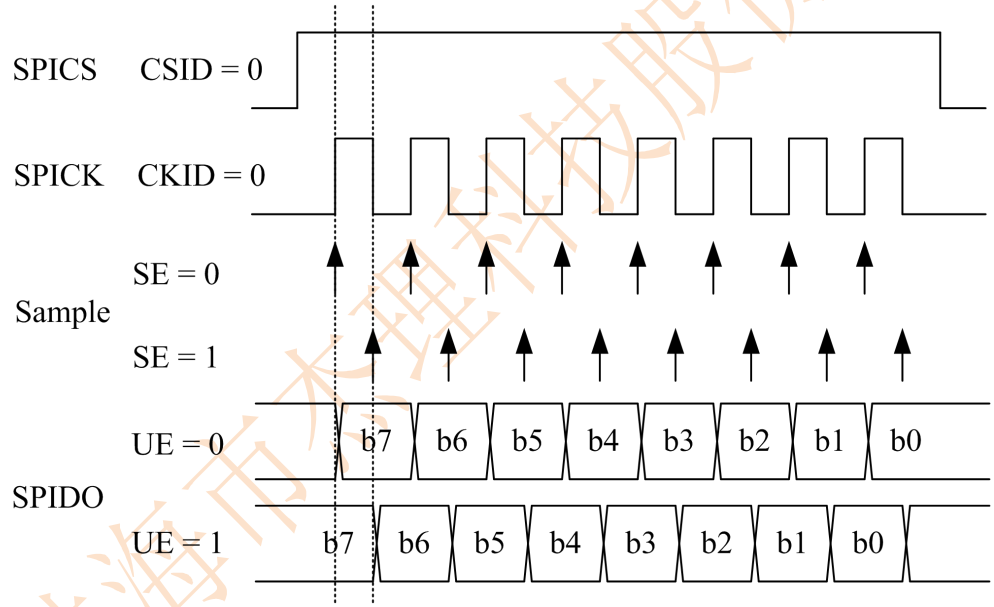


图 7-1. SPICS 空闲时为低电平，SPICLK 空闲时为低电平的波形图

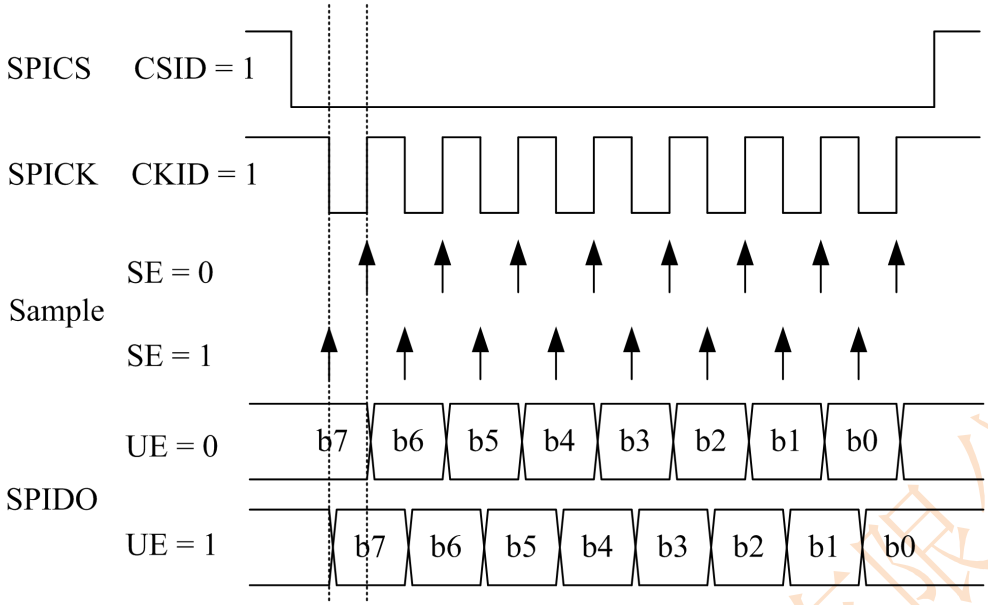


图 7-2. SPI CS 空闲时为高电平，SPI CLK 空闲时为高电平的波形图

7.4. 寄存器说明

寄存器列表	SPI0	SPI1	SPI2
SPIx_CON	SPI0_CON	SPI1_CON	SPI2_CON
SPIx_BUF	SPI0_BUF	SPI1_BUF	SPI2_BUF
SPIx_BAUD	SPI0_BAUD	SPI1_BAUD	SPI2_BAUD
SPIx_ADR	SPI0_ADR	SPI1_ADR	SPI2_ADR
SPIx_CNT	SPI0_CNT	SPI1_CNT	SPI2_CNT

1.SPIx_CON: SPIx control register (16bit addressing)

Bit	Name	RW	Description
15	PND	r	中断请求标志，当 1Byte 传输完成或 DMA 传输完成时会被硬件置 1。 有 3 种方法清除此标志 1. 向 PCLR 写入‘1’ 2. 写 SPIBUF 寄存器来启动一次传输 3. 写 SPICNT 寄存器来启动一次 DMA
14	PCLR	w	软件在此位写入‘1’将清除 PND 中断请求标志。
13	IE	rw	SPI 中断使能 0：禁止 SPI 中断 1：允许中断允许

12	DIR	rw	在单向模式或 DMA 操作时设置传输的方向 0：发送数据 1：接收数据
11-10	DATW	r	SPI 数据宽度设置 00：1bit 数据宽度 01：2bit 数据宽度 10：4bit 数据宽度 11:NA，不可设置为此项 *（注：SPI0 支持 1bit，2bit 和 4bit 模式，SPI1/SPI2 只支持 1bit 模式）*
9	reserved	r	0
8	reserved	r	0
7	CSID	rw	SPICS 信号极性选择 0：SPICS 空闲时为 0 电平 1：SPICS 空闲时为 1 电平
6	CKID	rw	SPICK 信号极性选择 0：SPICK 空闲时为 0 电平 1：SPICK 空闲时为 1 电平
5	UE	rw	更新数据边沿选择 0：在 SPICK 的上升沿更新数据 1：在 SPICK 的下降沿更新数据
4	SE	rw	采样数据边沿选择 0：在 SPICK 的上升沿采样数据 1：在 SPICK 的下降沿采样数据
3	BIDIR	rw	单向/双向模式选择 0：单向模式，数据单向传输，同一时刻只能发送或者接收数据。 数据传输方向因收发而改变，所以由硬件控制，不受写 IO 口 DIR 影响。 1：双向模式，数据双向传输，同时收发数据，但 DMA 只支持一个方向的数据传输。 数据传输方向设置后不改变，所以由软件控制，通过写 IO 口 DIR 控制。
2	CSE	rw	SPICS 信号使能 0：不使用 SPICS 信号 1：使用 SPICS 信号
1	SLAVE	rw	从机模式
0	SPIE	rw	SPI 接口使能 0：关闭 SPI 接口 1：打开 SPI 接口

2.SPIx_BAUD: SPI baudrate setting register (8bit addressing, write only).

Bit	Name	RW	Description
7-0	SPI_BAUD	rw	

SPI 主机时钟设置寄存器

SPICK = system clock / (SPIBAUD + 1)

3.SPI_BUF: SPI buffer register (8bit addressing)

Bit	Name	RW	Description
7-0	SPI_BUF	rw	

发送寄存器和接收寄存器共用此 SFR 地址。写入至发送寄存器，从接收寄存器读出。

4.SPI_ADR: SPI DMA start address register (25bit addressing, write only).

Bit	Name	RW	Description
24-0	SPI_ADR	rw	

SPI DMA 起始地址寄存器，只写，读出为不确定值

5.SPI_CNT: SPI DMA counter register (16bit addressing, write only)

Bit	Name	RW	Description
15-0	SPI_CNT	rw	

SPI DMA 计数寄存器，只写，读出为不确定值

此寄存器用于设置 DMA 操作的数目（按 Byte 计）并启动 DMA 传输。

如，需启动一次 512Byte 的 DMA 传输，写入 0x0200，此写入动作将启动本次传输。

第 8 章. PAP

8.1. 特征

- 工作于主机模式。
- 支持 8bit 的数据宽带
- 支持 CPU 以及 DMA 操作模式。

8.2. 概述

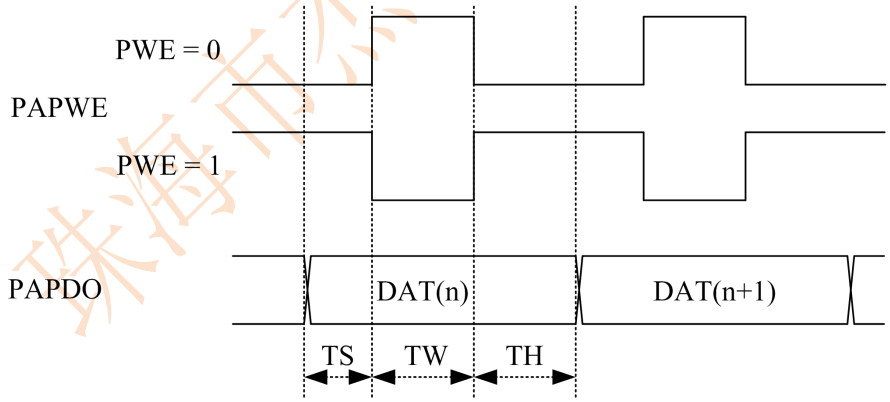
PAP 接口（Parallel Active Port）是一个并行主动数据接口，它工作于主动模式，支持 8bit 数据宽度，具有读/写使能信号，并可支持 DMA 方式发送/接收数据。

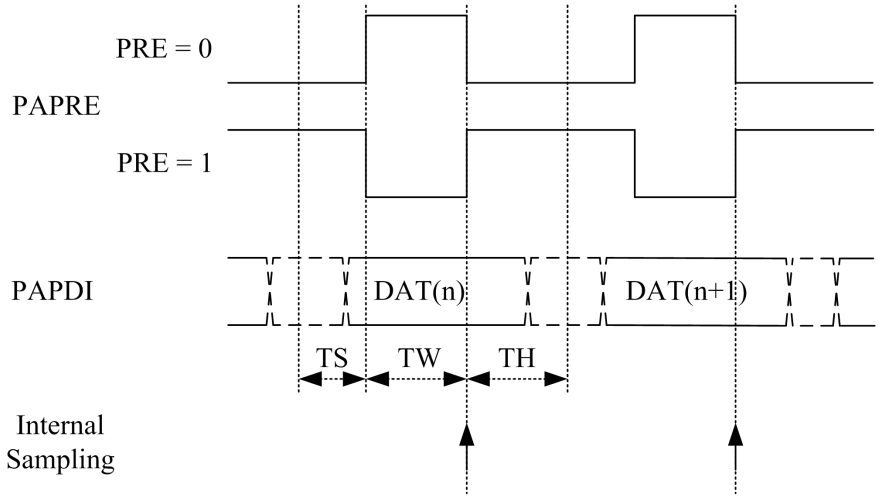
PAP 传输支持由 CPU 直接驱动，写 PAPBUF 的动作将启动一次传输；PAP 传输也支持 DMA 操作，每次 DMA 操作支持的数据量为 1-65535Byte，写 PAPCNT 的动作将启动一次传输。

PAP 在 DMA 发送数据（写）时支持普通模式和数据扩展模式。普通模式是指：PAP 接口直接将原始数据发送出去。数据扩展模式是指，PAP 接口可预设两个 16 位的数值 DAT0 和 DAT1，在发送过程中，硬件依次检查原始数据（1Byte）的每一 bit（从 LSB 到 MSB，或从 MSB 到 LSB），若该 bit 为 1，则发送预设值 DAT1，否则发送预设值 DAT0。此状态下， DAT0/DAT1 先后分两次写出。

注：PAP 在接收数据（读）时只支持普通模式，请勿在读状态下使能数据扩展模式，否则会导致不可预计的错误。

8.3. 传输波形





8.4. 寄存器说明

1.PAPCON: PAP contrl register 0(24bit addressing)

Bit	Name	RW	Description
23-19	reserved	r	预留
18	PAPIE	rw	PAPIE:中断使能 0：关闭，不允许 PAP 引发 CPU 中断 1：打开，允许 PAP 引发 CPU 中断
17	EXTMSB	rw	EXTMSB:数据扩展模式下，数据扩展顺序设置 0：从 LSB 到 MSB 逐位检查原始数据 1：从 MSB 到 LSB 逐位检查原始数据
16	EXTE	rw	EXTE:数据扩展模式使能 0：普通模式 1：数据扩展模式，此模式只支持数据发送（写），勿在数据接收（读）时设置此位
15-14	TS[1:0]	rw	TS1-0:数据建立时间设置 0x0：数据建立时间为 0 0x1：数据建立时间为 1 个系统时钟的宽度 0x2：数据建立时间为 2 个系统时钟的宽度 0x3：数据建立时间为 3 个系统时钟的宽度
13-12	TH[1:0]	rw	TH1-0:数据保持时间设置 0x0：数据保持时间不小于 0 0x1：数据保持时间不小于 1 个系统时钟的宽度 0x2：数据保持时间不小于 2 个系统时钟的宽度 0x3：数据保持时间不小于 3 个系统时钟的宽度
11-8	TW[3:0]	rw	TW3-0：读/写使能信号宽度设置 0x0：读/写使能信号宽度为 16 个系统时钟的宽度 0x1：读/写使能信号宽度为 1 个系统时钟的宽度

			0x2：读/写使能信号宽度为 2 个系统时钟的宽度 0x3：读/写使能信号宽度为 3 个系统时钟的宽度 0x4：读/写使能信号宽度为 4 个系统时钟的宽度 0x5：读/写使能信号宽度为 5 个系统时钟的宽度 0x6：读/写使能信号宽度为 6 个系统时钟的宽度 0x7：读/写使能信号宽度为 7 个系统时钟的宽度 0x8：读/写使能信号宽度为 8 个系统时钟的宽度 0x9：读/写使能信号宽度为 9 个系统时钟的宽度 0xA：读/写使能信号宽度为 10 个系统时钟的宽度 0xB：读/写使能信号宽度为 11 个系统时钟的宽度 0xC：读/写使能信号宽度为 12 个系统时钟的宽度 0xD：读/写使能信号宽度为 13 个系统时钟的宽度 0xE：读/写使能信号宽度为 14 个系统时钟的宽度 0xF：读/写使能信号宽度为 15 个系统时钟的宽度
7	PND	r	PND:中断请求标志，当一次传输完成或 dma 传输完成时会被硬件置 1。 有 3 种方法清除此标志 1.向 PCLR 写入‘1’ 2.写 PAPBUFL 寄存器来启动下一次传输 3.写 PAPCNT 寄存器来启动下一次 dma
6	PCLR	w	PCLR:软件在此位写入‘1’将清除 PND 中断请求标志，写入‘0’无效
5	DW16ED	rw	DWED：数据大小端选择 0:数据至端口低位 1:数据至端口高位
4	DW16EN	rw	DW16EN：8bit/16bit 模式选择 0：8bit 模式 1：N/A
3	PRE	rw	PRE:读使能信号极性选择 0：读使能信号空闲时为 0 电平，有效时为 1 电平 1：读使能信号空闲时为 1 电平，有效时为 0 电平
2	PWE	rw	PWE:写使能信号极性选择 0：写使能信号空闲时为 0 电平，有效时为 1 电平 1：写使能信号空闲时为 1 电平，有效时为 0 电平
1	DIR	rw	DIR:传输方向设置 0：发送数据 1：接收数据
0	PAPE	rw	PAPE:PAP 接口使能 0：关闭 PAP 接口 1：打开 PAP 接口

2.PAPBUF: PAP buffer register (16bit addressing)

Bit	Name	RW	Description
23-19	PAP_BUF	rw	

发送寄存器和接收寄存器共用此 SFR 地址。写入至发送寄存器，从接收寄存器读出。写此寄存器将启动一次发送或接收操作。

3.PAPDAT0: PAP data register 0 (16bit addressing)

Bit	Name	RW	Description
23-19	PAP_DAT0	rw	

用于数据扩展模式的 DAT0 寄存器，只写，读出为不确定值

4.PAPDAT1: PAP data register 1 (16bit addressing)

Bit	Name	RW	Description
23-19	PAP_DAT1	rw	

用于数据扩展模式的 DAT1 寄存器，只写，读出为不确定值

5.PAPADR: PAP dma start address register (32bit addressing)

Bit	Name	RW	Description
23-19	PAP_ADR	rw	

PAP dma 起始地址寄存器，只写，读出为不确定值

6.PAPCNT: PAP dma counter register (16bit addressing)

Bit	Name	RW	Description
23-19	PAP_CNT	rw	

PAP dma 计数寄存器，只写，读出为不确定值

此寄存器用于设置 dma 操作的数目（按 Byte 计）并启动 dma 传输。 如，需启动一次 512Byte 的 dma 传输，写入 0x0200，此写入动作将启动本次传输。

第 9 章. ADC

9.1. 概述

10Bit ADC(A/D 转换器)，其时钟最大不可超过 1MHz。

9.2. 寄存器说明

1.ADC_CON: ADC control register

Bit	Name	RW	Description
31-16	-	-	预留
15-12	WAIT_TIME	rw	WAIT_TIME： 启动延时控制，实际启动延时为此数值乘 8 个 ADC 时钟
11-8	CH_SEL	rw	CH_SEL：通道选择 0000：选择 PA0 0001：选择 PA1 0010：选择 PA3 0011：选择 PA4 0100：选择 PA6 0101：选择 PB1 0110：选择 PB3 0111：选择 PB4 1000：选择 PB6 1001：选择 PB7 1010：选择 PC2 1011：选择 PC4 1100：选择 PC5 1101：选择 USBDP 1110：选择 USBDM 1111：选择 VDDIO to ADC(PLL,PMU,AUDIO,BT,FM 注意同一时间只能开一个通道)
7	PND	r	PND：只读，中断请求位，当 ADC 完成一次转换后，此位会被设置为‘1’，需由软件清‘0’
6	CPND	w	CPND：只写，写‘1’清除中断请求位，写‘0’无效
5	ADC_IE	rw	ADC_IE：ADC 中断允许
4	ADC_EN	rw	ADC_EN：ADC 控制器使能
3	ADC_AE	rw	ADC_AE：ADC 模拟模块使能
2-0	ADC_BAUD	rw	ADC_BAUD： ADC 时钟频率选择 000：LSB 时钟 1 分频 001：LSB 时钟 6 分频 010：LSB 时钟 12 分频 011：LSB 时钟 24 分频 100：LSB 时钟 48 分频 101：LSB 时钟 72 分频

			110：LSB 时钟 96 分频 111：LSB 时钟 128 分频

2.P3_ANA_CON4 ANA control SFR（PMU to ADC）

Bit	Name	RW	Description
7-5	reserved	rw	预留
4	VGB_SEL	rw	0：main VBG 1: weak VBG
3-1	CHANNEL_ADC_S[2:0]	rw	CHANNEL_ADC_S2-0:select channel to ADC 000(default): VBG 001: VDC13 010: SYSVDD 011: VTEMP 100: PROGF 101: 1/4 VBAT 110: 1/4 LDO5v 111: WVDD
0	PMU_DET_EN	r	PMU_DET_EN:PMU voltage detect to ADC output enable

3.ADA_CON3: ADA control register（AUDIO to ADC）

Bit	Name	RW	Description
30	R_VOUTR_TEST_EN_11v	rw	1: R_VOUTR to SARADC
29	R_VOUTL_TEST_EN_11v	rw	1: R_VOUTL to SARADC
28	DACVDD_TEST_EN_11v	rw	1: DACVDD to SARADC
27	CTADCREF_TEST_11v	rw	1: CTADCREF to SARADC
26	MICLDO_TEST_11v	rw	1: MICLDO to SARADC
25	F_VOUTR_TEST_EN_11v	rw	1: F_VOUTR to SARADC
24	F_VOUTL_TEST_EN_11v	rw	1: F_VOUTL to SARADC

Note:only one TEST channel can set to 1 while others must set to 0.

4.PLL_CON1: pll to adc

Bit	Name	RW	Description
18	PLL_TEST_EN	rw	pll to adc enable
17-16	PLL_TEST_S<1:0>	rw	00bias current 10u（CP 不测）-- 01PLL analog block voltage1.0v~1.2v 10PLL digital block voltage1.0v~1.2v 11Pll output 480meg clock（环路测试实现）

5.WLA_CON25: FM to ADC

BIT	NAME	RW	Description
19	FMTB_ADC_TEST_EN	rw	FM to ADC en
23-21	FMTX_SEL	rw	

6.BT to adc

en: WLA_CON4[6]

第 10 章. USB_bridge

10.1. 概述

负责控制系统与 USB 模块之间的通讯，包括：

1. 接收 CPU 的命令，控制 SIE；
2. 管理 endpoint mapping；
3. 负责 SIE 和 memory 的通讯；
4. 控制 USB PHY。

10.2. 寄存器说明

1.USB_CON0

Bit	Name	RW	Description
31-24	-	-	预留
23	EP4_DISABLE	rw	关闭端点 4，不会返回 ack，nak，stall
22	EP3_DISABLE	rw	关闭端点 3，不会返回 ack，nak，stall
21	EP2_DISABLE	rw	关闭端点 2，不会返回 ack，nak，stall
20	EP1_DISABLE	rw	关闭端点 1，不会返回 ack，nak，stall
19	-	-	预留
18	LOWP_MD_	rw	低有效，默认为使用低功耗模式 0:使用低功耗模式，即系统时钟可跑低于 48m； 1:不使用低功耗模式，即要用 USB 模块时系统得跑 48m 或 48m 以上；
17	SE_DP	r	DP 输入的电平
16	SE_DM	r	DM 输入的电平
15	CHKDPO	r	DP 外接下拉检查结果. 当 PDCHKDP=0，CHKDPO=1
14	SIE_PND	r	SIE 中断请求标志，通过访问 USB 模块清除标志
13	SOF_PND	r	SOF 中断请求标志
12	CLR_SOFP	w	写 1 清除 SOF 中断请求标志，写 0 无效
11	SIEIE	rw	SIE 中断使能
10	SOFIE	rw	SOF 中断使能
9	PDCHKDP	rw	DP 外接下拉检查使能 0:disable 1:enable
8-7	-	-	预留
6	USB_TEST	rw	USB 测试模式
5	VBUS	rw	USB 电源
4	CID	rw	USB 工作模式： 0:host

			1:device
3	TM1	rw	用于缩短检测连接时间(short connect timeout)： 0:disable 1:enable
2	USB_NRST	rw	USB 模块复位： 0:reset 1:release reset
1	LOW_SPEED	rw	低速 USB_DMA 使能： 0:disable 1:enable
0	PHY_ON	rw	USB_PHY 使能: 0:disable 1:enable

2.USB_CON1

Bit	Name	RW	Description
31-16	-	-	预留
15	MC_ACK	r	USB 模块 ACK 信号： 0:busy 1:ready
14	MC_RNW	w	USB 寄存器读写控制： 0:write 1:read
13-8	MC_ADR	w	访问 USB 寄存器地址
7-0	MC_DAT	rw	访问 USB 寄存器的数据

3.EP0_CNT，EP1_CNT，EP2_CNT，EP3_CNT，EP4_CNT

Bit	Name	RW	Description
31-0	EP(n)_CNT	w	usb endpoint 0-4 发送数据个数，单位为 byte

4.EP0_ADR

Bit	Name	RW	Description
31-0	EP0_ADR	w	usb endpoint 0 发送/接收数据的起始地址

5.EP1_TADR，EP2_TADR，EP3_TADR，EP4_TADR

Bit	Name	RW	Description
31-0	EP(n)_TADR	w	usb endpoint 1-4 发送数据的起始地址

6.EP1_RADR，EP2_RADR，EP3_RADR，EP4_RADR

Bit	Name	RW	Description
31-0	EP(n)_RADR	w	usb endpoint 1-4 接收数据的起始地址

7.USB_IO_CON0

Bit	Name	RW	Description
31-15	-	-	预留
14	DMDIEH	rw	3.3V DM 数字输入使能
13	DPDIEH	rw	3.3V DP 数字输入使能
12	SR	rw	输出驱动能力选择 0:slow 1:fast
11	IO_MODE	rw	IO 模式使能 0:USB 模式 1:普通 IO 模式
10	DMDIE	rw	1.2V DM 数字输入使能
9	DPDIE	rw	1.2V DP 数字输入使能
8	IO_PU_MODE	rw	IO 上下拉分两种模式 0:USB 模式 1:普通 IO 模式
7	DMPU	rw	DM 上拉
6	DPPU	rw	DP 上拉
5	DMPD	rw	DM 下拉
4	DPPD	rw	DP 下拉
3	DMIE	rw	DM 方向 0:输出 1:输入
2	DPIE	rw	DP 方向 0:输出 1:输入
1	DMOUT	rw	DM 输出电平
0	DPOUT	rw	DP 输出电平

8.USB_IO_CON1

Bit	Name	RW	Description
31-2	-	-	预留
1	DMIN	r	DM 输入电平
0	DPIN	r	DP 输入电平

第 11 章. AUDIO IIS

11.1. 概述

Audio IIS 接口(以下简称 ALNK)是一个通用的双声道音频接口，用于连接片外的 DAC 或 ADC，连接信号有 MCLK，SCLK，LRCK，DATA。支持 IIS/左对齐/右对齐/DSP0/DSP1 共 5 种模式，原生支持 16/24bit 数据位宽，对 18/20/32bit 位宽的设备可提供兼容支持。ALNK 通过 DMA 的方式与 片内系统进行数据连接，不论输入或输出，每条通道占用 buffer 的大小可由软件配置。

MCLK 是音频接口的主时钟，Delta-Sigma 类型的 DAC/ADC 都是需要此信号的。MCLK 可由 ALNK 提供给 DAC/ADC，也可由 DAC/ADC 提供给 ALNK。

ALNK 可配置为主机模式或从机模式。主机模式是指 SCLK 和 LRCK 由本模块提供，此时需从外部提供相应采样率参考时钟。有 3 种方式可供选择：

1) 挂接 22.5792MHz 晶振以支持 44.1KHz 组别的采样率。

2) 挂接 24.576MHz 晶振以支持 48/32KHz 组别的采样率。

3) 从 PA4/PC0/PG8 端口输入相应频率的 MCLK

从机模式是指 SCLK 和 LRCK 由外部 DAC/ADC 提供，此时采样率由外部提供的 LRCK 确定，因此芯片不需外接晶振来提供采样率参考时钟。

ALNK 具备 4 条独立的通道，可相互独立地工作，每条通道都可独立配置为输入或输出，也可配置为不同的连接模式。需注意的是它们共用了 MCLK/SCLK/LRCK 信号，因此使用上有一定的 限制。根据 LRCK 信号的不同，将 IIS/左对齐/右对齐定义为基本模式，DSP0/DSP1 定义为扩展模式。有：

1) 4 条通道只能同时工作于基本模式或扩展模式。

不可一些通道工作于基本模式，另一些通道工作于扩展模式。

2) 基本模式下每条通道都只支持双声道立体声。

扩展模式下每条通道均可支持单声道或立体声，这由硬件自动适应，当每帧的 SCLK 时钟个数大于等于立体声所需的时钟个数时，该通道工作于立体声状态，否则工作于单声道状态（只有左声道）。

下表为 ALNK 支持的采样率设置和 MCKD 关系的列表

SR(KHz)	64fs	128fs	192fs	256fs	384fs	512fs	768fs
8					3.072M 可用		6.144M 推荐
11.025				2.8224M 可用		5.6448M 推荐	
12				3.072M 可用		6.144M 推荐	

16			3.072M 可用		6.144M 可用		12.288M 推荐
22.05		2.8224M 可用		5.6448M 可用		11.2896M 推荐	
24		3.072M 可用		6.144M 可用		12.288M 推荐	
32			6.144M 可用		12.288M 可用		24.576M 推荐
44.1		5.6448M 可用		11.2896M 可用		22.5792M 推荐	
48		6.144M 可用		12.288M 可用		24.576M 推荐	
64			12.288M 可用		24.576M 推荐		49.152M 可用
88.2	5.6448M 可用	11.2896M 可用		22.5792M 推荐		45.1584M 可用	
96	6.144M 可用	12.288M 可用		24.576M 推荐		49.152M 可用	
128			24.576M 推荐		49.152M 可用		
176.4	11.2896M 可用	22.5792M 推荐		45.1584M 可用			
192	12.288M 可用	24.576M 推荐		49.152M 可用			

下表为 ALNK0 和 ALNK1 使用的 IO 端口列表

	信号名称	占用的 IO 端口	
		ALNK0_IOS=0	ALNK0_IOS=1
ALNK0	MCLK	PA8	PA15
	SCLK	PA2	PA9
	LRCK	PA3	PA10
	CH0DAT	PA4	PA11
	CH1DAT	PA5	PA12

	CH2DAT	PA6	PA13
	CH3DAT	PA7	PA14
ALNK1	信号名称	占用的 IO 端口	
	MCLK	PB0	
	SCLK	PC0	
	LRCK	PC1	
	CH0DAT	PC2	
	CH1DAT	PC3	
	CH2DAT	PC4	
	CH3DAT	PC5	

下表为每路 ALNK 使用的 SFR 列表

SFR 名称	描述
ALNK_CON0	ALNK 控制寄存器 0
ALNK_CON1	ALNK 控制寄存器 1
ALNK_CON2	ALNK 控制寄存器 2
ALNK_CON3	ALNK 控制寄存器 3
ALNK_ADR0	通道 0 DMA 起始地址寄存器
ALNK_ADR1	通道 1 DMA 起始地址寄存器
ALNK_ADR2	通道 2 DMA 起始地址寄存器
ALNK_ADR3	通道 3 DMA 起始地址寄存器
ALNK_LEN	通道 0-3 DMA 数据长度寄存器

AC696X 中包含两个完全相同的 audio link 模块(ALNK0 和 ALNK1)，ALNK0 和 ALNK1 相互独立，因此，AC696X 支持同时 4 声道输入和 4 声道输出。

AC696X 中，ALNK0 的中断向量号为 11，ALNK1 的中断向量号为 36。

由于 ALNK0 和 ALNK1 的内部结构一模一样，这里只对 ALNK0 的相关寄存器配置和数据组织结构进行说明。

11. 2. 控制寄存器

1. ALNK_CON0: control register 0 (16bit addressing)

Bit	Name	RW	Description
15	FLAG3	rw	通道 x dma buffer flag 0: 当前正在使用 buf0，buf1 可被读写 1: 当前正在使用 buf1，buf0 可被读写
14	FLAG2	rw	
13	FLAG1	rw	
12	FLAG0	rw	
11	ALNKE	rw	ALNK 模块使能 0:模块关闭 1:模块打开
10	SCKINV	rw	SCLK 边沿选择 0:SCLK 的下降沿更新数据，上升沿采样数据 1:SCLK 的上升沿更新数据，下降沿采样数据
9	F32E	rw	主机模式下，每帧数据的 SCLK 个数 0:64 SCLKs per-frame 1:32 SCLKs per-frame
8	MOE	rw	MCLK 输出时钟使能 0:不输出 MCLK 至对应 IO 端口 1:输出 MCLK 至对应 IO 端口
7	SOE	rw	SCLK/LRCK 时钟输出使能 0:不输出 SCLK/LRCK 至对应 IO 端口 1:输出 SCLK/LRCK 至对应 IO 端口
6	DSPEN	rw	ALNK 模块工作模式选择 0:ALNK 工作于基本模式（IIS/左对齐/右对齐） 1:ALNK 工作于扩展模式（DSP0/DSP1）
5~0	-	-	预留

2. ALNK_CON1: control register 1 (16bit addressing)

Bit	Name	RW	Description
15	T3DIR	rw	通道 x 方向设置 0:发送 1:接收 通道 x 数据位宽设置 0:16bit 1:24bit 通道 x 工作模式设置，与 DSPEN 一起决定该通道的工作模式 DSPEN TxMOD x 0 不使用通道 x
14	T3LEN	rw	
13~12	T3MOD	rw	
11	T2DIR	rw	
10	T2LEN	rw	
9~8	T2MOD	rw	
7	T1DIR	rw	

6	T1LEN	rw	0	1	IIS（数据延后 1bit）
5~4	T1MOD	rw	0	2	左对齐
3	T0DIR	rw	0	3	右对齐
2	T0LEN	rw	1	1	DSP0（数据延后 1bit）
1~0	T0MOD	rw	1	2	DSP1
			others		预留，不可设置

3. ALNK_CON2: control register 2 (8bit addressing)

Bit	Name	RW	Description
7	PND3	r	通道 x Pending，当 dma buf0 或 buf1 被使用完毕后，此位被硬件置 1
6	PND2	r	
5	PND1	r	
4	PND0	r	
3	CPND3	w	写 1 清除 Pending，写 0 无效
2	CPND2	w	
1	CPND1	w	
0	CPND0	w	

4. ALNK_CON3: control register 3 (8bit addressing)

Bit	Name	RW	Description
7~5	LRDIV[2:0]	rw	ALNK 采样率设置 000:从外部输入 LRCK（即从机模式） 001:LRCK 为 MCKD 的 1/64，即 64fs 010:LRCK 为 MCKD 的 1/128，即 128fs 011:LRCK 为 MCKD 的 1/192，即 192fs 100:LRCK 为 MCKD 的 1/256，即 256fs 101:LRCK 为 MCKD 的 1/384，即 384fs 110:LRCK 为 MCKD 的 1/512，即 512fs 111:LRCK 为 MCKD 的 1/768，即 768fs
4~2	MDIV[2:0]	rw	MCLK 前置分频器设置 000:MCKD 为 MCLK 的 1 分频 001:MCKD 为 MCLK 的 2 分频 010:MCKD 为 MCLK 的 4 分频 011:MCKD 为 MCLK 的 8 分频 1xx:MCKD 为 MCLK 的 16 分频
1~0	MDIV[1:0]	rw	设置 MCLK 的来源 00:从外部输入 MCLK 01:系统时钟

			10:OSC CLK 11:PLL ALNK CLK 注意：OSC CLK 和 PLL ALNK CLK 参考 Clock System Spec 的说明。

5. ALNK_ADR: alnk dma start address register (26bit addressing)

ALNK DMA 操作起始地址寄存器，在使用 ALNK 之前，必须由软件初始化对齐至 4Byte。

6. ALNK_LEN: alnk dma sample length register (16bit addressing)

ALNK DMA 样点长度寄存器，在使用 ALNK 之前，必须由软件初始化为 2 的倍数，允许写入值为 2-32768，超出此范围的设置值可能导致不可预料的错误。

例如，当 ALNK_LEN 设置为 100 时，则 ALNK 每条通道每次 DMA 过程消耗 100 个样点。此时每通道 DMA buffer 占用的空间为：

16bit data: 100 * 2(CH) * 2(Byte) * 2(dual buffer) = 800 Byte

24bit data: 50 * 2(CH) * 4(Byte) * 2(dual buffer) = 800 Byte

【注】：2(CH)代表左、右声道。

11.3. 数据组织结构

ALNK 的数据都是通过 DMA 的方式与片内系统连接的，使用了 dual-buffer（乒乓缓冲）的方式，每条通道的 buf0/buf1 容纳样点数由 ALNK_LEN 寄存器指定，总 buffer 需求为：

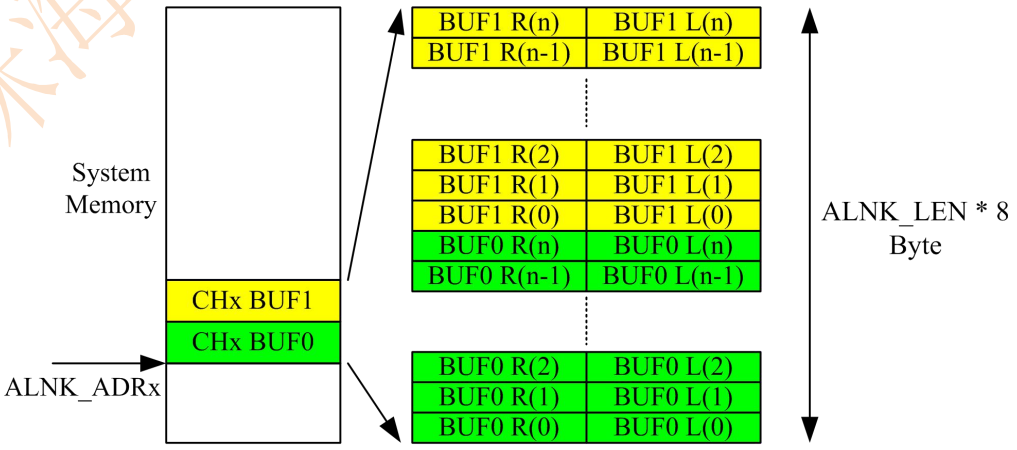
16bit data: ALNK_LEN * 2(CH) * 2(Byte) * 2(dual buffer) = ALNK_LEN * 8 Byte

24bit data: ALNK_LEN / 2 * 2(CH) * 4(Byte) * 2(dual buffer) = ALNK_LEN * 8 Byte

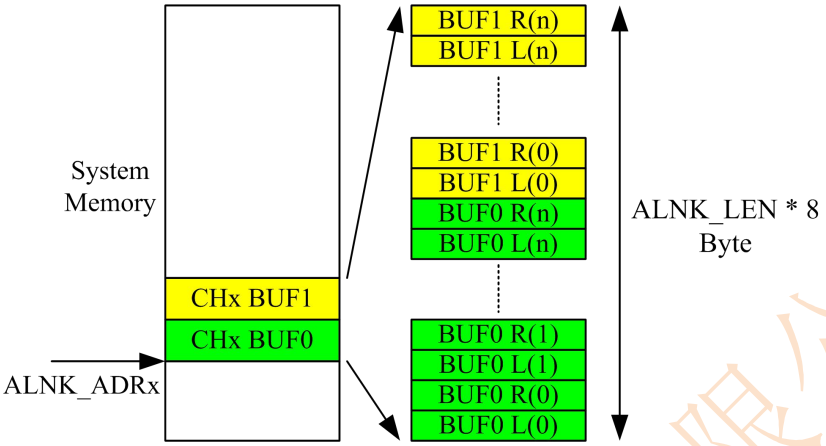
当四条通道一起使用时，共需要 4 倍大小的 buffer。

特别的，在单声道状态下，只有左声道会被使用到，右声道将被停用。

16bit 数据宽度时，数据组织如下图。



24bit 数据宽度时，数据组织如下图。



第 12 章. PDM LINK

12.1. 概述

Pdm Link(简称 PLNK)是一种数字麦克风(digital mic, DMIC)接口，存在主机和从机两种形式。主机形式的 PLNK 接口的同步时钟(SCLK)由 PLNK 本身产生；从机形式的 PLNK 接口的同步时钟(SCLK)由外部提供。

AC696X 中的 PLNK 为主机形式，该接口用于将外部输入进来的 DSM 形式的信号进行“解调”，恢复成数字码，留作后续处理。

下表为 PLNK 使用的 IO 端口列表

信号名称	占用 IO 端口	
PLNK_SCLK	IOMAP_CON1[20]=1	PA3
	IOMAP_CON1[20]=0	output channel2[4]
PLNK_DAT0	input channel8	
PLNK_DAT1	IOMAP_CON3[15]=0	PA4
	IOMAP_CON3[15]=1	input channel9

12.2. 寄存器说明

1.PLNK_CON: ADC control register (16bit addressing)

Bit	Name	RW	Description
31-16	-	-	预留
15	PND	r	当 dma buf0 或 buf1 被使用完毕后，此位被硬件置 1
14	CPND	w	写 1 清除 PND
13-10	-	-	预留
9	FLAG	r	通道 dma buffer flag 0: 当前正在使用 buf0，buf1 可被读写 1: 当前正在使用 buf1，buf0 可被读写
8	IE	rw	PLNK 中断使能
7-6	CH1MD	rw	通道 1 输入模式选择 0: 输入口 PLNK_DAT1 上升沿采样 1: 输入口 PLNK_DAT1 下降沿采样 2: 输入口 PLNK_DAT0 上升沿采样 3: 输入口 PLNK_DAT0 下降沿采样
5	CH1SC	rw	通道 1 增益控制 0:0dB 1:-6dB

4	CH1EN	rw	通道 1 使能
3-2	CH0MD	rw	通道 0 输入模式选择 0: 输入口 PLNK_DAT0 上升沿采样 1: 输入口 PLNK_DAT0 下降沿采样 2: 输入口 PLNK_DAT1 上升沿采样 3: 输入口 PLNK_DAT1 下降沿采样
1	CH0SC	rw	通道 0 增益控制 0:0dB 1:-6dB
0	CH0EN	rw	通道 0 使能

7.PLNK_SMR:plnk clock divider register(8bit)

SCLK 分频设置，分频数为 lsb_clk/(PLNK_SMR+1)

2.PLNK_ADR: plnk dma start address register (25bit addressing)

PLNK DMA 操作起始地址寄存器，在使用 PLNK 之前，必须由软件初始化对齐至 4Byte。

3.PLNK_LEN: plnk dma sample length register (16bit addressing)

PLNK DMA 样点长度寄存器，在使用 PLNK 之前，必须由软件初始化为 2 的倍数，允许写入值为 2-32768，超出此范围的设置值可能导致不可预料的错误。

例如，当 PLNK_LEN 设置为 100 时，则 PLNK 每条通道每次 DMA 过程消耗 100 个样点。此时每通道 DMA buffer 占用的空间为：

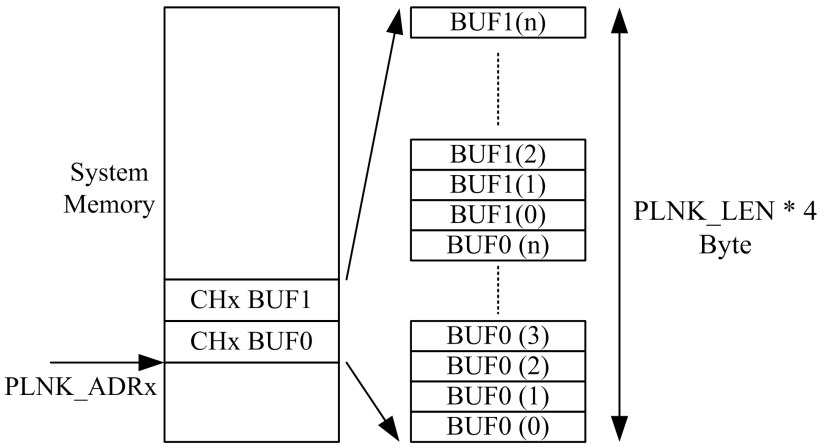
100 * 2(Byte) * 2(dual buffer) = 400 Byte

12.3.数据组织架构

PLNK 的数据都是通过 DMA 的方式与片内系统连接的，使用了 dual-buffer（乒乓缓冲）的方式，每条通道的 buf0/buf1 容纳样点数由 PLNK_LEN 寄存器指定，总 buffer 需求为：

PLNK_LEN * 2(Byte) * 2(dual buffer) = PLNK_LEN * 4 Byte

数据组织如下图



第 13 章. PULSE COUNTER

13.1. 概述

以下为 pulse_counter/触摸键的寄存器说明及使用方法。

13.2. 寄存器说明

1.PL_CNT_CON

Bit	Name	RW	Description
7-4	-	-	预留
3-2	clk_sel	rw	pulse counter 时钟选择 00: 选择 osc_clk 作为计数时钟; 01: 选择 clk_mux_in 作为计数时钟（iomc3[18:13] intput channel 选择）; 10: 选择 pll_192m 作为计数时钟; 11: 选择 pll_240m 作为计数时钟; 备注：时钟选择的频率越大，pl_cnt_value 计数值会越大，分辨率会越高，触摸键的灵敏度也越高
1	en	rw	触摸键使能，当 cap_mux_in（input channel 2）为 1 时，PLCNTVL 累加计数，计满后归 0 再重新累加 0: 触摸键禁止; 1: 触摸键使能
0	test_en	rw	触摸键测试使能（此位专用于测试） 0: 测试模式未使能; 1: 测试模式使能;

NOTE:当 cap_mux_in(input channel 2)选择 TMRx_PWMOUT, pulse counter 时钟选择 clk_mux_in（input channel 4）时，可以用内部固定周期的 PWM，计算芯片 IO 上不确定的时钟频率。

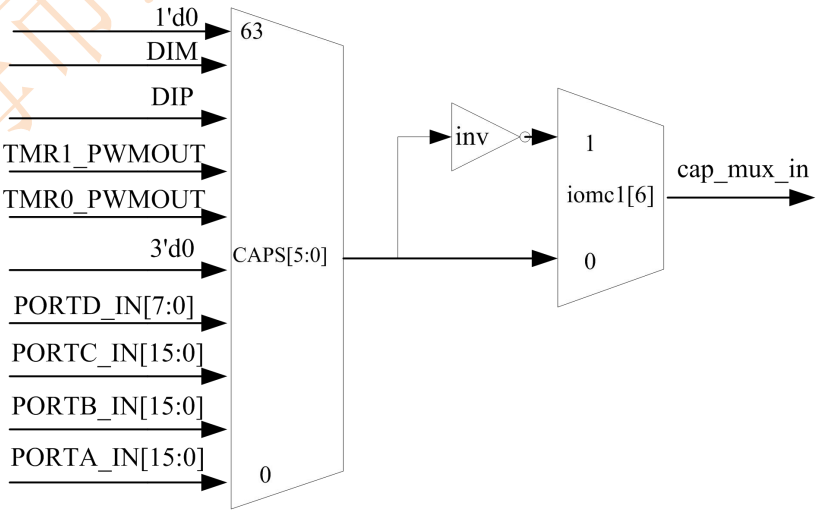


图 13-1 cap_mux_in(input channel 2)电路控制图

13.3. 例程

以下以 PA1 为例：

1.变量定义：

int Touchkey_value_old，Touchkey_value_new，Touchkey_value_delta；　　//32bit 有符号整数

2. 选择检测管脚

IOMC2 &= ~(0xff00)；//

IOMC2 |= 0x01 << 16；　　//选择 PA1

PORTA_OUT |= BIT(1)；

PORTA_DIR &= ~BIT(1)；　　//PA1 输出为 1

PORTA_PD |= BIT(1)；//PA1 下拉打开

3. 初始化计数器配置

PLCNTCON &= ~(0xC)；//

PLCNTCON |= BIT(3)；//选择 pll_192m 作为触摸键时钟

PLCNTCON |= BIT(1)；//使能计数器

4. 检测电容

Touchkey_value_old = PLCNTVL；//读取旧值

PORTA_DIR |= BIT(1)；//对应管脚输入使能

While(PORTA_IN & BIT(1))；

Touchkey_value_new = PLCNTVL；

If(Touchkey_value_old　>　Touchkey_value_new)then

Touchkey_value_new += 0x10000；

Touchkey_value_delta = Touchkey_value_new - Touchkey_value_old；

第 14 章. RDEC

14.1. 概述

RDEC（rotate decoder）是一个用于旋转编码器检测的模块，它支持两线输入的旋转编码器，可以检测旋转方向和旋转步数。

AC696X 中，一共支持 3 路 RDEC，每路 RDEC 均可独立工作且互不影响。其中，RDEC0 的中断号为 25，RDEC1 的中断号为 49，RDEC2 的中断号为 50。

下表为各 RDEC 使用的 IO 端口列表

模块	信号名称	IO 选择	
RDEC0	RDEC0_SIN[0]	IOMAP_CON1[12]=0	IOMAP_CON1[12]=1
		PA3	Input channel 6
	RDEC0_SIN[1]	IOMAP_CON1[13]=0	IOMAP_CON1[13]=1
		PA4	Input channel 7
RDEC1	RDEC1_SIN[0]	IOMAP_CON2[6]=0	IOMAP_CON2[6]=1
		PB2	Input channel 6
	RDEC1_SIN[1]	IOMAP_CON2[7]=0	IOMAP_CON2[7]=1
		PB3	Input channel 7
RDEC2	RDEC2_SIN[0]	IOMAP_CON2[14]=0	IOMAP_CON2[14]=1
		PB4	Input channel 6
	RDEC2_SIN[1]	IOMAP_CON2[15]=0	IOMAP_CON2[15]=1
		PB5	Input channel 7

下表为每路 RDEC 使用的 SFR 列表

SFR 名称	描述
RDEC_CON	RDEC 控制寄存器
RDEC_DAT	RDEC 结果读出寄存器

14. 2. 控制寄存器

1. RDEC_CON8bit 宽度

Bit	Name	RW	Description
7	PND	r	PND，只读。写入无效。
6	CPND	w	CPND，只写。写入 1 清除 PND，读出永远为 0。
5~2	RDEC_SPD	rw	RDEC_SPD，采样速率设置 Tsr = (2^ RDEC_SPD) / Flsb Flsb 为低速外设总线的频率，软件应当设置 RDEC_SPD 使 Tsr 介于 0.5~2mS 之间
1	RDEC_POL	rw	RDEC_POL 0：输入引脚无信号时处于 1 状态（外部引脚上拉） 1：输入引脚无信号时处于 0 状态（外部引脚下拉）
0	EN	rw	RDEC_EN 0: 模块关闭 1: 模块打开

2. RDEC_DAT 8bit 宽度，只读

Bit	Name	RW	Description
7~0	RDEC_DAT	r	此寄存器为 8 位有符号数，表示旋转编码器正反向旋转的步数。 当检测到旋转编码器动作时，PND 会设置为 1，软件可通过中断或查询方式来读取此寄存器。也可以完全不理会 PND，软件隔一段时间来访问此寄存器，但需注意检测时间不能过长，如果此寄存器的数值超过-128~127 的区间，将会产生溢出，无法表示出旋转编码器的正确动作。 【注意】：清 PND 的动作会将此寄存器一同清 0。

第 15 章. SPDIF

15.1. 概述

spdif（Sony/Philips Digital Interface）音频接口包含一个 mater 和 slave；master 和 slave 各有一个 IO 和外部进行通讯。

15.2. slave 使用介绍

打开模块使能 ss_en，触发模块开始 ss_str，硬件会在接收的 channel status bit 的数目（以 byte 为单位）等于所设置的数目（csbr_cnt，以 byte 为单位）时，产生中断，通过读取 ss_csb5~0（采集的方式是从高位移位进寄存器）的信息知道接收的音频数据格式等相关信息；

再设置数据位数模式 dat_dma_md(16/24bit)，并打开 dat_dma_en(inf_dma_en 根据需要打开)；dma（dat/inf）的模式采用乒乓 buffer 的方式，中断产生后可以通过读取相应的 buffer（dat/inf）使用情况（dat_pha/inf_pha）来判断哪块 buffer 可以用。

15.3. 寄存器说明

1.ss_con ：spdif slave configuration (32bits)

Bit	Name	RW	Description
31:16	rle_cnt	w/r	接收电平标志计数值
15:14	ss_cks	w/r	模块采样时钟选择 0:pll_192m（默认） 1:pll_96m 2:pll_48m 3:1’b1
13	inf_pha	r	0:当前使用为 information buffer0 1:当前使用为 information buffer1
12	dat_pha	r	0:当前使用为 data buffer0 1:当前使用为 data buffer1
11	rl_err	r	电平长度接收错误标志。 当模块在正常收数的过程中，突然接收到的高/低电平（热拔或者是光纤的自激行为）的计数值（以时钟周期为单位） 等于 rle_cnt 计数值时产生的标志； 建议接收到该标志时，将模块复位重启。
10	d_err	r	数据错误：接收到的这包数据中偶校验出错或者有数据的 valid_bit 为 1
9	b_err	r	块错误：块的序文（Z）接收不匹配

8	f_err	r	帧错误：子帧的序文（X/Y）接收不匹配
7	err_clr	w	错误清除，可以同时清除 f_err，b_err，d_err，rl_err
6	i_pnd	r	information 接收数目达到 inf_len 后中断标志
5	i_pnd_clr	w	information pnd clear
4	d_pnd	r	data 接收数目达到 dat_len 后中断标志
3	d_pnd_clr	w	data pnd clear
2	reserve		
1	ss_str	w	start 脉冲，启动工作
0	ss_en	w/r	模块使能

2.SS_IO_CON: spdif slave io configuration (32bits)

Bit	Name	RW	Description
31:28	reserve		
27	Io_doe	w/r	spdif 输出 io 使能控制
26:24	io_selo	w/r	3’b100:io 引脚 a 信号输出 3’b 101:io 引脚 b 信号输出 3’b 110:io 引脚 c 信号输出 3’b 111:io 引脚 d 信号输出 3’b 0XX:无输出
23:20	et_den	w/r	4’b0001: io 引脚 a 拔出检测使能 4’b0010: io 引脚 b 拔出检测使能 4’b0100: io 引脚 c 拔出检测使能 4’b1000: io 引脚 d 拔出检测使能
19	et_pnd	r	设备拔出（extract）pnd
18	et_pnd_clr	w	设备拔出（extract）pnd 清除
17	is_pnd	r	设备插入（inset）pnd
16	is_pnd_clr	w	设备插入（inset）pnd 清除
15:12	online	r	4’b0001: io 引脚 a 有设备在线 4’b0010: io 引脚 b 有设备在线 4’b0100: io 引脚 c 有设备在线 4’b1000: io 引脚 d 有设备在线
11	reserve		
10:8	is_det	r	1:io 引脚 a 有设备插入

			2:io 引脚 b 有设备插入 3:io 引脚 c 有设备插入 4:io 引脚 d 有设备插入 others:无
7:4	is_den	w/r	4’b0001: io 引脚 a 插入检测使能 4’b0010: io 引脚 b 插入检测使能 4’b0100: io 引脚 c 插入检测使能 4’b1000: io 引脚 d 插入检测使能
3	reserve		
2:0	io_seli	w/r	3’b100:io 引脚 a 输入 3’b 101:io 引脚 b 输入 3’b 110:io 引脚 c 输入 3’b 111:io 引脚 d 输入 3’b 0XX:无输入

3.SS_DMA_CON: dma configuration (32bits)

Bit	Name	RW	Description
31:22	reserve		
21	b_err_ie	w/r	b_err 中断使能
20	f_err_ie	w/r	f_err 中断使能
19	et_pnd_ie	w/r	et_pnd 中断使能
18	is_pnd_ie	w/r	is_pnd 中断使能
17	i_pnd_ie	w/r	i_pnd 中断使能
16	d_pnd_ie	w/r	d_pnd 中断使能
15	c_pnd_ie	w/r	c_pnd 中断使能
14	csbr_pnd	r	channel status bit 接收到达所要数目标志
13	csbr_pnd_clr	w	csbr_pnd clear
12:8	csbr_cnt	w/r	channel status bit 接收数目（byte 为单位）
7:4	reserve		
3	rle_dect_en	w/r	receive level error detect enable
2	inf_dma_en	w/r	information dma enable
1	dat_dma_md	w/r	0: 24bit 1: 16bit

0	dat_dma_en	w/r	data dma enable

4.SS_DMA_LEN:inf_len + dat_len (32bits)

Bit	Name	RW	Description
31:16	inf_dma_len	w/r	信息 dma 的样点数
15:0	dat_dma_len	w/r	数据 dma 的样点数

第 16 章. IRFLT

16.1. 概述

IRFLT 是一个专用的硬件模块，用于去除掉红外接收头信号上的窄脉冲信号，提升红外接收解码的质量。IRFLT 使用一个固定的时基对红外信号进行采样，必须连续 4 次采样均为‘1’时，输出信号才会变为‘1’；必须连续 4 次采样均为‘0’时，输出信号才会变为‘0’。换言之，脉宽小于 3 倍时基的窄脉冲将被滤除。改变该时基的产生可兼容不同的系统工作状态，也可在一定范围内调整对红外信号的过滤效果。通过对 IOMC（IO re-mapping）寄存器的配置，可以将 IRFLT 插入到系统 6 个 timer 中某一个的捕获引脚之前。

例如通过 IOMC 寄存器选择了 IRFLT 对 timer1 有效，并且 IRFLT_EN 被使能之后，则 IO 口的信号会先经过 IRFLT 进行滤波，然后再送至 timer1 中进行边沿捕获。

16.1.1. 硬件接口

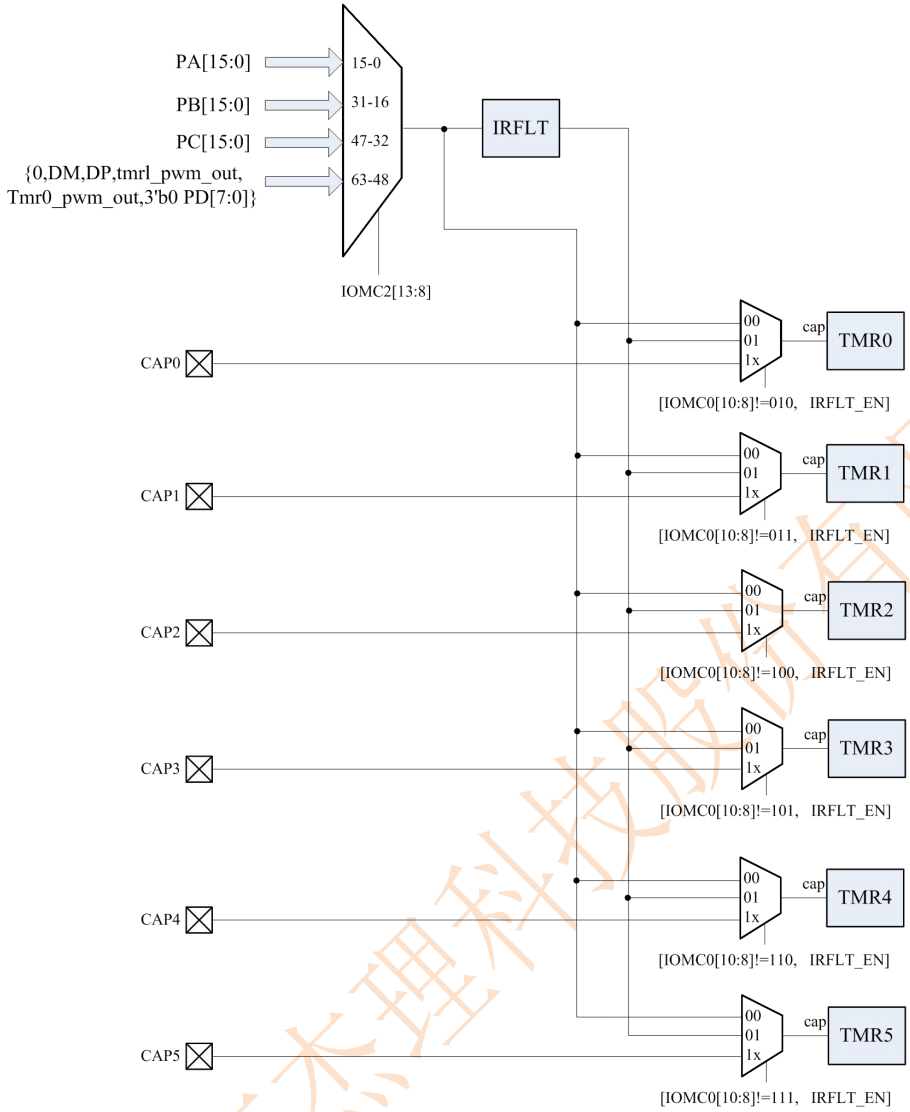


图 16-1 硬件接口示意图

16.1.2. 时基选择

PSEL 选定的分频倍数 N 和 TSRC 选定的驱动时钟的周期 Tc 共同决定了 IRFLT 用于采样红外接收信号的时基 Ts：

$$Ts = Tc * N$$

例如，当选择 32KHz 的 OSC 时钟，并且分频倍数为 1 时，Ts = 30.5uS。根据 IRFLT 的工作规则，所有小于(30.5*3=91.5uS)的窄脉冲信号，均会被滤除。

又如，当选择 48MHz 的系统时钟，并且分频倍数为 1024 时，Ts = 21.3uS。根据 IRFLT 的工作规则，所有小于(21.3*3=63.9uS)的窄脉冲信号，均会被滤除。

16. 2. 寄存器说明

1.IRFLT_CON: irda filter contrl register (8bit addressing)

Bit	Name	RW	Description
7-4	PSEL[3:0]	rw	PSEL[3-0]：时基发生器分频选择 0000：分频倍数为 1； 0001：分频倍数为 2； 0010：分频倍数为 4； 0011：分频倍数为 8； 0100：分频倍数为 16； 0101：分频倍数为 32； 0110：分频倍数为 64； 0111：分频倍数为 128； 1000：分频倍数为 256； 1001：分频倍数为 512； 1010：分频倍数为 1024； 1011：分频倍数为 2048； 1100：分频倍数为 4096； 1101：分频倍数为 8192； 1110：分频倍数为 16384； 1111：分频倍数为 32768；
3-2	TSRC[1-0]	r	TSRC[1-0]：时基发生器驱动源选择 00：选择 LSB_CLK 来驱动时基发生器； 01：选择 RC 时钟来驱动时基发生器； 10：选择 OSC_CLK 时钟来驱动时基发生器； 11：选择 PLL_48M 时钟来驱动时基发生器；
1	reserved	r	预留
0	IRFLT_EN	rw	IRFLT_EN： IRFLT 使能 0：关闭 IRFLT； 1：打开 IRFLT；

16. 3. 例程

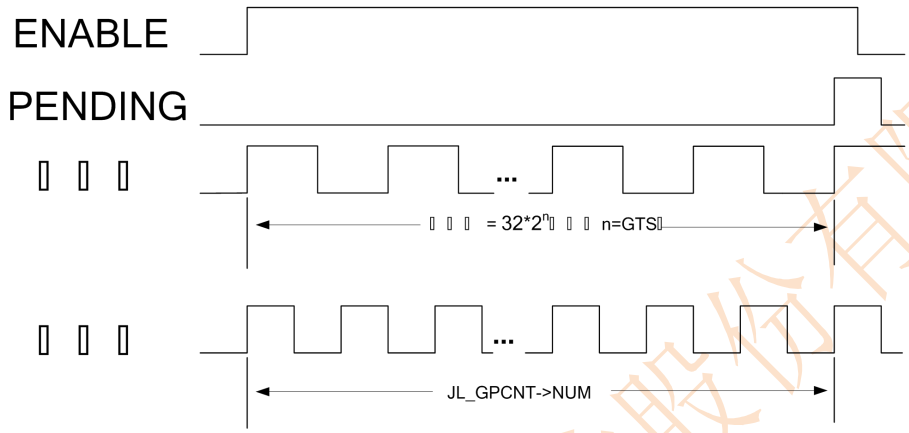
```
#define IR_PA8          0//PA8
void IR_init(void)
{
    ///选择 IO
    PORTA_DIR |= BIT(8);

    IOMC0 &= ~(0xF<<8);
    IOMC0 |= (IR_CTR<<8);//选择红外输入脚
```

```
    IOMC0 |= (2<<4);        //选择捕获 timer

    if(get_apb_clk() >= 12000000L)
    {
        IRFLT_CON = 0xa1;    //1024 分频倍数，使能 IR
        TMR2_CON  = 0x33;    //预分频 64，IO 口下降沿捕获
    }
    else
    {
        IRFLT_CON = 0x41;    //4096 分频倍数，使能 IR
        TMR2_CON  = 0x23;    //预分频 64，IO 口下降沿捕获
    }

    timer2_pad = (get_apb_clk()/1000)/64;

    printf("timer 2 init\n");
    int_enter_pr[TIMER2_INT-1] = timer2_isr;
}
```

第 17 章. GPCNT

17.1. 模块说明

GPCNT 为时钟脉冲计数器，用于计算两个时钟周期的比例，常用一个时钟（次时钟）计算另一个时钟（主时钟）的周期。（由于计算次时钟的周期数有 1 个 cycle 的误差，通常我们用频率较高的已知时钟作为次时钟，用频率较低的未知时钟作为主时钟）



17.2. 寄存器说明

1.JL_GPCNT->CON: GPCNT configuration register

Bit	Name	RW	default	Description
14:12	GSS	rw	0	主时钟选择： 000: lsb_clk 001: osc_clk 010: input channel2（见 IOMAP_Control.doc） 011: input channel4（见 IOMAP_Control.doc） 100: 时钟系统输入（见 clock_system.doc） 101: ring_osc 110: pll_480m 111: input channel1（见 IOMAP_Control.doc）
11:8	GTS	rw	0	主时钟周期数选择： 主时钟周期数 = $32*2^n$ （其中 n=GTS）
7	PND	rw	0	中断请求标志（当 **JL_GPCNT->CON[0]**置 1 后，主时钟达到 **JL_GPCNT->CON[11:8]**设定的周期

				数后，PENDING 置 1，并请求中断）： 0: 无 PENDING； 1: 有 PENDING
6	CLR_PND	rw	0	清除中断请求标志位： 0: 无效； 1: 清 PENDING
5:4	reserved	rw	0	
3:1	CSS	rw	0	次时钟选择： 000: lsb_clk 001: osc_clk 010: input channel2（见 IOMAP_Control.doc） 011: input channel4（见 IOMAP_Control.doc） 100: 时钟系统输入（见 clock_system.doc） 101: ring_osc 110: pll_480m 111: input channel1（见 IOMAP_Control.doc）
0	ENABLE	rw	0	GPCNT 模块使能位： 0: 不使能； 1: 使能

（注：需配置好其他位，才将 ENABLE 置 1。）

2.JL_GPCNT->NUM: The number of GPCNT clock cycle register

Bit	Name	RW	default	Description
31:0	NUM	r	0	在 **JL_GPCNT->CON[0]**置 1 到 PENDING 置 1（中断到来）之间，次时钟跑的周期数。