

概述

CH569 和 CH565 微控制器使用 RISC-V3A 内核，支持 RISC-V 指令的 IMAC 子集。片上采取 128 位数据宽度的 DMA 以支持多个高速外设的高带宽需求，实现大数据量的高速传输。外设包括 USB3.0 超速、USB2.0 高速主机和设备控制器及收发器 PHY、千兆以太网控制器、专用高速 SerDes 控制器及收发器 PHY、SD/EMMC 接口控制器、加解密模块、高速并行接口、数字视频接口 DVP 等，可广泛应用于流媒体、即时存储、超高速 FIFO、通讯延长、安防监控等应用场景。

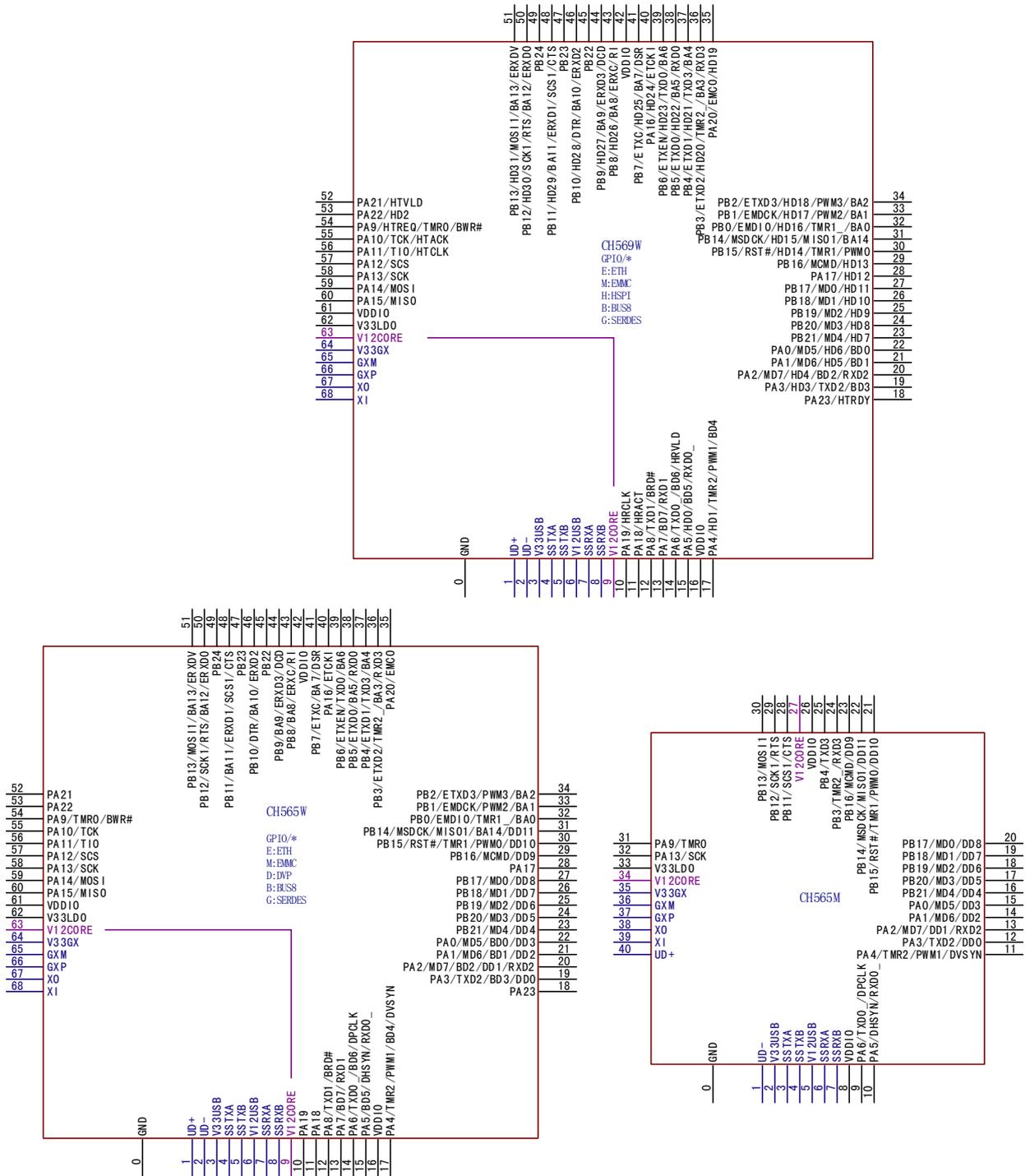
功能

- 内核
 - 支持 RV32IMAC 指令集组合，硬件乘法和除法
 - 快速可编程中断控制器+快速中断响应
 - 静态分支预测、冲突处理机制
 - 低功耗两级流水线
 - 最高 120MHz 系统主频
- 系统存储
 - 448KB 用户应用程序存储区 CodeFlash
 - 32KB 用户数据存储区 DataFlash
 - 24KB 系统引导程序存储区 BootLoader
 - 8KB 系统非易失配置信息存储区 InfoFlash
 - 32/64/96KB 可配置的 128 位宽 SRAM (RAMX)
 - 16KB 的 32 位宽 SRAM (RAMS)
- 电源及低功耗
 - 支持低功耗模式
 - 支持部分 GPIO、USB、以太网信号唤醒
- 定时器及 PWMX
 - 3 组 26 位定时器，
 - 支持定时、计数、信号捕捉、PWM 调制输出
 - TMR1 和 TMR2 支持 DMA
 - 4 组扩展 PWM 输出，占空比可调
- 通用异步串口 UART：
 - 4 组 UART，最高波特率 7.5Mbps，兼容 16C550
 - 内置 FIFO，多个触发级
- 串行外设 SPI
 - 2 组 SPI 接口，支持主从 (Master/Slave) 模式
 - 内置 FIFO，支持 DMA
- 主动并口：8 位数据，15 位地址总线
- 通用 I/O 端口
 - 49 个普通 IO，8 个可设置电平或边沿中断
 - 支持 3.3V 或 2.5V，部分引脚支持复用及映射
- 其他：看门狗、SysTick、调试接口等
- 超速 USB3.0 控制器及收发器（内置 PHY）
 - 支持 USB3.0 Host/Device 模式、OTG 功能
 - 支持控制、批量、中断、实时/同步传输
 - 主机支持 USB3.0 HUB
 - 支持 U1/U2/U3 低功耗状态
- 高速 USB2.0 控制器及收发器（内置 PHY）
 - 支持 USB2.0 Host/Device 模式
 - 支持控制、批量、中断、实时/同步传输
 - 支持数据收发双缓冲
- 千兆以太网控制器 ETH
 - 符合 IEEE 802.3 协议规范
 - 提供 RGMII 和 RMII 接口，连接外置的 PHY
 - 通过 PHY，支持 10/100/1000Mbps 的传输速率
- EMMC 控制器
 - 符合 SD3.0 规范的 UHS-ISDR50 模式并向下兼容
 - 符合 EMMC 卡 4.4 和 4.5.1 规范，兼容 5.0 规范
 - 支持 1/4/8 线数据通讯，最高 96MHz 通讯时钟
- CH569 高速并行接口 HSPI
 - 8/16/32 位数据宽度可配置
 - 内置 FIFO，支持 DMA，双缓冲收发
 - 最快传输速度约为 3.8Gbps (32 位@120MHz)
- CH565 数字视频接口 DVP
 - 可配置 8/10/12 位数据宽度
 - 支持 YUV、RGB、JPEG 压缩数据
- ECEC 加密模块
 - 支持 AES/SM4 算法，8 种组合加解密模式
 - 支持 SRAM/EMMC/HSPI 外设接口数据加解密
- 远距离 SerDes 控制器及收发器（内置 PHY）
 - 8b/10b 编解码，1.2Gbps 高速差分信号通讯
 - 通过一对差分网线 600Mbps 传输距离达 90m
- 芯片 ID 号：唯一 64bit ID 识别号
- 封装：QFN68、QFN40

第 1 章 引脚信息

1.1 引脚排列

图 1-1 CH569 和 CH565 引脚分布



注 1: 0#引脚是 QFN 封装的底板。

1.2 型号对比

表 1-1 CH569 和 CH565 产品资源分配

资源		型号	CH569W	CH565W	CH565M
闪存	用户闪存		448KB	448KB	448KB
	数据闪存		32KB		
	引导闪存		24KB		
SRAM (RAMX+RAMS)			32/64/96KB + 16KB		
通用 I/O			49	49	22
定时器	通用定时器 Timer		3	3	3
	看门狗		√		
	SysTick		√		
PWMX+PWM			4 + 3	4 + 3	2 + 3
UART			4	4	3
SPI (主+从)			2	2	1
主动并口 (BUS8)			√	√	×
千兆以太网 (ETH-GMAC)			√	√	×
数字图像接口 (DVP)			×	√	√
EMMC			√	√	√
AES/SM4 加密			√	√	√
高速并行接口 (HSPI)			√	×	×
超速 USB3.0			√	√	√
高速 USB2.0			√	√	√
高速 SerDes			√	√	√
调试接口			√	√	×

1.3 引脚描述

1.3.1 CH569 引脚定义

表 1-2 CH569 引脚定义

引脚号	引脚名称	类型	主功能 (复位后) / 复用功能及映射	功能描述
0	GND	P	GND	公共接地端 (底板)
1	UD+	USB2.0	DP	USB2.0 信号线 D+
2	UD-	USB2.0	DN	USB2.0 信号线 D-
3	V33USB	P	V33USB	USB PHY 的 3.3V 电源输入, 外接电容 0.1uF
4	SSTXA	USB3.0	SSTXA	USB3.0 超速信号线 TXA
5	SSTXB	USB3.0	SSTXB	USB3.0 超速信号线 TXB
6	V12USB	P	V12USB	USB PHY 的 1.2V 电源输入, 外接电容 0.1uF
7	SSRXA	USB3.0	SSRXA	USB3.0 超速信号线 RXA
8	SSRXB	USB3.0	SSRXB	USB3.0 超速信号线 RXB
9	V12CORE	P	V12CORE	系统的 1.2V 内核电压输入, 需连接到另一个 V12CORE 引脚, 外接电容 0.1uF
10	PA19	I/O	PA19 /HRCLK	PA19: 通用双向数字 I/O 引脚 HRCLK: 高速并行接口接收采样时钟输入
11	PA18	I/O	PA18	PA18: 通用双向数字 I/O 引脚

			/HRACT	HRACT: 高速并行接口的发送请求信号输入
12	PA8	I/O	PA8 /TXD1/BRD#	PA8: 通用双向数字 I/O 引脚 TXD1: UART1 外设的 TXD 引脚输出 BRD#: 主动并口的读控制信号输出, 低有效
13	PA7	I/O	PA7 /BD7/RXD1	PA7: 通用双向数字 I/O 引脚 BD7: 主动并口数据线 7 RXD1: UART1 外设的 RXD 引脚输入
14	PA6	I/O	PA6 /TXD0_/BD6/HRVLD	PA6: 通用双向数字 I/O 引脚 TXD0_: UART0 外设的 TXD 功能引脚映射 BD6: 主动并口数据线 6 HRVLD: 高速并行接口的数据发送状态引脚输入
15	PA5	I/O	PA5 /HD0/BD5/RXD0_	PA5: 通用双向数字 I/O 引脚 HD0: 高速并行接口数据线 0 BD5: 主动并口数据线 5 RXD0_: UART0 外设的 RXD 功能引脚映射
16	VDD10	P	VDD10	I/O 电源 3.3V 或 2.5V 输入, 外接电容 0.1uF
17	PA4	I/O	PA4 /HD1/TMR2/PWM1/BD4	PA4: 通用双向数字 I/O 引脚 HD1: 高速并行接口数据线 1 TMR2: 定时器 2 外设的输入捕获及 PWM 输出引脚 PWM1: PWMX 外设通道 1 输出 BD4: 主动并口数据线 4
18	PA23	I/O	PA23 /HTRDY	PA23: 通用双向数字 I/O 引脚 HTRDY: 高速并行接口检测接收状态引脚输入
19	PA3	I/O	PA3 /HD3/TXD2/BD3	PA3: 通用双向数字 I/O 引脚 HD3: 高速并行接口数据线 3 TXD2: UART2 外设的 TXD 引脚输出 BD3: 主动并口数据线 3
20	PA2	I/O	PA2 /MD7/HD4/BD2/RXD2	PA2: 通用双向数字 I/O 引脚 MD7: EMMC 外设数据线 7 HD4: 高速并行接口数据线 4 BD2: 主动并口数据线 2 RXD2: UART2 外设的 RXD 引脚输入
21	PA1	I/O	PA1 /MD6/HD5/BD1	PA1: 通用双向数字 I/O 引脚 MD6: EMMC 外设数据线 6 HD5: 高速并行接口数据线 5 BD1: 主动并口数据线 1
22	PA0	I/O	PA0 /MD5/HD6/BD0	PA0: 通用双向数字 I/O 引脚 MD5: EMMC 外设数据线 5 HD6: 高速并行接口数据线 6 BD0: 主动并口数据线 0
23	PB21	I/O	PB21 /MD4/HD7	PB21: 通用双向数字 I/O 引脚 MD4: EMMC 外设数据线 4 HD7: 高速并行接口数据线 7
24	PB20	I/O	PB20 /MD3/HD8	PB20: 通用双向数字 I/O 引脚 MD3: EMMC 外设数据线 3 HD8: 高速并行接口数据线 8
25	PB19	I/O	PB19 /MD2/HD9	PB19: 通用双向数字 I/O 引脚 MD2: EMMC 外设数据线 2 HD9: 高速并行接口数据线 9
26	PB18	I/O	PB18	PB18: 通用双向数字 I/O 引脚

			/MD1/HD10	MD1: EMMC 外设数据线 1 HD10: 高速并行接口数据线 10
27	PB17	I/O	PB17 /MDO/HD11	PB17: 通用双向数字 I/O 引脚 MDO: EMMC 外设数据线 0 HD11: 高速并行接口数据线 11
28	PA17	I/O	PA17 /HD12	PA17: 通用双向数字 I/O 引脚 HD12: 高速并行接口数据线 12
29	PB16	I/O	PB16 /MCMD/HD13	PB16: 通用双向数字 I/O 引脚 MCMD: EMMC 外设命令信号线 HD13: 高速并行接口数据线 13
30	PB15	I/O	PB15 /RST#/HD14/TMR1/PWMO	PB15: 通用双向数字 I/O 引脚 RST#: 外部复位输入引脚, 低有效 HD14: 高速并行接口数据线 14 TMR1: 定时器 1 外设的输入捕获及 PWM 输出引脚 PWMO: PWMX 外设通道 0 输出
31	PB14	I/O	PB14 /MSDCK/HD15/MISO1 /BA14	PB14: 通用双向数字 I/O 引脚 MSDCK: EMMC 外设的时钟信号线输出 HD15: 高速并行接口数据线 15 MISO1: SPI1 外设的主机输入从机输出 MISO 引脚 BA14: 主动并口地址线 14
32	PB0	I/O	PB0 /EMD10/HD17/TMR1_ /BA0	PB0: 通用双向数字 I/O 引脚 EMD10: 以太网控制器 SMI 接口数据引脚 HD17: 高速并行接口数据线 17 TMR1_: 定时器 1 外设的输入捕获及 PWM 输出引脚映射 BA0: 主动并口地址线 0
33	PB1	I/O	PB1 /EMDCK/HD17/PWM2/BA1	PB1: 通用双向数字 I/O 引脚 EMDCK: 以太网控制器 SMI 接口时钟输出引脚 HD17: 高速并行接口数据线 17 PWM2: PWMX 外设通道 2 输出 BA1: 主动并口地址线 1
34	PB2	I/O	PB2 /ETXD3/HD18/PWM3/BA2	PB2: 通用双向数字 I/O 引脚 ETXD3: 以太网控制器 MII 接口发送数据线 3 HD18: 高速并行接口数据线 18 PWM3: PWMX 外设通道 3 输出 BA2: 主动并口地址线 2
35	PA20	I/O	PA20 /EMC0/HD19	PA20: 通用双向数字 I/O 引脚 EMC0: 以太网控制器时钟输出引脚 HD19: 高速并行接口数据线 19
36	PB3	I/O	PB3 /ETXD2/HD20/TMR2_ /BA3/RXD3	PB3: 通用双向数字 I/O 引脚 ETXD2: 以太网控制器 MII 接口发送数据线 2 HD20: 高速并行接口数据线 20 TMR2_: 定时器 2 外设的输入捕获及 PWM 输出引脚映射 BA3: 主动并口地址线 3 RXD3: UART3 外设的 RXD 引脚输入
37	PB4	I/O	PB4 /ETXD1/HD21/TXD3 /BA4	PB4: 通用双向数字 I/O 引脚 ETXD1: 以太网控制器 MII 接口发送数据线 1 HD21: 高速并行接口数据线 21 TXD3: UART3 外设的 TXD 引脚输出 BA4: 主动并口地址线 4
38	PB5	I/O	PB5	PB5: 通用双向数字 I/O 引脚

			/ETXD0/HD22/BA5 /RXD0	ETXD0: 以太网控制器 MII 接口发送数据线 0 HD22: 高速并行接口数据线 22 BA5: 主动并口地址线 5 RXD0: UART0 外设的 RXD 引脚输入
39	PB6	I/O	PB6 /ETXEN/HD23/TXD0/BA6	PB6: 通用双向数字 I/O 引脚 ETXEN: 以太网控制器 MII 接口发送数据有效引脚输出 HD23: 高速并行接口数据线 23 TXD0: UART0 外设的 TXD 引脚输出 BA6: 主动并口地址线 6
40	PA16	I/O	PA16 /HD24/ETCK1	PA16: 通用双向数字 I/O 引脚 HD24: 高速并行接口数据线 24 ETCK1: 以太网控制器的 125M 时钟输入引脚
41	PB7	I/O	PB7 /ETXC/HD25/BA7 /DSR	PB7: 通用双向数字 I/O 引脚 ETXC: 以太网控制器 MII 接口发送时钟引脚输出 HD25: 高速并行接口数据线 25 BA7: 主动并口地址线 7 DSR: UART0 外设 MODEM 信号的 DSR 引脚
42	VDD10	P	VDD10	I/O 电源 3.3V 或 2.5V 输入, 外接电容 0.1uF
43	PB8	I/O	PB8 /HD26/BA8/ERXC/RI	PB8: 通用双向数字 I/O 引脚 HD26: 高速并行接口数据线 26 BA8: 主动并口地址线 8 ERXC: 以太网控制器 MII 接口接收时钟输入引脚 RI: UART0 外设 MODEM 信号的 RI 引脚
44	PB9	I/O	PB9 /HD27/BA9/ERXD3/DCD	PB9: 通用双向数字 I/O 引脚 HD27: 高速并行接口数据线 27 BA9: 主动并口地址线 9 ERXD3: 以太网控制器 MII 接口接收数据线 3 DCD: UART0 外设 MODEM 信号的 DCD 引脚
45	PB22	I/O	PB22	PB22: 通用双向数字 I/O 引脚
46	PB10	I/O	PB10 /HD28/DTR/BA10/ERXD2	PB10: 通用双向数字 I/O 引脚 HD28: 高速并行接口数据线 28 DTR: UART0 外设 MODEM 信号的 DTR 引脚 BA10: 主动并口地址线 10 ERXD2: 以太网控制器 MII 接口接收数据线 2
47	PB23	I/O	PB23	PB23: 通用双向数字 I/O 引脚
48	PB11	I/O	PB11 /HD29/BA11/ERXD1 /SCS1/CTS	PB11: 通用双向数字 I/O 引脚 HD29: 高速并行接口数据线 29 BA11: 主动并口地址线 11 ERXD1: 以太网控制器 MII 接口接收数据线 1 SCS1: SPI1 外设的片选信号 SCS 引脚 CTS: UART0 外设 MODEM 信号的 CTS 引脚
49	PB24	I/O	PB24	PB24: 通用双向数字 I/O 引脚
50	PB12	I/O	PB12 /HD30/SCK1/RTS /BA12/ERXD0	PB12: 通用双向数字 I/O 引脚 HD30: 高速并行接口数据线 30 SCK1: SPI1 外设的时钟信号 SCK 引脚 RTS: UART0 外设 MODEM 信号的 RTS 引脚 BA12: 主动并口地址线 12 ERXD0: 以太网控制器 MII 接口接收数据线 0
51	PB13	I/O	PB13 /HD31/MOS11/BA13	PB13: 通用双向数字 I/O 引脚 HD31: 高速并行接口数据线 31

			/ERXDV	MOSI1: SPI1 外设的主机输出从机输入 MOSI 引脚 BA13: 主动并口地址线 13 ERXDV: 以太网控制器 MII 接口接收数据有效引脚输入
52	PA21	I/O	PA21 /HTVLD	PA21: 通用双向数字 I/O 引脚 HTVLD: 高速并行接口的数据发送状态引脚输出
53	PA22	I/O	PA22 /HD2	PA22: 通用双向数字 I/O 引脚 HD2: 高速并行接口数据线 2
54	PA9	I/O	PA9 /HTREQ/TMRO/BWR#	PA9: 通用双向数字 I/O 引脚 HTREQ: 高速并行接口的发送请求信号输出 TMRO: 定时器 0 外设的输入捕获及 PWM 输出引脚 BWR#: 主动并口的写控制信号输出, 低有效
55	PA10	I/O	PA10 /TCK/HTACK	PA10: 通用双向数字 I/O 引脚 TCK: Debug 功能的时钟输入引脚 HTACK: 高速并行接口的接收状态引脚输出
56	PA11	I/O	PA11 /TIO/HTCLK	PA11: 通用双向数字 I/O 引脚 TIO: Debug 功能的数据输入输出硬件 HTCLK: 高速并行接口的通讯时钟引脚输出
57	PA12	I/O	PA12 /SCS	PA12: 通用双向数字 I/O 引脚 SCS: SPI0 外设的片选信号 SCS 引脚
58	PA13	I/O	PA13 /SCK	PA13: 通用双向数字 I/O 引脚 SCK: SPI0 外设的时钟信号 SCK 引脚
59	PA14	I/O	PA14 /MOSI	PA14: 通用双向数字 I/O 引脚 MOSI: SPI0 外设的主机输出从机输入 MOSI 引脚
60	PA15	I/O	PA15 /MISO	PA15: 通用双向数字 I/O 引脚 MISO: SPI0 外设的主机输入从机输出 MISO 引脚
61	VDDIO	P	VDDIO	I/O 电源 3.3V 或 2.5V 输入, 外接电容 0.1uF
62	V33LDO	P	V33LDO	内部 LDO 电源的 3.3V 或 2.5V 输入, 外接电容 0.1uF
63	V12CORE	P	V12CORE	系统的 1.2V 内核电压输出, 外接电容 3.3uF
64	V33GX	P	V33GX	SerDes PHY 的 3.3V 输入, 外接电容 0.1uF
65	GXM	SDP	GXM	SerDes 差分信号线 GXM
66	GXP	SDP	GXP	SerDes 差分信号线 GXP
67	XO	O	XO	外部高速晶体振荡器反相输出端
68	XI	I	XI	外部高速晶体振荡器输入端

1.3.2 CH565 引脚定义

表 1-3 CH565 引脚定义

引脚号		引脚名称	类型	主功能（复位后）/ 复用功能及映射	功能描述
CH565W	CH565M				
0	0	GND	P	GND	公共接地端（底板）
1	40	UD+	USB2.0	DP	USB2.0 信号线 D+
2	1	UD-	USB2.0	DN	USB2.0 信号线 D-
3	2	V33USB	P	V33USB	USB PHY 的 3.3V 电源输入, 外接电容 0.1uF
4	3	SSTXA	USB3.0	SSTXA	USB3.0 超速信号线 TXA
5	4	SSTXB	USB3.0	SSTXB	USB3.0 超速信号线 TXB
6	5	V12USB	P	V12USB	USB PHY 的 1.2V 电源输入, 外接电容 0.1uF
7	6	SSRXA	USB3.0	SSRXA	USB3.0 超速信号线 RXA

8	7	SSRXB	USB3.0	SSRXB	USB3.0 超速信号线 RXB
9	27	V12CORE	P	V12CORE	系统的 1.2V 内核电压输入, 需连接到另一个 V12CORE 引脚, 外接电容 0.1uF
10	-	PA19	I/O	PA19	PA19: 通用双向数字 I/O 引脚
11	-	PA18	I/O	PA18	PA18: 通用双向数字 I/O 引脚
12	-	PA8	I/O	PA8 /TXD1/BRD#	PA8: 通用双向数字 I/O 引脚 TXD1: UART1 外设的 TXD 引脚输出 BRD#: 主动并口的读控制信号输出, 低有效
13	-	PA7	I/O	PA7 /BD7/RXD1	PA7: 通用双向数字 I/O 引脚 BD7: 主动并口数据线 7 RXD1: UART1 外设的 RXD 引脚输入
14	9	PA6	I/O	PA6 /TXD0_/BD6/DPCLK	PA6: 通用双向数字 I/O 引脚 TXD0_: UART0 外设的 TXD 功能引脚映射 BD6: 主动并口数据线 6 DPCLK: 数字视频接口时钟信号输入
15	10	PA5	I/O	PA5 /BD5/DHSYN/RXD0_	PA5: 通用双向数字 I/O 引脚 BD5: 主动并口数据线 5 DHSYN: 数字视频接口行同步信号输入 RXD0_: UART0 外设的 RXD 功能引脚映射
16	8	VDD10	P	VDD10	I/O 电源 3.3V 或 2.5V 输入, 外接电容 0.1uF
17	11	PA4	I/O	PA4 /TMR2/PWM1/BD4 /DVSYN	PA4: 通用双向数字 I/O 引脚 TMR2: 定时器 2 外设的输入捕获及 PWM 输出引脚 PWM1: PWMX 外设通道 1 输出 BD4: 主动并口数据线 4 DVSYN: 数字视频接口帧同步信号输入
18	-	PA23	I/O	PA23	PA23: 通用双向数字 I/O 引脚
19	12	PA3	I/O	PA3 /TXD2/BD3/DD0	PA3: 通用双向数字 I/O 引脚 TXD2: UART2 外设的 TXD 引脚输出 BD3: 主动并口数据线 3 DD0: 数字视频接口数据线 0
20	13	PA2	I/O	PA2 /MD7/BD2/DD1 /RXD2	PA2: 通用双向数字 I/O 引脚 MD7: EMMC 外设数据线 7 BD2: 主动并口数据线 2 DD1: 数字视频接口数据线 1 RXD2: UART2 的 RXD 引脚输入
21	14	PA1	I/O	PA1 /MD6/BD1/DD2	PA1: 通用双向数字 I/O 引脚 MD6: EMMC 外设数据线 6 BD1: 主动并口数据线 1 DD2: 数字视频接口数据线 2
22	15	PA0	I/O	PA0 /MD5/BDO/DD3	PA0: 通用双向数字 I/O 引脚 MD5: EMMC 外设数据线 5 BDO: 主动并口数据线 0 DD3: 数字视频接口数据线 3
23	16	PB21	I/O	PB21 /MD4/DD4	PB21: 通用双向数字 I/O 引脚 MD4: EMMC 外设数据线 4 DD4: 数字视频接口数据线 4
24	17	PB20	I/O	PB20 /MD3/DD5	PB20: 通用双向数字 I/O 引脚 MD3: EMMC 外设数据线 3 DD5: 数字视频接口数据线 5
25	18	PB19	I/O	PB19	PB19: 通用双向数字 I/O 引脚

				/MD2/DD6	MD2: EMMC 外设数据线 2 DD6: 数字视频接口数据线 6
26	19	PB18	I/O	PB18 /MD1/DD7	PB18: 通用双向数字 I/O 引脚 MD1: EMMC 外设数据线 1 DD7: 数字视频接口数据线 7
27	20	PB17	I/O	PB17 /MD0/DD8	PB17: 通用双向数字 I/O 引脚 MD0: EMMC 外设数据线 0 DD8: 数字视频接口数据线 8
28	-	PA17	I/O	PA17	PA17: 通用双向数字 I/O 引脚
29	23	PB16	I/O	PB16 /MCMD/DD9	PB16: 通用双向数字 I/O 引脚 MCMD: EMMC 外设命令信号线 DD9: 数字视频接口数据线 9
30	21	PB15	I/O	PB15 /RST#/TMR1/PWMO /DD10	PB15: 通用双向数字 I/O 引脚 RST#: 外部复位输入引脚, 低有效 TMR1: 定时器 1 外设的输入捕获及 PWM 输出引脚 PWMO: PWM 外设通道 0 输出 DD10: 数字视频接口数据线 10
31	22	PB14	I/O	PB14 /MSDCK/MISO1 /BA14/DD11	PB14: 通用双向数字 I/O 引脚 MSDCK: EMMC 外设的时钟信号线输出 MISO1: SPI1 外设的主机输入从机输出 MISO 引脚 BA14: 主动并口地址线 14 DD10: 数字视频接口数据线 11
32	-	PB0	I/O	PB0 /EMD10/TMR1_ /BA0	PB0: 通用双向数字 I/O 引脚 EMD10: 以太网控制器 SMI 接口数据引脚 TMR1_: 定时器 1 外设的输入捕获及 PWM 输出引脚映射 BA0: 主动并口地址线 0
33	-	PB1	I/O	PB1 /EMDCK/PWM2/BA1	PB1: 通用双向数字 I/O 引脚 EMDCK: 以太网控制器 SMI 接口时钟输出引脚 PWM2: PWM 外设通道 2 输出 BA1: 主动并口地址线 1
34	-	PB2	I/O	PB2 /ETXD3/PWM3/BA2	PB2: 通用双向数字 I/O 引脚 ETXD3: 以太网控制器 MII 接口发送数据线 3 PWM3: PWM 外设通道 3 输出 BA2: 主动并口地址线 2
35	-	PA20	I/O	PA20 /EMC0	PA20: 通用双向数字 I/O 引脚 EMC0: 以太网控制器时钟输出引脚
36	24	PB3	I/O	PB3 /ETXD2/TMR2_ /BA3/RXD3	PB3: 通用双向数字 I/O 引脚 ETXD2: 以太网控制器 MII 接口发送数据线 2 TMR2_: 定时器 2 外设的输入捕获及 PWM 输出引脚映射 BA3: 主动并口地址线 3 RXD3: UART3 外设的 RXD 引脚输入
37	25	PB4	I/O	PB4 /ETXD1/TXD3/BA4	PB4: 通用双向数字 I/O 引脚 ETXD1: 以太网控制器 MII 接口发送数据线 1 TXD3: UART3 的 TXD 引脚输出 BA4: 主动并口地址线 4
38	-	PB5	I/O	PB5 /ETXD0/BA5/RXD0	PB5: 通用双向数字 I/O 引脚 ETXD0: 以太网控制器 MII 接口发送数据线 0 BA5: 主动并口地址线 5 RXD0: UART0 外设的 RXD 引脚输入
39	-	PB6	I/O	PB6	PB6: 通用双向数字 I/O 引脚

				/ETXEN/TXD0/BA6	ETXEN: 以太网控制器 MII 接口发送数据有效引脚输出 TXD0: UART0 外设的 TXD 引脚输出 BA6: 主动并口地址线 6
40	-	PA16	I/O	PA16 /ETCK1	PA16: 通用双向数字 I/O 引脚 ETCK1: 以太网控制器的 125M 时钟输入引脚
41	-	PB7	I/O	PB7 /ETXC/BA7/DSR	PB7: 通用双向数字 I/O 引脚 ETXC: 以太网控制器 MII 接口发送时钟引脚输出 BA7: 主动并口地址线 7 DSR: UART0 外设 MODEM 信号的 DSR 引脚
42	26	VDD10	P	VDD10	I/O 电源 3.3V 或 2.5V 输入, 外接电容 0.1uF
43	-	PB8	I/O	PB8 /BA8/ERXC/RI	PB8: 通用双向数字 I/O 引脚 BA8: 主动并口地址线 8 ERXC: 以太网控制器 MII 接口接收时钟输入引脚 RI: UART0 外设 MODEM 信号的 RI 引脚
44	-	PB9	I/O	PB9 /BA9/ERXD3/DCD	PB9: 通用双向数字 I/O 引脚 BA9: 主动并口地址线 9 ERXD3: 以太网控制器 MII 接口接收数据线 3 DCD: UART0 外设 MODEM 信号的 DCD 引脚
45	-	PB22	I/O	PB22	PB22: 通用双向数字 I/O 引脚
46	-	PB10	I/O	PB10 /DTR/BA10/ERXD2	PB10: 通用双向数字 I/O 引脚 DTR: UART0 外设 MODEM 信号的 DTR 引脚 BA10: 主动并口地址线 10 ERXD2: 以太网控制器 MII 接口接收数据线 2
47	-	PB23	I/O	PB23	PB23: 通用双向数字 I/O 引脚
48	28	PB11	I/O	PB11 /BA11/ERXD1 /SCS1/CTS	PB11: 通用双向数字 I/O 引脚 BA11: 主动并口地址线 11 ERXD1: 以太网控制器 MII 接口接收数据线 1 SCS1: SPI1 外设的片选信号 SCS 引脚 CTS: UART0 外设 MODEM 信号的 CTS 引脚
49	-	PB24	I/O	PB24	PB24: 通用双向数字 I/O 引脚
50	29	PB12	I/O	PB12 /SCK1/RTS /BA12/ERXD0	PB12: 通用双向数字 I/O 引脚 SCK1: SPI1 外设的时钟信号 SCK 引脚 RTS: UART0 外设 MODEM 信号的 RTS 引脚 BA12: 主动并口地址线 12 ERXD0: 以太网控制器 MII 接口接收数据线 0
51	30	PB13	I/O	PB13 /MOSI1/BA13 /ERXDV	PB13: 通用双向数字 I/O 引脚 MOSI1: SPI1 外设的主机输出从机输入 MOSI 引脚 BA13: 主动并口地址线 13 ERXDV: 以太网控制器 MII 接口接收数据有效引脚输入
52	-	PA21	I/O	PA21	PA21: 通用双向数字 I/O 引脚
53	-	PA22	I/O	PA22	PA22: 通用双向数字 I/O 引脚
54	31	PA9	I/O	PA9 /TMRO/BWR#	PA9: 通用双向数字 I/O 引脚 TMRO: 定时器 0 外设的输入捕获及 PWM 输出引脚 BWR#: 主动并口的写控制信号输出, 低有效
55	-	PA10	I/O	PA10 /TCK	PA10: 通用双向数字 I/O 引脚 TCK: Debug 功能的时钟输入引脚
56	-	PA11	I/O	PA11 /TIO	PA11: 通用双向数字 I/O 引脚 TIO: Debug 功能的数据输入输出硬件
57	-	PA12	I/O	PA12 /SCS	PA12: 通用双向数字 I/O 引脚 SCS: SPI0 外设的片选信号 SCS 引脚

58	32	PA13	I/O	PA13 /SCK	PA13: 通用双向数字 I/O 引脚 SCK: SPI0 外设的时钟信号 SCK 引脚
59	-	PA14	I/O	PA14 /MOSI	PA14: 通用双向数字 I/O 引脚 MOSI: SPI0 外设的主机输出从机输入 MOSI 引脚
60	-	PA15	I/O	PA15 /MISO	PA15: 通用双向数字 I/O 引脚 MISO: SPI0 外设的主机输入从机输出 MISO 引脚
61		VDDI0	P	VDDI0	I/O 电源 3.3V 或 2.5V 输入, 外接电容 0.1uF
62	33	V33LDO	P	V33LDO	内部 LDO 电源的 3.3V 或 2.5V 输入, 外接电容 0.1uF
63	34	V12CORE	P	V12CORE	系统的 1.2V 内核电压输出, 外接电容 3.3uF
64	35	V33GX	P	V33GX	SerDes PHY 的 3.3V 输入, 外接电容 0.1uF
65	36	GXM	SDP	GXM	SerDes 差分信号线 GXM
66	37	GXP	SDP	GXP	SerDes 差分信号线 GXP
67	38	XO	0	XO	外部高速晶体振荡器反相输出端
68	39	XI	I	XI	外部高速晶体振荡器输入端

注:

- (1) I: 输入; 0: 输出; I/O: 输入输出。
- (2) P: 电源。
- (3) USB: USB 信号。
- (4) SDP: SerDes PHY 信号。

第 2 章 系统结构及存储器

2.1 系统结构

图 2-1 系统框图

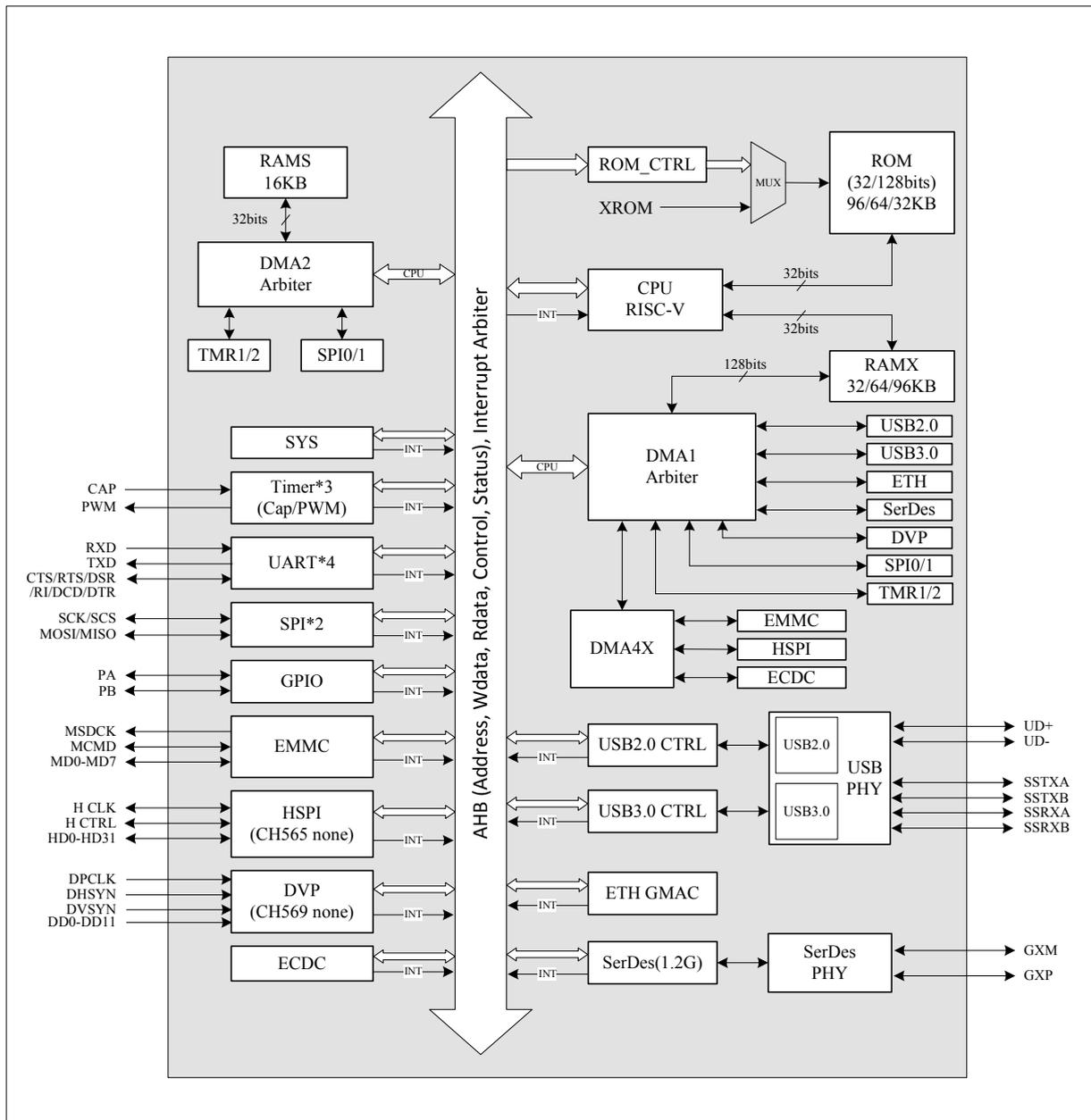


图 2-1 是系统内部结构框图。在 32bit 系统总线（AHB）上挂载了 RISC-V 内核，DMA1/DMA2 仲裁控制器、ROM 控制模块、SRAM 以及各种外设模块。内核通过系统总线访问各个外设或模块，并接收外部中断信号触发中断服务。系统默认 32K 字节大小的 ROM 代码全速零等待运行，保障指令取址速度和系统主频一致。用户可以使用的 SRAM 区域分为 RAMS 和 RAMX 区域，其中 RAMX 大小可以通过接口配置，多余区域将用于 ROM 代码的 SRAM 映射范围。

- RAMS: 共 16K 字节大小，32 位宽访问的 SRAM，寻址范围 0x20000000~0x20003FFF
- RAMX: 可配置 32K/64K/96K 字节大小，128 位宽访问的高速 SRAM，寻址范围 0x20020000~0x20037FFF
- DMA1: 建立了 RAMX 和 CPU、普通外设、高速外设之间的访问。高速外设包括：EMMC、HSPI、ECDC、USB2.0、USB3.0、ETH、SerDes、DVP。

- DMA2: 建立了 RAMS 和 CPU、普通外设之间的访问。普通外设包括 Timer1、Timer2、SPI0、SPI1

注 1: CH569 微控制器系统没有 DVP 外设模块, CH565 微控制器系统没有 HSPI 模块, 其他资源一致。

注 2: 高速外设只能访问 RAMX 区域, 应用代码需要注意高速外设寄存器配置的 DMA 地址范围。

2.2 存储器映射

系统总线的寻址空间主要包括: 非易失存储区 (CODE)、静态存储区 (RAMS 和 RAMX)、外设、外部总线等几个区域。详细分配参考图 2-2 所示。

图 2-2 存储器映射

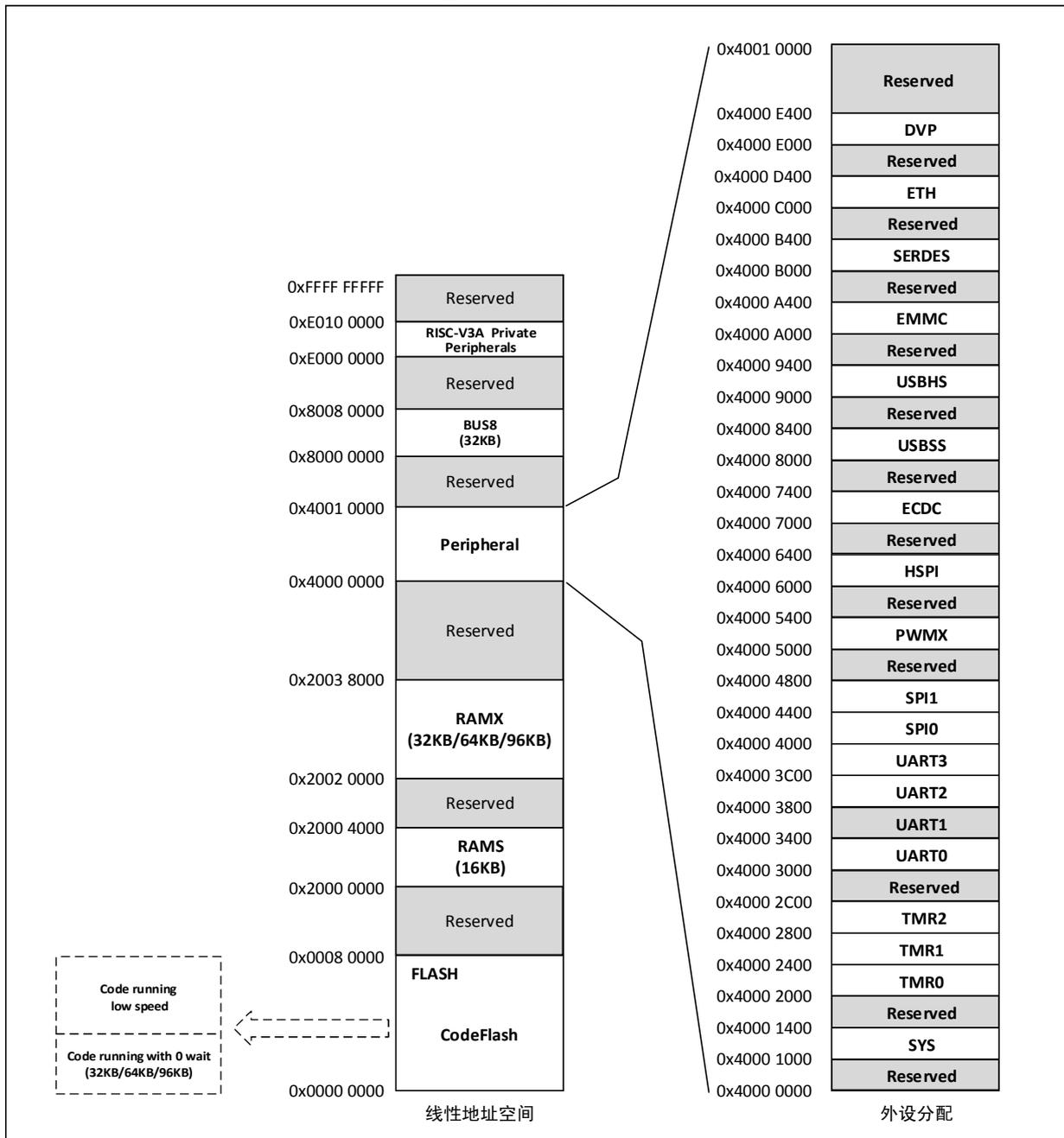


表 2-1 存储器映射区域地址

地址范围	用途	描述
------	----	----

0x00000000-0x0007FFFF	片上非易失性存储器	Flash 存储器 (512KB)
0x00080000-0x1FFFFFFF	保留	-
0x20000000-0x20003FFF	RAMS, 通常用于存储数据	16KB 通用 SRAM
0x20004000-0x2001FFFF	保留	-
0x20020000-0x20037FFF	RAMX, 用于高速外设的 DMA 数据存取	32/64/96KB 可配置 SRAM
0x20038000-0x3FFFFFFF	保留	-
0x40000000-0x4000FFFF	各种功能模块或外设	各种外设地址空间
0x40010000-0x7FFFFFFF	保留	-
0x80000000-0x80007FFF	外部总线	32KB
0x80008000-0xFFFFFFFF	保留	-

2.2.1 片上非易失存储区映射

非易失存储区主要包括 CodeFlash、DataFlash、BootLoader、InfoFlash 几个部分, 其中 InfoFlash 存储了厂商数据, 一般在出厂前有厂家写入配置信息, 见表 2-2 说明。

表 2-2 非易失性信息描述

字段	名称	描述	默认值
[31:30]	USER_MEM	系统的 RAMX/ROM 容量重定义配置。 00: RAMX 32KB + ROM 96KB 01: RAMX 64KB + ROM 64KB 1x: RAMX 96KB + ROM 32KB 注: 上述 ROM 空间支持全速零等待运行, 若 code 大小超出上述 ROM 空间, 则超出部分的 code 将以约 1/8 的低速运行。	10b
29	LOCKUP_RST_EN	内核 LOCKUP 复位系统使能。 1: 内核产生 LOCKUP 时, 复位系统; 0: 内核产生 LOCKUP 时, 不复位系统。	0
28	RESERVED2	保留。	0
[27:12]	RESERVED2	保留。	FFFFh
[11:10]	RESERVED3	保留。	00b
[9:8]	RESERVED4	保留。	10b
7	CODE_READ_EN	外部编程器读 FLASH 使能。 1: 允许外部编程器读取 FLASH (代码开放); 0: 禁止外部编程器读取 FLASH (代码保护)。	0
6	BOOT_EN	Boot loader 功能使能。 1: 使能 BOOT 引导程序; 0: 禁止 BOOT 引导程序。	1
5	DEBUG_EN	调试接口使能。 1: 使能调试接口功能; 0: 禁用调试接口功能。 注: 使用调试功能时, 必须要禁止外部复位, 否则调试功能无法正常使用。	0
4	RESET_EN	外部复位使能。 1: 允许来自 PB15 的外部复位输入, PB15 用作系统外部复位引脚; 0: 禁用来自 PB15 的外部复位输入, PB15 用作普通 GPIO。	0
[3:0]	RESERVED5	保留。	0101b

2.2.2 片上 DATA 区域映射

表 2-3 片上静态存储区分配

地址范围	描述	大小
0x20000000-0x20003FFF	RAMS 区, CPU 存取变量及低速外设存取 DMA 数据。	16KB
0x20020000-0x20027FFF	USER_MEM=00b 时, 高速外设 DMA 寻址的 RAMX 区	32KB
0x20020000-0x2002FFFF	USER_MEM=01b 时, 高速外设 DMA 寻址的 RAMX 区	64KB
0x20020000-0x20037FFF	USER_MEM=1xb 时, 高速外设 DMA 寻址的 RAMX 区	96KB

系统内部含有丰富的 SRAM 资源供 CPU 存取变量及堆栈数据, 同时满足各个外设的 DMA 数据存取需求。内部的 SRAM 资源主要分成两类: 一类主要用于 CPU 存取变量和低速外设 DMA 数据存储的 RAMS; 另一类主要用于高速外设 DMA 数据存储的 RAMX (SRAM1+SRAM2)。CPU、SPI、TMR 通过正确设置 DMA 地址可以访问 RAMS 和 RAMX 所有资源, 但是, 其他所有的高速外设的 DMA 只能使用 RAMX 资源, 不再和 CPU 访存行为产生竞争。应用代码在使用不同外设时需要注意数据存储的地址分配。

RAMS 区域访问位宽 32bits, 起始地址 0x20000000, 总共 16KB; RAMX 区域访问位宽 128bits, 起始地址 0x20020000, 大小取决于用户配置字 USER_MEM 的设定, 支持 32KB/64KB/96KB 三种容量弹性选择。

2.2.3 唯一 ID 号及 Flash 操作

每个芯片出厂时都具有唯一的 ID 号, 即芯片身份识别号。该 ID 数据及其校验和共 8 字节, 存储于芯片内部只读区域中, 具体操作请参考例子程序。

关于 DataFlash 和 CodeFlash 区域的操作, 将提供库函数, 请参考例子程序。

2.2.4 外设地址分配

系统提供了 18 个外设和 1 个外部总线接口, 每个外设占用一定的地址空间, 外设寄存器的实际访问地址为: 基地址+偏移地址。在后续章节中, 寄存器的地址按基地址和偏移地址分开说明。

下表为后续章节寄存器描述中“访问”的解释说明:

寄存器位属性	属性描述
RF	只读属性, 读取值为固定值, 不受复位影响。
RO	只读属性, 由硬件改变。
RZ	只读属性, 读操作后自动清 0。
WO	只写属性 (不可读, 读取值不确定)
WA	只写属性, 安全模式下可写入。
WZ	只写属性, 写操作后自动位清 0。
RW	可读, 可写。
RWA	可读, 安全模式下可写入。
RW1	可读, 写 1 有效, 写 0 无效。
RW0	可读, 写 0 有效, 写 1 无效。
RW1T	可读, 写 0 无效, 写 1 翻转。

第 3 章 系统控制

3.1 电源控制

系统的电源控制设计为区域划分单独供电。所以外部需要提供多个 3.3V 电压源，以保证内核、GPIO、内部 LDO、内置 USB PHY 以及内置的高速 SerDes PHY 的正常运行。

- VDDIO: 系统 IO 端口电源输入，多组 VDDIO 共同为所有 IO 端口供电。
- V33LDO: 系统内部 LDO 电压调整器模块的电源输入。
- V12CORE: 内核的 1.2V 电压输出和输入。
- V33GX: SerDes 模块收发器供电端。
- V33USB、V12USB: USB3.0 和 USB2.0 模块收发器供电端。

系统复位后，微控制器处于正常运行状态。当 CPU 不需要继续运行，或者某些功能模块不需要使用时，可以关闭这些模块的时钟或独立电源，以降低功耗。

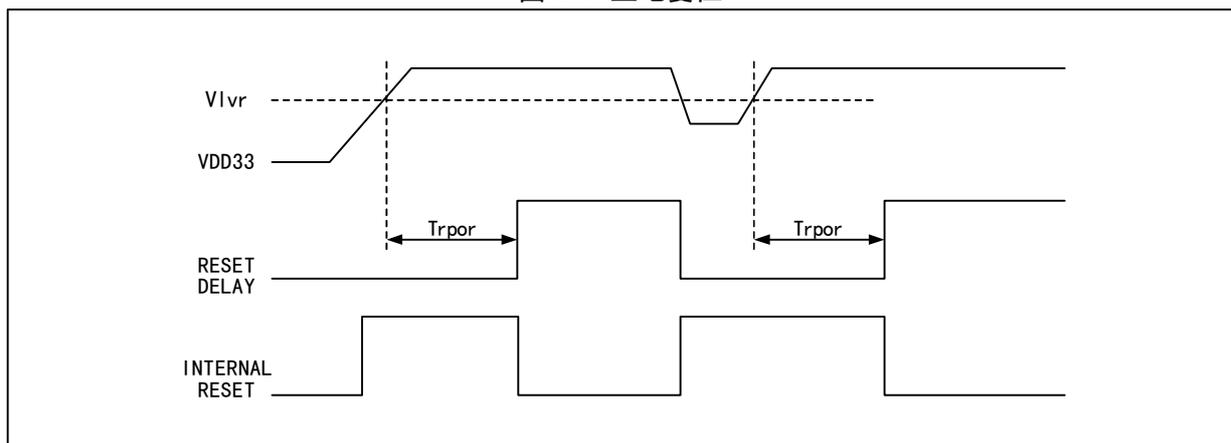
3.2 复位控制

系统提供了 4 种复位形式：上电复位、手工复位、软件复位和看门狗复位。仅上电复位会导致系统重新加载配置信息，并重新加载程序代码到 SRAM 中。寄存器 R8_RST_BOOT_STAT 记录最近一次复位的复位源。寄存器 R8_GLOB_RESET_KEEP 只在上电复位时被复位，而不受其它复位形式影响。

3.2.1 上电复位

当电源上电时，芯片内部 POR 模块会产生上电复位时序，并延时以等待电源稳定。在运行过程中，当电源电压低于 V_{lvr} 时，芯片内部 LVR 模块会产生低压复位直到电压回升，并延时以等待电源稳定。3-1 为上电复位过程以及低压复位过程。

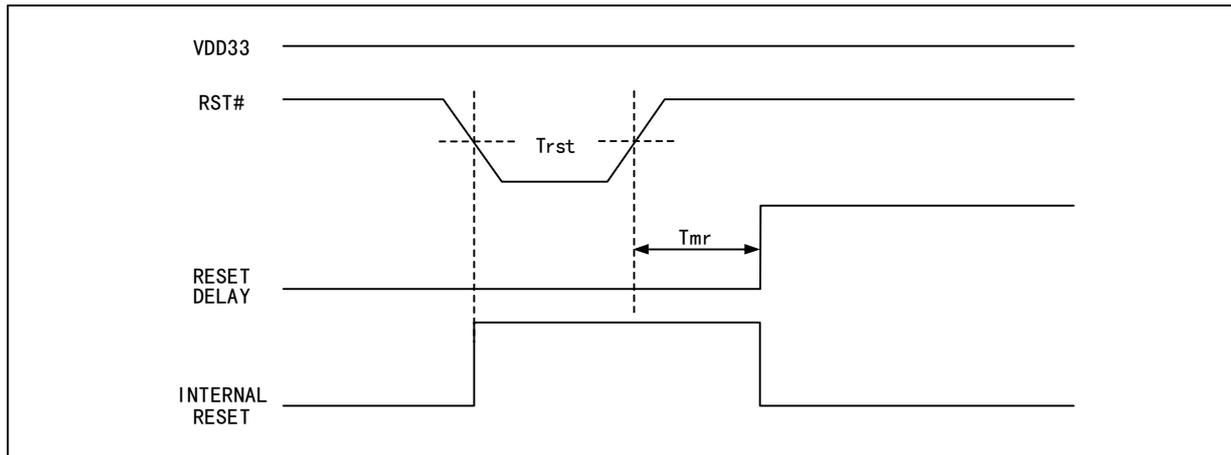
图 3-1 上电复位



3.2.2 手工复位

外部手工复位由外部加到 RST#引脚的低电平产生。当外部的低电平复位信号持续时间大于最小复位脉冲宽度 (T_{rst}) 时，将触发系统复位，并经过 T_m 时间完成复位过程。

图 3-2 外部复位



3.2.3 软件复位

系统提供了两种内部软件复位方式，以便在某些特定情况下，不需要外部干预进行软件复位。

- 1) 设置复位配置寄存器 R8_RST_WDOG_CTRL 的 RB_SOFTWARE_RESET 位为 1 即可，该位将自动清 0。
- 2) 通过配置内核的中断配置寄存器 PFIC_CFGR 的 SYSRESET 位置 1 复位系统，具体参考对应章节。

3.2.4 看门狗复位

看门狗功能是基于一个 8 位的递增计数器，计数时钟时基为 $F_{sys}/524288$ 。当开启了看门狗超时复位功能 (RB_WDOG_RST_EN=1)，一旦此计数器溢出会触发系统复位。

3.2.5 复位特性

复位参数请参考 19 章。

3.3 时钟控制

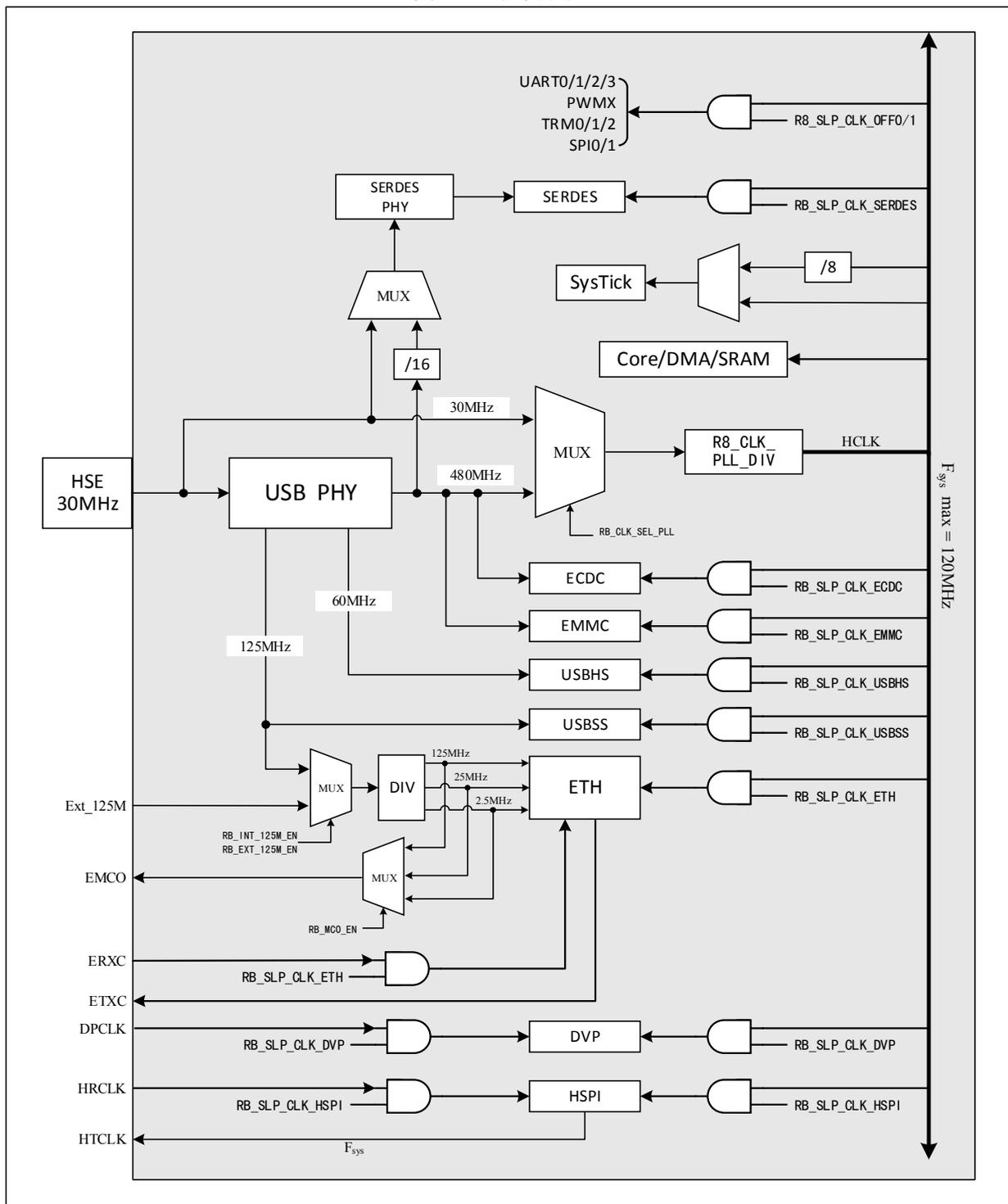
系统工作需要外部提供 30MHz 的高速时钟信号 HSE，包括外部晶体/陶瓷谐振器产生或者外部高速时钟送入。

如图 3-3 所示，30MHz 时钟在系统内部被送入 USB PHY 模块，此模块将产生几个时钟频率，包括 125MHz 时钟用于 USB3.0 控制器及以太网模块内部时钟来源，60MHz 时钟用于 USB2.0 控制器，480MHz 时钟作为 PLL 时钟频率 F_{pll} 。

- 系统总线时钟 HCLK 来源于 F_{pll} 分频或者 HSE 分频，由 R8_CLK_CFG_CTRL 和 R8_CLK_PLL_DIV 寄存器配置决定，其频率大小为 $F_{sys}(T_{sys})$ 。HCLK 时钟作用于系统所有寄存器和 SRAM 的访问、各外设模块数字逻辑运行、DMA、内核各单元运行等部分。
- 高速并行接口 HSPI 模块，使用 HCLK 时钟实现数据发送，输出 HTCLK 频率等于 F_{sys} 。
- 高速 SerDes PHY 的参考时钟 F_{SDSP} 来源 HSE 或者 F_{pll} 的 16 分频， F_{SDSP} 经倍频产生 SerDes 控制器所需的时钟 F_{SDS} 供 SerDes 收发数据。
- 以太网控制器所需的 125MHz/25MHz/2.5MHz 三个时钟来源于内部 125MHz 时钟或者外部 ETCKI 引脚输入的 125MHz 时钟，由 R8_CLK_MOD_AUX 寄存器配置决定。同时软件可配置由 EMCO 引脚输出三个时钟用于连接外部的以太网 PHY。
- 各外设模块时钟可以独立进行打开或关闭。为降低芯片功耗，可以关闭不使用的功能模块时钟。

具体参考 R8_SLP_CLK_OFF0 和 R8_SLP_CLK_OFF1 寄存器的描述。

图 3-3 时钟树结构



3.4 系统时钟配置

系统默认使用 (HSE/2) 作为系统时钟，即 15MHz。应用代码通过配置寄存器 R8_CLK_CFG_CTRL 和 R8_CLK_PLL_DIV 值，可以得到 2MHz~120MHz 范围的 F_{sys} 。

当 RB_CLK_SEL_PLL=1 时，系统时钟为 480MHz 分频，分频系数为 R8_CLK_PLL_DIV。

当 RB_CLK_SEL_PLL=0 时，系统时钟为 30MHz 分频，分频系数为 R8_CLK_PLL_DIV。

在配置系统时钟时，需注意以下事项：

- 1) 在配置 R8_CLK_CFG_CTRL 时，位[7:6]必须为 01b，否则该寄存器配置无效。
- 2) 在配置 R8_CLK_PLL_DIV 时，位[7:6]必须为 10b，否则该寄存器配置无效。
- 3) 系统时钟源在从 30MHz 切换到 480MHz 之前，必须先将分频系数设置正确，如果直接切换时钟源，高频时钟可能使系统不可预知的异常。

3.5 低功耗模式及唤醒

系统复位结束后，微控制器处于正常工作模式，当 MCU 不需要运行时，可以进入低功耗模式来节省功耗。系统提供 3 档低功耗模式：空闲模式、暂停模式和睡眠模式。

- 空闲模式：即所有外设保持正常运行，内核停止运行，时钟系统及外设模块正常运转。当检测到唤醒事件后，可以立即唤醒，最短唤醒时间。如果有不需要使用的外设，可以关闭相应的外设时钟控制位，降低部分功耗。
- 暂停模式：即在空闲模式的基础上，PLL 停止工作，时钟系统不再运转，相关外设都停止。可以降低大部分功耗。检测到唤醒事件后，首先唤醒系统时钟，然后再唤醒内核，系统重新开始运行。
- 睡眠模式：在暂停模式基础上，停止晶振振荡，达到最低功耗。此模式唤醒需要等待晶振稳定才能正常工作，所有需要较长的唤醒时间。

低功耗模式下，系统支持部分 GPIO 端口唤醒、高速 USB2.0 唤醒、超速 USB3.0 唤醒及以太网唤醒等四种唤醒方式，参考 R8_SLP_WAKE_CTRL 寄存器。下表详细描述了三种低功耗模式的特征及其唤醒途径：

表 3-1 低功耗模式

模式	特征	进入条件	唤醒事件
空闲模式	外设均正常运行，内核停止运行（内核时钟停止），时钟系统正常运转，但可以通过外设时钟控制位选择关闭各外设的时钟。	设置内核 PFIC_SCTLR 寄存器的 SleepDeep 字段为 0。 设置唤醒条件后执行 __WFI() 或 __WFE()。	所有唤醒源
暂停模式	内核停止运行，PLL 停止工作，时钟系统不再运转，相关外设都停止。	设置内核 PFIC_SCTLR 寄存器的 SleepDeep 字段为 1。RB_SLP_USBHS_PWRDN 置 1。 设置唤醒条件后执行 __WFI() 或 __WFE()。	所有唤醒源
睡眠模式	内核停止运行，PLL 停止工作，外部晶振停振。达到系统最低功耗。	设置内核 PFIC_SCTLR 寄存器的 SleepDeep 字段为 1。RB_SLP_USBHS_PWRDN 置 1。USB3.0 控制器进入低功耗（参考例程）。 设置唤醒条件后执行 __WFI() 或 __WFE()。	所有唤醒源

3.6 寄存器描述

系统控制相关寄存器基地址：0x40001000

表 3-2 系统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	0x00
R8_CHIP_ID	0x01	芯片 ID 寄存器	-
R8_SAFE_ACCESS_ID	0x02	安全访问 ID 寄存器	0x02
R8_WDOG_COUNT	0x03	看门狗计数器寄存器	0x00

R8_GLOB_ROM_CFG	0x04	ROM 配置寄存器	0x80
R8_RST_BOOT_STAT	0x05	BOOT 状态寄存器	0xXX
R8_RST_WDOG_CTRL	0x06	复位配置寄存器	0x00
R8_GLOB_RESET_KEEP	0x07	复位保持寄存器	0x00
R8_CLK_PLL_DIV	0x08	PLL 输出时钟分频寄存器	0x42
R8_CLK_CFG_CTRL	0x0A	时钟配置寄存器	0x80
R8_CLK_MOD_AUX	0x0B	时钟辅助寄存器	0x00
R8_SLP_CLK_OFF0	0x0C	睡眠控制寄存器 0	0x00
R8_SLP_CLK_OFF1	0x0D	睡眠控制寄存器 1	0x00
R8_SLP_WAKE_CTRL	0x0E	唤醒控制寄存器	0x00
R8_SLP_POWER_CTRL	0x0F	低功耗电源管理寄存器	0x00
R16_SERD_ANA_CFG1	0x20	SerDes PHY 模拟参数配置寄存器 1	0x005A
R32_SERD_ANA_CFG2	0x24	SerDes PHY 模拟参数配置寄存器 2	0x00423015

安全访问标记寄存器 (R8_SAFE_ACCESS_SIG)

位	名称	访问	描述	复位值
[7:0]	R8_SAFE_ACCESS_SIG	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 16 个主时钟周期 (Tsys), 超过则自动保护。可以写入其它任意值强制直接退出安全访问模式, 回到保护状态。	0
[6:4]	RB_SAFE_ACC_TIMER	RO	当前安全访问时间计数。	0
[1:0]	RB_SAFE_ACC_MODE	RO	当前安全访问模式状态: 11: 安全模式, 可访问属性 RWA 寄存器; 其他: 非安全模式, 禁止保护寄存器的写操作。	0

芯片 ID 寄存器 (R8_CHIP_ID)

位	名称	访问	描述	复位值
[7:0]	R8_CHIP_ID	RF	用于识别芯片号。	-

安全访问 ID 寄存器 (R8_SAFE_ACCESS_ID)

位	名称	访问	描述	复位值
[7:0]	R8_SAFE_ACCESS_ID	RF	固定值 02h。	02h

看门狗计数器寄存器 (R8_WDOG_COUNT)

位	名称	访问	描述	复位值
[7:0]	R8_WDOG_COUNT	RW	可预置初值的看门狗计数器, 一直自动递增, 可从 0xFF 循环到 0x00 再继续。 计数时基 = Fsys/524288。	0

ROM 配置寄存器 (R8_GLOB_ROM_CFG)

位	名称	访问	描述	复位值
[7:5]	Reserved	RWA	保留。[7:6] 必须写 10b, 读为 0。	0
4	RB_ROM_CODE_OFS	RWA	配置用户代码在 FlashROM 中的起始偏移地址,	0

			该值仅在上电复位时才能清0，不受其他复位影响。 1: 0x04000; 0: 0x00000。	
3	RB_ROM_CODE_WE	RWA	Flash ROM 代码和数据区擦除/写使能。 1: 可编程/擦除; 0: 写保护。	0
2	RB_ROM_DATA_WE	RWA	Flash ROM 数据区擦除/写使能。 1: 可编程/擦除; 0: 写保护。	0
1	RB_CODE_RAM_WE	RWA	代码 SRAM 区写使能位。 1: 写使能; 0: 写保护。	0
0	RB_ROM_EXT_RE	RO	外部编程器读 Flash ROM 使能。 1: 读使能; 0: 读保护。	0

BOOT 状态寄存器 (R8_RST_BOOT_STAT)

位	名称	访问	描述	复位值
[7:6]	Reserved	RO	保留。	11b
5	RB_BOOT_LOADER	RO	引导程序状态。 1: 引导程序状态 (BootLoader); 0: 用户程序状态。	0
4	RB_CFG_DEBUG_EN	RO	调试使能控制状态。 1: 使能; 0: 禁止。	0
3	RB_CFG_BOOT_EN	RO	引导程序使能控制状态: 1: 使能; 0: 禁止。	1
2	RB_CFG_RESET_EN	RO	外部复位引脚 (RST#) 使能控制状态。 1: 外部输入低电平信号复位; 0: 禁止。	0
[1:0]	RB_RESET_FLAG	RO	最近一次复位状态。 00: 软件复位。来源: 内部软件请求复位并且 RB_BOOT_LOADER=0。 01: 上电复位。来源: 芯片工作电压低于阈值电压。 10: 看门狗复位。来源: 看门狗计数器溢出。 11: 手动复位。来源: RST#脚输入低电平。	xx

复位配置寄存器 (R8_RST_WDOG_CTRL)

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。其中[7:6]必须写01b。	0
3	RB_WDOG_INT_FLAG	RW1	看门狗计数器溢出标志。 1: 发生了看门狗计数溢出, 即检测到寄存器 R8_WDOG_COUNT 递增 0xFF 到 0x00; 0: 看门狗计数未溢出。 写1清0或重新加载看门狗计数器值自动清0。	0
2	RB_WDOG_INT_EN	RWA	看门狗计数器溢出中断使能。 1: 使能看门狗计数器溢出中断; 0: 禁止看门狗计数器溢出中断。	0
1	RB_WDOG_RST_EN	RWA	看门狗计数溢出复位使能。 1: 计数器溢出时产生复位; 0: 计数器溢出时不复位。	0
0	RB_SOFTWARE_RESET	WA/ WZ	系统软件复位, 自动清零: 1: 系统复位; 0: 无动作。	0

复位保持寄存器 (R8_GLOB_RESET_KEEP)

位	名称	访问	描述	复位值
[7:0]	R8_GLOB_RESET_KEEP	RW	复位保持寄存器, 只有上电复位才能复位此寄存器的值, 此寄存器不受手动复位、软件复位或者看门狗复位影响。	0

PLL 输出时钟分频寄存器 (R8_CLK_PLL_DIV)

位	名称	访问	描述	复位值
[7:0]	R8_CLK_PLL_DIV	RWA	低 4 位有效, [7:6]位必须写 01b, 最小值 2。	42h

时钟配置寄存器 (R8_CLK_CFG_CTRL)

位	名称	访问	描述	复位值
[7:2]	Reserved	RWA	保留, [7:6]必须写 10b。	100000b
1	RB_CLK_SEL_PLL	RWA	系统时钟源选择。 1: 来源 USB PHY 提供的 480MHz; 0: 来源外部晶振 30MHz。	0
0	RB_CLK_PLL_SLEEP	RWA	PLL 睡眠控制。 1: PLL 睡眠; 0: PLL 正常工作。	0

时钟辅助寄存器 (R8_CLK_MOD_AUX)

位	名称	访问	描述	复位值
[7:5]	Reserved	RWA	保留。	0
4	RB_MCO_EN	RWA	MCO 引脚输出使能控制位: 1: 使能; 0: 禁止。	0
[3:2]	RB_MCO_SEL_MSK	RWA	MCO 引脚输出时钟选择。 00: 输出 125MHz 时钟; 01: 输出 25MHz 时钟; 1x: 输出 2.5MHz 时钟。	0
1	RB_EXT_125M_EN	RWA	外部 125MHz 时钟用于以太网的使能。 1: 使能; 0: 禁止。	0
0	RB_INT_125M_EN	RWA	USB PHY 125MHz 时钟用于以太网的使能位: 1: 使能; 0: 禁止。	0

注: RB_EXT_125M_EN 和 RB_INT_125M_EN 只能选择一个时钟使能, 如果要切换, 需要先把一个禁止再使能另一个时钟。

睡眠控制寄存器 0 (R8_SLP_CLK_OFF0)

位	名称	访问	描述	复位值
7	RB_SLP_CLK_UART3	RWA	UART3 时钟控制。 1: 关闭; 0: 开启。	0
6	RB_SLP_CLK_UART2	RWA	UART2 时钟控制。 1: 关闭; 0: 开启。	0
5	RB_SLP_CLK_UART1	RWA	UART1 时钟控制。 1: 关闭; 0: 开启。	0
4	RB_SLP_CLK_UART0	RWA	UART0 时钟控制。	0

			1: 关闭; 0: 开启。	
3	RB_SLP_CLK_PWMX	RWA	PWM 时钟控制。 1: 关闭; 0: 开启。	0
2	RB_SLP_CLK_TMR2	RWA	TMR2 时钟控制。 1: 关闭; 0: 开启。	0
1	RB_SLP_CLK_TMR1	RWA	TMR1 时钟控制。 1: 关闭; 0: 开启。	0
0	RB_SLP_CLK_TMRO	RWA	TMRO 时钟控制。 1: 关闭; 0: 开启。	0

睡眠控制寄存器 1 (R8_SLP_CLK_OFF1)

位	名称	访问	描述	复位值
7	RB_SLP_CLK_DVP	RO	DVP 控制器时钟控制。 1: 关闭; 0: 开启。	0
6	RB_SLP_CLK_SERD	RWA	SerDes 控制器时钟控制。 1: 关闭; 0: 开启。	0
5	RB_SLP_CLK_USBSS	RWA	USB3.0 控制器时钟控制。 1: 关闭; 0: 开启。	0
4	RB_SLP_CLK_USBHS	RWA	USB2.0 控制器时钟控制。 1: 关闭; 0: 开启。	0
3	RB_SLP_CLK_HSPI	RWA	高速并口控制器时钟控制。 1: 关闭; 0: 开启。	0
2	RB_SLP_CLK_EMMC	RWA	EMMC 时钟控制。 1: 关闭; 0: 开启。	0
1	RB_SLP_CLK_SPI1	RWA	SPI1 时钟控制。 1: 关闭; 0: 开启。	0
0	RB_SLP_CLK_SPI0	RWA	SPI0 时钟控制。 1: 关闭; 0: 开启。	0

注: CH569 微控制器, 没有 DVP 功能, 所以 RB_SLP_CLK_DVP 控制位保留, 写无效。CH565 微控制器, 没有 HSPI 功能, 所以 RB_SLP_CLK_HSPI 控制位保留, 写无效。

唤醒控制寄存器 (R8_SLP_WAKE_CTRL)

位	名称	访问	描述	复位值
[7:6]	Reserved	RO	保留。	0
5	RB_SLP_ETH_WAKE	RWA	以太网低功耗下唤醒使能控制。 1: 使能以太网唤醒功能; 0: 禁止以太网唤醒功能。	0
4	RB_SLP_GPIO_WAKE	RWA	GPIO 端口低功耗下唤醒使能控制。 1: 使能 GPIO 唤醒功能; 0: 禁止 GPIO 唤醒功能。	0
3	RB_SLP_CLK_ECDC	RWA	加密模块时钟关闭控制。 1: 关闭加密模块时钟; 0: 开启加密模块时钟。	0
2	RB_SLP_CLK_ETH	RWA	以太网时钟关闭控制。 1: 关闭以太网时钟; 0: 开启以太网时钟。	0
1	RB_SLP_USBSS_WAKE	RWA	USB3.0 低功耗下唤醒使能控制。 1: 使能 USB3.0 唤醒功能;	0

			0: 禁止 USB3.0 唤醒功能。	
0	RB_SLP_USBHS_WAKE	RWA	USB2.0 低功耗下唤醒使能控制。 1: 使能 USB2.0 唤醒功能 0: 禁止 USB2.0 唤醒功能	0

低功耗电源管理寄存器 (R8_SLP_POWER_CTRL)

位	名称	访问	描述	复位值
[7:1]	Reserved	R0	保留。	0
0	RB_SLP_USBHS_PWRDN	RWA	USBHS 模块电源控制。 1: 模块下电; 0: 正常上电。	0

SerDes PHY 模拟参数配置寄存器 1 (R16_SERD_ANA_CFG1)

位	名称	访问	描述	复位值
[15:10]	Reserved	R0	保留。	0
9	RB_SERD_DN_TST	RWA	SerDes PHY 的 GXM 脚测试使能。 1: 内部测试模式。GXM 输出内部电压值, 不再作为收发数据串行口; 0: SerDes 工作模式。GXM 作为收发数据串行口。	0
8	RB_SERD_30M_SEL	RWA	SerDes PHY 的参考时钟源选择。 1: 选择 USB PHY480MHz 的 16 分频作为参考时钟; 0: 选择 30MHz 晶振作为参考时钟。	0
[7:0]	RB_SERD_PLL_CFG	RWA	SerDes PHY 内置 PLL 配置位	5Ah

SerDes PHY 模拟参数配置寄存器 2 (R32_SERD_ANA_CFG2)

位	名称	访问	描述	复位值
[31:25]	Reserved	R0	保留。	0
[24:0]	RB_SERD_TRX_CFG	RWA	SerDes PHY 的收发参数配置。 具体配置值需根据传输速度调整, 请参考 SerDes 软件例程提供的配置。	423015h

第 4 章 中断控制器 (PFIC)

系统内置快速可编程中断控制器 (PFIC - Programmable Fast Interrupt Controller)，最多支持 255 个中断向量。当前系统管理了 21 个外设中断通道和 5 个内核中断通道，其他中断源保留。

4.1 主要特性

- 21+3个可单独屏蔽中断，每个中断请求都有独立的触发和屏蔽位、状态位
- 提供一个不可屏蔽中断NMI
- 特有快速中断进出机制，硬件自动压栈和恢复，无需指令开销
- 特有快速中断响应机制，4 路可编程直达中断向量地址

4.2 系统定时器

内核自带了一个 64 位递减型计数器 (SysTick)，支持 HCLK 或者 HCLK/8 作为时基，具有较高优先级，校准后可用于时间基准。

4.3 中断和异常的向量表

表 4-1 系列产品向量表

编号	优先级	类型	名称	描述	入口地址
0		-	-		0x00000000
1	-3	固定	Reset	复位	0x00000004
2	-2	固定	NMI	不可屏蔽中断	0x00000008
3	-1	固定	EXC	异常中断	0x0000000C
4-11	-	-	-	保留	
12	0	可编程	SysTick	系统定时器中断	0x00000030
13	-	-	-	保留	
14	1	可编程	SWI	软件中断	0x00000038
15	-	-	-	保留	
16	2	可编程	WDOG	看门狗超时复位中断	0x00000040
17	3	可编程	TMR0	定时器 0 中断	0x00000044
18	4	可编程	GPIO	GPIO 端口中断	0x00000048
19	5	可编程	SPI0	SPI0 中断	0x0000004C
20	6	可编程	USBSS	USB3.0 中断	0x00000050
21	7	可编程	LINK	USB3.0 链路层中断	0x00000054
22	8	可编程	TMR1	定时器 1 中断	0x00000058
23	9	可编程	TMR2	定时器 2 中断	0x0000005C
24	10	可编程	UART0	串口 0 中断	0x00000060
25	11	可编程	USBHS	USB2.0 中断	0x00000064
26	12	可编程	EMMC	EMMC 中断	0x00000068
27	13	可编程	DVP	DVP 中断	0x0000006C
28	14	可编程	HSPI	HSPI 中断	0x00000070
29	15	可编程	SPI1	SPI1 中断	0x00000074
30	16	可编程	UART1	串口 1 中断	0x00000078
31	17	可编程	UART2	串口 2 中断	0x0000007C

32	18	可编程	UART3	串口 3 中断	0x00000080
33	19	可编程	SerDes	SerDes 控制器中断	0x00000084
34	20	可编程	ETH	以太网中断	0x00000088
35	21	可编程	PMT	以太网电源管理中断	0x0000008C
36	22	可编程	ECDC	加密模块中断	0x00000090

4.4 寄存器描述

4.4.1 PFIC 寄存器描述

PFIC 相关寄存器物理基地址：0xE000E000

表 4-2 PFIC 相关寄存器列表

名称	偏移地址	描述	复位值
R32_PFIC_ISR1	0x00	PFIC 中断使能状态寄存器 1	0x00000000
R32_PFIC_ISR2	0x04	PFIC 中断使能状态寄存器 2	0x00000000
R32_PFIC_IPR1	0x20	PFIC 中断挂起状态寄存器 1	0x00000000
R32_PFIC_IPR2	0x24	PFIC 中断挂起状态寄存器 2	0x00000000
R32_PFIC_ITHRESDR	0x40	PFIC 中断优先级阈值配置寄存器	0x00000000
R32_PFIC_FIBADDRR	0x44	PFIC 快速中断服务基地址寄存器	0x00000000
R32_PFIC_CFGR	0x48	PFIC 中断配置寄存器	0x00000000
R32_PFIC_GISR	0x4C	PFIC 中断全局状态寄存器	0x00000000
R32_PFIC_FIOADDRR0	0x60	PFIC 快速中断 0 偏移地址寄存器	0x00000000
R32_PFIC_FIOADDRR1	0x64	PFIC 快速中断 1 偏移地址寄存器	0x00000000
R32_PFIC_FIOADDRR2	0x68	PFIC 快速中断 2 偏移地址寄存器	0x00000000
R32_PFIC_FIOADDRR3	0x6C	PFIC 快速中断 3 偏移地址寄存器	0x00000000
R32_PFIC_IENR1	0x100	PFIC 中断使能设置寄存器 1	0x00000000
R32_PFIC_IENR2	0x104	PFIC 中断使能设置寄存器 2	0x00000000
R32_PFIC_IRER1	0x180	PFIC 中断使能清除寄存器 1	0x00000000
R32_PFIC_IRER2	0x184	PFIC 中断使能清除寄存器 2	0x00000000
R32_PFIC_IPSR1	0x200	PFIC 中断挂起设置寄存器 1	0x00000000
R32_PFIC_IPSR2	0x204	PFIC 中断挂起设置寄存器 2	0x00000000
R32_PFIC_IPRR1	0x280	PFIC 中断挂起清除寄存器 1	0x00000000
R32_PFIC_IPRR2	0x284	PFIC 中断挂起清除寄存器 2	0x00000000
R32_PFIC_IACR1	0x300	PFIC 中断激活状态寄存器 1	0x00000000
R32_PFIC_IACR2	0x304	PFIC 中断激活状态寄存器 2	0x00000000
R32_PFIC_IPRIORx	0x400	PFIC 中断优先级配置寄存器	0x00000000
R32_PFIC_SCTLR	0xD10	PFIC 系统控制寄存器	0x00000000

PFIC 中断使能状态寄存器 1 (PFIC_ISR1)

位	名称	访问	描述	复位值
[31:12]	INTSTA	R0	31#及以下中断当前使能状态。 1: 当前编号中断已使能; 0: 当前编号中断未启用。	0
[11:0]	Reserved	R0	保留。 Reset、NMI、EXC 中断忽略, 下同。	0

PFIC 中断使能状态寄存器 2 (PFIC_ISR2)

位	名称	访问	描述	复位值
[31:28]	Reserved	R0	保留。	0
[27:0]	INTENSTA	R0	32#及以上中断当前使能状态。 1: 当前编号中断已使能; 0: 当前编号中断未启用。	0

PFIC 中断挂起状态寄存器 1 (PFIC_IPR1)

位	名称	访问	描述	复位值
[31:12]	PENDSTA	R0	31#及以下中断当前挂起状态。 1: 当前编号中断已挂起; 0: 当前编号中断未挂起。	0
[11:0]	Reserved	R0	保留。	0

PFIC 中断挂起状态寄存器 2 (PFIC_IPR2)

位	名称	访问	描述	复位值
[31:28]	Reserved	R0	保留。	0
[27:0]	PENDSTA	R0	32#及以上中断当前挂起状态。 1: 当前编号中断已挂起; 0: 当前编号中断未挂起。	0

PFIC 中断优先级阈值配置寄存器 (PFIC_ITHRESDR)

位	名称	访问	描述	复位值
[31:8]	Reserved	R0	保留。	0
[7:0]	THRESHOLD	RW	中断优先级阈值设置值。 低于当前设置值的中断优先级值, 当挂起时不执行中断服务; 此寄存器为 0 时表示阈值寄存器功能无效。 [7:4]: 优先级阈值。 [3:0]: 保留, 固定为 0, 写无效。	0

PFIC 快速中断服务基地址寄存器 (PFIC_FIBADDR)

位	名称	访问	描述	复位值
[31:28]	BASEADDR	RW	快速中断响应的目标跳转地址高 4 位。 与 PFIC_FIOFADDR* 共同构成对应编号的快速中断向量 (中断服务程序的 32 位跳转地址)。	0
[27:0]	Reserved	R0	保留。	0

PFIC 中断配置寄存器 (PFIC_CFGR)

位	名称	访问	描述	复位值
[31:16]	KEYCODE	WO	对应不同的目标控制位, 需要同步写入相应的安全访问标识数据才能修改, 读出数据固定为 0。 KEY1 = 0xFA05; KEY2 = 0xBCAF; KEY3 = 0xBEEF。	0
[15:8]	Reserved	R0	保留。	0
7	SYSRESET	WO	系统复位 (同步写入 KEY3)。自动清 0。	0

			写 1 有效，写 0 无效。	
6	PFICRESET	WO	PFIC 控制模块复位，自动清 0。 写 1 有效，写 0 无效。	0
5	EXCRESET	WO	异常中断清除挂起（同步写入 KEY2）。 写 1 有效，写 0 无效。	0
4	EXCSET	WO	异常中断挂起（同步写入 KEY2）。 写 1 有效，写 0 无效。	0
3	NMIRESET	WO	NMI 中断清除挂起（同步写入 KEY2）。 写 1 有效，写 0 无效。	0
2	NMISSET	WO	NMI 中断挂起（同步写入 KEY2）。 写 1 有效，写 0 无效。	0
1	NESTCTRL	RW	嵌套中断使能控制： 1: 关闭； 0: 打开（同步写入 KEY1）。	0
0	HWSTKCTRL	RW	硬件压栈使能控制： 1: 关闭； 0: 打开（同步写入 KEY1）。	0

PFIC 中断全局状态寄存器 (PFIC_GISR)

位	名称	访问	描述	复位值
[31:10]	Reserved	RO	保留。	0
9	GPENDSTA	RO	当前是否有中断处于挂起： 1: 有； 0: 没有。	0
8	GACTSTA	RO	当前是否有中断被执行： 1: 有； 0: 没有。	0
[7:0]	NESTSTA	RO	当前中断嵌套状态，目前支持 2 级嵌套，[1:0] 有效。 3: 第 2 级中断中； 1: 第 1 级中断中； 0: 没有中断发生； 其他: 不可能情况。	0

PFIC 快速中断 0 偏移地址寄存器 (PFIC_FIOFADDR0)

位	名称	访问	描述	复位值
[31:24]	IRQID0	RW	快速中断 0 的编号。	0
[23:0]	OFFADDR0	RW	快速中断 0 服务程序低 24 位地址，其中低 20 位配置有效，[23:20] 固定为 0。	0

PFIC 快速中断 1 偏移地址寄存器 (PFIC_FIOFADDR1)

位	名称	访问	描述	复位值
[31:24]	IRQID1	RW	快速中断 1 的编号。	0
[23:0]	OFFADDR1	RW	快速中断 1 服务程序低 24 位地址，其中低 20 位配置有效，[23:20] 固定为 0。	0

PFIC 快速中断 2 偏移地址寄存器 (PFIC_FIOFADDR2)

位	名称	访问	描述	复位值
[31:24]	IRQID2	RW	快速中断 2 的编号。	0
[23:0]	OFFADDR2	RW	快速中断 2 服务程序低 24 位地址，其中低 20 位配置有效，[23:20] 固定为 0。	0

PFIC 快速中断 3 偏移地址寄存器 (PFIC_FIOFADDR3)

位	名称	访问	描述	复位值
[31:24]	IRQID3	RW	快速中断 3 的编号。	0
[23:0]	OFFADDR3	RW	快速中断 3 服务程序低 24 位地址, 其中低 20 位配置有效, [23:20] 固定为 0。	0

PFIC 中断使能设置寄存器 1 (PFIC_IENR1)

位	名称	访问	描述	复位值
[31:12]	INTEN	RW1	31#及以下中断使能控制。 1: 当前编号中断使能; 0: 无影响。	0
[11:0]	Reserved	R0	保留。	0

PFIC 中断使能设置寄存器 2 (PFIC_IENR2)

位	名称	访问	描述	复位值
[31:28]	Reserved	R0	保留。	0
[27:0]	INTEN	RW1	32#及以上中断使能控制。 1: 当前编号中断使能; 0: 无影响。	0

PFIC 中断使能清除寄存器 1 (PFIC_IRER1)

位	名称	访问	描述	复位值
[31:12]	INTRESET	RW1	31#及以下中断关闭控制。 1: 当前编号中断关闭; 0: 无影响。	0
[11:0]	Reserved	R0	保留。	0

PFIC 中断使能清除寄存器 2 (PFIC_IRER2)

位	名称	访问	描述	复位值
[31:28]	Reserved	R0	保留。	0
[27:0]	INTRESET	RW1	32#及以上中断关闭控制。 1: 当前编号中断关闭; 0: 无影响。	0

PFIC 中断挂起设置寄存器 1 (PFIC_IPSR1)

位	名称	访问	描述	复位值
[31:12]	PENDSET	RW1	31#及以下中断挂起设置。 1: 当前编号中断挂起; 0: 无影响。	0
[11:0]	Reserved	R0	保留。	0

PFIC 中断挂起设置寄存器 2 (PFIC_IPSR2)

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

[31:28]	Reserved	R0	保留。	0
[27:0]	PENDESET	RW1	32#及以上中断挂起设置。 1: 当前编号中断挂起; 0: 无影响。	0

PFIC 中断挂起清除寄存器 1 (PFIC_IPRR1)

位	名称	访问	描述	复位值
[31:12]	PENDESET	RW1	31#及以下中断挂起清除。 1: 当前编号中断清除挂起状态; 0: 无影响。	0
[11:0]	Reserved	R0	保留。	0

注: 对于 Reset、NMI、EXG 编号的中断上述寄存器无效。

PFIC 中断挂起清除寄存器 2 (PFIC_IPRR2)

位	名称	访问	描述	复位值
[31:28]	Reserved	R0	保留。	0
[27:0]	PENDESET	RW1	32#及以上中断挂起清除。 1: 当前编号中断清除挂起状态; 0: 无影响。	0

PFIC 中断激活状态寄存器 1 (PFIC_IACTR1)

位	名称	访问	描述	复位值
[31:12]	IACTS	RW1	31#及以下中断执行状态。 1: 当前编号中断执行中; 0: 当前编号中断没执行。	0
[11:0]	Reserved	R0	保留。	0

PFIC 中断激活状态寄存器 2 (PFIC_IACTR2)

位	名称	访问	描述	复位值
[31:28]	Reserved	R0	保留。	0
[27:0]	IACTS	RW1	32#及以上中断执行状态。 1: 当前编号中断执行中; 0: 当前编号中断没执行。	0

PFIC 中断优先级配置寄存器 (PFIC_IPRIORx) (x=0-63)

控制器支持 256 个中断 (0-255), 每个中断使用 8bit 来设置控制优先级。

	31	24	23	16	15	8	7	0
IPRIOR63	PRIO_255			PRIO_254		PRIO_253		PRIO_252
...
IPRIORx	PRIO_(4x+3)			PRIO_(4x+2)		PRIO_(4x+1)		PRIO_(4x)
...
IPRIOR0	PRIO_3			PRIO_2		PRIO_1		PRIO_0

位	名称	访问	描述	复位值
[2047:2040]	IP_255	RW	同 IP_0 描述。	0
...
[31:24]	IP_3	RW	同 IP_0 描述。	0
[23:16]	IP_2	RW	同 IP_0 描述。	0
[15:8]	IP_1	RW	同 IP_0 描述。	0
[7:0]	IP_0	RW	编号 0 中断优先级配置： [7:4]：优先级控制位。 [3:0]：保留，固定为 0，写无效。 优先级数值越小则优先级越高。只有 2 级中断嵌套，即只能抢占 1 次。	0

PFIC 系统控制寄存器 (PFIC_SCTLR)

位	名称	访问	描述	复位值
[31:6]	Reserved	RO	保留。	0
5	SETEVENT	WO	设置事件，可以唤醒 WFE 的情况。	0
4	SEVONPEND	RW	当发生事件或者中断挂起状态时，可以从 WFE 指令后唤醒系统，如果未执行 WFE 指令，将在下次执行该指令后立即唤醒系统。 1: 启用的事件和所有中断（包括未开启中断）都能唤醒系统； 0: 只有启用的事件和启用的中断可以唤醒系统。	0
3	WFIOWFE	RW	将 WFI 指令当成是 WFE 执行。 1: 将之后的 WFI 指令当做 WFE 指令； 0: 无作用。	0
2	SLEEPDEEP	RW	控制系统的低功耗模式： 1: deepsleep 0: sleep	0
1	SLEEPONEXIT	RW	控制离开中断服务程序后，系统状态： 1: 系统进入低功耗模式； 0: 系统进入主程序。	0
0	Reserved	RO	保留。	0

4.4.2 SysTick 寄存器描述

STK 相关寄存器物理基地址：0xE000F000

表 4-3 STK 相关寄存器列表

名称	偏移地址	描述	复位值
R32_STK_CTLR	0x00	系统计数控制寄存器	0x00000000
R32_STK_CNTL	0x04	系统计数器低位寄存器	0x00000000
R32_STK_CNTH	0x08	系统计数器高位寄存器	0x00000000
R32_STK_CMPLR	0x0C	计数重加载低位寄存器	0x00000000
R32_STK_CMPHR	0x10	计数重加载高位寄存器	0x00000000
R32_STK_CNTFG	0x14	计数器计数标志寄存器	0x00000000

系统计数控制寄存器 (STK_CTLR)

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

[31:9]	Reserved	RO	保留。	0
8	STRELOAD	W1	重装载控制，写 1 将计数重加载寄存器（64 位）数值更新到当前计数器寄存器中。	0
[7:3]	Reserved	RO	保留。	0
2	STCLK	RW	计数器时钟源选择： 1: HCLK 做时基； 0: HCLK/8 做计数时基。	0
1	STIE	RW	计数器中断使能控制位： 1: 使能计数器中断； 0: 无计数器中断。	0
0	STE	RW	系统计数器使能控制位： 1: 启动系统计数器 STK； 0: 关闭系统计数器 STK，计数器停止计数。	0

系统计数器低位寄存器（STK_CNTL）

位	名称	访问	描述	复位值
[31:0]	CNTL	RW	当前计数器计数值低 32 位。计数递减。 当 CNT[63:0] 值减为 0 时，置位 STK 中断标志。	0

系统计数器高位寄存器（STK_CNTH）

位	名称	访问	描述	复位值
[31:0]	CNTH	RW	当前计数器计数值高 32 位。计数递减。 当 CNT[63:0] 值减为 0 时，置位 STK 中断标志。	0

注：寄存器 STK_CNTL 和寄存器 STK_CNTH 共同构成了 64 位递增的系统计数器。

计数重加载低位寄存器（STK_CMPLR）

位	名称	访问	描述	复位值
[31:0]	CMPL	RW	设置重加载计数器值低 32 位。	0

计数重加载高位寄存器（STK_CMPHR）

位	名称	访问	描述	复位值
[31:0]	CMPH	RW	设置重加载计数器值高 32 位。	0

注：寄存器 STK_CMPLR 和寄存器 STK_CMPHR 共同构成了 64 位计数器比较值。

计数器计数标志寄存器（STK_CNTFG）

位	名称	访问	描述	复位值
[31:2]	Reserved	RO	保留。	0
1	CNTIF	RWO	计数器减为 0 标志。写 0 清除，写 1 无效。	0
0	SWIE	RW	系统软件中断使能位。 1: 打开软件中断，同时触发软件中断服务； 0: 关闭软件中断。	0

注：SWIE=1 将执行软件中断服务，如果不在中断服务中关闭 SWIE 功能，离开中断服务将再次触发软件中断服务。

第 5 章 通用 I/O 和复用功能

5.1 GPIO 简介

系统提供了 2 组 GPIO 端口 PA 和 PB，共 49 个通用输入输出引脚，部分引脚具有中断、复用及映射功能。

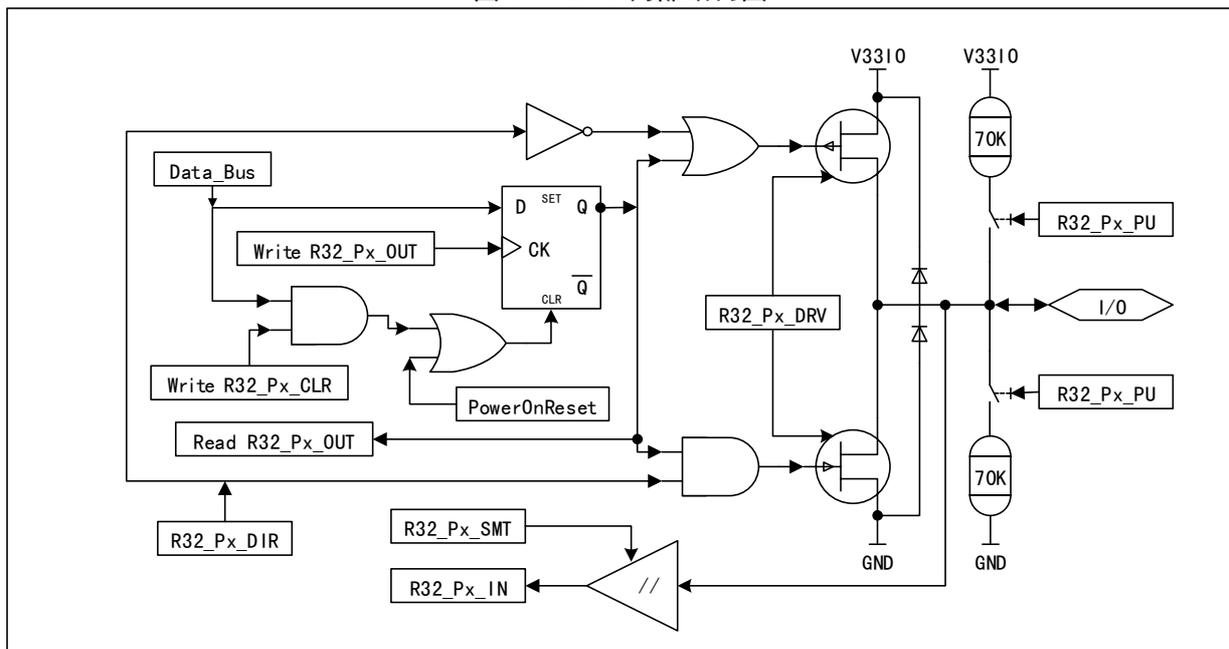
每个 GPIO 端口有一个 32 位方向配置寄存器 R32_Px_DIR，一个 32 位输入数据寄存器 R32_Px_PIN，一个 32 位数据输出寄存器 R32_Px_OUT，一个 32 位数据复位寄存器 R32_Px_CLR，一个 32 位上拉配置寄存器 R32_Px_PU，一个 32 位开漏输出和下拉使能寄存器 R32_Px_PD，一个 32 位驱动能力配置寄存器 R32_Px_DRV，一个 32 位低斜率输出和施密特输入寄存器 R32_Px_SMT。

PA 端口中，PA[0]–PA[23]位有效，对应芯片上 24 个 GPIO 引脚；PB 端口中，PB[0]–PB[24]位有效，对应芯片上 25 个 GPIO 引脚；其中 8 个 I/O 引脚具有中断功能，并可实现睡眠唤醒功能，它们分别是：PA2/PA3/PA4 和 PB3/PB4/PB11/PB12/PB15。

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。如果引脚的复用功能没有开启，则做为通用 I/O 口使用。

下图是 GPIO 内部结构框图：

图 5-1 GPIO 内部结构图



每个 GPIO 口都可以配置成 5 种模式，具体见下表：

表 5-1 GPIO 引脚模式配置

PA_SMT	模式		Px_DIR	Px_PU	PA_PD	Px_DRV
X	浮空输入/高阻输入		0	0	0	X
1	无施密特触发	带上拉电阻的输入	0	1	0	X
		带下拉电阻的输入	0	0	1	X
0	支持施密特触发	带上拉电阻的施密特输入	0	1	0	X
		带下拉电阻的施密特输入	0	0	1	X
1	低斜率	推挽输出，驱动能力 8mA 级别	1	X	0	0
		推挽输出，驱动能力 16mA 级别	1	X	0	1
0	快速	推挽输出，驱动能力 8mA 级别	1	X	0	0

		推挽输出, 驱动能力 16mA 级别	1	X	0	1
X		开漏输出, 驱动能力 8mA 级别	1	X	1	0
		开漏输出, 驱动能力 16mA 级别	1	X	1	1

5.2 外部中断/唤醒

芯片的 8 个 I/O 管脚 PA2/PA3/PA4 和 PB3/PB4/PB11/PB12/PB15 具有中断功能, 并可实现睡眠唤醒。为了使用外部中断, 端口位必须配置成输入模式。并提供 4 种触发模式: 高电平、低电平、上升沿、下降沿。

唤醒功能需要打开端口位的中断使能 R8_GPIO_INT_ENABLE, 并开启寄存器 R8_SLP_WAKE_CTRL 中的 GPIO 唤醒控制位 RB_SLP_GPIO_WAKE。

5.3 GPIO 的复用与重映射

5.3.1 复用功能

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, 启用不同功能模块后, 相应引脚被配置成各自功能模块对应的功能引脚。

如果一个管脚复用多个功能, 并且多个功能都已开启, 那么复用功能的优先级顺序请参考 1.3 节引脚的“复用功能及映射”列表中功能顺序, 优先级从高到低。

例如: PA0 脚复用为 MD5/HD6/BDO, 则 EMMC 的数据线功能优先, 主动并口数据线功能最低。这样可以在多个复用功能中, 将功能优先级最低的部分不需使用引脚启用相对更高优先级的复用功能。

以下各表列出了各个功能模块所使用的 I/O 引脚配置。

表 5-2 定时器 x

TMRO/1/2/3 引脚	功能配置	GPIO 配置
TMRx	输入捕捉通道 x	输入 (浮空/上拉/下拉)
	输出 PWM 通道 x	推挽输出

表 5-3 UARTx 接口

UART0/1/2/3 引脚	功能配置	GPIO 配置
TXDx	串口发送 x	推挽输出
RXDx	串口接收 x	上拉输入 (推荐) 或浮空输入
RTS, DTR	MODEM 信号输入或 RS485 控制	推挽输出
CTS, DSR, RI, DCD	MODEM 信号输入	上拉输入 (推荐) 或浮空输入

表 5-3 SPIx 接口

SPIO/1 引脚	功能配置	GPIO 配置
SCKx	主模式时钟输出	推挽输出
	从模式时钟输入	输入 (浮空/上拉/下拉)
MOSIx	全双工模式-主模式	推挽输出
	全双工模式-从模式	输入 (浮空/上拉/下拉)
	半双工模式-主模式	未用到, 可做通用 I/O
	半双工模式-从模式	未用到, 可做通用 I/O
MISOx	全双工模式-主模式	输入 (浮空/上拉/下拉)
	全双工模式-从模式	输入 (推荐上拉, 片选后硬件自动切为推挽输出) 或推挽输出 (禁止用于总线连接方式)

SCS	半双工模式-主模式	输入或推挽输出，软件切换
	半双工模式-从模式	输入（推荐上拉，片选后硬件自动切为推挽输出）
	主模式片选输出	推挽输出（可换用其他引脚）
	从模式片选输入	上拉输入（推荐）或浮空输入

表 5-4 高速并行接口

HSPI 引脚	功能配置	GPIO 配置
HTCLK, HTREQ, HTVLD, HTACK	控制输出信号	推挽输出
HTRDY, HRCLK, HRACT, HRVLD	控制输入信号	下拉输入（推荐）或浮空输入
HDO~HD31	数据信号	浮空输入

表 5-5 DVP 接口

DVP 引脚	功能配置	GPIO 配置
DVSYNC, DHSYNC	同步信号输入	上拉输入（推荐）或浮空输入
DPCLK	时钟信号输入	上拉输入（推荐）或浮空输入
DDO~DD31	数据信号	上拉输入（推荐）或浮空输入

表 5-6 EMMC 接口

EMMC 引脚	功能配置	GPIO 配置
MSDCK	时钟信号输出	推挽输出
MCMD	命令信号输出	推挽输出
MDO~MD7	数据信号	上拉输入（推荐）或浮空输入

表 5-7 以太网接口

MI1 引脚	功能配置	GPIO 配置
ETXD0~ETXD3	以太网发送数据信号	推挽输出
ERXD0~ERXD3	以太网接收数据信号	上拉输入（推荐）或浮空输入
ERXDV, ERXC	控制及时钟信号输入	上拉输入（推荐）或浮空输入
ETXC, ETXEN	控制及时钟信号输出	推挽输出
EMDIO	SMI 管理数据	上拉输入（推荐）或浮空输入
EMDCK	SMI 管理时钟	推挽输出
ETCKI	以太网时钟输入	上拉输入（推荐）或浮空输入
EMCO	以太网时钟输出	推挽输出

表 5-8 BUS8 接口

BUS8 引脚	功能配置	GPIO 配置
BRD#, BWR#	总线控制	推挽输出
BDO~BD7, BAO~BA14	总线数据，地址	上拉输入（推荐）或浮空输入

表 5-9 高速模拟信号接口

高速模拟引脚	功能配置	GPIO 配置
UD+, UD-	连接内部 USB PHY	浮空输入
SSTXA, SSTXB		浮空输入
SSRXA, SSRXB		浮空输入
GXM, GXP	连接内部 SerDes PHY	浮空输入

5.3.2 功能引脚重映射

为了让外设功能同时使用率达到最优，可以通过设置 R8_PIN_ALTERNATE 寄存器将一些功能硬件

重映射到其他引脚上。系统支持 UART0、TMR1、TMR2 外设引脚的重映射，具体参考下表：

表 5-10 重映射引脚

外设功能	默认引脚	重映射引脚
UART0	PB5/PB6	PA5/PA6
TMR1/PWM5	PB15	PB0
TMR2/PWM6	PA4	PB3

5.4 寄存器描述

GPIO 相关寄存器物理基地址为：0x40001000

表 5-11 GPIO 相关寄存器列表

名称	偏移地址	描述	复位值
R8_GPIO_INT_FLAG	0x1C	GPIO 中断标志寄存器	0x00
R8_GPIO_INT_ENABLE	0x1D	GPIO 中断使能寄存器	0x00
R8_GPIO_INT_MODE	0x1E	GPIO 中断触发模式寄存器	0x00
R8_GPIO_INT_POLAR	0x1F	GPIO 中断极性寄存器	0x00
R32_PA_DIR	0x40	PA 端口方向设置寄存器	0x00000000
R32_PA_PIN	0x44	PA 端口数据输入寄存器	0x00XXXXXX
R32_PA_OUT	0x48	PA 端口数据输出寄存器	0x00000000
R32_PA_CLR	0x4C	PA 端口数据复位寄存器	0x00000000
R32_PA_PU	0x50	PA 端口上拉使能寄存器	0x00000000
R32_PA_PD	0x54	PA 端口开漏输出和输入下拉使能寄存器	0x00000000
R32_PA_DRV	0x58	PA 端口驱动能力配置寄存器	0x00000000
R32_PA_SMT	0x5C	PA 端口低斜率输出和施密特输入寄存器	0x00000000
R32_PB_DIR	0x60	PB 端口方向设置寄存器	0x00000000
R32_PB_PIN	0x64	PB 端口数据输入寄存器	0x0XXXXXXX
R32_PB_OUT	0x68	PB 端口数据输出寄存器	0x00000000
R32_PB_CLR	0x6C	PB 端口数据复位寄存器	0x00000000
R32_PB_PU	0x70	PB 端口上拉配置寄存器	0x00000000
R32_PB_PD	0x74	PB 端口开漏输出和输入下拉配置寄存器	0x00000000
R32_PB_DRV	0x78	PB 端口驱动能力配置寄存器	0x00000000
R32_PB_SMT	0x7C	PB 端口低斜率输出和施密特输入寄存器	0x00000000
R8_PIN_ALTERNATE	0x12	复用重映射配置寄存器	0x00

GPIO 中断标志寄存器 (R8_GPIO_INT_FLAG)

位	名称	访问	描述	复位值
7	RB_GPIO_PB15_IF	RW1	PB15 引脚中断标志位，写 1 清零。 1: 有中断； 0: 无中断。	0
6	RB_GPIO_PB12_IF	RW1	PB12 引脚中断标志位，写 1 清零。 1: 有中断； 0: 无中断。	0
5	RB_GPIO_PB11_IF	RW1	PB11 引脚中断标志位，写 1 清零。 1: 有中断； 0: 无中断。	0
4	RB_GPIO_PB4_IF	RW1	PB4 引脚中断标志位，写 1 清零。 1: 有中断； 0: 无中断。	0
3	RB_GPIO_PB3_IF	RW1	PB3 引脚中断标志位，写 1 清零。 1: 有中断； 0: 无中断。	0
2	RB_GPIO_PA4_IF	RW1	PA4 引脚中断标志位，写 1 清零。	0

			1: 有中断; 0: 无中断。	
1	RB_GPIO_PA3_IF	RW1	PA3 引脚中断标志位, 写 1 清零。 1: 有中断; 0: 无中断。	0
0	RB_GPIO_PA2_IF	RW1	PA2 引脚中断标志位, 写 1 清零。 1: 有中断; 0: 无中断。	0

GPIO 中断使能寄存器 (R8_GPIO_INT_ENABLE)

位	名称	访问	描述	复位值
7	RB_GPIO_PB15_IE	RW	PB15 引脚中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0
6	RB_GPIO_PB12_IE	RW	PB12 引脚中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0
5	RB_GPIO_PB11_IE	RW	PB11 引脚中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0
4	RB_GPIO_PB4_IE	RW	PB4 引脚中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0
3	RB_GPIO_PB3_IE	RW	PB3 引脚中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0
2	RB_GPIO_PA4_IE	RW	PA4 引脚中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0
1	RB_GPIO_PA3_IE	RW	PA3 引脚中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0
0	RB_GPIO_PA2_IE	RW	PA2 引脚中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0

GPIO 中断触发模式寄存器 (R8_GPIO_INT_MODE)

位	名称	访问	描述	复位值
7	RB_GPIO_PB15_IM	RW	PB15 引脚中断模式选择。 1: 边沿触发; 0: 电平触发。	0
6	RB_GPIO_PB12_IM	RW	PB12 引脚中断模式选择。 1: 边沿触发; 0: 电平触发。	0
5	RB_GPIO_PB11_IM	RW	PB11 引脚中断模式选择。 1: 边沿触发; 0: 电平触发。	0
4	RB_GPIO_PB4_IM	RW	PB4 引脚中断模式选择。 1: 边沿触发; 0: 电平触发。	0
3	RB_GPIO_PB3_IM	RW	PB3 引脚中断模式选择。 1: 边沿触发; 0: 电平触发。	0
2	RB_GPIO_PA4_IM	RW	PA4 引脚中断模式选择。 1: 边沿触发; 0: 电平触发。	0
1	RB_GPIO_PA3_IM	RW	PA3 引脚中断模式选择。	0

			1: 边沿触发; 0: 电平触发。	
0	RB_GPIO_PA2_IM	RW	PA2 引脚中断模式选择。 1: 边沿触发; 0: 电平触发。	0

GPIO 中断极性寄存器 (R8_GPIO_INT_POLAR)

位	名称	访问	描述	复位值
7	RB_GPIO_PB15_IP	RW	PB15 引脚中断极性选择。 1: 高电平/上升沿; 0: 低电平/下降沿。	0
6	RB_GPIO_PB12_IP	RW	PB12 引脚中断极性选择。 1: 高电平/上升沿; 0: 低电平/下降沿。	0
5	RB_GPIO_PB11_IP	RW	PB11 引脚中断极性选择。 1: 高电平/上升沿; 0: 低电平/下降沿。	0
4	RB_GPIO_PB4_IP	RW	PB4 引脚中断极性选择。 1: 高电平/上升沿; 0: 低电平/下降沿。	0
3	RB_GPIO_PB3_IP	RW	PB3 引脚中断极性选择。 1: 高电平/上升沿; 0: 低电平/下降沿。	0
2	RB_GPIO_PA4_IP	RW	PA4 引脚中断极性选择。 1: 高电平/上升沿; 0: 低电平/下降沿。	0
1	RB_GPIO_PA3_IP	RW	PA3 引脚中断极性选择。 1: 高电平/上升沿; 0: 低电平/下降沿。	0
0	RB_GPIO_PA2_IP	RW	PA2 引脚中断极性选择。 1: 高电平/上升沿; 0: 低电平/下降沿。	0

PA 端口方向设置寄存器 (R32_PA_DIR)

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:0]	R32_PA_DIR	RW	PA 引脚当前输入输出方向。 1: 输出模式; 0: 输入模式。	0

PA 端口输入数据寄存器 (R32_PA_PIN)

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:0]	R32_PA_PIN	R0	PA 引脚当前电平状态。 1: 高电平; 0: 低电平。 注: 仅在方向寄存器 (R32_PA_DIR) 对应位为 0 时, 该位值有效。	0

PA 端口输出数据寄存器 (R32_PA_OUT)

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

[31:24]	Reserved	R0	保留。	0
[23:0]	R32_PA_OUT	RW	PA 引脚输出电平状态。 1: 输出高电平; 0: 输出低电平。 注: 仅在方向寄存器 (R32_PA_DIR) 对应位为 1 时, 该位值有效。	0

PA 端口数据复位寄存器 (R32_PA_CLR)

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:0]	R32_PA_CLR	WZ	PA 引脚输出低电平。 1: 输出低电平; 0: 无作用。 注: 当某位置 1 时, 寄存器 R32_PA_OUT 对应位将清 0。	0

PA 端口上拉配置寄存器 (R32_PA_PU)

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:0]	R32_PA_PU	RW	PA 引脚上拉功能使能。 1: 使能引脚上拉; 0: 关闭引脚上拉。	0

PA 端口开漏输出和输入下拉使能寄存器 (R32_PA_PD)

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:0]	R32_PA_PD	RW	引脚方向为输出模式时: 1: 使能引脚开漏输出; 0: 关闭引脚开漏输出。 引脚方向为输入模式时: 1: 使能引脚下拉; 0: 关闭引脚下拉。	0

PA 端口驱动能力配置寄存器 (R32_PA_DRV)

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:0]	R32_PA_DRV	RW	PA 引脚输出驱动能力。 1: 驱动电流最大 16mA; 0: 驱动电流最大 8mA。	0

PA 端口低斜率输出和施密特输入寄存器 (R32_PA_SMT)

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:0]	R32_PA_SMT	RW	引脚方向为输出模式时: 1: 使能引脚低斜率输出; 0: 关闭引脚低斜率输出。 引脚方向为输入模式时: 1: 使能引脚施密特触发器输入;	1

			0: 关闭引脚施密特触发器输入。	
--	--	--	------------------	--

PB 端口方向设置寄存器 (R32_PB_DIR)

位	名称	访问	描述	复位值
[31:25]	Reserved	R0	保留。	0
[24:0]	R32_PB_DIR	RW	PB 引脚当前输入输出方向。 1: 输出模式; 0: 输入模式。	0

PB 端口输入数据寄存器 (R32_PB_PIN)

位	名称	访问	描述	复位值
[31:25]	Reserved	R0	保留。	0
[24:0]	R32_PB_PIN	R0	PB 引脚当前电平状态。 1: 高电平; 0: 低电平。 注: 仅在方向寄存器 (R32_PB_DIR) 对应位为 0 时, 该位值有效。	0

PB 端口输出数据寄存器 (R32_PB_OUT)

位	名称	访问	描述	复位值
[31:25]	Reserved	R0	保留。	0
[24:0]	R32_PB_OUT	RW	PB 引脚输出电平状态。 1: 输出高电平; 0: 输出低电平。 注: 仅在方向寄存器 (R32_PB_DIR) 对应位为 1 时, 该位值有效。	0

PB 端口数据复位寄存器 (R32_PB_CLR)

位	名称	访问	描述	复位值
[31:25]	Reserved	R0	保留。	0
[24:0]	R32_PB_CLR	WZ	PB 引脚输出低电平。 1: 输出低电平; 0: 无作用。 注: 当某位置 1 时, 寄存器 R32_PB_OUT 对应位将清 0。	0

PB 端口上拉配置寄存器 (R32_PB_PU)

位	名称	访问	描述	复位值
[31:25]	Reserved	R0	保留。	0
[24:0]	R32_PB_PU	RW	PB 引脚上拉功能使能。 1: 使能引脚上拉; 0: 关闭引脚上拉。	0

PB 端口开漏输出和输入下拉使能寄存器 (R32_PB_PD)

位	名称	访问	描述	复位值
[31:25]	Reserved	R0	保留。	0
[24:0]	R32_PB_PD	RW	引脚方向为输出模式时: 1: 使能引脚开漏输出; 0: 关闭引脚开漏输出。	0

			引脚方向为输入模式时： 1: 使能引脚下拉； 0: 关闭引脚下拉。	
--	--	--	---	--

PB 端口驱动能力配置寄存器 (R32_PB_DRV)

位	名称	访问	描述	复位值
[31:25]	Reserved	R0	保留。	0
[24:0]	R32_PB_DRV	RW	PB 引脚输出驱动能力。 1: 驱动电流最大 16mA; 0: 驱动电流最大 8mA。	0

PB 端口低斜率输出和施密特输入寄存器 (R32_PB_SMT)

位	名称	访问	描述	复位值
[31:25]	Reserved	R0	保留。	0
[24:0]	R32_PB_SMT	RW	引脚方向为输出模式时： 1: 使能引脚低斜率输出； 0: 关闭引脚低斜率输出。 引脚方向为输入模式时： 1: 使能引脚施密特触发器输入； 0: 关闭引脚施密特触发器输入。	1

复用重映射配置寄存器 (R8_PIN_ALTERNATE)

位	名称	访问	描述	复位值
[7:5]	Reserved	R0	保留。	0
4	RB_PIN_UART0	RW	UART0 重映射配置。 1: RXD0/TXD0 到 PA5/PA6 脚； 0: RXD0/TXD0 到 PB5/PB6 脚。	0
3	Reserved	R0	保留。	0
2	RB_PIN_TMR2	RW	TIMER2 重映射配置。 1: TMR2/PWM6/CAP2 到 PB3 脚； 0: TMR2/PWM6/CAP2 到 PA4 脚。	0
1	RB_PIN_TMR1	RW	TIMER1 重映射配置。 1: TMR1/PWM5/CAP1 到 PB0 脚； 0: TMR1/PWM5/CAP1 到 PB15 脚。	0
0	RB_PIN_MII	RW	以太网收发接口配置。 1: 以太网采用 RGMII 接口； 0: 以太网采用 RMII 接口。	1

第 6 章 串行外设接口 (SPI)

SPI 是一种全双工串行接口，总线上连接有一个主机和若干从机，同一时刻，仅有一对主从在通讯。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 MISO (主机输入/从机输出引脚) 和 SPI 串行数据引脚 MOSI (主机输出/从机输入引脚)。

6.1 主要特征

系统提供了 2 路 SPI 接口，其功能和操作都一致。

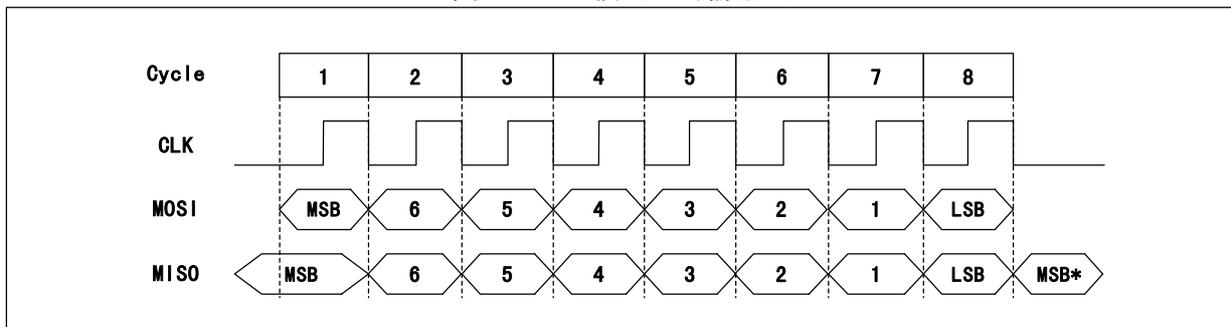
- 支持主机模式和从机模式
- 兼容串行外设接口 (SPI) 规范
- 支持模式 0 和模式 3 数据传输方式
- 8 位数据传输方式，数据位序可选：低位在前或高位在前
- 时钟频率最高接近 F_{sys} 一半
- 8 字节 FIFO 空间，多种传输标志通知
- 从机模式下支持首字节为命令模式或数据流模式
- 支持 DMA 数据传输

6.2 SPI 数据传输格式

SPI 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 (RB_SPIx_CTRL_MOD) 的 RB_SPI_MST_SCK_MOD 位来进行选择。

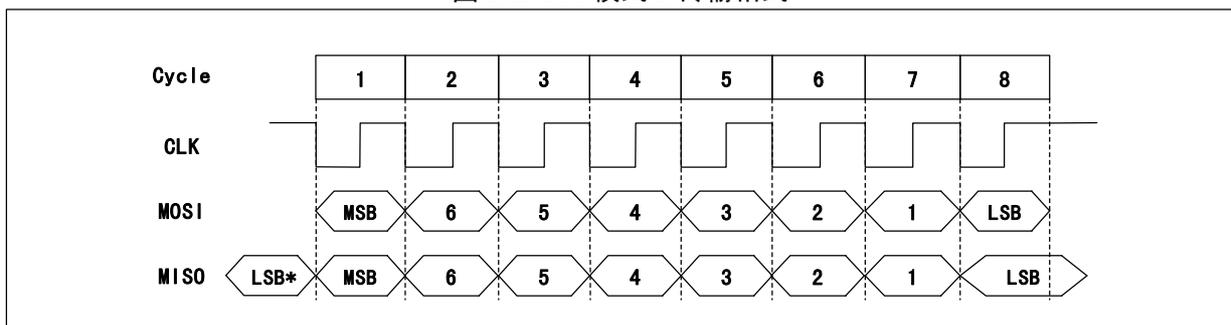
模式 0: 时钟空闲电平为低电平，并在时钟第 1 个边沿开始采样 (上升沿采样)。RB_SPI_MST_SCK_MOD=0

图 6-1 SPI 模式 0 传输格式



模式 3: 时钟空闲电平为高电平，并在时钟第 2 个边沿开始采样 (上升沿采样)。RB_SPI_MST_SCK_MOD=1

图 6-2 SPI 模式 3 传输格式



6.3 SPI 功能描述

6.3.1 SPI 主机模式

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_MOD) 的 RB_SPI_MODE_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_MOD) 的 RB_SPI_MST_SCK_MOD 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_MOD) 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_MOD) 的 RB_SPI_MOSI_OE 位和 RB_SPI_SCK_OE 位为 1，RB_SPI_MISO_OE 位为 0，并设置 PA 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_MOD) 的 RB_SPI_FIFO_DIR 位为 0，配置 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_SPIx_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_SPIx_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_MOD) 的 RB_SPI_FIFO_DIR 位为 1，配置 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。

6.3.2 SPI 从机模式

SPI 从机模式下，SCK 引脚用于接收连接的 SPI 主机的串行时钟，需要选择硬件片选引脚。

配置步骤：

1. 设置 SPI 模式配置寄存器 (R8_SPIx_CTRL_MOD) 的 RB_SPI_MODE_SLAVE 位为 1，配置 SPI 为从机模式；
2. 根据需要设置配置寄存器 (R8_SPIx_CTRL_MOD) 的 RB_SPI_SLV_CMD_MOD 位，选择从机首字节模式或数据流模式；
3. 设置 SPI 模式配置寄存器 (R8_SPIx_CTRL_MOD) 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 R8_SPI0_CTRL_MOD 的 RB_SPI_MOSI_OE 和 RB_SPI_SCK_OE 为 0，RB_SPI_MISO_OE 为 1，并设置 GPIO 方向配置寄存器 (R32_PA/PB_DIR) 使 MOSI 引脚、SCK 引脚和 SCS 引脚为输入，MISO 引脚为输入（支持总线下多个从机连接，被片选后 MISO 会自动切换为输出，也支持一主一从）或输出（仅用于

一主一从对连)。在 SPI 从机模式下 MISO 的 I/O 引脚方向，除了能够由 GPIO 方向配置寄存器设为输出之外，还支持 SPI 片选有效期间自动切换为输出，但其输出数据由 RB_SPI_MISO_OE 选择，为 1 时输出 SPI 数据，为 0 时输出 GPIO 数据输出寄存器的数据。建议，设置 MISO 引脚为输入，使 MISO 在片选无效时不输出，便于多机操作时共享 SPI 总线；

5. 可选的，设置 SPI0 从机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），控制器得到外部 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）。R8_SPI0_SLAVE_PRE 的位 7 将在 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 片选就能快速获得 R8_SPI0_SLAVE_PRE 的位 7 的值（通常是向主机提供一个忙状态，便于主机快速查询）；

6. 可选步骤，如果启用 DMA，那么需将收发缓冲区起始地址写入 R16_SPI_DMA_BEG，结束地址（不含）写入 R16_SPI_DMA_END，必须在设置完 RB_SPI_FIFO_DIR 后才能置位 RB_SPI_DMA_ENABLE。

数据发送过程：

1. 设置 SPI 模式配置寄存器 (R8_SPIx_CTRL_MOD) 的 RB_SPI_FIFO_DIR 位为 0，配置 FIFO 方向为输出；
2. 将发送数据写入到 FIFO 寄存器 (R8_SPIx_FIFO) 中，由外部 SPI 主机决定何时取走数据；如果是启用 DMA，则由 DMA 自动加载 FIFO 完成此步骤；
3. 查询 R8_SPI0_FIFO_COUNT，如果未满载则继续向 FIFO 写入待发送的数据。

数据接收过程：

1. 设置 SPI 模式配置寄存器 (R8_SPIx_CTRL_MOD) 的 RB_SPI_FIFO_DIR 位为 1，配置 FIFO 方向为输入；
2. 查询 R8_SPI0_FIFO_COUNT，如果不空则说明已接收到数据，通过读取 R8_SPI0_FIFO 取走数据；如果是启用 DMA，则由 DMA 自动读取 FIFO 完成此步骤；
3. 单个字节的数据接收，也可以不使用 FIFO，可以直接读取 R8_SPI0_BUFFER。

6.4 寄存器描述

SPI0 相关寄存器物理基地址为：0x40004000

SPI1 相关寄存器物理基地址为：0x40004400

表 6-1 SPI0 相关寄存器列表

名称	偏移地址	描述	复位值
R8_SPI0_CTRL_MOD	0x00	SPI0 模式配置寄存器	0x02
R8_SPI0_CTRL_CFG	0x01	SPI0 配置寄存器	0x00
R8_SPI0_INTER_EN	0x02	SPI0 中断使能寄存器	0x00
R8_SPI0_CLOCK_DIV	0x03	SPI0 主机模式时钟分频寄存器	0x10
R8_SPI0_SLAVE_PRE		SPI0 设备模式预置数据寄存器	
R8_SPI0_BUFFER	0x04	SPI0 数据缓冲区	0xFF
R8_SPI0_RUN_FLAG	0x05	SPI0 工作状态寄存器	0x00
R8_SPI0_INT_FLAG	0x06	SPI0 中断标志寄存器	0x00
R8_SPI0_FIFO_COUNT	0x07	SPI0 收发 FIFO 计数寄存器	0x00

R16_SPI0_TOTAL_CNT	0x0C	SPI0 收发数据长度寄存器	0x0000
R8_SPI0_FIFO	0x10	SPI0 FIFO 寄存器	0xXX
R8_SPI0_FIFO_COUNT1	0x13	SPI0 收发 FIFO 计数寄存器	0x00
R32_SPI0_DMA_NOW	0x14	SPI0 DMA 缓冲区当前地址	0xXXXX
R32_SPI0_DMA_BEG	0x18	SPI0 DMA 缓冲区起始地址	0xXXXX
R32_SPI0_DMA_END	0x1C	SPI0 DMA 缓冲区结束地址	0xXXXX

表 6-2 SPI1 相关寄存器列表

名称	偏移地址	描述	复位值
R8_SPI1_CTRL_MOD	0x00	SPI1 模式配置寄存器	0x02
R8_SPI1_CTRL_CFG	0x01	SPI1 配置寄存器	0x00
R8_SPI1_INTER_EN	0x02	SPI1 中断使能寄存器	0x00
R8_SPI1_CLOCK_DIV	0x03	SPI1 主机模式时钟分频寄存器	0x10
R8_SPI1_SLAVE_PRE		SPI1 设备模式预置数据寄存器	
R8_SPI1_BUFFER	0x04	SPI1 数据缓冲区	0xXX
R8_SPI1_RUN_FLAG	0x05	SPI1 工作状态寄存器	0x00
R8_SPI1_INT_FLAG	0x06	SPI1 中断标志寄存器	0x00
R8_SPI1_FIFO_COUNT	0x07	SPI1 收发 FIFO 计数寄存器	0x00
R16_SPI1_TOTAL_CNT	0x0C	SPI1 收发数据长度寄存器	0x0000
R8_SPI1_FIFO	0x10	SPI1 FIFO 寄存器	0xXX
R8_SPI1_FIFO_COUNT1	0x13	SPI1 收发 FIFO 计数寄存器	0x00
R32_SPI1_DMA_NOW	0x14	SPI1 DMA 缓冲区当前地址	0xXXXX
R32_SPI1_DMA_BEG	0x18	SPI1 DMA 缓冲区起始地址	0xXXXX
R32_SPI1_DMA_END	0x1C	SPI1 DMA 缓冲区结束地址	0xXXXX

SPI 模式配置寄存器 (R8_SPIx_CTRL_MOD) (x=0/1)

位	名称	访问	描述	复位值
7	RB_SPI_MISO_OE	RW	MISO 引脚输出使能 (可在 2 线模式数据线切换方向使用)。 1: MISO 引脚输出使能; 0: MISO 引脚输出禁止。	0
6	RB_SPI_MOSI_OE	RW	MOSI 引脚输出使能。 1: MOSI 引脚输出使能; 0: MOSI 引脚输出禁止。	0
5	RB_SPI_SCK_OE	RW	SCK 引脚输出使能。 1: SCK 引脚输出使能; 0: SCK 引脚输出禁止。	0
4	RB_SPI_FIFO_DIR	RW	FIFO 方向设置。 1: 输入模式 (指示接收数据); 0: 输出模式 (指示发送数据)。	0
3	RB_SPI_SLV_CMD_MOD	RW	SPI 从机模式首字节方式选择。 1: 首字节命令模式; 0: 数据流模式。 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, 且标志寄存器中的 RB_SPI_IF_FST_BYTE 位将置 1。 该位仅在从机模式下有效。	0
	RB_SPI_MST_SCK_MOD	RW	主机模式时钟空闲方式选择。 1: 模式 3 (空闲时 SCK 为高电平);	0

			0: 模式 0 (空闲时 SCK 为低电平)。 该位仅在主机模式下有效。	
2	RB_SPI_2WIRE_MOD	RW	从机模式 2 线或 3 线 SPI 模式配置位: 1: 2 线模式/半双工 (SCK, MISO); 0: 3 线模式/全双工 (SCK, MOSI, MISO)。	0
1	RB_SPI_ALL_CLEAR	RW	FIFO/计数器/中断标志寄存器清零。 1: 强制清空或清零; 0: 不清除。	1
0	RB_SPI_MODE_SLAVE	RW	SPI 主从模式选择位: 1: 从机模式; 0: 主机模式。	0

SPI 配置寄存器 (R8_SPIx_CTRL_CFG) (x=0/1)

位	名称	访问	描述	复位值
[7:6]	Reserved	RO	保留。	0
5	RB_SPI_BIT_ORDER	RW	SPI 数据位序选择。 1: 低位在前; 0: 高位在前。	0
4	RB_SPI_AUTO_IF	RW	使能访问 BUFFER/FIFO 自动清除标志位 RB_SPI_IF_BYTE_END 的功能。 1: 使能; 0: 禁止。	0
3	Reserved	RO	保留。	0
2	RB_SPI_DMA_LOOP	RW	DMA 地址循环使能。 1: 循环; 0: 单次。	0
1	Reserved	RO	保留。	0
0	RB_SPI_DMA_ENABLE	RW	DMA 功能使能。 1: 使能 DMA; 0: 禁止 DMA。	0

注: 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_SPIx_DMA_BEG) 和 DMA 结束地址寄存器 (R16_SPIx_DMA_END)。

SPI 中断使能寄存器 (R8_SPIx_INTER_EN) (x=0/1)

位	名称	访问	描述	复位值
7	RB_SPI_IE_FST_BYTE	RW	从机模式的首字节命令模式下, 首字节接收中断使能。 1: 使能中断; 0: 禁止中断。	0
[6:5]	Reserved	RO	保留。	0
4	RB_SPI_IE_FIFO_OV	RW	FIFO 溢出 (接收时 FIFO 满或发送时 FIFO 空) 中断使能。 1: 使能中断; 0: 禁止中断。	0
3	RB_SPI_IE_DMA_END	RW	DMA 结束中断使能。 1: 使能中断; 0: 禁止中断。	0
2	RB_SPI_IE_FIFO_HF	RW	FIFO 使用过半中断使能。 1: 使能中断; 0: 禁止中断。	0
1	RB_SPI_IE_BYTE_END	RW	SPI 单字节传输完成中断使能。 1: 使能中断; 0: 禁止中断。	0
0	RB_SPI_IE_CNT_END	RW	SPI 全部字节传输完成中断使能。 1: 使能中断; 0: 禁止中断。	0

SPI 主机模式时钟分频寄存器 (R8_SPIx_CLOCK_DIV) (x=0/1)

位	名称	访问	描述	复位值
[7:0]	R8_SPI_CLOCK_DIV	RW	主机模式分频系数，最小值为 2 SPI 时钟频率 $F_{SPI} = \text{主频 } F_{sys}/\text{分频系数}$ 。	10h

SPI 设备模式预置数据寄存器 (R8_SPIx_SLAVE_PRE)

位	名称	访问	描述	复位值
[7:0]	R8_SPIx_SLAVE_PRE	RW	从机模式下，预置的首个返回数据。 用于接收首字节数据后的返回数据。	10h

SPI 数据缓冲区 (R8_SPIx_BUFFER) (x=0/1)

位	名称	访问	描述	复位值
[7:0]	R8_SPIx_BUFFER	RW	SPI 数据发送和接收缓冲区	X

SPI 工作状态寄存器 (R8_SPIx_RUN_FLAG) (x=0/1)

位	名称	访问	描述	复位值
7	RB_SPI_SLV_SELECT	RO	从机模式被片选状态。 1: 正被选中; 0: 没被选中。	0
6	RB_SPI_SLV_CS_LOAD	RO	从机模式被片选后首次加载状态。 1: 正在加载 R8_SPIx_SLAVE_PRE; 0: 尚未加载或者已完成 (可修改预加载值)。	0
5	RB_SPI_FIFO_READY	RO	FIFO 准备完成状态。 1: FIFO 就绪 (R16_SPIx_TOTAL_CNT 非 0, 且接收时 FIFO 未满载或发送时 FIFO 不空) 0: FIFO 未准备好。	0
4	RB_SPI_SLV_CMD_ACT	RO	从机模式下命令接收完成状态位, 即交换完首字节数据。 1: 刚刚交换完成的是首字节; 0: 首字节尚未交换或不是首字节。	0
[3:0]	Reserved	RO	保留。	0

SPI 中断标志寄存器 (R8_SPIx_INT_FLAG) (x=0/1)

位	名称	访问	描述	复位值
7	RB_SPI_IF_FST_BYTE	RW1	从机模式下，接收到首字节标志，写 1 清 0。 1: 已接收到; 0: 未接收到。	0
6	RB_SPI_FREE	RO	当前 SPI 空闲状态。 1: 空闲; 0: 非空闲。	1
5	Reserved	RO	保留。	0
4	RB_SPI_IF_FIFO_OV	RW1	FIFO 溢出 (接收时 FIFO 满或发送时 FIFO 空) 标志位，写 1 清 0。 1: 溢出; 0: 未溢出。	0
3	RB_SPI_IF_DMA_END	RW1	DMA 完成标志，写 1 清 0。 1: 已完成; 0: 未完成。	0
2	RB_SPI_IF_FIFO_HF	RW1	FIFO 使用过半 (接收时 $FIFO \geq 4$ 或发送时 $FIFO < 4$) 标志，写 1 清 0。 1: FIFO 使用已过半; 0: FIFO 使用未过半。	0
1	RB_SPI_IF_BYTE_END	RW1	SPI 单字节传输完成标志，写 1 清 0。	0

			1: 单字节传输完成; 0: 传输未完成。	
0	RB_SPI_IF_CNT_END	RW1	SPI 全部字节传输完成标志, 写 1 清 0。 1: 全部传输完成; 0: 传输未完成。	0

SPI 收发 FIFO 计数寄存器 (R8_SPIx_FIFO_COUNT) (x=0/1)

位	名称	访问	描述	复位值
[7:0]	R8_SPIx_FIFO_COUNT	RW	当前 FIFO 中字节计数。	0

SPI 收发 FIFO 计数寄存器 (R8_SPIx_FIFO_COUNT1) (x=0/1)

位	名称	访问	描述	复位值
[7:0]	R8_SPIx_FIFO_COUNT1	RW	当前 FIFO 中字节计数, 等效寄存器 R8_SPIx_FIFO_COUNT。	0

SPI 收发数据总长度寄存器 (R16_SPIx_TOTAL_CNT) (x=0/1)

位	名称	访问	描述	复位值
[15:0]	R16_SPIx_TOTAL_CNT	RW	主机模式下 SPI 数据收发总字节数, 低 12 位有效。在使用 DMA 时一次最多可以发送 4095 个字节。不支持从机模式。	0

SPI FIFO 寄存器 (R8_SPIx_FIFO) (x=0/1)

位	名称	访问	描述	复位值
[7:0]	R8_SPIx_FIFO	RO/ WO	SPI FIFO 寄存器。FIFO 大小为 8 个字节。	0

寄存器 R8_SPIx_BUFFER 和 R8_SPIx_FIFO 均为 SPI 数据相关寄存器, 主要区别在于:

读 R8_SPIx_BUFFER 是取自 SPI 最近一次交换到的数据, 不影响 FIFO 和 R8_SPIx_FIFO_COUNT, 主机模式下写 R8_SPIx_BUFFER 是直接发送该字节, 从机模式下写操作未定义;

读 R8_SPIx_FIFO 是取自 FIFO 中最早交换到的数据, 将减少 FIFO 和 R8_SPIx_FIFO_COUNT, 写 R8_SPIx_FIFO 是将数据暂存到 FIFO 中, 在从机模式下由外部 SPI 主机决定何时取走, 在主机模式下当 R16_SPIx_TOTAL_CNT 非 0 时自动启动发送。

SPI DMA 缓冲区当前地址 (R32_SPIx_DMA_NOW)

位	名称	访问	描述	复位值
[31:18]	Reserved	RO	保留。	0
[17:0]	R16_SPIx_DMA_NOW	RW	DMA 数据缓冲区当前地址。 可用于计算已转换次数, 计算方法: CNT=R16_SPIx_DMA_NOW-R16_SPIx_DMA_BEG	X

SPI DMA 缓冲区起始地址 (R32_SPIx_DMA_BEG)

位	名称	访问	描述	复位值
[31:18]	Reserved	RO	保留。	0
[17:0]	R16_SPIx_DMA_BEG	RW	DMA 数据缓冲区起始地址。地址必须 4 字节对齐。	X

注: 此 DMA 地址可范围 RAMS 和 RAMX 区域。

SPI DMA 缓冲区结束地址 (R32_SPIx_DMA_END)

位	名称	访问	描述	复位值
[31:18]	Reserved	RO	保留。	0
[17:0]	R16_SPIx_DMA_END	RW	DMA 数据缓冲区结束地址 (不含)。地址必须 4 字节对齐。	X

注：此 DMA 地址可范围 RAMS 和 RAMX 区域。

第 7 章 通用异步收发器 (UART)

系统提供了 4 组全双工的异步串口, UART0/1/2/3。支持全双工和半双工串口通讯, 其中 UART0 提供发送状态引脚用于切换 RS485, 并且支持 MODEM 调制解调器信号 CTS、DSR、RI、DCD、DTR、RTS。

7.1 主要特征

- 兼容 16C550 异步串口并且有所增强
- 支持 5、6、7 或者 8 个数据位以及 1 或者 2 个停止位
- 支持奇、偶、无校验、空白 0、标志 1 等校验方式
- 可编程通讯波特率, 支持 115200bps 以及最高达 6Mbps 的通讯波特率
- 内置 8 个字节的 FIFO 先进先出缓冲器, 支持 4 个 FIFO 触发级
- UART0 支持 MODEM 调制解调器信号 CTS、DSR、RI、DCD、DTR、RTS, 可转成 RS232 电平
- 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
- 支持串口帧错误检测、支持 Break 线路间隔检测
- 支持全双工和半双工串口通讯, UART0 提供发送状态引脚用于切换 RS485

7.2 功能描述

UART0/1/2/3 输出引脚都是 3.3V 的 TTL 电平。异步串口方式下引脚包括: 数据传输引脚 (支持 UART0/1/2/3) 和 MODEM 联络信号引脚 (只支持 UART0)。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。

4 组 UART 各自拥有独立的收发缓冲区及 8 字节 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。

模块最高支持 7.5Mbps 的波特率, 对于常用的波特率档位, 需要选择合适的模块基准时钟和分频系数来减少硬件计算误差。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%。

7.2.1 波特率计算

1) 计算基准时钟, 设置 R8_UART0_DIV 寄存器, 最大值 127;

2) 计算波特率, 设置 R16_UART0_DL 寄存器;

波特率公式 = $F_{sys} * 2 / R8_UART0_DIV / 16 / R16_UART0_DL$.

7.2.2 串口发送

串口发送的“THR 寄存器空”中断 (IIR 寄存器的低 4 位为 02H) 是指发送 FIFO 空。当读取 IIR 寄存器后, 该中断被清除, 或者当向 THR 写入下一个数据后, 该中断也能被清除。如果仅仅是向 THR 写入一个字节, 那么由于该字节很快被转移到发送移位寄存器 TSR 中开始发送, 所以很快会再次产生发送 THR 寄存器空中断的请求, 此时可以写入下一个准备发送的数据。当 TSR 寄存器中的数据被全部

移出后，串口发送才真正完成，此时 LSR 寄存器的 RB_LSR_TX_ALL_EMP 位变为 1 有效。

在中断触发方式下，当收到串口发送保持寄存器 THR 空的中断后，如果已使能 FIFO，那么可以向 THR 寄存器及 FIFO 一次写入最多 8 字节，然后控制器会按顺序自动发送；如果禁止 FIFO，那么一次只能写入一个字节；如果没有数据需要发送，那么可以直接退出（之前读取 IIR 时已经自动清除中断）。

在查询方式下，可以根据 LSR 寄存器的 RB_LSR_TX_FIFO_EMP 位判断发送 FIFO 是否为空，当此位为 1 则可以向 THR 寄存器及 FIFO 写入数据，如果使能 FIFO，那么一次可以写入最多 8 个字节。

7.2.3 串口接收

串口接收数据可用中断（IIR 寄存器的低 4 位为 04H）是指接收 FIFO 中的已有数据字节数已经到或超过由 FCR 寄存器的 RB_FCR_FIFO_TRIG 设置选择的 FIFO 触发点。当从 RBR 读取数据使 FIFO 字数低于 FIFO 触发点时，该中断被清除。

串口接收数据超时中断（IIR 寄存器的低 4 位为 0CH）是指接收 FIFO 中至少有一个字节的数据，并且从上一次串口接收到数据和从上一次被单片机取走数据开始，已经等待了相当于接收 4 个数据的时间。当再次接收到一个新的数据后，该中断被清除，或者当单片读取一次 RBR 寄存器后，该中断也能被清除。当接收 FIFO 全空时，LSR 寄存器的 RB_LSR_DATA_RDY 位为 0，当接收 FIFO 中有数据时，RB_LSR_DATA_RDY 位为 1 有效。

在中断触发方式下，当收到串口接收数据超时的中断后，可以读取 R8_UARTx_RFC 寄存器查询当前 FIFO 中剩余数据计数，直接读取全部数据，或者不断查询 LSR 寄存器的 RB_LSR_DATA_RDY 位，如果此位有效则读数据，直到此位无效。当收到串口接收数据可用的中断后，可以先从 RBR 寄存器读取由 FCR 寄存器的 RB_FCR_FIFO_TRIG 设定的字节数，然后直接读取该字节个数的数据，或者也可以根据 RB_LSR_DATA_RDY 位和 R8_UARTx_RFC 寄存器读取当前 FIFO 中所有数据。

在查询方式下，单片机可以根据 LSR 寄存器的 RB_LSR_DATA_RDY 位判断接收 FIFO 是否为空，和读取 R8_UARTx_RFC 寄存器获取当前 FIFO 中数据计数，来获取串口接收的所有数据。

7.2.4 硬件流控制

硬件流控制包括自动 CTS（MCR 寄存器的 RB_MCR_AU_FLOW_EN 为 1）和自动 RTS（MCR 寄存器的 RB_MCR_AU_FLOW_EN 和 RB_MCR_RTS 都为 1）。

如果使能自动 CTS，那么 CTS 引脚在串口发送数据之前必须有效。串口发送器在发送下一个数据之前会检测 CTS 引脚，当 CTS 引脚状态有效时，发送器发送下一个数据。为了使发送器停止发送后面的数，CTS 引脚必须在当前发送的最后一个停止位的中间时刻之前被无效。自动 CTS 功能减少了向单片机系统申请的中断。当使能硬件流控制后，由于控制器会根据 CTS 引脚状态自动控制发送器，所以 CTS 引脚电平的改变不会触发 MODEM 中断。如果使能自动 RTS，那么仅当 FIFO 中有足够空间接收数据时才使 RTS 引脚输出有效，而在接收 FIFO 满时使 RTS 引脚输出无效。如果接收 FIFO 中的数据被全部取走或清空，那么 RTS 引脚输出有效。当到达接收 FIFO 的触发点时（接收 FIFO 中已有字节数不少于 FCR 寄存器的 RB_FCR_FIFO_TRIG 设定的字节数），RTS 引脚输出无效，并且允许对方发送器在 RTS 引脚无效后再发送一个另外的数据。一旦接收 FIFO 被取空数据，RTS 引脚就会自动重有效，从而使对方的发送器恢复发送。如果自动 CTS 和自动 RTS 都被使能（MCR 寄存器的 RB_MCR_AU_FLOW_EN 和 RB_MCR_RTS 都为 1），那么当己方的 RTS 引脚连接对方的 CTS 引脚时，除非己方的接收 FIFO 中有足够的空间，否方不会发送数据。因此，通过这种硬件流控制，可以避免串口接收时的 FIFO 溢出和超时

错误。

7.3 寄存器描述

UART0 相关寄存器物理基地址为：0x40003000

UART1 相关寄存器物理基地址为：0x40003400

UART2 相关寄存器物理基地址为：0x40003800

UART3 相关寄存器物理基地址为：0x40003C00

表 7-1 UART0 相关寄存器列表

名称	偏移地址	描述	复位值
R8_UART0_MCR	0x00	调制解调器 MODEM 控制寄存器	0x00
R8_UART0_IER	0x01	中断使能寄存器	0x00
R8_UART0_FCR	0x02	FIFO 控制寄存器	0x00
R8_UART0_LCR	0x03	线路控制寄存器	0x00
R8_UART0_IIR	0x04	中断识别寄存器	0x01
R8_UART0_LSR	0x05	线路状态寄存器	0xC0
R8_UART0_MSR	0x06	调制解调器 MODEM 状态寄存器	0xX0
R8_UART0_RBR	0x08	接收缓冲寄存器	0xXX
R8_UART0_THR	0x08	发送保持寄存器	0xXX
R8_UART0_RFC	0x0A	接收 FIFO 计数寄存器	0xXX
R8_UART0_TFC	0x0B	发送 FIFO 计数寄存器	0xXX
R16_UART0_DL	0x0C	波特率除数锁存器	0XXXXX
R8_UART0_DIV	0x0E	预分频除数寄存器	0xXX
R8_UART0_ADR	0x0F	从机地址寄存器	0xFF

表 7-2 UART1 相关寄存器列表

名称	偏移地址	描述	复位值
R8_UART1_MCR	0x00	调制解调器 MODEM 控制寄存器	0x00
R8_UART1_IER	0x01	中断使能寄存器	0x00
R8_UART1_FCR	0x02	FIFO 控制寄存器	0x00
R8_UART1_LCR	0x03	线路控制寄存器	0x00
R8_UART1_IIR	0x04	中断识别寄存器	0x01
R8_UART1_LSR	0x05	线路状态寄存器	0xC0
R8_UART1_RBR	0x08	接收缓冲寄存器	0xXX
R8_UART1_THR	0x08	发送保持寄存器	0xXX
R8_UART1_RFC	0x0A	接收 FIFO 计数寄存器	0xXX
R8_UART1_TFC	0x0B	发送 FIFO 计数寄存器	0xXX
R16_UART1_DL	0x0C	波特率除数锁存器	0XXXXX
R8_UART1_DIV	0x0E	预分频除数寄存器	0xXX

表 7-3 UART2 相关寄存器列表

名称	偏移地址	描述	复位值
R8_UART2_MCR	0x00	调制解调器 MODEM 控制寄存器	0x00
R8_UART2_IER	0x01	中断使能寄存器	0x00
R8_UART2_FCR	0x02	FIFO 控制寄存器	0x00
R8_UART2_LCR	0x03	线路控制寄存器	0x00
R8_UART2_IIR	0x04	中断识别寄存器	0x01

R8_UART2_LSR	0x05	线路状态寄存器	0xC0
R8_UART2_RBR	0x08	接收缓冲寄存器	0xFF
R8_UART2_THR	0x08	发送保持寄存器	0xFF
R8_UART2_RFC	0x0A	接收 FIFO 计数寄存器	0xFF
R8_UART2_TFC	0x0B	发送 FIFO 计数寄存器	0xFF
R16_UART2_DL	0x0C	波特率除数锁存器	0xFFFF
R8_UART2_DIV	0x0E	预分频除数寄存器	0xFF

表 7-4 UART3 相关寄存器列表

名称	偏移地址	描述	复位值
R8_UART3_MCR	0x00	调制解调器 MODEM 控制寄存器	0x00
R8_UART3_IER	0x01	中断使能寄存器	0x00
R8_UART3_FCR	0x02	FIFO 控制寄存器	0x00
R8_UART3_LCR	0x03	线路控制寄存器	0x00
R8_UART3_IIR	0x04	中断识别寄存器	0x01
R8_UART3_LSR	0x05	线路状态寄存器	0xC0
R8_UART3_RBR	0x08	接收缓冲寄存器	0xFF
R8_UART3_THR	0x08	发送保持寄存器	0xFF
R8_UART3_RFC	0x0A	接收 FIFO 计数寄存器	0xFF
R8_UART3_TFC	0x0B	发送 FIFO 计数寄存器	0xFF
R16_UART3_DL	0x0C	波特率除数锁存器	0xFFFF
R8_UART3_DIV	0x0E	预分频除数寄存器	0xFF

调制解调器 MODEM 控制寄存器 (R8_UARTx_MCR) (x=0/1/2/3)

位	名称	访问	描述	复位值
7	RB_MCR_HALF	RW	半双工收发模式使能 (仅 UART0 支持)。 1: 使能半双工收发模式, 发送优先, 不发送时为接收; 0: 禁止半双工收发模式。	0
6	RB_MCR_TNOW	RW	DTR 引脚输出正在发送状态 (TNOW) 使能 (仅 UART0 支持), 可以用于控制 RS485 收发切换。 1: 使能; 0: 禁止。	0
5	RB_MCR_AU_FLOW_EN	RW	CTS 和 RTS 硬件自动流使能 1: 使能; 0: 禁止。 流控模式下, 如果此位为 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, 当此位为 1 时的 CTS 输入状态变化不会产生 MODEM 状态中断。如果此位为 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。	0
4	RB_MCR_LOOP	RW	内部回路的测试模式使能 (仅 UART0 支持)。 1: 使能; 0: 禁止。 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	0

			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。	
3	RB_MCR_OUT2	RW	串口的中断请求使能控制。 1: 使能; 0: 禁止。	0
2	RB_MCR_OUT1	RW	用户自定义 MODEM 控制位 (仅 UART0 支持), 没有连接实际输出引脚: 1: 置高; 0: 置低。	0
1	RB_MCR_RTS	RW	RTS 引脚输出电平控制 (仅 UART0 支持)。 1: RTS 信号输出有效 (低电平); 0: RTS 信号输出高电平 (默认)。	0
0	RB_MCR_DTR	RW	DTR 引脚输出电平控制 (仅 UART0 支持)。 1: DTR 信号输出有效 (低电平); 0: DTR 信号输出高电平 (默认)。	0

中断使能寄存器 (R8_UARTx_IER) (x=0/1/2/3)

位	名称	访问	描述	复位值
7	RB_IER_RESET	WZ	串口软件复位控制, 硬件自动清零。 1: 复位串口; 0: 正常工作。	0
6	RB_IER_TXD_EN	RW	串口 TXD 引脚输出使能。 1: 使能输出; 0: 禁止。	0
5	RB_IER_RTS_EN	RW	RTS 引脚输出使能 (仅 UART0 支持)。 1: 使能输出; 0: 禁止。	0
4	RB_IER_DTR_EN	RW	DTR 引脚输出使能 (仅 UART0 支持)。 1: 使能输出; 0: 禁止。	0
3	RB_IER_MODEM_CHG	RW	UART0 的调制解调器输入状态变化中断使能 (仅 UART0 支持)。 1: 使能中断; 0: 禁止中断。	0
2	RB_IER_LINE_STAT	RW	接收线路状态中断使能。 1: 使能中断; 0: 禁止中断。	0
1	RB_IER_THR_EMPTY	RW	发送保持寄存器空中断使能。 1: 使能中断; 0: 禁止中断。	0
0	RB_IER_RECV_RDY	RW	接收数据中断使能。 1: 使能中断; 0: 禁止中断。	0

FIFO 控制寄存器 (R8_UARTx_FCR) (x=0/1/2/3)

位	名称	访问	描述	复位值
[7:6]	RB_FCR_FIFO_TRIG	RW	接收 FIFO 的中断和硬件流控制的触发点设置域。 00: 1 字节; 01: 2 字节; 10: 4 字节; 11: 7 字节。 该域用来设置接收 FIFO 的中断和硬件流控制的触发点, 例如: 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。	0
[5:3]	Reserved	RO	保留	0
2	RB_FCR_TX_FIFO_CLR	WZ	发送 FIFO 数据清空使能, 硬件自动清零。 1: 清空发送 FIFO 中的数据 (不含 TSR);	0

			0: 不清空发送 FIFO 中的数据。	
1	RB_FCR_RX_FIFO_CLR	WZ	接收 FIFO 数据清空使能, 硬件自动清零。 1: 清空接收 FIFO 中的数据 (不含 RSR); 0: 不清空接收 FIFO 中的数据。	0
0	RB_FCR_FIFO_EN	RW	FIFO 使能。 1: 启用 FIFO, 内部 FIFO 大小 8 字节; 0: 禁用 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。	0

线路控制寄存器 (R8_UARTx_LCR) (x=0/1/2/3)

位	名称	访问	描述	复位值
7	RB_LCR_DLAB/ RB_LCR_GP_BIT	RW	串口通用位, 用户自定义。	0
6	RB_LCR_BREAK_EN	RW	强制产生 BREAK 线路间隔使能。 1: 强制产生; 0: 不产生。	0
[5:4]	RB_LCR_PAR_MOD	RW	奇偶校验位格式设置, 仅当 RB_LCR_PAR_EN 位为 1 时有效。 00: 奇校验; 01: 偶校验; 10: 标志位 (MARK, 置 1); 11: 空白位 (SPACE, 清 0)。	0
3	RB_LCR_PAR_EN	RW	奇偶校验位使能。 1: 允许发送时产生和接收时校验奇偶校验位; 0: 无奇偶校验位。	0
2	RB_LCR_STOP_BIT	RW	停止位格式设置。 1: 两个停止位; 0: 一个停止位。	0
[1:0]	RB_LCR_WORD_SZ	RW	串口数据长度设置。 00: 5 个数据位; 01: 6 个数据位; 10: 7 个数据位; 11: 8 个数据位。	0

中断识别寄存器 (R8_UARTx_IIR) (x=0/1/2/3)

位	名称	访问	描述	复位值
[7:6]	RB_IIR_FIFO_ID	RO	串口 FIFO 启用状态。 11: FIFO 已启用; 00: FIFO 未启用。	0
[5:4]	Reserved	RO	保留。	0
[3:0]	RB_IIR_INT_MASK	RO	中断标志域: 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。具体请参看表 7-3。	0
0	RB_IIR_NO_INT	RO	串口无中断标志。 1: 无中断; 0: 有中断。	1

表 7-3 IIR 寄存器中 RB_IIR_INT_MASK 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,且该地址与预置从机地址相匹配或是广播地址。 注: 该中断只适用 UART0。	读 IIR 或禁用多机模式
0	1	1	0	1	接收线路状态	OVER_ERR、PAR_ERR、FRAM_ERR、BREAK_ERR	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, 或者, RB_IER_THR_EMPTY 位从 0 变 1 触发。	读 IIR 或写 THR
0	0	0	0	4	MODEM 输入变化	Δ CTS、 Δ DSR、 Δ RI、 Δ DCD 置 1 触发。	读 MSR

线路状态寄存器 (R8_UARTx_LSR) (x=0/1/2/3)

位	名称	访问	描述	复位值
7	RB_LSR_ERR_RX_FIFO	RO	接收 FIFO 错误标志位: 1: 接收 FIFO 中存在至少一个 PAR_ERR、FRAM_ERR 或 BREAK_ERR 错误; 0: 接收 FIFO 不存在错误。	0
6	RB_LSR_TX_ALL_EMP	RO	发送保持寄存器 THR 和发送移位寄存器 TSR 全空标志。 1: 两者全空; 0: 两者非全空。	1
5	RB_LSR_TX_FIFO_EMP	RO	发送 FIFO 空标志。 1: 发送 FIFO 空; 0: 发送 FIFO 非空。	1
4	RB_LSR_BREAK_ERR	RZ	BREAK 线路间隔检测标志。 1: 检测到 BREAK 线路间隔; 0: 未检测到 BREAK 线路间隔。	0
3	RB_LSR_FRAME_ERR	RZ	数据帧错误标志。 1: 正在从接收 FIFO 中读取的数据存在帧错误, 缺少有效停止位; 0: 当前读取的数据帧正确。	0
2	RB_LSR_PAR_ERR	RZ	接收数据奇偶校验错误标志。 1: 正在从接收 FIFO 中读取的数据的奇偶校验错误; 0: 当前读取的数据奇偶校验正确。	0
1	RB_LSR_OVER_ERR	RZ	接收 FIFO 缓冲区溢出标志。 1: 已溢出; 0: 未溢出。	0
0	RB_LSR_DATA_RDY	RO	接收 FIFO 中有数据标志。 1: FIFO 中有数据; 0: 无数据。	0

调制解调器 MODEM 状态寄存器 (R8_UART0_MSR) (仅 UART0 支持)

位	名称	访问	描述	复位值
7	RB_MSR_DCD	RO	DCD 引脚状态。 1: DCD 引脚有效 (低电平); 0: DCD 引脚无效 (高电平)。	x
6	RB_MSR_RI	RO	RI 引脚状态。 1: RI 引脚有效 (低电平); 0: RI 引脚无效 (高电平)。	x
5	RB_MSR_DSR	RO	DSR 引脚状态。 1: DSR 引脚有效 (低电平); 0: DSR 引脚无效 (高电平)。	x

4	RB_MSR_CTS	RO	CTS 引脚状态。 1: CTS 引脚有效 (低电平); 0: CTS 引脚无效 (高电平)。	x
3	RB_MSR_DCD_CHG	RZ	DCD 引脚输入状态变化标志。 1: 发生过变化; 0: 无变化。	0
2	RB_MSR_RI_CHG	RZ	RI 引脚输入状态变化标志。 1: 发生过变化; 0: 无变化。	0
1	RB_MSR_DSR_CHG	RZ	DSR 引脚输入状态变化标志。 1: 发生过变化; 0: 无变化。	0
0	RB_MSR_CTS_CHG	RZ	CTS 引脚输入状态变化标志。 1: 发生过变化; 0: 无变化。	0

接收缓冲寄存器 (R8_UARTx_RBR) (x=0/1/2/3)

位	名称	访问	描述	复位值
[7:0]	R8_UARTx_RBR	RO	数据接收缓冲寄存器。 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。	X

发送保持寄存器 (R8_UARTx_THR) (x=0/1/2/3)

位	名称	访问	描述	复位值
[7:0]	R8_UARTx_THR	WO	发送保持寄存器。 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。	X

接收 FIFO 计数寄存器 (R8_UARTx_RFC) (x=0/1/2/3)

位	名称	访问	描述	复位值
[7:0]	R8_UARTx_RFC	RO	当前接收 FIFO 中数据计数。最大值 8。	X

发送 FIFO 计数寄存器 (R8_UARTx_TFC) (x=0/1/2/3)

位	名称	访问	描述	复位值
[7:0]	R8_UARTx_TFC	RO	当前发送 FIFO 中数据计数。最大值 8。	X

波特率除数锁存器 (R16_UARTx_DL) (x=0/1/2/3)

位	名称	访问	描述	复位值
[15:0]	R16_UARTx_DL	RW	16 位除数用于计算波特率。 公式: 除数=串口内部基准时钟 $F_{UART}/16/\text{波特率}$ 。 例: 如果串口内部基准时钟为 1.8432MHz, 所需波特率 9600bps, 则除数=1843200/16/9600=12。	X

预分频除数寄存器 (R8_UARTx_DIV) (x=0/1/2/3)

位	名称	访问	描述	复位值
[7:0]	R8_UARTx_DIV	RW	用于计算串口的内部基准时钟，低 7 位有效。 公式：除数= $F_{sys} * 2 /$ 串口内部基准时钟，最大值 127。	X

从机地址寄存器(R8_UART0_ADR) 仅 UART0 使用

位	名称	访问	描述	复位值
[7:0]	R8_UART0_ADR	RW	串口 0 从机地址。 FFh：不使用； 其他：从机地址。	FFh

R8_UART0_ADR 预置本机作为从机时的地址，用于在多机通讯时自动比较接收到的地址，并在地址匹配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包。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_UART0_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_UART0_ADR 为 0FFH 时或者 RB_LCR_PAR_EN=0 时，禁用总线地址自动比较功能。

R8_UART0_ADR 不为 0FFH 并且 RB_LCR_PAR_EN=1 时，启用总线地址自动比较功能，同时应该配置下述参数：RB_LCR_WORD_SZ 控制域位 11b 以选择 8 个数据位方式，对于地址字节为 MARK 的情况（即数据字节的位 9 为 0），应设置 RB_LCR_PAR_MOD 控制域 10b，对于地址字节为 SPACE 的情况（即数据字节的位 9 为 1），应设置 RB_LCR_PAR_MOD 控制域 11b。

第 8 章 通用定时器 (TMRx)

系统提供了 3 个 26 位定时器, TMR0、TMR1 和 TMR2, 最长定时时间为 2^{26} 个时钟周期。所有定时器均支持捕捉、PWM 以及中断功能, 另外 TMR1 和 TMR2 支持 DMA 功能。

8.1 主要特征

- 3 个 26 位定时器, 每个定时器定时时间最大为 2^{26} 个时钟周期;
- 每个定时器都支持 PWM 功能;
- 每个定时器都支持捕捉功能;
- 每个定时器都支持定时器中断, 其中 TMR1 和 TMR2 支持 DMA 及中断;
- 捕捉功能可设置为电平变化捕捉功能和高或低电平保持时间捕捉功能;
- PWM 功能支持动态的调整 PWM 占空比设置;

8.2 功能描述

8.2.1 定时、计数功能

3 个定时器支持的最长定时时间为 2^{26} 个时钟周期。如果系统时钟周期为 96M, 则最长定时时间为: $10.4\text{ns} \times 2^{26} \approx 0.7\text{s}$ 。如果系统时钟低于 96M, 则定时时间更长。

定时功能设置如下:

1. 设置寄存器 R32_TMRx_CNT_END 为需要定时的时间值;

具体计算方法为: $\text{Time} = F_{\text{sys}} * R32_TMRx_CNT_END$

2. 设置寄存器 R8_TMRx_CTRL_MOD 中的 RB_TMR_MODE_IN 位为 0, RB_TMR_ALL_CLEAR 位为 0;

3. 将寄存器 R8_TMRx_CTRL_MOD 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_TMRx_INT_FLAG 的 RB_TMR_IF_CYC_END 位将置 1, 需写 1 清零。如果使能了相应中断位, 将触发中断服务。

8.3.2 PWM 功能

每个定时器均具有 PWM 输出功能。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 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 功能设置如下:

1. 设置 PWM 总周期寄存器 R32_TMRx_CNT_END, 最小值为 1, 该寄存器值必须大于等于 R32_TMRx_FIFO 寄存器的值;

2. 设置数据寄存器 R32_TMRx_FIFO, 最小值为 0, 对应占空比 0%, 最大值同 R32_TMRx_CNT_END, 对应占空比 100%, 占空比计算: $R32_TMRx_FIFO / R32_TMRx_CNT_END$ 。TMR1 和 TMR2 支持连续动态数据 (DMA), 可以模仿出特殊波形;

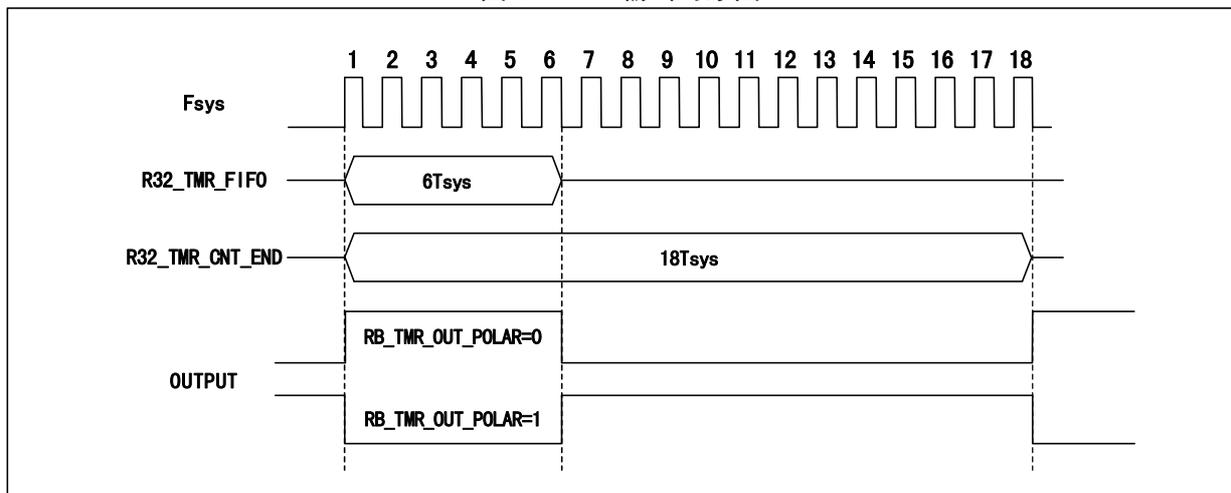
3. 将模式设置寄存器 (R8_TMRx_CTRL_MOD) 中的 RB_TMR_MODE_IN 位清 0, 启用 PWM 模式; 同时将 RB_TMR_ALL_CLEAR 位置 1 再清 0 强制清除 FIFO 和 COUNT; 设置 RB_TMR_OUT_POLAR 位选择输出极性。

如果需要设置重复次数，则根据需要设置 RB_TMR_PWM_REPEAT 域。

4. 将模式设置寄存器 (R8_TMRx_CTRL_MOD) 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 R8_TMRx_INT_FLAG 寄存器了解 PWM 是否完成以及 PWM 过程中是否产生错误;

举例: 设置 RB_TMR_OUT_POLAR 位为 0, R32_TMRx_FIFO 为 6, R32_TMRx_CNT_END 为 18, 则产生 PWM 的基本时序图如下所示, 其占空比为: $\text{PWM 占空比} = \text{R32_TMRx_FIFO} / \text{R32_TMRx_CNT_END} = 1/3$

图 8-1 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。

8.3.3 捕获功能

每个定时器均具有捕获功能, 其中 TMR1 和 TMR2 的捕获功能支持 DMA 方式存储。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:

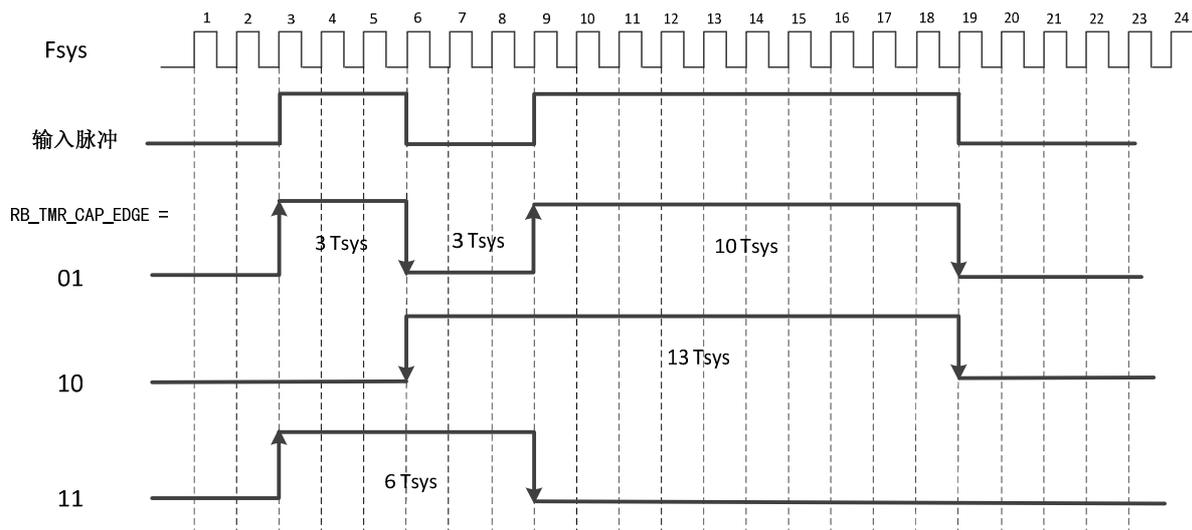
表 8-1 捕获触发模式说明表

捕获模式选择位 (RB_TMR_CAP_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_TMRx_FIFO）的有效数据最高位（位 25）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 25 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_TMRx_FIFO）的有效数据最高位（位 25）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_TMRx_CNT_END，需加上后一组数据累计为单个输入变化周期。具体说明如图 8-2 所示：

图 8-2 以系统时钟周期为捕获周期



当 RB_TMR_CAP_EDGE = 01b 时，设置为边沿触发采样，采样到的时间宽度为 3Tsys、3Tsys、10Tsys；
 当 RB_TMR_CAP_EDGE = 10b 时，设置为下降沿至下降沿采样，采样到的时间宽度为 13Tsys；
 当 RB_TMR_CAP_EDGE = 11b 时，设置为上升沿至上升沿采样，采样到的时间宽度为 6Tsys。

捕获功能设置如下：

1. 设置寄存器 R32_TMRx_CNT_END 用于设定捕获超时时间，默认最大超时时间为 2^{26} 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_TMRx_FIFO 寄存器的位 25 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_MOD) 的 RB_TMR_MODE_IN 置 1，RB_TMR_CAP_COUNT 位置 0，同时 RB_TMR_ALL_CLEAR 位清 0，用于清除 FIFO 和 COUNT。同时设置 RB_TMR_CAP_EDGE 控制域位选择捕获模式；
4. 如果需要启用中断，则将中断寄存器 R8_TMRx_INTER_EN 中相应位置 1，启动相应中断；
5. 如果需要采用 DMA 方式（只支持 TMR1 和 TMR2）保存捕获的数据，需将 R8_TMRx_CTRL_DMA 寄存器 RB_TMR_DMA_ENABLE 位置 1，使能 DMA 功能，同时将寄存器 R16_TMRx_DMA_BEG 设置为存储捕获数据缓冲区的首地址，将寄存器 R16_TMRx_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_TMRx_CTRL_MOD 的 RB_TMR_COUNT_EN 位置 1，使能定时器模块，启动捕获功能；
7. 捕获完成后，寄存器 R8_TMRx_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_TMRx_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。

8.3 寄存器描述

TMR0 相关寄存器物理基地址为：0x40002000

TMR1 相关寄存器物理基地址为：0x40002400

TMR2 相关寄存器物理基地址为：0x40002800

表 8-2 TMR0 相关寄存器列表

名称	偏移地址	描述	复位值
R8_TMR0_CTRL_MOD	0x00	模式设置寄存器	0x02
R8_TMR0_INTER_EN	0x02	中断使能寄存器	0x00
R8_TMR0_INT_FLAG	0x06	中断标志寄存器	0x00
R8_TMR0_FIFO_COUNT	0x07	FIFO 计数寄存器	0x00
R32_TMR0_COUNT	0x08	当前计数值寄存器	0x00000000
R32_TMR0_CNT_END	0x0C	计数终值设置寄存器	0x00000000
R32_TMR0_FIFO	0x10	FIFO 寄存器	0x00000000

表 8-3 TMR1 相关寄存器列表

名称	偏移地址	描述	复位值
R8_TMR1_CTRL_MOD	0x00	模式设置寄存器	0x02
R8_TMR1_CTRL_DMA	0x01	DMA 控制寄存器	0x00
R8_TMR1_INTER_EN	0x02	中断使能寄存器	0x00
R8_TMR1_INT_FLAG	0x06	中断标志寄存器	0x00
R8_TMR1_FIFO_COUNT	0x07	FIFO 计数寄存器	0x00
R32_TMR1_COUNT	0x08	当前计数值寄存器	0x00000000
R32_TMR1_CNT_END	0x0C	计数终值寄存器	0x00000000
R32_TMR1_FIFO	0x10	FIFO 寄存器	0x00000000
R32_TMR1_DMA_NOW	0x14	DMA 当前缓冲区地址	0x0000
R32_TMR1_DMA_BEG	0x18	DMA 起始缓冲区地址	0x0000
R32_TMR1_DMA_END	0x1C	DMA 结束缓冲区地址	0x0000

表 8-4 TMR2 相关寄存器列表

名称	偏移地址	描述	复位值
R8_TMR2_CTRL_MOD	0x00	模式设置寄存器	0x02
R8_TMR2_CTRL_DMA	0x01	DMA 控制寄存器	0x00
R8_TMR2_INTER_EN	0x02	中断使能寄存器	0x00
R8_TMR2_INT_FLAG	0x06	中断标志寄存器	0x00
R8_TMR2_FIFO_COUNT	0x07	FIFO 计数寄存器	0x00
R32_TMR2_COUNT	0x08	当前计数值寄存器	0x00000000
R32_TMR2_CNT_END	0x0C	计数终值寄存器	0x00000000
R32_TMR2_FIFO	0x10	FIFO 寄存器	0x00000000
R32_TMR2_DMA_NOW	0x14	DMA 当前缓冲区地址	0x0000
R32_TMR2_DMA_BEG	0x18	DMA 起始缓冲区地址	0x0000
R32_TMR2_DMA_END	0x1C	DMA 结束缓冲区地址	0x0000

模式设置寄存器 (R8_TMRx_CTRL_MOD) (x=0/1/2)

位	名称	访问	描述	复位值
[7:6]	RB_TMR_CAP_EDGE	RW	捕获模式下，选择捕捉触发方式。 00：不触发； 01：捕获任何边沿变化之间的时间；	0

			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。	
[7:6]	RB_TMR_PWM_REPEAT	RW	PWM 模式下, 数据重复方式: 00: 重复 1 次; 01: 重复 4 次; 10: 重复 8 次; 11: 重复 16 次。	0
5	Reserved	RO	保留。	0
4	RB_TMR_CAP_COUNT	RW	RB_TMR_MODE_IN=1 下输入模式的子模式。 1: 计数功能; 0: 捕获功能。	0
4	RB_TMR_OUT_POLAR	RW	PWM 模式下, 输出极性设置。 1: 默认高电平, 低电平有效; 0: 默认低电平, 高电平有效。	0
3	RB_TMR_OUT_EN	RW	定时器输出使能位 1: 输出使能; 0: 输出禁止。	0
2	RB_TMR_COUNT_EN	RW	定时器模块使能。 1: 使能; 0: 禁止。	0
1	RB_TMR_ALL_CLEAR	RW	定时器 FIFO/计数器/中断标志清零。 1: 强制清空和清除; 0: 无动作。	1
0	RB_TMR_MODE_IN	RW	定时器模式设置。 1: 输入模式, 捕获/计数功能; 0: 定时模式/PWM 模式	0

中断使能寄存器 (R8_TMRx_INTER_EN) (x=0/1/2)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	0
4	RB_TMR_IE_FIFO_OV	RW	FIFO 溢出 (捕捉模式 FIFO 满或 PWM 模式 FIFO 空) 中断使能。 1: 使能中断; 0: 禁止中断。	0
3	RB_TMR_IE_DMA_END	RW	DMA 完成中断使能 (仅支持 TMR1/TMR2)。 1: 使能中断; 0: 禁止中断。	0
2	RB_TMR_IE_FIFO_HF	RW	FIFO 使用过半 (捕捉模式 FIFO>=4 或 PWM 模式 FIFO<4) 中断使能。 1: 使能中断; 0: 禁止中断。	0
1	RB_TMR_IE_DATA_ACT	RW	捕获模式下, 捕获到新数据中断使能。 PWM 模式下, 有效电平结束中断使能。 1: 使能中断; 0: 禁止中断。	0
0	RB_TMR_IE_CYC_END	RW	捕获模式下, 捕获超时中断使能。 PWM 模式下, PWM 时钟周期结束中断使能。 定时模式下, 电视周期结束中断使能。 1: 使能中断; 0: 禁止中断。	0

中断标志寄存器 (R8_TMRx_INT_FLAG) (x=0/1/2)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	0
4	RB_TMR_IF_FIFO_OV	RW1	FIFO 溢出 (捕捉模式 FIFO 满或 PWM 模式 FIFO 空) 标志。写 1 清零。 1: 标志触发; 0: 未触发。	0
3	RB_TMR_IF_DMA_END	RW1	DMA 完成标志 (仅支持 TMR1/TMR2)。 1: 标志触发; 0: 未触发。	0

2	RB_TMR_IF_FIFO_HF	RW1	FIFO 使用过半捕捉模式 FIFO>=4 或 PWM 模式 FIFO<4) 标志。写 1 清零。 1: 标志触发; 0: 未触发。	0
1	RB_TMR_IF_DATA_ACT	RW1	捕获模式下, 捕获到新数据标志。写 1 清零。 PWM 模式下, 有效电平结束标志。写 1 清零。 1: 标志触发; 0: 未触发。	0
0	RB_TMR_IF_CYC_END	RW1	捕获模式下, 捕获超时标志。写 1 清零。 PWM 模式下, PWM 时钟周期结束标志。写 1 清零。 定时模式下, 电视周期结束标志。写 1 清零。 1: 标志触发; 0: 未触发。	0

FIFO 计数寄存器 (R8_TMRx_FIFO_COUNT) (x=0/1/2)

位	名称	访问	描述	复位值
[7:0]	R8_TMRx_FIFO_COUNT	RO	FIFO 内数据字节计数, 最大值 8。	X

当前计数值寄存器 (R32_TMRx_COUNT) (x=0/1/2)

位	名称	访问	描述	复位值
[31:0]	R32_TMRx_COUNT	RO	计数器当前计数值。	X

计数终值设置寄存器 (R32_TMRx_CNT_END) (x=0/1/2)

位	名称	访问	描述	复位值
[31:0]	R32_TMRx_CNT_END	RW	定时器模式下, 一个定时周期时钟数。 PWM 模式下, PWM 周期总时钟数。 捕获模式下, 捕获超时时钟数。 仅低 26 位有效, 最大值 67108863。 <i>注: R32_TMRx_COUNT 是从 0 起计数, 所以最大值为 R32_TMRx_CNT_END 减 1。</i>	X

FIFO 寄存器 (R32_TMRx_FIFO) (x=0/1/2)

位	名称	访问	描述	复位值
[31:0]	R32_TMRx_FIFO	RO/ WO	FIFO 数据寄存器, 仅低 26 位有效。	X

DMA 控制寄存器 (R8_TMRx_CTRL_DMA) (x=1/2)

位	名称	访问	描述	复位值
[7:3]	Reserved	RO	保留。	0
2	RB_TMR_DMA_LOOP	RW	DMA 地址循环功能使能 (仅支持 TMR1/TMR2)。 1: 地址循环; 0: 地址单次。 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。	0
1	Reserved	RO	保留。	0
0	RB_TMR_DMA_ENABLE	RW	DMA 功能使能 (仅支持 TMR1/TMR2)。 1: 使能 DMA; 0: 禁止 DMA。	0

DMA 当前缓冲区地址 (R32_TMRx_DMA_NOW) (x=1/2)

位	名称	访问	描述	复位值
[31:18]	Reserved	RO	保留。	0
[17:0]	R16_TMRx_DMA_NOW	RW	DMA 数据缓冲区当前地址。 可以用于计算已转换此时, 计算方法为: COUNT=(TMR_DMA_NOW-TMR_DMA_BEG)/4	X

DMA 起始缓冲区地址 (R32_TMRx_DMA_BEG) (x=1/2)

位	名称	访问	描述	复位值
[31:18]	Reserved	RO	保留。	0
[17:0]	R16_TMRx_DMA_BEG	RW	DMA 数据缓冲区起始地址, 地址必须 4 字节对齐。 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。	X

注: 此 DMA 地址可范围 RAMS 和 RAMX 区域。

DMA 结束缓冲区地址 (R32_TMRx_DMA_END) (x=1/2)

位	名称	访问	描述	复位值
[31:18]	Reserved	RO	保留。	0
[17:0]	R16_TMRx_DMA_END	RW	DMA 数据缓冲区结束地址 (不含), 地址必须 4 字节对齐。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。	X

注: 此 DMA 地址可范围 RAMS 和 RAMX 区域。

第 9 章 脉冲宽度调制 (PWMX)

PWMX 模块提供了 4 路 PWM 输出，占空比可调，PWM 周期固定可选 2 种模式，操作简单。扩展的 PWM 管脚输出标识为 PWM0/PWM1/PWM2/PWM3。

9.1 模块配置

1. 设置 PWM0-PWM3 引脚方向为输出。可选地，设置相应 I/O 的驱动能力；
2. 设置寄存器 R8_PWM_CLOCK_DIV，计算 PWM 的时钟基准频率；
3. 设置 PWM 模式控制寄存器 R8_PWM_CTRL_MOD，配置 PWMx 的输出极性，并开启相应的 PWMx (RB_PWMx_OUT_EN 位置 1) 输出；
4. 设置 R8_PWM_CTRL_CFG 寄存器和 R32_PWM_DATA 寄存器，配置输出的 PWM 占空比。

计算公式：PWMx 占空比 = R8_PWMx_DATA / (RB_PWM_CYCLE_SEL ? 255 : 256)

注：如果 R8_PWM_CTRL_MOD 寄存器中，对应的 RB_PWMx_OUT_EN 输出使能一直开启，会不断输出 PWM 波形直到 RB_PWMx_OUT_EN 被禁止。

9.2 寄存器描述

PWMX 相关寄存器物理基地址为：0x40005000

表 9-1 PWM0/1/2/3 相关寄存器列表

名称	偏移地址	描述	复位值
R8_PWM_CTRL_MOD	0x00	PWM 模式控制寄存器	0x00
R8_PWM_CTRL_CFG	0x01	PWM 配置控制寄存器	0x00
R8_PWM_CLOCK_DIV	0x02	PWM 时钟分频寄存器	0x00
R32_PWM_DATA	0x04	PWM0/1/2/3 数据保持寄存器	X
R8_PWM0_DATA	0x04	PWM0 数据保持寄存器	X
R8_PWM1_DATA	0x05	PWM1 数据保持寄存器	X
R8_PWM2_DATA	0x06	PWM2 数据保持寄存器	X
R8_PWM3_DATA	0x07	PWM3 数据保持寄存器	X

PWM 模式控制寄存器 (R8_PWM_CTRL_MOD)

位	名称	访问	描述	复位值
7	RB_PWM3_POLAR	RW	PWM3 输出极性控制。 1: 默认高电平，低有效； 0: 默认低电平，高有效。	0
6	RB_PWM2_POLAR	RW	PWM2 输出极性控制。 1: 默认高电平，低有效； 0: 默认低电平，高有效。	0
5	RB_PWM1_POLAR	RW	PWM1 输出极性控制。 1: 默认高电平，低有效； 0: 默认低电平，高有效。	0
4	RB_PWM0_POLAR	RW	PWM0 输出极性控制。 1: 默认高电平，低有效； 0: 默认低电平，高有效。	0

3	RB_PWM3_OUT_EN	RW	PWM3 输出使能。 1: 使能; 0: 禁止。	0
2	RB_PWM2_OUT_EN	RW	PWM2 输出使能。 1: 使能; 0: 禁止。	0
1	RB_PWM1_OUT_EN	RW	PWM1 输出使能。 1: 使能; 0: 禁止。	0
0	RB_PWM0_OUT_EN	RW	PWM0 输出使能。 1: 使能; 0: 禁止。	0

PWM 配置控制寄存器 (R8_PWM_CTRL_CFG)

位	名称	访问	描述	复位值
[7:1]	Reserved	RW	保留。	0
0	RB_PWM_CYCLE_SEL	RW	PWM 周期选择。 1: 255 个时钟; 0: 256 个时钟。	0

PWM 时钟分频寄存器 (R8_PWM_CLOCK_DIV)

位	名称	访问	描述	复位值
[7:0]	R8_PWM_CLOCK_DIV	RW	PWM 基准时钟分频系数。 计算: $CLK = F_{sys} / R8_PWM_CLOCK_DIV$ 。	0

PWM0/1/2/3 数据保持寄存器 (R32_PWM_DATA)

位	名称	访问	描述	复位值
[31:24]	R8_PWM3_DATA	RW	PWM3 数据保持寄存器。	X
[23:16]	R8_PWM2_DATA	RW	PWM2 数据保持寄存器。	X
[15:8]	R8_PWM1_DATA	RW	PWM1 数据保持寄存器。	X
[7:0]	R8_PWM0_DATA	RW	PWM0 数据保持寄存器。	X

第 10 章 高速并行接口 (HSPI)

并行接口是指采用多根数据线并行传输数据的方式。系统提供了一套高速并行传输接口HSPI (High-Speed Parallel Interface)，通过8根收发控制信号线和32根数据线，实现接口半双工模式的高速传输，速度可达3.8Gbps，可用于大数据的快速转移、自定义高速传输等应用。

10.1 主要特征

- 可配置的8位、16位、32位数据宽度
- 提供硬件自动响应模式
- 提供突发传输模式
- 基于数据包收发结构，便于应用扩展
- 内置多级128-bit FIFO，DMA传输
- 传输可靠：含CRC校验、收发序列号
- 支持双缓冲收发机制
- 传输速度可达3.8Gbps

10.2 功能描述

HSPI定义了一种8/16/32位并行数据高速通讯的传输接口及简单可靠的交互协议。模块通过内置多级收发FIFO、突发模式、DMA机制等方式，实现了接口数据的快速转移。同时，由数据包结构、应答机制、收发序列来保障传输的可靠性，具有简单、高效的特点。

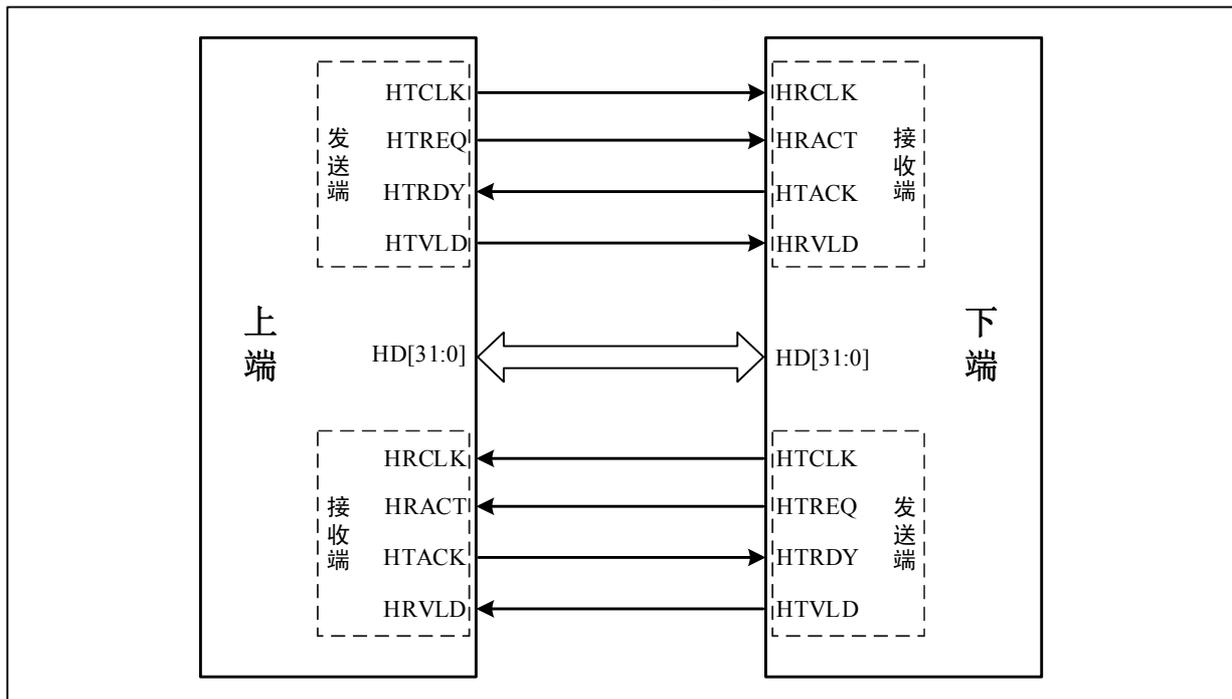
10.2.1 硬件接口描述

HSPI接口硬件包括：8个控制信号线和32个数据信号线，见表10-1描述。其中控制信号分为发送端和接收端，各4根控制线。模块工作时，一方的发送端控制信号要连接另一方接收端控制信号，如图10-1所示。由于只有一组数据线，所以HSPI只能是半双工通讯模式，针对通讯双方同时请求发送的情况，模块会根据当前配置的是上端模式还是下端模式仲裁数据发送优先权。

表10-1 HSPI硬件接口

HSPI接口定义		IO模式配置	功能描述
控制线	发送端	HTCLK	推挽输出 输出通讯时钟信号，时钟频率来自系统主频。
		HTREQ	推挽输出 输出发送请求信号，高电平有效。连接HRACT引脚。
		HTRDY	下拉输入 检测接收端状态。
		HTVLD	推挽输出 输出数据发送状态，高电平有效。连接HRVLD引脚。
	接收端	HRCLK	下拉输入 输入采样时钟，作为接收数据的采样时钟基准。
		HRACT	下拉输入 输入发送请求信号，有效后可驱动HTACK信号。
		HTACK	推挽输出 输出准备接收状态信号，高电平有效。连接HTRDY引脚。
		HRVLD	下拉输入 检测HTVLD引脚状态。
数据线	双向	HDO~HD31	浮空输入 并行数据端口，支持8位、16位、32位数据格式。

图10-1 HSPI接口连接

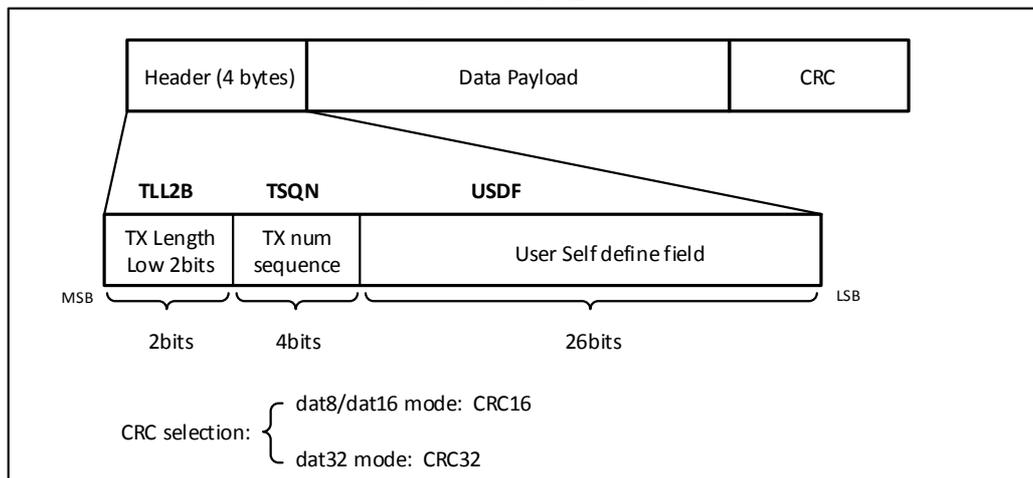


10.2.2 数据包传输格式

HSPI接口传输以数据包为单位，按照低8/16/32位在前的方式传输。一个完整的数据包结构包括：

- 1) 4字节包头，包括2-bit长度零头（TLL2B）、4-bit包序列号（TSQN）、26-bit用户自定义（USDF）
- 2) 可配置长度的有效数据负载
- 3) 校验码，2字节CRC16或4字节CRC32

图10-2 数据包格式



- 包头（Header）：是TLL2B、TSQN、USDF字段组成的一个32位数，在发送时由硬件固定添加。其中TLL2B字段指示当前传输的数据包中，有效数据负载部分在最后一次数据线宽度中占用的有效位宽（传输长度和数据线位宽不对齐），硬件自动处理，应用代码无需关心；TSQN字段指示当前传输数据包的序列号（0-15）。每次成功传输，硬件会自动执行下次序列号的递增，这种方式可以保证数据包收发的连续性，对应寄存器中的RB_HSPI_TX_NUM和RB_HSPI_RX_NUM位；USDF字段是留给用户自定义扩展功能，可以填充上层协议的命令字、地址域、时间戳信息等，对应寄存器

HSPI_UDF0/HSPI_UDF1。

- 有效数据负载：大批量数据传输区域。发送端，应用程序可配置其发送长度（1-4096）和待发送数据的SRAM地址（HSPI_TX_ADDR0/HSPI_TX_ADDR1）；接收端，应用代码可配置待接收数据的SRAM地址（HSPI_RX_ADDR0/HSPI_RX_ADDR1），并从接收长度寄存器中获取接收有效数据负载长度。
注：HSPI 模块 DMA 寻址 RAMX 区域，最低地址 0x20020000，应用程序注意地址分配。
- CRC校验码：对包头+有效数据负载的校验结果，以此保证传输的可靠性。当配置传输数据位宽为8/16位数据模式时，使用CRC16多项式（0x8005）： $X^{16}+X^{15}+X^2+1$ ；当配置传输数据位宽为32位数据模式时，使用CRC32多项式（0x4C11DB7）： $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$ 。

10.2.3 传输配置及过程

下面表述HSPI接口传输过程时，具有优先发送权的一端接口称为“上端”，另外一端接口称为“下端”，双方都可以进行发送和接收，并且两端的发送和接收处理方式一致。接口发送功能由应用代码软件触发，接收功能在模块运行后硬件自动采样接收。

- HSPI接口初始化

HSPI接口内部的复位信号控制（HSPI_CTRL寄存器中）：

- 1) RB_HSPI_ENABLE：模块基准时钟输出（HTCLK）及内部逻辑运行的使能控制位。
- 2) RB_HSPI_TRX_RST：复位HSPI模块数字逻辑电路、状态机，让模块恢复初始工作状态。
- 3) RB_HSPI_ALL_CLR：复位HSPI模块内部FIFO数据和中断标志状态。

注：以上控制位都不会改变HSPI寄存器的配置值。

初始化流程：

- 1) 按照10.2.1描述，设置HSPI接口引脚的IO口模式。
- 2) 设置HSPI_CFG寄存器当前接口的上下端模式和数据宽度。
- 3) 可选择，设置HSPI_CFG寄存器，配置硬件应答功能和双发送或接收缓冲区功能。
- 4) 设置HSPI_AUX寄存器发送/接收采样边沿。
- 5) 可选择，设置HSPI_INT_EN寄存器，让接口在相应条件下可触发HSPI中断。
- 6) 配置HSPI_CTRL寄存器，清除RB_HSPI_ALL_CLR和RB_HSPI_TRX_RST位，RB_HSPI_ENABLE位和RB_HSPI_DMA_EN位置1，打开HSPI功能和内部DMA。

- 单包数据收发

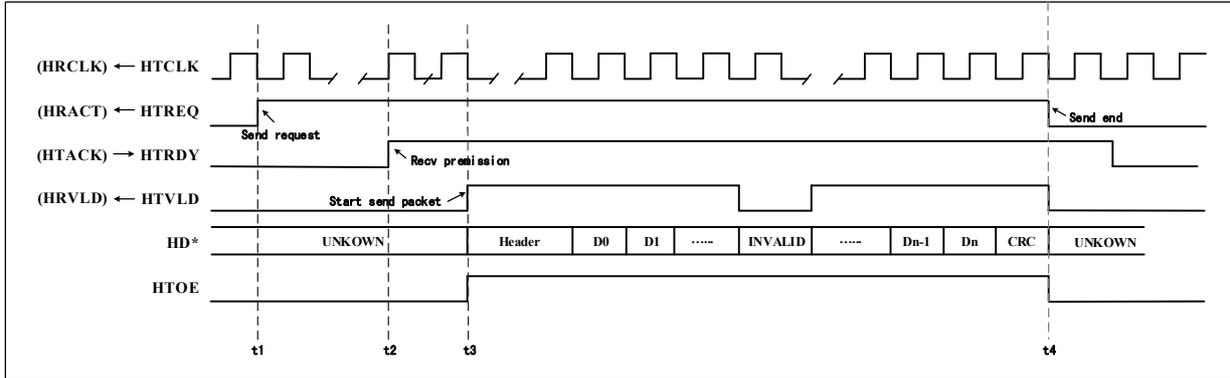
以上端发送，下端接收处理为例。上端发送时，首先硬件驱动发送控制线HTREQ引脚输出高电平，告知下端有数据要传输。此时，如果下端可接收，其硬件将驱动接收控制线HTACK (HTRDY) 输出高电平，否则维持低电平。上端读取到HTRDY上的有效电平后开始准备待发送的数据，内部DMA从SRAM中调度数据到模块内置FIFO中，至少10个时钟周期，具体时间取决于当前系统总线带宽分配。之后置起HTVLD引脚信号并接管并行数据线控制权，根据模块配置的并行位宽大小和时钟发送边沿发送数据包。数据包格式参考10.2.2节描述。

如果发送过程中，由于模块内部调度导致DMA没有及时填充数据到FIFO中，FIFO处于空状态，HTVLD信号将被拉低，此时总线上的数据无效，待FIFO中有数据后，HTVLD会重新被置起，继续数据发送。当所有数据发送完成后，上端硬件驱动HTVLD/HTREQ信号拉低。此时一个完整的数据包发送完成。如

图10-3所示的收发时序。若要启动新的一包数据传输，将重复上述过程。

下端接收时，以HRCLK送入时钟信号作为数据采样时钟基准，软件可设置数据采样边沿。当HRVLD信号有效（高电平）下，接收方在时钟有效沿采样并行数据，并保存到内部FIFO中，由DMA转移到软件配置的接收SRAM区域内，并支持双DMA缓冲区交替保存方式。当HRACT信号无效时，下端认为一个数据包接收完成，校验CRC值、序列号，记录有效数据长度和用户自定义字段到相应寄存器中，并向应用代码上报接收标志状态。

图10-3 数据包收发时序



启动发送前，需要配置好待发送数据包：

- 1) DMA发送地址：设置HSP1_TX_ADDR0寄存器。若使用双缓冲模式，还需设置HSP1_TX_ADDR1寄存器。
- 2) 发送包序号：设置HSP1_TX_SC寄存器。若使用双缓冲模式，还可以设置DMA起始访问地址。
- 3) 用户自定义字段：设置HSP1_UDF0/HSP1_UDF1寄存器，由发送序列号奇偶决定使用哪个寄存器。
- 4) 发送长度：设置HSP1_DMA_LEN0寄存器，若使用双缓冲模式，还需设置HSP1_DMA_LEN1寄存器。

通过RB_HSP1_SW_ACT位置1，软件开启一次包发送。之后需要通过查询或中断的方式检测发送结果，通过访问HSP1_INT_FLAG寄存器的RB_HSP1_IF_T_DONE位判断单包发送是否完成（HTREQ无效）。

准备接收前，需要配置好待接收数据包的存储：

- 1) DMA接收地址，设置HSP1_RX_ADDR0寄存器。若使用双缓冲模式，还需设置HSP1_RX_ADDR1寄存器。
- 2) 接收序列号：存储在HSP1_RX_SC寄存器。若使用双缓冲模式，还可以设置DMA起始接收地址；
- 3) 用户自定义字段：存储在HSP1_UDF0/HSP1_UDF1寄存器中，由接收序列号奇偶决定使用哪个寄存器。

当硬件置位RB_HSP1_IF_R_DONE标志后，表示数据包接收完成，此时读取HSP1_RTX_STATUS寄存器的RB_HSP1_NUM_MIS位和RB_HSP1_CRC_ERR位，判断CRC校验和序列号是否正确，如果正确表示当前数据包有效，通过HSP1_RX_LEN0寄存器获取数据包有效长度，若使用双缓冲模式，还用到HSP1_RX_LEN1寄存器。

● 硬件自动应答

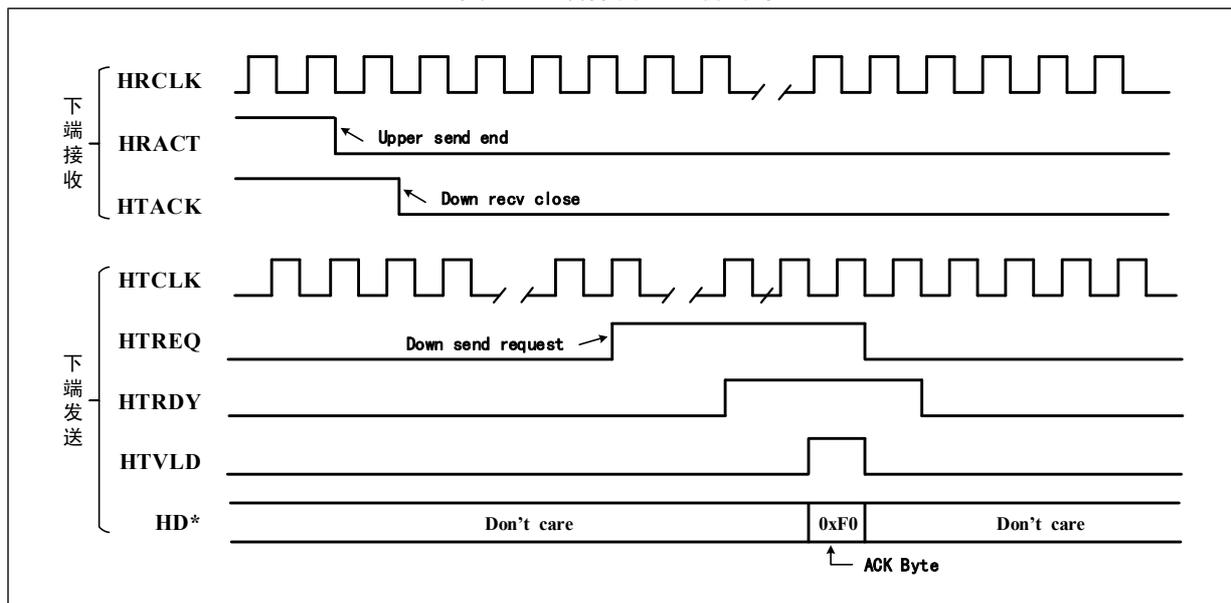
硬件自动应答功能是接收端对发送端所发送数据包的正确性、连续性的一种回应。通过设置HSP1_CFG寄存器的HSP1_HW_ACK位置1，启用此功能。

此模式下，接收端在收到一个完整的数据包时（HTACK引脚状态无效），如果判断其CRC校验正确、接收序列号匹配，硬件会自动发送一个特殊应答字节（0xF0）表示当前数据包已被成功接收。否则不发送任何应答，此时发送端将查询到应答超时状态。

上述发送应答过程无需应用程序的介入，发送端发送完成后自动监测应答符，接收端收到一个包

后自动发送应答字节。

图10-4 硬件自动应答时序



● DMA双缓冲

模块对收发地址和长度各提供了2组寄存器，主要用于DMA的双缓冲模式。通过设置HSPI_CFG寄存器的RB_HSPI_RX_TOG_EN和RB_HSPI_TX_TOG_EN位，可分别开启接收和发送的DMA双缓冲功能。此时每成功发送或接收一个数据包后，模块的地址和长度寄存器访问会转向另一组寄存器。RB_HSPI_TX_TOG和RB_HSPI_RX_TOG位分别指示当前使用的发送和接收寄存器标志。

通过DMA双缓冲模式，应用代码可以在硬件对一组缓冲区进行收发操作时，处理另一组缓冲区的数据，实现硬件收发和软件处理同步进行，提高数据传输效率。

注：如果没有使用双缓冲模式，但开启了突发发送模式，那么要及时更新发生DMA地址和长度，否则会导致数据包重复发送。图10-3的时序图中， $t1-t3$ 时间内不能修改地址和长度，此时模块内部还在使用当前寄存器， $t3-t4$ 时间内可修改切换。

● 发送优先权

HSPI接口既可以接收也可以发送数据，但因为共用数据线，所以某一时刻只能发送或接收（半双工传输模式）。根据“单包数据收发时序”，发送请求由HTREQ有效电平触发，所以当HSPI接口两端优先驱动HTREQ引脚高电平的一方获取发送权限。如果双方同时驱动HTREQ引脚，那么配置为“上端”模式的一方优先获取发送权限。通过HSPI_CFG寄存器的RB_HSPI_MODE位配置HSPI接口当前为“上端”或“下端”模式。

● 突发模式发送

由于每个数据包的发送都需要软件参与提供触发，所以为了提高发送效率，HSPI接口提供了“突发模式发送”，硬件自动发送指定数量的数据包，不需要人为干预。应用代码要配置数据包总个数（RB_HSPI_BURST_LEN），最大支持256个连续数据包。通过设置HSPI_BURST_CFG寄存器的RB_HSPI_BURST_EN位置1启用此模式。突发模式下，当获得软件触发信号后（RB_HSPI_SW_ACT位置1），硬件将根据地址和长度寄存器发送第一个数据包，满足“单包数据收发时序”。单包发送完成后，将立即启动下一个数据包的发送，无需等待软件触发信号，直到指定数量的数据包都发送完成，应用代

码可以读取HSP1_BURST_CNT寄存器获取当前已成功发送数据包个数，全部传输完成，硬件会自动清0。突发模式下，发送完成或者等待应答超时，硬件将置位RB_HSP1_IF_B_DONE标志，同时关闭突发模式功能（RB_HSP1_BURST_EN=0）。如果要继续使用突发模式需要软件重新开启。突发模式发送下，每个独立数据包发送完成硬件都会置位中断标志RB_HSP1_IF_T_DONE，但是只有发送成功，发送/接收序列号才会递增。

如果开启了硬件自动应答功能，发送端在每个单包发送完成并接受到应答字节后，才会启动下一个数据包发送。如果应答超时或应答错误将中止当前的突发传输。

如果开启了DMA双缓冲功能，每成功发送或接收完成一个数据包，将转换发送/接收地址和长度寄存器。

注1：多个连续数据包发送中，TSQN字段的奇偶状态决定了USDF字段对应的寄存器，与是否开启双缓冲无关。

注2：如果在突发模式的数据包发送间隔中，发生了“发送优先权”的选择，突发模式发送将被中止，先进行优先权发送处理，之后会继续发送。

注3：使用突发模式时，必须使能硬件自动应答模式，以确认多个连续包被成功发送。

10.4 寄存器描述

HSP1 外设相关寄存器基地址：0x40006000

表10-1 HSP1相关寄存器列表

名称	偏移地址	描述	复位值
R8_HSP1_CFG	0x00	HSP1 配置寄存器	0x82
R8_HSP1_CTRL	0x01	HSP1 控制寄存器	0x18
R8_HSP1_INT_EN	0x02	HSP1 中断使能寄存器	0x00
R8_HSP1_AUX	0x03	HSP1 辅助寄存器	0x00
R32_HSP1_TX_ADDRO	0x04	HSP1 发送地址寄存器 0	0x0000000
R32_HSP1_TX_ADDR1	0x08	HSP1 发送地址寄存器 1	0x0000000
R32_HSP1_RX_ADDRO	0x0C	HSP1 接收地址寄存器 0	0x0000000
R32_HSP1_RX_ADDR1	0x10	HSP1 接收地址寄存器 1	0x0000000
R16_HSP1_DMA_LEN0	0x14	HSP1 发送长度寄存器 0	0x0000
R16_HSP1_RX_LEN0	0x16	HSP1 接收长度寄存器 0	0x0000
R16_HSP1_DMA_LEN1	0x18	HSP1 发送长度寄存器 1	0x0000
R16_HSP1_RX_LEN1	0x1A	HSP1 接收长度寄存器 1	0x0000
R16_HSP1_BURST_CFG	0x1C	HSP1 突发模式配置寄存器	0x0000
R8_HSP1_BURST_CNT	0x1E	HSP1 突发模式计数器	0x00
R32_HSP1_UDF0	0x20	HSP1 用户自定义字段 0	0x0000000
R32_HSP1_UDF1	0x24	HSP1 用户自定义字段 1	0x0000000
R8_HSP1_INT_FLAG	0x28	HSP1 中断标志寄存器	0x00
R8_HSP1_RTX_STATUS	0x29	HSP1 接收和发送状态寄存器	0x00
R8_HSP1_TX_SC	0x2A	HSP1 发送序列控制寄存器	0x00
R8_HSP1_RX_SC	0x2B	HSP1 接收序列控制寄存器	0x00

HSP1 配置寄存器（R8_HSP1_CFG）

位	名称	访问	描述	复位值
7	RB_HSP1_HW_ACK	RW	应答模式配置。 1: 硬件自动应答模式;	1

			0: 无应答模式。	
6	RB_HSPI_RX_TOG_EN	RW	双缓冲接收使能。 1: 缓冲区由地址 0 和地址 1 交替接收; 0: 缓冲区由地址 0 接收。	0
5	RB_HSPI_TX_TOG_EN	RW	双缓冲发送使能。 1: 缓冲区由地址 0 和地址 1 交替发送; 0: 缓冲区由地址 0 发送。	0
4	Reserved	RO	保留。	0
[3:2]	RB_HSPI_MSK_SIZE	RW	并行数据大小。 00: 8 位模式; 01: 16 位模式; 1x: 32 位模式。	0
1	RB_HSPI_DUDMA	RW	模块内部的双 DMA 模式使能。开启此功能可以使 HSPI 接口的系统 DMA 带宽提高 1 倍。 1: 使能双 DMA 请求; 0: 禁止双 DMA 请求。	1
0	RB_HSPI_MODE	RW	发送优先级配置。当 HSPI 接口两端同时发出请求时, 配置上端模式优先执行 1: 上端模式 (优先); 0: 下端模式。	0

HSPI 控制寄存器 (R8_HSPI_CTRL)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	0
4	RB_HSPI_TRX_RST	RW	HSPI 模块收发逻辑复位控制。 1: 执行内部逻辑单元复位; 0: 无动作。	1
3	RB_HSPI_ALL_CLR	RW	HSPI 模块内置 FIFO/中断标志复位控制。 1: 执行 FIFO 和中断标志复位; 0: 无动作。	1
2	RB_HSPI_SW_ACT	RW	软件触发数据包发送。软件置 1, 硬件清除。 1: 触发一次数据包发送; 0: 无动作。	0
1	RB_HSPI_DMA_EN	RW	DMA 使能。 1: 使能 DMA 功能; 0: 禁止 DMA 功能。	0
0	RB_HSPI_ENABLE	RW	HSPI 模块功能使能。 1: 使能 HSPI; 0: 禁止 HSPI。	0

HSPI 中断使能寄存器 (R8_HSPI_INT_EN)

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0
3	RB_HSPI_IE_B_DONE	RW	突发模式下, 突发序列发送完成中断使能。 1: 使能中断; 0: 禁止中断。	0
2	RB_HSPI_IE_FIFO_OV	RW	收发 FIFO 溢出中断使能。 1: 使能中断; 0: 禁止中断。	0
1	RB_HSPI_IE_R_DONE	RW	单包接收完成中断使能。 1: 使能中断; 0: 禁止中断。	0
0	RB_HSPI_IE_T_DONE	RW	单包发送完成中断使能。	0

			1: 使能中断;	0: 禁止中断。	
--	--	--	----------	----------	--

HSP1 辅助寄存器 (R8_HSPI_AUX)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	0
[4:3]	RB_HSPI_ACK_CNT_SEL	RW	延迟发送 ACK 模式下, 延迟时间控制: 00: 延迟 $2T_{sys}$ 发送 ACK; 01: 延迟 $4T_{sys}$ 发送 ACK; 10: 延迟 $6T_{sys}$ 发送 ACK; 11: 延迟 $8T_{sys}$ 发送 ACK。	0
2	RB_HSPI_ACK_TX_MOD	RW	硬件自动 ACK 发送时间配置: 1: 当 HRVLD 信号变无效时, 延迟发送 ACK; 0: 当所有数据写入 SRAM 时, 立即发送 ACK。	0
1	RB_HSPI_RCK_MOD	RW	接收数据采样边沿配置。 1: 时钟下降沿采样数据; 0: 时钟上升沿采样数据。	0
0	RB_HSPI_TCK_MOD	RW	发送数据时钟极性配置。 1: 时钟下降沿发送数据; 0: 时钟上升沿发送数据。	0

HSP1 发送地址 0 寄存器 (R32_HSPI_TX_ADDR0)

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
[16:0]	RB_HSPI_TX_ADDR0	RW	配置 DMA 发送地址 0。低 4 位固定为 0 (16 字节对齐)。	0

注: DMA 寻址 RAMX 区域。

HSP1 发送地址 1 寄存器 (R32_HSPI_TX_ADDR1)

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
[16:0]	RB_HSPI_TX_ADDR1	RW	配置 DMA 发送地址 1。低 4 位固定为 0 (16 字节对齐)。 注: 双缓冲模式下使用。	0

注: DMA 寻址 RAMX 区域。

HSP1 接收地址 0 寄存器 (R32_HSPI_RX_ADDR0)

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
[16:0]	RB_HSPI_RX_ADDR0	RW	配置 DMA 接收地址 0。低 4 位固定为 0 (16 字节对齐)。	0

注: DMA 寻址 RAMX 区域。

HSP1 接收地址 1 寄存器 (R32_HSPI_RX_ADDR1)

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
[16:0]	RB_HSPI_RX_ADDR1	RW	配置 DMA 接收地址 1。低 4 位固定为 0 (16 字节)	0

			对齐)。 注：双缓冲模式下使用。	
--	--	--	---------------------	--

注：DMA 寻址 RAMX 区域。

HSPI DMA 发送长度 0 寄存器 (R16_HSPI_DMA_LEN0)

位	名称	访问	描述	复位值
[15:12]	Reserved	RO	保留。	0
[11:0]	RB_HSPI_DMA_LEN0	RW	配置 DMA 发送长度 0。实际发送长度 (LEN0+1)。	0

HSPI 接收长度 0 寄存器 (R16_HSPI_RX_LEN0)

位	名称	访问	描述	复位值
[15:12]	Reserved	RO	保留。	0
[11:0]	RB_HSPI_RX_LEN0	RW	配置 DMA 接收长度 0。数据 0 表示 4096 字节。	0

HSPI DMA 发送长度 1 寄存器 (R16_HSPI_DMA_LEN1)

位	名称	访问	描述	复位值
[15:12]	Reserved	RO	保留。	0
[11:0]	RB_HSPI_DMA_LEN1	RW	配置 DMA 发送长度 1。实际发送长度 (LEN0+1)。 注：双缓冲模式下使用。	0

HSPI 接收长度 1 寄存器 (R16_HSPI_RX_LEN1)

位	名称	访问	描述	复位值
[15:12]	Reserved	RO	保留。	0
[11:0]	RB_HSPI_RX_LEN1	RW	配置 DMA 接收长度 1。数据 0 表示 4096 字节。 注：双缓冲模式下使用。	0

HSPI 突发配置寄存器 (R16_HSPI_BURST_CFG)

位	名称	访问	描述	复位值
[15:8]	RB_HSPI_BURST_LEN	RW	配置突发传输包个数。	0
[7:1]	Reserved	RO	保留。	0
0	RB_HSPI_BURST_EN	RW	突发传输使能。软件置 1，突发传输完成后错误时硬件自动清除此位。 1：使能突发传输； 0：禁止突发模式。	0

HSPI 突发计数器寄存器 (R8_HSPI_BURST_CNT)

位	名称	访问	描述	复位值
[7:0]	RB_HSPI_BURST_CNT	RO	突发传输计数器。记录当前突发模式下已成功发送包数。使能突发功能的操作将自动复位计数器，达到突发设置包数硬件自动复位计数器。	0

HSPI 用户自定义字段 0 寄存器 (R32_HSPI_UDF0)

位	名称	访问	描述	复位值
[31:26]	Reserved	RO	保留。	0

[25:0]	RB_HSPI_UDFO	RW	发送时，由软件设置要发送的自定义内容； 接收时，存储接收到数据包中的自定义内容。 注：接收/发送序列控制寄存器中的序列号为偶数选择此寄存器。	0
--------	--------------	----	--	---

HSPI 用户自定义字段 1 寄存器 (R32_HSPI_UDF1)

位	名称	访问	描述	复位值
[31:26]	Reserved	RO	保留。	0
[25:0]	RB_HSPI_UDF1	RW	发送时，由软件设置要发送的自定义内容； 接收时，存储接收到数据包中的自定义内容。 注：接收/发送序列控制寄存器中的序列号为奇数选择此寄存器。	0

HSPI 中断标志寄存器 (R8_HSPI_INT_FLAG)

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0
3	RB_HSPI_IF_B_DONE	RW1	突发模式下，突发序列包发送完成标志。 1: 突发发送完成； 0: 无事件。	0
2	RB_HSPI_IF_FIFO_OV	RW1	收发 FIFO 溢出标志。 1: 接收或发送 FIFO 溢出； 0: 无事件。	0
1	RB_HSPI_IF_R_DONE	RW1	单包接收完成标志。 1: 数据接收完成； 0: 无事件。 注：接收完成后需要读取状态寄存器判断接收数据 CRC 是否正确。	0
0	RB_HSPI_IF_T_DONE	RW1	单包发送完成标志。 1: 数据发送完成； 0: 无事件。 注：硬件自动应答模式下需要读取状态寄存器判断发送状态是否成功。	0

注：即使不配置 R8_HSPI_INT_EN 寄存器，各标志仍然在满足条件下置位。

HSPI 收发状态寄存器 (R8_HSPI_RTX_STATUS)

位	名称	访问	描述	复位值
[7:3]	Reserved	RO	保留。	0
2	RB_HSPI_NUM_MIS	RO	接收序列号状态。 1: 不匹配； 0: 匹配。	0
1	RB_HSPI_CRC_ERR	RO	接收 CRC 校验状态。 1: CRC 校验错误； 0: CRC 校验正确。	0
0	Reserved	RO	保留。	0

HSPI 发送序列控制寄存器 (R8_HSPI_TX_SC)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	0
4	RB_HSPI_TX_TOG	RWT	发送标志。 1: 使用 DMA 发送地址 1，长度寄存器 1； 0: 使用 DMA 发送地址 0，长度寄存器 0。 双缓冲模式下，发送完成并成功时，硬件自动翻	0

			转此位。软件写 1 执行翻转，写 0 无效。 注：双缓冲模式下作用。	
[3:0]	RB_HSPI_TX_NUM	RW	发送序列号。 发送完成并成功，序列号自动递增，软件可设置。 注：此域决定发送包的 TSQN 字段。	0

注：在硬件自动应答模式下，只有正确接收到应答包时，会翻转发送标志，递增发送序列号；非硬件自动应答模式下，发送完成时（不判断应答），即翻转发送标志，递增发送序列。

HSPI 接收序列控制寄存器 (R8_HSPI_RX_SC)

位	名称	访问	描述	复位值
[7:5]	Reserved	R0	保留。	0
4	RB_HSPI_RX_TOG	RWT	DMA 接收地址标志。 1：当前接收使用 DMA 地址 1； 0：当前接收使用 DMA 地址 0。 接收完成时，如果 CRC 校验正确并且收发序列匹配，硬件自动翻转此位。 软件写 1 执行翻转，写 0 无效。 注：双缓冲模式下作用。	0
[3:0]	RB_HSPI_RX_NUM	RW	接收序列号。 接收完成时，如果 CRC 校验正确并且收发序列匹配，则接收序列号递增，软件可设置。 注：此域用来对比接收包的 TSQN 字段。	0

第 11 章 外部总线接口（BUS8）

系统带有一个外部总线控制器接口，支持32KB的寻址空间，地址范围为0x80000000至80007FFF。包括读信号引脚（PRD#，即PA8）、写信号引脚（PWR#，即PA9）、15位地址引脚（PB0–PB14）和8位数据引脚（PA0–PA7）。当未使用外部总线接口功能时，上述引脚可以作为通用I/O口使用。该接口支持静态存储器映射器件，包括RAM、ROM、Flash和一些外部I/O器件，可动态修改读写信号脉宽以及地址和数据的建立和保持时间。

11.1 主要特征

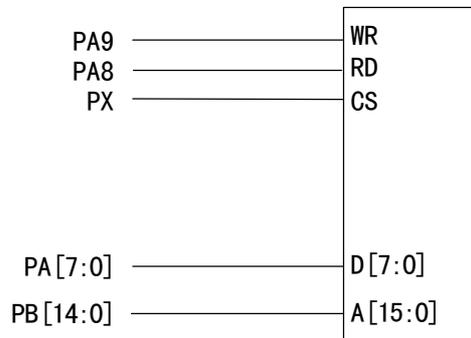
- 支持8位总线宽度
- 支持最多15位地址总线宽度
- 支持静态存储器映射器件，包括RAM、Flash和一些外部I/O器件
- 支持动态修改读写信号脉宽、地址和数据的建立和保持时间等

11.2 外部总线接口应用

11.2.1 外部总线接口

外部总线接口包括1根读信号引脚PRD#(PA8)、1根写信号引脚PWR#(PA9)、15根地址引脚A0–A14(PB0–PB14)及8根数据引脚D0–D7(PA0–PA7)。图11-1展示了该接口连接外部外部器件方式。

图11-1 BUS8接口连接外部芯片



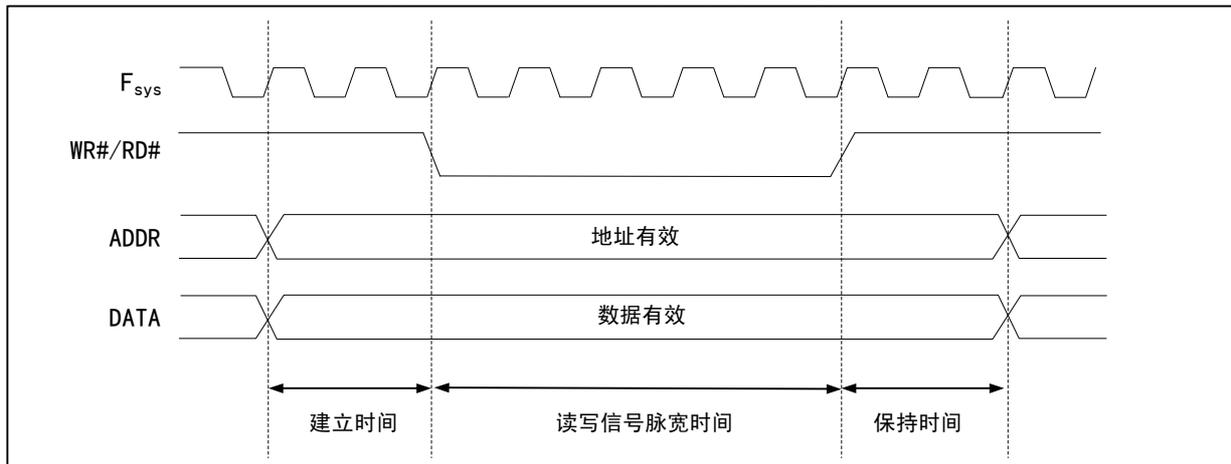
11.2.2 总线时序

BUS8总线工作时，其通讯时序中的数据建立、读写有效、数据保持几个时间参数可调。通过配置寄存器RB_XBUS_CONFIG中的bit[7:4]配置位。RB_XBUS_SETUP位是数据的建立时间，当设置为0时，外部总线建立时间为2个时钟周期(T_{sys})；当设置为1时，外部总线建立时间为3个时钟周期(T_{sys})；RB_XBUS_HOLD位是数据的保持时间，当设置为0时，数保持时间为2个时钟周期(T_{sys})；当设置为1时，数据保持时间为3个时钟周期(T_{sys})；RB_XBUS_WIDTH是总线数据读写有效窗口宽度，可选3个时钟、5个时钟、9个时钟或者16个时钟。

可选配置地址总线宽度，6位/10位/15位寻址范围。不用的地址线可以做其他功能使用。

图11-2举例了RB_XBUS_WIDTH=01b，RB_XBUS_HOLD=0b，RB_XBUS_SETUP=0b条件下的通讯时序。其中 F_{sys} 为总线时钟频率，RD#为读控制信号，WR#为写控制信号。

图11-2 BUS8总线接口读/写访问时序图



11.2.3 外部总线配置

1. 配置外部总线配置寄存器 (R8_XBUS_CONFIG)，设置RB_XBUS_ENABLE位为1使能外部总线接口，同时根据需要设置地址位控制域RB_XBUS_ADDR_OE，选择需要的寻址范围；
2. 配置外部总线配置寄存器 (R8_XBUS_CONFIG)，设置RB_XBUS_WIDTH位控制域，选择总线读写时钟脉宽；设置RB_XBUS_SETUP位和RB_XBUS_HOLD位，选择总线建立时间以及总线保持时间；
3. 配置外部总线接口相应引脚方向，参考5.3.1说明。
4. 之后即可驱动指向0x80000000至80007FFF地址范围的指针进行数据读写。

11.3 寄存器描述

BUS8 外设相关寄存器基地址：0x40001000

表11-1 BUS8相关寄存器列表

名称	偏移地址	描述	复位值
R8_XBUS_CONFIG	0x10	外部总线配置寄存器	0x00

外部总线配置寄存器 (R8_XBUS_CONFIG)

位	名称	访问	描述	复位值
7	RB_XBUS_SETUP	RW	外部总线数据建立时间： 1:3 个时钟周期； 0:2 个时钟周期。	0
6	RB_XBUS_HOLD	RW	外部总线数据保持时间： 1:3 个时钟周期； 0:2 个时钟周期。	0
[5:4]	RB_XBUS_WIDTH	RW	外部总线 RD/WR 有效时间： 00: 3 个时钟周期； 01: 5 个时钟周期； 10: 9 个时钟周期； 11: 16 个时钟周期。	0
[3:2]	RB_XBUS_ADDR_OE	RW	外部总线地址输出使能： 00: 无总线地址输出； 01: 地址线 PA[5:0]； 10: 地址线 PA[9:0]；	0

			11: 地址线 PA[14:0]。	
1	Resered	RO	保留。	0
0	RB_XBUS_ENABLE	RW	外部总线使能位: 1: 使能; 0: 禁止。	0

第 12 章 数字视频接口 (DVP)

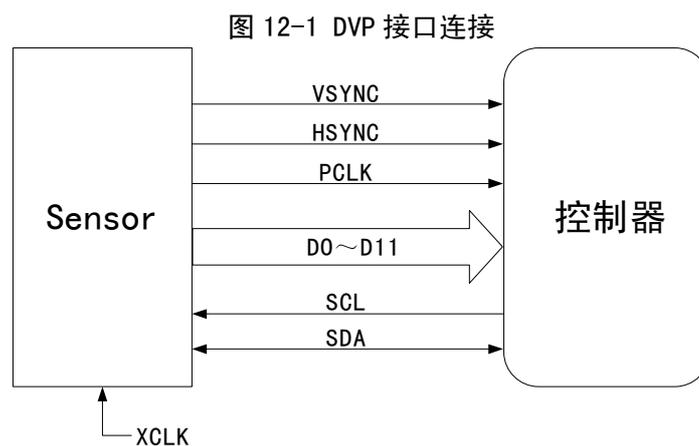
系统提供了一个数字图像接口 DVP (Digital Video Port)，它支持使用 DVP 接口时序来获取图像数据流，支持按原始的行、帧格式组织的图像数据，如 YUV、RGB 等，也支持如 JPEG 格式的压缩图像数据流。接收时，主要依靠 VSYNC 和 HSYNC 信号同步。

12.1 主要特征

- 可配置8/10/12位数据宽度模式
- 支持YUV、RGB格式数据
- 支持JPEG压缩格式数据
- 内置FIFO，支持DMA传输
- 支持双缓冲接收

12.2 功能描述

12.2.1 与传感器相连



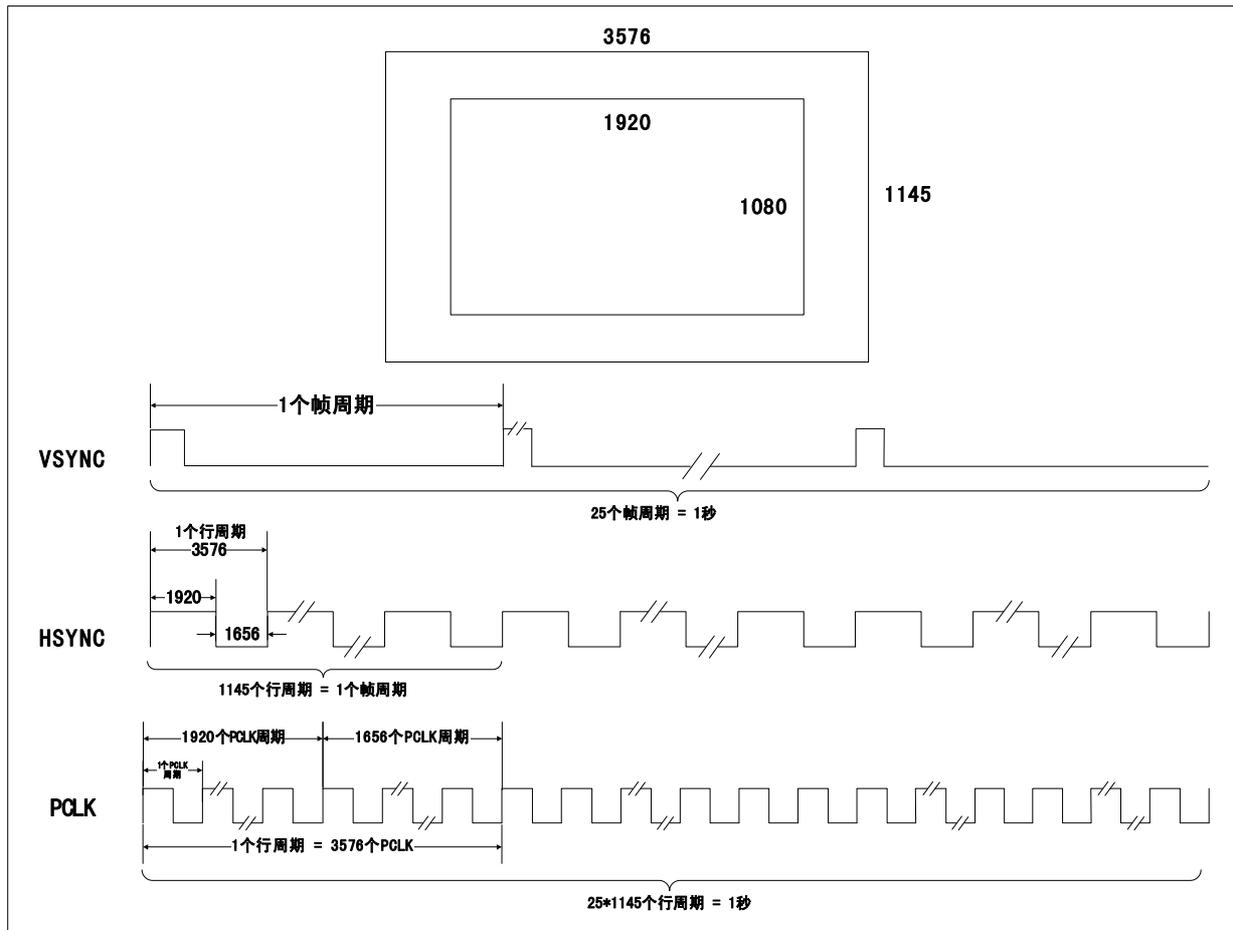
- PCLK (Pixel clk): 像素时钟，每个时钟对应一个像素数据（非压缩数据）。外部 DVP 接口传感器输出的 PCLK 时钟一般不超过 96MHz。
- HSYNC (Horizontal Synchronization): 行同步信号。
- VSYNC (Vertical Synchronization): 帧同步信号。
- DATA: 像素数据或压缩数据，位宽一般为 8/10/12 位。
- XCLK: Sensor 的参考时钟，可以由微控制器提供或外部提供，一般使用晶振。
- I2C 接口: 用来读写 Sensor 的内部寄存器，配置其参数。可通过控制的普通 GPIO 口模拟 I2C 时序通讯。

12.2.2 DVP 接口时序描述

一般使用的 DVP 接口传感器模块输出的数字信号数据流和图像大小存在一定的关系，下面以一个图像数据举例。

如图 12-2 所示，传感器内部是一个 3576*1145 大小完整的图像尺寸，经过内部缩放处理，最终从接口输出的图像数据大小位 1920*1080，图像刷新率为 25。

图 2-2 时序描述



PCLK 是一个像素传输的时间，所以 HSYNC 是 PCLK 的 3576 倍，在 3576 个像素中，只有 1920 个像素是有效的，剩下的 1656 个像素点时间内 Sensor 不传输数据。VSYNC 是帧同步型号，所以 VSYNC 时间是 PCLK 的 3576×1145 倍，同样，只有在 1920×1080 个有效像素时间内，Sensor 在传输数据。如果传感器传输的是 JPEG 压缩后的数据，可能无需使用 HSYNC 信号。

DVP 接口信号和图像数据的关系具体以选用的传感器数据手册说明为主。

12.2.3 RGB/YUV/JPEG 压缩数据格式说明

- RGB

三原色：红色、绿色、蓝色

- YUV

亮度信号Y，色度信号U和V、Pb和Pr、Cb和Cr。

- JPEG (Joint Photographic Experts Group)

有损失压缩，但损失的部分是人视觉不易察觉的部分，利用人眼对计算机色彩中高频信息部分不敏感的特点。去除视觉上多余信息（空间冗余度），去除数据本身的多余信息（结构冗余度）。

12.3 数字视频接口的应用

12.3.1 数字图像接口配置说明

- (1) DVP 的数据存储使用 RAMX 区域，最大可用空间为 96KB（取决于系统配置）。
- (2) DVP 的数据接收中，每一帧数据由 BUF0 和 BUF1 交替存储，从 BUF0 开始。对于 RGB 和 YUV 数据流，硬件在每次帧信号上升沿时会复位选择 BUF0 开始，存满一行数据将会切换的 BUF1，实现交替存储；对于 JPEG 压缩数据，硬件会根据设置的 DMA 接收长度设置 BUF0 和 BUF1 的切换阈值。
- (3) 当数据总线宽度为 10 或者 12 时，每接收一个数据时，系统将自动对数据进行无符号扩展到 16 位再进行存储。
- (4) R16_DVP_ROW_NUM 和 R16_DVP_COL_NUM 寄存器必须与传感器实际输出的图像大小匹配。
- (5) 在视频流 RGB 模式下，R16_DVP_COL_NUM 表示一行数据的有效 PCLK 周期数；在图像 JPEG 模式下，R16_DVP_COL_NUM 用于配置 DMA 长度，此模式下，R16_DVP_ROW_NUM 寄存器不起作用。

12.3.2 数字图像接口应用说明

在使用数字图像接口接收图像数据时，须正确配置 DVP 相关的控制寄存器，使其与图像传感器的模式相匹配，具体操作步骤如下：

- a) 通过 R8_DVP_CR1 寄存器清除 RB_DVP_ALL_CLR 字段，并解除接收复位。
- b) 通过 R8_DVP_CR0 寄存器配置接收数据有效位宽。
- c) 根据配置后的图像传感器输出的有效图像像素，配置 R16_DVP_ROW_NUM 和 R16_DVP_COL_NUM 寄存器，使其与 SENSOR 的输出相匹配。
- d) 通过 R32_DVP_DMA_BUF0/1 寄存器配置 DMA 接收地址。
- e) 通过 R8_DVP_INT_EN 寄存器使能行结束中断和帧结束中断。
- f) 通过 R8_DVP_CR1 寄存器使能 DMA，通过 R8_DVP_CR0 寄存器使能 DVP 接口。
- g) 等待相关接收中断的产生，及时处理接收数据。

12.4 寄存器描述

DVP 外设相关寄存器基地址：0x4000E000

表 12-1 数字图像接口相关寄存器列表

名称	偏移地址	描述	复位值
R8_DVP_CR0	0x00	DVP 控制寄存器 0	0x00
R8_DVP_CR1	0x01	DVP 控制寄存器 1	0x06
R8_DVP_INT_EN	0x02	DVP 中断使能寄存器	0x00
R16_DVP_ROW_NUM	0x04	图像行数配置寄存器	0x0000
R16_DVP_COL_NUM	0x06	图像列数配置寄存器	0x0000
R32_DVP_DMA_BUF0	0x08	DVP DMA 地址 0 寄存器	0x00000000
R32_DVP_DMA_BUF1	0x0C	DVP DMA 地址 1 寄存器	0x00000000
R8_DVP_INT_FLAG	0x10	DVP 中断标志寄存器	0x00
R8_DVP_FIFO_ST	0x11	DVP 接收 FIFO 状态寄存器	0x00
R16_DVP_ROW_CNT	0x14	DVP 行计数器	0x0000
R16_DVP_COL_CNT	0x16	DVP 列计数器	0x0000

DVP 配置寄存器 (R8_DVP_CR0)

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

7	RB_DVP_ROW_CM	RW	DVP 行计数模式控制。 1: 在列计数结束时计数; 0: 在 HSYNC 下降沿时计数。	0
6	RB_DVP_JPEG	RW	JPEG 模式使能。 1: JPEG 压缩格式; 0: 原始数据格式。	0
[5:4]	RB_DVP_MSK_DAT_MOD	RW	DVP 数据位宽配置。 00: 8 位模式; 01: 10 位模式; 1x: 12 位模式。	0
3	RB_DVP_P_POLAR	RW	PCLK 极性配置。 1: 在 PCLK 下降沿采样数据; 0: 在 PCLK 上升沿采样数据。	0
2	RB_DVP_H_POLAR	RW	HSYNC 极性配置。 1: HSYNC 低电平数据有效; 0: HSYNC 高电平数据有效。	0
1	RB_DVP_V_POLAR	RW	VSYNC 极性配置。 1: VSYNC 高电平数据有效; 0: VSYNC 低电平数据有效。	0
0	RB_DVP_ENABLE	RW	DVP 功能使能。 1: 使能 DVP; 0: 禁用 DVP。	0

DVP 配置寄存器 (R8_DVP_CR1)

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	-
3	RB_DVP_BUF_TOG	RWT	缓冲地址标志位。硬件控制翻转, 软件置 1 翻转该位, 写 0 无效。 1: 数据存储在接收地址 1; 0: 数据存储在接收地址 0。	0
2	RB_DVP_RCV_CLR	RW	接收逻辑复位控制。 1: 复位接收逻辑电路; 0: 取消复位操作。	1
1	RB_DVP_ALL_CLR	RW	标志与 FIFO 清除控制, 由软件写 1 或写 0: 1: 复位标志与 FIFO; 0: 取消复位操作。	1
0	RB_DVP_DMA_ENABLE	RW	DMA 使能控制位: 1: 使能 DMA; 0: 禁用 DMA。	0

DVP 中断使能寄存器 (R8_DVP_INT_EN)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	-
4	RB_DVP_IE_STP_FRM	RW	帧结束中断使能。在 VSYNC 上升沿时产生中断。 1: 使能帧结束中断; 0: 禁止帧结束中断。	0
3	RB_DVP_IE_FIFO_OV	RW	接收 FIFO 溢出中断使能。 1: 使能 FIFO 溢出中断; 0: 禁止 FIFO 溢出中断。	0
2	RB_DVP_IE_FRM_DONE	RW	帧接收完成中断使能。(计数器达到	0

			RAW/COL_NUM 配置值) 1: 使能帧接收完成中断; 0: 禁止帧接收完成中断。	
1	RB_DVP_IE_ROW_DONE	RW	行结束中断使能。(计数器达到 COL_NUM 配置值或 HSYNC 下降沿) 1: 使能行结束中断; 0: 禁止行结束中断。	0
0	RB_DVP_IE_STR_FRM	RW	新一帧开始中断使能。在 VSYNC 的下降沿时产生中断, 表示新的一帧开始, 数据即将到来。 1: 使能新一帧开始中断; 0: 禁止新一帧开始中断。	0

DVP 图像有效行数配置寄存器 (R16_DVP_ROW_NUM)

位	名称	访问	描述	复位值
[15:0]	RB_DVP_ROW_NUM	RW	在 RGB 模式下, 表示一帧图像数据内包含的行数。在 JPEG 模式下, 该寄存器无实际意义。	0

DVP 图像有效列数配置寄存器 (R16_DVP_COL_NUM)

位	名称	访问	描述	复位值
[15:0]	RB_DVP_COL_NUM	RW	在 RGB 模式下, 表示一行数据内包含的 PCLK 周期数。在 JPEG 模式下, 用于配置 DMA 接收长度。	0

DVP DMA 接收地址 0 寄存器 (R32_DVP_DMA_BUF0)

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	-
[16:0]	RB_DVP_DMA_BUF0	RW	DMA 接收地址 0。低 4 位固定为 0(16 字节对齐)。	0

注: DMA 寻址 RAMX 区域。

DVP DMA 接收地址 1 寄存器 (R32_DVP_DMA_BUF1)

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	-
[16:0]	RB_DVP_DMA_BUF1	RW	DMA 接收地址 1。低 4 位固定为 0(16 字节对齐)。	0

注: DMA 寻址 RAMX 区域。

DVP 中断标志寄存器 (R8_DVP_INT_FLAG)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	-
4	RB_DVP_IF_STP_FRM	RW	帧结束中断标志, 高有效, 写 1 清零。	0
3	RB_DVP_IF_FIFO_OV	RW	接收 FIFO 溢出中断标志, 高有效, 写 1 清零。	0
2	RB_DVP_IF_FRM_DONE	RW	帧结束中断标志, 高有效, 写 1 清零。	0
1	RB_DVP_IF_ROW_DONE	RW	行结束中断标志, 高有效, 写 1 清零。	0
0	RB_DVP_IF_STR_FRM	RW	帧开始中断标志, 高有效, 写 1 清零。	0

DVP 接收 FIFO 状态寄存器 (R8_DVP_FIFO_ST)

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

7	Reserved	RO	保留。	0
[6:4]	RB_DVP_FIFO_CNT	RO	FIFO 计数器。	0
3	Reserved	RO	保留。	0
2	RB_DVP_FIFO_OV	RO	FIFO 溢出状态。 1: FIFO 溢出; 0: FIFO 未溢出。	0
1	RB_DVP_FIFO_FULL	RO	FIFO 满状态。 1: 缓存已满; 0: FIFO 未滿。	0
0	RB_DVP_FIFO_RDY	RO	FIFO 就绪状态。 1: FIFO 中有数据; 0: FIFO 中无数据。	0

DVP 接收图像行数寄存器 (R16_DVP_ROW_CNT)

位	名称	访问	描述	复位值
[15:0]	RB_DVP_ROW_CNT	RO	实际接收中，一帧图像数据包含的行数，此寄存器在帧结束时更新。 在 JPEG 格式下，该寄存器的值没有意义。	0

DVP 接收图像列数寄存器 (R16_DVP_COL_CNT)

位	名称	访问	描述	复位值
[15:0]	RB_DVP_COL_CNT	RO	原始数据格式下： 表示实际接收中，一行图像数据包含的 PCLK 数量，此寄存器在行结束时更新。 JPEG 格式下： 表示实际接收中，一帧数据包含的 PCLK 的数量，此寄存器在帧结束时更新。	0

第 13 章 USB2.0 控制器及收发器 (USBHS)

USB2.0 控制器具有 USB 主机控制器和 USB 设备控制器双重角色，并且内嵌 USB-PHY 单元。当作为主机控制器时，它可支持低速、全速和高速的 USB 设备。当作为设备控制器时，可以灵活设置为低速、全速或高速模式以适应各种应用。

13.1 主要特性

- 支持USB 2.0、USB 1.1、USB 1.0协议规范
- 支持USB Host主机功能和USB Device设备功能
- 主机和设备均支持控制传输、批量传输、中断传输、实时/同步传输
- 主机支持高速HUB
- 设备模式可配置为高速、全速、低速设备
- 设备模式下提供8组上下传输通道，支持配置16个端点号
- 支持DMA直接访问各端点缓冲区的数据
- 支持总线复位、挂起、远程唤醒和恢复功能
- 除设备端点0外，其他端点均支持最大1024字节的数据包，可使用双缓冲功能

13.2 功能描述

13.3 USB 设备模式配置

1. 设置 R8_USB_CTRL 寄存器，RB_USB_MODE 位为 0，配置 USB 设备模式；
2. 设置 R8_USB_CTRL 寄存器，清除 RB_USB_RESET_SIE 和 RB_USB_CLR_ALL 位为 0，设置 RB_USB_INT_BUSY 和 RB_USB_DMA_EN 位为 1，配置 RB_USB_SPTP_MASK 选择 USB 设备的速度，如果设置为高速设备，但当前主机是全速主机，那么控制器会自动降速切换到全速，实际的通讯速度可以查询 R8_USB_SPD_TYPE 寄存器。
3. 清除设备地址寄存器 R8_USB_DEV_AD 和中断标志寄存器 R8_USB_INT_FG，可选操作，使能需要的中断，写 R8_USB_INT_EN 寄存器；
4. 配置设备端点数据收发缓存区模式寄存器 R8_UEP4_1_MOD / R8_UEP2_3_MOD / R8_UEP5_6_MOD / R8_UEP7_MOD，以及收发控制寄存器 R8_UEPn_TX_CTRL / R8_UEPn_RX_CTRL；
5. 设置端点最大包接收长度 R16_UEPn_MAX_LEN 寄存器和端点数据收发起始地址 R32_UEPn_RX_DMA / R32_UEPn_TX_DMA；
6. 设置 R8_USB_CTRL 寄存器，RB_DEV_PU_EN 位为 1，启用 USB 设备功能。

13.4 USB 主机模式配置

1. 设置 R8_USB_CTRL 寄存器，RB_USB_MODE 位为 1，配置 USB 主机模式；
2. 设置 R8_USB_CTRL 寄存器，清除 RB_USB_RESET_SIE 和 RB_USB_CLR_ALL 位为 0，设置 RB_USB_INT_BUSY 和 RB_USB_DMA_EN 位为 1，配置 RB_USB_SPTP_MASK 选择 USB 主机的速度，如果设置为高速主机，但当前连接设备是全速，那么控制器会自动降速切换到全速，实际的通讯速度可以查询 R8_USB_SPD_TYPE 寄存器。

3. 清除设备地址寄存器 R8_USB_DEV_AD 和中断标志寄存器 R8_USB_INT_FG，可选操作，使能需要的中断，写 R8_USB_INT_EN 寄存器；
4. 配置主机端点数据收发模式寄存器 R8_UH_EP_MOD，打开收发通道。以及收发控制寄存器 R8_UH_RX_CTRL / R8_UH_TX_CTRL；
5. 设置端点最大包接收长度 R16_UH_MAX_LEN 寄存器和主机端点数据收发起始地址 R32_UEPn_RX_DMA / R32_UEPn_TX_DMA；
6. 设置 R8_UHOST_CTRL 寄存器的 RB_UH_AUTOSOF_EN 位为 1，开启端口自动发送 SOF 包。

13.5 寄存器描述

USB2.0 主从控制器（内置 PHY），可以灵活配置为主机功能或者设备功能。相关寄存器分为 3 个部分，部分寄存器是在主机和设备模式下进行复用的。

- USB全局寄存器
- USB设备控制器寄存器
- USB主机控制器寄存器

USBHS 相关寄存器物理基地址：0x40009000

13.5.1 全局寄存器描述

表 13-1 USBHS 全局寄存器列表

名称	偏移地址	描述	复位值
R8_USB_CTRL	0x00	USB 控制寄存器	0x06
R8_USB_INT_EN	0x02	USB 中断使能寄存器	0x00
R8_USB_DEV_AD	0x03	USB 地址寄存器	0x00
R16_USB_FRAME_NO	0x04	USBHS 帧号寄存器	0x0000
R8_USB_SUSPEND	0x06	USB 挂起控制寄存器	0x00
R8_USB_SPD_TYPE	0x08	USB 当前速度类型寄存器	0x00
R8_USB_MIS_ST	0x09	USB 杂项状态寄存器	0x20
R8_USB_INT_FG	0x0A	USB 中断标志寄存器	0x00
R8_USB_INT_ST	0x0B	USB 中断状态寄存器	0xXX
R16_USB_RX_LEN	0x0C	USB 接收长度寄存器	0xFFFF

USB 控制寄存器（R8_USB_CTRL）

位	名称	访问	描述	复位值
7	RB_USB_MODE	RW	USB 工作模式选择位： 0：设备模式（DEVICE）； 1：主机模式（HOST）。	0
[6:5]	RB_USB_SPTP_MASK	RW	设置 USB 总线信号传输速率选择位： 00：全速； 01：高速； 10：低速。	0
4	RB_DEV_PU_EN	RW	设备模式下，USB 设备使能和内部上拉电阻控制位： 1：使能 USB 设备传输并且启用内部上拉电阻； 0：不启用。	0
3	RB_USB_INT_BUSY	RW	USB 传输完成中断标志未清零前自动暂停使能位： 1：在中断标志 RB_USB_IF_TRANSFER 未清零前自	0

			动暂停，设备模式下自动应答忙 NAK，主机模式下自动暂停后续传输； 0：不暂停。	
2	RB_USB_RESET_SIE	RW	USB 协议处理器软件复位控制位： 1：强制复位 USB 协议处理器（SIE），需要软件清零； 0：不复位。	1
1	RB_USB_CLR_ALL	RW	1：清空 USB 中断标志和 FIFO，需要软件清零； 0：不清空。	1
0	RB_USB_DMA_EN	RW	USB 的 DMA 和 DMA 中断控制位： 1：使能 DMA 功能和 DMA 中断； 0：关闭 DMA。	0

USB 中断使能寄存器（R8_USB_INT_EN）

位	名称	访问	描述	复位值
7	RB_USB_IE_DEV_NAK	RW	USB 设备模式，接收到 NAK 中断使能： 1：使能相应中断； 0：禁止相应中断。	0
6	RB_USB_IE_ISOACT	RW	同步传输收到控制令牌包中断使能： 1：使能相应中断； 0：禁止相应中断。	0
5	RB_USB_IE_SETUPACT	RW	SETUP 事务完成中断： 1：使能相应中断； 0：禁止相应中断。	0
4	RB_USB_IE_FIFOOV	RW	内部 FIFO 溢出中断使能： 1：使能相应中断； 0：禁止相应中断。	0
3	RB_USB_IE_SOF	RW	USB 主机模式，SOF 定时中断使能： 1：使能相应中断； 0：禁止相应中断。 USB 设备模式，接收 SOF 包中断	0
2	RB_USB_IE_SUSPEND	RW	USB 总线挂起或唤醒事件中断使能： 1：使能相应中断； 0：禁止相应中断。	0
1	RB_USB_IE_TRANS	RW	USB 传输完成中断使能： 1：使能相应中断； 0：禁止相应中断。	0
0	RB_USB_IE_DETECT	RW	USB 主机模式下，USB 设备连接或断开事件中断使能： 1：使能相应中断； 0：禁止相应中断。	0
0	RB_USB_IE_BUSRST	RW	USB 设备模式，USB 总线复位事件中断使能： 1：使能相应中断； 0：禁止相应中断。	0

USB 设备地址寄存器（R8_USB_DEV_AD）

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0

[6:0]	USB_ADDR_MASK	RW	主机模式下是当前操作的 USB 设备(HUB)的地址。设备模式下是该 USB 设备的地址。	0
-------	---------------	----	---	---

USB 帧号寄存器 (R16_USB_FRAME_NO)

位	名称	访问	描述	复位值
[15:0]	USB_FRAME_NO	RO	帧号，主机模式下表示即将发送的 SOF 包的帧号，设备模式下表示当前接收到的 SOF 包的帧号。其中低 11 位为有效帧号，高 3 位为高速模式的微帧号。	0

USB_FRAME_NO 是 16 为寄存器，其中低 11 位表示 SOF 包帧号，高 3 为表示当前属于第几个微帧，可在操作高速 HUB 下进行中断、同步/实时传输时使用。

USB 挂起寄存器 (R8_USB_SUSPEND)

位	名称	访问	描述	复位值
[7:2]	Reserved	RO	保留。	0
1	RB_DEV_WAKEUP	RW	远程唤醒控制位： 1: 远程唤醒主机； 0: 无动作。	0
0	保留	RO	保留。	0

注：设备远程唤醒主机时，将 bUS_RESUME 位拉高再拉低即可。

USB 速度类型寄存器 (R8_USB_SPD_TYPE)

位	名称	访问	描述	复位值
[7:2]	Reserved	RO	保留。	0
[1:0]	RB_USBSPEED_MASK	RO	实际传输的速度。 在主机模式下，表示当前连接的设备速度类型， 在设备模式下，表示当前设备工作的速度类型； 00: 全速； 01: 高速； 10: 低速。	0

注：区别于 R8_USB_CTRL 寄存器中的 RB_USB_SPTP_MASK，RB_USB_SPTP_MASK 表示期望处于的最高速度，假设在设备模式下，设置 RB_USB_SPTP_MASK 为高速，当该设备连接在一个全速主机下，则实际的速度类型就是全速，通过查询 RB_USBSPEED_MASK 寄存器可以获知。在主机模式下，设置 RB_USB_SPTP_MASK 为高速，当连接一个全速设备时，则实际通讯速度就是全速，通过查询 RB_USBSPEED_MAS 寄存器可以获知。

USB 杂项状态寄存器 (R8_USB_MIS_ST)

位	名称	访问	描述	复位值
7	RB_USB_SOF_PRES	RO	USB 主机模式下 SOF 包预示状态位： 1: 将要发送 SOF 包，此时如有其它 USB 数据包将被自动延后； 0: 无 SOF 包发送。	X
6	RB_USB_SOF_ACT	RO	USB 主机模式下 SOF 包传输状态位： 1: 正在发出 SOF 包； 0: 发送完成或者空闲。	X
5	RB_USB_SIE_FREE	RO	USB 协议处理器的空闲状态位： 1: 协议器空闲； 0: 忙，正在进行 USB 传输。	1
4	RB_USB_FIFO_RDY	RO	USB 接收 FIFO 数据就绪状态位：	0

			1: 接收 FIFO 非空; 0: 接收 FIFO 为空。	
3	RB_USBBUS_RESET	R0	USB 总线复位状态位: 1: 当前 USB 总线处于复位态; 0: 当前 USB 总线处于非复位态。	0
2	RB_USBBUS_SUSPEND	R0	USB 挂起状态位: 1: USB 总线处于挂起态; 0: USB 总线处于非挂起态。	0
1	RB_USB_ATTACH	R0	USB 主机模式下端口的 USB 设备连接状态位: 1: 端口已经连接 USB 设备; 0: 端口没有 USB 设备连接。	0
0	RB_HOST_SPLIT_EN	R0	USB 主机模式下, SPLIT 包发送允许位: 1: 允许发送 SPLIT 包; 0: 禁止发送。	0

USB 中断标志寄存器 (R8_USB_INT_FG)

位	名称	访问	描述	复位值
7	Reserved	R0	保留。	0
6	RB_USB_IF_ISOACT	RW1	同步传输收到控制令牌包中断标志, 写 1 清零。 1: 开始发送/接收数据触发; 0: 无事件。	0
5	RB_USB_IF_SETUOACT	RW1	SETUP 事务完成中断标志, 写 1 清零。 1: SETUP 事务完成; 0: 无事件。	0
4	RB_USB_IF_FIFOOV	RW1	USB FIFO 溢出中断标志, 写 1 清零。 1: FIFO 溢出触发; 0: 无事件。	0
3	RB_USB_IF_HST_SOF	RW1	USB 主机模式下 SOF 定时中断标志, 写 1 清零。 1: SOF 传输完成触发; 0: 无事件。	0
2	RB_USB_IF_SUSPEND	RW1	USB 总线挂起或唤醒事件中断标志, 写 1 清零。 1: USB 挂起事件或唤醒事件触发; 0: 无事件。	0
1	RB_USB_IF_TRANSFER	RW1	USB 传输完成中断标志, 写 1 清零。 1: 一个 USB 传输完成触发; 0: 无事件。	0
0	RB_USB_IF_DETECT	RW1	USB 主机模式下 USB 设备连接或断开事件中断标志, 写 1 清零。 1: 检测到 USB 设备连接或断开触发; 0: 无事件。	0
0	RB_USB_IF_BUSRST	RW1	USB 设备模式下 USB 总线复位事件中断标志, 写 1 清零。 1: USB 总线复位事件触发; 0: 无事件。	0

USB 中断状态寄存器 (R8_USB_INT_ST)

位	名称	访问	描述	复位值
7	RB_USB_ST_NAK	R0	USB 设备模式下, NAK 响应状态位:	0

			1: 当前 USB 传输过程中回应 NAK; 0: 无 NAK 响应。	
6	RB_USB_ST_TGOK	RO	当前 USB 传输 DATA0/1 同步标志匹配状态位: 1: 同步; 0: 不同步。	0
[5:4]	RB_DEV_TOKEN_MASK	RO	设备模式下, 当前 USB 传输事务的令牌 PID 标识。	X
[3:0]	RB_DEV_ENDP_MASK	RO	设备模式下, 当前 USB 传输事务的端点号。	X
[3:0]	RB_HOST_RES_MASK	RO	主机模式下, 当前 USB 传输事务的应答 PID 标识, 0000 表示设备无应答或超时; 其它值表示应答 PID。	X

RB_DEV_TOKEN_MASK 用于 USB 设备模式下标识当前 USB 传输事务的令牌 PID: 00 表示 OUT 包; 01 表示 SOF 包; 10 表示 IN 包;

RB_HOST_RES_MASK 仅在主机模式下有效。在主机模式下, 若主机发送 OUT/SETUP 令牌包时, 则该 PID 是握手包 ACK/NAK/STALL/NYET, 或者是设备无应答/超时。若主机发送 IN 令牌包, 则该 PID 是数据包的 PID (DATA0/DATA1/DATA2/MDATA) 或者握手包 PID。

USB 接收长度寄存器 (R16_USB_RX_LEN)

位	名称	访问	描述	复位值
[15:0]	USB_RX_LEN	RO	当前 USB 端点接收到的数据计数, 其中低 11 位有效, 高 5 位固定为 0。	X

13.5.2 设备寄存器描述

USB 设备控制器提供了端点号 0-7 共 8 组双向端点配置寄存器, 端点号 1-7 的配置可映射端点号 8-15 的配置, 除端点 0 之外的所有端点的最大数据包长度都是 1024 字节, 端点 0 的最大数据包长度为 64 字节。

- 端点 0 是默认端点, 支持控制传输, 发送和接收共用一个 64 字节数据缓冲区
- 端点 1-15, 可配置独立的最高 1024 字节发送和接收缓冲区或者最高双 1024 字节数据缓冲区, 支持批量传输、中断传输和实时/同步传输。

端点 0 具有独立的 DMA 地址, 收发共用, 端点 1-7 (8-15) 的发送和接收各有一个 DMA 地址。通过 R8_UEPn*_MODE 寄存器可以设置数据缓冲区的模式为双缓冲或单缓冲。若使用双缓冲区模式, 该端点只能配置单方向传输。

每组端点都具有收发控制寄存器 UEPn_TX_CTRL、UEPn_RX_CTRL 和发送长度寄存器 UEPn_T_LEN 和 UEPn*_DMA (n=0~7), 用于配置该端点的同步触发位、对 OUT 事务和 IN 事务的响应以及发送数据的长度等参数。

作为 USB 设备所必要的 USB 总线上拉电阻可以由软件随时设置是否启用, 当 USB 控制寄存器 R8_USB_CTRL 中的 RB_DEV_PU_EN 置 1 时, 硬件根据 bUC_SPEED_TYPE 的速度设置, 在内部为 USB 总线的 DP/DM 引脚连接上拉电阻, 并启用 USB 设备功能。

当检测到 USB 总线复位、USB 总线挂起或唤醒事件, 或者当 USB 成功处理完数据发送或者数据接收后, USB 协议处理器都将设置相应的中断标志, 如果中断使能打开, 还会产生相应的中断请求。应用程序可以直接查询或在 USB 中断服务程序中查询并分析中断标志寄存器 R8_USB_INT_FG, 根据 RB_USB_IF_BUSRST 和 RB_USB_IF_SUSPEND 进行相应的处理; 并且, 如果 RB_USB_IF_TRANSFER 有效, 那么还需要继续分析 USB 中断状态寄存器 R8_USB_INT_ST, 根据当前端点号 RB_DEV_ENDP_MASK 和当前事务令牌 PID 标识 RB_DEV_TOKEN_MASK 进行相应的处理。如果事先设定了各个端点的 OUT 事务的同

步触发位 RB_UEP_R_TOG_MASK，那么可以通过 RB_USB_ST_TOGOK 判断当前所接收到的数据包的同步触发位是否与该端点的同步触发位匹配，如果数据同步，则数据有效；如果数据不同步，则数据应该被丢弃。每次处理完 USB 发送或者接收中断后，都应该正确修改相应端点的同步触发位，用于下次所发送的数据包或者下次所接收的数据包是否同步检测；另外，设置 RB_UEP_T_AUTOTOG/RB_UEP_R_AUTOTOG 可以在发送成功或者接收成功后自动翻转相应的同步触发位。

各个端点准备发送的数据在各自的缓冲区中，准备发送的数据长度是独立设定在 R16_UEPn_T_LEN 中；各个端点接收到的数据在各自的缓冲区中，但是接收到的数据长度都在 USB 接收长度寄存器 R16_USB_RX_LEN 中，可以在 USB 接收中断时根据当前端点号区分，每个端点可接收的最大包长度需要提前写入到 R16_UEPn_MAX_LEN 寄存器中。

表 13-2 USB 设备寄存器列表

名称	偏移地址	描述	复位值
R8_UEP4_1_MOD	0x10	端点 1(9)/4(8/12) 模式控制寄存器	0x00
R8_UEP2_3_MOD	0x11	端点 2(10)/3(11) 模式控制寄存器	0x00
R8_UEP5_6_MOD	0x12	端点 5(13)/6(14) 模式控制寄存器	0x00
R8_UEP7_MOD	0x13	端点 7(15) 模式控制寄存器	0x00
R32_UEP0_RT_DMA	0x14	端点 0 缓冲区的起始地址	0x000XXXXX
R32_UEP1_RX_DMA	0x18	端点 1(9) 接收缓冲区的起始地址	0x000XXXXX
R32_UEP2_RX_DMA	0x1C	端点 2(10) 接收缓冲区的起始地址	0x000XXXXX
R32_UEP3_RX_DMA	0x20	端点 3(11) 接收缓冲区的起始地址	0x000XXXXX
R32_UEP4_RX_DMA	0x24	端点 4(8/12) 接收缓冲区的起始地址	0x000XXXXX
R32_UEP5_RX_DMA	0x28	端点 5(13) 接收缓冲区的起始地址	0x000XXXXX
R32_UEP6_RX_DMA	0x2C	端点 6(14) 接收缓冲区的起始地址	0x000XXXXX
R32_UEP7_RX_DMA	0x30	端点 7(15) 接收缓冲区的起始地址	0x000XXXXX
R32_UEP1_TX_DMA	0x34	端点 1(9) 发送缓冲区的起始地址	0x000XXXXX
R32_UEP2_TX_DMA	0x38	端点 2(10) 发送缓冲区的起始地址	0x000XXXXX
R32_UEP3_TX_DMA	0x3C	端点 3(11) 发送缓冲区的起始地址	0x000XXXXX
R32_UEP4_TX_DMA	0x40	端点 4(8/12) 发送缓冲区的起始地址	0x000XXXXX
R32_UEP5_TX_DMA	0x44	端点 5(13) 发送缓冲区的起始地址	0x000XXXXX
R32_UEP6_TX_DMA	0x48	端点 6(14) 发送缓冲区的起始地址	0x000XXXXX
R32_UEP7_TX_DMA	0x4C	端点 7(15) 发送缓冲区的起始地址	0x000XXXXX
R16_UEP0_MAX_LEN	0x50	端点 0 接收最大长度包寄存器	0xFFFF
R16_UEP1_MAX_LEN	0x54	端点 1(9) 接收最大长度包寄存器	0xFFFF
R16_UEP2_MAX_LEN	0x58	端点 2(10) 接收最大长度包寄存器	0xFFFF
R16_UEP3_MAX_LEN	0x5C	端点 3(11) 接收最大长度包寄存器	0xFFFF
R16_UEP4_MAX_LEN	0x60	端点 4(8/12) 接收最大长度包寄存器	0xFFFF
R16_UEP5_MAX_LEN	0x64	端点 5(13) 接收最大长度包寄存器	0xFFFF
R16_UEP6_MAX_LEN	0x68	端点 6(14) 接收最大长度包寄存器	0xFFFF
R16_UEP7_MAX_LEN	0x6C	端点 7(15) 接收最大长度包寄存器	0xFFFF
R16_UEP0_T_LEN	0x70	端点 0 发送长度寄存器	0xFFFF
R8_UEP0_TX_CTRL	0x72	端点 0 发送控制寄存器	0x00
R8_UEP0_RX_CTRL	0x73	端点 0 接收控制寄存器	0x00
R16_UEP1_T_LEN	0x74	端点 1(9) 发送长度寄存器	0xFFFF
R8_UEP1_TX_CTRL	0x76	端点 1(9) 发送控制寄存器	0x00
R8_UEP1_RX_CTRL	0x77	端点 1(9) 接收控制寄存器	0x00
R16_UEP2_T_LEN	0x78	端点 2(10) 发送长度寄存器	0xFFFF
R8_UEP2_TX_CTRL	0x7A	端点 2(10) 发送控制寄存器	0x00

R8_UEP2_RX_CTRL	0x7B	端点 2(10) 接收控制寄存器	0x00
R16_UEP3_T_LEN	0x7C	端点 3(11) 发送长度寄存器	0xFFFF
R8_UEP3_TX_CTRL	0x7E	端点 3(11) 发送控制寄存器	0x00
R8_UEP3_RX_CTRL	0x7F	端点 3(11) 接收控制寄存器	0x00
R16_UEP4_T_LEN	0x80	端点 4(8/12) 发送长度寄存器	0xFFFF
R8_UEP4_TX_CTRL	0x82	端点 4(8/12) 发送控制寄存器	0x00
R8_UEP4_RX_CTRL	0x83	端点 4(8/12) 接收控制寄存器	0x00
R16_UEP5_T_LEN	0x84	端点 5(13) 发送长度寄存器	0xFFFF
R8_UEP5_TX_CTRL	0x86	端点 5(13) 发送控制寄存器	0x00
R8_UEP5_RX_CTRL	0x87	端点 5(13) 接收控制寄存器	0x00
R16_UEP6_T_LEN	0x88	端点 6(14) 发送长度寄存器	0xFFFF
R8_UEP6_TX_CTRL	0x8A	端点 6(14) 发送控制寄存器	0x00
R8_UEP6_RX_CTRL	0x8B	端点 6(14) 接收控制寄存器	0x00
R16_UEP7_T_LEN	0x8C	端点 7(15) 发送长度寄存器	0xFFFF
R8_UEP7_TX_CTRL	0x8E	端点 7(15) 发送控制寄存器	0x00
R8_UEP7_RX_CTRL	0x8F	端点 7(15) 接收控制寄存器	0x00

USB 端点 1(9)/4(8/12) 模式控制寄存器 (R8_UEP4_1_MOD)

位	名称	访问	描述	复位值
7	RB_UEP1_RX_EN	RW	1: 使能端点 1(9) 接收 (OUT); 0: 禁止端点 1(9) 接收。	0
6	RB_UEP1_TX_EN	RW	1: 使能端点 1(9) 发送 (IN); 0: 禁止端点 1(9) 发送。	0
5	Reserved	RO	保留。	0
4	RB_UEP1_BUF_MOD	RW	端点 1(9) 数据缓冲区模式控制位。 注: 该位为 1 时, UEP1_RX_EN 和 UEP1_TX_EN 不能同时为 1。	0
3	RB_UEP4_RX_EN	RW	1: 使能端点 4(8/12) 接收 (OUT); 0: 禁止端点 4(8/12) 接收。	0
2	RB_UEP4_TX_EN	RW	1: 使能端点 4(8/12) 发送 (IN); 0: 禁止端点 4(8/12) 发送。	0
1	Reserved	RO	保留。	0
0	RB_UEP4_BUF_MOD	RW	端点 4(8/12) 数据缓冲区模式控制位。 注: 该位为 1 时, UEP4_RX_EN 和 UEP4_TX_EN 不能同时为 1。	0

注: 端点 1 配置选项映射端点 9, 端点 4 配置选项映射端点 8 或 12。

端点 2(10)/3(11) 模式控制寄存器 (R8_UEP2_3_MOD)

位	名称	访问	描述	复位值
7	RB_UEP3_RX_EN	RW	1: 使能端点 3(11) 接收 (OUT); 0: 禁止端点 3(11) 接收。	0
6	RB_UEP3_TX_EN	RW	1: 使能端点 3(11) 发送 (IN); 0: 禁止端点 3(11) 发送。	0
5	Reserved	RO	保留。	0
4	RB_UEP3_BUF_MOD	RW	端点 3(11) 数据缓冲区模式控制位。 注: 该位为 1 时, UEP3_RX_EN 和 UEP3_TX_EN 不能同时为 1。	0
3	RB_UEP2_RX_EN	RW	1: 使能端点 2(10) 接收 (OUT);	0

			0: 禁止端点 2(10)接收。	
2	RB_UEP2_TX_EN	RW	1: 使能端点 2(10)发送 (IN); 0: 禁止端点 2(10)发送。	0
1	Reserved	RO	保留。	0
0	RB_UEP2_BUF_MOD	RW	端点 2(10)数据缓冲区模式控制位。 注: 该位为 1 时, UEP2_RX_EN 和 UEP2_TX_EN 不能同时为 1。	0

注: 端点 3 配置选项映射端点 11, 端点 2 配置选项映射端点 10。

端点 5(13)/6(14) 模式控制寄存器 (R8_UEP5_6_MOD)

位	名称	访问	描述	复位值
7	RB_UEP6_RX_EN	RW	1: 使能端点 6(14)接收 (OUT); 0: 禁止端点 6(14)接收。	0
6	RB_UEP6_TX_EN	RW	1: 使能端点 6(14)发送 (IN); 0: 禁止端点 6(14)发送。	0
5	Reserved	RO	保留。	0
4	RB_UEP6_BUF_MOD	RW	端点 6(14)数据缓冲区模式控制位。 注: 该位为 1 时, UEP6_RX_EN 和 UEP6_TX_EN 不能同时为 1。	0
3	RB_UEP5_RX_EN	RW	1: 使能端点 5(13)接收 (OUT); 0: 禁止端点 5(13)接收。	0
2	RB_UEP5_TX_EN	RW	1: 使能端点 5(13)发送 (IN); 0: 禁止端点 5(13)发送。	0
1	Reserved	RO	保留。	0
0	RB_UEP5_BUF_MOD	RW	端点 5(13)数据缓冲区模式控制位。 注: 该位为 1 时, UEP5_RX_EN 和 UEP5_TX_EN 不能同时为 1。	0

注: 端点 5 配置选项映射端点 13, 端点 6 配置选项映射端点 14。

端点 7(15) 模式控制寄存器 (R8_UEP7_MOD)

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0
3	RB_UEP7_RX_EN	RW	1: 使能端点 7(15)接收 (OUT); 0: 禁止端点 7(15)接收。	0
2	RB_UEP7_TX_EN	RW	1: 使能端点 7(15)发送 (IN); 0: 禁止端点 7(15)发送。	0
1	Reserved	RO	保留。	0
0	RB_UEP7_BUF_MOD	RW	端点 7(15)数据缓冲区模式控制位。 注: 该位为 1 时, UEP7_RX_EN 和 UEP7_TX_EN 不能同时为 1。	0

注: 端点 7 配置选项映射端点 15。

由 RB_UEPn_RX_EN 和 RB_UEPn_TX_EN 以及 RB_UEPn_BUF_MOD 组合分别配置 USB 端点 1-15 的数据缓冲区模式, 具体参考表 12-4。

表 12-4 端点 n 缓冲区模式 (n=1-7(8-15))

RB_UEPn_RX_EN	RB_UEPn_TX_EN	RB_UEPn_BUF_MOD	描述: 以 R32_UEPn_DMA 为起始地址由低向高排列
---------------	---------------	-----------------	--------------------------------

0	0	x	端点被禁用, 未用到 UEPn*_DMA 缓冲区。
1	0	0	接收 (OUT) 缓冲区首地址为 R32_UEPn_RX_DMA
1	0	1	RB_UEP_R_TOG_MASK=0, 使用缓冲区 R32_UEPn_RX_DMA RB_UEP_R_TOG_MASK=1, 使用缓冲区 R32_UEPn_TX_DMA
0	1	0	发送 (IN) 缓冲区首地址为 R32_UEPn_TX_DMA。
0	1	1	RB_UEP_T_TOG_MASK=0, 使用缓冲区 R32_UEPn_TX_DMA RB_UEP_T_TOG_MASK=1, 使用缓冲区 R32_UEPn_RX_DMA

注: 上表的配置选择支持 $n=1-7$; 端点 8-15 配置映射端点 1-7 配置。

USB 端点 0 收发缓冲区起始地址 (R32_UEP0_RT_DMA)

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
[16:0]	UEP0_RT_DMA	RW	端点 0 接收或发送缓冲区起始地址, 低 4 位固定为 0 (16 字节对齐)。	X

注: DMA 寻址 RAMX 区域。

USB 端点 n 接收缓冲区起始地址 (R32_UEPn_RX_DMA) ($n=1-7(8-15)$)

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
[16:0]	UEPn_RX_DMA	RW	端点 n 接收缓冲区起始地址, 低 4 位固定为 0 (16 字节对齐)。	X

注: DMA 寻址 RAMX 区域。

USB 端点 n 接收缓冲区起始地址 (R32_UEPn_TX_DMA) ($n=1-7(8-15)$)

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
[16:0]	UEPn_TX_DMA	RW	端点 n 发送缓冲区起始地址, 低 4 位固定为 0 (16 字节对齐)。	X

注: DMA 寻址 RAMX 区域。

端点 n 接收最大长度包寄存器 (R16_UEPn_MAX_LEN) ($n=0-7(8-15)$)

位	名称	访问	描述	复位值
[15:0]	UEPn_MAX_LEN	RW	端点 n 待接收最大包长度。此寄存器设置用于防止硬件 DMA 接收数据溢出 SRAM 边界。	X

注: 这个最大包长度决定了端点可接收数据最大长度, 超出此长度的数据会被丢弃, 不会写入缓冲区。

端点 n 发送长度寄存器 (R16_UEPn_T_LEN) ($n=0-7(8-15)$)

位	名称	访问	描述	复位值
[15:0]	UEPn_T_LEN	RW	设置 USB 端点 n 准备发送的数据字节数, 低 11 位有效, 高 5 位固定为 0	X

端点 n 发送控制寄存器 (R8_UEPn_TX_CTRL) ($n=0-7(8-15)$)

位	名称	访问	描述	复位值
[7:6]	Reserved	RO	保留。	0
5	RB_UEP_T_AUTOTOG	RW	同步触发位自动翻转使能控制位:	0

			1: 数据发送成功后自动翻转相应的同步触发位; 0: 不自动翻转, 可以手动切换。 <i>注: 只支持非 0 端点, 实时/同步传输不支持。</i>	
[4:3]	RB_UEP_T_TOG_MASK	RW	USB 端点 n 的发送器 (处理 IN 事务) 准备的同步触发位: 00: 发送 DATA0; 01: 发送 DATA1; 10: 发送 DATA2; 11: 发送 MDATA。	0
2	RB_UEP_TRES_NO	RW	1: 期望无响应, 用于实现非端点 0 的实时/同步传输。此时忽略 MASK_UEP_T_RES; 0: 期望响应。	0
[1:0]	RB_UEP_TRES_MASK	RW	端点 n 的发送器对 IN 事务的响应控制: 00: 数据就绪并期望 ACK; 10: 应答 NAK 或忙; 11: 应答 STALL 或错误。	0

端点 n 接收控制寄存器 (R8_UEPn_RX_CTRL) (n=0-7 (8-15))

位	名称	访问	描述	复位值
[7:6]	Reserved	RO	保留。	0
5	RB_UEP_R_AUTOTOG	RW	同步触发位自动翻转使能控制位: 1: 数据接收成功后自动翻转相应的同步触发位; 0: 不自动翻转, 可以手动切换。 <i>注: 只支持非 0 端点, 实时/同步传输不支持。</i>	0
[4:3]	RB_UEP_R_TOG_MASK	RW	USB 端点 n 的接收器 (处理 OUT 事务) 期望的同步触发位: 00: 期望 DATA0; 01: 期望 DATA1; 10: 期望 DATA2; 11: 期望 MDATA。 <i>注: 对于实时/同步传输无效。</i>	0
2	RB_UEP_RRES_NO	RW	1: 期望无响应, 用于实现非端点 0 的实时/同步传输。此时忽略 MASK_UEP_R_RES; 0: 期望响应。	0
[1:0]	RB_UEP_RRES_MASK	RW	端点 n 的接收器对 OUT 事务的响应控制: 00: 应答 ACK; 10: 应答 NAK 或忙; 11: 应答 STALL 或错误; 01: 应答 NYET。 <i>注: 对于实时/同步传输无效。</i>	0

13.5.3 USB 主机寄存器

USB 主机控制器提供了一组双向主机端点, 包括一个发送端点 OUT 和一个接收端点 IN, 数据包的最大长度是 1024 个字节, 支持控制传输、中断传输、批量传输和实时/同步传输。

主机端点发起的每一个 USB 事务, 在处理结束后总是自动设置中断标志 RB_USB_IF_TRANSFER。应用程序可以直接查询或在 USB 中断服务程序中查询并分析中断标志寄存器 R8_USB_INT_FG, 根据各中断标志分别进行相应的处理; 并且, 如果 RB_USB_IF_TRANSFER 有效, 那么还需要继续分析 USB 中

断状态寄存器 RB_USB_INT_ST，根据当前 USB 传输事务的应答 PID 标识 RB_HOST_RES_MASK 进行相应的处理。

如果事先设定了主机接收端点的 IN 事务的同步触发位 RB_UH_R_TOG_MASK，那么可以通过 RB_USB_ST_TOGOK 判断当前所接收到的数据包的同步触发位是否与主机接收端点的同步触发位匹配，如果数据同步，则数据有效；如果数据不同步，则数据应该被丢弃。每次处理完 USB 发送或者接收中断后，都应该正确修改相应主机端点的同步触发位，用于同步下次所发送的数据包和检测下次所接收的数据包是否同步；另外，通过设置 RB_UEP_R_AUTOTOG 可以在发送成功或接收成功后自动翻转相应的同步触发位。

USB 主机令牌设置寄存器 R16_UH_EP_PID 用于设置被操作的目标设备的端点号和本次 USB 传输事务的令牌 PID 包标识。SETUP 令牌和 OUT 令牌所对应的数据由主机发送端点提供，准备发送的数据在 R32_UH_TX_DMA 缓冲区中，准备发送的数据长度设置在 R16_UH_TX_LEN 中；IN 令牌所对应的数据由目标设备返回给主机接收端点，接收到的数据存放在 R32_UH_RX_DMA 缓冲区中，接收到的数据长度存放在 R16_USB_RX_LEN 中，主机端点可接收的最大包长度需要提前写入到 R16_UH_MAX_LEN 寄存器中。

表 13-3 USB 主机相关寄存器列表

名称	偏移地址	描述	复位值
R8_UHOST_CTRL	0x01	USB 主机控制寄存器	0x00
R8_UH_EP_MOD	0x11	USB 主机端点模式控制寄存器	0x00
R32_UH_RX_DMA	0x1C	USB 主机接收缓冲区起始地址	0x000XXXXX
R32_UH_TX_DMA	0x3C	USB 主机发送缓冲区起始地址	0x000XXXXX
R16_UH_MAX_LEN	0x58	USB 主机接收最大长度包寄存器	0xXXXX
R16_UH_EP_PID	0x78	USB 主机令牌设置寄存器	00
R8_UH_RX_CTRL	0x7B	USB 主机接收端点控制寄存器	0x00
R16_UH_TX_LEN	0x7C	USB 主机发送长度寄存器	0xXXXX
R8_UH_TX_CTRL	0x7E	USB 主机发送端点控制寄存器	0x00
R16_UH_SPLIT_DATA	0x80	USB 主机发送 SPLIT 包的数据	0xXXXX

USB 主机控制寄存器 (R8_UHOST_CTRL)

位	名称	访问	描述	复位值
7	RB_UH_AUTOSOF_EN	RW	自动产生 SOF 包使能控制。 1: 主机自动发生 SOF 包； 0: 不产生 SOF 包。 注：该位在设备断开连接时由硬件自动清零。	0
[6:3]	Reserved	RO	保留。	0
2	RB_UH_BUS_RESUME	RW	主机模式下，总线挂起状态，主机唤醒设备。	0
1	RB_UH_BUS_SUSPEND	RW	USB 主机发送挂起信号。	0
0	RB_UH_BUS_RESET	RW	USB 主机发送总线复位信号。	0

注：复位的时间由 RB_UH_BUS_RESET 的高电平持续时间决定。如果主机唤醒设备，则以 RB_UH_BUS_RESUME 边沿方式决定，所以唤醒只需将 RB_UH_BUS_RESUME 拉高再拉低即可。

USB 主机端点模式控制寄存器 (R8_UH_EP_MOD)

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
6	RB_UH_TX_EN	RW	主机发送端点发送 (SETUP/OUT) 使能位： 1: 使能端点发送；	0

			0: 禁止端点发送。	
[5:4]	Reserved	RO	保留。	0
3	RB_UH_RX_EN	RW	主机接收端点接收 (IN) 使能位: 1: 使能端点接收; 0: 禁止端点接收。	0
[2:0]	Reserved	RO	保留。	0

USB 主机接收缓冲区起始地址 (R32_UH_RX_DMA)

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
[16:0]	UH_RX_DMA	RW	主机端点数据接收缓冲区起始地址, 最低 4 位固定为 0 (16 字节对齐)。	X

注: DMA 寻址 RAMX 区域。

USB 主机发送缓冲区起始地址 (R32_UH_TX_DMA)

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
[16:0]	UH_TX_DMA	RW	主机端点数据发送缓冲区起始地址, 最低 4 位固定为 0 (16 字节对齐)。	X

注: DMA 寻址 RAMX 区域。

USB 主机接收最大长度包寄存器 (R16_UH_MAX_LEN)

位	名称	访问	描述	复位值
[15:0]	UH_MAX_LEN	RW	主机端点接收数据的最大包长度。低 11 位有效, 高 5 位固定为 0。	X

注: 这个最大包大小决定了端点可接收数据最大长度, 超出此长度的数据会被丢弃, DMA 不会送入自定义区域。

USB 主机令牌设置寄存器 (R16_UH_EP_PID)

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
[7:4]	RB_UH_TOKEN_MASK	RW	设置本次 USB 传输事务的令牌 PID 包标识。	0
[3:0]	RB_UH_EPNUM_MASK	RW	设置本次被操作的目标设备的端点号。	0

USB 主机接收端点控制寄存器 (R8_UH_RX_CTRL)

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
6	RB_UH_RDATA_NO	RW	1: 不期待数据包, 用于主机模式下的操作高速 HUB; 0: 期望数据包 (IN)。	0
5	RB_UH_R_AUTOTOG	RW	同步触发位自动翻转使能控制。 1: 数据接收成功后自动翻转相应的期待同步触发位; 0: 不自动翻转, 可以手动切换。	0
[4:3]	RB_UH_R_TOG_MASK	RW	主机接收器 (处理 IN 事务) 期望的同步触发位。 00: 期望 DATA0;	0

			01: 期望 DATA1; 10: 期望 DATA2; 11: 期望 MDATA。	
2	RB_UH_RRES_NO	RW	1: 无应答, 用于实现非端点 0 的实时/同步传输。 此时忽略 MASK_UEP_R_RES; 0: 接收数据成功后发送应答。	0
[1:0]	RB_UH_RRES_MASK	RW	主机接收器对 IN 事务的响应控制位: 00: 应答 ACK; 对于实时/同步传输无效。	0

USB 主机发送长度寄存器 (R16_UH_TX_LEN)

位	名称	访问	描述	复位值
[15:0]	UH_TX_LEN	RW	设置 USB 主机发送端点准备发送的数据字节数, 仅低 11 位有效, 高 5 位固定为 0。	X

USB 主机发送端点控制寄存器 (R8_UH_TX_CTRL)

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
6	RB_UH_TDATA_NO	RW	1: 不发送数据包 (PING/SPLIT); 0: 发送数据包 (OUT/SETUP)。	0
5	RB_UH_T_AUTOTOG	RW	同步触发位自动翻转使能控制位: 1: 数据发送成功后自动翻转相应的同步触发位; 0: 不自动翻转, 可以手动切换。	0
[4:3]	RB_UH_T_TOG_MASK	RW	USB 主机发送器 (处理 SETUP/OUT 事务) 准备的同步触发位 00 表示发送 DATA0; 01 表示发送 DATA1; 10 表示发送 DATA2; 11 表示发送 MDATA。	0
2	RB_UH_TRES_NO	RW	1: 无应答, 用于实现非端点 0 的实时/同步传输。 此时忽略 MASK_UEP_T_RES; 0: 发送数据成功后期待应答。	0
[1:0]	RB_UH_TRES_MASK	RW	USB 主机发送器对 SETUP/OUT 事务的响应控制位 00: 期望应答 ACK; 10: 期望应答 NAK 或忙; 11: 期望应答 STALL 或错误; 01: 期望应答 NYET。 对于实时/同步传输无效。	0

USB 主机发送 SPLIT 包的数据 (R16_UH_SPLIT_DATA)

位	名称	访问	描述	复位值
[15:0]	UH_SPLIT_DATA	RW	主机端点发送 SPLIT 包的数据内容, 低 12 位有效, 高 4 位固定为 0	X

第 14 章 SD/EMMC 控制器

系统提供 1 组 SDIO 控制器主机接口, 传输时钟可达 96MHz, 支持 1/4/8 线通讯模式, 可外接 SD/TF 卡、EMMC 卡等器件。应用程序代码可灵活设置数据收发的各种命令、应答包、有效数据包的模式和长度, 双缓冲长度切换界限等参数。

14.1 主要特征

- 支持SD物理层1.0、2.0规范, 支持SD3.0规范的UHS-I SDR50模式(向前兼容)
- 符合EMMC卡4.4和 4.5.1规范, 兼容5.0规范, 兼容HS200模式
- 通讯模式支持单线、四线, 八线模式
- 最高通讯时钟96MHz
- 灵活可设置的数据包长度、命令格式、应答状态
- 提供硬件自动在数据块间隔时停止时钟功能
- 支持SD卡、SDIO卡、EMMC卡等符合SD接口协议的设备
- 支持接口数据传输的硬件AES和SM4算法加解密
- DMA双缓冲功能

14.2 功能描述

14.2.1 通讯时钟

- 频率配置

控制器接口提供了 2 种时钟模式: 低速模式和高速模式。一般 SD 协议规定对于 SD 接口的设备的前期初始化使用 400KHz 左右的通讯时钟来保证更好的通讯兼容性, 当获取了外接设备的参数信息后, 可以根据其内部支持的配置, 切换到更高的时钟频率进行通讯。

设置 R16_EMMC_CLK_DIV 寄存器的 bit9 选择时钟模式, 设置 bit[4:0]分频值可得到最终的 SDIO 接口输出时钟。bit8 控制 EMMC 外设是否输出时钟信号到 MSDCK 引脚, 需要保证 MSDCK 引脚的 IO 模式配置为推挽输出。

- 相位配置

在和外部 SDIO 接口设备通讯时, 由于较高的时钟频率、硬件走线、器件特性等因素, 会导致信号采样出错。

系统 SDIO 控制器主机接口提供了输出时钟相位翻转 (偏移 180°) 功能, 通过设置寄存器 R16_EMMC_CLK_DIV 的 bit10 位写 1, 可以让输出的时钟信号物理翻转, 但是控制器内部的时钟仍然保持不变。这种方式可以调节通讯时序。

14.2.2 命令发送及响应

SDIO 主机接口向从机发出的命令长度固定的 48bit 数据包, 在 MCMD 线上串行传送。有的命令不需要从机响应, 有的命令需要从机响应不同长度的应答包或数据包。

- 一般的命令格式: 开始位(0)+传输位(1)+命令索引+参数+CRC7+结束位(1)

其中开始位、传输位、CRC7、结束位由硬件自动填充, 命令索引和参数根据命令的不同而不同,

需要应用代码传给硬件控制器。通过写 R32_EMMC_ARGUMENT 寄存器，配置 SD 命令中的参数域；写 R16_EMMC_CMD_SET 寄存器的 bit[5:0] 完成命令索引的填充，同时该寄存器的 bit[11:8] 配置硬件此命令期望的应答包检测。对 R16_EMMC_CMD_SET 寄存器的写操作将触发硬件在 MCMD 信号线上完成命令包序列传输。

- 短响应格式：开始位(0)+传输位(0)+命令索引+参数+CRC7+结束位(1)
- 长响应格式：开始位(0)+传输位(0)+保留域(111111b)+数据+结束位(1)

从机响应的应答包按长度分为 48 位和 136 位，其中对用户有效数据位分别位 32 位和 128 位。当主机命令发送后，可以通过查询 R16_EMMC_INT_FG 寄存器，如果标志位 RB_EMMC_IF_CMDDONE 置 1，即可读取 R32_EMMC_RESPONSE 寄存器获取从机接口应答包中的有效数据。如果应答包异常，寄存器 R16_EMMC_INT_FG 的 bit[2:0] 有相应标志置位。

14.2.3 连续读取多个数据块

对于 SD 协议命令中对于读取数据（通过数据线传输）的操作，需要如下配置：

1. 初始化：配置 R8_EMMC_CONTROL 寄存器，设置命令和数据的采样边沿，收发的数据线宽度，启用 DMA 功能。配置 RB_EMMC_TOCNT_MASK 寄存器，设置命令超时时间和数据超时时间。配置 R32_EMMC_DMA_BEG1 寄存器，设置读出数据保存的 DMA 起始地址。配置 R32_EMMC_BLOCK_CFG 寄存器，设置传输过程中单次数据块长度和总共需要读取的数据块个数。配置 R32_EMMC_TRAN_MODE 寄存器，设置 DMA 传输方向为“SD 到控制器”，可选设置是否需要在块传输间隔时间自动停止时钟，给应用代码足够的时间进行处理工作，之后软件再启动传输。
2. 发送命令给 SDIO 从机接口设备，此命令是需要从机设备从数据线传输数据给主机接口的命令，例如 SD 协议中的 CMD17、CMD18。
3. 控制器接口从数据线读取数据：当从机接口设备响应了主机命令后，如果设备准备好数据传输，会基于主机时钟信号从数据线上输出数据，起始信号为低电平。主机控制器端采样信号并由内部 DMA 转移到用户配置的 RAMX 区域。每转移 RB_EMMC_BKSIZE_MASK 设置大小的数据，表示完成一个数据块传输，同时内部总传输数据块数目减 1，直到所有的数据块传输结束，内部 DMA 将不再进行数据转移。此过程中，每完成一个数据块的读取，硬件将会置位 RB_EMMC_IF_BKGAP 标志，当所有的数据块读完，硬件将会置位 RB_EMMC_IF_TRANDONE 标志，如果控制器在等待接收过程中发生数据超时或者接收数据的 CRC 错误，将会分别置位 RB_EMMC_IF_DATTMO 和 RB_EMMC_IF_TRANERR 标志。应用代码可以配置中断使能寄存器 R16_EMMC_INT_EN 的相应中断控制位，当对应中断标志置位时，可以触发 EMMC 的中断服务。
4. 检测状态：R32_EMMC_STATUS 寄存器可是实时查询数据线 MDO 和命令线 MCMD 的电平状态，在一些协议规定的时序上做判断，MASK_BLOCK_NUM 域记录了当前命令传输中已经成被 DMA 成功转移的数据块个数。

14.2.4 连续写入多个数据块

1. 初始化：配置 R8_EMMC_CONTROL 寄存器，设置命令和数据的采样边沿，收发的数据线宽度，启用 DMA 功能。配置 RB_EMMC_TOCNT_MASK 寄存器，设置命令超时时间和数据超时时间。配置 R32_EMMC_BLOCK_CFG 寄存器，设置传输过程中单次数据块长度和总共需要读取的数据块个数。配置 R32_EMMC_TRAN_MODE 寄存器，设置 DMA 传输方向为“控制器到 SD”。
2. 发送命令给 SDIO 从机接口设备，此命令是需要主机接口通过数据线向从机设备传输数据的命令，

例如 SD 协议中的 CMD24、CMD25。

3. 控制器接口向数据线写数据：当从机接口设备响应了主机命令后，作出响应包并释放了数据线状态（非低电平），此时主机可以驱动数据线输出一个数据块。通过写 DMA 寄存器 R32_EMMC_DMA_BEG1 或者向 R32_EMMC_WRITE_CONT 寄存器执行写操作，都会启动主机接口驱动数据线，区别在于，操作寄存器 R32_EMMC_DMA_BEG1，硬件会将此时 R32_EMMC_DMA_BEG1 寄存器的值加载到内部，并从此地址取数据送入控制器发送数据；而对 R32_EMMC_WRITE_CONT 寄存器执行写操作，硬件将使用内部当前 DMA 移动到的位置开始向后继续取 SRAM 数据送入控制器，所以如果需要改变数据的 DMA 地址（与之前非连续）需要重新写 R32_EMMC_DMA_BEG1 寄存器执行继续数据线数据输出，如果与之前写数据的地址连续，硬件内部会自动取数据地址进行移动，可以直接对 R32_EMMC_WRITE_CONT 寄存器执行写操作来让控制器继续数据线数据输出。每输出 RB_EMMC_BKSIZE_MASK 设置大小的数据，表示完成一个数据块传输，同时内部总传输数据块数目减 1，直到所有的数据块传输结束，内部 DMA 将不再进行数据输出。此过程中，每完成一个数据块的读取，硬件将会置位 RB_EMMC_IF_BKGAP 标志，当所有的数据块读完，硬件将会置位 RB_EMMC_IF_TRANDONE 标志，如果控制器在等待接收过程中发生数据超时或者接收数据的 CRC 错误，将会分别置位 RB_EMMC_IF_DATTMO 和 RB_EMMC_IF_TRANERR 标志。应用代码可以配置中断使能寄存器 R16_EMMC_INT_EN 的相应中断控制位，当对应中断标志置位时，可以触发 EMMC 的中断服务。

注：控制器接口向外部设备写入数据块时，需要等待命令发送并应答完成后才能设置 DMA 地址，因为设置 DMA 地址的操作即通知硬件驱动数据线，此时命令如果还没有发送完成，设备将不会接受数据。

注：在某次命令执行的传输过程中（包括数据读和数据写），如果出现 CRC 错误（RB_EMMC_IF_TRANERR），硬件将不计入成功计数，也不会对总传输块计数进行递减，会继续下一个块的传输。但是 DMA 仍然会将此 CRC 错误的数据块放入相应的 SRAM 地址区域。应用代码要注意此刻的处理。

14.2.5 DMA 双缓冲功能

系统提供了 DMA 双缓冲功能，有 2 个 DMA 地址寄存器 R32_EMMC_DMA_BEG1 和 R32_EMMC_DMA_BEG2，以及 DMA 切换的数据块长度阈值 RB_EMMC_DMATN_CNT。当 R32_EMMC_DMA_BEG1 寄存器地址开始的数据块长度阈值传输完成，硬件将切换到 R32_EMMC_DMA_BEG2 设置地址区域进行数据传输，并在达到传输阈值后再次切换到 R32_EMMC_DMA_BEG1 区，以此循环，直到 RB_EMMC_BKNUM_MASK 配置的总数据块数量全部传输完成。

每次对 R32_EMMC_BLOCK_CFG 寄存器的 RB_EMMC_BKNUM_MASK 写入非 0 数据，将会启动数据传输，同时硬件也会将 DMA 指向 R32_EMMC_DMA_BEG1 地址区域。传输过程中的标志置位条件和前面描述一致。应用代码可选择开启相应中断位触发 EMMC 中断服务。

14.3 寄存器描述

SD/EMMC 相关寄存器基地址：0x4000A000

表 14-1 EMMC 相关寄存器列表

名称	偏移地址	描述	复位值
R16_EMMC_CLK_DIV	38h	时钟配置寄存器	0x0213
R32_EMMC_ARGUMENT	00h	命令参数寄存器	0x00000000
R16_EMMC_CMD_SET	04h	命令设置寄存器	0x0000
R32_EMMC_RESPONSE0	08h	应答参数寄存器 0	0x00000000

R32_EMMC_RESPONSE1	0Ch	应答参数寄存器 1	0x00000000
R32_EMMC_RESPONSE2	10h	应答参数寄存器 2	0x00000000
R32_EMMC_RESPONSE3	14h	应答参数寄存器 3	0x00000000
R32_EMMC_WRITE_CONT	14h	继续写启动寄存器	0x00000000
R8_EMMC_CONTROL	18h	控制寄存器	0x15
R8_EMMC_TIMEOUT	1Ch	超时计数寄存器	0x0C
R32_EMMC_STATUS	20h	状态寄存器	0x00000000
R16_EMMC_INT_FG	24h	中断标志寄存器	0x0000
R16_EMMC_INT_EN	28h	中断使能寄存器	0x0000
R32_EMMC_DMA_BEG1	2Ch	DMA 起始地址 1 寄存器	0x0000XXXX
R32_EMMC_BLOCK_CFG	30h	传输块配置寄存器	0x00000000
R32_EMMC_TRAN_MODE	34h	传输模式寄存器	0x00000000
R32_EMMC_DMA_BEG2	3Ch	DMA 起始地址 2 寄存器	0x0000XXXX

时钟配置寄存器 (R16_EMMC_CLK_DIV)

位	名称	访问	描述	复位值
[15:11]	Reserved	RO	保留。	0
10	RB_EMMC_PHASEINV	RW	SDCK 时钟输出相位反转。	0
9	RB_EMMC_CLKMode	RW	时钟频率模式选择位： 1: 高速模式，25M-100MHz； 0: 低速模式，400KHz。	1
8	RB_EMMC_CLKOE	RW	SD 物理时钟信号线输出控制位： 1: 打开，输出通讯时钟； 0: 关闭。	0
[7:5]	Reserved	RO	保留。	0
[4:0]	RB_EMMC_DIV_MASK	RW	SD 控制器时钟 (SDCLK) 分频因子： 当 bSDCLK_Mode=1, 则 SDCLK = 480M/MASK_CLK_PRE； 当 bSDCLK_Mode =0, 则 SDCLK = 480M/MASK_CLK_PRE/64。 最小值为 2，写 1 等效关闭 SDC 模时钟。	13h

命令参数寄存器 (R32_EMMC_ARGUMENT)

位	名称	访问	描述	复位值
[31:0]	EMMC_ARGUMENT	RW	32 位命令参数寄存器。	0

命令设置寄存器 (R16_EMMC_CMD_SET)

位	名称	访问	描述	复位值
[15:12]	Reserved	RO	保留。	0
11	RB_EMMC_CKIDX	RW	校验应答的命令索引： 1: 需要； 0: 不需要。	0
10	RB_EMMC_CKCRG	RW	校验应答的 CRC： 1: 需要； 0: 不需要。	0
[9:8]	RB_EMMC_RPTY_MASK	RW	期望应答类型： 00b: 无应答； 01b: 应答长度为 136 位； 10b: 应答长度为 48 位；	0

			11b: 应答长度为 48 位, 且为 R1b 类型应答。	
[7:6]	Reserved	RO	保留。	0
[5:0]	RB_EMMC_CMDIDX_MASK	RW	当前发送命令的索引号。	0

应答参数寄存器 (R32_EMMC_RESPONSE)

位	名称	访问	描述	复位值
[31:0]	R32_EMMC_RESPONSE0	RO	应答参数寄存器 0	0
[63:32]	R32_EMMC_RESPONSE1	RO	应答参数寄存器 1	0
[95:64]	R32_EMMC_RESPONSE2	RO	应答参数寄存器 2	0
[127:96]	R32_EMMC_RESPONSE3	RO	应答参数寄存器 3	0
[127:96]	R32_EMMC_WRITE_CONT	WO	复用 EMMC_RESPONSE3 寄存器, 用于在多块写过程中, 启动写操作。	0

注: 应答长度为 136 位时, 有效数据为 128 位, 应答长度为 48 位时, 有效数据长度为 32 位, EMMC_RESPONSE3 寄存器存放的是应答的有效数据参数。

EMMC_RESPONSE3 寄存器存在复用, 寄存器复用为: 在使用 CMD25 命令向卡连续写入多块数据时, 在块传输完成时, 如果不需要更改 DMA 地址, 写此寄存器的动作, 启动当前地址继续写数据操作。如果需要更改 DMA 地址, 则写 DMA 地址寄存器, 就会启动写操作, 就不用再用写寄存器的方式启动。

控制寄存器 (R8_EMMC_CONTROL)

位	名称	访问	描述	复位值
[7:6]	Reserved	RO	保留。	0
5	RB_EMMC_NEGSMP	RW	Cmd 和 Data 信号线采样模式选择位: 1: 下降沿采样; 0: 上升沿采样。	0
4	RB_EMMC_RST_LGC	RW	模块内部逻辑收发复位。 1: 执行复位; 0: 正常工作;	1
3	RB_EMMC_DMAEN	RW	控制器 DMA 使能: 1: 使能 DMA; 0: 关闭 DMA。	0
2	RB_EMMC_ALL_CLR	RW	内部 FIFO 和中断标志复位。 1: 执行复位; 0: 正常工作。	1
[1:0]	RB_EMMC_LW_MASK	RW	收发数据有效数据宽度: 00: 收发器仅使用 dat[0], 单数据线; 01: 收发器使用 dat[3:0], 4 数据线; 10: 收发器使用 dat[7:0], 8 数据线。	01b

超时控制寄存器 (R8_EMMC_TIMEOUT)

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0
[3:0]	RB_EMMC_TOCNT_MASK	RW	应答/数据超时配置。 非 0: 设置超时时间, 有效数值 0-12; 计算: 模块时钟周期 * 4194304 * MASK_TOCNT。 例如: SDCLK 周期为 10ns, 写入 12, 则超时时间为 10ns * (4194304) * (12) = 503ms。 0: 不使能内部超时机制。	Ch

注1: 上述数据超时包含下列 4 种情况:

- 1) R1b 应答之后的 DAT[0] busy 超时;
- 2) 写数据块时, CRC status 之后的 DAT[0] busy 超时;
- 3) 写数据块时, 等待 CRC status 超时;
- 4) 读数据块时, 等待起始位超时。

注2: 命令的应答同样支持超时机制, 如果应答超时, 则由中断寄存器中的 R16_EMMC_INT_FG 中断给出。命令超时采用协议给出的超时最大值: 64Tsdclk。

状态指示寄存器 (R32_EMMC_STATUS)

位	名称	访问	描述	复位值
[32:18]	Reserved	RO	保留。	0
17	RB_EMMC_DATOSTA	RO	DAT0 数据线当前电平状态。 1: 高电平; 0: 低电平。	0
16	RB_EMMC_CMDSTA	RO	CMD 信号线当前电平状态 1: 高电平; 0: 低电平。	0
[15:0]	MASK_BLOCK_NUM	RO	当前多块传输操作中, 已经成功传输的块数。	0

中断标志寄存器 (R16_EMMC_INT_FG)

位	名称	访问	描述	复位值
[15:10]	保留	RO	保留。	0
9	RB_EMMC_IF_SDIOINT	RW1	SDIO 卡中断标志, 写 1 清零。 1: SDIO 卡产生卡中断; 0: 无事件。	0
8	RB_EMMC_IF_FIFO_OV	RW1	FIFO 溢出标志, 写 1 清零。 1: FIFO 溢出触发; 0: 无事件。	0
7	RB_EMMC_IF_BKGAP	RW1	单块传输完成标志, 写 1 清零。 1: 单块收发完成触发; 0: 无事件。	0
6	RB_EMMC_IF_TRANDONE	RW1	所有块数传输完成标志, 写 1 清零。 1: 请求块数都传输完成触发; 0: 无事件。	0
5	RB_EMMC_IF_TRANERR	RW1	数据传输 CRC 错误标志, 写 1 清零。 1: CRC 错误触发; 0: 无事件。	0
4	RB_EMMC_IF_DATTMO	RW1	数据超时标志, 写 1 清零。 1: 数据超时触发; 0: 无事件。	0
3	RB_EMMC_IF_CMDDONE	RW1	命令完成标志, 写 1 清零。 1: 发送命令, 并且收到应答完毕; 0: 无事件。	0
2	RB_EMMC_IF_REIDX_ER	RW1	应答索引号校验错误标志, 写 1 清零。 1: 应答索引号校验错误触发; 0: 无事件。	0
1	RB_EMMC_IF_RECRC_WR	RW1	应答 CRC 校验错误标志, 写 1 清零。 1: 应答 CRC 校验错误触发; 0: 无事件。	0
0	RB_EMMC_IF_RE_TMOUT	RW1	接收应答超时标志, 写 1 清零。	0

			1: 应答超时触发; 0: 无事件。	
--	--	--	-----------------------	--

中断使能寄存器 (R16_EMMC_INT_EN)

位	名称	访问	描述	复位值
[15:10]	Reserved	RO	保留。	0
9	RB_EMMC_IE_SDIOINT	RW	SDIO 卡中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0
8	RB_EMMC_IE_FIFO_OV	RW	FIFO 溢出中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0
7	RB_EMMC_IE_BKGAP	RW	单块传输完成中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0
6	RB_EMMC_IE_TRANDONE	RW	所有块传输完成中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0
5	RB_EMMC_IE_TRANERR	RW	块传输 CRC 错误中断: 1: 使能相应中断; 0: 禁止相应中断。	0
4	RB_EMMC_IE_DATTMO	RW	数据超时中断: 1: 使能相应中断; 0: 禁止相应中断。	0
3	RB_EMMC_IE_CMDDONE	RW	命令完成中断: 1: 使能相应中断; 0: 禁止相应中断。	0
2	RB_EMMC_IE_REIDX_ER	RW	应答索引校验错误中断: 1: 使能相应中断; 0: 禁止相应中断。	0
1	RB_EMMC_IE_RECRC_WR	RW	应答 CRC 校验错误中断: 1: 使能相应中断; 0: 禁止相应中断。	0
0	RB_EMMC_IE_RE_TMOUT	RW	命令应答超时中断: 1: 使能相应中断; 0: 禁止相应中断。	0

数据块 DMA 起始地址寄存器 (R32_EMMC_DMA_BEG1)

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
[16:0]	RB_EMMC_DMAAD1_MASK	RW	读写数据缓存区起始地址, 低 4 位固定为 0 (16 字节对齐)。	0

注: 此寄存器在从 SD 中读数据时, 保存着读到的数据在 SRAM 中的起始地址。在写向 SD 卡写数据时, 保存着将要写的的数据在 SRAM 中的起始地址。寻址在 RAMX 区域。

如果执行连续的多块读写 SD 操作, 那么在单块传输完成后 (RB_EMMC_IF_BKGAP), 用户根据需要进行写 DMA_BEG1 寄存器来更改 DMA 地址。不可在传输过程中更改 DMA 地址, 否则可能引起数据计数错误。在进行连续多块写时, 单块传输完成后, 需要写 R32_EMMC_WRITE_CONT 或 DMA_BEG1 寄存器的

方式启动继续写操作。多块读不需要。

传输块配置寄存器 (R32_EMMC_BLOCK_CFG)

位	名称	访问	描述	复位值
[31:28]	Reserved	RO	保留。	0
[27:16]	RB_EMMC_BKSIZE_MASK	RW	单块传输大小 (1-2048 字节)。	0
[15:0]	RB_EMMC_BKNUM_MASK	RW	本次 DMA 要传输的块计数 (1~65535 块), 内部自动清零, 块数不为零则启动接收或发送。	0

传输模式寄存器 (R32_EMMC_TRAN_MODE)

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
16	RB_EMMC_DULEDMA_EN	RW	使能DMA双缓冲功能。	0
15	Reserved	RO	保留。	0
[14:8]	RB_EMMC_DMATN_CNT	RW	双缓冲模式下, 设置缓冲区切换的块计数值。	10h
[7:6]	RB_EMMC_FIFO_RDY	RW	写EMMC时FIFO ready选择信号, 调节内部FIFO和DMA关系。	01b
5	Reserved	RW	必须写0。	0
4	RB_EMMC_AUTOGAPSTOP	RW	硬件自动数据块停时钟使能。	0
3	Reserved	RW	必须写0。	0
2	RB_EMMC_MODE_BOOT	RW	设置eMMC卡传输模式: 1: 引导模式; 0: 正常模式。 注: 仅用于eMMC卡。	0
1	RB_EMMC_GAP_STOP	RW	数据块完成时钟停止模式。读块数据时, 将此位置 1, 则计数一个数据块的采样后硬件自动关闭时钟输出, 开启新的块传输, 需要软件清 0, 延时 1 个 SCLK 时间后, 重新置 1 再次计数。如果使能了 RB_EMMC_AUTOGAPSTOP 功能, 此位硬件置 1, 软件只需清 0。	0
0	RB_EMMC_DMA_DIR	RW	DMA 传输的方向: 1: 控制器到 SD; 0: SD到控制器。	0

数据块 DMA 起始地址寄存器 (R32_EMMC_DMA_BEG2)

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
[16:0]	RB_EMMC_DMAAD2_MASK	RW	读写数据缓存区起始地址, 低 4 位固定为 0 (16 字节对齐)。	0

注: 此寄存器在从 SD 中读数据时, 保存着读到的数据在 SRAM 中的起始地址。在写向 SD 卡写数据时, 保存着将要写的数据在 SRAM 中的起始地址。寻址在 RAMX 区域。

第 15 章 加密模块 (ECDC)

系统内置了分组密码算法模块，支持 AES 和 SM4 两种分组密码算法以及电子密码本 (ECB) 和计数器 (CTR) 模式。模块以 128 位数据大小为基本单位完成一次加解密过程，提供了部分外设接口和 SRAM 之间数据转移时硬件自动加解密以及本身 SRAM 数据的硬件加解密模式。

15.1 主要特征

- SM4 算法 128 bit 密钥的 ECB 模式和 CTR 模式
- AES 算法 128/192/256 bit 密钥的 ECB 模式和 CTR 模式
- 支持存储器到存储器、存储器到 HSPI、存储器到 EMMC 的数据加解密
- 支持软件写 SFR 的方式直接加密单个 128bit 数据
- 支持 DMA 的方式加解密软件指定长度的数据块

15.2 AES/SM4 算法

AES (Advanced Encryption Standard) 算法是一种区块加密法，采用对称分组密码体制，是对称密钥加密中最流行的算法之一。SM4 分组密码算法一般是用于无线局域网和可信计算机的专用分组密码算法，同时也可用于其它环境下的数据加密保护。

在数据加解密过程中，需要载入密钥。对于 AES 算法，根据设置密钥长度为 128/192/256 比特，使用时，需分别将用户密钥扩展为 $11 \times 128/13 \times 128/15 \times 128$ -bit 的扩展密钥。而 SM4 算法，将 128-bit 的用户密钥扩展成为 32×32 -bit 的扩展密钥。这些扩展密钥被保存在内部寄存器中，以方便在加解密时使用。

15.3 ECB 与 CTR 模式

AES/SM4 支持两种模式，电子密码本 (ECB) 模式和计数器 (CTR) 模式，其中 CTR 模式的安全性能要高于 ECB 模式。如图 16-1 和 16-2 所示的工作框图。

图 15-1 ECB 模式加解密过程

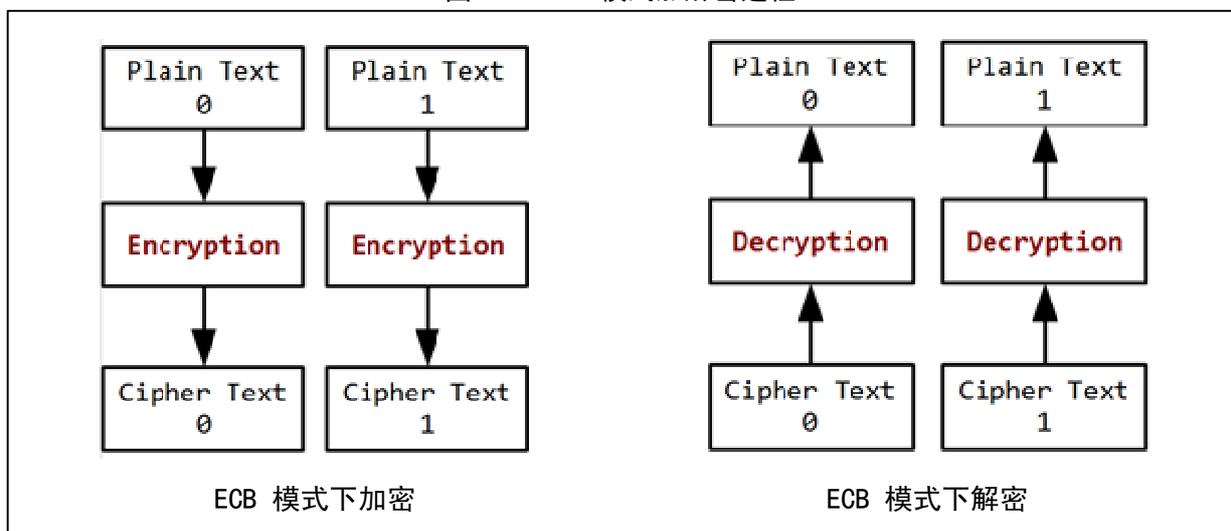
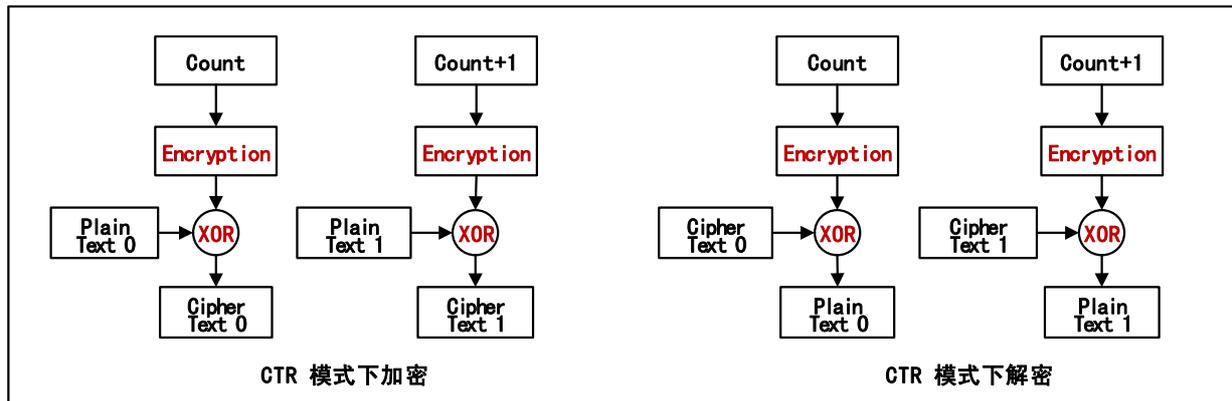


图 15-2 CTR 模式加解密过程



在 ECB 模式下，明文 (plain text) 与密文 (cipher text) 是一一对应的，加密之后的明文直接作为密文；而在 CTR 模式下，需要预先载入 128-bit 的计数值，将计数值加密，加密之后的计数值与明文异或作为密文。值得注意的是，CTR 解密模式下，也是只对计数值进行加密，而非解密。

15.4 模块应用

15.4.1 加解密模块初始化

- 时钟配置：设置 R16_ECEC_CTRL 寄存器的 bit[5:4]，配置模块运行的时钟。如果不使用此模块可以将其设置为 0，关闭模块降低功耗。
- 数据大小端：设置 R16_ECEC_CTRL 寄存器的 bit13，可选择加解密数据的大小端存储方式。按 128bits（16 字节）一包大小端存储。
- 密钥扩展：加解密过程中使用的不是用户密钥而是扩展密钥。当设置完用户密钥长度和内容后，需要通知硬件执行密钥扩展动作，当密钥扩展后会被保存到模块内部寄存器。之后才可以正常进行数据加解密工作。设置 R16_ECEC_CTRL 寄存器 bit[11:8]，选择加解密算法、分组密码模式、密钥长度；然后配置用户密钥寄存器组 R32_ECDC_KEY，填入用户密钥内容。将 R16_ECEC_CTRL 寄存器的 bit0 置 1 再清 0 后，硬件开始执行密钥扩展功能。应用软件查询 R8_ECDC_INT_FG 寄存器的 RB_ECDC_IF_EKDONE 位，如果置 1 表示密钥扩展完成，清除 RB_ECDC_IF_EKDONE 位状态。如果开启了 R8_ECDC_INT_EN 寄存器的 RB_ECDC_IE_EKDONE 功能，在 RB_ECDC_IF_EKDONE 标志置位的同时将触发中断服务。
- 计数器初值：如果使用 CTR 模式进行加解密，还需一个初始的计数器值。设置 CTR 模式计数值寄存器组 R32_ECDC_IV 填入 128 位计数器值，每次成功完成一次数据加解密（128bit 数据）模块内部的计数器值将加 1。ECB 模式无需此计数器值。

15.4.2 几种加解密配置

模块的加解密数据流包含存储器到存储器、存储器到 HSPI、存储器到 EMMC、单次 128bit 数据加解密几种方式，具体可以参考表 15-1 的配置。

表 15-1 不同数据流模式配置

加解密数据流	RB_ECDC_WRSRAM_EN	RB_ECDC_RDPERI_EN	RB_ECDC_WRPERI_EN	RB_ECDC_MODE_SEL	其他
RAMX 到外设加密	0	0	1	0	一旦开启加解密功能，将在数据转移过程中进行转换，
RAMX 到外设解密	0	0	1	1	

外设到 RAMX 加密	0	1	0	0	RAMX 的地址就是外设中配置的 DMA 地址。	
外设到 RAMX 解密	0	1	0	1		
RAMX 数据加密	1	1	0	0	需配置寄存器 R32_ECDC_SRAM_ADDR 和 R32_ECDC_SRAM_LEN	
		0	1			
RAMX 数据解密	1	1	0	1		
		0	1			
128bits 数据单次加密	-	1	0	0	写 R32_ECDC_SGSD 寄存器， 填入原始数据（明文或密文）	
		0	1			
128bits 数据单次解密	-	1	0	1		读 R32_ECDC_SGRT 寄存器， 读出加解密后的结果数据。
		0	1			

注：外设包括 HSPI 和 EMMC 模块。

表 15-1 中描述的几种方式，存储器和外设之间数据的转换由硬件自动穿插在数据转移过程中，外设功能接口的传输启动和完成标志同时代表着加解密转换的开始和完成。

存储器到存储器之间的数据转换开始由 R32_ECDC_SRAM_LEN 长度寄存器写入非 0 数据启动，通过查询 R8_ECDC_INT_FG 寄存器的 RB_ECDC_IF_WRSRAM 位，如果置 1 表示存储器到存储器数据转换完成，清除 RB_ECDC_IF_WRSRAM 位状态。如果开启了 R8_ECDC_INT_EN 寄存器的 RB_ECDC_IE_WRSRAM 功能，在 RB_ECDC_IF_WRSRAM 标志置位的同时将触发中断服务。

单次加解密方式必须配置为大端方式（RB_ECDC_DAT_MOD=1）执行。128bits 数据单次加解密转换开始向 R32_ECDC_SGSD 寄存器的最高 32 位写入操作启动，通过查询 R8_ECDC_INT_FG 寄存器的 RB_ECDC_IF_SINGLE 位，如果置 1 表示单次 128bits 数据转换完成，清除 RB_ECDC_IF_SINGLE 位状态。如果开启了 R8_ECDC_INT_EN 寄存器的 RB_ECDC_IE_SINGLE 功能，在 RB_ECDC_IF_SINGLE 标志置位的同时将触发中断服务。

15.5 寄存器描述

ECDC 模块相关寄存器物理基地址：0x40007000

表 15-2 ECDC 模块寄存器列表

名称	偏移地址	描述	复位值
R16_ECEC_CTRL	0x00	ECDC 控制寄存器	0x0020
R8_ECDC_INT_EN	0x02	ECDC 中断使能寄存器	0x00
R8_ECDC_INT_FG	0x06	ECDC 中断标志寄存器	0x00
R32_ECDC_KEY_255T224	0x08	用户密钥寄存器 7	0xFFFFFFFF
R32_ECDC_KEY_223T192	0x0C	用户密钥寄存器 6	0xFFFFFFFF
R32_ECDC_KEY_191T160	0x10	用户密钥寄存器 5	0xFFFFFFFF
R32_ECDC_KEY_159T128	0x14	用户密钥寄存器 4	0xFFFFFFFF
R32_ECDC_KEY_127T96	0x18	用户密钥寄存器 3	0xFFFFFFFF
R32_ECDC_KEY_95T64	0x1C	用户密钥寄存器 2	0xFFFFFFFF
R32_ECDC_KEY_63T32	0x20	用户密钥寄存器 1	0xFFFFFFFF
R32_ECDC_KEY_31T0	0x24	用户密钥寄存器 0	0xFFFFFFFF
R32_ECDC_IV_127T96	0x28	CTR 模式计数值寄存器 3	0xFFFFFFFF
R32_ECDC_IV_95T64	0x2C	CTR 模式计数值寄存器 2	0xFFFFFFFF
R32_ECDC_IV_63T32	0x30	CTR 模式计数值寄存器 1	0xFFFFFFFF
R32_ECDC_IV_31T0	0x34	CTR 模式计数值寄存器 0	0xFFFFFFFF
R32_ECDC_SGSD_127T96	0x40	单次加解密原始数据 3	0x00000000
R32_ECDC_SGSD_95T64	0x44	单次加解密原始数据 2	0x00000000

R32_ECDC_SGSD_63T32	0x48	单次加解密原始数据 1	0x00000000
R32_ECDC_SGSD_31T0	0x4C	单次加解密原始数据 0	0x00000000
R32_ECDC_SGRT_127T96	0x50	单次加解密结果 3	0x00000000
R32_ECDC_SGRT_95T64	0x54	单次加解密结果 2	0x00000000
R32_ECDC_SGRT_63T32	0x58	单次加解密结果 1	0x00000000
R32_ECDC_SGRT_31T0	0x5C	单次加解密结果 0	0x00000000
R32_ECDC_SRAM_ADDR	0x60	加解密 SRAM 区域起始地址	0x00000000
R32_ECDC_SRAM_LEN	0x64	加解密 SRAM 大小 (/128bit)	0x00000000

ECDC 控制寄存器 (R16_ECDC_CTRL)

位	名称	访问	描述	复位值
[15:14]	Reserved	RO	保留。	0
13	RB_ECDC_DAT_MOD	WO	加解密数据大小端方式选择。 1: 大端方式; 0: 小端方式。 注: 单次 128bits 加解密下必须配置为大端方式。	0
12	Reserved	RO	保留。	0
[11:10]	RB_ECDC_KLEN_MASK	RW	密钥长度设置。 00: 128-bit; 01: 192-bit; 10: 256-bit; 11: 保留。	0
9	RB_ECDC_CIPHER_MOD	RW	分组密码模式选择。 1: CTR 模式; 0: ECB 模式。	0
8	RB_ECDC_ALGRM_MOD	RW	加解密算法模式选择。 1: AES; 0: SM4。	0
7	RB_ECDC_WRSRAM_EN	RW	使能 SRAM 数据加解密。需要控制位 bit2/bit1 配合使用。 1: 开启; 0: 关闭。	0
[6:4]	RB_ECDC_CLKDIV_MASK	RW	加解密时钟分频系数。 计算: $EDclk=480M/ED_CLK_PRE$ 。 最小值 2, 写 1 等效关闭 ECDC 模块运算时钟。	10b
3	RB_ECDC_MODE_SEL	RW	加解密模式选择。 1: 解密模式; 0: 加密模式。	0
2	RB_ECDC_WRPERRI_EN	RW	使能数据从 SRAM 到外设进行加解密控制。 1: 加解密使能; 0: 不工作。	0
1	RB_ECDC_RDPERI_EN	RW	使能数据从外设到 SRAM 进行加解密控制。 1: 加解密使能; 0: 不工作。	0
0	RB_ECDC_KEYEX_EN	RW	密钥扩展功能使能控制位, 高电平脉冲启动。在配置了密钥长度后, 需要此位置 1, 并等待密钥扩展完成才能进行加解密过程。 注: 应用代码驱动需要将其置高再置低。	0

ECDC 中断使能寄存器 (R8_ECDC_INT_EN)

位	名称	访问	描述	复位值
[7:3]	Reserved	RO	保留。	0
2	RB_ECDC_IE_WRSRAM	WO	存储器到存储器进行加解密完成中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0
1	RB_ECDC_IE_SINGLE	RW	单次加解密完成中断使能。针对 128bit 寄存器	0

			转换完成。 1: 使能相应中断; 0: 禁止相应中断。	
0	RB_ECDC_IE_EKDONE	RW	密钥扩展完成中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0

ECDC 中断标志寄存器 (R8_ECDC_INT_FG)

位	名称	访问	描述	复位值
[7:3]	Reserved	R0	保留。	0
2	RB_ECDC_IF_WRSRAM	RW1	存储器到存储器加解密完成中断标志, 写 1 清 0。 1: 加解密完成事件; 0: 无事件。	0
1	RB_ECDC_IF_SINGLE	RW1	单次加解密完成中断标志, 写 1 清 0。 1: 单次加解密完成事件; 0: 无事件。	0
0	RB_ECDC_IF_EKDONE	RW1	密钥扩展完成中断标志, 写 1 清 0。 1: 密钥扩展完成事件; 0: 无事件。	0

用户密钥寄存器组 (R32_ECDC_KEY)

位	名称	访问	描述	复位值
[31:0]	R32_ECDC_KEY_255T224	RW	用户密钥 224-256 位。	X
[31:0]	R32_ECDC_KEY_223T192	RW	用户密钥 192-223 位。	X
[31:0]	R32_ECDC_KEY_191T160	RW	用户密钥 160-191 位。	X
[31:0]	R32_ECDC_KEY_159T128	RW	用户密钥 128-159 位。	X
[31:0]	R32_ECDC_KEY_127T96	RW	用户密钥 96-127 位。	X
[31:0]	R32_ECDC_KEY_95T64	RW	用户密钥 64-95 位。	X
[31:0]	R32_ECDC_KEY_63T32	RW	用户密钥 32-63 位。	X
[31:0]	R32_ECDC_KEY_31T0	RW	用户密钥 0-31 位。	X

CTR 模式计数值寄存器组 (R32_ECDC_IV)

位	名称	访问	描述	复位值
[31:0]	R32_ECDC_IV_127T96	RW	计数值 96-127 位。	X
[31:0]	R32_ECDC_IV_95T64	RW	计数值 64-95 位。	X
[31:0]	R32_ECDC_IV_63T32	RW	计数值 32-63 位。	X
[31:0]	R32_ECDC_IV_31T0	RW	计数值 0-31 位。	X

单次加解密原始数据寄存器组 (R32_ECDC_SGSD)

位	名称	访问	描述	复位值
[31:0]	R32_ECDC_SGSD_127T96	RW	原始数据 96-127 位。	0
[31:0]	R32_ECDC_SGSD_95T64	RW	原始数据 64-95 位。	0
[31:0]	R32_ECDC_SGSD_63T32	RW	原始数据 32-63 位。	0
[31:0]	R32_ECDC_SGSD_31T0	RW	原始数据 0-31 位。	0

注: 内部加解密固定 128bit 大小进行。当执行完 R32_ECDC_SGSD_31T0 寄存器的写操作, 硬件自动开始进行单次加解密转换。通过 RB_ECDC_IF_SINGLE 表示判断转换是否完成。

单次加解密结果寄存器组 (R32_ECDC_SGRT)

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

[31:0]	R32_ECDC_SGRT_127T96	RW	数据 96-127 位。	0
[31:0]	R32_ECDC_SGRT_95T64	RW	数据 64-95 位。	0
[31:0]	R32_ECDC_SGRT_63T32	RW	数据 32-63 位。	0
[31:0]	R32_ECDC_SGRT_31T0	RW	数据 0-31 位。	0

注：加解密结果数据固定大端方式存储，128bits 结构。

加解密 SRAM 区域起始地址 (R32_ECDC_SRAM_ADDR)

位	名称	访问	描述	复位值
[32:17]	Reserved	RO	保留。	0
[16:0]	ECDC_SRAM_ADDR	RW	加解密 SRAM 数据的起始地址，低 4 位固定为 0 (16 字节对齐)。	0

注：此 DMA 寻址在 RAMX 区域。

加解密 SRAM 大小 (R32_ECDC_SRAM_LEN)

位	名称	访问	描述	复位值
[31:13]	Reserved	RO	保留。	0
[12:0]	ECDC_SRAM_LEN	RW	加解密 SRAM 数据长度，单位为 128bit。当写入非零值时，将启动 SRAM 数据的加解密过程。	0

第 16 章 串并互转控制器及收发器 (SerDes)

系统内置了支持信号隔离和远距离传输的 SerDes 模块，支持 1.2Gbps 高速差分信号 (GXM/GXP 引脚)，可以通过光纤模块或网线中的一个差分对等传输媒体，进行远距离数据传输。

SerDes 模块外设基地址：0x4000B000

主要特征

- 可编程的数据收发速率，最高支持1.2Gbps
- 内置8b/10b编解码和CRC校验，支持序列号匹配
- 内置FIFO，支持收发双缓冲模式
- 支持DMA功能，访问地址支持字节对齐
- 提供多种传输中断标志和状态，及时向应用层反馈信息
- 差分收发，可直接驱动光纤模块
- 实测在600Mbps速率下通过一对超5类差分网线传输距离约70米
- 实测在600Mbps速率下通过一对6类差分网线传输距离约90米
- 实测在1.2Gbps速率下通过一对6类差分网线传输距离约25米
- 实测在360Mbps速率下通过一对6类差分网线传输距离约100米

具体应用请参考和调用提供的子程序库。

第 17 章 USB3.0 控制器及收发器 (USBSS)

USB3.0 控制器模块包含 USB 主机控制器和 USB 设备控制器功能, 搭配系统内置的物理收发器 PHY, 可实现 USB3.0 接口产品功能。支持 5Gbps 的 USB SuperSpeed 超高速信号, 硬件接口包括 2 对超高速差分信号线 (SSRXA/SSRXB 和 SSTXA/SSTXB, A/B 可连+/-或-/+)。

此控制器模块为应用代码提供了链接层寄存器访问接口, 用于管理设备的连接和断开、总线状态、电源模式。提供了主机 (HOST) 功能访问接口, 设备 (DEVICE) 功能访问接口, 用于实现 USB3.0 协议规范的各种数据传输及上层协议。

USBSS 模块外设基地址: 0x40008000

主要特征

- 支持USB3.0协议接口规范
- 支持USB Host主机功能和USB Device设备功能
- 支持OTG功能
- 支持驱动USB3.0 HUB
- 主机和设备均支持控制传输、批量传输、中断传输、实时/同步传输
- DMA方式直接访问各端点缓冲区的数据
- 电源管理, 支持U1/U2/U3低功耗状态
- 非0端点均支持最大1024字节的数据包, 支持突发模式

具体应用请参考和调用提供的子程序库。

第 18 章 千兆以太网控制器 (ETH-GMAC)

系统集成了千兆以太网控制器 (Gigabit Ethernet Transceiver)，其 Link 速率最高支持 1Gbps，充当数据链路层的角色。应用时，硬件接口通过 RGMII 或 RMII 信号连接外置 PHY (千兆/百兆/速度自适应)，并结合 TCP/IP 协议栈接口实现网络产品的开发。

模块设计了符合 IEEE 802.3 的 MAC 控制器，配合 128 位 DMA 完成 MAC 和 SRAM 数据的快速转移，并提供多种配置选项、硬件处理、状态呈现等功能。

ETH 模块外设基地址：0x4000C000

主要特征

- 符合 IEEE 802.3 协议规范及设计
- 提供 RGMII 和 RMII 接口，连接外置的以太网 PHY
- 通过外接的 PHY 接口，支持 10/100/1000Mbps 的数据传输速率
- 支持全双工操作
- 支持 SMI 接口对外置 PHY 进行配置和管理
- 硬件自动完成对 IPv4 和 IPv6 包完整性校验，上报应用软件
- 硬件自动完成对 IP/ICMP/UDP/TCP 包的校验和计算及帧长度填充
- 多种 MAC 地址过滤模式

具体应用请参考和调用提供的子程序库。

第 19 章 参数

19.1 绝对最大值

临界或者超过绝对最大值可能导致芯片工作不正常甚至损坏。

表 19-1 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	VDDLDO=VDDIO=3.3V	-20	85	°C
		V33USB=V33GX=3.3V			
		V12USB=V12CORE=1.2V			
TS	存储时的环境温度	-55	125	°C	
V12CORE	系统内核电压	-0.3	1.5	V	
VDDLDO	系统电源电压	-0.4	4.2	V	
VDDIO	GPIO 端口电压	-0.4	4.2	V	
V12USB	USB PHY 信号引脚电压	-0.3	1.5	V	
V33USB	USB PHY 电源电压	-0.4	4.2	V	
V33GX	SerDes PHY 电源电压	-0.4	4.2	V	

19.2 电气参数

测试条件：TA=25°C，VDDLDO=VDDIO=V33USB=V33GX=3.3V，V12CORE=V12USB=1.2V，F_{sys}=80MHz。

表 19-2 电气参数表

名称	参数说明	最小值	典型值	最大值	单位
V12CORE	系统内核电压	1.15	1.2	1.3	V
VDDLDO	内部 LDO 输入电源电压	2.3	3.3	3.6	V
VDDIO	GPIO 端口电压	2.3	3.3	3.6	V
V12USB	USB PHY 低压电源电压	1.15	1.2	1.3	V
V33USB	USB PHY 电源电压	3.15	3.3	3.5	V
V33GX	SerDes PHY 电源电压	3.0	3.3	3.6	V
I _{CC}	工作时的总电源电流，常规测试条件		37		mA
I _{SLP}	低功耗状态的电源电流 (I/O 引脚输出无负载或 输入上拉)	空闲模式 F _{sys} =80MHz (外设时钟都开启)		26	mA
		暂停模式		5	mA
		睡眠模式		1.3	mA
V _{IVR}	V33GX 端低压复位阈值电压	2.6	2.8	3.0	V

表 19-3 I/O 特性参数

名称	参数说明	最小值	典型值	最大值	单位
V _{IL}	输入低电平电压 (VDDIO=3.3V)			0.8	V
V _{IH}	输入高电平电压 (VDDIO=3.3V)	2.0			V
V _{t-}	施密特触发输入负向到阈值	0.8	1.1		V
V _{t+}	施密特触发输入正向到阈值		1.6	2.0	V
V _{OL}	输出低电平电压 (8/16mA 吸入电流)			0.4	V
V _{OH}	输出高电平电压 (8/16mA 输出电流)	2.4			V

R_{PU}	输入上拉电阻	40	75	190	$k\Omega$
R_{PD}	输入下拉电阻	30	75	190	$k\Omega$
I_{in}	输入漏电流		± 1	± 10	μA
C_{IN}	输入电容 (VDDIO=2.5V 或 3.3V)	1.5	2.5	3.5	pF

19.3 功能模块功耗

测试条件：TA=25℃，VDDLDO=VDDIO=V33USB=V33GX=3.3V，V12CORE=V12USB=1.2V。

表 19-4 各模块动态功耗

模块	F_{sys}					单位
	30M	60M	80M	96M	120M	
ETH	-	-	2			mA
USBHS 传输	-	28				mA
USBSS 传输	-	-	110			mA
DVP	12					mA
HSPI	2	5	6	6	7	mA
SerDes-1.2Gbps (RX/TX 同时打开)	60					mA
ECDC-240MHz	2.2	2.7	2.9	3.2	3.7	mA
ECDC-160MHz	0.6	1.1	1.4	1.6	2.2	mA
TMR+UART+SPI+PWMX	1.8	3.5	4.0	6.0	7.5	mA
Core+BUS8+DMA+SRAM	4.0	8.0	10.7	12.7	15.9	mA

19.4 时间参数

测试条件：TA=25℃，VDDLDO=VDDIO=V33USB=V33GX=3.3V，V12CORE=V12USB=1.2V。

表 19-5 时间参数表

名称	参数说明	最小值	典型值	最大值	单位	
T_{rst}	外部复位输入 RST#有效信号宽度	50	$2 * T_{sys}$	-	ns	
T_{mr}	上电复位后的复位延时+加载时间	15	27	35	ms	
T_{sro}	外部/软件复位输入后的复位延时+加载时间	3	4.4	6	ms	
T_{wak}	从低功耗状态退出的唤醒时间	空闲模式	0.3	0.5	1.5	μs
		暂停模式	20	27	40	μs
		睡眠模式		$T_{SUHSE}+30$		μs

注：上表中 T_{wak} 参数是基于 80MHz 主频时钟，降低主频将增加延时。

对于睡眠模式下低功耗，唤醒需要考虑晶体 HSE 起振时间，所以 T_{wak} 额外增加约 0.1~1ms（启动到可用 T_{SUHSE} ）。

第 20 章 封装

芯片封装

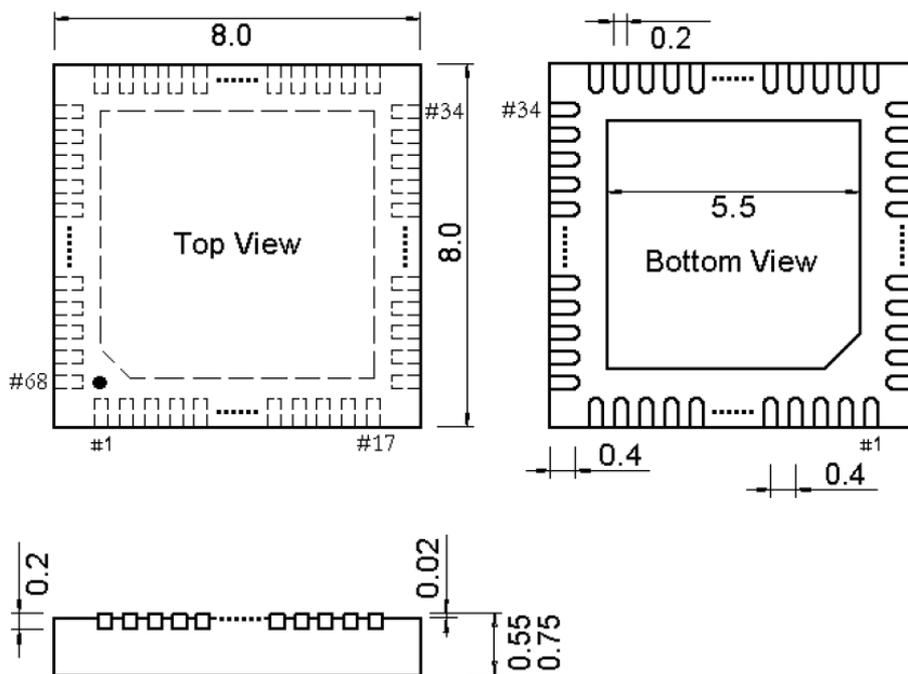
封装形式	塑体宽度	引脚间距		封装说明	订货型号
QFN68	8*8mm	0.4mm	15.75mil	方形无引线 68 脚	CH569W
QFN68	8*8mm	0.4mm	15.75mil	方形无引线 68 脚	CH565W
QFN40	5*5mm	0.4mm	15.75mil	方形无引线 40 脚	CH565M

说明:

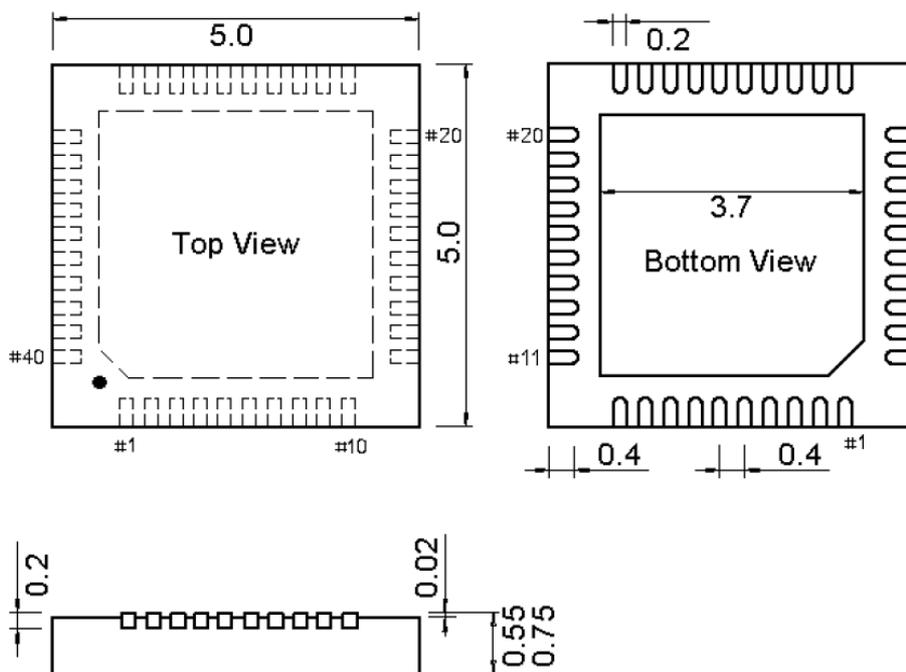
尺寸标注的单位是 mm (毫米)

引脚中心间距是标称值, 没有误差, 除此之外的尺寸误差不大于 ±0.5mm。

QFN68-8*8



QFN40-5*5



X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [32-bit Microcontrollers - MCU category](#):

Click to view products by [Qin Heng manufacturer](#):

Other Similar products are found below :

[MCF51AC256AVFUE](#) [MCF51AC256BCFUE](#) [MCF51AC256BVFUE](#) [MB91F464AAPMC-GSE2](#) [R5S726B0D216FP#V0](#) [MB91F248PFV-GE1](#) [MB91243PFV-GS-136E1](#) [SAK-TC1782F-320F180HR BA](#) [TC364DP64F300WAAKXUMA1](#) [R5F566NNDDFP#30](#)
[R5F566NNDDFC#30](#) [R5F566NNDDBD#20](#) [MC96F8216ADBN](#) [A96G181HDN](#) [A96G140KNN](#) [A96G174FDN](#) [A31G213CL2N](#)
[A96G148KNN](#) [A96G174AEN](#) [AC33M3064TLBN-01](#) [V3s](#) [T3](#) [A40i-H](#) [V526](#) [A83T](#) [R11](#) [V851s](#) [A133](#) [F1C100S](#) [T3L](#) [T507](#) [A33](#) [A63](#)
[T113-i](#) [H616](#) [V853](#) [V533](#) [R16-J](#) [V536-H](#) [A64-H](#) [V831](#) [V3LP](#) [F133-A](#) [R128-S2](#) [D1-H](#) [ADUCM360BCPZ128-TR](#) [AT32F435VMT7](#)
[AT32F437ZMT7](#) [AT32F421G8U7](#) [AT32F421K8T7](#)