BL1824x

低功耗蓝牙兼容 & 2.4-GHz 专用 SOC 芯片 V0.1

数据手册

目录

版本历史

Version	Revision	Date	Author	Reviewer
V1.0	Initial version	2024/06/26		

1. BL1824 概览

1.1. 简述

BL1824 是应用于 BLE 和专有 2.4GHz 兼容的最优功耗 SOC 芯片。 内部集成了高性能低功耗射频收发器，蓝牙基带，和丰富的外围接口。 BL1824 也集成了提供高效电源管理的 PMU。她应用于 BLE 系统，私有的 2.4G 系统，人机接口设备（键盘，鼠标，和遥控器）， 运动休闲设备，手机附件和消费电子等领域。

BL1824 兼容蓝牙 5.1 版本，支持所有蓝牙 4.2 标准的功能。

芯片集成了 64MHz 主频的高性能 MCU，DMA, GPIO, SPI, UART, Timer, Watchdog 等，支持 32MHz 外部晶振，还集成了多用途的 10 位 ADC.

BL1824 集成了 16K 的片上 SRAM, 24K OTP，支持用户定义的 IDE 系统，片上 MCU 的 JTAG 开发调试。

1.2. 芯片特性

- CPU
  - ARM® Cortex™-M4, 最大支持 64MHz 主频
- 储存器
  - 24KB OTP
  - 16KB SRAM
  - 串行 Flash：128KB
  - I-Cache RAM 2KB
- 时钟
  - 32MHz 晶振, 32MHz RC 震荡器, 32.768KHz RC 低频时钟
- 链路控制
  - BT 5.1 LE PHY 连接控制器
  - 专用 2.4-GHz 连接控制器
- RF 收发器
  - 灵敏度： - 97dBm @BLE 1Mbps
  - 发射功率：-20 ~ +10dBm
  - 接收尖峰电流：13.5mA
  - 发射尖峰电流：14mA @0dBm
- 软件协议栈
  - 兼容 BLE V5.1
  - 支持 Mesh 组网
  - 支持外挂 MCU 应用的蓝牙网络处理接口
  - 所有示例应用程序和概要文件
  - 支持 OTA 升级
  - SWD 接口
- 外围设备

- DMA x 4
- UART x 3
- 可灵活配置的 GPIO
- SPI 主从接口 x 1
- PWM x 6
- 看门狗
- 16-bit Timer x 3
- 单端 10 位 GPADC x 7
- 硬件 AES 加密

- 电源管理
  - 睡眠电流：3.0uA(32K 时钟运行, 16KB RAM 保持状态下)
  - 工作电压：1.8V~3.6V
- 工作温度
  - -40℃ ~ +105℃

1.3. 系统功能框图

BL1824 是一个低功耗的蓝牙收发芯片， 芯片集成了蓝牙基带，PHY，和专用 2.4G 协议。MCU 处理系统的硬件包含了 AHB 总线, RAM, DMA, SFLASH, GPIO 通过 AHB 交换数据, 所有其它外设处理数据也是通过 AHB 到 APB 桥和 APB 总线进行的.

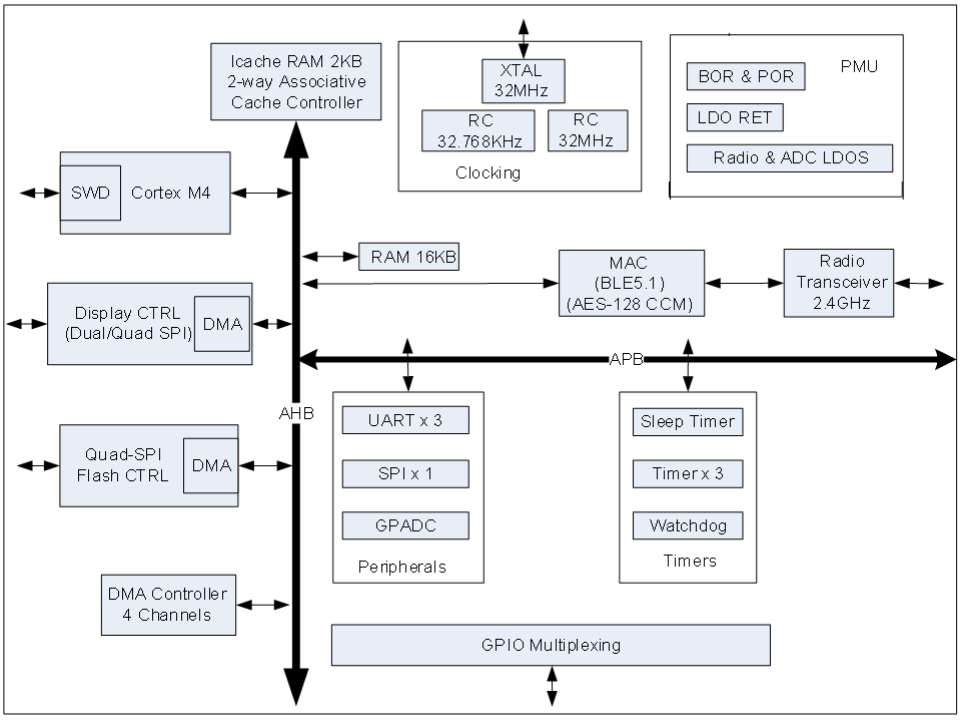


图 1.1 BL1824 系统框图

1.4. 应用

BL1824 可以应用于以下领域：

.Mesh 灯控

.蓝牙遥控器

.数字钥匙

.智能门锁

.IoT 模块

.运动健康

.有源标签

图 1.2 BL1824 典型应用

2. 脚位图

2.1. BL1824 SOP16

BL1824xPG 管脚定义如下：

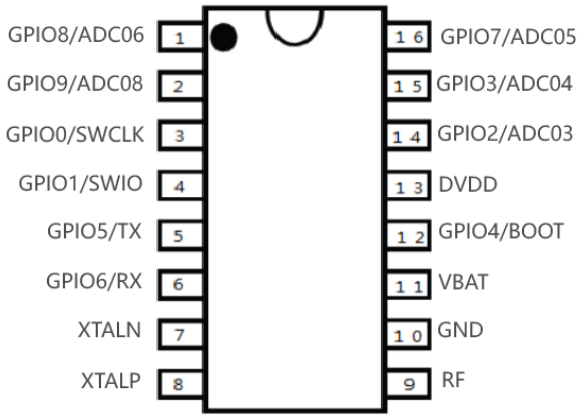


图 2.1 BL1824 SOP16 管脚定义

2.2. 管脚描述

名称	1824AP	类型	描述	备注
GPIO8/ ADC06	1	Digital/ Analog	Digital GPIO/ GPADC input	Note 1
GPIO9/ ADC08	2	Digital/ Analog	Digital GPIO/ GPADC input	Note 1
GPIO0/ SWCLK	3	Digital	Digital GPIO/SWCLK	Note 1
GPIO1/ SWDIO	4	Digital	Digital GPIO/SWDIO	Note 1
GPIO5/ UART_TX	5	Digital	Digital GPIO/UART_TX	Note 1
GPIO6/ UART_RX	6	Digital	Digital GPIO/UART_RX	Note 1
XTAL32M_N	7	Analog	32M crystal oscillator N input	
XTAL32M_P	8	Analog	32M crystal oscillator P input	
RF	9	RF	RF input/output	

GND	10	Power	Digital ground	
VBAT	11	Power	Power supply	
GPIO4/BOOT	12	Digital	Digital GPIO/BOOT	Note 1
DVDD	13	Power	Digital Circuit Power	
GPIO02/ ADC03	14	Digital/ Analog	Digital GPIO/ GPADC input	Note 1
GPIO03/ ADC04	15	Digital/ Analog	Digital GPIO/ GPADC input	Note 1
GPIO07/ ADC05	16	Digital/ Analog	Digital GPIO/ GPADC input	Note 1

注 1: 所有数字外设管脚都可以定义为任意的 GPIO.

(**): 为了确保复位正常，所有的 RSTB 脚被拉低的时间应该大于 40us.

表 2.1 BL1824 管脚定义

3. MCU 系统

BL1824 MCU 系统包含 ARM Cortex-M4 处理器，他的总线以及外设，包括所有的可映射的 GPIO，都图示于下图：

BL1824 处理器具有 32 位的指令处理系统，他的 Thumb-2 模式支持使用 16 位和 32 位的混合指令，以最大化代码执行密度。

MCU 的存储器有一个特殊的保留电压，使它在不同的应用场景下具有不同的记忆模式：

- OFF
- ON
- Retention

下面是 Cortex-M4 所支持的功能选项：

- 系统节拍定时器 (SysTick)
- Flash 补丁和断点单元 (FPB)

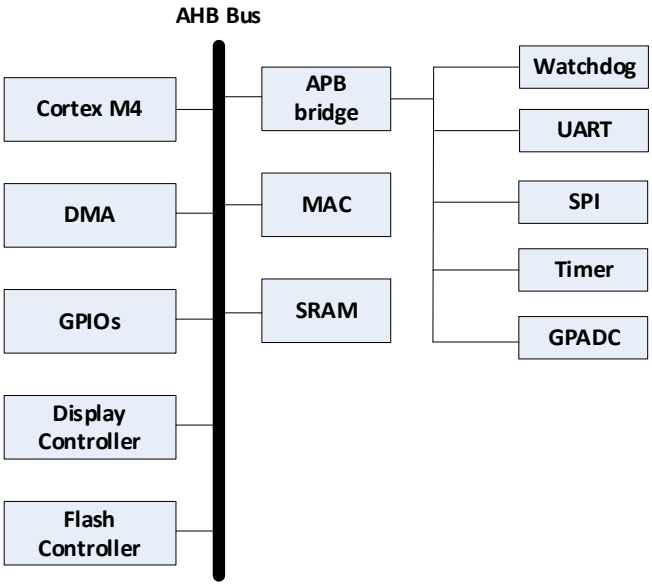


Figure 3.1 Micro-controller Subsystem

3.1. MCU 调试

调试使用串行调试线(SWD).

3.2. 中断向量表

BL1824 中断向量表：

Number	Interrupt name	Bit	Description
0	BT BB COMBO	1	BLE 事件
1	BB_NATIVE_INT	1	基带睡眠唤醒
2	DMA COMBO	1	DMA 中断
3	GPIO COMBO	1	GPIO 中断
4	TIMER COMBO	1	Timer 中断
5	2P4G_RF_IRQ	1	专用 2.4G 协议中断
6	2P4G_RF_SPI_IRQ	1	2.4G 的 SPI 通讯中断
7	PMU_TIMER	1	系统定时器
8	LCD_SPI_INT	1	显示控制总中断
9	UART1 COMBO	1	UART1 总中断
10	OTP_INT	1	OTP 中断
11	PIN_WAKEUP_INT	1	GPIO 唤醒中断
12	ADC	1	GPADC 中断
13	SPI MASTER 0 COMBO	1	SPI0 中断
14	SFLASH INT	1	FLASH 控制总中断
15~22	SOFT_INT	8	软中断
24	CRY32M_DIG_READY	1	32M 晶振起振标志
25	UART0 INT	1	UART0 中断
26	GPIO INT	1	GPIO COMBO 中断
28-30	TIMER	3	Timers 总中断
32	UART2 INT	1	UART2 中断

表 3.1 MCU 中断向量表

3.3. 电气特性

3.3.1. 最大使用条件

Parameter	Minimum	Maximum	Units
供电电压 (VBAT)	-0.3	3.9	V
做大结温度	-40	125	ºC
储藏温度	-40	125	ºC

表 3.2 BL1824 绝对最大使用条件

3.3.2. 推荐工作条件

Rating	Min	Typ	Max	Unit
操作温度	-40	-	85	ºC
数字内核电压	1.0	1.1	1.2	V
供电电压 (VBAT)	1.8	3.3	3.6	V
I/O 电压	VBAT	VBAT	VBAT	V

表 3.3 BL1824 推荐工作条件

3.3.3. ESD 性能

Parameter	Condition	Min	Typ	Max	Unit
人体接触模式(HBM)	测试方法: ESDA/JEDEC JS-001-2017	-	±3000	-	V
机械接触模式(MM)	所有脚位，测试方法: JESD22 -A115C	-	±100	-	V

表 3.4 ESD 特性

3.4. 功能模块寻找表

Base Address	Module
0x40000000	SYS_REG
0x40001000	CPM
0x40004000	RNG
0x4000B000	2.4G_RF
0x40020000	BT_PHY
0x40040000	UART1
0x40050000	SPI0
0x40080000	UART0
0x400A0000	DA_IF
0x400C0000	TIMER
0x400E0000	PMU
0x400F0000	UART2

表 3.5 模块寻址映射表

4. 内存

4.1. 内存介绍

BL1824 SOC 芯片的存储器包括 SRAM 和堆栈 Flash， CPU 和外设可以访问该存储器，进而更好的处理外设下面将介绍内存的地址映射。

4.2. 内存映射

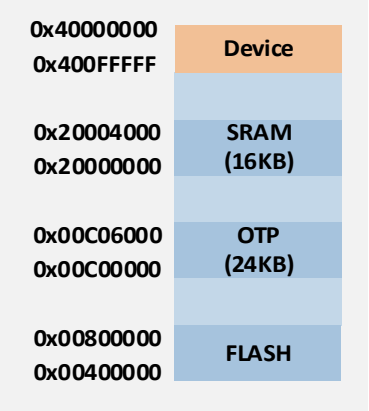


图 4.1 内存映射

4.3. APB 寻址空间

下图是内存映射的 APB 部分：

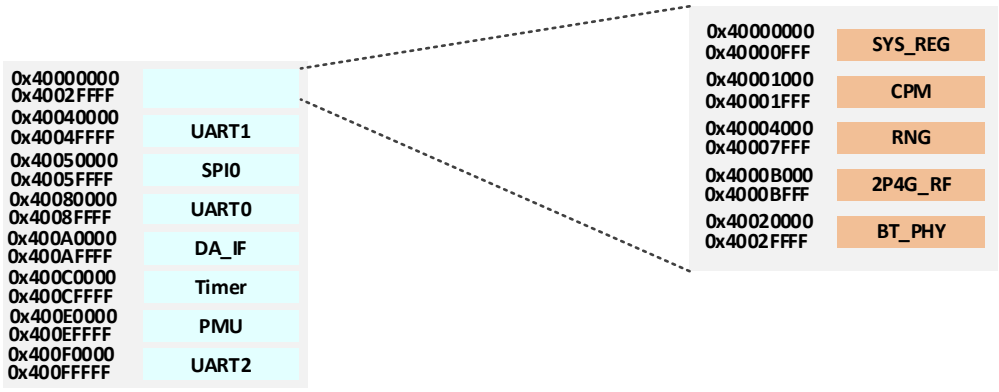


图 4.2 APB 内存映射

5. PMU

5.1. 电源管理介绍

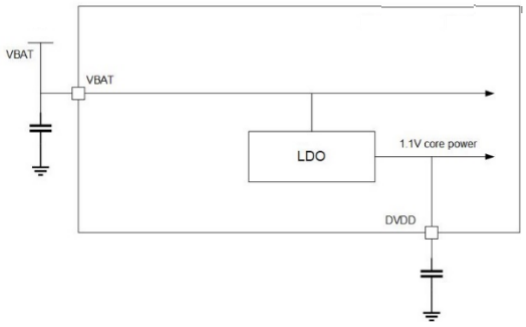


图 5.1 电源管理单元架构

电源管理单元 (PMU)为不同的功能模块提供可变的电压和电流偏置，实现其电源的开、关功能。

5.2. 数字 LDO

数字 LDO 管理单元为所有的数字逻辑模块和存储模块提供电源。

Parameter	Symbol	Min	Typ	Max	Unit	Comment
输入电压		1.8	3.3	3.6	V	
输出电压		1.0	1.1	1.2	V	
外接负载电容			1.0		μF	

表 5.1 数字核心电源技术参数

5.3. POR/BOD

上电复位(POR) 电路在供电电压达到要求的电压水平时使系统复位。掉电侦测(BOD)电路在电压降到掉电阈值的时候使系统进入复位状态.

参数	Symbol	Min	Typ	Max	Unit	Comment
掉电阈值			1.67		V	

表 5.2 POR/BOD 技术参数

6. 外设

6.1. 管脚复用

6.1.1. 简介

BL1824 有一个可配置的管脚复用模块(Pin MUX)，它能将不同的外设映射到任意的 GPIO 上。

6.1.2. 主要特性

PINMUX 具有下列特性:

- 外设管脚复用有很多种选择。
- 外设管脚复用可以由寄存器 REG_GPIOX_MUX 配置。

6.1.3. 功能描述

所有管脚复用选择如下表所示：

PINMUX 有更多的复用选择。当你为一个接口选择一个复用选项时，请确保该接口的所有信号都应配置为所选的复用选项。以下的 PINMUX 表设置了接口的信号。

NAME	NUMBER
PINMUX_JTAG_MODE_CFG	0
PINMUX_DBG_MODE_CFG	1
PINMUX_SPI0_MST_SDA_I_CFG	3
PINMUX_SPI0_MST_SDA_O_CFG	4
PINMUX_SPI0_MST_CSN_CFG	5
PINMUX_SPI0_MST_SCK_CFG	6
PINMUX_SFLASH_SI_CFG	8
PINMUX_SFLASH_SO_CFG	9
PINMUX_SFLASH_HD_CFG	10
PINMUX_SFLASH_WP_CFG	11
PINMUX_SFLASH_CK_CFG	12
PINMUX_SFLASH_CSN_CFG	13
PINMUX_UART1_SDA_I_CFG	15
PINMUX_UART1_SDA_O_CFG	16
PINMUX_UART1_CTS_I_N_CFG	17
PINMUX_UART1_RTS_O_N_CFG	18

PINMUX_TX_EXT_PD_CFG	19
PINMUX_RX_EXT_PD_CFG	20
PINMUX_GPIO_MODE_CFG	28
PINMUX_SFLASH1_CSN_1_CFG	24
PINMUX_SFLASH1_SI_CFG	25
PINMUX_SFLASH1_SO_CFG	29
PINMUX_SFLASH1_HD_CFG	30
PINMUX_SFLASH1_WP_CFG	31
PINMUX_SFLASH1_CK_CFG	32
PINMUX_SFLASH1_CSN_CFG	33
PINMUX_TIMER0_ETR_CFG	34
PINMUX_TIMER1_ETR_CFG	35
PINMUX_TIMER2_ETR_CFG	36
PINMUX_TIMER0_BKIN_CFG	37
PINMUX_TIMER1_BKIN_CFG	38
PINMUX_TIMER2_BKIN_CFG	39
PINMUX_TIMER0_IO_0_CFG	40
PINMUX_TIMER0_IO_1_CFG	41
PINMUX_TIMER0_TOGGLE_N_0_CFG	44
PINMUX_TIMER1_IO_0_CFG	47
PINMUX_TIMER1_IO_1_CFG	48
PINMUX_TIMER1_TOGGLE_N_0_CFG	51
PINMUX_TIMER2_IO_0_CFG	54
PINMUX_TIMER2_IO_1_CFG	55
PINMUX_TIMER2_TOGGLE_N_0_CFG	58
PINMUX_UART0_SDA_I_CFG	59
PINMUX_UART0_SDA_O_CFG	60
PINMUX_UART2_SDA_I_CFG	61
PINMUX_UART2_SDA_O_CFG	62

表 6.1 外设管脚复用

6.1.4. PINMUX 寄存器映射

Address	Name	Description
0X40000080	PIN_MUX_CTRL_1	PINMUX 控制
0X40000084	PIN_MUX_CTRL_2	PINMUX 控制
0X40000088	PIN_MUX_CTRL_3	PINMUX 控制
0X4000008C	PIN_MUX_CTRL_4	PINMUX 控制
0X40000090	PIN_MUX_CTRL_5	PINMUX 控制

PIN_MUX_CTRL_1 address: 0x40000080

Bit	R/W	Reset	Name	Description
31:30	N/A	0x0	N/A	保留
29:24	RW	0x0	GPIO3_MUX_REG	gpio3 复用配置
23:22	N/A	0x0	N/A	保留
21:16	RW	0x0	GPIO2_MUX_REG	gpio2 复用配置
15:14	N/A	0x0	N/A	保留
13:8	RW	0x0	GPIO1_MUX_REG	gpio1 复用配置
7:6	N/A	0x0	N/A	保留
5:0	RW	0x0	GPIO0_MUX_REG	gpio0 复用配置

PIN_MUX_CTRL_2 address: 0x40000084

Bit	R/W	Reset	Name	Description
31:30	N/A	0x0	N/A	保留
29:24	RW	0x0	GPIO7_MUX_REG	gpio7 复用配置
23:22	N/A	0x0	N/A	保留
21:16	RW	0x0	GPIO6_MUX_REG	Gpio6 复用配置
15:14	N/A	0x0	N/A	保留
13:8	RW	0x0	GPIO5_MUX_REG	gpio5 复用配置
7:6	N/A	0x0	N/A	保留
5:0	RW	0x0	GPIO4_MUX_REG	gpio4 复用配置

PIN_MUX_CTRL_3 address: 0x40000088

Bit	R/W	Reset	Name	Description
31:30	N/A	0x0	N/A	保留
29:24	RW	0x0	GPIO11_MUX_REG	gpio11 复用配置
23:22	N/A	0x0	N/A	保留
21:16	RW	0x0	GPIO10_MUX_REG	gpio10 复用配置
15:14	N/A	0x0	N/A	保留
13:8	RW	0x0	GPIO9_MUX_REG	gpio9 复用配置
7:6	N/A	0x0	N/A	保留
5:0	RW	0x0	GPIO8_MUX_REG	gpio8 复用配置

PIN_MUX_CTRL_4 address: 0x4000008C

Bit	R/W	Reset	Name	Description
31:30	N/A	0x0	N/A	保留
29:24	RW	0x1c	GPIO15_MUX_REG	gpio15 复用配置
23:22	N/A	0x0	N/A	保留
21:16	RW	0x0	GPIO14_MUX_REG	gpio14 复用配置
15:14	N/A	0x0	N/A	保留
13:8	RW	0x0	GPIO13_MUX_REG	gpio13 复用配置
7:6	N/A	0x0	N/A	保留
5:0	RW	0x0	GPIO12_MUX_REG	gpio12 复用配置

PIN_MUX_CTRL_5 address: 0x40000090

Bit	R/W	Reset	Name	Description
31:22	N/A	0x0	N/A	保留
21:16	RW	0x1c	GPIO18_MUX_REG	gpio18 复用配置
15:14	N/A	0x0	N/A	保留
13:8	RW	0x1c	GPIO17_MUX_REG	gpio17 复用配置
7:6	N/A	0x0	N/A	保留
5:0	RW	0x1c	GPIO16_MUX_REG	gpio16 复用配置

6.2. DMA

6.2.1. 简介

DMA 是直接存储处理控制器，它能高效的在总线上传输数据区块。

6.2.2. 主要功能

- 兼容 AMBA™ 2 AHB 协议规范。
- 支持 4 个 DMA 通道。
- 支持多达 8 对请求/确认硬件握手。
- 提供具有 2 个优先级的轮询仲裁。
- 支持 8/16/32 位宽的数据传输。

6.2.3. 功能描述

DMA 支持多达 8 个通道。每个 DMA 通道提供一组寄存器以描述预期的数据传输。可以同时使能多个 DMA 通道， 但是在同一时刻 DAM 控制器只服务于一个通道。

以下图示展示了通道数据传输时序。为了避免通道空闲，DMA 控制器轮流服务所有准备就绪的通道，每次最多执行 SrcBurstSize 的数据传输。因此，当总传输大小（TranSize）大于源突发大小（SrcBurstSize）时，通道的数据传输可能会被分割成几个部分。当一个通道的全部数据传输完成后，DMA 控制器将更新中断状态寄存器 IntStatus，并且如果启用了终端计数中断，将产生 dma_int 中断信号。

当发生错误时，通道的数据传输将被停止。软件也可以中止通道的数据传输。在这两种情况下，DMA 控制器将禁用该通道，并且如果启用了相应的中断，将产生 dma_int 中断。

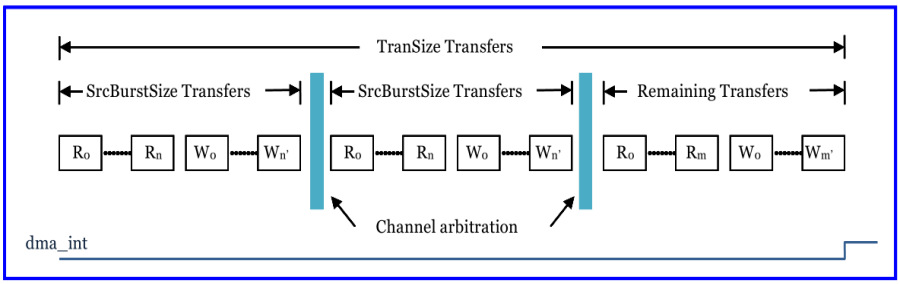


图 6.1 DMA 数据传输实例

6.2.3.1. 通道仲裁

DMA 提供两个优先级用于通道仲裁。每个通道都通过通道控制寄存器 ChnCtrl 的优先级字段与一个优先级相关联。在通道仲裁过程中，DMA 控制器首先选择一个高优先级的通道。如果没有高优先级的通道，则选择低优先级的通道。相同优先级的通道将通过轮询方式

选择。

6.2.3.2. 硬件握手

DMA 提供最多 16 对硬件握手信号（dma_req/dma_ack），用于与低速设备的数据传输。下图给出了硬件握手的一个示例。设备只有在准备好足够数据进行传输或有足够的空闲空间接收传入数据时才应产生中断 dma_req。当 DMA 控制器发现有 dma_req 中断时，它才会发出总线请求进行数据的读/写，从而避免总线无限期地处于等待状态。当 DMA 控制器完成与设备的 SrcBurstSize 的数据传输时，它会产生中断 dma_ack。设备在检测到 dma_ack 中断后应取消中断请求 dma_req。DMA 控制器在检测到 dma_req 被取消中断后应取消中断 dma_ack。如果在数据传输过程中遇到错误，DMA 控制器将禁用通道并且不产生中断 dma_ack。错误处理软件应重置 DMA 传输源和目标，以取消中断 dma_req。

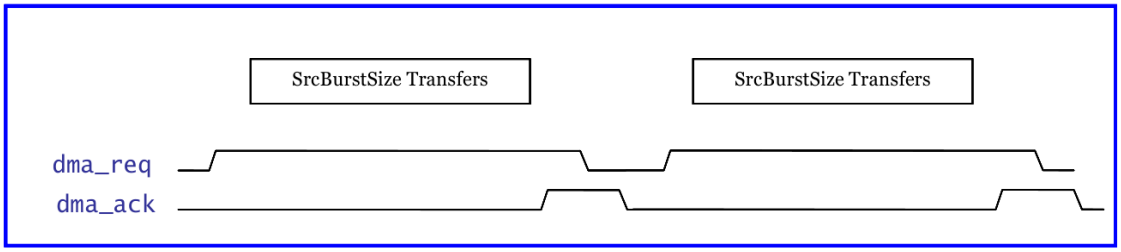


图 6.2 硬件握手实例

6.2.3.3. 链式传输

DMA 提供了链式传输功能，可以连续传输多个数据块，而无需主处理器的干预。

在开始链式传输之前，必须构建一个链表结构来描述要移动的数据块和相关的控制设置。链表的第一个元素（链表的头部）由通道控制寄存器描述。链表的其余元素由存储在内存中的链表描述符指定，链表描述符中保存了加载到通道控制寄存器以继续数据传输的控制值。下图显示了链表结构的一个示例。

当通道启用时，DMA 控制器将首先根据通道控制寄存器传输数据。数据传输完成后，DMA 控制器将通过跟随 ChnLLPointer 继续数据传输。如果 ChnLLPointer 不为零，则将指向的链表描述符的内容加载到通道控制寄存器中。加载的描述符成为链表的新头部，此过程重复进行，直到 ChnLLPointer 为零。

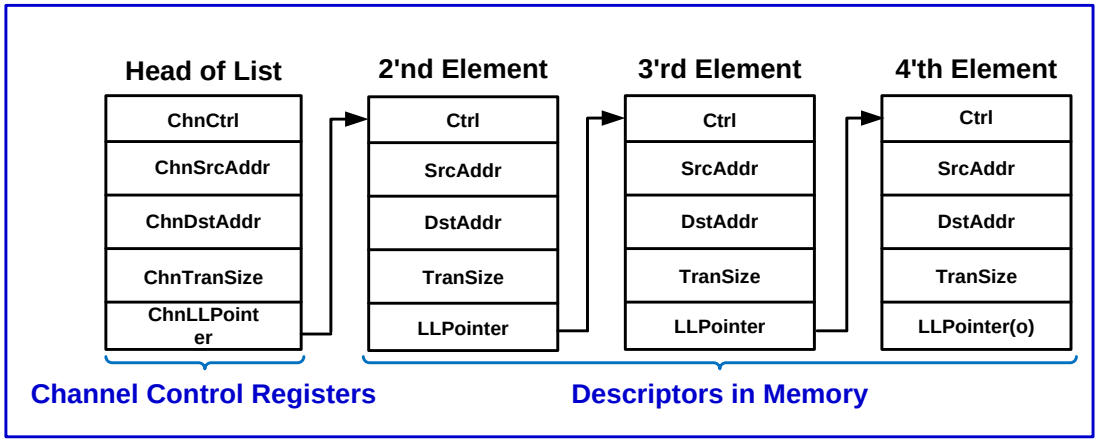


图 6.3 链式传输的链表结构

当一个通道的终端计数中断（IntTCMask）被启用时，DMA 控制器会在完成列表头部的数据传输后生成一个中断并禁用该通道。如果 ChnLLPointer 不为零，那么在生成中断之前，通道控制寄存器将被预加载下一个描述符。中断处理软件可以通过重新启用通道来恢复链式传输。

下表显示了链表描述符的格式。每个描述符字的位字段定义与相应的通道控制寄存器相同，除了通道启用位，在链表描述符中是保留的。

Name	Offset	Description	Format
Ctrl	0x44+n*0x14	通道控制	参加 DMA 寄存器映射
SrcAddr	0x48+n*0x14	源地址	参加 DMA 寄存器映射
DstAddr	0x4c+n*0x14	目的地址	参加 DMA 寄存器映射
TranSize	0x50+n*0x14	总传输大小	参加 DMA 寄存器映射
LLPointer	0x54+n*0x14	链表指针	参加 DMA 寄存器映射

表 6.2 链表描述符格式

6.2.3.4. 数据顺序

DMA 提供三种地址控制模式：递增模式、递减模式和固定模式。在递增模式下，DMA 控制器访问源/目标数据后地址会增加。在递减模式下，DMA 控制器访问源/目标数据后地址会减少。在固定模式下，DMA 控制器访问源/目标数据后地址保持不变。

当源地址控制模式与目标地址控制模式相同时，DMA 控制器会在源和目标之间保持数据的相同字节顺序。当源地址控制模式与目标地址控制模式相反时，写入目标的数据将与从源读取的数据的字节顺序相反。固定模式的数据顺序与递增模式相同处理。图 6.4，图 6.5，图 6.6 分别说明了当源地址模式为递增、递减和固定时，目标数据的字节顺序。

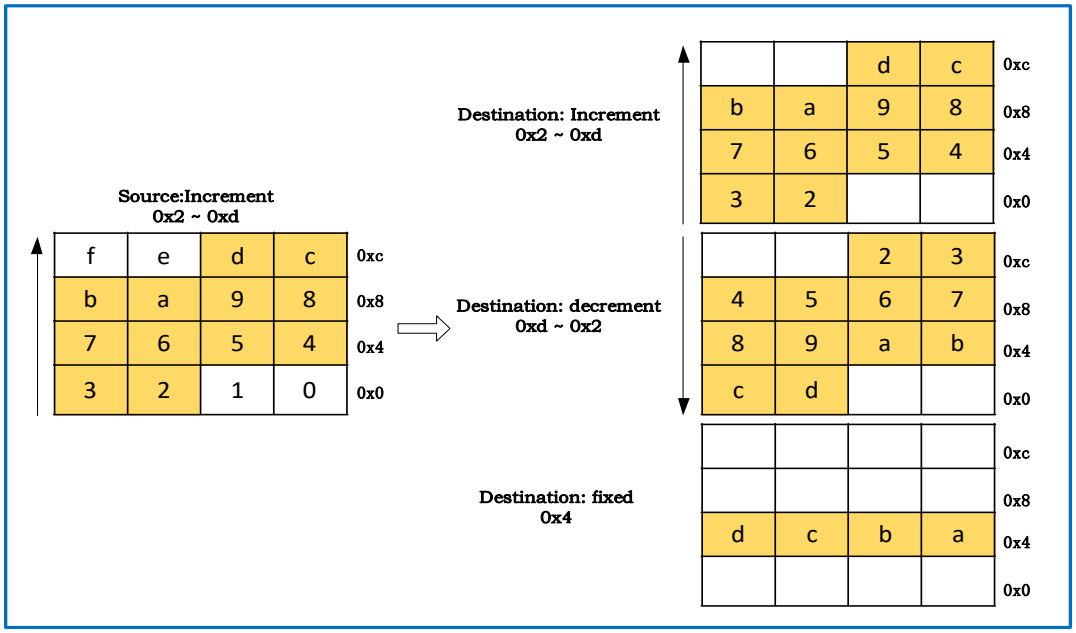


图 6.4 源地址为递增时的目标数据顺序

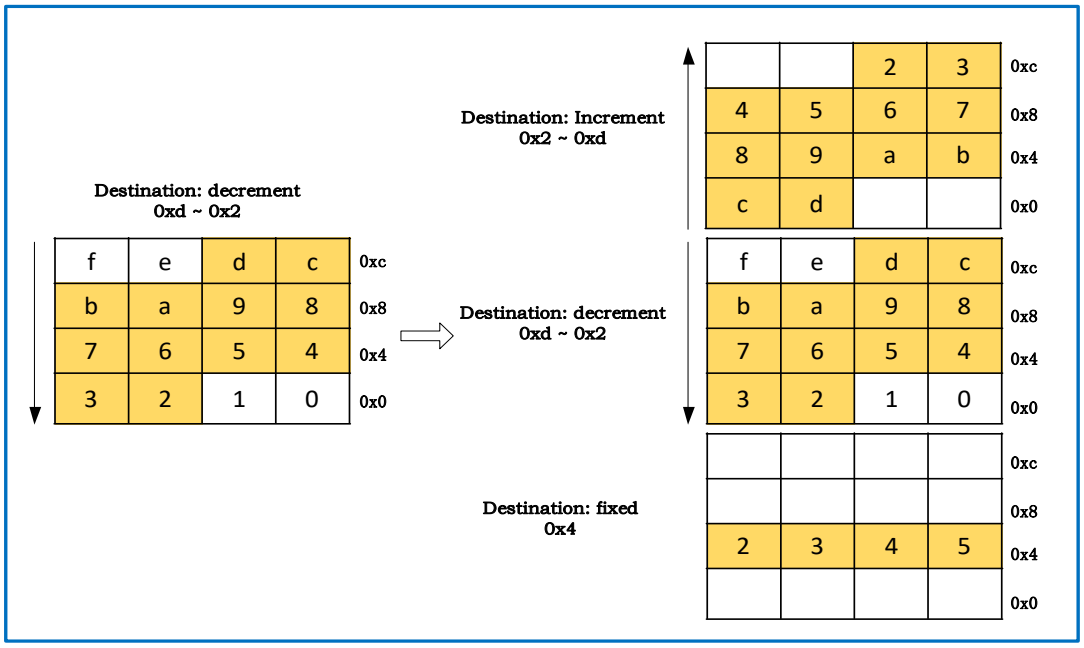


图 6.5 源地址为递减时的目标数据顺序

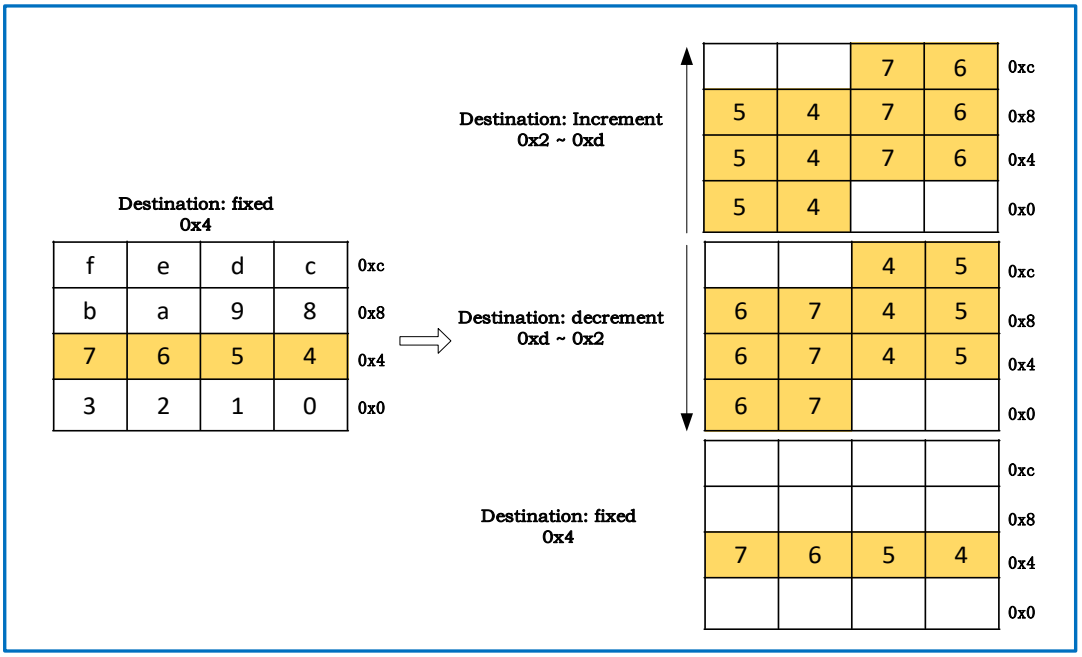


Figure 6.6 源地址为固定时的目标数据顺序

6.2.4. DMA 寄存器映射

offset	Name	Description
0x04	NA	保留
0x08	NA	保留
0x0C	NA	保留
0x10	NA	保留
0x14	NA	保留

0x18	NA	保留
0x1C	NA	保留
0x20	DMACTRL	DMAC 控制寄存器
0x24	NA	保留
0x28	NA	保留
0x2C	NA	保留
0x30	INTSTATUS	中断状态寄存器
0x34	CHEN	通道使能寄存器
0x38	NA	保留
0x3C	NA	保留
0x40	CHABORT	通道终止寄存器
0x44+n*0x14	CHNCTRL	通道 n 控制寄存器
0x48+n*0x14	CHNSRCADDR	通道 n 源地址寄存器
0x4C+n*0x14	CHNDSTADDR	通道 n 目的地址寄存器
0x50+n*0x14	CHNTRANSIZE	通道 n 传输大小寄存器
0x54+n*0x14	CHNLLPOINTER	通道 n 链表指针寄存器

DMA 寄存器描述

下面章节描述 DMA 寄存器的细节。每一列的缩略语概括如下：

RO: 只读

WO: 只写

R/W: 读写

R/W1C: 可读，只可将 1 写成 0.

DMAC 控制寄存器 (offset 0x20)

Bit	R/W	Reset	Name	Description
31:1	NA	NA	RESERVED	NA
0	W	0x0	RESET	软复位. 置 1， 将复位 DMA 核心，禁止所有通道。

中断状态寄存器(offset 0x30)

此寄存器包含终端计数、错误和中止状态。当通道遇到终端计数事件时，通道的终端计数状态被置位。当通道遇到错误/中止事件时，通道的错误/中止状态被置位。每个通道有一个状态位，当相应的通道未配置时，状态位为零。

Bit	R/W	Reset	Name	Description
31:24	NA	NA	RESERVED	NA
23:16	R/W$_1$C	0x0	TC	DMA 通道的终端计数状态，每个通道一个位。当一个通道传输完成且没有中止或错误事件时，终端计数状态被置为有效。 0=通道 N 没有终端计数状态 1=通道 N 有终端计数状态

15:8	R/W$_1$C	0x0	ABORT	通道的中止状态，每个通道一个比特位。 当通道传输被中止时，会置位中止状态。 0=通道 N 没有中止状态 1=通道 N 有中止状态
7:0	R/W$_1$C	0x0	ERROR	错误状态，每个通道一个位。 当通道传输遇到以下错误事件时，会断言错误状态： 总线错误 未对齐的地址 未对齐的传输宽度 保留的配置 0=通道 N 没有错误状态 1=通道 N 有错误状态

通道使能寄存器 (Offset 0x34)

寄存器显示 DMA 通道启用状态。状态字段仅在相应的通道被配置时存在。此寄存器是所有 ChnCtrl 寄存器启用字段的别名。

Bit	R/W	Reset	Name	Description
N:0	R	0x0	CHEN	所有 ChnCtrl 寄存器的使能字段的别名

通道终止寄存器 (Offset 0x40)

该寄存器控制 DMA 通道传输的中止，每个通道一位。向对应通道写入 1 以停止当前通道的传输。中止位在触发通道中止事件后由硬件自动清除。

Bit	R/W	Reset	Name	Description
N:0	W	0x0	CHABORT	向此字段写入 1 以停止通道传输。 只有在相应的通道启用时才能设置这些位。否则，对于未启用的通道，写入操作将被忽略。

通道 n 控制寄存器 (Offset 0x44+n*0x14)

Bit	R/W	Reset	Name	Description
31:30	R/W	NA	RESERVED	NA
29	R/W	0x0	PRIORITY	通道优先级。 0=较低优先级 1=较高优先级
28:25	NA	NA	RESERVED	NA
24:22	R/W	0x0	SRCBURSTSIZE	源突发大小。此字段指示在 DMA 通道重新仲裁之前的传输次数。 突发的总字节数是 SrcBurstSize * SrcWidth。 0x0: 1 次传输 0x1: 2 次传输 0x2: 4 次传输 0x3: 8 次传输 0x4: 16 次传输

				0x5: 32 次传输 0x6: 64 次传输 0x7: 128 次传输
21:20	R/W	0x2	SRCWIDTH	源传输宽度 0x0: 字节传输 0x1: 半字传输 0x2: 字传输 0x3: 保留，设置此值会触发错误异常
19:18	R/W	0x2	DSTWIDTH	目标传输宽度。 总传输字节和总突发字节都应与目标传输宽度对齐；否则将触发错误事件。 例如，如果总传输字节没有对齐到字或半字，则应将目标传输宽度设置为字节传输。 请参阅上面的 SrcBurstSize 字段以获取总突发字节的定义，以及第 3.12 节的描述的总传输字节的定义。 0x0：字节传输 0x1：半字传输 0x2：字传输 0x3：保留，将该字段设置为此值将触发错误异常。
17	R/W	0x0	SRCMODE	源 DMA 握手模式 0=普通模式 1=握手模式
16	R/W	0x0	DSTMODE	目标 DMA 握手模式 0=普通模式 1=握手模式
15:14	R/W	0x0	SRCADDRCTRL	源地址控制模式 0x0：地址增模式 0x1：地址减模式 0x2：固定地址模式 0x3：保留，设置此值会触发错误异常
13:12	R/W	0x0	DSTADDRCTRL	目标地址控制 0x0：地址递增模式 0x1：地址递减模式模式 0x2：固定地址 0x3：保留，设置此值会触发错误异常
11:8	R/W	0x0	SRCREQSEL	源 DMA 请求选择。选择源设备连接的请求/确认握手对。
7:4	R/W	0x0	DSTREQSEL	目标 DMA 请求选择。选择目标设备连接的请求/确认握手对。

3	R/W	0x0	INTABTMASK	通道中止中断屏蔽。 0=允许中止中断 1=禁用中止中断
2	R/W	0x0	INTERRMASK	通道错误中断屏蔽。 0=允许错误中断被触发 1=禁用错误中断
1	R/W	0x0	INTTCMASK	通道终端计数中断屏蔽 0=允许触发终端计数中断 1=禁用终端计数中断
0	R/W	0x0	ENABLE	通道使能位 0x0: 禁用通道 0x1: 使能通道

通道 n 源地址寄存器 (Offset 0x48+n*0x14)

Bit	R/W	Reset	Name	Description
31:0	R/W	0x0	SRCADDR	源起始地址。当一次传输完成时，其值会更新为结束地址+sizeof(SrcWidth)的大小。这个地址必须与源传输大小对齐；否则，将触发一个错误事件。

通道 n 目的地址寄存器 (Offset 0x4C+n*0x14)

Bit	R/W	Reset	Name	Description
31:0	R/W	0x0	DSTADDR	目标起始地址。当一次传输完成时，其值会更新为：结束地址+sizeof(DstWidth)的大小。这个地址必须与目标传输大小对齐；否则，将触发错误事件。

通道 n 传输大小寄存器 (Offset 0x50+n*0x14)

Bit	R/W	Reset	Name	Description
31:22	NA	NA	RESERVED	NA
21:0	R/W	0x0	TRANSIZE	源传输总的大小。传输的总字节数是：TranSize*SrcWidth。当 DMA 传输完成时，该值会被更新为零。如果启用了总传输大小为零的通道，将触发错误事件，并终止传输。

通道 n 链表指针寄存器:(Offset 0x54+n*0x14)

Bit	R/W	Reset	Name	Description
31:2	R/W	0x0	LLPOINTER	指向下一个块描述符的指针。指针必须是字对齐的。
1:0	NA	NA	RESERVED	NA

6.3. SPI 控制接口

6.3.1. 介绍

串行外设接口 (SPI)主要用于主机处理器与外设之间的同步串行通信。SPI 控制器是摩托罗拉 SPI 兼容接口。至少可以使用两个片选信号连接两个设备，数据输入（SPI_DI）、数据输出（SPI_DO）和时钟（SPI_CLK）信号对这两个设备都是共用的。SPI 接口也可以用来连接 SPI 闪存设备；读/写等命令是用户可配置的。

6.3.2. 主要特性

如下特性:

- Motorola SPI 兼容的 4 线接口。
- 仅仅支持 SPI 主模式。
- 支持 DMA 读写，DMA 最大可以读取 65535 Bytes (64KB -1)。
- 操作速度软件可配置。
- 支持对 Flash 设备的透明读取。

6.3.3. 功能描述

AHB 主接口将 SPI FIFO 中的数据传输到系统内存以进行 SPI 读取，或者将系统内存中的数据传输到 SPI FIFO 以进行 SPI 写入。CPU 使用 AHB 从接口来设置 SPI 读取或写入事务。单缓冲 FIFO 用于通过 AHB 主接口进行数据传输。同一个 FIFO 用于 SPI 读取和 SPI 写入。当事务完成时，如果已启用，可以生成一个中断，以通知 CPU 请求的事务已完成。

对于宽度小于 32 位的 SPI 设备，需要考虑 SPI 设备的字节序。由于数据是按最高有效字节（MSB）首先发送的，当 SPI 设备是小端字节序时，此模块在发送数据到 SPI 设备之前和从 SPI 设备接收数据之后，会内部交换数据字节（对于 8 位设备）或半字（对于 16 位设备）。SPI 设备的字节序是通过寄存器配置的，并在 SPI 配置寄存器的 WIDTH 字段中进行编程。下面的框架指示了为特定 SPI 设备配置发送的位顺序：

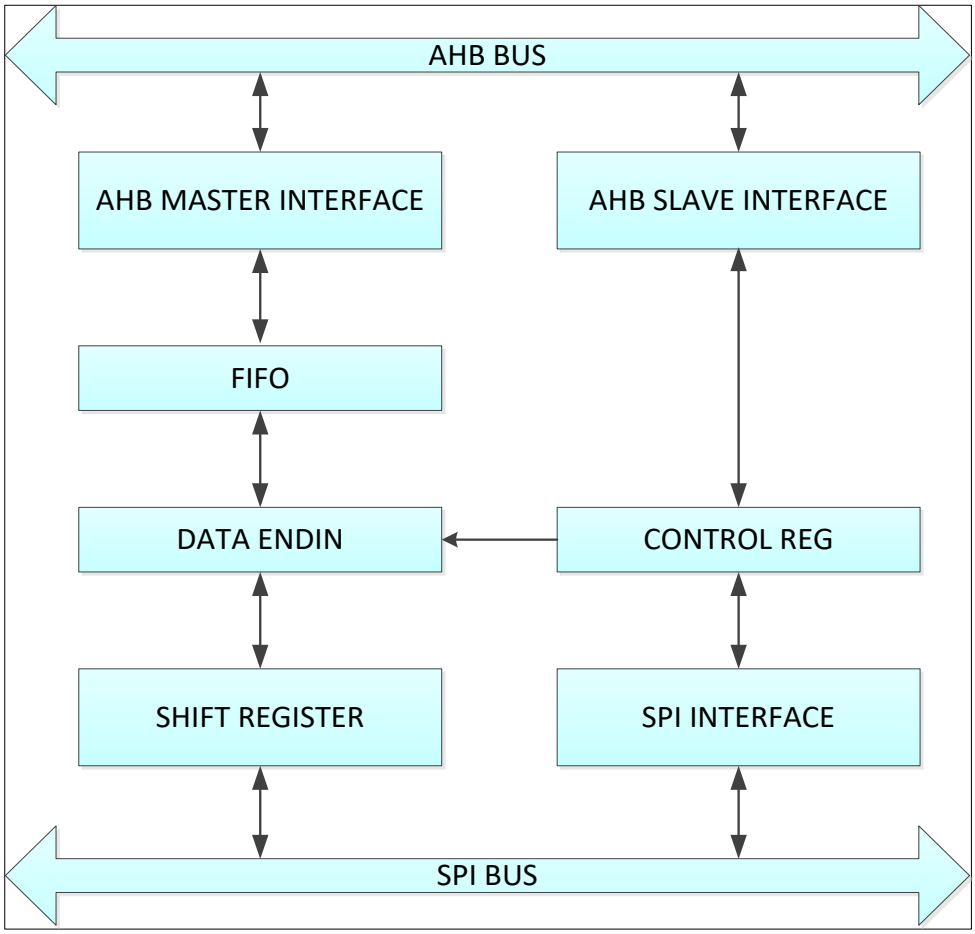


图 6.7 特定 SPI 设备配置

6.3.4. SFLASH 地址映射

SPI 地址映射到透明读取空间。 下面是 SPI 地址映射:

Address Range	Description
0x5000 0000 to 0x50FF FFFF(un-cacheable)	透明读取空间 (物理地址)
0x0000 0000 to 0x00FF FFFF (cacheable/un-cacheable)	重映射地址空间
0x5100 0000 to 0x51FF FFFF	SPI 控制寄存器空间

6.3.5. SFLASH 寄存器映射

Offset	Name	Description
0x00000	SPI_INTR_STATUS	SPI 中断状态寄存器
0X00004	SPI_RAW_INTR_STATUS	SPI Raw 中断状态寄存器
0X00008	SPI_INTR_MASK	SPI 中断屏蔽寄存器
0X0000C	SPI_COMMAND	SPI 命令寄存器
0X00010	SPI_COMMAND_DATA0_REG	SPI 命令的 data0 寄存器
0X00014	SPI_COMMAND_DATA1_REG	SPI 美丽的 data1 寄存器
0X00018	SPI_READ0_REG	SPI Read0 寄存器
0X0001C	SPI_READ1_REG	SPI Read1 寄存器
0X00020	SPI_ADDRESS_REG	SPI 地址寄存器
0X00024	SPI_READ_OPCODE_REG	SPI Read Opcode 寄存器
0X00028	SPI_CONFIGURATION_0	SPI 配置寄存器 0
0X0002C	SPI_CS_CONFIGURATION_0	SPI CS 配置寄存 0
0X00030	SPI_CONFIGURATION_1	SPI 配置寄存器 1
0X00034	SPI_CS_CONFIGURATION_1	SPI CS 配置寄存器 1
0X00038	TRANSPARENT_REMAP	透明重映射寄存器
0X0003c	REG_WP_HOLD	Reg_wp_hold 寄存器
0X00040	REG_SPI_CFG0	Reg_spi_cfg0 寄存器
0X00044	REG_SPI_CFG1	Reg_spi_cfg1 寄存器

SPI_INTR_STATUS address offset: 0x000

Bit	R/W	Reset	Name	Description
31:1	N/A	0x0	RESERVED	reserved
0	RO	0x0	SPI_CMD_DONE	SPI 命令完成

SPI_RAW_INTR_STATUS address offset: 0x004

Bit	R/W	Reset	Name	Description
31:1	N/A	0x0	RESERVED	reserved
0	RW	0x0	SPI_RAW_INTR_STATUS	SPI 命令完成中断 (屏蔽前的内部中断). SPI 命令完成时设置. 写入 1 清除中断状态.

SPI_INTR_MASK address offset: 0x008

Bit	R/W	Reset	Name	Description
31:1	N/A	0x0	RESERVED	reserved
0	RW	0x0	SPI_CMD_DONE_MASK	SPI 命令完成中断屏蔽。当设置为 1 时，允许中断触发中断。

SPI_COMMAND address offset: 0x00c

Bit	R/W	Reset	Name	Description
31:12	RW	0x0	DATA_BYTES	此字段指定了在发送命令数据位之后要传输的数据字节数。有效值为 0-65535。对于读取命令，如果此字段不是 4 的倍数，在将数据写入系统内存之前将填充零。此值在 SPI 通讯过程中递减，直到减到 0。
11:5	RW	0x0	CMD_BITS	此字段指定要发送的命令数据的位数。有效值为 0-64。此值在 SPI 通讯过程中递减，直到减到 0。请注意，64 个命令数据位定义在 2 个 32 位寄存器中。前 32 个命令数据位从 COMMAND_DATA0 寄存器发送，接下来的 32 位从 COMMAND_DATA1 寄存器发送。
4	RW	0x0	KEEP_CS	当这个位被设置时，CS 将在命令完成后保持断言状态。
3	RW	0x0	DATA_2_LANE_EN	2 数据通道模式启用，仅在 spi_if_mode 设置为 3 线模式 1 时可用。
2	RW	0x0	CHIP_SELECT	片选 0 = CS0, 1 = CS1
1:0	RW	0x0	COMMAND	指令。此字段在 SPI 传输完成后会自动清除。 0x0：无操作（NOP） 0x1：读取。在发送指令数据位后，数据会被传输到内存。 0x2：写入。在发送指令数据位后，数据会被传输到 SPI 设备。

SPI_COMMAND_DATA0_REG address offset: 0x010

Bit	R/W	Reset	Name	Description
31:0	RW	0x0	COMMAND_DATA	这是发送的命令数据，它在前 32 个 SPI 时钟周期内发送，具体取决于 CMD_BITS 字段。它是以 MSB（最高有效位）优先发送的（数据从第 31 位左移出）。在 SPI 事务期间，这个值保持不变。

SPI_COMMAND_DATA1_REG address offset: 0x014

Bit	R/W	Reset	Name	Description
31:0	RW	0x0	COMMAND_DATA	这是发送的命令数据，它在前 32 个 SPI 时钟周期内发送，具体取决于 CMD_BITS 字段。它是以 MSB（最高有效位）开始发送的（数据是从第 31 位向左移位输出的）。在 SPI 事务过程中，这个值被保持。

SPI_READ_DATA0_REG address offset: 0x018

Bit	R/W	Reset	Name	Description
31:0	RW	0x0	READ_DATA0	这个寄存器保存在前 32 个 SPI 时钟周期内捕获的数据。它是以最高位优先捕获的（数据从第 0 位左移入）。未使用的前导位将是 0。【怎么和下面一样？】

spi_read_data1_reg address offset: 0x01c

Bit	R/W	Reset	Name	Description
31:0	RW	0x0	READ_DATA1	这个寄存器保存在前 32 个 SPI 时钟周期内捕获的数据。它是以 MSB（最高有效位）优先捕获的（数据从第 0 位左移入）。未使用的前导位将是 0。

SPI_ADDRESS_REG address offset: 0x020

Bit	R/W	Reset	Name	Description
31:0	RW	0x0	ADDRESS	这个寄存器保存了系统内存地址用于数据传输。当从系统内存读取数据时（如果是写命令的情况下），或者当向系统内存写入数据时（如果是读命令的情况下），这个寄存器的值会增加 4。这个寄存器的最低两位始终为 0，以强制字对齐。

SPI_READ_OPCODE_REG address offset: 0x024

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	RESERVED	reserved
15:8	RW	0x3b	CS1_OPCODE	这个寄存器保存了用于在 CS1 地址空间发生透明读取时从串行闪存设备读取的 OPCODE。
7:0	RW	0x3b	CS0_OPCODE	这个寄存器保存了用于在 CS0 地址空间发生透明读取时从串行闪存设备读取的 OPCODE。

SPI_CONFIGURATION_0 address offset: 0x028

Bit	R/W	Reset	Name	Description
31:24	N/A	0x0	RESERVED	reserved
23:21	RW	0x0	SPI_DATA_MODE	0x0: 普通数据 0x1: RGB565 0x2: RG666 0x3: RGB888

20:18	RW	0x0	SPI_IF_MODE	0x0: 像 Flash 一样的; 0x1: sitronix 3 线模式 1 0x2: sitronix 3 线模式 2 0x3: sitronix 4 线模式 1 0x4: sitronix 4 线模式 2
17:16	RW	0x0	WIDTH	数据读/写的宽度。这个字段确保在小端系统中，数据能够正确地写入小于 32 位宽的设备。大端系统应使用 32 位数据设置。 00: 8 位数据 01: 16 位数据 1X: 32 位数据 注: 0x1 为 RGB565; 0x2 为 RGB666 和 RGB888
15	N/A	0x0	RESERVED	reserved
14	RW	0x0	FE_DLY_SAMPLE	下降沿延迟采样。 1：在延迟采样时钟（SPI_CLK_DLY）的下降沿采样 SPI_DI。 0：在内部采样时钟（SPI_CLK）的下降沿采样 SPI_DI。启用此位将允许更高频率的操作。 如果设置，dly_sample[13:12]必须设置为 1 或更大。这仅对 SPI 模式 0 和 3 有效。对于模式 1 和 2，这些位必须设置为 0。
13:12	RW	0x0	DLY_SAMPLE	延迟采样，这是在 SPI_CLK 下降沿之后，用于采样 SPI_DI 的 REF_CLK 的数量。设置这些位可以实现更高频率的操作。如果设置了 fe_dly_sample[14]，此字段必须设置为 1 或更大。这仅对 SPI 模式 0 和 3 有效。对于模式 1 和 2，这些位必须设置为 0。
11	N/A	0x0	RESERVED	reserved
10	RW	0x0	BP_CLOCK_DIV	忽略时钟分频器
9	RW	0x0	CPOL	时钟极性。 0：时钟在空闲时为低电平，每个时钟脉冲由上升沿后跟下降沿组成。 1：时钟在空闲时为高电平，每个时钟脉冲由下降沿后跟上升沿组成。
8	RW	0x0	CPHA	时钟相位。 0：输入数据在每个时钟脉冲的第一个边沿时被时钟控制。 1：输入数据在每个时钟脉冲的第二个边沿时被

				时钟控制。
7:0	RW	0x2	CLOCK_DIV	这个寄存器是系统时钟的分频器，用于生成 SPI 时钟。为了保持 50%的占空比时钟，应该只编程偶数值。此寄存器的最小值为 2。

SPI_CS_CONFIGURATION_0 address offset: 0x02c

Bit	R/W	Reset	Name	Description
31:24	RW	0x0a	CS_RECOVER	芯片选择恢复时间。 这是在芯片选择被取消后，必须经过的系统周期数，之后才能再次或对任何其他芯片选择进行断言。
23:16	RW	0x0a	CS_HOLD	芯片选择保持时间。 这是在最后一个时钟周期和芯片选择信号撤销之间的系统时钟周期数。
15:8	RW	0x0a	CS_SETUP	芯片选择设置时间。 这是芯片选择信号被激活到第一个时钟脉冲之间所需的系统时钟周期数。
7:1	N/A	0x0	RESERVED	reserved
0	RW	0x0	CS_POL	芯片选择极性。 0：芯片选择是低电平有效。 1：芯片选择是高电平有效。

SPI_CONFIGURATION_1 address offset: 0x030

Bit	R/W	Reset	Name	Description
31:24	N/A	0x0	RESERVED	reserved
23:21	RW	0x0	SPI_DATA_MODE	0x0: 普通数据 0x1: RGB565 0x2: RG666 0x3: RGB888
20:18	RW	0x0	SPI_IF_MODE	0x0: 像 Flash 一样的; 0x1: sitronix 3 线模式 1 0x2: sitronix 3 线模式 2 0x3: sitronix 4 线模式 1 0x4: sitronix 4 线模式 2
17:16	RW	0x0	WIDTH	数据读/写的宽度。这个字段确保在小端系统中，数据能够正确地写入小于 32 位宽的设备。大端系统应使用 32 位数据设置。

				00: 8 位数据 01: 16 位数据 1X: 32 位数据 注: 0x1 为 RGB565; 0x2 为 RGB666 和 RGB888
15	N/A	0x0	RESERVED	reserved
14	RW	0x0	FE_DLY_SAMPLE	下降沿延迟采样。 1：在延迟采样时钟（SPI_CLK_DLY）的下降沿采样 SPI_DI。 0：在内部采样时钟（SPI_CLK）的下降沿采样 SPI_DI。启用此位将允许更高频率的操作。 如果设置，dly_sample[13:12]必须设置为 1 或更大。这仅对 SPI 模式 0 和 3 有效。对于模式 1 和 2，这些位必须设置为 0。
13:12	RW	0x0	DLY_SAMPLE	下降沿延迟采样。 1：在延迟采样时钟（SPI_CLK_DLY）的下降沿采样 SPI_DI。 0：在内部采样时钟（SPI_CLK）的下降沿采样 SPI_DI。启用此位将允许更高频率的操作。 如果设置，dly_sample[13:12]必须设置为 1 或更大。这仅对 SPI 模式 0 和 3 有效。对于模式 1 和 2，这些位必须设置为 0。
11	N/A	0x0	RESERVED	reserved
10	RW	0x0	BP_CLOCK_DIV	忽略时钟分频器
9	RW	0x0	CPOL	时钟极性。 0：时钟在空闲时为低电平，每个时钟脉冲由上升沿后跟下降沿组成。 1：时钟在空闲时为高电平，每个时钟脉冲由下降沿后跟上升沿组成。
8	RW	0x0	CPHA	时钟相位。 0：输入数据在每个时钟脉冲的第一个边沿时被时钟控制。 1：输入数据在每个时钟脉冲的第二个边沿时被时钟控制。
7:0	RW	0x2	CLOCK_DIV	这个寄存器是系统时钟的分频器，用于生成 SPI 时钟。为了保持 50%的占空比时钟，应该只编程偶数值。此寄存器的最小值为 2。

SPI_CS_CONFIGURATION_1 address offset: 0x034

Bit	R/W	Reset	Name	Description
31:24	RW	0x0a	CS_RECOVER	芯片选择恢复时间。 这是在芯片选择被取消后，必须经过的系统周期数，之后才能再次或对任何其他芯片选择进行断言。
23:16	RW	0x0a	CS_HOLD	芯片选择保持时间。 这是在最后一个时钟周期和芯片选择信号撤销之间的系统时钟周期数。
15:8	RW	0x0a	CS_SETUP	芯片选择设置时间。 这是芯片选择信号被激活到第一个时钟脉冲之间所需的系统时钟周期数。
7:1	N/A	0x0	RESERVED	reserved
0	RW	0x0	CS_POL	芯片选择极性。 0：芯片选择是低电平有效。 1：芯片选择是高电平有效。

SPI_TRANSPARENT_REMAP address offset: 0x038

Bit	R/W	Reset	Name	Description
31:25	N/A	0x0	RESERVED	reserved
24:0	RW	0x0	REMAP_BASE	当控制器处于透明访问模式时，SPI 设备的地址将是 AHB 总线上的地址与该字段中 remap_base 配置进行 OR 运算的结果。

WP_HOLD address offset: 0x03C

Bit	R/W	Reset	Name	Description
31:2	N/A	0x0	RESERVED	reserved
1	RW	0x0	HOLD	将保持位输出到 spi 设备
0	RW	0x0	WP	写保护位输出到 spi 设备.

SW_SPI_CFG0 address offset: 0x040

Bit	R/W	Reset	Name	Description
30:24	RW	0x0	SW_DUMMY_CYCLE_CNT	数据阶段之前的空操作周期数.
22:16	RW	0x0	CMD_P1_BIT_CNT	Spi cmd phase1 位数 - cmd phase1 的数据将来自 reg_cmd_data1 寄存器。

13:12	RW	0x0	CMD_P1_BUS_WIDTH	Spi cmd phase1 总线宽度 0: 1 位; 1: 2 位; 2: 4 位；
9:8	RW	0x0	CMD_P0_BUS_WIDTH	Spi cmd phase0 总线宽度 0: 1 位; 1: 2 位; 2: 4 位；
6:0	RW	0x0	CMD_P0_BIT_CNT	Spi cmd phas0 位数. Cmd phase1 数据来自于寄存器 reg_cmd_data0.

SW_SPI_CFG1 address offset: 0x044

Bit	R/W	Reset	Name	Description
31	RW	0x0	SW_CFG_EN	1：控制器根据 SW_SPI_CFG0/SW_SPI_CFG1 寄存器中的软件配置，驱动 SPI 序列到 SPI 设备。 0：控制器从 SPI 命令中解码序列信息。
30:27	N/A	0x0	RESERVED	reserved
26:24	RW	0x0	BUF_WIDTH_BYTES	按字节计数的缓冲区宽度。
21:20	RW	0x0	SDATA_BUS_WIDTH	SPI 数据相位总线宽度 0: 1 位; 1: 2 位; 2: 4 位；
19:0	RW	0x0	SDATA_BYTE_CNT	输入/输出数据字节数

6.3.6. SFLASH 软件编程指导

6.3.6.1. 读 ID

为了读取闪存设备的 ID，SPI 主设备发送 8 位读取 ID 命令；然后闪存设备会返回 24 位 ID 值（总共传输 32 位）。因此，这个传输的命令位数将是 32 位；数据字节数为 0。ID 值将在 ReadData0 寄存器中可用。

- 将 SPI 命令数据 0 寄存器写入读取 ID 命令（0x9f）；请记住命令是先传输最高有效位（MSB）；因此写入 0x9f000000（SPI 命令数据 1 寄存器在此传输中不使用）。
- 编写 SPI 命令寄存器，具体细节如下：命令位数 = 32，数据字节数 = 0，Keep_CS = 0，Chip_select = 所需的芯片选择编号（例如 0），Command = Read，即 1。
- 等待，直到命令完成。一种方法是通过轮询 SPI Raw 中断状态寄存器中的 spi_raw_intr_status 位。另一种选择是使用中断。

- 从 SPI Read0 寄存器（位[23:0]）读取 ID 值。数据是先捕获最高位（MSB），即从第 0 位左移，因此最低位（LSB）始终是 0。

6.3.6.2. 读状态寄存器

读取状态寄存器命令可以在发起擦除或写入操作后发出，以检查该操作的状态。SPI 主设备发送 8 位命令，然后闪存返回 8 位状态值。因此，总共传输了 16 位。因此，此传输的命令位数将是 16；数据字节数将是 0。状态值将在读取数据 0 寄存器的低 8 位上可用。

- 将读取状态命令（0x05）写入 SPI 命令数据 0 寄存器；请记住，命令是先传输最高有效位（MSB）；因此写入 0x05000000（SPI 命令数据 1 寄存器在此传输中不使用）。
- 写 SPI 命令寄存器，包含以下细节：Write the SPI Command Register with the following details:
  - 命令位数 = 16，数据字节数 = 0，Keep_CS = 0，Chip_select = 需要的芯片选择编号（例如 0），Command= Read，即 1。
- 等待，直到命令完成。一种方法是通过轮询 SPI Raw 中断状态寄存器中的 spi_raw_intr_status 位。另一种选择是使用中断。
- 从 SPI Read0 寄存器（位[7:0]）读取状态值。数据是按 MSB（最高有效位）首先捕获的，即从位 0 开始左移，因此 LSB（最低有效位）将始终是 0。

6.3.6.3. DMA 写操作

对于 DMA 写操作，一次只能写入最多由闪存支持的页面大小的字节数。发送页面写入或页面编程命令（8 位）时，需要附带 24 位的起始闪存地址；之后跟随所需字节的数据。因此，命令位数应设置为 32 位，数据字节数设置为所需数量。

- 如果闪存只支持页编程命令，则执行扇区擦除操作；如果它支持页写入，则无需单独执行擦除命令。
- 向闪存发送写使能命令。
- 将页编程（或页写入）命令和 24 位的闪存起始地址写入 SPI 命令 Data0 寄存器；请记住，命令是按最高位（MSB）先传输的；因此，例如，如果闪存的起始地址是 0，则写入 0x02000000 以执行页面程序。（SPI 命令 Data1 寄存器在此传输中不使用）。
- 在 SPI 地址寄存器中设置 DMA 源地址。
- 写 SPI 命令寄存器，包含以下详细信息:
  - 命令位数=32，数据字节数=N（所需数量），Keep_CS=0，Chip_select=所需芯片选择编号（例如 0），Command = Write，即 2。
- 等待，直到命令完成。一种方法是通过轮询 SPI Raw 中断状态寄存器中的 spi_raw_intr_status 位。另一种选择是使用中断。
- 使用 Read-Status-Register 命令来确定闪存何时完成写入操作。

6.3.6.4. DMA 读操作

对于 DMA 读取操作，可以使用单一命令读取整个闪存。普通读取命令（0x03）只能在一定频率下使用（例如 M25P128 为 20 MHz）；超过该频率应使用快速读取命令（0x0b）。使用普通读取命令时，SPI 主机会发送一个 8 位的读取命令和 24 位的起始闪存地址；因此，普通读取命令的命令位数为 32 位。使用快速读取命令时，SPI 主机会发送一个 8 位的快速读取命令和 24 位的起始闪存地址；在开始返回有效数据之前，闪存设备会返回一个虚拟字节。因此，快速读取命令的命令位数为 40 位。

- 根据 SPI 时钟频率，使用普通读取命令（0x03）或快速读取命令（0x0b）写入 SPI 命令数据 0 寄存器，和 24 位起始闪存地址；请记住命令是先传输最高有效位（MSB）；因此，例如如果闪存的起始地址是 0，那么写入 0x0b000000 以执行快速读取。为了在虚拟字节间隔期间确保 MOSI 线不切换，写入 0x00000000 到 SPI 命令数据 1 寄存器进行快速读取，这样做更安全。
- 在 SPI 地址寄存器中设置 DMA 目标地址。
- 写 SPI 命令寄存器，包含以下详细信息:
  - 命令位数=正常读取时为 32 位或快速读取时为 40 位，数据字节数=N（所需数量），Keep_CS=0，Chip_select=所需芯片选择编号（例如 0），Command = Read，即 1。
- 等待，直到命令完成。一种方法是通过轮询 SPI Raw 中断状态寄存器中的 spi_raw_intr_status 位。另一种选择是使用中断。
- 数据从系统内存的 DMA 中目标地址起始处是有效的。

6.4. GPIO

6.4.1. 简介

BL1824 具有可以连接到各种信号接口的 GPIO。下图显示了一个 I/O 端口位的基本结构。

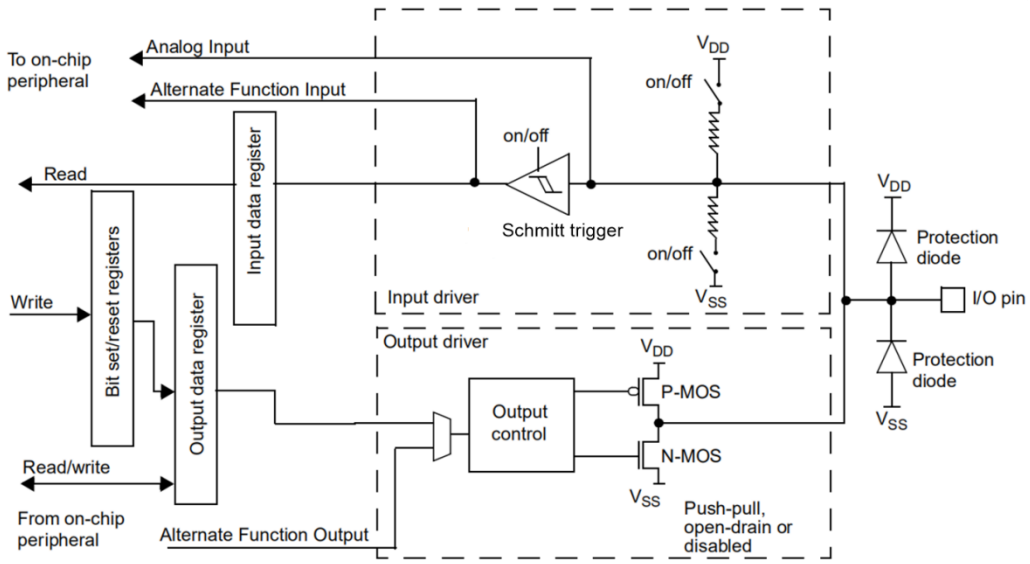


图 6.8 标准 I/O 端口位的基本结构

6.4.2. 主要特性

- BL1824: 19 个通用 I/O GPIOs (max)

6.4.3. 功能描述

GPIO 脚可以配置为输入输出，上拉/下拉电阻，输出保持。对于需要直接访问的应用，固件可以通过使用 GPIOx 寄存器来读写 IO。

数据手册中列出 I 每个 I/O 端口的具体硬件特性，通用输入输出（GPIO）端口的每个端口位都可以通过软件单独配置为几种模式:

- Input floating
- Input pull-up
- Input-pull-down
- Output open-drain
- Output push-pull
- Open-drain,pull up

6.4.3.1. 外部中断

所有端口都具有外部中断能力。要使用外部中断，必须将端口配置为输入模式。并且可以通过软件配置寄存器设置多种触发方式。

- Falling edge
- Rising edge
- Both edge
- High level
- Low level
- Disable trigger

6.4.3.2. 输入配置

I/O 配置成输入:

- 禁止输出缓冲区。
- 激活施密特触发器。
- 弱上拉和下拉电阻是否激活取决于输入配置（上拉、下拉或浮空）。
- I/O 口上的当前状态被采样到输入数据寄存器中。
- 读取输入数据寄存器以获取 I/O 状态。

下图展示了 I/O 端口位的输入配置：

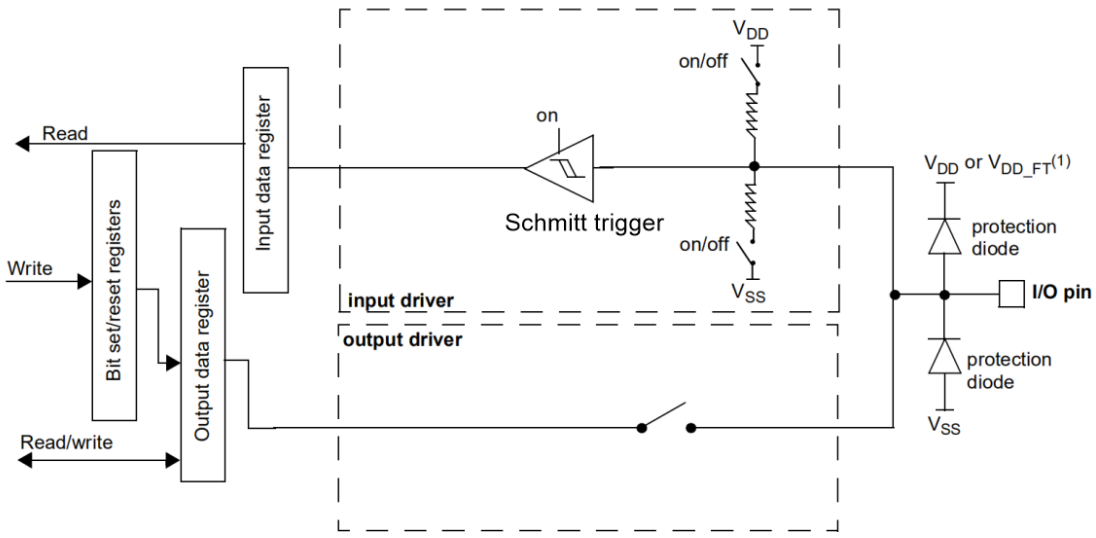


图 6.9 输入的悬空、上拉、下拉配置

6.4.3.3. 输出配置

当 I/O 口配置成输出模式:

- 输出缓存使能：
  - 开漏模式: 输出寄存器中的“0”激活 N-MOS，而输出寄存器中的“1”使端口处于高阻态（P-MOS 永远不会被激活）。
  - 推挽模式: 输出寄存器中的“0”激活 N-MOS，而输出寄存器中的“1”激活 P-MOS。
- 施密特触发器输入被激活了。
- 弱上拉和下拉电阻被禁用。
- I/O 脚上的数据被采样到输入数据寄存器中。
- 对输入数据寄存器的读取访问以获取开漏模式下的 I/O 状态。
- 对输出数据寄存器的读取访问将获取在推挽模式下最后一次写入的值。

下图形展示了 I/O 端口位的输出配置：

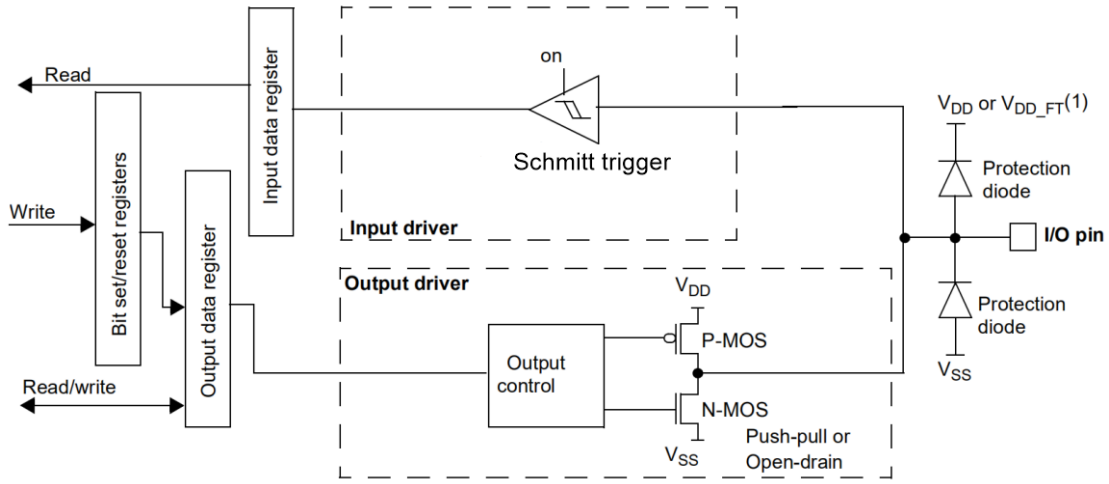


图 6.10 输出配置

6.4.3.4. 模拟功能配置

当 I/O 端口被配置为模拟配置时:

- 输出缓冲区是禁用的。
- 施密特触发器输入被停用，为 I/O 引脚的每个模拟值提供零消耗。施密特触发器的输出被强制设定为一个恒定值（0）。
- 弱上拉和下拉电阻被禁用。
- 读取输入数据寄存器的访问得到值“0”。
- 下面的图展示了 I/O 端口位的高阻抗模拟配置。

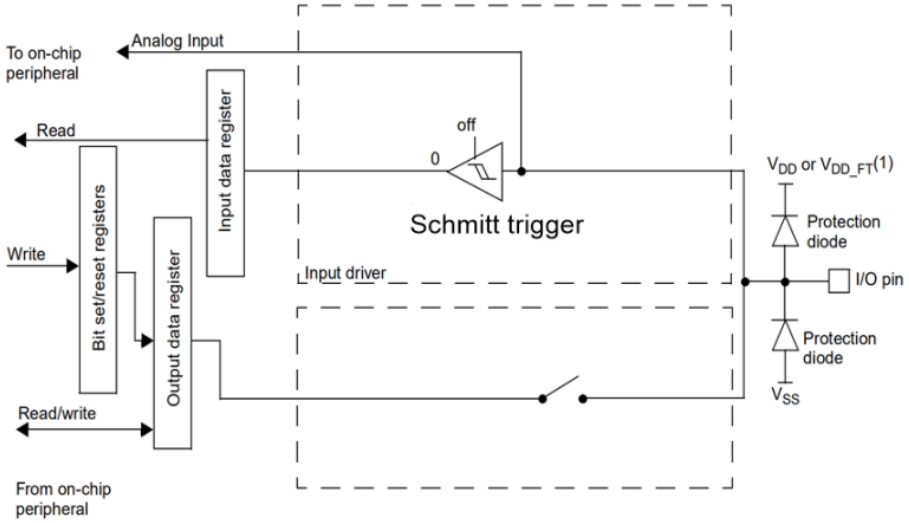


图 6.11 高阻抗-模拟配置

6.4.4. GPIO 寄存器映射

Offset	Name	Description
0x00000	GPIO_DATA	Data
0X00004	GPIO_DATAOUT	Data output latch
0X00010	GPIO_OUTENSET	Output enable set
0X00014	GPIO_OUTENCLR	Output enable clear
0X00018	GPIO_ALTFUNCSET	Alternative function set
0X0001c	GPIO_ALTFUNCCLR	Alternative function set
0X00020	GPIO_INTENSET	Interrupt enable set
0X00024	GPIO_INTENCLR	Interrupt enable clear
0X00028	GPIO_INTTYPESET	Interrupt type set
0X0002c	GPIO_INTTYPECLR	Interrupt type clear
0X00030	GPIO_INTPOLSET	Interrupt polarity set
0X00034	GPIO_INTPOLCLR	Interrupt polarity clear
0X00038	GPIO_INTSTATUS	Interrupt status
0x00040	GPIO_INTBOTHSET	Interrupt type1 set
0x00044	GPIO_INTBOTHCLR	Interrupt type1 clear
0x01000-0x13FC	GPIO_MASKBYTE0	Byte 0 masked access
0x01400-0x17FC	GPIO_MASKBYTE1	Byte 1 masked access
0x01800-0x1BFC	GPIO_MASKBYTE2	Byte 2 masked access
0x01c00-0x1FFC	GPIO_MASKBYTE3	Byte 3 masked access
0x400E002C	GPIO_OE_CTRL	GPIO output enable control
0x400E0034	GPIO_PU_CTRL	GPIO pull up control
0x400E0040	GPIO_ODA_CTRL	GPIO output
0x400E0054	GPIO_IE_CTRL	GPIO input enable control

0x400E00A0	GPIO_ODE_CTRL	GPIO open drain control
0x400E00A8	GPIO_PD_CTRL	GPIO pull down control
0x400E00C4	GPIO_DRV_CTRL_0	GPIO driver control register
0x400E00CC	GPIO_DRV_CTRL_2	GPIO driver control register

DATA address offset: 0x00000

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved
18:0	RW	N/A	DATA	**Read** Sampled at pin. **Write** To data output register.

DATAOUT address offset: 0x00004

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved
18:0	RW	0x0	DATAOUT	Data output Register value: **Read** Current value of data output register. **Write** To data output register.

OUTENSET address offset: 0x00010

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved
18:0	RW	0x0	OUTENSET	Output enable set: **Write** **1** Set the output enable bit. **0** No effect. **Read back** **0** Indicates the signal direction as input. **1** Indicates the signal direction as output

OUTENCLR address offset: 0x00014

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved
18:0	RW	0x0	OUTENCLR	Output enable clear: **Write** **1** Clears the output enable bit. **0** No effect. **Read back** **0** Indicates the signal direction as input. **1** Indicates the signal direction as output

ALTFUNCSET address offset: 0x00018

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved

18:0	RW	0x0	ALTFUNCSET	Alternative function set: **Write** **1** Sets the ALTFUNC bit. **0** No effect. **Read back** **0** For I/O. **1** For an alternate function.

ALTFUNCCLR address offset: 0x0001C

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved
18:0	RW	0x0	ALTFUNCCLR	Alternative function clear: **Write** **1** Clears the ALTFUNC bit. **0** No effect. **Read back** **0** For I/O. **1** For an alternate function

INTENSET address offset: 0x00020

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved
18:0	RW	0x0	INTENSET	Interrupt enable set: **Write** **1** Sets the enable bit. **0** No effect. **Read back** **0** Interrupt disabled. **1** Interrupt enabled

INTENCLR address offset: 0x00024

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved
18:0	RW	0x0	INTENCLR	Interrupt enable clear: **Write** **1** Clear the enable bit. **0** No effect. **Read back** **0** Interrupt disabled. **1** Interrupt enabled

INTTYPESET address offset: 0x00028

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved

18:0	RW	0x0	INTTYPESET	Interrupt type set: **Write** **1** Sets the interrupt type bit. **0** No effect. **Read back** **0** For LOW or HIGH level. **1** For falling edge or rising edge

INTTYPECLR address offset: 0x0002c

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved
18:0	RW	0x0	INTTYPECLR	Interrupt type clear: **Write** **1** Clears the interrupt type bit. **0** No effect. **Read back** **0** For LOW or HIGH level. **1** For falling edge or rising edge.

INTPOLSET address offset: 0x00030

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved
18:0	RW	0x0	INTPOLSET	Polarity-level, edge IRQ configuration: **Write** **1** Sets the interrupt polarity bit. **0** No effect. **Read back** **0** For LOW level or falling edge. **1** For HIGH level or rising edge.

INTPOLCLR address offset: 0x00034

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved
18:0	RW	0x0	INTPOLCLR	Polarity-level, edge IRQ configuration: **Write** **1** Clears the interrupt polarity bit. **0** No effect. **Read back** **0** For LOW level or falling edge. **1** For HIGH level or rising edge.

INTSTATUS address offset: 0x00038

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved

18:0	RW	0x0	INTSTATUS	Write one to clear interrupt request: **Write** IRQ status clear Register. Write: **1** To clear the interrupt request. **0** No effect. **Read back** IRQ status Register

INTTYPE1SET address offset: 0x00040

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved
18:0	RW	0x0	INTTYPE1SET	Interrupt type1 set: **Write** **1** if the corresponding bit of INTTYPE SET is also 1, both falling edge and rising edge triggers interrupt; else no effect **0** No effect. **Read back** **0** double edge detect disable **1** double edge detect enable

INTTYPE1CLR address offset: 0x00044

Bit	R/W	Reset	Name	Description
31:19	N/A	N/A	N/A	reserved
18:0	RW	0x0	INTTYPE1CLR	Interrupt type1 clear: **Write** **1** Clears the interrupt type1 bit. **0** No effect. **Read back** **0** double edge detect disable. **1** double edge detect enable.

MASKBYTE0 address offset: 0x01000-0x013FC

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
7:0	RW	0x0	MASKBYTE0	BYTE0 masked access. Bits[9:2] of the address value are used as enable bit mask for the access:**[7:0]** Data for BYTE0 access, with bits[9:2] of address value used as enable mask for each bit

MASKBYTE1 address offset: 0x01400-0x017FC

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
7:0	RW	0x0	MASKBYTE1	BYTE1 masked access.

				Bits[9:2] of the address value are used as enable bit mask for the access:**[7:0]** Data for BYTE1 access, with bits[9:2] of address value used as enable mask for each bit

MASKBYTE2 address offset: 0x01800-0x01BFC

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
7:0	RW	0x0	MASKBYTE2	BYTE2 masked access. Bits[9:2] of the address value are used as enable bit mask for the access:**[7:0]** Data for BYTE2 access, with bits[9:2] of address value used as enable mask for each bit

MASKBYTE3 address offset: 0x01C00-0x01FFC

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
7:0	RW	0x0	MASKBYTE3	BYTE3 masked access. Bits[9:2] of the address value are used as enable bit mask for the access:**[7:0]** Data for BYTE3 access, with bits[9:2] of address value used as enable mask for each bit

GPIO_OE_CTRL address offset: 0x002C

Bit	R/W	Reset	Name	Description
19	RW	0x0	gpio_oeb_sel	1: gpio_oeb and gpio_out is controlled by reg, 0: gpio_oeb and gpio_out is controlled by hw or gpio_auto_latch_ctrl
18:0	RW	0x7ffff	gpio_oeb_reg[18:0]	Only valid when gpio_oeb_sel=1 GPIO[18:0] output enable control 0: output enable

GPIO_PU_CTRL address offset: 0x0034

Bit	R/W	Reset	Name	Description
30	RW	0x0	gpio_pl_up_4k_17	GPIO17 4K pull-up resistance enable 1: Enable 0: Disable
29	RW	0x0	gpio_pl_up_4k_16	GPIO16 4K pull-up resistance enable 1: Enable 0: Disable
28	RW	0x0	gpio_pl_up_2m_5	GPIO5 2m pull-up resistance enable 1: Enable 0: Disable
27	RW	0x0	gpio_pl_up_2m_4	GPIO4 2m pull-up resistance enable

				1: Enable 0: Disable
26	RW	0x0	gpio_pl_up_2m_1	GPIO1 2m pull-up resistance enable 1: Enable 0: Disable
25	RW	0x0	gpio_pl_up_2m_0	GPIO0 2m pull-up resistance enable 1: Enable 0: Disable
24:0	RW	0x010	gpio_pu_reg[24:0]	GPIO[24:0] pull up control, high active

GPIO_ODA_CTRL address offset: 0x0040

Bit	R/W	Reset	Name	Description
18:0	RW	0x0	gpio_oda_ctrl[18:0]	GPIO[18:0] output data (Only valid when gpio_oeb_sel=1)

GPIO_IE_CTRL address offset: 0x0054

Bit	R/W	Reset	Name	Description
24:19	RW	0x3f	gpio_sf_ctrl	sflash io input enable, high active
18:0	RW	0x1f	gpio_ie_ctrl[18:0]	GPIO[18:0] input enable, high active

GPIO_ODE_CTRL address offset: 0x00a0

Bit	R/W	Reset	Name	Description
19	RW	0x0	gpio_pmu_dbg	1: debug interface output pmu debug signal
18:0	RW	0x0	gpio_ode_reg[18:0]	gpio[18:0] open drain control, high active

GPIO_PD_CTRL address offset: 0x00a8

Bit	R/W	Reset	Name	Description
24:0	RW	0x0	gpio_pd_reg[24:0]	gpio[24:0] pull down control, high active

GPIO_DRV_CTRL_0 address offset: 0x00c4

Bit	R/W	Reset	Name	Description
24:0	RW	0x0	gpio_drv_ctrl_0[24:0]	GPIO driver strength The gpio_drv_ctrl is the pad control interface, and each pad corresponds to one interface, with a total of 25. The gpio_drv_ctrl is a 2bit control signal, gpio_drv_ctrl_0 is the bit0, gpio_drv_ctrl_1 is the bit1. **gpio_drv_ctrl drive power:** 00: minimum 11: maximum (Detailed definitions are given in the Analog document.)

GPIO_DRV_CTRL_2 address offset: 0x00cc

Bit	R/W	Reset	Name	Description
24:0	RW	0x1ffffff	gpio_drv_ctrl_1[24:0]	GPIO driver strength The gpio_drv_ctrl is the pad control interface, and each pad corresponds to one interface, with a total of 25. The gpio_drv_ctrl is a 2bit control signal, gpio_drv_ctrl_0 is the bit0, gpio_drv_ctrl_1 is the bit1. **gpio_drv_ctrl drive power:** 00: minimum 11: maximum (Detailed definitions are given in the Analog document.)

GPIO_ODA_CTRL address offset: 0x400E0040

Bit	R/W	Reset	Name	Description
18:0	RW	0x0	gpio_oda_ctrl[18:0]	GPIO[18:0] output data (Only valid when gpio_oeb_sel=1)

GPIO_IE_CTRL address offset: 0x400E0054

Bit	R/W	Reset	Name	Description
18:0	RW	0x1f	gpio_ie_ctrl[18:0]	GPIO[18:0] input enable, high active

GPIO_ODE_CTRL address offset: 0x400E00a0

Bit	R/W	Reset	Name	Description
18:0	RW	0x0	gpio_ode_reg[18:0]	gpio[18:0] open drain control, high active

GPIO_PD_CTRL address offset: 0x400E00a8

Bit	R/W	Reset	Name	Description
18:0	RW	0x0	gpio_pd_reg[18:0]	gpio[18:0] pull down control, high active

GPIO_DRV_CTRL_0 address offset: 0x400E00c4

Bit	R/W	Reset	Name	Description
18:0	RW	0x0	gpio_drv_ctrl_0[18:0]	GPIO driver strength

GPIO_DRV_CTRL_2 address offset: 0x400E00cc

Bit	R/W	Reset	Name	Description
18:0	RW	0x7ffff	gpio_drv_ctrl_1[18:0]	GPIO driver strength

6.5. UART1

6.5.1. 简介

UART 是一个可编程的通用异步接收/发送器（UART）。这个组件符合 AMBA 2.0 标准的高级外设总线（APB）从设备。

UART 是根据行业标准 16550 模型设计的。然而，寄存器地址空间已经重新和 32 位数据对齐，以适应 APB 总线实现。UART 用于与外设、调制解调器（数据通信设备，DCE）等进行串行通信。数据由主设备（CPU）通过 APB 总线写入 UART，并转换为串行形式传输到目标设备。UART 还接收串行数据，并存储起来供主设备（CPU）读取。

UART 包含控制字符长度、波特率、奇偶校验和中断产生的寄存器。尽管 UART 只有一个中断输出信号（intr），但有几种优先级不同的中断类型可能导致其断言。每种中断类型都可以通过控制寄存器单独启用/禁用。

6.5.2. 主要特性

- AMBA APB interface allows easy integration into AMBA SOC implementations
- Configurable parameters for the following:
  - APB data bus widths of 8, 16 and 32
  - Additional DMA interface signals for compatibility with design ware DMA interface
  - DMA interface signal polarity
  - Transmit and receive FIFO depths of none, 16, 32, 64,…,2048
  - Use of two clocks (pclk and sclk) instead of one (pclk)
  - IrDA 1.0 SIR mode support with up to 115.2 Kbaud data rate and a pulse duration (width) as follows: width = 3/16 × bit period as specified in the IrDA physical layer specification
  - IrDA 1.0 SIR low-power reception capabilities
  - Baud clock reference output signal
  - Clock gate enable output(s) used to indicate that the TX and RX pipeline is clear (no data) and no activity has occurred for more than one character time, so clocks may be gated
  - FIFO access mode (for FIFO testing) so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master
  - Additional FIFO status registers
  - Shadow registers to reduce software overhead and also include a software programmable reset
  - Auto Flow Control mode as specified in the 16750 standard
  - Loop back mode that enables greater testing of Modem Control and Auto Flow Control features (Loop back support in IrDA SIR mode is available)
  - Transmitter Holding Register Empty (THRE) interrupt mode

- Busy functionality
- Ability to set some configuration parameters in instantiation
- Configuration identification registers present
- Functionality based on the 16550 industry standard, as follows:
  - Programmable character properties, such as number of data bits per character (5-8), optional
    parity bit (with odd or even select) and number of stop bits (1, 1.5 or 2)
  - Line break generation and detection
  - DMA signaling with two programmable modes
  - Prioritized interrupt identification
- Programmable FIFO enable/disable
- Programmable serial data baud rate as calculated by the following:
  - baud rate = (serial clock frequency)/(16× divisor)
- External read enable signal for RAM wake-up when using external RAMs
- Modem and status lines are independently controlled
- Complete RTL version
- Separate system resets for each clock domain to prevent metastability

6.5.3. 功能描述

6.5.3.1. UART(RS232) 串行通讯协议

因为 UART 与选定设备之间的串行通信是异步的，所以会在串行数据中添加额外的位（起始位和停止位）来指示开始和结束。利用这些位可以使两个设备同步。这种串行数据结构——伴随着起始位和停止位——被称为一个字符，如下图所示。

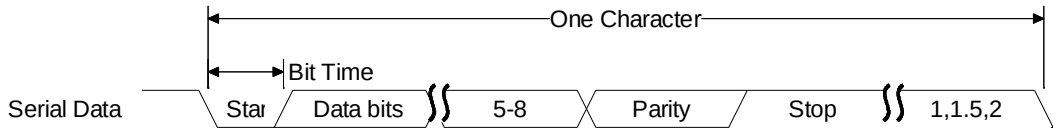


Figure 6.12 Serial Data Format

可以在串行字符中添加一个额外的奇偶校验位。这个位出现在最后一个数据位之后，在停止位之前，以便于 UART 能够对接收到的数据执行简单的错误检查。

UART 线控制寄存器（“LINE_CTRL”寄存器）用于控制串行字符的特性。数据字中的各个位在起始位之后发送，从最低有效位（LSB）开始。之后是可选的奇偶校验位，然后是停止位，停止位可以是 1 位、1.5 位或 2 位。

传输中的所有位（除了使用 1.5 位停止位时的半停止位）都以完全相同的时间持续传输。这被称为位周期或位时间。一个位时间等于 16 个波特时钟。为了确保线路稳定性，接收器在检测到起始位后，大约在位时间的中间点对串行输入数据进行采样。由于每个位传输的确切波特时钟数是已知的，计算采样中点并不困难，即在起始位的中间采样点之后的每 16 个波特时钟。下图显示了串行字符中前几个位的采样点。

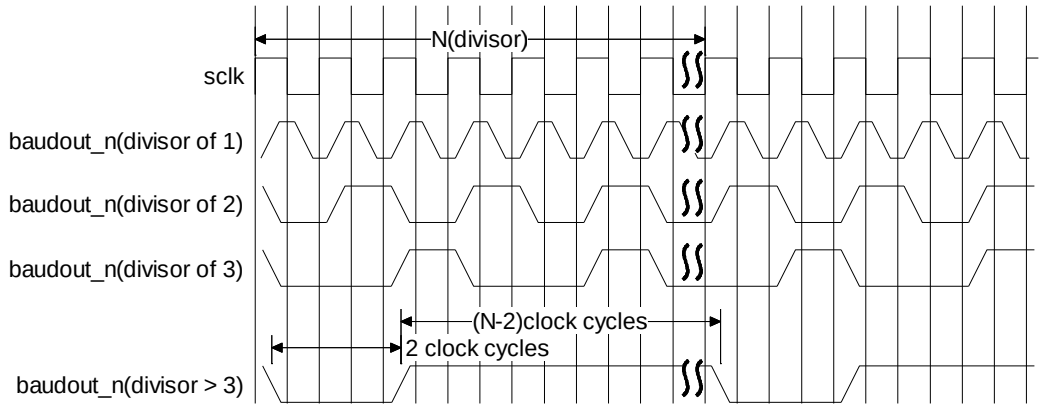


Figure 6.13 Receiver Serial Data Sample Points

作为 16550 标准的一部分，提供了一个可选的波特率时钟参考输出信号（baudout_n），以向需要此信息的接收设备提供定时信息。串行通信接口的波特率由串行时钟（sclk 或在单一时钟实现中的 pclk）和除数锁存寄存器（DLH 和 DLL）控制。下图显示了不同除数值下 baudout_n 输出的时序图。

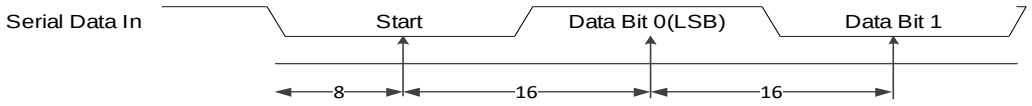


Figure 6.14 Baud Clock Reference Timing Diagram

6.5.3.2. IrDA 1.0 SIR 协议

红外数据协会（IrDA）1.0 串行红外（SIR）模式支持使用红外辐射作为传输介质与远程设备进行双向数据通信。IrDA 1.0 SIR 模式规定了最大波特率为 115.2K 波特。

6.5.3.3. 中断

每当启用并激活了几个优先级中断类型中的任意一个时，就会断言 UART 中断输出信号（intr）。以下中断类型可以通过 IER 寄存器启用:

- 以下是可配置参数：
- 接收器错误
- 接收器数据可用
- 字符超时（仅在 FIFO 模式下）
- 在/低于阈值时传输器保持寄存器为空（在可编程 THRE 中断模式下）
- 调制解调器状态
- 忙碌检测指示

6.5.4. UART 寄存器映射

Offset	RW	Reset	Name	Description
0x0000	R	0x0	RBR	Receive Buffer Register Dependencies:LCR[7]bit = 0
	W	0x0	THR	Transmit Holding Register Dependencies:LCR[7]bit = 0
	R/W	0x0	DLL	Divisor Latch (Low) Dependencies: LCR[7] bit = 1
0x0004	R/W	0x0	DLH	Divisor Latch (High) Dependencies: LCR[7] bit = 1
	R/W	0x0	IER	Interrupt Enable Register Dependencies: LCR[7] bit = 0
0x0008	R	0x01	IIR	Interrupt Identification Register
	W	0x0	FCR	FIFO Control Register
0x000C	R/W	0x0	LCR	Line Control Register
0x0010	R/W	0x0	MCR	Modem Control Register
0x0014	R	0x60	LSR	Line Status Register
0x0018	R	0x0	MSR	Modem Status Register
0X001C	R/W	0x0	SCR	Scratchpad Register
0x0020	R/W	0x0	LPDLL	Low Power Divisor Latch (Low) Register
0x0024	R/W	0x0	LPDLH	Low Power Divisor Latch (High) Register
0x0028			ISO7816_CTRL0	ISO7816 Control Register (only valid in UART1)
0x002C			ISO7816_CTRL1	ISO7816 Control Register (only valid in UART1)
0x0030--0x006C	R	0x0	SRBR	Shadow Receive Buffer Register Dependencies: LCR[7] bit = 0
	W	0x0	STHR	Shadow Transmit Holding Register Dependencies: LCR[7] bit = 0
0x0070	R/W	0x0	FAR	FIFO Access Register
0x0074	R	0x0	TFR	Transmit FIFO Read
0x0078	W	0x0	RFW	Receive FIFO Write
0x007C	R	0x6	USR	UART Status Register
0x0080	R	0x0	TFL	Transmit FIFO Level Width: FIFO_ADDR_WIDTH + 1
0x0084	R	0x0	RFL	Receive FIFO Level Width: FIFO_ADDR_WIDTH + 1

0x0088	W	0x0	SRR	Software Reset Register
0x008C	R/W	0x0	SRTS	Shadow Request to Send
0x0090	R/W	0x0	SBCR	Shadow Break Control Register
0x0094	R/W	0x0	SDMAM	Shadow DMA Mode
0x0098	R/W	0x0	SFE	Shadow FIFO Enable
0x009C	R/W	0x0	SRT	Shadow RCVR Trigger
0x00A0	R/W	0x0	STET	Shadow TX Empty Trigger
0x00A4	R/W	0x0	HTX	Halt TX
0x00A8	W	0x0	DMASA	DMA Software Acknowledge
0x00AC-0x00F0				
0x00F4	R	Configuration-dependent	CPR	Component Parameter Register
0x00F8	R	See the Releases table in the AMBA 2 release notes.	UCV	UART Component Version
0x00FC	R	0x44570110	CTR	Component Type Register

RBR address offset: 0x0000

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7:0	R	0x0	RBR	Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LCR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an over-run error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an overrun error occurs.

THR address offset: 0x0000

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved

7:0	W	0x0	THR	Data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] = 0) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

DLH address offset: 0x0004

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7:0	RW	0x0	DLH	Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design (CLOCK_MODE == Enabled)) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest uart clock should be allowed to pass before transmitting or receiving data.

DLL address offset: 0x0000

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7:0	RW	0x0	DLL	Lower 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design

				(CLOCK_MODE == Enabled)) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLL is set, at least 8 clock cycles of the slowest uart clock should be allowed to pass before transmitting or receiving data.

IER address offset: 0x0004

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7	RW	0x0	PTIME	This is used to enable/disable the generation of THRE Interrupt 0 = disabled 1 = enabled
6:4	N/A	0x0	N/A	reserved
3	RW	0x0	EDSSI	Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0 = disabled 1 = enabled
2	RW	0x0	ELSI	Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt 0 = disabled 1 = enabled
1	RW	0x0	ETBEI	Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 = disabled 1 = enabled
0	RW	0x0	ERBFI	Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled

IIR address offset: 0x0008

Bit	R/W	Reset	Name	Description

31:8	N/A	0x0	N/A	reserved
7:6	R	0x0	FIFOSE	FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 00 = disabled 11 = enabled
5:4	N/A	0x0	N/A	reserved
3:0	R	0x1	IID	Interrupt ID. This indicates the highest priority pending interrupt which can be one of the following types 0000 = modem status 0001 = no interrupt pending 0010 = THR empty 0100 = received data available 0110 = receiver line status 0111 = busy detect 1100 = character timeout

FCR address offset: 0x0008

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7:6	W	0x0	RT	RCVR Trigger. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. For details on DMA support, The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO/4 full 10 = FIFO/2 full 11 = FIFO 2 less than full
5:4	W	0x0	TET	TX Empty Trigger. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. For details on DMA support, The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO/4 full 11 = FIFO/2 full
3	W	0x0	DMAM	DMA Mode. For details on DMA support, 0 = mode 0; 1 = mode 1
2	W	0x0	XFIFOR	XMIT FIFO Reset. This resets the control portion of the transmit FIFO and treats the FIFO as empty.

				This bit is 'self-clearing'.
1	W	0x0	RFIFOR	RCVR FIFO Reset. This resets the control portion of the receive FIFO and treats the FIFO as empty. This bit is 'self-clearing'.
0	W	0x0	FIFOE	FIFO Enable. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs.Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset.

LCR address offset: 0x000C

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7	RW	0x0	DLAB	Divisor Latch Access Bit. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART.
6	RW	0x0	BC	Break Control Bit.This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loop back Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE == Enabled and active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loop back Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.
5	N/A	0x0	N/A	reserved
4	RW	0x0	EPS	Even Parity Select.If UART_16550_COMPATIBLE == NO, then write able only when UART is not busy (USR[0] is zero); otherwise always writable, always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked. 0 = mode 0 1 = mode 1
3	RW	0x0	PEN	Parity Enable.If UART_16550_COMPATIBLE == NO, then writeable only when UART is not busy (USR[0] is zero); otherwise always writable, always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively 0 = parity disabled 1 = parity enabled

2	RW	0x0	STOP	Number of stop bits. If UART_16550_COMPATIBLE == NO, then write able only when UART is not busy (USR[0] is zero); otherwise always writable, always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 0 = 1 stop bit 1 = 1.5 stop bits (DLS==0) 1 = 2 stop bits (DLS!=0)
1:0	RW	0x0	DLS	Data Length Select This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits

MCR address offset: 0x0010

Bit	R/W	Reset	Name	Description
31:7	N/A	0x0	N/A	reserved
6	RW	0x0	SIRE	SIR Mode Enable. 0 = disable 1 = enable
5	RW	0x0	AFCE	Auto Flow Control Enable. 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled
4	RW	0x0	LB	Loop Back Bit. This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled or not active, MCR[6] set to zero), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loop back mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally.

				If operating in infrared mode (SIR_MODE == Enabled AND active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.
3	RW	0x0	OUT2	OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 0 = out2_n de-asserted (logic 1) 1 = out2_n asserted (logic 0)
2	RW	0x0	OUT1	OUT1. This is used to directly control the user-designated Output1 (out1_n) output. The value written to this location is inverted and driven out on out1_n, that is: 0 = out1_n de-asserted (logic 1) 1 = out1_n asserted (logic 0)
1	RW	0x0	RTS	Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high.In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. Note that in Loop back mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.
0	RW	0x0	DTR	Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: 0 = dtr_n de-asserted (logic 1) 1 = dtr_n asserted (logic 0)

LSR address offset: 0x0014

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7	R	0x0	RFE	Receiver FIFO Error bit.This bit is only relevant when

				FIFO_MODE != NONE AND FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 0 = no error in RX FIFO 1 = error in RX FIFO
6	R	0x1	TEMT	Transmitter Empty bit. If in FIFO mode (FIFO_MODE != NONE) and FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If in non-FIFO mode or FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.
5	R	0x1	THRE	Transmit Holding Register Empty bit.If THRE_MODE_USER == Disabled or THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If THRE_MODE_USER == Enabled AND FIFO_MODE != NONE and both modes are active (IER[7] set to one and FCR[0] set to one respectively), the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.
4	R	0x0	BI	Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. If in UART mode (SIR_MODE == Disabled), it is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits. If in infrared mode (SIR_MODE == Enabled), it is set whenever the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of start time + data bits + parity + stop bits. A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART. In the FIFO mode, the character associated with the break condition is carried through the FIFO and is

				revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read.
3	R	0x0	FE	Framing Error bit.This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no framing error 1 =framing error
2	R	0x0	PE	Parity Error bit.This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no parity error 1 = parity error
1	R	0x0	OE	Overrun error bit.This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The

				data in the FIFO is retained and the data in the receive shift register is lost. 0 = no overrun error 1 = overrun error
0	R	0x0	DR	Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready

MSR address offset: 0x0018

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7	R	0x0	DCD	Data Carrier Detect.This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set. 0 = dcd_n input is de-asserted (logic 1) 1 = dcd_n input is asserted (logic 0)
6	R	0x0	RI	Ring Indicator. This is used to indicate the current state of the modem control line ri_n. This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by the modem or data set. 0 = ri_n input is de-asserted (logic 1) 1 = ri_n input is asserted (logic 0)
5	R	0x0	DSR	Data Set Ready.This is used to indicate the current state of the modem control line dsr_n. This bit is the complement of dsr_n. When the Data Set Ready input (dsr_n) is asserted it is an indication that the modem or data set is ready to establish communications with the OM_uart. 0 = dsr_n input is de-asserted (logic 1) 1 = dsr_n input is asserted (logic 0)
4	R	0x0	CTS	Clear to Send.This is used to indicate the current state of the modem control line cts_n. This bit is the complement of cts_n. When the Clear to Send input (cts_n) is asserted it is an indication that the modem or data set is ready to exchange data with the OM_uart. 0 = cts_n input is de-asserted (logic 1) 1 = cts_n input is asserted (logic 0)
3	R	0x0	DDCD	Delta Data Carrier Detect.This is used to indicate that

				the modem control line dcd_n has changed since the last time the MSR was read. 0 = no change on dcd_n since last read of MSR 1 = change on dcd_n since last read of MSR Reading the MSR clears the DDCD bit. In Loop back Mode (MCR[4] = 1), DDCD reflects changes on MCR[3] (Out2). Note, if the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted.
2	R	0x0	TERI	Trailing Edge of Ring Indicator.This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read. 0 = no change on ri_n since last read of MSR 1 = change on ri_n since last read of MSR
1	R	0x0	DDSR	Delta Data Set Ready.This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read. 0 = no change on dsr_n since last read of MSR 1 = change on dsr_n since last read of MSR Reading the MSR clears the DDSR bit. In Loop back Mode (MCR[4] = 1), DDSR reflects changes on MCR[0] (DTR). Note, if the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDSR bit is set when the reset is removed if the dsr_n signal remains asserted.
0	R	0x0	DCTS	Delta Clear to Send.This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read. 0 = no change on cts_n since last read of MSR 1 = change on cts_n since last read of MSR Reading the MSR clears the DCTS bit. In Loop back Mode (MCR[4] = 1), DCTS reflects changes on MCR[1] (RTS). Note, if the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise), then the DCTS bit is set when the reset is removed if the cts_n signal remains asserted.

SCR address offset: 0x001C

Bit	R/W	Reset	Name	Description

31:8	N/A	0x0	N/A	Reserved and read as zero
7:0	RW	0x0	Scratchpad Register	This register is for programmers to use as a temporary storage space. It has no defined purpose in the OM_uart.

LPDLL address offset: 0x0020

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	Reserved
7:0	RW	0x0	LPDLL	This register makes up the lower 8-bits of a 16-bit, read/write, Low Power Divisor Latch register that contains the baud rate divisor for the UART,which must give a baud rate of 115.2K. This is required for SIR Low Power (minimum pulse width) detection at the receiver. If UART_16550_COMPATIBLE == No, then this register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero); otherwise this register may be accessed only when the DLAB bit (LCR[7]) is set. The output low-power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: Low power baud rate = (serial clock frequency)/(16* divisor) Therefore, a divisor must be selected to give a baud rate of 115.2K. NOTE: When the Low Power Divisor Latch registers (LPDLL and LPDLH) are set to 0, the low-power baud clock is disabled and no low-power pulse detection (or any pulse detection) occurs at the receiver. Also, once the LPDLL is set, at least eight clock cycles of the slowest uart clock should be allowed to pass before transmitting or receiving data.

LPDLH address offset: 0x0024

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7:0	RW	0x0	LPDLL	This register makes up the upper 8-bits of a 16-bit, read/write, Low Power Divisor Latch register that contains the baud rate divisor for the UART,which must give a baud rate of 115.2K. This is required for SIR Low Power (minimum pulse width) detection at the receiver. If UART_16550_COMPATIBLE == No, then this register may only be accessed when the

				DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero); otherwise this register may be accessed only when the DLAB bit (LCR[7]) is set. The output low-power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: Low power baud rate = (serial clock frequency)/(16* divisor) Therefore, a divisor must be selected to give a baud rate of 115.2K. NOTE: When the Low Power Divisor Latch registers (LPDLL and LPDLH) are set to 0, the low-power baud clock is disabled and no low-power pulse detection (or any pulse detection) occurs at the receiver. Also, once the LPDLH is set, at least eight clock cycles of the slowest uart clock should be allowed to pass before transmitting or receiving data.

FAR address offset: 0x0070

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	Reserved and read as zero
7:0	RW	0x0	FIFO Access Register	Writes have no effect when FIFO_ACCESS == No, always readable. This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are implemented and enabled. When FIFOs are not implemented or not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFOs are treated as empty.

ISO7816_CTRL0 address offset: 0x0028

Bit	R/W	Reset	Name	Description
31:13	N/A	0x0	N/A	reserved
12	R	0x0	tx_done	TX is done
11:4	RW	0x0	sample_dly	sample_dly is used to adjust the sample timing of SIN

3	RW	0x0	retrans_en	parity error re-trans enable
2	RW	0x0	trx_oen	0: TX 1: RX
1	RW	0x0	nack_enable	noack is enable
0	RW	0x0	iso7816_en	ISO7816 is enable

ISO7816_CTRL1 address offset: 0x002C

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
15:8	R	0x0	tx_perr_cnt	tx parity error counter
7:0	R	0x0	rx_perr_cnt	rx parity error counter

SRBR address offset: 0x0030--0x006C

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7:0	R	0x0	SRBR	This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs.

STHR address offset: 0x0030--0x006C

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7:0	W	0x0	STHR	This is a shadow register for the THR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared

				output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

FAR address offset: 0x0070

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	Reserved and read as zero
7:0	RW	0x0	FIFO Access Register	Writes have no effect when FIFO_ACCESS == No, always readable. This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are implemented and enabled. When FIFOs are not implemented or not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFOs are treated as empty.

TFR address offset: 0x0074

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	Reserved and read as zero
7:0	R	0x0	Transmit FIFO Read	Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, reading this register gives the data at the top of

				the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO. When FIFOs are not implemented or not enabled, reading this register gives the data in the THR.

RFW address offset: 0x0078

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	Reserved and read as zero
9	W	0x0	RFFE	Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, this bit is used to write framing error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write framing error detection information to the RBR.
8	W	0x0	RFPE	Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, this bit is used to write parity error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write parity error detection information to the RBR.
7:0	W	0x0	RFWD	Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs are not implemented or not enabled, the data that is written to the RFWD is pushed into the RBR.

USR address offset: 0x007C

Bit	R/W	Reset	Name	Description
31:5	N/A	0x0	N/A	reserved
4	R	0x0	RFF	Receive FIFO Full.This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO is completely full. 0 = Receive FIFO not full 1 = Receive FIFO Full

3	R	0x0	RFNE	Receive FIFO Not Empty.This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty
2	R	0x1	TFE	Transmit FIFO Empty.This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty
1	R	0x1	TFNF	Transmit FIFO Not Full.This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO in not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full
0	R	0x0	BUSY	UART Busy.This bit is valid only when UART_16550_COMPATIBLE == NO and indicates that a serial transfer is in progress, ; when cleared, indicates that the uart is idle or inactive. 0 = uart is idle or inactive 1 = uart is busy (actively transferring data)

TFL address offset: 0x0080

Bit	R/W	Reset	Name	Description
31:5	N/A	0x0	N/A	reserved
4:0	R	0x0	TFL	Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO.

RFL address offset: 0x0084

Bit	R/W	Reset	Name	Description
31:5	N/A	0x0	N/A	reserved
4:0	R	0x0	RFL	Receive FIFO Level. This is indicates the number of data entries in the receive FIFO.

SRR address offset: 0x0088

Bit	R/W	Reset	Name	Description
31:3	N/A	0x0	N/A	reserved
2	W	0x0	XFR	XMIT FIFO Reset.This is a shadow register for the XMIT FIFO Reset bit (FCR[2]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the transmit FIFO. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX

				request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
1	W	0x0	RFR	RCVR FIFO Reset.This is a shadow register for the RCVR FIFO Reset bit (FCR[1]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the receive FIFO This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
0	W	0x0	UR	UART Reset. This asynchronously resets the uart and synchronously removes the reset assertion.For a two clock implementation both pclk and sclk domains are reset.

SRTS address offset: 0x008C

Bit	R/W	Reset	Name	Description
31:1	N/A	0x0	N/A	reserved
0	RW	0x0	SRTS	Shadow Request to Send.This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the uart is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] = 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] = 1) and FIFOs enable (FCR[0] = 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). Note that in Loopback mode (MCR[4] = 1), the rts_n output is held inactive-high while the value of this location is internally looped back to an input.

SBCR address offset: 0x0090

Bit	R/W	Reset	Name	Description
31:1	N/A	0x0	N/A	reserved
0	RW	0x0	SBCR	Shadow Break Control Bit.This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE == Enabled and active (MCR[6] = 1) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver.

SDMAM address offset: 0x0094

Bit	R/W	Reset	Name	Description
31:1	N/A	0x0	N/A	reserved
0	RW	0x0	SDMAM	Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the DMA Mode bit gets updated. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA == NO). 0 = mode 0 1 = mode 1

SFE address offset: 0x0098

Bit	R/W	Reset	Name	Description
31:1	N/A	0x0	N/A	reserved
0	RW	0x0	SFE	Shadow FIFO Enable.This is a shadow register for the FIFO enable bit (FCR[0]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the FIFO enable bit gets updated.This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. If this bit is set to zero

				(disabled) after being enabled then both the XMIT and RCVR controller portion of FIFOs are reset.

SRT address offset: 0x009C

Bit	R/W	Reset	Name	Description
31:2	N/A	0x0	N/A	reserved
1:0	RW	0x0	SRT	Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]).This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the RCVR trigger bit gets updated. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. It also determines when the dma_rx_req_n signal is asserted when DMA Mode (FCR[3]) = 1. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO ¼ full 10 = FIFO ½ full 11 = FIFO 2 less than full

STET address offset: 0x00A0

Bit	R/W	Reset	Name	Description
31:2	N/A	0x0	N/A	reserved
1:0	RW	0x0	STET	Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the TX empty trigger bit gets updated. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO ¼ full 11 = FIFO ½ full Dependencies: Writes have no effect when THRE_MODE_USER = = Disabled.

HTX address offset: 0x00A4

Bit	R/W	Reset	Name	Description
31:1	N/A	0x0	N/A	reserved
0	RW	0x0	HTX	This register is used to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0 = Halt TX disabled 1 = Halt TX enabled Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation. Dependencies: Writes have no effect when FIFO_MODE == None.

DMASA address offset: 0x00A8

Bit	R/W	Reset	Name	Description
31:1	N/A	0x0	N/A	reserved
0	W	0x0	DMASA	This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the uart should clear its request. This causes the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.

CPR address offset: 0x00F4

Bit	R/W	Reset	Name	Description
31:24	N/A	0x0	N/A	Reserved and read as zero
23:16	R	0x0	FIFO_MODE	0x00 = 0 0x01 = 16 0x02 = 32 to 0x80 = 2048 0x81- 0xff = reserved
15:14	R	0x0	N/A	Reserved and read as zero
13	R	0x0	DMA_EXTRA	0 = FALSE 1 = TRUE
12	R	0x0	UART_ADD_ENCODED_PARAMS	0 = FALSE 1 = TRUE
11	R	0x0	SHADOW	0 = FALSE 1 = TRUE
10	R	0x0	FIFO_STAT	0 = FALSE 1 = TRUE
9	R	0x0	FIFO_ACCESS	0 = FALSE

				1 = TRUE
8	R	0x0	ADDITIONAL_FEAT	0 = FALSE 1 = TRUE
7	R	0x0	SIR_LP_MODE	0 = FALSE 1 = TRUE
6	R	0x0	SIR_MODE	0 = FALSE 1 = TRUE
5	R	0x0	THRE_MODE	0 = FALSE 1 = TRUE
4	R	0x0	AFCE_MODE	0 = FALSE 1 = TRUE
3:2	N/A	0x0	N/A	Reserved and read as zero
1:0	R	0x0	APB_DATA_WIDTH	00 = 8 bits 01 = 16 bits 10 = 32 bits 11 = reserved

UCV address offset: 0x00F8

Bit	R/W	Reset	Name	Description
31:0	R	See the releases table in the AMBA 2 release notes.	UART Component Version	ASCII value for each number in the version, followed by *. For example 32_30_31_2A represents the version 2.01*

CTR address offset: 0x00FC

Bit	R/W	Reset	Name	Description
31:0	R	0x44570110	Peripheral ID	This register contains the peripherals identification code.

6.6. UART0/UART2

6.6.1. 简介

串行通信接口（UART）提供了一个灵活的全双工同步/异步接收/发送器。它可以在三种异步模式下运行。UART 可以在接收寄存器中保留新数据，直到第二次传输完成。

6.6.2. 操作模式

6.6.2.1. 模式 1

在模式 1 下，uart 作为异步发送/接收器运行，具有 8 位数据位和可编程波特率。根据 BAUD_RATE_CTRL_0 的 baud_rate_reg 设置或通过 TIML_CTRL 和 TIMH_CTRL 寄存器设置 uart 定时器寄存器波特率。此外，通过使用 BAUD_RATE_CTRL_1 寄存器的 Baud_rate_sel 位，波特率可以加倍。

通过向 THR 寄存器写入数据来启动传输。o_uart_sda 引脚输出数据。首先传输的是起始位（始终为 0），然后是 8 位数据，之后传输停止位（始终为 1）。

i_uart_sda 输入数据。当接收开始时，uart 与检测到的 i_uart_sda 引脚上的下降沿同步。接收完成后，输入数据可在 RBR 寄存器中获得，停止位的值可在 UART_CTRL 寄存器的 stop_bit_sel 中获得。在接收过程中，应保持 stop_bit_sel 和 RBR。

6.6.2.2. 模式 2

在模式 2 下，uart 作为异步发送/接收器运行，数据位为 9 位，波特率固定为 uart_clk/32 或 uart_clk/64，具体取决于 BAUD_RATE_CTRL_1 寄存器的 Baud_rate_sel 位的设置。通过向 THR 寄存器写入数据来启动传输。o_uart_sda 引脚输出数据。首先传输的是起始位（始终为 0），然后是 9 位数据位，其中第 9 位来自 UART_CTRL 寄存器的 tx_bit8_sel 位，之后传输停止位（始终为 1）。

i_uart_sda 引脚输入数据。当接收开始时，uart 与在 i_uart_sda 检测到的下降沿同步。接收完成后，输入数据可在 RBR 寄存器中获得，第 9 位数据可在 UART_CTRL 寄存器的 rx_bit8_sel 中获得。在接收过程中，RBR 和 rx_bit8_sel 保持不变，直到接收完成。

6.6.2.3. 模式 3

Mode 2 和 Mode 3 之间的唯一区别在于，在 Mode 3 中，可以使用内部波特率发生器或定时器 1 来指定波特率。

在 Mode 3 模式下，uart 作为异步发送/接收器运行，具有 9 位数据位和可编程波特率。根据 baud_rate_ctrl_0 寄存器的 baud_rate_reg 设置，可以使用定时器 1 溢出，或 TIML_CTRL 和 TIMH_CTRL 波特率发生器被用。此外，通过使用 baud_rate_ctrl_1 寄存器的 Baud_rate_sel 位，波特率可以加倍。

通过向 THR 寄存器写入数据来启动传输。o_uart_sda 引脚输出数据。首先传输的是起始位（始终为 0），然后是 9 位数据位，其中第 9 位来自 UART_CTRL 寄存器的 tx_bit8_sel 位，之后传输停止位（始终为 1）。

i_uart_sda 引脚输入数据。当接收开始时，uart 与在 i_uart_sda 引脚检测到的下降沿同步。接收完成后，输入数据可在 RBR 寄存器中获得，第 9 位数据可在 UART_CTRL 寄存器的 rx_bit8_sel 中获得。在接收过程中，RBR 和 rx_bit8_sel 在完成之前保持不变。

6.6.2.4. 串行端口 0 多机通信

串行接口 0 的模式 2 和模式 3 接收 9 位数据的功能可用于多处理器通信。

当 UART_CTRL 寄存器的 mc_en 位被设置时，只有当接收到的第 9 位（UART_CTRL 中的 rx_bit8_sel）为 1 时才会生成接收中断。否则，在接收时不会生成中断。

为了利用这一功能进行多机通信，从机将它们的 mc_en 位设置为 1。主机传输从机的地址，将第 9 位设置为 1，导致所有从机产生接收中断。从机的软件将接收到的字节与它们的网络地址进行比较。如果匹配，被寻址的从处理器将清除其 mc_en 标志，并且主机将剩余的消息以第 9 位设置为 0 的方式传输。其他从处理器保持它们的 mc_en 设置为 1，以便它们忽略主机发送的其余消息。

6.6.3. UART0/UART2 寄存器

Offset	Name	Description
0x0000	CON	Control Register
0x0004	BUF	Data Buffer
0x0040	RELL	Baud Rate Generator Reload Register (low-order byte)
0x0044	RELH	Baud Rate Generator Reload Register (high-order byte)
0x0400	PCON	Power Control Register
0x0404	ADCON	Baud Rate Select register

UART_CTRL address offset: 0x0000

(The CON register controls the function of UART0/UART2.)

Bit	R/W	Reset	Name	Description
31:10	N/A	0x0	N/A	reserved
9	R/W	0x0	TXEN	Tx interrupt enable register
8	R/W	0x0	RXEN	Rx interrupt enable register
7:6	R/W	0x0	MODE	Model select register **01:Mode1**,8-bit uart, band rate=((2^smod)*pclk/64)*(2^10-rel)，rel is the contents of TIM registers (TIMH,TIML) **10:Mode2**,9-bit uart, In mode 2,the UART operates as asynchronous transmitter/receiver with 9 data bits and baud rate fixed to uart_clk/32 or uart_clk /64,

				depending on the setting of Baud_rate_sel bit of the BAUD_RATE_CTRL_1 register. . **11:Mode3**,8-bit uart,band rate variable
5	R/W	0x0	MC_EN	Multiprocessor communication enable control register
4	RW	0x0	SRX_EN	**Serial reception enable** If set HIGH serial reception at uart is enabled. Otherwise serial reception at uart is disabled.
3	RW	0x0	TX_BIT8_SEL	**Transmitter bit 8** This bit is used while transmitting data through uart in Modes 2 and 3. The state of this bit corresponds with the state of the 9th transmitted bit (e.g. parity check or multiprocessor communication). It is controlled by software.
2	R/W	0x0	RX_BIT8_SEL	**Received bit 8** This bit is used while receiving data through UART in Modes 2 and 3. It reflects the state of the 9th received bit. In Mode 1, if multiprocessor communication is enabled (sm20 = 0), this bit is the stop bit that was received
1	R/W	0x0	TX_INT	**Transmit interrupt flag** It indicates completion of a serial transmission at UART. It is at the beginning of a stop bit in all modes. It must be cleared by software.
0	R/W	0x0	RX_INT	**Receive interrupt flag** It is set by hardware after completion of a serial reception at UART. It is in the middle of a stop bit in all modes. It must be cleared by software.

THR_CTRL address offset: 0x0004

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7:0	R/W	0x0	THR	Writing data to this register sets data in serial output buffer and starts the transmission through UART. Reading from the BUF reads data from the serial receive buffer.

TIML address offset: 0x0040

Bit	R/W	Reset	Name	Description

31:8	N/A	0x0	N/A	reserved
7:0	R/W	0xD9	TIML	UART Reload Register is used for UART baud rate generation. Only 10 bits are used. 8 bits from the RELL as lower bits and 2 bits from the TIMH as higher bits.

TIMH address offset: 0x0044

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7:0	R/W	0x03	TIMH	UART Reload Register is used for UART baud rate generation. Only 10 bits are used. 8 bits from the TIML as lower bits and 2 bits from the TIMH as higher bits.

BAUD_RATE_CTRL_1 address offset: 0x0400

Bit	R/W	Reset	Name	Description
31:1	N/A	0x0	N/A	reserved
0	R/W	0x03	SMOD	UART baud rate select (baud rate double)(in mode1 and mode3)

BAUD_RATE_CTRL_0 address offset: 0x0404

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7:0	R/W	0x0	BD	Uart baud rate select (in modes 1 and 3) When 1, additional internal baud rate generator is used(need set to 1)

6.7. SPI

6.7.1. 简介

SPI0 模块可以工作在主模式或从模式，并发送或接收 1 字节的数据。典型的 SPI 传输以 SPI_RDY 引脚下降沿（作为片选）开始，接着是 8 个时钟周期，在此期间，8 位数据通过 SPI_DO 引脚移出，同时 8 位数据通过 SPI_DI 引脚移入，然后 SPI_RDY 引脚变高。SPI 主/从硬件可以编程使用 SPI_RDY 引脚作为双向就绪信号，在主设备可以发出时钟之前，该引脚必须为高。

对于正常的（片选控制的）传输，主设备可以在字节传输后保持 SPI_RDY 线为低（即，发送多个字节）。在传输和空闲期间，可以单独控制 SPI_DO 引脚的驱动状态（驱动/不驱动）。为了允许 3 线 SPI 接口，输入数据可以交替在 SPI_DO 引脚上读取。在主模式下，当写入 SPI_Write_Data 寄存器时，传输 1 字节。在从模式下，传输由主设备控制。当 FIFO 使能位被重置时，写入 SPI0_WDATA 将启用发送数据和接收数据。如果启用了中断，SPI 硬件将在每次传输一个字节后生成中断。当 FIFO 使能位被设置时，在发送器 FIFO 非空时传输字节。接收器 FIFO 接收与写入发送器 FIFO 相同数量的数据。当达到 RCVR FIFO 触发级别或 XMIT FIFO 为空时，SPI 硬件生成中断。

6.7.2. 功能描述

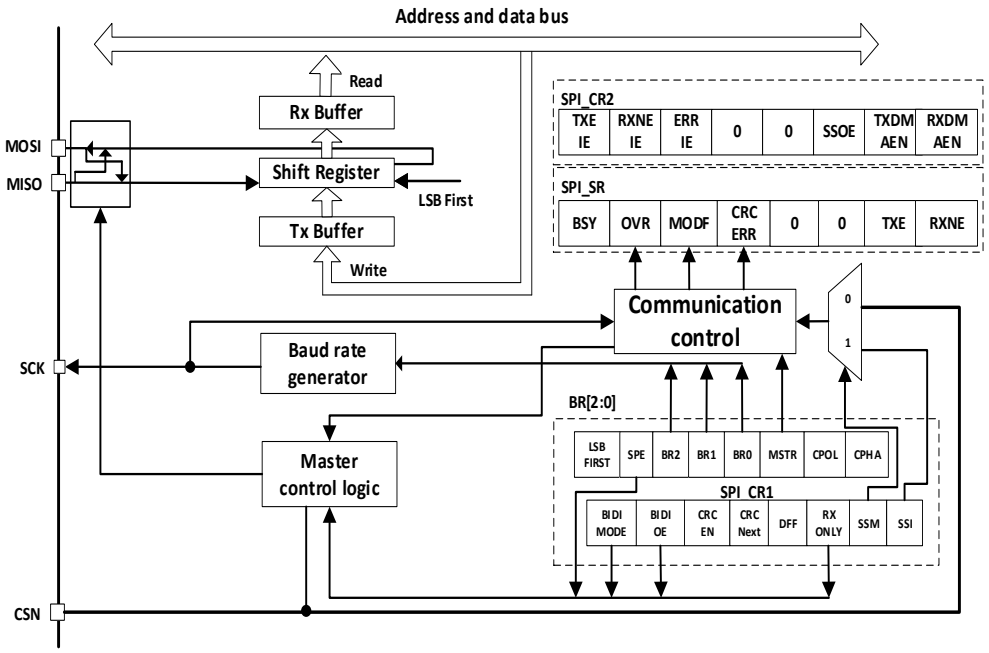


图 6.15 SPI 框图

6.7.2.1. 配置 SPI 从模式

在从机配置中，串行时钟信号通过 SCK 引脚从主设备接收。在 SPI_CTRL 寄存器中设置的位[16]的值不会影响数据传输速率。

注意：建议在主设备发送时钟之前启用 SPI 从机。否则，可能会发生不期望的数据传输。从机的数据寄存器需要在通信时钟的第一个边沿之前或正在进行的通信结束之前准备好。在启用从机和主机之前，必须将通信时钟的极性设置为稳定状态值。

按照以下步骤配置 SPI 从模式:

- 选择 CPOL 和 CPHA 位以定义数据传输和串行时钟之间的四种关系之一。为了正确的数据传输，从设备和主设备中的 CPOL 和 CPHA 位必须以相同的方式配置。
- 帧格式（取决于 SPI_CTRL 寄存器中位[20]的值，是 MSB 优先还是 LSB 优先）必须与主设备相同。

在这种配置中，MOSI 引脚是数据输入，而 MISO 引脚是数据输出。

6.7.2.2. 配置 SPI 主模式

在主配置中，串行时钟由模块生成。

步骤：

- 选择 SPI_CTL[15:0]位来定义串行时钟波特率（参见 SPI_CTL 寄存器）。
- 选择 CPOL 和 CPHA 位以定义数据传输和串行时钟之间的四种关系之一。
- 在 SPI_CTL 寄存器中配置 MSB_FIRST_H 位以定义帧格式。

在这种配置中，MOSI 引脚是数据输出，而 MISO 引脚是数据输入。

6.7.2.3. 状态标志

为应用提供了六个状态标志，以便完全监控 SPI 总线的状态。

Running flag (Run)

当这个标志被设置时，它表明 SPI 是活动的。

SPI RDY IN flag (SRI))

这个标志表示数据是从 SPI_RDY 引脚读取的。

SPI RDY OUT flag (SRO)

这个标志表示数据正在通过 SPI_RDY 引脚驱动。

Wait for RDY flag (WFR)

这个标志表示在读取 1 时等待 SPI_RDY 线变为高电平。

Bit Count flag (BC)

当前正在写入/读取的字节的位数。

SPI INT flag (SI)

SPI 中断状态，写入 1 以清除。

6.7.2.4. 用 DMA 进行 SPI 通讯

为了以最大速度运行，SPI 需要被提供用于传输的数据，并且需要读取 Rx 缓冲区中的接收到的数据以避免溢出。为了便于数据传输，SPI 具有 DMA 功能，实现了简单的请求/确认协议。

当 SPI_DMACR 寄存器中的使能位被启用时，会请求 DMA 访问。必须分别向 TX 和 RX 缓冲区发出请求。

6.7.3. SPI 寄存器映射

Offset	Name	Description
0x0000	SPI_CTRL	SPI Control Register
0x0004	SPI_WDATA	Data transmit to the SPI port
0x0008	SPI_RDATA	Data receive from the SPI port
0x000c	SPI_STAT	SPI status registers
0x0010	DMACR	DMA control registers
0x0014	DMATDLR	DMA tx request level registers
0x0018	DMARDLR	DMA rx request level registers
0x001c	CSNCTRL	CSN control register

SPI_CTRL address offset: 0x0000

Bit	R/W	Reset	Name	Description
31	R/W	0x0	TX_FIFO_ENABLE	Data write to Write Data Register go to 128 byte Transmitter FIFO 1: enable 0: disable
30	R/W	0x0	RX_FIFO_ENABLE	Data read from Read Data register come from 128 byte Receiver FIFO 1: enable 0: disable
29	R/W	0x0	TX_CLR_FIFO	Reset Transmitter FIFO pointers and Byte counters and Overrun status
28	R/W	0x0	RX_CLR_FIFO	Reset Receiver FIFO pointers and Byte counters and Overrun status
27:26	R/W	0x0	RX_TRIGGER_LEVEL	Receiver FIFO Trigger Level: set the trigger level of interrupt 00=8byte 01=32byte

				10=64byte 11=96byte
25	R/W	0x1	INACTIVE_DO_EN	1=SPI_DO pin is high-Z while byte is not being transferred 0=SPI_DO pin is driven while byte is not being transferred
24	R/W	0x0	ACTIVE_DO_EN	1=SPI_DO pin is high-Z while byte is being transferred 0=SPI_DO pin is driven while byte is being transferred
23	R/W	0x0	BIDIRECT_DATA_H	1=Data is written and read on SPI_DO pin 0=Data is written on SPI_DO pin, and read on SPI_DI pin
22	R/W	0x0	USE_RDY_OUT_H	1=Master/Slave use SPI_RDY pin as bidirect Ready line 0=Master/Slave use SPI_RDY pin as SPI chip enable
21	R/W	0x0	INVERT_CLOCK_H	1=Clock is inverted(low when IDEL) 0=Clock is not inverted(high when IDLE)
20	R/W	0x0	MSB_FIRST_H	1=MSB is sent/received first 0=LSB is sent/received first
19	R/W	0x0	SOFT_RESET_H	1=Soft reset SPI hardware, except setup registers 0=Allow SPI to run
18	R/W	0x0	MASTER_CE_AT_END	Level of SPI_RDY(CE) pin after a transfer in master mode
17	R/W	0x0	MODE1_H	1= mode 1(use second clock edge) 0= mode 0 (use first clock edge)
16	R/W	0x0	MASTER_ENABLE_H	1: SPI in master mode 0: SPI in slave mode
15:0	R/W	0x0	CLK_DIVIDER	For Master mode only. Set high and low time of SPI_CLK to (Clk_Divider+1)

SPI_WDATA address offset: 0x0004

Bit	R/W	Reset	Name	Description
31:8	R/W	N/A	N/A	reserved
7:0	R/W	0x0	WRITE_DATA	data to be written out the SPI port

SPI_RDATA address offset: 0x0008

Bit	R/W	Reset	Name	Description
31:8	R	0x0	N/A	reserved
7:0	R	0x0	READ_DATA	data read from the SPI port

SPI_STAT address offset: 0x000c

Bit	R/W	Reset	Name	Description
31	R/W	0x0	SPI_INT	When read,it reflects the SPI interrupt status.When write 1,it dis-asserts the SPI interrupt
30:28	R	0x0	BIT_COUNT	Bit count of current byte being written/read
27	R	0x0	WAIT_FOR_RDY_H	1=SPI is waiting for SPI_RDY line to go high
26	R	0x0	SPI_RDY_OUT	Level being driven on SPI_RDY pin
25	R	0x0	SPI_RDY_IN	Level being driven on SPI_RDY pin
24	R	0x0	SPI_ACTIVE_H	1=SPI transfer is in process
23:16	R/W	0x0	TX_BYTE_CNT	TX FIFO byte count
15:8	R/W	0x0	RX_BYTE_CNT	RX FIFO byte count
7	R/W	0x0	RX_FIFO_TRIG	RX FIFO trigger level reached
6	R/W	0x0	RX_TRIG_INT_EN	This bit enable RX FIFO trigger level interrupt
5	R/W	0x0	TX_EMPTY	TX FIFO empty flag
4	R/W	0x0	TX_EMPTY_INT_EN	This bit enable Transmitter FIFO empty interrupt 1: enable 0: disable
3	R/W	0x0	N/A	reserved
2	R/W	0x0	TX_FIFO_OVERRUN	Overrun Error. Set When transmitter FIFO is full and shift register contains next character.
1	R/W	0x0	RX_FIFO_OVERRUN	RX overrun error flag Overrun Error. Set When receiver FIFO is full and shift register contains next character.
0	R/W	0x0	RX_NOTEMPTY_H	This bit set to one whenever a complete incoming byte has been received. This bit reset to zero by reading all of the data in the Receiver FIFO 1: no empty 0: empty

SPI_ DMACR address offset: 0x0010

Bit	R/W	Reset	Name	Description
312	NA	0x0	N/A	reserved
1	R/W	0x0	TDMAE	transmit DMA Enable, this bit enables/disables the transmit FIFO DMA channel. This bit-field enable the dma tx transfer, while set 1 and the value of **tx_full** is 0, the **dma_tx_single** which indicates the status of SPI DMA TX transfer will be set 1. And **dma_tx_single** will be reset when this bit's value is 0,or any vaule of **dma_tx_ack** and **tx_full** is 1. 0: Transmit DMA disable 1: Transmit DMA enable
0	R/W	0x0	RDMAE	Receive DMA enable.This bit enables/disables the transmit FIFO DMA channel.This bit-field enable the dma tx transfer, while set 1 and the value of **rx_empty** is 0,the **dma_rx_single** which indicates the status of SPI DMA RX transfer will be set 1. And **dma_rx_single** will be reset when this bit's value is 0,or any vaule of **dma_rx_ack** and **rx_empty** is 1. 0: Receive DMA disable 1: Receive DMA enable

SPI_DMATDLR address offset: 0x0014

Bit	R/W	Reset	Name	Description
SPI_TX_ABW-1:0	RW	0x0	DMATDLR	This register is only valid when the SPI is configured with a set of DMA interface signals. When SPI is not configured for DMA operation , this register will not exist and writing to its address will have no effect; reading from its address will return zero. This bit field controls the level at which a DMA request is made by the transmit logic. while **txflr <= dmatdlr, dma_tx_req** while be set 1.

SPI_DMARDLR address offset: 0x0018

Bit	R/W	Reset	Name	Description
SPI_RX_ABW-1:0	RW	0x0	DMARDLR	This register is only valid when the SPI is configured with a set of DMA interface signals. When SPI is not configured for DMA operation , this register will not exist and writing to its address will have no effect; reading from its address will return zero. while **rxflr >= dmardlr + 1, dma_rx_req** while be set 1.

SPIx_CSNCTRL address offset: 0x001c

Bit	R/W	Reset	Name	Description
31:2	N/A	0x0	N/A	reserved
1	RW	0x0	CS_GPO	When the value of cs_mode is 1,the value of **o_spi_mst_csn** is equal to cs_gpo
0	RW	0x0	CS_MODE	0:**o_spi_mst_csn** will be valued by device 1:**o_spi_mst_csn** will be valued by cs_gpo

6.8. TIMER

6.8.1. 简介

定时器包含三个相同的 16 位定时器计数器通道。每个通道都可以独立编程以执行包括频率测量、事件计数、间隔测量、脉冲生成、延迟定时和脉冲宽度调制在内的广泛功能。每个通道驱动一个内部中断信号，该信号可以编程生成处理器中断。

6.8.2. 主要特性

- 16 位向上、向下、向上/下自动重载计数器。
- 16 位可编程分频器（允许在运行中以任意因子将计数器时钟频率分频）
- 多达 2 个独立通道:
  - 输入捕获(仅定时器 1)
  - 输出比较(所有定时器)
  - PWM 产生器 (边沿和中心对齐模式)
  - 单脉冲模式输出
- 具有可编程死区时间的互补输出（每个定时器只有一个通道）。
- 同步电路用于使用外部信号控制定时器，并将多个定时器连接在一起。
- 重复计数器，仅在计数器的给定周期数后更新定时器寄存器。
- 中断输入，将定时器的输出信号置于复位状态或已知状态。
- 在以下事件上产生中断或 DMA:
  - 更新: 计数器溢出/下溢，计数器初始化（通过软件或内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较
  - 中断输入
- 支持增量（正交）编码器和霍尔传感器电路进行定位
- 触发输入作为外部时钟或周期性电流管理

6.8.3. 功能描述

6.8.3.1. 模块框图

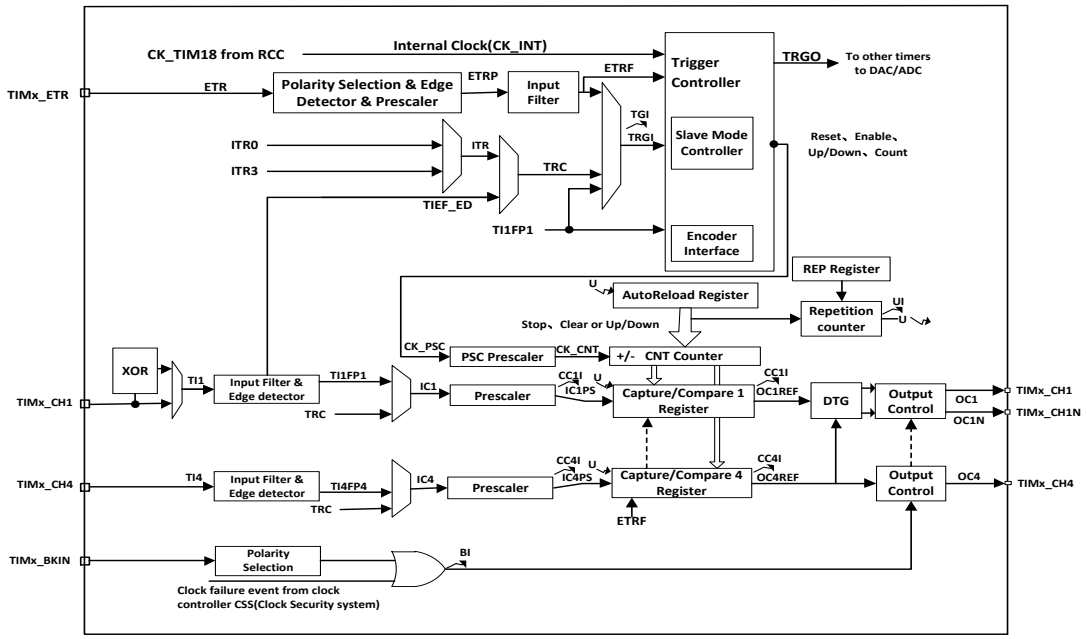


图 6.16 通用定时器描述

注:

中断和 DMA 输出

事件

Reg 根据控制位的设置，在 U 事件期间，预加载寄存器的内容被转移到工作寄存器中

6.8.3.2. 时钟选择

计数器的时钟可以由以下时钟源提供:

- 内部时钟 (CK_INT)
- 外部时钟 1: 外部输入脚
- 外部时钟 2: 外部触发输入 ETR
- 内部触发输入 ETR (ITRx): 使用一个定时器作为另一个定时器的预分频器，例如，你可以配置定时器 1 作为定时器 2 的预分频器。更多细节请参考使用一个定时器作为另一个定时器的预分频器。

内部时钟源 (CK_INT)

如果从模式控制器被禁用（SMS=000），那么 CEN、DIR（在 TIMx_CR1 寄存器中）和 UG 位（在 TIMx_EGR 寄存器中）是实际的控制位，并且只能通过软件进行更改（UG 位除外，它会自动清零）。一旦 CEN 位被写入 1，预分频器就会由内部时钟 CK_INT 进行时钟计数。

外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，选择此模式。计数器可以在选定输入的每个上升沿或下降沿进行计数。

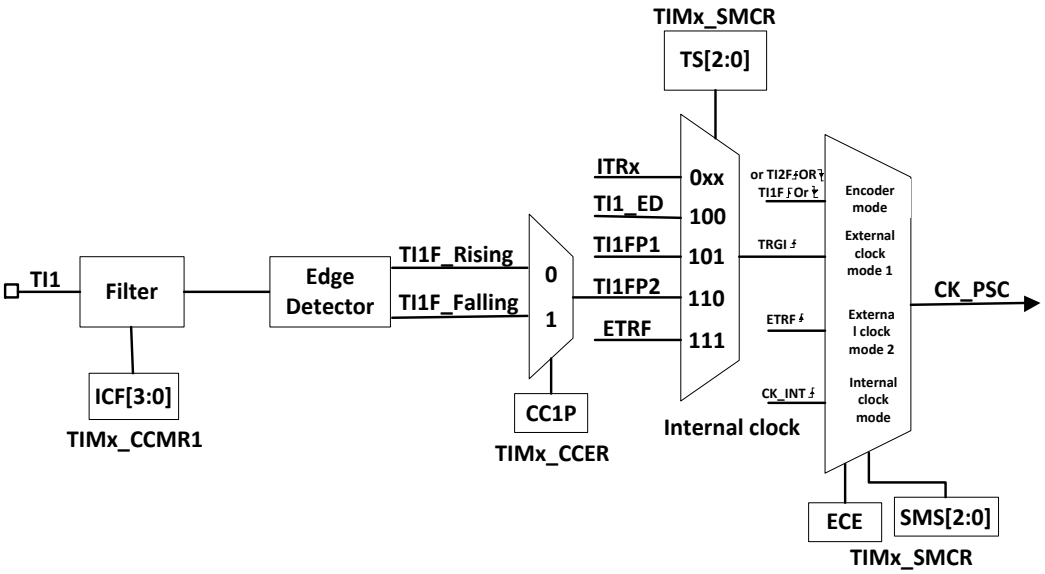


图 6.17 TI1 外部时钟连接实例

例如，要配置向上计数器以响应 TI1 输入上的上升沿进行计数，请按以下步骤:

- 通过在 TIMx_CCMR1 寄存器中写入 CC1S='01'来配置通道 1，使其在 TI1 输入上检测上升沿。
- 通过写入 TIMx_CCMR1 寄存器中的 IC1F[3:0]位来配置输入滤波器的持续时间（如果不需要滤波器，请保持 IC1F=0000）。
- 通过在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=0 来选择上升沿极性。
- 通过在 TIMx_SMCR 寄存器中写入 SMS=111 来将定时器配置为外部时钟模式 1。
- 通过在 TIMx_SMCR 寄存器中写入 TS=101 来选择 TI1 作为触发输入源。
- 通过在 TIMx_CR1 寄存器中写入 CEN=1 来启用计数器。

注意：捕获预分频器不是用于触发的，因此无需配置。当 TI1 上出现上升沿时，计数器计数一次，并且 TIF 标志被设置。

TI1 上升沿和计数器实际时钟之间的延迟是由于 TI1 输入上的重新同步电路造成的。

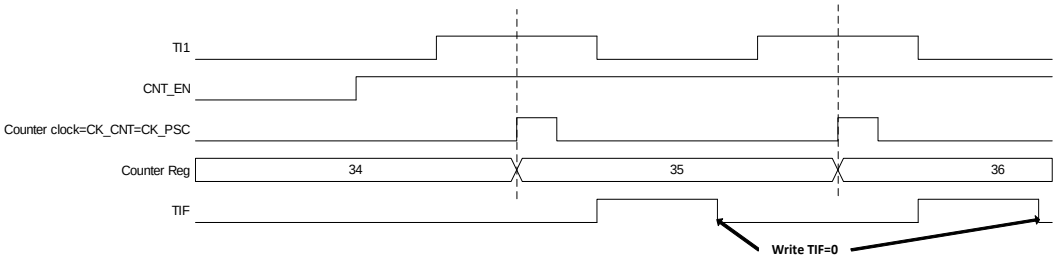


图 6.18 外部时钟模式 1 的控制时序

外部时钟源模式 2

通过在 TIMx_SMCR 寄存器中写入 ECE=1 来选择这种模式。

计数器可以在外部触发输入 ETR 的每个上升沿或下降沿进行计数。

下图提供了外部触发输入块的概览：

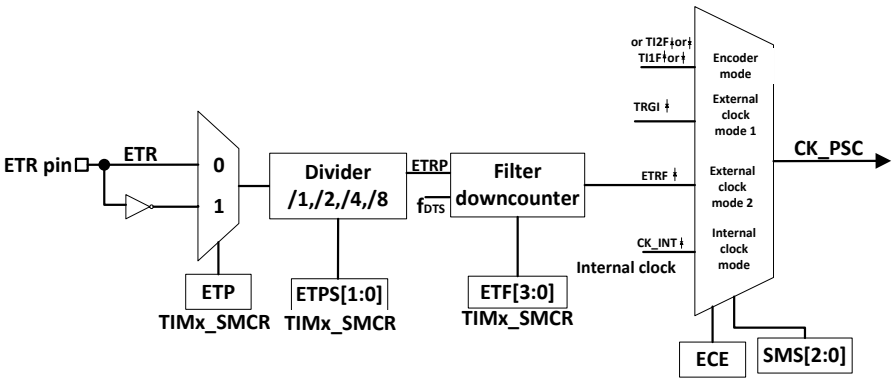


图 6.19 外部触发器输入阻塞

例如，要配置上升计数器以每 2 个 ETR 上升沿计数，请使用以下步骤：

- 由于这个例子中不需要过滤器，将 ETF[3:0]=0000 写入 TIMx_SMCR 寄存器。
- 通过写入 ETPS[1:0]=01 到 TIMx_SMCR 寄存器来设置预分频器。
- 通过写入 ETP=0 到 TIMx_SMCR 寄存器来选择 ETR 引脚上的上升沿检测。
- 通过写入 ECE=1 到 TIMx_SMCR 寄存器来启用外部时钟模式 2。
- 通过写入 CEN=1 到 TIMx_CR1 寄存器来启用计数器。

注意：计数器每 2 个 ETR 上升沿计数一次。

ETR 上升沿和计数器实际时钟之间的延迟是由于 ETRP 信号上的重新同步电路造成的。

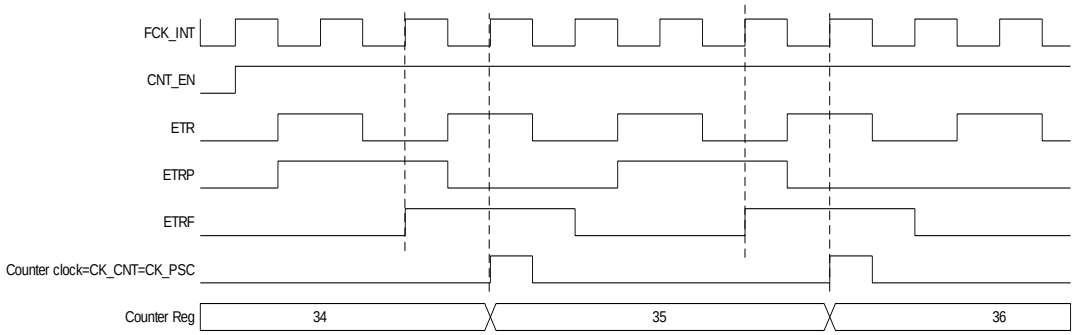


图 6.20 外部时钟模式 2 下的控制时序

6.8.3.3. 捕获/比较通道

每个捕获/比较通道都是围绕一个捕获/比较寄存器（包括一个影子寄存器）、捕获的输入阶段（带有数字滤波器、多路复用和预分频器）以及一个输出阶段（带有比较器和输出控制）构建的。

下图提供了一个捕获/比较通道的概览。

输入阶段对相应的 TIx 输入进行采样，以生成一个经过滤波的信号 TIxF。然后，带有极性选择的边沿检测器生成一个信号（TIxFPx），该信号可以被从属模式控制器用作触发输入，或者用作捕获命令。在捕获寄存器（ICxPS）之前进行预分频。

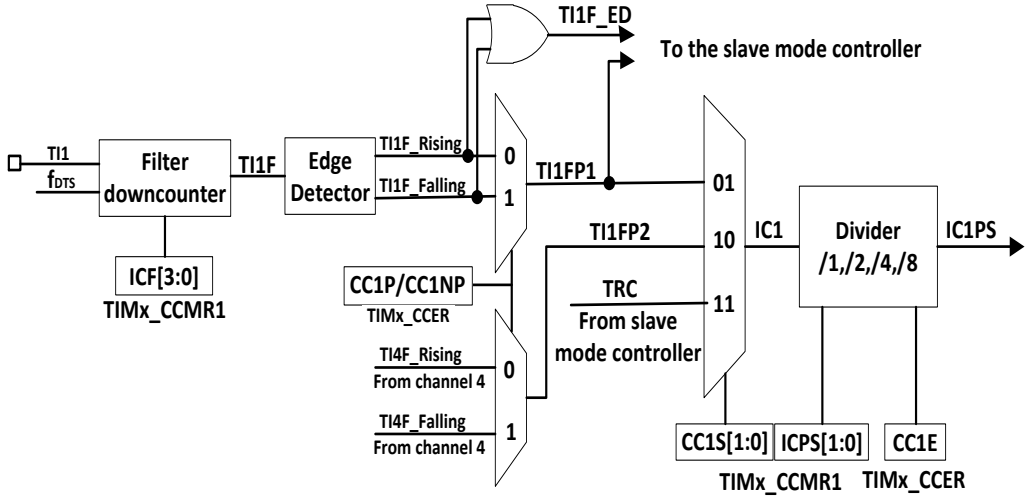


图 6.21 捕获比较通道

(例如：通道 1 输入阶段)

注意：输出部分产生一个中间波形 OCxRef（高效率）作为参考，链的末端决定最终输出信号的极性。

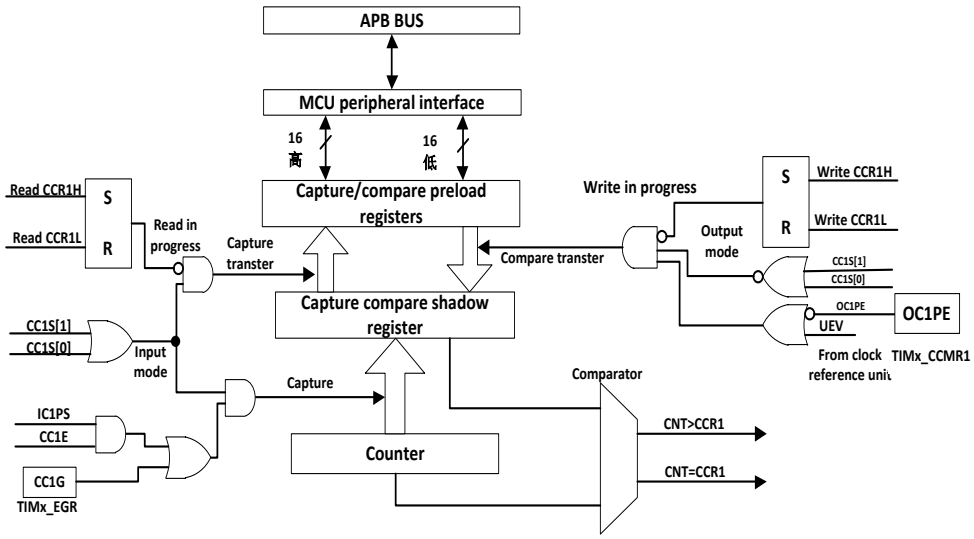


图 6.22 捕获比较通道 1 的主要电路

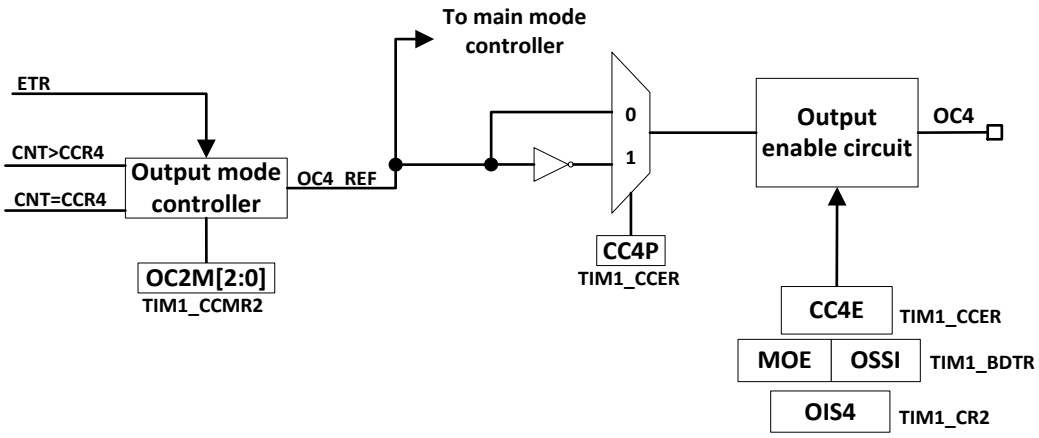


图 6.23 捕获比较器的通道输出部分(通道 4)

捕获/比较模块由一个预加载寄存器和一个影子寄存器组成。写入和读取操作始终访问预加载寄存器。

在捕获模式下，捕获实际上是在影子寄存器中完成的，然后将其复制到预加载寄存器中。

在比较模式下，预加载寄存器的内容被复制到影子寄存器中，然后与计数器进行比较。

6.8.3.4. PWM 模式

脉冲宽度调制（PWM）模式允许您生成一个信号，其频率由 TIMx_ARR 寄存器的值决定，占空比由 TIMx_CCRx 寄存器的值决定。

每个通道（每个 OCx 输出一个 PWM）可以独立选择 PWM 模式，通过在 TIMx_CCMRx 寄存器的 OCxM 位写入‘110’（PWM 模式 1）或‘111’（PWM 模式 2）来实现。您必须通过设置 TIMx_CCMRx 寄存器中的 OCxPE 位来启用相应的预加载寄存器，并且最终通过设置 TIMx_CR1 寄存器中的 ARPE 位来启用自动重载预加载寄存器（在向上计数或中心对齐模式下）。

由于预加载寄存器只有在更新事件发生时才转移到影子寄存器，所以在启动计数器之前，您必须通过设置 TIMx_EGR 寄存器中的 UG 位来初始化所有寄存器。

OCx 极性可以通过 TIMx_CCER 寄存器中的 CCxP 位进行软件编程。它可以被编程为高电平有效或低电平有效。OCx 输出通过 TIMx_CCER 和 TIMx_BDTR 寄存器中的 CCxE、CCxNE、MOE、OSSI 和 OSSR 位的组合来启用。有关更多详细信息，请参阅 TIMx_CCER 寄存器描述。

在 PWM 模式（1 或 2）下，TIMx_CNT 和 TIMx_CCRx 始终进行比较，以确定 TIMx_CCRx ≤ TIMx_CNT 还是 TIMx_CNT ≤ TIMx_CCRx（取决于计数器的方向）。

根据 TIMx_CR1 寄存器中的 CMS 位，定时器能够生成边沿对齐模式或中心对齐模式的 PWM：

PWM 边沿对齐模式

- 向上计数配置:
  - 当 TIMx_CR1 寄存器中的 DIR 位为低时，向上计数模式是激活的。
  - 在接下来的例子中，我们考虑 PWM 模式 1。只要 TIMx_CNT 小于 TIMx_CCRx，参考 PWM 信号 OCxREF 就保持高电平，否则它变为低电平。如果 TIMx_CCRx 中的比较值大于自动重装载值（在 TIMx_ARR 中），那么 OCxREF 将保持在‘1’。

如果比较值为 0，则 OCxRef 保持在‘0’。

以下图形显示了在 TIMx_ARR=8 的一个示例中一些边缘对齐的 PWM 波形。

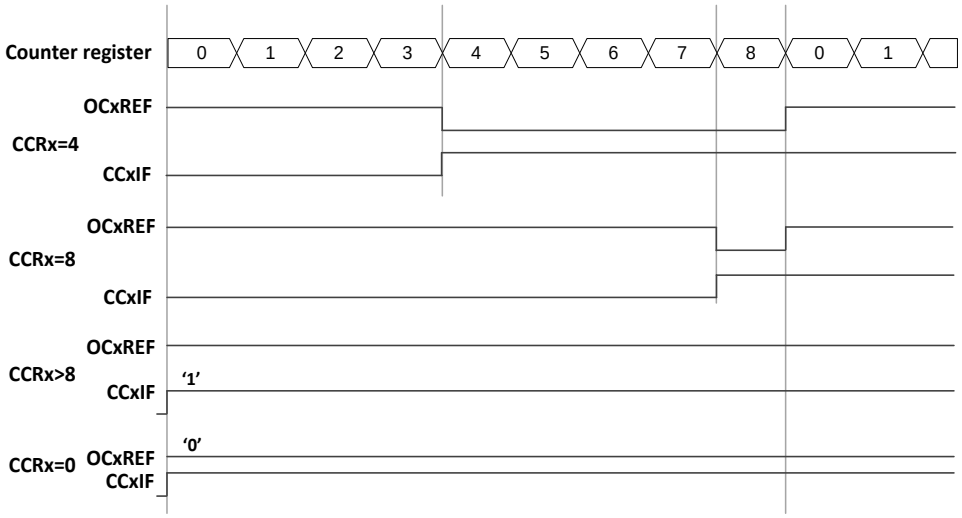


图 6.24 边沿对齐的 PWM 波形 (ARR=8)

- 向下计数配置:
  - 当 TIMx_CR1 寄存器中的 DIR 位为高时，下计数是激活的。
  - 在 PWM 模式 1 中，只要 TIMx_CNT 大于 TIMx_CCRx，参考信号 OCxRef 就保持低电平，否则它会变成高电平。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重载值，那么 OCxREF 将保持在‘1’。在这种模式下，0%的 PWM 是不可能的。

PWM 中心对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不等于'00'时（所有其他配置对 OCxRef/OCx 信号有相同的影响），中心对齐模式被激活。比较标志位在计数器向上计数时、向下计数时或在向上和向下计数时根据 CMS 位的配置被设置。TIMx_CR1 寄存器中的方向位（DIR）由硬件更新，软件不得更改。

下图显示了一些中心对齐 PWM 波形的一个示例：

- TIMx_ARR = 8.
- PWM 模式 1.
- 当计数器按照 TIMx_CR1 寄存器中选择的中心对齐模式 1 进行倒计时时，会设置该标志。

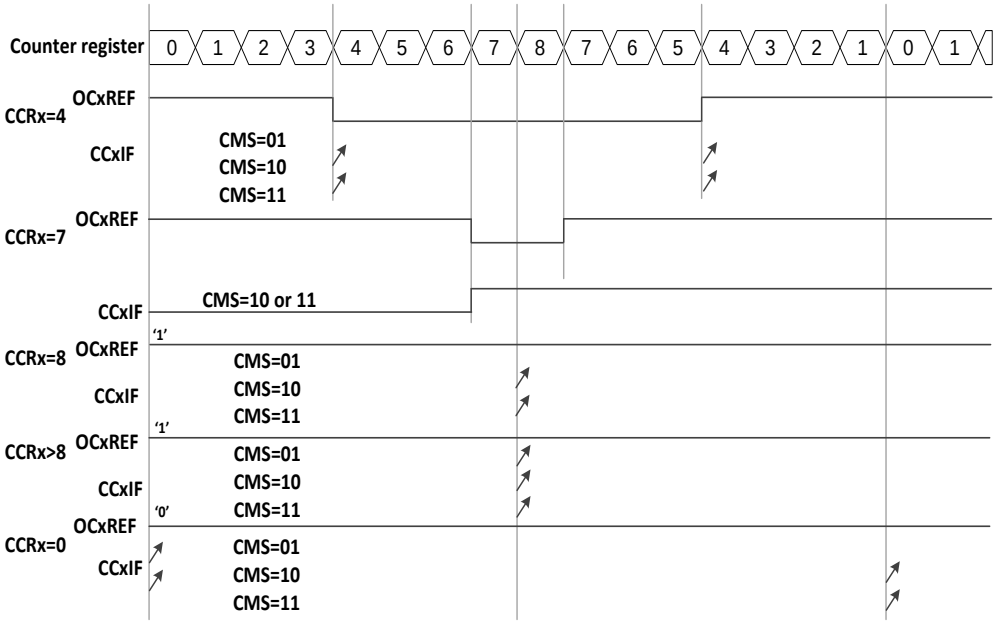


图 6.25 中心对齐的 PWM 波形 (ARR=8)

使用居中对齐模式的提示：

- 当以中心对齐模式启动时，将使用当前的上下配置。这意味着计数器根据 TIMx_CR1 寄存器中 DIR 位写入的值向上或向下计数。此外，软件不得同时更改 DIR 和 CMS 位。
- 在居中对齐模式下运行时写计数器是不推荐的，因为这可能导致意外的结果。特别是：
  - 如果你在计数器中写入的值大于自动重装载值（TIMx_CNT>TIMx_ARR），方向不会更新。例如，如果计数器正在向上计数，它将继续向上计数。
  - 如果你写入 0 或者写入 TIMx_ARR 值到计数器中，方向会更新，但不会生成更新事件 UEV。
- 使用中心对齐模式最安全的方法是在启动计数器之前通过软件生成更新（在 TIMx_EGR 寄存器中设置 UG 位），并且在计数器运行时不写入计数器。

6.8.3.5. 互补输出和死区时间插入

高级控制定时器（TIMER）可以输出两个互补信号，并管理输出的关闭和开启时刻。这段时间通常被称为死区时间，您需要根据连接到输出端的设备及其特性（例如电平转换器的固有延迟、功率开关的延迟等）进行调整。

您可以独立地为每个输出选择输出极性（主输出 OCx 或互补输出 OCxN）。这是通过向 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位写入数据来实现的。

互补信号 OCx 和 OCxN 是通过几个控制位的组合激活的：TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位，以及 TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。特别是，当切换到空闲状态（MOE 下降到 0）时，激活死区时间。

通过设置 CCxE 和 CCxNE 位来启用死区时间插入，如果存在断路电路，则还需要设置 MOE 位。每个通道都有一个 10 位的死区时间发生器。它从参考波形 OCxREF 生成两个输出 OCx 和 OCxN。

如果 OCx 和 OCxN 为高电平有效：

- OCx 输出信号与参考信号相同，除了上升沿，它相对于参考信号的上升沿是延迟的。

- OCxN 输出信号与参考信号相反，除了上升沿，它相对于参考信号的下降沿是延迟的。

如果延迟大于活动输出（OCx 或 OCxN）的宽度，则相应的脉冲不会生成。

以下图表显示了死区时间发生器的输出信号与参考信号 OCxREF 之间的关系。（在这些示例中，我们假设 CCxP=0, CCxNP=0, MOE=1, CCxE=1 和 CCxNE=1）。

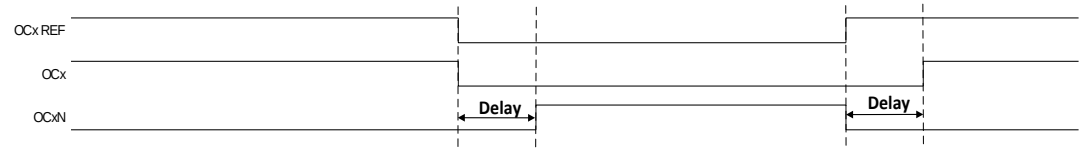


图 6.26 互补输出和死区插入

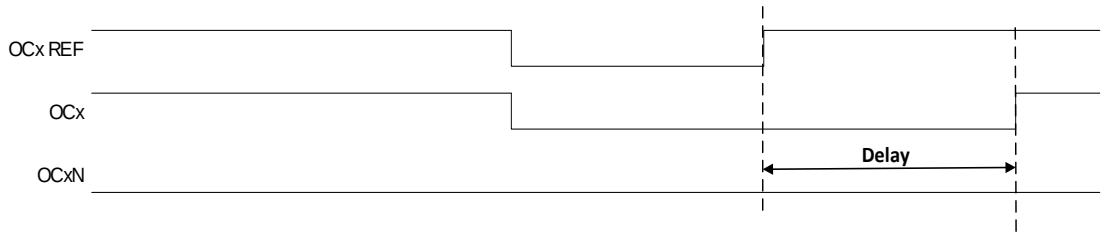


图 6.27 延时大于负脉冲宽度的死区波形

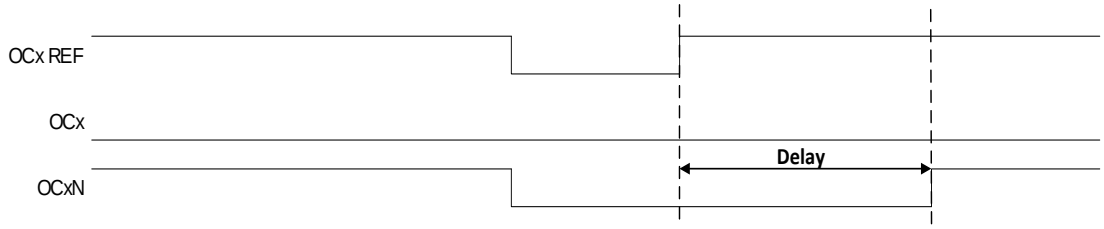


Figure 6.28 延时大于正脉冲宽度的死区波形

死区时间延迟对于每个通道都是相同的，并且可以通过 TIMx_BDTR 寄存器中的 DTG 位进行编程。

将OCxREF重定向到OCx或OCxN

在输出模式（强制、输出比较或 PWM）下，通过配置 TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位，OCxREF 可以重定向到 OCx 输出或 OCxN 输出。

这允许你在其中一个输出发送特定波形（如 PWM 或静态激活电平），而互补输出保持在其非激活电平。其他替代可能性是使两个输出都处于非激活电平，或者使两个输出都处于激活状态并具有死区时间的互补。

注意：当仅启用 OCxN（CCxE=0, CCxNE=1）时，它不会被反转，并且一旦 OCxREF 为高电平就会变为激活状态。例如，如果 CCxNP=0，则 OCxN=OCxRef。另一方面，当 OCx 和 OCxN 都被启用（CCxE=CCxNE=1）时，OCx 在 OCxREF 为高电平时变为激活状态，而 OCxN 则相反，在 OCxREF 为低电平时变为激活状态。

6.8.4. TIMER 寄存器映射

Offset	Name	Description
0x0000	TIM_CR1	Timer control register 1
0x0004	TIM_CR2	Timer control register 2
0x0008	TIM_SMCR	Timer slave mode control register
0x000c	TIM_DIER	Timer DMA/interrupt enable register
0x0010	TIM_SR	Timer status register
0x0014	TIM_EGR	Timer event generation register
0x0018	TIM_CCMR1	Timer capture/compare mode register1
0x001c	TIM_CCMR2	Timer capture/compare mode register2
0x0020	TIM_CCER	Timer capture/compare enable register
0x0024	TIM_CNT	Timer counter
0x0028	TIM_PSC	Timer prescaler
0x002c	TIM_ARR	Timer auto-reload register
0x0030	TIM_RCR	Timer reperepetition counter register
0x0034	TIM_CCR1	Timer capture/compare register1
0x0040	TIM_CCR4	Timer capture/compare register4
0x0044	TIM_BDTR	Timer break and dead-time register
0x0048	TIM_DCR	Timer dma control register
0x004c	TIM_DMAR	Timer dma address for full transfer

TIM_CR1 address offset: 0x0000

Bit	R/W	Reset	Name	Description
31:10	N/A	0x0	N/A	reserved
9:8	RW	0x0	CKD	Clock division This bit-field indicates the division ratio between the timer clock (tCK_INT) frequency and the dead-time and sampling clock (tDTS)used by the dead-time generators and the digital filters
7	RW	0x0	ARPE	Auto-reload preload enable 0: TIM_ARR register is not buffered 1: TIM_ARR register is buffered
6:5	RW	0x0	CMS	Center-aligned mode selection 00: Edge-aligned mode. The counter counts up or down depending on the direction bit (DIR). 01: Center-aligned mode 1. The counter counts up and down alternatively. Output compare interrupt flags of channels are set only when the counter is counting down. 10: Center-aligned mode 2.

				The counter counts up and down alternatively. Output compare interrupt flags of channels are set only when the counter is counting up. 11: Center-aligned mode 3. The counter counts up and down alternatively. Output compare interrupt flags of channels are set both when the counter is counting up or down.
4	RW	0x0	DIR	Direction 0: Counter used as upcounter 1: Counter used as downcounter
3	RW	0x0	OPM	One pulse mode 0: Counter is not stopped at update event 1: Counter stops counting at the next update event (clearing the bit CEN)
2	RW	0x0	URS	Update request source This bit is set and cleared by software to select the UEV event sources.
1	RW	0x0	UDIS	Update disable, This bit is set and cleared by software to enable/disable UEV event generation.
0	RW	0x0	CEN	Counter enable 0: Counter disabled 1: Counter enabled

TIM_CR2 address offset: 0x0004

Bit	R/W	Reset	Name	Description
31:15	N/A	0x0	N/A	reserved
14	RW	0x0	OIS4	Output Idle state 4 (OC4 output) 0:OC4=0 (after a dead-time if OC4N is implemented) when MOE=0 1:OC4=1 (after a dead-time if OC4N is implemented) when MOE=0
13:10	N/A	0x0	N/A	reserved
9	RW	0x0	OIS1N	Output Idle state 1 (OC1N output) 0: OC1N=0 after a dead-time when MOE=0 1: OC1N=1 after a dead-time when MOE=0
8	RW	0x0	OIS1	Output Idle state 1 (OC1 output) 0: OC1=0 (after a dead-time if OC1N is implemented) when MOE=0 1: OC1=1 (after a dead-time if OC1N is implemented) when MOE=0
7	RW	0x0	TI1S	TI1 selection 0: The TIM_CH1 pin is connected to TI1 input 1:The TIM_CH1, CH2 and CH3 pins are

				connected to the TI1 input (XOR combination)
6:4	RW	0x0	MMS	Master mode selection
3	RW	0x0	CCDS	Capture/compare DMA selection 0: CC DMA request sent when CC event occurs 1: CC DMA requests sent when update event occurs
2	RW	0x0	CCUS	Capture/compare control update selection 0: When capture/compare control bits are preloaded (CCPC=1), they are updated by setting the COMG bit only 1: When capture/compare control bits are preloaded (CCPC=1), they are updated by setting the COMG bit or when an rising edge occurs on TRGI
1	N/A	0x0	N/A	reserved
0	RW	0x0	CCPC	Capture/compare preloaded control 0: CCE, CCNE and OCM bits are not preloaded 1: CCE, CCNE and OCM bits are preloaded

TIM_SMCR address offset: 0x0008

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
15	RW	0x0	ETP	External trigger polarity This bit selects whether ETR or ETR is used for trigger operations 0: ETR is non-inverted, active at high level or rising edge. 1: ETR is inverted, active at low level or falling edge.
14	RW	0x0	ECE	External clock enable This bit enables External clock mode 2. 0: External clock mode 2 disabled 1: External clock mode 2 enabled.
13:12	RW	0x0	ETPS	External trigger prescaler
11:8	RW	0x0	ETF	External trigger filter
7	RW	0x0	MSM	Master/slave mode
6:4	RW	0x0	TS	Trigger selection
3	N/A	0x0	N/A	reserved
2:0	RW	0x0	SMS	Slave mode selection

TIM_DIER address offset: 0x000c

Bit	R/W	Reset	Name	Description
31:15	N/A	0x0	N/A	reserved
14	RW	0x0	TDE	Trigger DMA request enable 0: Trigger DMA request disabled

				1: Trigger DMA request enabled
13	RW	0x0	COMDE	COM DMA request enable 0: COM DMA request disabled 1: COM DMA request enabled
12	RW	0x0	CC4DE	Capture/Compare 4 DMA request enable 0: CC4 DMA request disabled 1: CC4 DMA request enabled
11	N/A	0x0	N/A	reserved
10	N/A	0x0	N/A	reserved
9	RW	0x0	CC1DE	Capture/Compare 1 DMA request enable 0: CC1 DMA request disabled 1: CC1 DMA request enabled
8	RW	0x0	UDE	Update DMA request enable 0: Update DMA request disabled 1: Update DMA request enabled
7	RW	0x0	BIE	Break interrupt enable 0: Break interrupt disabled 1: Break interrupt enabled
6	RW	0x0	TIE	Trigger interrupt enable 0: Trigger interrupt disabled 1: Trigger interrupt enabled
5	RW	0x0	COMIE	COM interrupt enable 0: COM interrupt disabled 1: COM interrupt enabled
4	RW	0x0	CC4IE	Capture/Compare 4 interrupt enable 0: CC4 interrupt disabled 1: CC4 interrupt enabled
3	N/A	0x0	N/A	reserved
2	N/A	0x0	N/A	reserved
1	RW	0x0	CC1IE	Capture/Compare 1 interrupt enable 0: CC1 interrupt disabled 1: CC1 interrupt enabled
0	RW	0x0	UIE	Update interrupt enable 0: Update interrupt disabled 1: Update interrupt enabled

TIM_SR address offset: 0x0010

Bit	R/W	Reset	Name	Description
31:13	N/A	0x0	N/A	reserved
12	RW	0x0	CC4OF	Capture/Compare 4 over capture flag
11	N/A	0x0	N/A	reserved
10	N/A	0x0	N/A	reserved
9	RW	0x0	CC1OF	Capture/Compare 1 over capture flag

8	N/A	0x0	N/A	reserved
7	RW	0x0	BIF	Break interrupt flag
6	RW	0x0	TIF	Trigger interrupt flag
5	RW	0x0	COMIF	COM interrupt flag
4	RW	0x0	CC4IF	Capture/Compare 4 interrupt flag
3	N/A	0x0	N/A	reserved
2	N/A	0x0	N/A	reserved
1	RW	0x0	CC1IF	Capture/Compare 1 interrupt flag
0	RW	0x0	UIF	Update interrupt flag This bit is set by hardware on an update event. It is cleared by software. 0: No update occurred. 1: Update interrupt pending.

TIM_EGR address offset: 0x0014

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7	W	0x0	BG	Break generation
6	W	0x0	TG	Trigger generation
5	W	0x0	COMG	Capture/Compare control update generation
4	W	0x0	CC4G	Capture/Compare 4 generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0: No action 1: A capture/compare event is generated on channel 4:
3	N/A	0x0	N/A	reserved
2	N/A	0x0	N/A	reserved
1	W	0x0	CC1G	Capture/Compare 1 generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. 0: No action 1: A capture/compare event is generated on channel 1:
0	W	0x0	UG	Update generation This bit can be set by software, it is automatically cleared by hardware. 0: No action 1: Reinitialize the counter and generates an update of the registers.

TIM_CCMR1 address offset: 0x0018 (Output compare mode)

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7	RW	0x0	OC1CE	Output Compare 1 clear enable
6:4	RW	0x0	OC1M	Output Compare 1 mode
3	RW	0x0	OC1PE	Output Compare 1 preload enable
2	RW	0x0	OC1FE	Output Compare 1 fast enable
1:0	RW	0x0	CC1S	Capture/Compare 1 Selection

TIM_CCMR1 address offset: 0x0018 (Input capture mode)

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7:4	RW	0x0	IC1F	Input capture 1 filter
3:2	RW	0x0	IC1PSC	Input capture 1 prescaler
1:0	RW	0x0	CC1S	Capture/Compare 1 Selection

TIM_CCMR2 address offset: 0x001c (Output compare mode)

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
15	RW	0x0	OC4CE	Output Compare 4 clear enable
14:12	RW	0x0	OC4M	Output Compare 4 mode
11	RW	0x0	OC4PE	Output Compare 4 preload enable
10	RW	0x0	OC4FE	Output Compare 4 fast enable
9:8	RW	0x0	CC4S	Capture/Compare 4 Selection
7:0	N/A	0x0	N/A	reserved

TIM_CCMR2 address offset: 0x001c (Input capture mode)

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
15:12	RW	0x0	IC4F	Input capture 4 filter
11:10	RW	0x0	IC4PSC	Input capture 4 prescaler
9:8	RW	0x0	CC4S	Capture/Compare 4 Selection
1:0	N/A	0x0	N/A	reserved

TIM_CCER address offset: 0x0020

Bit	R/W	Reset	Name	Description
31:14	N/A	0x0	N/A	reserved
13	RW	0x0	CC4P	Capture/Compare 4 output polarity
12	RW	0x0	CC4E	Capture/Compare 4 output enable
11:4	N/A	0x0	N/A	reserved
3	RW	0x0	CC1NP	Capture/Compare 1 complementary output polarity
2	RW	0x0	CC1NE	Capture/Compare 1 complementary output enable

1	RW	0x0	CC1P	Capture/Compare 1 output polarity
0	RW	0x0	CC1E	Capture/Compare 1 output enable

TIM_CNT address offset: 0x0024

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
15:0	RW	0x0	CNT	Counter value

TIM_PSC address offset: 0x0028

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
15:0	RW	0x0	PSC	Prescaler value

TIM_ARR address offset: 0x002c

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
15:0	RW	0x0	ARR	Prescaler value ARR is the value to be loaded in the actual auto-reload register.

TIM_RCR address offset: 0x0030

Bit	R/W	Reset	Name	Description
31:8	N/A	0x0	N/A	reserved
7:0	RW	0x0	REP	Repetition counter value

TIM_CCR1 address offset: 0x0034

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
15:0	RW	0x0	CCR1	Capture/Compare 1 value

TIM_CCR4 address offset: 0x0040

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
15:0	RW	0x0	CCR4	Capture/Compare 4 value

TIM_BDTR address offset: 0x0044

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
15	RW	0x0	MOE	Main output enable
14	RW	0x0	AOE	Automatic output enable
13	RW	0x0	BKP	Break polarity
12	RW	0x0	BKE	Break enable
11	RW	0x0	OSSR	Off-state selection for Run mode

10	RW	0x0	OSSI	Off-state selection for Idle mode
9:8	RW	0x0	LOCK	Lock configuration
7:0	RW	0x0	DTG	Dead-time generator setup

TIM_DCR address offset: 0x0048

Bit	R/W	Reset	Name	Description
31:13	N/A	0x0	N/A	reserved
12:8	RW	0x0	DBL	DMA burst length
7:5	N/A	0x0	N/A	reserved
4:0	RW	0x0	DBA	DMA base address

TIM_DMAR address offset: 0x004c

Bit	R/W	Reset	Name	Description
31:16	N/A	0x0	N/A	reserved
15:0	RW	0x0	DMAB	DMA register for burst accesses

6.9. OTP

6.9.1. 简介

OTP 控制器以自动化和透明的方式实现了 OTP 宏单元的所有功能。
控制器通过 AHB 总线主控简化了所有数据传输（读取和编程）以及测试项目。

6.9.2. 主要特性

- 符合 AMBA™ 2 AHB 协议规范
- 自动单错误校正码（ECC）- 6 位（在 OTP 单元中实现）
- 从 OTP 单元单次读取访问中读取 32 位数据
- 用于编程的单字缓冲区。不支持突发编程
- 空字为 0xFFFFFFFF。按 32 位字编程零
- 通过 AHB 从机内存对 OTP 存储单元进行透明随机地址访问
- otp 地址：otp 存储大小为 24k 字节，地址范围从 0x6000000 到 0x60006ffc；otp 寄存器从 0x60100000 到 0x60100030
- 访问速度：快速读取速度为 16MHz*32 位；字编程速度从 200 微秒（全 1 数据）到 1100 微秒（全 0 数据）；

6.9.3. 功能描述

6.9.3.1. 框图

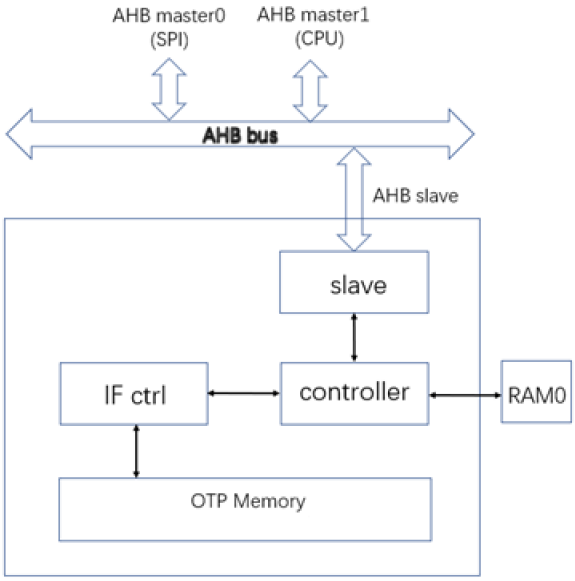


图 6.29 OTP 控制器方框图

6.9.3.2. 操作流程

块儿写流程:

块写操作将从 RAM 读取一个数据块，并将这些数据编程到一次性可编程存储器(OTP mem)中，在执行块写操作之前，请确保写使能位(write_en bit)已设置，并且测试行使能(test_row_en)为 0。

- **软件:**
  - 读取状态寄存器中的 OTP_READY 位，确保 OTP 控制处于就绪状态；
  - 设置 **RAM Config register** 的 ram_r_baddr 和 ram_r_length 位；
  - 设置 **OTP Write Address register**；
  - 将 **Write Enable register** 设置为 32’h1133_5577 以启用 OTP 写权限；
  - 向 **Command register** 写入 16’h5a5a；
  - 将 **Execute register** 设置为 1 以执行块写命令；
- **硬件:**
  - 在接收到块写命令后，硬件将从 RAM1 读取数据并将其写入 OTP 内存，一旦所有数据写入完成，OTP 控制器将再次回到就绪状态。
- **软件:**
  - 读取 OTP STATUS 寄存器直到 otp_ready 位再次为 1;
  - 将 **Write Enable register** 设置为 32'haabb_ccdd，以禁用 OTP 写权限;

软件案例说明：

在这种情况下，RAM1 中的 2 个字（8 字节），从字节地址 0 到字节地址 7（字节地址），将被编程到 OTP 存储器的字地址 0 和字地址 1 的位置。

- 读 OTP_READY 位，直到 OTP_READY = 1;
- 设 ram_r_baddr = 0 ， ram_r_length = 8;
- 设 OTP Write Address register = d’4091;
- 设 Write Enable register = 32’h1133_5577;
- 写 16’h5a5a 到命令寄存器;
- 设 Execute register = 1;
- 读 OTP_READY 位。 直到 OTP_READY = 1;
- 设 Write Enable register = 32’haabb_ccdd;

字写流程:

字写操作将一次性将一个字（4 字节）编程到一次性可编程存储器（OTP mem）中；在这种情况下，一个字将被编程到 OTP 存储器的字地址 1 位置（字节地址是 4 到 7）。

- **软件:**
  - 读状态寄存器 OTP_READY， 确保 OTP 控制是 READY 状态;
  - **设 OTP Write Address register = 0x4**;
  - **设 OTP Write Data register**;
  - 设 **Write Enable register =** 32’h1133_5577 ， 使能 OTP 写的权限;
  - 写 16’h0123 到 **Command register**;
  - 设 **Execute register = 1 执行字写操作**;
- **硬件:**
  - 在接收到写命令后，硬件将数据编程到 OTP（一次性可编程）存储器中，然后 OTP 控制器将再次回到就绪状态。
- **软件:**
  - 读 OTP STATUS 寄存器直到 otp_ready=1;
  - 设 **Write Enable register =** 32’haabb_ccdd 禁止 OTP 写权限;

软件案例说明

在这种情况下，4 字节的数据 0x0000aa55 将被编程到 OTP 存储器字地址 1 的位置（字节地址是 0x4~0x7）：

- 读 OTP_READY 位，直到 OTP_READY = 1;
- 设 OTP Write Wdata register = 0x0000aa55;
- 设 OTP Write Address register = **0x4**;
- 设 Write Enable register = 32’h1133_5577;
- 写 16’h01123 到命令寄存器；
- 设 Execute register = 1;
- 读 OTP_READY 位直到 OTP_READY = 1;
- 设 Write Enable register = 32’haabb_ccdd;

透明读取流程

透明读取功能使用一个 AHB 从接口，该接口用于读取 OTP 存储器的内容，并且是只读的；

这个 AHB 从接口支持 AHB 协议中包含的所有突发类型，包括：SINGLE、INCR、INCR4、INCR8、INCR16、WRAP4、WRAP8、WRAP16；

SINGLE 读取支持字节、半字和字的 hsize；

BURST 读取仅支持字的 hsize；

Initial margin read FF 测试(for main array)

Initial margin read FF 测试是为了检查在 CP/FT 阶段所有 4k 字的主阵列是否全部为 1；

- **Soft:**
  - 读 OTP_READY 位,确保 OTP 控制是在准备好状态;
  - 设 **OTP_CFG register RMODE** ;
  - 写 16'h4567 到 **Command register**;
  - 设 **Execute register = 1 到执行字写命令**;
- **硬件:**
  - 在接收到 M_CHECK 命令后，OTP 控制器将读取主阵列中的所有 4k 字，并自动检查所有数据是否为 1；在检查所有数据是否为 1 或读取到错误数据后，此过程将结束，OTP 控制器将回到读取就绪状态。
- **软件:**
  - Read OTP STATUS register until otp_ready bit is 1 again;
  - read bit TEST_RESULT;if TEST_ RESULT =0 indicate test is pass,or test is fail;

OTP 存储器状态:

考虑降低 OTP 存储器的功耗，软件可以通过命令寄存器控制 OTP 存储器在四种功耗状态之间的转换；下图展示了需要设置的用于状态转换的命令，无效的命令将被 OTP 控制器忽略；

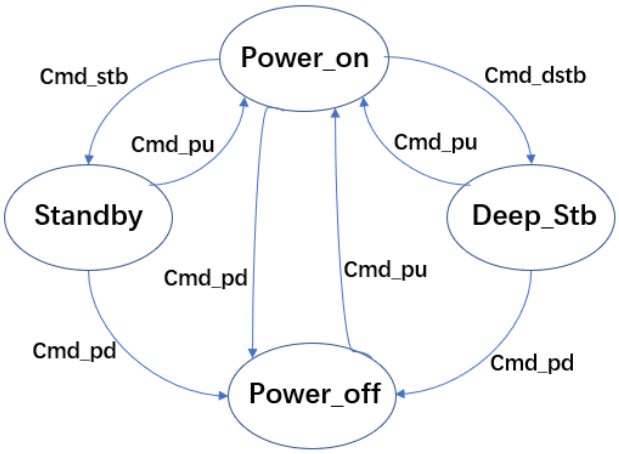


图 6.30 OTP 内存状态转换图

6.9.4. OTP 寄存器映射

Offset	Name	Description
0x0000	OTP_CONFIG	OTP Config Register
0x0004	OTP_COMMAND	OTP Command Control Register
0x0008	OTP_W_PROTECT	OTP Write Enable Register
0x000c	OTP_TROW_EN	OTP Test_row_en Register
0x0010	OTP_EXECUTE	OTP Command Execute register
0x0014	OTP_ADDR	OTP Write Address register
0x0018	OTP_WDATA	OTP Write Data register
0x001c	OTP_RAW_BADDR_L	OTP RAM Config register
0x0020	OTP_STATUS	OTP Status Control Register
0x0024	OTP_INT	OTP Int register
0x0028	OTP_PROTECT_CODE	OTP Read Protect Code register
0x002c	OTP_PCLK TIMING CFG	OTP PCLK TIMING CFG Register
0x0030	OTP_PCLK TIMING CFG EN	OTP PCLK TIMING CFG Enable Register

CONFIG address offset: 0x4000+0x00

Bit	R/W	Reset	Name	Description
31:6	R	26'h0	RES	
5:4	W/R	2'b01	FDIV	Frequency division from 64mhz 2'b00:don't divide,otp_clk=64hmz; 2'b01:divide by 2, otp_clk=32hmz; 2'b10:divide by 4, otp_clk=16hmz;
3:0	W/R	4'b0000	RMODE	4'b0000:user read mode 4'b0001:initial margin read mode(test mode) 4'b0100:PGM margin read mode(test mode) 4'b1001:high temp initial margin read mode(test mode) 4'b1100:high temp PGM margin read mode(test mode)

Note:RMODE will only change after command "switch read mode" execute;

COMMAND address offset: 0x4000+0x04

Bit	R/W	Reset	Name	Description
31:16	R		RES	
15: 0	W/R	6'h0	COMMAND	16'h0:res 16'h5a5a:block write command, write otp which data read from RAM;(WEN bit must be set, TROW_EN must be 0) 16'h0123:word write command

				(WEN bit must be set) 16'h4567:initial margin read_FF command, Check if 4K Main array initial value is all 1; 16'h0110:power_down; 16'h0220:power_up; 16'h0330:goto standby state; 16'h0440:goto deep standby state; 16'h0550:switch read mode;

Note:all command only effect after set **Execute** reg;

WRITE ENABLE address offset: 0x4000+0x08

Bit	R/W	Reset	Name	Description
31:0	WR	32'haab b_ccdd	WRITE ENABLE	32'h1133_5577:enable, OTP mem can be programed; 32'haabb_ccdd: disable, OTP mem can't be programed;

TEST ROW EN address offset: 0x4000+0x0c

Bit	R/W	Reset	Name	Description
31:0	WR	32' hffff_aabb	TEST_ROW_EN	32'h00005566:enable test row access; 32'hffffaabb:disable test row access;

Note1:test row is individual memory block for testing,it does not include in main array size;
Note2:the size of test row is 16 word;before test_row_en is set,a right RMODE must be set;

COMMAND EXECUTE address offset: 0x4000+0x10

Bit	R/W	Reset	Name	Description
31:1	R		RES	
0	WO	1'b0	EXECUTE	Write 1 to this bit will execute current command and read always return 0;

OTP WRITE address offset: 0x4000+0x14

Bit	R/W	Reset	Name	Description
31:15	R	0	-	Res
14:0	W/R	15'h0	OTP_WADDR	Write OTP address in single word operation or OTP base address in block write operation which data from RAM1; Range:15'd0~15'd24572;(OTP mem size is d'24576)

Note:When COMMAND=16'h0123,OTP_ADDR is otp address for single word write operation;When COMMAND=16'h5a5a,OTP_ADDR is write otp base address for block write operation;

WRITE DATA address offset: 0x4000+0x18

Bit	R/W	Reset	Name	Description
31:0	W/R	32'h0	WDATA	write data to OTP memory;

Note: Only valid When execute word write command;

RAM CONFIG address offset: 0x4000+0x1c

Bit	R/W	Reset	Name	Description
31:30			-	Res
29:16	W/R	14'h0	RAM_R_LENGTH	read data byte num from RAM1, range is 4~8192; 'd4:num=4; 'd8:num=8; 'd8192:num=8192;(RAM1 size is 2048x32)
15:13				Res
12:0	W/R	13'h0	RAM_R_BADDR	read data byte base address of RAM1, range is 0~8188; 0,4,8…8188;

Note1: only valid when execute block write command;

Note2: (RAM_r_length/4<=d'2048)&& (OTP_waddr/4+ RAM_r_length/4<=d'4096);

STATUS address offset: 0x4000+0x20

Bit	R/W	Reset	Name	Description
31:14	R	0	-	Res
13	R	1'b1	W_PROTECT EN	1'b1:enable;program is forbbiden; 1'b0:disable;program is permit;
12	R	1'b0	TEST_ROW_EN	1'b1:enable; 1'b0:disable;
11:9	R	3'h000	CTRL_ST	Otp controller state: 3'b000:power on(ready) 3'b001:standby 3'b010:deep standby 3'b011:power off 3'b100:programming 3'b101:busy
8	R	1'b1	PENVDD2_VDD2	VDD2 EN
7	R	N/A	VDD2_RDY	VDD2 ready
6	R	1'b0	R_PROTECT EN	1'b1:enable; 1'b0:disable;
5:2	R	4'h0	PTM	Otp memory operation mode
1	R	1'b0	TESTSTU	Check if 4K Main array initial value is all 1; 1'b1:fail, Main array initial value is not all 1; 1'b0:right, Main array initial value is all 1;

0	R	N/A	OTP_READY	1’b1:OTP is in idle state and can accept new command 1’b0:not ready;

OTP INT address offset: 0x4000+0x24

Bit	R/W	Reset	Name	Description
31:4	R	0	-	Res
3	R	1’b0	OTP_INT	Otp int= Otp_int_raw& Otp_int_en; 1’b1:otp interrupt happen; 1’b0:no otp interrupt
2	R	N/A	OTP_INT_RAW	Otp int status before otp_int_en Before otp_ready Otp_int_raw = 0, after otp_ready Otp_int_raw = 1;
1	W	1’b0	OTP_INT_CLR	Set this bit will clear bit2 and bit3
0	WR	1’b0	OTP_INT_EN	Interrupt enable 1’b1:enable 1’b0:disable

READ PROTECT CODE address offset: 0x4000+0x28

Bit	R/W	Reset	Name	Description
31: 0	R	32’hffff_ffff	READ_PROTECT_CODE	Read protect code,read from otp mem last word address; user only can write data **32’h1234_abcd** to this address to enable otp read protect;

PCLK TIMING CFG address offset: 0x4000+0x2c

Bit	R/W	Reset	Name	Description
31:22	R	0	-	Res
21:20	WR	2’h1	PCLK16M_SOFT _H	Pclk16 cycle number; Pclk16_soft_h >=1;
19:16	WR	4’h2	PCLK16M_SOFT_CYCLE	Pclk16 cycle number; Pclk16_soft_cycle>=2;
15:14	R	2’h0	-	Res
13:12	WR	2’h1	PCLK32M_SOFT _H	Pclk32 cycle number; Pclk32_soft_h >=1;
11:8	WR	4’h3	PCLK32M_SOFT_CYCLE	Pclk32 cycle number; Pclk32_soft_cycle>=2;
7:6	R	2’h0	-	Res
5:4	WR	2’h2	PCLK64M_SOFT _H	Pclk64 cycle number; Pclk64_soft_h >=2;
3:0	WR	4’h5	PCLK64M_SOFT_CYCLE	Pclk64 cycle number; Pclk64_soft_cycle>=4;

PCLK TIMING CFG EN address offset: 0x4000+0x30

Bit	R/W	Reset	Name	Description
31:15	R	0	-	Res
14	R	0	CLK_CALI_DONE	
13:12	R	Depend on freq_div	PCLK_H	Pclk effect cycle number;
11:8	R	Depend on freq_div	PCLK_CYCLE	Pclk effect cycle number;
7:1	R	2'h0	-	Res
0	WR	1'b0	PCLK_SOFT_SET	Pclk_soft_cycle and Pclk_soft_h set enable: 1'b1:enable, 1'b0:disable If Pclk_soft_set =1,pclk param set by Pclk_soft_cycle and Pclk_soft_h; If Pclk_soft_set =0, pclk param set by hardware;

在系统时钟校准完成之前，硬件会根据 OTP 模块的时钟偏差设置一个慢速的 pclk 参数；
软件可以通过写入 PCLK TIMING CFG EN 寄存器并设置 Pclk_soft_set = 1 来改变 pclk 的时序；
系统时钟校准完成后，OTP 读取时序将使用一个稳定且快速的参数；

Otp frequency	Read speed(pclk freq)
16MHz	8MHz*32bit
32MHz	10.6MHz*32bit
64MHz	12.8MHz*32bit

Table 6.3 OTP Read Timing Set

Fastest speed set

OTP 读取速度可以通过设置 PCLK 时序配置寄存器和 PCLK 时序配置使能寄存器达到最大速度 16MHz；Pclk_soft_set = 1；

Otp frequency	Read speed (pclk freq)	pclk timing cfg register set	
16MHz	8MHz*32bit	Pclk16M_soft _h = 1	Pclk16M_soft_cycle = 2
32MHz	16MHz*32bit	Pclk32M_soft _h = 1	Pclk32M_soft_cycle = 2
64MHz	16MHz*32bit	Pclk64M_soft _h = 2	Pclk64M_soft_cycle = 4

Table 6.4 OTP Read Speed

6.10. GPADC

6.10.1. 简介

BL1824配备了高速低功耗的10位通用模拟数字转换器（GPADC）。它可以在单极性（单端）模式下工作。

6.10.2. 主要特性

- 10 位动态模数转换器，转换时间 8 微秒
- 最大采样率 125k 样本/秒
- 单端输入
- 7 个单端外部输入通道
- 电池监控功能
- 偏移和零点校准
- 支持电压输入范围：[0.2V,1.3V)

6.10.3. GPADC 寄存器映射

Offset	Reset	Name	Description
0x002	0x145050	DLY_CFG	配置 pd 时间
0x004	0x1	ADC_CFG0	开始 AD 转换
0x005	0x1F400045	ADC_CFG1	配置通道和序列时间
0x006	0x10800	ADC_CFG2	配置通道和序列
0x008	0x0	ADC_SW_TRIGGER	触发器
0x00C	0x0	CH_0_CFG	通道 0 gpadc_mode_verf and gpadc_bp
0x00D	0x0	CH_1_CFG	通道 1 gpadc_mode_verf and gpadc_bp
0x00E	0x0	CH_2_CFG	通道 2 gpadc_mode_verf and gpadc_bp
0x00F	0x0	CH_3_CFG	通道 3 gpadc_mode_verf and gpadc_bp
0x010	0x0	CH_4_CFG	通道 4 gpadc_mode_verf and gpadc_bp
0x011	0x0	CH_5_CFG	通道 5 gpadc_mode_verf and gpadc_bp
0x012	0x0	CH_6_CFG	通道 6 gpadc_mode_verf and gpadc_bp
0x013	0x0	CH_7_CFG	通道 7 gpadc_mode_verf and gpadc_bp
0x014	0x0	GAIN_ERR_REG	增益误差数据
0x015	0x0	VOS_REG	vos 数据
0x016	0x0	VOS_TEMP_REG	vos_temp 数据

0x01C	0x0	CH_0_DATA	通道 0 Adc 数据
0x01D	0x0	CH_1_DATA	通道 1 Adc 数据
0x01E	0x0	CH_2_DATA	通道 2 Adc 数据
0x01F	0x0	CH_3_DATA	通道 3 Adc 数据
0x020	0x0	CH_4_DATA	通道 4 Adc 数据
0x021	0x0	CH_5_DATA	通道 5 Adc 数据
0x022	0x0	CH_6_DATA	通道 6 Adc 数据
0x023	0x0	CH_7_DATA	通道 7 Adc 数据

7. 通讯系统

7.1. 支持的功能

BL1824 片上蓝牙系统兼容蓝牙标准 5.1.

7.2. 无线收发器

蓝牙低能耗协议的射频部分由无线电收发器实现，它与蓝牙 5.1 物理层（PHY）一起，提供了可靠的无线通信。所有的射频模块都由片上低压差线性稳压器（LDO）供电。蓝牙低能耗无线电包括接收器、发射器、合成器、接收/发射组合模块以及偏置 LDO。

7.2.1. 蓝牙射频接收器

BL1824 接收器采用低中频架构。射频信号首先通过一个集成变频器，该变频器是接收器和发射器共享的。变频器驱动一个差分可变增益的低噪声放大器（LNA），LNA 进行小信号放大，然后通过低中频下变频混频器。混频器相当于一个三阶复数带通滤波器（BPF），用于执行信道选择和镜像抑制。然后，中频信号由两个逐次逼近式模数转换器（SAR ADCs）进行数字化，以便在数字域进行进一步的信号处理。

7.2.2. 蓝牙射频发生器

BL1824 发射器采用直接调制架构。数字基带信号直接调制到被称为两点调制的 VCO 和 PLL 的分频器上，经过三级 B 类功率放大器后，通过天线输出射频信号。

7.2.3. 频率合成器

BL1824 频率合成器是一个完全集成的 Σ Δ 分数 N 锁相环，用于将 VCO 锁定到参考晶体振荡器。该合成器使用几个集成的线性稳压器，以更好地隔离各个模块。

7.3. 蓝牙基带

BLE（蓝牙低能耗）核心是一个符合蓝牙 5.1 标准的蓝牙基带控制器，与蓝牙智能规范兼容，负责数据包的编码/解码和数据帧的处理。

7.3.1. 主要功能

- 支持所有设备类别（广播者，中心，观察者，外围设备）
- 所有数据包类型（广播/数据/控制）
- 加密（AES / CCM）
- 比特流处理（CRC，白化）
- 频率跳变计算
- 支持 32.768kHz 的低功耗模式

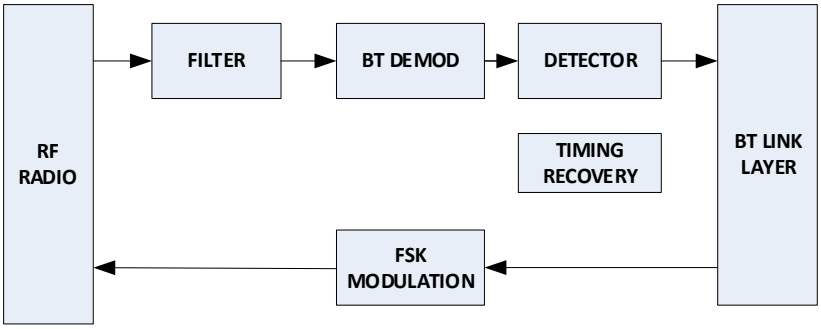


图 7.1 BL1824 BT 基带

7.4. 性能

7.4.1. BLE 接收机性能

[供电电压 = 3.3V @ 25℃]

Parameter		Min	Typ	Max	Unit
灵敏度			-97		dBm
最大接收信号		-		0	dBm
同频信道选择灵敏度		-		6	dB
临频信道选择灵敏度 Note: F0=2440 MHz	F = F0+1MHz	-	-	-18	dB
	F = F0 -1MHz	-	-	-23	dB
	F = F0+2MHz	-	-	-27	dB
	F = F0-2MHz	-	-	-23	dB
	F = F0+3MHz	-	-	-35	dB
	F = F0-3MHz	-	-	-29	dB

表 7.1 BL1824 BLE 接收机性能

7.4.2. BLE 发射机性能

[工作电压 = 3.3V @ 25℃]

Parameter	Min	Typ	Max	Unit

Parameter		Min	Typ	Max	Unit
最大传输功率		-	10	-	dBm
射频功率范围		-20	-	10	dBm
射频功率控制精度			1		dB
ACP	F = F0±2MHz	-	-		dBm
Note: F0=2440MHz	F = F0±>3MHz	-	-		dBm
Δf1avg maximum modulation		225	250	275	kHz
Δf2max maximum modulation		100%			
Δf2avg/Δf1avg		0.84			
频率精度			4.03		kHz
频率偏移			4.02		KHz
频率漂移			-3.31		KHz
频率漂移率			-3.13		KHz/50us
初始频率漂移			-2.25		KHz
二次谐波失真			-43		dBm
三次谐波失真		-	-50		dBm

表 7.2 BL1824 BLE 发射机性能

8. 应用原理图

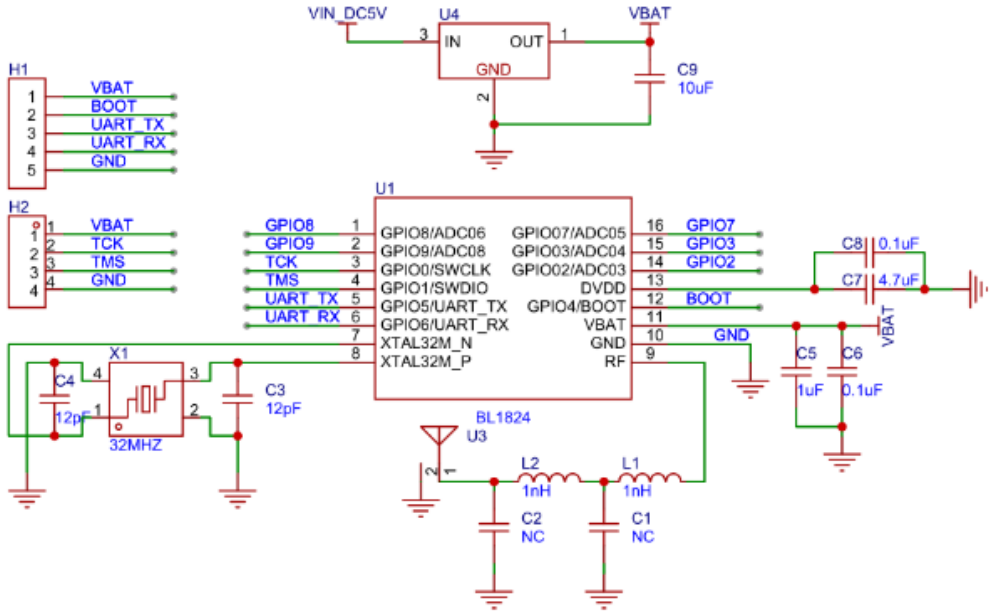
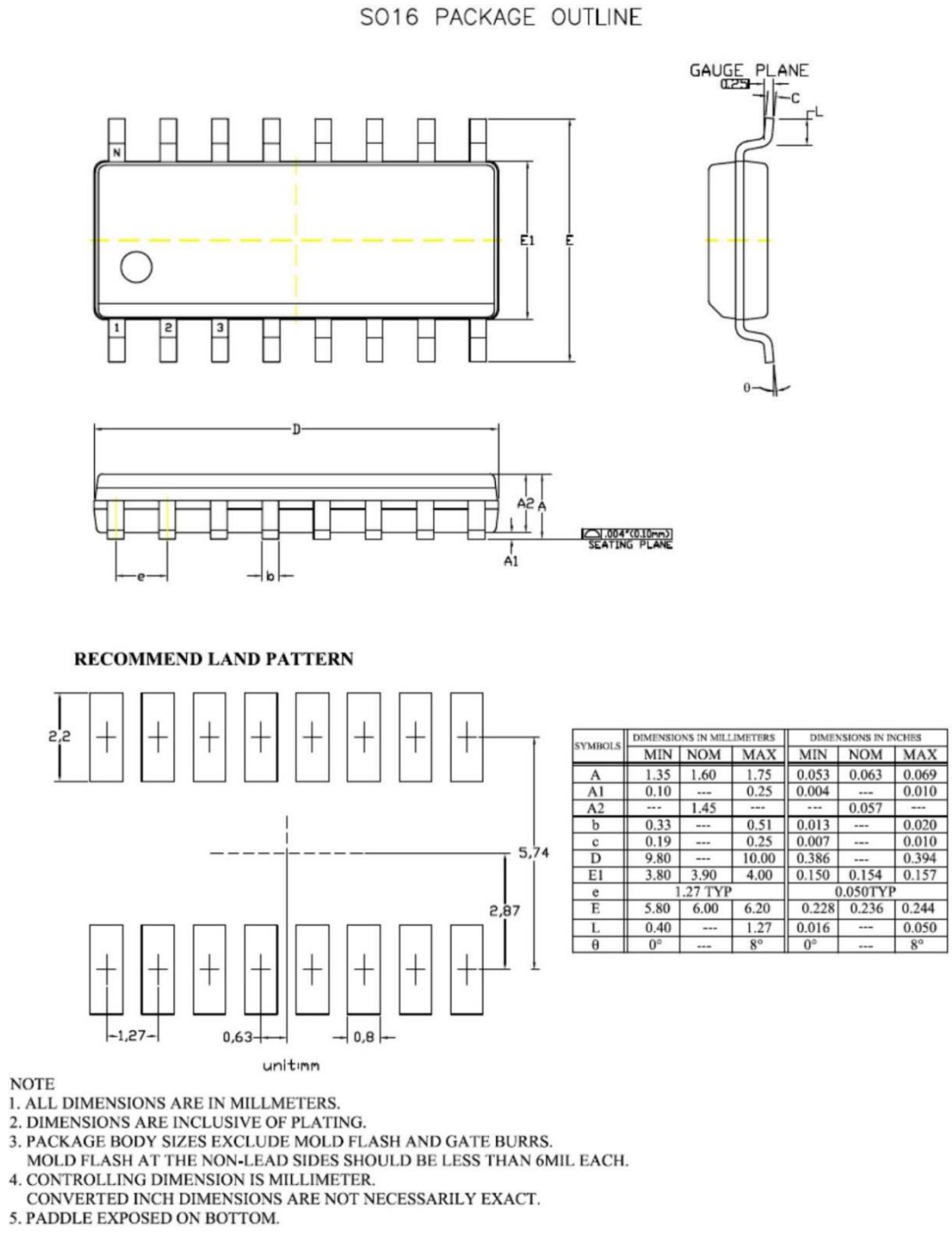


图 8.1 BL1824 原理图

9. 封装信息

BL1824 的 SOP16 封装如下:



SYMBOLS	DIMENSIONS IN MILLIMETERS			DIMENSIONS IN INCHES		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.35	1.60	1.75	0.053	0.063	0.069
A1	0.10	---	0.25	0.004	---	0.010
A2	---	1.45	---	---	0.057	---
b	0.33	---	0.51	0.013	---	0.020
c	0.19	---	0.25	0.007	---	0.010
D	9.80	---	10.00	0.386	---	0.394
E1	3.80	3.90	4.00	0.150	0.154	0.157
e	1.27 TYP			0.050TYP		
E	5.80	6.00	6.20	0.228	0.236	0.244
L	0.40	---	1.27	0.016	---	0.050
θ	0°	---	8°	0°	---	8°

NOTE
1. ALL DIMENSIONS ARE IN MILLMETERS.
2. DIMENSIONS ARE INCLUSIVE OF PLATING.
3. PACKAGE BODY SIZES EXCLUDE MOLD FLASH AND GATE BURRS.
   MOLD FLASH AT THE NON-LEAD SIDES SHOULD BE LESS THAN 6MIL EACH.
4. CONTROLLING DIMENSION IS MILLIMETER.
   CONVERTED INCH DIMENSIONS ARE NOT NECESSARILY EXACT.
5. PADDLE EXPOSED ON BOTTOM.

图 9.1 BL1824 **sop16 封装**

10. 缩略语

Name	Description
ADC	Analog to Digital Converter
AGC	Automatic Gain Control
AON	Always-on
APB	Advanced Peripheral Bus
BB	Base band
BLE	Bluetooth Low Energy
BOD	Brown-out Detector
IFS	Inter Frame Spacing
LDO	Low Dropout
LNA	Low Noise Amplifier
LPD	Low Power Domain
NVM	Non-volatile memory
PLL	Phase Locked Loop
PMU	Power Management Unit
RNG	RING Oscillator
SOC	System-on-chip
TPMS	Tire pressure monitor system
W1C	Write 1 to clear
XO	Crystal Oscillator
Typ	Typical
SNR	Signal to Noise Ratio
PA	Power Amplifier
IRQ	Interrupt Request
LSB	Least Significant Bit
MSB	Most Significant Bit
DFE	Digital Front End

表 10.1 术语表和缩略语

11. 参考文档

参考文档列表:

文档	描述
BL1824 应用手册	向开发者介绍 BL1824 最小系统板相关的资源.
BL1824 SDK 用户指导	向开发者介绍 BL1824 相关的资源

表 111.1 参考文档