杰理 AD17N 芯片手册 V1.0

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

2023 年 7 月 6 日

目录

第 1 章 引言

1.1. 编写目的

此说明书主要对杰理 AD17N 芯片介绍。

AD17N 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 14K 普通 Ram + 4K Cache Ram 的内存空间。

1.2. 文档修改日志

版本	日期	描述
1.0	2023 / 7 / 6	首版 CPU 文档
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 32 个中断；有 8 级中断优先级。

2.2. 中断说明

2.2.1. 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
31-4	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

2.2.2. 中断控制寄存器 ICFGx

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

2.2.3. 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
31	ICFG3[31:28]	BASE + 中断号 x4	hsoft3
30	ICFG3[27:24]	BASE + 中断号 x4	hsoft2
29	ICFG3[23:20]	BASE + 中断号 x4	hsoft1
28	ICFG3[19:16]	BASE + 中断号 x4	hsoft0

27	ICFG3[15:12]	BASE + 中断号 x4	
26	ICFG3[11:08]	BASE + 中断号 x4	
25	ICFG3[07:04]	BASE + 中断号 x4	src
24	ICFG3[03:00]	BASE + 中断号 x4	apa
23	ICFG2[31:28]	BASE + 中断号 x4	
22	ICFG2[27:24]	BASE + 中断号 x4	
21	ICFG2[23:20]	BASE + 中断号 x4	
20	ICFG2[19:16]	BASE + 中断号 x4	mcpwm_tmr
19	ICFG2[15:12]	BASE + 中断号 x4	mcpwm_chx
18	ICFG2[11:08]	BASE + 中断号 x4	gpcnt
17	ICFG2[07:04]	BASE + 中断号 x4	lrct
16	ICFG2[03:00]	BASE + 中断号 x4	osa
15	ICFG1[31:28]	BASE + 中断号 x4	gpadc
14	ICFG1[27:24]	BASE + 中断号 x4	port
13	ICFG1[23:20]	BASE + 中断号 x4	pmu_tmr0
12	ICFG1[19:16]	BASE + 中断号 x4	pmu_soft
11	ICFG1[15:12]	BASE + 中断号 x4	iic0
10	ICFG1[11:08]	BASE + 中断号 x4	spi1
09	ICFG1[07:04]	BASE + 中断号 x4	spi0
08	ICFG1[03:00]	BASE + 中断号 x4	uart1
07	ICFG0[31:28]	BASE + 中断号 x4	uart0
06	ICFG0[27:24]	BASE + 中断号 x4	timer2
05	ICFG0[23:20]	BASE + 中断号 x4	timer1
04	ICFG0[19:16]	BASE + 中断号 x4	timer0
03	ICFG0[15:12]	BASE + 中断号 x4	tick_tmr
02	ICFG0[11:08]	BASE + 中断号 x4	
01	ICFG0[07:04]	BASE + 中断号 x4	exception(misalign/watchdog)
00	ICFG0[03:00]	BASE + 中断号 x4	

2.3. MEMORY

Memory 主要有对 FLASH 的管理和内部 RAM 的管理：

从 0x4000000 开始映射内置 FLASH，一共有 64M 字节的地址空间；从 0x3f01000 ~ 0x3f047ff 是系统主要 RAM 的区域。

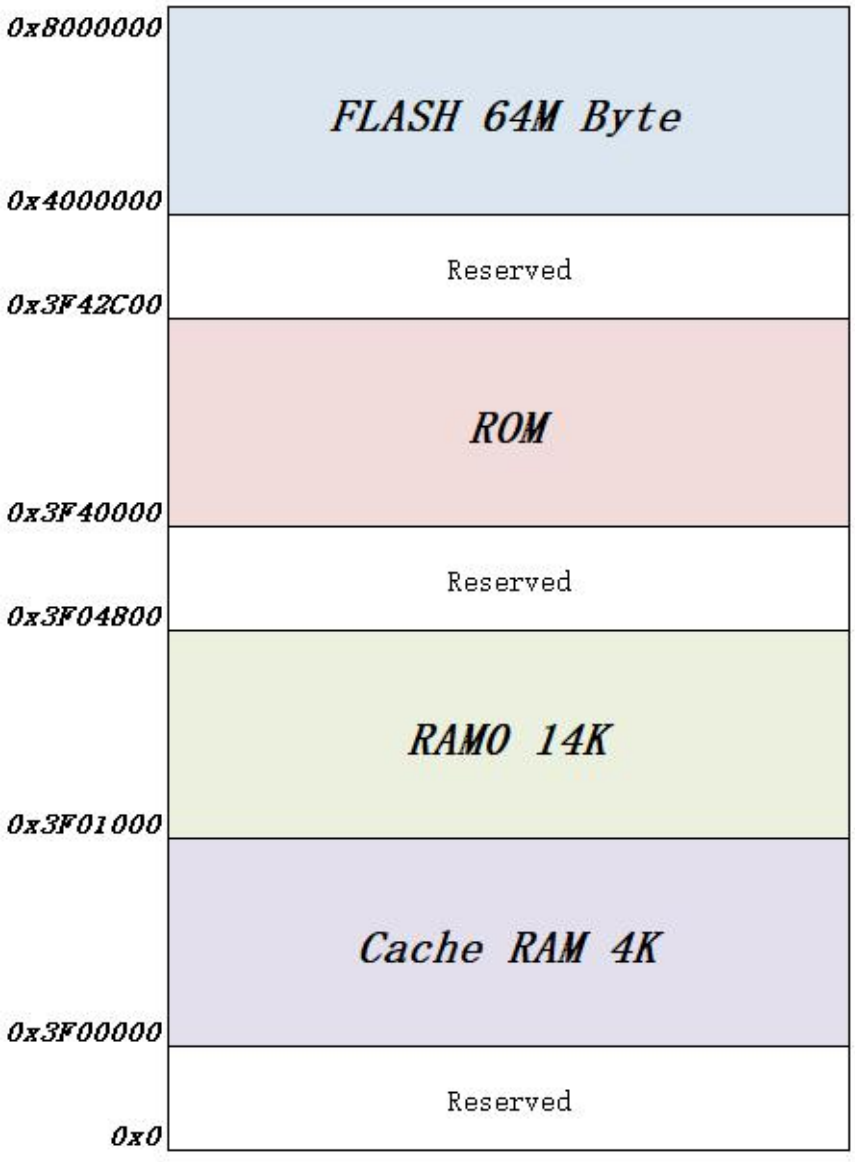


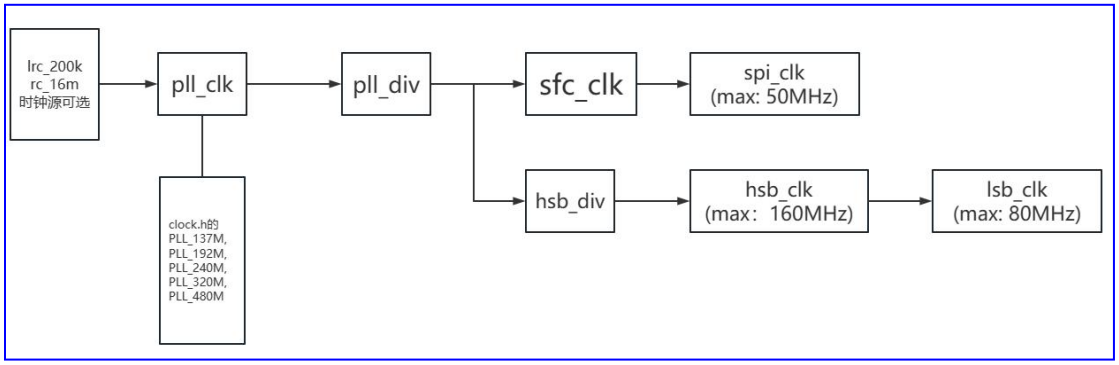
图 2.2 AD17N CPU MAPPING

第 3 章 时钟系统

3.1. 时钟源

1. SH57 具备如下 3 个原生时钟源：
   a) rc_250k：内置 RC 振荡器，振荡频率约 200KHz，用于复位系统和 watch dog 功能时钟。
   b) rc_16m：内置 RC 振荡器，用于系统启动的初始时钟。
   c) lrc_200k：内置低温度电压漂移 RC 振荡器，用于低功耗计数和 PLL 参考时钟。
2. SH57 衍生时钟源：
   a) 来自片内 SYS_PLL 的输出，SYSPLL 工作范围覆盖 240MHz~480MHz，输出 PLL_D1P0，PLL_D1P5，PLL_D2P0，PLL_D2P5，PLL_D3P5 的时钟，如果 SYSPLL 配置为 480MHz，则能输出 480MHz，320MHz，240MHz，192MHz，137.14MHz 的时钟，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。

SH57 最大系统时钟 HSB 不超过 160MHz，LSB 时钟不超过 80MHz。



第 4 章 循环冗余校验(CRC16)

4.1. 模块说明

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： $X^{16}+X^{12}+X^{5}+X^{1}$

4.2. 寄存器 SFR 列表

1. CRC_REG: CRC register

Bit	Name	RW	Description
31:16	-	r	预留
15:0	CRC_REG	rw	写入初始值，CRC 计算完毕，读取校验码

JL_CRC -> REG: CRC16 校验码

2. CRC_FIFO: CRC FIFO register

Bit	Name	RW	Description
31:8	-	r	预留
7:0	CRC_FIFO	w	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. 寄存器 SFR 列表

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

Bit	Name	RW	Description
7	PND	r	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，Reset 值为 0 0：without pending 1：with pending
6	CPND	w	写 1 清除中断标记位
5	WDRMD	rw	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	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	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

Bit	Name	RW	Description
7	-	-	其它功能
6	WDT EXPT EN	rw	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	-	-	其它功能

第 6 章 IIC 模块

6.1. 模块说明

IIC 接口是一个可兼容大部分 IIC 总线标准的串行通讯接口。在上面传输的数据以 Byte 为最小单位，且永远是 MSB 在前。

1. IIC 接口主要支持特性

（1）主模式和从模式，不支持多机功能；

（2）支持标准速度模式（Standard-mode，Sm，高达 100kHz），快速模式（Fast-mode，Fm，高达 400kHz），快速增强模式（Fast-mode Plus，Fm+，高达 1MHz）；

（3）7bit 主从机寻址模式，可配置从机地址应答，广播大致响应；

（4）总线时钟延展功能可配置；

（5）总线数据建立时间和保持时间可配置；

（6）基于便捷的事务操作驱动总线；

（7）数据接收发送各独立具有 1 字节缓冲；

（8）可配置总线信号数字滤波器；

2. 主机

（1）IIC 接口 SCL 时钟由本机产生，提供给片外 IIC 设备使用；

（2）IIC 接口 SCL 时钟可配置，SCL 时钟周期为：

$$T_{IIC\_BUS\_SCL} = T_{IIC\_CLK}/(2 * BUAD\_CNT) + T_{电阻上拉}$$

3. 从机

（1）IIC 接口 SCL 时钟由片外 IIC 设备产生，经内部 IIC_clk 滤波采样后给本机使用；

（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（开启广播地址响应，在接收到广播地址时也会响应），可配置硬件自动回应（ack

位为“0”）；

4. 特殊注意点

时钟要求：BAUD、TSU、THD 值设置应当满足一下关系

BAUD_CNT>(SETUP_CNT+HOLD_CNT)

IO 设置：在选择 IO 作为 IIC 接口的 SCL、SDA 输入输出时，必须配置该两个 IO 的 SPL 寄存器位为 1；

事务寄存器：写入事务后需等待硬件将写入的事务加载后(tx_task_load/rx_task_load 置位即表示对应事务被硬件加载，task_done_pnd 置位即表示事务已被硬件加载且执行完成)才可写入新的事务；

【注意】：IIC 的通信频率不仅与所配置的波特率有关，还会收到 IIC 总线上拉时间的影响，请尽可能符合 IIC 总线硬件电气要求。

6.2. 寄存器 SFR 列表

1. IIC_CON0: iic control register0

Bit	Name	RW	Default	RV	Description
31-11	RESERVED	-	-	-	预留
10	BADDR_RESP_EN	rw	0	-	广播地址响应使能： 1：响应总线上的广播地址； 0：不响应总线上的广播地址；
9	AUTO_SLV_TX	rw	0	-	从机自动发送数据事务使能（主机模式下无效）： 0：不使能 1：使能 从机接收匹配的地址且为从机接收数据时，接下来硬件会自动填充 SEND_DATA 事务操作，使得硬件能不间断的发送数据； （作为从机与不支持时钟延展的主机通信时必填使能，其他工作场景依据需求使能）
8	AUTO_SLV_RX	rw	0	-	从机自动接收数据事务使能（主机模式下无效）： 0：不使能 1：使能 从机接收匹配的地址且为从机接收数据时，接下来硬件会自动填充 RECV_DATA 事务操作，使得在软

					件读取数据时，硬件接口能继续不间断接收数据；（作为从机与不支持时钟延展的主机通信时必填使能，其他工作场景依据需求使能）
7	AUTO_ADR_RESP	rw	0	-	从机接收地址硬件自动响应使能： 作为从机时，在接收到地址数据后（接收的第一字节数据） 0：不自动进行响应，即不对接收到的地址数据进行 NACK/ACK 事务； 1：自动进行响应，接收到匹配的地址（含广播地址）自动响应 ACK；接收到不匹配的地址，自动响应 NACK （从机与不支持时钟延展的主机通信时必须使能，其他工作场景依据需求使能）
6	IGNORE_NACK	rw	0	-	NACK 调试模式使能 0：在发送数据中遇到 NACK 后会停止接收发送 1：在发送数据中遇到 NACK 后不会中断接收发送事务（仅用于调试）
5	NO_STRETCH	rw	0	-	IIC 主从时钟延展功能选择 0：支持时钟延展 1：不支持时钟延展
4-3	FLT_SEL	r	0	-	IIC 接口数字滤波器选择 11：滤除 3*$T_{iic\_baud\_clk}$ 以下尖峰脉宽； 10：滤除 2*$T_{iic\_baud\_clk}$ 以下尖峰脉宽； 01：滤除 $T_{iic\_baud\_clk}$ 以下尖峰脉宽； 00：关闭数字滤波器；
2	SLAVE_MODE	w	0	-	IIC 接口主从机模式： 0：主机模式 1：从机模式
1	I2C_RST	rw	0	-	IIC 接口复位（使能后再释放复位） 0：IIC 接口复位 1：IIC 接口释放
0	EN	rw	0	-	IIC 接口使能 0：关闭 IIC 接口 1：打开 IIC 接口

2. IIC_TASK: IIC task register

Bit	Name	RW	Default	RV	Description
31-13	RESERVED	-	-	-	预留
12	SLAVE_CALL	r	-	-	从机地址匹配:（仅 debug，用户正常流程不可使用）
11	BC_CALL	r	-	-	广播地址匹配:（仅 debug，用户正常流程不可使用）

10	SLAVE_RW	r	-	-	从机读写匹配:(仅 debug,用户正常流程不可使用)
9-6	RUNNING_TASK	r	-	-	正在运行匹配:(仅 debug,用户正常流程不可使用)
5	TASK_RUN_RDY	r	-	-	事务运行状态:(仅 debug,用户正常流程不可使用)
4	TASK_LOAD_RDY	r	-	-	事务加载状态:(仅 debug,用户正常流程不可使用)
3-0	RUN_TASK	w	0x0	-	事务操作:以下 9 种事务操作涵盖了主从机接收发送时的总线行为序列(具体解析见下节) 0x0: SEND_RESET 0x1: SEND_ADDR 0x2: SEND_DATA 0x3: SEND_ACK 0x4: SEND_NACK 0x5: SEND_STOP 0x6: SEND_NACK_STOP 0x7: RECV_DATA 0x8: RECV_DATA_WITH_ACK 0x8: RECV_DATA_WITH_NACK

3. IIC_PND: IIC pending register

Bit	Name	RW	Default	RV	Description
31-29	RESERVED	-	-	-	预留
28	RXTASK_LOAD_PND	r	0	-	硬件将当前的接收相关的 task 事务加载完成
27	TXTASK_LOAD_PND	r	0	-	硬件将当前的发送相关的 task 事务加载完成
26	RXBYTE_DONE_PND	r	0	-	接收 1byte 的数据
25	ADR_MATCH_PND	r	0	-	作为从机接收到与 ADDR 相符的地址(开始广播地址响应使能后,匹配广播地址也会起该 PND)
24	RXNACK_PND	r	0	-	发送地址/数据后接收到 NACK
23	RXACK_PND	r	0	-	发送地址/数据后接收到 ACK
22	STOP_PND	r	0	-	接收到总线 STOP 信号序列
21	RESTART_PND	r	0	-	接收到总线 RESTART 信号序列
20	TASK_PND	r	0	-	事务执行完成 PND
19	RESERVED	-	-	-	预留
18	RXTASK_LOAD_CLR	w	-	-	写“1”清除对应的 PND
17	TXTASK_LOAD_CLR	w	-		写“1”清除对应的 PND
16	RXDATA_DONE_CLK	w	-		写“1”清除对应的 PND
15	ADR_MATCH_CLR	w	-		写“1”清除对应的 PND
14	RXNACK_CLR	w	-		写“1”清除对应的 PND
13	RXACK_CLR	w	-		写“1”清除对应的 PND
12	STOP_CLR	w	-		写“1”清除对应的 PND

11	RESTART_CLR	w	-		写“1”清除对应的 PND
10	TASK_CLR	w	-		写“1”清除对应的 PND
9	RESERVED	-	-	-	预留
8	RXTASK_LOAD_IE	rw	0		对应 PND 的中断使能
7	TXTASK_LOAD_IE	rw	0		对应 PND 的中断使能
6	RXTASK_DONE_IE	rw	0		对应 PND 的中断使能
5	ADR_MATCH_IE	rw	0		对应 PND 的中断使能
4	RXNACK_IE	rw	0		对应 PND 的中断使能
3	RXACK_IE	rw	0	-	对应 PND 的中断使能
2	STOP_IE	rw	0	-	对应 PND 的中断使能
1	RESTART_IE	rw	0	-	对应 PND 的中断使能
0	TASK_IE	rw	0	-	对应 PND 的中断使能

4. IIC_TXBUF: IIC tx buff register

Bit	Name	RW	Default	RV	Description
31-8	RESERVED	-	-	-	预留
7-0	TX_BUFF	rw	-	-	待发送数据

5. IIC_RXBUF: IIC rx buff register

Bit	Name	RW	Default	RV	Description
31-8	RESERVED	-	-	-	预留
7-0	RX_BUFF	r	-	-	已接收数据

6. IIC_ADDR: IIC device address register

Bit	Name	RW	Default	RV	Description
31-8	RESERVED	-	-	-	预留
7-1	IIC DEVICE ADDR	rw	0x00	-	IIC 设备地址（仅支持 7bit 模式）
0	W/R OPTION	rw	0x0	-	作为主机发送地址时的读写标志： 0：发送数据的从机； 1：读取从机的数据；

7. IIC_BAUD: IIC baud clk register

Bit	Name	RW	Default	RV	Description
31-12	RESERVED	-	-	-	预留
11-0	BAUD_CNT	rw	-	-	IIC 总线传输时钟 SCL 速率配置： $F_{IIC\_BUS\_SCL} = F_{IIC\_CLK} / BUAD\_CNT$ 注意：不可设置为 0

8. IIC_TSU: IIC setup time register

Bit	Name	RW	Default	RV	Description
31-7	RESERVED	-	-	-	预留
6-0	SETUP_CNT	rw	-	-	IIC 总线 SDA 信号更新后到 SCL 时钟上升的最少时间配置： $T_{setup} = T_{IIC\_CLK} * SETUP\_CNT$ 无特殊情况软件配置为 2；

9. IIC_THD: IIC hold time register

Bit	Name	RW	Default	RV	Description
31-8	RESERVED	-	-	-	预留
7-0	TX_BUFF	rw	-	-	IIC 总线 SCL 时钟下降到 SDA 信号更新去前的最少时间配置： $T_{hold} = T_{IIC\_CLK} * HOLD\_CNT$ 无特殊情况软件配置为 2；

10. IIC_DBG: IIC debug register

Bit	Name	RW	Default	RV	Description
31-16	SDA_DBG	r	-	-	SDA 相关调试信号
15-0	SCL_DBG	r	-	-	SCL 相关调试信号

6.3. 基本事务操作

IIC_TASK 事务寄存器允许写入以下 9 种事务操作值来驱动 IIC 接口做出相应总线行为序列：

1.0x0：SEND_RESET

工作在主机模式下，向 IIC 总线发送 START 和 STOP；

2.0x1：SEND_ADDR

工作在主机模式下，向 IIC 总线发送 START 和 ADDR 的数据，并接收从机回复的 ACK/NACK；

3.0x2：SEND_DATA

工作在主机模式下，向 IIC 总线发送 TX BUFF 的数据，并接收从机回复的 ACK/NACK；

4.0x3：SEND_ACK

工作在主机/从机模式下，向 IIC 总线发送 ACK；

5.0x4：SEND_NACK

工作在主机/从机模式下，向 IIC 总线发送 NACK；

6.0x5：SEND_STOP

工作在主机/从机模式下，向 IIC 总线发送 STOP；

7.0x6：SEND_NACK_STOP

工作在主机/从机模式下，向 IIC 总线发送 NACK 和 STOP；

8.0x7：RECV_DATA

工作在主机/从机模式下，从 IIC 总线上接收一个字节数据，但不进行 ACK/NACK（持续拉低 SCL 线，总线上的设备需支持时钟延展），知道 SEND_ACK/SEND_NACK 的事务被填充 SCL 才会撤销拉低；

9.0x8：RECV_DATA_WITH_ACK

工作在主机/从机模式下，从 IIC 总线上接收一个字节数据，且随后向 IIC 总线发送 ACK；

10.0x9：RECV_DATA_WITH_NACK

工作在主机/从机模式下，从 IIC 总线上接收一个字节数据，且随后向 IIC 总线发送 NACK；

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。

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。

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 1-65535Byte。写 SPI_CNT 的动作将启动一次 DMA 传输。

7.2. 寄存器 SFR 列表

1. SPIx_CON: SPIx control register 0

Bit	Name	RW	Default	RV	Description
31-22	RESERVED	-	-	-	预留
21	OF_PND	r	0	-	dma fifo overflow，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 发生溢出就会产生 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	-	of_pnd 使能位
18	UF_PND	r	0	1	dma fifo overflow，当 spi 做从机 dma 接收时，fifo 空且 dma 响应慢，而新接收的数据到来时导致 fifo 发生溢出就会产生 of_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	预留
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	0x0	0x0	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

9-8	RESERVED	-	-	-	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：SPICS 空闲时为 0 电平 1：SPICS 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	SPIEN	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 baud rate setting register

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

Bit	Name	RW	Default	RV	Description
7-0	SPI_BUF	rw	0x0	0x0	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. SPIx_ADR: SPI DMA start address register

Bit	Name	RW	Default	RV	Description
7-0	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: SPI DMA counter register

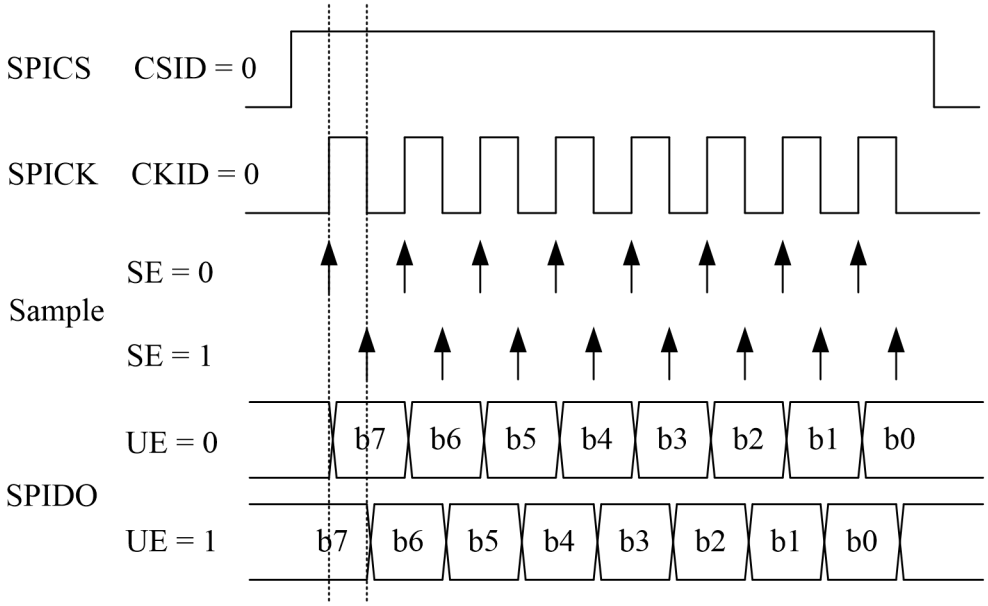
Bit	Name	RW	Default	RV	Description
31-0	SPI_CNT	rw	0x0	0x0	SPI DMA 计数寄存器，读出为当下剩余读写数量， 此寄存器用于设置 DMA 操作的数目（按 Byte 计） 并启动 DMA 传输 如，需启动一次 512Byte 的 DMA 传输，写入 0x0200， 此写入动作将启动本次传输。

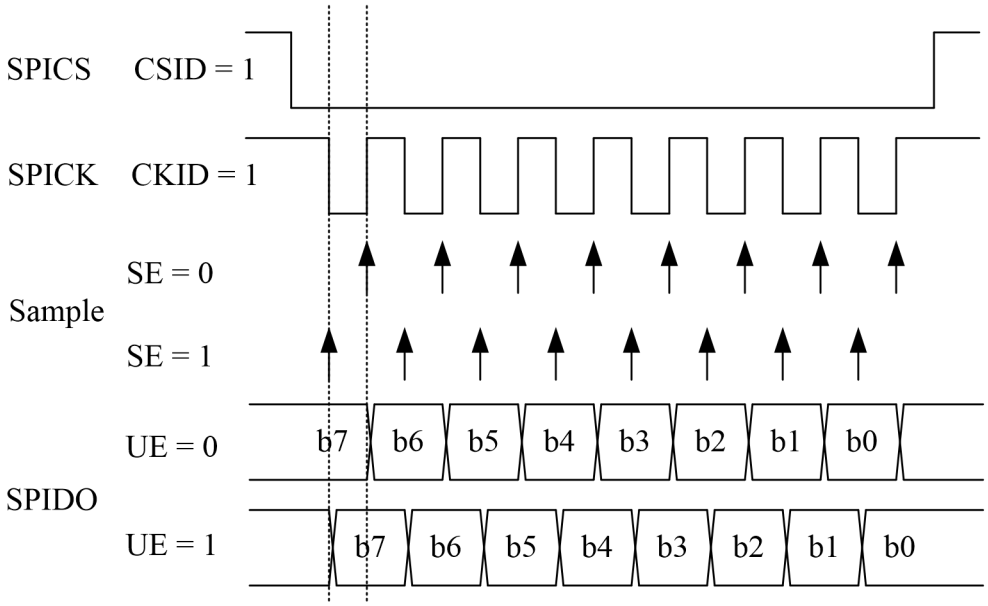
6. SPIx_CON1:SPIx control register

Bit	Name	RW	Default	RV	Description
7-0	SPI_BAUD	-	-	-	SPI 主机时钟设置寄存器 SPICK=system clock/(SPIBAUD + 1)
2	MIX_MODE	rw	0	-	3wire 模式混合接线
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]

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。

传输波形图：




SPICS
CSID = 1
SPICK
CKID = 1
Sample
SE = 0
SE = 1
SPIDO
UE = 0
UE = 1
b7
b6
b5
b4
b3
b2
b1
b0
b7
b6
b5
b4
b3
b2
b1
b0

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。可通过 CPU 访问，逐次读取各个通道电压值。

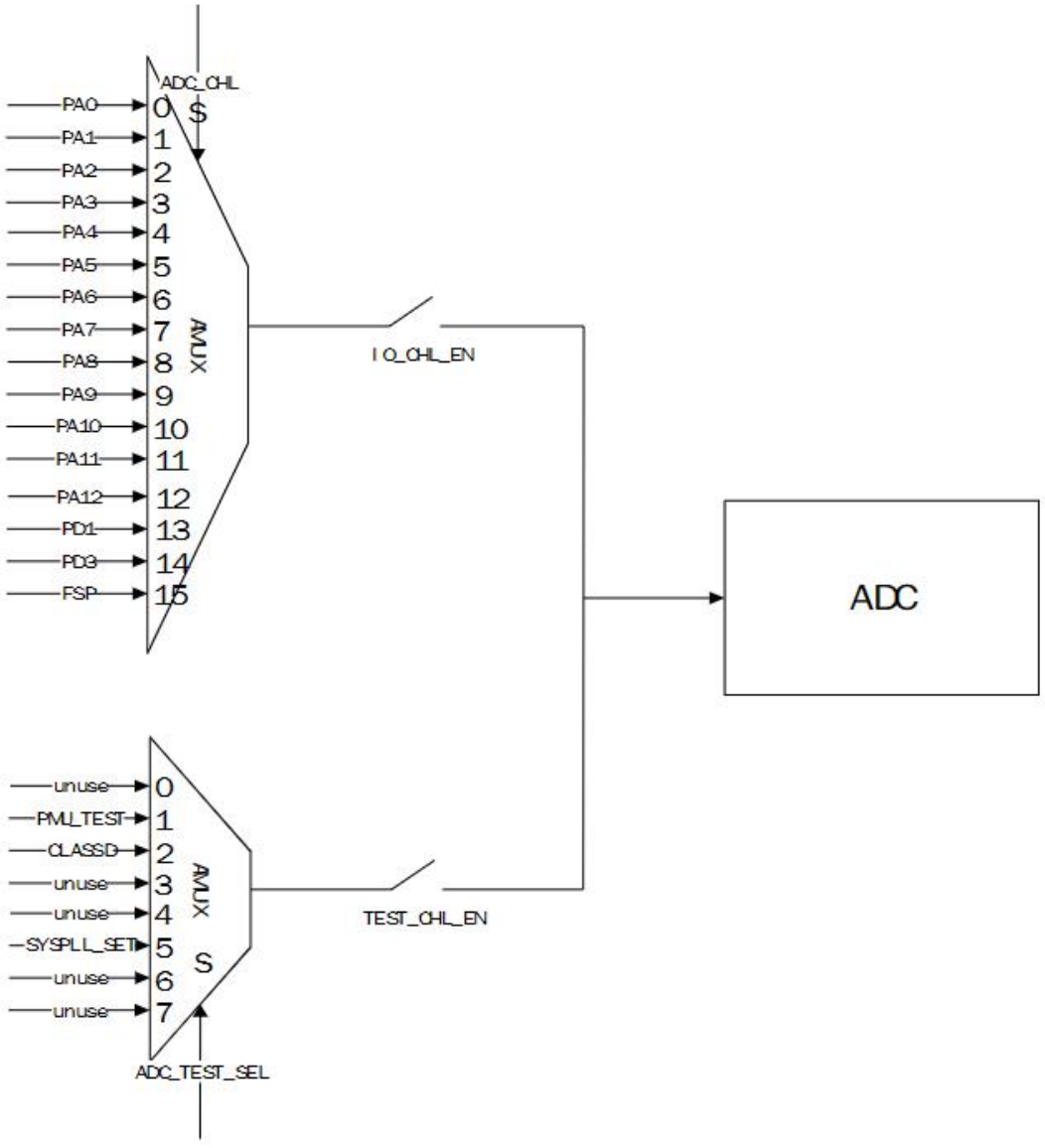


图 8.1 SARADC 通道选择结构图

8.2. 寄存器 SFR 列表

1. ADC_CON: ADC control register

数模转换器(ADC)

Bit	Name	RW	Default	RV	Description
31	ADC_DEN	rw	0x0	1	ADC 控制逻辑使能为，正常工作需要设置为 1
30-27	reserved	r	0x0	0	预留
26	ADC_ISEL	rw	0x0	0	ADC 电流配置参数
25-23	ADC_TESTSEL	rw	0	0	CPU 访问模式下，内部测试信号选择
22-19	ADC_CHL	rw	0	0	CPU 访问模式下，IO 输入通道选择
18-15	WAIT_TIME	rw	0	-	启动延时控制，实际启动延时为 N*8 个 ADC_CLK
14-8	ADC_BAUD	rw	0	-	ADC_BAUD：0 时 ADC_CLK = LSB_CLK 否则 ADC_CLK=LSB_CLK/(ADC_BAUD*2)
7	PND	r	1	-	PND：CPU 读取通道电压完成标志位
6	CPND	rw	x	-	CPND：写 1 启动一次转换，同时 PND 标志清理 0
5	RESERVED	r	0x0	0	预留
4	DMA_CHL_CTL	rw	0	0	DMA 模式下 ADC 输入信号通道选择 0：输入信号选择 IO 通道 1：输入信号选择内部测试信号
3	TEST_CHL_EN	rw	0x0	0	CPU 访问模式下选择内部测试信号
2	IO_CHL_EN	rw	0	1	CPU 访问模式下 IO 通道使能位
1	IE	rw	0	1	CPU 访问模式下 PND 产生中断使能位
0	ADC_AEN	rw	0x0	0	1：ADC 模拟使能常开 0：ADC 模拟使能由硬件根据需要自动开关

2. ADC_RES: ADC result register

Bit	Name	RW	Default	RV	Description
31	RESERVED	-	-	-	预留
9-0	RES	r	x	-	CPU 访问 ADC 结果

3. P3_ANA_CON4 PMU2ADC control SFR

Bit	Name	RW	Default	RV	Description
7	LVDBG_TOADC_EN	rw	0	0	LVDBG output to ADC enable
6	WVBG_TOADC_EN	rw	0	0	WVBG output to ADC enable
5	MDBG_TOADC_EN	rw	0	0	MVBG output to ADC enable
4	VBG_BUFFER_EN	rw	0	0	VBG buffer enable bit
3-1	CHANNEL_ S2-0	rw	0	0	CHANNEL_ADC_S2-0: select channel

					to ADC 000(default): VBG 001：VDC12 010：DVDD 011：VTEMP 100：WVDD 101：VDDIO 110：1/4 VBAT 111：RESERVED
0	PMU_DET_OE	rw	0	0	PMU voltage detect to ADC output enable

第 9 章 时钟脉冲计数器(GPCNT)

9.1. 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，即用一个已知时钟（主时钟）计算另一个时钟（次时钟）的周期。

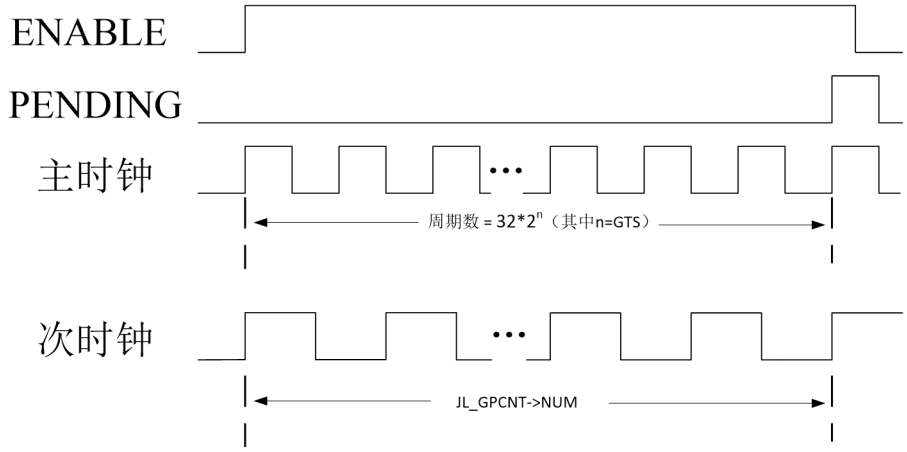


图 1 GPCNT 时钟计算示意图

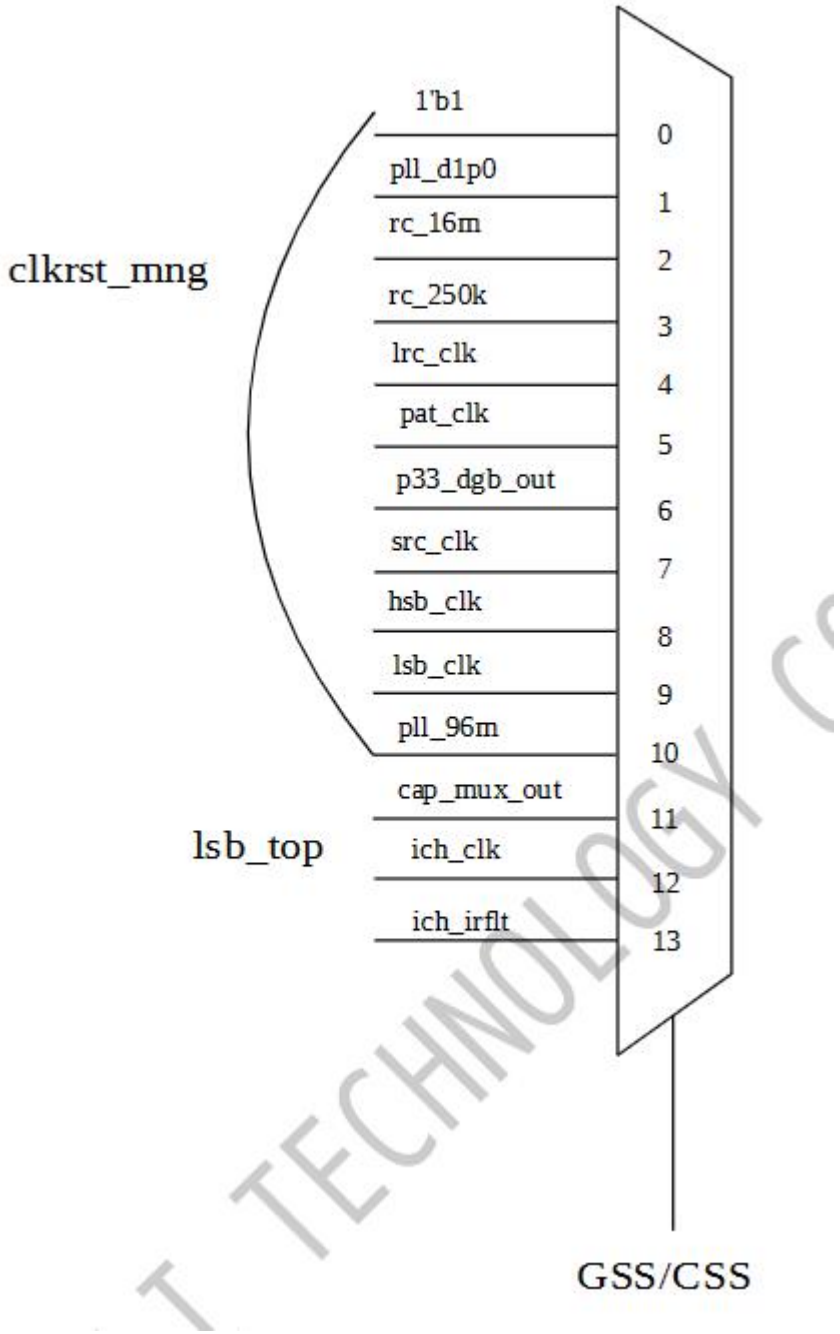


图 2 GPCNT 时钟源选择示意图

9.2. 寄存器 SFR 列表

1. JL_GPCNT->CON: GPCNT configuration register

Bit	Name	RW	Default	RV	Description
31	PND	r	0	-	中断请求标志（当 JL_GPCNT->CON[0]置 1 后，主时钟达到 JL_GPCNT->[11:8]设定

					的周期数后，PENDING 置 1，并请求中断）： 0：无 PENDING 1：有 PENDING
30	CLR_PND	w	x	-	清除中断请求标志位 0：无效 1：清 PENDING
29-20	RESERVED	-	-	-	预留
19-16	GTS	rw	0	-	主时钟周期数选择： 主时钟周期数 = 32*2^n （其中 n=GTS）
15-13	RESERVED	-	-	-	预留
12-8	GSS	rw	0x0	-	主时钟选择（计数时钟）：见图 2
7-6	RESERVED	-	-	-	预留
5-1	CSS	r	1	-	次时钟选择（被计数时钟）：见图 1-2
0	ENABLE	rw	0	-	GPCNT 模块使能位

（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	RV	Description
31	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。

第 10 章 16 位定时器(Timer16)

10.1. 模块说明

Timer16 是一个集合了定时/计数/捕获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。

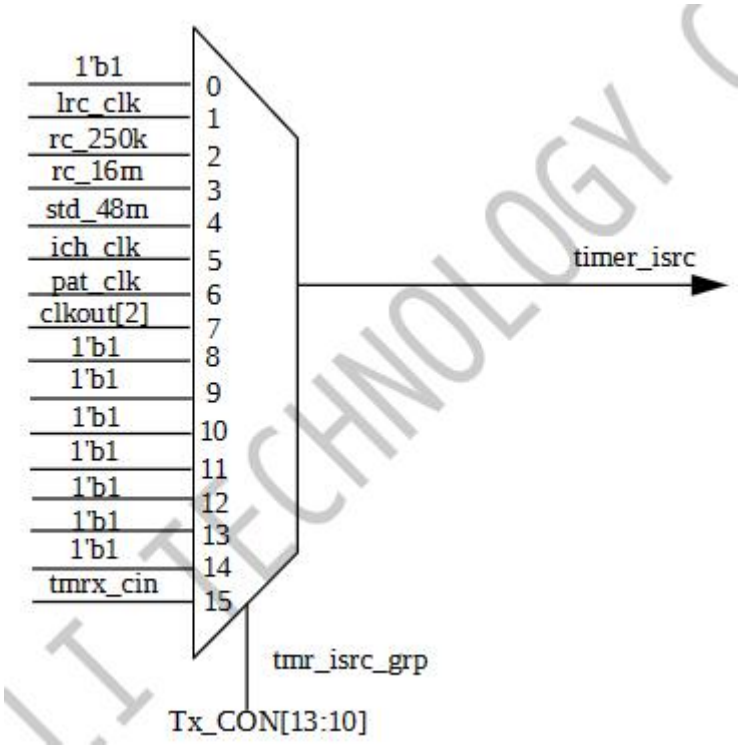


图 1 TIMER 时钟源选择示意图

10.2. 寄存器 SFR 列表

1. JL_TIMERx->CON: timer x configuration 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

16 位定时器(Timer16)

15	PND	r-	0	-	PND：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	-	PCLR：软件在此位写入'1'将清除 PND 中断请求标志
13-10	SSEL	rw	0x0	-	timer 驱动源选择见图 1 【注意】：选择的时钟需要要使用 PSET 分频到(lsb_clk/2)以下！
9	PWM_INV	rw	0	-	PWM_INV：PWM 信号输出反向
8	PWM_EN	rw	0	-	PWM_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	-	PSET：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) / fix io (timer0) 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 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. JL_TIMERx->CNT: timer x counter register

Bit	Name	RW	Default	RV	Description

31-16	RESERVED	-	-	-	预留
15-0	Tx_CNT	rw	x	-	Timer16 的计数寄存器

3. JL_TIMERx->PRD: timer x period register

Bit	Name	RW	Default	RV	Description
31-16	RESERVED	-	-	-	预留
15-0	Tx_PR	rw	x	-	Timer16 的周期寄存器

在定时/计数模式下，当 TIMERx_CNT == TIMERx_PRD 时，TMRx_CNT 会被清 0。在上升沿/下降沿捕获模式下，TIMERx_PRD 是作为捕获寄存器使用的，当捕获发生时，TIMERx_CNT 的值会被复制到 TIMERx_PRD 中。而此时 TIMERx_CNT 自由的由 0-65535-0 计数，不会和 TIMERx_PRD 进行比较清 0

4. JL_TIMERx->PWM: timer x PWM register

Bit	Name	RW	Default	RV	Description
31-16	RESERVED	-	-	-	预留
15-0	Tx_CNT	rw	x	-	Timer16 的 PWM 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%

此寄存器带有缓冲，写此寄存器的动作不会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 11 章 红外滤波模块(IRFLT)

11.1. 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 寄存器的配置，可以将 IRFLT 插入到系统 3 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。

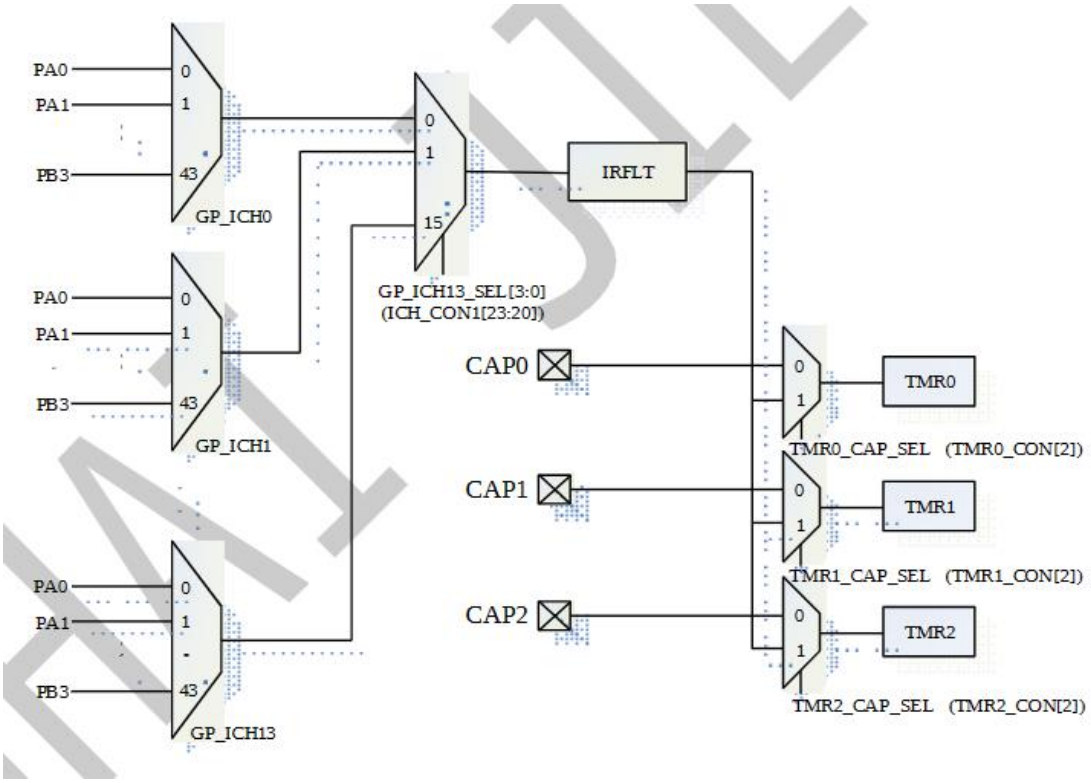


图 1 IRFLT 模块示意图

11.2. 寄存器 SFR 列表

1. JL_IRFLT->CON: irda filter configuration register

Bit	Name	RW	Description
7:4	PSEL	rw	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	时基发生器驱动源选择，见图 2
1	-	r	预留
0	IRFLT_EN	rw	IRFLT 使能 0: 关闭 IRFLT 1: 打开 IRFLT

11.3. 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 24MHz 的系统时钟，并且分频倍数为 512 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。

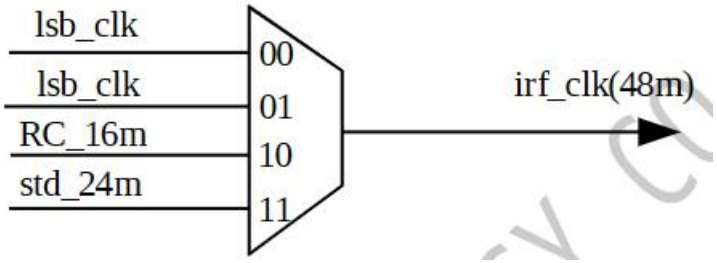


图 2 irflt_clk 时钟结构

第 12 章 MCPWM

12.1. 模块说明

MCPWM 功能块包括：

7 个 timer 时基模块，7 对独立的 PWM 通道，7 路故障保护输入。框图如下：

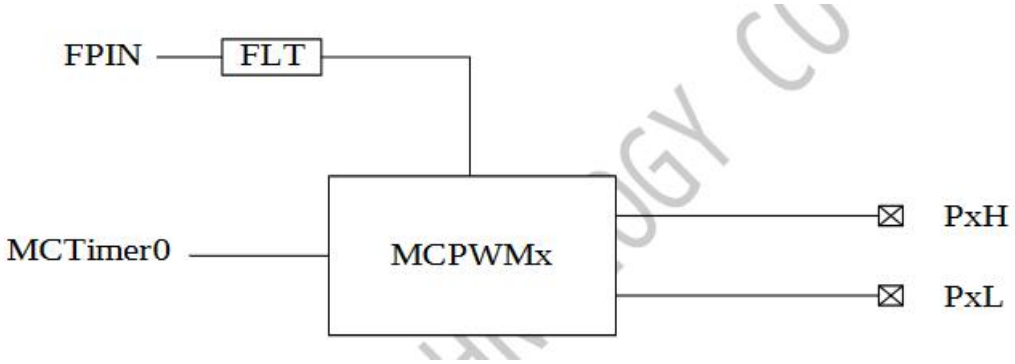


图 1 PWM 模块示意图

12.1.1. 定时器 MCTIMER

mctimer 是 16 位定时器，可作为 pwmx 的时基控制

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

12.1.2. MCPWM 模块引脚

PWM0: PWMCHXH、PWMCHXL

PWM1: PWMCHXH、PWMCH1L

PWM2: PWMCHXH、PWMCH2L

.....

故障输入引脚：FPIN

12.1.3. MCPWM 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

12.2. 寄存器 SFR 列表

1. TMRx_CON: Timer contrl 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
14	RESERVED	-	-	-	预留
13	UFPND	w	x	-	递减借位标志 0：没溢出 1：已发生溢出
12	OFPN	rw	0x0	-	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：消除 UFPND
10	OFCLR	w	-	1	清除 OFPND 标志位 0：无效 1：消除 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：禁止 1：允许
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：递减模式

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. TMR_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 control 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（0-5 为 io 输入，6-7 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

第 13 章 UART

13.1. UART0 模块说明

UART0 只支持单 byte 传输模式，不支持 DMA 传输模式。不支持小数分频。

13.2. UART0 寄存器 SFR 列表

1. UART0_CON0: UART0 control register 0

Bit	Name	RW	Default	RV	Description
15	TPND	r	0	-	TX pending:
14	RPND	r	0	-	RX pending & Dma_Wr_Buf_Empty:(数据接收不完 Pending 不会为 1)
13	CLRTPND	w	0	-	Clear TX pending: 0：useless 1：clear pending
12	CLRRPND	w	0	-	Clear RX pending: 0：useless 1：clear pending
11-5	RESERVED	-	-	-	预留
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	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	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	UART 模块接收使能
0	UTTXEN	rw	0	1	UART 模块发送使能

2. UART0_CON1: UART0 control register 1

Bit	Name	RW	Default	RV	Description
15-10	RESERVED	-	0	-	预留
9	TX_INV	r	0	0	Tx 发送数据电平取反 0：不取反 1：取反
8	RX_INV	rw	0	0	Rx 接收数据电平取反

					0：不取反 1：取反
7-6	RESERVED	-	-	-	预留
5	RX_BYPASS	rw	1	1	Rx 接收数据通路直通使能 0：滤波输入 1：直通输入
4	RX_DISABLE	r	1	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）
3-0	RESERVED	-	-	-	预留

3. UART0_CON2: UART0 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	RB8	r	0	1	RB8：9bit 模式时，RX 模式接收的第 9 位
1	RESERVED	-	-	-	预留
0	M9EN	rw	0	0	9bit 模式使能

4. UART0_BAUD: UART0 baud register

Bit	Name	RW	Default	RV	Description
15-9	UTx_BAUD	w	0x0	0x0	注释如下所示

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 / ((**UARTx_BAUD**+1) *4 + BAUD_FRAC)

当 **DIVS**=1 时，

Baudrate = Freq_syst / ((**UARTx_BAUD**+1) *3 + BAUD_FRAC)

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

5. UART0_BUF: UARTx data buffer register

Bit	Name	RW	Default	RV	Description
31-8	RESERVED	-	0	-	预留
7-0	UT_BUF	r	0	-	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

13.3. UART1 模块说明

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，UT1_RXSADR 寄存器保存它的起始地址，UT1_RXEADR 寄存器保存它的结束地址。同时，在 DMA 接收过程中，会有一个超时计数器（UT1_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，满足以下顺序，则可产生 OT_PND

（1）写 UT1_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 清零）

2. 读接收的数据量时，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）；

13.4. UART1 寄存器 SFR 列表

1. UART1_CON(0): UART1 control 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. UART1_CON1: UART1 control register 0

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
12-10	RESERVED	-	0	-	预留
9	TX_INV	rw	0	0	Tx 发送数据电平取反 0：不取反 1：取反
8	RX_INV	rw	0	0	Rx 接收数据电平取反 0：不取反 1：取反
7	CTS_INV	rw	0	-	CTS 流控有效电平选择（对方允许我方发送数据） 0：高电平有效 1：低电平有效
6	RTS_INV	rw	0	-	RTS 流控有效电平选择（允许对方发送数据） 0：高电平有效 1：低电平有效
5	RX_BYPASS	rw	0	1	Rx 接收数据通路直流使能 0：滤波输入 1：直通输入
4	RX_DISABLE	r	0	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. UART1_CON2: UART1 control register 0.

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	RB8	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. UART1_BAUD: UARTx baud register (16bit addressing, Write Only)

Bit	Name	RW	Default	RV	Description
15-0	UTx_BAUD	w	0x0	0x0	注释如下所示

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 / ((**UARTx_BAUD**+1) *4 + BAUD_FRAC)

当 **DIVS**=1 时，

Baudrate = Freq_syst / ((**UARTx_BAUD**+1) *3 + BAUD_FRAC)

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

5. UART1_BUF: UARTx data buffer regitster

Bit	Name	RW	Default	RV	Description
31-8	RESERVED	-	0	-	预留
7-0	UT_BUF	rw	0	-	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. UART1_TXADR: UARTx TX DMA buffer register

Bit	Name	RW	Default	RV	Description
31-25	RESERVED	-	0	-	预留
24-0	UTx_TXADR	w	0x0	0x0	DMA 发送数据的起始地址

7. UART1_TXCNT: uart x TX DMA count register

Bit	Name	RW	Default	RV	Description
31-0	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_CON0[15]）。

8. UART1_RXCNT: uart x RX DMA count

Bit	Name	RW	Default	RV	Description
31-0	UTx_RXCNT	w	0	-	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_CON0[14]）。

9. UART1_RXSADR: uart x RX DMA start address regitster

Bit	Name	RW	Default	RV	Description
31-25	RESERVED	-	0	-	预留
24-0	UTx_TXADR	w	0x0	0x0	DMA 接收数据时，循环 buffer 的起始地址，需 4byte 地址对齐。

10. UART1_RXEADR: uart x RX DMA end address register

Bit	Name	RW	Default	RV	Description
31-25	RESERVED	-	0	-	预留
24-0	UTx_RXEADR	w	0x0	0x0	DMA 接收数据时，循环 buffer 的结束地址，需要 4byte 地址对齐。 【注意】：UTx_RXSADR = UTx_RXEADR 时没有循环 buffer，接收地址会递增

11. UART1_HRXCNT: uart x have RX DMA count register

Bit	Name	RW	Default	RV	Description
31-0	UTx_HRXCNT	r	0x0	0x0	当设 RDC（UTx_CON0[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. UART1_OTCNT: uart x Over Time count 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. UART1_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 14 章 IO_Mapping_Control

14.1. 模块说明

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。

14.2. 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 17 个唤醒事件来源，分别对应 17 个 IO 电平变化。

IO 唤醒源有以下 17 个：

事件 0：PA0

事件 1：PA1

事件 2：PA2

事件 3：PA3

事件 4：PA4

事件 5：PA5

事件 6：PA6

事件 7：PA7

事件 8：PA8

事件 9：PA9

事件 10：PA10

事件 11：PA11

事件 12：PA12

事件 13：PB0

事件 14：PB1

事件 15：PB2

事件 16：PB3

14.3. 寄存器 SFR 列表

1. JL_IOMC->IOMC0:IO Mapping Control register0

Bit	Name	RW	Default	RV	Description
31-13	RESERVED	-	-	-	预留
12-10	SPI_ICK_SEL	rw	0	-	SPI 输入时钟延迟级别数选择
9	SPI_DUPLEX	rw	0	-	spi1 从机输入时钟选择 0：spi1_clki 1：spi1_clko
8-5	RESERVED	-	0	-	
4	CAP_MUX_EDGE	rw	0	-	CAPTURE 输出边沿选择 0：上升沿 1：下降沿
3	RESERVED	-	-	-	预留
2	SPI0_IOS	rw	0	-	当 SPI0 使用固定的 IO 模式时的 IO 选择 0：使用 A 组 IO（对应 SFC 的 A 组） 1：使用 B 组 IO
1	SPI0_IO_MODE	rw	0	-	SPI0 IO 模式选择（优先级高于 SPI0_IOS） 0：固定 IO 模式 1：crossbar 模式
0	SFC_IOS	rw	0	-	SFC 模块 IO 选择 0：使用 A 组 IO 1：使用 B 组 IO

2. OCH_CON0:outputchannel Control register0

Bit	Name	RW	Default	RV	Description
31-26	RESERVED	-	-	-	预留
25	APA_OCH1_REG	rw	0x0	-	
24	APA_OCH0_REG	rw	0x0	-	
23-20	APA_OCH1_SEL	rw	0x0	-	APA_通用输出通道选择 0-13 与 APA_OCH0 一致 14：apa_och1_reg
19-16	APA_OCH0_SEL	rw	0x0	-	APA0 通用输出通道选择 0：tmr_pwm1

					1：tmr_pwm2 2：uart1_rts 3：clk_out0 4：clk_out1 5：clk_out2 6：gp_ich4 7：gp_ich5 8：gp_ich2 9：gp_ich3 10：uart0_tx 11：uart1_tx 12：mc_pwm0_h 13：mc_pwm0_l 14：apa_och0_reg
15-14	OCH7_SEL	rw	0x0	-	通用输出通道选择 0：gp_ich5
13-12	OCH6_SEL	rw	0x0	-	通用输出通道选择 0：gp_ich4
11-10	OCH5_SEL	rw	0x0	-	通用输出通道选择 0：gp_ich2
9-8	OCH4_SEL	rw	0x0		通用输出通道选择 0：clk_out2
7-6	OCH3_SEL	rw	0x0		通用输出通道选择 0：clk_out1
5-4	OCH2_SEL	rw	0x0		通用输出通道选择 0：clk_out0
3-2	OCH1_SEL	rw	0x0		通用输出通道选择 0：tmr2_pwm
1-0	OCH0_SEL	rw	0x0		通用输出通道选择 0：tmr1_pwm

3. ICH_CON0:inputputchannel Control register0

Bit	Name	RW	Default	RV	Description
31-28	ICH_IFRLT	rw	0x0	-	功能输入通道选择 0：GP_ICH0 1：GP_ICH1 2：GP_ICH2 3：GP_ICH3 4：GP_ICH4 5：GP_ICH5 6：GP_ICH6

					7：GP_ICH7 10：TMR1_PWM 11：TMR2_PWM
27-24	ICH_CAP	rw	0x0	-	功能输入通道选择，同 ICH_IFRLT
23-20	ICH_CLK	rw	0x0	-	（外部输入数字时钟）功能输入通道选择，同 ICH_IFRLT
19-16	ICH_UART1_CTS	rw	0x0	-	功能输入通道选择，同 ICH_IFRLT
15-12	ICH_TMR1_CAP	rw	0x0	-	功能输入通道选择，同 ICH_IFRLT
11-8	ICH_TMR2_CIN	rw	0x0	-	功能输入通道选择，同 ICH_IFRLT
7-4	ICH_TMR1_CAP	rw	0x0	-	功能输入通道选择，同 ICH_IFRLT
3-0	ICH_TMR1_CIN	rw	0x0	-	功能输入通道选择，同 ICH_IFRLT

4. ICH_CON1:inputchannel Control register1

Bit	Name	RW	Default	RV	Description
31-12	RESERVED	-	-	-	预留
11-8	MC_PWM_FP[1]	rw	0x0	-	功能输入通道选择，同 ICH_IFRLT
7-4	MC_PWM_FP[0]	rw	0x0	-	功能输入通道选择，同 ICH_IFRLT
3-0	ICH_CLK_MUX	rw	0x0	-	（外部输入数字时钟）功能输入通道选择，同 ICH_IFRLT

5. WKUP_CON0: wakeup enable control register

Bit	Name	RW	Default	RV	Description
31-17	RESERVED	-	0x0	-	预留
16-0	WKUP_EN	rw	0x0	-	对应唤醒源边沿选择 0：关闭唤醒功能 1：打开唤醒功能

6. WKUP_CON1: wakeup edge control register

Bit	Name	RW	Default	RV	Description
31-17	RESERVED	-	0x0	-	预留
16-0	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

7. WKUP_CON2: wakeup clear pending control register

Bit	Name	RW	Default	RV	Description
31-17	RESERVED	-	0x0	-	预留
16-0	WKUP_CPND	w	0x0	-	对应唤醒中断清除，写 1 清零

8. WKUP_CON3: wakeup pending control register

Bit	Name	RW	Default	RV	Description
31-17	RESERVED	-	0x0	-	预留
16-0	WKUP_PND	r	0x0	-	对应唤醒源中断标记