

# CH561 使用手册

版本：V1.00

<http://wch.cn>

## 如何阅读本说明书

关于缩写：

本说明书中在对寄存器的描述中使用了下列缩写：

缩写词	描述
RF	读取值为定值，不受复位影响
RO	只读
WO	只写(读取值为 0 或无效)
RZ	只读，读取完之后自动清 0
WZ	写清 0
RW	可读可写
RW1	读/写 1 清 0
WA	只写（安全模式下），读取值为 0 或无效
RWA	可读写（安全模式下）
h	表示 16 进制数
b	表示 2 进制数
o	表示 8 进制数

关于地址：

以太网相关寄存器物理起始地址为：0x0040 6000

本说明书描述中除非特殊说明，否则地址均指偏移地址。

关于术语：

本说明书用到的与以太网相关的首字母缩写词、简写以及定义：

术语	定义
AHB	先进的高性能总线
CRC	循环冗余校验
DMA	直接存储器访问
FCS	帧校验序列
Fragment	一个以太网帧或其中的一部分，一个以太网帧可以是一个或多个片段(fragment)
Frame	一个以太网帧由目标地址、源地址、长度/类型区、有效载荷以及帧检验序列组成
MAC	媒体访问控制器
MI	媒体独立接口
Packet	通过以太网传输的帧：一个包由导言、起始帧定界符和以太网帧组成
PHY	以太网物理层
Rx	接收
Tx	发送
WoL	LAN 上唤醒
TCP/IP	传输控制协议/网际协议

## 第 1 章 概述

CH561 系列单片机是 32 位 RISC 精简指令集 CPU，指令集兼容 ARMv5TE，支持 16 位 Thumb 指令和增强 DSP 指令。默认系统主频为 100MHz，最高可达 130MHz。高度集成的外设以及高性能，使其可以广泛的应用于各种嵌入式应用。

CH561 的外设组件包含最高 64K 的 FLASH 闪存、32K 的 SRAM、28K 的 DataFlash、百兆以太网、2 个 SPI 接口、2 个 UART 接口、1 通道 10 位 ADC、4 个通用定时器、8 位被动并行接口、在系统编程 ISP 接口和多达 27 个通用 I/O 管脚。

### 1.1 功能特性

- 内核：  
32 位 RISC 指令集 CPU，兼容 ARMv5TE 指令集，支持 16 位 Thumb 指令和增强 DSP 指令，默认系统主频为 100MHz。
- FLASH 和 RAM：  
CH561 片内 FLASH 最高为 64K，用于保存程序代码，支持写保护。FLASH 闪存拥有 10 万次擦写、20 年保存能力，支持 ICP/ISP 升级；片内 SRAM 共 32K，用于数据缓冲。
- DataFlash：  
片内 DataFlash 共 28K，用于保存非易失性数据。
- 以太网：  
内置 MAC 和基于 DSP 的 10/100Base-TX 快速 Ethernet 收发器 (PHY)，支持 10/100Mbps 双绞线网络通讯，兼容 IEEE 802.3, 802.3u 和 FDDI-TP-PMD，支持 Wake-On-LAN 功能，收发各独立的 2K 字节 FIFO，支持 DMA。  
每颗芯片都内置 IEEE 分配的唯一 MAC 地址。
- SPI：  
2 组 SPI 控制器 SPI0、SPI1，速度可调，最高可达系统频率 SCLK 的一半约 50MHz，支持输入输出单工复用。SPI0 支持 Master/Slave 主从模式，32 级 FIFO，支持 DMA；SPI1 只支持主机模式，16 级 FIFO。
- UART：  
2 组兼容 16C550 的异步串口 UART，速度可调，最高达系统主频 SCLK 的八分之一约 16Mbps，一组支持全部 Modem 信号，16 级 FIFO；另一组只支持 CTS 和 RTS，但支持 9 位串口模式下总线地址自动比较和 SIR 之 IrDA 编解码，32 级 FIFO。
- ADC：  
CH561 支持 1 通道 10 位 A/D 转换器，采样率可调，最高达每秒 1M 次，8 级 FIFO，支持设定参考值在到达后触发中断，支持两通道自动轮测，可用软件对信号进行算法解码，支持 DMA。
- TIMER：  
4 组 28 位定时器，支持输入信号宽度采样/信号捕捉(高/低/翻转)，支持 PWM 输出，8 级 FIFO，其中 3 组支持 DMA。
- 低功耗睡眠：  
支持低功耗睡眠方式，支持部分 GPIO 等外部唤醒。
- 芯片 ID 号：  
每个芯片具有 64bit 唯一的 ID 识别号，便于通讯识别或者数据加密。
- GPIO：  
提供 27 个 GPIO 引脚，部分引脚支持上拉电阻、下拉电阻、施密斯输入、两种驱动能力，部分引脚支持电平或者边沿中断。
- 电源：  
CPU 操作电压范围：1.65V---1.95V(1.8V±10%)；  
I/O 操作电压范围：3.0V---3.6V(3.3V±10%)，部分 I/O 口可承受 5V 电压。

## 1.2 结构框图

以下为 CH561 芯片简化内部结构框图。

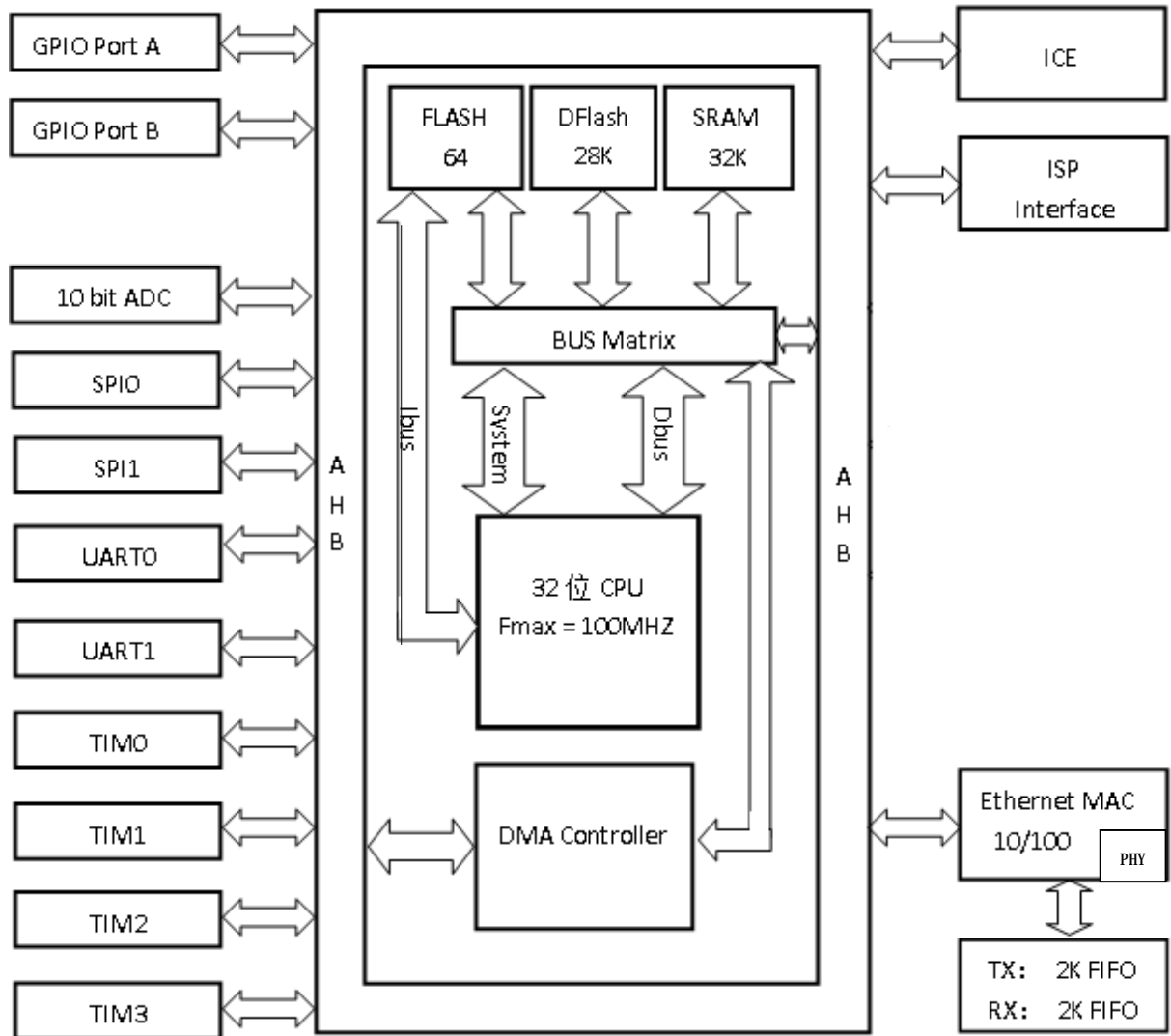


图 1-1 CH561 简化结构框图

## 1.3 芯片封装

芯片型号	芯片封装	
	名称	描述
CH561Q	LQFP64M	LQFP 封装; 64 脚; 本体 10x10mm

## 第 2 章 引脚信息

### 2.1 引脚排列

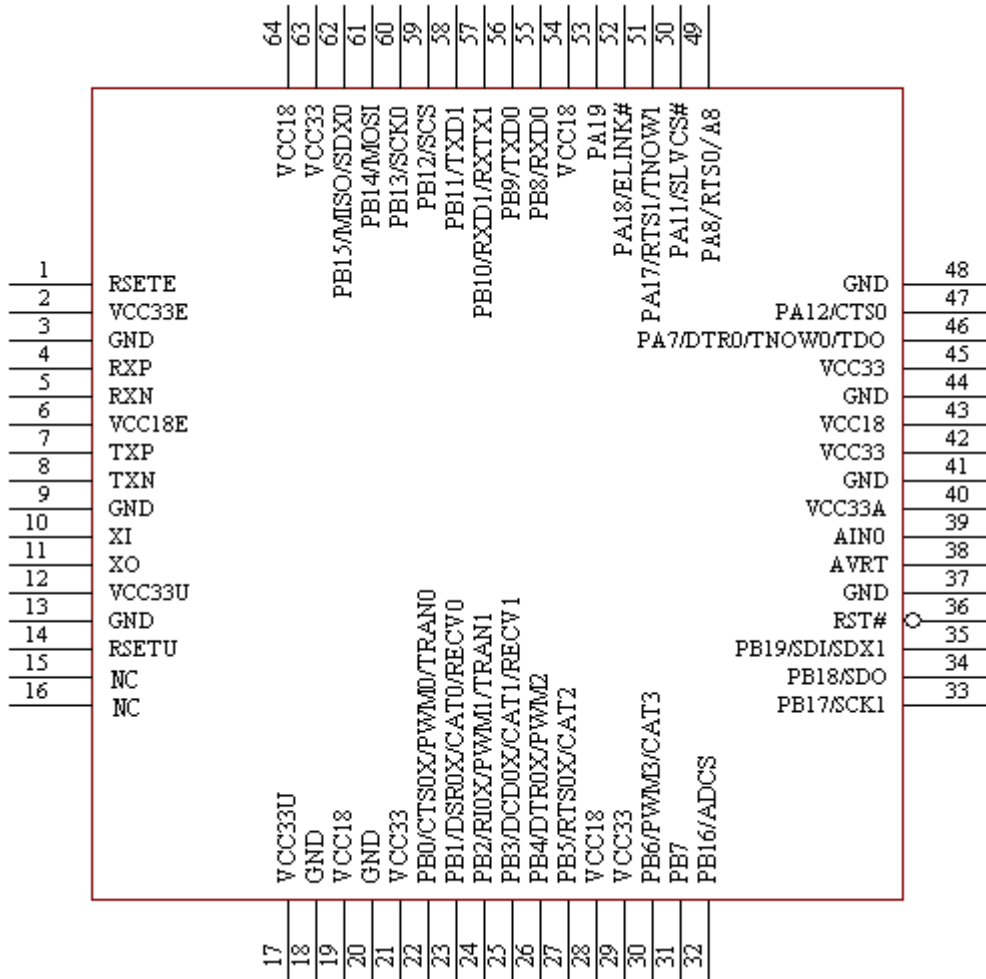


图 2-1 LQFP64 封装引脚排列

### 2.2 引脚描述

管脚名称	类型	I/O 电平	主功能/ 其它功能	功能描述
1	RSETE	I	RSETE	对地接电阻，用于 ETH- PHY 电阻值范围为 12K 到 18K，12K 时性能最优，18K 时 功耗最低，默认可接 15K
2	VCC33E	P	VCC33E	<b>3.3V 电源：</b> 以太网部分电源
3	GND	P	GND	<b>地：</b> 公共接地端，0V 参考点
4	RXP	ETH	RXP	以太网 RXP 信号
5	RXN	ETH	RXN	以太网 RXN 信号

6	VCC18E	P		VCC18E	1.8V 电源： 以太网部分电源
7	TXP	ETH		TXP	以太网 TXP 信号
8	TXN	ETH		TXN	以太网 TXN 信号
9	GND	P		GND	地： 公共接地端，0V 参考点
10	XI	I		XI	晶体振荡输入端
11	XO	I/O		XO	晶体振荡反向输出端
12	VCC33U	P		VCC33U	3.3V 电源
13	GND	P		GND	地： 公共接地端，0V 参考点
14	RSETU	I		RSETU	12K 电阻到地
15	NC	-		NC	悬空
16	NC	-		NC	悬空
17	VCC33U	P		VCC33U	3.3V 电源
18	GND	P		GND	地： 公共接地端，0V 参考点
19	VCC18	P		VCC18	1.8V 内核电源： 内部电路的电源
20	GND	P		GND	地： 公共接地端，0V 参考点
21	VCC33	P		VCC33	3.3V 端口电源： I/O 口电源
22	PB0	I/O		PB0/CTS0X /PWM0	PB0：通用双向数字 I/O 引脚 CTS0X：UART0 的 MODEM 信号，清除发送 PWM0：脉宽调制输出 0 引脚
23	PB1	I/O		PB1/DSR0X /CAT0	PB0：通用双向数字 I/O 引脚 DSR0X：UART0 的 MODEM 信号，数据装置就绪 CAT0：定时器 0 捕获输入引脚
24	PB2	I/O		PB2/RIOX /PWM1	PB2：通用双向数字 I/O 引脚 RIOX：UART0 的 MODEM 信号，振铃指示 PWM1：脉宽调制输出 1 引脚
25	PB3	I/O		PB3/DCDOX /CAT1	PB3：通用双向数字 I/O 引脚 DCDOX：UART0 的 MODEM 信号，载波检测 CAT1：定时器 1 捕获输入引脚
26	PB4	I/O		PB4/DTROX /PWM2	PB4：通用双向数字 I/O 引脚 DTROX：UART0 的 MODEM 信号，数据终端就绪 PWM2：脉宽调制输出 2 引脚
27	PB5	I/O		PB5/RTSOX /CAT2	PB5：通用双向数字 I/O 引脚 RTSOX：UART0 的 MODEM 信号，请求发送

					CAT2: 定时器 2 捕获输入引脚
28	VCC18	P		VCC18	1.8V 内核电源: 内部电路的电源
44	GND	P		GND	地: 公共接地端, 0V 参考点
29	VCC33	P		VCC33	3.3V 端口电源: I/O 口电源
30	PB6	I/O		PB6/PWMB / CAT3	PB6: 通用双向数字 I/O 引脚 PWMB: 脉宽调制输出 3 CAT3: 定时器 3 捕获输入引脚
31	PB7	I/O		PB7	PB7: 通用双向数字 I/O 引脚
32	PB16	I/O		PB16/ADCS	PB16: 通用双向数字 I/O 引脚 ADCS: 在 ADC 通道自动切换模式指示当前通道状态; 在非自动切换模式输出采样脉冲
33	PB17	I/O		PB17/SCK1	PB17: 通用双向数字 I/O 引脚 SCK1: SPI1 串行时钟输出引脚
34	PB18	I/O		PB18/SDO	PB18: 通用双向数字 I/O 引脚 SDO: SPI1 串行数据输出引脚
35	PB19	I/O		PB19/SDI /SDX1	PB19: 通用双向数字 I/O 引脚 SDI: SPI1 串行数据输入引脚 SDX1: SPI1 单工模式下串行数据输入输出引脚
36	RST#	I		RST#	外部复位输入, 内置上拉电阻 该管脚的低电平将器件复位, 并使 I/O 口和外围功能恢复默认状态, 处理器从地址 0 开始执行
37	GND	P		GND	地: 公共接地端, 0V 参考点
38	AVRT	P		AVRT	ADC 参考电源正极, 标称电压与 VCC33 相同, 但应当互相隔离以减少噪声干扰
39	AIN0	I		AIN0	A/D 转换输入通道 0 引脚
40	VCC33A	P		VCC33A	ADC 模拟 3.3V 电源 标称电压和 VCC33 相同, 但应当互相隔离以减少噪声干扰
41	GND	P		GND	地: 公共接地端, 0V 参考点
42	VCC33	P		VCC33	3.3V 电源: ADC 部分电源
43	VCC18	P		VCC18	1.8V 内核电源: 内部电路的电源
45	VCC33	P		VCC33	3.3V 端口电源: I/O 口电源
46	PA7	I/O		PA7/DTR0/ TNOWO/TDO /A7	PA7: 通用双向数字 I/O 引脚 DTR0: UART0 的 MDDEM 信号, 数据终端就绪 TNOWO: UART0 的单工 RS485 收发使能引脚 A7: 外部存储器地址线 7
47	PA12	I/O		PA12/CTS0 /A12	PA12: 通用双向数字 I/O 引脚 CTS0: UART0 的 MDDEM 信号, 清除发送 A12: 外部存储器地址线 12

48	GND	P		GND	<b>地：</b> 公共接地端，0V 参考点
49	PA8	I/O	FT	PA8/RTS0/A8	PA8： PA 口通用 I/O RTS0： UART0 的 MODEM信号，请求发送 A8： 外部存储器地址线 8
50	PA11	I/O	FT	PA11/SLVCS# /A11	PA11： 通用双向数字 I/O 引脚 SLVCS#： 被动并口片选输入引脚 A11： 外部存储器地址线 11
51	PA17	I/O	FT	PA17/RTS1 /TNOW1/A17	PA17： 通用双向数字 I/O 引脚 RTS1： UART1 的 MODEM信号，请求发送 TNOW1： UART1 的单工 RS485 收发使能引脚 A17： 外部存储器地址线 17
52	PA18	I/O	FT	PA18/ELINK# /A18	PA18： 通用双向数字 I/O 引脚 ELINK#： 以太网通讯连接指示灯驱动引脚 A18： 外部存储器地址线 18
53	PA19	I/O	FT	PA19 /A19	PA19： 通用双向数字 I/O 引脚 A19： 外部存储器地址线 19
54	VCC18	P		VCC18	<b>1.8V 内核电源：</b> 内部电路的电源
55	PB8	I/O	FT	PB8/RXD0	PB8： 通用双向数字 I/O 引脚 RXD0： UART0 接收器输入引脚
56	PB9	I/O	FT	PB9/TXD0	PB9： 通用双向数字 I/O 引脚 TXD0： UART0 发送器输出引脚
57	PB10	I/O	FT	PB10/RXD1 /RXTX1	PB10： 通用双向数字 I/O 引脚 RXD1： UART1 接收器输入引脚 RXTX1： UART1 单工收发输入输出引脚
58	PB11	I/O	FT	PB11/TXD1	PB11： 通用双向数字 I/O 引脚 TXD1： UART1 发送器输出引脚
59	PB12	I/O	FT	PB12/SCS	PB12： 通用双向数字 I/O 引脚 SCS： SPI0 从机片选输入引脚
60	PB13	I/O	FT	PB13/SCK0	PB13： 通用双向数字 I/O 引脚 SCK0： SPI0 串行时钟引脚，主机输出/从机输入
61	PB14	I/O	FT	PB14/MDSI	PB14： 通用双向数字 I/O 引脚 MDSI： SPI0 串行数据引脚，主机输出/从机输入
62	PB15	I/O	FT	PB15/MISO /SDX0	PB15： 通用双向数字 I/O 引脚 MISO： SPI0 串行数据引脚，主机输入/从机输出 SDX0： SPI0 单工模式下串行数据输入输出引脚
63	VCC33	P		VCC33	<b>3.3V 端口电源：</b> I/O 口电源
64	VCC18	P		VCC18	<b>1.8V 内核电源：</b> 内部电路的电源

注：

- (1)、I： 输入；O： 输出；P： 电源；ETH： 以太网信号；Hi Z： 高阻；
- (2)、FT： 兼容 5V；
- (3)、其中部分功能仅在部分封装中支持，具体请参看引脚描述表。



## 第 3 章 存储器映射

### 3.1 存储器映射图

CH561 包含有一个 4GB 的地址空间，各存储器映射主要包含几个不同区域，见下图所示。

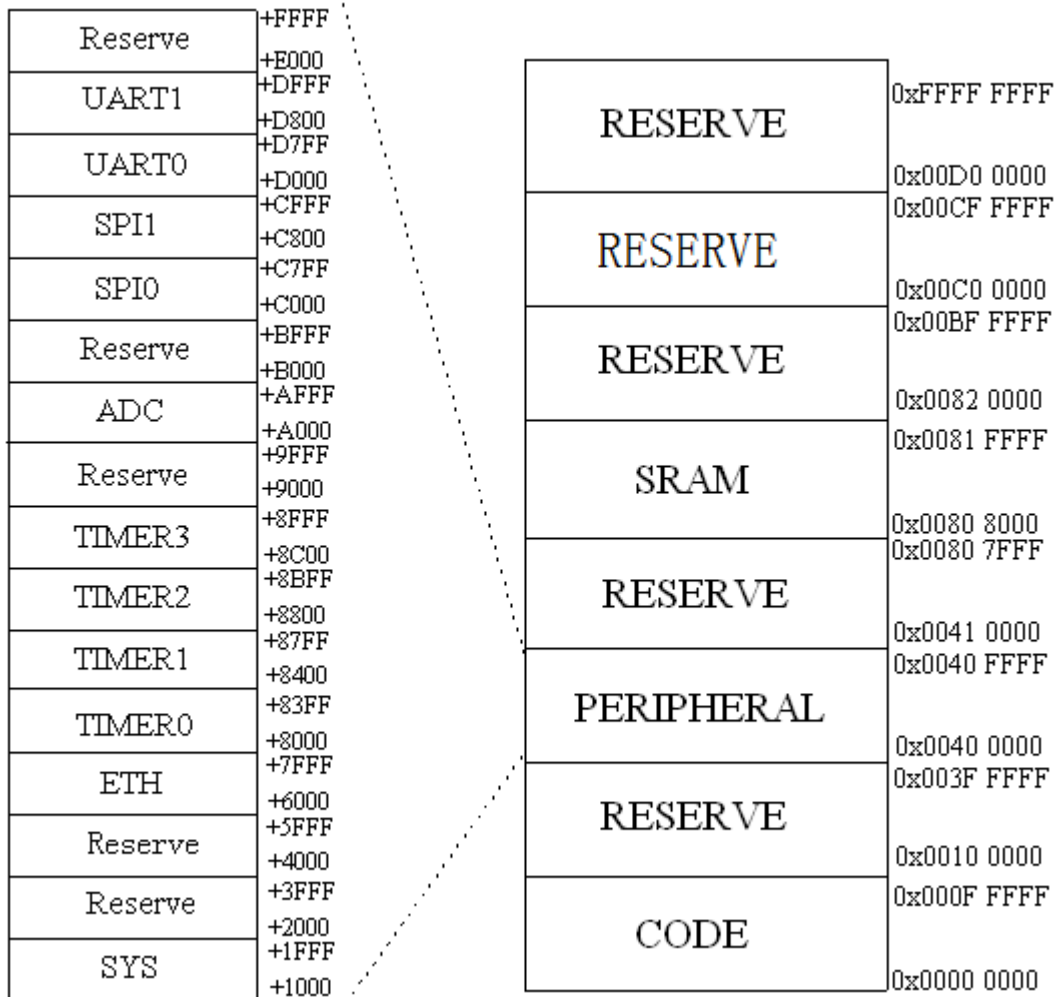


图 3-1 存储器映射图

### 3.2 存储器映射表

各存储器映射区域地址范围如下表所示：

地址范围	用途	描述
0x0000 0000-0x000F FFFF	片上非易失性存储器	Flash 存储器(1MB)
0x0010 0000-0x003F FFFF	保留	-
0x0040 0000-0x0040 FFFF	各种外设	多个外设模块
0x0041 0000-0x007F FFFF	保留	-
0x0080 8000-0x0081 FFFF	片上 SRAM, 通常用于存储数据	32K
0x0082 0000-0x00BF FFFF	保留	-
0x00C0 0000-0x00CF FFFF	保留	-
0x00D0 0000-0xFFFF FFFF	保留	-

### 3.3 外设地址分配

CH561 主要包含 12 个外设，每个外设占用一定的地址空间，下表为各个外设的地址分配表。

外设编号	外设名称	外设基址
1	SYS	0x0040 1000
2	ETH	0x0040 6000
3	TIMER0	0x0040 8000
4	TIMER1	0x0040 8400
5	TIMER2	0x0040 8800
6	TIMER3	0x0040 8C00
7	ADC	0x0040 A000
8	SPI0	0x0040 C000
9	SPI1	0x0040 C800
10	UART0	0x0040 D000
11	UART1	0x0040 D800

## 第 4 章 电源和复位控制

### 4.1 电源控制

CH561 需要外部提供 2 组工作电压，CPU 工作电压为 1.8V，I/O 操作电压为 3.3V。由图 2-1 和图 2-2 的封装引脚排列可以看出，外部提供了多组电源和地。另外，为提高转换的精确度，ADC 使用了一个独立的外部的电源 (AVRT 和 AVRB) 供电，过滤和屏蔽来自印刷电路板上的毛刺干扰。

在系统或电源复位后，CH561 处于运行状态。当 CPU 不需要继续运行时，或某些功能模块不需要使用时，可以关闭这些模块的时钟或电源，以降低功耗。具体见第 5 章时钟控制关于寄存器部分的讲解。

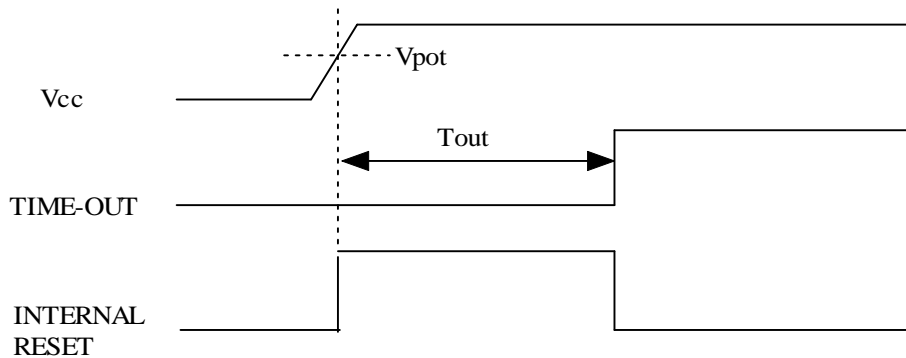
### 4.2 复位控制

CH561 支持 4 种复位形式，分别为电源上电复位，外部手工复位，内部软件复位和看门狗复位。其中前 3 种复位会导致重新加载配置信息，并重新加载程序代码到 RAM 缓冲区中，加载时间约为 3mS。

寄存器 R8\_GLOB\_RESET\_KEEP 和位 RB\_GLOB\_BOOT\_APP，只在上电复位时被复位，而不受其它 3 种复位影响。

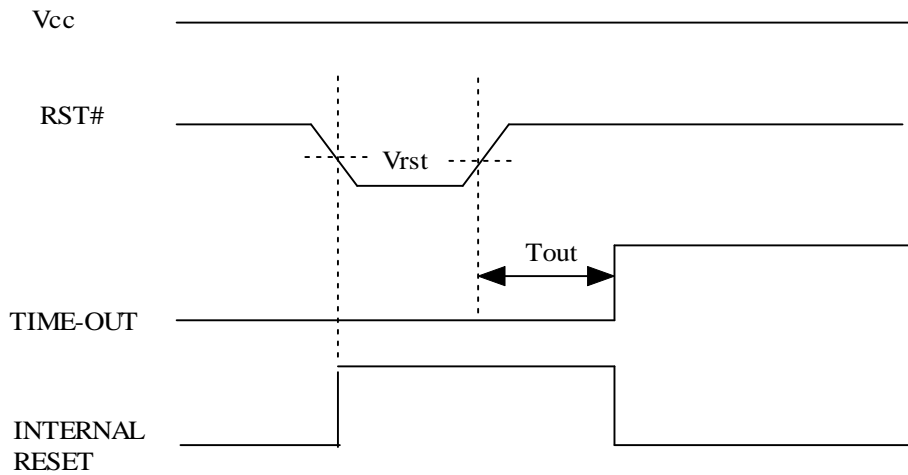
#### 4.2.1 电源上电复位

当电源电压低于上电复位门限  $V_{pot}$  时，CH561 发生复位。下图为 CH561 上电复位图。



#### 4.2.2 外部手工复位

外部手工复位由外加于 RST# 引脚的低电平产生。当复位低电平持续时间大于最小复位脉冲宽度 ( $T_{rst}$ ) 时即触发 CH561 芯片进行复位。



#### 4.2.3 内部软件复位

CH561 提供了内部软件复位功能，以便在某些特定情况下，不需要外部干预进行软件复位。设置全局复位配置寄存器(R8\_GLOB\_RST\_CFG)的位 RB\_GLOB\_FORCE\_RST 为 1，即可实现软件复位。该位会自动清 0。

#### 4.2.4 看门狗复位

看门狗定时器溢出时将产生内部复位脉冲，控制 CH561 进行复位。看门狗周期约为  $2^{23}=8388608$  个主时钟周期，对于 100MHz 主频约为 84ms。清零 R8\_WDOG\_CLEAR 可以复位看门狗定时器。

#### 4.2.5 复位特性

符号	参数	最小值	典型值	最大值	单位
Vpot	上电复位门限电压	1.0	1.5	1.6	V
Vrst	RST#门限电压	1.3	1.5	-	V
Trst	外部复位输入 RST#有效信号宽度	20	-	-	ns
Tout	外部复位输入后的复位延时+加载时间	4+3	8+3	10+3	mS

## 第 5 章 时钟控制

### 5.1 时钟框图

CH561 内部各功能模块时钟结构如下图所示：

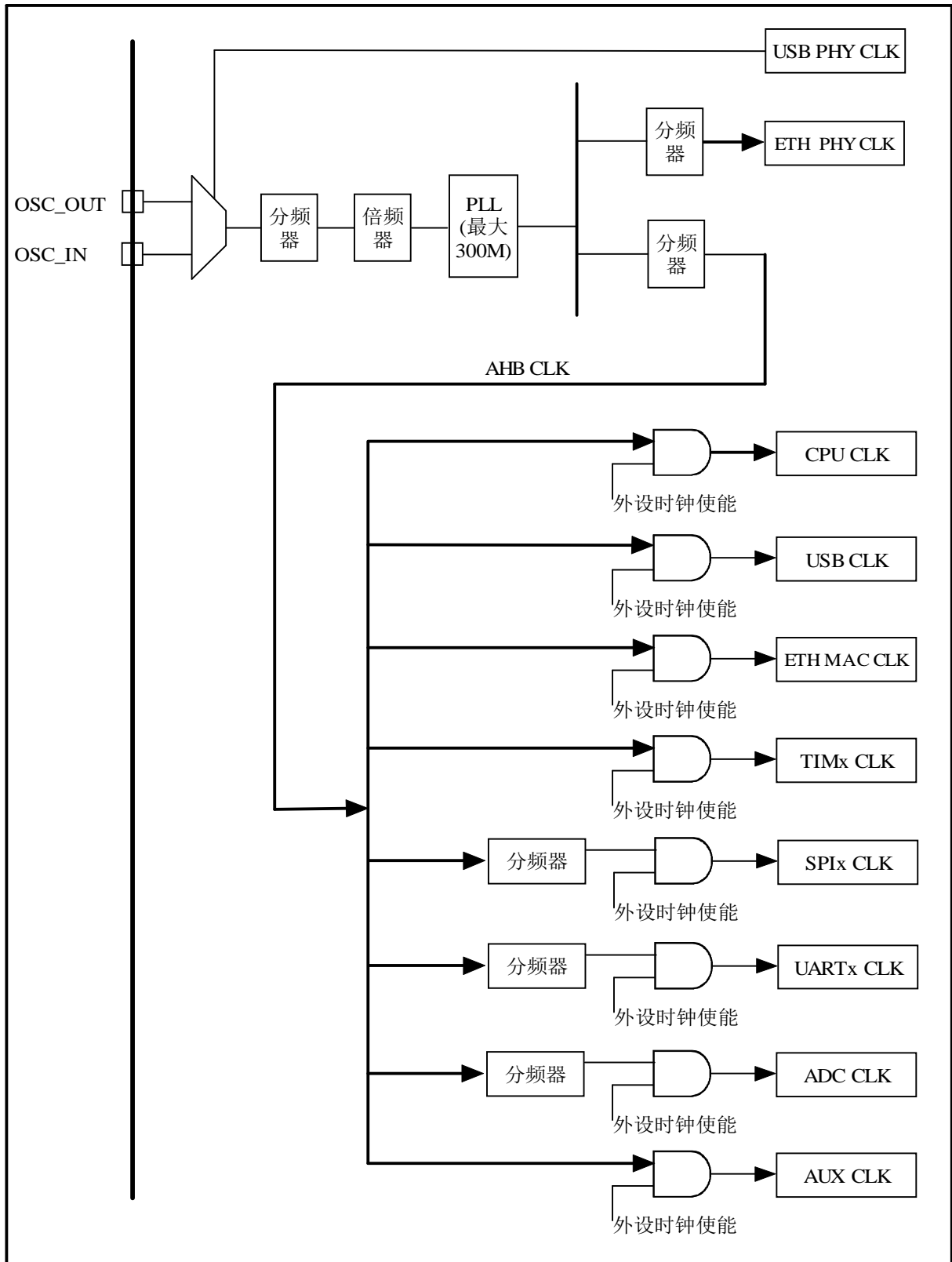


图 5-1 时钟结构框图

外部时钟输入 CH561 后, 先经过分频器进行分频(R8\_PLL\_PRE\_DIV)得到参考频率, 再经过倍频器进行倍频(R8\_PLL\_LOOP\_DIV)得到 PLL 时钟。PLL 时钟经过 RB\_PLL\_ETH\_DIV 分频后, 作为以太网时钟, 该时钟必须为 25MHZ。PLL 时钟经过 RB\_PLL\_SYS\_DIV 分频后, 作为 CPU 主时钟, 建议主时钟频率范围为: 30MHZ- 130MHZ。各部分时钟都有对应的寄存器或位, 进行单独的打开或关闭。为降低芯片功耗, 可以关闭不使用的功能模块。

## 5.2 寄存器描述

时钟及CPU控制相关寄存器物理起始地址为: 0x0040 0000

时钟及 CPU 控制相关寄存器列表(地址范围: 0x00--0x1E)

表 5-1 时钟及 CPU 控制相关寄存器列表

名称	偏移地址	描述	复位值
R8_SAFE_ACCESS_SIG	0x00	安全访问标记寄存器	8h00
R8_SAFE_ACCESS_ID	0x02	安全访问 ID 寄存器	8h01
R8_WDOG_CLEAR	0x03	看门狗清 0 寄存器	8h00
R8_GLOB_MEM_CFG	0x04	全局配置寄存器	8h80
R8_GLOB_LOCK_PORT	0x05	全局端口锁定配置寄存器	8h00
R8_GLOB_RST_CFG	0x06	全局复位配置寄存器	8h40
R8_GLOB_RESET_KEEP	0x07	复位保持寄存器	8h00
R8_PLL_EXT_CFG	0x08	PLL 配置寄存器	8h40
R8_PLL_OUT_DIV	0x09	PLL 输出时钟分频寄存器	8h40
R8_PLL_PRE_DIV	0x0A	PLL 预分频寄存器	8h00
R8_PLL_LOOP_DIV	0x0B	PLL 回环分频寄存器	8h00
R8_SLP_CLK_OFF0	0x0C	时钟控制寄存器 0	8h00
R8_SLP_CLK_OFF1	0x0D	时钟控制寄存器 1	8h00
R8_SLP_WAKE_CTRL	0x0E	休眠控制寄存器	8h00
R8_SLP_CTRL_PLL	0x0F	PLL 休眠控制寄存器	8h00
R8_MISC_CTRL_ETH	0x1C	以太网控制寄存器	8h00

安全访问标记寄存器(R8\_SAFE\_ACCESS\_SIG):

名称	位	访问	描述	复位值
R8_SAFE_ACCESS_SIG	[7: 0]	WO	安全访问标记寄存器, 部分寄存器(访问属性为 RWA)为保护寄存器, 必须进入安全访问模式才能进行读写操作。对该寄存器先写入 0x57, 再写入 0xA8 即可进入安全访问模式, 并且限时约 110 个主时钟周期, 超过则自动保护	8h0

安全访问 ID 寄存器(R8\_SAFE\_ACCESS\_ID):

名称	位	访问	描述	复位值
R8_SAFE_ACCESS_ID	[7: 0]	RF	安全访问 ID 寄存器	8h01

看门狗清 0 寄存器(R8\_WDOG\_CLEAR):

名称	位	访问	描述	复位值
R8_WDOG_CLEAR	[7: 0]	WO	看门狗清 0 寄存器	8h00

全局配置寄存器(R8\_GLOB\_MEM\_CFG):

名称	位	访问	描述	复位值
----	---	----	----	-----

RB_GLOB_CFG_FLAG	[7: 6]	RO	<b>全局配置标志位</b> 写操作时, 该域必须写入 2b10	2h2
Reserved	5	-	保留	-
RB_GLOB_BOOT_APP	4	RO	<b>引导程序标志位</b> 0: 引导程序状态(或无引导程序) 1: 用户应用程序状态	1h0 1h1
Reserved	[3: 2]	-	保留	-
RB_GLOB_MEM_CFG	[1: 0]	RWA	<b>内存配置域</b> 0X: 64K 代码空间, 32K 数据空间	2h2

全局端口锁定配置寄存器(R8\_GLOB\_LOCK\_PORT):

名称	位	访问	描述	复位值
Reserved	[7: 6]	-	写操作时, 该域必须写入 2b00	2h3
Reserved	[5: 4]	-	保留	-
RB_GLOB_LOCK_PD	3		<b>GPI0 PD 端口锁定位</b> 0: 未锁定 1: 锁定, R32/R8_PB_*全部写保护	1h0
Reserved	2	-	保留	-
RB_GLOB_LOCK_PB	1		<b>GPI0 PB 端口锁定位</b> 0: 未锁定 1: 锁定, R32/R8_PB_*全部写保护	1h0
RB_GLOB_LOCK_PA	0		<b>GPI0 PA 端口锁定位</b> 0: 未锁定 1: 锁定, R32/R8_PA_*全部写保护	1h0

全局复位配置寄存器(R8\_GLOB\_RST\_CFG):

名称	位	访问	描述	复位值
Reserved	[7: 6]	-	写操作时, 该域必须写入 2b01	2h0
RB_GLOB_DEBUG_EN	5	RO	<b>当前调试禁止/使能状态位</b> 0: 当前禁止调试 1: 当前使能调试	1h0
Reserved	[4: 2]	-	保留	-
RB_GLOB_WDOG_EN	1	RWA	<b>看门狗禁止/使能位</b> 0: 禁止看门狗 1: 使能看门狗	1h0
RB_GLOB_FORCE_RST	0	WA/ WZ	<b>软复位位</b> 0: 无动作 1: 强制软件复位芯片	1h0

复位保持寄存器(R8\_GLOB\_RESET\_KEEP):

名称	位	访问	描述	复位值
R8_GLOB_RESET_KEEP	[7: 0]	RW	<b>复位保持寄存器</b> 该寄存器值不受手工复位、软件复位或者看门狗复位影响	8h0

PLL 配置寄存器(R8\_PLL\_EXT\_CFG):

名称	位	访问	描述	复位值
Reserved	[7: 6]	-	写操作时, 该域必须写入 2b01	2h0

Reserved	[5: 4]	-	保留	-
RB_PLL_ETH_OSC	3	RWA	<b>以太网时钟源选择位</b> 0: PLL 产生 1: 外部晶振产生	1h0
RB_PLL_XT_FREQ	2	RWA	<b>PLL 外部晶振频率位</b> 0: 12MHz 1: 30 MHz	1h1
Reserved	[1: 0]	-	保留	-

PLL 输出时钟分频寄存器(R8\_PLL\_OUT\_DIV):

名称	位	访问	描述	复位值
Reserved	[7: 6]	-	写操作时, 该域必须写入 2b10	2h0
RB_PLL_SYS_DIV	[5: 4]	RWA	<b>系统时钟分频系数</b> 有效值为 0-3, 0 代表 4	2h0
RB_PLL_ETH_DIV	[3: 0]	RWA	<b>以太网时钟分频系数</b> 有效值为 1-16	4ha

PLL 预分频寄存器(R8\_PLL\_PRE\_DIV):

名称	位	访问	描述	复位值
Reserved	[7: 3]	-	保留	-
R8_PLL_PRE_DIV	[2: 0]	RWA	<b>预分频系数</b>	3h3

PLL 倍频寄存器(R8\_PLL\_LOOP\_DIV):

名称	位	访问	描述	复位值
Reserved	[7: 6]	-	保留	-
R8_PLL_LOOP_DIV	[5: 0]	RWA	<b>PLL 倍频系数</b>	6h19

时钟控制寄存器 0(R8\_SLP\_CLK\_OFF0):

名称	位	访问	描述	复位值
RB_SLP_CLK_UART1	7	RWA	<b>UART1 时钟控制位</b> 0: UART1 时钟开启 1: UART1 时钟关闭	1h0
RB_SLP_CLK_UART0	6	RWA	<b>UART0 时钟控制位</b> 0: UART0 时钟开启 1: UART0 时钟关闭	1h0
RB_SLP_CLK_SPI1	5	RWA	<b>SPI1 时钟控制位</b> 0: SPI1 时钟开启 1: SPI1 时钟关闭	1h0
RB_SLP_CLK_SPI0	4	RWA	<b>SPI0 时钟控制位</b> 0: SPI0 时钟开启 1: SPI0 时钟关闭	1h0
RB_SLP_CLK_TMR3	3	RWA	<b>TIM3 时钟控制位</b> 0: TIM3 时钟开启 1: TIM3 时钟关闭	1h0
RB_SLP_CLK_TMR2	2	RWA	<b>TIM2 时钟控制位</b> 0: TIM2 时钟开启 1: TIM2 时钟关闭	1h0
RB_SLP_CLK_TMR1	1	RWA	<b>TIM1 时钟控制位</b>	1h0



			0: TIM1 时钟开启 1: TIM1 时钟关闭	
RB_SLP_CLK_TMRO	0	RWA	<b>TIM0 时钟控制位</b> 0: TIM0 时钟开启 1: TIM0 时钟关闭	1h0

时钟控制寄存器 1 (R8\_SLP\_CLK\_OFF1):

名称	位	访问	描述	复位值
RB_SLP_CLK_ETH	7	RWA	<b>以太网时钟控制位</b> 0: 以太网时钟开启 1: 以太网时钟关闭	1h0
Reserved	6	RWA	<b>保留</b>	1h0
RB_SLP_CLK_GPIO	5	RWA	<b>GPIO 时钟控制位</b> 0: GPIO 时钟开启 1: GPIO 时钟关闭, 以下寄存器写保护: 主动并口模块所有寄存器/外部总线, 直跳中断入口 R32_INT_VEC_*, GPIO 中断模式 R32/R8_INT_MODE_*, GPIO 中断极性 R32/R8_INT_POLAR_*, 端口上拉/下拉 R32/R8_P*_PU/PD, 端口驱动/施密特 R32/R8_P*_DRV/SMT	1h0
RB_SLP_CLK_ADC	4	RWA	<b>ADC 时钟控制位</b> 0: ADC 时钟开启 1: ADC 时钟关闭	1h0
Reserved	[3: 0]	-	<b>保留</b>	-

休眠控制寄存器 (R8\_SLP\_WAKE\_CTRL):

名称	位	访问	描述	复位值
RB_SLP_ETH_PWR_DN	7	RWA	<b>以太网模块电源控制位</b> 0: 以太网模块上电 1: 以太网模块下电	1h1
RB_SLP_WOL_WAKE	6	RWA	<b>以太网 WOL 唤醒使能/禁止位</b> 0: 禁止 1: 使能	1h0
Reserved	5	RWA	<b>保留</b>	1h0
Reserved	4	RWA	<b>保留</b>	1h0
RB_SLP_PD_WAKE	3	RWA	<b>GPIO PD 端口唤醒使能/禁止位</b> 0: 禁止 1: 使能	1h0
Reserved	2	-	<b>保留</b>	-
RB_SLP_PB_WAKE	1	RWA	<b>GPIO PB 端口唤醒使能/禁止位</b> 0: 禁止 1: 使能	1h0
RB_SLP_PA_WAKE	0	RWA	<b>GPIO PA 端口唤醒使能/禁止位</b> 0: 禁止 1: 使能	1h0

PLL 休眠控制寄存器 (R8\_SLP\_CTRL\_PLL):

名称	位	访问	描述	复位值
R8_SLP_CTRL_PLL	[7: 0]	WA	<b>PLL 休眠控制域</b> 该寄存器写入 0x6A, 将使能 CPU 进入待机状态, 此时 PLL 仍然在工作, CPU 可以被已使能的各种中断快速唤醒; 该寄存器写入 0x95, 将使能 PLL 进入全局休眠状态(包括 CPU), 仅支持部分 GPIO 或者 ETH-WOL 唤醒	8h0
RB_SLP_PLL_STATUS	7	RO	<b>当前 PLL 休眠状态</b> 0: PLL 未休眠 1: PLL 处于休眠状态	1h0
RB_SLP_CPU_STATUS	6	RO	<b>当前 CPU 待机状态</b> 0: CPU 未休眠 1: CPU 处于休眠状态	1h0
Reserved	[5: 1]	-	保留	-
RB_SLP_WOL_STATUS	0	RO	<b>当前以太网 WOL 远程唤醒信号状态</b> 0: 以太网模块未请求唤醒(WOL=0) 1: 以太网模块请求唤醒(WOL=1)	1h0

以太网控制寄存器(R8\_MSC\_CTRL\_ETH):

名称	位	访问	描述	复位值
RB_MSC_ETH_RST	7	RW	<b>以太网控制器复位位</b> 0: 无动作 1: 以太网控制器复位 在初始化以太网控制器前, 必须先执行以太网复位动作。 主要步骤为: (1)、该位置 1, 复位以太网模块; (2)、置 RB_SLP_ETH_PWR_DN 为 0, 给以太网模块供电; (3)、延时几十毫秒; (4)、该位清 0, 结束以太网模块复位; (5)、延时几十毫秒; (6)、以太网初始化及其他操作;	1h0
Reserved	[6: 3]	-	保留	-
RB_MSC_ETH_LED	2		<b>以太网控制器连接指示灯使能/禁止位</b> 0: 禁止 1: 使能 ELINK#引脚输出连接状态	1h0
RB_MSC_ETH_TX	1		<b>以太网控制器发送指示灯使能/禁止位</b> 0: 禁止 1: 使能 ELED#引脚输出发送状态	1h0
RB_MSC_ETH_RX	0		<b>以太网控制器接收指示灯使能/禁止位</b> 0: 禁止 1: 使能 ELED#引脚输出接收状态	1h0

### 5.3 全局休眠后的唤醒

全局休眠状态是最低功耗状态，此时 PLL 停止工作，CH561 内部时钟暂停，CPU 不再工作，也不响应任何中断，但是如果唤醒后 CPU 开始工作，发现唤醒事件同时也属于中断事件（例如某 GPIO 唤醒和 GPIO 中断），那么会当作中断处理。

为降低功耗，在进入全局休眠状态前，休眠期间不会用到的物理 Phy 模块（例如 ADC/ETH）应该提前关闭。另外，各个 GPIO 引脚不能处于悬浮状态，要么是输出状态，要么是输入状态但是外部提供高电平或者低电平，如果是输入状态且外部没有提供高或低电平，那么应该启用该引脚的内部下拉电阻。不推荐使用内部上拉电阻的原因是，对于支持 5V 电压输入的 FT 引脚，启用上拉电阻后仍然会有少量静态电流，除非由外部电路提供上拉电阻拉到 VCC 电压。

在全局休眠状态下，CH561 仅支持部分 GPIO 或者 WOL 唤醒，参考 R8\_SLP\_WAKE\_CTRL，分别由 RB\_SLP\_PA\_WAKE、RB\_SLP\_PB\_WAKE、RB\_SLP\_PD\_WAKE、RB\_SLP\_WOL\_WAKE 控制，其中前 3 组为 GPIO 唤醒控制。

GPIO 唤醒事件源与 GPIO 中断事件源相同（除了不需要 R32\_INT\_MDE\_Px），当 R32\_INT\_POLAR\_Px 对应的位为 0 时，GPIO 引脚出现低电平时唤醒，当 R32\_INT\_POLAR\_Px 对应的位为 1 时，GPIO 引脚出现高电平时唤醒。以设置允许 PA[8] 低电平唤醒 CH561 为例，应该设置 R32\_INT\_POLAR\_PA[8]=0，设置 R32\_INT\_ENABLE\_PA[8]=1，设置 RB\_SLP\_PA\_WAKE=1，然后设置其它相关寄存器进入休眠状态。如果希望 PA[8] 同时作为中断事件，那么还需要设置 RB\_IE\_IRQ\_GLOB 或 RB\_IE\_FIQ\_GLOB，否则清 0。

## 第 6 章 通用和复用功能 I/O

### 6.1 GPIO 简介

CH561 最多拥有 27 个输入输出引脚，部分引脚具有复用功能。每个 GPIO 端口有一个 32 位方向配置寄存器( R32\_Px\_DIR )，一个 32 位数据输入寄存器( R32\_Px\_IN )，一个 32 位数据输出寄存器( R32\_Px\_OUT )，一个 32 位清除寄存( R32\_Px\_CLR )，一个 32 位上拉配置寄存器( R32\_Px\_PU )，一个 32 位开漏输出和输入下拉配置寄存器(R32\_Px\_PD )，一个 32 位 I/O 驱动能力配置寄存器( R32\_Px\_DRV )和一个 32 位施密特触发使能配置寄存器(R32\_Px\_SMT )。

CH561 所有 I/O 口分为 PA、PB 和 2 个端口，每个端口分别为：

Port A(PA)：7 位输入输出端口；

Port B(PB)：20 位输入输出端口；

每个 I/O 端口位可以自由编程，但是 I/O 端口寄存器必须按 8 位、16 位或者 32 位字访问。如果引脚没有被配置成复用功能，则可以做为通用 I/O 口使用。

### 6.2 GPIO 寄存器描述

GPIO 相关寄存器物理起始地址为：0x0040 0080

GPIO 相关寄存器列表(地址范围：0x00---0x3C)

表 6-1 GPIO 相关寄存器列表

名称	偏移地址	描述	复位值
R32_PA_DIR	0x00	PA 端口方向设置寄存器	32h0000 0000
R32_PA_PIN	0x04	PA 端口数据输入寄存器	32hxxxx xxxx
R32_PA_OUT	0x08	PA 端口数据输出寄存器	32h0000 0000
R32_PA_CLR	0x0C	PA 端口输出清 0 寄存器	32h0000 0000
R32_PA_PU	0x10	PA 端口上拉配置寄存器	32h0000 0000
R32_PA_PD	0x14	PA 端口开漏输出和输入下拉配置寄存器	32h0000 0000
R32_PA_DRV	0x18	PA 端口驱动能力配置寄存器	32h0000 0000
R32_PA_SMT	0x1C	PA 端口施密特触发器使能配置寄存器	32h0000 0000
R32_PB_DIR	0x20	PB 端口方向设置寄存器	32h0000 0000
R32_PB_PIN	0x24	PB 端口数据输入寄存器	32hxxxxx xxxx
R32_PB_OUT	0x28	PB 端口数据输出寄存器	32h0000 0000
R32_PB_CLR	0x2C	PB 端口输出清 0 寄存器	32h0000 0000
R32_PB_PU	0x30	PB 端口上拉配置寄存器	32h0000 0000
R32_PB_PD	0x34	PB 端口开漏输出和输入下拉配置寄存器	32h0000 0000
R32_PB_DRV	0x38	PB 端口驱动能力配置寄存器	32h0000 0000
R32_PB_SMT	0x3C	PB 端口施密特触发器使能配置寄存器	32h0000 0000

GPIO 中断相关寄存器物理起始地址为：0x0040 0050

GPIO 中断相关寄存器列表(地址范围：0x00---0x1C)

表 6-2 GPIO 中断相关寄存器列表

名称	偏移地址	描述	复位值
R32_INT_STATUS_PA	0x00	PA 端口中断状态寄存器	32h0000 0000

R32_INT_ENABLE_PA	0x04	PA 端口中断使能寄存器	32h0000 0000
R32_INT_MDE_PA	0x08	PA 端口中断模式寄存器	32h0000 0000
R32_INT_POLAR_PA	0x0C	PA 端口中断极性寄存器	32h0000 0000
R32_INT_STATUS_PB	0x10	PB 端口中断状态寄存器	32h0000 0000
R32_INT_ENABLE_PB	0x14	PB 端口中断使能寄存器	32h0000 0000
R32_INT_MDE_PB	0x18	PB 端口中断模式寄存器	32h0000 0000
R32_INT_POLAR_PB	0x1C	PB 端口中断极性寄存器	32h0000 0000

PA 端口方向设置寄存器(R32\_PA\_DIR)：

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PA[19: 17]	[19: 17]	RW	<b>引脚当前输入输出方向控制域</b> 0: 该引脚方向为输入模式 1: 该引脚方向为输出模式	3h0
Reserved	[16: 13]	-	保留	-
PA[12: 11]	[12: 11]	RW	<b>引脚当前输入输出方向控制域</b> 0: 该引脚方向为输入模式 1: 该引脚方向为输出模式	2h0
Reserved	[10: 9]	-	保留	-
PA[8: 7]	[8: 7]	RW	<b>引脚当前输入输出方向控制域</b> 0: 该引脚方向为输入模式 1: 该引脚方向为输出模式	2h0
Reserved	[6: 0]	-	保留	-

PA 端口输入数据寄存器(R32\_PA\_PIN)：

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PA[19: 17]	[19: 17]	RW	<b>引脚当前电平状态域</b> 0: 该引脚输入低电平 1: 该引脚输入高电平 仅在方向寄存器(R32_PA_DIR)对应位为0时, 该位值有效	3h0
Reserved	[16: 13]	-	保留	-
PA[12: 11]	[12: 11]	RW	<b>引脚当前电平状态域</b> 0: 该引脚输入低电平 1: 该引脚输入高电平 仅在方向寄存器(R32_PA_DIR)对应位为0时, 该位值有效	2h0
Reserved	[10: 9]	-	保留	-
PA[8: 7]	[8: 7]	RW	<b>引脚当前电平状态域</b> 0: 该引脚输入低电平 1: 该引脚输入高电平 仅在方向寄存器(R32_PA_DIR)对应位为0时, 该位值有效	2h0
Reserved	[6: 0]	-	保留	-

PA 端口输出数据寄存器 (R32\_PA\_OUT)：

名称	位	访问	描述	复位值
----	---	----	----	-----

Reserved	[31: 20]	-	保留	-
PA[19: 17]	[19: 17]	RW	<b>引脚输出高低电平控制域</b> 0: 该引脚输出低电平 1: 该引脚输出高电平 仅在方向寄存器 (R32_PA_DIR) 对应位为 1 时, 才有效	3h0
Reserved	[16: 13]	-	保留	
PA[12: 11]	[12: 11]	RW	<b>引脚输出高低电平控制域</b> 0: 该引脚输出低电平 1: 该引脚输出高电平 仅在方向寄存器 (R32_PA_DIR) 对应位为 1 时, 才有效	2h0
Reserved	[10: 9]	-	保留	
PA[8: 7]	[8: 7]	RW	<b>引脚输出高低电平控制域</b> 0: 该引脚输出低电平 1: 该引脚输出高电平 仅在方向寄存器 (R32_PA_DIR) 对应位为 1 时, 才有效	2h0
Reserved	[6: 0]	-	保留	-

PA 端口输出清 0 寄存器 (R32\_PA\_CLR):

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PA[19: 17]	[19: 17]	RW	保持/清除数据输出寄存器 (R32_PA_OUT) 对应位的值 0: 该引脚输出不改变 1: 该引脚输出低电平	3h0
Reserved	[16: 13]	-	保留	
PA[12: 11]	[12: 11]	RW	保持/清除数据输出寄存器 (R32_PA_OUT) 对应位的值 0: 该引脚输出不改变 1: 该引脚输出低电平	2h0
Reserved	[10: 9]	-	保留	
PA[8: 7]	[8: 7]	RW	保持/清除数据输出寄存器 (R32_PA_OUT) 对应位的值 0: 该引脚输出不改变 1: 该引脚输出低电平	2h0
Reserved	[6: 0]	-	保留	-

PA 端口上拉配置寄存器 (R32\_PA\_PU):

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PA[19: 17]	[19: 17]	RW	<b>引脚上拉功能启用/关闭控制域</b> 0: 关闭该引脚上拉功能 1: 启用该引脚上拉功能	3h0
Reserved	[16: 13]	-	保留	
PA[12: 11]	[12: 11]	RW	<b>引脚上拉功能启用/关闭控制域</b> 0: 关闭该引脚上拉功能	2h0

			1: 启用该引脚上拉功能	
Reserved	[10: 9]	-	保留	
PA[8: 7]	[8: 7]	RW	<b>引脚上拉功能启用/关闭控制域</b> 0: 关闭该引脚上拉功能 1: 启用该引脚上拉功能	2h0
Reserved	[6: 0]	-	保留	-

PA 端口开漏输出和输入下拉配置寄存器 (R32\_PA\_PD):

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PA[19: 17]	[19: 17]	RW	方向寄存器 (R32_PA_DIR) 对应位配置为 1 (即输出模式) 时: 0: 关闭该引脚开漏输出功能 1: 启用该引脚开漏输出功能 方向寄存器 (R32_PA_DIR) 对应位配置为 0 (即输入模式) 时: 0: 关闭该引脚下拉功能 1: 启用该引脚下拉功能, 如果上拉配置寄存器 (R32_PA_PU) 对应位配置也为 1 则启用输入状态弱保持功能	3h0
Reserved	[16: 13]	-	保留	
PA[12: 11]	[12: 11]	RW	方向寄存器 (R32_PA_DIR) 对应位配置为 1 (即输出模式) 时: 0: 关闭该引脚开漏输出功能 1: 启用该引脚开漏输出功能 方向寄存器 (R32_PA_DIR) 对应位配置为 0 (即输入模式) 时: 0: 关闭该引脚下拉功能 1: 启用该引脚下拉功能, 如果上拉配置寄存器 (R32_PA_PU) 对应位配置也为 1 则启用输入状态弱保持功能	2h0
Reserved	[10: 9]	-	保留	
PA[8: 7]	[8: 7]	RW	方向寄存器 (R32_PA_DIR) 对应位配置为 1 (即输出模式) 时: 0: 关闭该引脚开漏输出功能 1: 启用该引脚开漏输出功能 方向寄存器 (R32_PA_DIR) 对应位配置为 0 (即输入模式) 时: 0: 关闭该引脚下拉功能 1: 启用该引脚下拉功能, 如果上拉配置寄存器 (R32_PA_PU) 对应位配置也为 1 则启用输入状态弱保持功能	2h0
Reserved	[6: 0]	-	保留	-

PA 端口驱动能力配置寄存器 (R32\_PA\_DRV):

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PA[19: 17]	[19: 17]	RW	<b>引脚输出驱动能力控制域</b>	3h0

			0: 驱动电流最大为 4mA 1: 驱动电流最大为 16mA	
Reserved	[16: 13]	-	保留	
PA12	12	RW	<b>引脚输出驱动能力控制域</b> 0: 驱动电流最大为 4mA 1: 驱动电流最大为 16mA	1h0
Reserved	[11: 0]	-	保留	-

PA 端口施密特触发器使能配置寄存器 (R32\_PA\_SMT) :

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PA[19: 17]	[19: 17]	RW	<b>引脚施密特触发器功能启用/关闭控制域</b> 0: 关闭该引脚施密特触发器输入功能, 关闭该引脚低斜率输出功能 1: 启用该引脚施密特触发器输入功能, 启用该引脚低斜率输出功能	3h0
Reserved	[16: 13]	-	保留	-
PA[11: 12]	[12: 11]	RW	<b>引脚施密特触发器功能启用/关闭控制域</b> 0: 关闭该引脚施密特触发器输入功能, 关闭该引脚低斜率输出功能 1: 启用该引脚施密特触发器输入功能, 启用该引脚低斜率输出功能	2h0
PA[9: 10]	[10: 9]	-	保留	-
PA8	8	RW	<b>引脚施密特触发器功能启用/关闭控制域</b> 0: 关闭该引脚施密特触发器输入功能, 关闭该引脚低斜率输出功能 1: 启用该引脚施密特触发器输入功能, 启用该引脚低斜率输出功能	1h0
Reserved	[7: 0]	-	保留	-

PB 端口方向设置寄存器 (R32\_PB\_DIR) :

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PB[19: 0]	[19: 0]	RW	<b>引脚当前输入输出方向控制域</b> 0: 该引脚方向为输入模式 1: 该引脚方向为输出模式	20h0

PB 端口输入数据寄存器 (R32\_PB\_PIN) :

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PB[19: 0]	[19: 0]	RO	<b>引脚当前电平状态域</b> 0: 该引脚输入低电平 1: 该引脚输入高电平 仅在方向寄存器 (R32_PB_DIR) 对应位为 0 时, 该位值有效	20hx



PB 端口输出数据寄存器 (R32\_PB\_OUT) :

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PB[19: 0]	[21: 0]	RW	引脚输出高低电平控制域 0: 该引脚输出低电平 1: 该引脚输出高电平 仅在方向寄存器 (R32_PB_DIR) 对应位为 1 时, 才有效	22h0

PB 端口输出清 0 寄存器 (R32\_PB\_CLR) :

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PB[19: 0]	[19: 0]	RZ	保持/清除数据输出寄存器 (R32_PB_OUT) 对应位的值 0: 该引脚输出不改变 1: 该引脚输出低电平	20h0

PB 端口上拉配置寄存器 (R32\_PB\_PU) :

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PB[19: 0]	[19: 0]	RW	引脚上拉功能启用/关闭控制域 0: 关闭该引脚上拉功能 1: 启用该引脚上拉功能	22h0

PB 端口开漏输出和输入下拉配置寄存器 (R32\_PB\_PD) :

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PB[19: 0]	[19: 0]	RW	方向寄存器 (R32_PB_DIR) 对应位配置为 1 (即输出模式) 时: 0: 关闭该引脚开漏输出功能 1: 启用该引脚开漏输出功能 方向寄存器 (R32_PB_DIR) 对应位配置为 0 (即输入模式) 时: 0: 关闭该引脚下拉功能 1: 启用该引脚下拉功能, 如果上拉配置寄存器 (R32_PB_PU) 对应位配置也为 1 则启用输入状态弱保持功能	20h0

PB 端口驱动能力配置寄存器 (R32\_PB\_DRV) :

名称	位	访问	描述	复位值
Reserved	[31: 16]	-	保留	-
PB[15: 8]	[15: 8]	RW	引脚输出驱动能力控制域 0: 驱动电流最大为 4mA 1: 驱动电流最大为 16mA	8h0
Reserved	[7: 0]	-	保留	-

PB 端口施密特触发器使能配置寄存器 (R32\_PB\_SMT) :

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PB[19: 0]	[19: 0]	RW	引脚施密特触发器功能启用/关闭控制域 0: 关闭该引脚施密特触发器输入功能, 关闭该引脚低斜率输出功能 1: 启用该引脚施密特触发器输入功能, 启用该引脚低斜率输出功能	20h0

PA 端口中断状态寄存器(R32\_INT\_STATUS\_PA):

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PA[17: 19]	[19: 17]	RW	PA 端口中断状态域 0: 该引脚无中断产生 1: 该引脚有中断产生	3h0
Reserved	[16: 13]	-	保留	-
PA[11: 12]	[12: 11]	RW	PA 端口中断状态域 0: 该引脚无中断产生 1: 该引脚有中断产生	2h0
Reserved	[10: 9]	-	保留	-
PA8	8	RW	PA 端口中断状态域 0: 该引脚无中断产生 1: 该引脚有中断产生	1h0
Reserved	[7: 0]	-	保留	-

PA 端口中断使能寄存器(R32\_INT\_ENABLE\_PA):

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PA[17: 19]	[19: 17]	RW	PA 端口中断使能/禁止控制域 0: 禁止该引脚产生中断 1: 使能该引脚产生中断	3h0
Reserved	[16: 13]	-	保留	-
PA[11: 12]	[12: 11]	RW	PA 端口中断使能/禁止控制域 0: 禁止该引脚产生中断 1: 使能该引脚产生中断	2h0
Reserved	[10: 9]	-	保留	-
PA8	8	RW	PA 端口中断使能/禁止控制域 0: 禁止该引脚产生中断 1: 使能该引脚产生中断	1h0
Reserved	[7: 0]	-	保留	-

PA 端口中断模式寄存器(R32\_INT\_MODE\_PA):

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PA[17: 19]	[19: 17]	RW	PA 端口中断模式控制域 0: 电平中断 1: 边沿中断	3h0
Reserved	[16: 13]	-	保留	-

PA[11: 12]	[12: 11]	RW	PA 端口中断模式控制域 0: 电平中断 1: 边沿中断	2h0
Reserved	[10: 9]	-	保留	-
PA8	8	RW	PA 端口中断模式控制域 0: 电平中断 1: 边沿中断	1h0
Reserved	[7: 0]	-	保留	-

PA 端口中断极性寄存器(R32\_INT\_POLAR\_PA):

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PA[17: 19]	[19: 17]	RW	PA 端口中断极性控制域 0: 低电平/下降沿产生中断 1: 高电平/上升沿产生中断	3h0
Reserved	[16: 13]	-	保留	-
PA[11: 12]	[12: 11]	RW	PA 端口中断极性控制域 0: 低电平/下降沿产生中断 1: 高电平/上升沿产生中断	2h0
Reserved	[10: 9]	-	保留	-
PA8	8	RW	PA 端口中断极性控制域 0: 低电平/下降沿产生中断 1: 高电平/上升沿产生中断	1h0
Reserved	[7: 0]	-	保留	-

PB 端口中断状态寄存器(R32\_INT\_STATUS\_PB):

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PB[19: 0]	[19: 0]	RW	PB 端口中断状态域 0: 该引脚无中断产生 1: 该引脚有中断产生	20h0

PB 端口中断使能寄存器(R32\_INT\_ENABLE\_PB):

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PB[19: 0]	[19: 0]	RW	PB 端口中断使能/禁止控制域 0: 禁止该引脚产生中断 1: 使能该引脚产生中断	20h0

PB 端口中断模式寄存器(R32\_INT\_MDE\_PB):

名称	位	访问	描述	复位值
Reserved	[31: 20]	-	保留	-
PB[19: 0]	[19: 0]	RW	PB 端口中断模式控制域 0: 电平中断 1: 边沿中断	20h0

PB 端口中断极性寄存器(R32\_INT\_POLAR\_PB):

名称	位	访问	描述	复位值
----	---	----	----	-----

Reserved	[31: 20]	-	保留	-
PB[19: 0]	[19: 0]	RW	PB 端口中断极性控制域 0: 低电平/下降沿产生中断 1: 高电平/上升沿产生中断	20h0

### 6.3 GPIO 复用功能

CH561 部分 I/O 引脚具有复用功能，上电后默认所有 I/O 引脚均为通用 I/O 功能，使能不同功能模块后，相应引脚被配置成各自功能模块对应的功能引脚。以下各表为各个功能模块所使用的 I/O 引脚。

表 6-2 串行外设接口(SPI0)

引脚	GPIO	功能描述
SCS	PB12	SPI0 从机片选输入引脚
SCK0	PB13	SPI0 串行时钟引脚，主机输出/从机输入
MDSI	PB14	SPI0 串行数据引脚，主机输出/从机输入
MISO	PB15	SPI0 串行数据引脚，主机输入/从机输出

表 6-3 串行外设接口(SPI1)

引脚	GPIO	功能描述
SCK1	PB17	SPI1 串行时钟输出引脚
SDO	PB18	SPI1 串行数据输出引脚
SDI	PB19	SPI1 串行数据输入引脚

表 6-4 通用异步收发器(UART0)

引脚	GPIO	功能描述
RXD0	PB8	UART0 接收器输入引脚
TXD0	PB9	UART0 发送器输出引脚
DTR	PA7	UART0 的 MODEM 信号，数据终端就绪
RTS	PA8	UART0 的 MODEM 信号，请求发送
CTS	PA12	UART0 的 MODEM 信号，清除发送

表 6-5 通用异步收发器(UART1)

引脚	GPIO	功能描述
RXD1	PB10	UART1 接收器输入引脚
TXD1	PB11	UART1 发送器输出引脚

表 6-8 ISP 下载(ISP)

引脚	GPIO	功能描述
SCS	PB12	ISP 下载片选输入引脚
SCK	PB13	ISP 下载时钟输入引脚
MDSI	PB14	ISP 下载数据输入引脚
MISO	PB15	ISP 下载数据输出引脚
RST	RST#	ISP 下载复位输入引脚

## 第 7 章 串行外设接口 SPI

### 7.1 SPI 简介

SPI 是一种全双工串行接口，可处理多个连接到指定总线上的主机和从机。在数据过程中，总线上只能有一个主机和一个从机通信。在数据传输中，主机总是会发送一帧 8 位数据，而从机也总会向主机发送一帧字节数据。通常 SPI 接口由 4 个引脚组成：SPI 片选引脚(SCS)、SPI 时钟引脚(SCK)、SPI 串行数据引脚(主机输入/从机输出引脚 MISO)和 SPI 串行数据引脚(主机输出/从机输入引脚 MOSI)。

CH561 芯片拥有 2 个 SPI 接口，其各自特性分别如下所示：

SPI0 特性：

- (1)、支持主机模式和设备模式；
- (2)、兼容串行外设接口(SPI)规范；
- (3)、支持模式 0 和模式 3 数据传输方式；
- (4)、8 位数据传输方式；
- (5)、时钟频率最高可达 50MHz；
- (6)、32 级 FIFO；
- (7)、设备模式支持首字节为命令模式和数据流模式；
- (8)、支持 DMA 数据传输；

SPI1 特性：

- (1)、只支持主机模式；
- (2)、支持模式 0 和模式 3 数据传输方式；
- (3)、8 位数据传输方式；
- (4)、时钟频率最高可达 50MHz；
- (5)、16 级 FIFO；

### 7.2 SPI 寄存器描述

SPI0 相关寄存器物理起始地址为：0x0040 C000

SPI0 相关寄存器列表(地址范围：0x00---0x1C)

表 7-1 SPI0 相关寄存器列表

名称	偏移地址	描述	复位值
R32_SPI0_CONTROL	0x00	SPI0 控制寄存器	32h1000 0002
R8_SPI0_CTRL_MDD	0x00	SPI0 模式配置寄存器	8h02
R8_SPI0_CTRL_DMA	0x01	SPI0 DMA 控制寄存器	8h00
R8_SPI0_INTER_EN	0x02	SPI0 中断使能寄存器	8h00
R8_SPI0_CLOCK_DIV	0x03	SPI0 主机模式时钟分频寄存器	8h10
R8_SPI0_SLAVE_PRE		SPI0 设备模式预置数据寄存器	
R32_SPI0_STATUS	0x04	SPI0 状态寄存器	32h0000 0000
R8_SPI0_BUFFER	0x04	SPI0 数据缓冲区	8h00
R8_SPI0_RUN_FLAG	0x05	SPI0 工作标志寄存器	8h00
R8_SPI0_INT_FLAG	0x06	SPI0 中断标志寄存器	8h00
R8_SPI0_FIFO_COUNT	0x07	SPI0 发送/接收 FIFO 计数寄存器	8h00
R16_SPI0_TOTAL_CNT	0x0C	SPI0 发送/接收数据长度寄存器	16h0000
R32_SPI0_FIFO	0x10	SPI0 FIFO 寄存器	32h0000 0000
R8_SPI0_FIFO	0x10	SPI0 FIFO 寄存器	8h00
R8_SPI0_FIFO_COUNT1	0x13	SPI0 发送/接收 FIFO 计数寄存器	8h00

R16_SPI0_DMA_NOW	0x14	SPI0 DMA 缓冲区当前地址	16h0000
R16_SPI0_DMA_BEG	0x18	SPI0 DMA 缓冲区开始地址	16h0000
R16_SPI0_DMA_END	0x1C	SPI0 DMA 缓冲区结束地址	16h0000

SPI1 相关寄存器物理起始地址为：0X0040 C800

SPI1 相关寄存器列表(地址范围：0x00---0x13)

表 7-2 SPI1 相关寄存器列表

名称	偏移地址	描述	复位值
R32_SPI1_CONTROL	0x00	SPI1 控制寄存器	32h1000 0002
R8_SPI1_CTRL_MD	0x00	SPI1 模式配置寄存器	8h02
R8_SPI1_CTRL_DMA	0x01	SPI1 DMA 控制寄存器	8h00
R8_SPI1_INTER_EN	0x02	SPI1 中断使能寄存器	8h00
R8_SPI1_CLOCK_DIV	0x03	SPI1 主机模式时钟分频寄存器	8h02
R32_SPI1_STATUS	0x04	SPI1 状态寄存器	32h0000 0000
R8_SPI1_BUFFER	0x04	SPI1 数据缓冲区	8h00
R8_SPI1_RUN_FLAG	0x05	SPI1 工作标志寄存器	8h00
R8_SPI1_INT_FLAG	0x06	SPI1 中断标志寄存器	8h00
R8_SPI1_FIFO_COUNT	0x07	SPI1 发送/接收 FIFO 计数寄存器	8h00
R16_SPI1_TOTAL_CNT	0x0C	SPI1 发送/接收数据长度寄存器	16h0000
R32_SPI1_FIFO	0x10	SPI1 FIFO 寄存器	32h0000 0000
R8_SPI1_FIFO	0x10	SPI1 FIFO 寄存器	8h00
R8_SPI1_FIFO_COUNT1	0x13	SPI1 发送/接收 FIFO 计数寄存器	8h00

SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MD)：

名称	位	访问	描述	复位值
RB_SPI_MISO_OE	7	RW	MISO 引脚输出使能/禁止位, 仅 SPI0 0: MISO 引脚输出禁止 1: MISO 引脚输出使能 2 线或 3 线 SPI 模式配置位, 仅 SPI1 0: 3 线 SPI 模式(SCK1、SD0、SDI) 1: 2 线 SPI 模式(SCK1、SDX1)	1h0
RB_SPI_MOSI_OE	6	RW	MOSI/SDO 引脚输出使能/禁止位 0: MOSI 引脚/SDO 引脚输出禁止 1: MOSI 引脚/SDO 引脚输出使能	1h0
RB_SPI_SCK_OE	5	RW	SCK 引脚输出使能/禁止位 0: SCK 引脚输出禁止 1: SCK 引脚输出使能	1h0
RB_SPI_FIFO_DIR	4	RW	FIFO 方向设置位 0: 输出模式(主机模式写数据) 1: 输入模式(主机模式读数据)	1h0
RB_SPI_SLV_CMD_MD	3	RW	设备模式首字节配置位, 仅 SPI0 0: 数据流模式 1: 首字节命令模式 在首字节命令模式下, 当接收到 SPI 片选有效后的首字节数据后, 将视为命令码, 且中断标志寄存器中的 RB_SPI_IF_FST_BYTE 位将置 1, 该位仅在设备模式下有效	1h0

RB_SPI_MST_SCK_MOD	3	RW	<b>主机时钟模式配置位</b> 0: 模式 0 (空闲时 SCK 为低电平) 1: 模式 3 (空闲时 SCK 为高电平) 该位仅在主机模式下有效	1h0
RB_SPI_2WIRE_MOD	2	RW	<b>2 线或 3 线 SPI 模式配置位, 仅 SPI0</b> 0: 3 线 SPI 模式 (SCK0、MDSI、MISO) 1: 2 线 SPI 模式 (SCK0、SDX0)	1h0
RB_SPI_ALL_CLEAR	1	RW	<b>FIFO 寄存器及计数寄存器清 0 位</b> 0: 不清除 1: 强制清除	1h0
RB_SPI_MODE_SLAVE	0	RW	<b>SPI 主从模式选择位, 仅 SPI0</b> 0: 主机模式 1: 设备模式 注: SPI1 不支持设备模式	1h0

## SPI DMA 控制寄存器 (R8\_SPIx\_CTRL\_DMA):

名称	位	访问	描述	复位值
Reserved	[7: 3]	-	保留	-
RB_SPI_DMA_LOOP	2	RW	<b>DMA 地址循环模式使能/禁止位</b> 0: 禁止 DMA 地址循环模式功能 1: 使能 DMA 地址循环模式功能 如果使能 DMA 地址循环模式功能, 当 DMA 地址增加到设置的末尾地址时, 自动循环指向设置的首地址 注: SPI1 不支持	1h0
RB_SPI_DMA_BURST	1	RW	<b>DMA 块操作使能/禁止位</b> 0: 禁止 DMA 块操作功能 1: 使能 DMA 块操作功能 如果使能 DMA 块操作功能, 则该 DMA 更少占用系统总线 注: SPI1 不支持	1h0
RB_SPI_DMA_ENABLE	0	RW	<b>DMA 功能使能/禁止位</b> 0: 禁止 DMA 功能 1: 使能 DMA 功能 注: SPI1 不支持	1h0

如果使能 DMA 地址循环模式功能, 当 DMA 地址增加到设置的末尾地址时, 自动循环指向设置的首地址, 而不需要重新设置 DMA 起始地址寄存器 (R16\_TIM\_DMA\_BEG) 和 DMA 结束地址寄存器 (R16\_TIM\_DMA\_END)。

## SPI 中断使能寄存器 (R8\_SPIx\_INTER\_EN):

名称	位	访问	描述	复位值
RB_SPI_IE_FST_BYTE	7	RW	<b>第一个字节接收中断使能/禁止位</b> 0: 禁止接收到第一个字节产生中断 1: 使能接收到第一个字节产生中断 开启此功能需要设置 SPI 为设备模式, 同时需要将 RB_SPI_SLV_CMD_MOD 位置 1 注: SPI1 不支持	1h0
Reserved	6	-	保留	-

RB_SPI_IE_DMA_ERR	5	RW	DMA 传输错误中断使能/禁止位 0: 禁止产生 DMA 传输错误中断 1: 使能产生 DMA 传输错误中断 注: SPI1 不支持	1h0
RB_SPI_IE_FIFO_OV	4	RW	FIFO 溢出中断使能/禁止位 0: 禁止产生 FIFO 溢出中断 1: 使能产生 FIFO 溢出中断	1h0
RB_SPI_IE_DMA_END	3	RW	DMA 结束中断使能/禁止位 0: 禁止产生 DMA 结束中断 1: 使能产生 DMA 结束中断 注: SPI1 不支持	1h0
RB_SPI_IE_FIFO_HF	2	RW	FIFO 过半中断使能/禁止位 0: 禁止产生 FIFO 过半中断 1: 使能产生 FIFO 过半中断	1h0
RB_SPI_IE_BYTE_END	1	RW	SPI 单字节传输完成中断使能/禁止位 0: 禁止产生 SPI 单字节传输完成中断 1: 使能产生 SPI 单字节传输完成中断	1h0
RB_SPI_IE_CNT_END	0	RW	SPI 全部字节传输完成中断使能/禁止位 0: 禁止产生 SPI 全部字节传输完成中断 1: 使能产生 SPI 全部字节传输完成中断	1h0

SPI 主机模式时钟分频寄存器 (R8\_SPIx\_CLOCK\_DIV) :

名称	位	访问	描述	复位值
SPIx_CLOCK_DIV	[7: 0]	RW	主机模式分频系数, 最小值为 2 SPI 时钟频率=主频/分频系数	8h02

SPI 设备模式预置数据寄存器 (R8\_SPIx\_SLAVE\_PRE) :

名称	位	访问	描述	复位值
SPIx_SLAVE_PRESET	[7: 0]	RW	设备模式下预置的返回数据 用于接收首字节数据后的返回数据 注: SPI1 不支持	8h02

SPI 数据缓冲区 (R8\_SPIx\_BUFFER) :

名称	位	访问	描述	复位值
SPIx_BUFFER	[7: 0]	RW	SPI 数据发送和接收缓冲区	8h0

SPI 工作标志寄存器 (R8\_SPIx\_RUN\_FLAG) :

名称	位	访问	描述	复位值
RB_SPI_SLV_SELECT	7	RO	SPI 设备模式选择状态位 注: SPI1 不支持	1h0
RB_SPI_SLV_CS_LOAD	6	RO	SPI 设备模式片选后首次加载状态位 注: SPI1 不支持	1h0



RB_SPI_FIFO_READY	5	RO	<b>FIFO 准备完成标志位</b> 0: FIFO 未准备好 1: FIFO 准备完成	1h0
RB_SPI_SLV_CMD_ACT	4	RO	<b>SPI 设备模式下命令接收完成标志位, 即交换完首字节数据</b> 注: SPI1 不支持	1h0
Reserved	[3: 0]	-	保留	-

## SPI 中断标志寄存器 (R8\_SPIx\_INT\_FLAG) :

名称	位	访问	描述	复位值
RB_SPI_IF_FST_BYTE	7	RWI	<b>设备模式下首字节接收中断标志位</b> 0: 未接收到首字节 1: 接收到首字节 注: SPI1 不支持	1h0
RB_SPI_FREE	6	RO	<b>SPI 空闲状态位</b> 0: 当前 SPI 处于非空闲状态 1: 当前 SPI 处于空闲状态	1h0
RB_SPI_IF_DMA_ERR	5	RWI	<b>DMA 传输错误中断标志位</b> 0: DMA 传输过程中未产生错误 1: DMA 传输过程中产生错误 注: SPI1 不支持	1h0
RB_SPI_IF_FIFO_OV	4	RWI	<b>FIFO 溢出中断标志位</b> 0: FIFO 未溢出 1: FIFO 发生溢出	1h0
RB_SPI_IF_DMA_END	3	RWI	<b>DMA 结束中断标志位</b> 0: DMA 传输未结束 1: DMA 传输结束 注: SPI1 不支持	1h0
RB_SPI_IF_FIFO_HF	2	RWI	<b>FIFO 过半中断标志位</b> 0: 数据未达到 FIFO 缓冲区一半 1: 数据达到 FIFO 缓冲区一半	1h0
RB_SPI_IF_BYTE_END	1	RWI	<b>SPI 单字节传输完成中断标志位</b> 0: SPI 单字节传输未完成 1: SPI 单字节传输完成	1h0
RB_SPI_IF_CNT_END	0	RWI	<b>SPI 全部字节传输完成中断标志位</b> 0: SPI 全部字节传输未完成 1: SPI 全部字节传输完成	1h0

## SPI 发送/接收 FIFO 计数寄存器 (R8\_SPIx\_FIFO\_COUNT) :

名称	位	访问	描述	复位值
SPIx_FIFO_COUNT	[7: 0]	RO	SPI 发送或接收字节数	8h0

## SPI 发送/接收数据总长度寄存器 (R16\_SPIx\_TOTAL\_CNT) :

名称	位	访问	描述	复位值
Reserved	[15: 12]	-	保留	-
SPIx_TOTAL_CNT	[11: 0]	RW	<b>SPI 数据发送字节数</b> 在使用 DMA 时一次最多可以发送 4095 个字节	12h0

SPI FIFO 寄存器(R8\_SPIx\_FIFO)：

名称	位	访问	描述	复位值
SPIx_FIFO	[7: 0]	RO/W 0	SPI FIFO 寄存器 FIFO 大小为 32 个字节	8h0

SPI 发送/接收 FIFO 计数寄存器(R8\_SPIx\_FIFO\_COUNT1)：

名称	位	访问	描述	复位值
R8_SPIx_FIFO_COUNT1	[7: 0]		SPI FIFO 计数	8h0

寄存器 R8\_SPIx\_BUFFER 和寄存器 R8\_SPIx\_FIFO 均为 SPI 数据相关寄存器，主要区别在于：后者读取完一个字节数据之后，因为是从 FIFO 读取，所以长度寄存器(R16\_SPI\_TOTAL\_CNT)的值自动减 1，而前者读取完一个字节，长度寄存器值不变。

SPI DMA 缓冲区当前地址(R16\_SPIx\_DMA\_NOW)：

名称	位	访问	描述	复位值
R16_SPIx_DMA_NOW	[15: 0]	RO/W 0	DMA 缓冲区当前地址 通过查询此值可以判断 DMA 运行情况 注：SPI1 不支持	16h0

SPI0 DMA 缓冲区开始地址(R16\_SPI0\_DMA\_BEG)：

名称	位	访问	描述	复位值
R16_SPI0_DMA_BEG	[15: 0]	RW	DMA 缓冲区开始地址 指向 SPI 接收或者发送数据缓冲区开始地址 注：SPI1 不支持	16h0

SPI0 DMA 缓冲区结束地址(R16\_SPI0\_DMA\_END)：

名称	位	访问	描述	复位值
R16_SPI0_DMA_END	[15: 0]	RW	DMA 缓冲区结束地址 指向 SPI 接收或者发送数据缓冲区结束地址 注：SPI1 不支持	16h0

### 7.3 SPI 传输格式

SPI 仅支持模式 0 和模式 3 两种传输格式，可以通过设置 SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MDD)的 RB\_SPI\_MST\_SCK\_MDD 位来进行选择。

数据传输格式如下图所示：

模式 0：RB\_SPI\_MST\_SCK\_MDD = 0

模式0时序图

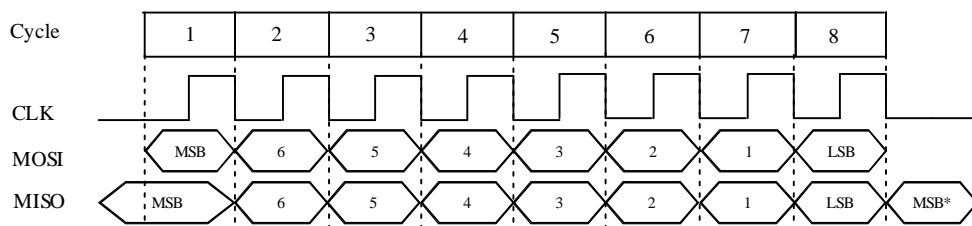


图 7-1 SPI 模式 0 传输格式

模式 3:  $RB\_SPI\_MST\_SCK\_MDD = 1$

模式3时序图

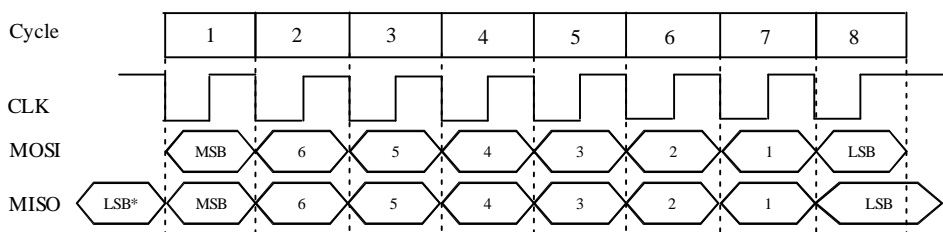


图 7-2 SPI 模式 3 传输格式

## 7.4 SPI 配置

### 7.4.1 SPI 主机模式配置

SPI 主机模式下，在 SCK 引脚产生串行时钟，片选引脚可以指定为任意 I/O 引脚。

配置步骤：

(1)、设置 SPI 主机模式时钟分频寄存器(R8\_SPIx\_CLOCK\_DIV)，配置 SPI 时钟速度；

(2)、设置 SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MDD)的 RB\_SPI\_MDE\_SLAVE 位为 0，配置 SPI 为主机模式；

(3)、设置 SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MDD)的 RB\_SPI\_MST\_SCK\_MDD 位，根据连接的设备需求设置为模式 0 或模式 3；

(4)、设置 SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MDD)的 RB\_SPI\_FIFO\_DIR 位，配置 FIFO 方向，若为 1 则当前 FIFO 方向为数据输入，若为 0 则当前 FIFO 方向为数据输出。

(5)、设置 SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MDD)的 RB\_SPI\_MOSI\_OE 位和 RB\_SPI\_SCK\_OE 位为 1，RB\_SPI\_MISO\_OE 位为 0，并设置 PB 口方向寄存器(R32\_PB\_DIR)中 MOSI 引脚和 SCK 引脚对应的位为 1，MISO 引脚对应的位为 0，将 MOSI 引脚和 SCK 引脚方向配置为输出，MISO 引脚方向配置为输入；

数据发送过程：

(1)、设置 SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MDD)的 RB\_SPI\_FIFO\_DIR 位为 0，置当前 FIFO 方向为输出；

(2)、写 R16\_SPIx\_TOTAL\_CNT 寄存器，设置要发送的数据长度；

(3)、写 R8\_SPIx\_FIFO 寄存器，往 FIFO 里写要发送的数据，如果 R8\_SPI0\_FIFO\_COUNT 小于 FIFO 大小则可以继续往 FIFO 写数据；

(4)、所有数据写入 FIFO 后，等待 R16\_SPIx\_TOTAL\_CNT 寄存器为 0，则说明数据发送完成，如果发送一字节，也可以等待 R8\_SPI0\_FIFO\_COUNT 为 0，则说明 FIFO 中没有数据即数据已发送完毕。

数据接收过程：

(1)、设置 SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MDD)的 RB\_SPI\_FIFO\_DIR 位为 1，置当前 FIFO 方向为输入；

(2)、写 R16\_SPIx\_TOTAL\_CNT 寄存器，设置要接收的数据长度；

(3)、等待 R8\_SPIx\_FIFO\_COUNT 寄存器不为 0，则说明接收到返回数据；

(4)、读取 R8\_SPIx\_FIFO 中的值即为接收到的数据。

### 7.4.2 SPI 从机模式配置

SPI 从机模式下，SCK 引脚用于接收连接的 SPI 主机的串行时钟。

配置步骤：

(1)、设置 SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MDD)的 RB\_SPI\_MDE\_SLAVE 位为 1，配置 SPI 为从机模式；

- (2)、根据需要设置 SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MDD)的 RB\_SPI\_SLV\_CMD\_MDD 位；
- (3)、设置 SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MDD)的 RB\_SPI\_FIFO\_DIR 位，配置 FIFO 方向，若为 1 则当前 FIFO 方向为数据输入，若为 0 则当前 FIFO 方向为数据输出；
- (4)、设置 SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MDD)的 RB\_SPI\_MDSI\_OE 位和 RB\_SPI\_SCK\_OE 位为 0，RB\_SPI\_MISO\_OE 位为 1，并设置 PB 口方向寄存器(R32\_PB\_DIR)中 MDSI 引脚、SCK 引脚和 SCS 引脚对应的位为 0，MISO 引脚对应的位为 1，将 MDSI 引脚、SCK 引脚和 SCS 引脚方向配置为输入，MISO 引脚方向配置为输出；注：在 SPI 设备模式下 MISO 引脚的 I/O 引脚方向，除了能够由 MISO 引脚方向配置为输出之外，还支持 SPI 片选有效期间自动配置为输出，但其输出数据由 RB\_SPI\_MISO\_OE 选择，当 RB\_SPI\_MISO\_OE 为 1 时输出 SPI 数据，当 RB\_SPI\_MISO\_OE 为 0 时输出 GPIO 寄存器的数据。
- (5)、可选的，设置 SPI 设备模式预置数据寄存器(R8\_SPI0\_SLAVE\_PRE)，用于被片选后首次自动加载到缓冲区中用于对外输出。当 8 个时钟过去之后（即首个数据字节在主从双方之间交换完毕），CH561 得到外部 SPI 主机发来的首字节数据（命令码），外部 SPI 主机交换得到 R8\_SPI0\_SLAVE\_PRE 中的预置数据（状态值）。该寄存器的位 7 将 SPI 片选有效后的 SCK 低电平期间自动加载到 MISO 引脚上，对于 SPI 模式 0（CLK 默认为低电平），如果 CH561 预置了 R8\_SPI0\_SLAVE\_PRE 的位 7，那么外部 SPI 主机将在 SPI 片选有效但尚未传输数据时，就能够通过查询 MISO 引脚得到 R8\_SPI0\_SLAVE\_PRE 的位 7 的预置值，从而通过仅仅有效一下 SPI 片选就能获得 R8\_SPI0\_SLAVE\_PRE 的位 7 的值。

数据发送过程：

- (1)、设置 SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MDD)的 RB\_SPI\_FIFO\_DIR 位为 0，配置当前 FIFO 方向为数据输出；
- (2)、将发送数据写入到 FIFO 寄存器(R8\_SPIx\_FIFO)中，并将 SPI 发送/接收数据总长度寄存器(R16\_SPIx\_TOTAL\_CNT)加 1，建议一次性将 R16\_SPIx\_TOTAL\_CNT 为较大的值，这样，只要 FIFO 中有数据就会自动发送，FIFO 空时自动暂停，不必每次设置 R16\_SPIx\_TOTAL\_CNT；
- (3)、如果发送单个字节，则等待 R16\_SPIx\_TOTAL\_CNT 寄存器为 0，等待数据发送完毕；如果发送多个字节，则可以一次性最多写入 32 个数据到 FIFO 寄存器(R8\_SPIx\_FIFO)中，再等待发送完毕；

数据接收过程：

- (1)、设置 SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MDD)的 RB\_SPI\_FIFO\_DIR 位为 1，配置当前 FIFO 方向为数据输入；
- (2)、等待查询 SPI 发送/接收数据总长度寄存器(R16\_SPIx\_TOTAL\_CNT)，如果该寄存器不为 0，则表示接收到数据，通过读取 FIFO 寄存器(R8\_SPIx\_FIFO)来获取接收到的数据。

单个字节的数据接收，也可以不使用 FIFO，可以直接读 SPI 数据缓冲区寄存器(R8\_SPIx\_BUFFER)得到对方给的当前数据启动传输。

## 7.5 DMA 功能

CH561 芯片的 SPI0 具有 DMA 功能，SPI1 不具备此功能。通过使能 DMA 功能，可以在减少软件干预的基础上更便捷的实现 SPI 数据的接收和发送。

### 7.5.1 SPI 主机模式 DMA 发送数据配置

- (1)、按照 7.4.1 节，将 SPI0 配置为主机模式；
- (2)、如果需要产生 DMA 完成中断，则将 SPI 中断使能寄存器(R8\_SPIx\_INTER\_EN)的 RB\_SPI\_IE\_DMA\_END 位置 1；
- (3)、初始化 R16\_SPI\_DMA\_BEG 寄存器为 SPI 数据发送缓冲区开始地址；
- (4)、初始化 R16\_SPI\_DMA\_END 寄存器为 SPI 数据发送缓冲区结束地址；
- (5)、清 SPI 中断状态寄存器(R8\_SPIx\_INT\_FLAG)；
- (6)、初始化 R16\_SPI\_TOTAL\_CNT 寄存器为需要发送的数据个数，如果 DMA 已使能则启动传输；
- (7)、如果需要使能 DMA 地址循环模式功能，则需要设置 SPI 的 DMA 控制寄存器(R8\_SPIx\_CTRL\_DMA)的 RB\_SPI\_DMA\_LOOP 位为 1；
- (8)、设置 SPI 的 DMA 控制寄存器(R8\_SPIx\_CTRL\_DMA)的 RB\_SPI\_DMA\_ENABLE 位为 1，使能 DMA 进行数据发送。

### 7.5.2 SPI 主机模式 DMA 接收数据配置

- (1)、按照 7.4.1 节，将 SPI0 配置为主机模式；
- (2)、设置 SPI 模式配置寄存器(R8\_SPIx\_CTRL\_MDD)的 RB\_SPI\_FIFO\_DIR 位为 1，配置 FIFO 方向为输入；
- (3)、如果需要产生 DMA 中断将寄存器 R8\_SPI\_INTER\_EN 的 RB\_SPI\_IE\_DMA\_END 位置 1，使能产生 DMA 结束中断；
- (4)、初始化 R16\_SPI\_DMA\_BEG 寄存器为 SPI 数据接收缓冲区开始地址；
- (5)、初始化 R16\_SPI\_DMA\_END 寄存器为 SPI 数据接收缓冲区结束地址；
- (6)、清 SPI 中断状态寄存器(R8\_SPIx\_INT\_FLAG)；
- (7)、初始化 R16\_SPI\_TOTAL\_CNT 寄存器为需要接收的数据个数，如果 DMA 已使能则启动传输；
- (8)、如果需要使能 DMA 地址循环模式功能，则需要设置 SPI 的 DMA 控制寄存器(R8\_SPIx\_CTRL\_DMA)的 RB\_SPI\_DMA\_LOOP 位为 1；
- (9)、设置 SPI 的 DMA 控制寄存器(R8\_SPIx\_CTRL\_DMA)的 RB\_SPI\_DMA\_ENABLE 位为 1，使能 DMA 进行数据接收。如果 DMA 使能在前，在设置完 R16\_SPI\_TOTAL\_CNT 后即自动开始传输。

## 第 8 章 通用异步收发器 UART

### 8.1 UART 简介

CH561 芯片拥有 2 个全双工的异步串口，UART0 和 UART1。支持全双工和半双工串口通讯，提供发送状态引脚用于切换 RS485。UART0 支持 MODEM 调制解调器信号 CTS、DSR、RI、DCD、DTR、RTS。

UART 特性：

- (1)、兼容 16C550 异步串口并且有所增强；
- (2)、支持 5、6、7 或者 8 个数据位以及 1 或者 2 个停止位；
- (3)、支持奇、偶、无校验、空白 0、标志 1 等校验方式；
- (4)、可编程通讯波特率，支持 115200bps 以及最高达 12Mbps 的通讯波特率；
- (5)、串口 0 内置 16 个字节的 FIFO 先进先出缓冲器，支持 4 个 FIFO 触发级；
- (6)、串口 1 内置 32 个字节的 FIFO 先进先出缓冲器，支持 4 个 FIFO 触发级；
- (7)、支持 MODEM 调制解调器信号 CTS、DSR、RI、DCD、DTR、RTS，可转成 RS232 电平；
- (8)、支持硬件流控制信号 CTS 和 RTS 自动握手和自动传输速率控制，兼容 TL16C550C；
- (9)、支持串口帧错误检测、支持 Break 线路间隔检测；
- (10)、支持全双工和半双工串口通讯，提供发送状态引脚用于切换 RS485；
- (11)、内置 SIR 红外线编解码器，支持 2400bps 到 115200bps 波特率的 IrDA 红外通讯；

### 8.2 UART 寄存器描述

UART0 相关寄存器物理起始地址为：0x0040 D000

UART0 相关寄存器列表(地址范围：0x00---0x07)

表 8-1 UART0 相关寄存器列表

名称	偏移地址	描述	复位值
R8_UART0_RBR	0x00	接收缓冲寄存器	8hxx
R8_UART0_THR	0x00	发送保持寄存器	8hxx
R8_UART0_IER	0x01	中断使能寄存器	8h00
R8_UART0_IIR	0x02	中断识别寄存器	8h01
R8_UART0_FCR	0x02	先进先出缓冲区 FIFO 控制寄存器，用于使能和复位 FIFO	8h00
R8_UART0_LCR	0x03	线路控制寄存器，用于控制串口通讯的格式	8h00
R8_UART0_MCR	0x04	调制解调器 MODEM 控制寄存器，用于控制 MODEM 输出	8h00
R8_UART0_LSR	0x05	线路状态寄存器，用于查询方式分析串口状态	8h60
R8_UART0_MSR	0x06	调制解调器 MODEM 状态寄存器，用于查询 MODEM 状态	8hxx
R8_UART0_DIV	0x07	预分频除数寄存器	最高位 0，其他位保持
R8_UART0_DLL	0x00	波特率除数锁存器，DLL 是低字节	保持
R8_UART0_DLM	0x01	波特率除数锁存器，DLM 是高字节	8h80

UART1相关寄存器物理起始地址为：0x0040 D800

UART1 相关寄存器列表(地址范围：0x00--0x07)

表 8-2 UART1 相关寄存器列表

名称	偏移地址	描述	复位值
R8_UART1_RBR	0x00	接收缓冲寄存器	8hxx
R8_UART1_THR	0x00	发送保持寄存器	8hxx
R8_UART1_IER	0x01	中断使能寄存器	8h00
R8_UART1_IIR	0x02	中断识别寄存器	8h01
R8_UART1_FCR	0x02	先进先出缓冲区 FIFO 控制寄存器，用于使能和复位 FIFO	8h00
R8_UART1_LCR	0x03	线路控制寄存器，用于控制串口通讯的格式	8h00
R8_UART1_MCR	0x04	调制解调器 MODEM控制寄存器，用于控制 MODEM输出	8h00
R8_UART1_LSR	0x05	线路状态寄存器，用于查询方式分析串口状态	8h60
R8_UART1_MSR	0x06	调制解调器 MODEM状态寄存器，用于查询 MODEM状态	8hxx
R8_UART1_ADR	0x07	串口 1 总线地址预置寄存器	8h00
R8_UART1_DIV	0x07	预分频除数寄存器	最高位 0，其他位保持
R8_UART1_DLL	0x00	波特率除数锁存器，DLL 是低字节	保持
R8_UART1_DLM	0x01	波特率除数锁存器，DLM是高字节	8h80

接收缓冲寄存器 (R8\_UARTx\_RBR) :

名称	位	访问	描述	复位值
UARTx_RBR	[7: 0]	RO	<b>接收缓冲寄存器</b> 如果 LSR 的 DATA_RDY 位为 1，则可以从该寄存器读取接收到的数据； 如果 FIFO_EN 为 1，则从串口移位寄存器 RSR 接收到的数据首先被存放于接收 FIFO 中，然后通过该寄存器读出	8hxx

发送缓冲寄存器 (R8\_UARTx\_THR) :

名称	位	访问	描述	复位值
UARTx_THR	[7: 0]	WO	<b>发送保持寄存器</b> 包括发送 FIFO，用于写入准备发送的数据；如果 FIFO_EN 为 1，则写入的数据首先被存放于发送 FIFO 中，然后通过发送移位寄存器 TSR 逐个输出	8hxx

中断使能寄存器 (R8\_UARTx\_IER) :

名称	位	访问	描述	复位值
RB_IER_RESET	7	RW	<b>串口软件复位使能位</b> 0：不进行软件复位	1h0

			1: 软件复位该串口 该位能够自动清 0, 无需软件清 0	
RB_IER_TXD_EN	6	RW	<b>TXD 输出使能/禁止位</b> 0: 禁止 TXD 输出到引脚 1: 使能 TXD 输出到引脚	1h0
RB_IER_MOUT_EN	5	RW	<b>RTS、DTR 输出使能/禁止位</b> 0: 禁止 RTS、DTR 输出到引脚 1: 使能 RTS、DTR 输出到引脚	1h0
RB_IER_MODEM_IO	4	RW	<b>对于串口 0: 串口调制解调器 MODEM 信号引脚选择位</b> 0: MODEM信号引脚选择 PA 端口 1: MODEM信号引脚选择 PB 端口	1h0
RB_IER_IRDA_MDD	4	RW	<b>对于串口 1: IrDA 功能使能/禁止位</b> 0: 禁止 IRDA 红外编解码模式 1: 使能 IRDA 红外编解码模式 在启用 IRDA 模式后, RXD 和 TXD 引脚可以直接连接 ZHX1810、HSD3000、TFBS4711、TFDU4100 等红外线收发器的 RXD 和 TXD 引脚	1h0
RB_IER_MODEM_CHG	3	RW	<b>调制解调器输入状态变化中断使能/禁止位</b> 0: 禁止产生调制解调器输入状态变化中断 1: 使能产生调制解调器输入状态变化中断	1h0
RB_IER_LINE_STAT	2	RW	<b>接收线路状态中断使能/禁止位</b> 0: 禁止产生接收线路状态中断 1: 使能产生接收线路状态中断	1h0
RB_IER_THR_EMPTY	1	RW	<b>发送保持寄存器空中断使能/禁止位</b> 0: 禁止产生发送保持寄存器空中断 1: 使能产生发送保持寄存器空中断	1h0
RB_IER_RECV_RDY	0	RW	<b>接收到数据中断使能/禁止位</b> 0: 禁止产生接收到数据中断 1: 使能产生接收到数据中断	1h0

## 中断识别寄存器(R8\_UARTx\_IIR):

名称	位	访问	描述	复位值
RB_IIR_FIFO_ID	7	RO	<b>串口 FIFO 启用状态位</b> 0: FIFO 未启用 1: FIFO 已启用	1h0
RB_IIR_FIFO_EN	6	RO	<b>串口 FIFO 启用状态位</b> 0: FIFO 未启用 1: FIFO 已启用	1h0
Reserved	[5: 4]	-	-	-
RB_IIR_INT_MASK	[3: 1]	RO	<b>中断标志域</b> 如果 RB_IIR_NO_INT 位为 0, 则表示有中断产生, 需要读取该域判断中断源 具体请参看下表	3h0



RB_IIR_NO_INT	0	RO	<b>中断状态标志位</b> 0: 有中断产生 1: 无中断产生	1h1
---------------	---	----	--	-----

中断识别寄存器(R8\_UARTx\_IIR)的RB\_IIR\_NO\_INT位以及RB\_IIR\_INT\_MASK域的每一个位所表示的含义如下表所示:

表 8-3 IIR 寄存器位含义

IIR 寄存器位				优先级	中断类型	中断源	清中断方法
IID3	IID2	IID1	NOINT				
0	0	0	1	无	没有中断产生	没有中断	
1	1	1	0	0	总线地址匹配	接收到 1 个数据是串口总线地址, 且该地址与预置值相匹配或是广播地址 <b>注: 该中断只适用 UART1</b>	读 IIR 或禁用
0	1	1	0	1	接收线路状态	OV_ERR、PAR_ERR、FRAM_ERR、BREAK_IN	读 LSR
0	1	0	0	2	接收数据可用	接收到的字节数达到 FIFO 的触发点	读 RBR
1	1	0	0	2	接收数据超时	超过 4 个数据的时间未收到下一数据	读 RBR
0	0	1	0	3	THR 寄存器空	发送保持寄存器空, IE_THRE 从 0 变 1 可以重新使能中断	读 IIR 或写 THR
0	0	0	0	4	MODEM输入变化	△CTS、△DSR、△RI、△DCD	读 MSR

FIFO 控制寄存器(R8\_UARTx\_FCR):

名称	位	访问	描述	复位值
RB_FCR_FIFO_TRIG	[7: 6]	WO	<b>接收 FIFO 的中断和硬件流控制的触发点设置域</b> <b>串口 0:</b> 00: 1 字节 01: 4 字节 10: 8 字节 11: 14 字节 <b>串口 1:</b> 00: 2 字节 01: 8 字节 10: 16 字节 11: 28 字节 该域用来设置接收 FIFO 的中断和硬件流控制的触发点, 对于串口 0, 00 对应 1 个字节, 即接收满 1 个字节产生接收数据可用的中断, 并在使能硬件流控制时自动无效 RTS 引脚	2h0
Reserved	[5: 3]	-	-	-
RB_FCR_TX_FIFO_CLR	2	WO	<b>发送 FIFO 数据清空使能位</b> 0: 不清空发送 FIFO 中的数据 1: 清空发送 FIFO 中的数据(不含 TSR) 该位能够自动清 0, 无需软件清 0	1h0
RB_FCR_RX_FIFO_CLR	1	WO	<b>接收 FIFO 数据清空使能位</b> 0: 不清空接收 FIFO 中的数据	1h0

			1: 清空接收 FIFO 中的数据(不含 RSR) 该位能够自动清 0, 无需软件清 0	
RB_FCR_FIFO_EN	0	WO	<b>FIFO 使能/禁止位</b> 0: 禁用 FIFO 1: 启用 FIFO 禁用 FIFO 后为 16C450 兼容模式, 相当于 FIFO 只有一个字节 (RECV_TG1=0、RECV_TG0=0、FIFO_EN=1), 建议启用 FIFO	1h0

线路控制寄存器 (R8\_UARTx\_LCR) :

名称	位	访问	描述	复位值
RB_LCR_DLAB	7	RW	<b>除数锁存器存取使能位</b> 0: 允许存取 RBR/THR/IER/ADR 寄存器 1: 允许存取 DLL 和 DLM 以及 DIV 寄存器	1h0
RB_LCR_BREAK_EN	6	RW	<b>强制产生 BREAK 线路间隔使能/禁止位</b> 0: 不产生 BREAK 线路间隔 1: 强制产生 BREAK 线路间隔	1h0
RB_LCR_PAR_MDD	[5: 4]	RW	<b>奇偶校验位格式设置域</b> 00: 奇校验 01: 偶校验 10: 标志位 (MARK, 置 1) 11: 空白位 (SPACE, 清 0) 该域仅当 RB_LCR_PAR_EN 位为 1 时有效	2h0
RB_LCR_PAR_EN	3	RW	<b>奇偶校验位使能/禁止位</b> 0: 无奇偶校验位 1: 允许发送时产生和接收时校验奇偶校验位	1h0
RB_LCR_STOP_BIT	2	RW	<b>停止位格式设置位</b> 0: 一个停止位 1: 两个停止位	1h0
RB_LCR_WORD_SZ	[1: 0]	RW	<b>字长设置域</b> 00: 5 个数据位 01: 6 个数据位 10: 7 个数据位 11: 8 个数据位	2h0

调制解调器 MDEM 控制寄存器 (R8\_UARTx\_MCR) :

名称	位	访问	描述	复位值
RB_MCR_HALF	7	RW	<b>半双工收发模式使能/禁止位</b> 0: 禁止半双工收发模式 1: 进入半双工收发模式, 发送优先, 不发送时为接收	1h0
RB_MCR_TNOW	6	RW	串口 0 的 DTR 引脚或串口 1 的 RTS 引	1h0

			<b>脚输出正在发送状态位</b> 0: 无效 1: 串口 0 的 DTR 引脚或串口 1 的 RTS 引脚输出正在发送状态 TNOW, 可以用于控制 RS485 收发切换	
RB_MCR_AU_FLOW_EN	5	RW	<b>允许 CTS 和 RTS 硬件自动流控制位</b> 0: 无效 1: 允许 CTS 和 RTS 硬件自动流控制 如果 AFE 为 1, 那么仅在检测到 CTS 引脚输入有效 (低电平有效) 时串口才继续发送下一个数据, 否则暂停串口发送, 当 AFE 为 1 时的 CTS 输入状态变化不会产生 MODEM 状态中断。如果 AFE 为 1 并且 RTS 为 1, 那么当接收 FIFO 空时, 串口会自动有效 RTS 引脚 (低电平有效), 直到接收的字节数达到 FIFO 的触发点时, 串口才自动无效 RTS 引脚, 并能够在接收 FIFO 空时再次有效 RTS 引脚。使用硬件自动带率控制, 可将己方的 CTS 引脚接到对方的 RTS 引脚, 并将己方的 RTS 引脚送到对方的 CTS 引脚	1h0
RB_MCR_LOOP	4	RW	<b>内部回路的测试模式使能/禁止位</b> 0: 禁止内部回路的测试模式 1: 使能内部回路的测试模式 在内部回路的测试模式下, 串口所有对外输出引脚均为无效状态, TXD 内部返回到 RXD (即 TSR 的输出内部返回到 RSR 的输入), RTS 内部返回到 CTS, DTR 内部返回到 DSR, OUT1 内部返回到 RI, OUT2 内部返回到 DCD <b>注: 该位仅对 UART0 有效</b>	1h0
RB_MCR_OUT2	3	RW	<b>串口的中断请求输出使能/禁止位</b> 0: 禁止串口的中断请求输出 1: 使能串口的中断请求输出	1h0
RB_MCR_OUT1	2	RW	<b>MODEM 控制位用户自定义使能/禁止位</b> 0: 禁止用户自定义 MODEM 控制位 1: 允许用户自定义 MODEM 控制位 没有连接实际输出引脚 <b>注: 该位仅对 UART0 有效</b>	1h0
RB_MCR_RTS	1	RW	<b>RTS 引脚输出有效使能/禁止位</b> 0: 禁止 RTS 引脚输出有效 1: 使能 RTS 引脚输出有效 (低电平有效)	1h0
RB_MCR_DTR	0	RW	<b>DTR 引脚输出有效使能/禁止位</b> 0: 禁止 DTR 引脚输出有效 1: 使能 DTR 引脚输出有效 (低电平有效)	1h0

			注：该位仅对 UART0 有效	
--	--	--	-----------------	--

线路状态寄存器 (R8\_UARTx\_LSR) :

名称	位	访问	描述	复位值
RB_LSR_ERR_RX_FIFO	7	RO	<b>接收 FIFO 错误标志位</b> 0: 接收 FIFO 未存在错误 1: 接收 FIFO 中存在至少一个 PAR_ERR、FRAM_ERR 或 BREAK_IN 错误	1h0
RB_LSR_TX_ALL_EMP	6	RO	<b>发送保持寄存器 THR 和发送移位寄存器 TSR 全空标志位</b> 0: 两者非全空 1: 两者全空	1h1
RB_LSR_TX_FIFO_EMP	5	RO	<b>发送保持寄存器 THR 空标志位</b> 0: 发送保持寄存器非空 1: 发送保持寄存器空	1h1
RB_LSR_BREAK_ERR	4	RZ	<b>BREAK 线路间隔检测标志位</b> 0: 未检测到 BREAK 线路间隔 1: 检测到 BREAK 线路间隔	1h0
RB_LSR_FRAME_ERR	3	RZ	<b>数据帧错误标志位</b> 0: 数据帧正确 1: 表示正在从接收 FIFO 中读取的数据的帧错误, 缺少有效的停止位	1h0
RB_LSR_PAR_ERR	2	RZ	<b>奇偶校验错误标志位</b> 0: 奇偶校验正确 1: 表示正在从接收 FIFO 中读取的数据的奇偶校验错	1h0
RB_LSR_OVER_ERR	1	RZ	<b>接收 FIFO 缓冲区溢出标志位</b> 0: 未溢出 1: 已溢出	1h0
RB_LSR_DATA_RDY	0	RO	<b>接收 FIFO 中有接收到的数据标志位</b> 0: 无数据 1: 有数据 读取 FIFO 中所有数据后, 该位自动清 0	1h0

调制解调器 MDEM 状态寄存器 (R8\_UARTx\_MSR) :

名称	位	访问	描述	复位值
RB_MSR_DCD	7	RO	<b>DCD 引脚状态位</b> 该位是 DCD 引脚的位反, 为 1 表示 DCD 引脚有效 (低电平有效) 注：该位仅对 UART0 有效	1hx
RB_MSR_RI	6	RO	<b>RI 引脚状态位</b> 该位是 RI 引脚的位反, 为 1 表示 RI 引脚有效 (低电平有效) 注：该位仅对 UART0 有效	1hx
RB_MSR_DSR	5	RO	<b>DSR 引脚状态位</b> 该位是 DSR 引脚的位反, 为 1 表示 DSR	1hx

			引脚有效（低电平有效）。 <b>注：该位仅对 UART0 有效</b>	
RB_MSR_CTS	4	RO	CTS 引脚状态位 该位是 CTS 引脚的位反, 为 1 表示 CTS 引脚有效（低电平有效）	1hx
RB_MSR_DCD_CHG	3	RZ	DCD 引脚输入状态变化标志位 0: 无变化 1: 发生变化 <b>注：该位仅对 UART0 有效</b>	1hx
RB_MSR_RI_CHG	2	RZ	RI 引脚输入状态变化标志位 0: 无变化 1: 发生变化 <b>注：该位仅对 UART0 有效</b>	1hx
RB_MSR_DSR_CHG	1	RZ	DSR 引脚输入状态变化标志位 0: 无变化 1: 发生变化 <b>注：该位仅对 UART0 有效</b>	1hx
RB_MSR_CTS_CHG	0	RZ	CTS 引脚输入状态变化标志位 0: 无变化 1: 发生变化	1hx

预分频除数寄存器 (R8\_UARTx\_DIV) :

名称	位	访问	描述	复位值
UARTx_DIV	[7: 0]	RW	用于将系统主时钟倍频后再进行预分频, 产生串口的内部基准时钟, 该除数 = 系统主时钟 * 2 / 串口内部基准时钟, 最大值 127 如果系统主时钟为 100MHz, 除数为 108, 那么串口内部基准时钟为 1.852MHz, 与常用的基准时钟 1.8432 相差 0.47%	8h00

串口 1 总线地址预置寄存器 (R8\_UART1\_ADR) :

名称	位	访问	描述	复位值
R8_UART1_ADR	[7: 0]	RW	<b>总线地址预置寄存器</b> 预置本机地址用于在接收到地址时自动比较, 并在地址匹配时或者在接收到广播地址 0FFH 时产生中断, 同时允许接收后续数据。在地址没有匹配之前不接收任何数据, 开始发送数据后或者重写 R8_UART1_ADR 寄存器后停止接收任何数据, 直到下次地址再次匹配或者接收到广播地址时再允许接收。R8_UART1_ADR 为 0FFH 时或者 RB_LCR_PAR_EN 为 0 时则禁用总线地址自动比较功能。R8_UART1_ADR 不为 0FFH 并且 RB_LCR_PAR_EN 为 1 时启用总线地址自动比较功能, 同时应该配置下述参数: 选择 8 个数据位	8h00

			方式，RB_LCR_PAR_MOD1 应该为 1，RB_LCR_PAR_MOD0 为 1 时用于地址字节 MARK 模式（即数据字节的位 9 为 0），RB_LCR_PAR_MOD0 为 0 时用于地址字节 SPACE 模式（即数据字节的位 9 为 1）。
--	--	--	--

波特率除数锁存器(R8\_UARTx\_DLL/R8\_UARTx\_DLM)：

名称	位	访问	描述	复位值
UARTx_DLL UARTx_DLM	[7: 0]	RW	DLL 是低字节，DLM是高字节，两者组成的 16 位除数用于由 16 位计数器构成的串口波特率产生器 该除数 = 串口内部基准时钟 / 16 / 所需通讯波特率 如果串口内部基准时钟为 1.8432MHz，所需波特率为 9600bps，则除数=1843200/16/9600=12	8h00

### 8.3 UART 应用

CH561 芯片的 UART 输出引脚都是 3.3V LVCMOS 电平，兼容 5V TTL 电平，RXD 和 TXD 以及 PA 端口的 MODEM 引脚能够承受 5V 耐压。异步串口方式下引脚包括：数据传输引脚和 MODEM 联络信号引脚。数据传输引脚包括：TXD 引脚和 RXD 引脚，默认都是高电平；MODEM 联络信号引脚包括：CTS 引脚、DSR 引脚、RI 引脚、DCD 引脚、DTR 引脚、RTS 引脚，默认都是高电平。所有这些 MODEM 联络信号都可以作为通用 I/O 引脚，由应用程序控制并定义其用途。

UART 内置了独立的收发缓冲区及 FIFO，支持单工、半双工或者全双工异步串行通讯。串行数据包括 1 个低电平起始位，5、6、7 或 8 个数据位，0 个或者 1 个附加校验位或者标志位，1 个或者 2 个高电平停止位，支持奇校验/偶校验/标志校验/空白校验。支持常用通讯波特率：1200、2400、4800、9600、19.2K、38.4K、57.6K、115.2K、230.4K、460.8K、921.6K、1.8432M、2.7648M、7.8125M 等。串口发送信号的波特率误差小于 0.2%，串口接收信号的允许波特率误差不大于 2%。

## 第 9 章 通用定时器 TIM<sub>k</sub>

### 9.1 TIM<sub>k</sub> 简介

CH561 芯片带有 4 个 28 位定时器，TIM0、TIM1、TIM2 和 TIM3，最长定时时间为  $2^{28}$  个时钟周期。所有定时器均支持 PWM 功能，支持中断功能，同时定时器 0, 1, 2 支持捕获以及 PWM 的 DMA 功能。特性：

- (1)、4 个 28 位定时器，每个定时器定时时间最大为  $2^{28}$  个时钟周期；
- (2)、每个定时器都支持 PWM 功能；
- (3)、定时器 0, 1, 2 支持捕获功能；
- (4)、每个定时器都支持定时器中断，TIM0、TIM1 和 TIM2 支持 DMA 中断；
- (5)、定时器 0, 1, 2 支持 PWM 以及捕获的 DMA 功能，捕获功能可设置为电平变化捕获功能和高或低电平保持时间捕获功能；
- (6)、PWM 功能支持动态的调整 PWM 占空比设置；
- (7)、定时器 3 支持外部输入时钟的计数器功能。

### 9.2 TIM<sub>k</sub> 寄存器描述

TIM0 相关寄存器物理起始地址为：0x0040 8000

TIM0 相关寄存器列表(地址范围：0x00---0x1D)

表 9-1 TIM0 相关寄存器列表

名称	偏移地址	描述	复位值
R8_TMRO_CTRL_MD	0x00	模式设置寄存器	8h00
R8_TMRO_CTRL_DMA	0x01	DMA 控制寄存器	8h00
R8_TMRO_INTER_EN	0x02	中断使能寄存器	8h00
R8_TMRO_INT_FLAG	0x06	中断标志寄存器	8h00
R8_TMRO_FIFO_COUNT	0x07	计数状态寄存器	8h00
R32_TMRO_COUNT	0x08	当前计数值寄存器	32h0000 0000
R32_TMRO_CNT_END	0x0C	计数终值寄存器	32h0000 0000
R32_TMRO_FIFO	0x10	FIFO 寄存器	32h0000 0000
R16_TMRO_DMA_NOW	0x14	DMA 当前缓冲区地址	16h0000
R16_TMRO_DMA_BEG	0x18	DMA 开始缓冲区地址	16h0000
R16_TMRO_DMA_END	0x1C	DMA 结束缓冲区地址	16h0000

TIM1 相关寄存器物理起始地址为：0x0040 8400

TIM1 相关寄存器列表(地址范围：0x00---0x1D)

表 9-2 TIM1 相关寄存器列表

名称	偏移地址	描述	复位值
R8_TMR1_CTRL_MD	0x00	模式设置寄存器	8h00
R8_TMR1_CTRL_DMA	0x01	DMA 控制寄存器	8h00
R8_TMR1_INTER_EN	0x02	中断使能寄存器	8h00
R8_TMR1_INT_FLAG	0x06	中断标志寄存器	8h00
R8_TMR1_FIFO_COUNT	0x07	计数状态寄存器	8h00
R32_TMR1_COUNT	0x08	当前计数值寄存器	32h0000 0000
R32_TMR1_CNT_END	0x0C	计数终值寄存器	32h0000 0000
R32_TMR1_FIFO	0x10	FIFO 寄存器	32h0000 0000
R16_TMR1_DMA_NOW	0x14	DMA 当前缓冲区地址	16h0000

R16_TMR1_DMA_BEG	0x18	DMA 开始缓冲区地址	16h0000
R16_TMR1_DMA_END	0x1C	DMA 结束缓冲区地址	16h0000

TIM2相关寄存器物理起始地址为：0x0040 8800

TIM2 相关寄存器列表(地址范围：0x00---0x1D)

表 9-3 TIM2 相关寄存器列表

名称	偏移地址	描述	复位值
R8_TMR2_CTRL_MDD	0x00	模式设置寄存器	8h00
R8_TMR2_CTRL_DMA	0x01	DMA 控制寄存器	8h00
R8_TMR2_INTER_EN	0x02	中断使能寄存器	8h00
R8_TMR2_INT_FLAG	0x06	中断标志寄存器	8h00
R8_TMR2_FIFO_COUNT	0x07	计数状态寄存器	8h00
R32_TMR2_COUNT	0x08	当前计数值寄存器	32h0000 0000
R32_TMR2_CNT_END	0x0C	计数终值寄存器	32h0000 0000
R32_TMR2_FIFO	0x10	FIFO 寄存器	32h0000 0000
R16_TMR2_DMA_NOW	0x14	DMA 当前缓冲区地址	16h0000
R16_TMR2_DMA_BEG	0x18	DMA 开始缓冲区地址	16h0000
R16_TMR2_DMA_END	0x1C	DMA 结束缓冲区地址	16h0000

TIM3相关寄存器物理起始地址为：0x0040 8C00

TIM3 相关寄存器列表(地址范围：0x00---0x10)

表 9-4 TIM3 相关寄存器列表

名称	偏移地址	描述	复位值
R8_TMR3_CTRL_MDD	0x00	模式设置寄存器	8h00
R8_TMR3_INTER_EN	0x02	中断使能寄存器	8h00
R8_TMR3_INT_FLAG	0x06	中断标志寄存器	8h00
R8_TMR3_FIFO_COUNT	0x07	计数状态寄存器	8h00
R32_TMR3_COUNT	0x08	当前计数值寄存器	32h0000 0000
R32_TMR3_CNT_END	0x0C	计数终值寄存器	32h0000 0000
R32_TMR3_FIFO	0x10	FIFO 寄存器	32h0000 0000

模式设置寄存器(R8\_TMRx\_CTRL\_MDD)：

名称	位	访问	描述	复位值
RB_TMR_CATCH_EDGE	[7: 6]	RW	<b>捕获模式边沿触发方式设置域</b> 00: 禁止触发 01: 捕获任何边沿变化之间的时间 10: 捕获下降沿到下降沿之间时间 11: 捕获上升沿到上升沿之间时间	2h0
RB_TMR_PWM_REPEAT	[7: 6]	RW	<b>PWM重复模式设置域</b> 00: 重复 1 次 01: 重复 4 次 10: 重复 8 次 11: 重复 16 次	2h0
RB_TMR3_MODE_COUNT	5	RW	<b>TIM3 外部时钟计数器模式设置位</b> 0: 定时、PWM以及捕获模式 1: 外部时钟计数器模式	1h0
RB_TMR_MODE_NRZI	5	RW	<b>TIM0、TIM1 串行编解码模式</b> 0: 定时、PWM以及捕获模式	1h0



			1: 保留	
RB_TMR_CAT_WIDTH	4	RW	<b>捕获模式下最小捕获脉宽设置位</b> 0: 16 个时钟周期, 小于则忽略 1: 8 个时钟周期, 小于则忽略	1h0
RB_TMR_OUT_POLAR	4	RW	<b>PWM模式下默认输出极性设置位</b> 0: 默认低电平, 高电平有效 1: 默认高电平, 低电平有效	1h0
RB_TMR_OUT_EN	3	RW	<b>定时器输出使能/禁止位</b> 0: 定时器输出禁止 1: 定时器输出使能	1h0
RB_TMR_COUNT_EN	2	RW	<b>定时器计数使能/禁止位</b> 0: 定时器计数禁止 1: 定时器计数使能	1h0
RB_TMR_ALL_CLEAR	1	RW	<b>计数器的 FIFO 和 COUNT 寄存器清 0 位</b> 0: 无动作 1: 清除	1h0
RB_TMR_MODE_IN	0	RW	<b>定时器输出模式设置位</b> 0: 定时器/PWM模式 1: 捕获模式	1h0

DMA 控制寄存器(R8\_TMRx\_CTRL\_DMA) :

名称	位	访问	描述	复位值
Reserved	[7: 3]	-	-	-
RB_TMR_DMA_LOOP	2	RW	<b>DMA 地址循环模式使能/禁止位</b> 0: 禁止 DMA 地址循环模式功能 1: 使能 DMA 地址循环模式功能 如果使能 DMA 地址循环模式功能, 当 DMA 地址增加到设置的末尾地址时, 自动循环指向设置的首地址 <b>注: TIMB 不支持</b>	1h0
RB_TMR_DMA_BURST	1	RW	<b>DMA 块操作使能/禁止位</b> 0: 禁止 DMA 块操作功能 1: 使能 DMA 块操作功能 如果使能 DMA 块操作功能, 则该 DMA 更少占用系统总线 <b>注: TIMB 不支持</b>	1h0
RB_TMR_DMA_ENABLE	0	RW	<b>DMA 功能使能/禁止位</b> 0: 禁止 DMA 功能 1: 使能 DMA 功能 <b>注: TIMB 不支持</b>	1h0

中断使能寄存器(R8\_TMRx\_INTER\_EN) :

名称	位	访问	描述	复位值
RB_TMR3_FORCE_EN	7	RW	<b>TIM0、TIM1 和 TIM2 强制同时使能位</b> 0: 无效 1: TIM0、TIM1 和 TIM2 同时使能 该位独立于 RB_TMR_COUNT_EN 位	1h0

Reserved	6	-	-	-
RB_TMR_IE_DMA_ERR	5	RW	DMA 错误响应中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_TMR_IE_FIFO_OV	4	RW	FIFO 溢出中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_TMR_IE_DMA_END	3	RW	DMA 结束中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_TMR_IE_FIFO_HF	2	RW	FIFO 过半中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_TMR_IE_DATA_ACT	1	RW	捕获模式下电平变化中断使能/禁止位 0: 禁止 1: 使能 PWM模式下 PWM结束中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_TMR_IE_CYC_END	0	RW	捕获模式下捕获超时中断使能/禁止位 0: 禁止 1: 使能 PWM模式下 PWM时钟周期结束使能/禁止位 0: 禁止 1: 使能	1h0

中断标志寄存器 (R8\_TMRx\_INT\_FLAG) :

名称	位	访问	描述	复位值
Reserved	[7: 6]	-	-	-
RB_TMR_IF_DMA_ERR	5	RW	DMA 响应错误标志位 0: 无错误发生 1: DMA 发生错误	1h0
RB_TMR_IF_FIFO_OV	4	RW	FIFO 溢出标志位 0: 未溢出 1: 已溢出	1h0
RB_TMR_IF_DMA_END	3	RW	DMA 完成标志位 0: 未完成 1: 已完成	1h0
RB_TMR_IF_FIFO_HF	2	RW	FIFO 过半标志位 0: FIFO 计数未过半 1: FIFO 计数已过半	1h0
RB_TMR_IF_DATA_ACT	1	RW	捕获模式下捕获到边沿变化标志位 0: 未捕获到边沿变化 1: 捕获到边沿变化 PWM模式下 PWM输出完成标志位 0: 未完成	1h0

			1: 已完成	
RB_TMR_IF_CYC_END	0	RW	捕获模式下超时标志位 0: 未超时 1: 已超时 PWM模式下 PWM时钟周期结束标志位 0: 未结束 1: 已结束	1h0

计数状态寄存器 (R8\_TMRx\_FIFO\_COUNT):

名称	位	访问	描述	复位值
TMRx_FIFO_COUNT	[7: 0]	RO	FIFO 计数	8h0

当前计数值寄存器 (R32\_TMRx\_COUNT):

名称	位	访问	描述	复位值
Reserved	[31: 28]	-	-	-
TMRx_COUNT	[27: 0]	RO	计数器当前计数值	28h0

计数终值寄存器 (R32\_TMRx\_CNT\_END):

名称	位	访问	描述	复位值
Reserved	[31: 28]	-	保留	-
TMRx_CNT_END	[27: 0]	RW	定时器模式下定时时钟数; PWM模式下 PWM周期总时钟数; 捕获模式下捕获超时时钟数; 注: R32_TMRx_COUNT 是从 0 起计数, 所以最大值为 R32_TMRx_CNT_END 减 1	28h0

FIFO 寄存器 (R32\_TMRx\_FIFO):

名称	位	访问	描述	复位值
Reserved	[31: 28]	-	保留	-
TMRx_FIFO	[27: 0]	RO/ WO	FIFO 数据寄存器	28h0

DMA 当前缓冲区地址 (R16\_TMRx\_DMA\_NOW):

名称	位	访问	描述	复位值
TMRx_DMA_NOW	[15: 0]	RW	DMA 数据缓冲区当前地址 可以作为已转换次数的计算, 计算方法为: $COUNT = (TMR\_DMA\_NOW - TMR\_DMA\_BEG) / 4$	16h0

DMA 开始缓冲区地址 (R16\_TMRx\_DMA\_BEG):

名称	位	访问	描述	复位值
TMRx_DMA_BEG	[15: 0]	RW	DMA 数据缓冲区开始地址 即 PWM 进行数据发送或者捕获模式 下, 开始捕获到的数据从此缓冲区 地址开始, 该地址必须 4 字节对齐	16h0

DMA 结束缓冲区地址 (R16\_TMRx\_DMA\_END) :

名称	位	访问	描述	复位值
TMRx_DMA_END	[15: 0]	RW	DMA 数据缓冲区结束地址 即 PWM 进行数据发送或者捕获模式下, 开始捕获到的数据至此缓冲区地址结束, 该地址必须 4 字节对齐	16h0

## 9.3 TIMx 功能

### 9.3.1 定时、计数功能

CH561 的 4 个定时器, 每一个支持最长定时时间为  $2^{28}$  个时钟周期。如果系统时钟周期为 100M, 则最长定时时间为:  $10\text{ns} \times 2^{28} \approx 2.68\text{s}$ 。如果系统时钟低于 100M, 则定时时间更长。

4 个定时器都有独立的中断, 每一个中断支持 FIQ 中断和 IRQ 中断。

定时功能寄存器初始化如下:

- (1)、设置寄存器 R32\_TMR\_CNT\_END 为需要定时的时间值;  
具体计算方法为:  $\text{Time} = \text{Pclk} * \text{R32\_TMR\_CNT\_END}$
- (2)、设置寄存器 R8\_TMR\_CTRL\_MDD 中的 RB\_TMR\_MMODE\_IN 位为 0, RB\_TMR\_ALL\_CLEAR 位为 0;
- (3)、将寄存器 R8\_TMR\_CTRL\_MDD 的 RB\_TMR\_COUNT\_EN 位置 1, 启动定时器功能;
- (4)、定时时间结束时, 寄存器 R8\_INT\_FLAG\_0 的 RB\_IF\_TMR 位将置 1。

### 9.3.2 PWM 功能

CH561 芯片的 4 个定时器, 均具有 PWM 功能。PWM 可设置默认输出极性为高电平或低电平, 重复次数可选为 1, 4, 8 或 16 次, 该重复功能结合 DMA 可以用于模仿 DAC 的效果。PWM 输出最短时间周期为 1 个系统时钟, 可动态修改 PWM 的占空比, 模仿出特殊波形, 例如正弦波。

PWM 功能操作:

PWM 输出时需要设置寄存器 (R32\_TMRx\_FIFO) 和寄存器 (R32\_TMRx\_CNT\_END), R32\_TMRx\_FIFO 为数据寄存器, R32\_TMRx\_CNT\_END 为 PWM 总周期寄存器。

PWM 操作步骤如下:

- (1)、设置 PWM 总周期寄存器 R32\_TMR\_CNT\_END, 最小值为 1, 该寄存器的值必须大于等于 R32\_TMR\_FIFO 寄存器的值;
- (2)、设置数据寄存器 R32\_TMR\_FIFO, 最小值为 0, 对应占空比 0%, 最大值同 R32\_TMR\_CNT\_END, 对应占空比 100%, 支持连续动态数据 (DMA), 可以模仿出特殊波形;
- (3)、将模式设置寄存器 (R8\_TMRx\_CTRL\_MDD) 中的 RB\_TMR\_MMODE\_IN 位清 0, 启用 PWM 模式; 同时将 RB\_TMR\_ALL\_CLEAR 位清 0 强制清除 FIFO 和 COUNT; 如果需要设置重复次数, 则根据需要设置 RB\_TMR\_PWM\_REPEAT 域。如果当前定时器是 TIMD 或 TIMI, 则需要将 RB\_TMR\_MMODE\_NRZI 位清 0, 如果当前定时器是 TIMB, 则需要将 RB\_TMR3\_MMODE\_COUNT 位清 0;
- (4)、将模式设置寄存器 (R8\_TMRx\_CTRL\_MDD) 中的 RB\_TMR\_COUNT\_EN 位和 RB\_TMR\_OUT\_EN 位置 1, 开启 PWM 功能;
- (5)、将 PWM 对应的 I/O 引脚设置为输出;
- (6)、如果需要启用中断则设置相应的中断使能寄存器位;
- (7)、PWM 完成后, 如果开启中断则产生相对应的定时器中断, 同时通过读取 TMR\_INT\_FLAG 寄存器了解 PWM 是否完成以及 PWM 过程中是否产生错误;

例如: 设置 RB\_TMR\_OUT\_POLAR 位为 0, R32\_TMR\_FIFO 为 6, R32\_TMR\_CNT\_END 为 18, 则产生 PWM 的基本时序图如下所示, 其占空比为:

$$\text{PWM 占空比} = \text{R32\_TMR\_FIFO} / \text{R32\_TMR\_CNT\_END} = 1/3$$

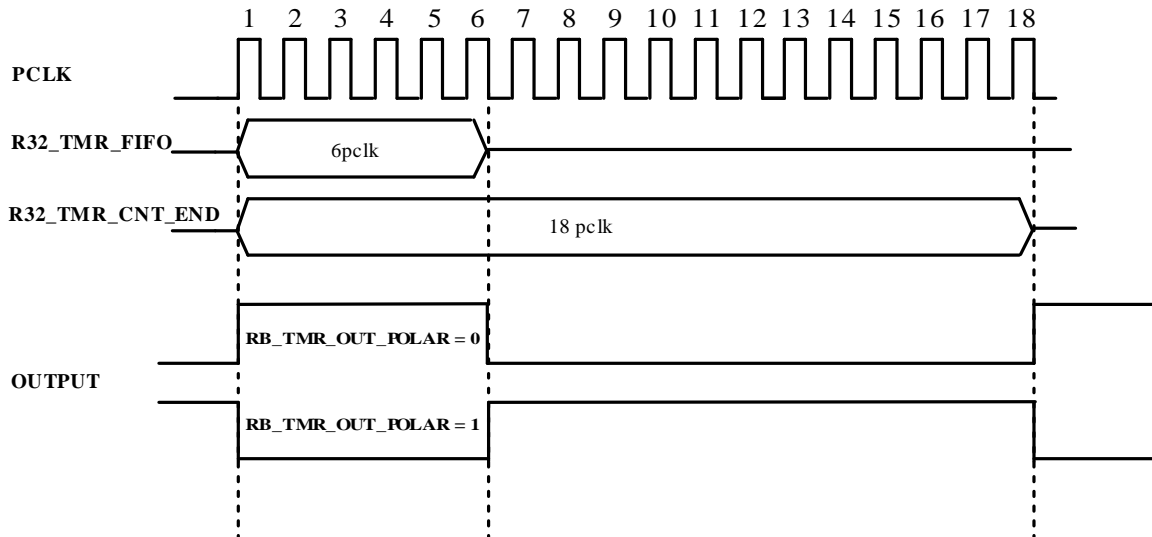


图 9-1 PWM输出时序图

如果 RB\_TMR\_PWM\_REPEAT 域设置为 00 则表示上述过程重复 1 次，01 表示重复 4 次，10 表示重复 8 次，11 表示重复 16 次。重复之后再取 FIFO 中的下一个数据继续。

### 9.3.3 捕获功能

CH561 芯片的 4 个定时器都具备捕获功能，TIMD、TIM1、TIM2 捕获功能支持 DMA 方式存储。捕获最短时间周期可以选择 8 个时钟周期或 16 个时钟周期，捕获模式可以选择任何边沿触发开始至任何边沿触发结束、上升沿触发开始至上升沿触发结束或下降沿触发开始至下降沿触发结束三种模式。以下为捕获触发模式说明表：

表 9-1 捕获触发模式说明表

捕获模式选择位 (RB_TMR_CATCH_EDGE)	触发方式	图示
00	禁止捕获	无
01	边沿触发	
10	下降沿至下降沿	
11	上升沿至上升沿	

边沿触发模式下有 2 种触发状态，可以捕获高电平宽度或低电平宽度。数据寄存器（R32\_TMR\_FIFO）的有效数据最高位（位 27）为 1 表示捕获电平为高电平，反之则捕获低电平。如果连续多组数据的位 27 都是 1（或 0），说明该高（或低）电平的宽度超过超时值，需多组累计。

下降沿至下降沿、或者上升沿至上升沿触发模式下，可以捕获一个输入变化周期。数据寄存器（R32\_TMR\_FIFO）的有效数据最高位（位 27）为 0 表示正常采样到一个周期，为 1 则表示输入变化周期超过超时值 R32\_TMR\_CNT\_END，需加上后一组数据累计为单个输入变化周期。

捕获周期最短可以选择 8 个时钟周期或 16 个时钟周期，再短则忽略。具体说明如下图所示：

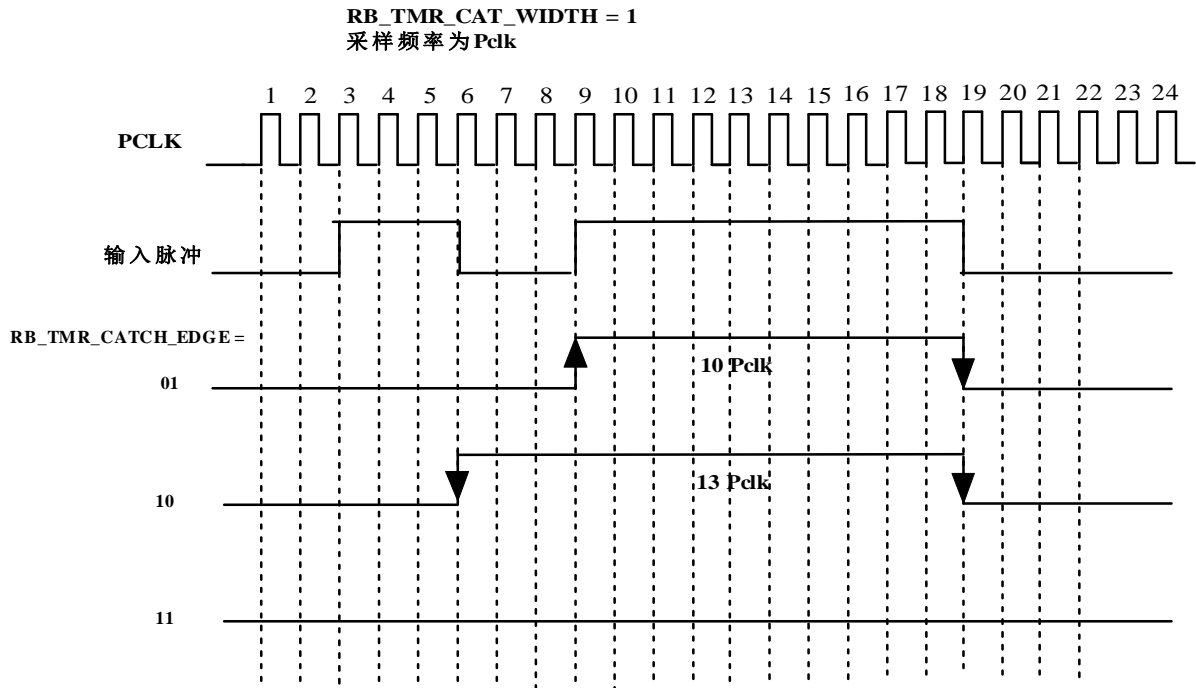


图 9-2 捕获周期为 8 个时钟周期

如上图所示，每个时钟周期内采样一次，

当 RB\_TMR\_CATCH\_EDGE = 2b01 时，设置为边沿触发采样，采样到的时间宽度为 10Pclk；

当 RB\_TMR\_CATCH\_EDGE = 2b10 时，设置为下降沿至下降沿采样，采样到的时间宽度为 13Pclk；

当 RB\_TMR\_CATCH\_EDGE = 2b11 时，设置为上升沿至上升沿采样，采样到的时间宽度为 0(由于输入脉冲的两个上升沿时间为 6 个时钟周期，低于当前设置的最小捕获脉冲宽度)。

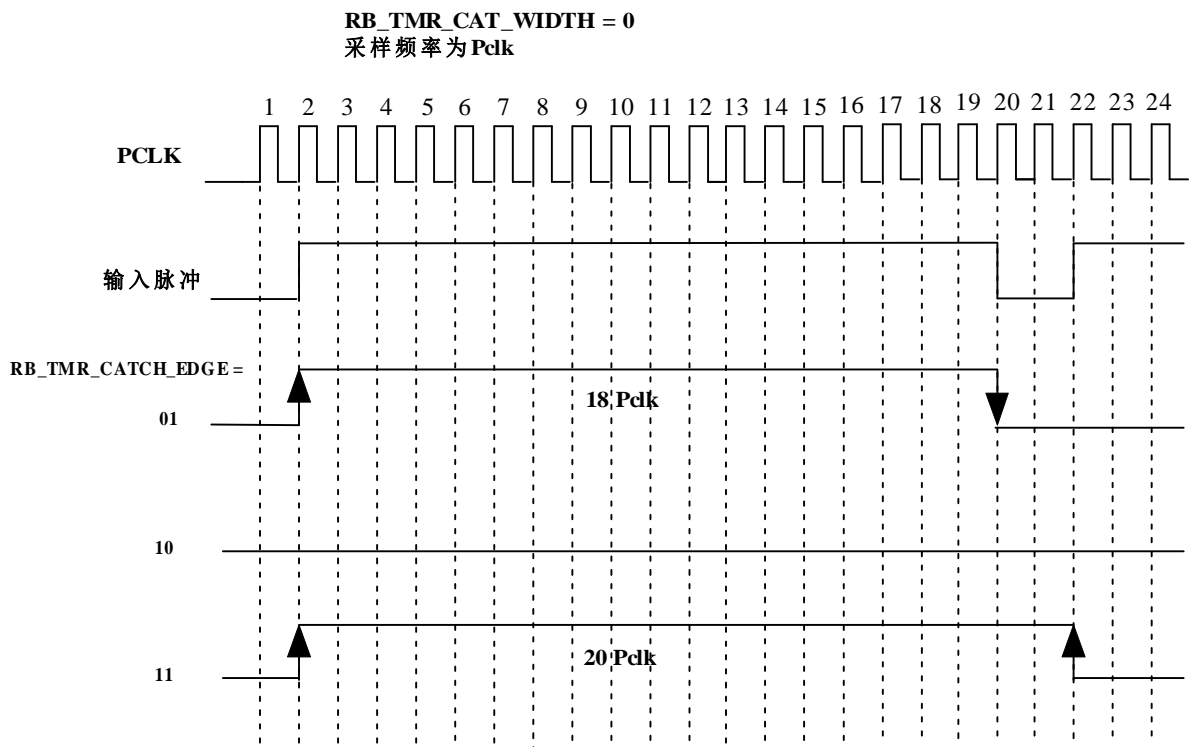


图 9-3 捕获周期为 16 个时钟周期

如上图所示，每个时钟周期内采样一次，

当 RB\_TMR\_CATCH\_EDGE = 2b01 时，设置为边沿触发采样，采样到的时间宽度为 18Pclk；  
当 RB\_TMR\_CATCH\_EDGE = 2b10 时，设置为下降沿至下降沿采样，采样到的时间宽度为 0；  
当 RB\_TMR\_CATCH\_EDGE = 2b11 时，设置为上升沿至上升沿采样，采样到的时间宽度为 20Pclk。

捕获模式操作步骤：

(1)、设置寄存器 R32\_TMR\_CNT\_END 用于设定捕获超时时间，默认最大超时时间为  $2^{28}$  个时钟周期，建议设置合理的超时，避免输入长时间无变化时长时间无数据，如果在最大超时时间内没有检测到电平变化，则 R32\_TMR\_FIFO 寄存器的位 27 置 1；

(2)、将捕获对应的 I/O 引脚方向设置为输入；

(3)、将模式设置寄存器 (R8\_TMRx\_CTRL\_MDD) 的 RB\_TMR\_MODE\_IN 位置 1，同时 RB\_TMR\_ALL\_CLEAR 位清 0，用于清除 FIFO 和 COUNT。如果当前定时器是 TIMB，则将 RB\_TMR3\_MODE\_COUNT 位清 0，如果当前定时器是 TIMD 或 TIMI，则将 RB\_TMR\_MODE\_NRZI 位清 0。根据捕获精度需要，可以将 RB\_TMR\_CAT\_WIDTH 置 1 或清 0，同时设置 RB\_TMR\_CATCH\_EDGE 位选择捕获模式；

(4)、如果需要启用中断，则将寄存器 R8\_INT\_EN\_IRQ\_0 中的 RB\_IE\_IRQ\_TMR 位置 1，启动定时器中断；

(5)、如果需要采用 DMA 方式保存捕获的数据，则需要将寄存器 R16\_TMR\_DMA\_BEG 设置为存储捕获数据缓冲区的首地址，将寄存器 R16\_TMR1\_DMA\_END 设置为存储捕获数据缓冲区的结束地址；

(6)、将寄存器 R8\_TMR\_CTRL\_MDD 的 RB\_TMR\_COUNT\_EN 位置 1，启动捕获功能；

(7)、捕获完成后，寄存器 R8\_INT\_FLAG\_0 中的 RB\_IF\_TMR 位将置 1，同时寄存器 R8\_TMR\_INT\_FLAG 会产生相应的中断状态。默认捕获的数据存放在寄存器 R32\_TMR\_FIFO 中，如果采用 DMA 数据传输，则捕获到的数据自动存放在 DMA 设置的数据缓冲区中。

## 第 10 章 模数转换器 ADC

### 10.1 ADC 简介

CH561 芯片具有 1 个 10 位逐次逼近型模拟数字转换器，具有 DMA 功能。

ADC 特性：

- (1)、10 位分辨率；
- (2)、ADC 输入电压范围： $0 \leq V_{IN} \leq AV_{RT}$ ；
- (3)、采样脉冲宽度可选为 1 个时钟周期或 2 个时钟周期；
- (4)、可选自动采样或手动采样模式；
- (5)、支持设定参考值在到达后触发中断；

### 10.2 ADC 寄存器描述

ADC 相关寄存器物理起始地址为：0x0040 A000

ADC 相关寄存器列表(地址范围：0x00---0x1F)

表 10-1 ADC 相关寄存器列表

名称	偏移地址	描述	复位值
R8_ADC_CTRL_MDD	0x00	ADC 模式控制寄存器	8h00
R8_ADC_CTRL_DMA	0x01	DMA 控制寄存器	8h00
R8_ADC_INTER_EN	0x02	中断使能寄存器	8h00
R8_ADC_CLOCK_DIV	0x03	时钟分频系数寄存器	8h10
R16_ADC_DATA	0x04	转换数据寄存器	8h00
R8_ADC_INT_FLAG	0x06	中断状态寄存器	8h40
R8_ADC_FIFO_COUNT	0x07	FIFO 计数寄存器	8h00
R16_ADC_CMP_VALUE	0x0C	比较模式参考值寄存器	16h0000
R16_ADC_FIFO	0x10	ADC FIFO 寄存器	16h0000
R16_ADC_DMA_NOW	0x14	DMA 当前缓冲区地址寄存器	16h0000
R16_ADC_DMA_BEG	0x18	DMA 开始缓冲区地址寄存器	16h0000
R16_ADC_DMA_END	0x1C	DMA 结束缓冲区地址寄存器	16h0000

ADC 模式控制寄存器(R8\_ADC\_CTRL\_MDD)：

名称	位	访问	描述	复位值
RB_ADC_POWER_ON	7	RW	ADC 功能模块使能/禁止位 0：禁止 1：使能，使能时间需要数微秒	1h0
RB_ADC_SAMPLE_WID	6	RW	ADC 自动采样脉宽设置位 0：1 时钟周期 1：2 时钟周期	1h0
RB_ADC_CHAN_MDD	[5: 4]	RW	ADC 通道选择域 00：选择通道 0，固定设置为 00。	2h0
RB_ADC_CYCLE_CLK	[3: 0]	RW	ADC 时钟采样模式设置位 0：手动采样 其它：自动采样周期的时钟数	4h0

DMA 控制寄存器(R8\_ADC\_CTRL\_DMA)：

名称	位	访问	描述	复位值
----	---	----	----	-----



RB_ADC_MAN_SAMPLE	7	RW	ADC 手动采样控制位 0: 停止采样 1: 开始采样	1h0
RB_ADC_CHAN_OE	6	RW	ADC 通道控制 ADCS 输出使能/禁止位 0: 禁止 1: 使能	1h0
Reserved	[5: 3]	-	保留	-
RB_ADC_DMA_LOOP	2	RW	DMA 地址循环功能使能/禁止位 0: 禁止循环功能 1: 使能循环功能 如果启用 DMA 地址循环功能, 则当上次 DMA 结束后, DMA 地址自动重新装载为寄存器 R16_ADC_DMA_BEG 中的值	1h0
RB_ADC_DMA_BURST	1	RW	DMA 块操作使能/禁止位 0: 禁止 1: 使能	1h0
RB_ADC_DMA_ENABLE	0	RW	DMA 功能使能/禁止位 0: 禁止 DMA 功能 1: 使能 DMA 功能	1h0

中断使能寄存器 (R8\_ADC\_INTER\_EN) :

名称	位	访问	描述	复位值
RB_ADC_CMP_MDD_GT	7	RW	ADC 参考值大小比较使能/禁止位 0: 低于参考值触发 1: 高于参考值触发	1h0
RB_ADC_CMP_MDD_EQ	6	RW	ADC 参考值相等比较使能/禁止位 0: 不包括相等比较, 相等不触发 1: 包含相等比较, 相等也触发	1h0
RB_ADC_IE_DMA_ERR	5	RW	DMA 传输过程错误响应中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_ADC_IE_FIFO_OV	4	RW	FIFO 溢出中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_ADC_IE_DMA_END	3	RW	DMA 结束中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_ADC_IE_FIFO_HF	2	RW	FIFO 接收过半中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_ADC_IE_ADC_END	1	RW	ADC 转换完成中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_ADC_IE_ADC_CMP	0	RW	当前 ADC 比较激活中断使能/禁止位 0: 禁止 1: 使能	1h0

时钟分频系数寄存器(R8\_ADC\_CLOCK\_DIV) :

名称	位	访问	描述	复位值
ADC_CLOCK_DIV	[7: 0]	RW	采样时钟分频系数	1h10

转换数据寄存器(R16\_ADC\_DATA) :

名称	位	访问	描述	复位值
Reserved	[15: 10]	-	保留	-
ADC_DATA	[9: 0]	RO	ADC 采集数据内容, 默认是右对齐, 采样电压计算方式为: $V_{in} = (ADC\_DATA/1024) * AVRT$	10h0

中断状态寄存器(R8\_ADC\_INT\_FLAG) :

名称	位	访问	描述	复位值
RB_ADC_CHAN_INDEX	7	RO	自动转换模式下 ADC 当前通道 0: 通道 0 1: 通道 1	1h0
RB_ADC_EOC_FLAG	6	RO	ADC 转换结束状态位 0: 未结束 1: 已结束	1h1
RB_ADC_IF_DMA_ERR	5	RWI	DMA 传输错误中断标志位 0: 无错误 1: 发生错误	1h0
RB_ADC_IF_FIFO_OV	4	RWI	FIFO 溢出中断标志位 0: 未溢出 1: 已溢出	1h0
RB_ADC_IF_DMA_END	3	RWI	DMA 结束标志位 0: 未结束 1: 已结束	1h0
RB_ADC_IF_FIFO_HF	2	RWI	FIFO 接收过半中断标志位 0: ADC 转换数据未达到 FIFO 一半 1: : ADC 转换数据大于或等于 FIFO 一半	1h0
RB_ADC_IF_ADC_END	1	RWI	ADC 转换结束标志位 0: ADC 转换未结束 1: ADC 转换结束	1h0
RB_ADC_IF_ADC_CMP	0	RWI	ADC 当前值与参考值比较激活中断标志位 0: 未激活 (未达到触发条件) 1: 已激活 (已达到触发条件)	1h0

FIFO 计数寄存器(R8\_ADC\_FIFO\_COUNT) :

名称	位	访问	描述	复位值
ADC_FIFO_COUNT	[7: 0]	RO	FIFO 里面已存储的 ADC 转换数据个数	8h0

比较模式参考值寄存器(R16\_ADC\_CMP\_VALUE) :

名称	位	访问	描述	复位值
Reserved	[15: 10]	-	保留	-

ADC_CMP_VALUE	[9: 0]	RW	比较模式下参与比较的参考值	10h0
---------------	--------	----	---------------	------

ADC FIFO 寄存器(R16\_ADC\_FIFO):

名称	位	访问	描述	复位值
Reserved	[15: 10]	-	保留	-
ADC_FIFO	[9: 0]	RO	ADC 当前转换值	10h0

DMA 当前缓冲区地址寄存器(R16\_ADC\_DMA\_NOW):

名称	位	访问	描述	复位值
ADC_DMA_NOW	[15: 0]	RW	此寄存器表示 DMA 已运行的当前地址, 可用于计算已转换的次数, 计算方法为: $COUNT = (ADC\_DMA\_NOW - ADC\_DMA\_BEG) / 2$	16h0

DMA 开始缓冲区地址寄存器(R16\_ADC\_DMA\_BEG):

名称	位	访问	描述	复位值
ADC_DMA_BEG	[15: 0]	RW	ADC 转换数据 DMA 存储开始地址, 该地址必须 2 字节对齐	16h0

DMA 结束缓冲区地址寄存器(R16\_ADC\_DMA\_END):

名称	位	访问	描述	复位值
ADC_DMA_END	[15: 0]	RW	ADC 转换数据 DMA 存储结束地址, 该地址必须 2 字节对齐	16h0

### 10.3 ADC 功能

ADC 采样模式配置步骤:

- (1)、设置 ADC 模式控制寄存器(R8\_ADC\_CTRL\_MDD)中的 RB\_ADC\_POWER\_ON 位为 1, 使能 ADC 模块;
- (2)、设置采样时钟分频系数寄存器(R8\_ADC\_CLOCK\_DIV), 选择相应的时钟频率, 时钟频率最大为 12MHZ, 最小为 1MHZ;
- (3)、设置 ADC 模式控制寄存器(R8\_ADC\_CTRL\_MDD)中的 RB\_ADC\_CYCLE\_CLK 和 RB\_ADC\_SAMPLE\_WID, 如果 RB\_ADC\_CYCLE\_CLK 设置为 0, 则为手动采样模式, 如果 RB\_ADC\_CYCLE\_CLK 设置为非 0 值, 则为自动采样模式, 此时 RB\_ADC\_CYCLE\_CLK 为自动采样周期, 其值必须大于 11;
- (4)、设置 ADC 模式控制寄存器(R8\_ADC\_CTRL\_MDD)中的 RB\_ADC\_CHAN\_MDD 域值, 选择 ADC 采集的通道;
- (5)、建议延时 100 微秒以上, 等待 ADC 及 ADC 时钟正常工作。如果当前是手动采样模式, 则需要再设置 RB\_ADC\_MAN\_SAMPLE 位为 1, 延时至少 150 纳秒后再设置为 0, 以产生手工采样脉冲;
- (6)、如果需要使用中断, 则设置 ADC 中断使能寄存器(R8\_ADC\_INTER\_EN)相应位;
- (7)、读 FIFO 计数寄存器(R8\_ADC\_FIFO\_COUNT)得到 FIFO 中数据个数, 然后清除 ADC FIFO 寄存器(R16\_ADC\_FIFO)里面的数据, 特别是对于循环采样, 需清除上一次的数据;
- (8)、如果需要使用 DMA, 则设置 DMA 开始缓冲区地址寄存器(R16\_ADC\_DMA\_BEG)为用户定义的数据存放缓冲区的起始地址值, DMA 结束缓冲区地址寄存器(R16\_ADC\_DMA\_END) 为用户定义的数据存放缓冲区的结束地址值, 并设置 ADC 模式控制寄存器(R8\_ADC\_CTRL\_DMA)中的 RB\_ADC\_DMA\_ENABLE 位为 1, 开启 DMA 功能;
- (9)、等待中断状态标志寄存器(ADC\_INT\_FLAG)中 RB\_ADC\_IF\_DMA\_END 位置 1, 表示 ADC 数据转换结束, 从转换数据寄存器(R16\_ADC\_DATA)中读取 ADC 采集的数据, 如果使能 DMA 功能, 则从之前指定的 DMA 缓冲区中读取数据;

对于自动采样模式, 建议丢弃首个 ADC 结果数据。

**比较模式：**

ADC 模式还支持比较模式，初始化时，除了前面的初始化外，还需要设置比较模式参考值寄存器 (R16\_ADC\_CMP\_VALUE) 值，设置中断使能寄存器 (ADC\_INTER\_EN) 中的 RB\_ADC\_CMP\_MOD\_GT 位选择低于比较值触发还是高于比较值触发，以及 RB\_ADC\_CMP\_MOD\_EQ 位选择是否包含相等比较。

## 第 11 章 中断控制器

### 11.1 中断控制器简介

CH561 芯片支持多个中断源，这些中断源可以是芯片内部的中断源，也可以是芯片外部的中断源。包括 UART0、UART1、SPI0、SPI1、ADC、TIM0、TIM1、TIM2、TIMB、以太网、被动并口和 GPIO 等。

CH561 芯片有两种中断模式，分别是 IRQ 中断和 FIQ 中断。FIQ 中断的优先级较高，IRQ 中断优先级较低。所有中断源均支持 IRQ 和 FIQ 中断模式。每种中断模式还再分为查询入口中断和直跳入口中断，默认都是查询入口中断。其中，中断源 TIM0、SPI0 和 PB 端口，分别具有一个 32 位独立的直跳入口中断向量寄存器，支持直跳入口中断。

特性：

- (1)、支持多个中断源；
- (2)、支持 IRQ 中断和 FIQ 中断两种中断模式；
- (3)、所有中断源均支持 IRQ 和 FIQ 中断模式；
- (4)、部分中断源支持直跳入口中断；

### 11.2 寄存器描述

中断相关寄存器物理起始地址为：0x0040 0000

中断相关寄存器列表(地址范围：0x20---0x4F)

表 11-1 中断相关寄存器列表

名称	偏移地址	描述	复位值
R8_INT_VEC_IRQ	0x20	IRQ 中断直跳入口控制寄存器	8h00
R8_INT_VEC_FIQ	0x21	FIQ 中断直跳入口控制寄存器	8h00
R32_INT_VEC_TMRO	0x24	TMRO 直跳入口中断向量寄存器	32h0000 0000
R32_INT_VEC_SPI0	0x28	SPI0 直跳入口中断向量寄存器	32h0000 0000
R32_INT_VEC_PB	0x2C	PB 端口直跳入口中断向量寄存器	32h0000 0000
R8_INT_FLAG_0	0x40	中断标志低 8 位寄存器	8h00
R8_INT_FLAG_1	0x41	中断标志高 8 位寄存器	8h00
R8_INT_EN_IRQ_0	0x48	IRQ 中断使能低 8 位寄存器	8h00
R8_INT_EN_IRQ_1	0x49	IRQ 中断使能高 8 位寄存器	8h00
R8_INT_EN_IRQ_GLOB	0x4B	IRQ 全局中断使能寄存器	8h00
R8_INT_EN_FIQ_0	0x4C	FIQ 中断使能低 8 位寄存器	8h00
R8_INT_EN_FIQ_1	0x4D	FIQ 中断使能高 8 位寄存器	8h00
R8_INT_EN_FIQ_GLOB	0x4F	FIQ 全局中断使能寄存器	8h00

IRQ 普通中断直跳入口控制寄存器(R8\_INT\_VEC\_IRQ)：

名称	位	访问	描述	复位值
Reserved	[7: 3]	-	保留	-
RB_IV_IRQ_PB	2	RWA	GPIO PB 端口 IRQ 中断直跳入口使能/禁止位 0: 禁止 1: 使能	1h0
RB_IV_IRQ_SPI0	1	RWA	SPI0 IRQ 中断直跳入口使能/禁止位 0: 禁止 1: 使能	1h0
RB_IV_IRQ_TMRO	0	RWA	TMRO IRQ 中断直跳入口使能/禁止位	1h0

			0: 禁止 1: 使能	
--	--	--	----------------	--

FIQ 快速中断直跳入口控制寄存器 (R8\_INT\_VEC\_FIQ):

名称	位	访问	描述	复位值
Reserved	[7: 3]	-	保留	-
RB_IV_FIQ_PB	2	RWA	GPIO PB 端口 FIQ 中断直跳入口使能/禁止位 0: 禁止 1: 使能	1h0
RB_IV_FIQ_SPI0	1	RWA	SPI0 FIQ 中断直跳入口使能/禁止位 0: 禁止 1: 使能	1h0
RB_IV_FIQ_TMRO	0	RWA	TMRO FIQ 中断直跳入口使能/禁止位 0: 禁止 1: 使能	1h0

TMRO 直跳入口中断向量寄存器 (R32\_INT\_VEC\_TMRO):

名称	位	访问	描述	复位值
R32_INT_VEC_TMRO	[31: 0]	RWA	TMRO 直跳入口中断向量	32h0

SPI0 直跳入口中断向量寄存器 (R32\_INT\_VEC\_SPI0):

名称	位	访问	描述	复位值
R32_INT_VEC_SPI0	[31: 0]	RWA	SPI0 直跳入口中断向量	32h0

GPIO PB 端口直跳入口中断向量寄存器 (R32\_INT\_VEC\_PB):

名称	位	访问	描述	复位值
R32_INT_VEC_PB	[31: 0]	RWA	GPIO PB 端口直跳入口中断向量	32h0

中断标志低 8 位寄存器 (R8\_INT\_FLAG\_0):

名称	位	访问	描述	复位值
RB_IF_UART1	7	RO	串口 1 中断标志位 0: 串口 1 无中断产生 1: 串口 1 有中断产生	1h0
RB_IF_UART0	6	RO	串口 0 中断标志位 0: 串口 0 无中断产生 1: 串口 0 有中断产生	1h0
RB_IF_SPI1	5	RO	SPI1 中断标志位 0: SPI1 无中断产生 1: SPI1 有中断产生	1h0
RB_IF_SPI0	4	RO	SPI0 中断标志位 0: SPI0 无中断产生 1: SPI0 有中断产生	1h0
RB_IF_TMR3	3	RO	定时器 3 中断标志位 0: 定时器 3 无中断产生 1: 定时器 3 有中断产生	1h0
RB_IF_TMR2	2	RO	定时器 2 中断标志位 0: 定时器 2 无中断产生	1h0

			1: 定时器 2 有中断产生	
RB_IF_TMR1	1	RO	<b>定时器 1 中断标志位</b> 0: 定时器 1 无中断产生 1: 定时器 1 有中断产生	1h0
RB_IF_TMR0	0	RO	<b>定时器 0 中断标志位</b> 0: 定时器 0 无中断产生 1: 定时器 0 有中断产生	1h0

中断标志高 8 位寄存器 (R8\_INT\_FLAG\_1):

名称	位	访问	描述	复位值
RB_IF_ETH	7	RO	<b>以太网中断标志位</b> 0: 以太网无中断产生 1: 以太网有中断产生	1h0
Reserved	6	RO	<b>保留</b>	1h0
RB_IF_SLV	5	RO	<b>被动并口中断标志位</b> 0: 被动并口无中断产生 1: 被动并口有产生中断	1h0
RB_IF_ADC	4	RO	<b>ADC 采集中断标志位</b> 0: ADC 采集无中断产生 1: ADC 采集有产生中断	1h0
RB_IF_PD	3	RO	<b>PD 口外部中断标志位</b> 0: PD 口无外部中断产生 1: PD 口有产生外部中断	1h0
RB_INT_WOL_STATUS	2	RO	<b>当前以太网 WOL 远程唤醒信号状态</b> 0: 以太网模块未请求唤醒 (WOL=0) 1: 以太网模块请求唤醒 (WOL=1)	1h0
RB_IF_PB	1	RO	<b>PB 口外部中断标志位</b> 0: PB 口无外部中断产生 1: PB 口有外部中断产生	1h0
RB_IF_PA	0	RO	<b>PA 口外部中断标志位</b> 0: PA 口无外部中断产生 1: PA 口有外部中断产生	1h0

IRQ 中断使能低 8 位寄存器 (R8\_INT\_EN\_IRQ\_0):

名称	位	访问	描述	复位值
RB_IE_IRQ_UART1	7	RW	<b>串口 1 IRQ 中断使能/禁止位</b> 0: 禁止 1: 使能	1h0
RB_IE_IRQ_UART0	6	RW	<b>串口 0 IRQ 中断使能/禁止位</b> 0: 禁止 1: 使能	1h0
RB_IE_IRQ_SPI1	5	RW	<b>SPI1 IRQ 中断使能/禁止位</b> 0: 禁止 1: 使能	1h0
RB_IE_IRQ_SPI0	4	RW	<b>SPI0 IRQ 中断使能/禁止位</b> 0: 禁止 1: 使能	1h0
RB_IE_IRQ_TMR3	3	RW	<b>定时器 3 IRQ 中断使能/禁止位</b>	1h0

			0: 禁止 1: 使能	
RB_IE_IRQ_TMR2	2	RW	定时器 2 IRQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_IRQ_TMR1	1	RW	定时器 1 IRQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_IRQ_TMRO	0	RW	定时器 0 IRQ 中断使能/禁止位 0: 禁止 1: 使能	1h0

IRQ 中断使能高 8 位寄存器 (R8\_INT\_EN\_IRQ\_1):

名称	位	访问	描述	复位值
RB_IE_IRQ_ETH	7	RW	以太网 IRQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
Reserved	6	RW	保留	1h0
RB_IE_IRQ_SLV	5	RW	被动并口 IRQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_IRQ_ADC	4	RW	ADC IRQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_IRQ_PD	3	RW	PD 端口 IRQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
Reserved	2	-	保留	-
RB_IE_IRQ_PB	1	RW	PB 端口 IRQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_IRQ_PA	0	RW	PA 端口 IRQ 中断使能/禁止位 0: 禁止 1: 使能	1h0

IRQ 全局中断使能寄存器 (R8\_INT\_EN\_IRQ\_GLOB):

名称	位	访问	描述	复位值
RB_IE_IRQ_GLOB	7	RW	IRQ 全局中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_IRQ_WOL	6	RW	以太网唤醒请求中断使能/禁止位 0: 禁止 1: 使能	1h0
Reserved	[5: 0]	-	保留	-

FIQ 中断使能低 8 位寄存器 (R8\_INT\_EN\_FIQ\_0):

名称	位	访问	描述	复位值
RB_IE_FIQ_UART1	7	RW	串口 1 FIQ 中断使能/禁止位	1h0



			0: 禁止 1: 使能	
RB_IE_FIQ_UART0	6	RW	串口 0 FIQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_FIQ_SPI1	5	RW	SPI1 FIQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_FIQ_SPI0	4	RW	SPI0 FIQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_FIQ_TMR3	3	RW	定时器 3 FIQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_FIQ_TMR2	2	RW	定时器 2 FIQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_FIQ_TMR1	1	RW	定时器 1 FIQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_FIQ_TMR0	0	RW	定时器 0 FIQ 中断使能/禁止位 0: 禁止 1: 使能	1h0

FIQ 中断使能高 8 位寄存器 (R8\_INT\_EN\_FIQ\_1):

名称	位	访问	描述	复位值
RB_IE_FIQ_ETH	7	RW	以太网 FIQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
Reserved	6	RW	保留	1h0
RB_IE_FIQ_SLV	5	RW	被动并口 FIQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_FIQ_ADC	4	RW	ADC FIQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_FIQ_PD	3	RW	PD 端口 FIQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
Reserved	2	-	保留	-
RB_IE_FIQ_PB	1	RW	PB 端口 FIQ 中断使能/禁止位 0: 禁止 1: 使能	1h0
RB_IE_FIQ_PA	0	RW	PA 端口 FIQ 中断使能/禁止位 0: 禁止 1: 使能	1h0

FIQ 全局中断使能寄存器 (R8\_INT\_EN\_FIQ\_GLOB):

名称	位	访问	描述	复位值
RB_IE_FIQ_GLOB	7	RW	FIQ 全局中断使能/禁止位 0: 禁止 1: 使能	1h0
Reserved	[6: 0]	-	保留	-

### 11.3 中断使用说明

使用CH561芯片的中断功能，除了需要配置对应的上述寄存器之外，还需要配置外设对应的中断使能寄存器。例如，TIMD中断配置，除了将RB\_IE\_IRQ\_TMRO(或者RB\_IE\_FIQ\_TMRO)置位，同时全局中断使能位RB\_IE\_IRQ\_GLOB(RB\_IE\_FIQ\_GLOB)置位外，还需要提前对TIMD的中断使能寄存器进行设置。中断配置完毕后，如果有对应的中断产生，相应的中断标志位置1，并进入中断服务程序，进行相应的处理，例如解决中断原因，清除中断标志。

查询入口中断是指在进入IRQ或者FIQ中断后，执行一些指令查询中断标志寄存器分析是哪个中断源产生的中断，然后跳转到相应的中断服务处理程序。除非只使能了个别中断源，或者是明知哪个中断源，否则查询过程和二次跳转过程会浪费少量指令周期。

直跳入口中断是指在发生相应的中断时，根据直跳入口中断向量寄存器的预置值，直接跳转到相应的中断服务处理程序，从而节约查询和跳转时间。除此之外，与查询入口中断没有区别。

FIQ中断配置主要步骤：

(1)、初始化外设，配置外设对应的中断使能寄存器；

(2)、配置FIQ中断使能寄存器(R8\_INT\_EN\_FIQ\_0或R8\_INT\_EN\_FIQ\_1)对应位，使能该外设的FIQ中断；

(3)、配置FIQ全局中断使能寄存器(R8\_INT\_EN\_FIQ\_GLOB)。

以下程序将被动并口配置为FIQ中断：

```
R8_SAFE_ACCESS_SIG = 0x57;           // 解锁步骤1
R8_SAFE_ACCESS_SIG = 0xA8;           // 解锁步骤2
R8_SLV_CONFIG = ( RB_SLV_ENABLE | RB_SLV_IE_CMD ); // 使能被动并口及命令写入中断
R8_SAFE_ACCESS_SIG = 0x00 ;

R8_INT_EN_FIQ_1 |= RB_IE_FIQ_SLV;    // 使能FIQ中的被动并口中断
R8_INT_EN_FIQ_GLOB |= RB_IE_FIQ_GLOB; // 全局中断使能
```

IRQ中断配置主要步骤：

(1)、初始化外设，配置外设对应的中断使能寄存器；

(2)、配置IRQ中断使能寄存器(R8\_INT\_EN\_IRQ\_0或R8\_INT\_EN\_IRQ\_1)对应位，使能该外设的IRQ中断；

(3)、配置IRQ全局中断使能寄存器(R8\_INT\_EN\_IRQ\_GLOB)。

以下程序将UART1配置为IRQ中断：

```
R8_UART1_IER |= RB_IER_RECV_RDY;    // 使能UART1接收中断
R8_UART1_MCR |= RB_MCR_OUT2;        // 允许串口中断请求输出

R8_INT_EN_IRQ_0 |= RB_IE_IRQ_UART1; // 使能IRQ中的UART1中断
R8_INT_EN_IRQ_GLOB |= RB_IE_IRQ_GLOB; // 全局中断使能
```

直跳入口中断配置主要步骤：

(1)、初始化外设，配置外设对应的中断使能寄存器；

(2)、配置TMRO、SPIO或PB端口对应的独立的直跳入口中断向量寄存器，该寄存器通过调用不同的宏进行计算，可以配置为FIQ中断或IRQ中断；

(3)、设置直跳入口中断向量控制寄存器(R8\_INT\_VEC\_FIQ或R8\_INT\_VEC\_IRQ)，重新映射该外设

中断;

(4)、配置IRQ中断使能寄存器(R8\_INT\_EN\_IRQ\_0或R8\_INT\_EN\_IRQ\_1)对应位,使能该外设的IRQ中断;或者配置FIQ中断使能寄存器(R8\_INT\_EN\_FIQ\_0或R8\_INT\_EN\_FIQ\_1)对应位,使能该外设的FIQ中断;

(5)、配置IRQ全局中断使能寄存器(R8\_INT\_EN\_IRQ\_GLOB),或者配置FIQ全局中断使能寄存器(R8\_INT\_EN\_FIQ\_GLOB)。

以下程序将SPI0配置为特殊快速中断:

```
R8_SAFE_ACCESS_SIG = 0x57;           // 解锁步骤1
R8_SAFE_ACCESS_SIG = 0xA8;           // 解锁步骤2
R32_INT_VEC_SPI0 = MAKE_INT_VEC_FIQ( SPI0_New_ISR ); // 设置SPI0中断向量
R8_INT_VEC_FIQ |= RB_IV_FIQ_SPI0;    // 重新映射FIQ
R8_SAFE_ACCESS_SIG = 0x00;

R8_INT_EN_FIQ_0 |= RB_IE_FIQ_SPI0;   // 使能FIQ中的SPI0中断
R8_INT_EN_FIQ_GLOB |= RB_IE_FIQ_GLOB; // 全局中断使能
```

## 第 12 章 以太网控制器 (ETH)

### 12.1 概述

CH561 以太网控制器集成 MAC 和 PHY。

CH561 的 MAC 是一个具有 DMA 功能的高性能的 10/100M 以太网控制器，包含 DMA、FIFO、MAC 和 PHY。它完全兼容 IEEE802.3 协议，内部 DMA 控制器处理 FIFO 和系统内存之间的所有的数据传输。CH561 的 MAC 内部包含 4KB 的缓冲区 (TX FIFO 和 RX FIFO 各占 2KB)，用于缓冲数据。MAC 和 PHY 支持两种速度 10Mbps 和 100Mbps。

CH561 的 MAC 提供 LAN 唤醒 (Wake-on-LAN) 功能，支持三种唤醒事件：连接状态改变，魔法包以及唤醒帧。CH561 可以通过此功能进行远程唤醒。

CH561 内置了 IEEE 分配的唯一 MAC 地址，网络库里提供函数读取接口。

主要特性：

- (1)、支持 DMA，用于发送和接收数据包；
- (2)、LAN 唤醒 (Wake-on-LAN) 功能，支持三种唤醒事件：连接状态改变，魔法包以及唤醒帧；
- (3)、独立的接收和发送缓冲区，每个缓冲区大小为 2KB；
- (4)、与 802.3x 全双工流控和半双工背压流控完全兼容；
- (5)、支持全双工，半双工和回环模式；
- (6)、以帧为单位自动计算 CRC 和产生帧填充 (PAD)；
- (7)、支持多种模式的地址过滤；
- (8)、支持自动协商机制；
- (9)、支持 MDI/MDIX 线路自动转换 (Auto-MDIX)；
- (10)、支持 10Mbps、100Mbps；
- (11)、支持全双工、半双工以及回环模式；
- (12)、完全兼容 10BASE-TX 和 10BASE-T；
- (13)、芯片内置由 IEEE 分配的唯一 MAC 地址；

### 12.2 功能模块

#### 12.2.1 DMA 仲裁器 (DMA\_ARBITER)

DMA\_ARBITER 将决定 TXDMA 和 RXDMA 两者之间谁拥有对总线的使用权。

#### 12.2.2 TXDMA

TXDMA 主要功能如下：

- (1)、读取发送描述符，并将发送状态回写到发送描述符中；
- (2)、将发送缓冲区的数据包写到 TX FIFO 中；
- (3)、控制 TX FIFO 的读写动作；

当 TXDMA 需要将发送包发送到以太网上，它首先取出发送描述符的相关信息以及发送缓冲区基地址和大小，然后将数据写到 TX FIFO 中并请求 TXMAC 读取发送包，一旦数据包发送完毕，TXMAC 会将发送状态传递给 TXDMA，TXDMA 再将这个状态写到相应的发送描述符中。

#### 12.2.3 RXDMA

RXDMA 主要功能如下：

- (1)、读取接收描述符，并将接收状态回写到接收描述符中；
- (2)、将 RX FIFO 的数据写接收缓冲区中；
- (3)、控制 RX FIFO 的读写动作；

当 MAC 收到一包数据时，它首先将数据保存在 RX FIFO 中，然后 RXDMA 将会把这包数据写到接收缓冲区中，并将接收状态写到相应的接收描述符中。

#### 12.2.4 TXMAC

TXMAC 主要用于将 TX FIFO 的数据包发送到以太网上。功能模块包括 CRC 检验、前导包、Jam 发生器以及传输状态机。

当 TXMAC 需要发送数据时，首先检测以太网是否为空闲状态，只有在空闲状态下才会启动发送，否则一直等待。TXMAC 将自动添加前导包和 CRC 校验。如果 TXMAC 在传输中检测到冲突，它将会发送一次 Jam 信号并判断本次冲突是否已经超过 16 次，如果不是则等待补偿时间后再发送。补偿时间是一个随机时间，随机时间单位为 512 位，发送流程如下图所示：

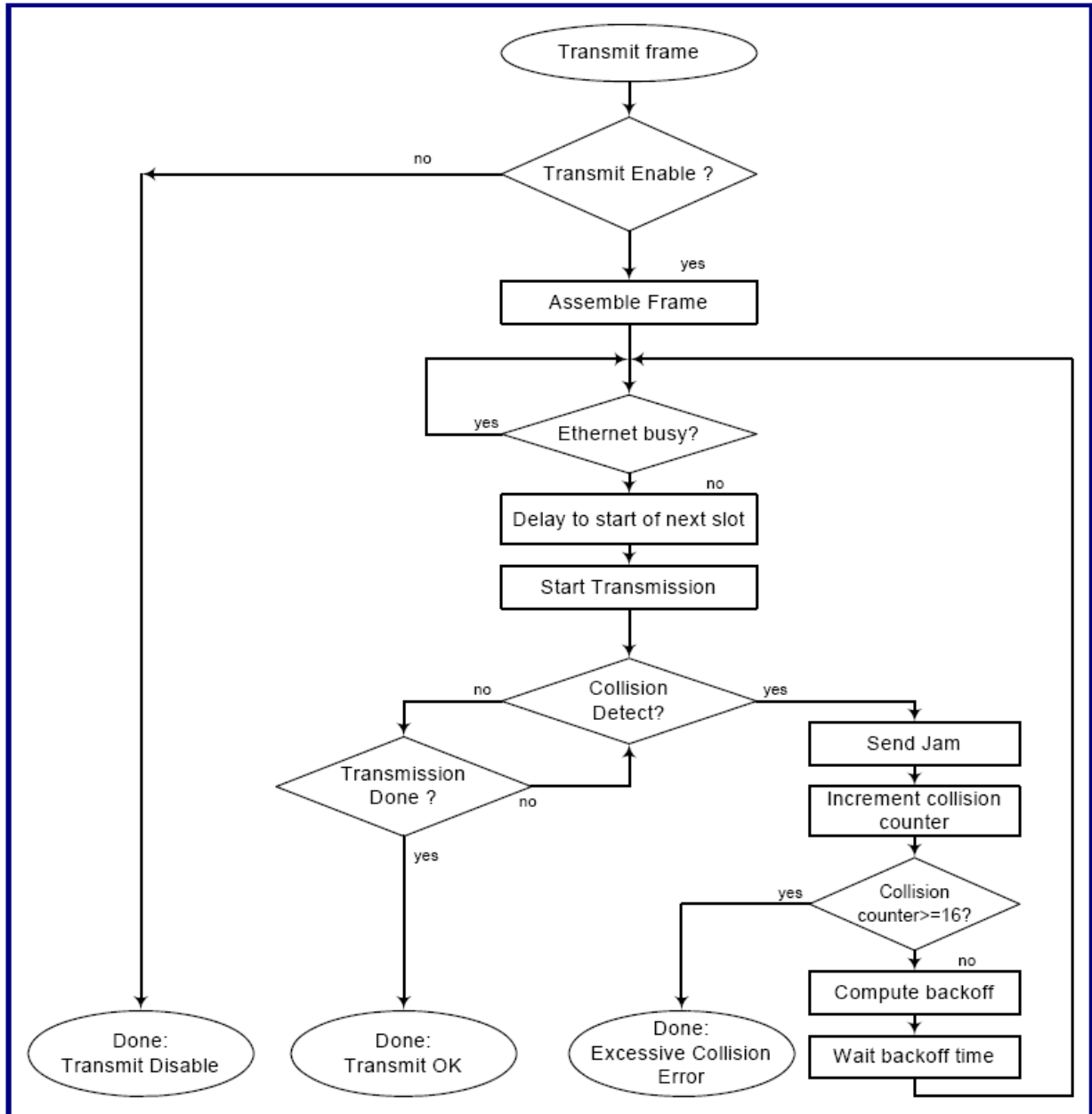


图 15-1 发送流程图

#### 12.2.5 RXMAC

RXMAC 的主要功能是将以太网数据接收到 RX FIFO 中，功能模块主要包括地址识别、CRC 检测以及接收状态机。

当以太网上有数据包进入时，RXDMA 将数据包从 RXMAC 接收到 RX FIFO 中，如果地址过滤通过并且 CRC 检测正确，RXDMA 将数据包保存在 RX FIFO 中，否则将丢弃此数据包，接收流程如下图所示：

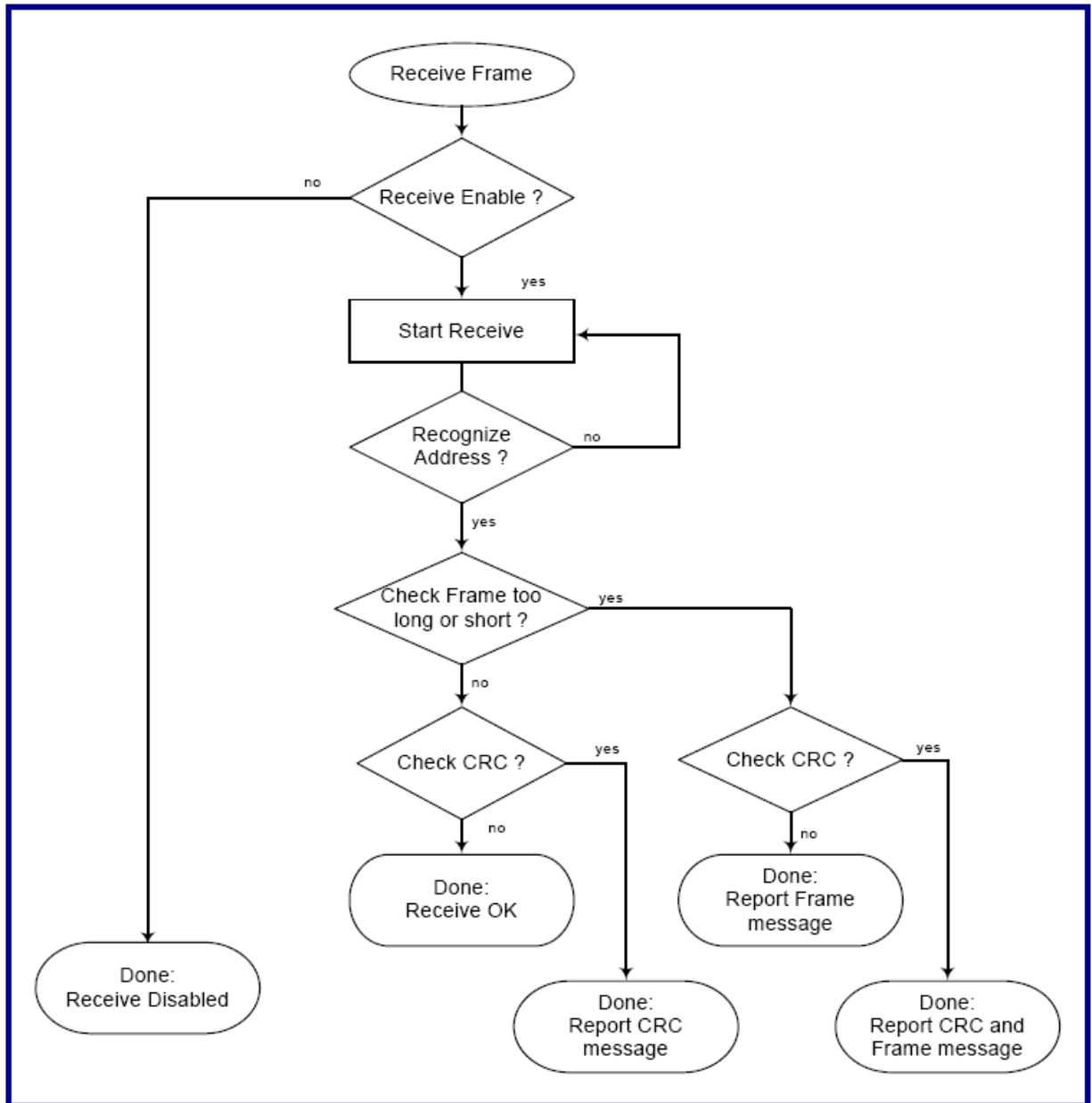


图 15-2 接收流程图

### 12.2.6 寄存器(REGISTER)

寄存器保存了 MAC 的相关配置信息，后续章节将进行详细讲解。

## 12.3 功能描述

### 12.3.1 发送描述符和数据缓冲区

MAC通过发送描述符队列对发送缓冲区进行管理，发送描述符和发送缓冲区均位于系统内存中。MAC将发送缓冲区的数据写到其内部的TX FIFO中，然后再发送到以太网上。发送缓冲区的指针保存在发送描述符中。

发送描述符队列的基地址保存在寄存器R32\_ETHE\_TXR\_BADR中。每一个发送描述符中均包含一个发送缓冲区，一个发送缓冲区可以组成一个完整的帧或者一个帧的一部分，但是一个发送缓冲区只能在

同一个帧中发送。发送描述符包含了发送缓冲区状态，但是发送缓冲区中只能是需要发送的数据。下图为发送描述符结构图。

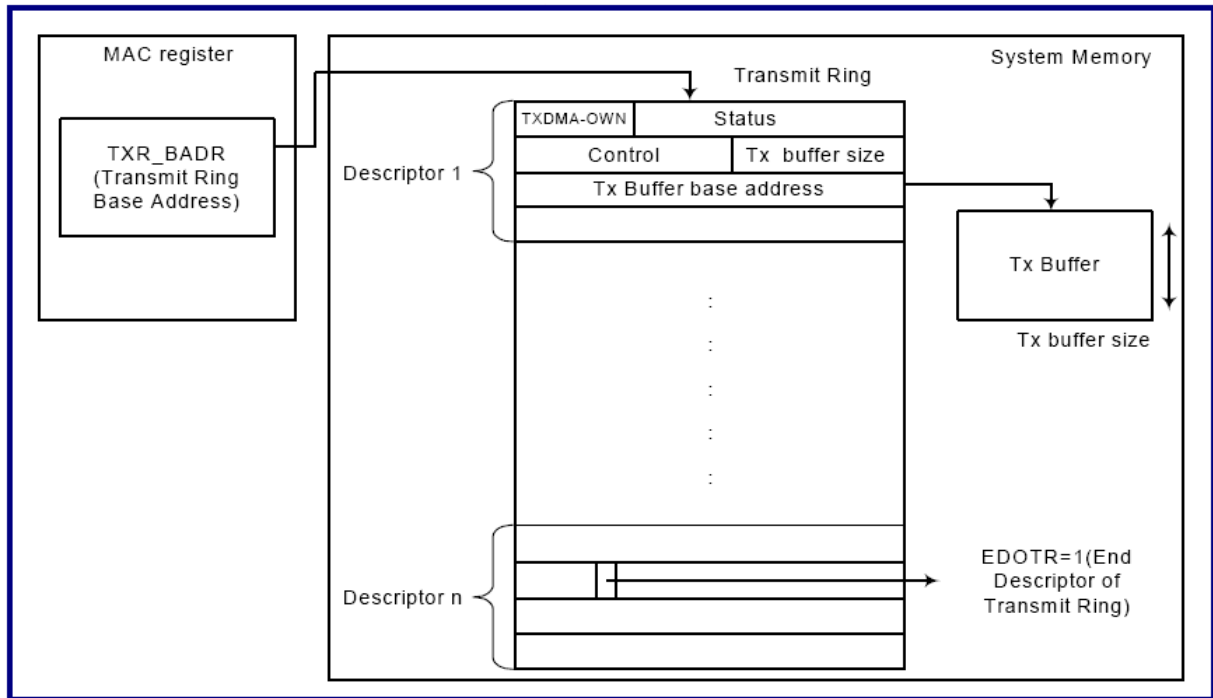


图15-3 发送描述符结构图

发送描述符的结构如下：

TXDES0	TXDMA_OWN	Status
TXDES1	Control	TX buffer size
TXDES2	TX buffer base address	发送缓冲区基地址，4字节对齐

注意：

- (1)、每个描述符的地址必须 16 字节对齐；
- (2)、软件必须以字方式对描述符进行访问(大端模式下)；
- (3)、局限性：发送包的最大长度为 2028 字节。

每个发送描述符包含三个元素，分别为：TXDES0、TXDES1 和 TXDES2。

TXDES0：包含了帧状态和描述符权限

名称	位	描述
TXDMA_OWN	31	<b>TXDMA 使用权限位</b> 0：软件拥有本描述符使用权限 1：MAC 拥有本描述符使用权限 帧传输完成时，MAC 会自动清除此位
Reserved	[30: 2]	保留
TXPKT_EXSCOL	1	0：产生的帧冲突未超过 16 次 1：产生的帧冲突超过 16 次，此传输被终止 仅当 FTS = 1 时，该位有效
TXPKT_LATECOL	0	该位置 1 表示最后一次冲突导致发送终止， 仅当 FTS = 1 时，该位有效

TXDES1：控制位与发送缓冲区大小

名称	位	描述
EDOTR	31	此发送描述符是否为描述符队列的最后一个描述符 0：否

		1: 是
TXIC	30	<b>帧传输是否完成标志位</b> 0: 帧传输未完成 1: 帧传输完成, MAC 产生中断 仅当 FTS = 1, 且 R32_ETHE_ITC 的第 8 位和第 14 位 (TXINT_HTR 和 TXINT_CNT) 为 0 时, 该位有效
TX2FIC	29	<b>数据写成功中断使能位</b> 0: 数据成功写到 TXFIFO 后不产生中断; 1: 数据成功写到 TXFIFO 后产生中断; 仅当 FTS = 1 时, 该位有效
FTS	28	该位置 1 表示当前描述符是发送包的第一个数据段的描述符, 仅当一个数据包分多个段进行描述时使用
LTS	27	该位置 1 表示当前描述符是发送包的最后一个数据段的描述符
Reserved	[26: 11]	保留
TXBUF_SIZE	[10: 0]	发送缓冲区大小 0: 此描述无效

TXDES2: 发送缓冲区的基地址

名称	位	描述
TXBUF_BADR	[31: 0]	发送缓冲区的基地址

### 12.3.2 接收描述符和数据缓冲区

MAC通过接收描述符来对接收缓冲区进行管理, 接收描述符和接收缓冲区均在系统内存中。MAC首先将以太网的数据保存在RX FIFO中, 然后再把数据从RX FIFO中保存到位于系统内存区域的接收缓冲区中。不管接收描述符中的接收缓冲区地址是多少, DMA通道都将把第一个数据段放到2字节对齐的地方, 如果接收到的数据段多于1个(Packetsize >receive buffer size), 后续的数据段将被放在4字节对齐的地址上。系统根据接收描述符中的FRS来判断有效数据的起始地址。

接收描述符队列的基地址保存在寄存器RXR\_BADR中, 每一个接收描述符中都包含一个接收缓冲区地址。一个接收缓冲区由一帧或者一帧的一部分构成, 但是不会超过一个帧。接收描述符包含了接收缓冲区地址以及接收状态, 但是不包含接收数据本身, 数据本身保存在接收缓冲区中。下图为接收描述符结构图。



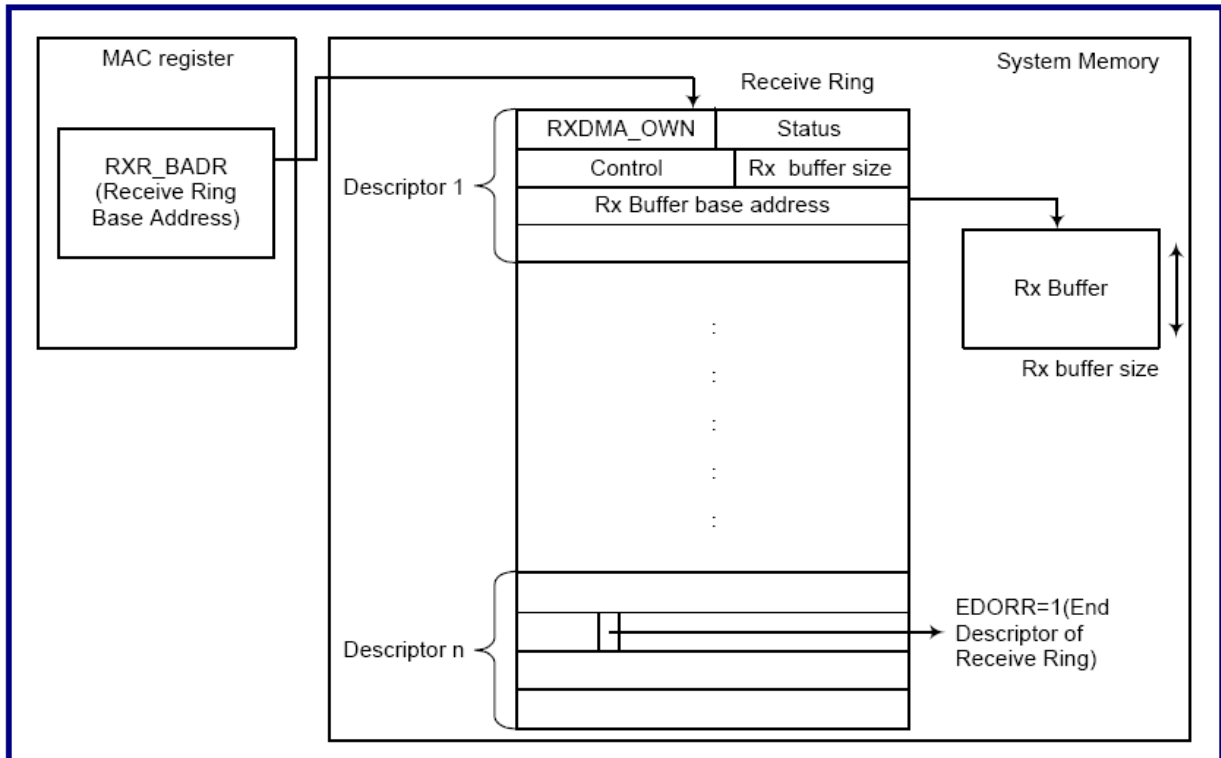


图12-4 接收描述符结构图

接收描述符的结构如下：

RXDES0	RXDMA_OWN	Status
RXDES1	Control	RX buffer size 双字大小
RXDES2	RX buffer base address	接收缓冲区基地址，4字节对齐

注意：

- (1)、每个描述符的地址必须 16 字节对齐；
- (2)、软件必须以字方式对描述符进行访问(大端模式下)；
- (3)、局限性：接收包的最大长度为 2020 字节；

RXDES0：包含帧状态和描述符权限

名称	位	描述
RXDMA_OWN	31	<b>RXDMA 使用权限位</b> 0：软件拥有本描述符使用权限； 1：MAC 拥有本描述符使用权限； 帧传输完成时或者接收缓冲区已满，MAC 自动清除此位
Reserved	30	保留
FRS	29	FRS-First Receive Segment descriptor 该位置 1 表示当前描述符是接收数据包的第一个数据段的描述符
LRS	28	LRS-Last Receive Segment descriptor 该位置 1 表示当前描述符是接收数据包的最后一个数据段的描述符
Reserved	[27: 23]	保留
RX_ODD_NB	22	RX_ODD_NB - Receive Odd Nibbles 该位置 1 表示接收到一个包含半字节的数据包
RUNT	21	RUNT- Runt packet 该位置 1 表示接收到的数据包长度小于 64 字节

		该位仅当 FRS = 1 时有效
FTL	20	FTL - Frame Too Long 该位置 1 表示接收到的包长度大于 1518 字节
CRC_ERR	19	CRC_ERR-Crc Error 0: CRC 校验正确; 1: CRC 校验错误;
RX_ERR	18	RX_ERR-Receive Error 0: 接收数据正常; 1: 接收数据发生错误;
BROADCAST	17	BROADCAST- Broadcast Frame 0: 接收的数据包非广播帧; 1: 接收的数据包为广播帧;
MULTICAST	16	MULTICAST-Multicast Frame 0: 接收的数据包非多播帧 1: 接收的数据包为多播帧
Reserved	[15: 11]	保留
RFL	[10: 0]	RFL-Receive Frame Length 接收到的数据帧长度

#### RXDES1: 控制位与接收缓冲区大小

名称	位	描述
EDORR	31	EDORR-End Descriptor of Receive Ring 该位置 1 表示该描述符为接收描述符队列的最后一个描述符
Reserved	[30: 11]	保留
RXBUF_SIZE	[10: 0]	RXBUF_SIZE - Receive buffer size 接收缓冲区大小, 单位为字节

#### RXDES2: 接收缓冲区的基地址

名称	位	描述
RXBUF_BADR	[31: 0]	Receive buffer base address 接收缓冲区的基地址

### 12.3.3 发送数据包

当软件发送数据时, 它首先将需要发送的数据包写入到发送缓冲区中, 然后再将发送长度和发送缓冲区地址写到发送描述符中并触发 MAC 发送这个数据包。当整个数据包被写到 TX FIFO 后, MAC 开始将此数据发送到以太网上。成功发送后, MAC 将产生相应的中断通知软件。

### 12.3.4 接收数据包

任意时刻有数据包接收, 如果 CRC 校验正确并通过地址过滤, MAC 将把本包数据保存在 RX FIFO 中, 然后发起 DMA 传输, 将数据搬移到系统 RAM 中。如果成功接收到一包数据, MAC 将会产生相应的中断通知软件。

### 12.3.5 地址过滤

MAC 具有多种过滤机制, 可以通过设置以下位来实现:

- (1)、RX\_BROADPKT: MAC 控制寄存器的第 17 位;

- (2)、RX\_MULTIPKT: MAC 控制寄存器的第 16 位;
- (3)、RCV\_ALL: MAC 控制寄存器的第 12 位;
- (4)、HT\_MULTI\_EN: MAC 控制寄存器的第 9 位;

下表为 MAC 控制寄存器中地址过滤设置位与数据包过滤对应关系:

RCV_ALL	RX_MULTIPKT	RX_BROADPKT	HT_MULTI_EN	Group	Description
0	0	0	0	A	只接收与 MAC_ADR 相匹配的数据包
0	0	0	1	B	(1)、接收与 MAC_ADR 相匹配的数据包; (2)、接收与 hash 表相匹配的数据包;
0	0	1	0	C	(1)、接收与 MAC_ADR 相匹配的数据包; (2)、接收广播包;
0	0	1	1	D	(1)、接收与 MAC_ADR 相匹配的数据包; (2)、接收与 hash 表相匹配的数据包; (3)、接收广播包;
0	1	x	x	E	(1)、接收与 MAC_ADR 相匹配的数据包; (2)、接收所有的多播地址数据包;
1	x	x	x	F	接收所有数据包

注:

- (1)、MAC\_ADR 是指寄存器 R32\_ETHE\_MAC\_MADR 和寄存器 R32\_ETHE\_MAC\_LADR;
- (2)、“x”表示该位为 0 或 1;

### 12.3.6 DMA 仲裁配置

DMA 仲裁的配置由寄存器 R32\_ETHE\_DBLAC 的第 9 位(RX\_THR\_EN)决定。

RX\_THR\_EN = 0, TXDMA 和 RXDMA 具有相同的优先级。当 TXDMA 和 RXDMA 同时请求 DAM 传输时, 最后一个使用 DMA 通道的将有较低的优先级。

RX\_THR\_EN = 1, 当 RX FIFO 中已使用的空间大于或者等于 RXFIFO\_HTHR, RXDMA 将具有较高的优先级, RXDMA 将保持较高的优先级直到已使用的空间小于或者等于 RXFIFO\_LTHR, 这时 TXDMA 的优先级高于 RXDMA。因此为了让 MAC 能够正常工作, 软件必须设置 RXFIFO\_HTHR 大于 RXFIFO\_LTHR。

### 12.3.7 LAN 上唤醒(Wake-On\_LAN)

MAC 支持 LAN 上唤醒功能, 包含三种唤醒事件: 连接状态改变、魔法包和唤醒帧。

### 12.3.8 连接状态改变

连接状态改变是指以太网连接的状态改变的事件。PHY 提供一个 Phy\_Linksts 信号, 如果以太网的连接状态改变, 则 Phy\_Linksts 的状态也会随之而改变。一旦 MAC 进入省电模式并且连接状态改变使能, 连接状态改变将会被认为是一次唤醒事件。

### 12.3.9 魔法包(Magic Packet)

魔法包的结构如下: 在其数据部分包含 16 个重复的、无间隔也无中断的站地址; 在此之前是 6 个值为 0xFF 的魔法包同步字节; 魔法包数据部分可能还有一些魔法包模式的周围环境。整个魔法包在以太网帧的结构上必须是齐全的, 结构如下:

DA + SA + ... + 6 \* ( 8hFF ) + ... + 16 \* ( Node address of network adaptor ) + ...

一旦 MAC 进入省电模式并且魔法包使能, 接收到正确的魔法包将会认为是一次唤醒事件。

### 12.3.10 唤醒帧(Wake-up Frame)

当远程机器需要唤醒本系统时，使用唤醒帧唤醒模式，不需要远程机器特定的应用软件，而是基于已有的网络协议即可唤醒，例如发送 ARP 请求。

在设置 MAC 唤醒帧模式之前，系统需要将唤醒帧列表传递给驱动，驱动将根据唤醒帧列表和掩码来初始化 MAC 相应的寄存器。

实现原理：

MAC 最多支持 128 字节掩码，所以唤醒帧长度大于 128 字节的数据流将不参与 CRC 计算，驱动应该根据唤醒帧和掩码值计算出 4 字节的 CRC 值，选择相应的唤醒帧(寄存器 R32\_ETHE\_WOLCR LAN 第 16、17 位)将 CRC 值写到寄存器 R32\_ETHE\_WFCRC 中，掩码为 0 的的字节将不参与 CRC 计算。

MAC 内部每一个唤醒帧都有一个 CRC 计算电路，当接收到一个帧时，MAC 将根据掩码寄存器和接收帧来计算一个 4 字节的 CRC 值，并将此值和寄存器 R32\_ETHE\_WFCRC 进行对比，如果对比结果相同则认为是一个唤醒事件并产生唤醒中断，否则丢弃。如果此时为掉电模式，MAC 并不把接收到的数据保存到系统 RAM 中，所以驱动接收不到唤醒帧。

### 12.3.11 掉电模式

MAC 支持掉电模式，用于减少功耗。在掉电模式下主要有以下特征：

- (1)、不产生中断；
- (2)、不发送数据包；
- (3)、不保存 RX FIFO 中的数据包；
- (4)、当有唤醒事件发生时，产生 WoL；

### 12.3.12 流控制

MAC 支持流控制，与 802.3x 全双工流控和半双工背压流控完全兼容。全双工模式下采用 802.3X 流控制，当 A 和 B 在全双工模式下发送和接收数据时，如果 B 的 RX FIFO 将要溢出，B 将向 A 发送一个暂停帧以防止数据帧丢失。A 将抑制发送，在此期间 B 处理已经接收数据。当暂停时间失效后，A 将继续向 B 发送数据包。全双工流控制简要特点如下：

- (1)、软件对暂停帧的暂停时间进行配置；
- (2)、MAC 能够根据 RX FIFO 的阈值进行自动发送暂停帧；
- (3)、软件可以通过写寄存器来发送暂停帧；

背压模式主要是用在半双工模式中，当 A 在半双工的模式下进行收发数据，如果 RX FIFO 将要溢出，A 将发送连续的阻塞报文(Jam Pattern)避免接收到的数据包被保存在 RX FIFO 中，在此期间 A 将处理已经接收到的数据。当 RX FIFO 的接收能力恢复后阻塞报文(Jam Pattern)将不再发送。半双工背压模式简要特点如下：

- (1)、软件可以对阻塞报文(Jam Pattern)的长度进行配置；
- (2)、MAC 能够根据 RX FIFO 的阈值决定发送阻塞(Jam Pattern)报文；

## 12.4 寄存器描述

MAC 相关寄存器物理起始地址为：0x0040 6000

MAC 相关寄存器列表(地址范围：0x00---0xFF)

表 15-1 MAC 相关寄存器列表

名称	偏移地址	描述	复位值
R32_ETHE_ISR	0X00	中断状态寄存器	32h0000 0000
R32_ETHE_IMR	0X04	中断使能寄存器	32h0000 0000
R32_ETHE_MAC_MADR	0x08	MAC 地址高字节寄存器	32h0000 0000
R32_ETHE_MAC_LADR	0x0C	MAC 地址低字节寄存器	32h0000 0000

R32_ETHE_MAHT0	0X10	多播过滤 hash 表 0	32h0000 0000
R32_ETHE_MAHT1	0X14	多播过滤 hash 表 1	32h0000 0000
R32_ETHE_TXPD	0X18	发送轮询启动寄存器	32h0000 0000
R32_ETHE_RXPD	0X1C	接收轮询启动寄存器	32h0000 0000
R32_ETHE_TXR_BADR	0X20	发送描述符队列基地址	32h0000 0000
R32_ETHE_RXR_BADR	0X24	接收描述符队列基地址	32h0000 0000
R32_ETHE_ITC	0X28	中断定时控制寄存器	32h0000 0000
R32_ETHE_APTC	0X2C	自动轮询定时控制寄存器	32h0000 0000
R32_ETHE_DBLAC	0X30	DMA 突发长度与仲裁控制寄存器	32h0000 0000
R32_ETHE_MACCR	0X88	MAC 控制寄存器	32h0000 0000
R32_ETHE_MACSR	0X8C	MAC 状态寄存器	32h0000 0000
R32_ETHE_PHYCR	0X90	PHY 控制寄存器	32h0000 0000
R32_ETHE_PHYWDATA	0X94	PHY 写数据寄存器	32h0000 0000
R32_ETHE_FCR	0X98	流控制寄存器	32h0000 A400
R32_ETHE_BPR	0X9C	背压寄存器	32h0000 0000
R32_ETHE_WOLCR	0XA0	LAN 上唤醒控制寄存器	32h0000 0000
R32_ETHE_WOLSR	0XA4	LAN 上唤醒状态寄存器	32h0000 0000
R32_ETHE_WFCRC	0XA8	唤醒帧 CRC 寄存器	32h0000 0000
R32_ETHE_WFBM1	0XB0	唤醒帧掩码 1	32h0000 0000
R32_ETHE_WFBM2	0XB4	唤醒帧掩码 2	32h0000 0000
R32_ETHE_WFBM3	0XB8	唤醒帧掩码 3	32h0000 0000
R32_ETHE_WFBM4	0XBC	唤醒帧掩码 4	32h0000 0000
R32_ETHE_DMAFIFOS	0XC8	DMA/FIFO 状态寄存器	32h0C00 0000

PHY 相关寄存器列表(地址范围: 0x00---0x01)

注: PHY寄存器的宽度为16位

表 12-2 PHY 相关寄存器列表

名称	偏移地址	描述	复位值
R16_PHY_BMCR	0X00	基本模式控制寄存器	16h3100
R16_PHY_BMSR	0X01	基本模式状态寄存器	16h7809

#### 15.4.1 MAC 寄存器

中断状态寄存器(R32\_ETHE\_ISR):

名称	位	访问	描述	复位值
Reserved	[31: 10]	-	保留	-
RB_PHYSTS_CHG	9	RC	PHY 连接状态改变标志 0: 连接状态未改变 1: 连接状态发生改变	1h0
Reserved	8	-	保留	1h0
RB_RPKT_LOST	7	RC	RXFIFO 溢出标志 0: RXFIFO 未溢出 1: RXFIFO 已溢出 如果 RXFIFO 发生溢出, 则接收数据丢失	1h0
RB_RPKT_SAV	6	RC	数据成功接收到 RXFIFO 标志	1h0
RB_XPKT_LOST	5	RC	该位置 1 表示发送数据丢失, 由迟到冲突或者冲突次数过多所导致	1h0

RB_XPKT_OK	4	RC	该位置 1 表示数据成功发送	1h0
RB_NOTXBUF	3	RC	该位置 1 表示发送缓冲区无效	1h0
RB_XPKT_FINISH	2	RC	该位置 1 表示 TXDMA 将数据成功写入到 TXFIFO 中	1h0
RB_NORXBUF	1	RC	该位置 1 表示接收缓冲区无效	1h0
RB_RPKT_FINISH	0	RC	该位置 1 表示 RXDMA 将数据成功写入到 RXFIFO 中	1h0

中断使能寄存器 (R32\_ETHE\_IMR) :

名称	位	访问	描述	复位值
Reserved	[31: 10]	-	保留	-
RB_PHYSTS_CHG_EN	9	RW	PHY 连接状态改变中断使能/禁止位 0: 禁止 PHY 连接状态改变中断 1: 使能 PHY 连接状态改变中断	1h0
Reserved	8	-	保留	1h0
RB_RPKT_LOST_EN	7	RW	RXFIFO 溢出中断使能/禁止位 0: 禁止 RXFIFO 溢出中断 1: 使能 RXFIFO 溢出中断	1h0
RB_ISR_RPKT_SAV_EN	6	RW	数据成功接收到 RXFIFO 中断使能/禁止位 0: 禁止数据成功接收中断 1: 使能数据成功接收中断	1h0
RB_XPKT_LOST_EN	5	RW	数据发送丢失中断使能/禁止位 0: 禁止数据发送丢失中断 1: 使能数据发送丢失中断	1h0
RB_XPKT_OK_EN	4	RW	数据发送成功中断使能/禁止位 0: 禁止数据发送成功中断 1: 使能数据发送成功中断	1h0
RB_NOTXBUF_EN	3	RW	发送缓冲区无效中断使能/禁止位 0: 禁止发送缓冲区无效中断 1: 使能发送缓冲区无效中断	1h0
RB_XPKT_FINISH_EN	2	RW	TXDMA 写数据到 TXFIFO 中断使能/禁止位 0: 禁止 TXDMA 写数据到 TXFIFO 中断 1: 使能 TXDMA 写数据到 TXFIFO 中断	1h0
RB_NORXBUF_EN	1	RW	接收缓冲区无效中断使能/禁止位 0: 禁止接收缓冲区无效中断 1: 使能接收缓冲区无效中断	1h0
RB_RPKT_FINISH_EN	0	RW	RXDMA 将数据成功的写入接收缓冲区中 RXDMA 写数据到接收缓冲区成功中断使能/禁止位 0: 禁止 RXDMA 写数据到接收缓冲区成功中断 1: 使能 RXDMA 写数据到接收缓冲区成功中断	1h0

MAC 地址高字节寄存器 (R32\_ETHE\_MAC\_MADR) :

名称	位	访问	描述	复位值
Reserved	[31: 16]	-	保留	-
RB_MAC_MADR	[15: 0]	RW	MAC 地址的高 2 个字节	16h0

MAC 地址低字节寄存器 (R32\_ETHE\_MAC\_LADR) :

名称	位	访问	描述	复位值
RB_MAC_LADR	[31: 0]	RW	MAC 地址的低 4 个字节	32h0

多播过滤 hash 表 0 (R32\_ETHE\_MAHT0) :

名称	位	访问	描述	复位值
RB_MATH0	[31: 0]	RW	多播过滤 hash 表 0 (hash table[31:0])	32h0

多播过滤 hash 表 1 (R32\_ETHE\_MAHT1) :

名称	位	访问	描述	复位值
RB_MATH1	[31: 0]	RW	多播过滤 hash 表 1 (hash table[64:32])	32h0

发送轮询启动寄存器 (R32\_ETHE\_TXPD) :

名称	位	访问	描述	复位值
RB_TXPT	[31: 0]	WO	写任意值将会触发 MAC 读取发送描述符, 如果检测到 txdma_own = 1, 将会把数据写入到 TXFIFO 中。 只写寄存器, 读操作总是返回 0	32h0

接收轮询启动寄存器 (R32\_ETHE\_RXPD) :

名称	位	访问	描述	复位值
RB_RXPT	[31: 0]	WO	写任意值将会触发 MAC 读取接收描述符, 如果检测到 rxdma_own = 1, 将会把接收到的数据从 RXFIFO 写到接收缓冲区中。 只写寄存器, 读操作总是返回 0	32h0

发送描述符队列基地址寄存器 (R32\_ETHE\_TXR\_BADR) :

名称	位	访问	描述	复位值
RB_TXR_BADR	[31: 0]	RW	发送描述符队列的基地址, 这个地址必须 16 位对齐, MAC 将会把位[3:0]当做 0 处理	32h0

接收描述符队列基地址 (R32\_ETHE\_RXR\_BADR) :

名称	位	访问	描述	复位值
RB_RXR_BADR	[31: 0]	RW	接收描述符队列的基地址, 这个地址必须 16 位对齐, MAC 会把位[3:0]当做 0 处理	32h0

中断定时控制寄存器 (R32\_ETHE\_ITC) :

名称	位	访问	描述	复位值
Reserved	[31: 16]	-	保留	-
RB_TXINT_TIME_SEL	15	RW	<b>TX 周期时间</b> 0: 100Mbps $\Rightarrow$ 5.12us 10Mbps $\Rightarrow$ 51.2us 1: 100Mbps $\Rightarrow$ 81.92us 10Mbps $\Rightarrow$ 819.2us	1h0
RB_TXINT_THR	[14: 12]	RW	<b>发送最大挂起中断数</b> 该域定义了在中断产生之前所能挂起的最大中断数， 当 TXINT_THR 不等于 0 时，以太网控制传输数据包的个数等于 TXINT_THR 时将产生中断； 当 TXINT_THR = 0，且 TXINT_CNT = 0 时，是否产生中断由发送描述符中的 TDES1 的 TXIC 决定	3h0
RB_TXINT_CNT	[11: 8]	RW	<b>最大等待时间</b> 该域定义了发送完数据后产生中断的最大等待时间，单位为： TXINT_TIME_SEL。 TXINT_CNT = 0 时，禁用此功能； 当 TXINT_THR = 0，且 TXINT_CNT = 0 时，是否产生中断由发送描述符中的 TDES1 的 TXIC 决定	4h0
RB_RXINT_TIME_SEL	7	RW	<b>RX 周期时间</b> 0: 100Mbps $\Rightarrow$ 5.12us 10Mbps $\Rightarrow$ 51.2us 1: 100Mbps $\Rightarrow$ 81.92us 10Mbps $\Rightarrow$ 819.2us	1h0
RB_RXINT_THR	[6: 4]	RW	<b>接收最大挂起中断数</b> 该域定义了在中断产生之前所能挂起的最大中断数。 当 RXINT_THR 不等于 0 时，MAC 接收数据包的个数等于 RXINT_THR 时将产生中断。 当 RXINT_THR = 0，且 RXINT_CNT = 0 时，MAC 收到数据包后会立即产生中断	3h0
RB_RXINT_CNT	[3: 0]	RW	<b>最大等待时间</b> 该域定义了接收完数据后产生中断的最大等待时间，时间单位为： RXINT_TIME_SEL。 RXINT_CNT = 0 时，禁用此功能， 当 RXINT_THR = 0，且 RXINT_CNT = 0 时，MAC 收到数据包后会立即产生中断	4h0

中断定时控制寄存器允许软件通过设置此寄存器来减少接收和发送中断的次数，这可以有效的降低 CPU 的使用率。此寄存器定义了接收数据包和发送数据包的中断阈值以及两个定时器。中断阈值规定了在产生中断之前能够挂起的最大的接收(发送)中断数，定时器则规定了在一个数据包被接收(发



送)后到产生中断的最大时间。中断阈值允许多个数据包产生一次中断，定时器允许此中断可以被挂起的最大的时间。中断阈值可以有效的避免吞吐量受到影响，定时器则可以有效的避免资源被长期占用。

在MAC内部有一个发送计数器(TXPKT\_CNT)是为了计算被发送的数据包的个数，当这个计数器达到了TXPKT\_CNT(TXPKT\_CNT 不等于0)，MAC将会产生中断并将TXPKT\_CNT清零。接收机制同发送机制类似，在MAC内部也有一个接收计数器(RXPKT\_CNT)是为了计算接收的数据包的个数，当接收的数据包个数达到RXPKT\_CNT(RXPKT\_CNT不等于0)，MAC会产生中断并将RXPKT\_CNT清零。

TXINT_THR = 0	TXINT_CNT = 0	MAC行为
True	True	如果描述符中的TXIC为1，数据包成功发送后立即产生中断，TXPKT_CNT清零
True	False	数据包成功发送并且定时器达到TXINT_CNT的值后产生中断，TXPKT_CNT清零
False	True	当TXPKT_CNT = TXINT_THR时产生中断，TXPKT_CNT清零
False	False	以下条件会产生中断： (1)、TXPKT_CNT = TXINT_THR； (2)、TXPKT_CNT = 1，且定时器达到TXPKT_CNT值； TXPKT_CNT清零

RXINT_THR = 0	RXINT_CNT = 0	MAC行为
True	True	如果描述符中的RXIC为1，数据包成功接收后立即产生中断，RXPKT_CNT清零
True	False	数据接收成并且定时器达到RXINT_CNT的值后产生中断，RXPKT_CNT清零
False	True	当RXPKT_CNT = RXINT_THR时产生中断，RXPKT_CNT清零
False	False	以下条件会产生中断： RXPKT_CNT = RXINT_THR RXPKT_CNT = 1，且定时器达到RXPKT_CNT值， RXPKT_CNT清零

自动轮询定时器控制寄存器(R32\_ETHE\_APTC)：

名称	位	访问	描述	复位值
Reserved	[31: 13]	-	保留	-
RB_TXPOLL_TIME_SEL	12	RW	<b>TX 轮询周期</b> 0: 100Mbps $\Rightarrow$ 5.12us 10Mbps $\Rightarrow$ 51.2us 1: 100Mbps $\Rightarrow$ 81.92us 10Mbps $\Rightarrow$ 819.2us	1h0
RB_TXPOLL_CNT	[11: 8]	RW	<b>发送描述符自动轮询周期</b> 时间单位为：RB_TXPOLL_TIME_SEL， RB_TXPOLL_CNT = 0，MAC 将不自动 轮询发送描述符； RB_TXPOLL_CNT 不等于 0 时，MAC 将 自动轮询发送描述符队列	4h0
Reserved	[7: 5]	-	保留	-
RB_RXPOLL_TIME_SEL	4	RW	<b>RX 轮询周期</b>	1h0

			0: 100Mbps $\Rightarrow$ 5.12us 10Mbps $\Rightarrow$ 51.2us 1: 100Mbps $\Rightarrow$ 81.92us 10Mbps $\Rightarrow$ 819.2us	
RB_RXINT_CNT	[3: 0]	RW	<b>接收描述符自动轮询周期</b> 时间单位为: RB_RXPOLL_TIME_SEL, RB_RXPOLL_CNT = 0, MAC 将不自动 轮询接收描述符; RB_RXPOLL_CNT 不等于 0 时, MAC 将 自动轮询接收描述符队列	4h0

通过对R32\_ETHT\_APTC配置, 可以让MAC自动轮询描述队列从而减少CPU的利用。当发送自动轮询开启, MAC将自动轮询发送描述符直到轮询时间结束。如果此功能禁止, 软件在准备好数据后, 可以通过对R32\_ETHER\_TXPD进行写任意数操作来触发MAC读发送描述符。当接收轮询功能开启, MAC会自动的轮询接收描述符, 轮询周期为RB\_RXPOLL\_TIME\_SEL\* RB\_RXINT\_CNT; 如果此功能禁止, 软件释放接收描述符后, 可以通过对R32\_ETHER\_RXPD进行写任意数操作来触发MAC读接收描述符。该寄存器推荐使用值为: 32h0000 0001。

#### DMA 突发长度与仲裁控制寄存器 (R32\_ETHER\_DBLAC):

名称	位	访问	描述	复位值
Reserved	[31: 16]	-	保留	-
RB_INCR_SEL	[15: 14]	RW	<b>突发传输长度控制域</b> 当 MAC 在总线上执行 INCR 突发传输命令, 该域定义了一次突发传输的长度 00: 无限制 01: 无限制 10: 最大数据长度为 128 字节 (32*4) 11: 最大数据长度为 256 字节 (64*4)	2h0
Reserved	[13: 10]	-	保留	-
RB_RX_THR_EN	9	RW	<b>RX FIFO 阈值仲裁使能位</b> 0: 不使能 1: 使能	1h0
RB_RXFIFO_HTHR	[8: 6]	RW	<b>RX FIFO 高阈值</b> 当 RXFIFO 的使用空间大于或者等于 RX 的 FIFO 高阈值, RXDMA 的优先级高于 TXDMA。RXDMA 保持高优先级, 直到 RXFIFO 使用空间小于或者等于 RXFIFO 的低阈值。用户必须保证 RXFIFO_HTHR 大于 RXFIFO_LTHR, 以 256 字节为单位 000: 阈值为 0 001: 1/8 RXFIFO, 256 字节 010: 2/8 RXFIFO, 512 字节 011: 3/8 RXFIFO, 768 字节 100: 4/8 RXFIFO, 1024 字节 101: 5/8 RXFIFO, 1280 字节 110: 6/8 RXFIFO, 1536 字节 111: 7/8 RXFIFO, 1792 字节	3h0
RB_RXFIFO_LTHR	[5: 3]	RW	<b>RXFIFO 低阈值</b>	3h0

			当 RXFIFO 使用的空间少于或者等于 RXFIFO 的低阈值, TXDMA 的优先级高于 RXDMA, 以 256 字节为单位 000: 阈值为 0 001: 1/8 RXFIFO, 256 字节 010: 2/8 RXFIFO, 512 字节 011: 3/8 RXFIFO, 768 字节 100: 4/8 RXFIFO, 1024 字节 101: 5/8 RXFIFO, 1280 字节 110: 6/8 RXFIFO, 1536 字节 111: 7/8 RXFIFO, 1792 字节	
Reserved	[2: 0]	-	保留	-

该寄存器推荐使用值为: 32h0000 0390。

MAC 控制寄存器(R32\_ETHE\_MACCR):

名称	位	访问	描述	复位值
Reserved	[31: 19]	-	保留	-
RB_SPEED_100	18	RW	<b>速度选择位</b> 0: 10Mbps 1: 100Mbps	1h0
RB_RX_BROADPKT	17	RW	<b>接收广播包使能位</b> 0: 未使能 1: 使能	1h0
RB_RX_MULTIPKT	16	RW	<b>接收所有多播包使能位</b> 0: 未使能 1: 使能	1h0
RB_FULLLDUP	15	RW	<b>全双工使能位</b> 0: 未使能 1: 使能	1h0
RB_CRC_APD	14	RW	<b>发送包添加 CRC 使能位</b> 0: 未使能 1: 使能	1h0
Reserved	13	RW	保留	
RB_RCV_ALL	12	RW	<b>接收所有数据包使能位</b> 0: 未使能 1: 使能	1h0
RB_RX_FTL	11	RW	<b>长数据包接收使能位</b> 0: 未使能 1: 使能 该位置 1, 允许接收长度大于 1518 字节的数据包	1h0
RB_RX_RUNT	10	RW	<b>短数据包接收使能位</b> 0: 未使能 1: 使能 该位置 1, 允许接收长度小于 64 字节的数据包	1h0
RB_HT_MULTI_EN	9	RW	<b>hash 表过滤使能位</b> 0: 未使能 1: 使能	1h0

RB_RCV_EN	8	RW	<b>接收使能位</b> 0: 未使能 1: 使能	1h0
Reserved	7	-	保留	-
RB_ENRX_IN_HALFTX	6	RW	半双工模式下, 该位置 1 表示发送数据包时允许接收	1h0
RB_XMF_EN	5	RW	<b>发送使能位</b> 0: 未使能 1: 使能	1h0
RB_CRC_DIS	4	RW	<b>接收包 CRC 校验使能位</b> 0: 未使能 1: 使能	1h0
RB_LOOP_EN	3	RW	<b>回环模式使能位</b> 0: 未使能 1: 使能	1h0
RB_SW_RST	2	RW	<b>软件复位</b> 向此位写 1 将使能软件复位, 软件复位将持续 64 个总线时钟, 然后自动清零	1h0
RB_RDMA_EN	1	RW	<b>接收 DMA 使能位</b> 0: 未使能 1: 使能 如果此位为 0, 则接收立即停止	1h0
RB_TDMA_EN	0	RW	<b>发送 DMA 使能位</b> 0: 未使能 1: 使能 如果此位为 0, 则发送立即停止	1h0

## MAC 状态寄存器(R32\_ETHE\_MACSR):

名称	位	访问	描述	复位值
Reserved	[31: 12]	-	保留	-
RB_COL_EXCEED	11	RC	冲突超过 16 次	1h0
RB_LATE_COL	10	RC	检测到冲突	1h0
RB_XPKT_LOST	9	RC	由于最近一次冲突或者冲突过多导致发送包丢失	1h0
RB_XPKT_OK	8	RC	数据包发送成功	1h0
RB_RUNT	7	RC	接收器检测到长度小于 64 字节的帧	1h0
RB_FTL	6	RC	接收器检测到长度大于 1518 字节的帧	1h0
RB_CRC_ERR	5	RC	接收包 CRC 校验正确, 该位仅在 RB_CRC_DIS 为 0 时有效	1h0
RB_RPKT_LOST	4	RC	RXFIFO 溢出, 接收数据包丢失	1h0
RB_RPKT_SAVE	3	RC	数据包被成功接收到 RXFIFO 中	1h0
RB_COL	2	RC	由于冲突导致接收包丢失	1h0
RB_BROADCAST	1	RC	接收到广播包	1h0
RB_MULTICAST	0	RC	接收到多播包	1h0

## PHY 控制寄存器(R32\_ETHE\_PHYCR):

名称	位	访问	描述	复位值
Reserved	[31: 28]	-	保留	-
RB_MIWD	27	RW	该位置 1 初始化写序列写操作完成后自动清零	1h0
RB_MIRD	26	RW	该位置 1 初始化读序列读操作完成后自动清零	1h0
RB_REGAD	[25: 21]	RW	PHY 寄存器地址	5h0
RB_PHYAD	[20: 16]	RW	PHY 地址(必须写 0x10)	5h0
RB_MIRDATA	[15: 0]	RO	读数据寄存器, 从 PHY 寄存器读出的数据将保存在 RB_MIRDATA 中	16h0

PHY 写数据寄存器 (R32\_ETHE\_PHYWDATA) :

名称	位	访问	描述	复位值
Reserved	[31: 16]	-	保留	-
RB_MIWDATA	[15: 0]	RW	写数据寄存器, 向 PHY 写入的数据保存在 RB_MIWDATA 中	16h0

流控制寄存器 (R32\_ETHE\_FCR) :

名称	位	访问	描述	复位值
RB_PAUSE_TIME	[31: 16]	RW	<b>暂停帧持续时间</b> 单位为 512 位时间, 512 位时间为: 100Mbps $\Rightarrow$ 5.12us 10Mbps $\Rightarrow$ 51.2us	16h0
RB_FC_HIGH	[15: 12]	RW	<b>RX FIFO 剩余空间高阈值</b> 当 RX FIFO 的剩余空间大于高阈值时, 发送一个暂停时间为 32h0000 的暂停帧, 单位为 128 字节, 默认值为 4hA	4hA
RB_FC_LOW	[11: 8]	RW	<b>RX FIFO 剩余空间低阈值</b> 当 RX FIFO 的剩余空间小于低阈值时, 发送一个暂停时间为 RB_PAUSE_TIME 的暂停帧, 单位为 128 字节, 默认值为 4h4	4h4
Reserved	[7: 5]	-	保留	-
RB_RX_PAUSE	4	RC	<b>接收到暂停帧标志位</b> 0: 未接收到暂停帧 1: 接收到暂停帧	1h0
RB_TXPAUSED	3	RO	由于接收到一个暂停帧, 发送包被暂停	1h0
RB_FCTHR_EN	2	RW	<b>阈值模式的流控制使能位</b> 0: 未使能 1: 使能	1h0
RB_TX_PAUSE	1	RW	<b>发送暂停帧</b> 软件可以通过设置该位来启动发送暂停帧, 发送完毕后自动清零	1h0
RB_FC_EN	0	RW	<b>流控制模式使能位</b> 0: 未使能 1: 使能	1h0

背压寄存器(R32\_ETHE\_BPR)：

名称	位	访问	描述	复位值
Reserved	[31: 2]	-	保留	-
RB_BK_LOW	[11: 8]	RW	<b>RXFIFO 剩余空间低阈值</b> 当 RX FIFO 的剩余空间小于此低阈值时，MAC 将会产生 Jam	4h4
RB_BKJAM_LEN	[7: 4]	RW	<b>背压 Jam 长度</b> 0000: 4 字节 0001: 8 字节 0010: 16 字节 0011: 32 字节 0100: 64 字节 0101: 128 字节 0110: 256 字节 0111: 512 字节 1000: 1024 字节 1001: 1518 字节 1010: 2048 字节 其它: 4 字节	4h0
Reserved	[3: 2]	-	保留	-
RB_BK_MODE	1	RW	<b>背压模式</b> 0: 接收任意数据包产生 Jam 1: 当接收数据包地址匹配时产生 Jam	1h0
RB_BK_EN	0	RW	<b>背压模式使能位</b> 0: 未使能 1: 使能	1h0

LAN 上唤醒控制寄存器(R32\_ETHE\_WOLCR)：

名称	位	访问	描述	复位值
Reserved	[31: 26]	-	保留	-
RB_WOL_TYPE	[25: 24]	RW	<b>WoL 信号类型，必须写 00</b>	2h0
Reserved	[23: 19]	-	保留	-
RB_SW_PDNPHY	18	RW	软件通过此位控制 PHY 进入掉电模式，建议用 RB_SLP_ETH_PWR_DN 代替	1h0
RB_WAKEUP_SEL	[17: 16]	RW	<b>唤醒帧选择位</b> 当驱动访问唤醒帧 CRC 或者字节掩码时，用于选择相应的唤醒帧 00: 选择唤醒帧 1 01: 选择唤醒帧 2 10: 选择唤醒帧 3 11: 选择唤醒帧 4	2h0
RB_POWER_STATE	[15: 14]	RW	<b>电源状态位</b> 这两位主要是确定当前的电源状态 00: D0 01: D1 10: D2	2h0

			11: D3	
Reserved	[13: 7]	-	保留	-
RB_WAKEUP4_EN	6	RW	唤醒帧 4 使能位 0: 未使能 1: 使能	1h0
RB_WAKEUP3_EN	5	RW	唤醒帧 3 使能位 0: 未使能 1: 使能	1h0
RB_WAKEUP2_EN	4	RW	唤醒帧 2 使能位 0: 未使能 1: 使能	1h0
RB_WAKEUP1_EN	3	RW	唤醒帧 1 使能位 0: 未使能 1: 使能	1h0
RB_MAGICPKT	2	RW	魔法包唤醒使能位 0: 未使能 1: 使能	1h0
RB_LINKCHG1_EN	1	RW	连接改变为 1 唤醒使能位(连接) 0: 未使能 1: 使能	1h0
RB_LINKCHG0_EN	0	RW	连接改变为 0 唤醒使能位(断开) 0: 未使能 1: 使能	1h0

## LAN 上唤醒状态寄存器(R32\_ETHE\_WOLSR) :

名称	位	访问	描述	复位值
Reserved	[31: 7]	-	保留	-
RB_WAKEUP4_STS	6	RWI	唤醒帧 4 唤醒事件标志位 0: 无唤醒帧 4 唤醒事件产生 1: 有唤醒帧 4 唤醒事件产生	1h0
RB_WAKEUP3_STS	5	RWI	唤醒帧 3 唤醒事件标志位 0: 无唤醒帧 3 唤醒事件产生 1: 有唤醒帧 3 唤醒事件产生	1h0
RB_WAKEUP2_STS	4	RWI	唤醒帧 2 唤醒事件标志位 0: 无唤醒帧 2 唤醒事件产生 1: 有唤醒帧 2 唤醒事件产生	1h0
RB_WAKEUP1_STS	3	RWI	唤醒帧 1 唤醒事件标志位 0: 无唤醒帧 1 唤醒事件产生 1: 有唤醒帧 1 唤醒事件产生	1h0
RB_MAGICPKT_STS	2	RWI	魔法包唤醒事件标志位 0: 无魔法包唤醒事件产生 1: 有魔法包唤醒事件产生	1h0
RB_LINKCHG1_STS	1	RWI	连接唤醒事件标志位 0: 无连接唤醒事件产生 1: 有连接唤醒事件产生	1h0
RB_LINKCHG0_STS	0	RWI	断开唤醒事件标志位 0: 无断开唤醒事件产生 1: 有断开唤醒事件产生	1h0

唤醒帧 CRC 寄存器(R32\_ETHE\_WFCRC)：

名称	位	访问	描述	复位值
RB_WFCRC	[31: 0]	RW	<b>唤醒帧 CRC 值</b> 当 WAKEUP_SEL = 2b00 时，读写唤醒帧 1 的 CRC 寄存器； 当 WAKEUP_SEL = 2b01 时，读写唤醒帧 2 的 CRC 寄存器； 当 WAKEUP_SEL = 2b10 时，读写唤醒帧 3 的 CRC 寄存器； 当 WAKEUP_SEL = 2b11 时，读写唤醒帧 4 的 CRC 寄存器	32h0

唤醒帧掩码 1 双字寄存器 (R32\_ETHE\_WFBM1)：

名称	位	访问	描述	复位值
RB_WFBM1	[31: 0]	RW	<b>唤醒帧掩码 1 双字寄存器</b> 此寄存器保存唤醒帧 1-32 字节的掩码，位 0 为第 1 字节掩码，位 31 为第 32 字节掩码 当 WAKEUP_SEL = 2b00 时，读写唤醒帧 1 的 R32_ETHE_WFBM1 寄存器； 当 WAKEUP_SEL = 2b01 时，读写唤醒帧 2 的 R32_ETHE_WFBM1 寄存器； 当 WAKEUP_SEL = 2b10 时，读写唤醒帧 3 的 R32_ETHE_WFBM1 寄存器； 当 WAKEUP_SEL = 2b11 时，读写唤醒帧 4 的 R32_ETHE_WFBM1 寄存器；	32h0

唤醒帧掩码 2 双字寄存器 (R32\_ETHE\_WFBM2)：

名称	位	访问	描述	复位值
RB_WFBM2	[31: 0]	RW	<b>唤醒帧掩码 2 双字寄存器</b> 此寄存器保存唤醒帧 33-64 字节的掩码，位 0 为第 33 字节掩码，位 31 为第 64 字节掩码 当 WAKEUP_SEL = 2b00 时，读写唤醒帧 1 的 R32_ETHE_WFBM2 寄存器； 当 WAKEUP_SEL = 2b01 时，读写唤醒帧 2 的 R32_ETHE_WFBM2 寄存器； 当 WAKEUP_SEL = 2b10 时，读写唤醒帧 3 的 R32_ETHE_WFBM2 寄存器； 当 WAKEUP_SEL = 2b11 时，读写唤醒帧 4 的 R32_ETHE_WFBM2 寄存器；	32h0

唤醒帧掩码 3 双字寄存器 (R32\_ETHE\_WFBM3)：

名称	位	访问	描述	复位值
RB_WFBM3	[31: 0]	RW	<b>唤醒帧掩码 3 双字寄存器</b> 此寄存器保存唤醒帧 65-96 字节的掩码，位 0 为第 65 字节掩码，位 31 为第 96 字节掩码	32h0



			当 WAKEUP_SEL = 2b00 时, 读写唤醒帧 1 的 R32_ETHE_WFBM 寄存器; 当 WAKEUP_SEL = 2b01 时, 读写唤醒帧 2 的 R32_ETHE_WFBM 寄存器; 当 WAKEUP_SEL = 2b10 时, 读写唤醒帧 3 的 R32_ETHE_WFBM 寄存器; 当 WAKEUP_SEL = 2b11 时, 读写唤醒帧 4 的 R32_ETHE_WFBM 寄存器;	
--	--	--	--	--

唤醒帧掩码 4 双字寄存器 (R32\_ETHE\_WFBM) :

名称	位	访问	描述	复位值
RB_WFBM	[31: 0]	RW	<b>唤醒帧掩码 4 双字寄存器</b> 此寄存器保存唤醒帧 97-128 字节的掩码, 位 0 为第 97 字节掩码, 位 31 为第 128 字节掩码 当 WAKEUP_SEL = 2b00 时, 读写唤醒帧 1 的 R32_ETHE_WFBM 寄存器; 当 WAKEUP_SEL = 2b01 时, 读写唤醒帧 2 的 R32_ETHE_WFBM 寄存器; 当 WAKEUP_SEL = 2b10 时, 读写唤醒帧 3 的 R32_ETHE_WFBM 寄存器; 当 WAKEUP_SEL = 2b11 时, 读写唤醒帧 4 的 R32_ETHE_WFBM 寄存器;	32h0

DMA 与 FIFO 状态寄存器 (R32\_ETHE\_DMAFIFOS) :

名称	位	访问	描述	复位值
RB_TXD_REQ	31	RO	TXDMA 请求	1h0
RB_RXD_REQ	30	RO	RXDMA 请求	1h0
RB_DAR_TXGNT	29	RO	TXDMA 准许	1h0
RB_DAR_RXGNT	28	RO	RXDMA 准许	1h0
RB_TXFIFO_EMPTY	27	RO	TX FIFO 空	1h1
RB_RXFIFO_EMPTY	26	RO	RX FIFO 空	1h1
Reserved	[25: 15]	-	保留	-
RB_TXDMA2_SM	[14: 12]	RO	TXDMA2 状态机	3h0
RB_TXDMA1_SM	[11: 8]	RO	TXDMA1 状态机	4h0
Reserved	7	-	保留	-
RB_RXDMA2_SM	[6: 4]	RO	RXDMA2 状态机	3h0
RB_RXDMA1_SM	[3: 0]	RO	RXDMA1 状态机	4h0

### 12.4.2 PHY 寄存器

基本模式控制寄存器 (R16\_PHY\_BMCR) :

名称	位	访问	描述	复位值
RB_RESET	15	RW	<b>软件复位</b> 0: 无动作 1: 进行软件复位, 建议用 RB_MISC_ETH_RST 代替 复位完毕后该位自动清 0	1h0

RB_LOOK_BACK	14	RW	<b>回环模式控制位</b> 0: 正常操作 1: 回环模式	1h0
RB_SPEED_SEL	13	RW	<b>速度选择位</b> 0: 10Mbps 1: 100Mbps	1h1
RB_AUTO_NEG	12	RW	<b>自动协商使能/禁止位</b> 0: 自动协商禁止 1: 自动协商使能	1h1
RB_POWER_DOWN	11	RW	<b>掉电模式控制位</b> 0: 正常操作 1: 掉电模式	1h0
RB_ISOLATE	10	RW	<b>隔离模式控制位</b> 0: 正常操作 1: 隔离模式使能	1h0
RB_RES_AUTO_NEG	9	RW	<b>重启自动协商控制位</b> 0: 正常操作 1: 自动重启自动协商	1h0
RB_DUPLEX_MDE	8	RW	<b>全双工模式控制位</b> 0: 正常操作 1: 全双工模式使能	1h1
RB_COLLISION_TEST	7	RW	<b>冲突测试控制位</b> 0: 正常操作 1: 冲突测试模式使能	1h0
Reserved	[6: 0]	-	保留, 写 0, 读无意义	-

该寄存器在复位后已经设置为最佳值, 如果没必要请不要对此寄存器做更改。

基本模式状态寄存器(R16\_PHY\_BMSR):

名称	位	访问	描述	复位值
RB_100BASE_T4	15	R0	100BASE_T4 0: PHY 不可以工作在 100BASE_T4 模式下; 1: PHY 可以工作在 100BASE_T4 模式下;	1h0
RB_100BASE_TX_FULL_DUPLEX	14	R0	100BASE_TX 全双工 0: PHY 不可以工作在 100BASE_TX 全双工模式下 1: PHY 可以工作在 100BASE_TX 全双工模式下;	1h1
RB_100BASE_T_HALF_DUPLEX	13	R0	100BASE_T 半双工 0: PHY 不可以工作在 100BASE_T 半双工模式下; 1: PHY 可以工作在 100BASE_T 半双工模式下;	1h1
RB_10BASE_T_FULL_DUPLEX	12	R0	10BASE_T 全双工 0: PHY 不可以工作在 10BASE_T 全双工模式下; 1: PHY 可以工作在 10BASE_T 全双工模式下;	1h1

RB_10BASE_T_HALF_DUPLX	11	R0	10BASE_T 半双工 0: PHY 不可以工作在 10BASE_T 半双工模式下; 1: PHY 可以工作在 10BASE_T 半双工模式下;	1h1
Reserved	[10: 7]	-	保留	-
RB_MF_PRE_SUP	6	R0	内部使用	1h1
RB_AUTO_NEG_CMP	5	R0	自动协商结束标志位 0: 自动协商未结束; 1: 自动协商已结束;	1h0
RB_REMOTE_FAULT	4	R0	远程故障标志位 0: 未检测到远程故障; 1: 检测到远程故障;	1h0
RB_AUTO_NEG_ABILITY	3	R0	自动协商标志位 0: PHY 不可以工作在自动协商模式下; 1: PHY 可以工作在自动协商模式下;	1h1
RB_LINK_STATUS	2	R0	连接状态标志位 0: 未建立有效连接; 1: 建立有效连接;	1h0
RB_JAB_DET	1	R0	Jabber 检测标志位 0: 未检测到 Jabber; 1: 检测到 Jabber;	1h0
RB_EXTEN_CAP	0	R0	扩展功能标志位 0: 只支持基本功能寄存器中描述的功能 1: 支持扩展寄存器描述的扩展功能	1h1

## 12.5 操作指南及以太网帧

### 12.5.1 MAC 初始化

CH561 的 MAC 部分初始化必须遵循以下步骤:

- (1)、设置寄存器R8\_MISC\_CTRL\_ETH的RB\_MISC\_ETH\_RST位为1, 进行以太网模块硬件复位;
- (2)、设置寄存器R8\_SLP\_WAKE\_CTRL的RB\_SLP\_ETH\_PWR\_DN位为0, 给以太网模块供电;
- (3)、延时几十毫秒;
- (4)、设置寄存器R8\_MISC\_CTRL\_ETH的RB\_MISC\_ETH\_RST位为0, 结束以太网模块硬件复位;
- (5)、延时几十毫秒;
- (6)、进行以太网及其它初始化;

### 12.5.2 发送以太网数据包

以太网数据帧发送过程主要操作步骤如下:

初始化步骤:

- (1)、为发送描述符队列和发送缓冲区分配内存;
- (2)、初始化发送队列;
- (3)、将发送描述符队列的基地址写入寄存器R32\_ETHE\_TXR\_BADR中;
- (4)、设置寄存器R32\_ETHE\_IMR
- (5)、设置MAC地址, 将MAC地址写入寄存器R32\_ETHE\_MAC\_LADR和R32\_ETHE\_MAC\_MDR中;

- (6)、设置寄存器R32\_ETHE\_ITC，选择发送中断触发方式；
- (7)、设置寄存器R32\_ETHE\_APTC，选择发送轮询方式；
- (8)、设置寄存器R32\_ETHE\_MACCR，对MAC进行有效配置，并使能TXDMA通道；
- (9)、向R32\_ETHE\_TXPD写入任意值，触发MAC轮询发送描述符。

发送步骤：

- (1)、发送前软件应该检查是否有空余的发送描述符可用，如果没有则等待，直到描述符可用；
- (2)、将需要发送的数据写入到发送缓冲区中；
- (3)、设置发送描述符；
- (4)、设置寄存器R32\_ETHE\_TXPD，触发MAC轮询发送描述符；
- (5)、等待中断；
- (6)、当中断产生，软件将去查询是否为发送中断，如若ISR[4] = 1，则说明数据包已经成功的被发送；如果ISR[5] = 1，则说明数据包由于冲突原因而被停止发送。

注意：

- (1)、在对发送描述符进行操作时，必须最后操作TXDES0，有下面两种顺序可选：

循序A：

- 设置TXDES2
- 设置TXDES1
- 设置TXDES0

循序B：

- 设置TXDES1
- 设置TXDES2
- 设置TXDES0

- (2)、如果一个发送数据包需要多个发送描述符时，第一个描述符必须被最后设置。

### 12.5.3 接收以太网数据包

以太网数据帧接收过程主要操作步骤如下：

初始化步骤：

- (1)、为接收描述符队列和接收缓冲去分配内存；
- (2)、初始化接收描述符队列；
- (3)、将发送描述符队列的基地址写入寄存器R32\_ETHE\_RXR\_BADR中；
- (4)、设置寄存器R32\_ETHE\_IMR；
- (5)、设置MAC地址，将MAC地址写入寄存器R32\_ETHE\_MAC\_LADR和R32\_ETHE\_MAC\_MADR中；
- (6)、设置多播地址hash表；
- (7)、设置寄存器R32\_ETHE\_ITC，选择接收中断触发方式；
- (8)、设置寄存器R32\_ETHE\_APTC，选择接收轮询方式；
- (9)、设置寄存器R32\_ETHE\_MACCR，对MAC进行有效配置，并使能RXDMA通道；
- (10)、向寄存器R32\_ETHE\_RXPD写入任意值，触发以太网控制器轮询接收描述符。

接收步骤：

- (1)、等待中断；
- (2)、当中断产生，软件查询是否为接收中断，如果RB\_ISR[0] = 1，说明数据包被成功的接收并已经写入到接收缓冲区中，然后软件将拥有此描述符的所有权将数据取出；
- (3)、当软件访问接收描述符后，应当及时释放，将该描述符的拥有权交给MAC所有；
- (4)、如果接收自动轮询被禁止，则软件应该向寄存器R32\_ETHE\_RXPD写入任意值，触发MAC轮询接收描述符。

### 12.5.4 掉电模式的进入和退出

进入掉电模式步骤：

- (1)、设置寄存器R32\_ETHE\_MACCR位0为0；

- (2)、查询寄存器R32\_ETHE\_DMAFIFOS, 等待TX FIFO为空;
  - (3)、设置寄存器R32\_ETHE\_MACCR的位5为0, 停止传输;
  - (4)、设置寄存器R32\_ETHE\_MACCR的位8为0, 停止接收;
  - (5)、查询寄存器R32\_ETHE\_DMAFIFOS, 等待RX FIFO为空;
  - (6)、设置寄存器R32\_ETHE\_MACCR的位0为0, 停止接收;
  - (7)、如果软件需要支持唤醒帧唤醒, 则需要对寄存器R32\_ETHE\_WFBM1、R32\_ETHE\_WFBM2、R32\_ETHE\_WFBM3和R32\_ETHE\_WFBM4写入适当的值;
  - (8)、向寄存器R32\_ETHE\_WOLSR写入0xFFFF FFFF, 清寄存器R32\_ETHE\_WOLSR;
  - (9)、根据需要的唤醒时间来设置寄存器R32\_ETHE\_WOLCR, 同时通过对此寄存器设置来进入掉电模式;
  - (10)、设置寄存器R32\_ETHE\_MACCR的位8为1, 使能接收。
- 退出掉电模式步骤:
- (1)、等待唤醒事件产生;
  - (2)、设置寄存器R32\_ETHE\_MACCR的位1为0, 停止接收;
  - (3)、查询寄存器R32\_ETHE\_WOLSR, 检查是否有唤醒时间产生;
  - (4)、设置寄存器R32\_ETHE\_WOLCR, 退出掉电模式;
  - (5)、设置寄存器 R32\_ETHE\_MACCR 的第 2 位, 对 MAC 进行软件复位并等待此位为 0, 确保复位结束;
  - (6)、重新初始化 MAC。

### 12.5.5 PHY 寄存器操作

PHY寄存器读操作步骤:

- (1)、将PHY寄存器地址写入REGAD(R32\_ETHE\_PHYCR[25:21]);
- (2)、将MIIRD置1(R32\_ETHE\_PHYCR[26]);
- (3)、等待MIIRD为0, MIRDATA(R32\_ETHE\_PHYCR[15:0])即为相应寄存器中的数据。

PHY寄存器写操作步骤:

- (1)、将PHY寄存器地址写入REGAD(R32\_ETHE\_PHYCR[25:21]);
- (2)、将MIWD置1(R32\_ETHE\_PHYCR[27]);
- (3)、将数据写入R32\_ETHE\_PHYWDATA中;
- (4)、等待MIWD(R32\_ETHE\_PHYCR[27])为0。

## 第 13 章 参数

### 13.1 绝对最大值

临界或者超过绝对最大值将可能导致芯片工作不正常甚至损坏。

表 1-绝对最大值参数表

名称	参数说明		最小值	最大值	单位
TA	工作时的环境温度	VCC33=3.3V VCC18=1.8V	-40	85	°C
TS	储存时的环境温度		-55	125	°C
VCC33	电源电压 (VCC33 接电源, GND 接地)		-0.4	4.2	V
VCC18	电源电压 (VCC18 接电源, GND 接地)		-0.4	2.3	V
VI0	输入或者输出引脚上的电压		-0.4	VCC33+0.4	V
VI05	支持 5V 耐压的输入或者输出引脚上的电压		-0.4	5.4V	V

### 13.2 电气参数

测试条件: TA=25°C, VCC33=3.3V、VCC18=1.8V, SYS\_CLK=100MHZ。

表 2-电气参数表

名称	参数说明		最小值	典型值	最大值	单位
VCCxx	电源电压	VCC33	2.7	3.3	3.6	V
		VCC18	1.65	1.8	1.95	
ICC	工作时的总电源电流	VCC33=3.3V	-	115	-	mA
ISLP	低功耗状态的电源电流 I/O 引脚悬空/内部下拉	VCC33=3.3V	0.05	0.3		mA
VIL	低电平输入电压		-0.4	-	0.7	V
VIH	高电平输入电压		2.0	-	VCC33+0.4	V
VOL	低电平输出电压 (4mA 吸入电流)		-	-	0.4	V
VOH	高电平输出电压 (4mA 输出电流)		VCC33-0.4	-	-	V
IUP	内置上拉电阻的输入端的输入电流		20	40	100	uA
IDN	内置下拉电阻的输入端的输入电流		-20	-40	-100	uA
Vpot	电源上电复位的电压门限		1.0	1.5	1.6	V
Vrst	RST#门限电压		1.3	1.5	-	V

### 13.3 功能模块静态电流

测试条件: TA=25°C, VCC33=3.3V、VCC18=1.8V。

表 3-功能模块静态电流表

频率	50M	75M	100M	150M	单位
ADC	0.5	0.6	1.05	1.4	mA
ADC-Phy	0.6	0.6	0.6	0.6	mA
ETH	8.3	12.2	16.2	24.2	mA
ETH-Phy	48.5	48.5	48.5	48.5	mA

GPIO	1.1	1.5	2.2	3.0	mA
TIM0	1.1	1.5	2.0	3.0	mA
TIM1	1.1	1.5	2.0	3.0	mA
TIM2	0.9	1.2	1.6	2.4	mA
TIM3	0.8	1	1.2	2.2	mA
SPI0	1.0	1.5	1.9	2.9	mA
SPI1	0.5	0.7	0.9	1.4	mA
UART0	0.4	0.5	0.6	1.1	mA
UART1	0.5	0.7	0.9	1.4	mA
PLL	4	4	4	4	mA
Core/DMA	31.6	48.4	64.1	91.2	mA

注：上电复位后 ETH-Phy 默认处于关闭状态

## 113.4 时序参数

测试条件：TA=25°C，VCC33=3.3V、VCC18=1.8V，SYS\_CLK=100MHZ。

表 4-时序参数表

名称	参数说明	最小值	典型值	最大值	单位
Trst	外部复位输入 RST#有效信号宽度	20	-	-	nS
Tout	外部复位输入后的复位延时+加载时间	7	11	13	mS
TWAK	从低功耗状态退出的唤醒时间	1	3	10	mS

## 第 14 章 修改记录

版本	日期	说明
V1.00	2017.03.6	初版发行