杰理 AD16N 芯片介绍 V1.1

珠海市杰理科技股份有限公司

Zhuhai Jieli Technology Co.,LTD

目录

第 1 章 引言

1.1 编写目的

此说明书主要对杰理 AD16N 芯片介绍。

AD16N 是一颗具备较强运算能力的 CPU，内部支持 64M 字节 FLASH 的 cache 寻址；内部有 40K 普通 RAM 空间以及 4*4K 的 CACHE RAM 空间。除此之外，AD16N 内部还有 32K OTP 空间；

1.2 文档修改日志

版本	日期	描述
1.0	2022 / 9 / 14	OTP 工程 SDK 手册初始版本
1.1	2023 / 3 / 9	添加 RDEC 模块寄存器说明，补充时钟系统结构框图
更新：		

第 2 章 cpu 介绍

2.1 说明

CPU 是一颗具备较强运算能力的 32 位处理器。CPU 有 64 个中断；有 8 级中断优先级。

2.2 cpu 内部 sfr

1) **icfg**: *icfg 是中断配置寄存器，包含一些*

Bit	名称	读/写	默认值	简介
31-10	-	-	-	-
9-8	GIE[1:0]	RW	0b10	总中断 = GIE[1] & GIE[0];同时打开才能进中断
7-0	-	-	-	-

2) **IDEL**: *CPU 进入 IDLE*
CPP 代码里面写：asm violate("idle");
ASM 代码里面写：idle;

2.3 中断说明

2.3.1 中断源

CPU 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
2-0	系统中断源	受 IE, IP 和 GIE 控制

2.3.2 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。：

2.3.3 总中断

CPU 增设了 GIE0 & GIE1。当 GIE0 于 GIE1 同时为 1 时，总中断才打开。

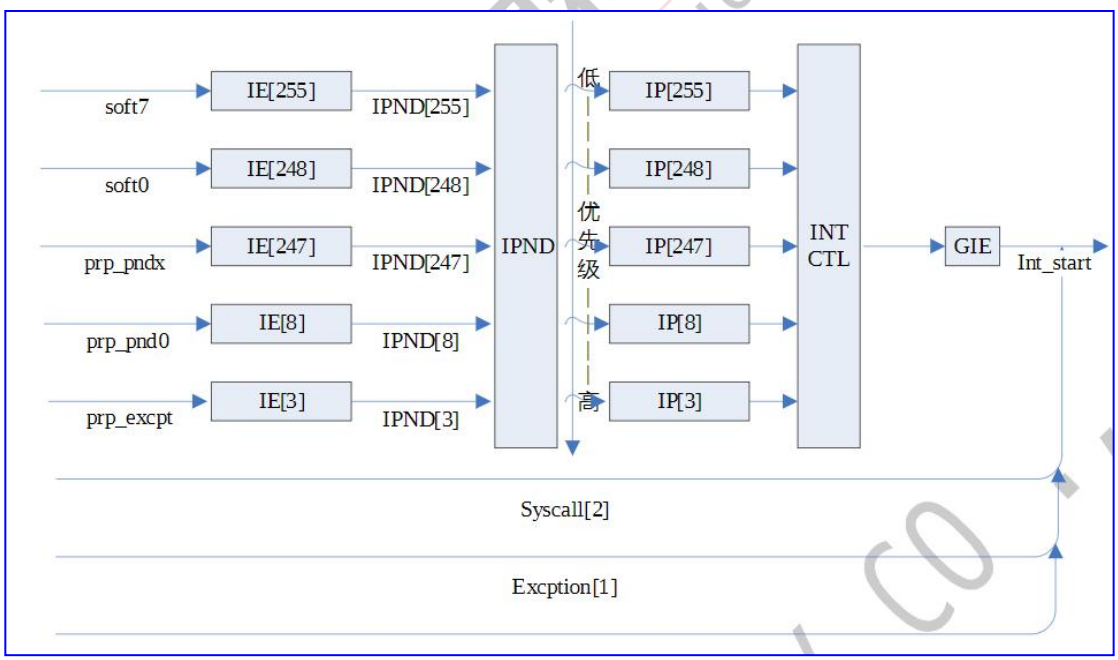


图 2.1 中断示意图

2.3.4 中断入口(中断列表，中断优先级)

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。

中断只会压 PC 入 RETI，中断返回时硬件从 RETI 取出返回地址。

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	
52	ICFG6[19:16]	BASE + 中断号×4	
51	ICFG6[15:12]	BASE + 中断号×4	
50	ICFG6[11:08]	BASE + 中断号×4	EQ
49	ICFG6[07:04]	BASE + 中断号×4	SRC_HW
48	ICFG6[03:00]	BASE + 中断号×4	
47	ICFG5[31:28]	BASE + 中断号×4	
46	ICFG5[27:24]	BASE + 中断号×4	AUDIO
45	ICFG5[23:20]	BASE + 中断号×4	
44	ICFG5[19:16]	BASE + 中断号×4	
43	ICFG5[15:12]	BASE + 中断号×4	
42	ICFG5[11:08]	BASE + 中断号×4	
41	ICFG5[07:04]	BASE + 中断号×4	
40	ICFG5[03:00]	BASE + 中断号×4	
39	ICFG4[31:28]	BASE + 中断号×4	MCPWN_TMR
38	ICFG4[27:24]	BASE + 中断号×4	MCPWN_CHX
37	ICFG4[23:20]	BASE + 中断号×4	
36	ICFG4[19:16]	BASE + 中断号×4	PPM
35	ICFG4[15:12]	BASE + 中断号×4	PPS
34	ICFG4[11:08]	BASE + 中断号×4	SLCD
33	ICFG4[07:04]	BASE + 中断号×4	RDEC0
32	ICFG4[03:00]	BASE + 中断号×4	
31	ICFG3[31:28]	BASE + 中断号×4	LEDC
30	ICFG3[27:24]	BASE + 中断号×4	GPCNT
29	ICFG3[23:20]	BASE + 中断号×4	LRCT
28	ICFG3[19:16]	BASE + 中断号×4	OSA
27	ICFG3[15:12]	BASE + 中断号×4	LED
26	ICFG3[11:08]	BASE + 中断号×4	GPADC
25	ICFG3[07:04]	BASE + 中断号×4	PORT
24	ICFG3[03:00]	BASE + 中断号×4	
23	ICFG2[31:28]	BASE + 中断号×4	

22	ICFG2[27:24]	BASE + 中断号×4	
21	ICFG2[23:20]	BASE + 中断号×4	
20	ICFG2[19:16]	BASE + 中断号×4	PMU_SOFT
19	ICFG2[15:12]	BASE + 中断号×4	USB_SIE
18	ICFG2[11:08]	BASE + 中断号×4	USB_SOF
17	ICFG2[07:04]	BASE + 中断号×4	IIC0
16	ICFG2[03:00]	BASE + 中断号×4	SD0
15	ICFG1[31:28]	BASE + 中断号×4	
14	ICFG1[27:24]	BASE + 中断号×4	SPI1
13	ICFG1[23:20]	BASE + 中断号×4	SPI0
12	ICFG1[19:16]	BASE + 中断号×4	
11	ICFG1[15:12]	BASE + 中断号×4	UART1
10	ICFG1[11:08]	BASE + 中断号×4	UART0
09	ICFG1[07:04]	BASE + 中断号×4	
08	ICFG1[03:00]	BASE + 中断号×4	
07	ICFG0[31:28]	BASE + 中断号×4	
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

2.3.5 其他

1、清外设模块内部的中断即可清除外设中断源。

2、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

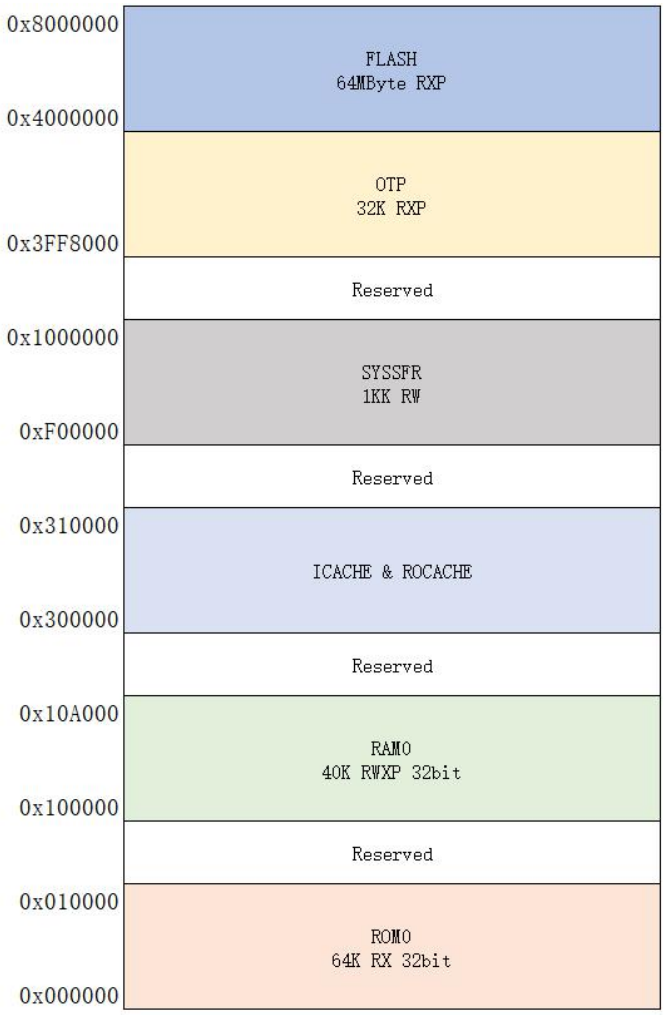
2.4 MEMORY

Memory 主要有对 FLASH 的管理、内部 RAM 的管理和内部 OTP 的管理：

从 0x4000000 开始映射内置 FLASH，一共有 64M 的地址空间；

从 0x100000 到 0x10a000 是系统普通 RAM 的区域。

从 0x3ff8000 到 0x4000000 是系统内部 OTP 的区域。



CPU MAPPING

第 3 章 时钟系统

3.1 时钟源

3.1.1 原生时钟源

UC03 具备如下 5 个原生时钟源，可直接驱动系统运行：

时钟	概述
rc_16m	内置 RC 振荡器，用于系统启动的初始时钟
lrc	内置低温度电压漂移 RC 振荡器，用于 PLL 参考时钟
xosc_12m	外挂 12-26MHz 晶振，可用于系统时钟和 PLL 参考时钟
xosc_32k	外挂 32.768MHz 晶振，用于 RTC 走时和 LRC trimming，不支持 PLL

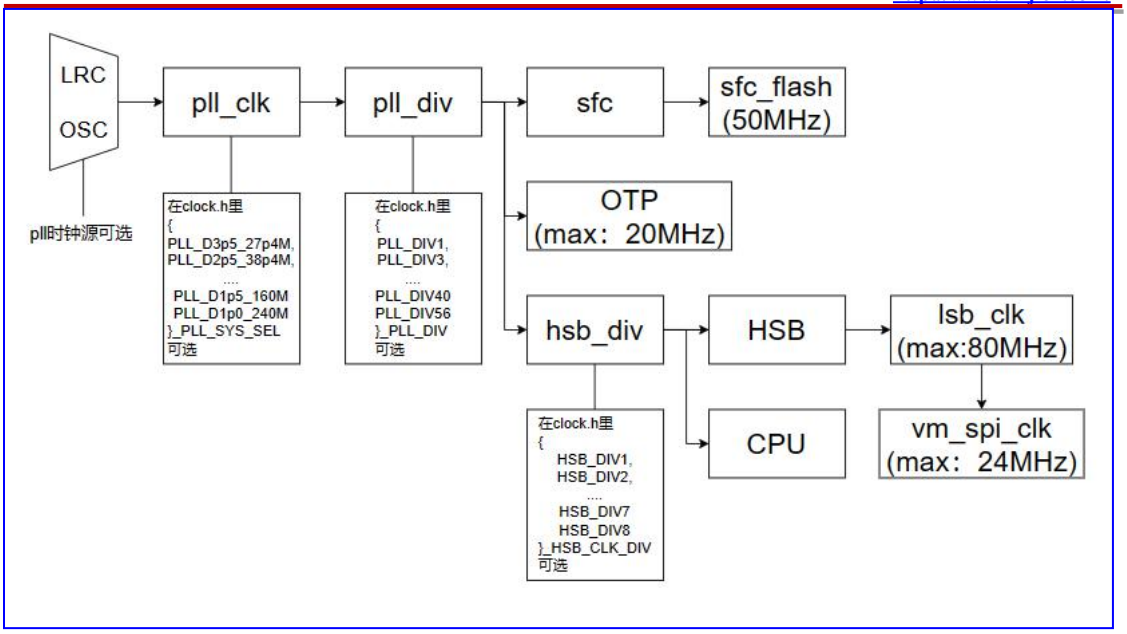
3.1.2 衍生时钟源

来自片内的 PLL0，可选的输出时钟有 96MHz、144MHz、192MHz 和 240MHz，且可同时输出 PLL0 的 1 分频，1.5 分频，2 分频，2.5 分频以及 3.5 分频的时钟。

如果 PLL0 配置为 192MHz，则能输出 192MHz、128MHz、96MHz、76.8MHz 和 54.86MHz；PLL0 配置其他输出时钟同理，最多可组合出 20 种不同的 PLL 时钟。

每个时钟都有独立的使能端。在不使用该时钟时，软件应关闭其使能端以防止额外电源消耗。

UC03 最大系统时钟 HSB 不超过 160MHz，LSB 时钟不超过 80MHz。OTP 的时钟不超过 20M。


LRC
OSC
pll时钟源可选
pll_clk
pll_div
sfc
sfc_flash
(50MHz)
在clock.h里
{
PLL_D3p5_27p4M,
PLL_D2p5_38p4M,
....
PLL_D1p5_160M
PLL_D1p0_240M
}_PLL_SYS_SEL
可选
在clock.h里
{
PLL_DIV1,
PLL_DIV3,
....
PLL_DIV40
PLL_DIV56
}_PLL_DIV
可选
OTP
(max：20MHz)
hsb_div
HSB
lsb_clk
(max:80MHz)
CPU
vm_spi_clk
(max：24MHz)
在clock.h里
{
HSB_DIV1,
HSB_DIV2,
....
HSB_DIV7
HSB_DIV8
}_HSB_CLK_DIV
可选

第 4 章 循环冗余校验(CRC)

4.1 概述

CRC(Cyclic Redundancy Check，循环冗余校验)主要用于数据的校验，每次运算 8bits，多项式为： $X^{16}+X^{12}+X^{5}+X^{1}$ （CRC-16/XMODEM）

CRC 运算不能直接使用寄存器操作，需要用 SDK 中提供的接口实现。

4.2 寄存器说明

1. JL_CRC->REG: CRC16 校验码

Bit	Name	RW	Default	RV	Dexcription
31-16	RESERVED	-	-	-	预留
15-0	CRC_REG	rw	x	-	写入初始值，CRC 计算完毕，读取校验码

2. JL_CRC->FIFO: 运算数据输入

Bit	Name	RW	Default	RV	Description
31:8	RESERVED	-	-	-	预留
7:0	CRC_FIFO	w	x	-	运算数据输入，HSB 先运算，LSB 后运算

第 5 章 看门狗

5.1 模块说明

WDT(watch dog timer)看门狗定时器用于防止系统软件进入死循环等不正确的状态。它设定了一个时间间隔，软件必须每在此时间间隔内进行进行一次“清看门狗”的操作，否则看门狗将溢出，并导致系统复位（或引发中断，主要用于程序的调试）。

每次系统复位之后，看门狗默认处于关闭的状态。软件可以随时将其打开。当发生系统复位时看门狗也会被关闭。

WDT 设计在 P33 系统里，读写 WDT_CON 需要用 P33 接口，写 CON 前无需再写 CRC_REG 打 key，并且支持低功耗模式下运行。

低功耗模式下，WDT 支持：

1. 直接复位芯片
2. 唤醒芯片，进入异常
3. 唤醒芯片，进入 RTC 中断（需关闭看门狗异常使能）

5.2 数字模块控制寄存器

1. P3_WDT_CON: Watchdog control register(8 bit addressing)

Bit	Name	RW	Default	RV	Description
7	PND	r	0	-	wdt 中断请求标志，当 WDRMD 设置为 1 时，WDT 溢出硬件会将此位置 1
6	CPND	w	-	-	写 1 清除中断标记位
5	WDRMD	rw	x	-	看门狗模式选择： 0：看门狗溢出将导致系统复位，这是看门狗的主要工作模式 1：看门狗溢出将 WINT 置 1，可产生中断或异常，这种模式主要用于调试
4	WDTEN	rw	0	-	看门狗定时器使能。 0：看门狗定时器关闭 1：看门狗定时器打开
3~0	TSEL3-0	rw	x		看门狗溢出时间选择 0000：1mS 0001：2mS 0010：4mS 0011：8mS 0100：16mS 0101：32mS 0110：64mS 0111：128mS 1000：256mS

					1001：512mS 1010：1S 1011：2S 1100：4S 1101：8S 1110：16S 1111：32S Note：上述溢出时间只是参考值。实际上，wdt 由不准确的片内 RC 振荡器驱动，其实际溢出时间可能会有高达 100%的偏差，且不同芯片之间也无法保证一致性。所以在选择溢出时间时必须留有足够余量

2. P3_VLD_KEEP ： wdt exception register (8 bit addressing)

Bit	Name	RW	Default	RV	Description
7	RESERVED	-	-	-	预留
6	WDT EXPT EN	rw	0	0	看门狗异常使能 0: 看门狗异常关闭 1: 看门狗异常打开
5~0	RESERVED	-	-	-	预留

第 6 章 IIC 模块

6.1 模块说明

IIC 通讯模块。该芯片只有一个 IIC 模块，仅可做主机使用。IIC 有 4 组 IO：(请参考另外一章： IO_MAPPING_CONTROL)

6.2 数字模块控制寄存器

1. IIC_CON0 ： IIC control register(32 bit addressing)

Bit	Name	RW	Default	RV	Description
31-8	RESERVED	-	0	-	预留
7	IE	rw	0	0	收发结束中断使能
6	END_BIT_IE	rw	0	0	“结束位”中断使能 0 ： 不使能 1 ： 使能
5-4	RESERVED	-	0	-	预留
3	ADD_END_BIT	rw	0	0	添加“结束位”，只在主机模式有效 0 ： 不添加“结束位” 1 ： 添加“结束位”
2	ADD_START_BIT	rw	0	0	添加“起始位”，只在主机模式有效 0 ： 不添加“起始位” 1 ： 添加“起始位”
1	RESERVED	rw	0	0	必须为 0
0	EN	rw	0	0	IIC 接口使能 0 ： 关闭 IIC 接口 1 ： 打开 IIC 接口

2. IIC_STA ： IIC statement register (32 bit addressing)

Bit	Name	RW	Default	RV	Description
31-8	RESERVED	-	0	-	预留
7	PND	r	0	-	普通中断标志，iic 在做主机模式下，发生或接收 1byte 数据，硬件会将该位置 1
6	END_BIT_PND	r	0	-	“结束位”中断标志
5	CLR_PND	w	0	-	清除收发结束中断标志，写 1 清零

4	CLR_EB_PND	w	1	-	清除“结束位”中断标志，写 1 清零
3	ACK_IN	r	0	-	SDA 线上接收的响应标志 0 ： 响应（ACK） 1 ： 不响应（NACK）
2	ACK_OUT	rw	1	-	设置输出响应标志 0 ： 响应（ACK） 1 ： 不响应（NACK）
1	START_FLAG	r	0	-	“起始位”标志，需由 CLR_START_FLAG 写 1 清除
0	CLR_START_FLAG	w	0	-	清空“起始位”标志，写 1 清除

3. IIC_BUAD ： IIC baud register (8 bit addressing)

Bit	Name	RW	Default	RV	Description
7~0	IIC_BAUD	rw	0	-	IIC 的波特率配置寄存器或地址寄存器

（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 = Fsys_clk/((1+IIC_BAUD)*2)+1，电阻上拉的时间，上拉电阻越大，频率越低。

4. IIC_BUF ： IIC buf register (8 bit addressing)

Bit	Name	RW	Default	RV	Description
7~0	IIC_BUF	rw	0	0	IIC 的 buf 寄存器，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。

（1）做主机时，写 IIC_BUF 会启动一次通信，写什么值就会发什么值出去，当普通 pending 出现，则表示发送完毕。此时可检查确认位（ACK , IIC_STA[3]）,以了解从机是否接收到。

（2）若在启动通信前选择了“加结束位”，则发送完该字节后会顺便发送结束位。

（3）若在启动通信前选择了“加起始位”，则发送该字节前会先发送起始位。若启动通信前时钟和数据线处于空闲状态（上拉），则无论有无选择“加起始位”，起始位都会自动加上。

（4）若要接收则需要往 IIC_BUF 写 oxFF，等到普通 Pending 出现，则表示接收完毕，CPU 可以读 IIC_BUF 得到接收的值。若在写 0xFF 前选择了“加结束”，则接收完会顺便发送结束位。

第 7 章 SPI 模块

7.1 模块说明

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。该芯片有 2 个 SPI：SPI0、SPI1

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。

从机：SPI 接口时钟由片外 SPI 设备产生，提供给本机使用。

工作于主机模式时，SPI 接口的驱动时钟可配置，范围为 系统时钟~系统时钟/256。

工作于从机模式时，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，在系统不繁忙情况下可以接近系统时钟速率。

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。

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。

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。

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为 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。

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-(2^32 - 1)Byte。写 SPI_CNT 的动作将启动一次 DMA 传输。

[注意]：

该模块所有说明及配置仅作为 SPIx 模块使用，不适用到 SFC 中

7.2 数字模块控制寄存器

1. SPIx_CON ： SPIx control register0 (32 bit addressing)

Bit	Name	RW	Default	RV	Description
31-22	RESERVED	-	0	-	预留
21	OF_PND	r	0	-	Dma fifo overflow 中断，当 spi 做从机 dma 接收时，fifo 满且 dma 响应慢，而新接收的数据到来时导致 fifo 发出溢出就会产生 of_pnd **[注意]**仅做 debug 使用！！！
20	OF_PND_CLR	w	0	-	清除 of_pnd，写 1 清零
19	OF_IE	rw	0	1	of_pnd 使能位
18	UF_PND	r	0	-	dma fifo underflow 中断，当 spi 做从机 dma 发送时，fifo 空且 dma 响应慢，而主机端请求新数据时导致 fifo 读空就会产生 uf_pnd **[注意]**仅做 debug 使用
17	UF_PND_CLR	w	0	-	清除 uf_pnd，写 1 清零
16	UF_IE	rw	0	1	uf_pnd 使能位
15	PND	r	0	0	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 种方法清除此标志位 向 PCLR 写入‘1’ 写 SPIBUF 寄存器来启动一次传输 写 SPICNT 寄存器来启动一次 DMA
14	CPND	w	0	1	软件在此位写入‘1’将清除 PND 中断请求标志
13	IE	rw	0	1	SPI 中断使能 0 ： 禁止 SPI 中断 1 ： 允许 SPI 中断
12	DIR	rw	0	0	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	0	0	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，不可设置位此项
9-8	RESERVED	-	0	-	预留

7	CSID	rw	0	0	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	0	0	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	0	0	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	0	0	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	BDIR	rw	0	0	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	0	1	SPICS 信号使能，always set 为 1
1	SLAVE	rw	0	0	从机模式
0	SPIR	rw	0	0	SPI 接口使能 0 ： 关闭 SPI 接口 1 ： 打开 SPI 接口

2. SPI x _BAUD ： SPI baudrate setting register (8 bit addressing)

Bit	Name	RW	Default	RV	Description
7~0	SPI_BAUD	rw	0x0	0x0	SPI 主机时钟设置寄存器 SPICK = system clock / (SPIBAUD + 1)

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

Bit	Name	RW	Default	RV	Description
7~0	SPIx_BUF	rw	0x0	0x0	发送寄存器和接收寄存器共用此 SFR 地址。吸入至发送寄存器，从接收寄存器读出。

4. SPIx_ADR ： SPIx DMA address register (32 bit addressing)

Bit	Name	RW	Default	RV	Description
31~0	SPI_ADR	rw	0x0	0x0	SPI DMA 起始地址寄存器，只写，读出为不确定值。

5. SPIx_CNT ： SPIx DMA counter register (32 bit addressing)

Bit	Name	RW	Default	RV	Description
31~0	SPI_CNT	rw	0x0	0x0	SPI DMA 计数寄存器,读出为当下剩余读写数量。 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。

6. SPIx_CON1 ： SPIx control1 register

Bit	Name	RW	Default	RV	Description
31~2	RESERVED	-	-	-	预留
1-0	SPI_BIT_MODE	rw	0	0	设置 spi 输入输出位流模式，默认 0 为标准格式 0 ： [7,6,5,4,3,2,1,0] 1 ： [0,1,2,3,4,5,6,7] 2 ： [3,2,1,0,7,6,5,4] 3 ： [4,5,6,7,0,1,2,3]

传输波形图：

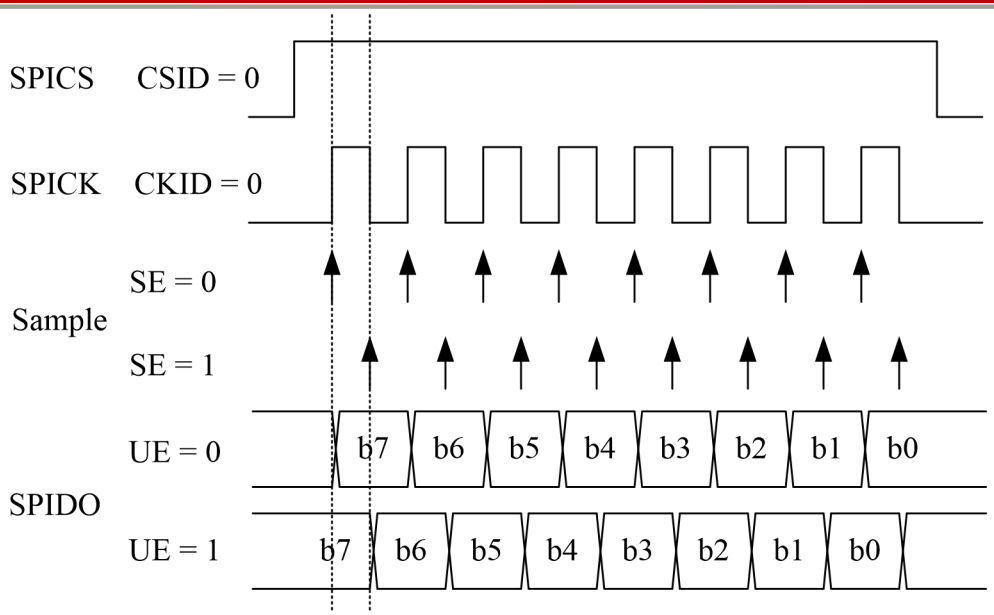


图 1-1　传输波形 1

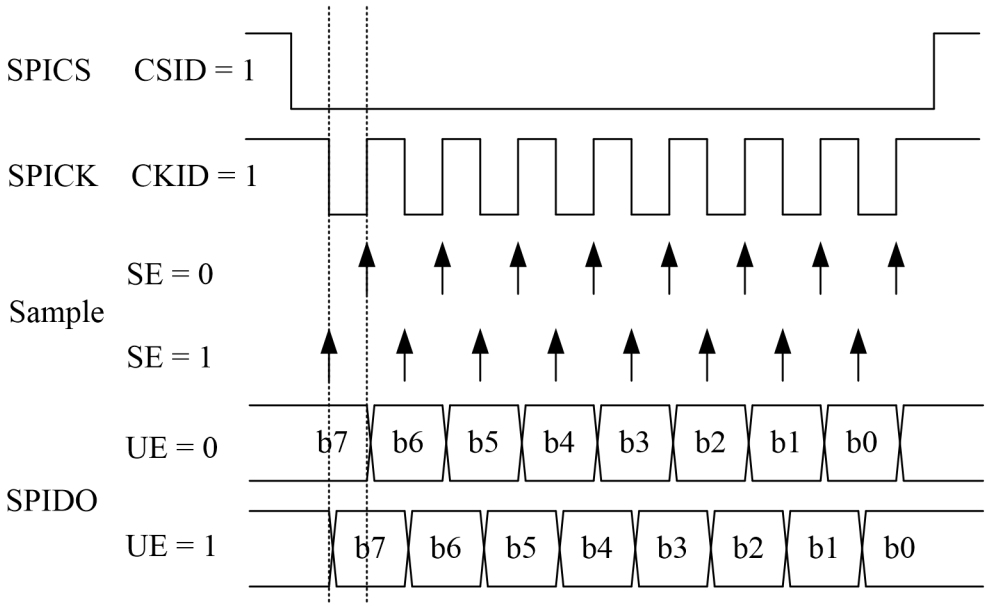


图 1-2　传输波形 2

第 8 章 数模转换器(ADC)

8.1 模块说明

10Bit ADC(A/D 转换器)，其时钟最大不可超过 1MHz。模块输出精度：10Bit。

8.2 数字模块控制寄存器

1. ADC_CON: ADC configuration register

Bit	Name	RW	Default	RV	Description
31-24	RESERVED	-	-	-	预留
23-21	ADC_ASEL	rw	0x0	0	ADC 模式选择： 000：无效 001：ADC 采集 IO 通道电压 010：ADC 采集内部模拟通道电压 其他：非法值
20-18	ADC_ASEL	rw	0x0	-	内部模拟通道选择： 000：PMU 其他：非法值
17	ADC_CLKEN	r	0	0	ADC 时钟使能
16	ADCISEL	w	0	0	ADC CMP power select, H is choosed to lowpower.
15-12	WAIT_TIME	rw	x	-	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	x	-	IO 通道选择： 0000：选择 PA0 0001：选择 PA4 0010：选择 PA6 0011：选择 PA8 0100：选择 PA10 0101：选择 PB0 0110：选择 PB2 0111：选择 PB4 1000：选择 PB6 1001：选择 PB7 1010：选择 PB8 1011：选择 PC0 1100：选择 PC1 1101：选择 PC3

					1110：选择 USBDP 1111：选择 USBDM
7	PND	r	1	-	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	X	-	CPND：只写，写‘1’清除中断请求位，写‘0’无效
5	ADC_IE	rw	0	-	ADC_IE：ADC 中断允许
4	ADC_EN	rw	0	-	ADC_EN：ADC 控制器使能
3	ADC_AE	rw	0	-	ADC_AE：ADC 模拟模块常使能
2-0	ADC_BAUD	rw	0x0	-	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. ADC_RES: ADC result 32-bit register

Bit	Name	RW	Default	RV	Description
31-10	Reserved	-	-	-	预留
9-0	ADC_RES	r	x	-	ADC 输出结果

第 9 章 32 位定时器(Timer32)

9.1 模块说明

Timer32 是一个集合了定时/计数/捕获/PWM 功能于一体的多动能 16 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。

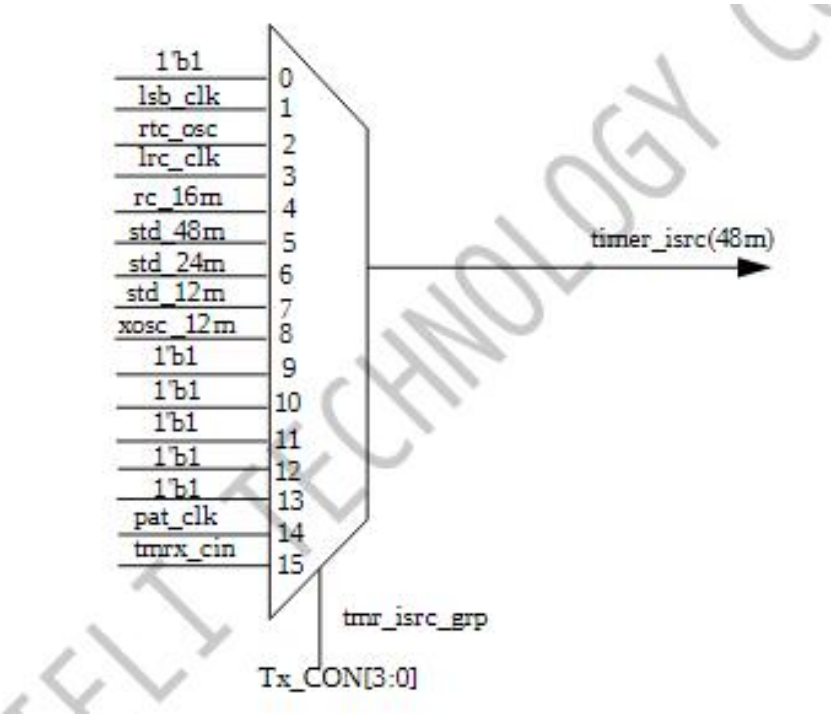


图 1 TIMER 时钟源选择示意图

9.2 控制寄存器

寄存器列表	TIMER0	TIMER1	TIMER2	TIMER3	TIMER4	TIMER5
Tx_CON	T0_CON	T0_CON	T2_CON	T3_CON	T4_CON	T5_CON
Tx_CNT	T0_CNT	T0_CNT	T2_CNT	T3_CNT	T4_CNT	T5_CNT
Tx_PRD	T0_PRD	T0_PRD	T2_PRD	T3_PRD	T4_PRD	T5_PRD
Tx_PWM	T0_PWM	T0_PWM	T2_PWM	T3_PWM	T4_PWM	T5_PWM

9.3 数字模块控制寄存器

1. Tx->CON: timer x control register

Bit	Name	RW	Default	RV	Description
31-17	RESERVED	-	-	-	预留
16	DUAL_EDGE_EN	rw	0	-	DUAL_EDGE_EN：双边沿捕获模式，当 MODE = 2 或 3 时，使能该位即变成双边沿捕获模式，在上沿和下沿会将 TxCNT 的值写入 TxPR。
15	PND	r	0	-	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	x	-	软件在此位写入‘1’将清除 PND 中断请求标志。
13-10	SSEL	rw	0x0	-	SSEL3-0：timer 驱动源选择见图 1 【注意】：选中的时钟需要使用上面 PSET 分频到（lsb_clk/2）一下
9	PWM_INV	rw	0	-	PWM0_INV：PWM 信号输出反向。
8	PWM_EN	rw	0	-	PWM0_EN：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	0x0	-	PSET1-0：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	CSEL	rw	0x0	-	捕获模式端口选择： 0：IO mux in (crossbar) 1：IRFLT_OUT
1-0	MODE	rw	0x0	-	MODE1-0：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 CNT 的值捕捉到 Tx_PR 中）； 11： IO 口下降沿捕获模式（当 IO 下降沿到来时，把 CNT 的值捕捉到 Tx_PR 中）。

2. Tx->CNT: timer x counter register

Bit	Name	RW	Default	RV	Description
31-0	Tx_CNT	rw	x	-	timer32 的计数寄存器

3. Tx->PR: timer x period register

Bit	Name	RW	Default	RV	Description
31-0	Tx_PR	rw	x	-	timer32 的计数周期寄存器

在定时/计数模式下，当 TxCNT == TxPR 时，Tx_CNT 会被清 0。

在上升沿/下降沿捕获模式下，Tx_PRD 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_PRD 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	Default	RV	Description
31-0	Tx_PWM	rw	x	-	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 10 章 红外滤波模块(IRFLT)

10.1 模块说明

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。

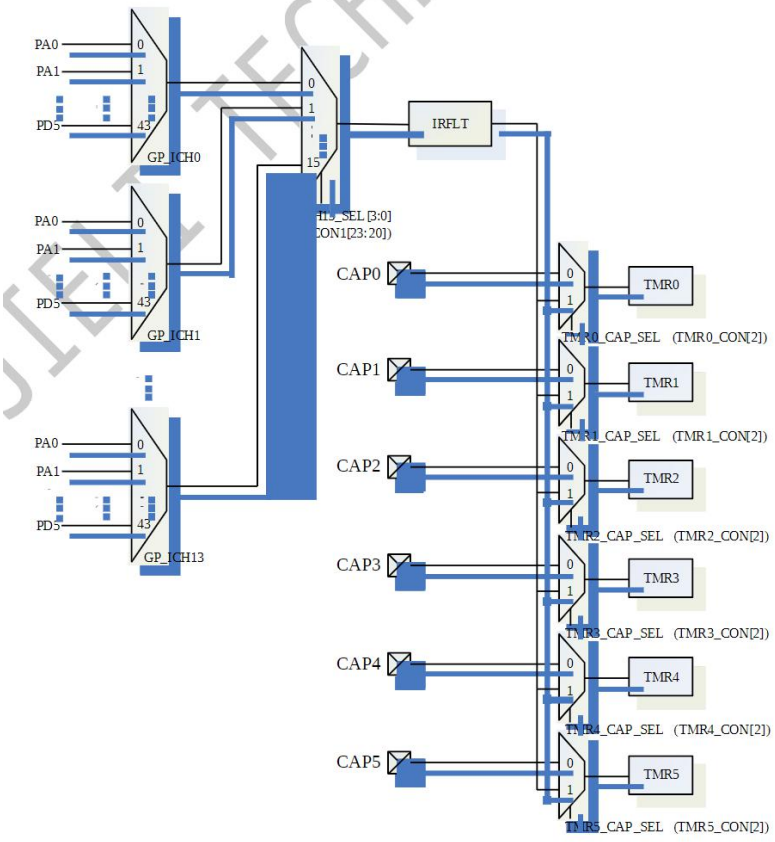


图 1 IRFLT 模块示意图

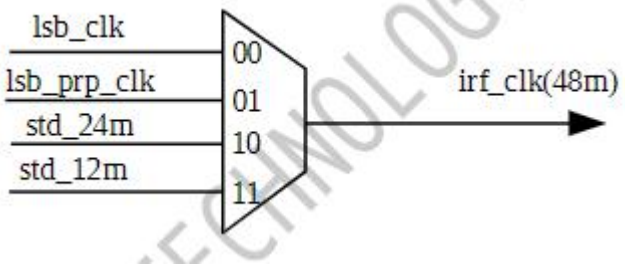


图 2　输入时钟源 irflt_clk 时钟结构

10.2 寄存器 SFR 列表

1. JL_IRFLT->CON: irda filter configuration register

Bit	Name	RW	Default	Rrv	Description
7-4	PSEL	rw	x	-	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	rw	x	-	时基发生器驱动源选择，见图 1
1	RESERVED	-	-	-	预留
0	IRFLT_EN	rw	0	-	IRFLT 使能 0: 关闭 IRFLT 1: 打开 IRFLT

10.3 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。

第 11 章 MCPWM

11.1 模块说明

11.1.1 概述

MCPWM 功能模块包括：8 个 MCTimer0 时基模块，8 对独立的 PWM 通道，4 路故障保护输入。框图如下：

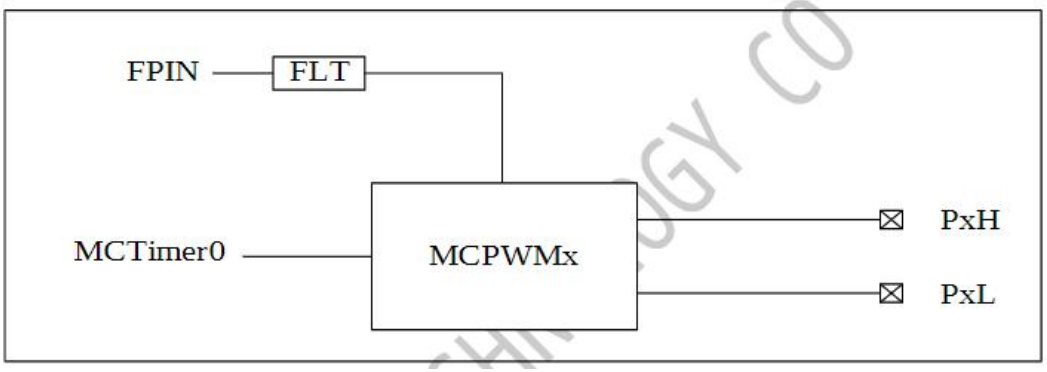


图 1 MCPWM 模块示意图

11.1.2 定时器 MCTIMER

1.概述

MCTimer 是 16 位定时器，可作为 PWMx 的时基控制

2.模块特性

（1）16 位定时功能

（2）带缓冲的周期寄存器

（3）支持多种工作模式：递增、递增-递减、外部引脚控制递增递减

（4）多种中断模式：上溢出中断、下溢出中断、上下溢出中断

11.1.3 模块引脚

PWM0: PWMCHXH、PWMCHXL

PWM1: PWMCHXH、PWMCH1L

PWM2: PWMCHXH、PWMCH2L

.....

故障输入引脚：FPIN

11.1.4 模块特性

1.边沿对齐和中心对齐输出模式
2.可运行过程中更改 PWM 频率、占空比
3.多种更新频率/占空比模式：上溢出重载、下溢出重载
4.灵活配置每对通道的有效电平状态
5.可编程死区控制
6.硬件故障输入引脚

11.2 数字模块控制寄存器

1. TMRx_CON: Timer control register

Bit	Name	RW	Default	RV	Description
31-16	RESERVED	-	-	-	预留
15	INCF	rw	0	-	递增递减标志位 0：递减 1：递增
13	UFPND	r	0	-	递减借位标志 0：没借位 1：已发生借位
12	OFPND	r	0	-	计数溢出（TMRXCNT==TMRXPR）标志 0：没溢出 1：已发生溢出
11	UFCLR	w	-	1	清除 UFPND 标志位 0：无效 1：清除 UFPND
10	OFCLR	w	-	1	清除 OFPND 标志位 0：无效 1：清除 OFPND
9	UFIE	rw	0	1	定时递减借位中断使能 0：禁止 1：允许
8	OFIE	rw	0	1	计数溢出中断使能 0：禁止 1：允许
7	CKSRC	rw	0	-	定时器时钟源 0：内部时钟 1：外部引脚时钟 TMRXCK(pwm0-3 有，pwm4-7 固定 l'b0)
6-3	CKPS	rw	0	-	时钟预分频设置，TCK/(2^TCKPS)

2	RESERVED	-	-	-	预留
1-0	MODE	rw	0	-	定时器工作模式 00：保持计数值 01：递增模式 10：递增递减循环模式 11：由外部引脚控制递增或递减

2. TMRx_CNT: counter initial value register

Bit	Name	RW	Default	RV	Description
15-0	TMR_CNT	rw	x	-	计数/定时初值

3. TMRx_PR: counter target value register

Bit	Name	RW	Default	RV	Description
15-0	TMR_PR	rw	x	-	计数/定时目标值

4. CHx_CON0: pwm configuration register0

Bit	Name	RW	Default	RV	Description
15-12	DTCKPS	rw	0	-	死区时钟预分频，Tsys/(2^DTCKPS)
11-7	DTPR	rw	0	-	死区时间控制 死区时间：Tsys/(2^DTCKPS)×(DTPR+1)
6	DTEN	rw	0	-	死区允许控制，对应 PWMCHxH、PWMCHxL 0：禁止 1：允许
5	L_INV	rw	0	-	对应 PWMCHxL 输出反向控制 0：反向禁止 1：反向允许
4	H_INV	rw	0	-	对应 PWMCHxH 输出反向控制 0：反向禁止 1：反向允许
3	L_EN	rw	0	-	对应 PWMCHxL 输出允许控制 0：禁止 PWMCHxH 1：允许 PWMCHxH
2	H_EN	rw	0	-	对应 PWMCHxH 输出允许控制 0：禁止 PWMCHxH 1：允许 PWMCHxH
1-0	CMP_LD	rw	0	-	CHx_CMP 重新载入控制 00：时基 TMRX_CNT 等于“0”载入 01：时基 TMRX_CNT 等于“0”或者等于

					TMRX_PR 的时候载入 10：时基 TMRX_CNT 等于 TMRX_PR 时载入 11： 立即载入

5. CHX_CON1: pwm control register1

Bit	Name	RW	Default	RV	Description
15	FPND	r	0	-	故障保护输入标志，只读，写无效 读 0：未发生保护 读 1：已发生保护，模块的 PWM 引脚会变成高阻态
14	FCLR	w	-	-	清除 FPND 标志位，只写，读为“0” 写 0：无效 写 1：清除 FPND
13-12	RESERVD	-	-	-	预留
11	INTEN	rw	0	1	FPND 中断允许 0：禁止 1：允许
10-8	TMRSEL	rw	0	-	选择 TMR0-7 作为 PWM 时基 0-7：选择时基
7-5	RESERVED	-	-	-	预留
4	FPINEN	rw	0	1	故障保护输入允许控制 0：禁止保护 1：允许保护
3	FPINAUTO	rw	0	1	故障自动保护控制 0：禁止自动保护 1：允许自动保护 当检测到故障引脚有效信号时，自动把 PWM 引脚设成高阻态，直到软件清除 FPND。
2-0	FPINSEL	rw	0	-	故障保护输入引脚选择 0-7：选择 FPIN 作为保障输入（FPIN 跟输出组数对应，多余 FPIN 输入固定为 0！！！）

6. CHX_CMPH: pwm high port compare register

Bit	Name	RW	Default	RV	Description
15-0	MMRX_PRH	rw	-	-	带缓冲的 16 位比较寄存器，对应 PWMCHxH 引脚的占空比控制

7. CHX_CMPL: pwm low port compare register

Bit	Name	RW	Default	RV	Description

15-0	TMRX_PRL	rw	-	-	带缓冲的 16 位比较寄存器，对应 PWMCHxL 引脚的占空比控制

8. FPIN_CON: input filter control register

Bit	Name	RW	Default	RV	Description
23-16	EDGE	rw	0	-	FPINx 边沿选择 0：下降沿 1：上升沿
15-8	FLT_EN	rw	0	-	FLT_EN：FPINx 滤波使能开关 0：滤波关闭 1：滤波开启
7-6	RESERVED	-	-	-	预留
5-0	FLT_PR	rw	0	-	FLT_PR：滤波宽度选择 滤波宽度=16×FLT_PR×lsb_clk

9. MCPWM_CON: mcpwm control register

Bit	Name	RW	Default	RV	Description
15-8	TMR_EN	rw	0	-	定时器计数开关控制（tmr7-0） 0：定时器计数关闭 1：定时器计数打开
7-0	PWM_EN	rw	0	-	模块开关控制（pwm7-0） 0：模块关闭 1：模块开启

11.3 使用说明

PWM 输出的占空比 N 的计算公式如下：

N = (PWMCMPx/ (PWM_TMRx_PR+1)) * 100%

PWM 行为：计时器计到 0 时开始翻转为高电平，计时器计数到值 CMP 时翻转为低电平，待计时溢出回归为 0 时又转为高电平，以此往复。所以可以设定好各个 PWM 的定时器 PWM_TMRx_CNT 的初值，然后同时启动各路 PWM_EN 来获得准确的相位差。如 PWM_TMR0_CNT 初值为 0，则 PWM1 和 PWM0 的之间的相位差时间为（PWMTMR1PR-PWM_TMR0_CNT+1）* 时钟(时钟为 TCK1/（2^TCKPS1）)。

第 12 章 UART

12.1 UART 模块说明

UART0 只支持普通 BUF 传输模式，不支持 DMA 和流量控制。

UART1 支持接收带循环 Buffer 的 DMA 模式和普通模式。

UART1 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。

[注意]：

1. OT_PND 触发流程，满足以下步骤，则可产生 OT_PND：
   (1) 写 UTx_OTCNT，数值不为 0 启动 OT 功能；
   (2) 收到 n 个 byte 数据；
   (3) 从最后一个下降沿开始，等待 OT 超时（每来一个下降沿都会重新计时，时钟与接收 baud_clk 一致）；
   (4) 当 OTCNT 与超时时间相等，OT_PND 置 1（只能通过 CLR_OTPND）；
2. 读接收的数据数量时，先将 RDC 该 bit 写 1，接着 asm(“csync”)清空流水线，然后软件查询 RDC_OVER 置 1，置 1 后即可读取 HRXCNT 数值，RDC_OVER 只在 RDC 写 1 之后生效，平常为无效状态（不管 0 或 1）；

12.2 输入时钟结构

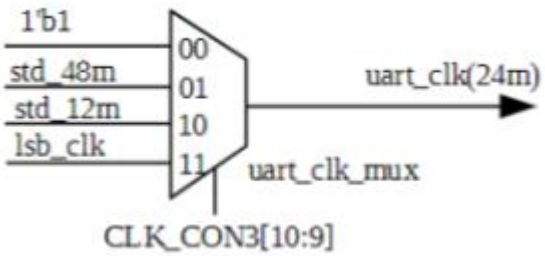


图 1　uart_clk 时钟结构

12.2 数字模块控制寄存器

1. UTx_CON0: uart x confrol register 0

Bit	Name	RW	Default	RV	Description
31-16	RESERVED	-	0	-	预留
15	TPND	r	0	-	The TX pending: 0：without pending 1：with pending
14	RPND	r	0	-	The RX pending & Dma_Wr_Buf_Empty:(数据接收不完 Pending 不会为 1) 0：without pending 1：with pending
13	CLRTPND	w	0	-	Clear TX pending: 0：useless 1：clear pending
12	CLRRPND	w	0	-	Clear RX pending: 0：useless 1：clear pending
11	OTPND	r	0	-	OTPND：Over Time Pending
10	CLR_OTPND	w	0	-	CLR_OTPND：清空 OTPND
9	RESERVED	-	0	-	预留
8	RDC_OVER	r	0	-	用于判断写 RDC 后数据是否已存入内存标志，RDC 写 1 时会清零，当数据存入内存后置 1
7	RDC	w	0	-	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	0	-	RXMODE(*)：读模式选择 0：普通模式，不用 DMA 1：DMA 模式
5	OT_IE	rw	0	-	OTIE:OT 中断允许 0：不允许 1：允许
4	DIVS	rw	0	-	DIVS：前 3 分频选择，0 为 4 分频，1 为 3 分频
3	RXIE	rw	0	1	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	0	1	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	UTRXEN	rw	0	1	UTRXEN：UART 模块接收使能

0	UTTXEN	rw	0	1	UTTXEN：UART 模块发送使能

2. UTx_CON1: uart x control register 1

Bit	Name	RW	Default	RV	Description
15	CTSPND	r	0	0	CTSPND : CTS 中断 pending
14	CLR_CTSPND	w	0	-	CLR_CTSPND：清除 CTS pending
13	CLRRTS	w	0	-	CLRRTS：清除 RTS 0：N/A 1：清空 RTS
12-5	RESERVED	-	0	-	预留
4	RX_DISABLE	r	0	-	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	0	-	CTSIE：CTS 中断使能 0：禁止中断 1：中断允许
2	CTSE	rw	0	-	CTSE：CTS 使能 0：禁止 CTS 硬件流控制 1：允许 CTS 硬件流控制
1	RTS_DMAEN	rw	0	-	RTS_DMAEN：RTS 接收数据流控制使能 0：禁止 1：允许
0	RTSE	rw	0	-	RTSE：RTS 使能 0：禁止 RTS 硬件流控制 1：允许 RTS 硬件流控制 **[注意]**：只有 UART1 有该功能

3. UTx_CON2：uart x control register 2

Bit	Name	RW	Default	RV	Description
15-9	RESERVED	-	0	-	预留
8	CHK_PND	r	0	-	校验中断标志
7	CLR_CHKPND	w	0	-	校验中断 CHK_PND 清除，置 1 清除
6	CHK_IE	rw	0	1	校验中断使能，置 1 使能
5-4	CHECK_MODE	rw	0	-	校验模式选择 0：常 0 1：常 1 2：偶校验

					3：奇校验
3	CHECK_EN	rw	0	0	校验功能使能位，置 1 使能
2	RB9	r	0	0	RB8：9bit 模式时，RX 接收的第 9 位
1	RESERVED	-	0	-	预留
0	M9EN	rw	0	0	M9EN：9bit 模式使能

4. UTx_BAUD：uart x baudrate register

Bit	Name	RW	Default	RV	Description
15-0	UTx_BAUD	w	0x0	0x0	CTSPND : CTS 中断 pending

uart 的 UTx_DIV 的整数部分

串口频率分配器因子(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)

（其中，Freq_sys 是 apb_clk，指慢速设备总线的时钟，非系统时钟）

5. UTx_BUF：uart x data buffer register

Bit	Name	RW	Default	RV	Description
31-0	RESERVED	-	0	-	预留
7-0	UT_BUF	w	0x0	0x0	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

6. UTx_TXADR：uart x TX DMA buffer register

Bit	Name	RW	Default	RV	Description
31-27	RESERVED	-	0	-	预留
26-0	UT_BUF	w	0x0	0x0	DMA 发送数据的起始地址

7. UTx_TXCNT：uart x TX DMA buffer register

Bit	Name	RW	Default	RV	Description
15-0	UTx_TXCNT	w	0x0	0x0	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]).

8. UTx_RXCNT：uart x RX DMA counter register

Bit	Name	RW	Default	RV	Description

31-0	UTx_RXCNT	w	0x0	0x0	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]).

9. UTx_RXSADR ：uart x RX DMA start address register

Bit	Name	RW	Default	RV	Description
31-27	RESERVED	-	0	-	预留
26-0	UTx_RXSADR	w	0x0	0x0	DMA 接收数据时，循环 buffer 的结束地址，需 4byte 地址对齐

10. UTx_RXEADR ：uart x RX DMA end address register

Bit	Name	RW	Default	RV	Description
31-27	RESERVED	-	0	-	预留
26-0	UTx_RXEADR	w	0x0	0x0	DMA 接收数据时，循环 buffer 的结束地址，需 4byte 地址对齐

11. UTx_HRXCNT：uart x have RX DMA counter register

Bit	Name	RW	Default	RV	Description
31-0	UTx_HRXCNT	r	0x0	0x0	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。

12. UTx_OTCNT：uart x Over Timer counter register

Bit	Name	RW	Default	RV	Description
31-0	UTx_OTCNT	w	0x0	0x0	设置串口设备在等待 RX 下降沿的时间，在设置的时间内没收到 RX 下降沿，则产生 OT_PND Time(ot) = Time(rx_baud_clk)*UTx_OTCNT; 例如： 接收波特率时间为 100ns，UTx_OTCNT=10 那么 OT 的时间即为 1000ns

13. UTx_ERR_CNT：uart x error byte counter register

Bit	Name	RW	Default	RV	Description
31-0	UTx_ERR-CNT	r	0x0	-	校验错误数量计数，当打开 CHECK_EN 后，

					当第一次校验出错时，该寄存器会记录当前 byte 对应的数量。 [注意]：只记录第一次出错的数量，清除 CHK_PND 会重新记录。

第 13 章 IO_Mapping_Control

13.1 模块说明

13.1.1 概述

IOMC(IO remapping control)控制寄存器主要用于控制 IO 除 crossbar 之外的一些拓展功能的配置

13.1.2 IO 唤醒

Wakeup 是一个异步事件唤醒模块。它可以将处于 standby/sleep 状态下的系统唤醒，进入正常工作模式。当系统处于正常模式时，则 Wakeup 源作为中断源，Wakeup 模块有 4 个唤醒事件来源，分别对应 4 个 IO 电平变化。

IO 唤醒源有以下 4 个：

事件 0：ich_wkup
事件 1：irflt
事件 2：uart0_rxas
事件 3：uart1_rxas

1. 各引脚功能如下表

引脚	软关机唤醒时 IO 状态	LCD	Others	OSC	AUDIO	PPS/PPM	SARADC	SFC	SPIO
PA0	保持	SEG0			AIN_LDO		ADC0		
PA1	保持	SEG1			AIN_A0				
PA2	保持	SEG2/COM5_B			AIN_A1				
PA3	保持	SEG3/COM4_B				PPM_DAT1			
PA4	保持	SEG4/COM3_B					ADC1		
PA5	保持	SEG5/COM2_B				PPM_DAT0			
PA6	保持	SEG6/COM1_B					ADC2		
PA7	保持	SEG7/COM0_B							
PA8(rtl 上拉)	保持	SEG8	长按 Reset				ADC3		
PA9	保持	SEG9							
PA10	保持	SEG10					ADC4		
PA11	保持	SEG11							
PA12	保持	SEG12							
PA13(rtl 下拉)	保持	SEG13							
PA14(rtl 下拉)	保持	SEG14							

PA15	保持	SEG15							
PB0	保持	SEG16					ADC5		
PB1	保持	SEG17							
PB2	保持	SEG18					ADC6		
PB3	保持	SEG19							
PB4	保持	SEG20					ADC7		
PB5	保持	SEG21		OSCI_12M					
PB6	保持	SEG22		OSCO_12M			ADC8		
PB7	保持	SEG23					ADC9		
PB7	保持	SEG24	LVD				ADC10		
PB9	保持	SEG25		OSCI_32K					
PB10	保持			OSCO_32K		PPS_DAT			
PC0	保持 flashB	SEG26/COM5_A					ADC11	SFC_DO B (0)	SPI D0 B (0)
PC1	保持 flashB	COM4_A					ADC12	SFC_CLK B	SPI0_CLK B
PC2(B 口 FLASH 供电脚)	保持 flashB	COM3_A							
PC3	保持 flashB	COM2_A	SDPG				ADC13	SFC_DI B (1)	SPI0_DI B (1)
PC4(rom 上拉)	保持 flashB	COM1_A	SD 辅助供电 IO		AIN_L			SFC_CS B	SPI0_CS B
PC5	保持	COM0_A			AIN_R				
PD0	保持 flashA	SEG27						SFC_CLK A	SPI0_CLK A
PD1	保持 flashA	SEG28						SFC_DO A (0)	SPI0_D0 A (0)
PD2	保持 flashA	SEG29						SFC_DI A (1)	SPI0_D1 A (1)
PD3(rom 上拉)	保持 flashA	SEG30						SFC_CS A	SPI0_CS A
PD4(A 口 FLASH 供电脚)	保持 flashA	SEG31							
USBDP(rtl 下拉)	预设 2						ADC14		
USBDM(rtl 上拉)	预设 2						ADC15		
VPWR(PP0)	预设 1								

注：1.SPI Flash 的启动顺序时先 A 口后 B 口
2.PD 不支持 crossbar，不支持 PMU IO 功能，不支持 pwm_led；PP0 不支持 pwm_led
3.保持：软关机唤醒时，IO 状态可以保持
4.不保持：软关机唤醒时，IO 状态会变成高阻
5.预设 1：软关机唤醒时，maskrom 根据预先设定，IO 状态可以设置为任意状态
6.预设 2：软关机唤醒时，maskrom 根据预先设定，IO 状态可以设置为上拉、下拉、或高阻
7.保持 flash：软关机唤醒时，maskrom 根据预先设定，3 选 1：1.flash A 口保持；2.flash B 口保持；3.flash A、B 口都保持；

13.2 数字模块控制寄存器

1. IOMC0: IO Mapping Control register0

Bit	Name	RW	Default	RV	Description
31-13	RESERVED	-	-	-	预留
12-10	SPI_ICK_SEL	rw	0	-	SPI1 从机选择 SPI0_CLK 延迟参数设置，默认不配置
9	SPI_DUPLEX	rw	0	-	SPI1 从机输入时钟选择 0：SPI1_CLK 1：SPI0_CLK
8	PPM1_PU_EN	rw	0	-	PPM 的 A 组端口上拉使能
7	PPM0_PU_EN	rw	0	-	PPM 的 B 组端口上拉使能
6	PPM_IOS	rw	0	-	PPM 输入输出端口组选择 0：A 组 1：B 组
5	SLCD_CON_IOS	rw	0	-	SLCD COM 输出端口组选择 0：A 组 1：B 组
4	CAP_MUX_EDGE	rw	0	-	CAPTURE 输出边沿选择 0：上升沿 1：下降沿
3	RESERVED	rw	0	-	保留 0
2	SPI0_IOS	rw	0	-	SPI 固定端口组选择 0：A 组 1：B 组（与 SFC 端口组对应，具体看 IO_Mapping）
1	SPI0_CROSSBAR	rw	0	-	SPI0 IO 模式选择 0：固定 IO 模式 1：crossbar 模式
0	SFC_IOS	rw	0	-	SFC 端口组选择 0：A 组 1：B 组（具体看 io_mapping）

2. OCH_CON0: output channel control register0

Bit	Name	RW	Default	RV	Description
31-30	RESERVED	-	-	-	预留
29-25	OCH5_SEL	rw	0x0	-	通用输出通道选择 0：tmr0_pwm 1：tmr1_pwm 2：tmr2_pwm 3：clk_out0 4：clk_out1 5：clk_out2

					6：clk_out3 7：gp_ich6 8：gp_ich7 9：ledc_out
24-20	OCH4_SEL	rw	0x0	-	通用输出通道选择，同 OCH5_SEL
19-15	OCH3_SEL	rw	0x0	-	通用输出通道选择，同 OCH5_SEL
14-10	OCH2_SEL	rw	0x0	-	通用输出通道选择，同 OCH5_SEL
9-5	OCH1_SEL	rw	0x0	-	通用输出通道选择，同 OCH5_SEL
4-0	OCH0_SEL	rw	0x0	-	通用输出通道选择，同 OCH5_SEL

3. OCH_CON1: output channel control register1

Bit	Name	RW	Default	RV	Description
31-10	RESERVED	-	-	-	预留
9-5	OCH7_SEL	rw	0x0	-	通用输出通道选择，同 OCH5_SEL
4-0	OCH6_SEL	rw	0x0	-	通用输出通道选择，同 OCH5_SEL

4. ICH_CON0: input channel control register0

Bit	Name	RW	Default	RV	Description
31-28	ICH_IRFLT	rw	0x0	-	功能输入通道选择 0：GP_ICH0 1：GP_ICH1 ..... 7：GP_ICH7 8：TMR0_PWM 9：TMR1_PWM
27-24	ICH_WKUP	rw	0x0	-	功能输入通道选择，同上
23-20	ICH_TMR2_CAP	rw	0x0	-	功能输入通道选择，同上
19-16	ICH_TMR2_CIN	rw	0x0	-	功能输入通道选择，同上
15-12	ICH_TMR1_CAP	rw	0x0	-	功能输入通道选择，同上
11-8	ICH_TMR1_CIN	rw	0x0	-	功能输入通道选择，同上
7-4	ICH_TMR0_CAP	rw	0x0	-	功能输入通道选择，同上
3-0	ICH_TMR0_CIN	rw	0x0	-	功能输入通道选择，同上

5. ICH_CON1:input channel control register0

Bit	Name	RW	Default	RV	Description
31-8	RESERVED	-	0x0	-	预留
7-4	ICH_CLK_PIN	rw	0x0	-	功能输入通道选择，同上

3-0	OCH6_SEL	rw	0x0	-	功能输入通道选择，同上

6. WKUP_CON0: wakeup enable control register

Bit	Name	RW	Default	RV	Description
31-4	RESERVED	-	0x0	-	预留
3-0	WKUP_EN	rw	0x0	-	对应唤醒源使能 0：关闭唤醒功能 1：打开唤醒功能

7. WKUP_CON1: wakeup edge control register

Bit	Name	RW	Default	RV	Description
31-4	RESERVED	-	0x0	-	预留
3-0	WKUP_EDGE	rw	0x0	-	对应唤醒源边沿选择 0：上升沿唤醒 1：下降沿唤醒

8. WKUP_CON2: wakeup clear pending control register

Bit	Name	RW	Default	RV	Description
31-4	RESERVED	-	0x0	-	预留
3-0	WKUP_CPND	rw	0x0	-	对应唤醒源中断清除，写 1 清零

9. WKUP_CON3: wakeup pending control register

Bit	Name	RW	Default	RV	Description
31-4	RESERVED	-	0x0	-	预留
3-0	WKUP_PND	rw	0x0	-	对应唤醒源中断标记

第 14 章 LCDC

14.1 模块说明

14.1.1 概述

LCD 模块控制器，主要负责控制推 LCD 屏幕。

最大可以推 6COM & nSEG 的屏幕（SEG 数量参考 IO mapping）；支持 1/2、1/3、1/4 bias，支持强弱驱灵活调整；可以使用 RTC 晶振时钟 32KHz 或内部 LRC（约 32KHz）作为模块时钟。

14.1.2 数字模块控制寄存器

1. LCDC_CON0 : LCD control register 0(32bit addressing)

Bit	Name	RW	Default	RV	Description
31	HD_EN1	rw	0x0	-	High drive EN1，强力强驱电源开关； 0：disable 1：enable
30	HD_EN0	rw	0x0	-	High drive EN0，中等强驱电源开关； 0：disable 1：enable
29-28	HD_ISEL	rw	0x0	-	Bias current selection of LCD buffer，LCD 驱动电流选择： 00：200nA； 01：300nA； 10：400nA； 11：500nA；
28-27	RESERVED	r	0	-	预留
26-17	TEST_FLAG[9:0]	rw	0	-	测试模式通道使能：IO 直接输出 LCD 对应通道 TEST_FLAG[0]：SEGIOoutputLCDVDD TEST_FLAG[1]：SEGIOoutputLCDV1.2 TEST_FLAG[2]：SEGIOoutputLCDV1.1 TEST_FLAG[3]：SEGIOoutputLCDV1.0 TEST_FLAG[4]：SEGIOoutputPD TEST_FLAG[5]：COMIOoutputLCDVDD TEST_FLAG[6]：COMIOoutputLCDV1.2 TEST_FLAG[7]：COMIOoutputLCDV1.1

					TEST_FLAG[8]：COMIOoutputLCDV1.0 TEST_FLAG[9]：COMIOoutputLCDPD
16	TEST_EN	rw	0	-	开启测试模式。IO 直接输出 LCD 静态电平；
15-14	COMCNT	rw	0	-	选择 COM 的数目(Duty)： 00：3COM 01：4COM 10：5COM 11：6COM 1/Duty=1/(COMCNT+3)
13-12	CHGMOD	rw	0	-	充电模式控制 00：一直用弱充电模式 01：一直用强充电模式 10：交替充电模式 A 11：交替充电模式 B 交替充电模式 A 在状态切换时开始转强驱动，经过 CHGDUTY+1 个 32KHz 时钟后撤销强驱动 交替充电模式 B 在状态切换前半个周期(32KHz)时钟后撤销强驱动 【注意】：强充电模式需要与 HD_EN0\1 配合使用，即开启 HD_EN0\1 为 0b01，或 0b10 才能开启强充电模式。
11-8	CHGDUTY	rw	0	-	交替充电模式下强充电占的Cycle数（按32KHz时钟） 0000：1； 0001：2； 0010：3； 0011：4； ..... 1111：16；
7	FF	rw	0	-	帧频率控制 0：FLCD=32KHz/128 1：FLCD=32KHz/64 FLCD 为模块状态切换时钟频率
6-4	VLCDS	rw	0	-	VLCD 控制： 000：2.6V 001：2.7V ..... 110：3.2V 111：3.3V
3-2	BIAS	rw	0	-	BIAS 选择 00：模拟模块工作禁止 01：1/2bias

					10：1/3bias 11：1/4bias
1	DOTEN	rw	0	1	COM0-SEG0 输出一个 1Hz 的信号，可作为“跳秒”脚
0	LCDEN	rw	0	-	LCD enable

2. LCDC_CON1 : LCDC control register 1(32bit addressing).

Bit	Name	RW	Default	RV	Description
31-16	RESERVED	-	-	-	预留
15	FRAME_PND	rw	0	-	帧 pending：断续推屏模式下，完成一帧的推屏后该位置 1
14	FRAME_CPND	rw	0	-	清除帧 pending
13-8	RESERVED		-	-	预留
7-6	CLK_DIV	rw	0	-	模块时钟预分频选择： 00：div1; 01：div2； 10：div4； 11：div8；
5-3	RESERVED	-	-	-	预留
2	LCD_IE	rw	0	-	断续推屏模式下，完成一帧的推屏后，LCD 帧中断使能： 0：不使能 1：使能
1	CTU_EN	rw	0	-	连续推屏模式使能： 0：断续推屏模式（完成一帧会起帧 pending，同时停止推屏） 1：连续推屏模式（完成一帧不会起帧 pending，硬件会自动继续进行下一帧的推屏）
0	SW_KST	w	0	-	LCD 推屏开始位，使能 LCD 模块后，向该位写入 1，开始 LCD 推屏

3. LCDC_CON3 : clock system control register 3(32bit addressing).

Bit	Name	RW	Default	RV	Description
31-14	x	x	x	x	详见 Clock_System 文档
13-11	LCD_CKSEL	rw	0x0	-	选择 LCD 模块的时钟 000：1’b1; 001：LRC 时钟作为 LCD 模块工作时钟； 010：RTOSL 时钟作为 LCD 模块工作时钟； 011：WCLK 晶振时钟作为 LCD 模块工作时钟； 100：LSB 时钟作为 LCD 模块工作时钟；
10-0	x	x	x	x	详见 Clock_System 文档

4. SEG_IO_EN0 : SEG IO enable register 3(32bit addressing).

Bit	Name	RW	Default	RV	Description
31-0	SEG_IO_EN0	rw	0	-	选择 SEGx 线的使能位

5. SEG0_DAT : SEG Data register 0(32bit addressing).

Bit	Name	RW	Default	RV	Description
31-0	SEG0_DAT	rw	0	-	SEGx 线与 COM0 线上的数据位

6. SEG1_DAT : SEG Data register 1(32bit addressing).

Bit	Name	RW	Default	RV	Description
31-0	SEG1_DAT	rw	0	-	SEGx 线与 COM1 线上的数据位

7. SEG2_DAT : SEG Data register 2(32bit addressing).

Bit	Name	RW	Default	RV	Description
31-0	SEG2_DAT	rw	0	-	SEGx 线与 COM2 线上的数据位

8. SEG3_DAT : SEG Data register 3(32bit addressing).

Bit	Name	RW	Default	RV	Description
31-0	SEG3_DAT	rw	0	-	SEGx 线与 COM3 线上的数据位

9. SEG4_DAT : SEG Data register 4(32bit addressing).

Bit	Name	RW	Default	RV	Description
31-0	SEG4_DAT	rw	0	-	SEGx 线与 COM4 线上的数据位

10. SEG5_DAT : SEG Data register 5(32bit addressing).

Bit	Name	RW	Default	RV	Description
31-0	SEG5_DAT	rw	0	-	SEGx 线与 COM5 线上的数据位

第 15 章 SDC

15.1 模块说明

15.1.1 概述

SD 接口是一个标准的遵守 SD 协议的串行通讯接口，在上面传输的数据 1Byte（8bit）为最小单位。

- ✧ SD 接口只支持主机模式；
- ✧ SD 接口支持 1bit data 和 4bit data 模式；
- ✧ 1bit data 模式：串行数据通过一根 DAT 线传输，一个字节需要 8 个 SD 时钟；
- ✧ 4bit data 模式：串行数据通过四根 DAT 线传输，一个字节需要 2 个 SD 时钟；
- ✧ SD 接口支持高速模式和低速模式，最高始终速度可以达到 50MHz；
- ✧ SD 接口支持命令 DMA 和数据 DMA，两个 DMA 是相互独立的；

第 16 章 USB BRIDGE

16.1 模块说明

16.1.1 概述

支持 USB 1.1，支持 bulk、interrupt、iso、control_transfer。

目前 USB 总共有 1+3 对 Endpoint，为 EP0~EP3，其中对应的深度为：

- ✧ EP0：64 字节；
- ✧ EP1 TX/RX：64 字节；
- ✧ EP2 TX/RX：64 字节；
- ✧ EP3 TX/RX：512 字节；

USB I/O 可以做普通 I/O 使用，其中：

	PU 上拉电阻（单位：欧姆）	PD 下拉电阻（单位：欧姆）
USB DM	183K	15K
USB DP	1.55K	15K

第 17 章 RDEC

17.1 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。

在 AD16 中，支持一路 RDEC，RDEC0 除了能当作普通旋转编码器之外，还兼容鼠标滑轮的定时模式。RDEC0 的中断号为 33。

【注意】：支持两种旋转编码方式，第一种为普通编码方式，第二种为鼠标编码方式，其中在鼠标编码方式下仅支持定时方式。

17.2 数字模块控制寄存器

1. RDECx_CON：rdec control register

Bit	Name	RW	Default	RV	Description
15-9	RESERVED	-	-	-	预留
8	RDEC_MODE	rw	0	0	RDEC_MODE 0：普通模式 1：鼠标定时模式
7	PND	r	0	0	PND，只读，写入无效
6	CPND	w	0	0	CPND，只写。写入 1 清除 PND，读出永远为 0
5-2	RDEC_SPND	rw	0x0	0x0	RDEC_SPD，采样速率设置 Tsr=(2^RDEC_SPD)/F_{lsb} F_{lsb} 为低速外设总线频率，软件应当设置 RDEC_SPD 使 Tsr 介于 0.5~2mS 之间
1	RDEC_POL	rw	0	0	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	0	0	RDEC_EN 0：模块关闭 1：模块打开

2. RDECxDAT：rdec control register

Bit	Name	RW	Default	RV	Description
7-0	RDEC_DAT	r	0x0	0x0	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。 在普通模式下：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。PND 起来后，DAT 的值为 1 则表示当前状态为正转，为-1 则表示当前状态为反转。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。 在鼠标滑轮模式的定时模式下，该寄存器应配合 RDEC_SMP 寄存器使用，当采样个数满足 RDEC_SMP 要求时，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，而此时该寄存器里的值表示在这段时间内鼠标滚轮的增减量。 【注意】：清 PND 的动作会更新此寄存器。

3. RDECxSMP：rdec sample register

Bit	Name	RW	Default	RV	Description
7-0	RDEC_SMP	rw	0x0	0x0	RDEC_SMP： 0000：640 0001：2559 0010：5118 0011：7676 0100：10235 0101：12793 0110：15352 0111：17910 1000：64 该寄存器通过设置采样个数，表示在 Tsmp=Tsr*RDEC_SMP 的时间内将旋转编码器的正反转步数记录在 RDEC_DAT 的寄存器中。当采样个数满足设置要求时，PND 将会被置 1。