



GOAL Hi3510 V100 通信媒体处理器芯片

用户指南

文档版本	04
发布日期	2007-04-20
BOM编码	N/A

深圳市海思半导体有限公司为客户提供全方位的技术支持，用户可与就近的海思办事处联系，也可直接与公司总部联系。

深圳市海思半导体有限公司

地址： 深圳市龙岗区坂田华为基地华为电气生产中心 邮编： 518129

网址： <http://www.hisilicon.com>

客户服务电话： 0755-28788858

客户服务传真： 0755-28788838

客户服务邮箱： support@hisilicon.com.

版权所有 © 深圳市海思半导体有限公司 2007。 保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



、Hisilicon、海思，均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。



目 录

前 言.....	1
1 介绍.....	1-1
1.1 概述.....	1-2
1.2 主要特点.....	1-3
1.2.1 内嵌 RISC 内核.....	1-3
1.2.2 内嵌 DSP 内核.....	1-3
1.2.3 视频编解码.....	1-3
1.2.4 视频处理性能.....	1-3
1.2.5 图形处理.....	1-3
1.2.6 音频编解码.....	1-4
1.2.7 安全加密引擎.....	1-4
1.2.8 以太网交换接口.....	1-4
1.2.9 视频接口.....	1-4
1.2.10 音频接口.....	1-4
1.2.11 外设接口.....	1-4
1.2.12 外部存储器接口.....	1-5
1.2.13 嵌入式操作系统.....	1-5
1.2.14 芯片物理规格.....	1-5
1.3 应用领域.....	1-5
1.4 典型应用.....	1-5
2 处理器子系统.....	2-1
2.1 概述.....	2-2
2.2 ARM926EJ-S 主要特点.....	2-2
2.3 DSP 子系统主要特点.....	2-2
2.4 Hi3510 地址映射关系.....	2-4
3 多端口静态和动态存储控制器.....	3-1
3.1 概述.....	3-2
3.2 功能描述.....	3-2
3.3 信号描述.....	3-2



3.4 寄存器概览.....	3-4
3.5 寄存器描述.....	3-8
3.5.1 MEMC_CONTROL.....	3-8
3.5.2 MEMC_STATUS	3-8
3.5.3 MEMC_CONFIG.....	3-9
3.5.4 MEMC_DYNAMICCONTROL.....	3-9
3.5.5 MEMC_DYNAMICREFRESH.....	3-11
3.5.6 MEMC_DYNAMICREADCONFIG.....	3-11
3.5.7 MEMC_DYNAMICTRP	3-12
3.5.8 MEMC_DYNAMICTRAS	3-12
3.5.9 MEMC_DYNAMICTSREX.....	3-12
3.5.10 MEMC_DYNAMICCTWR.....	3-13
3.5.11 MEMC_DYNAMICTRC	3-13
3.5.12 MEMC_DYNAMICTRFC.....	3-13
3.5.13 MEMC_DYNAMICCTXSR	3-14
3.5.14 MEMC_DYNAMICTRRD.....	3-14
3.5.15 MEMC_DYNAMICCTMRD	3-14
3.5.16 MEMC_DYNAMICCTCDLR.....	3-15
3.5.17 MEMC_STATICEXTENDEDWAIT	3-15
3.5.18 MEMC_DYNAMICCONFIG0.....	3-16
3.5.19 MEMC_DYNAMICRASCAS0	3-19
3.5.20 MEMC_STIATICCONFIG0~3	3-20
3.5.21 MEMC_STIATICWAITWEN0~3.....	3-21
3.5.22 MEMC_STIATICWAITOPEN0~3.....	3-21
3.5.23 MEMC_STIATICWAITRD0~3	3-21
3.5.24 MEMC_STIATICWAITPAGE0~3.....	3-22
3.5.25 MEMC_STIATICWAITWR0~3.....	3-22
3.5.26 MEMC_STIATICWAITTURN0~3	3-23
3.5.27 MEMC_AHBCONTROL0~4.....	3-23
3.5.28 MEMC_AHBSTATUS0~4	3-23
3.5.29 MEMC_AHBTIMEOUT0~4	3-24
4 多端口 DDR SDRAM 存储控制器.....	4-1
4.1 概述.....	4-2
4.2 特点.....	4-2
4.3 信号描述.....	4-2
4.4 寄存器概览.....	4-3
4.5 寄存器描述.....	4-5
4.5.1 DDRC_CONTROL.....	4-5
4.5.2 DDRC_STATUS	4-6
4.5.3 DDRC_CONFIG.....	4-6



4.5.4	DDRC_DYNAMICCONTROL	4-7
4.5.5	DDRC_DYNAMICREFRESH	4-7
4.5.6	DDRC_DYNAMICREADCONFIG	4-8
4.5.7	DDRC_DYNAMICTRP	4-8
4.5.8	DDRC_DYNAMICTRAS	4-9
4.5.9	DDRC_DYNAMICTSREX	4-9
4.5.10	DDRC_DYNAMICTWR	4-9
4.5.11	DDRC_DYNAMICTRC	4-10
4.5.12	DDRC_DYNAMICTRFC	4-10
4.5.13	DDRC_DYNAMICTXSR	4-11
4.5.14	DDRC_DYNAMICTRRD	4-11
4.5.15	DDRC_DYNAMICTMRD	4-11
4.5.16	DDRC_DYNAMICTCDLR	4-12
4.5.17	DDRC_DYNAMICCONFIG0	4-12
4.5.18	DDRC_DYNAMICRASCAS0	4-13
4.5.19	DDRC_AHBCONTROL0~4	4-14
4.5.20	DDRC_AHBSTATUS0~4	4-14
4.5.21	DDRC_AHBTIMEOUT0~4	4-15
5	中断控制器 (VIC)	5-1
5.1	概述	5-2
5.2	特点	5-2
5.3	信号描述	5-2
5.4	工作方式	5-2
5.5	寄存器概览	5-3
5.6	寄存器描述	5-5
5.6.1	VIC_IRQSTATUS	5-5
5.6.2	VIC_FIQSTATUS	5-5
5.6.3	VIC_RAWINTR	5-6
5.6.4	VIC_INTSELECT	5-6
5.6.5	VIC_INTENABLE	5-6
5.6.6	VIC_INTENCLEAR	5-7
5.6.7	VIC_SOFTINT	5-7
5.6.8	VIC_SOFTINTCLEAR	5-8
5.6.9	VIC_PROTECTION	5-8
5.6.10	VIC_VECTADDR	5-9
5.6.11	VIC_DEFVECTADDR	5-9
5.6.12	VIC_VECTADDR0~15	5-9
5.6.13	VIC_VECTCNTL0~15	5-10
6	时钟、复位和系统控制器	6-1
6.1	概述	6-2



6.2 功能描述	6-2
6.2.1 时钟	6-2
6.2.2 复位	6-4
6.2.3 系统控制器	6-6
6.3 寄存器概览	6-10
6.4 寄存器描述	6-11
6.4.1 SC_CTRL	6-11
6.4.2 SC_SYSSSTAT	6-13
6.4.3 SC_ITMCTRL	6-14
6.4.4 SC_IMSTAT	6-14
6.4.5 SC_XTALCTRL	6-15
6.4.6 SC_PLLCTRL	6-16
6.4.7 SC_PLLFCTRL	6-16
6.4.8 SC_PERCTRL0	6-18
6.4.9 SC_PERCTRL1	6-19
6.4.10 SC_PEREN	6-21
6.4.11 SC_PERDIS	6-22
6.4.12 SC_PERCLKEN	6-23
6.4.13 SC_PERSTAT	6-23
6.4.14 Version_ID0	6-24
6.4.15 Version_ID1	6-24
6.4.16 Version_ID2	6-24
6.4.17 Version_ID3	6-25
7 直接存储访问控制器 (DMAC)	7-1
7.1 概述	7-2
7.2 特点	7-2
7.3 功能描述	7-3
7.4 接口信号描述	7-4
7.5 工作方式	7-5
7.6 寄存器概览	7-6
7.7 寄存器描述	7-8
7.7.1 DMAC_INTSTATUS0	7-8
7.7.2 DMAC_INTTCSTATUS0	7-9
7.7.3 DMAC_INTTCCLEAR	7-9
7.7.4 DMAC_INTERRORSTATUS0	7-10
7.7.5 DMAC_INTERRCLR	7-10
7.7.6 DMAC_RAWINTTCSTATUS	7-10
7.7.7 DMAC_RAWINTERRORSTATUS	7-11
7.7.8 DMAC_ENBLDCHNS	7-11
7.7.9 DMAC_SOFTBREQ	7-12



7.7.10 DMAC_SOFTSREQ.....	7-12
7.7.11 DMAC_SOFTLBREQ.....	7-13
7.7.12 DMAC_SOFTLSREQ.....	7-13
7.7.13 DMAC_CONFIGURATION.....	7-13
7.7.14 DMAC_SYNC.....	7-14
7.7.15 DMAC_INTSTATUS1.....	7-15
7.7.16 DMAC_INTTCSTATUS1.....	7-15
7.7.17 DMAC_INTERRORSTATUS1.....	7-15
7.7.18 DMAC_CxSRCADDR.....	7-16
7.7.19 DMAC_CxDESTADDR.....	7-17
7.7.20 DMAC_CxLLI.....	7-17
7.7.21 DMAC_CxCONTROL.....	7-18
7.7.22 DMAC_CxCONFIGURATION.....	7-22
8 TIMER、WatchDog 和 RTC.....	8-1
8.1 TIMER.....	8-2
8.1.1 概述.....	8-2
8.1.2 特点.....	8-2
8.1.3 寄存器概览.....	8-2
8.1.4 寄存器描述.....	8-3
8.2 WatchDog.....	8-8
8.2.1 概述.....	8-8
8.2.2 特点.....	8-8
8.2.3 寄存器概览.....	8-8
8.2.4 寄存器描述.....	8-9
8.3 RTC.....	8-11
8.3.1 概述.....	8-11
8.3.2 特点.....	8-12
8.3.3 寄存器概览.....	8-12
8.3.4 寄存器描述.....	8-12
9 视频编解码单元.....	9-1
9.1 概述.....	9-2
9.2 视频编解码协处理器.....	9-2
9.3 视频编解码器.....	9-3
9.3.1 视频编码器.....	9-3
9.3.2 视频解码器.....	9-3
10 2D 图形加速引擎(TDE).....	10-1
10.1 概述.....	10-2
10.2 功能描述.....	10-2
10.2.1 特点.....	10-2



10.2.2 支持数据格式.....	10-3
10.2.3 术语描述.....	10-3
10.2.4 支持功能.....	10-4
10.3 工作方式.....	10-5
10.3.1 中断.....	10-6
10.3.2 复位.....	10-6
10.3.3 配置.....	10-6
10.4 寄存器概览.....	10-8
10.5 寄存器描述.....	10-9
10.5.1 控制寄存器 (TDE_CTRL)	10-9
10.5.2 位图大小寄存器 (TDE_SIZE)	10-11
10.5.3 源位图起始地址 (TDE_SOURCE_ADDR)	10-11
10.5.4 目标位图起始地址 (TDE_DET_ADDR)	10-12
10.5.5 输出位图起始地址 (TDE_GOUT_ADDR)	10-12
10.5.6 源位图行间距 (TDE_SOURCE_STRIDE)	10-12
10.5.7 输出位图行间距 (TDE_GOUT_STRIDE)	10-13
10.5.8 颜色填充值 (TDE_COLOR_FILLED)	10-13
10.5.9 color space 最大值 (TDE_COLOR_MAX)	10-14
10.5.10 color space 最小值 (TDE_COLOR_MIN)	10-14
10.5.11 TDE 状态寄存器 (TDE_STATUS)	10-15
11 视频输入单元 (VIU)	11-1
11.1 概述.....	11-2
11.2 功能描述.....	11-2
11.3 信号描述.....	11-3
11.4 工作方式.....	11-3
11.4.1 BT.656/601 YUV4:2:2	11-3
11.4.2 数字 camera 接口.....	11-9
11.4.3 Raw data 接口	11-10
11.4.4 Bayer RGB 输入时序	11-10
11.5 图像存储方式.....	11-11
11.5.1 YC planar 存储.....	11-11
11.5.2 Y/ CB/CR (R/G/B) planar 存储	11-12
11.5.3 Packets 存储.....	11-13
11.5.4 Raw data 存储	11-13
11.6 寄存器概览.....	11-14
11.7 寄存器描述.....	11-15
11.7.1 配置寄存器.....	11-15
11.7.2 时序寄存器.....	11-18
11.7.3 图像获取窗口寄存器.....	11-19



11.7.4 图像存储行间距寄存器	11-21
11.7.5 地址属性寄存器	11-22
11.7.6 控制寄存器	11-23
11.7.7 中断 mask 寄存器	11-25
11.7.8 中断状态寄存器	11-26
11.7.9 状态寄存器	11-26
11.7.10 Y 分量数据存储大小寄存器	11-27
11.7.11 U 分量数据存储大小寄存器	11-27
11.7.12 V 分量数据存储大小寄存器	11-28
11.7.13 VBI 数据位置寄存器	11-28
11.7.14 VBI 数据获取存储寄存器	11-31
11.7.15 图像块屏蔽设置寄存器	11-32
11.7.16 亮度统计寄存器	11-33
12 视频输出单元 (VOU)	12-1
12.1 概述	12-2
12.2 特点	12-2
12.3 信号描述	12-3
12.4 接口协议	12-4
12.5 工作方式	12-4
12.6 寄存器概览	12-5
12.7 寄存器描述	12-6
12.7.1 控制寄存器和硬件鼠标	12-6
12.7.2 中断使能寄存器	12-11
12.7.3 中断状态寄存器	12-12
12.7.4 状态寄存器	12-12
12.7.5 垂直同步寄存器 1	12-13
12.7.6 垂直同步寄存器 2	12-14
12.7.7 水平同步寄存器	12-14
12.7.8 图像寄存器	12-15
12.7.9 叠加图像 1 寄存器	12-16
12.7.10 叠加图像 2 寄存器	12-16
12.7.11 图像偏移寄存器	12-17
12.7.12 背景颜色寄存器	12-17
12.7.13 clip 值寄存器	12-17
12.7.14 Key 的 mask 值寄存器	12-18
12.7.15 叠加图像 1 key 值寄存器	12-18
12.7.16 叠加图像 2 key 值寄存器	12-19
12.7.17 硬件鼠标颜色 0 寄存器	12-19
12.7.18 硬件鼠标颜色 1 寄存器	12-19



12.7.19 硬件鼠标颜色 2 寄存器	12-20
12.7.20 硬件鼠标颜色 3 寄存器	12-20
12.7.21 叠加图像 1 起始位置寄存器	12-20
12.7.22 叠加图像 2 起始位置寄存器	12-21
12.7.23 硬件鼠标起始位置寄存器	12-21
12.7.24 叠加图像 1 alpha 值寄存器	12-22
12.7.25 叠加图像 2 alpha 值寄存器	12-22
12.7.26 视频图像亮度分量地址寄存器	12-23
12.7.27 视频图像色度分量地址寄存器	12-23
12.7.28 叠加图像 1 地址寄存器	12-23
12.7.29 叠加图像 2 地址寄存器	12-24
12.7.30 硬件鼠标地址寄存器	12-24
12.7.31 主图像亮度和色度行偏移量寄存器	12-24
12.7.32 叠加图像行偏移量寄存器	12-25
13 串行输入输出接口 (SIO)	13-1
13.1 概述	13-2
13.2 特点	13-2
13.2.1 PCM 接口特点	13-2
13.2.2 I ² S 接口特点	13-2
13.3 信号描述	13-3
13.4 工作方式	13-3
13.4.1 时序	13-3
13.4.2 应用说明	13-4
13.5 寄存器概览	13-5
13.6 寄存器描述	13-6
13.6.1 SIO_MODE	13-6
13.6.2 SIO_INTR_STATUS	13-6
13.6.3 SIO_INTR_CLR	13-7
13.6.4 发送数据寄存器	13-7
13.6.5 接收数据寄存器	13-8
13.6.6 I2S_CT_SET	13-8
13.6.7 SIO_ICD	13-10
13.6.8 SIO_RX_STA	13-10
13.6.9 SIO_TX_STA	13-10
13.6.10 SIO_PCM_CT_SET/ SIO_PCM_CT_CLR	13-11
14 通用目的输入输出接口 (GPIO)	14-1
14.1 概述	14-2
14.2 特点	14-2
14.3 信号描述	14-2



14.4 工作方式.....	14-3
14.4.1 复用说明.....	14-3
14.4.2 使用指南.....	14-7
14.5 寄存器概览.....	14-9
14.6 寄存器描述.....	14-10
14.6.1 GPIO_DATA.....	14-10
14.6.2 GPIO_DIR.....	14-10
14.6.3 GPIO_IS.....	14-10
14.6.4 GPIO_IBE.....	14-11
14.6.5 GPIO_IEV.....	14-11
14.6.6 GPIO_IE.....	14-12
14.6.7 GPIO_RIS.....	14-12
14.6.8 GPIO_MIS.....	14-12
14.6.9 GPIO_IC.....	14-13
14.6.10 GPIO_AFSEL.....	14-13
14.7 GPIO 配置示例.....	14-13
14.7.1 注意事项.....	14-13
14.7.2 配置过程.....	14-14
14.7.3 GPIO 中断配置树形图.....	14-14
15 通用异步收发器 (UART)	15-1
15.1 概述.....	15-2
15.2 特点.....	15-2
15.3 信号描述.....	15-2
15.4 工作方式.....	15-3
15.5 寄存器概览.....	15-3
15.6 寄存器描述.....	15-4
15.6.1 UART_DR.....	15-4
15.6.2 UART_RSR/UART_ECR.....	15-4
15.6.3 UART_FR.....	15-5
15.6.4 UART_LPR.....	15-6
15.6.5 UART_IBRD.....	15-7
15.6.6 UART_FBRD.....	15-7
15.6.7 UART_LCR_H.....	15-8
15.6.8 UART_CR.....	15-9
15.6.9 UART_IFLS.....	15-10
15.6.10 UART_IMSC.....	15-11
15.6.11 UART_RIS.....	15-12
15.6.12 UART_MIS.....	15-13
15.6.13 UART_ICR.....	15-14
15.6.14 UART_DMACR.....	15-15



16 SSP 接口	16-1
16.1 概述	16-2
16.2 特点	16-2
16.3 信号描述	16-3
16.4 工作方式	16-3
16.5 寄存器概览	16-5
16.6 寄存器描述	16-5
16.6.1 控制寄存器 0 (SSP_CR0)	16-5
16.6.2 控制寄存器 1 (SSP_CR1)	16-6
16.6.3 接收/发送 FIFO (SSP_DR)	16-7
16.6.4 状态寄存器 (SSP_SR)	16-7
16.6.5 时钟生成寄存器 (SSP_CPSR)	16-8
16.6.6 中断屏蔽设置/清除寄存器 (SSP_IMSC)	16-9
16.6.7 原始中断状态寄存器 (SSP_RIS)	16-9
16.6.8 屏蔽后中断状态寄存器 (SSP_MIS)	16-10
16.6.9 中断清除寄存器 (SSP_ICR)	16-10
16.6.10 DMA 控制寄存器 (SSP_DMACR)	16-11
17 I²C 接口	17-1
17.1 概述	17-2
17.2 特点	17-2
17.3 信号描述	17-2
17.4 工作方式	17-2
17.5 寄存器概览	17-3
17.6 寄存器描述	17-5
17.6.1 IIC_CON	17-5
17.6.2 IIC_TAR	17-6
17.6.3 IIC_SAR	17-6
17.6.4 IIC_HS_MADDR	17-7
17.6.5 IIC_DATA_CMD	17-7
17.6.6 IIC_SS_SCL_HCNT	17-8
17.6.7 IIC_SS_SCL_LCNT	17-8
17.6.8 IIC_FS_SCL_HCNT	17-8
17.6.9 IIC_FS_SCL_LCNT	17-8
17.6.10 IIC_HS_SCL_HCNT	17-9
17.6.11 IIC_HS_SCL_LCNT	17-9
17.6.12 IIC_INTR_STAT	17-9
17.6.13 IIC_INTR_MASK	17-10
17.6.14 IIC_RAW_INTR_STAT	17-11
17.6.15 IIC_RX_TL	17-13
17.6.16 IIC_TX_TL	17-13



17.6.17 IIC_CLR_INTR	17-13
17.6.18 IIC_CLR_RX_UNDER.....	17-14
17.6.19 IIC_CLR_RX_OVER	17-14
17.6.20 IIC_CLR_TX_OVER.....	17-14
17.6.21 IIC_CLR_RD_REQ	17-15
17.6.22 IIC_CLR_TX_ABRT	17-15
17.6.23 IIC_CLR_RX_DONE	17-15
17.6.24 IIC_CLR_ACTIVITY	17-16
17.6.25 IIC_CLR_STOP_DET	17-16
17.6.26 IIC_CLR_START_DET	17-16
17.6.27 IIC_CLR_GEN_CALL	17-16
17.6.28 IIC_ENABLE.....	17-17
17.6.29 IIC_STATUS	17-17
17.6.30 IIC_TXFLR.....	17-18
17.6.31 IIC_RXFLR.....	17-18
17.6.32 IIC_TX_ABRT_SOURCE	17-19
17.6.33 IIC_DMA_CR.....	17-20
17.6.34 IIC_DMA_TDLR.....	17-20
17.6.35 IIC_DMA_RDLR.....	17-20
17.6.36 IIC_COMP_PARAM_1	17-21
18 以太网交换单元 (SF)	18-1
18.1 概述.....	18-2
18.2 特点.....	18-2
18.3 信号描述.....	18-4
18.4 工作方式.....	18-6
18.4.1 接口时序.....	18-6
18.4.2 SF 初始化配置	18-11
18.4.3 SF 中断管理	18-15
18.4.4 CPU 端口收发帧	18-15
18.5 寄存器概览.....	18-18
18.6 寄存器描述.....	18-19
18.6.1 MAC 寄存器组.....	18-19
18.6.2 统计计数寄存器组.....	18-22
18.6.3 MDIO 寄存器组	18-35
18.6.4 IRF 和 ERF 寄存器组	18-39
18.6.5 全局寄存器组.....	18-41
18.6.6 IQM 管理寄存器组	18-49
18.6.7 BM 管理寄存器组.....	18-55
18.7 外部端口 RMON/SNMP 统计计数结果寄存器	18-65
18.7.1 ANYPORT 帧配置表.....	18-73



18.7.2 ANYTYPE 帧配置表.....	18-74
18.7.3 VLAN 表.....	18-75
19 加密单元.....	19-1
19.1 DES 加密单元	19-2
19.1.1 概述.....	19-2
19.1.2 特点.....	19-2
19.1.3 工作方式.....	19-2
19.1.4 寄存器概览.....	19-3
19.1.5 寄存器描述.....	19-4
19.2 AES 加密单元.....	19-12
19.2.1 概述.....	19-12
19.2.2 特点.....	19-12
19.2.3 工作方式.....	19-12
19.2.4 寄存器概览.....	19-14
19.2.5 寄存器描述.....	19-15
20 模式配置与接口调试	20-1
20.1 概述.....	20-2
20.2 信号描述.....	20-2
20.3 工作方式.....	20-3
20.3.1 模式配置.....	20-3
20.3.2 调试接口.....	20-4
20.3.3 调试模式.....	20-4
21 时序和参数.....	21-1
21.1 时序图例.....	21-2
21.2 USB 时钟时序和参数	21-2
21.3 存储接口时序参数.....	21-3
21.3.1 SDRAM 接口时序参数.....	21-3
21.3.2 DDR 接口时序参数	21-3
21.4 串行口时序参数	21-5
21.4.1 I ² C 时序参数.....	21-5
21.4.2 SSP 时序参数	21-6
21.5 ETM 时序参数.....	21-6
21.6 SIO 接口时序参数.....	21-7
21.7 VI 视频输入接口时序参数.....	21-8
21.8 VO 视频输出接口时序参数	21-9
21.9 SF 以太网交换 RMII 接口时序 (50MHz) 参数.....	21-10
22 电性能参数.....	22-1
22.1 DC 参数	22-2



22.2 极限参数.....	22-3
22.3 推荐工作条件.....	22-3
23 管脚描述.....	23-1
23.1 接口说明.....	23-2
23.2 接口信号.....	23-2
23.3 复用信号.....	23-18
23.4 电源、地和 NC 管脚.....	23-23
24 封装、管脚分布.....	24-1
24.1 封装.....	24-2
24.2 管脚分布.....	24-4
A 管脚功能速查表.....	A-1
B 术语.....	B-1
C 缩略语.....	C-1



插图目录

图 1-1 Hi3510 内部逻辑框图.....	1-2
图 1-2 Hi3510 宽带 IP 可视电话典型应用	1-6
图 1-3 Hi3510 IP Camera 典型应用	1-6
图 2-1 Remap 前 Hi3510 地址映射关系图.....	2-5
图 2-2 Remap 后 Hi3510 地址映射关系图.....	2-6
图 3-1 MEMC 功能框图	3-2
图 4-1 DDRC 功能框图.....	4-2
图 6-1 复位信号结构图.....	6-5
图 6-2 各种状态的迁移图.....	6-7
图 7-1 DMAC 功能框图.....	7-3
图 7-2 DMAC 的请求线与其他外设的对应关系.....	7-4
图 8-1 TIMER 功能框图	8-2
图 10-1 TDE 功能框图.....	10-2
图 10-2 TDE 配置举例	10-7
图 10-3 TDE 位图存储示意图	10-8
图 11-1 VIU 功能框图.....	11-2
图 11-2 像素输入时序.....	11-4
图 11-3 模拟全电视信号对于的数字行采样时序	11-4
图 11-4 ITU-R BT. 601 行时序图	11-5
图 11-5 NTSC 制式垂直同步时序图.....	11-7
图 11-6 PAL 制式垂直同步时序图.....	11-7
图 11-7 软件配置的水平时序图	11-9
图 11-8 软件配置的垂直时序图	11-9
图 11-9 数字 camera 支持的水平、垂直时序图.....	11-10
图 11-10 bayer 数据垂直时序	11-10



图 11-11 bayer 数据奇数行水平时序	11-11
图 11-12 bayer 数据偶数行水平时序	11-11
图 11-13 YUV4:2:2 的存储方式	11-12
图 11-14 big endian 和 little endian 图像存储方式	11-12
图 11-15 Y/CB/CR 或 R/G/B 图像存储 big endian&little endian 方式	11-13
图 11-16 图像存储 packets 方式	11-13
图 11-17 raw data 8bit 存储方式	11-13
图 11-18 raw data 9/10bit 存储方式	11-13
图 11-19 图像获取参数示意图	11-20
图 11-20 VIU 的硬件工作流程	11-24
图 11-21 VBI 数据位置示意图	11-29
图 12-1 VOU 功能框图	12-2
图 13-1 PCM 接口时序（上升沿发送）	13-3
图 13-2 PCM 接口时序（下降沿发送）	13-4
图 13-3 I2S 接口时序	13-4
图 13-4 音频接口 DSP MODE 时序	13-4
图 14-1 键盘扫描阵列示意图	14-8
图 14-2 GPIO 中断配置树形图	14-15
图 15-1 UART 帧格式	15-3
图 16-1 SSP 功能框图	16-2
图 16-2 TI Synchronous Serial Frame Protocol（Single Transfers）接口时序	16-3
图 16-3 TI Synchronous Serial Frame Protocol（Multiple Transfers）接口时序	16-3
图 16-4 Motorola SPI Frame Protocol（Single Transfers）接口时序	16-4
图 16-5 Motorola SPI Frame Protocol（Multiple Transfers）接口时序	16-4
图 16-6 National Semiconductor MicroWire Protocol（Single Transfers）接口时序	16-4
图 16-7 National Semiconductor MicroWire Protocol（Multiple Transfers）接口时序	16-4
图 17-1 I ² C 数据传输格式	17-3
图 18-1 SF 功能框图	18-2
图 18-2 MII 接口接收时序图	18-6
图 18-3 MII 接口发送时序图	18-7
图 18-4 RMII 接口时序图	18-7
图 18-5 100Mbit/s MII 接口接收时序图	18-8



图 18-6 100Mbit/s MII 接口发送时序图	18-8
图 18-7 10Mbit/s MII 接口接收时序图	18-9
图 18-8 10Mbit/s MII 接口发送时序图	18-9
图 18-9 100Mbit/s RMII 接口接收时序图.....	18-9
图 18-10 100Mbit/s RMII 接口发送时序图.....	18-10
图 18-11 MDIO 读时序图	18-10
图 18-12 MDIO 写时序图	18-10
图 18-13 MDIO 接收时序图	18-11
图 18-14 MDIO 发送时序图	18-11
图 19-1 DES 操作流程图中.....	19-3
图 19-2 AES 加密单元操作流程图中.....	19-13
图 20-1 采用 ETM 进行调试的系统示例图.....	20-5
图 20-2 单独对 DSP 程序进行调试的系统示例图.....	20-6
图 20-3 Dual-Core 调试系统示例图	20-7
图 21-1 时序图例.....	21-2
图 21-2 USB 时钟时序图	21-2
图 21-3 SDRAM 接口时序.....	21-3
图 21-4 DDR 接口时序	21-4
图 21-5 DDR 接口输出时序	21-4
图 21-6 DDR 接口输入时序	21-4
图 21-7 I ² C 传输时序	21-5
图 21-8 SSP 主模式时序	21-6
图 21-9 ETM 时钟时序图	21-6
图 21-10 ETM 接口时序图	21-7
图 21-11 SIO 接口时序	21-8
图 21-12 VI 视频输入接口时序.....	21-9
图 21-13 VO 视频输出接口时序	21-9
图 21-14 50MHz 时 RMII 接口时序.....	21-10
图 24-1 芯片尺寸视图（俯视图）	24-2
图 24-2 芯片尺寸视图（仰视图）	24-2
图 24-3 芯片尺寸视图（侧视图）	24-3
图 24-4 Detail A 的放大图	24-3



图 24-5 Detail B 的放大图	24-3
图 24-6 Hi3510 管脚分布图（俯视图）	24-6



表格目录

表 2-1 DSP 子系统中断源分配表.....	2-3
表 3-1 多端口静态存储控制器接口信号描述.....	3-3
表 3-2 多端口动态存储控制器接口信号描述.....	3-3
表 3-3 MEMC 寄存器概览（基址是 0x1011_0000）.....	3-4
表 3-4 16bit 外部总线地址映射（Row, Bank, Column）.....	3-16
表 3-5 16bit 外部总线, Low-power SDRAM 地址映射（Bank, Row, Column）.....	3-17
表 3-6 32bit 外部总线地址映射（Row, Bank, Column）.....	3-18
表 3-7 32bit 外部总线 Low-power SDRAM 地址映射（Bank, Row, Column）.....	3-18
表 4-1 多端口 DDR SDRAM 存储控制器接口信号描述.....	4-2
表 4-2 DDRC 寄存器概览（基址是 0x1015_0000）.....	4-3
表 4-3 16bit 外部总线地址映射（Row, Bank, Column）.....	4-13
表 5-1 VIC 接口信号描述.....	5-2
表 5-2 中断请求分配.....	5-3
表 5-3 VIC 寄存器概览（基址是 0x1014_0000）.....	5-3
表 6-1 4 个外部输入的时钟比较.....	6-2
表 6-2 时钟接口信号描述.....	6-3
表 6-3 复位单元接口信号描述.....	6-5
表 6-4 系统控制器状态和时钟切换对应关系表.....	6-8
表 6-5 系统控制器寄存器概览（基址是：0x101E_0000）.....	6-11
表 7-1 DMA 接口信号描述.....	7-5
表 7-2 DMAC 硬件请求线和相应设备的对应关系.....	7-5
表 7-3 DMAC 寄存器概览（基址是 0x1013_0000）.....	7-6
表 7-4 DBSize、SBSize 的值和对应 burst 长度的关系说明.....	7-20
表 7-5 DWidth 和 SWidth 的值和对应传输位宽的关系说明.....	7-21
表 7-6 DMAC_CxCONTROL 寄存器 Prot 段属性及定义.....	7-22



表 7-7 流控及传输类型说明.....	7-25
表 8-1 TIMER12 寄存器概览（基址是 0x101E_2000）	8-3
表 8-2 TIMER34 寄存器概览（基址是 0x101E_3000）	8-3
表 8-3 WatchDog 寄存器概览（基址是 0x101E_1000）	8-9
表 8-4 RTC 寄存器概览（基址是 0x101E_8000）	8-12
表 10-1 TDE 寄存器概览（基址是 0x9001_0000）	10-8
表 10-2 CMD 寄存器详细描述.....	10-10
表 10-3 ROP2 操作符详细描述.....	10-10
表 11-1 视频输入接口（VIU）信号.....	11-3
表 11-2 ITU-R BT.656 YUV 4:2:2 行数据格式.....	11-5
表 11-3 SAV/EAV Format	11-5
表 11-4 有效 SAV/EAV 值.....	11-5
表 11-5 PAL 和 NTSC 制式 TV 图像帧对比	11-6
表 11-6 ITU-R BT.601 一帧有效行数据.....	11-8
表 11-7 ITU-R BT.656 帧时序	11-8
表 11-8 VIU 寄存器概览（基址是 0x9000_0000）	11-14
表 12-1 VOU 视频输出接口信号	12-3
表 12-2 SIO 与 LCD 复用关系对应表	12-4
表 12-3 VOU 寄存器概览（基址是 0x1012_0000）	12-5
表 12-4 控制寄存器.....	12-7
表 12-5 main_mode 的描述.....	12-9
表 12-6 32×32×2bpp 双色和透明模式列表	12-10
表 12-7 32×32×2bpp 四色模式列表	12-10
表 12-8 2×32×2bpp 三色和透明模式列表	12-11
表 13-1 SIO 接口信号描述	13-3
表 13-2 SIO0 寄存器概览（基址是 0x8008_0000）	13-5
表 13-3 SIO1 寄存器概览（基址是 0x9002_0000）	13-5
表 13-4 配置寄存器表.....	13-8
表 14-1 GPIO 接口信号描述	14-2
表 14-2 GPIO 复用描述	14-3
表 14-3 GPIO 复用对照表	14-4
表 14-4 8 组 GPIO 寄存器的基址.....	14-9



表 14-5 GPIO 寄存器一览表	14-9
表 15-1 UART 接口信号描述	15-2
表 15-2 UART 寄存器概览	15-3
表 15-3 典型的波特率及误差对应关系表	15-7
表 16-1 SSP 接口信号描述	16-3
表 16-2 SSP 寄存器概览（基址是 0x101F_4000）	16-5
表 17-1 I ² C 接口信号描述	17-2
表 17-2 I2C 寄存器概览（基址是 0x101F_6000）	17-3
表 18-1 以太网交换单元接口信号描述	18-4
表 18-2 MII 接口的时序参数	18-7
表 18-3 RMI 接口的时序参数	18-7
表 18-4 MDIO 接收时序参数表	18-11
表 18-5 MDIO 发送时序参数表	18-11
表 18-6 寄存器 SF_MDIO_REG0 中对比特的描述（写 PHY 寄存器）	18-12
表 18-7 寄存器 SF_MDIO_REG0 中对比特的描述（读 PHY 寄存器）	18-13
表 18-8 CPU 接收帧描述子数据结构	18-15
表 18-9 CPU 发送帧描述子数据结构	18-17
表 18-10 以太网交换单元寄存器概览（基址是 0xA002_0000）	18-18
表 18-11 STATIS 统计结果部分寄存器的属性和定义	18-65
表 18-12 ANYPORT 表项数据结构	18-74
表 18-13 ANYTYPE 表项数据结构	18-75
表 18-14 VLAN 表项数据结构	18-76
表 19-1 DES 寄存器概览（基址是 0x101F_B000）	19-4
表 19-2 DES_DIN 在不同的工作模式和位宽下的意义	19-4
表 19-3 DES 单元中工作模式和配置 DES_IVIN 的情况	19-5
表 19-4 DES 单元中密钥与 DES_KEY1 的关系	19-6
表 19-5 DES 单元中密钥与 DES_KEY2 的关系	19-7
表 19-6 DES 单元中密钥与 DES_KEY3 的关系	19-8
表 19-7 DES_DOUT 在不同的工作模式和位宽下的意义	19-9
表 19-8 DES 单元中工作模式和配置 DES_IVOUT 的情况	19-10
表 19-9 DES 的状态描述	19-11
表 19-10 AES 控制寄存器概览（基址是 0x101FE000）	19-14



表 19-11 AES_DIN 在不同的工作模式和位宽下的意义.....	19-15
表 19-12 AES 单元中工作模式和配置 AES_IVIN 的关系.....	19-16
表 19-13 AES_DOUT 在不同的工作模式和位宽下的意义.....	19-17
表 19-14 AES_IVOUT 在不同的工作模式下的意义.....	19-18
表 19-15 AES 的状态描述.....	19-19
表 20-1 模式配置接口信号描述.....	20-2
表 20-2 JTAG 接口信号描述.....	20-2
表 20-3 ETM9 调试接口信号描述.....	20-3
表 20-4 TEST_MODE 的含义.....	20-3
表 21-1 USB 时钟时序参数.....	21-2
表 21-2 SDRAM 接口时序参数列表.....	21-3
表 21-3 DDR 接口时序参数列表.....	21-4
表 21-4 I ² C 接口时序参数.....	21-5
表 21-5 SSP 主模式时序参数.....	21-6
表 21-6 ETM 时钟时序参数.....	21-7
表 21-7 ETM 接口时序参数.....	21-7
表 21-8 SIO 接口时序参数.....	21-8
表 21-9 VI 视频输入接口时序参数.....	21-9
表 21-10 VO 视频输出接口时序参数.....	21-9
表 21-11 50MHz 时 RMI 接口的时序参数.....	21-10
表 22-1 DC 参数表 (VDDIO33=3.3V).....	22-2
表 22-2 DC 参数表 (VDDIO25=2.5V).....	22-2
表 22-3 极限参数表.....	22-3
表 22-4 推荐工作条件.....	22-3
表 23-1 接口符号说明.....	23-2
表 23-2 Hi3510 接口信号描述表.....	23-2
表 23-3 管脚复用说明.....	23-19
表 23-4 电源和地管脚描述.....	23-23
表 24-1 封装参数说明表.....	24-4
表 24-2 Hi3510 V100 管脚数目统计表.....	24-4
表 24-3 管脚排列表 (按管脚名排序).....	24-7
表 24-4 管脚排列表 (按位置排序).....	24-13



表 A-1 管脚功能速查表..... A-1



前 言

概述

本节介绍本文档的内容、对应的产品版本、适用的读者对象、行文表达约定、历史修订记录等。

产品版本

与本文档相对应的产品版本如下所示。

产品名称	产品版本
Hi3510芯片	V100

读者对象

本文档（本指南）主要适用于以下工程师：

- 电子产品设计维护人员
- 电子产品元器件市场销售人员

内容简介

本文档介绍了数字媒体处理器芯片 Hi3510（以下简称 Hi3510）的特性、逻辑结构，详细描述芯片的各个模块的功能和模块相关寄存器含义，用图表的方式给出了接口时序关系和相关参数，并详细描述芯片的管脚定义和用途以及芯片的性能参数和封装尺寸。最后提供了术语表和缩略语。全书共分为 24 章和 1 个附录。

章节	内容
1 介绍	简要介绍Hi3510芯片主要特点、内部模块的功能和典型应用方案。
2 处理器子系统	详细介绍ARM926EJS和DSP两个子系统的特点，并介绍地址映射关系。
3 多端口静态和动态存储控制器	详细介绍多端口静态和动态存储控制器的功能、工作方式、各个寄存器的用途和使用方法。



章节	内容
4 多端口DDR SDRAM储存控制器	详细介绍多端口DDR SDRAM存储控制器（DDRC）的功能、工作方式、各个功能寄存器的用途和使用方法。
5 中断向量控制器（VIC）	详细介绍VIC的功能、工作方式、各个功能寄存器的用途和使用方法。
6 时钟、复位和系统控制器	详细介绍时钟单元和复位单元的功能和使用方法，并且还介绍了系统控制器的运行模式及各个功能寄存器的用途和使用方法。
7 直接存储访问控制器（DMAC）	详细介绍DMAC的功能、工作方式、各个功能寄存器的用途和使用方法。
8 Timer、WatchDog和RTC	详细介绍Timer、WatchDog和RTC的功能、工作方式及各个功能寄存器的用途和使用方法。
9 视频解码器单元	详细描述视频协处理器和视频编解码单元的特点。
10 2D图形加速引擎（TDE）	详细介绍TDE的功能、工作方式、各个功能寄存器的用途和使用方法。
11 视频输入单元（VIU）	详细介绍VIU的基本功能、接口信号、各个功能寄存器的用途和使用方法。
12 视频输出单元（VOU）	详细介绍VOU的功能、工作方式、各个功能寄存器的用途和使用方法。
13 串行输入输出接口（SIO）	详细介绍SIO的功能、工作方式、各个功能寄存器的用途和使用方法。
14 通用目的输入输出接口（GPIO）	详细介绍GPIO的功能、工作方式、各个功能寄存器的用途和使用方法。
15 通用异步收发器（UART）	详细介绍UART的功能、工作方式、各个功能寄存器的用途和使用方法。
16 SSP接口	详细介绍SSP接口的功能、工作方式、各个功能寄存器的用途和使用方法。
17 I ² C接口	详细介绍I ² C接口的功能、工作方式、各个功能寄存器的用途和使用方法。
18 以太网交换单元（SF）	详细介绍SF的功能、工作方式、各个功能寄存器的用途和使用方法。
19 加密单元	详细介绍DES和AES的功能、工作方式、各个功能寄存器的用途和使用方法。
20 模式配置与接口调试	详细介绍模式配置与接口调试的接口信号及工作方式。
21 时序和参数	详细介绍Hi3510芯片各个接口的时序和参数。
22 电性能参数	给出芯片的电性能参数、应用环境参数和可靠性参数。








章节	内容
23 管脚描述	详细列出Hi3510芯片接口信号功能和复用说明。
24 封装和管脚分布	描述Hi3510芯片的封装尺寸、管脚分布。
附录A 管脚功能速查表	列出Hi3510芯片的管脚，可实现快速查找。
附录B 术语表	收录本书中出现的术语，并给出了中文解释。
附录C 缩略语表	收录本书中出现的缩略语，并给出英文全称及中文解释。

约定

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	以本标志开始的文本表示有高度潜在危险，如果不能避免，会导致人员死亡或严重伤害。
 警告	以本标志开始的文本表示有中度或低度潜在危险，如果不能避免，可能导致人员轻微或中等伤害。
 注意	以本标志开始的文本表示有潜在风险，如果忽视这些文本，可能导致设备或器件损坏、数据丢失、设备性能降低或不可预知的结果。
 窍门	以本标志开始的文本能帮助您解决某个问题或节省您的时间。
 说明	以本标志开始的文本是正文的附加信息，是对正文的强调和补充。

通用格式约定

格式	说明
宋体	正文采用宋体表示。
黑体	一级、二级、三级标题采用黑体。
楷体	警告、提示等内容一律用楷体，并且在内容前后增加线条与正文隔离。



格式	说明
“Terminal Display”格式	“Terminal Display”格式表示屏幕输出信息。此外，屏幕输出信息中夹杂的用户从终端输入的信息采用加粗字体表示。

说明

寄存器访问类型说明

符号	说明
R	表示寄存器的属性为只读。
W	表示寄存器的属性为只写。
R/W	表示寄存器的属性为可读和可写。
RO	只读型，CPU只能对寄存器读操作。
RW	读/写型，CPU能够对寄存器读操作或写操作。
RWC	读/写清零型，CPU可读但只能写0，内部功能模块只能将寄存器设置为1。
RWS	读/写置位型，CPU可读但只能写1，内部功能模块只能将其设置为0。
RC	读清零型，CPU可读，读取后该寄存器被内部功能模块清零，但只能将其置位。
RCI	读清零/递增型，CPU可读，只能通过将其写0才能将其清零，内部功能模块在计数条件满足时对寄存器进行加1操作，实现相关统计功能。

表格内容说明

内容	说明
-	表格中的无内容单元。
*	表格中的内容用户可根据需要进行信息配置。

数值单位说明

在描述数据容量（如 RAM 容量）时：1K 代表 1024；1M 代表 1,048,576。

在描述其他数据（如频率、数据速率等）时：1K 代表 1000；1M 代表 1,000,000；1G 代表 1,000,000,000。

地址、数据的 2 进制、16 进制表达方式说明如下：



符号	举例	说明
0xXXXX、0xXX等	0xFE04、0x18	用16进制表示的地址值或复位值
XXXH、XXH等	174H、74H	用16进制表示的数据值
XXXB、XXB、XB等	001B、10B、0B	用2进制表示的数据值
X	0、1	用2进制表示的数据值

修改记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修改日期	版本	修改说明
2007-04-20	04	<ul style="list-style-type: none"> • 修改了 22 章电性能参数一章的部分管脚信息。 • 增加了表 23-2 中“复位状态”一列，修改部分管脚的描述。 • 修改了“<code>IIC_COMP_PARAM_1</code>”的复位值。
2007-01-31	03	<ul style="list-style-type: none"> • 套用新模板。 • 调整原文档中各章节的顺序。 • 寄存器描述中将复位值总体写在表格的前面，不再在表格中对应每个比特列出。 • 将概述中“DRM 和 De-interlace 处理”的特性删除。 • 视频编解码支持“H.264 baseline profile@Level 2.2”改为“H.264 baseline profile@Level 3.0”； • “1.0W 典型功耗”改为“600mW 典型功耗（DVS、IPCamera 应用环境下）”。 • Remap 前后的地址映射关系图都有修改。 • 多个章节增加了功能框图。 • GPIO 一章增加了配置示例。 • SF 一章进行了较多的内容改动。



修改日期	版本	修改说明
2006-06-01	02	<ul style="list-style-type: none">• 删除 SF 的 NAT 和内环回等功能及相关寄存器描述。• 修改“任意比例的视频、图形缩放”为“支持视频、图形缩放”。• 发现正文的 Word 文档的所有二级标题都是从 1.1 开始，重新调整。• 修改了第 23 章（23.2.2 ETM9 接口说明）的 GPIO 描述• 修改“信号描述”一章的标题“复位信号”为“复用信号”。• 修改“时钟与复位”一章中的表 22-2 中“25MHz 钟振时钟”为“27MHz 钟振时钟”；• 修改 GPIO 复用信号 GPIO2[1]、GPIO2[0]的复用描述。• 修改了正文中的某些章节的第一小节序号，补充了 MEMC 的缩略语解释。• 因翻译发现部分问题，修改了第 3 章、第 7、8、9、10、11 章的描述，修改了第 12 章的信号描述。• 修改 ZSP、ZSP500 为 DSP。
2005-11-11	01	<ul style="list-style-type: none">• 第一次发布。



1 介绍

关于本章

本章描述内容如下表所示。

标题	内容
1.1 概述	概述 Hi3510 芯片。
1.2 主要特点	介绍 Hi3510 芯片中视频编解码、图形处理、以太网交换接口、外围接口和芯片物理规格等单元的特点。
1.3 应用领域	简单列举 Hi3510 芯片的应用领域。
1.4 典型应用	举例说明 Hi3510 芯片的应用。



1.1 概述

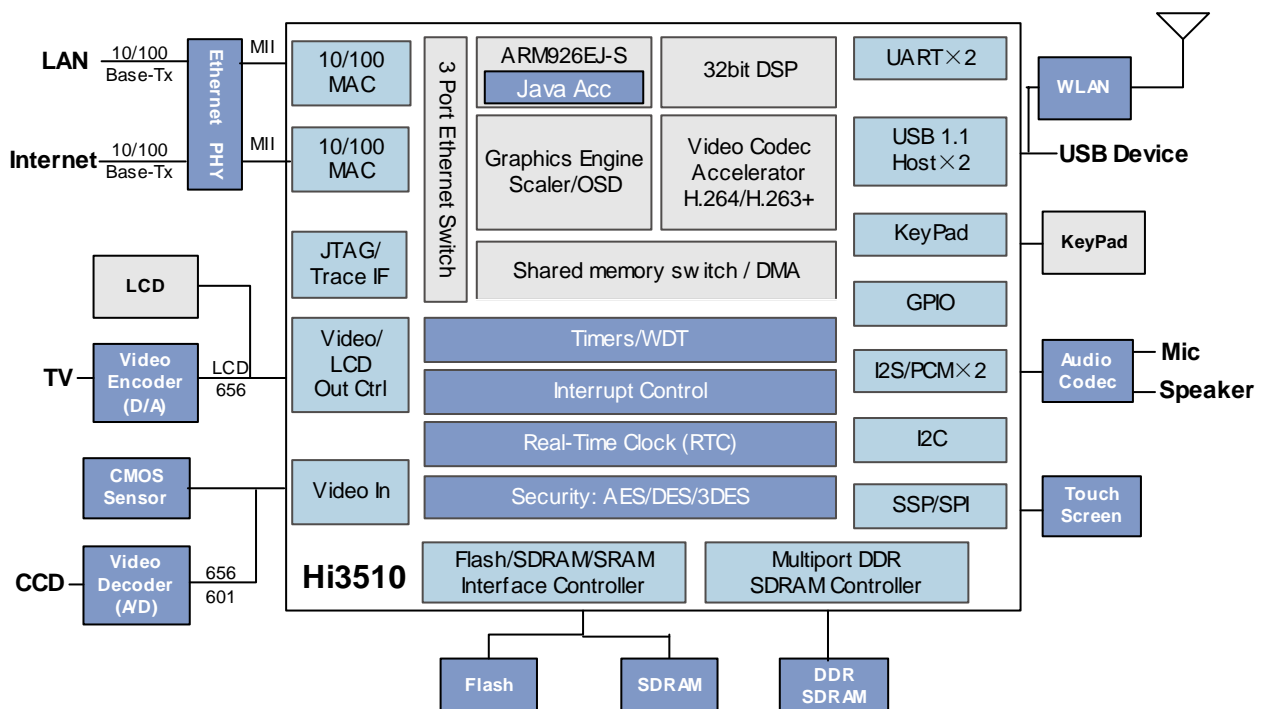
Hi3510 是一款基于 ARM9、DSP 双处理器内核以及硬件加速引擎的高集成、可编程、支持 MPEG-4 AVC/H.264 等多协议的高性能通信媒体处理器芯片，可广泛应用于实时视频通信、数字图像监控等领域。

视频编解码单元能够支持 MPEG-4 AVC/H.264 Baseline、H.263+、H.261 和 JPEG 等多种协议的实时编解码。MPEG-4 AVC/H.264 先进的运动估计、运动补偿、de-blocking 技术极大提高了压缩效率及视频质量。加密和数字水印技术为数据和通信的安全提供了强有力的保障。

图形处理单元能够提供视频去噪、图像增强和运动检测功能；支持视频、图形缩放；支持 OSD、2D 图形加速，为应用图形界面开发提供丰富的特性。

内部集成的 3 端口以太网交换提供了强大的网络通信功能；硬件加密和数字水印技术提供了丰富的安全特性。支持视音频接口、以太网接口、USB、UART、I²C、SPI/SSP、GPIO 等多种丰富的外设接口。

图1-1 Hi3510 内部逻辑框图





1.2 主要特点

本节简单描述了 Hi3510 的特点。

1.2.1 内嵌 RISC 内核

- ARM926EJ-S, 16KB 指令 Cache 和 16KB 数据 Cache
- 内嵌 16KB 指令紧耦合存储器和 8KB 数据紧耦合存储器
- 哈佛结构的 32 位 RISC 处理器
- DSP 增强结构, 内嵌 32×16 MAC (Multiply and Accumulate, 即乘累加器)
- Java 硬件加速
- 内置 MMU, 支持多种开放式操作系统
- 工作频率可达到 240MHz

1.2.2 内嵌 DSP 内核

- 3 个 ALU (1 个 40 位, 2 个 16 位)
- 8 级流水线设计
- 4 发射超标量结构, 双 MAC (Multiply and Accumulate) 结构

1.2.3 视频编解码

- H.264 视频编解码 baseline profile@Level 2.2
- H.263+视频编解码
- H.261 视频编解码
- JPEG 编码, 支持百万像素级分辨率

1.2.4 视频处理性能

- 同时编解码可以达到 30fps@CIF
- 最大支持 D1 分辨率
- 运动检测
- 支持数据带宽 16kbit/s~3Mbit/s

1.2.5 图形处理

- 支持视频、图形缩放
- 视频层、2 个叠加层和硬件鼠标层 alpha 叠加
- 视频去噪, 图像增强
- 2D 图形加速引擎, 支持游戏以及丰富的图像界面



1.2.6 音频编解码

- 可以通过 DSP 内核实现多种音频、语音编解码功能

1.2.7 安全加密引擎

- 硬件实现 AES/DES/3DES 多种加密算法
- 数字水印技术

1.2.8 以太网交换接口

- 1 个 MDIO 接口
- 2 个 MII/RMII 接口
- 2 个外部 10/100 Mbit/s 以太网外部端口与 1 个内部 CPU 端口
- 3 端口之间通过存储转发的方式实现数据交换
- 以太网交换模块可以配置为普通、监听 2 种工作模式
- 支持广播帧、IP 多播帧及两类可配置特殊帧的识别和转发
- 支持 IEEE 802.1p 输出优先级配置及 802.1Q VLAN 处理

1.2.9 视频接口

输入:

- 8 bit ITU-R BT.656/601 YCbCr 4:2:2
- 数字 camera 接口
- Raw Data 接口

输出:

- 8 bit ITU-R BT.656/601 YCbCr 4:2:2
- 24 位 LCD 接口, RGB/YCbCr 数据格式
- Raw Data 接口

1.2.10 音频接口

- 2 个 I²S 音频接口, 输入、输出接口各 2 个通道
- 16 位采样精度, 采样率可配置
- PCM 接口

1.2.11 外设接口

- USB1.1 Host 接口, 支持低速、全速模式
- UART、I²C 接口、SSP/SPI 串行接口
- GPIO、键盘接口



1.2.12 外部存储器接口

- 支持 16 数据位宽 DDR SDRAM 接口
- 支持 32 数据位宽 SDRAM 接口
- 支持 8/16/32 数据位宽外部静态存储接口
- 支持 8/16/32 数据位宽扩展总线接口

1.2.13 嵌入式操作系统

- Linux、WinCE、VxWorks

1.2.14 芯片物理规格

- 600mW 典型功耗（DVS、IP Camera 应用环境下）
- 支持多级省电模式
- 0.13 μ m 工艺，1.2/2.5/3.3 V 芯片供电电压
- 400 pin LFBGA 封装，0.8mm 管脚间距，19mm \times 19mm \times 1.36mm
- 工作环境温度：-25 $^{\circ}$ C \sim +85 $^{\circ}$ C

1.3 应用领域

Hi3510 可应用在以下几个领域：

- 视频通信终端
- 宽带可视电话
- 网络视频监控

1.4 典型应用

图 1-2 和图 1-3 分别描述了 Hi3510 在宽带可视电话、网络视频监控的典型应用。

图1-2 Hi3510 宽带 IP 可视电话典型应用

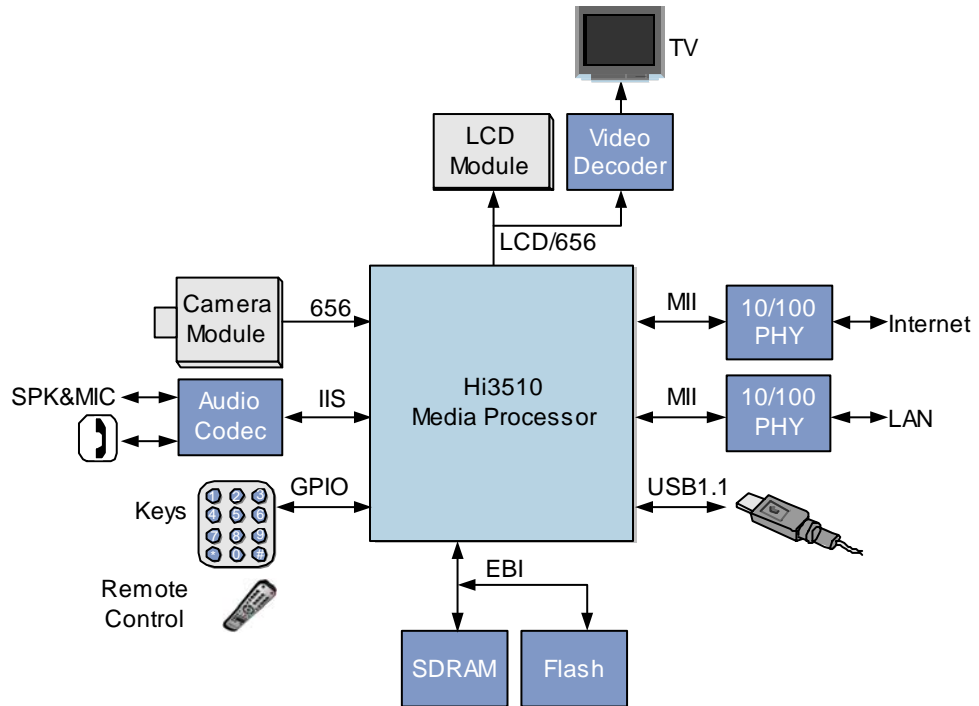
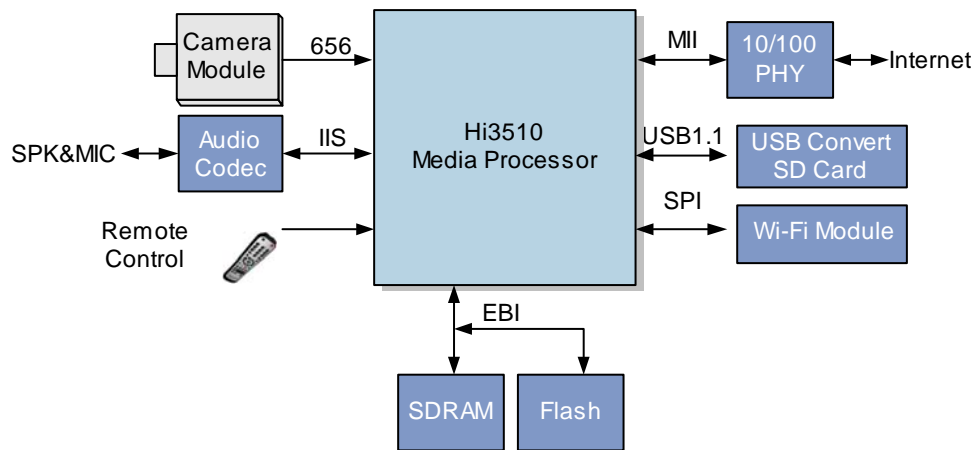


图1-3 Hi3510 IP Camera 典型应用





2 处理器子系统

关于本章

本章描述内容如下表所示。

标题	内容
2.1 概述	概述介绍处理器子系统。
2.2 ARM926EJ-S 主要特点	介绍 ARM926EJ-S 的主要特点。
2.3 DSP 子系统主要特点	介绍 DSP 子系统的主要特点。
2.4 Hi3510 地址映射关系	介绍 Hi3510 Remap 前后的地址映射。



2.1 概述

Hi3510 采用双核架构，集成了 ARM926EJ-S 和高性能 DSP。ARM926EJ-S 是芯片的主控处理器，负责控制芯片各个模块的工作以及运行操作系统、网络协议、应用软件等。DSP 系统主要负责视音频编解码业务处理，通过协处理器配合视频编解码模块共同完成 H.261/H.263/H.264 的编解码。

2.2 ARM926EJ-S 主要特点

ARM926EJ-S 的主要特点如下：

- 采用 32 位 ARM v5TE ISA，5 级流水，内嵌 DSP 指令扩展，兼容 32 位 ARM、16 位 Thumb 指令集。
- 提供独立的 16kB 指令 Cache、16kB 数据 Cache、4 路 Set-associate 和 8word Cache line；数据 Cache 支持 Write-back 和 Write-through 操作；支持 Cache 锁定。
- Cache 支持伪随机或者 Round-robin 替换算法，并可进行配置。
- 独立 32 位指令和数据 AHB 总线接口，总线工作频率为 ARM926EJ-S 系统时钟的 1/2 分频。
- 包含 MMU，支持 VxWorks、Linux、WindowCE、PalmOS 等开放操作系统。
- 提供独立 16kB 指令紧耦合存储器 TCM 和 8kB 数据紧耦合存储器 TCM。
- 小端字序模式（little endian）。
- 支持快速中断请求 FIQ 和一般中断请求 IRQ。
- 支持 JTAG 和 ETM 调试接口。
- 支持动态功耗管理和静态功耗管理。

2.3 DSP 子系统主要特点

Hi3510 的 DSP 子系统包括 DSP Core、片上指令存储器 IMEM、片上数据存储器 DMEM、中断控制器、定时器和 JTAG 接口单元等。

DSP Core 是一款类 RISC（RISC-like）、4-issues、8 级流水线设计的超标量（super-scalar）DSP 处理器。由取指单元（PFU）、指令单元（ISU）、数据访问单元（LSU）、运算单元（ALU、MAU）、TIMER、中断处理单元（ICU）以及寄存器组组成。

DSP 子系统的主要特点如下：

- 运算单元（ALU、MAU）在一个周期最多能够执行 4 条指令，它主要包括：
 - 2 个 16 位的 ALU
 - 1 个 40 位的 ALU
 - 2 个 16 位的 MAC
- 取指单元（PFU），主要包括：



- 1 个 8line (8word per line) 的指令 Cache
- Branch prediction logic
可以高效地进行指令预取，将 Cache miss 的几率降到最低，保证有效跳转时指令流水线能够全速运行。
- 数据访问单元 (LSU) 主要包括：
 - 2 个 32 位的 Load/Store 端口
 - 2 个地址产生单元 AGU
可以提高访问的并行度。
- 指令单元 (ISU)，主要用于：
 - 指令的译码、分析指令的相关性 (数据相关性、资源相关性等)
 - 对指令进行分组 (将能够并行执行的指令分为一组)
保证流水线的高效正常工作。
- TIMER 主要功能是为 DSP 子系统提供系统定时，在 DSP 中，提供了 2 个 16 位的定时器。
- ICU 是 DSP 提供的中断处理单元，它能够支持 16 个中断源，具体中断源的分配如表 2-1 所示。

表2-1 DSP 子系统中断源分配表

中断号	优先级	中断源	中断类型
0	0、1、2、3	DMAC 中断	外部可屏蔽中断，可将 DSP 从 Idle 和 Sleep 状态唤醒。
1	0、1、2、3	ARM 中断	
2	0、1、2、3	VIU 中断	
3	0、1、2、3	VOU 中断	
4	0、1、2、3	DSU 中断	
5	0、1、2、3	DSP 内部 TIMER0 中断	内部中断。
6	0、1、2、3	DSP 内部 TIMER1 中断	
7	0、1、2、3	保留	可屏蔽中断，可将 DSP 从 Idle 状态唤醒。
8	0、1、2、3	SIO0 中断	
9	0、1、2、3	de-blocking 中断	
10	0、1、2、3	视频编码模块 VENC 中断	
11	0、1、2、3	视频解码模块 VDEC 中断	
12	0、1、2、3	SIO1 中断	
13	0、1、2、3	DSP 访问外部空间错误中断	
14	4	DEI 调试中断，用于 JTAG 调试	



中断号	优先级	中断源	中断类型
15	5	ARM 通过系统控制器 PeriphCtrl1[29]发出的中断	不可屏蔽中断，主要用于将 DSP 从睡眠状态唤醒。

注：

中断源 0~13 的中断优先级可以配置；

DEI 和 NMI 是固定优先级，不可以配置；

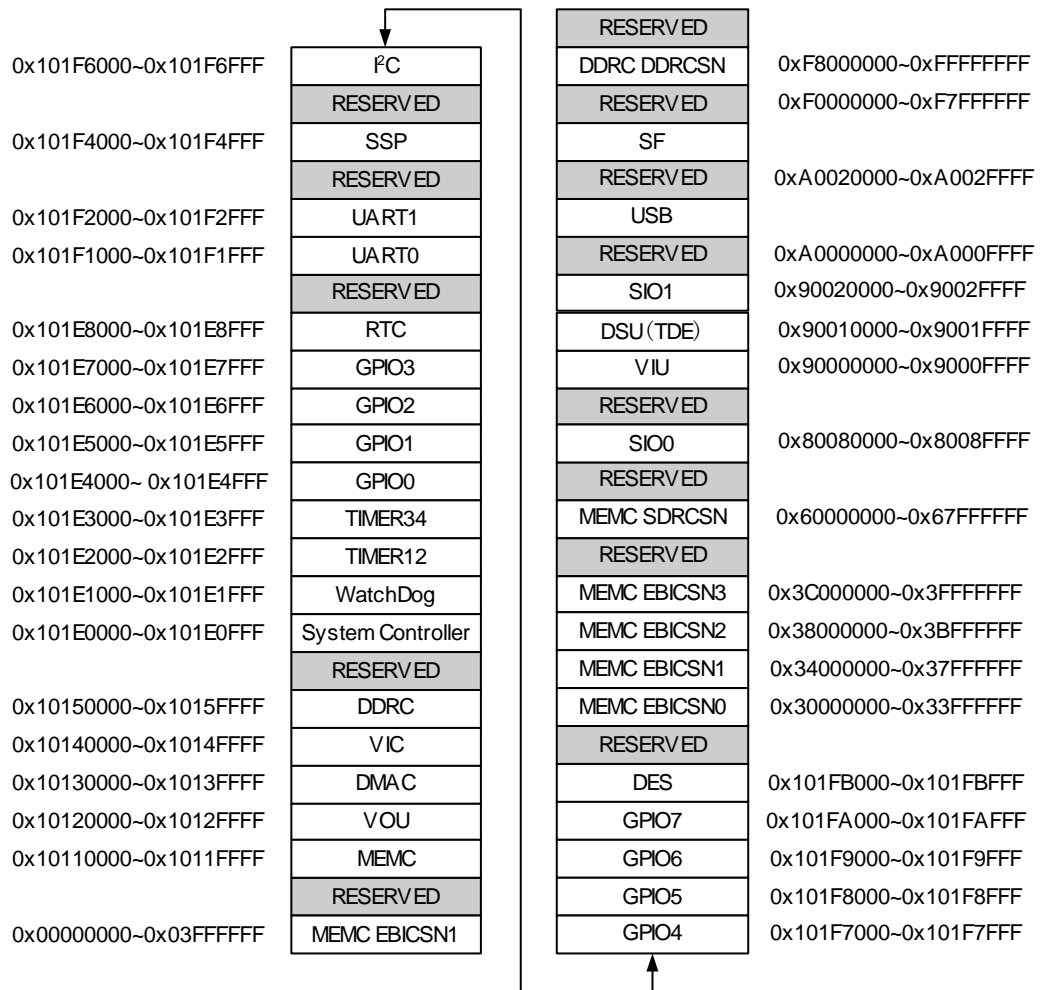
中断优先级编号越大表示优先级越高。

2.4 Hi3510 地址映射关系

Hi3510 采用统一编址方式。图 2-1 和图 2-2 分别是 Hi3510 Remap 前后的地址映射。



图2-1 Remap 前 Hi3510 地址映射关系图

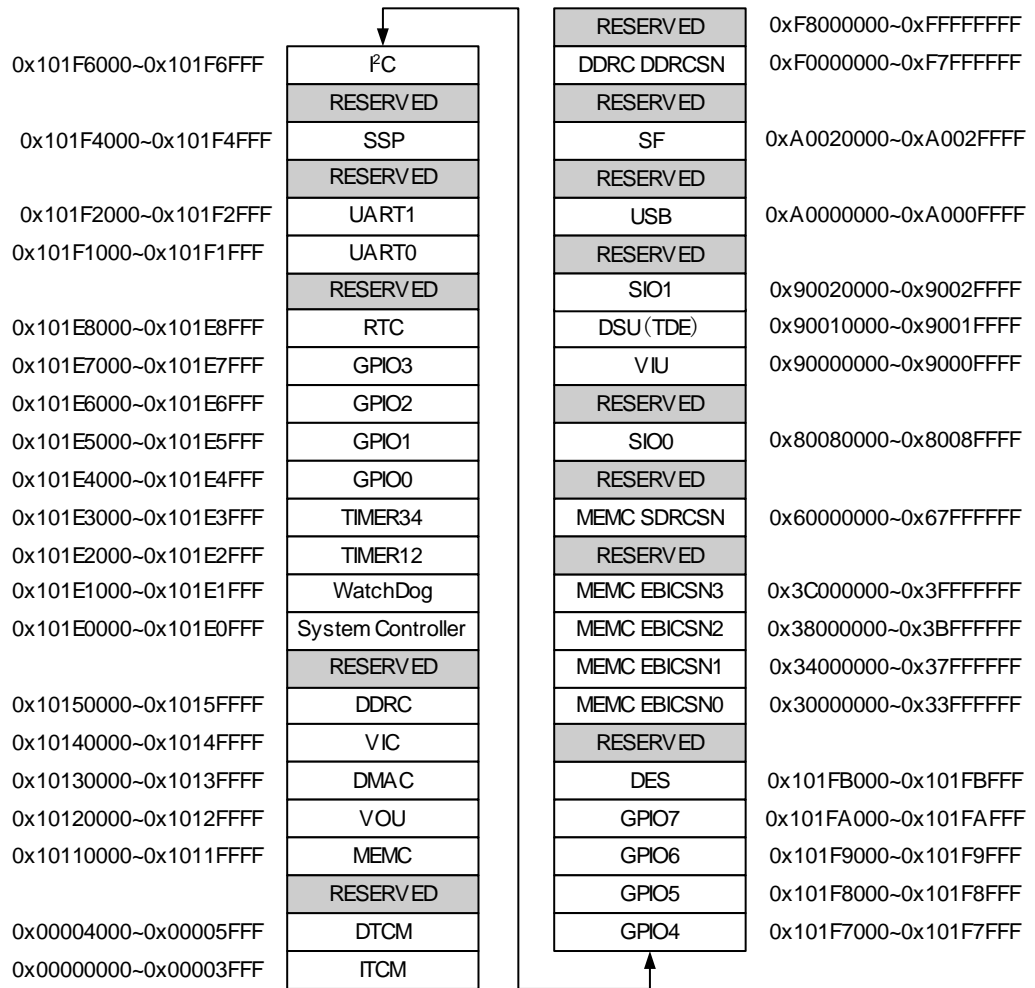




注意

- Remap 后软件必须通过 ARM926EJ-S CP15 协处理器配置 ITCM 基址为 0x00000000; 建议将 DTCM 的基址设置为 0x00004000。
- 保留地址空间不容许访问, 否则会出现不可预知后果。

图2-2 Remap 后 Hi3510 地址映射关系图





3 多端口静态和动态存储控制器

关于本章

本章描述内容如下表所示。

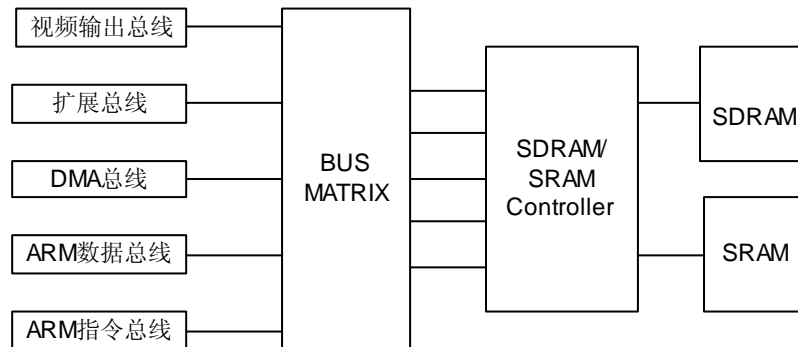
标题	内容
3.1 概述	概括介绍 MEMC。
3.2 功能描述	概括介绍 MEMC 的特点。
3.3 信号描述	描述 MEMC 的输入输出管脚信号，
3.4 寄存器概览	概括介绍 MEMC 的寄存器。
3.5 寄存器描述	详细描述 MEMC 的寄存器。



3.1 概述

多端口静态和动态存储控制器（MEMC）用于提供访问外部 SDR SDRAM 和静态异步存储器的通道，为系统提供灵活的外部存储系统方案。其他单元可通过该控制器访问芯片外部的静态（包括 NOR Flash）和动态存储器。MEMC 功能框图如图 3-1 所示。

图3-1 MEMC 功能框图



3.2 功能描述

MEMC 有以下特点：

- 提供 1 个动态 SDR SDRAM 接口
- 支持最大容量为 128MByte（1024M bits），位宽可为 32/16 位的 SDR SDRAM
- 提供 4 个异步静态 Memory 接口，每个最大容量支持 64M Byte（512M bits）
- 支持同步或异步 Page 模式访问，支持 NOR Flash 访问，支持 8、16、32 数据位宽
- 提供 1 个寄存器端口，用于配置 Memory 的接口时序
- 静态 Memory 接口和动态 Memory 接口的数据总线、写信号、地址总线复用
- 连接 SDR SDRAM 时，支持的 burst 长度为 1 或 2
- 支持 SDR SDRAM 的 Auto Refresh 和 Self Refresh
- 控制时钟输出使能，以降低 SDRAM 存储器的功耗

3.3 信号描述

本节描述 MEMC 的输入输出管脚信号。多端口静态存储控制器接口信号描述如表 3-1 所示；多端口动态存储控制器接口信号描述如表 3-2 所示。



表3-1 多端口静态存储控制器接口信号描述

管脚名	方向	描述
EBICS3N	O	静态 Memory 片选信号 3，低电平有效。默认为高电平。
EBICS2N	O	静态 Memory 片选信号 2，低电平有效。默认为高电平。
EBICS1N	O	静态 Memory 片选信号 1，低电平有效。默认为高电平。 上电复位，地址 0 访问时，该信号输出为低电平。 一般用于连接 Boot ROM 器件。
EBICS0N	O	静态 Memory 片选信号 0，低电平有效。
BOOTSEL[1:0]	I	用于设置静态 Memory 片选 1 上所挂接 Memory 的数据位宽。 00: 8 位; 01: 16 位; 10: 32 位; 11: Reserved。
EBIBLS3	O	静态 Memory 字节选择信号，低电平有效；对应数据 EBIDQ[31:24]。
EBIBLS2	O	静态 Memory 字节选择信号，低电平有效；对应数据 EBIDQ[23:16]。
EBIBLS1	O	静态 Memory 字节选择信号，低电平有效；对应数据 EBIDQ[15:8]。
EBIBLS0	O	静态 Memory 字节选择信号，低电平有效；对应数据 EBIDQ[7:0]。
EBIOEN	O	静态 Memory 输出使能信号，低电平有效。
EBIWEN	O	静态 Memory 写使能信号，低电平有效。
EBIDQ[31:0]	I/O	静态 Memory 接口 32 位数据总线。
EBIADR[25:0]	O	静态 Memory 接口地址线。

表3-2 多端口动态存储控制器接口信号描述

信号名	方向	描述
SDRCKFB	I	用于读 SDRAM 的数据采样，通过 PCB 布线来保证 MEMC 可靠采样来自 SDRAM 的数据。
SDRCK1	O	输出到 SDRAM 的时钟信号。



信号名	方向	描述
SDRCK2	O	输出到 SDRAM 的时钟信号，当连接 2 片 SDR SDRAM 时，SDRCK1 和 SDRCK2 可送给不同的 SDRAM。
SDRRASN	O	输出到 SDRAM 的行地址选通信号，低电平有效。
SDRCASN	O	输出到 SDRAM 的列地址选通信号，低电平有效。
SDRCSN	O	SDR SDRAM 片选信号，低电平有效。
SDRDM3	O	输出到 SDRAM 的数据屏蔽信号，对应数据位 EBIDQ[31:24]。
SDRDM2	O	输出到 SDRAM 的数据屏蔽信号，对应数据位 EBIDQ[23:16]。
SDRDM1	O	输出到 SDRAM 的数据屏蔽信号，对应数据位 EBIDQ[15:8]。
SDRDM0	O	输出到 SDRAM 的数据屏蔽信号，对应数据位 EBIDQ[7:0]。
EBIWEN	O	SDRAM 写使能信号，低电平有效。
EBIDQ[31:16]	I/O	接口 SDRAM 的高 16 位数据线。
EBIDQ[15:0]	I/O	接口 SDRAM 的低 16 位数据线。
SDRCKE	O	SDRAM 接口时钟使能信号，高电平有效。
EBIADR[14:13]	O	分别对应为 SDRAM 接口的 bank 选择信号。
EBIADR[12:0]	O	输出到 SDRAM 的地址线。

3.4 寄存器概览

MEMC 寄存器的地址位宽 32 位，地址范围：0x1011_0000~0x1011_FFFF。

表3-3 MEMC 寄存器概览（基址是 0x1011_0000）

偏移地址	名称	描述	页码
0x000	MEMC_CONTROL	MEMC 控制寄存器	3-8
0x004	MEMC_STATUS	MEMC 状态寄存器	3-8
0x008	MEMC_CONFIG	MEMC 配置寄存器	3-9
0x020	MEMC_DYNAMIC CONTROL	SDR SDRAM 动态 Memory 控制寄存器	3-9



偏移地址	名称	描述	页码
0x024	MEMC_DYNAMIC REFRESH	SDR SDRAM 动态 Memory 刷新寄存器	3-11
0x028	MEMC_DYNAMIC READCONFIG	SDR SDRAM 动态 Memory 读配置寄存器	3-11
0x030	MEMC_DYNAMIC TRP	SDR SDRAM 动态 Memory 预充电命令周期寄存器	3-12
0x034	MEMC_DYNAMIC TRAS	SDR SDRAM 动态 Memory 激活到预充电命令的周期寄存器	3-12
0x038	MEMC_DYNAMIC SREX	SDR SDRAM 动态 Memory 自我刷新退出时间寄存器	3-12
0x044	MEMC_DYNAMIC TWR	SDR SDRAM 动态 Memory 写恢复时间寄存器	3-13
0x048	MEMC_DYNAMIC TRC	SDR SDRAM 动态 Memory 激活到激活时间寄存器	3-13
0x04C	MEMC_DYNAMIC TRFC	SDR SDRAM 动态 Memory 自动刷新寄存器	3-13
0x050	MEMC_DYNAMIC TXSR	SDR SDRAM 动态 Memory 退出自我刷新寄存器	3-14
0x054	MEMC_DYNAMIC TRRD	SDR SDRAM 动态 Memory 激活 bank A 到激活 bank B 的时间寄存器	3-14
0x058	MEMC_DYNAMIC TMRD	SDR SDRAM 动态 Memory load mode 寄存器	3-14
0x05C	MEMC_DYNAMIC TCDLR	SDR SDRAM 动态 Memory 最后一个数据输入到读命令的时间寄存器	3-15
0x080	MEMC_STATICEXTENDEDWAIT	静态 Memory 扩展等待寄存器	3-15
0x100	MEMC_DYNAMIC CONFIG0	SDR SDRAM 动态 Memory 配置寄存器 0	3-16
0x104	MEMC_DYNAMIC RASCAS0	SDR SDRAM 动态 Memory RAS 及 CAS 延时寄存器 0	3-19
0x200	MEMC_STATICCONFIG0	静态 Memory 配置寄存器 0	3-20
0x204	MEMC_STATICWRITEWEN0	静态 Memory 写使能延迟寄存器 0	3-21
0x208	MEMC_STATICWRITEOEN0	静态 Memory 输出使能延迟寄存器 0	3-21



偏移地址	名称	描述	页码
0x20C	MEMC_STIATICW AITRD0	静态 Memory 读延迟寄存器 0	3-21
0x210	MEMC_STIATICW AITPAGE0	静态 Memory page mode 延迟寄存器 0	3-22
0x214	MEMC_STIATICW AITWR0	静态 Memory 写延迟寄存器 0	3-22
0x218	MEMC_STIATICW AITTURN0	静态 Memory Turn Round Delay 寄存器 0	3-23
0x220	MEMC_STIATICC ONFIG1	静态 Memory 配置寄存器 1	3-20
0x224	MEMC_STIATICW AITWEN1	静态 Memory 写使能延迟寄存器 1	3-21
0x228	MEMC_STIATICW AITOEN1	静态 Memory 输出使能延迟寄存器 1	3-21
0x22C	MEMC_STIATICW AITRD1	静态 Memory 读延迟寄存器 1	3-21
0x230	MEMC_STIATICW AITPAGE1	静态 Memory page mode 延迟寄存器 1	3-22
0x234	MEMC_STIATICW AITWR1	静态 Memory 写延迟寄存器 1	3-22
0x238	MEMC_STIATICW AITTURN1	静态 Memory Turn Round Delay 寄存器 1	3-23
0x240	MEMC_STIATICC ONFIG2	静态 Memory 配置寄存器 2	3-20
0x244	MEMC_STIATICW AITWEN2	静态 Memory 写使能延迟寄存器 2	3-21
0x248	MEMC_STIATICW AITOEN2	静态 Memory 输出使能延迟寄存器 2	3-21
0x24C	MEMC_STIATICW AITRD2	静态 Memory 读延迟寄存器 2	3-21
0x250	MEMC_STIATICW AITPAGE2	静态 Memory page mode 延迟寄存器 2	3-22
0x254	MEMC_STIATICW AITWR2	静态 Memory 写延迟寄存器 2	3-22
0x258	MEMC_STIATICW AITTURN2	静态 Memory Turn Round Delay 寄存器 2	3-23
0x260	MEMC_STIATICC ONFIG3	静态 Memory 配置寄存器 3	3-20



偏移地址	名称	描述	页码
0x264	MEMC_STIATICW AITWEN3	静态 Memory 写使能延迟寄存器 3	3-21
0x268	MEMC_STIATICW AITOEN3	静态 Memory 输出使能延迟寄存器 3	3-21
0x26C	MEMC_STIATICW AITRD3	静态 Memory 读延迟寄存器 3	3-21
0x270	MEMC_STIATICW AITPAGE3	静态 Memory page mode 延迟寄存器 3	3-22
0x274	MEMC_STIATICW AITWR3	静态 Memory 写延迟寄存器 3	3-22
0x278	MEMC_STIATICW AITTURN3	静态 Memory Turn Round Delay 寄存器 3	3-23
0x400	MEMC_AHBCONT ROL0	MEMC AHB 控制寄存器 0	3-23
0x404	MEMC_AHBSTAT US0	MEMC AHB 状态寄存器 0	3-23
0x408	MEMC_AHBTIME OUT0	MEMC AHB Timeout 寄存器 0	3-24
0x420	MEMC_AHBCONT ROL1	MEMC AHB 控制寄存器 1	3-23
0x424	MEMC_AHBSTAT US1	MEMC AHB 状态寄存器 1	3-23
0x428	MEMC_AHBTIME OUT1	MEMC AHB Timeout 寄存器 1	3-24
0x440	MEMC_AHBCONT ROL2	MEMC AHB 控制寄存器 2	3-23
0x444	MEMC_AHBSTAT US2	MEMC AHB 状态寄存器 2	3-23
0x448	MEMC_AHBTIME OUT2	MEMC AHBTimeout 寄存器 2	3-24
0x460	MEMC_AHBCONT ROL3	MEMC AHB 控制寄存器 3	3-23
0x464	MEMC_AHBSTAT US3	MEMC AHB 状态寄存器 3	3-23
0x468	MEMC_AHBTIME OUT3	MEMC AHB Timeout 寄存器 3	3-24
0x480	MEMC_AHBCONT ROL4	MEMC AHB 控制寄存器 4	3-23



偏移地址	名称	描述	页码
0x484	MEMC_AHBSTAT US4	MEMC AHB 状态寄存器 4	3-23
0x488	MEMC_AHBTIME OUT4	MEMC AHB Timeout 寄存器 4	3-24

3.5 寄存器描述

本节详细描述了 MEMC 的寄存器。

3.5.1 MEMC_CONTROL

- 偏移地址：0x000
- 操作类型：R/W
- 复位值：0x1
- 复位方式：h

比特	名称	描述
[31:3]	Reserved	保留。
[2]	L	模式选择。 0: 普通模式; 1: 低功耗模式。
[1]	Reserved	保留。
[0]	E	MEMC 使能标志位。 0: MEMC disable; 1: MEMC enable (上电复位值)。

3.5.2 MEMC_STATUS

- 偏移地址：0x004
- 操作类型：R
- 复位值：0x9
- 复位方式：h

比特	名称	描述
[31:3]	Reserved	保留。



比特	名称	描述
[2]	SA	模式选择。 0: 普通模式; 1: self-refresh 模式。
[1]	Reserved	保留。
[0]	B	MEMC 状态标志位。 0: MEMC idle; 1: MEMC busy, 表示正忙于执行存储器的传输命令, 自动刷新或正处于自我刷新模式。

3.5.3 MEMC_CONFIG

- 偏移地址: 0x008
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h

比特	名称	描述
[31:1]	Reserved	保留。
[0]	B	模式选择。 0: little endian 模式; 1: big endian 模式。 Hi3510 中只支持 little endian 模式, 即 MEMC_BIGENDIAN 固定为 0。 上电复位值信号由信号 MEMC_BIGENDIAN 决定, 该复位值可以被软件更改。该域不受 HRESETn 的影响。

3.5.4 MEMC_DYNAMICCONTROL

- 偏移地址: 0x020
- 操作类型: R/W
- 复位值: 0xE_000E
- 复位方式: h

比特	名称	描述
[31:14]	Reserved	保留。



比特	名称	描述
[13]	DP	模式选择。 0: 普通模式; 1: 进入深度睡眠, 设备断电并不再刷新。
[12:9]	Reserved	保留。
[8:7]	I	SDRAM 命令选择。 00: 发出 SDRAM NORMAL 操作命令; 01: 发出 SDRAM MODE 命令; 10: 发出 SDRAM PALL (Precharge all) 命令; 11: 发出 SDRAM NOP 操作命令。
[6]	Reserved	保留。
[5]	MCC	存储器时钟控制。 0: SDRCK1、SDRCK2 enable; 1: SDRCK1、SDRCK2 disable。
[4]	Reserved	保留。
[3]	SRMCC	0: SDRCK1、SDRCK2 在 self-refresh 模式下停止; 1: SDRCK1、SDRCK2 连续工作。
[2]	SR	模式选择。 0: 普通模式; 1: 进入自我刷新模式。 软件可以通过向该位写 1 进入自我刷新模式, 写 0 进入普通模式。 MEMC 状态寄存器的自我刷新响应位对此作出的响应决定 MEMC 处于何种模式。
[1]	CS	动态存储器的时钟控制。 0: 当所有的 SDRAM 处于 idle 时 SDRCK1、SDRCK2 停止; 1: SDRCK1、SDRCK2 连续工作。 说明 该功能仅用于 SDRAM。
[0]	CE	动态存储器的时钟使能。 0: 空闲设备的时钟使能无效以省电; 1: 所有时钟使能一直有效。 说明 初始化时该位必须为 1。



3.5.5 MEMC_DYNAMICREFRESH

- 偏移地址：0x024
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h

比特	名称	描述
[31:11]	Reserved	保留。
[10:0]	REFRESH	刷新周期。 11'h0: 刷新 disable; 11'h1~11'h7FF: SDRAM 刷新周期时间为 $16 \times n$ 时钟 cycle。n 表示对应的十进制值。 例如 11'h8: 128 个时钟 cycle (16×8)。

3.5.6 MEMC_DYNAMICREADCONFIG

- 偏移地址：0x028
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h

比特	名称	描述
[31:5]	Reserved	保留。
[4]	SRP	SDR_SDRAM 采读数据。 0: SDR_SDRAM 用 HCLK 的下沿采读数据; 1: SDR_SDRAM 用 HCLK 的上沿采读数据。
[3:2]	Reserved	保留。
[1:0]	SRD	SDR SDRAM 读数据策略。 00: 时钟延迟策略, 使用 MEMC_CLKOUT (时钟输出延迟, 命令不延迟); 01: 命令延迟策略, 使用 MEMC_CLKDELAY (命令延迟, 时钟输出不延迟); 10: 命令延迟策略+1 个时钟周期, 使用 MEMC_CLKDELAY (命令延迟, 时钟输出不延迟); 11: 命令延迟策略+2 个时钟周期, 使用 MEMC_CLKDELAY (命令延迟, 时钟输出不延迟)。



3.5.7 MEMC_DYNAMICTRP

- 偏移地址：0x030
- 操作类型：R/W
- 复位值：0xF
- 复位方式：h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	tRP	Precharge 命令周期。 4'h0~4'hF: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 4'hF: 16 个时钟 cycle。

3.5.8 MEMC_DYNAMICTRAS

- 偏移地址：0x034
- 操作类型：R/W
- 复位值：0xF
- 复位方式：h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	tRAS	Active 到 Precharge 命令周期（配置应大于等于 3）。 4'h0~4'hF: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 4'hF: 16 个时钟 cycle。

3.5.9 MEMC_DYNAMICTSREX

- 偏移地址：0x038
- 操作类型：R/W
- 复位值：0x7F
- 复位方式：h

比特	名称	描述
[31:7]	Reserved	保留。



比特	名称	描述
[6:0]	tSREX	自我刷新的退出时间。 7'h0~7'h7F: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 7'h7F: 128 个时钟 cycle。

3.5.10 MEMC_DYNAMICICTWR

- 偏移地址: 0x044
- 操作类型: R/W
- 复位值: 0xF
- 复位方式: h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	tWR	写恢复时间。 4'h0~4'hF: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 4'hF: 16 个时钟 cycle。

3.5.11 MEMC_DYNAMICITRC

- 偏移地址: 0x048
- 操作类型: R/W
- 复位值: 0x1F
- 复位方式: h

比特	名称	描述
[31:5]	Reserved	保留。
[4:0]	tRC	Active 到 Active 命令周期。 5'h0~5'h1F: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 5'h1F: 32 个时钟 cycle。

3.5.12 MEMC_DYNAMICITRFC

- 偏移地址: 0x04C
- 操作类型: R/W
- 复位值: 0x1F



- 复位方式: h

比特	名称	描述
[31:5]	Reserved	保留。
[4:0]	tRFC	自动刷新到其他命令周期。 5'h0~5'h1F: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 5'h1F: 32 个时钟 cycle。

3.5.13 MEMC_DYNAMICIXSR

- 偏移地址: 0x050
- 操作类型: R/W
- 复位值: 0xFF
- 复位方式: h

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	tXSR	退出 self-refresh 到 Active 命令的周期。 8'h0~8'hFF: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 8'hFF: 256 个时钟 cycle。

3.5.14 MEMC_DYNAMICITRRD

- 偏移地址: 0x054
- 操作类型: R/W
- 复位值: 0xF
- 复位方式: h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	tRRD	Active bank A 到 Active bank B 的延时。 4'h0~4'hF: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 4'hF: 16 个时钟 cycle。

3.5.15 MEMC_DYNAMICITMRD

- 偏移地址: 0x058



- 操作类型: R/W
- 复位值: 0xF
- 复位方式: h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	tMRD	Load mode register 到 Active 命令的延时。 4'h0~4'hF: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 4'hF: 16 个时钟 cycle。

3.5.16 MEMC_DYNAMICICTCDLR

- 偏移地址: 0x05C
- 操作类型: R/W
- 复位值: 0xF
- 复位方式: h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	tCDLR	最后一个数据输入到读命令的延时。 4'h0~4'hF: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 4'hF: 16 个时钟 cycle。

3.5.17 MEMC_STATICEXTENDEDWAIT

- 偏移地址: 0x080
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h

比特	名称	描述
[31:10]	Reserved	保留。
[9:0]	EXTENDED WAIT	10'h0~10'h3FF: (n+1) × 16 个时钟 cycle, n 表示对应的十进制值。 例如 10'h0: 16 个时钟 cycle。 建议在系统初始化时或没有电流时修改此值, 但如果需要的话, 也可以在正常操作时修改。



3.5.18 MEMC_DYNAMICCONFIG0

- 偏移地址：0x100
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h

比特	名称	描述
[31:21]	Reserved	保留。
[20]	P	写保护。 0: 没有写保护; 1: 写保护。
[19:15]	Reserved	保留。
[14:7]	AM	地址映射。具体请参见表 3-4~表 3-7。
[6:3]	Reserved	保留。
[2:0]	MD	存储器设备类型。 只能写下列值: 000: SDR-SDRAM; 010: low-power SDR-SDRAM。

AM 配置所对应的地址映射的详细内容如表 3-4~表 3-7 所示。

表3-4 16bit 外部总线地址映射 (Row, Bank, Column)

[14]	[13:12]	[11:9]	[8:7]	描述
0	00	000	00	16Mbits (2M×8bits) , 2Banks, row length=11, column length=9
0	00	000	01	16Mbits (1M×16bits) , 2Banks, row length=11, column length=8
0	00	001	00	64Mbits (8M×8bits) , 4Banks, row length=12, column length=9
0	00	001	01	64Mbits (4M×16bits) , 4Banks, row length=12, column length=8
0	00	010	00	128Mbits (16M×8bits) , 4Banks, row length=12, column length=10
0	00	010	01	128Mbits (8M×16bits) , 4Banks, row length=12, column length=9



[14]	[13:12]	[11:9]	[8:7]	描述
0	00	011	00	256Mbits (32M × 8bits), 4Banks, row length=13, column length=10
0	00	011	01	256Mbits (16M × 16bits), 4Banks, row length=13, column length=9
0	00	100	00	512Mbits (64M × 8bits), 4Banks, row length=13, column length=11
0	00	100	01	512Mbits (32M × 16bits), 4Banks, row length=13, column length=10

表3-5 16bit 外部总线, Low-power SDRAM 地址映射 (Bank, Row, Column)

[14]	[13:12]	[11:9]	[8:7]	描述
0	01	000	00	16Mbits (2M × 8bits), 2Banks, row length=11, column length=9
0	01	000	01	16Mbits (1M × 16bits), 2Banks, row length=11, column length=8
0	01	001	00	64Mbits (8M × 8bits), 4Banks, row length=12, column length=9
0	01	001	01	64Mbits (4M × 16bits), 4Banks, row length=12, column length=8
0	01	010	00	128Mbits (16M × 8bits), 4Banks, row length=12, column length=10
0	01	010	01	128Mbits (8M × 16bits), 4Banks, row length=12, column length=9
0	01	011	00	256Mbits (32M × 8bits), 4Banks, row length=13, column length=10
0	01	011	01	256Mbits (16M × 16bits), 4Banks, row length=13, column length=9
0	01	100	00	512Mbits (64M × 8bits), 4Banks, row length=13, column length=11
0	01	100	01	512Mbits (32M × 16bits), 4Banks, row length=13, column length=10



表3-6 32bit 外部总线地址映射 (Row, Bank, Column)

[14]	[13:12]	[11:9]	[8:7]	描述
1	00	000	00	16Mbits (2M × 8bits), 2Banks, row length=11, column length=9
1	00	000	01	16Mbits (1M × 16bits), 2Banks, row length=11, column length=8
1	00	001	00	64Mbits (8M × 8bits), 4Banks, row length=12, column length=9
1	00	001	01	64Mbits (4M × 16bits), 4Banks, row length=12, column length=8
1	00	010	00	128Mbits (16M × 8bits), 4Banks, row length=12, column length=10
1	00	010	01	128Mbits (8M × 16bits), 4Banks, row length=12, column length=9
1	00	011	00	256Mbits (32M × 8bits), 4Banks, row length=13, column length=10
1	00	011	01	256Mbits (16M × 16bits), 4Banks, row length=13, column length=9
1	00	100	00	512Mbits (64M × 8bits), 4Banks, row length=13, column length=11
1	00	100	01	512Mbits (32M × 16bits), 4Banks, row length=13, column length=10

表3-7 32bit 外部总线 Low-power SDRAM 地址映射 (Bank, Row, Column)

[14]	[13:12]	[11:9]	[8:7]	描述
1	01	000	00	16Mbits (2M × 8bits), 2Banks, row length=11, column length=9
1	01	000	01	16Mbits (1M × 16bits), 2Banks, row length=11, column length=8
1	01	001	00	64Mbits (8M × 8bits), 4Banks, row length=12, column length=9
1	01	001	01	64Mbits (4M × 16bits), 4Banks, row length=12, column length=8
1	01	010	00	128Mbits (16M × 8bits), 4Banks, row length=12, column length=10



[14]	[13:12]	[11:9]	[8:7]	描述
1	01	010	01	128Mbits (8M × 16bits), 4Banks, row length=12, column length=9
1	01	011	00	256Mbits (32M × 8bits), 4Banks, row length=13, column length=10
1	01	011	01	256Mbits (16M × 16bits), 4Banks, row length=13, column length=9
1	01	100	00	512Mbits (64M × 8bits), 4Banks, row length=13, column length=11
1	01	100	01	512Mbits (32M × 16bits), 4Banks, row length=13, column length=10

3.5.19 MEMC_DYNAMICRASCAS0

- 偏移地址: 0x104
- 操作类型: R/W
- 复位值: 0x783
- 复位方式: h

比特	名称	描述
[31:11]	Reserved	保留。
[10:7]	CAS	CAS 延时, 即 CL, 单位为 cycle。 0000: 保留; 1000: 4; 0001: 0.5; 1001: 4.5; 0010: 1; 1010: 5; 0011: 1.5; 1011: 5.5; 0100: 2; 1100: 6; 0101: 2.5; 1101: 6.5; 0110: 3; 1110: 7; 0111: 3.5; 1111: 7.5。
[6:4]	Reserved	保留。
[3:0]	RAS	RAS 延时, 即激活到读或写的延时。 4'h0: 保留; 4'h1~4'hF: n cycle, n 表示对应的十进制值。 例如 4'h3: 3cycle。 配置应比 tRAS 小。



3.5.20 MEMC_STIATICCONFIG0~3

- 偏移地址：0x200、0x220、0x240、0x260。
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h

比特	名称	描述
[31:9]	Reserved	保留。
[8]	EW	扩展等待。 0：扩展等待不使能； 1：扩展等待使能。
[7]	PB	Byte lane state。 0：读写字节选择； 1：MEMC_BLS_N[1:0]的相应位为低表示读； MEMC_BLS_N[1:0]的相应位为高表示写。 片选 1 的上电复位值 0。
[6]	PC	片选的极性。 0：片选低有效； 1：片选高有效。 片选 1 的上电复位值 0。
[5:4]	Reserved	保留。
[3]	PM	Page mode。 0：Disable； 1：Asynchronous page mode enabled（page length four）。 异步 page 模式支持 burst four（或更高）设备。不支持 burst two 的设备，且只能正常访问。
[2]	Reserved	保留。
[1:0]	MW	Memory width。 00：8 位（片选 0、2、3 的上电复位值）； 01：16 位； 10：32 位； 11：Reserved。 片选 1 的上电复位值由管脚信号 BOOTSEL0、1 决定。



3.5.21 MEMC_STIATICWAITWEN0~3

- 偏移地址：0x204、0x224、0x244、0x264
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	WAITWEN	等待写使能，即从片选到写使能的延迟。 4'h0：在片选有效和写使能之间有 1 个 HCLK cycle 的延时。 4'h1~4'hF：(n+1) 个 HCLK cycle 的延迟，n 表示对应的十进制值。

注：建议在系统初始化和没有电流时配置，可以通过在 MEMC 处于 IDLE 状态时配置来保证。

3.5.22 MEMC_STIATICWAITOPEN0~3

- 偏移地址：0x208、0x228、0x248、0x268
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	WAITOEN	等待输出使能，即从片选或地址变化到输出使能的延迟。 4'h0：无延时； 4'h1~4'hF：n 个 HCLK cycle 的延迟，n 表示对应的十进制值。

3.5.23 MEMC_STIATICWAITRD0~3

- 偏移地址：0x20C、0x22C、0x24C、0x26C
- 操作类型：R/W
- 复位值：0x1F
- 复位方式：h

比特	名称	描述
[31:5]	Reserved	保留。



比特	名称	描述
[4:0]	WAITRD	等待读访问，即从片选到读访问的延迟。 Non page mode or asynchronous page mode read, 第 1 次读的延迟。 5'h0~5'h1E: (n+1) HCLK cycles for read access. n 表示对应的十进制值。 例如 5'h1F: 32 HCLK cycles for read access.

3.5.24 MEMC_STIATICWAITPAGE0~3

- 偏移地址: 0x210、0x230、0x250、0x270
- 操作类型: R/W
- 复位值: 0x1F
- 复位方式: h

比特	名称	描述
[31:5]	Reserved	保留。
[4:0]	WAITPAGE	异步 page 模式中第 1 个读后面的读等待 cycle 数。 5'h0~5'h1E: (n+1) HCLK cycles read access time; n 表示对应的十进制值。 例如 5'h1F: 32 HCLK cycles read access time.

3.5.25 MEMC_STIATICWAITWR0~3

- 偏移地址: 0x214、0x234、0x254、0x274
- 操作类型: R/W
- 复位值: 0x1F
- 复位方式: h

比特	名称	描述
[31:5]	Reserved	保留。
[4:0]	WAITWR	从片选到写访问的延时。 5'h0~5'h1E: (n+2) HCLK cycles write access time; n 表示对应的十进制值。 例如 5'h1F: 33 HCLK cycles write access time.



3.5.26 MEMC_STIATICWAITTURN0~3

- 偏移地址：0x218、0x238、0x258、0x278
- 操作类型：R/W
- 复位值：0xF
- 复位方式：h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	WAITTURN	总线 turnaround 的 cycles 数。 4'h0~4'hE: (n+2) HCLK cycles ; n 表示对应的十进制值。 例如 4'hF: 16 HCLK cycles。

3.5.27 MEMC_AHBCONTROL0~4

这 5 个寄存器都是 1 位，可读可写，用于控制 AHB 的接口操作。这些寄存器在正常操作时可以更改。

说明

MEMC 有 5 个 AHB 端口 0~4，分别对应 5 层 AHB 总线：CLCD、EXPS、DMA_M、ARMD、ARMI。

- 偏移地址：0x400、0x420、0x440、0x460、0x480
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h

比特	名称	描述
[31:1]	Reserved	保留。
[0]	E	Buffer enable, 对 AHB 端口字节、半字 Buffer 进行使能。 该寄存器可以在正常操作时配置。 0: Buffer 不使能; 1: Buffer 使能。

3.5.28 MEMC_AHBSTATUS0~4

这 5 个寄存器都是 1 位，只读，用于提供 AHB 的接口状态信息。

- 偏移地址：0x404、0x424、0x444、0x464、0x484
- 操作类型：R



- 复位值: 0x0
- 复位方式: h

比特	名称	描述
[31:2]	Reserved	保留。
[1]	S	Buffer 状态标志位。 0: Buffer 空; 1: Buffer 有数据。
[0]	Reserved	保留。

3.5.29 MEMC_AHBTIMEOUT0~4

这 5 个寄存器均为 10 位，可读可写，用于保证每个 AHB 端口都可以在一个配置好的 cycle 数目之内被服务。当 AHB 请求有效后，该寄存器的值被下载到一个递减计数器中。如果计数器的值为 0 该端口还没有被服务，该端口的优先级就会增加，直到该端口的请求被服务。

- 偏移地址: 0x408、0x428、0x448、0x468、0x488
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h

比特	名称	描述
[31:10]	Reserved	保留。
[9:0]	AHBTIMEOUT	AHB 端口的 timeout。 0: timeout disabled; 1~1023: time out 到达之前的 AHB 时钟 cycle 数目。



4 多端口 DDR SDRAM 存储控制器

关于本章

本章描述内容如下表所示。

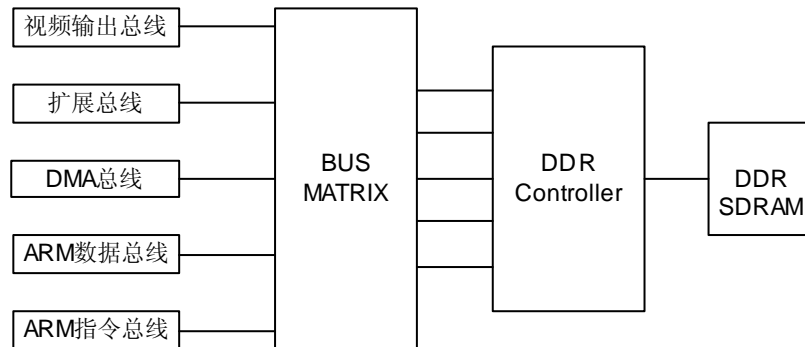
标题	内容
4.1 概述	概述多端口 DDRC。
4.2 特点	列举多端口 DDRC 的特点。
4.3 信号描述	描述多端口 DDRC 的输入输出管脚信号
4.4 寄存器概览	概括介绍 DDRC 的寄存器。
4.5 寄存器描述	详细描述 DDRC 的寄存器。



4.1 概述

多端口 DDR SDRAM 动态存储控制器 (DDRC) 用于提供访问外部 DDR SDRAM 的通道，为系统提供高速的外部存储系统方案。其他单元可以通过该控制器访问芯片外部 DDR SDRAM 动态存储器。图 4-1 为 DDRC 的功能框图。

图4-1 DDRC 功能框图



4.2 特点

DDRC 有以下特点：

- 提供动态 DDR SDRAM 接口，支持最大容量为 128MB（可利用 2 片 8bit 512Mbit 的拼接）、位宽为 16 位的 DDR SDRAM；
- 提供 5 个 AHB Slave 端口访问存储器；
- 提供 1 个寄存器端口，用于配置 DDR SDRAM 的接口时序；
- 支持 DDR SDRAM burst 长度为 2 的配置模式；
- 支持 DDR SDRAM 的 Auto Refresh 和 Self Refresh；
- 控制时钟输出使能以降低 DDR SDRAM 的功耗；
- DDRC 使能控制，当 DDRC 不使能时，可以降低功耗。

4.3 信号描述

本节描述 DDRC 的输入输出管脚信号，如表 4-1 所示。

表4-1 多端口 DDR SDRAM 存储控制器接口信号描述

信号名	方向	功能简述
DDRCKP	O	输出到 DDR SDRAM 的正相时钟信号。



信号名	方向	功能简述
DDRCKN	O	输出到 DDR SDRAM 的反相时钟信号。
DDRRASN	O	输出到 DDR SDRAM 的行地址选通信号，低电平有效，默认高电平。
DDRCASN	O	输出到 DDR SDRAM 的列地址选通信号，低电平有效，默认高电平。
DDRCSN	O	DDR SDRAM 片选信号，低电平有效，默认高电平。
DDRDM1	O	输出到 DDR SDRAM 的数据字节屏蔽信号，对应数据总线 DDRDQ[15:8]。
DDRDM0	O	输出到 DDR SDRAM 的数据字节屏蔽信号，对应数据总线 DDRDQ[7:0]。
DDRWEN	O	DDR SDRAM 写使能信号，低电平有效。
DDRDQ[15:0]	I/O	DDR SDRAM 接口数据线。
DDRCKE	O	DDR SDRAM 接口时钟使能信号，高电平有效。
DDRBA1	O	DDR SDRAM bank 1 选择信号。
DDRBA0	O	DDR SDRAM bank 0 选择信号。
DDRADR[12:0]	O	DDR SDRAM 地址信号。
DDRDQS1	I/O	DDR SDRAM 数据 Strobe 信号，对应数据总线 DDRDQ[15:8]。
DDRDQS0	I/O	DDR SDRAM 数据 Strobe 信号，对应数据总线 DDRDQ[7:0]。
DDRRCVENI	I	DDRC 接收使能输入。
DDRRCVENO	O	DDRC 接收使能输出。

4.4 寄存器概览

DDRC 寄存器的地址位宽 32 位，地址范围：0x1015_0000~0x1015_FFFF。

表4-2 DDRC 寄存器概览（基址是 0x1015_0000）

偏移地址	寄存器名	功能简述	页码
0x000	DDRC_CONTROL	DDRC 控制寄存器	4-5
0x004	DDRC_STATUS	DDRC 状态寄存器	4-6



偏移地址	寄存器名	功能简述	页码
0x008	DDRC_CONFIG	DDRC 配置寄存器	4-6
0x020	DDRC_DYNAMICCONTROL	DDR SDRAM 动态 Memory 控制寄存器	4-7
0x024	DDRC_DYNAMICREFRESH	DDR SDRAM 动态 Memory 刷新寄存器	4-7
0x028	DDRC_DYNAMICREADCONFIG	DDR SDRAM 动态 Memory 读配置寄存器	4-8
0x030	DDRC_DYNAMICTRP	DDR SDRAM 动态 Memory precharge 命令周期寄存器	4-8
0x034	DDRC_DYNAMICTRAS	DDR SDRAM 动态 Memory 激活到预充电命令的周期寄存器	4-9
0x038	DDRC_DYNAMICTSREX	DDR SDRAM 动态 Memory Self-refresh 退出时间寄存器	4-9
0x044	DDRC_DYNAMICTWR	DDR SDRAM 动态 Memory 写恢复时间寄存器	4-9
0x048	DDRC_DYNAMICTRC	DDR SDRAM 动态 Memory 激活到激活时间寄存器	4-10
0x04C	DDRC_DYNAMICTRFC	DDR SDRAM 动态 Memory 自动刷新寄存器	4-10
0x050	DDRC_DYNAMICTXSR	DDR SDRAM 动态 Memory 退出 self-refresh 寄存器	4-11
0x054	DDRC_DYNAMICTRRD	DDR SDRAM 动态 Memory 激活 bank A 到激活 bank B 的时间寄存器	4-11
0x058	DDRC_DYNAMICTMRD	DDR SDRAM 动态 Memory load mode 寄存器	4-11
0x05C	DDRC_DynamictCDLR	DDR SDRAM 动态 Memory 最后一个数据输入到读命令的时间寄存器	4-12
0x100	DDRC_DYNAMICCONFIG0	DDR SDRAM 动态 Memory 配置寄存器 0	4-12
0x104	DDRC_DYNAMICRASCAS0	DDR SDRAM 动态 Memory RAS 及 CAS 延时寄存器 0	4-13
0x400	DDRC_AHBCONTROL0	DDRC AHB 控制寄存器 0	4-14
0x404	DDRC_AHBSTATUS0	DDRC AHB 状态寄存器 0	4-14



偏移地址	寄存器名	功能简述	页码
0x408	DDRC_AHBTIMEOUT0	DDRC AHB Timeout 寄存器 0	4-15
0x420	DDRC_AHBCONTROL1	DDRC AHB 控制寄存器 1	4-14
0x424	DDRC_AHBSTATUS1	DDRC AHB 状态寄存器 1	4-14
0x428	DDRC_AHBTIMEOUT1	DDRC AHB Timeout 寄存器 1	4-15
0x440	DDRC_AHBCONTROL2	DDRC AHB 控制寄存器 2	4-14
0x444	DDRC_AHBSTATUS2	DDRC AHB 状态寄存器 2	4-14
0x448	DDRC_AHBTIMEOUT2	DDRC AHB Timeout 寄存器 2	4-15
0x460	DDRC_AHBCONTROL3	DDRC AHB 控制寄存器 3	4-14
0x464	DDRC_AHBSTATUS3	DDRC AHB 状态寄存器 3	4-14
0x468	DDRC_AHBTIMEOUT3	DDRC AHB Timeout 寄存器 3	4-15
0x480	DDRC_AHBCONTROL4	DDRC AHB 控制寄存器 4	4-14
0x484	DDRC_AHBSTATUS4	DDRC AHB 状态寄存器 4	4-14
0x488	DDRC_AHBTIMEOUT4	DDRC AHB Timeout 寄存器 4	4-15

4.5 寄存器描述

本节详细描述了 DDRC 的寄存器。

4.5.1 DDRC_CONTROL

- 偏移地址：0x000
- 操作类型：R/W
- 复位值：0x1
- 复位方式：h

比特	名称	描述
[31:3]	Reserved	保留。
[2]	L	模式选择。 0：普通模式； 1：低功耗模式。
[1]	Reserved	保留。



比特	名称	描述
[0]	E	DDRC 使能标志位。 0: DDRC disable; 1: DDRC enable。

4.5.2 DDRC_STATUS

- 偏移地址: 0x004
- 操作类型: R
- 复位值: 0x7
- 复位方式: h

比特	名称	描述
[31:3]	Reserved	保留。
[2]	SA	模式选择。 0: 普通模式; 1: self-refresh 模式;
[1]	Reserved	保留。
[0]	B	DDRC 状态标志位。 0: DDRC idle; 1: DDRC busy, 表示正忙于执行存储器的传输命令, 自动刷新或正处于自我刷新模式。

4.5.3 DDRC_CONFIG

- 偏移地址: 0x008
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h

比特	名称	描述
[31:1]	Reserved	保留。
[0]	B	模式选择。 0: little endian 模式; 1: big endian 模式。 Hi3510 中只支持 little endian 模式。



4.5.4 DDRC_DYNAMICCONTROL

- 偏移地址：0x020
- 操作类型：R/W
- 复位值：0xE
- 复位方式：h

比特	名称	描述
[31:9]	Reserved	保留。
[8:7]	I	DDR SDRAM 命令选择。 00: 发出 DDR SDRAM NORMAL 操作命令; 01: 发出 DDR SDRAM MODE 命令; 10: 发出 DDR SDRAM PALL (Precharge all) 命令; 11: 发出 DDR SDRAM NOP 操作命令。
[6]	Reserved	保留。
[5]	MCC	DDRCKP 使能标志位。 0: DDRCKP enable; 1: DDRCKP disable。
[4]	IMCC	DDRCKN 使能标志位。 0: DDRCKN enable; 1: DDRCKN disable。
[3]	SRMCC	DDRCKP 及 DDRCKN 工作状态位。 0: DDRCKP 及 DDRCKN 在 self-refresh 模式下停止; 1: DDRCKP 及 DDRCKN 连续工作。
[2]	SR	模式选择。 0: 普通模式; 1: 进入自我刷新模式。
[1]	CS	动态存储器的时钟控制。 该比特位只能配置为 1。
[0]	CE	动态存储器的时钟使能位。 0: 空闲设备的时钟使能无效, 以此省电; 1: 所有时钟使能一直有效。

4.5.5 DDRC_DYNAMICREFRESH

- 偏移地址：0x024



- 操作类型：R/W
- 复位值：0x0
- 复位方式：h

比特	名称	描述
[31:11]	Reserved	保留。
[10:0]	REFRESH	刷新周期。 11'h0: 刷新 disable; 11'h1~11'h7FF: SDRAM 刷新周期时间为 $16 \times n$ 时钟 cycle, n 表示对应的十进制值。 例如 11'h8: 128 个时钟 cycle (16×8)。

4.5.6 DDRC_DYNAMICREADCONFIG

- 偏移地址：0x028
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h

比特	名称	描述
[31:13]	Reserved	保留。
[12]	DRP	DDR 采读数据的极性。 0: 用 HCLK 的下降沿采读数据; 1: 用 HCLK 的上升沿采读数据。
[11:10]	DRRW	读数据时采样等待周期。
[9:8]	DRD	DDR SDRAM 读数据策略。 00: 保留; 01: 命令延迟策略; 10: 命令延迟策略+1 个时钟周期; 11: 命令延迟策略+2 个时钟周期。
[7:0]	Reserved	保留。

4.5.7 DDRC_DYNAMICTRP

- 偏移地址：0x030
- 操作类型：R/W
- 复位值：0xF



- 复位方式: h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	tRP	Precharge 命令周期。 4'h0~4'hF: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 4'hF: 16 个时钟 cycle。

4.5.8 DDRC_DYNAMICTRAS

- 偏移地址: 0x034
- 操作类型: R/W
- 复位值: 0xF
- 复位方式: h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	tRAS	Active 到 Precharge 命令周期 (配置应大于 3)。 4'h0~4'hF: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 4'hF: 16 个时钟 cycle。

4.5.9 DDRC_DYNAMICTSREX

- 偏移地址: 0x038
- 操作类型: R/W
- 复位值: 0x7F
- 复位方式: h

比特	名称	描述
[31:7]	Reserved	保留。
[6:0]	tSREX	自我刷新的退出时间。 7'h0~7'h7F: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 7'h7F: 128 个时钟 cycle。

4.5.10 DDRC_DYNAMICTWR

- 偏移地址: 0x044



- 操作类型：R/W
- 复位值：0xF
- 复位方式：h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	tWR	写恢复时间。 4'h0~4'hF: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 4'hF: 16 个时钟 cycle。

4.5.11 DDRC_DYNAMICTRC

- 偏移地址：0x048
- 操作类型：R/W
- 复位值：0x1F
- 复位方式：h

比特	名称	描述
[31:5]	Reserved	保留。
[4:0]	tRC	Active 到 Active 命令周期。 5'h0~5'h1F: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 5'h1F: 32 个时钟 cycle。

4.5.12 DDRC_DYNAMICTRFC

- 偏移地址：0x04C
- 操作类型：R/W
- 复位值：0x1F
- 复位方式：h

比特	名称	描述
[31:5]	Reserved	保留。
[4:0]	tRFC	自动刷新到其他命令周期。 5'h0~5'h1F: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 5'h1F: 32 个时钟 cycle。



4.5.13 DDRC_DYNAMICICTXSR

- 偏移地址：0x050
- 操作类型：R/W
- 复位值：0xFF
- 复位方式：h

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	tXSR	退出 self-refresh 到 Active 命令的周期。 8'h0~8'hFF: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 8'hFF: 256 个时钟 cycle。

4.5.14 DDRC_DYNAMICICTRRD

- 偏移地址：0x054
- 操作类型：R/W
- 复位值：0xF
- 复位方式：h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	tRRD	Active bank A 到 Active bank B 的延时。 4'h0~4'hF: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 4'hF: 16 个时钟 cycle。

4.5.15 DDRC_DYNAMICICTMRD

- 偏移地址：0x058
- 操作类型：R/W
- 复位值：0xF
- 复位方式：h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	tMRD	Load mode register 到 Active 命令的延时。 4'h0~4'hF: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 4'hF: 16 个时钟 cycle。



4.5.16 DDRC_DYNAMICICTCDLR

- 偏移地址：0x05C
- 操作类型：R/W
- 复位值：0xF
- 复位方式：h

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	tCDLR	最后一个数据输入到读命令的延时。 4'h0~4'hF: (n+1) 个时钟 cycle, n 表示对应的十进制值。 例如 4'hF: 16 个时钟 cycle。

4.5.17 DDRC_DYNAMICCONFIG0

- 偏移地址：0x100
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h

比特	名称	描述
[31:21]	Reserved	保留。
[20]	P	写保护。 0: 没有写保护; 1: 写保护。
[19:15]	Reserved	保留。
[14:7]	AM	地址映射。详细内容请参见表 4-3。
[6:3]	Reserved	保留。
[2:0]	MD	存储器设备类型。 100: DDR-SDRAM。 必须设置为 3'h4。



AM 配置所对应的地址映射如表 4-3 所示。

表4-3 16bit 外部总线地址映射 (Row, Bank, Column)

[14]	[13:12]	[11:9]	[8:7]	描述
0	00	000	00	16Mbit (2M × 8bits), 2Banks, row length=11, column length=9
0	00	000	01	16Mbit (1M × 16bits), 2Banks, row length=11, column length=8
0	00	001	00	64Mbit (8M × 8bits), 4Banks, row length=12, column length=9
0	00	001	01	64Mbit (4M × 16bits), 4Banks, row length=12, column length=8
0	00	010	00	128Mbit (16M × 8bits), 4Banks, row length=12, column length=10
0	00	010	01	128Mbit (8M × 16bits), 4Banks, row length=12, column length=9
0	00	011	00	256Mbit (32M × 8bits), 4Banks, row length=13, column length=10
0	00	011	01	256Mbit (16M × 16bits), 4Banks, row length=13, column length=9
0	00	100	00	512Mbit (64M × 8bits), 4Banks, row length=13, column length=11
0	00	100	01	512Mbit (32M × 16bits), 4Banks, row length=13, column length=10

4.5.18 DDRC_DYNAMICRASCAS0

- 偏移地址: 0x104
- 操作类型: R/W
- 复位值: 0x783
- 复位方式: h



比特	名称	描述
[31:11]	Reserved	保留。
[10:7]	CAS	CAS 延时，即 CL，单位为 cycle。 0000: 保留； 0001: 0.5; 0010: 1; 0011: 1.5; 0100: 2。
[6:4]	Reserved	保留。
[3:0]	RAS	RAS 延时，即激活到读或写的延时（应配置比 tRAS 小）。 4'h0: 保留； 4'h1~4'hF: n cycles。n 表示对应的十进制值。 例如 4'h3: 3 cycle。

4.5.19 DDRC_AHBCONTROL0~4

这 5 个寄存器都只有 1 位，用于控制 AHB 的接口操作。这些寄存器可以在正常操作时更改。

说明

DDRC 有 AHB 端口 0~4，这 5 个端口分别对应 5 层 AHB 总线：CLCD、EXPS、DMA_M、ARMD、ARMI。

- 偏移地址：0x400、0x420、0x440、0x460、0x480
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h

比特	名称	描述
[31:1]	Reserved	保留。
[0]	E	Buffer enable，对 AHB 端口 Buffer 进行使能。 该寄存器可以在正常操作时配置。 0: Buffer 不使能； 1: Buffer 使能。

4.5.20 DDRC_AHBSTATUS0~4

这 5 个寄存器都只有 1 位，用于提供 AHB 的接口状态信息。

- 偏移地址：0x404、0x424、0x444、0x464、0x484
- 操作类型：R



- 复位值: 0x0
- 复位方式: h

比特	名称	描述
[31:2]	Reserved	保留。
[1]	S	Buffer 状态标志。 0: Buffer 空; 1: Buffer 有数据。
[0]	Reserved	保留。

4.5.21 DDRC_AHBTIMEOUT0~4

这 5 个寄存器均为 10 位，可读可写，用于保证每个 AHB 端口都可以在一个配置好的 cycle 数目之内被服务。当 AHB 请求有效后，该寄存器的值被下载到一个递减计数器中。如果该计数器的值递减到 0 时，该端口还没有被服务，则该端口的优先级就会增加，直到其请求被服务。

- 偏移地址: 0x408、0x428、0x448、0x468、0x488
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h

比特	名称	描述
[31:10]	Reserved	保留。
[9:0]	AHBTIMEOUT	AHB 端口的 timeout。 0: timeout disabled; 1~1023: time out 到达之前的 AHB 时钟 cycle 数目。



5 中断控制器 (VIC)

关于本章

本章描述内容如下表所示。

标题	内容
5.1 概述	概括介绍 VIC。
5.2 特点	简单介绍 VIC 的特点。
5.3 信号描述	描述 VIC 的外部输入输出管脚信号。
5.4 工作方式	描述 VIC 的中断分配。
5.5 寄存器概览	概括介绍 VIC 的寄存器。
5.6 寄存器描述	详细描述 VIC 的寄存器。



5.1 概述

VIC (Vectored Interrupt Controller) 为系统提供中断管理功能。

VIC 除了具有普通中断控制器的中断判断和响应功能外, 还具有向量中断机制, 当多中断源请求中断时, 硬件优先级逻辑决定哪个优先级别的中断先被服务。同时, 硬件逻辑提供对应向量表基地址的偏移值, 它可以通过跳转指令到对应的服务子程序。

因此, 软件程序依靠硬件结构仅需要执行几条指令就可以取得向量表入口, 执行中断服务程序。与纯软件处理相比, 向量中断机制可以减少中断延迟。

5.2 特点

向量中断控制器 VIC 有以下特点:

- 支持 32 个外部中断源或者软件触发中断源, 主要包括:
 - 16 个向量 IRQ 中断源
 - 16 个普通中断源
- 支持快速中断 FIQ 和普通中断 IRQ 输出
- 支持中断屏蔽
- 支持原始中断状态查询和屏蔽后中断状态查询
- 支持多中断源仲裁
- 支持硬件优先级控制: $FIQ > \text{向量 IRQ} > \text{IRQ}$ 。在 IRQ 中断之间没有优先级差别, IRQ 中断的优先级可由软件设置中断屏蔽位来控制。

5.3 信号描述

本节描述 VIC 的外部输入输出管脚信号, 如表 5-1 所示。

表5-1 VIC 接口信号描述

信号名	方向	描述
INTRN	I	外部中断请求输入, 低电平有效。

5.4 工作方式

本节描述了 VIC 的中断分配。

向量中断控制器 VIC 提供 32 个外部中断源或者软件触发中断源, 包括 16 个向量 IRQ 中断源和 16 个普通中断源, 如表 5-2 所示。



表5-2 中断请求分配

中断号	功能	中断号	功能
0	看门狗中断源请求信号	16	VO 中断
1	软件可编程中断	17	DMAC 中断
2	ARM926EJ-S 调试 COMMRx 中断	18	AES 中断
3	ARM926EJ-S 调试 COMMTX 中断	19	GPIO4 中断
4	TIMER1/2 中断	20	GPIO5 中断
5	TIMER3/4 中断	21	GPIO6 中断
6	GPIO0 中断	22	GPIO7 中断
7	GPIO1 中断	23	VI 中断
8	GPIO2 中断	24	DSU 中断
9	GPIO3 中断	25	SIO0 中断
10	RTC 中断	26	DBLK 中断
11	SSP 中断	27	DSP 到 ARM 中断 0
12	UART0 中断	28	DSP 到 ARM 中断 1
13	UART1 中断	29	SIO1 中断
14	I ² C 中断	30	SF 中断
15	USB 中断	31	芯片外部中断

5.5 寄存器概览

VIC 寄存器的地址位宽 32 位，地址范围：0x1014_0000~0x1014_FFFF。

表5-3 VIC 寄存器概览（基址是 0x1014_0000）

偏移地址	名称	描述	页码
0x000	VIC_IRQSTATUS	IRQ Status 寄存器	5-5
0x004	VIC_FIQSTATUS	FIQ Status 寄存器	5-5
0x008	VIC_RAWINTR	Raw Interrupt Status 寄存器	5-6
0x00C	VIC_INTSELECT	Interrupt Select 寄存器	5-6
0x010	VIC_INTENABLE	Interrupt Enable 寄存器	5-6



偏移地址	名称	描述	页码
0x014	VIC_INTENCLEAR	Interrupt Enable Clear 寄存器	5-7
0x018	VIC_SOFTINT	software Interrupt 寄存器	5-7
0x01C	VIC_SOFTINTCLEAR	Software Interrupt Clear 寄存器	5-8
0x020	VIC_PROTECTION	Protection Enable 寄存器	5-8
0x024	VIC_VECTADDR	Vector Address 寄存器	5-9
0x028	VIC_DEFVECTADDR	Default Vector Address 寄存器	5-9
0x100	VIC_VECTADDR0	Vector Address 寄存器 0	5-9
0x104	VIC_VECTADDR1	Vector Address 寄存器 1	5-9
0x108	VIC_VECTADDR2	Vector Address 寄存器 2	5-9
0x10C	VIC_VECTADDR3	Vector Address 寄存器 3	5-9
0x110	VIC_VECTADDR4	Vector Address 寄存器 4	5-9
0x114	VIC_VECTADDR5	Vector Address 寄存器 5	5-9
0x118	VIC_VECTADDR6	Vector Address 寄存器 6	5-9
0x11C	VIC_VECTADDR7	Vector Address 寄存器 7	5-9
0x120	VIC_VECTADDR8	Vector Address 寄存器 8	5-9
0x124	VIC_VECTADDR9	Vector Address 寄存器 9	5-9
0x128	VIC_VECTADDR10	Vector Address 寄存器 10	5-9
0x12C	VIC_VECTADDR11	Vector Address 寄存器 11	5-9
0x130	VIC_VECTADDR12	Vector Address 寄存器 12	5-9
0x134	VIC_VECTADDR13	Vector Address 寄存器 13	5-9
0x138	VIC_VECTADDR14	Vector Address 寄存器 14	5-9
0x13C	VIC_VECTADDR15	Vector Address 寄存器 15	5-9
0x200	VIC_VECTCNTL0	Vector Control 寄存器 0	5-10
0x204	VIC_VECTCNTL1	Vector Control 寄存器 1	5-10
0x208	VIC_VECTCNTL2	Vector Control 寄存器 2	5-10
0x20C	VIC_VECTCNTL3	Vector Control 寄存器 3	5-10
0x210	VIC_VECTCNTL4	Vector Control 寄存器 4	5-10
0x214	VIC_VECTCNTL5	Vector Control 寄存器 5	5-10
0x218	VIC_VECTCNTL6	Vector Control 寄存器 6	5-10



偏移地址	名称	描述	页码
0x21C	VIC_VECTCNTL7	Vector Control 寄存器 7	5-10
0x220	VIC_VECTCNTL8	Vector Control 寄存器 8	5-10
0x224	VIC_VECTCNTL9	Vector Control 寄存器 9	5-10
0x228	VIC_VECTCNTL10	Vector Control 寄存器 10	5-10
0x22C	VIC_VECTCNTL11	Vector Control 寄存器 11	5-10
0x230	VIC_VECTCNTL12	Vector Control 寄存器 12	5-10
0x234	VIC_VECTCNTL13	Vector Control 寄存器 13	5-10
0x238	VIC_VECTCNTL14	Vector Control 寄存器 14	5-10
0x23C	VIC_VECTCNTL15	Vector Control 寄存器 15	5-10

5.6 寄存器描述

本节详细描述了 VIC 的寄存器。

5.6.1 VIC_IRQSTATUS

该寄存器为 IRQ 屏蔽后的中断寄存器，显示经过寄存器 VIC_INTENABLE 和 VIC_INTSELECT 屏蔽后的中断状态。

- 偏移地址：0x000
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:0]	IRQStatus	某位对应的中断状态。 0: 对应的 IRQ 屏蔽后的中断输入无效。 1: 对应的 IRQ 屏蔽后的中断输入有效，并向处理器发出 IRQ 中断。

5.6.2 VIC_FIQSTATUS

该寄存器为 FIQ 屏蔽后的中断寄存器，显示经过寄存器 VIC_INTENABLE 和 VIC_INTSELECT 屏蔽后的中断状态。

- 偏移地址：0x004



- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	FIQStatus	某位为高电平时表示对应的中断输入有效, 并向处理器发出 FIQ 中断。

5.6.3 VIC_RAWINTR

该寄存器显示屏蔽前的中断状态。

- 偏移地址: 0x008
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	RawInterrupt	某位为高电平时表示对应的屏蔽前中断输入有效。

5.6.4 VIC_INTSELECT

该寄存器为中断选择寄存器。

- 偏移地址: 0x00C
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	IntSelect	该寄存器的每位选择对应的中断源是生成一个 IRQ 中断还是生成一个 FIQ 中断。 0: IRQ 中断; 1: FIQ 中断。

5.6.5 VIC_INTENABLE

该寄存器为中断使能寄存器。

- 偏移地址: 0x010
- 操作类型: R/W



- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	IntEnable	<p>读该寄存器时, 返回的是各中断源的屏蔽状态。</p> <p>0: 被屏蔽;</p> <p>1: 未被屏蔽。</p> <p>写该寄存器时, 其作用是按位使能 IRQ 中断源。</p> <p>0: 对应位被清零, 对应的中断源被屏蔽;</p> <p>1: 对应位被置位, 对应的中断源的屏蔽被打开。</p> <p>复位时, 由于 VIC_INTENABLE 的值变为 0x0, 故所有中断源都被屏蔽。</p>

5.6.6 VIC_INTENCLEAR

该寄存器为中断清除寄存器。

- 偏移地址: 0x014
- 操作类型: W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	IntEnable Clear	<p>读该寄存器时, 返回值为 0。</p> <p>写该寄存器时, 其作用是按位屏蔽 IRQ 中断源。</p> <p>0: 对应位的当前值不受影响;</p> <p>1: 对应位被清零, 对应的中断源被屏蔽。</p>

5.6.7 VIC_SOFTINT

该寄存器为屏蔽前软中断寄存器。

- 偏移地址: 0x018
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s



比特	名称	描述
[31:0]	SoftInt	在指定的中断源上产生一个屏蔽前软中断。 0: 对应位不受影响; 1: 对应位置位, 产生一个软中断。

5.6.8 VIC_SOFTINTCLEAR

该寄存器为软中断清除寄存器。

- 偏移地址: 0x01C
- 操作类型: W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	SoftIntClear	将 VIC_SOFTINT 寄存器的特定位清零。 0: 寄存器 VIC_SOFTINT 的对应位不受影响; 1: 将寄存器 VIC_SOFTINT 的对应位清零。

5.6.9 VIC_PROTECTION

该寄存器为保护控制寄存器。

- 偏移地址: 0x020
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:1]	Reserved	保留。读时返回 0。写时无影响。
[0]	Protection	使能/去掉寄存器的访问保护。 0: 去掉寄存器访问保护, AMBA 总线采用特权模式和用户模式 (user mode) 都可以访问 VIC 的寄存器。 1: 使能寄存器访问保护, 只有 AMBA 总线采用特权模式 (privileged mode) 才能访问 VIC 的寄存器。 复位时该寄存器被清零。用户模式或特权模式都可访问 VIC 的寄存器。 当 AMBA 总线的主设备无法产生正确的保护信息时 (HPROT), 让该寄存器处于复位后的状态即可允许在用户模式下访问 VIC 的寄存器。



5.6.10 VIC_VECTADDR

该寄存器为中断地址寄存器。

- 偏移地址: 0x024
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	VectorAddr	包含当前等待服务的中断服务程序 (ISR) 的入口地址。 <ul style="list-style-type: none"> • 读该寄存器可得到 ISR 的地址, 同时通知中断优先级仲裁硬件当前中断正在被服务。 • 写该寄存器则通知中断优先级仲裁硬件对当前中断的服务已经完成。

5.6.11 VIC_DEFVECTADDR

该寄存器为缺省中断地址寄存器。

- 偏移地址: 0x028
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	Default VectorAddr	包含缺省的 ISR 地址。

5.6.12 VIC_VECTADDR0~15

该寄存器为矢量中断地址寄存器。

- 偏移地址: 0x100~0x13C
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	VectorAddr 0~15	这 16 个寄存器包含了 16 个矢量 IRQ 中断各自对应的 ISR 地址。



5.6.13 VIC_VECTCNTL0~15

该寄存器为矢量中断控制寄存器。

- 偏移地址：0x200~0x23C
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:6]	Reserved	保留。
[5]	E	矢量中断屏蔽位。 <ul style="list-style-type: none"> • 写入时： <ul style="list-style-type: none"> 0：将由 IntSource 字段选定的中断请求屏蔽； 1：允许由 IntSource 字段选定的中断请求通过。 • 复位时该位被清零。
[4:0]	IntSource	从 32 个 IRQ 中断源中选择一个作为本矢量中断模块的矢量中断输出。 VIC 中包含 16 个矢量中断模块，故可产生 16 个矢量中断。



6 时钟、复位和系统控制器

关于本章

本章描述内容如下表所示。

标题	内容
6.1 概述	概括介绍时钟、复位和系统控制器。
6.2 功能描述	概括介绍时钟、复位和系统控制器的功能。
6.3 寄存器概览	概括介绍系统控制器的寄存器。
6.4 寄存器描述	详细描述系统控制器的寄存器。



6.1 概述

时钟、复位单元与系统控制器一起配合为内部各模块提供时钟、控制系统时钟的切换、复位控制等功能。系统控制器提供了控制系统运行接口，用于控制系统运行模式和时钟频率，监控系统运行状态。

6.2 功能描述

本节主要描述时钟、复位单元和系统控制器的功能特点。

6.2.1 时钟

时钟单元有以下特点：

- 支持 4 个外部时钟输入，其中 2 个时钟可选；
- 2 个管脚，可复用为时钟输出；
- 内部集成 2 个 PLL；
- 2 个复位输入；
- 提供多个模块的时钟使能控制；
- 提供全局软复位和多个模块的模块软复位；
- 支持在 32.768kHz、27MHz 和内部 PLL 倍频时钟间进行时钟切换。

外部输入的时钟

Hi3510 支持 4 个外部时钟输入：

- 32.768kHz 晶振时钟
- 27MHz 晶振时钟（系统主时钟）
- 27MHz 晶振时钟（视频输出时钟，可选）
- 48MHz 晶振时钟（USB 时钟，可选）

这 4 个外部输入的时钟比较如表 6-1 所示。

表6-1 4 个外部输入的时钟比较

时钟	功能
32.768kHz 晶振时钟	<ul style="list-style-type: none"> • 睡眠模式下作为系统时钟源，其余时钟均被关闭以降低系统功耗； • 作为 RTC 的时钟源； • 作为 Watch Dog 或 TIMER 的计数时钟使能信号； • 复位信号的去抖。



时钟	功能
27MHz 晶振时钟 (系统主时钟)	当系统处于 SLOW 模式时，作为 ARM 子系统和 DSP 子系统的时钟源，PLL 可以被关闭以降低系统功耗； 为 PLL 提供时钟源，Hi3510 集成 2 个 PLL（倍频系数可编程），时钟频率如下： <ul style="list-style-type: none"> • ARM PLL：合成时钟频率最高为 230MHz； • DSP PLL：合成时钟频率最高为 160MHz。
27MHz 晶振时钟 (视频输出时钟)	27MHz 晶振时钟作为 VOU 和 VDAC 器件的时钟（可选）。 当采用主晶振（此时 XIN2 接 27MHz 晶振）作为 VOU 时钟时，VOCK 被配置为输出时钟，为 VDAC 器件提供时钟。
48MHz 晶振时钟 (USB 时钟)	48MHz 晶振时钟作为 USB1.1 Host 的参考时钟（可选）。 当采用片内分频方案产生 USB 时钟时，可不接外部晶振。

时钟源选择和控制

时钟模块有以下功能：

- 根据设定的系统模式（SLEEP、DOZE、SLOW、NORMAL），选择 32.768kHz 晶振时钟、27MHz 晶振时钟和 PLL 输出时钟之一作为 ARM926EJ-S core 时钟、AHB 总线时钟和其他 ARM 子系统时钟源；
- 根据系统模式（SLEEP、DOZE、SLOW、NORMAL），选择 32.768kHz 晶振时钟、27MHz 晶振时钟和 PLL 输出时钟之一作为 DSP core 时钟、DSP 子系统时钟源；
- 支持外设时钟关断，以降低系统功耗。

说明

ARM926EJ-S core 时钟频率和 AHB 总线时钟频率固定为 2:1 关系，由 FUNCSEL 管脚确定，其中 FUNCSEL2 为 1'h0，表示 ARM926EJ-S core 时钟与 AHB 总线时钟频率为 2:1。

时钟接口信号

时钟接口信号如表 6-2 所示。

表6-2 时钟接口信号描述

信号名	方向	描述
XIN1	I	32.768kHz 晶振时钟输入。
XOUT1	O	32.768kHz 晶振时钟输出。当不接 VOCK 晶振时钟输入时，可接 27MHz 晶振时钟输入。
XIN2	I	27MHz 晶振时钟输入。
XOUT2	O	27MHz 晶振时钟输出，可选。



信号名	方向	描述
XIN3	I	48MHz 晶振时钟输入。
XOUT3	O	48MHz 晶振时钟输出。
VOCK	I/O	VOU 时钟输入输出。 当作为输入时，Hi3510 的 VOU 模块 27MHz 时钟由外部提供； 当不接外部钟振时钟输入时，可配置为输出时钟，为 VDAC 器件提供时钟。
GPIO3[4]	I/O	可复用为 nVOCLK 输出，当 VOCK 接外部钟振时钟输入时，为 VDAC 提供时钟。 说明： nVOCK 时钟是 VOCK 的备份时钟，其用法和性质和 VOCK 完全相同。

6.2.2 复位

本小节主要介绍复位控制和复位接口信号。

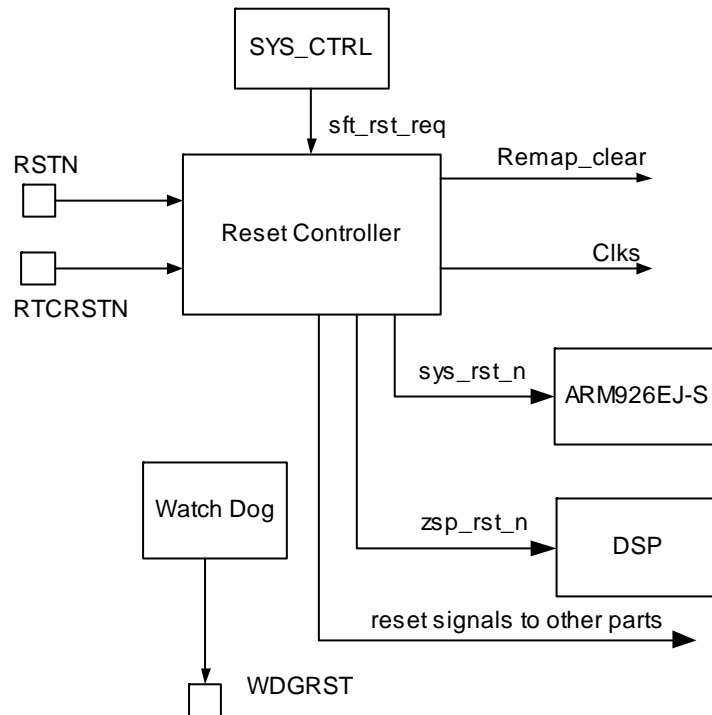
复位控制

复位模块产生系统各种复位信号、外设复位初始化控制信号，有以下特性：

- 对系统上电复位、软复位进行组合，生成系统异步复位信号；
- 对组合后的复位信号进行同步处理，生成 Hi3510 芯片内各模块同步复位以及芯片外部器件复位信号；
- 与系统控制器一起配合生成各模块的软复位信号。

芯片复位和模块复位产生电路如图 6-1 所示。

图6-1 复位信号结构图



Reset/Remap Controller 对以下四种复位信号进行组合：

- 系统上电复位信号 RSTN
- RTC 复位信号 RTCRSTN
- 来自系统控制器的全局软复位信号 sft_rst_req
- 来自系统控制器的外设时钟/复位控制信号

生成各种组合后的异步复位信号，同步处理后得到芯片所需的复位信号。

复位接口信号

复位接口信号如表 6-3 所示。

表6-3 复位单元接口信号描述

信号名	方向	描述
RSTN	I	系统上电复位信号输入，导致全芯片复位，低电平有效。
RTCRSTN	I	RTC 上电复位输入，低电平有效，建议在 PCB 板上与 RSTN 连接同一复位源。
WDGRST	O	看门狗复位输出，低电平有效，OD 输出。



6.2.3 系统控制器

系统控制器具有以下特性：

- 通过状态机控制系统运行模式；
- 晶振控制和 PLL 控制；
- 定义系统的中断响应；
- 复位状态检测和软复位产生；
- 系统地址重映射控制；
- 提供通用外设控制寄存器；
- 系统/外设时钟控制和状态检测。

系统运行模式控制

系统控制器控制系统运行模式和系统时钟源的切换。具体由系统控制器模式控制寄存器 ModeCtrl (SC_CTRL[2:0]) 配置：

- 000：系统切换到 SLEEP 状态。
- 001：系统切换到 DOZE 状态；
- 01X：系统切换到 SLOW 状态；
- 1XX：系统切换到 NORMAL 状态；

说明

上述寄存器比特值为 X，表示可以为 0 或者 1。

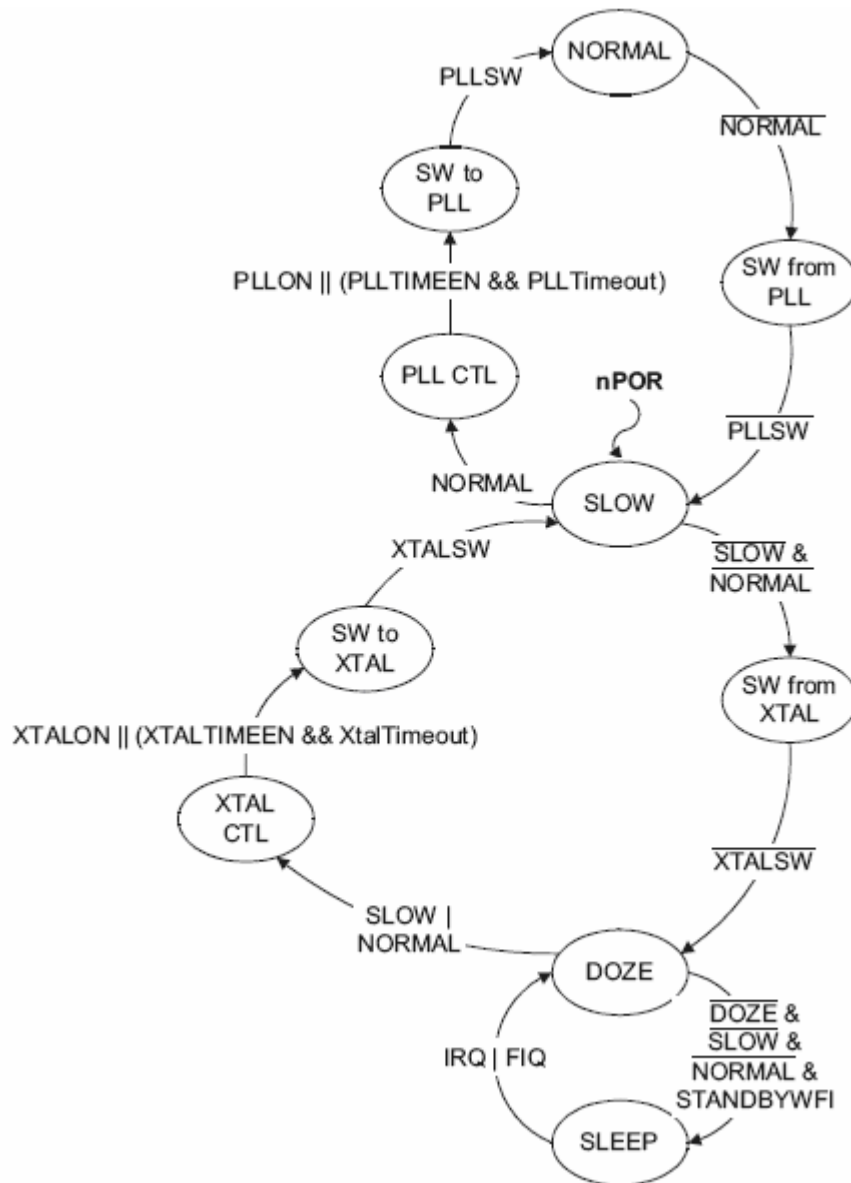
当系统模式被设置后，状态机将控制模式的自动切换，无需软件的干预。当前系统状态可通过读取 SC_CTRL[ModeStatus] 的信息获得。

上电复位后，系统控制器默认处于 SLOW 状态。

中断模式下，当 VIC 接收到中断输入，待切换模式由中断响应模式寄存器指定，而不是由 ModeCtrl 寄存器指定。

各种状态的迁移图如图 6-2 所示。

图6-2 各种状态的迁移图



系统控制器和时钟模块配合完成系统时钟和系统模式的切换。当状态机状态发生迁移时：

1. 系统控制器发出时钟切换指示信号；
2. 时钟模块进行时钟切换，并向系统控制器反馈切换完成指示信号；
3. 系统控制器检测到切换完成指示信号，完成模式切换。

系统控制器状态机状态和系统时钟之间的关系如表 6-4 所示。



表6-4 系统控制器状态和时钟切换对应关系表

系统控制器状态	32.768kHz晶振使能状态	27MHz晶振使能状态	PLL 使能状态	系统时钟状态
NORMAL	使能	使能	使能	ARM 和 DSP 子系统的工作时钟都来自 PLL 输出。
SLOW	使能	使能	不使能	ARM 和 DSP 子系统的工作时钟都来自 27MHz 晶振输入。
DOZE	使能	不使能	不使能	ARM 和 DSP 子系统的工作时钟都来自 32.768kHz 晶振输入。
SLEEP	使能	不使能	不使能	除系统控制器工作在 32.768kHz 外，其他模块的时钟都处于关闭状态。

- SLEEP 模式

在 SLEEP 模式下，除系统控制器时钟由低速 32.768kHz 晶振时钟驱动外，其他模块的时钟都被关闭。

该模式下当有 FIQ 或者 IRQ 中断发生时，系统迁移到 DOZE 状态，并且 ModeCtrl 寄存器的值由 SLEEP 对应的 000 自动更新为 DOZE 对应的 001。

- DOZE 模式

在 DOZE 模式下，系统时钟和系统控制器时钟由 32.768kHz 晶振驱动。

该模式可能发生的状态迁移有：

- 如果 ModeCtrl 被设置为 SLOW 模式或者 NORMAL 模式，系统将进入晶振控制状态 XTAL CTL，打开晶振使能 (XTALEN = 1)，对 27MHz 时钟晶振进行初始化。当晶振稳定后，系统迁移到 SW to XTAL 状态，将系统时钟从 32.768kHz 切换到 27MHz 时钟，切换完成后 (XTALSW=1)，进入 SLOW 模式；

说明

系统控制器 SC_XTALCTRL[18:3]定义了 27MHz 晶振的稳定时间，当晶振被使能时，超时计数器开始计数，用户可通过查询 SC_XTALCTRL[2]判断 27MHz 时钟是否已经稳定。

- 如果 ModeCtrl 被设置为 SLEEP 模式，并且 ARM926EJ-S 处于 wait-for-interrupt 状态，系统进入 SLEEP 模式。用户可通过设置 ARM926EJ-S 系统控制协处理器 CP15 R7，使处理器进入低功耗状态。

- SLOW 模式

在 SLOW 模式下，ARM 子系统和 DSP 子系统都工作于 27MHz 时钟。

该模式下可能发生的状态迁移有：

- 如果 ModeCtrl 被设为 NORMAL 模式，系统将进入 PLL 控制状态 PLL CTL，使能 PLL (PLLEN = 1)。当 PLL 稳定后，系统进入 SW TO PLL 状态，将系统时钟切换到 PLL 时钟，切换完成后 (PLLSW=1)，进入 NORMAL 模式；



说明

系统控制器 SC_PLLCTRL[27:3]定义了PLL的稳定时间，当PLL被使能时，超时计数器开始计数，用户可通过查询SC_PLLCTRL[2]判断PLL是否已经稳定。

- 如果 ModeCtrl 被设为比 SLOW 更低的模式 (DOZE 或者 SLEEP)，系统迁移到 SW FROM XTAL 状态，将系统时钟切换到 32.768kHz，切换完成后 (XTALSW=0)，进入 DOZE 模式。

- NORMAL 模式

在 NORMAL 模式下，ARM 子系统和 DSP 子系统工作于 2 个 PLL 的输出时钟。

该模式下，如果 ModeCtrl 被设为非 NORMAL 模式，系统迁移到 SW FROM PLL 状态，将系统时钟切换到 27MHz 时钟，切换完成后 (PLLSW=0)，进入 SLOW 模式。

晶振和 PLL 控制

系统控制器系统状态机可用于控制外部晶振时钟以及片内 PLL 的使能，各种模式下晶振状态和 ARM PLL 的状态请参见表 6-4。

PLL 频率控制

系统控制器集成了 2 个 PLL 频率控制寄存器，分别用于定义 PLL 的倍频系数。具体请参见“6.4 寄存器描述”。

中断响应模式

中断响应模式用于定义中断发生后系统状态机所处的模式。中断响应模式由中断模式控制寄存器组进行控制，该组寄存器定义了如下功能：

- 中断响应模式是否使能；
- 中断发生后系统状态机的模式；
- 触发中断响应模式的中断类型是 FIQ 还是 IRQ；
- 中断模式状态查询和清除机制。



注意

- 中断响应模式只支持系统运行频率从低速切换到高速，例如从 DOZE 模式切换到 NORMAL；
- 中断响应模式不支持系统运行频率从高速切换到低速，例如从 NORMAL 模式切换到 SLOW 模式。

复位控制

系统控制器支持对芯片全局以及局部模块进行软复位（请参见“6.4.2 SC_SYSSTAT”和“6.4.8 SC_PERCTRL0”的描述），功能描述如下：

- 可触发系统软复位；



- DSP 子系统软复位控制和复位状态检测；
- 其他各模块单独的软复位控制。

系统地址重映射控制

系统控制器提供地址重映射控制信号，支持地址译码单元对系统存储地址空间进行重新映射和分配。上电复位后，芯片的 0 地址对应 MEMC 片选 1 所占的地址空间；可通过系统控制器提供的 Remap 清除地址重映射。

Watch Dog 和 TIMER 时钟使能控制

时钟使能可以使计数频率独立于系统时钟频率，即使系统时钟发生改变，计数器仍会保持固定的计数频率，系统控制器时钟提供以下使能控制功能：

- 支持对输入的计数时钟进行采样，生成时钟使能信号，输出给 Watch Dog 和 TIMER 模块；
- 可通过软件强制将 Watch Dog 和 TIMER 计数时钟使能拉高，使其内部计数器停止计数，当系统处于 Debug 模式时，Watch Dog 计数功能也会被禁止；
- 支持对 TIMER 的计数时钟源进行选择。

SDRAM 刷新频率应用说明



注意

HCLK 总线频率发生改变的场景包括：

- PLL 频率控制参数发生改变；
- 系统模式发生切换。

SDRAM 刷新周期是以 AHB 总线时钟（HCLK）周期为单位进行设置的（请参见“[3.5.5 MEMC_DYNAMICREFRESH](#)”），因此当总线频率发生改变时，刷新周期必须被重新配置：

- 当 HCLK 时钟频率提高时，必须在频率改变后重新设置刷新周期，否则会导致刷新太快，增大芯片功耗；
- 当 HCLK 时钟频率降低时，必须在频率改变之前以目标工作时钟频率为单位重新设置刷新周期，否则会导致刷新太慢，SDRAM 无法正常工作；
- 在 DOZE 模式和 SLEEP 模式下，HCLK 频率太低甚至被停止，必须使 SDRAM 进入自刷新状态，此时 SDRAM 不可访问。

6.3 寄存器概览

系统控制器寄存器的地址位宽 32 位，地址范围：0x101E_0000~0x101E_0FFF。



表6-5 系统控制器寄存器概览（基址是：0x101E_0000）

偏移地址	名称	描述	页码
0x00	SC_CTRL	系统控制寄存器	6-11
0x04	SC_SYSSTAT	系统状态寄存器	6-13
0x08	SC_ITMCTRL	中断模式控制寄存器	6-14
0x0C	SC_IMSTAT	中断模式状态寄存器	6-14
0x10	SC_XTALCTRL	晶振控制寄存器	6-15
0x14	SC_PLLCTRL	PLL 控制寄存器	6-16
0x18	SC_PLLFCTRL	PLL 频率控制寄存器	6-16
0x1C	SC_PERCTRL0	外设控制寄存器 0	6-18
0x20	SC_PERCTRL1	外设控制寄存器 1	6-19
0x24	SC_PEREN	外设时钟使能寄存器	6-21
0x28	SC_PERDIS	外设时钟禁止寄存器	6-22
0x2C	SC_PERCLKEN	外设时钟使能状态寄存器	6-23
0x30	SC_PERSTAT	保留寄存器	6-23
0xEE0	Version_ID0	版本寄存器的 bit[7:0]	6-24
0xEE4	Version_ID1	版本寄存器的 bit[15:8]	6-24
0xEE8	Version_ID2	版本寄存器的 bit[23:16]	6-24
0xEEC	Version_ID3	版本寄存器的 bit[31:24]	6-25

6.4 寄存器描述

本节详细描述了系统控制器寄存器。

6.4.1 SC_CTRL

系统控制寄存器 SC_CTRL 用于定义系统所需的配置参数，受系统上电复位和全局软复位控制。

- 偏移地址：0x000
- 复位值：0x212
- 复位方式：h/s



比特	名称	操作类型	描述
[31:24]	Reserved	-	保留。
[23]	WDogEnOv	R/W	看门狗时钟使能控制。 0: 使能由采样 32.768kHz 时钟产生; 1: 使能被强制拉高。
[22]	TimerEn3Ov	R/W	TIMER3 时钟使能控制。 0: 使能信号通过采用参考时钟得到, 参考时钟的选择由 TimerEn3Sel 指定; 1: 使能信号被强制拉高。
[21]	TimerEn3Sel	R/W	TIMER3 时钟使能参考时钟选择。 0: 选择 32.768kHz 睡眠时钟; 只能设置为 0。
[20]	TimerEn2Ov	R/W	TIMER2 时钟使能控制。 0: 使能信号通过采用参考时钟得到, 参考时钟的选择由 TimerEn2Sel 指定; 1: 使能信号被强制拉高。
[19]	TimerEn2Sel	R/W	TIMER2 时钟使能参考时钟选择。 0: 选择 32.768kHz 睡眠时钟; 只能设置为 0。
[18]	TimerEn1Ov	R/W	TIMER1 时钟使能控制。 0: 使能信号通过采用参考时钟得到, 参考时钟的选择由 TimerEn1Sel 指定; 1: 使能信号被强制拉高。
[17]	TimerEn1Sel	R/W	TIMER1 时钟使能参考时钟选择。 0: 选择 32.768kHz 睡眠时钟; 只能设置为 0。
[16]	TimerEn0Ov	R/W	TIMER0 时钟使能控制。 0: 使能信号通过采用参考时钟得到, 参考时钟的选择由 TimerEn0Sel 指定; 1: 使能信号被强制拉高。
[15]	TimerEn0Sel	R/W	TIMER0 时钟使能参考时钟选择。 0: 选择 32.768kHz 睡眠时钟。 只能设置为 0。
[14:10]	Reserved	-	保留。



比特	名称	操作类型	描述
[9]	RemapStat	R	静态 Boot Memory 地址重映射状态指示。 0: MEMC EBICS1N 片选对应的静态 Boot Memory 位于高端地址空间; 1: MEMC EBICS1N 片选对应的静态 Boot Memory 被 Remap 到地址 0。
[8]	RemapClear	R/W	MEMC EBICS1N 片选对应的静态 Boot Memory 地址 Remap 清除请求。 0: 保持 Remap 状态; 1: 清除 Remap。 Clear Remap 前后地址映射关系参见处理器子系统章节地址映射图。
[7]	Reserved	-	保留, 写 0。
[6:3]	ModeStatus	R	系统控制器系统状态机当前工作模式指示。 0000: SLEEP; 0110: PLL CTL; 0001: DOZE; 1001: SW from XTAL; 0010: SLOW; 1010: SW from PLL; 0011: XTAL CTL; 1011: SW to XTAL; 0100: NORMAL; 1110: SW to PLL。
[2:0]	ModeCtrl	R/W	系统期望的工作模式定义。 000: SLEEP; 001: DOZE; 01X: SLOW; 1XX: NORMAL。

6.4.2 SC_SYSSTAT

系统状态寄存器 SC_SYSSTAT 用于监测系统的状态。往该寄存器写入任意值会触发芯片全局软复位, 全局软复位的效果与芯片上电复位信号 RSTN 一样。

- 偏移地址: 0x004
- 操作类型: R/W
- 复位值: 0x3
- 复位方式: h/s

比特	名称	描述
[31:2]	Reserved	保留。
[1]	BatOk	始终为 1。



比特	名称	描述
[0]	BootStatus	始终为 1。 1: 从 MEMC EBICSN1 启动。

6.4.3 SC_ITMCTRL

中断模式控制寄存器 SC_ITMCTRL 用于控制中断发生时的系统模式。

- 偏移地址：0x008
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:8]	Reserved	保留，只能写 0。
[7]	InMdType	设置触发系统进入中断模式的中断类型。 0: FIQ; 1: FIQ 或者 IRQ。
[6:4]	RSVD	保留，只能写 0。
[3:1]	ItMdCtrl	设置中断模式下系统最低的工作模式，该寄存器的值和 ModeCtrl 寄存器的值相或后作为中断发生后系统所处的工作模式。请参见“6.4.1 SC_CTRL”。
[0]	ItMdEn	中断模式使能。 0: 中断模式被禁止; 1: 当有中断发生时，进入中断模式。

6.4.4 SC_IMSTAT

中断模式状态寄存器 SC_IMSTAT 用于监测和控制系统中断模式。

说明

当中断服务程序结束执行时，必须将中断响应模式清除。

- 偏移地址：0x00C
- 操作类型：R/W
- 复位值：-
- 复位方式：h/s



比特	名称	描述
[31:1]	Reserved	保留，只能写 0。
[0]	ItMdStat	中断模式状态，用于使能中断模式。 0：中断模式没有激活； 1：激活中断模式，允许软件控制中断模式逻辑。

6.4.5 SC_XTALCTRL

晶振控制寄存器 SC_XTALCTRL 用于对 27MHz 晶振进行控制。该寄存器只在上电复位时被复位。系统在 SLOW 和 NORMAL 模式下，27MHz 晶振须被使能。

- 偏移地址：0x010
- 复位值：0x2
- 复位方式：h

比特	名称	操作类型	描述
[31:19]	Reserved	-	保留，只能写 0。
[18:3]	XtalTime	R/W	设置 27MHz 晶振的稳定时间，单位为 32.768kHz 时钟周期。稳定时间计算公式： $65536 - \text{XtalTime}$ 。 如果该寄存器的 XtalOver 为 0，必须按 27MHz 晶振稳定时间设置正确的值。
[2]	XtalStat	R	27MHz 晶振稳定状态位。 0：未稳定； 1：已稳定。
[1]	XtalEn	R/W	晶振使能位。当 XtalOver 位为 1，芯片晶振使能将由该比特直接驱动。 0：关断晶振； 1：使能晶振。
[0]	XtalOver	R/W	设置芯片晶振使能输出是受软件控制还是由系统状态机控制。 0：状态机控制方式； 1：软件控制方式。 目前 Hi3510 仅支持状态机控制方式，所以该位必须写 0。



6.4.6 SC_PLLCTRL

PLLs 控制寄存器用于控制合成 ARM 子系统和 DSP 子系统时钟的 PLL 工作方式。

- 偏移地址：0x014
- 复位值：0x0
- 复位方式：h

比特	名称	操作类型	描述
[31:28]	Reserved	-	保留，只能写 0。
[27:3]	PllTime	R/W	设置 PLL 的稳定时间，单位为 27MHz 时钟周期。稳定时间计算公式： $33554432 - \text{PllTime}$ 。 该寄存器的 PllOver 必须设置为 0，设定的 PLL 稳定时间不应小于 0.5ms。
[2]	PllStat	R	PLL 锁定状态指示。 0：未锁定； 1：已锁定。
[1]	PllEn	R/W	PLL 使能控制，如果 PllOver 比特被设为 1，PLL 使能由该比特控制。 0：关断 PLL； 1：使能 PLL。
[0]	PllOver	R/W	软件控制 PLL 使能方式。 0：状态机控制； 1：软件控制。 目前 Hi3510 仅支持状态机控制方式，所以该位必须写 0。

6.4.7 SC_PLLFCTRL

该寄存器用于控制 ARM 子系统和 DSP 子系统 PLLs 的合成频率。PLL 输出频率和输入

频率的关系为： $F_{\text{out}} = F_{\text{in}} \times \frac{xM}{xN} \times \frac{1}{NO}$ （x 表示 ARM 或 DSP），其中：

- → F_{out} 为 $x\text{PLL}$ 输出频率
- → F_{in} 为 $x\text{PLL}$ 输入频率
- → xM 为 $x\text{PLL}$ 反馈时钟分频因子
- → xN 为 $x\text{PLL}$ 输入时钟分频因子
- → NO 为输出时钟分频因子



$$xM = M0 \times 1 + M1 \times 2 + M2 \times 4 + M3 \times 8 + M4 \times 16 + M5 \times 32 + M6 \times 64 + M7 \times 128$$

$$xN = N0 \times 1 + N1 \times 2 + N2 \times 4 + N3 \times 8$$

$$NO = 2^{xOD0 + 2 \times xOD1}$$

- → M7~M0 分别对应 xP11MD[7:0]
- → N3~N0 分别对应 xP11ND[3:0]
- → OD1、OD0 分别对应 xP11OD[1:0]

以上参数要满足以下条件：

$$1\text{MHz} \leq \frac{F_m}{xN} \leq 25\text{MHz}$$

$$200\text{MHz} \leq F_{out} \times NO \leq 1000\text{MHz}$$

$$xM \geq 2、xN \geq 2$$

- 偏移地址：0x018
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h

比特	名称	描述
[31]	Reserved	保留。
[30]	ZSPPIIBP	DSP PLL Bypass 控制。 0: DSP PLL 不被 bypass; 1: DSP PLL 被 bypass。
[29:28]	ZSPPIIOD	DSP PLL 输出时钟分频因子。
[27:20]	ZSPPIIMD	DSP PLL 反馈时钟分频因子。
[19:16]	ZSPPIIND	DSP PLL 输入时钟分频因子。
[15]	Reserved	保留。
[14]	ARMPPIIBP	ARM PLL Bypass 控制。 0: ARM PLL 不被 bypass; 1: ARM PLL 被 bypass
[13:12]	ARMPPIIOD	ARM PLL 输出时钟分频因子。
[11:4]	ARMPPIIMD	ARM PLL 反馈时钟分频因子。
[3:0]	ARMPPIIND	ARM PLL 输入时钟分频因子。



6.4.8 SC_PERCTRL0

外设控制寄存器 SC_PERCTRL0 用于芯片部分模块的软复位控制。

- 偏移地址：0x01C
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h

比特	名称	描述
[31:16]	Reserved	保留。
[15]	DdblKsrst	解码 de-blocking 模块软复位控制。 0: 撤消 DDBLK 模块软复位; 1: DDBLK 模块软复位。
[14]	UsbSrst	USB 模块软复位控制。 0: USB 模块软复位; 1: 撤消 USB 模块软复位。
[13]	SfSrst	SF 模块软复位控制。 0: SF 模块软复位; 1: 撤消 SF 模块软复位。
[12]	Reserved	保留。
[11]	VdecZspSrst	视频解码 DSP 时钟域电路软复位控制。 0: 撤消视频解码 DSP 时钟域电路软复位; 1: 视频解码 DSP 时钟域电路软复位。
[10]	VdecArmSrst	视频解码 AHB 总线时钟域电路软复位控制。 0: 撤消视频解码 AHB 总线时钟域电路软复位; 1: 视频解码 AHB 总线时钟域电路软复位。
[9]	VencZspSrst	视频编码 DSP 时钟域电路软复位控制。 0: 撤消视频编码 DSP 时钟域电路软复位; 1: 视频编码 DSP 时钟域电路软复位。
[8]	VencArmSrst	视频编码 AHB 总线时钟域电路软复位控制。 0: 撤消视频编码 AHB 总线时钟域电路软复位; 1: 视频编码 AHB 总线时钟域电路软复位。
[7]	DsuSrst	DSU 模块软复位控制。 0: 撤消 DSU 模块软复位; 1: DSU 模块软复位。



比特	名称	描述
[6]	VoSbrst	VOU AHB 总线时钟域电路软复位控制。 0: 撤消 VOU AHB 总线时钟域电路软复位; 1: VOU AHB 总线时钟域电路软复位。
[5]	VoSvrst	VOU 输出时钟域电路软复位控制。 0: VOU 输出时钟域电路软复位; 1: 撤消 VOU 输出时钟域电路软复位。
[4]	ViuSbrst	VIU AHB 总线时钟域电路软复位控制。 0: 撤消 VIU AHB 总线时钟域电路软复位; 1: VIU AHB 总线时钟域电路软复位。
[3]	ViuSvrst	VIU 27MHz 时钟域电路软复位控制。 0: VIU 27MHz 时钟域电路软复位; 1: 撤消 VIU 27MHz 时钟域电路软复位。
[2]	ZspBrgSrst	DSP 桥软复位控制 ^a 。 0: 撤消 DSP 桥软复位; 1: DSP 桥软复位。
[1]	ZspPeriphSrst	DSP 外设软复位控制 ^a 。 0: 撤消 DSP 外设软复位; 1: DSP 外设软复位。
[0]	ZspCoreSrst	DSP Core 软复位控制。 0: DSP Core 软复位; 1: 撤消 DSP Core 软复位。

注: a 表示 ZspBrgSrst 和 ZspPeriphSrst 必须同时进行软复位, 否则可能产生不可预知异常。

6.4.9 SC_PERCTRL1

外设控制寄存器 SC_PERCTRL1 用于外设的通用控制。

- 偏移地址: 0x020
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h



比特	名称	描述
[31]	Arm2ZspInt	ARM 向 DSP 发中断控制。 该中断连接 DSP 的中断 1。 ARM 软件可根据 DSP 中断 1 的中断触发类型控制向 DSP 请求中断。
[30]	Zsp2ArmInt	DSP 向 ARM 发中断控制（对应于 VIC 中断控制器的 DSP2ARM 中断 2，bit29）。 0：清除 DSP 请求中断 ARM； 1：DSP 请求中断 ARM。
[29]	ZspNmiLnt	DSP NMI 中断。 0：清除向 DSP 发出的 NMI 中断； 1：向 DSP 发送 NMI 中断。
[28]	BrgHAddrCfg	DSP-to-ARM 异步桥的高位地址配置方式控制。 0：通过内部寄存器配置异步桥高位地址； 1：通过 DSP 协处理器配置异步桥高位地址。
[27:25]	Reserved	保留。
[24]	VockInvCtrl	控制 nVOCLK 与 VOCK 时钟相位。 0：nVOCLK 与 VOCK 时钟反相输出； 1：nVOCLK 与 VOCK 时钟同相输出。
[23:20]	DdrcCkeInit	控制 DDR SDRAM 的 self-refresh 模式。 上电时，该信号必须为 1； DDR SDRAM 进入 self-refresh 时，该信号必须设置为 0。
[19:16]	MpmcCkeInit	控制 SDR SDRAM 的 self-refresh 模式。 上电时，该信号必须为 1； SDRAM 进入 self-refresh 时，该信号必须设置为 0。
[15:8]	ZspBootHaddr	用于设定 DSP boot 地址的高 8bit，低 8bit 全为 1'b0。
[7]	DdrcDqmInit	控制 DDR SDRAM 的 self-refresh 模式。 上电时，该信号必须为 1； DDR SDRAM 进入 self-refresh 时，该信号必须设置为 0。
[6]	MpmcDqmInit	控制 SDR SDRAM 的 self-refresh 模式。 上电时，该信号必须为 1； SDRAM 进入 self-refresh 时，该信号必须设置为 0。
[5]	VonClkOen	GPIO3[4]复用为 nVOCLK 时的输出三态控制。 0：GPIO3[4]复用 nVOCLK 时输出为高阻； 1：GPIO3[4]复用 nVOCLK 时输出有效。



比特	名称	描述
[4]	DdrPwrDown	DDRC 管脚低功耗模式控制。 0: DDRC 管脚输入有效; 1: DDRC 管脚进入低功耗状态, 输入无效。
[3]	LcdpOen	LCD 输出三态控制。 0: LCD 输出管脚复用为 LCDP 时, 其输出为高阻; 1: LCD 输出管脚复用为 LCDP 时, 其输出有效。 说明: <ul style="list-style-type: none"> • 由于 LCD[23:20] 和 {GPIO6[0],GPIO5[7:5]} 复用, 在使用 {GPIO6[0],GPIO5[7:5]} 的时候也需要配置该位有效; • LCD[19:15]和{SIOXFS1,SIODO1,SIOXCK1,SIODI1,SIORFS1}复用, 在使用 SIO1 的时候也需要配置该位有效。
[2]	VoClkOen	VOCK 管脚输出三态控制。 0: VOCK 输出为高阻; 1: VOCK 输出有效。
[1]	UsbCkSel	USB 时钟的输入源选择信号。 0: USB 时钟来自外部晶振输入 (XIN3, 48MHz 晶振); 1: USB 时钟由内部分频产生。
[0]	VoCkSel	VOU 时钟输入源选择信号。 0: VOU 时钟来自外部晶振输入 (VOCK); 1: VOU 时钟来自主时钟晶振 (XIN2, 实际接 27MHz 晶体)。

6.4.10 SC_PEREN

外设时钟使能寄存器 SC_PEREN 用于使能外设时钟。

对比特写 1, 使能对应的时钟; 写 0 不影响时钟使能的状态。

软件必须在初始化时使能各模块的时钟后才能使之正常工作。

- 偏移地址: 0x024
- 操作类型: W
- 复位值: -
- 复位方式: h/s

比特	名称	描述
[31:12]	Reserved	保留, 只能写 0。
[11]	ZspClken	DSP 时钟使能控制。



比特	名称	描述
[10]	DdblClken	视频解码 de-blocking 模块时钟使能控制。
[9]	VdecClken	视频解码模块时钟使能控制。
[8]	VencClken	视频编码模块时钟使能控制。
[7]	DsuClken	DSU 模块时钟使能控制。
[6]	SfClken	SF 模块时钟使能控制。
[5]	DesClken	DES 模块时钟使能控制。
[4]	UsbClkEn	USB 模块时钟使能控制。
[3]	SspClkEn	SSP 模块时钟使能控制。
[2]	ViuClkEn	VIU 模块时钟使能控制。
[1]	VouClkEn	VOU 模块时钟使能控制。
[0]	Etm9ClkEn	ETM9 模块时钟使能控制。

6.4.11 SC_PERDIS

外设时钟禁止寄存器 SC_PERDIS 用于禁止外设时钟。

对比特写 1，禁止对应的时钟；写 0 不影响时钟使能的状态。

- 偏移地址：0x028
- 操作类型：W
- 复位值：-
- 复位方式：h/s

比特	名称	描述
[31:12]	Reserved	保留，只能写 1。
[11]	ZspClkDis	DSP 时钟禁止控制。
[10]	DdblClkDis	视频解码 de-blocking 模块时钟禁止控制。
[9]	VdecClkDis	视频解码模块时钟禁止控制。
[8]	VencClkDis	视频编码模块时钟禁止控制。
[7]	DsuClkDis	DSU 模块时钟禁止控制。
[6]	SfClkDis	SF 模块时钟禁止控制。
[5]	DesClkDis	DES 模块时钟禁止控制。
[4]	UsbClkDis	USB 模块时钟禁止控制。



比特	名称	描述
[3]	SspClkDis	SSP 模块时钟禁止控制。
[2]	ViuClkDis	VIU 模块时钟禁止控制。
[1]	VouClkDis	VOU 模块时钟禁止控制。
[0]	Etm9ClkDis	ETM9 模块时钟禁止控制。

6.4.12 SC_PERCLKEN

外设使能状态寄存器 SC_PERCLKEN 用于回读各外设时钟使能控制信号的状态。

- 偏移地址：0x02C
- 操作类型：R
- 复位值：0xFFF
- 复位方式：h/s

比特	名称	描述
[31:12]	Reserved	保留，读返回 0。
[11]	ZspClkStatus	DSP 时钟使能状态。
[10]	DdblClkStatus	视频解码 de-blocking 模块时钟使能状态。
[9]	VdecClkStatus	视频解码模块时钟使能状态。
[8]	VencClkStatus	视频编码模块时钟使能状态。
[7]	DsuClkStatus	DSU 模块时钟使能状态。
[6]	SfClkStatus	SF 模块时钟使能状态。
[5]	DesClkStatus	DES 模块时钟使能状态。
[4]	UsbClkStatus	USB 模块时钟使能状态。
[3]	SspClkStatus	SSP 模块时钟使能状态。
[2]	ViuClkStatus	VIU 模块时钟使能状态。
[1]	VouClkStatus	VOU 模块时钟使能状态。
[0]	Etm9ClkStatus	ETM9 模块时钟使能状态。

6.4.13 SC_PERSTAT

- 偏移地址：0x030
- 操作类型：R



- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:0]	Reserved	保留。

6.4.14 Version_ID0

Hi3510 芯片的版本寄存器由 Version_ID0~Version_ID3 的低 8 位组成，共 32 位。

- 偏移地址：0xEE0
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	Vesion_ID0	Hi3510 版本寄存器的 bit[7:0]，只读。

6.4.15 Version_ID1

- 偏移地址：0xEE4
- 操作类型：R
- 复位值：0x1
- 复位方式：h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	Vesion_ID1	Hi3510 版本寄存器的 bit[15:8]，只读。

6.4.16 Version_ID2

- 偏移地址：0xEE8
- 操作类型：R
- 复位值：0x10
- 复位方式：h/s

比特	名称	描述
[31:8]	Reserved	保留。



比特	名称	描述
[7:0]	Vesion_ID2	Hi3510 版本寄存器的 bit[23:16]，只读。

6.4.17 Version_ID3

- 偏移地址：0xEEC
- 操作类型：R
- 复位值：0x35
- 复位方式：h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	Vesion_ID3	Hi3510 版本寄存器的 bit[31:24]，只读。



7 直接存储访问控制器 (DMAC)

关于本章

本章描述内容如下表所示。

标题	内容
7.1 概述	概括介绍 DMAC。
7.2 特点	概括介绍 DMAC 的特点。
7.3 功能描述	概括介绍 DMAC 的功能。
7.4 接口信号描述	介绍 DMAC 的外部接口信号。
7.5 工作方式	描述 DMAC 的工作原理。
7.6 寄存器概览	概括介绍 DMAC 的寄存器。
7.7 寄存器描述	详细描述 DMAC 的寄存器。



7.1 概述

Hi3510 提供了 1 个 16 路请求、8 通道的高性能 DMA 控制器 (DMAC)，DMAC 用来完成芯片各个模块间快速的数据搬移。

7.2 特点

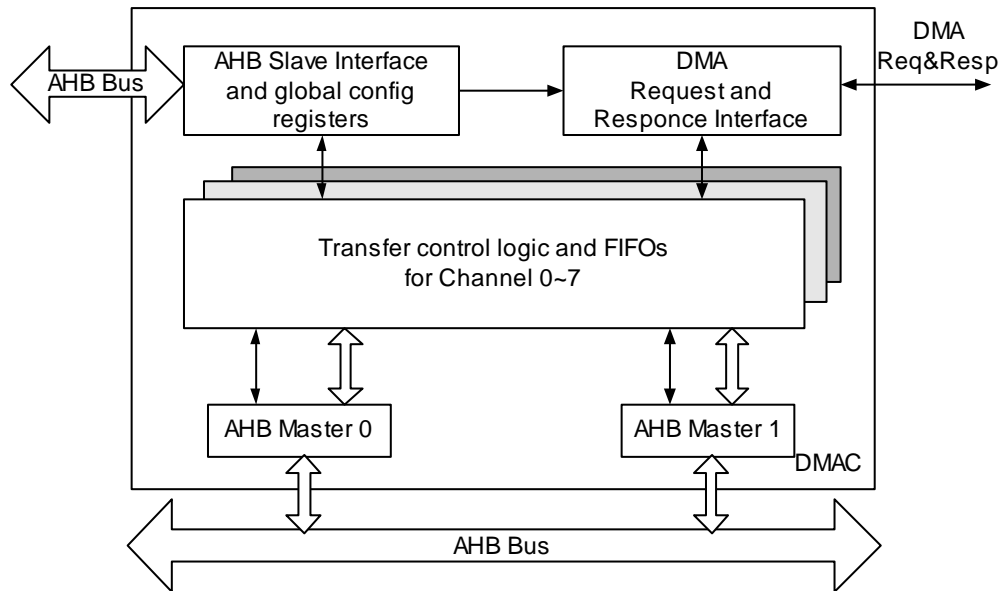
DMAC 有以下特点：

- 支持 8 位、16 位、32 位数据传输
- 提供 8 个 DMA 通道，每个通道可配置用于一种单向传输
- 提供 16 个 DMA 请求输入，可通过配置，分配给 8 个 DMA 通道上的不同源外设和目的外设
- 支持 4 种数据传送方向：
 - Memory 至外设
 - Memory 至 Memory
 - 外设至 Memory
 - 外设至外设
- 支持 Single 和 Burst 两种传输模式
- 支持通过编程决定 DMA burst 长度
- 支持链表 DMA 传输
- 源地址和目的地址可分别配置为在 DMA 传输过程中自动递增或不递增
- 硬件决定 DMA 通道优先级，优先级从高到低对应的通道号依次为：0~7；当来自两个外设的 DMA 请求同时有效时，优先级高的通道先开始传输
- 对于低优先级通道（6、7 通道）的操作，支持在连续进行 4、8、16burst 总线操作后，自动插入 Idle 周期，以便其他设备能够抢占总线进行传送，从而避免过长的等待
- DMAC 进行流量控制，DMA 传输长度由 DMAC 控制
- 支持三种中断状态查询：
 - 传输完成中断
 - 传输出错中断
 - 传输完成中断、传输出错中断的组合中断
- 提供 1 个可屏蔽中断输出
- 支持 DMAC 使能禁止，用于功耗控制
- 支持软件控制的 DMA 请求
- 支持 1 路芯片外部 DMA 请求操作

7.3 功能描述

DMAC 的内部结构如图 7-1 所示。

图7-1 DMAC 功能框图



DMAC 的每一个通道都内含一组传输控制逻辑和一个 FIFO，传输控制逻辑自动完成以下过程：

1. 从软件指定的源地址位置读取数据
2. 缓存到通道内含的 FIFO 中
3. 从通道 FIFO 中取出数据
4. 写入到软件指定的目的地址位置

每一个通道的传输控制逻辑访问源地址和目的地址是通过两个 AHB Master 实现的。当一个通道需要通过总线访问源地址或目的地址时，需要申请占用软件指定的 AHB Master。

通道的源端和目的端可使用不同的 AHB Master，也可使用同一个 AHB Master，由软件配置指定。在每个 AHB Master 内均包含一个仲裁器，用于仲裁多个通道发起的占用 AHB Master 的请求。

DMAC 的 8 个通道具有固定优先级，当多个通道同时发起占用 AHB Master 的请求时，优先级高的通道获得 AHB Master 的使用权。

DMAC 的 16 条请求线分别与不同外设的 DMA 请求线相连，各条请求线与外设的对应关系如表 7-2 所示。DMA 请求线与通道的源外设、目的外设的连接关系可由软件配置。例如，DMA 请求线 2 为 SIO0 的接收通道请求线，若希望使用通道 3 用于传输 SIO0 的接收数据，则应配置 DMA 请求线 2 与通道 3 的源端相连。

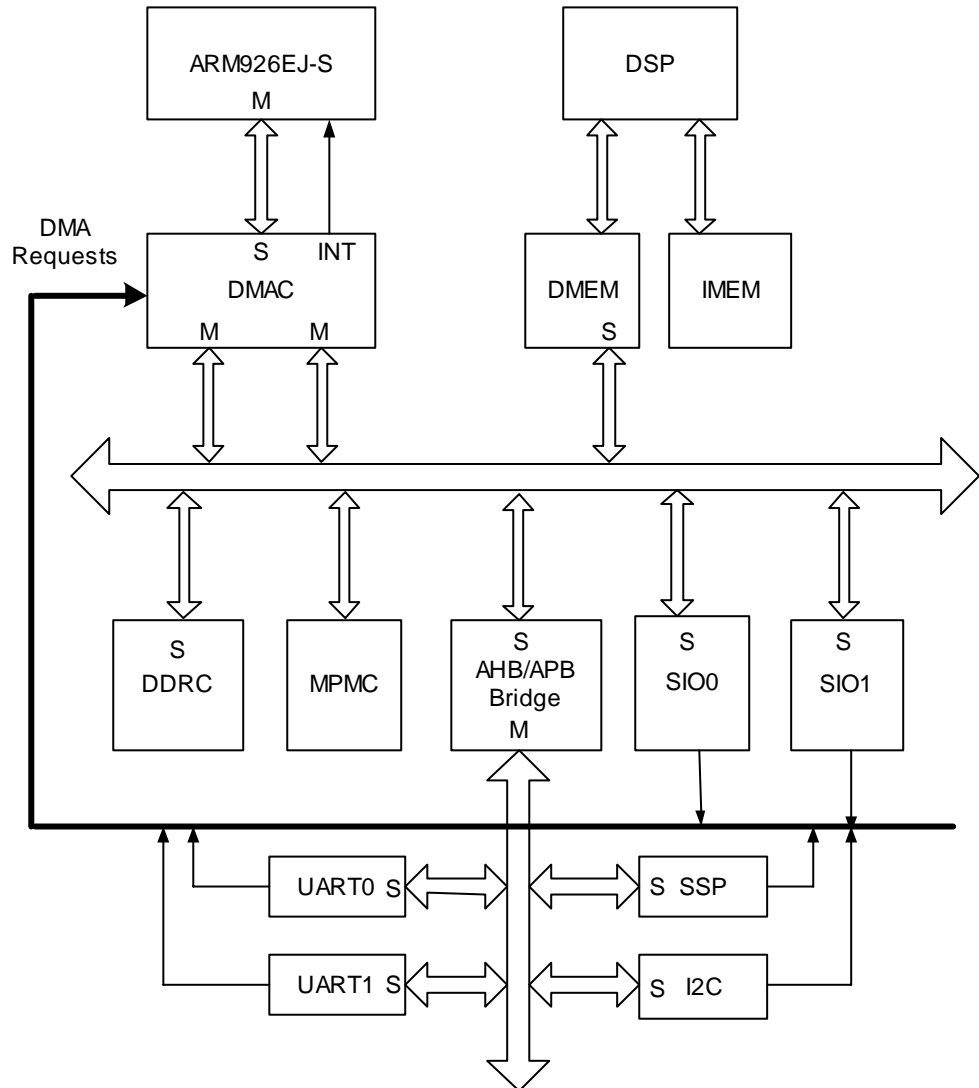


说明

Memory 没有 DMA 请求线，当 DMA 传输的一方为 Memory 时，DMAC 默认其 DMA 请求是始终有效的。

DMAC 与芯片内部其他模块的连接关系如图 7-2 所示。

图7-2 DMAC 的请求线与其他外设的对应关系



7.4 接口信号描述

表 7-1 为 DMAC 在 Hi3510 芯片中的外部接口信号表。



表7-1 DMA 接口信号描述

信号名	方向	描述
DREQ	I	外部 burst DMA 请求输入，低电平有效。
DACK	O	外部 burst DMA 请求响应输出，低电平有效。

Hi3510 的 ARM 处理器和 DSP 处理器都可以通过软件控制 DMAC。

DMAC 提供 2 个中断信号：

- 1 个送给 ARM 子系统的 VIC 控制器
- 1 个送给 DSP 子系统

DMAC 的硬件请求线和相应设备的对应关系如表 7-2 所示。

表7-2 DMAC 硬件请求线和相应设备的对应关系

请求线编号	对应设备	请求线编号	对应设备
0	I ² C 接收通道	8	SSP 接收通道
1	I ² C 发送通道	9	SSP 发送通道
2	SIO0 接收通道	10	SIO1 接收通道
3	SIO0 发送通道	11	SIO1 发送通道
4	外部 DMA 请求通道	12	UART1 接收通道
5	保留	13	UART1 发送通道
6	保留	14	UART0 接收通道
7	保留	15	UART0 发送通道

7.5 工作方式

DMAC 的工作原理为：

1. 软件选定 DMAC 的一个通道用于 DMA 传输，配置该通道的源地址、目的地址、链表指针、传输数据个数、源/目的外设请求线号及访问源/目的端所使用的 AHB Master，并启动该通道；
2. 源外设向 DMAC 发起 DMA 请求（源设备为 Memory 除外）；
3. DMAC 通道响应源外设 DMA 请求，从源外设读取数据并存入通道内部的 FIFO 中；
4. 目的外设向 DMAC 发起 DMA 请求（目的设备为 Memory 除外）；



5. DMAC 通道响应目的外设 DMA 请求，从通道内部的 FIFO 中取出数据并写入目的外设；
6. 重复步骤 2 至 5 直到指定的传输数据全部完成传输。

7.6 寄存器概览

DMAC 寄存器的地址位宽 32 位，地址范围：0x1013_0000~0x1013_FFFF。

表7-3 DMAC 寄存器概览（基址是 0x1013_0000）

偏移地址	寄存器名称	描述	页码
0x000	DMAC_INTSTATUS0	DMAC 中断状态寄存器 0。	7-8
0x004	DMAC_INTTCSTATUS0	DMAC Terminal Count 中断状态寄存器 0。	7-9
0x008	DMAC_INTTCLEAR	DMAC Terminal Count 中断清除寄存器。	7-9
0x00C	DMAC_INTERRORSTATUS0	DMAC 错误中断状态寄存器 0。	7-10
0x010	DMAC_INTERRCLR	DMAC 错误中断清除寄存器。	7-10
0x014	DMAC_RAWINTTCSTATUS	DMAC Terminal Count 原始中断状态寄存器。	7-10
0x018	DMAC_RAWINTERRORSTATUS	DMAC 原始错误中断状态寄存器。	7-11
0x01C	DMAC_ENBLDCHNS	DMAC 通道使能状态寄存器。	7-11
0x020	DMAC_SOFTBREQ	软件 Burst DMA 请求寄存器。	7-12
0x024	DMAC_SOFTSREQ	软件 Single DMA 请求寄存器。	7-12
0x028	DMAC_SOFTLBREQ	软件 Last Burst DMA 请求寄存器。	7-13
0x02C	DMAC_SOFTLSREQ	软件 Last Single DMA 请求寄存器。	7-13
0x030	DMAC_CONFIGURATION	DMAC 配置寄存器。	7-13
0x034	DMAC_SYNC	DMAC 同步寄存器。	7-14
0x040	DMAC_INTSTATUS1	DMAC 中断状态寄存器 1。	7-15
0x044	DMAC_INTTCSTATUS1	DMAC Terminal Count 中断状态寄存器 1。	7-15
0x048	DMAC_INTERRORSTATUS1	DMAC 错误中断状态寄存器 1。	7-15



偏移地址	寄存器名称	描述	页码
0x100	DMAC_C0SRCADDR	Channel 0 源地址寄存器。	7-16
0x104	DMAC_C0DESTADDR	Channel 0 目的地址寄存器。	7-17
0x108	DMAC_C0LLI	Channel 0 链表项寄存器。	7-17
0x10C	DMAC_C0CONTROL	Channel 0 控制寄存器。	7-18
0x110	DMAC_C0CONFIGURATION	Channel 0 配置寄存器。	7-22
0x120	DMAC_C1SRCADDR	Channel 1 源地址寄存器。	7-16
0x124	DMAC_C1DESTADDR	Channel 1 目的地址寄存器。	7-17
0x128	DMAC_C1LLI	Channel 1 链表项寄存器。	7-17
0x12C	DMAC_C1CONTROL	Channel 1 控制寄存器。	7-18
0x130	DMAC_C1CONFIGURATION	Channel 1 配置寄存器。	7-22
0x140	DMAC_C2SRCADDR	Channel 2 源地址寄存器。	7-16
0x144	DMAC_C2DESTADDR	Channel 2 目的地址寄存器。	7-17
0x148	DMAC_C2LLI	Channel 2 链表项寄存器。	7-17
0x14C	DMAC_C2CONTROL	Channel 2 控制寄存器。	7-18
0x150	DMAC_C2CONFIGURATION	Channel 2 配置寄存器。	7-22
0x160	DMAC_C3SRCADDR	Channel 3 源地址寄存器。	7-16
0x164	DMAC_C3DESTADDR	Channel 3 目的地址寄存器。	7-17
0x168	DMAC_C3LLI	Channel 3 链表项寄存器。	7-17
0x16C	DMAC_C3CONTROL	Channel 3 控制寄存器。	7-18
0x170	DMAC_C3CONFIGURATION	Channel 3 配置寄存器。	7-22
0x180	DMAC_C4SRCADDR	Channel 4 源地址寄存器。	7-16
0x184	DMAC_C4DESTADDR	Channel 4 目的地址寄存器。	7-17
0x188	DMAC_C4LLI	Channel 4 链表项寄存器。	7-17
0x18C	DMAC_C4CONTROL	Channel 4 控制寄存器。	7-18
0x190	DMAC_C4CONFIGURATION	Channel 4 配置寄存器。	7-22
0x1A0	DMAC_C5SRCADDR	Channel 5 源地址寄存器。	7-16
0x1A4	DMAC_C5DESTADDR	Channel 5 目的地址寄存器。	7-17
0x1A8	DMAC_C5LLI	Channel 5 链表项寄存器。	7-17
0x1AC	DMAC_C5CONTROL	Channel 5 控制寄存器。	7-18



偏移地址	寄存器名称	描述	页码
0x1B0	DMAC_C5CONFIGURATION	Channel 5 配置寄存器。	7-22
0x1C0	DMAC_C6SRCADDR	Channel 6 源地址寄存器。	7-16
0x1C4	DMAC_C6DESTADDR	Channel 6 目的地址寄存器。	7-17
0x1C8	DMAC_C6LLI	Channel 6 链表项寄存器。	7-17
0x1CC	DMAC_C6CONTROL	Channel 6 控制寄存器。	7-18
0x1D0	DMAC_C6CONFIGURATION	Channel 6 配置寄存器。	7-22
0x1E0	DMAC_C7SRCADDR	Channel 7 源地址寄存器。	7-16
0x1E4	DMAC_C7DESTADDR	Channel 7 目的地址寄存器。	7-17
0x1E8	DMAC_C7LLI	Channel 7 链表项寄存器。	7-17
0x1EC	DMAC_C7CONTROL	Channel 7 控制寄存器。	7-18
0x1F0	DMAC_C7CONFIGURATION	Channel 7 配置寄存器。	7-22

7.7 寄存器描述

本节详细描述了 DMAC 的寄存器。

7.7.1 DMAC_INTSTATUS0

中断状态寄存器 DMAC_INTSTATUS0 给出了经过屏蔽后的中断状态。该寄存器的每一位对应着 DMAC 的一个通道。当其某一位为高电平时表示相应的通道有中断请求产生，该中断请求可能来自该通道的 Error 中断或 Terminal Count 中断。如某一通道的配置寄存器 DMAC_CxCONFIGURATION 中的 ITC0 和 IE0 没有置位，则该对应中断被屏蔽。

Hi3510 中的 DMAC 支持 ARM926EJ 和 DSP 对其进行操作，它提供了 2 个中断信号，1 个送给 ARM926EJ-S，另外 1 个送给 DSP。

当 DMAC 向 ARM926EJ-S 发出中断时，ARM 需要查询 DMAC_INTSTATUS0（请参见本节）、DMAC_INTTCSTATUS0（请参见“7.7.2 DMAC_INTTCSTATUS0”）和 DMAC_INTERRORSTATUS0（请参见“7.7.4 DMAC_INTERRORSTATUS0”）这几个寄存器，以获得中断的详细信息。

- 偏移地址：0x000
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s



比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	IntStatus0	DMA 各通道经屏蔽后的中断状态，每比特对应一个通道。

7.7.2 DMAC_INTTCSTATUS0

传输结束状态寄存器 DMAC_INTTCSTATUS0 表示经过屏蔽后的 Terminal Count 中断状态，对应的屏蔽位为寄存器 DMAC_CxCONFIGURATION 的 ITC0 位（其中 x 表示通道号 0~7）。该寄存器必须和寄存器 DMAC_INTSTATUS0 结合在一起使用。

- 偏移地址：0x004
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	IntTCStatus0	经过屏蔽后的 Terminal Count 中断状态，每比特对应一个通道。

7.7.3 DMAC_INTTCLEAR

传输结束状态清除寄存器 DMAC_INTTCLEAR 用于清除 Terminal Count 中断。

当写该寄存器时，如果某位写入值 1，则状态寄存器 DMAC_INTTCSTATUS0 和 DMAC_INTTCSTATUS1 的相应位同时被清零；如果某位写入值 0，则状态寄存器的相应位无影响。

读该寄存器时返回值是 0。

- 偏移地址：0x008
- 操作类型：W
- 复位值：-
- 复位方式：h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	IntTCClear	Terminal Count 中断清除请求，每比特对应一个通道。



7.7.4 DMAC_INTERRORSTATUS0

中断错误状态寄存器 DMAC_INTERRORSTATUS0 表示经过屏蔽后的出错中断状态，对应的屏蔽位为寄存器 DMAC_CxCONFIGURATION 的 IE0（其中 x 表示通道号 0~7）。若使用组合中断信号 DMAC_INTR 用于向 CPU 发出中断，则该寄存器必须和寄存器 DMAC_INTSTATUS0 结合在一起使用。

- 偏移地址：0x00C
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	IntErrorStatus0	经过屏蔽后的出错中断状态，每比特对应一个通道。

7.7.5 DMAC_INTERRCLR

中断错误清除寄存器 DMAC_INTERRCLR 用于清除出错中断。

当写该寄存器时，如某位写入值 1，则状态寄存器 DMAC_INTTCSTATUS0 和 DMAC_INTTCSTATUS1 的相应位同时清零；如某位写入值 0，则状态寄存器的相应位无影响。

- 偏移地址：0x010
- 操作类型：W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	IntErrClr	清除出错中断。 0：该比特对状态寄存器的相应位无影响； 1：该比特对状态寄存器的相应位清零。

7.7.6 DMAC_RAWINTTCSTATUS

传输结束原始中断状态寄存器 DMAC_RAWINTTCSTATUS 给出了各通道屏蔽前的 Terminal Count 中断状态。

某位为高电平表示对应的通道发出了 Terminal Count 中断请求。

- 偏移地址：0x014



- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	RawIntTCStatus	原始 Terminal Count 中断状态。

7.7.7 DMAC_RAWINTERRORSTATUS

传输错误原始中断状态寄存器 DMAC_RAWINTTCSTATUS 给出了各通道屏蔽前的出错中断状态。

某位为高电平表示对应的通道发出了出错中断请求。

- 偏移地址: 0x018
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	RawIntErrStatus	各通道屏蔽前的出错中断状态。

7.7.8 DMAC_ENBLDCHNS

通道使能状态寄存器 DMAC_ENBLDCHNS 用于表明被使能的通道。

寄存器 DMAC_ENBLDCHNS 的某位为 1, 表示对应的通道被使能; 某个通道是否被使能由该通道的通道寄存器 DMAC_CxCONFIGURATION[Enable]位决定; 当某个通道的 DMA 传输结束时, 寄存器 DMAC_ENBLDCHNS 中与该通道对应的位即被清零。

- 偏移地址: 0x01C
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	EnabledChannels	通道使能状态, 每比特对应一个通道。



7.7.9 DMAC_SOFTBREQ

Software Burst Request 寄存器 DMAC_SOFTBREQ 用于供软件控制产生 DMA burst 传输请求。

向该寄存器的相应位写 1 实现 DMA burst 传输请求的产生，当传输结束时该寄存器中的相应位被清零；向该寄存器的相应位写 0 没有任何影响。

读该寄存器可得知当前正在请求 DMA 传输的设备，外设和该寄存器都可以产生一个 DMA 请求。

说明

建议软件和硬件在不同的位上请求 DMA 传输。

- 偏移地址：0x020
- 操作类型：W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:16]	Reserved	保留。
[15:0]	SoftBReq	用于软件控制产生 DMA burst 请求。

7.7.10 DMAC_SOFTSREQ

Software Single Request 寄存器 DMAC_SOFTSREQ 用于供软件控制产生 DMA signal 传输请求。

向该寄存器的相应位写 1 实现 DMA signal 传输请求的产生，当传输结束时该寄存器中的相应位被清零；向该寄存器的位写 0 没有任何影响；读该寄存器可得知当前正在请求 DMA 传输的设备，通过 DMAC 的 16 个 DMA 请求输入信号和该寄存器都可以产生一个 DMA 请求。

说明

建议软件和硬件在不同的位上请求 DMA 传输。

- 偏移地址：0x024
- 操作类型：W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:16]	Reserved	保留。
[15:0]	SoftSReq	用于软件控制产生 DMA signal 传输请求。



7.7.11 DMAC_SOFTLBREQ

Software Last Burst Request Register 寄存器 DMAC_SOFTLBREQ 用于供软件控制产生 DMA last burst 传输请求。

向该寄存器的位写 1 可产生一个 DMA last burst 传输请求；向该寄存器的位写 0 没有任何影响；读该寄存器返回当前请求 DMA 传输的外设。

- 偏移地址：0x028
- 操作类型：W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:16]	Reserved	保留。
[15:0]	SoftLBReq	由软件发起 last burst 请求。

7.7.12 DMAC_SOFTLSREQ

Software Last Single Request Register 寄存器 DMAC_SOFTLSREQ 用于供软件控制产生 DMA last single 传输请求。

向该寄存器的位写 1 可产生一个 DMA last single 传输请求；向该寄存器的位写 0 没有任何影响；读该寄存器返回当前请求 DMA 传输的外设。

- 偏移地址：0x02C
- 操作类型：W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:16]	Reserved	保留。
[15:0]	SoftLSReq	由软件发起 last signal 传输请求。

7.7.13 DMAC_CONFIGURATION

配置寄存器 DMAC_CONFIGURATION 用于配置 DMAC 的操作。

通过写该寄存器的 M1 (bit1) 和 M2 (bit2)，可改变 DMAC 的 2 个 AHB master 接口的大小端模式 (endianness)；复位时 DMAC 的 2 个 AHB master 接口设为 little-endian 模式。



说明

2 个 AHB master 接口可以不必采用相同的 endianness。

- 偏移地址：0x030
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:3]	Reserved	保留。
[2]	M2	AHB Master 2 endianness 配置位。 0: little-endian 模式; 1: big-endian 模式。
[1]	M1	AHB Master 1 endianness 配置位。 0: little-endian 模式; 1: big-endian 模式。
[0]	E	PrimeCell DMAC enable。 0: 关闭 DMAC; 1: 启动 DMAC。 关闭 DMAC 可减少功耗。

7.7.14 DMAC_SYNC

同步寄存器 DMAC_SYNC 用于控制是否需要为 DMA 请求信号提供同步逻辑。当发起 DMA 请求的模块工作在与总线时钟不同的频率时，其 DMA 请求线在进入 DMAC 时需要进行同步处理，此时需要在寄存器 DMAC_SYNC 中把与该模块 DMA 请求线对应的位置位。

说明

最好配置为禁用状态。

- 偏移地址：0x034
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:16]	Reserved	保留。
[15:0]	DMAC_Sync	控制是否需要请求线进行同步。



7.7.15 DMAC_INTSTATUS1

中断状态寄存器 DMAC_INTSTATUS1 给出了经过屏蔽后的中断状态，如某一通道的配置寄存器 DMAC_CxCONFIGURATION 中的 ITC1 和 IE1 没有置位，则该寄存器的相应位被屏蔽。

Hi3510 中的 DMAC 支持 ARM926EJ 和 DSP 对其进行操作，它提供了 2 个中断信号，一个送给 ARM926EJ-S，另外一个送给 DSP。

当其向 DSP 发出中断时，DSP 需要查询 DMAC_INTSTATUS1（请参见本节）、DMAC_INTTCSTATUS1（请参见“7.7.16 DMAC_INTTCSTATUS1”）和 DMAC_INTERRORSTATUS1（请参见“7.7.17 DMAC_INTERRORSTATUS1”）寄存器，以获得中断的详细信息。

- 偏移地址：0x040
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	IntStatus1	经屏蔽后的中断状态，由 DSP 进行操作。

7.7.16 DMAC_INTTCSTATUS1

传输结束状态寄存器 DMAC_INTTCSTATUS1 表示经过屏蔽后的 terminal count 中断状态，对应的屏蔽位为寄存器 DMAC_CxCONFIGURATION 的 ITC1 位。

- 偏移地址：0x044
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	IntTCStatus1	经过屏蔽后的 terminal count 中断状态，由 DSP 进行操作。

7.7.17 DMAC_INTERRORSTATUS1

传输结束状态寄存器 DMAC_INTERRORSTATUS1 表示经过屏蔽后的出错中断状态，对应的屏蔽位为 DMAC_CxCONFIGURATION 的 IE1 位。

- 偏移地址：0x048
- 操作类型：R



- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7:0]	IntErrorStatus1	经过屏蔽的出错中断状态, 由 DSP 进行操作。

7.7.18 DMAC_CxSRCADDR

DMAC_CxSRCADDR 寄存器的偏移地址为 $0x100 + N \times 0x20$ 。其中 N 的取值为 0~7, 分别对应 DMA 通道 0~DMA 通道 7。

DMAC 提供了 8 个通道, 每个通道都包括一系列通道寄存器, 这些寄存器包括:

- 8 个 DMAC_CxSRCADDR 寄存器
- 8 个 DMAC_CxDESTADDR 寄存器
- 8 个 DMAC_CxLLI 寄存器
- 8 个 DMAC_CxCONTROL 寄存器
- 8 个 DMAC_CxCONFIGURATION 寄存器

当 DMA 从 Memory 中载入 LLI 时, 前 4 个寄存器是由 DMAC 自动更新。



注意

在 DMA 传输正在进行时, 更新通道寄存器会导致 DMAC 产生不可预测的行为。要改变通道的配置, 必须先关闭通道然后再配置相关寄存器。

通道源地址寄存器 DMAC_CxSRCADDR 包含了当前待传数据的源地址 (byte-aligned)。每个寄存器在对应的通道被启动前都要由软件对其直接编程。当通道被启动后, 该寄存器在下列情况下更新:

- 当源地址递增时
- 当传完一个完整的数据 packet 后, 从 LLI 中载入时

当该通道处于活动状态时, 读该寄存器得不到有效信息。这是因为当软件得到读出的寄存器值, 该寄存器的值已经随着通道传输改变了。对该寄存器的读操作一般是用在通道停止传输的时候, 此时读取值显示的是 DMAC 读最后一项时的源地址。



注意

源地址和目的地址必须与源设备和目的设备的传输宽度对齐。

- 偏移地址： $0x100 + N \times 0x20$
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:0]	SrcAddr	DMA 源地址。

7.7.19 DMAC_CxDESTADDR

DMAC_CxDESTADDR 寄存器的偏移地址为 $0x104 + N \times 0x20$ ，其中 N 的取值为 0~7，分别对应 DMA 通道 0~DMA 通道 7。

通道目的地址寄存器 DMAC_CxDESTADDR 包含了当前待传数据的目的地址 (byte-aligned)。每个寄存器在对应的通道被启动前都要由软件对其直接编程。当通道被启动后，该寄存器在下列情况下更新：

- 当目的地址递增时
- 当传完一个完整的数据 packet 后，从 LLI 中载入时

当该通道处于活动状态时，读该寄存器得不到有效信息。这是因为当软件得到读出的寄存器值，该寄存器的值已经随着通道传输改变了。对该寄存器的读操作一般是用在通道停止传输的时候，此时读取值显示的是 DMAC 写最后一项时的目的地址。

- 偏移地址： $0x104 + N \times 0x20$
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:0]	DestAddr	DMA 目标地址。

7.7.20 DMAC_CxLLI

通道链表项寄存器 DMAC_CxLLI 包含了下一个 LLI (链表节点) 的地址 (word-aligned)。

如果该寄存器的值为 0，表示当前 LLI 是链表上最后一个 LLI，则当本 LLI 对应的数据 packet 全部传完后该通道就会被关闭。



DMAC_CxLLI 寄存器的偏移地址为 $0x108 + N \times 0x20$ ，其中 N 的取值为 0~7，分别对应 DMA 通道 0~DMA 通道 7。



注意

该寄存器的 LLI 字段不应指定一个大于 $0xFFFF_FFF0$ 的数。否则，1 个 4 字的 burst 传输将使地址回卷到 $0x0000_0000$ 处，导致 LLI 数据结构不能存储在连续的地址区域中。

- 偏移地址： $0x108 + N \times 0x20$
- 操作类型： R/W
- 复位值： 0x0
- 复位方式： h/s

比特	名称	描述
[31:2]	LLI	Linked list item。 下一个 LLI 地址的[31:2]位，地址位[1:0]为 0。
[1]	R	保留。 写入时必须写 0；读出时屏蔽该位。
[0]	LM	用于载入下一个 LLI 的 AHB master。 0： AHB Master 1； 1： AHB Master 2。

7.7.21 DMAC_CxCONTROL

通道控制寄存器 DMAC_CxCONTROL 包含了 DMA 通道控制信息如 transfer size、burst 长度及传输位宽等。每个寄存器在对应的通道被启动前都要由软件对其直接编程。当通道被启动后，该寄存器的值在传完一个完整的数据 packet 后，从 LLI 中载入时更新。

当该通道处于活动状态时，读该寄存器得不到有效信息。这是因为当软件得到读出的寄存器值，该寄存器的值已经随着通道传输改变了。对该寄存器的读操作一般是用在通道停止传输时。

DMAC_CxCONTROL 寄存器的偏移地址为 $0x10C + N \times 0x20$ ，其中 N 的取值为 0~7，分别对应 DMA 通道 0~DMA 通道 7。

- 偏移地址： $0x108 + N \times 0x20$
- 操作类型： R/W
- 复位值： 0x0
- 复位方式： h/s



比特	名称	描述
[31]	I	Terminal count 中断使能位。 该位用于决定当前 LLI 是否触发 terminal count 中断。
[30:28]	Prot	AHB master 发出的访问保护 HPROT[2:0]信号，这几位的具体含义请参见表 7-4 所示。
[27]	DI	Destination Increment。 当该位置位，目的地址每传一个数就递增一次。
[26]	SI	Source Increment.当该位置位，源地址每传一个数就递增一次。
[25]	D	设置访问目的设备的 AHB master。 0: 使用 master 1 访问； 1: 使用 master 2 访问。
[24]	S	设置访问源设备的 AHB master。 0: 使用 master 1 访问； 1: 使用 master 2 访问。
[23:21]	DWidth	目的设备传输位宽。 <ul style="list-style-type: none"> • 宽于 AHB master 位宽的传输位宽是非法的。 • 目的设备和源设备的位宽可以不一样，硬件自动对数据进行 pack 和 unpack。 • Dwidth 的值和具体的位宽对应关系如表 7-5 所示。
[20:18]	SWidth	源设备传输位宽。 <ul style="list-style-type: none"> • 宽于 AHB master 位宽的传输位宽是非法的。 • 目的设备和源设备的位宽可以不一样，硬件自动对数据进行 pack 和 unpack。 • Swidth 的值和具体的位宽对应关系如表 7-5 所示。
[17:15]	DBSize	目的设备 burst 长度。 表示一次目的设备 burst 传输所需传输的数据个数，即当 DMAC_CxBREQ 有效时，传输的数据个数。 <ul style="list-style-type: none"> • 该值必须设为目的外设支持的 burst 大小，或者若目的设备为存储器，则设为 Memory 中的缓冲区大小。 • 该 burst 的长度与 AHB HBURST 信号无关。DBSize 的值和具体的传输长度的对应关系见表 7-4 所示。



比特	名称	描述
[14:12]	SBSize	<p>源设备 burst 长度。</p> <p>表示一次源设备 burst 传输所需传输的数据个数，即当 DMAC_CxBREQ 有效时，传输的数据个数。</p> <ul style="list-style-type: none"> 该值必须被设为源外设支持的 burst 大小，或者若源设备为存储器时，被设为 Memory 中的缓冲区大小。 该 burst 的长度与 AHB HBURST 信号无关。SBSize 的值和具体的传输长度的对应关系见表 7-4 所示。
[11:0]	Transfer Size	<p>源外设待传数据的个数。</p> <p>当 DMAC 作为流控制器，可通过写该寄存器设定 DMA 传输的长度。</p> <p>读该寄存器可得到在与目的设备相连的总线上已传出的数据个数。</p> <ul style="list-style-type: none"> 当该通道处于活动状态时，读该寄存器得不到有效信息。这是因为当软件得到读出的寄存器值，该寄存器的值已经随着通道传输改变了。对该寄存器的读操作一般是在通道被启动后然后又停止传输时。 当 DMAC 不是流控制器时，该字段无意义。

DMAC_CxCONTROL 寄存器的 DBSize 及 SBSize 的值与其对应的 burst 长度如表 7-4 所示。

表7-4 DBSize、SBSize 的值和对应 burst 长度的关系说明

DBSize、SBSize 的值	Burst 的长度
000	1
001	4
010	8
011	16
100	32
101	64
110	128
111	256

DMAC_CxCONTROL 寄存器的 DWidth 和 SWidth 的值与其对应传输位宽如表 7-5 所示。



表7-5 DWidth 和 SWidth 的值和对应传输位宽的关系说明

SWidth 和 DWidth 的值	传输位宽
000	Byte (8bit)
001	Halfword (16bit)
010	Word (32bit)
011	Reserved
100	Reserved
101	Reserved
110	Reserved
111	Reserved

配置寄存器 DMAC_CxCONTROL 时应注意以下事项:

- 当源设备的传输宽度小于目的设备传输宽度时，源设备的传输宽度乘以 transfer size 应为目的设备传输宽度的整数倍。否则 FIFO 中的数据将会滞留并丢失；
- SWidth 和 DWidth 字段一定不能设置为未定义的位宽；
- transfer size 字段若被写为 0 且 DMAC 又是流控制器，则 DMAC 将不会发生任何传输动作。编程者应负责关闭此 DMA 通道并对此通道重新编程；
- 不对 DMAC_CxCONTROL 寄存器进行普通的写入/读出测试。因为：
 - transfer size 字段不是一个普通的可写入并读回相同值的寄存器字段；
 - 当写入时，该字段正如一个控制寄存器。因为其决定了 DMAC 应传输多少个数据；
 - 当读回时，该字段则相当于一个状态寄存器。因为其返回剩下的待传数据个数（以源设备位宽为单位），所以当 transfer size 被读取时其返回的值为目的设备端完成的传输数据个数乘以一个因子（为源设备/目的设备传输位宽之比），该个数储存在 trfSizeDst 的计数器中。
- 在一次 peripheral-to-peripheral 传输中，若目的外设作为流控制器且 $DWidth < SWidth$ ，则目的设备请求传输的数据个数（以字节数计算）必须是 SWidth（以字节数计算）的整数倍，否则 DMAC 会从源外设中取更多的数据，这将造成数据丢失。
- 当 transfer size 字段的设置值大于源外设或目的外设中的 FIFO 的深度（是外设的 FIFO，不是 DMAC 的 FIFO），DMAC 的源地址或目的地址必须被设为不递增模式，否则有可能导致外设的 FIFO 溢出。

访问保护和访问信息

AHB 访问信息在传输发生时由 master 接口信号提供给源外设或目的外设。这些访问信息是通过对通道寄存器编程设定的 (DMAC_CxCONTROL[Prot]和 DMAC_CxCONFIGURATION[Lock])。表 7-6 给出了使用这 3 个保护位的含义。



表7-6 DMAC_CxCONTROL 寄存器 Prot 段属性及定义

比特	名称	描述
[2]	Cacheable or noncacheable	<p>指明访问的是 cacheable 还是 noncacheable。</p> <p>0: noncacheable; 1: cacheable。</p> <p>例如, 该位可用于告知一个 AMBA 桥: 当其发现 8 个数的 burst 读的第 1 个读操作时, 该桥可在目标总线上直接发起一个 8 个数的 burst 读, 而不用将源总线上的读操作一次一个的传到目标总线。</p> <p>该位控制 AHB 总线信号 HPROT[3]的输出。</p>
[1]	Bufferable or nonbufferable	<p>指明访问是可缓冲的还是不可缓冲的。</p> <p>0: nonbufferable; 1: bufferable。</p> <p>例如, 该位可用于告知一个 AMBA 桥在源端总线上写操作可以以零等待状态完成, 而无需等该桥把操作仲裁到目的总线上, 也无需等 slave 接收完数据。</p> <p>该位控制 AHB 总线信号 HPROT[2]的输出。</p>
[0]	Privileged or User	<p>指明访问是 user 模式的还是 privileged 模式的。</p> <p>0: user mode; 1: privileged mode。</p> <p>该位控制 AHB 总线信号 HPROT[1]的输出。</p>

7.7.22 DMAC_CxCONFIGURATION

通道配置寄存器 DMAC_CxCONFIGURATION 的地址为 $0x110 + N \times 0x20$, 其中 N 的取值为 0~7, 分别对应 DMA 通道 0~DMA 通道 7。

8 个 DMAC_CxCONFIGURATION 分别用于配置各自的 DMA 通道。

该寄存器在新的 LLI 被载入时不会被更新。

- 偏移地址: $0x110 + N \times 0x20$
- 复位方式: h/s
- 复位值: 0x0



注意

当刚通过写 Channel Enable 位关闭一个通道时，必须要等到轮询到寄存器 DMAC_ENBLDCHNS 中的相应位为 0 之后才能将 Channel Enable 位重新置位。这是因为通道实际的关闭并没有在将 Channel Enable 位清零后立即生效。AHB burst 的运行延时必须要考虑到。

比特	名称	操作类型	描述
[31:21]	Reserved	-	保留。 写入时必须写入 0，读出时被屏蔽。
[20]	ITC1	R/W	Terminal count 中断的第 1 屏蔽位。 当该位被清零时，本通道的 INTTC2 中断输出被屏蔽。
[19]	IE1	R/W	Error 中断的第 1 屏蔽位。 当该位被清零时，本通道的 INTERR2 中断被屏蔽。
[18]	H	R/W	Halt 位。 0: 允许 DMA 请求； 1: 忽略后来的 DMA 请求，通道 FIFO 中的内容都被传完。 该位可以和 Active 位以及 Channel Enable 位一起用于无数据丢失地关闭一个 DMA 通道。
[17]	A	R	Active 位。 0: 通道 FIFO 中没有数据； 1: 通道 FIFO 中有数据。 该位可以和 Halt 位以及 Channel Enable 位一起用于无数据丢失地关闭一个 DMA 通道。
[16]	L	R/W	Lock 位。 当被置位时在 AHB 总线上使用 lock 传输。
[15]	ITC0	R/W	Terminal count 中断的第 0 屏蔽位。 当该位被清零时，本通道的 terminal count 中断被屏蔽。
[14]	IE0	R/W	Error 中断的第 0 屏蔽位。 当该位被清零时，本通道的 error 中断被屏蔽。



比特	名称	操作类型	描述
[13:11]	FlowCtrl	R/W	<p>流控及传输类型字段。</p> <ul style="list-style-type: none"> 流控制器可以是 DMAC、源外设和目的外设。 传输类型可以是 memory-to-peripheral、peripheral-to-memory、peripheral-to-peripheral、memory-to-memory。详细描述请参见表 7-7。
[10]	Reserved	-	<p>保留。</p> <p>写入时必须写入 0；读出时被屏蔽。</p>
[9:6]	DestPeripheral	R/W	<p>Destination Peripheral。</p> <p>该字段用于选择一个外设请求信号作为本通道的 DMA 目的外设的请求信号。</p> <p>如果 DMA 传输的目的设备是存储器则该字段被忽略。</p>
[5]	Reserved	-	<p>保留。</p> <p>写入时必须写入 0，读出时被屏蔽。</p>
[4:1]	SrcPeripheral	R/W	<p>Source Peripheral。</p> <p>该字段用于选择一个外设请求信号作为本通道的 DMA 源外设的请求信号。</p> <p>如果 DMA 传输的源设备是存储器则该字段被忽略。</p>



比特	名称	操作类型	描述
[0]	E	R/W	<p>Channel Enable 位。</p> <p>0: 通道被关闭;</p> <p>1: 通道被启动。</p> <p>启动通道:</p> <ul style="list-style-type: none"> • 通过将该位置位可启动通道。 • 若简单的将该位置位以启动通道就会引发不可预测性的后果, 所以必须先重新初始化通道才能再次启动通道。 <p>关闭通道:</p> <ul style="list-style-type: none"> • 通过将该位清零可关闭通道。将该位被清零时, 当前的 AHB 传输会继续执行直到完成; 通道被关闭, FIFO 中剩余的数据全部丢失。 • 当最后一个 LLI 完成或传输中出现错误时, 通道也会被关闭, 同时该位被清零。 • 如果要关闭通道而又不想通道 FIFO 中的数据丢失, 则 Halt 位也必须同时被置位, 从而使通道忽略后来的 DMA 请求。其次必须轮询 Active 位直到其值变为 0, 表明通道 FIFO 中不再留有数据。此时才能够清除 Enable 位。 <p>读该寄存器和 DMAC_ENBLDCHNS 可得知本通道当前是处于什么状态。</p>

表 7-7 描述了 DMAC_CxCONFIGURATION 寄存器的 FlowCntrl 字段对应的流控和传输类型。

表7-7 流控及传输类型说明

比特	传输类型	控制器
000	Memory to memory	DMAC
001	Memory to peripheral	DMAC
010	Peripheral to memory	DMAC
011	Source peripheral to destination peripheral	DMAC
100	Source peripheral to destination peripheral	Destination peripheral
101	Memory to peripheral	Peripheral
110	Peripheral to memory	Peripheral
111	Source peripheral to destination peripheral	Source peripheral



8 TIMER、WatchDog 和 RTC

关于本章

本章描述内容如下表所示。

标题	内容
8.1 TIMER	介绍 TIMER 单元。
8.2 WatchDog	介绍 WatchDog 单元。
8.3 RTC	介绍 RTC 单元。

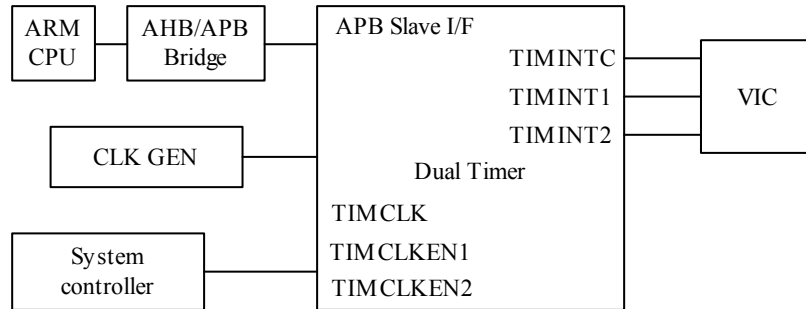


8.1 TIMER

8.1.1 概述

TIMER 单元提供 4 个计数器，用于产生中断输出。TIMER 功能框图如图 8-1 所示。

图8-1 TIMER 功能框图



8.1.2 特点

TIMER 单元有以下特点：

- 4 个 32 位/16 位下跳计数器，具有 free-running、periodic 和 one-shot 模式
- 共用时钟，4 个 TIMER 可以独立进行时钟使能，灵活的给出时钟间隔
- TIMER 计数到达 0 时，产生中断输出

8.1.3 寄存器概览

TIMER 单元包括 2 组寄存器：TIMER12 和 TIMER34，寄存器地址位宽都是 32 位。

📖 说明

- TIMER12 中 1、2 分别对应 TIMER1、TIMER2
- TIMER34 中 3、4 分别对应 TIMER3、TIMER4

TIMER1 与 TIMER2 在一组寄存器 TIEMR12 中，地址范围是 0x101E_2000~0x101E_2FFF。TIEMR12 寄存器概览如表 8-1 所示。

- TIMER1 的偏移地址是 0x000
- TIMER2 的偏移地址是 0x020

TIMER3 与 TIMER4 在一组寄存器 TIMER34 中，地址范围是 0x101E_3000~0x101E_3FFF。TIMER34 寄存器概览如表 8-2 所示。

- TIMER3 的偏移地址是 0x000
- TIMER4 的偏移地址是 0x020



表8-1 TIMER12 寄存器概览（基址是 0x101E_2000）

TIMER1 的偏移地址	TIMER2 的偏移地址	寄存器名称	页码
0x000	0x020	TIMERx_LOAD	8-3
0x004	0x024	TIMERx_VALUE	8-4
0x008	0x028	TIMERx_CONTROL	8-5
0x00C	0x02C	TIMERx_INTCTR	8-6
0x010	0x030	TIMERx_RIS	8-7
0x014	0x034	TIMERx_MIS	8-7
0x018	0x038	TIMERx_BGLOAD	8-7

表8-2 TIMER34 寄存器概览（基址是 0x101E_3000）

TIMER3 的偏移地址	TIMER4 的偏移地址	寄存器名称	页码
0x000	0x020	TIMERx_LOAD	8-3
0x004	0x024	TIMERx_VALUE	8-4
0x008	0x028	TIMERx_CONTROL	8-5
0x00C	0x02C	TIMERx_INTCTR	8-6
0x010	0x030	TIMERx_RIS	8-7
0x014	0x034	TIMERx_MIS	8-7
0x018	0x038	TIMERx_BGLOAD	8-7

8.1.4 寄存器描述

本节描述了 TIMER12 寄存器的功能，TIMER12 和 TIMER34 的各个寄存器偏移地址和功能相同，TIMER34 寄存器的功能请参照 TIMER12 寄存器的功能。

TIMERx_LOAD



TIMERx_LOAD 中 x 表示 1 或 2，分别对应 TIMER1、TIMER2；

TIMERx_VALUE ~ TIMERx_BGLOAD 各寄存器中的 x 表示的含义如上所述。

TIMERx_LOAD 寄存器是计数初值寄存器，保存着定时器的计数初值。

**注意**

向 `TIMERx_LOAD` 寄存器写入的最小有效值为 1；当向 `TIMERx_LOAD` 写 0 时，Dual-Timer 将会立刻产生一个中断。

当定时器处于 Periodic 模式，且计数值递减到 0 时，就将 `TIMERx_LOAD` 的值重新载入计数器。当直接写该寄存器时，定时器当前的计数器将在 `TIMCLKENX` 使能的下一个 `TIMCLK` 的上升沿更新为写入值。

当向 `TIMERx_BGLOAD` 寄存器写入值时，`TIMERx_LOAD` 的值也会被覆盖，但定时器计数的当前值不会受到影响。

若在 `TIMCLKENX` 使能的 `TIMCLK` 的上升沿到来之前，向 `TIMERx_BGLOAD` 寄存器和 `TIMERx_LOAD` 寄存器都写入值，则

1. 在 `TIMCLKENX` 使能的 `TIMCLK` 的下一个上升沿到来时，定时计数器的值更新为 `TIMERx_LOAD` 的写入值；
2. 每当计数器递减到 0 时，其值重载为 `TIMERx_BGLOAD` 与 `TIMERx_LOAD` 中最后被写入时的值。

在分别对 `TIMERx_BGLOAD` 寄存器和 `TIMERx_LOAD` 寄存器进行了 2 次写入之后，

- 读 `TIMERx_LOAD` 返回的值为 `TIMERx_BGLOAD` 的写入值。
- 读 `TIMERx_LOAD` 的返回值总为 Periodic 模式下的下次计数器递减到 0 时将载入的有效值。
- 偏移地址：
 - `TIMER1_LOAD`: 0x000
 - `TIMER2_LOAD`: 0x020
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	<code>TIMER1_LOAD</code>	定时器 1 的计数初值。
[31:0]	<code>TIMER2_LOAD</code>	定时器 2 的计数初值。

TIMERx_VALUE

`TIMER1` 和 `TIMER2` 各有一个 `VALUE` 寄存器，二者的功能相同。

`TIMERx_VALUE` 寄存器是当前计数值寄存器，给出正在递减的定时计数器的当前值。

当向 `TIMERx_LOAD` 寄存器的写操作发生后，`TIMERx_VALUE` 在 `PCLK` 时钟域立刻反映出计数器的新载入值，无需等到下一个被 `TIMCLKENX` 使能的 `TIMCLK` 时钟沿。



注意

当定时器处于 16 位模式时，32 位的 TIMERx_LOAD 寄存器的高 16 位并未被自动设为 0。若该定时器以前处于 32 位模式并且自从进入 16 位模式，TIMERx_LOAD 从未被写过，则 TIMERx_LOAD 寄存器的高 16 位可能具有非零值。

- 偏移地址：
 - TIMER1_VALUE: 0x004
 - TIMER2_VALUE: 0x024
- 操作类型: R
- 复位值: 0x1
- 复位方式: h/s

比特	名称	描述
[31:0]	TIMER1_VALUE	正在递减的定时计数器 1 的当前值。
[31:0]	TIMER2_VALUE	正在递减的定时计数器 2 的当前值。

TIMERx_CONTROL

TIMERx_CONTROL 是 TIMER 控制寄存器。TIMER1 和 TIMER2 各有一个 CONTROL 寄存器，二者除偏移地址不同外其他特性相同。

- 偏移地址：
 - TIMER1_CONTROL: 0x008
 - TIMER2_CONTROL: 0x028
- 操作类型: R/W
- 复位值: 0x20
- 复位方式: h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7]	TimerEn	计数器使能位（写入时）。 0: 关闭定时器模块（缺省值）； 1: 开启定时器模块。 读时返回本字段的当前值。



比特	名称	描述
[6]	TimerMode	定时器的计数模式位。 0: 定时器模块处于 free-running mode (缺省值) ; 1: 定时器模块处于 periodic mode。 读时返回本字段的当前值。
[5]	IntEnable	中断屏蔽位 (写入时)。 0: 不允许定时器模块发出中断; 1: 允许定时器模块发出中断 (缺省值)。 读时返回本字段的当前值。
[4]	Reserved	保留。写入无效, 读时返回 0。
[3:2]	TimerPre	该字段用于设置 TIMER 的预分频因子。 00: 不经过预分频, 时钟频率除以 1 (缺省值) ; 01: 4 级预分频, 将 TIMER 时钟频率除以 16; 10: 8 级预分频, 将 TIMER 时钟频率除以 256; 11: 未定义, 若设为该值, 相当于预分频因子等于 10。
[1]	TimerSize	选择 16 位/32 位计数器操作模式。 0: 16 位计数器 (缺省值) ; 1: 32 位计数器。
[0]	OneShot	选择计数模式为一次性计数或者回卷计数。 0: 回卷 (wrapping) 计数 (缺省值) ; 1: 一次性计数。

TIMERx_INTCLR

TIMERx_INTCLR 是中断清除寄存器。TIMER1 和 TIMER2 各有一个 INTCLR 寄存器, 二者除偏移地址不同外其他特性相同。对该寄存器的任何写操作都会清除相应计数器的中断状态。

- 偏移地址:
 - TIMER1_INTCLR: 0x00C
 - TIMER2_INTCLR: 0x02C
- 操作类型: W
- 复位值: -
- 复位方式: h/s

偏移地址	名称	描述
0x00C	TIMER1_INTCLR	写该寄存器, 清除计数器的中断输出。



偏移地址	名称	描述
0x02C	TIMER2_INTCLR	写该寄存器，清除计数器的中断输出。

TIMERx_RIS

TIMERx_RIS 是原始中断寄存器。TIMER1 和 TIMER2 各有一个 RIS 寄存器，二者除偏移地址不同外其他特性相同。

- 偏移地址：
 - TIMER1_RIS: 0x010
 - TIMER2_RIS: 0x030
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:1]	Reserved	保留。写入无效，读时返回 0。
[0]	TimerXRIS	读时返回计数器的原始中断状态。

TIMERx_MIS

TIMERx_MIS 是屏蔽后中断寄存器。TIMER1 和 TIMER2 各有一个 MIS 寄存器，二者除偏移地址不同外其他特性相同。

- 偏移地址：
 - TIMER1_MIS: 0x014
 - TIMER2_MIS: 0x034
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:1]	Reserved	保留。
[0]	TimerXMIS	读时返回屏蔽后的计数器中断状态。

TIMERx_BGLOAD

TIMERx_BGLOAD 是循环模式计数初值寄存器。TIMER1 和 TIMER2 各有一个 BGLoad 寄存器，二者除偏移地址不同外其他特性相同。



TIMERx_BGLOAD 寄存器中包含了定时器的计数初值。该寄存器用于在循环模式下，当定时器的计数值递减到 0 时重新载入计数初值。

TIMERx_BGLOAD 寄存器提供了访问 TIMERx_LOAD 寄存器的另一种方法，在写入 TIMERx_BGLOAD 寄存器时，不会导致定时器立即写入值重新开始计数。

- 偏移地址：
 - TIMER1_BGLOAD: 0x018
 - TIMER2_BGLOAD: 0x038
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	TimerXBGLoad	在 periodic 模式下，当定时器的计数值递减到 0 时重新载入计数初值。

8.2 WatchDog

8.2.1 概述

系统看门狗（WatchDog）用于系统异常恢复。如果没有收到更新，会隔一定时间（可编程）产生一系统复位信号，而在此之前可以屏蔽复位或者在复位信号产生之前更新计数器。

8.2.2 特点

WatchDog 有 1 个可编程 32 位定时器，有以下特点：

- 支持超时间隔、初始值可配，计数器支持 free-running 递减模式，锁定寄存器来防止寄存器异常；
- 支持时钟使能，控制计数时间间隔；
- 支持超时中断发生，若中断未被响应，再次超时则产生复位信号；
- 支持中断屏蔽功能；
- 支持时钟使能控制，关断 WatchDog 时钟可降低系统功耗。

8.2.3 寄存器概览

WatchDog 的寄存器的地址位宽 32 位，地址范围：0x101E_1000~0x101E_1FFF。



表8-3 WatchDog 寄存器概览（基址是 0x101E_1000）

偏移地址	寄存器名称	描述	页码
0x000	WDT_LOAD	计数初值寄存器	8-9
0x004	WDT_VALUE	正在递减计数的计数器当前值寄存器	8-9
0x008	WDT_CONTROL	控制寄存器	8-10
0x00C	WDT_INTCLR	中断清除寄存器	8-10
0x010	WDT_RIS	原始中断寄存器	8-10
0x014	WDT_MIS	屏蔽后中断寄存器	8-11
0xC00	WDT_LOCK	LOCK 寄存器	8-11

8.2.4 寄存器描述

本节详细描述了 WatchDog 的寄存器。

WDT_LOAD

- 偏移地址：0x000
- 操作类型：R/W
- 复位值：0xFFFF_FFFF
- 复位方式：h/s

比特	名称	描述
[31:0]	WDT_Load	看门狗的计数初值，计数时钟频率为 ARM926 处理器频率的 1/2。

WDT_VALUE

- 偏移地址：0x004
- 操作类型：R/W
- 复位值：0xFFFF_FFFF
- 复位方式：h/s

比特	名称	描述
[31:0]	WDT_Value	该寄存器给出正在递减计数的计数器当前值。



WDT_CONTROL

- 偏移地址：0x008
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:2]	Reserved	保留。
[1]	RESEN	允许 WatchDog 单元输出复位信号 WDOGRES，相当于复位信号的屏蔽信号。 0：不允许复位输出； 1：允许复位输出。
[0]	INTEN	中断使能信号，允许中断信号 WDOGINT 输出。 0：计数器停止计数，计数值保持当前值不变； 1：启动计数器且使能中断。 若中断已被禁止，则当中断再次被使能时，计数器从 WDT_LOAD 中载入初值并重新开始计数。

WDT_INTCLR

- 偏移地址：0x00C
- 操作类型：W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:0]	WDT_IntClr	对该寄存器写入任意值可清除中断，并使寄存器从 WDT_LOAD 中重新载入初值。

WDT_RIS

- 偏移地址：0x010
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:1]	Reserved	保留。



比特	名称	描述
[0]	WDT_RIS	计数器的原始中断状态。 当计数器的计数值减到 0 时，该位被置 1。

WDT_MIS

- 偏移地址：0x014
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:1]	Reserved	保留。
[0]	WDT_MIS	计数器的屏蔽后中断状态。

WDT_LOCK

- 偏移地址：0xC00
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:0]	WDT_LOCK	写该寄存器可以控制所有寄存器的写权限。 <ul style="list-style-type: none"> • 0x1ACC_E551：可打开所有寄存器的写权限； • 其他：关闭所有寄存器的写权限 读该寄存器返回加锁的状态而不是写入该寄存器的值： <ul style="list-style-type: none"> • 0x0000_0000：写访问是允许的（未加锁）； • 0x0000_0001：写访问是禁止的（已加锁）。

8.3 RTC

8.3.1 概述

实时时钟（RTC）通过对晶振所产生的振荡频率分频和累加，得到年、月、日、时、分、秒等时间信息。



8.3.2 特点

RTC 单元有以下特性：

- 使用 32.768kHz 晶振的 2^{15} 分之一分频，即 1Hz 作为计数时钟；
- 支持 32 位递增自由式计数器，支持告警功能；
- 支持 32 位比较寄存器的值可配。当计数器值和预设值相等时，中断发生，中断可屏蔽。

8.3.3 寄存器概览

RTC 寄存器的地址位宽 32 位，地址范围：0x101E_8000~0x101E_8FFF。

表8-4 RTC 寄存器概览（基址是 0x101E_8000）

偏移地址	寄存器名称	描述	页码
0x000	RTC_DR	数据寄存器	8-12
0x004	RTC_MR	比较寄存器	8-13
0x008	RTC_LR	加载寄存器	8-13
0x00C	RTC_CR	使能寄存器	8-13
0x010	RTC_IMSC	中断屏蔽设置/清除寄存器	8-14
0x014	RTC_RIS	原始中断状态寄存器	8-14
0x018	RTC_MIS	屏蔽后中断状态寄存器	8-14
0x01C	RTC_ICR	中断清除寄存器	8-14

8.3.4 寄存器描述

本节详细描述了 RTC 的寄存器。

RTC_DR

- 偏移地址：0x000
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:0]	RTC data register	返回当前 RTC 值。



RTC_MR

- 偏移地址：0x004
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:0]	RTC match register	比较寄存器，当计数器值与比较寄存器值相等时，产生中断信号。

RTC_LR

- 偏移地址：0x008
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:0]	RTC load register	加载寄存器。

RTC_CR

- 偏移地址：0x00C
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:1]	Reserved	保留。
[0]	RTC start	1: RTC 处于使能状态。 一旦使能，则对它的任何写操作都已经不起作用（系统复位除外）。 读该寄存器返回 RTC 当前值。

RTC_IMSC

- 偏移地址：0x010
- 操作类型：R/W
- 复位值：0x0



- 复位方式：h/s

比特	名称	描述
[31:1]	Reserved	保留。
[0]	RTC_IMSC	0: 清除屏蔽; 1: 设置屏蔽。

RTC_RIS

- 偏移地址：0x014
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:1]	Reserved	保留。
[0]	RTC_RIS	给出 RTC_INTR 中断的原始中断状态（屏蔽之前的状态）。

RTC_MIS

- 偏移地址：0x018
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:1]	Reserved	保留。
[0]	RTC_MIS	给出 RTC_INTR 中断的中断状态（屏蔽之后的状态）。

RTC_ICR

- 偏移地址：0x01C
- 操作类型：W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:1]	Reserved	保留。



比特	名称	描述
[0]	RTC_ICR	清除 RTC_INTR 中断。 0: 无效; 1: 清除中断。



9 视频编解码单元

关于本章

本章描述内容如下表所示。

标题	内容
9.1 概述	概括介绍视频编解码单元。
9.2 视频编解码协处理器	介绍视频编解码协处理器。
9.3 视频编解码器	介绍视频编码器、解码器。



9.1 概述

视频编解码单元包括视频编解码协处理器和视频编解码器，实现对多协议视频编解码过程的硬件加速。

9.2 视频编解码协处理器

Hi3510 包含了一系列视频编解码协处理器，用于提高视频编解码速度。

视频编解码协处理器有以下特点：

- 高位数 0 协处理器
用于对输入的 32 位数的 0 的个数（指从最高位开始的连续的 0 的个数）进行计数。
- 高位数 1 协处理器
用于对输入的 32 位数中所有的值为 1 的位进行计数。
- 位倒置协处理器，用于将输入数的高低位倒置，即
 - 输入数的 bit[31]变为输出数的 bit[0]
 - 输入数的 bit[30]变为输出数的 bit[1]
 - 输入数的 bit[29]变为输出数的 bit[2]
 -
- 字节倒置协处理器，用于将输入的 32 位数的 4 个字节倒换位置，即
 - 输入数中的 byte[3]变为输出数的 byte[0]
 - 输入数中的 byte[2]变为输出数的 byte[1]
 - 输入数中的 byte[1]变为输出数的 byte[2]
 - 输入数中的 byte[0]变为输出数的 byte[3]
- H264 VLC 系数编码协处理器
用于产生完成 H.264 残差系数对应的 CAVLC 语法结构。
- H264 VLD 系数解码协处理器
用于根据 H.264 参考块的 CAVLC 语法结构计算出残差系数。
- getBits 协处理器
用于从码流中截取出用于 VLD 解码的变长码字。
- 查表协处理器
用于实现查表功能，通过查表得出与 H.264 的某些语法元素对应的变长码字。
- 视频编码/视频解码接口协处理器
用于读写视频编解码模块中的寄存器，启动视频编解码的硬件加速功能。



9.3 视频编解码器

视频编解码器单元实现了 H.264、H.263、H.261 协议标准规定的：

- 运动估计/运动补偿
- DCT/IDCT 变换
- 量化/反量化
- Zigzag 扫描
- de-blocking 滤波
- 环路滤波

视频编解码器单元还提供了 DSP 协处理器接口，具有视频的硬件加速功能。

9.3.1 视频编码器

视频编码器特点如下：

- 支持的图像格式：D1、CIF、QCIF；码率最高可达到 D1@30fps，具有 4Mbit/s 的图像编码能力；
- 支持 H.264 baseline profile；
- H.264 标准模式下支持 1/4 象素精度运动估计；
- 支持 H.263 的 4MV（选项 F）、de-blocking 选项（选项 J）、MQ 选项（选项 T）及 SEI 选项（选项 W、选项 L）；
- H.263 标准模式下支持 1/2 象素精度运动估计；
- 支持 H.261 编码，支持 H.261 环路滤波；
- 运动估计采用全搜索方式；
- 码率范围可调，支持按照每个宏块调整量化步长的码率控制策略；
- 提供到 DSP 的中断。

9.3.2 视频解码器

视频解码器特点如下：

- 支持 H.264 Baseline Profile、Level 2.2，并支持 D1（720×576）@30fps,4Mbit/s 的解码能力；
- 支持 H.263+，支持 4MV（选项 F）、UMV 选项（选项 D）、MQ 选项（选项 T）及 de-blocking 选项（选项 J）；
- 支持 H.261 解码，支持 H.261 环路滤波；
- 提供到 DSP 的中断。



10 2D 图形加速引擎(TDE)

关于本章

本章描述内容如下表所示。

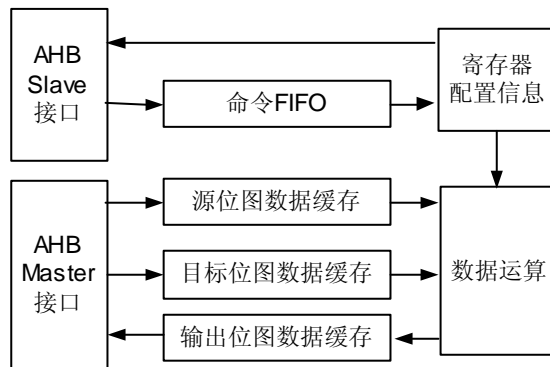
标题	内容
10.1 概述	概括的介绍 TDE。
10.2 功能描述	简单介绍 TDE 的功能。
10.3 工作方式	简单介绍中断和任务查询、复位和配置。
10.4 寄存器概览	概括的介绍 TDE 的寄存器。
10.5 寄存器描述	详细描述 TDE 的寄存器。



10.1 概述

TDE (Two Dimensional Engine) 利用硬件绘制图形, 可以大大减少对 CPU 的占用, 同时提高内存带宽的资源利用率。TDE 通过 AHB Master 总线接口读写位图数据, 通过 AHB Slave 总线接口获得 CPU 的寄存器配置信息。TDE 单元的功能框图如图 10-1 所示。

图10-1 TDE 功能框图



10.2 功能描述

本节主要从以下四个方面来描述 TDE 单元的功能。

- [特点](#)
- [支持数据格式](#)
- [术语描述](#)
- [支持功能](#)

10.2.1 特点

TDE 单元有以下特点:

- 源位图、目标位图和输出位图支持 RGB 444、ARGB 4444、RGB 555、RGB 565、ARGB 1555、RGB 888、ARGB 8888 像素格式
- 支持位块传送 (Blit)
- 支持在固定矩形区域的 pattern fill
- 支持硬件画垂直、水平直线
- 支持硬件矩形填充
- 支持 alpha blending 操作
- 支持 1/4/8 bit 的 alpha 值, 支持像素 alpha 值的操作和全局 alpha 值操作
- 支持源位图的 color space 操作



- 支持 ROP2
- 支持地址 half word(32bit word)操作

10.2.2 支持数据格式

TDE 支持的像素格式包括 RGB 444、ARGB 4444、RGB 555、RGB 565、ARGB 1555、RGB 888、ARGB 8888，具体存储方式参见 VOU 的像素格式。TDE 内部运算时，每个像素的每个分量是以 8bit 位宽运算。对于不足 8bit 的分量，将最高位填充到低位。如支持 RGB444 像素格式时，某像素值为 R:0x3、G:0x8、B:0xF，则扩展后的像素分别为 R:0x30、G:0x8F、B:0xFF。

10.2.3 术语描述

源位图

- blit 操作时，源位图指 TDE 读入的前景位图
- pattern fill 操作时，源位图指 TDE 读入的 pattern 位图

目标位图

如果 blit、pattern fill 或者 solid draw 操作时有 alpha blending 或者 ROP2，则需要读入背景位图，该前景位图即目标位图。

输出位图

源位图和目标位图运算后，经过搬移后就得到输出位图。

blit

位块传送 (bit block transfer)，指将位图数据从一个设备存储区域搬移到另外一个设备存储区域的过程。例如，blit 可以将位图从内存区域搬移到显示区域。blit 可以带有 alpha blending、color space 和 ROP2 等属性。

pattern fill

用一个固定大小的位图 pattern (例如 8×8 像素) 来平铺填充存储空间的某一区域。pattern fill 可以带有 alpha blending、color space 和 ROP2 等属性。

solid draw

硬件画线/矩形，指在给定的区域内，按照指定的大小和颜色画线/矩形。solid draw 可以带有 alpha blending 和 ROP2 等属性。

alpha blending

对多个位图的像素值按照 alpha 值进行加权求和，可得到一个 alpha 混和后的输出位图。TDE 将各种格式像素中的 alpha 值转换为 8bit (不足 8bit 的低位填 0)，根据 8bit 的 alpha 值叠加。当 alpha 的最高位为 1 时， $P_{output} = P_{source}$ ；当 alpha 的最高位为 0



时, $P_{output} = P_{source} * alpha + P_{destination} * (1 - alpha)$ 。因此 alpha[7]为整数位, alpha[6:0]为小数位。

ROP2

指对 2 种位图 (如源位图和目标位图、pattern 和目标位图) 的颜色值进行布尔运算, 运算操作符的详细描述请参见表 10-3。

color space

关键色, 指在设定的颜色下限值 (colorspace_min) 和颜色上限值 (colorspace_max) 之间的颜色。在 blit 操作中, 将具有这种颜色的区域视为透明。需要注意的是, colorspace_min 和 colorspace_max 分 RGB 三分量设置, 每个分量都是 8bit, 如输入像素格式不足 8bit, TDE 将会对其扩展 (请参见“10.2.2 支持数据格式”) 后再与 colorspace_min 和 colorspace_max 进行比较。

clip

剪裁。TDE 支持将超出输出位图大小范围的 pattern 裁掉。

像素 alpha 值和全局 alpha 值

像素 alpha 值指每一个像素都有单独的 alpha 值, 如 ARGB4444 像素格式; 全局 alpha 值是指整个位图使用相同的 alpha 值, 该 alpha 值可通过寄存器配置得到, 而不是通过位图像素得到。

10.2.4 支持功能

- blit
将位图从源地址搬移到目标地址, 直接覆盖目标区域数据。
- blit 且 ROP2
将位图从源地址搬移到目标地址, 并且与目标区域数据进行 ROP2。
- blit 且 alpha blending
将位图从源地址搬移到目标地址, 并且与目标区域数据进行 alpha blending。
- blit 且 color space
将位图从源地址搬移到目标地址, 在此过程中, 满足 colorspace_min ≤ 像素颜色值 ≤ colorspace_max 的位图像素不写入对应目标区域。
- blit 且 color space 和 alpha blending
将位图从源地址搬移到目标地址, 在此过程中, 满足 colorspace_min ≤ 像素颜色值 ≤ colorspace_max 的位图像素不写入对应目标区域, 写入数据与目标区域数据进行 alpha blending。
- blit 且 color space 和 ROP2
将位图从源地址搬移到目标地址, 在此过程中, 满足 colorspace_min ≤ 像素颜色值 ≤ colorspace_max 的位图像素不写入对应目标区域, 写入数据与目标区域数据进行 ROP2。
- pattern fill



用一个固定大小的位图 `pattern`（例如 8×8 像素）来平铺填充存储空间的某一区域。

- **pattern fill 且 ROP2**
将数据块重复填充到目标区域，并且与目标区域的数据进行 ROP2。
- **pattern fill 且 alpha blending**
将数据块重复填充到目标区域，并且与目标区域的数据进行 alpha blending。
- **pattern fill 且 color space**
将数据块重复填充到目标区域。在此过程中，满足 $\text{colorspace_min} \leq \text{像素颜色值} \leq \text{colorspace_max}$ 的 `pattern` 像素不写入对应目标区域。
- **pattern fill 且 color space 和 alpha blending**
将数据块重复填充到目标区域。在此过程中，满足 $\text{colorspace_min} \leq \text{像素颜色值} \leq \text{colorspace_max}$ 的 `pattern` 像素不写入对应目标区域；写入的数据与目标区域的数据进行 alpha blending。
- **pattern fill 且 color space 和 ROP2**
将数据块重复填充到目标区域。在此过程中，满足 $\text{colorspace_min} \leq \text{像素颜色值} \leq \text{colorspace_max}$ 的 `pattern` 像素不写入对应目标区域，写入数据与目标区域数据进行 ROP2。
- **solid draw**
根据配置的线宽、线高和颜色值，画水平直线、垂直直线、矩形。
- **solid draw 且 alpha blending**
根据配置的线宽、线高和颜色值，画水平直线、垂直直线、矩形，并且线/矩形与目标区域数据进行 alpha blending。
- **solid draw 且 ROP2**
根据配置的线宽、线高和颜色值，画水平直线、垂直直线、矩形，并且线/矩形与目标区域数据进行 ROP2。

10.3 工作方式

本节主要从以下三个方面来描述 TDE 单元的工作方式。

- [中断](#)
- [复位](#)
- [配置](#)



10.3.1 中断

TDE 不产生任何中断。由于 TDE 处理一次任务不一定生成整个显示位图，而可能仅仅是位图中的一条线、一个矩形等，因此 TDE 没有必要频繁的产生中断。

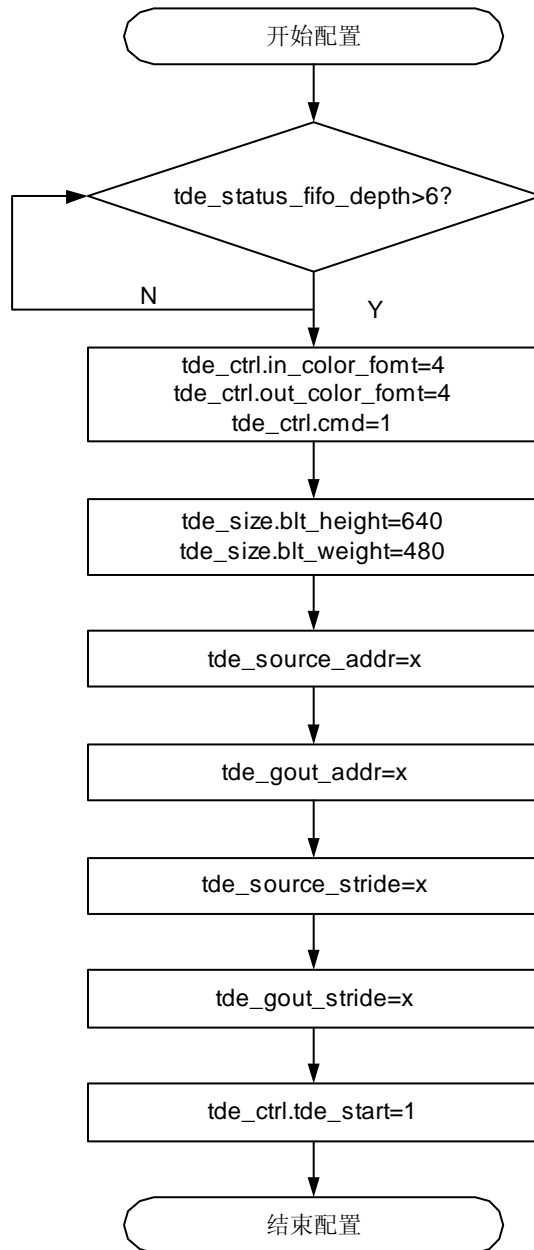
10.3.2 复位

TDE 的复位方式分为硬复位和软复位，软复位是通过系统控制器中的 SC_PERCTRL1[TdeSrstReqn]控制。可读写寄存器（偏移地址 0x300~0x324）仅支持硬件复位，其他寄存器软硬复位均可。

10.3.3 配置

图 10-2 以 blit 操作举例说明 CPU 如何对 TDE 配置。其中 tde_source_addr、tde_gout_addr 分别是源位图、输出位图的起始地址，tde_source_stride 和 tde_gout_stride 分别是源位图、输出位图的行间距，请参见图 10-3。

图10-2 TDE 配置举例



TDE 内部有一个深度为 32word 的命令 FIFO 缓存（命令 FIFO 位置请参见图 10-1），CPU 向 TDE 发出配置命令。如果命令 FIFO 非空，且 TDE 处于空闲状态，则 TDE

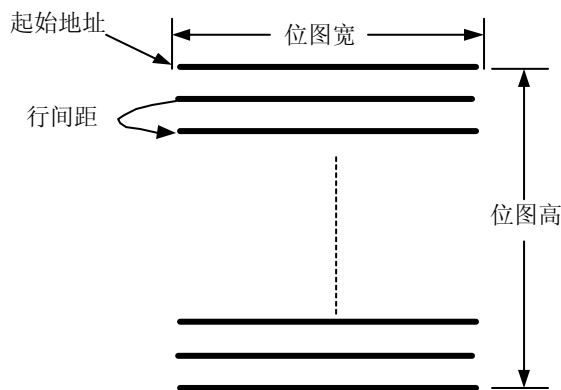
- 从命令 FIFO 中读取命令
- 读到 TDE_CTRL[tde_start]=1 后开始执行此次任务，并暂停读取命令。此时
 - CPU 可以继续向命令 FIFO 中写入命令，直到 FIFO 没有足够的空间（可以通过 TDE_STATUS[fifo_depth]判断）。
 - 当 FIFO 空间少于 3 个 word 时就会报警（TDE_STATUS[history_fifooverflow]=1）。



如果前后两次配置只有某几个部分是不同的，则只需配置不同的部分即可。例如：两次 blit 的起始地址、行间距、像素格式可能不变，只是位图大小变化，因此只需配置位图大小，并置位 TDE_CTRL[tde_start]即可。

图 10-3 说明了位图数据在存储空间中的存储方式和位图关键配置信息。

图10-3 TDE 位图存储示意图



10.4 寄存器概览

TDE 寄存器的地址位宽 32 位，地址范围是 0x9001_0000~0x9001_FFFF。

表10-1 TDE 寄存器概览（基址是 0x9001_0000）

偏移地址	名称	描述	页码
0x300	TDE_CTRL	控制寄存器	10-9
0x304	TDE_SIZE	位图大小寄存器	10-11
0x308	TDE_SOURCE_ADDR	源位图起始地址	10-11
0x30C	TDE_DET_ADDR	目标位图起始地址	10-12
0x310	TDE_GOUT_ADDR	输出位图起始地址	10-12
0x314	TDE_SOURCE_STRIDE	源位图行间距	10-12
0x318	TDE_GOUT_STRIDE	输出位图行间距	10-13
0x31C	TDE_COLOR_FILLED	颜色填充值	10-13
0x320	TDE_COLOR_MAX	Color space 最大值	10-14
0x324	TDE_COLOR_MIN	Color space 最小值	10-14
0x328~0x37C	RESERVED	保留	-
0x380	TDE_STATUS	TDE 状态寄存器	10-15



10.5 寄存器描述

本节详细描述了 TDE 单元的寄存器。

10.5.1 控制寄存器 (TDE_CTRL)

控制寄存器用于配置 TDE 的一些全局参数，并且可以控制启动 TDE。

- 偏移地址：0x300
- 复位值：0x0
- 复位方式：h/s

比特	名称	操作类型	描述
[31:28]	in_color_fmt	R/W	输入位图数据格式。 0000: RGB 4:4:4; 0101: 保留; 0001: ARGB 4:4:4:4; 0110: RGB 8:8:8; 0010: RGB 5:5:5; 0111: ARGB 8:8:8:8; 0011: RGB 5:6:5; 1000~1111: 保留。 0100: ARGB 1:5:5:5;
[27:24]	out_color_fmt	R/W	输出位图数据格式。 0000: RGB 4:4:4; 0101: 保留; 0001: ARGB 4:4:4:4; 0110: RGB 8:8:8; 0010: RGB 5:5:5; 0111: ARGB 8:8:8:8; 0011: RGB 5:6:5; 1000~1111: 保留。 0100: ARGB 1:5:5:5;
[23:16]	alpha	R/W	全局 alpha 值。 • 当寄存器 bit[9:8]为 0, 该 alpha 值参加 alpha 叠加运算 • 当寄存器 bit[7]为 0, 该 alpha 值是输出位图带的 alpha 值
[15:13]	reserved	R	保留。
[12:7]	tde_cmd	R/W	2D 加速命令字寄存器, 请参见表 10-2。
[6]	reserved	R	保留。
[5:2]	ROP2_code	R/W	ROP2 操作代码, 请参见表 10-3。
[1]	reserved	R	保留。
[0]	tde_start	R/W	2D 加速引擎启动位。 CPU 写 1 表示启动 TDE 执行 2D 命令。



表10-2 CMD 寄存器详细描述

位	名称	描述
[12:11]	cmd	TDE 操作命令。 00: 无命令操作; 01: blit; 10: pattern fill with clip; 11: solid draw。
[10]	color space	色度 keying 使能。 0: 不实现 color space 操作; 1: 对源位图实现 color space 操作。
[9:8]	data_conv	数据运算。 00: 不进行操作; 01: 进行 ROP2 操作; 10: 进行 alpha 操作, alpha 值来自源位图; 11: 进行 alpha 操作, alpha 值来自模块内部 alpha 寄存器。
[7]	alpha_from	输出数据的 alpha 来源。 输出数据的 alpha 值有 2 个来源: 0: 在输出数据类型含有 alpha 值, 该值来自源位图; 1: 位图中的 alpha 值来自与模块内部的 alpha 寄存器。

表10-3 ROP2 操作符详细描述

ROP2 Code	在 blit 中的含义	在 pattern fill 含义
0000	0 (Blackness)	0 (Blackness)
0001	$\sim(S + D)$	$\sim(P + D)$
0010	$\sim S \& D$	$\sim P \& D$
0011	$\sim S$	$\sim P$
0100	$S \& \sim D$	$P \& \sim D$
0101	$\sim D$	$\sim D$
0110	$S \wedge D$	$P \wedge D$
0111	$\sim(S \& D)$	$\sim(P \& D)$
1000	$S \& D$	$P \& D$
1001	$\sim(S \wedge D)$	$\sim(P \wedge D)$
1010	D	D



ROP2 Code	在 blit 中的含义	在 pattern fill 含义
1011	$\sim S + D$	$\sim P + D$
1100	S	P
1101	$S + \sim D$	$P + \sim D$
1110	$S + D$	$P + D$
1111	1 (Whiteness)	1 (Whiteness)

注:

S: 表示 source; D: 表示 destination; P: 表示 pattern。

\sim : 表示按位取反; \wedge : 表示按位异或; $\&$: 表示按位与, $+$: 表示按位或。

10.5.2 位图大小寄存器 (TDE_SIZE)

定义了 blit 或者 solid draw 的位图大小。

- 偏移地址: 0x304
- 复位值: 0x0
- 复位方式: h/s

比特	名称	操作类型	描述
[31:24]	reserved	R	保留。
[23:12]	blt_height	R/W	blit (或者 solid draw) 位图行数寄存器, blt_height 值等于实际位图高度减 1。 例如实际高度为 480, 则 blt_height 为 479。
[11:0]	blt_width	R/W	blit (或者 solid draw) 位图像素宽度寄存器, blt_width 值等于实际位图宽度减 1。 例如实际宽度为 640, 则 blt_width 为 639。

注:

对 pattern、位图大小寄存器 TDE_SIZE (偏移地址 0x804) 使用时:

在 blit 操作中, blt_size 为源位图的大小, 如果需要与目标位图操作, 目标位图与源位图大小一致。

在 pattern Fill 操作时, blt_size 为 pattern 操作的目的区域。超出 blt_size 的 pattern 部分被裁减掉。

10.5.3 源位图起始地址 (TDE_SOURCE_ADDR)

源位图数据的首地址, 定义了源位图第一行的起始地址。

- 偏移地址: 0x308
- 操作类型: R/W
- 复位值: 0x0



- 复位方式：h/s

比特	名称	描述
[31:0]	source_addr	源位图数据首地址。

注：该地址按照 half word(32 bit word)对齐。

10.5.4 目标位图起始地址 (TDE_DET_ADDR)

目标位图数据的首地址，定义了目标位图第一行的起始地址。

- 偏移地址：0x30C
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:0]	det_addr	目标位图数据首地址。

注：该地址按照 half word(32 bit word)对齐。

10.5.5 输出位图起始地址 (TDE_GOUT_ADDR)

输出位图数据的首地址，定义了输出位图第一行的起始地址。

- 偏移地址：0x310
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:0]	gout_addr	输出位图数据首地址。

注：该地址按照 half word(32 bit word)对齐。

10.5.6 源位图行间距 (TDE_SOURCE_STRIDE)

行间距描述了本行起始地址与下一行起始地址之间的距离。

- 偏移地址：0x314
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s



比特	名称	描述
[31:16]	source_stride	源位图行间距离。
[15:0]	det_stride	目标位图行间距离。

注：该地址按照 half word(32 bit word)对齐。

10.5.7 输出位图行间距 (TDE_GOUT_STRIDE)

- 偏移地址：0x318
- 复位值：0x0
- 复位方式：h/s

比特	名称	操作类型	描述
[31]	fifo_wrprotect	R/W	控制命令写入，当命令 FIFO 空间不够时是否允许写入。不用测试。 0：不允许写入； 1：允许写入。
[30]	endian_sel	R/W	0：2D 处理数据是 little endian； 1：2D 处理数据是 big endian。
[29:26]	reserved	R	保留。
[25:21]	pattern_width	R/W	pattern 的宽度。pattern_width 值等于实际 pattern 宽度减 1。如实际宽度为 8，则 pattern_width=7。Pattern 的最大宽度为 32 个像素（2 像素/字）或者 16 个像素（1 像素/字）。
[20:16]	pattern_height	R/W	pattern 的高度。pattern_height 值等于实际 pattern 高度减 1。如实际高度为 8，则 pattern_height=7。
[15:0]	gout_stride	R/W	输出位图行间距离。

注：该地址按照 half word(32 bit word)对齐。

10.5.8 颜色填充值 (TDE_COLOR_FILLED)

在执行 solid draw 命令时，填充目标区域的颜色值。

- 偏移地址：0x31C
- 复位值：0x0
- 复位方式：h/s



比特	名称	操作类型	描述
[31:24]	reserved	R	保留。
[23:16]	color_filled_r	R/W	颜色填充值 R 分量。
[15:8]	color_filled_g	R/W	颜色填充值 G 分量。
[7:0]	color_filled_b	R/W	颜色填充值 B 分量。

10.5.9 color space 最大值 (TDE_COLOR_MAX)

当 TDE_CTRL[color space]使能时，像素值在[TDE_COLOR_MIN, TDE_COLOR_MAX]区间的像素被视为关键色。请参见“10.2.3 color space”。

- 偏移地址：0x320
- 复位值：0x0
- 复位方式：h/s

比特	名称	操作类型	描述
[31:24]	reserved	R	保留。
[23:16]	color_max_r	R/W	color space 最大颜色值的红色值。
[15:8]	color_max_g	R/W	color space 最大颜色值的绿色值。
[7:0]	color_max_b	R/W	color space 最大颜色值的蓝色值。

10.5.10 color space 最小值 (TDE_COLOR_MIN)

当 TDE_CTRL[color space]使能时，像素值在[TDE_COLOR_MIN, TDE_COLOR_MAX]区间的像素被视为关键色。请参见“color space”。

- 偏移地址：0x324
- 复位值：0x0
- 复位方式：h/s

比特	名称	操作类型	描述
[31:24]	reserved	R	保留。
[23:16]	color_min_r	R/W	color space 最小颜色值的红色值。
[15:8]	color_min_g	R/W	color space 最小颜色值的绿色值。
[7:0]	color_min_b	R/W	color space 最小颜色值的蓝色值。



10.5.11 TDE 状态寄存器 (TDE_STATUS)

- 偏移地址: 0x324
- 操作类型: R
- 复位值: 0x1E
- 复位方式: h/s

比特	名称	描述
[31:8]	reserved	保留。
[7]	history_fifo_verflow	命令 FIFO 溢出历史告警寄存器。 当软件写入命令个数超过 fifo_depth 的值, 命令 FIFO 有可能溢出, 置为 1。 CPU 读后清 0。
[6:4]	reserved	保留。
[3:0]	fifo_depth	表示 2D 的命令 FIFO 中剩余的空间。

注: 对命令 FIFO 的操作请参见“[10.3.3 配置](#)”。



11 视频输入单元 (VIU)

关于本章

本章描述内容如下表所示。

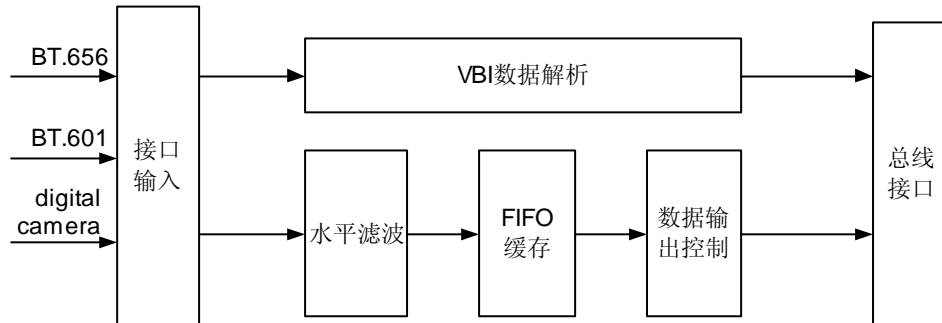
标题	内容
11.1 概述	概括介绍 VIU。
11.2 功能描述	概括介绍 VIU 的特点。
11.3 信号描述	描述 VIU 的外部输入输出管脚信号。
11.4 工作方式	介绍 VIU 的接口时序。
11.5 图像存储方式	介绍 VIU 的图像存储方式。
11.6 寄存器概览	概括介绍 VIU 的寄存器。
11.7 寄存器描述	详细描述 VIU 的寄存器。



11.1 概述

视频输入单元 VIU (Video Input Unit) 可以通过 ITU-R BT.656/601 接口或者 digital camera 接口, 接收视频数据, 存入指定的内存区域。在此过程中, VIU 可以对视频图像数据进行 1/2 的 Down Scaling 和色度重采样。VIU 的功能框图如图 11-1 所示。

图11-1 VIU 功能框图



11.2 功能描述

VIU 有以下特点:

- 以 ITU-R BT. 656/ITU-R BT. 601 标准采集 YUV 4:2:2 视频流
- 支持 Bayer RGB 和 5:5:5 / 5:6:5 / 8:8:8 / RGB 数据输入
- Bayer 输入的 Pattern 支持: GRBG、RGGB、BGGR、GBRG
- 在 ITU-R BT.601 模式下, 支持主模式和从模式
- 输入数据支持 UYVY、VYUV、YUYV、YVYU 数据输入顺序
- 支持直接数字 Camera 连接
- 支持 1/2 水平 Down Scaling
- 支持色度重新采样, 从 Cosited 格式到 Interspersed 格式的水平转换
- 支持 raw 数据采集
- 数据位宽支持 8-bit
- 支持在一个特定窗口 (Window) 内获取数据
- 输出格式支持多通道模式
- 输出格式支持 YCbCr 4:2:2 单通道模式
- 输出格式支持 Y (R)、Cb (G)、Cr (B) 的三通道模式
- 输出格式支持 raw data 存储模式
- 支持图像块填充和块屏蔽
- 支持 VBI 数据接收
- 支持图像亮度统计



11.3 信号描述

表 11-1 描述了 VIU 的外部输入输出管脚信号。

表11-1 视频输入接口 (VIU) 信号

信号名	方向	描述
VI_DAT[7:0]	I	8-bit 位宽的视频输入数据或者 Raw Data。
VICK	I	视频输入时钟。
VIHS	I/O	水平同步脉冲或者数据有效信号。 在主模式下，该管脚作为输出；在从模式下，该管脚作为输入。 配置寄存器的 hsync 位，控制该信号是水平同步脉冲或者数据有效电平。 hsync=0: 表示数据有效。 hsync=1: 表示同步脉冲。
VIVS	I/O	垂直同步脉冲或者场指示信号。 在主模式下，该管脚作为输出；在从模式下，该管脚作为输入。 配置寄存器的 vsync 位，控制该信号是垂直同步脉冲或者场指示信号。 vsync=0: 表示场指示信号。 vsync=1: 表示同步脉冲。

11.4 工作方式

视频输入接口主要支持：

- [BT.656/601 YUV4:2:2](#)
- [数字 camera 接口](#)
- [Raw data 接口](#)
- [Bayer RGB 输入时序](#)

11.4.1 BT.656/601 YUV4:2:2

像素时序

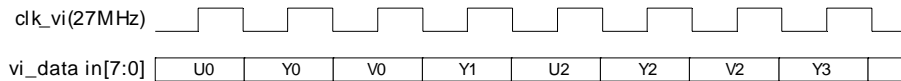
在 ITU-R BT.656/601 YUV4:2:2 建议中：

- 亮度信号与色度信号的采样比例为 2: 1，2 个亮度信号共用 1 个色度 UV 信号。



- 推荐每行亮度采样 720 个有效图像像素点，色度采样 360 个有效图像像素点，色度在对应偶数像素点采样（起始采样点为 0 像素点），以 co-sited 格式采样。
- 亮度色度信号在同一个 8bit 通道内传输的情况下，传输顺序为： $C_B0 Y_0 C_R0 Y_1 C_B2 Y_3 C_R2 Y_3 \dots C_B718 Y718 C_R718 Y719$ 。时钟频率为 27MHz。图 11-2 为像素输入时序关系图（其中 U 代表 C_B ，V 代表 C_R ）。

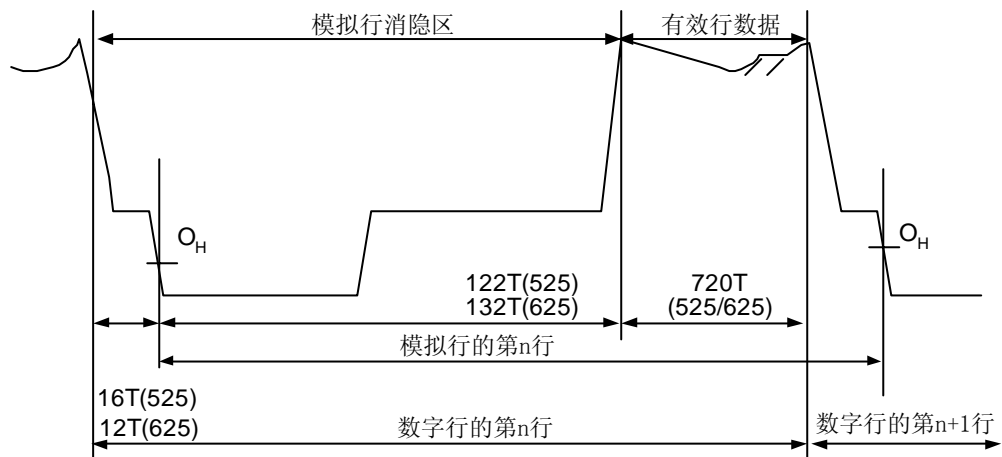
图11-2 像素输入时序



行时序

ITU-R BT.656/601 建议规定了 PAL (Phase-Alternation Line) 制式和 NTSC (National Television System Committee) 制式一行的整个采样数（包括行消隐区）。数字有效行的 720 个亮度取样与 625 行、525 行制式的模拟同步基准间的关系如图 11-3 所示（其中 T 表示像素周期， O_H 表示行同步前沿，半幅度基准）。

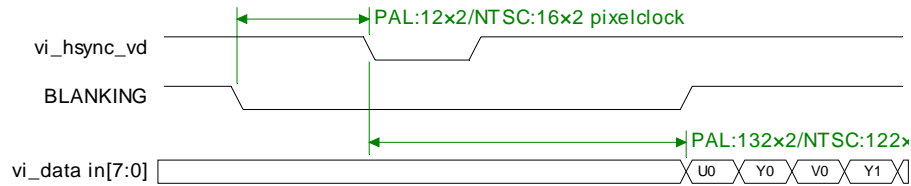
图11-3 模拟全电视信号对于的数字行采样时序



ITU-R BT.656/601 采用不同的方法标志行同步信号。

- ITU-R BT.601 建议
行同步由行同步信号 HSYNC 产生，HSYNC 的下降沿（上升沿、下降沿可编辑）表示新的一行开始，HSYNC 同步脉冲在图 11-3 的 O_H 处产生。行同步脉冲的宽度可编辑。ITU-R BT.601 的行时序如图 11-4 所示。图中“BLANKING”表示行消隐，VI 中无此管脚，仅用此信号来表示有效数据。

图11-4 ITU-R BT. 601 行时序图



• ITU-R BT.656 建议

同步信号集成在数据流中，数据流中的特殊 byte SAV 和 EAV 分别表示有效行数据的开始和结束。在视频数据流中，由 FF 00 00 (FF、00 为图像编码数据的保留字节，为非图像数据) 构成的定时基准码字的码头来标志紧接着的一个字节为 SAV 或者 EAV，ITU-R BT.656 的行数据流格式如表 11-2 所示。

表11-2 ITU-R BT.656 YUV 4:2:2 行数据格式

行消隐期		定时基准码				720 有效像素 YUV4:2:2						定时基准码				行消隐期	
...	10	FF	00	00	SAV	CB0	Y0	CR0	Y1	...	Y719	FF	00	00	EAV	80	...

SAV 和 EAV 依据 SAV/EAV 字节内的特殊 bit 位“H”区分。SAV/EAV 还包含了场同步比特位“V”和场号“F”。SAV/EAV 的具体描述如表 11-3 所示。

表11-3 SAV/EAV Format

Bit7	Bit6 (F)	Bit5 (V)	Bit4 (H)	Bit[3:0]
1 (固定值)	场指示位 1st field: F=0 2nd field: F=1	垂直消隐位 VBI: V=1 Active video: V=0	H=0 in SAV H=1 in EAV	保留

在 ITU-R BT.656 建议中，采用了 8 个有效保留位来定义有效的 SAV 和 EAV，4 个有效保留位起纠错的作用（发生 1bit 错误，可以纠错；发生 2bit 错误，可以报错）。有效的 SAV/EAV 值如表 11-4 所示。

表11-4 有效 SAV/EAV 值

编码	二进制值	场号	垂直消隐期
SAV	1000 0000	1	-
EAV	1001 1101	1	-
SAV	1010 1011	1	是
EAV	1011 0110	1	是



编码	二进制值	场号	垂直消隐期
SAV	1100 0111	2	-
EAV	1101 1010	2	-
SAV	1110 1100	2	是
EAV	1111 0001	2	是

帧时序

在 ITU-R BT.601 和 ITU-R BT.656 建议中，规定 PAL 和 NTSC 制式 TV 图像帧为隔行扫描。

表11-5 PAL 和 NTSC 制式 TV 图像帧对比

制式	行/帧	帧/秒
PAL 制	656 行	每秒 25 帧，1 帧分奇偶场
NTSC 制	525 行	每秒 30 帧，1 帧分奇偶场

对于 ITU-R BT.601 建议，把 SYNC、FIELD 信号作为垂直同步信号、场同步信号。

对于 ITU-R BT.656 建议，垂直同步信号包含在 SAV 和 EAV 内。

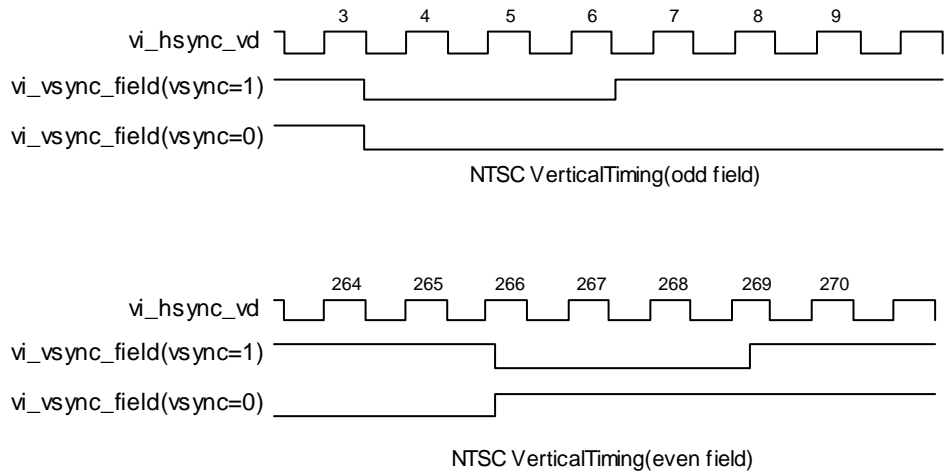
- ITU-R BT.601 垂直时序

ITU-R BT.601 建议：信号 VSYNC/FIELD 作为垂直同步信号。VSYNC 的脉冲或者 FIELD 的跳变标志奇偶场的开始。

图 11-5 和图 11-6 分别为 VI (Video Input) 在 NTSC 制式和 PAL 制式的垂直时序关系图。其中 vi_hsync_vd 为水平同步脉冲，vi_vsync_field 在 vsync=1 时为垂直同步脉冲；在 vsync=0 时为场同步信号。

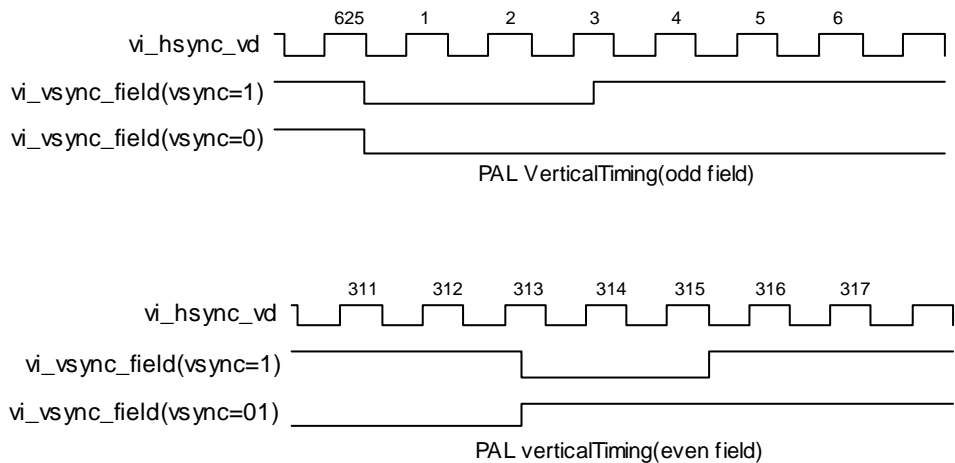


图11-5 NTSC 制式垂直同步时序图



在 NTSC 隔行扫描制式下，第 1 场的垂直同步信号在第 4 行的起始位置变为低电平，持续 3 行低电平后，在第 7 行的起始位置变为高电平。VIU 准备从第 22 行开始接收 240 行到第 261 行数据。第 2 场的垂直同步信号在第 266 行的中间位置变为低电平，持续 3 行低电平后，在第 269 行中间位置变为高电平。VIU 准备从第 285 行开始接收 240 行到第 525 行数据。

图11-6 PAL 制式垂直同步时序图



在 PAL 隔行扫描制式下，第 1 场的垂直同步信号在第 1 行起始位置变为低电平，持续 2.5 行低电平后，在第 3 行的中间位置变为高电平。VIU 准备从第 24 行开始接收 287 行到第 310 行数据。第 2 场的垂直同步信号在第 313 行的中间位置变为低电平，持续 2.5 行低电平后，在第 316 行的起始位置变为高电平。VIU 准备从第 336 行开始接收 287 行到第 622 行数据。

在所有行中，有效行数据如表 11-6 所述。



表11-6 ITU-R BT.601 一帧有效行数据

制式	场	有效行
NTSC	奇数场	22~261
	偶数场	285~524
PAL	奇数场	23~310
	偶数场	336~623

- ITU-R BT.656 垂直时序

ITU-R BT.656 隔行扫描帧时序定义如表 11-7 所示。F=1 表示第二场（偶数场），F=0 表示第一场（奇数场）。V=1 表示垂直消隐区，V=0 表示有效行数据区。

表11-7 ITU-R BT.656 帧时序

行号		F 位	V 位	描述
525/60	625/50			
1~3	624~625	1	1	第二场垂直消隐。
4~19	1~22	0	1	第一场垂直消隐，SAV/EAV 改变为显示第一场。
20~263	23~310	0	0	第一场有效数据行。
264~265	311~312	0	1	第一场垂直消隐。
266~282	313~335	1	1	第二场垂直消隐，SAV/EAV 改变为显示第二场。
283~525	336~623	1	0	第二场有效数据行。

用户自定义的同步时序

软件必须配置寄存器 `field_sync`、`line_sync` 和 `vi_frame`，VIU 才能工作。在软件不干预的情况下，系统默认为 PAL 制式隔行扫描。软件配置的水平、垂直时序关系如图 11-7 和图 11-8 所示。



图11-7 软件配置的水平时序图

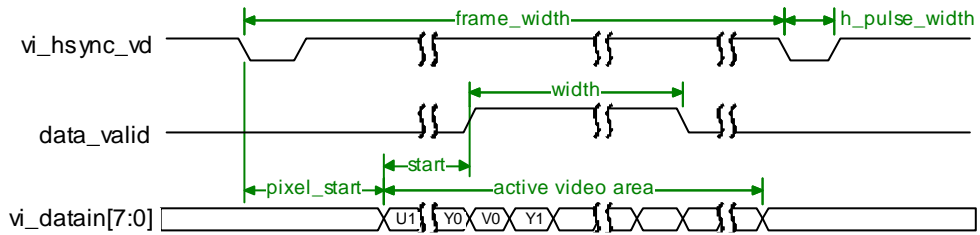
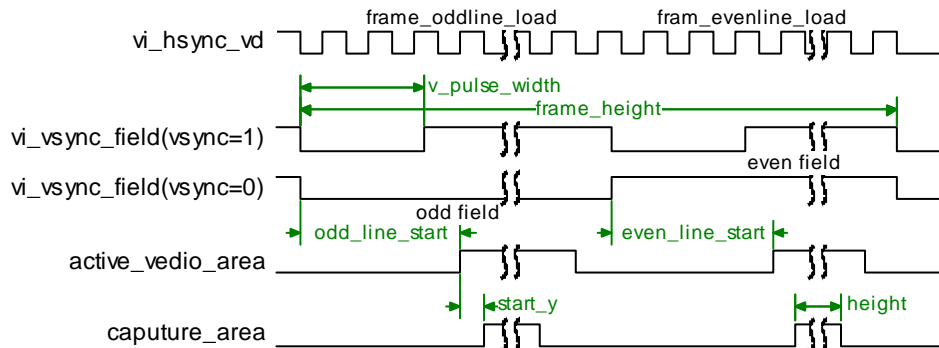


图11-8 软件配置的垂直时序图



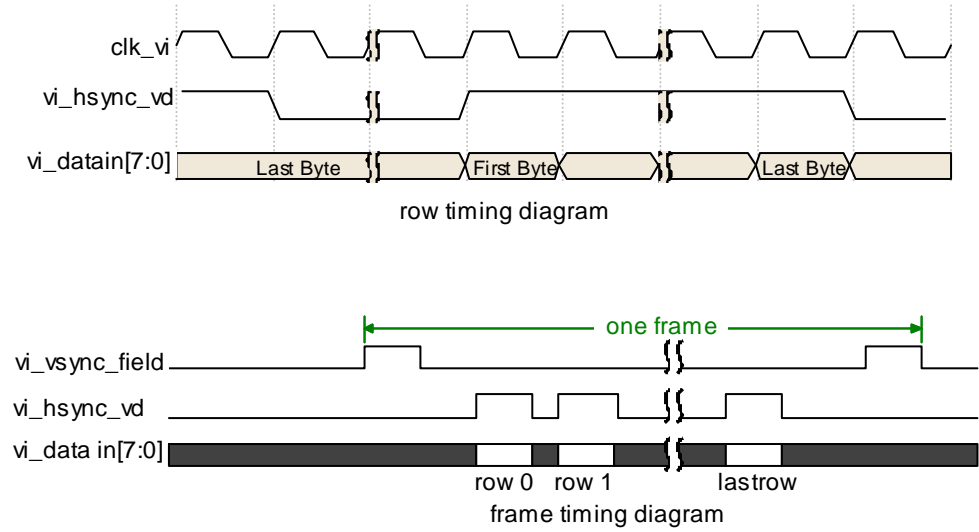
在软件配置的垂直时序图中， active_video_area 和 capture_area 分别表示活动图像区域和实际图像获取区域。

11.4.2 数字 camera 接口

VIU 支持对 VGA, QVGA 等摄像头的直接数据传输。在数字 camera 支持的水平和垂直时序如图 11-9 所示。



图11-9 数字 camera 支持的水平、垂直时序图



11.4.3 Raw data 接口

Raw data 传输方式下，输入数据按照输入时钟频率输入。VIU 直接在数据输入管脚 vi_datain 接收数字数据。VIU 工作在从模式，且只有在 vi_hsync_vd 有效的情况下才会采集 vi_datain 上的数据。

VIU 对这种 raw data 不作任何处理，直接存储在 SDRAM 中。每接收完指定大小的 raw data，VIU 将产生完成中断 vi_cc_int 通知 CPU（如果中断使能）。

11.4.4 Bayer RGB 输入时序

VIU 支持对 camera bayer RGB 数据接收。bayer 支持的垂直和水平时序如图 11-10、图 11-11 和图 11-12 所示。

图11-10 bayer 数据垂直时序

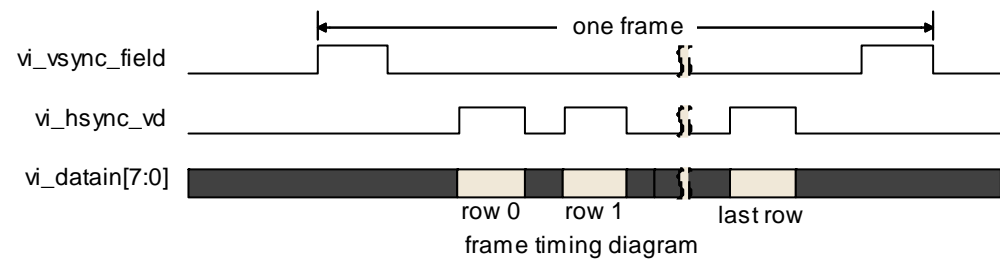




图11-11 bayer 数据奇数行水平时序

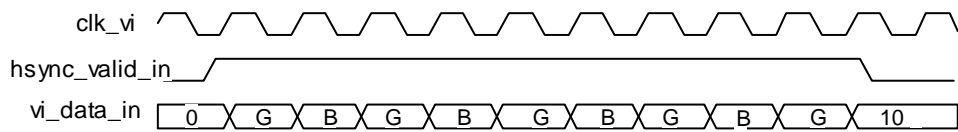
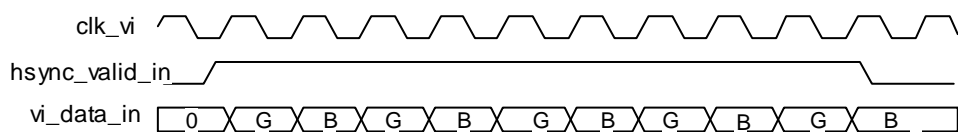


图11-12 bayer 数据偶数行水平时序



11.5 图像存储方式

图像存储方式主要包括：

- YC planar 存储
- Y/ CB/CR (R/G/B) planar 存储
- Packets 存储
- Raw data 存储

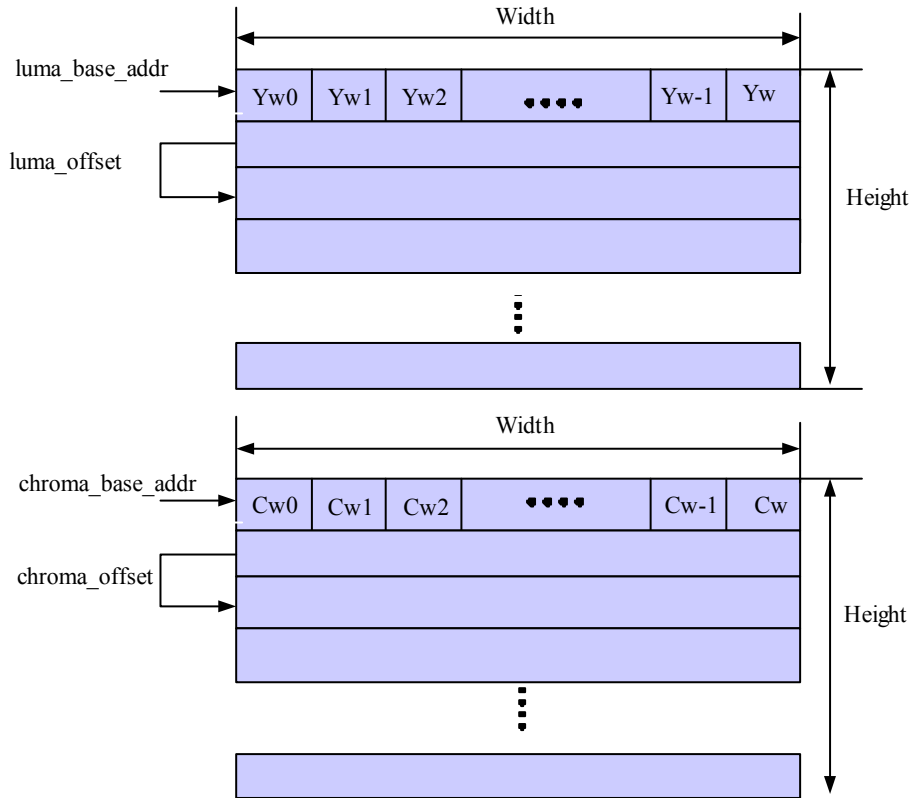
11.5.1 YC planar 存储

系统设定了视图区域后，对读入数据按照 planar 方式存储，即亮度分量和色度分量分别存储在 SDRAM 中的亮度存储空间和色度存储空间。

- 在 1 行内，亮度、色度分量各自连续存储；
- 连续 2 行之间的存储，可以通过系统定义的行首与行首之间的存储间隔参数 offset 定义。亮度和色度分量在 SDRAM 中的存储位置由起始地址 base_addr 来指示。全分辨率采集的 YUV4:2:2 存储结构如图 11-13 所示。

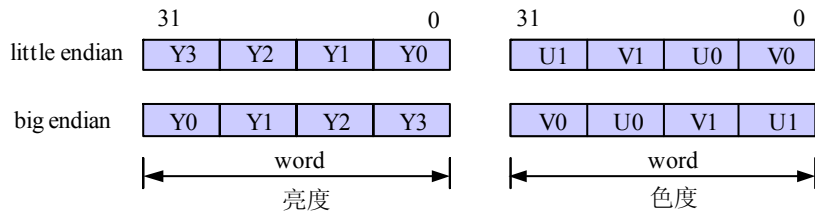


图11-13 YUV4:2:2 的存储方式



在 SDRAM，数据的存储是以 word（32bit）为单位。由 4 个 8bit 像素组成一个 32bit 的 word，在 4 个字节构成一个 word 时有 2 种方式：big endian 和 little endian。图 11-14 是以亮度和色度分量为例来说明 big endian 和 little endian 的存储方式。

图11-14 big endian 和 little endian 图像存储方式



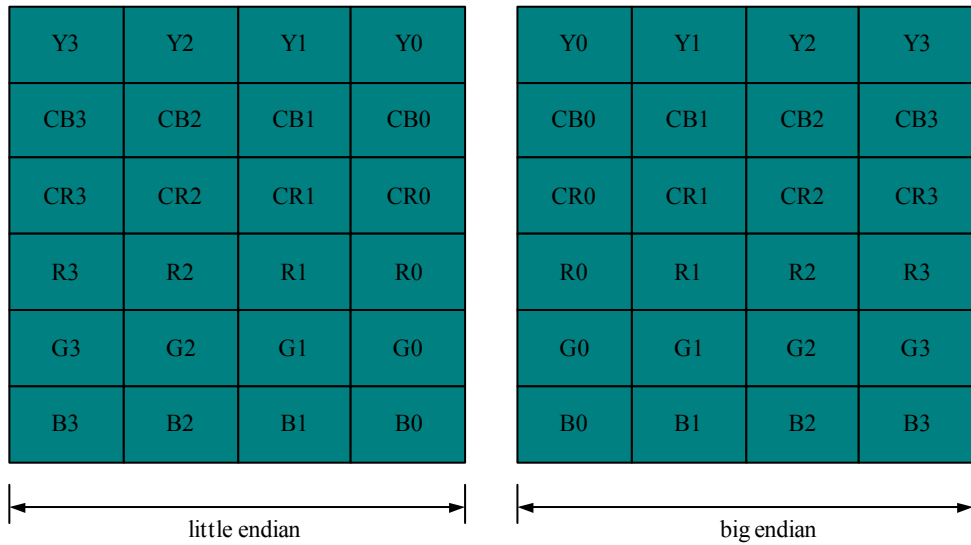
VIU 默认 SDRAM 采用 little endian 方式存储数据。VIU 有一个 little_endian bit 位来选择 little endian 和 big endian 的存储方式。

11.5.2 Y/ CB/CR (R/G/B) planar 存储

Ycbcr(rgb) planar 存储方式将图象的 Y/CB/CR 或者 R/G/B 三个分量分别存储。



图11-15 Y/CB/CR 或 R/G/B 图像存储 big endian&little endian 方式



11.5.3 Packets 存储

Packets 方式存储是将 4:2:2 Y/CB/CR 数据按图 11-16 的方式存储。

图11-16 图像存储 packets 方式



11.5.4 Raw data 存储

Raw data 方式存储是将数据按数据顺序依次存放到一个 word 中。由于在 memory 中，1 个 word 由 4 个 8bit 组成，当数据为 8bit、9bit 或 10bit 时，其存储方式如图 11-17 和图 11-18 所示。

图11-17 raw data 8bit 存储方式



图11-18 raw data 9/10bit 存储方式





11.6 寄存器概览

VIU 寄存器地址位宽 32 位，地址范围：0x9000_0000~0x9000_FFFF。

表11-8 VIU 寄存器概览（基址是 0x9000_0000）

偏移地址	名称	描述	页码
0x00	VI_CFG	VIU 配置寄存器 ^{bc}	11-15
0x04	VI_VSYNC1	VIU 奇场时序寄存器 ^b	11-18
0x08	VI_VSYNC2	VIU 偶场时序寄存器 ^b	11-18
0x0C	VI_HSYNC	VIU 行时序寄存器 ^b	11-18
0x14	VI_CAP_START	VIU 图像获取起始位置寄存器 ^{ab}	11-19
0x18	VI_CAP_SIZE	VIU 图像获取大小寄存器 ^{abc}	11-19
0x1C	VI_LINE_OFFSET	图像存储行间距寄存器 ^{ab}	11-21
0x20	VI_ABASE_ADDR	A 通道基地址寄存器 ^{abc}	11-22
0x24	VI_BBASE_ADDR	B 通道基地址寄存器 ^{ab}	11-22
0x28	VI_CBASE_ADDR	C 通道基地址寄存器 ^{ab}	11-22
0x2C	VI_CTRL	VIU 控制寄存器 ^{bc}	11-23
0x30	VI_INT_MASK	VIU 中断 mask 寄存器 ^{bc}	11-25
0x34	VI_INT_STATUS	VIU 中断状态寄存器 ^{bc}	11-26
0x38	VI_STATUS	VIU 状态寄存器 ^{bc}	11-26
0x40	VI_Y_STORESIZE	Y 分量数据存储大小寄存器 (Bayer 时为 G 通道) ^{abc}	11-27
0x44	VI_U_STORESIZE	U 分量数据存储大小寄存器 (Bayer 时为 R 通道) ^{ab}	11-27
0x48	VI_V_STORESIZE	V 分量数据存储大小寄存器 (Bayer 时为 B 通道) ^{ab}	11-28
0x70	VBI_B1_START	VBI 数据获取位置寄存器 1 ^a	11-29
0x74	VBI_B1_SIZE	VBI 数据大小寄存器 1 ^a	11-29
0x78	VBI_B2_START	VBI 数据获取位置寄存器 2 ^a	11-30
0x7C	VBI_B2_SIZE	VBI 数据大小寄存器 2 ^a	11-30
0x80	VBI_B1_WORD1	VBI 数据 1 第 1 个 word	11-31
0x84	VBI_B1_WORD2	VBI 数据 1 第 2 个 word	11-31



偏移地址	名称	描述	页码
0x88	VBI_B1_WORD3	VBI 数据 1 第 3 个 word	11-31
0x8C	VBI_B1_WORD4	VBI 数据 1 第 4 个 word	11-31
0x90	VBI_B1_WORD5	VBI 数据 1 第 5 个 word	11-31
0x94	VBI_B1_WORD6	VBI 数据 1 第 6 个 word	11-31
0x98	VBI_B1_WORD7	VBI 数据 1 第 7 个 word	11-31
0x9C	VBI_B1_WORD8	VBI 数据 1 第 8 个 word	11-31
0xA0	VBI_B2_WORD1	VBI 数据 2 第 1 个 word	11-31
0xA4	VBI_B2_WORD2	VBI 数据 2 第 2 个 word	11-31
0xA8	VBI_B2_WORD3	VBI 数据 2 第 3 个 word	11-31
0xAC	VBI_B2_WORD4	VBI 数据 2 第 4 个 word	11-31
0xB0	VBI_B2_WORD5	VBI 数据 2 第 5 个 word	11-31
0xB4	VBI_B2_WORD6	VBI 数据 2 第 6 个 word	11-31
0xB8	VBI_B2_WORD7	VBI 数据 2 第 7 个 word	11-31
0xBC	VBI_B2_WORD8	VBI 数据 2 第 8 个 word	11-31
0xC0	BLOCK_COVER_START	图像块屏蔽起始地址寄存器 ^a	11-32
0xC4	BLOCK_COVER_SIZE	图像块屏蔽大小寄存器 ^a	11-32
0xC8	BLOCK_COVER_COLOR	图像块屏蔽颜色寄存器 ^a	11-32
0xCC	LUM_ADDER	亮度统计寄存器	11-33

注:

- a: 非及时性寄存器，只有在新的一帧或一场的起始时刻才会被载入到工作寄存器；
- b: 在直接数字 camera 模式下有效的寄存器；
- c: 在 raw data 模式下有效的寄存器。

11.7 寄存器描述

本节详细描述了 VIU 寄存器。

11.7.1 配置寄存器

配置寄存器反映了硬件当前的工作状态，为读写寄存器。

- 偏移地址: 0x00



- 操作类型: R/W
- 复位值: 0xC0065
- 复位方式: h/s

比特	名称	描述
[31:20]	Reserved	保留。
[19]	down_scaling	1/2 缩放。 0: 不进行 1/2 缩放; 1: 图像 1/2 采集, 水平 1/2 缩放。
[18]	chroma_resample	色度重采样。 0: 不进行色度重新采样; 1: 色度重新采样, 从 Co-sited 到 Interspersed 转换。
[17:16]	store_method	存储方式 (Store Method)。 00: 三通道 YCbCr 或者三通道 RGB; 01: 双通道 Y 和 C (CbCr) (在 Bayer 模式下不支持); 10: 单通道 YCbCr 4:2:2 (在 Bayer 模式下不支持); 11: raw data 方式存储 (RGB 555、RGB 565、RGB 888、raw data 模式下只能用这种模式)。
[15:14]	cap_sel	图像数据获取选择 (Capture Select)。 00: 仅对奇数场采集; 01: 仅对偶数场都采集; 10: 奇数偶数场都采集; 11: 保留。 默认值为 10。
[13:12]	cap_seq	YUV 输入顺序寄存器, 默认为值 00。 00: UYVY (Bayer 模式下第一行 GRGR, 第二行 BGBG); 01: VYUY (Bayer 模式下第一行 RGRG, 第二行 GBGB); 10: YUYV (Bayer 模式下第一行 BGBG, 第二行 GRGR); 11: YVYU (Bayer 模式下第一行 GBGB, 第二行 RGRG)。
[11]	little_endian	Little endian 存储方式。 0: big endian 方式存储; 1: little endian 方式存储。 默认为 1。



比特	名称	描述
[10]	dl	数据类型选择 (Data Select)。 0: YCbCr 或者 RGB 565、RGB 555、RGB 888 数据类型; 1: Bayer RGB 数据类型; 在 Digital Camera 模式下有效。
[9:8]	Data_width	数据位宽 (Data Width)。 00: 8-bit; 01: 9-bit; 10: 10-bit; 11: 保留。 默认为 00。
[7:6]	cap_mode	数据接收模式。 00: ITU-R BT.656 模式; 01: ITU-R BT.601 模式; 10: 数字 Camera 模式; 11: raw data 模式。 默认值为 00。
[5]	Reserved	保留。
[4]	vsync	管脚 VIVS 配置信号。 0: 场号 (奇场或者偶场) 或者行有效信号; 在 ITU-R BT.601 模式下表示场号, 在 Camera 或者 Bayer 接口情况下表示行有效信号。 1: 垂直同步脉冲; 默认值为 1。
[3]	vsync_neg	管脚 VIVS 极性配置。 0: 正电平有效。 • 在脉冲情况下 (vsync=1), 正脉冲表示同步脉冲 • 在场号模式下, 正电平表示偶数场, 负电平表示奇数场 • 在行有效情况下, 正电平表示行有效 1: 负电平有效。 • 在脉冲情况下 (vsync=1), 负脉冲表示同步脉冲 • 在场号模式下, 负电平表示偶数场, 正电平表示奇数场 • 在行有效信号情况下, 负电平表示行有效



比特	名称	描述
[2]	hsync	管脚 VIHS 配置信号。 0: VIHS 表示数据有效; 1: VIHS 表示水平同步脉冲。 默认值为 1。
[1]	hsync_neg	管脚 VIHS 极性配置。 0: 正电平有效。 • 在脉冲情况下 (hsync=1), 正脉冲表示同步脉冲 • 在数据有效情况下 (hsync=0), 正电平表示数据有效 1: 负电平有效。 • 在脉冲情况下 (hsync=1), 负脉冲表示同步脉冲 • 在数据有效情况下 (hsync=0), 负电平表示数据有效
[0]	master_mode	主模式选择。 0: 从模式; 1: 主模式。

11.7.2 时序寄存器

奇场时序寄存器 (VI_VSYNC1) 配置奇场的工作时序, 为读写寄存器。

- 偏移地址: 0x04
- 操作类型: R/W
- 复位值: 0x1500_511F
- 复位方式: h/s

偶场时序寄存器 (VI_VSYNC2) 配置偶场的工作时序, 为读写寄存器。

- 偏移地址: 0x08
- 操作类型: R/W
- 复位值: 0x1600_811F
- 复位方式: h/s

行时序寄存器 (VI_HSYNC) 配置行同步时序, 为读写寄存器。

- 偏移地址: 0x0C
- 操作类型: R/W
- 复位值: 0x41CC_057F
- 复位方式: h/s



寄存器名	比特	名称	描述
VI_VSYNC1	[31:24]	act1_voff	第 1 场开始到活动图像行距离（其值为实际行数减 1），默认为 21。
	[23:16]	act1_vbb	第 1 场活动图像结束到第 2 场开始的行距（其值为实际行数减 1），默认为 0。
	[15:12]	vsyn_width	场同步信号宽度（仅对 ITU-R BT.601 有效），在 vsyn_width 的最低位为 0.5 小数宽度标志，vsyn_wdith[3:1]为整数，默认为 2.5 (0101)。
	[11:0]	act1_height	第 1 场活动图像的高度（行数，其值为实际行数减 1），默认值为 287。
VI_VSYNC2	[31:24]	act2_voff	第 2 场开始到活动图像行距离（其值为实际行数减 1），默认为 22。
	[23:16]	act2_vbb	第 2 场活动图像结束到第 1 场开始的行距（其值为实际行数减 1），默认为 0。
	[15:12]	hsyn_width_msb	水平同步信号宽度的高 4 位（仅限于 ITU-R BT.601），默认为 (1000)，水平同步脉冲宽度默认为 128。
	[11:0]	act2_height	第 2 场活动图像的高度（行数，其值为实际行数减 1），默认为 287。
VI_HSYNC	[31:22]	act_hoff	上一行结束到本行活动数据区域的距离，以时钟为单位，（ITU-R BT.656/601 为像素的两倍），默认为 (264 - 1)。
	[21:16]	act_hbb	活动区域结束到本行的结束距离，默认为 (12 - 1)。
	[15:12]	hsyn_width_lsb	水平同步信号宽度的低 4 位（仅限于 ITU-R BT.601），默认为 (0000)，水平同步脉冲宽度默认为 128。
	[11:0]	act_width	活动图像宽度，以时钟为单位，所以在 ITU-R BT.656/601 模式下为像素的两倍（其值为实际值减 1）。 例如，默认为 1440（720 个像素）其写入的值为 1439。

11.7.3 图像获取窗口寄存器

图像获取起始位置寄存器（VI_CAP_START）配置数据获取的起始位置，为读写寄存器。

- 偏移地址：0x14



- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

图像获取大小寄存器 (VI_CAP_SIZE) 配置数据获取的大小, 为读写寄存器。

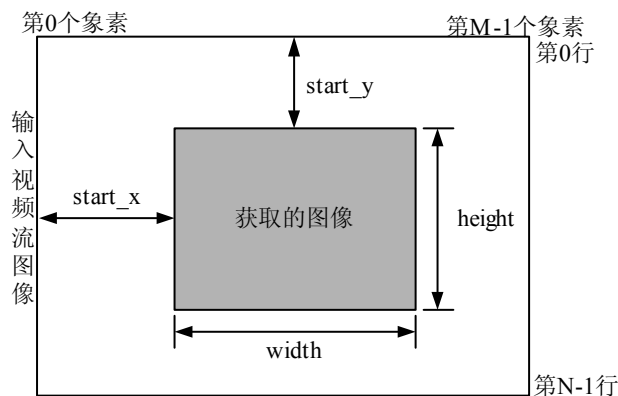
- 偏移地址: 0x18
- 操作类型: R/W
- 复位值: 0x16_077F
- 复位方式: h/s

VI_CAP_START 和 VI_CAP_SIZE 描述了从输入视频图像中获取矩形图像, 如图 11-19 所示。

- VI_CAP_START 寄存器描述了相对于输入图像数据流中获取图像起始坐标, start_y 和 start_x。
- VI_CAP_SIZE 描述了获取图像的大小, 宽度 (width) 和高度 (height)。

width 和 start_x 以输入像素为单位 (即亮度像素为单位); 在 YUV4:2:2 图像数据获取时, height 和 start_y 以行为单位。

图11-19 图像获取参数示意图



- start_y 和 start_x 在 VI_CAP_START 寄存器中分别占用 12 位宽度。
- width 和 height 在 VI_CAP_SIZE 寄存器中分别占用 12 位宽度。

寄存器名	比特	名称	描述
VI_CAP_START	[31:24]	Reserved	保留。
	[23:12]	start_y	开始获取图像的行号。
	[11:0]	start_x	开始获取图像的像素号。
VI_CAP_	[31:24]	Reserved	保留。



寄存器名	比特	名称	描述
SIZE	[23:12]	Height	获取图像的行数（高度）。 在 raw data 下，为获取数据的行数（高度）减 1。
	[11:0]	Width	获取图像一行的像素数（宽度）。 在 raw data 模式下，为实际需要接收的一行 raw data 的数据量（需为 64 的倍数）减 1（以 byte 为单位），所以在 raw data 模式下，一次接收的数据量是 $(height + 1) \times (width + 1)$ byte。

11.7.4 图像存储行间距寄存器

图像存储行间距寄存器 (VI_LINE_OFFSET) 配置获取数据存发的行偏移的大小，为读写寄存器。

- 偏移地址：0x1C
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:20]	aline_offset	以 word 为单位，16word 对齐。 <ul style="list-style-type: none"> • 在 raw data 模式下，仅该 stride 有效； • 在单通道 YCbCr 4:2:2 数据格式下，仅该 stride 有效； • 在双通道 Y/C 或者三通道 Y/Cb/Cr 或者 R/G/B 存储情况下，表示 Y 或者 G 的行 stride。
[19:10]	bline_offset	以 word 为单位，16word 对齐。 <ul style="list-style-type: none"> • 在 raw data 模式下，无效； • 在单通道 YCbCr 4:2:2 数据格式下，无效； • 在双通道 Y/C 模式下，表示 C 分量的行 stride； • 在三通道 Y/Cb/Cr 或者 R/G/B 存储模式下，表示 Cb 或者 R 分量行 stride。
[9:0]	cline_offset	以 word 为单位，16word 对齐。 <ul style="list-style-type: none"> • 在 raw data 模式下，无效； • 在单通道 YCbCr 4:2:2 数据格式下，无效； • 在双通道 Y/C 模式下，无效； • 在三通道 Y/Cb/Cr 或者 R/G/B 存储模式下，表示 Cr 或者 B 分量行 stride。



11.7.5 地址属性寄存器

a 通道基地址寄存器 (VI_ABASE_ADDR) 配置 Y 或者 G 分量的存放开始地址, 为读写寄存器。

- 偏移地址: 0x20
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

b 通道基地址寄存器 (VI_BBASE_ADDR) 配置 CB 或者 R 分量的存放开始地址, 为读写寄存器。

- 偏移地址: 0x24
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

c 通道基地址寄存器 (VI_CBASE_ADDR) 配置 CR 或者 B 分量的存放开始地址, 为读写寄存器。

- 偏移地址: 0x28
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

寄存器名	比特	描述
VI_ABASE_ADDR	[31:0]	最低 2 位为 0, 64byte 对齐。 <ul style="list-style-type: none"> • 在 raw data 模式下, 仅该地址有效; • 在单通道 YCbCr 4:2:2 数据格式下, 仅该地址有效; • 在双通道 Y/C 或者 Y/Cb/Cr 或者三通道 R/G/B 存储模式下, 表示 Y 或者 G 的首地址。
VI_BBASE_ADDR	[31:0]	最低 2 位为 0, 64byte 对齐。 <ul style="list-style-type: none"> • 在 raw data 模式下, 无效; • 在单通道 YCbCr 4:2:2 数据格式下, 无效; • 在双通道 Y/C 模式下, 表示 C 分量的首地址; • 在三通道 Y/Cb/Cr 或者 R/G/B 存储模式下, 表示 Cb 或者 R 分量首地址。



寄存器名	比特	描述
VI_CBASE_ADDR	[31:0]	<p>最低 2 位为 0，64byte 对齐。</p> <ul style="list-style-type: none"> 在 raw data 模式下，无效； 在单通道 YCbCr 4:2:2 数据格式下，无效； 在双通道 Y/C 模式下，无效； 在三通道 Y/Cb/Cr 或者 R/G/B 存储模式下，表示 Cr 或者 B 分量首地址。

11.7.6 控制寄存器

数据控制寄存器 (VI_CTRL) 控制数据获取的开始和结束，为读写寄存器。

控制寄存器实现：

- VIU 的使能
- reg_newer 功能
- 中断控制

VIU 的使能

VIU 只有在 vi_en 等于 1 的情况下才会工作。软件在使能 VIU 模块之前，必须先复位 VIU。

VIU 的 reg_newer 功能

- 在软件使能 VIU 之前，软件应该完成以下操作：
 - 完成对 VIU 的图像寄存器的写操作；
 - 写 reg_newer 位，通知 VIU 模块当前的寄存器已经准备就绪。
- 当使能 VIU 后，VIU 逻辑将开始工作。

当一场（帧）到来的时候，则：

 - 如果 reg_newer 为 0，则 VIU 将不会接收数据，置硬件状态为 **snooze**，等待下一场（帧）的数据的到来；
 - 如果 reg_newer 为 1，则开始接收数据，同时给出寄存器更新中断 (reg_updata_int)，并设置硬件状态为 **busy**。

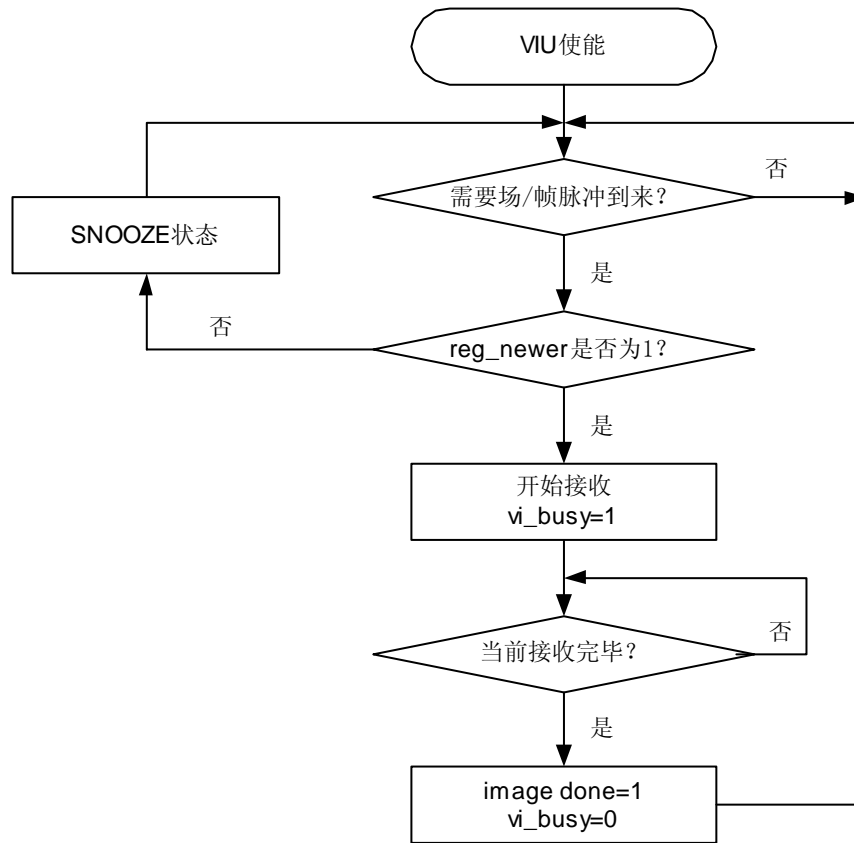
当接收完毕当前数据后，清除硬件 **busy** 状态。等到下一场（帧）到来的时候，则：

 - 如果 reg_newer 为 0，则放弃下一场（帧）数据的接收；
 - 如果 reg_newer 为 1，则可以紧接着前一次数据继续接收下一场（帧）的数据。

VIU 的工作流程如图 11-20 所示。



图11-20 VIU 的硬件工作流程



在 ITU-R BT.656/601 和直接数字 camera 模式下，每接收完一场/帧规定数据，在下一场的到来时将检测 `reg_newer` 位。如果 `reg_newer` 位为 1（表示软件已经更新或者确认 VIU 的寄存器），VIU 将自动 load 软件所配置的寄存器值到工作寄存器（工作寄存器软件不可访问），并开始接收下一场/帧数据，否则只有等到 `reg_newer` 为 1 且新的一场/帧到来时开始接收数据。

在 Raw data 模式下，在接收完毕规定的的数据，同时检测 `reg_newer` 位。如果 `reg_newer` 位为 1，则继续接收数据，否则将等待 `reg_newer` 为 1 时才接收数据。

- 偏移地址：0x2C
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:6]	Reserved	保留。
[5]	Blcok_st one_en	图像块填充使能。 0: 图像块填充不使能; 1: 图像块填充使能。



比特	名称	描述
[4]	Vbi_en	VBI 数据获取使能。 0: 不获取 VBI 数据; 1: 获取 VBI 数据。
[3]	int_pulse_select	中断方式选择信号。 0: 以电平方式中断; 1: 以脉冲方式中断。
[2]	fir_en	滤波器使能信号。 0: 滤波器不使能; 1: 滤波器使能。 注: 在 fir_en 使能的情况下, 配置寄存器的 down_scaling 和 chrom_resample 不能全为 0。
[1]	reg_newer	下一场 (帧) 寄存器准备完毕。 0: 下一需要接收场 (帧) 寄存器未准备好, 硬件将放弃下一场 (帧) 的接收。 1: 下一需要接收的场 (帧) 寄存器准备好, 在检测到下一场 (帧) 的场 (帧) 开始时, 硬件开始接收下一场数据。 VIU 硬件在自动更新内部工作寄存器后, 将自动清零该位。
[0]	vi_en	VIU 模块使能信号, 高电平有效。

11.7.7 中断 mask 寄存器

中断 mask 寄存器 (VI_INT_MASK) 配置中断的屏蔽与否, 为读写寄存器。

- 偏移地址: 0x30
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:7]	Reserved	保留。
[6]	frame_pulse_int_en	场起始中断使能。
[5]	reg_update_int_en	寄存器更新中断使能。
[4]	proc_err_int_en	ITU-R BT.656 情况下, 保护比特位错误中断使能。
[3]	buserr_int_en	总线错误中断使能。
[2]	frame_loss_int_en	场 (帧) 丢失中断使能。



比特	名称	描述
[1]	ovf_int_en	内部 FIFO 溢出错误中断使能。
[0]	cc_int_en	数据获取完毕中断使能。

11.7.8 中断状态寄存器

中断寄存器 (VI_INT_STATUS)，读写寄存器（只能写 1）。对中断状态寄存器的相应位写 1，即可清除该中断位。

- 偏移地址：0x34
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:7]	Reserved	保留，读为 0。
[6]	frame_pulse_int	场起始中断。
[5]	reg_update_int	工作寄存器更新中断状态位。
[4]	proc_err_int	输入数据出错中断 (ITU-R BT.656 模式)。
[3]	error_int	AHB 总线错误中断状态位。
[2]	frame_loss_int	场数据丢失中断状态位。
[1]	buf_ovf_int	内部缓冲 FIFO 溢出中断状态位。
[0]	cc_int	当前图像数据获取完毕中断状态位 (cc: capture completion)。

11.7.9 状态寄存器

状态寄存器 (VI_STATUS)，只读寄存器。

- 偏移地址：0x38
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7]	vi_busy	VIU 当前正处于工作状态。



比特	名称	描述
[6]	field2	当前接收为偶数场。
[5]	snooze	当前 VIU 处于睡眠状态。
[4]	proc_err	输入数据出错状态。
[3]	bus_err	总线错误状态。
[2]	frame_loss	VIU 丢失一场数据。
[1]	buf_ovf	VIU 内部 buffer 溢出。
[0]	image_done	VIU 接收完毕当前场数据（对于 raw data 传输为接收满规定数据）。

11.7.10 Y 分量数据存储大小寄存器

Y 分量数据存储大小寄存器 (VI_Y_STORESIZE)，读写寄存器。

- 偏移地址：0x40
- 操作类型：R/W
- 复位值：0xF_00A0
- 复位方式：h/s

比特	名称	描述
[31:24]	Reserved	保留。
[23:12]	y_height	Y 分量 (Bayer 时为 G 分量) 数据存储高度 (以行为单位)，在 raw data 接收模式时为实际获取数据块的大小 (height + 1)。
[11:0]	y_width	Y 分量 (Bayer 时为 G 分量) 数据存储宽度 (以 word 为单位，不够一个 word，以一个 word 算)、在 raw data 接收模式时为实际获取的数据长度比 4。

11.7.11 U 分量数据存储大小寄存器

U 分量数据存储大小寄存器 (VI_U_STORESIZE)，读写寄存器。

- 偏移地址：0x44
- 操作类型：R/W
- 复位值：0xF_00A0
- 复位方式：h/s

比特	名称	描述
[31:24]	Reserved	保留。



比特	名称	描述
[23:12]	c_height	U 分量 (Bayer 时为 R 分量) 数据存储高度 (以行为单位)。
[11:0]	c_width	U 分量 (Bayer 时为 R 分量) 数据存储宽度 (以 word 为单位, 不够一个 word, 以一个 word 算)。 在 packet 存储模式和在 raw data 接收模式下无效。

11.7.12 V 分量数据存储大小寄存器

V 分量数据存储大小寄存器 (VI_V_STORESIZE), 读写寄存器。

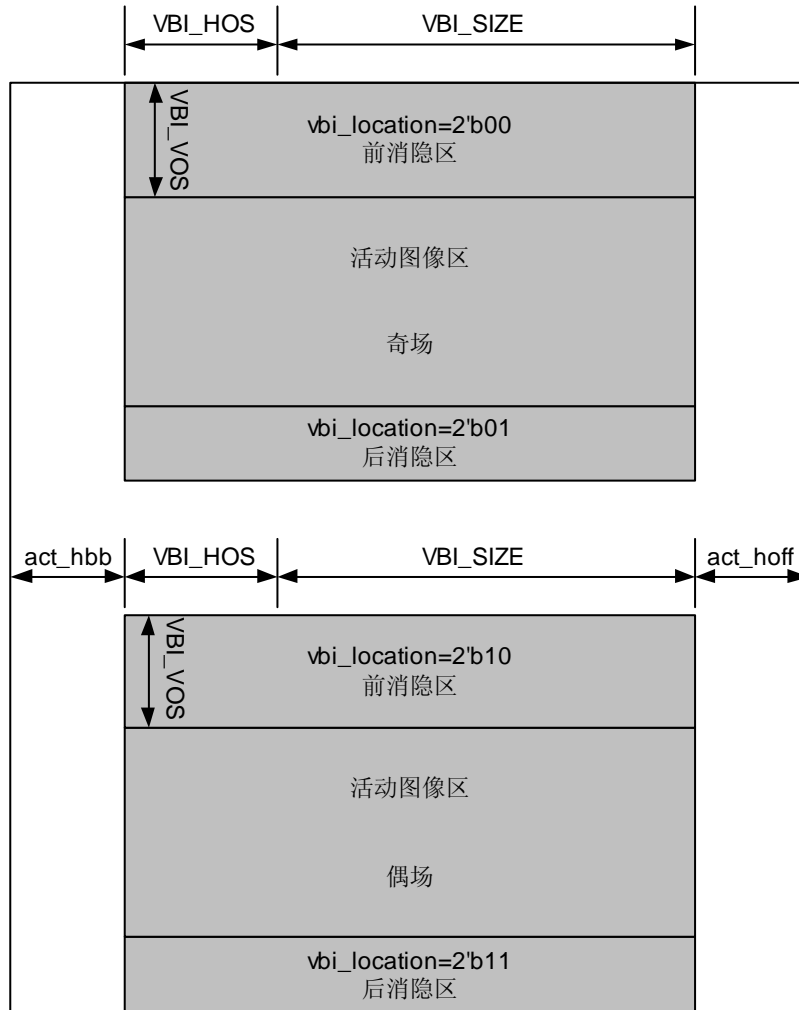
- 偏移地址: 0x48
- 操作类型: R/W
- 复位值: 0xF_00A0
- 复位方式: h/s

比特	名称	描述
[31:24]	Reserved	保留。
[23:12]	v_height	V 分量 (Bayer 时为 B 分量) 数据存储高度 (以行为单位)。
[11:0]	v_width	V 分量 (Bayer 时为 B 分量) 数据存储宽度 (以 word 为单位, 不够一个 word, 以一个 word 算)。 在 y/c 存储模式、packet 存储模式和 raw data 模式下无效。

11.7.13 VBI 数据位置寄存器

通常在消隐期间 (VBI) 传输 1 行到 2 行数据, 提供其他的数据增值服务, 所以在消隐期间需要对指定行的数据进行解析, 如图 11-21 所示。

图11-21 VBI 数据位置示意图



VBI 数据块 1

VBI 数据块 1 位置寄存器 (VBI_B1_START), 读写寄存器。

VIU 支持在一行内获取 16 个 word (64 byte) 或者在两行内获取 8 个 word (32 byte) 的 VBI 数据。

- 偏移地址: 0x70
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

VBI 数据块 1 大小 (VBI_B1_SIZE), 读写寄存器。

- 偏移地址: 0x74
- 操作类型: R/W
- 复位值: 0x20



- 复位方式: h/s

寄存器名	比特	名称	描述
VBI_B1_START	[31:26]	Reserved	保留。
	[25:24]	Vbi_location1	VBI 数据所在的消隐区。
	[23:12]	Vbi-vos1	消隐区开始到 VBI 数据行的距离 (以行为单位)。
	[11:0]	Vbi-hos1	行有效数据开始到 VBI 数据开始的水平偏移 (以接口时钟为单位)。
VBI_B1_SIZE	[31:12]	Reserved	保留。
	[11:0]	Vbi-size1	获取 VBI 数据的大小 (以字节为单位) 固定为 32。

VBI 数据块 2

VBI 数据块 2 位置寄存器 (VBI_B2_START), 读写寄存器。

- 偏移地址: 0x78
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

VBI 数据块 2 大小寄存器 (VBI_B2_SIZE), 读写寄存器。

- 偏移地址: 0x7C
- 操作类型: R/W
- 复位值: 0x20
- 复位方式: h/s

寄存器名	比特	名称	描述
VBI_B2_START	[31:26]	Reserved	保留。
	[25:24]	Vbi_location2	VBI 数据所在的消隐区。
	[23:12]	Vbi-vos2	消隐区开始到 VBI 数据行的距离 (以行为单位)。
	[11:0]	Vbi-hos2	行有效数据开始到 VBI 数据开始的水平偏移 (以接口时钟为单位)。
VBI_B2_SIZE	[31:24]	Reserved	保留。
	[23:12]	Reserved	保留。



寄存器名	比特	名称	描述
	[11:0]	Vbi-size2	获取 VBI 数据的大小（以字节为单位）固定为 32。

11.7.14 VBI 数据获取存储寄存器

获取的 16 个 word 的 VBI 数据存放在 16 个 32 位的寄存器组。第 1 组从 VBI_B1_WORD1 开始存放；第 2 组从 VBI_B2_WORD1 开始存放。

16 组 VBI 数据寄存器，只读寄存器。

- 偏移地址：0x80~0xBC
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

偏移地址	名称	描述
0x80	VBI_B1_WORD1	VBI 数据 1 组第 1 个 word。
0x84	VBI_B1_WORD2	VBI 数据 1 组第 2 个 word。
0x88	VBI_B1_WORD3	VBI 数据 1 组第 3 个 word。
0x8C	VBI_B1_WORD4	VBI 数据 1 组第 4 个 word。
0x90	VBI_B1_WORD5	VBI 数据 1 组第 5 个 word。
0x94	VBI_B1_WORD6	VBI 数据 1 组第 6 个 word。
0x98	VBI_B1_WORD7	VBI 数据 1 组第 7 个 word。
0x9C	VBI_B1_WORD8	VBI 数据 1 组第 8 个 word。
0xA0	VBI_B2_WORD1	VBI 数据 2 组第 1 个 word。
0xA4	VBI_B2_WORD2	VBI 数据 2 组第 2 个 word。
0xA8	VBI_B2_WORD3	VBI 数据 2 组第 3 个 word。
0xAC	VBI_B2_WORD4	VBI 数据 2 组第 4 个 word。
0xB0	VBI_B2_WORD5	VBI 数据 2 组第 5 个 word。
0xB4	VBI_B2_WORD6	VBI 数据 2 组第 6 个 word。
0xB8	VBI_B2_WORD7	VBI 数据 2 组第 7 个 word。
0xBC	VBI_B2_WORD8	VBI 数据 2 组第 8 个 word。



11.7.15 图像块屏蔽设置寄存器

图像块屏蔽起始位置寄存器 (BLOCK_COVER_START) 配置屏蔽的起始位置，为读写寄存器。

- 偏移地址：0xC0
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

图像块屏蔽大小寄存器 (BLOCK_COVER_SIZE) 配置屏蔽数据块的大小，为读写寄存器。

- 偏移地址：0xC4
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

图像块屏蔽颜色寄存器 (BLOCK_COVER_COLOR) 配置屏蔽的颜色，为读写寄存器。

- 偏移地址：0xC8
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

寄存器名	比特	名称	描述
BLOCK_COVER_START	[31:24]	Reserved	保留。
	[23:12]	block_cover_starty	开始图像块填充的行号。
	[11:0]	block_cover_startx	开始图像块填充的象素号。
BLOCK_COVER_SIZE	[31:24]	Reserved	保留。
	[23:12]	block_cover_height	图像块填充的高度（以行为单位）。
	[11:0]	Block_cover_width	图像块填充的宽度（以象素为单位）。
BLOCK_COVER_COLOR	[31:30]	Reserved	保留。
	[29:20]	Block_cover_y	填充色的 y 分量 (bayer_rgb 时为 G 分量)。
	[19:10]	Block_cover_u	填充色的 u 分量 (bayer_rgb 时为 R 分量)。
	[9:0]	Block_cover_v	填充色的 v 分量 (bayer_rgb 时为 B 分量)。

注：当数据位宽为 8bit 或 9bit 时，Block_cover_y、Block_cover_u、Block_cover_v 的高 8 位或高 9 位有效，低位补零。



11.7.16 亮度统计寄存器

亮度统计寄存器 (LUM_ADDER)，为只读寄存器。

- 偏移地址：0xCC
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:0]	Lum_adder	亮度数值累加。

注：当打开图像块屏蔽功能后，亮度统计寄存器对被屏蔽的图像块亮度不敏感。



12 视频输出单元 (VOU)

关于本章

本章描述内容如下表所示。

标题	内容
12.1 概述	概括介绍 VOU。
12.2 特点	概括介绍 VOU 的特点。
12.3 信号描述	描述 VOU 的视频接口的外部输入输出管脚信号。
12.4 接口协议	介绍 VOU 的主要接口协议。
12.5 工作方式	描述 LCD 与 SIO 的复用关系。
12.6 寄存器概览	概括介绍 VOU 的寄存器。
12.7 寄存器描述	详细描述 VOU 的寄存器。

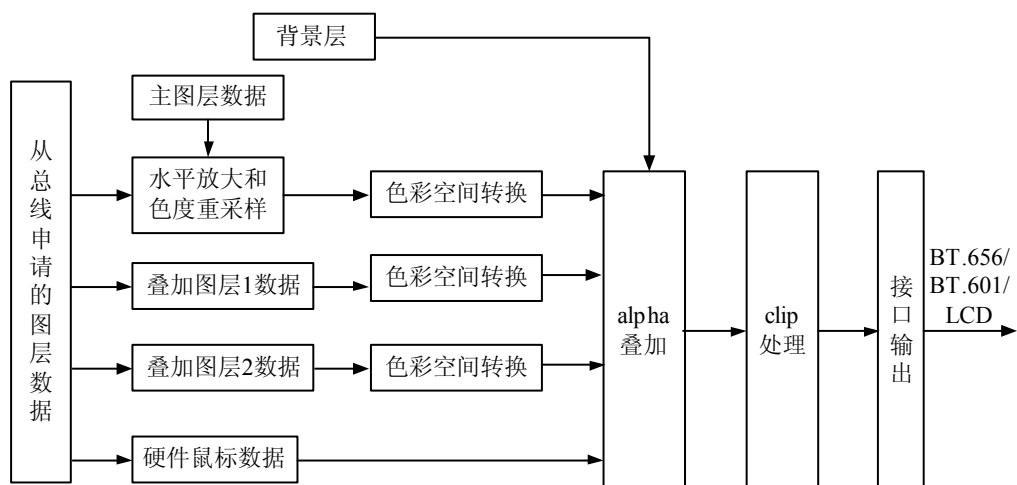


12.1 概述

视频输出单元 VOU (Video Output Unit) 可以通过 ITU-R BT.656/601 接口或者数字 LCD 接口, 把存入到指定位置的视频数据送给芯片外模块。在此过程中, VOU 可以对视频图像数据进行水平放大和色度重采样; 并且支持背景层、视频层、图形层和硬件鼠标层; 视频层和图形层支持 alpha 叠加和色度 keying。视频层、图形层的大小以及在显示屏的起始位置可任意调节。

VOU 功能框图如图 12-1 所示。

图12-1 VOU 功能框图



12.2 特点

视频输出单元有以下特点:

- 支持 8 位 ITU-R BT.656/ITU-R BT.601 YUV 4:2:2 标准输出接口 (PAL 制式/NTSC 制式@27MHz)
- 支持 TFT 24 位 RGB / YUV LCD 控制接口, 向下兼容 RGB 565、RGB 555、RGB 666 格式
- 支持 raw data 的数据传输方式, 可以对输入数据流不进行任何处理直接输出
- 支持隔行、逐行两种数据读取方式
- 水平方向支持 2 倍亮度 (Y)、2 倍色度 (UV) 插值
- 支持色度重新采样, 数据格式从 Interspersed 到 Co-sited 转换
- 输入主图像数据支持 YUV 4:2:0, YUV 4:2:2 和 YUV 4:4:4 数据格式, 亮度色度分开存储, 其中色度分量按字节间插存储
- 叠加图像支持 packed YUV 4:2:2+ α 、packed RGB 555+ α 、packed YUV 4:4:4+ α 、packed RGB 8:8:8+ α



- 支持 32 × 32 象素大小的 4 色硬件鼠标叠加
- 支持视频图像层、叠加图像 1、叠加图像 2 和硬件鼠标的四层叠加，其中叠加图像 1 和叠加图像 2 的叠加支持 129 层 alpha 叠加，支持 chroma keying 和 mask
- 支持 24BPP 的背景颜色层
- 支持在 ITU-R BT.601/656 模式下，对输出象素值 Clipping 处理，可以把输出象素值限制在用户配置范围内
- 支持内部 FIFO Underflow 告警，在数据低带宽的情况下发出中断

12.3 信号描述

本节描述了 VOU 的视频接口的外部输入输出管脚信号，如表 12-1 所示。

表12-1 VOU 视频输出接口信号

接口信号	方向	描述
VOCK	I/O	象素输出时钟。 根据接口特性，输入相应的时钟频率。 输出时，请参见表 6-2 描述。
VOHS	I/O	行同步脉冲，可以配置为输入输出。 <ul style="list-style-type: none"> • 在主模式下，配置为输出，表示同步行脉冲信号，脉冲宽度和脉冲起始位置以及高低电平可配置。 • 在从模式下，配置为输入，适合 ITU-R BT.601 标准。
VOVS	I/O	帧同步脉冲，可以配置为输入输出。 <ul style="list-style-type: none"> • 在主模式下，配置为输出。表示同步场信号脉冲，脉冲宽度和位置以及高低电平可配置。 • 在从模式下，配置为输入，适合 ITU-R BT.601 标准。
LCDCB	I/O	数据有效信号或者场号信号。高低电平可配置。 配置成为输出时可配置成为数据有效信号、场号信号。 0: 表示场号信号; 1: 表示数据有效信号。 配置成为输入时表示场号信号。
VODAT[7:0]	O	输出象素值。 <ul style="list-style-type: none"> • 当 TV 显示时，仅 VODAT[7:0]有效。 • 在 LCD 接口模式下：VODAT[7:0]为 B 分量或者 V 分量。 所有这行字节都是以高位右对齐的。



接口信号	方向	描述
LCDP[23:8]	O	输出象素值。 在 LCD 接口模式下： <ul style="list-style-type: none"> • LCDP[23:16]为 R 分量或者 Y 分量； • LCDP[15:8]为 G 分量或者 U 分量； • VODAT[7:0]为 B 分量或者 V 分量。 所有这行字节都是以高位右对齐的。

12.4 接口协议

视频输出接口支持 ITU-R BT.656/601 接口或者 LCD 接口。

12.5 工作方式

GPIO7[2]用于 LCD 输出和第 2 组 SIO 信号的复用控制。管脚复用关系如表 12-2 所示。

表12-2 SIO 与 LCD 复用关系对应表

位置	复用信号 1	复用信号 2	描述
K2	LCDP19	SIOXFS1	当 GPIO7[2]=0, 作为 LCDP19; 当 GPIO7[2]=1, 作为 SIOXFS1。
K3	LCDP18	SIODO1	当 GPIO7[2]=0, 作为 LCDP18; 当 GPIO7[2]=1, 作为 SIODO1。
K4	LCDP17	SIOXCK1	当 GPIO7[2]=0, 作为 LCDP17; 当 GPIO7[2]=1, 作为 SIOXCK1。
L1	LCDP16	SIODI1	当 GPIO7[2]=0, 作为 LCDP16; 当 GPIO7[2]=1, 作为 SIODI1。
L2	LCDP15	SIORFS1	当 GPIO7[2]=0, 作为 LCDP15; 当 GPIO7[2]=1, 作为 SIORFS1。



12.6 寄存器概览

VOU 地址位宽 32 位，地址范围：0x1012_0000~0x1012_00F8。VOU 寄存器如表 12-3 所示。

表12-3 VOU 寄存器概览（基址是 0x1012_0000）

偏移地址	名称	描述	页码
0x00	vo_ctrl	VOU 控制寄存器 ^b	12-6
0x04	vo_int_mask	VOU 中断使能寄存器 ^b	12-11
0x08	vo_int_status	VOU 中断状态寄存器 ^b	12-12
0x0C	vo_status	VOU 状态寄存器	12-12
0x10	vo_vsync1	垂直同步寄存器 1 ^b	12-13
0x14	vo_vsync2	垂直同步寄存器 2 ^b	12-14
0x18	vo_hsync	水平同步寄存器 ^b	12-14
0x1C	vo_image	图像寄存器 ^a	12-15
0x20	vo_ovl1_image	叠加图像 1 寄存器 ^a	12-16
0x24	vo_ovl2_image	叠加图像 2 寄存器 ^a	12-16
0x28	Reserved	保留	-
0x2C	vo_image_off	图像偏移寄存器 ^a	12-17
0x30	vo_bg_color	背景颜色寄存器 ^a	12-17
0x34	vo_clip	clip 值寄存器 ^a	12-17
0x38	vo_mask	key 的 mask 值寄存器 ^a	12-18
0x3C	vo_ovl1_key	叠加图像 1 key 值寄存器 ^a	12-18
0x40	vo_ovl2_key	叠加图像 2 key 值寄存器 ^a	12-19
0x44	vo_hc_color0	硬件鼠标颜色 0 寄存器 ^a	12-19
0x48	vo_hc_color1	硬件鼠标颜色 1 寄存器 ^a	12-19
0x4C	vo_hc_color2	硬件鼠标颜色 2 寄存器 ^a	12-20
0x50	vo_hc_color3	硬件鼠标颜色 3 寄存器 ^a	12-20
0x54	vo_ovl1_start	叠加图像 1 起始位置寄存器 ^a	12-20
0x58	vo_ovl2_start	叠加图像 2 起始位置寄存器 ^a	12-21
0x5C	vo_hc_start	硬件鼠标起始位置寄存器 ^a	12-21



偏移地址	名称	描述	页码
0x60	vo_ovl1_alpha	叠加图像 1 alpha 值寄存器 ^a	12-22
0x64	vo_ovl2_alpha	叠加图像 2 alpha 值寄存器 ^a	12-22
0x68	vo_mladdr	主图像亮度分量地址寄存器 ^a	12-23
0x6C	vo_mcaddr	主图像色度分量地址寄存器 ^a	12-23
0x70	Reserved	保留	-
0x74	vo_ovl1addr	叠加图像 1 地址寄存器 ^a	12-23
0x78	vo_ovl2addr	叠加图像 2 地址寄存器 ^a	12-24
0x7C	vo_hcaddr	硬件鼠标地址寄存器 ^a	12-24
0x80	vo_moffset	主图像的亮度和色度行偏移量寄存器 ^a	12-24
0x84	vo_ovloffset	叠加图像行偏移量寄存器 ^a	12-25
0x88~0xEC	reserved	保留	-
0xF0	reserved	保留	-
0xF4	reserved	保留	-
0xF8	reserved	保留	-
0xFC	reserved	保留	-

注:

a: 非及时性寄存器, 只有在固定时刻才会被载入到工作寄存器;

b: 及时性寄存器, 软件写该寄存器将及时生效。

12.7 寄存器描述

本节详细描述了 VOU 的寄存器。

12.7.1 控制寄存器和硬件鼠标

本小节描述了控制寄存器和硬件鼠标的 3 种模式。

控制寄存器

控制寄存器控制 VOU 的整个工作流程以及一些配置信息。

- 偏移地址: 0x00
- 操作类型: R/W
- 复位值: 0x0



- 复位方式: h/s

表12-4 控制寄存器

比特	名称	描述
[31]	vo_en	VOU 工作使能。 在 VOU 启动 (即 vo_en 置 1) 之前, 先配置好需要显示的第一帧数据的所有寄存器, 最后再置 vo_en 为 1。
[30]	main_en	主图像使能, 默认为 0。
[29]	ovl1_en	叠加图像 1 使能。
[28]	ovl2_en	叠加图像 2 使能。
[27]	hc_en	硬件鼠标使能。
[26]	raw_start	raw data 模式, 寄存器更新使能。 在 vo_en 的情况下, 在当前无 raw data 传输情况下, 如果 raw_start 为高电平, 则进行一次 raw data 的传输。
[25]	little_end ian	数据以小段模式。默认为 0。
[24]	master_ mode	工作模式。 0: 从模式; 1: 主模式。 默认为从模式。
[23:20]	main_mo de	请参见表 12-5。
[19:18]	hc_mode	硬件鼠标模式。 00: 双色和透明模式; 01: 4 色模式; 10: 3 色和透明模式; 11: 保留。 模式相对于颜色值的使用请参见“硬件鼠标的 3 种模式”。
[17]	rgb_out	输出 RGB 数据。 在 TV 输出模式下: 该位无效, 输出总是 YUV 数据。 在 LCD 输出模式下: <ul style="list-style-type: none"> • 在视频数据使能的情况下 (main_en = 1), 该位不起作用, 数据输出类型由 Main_mode 规定; • 在视频数据未使能的情况下 (main_en = 0) • 当 rgb_out 为 1 时, 输出 RGB 数据; 否则为输出数据格式和输入保持一致。



比特	名称	描述
[16]	clip_en	TV 输出 clipping 使能。
[15:14]	ovl1_type	叠加图像 1 的类型： 00: YUV 4:2:2+ α ; 01: YUV 4:4:4+ α ; 10: RGB 5:5:5+ α ; 11: RGB 8:8:8+ α 。
[13]	ovl1_key_en	第 1 层叠加 keying 使能。
[12]	ovl1_mask_en	第 1 层 keying mask 使能。
[11:10]	ovl2_type	叠加图像 2 的类型。具体描述同 ovl1_type。
[9]	ovl2_key_en	第 2 层叠加 keying 使能。
[8]	ovl2_mask_en	第 2 层 keying mask 使能。
[7]	ieo	invert output enable: 翻转输出有效信号指示。 0: 高电平有效; 1: 低电平有效。
[6]	ihs	invert horizontal signal: 翻转水平同步信号。 0: 高脉冲有效; 1: 低脉冲有效。
[5]	ivs	invert vertical signal: 翻转垂直同步信号。 0: 高脉冲有效; 1: 低脉冲有效。
[4]	cb	vo_cb 信号选择。 0: field2 信号。 1: 数据有效指示;
[3]	ifs	invert field2 signal: 翻转 field2 信号 (仅当 cb=0 时有效)。 0: 奇数场用 0 表示, 偶数场用 1 表示; 1: 奇数场用 1 表示, 偶数场用 0 表示。
[2]	field_syn	从模式时, 由 field2 作为垂直同步信号。



比特	名称	描述
[1:0]	out_format	数据接口格式： 00: 表示 BT656 (默认)； 01: 表示 BT601； 10: 表示 LCD 显示； 11: 表示 Raw Data 数据传输。

控制寄存器 main_mode 位的描述如表 12-5 所示。

表12-5 main_mode 的描述

bit[23:20]	描述
0000	YUV 4:2:2 Co-sited 格式输入，无放大。
0001	YUV 4:2:2 Interspered 格式输入，Interspered 到 co-sited 色度重新采样。
0010	YUV 4:2:2 Co-sited 格式输入，水平放大两倍，默认值。
0011	YUV 4:2:2 Interspered 格式输入，水平放大两倍且色度从 Interspered 到 Co-sited 转换。
0100	YUV 4:2:0 Co-sited 格式输入，无放大，色度垂直方向重用。
0101	YUV 4:2:0 Interspered 格式输入，Interspered 到 co-sited 色度重新采样，色度垂直方向重用。
0110	YUV 4:2:0 Co-sited 格式输入，水平放大两倍，色度垂直方向重用。
0111	YUV 4:2:0 Interspered 格式输入，水平放大两倍且色度从 Interspered 到 Co-sited 转换，色度垂直方向重用。
1000	YUV 4:4:4 输入，YUV 到 RGB 转换。
1001	YUV 4:4:4 输入，水平放大两倍，YUV 到 RGB 转换。
1010	YUV 4:2:2 Co-sited 格式输入，色度水平放大两倍，YUV 到 RGB 转换。
1011	YUV 4:2:2 Interspered 格式输入，色度水平放大两倍且色度从 Interspered 到 Co-sited 转换，YUV 到 RGB 转换。
1100	YUV 4:4:4 输入，无水平放大。
1101	YUV 4:4:4 输入，水平放大两倍。
1110	YUV 4:2:0 Co-sited 格式输入，色度水平放大两倍，色度垂直方向重用，YUV 到 RGB 转换。



bit[23:20]	描述
1111	YUV 4:2:0 Interspersed 格式输入，色度水平放大两倍且色度从 Interspersed 到 Co-sited 转换，色度垂直方向重用，YUV 到 RGB 转换。

硬件鼠标的 3 种模式

控制寄存器中硬件鼠标有以下 3 种模式。

- 32×32×2bpp 双色和透明模式

该模式用于支持微软的 Windows 鼠标数据格式。每一个象素用 2 位表示，代表 4 种颜色。

- 第 1、2 种颜色用于画一个鼠标
- 第 3 种颜色用于透明处理（允许在鼠标后面的主图像被显示）
- 第 4 种颜色用于反向透明处理（鼠标后面的主图像被显示，但是其显示颜色与原主图像颜色相反）

该模式列表如表 12-6 所示。

表12-6 32×32×2bpp 双色和透明模式列表

bit/pixel	在相应位置的颜色显示
00	鼠标颜色 0。
01	鼠标颜色 1。
10	透明处理。在当前硬件鼠标后的主图像象素被显示。
11	反向透明处理。在当前硬件鼠标后的主图像被反色显示。

- 32×32×2bpp 四色模式

该模式提供 4 种颜色来显示鼠标。每一个象素用 2 位表示，代表如表 12-7 所示的 4 种颜色。

表12-7 32×32×2bpp 四色模式列表

bit/pixel	在相应位置的颜色显示
00	鼠标颜色 0。
01	鼠标颜色 1。
10	鼠标颜色 2。
11	鼠标颜色 3。



- 32×32×2bpp 三色和透明模式
该模式提供 3 种颜色来显示鼠标，其中一种颜色对主图像透明处理（覆盖在鼠标后的象素被显示）。每一个象素用 2 位表示，4 种颜色的显示如表 12-8 所示。

表12-8 2×32×2bpp 三色和透明模式列表

bit/pixel	在相应位置的颜色显示
00	鼠标颜色 0。
01	鼠标颜色 1。
10	鼠标颜色 2。
11	透明处理。覆盖在硬件鼠标后的象素被显示。

12.7.2 中断使能寄存器

中断使能寄存器用于对各个中断使能。当中断使能位为 1 时，相应的中断使能，否则该中断被屏蔽。

- 偏移地址：0x04
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31]	FM (Frame_Mode)	帧模式，该位只有在隔行输出情况下（TV 输出）才有效。 0：当前所有配置寄存器都以场方式进行； 1：当前所有寄存器都以帧方式进行。
[30:7]	Rerved	保留，读为 0。
[6]	bus_err_int_en	AHB master 读 AHB 总线错误中断使能。
[5]	lbw_int_en	带宽分配太低出现内部 FIFO underflow 中断使能。
[4]	ovl1_rint_en	叠加图像 1 数据读取完毕中断使能。
[3]	ovl2_rint_en	叠加图像 2 数据读取完毕中断使能。
[2]	hc_rint_en	硬件鼠标数据读取完毕中断使能。
[1]	main_rint_en	主图像数据读取完毕中断使能。
[0]	reload_int_en	图像地址重新 load 中断使能。



12.7.3 中断状态寄存器

中断状态寄存器的某一位产生中断，并且该位对应的中断使能位为 1 时，相应的状态位置 1。中断状态寄存器通过软件写入 1 的方式复位。

- 偏移地址：0x08
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:7]	Reserved	保留，读为 0。
[6]	bus_err_int	AHB master 读 AHB 总线错误中断。
[5]	lbw_int	带宽分配太低出现内部 FIFO underflow 中断。
[4]	ovl1_rint	叠加图像 1 数据读取完毕中断。
[3]	ovl2_rint	叠加图像 2 数据读取完毕中断。
[2]	hc_rint	硬件鼠标数据读取完毕中断。
[1]	main_rint	主图像数据读取完毕中断。
[0]	reload_int	图像地址重新 load 中断。

12.7.4 状态寄存器

状态寄存器表示硬件当前的工作状态，软件只能读。

- 偏移地址：0x0C
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:12]	Reserved	保留，读为 0。
[11]	main_done	主图像数据读取完毕。
[10]	ovl1_done	叠加图像 1 数据读取完毕。
[9]	ovl2_done	叠加图像 2 数据读取完毕。
[8]	hc_done	硬件鼠标数据读取完毕。
[7]	ml_under_flow	主图像亮度分量内部 FIFO underflow。



比特	名称	描述
[6]	mc_under_flow	主图像色度分量内部 FIFO underflow。
[5]	Reserved	保留，读为 0。
[4]	ovl1_under_flow	叠加图像 1 内部 FIFO underflow。
[3]	ovl2_under_flow	叠加图像 2 内部 FIFO underflow。
[2]	bus_err	Master 读取总线错误状态。
[1]	Reserved	保留，读为 0。
[0]	field2	第 2 场指示信号。

12.7.5 垂直同步寄存器 1

TV 接口的奇场或者在 LCD 接口垂直同步寄存器，默认为 PAL 制式，也可以配置为 NTSC 制式。

该寄存器配置后立即生效，影响管脚 VOVS 的时序。

- 偏移地址：0x10
- 操作类型：R/W
- 复位值：0x1501_511F
- 复位方式：h/s

比特	名称	描述
TV 接口		
[31:24]	act1_voff	第 1 场开始到活动图像行距离（其值为实际行数减 1）。 默认（PAL 制式）为 21，NTSC 制式配置为 15。
[23:16]	act1_vbb	第 1 场活动图像结束到第 2 场开始的行距（其值为实际行数减 1）。 默认为 1，NTSC 制式配置为 1。
[15:12]	vsyn_width	场同步信号宽度（仅对 ITU-R BT.601 有效），在 vsyn_width 的最低位为 0.5 小数宽度标志，vsyn_wdith[3:1]为整数。 默认为 2.5（0101），NTSC 制式配置为 6。
[11:0]	act1_height	第 1 场活动图像的高度（行数，其值为实际行数减 1）。 默认值为 287，NTSC 制式配置为 243。
LCD 接口		
[31:24]	vfp	LCD 显示最后一行到垂直同步信号之间的行距（其值为实际行数减 1），以行为单位。



比特	名称	描述
[23:16]	vbp	LCD 显示垂直同步信号与下一帧第 1 行之间的行距（其值为实际行数减 1），以行为单位。
[15:12]	vsw	LCD 显示垂直同步信号的宽度（其值为实际行数减 1）。
[11:0]	lpp	LCD 一帧包含的行数（其值为实际行数减 1）。

12.7.6 垂直同步寄存器 2

TV 接口的偶场同步配置寄存器，默认为 PAL 制式，也可以配置为 NTSC 制式。

该寄存器配置后立即生效，影响管脚 VOVS 的时序。

- 偏移地址：0x14
- 操作类型：R/W
- 复位值：0x1601_811F
- 复位方式：h/s

比特	名称	描述
[31:24]	act2_voff	第 2 场开始到活动图像行距离（其值为实际行数减 1）。默认（PAL 制式）为 22，NTSC 制式配置为 16。
[23:16]	act2_vbb	第 2 场活动图像结束到第 1 场开始的行距（其值为实际行数减 1）。默认为 1，NTSC 制式配置为 2。
[15:12]	hsyn_width_msb	水平同步信号宽度的高 4 位（仅限于 ITU-R BT.601），默认为（1000），水平同步脉冲宽度默认为 128。NTSC 制式配置为 128，该位配置位 4'b0000。
[11:0]	act2_height	第 2 场活动图像的高度（行数，其值为实际行数减 1）。默认为 287，NTSC 制式配置为 242。

12.7.7 水平同步寄存器

TV 或者 LCD 接口情况下，水平同步配置寄存器，默认为 PAL 制式，可以配置为 NTSC 制式。

该寄存器配置后立即生效，影响管脚 VOVS 的时序。

- 偏移地址：0x18
- 操作类型：R/W
- 复位值：0x41D7_059F



- 复位方式: h/s

比特	名称	描述
TV 接口		
[31:22]	Reserved	保留。
[21:16]	act_hbb	活动区域结束到本行的结束距离，以时钟为单位。 默认 (PAL 制式) 为 (24-1)，NTSC 制式配置为 (244-1)。
[15:12]	hsyn_width_lsb	水平同步信号宽度的低 4 位 (仅限于 ITU-R BT.601)，默认为 (0000)，水平同步脉冲宽度默认为 128。NTSC 制式配置为 (32-1)。
[11:0]	act_width	活动图像宽度，以时钟为单位，所以在 ITU-R BT.656/601 模式下为像素的两倍 (其值为实际值减 1)。例如，默认为 1440 (720 个像素) 其写入的值为 1439。NTSC 制式配置为 128。
LCD 接口		
[31:24]	hfp	LCD 显示最后一个像素到水平同步信号之间的像素距 (其值为实际值减 1)，以像素为单位。
[23:16]	hbp	LCD 显示水平同步信号与下一行第 1 个像素之间的像素距离 (其值为实际值减 1)，以像素为单位。
[15:12]	hsw	LCD 显示水平同步信号的宽度 (其值为实际值减 1)。
[11:0]	ppl	LCD 一行包含的像素数 (其值为实际值减 1)。

12.7.8 图像寄存器

该寄存器存储了视频图像层图像的宽度和高度。

- 偏移地址: 0x1C
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:24]	Reserved	保留，读为 0。
[23:12]	image_height	视频输出图像的高度 (其值为实际高度减 1)，以行为单位。



比特	名称	描述
[11:0]	image_width	<p>视频输出图像的宽度，以时钟为单位。</p> <ul style="list-style-type: none"> 在 LCD 模式下，象素单位也就是时钟单位； 在 TV 显示模式下，象素单位为时钟单位的 1/2。 <p>写入值为实际值减 1。</p> <p>在 raw data 模式下，image_height 和 image_width 一起构成 raw data 数据大小指示（其值为 raw data 实际数据减 1）。image_height 为 raw data 输出的高位，image_width 为 raw data 数据大小的低位。</p>

12.7.9 叠加图像 1 寄存器

- 偏移地址：0x20
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:24]	Reserved	保留，读为 0。
[23:12]	ovl1_height	叠加图像 1 的高度（实际行数减 1），以行为单位。
[11:0]	ovl1_width	叠加图像 1 的宽度（填入值为实际象素宽度减 1），以象素为单位。

12.7.10 叠加图像 2 寄存器

- 偏移地址：0x24
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:24]	Reserved	保留，读为 0。
[23:12]	ovl2_height	叠加图像 2 的高度（以行为单位），实际行数减 1。
[11:0]	ovl2_width	叠加图像 2 的宽度（以象素为单位），填入值为实际象素宽度减 1。



12.7.11 图像偏移寄存器

- 偏移地址: 0x2C
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:24]	Reserved	保留, 读为 0。
[23:12]	image_voff	实际传输图像第 1 行相对于活动图形开始行的行距。
[11:0]	image_hoff	实际传输图像开始像素相对于活动图形开始像素的像素距离。

12.7.12 背景颜色寄存器

背景颜色寄存器规定了背景层的颜色 (YUV 数据格式)。

- 偏移地址: 0x30
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:24]	Reserved	保留, 读为 0。
[23:16]	bg_y	背景颜色 Y 分量。
[15:8]	bg_u	背景颜色 U 分量。
[7:0]	bg_v	背景颜色 V 分量。

12.7.13 clip 值寄存器

在 YUV 格式显示需要 clip 时, clip 值寄存器规定了亮度和色度的范围。

- 偏移地址: 0x34
- 操作类型: R/W
- 复位值: 0xEB10_F010
- 复位方式: h/s



比特	名称	描述
[31:24]	high_luma	亮度信号的最高门限值（默认值为 235）。
[23:16]	low_luma	亮度信号的最低门限值（默认值为 16）。
[15:8]	high_chroma	色度信号的最高门限值（默认值为 240）。
[7:0]	low_chroma	色度信号的最低门限值（默认值为 16）。

12.7.14 Key 的 mask 值寄存器

在做色度 keying 时，为了提高色度 key 值的范围，对每一个色度 key 值增加了低位的 mask 位。被屏蔽的位不参与比较。

- 偏移地址：0x38
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:16]	Reserved	保留，读为 0。
[15:12]	ovl2_mask_y	叠加图像 2 的亮度 mask 位。
[11:8]	ovl2_mask_c	叠加图像 2 的色度 mask 位。
[7:4]	ovl1_mask_y	叠加图像 1 的亮度 mask 位。
[3:0]	ovl1_mask_c	叠加图像 1 的色度 mask 位。

12.7.15 叠加图像 1 key 值寄存器

- 偏移地址：0x3C
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:24]	Reserved	保留，读为 0。
[23:16]	ovl1_key_y	叠加图像 1 的亮度 keying 比较值。
[15:8]	ovl1_key_u	叠加图像 1 的色度 U keying 比较值。
[7:0]	ovl1_key_v	叠加图像 1 的色度 V keying 比较值。



12.7.16 叠加图像 2 key 值寄存器

- 偏移地址: 0x40
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:24]	Reserved	保留, 读为 0。
[23:16]	ovl2_key_y	叠加图像 2 的亮度 keying 比较值。
[15:8]	ovl2_key_u	叠加图像 2 的色度 U keying 比较值。
[7:0]	ovl2_key_v	叠加图像 2 的色度 V keying 比较值。

12.7.17 硬件鼠标颜色 0 寄存器

- 偏移地址: 0x44
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:24]	Reserved	保留, 读为 0。
[23:16]	hc_yr0	硬件鼠标色度 0 的亮度 (或者红色) 值。
[15:8]	hc_ug0	硬件鼠标色度 0 的色度 U (或者绿色) 值。
[7:0]	hc_vb0	硬件鼠标色度 0 的色度 V (或者蓝色) 值。

12.7.18 硬件鼠标颜色 1 寄存器

- 偏移地址: 0x48
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:24]	Reserved	保留, 读为 0。
[23:16]	hc_yr1	硬件鼠标色度 1 的亮度 (或者红色) 值。



比特	名称	描述
[15:8]	hc_ug1	硬件鼠标色度 1 的色度 U (或者绿色) 值。
[7:0]	hc_vb1	硬件鼠标色度 1 的色度 V (或者蓝色) 值。

12.7.19 硬件鼠标颜色 2 寄存器

- 偏移地址: 0x4C
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:24]	Reserved	保留, 读为 0。
[23:16]	hc_yr2	硬件鼠标色度 2 的亮度 (或者红色) 值。
[15:8]	hc_ug2	硬件鼠标色度 2 的色度 U (或者绿色) 值。
[7:0]	hc_vb2	硬件鼠标色度 2 的色度 V (或者蓝色) 值。

12.7.20 硬件鼠标颜色 3 寄存器

- 偏移地址: 0x50
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:24]	Reserved	保留, 读为 0。
[23:16]	hc_yr3	硬件鼠标色度 3 的亮度 (或者红色) 值。
[15:8]	hc_ug3	硬件鼠标色度 3 的色度 U (或者绿色) 值。
[7:0]	hc_vb3	硬件鼠标色度 3 的色度 V (或者蓝色) 值。

12.7.21 叠加图像 1 起始位置寄存器

- 偏移地址: 0x54
- 操作类型: R/W



- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:24]	Reserved	保留, 读为 0。
[23:12]	ovl1_start_line	叠加图像 1 相对于实际传输图像, 开始叠加时的行坐标, 从 0 开始。
[11:0]	ovl1_start_pixel	叠加图像 1 相对于实际传输图像, 开始叠加时的像素坐标, 从 0 开始。 例如 (0, 0) (第 1 行的第 1 个像素开始叠加) 为图像顶点。

12.7.22 叠加图像 2 起始位置寄存器

- 偏移地址: 0x58
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:24]	Reserved	保留, 读为 0。
[23:12]	ovl2_start_line	叠加图像 2 相对于实际传输图像, 开始叠加时的行坐标, 从 0 开始。
[11:0]	ovl2_start_pixel	叠加图像 2 相对于实际传输图像, 开始叠加时的像素坐标, 从 0 开始。 例如 (0, 0) (第 1 行的第 1 个像素开始叠加) 为图像顶点。

12.7.23 硬件鼠标起始位置寄存器

- 偏移地址: 0x5C
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s



比特	名称	描述
[31]	hc_reload	硬件鼠标数据重新载入使能。 在一场/帧的开始, 如果该位为 1, 则重新从 SDRAM 中载入硬件鼠标数据, 并且硬件自动清零该位。
[30:24]	Reserved	保留, 读为 0。
[23:12]	hc_start_line	硬件鼠标相对于实际传输图像, 开始叠加时的行偏移量, 以行为单位。
[11:0]	hc_start_pixel	硬件鼠标相对于实际传输图像, 开始叠加时的象素偏移量, 以象素为单位。

12.7.24 叠加图像 1 alpha 值寄存器

- 偏移地址: 0x60
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:16]	Reserved	保留, 读为 0。
[15:8]	ovl1_alpha0	叠加图像 1 的 alpha 值 0。 RGB 5:5:5+ α 和 YUV 4:2:2+ α 中的 α 位来选择该寄存器值作为 alpha 叠加的 alpha 值。
[7:0]	ovl1_alpha1	叠加图像 1 的 alpha 值 1。 RGB 5:5:5+ α 和 YUV 4:2:2+ α 中的 α 位来选择该寄存器值作为 alpha 叠加的 alpha 值。

12.7.25 叠加图像 2 alpha 值寄存器

- 偏移地址: 0x64
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:16]	Reserved	保留, 读为 0。



比特	名称	描述
[15:8]	ovl2_alpha0	叠加图像 2 的 alpha 值 0。 RGB 5:5:5+ α 和 YUV 4:2:2+ α 中的 α 位来选择该寄存器值作为 alpha 叠加的 alpha 值。
[7:0]	ovl2_alpha1	叠加图像 2 的 alpha 值 1。 RGB 5:5:5+ α 和 YUV 4:2:2+ α 中的 α 位来选择该寄存器值作为 alpha 叠加的 alpha 值。

12.7.26 视频图像亮度分量地址寄存器

- 偏移地址: 0x68
- 操作类型: R/W
- 复位值: 0xFFFF_FFFC
- 复位方式: h/s

比特	名称	描述
[31:0]	vo_mladdr	主图像亮度存储 SDRAM 首地址 (为 WORD 地址, 低 2 位写入无效, 读数为 0), 或者 Raw Data 模式下数据块的首地址。

12.7.27 视频图像色度分量地址寄存器

- 偏移地址: 0x6C
- 操作类型: R/W
- 复位值: 0xFFFF_FFFC
- 复位方式: h/s

比特	名称	描述
[31:0]	vo_mcaddr	主图像色度存储 SDRAM 首地址 (为 WORD 地址, 低 2 位写入无效, 读数为 0)。

12.7.28 叠加图像 1 地址寄存器

- 偏移地址: 0x74
- 操作类型: R/W
- 复位值: 0xFFFF_FFFC
- 复位方式: h/s



比特	名称	描述
[31:0]	vo_ovl1addr	叠加图像 1 存储 SDRAM 首地址 (为 WORD 地址, 低 2 位写入无效, 读出为 0)。

12.7.29 叠加图像 2 地址寄存器

- 偏移地址: 0x78
- 操作类型: R/W
- 复位值: 0xFFFF_FFFC
- 复位方式: h/s

比特	名称	描述
[31:0]	vo_ovl2addr	叠加图像 2 存储 SDRAM 首地址 (为 WORD 地址, 低 2 位写入无效, 读出为 0)。

12.7.30 硬件鼠标地址寄存器

- 偏移地址: 0x7C
- 操作类型: R/W
- 复位值: 0xFFFF_FFFC
- 复位方式: h/s

比特	名称	描述
[31:0]	vo_hcaddr	硬件鼠标存储 SDRAM 首地址 (为 WORD 地址, 低 2 位写入无效, 读出为 0)。

12.7.31 主图像亮度和色度行偏移量寄存器

- 偏移地址: 0x80
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:16]	mluma_offset	主图像亮度信号存储在 SDRAM 的行首与行首之间的地址距离, 该值为 WORD 地址, 低 2 位写入无效, 读出为 0。



比特	名称	描述
[15:0]	mchroma_offset	主图像色度信号存储在 SDRAM 的行首与行首之间的地址距离，该值为 WORD 地址，低 2 位写入无效，读出为 0。

12.7.32 叠加图像行偏移量寄存器

- 偏移地址：0x84
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:16]	ovl2_offset	叠加图像 2 存储在 SDRAM 的行首与行首之间的地址距离，该值以 WORD 为单位。
[15:0]	ovl1_offset	叠加图像 1 存储在 SDRAM 的行首与行首之间的地址距离，该值为 WORD 地址，低 2 位写入无效，读出为 0。



13 串行输入输出接口 (SIO)

关于本章

本章描述内容如下表所示。

标题	内容
13.1 概述	概括介绍 SIO。
13.2 特点	描述 SIO 支持的 PCM 接口、I ² S 接口特点。
13.3 信号描述	描述 SIO 的外部输入输出管脚信号。
13.4 工作方式	描述 SIO 的接口时序和应用说明
13.5 寄存器概览	概括介绍 SIO 的寄存器。
13.6 寄存器描述	详细描述 SIO 的寄存器。



13.1 概述

SIO (Serial Input / Output) 接口支持 PCM 接口、I²S 接口或者 DSP 模式的音频接口。其中：

- PCM 接口主要用于语音通道；
- I²S 接口主要用于音频通道，如音乐回放等。

SIO 接口支持 8kHz、16kHz、22.05kHz、24kHz、44.1kHz 以及 48kHz 采样率。

13.2 特点

本节描述了 SIO 支持的 PCM 接口、I²S 接口特点。

13.2.1 PCM 接口特点

PCM 的接口特点如下：

说明

在 Hi3510 的 SIO 中，只支持集成 PCM，即 PCM 接口和 I²S 接口使用同一组管脚。

- 支持 8 位、16 位线性 PCM 编码
- 支持由 SIO 模块自己产生 bit Clock 以及帧同步信号，也可支持外接时钟及同步信号
- PCM 接口模块帧同步信号仅支持短脉冲同步信号（同步信号的持续时间为 1 个时钟周期）
- SIO 发送通道和接收通道有独立的中断信号，也可支持组合中断输出
- 帧数据可以在时钟上升沿或下降沿发送/接收
- PCM 接口只支持同步信号和数据对齐的方式

13.2.2 I²S 接口特点

I²S 的接口特点如下：

- I²S 接口支持 8 位、16 位工作模式
- I²S 模式仅支持 MSB first 模式
- I²S 接口在数据接收时：
 - 如果接收的数据宽度超过移位寄存器的最大宽度，则超出部分被截尾
 - 如果接收数据宽度比移位寄存器宽度小，先写入 MSB，后面多余部分补零
- I²S 接收通道和发送通道具有独立的 FIFO；每个通道的左声道和右声道也有独立的 FIFO，其 FIFO 深度为 8×16bit
- I²S 支持 FIFO 使能功能。在 FIFO 不使能状态下，接收和发送数据均不经过 FIFO，直接在一个 Buffer 中缓存



- I²S 支持 TX 和 RX 通道单独使能，如果某个通道不使能，则该通道的控制单元及数据存储单元都不会工作，以节省功耗
- 在 I²S 工作模式下，可支持由本模块产生或由对端芯片提供时钟和同步信号
- 支持 DSP 模式 I²S 接口
- SIO 支持 Burst 模式的 DMA 操作

13.3 信号描述

本节描述 SIO 的外部输入输出管脚信号。在 Hi3510 中提供了 2 组 SIO 接口，第 1 组 SIO 信号如表 13-1 所示。第 2 组 SIO 信号与 LCD 输出复用，请参见表 12-2。

表13-1 SIO 接口信号描述

信号名	方向	描述
SIODI0	I	I ² S 或 PCM 接口数据输入。
SIODO0	O	I ² S 或 PCM 接口数据输出。
SIOXFS0	I/O	I ² S 左右通道标识或 PCM 帧同步信号。
SIOCK0	I/O	I ² S 或 PCM 接口位流时钟。
SIORFS0	I/O	SIORFS0 为从模式下的 SIO 帧同步输入信号。 复用关系请参见表 20-3 的 GPIO2[0]的描述。

13.4 工作方式

本节描述了 SIO 的接口时序和应用说明。

13.4.1 时序

PCM 模式接口时序

PCM 接口支持在时钟的下降沿或者时钟的上升沿进行数据的发送和接收，其时序图如图 13-1、图 13-2 所示。

图13-1 PCM 接口时序（上升沿发送）

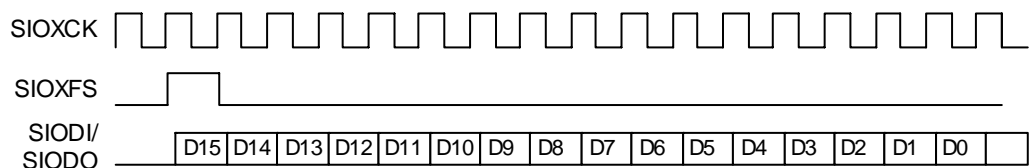
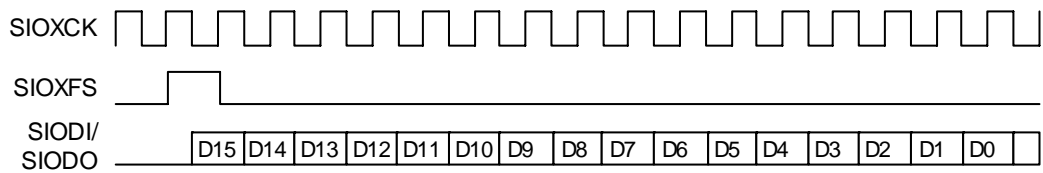




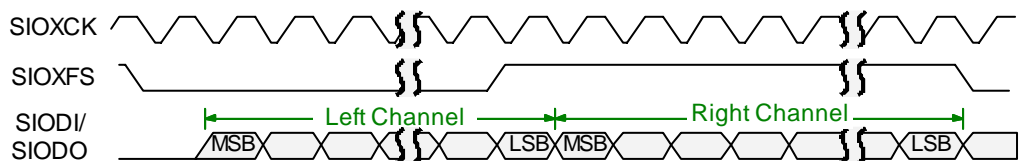
图13-2 PCM 接口时序 (下降沿发送)



I²S 模式接口时序

I²S 模式接口时序如图 13-3 所示。

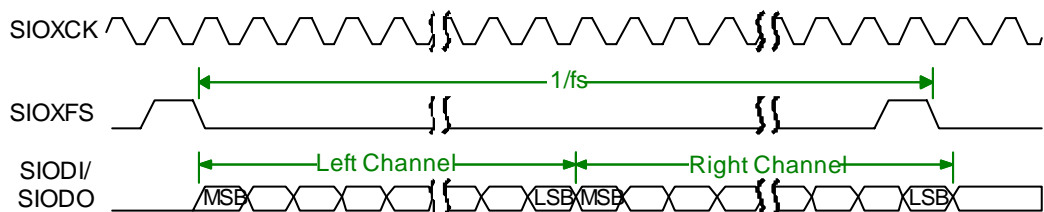
图13-3 I2S 接口时序



DSP 模式接口时序

DSP 模式接口时序如图 13-4 所示。

图13-4 音频接口 DSP MODE 时序



13.4.2 应用说明

SIO 支持单独的 (发送通道和接收通道分开) 使能控制, 以配合产品的低功耗设计。

说明

在某些应用场合, 如播放音乐时, 只需要使用其中的一个通道 (如发送通道), 此时可以关闭另外一个通道。在不使用 SIO 的时候, 建议将发送通道和接收通道全部关闭。

SIO 还提供了 DMA 接口, 可以直接进行数据的搬移工作, 以节省处理器资源, 避免 SIO 频繁中断处理器, 降低处理器的性能。



13.5 寄存器概览

SIO 有两组寄存器 SIO0 和 SIO1，两组寄存器的功能一样，只是基地址不同。

- SIO0 地址位宽 16 位，地址范围：0x8008_0000~0x8008_FFFF，如表 13-2 所示。
- SIO1 地址位宽 16 位，地址范围：0x9002_0000~0x9002_FFFF，如表 13-3 所示。

表13-2 SIO0 寄存器概览（基址是 0x8008_0000）

偏移地址	寄存器名称	页码
0x040	SIO_MODE	13-6
0x042	SIO_INTR_STATUS	13-7
0x044	SIO_INTR_CLR	13-7
0x046~0x048	发送数据寄存器 (I2S_LEFT_XD、I2S_RIGHT_XD、PCM_XD)	13-7
0x04A~0x04C	接收数据寄存器 (I2S_LEFT_RD、I2S_RIGHT_RD、PCM_RD)	13-8
0x04E~0x050	I2S_CT_SET	13-8
0x052	SIO_ICD	13-10
0x054	SIO_RX_STA	13-10
0x056	SIO_TX_STA	13-10
0x058~0x05A	SIO_PCM_CT_SET/SIO_PCM_CT_CLR	13-11

表13-3 SIO1 寄存器概览（基址是 0x9002_0000）

偏移地址	寄存器名称	页码
0x040	SIO_MODE	13-6
0x042	SIO_INTR_STATUS	13-7
0x044	SIO_INTR_CLR	13-7
0x046~0x048	发送数据寄存器 (I2S_LEFT_XD、I2S_RIGHT_XD、PCM_XD)	13-7
0x04A~0x04C	接收数据寄存器 (I2S_LEFT_RD、I2S_RIGHT_RD、PCM_RD)	13-8
0x04E~0x050	I2S_CT_SET	13-8
0x052	SIO_ICD	13-10
0x054	SIO_RX_STA	13-10



偏移地址	寄存器名称	页码
0x056	SIO_TX_STA	13-10
0x058~0x05A	SIO_PCM_CT_SET/SIO_PCM_CT_CLR	13-11

13.6 寄存器描述

SIO 的两组寄存器中，两组寄存器的功能一样，在本节以其中一组为例来描述 SIO 寄存器。

13.6.1 SIO_MODE

SIO_MODE 寄存器用来对 SIO 工作的基本模式进行选择。

- 偏移地址：0x040
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:2]	Reserved	保留，读回值为 0。
[1]	sep_pcm	分离 PCM 模式选择。 0: 集成 PCM 模式； 1: 分离 PCM 模式。 本版本该位应设置为 0。
[0]	pcm_mode	PCM/I2S 模式。 0: I2S 模式； 1: PCM 模式。

13.6.2 SIO_INTR_STATUS

- 偏移地址：0x042
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:2]	Reserved	保留。



比特	名称	描述
[1]	rx_intr	接收中断。1：发生了接收中断。
[0]	tx_intr	发送中断。1：发生了发送中断。

13.6.3 SIO_INTR_CLR

中断清除寄存器 SIO_INTR_CLR 向相应位写入 1 可清除对应位；写 0 时对相应位没有影响。该寄存器可以按位清除。

- 偏移地址：0x044
- 操作类型：W
- 复位方式：h/s

比特	名称	描述
[15:2]	Reserved	保留。
[1]	rx_intr	1：清除接收中断。
[0]	tx_intr	1：清除发送中断。

13.6.4 发送数据寄存器

发送数据寄存器由 3 个寄存器组成。

- 偏移地址：
 - I2S_LEFT_XD: 0x046
 - I2S_RIGHT_XD: 0x048
 - PCM_XD: 0x048
- 操作类型：W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:0]	I2S_LEFT_XD	I ² S 左声道发送数据寄存器。
[15:0]	I2S_RIGHT_XD	I ² S 右声道发送数据寄存器。
[15:0]	PCM_XD	PCM 发送数据寄存器。



13.6.5 接收数据寄存器

接收数据寄存器由 3 个寄存器组成。

- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

偏移地址	比特	名称	描述
0x04A	[15:0]	I2S_LEFT_RD	I ² S 左声道接收数据寄存器。
0x04C	[15:0]	I2S_RIGHT_RD	I ² S 右声道接收数据寄存器。
0x04C	[15:0]	PCM_RD	PCM 接收数据寄存器。

13.6.6 I2S_CT_SET

为了能够方便的对 I²S 控制寄存器进行位操作，在 SIO 中，为 I²S 控制寄存器设置了 2 个地址：

- 0x4E，设置寄存器地址。
当向寄存器中相应位写入 1 时，对应位被设为 1，写 0 无效。
- 0x50，清除寄存器地址。
当向寄存器中相应位写入 1 时，对应位被清除，写 0 无效。

在配置 SIO 的时钟及同步信号选择时需要注意，如果选择由外部输入（即设置 tx_clk_sel 和 tx_ws_sel 为 1），由于管脚复用的原因，需要按照表 13-4 所示进行配置。

表13-4 配置寄存器表

配置寄存器名称	需提前配置内容
SIO0（基地址 0x80080000）	同时配置 GPOUT7[3]为输出 1
SIO1（基地址 0x90020000）	同时配置 GPOUT7[2]为输出 1

- 偏移地址：0x04E、0x050
- 操作类型：R/W
- 复位值：0x800C
- 复位方式：h/s

比特	名称	描述
[15]	rst_n	通道复位，低电平有效。



比特	名称	描述
[14]	dsp_mode	I ² S 工作模式选择。 0: I ² S 模式; 1: DSP 模式。 在应用中, 应固定设置为 0。
[13]	rx_enable	接收通道使能。 0: 接收通道禁止; 1: 接收通道使能。
[12]	tx_enable	发送通道使能。 0: 发送通道禁止; 1: 发送通道使能
[11]	rx_fifo_disable	接收 FIFO Disable。 0: 接收 FIFO 使能; 1: 接收 FIFO 禁止。
[10]	tx_fifo_disable	发送 FIFO Disable。 0: 发送 FIFO 使能; 1: 发送 FIFO 禁止。
[9:7]	rx_fifo_threshold	接收 FIFO 阈值, 实际值等于设置值加 1。
[6:4]	tx_fifo_threshold	发送 FIFO 阈值, 实际值等于设置值加 1。
[3]	tx_clk_sel	发送时钟选择。 0: 选择内部产生; 1: 选择外部输入。
[2]	tx_ws_sel	发送同步信号选择。 0: 选择内部产生; 1: 选择外部输入。
[1]	rx_mode	接收信号模式。 0: 8 位模式; 1: 16 位模式。
[0]	tx_mode	发送信号模式。 0: 8 位模式; 1: 16 位模式。



13.6.7 SIO_ICD

分频寄存器是一个 16 位可读写的寄存器，用它来设置 SIO 模块的时钟分频系数，其分频值为设置值加 1。

设置值的计算公式为： $ICD = \text{总线频率} / (32 \times \text{音频采样频率}) - 1$ 。

说明

总线频率指芯片工作频率。

- 偏移地址：0x052
- 操作类型：R/W
- 复位值：0x1
- 复位方式：h/s

比特	名称	描述
[15:0]	sio_icd	SIO 时钟分频寄存器。

13.6.8 SIO_RX_STA

- 偏移地址：0x054
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:10]	Reserved	保留，读出时为 0。
[9:6]	rx_left_depth[3:0]	左声道接收 FIFO 深度指示。
[5:2]	rx_right_depth[3:0]	右声道接收 FIFO 深度指示。
[1]	rx_left_fifo_over	左声道接收 FIFO 上溢出指示。
[0]	rx_right_fifo_over	右声道接收 FIFO 上溢出指示。

13.6.9 SIO_TX_STA

- 偏移地址：0x056
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s



比特	名称	描述
[15:10]	Reserved	保留，读出时为 0。
[9:6]	tx_left_depth	左声道发送 FIFO 深度指示。
[5:2]	tx_right_depth	右声道发送 FIFO 深度指示。
[1]	tx_left_fifo_und	左声道发送 FIFO 下溢出指示。
[0]	tx_right_fifo_und	右声道发送 FIFO 下溢出指示。

13.6.10 SIO_PCM_CT_SET/ SIO_PCM_CT_CLR

为了能够方便的对 PCM 控制寄存器进行位操作，在 SIO 中，为 PCM 控制寄存器设置了 2 个地址：

- 0x58，设置寄存器地址。
当向寄存器中相应位写入 1 时，对应位被设为 1，写 0 无效。
- 0x5A，清除寄存器地址。
当向寄存器中相应位写入 1 时，对应位被清除，写 0 无效。

PCM 控制寄存器的读地址为 0x58。

- 偏移地址：0x058、0x05A
- 操作类型：R/W
- 复位值：0xC
- 复位方式：h/s

比特	名称	描述
[15]	tdm_mode	PCM 工作模式设置。 0: 正常模式； 1: TDM 模式。
[14]	edge_sel	发送时钟沿设置。 0: 下降沿发送； 1: 上升沿发送。 注意，此设置只是针对发送方向，在接收方向，芯片总是取与发送方向相反的取值。
[13]	rx_enable	接收通道使能。 0: 接收通道禁止； 1: 接收通道使能。



比特	名称	描述
[12]	tx_enable	发送通道使能。 0: 发送通道禁止; 1: 发送通道使能。
[11]	rx_fifo_disable	接收 FIFO Disable。 0: 接收 FIFO 使能; 1: 接收 FIFO 禁止。
[10]	tx_fifo_disable	发送 FIFO Disable。 0: 发送 FIFO 使能; 1: 发送 FIFO 禁止。
[9:7]	rx_fifo_threshold	接收 FIFO 阈值, 实际值等于设置值加 1。
[6:4]	tx_fifo_threshold	发送 FIFO 阈值, 实际值等于设置值加 1。
[3]	tx_clk_sel	发送时钟选择。 0: 选择内部产生; 1: 选择外部输入。
[2]	tx_ws_sel	发送同步信号选择。 0: 选择内部产生; 1: 选择外部输入。
[1]	rx_mode	接收信号模式。 0: 8 位模式; 1: 16 位模式。
[0]	tx_mode	发送信号模式。 0: 8 位模式; 1: 16 位模式。



14 通用目的输入输出接口 (GPIO)

关于本章

本章描述内容如下表所示。

标题	内容
14.1 概述	概括介绍 GPIO。
14.2 特点	概括介绍 GPIO 的特点。
14.3 信号描述	描述 GPIO 单元的输入输出管脚信号。
14.4 工作方式	描述 GPIO 的复用关系，并举例说明 GPIO 的应用。
14.5 寄存器概览	概括介绍 GPIO 的寄存器。
14.6 寄存器描述	详细描述 GPIO 的寄存器。
14.7 GPIO 配置示例	给出 GPIO 配置示例。



14.1 概述

通用目的输入输出接口 (GPIO) 用于生成或采集特定应用的输入或输出信号。GPIO 单元支持 8 组 GPIO, 每组 GPIO 提供 8 个可编程的输入输出引脚。Hi3510 包含 29 个独立 GPIO 管脚名, 其他 GPIO 管脚以复用管脚名出现在 Hi3510 管脚中, 或者保留。下面对 GPIO 管脚有详细描述。GPIO 单元的管脚主要有以下三类:

- 专用管脚
- 与其他管脚复用
- 在芯片内部作为专门用途

14.2 特点

GPIO 单元有以下特点:

- 当 GPIO 作为输入管脚时, 可作为中断源
- 当 GPIO 作为输出管脚时, 每个 GPIO 可以独立的清除或设置
- 除个别 GPIO 管脚外, 每个 GPIO 管脚可以配置为输入或者输出

14.3 信号描述

本节描述 GPIO 单元的输入输出管脚信号, 如表 14-1 所示。GPIO 管脚的复用请参见“14.4.1 复用说明”, 其他管脚复用将在相关章节描述。

表14-1 GPIO 接口信号描述

信号名	方向	描述
INTRN	I	全局中断。
GPIO0[7:0]、GPIO1[7:0]	I/O	通用目的输入输出接口。
GPIO2[7:0]、GPIO3[7:0]、 GPIO4[7:0]、GPIO5[7:0]、GPIO6[0]	I/O	外部复用管脚。
GPIO6[7:1]	-	保留。
GPIO7[5:0]	O	用作复用控制。
GPIO7[7:6]	-	保留。



14.4 工作方式

本节描述了 GPIO 的复用关系，并以实现键盘的 IO 接入系统为例，描述了 GPIO 的应用。

14.4.1 复用说明

GPIO 与 Trace、音频、VOU、SDRAM 和静态 RAM 总线存在复用关系，GPIO 的复用描述如表 14-2 所示。

说明

GPIO7[2]用作复用控制信号，控制 LCD 输出和第 2 个 SIO 的管脚的复用。关于 LCD 与 SIO 的复用关系，请参见表 12-2。

表14-2 GPIO 复用描述

GPIO 信号	描述
GPIO2[7:0]	GPIO2[7:0]与 Trace 数据输出 tracepkt[7:0]复用。 <ul style="list-style-type: none"> • GPIO2[0]可以复用作第 1 个音频接口同步信号 SIORFS0 • GPIO2[1]可以复用作 DSP 时钟输出 ZSPCLK，用于观测 DSP 时钟的输出。
GPIO3[2:0]	GPIO3[2:0]与 Trace 流水状态输出 pipestat[2:0]复用。
GPIO3[3]	GPIO3[3]与 Trace 同步输出 (tracesync) 复用。
GPIO3[4]	GPIO3[4]与 Trace 时钟输出 (traceclk) 复用，同时可以复用作 VOU 的随路时钟 nVOCLK (当 VOU 时钟采用片外晶振产生时，使用此时钟作为对端 VDAC 器件的时钟输入)。
GPIO3[5:3] GPIO4[7:0] GPIO5[4:0]	GPIO3[5:3]、GPIO4[7:0]、GPIO5[4:0]与 SDRAM 和静态 RAM 数据总线 EBIDQ[31:16]复用。
GPIO5[7:5] GPIO6[0]	GPIO5[7:5]、GPIO6[0]与 VOU 的输出像素值 LCDP[23:20]复用。 说明： GPIO5[7:5]、GPIO6[0]，只有 GPOUT 功能，即只能做为输出 IO 用。



表14-3 GPIO 复用对照表

位置	复用信号			描述
	1	2	3	
A10	GPIO 3[4]	traceclk	nVOCLK	当 GPIO7[5]=0, 作为 GPIO3[4]; 当 GPIO7[5]=1, 且 pTRACESEL=0, 作为 nVOCLK; 说明: 复用为 nVOCLK 需要配置系统控制器寄存器 SC_PERCTRL1[5]为 1。 当 GPIO7[5]=1, 且 pTRACESEL=1, 作为 Trace 时钟输出 (traceclk)。
C10	GPIO 3[3]	tracesync	-	当 pTRACESEL=0, 作为 GPIO3[3]; 当 pTRACESEL=1, 作为 Trace 同步输出 (tracesync)。
B10	GPIO 3[2]	pipestat[2]	-	当 pTRACESEL=0, 作为 GPIO3[2]; 当 pTRACESEL=1, 作为 Trace 流水状态输出 (pipestat[2])。
D11	GPIO 3[1]	pipestat[1]	-	当 pTRACESEL=0, 作为 GPIO3[1]; 当 pTRACESEL=1, 作为 Trace 流水状态输出 (pipestat[1])。
A9	GPIO 3[0]	pipestat[0]	-	当 pTRACESEL=0, 作为 GPIO3[0]; 当 pTRACESEL=1, 作为 Trace 流水状态输出 (pipestat[0])。
B9	GPIO 2[7]	tracepkt[7]	-	当 pTRACESEL=0, 作为 GPIO2[7]; 当 pTRACESEL=1, 作为 Trace 数据输出 (tracepkt[7])。
C9	GPIO 2[6]	tracepkt[6]	-	当 pTRACESEL=0, 作为 GPIO2[6]; 当 pTRACESEL=1, 作为 Trace 数据输出 (tracepkt[6])。
D10	GPIO 2[5]	tracepkt[5]	-	当 pTRACESEL=0, 作为 GPIO2[5]; 当 pTRACESEL=1, 作为 Trace 数据输出 (tracepkt[5])。
D9	GPIO 2[4]	tracepkt[4]	-	当 pTRACESEL=0, 作为 GPIO2[4]; 当 pTRACESEL=1, 作为 Trace 数据输出 (tracepkt[4])。
A8	GPIO 2[3]	tracepkt[3]	-	当 pTRACESEL=0, 作为 GPIO2[3]; 当 pTRACESEL=1, 作为 Trace 数据输出 (tracepkt[3])。



位置	复用信号			描述
	1	2	3	
B8	GPIO 2[2]	tracepkt [2]	-	当 pTRACESEL=0, 作为 GPIO2[2]; 当 pTRACESEL=1, 作为 Trace 数据输出 (tracepkt[2])。
C8	GPIO 2[1]	tracepkt [1]	ZSPCLK	当 GPIO7[4]=0, 作为 GPIO2[1]; 当 GPIO7[4]=1, 且 pTRACESEL=0, 作为 DSP 时钟输出 (ZSPCLK); 当 GPIO7[4]=1, 且 pTRACESEL=1, 作为 Trace 数据输出 (tracepkt[1])。
D8	GPIO 2[0]	Tracepkt [0]	SIORFS0	当 GPIO7[3]=0, 作为 GPIO2[0]; 当 GPIO7[3]=1, 且 pTRACESEL=0, 作为 SIORFS0; 当 GPIO7[3]=1, 且 pTRACESEL=1, 作为 tracepkt[0]。
J2	LCD P23	GPIO6[0]	-	输出像素值, LCD_R7 输出。 当 GPIO7[1]=0, 作为 LCDP23; 当 GPIO7[1]=1, 作为 GPOUT6[0]。 说明 这两种功能需要系统控制器寄存器 SC_PERCTRL1[LcdpOen]=1。
J3	LCD P22	GPIO5[7]	-	输出像素值, LCD_R6 输出。 当 GPIO7[1]=0, 作为 LCDP22; 当 GPIO7[1]=1, 作为 GPOUT5[7]。 说明 这两种功能需要系统控制器寄存器 SC_PERCTRL1[LcdpOen]=1。
K5	LCD P21	GPIO5[6]	-	输出像素值, LCD_R5 输出。 当 GPIO7[1]=0, 作为 LCDP21; 当 GPIO7[1]=1, 作为 GPOUT5[6]。 说明 这两种功能需要系统控制器寄存器 SC_PERCTRL1[LcdpOen]=1。
K1	LCD P20	GPIO5[5]	-	输出像素值, LCD_R4 输出。 当 GPIO7[1]=0, 作为 LCDP20; 当 GPIO7[1]=1, 作为 GPOUT5[5]。 说明 这两种功能需要系统控制器寄存器 SC_PERCTRL1[LcdpOen]=1。



位置	复用信号			描述
	1	2	3	
W14	EBID Q31	GPIO5[4]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ31; 当 GPIO7[0]=1, 作为 GPIO5[4]。
W13	EBID Q30	GPIO5[3]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ30; 当 GPIO7[0]=1, 作为 GPIO5[3]。
AA13	EBID Q29	GPIO5[2]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ29; 当 GPIO7[0]=1, 作为 GPIO5[2]。
AB14	EBID Q28	GPIO5[1]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ28; 当 GPIO7[0]=1, 作为 GPIO5[1]。
W12	EBID Q27	GPIO5[0]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ27; 当 GPIO7[0]=1, 作为 GPIO5[0]。
Y12	EBID Q26	GPIO4[7]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ26; 当 GPIO7[0]=1, 作为 GPIO4[7]。
AA12	EBID Q25	GPIO4[6]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ25; 当 GPIO7[0]=1, 作为 GPIO4[6]。
AB13	EBID Q24	GPIO4[5]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ24; 当 GPIO7[0]=1, 作为 GPIO4[5]。
Y11	EBID Q23	GPIO4[4]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ23; 当 GPIO7[0]=1, 作为 GPIO4[4]。
AA11	EBID Q22	GPIO4[3]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ22; 当 GPIO7[0]=1, 作为 GPIO4[3]。



位置	复用信号			描述
	1	2	3	
AB11	EBID Q21	GPIO4[2]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ21; 当 GPIO7[0]=1, 作为 GPIO4[2]。
AB10	EBID Q20	GPIO4[1]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ20; 当 GPIO7[0]=1, 作为 GPIO4[1]。
W10	EBID Q19	GPIO4[0]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ19; 当 GPIO7[0]=1, 作为 GPIO4[0]。
Y10	EBID Q18	GPIO3[7]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ18; 当 GPIO7[0]=1, 作为 GPIO3[7]。
AA10	EBID Q17	GPIO3[6]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ17; 当 GPIO7[0]=1, 作为 GPIO3[6]。
AB9	EBID Q16	GPIO3[5]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0]=0, 作为 EBIDQ16; 当 GPIO7[0]=1, 作为 GPIO3[5]。

14.4.2 使用指南

本节主要介绍 GPIO 实现键盘的 IO 接入系统的实现方式，键盘扫描阵列如图 14-1 所示。

本例中键盘采用 GPIO1[2:0] 3 个引脚和 GPIO1[6:3] 4 个引脚分别作为键盘接入的行和列，GPIO1[6:3]列电平拉高。

GPIO 的中断触发方式可以配置为边沿触发和电平触发，并且可以选择上升沿触发还是下降沿触发，高电平触发还是低电平触发。键盘将 GPIO1[6:3]中断触发方式设为双沿触发，即当按键按下和按键弹起时，都有中断发生。

键盘扫描策略

GPIO1[2:0]的引脚作为行选择，输出，初始化为低电平；GPIO1[6:3]引脚作为列选择，输入，初始化为高电平。

当按键被按下时，行选择和列选择连通。确定按键的方法如下：

- 确定按键的列



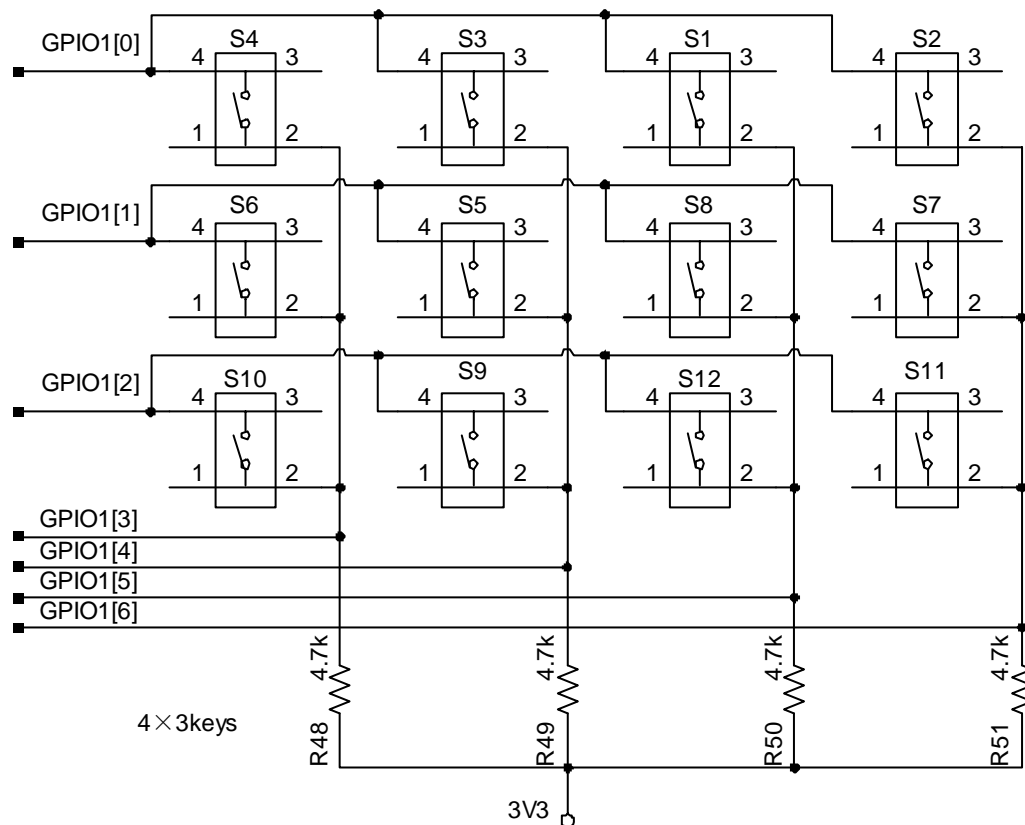
当 GPIO1[6:3] 中的 1 个引脚由高电平变为低电平后会触发中断，可以确定 GPIO1[6:3] 中的某一引脚连接的按键被按下，则可确定按键的列，即与电平变低引脚连接的列。

- 确定按键的行

GPIO1[2:0] 的 3 个引脚轮流输出低电平，若扫描到已确定按键的列为低电平则可确定按键的行，即与输出低电平的引脚连接的行。

按键的行和列的确定就可以唯一地确定 1 个按键。

图14-1 键盘扫描阵列示意图



去抖动

当按键按下时，电平不会平滑地由高变低。在电平变低的过程中会出现毛刺和抖动，然后才变为低电平。当按键弹起时，也同样会出现抖动。

如果在抖动的过程中读数，会有以下影响：

- 造成读数的不准确。
- 频繁的触发中断。

因此，软件采用延时一段时间的方法进行去抖动。在延时的时间内，电平抖动引起的所有中断将被屏蔽，延时时间结束后，再进行读数和其它操作。



按键时长计算

在实际的应用中，通过键盘扫描获得按键的坐标后往往需要获得按键时间的长度。当按键中断触发时开始计数，按键弹起中断触发时计数结束，两者之间的长度为按键时间的长度。

14.5 寄存器概览

GPIO 单元有 8 组 GPIO 寄存器 (GPIO0~GPIO7)，每组 GPIO 寄存器的地址位宽 8 位，如表 14-4 所示。GPIO0~GPIO7 具有相同的寄存器组，如表 14-5 所示。GPIO_n 的某寄存器地址为：GPIO_n 基址+该寄存器偏移地址，其中 n 表示寄存器组 0~7。

表14-4 8 组 GPIO 寄存器的基址

寄存器名称	基址	寄存器名称	基址
GPIO0	0x101E4000	GPIO4	0x101F7000
GPIO1	0x101E5000	GPIO5	0x101F8000
GPIO2	0x101E6000	GPIO6	0x101F9000
GPIO3	0x101E7000	GPIO7	0x101FA000

表14-5 GPIO 寄存器一览表

偏移地址	寄存器名称	描述	页码
0x000~ 0x3FC	GPIO_DATA	数据寄存器	14-10
0x400	GPIO_DIR	数据方向寄存器	14-10
0x404	GPIO_IS	中断触发 (interrupt sense) 寄存器	14-10
0x408	GPIO_IBE	双沿触发中断 (interrupt both edges) 寄存器	14-11
0x40C	GPIO_IEV	触发中断条件 (interrupt event) 寄存器	14-11
0x410	GPIO_IE	中断屏蔽寄存器	14-12
0x414	GPIO_RIS	原始中断状态寄存器	14-12
0x418	GPIO_MIS	屏蔽中断状态寄存器	14-12
0x41C	GPIO_IC	中断清除寄存器	14-13
0x420	GPIO_AFSEL	模式控制器选择寄存器	14-13



14.6 寄存器描述

本节详细描述了 GPIO 单元寄存器。

14.6.1 GPIO_DATA

GPIO_DATA 为数据寄存器，可以通过地址控制对该寄存器的位进行操作。

Data[7:0]分别对应地址位[9:2]，当地址位的某位为高时，则可以对相应的 Data 位进行读写操作；反之，若地址位的某位为低则不能进行有效操作。

例如：

若地址为 0x3FC（二进制：11_1111_1100），则对 Data[7:0]操作全部有效。

若地址为 0x200（二进制：10_0000_0000），则仅能对 Data[7]进行有效操作。

- 偏移地址：0x000~0x3FC
- 操作方式：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[7:0]	Data register	输入数据；输出数据。

14.6.2 GPIO_DIR

GPIO_DIR 为数据方向控制寄存器。

- 偏移地址：0x400
- 操作方式：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[7:0]	Data direct register	每 bit 各对应一个 GPIO 管脚。 0：输入； 1：输出。

14.6.3 GPIO_IS

GPIO_IS 为中断触发方式寄存器，可以选择边沿触发中断或电平触发中断。

- 偏移地址：0x404
- 操作类型：R/W



- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[7:0]	Interrupt sense register	每个 bit 各对应一个 GPIO 管脚: 0: 边沿触发中断; 1: 电平触发中断。 说明 系统在 Sleep 模式下, 如果正常使用 GPIO 中断必须使用电平触发方式。

14.6.4 GPIO_IBE

GPIO_IBE 可以选择单沿触发方式或双沿触发方式。

- 偏移地址: 0x408
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[7:0]	Interrupt sense register	每个 bit 各对应一个 GPIO 管脚: 0: 单沿触发。上升沿触发还是下降沿触发由寄存器 GPIO_IEV 决定; 1: 上升沿或下降沿都可以触发中断。

14.6.5 GPIO_IEV

GPIO_IEV 在单沿触发时可以选择上升沿或下降沿触发中断, 在电平触发时可以选择低电平或高电平触发。

- 偏移地址: 0x40C
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[7:0]	Interrupt event register	每个 bit 各对应一个 GPIO 管脚: 0: 下降沿或低电平触发中断; 1: 上升沿或高电平触发中断。



14.6.6 GPIO_IE

GPIO_IE 为中断屏蔽寄存器。

- 偏移地址: 0x410
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[7:0]	Interrupt mask register	每个 bit 各对应一个 GPIO 管脚: 0: 不使能相应管脚的中断; 1: 使能相应管脚的中断。

14.6.7 GPIO_RIS

GPIO_RIS 为原始中断状态寄存器。

- 偏移地址: 0x414
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[7:0]	Raw interrupt status	每个 bit 各对应一个 GPIO 管脚: 0: 中断触发条件没有匹配; 1: 中断触发条件已匹配。

14.6.8 GPIO_MIS

GPIO_MIS 为屏蔽后中断状态寄存器。

- 偏移地址: 0x418
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[7:0]	Masked interrupt status	每个 bit 各对应一个 GPIO 管脚: 0: GPIO 管脚没有触发中断; 1: GPIO 管脚已触发中断。



14.6.9 GPIO_IC

GPIO_IC 为中断清除寄存器。

- 偏移地址: 0x41C
- 操作类型: W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[7:0]	Interrupt clear status	每个 bit 各对应一个 GPIO 管脚: 0: 没有影响; 1: 清中断。

14.6.10 GPIO_AFSEL

GPIO_AFSEL 寄存器为硬件或软件方式选择寄存器。

- 偏移地址: 0x420
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[7:0]	Mode control select register	每个 bit 各对应一个 GPIO 管脚: 0: 使能软件控制模式; 1: 使能硬件控制模式。 禁止配置成硬件控制模式。

14.7 GPIO 配置示例

14.7.1 注意事项

配置时需要注意的事项是:

- GPIO5[7:5]和 GPIO6[0]只能实现输出功能, 输入功能被屏蔽;
- 实现 GPIO5[7:5]和 GPIO6[0]的 GPIO 功能及 LCDP 功能都需要先配置系统控制器寄存器 SC_PERCTRL1[3]为 1。
- 实现 GPIO3[4]的 nVOCLK 功能需要先配置系统控制器寄存器 SC_PERCTRL1[5]为 1, 实现 GPIO3[4]的 GPIO 功能无此要求。



14.7.2 配置过程

本节以设置 GPIO2[1]实现 GPIO 输入功能为例，说明 GPIO 配置过程。

说明

示例中 GPIO 寄存器地址以偏移地址表示。

- 配置复用控制信号 GPIO7[4]输出 0。

具体配置过程为：

- 向 GPIO7 寄存器 GPIO_DIR 写 0xFF，GPIO7 所有管脚为输出。
- 向 GPIO7 地址 0x040 写 0x00，即配置 GPIO7[4]输出 0。

地址位[9:2]表示 GPIO7 的位选，1 表示选择，0 表示不选。对于地址 0x40，其第 6 位为 1，表示 GPIO7[(6-2)=4]位被选中，写的值 0x00 只有第 4 位写到 GPIO7 的 GPIO_DATA 的寄存器中，其他位为无用值。

- 配置 GPIO2[1]为输入功能。

具体配置过程为：

- 配置 GPIO2 的寄存器 GPIO_DIR 为原 GPIO_DIR[7:0]与 0xFD 相与的结果，表示 GPIO2[1]为输入；
- 配置 GPIO_IS，选择边沿触发还是电平触发；
- 配置 GPIO_IBE，选择单沿触发还是双沿触发；
- 配置 GPIO_IEV，选择上升沿触发还是下降沿触发；
- 配置 GPIO_IE 为原 GPIO_IE 与 0x2 相与的结果，表示打开 GPIO2[1]中断使能；

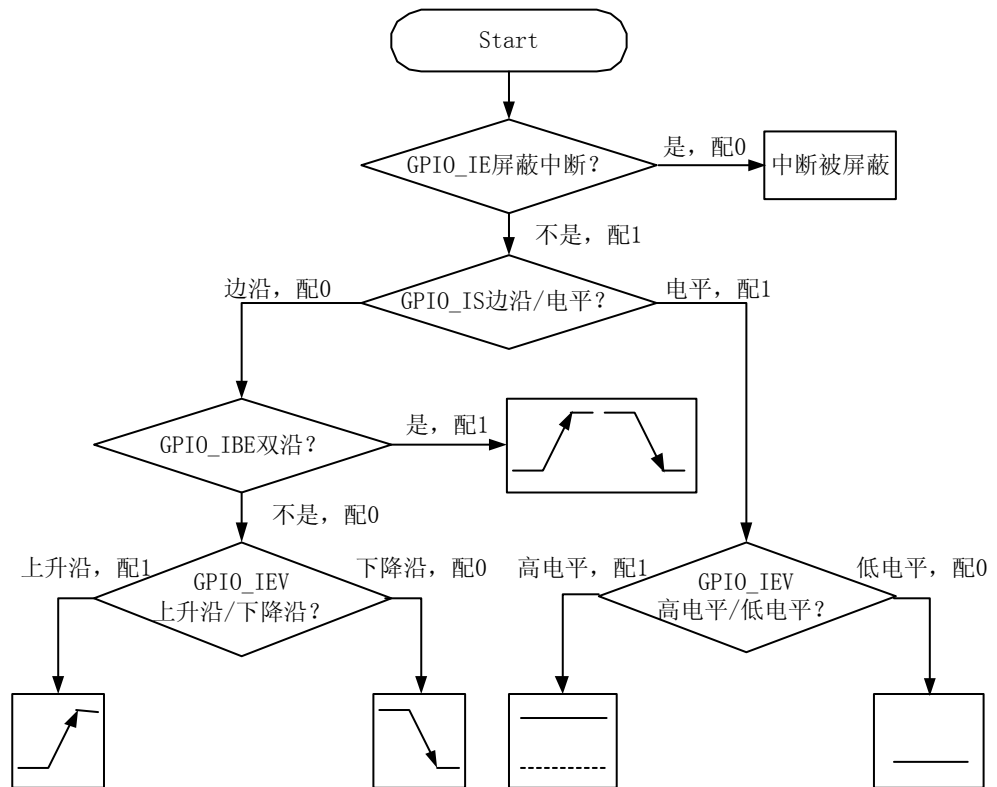
如果中断到来，读 GPIO_RIS&0x2，检查是否为 GPIO2[1]中断，如果是则写 GPIO_IC 为 0x2 清中断，然后配置 GPIO_IE 为原 GPIO_IE 与 0xFD 相与的结果，屏蔽中断，读地址 0x8，其数据即是 GPIO2[1]的值。

14.7.3 GPIO 中断配置树形图

图 14-2 给出 GPIO 中断选项配置的树形图，形象地展示 GPIO 中断配置过程，可以作为参考。



图14-2 GPIO 中断配置树形图





15 通用异步收发器 (UART)

关于本章

本章描述内容如下表所示。

标题	内容
15.1 概述	概括介绍 UART。
15.2 特点	概括介绍 UART 的特点。
15.3 信号描述	描述 UART 单元的输入输出管脚信号。
15.4 工作方式	描述 UART 的帧格式。
15.5 寄存器概览	概括介绍 UART 的寄存器。
15.6 寄存器描述	详细描述 UART 的寄存器。



15.1 概述

UART (Universal Asynchronous Receiver Transmitter) 将来自外围设备的数据进行串并转换, 将输出到外围设备的数据进行并串转换。UART 单元提供了 2 个标准的 UART, 即 UART0 和 UART1。UART1 支持用户高速数据传送和程序下载。

15.2 特点

UART 有以下特点:

- 支持 16×8bit 的 Tx 和 16×12bit 的 Rx FIFO
- 可编程数据位和停止位。数据位可通过编程设定为 5 位、6 位、7 位或 8 位, 停止位可通过编程设定为 1 位或 2 位
- 支持奇、偶校验方式或者无校验
- 可编程传送速率, 最大支持 460.8kbit/s 的速率
- 支持 Modem 状态中断、接收 FIFO 中断、发送 FIFO 中断、接收超时中断、错误中断和一个组合中断。Hi3510 使用组合中断连接到 VIC
- 支持初始中断状态查询和屏蔽后中断状态查询
- 支持通过编程禁止 UART 模块或者 UART TX/UART RX, 以降低功耗
- 由系统控制器发出 UART 时钟信号, 当需要关断 UART 模块时, 通过 Clock Gating 不发出 UART 时钟信号

15.3 信号描述

本节描述了 UART 单元的输入输出管脚信号, 如表 15-1 所示。

表15-1 UART 接口信号描述

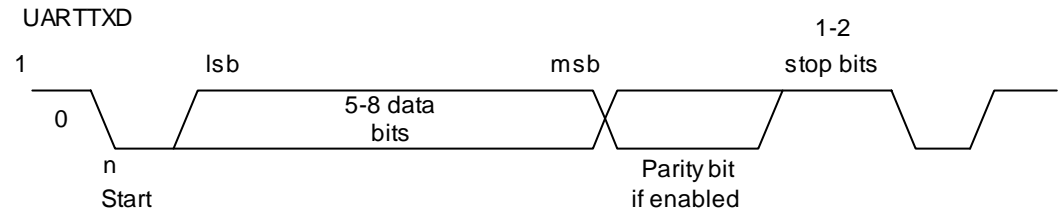
信号名	方向	描述
URXD0	I	UART0 数据接收
UTXD0	O	UART0 数据发送
URXD1	I	UART1 数据接收
UTXD1	O	UART1 数据发送
UCTS1	I	UART1 清除发送
URTS1	O	UART1 请求发送



15.4 工作方式

本节描述了 UART 的帧格式，如图 15-1 所示。

图15-1 UART 帧格式



15.5 寄存器概览

UART 提供 UART0 和 UART1 两个串口。

- UART0 寄存器的地址位宽 16 位，地址范围 0x101F_1000~0x101F_1FFF。
- UART1 寄存器的地址位宽 16 位，地址范围 0x101F_2000~0x101F_2FFF。

UART0 和 UART1 寄存器只是基址不同，因此不对二者做分别介绍。UART 寄存器概览如表 15-2 所示。

表15-2 UART 寄存器概览

偏移地址	名称	描述	页码
0x000	UART_DR	数据寄存器	15-4
0x004	UART_RSR/UART_ECR	接收状态寄存器/错误清除寄存器	15-4
0x018	UART_FR	标签寄存器	15-5
0x020	UART_LPR	保留	15-6
0x024	UART_IBRD	整数波特率寄存器	15-7
0x028	UART_FBRD	小数波特率寄存器	15-7
0x02C	UART_LCR_H	线控 (line control) 寄存器	15-8
0x030	UART_CR	控制寄存器	15-9
0x034	UART_IFLS	中断 FIFO 水线选择寄存器	15-10
0x038	UART_IMSC	中断屏蔽的设置/清除选择寄存器	15-11
0x03C	UART_RIS	屏蔽前中断状态寄存器	15-12
0x040	UART_MIS	屏蔽后中断状态寄存器	15-13



偏移地址	名称	描述	页码
0x044	UART_ICR	中断清除寄存器	15-14
0x048	UART_DMACR	DMA 控制寄存器	15-15

15.6 寄存器描述

UART 有两组寄存器，这两组寄存器的功能相同，本节以其中一组为例描述了 UART 寄存器。

15.6.1 UART_DR

- 偏移地址：0x000
- 复位值：-
- 复位方式：h/s

比特	名称	操作类型	描述
[15:12]	Reserved	R	保留。
[11]	OE	R	0: 接收 FIFO 有空位，则该位被清 0; 1: 接收 FIFO 满且已接收一个数据。
[10]	BE	R	0: 当写 UART_ECR 寄存器后，该位清 0。 1: 检测到 break 条件。即接收数据的输入保持低的时间比一个全字传输（包括 start、data、parity、stop bit）还要长。
[9]	PE	R	该位置 1 表示接收数据的校验结果和 UART_LCR_H 寄存器中的设置不匹配。 FIFO 模式下，该错误与 FIFO 顶部的 character 相关联。
[8]	FE	R	该位置 1 表示接收到的 character 的停止位错误（有效的停止位为 1）。 FIFO 模式下，该错误与 FIFO 顶部的 character 相关联。
[7:0]	DATA	R/W	接收数据和发送数据。

15.6.2 UART_RSR/UART_ECR

- 偏移地址：0x004



- 复位值: -
- 复位方式: h/s

比特	名称	操作类型	描述
[7:0]	UART_RS R/UART_E CR	W	对该寄存器写操作时, 该寄存器作错误清除寄存器 (UART_ECR)。写一次, 清掉 framing、parity、break 和 overrun 等错误。
[7:4]	Reserved	R	保留。
[3]	OE	R	溢出错误。 0: 向 UART_ECR (即此寄存器) 写可清状态。 1: 接收 FIFO 满且接收了一个数据。 当 FIFO 满时, FIFO 中的内容保持有效。因为不会有下一个数据写到 FIFO 中, 但是移位寄存器会溢出。CPU 必须立刻读数据以清空 FIFO。
[2]	BE	R	break error。 0: 向 UART_ECR (即此寄存器) 写可清状态。 1: break 条件被检测到。 Break 的条件: 接收数据的输入保持低的时间比一个全字传输 (定义了 start、data、parity、stop bit) 还要长。
[1]	PE	R	校验错误。 0: 向 UART_ECR (即此寄存器) 写可清状态。 1: 接收数据的校验结果和 UART_LCR_H 寄存器中的设置不匹配。 FIFO 模式下, 该错误与 FIFO 顶部的 character 相关联。
[0]	FE	R	Framing error。 0: 向 UART_ECR (即此寄存器) 写可清状态。 1: 接收到的 character 的停止位错误 (有效的停止位为 1)。

15.6.3 UART_FR

- 偏移地址: 0x018
- 操作类型: R
- 复位值: 0x00_90
- 复位方式: h/s



比特	名称	描述
[15:9]	Reserved	保留。
[8]	RI	Ring indicator, 是 UART ring indicator (nUARTRI) 的取反。即 modem 状态的输入为 0, 该位为 1。
[7]	TXFE	该位决定于 UART_LCR_H[FEN]位的状态。 <ul style="list-style-type: none"> • 如果 FIFO 不使能, 当发送寄存器空时, 该位置 1。 • 如果 FIFO 使能, 当发送 FIFO 为空时, 该位置 1。
[6]	RXFF	该位决定于 UART_LCR_H[FEN]位的状态。 <ul style="list-style-type: none"> • 如果 FIFO 不使能, 当接收寄存器满时, 该位置 1。 • 如果 FIFO 使能, 当接收 FIFO 为满时, 该位置 1。
[5]	TXFF	该位决定于 UART_LCR_H[FEN]位的状态。 <ul style="list-style-type: none"> • 如果 FIFO 不使能, 当发送寄存器满时, 该位置 1。 • 如果 FIFO 使能, 当发送 FIFO 为满时, 该位置 1。
[4]	RXFE	该位决定于 UART_LCR_H[FEN]位的状态。 <ul style="list-style-type: none"> • 如果 FIFO 不使能, 当接收寄存器空时, 该位置 1。 • 如果 FIFO 使能, 当接收 FIFO 为空时该位置 1。
[3]	BUSY	1: 表示 UART 正忙于发送数据。 该位一直保持到整个字节 (包括所有的停止位) 完全从移位寄存器中发送出去。 一旦发送 FIFO 非空, 该位就置位 (不管 UART 使能与否)。
[2]	DCD	该位是 UART data carrier detect (nUARTDCD) 的取反。即 modem 状态的输入为 0, 该位为 1。
[1]	DSR	该位是 UART data set ready (nUARTDSR) 的取反。即 modem 状态的输入为 0, 该位为 1。
[0]	CTS	该位是 UART clear to send (nUART CTS) 的取反。即 modem 状态的输入为 0, 该位为 1。

15.6.4 UART_LPR

- 偏移地址: 0x020
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[7:0]	Reserved	该寄存器保留不用。



15.6.5 UART_IBRD

- 偏移地址：0x024
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:0]	BAUD DIVINT	波特率分频值整数部分。复位时全为 0。

波特率分频计算公式：

$$\text{Baud Rate Divisor} = \text{UARTCLK} / (16 \times \text{Baud Rate}) = \text{BRD}_I + \text{BRD}_F$$

- BRD_I 为波特率分频值整数部分
- BRD_F 为波特率分频值小数部分

寄存器配置值的计算方法为： $m = \text{integer}(\text{BRD}_F \times 2^6 + 0.5)$

Hi3510 中的 UART 工作时钟不是使用业界标准的时钟模式，因此要产生 460.8K/230.4K/115.2K 等波特率时，必须使用分数分频，而分数分频会造成误差。典型的波特率及其误差如表 15-3 所示。

表15-3 典型的波特率及误差对应关系表

配置的波特率分频值整数部分	配置的波特率分频值小数部分	需要的波特率	实际产生的波特率	误差
0x1	0x5	230400	231911	0.656
0x2	0xB	115200	115101	0.086
0x3	0x10	76800	76923	0.160
0x6	0x21	38400	38369	0.081
0x11	0x17	14400	14401	0.007
0x68	0xB	2400	2400	~0
0x8E0	0x2F	110	110	~0

15.6.6 UART_FBRD

- 偏移地址：0x028
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

**注意**

- 整数波特率寄存器和小数波特率寄存器的值必须等到当前 character 发送和接收完毕才能更新。
- 最小的分频值为 1，最大的分频值为 $65536(2^{16}-1)$ 。也就是说 $UART_IBRD = 0$ 是无效的，而此时 $UART_FBRD$ 将被忽略。同样的，如果 $UART_IBRD = 65536$ ($0xFFFF$)， $UART_FBRD$ 就不能比 0 大，如果比 0 大，则会导致发送和接收的失败。

比特	名称	描述
[5:0]	BAUD DIVFRAC	波特率分频值小数部分。复位时全为 0。

15.6.7 UART_LCR_H

- 偏移地址：0x02C
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:8]	Reserved	保留，不能修改。
[7]	SPS	stick parity select。 当该寄存器的 bit1、bit2、bit7 被置位时，校验位就会作为 0 发送和检测。 当 bit1、bit7 被置位，bit2 为 0 时，校验位就会作为 1 发送和检测。 当 bit1、bit2、bit7 都被清零，那么 stick parity 不使能。
[6:5]	WLEN	指示发送和接收一个帧里数据位的数目。 00: 5 位; 10: 7 位; 01: 6 位; 11: 8 位。
[4]	FEN	0: 发送和接收 FIFO 不使能; 1: 发送和接收 FIFO 使能。
[3]	STP2	1: 发送的帧尾有 2 位停止位。 接收逻辑在接收时不检查这 2 位停止位。



比特	名称	描述
[2]	EPS	0: 在发送和接收过程中生成奇校验或检查奇校验; 1: 在发送和接收过程中生成偶校验或检查偶校验。 当 PEN 位为 0 时, 该位不起作用。
[1]	PEN	0: 不作校验; 1: 发送方向产生校验, 接收方向作校验检查。
[0]	BRK	send break。 如果将该位置 1, 则在完成当前 bit 的发送后, UARTTXD 连续输出低电平。 要正确的执行 break 命令, 软件将该位置 1 的时间必须超过 2 个完整帧。 在正常使用中, 该位必须清 0。

15.6.8 UART_CR

- 偏移地址: 0x030
- 操作类型: R/W
- 复位值: 0x03_00
- 复位方式: h/s

比特	名称	描述
[15]	CTSEn	1: 使能 CTS 硬件流控, 只有当 nUARTCTS 信号有效时才发送数据。
[14]	RTSEn	1: 使能 RTS 硬件流控, 只有当接收 FIFO 有空间时才请求接收数据。
[13]	Out2	该位为 UART modem 状态输出信号 nUARTOUT2 的取反。 即该位配置为 1, 输出信号为 0。 对 DTE (数据终端设备), 该位用于 RI (ring indicator) 信号。
[12]	Out1	该位为 UART modem 状态输出信号 nUARTOUT1 的取反。 即该位配置为 1, 输出信号为 0。 对 DTE (数据终端设备), 该位用于 DCD (数据载波检测) 信号。



比特	名称	描述
[11]	RTS	请求发送。 该位为 UART modem 状态输出信号 nUARTRTS 的取反。 即该位配置为 1, 输出信号为 0。 说明 当设定成 RTS 硬件流控时 (即 UART_CR[14]=1), 不能通过配置此比特的值而使其输出信号 nUARTRTS 的取反。
[10]	DTR	data transmit ready。 该位为 UART modem 状态输出信号 nUARTDTR 的取反。 即该位配置为 1, 输出信号为 0。
[9]	RXE	1: UART 的接收使能。 UART 或 SIR (取决于 SIREN 位的设置) 开始接收数据。 在接收的过程中如果 UART 被禁能, 那么当前 character 的接收就会在正常停止之前结束。
[8]	TXE	1: UART 的发送使能。 UART 或 SIR (取决于 SIREN 位的设置) 开始发送数据。 在发送的过程中如果 UART 被禁能, 那么当前 character 的发送就会在正常停止之前结束。
[7]	LBE	loop back enable。 如果该位置为 1, 同时 SIREN (bit1) 和 UARTRCR (内部控制测试寄存器: 偏移地址 0x80) 的 bit2 (SIR 测试使能位) 也置 1, 那么 nSIROUT 的输出直接环回到 SIRIN。 SIRTEST 必须置 1, 以覆盖 SIR 正常工作时的半双工模式, 环回测试结束后 SIRTEST 必须清零。 如果该位置为 1, SIRTEST 置为 0, 则 UARTRXD 输出环回到 UARTRXD。 不管是在 SIR 模式还是 UART 模式, 如果该位置 1, modem 的输出也环回到 modem 的输入。 复位时该位清零, 环回模式不使能。
[6:1]	Reserved	保留, 不能修改。
[0]	UARTEN	1: UART 使能。数据通过 UART 的信号或 SIR 的信号 (取决于 SIREN 位的设置) 进行发送和接收。 如果在发送和接收过程中将 UART 禁能, 则会在正常停止之前结束当前 character 的传送。

15.6.9 UART_IFLS

- 偏移地址: 0x034



- 操作类型: R/W
- 复位值: 0xC
- 复位方式: h/s

比特	名称	描述
[15:6]	Reserved	保留。
[5:3]	RXIFLSEL	接收中断 FIFO 的 level 选择, 接收中断的触发点如下。 000: 接收 FIFO \geq 1/8 full; 001: 接收 FIFO \geq 1/4 full; 010: 接收 FIFO \geq 1/2 full; 011: 接收 FIFO \geq 3/4 full; 100: 接收 FIFO \geq 7/8 full; 101~111: 保留。
[2:0]	TXIFLSEL	发送中断 FIFO 的 level 选择, 发送中断的触发点如下。 000: 发送 FIFO \leq 1/8full; 001: 发送 FIFO \leq 1/4full; 010: 发送 FIFO \leq 1/2full; 011: 发送 FIFO \leq 3/4full; 100: 发送 FIFO 开始 \leq 7/8full; 101~111: 保留。

15.6.10 UART_IMSC

- 偏移地址: 0x038
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[15:11]	Reserved	保留。
[10]	OEIM	溢出错误中断的屏蔽状态。 0: 使能中断; 1: 禁止中断。
[9]	BEIM	break 错误中断的屏蔽状态。 0: 使能中断; 1: 禁止中断。



比特	名称	描述
[8]	PEIM	校验中断的屏蔽状态。 0: 使能中断; 1: 禁止中断。
[7]	FEIM	帧错误中断的屏蔽状态。 0: 使能中断; 1: 禁止中断。
[6]	RTIM	接收超时中断的屏蔽状态。 0: 使能中断; 1: 禁止中断。
[5]	TXIM	发送中断的屏蔽状态。 0: 使能中断; 1: 禁止中断。
[4]	RXIM	接收中断的屏蔽状态。 0: 使能中断; 1: 禁止中断。
[3]	DSRMIM	nUARTDSR modem 中断的屏蔽状态。 0: 使能中断; 1: 禁止中断。
[2]	DCDMIM	nUARTDCD modem 中断的屏蔽状态。 0: 使能中断; 1: 禁止中断。
[1]	CTSMIM	nUARTCTS modem 中断的屏蔽状态。 0: 使能中断; 1: 禁止中断。
[0]	RIMIM	nUARTRI modem 中断的屏蔽状态。 0: 使能中断; 1: 禁止中断。

15.6.11 UART_RIS

- 偏移地址: 0x03C
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s



比特	名称	描述
[15:11]	Reserved	保留。
[10]	OERIS	屏蔽前溢出错误中断状态。
[9]	BERIS	屏蔽前 break 错误中断状态。
[8]	PERIS	屏蔽前校验中断状态。
[7]	FERIS	屏蔽前错误中断状态。
[6]	RTRIS	屏蔽前接收超时中断状态。
[5]	TXRIS	屏蔽前发送中断状态。
[4]	RXRIS	屏蔽前接收中断状态。
[3]	DSRMIS	屏蔽前 nUARTDSR modem 中断状态。
[2]	DCDMIS	屏蔽前 nUARTDCD modem 中断状态。
[1]	CTSMIS	屏蔽前 nUARTCTS modem 中断状态。
[0]	RIMIS	屏蔽前 nUARTRI modem 中断状态。

15.6.12 UART_MIS

- 偏移地址：0x040
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:11]	Reserved	保留。
[10]	OEMIS	屏蔽后的溢出错误中断状态。
[9]	BEMIS	屏蔽后的 break 错误中断状态。
[8]	PEMIS	屏蔽后的校验中断状态。
[7]	FEMIS	屏蔽后的错误中断状态。
[6]	RTMIS	屏蔽后的接收超时中断状态。
[5]	TXMIS	屏蔽后的发送中断状态。
[4]	RXMIS	屏蔽后的接收中断状态。
[3]	DSRMMIS	屏蔽后的 nUARTDSR modem 中断状态。
[2]	DCDMMIS	屏蔽后的 nUARTDCD modem 中断状态。



比特	名称	描述
[1]	CTSMMIS	屏蔽后的 nUARTCTS modem 中断状态。
[0]	RIMMIS	屏蔽后的 nUARTRI modem 中断状态。

15.6.13 UART_ICR

- 偏移地址: 0x044
- 操作类型: W
- 复位值: -
- 复位方式: h/s

比特	名称	描述
[15:11]	Reserved	保留。
[10]	OEIC	清除溢出错误中断。 0: 无效; 1: 清除中断。
[9]	BEIC	清除 break 错误中断。 0: 无效; 1: 清除中断。
[8]	PEIC	清除校验中断。 0: 无效; 1: 清除中断。
[7]	FEIC	清除错误中断。 0: 无效; 1: 清除中断。
[6]	RTIC	清除接收超时中断。 0: 无效; 1: 清除中断。
[5]	TXIC	清除发送中断。 0: 无效; 1: 清除中断。
[4]	RXIC	清除接收中断。 0: 无效; 1: 清除中断。



比特	名称	描述
[3]	DSRMIC	清除 nUARTDSR modem 中断。 0: 无效; 1: 清除中断。
[2]	DCDMIC	清除 nUARTDCD modem 中断。 0: 无效; 1: 清除中断。
[1]	CTSMIC	清除 nUARTCTS modem 中断。 0: 无效; 1: 清除中断。
[0]	RIMIC	清除 nUARTRI modem 中断。 0: 无效。 1: 清除中断。

15.6.14 UART_DMACR

- 偏移地址: 0x048
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[15:3]	Reserved	保留。
[2]	DMAONERR	1: 当 UART 错误中断有效时, DMA 的请求输出 (UARTRXDMASREQ 或 UARRTXDMABREQ) 不使能。
[1]	TXDMAE	1: 发送 FIFO 的 DMA 使能。
[0]	RXDMAE	1: 接收 FIFO 的 DMA 使能。



16 SSP 接口

关于本章

本章描述内容如下表所示。

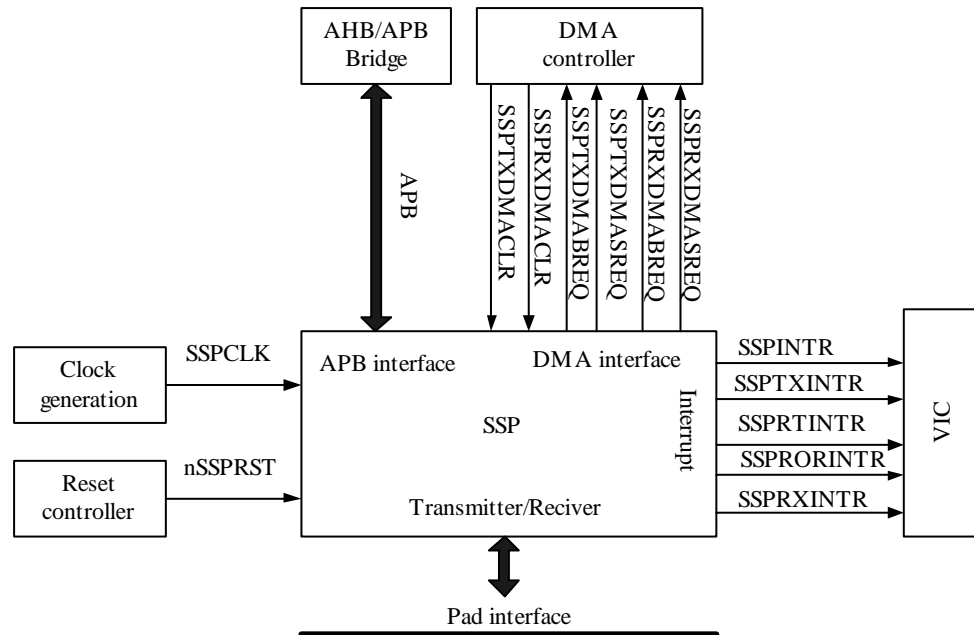
标题	内容
16.1 概述	概括介绍 SSP 接口。
16.2 特点	概括介绍 SSP 接口的特点。
16.3 信号描述	描述 SSP 单元的输入输出管脚信号。
16.4 工作方式	描述 SSP 的工作时序。
16.5 寄存器概览	概括介绍 SSP 接口的寄存器。
16.6 寄存器描述	详细描述 SSP 接口的寄存器。



16.1 概述

SSP (Synchronous Serial Protocol) 单元主要完成系统总线上的 16 位的并行接口到外部串行接口之间的数据转换。SSP 单元功能框图如图 16-1 所示。

图16-1 SSP 功能框图



16.2 特点

说明

Hi3510 暂不支持 SSP 从模式。

SSP 单元有以下特点：

- 产生主操作或从操作
- 可编程时钟率和 prescale
- 内部包含有收/发分开的 16bit 宽、8bit 深度的 FIFO 缓存（接收和发送各 1 个 FIFO）
- 输出接口时序协议可编程：SPI、Microwire、TI synchronous serial
- 数据帧大小可编程：4bit~16bit
- 发送 FIFO、接收 FIFO 和 receive overrun 的中断独立屏蔽
- 内部提供 loopback test mode
- 支持 DMA

16.3 信号描述

本节描述了 SSP 单元的输入输出管脚信号，如表 16-1 所示。

表16-1 SSP 接口信号描述

信号名	方向	描述
SSPSCLK	O	SSP 总线时钟。
SSPRXD	I	SSP 总线数据接收。
SSPTXD	O	SSP 总线数据发送。
SSPSFRM	O	SSP 帧或从设备选择输出信号。

16.4 工作方式

本节描述了 SSP 的工作时序。

SSP 可以配置成 3 种串行接口：

- A Texas Instruments synchronous serial interface
其接口时序如图 16-2 和图 16-3 所示。

图16-2 TI Synchronous Serial Frame Protocol (Single Transfers) 接口时序

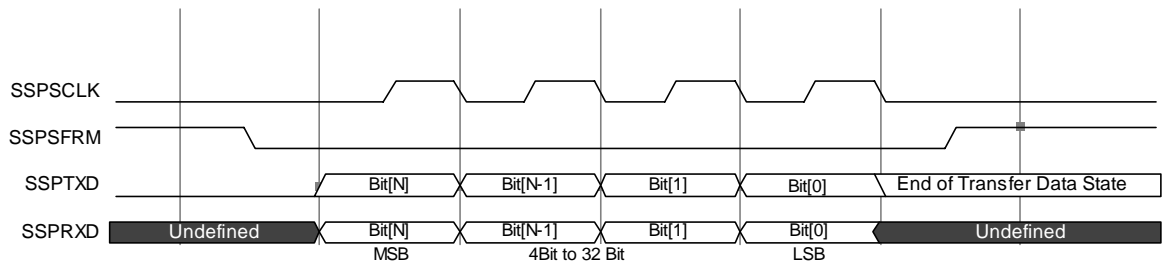
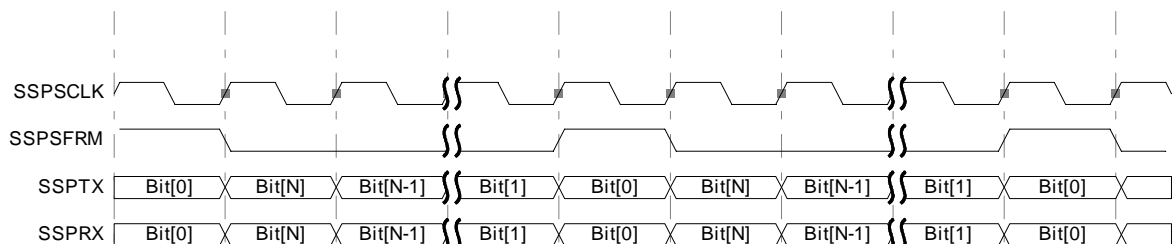


图16-3 TI Synchronous Serial Frame Protocol (Multiple Transfers) 接口时序



- A Motorola SPI-compatible interface



其接口时序如图 16-4 和图 16-5 所示。

图16-4 Motorola SPI Frame Protocol (Single Transfers) 接口时序

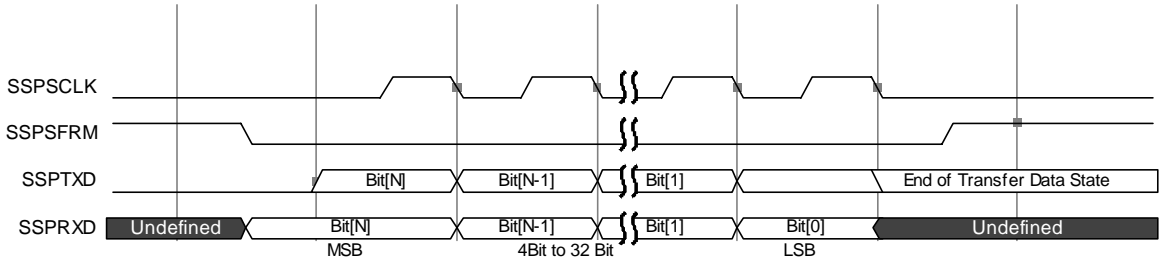
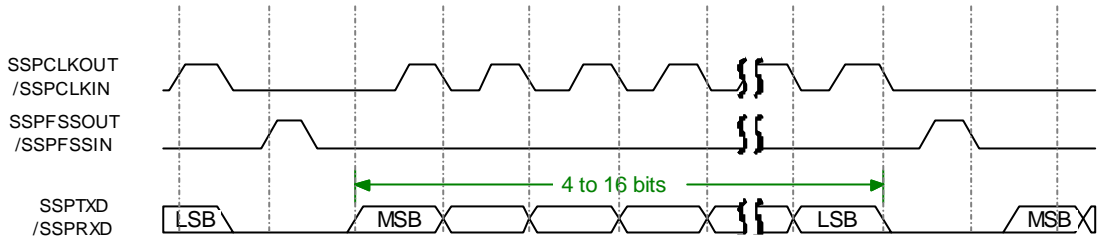


图16-5 Motorola SPI Frame Protocol (Multiple Transfers) 接口时序



- A National Semiconductor Microwire interface
其接口时序如图 16-6 和图 16-7 所示。

图16-6 National Semiconductor MicroWire Protocol (Single Transfers) 接口时序

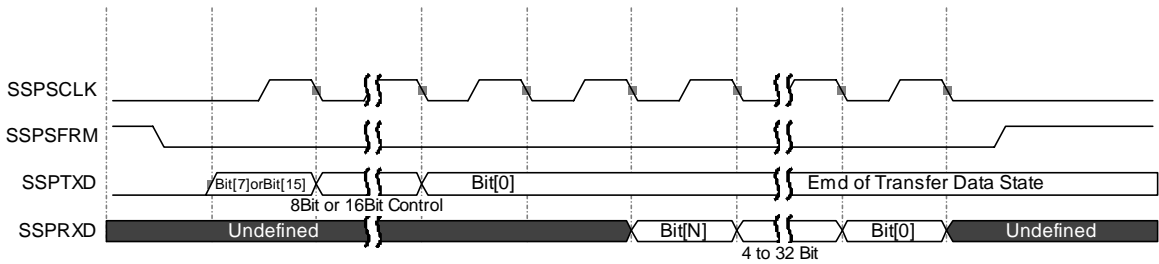
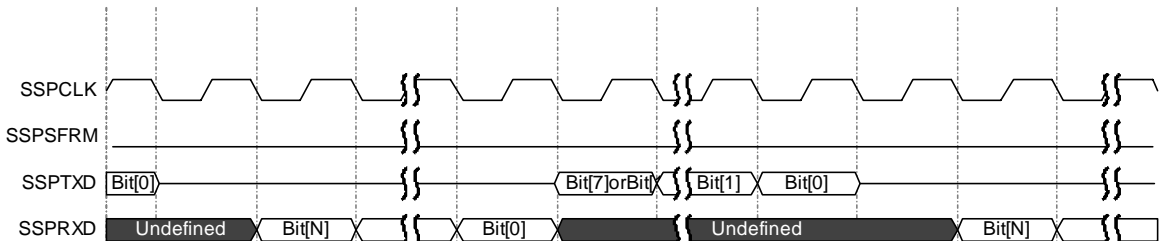


图16-7 National Semiconductor MicroWire Protocol (Multiple Transfers) 接口时序





16.5 寄存器概览

SSP 寄存器的地址位宽 16 位，地址范围是 0x101F_4000~0x101F_4FFF。

表16-2 SSP 寄存器概览（基址是 0x101F_4000）

偏移地址	名称	描述	页码
0x000	SSP_CR0	控制寄存器 0	16-5
0x004	SSP_CR1	控制寄存器 1	16-6
0x008	SSP_DR	接收/发送 FIFO	16-7
0x00C	SSP_SR	状态寄存器	16-7
0x010	SSP_CPSR	时钟生成寄存器	16-8
0x014	SSP_IMSC	中断屏蔽设置/清除寄存器	16-9
0x018	SSP_RIS	原始中断状态寄存器	16-9
0x01C	SSP_MIS	屏蔽后中断状态寄存器	16-10
0x020	SSP_ICR	中断清除寄存器	16-10
0x024	SSP_DMACR	DMA 控制寄存器	16-10

16.6 寄存器描述

本节详细描述了 SSP 单元的寄存器。

16.6.1 控制寄存器 0 (SSP_CR0)

- 偏移地址：0x000
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s



比特	名称	描述
[15:8]	SCR	<p>串行时钟率。</p> <p>SCR 的值用来产生 PrimeCell SSP 发送和接收的比特率（即 SPICK 频率）。</p> <p>比特率由下面公式计算：</p> $F_{SSPCLK} / (CPSDVR \times (1 + SCR))$ <p>CPSDVR 通过寄存器 SSP_CPSR 编程，当 CPSDVR 是一个 2~254 之间的偶数时候，SCR 是一个在 0~255 之间的值。</p> <p>说明</p> <p>SSPCLK 时钟为芯片总线工作时钟，ARM 工作时钟的 1/2。</p>
[7]	SPH	SSPCLK 相位（只对于 Motorola SPI 帧格式适用）。
[6]	SPO	SSPCLK 电平（只对于 Motorola SPI 帧格式适用）。
[5:4]	FRF	<p>帧格式选择。</p> <p>00: Motorola SPI 帧格式；</p> <p>01: TI 同步串行帧格式；</p> <p>10: National Microwire 帧格式；</p> <p>11: 保留。</p>
[3:0]	DSS	<p>数据大小选择。</p> <p>0000: 保留； 1000: 9 bit data;</p> <p>0001: 保留； 1001: 10 bit data;</p> <p>0010: 保留； 1010: 11 bit data;</p> <p>0011: 4 bit data; 1011: 12 bit data;</p> <p>0100: 5 bit data; 1100: 13 bit data;</p> <p>0101: 6 bit data; 1101: 14 bit data;</p> <p>0110: 7 bit data; 1110: 15 bit data;</p> <p>0111: 8 bit data; 1111: 16 bit data.</p>

16.6.2 控制寄存器 1 (SSP_CR1)

- 偏移地址: 0x004
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[15:4]	Reserved	保留。



比特	名称	描述
[3]	SOD	<p>Slave 模式 TXD 输出使能控制位。该位只与 Slave 模式有关 (MS=1)。</p> <p>在多 Slave 设备的系统中，SSP Master 采用广播的方式发信息给所有的 Slave，而只有一个 Slave 将数据驱动到自己的串行输出线上，这样所有 Slave 输出的 RXD 线是连接在一起的。对于这样的系统操作，如果 PrimeCell SSP 不准备驱动 SSP TXD，就置 SOD 为 1。</p> <p>0: SSP 可以在 Slave 模式驱动 SSP TXD 输出； 1: SSP 不能在 Slave 模式驱动 SSP TXD 输出。</p>
[2]	MS	<p>Master 或者 Slave 模式选择。该位只能在 SSP 处于非使能状态下改变。</p> <p>0: 设备被配置成 Master 模式（默认）； 1: 设备被配置成 Slave 模式。</p> <p>Hi3510 只支持 Master 模式，不支持 Slave 模式。</p>
[1]	SSE	<p>同步串行接口使能。</p> <p>0: SSP 操作禁能； 1: SSP 使能。</p>
[0]	LBM	<p>回环模式使能控制位。</p> <p>0: 正常的串行接口操作使能； 1: 发送串行移位寄存器输出，在内部是连接在接收串行移位寄存器输入上。</p>

16.6.3 接收/发送 FIFO (SSP_DR)

- 偏移地址: 0x008
- 操作类型: R/W
- 复位值: -
- 复位方式: h/s

比特	名称	描述
[15:0]	DATA	<p>发送/接收 FIFO。</p> <p>Read: 接收 FIFO; Write: 发送 FIFO;</p>

16.6.4 状态寄存器 (SSP_SR)

- 偏移地址: 0x00C



- 操作类型：R
- 复位值：0x3
- 复位方式：h/s

比特	名称	描述
[15:5]	Reserved	保留。
[4]	BSY	SSP 繁忙标记。 0: SSP 空闲; 1: SSP 当前状态。 SSP 当前状态主要有: <ul style="list-style-type: none"> • 正在同时发送和接收一帧数据 • 正在发送或接收一帧数据 • 发送 FIFO 内部非空
[3]	RFF	接收 FIFO 满状态标志位。 0: 接收 FIFO 非满; 1: 接收 FIFO 满。
[2]	RNE	接收 FIFO 非空状态标志位。 0: 接收 FIFO 空; 1: 接收 FIFO 非空。
[1]	TNF	发送 FIFO 非满状态标志位。 0: 发送 FIFO 满; 1: 发送 FIFO 非满。
[0]	TFE	发送 FIFO 空状态标志位。 0: 发送 FIFO 非空; 1: 发送 FIFO 空。

16.6.5 时钟生成寄存器 (SSP_CPSR)

- 偏移地址：0x0010
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:8]	Reserved	保留。



比特	名称	描述
[7:0]	CPSDVSR	Clock prescale divisor。 必须是 2~254 之间的偶数，依靠输入时钟 SSPCLK 的频率。最低位读作 0。

16.6.6 中断屏蔽设置/清除寄存器 (SSP_IMSC)

- 偏移地址：0x0014
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:4]	Reserved	保留。
[3]	TXIM	发送 FIFO 中断屏蔽。 0: 发送 FIFO 半空或更少情况下中断是屏蔽的； 1: 发送 FIFO 半空或更少情况下中断不是屏蔽的。
[2]	RXIM	接收 FIFO 中断屏蔽。 0: 接收 FIFO 半空或更少情况下中断是屏蔽的； 1: 接收 FIFO 半空或更少情况下中断不是屏蔽的。
[1]	RTIM	接收 timeout 中断。 0: 接收 FIFO 不空和在 timeout 期中断是屏蔽的； 1: 接收 FIFO 不空和在 timeout 期中断不是屏蔽的。
[0]	RORIM	接收 overrun 中断屏蔽。 0: 接收 FIFO 在满情况下的写中断是屏蔽的； 1: 接收 FIFO 在满情况下的写中断不是屏蔽的。

16.6.7 原始中断状态寄存器 (SSP_RIS)

- 偏移地址：0x0018
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:4]	Reserved	保留。



比特	名称	描述
[3]	TXRIS	给出 SSPTXINTR 中断的原始（屏蔽之前的）中断状态。
[2]	RXRIS	给出 SSPRXINTR 中断的原始（屏蔽之前的）中断状态。
[1]	RTRIS	给出 SSPRTXINTR 中断的原始（屏蔽之前的）中断状态。
[0]	RORRIS	给出 SSPRORXINTR 中断的原始（屏蔽之前的）中断状态。

16.6.8 屏蔽后中断状态寄存器（SSP_MIS）

- 偏移地址：0x001C
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:4]	Reserved	保留。
[3]	TXMIS	给出发送 FIFO 屏蔽中断状态（屏蔽之后）SSPTXINTR 中断。
[2]	RXMIS	给出接收 FIFO 屏蔽中断状态（屏蔽之后）SSPRXINTR 中断。
[1]	RTMIS	给出接收 timeout 屏蔽中断状态（屏蔽之后）SSPRTINTR 中断。
[0]	RORMIS	给出接收 overrun 屏蔽中断状态（屏蔽之后）SSPRORINTR 中断。

16.6.9 中断清除寄存器（SSP_ICR）

- 偏移地址：0x0020
- 操作类型：W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:2]	Reserved	保留。
[1]	RTIC	清除 SSPRTINTR 中断。写 1 清除。
[0]	RORIC	清除 SSPRORINTR 中断。写 1 清除。



16.6.10 DMA 控制寄存器 (SSP_DMACR)

- 偏移地址: 0x0024
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[15:2]	Reserved	保留。
[1]	TXDMAE	1: DMA 对于发送 FIFO 是使能的。
[0]	RXDMAE	1: DMA 对于接收 FIFO 是使能的。



17 I²C 接口

关于本章

本章描述内容如下表所示。

标题	内容
17.1 概述	概括介绍 I ² C。
17.2 特点	概括介绍 I ² C 的特点。
17.3 信号描述	描述 I ² C 单元的输入输出管脚信号。
17.4 工作方式	描述 I ² C 的工作时序。
17.5 寄存器概览	概括介绍 I ² C 的寄存器。
17.6 寄存器描述	详细描述 I ² C 的寄存器。



17.1 概述

I²C (The Inter-Integrated Circuit) 接口有 2 个信号: SDA (串行数据或地址线) 和 SCL (串行时钟线)。I²C 既可以作为 I²C 总线 Master, 也可以作为总线 Slave, 遵守 I²C 总线协议 2.1 版本。作为 Master 时, 可用于 Video Encoder、Video Decoder、Digital Camera 的控制接口。

17.2 特点

I²C 单元有以下特点:

- I²C 单元在 I²C 总线上可以作为接收器, 也可以作为发送器, 具体由数据的传输方向确定
- 支持标准的 100kbit/s 数据传输速率和快速模式 400kbit/s
- 提供 TX FIFO、RX FIFO, 支持 DMA 数据传输
- 支持中断上报和初始中断状态、屏蔽后中断状态的查询

17.3 信号描述

本节描述了 I²C 单元的输入输出管脚信号, 如表 17-1 所示。

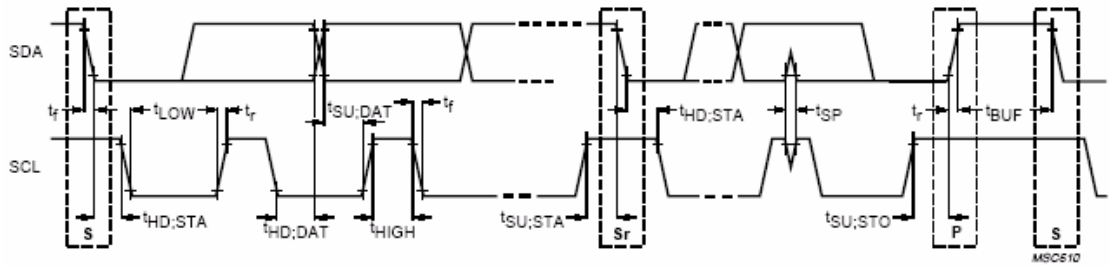
表17-1 I²C 接口信号描述

信号名	方向	描述
SDA	I/O	I ² C 总线数据/地址。
SCL	I/O	I ² C 总线时钟。

17.4 工作方式

本节描述了 I²C 的工作时序。

I²C 数据传输格式如图 17-1 所示, 此图来源于《THE I²C-BUS SPECIFICATION VERSION 2.1》。

图17-1 I²C 数据传输格式


I²C 完全符合 I²C Spec.version 2.1，详细工作机制请参见《THE I²C-BUS SPECIFICATION VERSION 2.1》。

17.5 寄存器概览

I²C 寄存器的地址位宽 16 位，地址范围是 0x101F_6000~0x101F_6FFF。

表17-2 I2C 寄存器概览（基址是 0x101F_6000）

偏移地址	名称	描述	页码
0x000	IIC_CON	I ² C 控制寄存器	17-5
0x004	IIC_TAR	I ² C 访问从设备地址寄存器	17-6
0x008	IIC_SAR	I ² C 作为从设备地址寄存器	17-6
0x00C	IIC_HS_MADDR	I ² C 的高速主机码配置寄存器	17-7
0x010	IIC_DATA_CMD	I ² C 数据通道寄存器	17-7
0x014	IIC_SS_SCL_HCNT	标准速度下的 SCL 时钟高电平时间配置寄存器	17-8
0x018	IIC_SS_SCL_LCNT	标准速度下的 SCL 时钟低电平时间配置寄存器	17-8
0x01C	IIC_FS_SCL_HCNT	快速速度下的 SCL 时钟高电平时间配置寄存器	17-8
0x020	IIC_FS_SCL_LCNT	快速速度下的 SCL 时钟低电平时间配置寄存器	17-8
0x024	IIC_HS_SCL_HCNT	高速速度下的 SCL 时钟高电平时间配置寄存器	17-9
0x028	IIC_HS_SCL_LCNT	高速速度下的 SCL 时钟低电平时间配置寄存器	17-9
0x02C	IIC_INTR_STAT	中断寄存器	17-9



偏移地址	名称	描述	页码
0x030	IIC_INTR_MASK	中断掩码寄存器	17-10
0x034	IIC_RAW_INTR_STAT	原始中断寄存器	17-11
0x038	IIC_RX_TL	RX_FIFO 的水线配置寄存器	17-13
0x03C	IIC_TX_TL	TX_FIFO 的水线配置寄存器	17-13
0x040	IIC_CLR_INTR	清所有中断的寄存器	17-13
0x044	IIC_CLR_RX_UNDER	清 RX_UNDER 中断寄存器	17-14
0x048	IIC_CLR_RX_OVER	清 RX_OVER 中断寄存器	17-14
0x04C	IIC_CLR_TX_OVER	清 TX_OVER 中断寄存器	17-14
0x050	IIC_CLR_RD_REQ	清 RD_REQ 中断寄存器	17-15
0x054	IIC_CLR_TX_ABORT	清 TX_ABORT 中断, 以及 IIC_TX_ABORT_SOURCE 中断寄存器	17-15
0x058	IIC_CLR_RX_DONE	清 RX_DONE 中断寄存器	17-15
0x05C	IIC_CLR_ACTIVITY	I ² C ACTIVITY 状态寄存器	17-16
0x060	IIC_CLR_STOP_DET	清 STOP_DET 中断寄存器	17-16
0x064	IIC_CLR_START_DET	清 START_DET 中断寄存器	17-16
0x068	IIC_CLR_GEN_CALL	清 GEN_CALL 中断寄存器	17-16
0x06C	IIC_ENABLE	I ² C 工作模式使能寄存器	17-17
0x070	IIC_STATUS	I ² C 状态寄存器	17-17
0x074	IIC_TXFLR	TX_FIFO 中的数据个数指示寄存器	17-18
0x078	IIC_RXFLR	RX_FIFO 中的数据个数指示寄存器	17-18
0x080	IIC_TX_ABORT_SOURCE	TX_ABORT 的源头中断寄存器	17-19
0x088	IIC_DMA_CR	I ² C DMA 通道开启控制寄存器	17-20
0x08C	IIC_DMA_TDLR	TX_FIFO DMA 操作时的水线配置寄存器	17-20



偏移地址	名称	描述	页码
0x090	IIC_DMA_RDLR	RX_FIFO DMA 操作时的水线配置寄存器	17-20
0x0F4	IIC_COMP_PAR AM_1	I ² C 配置参数寄存器	17-21

17.6 寄存器描述

本节详细描述了 I²C 的寄存器。

17.6.1 IIC_CON

- 偏移地址：0x000
- 操作类型：R/W
- 复位值：0x74
- 复位方式：h/s

比特	名称	描述
[15:7]	Reserved	保留位。
[6]	IIC_SLAVE_DISABLE	从设备功能使能。 0: 使能; 1: 不使能。
[5]	IIC_RESTART_EN	允许主设备产生“重新开始”条件。 0: 不允许, 此时无法实现“重新开始”条件的功能; 1: 允许。
[4]	IIC_10BITADDR_MASTER	作为主设备时发出 7 位地址/10 位地址。 0: 7 位地址; 1: 10 位地址。
[3]	IIC_10BITADDR_SLAVE	作为从设备时响应 7 位地址/10 位地址。 0: 7 位地址; 1: 10 位地址。



比特	名称	描述
[2:1]	SPEED	I ² C 操作速度的选择。 00: 非法, 但写 00 会被认为是配置的最大速度模式; 01: 标准速度 (100kbit/s); 10: 快速速度 (400kbit/s); 11: Reserved。 当写入值大于参数配置的最大值时, 会被认为是配置的最大值。 如配置成快速速度时, 写入 11 会被认为是写入 10。
[0]	MASTER_MODE	主设备功能使能位。 0: 不使能; 1: 使能。

17.6.2 IIC_TAR

- 偏移地址: 0x004
- 操作类型: R/W
- 复位值: 0x9C
- 复位方式: h/s

比特	名称	描述
[15:12]	Reserved	保留。
[11]	SPECIAL	执行 general call 和 start byte 功能使能信号。 0: 不执行; 1: 执行。
[10]	GC_OR_START	如果 SPECIAL 为 1, 决定执行功能是 general call 或 start byte。 0: general call; 1: start byte。
[9:0]	IIC_TAR	I ² C 作为主设备时要访问的从设备的地址。

17.6.3 IIC_SAR

- 偏移地址: 0x008
- 操作类型: R/W
- 复位值: 0x55



- 复位方式：h/s

比特	名称	描述
[15:10]	Reserved	保留。
[9:0]	IIC_SAR	I ² C 作为从设备时自己的地址。

17.6.4 IIC_HS_MADDR

- 偏移地址：0x00C
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:3]	Reserved	保留。
[2:0]	IIC_HS_MAR	能够发起高速传输的主设备的地址。

17.6.5 IIC_DATA_CMD

- 偏移地址：0x010
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:9]	Reserved	保留。
[8]	CMD	读/写控制位。 0：写； 1：读。 • 写操作时：表示 I ² C 将要向 I ² C 总线发送数据。此时低 8 位（DAT）是 I ² C 要向 I ² C 总线发送的数据。 • 读操作时：表示 I ² C 将从 I ² C 总线读回数据。此时 I ² C 会忽略低 8 位。
[7:0]	DAT	将要在 I ² C 总线上发送/接收的数据。 • 桥读这 8 位会读出在 I ² C 总线上接收的数据。 • 桥写这 8 位会把写入的数据发送到 I ² C 总线上。



17.6.6 IIC_SS_SCL_HCNT

- 偏移地址：0x014
- 操作类型：R/W
- 复位值：0x7A
- 复位方式：h/s

比特	名称	描述
[15:0]	IIC_SS_SCL_HCNT	标准速度下的 SCL 时钟高电平时间。 配置的最小值为 6，写入小于 6 的值时会被认为是 6。

17.6.7 IIC_SS_SCL_LCNT

- 偏移地址：0x018
- 操作类型：R/W
- 复位值：0x8F
- 复位方式：h/s

比特	名称	描述
[15:0]	IIC_SS_SCL_LCNT	标准速度下的 SCL 时钟低电平时间。 配置的最小值为 8，写入小于 8 的值时会被认为是 8。

17.6.8 IIC_FS_SCL_HCNT

- 偏移地址：0x01C
- 操作类型：R/W
- 复位值：0x13
- 复位方式：h/s

比特	名称	描述
[15:0]	IIC_FS_SCL_HCNT	快速速度下的 SCL 时钟高电平时间。 配置的最小值为 6，写入小于 6 的值时会被认为是 6。

17.6.9 IIC_FS_SCL_LCNT

- 偏移地址：0x020



- 操作类型: R/W
- 复位值: 0x28
- 复位方式: h/s

比特	名称	描述
[15:0]	IIC_FS_SCL_LCNT	快速速度下的 SCL 时钟低电平时间。 配置的最小值为 8，写入小于 8 的值时会被认为是 8。

17.6.10 IIC_HS_SCL_HCNT

- 偏移地址: 0x024
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[15:0]	IIC_HS_SCL_HCNT	高速速度下的 SCL 时钟高电平时间。 配置的最小值为 6，写入小于 6 的值时会被认为是 6。

17.6.11 IIC_HS_SCL_LCNT

- 偏移地址: 0x028
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[15:0]	IIC_HS_SCL_LCNT	高速速度下的 SCL 时钟低电平时间。 配置的最小值为 8，写入小于 8 的值时会被认为是 8。

17.6.12 IIC_INTR_STAT

- 偏移地址: 0x02C
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s



比特	名称	描述
[15:12]	Reserved	保留。
[11]	GEN_CALL	指示一次 general call 请求已被接收。 I ² C 将接收到的数据放在 RX Buffer 中。
[10]	START_DET	指示在 I ² C 总线上已经发生了开始条件。
[9]	STOP_DET	指示在 I ² C 总线上已经发生了停止条件。
[8]	ACTIVITY	指示 I ² C 已经处于 ACTIVITY 状态。
[7]	RX_DONE	作为从设备发送器时，发送的字节没被响应，此位被设置为 1。这种情况通常发生在传输的最后一个字节，用来指示传输结束。
[6]	TX_ABRT	有多种情况可以触发此位。详细描述请参见“ 17.6.32 IIC_TX_ABRT_SOURCE ”。
[5]	RD_REQ	作为从设备，I ² C 总线上其他的设备发出读请求时，此位被设置为 1。I ² C 将会拉低 SCL 直到 CPU 响应中断并写入请求的数据。
[4]	TX_EMPTY	当 TX_FIFO 到达或低于水位值时，此位被设置为 1；硬件自动清 0。
[3]	TX_OVER	当 TX_FIFO 已满，CPU 仍写入数据时，此位被设置为 1。
[2]	RX_FULL	当 RX_FIFO 到达或高于水位值时，此位被设置为 1；硬件自动清 0。
[1]	RX_OVER	当 RX_FIFO 已满，外设仍写入数据时，此位被设置为 1。
[0]	RX_UNDER	当 CPU 读空的 RX_FIFO 时，此位被设置为 1。

17.6.13 IIC_INTR_MASK

中断掩码寄存器。某位为 0 时会阻止其产生中断。

- 偏移地址：0x030
- 操作类型：R/W
- 复位值：0x8FF
- 复位方式：h/s

比特	名称	描述
[15:12]	Reserved	保留。



比特	名称	描述
[11]	M_GEN_CALL	0: 阻止产生中断; 1: 允许产生中断。
[10]	M_START_DET	0: 阻止产生中断; 1: 允许产生中断。
[9]	M_STOP_DET	0: 阻止产生中断; 1: 允许产生中断。
[8]	M_ACTIVITY	0: 阻止产生中断; 1: 允许产生中断。
[7]	M_RX_DONE	0: 阻止产生中断; 1: 允许产生中断。
[6]	M_TX_ABRT	0: 阻止产生中断; 1: 允许产生中断。
[5]	M_RD_REQ	0: 阻止产生中断; 1: 允许产生中断。
[4]	M_TX_EMPTY	0: 阻止产生中断; 1: 允许产生中断。
[3]	M_TX_OVER	0: 阻止产生中断; 1: 允许产生中断。
[2]	M_RX_FULL	0: 阻止产生中断; 1: 允许产生中断。
[1]	M_RX_OVER	0: 阻止产生中断; 1: 允许产生中断。
[0]	M_RX_UNDER	0: 阻止产生中断; 1: 允许产生中断。

17.6.14 IIC_RAW_INTR_STAT

- 偏移地址: 0x034
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s



比特	名称	描述
[15:12]	Reserved	保留。
[11]	R_GEN_CALL	原始中断状态。 0: 未发生中断; 1: 发生中断。
[10]	R_START_DET	原始中断状态。 0: 未发生中断; 1: 发生中断。
[9]	R_STOP_DET	原始中断状态。 0: 未发生中断; 1: 发生中断。
[8]	R_ACTIVITY	原始中断状态。 0: 未发生中断; 1: 发生中断。
[7]	R_RX_DONE	原始中断状态。 0: 未发生中断; 1: 发生中断。
[6]	R_TX_ABRT	原始中断状态。 0: 未发生中断; 1: 发生中断。
[5]	R_RD_REQ	原始中断状态。 0: 未发生中断; 1: 发生中断。
[4]	R_TX_EMPTY	原始中断状态。 0: 未发生中断; 1: 发生中断。
[3]	R_TX_OVER	原始中断状态。 0: 未发生中断; 1: 发生中断。
[2]	R_RX_FULL	原始中断状态。 0: 未发生中断; 1: 发生中断。



比特	名称	描述
[1]	R_RX_OVER	原始中断状态。 0: 未发生中断; 1: 发生中断。
[0]	R_RX_UNDER	原始中断状态。 0: 未发生中断; 1: 发生中断。

17.6.15 IIC_RX_TL

- 偏移地址: 0x038
- 操作类型: R/W
- 复位值: 0x3
- 复位方式: h/s

比特	名称	描述
[15:8]	Reserved	保留。
[7:0]	RX_TL	RX_FIFO 的水线值。 配置值超过最大值时被认为是最大值。

17.6.16 IIC_TX_TL

- 偏移地址: 0x03C
- 操作类型: R/W
- 复位值: 0x3
- 复位方式: h/s

比特	名称	描述
[15:8]	Reserved	保留。
[7:0]	TX_TL	TX_FIFO 的水线值。 配置值超过最大值时被认为是最大值。

17.6.17 IIC_CLR_INTR

- 偏移地址: 0x040
- 操作类型: R



- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[15:1]	Reserved	保留。
[0]	CLR_INTR	读此寄存器, 清组合中断、分立中断、IIC_TX_ABRT_SOURCE 中断。

17.6.18 IIC_CLR_RX_UNDER

- 偏移地址: 0x044
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[15:1]	Reserved	保留。
[0]	CLR_RX_UNDER	读此寄存器, 清 RX_UNDER 中断。

17.6.19 IIC_CLR_RX_OVER

- 偏移地址: 0x048
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[15:1]	Reserved	保留。
[0]	CLR_RX_OVER	读此寄存器, 清 RX_OVER 中断。

17.6.20 IIC_CLR_TX_OVER

- 偏移地址: 0x04C
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s



比特	名称	描述
[15:1]	Reserved	保留。
[0]	CLR_TX_OVER	读此寄存器，清 TX_OVER 中断。

17.6.21 IIC_CLR_RD_REQ

- 偏移地址：0x050
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:1]	Reserved	保留。
[0]	CLR_RD_REQ	读此寄存器，清 RD_REQ 中断。

17.6.22 IIC_CLR_TX_ABRT

- 偏移地址：0x054
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:1]	Reserved	保留。
[0]	CLR_TX_ABRT	读此寄存器，清 TX_ABRT 中断、IIC_TX_ABRT_SOURCE 寄存器。

17.6.23 IIC_CLR_RX_DONE

- 偏移地址：0x058
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:1]	Reserved	保留。
[0]	CLR_RX_DONE	读此寄存器，清 RX_DONE 中断。



17.6.24 IIC_CLR_ACTIVITY

- 偏移地址：0x05C
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:1]	Reserved	保留。
[0]	CLR_ACTIVITY	读此寄存器可获得 ACTIVITY 中断状态，硬件自动清 0。

17.6.25 IIC_CLR_STOP_DET

- 偏移地址：0x060
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:1]	Reserved	保留。
[0]	CLR_STOP_DET	读此寄存器，清 STOP_DET 中断。

17.6.26 IIC_CLR_START_DET

- 偏移地址：0x064
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[15:1]	Reserved	保留。
[0]	CLR_START_DET	读此寄存器，清 START_DET 中断。

17.6.27 IIC_CLR_GEN_CALL

- 偏移地址：0x068
- 操作类型：R



- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[15:1]	Reserved	保留。
[0]	CLR_GEN_CALL	读此寄存器, 清 GEN_CALL 中断。

17.6.28 IIC_ENABLE

- 偏移地址: 0x06C
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[15:1]	Reserved	保留。
[0]	ENABLE	I ² C 使能寄存器。 0: 分下面 2 种情况。 <ul style="list-style-type: none"> • 处于发送状态时, 在完成当前字节发送后停止, 同时删除 TX_FIFO 的数据。 • 处于接收状态时, 在接收完当前字节后不响应这次传输。 1: 使 I ² C 进入工作状态, 写 0 后可对其配置。软件不要在 I ² C 处于 ACTIVITY 时写 0。

17.6.29 IIC_STATUS

- 偏移地址: 0x070
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[4]	RFF	指示 RX_FIFO 是否已满。 0: RX_FIFO 未滿; 1: RX_FIFO 已滿。 当 RX_FIFO 已滿后此位置 1; 当有一个或多个空位后, 返回 0 值。



比特	名称	描述
[3]	RFNE	指示 RX_FIFO 是否已空。 0: RX_FIFO 已空; 1: RX_FIFO 未空。 当 RX_FIFO 已空后此位置 1; 当有一个或多个空位后, 返回 0 值。
[2]	TFE	指示 TX_FIFO 是否已空。 0: TX_FIFO 没空; 1: TX_FIFO 已空。
[1]	TFNF	指示 TX_FIFO 是否已满。 0: TX_FIFO 已满; 1: TX_FIFO 没满。
[0]	ACTIVITY	指示 I ² C 的 ACTIVITY 状态。

17.6.30 IIC_TXFLR

- 偏移地址: 0x074
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	TXFLR	指示 TX_FIFO 中的数据个数。

17.6.31 IIC_RXFLR

- 偏移地址: 0x078
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	RXFLR	指示 RX_FIFO 中的数据个数。



17.6.32 IIC_TX_ABRT_SOURCE

- 偏移地址：0x080
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:16]	Reserved	保留。
[15]	ABRT_SLVRD_INTX	作为从设备时，请求 CPU 向 TX_FIFO 中写入数据。而 CPU 却向 TX_FIFO 发送了一个读命令（IIC_DATA_CMD 寄存器中的 CMD 位为 1）。
[14]	ABRT_SLV_ARBLOST	作为从设备时，在向远程的主设备发送数据时丢失了总线。
[13]	ABRT_SLVFLUSH_TX_FIFO	作为从设备时，收到读命令时 TX_FIFO 中有数据。 此时从设备发起 TX_ABRT 中断并清除 TX_FIFO 中的数据。
[12]	ARB_LOST	作为主设备时，仲裁失败。
[11]	ARB_MASTER_DIS	尝试使用未使能 Master 功能的 I ² C 的 Master 功能。
[10]	ABRT_10B_RD_NORS TRT	不支持 restart 功能时，作为主设备时对 10 位地址的从设备发出了读命令。
[9]	ABRT_SBYTE_NORST RT	不支持 restart 功能时，作为主设备时尝试发送 start byte。
[8]	ABRT_HS_NORSTRT	不支持 restart 功能时，作为主设备时尝试高速操作。
[7]	ABRT_SBYTE_ACKDE T	作为主设备时发出 start byte 而被响应。
[6]	ABRT_HS_ACKDET	作为主设备要进行高速传输时，高速主机码被响应。
[5]	ABRT_GCALL_READ	作为主设备时发出 general call，而 CPU 向 I ² C 发出读命令。
[4]	ABRT_GCALL_NOAC K	作为主设备时发出 general call，但没被响应。
[3]	ABRT_TXDATA_NOA CK	作为主设备发送器，发送的地址被从设备响应，而发送的数据没被响应。



比特	名称	描述
[2]	ABRT_10ADDR2_NOACK	作为主设备时，发送的 10 位地址的第 2 字节没被响应。
[1]	ABRT_10ADDR1_NOACK	作为主设备时，发送的 10 位地址的第 1 字节没被响应。
[0]	ABRT_7B_ADDR_NOACK	作为主设备时，发送的 7 位地址没被响应。

17.6.33 IIC_DMA_CR

- 偏移地址：0x088
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[1]	TDMAE	是否打开 TX_FIFO 的 DMA 通道。 0: 不打开; 1: 打开。
[0]	RDMAE	是否打开 RX_FIFO 的 DMA 通道。 0: 不打开; 1: 打开。

17.6.34 IIC_DMA_TDLR

- 偏移地址：0x08C
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[2:0]	DMATDL	TX_FIFO DMA 操作时的水位值。 配置值超过最大值时被认为是最大值。

17.6.35 IIC_DMA_RDLR

- 偏移地址：0x090



- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[2:0]	DMARDL	RX_FIFO DMA 操作时的水位值。配置值超过最大值时被认为是最大值。

17.6.36 IIC_COMP_PARAM_1

- 偏移地址：0x0F4
- 操作类型：R
- 复位值：0x707EA
- 复位方式：h/s

比特	名称	描述
[31:24]	Reserved	保留。
[23:16]	TX_BUFFER_DEPTH	配置的 TX_FIFO 的深度值。
[15:8]	RX_BUFFER_DEPTH	配置的 RX_FIFO 的深度值。
[7]	ADD_ENCODED_PARAMS	配置的 ADD_ENCODED_PARAMS 参数值。 0: False, 未提供一个 IP 地址映射上访问 designware 组件的简单方法。 1: True, 提供一个 IP 地址映射上访问 designware 组件的简单方法。
[6]	HAS_DMA	是否配置了 DMA 接口。 0: 否; 1: 是。
[5]	INTR_IO	中断是组合输出还是分离输出。 0: 分离; 1: 组合。
[4]	HC_COUNT_VALUES	6 个分频数 * CNT 寄存器是只读/可写。 0: 可写; 1: 只读。



比特	名称	描述
[3:2]	MAX_SPEED_MODE	配置的最大速度模式。 00: 保留; 01: 标准; 10: 快速; 11: 高速。
[1:0]	APB_DATA_WIDTH	指示配置的 APB 数据总线宽度。 00: 8 位; 01: 16 位; 10: 32 位; 11: 保留。



18 以太网交换单元 (SF)

关于本章

本章描述内容如下表所示。

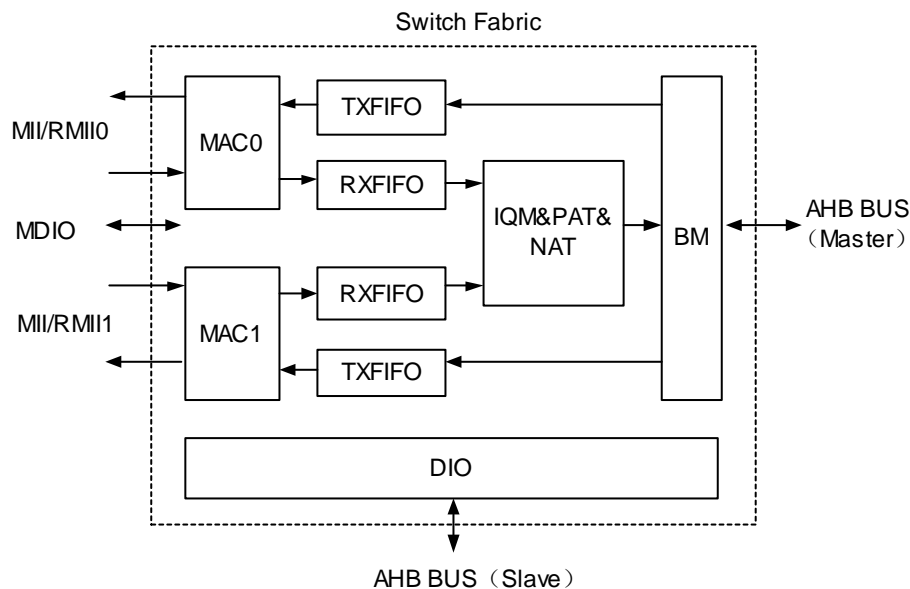
标题	内容
18.1 概述	概括介绍 SF 单元。
18.2 特点	概括介绍 SF 单元的特点。
18.3 信号描述	描述 SF 单元的外部输入输出的管脚信号。
18.4 工作方式	描述了 SF 的接口时序、初始化配置、中断管理和 CPU 端口收发帧。
18.5 寄存器概览	概括介绍 SF 单元的寄存器。
18.6 寄存器描述	详细描述 SF 单元的寄存器。



18.1 概述

SF (Switch Fabric) 实现芯片内部与外部通信的功能。SF 提供 2 个 10/100 Mbit/s 以太网端口，支持 1 个虚拟的 CPU 端口，实现 2 个外部端口和 CPU 端口之间无阻塞的数据包交换。在 3 个端口间的交换通过读写片外 SDRAM 指定区域来实现。SF 支持 VLAN 的功能。

图18-1 SF 功能框图



18.2 特点

提供 1 个 CPU 端口和 2 个外部端口

- SF 支持 1 个虚拟的 CPU 端口。
CPU 通过 SF 的 AHB_SLAVE 接口通知 SF 发送 SDRAM 中的帧数据或根据 SF 提供的信息接收 SDRAM 中的帧数据。
- 2 个外部端口为 10/100 Mbit/s 以太网端口，即端口 0 和端口 1。
 - 2 个外部端口均支持全双工和半双工模式；
 - 支持 MII (Media Independent Interface) 和 RMII (Reduced MII) 接口；
 - 端口 1 用于连接到上行网络，称为上行口；端口 0 用于连接 PC 等家用网络设备或一个小型的局域网，称为下行口。



以存储-转发的方式实现 3 个端口间数据的交换

- 端口接收数据时，SF 将接收到的帧数据按转发的端口缓存到片内 Memory 或片外 SDRAM 区中；
- 端口发送数据时，SF 从片内或片外的缓存队列中读取数据包发送到外部端口或通知 CPU 端口取数据；
- 外部端口给 CPU 的数据，直接写到 SDRAM 指定区域中，由 CPU 从中读取完成接收。CPU 发出的数据写到 SDRAM 中，通知 SF 从中读取，发送出去。

支持 2 种工作模式

- 普通工作模式：SF 对输入帧不做修改，直接按目的 MAC 地址转发或丢弃。
普通模式下的转发方式：
 - 特殊帧按控制转发；
 - CPU 端口的帧按 CPU 指定端口发送；
 - 外部端口接收的帧与本机 MAC 一致转发到 CPU 端口，否则转发到另外一个外部端口。
- 监听模式：通过设定监听模式，可指定 SF 一个外部端口为监听端口，此时 CPU 和另外一个端口间交换的数据全被复制到监听端口。

支持对特殊帧的识别和转发控制

- 广播帧
对目的 MAC 为全 1 的帧，CPU 可通过配置控制两个端口的转发方式，具体请参见 [SF_IQM_REG0](#)；
- 特殊目的 MAC 帧
对目的 MAC 在 48'h01_80_C2_00_00_00~48'h01_80_C2_00_00_10 和 48'h01_80_C2_00_00_20~48'h01_80_C2_00_00_2f 之间的帧，CPU 可按端口配置识别使能以及转发方式，具体请参见 [SF_IQM_REG0](#)；
- IP 多播帧
对目的 MAC 高 24 位是 16'h01_00_5E，第 23 位为 0 的 IP 报文，CPU 可按端口配置进行识别，并配置转发方式，具体请参见 [SF_IQM_REG0](#)；
- IGMP 帧
对目的 MAC 与 IP 多播帧相同，且 IP Protocol 为 2 的 IP 报文，CPU 可按端口配置进行识别，并配置转发方式；
- ANYTYPE 帧（任意目的 MAC 地址，以太网类型和 VLAN ID 帧）
CPU 可通过配置指定目的 MAC 或 VID 或 Ether TYPE 值为特定值的帧按 CPU 配置转发，每端口可配置（最多可配置 8 种）。
- ANYPORT 帧（任意协议和协议端口号帧）
CPU 可通过配置指定某种协议类型及协议端口号的 IP 报文发往 CPU 处理，每端口可独立配置（最多可配置 64 种），具体请参见[特殊帧配置说明](#)；



支持符合 IEEE 802.1Q 的基于端口的 VLAN，最多支持 8 个静态 VLAN

- 支持对两个外部端口所允许接收帧的 VLAN 格式进行控制；
- 支持对发送到两个外部端口的帧输出 VLAN 格式进行控制（按端口配置或按帧所属 VLAN 进行控制）；
- 支持对来自两个外部端口的帧转发是否受 VLAN 域限制可配置；
- 支持两个外部端口未知 VLAN、未知 VLAN 成员帧处理可配置。

支持对发送到外部端口的帧按源端口配置添加 IEEE.802.1p 优先级

18.3 信号描述

本节描述 SF 单元的外部输入输出的管脚信号，如表 18-1 所示。

表18-1 以太网交换单元接口信号描述

信号名	方向	频率 (Hz)	描述
RMIICLK	I	50M	RMII 接口时钟输入(MII 模式下,连接 GND)。
ATXD0	O	50M (RMII) 25M (MII)	端口 0MII0 发送数据 0 或 RMII0 发送数据 0。
ATXD1	O	50M (RMII) 25M (MII)	端口 0MII0 发送数据 1 或 RMII0 发送数据 1。
ATXD2	O	25M	端口 0MII0 发送数据 2; RMII0 模式下, 不连接。
ATXD3	O	25M	端口 0MII0 发送数据 3; RMII0 模式下, 不连接。
ATXEN	O	50M (RMII) 25M (MII)	端口 0MII0/RMII0 发送数据使能。
ATXERR	O	25M	端口 0MII0 发送数据错误; RMII0 模式下, 不连接。
ATXCK	I	25M	端口 0MII0 发送数据时钟; RMII0 模式下, 不连接。
ARXD0	I	50M (RMII) 25M (MII)	端口 0MII0 接收数据 0; 或 RMII0 接收数据 0。
ARXD1	I	50M (RMII) 25M (MII)	端口 0MII0 接收数据 1; 或 RMII0 接收数据 1。



信号名	方向	频率 (Hz)	描述
ARXD2	I	25M	端口 0MII0 接收数据 2; RMII0 模式下, 连接 GND。
ARXD3	I	25M	端口 0MII0 接收数据 3; RMII0 模式下, 连接 GND。
ARXDV	I	50M (RMII) 25M (MII)	端口 0MII0/RMII0 接收数据有效。
ARXERR	I	50M (RMII) 25M (MII)	端口 0MII0/RMII0 接收数据错误。
ARXCK	I	25M	端口 0MII0 接收数据时钟; RMII0 模式下, 连接 GND。
ACRS	I	25M	端口 0MII0 载波侦听; RMII0 模式下, 连接 GND。
ACOL	I	25M	端口 0MII0 碰撞指示; RMII0 模式下, 连接 GND。
BTXD0	O	50M (RMII) 25M (MII)	端口 1MII1 发送数据 0 或 RMII1 发送数据 0。
BTXD1	O	50M (RMII) 25M (MII)	端口 1MII1 发送数据 1 或 RMII1 发送数据 1。
BTXD2	O	25M	端口 1MII1 发送数据 2; RMII1 模式下, 不连接。
BTXD3	O	25M	端口 1MII1 发送数据 3; RMII1 模式下, 不连接。
BTXEN	O	50M (RMII) 25M (MII)	端口 1MII1/RMII1 发送数据使能。
BTXERR	O	25M	端口 1MII1 发送数据错误; RMII1 模式下, 不连接。
BTXCK	I	25M	端口 1MII1 发送数据时钟; RMII1 模式下, 不连接。
BRXD0	I	50M (RMII) 25M (MII)	端口 1MII1 接收数据 0 或 RMII1 接收数据 0。
BRXD1	I	50M (RMII) 25M (MII)	端口 1MII1 接收数据 1 或 RMII1 接收数据 1。



信号名	方向	频率 (Hz)	描述
BRXD2	I	25M	端口 1MII1 接收数据 2; RMII1 模式下, 连接 GND。
BRXD3	I	25M	端口 1MII1 接收数据 3; RMII1 模式下, 连接 GND。
BRXDV	I	50M (RMII) 25M (MII)	端口 1MII1/RMII1 接收数据有效。
BRXERR	I	50M (RMII) 25M (MII)	端口 1MII1/RMII1 接收数据错误。
BRXCK	I	25M	端口 1MII1 接收数据时钟; RMII1 模式下, 连接 GND。
BCRS	I	25M	端口 1MII1 载波侦听; RMII1 模式下, 连接 GND。
BCOL	I	25M	端口 1MII1 碰撞指示; RMII1 模式下, 连接 GND。
MDCK	O	2M	PHY 管理接口时钟输出。
MDIO	I/O	2M	PHY 管理接口数据输入输出, 需上拉处理。

18.4 工作方式

本节描述了 SF 的接口时序、初始化配置、中断管理和 CPU 端口收发帧。

18.4.1 接口时序

MII 接口时序

MII 接口接收时序和 MII 接口发送时序如图 18-2、图 18-3 所示。

图18-2 MII 接口接收时序图

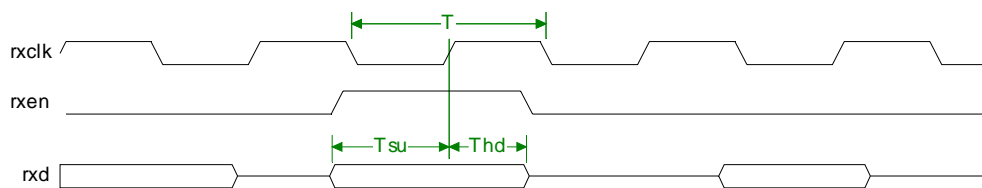




图18-3 MII 接口发送时序图

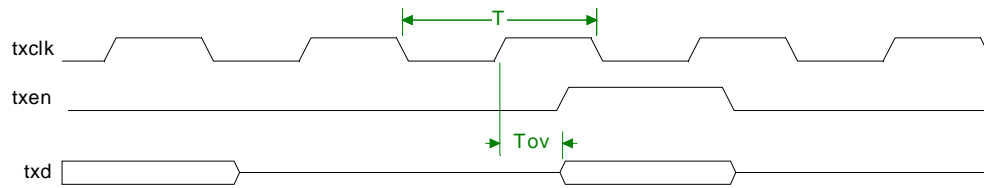


表18-2 MII 接口的时序参数

参数	符号	信号	最小值	最大值	单位
MII 信号建立时间	Tsu (RX)	RXER、RXEN、RXD[3:0]	6	-	ns
MII 信号保持时间	Thd (RX)	RXER、RXEN、RXD[3:0]	2	-	ns
MII 输出信号延时	Tov (MIITX)	TXD[1:0]、TXEN	2	8	ns

RMII 接口时序

RMII 接口时序如图 18-4 所示。

图18-4 RMII 接口时序图

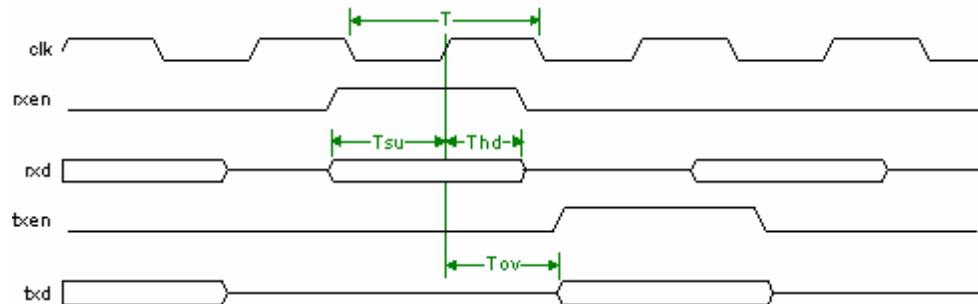


表18-3 RMII 接口的时序参数

参数	符号	信号	最小值	最大值	单位
RMII 信号建立时间	Tsu (RX)	RXER、RXEN、RXD[1:0]	4	-	ns

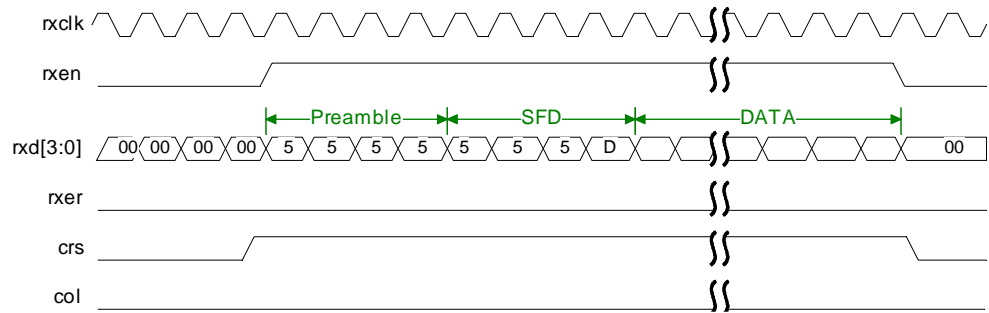


参数	符号	信号	最小值	最大值	单位
RMII 信号保持时间	Thd (RX)	RXER、RXEN、RXD[1:0]	2	-	ns
RMII 输出信号延时	Tov (RMIITX)	TXD[1:0]、TXEN	2	8	ns

100Mbit/s MII 接口时序

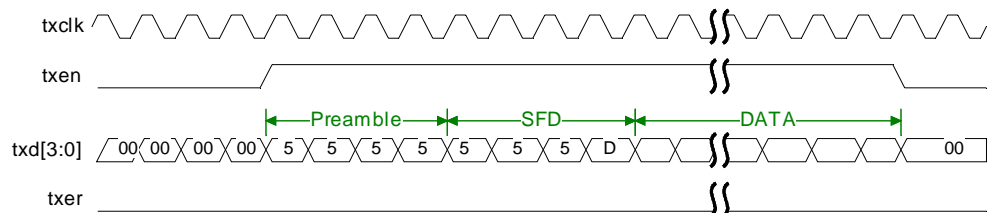
- 接收时序

图18-5 100Mbit/s MII 接口接收时序图



- 发送时序

图18-6 100Mbit/s MII 接口发送时序图

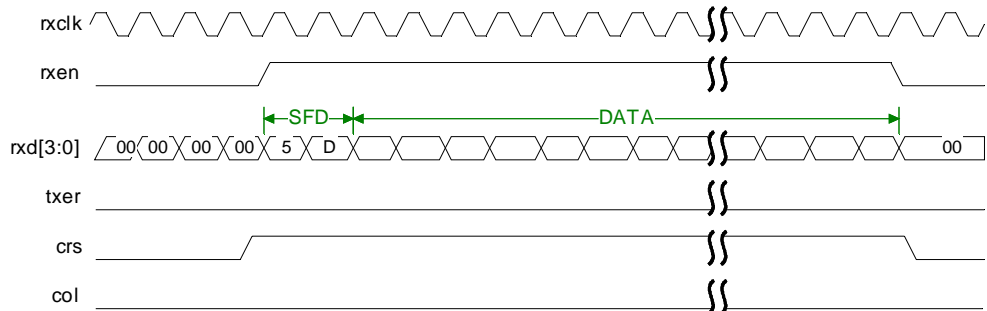


10Mbit/s MII 接口时序

- 接收时序

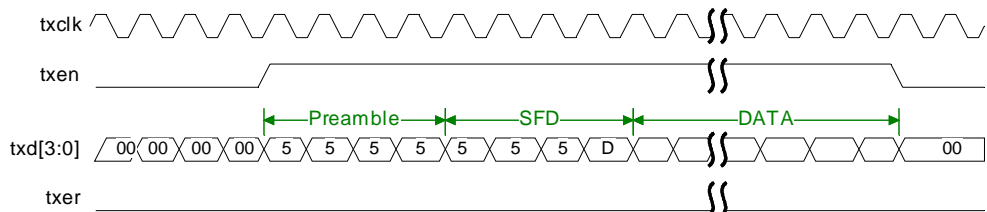


图18-7 10Mbit/s MII 接口接收时序图



- 发送时序

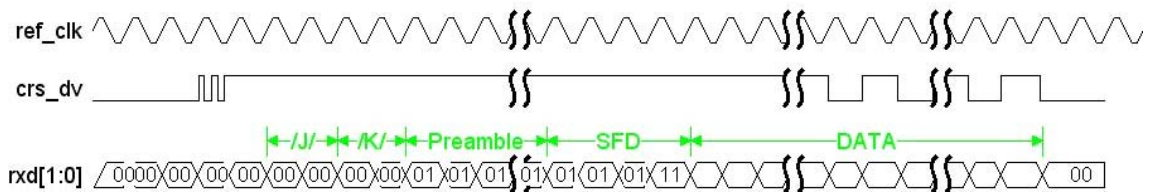
图18-8 10Mbit/s MII 接口发送时序图



100Mbit/s RMII 接口时序

- 接收时序

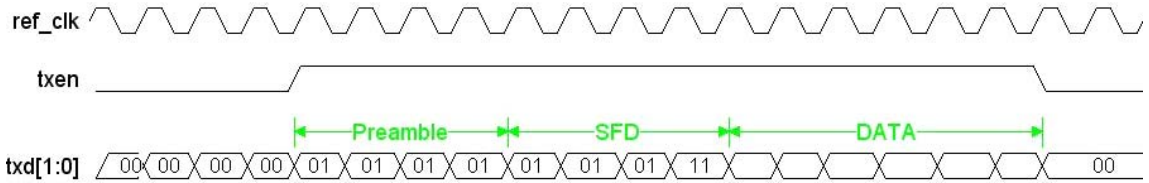
图18-9 100Mbit/s RMII 接口接收时序图



- 发送时序



图18-10 100Mbit/s RMII 接口发送时序图



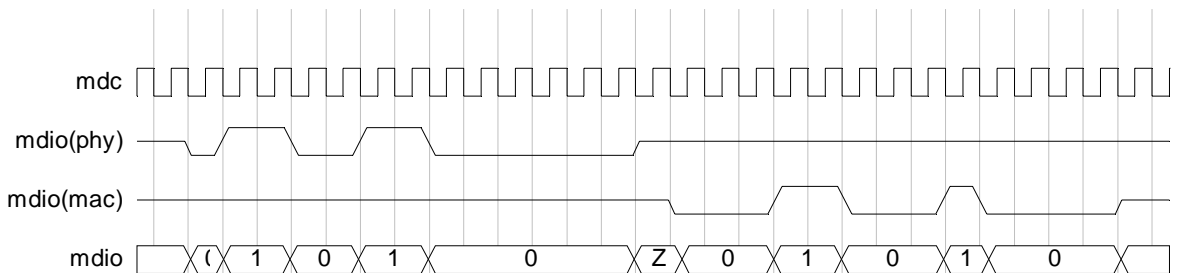
10Mbit/s RMII 接口时序

10M bit/s RMII `ref_clk` 为 50MHz, 与 100Mbit/s RMII 接口的区别为: `ref_clk` 每 10 个时钟采样一次。时序图略。

MDIO 接口时序

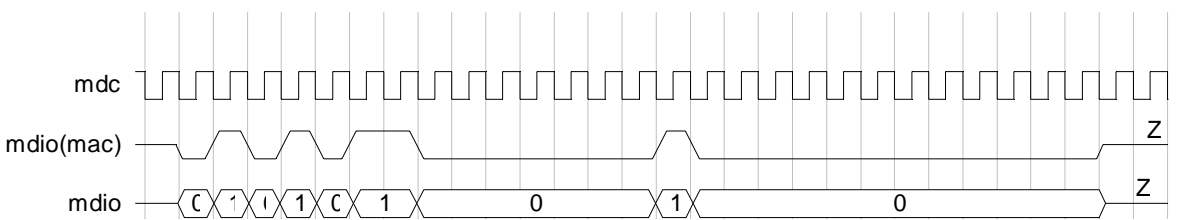
- MDIO 读时序

图18-11 MDIO 读时序图



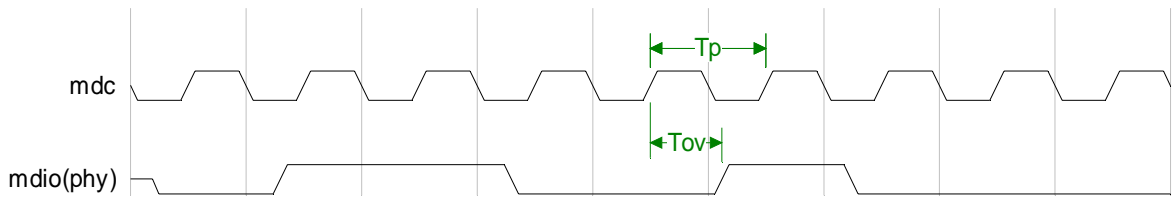
- MDIO 写时序

图18-12 MDIO 写时序图



- MDIO 接收时序

图18-13 MDIO 接收时序图



Tov 与 mdc 时钟周期 Tp 相关，可通过调整 MDC 频率进行改变，约为 Tp/2。

表18-4 MDIO 接收时序参数表

参数	符号	信号	最小值	最大值	单位
MDIO 接收数据延迟时间	Tov	MDIO	Tp/2	Tp	ns

- MDIO 发送时序

图18-14 MDIO 发送时序图

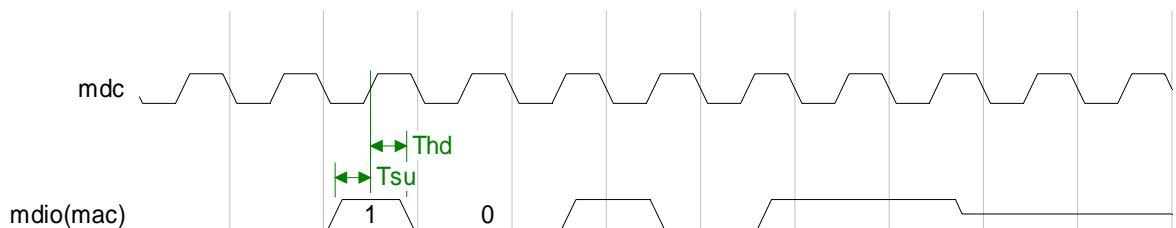


表18-5 MDIO 发送时序参数表

参数	符号	信号	最小值	最大值	单位
MDIO 发送数据建立时间	Tsu	MDIO	400	-	ns
MDIO 发送数据保持时间	Thd	MDIO	50	-	ns

18.4.2 SF 初始化配置

SF 需要进行一系列初始化配置之后才能正常工作，具体包括配置 SF 的工作模和配置本机信息等。关于本节描述涉及到的寄存器，请参见“18.6 寄存器描述”。

配置端口

- 配置端口状态获取方式



SF_MAC_REG2[stat_ctrl]是状态控制选择。通过配置此寄存器，可令 SF 通过 MDIO 接口扫描 PHY 寄存器获取端口状态或者由 CPU 配置端口状态。

端口状态主要包括：

- 端口连接状态；
- 速度模式；
- 双工模式。

- 配置自适应

配置自适应寄存器 (**SF_MDIO_REG4**, **SF_MDIO_REG5**)：当端口工作在自适应获得状态信息时，需要完成如下工作：

- 指定所需扫描的 PHY 状态位所在寄存器的位置；
- 指定所需扫描的 PHY 状态位在寄存器中的 bit 位置。

如 PHY 自协商结果的连接状态位位于 PHY 地址为 1 的寄存器的第 2bit

(link_status, Register1.2)，需配置 **SF_MDIO_REG4**[internal_addr_link]、

SF_MDIO_REG5[internal_addr_link]为 1，**SF_MDIO_REG4**[link_index]、

SF_MDIO_REG5[link_index]为 2，则 SF 会自动从 Register1.2 获取连接状态信息。

- 配置端口工作方式 (可选)

- **SF_MAC_REG4**：配置端口状态。

仅在端口连接状态由 CPU 配置时起作用，可通过此寄存器配置端口的连接状态、速度模式、双工模式。

- **SF_GLB_REG18**：配置 MAC 接口数据格式。

2 个端口可独立配置在 MII 接口或 RMII 接口模式下。

- 配置 PHY 工作模式 (可选)

- 配置 PHY 外部地址 (**SF_MDIO_REG2**)：

配置 2 个端口 PHY 外部物理地址。

- 配置 PHY 工作状态 (**SF_MDIO_REG0**)：

SF 提供灵活的 PHY 配置方式，使用 **SF_MDIO_REG0** 可以对 2 个端口的 PHY 所有寄存器进行读写控制。配置时需参考写 PHY 寄存器说明和读 PHY 寄存器说明。

- 写 PHY 寄存器说明

写入写寄存器指令后可查询 **SF_MDIO_REG0**[finish]位。当 finish 位被置为 1，表示 CPU 数据已写入指定的 PHY 寄存器中。

表18-6 寄存器 SF_MDIO_REG0 中对比特的描述 (写 PHY 寄存器)

比特位	描述
cpu_data_in	CPU 待写入 PHY 相应寄存器的数据。
finish	写入 0 表示启动操作。
rw	写入 1 表示此次操作为写。



phy_exaddr	为待写入端口 PHY 芯片的外部地址。
frq_dv	MDIO 接口时钟分频系数。
phy_inaddr	此次操作的寄存器对应 PHY 芯片的内部地址。

- 读 PHY 寄存器说明

写入读指令后可查询 SF_MDIO_REG0[finish]位，当 finish 位被置为 1，表示此次读操作已完成，读出的数据放在 MDIO_Reg1 中，可供 CPU 读取。

表18-7 寄存器 SF_MDIO_REG0 中对比特的描述（读 PHY 寄存器）

比特位	描述
cpu_data_in	不使用。
finish	写入 0 表示启动操作。
rw	写入 0 表示此次操作为读。
phy_exaddr	待写入端口 PHY 芯片的外部地址。
frq_dv	MDIO 接口时钟分频系数。
phy_inaddr	此次操作的寄存器对应 PHY 芯片的内部地址。

配置 SF 工作模式

- 配置 SF_GLB_REG11，使 SF 工作在普通模式。
- 配置 SF_GLB_REG11，使 SF 工作在监听模式并指定监听端口。

配置本机地址信息

配置 SF_GLB_REG4 和 SF_GLB_REG5，设定本机 MAC 地址。

缓存使用区域配置

- 配置 SF_GLB_REG17，设定 SF 缓存帧数据时使用 SDRAM 空间的起始地址。
- 配置 SF_BM_REG17，设定 SF 缓存 SDRAM 数据的大小端格式。

特殊帧配置

- 配置 SF_IQM_REG0 和 SF_IQM_REG1，设定上行口和下行口广播帧转发方式。
- 配置 SF_IQM_REG0 和 SF_IQM_REG1，设定上行口和下行口特殊目的 MAC 帧转发方式。



- 配置 SF_IQM_REG0 和 SF_IQM_REG1，设定上行口和下行口 ip 多播帧转发方式。
- 配置 SF_IQM_REG0 和 SF_IQM_REG1，设定上行口和下行口非特殊帧转发方式。
- 配置 SF_IQM_REG0 和 SF_IQM_REG1，设定上行口和下行口所有帧强制转发方式。
- 配置 SF_IQM_REG2，设定上行口和下行口进入的帧是否允许发往 CPU 端口和另外一个外部端口。
- 配置 ANYPORT 表，设定上行口和下行口需要进行识别的特定协议类型及目的 TCP/UDP 端口号的帧转发方式。
- 配置 ANYTYPE 表，设定上行口和下行口需要进行识别的特定目的 MAC、VLANID 和以太网类型帧的转发方式。

VLAN 配置

- 配置 SF_IQM_REG4，设定 SF 是否使能 VLAN。
- 配置 SF_IQM_REG4，设定 SF 转发是否限制在 VLAN 域中。
- 配置 VLAN 表，设定 SF 识别的 VLAN ID 及各 VLAN 对应 member 及各端口 VLAN 输出格式。
- 配置 SF_IERF_REG0，设定 2 个外部端口的端口 VLAN ID。
- 配置 SF_IQM_REG6，设定 2 个外部端口是否接收未知 VLAN 及未知 VLAN 转发方式和 tag 格式。
- 配置 SF_IQM_REG7，设定 2 个外部端口是否接收未知 VLAN 成员帧及未知 VLAN 成员帧转发方式。
- 端口 vlanmode 配置。配置 SF_IQM_REG5，设定端口 VLANMODE。
- 设定端口 tag 格式控制。配置 SF_IQM_REG3 和 SF_IERF_REG1，设定端口输出 tag 格式。

其他配置

- 配置 SF_MAC_REG0 和 SF_MAC_REG1，设定下行口和上行口接收最短帧前导码长度、输出最小帧间隙 (ipg) 和发送流控帧的帧间隔。建议使用缺省配置，在端口工作在 10M 模式时，可将最短帧前导码长度配置为 0 或 1。
- 配置 SF_MAC_REG3，对上行口和下行口进行端口复位。端口复位可以对上行口或下行口进行复位，使该端口相关的寄存器和缓存回到初始化状态。
- 配置 SF_STS_REG0 和 SF_STS_REG4，设定端口外环回。
- 配置 SF_STS_REG0 和 SF_STS_REG4，设定端口输出 CRC 是否重新计算。
- 配置 SF_STS_REG0 和 SF_STS_REG4，设定端口是否允许发送流控帧。
- 配置 SF_STS_REG0 和 SF_STS_REG4，设定端口是否允许接收超短帧及允许接收最小帧长。
- 配置 SF_STS_REG0 和 SF_STS_REG4，设定端口允许接收最大帧长。



- 配置 [SF_IERF_REG2](#)，配置 2 个端口是否修改 802.11p 优先级，及各端口对应优先级。

配置

- [SF_BM_REG1](#)，设定队列流控及解除流控的队列阈值。

18.4.3 SF 中断管理

中断寄存器

指示 SF 产生的中断类型。每位对应一种中断类型，请参见“[SF_GLB_REG0](#)”。

中断使能寄存器

每位对应一种中断类型 ([SF_GLB_REG1](#))，设定是否允许产生对应类型的中断。

中断输出屏蔽寄存器

每位对应一种类型的中断 ([SF_GLB_REG2](#))，设定对应类型的中断产生后是否允许发送中断通知 CPU。

18.4.4 CPU 端口收发帧

CPU 收帧

- 中断方式
CPU 可以打开 `int_cpu_rx` 通过中断收帧。此时当 SF 有帧需要 CPU 接收时，向 CPU 中断通知软件收帧。
- 查询方式
SF 使用 `bm_reg8` 指示相应队列是否有帧需要 CPU 接收。
- 收包过程
 - 查询 `bm_reg8` 获取收帧队列；
 - 读取 `bm_reg4~bm_reg6` 中选定队列的待收帧描述子；
 - 根据帧描述子提供的帧信息，从 SDRAM 取帧数据；
 - 收数据完成后，写 `bm_reg3` 对应队列的 bit 通知该队列收一帧完毕。

表18-8 CPU 接收帧描述子数据结构

比特	名称	描述
[31]	Reserved	保留。
[30]	tag_out	当前帧所带 tag 数据是否有效，当 tag_out 为 0 时表示当前帧所带 tag 无意义。
[29:19]	frame_len	帧的长度，单位为字节。



比特	名称	描述
[18:17]	spn	帧转发目的端口和源端口指示： 2'b00: 上行口发给 CPU 的单播报文； 2'b01: 上行口发给 CPU 和下行口的广播报文； 2'b10: 下行口发给 CPU 的单播报文； 2'b11: 下行口发给 CPU 和上行口的广播报文。
[16:15]	tag_type	bit16: 0: 不带 tag 的帧； 1: 带 tag 的帧。 在 bit16 为 1 时, bit15: 0: 带优先级的帧； 1: 带有效 VLAN ID 的帧。 在 bit16 为 1 时, bit15 无意义。
[14:8]	frame_type	当前帧类型指示。 7'h0~7'h3f: ANYPORT 帧, 低 5 位标志所属 ANYPORT 帧类型索引； 7'h40~7'h47: ANYTYPE 帧, 低 3 位标志所属 ANYTYPE 帧类型索引； 7'h48: 广播帧； 7'h50: SPCL 帧； 7'h58: IGMP 帧； 7'h60: IPM 帧； 7'h68: Normal IP 帧； 7'h78: 其他类型帧； 其他: 保留, 无意义。
[7:0]	fba	表示帧在片外缓存的地址索引。 帧在 SDRAM 缓存的地址可由如下方式计算： $start_address + fba \times 2048$ 。 其中 start_address 为 SF 被分配使用的 SDRAM 中帧缓存区域的起始地址。

- 查询接收队列长度
读 [SF_BM_REG16](#) 可获取当前 SF 中各队列待接收帧的数量。

CPU 发帧

- 中断方式



CPU 可以打开 `int_free_cpu` 通过中断发帧，此时当 SF 有空间接收 CPU 发送帧数据时，向 CPU 发中断通知软件发帧。

- 查询方式

CPU 读 `SF_BM_REG9~SF_BM_REG11` 判断各队列是否有空间接收 CPU 发出的帧。

- 发包过程

- 查询 `SF_BM_REG9~SF_BM_REG11` 判断 SF 是否有空间接收 CPU 端口发出的帧。

- 根据 `SF_BM_REG9~SF_BM_REG11` 提供的尾地址信息计算帧在 SDRAM 中的缓存地址。

当前帧缓存地址 = SF 使用缓存区首地址 + 帧描述子中地址 × 2kB

将帧数据及帧描述子写入到该缓存地址中。

以太网帧存储帧格式：

帧描述子	目的 MAC	源 MAC	Type	Data	FCS
------	--------	-------	------	------	-----

CPU 发出的帧在 SDRAM 中缓存时，必须包括帧描述子。

- 写帧描述子到 `bm_reg2` 通知 SF 将该帧入队；

- 查询队列使用空间。查询 `bm_reg16` 可获得当前 CPU 发送队列使用情况。

表18-9 CPU 发送帧描述子数据结构

名称	比特	描述
M/U	[31]	单播或广播指示/特殊帧指示。
tag_out	[30]	当前帧所带 tag 数据是否有效，当 tag_out 为 0 时表示需要 SF 将 tag 域数据剥掉之后转发，否则直接带 tag 输出。
fmlen	[29:19]	帧的长度，单位为字节。
spn	[18:14]	帧转发目的端口和源端口指示。 bit18: CPU 端口为源端口，此时必须为 1； bit17: 下行口为目的端口； bit16: 上行口为目的端口； bit15: 无意义，此时须为 0； bit14: 无意义，此时须为 0。
ipm	[13]	表示是否 IP 多播和特殊帧。



名称	比特	描述
tag_type	[12:11]	bit12: 0: 不带 tag 的帧; 1: 带 tag 的帧。 bit12 为 1 时, bit11: 0: 带优先级的帧; 1: 带有效 VLAN ID 的帧。 bit12 为 0 时, bit11 无意义。
fba	[10:0]	表示帧在片外缓存的偏移地址。

18.5 寄存器概览

以太网交换寄存器的地址位宽 32 位, 地址范围: 0xA002_0000~0xA002_FFFF, 以太网交换单元寄存器如表 18-10 所示。

表18-10 以太网交换单元寄存器概览 (基址是 0xA002_0000)

偏移地址	名称	描述	页码
0x0000~0x01FC	SF_STATIS	外部端口 RMON/SNMP 统计计数结果	18-65
0x0200~0x02FC	SF_ANYPORT	ANYPORT 帧配置表	18-73
0x0300~0x037C	SF_ANYTYPE	ANYTYPE 帧配置表	18-74
0x0380~0x039C	SF_VLAN	VLAN 表	18-75
0x0400~0x0450	SF_BM_REG[0:17]	BM 管理寄存器组	18-55
0x0480~0x04A4	SF_IQM_REG[0:10]	IQM 管理寄存器组	18-49
0x0500~0x054C	Reserved	保留	-
0x0580~0x059C	SF_STS_REG[0:7]	统计计数寄存器组	18-22
0x05A0~0x05BC	SF_MAC_REG[0:6]	MAC 寄存器组	18-19
0x05C0~0x05D4	SF_MDIO_REG[0:5]	MDIO 寄存器组	18-35
0x05E0~0x05E8	SF_IERF_REG[0:2]	IRF 和 ERF 寄存器组	18-39
0x0600~0x0668	SF_GLB_REG[0:26]	全局寄存器组	18-41



18.6 寄存器描述

本节详细描述了以太网交换单元的寄存器。

18.6.1 MAC 寄存器组

SF_MAC_REG0

- 偏移地址：0x05A0
- 操作类型：R/W
- 复位值：0x1E0_1FFF
- 复位方式：h/s

比特	名称	描述
[31:26]	Reserved	保留。
[25:23]	pre_cnt_limit_0	下行口（端口 0）MAC 连续检测到帧前导码的个数。
[22:16]	ipg_0	下行口发送帧间隙控制，以 bit 为单位。
[15:0]	para_0	下行口输出流控帧间隔参数。

说明

- 10M 模式下，pre_cnt_limit 建议配置为 0；
- 当端口处于连接状态时，在修改 ipg 配置后，需对相应端口进行端口复位。

SF_MAC_REG1

- 偏移地址：0x05A4
- 操作类型：R/W
- 复位值：0x1E0_1FFF
- 复位方式：h/s

比特	名称	描述
[31:26]	Reserved	保留。
[25:23]	pre_cnt_limit_1	上行口（端口 1）MAC 连续检测到帧前导码的个数。
[22:16]	ipg_1	上行口（端口 1）发送帧间隙控制，以 bit 为单位。
[15:0]	para_1	上行口（端口 1）输出流控帧间隔参数。



SF_MAC_REG2

- 偏移地址: 0x05A8
- 操作类型: R/W
- 复位值: 0xA
- 复位方式: h/s

比特	名称	描述
[31:4]	Reserved	保留。
[3:0]	stat_ctrl	外部端口工作状态信息选择控制寄存器。 <ul style="list-style-type: none"> • 2'h2: 使用 CPU 设定的状态信息; • 其他: 使用从 MDIO 接口获得的状态信息。 bit[3:2]对应上行口, bit[1:0]对应下行口。

SF_MAC_REG3

- 偏移地址: 0x05AC
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:2]	Reserved	保留。
[1:0]	reset_port	端口复位。 0: 解除端口复位; 1: 将对应端口进行复位。 bit1 对应上行口, bit0 对应下行口。

SF_MAC_REG4

- 偏移地址: 0x05B0
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:6]	Reserved	保留。



[5:4]	speed_stat_dio	CPU 设定外部端口速度模式。 0: 10M 模式; 1: 100M 模式。 bit5 对应上行口, bit4 对应下行口。
[3:2]	link_stat_dio	CPU 设定链接状态。 0: 无链接; 1: 链接正常。 bit3 对应上行口, bit2 对应下行口。
[1:0]	duplex_stat_dio	CPU 设定双工模式。 0: 半双工模式; 1: 全双工模式。 bit1 对应上行口, bit0 对应下行口。

SF_MAC_REG5

- 偏移地址: 0x05B4
- 操作类型: R
- 复位值: 0x40
- 复位方式: h/s

比特	名称	描述
[31:6]	Reserved	保留。
[5:4]	speed_stat	当前速度模式。 0: 10M 模式; 1: 100M 模式。 bit5 对应上行口, bit4 对应下行口。
[3:2]	link_stat	当前链接状态。 0: 无链接; 1: 链接正常。 bit3 对应上行口, bit2 对应下行口。
[1:0]	duplex_stat	当前双工模式。 0: 半双工模式; 1: 全双工模式。 bit1 对应上行口, bit0 对应下行口。



SF_MAC_REG6

- 偏移地址: 0x05B8
- 操作类型: RWC
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:6]	Reserved	保留。
[5:4]	speed_stat_ch	速度模式改变指示信号。 0: 速度模式没有发生过变化; 1: 速度模式发生过变化。 bit5 对应上行口, bit4 对应下行口, 写清零。
[3:2]	link_stat_ch	链接状态改变指示信号。 0: 连接状态没有发生过变化; 1: 连接状态发生过变化。 bit3 对应上行口, bit2 对应下行口, 写清零。
[1:0]	duplex_stat_ch	双工模式改变指示信号。 0: 双工模式没有发生过变化; 1: 双工模式发生过变化。 bit1 对应上行口, bit0 对应下行口, 写清零。

18.6.2 统计计数寄存器组

SF_STS_REG0

- 偏移地址: 0x0580
- 复位值: 0x27_55EE
- 复位方式: h/s

比特	名称	操作类型	描述
[31]	Reserved	-	保留。
[30]	crc_rx_0	R	端口 0 (下行口) 接收帧 CRC 错指示。 0: 接收帧CRC正常; 1: 接收帧 CRC 出错。



比特	名称	操作类型	描述
[29]	txcrc_bad_0	R	端口 0 (下行口) 发送帧 CRC 错指示。 0: 发送帧 CRC 正常; 1: 发生帧 CRC 出错。
[28]	crcgen_dis_0	R/W	端口 0 (下行口) CRC 生成禁止控制寄存器。 0: 输出帧 CRC 硬件重新计算; 1: 输出帧不计算 CRC。
[27]	cntr_rdclr_en_0	R/W	下行口统计计数器读清空使能信号。 0: 统计计数读清空不使能; 1: 统计计数读清空使能。
[26]	cntr_clr_all_0	R/W	下行口统计计数器清空信号。 0: 不清空 (缺省状态); 1: 清空统计计数。
[25]	cntr_roll_dis_0	R/W	下行口统计计数环回使能信号。 0: 允许统计计数环回; 1: 禁止统计计数环回。
[24:21]	colthreshold_0	R/W	下行口冲突次数阈值统计, 缺省为 1, 表示 dot3colcnt_0 针对发送出现一次冲突的帧进行计数 (请参见表 18-11 中 dot3colcnt_0 描述)。
[20]	Reserved	R/W	保留。
[19]	ex_loop_en_0	R/W	下行口外环回使能信号。 0: 外环回禁止; 1: 外环回使能。
[18]	pause_en_0	R/W	下行口流控帧发送使能信号。 0: 禁止端口发送流控帧 (半双工 JAM 序列); 1: 允许端口发送流控帧 (半双工 JAM 序列)。
[17]	rx_shframe_en0	R/W	下行口短帧接收使能信号。 0: 禁止接收短帧; 1: 允许接收短帧。
[16:11]	rx_min_thr_0	R/W	下行口允许接收的最小帧长, 单位为字节。
[10:0]	len_max_0	R/W	下行口允许接收的最大帧长, 单位为字节。



SF_STS_REG1

- 偏移地址: 0x0584
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:28]	rxsof_cnt_0	下行口接收到的帧头计数。
[27:24]	rxeof_cnt_0	下行口接收到的帧尾计数。
[23:20]	Rxcrcok_cnt_0	下行口接收到 CRC 校验正确的帧计数。
[19:16]	rxcrbad_cnt_0	下行口接收到 CRC 校验错误的帧计数。
[15:12]	txsof_cnt_0	下行口发送的帧头计数。
[11:8]	txeof_cnt_0	下行口发送的帧尾计数。
[7:4]	Txcrcok_cnt_0	下行口发送 CRC 校验正确的帧计数。
[3:0]	txcrbad_cnt_0	下行口发送 CRC 校验错误的帧计数。

SF_STS_REG2

下行口接收统计计数半满指示信号。

- 偏移地址: 0x0588
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:30]	Reserved	保留。
[29]	cntr_stat_rx[29]	0: ifinoctets 计数值不足一半; 1: ifinoctets 计数值已超过一半。
[28]	cntr_stat_rx[28]	0: pkts 计数值不足一半; 1: pkts 计数值已超过一半。
[27]	cntr_stat_rx[27]	0: octets 计数值不足一半; 1: octets 计数值已超过一半。
[26]	cntr_stat_rx[26]	0: dot3dribble 计数值不足一半; 1: dot3dribble 计数值已超过一半。



比特	名称	描述
[25]	cntr_stat_rx[25]	0: dot3unkpause 计数值不足一半; 1: dot3unkpause 计数值已超过一半。
[24]	cntr_stat_rx[24]	0: dot3inpause 计数值不足一半; 1: dot3inpause 计数值已超过一半。
[23]	cntr_stat_rx[23]	0: dot3internalrecerr 计数值不足一半; 1: dot3internalrecerr 计数值已超过一半。
[22]	cntr_stat_rx[22]	0: dot3fcserr 计数值不足一半; 1: dot3fcserr 计数值已超过一半。
[21]	cntr_stat_rx[21]	0: dot3alignmenterr 计数值不足一半; 1: dot3alignmenterr 计数值已超过一半。
[20]	cntr_stat_rx[20]	0: ifinbroadcast 计数值不足一半; 1: ifinbroadcast 计数值已超过一半。
[19]	cntr_stat_rx[19]	0: ifinmulticast 计数值不足一半; 1: ifinmulticast 计数值已超过一半。
[18]	cntr_stat_rx[18]	0: ifinerrors 计数值不足一半; 1: ifinerrors 计数值已超过一半。
[17]	cntr_stat_rx[17]	0: ifindiscards 计数值不足一半; 1: ifindiscards 计数值已超过一半。
[16]	cntr_stat_rx[16]	0: ifinnucastpkts 计数值不足一半; 1: ifinnucastpkts 计数值已超过一半。
[15]	cntr_stat_rx[15]	0: ifinucastpkts 计数值不足一半; 1: ifinucastpkts 计数值已超过一半。
[14]	cntr_stat_rx[14]	0: pkts_1518 计数值不足一半; 1: pkts_1518 计数值已超过一半。
[13]	cntr_stat_rx[13]	0: pkts1024_1518 计数值不足一半; 1: pkts1024_1518 计数值已超过一半。
[12]	cntr_stat_rx[12]	0: pkts512_1023 计数值不足一半; 1: pkts512_1023 计数值已超过一半。
[11]	cntr_stat_rx[11]	0: pkts256_511 计数值不足一半; 1: pkts256_511 计数值已超过一半。



比特	名称	描述
[10]	cntr_stat_rx[10]	0: pkts128_255 计数值不足一半; 1: pkts128_255 计数值已超过一半。
[9]	cntr_stat_rx[9]	0: pkts65_127 计数值不足一半; 1: pkts65_127 计数值已超过一半。
[8]	cntr_stat_rx[8]	0: pkts64 计数值不足一半; 1: pkts64 计数值已超过一半。
[7]	cntr_stat_rx[7]	0: multicastpkts 计数值不足一半; 1: multicastpkts 计数值已超过一半。
[6]	cntr_stat_rx[6]	0: broadcastpkts 计数值不足一半; 1: broadcastpkts 计数值已超过一半。
[5]	cntr_stat_rx[5]	0: jabber 计数值不足一半; 1: jabber 计数值已超过一半。
[4]	cntr_stat_rx[4]	0: fragments 计数值不足一半; 1: fragments 计数值已超过一半。
[3]	cntr_stat_rx[3]	0: oversizepkts 计数值不足一半; 1: oversizepkts 计数值已超过一半。
[2]	cntr_stat_rx[2]	0: undersizepkts 计数值不足一半; 1: undersizepkts 计数值已超过一半。
[1]	cntr_stat_rx[1]	0: crcerr 计数值不足一半; 1: crcerr 计数值已超过一半。
[0]	cntr_stat_rx[0]	0: dropevents 计数值不足一半; 1: dropevents 计数值已超过一半。

SF_STS_REG3

下行口发送计数半满指示信号。

- 偏移地址: 0x058C
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s



比特	名称	描述
[31:30]	Reserved	保留。
[29]	cntr_stat_tx[29]	0: dot3outpause 计数值不足一半; 1: dot3outpause 计数值已超过一半。
[28]	cntr_stat_tx[28]	0: ifoutoctets 计数值不足一半; 1: ifoutoctets 计数值已超过一半。
[27]	cntr_stat_tx[27]	0: collisions 计数值不足一半; 1: collisions 计数值已超过一半。
[26]	cntr_stat_tx[26]	0: octets_tx 计数值不足一半; 1: octets_tx 计数值已超过一半。
[25]	cntr_stat_tx[25]	0: pkts_tx 计数值不足一半; 1: pkts_tx 计数值已超过一半。
[24]	cntr_stat_tx[24]	0: dot3defer 计数值不足一半; 1: dot3defer 计数值已超过一半
[23]	cntr_stat_tx[23]	0: dot3colcnt 计数值不足一半; 1: dot3colcnt 计数值已超过一半。
[22]	cntr_stat_tx[22]	0: dot3excessivecol 计数值不足一半; 1: dot3excessivecol 计数值已超过一半。
[21]	cntr_stat_tx[21]	0: dot3latecol 计数值不足一半; 1: dot3latecol 计数值已超过一半。
[20]	cntr_stat_tx[20]	0: dot3multiplecol 计数值不足一半; 1: dot3multiplecol 计数值已超过一半。
[19]	cntr_stat_tx[19]	0: dot3singlecol 计数值不足一半; 1: dot3singlecol 计数值已超过一半。
[18]	cntr_stat_tx[18]	0: pkts_1518_tx 计数值不足一半; 1: pkts_1518_tx 计数值已超过一半。
[17]	cntr_stat_tx[17]	0: pkts1024_1518_tx 计数值不足一半; 1: pkts1024_1518_tx 计数值已超过一半。
[16]	cntr_stat_tx[16]	0: pkts512_1023_tx 计数值不足一半; 1: pkts512_1023_tx 计数值已超过一半。
[15]	cntr_stat_tx[15]	0: pkts256_511_tx 计数值不足一半; 1: pkts256_511_tx 计数值已超过一半



比特	名称	描述
[14]	cntr_stat_tx[14]	0: pkts128_255_tx 计数值不足一半; 1: pkts128_255_tx 计数值已超过一半。
[13]	cntr_stat_tx[13]	0: pkts65_127_tx 计数值不足一半; 1: pkts65_127_tx 计数值已超过一半。
[12]	cntr_stat_tx[12]	0: pkts64_tx 计数值不足一半; 1: pkts64_tx 计数值已超过一半。
[11]	cntr_stat_tx[11]	0: ifoutbroadcast 计数值不足一半; 1: ifoutbroadcast 计数值已超过一半。
[10]	cntr_stat_tx[10]	0: ifoutmulticast 计数值不足一半; 1: ifoutmulticast 计数值已超过一半。
[9]	cntr_stat_tx[9]	0: ifoutnucastpkts 计数值不足一半; 1: ifoutnucastpkts 计数值已超过一半。
[8]	cntr_stat_tx[8]	0: ifoutucastpkts 计数值不足一半; 1: ifoutucastpkts 计数值已超过一半。
[7]	cntr_stat_tx[7]	0: ifouterrors 计数值不足一半; 1: ifouterrors 计数值已超过一半。
[6]	cntr_stat_tx[6]	0: ifoutdiscards 计数值不足一半; 1: ifoutdiscards 计数值已超过一半。
[5]	cntr_stat_tx[5]	0: multicastpkts_tx 计数值不足一半; 1: multicastpkts_tx 计数值已超过一半。
[4]	cntr_stat_tx[4]	0: broadcastpkts_tx 计数值不足一半; 1: broadcastpkts_tx 计数值已超过一半。
[3]	cntr_stat_tx[3]	0: oversizepkts 计数值不足一半; 1: oversizepkts 计数值已超过一半。
[2]	cntr_stat_tx[2]	0: undersizepkts 计数值不足一半; 1: undersizepkts 计数值已超过一半。
[1]	cntr_stat_tx[1]	0: crcerr 计数值不足一半; 1: crcerr 计数值已超过一半。
[0]	cntr_stat_tx[0]	0: dropevents 计数值不足一半; 1: dropevents 计数值已超过一半。



SF_STS_REG4

- 偏移地址：0x0590
- 复位值：0x27_55EE
- 复位方式：h/s

比特	名称	操作类型	详细描述
[31]	Reserved	-	保留。
[30]	crc_rx_1	R	端口 1（上行口）接收帧 CRC 错指示。 0: 接收帧CRC正确； 1: 接收帧 CRC 出错。
[29]	txcrc_bad_1	R	端口 1（上行口）发送帧 CRC 错指示。
[28]	crcgen_dis_1	R/W	端口 1（上行口）CRC 生成禁止控制寄存器。 0: 输出帧 CRC 硬件重新计算； 1: 输出帧不计算 CRC。
[27]	cntr_rdcclr_en_1	R/W	上行口统计计数器读清空使能信号。 0: 统计计数读清空不使能； 1: 统计计数读清空使能。
[26]	cntr_clr_all_1	R/W	上行口统计计数器清空信号。 0: 不清空，缺省状态； 1: 清空统计计数。
[25]	cntr_roll_dis_1	R/W	上行口统计计数环回使能信号。 0: 允许统计计数环回； 1: 禁止统计计数环回。
[24:21]	colthreshold_1	R/W	上行口冲突次数阈值统计，缺省为 1，表示 dot3colcnt_1 针对发送出现一次冲突的帧的个数（请参见表 18-11 中 dot3colcnt_1 描述）。
[20]	Reserved	R/W	保留。
[19]	ex_loop_en_1	R/W	上行口外环回使能信号。 0: 外环回禁止； 1: 外环回使能。
[18]	pause_en_1	R/W	上行口流控帧发送使能信号。 0: 禁止端口发送流控帧（半双工 JAM 序列）； 1: 允许端口发送流控帧（半双工 JAM 序列）。



比特	名称	操作类型	详细描述
[17]	rx_shframe_en_1	R/W	上行口短帧接收使能信号。 0: 禁止接收短帧; 1: 允许接收短帧。
[16:11]	rx_min_thr_1	R/W	上行口允许接收的最小帧长, 单位为字节。
[10:0]	len_max_1	R/W	上行口允许接收的最大帧长, 单位为字节。

SF_STS_REG5

- 偏移地址: 0x0594
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:28]	rxsof_cnt_1	上行口接收到的帧头计数。
[27:24]	rxeof_cnt_1	上行口接收到的帧尾计数。
[23:20]	rxcrcok_cnt_1	上行口接收到 CRC 校验正确的帧计数。
[19:16]	rxcrbad_cnt_1	上行口接收到 CRC 校验错误的帧计数。
[15:12]	txsof_cnt_1	上行口发送的帧头计数。
[11:8]	txeof_cnt_1	上行口发送的帧尾计数。
[7:4]	txcrcok_cnt_1	上行口发送 CRC 校验正确的帧计数。
[3:0]	txcrbad_cnt_1	上行口发送 CRC 校验错误的帧计数。

SF_STS_REG6

上行口接收统计计数半满指示信号。

- 偏移地址: 0x0598
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s



比特	名称	描述
[31:30]	Reserved	保留。
[29]	cntr_stat_rx[29]	0: ifinoctets 计数值不足一半; 1: ifinoctets 计数值已超过一半。
[28]	cntr_stat_rx[28]	0: pkts 计数值不足一半; 1: pkts 计数值已超过一半。
[27]	cntr_stat_rx[27]	0: octets 计数值不足一半; 1: octets 计数值已超过一半。
[26]	cntr_stat_rx[26]	0: dot3dribble 计数值不足一半; 1: dot3dribble 计数值已超过一半。
[25]	cntr_stat_rx[25]	0: dot3unkpause 计数值不足一半; 1: dot3unkpause 计数值已超过一半。
[24]	cntr_stat_rx[24]	0: dot3inpause 计数值不足一半; 1: dot3inpause 计数值已超过一半。
[23]	cntr_stat_rx[23]	0: dot3internalrecerr 计数值不足一半; 1: dot3internalrecerr 计数值已超过一半。
[22]	cntr_stat_rx[22]	0: dot3fserr 计数值不足一半; 1: dot3fserr 计数值已超过一半。
[21]	cntr_stat_rx[21]	0: dot3alignmenterr 计数值不足一半; 1: dot3alignmenterr 计数值已超过一半。
[20]	cntr_stat_rx[20]	0: ifinbroadcast 计数值不足一半; 1: ifinbroadcast 计数值已超过一半。
[19]	cntr_stat_rx[19]	0: ifinmulticast 计数值不足一半; 1: ifinmulticast 计数值已超过一半。
[18]	cntr_stat_rx[18]	0: ifinerrors 计数值不足一半; 1: ifinerrors 计数值已超过一半。
[17]	cntr_stat_rx[17]	0: ifindiscards 计数值不足一半; 1: ifindiscards 计数值已超过一半。
[16]	cntr_stat_rx[16]	0: ifinnucastpkts 计数值不足一半; 1: ifinnucastpkts 计数值已超过一半。
[15]	cntr_stat_rx[15]	0: ifinucastpkts 计数值不足一半; 1: ifinucastpkts 计数值已超过一半。



比特	名称	描述
[14]	cntr_stat_rx[14]	0: pkts_1518 计数值不足一半; 1: pkts_1518 计数值已超过一半。
[13]	cntr_stat_rx[13]	0: pkts1024_1518 计数值不足一半; 1: pkts1024_1518 计数值已超过一半。
[12]	cntr_stat_rx[12]	0: pkts512_1023 计数值不足一半; 1: pkts512_1023 计数值已超过一半。
[11]	cntr_stat_rx[11]	0: pkts256_511 计数值不足一半; 1: pkts256_511 计数值已超过一半。
[10]	cntr_stat_rx[10]	0: pkts128_255 计数值不足一半; 1: pkts128_255 计数值已超过一半。
[9]	cntr_stat_rx[9]	0: pkts65_127 计数值不足一半; 1: pkts65_127 计数值已超过一半。
[8]	cntr_stat_rx[8]	0: pkts64 计数值不足一半; 1: pkts64 计数值已超过一半。
[7]	cntr_stat_rx[7]	0: multiicastpkts 计数值不足一半; 1: multicastpkts 计数值已超过一半。
[6]	cntr_stat_rx[6]	0: braodcastpkts 计数值不足一半; 1: broadcastpkts 计数值已超过一半。
[5]	cntr_stat_rx[5]	0: jabber 计数值不足一半; 1: jabber 计数值已超过一半。
[4]	cntr_stat_rx[4]	0: fragments 计数值不足一半; 1: fragments 计数值已超过一半。
[3]	cntr_stat_rx[3]	0: oversizepkts 计数值不足一半; 1: oversizepkts 计数值已超过一半。
[2]	cntr_stat_rx[2]	0: undersizepkts 计数值不足一半; 1: undersizepkts 计数值已超过一半。
[1]	cntr_stat_rx[1]	0: crcerr 计数值不足一半; 1: crcerr 计数值已超过一半。
[0]	cntr_stat_rx[0]	0: dropevents 计数值不足一半; 1: dropevents 计数值已超过一半。



SF_STS_REG7

上行口发送计数半满指示信号。

- 偏移地址：0x059C
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:30]	Reserved	保留。
[29]	cntr_stat_tx[29]	0: dot3outpause 计数值不足一半； 1: dot3outpause 计数值已超过一半。
[28]	cntr_stat_tx[28]	0: ifoutoctets 计数值不足一半； 1: ifoutoctets 计数值已超过一半。
[27]	cntr_stat_tx[27]	0: collisions 计数值不足一半； 1: collisions 计数值已超过一半。
[26]	cntr_stat_tx[26]	0: octets_tx 计数值不足一半； 1: octets_tx 计数值已超过一半。
[25]	cntr_stat_tx[25]	0: pkts_tx 计数值不足一半； 1: pkts_tx 计数值已超过一半。
[24]	cntr_stat_tx[24]	0: dot3defer 计数值不足一半； 1: dot3defer 计数值已超过一半。
[23]	cntr_stat_tx[23]	0: dot3colcnt 计数值不足一半； 1: dot3colcnt 计数值已超过一半。
[22]	cntr_stat_tx[22]	0: dot3excessivecol 计数值不足一半； 1: dot3excessivecol 计数值已超过一半。
[21]	cntr_stat_tx[21]	0: dot3latecol 计数值不足一半； 1: dot3latecol 计数值已超过一半。
[20]	cntr_stat_tx[20]	0: dot3multiplecol 计数值不足一半； 1: dot3multiplecol 计数值已超过一半。
[19]	cntr_stat_tx[19]	0: dot3singlecol 计数值不足一半； 1: dot3singlecol 计数值已超过一半。
[18]	cntr_stat_tx[18]	0: pkts_1518_tx 计数值不足一半； 1: pkts_1518_tx 计数值已超过一半。



比特	名称	描述
[17]	cntr_stat_tx[17]	0: pkts1024_1518_tx 计数值不足一半; 1: pkts1024_1518_tx 计数值已超过一半。
[16]	cntr_stat_tx[16]	0: pkts512_1023_tx 计数值不足一半; 1: pkts512_1023_tx 计数值已超过一半。
[15]	cntr_stat_tx[15]	0: pkts256_511_tx 计数值不足一半; 1: pkts256_511_tx 计数值已超过一半。
[14]	cntr_stat_tx[14]	0: pkts128_255_tx 计数值不足一半; 1: pkts128_255_tx 计数值已超过一半。
[13]	cntr_stat_tx[13]	0: pkts65_127_tx 计数值不足一半; 1: pkts65_127_tx 计数值已超过一半。
[12]	cntr_stat_tx[12]	0: pkts64_tx 计数值不足一半; 1: pkts64_tx 计数值已超过一半。
[11]	cntr_stat_tx[11]	0: ifoutbroadcast 计数值不足一半; 1: ifoutbroadcast 计数值已超过一半。
[10]	cntr_stat_tx[10]	0: ifoutmulticast 计数值不足一半; 1: ifoutmulticast 计数值已超过一半。
[9]	cntr_stat_tx[9]	0: ifoutnucastpkts 计数值不足一半; 1: ifoutnucastpkts 计数值已超过一半。
[8]	cntr_stat_tx[8]	0: ifoutucastpkts 计数值不足一半; 1: ifoutucastpkts 计数值已超过一半。
[7]	cntr_stat_tx[7]	0: ifouterrors 计数值不足一半; 1: ifouterrors 计数值已超过一半。
[6]	cntr_stat_tx[6]	0: ifoutdiscards 计数值不足一半; 1: ifoutdiscards 计数值已超过一半。
[5]	cntr_stat_tx[5]	0: multicastpkts_tx 计数值不足一半; 1: multicastpkts_tx 计数值已超过一半。
[4]	cntr_stat_tx[4]	0: broadcastpkts_tx 计数值不足一半; 1: broadcastpkts_tx 计数值已超过一半。
[3]	cntr_stat_tx[3]	0: oversizepkts 计数值不足一半; 1: oversizepkts 计数值已超过一半。



比特	名称	描述
[2]	cntr_stat_tx[2]	0: undersizepkts 计数值不足一半; 1: undersizepkts 计数值已超过一半。
[1]	cntr_stat_tx[1]	0: crcerr 计数值不足一半; 1: crcerr 计数值已超过一半。
[0]	cntr_stat_tx[0]	0: dropevents 计数值不足一半; 1: dropevents 计数值已超过一半。

18.6.3 MDIO 寄存器组

SF_MDIO_REG0

- 偏移地址: 0x05C0
- 操作类型: R/W
- 复位值: 0x8000
- 复位方式: h/s

比特	名称	描述
[31:16]	cpu_data_in	用于 MDIO 模块对 PHY 进行对写操作的数据寄存器。 进行写操作时, CPU 将要对 MDIO 写入的 16 数据先写入到该寄存器中。
[15]	finish	当 SF 完成对 PHY 的读/写操作后, 自动置 1。 当要进行第 2 次读写操作时, CPU 需先对该位进行写清零 (写入 0)。
[14]	Reserved	保留。
[13]	rw	用来表示对 PHY 的访问为读操作还是写操作。 0: 读操作; 1: 写操作。
[12:8]	phy_exaddr	对外部操作的 PHY 的对应的外部物理地址。 一个 MDIO 可以对外面的 2 个 PHY 进行读写访问。每个 PHY 有一个相应的地址。



比特	名称	描述
[7:5]	frq_dv	<p>对外部 PHY 进行读写操作时，对 MDC (MDIO 接口时钟) 的分频系数。</p> <p>这 3 位用来实现对 MDIO 的分频值的产生。</p> <p>frq_dv 与 MDC 频率对应关系 (以主时钟 100MHz 为例)：</p> <p>000: 对 SF 工作主时钟 50 分频, 2MHz;</p> <p>001: 对 SF 工作主时钟 100 分频, 1MHz;</p> <p>010: 对 SF 工作主时钟 150 分频, 666.7kHz;</p> <p>011: 对 SF 工作主时钟 200 分频, 500kHz;</p> <p>100: 对 SF 工作主时钟 250 分频, 400kHz;</p> <p>101: 对 SF 工作主时钟 300 分频, 333.3kHz;</p> <p>110: 对 SF 工作主时钟 350 分频, 285.7kHz;</p> <p>111: 对 SF 工作主时钟 400 分频, 250kHz。</p>
[4:0]	phy_inaddr	<p>所需操作的 PHY 寄存器的内部地址。</p> <p>PHY 内部有 32 个寄存器。用 5 位二进制数来表示寄存器地址。</p>

SF_MDIO_REG1

- 偏移地址: 0x05C4
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:16]	Reserved	保留。
[15:0]	cpu_data_out	用于对 PHY 进行读操作的数据寄存器, SF 将从 PHY 读取的 16 位数据将写入到该寄存器中供 CPU 读取。

SF_MDIO_REG2

- 偏移地址: 0x05C8
- 操作类型: R/W
- 复位值: 0x41
- 复位方式: h/s



比特	名称	描述
[31:10]	Reserved	保留。
[9:5]	phy_addr1	端口 1 (上行口) 所连接 PHY1 芯片的外部物理地址。
[4:0]	phy_addr0	端口 0 (下行口) 所连接 PHY0 芯片的外部物理地址。

SF_MDIO_REG3

- 偏移地址: 0x05CC
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:6]	Reserved	保留。
[5:4]	speed_mdio2mac	从 MDIO 接口得到的端口速度工作状态。 0: 10Mbit/s 方式; 1: 100Mbit/s 方式。 以 bitmap 的形式, 每一位对应一个端口: bit5: 对应上行口; bit4: 对应下行口。
[3:2]	link_mdio2mac	从 MDIO 接口得到的链接状态指示。 0: 未链接状态; 1: 链接状态。 以 bitmap 的形式, 每一位对应一个端口: bit3: 对应上行口; bit2: 对应下行口。
[1:0]	duplex_mdio2mac	从 MDIO 接口得到的外部端口双工工作状态, 处于半双工或全双工方式。 0: 半双工方式; 1: 全双工方式。 以 bitmap 的形式, 每一位对应一个端口: bit1: 对应上行口; bit0: 对应下行口。



 说明

此处 MDIO 接口获取状态是根据 SF_MDIO_REG4、SF_MDIO_REG5 配置从两个端口 PHY 寄存器扫描获得。

SF_MDIO_REG4

- 偏移地址：0x05D0
- 操作类型：R/W
- 复位值：0x463_1EA9
- 复位方式：h/s

比特	名称	描述
[31:27]	Reserved	保留。
[26:22]	internal_addr_speed_0	端口 0 (下行口) PHY 芯片存储状态信息 (速度) 的寄存器地址。缺省值按照 Intel 9785 设置。
[21:17]	internal_addr_link_0	端口 0 (下行口) PHY 芯片存储状态信息 (链接) 的寄存器地址。缺省值按照 Intel 9785 设置。
[16:12]	internal_addr_duplex_0	端口 0 (下行口) PHY 芯片存储状态信息 (双工) 的寄存器地址。缺省值按照 Intel 9785 设置。
[11:8]	speed_index_0	端口 0 (下行口) PHY 芯片状态寄存器中存储速度信息的偏移地址。缺省值按照 Intel 9785 设置。
[7:4]	link_index_0	端口 0 (下行口) PHY 芯片状态寄存器中存储链接信息的偏移地址。缺省值按照 Intel 9785 设置。
[3:0]	duplex_index_0	端口 0 (下行口) PHY 芯片状态寄存器中存储双工信息的偏移地址。缺省值按照 Intel 9785 设置。

 说明

如 PHY 芯片速度状态位于其地址为 17 的寄存器的第 14bit, 即可配置 internal_addr_speed_0 为 5'h11, speed_index_0 为 4'hE, 则此时 SF 可通过 MDIO 接口将该 bit 值读出作为 PHY 当前工作的速度模式信息。

SF_MDIO_REG5

- 偏移地址：0x05D4
- 操作类型：R/W
- 复位值：0x463_1EA9
- 复位方式：h/s

比特	名称	描述
[31:27]	Reserved	保留。



比特	名称	描述
[26:22]	internal_addr_speed_1	端口 1 (上行口) PHY 芯片存储状态信息 (速度) 的寄存器地址。缺省值按照 Intel 9785 设置。
[21:17]	internal_addr_link_1	端口 1 (上行口) PHY 芯片存储状态信息 (链接) 的寄存器地址。缺省值按照 Intel 9785 设置。
[16:12]	internal_addr_duplex_1	端口 1 (上行口) PHY 芯片存储状态信息 (双工) 的寄存器地址。缺省值按照 Intel 9785 设置。
[11:8]	speed_index_1	端口 1 (上行口) PHY 芯片状态寄存器中存储速度信息的偏移地址。缺省值按照 Intel 9785 设置。
[7:4]	link_index_1	端口 1 (上行口) PHY 芯片状态寄存器中存储链接信息的偏移地址。缺省值按照 Intel 9785 设置。
[3:0]	duplex_index_1	端口 1 (上行口) PHY 芯片状态寄存器中存储双工信息的偏移地址。缺省值按照 Intel 9785 设置。

18.6.4 IRF 和 ERF 寄存器组

SF_IERF_REG0

- 偏移地址: 0x05E0
- 操作类型: R/W
- 复位值: 0x1001
- 复位方式: h/s

比特	名称	描述
[31:24]	Reserved	保留。
[23:12]	port_vid_1	端口 1 (上行口) 对应的 Vlan ID。
[11:0]	port_vid_0	端口 0 (下行口) 对应的 Vlan ID。

SF_IERF_REG1

- 偏移地址: 0x05E4
- 操作类型: R/W
- 复位值: 0xC
- 复位方式: h/s



比特	名称	描述
[31:4]	Reserved	保留。
[3:2]	special_tag_en	对端口进行特殊目的 MAC 帧输出时，是否保持 tag 格式与输入一致。 0: 按端口 tag 格式配置决定输出 tag 格式； 1: 保持一致。 bit3: 对应端口 1（上行口）； bit2: 对应端口 0（下行口）。
[1:0]	tag_change	从以太网端口输出的 IP 多播帧为 untag 输入且带 tag 输出时是否要修改 VLAN ID。 0: 不修改，即当输入帧自带的 VLAN ID 有效时，不变输出，否则输出时带输入端口的 VLAN ID 进行输出； 1: 要修改为带输出端口的 VLAN ID 进行输出。 bit1: 对应端口 1（上行口）； bit0: 对应端口 0（下行口）。

SF_IERF_REG2

- 偏移地址：0x05E8
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:11]	Reserved	保留。
[10:9]	pri_change	从以太网端口输出的帧是否修改其 802.1p 优先级。 0: 保留其原有值； 1: 添加当前帧源端口（帧所来自的端口）对应的优先级。 bit10: 对应端口 1（上行口）； bit9: 对应端口 0（下行口）。
[8:6]	port_pri_2	当添加 802.1p 优先级时，来自 CPU 端口帧对应的优先级值。
[5:3]	port_pri_1	当添加 802.1p 优先级时，来自上行口的帧对应的优先级值。
[2:0]	port_pri_0	当添加 802.1p 优先级时，来自下行口的帧对应的优先级值。



18.6.5 全局寄存器组

SF_GLB_REG0

- 偏移地址：0x0600
- 操作类型：RWC（写 1 清零）
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:26]	Reserved	保留。
[25]	int_block_up	上行口发生广播帧队列阻塞（给 CPU 广播帧队列空，给上行口广播帧队列（D2BU）满）中断。
[24]	int_block_down	下行口发生广播帧队列阻塞（给 CPU 广播帧队列空，给下行口广播帧队列（U2BD）满）中断。
[23]	int_drop_up	上行口队列丢帧（BM 丢帧）中断，包括队列溢出、接收 CRC 错丢帧。
[22]	int_drop_down	下行口队列丢帧（BM 丢帧）中断，包括队列溢出、接收 CRC 错丢帧。。
[21]	int_cpu_rx_up	上行口队列有帧待 CPU 接收中断。
[20]	int_cpu_rx_down	下行口队列有帧待 CPU 接收中断。
[19]	int_cpu_full_up	上行口给 CPU 的帧接收队列满中断。
[18]	int_cpu_full_down	下行口给 CPU 的帧接收队列满中断。
[17]	int_fd_err	CPU 发送帧描述子错误中断。用于指示 CPU 发出的帧存储队列与帧描述子标识的目的端口不匹配。
[16]	int_err_bus	总线操作出错指示中断信号。
[15]	Reserved	保留。
[14]	int_cpu_rx	SF 有帧等待 CPU 接收。
[13]	int_tx_cpu	SF 发送完来自 CPU 的一帧数据指示。
[12]	int_free_cpu	SF 有空间接收来自 CPU 的帧。
[11]	int_link_ch	SF 连接状态变化中断。
[10]	int_speed_ch	SF 速度模式变化中断。
[9]	int_duplex_ch	SF 双工模式变化中断。
[8]	int_unkvlan	SF 接收到未知 VLAN 帧。



比特	名称	描述
[7]	int_unkvlanmem	SF 接收到未知 VLAN 成员帧。
[6]	int_mdio_finish	SF 完成一次 CPU 对 MDIO 接口的读/写操作指示。
[5:0]	Reserved	保留。

SF_GLB_REG1

- 偏移地址：0x0604
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:26]	Reserved	保留。
[25]	ien_block_up	上行口广播帧队列阻塞中断使能。
[24]	ien_block_down	下行口广播帧队列阻塞中断使能。
[23]	ien_drop_up	上行口队列丢帧中断使能。
[22]	ien_drop_down	下行口队列丢帧中断使能。
[21]	ien_cpu_rx_up	上行口队列有帧待 CPU 接收中断使能。
[20]	ien_cpu_rx_down	下行口队列有帧待 CPU 接收中断使能。
[19]	ien_cpu_full_up	上行口给 CPU 队列满中断使能。
[18]	ien_cpu_full_down	下行口给 CPU 队列满中断使能。
[17]	ien_fd_err	CPU 发送错误帧描述子指示中断使能。
[16]	ien_err_buf	总线出错指示中断使能。
[15]	Reserved	保留。
[14]	ien_cpu_rx	SF 有帧等待 CPU 接收中断使能。
[13]	ien_tx_cpu	SF 发送完来自 CPU 的一帧数据指示中断使能。
[12]	ien_free_cpu	SF 有空间接收来自 CPU 的帧中断使能。
[11]	ien_link_ch	SF 连接状态变化中断使能。
[10]	ien_speed_ch	SF 速度模式变化中断使能。
[9]	ien_duplex_ch	SF 双工模式变化中断使能。



比特	名称	描述
[8]	ien_unkvlan	SF 接收到未知 VLAN 帧中断使能。
[7]	ien_unkvlanm	SF 接收到未知 VLAN 成员帧中断使能。
[6]	ien_mdio_finish	SF 完成一次 CPU 对 MDIO 接口的读/写操作指示中断使能。
[5:0]	Reserved	保留。

说明

保留位缺省为 0，不允许修改。

SF_GLB_REG2

- 偏移地址：0x0608
- 操作类型：R/W
- 复位值：0x3FF_FFFF
- 复位方式：h/s

比特	名称	描述
[31:26]	Reserved	保留。
[25]	ioutsel_block_up	上行口广播帧队列阻塞中断输出选择。
[24]	ioutsel_block_down	下行口广播帧队列阻塞中断输出选择。
[23]	ioutsel_drop_up	上行口队列丢帧中断输出选择。
[22]	ioutsel_drop_down	下行口队列丢帧中断输出选择。
[21]	ioutsel_cpu_rx_up	上行口队列有帧待 CPU 接收中断输出选择。
[20]	ioutsel_cpu_rx_down	下行口队列有帧待 CPU 接收中断输出选择。
[19]	ioutsel_cpu_full_up	上行口给 CPU 队列满中断输出选择。
[18]	ioutsel_cpu_full_down	下行口给 CPU 队列满中断输出选择。
[17]	ioutsel_fd_err	CPU 发送错误帧描述子指示中断输出选择。
[16]	ioutsel_err_buf	总线出错指示中断输出选择。
[15]	Reserved	保留。
[14]	ioutsel_cpu_rx	SF 有帧等待 CPU 接收中断输出选择。
[13]	ioutsel_tx_cpu	SF 发送完来自 CPU 的一帧数据指示中断输出选择。



比特	名称	描述
[12]	ioutsel_free_cpu	SF 有空间接收来自 CPU 的帧中断输出选择。
[11]	ioutsel_link_ch	SF 连接状态变化中断输出选择。
[10]	ioutsel_speed_ch	SF 速度模式变化中断输出选择。
[9]	ioutsel_duplex_ch	SF 双工模式变化中断输出选择。
[8]	ioutsel_unkvlan	SF 接收到未知 VLAN 帧中断输出选择。
[7]	ioutsel_unkvlanm	SF 接收到未知 VLAN 成员帧中断输出选择。
[6]	ioutsel_mdio_finish	SF 完成一次 CPU 对 MDIO 接口的读/写操作指示中断输出选择。
[5:0]	Reserved	保留。

说明

保留位缺省为 0，不允许修改。

中断使能 (SF_GLB_REG1) 控制 SF_GLB_REG0 中状态信号 int_xxx 的产生，中断输出 (SF_GLB_REG2) 控制中断状态信号是否产生中断，通知 CPU 处理。

SF_GLB_REG3

- 偏移地址：0x060C
- 操作类型：R
- 复位方式：h/s

比特	名称	描述
[31:0]	Reserved	保留。

SF_GLB_REG4

- 偏移地址：0x0610
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:0]	self_mac0	本机 MAC 地址的低 32 位。

注：此处 MAC 配置数据需为大端格式 (big endian)。



SF_GLB_REG5

- 偏移地址: 0x0614
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:16]	Reserved	保留。
[15:0]	self_mac1	本机 MAC 地址的高 16 位。

注: 此处 MAC 配置数据需为大端 (big endian) 格式。

SF_GLB_REG6

- 偏移地址: 0x0618
- 操作类型: R/W
- 复位方式: h/s

比特	名称	描述
[31:0]	Reserved	保留。

SF_GLB_REG7

- 偏移地址: 0x061C
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	Reserved	保留。

SF_GLB_REG8

- 偏移地址: 0x0620
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s



比特	名称	描述
[31:0]	Reserved	保留。

SF_GLB_REG9

- 偏移地址: 0x0624
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	Reserved	保留。

SF_GLB_REG10

- 偏移地址: 0x0628
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	Reserved	保留。

SF_GLB_REG11

- 偏移地址: 0x062C
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:4]	Reserved	保留。
[3]	monitor_en	监听使能信号。 0: 禁止监听; 1: 进行监听。



[2]	monitor_port	监听端口选择，仅在监听使能时有效。 0：选择下行口为监听端口； 1：选择上行口为监听端口。
[1:0]	mode	SF 工作模式选择。 00：对应普通模式（正常工作模式）； 01~11：保留。

说明

保留位缺省为 0，不允许修改。

SF_GLB_REG12

- 偏移地址：0x0630
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7]	statis_init_rdy	STATIS 初始化完成信号。
[6]	vlan_init_rdy	VLAN 信息表初始化完成指示信号。
[5]	at_init_rdy	ANYTYPE 表初始化完成指示信号。
[4]	ap_init_rdy	ANYPORT 表初始化完成指示信号。
[3:0]	Reserved	保留。

说明

各表控制逻辑在复位信号有效时，自动对表项进行初始化操作，初始化完成后，相应指示信号自动置为 1。

SF_GLB_REG13

- 偏移地址：0x0634
- 复位值：-
- 复位方式：h/s

比特	名称	描述
[31:0]	Reserved	保留。



SF_GLB_REG14

- 偏移地址: 0x0638
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	Reserved	保留。

SF_GLB_REG15

- 偏移地址: 0x063C
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	Reserved	保留。

SF_GLB_REG16

- 偏移地址: 0x0640
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	Reserved	保留。

SF_GLB_REG17

- 偏移地址: 0x0644
- 操作类型: R/W
- 复位值: 0x1
- 复位方式: h/s

比特	名称	描述
[31:13]	Reserved	保留。



[12:0]	hd_addr_bm	SF 在片外存储器中缓存帧数据的地址区域首地址最高 13 位 (SF 使用起始地址开始的 288KB 空间来缓存数据), 即首地址 = {hd_addr_bm, 19'b0}。
--------	------------	--

SF_GLB_REG18

- 偏移地址: 0x0648
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:2]	Reserved	保留。
[1:0]	mode_rmii	SF 端口 RMII/MII 模式选择。 0: 对应 MII 模式; 1: 为 RMII 模式。 bit1 对应上行口, bit0 对应下行口。

说明

在网口处于连接状态时, 如果修改 MII/RMII 配置, 需在修改之后对相应端口进行端口复位。

18.6.6 IQM 管理寄存器组

SF_IQM_REG0

- 偏移地址: 0x0480
- 操作类型: R/W
- 复位值: 0x686
- 复位方式: h/s

比特	名称	描述
[31:17]	Reserved	保留。
[16]	igmp_en1	端口 1 (上行口) igmp 帧识别使能。 0: 端口 1IGMP 帧识别禁止; 1: 端口 1IGMP 帧识别使能。
[15:14]	igmp_ctrl1	端口 1 (上行口) igmp 帧转发配置。 00: 强制丢弃; 10: 强制发往 CPU; 01: 强制发往下行口; 11: 强制发往 CPU 和下行口。



比特	名称	描述
[13]	force_en1	端口 1 (上行口) 输入帧强制转发使能信号。 0: 端口 1 强制转发禁止; 1: 端口 1 强制转发使能。
[12:11]	force_ctrl1	端口 1 (上行口) 强制转发配置。 00: 强制丢弃; 10: 强制发往 CPU; 01: 强制发往下行口; 11: 强制发往 CPU 和下行口。
[10:9]	broad_ctrl1	端口 1 (上行口) 广播帧转发配置。 00: 强制丢弃; 10: 强制发往 CPU; 01: 强制发往下行口; 11: 强制发往 CPU 和下行口。
[8]	spcl_en1	端口 1 (上行口) 特殊帧识别使能。 0: 端口 1 特殊目的 MAC 帧识别禁止; 1: 端口 1 特殊目的 MAC 帧识别使能。
[7:6]	spcl_ctrl1	端口 1 (上行口) 特殊帧转发配置。 00: 丢弃; 10: 发往 CPU; 01: 发往下行口; 11: 发往 CPU 和下行口。
[5]	ipm_en1	端口 1 (上行口) ipm 帧识别使能。 0: 端口 1 ipm 帧识别禁止; 1: 端口 1 ipm 帧 (IP 多播帧) 识别使能。
[4:3]	ipm_ctrl1	端口 1 (上行口) ipm 帧转发配置。 00: 丢弃; 10: 发往 CPU; 01: 发往下行口; 11: 发往 CPU 和下行口。
[2]	Reserved	保留。
[1]	other_ctrl1	端口 1 (上行口) 非特殊帧转发配置。 0: 丢弃; 1: 正常转发。
[0]	Reserved	保留。

SF_IQM_REG1

- 偏移地址: 0x0484
- 操作类型: R/W
- 复位值: 0x686



- 复位方式: h/s

比特	名称	描述
[31:17]	Reserved	保留。
[16]	igmp_en0	端口 0 (下行口) igmp 帧识别使能。 0: 端口 0 IGMP 帧识别禁止; 1: 端口 0 IGMP 帧识别使能。
[15:14]	igmp_ctrl0	端口 0 (下行口) igmp 帧转发配置。 00: 强制丢弃; 10: 强制发往 CPU; 01: 强制发往上行口; 11: 强制发往 CPU 和上行口。
[13]	force_en0	端口 0 (下行口) 输入帧强制转发使能信号。 0: 端口 0 强制转发禁止; 1: 端口 0 强制转发使能。
[12:11]	force_ctrl0	端口 0 (下行口) 强制转发配置。 00: 强制丢弃; 10: 强制发往 CPU; 01: 强制发往上行口; 11: 强制发往 CPU 和上行口。
[10:9]	broad_ctrl0	端口 0 (下行口) 广播帧转发配置。 00: 强制丢弃; 10: 强制发往 CPU; 01: 强制发往上行口; 11: 强制发往 CPU 和上行口。
[8]	spcl_en0	端口 0 (下行口) 特殊帧识别使能。 0: 端口 0 特殊目的 MAC 帧识别禁止; 1: 端口 0 特殊目的 MAC 帧识别使能。
[7:6]	spcl_ctrl0	端口 0 (下行口) 特殊帧转发配置。 00: 丢弃; 10: 发往 CPU; 01: 发往上行口; 11: 发往 CPU 和上行口。
[5]	ipm_en0	端口 0 (下行口) ipm 帧识别使能。 0: 端口 0 ipm 帧识别禁止; 1: 端口 0 ipm 帧 (IP 多播帧) 识别使能。
[4:3]	ipm_ctrl0	端口 0 (下行口) ipm 帧转发配置。 00: 丢弃; 10: 发往 CPU; 01: 发往上行口; 11: 发往 CPU 和上行口。
[2]	Reserved	保留。



比特	名称	描述
[1]	other_ctrl0	端口 0 (下行口) 非特殊帧转发配置。 0: 丢弃; 1: 正常转发。
[0]	Reserved	保留。

SF_IQM_REG2

- 偏移地址: 0x0488
- 操作类型: R/W
- 复位值: 0x3
- 复位方式: h/s

比特	名称	描述
[31:2]	Reserved	保留。
[1:0]	cpu_ctrl	CPU 转发控制表。 0: 禁止发往指定端口; 1: 允许发往指定端口。 bit1 表示是否允许帧发往 CPU 端口, bit0 表示是否允许帧发往另一以太网端口。

SF_IQM_REG3

- 偏移地址: 0x048C
- 操作类型: R/W
- 复位值: 0x18
- 复位方式: h/s

比特	名称	描述
[31:5]	Reserved	保留。
[4:3]	tag_type	帧输出 tag 格式控制选择。 0: 按帧对应 VLAN 信息表中的配置 tag 格式; 1: 按端口配置 (tag_port) 帧 tag 格式。 bit4 表示选择端口 1 输出 tag 格式控制; bit3 表示选择端口 0 输出 tag 格式控制。
[2]	Reserved	保留。



比特	名称	描述
[1]	tag_port1	上行口输出帧是否添加 VLAN tag 控制（仅在 tag_type 端口 1 配置为 1 时有效）。
[0]	tag_port0	下行口输出帧是否添加 VLAN tag 控制（仅在 tag_type 端口 0 配置为 1 时有效）。

SF_IQM_REG4

- 偏移地址：0x0490
- 操作类型：R/W
- 复位值：0x3
- 复位方式：h/s

比特	名称	描述
[31:2]	Reserved	保留。
[1]	vlan_mode	配置 SF 帧转发是否限制在 VLAN 域内。 0：表示不限制； 1：表示进行 VLAN 域限制。
[0]	disable_vlan	配置 disable VLAN，即不进行 VLAN 转发控制。 0：使能 VLAN； 1：禁止 VLAN。

SF_IQM_REG5

- 偏移地址：0x0494
- 操作类型：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:6]	Reserved	保留。



比特	名称	描述
[5:3]	vlanmode_port1	上行口可接收帧 VLAN 格式控制。 001: 只接收不带 tag 的帧和优先级帧; 010: 只接收不带 tag 的帧和 tag 有效帧; 011: 只接收带 tag 的帧; 100: 只能接收 tag 有效帧; 101: 保留; 110: 可以接收所有的帧; 其他: 可以接收所有的帧。
[2:0]	vlanmode_port0	下行口可接收帧 VLAN 格式控制。

SF_IQM_REG6

- 偏移地址: 0x0498
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:8]	Reserved	保留。
[7:5]	unkvlan_tag	未知 VLAN 对应输出 tag 格式控制。 0: 未知 VLAN 帧带 VLAN tag 输出; 1: 未知 VLAN 帧不带 VLAN tag 输出。 bit7 对应 CPU 端口, bit6 对应上行口, bit5 对应下行口。
[4:3]	pass_mode	未知 VLAN 接收允许表。 0: 禁止接收; 1: 允许接收。 bit4 对应上行口, bit3 对应下行口。
[2:0]	unkvlan_ctrl	未知 VLAN 强制转发表。 0: 禁止; 1: 允许发送到该位对应的端口。 bit2 对应 CPU 端口, bit1 对应上行口, bit0 对应下行口。



SF_IQM_REG7

- 偏移地址: 0x049C
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:5]	Reserved	保留。
[4:3]	rxfilter	未知 VLAN 成员接收允许表。 0: 禁止; 1: 该端口允许接收未知 VLAN 成员帧。 bit4 对应上行口, bit3 对应下行口。
[2:0]	unkmem_ctrl	未知 VLAN 成员强制转发表。 0: 禁止; 1: 允许发送到该位对应的端口。 bit2 对应 CPU, bit1 对应上行口, bit0 对应下行口。

SF_IQM_REG8

- 偏移地址: 0x04A0
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:25]	Reserved	保留。
[24]	unkmemport	未知 VLAN 成员对应的源端口号。
[23:12]	unkmemid	未知 VLAN 成员对应的 VLAN ID 号。
[11:0]	unkvlanid	未知 VLAN 对应的 VLAN ID 号。

18.6.7 BM 管理寄存器组

SF_BM_REG0

- 偏移地址: 0x0400
- 操作类型: R/W



- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	Reserved	保留。

SF_BM_REG1

- 偏移地址: 0x0404
- 操作类型: R/W
- 复位值: 0x3_73BC
- 复位方式: h/s

比特	名称	描述
[31:18]	Reserved	保留。
[17:14]	q16_limit_up	CPU 配置深度为 16 的队列(u2bc, d2bc)上限阈值 (空闲空间)。缺省 4'hD。
[13:10]	q16_limit_down	CPU 配置深度为 16 的队列(u2bc, d2bc)下限阈值。缺省为 4'hC。
[9:5]	q32_limit_up	CPU 配置深度为 32 的队列(u2c, d2c)上限阈值 (空闲空间)。缺省 5'h1D。
[4:0]	q32_limit_down	CPU 配置深度为 32 的队列(u2c, d2c)下限阈值。缺省为 5'h1C。

说明

- u2d: SF 中的帧缓存队列, 存储源端口为上行口, 目的端口为下行口的单播帧。
- u2c: SF 中的帧缓存队列, 存储源端口为上行口, 目的端口为 CPU 端口的单播帧。
- u2bc: SF 中的帧缓存队列, 存储源端口为上行口, 目的端口为 CPU 端口的广播帧。
- u2bd: SF 中的帧缓存队列, 存储源端口为上行口, 目的端口为下行口的广播帧。
- d2u: SF 中的帧缓存队列, 存储源端口为下行口, 目的端口为上行口的单播帧。
- d2c: SF 中的帧缓存队列, 存储源端口为下行口, 目的端口为 CPU 端口的单播帧。
- d2bc: SF 中的帧缓存队列, 存储源端口为下行口, 目的端口为 CPU 端口的广播帧。
- d2bu: SF 中的帧缓存队列, 存储源端口为下行口, 目的端口为上行口的广播帧。

SF_BM_REG2

- 偏移地址: 0x0408
- 操作类型: R/W



- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	Fd_cpu	CPU 发送帧的帧描述子。

SF_BM_REG3

- 偏移地址: 0x040C
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:5]	Reserved	保留。
[4]	Rx_end_d2bc	CPU 读完下行口发往 CPU 的广播帧后给出的指示信号。
[3]	Rx_end_u2bc	CPU 读完上行口发往 CPU 的广播帧后给出的指示信号。
[2]	Rx_end_d2c	CPU 读完下行口发往 CPU 的单播帧后给出的指示信号。
[1]	Rx_end_u2c	CPU 读完上行口发往 CPU 的单播帧后给出的指示信号。
[0]	Flush_all	CPU 清空输入队列信号, 用于将输入队列中所有待接收数据清空。 写 1 表示 CPU 清除掉存储在片外缓存中所有待接收的帧数据。

SF_BM_REG4

- 偏移地址: 0x0410
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	fd_u2c	从上行口接收, 转发到 CPU 端口帧的帧描述子。

SF_BM_REG5

- 偏移地址: 0x0414



- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	fd_d2c	从下行口接收, 转发到 CPU 端口帧的帧描述子。

SF_BM_REG6

- 偏移地址: 0x0418
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	fd_u2bc	从上行口接收, 需要转发到 CPU 端口的广播帧的帧描述子。 此处广播帧指需要同时发送到 CPU 端口和另外一个外部端口的帧。

SF_BM_REG7

- 偏移地址: 0x041C
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:0]	fd_d2bc	从下行口接收, 需要转发到 CPU 端口的广播帧的帧描述子。 此处广播帧指需要同时发送到 CPU 端口和另外一个外部端口的帧。

SF_BM_REG8

- 偏移地址: 0x0420
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s



比特	名称	描述
[31:4]	Reserved	保留。
[3]	Fd_vld_d2bc	下行口发来的广播帧送给 CPU 的帧描述子有效信号。
[2]	Fd_vld_u2bc	上行口发来的广播帧送给 CPU 的帧描述子有效信号。
[1]	Fd_vld_d2c	下行口送给 CPU 的帧描述子有效信号。
[0]	Fd_vld_u2c	上行口送给 CPU 的帧描述子有效信号。

SF_BM_REG9

- 偏移地址：0x0424
- 操作类型：R
- 复位值：0x100
- 复位方式：h/s

比特	名称	描述
[31:9]	Reserved	保留。
[8]	Tail_vld_c2u	CPU 发给上行口的帧描述子队列尾地址有效信号。 0: 当前提供的队列尾地址无效; 1: 当前提供的队列尾地址有效。
[7:0]	Tail_c2u	CPU 发给上行口的帧描述子队列尾地址索引。 在 SDRAM 缓存的尾地址为 $start_address + tail_c2u \times 2048$ 。 其中 $start_address$ 为 SF 被分配使用的 SDRAM 中帧缓存区域的起始地址。

SF_BM_REG10

- 偏移地址：0x0428
- 操作类型：R
- 复位值：0x110
- 复位方式：h/s

比特	名称	描述
[31:9]	Reserved	保留。



比特	名称	描述
[8]	Tail_vld_c2d	CPU 发给下行口的帧描述子队列尾地址有效信号。 0: 当前提供的队列尾地址无效; 1: 当前提供的队列尾地址有效。
[7:0]	Tail_c2d	CPU 发给下行口的帧描述子队列尾地址索引。尾地址计算方法请参见 SF_BM_REG9 描述。

SF_BM_REG11

- 偏移地址: 0x042C
- 操作类型: R
- 复位值: 0x160
- 复位方式: h/s

比特	名称	描述
[31:9]	Reserved	保留。
[8]	Tail_vld_c2b	CPU 发的广播帧的帧描述子队列尾地址有效信号。 0: 当前提供的队列尾地址无效; 1: 当前提供的队列尾地址有效。
[7:0]	Tail_c2b	CPU 发的广播帧的帧描述子队列尾地址索引。尾地址计算方法请参见 SF_BM_REG9 描述。

SF_BM_REG12

- 偏移地址: 0x0430
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:23]	Reserved	保留。
[22:11]	Fcnt_u2c	上行口发往 CPU 的帧和上行口的广播帧使用片内缓存区域的深度 (单位: double word (32bit))。
[10:0]	Fcnt_u2d	上行口发往下行口的帧使用片内缓存区域的深度。



SF_BM_REG13

- 偏移地址：0x0434
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:23]	Reserved	保留。
[22:12]	Fcnt_d2c	下行口发往 CPU 的帧和下行口的广播帧使用片内缓存区域的深度（单位：double word（32bit））。
[11:0]	Fcnt_d2u	下行口发向上行口的帧使用片内缓存区域的深度。

SF_BM_REG14

- 偏移地址：0x0438
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:26]	Reserved	保留。
[25:20]	fmcnt_d2c	片内缓存中所存储的下行口发往 CPU 的帧的个数。
[19:13]	fmcnt_d2u	片内缓存中所存储的下行口发向上行口的帧的个数。
[12:6]	fmcnt_u2c	片内缓存中所存储的上行口发往 CPU 的帧的个数。
[5:0]	fmcnt_u2d	片内缓存中所存储的上行口发往下行口的帧的个数。

SF_BM_REG15

- 偏移地址：0x043C
- 操作类型：R
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[31:26]	Reserved	保留。
[25:22]	Cnt_d2bu	片外缓存中所存储的下行口发向上行口的广播帧的个数。



比特	名称	描述
[21:18]	Cnt_d2bc	片外缓存中所存储的下行口发往 CPU 的广播帧的个数。
[17:13]	Cnt_d2c	片外缓存中所存储的下行口发往 CPU 的单播帧的个数。
[12:9]	Cnt_u2bd	片外缓存中所存储的上行口发往下行口的广播帧的个数。
[8:5]	cnt_u2bc	片外缓存中所存储的上行口发往 CPU 的广播帧的个数。
[4:0]	Cnt_u2c	片外缓存中所存储的上行口发往 CPU 的单播帧的个数。

SF_BM_REG16

- 偏移地址: 0x0440
- 操作类型: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:16]	Reserved	保留。
[15:12]	cnt_c2bd	片外缓存中所存储的 CPU 发往下行口的广播帧的个数。
[11:8]	Cnt_c2d	片外缓存中所存储的 CPU 发往下行口的单播帧的个数。
[7:4]	cnt_c2bu	片外缓存中所存储的 CPU 发向上行口的广播帧的个数。
[3:0]	Cnt_c2u	片外缓存中所存储的 CPU 发向上行口的单播帧的个数。

SF_BM_REG17

- 偏移地址: 0x0444
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:1]	Reserved	保留。



比特	名称	描述
[0]	swap_en	SF 读、写片外数据大端格式、小端格式 (little endian) 转换使能。 0: 输出数据大端格式, 输入数据也为大端格式; 1: 输出数据大端格式转换为小端格式, 输入数据小端格式转大端格式。

SF_BM_REG18

- 偏移地址: 0x0448
- 操作类型: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:30]	Reserved	保留。
[29:24]	up_at1_cfg	上行口发送到 CPU 端口第 1 组 ANYTYPE 报文帧率限制寄存器。 5'h0: 无限制。
[23:22]	Reserved	保留。
[21:16]	up_at0_cfg	上行口发送到 CPU 端口第 0 组 ANYTYPE 报文帧率限制寄存器。 5'h0: 无限制。
[15:14]	Reserved	保留。
[13:8]	up_ap1_cfg	上行口发送到 CPU 端口第 1 组 ANYPORT 报文帧率限制寄存器。 5'h0: 无限制。
[7:6]	Reserved	保留。
[5:0]	up_ap0_cfg	上行口发送到 CPU 端口第 0 组 ANYPORT 报文帧率限制寄存器。 5'h0: 无限制。

SF_BM_REG19

- 偏移地址: 0x044C
- 操作类型: R/W



- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:30]	Reserved	保留。
[29:24]	up_others_cfg	上行口发送到 CPU 端口其他 (非特殊帧) 报文帧率限制寄存器。 5'h0: 无限制。
[23:22]	Reserved	保留。
[21:16]	up_broad_spcl_igmp_cfg	上行口发送到 CPU 端口广播/SPCL/IGMP 报文帧率限制寄存器。 5'h0: 无限制。
[15:6]	Reserved	保留。
[5:0]	up_ip_normal_cfg	上行口发送到 CPU 端口的非 TCP/UDP 报文帧率限制寄存器。 5'h0: 无限制。

说明

SF_BM_REG18~19 中 xxx_cfg 为 5'h0~5'h1F 时, 表示在 inter_cfg (SF_BM_REG20[21:16]) 配置时间内允许通过的帧数量为 $16 \times \text{xxx_cfg}$ 。

SF_BM_REG20

- 偏移地址: 0x0450
- 操作类型: W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[31:22]	Reserved	保留。
[21:16]	inter_cfg	帧率限制时间间隔控制。 5'h0: 无限制; 5'hF: 0.1677s。
[15:14]	Reserved	保留。
[13:8]	down_cfg	下行口发送到 CPU 端口所有报文帧率限制寄存器。 5'h0: 无限制。
[7:6]	Reserved	保留。



比特	名称	描述
[5:0]	up_cfg	上行口发送到 CPU 端口所有报文帧率限制寄存器。 5'h0: 无限制。

18.7 外部端口 RMON/SNMP 统计计数结果寄存器

统计计数表用于存储 2 个外部端口行为的统计计数结果，2 个端口各对应 56 个计数器。其中下行口对应 0x0000~0x00E4，上行口对应 0x0100~0x01E4。

统计结果可配置只读和读清零 2 种模式，由 SF_STS_REG0 和 SF_STS_REG4 控制，请参见“SF_STS_REG0”和“SF_STS_REG4”的第 27 位。

说明

表 18-11 的说明如下：

- 所列地址为相对地址（统计计数结果的地址），实际地址=统计计数缓存地址+相对地址。
- 共有 112（每端口 56）个统计结果，使用 512 字节空间，9 位地址线。
- 每类寄存器的各个比特的属性和定义均相同，只是针对不同的端口设置。
- 2 个端口的寄存器名用不同的后缀标识，“_0”对应端口 0（下行口），“_1”对应端口 1（上行口）。
- 操作类型：R
- 复位方式：h/s/s
- 复位值：0x0

表18-11 STATIS 统计结果部分寄存器的属性和定义

DIO 地址	比特	名称	描述
dropevents			
0x0000	[31:0]	dropevents_0	帧接收的过程中，SF 接收缓冲溢出事件的累计次数。 接收到 RXDV 下降沿时，如果 SF 接收缓冲发生溢出，则计数加 1。
0x0100	[31:0]	dropevents_1	帧接收的过程中，SF 接收缓冲溢出事件的累计次数。 接收到 RXDV 下降沿时，如果 SF 接收缓冲发生溢出，则计数加 1。
crcerr			
0x0004	[31:0]	crcerr_0	接收帧的帧长有效，但其 CRC 或 Alignment 检查出错的帧的个数。



DIO 地址	比特	名称	描述
0x0104	[31:0]	crcerr_1	接收帧的帧长有效, 但其 CRC 或 Alignment 检查出错的帧的个数。
undersizepkts			
0x0008	[31:0]	undersizepkts_0	接收帧的帧长小于设定的最小有效帧长 (缺省为 64 字节), 且 CRC 校验正确的帧的个数。
0x0108	[31:0]	undersizepkts_1	接收帧的帧长小于设定的最小有效帧长 (缺省为 64 字节), 且 CRC 校验正确的帧的个数。
oversizepkts			
0x000C	[31:0]	oversizepkts_0	接收帧的帧长大于设定的最大有效帧长, 且 CRC 校验正确的帧的个数。
0x010C	[31:0]	oversizepkts_1	接收帧的帧长大于设定的最大有效帧长, 且 CRC 校验正确的帧的个数。
fragments			
0x0010	[31:0]	fragments_0	接收帧的帧长小于设定的最小有效帧长 (缺省值为 64 字节), 并且其 CRC 或 Alignment 检查出错的帧的个数。
0x0110	[31:0]	fragments_1	接收帧的帧长小于设定的最小有效帧长 (缺省值为 64 字节), 并且其 CRC 或 Alignment 检查出错的帧的个数。
jabber			
0x0014	[31:0]	jabber_0	接收帧的帧长大于设定的最大有效值 (缺省值为 1518 字节), 并且其 CRC 或 Alignment 检查出错的帧的个数。
0x0114	[31:0]	jabber_1	接收帧的帧长大于设定的最大有效值 (缺省值为 1518 字节), 并且其 CRC 或 Alignment 检查出错的帧的个数。
broadcastpkts			
0x0018	[31:0]	broadcastpkts_0	仅为 CRC 校验正确的广播帧的计数。
0x0118	[31:0]	broadcastpkts_1	仅为 CRC 校验正确的广播帧的计数。
multicastpkts			
0x001C	[31:0]	multicastpkts_0	仅为 CRC 校验正确的多播帧的计数。
0x011C	[31:0]	multicastpkts_1	仅为 CRC 校验正确的多播帧的计数。



DIO 地址	比特	名称	描述
pkts64			
0x0020	[31:0]	pkts64_0	接收帧长为 64 字节的帧的计数，包含正确帧和错误帧。
0x0120	[31:0]	pkts64_1	接收帧长为 64 字节的帧的计数，包含正确帧和错误帧。
pkts65_127			
0x0024	[31:0]	pkts65_127_0	接收帧长为从 65 到 127 字节的帧的计数，包含正确帧和错误帧。
0x0124	[31:0]	pkts65_127_1	接收帧长为从 65 到 127 字节的帧的计数，包含正确帧和错误帧。
pkts128_255			
0x0028	[31:0]	pkts128_255_0	接收帧长为从 128 到 255 字节的帧的计数，包含正确帧和错误帧。
0x0128	[31:0]	pkts128_255_1	接收帧长为从 128 到 255 字节的帧的计数，包含正确帧和错误帧。
pkts256_511			
0x002C	[31:0]	pkts256_511_0	接收帧长为从 256 到 511 字节的帧的计数，包含正确帧和错误帧。
0x012C	[31:0]	pkts256_511_1	接收帧长为从 256 到 511 字节的帧的计数，包含正确帧和错误帧。
pkts512_1023			
0x0030	[31:0]	pkts512_1023_0	接收帧长为从 512 到 1023 字节的帧的计数，包含正确帧和错误帧。
0x0130	[31:0]	pkts512_1023_1	接收帧长为从 512 到 1023 字节的帧的计数，包含正确帧和错误帧。
pkts1024_1518			
0x0034	[31:0]	pkts1024_1518_0	接收帧长为从 1024 到 1518 字节的帧的计数，包含正确帧和错误帧。
0x0134	[31:0]	pkts1024_1518_1	接收帧长为从 1024 到 1518 字节的帧的计数，包含正确帧和错误帧。
pkts_1518			
0x0038	[31:0]	pkts1518_0	接收帧长大于 1518 字节但小于设定最大包长的帧的计数，包含正确帧和错误帧。



DIO 地址	比特	名称	描述
0x0138	[31:0]	pkts1518_1	接收帧长大于 1518 字节但小于设定最大包长的帧的计数, 包含正确帧和错误帧。
ifinucastpkts			
0x003C	[31:0]	ifinucastpkts_0	接收到所有单播帧计数 (DA 最高 byte 为偶数)。
0x013C	[31:0]	ifinucastpkts_1	接收到所有单播帧计数 (DA 最高 byte 为偶数)。
ifinnucastpkts			
0x0040	[31:0]	ifinnucastpkts_0	接收到所有非单播帧计数。
0x0140	[31:0]	ifinnucastpkts_1	接收到所有非单播帧计数。
ifindiscards			
0x0044	[31:0]	ifindiscards_0	接收的错帧数 (包含 CRC 错、超短帧、超长帧、Overflow 和传输错)。
0x0144	[31:0]	ifindiscards_1	接收的错帧数 (包含 CRC 错、超短帧、超长帧、Overflow 和传输错)。
ifinerrors			
0x0048	[31:0]	ifinerrors_0	接收的错帧数 (包含 CRC 错、超短帧、超长帧和传输错)。
0x0148	[31:0]	ifinerrors_1	接收的错帧数 (包含 CRC 错、超短帧、超长帧和传输错)。
ifinmulticast			
0x004C	[31:0]	ifinmulticast_0	接收的多播帧数 (包含错帧)。
0x014C	[31:0]	ifinmulticast_1	接收的多播帧数 (包含错帧)。
ifinbroadcast			
0x0050	[31:0]	ifinbroadcast_0	接收的广播帧数 (包含错帧)。
0x0150	[31:0]	ifinbroadcast_1	接收的广播帧数 (包含错帧)。
dot3alignmenterr			
0x0054	[31:0]	dot3alignmenterr_0	接收到的奇数个 nibble 的 CRC 错帧。
0x0154	[31:0]	dot3alignmenterr_1	接收到的奇数个 nibble 的 CRC 错帧。
dot3fcserr			



DIO 地址	比特	名称	描述
0x0058	[31:0]	dot3fcserr_0	接收到的偶数个 nibble 的 CRC 错帧。
0x0158	[31:0]	dot3fcserr_1	接收到的偶数个 nibble 的 CRC 错帧。
dot3internalrecerr			
0x005C	[31:0]	dot3internalrece rr_0	接收的错帧数 (包含 Overflow 错、超短包错和传输错)。
0x015C	[31:0]	dot3internalrece rr_1	接收的错帧数 (包含 Overflow 错、超短包错和传输错)。
dot3inpause			
0x0060	[31:0]	dot3inpause_0	接收的已知操作码的流控帧数。
0x0160	[31:0]	dot3inpause_1	接收的已知操作码的流控帧数。
dot3unkpause			
0x0064	[31:0]	dot3unkpause_0	接收的未知操作码的流控帧数。
0x0164	[31:0]	dot3unkpause_1	接收的未知操作码的流控帧数。
dot3dribble			
0x0068	[31:0]	dot3dribble_0	接收的奇数个 nibble 的 CRC 正确的帧。
0x0168	[31:0]	dot3dribble_1	接收的奇数个 nibble 的 CRC 正确的帧。
octets			
0x006C	[31:0]	octets_0	接收字节计数 (含正确帧和错误帧, 但 Preamble 字节以及未检测到有效 SFD 的帧中的字节不计在内)。
0x016C	[31:0]	octets_1	接收字节计数 (含正确帧和错误帧, 但 Preamble 字节以及未检测到有效 SFD 的帧中的字节不计在内)。
pkts			
0x0070	[31:0]	pkts_0	所有帧的计数, 包含正确帧和错误帧。
0x0170	[31:0]	pkts_1	所有帧的计数, 包含正确帧和错误帧。
ifinoctets			
0x0074	[31:0]	ifinoctets_0	所有接收字节的计数, 包含正确帧、错误帧以及 Preamble 中的字节。
0x0174	[31:0]	ifinoctets_1	所有接收字节的计数, 包含正确帧、错误帧以及 Preamble 中的字节。



DIO 地址	比特	名称	描述
broadcastpkts_tx			
0x0080	[31:0]	broadcastpkts_tx_0	发送的正确的广播帧的计数, 不含重传的广播帧。
0x0180	[31:0]	broadcastpkts_tx_1	发送的正确的广播帧的计数, 不含重传的广播帧。
multicastpkts_tx			
0x0084	[31:0]	multicastpkts_tx_0	发送的正确的多播帧的计数, 不含重传的多播帧。
0x0184	[31:0]	multicastpkts_tx_1	发送的正确的多播帧的计数, 不含重传的多播帧。
ifoutdiscards			
0x0088	[31:0]	ifoutdiscards_tx_0	帧发送的过程中, TXFIFO 下溢出事件的累计次数。
0x0188	[31:0]	ifoutdiscards_tx_1	帧发送的过程中, TXFIFO 下溢出事件的累计次数。
ifouterrors			
0x008C	[31:0]	ifouterrors_tx_0	发送过程中, 所有错帧的计数, 包含重传帧。
0x018C	[31:0]	ifouterrors_tx_1	发送过程中, 所有错帧的计数, 包含重传帧。
ifoutucastpkts			
0x0090	[31:0]	ifoutucastpkts_tx_0	发送的正确或错误的单播帧的计数, 不含重传的单播帧。
0x0190	[31:0]	ifoutucastpkts_tx_1	发送的正确或错误的单播帧的计数, 不含重传的单播帧。
ifoutnucastpkts			
0x0094	[31:0]	ifoutnucastpkts_tx_0	发送的正确或错误的非单播帧的计数, 不含重传的非单播帧。
0x0194	[31:0]	ifoutnucastpkts_tx_1	发送的正确或错误的非单播帧的计数, 不含重传的非单播帧。
ifoutmulticast			
0x0098	[31:0]	ifoutmulticast_tx_0	发送的正确或错误的多播帧的计数, 不含重传的多播帧。



DIO 地址	比特	名称	描述
0x0198	[31:0]	ifoutmulticast_tx_1	发送的正确或错误的多播帧的计数，不含重传的多播帧。
ifoutbroadcast			
0x009C	[31:0]	ifoutbroadcast_tx_0	发送的正确或错误的广播帧的计数，不含重传的广播帧。
0x019C	[31:0]	ifoutbroadcast_tx_1	发送的正确或错误的广播帧的计数，不含重传的广播帧。
pkts64_tx			
0x00A0	[31:0]	pkts64_tx_0	发送帧长为 64 字节的帧的计数，包含正确帧和错误帧（不含重传帧）。
0x01A0	[31:0]	pkts64_tx_1	发送帧长为 64 字节的帧的计数，包含正确帧和错误帧（不含重传帧）。
pkts65_127_tx			
0x00A4	[31:0]	pkts65_127_tx_0	发送帧长为 65 到 127 字节的帧的计数，包含正确帧和错误帧（不含重传帧）。
0x01A4	[31:0]	pkts65_127_tx_1	发送帧长为 65 到 127 字节的帧的计数，包含正确帧和错误帧（不含重传帧）。
pkts128_255_tx			
0x00A8	[31:0]	pkts128_255_tx_0	发送帧长为 128 到 255 字节的帧的计数，包含正确帧和错误帧（不含重传帧）。
0x01A8	[31:0]	pkts128_255_tx_1	发送帧长为 128 到 255 字节的帧的计数，包含正确帧和错误帧（不含重传帧）。
pkts256_511_tx			
0x00AC	[31:0]	pkts256_511_tx_0	发送帧长为 256 到 511 字节的帧的计数，包含正确帧和错误帧（不含重传帧）。
0x01AC	[31:0]	pkts256_511_tx_1	发送帧长为 256 到 511 字节的帧的计数，包含正确帧和错误帧（不含重传帧）。
pkts512_1023_tx			
0x00B0	[31:0]	pkts512_1023_tx_0	发送帧长为 512 到 1023 字节的帧的计数，包含正确帧和错误帧（不含重传帧）。
0x01B0	[31:0]	pkts512_1023_tx_1	发送帧长为 512 到 1023 字节的帧的计数，包含正确帧和错误帧（不含重传帧）。
pkts1024_1518_tx			



DIO 地址	比特	名称	描述
0x00B4	[31:0]	pkts1024_1518_tx_0	发送帧长为 1024 到 1518 字节的帧的计数, 包含正确帧和错误帧 (不含重传帧)。
0x01B4	[31:0]	pkts1024_1518_tx_1	发送帧长为 1024 到 1518 字节的帧的计数, 包含正确帧和错误帧 (不含重传帧)。
pkts_1518_tx			
0x00B8	[31:0]	pkts_1518_tx_0	发送帧长大于 1518 字节的帧的计数, 包含正确帧和错误帧 (不含重传帧)。
0x01B8	[31:0]	pkts_1518_tx_1	发送帧长大于 1518 字节的帧的计数, 包含正确帧和错误帧 (不含重传帧)。
dot3singlecol			
0x00BC	[31:0]	dot3singlecol_0	发生一次冲突后发送成功的帧数目。
0x01BC	[31:0]	dot3singlecol_1	发生一次冲突后发送成功的帧数目。
dot3multiplecol			
0x00C0	[31:0]	dot3multiplecol_0	发生多次 (一次以上, 不包含一次) 冲突后发送成功的帧数目。
0x01C0	[31:0]	dot3multiplecol_1	发生多次 (一次以上, 不包含一次) 冲突后发送成功的帧数目。
dot3latecol			
0x00C4	[31:0]	dot3latecol_0	发生 Late collision 的帧数目。
0x01C4	[31:0]	dot3latecol_1	发生 Late collision 的帧数目。
dot3excessivecol			
0x00C8	[31:0]	dot3excessivecol_0	由于重传次数大于 15 而丢弃的帧数目。
0x01C8	[31:0]	dot3excessivecol_1	由于重传次数大于 15 而丢弃的帧数目。
dot3colcnt			
0x00CC	[31:0]	dot3colcnt_0	冲突次数等于冲突阈值的帧数。 可以通过 SF_STS_REG0[24:21]配置此阈值。
0x01CC	[31:0]	dot3colcnt_1	冲突次数等于冲突阈值的帧数。 可以通过 SF_STS_REG4[24:21]配置此阈值。
dot3defer			
0x00D0	[31:0]	dot3defer_0	由于网络忙而被延迟转发的帧数。



DIO 地址	比特	名称	描述
0x01D0	[31:0]	dot3defer_1	由于网络忙而被延迟转发的帧数。
pkts_tx			
0x00D4	[31:0]	pkts_tx_0	全部发帧的计数, 含正确帧和错误帧, 但不含重传帧。
0x01D4	[31:0]	pkts_tx_1	全部发帧的计数, 含正确帧和错误帧, 但不含重传帧。
Reserved			
0x00D8	[31:0]	octets_tx_0	发送字节计数 (含重传帧以及正确帧和错误帧, 但 Preamble 字节不计在内)。
0x01D8	[31:0]	octets_tx_1	发送字节计数 (含重传帧以及正确帧和错误帧, 但 Preamble 字节不计在内)。
collisions			
0x00DC	[31:0]	collisions_0	Collision 发生的次数, 不含队列满导致的冲突次数。
0x01DC	[31:0]	collisions_1	Collision 发生的次数, 不含队列满导致的冲突次数。
Reserved			
0x00E0	[31:0]	ifoutoctets_0	发送的全部字节, 包含所有 Preamble 的字节以及正确帧、错误帧和重传帧的全部字节数。
0x01E0	[31:0]	ifoutoctets_1	发送的全部字节, 包含所有 Preamble 的字节以及正确帧、错误帧和重传帧的全部字节数。
dot3outpause			
0x00E4	[31:0]	dot3outpause_0	发送的流控帧数目。
0x01E4	[31:0]	dot3outpause_1	发送的流控帧数目。

18.7.1 ANYPORT 帧配置表

ANYPORT 表操作说明

ANYPORT 表使用地址空间为 0x0200~0x02FC, 每个表项使用一个地址, 共对应 64 个表项。上行口发送到 CPU 端口的 ANYPORT 报文分为 2 组进行帧率限制, 前 32 个表项为第 0 组, 后 32 个表项为第 1 组。



ANYPORT 表项数据结构

表18-12 ANYPORT 表项数据结构

比特	名称	描述
[31]	protocol_en1	上行口协议识别使能。 0: 不使能; 1: 使能。
[30]	port_en1	上行口协议端口识别使能。 0: 不使能; 1: 使能。
[29:28]	ctrl1	上行口 ANYPORT 帧转发控制: 00: 丢弃; 10: 发往 CPU 端口; 01: 发往下行口; 11: 同时发往 CPU 端口和下行口。
[27]	protocol_en0	下行口协议识别使能。 0: 不使能; 1: 使能。
[26]	port_en0	下行口协议端口识别使能。 0: 不使能; 1: 使能。
[25:24]	ctrl0	下行口 ANYPORT 帧转发控制: 00: 丢弃; 10: 发往 CPU 端口; 01: 发向上行口; 11: 同时发往 CPU 端口和上行口。
[23:16]	protocol	CPU 配置待识别 IP 协议类型。
[15:0]	port	CPU 配置待识别协议目的端口号 (仅识别 TCP/UDP 目的端口号)。

注: 此处配置数据需为大端 (big endian) 格式。

18.7.2 ANYTYPE 帧配置表

ANYTYPE 表操作说明

ANYTYPE 表使用地址空间为 0x0300~0x037C, 每个表项宽度为 128 位, 使用 4 个地址, 高 bit 对应高地址, 共对应 8 个表项。上行口发送到 CPU 端口的 ANYTYPE 报文可分为 2 组进行帧率限制, 前 4 个表项为第 0 组, 后 4 个表项为第 1 组。



ANYTYPE 表项数据结构

表18-13 ANYTYPE 表项数据结构

比特	名称	描述
[31]	protocol_en1	上行口协议识别使能。 0: 不使能; 1: 使能。
[30]	port_en1	上行口协议端口识别使能。 0: 不使能; 1: 使能。
[29:28]	ctrl1	上行口 ANYTYPE 帧转发控制: 00: 丢弃; 10: 发往 CPU 端口; 01: 发往下行口; 11: 同时发往 CPU 端口和下行口。
[27]	protocol_en0	下行口协议识别使能。 0: 不使能; 1: 使能。
[26]	port_en0	下行口协议端口识别使能。 0: 不使能; 1: 使能。
[25:24]	ctrl0	下行口 ANYTYPE 帧转发控制: 00: 丢弃; 10: 发往 CPU 端口; 01: 发往上行口; 11: 同时发往 CPU 端口和上行口。
[23:16]	protocol	CPU 配置待识别 IP 协议类型。
[15:0]	port	CPU 配置待识别协议目的端口号 (仅识别 TCP/UDP 目的端口号)。

注: 此处配置数据需为大端 (big endian) 格式。

18.7.3 VLAN 表

VLAN 表操作说明

VLAN 表使用地址空间为 0x0380~0x039C, 每个表项宽度为 32 位, 使用一个地址, 共对应 8 个表项。



VLAN 表项数据结构

表18-14 VLAN 表项数据结构

比特	名称	描述
[31:28]	Reserved	保留。
[27:16]	VID	当前 VLAN 对应的 VLAN ID。
[15:11]	Reserved	保留。
[10:8]	vpri	当前 VLAN 对应的 VLAN 优先级。
[7:6]	Reserved	保留。
[5:3]	vmem	当前 VLAN 端口成员。 0: 对应端口不是当前 VLAN 的成员; 1: 对应端口是当前 VLAN 的成员。 3 位从高到低分别对应 CPU 端口、上行口、下行口。
[2:0]	vtag	当前 VLAN 输出 tag 控制。 0: 当前 VLAN 的帧从对应端口按 VLAN 指定格式输出时不带 tag 输出; 1: 当前 VLAN 的帧从对应端口按 VLAN 指定格式输出时带 VLAN tag。 3 位从高到低分别对应 CPU 端口、上行口、下行口。



19 加密单元

关于本章

本章描述内容如下表所示。

标题	内容
19.1 DES 加密单元	详细描述 DES 加密单元。
19.2 AES 加密单元	详细描述 AES 加密单元。



19.1 DES 加密单元

本节主要描述 DES 加密单元。

19.1.1 概述

DES (Data Encryption Standard) 是一种分组对称加密算法。它可以对固定长度为 64 位的明文进行加密, 而且该算法的加解密使用相同的密钥。为了达到更高的安全强度, 可以通过调用 3DES 运算来实现。

19.1.2 特点

DES 单元有以下特点:

- 支持 DES 加解密运算和 3DES 加解密运算
- 支持 DES 的 4 种工作模式: ECB、CBC、CFB 和 OFB, 在 CFB 和 OFB 工作模式下, 可选择 1、8、64 位的位流操作
- 支持 3DES 的 4 种工作模式: ECB、CBC、CFB 和 OFB, 在 CFB 和 OFB 工作模式下, 可选择 1、8、64 位的位流操作
- 支持 3DES 运算的 2 个密钥方案和 3 个密钥方案
- 提供中断功能

19.1.3 工作方式

在进行操作的时候, 64 位的密钥实际上只有 56 位是有效的, 其中第 8、16、24、32、40、48、56、64 位是奇偶校验位, 在算法中无作用。解密操作与加密操作本质上是相同的, 将 64 位的密文作为输入, 加密用的密钥作为输入, 进行解密操作。

DES 操作流程如[图 19-1](#)所示。

图19-1 DES 操作流程图

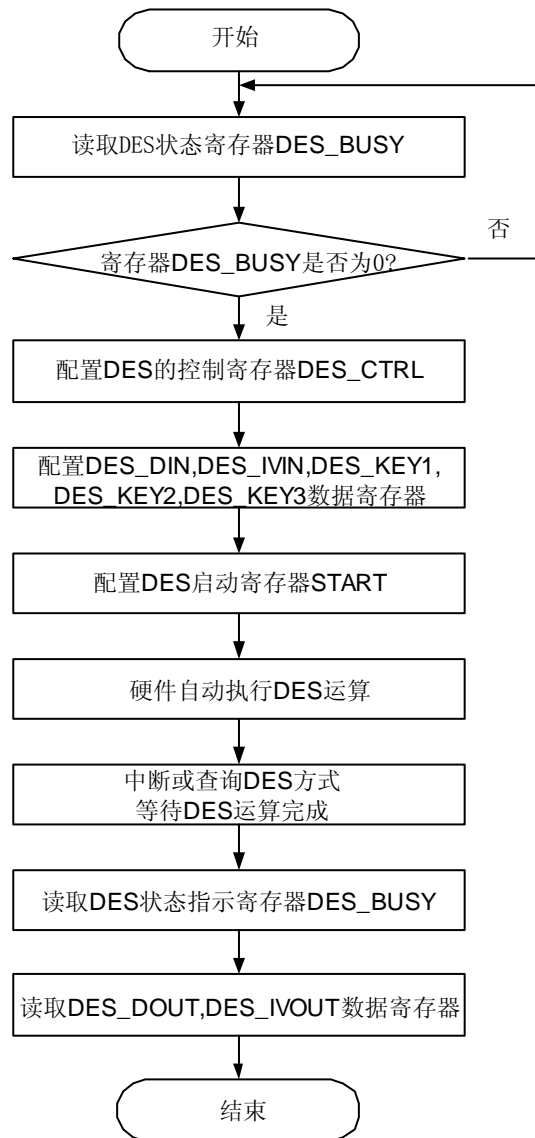


图 19-1 描述了对 DES 加密单元进行操作的流程。如果是执行多个分组的运算，除了反复执行以上的流程，还注意以下事项：

- 如果是相同的密钥，则除了第 1 个分组，其它分组运算时，都不需要配置密钥寄存器。
- 如果执行 CBC、CFB 或 OFB 运算，下一个分组的 IV 向量，是上一个分组的 IV 向量输出，不需要软件另外计算。
- 对于选择执行 3DES 的两个密钥方案，则需要将第 1 个 64 位的密钥配置到 DES_KEY1 和 DES_KEY3 寄存器中，将第 2 个 64 位的密钥配置到 DES_KEY2 寄存器中。

19.1.4 寄存器概览

DES 单元的寄存器地址范围：0x101F_B000~0x101F_BFFF。



表19-1 DES 寄存器概览（基址是 0x101F_B000）

偏移地址	名称	描述	页码
0x0	DES_DIN[63:0]	DES 单元的 64 位分组输入。	19-4
0x8	DES_IVIN[63:0]	DES 单元的向量 IV 分组的输入（ECB 操作模式下无需配置）。	19-5
0x10	DES_KEY1[63:0]	DES 单元用于 DES/3DES 操作的第 1 个密钥输入。	19-6
0x18	DES_KEY2[63:0]	DES 单元用于 3DES 操作时的第 2 个密钥输入。	19-7
0x20	DES_KEY3[63:0]	DES 单元用于 3DES 操作时的第 3 个密钥输入。	19-7
0x28	DES_CTRL[6:0]	控制 DES 操作的寄存器。	19-8
0x2C	DES_DOUT[63:0]	DES 单元 64 位处理的结果输出。	19-9
0x34	DES_IVOUT[63:0]	DES 单元操作完成之后的向量 IV 的输出（ECB 操作模式下无需读取）。	19-10
0x3C	INT_DES[0]	DES 单元运算中断寄存器，读清。CPU 读后就将其清 0。	19-11
0x40	DES_BUSY[0]	DES 加解密运算是否正在进行的状态指示寄存器。	19-11
0x44	START	DES 单元运算启动寄存器，该寄存器是只写寄存器。往该地址中写任何值都将启动加密或解密操作。	19-11

19.1.5 寄存器描述

分组输入寄存器(DES_DIN)

该寄存器是模块的 64 位分组的输入寄存器。根据工作模式和位宽的选择，具体意义有所不同，如表 19-2 所示。

表19-2 DES_DIN 在不同的工作模式和位宽下的意义

工作模式	位宽	描述
DES/3DES 采用 ECB 或 CBC 工作模式	-	寄存器 DES_DIN[63:0]存放 64 比特待运算数据。
DES/3DES 采用 CFB 或 OFB 模式	1 比特位流处理	寄存器 DES_DIN[63]存放 1 比特待运算数据，其余比特无效。



工作模式	位宽	描述
	8 比特位流处理	寄存器 DES_DIN[63:56]存放 8 比特待运算数据。
	64 比特位流处理	寄存器 DES_DIN[63:0]存放 64 比特待运算数据。

- 偏移地址：
 - DES_DIN_L: 0x000
 - DES_DIN_H: 0x004
- 操作方式: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[63:32]	DES_DIN_H	DES 单元的 64 比特分组输入的高 32 位数据。
[31:0]	DES_DIN_L	DES 单元的 64 比特分组输入的低 32 位数据。

向量 IV 输入分组寄存器 (DES_IVIN)

DES_IVIN 寄存器是 DES 的初始向量分组的输入寄存器。根据工作模式的选择，可以确定是否需要进行配置，如表 19-3 所示。

表19-3 DES 单元中工作模式和配置 DES_IVIN 的情况

工作模式	配置 DES_IVIN 情况
CBC、CFB 或 OFB 工作模式	需要配置 DES_IVIN
ECB 工作模式	不需要配置 DES_IVIN

- 偏移地址：
 - DES_IVIN_L: 0x008
 - DES_IVIN_H: 0x00C
- 操作方式: R/W
- 复位值: 0x0
- 复位方式: h/s



比特	名称	描述
[63:32]	DES_IVIN_H	DES 单元的向量 IV 分组输入的高 32 位数据。
[31:0]	DES_IVIN_L	DES 单元的向量 IV 分组输入的低 32 位数据。

注：ECB 操作模式下，不需要配置 DES_IVIN 寄存器。

第 1 个密钥寄存器 (DES_KEY1)

寄存器 DES_KEY1 存放 DES/3DES 的第 1 个密钥。

- 当进行 DES 运算时，寄存器 DES_KEY1 存放运算用的密钥。
- 当进行 3DES 运算时，寄存器 DES_KEY1 存放运算用的第 1 个密钥。

根据密钥的排列控制字的选择不同，各个位的值有所不同。具体内容如表 19-4 所示。

表19-4 DES 单元中密钥与 DES_KEY1 的关系

排列控制字方式	密钥与 DES_KEY1 的关系
高字节在前方式	key[63:0] = DES_KEY1[63:0] 。
低字节在前方式	key[63:56] = DES_KEY1[7:0]; key[55:48] = DES_KEY1[15:8]; key[47:40] = DES_KEY1[23:16]; key[39:32] = DES_KEY1[31:24]; key[31:24] = DES_KEY1[39:32]; key[23:16] = DES_KEY1[47:40]; key[15:8] = DES_KEY1[55:48]; key[7:0] = DES_KEY1[63:56]。

- 偏移地址：
 - DES_KEY1_L: 0x010
 - DES_KEY2_H: 0x014
- 操作方式：R/W
- 复位值：0x0
- 复位方式：h/s

比特	名称	描述
[63:32]	DES_KEY1_H	DES 单元的第 1 个密钥输入的高 32 位数据。
[31:0]	DES_KEY1_L	DES 单元的第 1 个密钥输入的低 32 位数据。



第 2 个密钥寄存器 (DES_KEY2)

寄存器 DES_KEY2 存放 3DES 的第 2 个密钥。

- 当进行 DES 运算时，不需要配置该寄存器。
- 当进行 3DES 运算时，寄存器 DES_KEY2 存放运算用的第 2 个密钥。

根据密钥的排列控制字的选择不同，各个位的值有所不同。具体内容如表 19-5 所示。

表19-5 DES 单元中密钥与 DES_KEY2 的关系

排列控制字方式	密钥与 DES_KEY2 的关系
高字节在前方式	key[63:0] = DES_KEY2[63:0]
低字节在前方式	key[63:56] = DES_KEY2[7:0]; key[55:48] = DES_KEY2[15:8]; key[47:40] = DES_KEY2[23:16]; key[39:32] = DES_KEY2[31:24]; key[31:24] = DES_KEY2[39:32]; key[23:16] = DES_KEY2[47:40]; key[15:8] = DES_KEY2[55:48]; key[7:0] = DES_KEY2[63:56]。

- 偏移地址：
 - DES_KEY2_L: 0x018
 - DES_KEY2_H: 0x01C
- 操作方式: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[63:32]	DES_KEY2_H	DES 单元的第 2 个密钥输入的高 32 位数据。
[31:0]	DES_KEY2_L	DES 单元的第 2 个密钥输入的低 32 位数据。

第 3 个密钥寄存器 (DES_KEY3)

- 如果 3DES 使用 3 个密钥方案，寄存器 DES_KEY3 存放运算的第 3 个密钥；
- 如果使用两个密钥方案，寄存器 DES_KEY3 存放运算用的第 1 个密钥。

根据密钥的排列控制字的选择不同，各个位的值有所不同。具体内容如表 19-6 所示。



表19-6 DES 单元中密钥与 DES_KEY3 的关系

排列控制字方式	密钥与 DES_KEY3 的关系
高字节在前方式	key[63:0] = DES_KEY3[63:0] 。
低字节在前方式	key[63:56] = DES_KEY3[7:0]; key[55:48] = DES_KEY3[15:8]; key[47:40] = DES_KEY3[23:16]; key[39:32] = DES_KEY3[31:24]; key[31:24] = DES_KEY3[39:32]; key[23:16] = DES_KEY3[47:40]; key[15:8] = DES_KEY3[55:48]; key[7:0] = DES_KEY3[63:56]。

- 偏移地址：
 - DES_KEY3_L: 0x020
 - DES_KEY3_H: 0x024
- 操作方式: R/W
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[63:32]	DES_KEY3_H	DES 单元的第 3 个密钥输入的高 32 位数据。
[31:0]	DES_KEY3_L	DES 单元的第 3 个密钥输入的低 32 位数据。

单元控制寄存器(DES_CTRL)

DES_CTRL 寄存器是完成对 DES/3DES 运算的各种寄存器的配置。

- 偏移地址: 0x028
- 操作方式: R/W
- 复位值: 0x0
- 复位方式: h/s



比特	名称	描述
[6]	DES_CTRL[6]	密钥的存放控制字。 0: key 的存放, 以高字节在前的方式; 1: key 的存放, 以低字节在前的方式。
[5:4]	DES_CTRL[5:4]	初始向量 IV 的移位位宽的控制字: 00: 64 位; 01: 8 位; 10: 1 位; 11: 保留 (具体处理按 64 位 bit 处理)。
[3:2]	DES_CTRL[3:2]	工作模式选择控制字。 00: ECB 模式; 01: CBC 模式; 10: CFB 模式; 11: OFB 模式。
[1]	DES_CTRL[1]	加、解密控制字。 0: 加密; 1: 解密。
[0]	DES_CTRL[0]	DES/3DES 运算选择。 0: DES 操作; 1: 3DES 操作。

分组输出寄存器 (DES_DOUT)

DES_DOUT 寄存器是 64 位的结果输出寄存器, 根据工作模式和位宽的选择, 各位表示的意义有所不同, 如表 19-7 所示。

表19-7 DES_DOUT 在不同的工作模式和位宽下的意义

工作模式	位宽	描述
DES/3DES 采用 ECB 或 CBC 工作模式	-	寄存器 DES_DOUT [63:0]存放 64 比特运算结果。
DES/3DES 采用 CFB 或 OFB 模式	1 比特位流处理	寄存器 DES_DOUT [63]存放 1 比特待运算结果, 其余比特为 0。
	8 比特位流处理	寄存器 DES_DOUT[63:56]存放 8 比特待运算结果, 其余比特为 0。
	64 比特位流处理	寄存器 DES_DOUT[63:0]存放 64 比特待运算结果。



- 偏移地址：
 - DES_DOUT_L: 0x02C
 - DES_DOUT_H: 0x030
- 操作方式: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[63:32]	DES_DOUT_H	DES 单元 64bit 处理的结果输出的高 32 位。
[31:0]	DES_DOUT_L	DES 单元 64bit 处理的结果输出的低 32 位。

注: DES_DOUT 寄存器为只读寄存器。

向量 IV 输出分组寄存器 (DES_IVOUT)

DES_IVOUT 寄存器是 64 位的结果输出寄存器。DES 单元中工作模式和配置 DES_IVOUT 的关系如表 19-8 所示。

表19-8 DES 单元中工作模式和配置 DES_IVOUT 的情况

工作模式	DES_IVOUT 中的值情况
CBC、CFB 或 OFB 工作模式	读取 DES_IVOUT 寄存器中的值，作为下一个分组运算时，即直接作为 DES_IVIN 寄存器的值。
ECB 工作模式	寄存器 DES_IVOUT 中的值无效。

- 偏移地址：
 - DES_IVOUT_L: 0x034
 - DES_IVOUT_H: 0x038
- 操作方式: R
- 复位值: 0x0
- 复位方式: h/s

比特	名称	描述
[63:32]	DES_IVIN_H	DES 单元操作完成之后的向量 IV 输出的高 32 位。
[31:0]	DES_IVIN_L	DES 单元操作完成之后的向量 IV 输出的低 32 位。

注: DES_IVOUT 寄存器为只读寄存器。



中断控制寄存器(INT_DES)

INT_DES 寄存器是 DES 加密单元的中断寄存器。该寄存器是读清寄存器，一旦检测到 CPU 对该地址的读操作之后，即将该寄存器中的值清零。

- 操作方式：RC
- 复位值：0x0
- 复位方式：h/s

偏移地址	名称	描述
0x03C	INT_DES[0]	DES 单元运算中断寄存器。 该寄存器是读清寄存器。

状态控制寄存器(DES_BUSY)

DES_BUSY 寄存器是 DES 单元的状态控制寄存器。DES 的状态描述如表 19-9 所示。

表19-9 DES 的状态描述

操作	DES_BUSY[0]	描述
DES 模块启动之后	置 1	表示正在进行运算
当运算完成后	置 0	表示操作完成

- 操作方式：R
- 复位值：0x0
- 复位方式：h/s

偏移地址	名称	描述
0x040	DES_BUSY[0]	DES 加解密运算正在进行的状态指示寄存器。 0：表示模块完成操作，或空闲状态； 1：表示模块正在执行运算。

启动控制寄存器(START)

START 寄存器是 DES 加密单元的启动控制寄存器。CPU 一旦往该寄存器执行写操作之后，DES 单元将执行运算。

- 操作方式：W
- 复位值：0x0
- 复位方式：h/s



偏移地址	名称	描述
0x044	START	DES 单元运算启动寄存器，往该地址中写任何值都将启动加密或解密操作。 该寄存器是只写寄存器。

19.2 AES 加密单元

本节主要描述 AES 加密单元。

19.2.1 概述

AES 是一种新的高级加密算法标准，用来替代 DES。它可以对固定长度为 128 位的分组进行加、解密处理，密钥的长度是 128 位、192 位和 256 位。该模块实现的是对 128 位分组使用 128 位密钥进行加、解密处理。对 AES 的实现符合 FIPS 197 标准的要求，对 AES 的工作模式的实现符合 NIST special 800-38A 要求。

19.2.2 特点

AES 单元主要有以下特点：

- 支持 AES 加解密运算
- 支持 AES 的 7 种工作模式：ECB、CBC、1-CFB、8-CFB、128-CFB、OFB 和 CTR
- 支持 128、192 和 256 位的密钥长度

19.2.3 工作方式

AES 加密单元操作流程如图 19-2 所示。

图19-2 AES 加密单元操作流程图

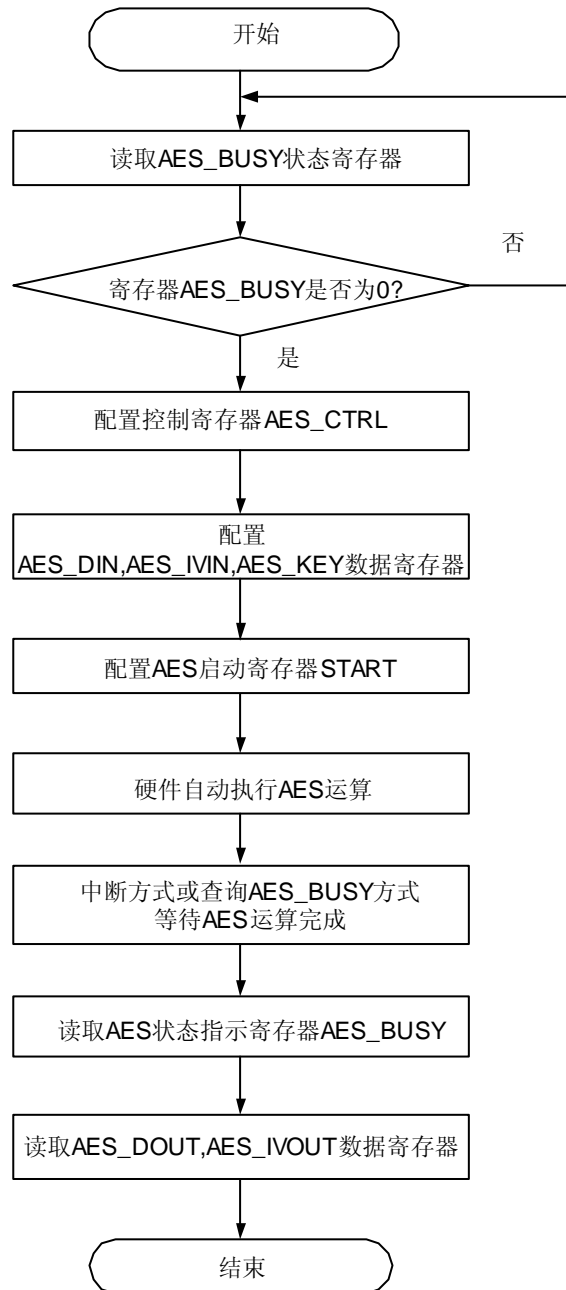


图 19-2 描述了对 AES 加密单元进行一轮操作的流程。如果是执行多个分组的运算，除了反复执行以上的流程，还有注意以下事项：

- 如果是相同的密钥，则除了第 1 个分组，其它分组运算时，都不需要配置密钥寄存器。
- 如果执行 CBC、CFB 或 OFB 运算，下一个分组的 IV 向量，是上一个分组的 IV 向量输出，不需要软件另外计算。



19.2.4 寄存器概览

表19-10 AES 控制寄存器概览（基址是 0x101FE000）

偏移地址	名称	描述	页码
0x0	AES_DIN[31:0]	AES 模块的 128 位分组输入的 0~31 位	19-15
0x4	AES_DIN[63:32]	AES 模块的 128 位分组输入的 32~63 位	19-15
0x8	AES_DIN[95:64]	AES 模块的 128 位分组输入的 64~95 位	19-15
0xC	AES_DIN[127:96]	AES 模块的 128 位分组输入的 96~127 位	19-15
0x10	AES_IVIN [31:0]	AES 模块的向量 IV 分组输入（ECB 工作模式下无需配置）的 0~31 位	19-16
0x14	AES_IVIN [63:32]	AES 模块的向量 IV 分组输入（ECB 工作模式下无需配置）的 32~63 位	19-16
0x18	AES_IVIN [95:64]	AES 模块的向量 IV 分组输入（ECB 工作模式下无需配置）的 64~95 位	19-16
0x1C	AES_IVIN [127:96]	AES 模块的向量 IV 分组输入（ECB 工作模式下无需配置）的 96~127 位	19-16
0x20	AES_KEY [31:0]	AES 模块的密钥输入的 0~31 位	19-16
0x24	AES_KEY [63:32]	AES 模块的密钥输入的 32~63 位	19-16
0x28	AES_KEY [95:64]	AES 模块的密钥输入的 64~95 位	19-16
0x2C	AES_KEY [127:96]	AES 模块的密钥输入的 96~127 位	19-16
0x30	AES_DOUT [31:0]	AES 模块 128 位处理的结果输出的 0~31 位，该寄存器是只读寄存器	19-17
0x34	AES_DOUT [63:32]	AES 模块 128 位处理的结果输出的 32~63 位，该寄存器是只读寄存器	19-17
0x38	AES_DOUT [95:64]	AES 模块 128 位处理的结果输出的 64~95 位，该寄存器是只读寄存器	19-17
0x3C	AES_DOUT [127:96]	AES 模块 128 位处理的结果输出的 96~127 位，该寄存器是只读寄存器	19-17
0x40	AES_IVOUT [31:0]	AES 模块操作完成之后的向量 IV 输出的 0~31 位，该寄存器是只读寄存器	19-17
0x44	AES_IVOUT [63:32]	AES 模块操作完成之后的向量 IV 输出的 32~63 位，该寄存器是只读寄存器	19-17
0x48	AES_IVOUT [95:64]	AES 模块操作完成之后的向量 IV 输出的 64~95 位，该寄存器是只读寄存器	19-17



偏移地址	名称	描述	页码
0x4C	AES_IVOUT [127:96]	AES 模块操作完成之后的向量 IV 输出的 96~127 位，该寄存器是只读寄存器	19-17
0x50	AES_CTRL[3:0]	控制 AES 单元操作的寄存器	19-18
0x54	INT_AES[0]	AES 模块运算中断寄存器。该寄存器是读清寄存器	19-19
0x58	AES_BUSY[0]	AES 加解密运算正在进行的状态指示寄存器，该寄存器是只读寄存器	19-19
0x5C	START	AES 单元运算启动寄存器，该寄存器是只写寄存器	19-19

19.2.5 寄存器描述

分组输入寄存器 (AES_DIN)

该寄存器是模块的 128 位分组的输入寄存器。根据工作模式和位宽的选择，具体意义有所不同，如表 19-11 所示。

表19-11 AES_DIN 在不同的工作模式和位宽下的意义

工作模式	位宽	描述
AES 采用 ECB 或 CBC 或 CTR 工作模式	-	寄存器 AES_DIN[127:0]存放 128 比特待运算数据。
AES 采用 CFB 或 OFB 模式	1 比特位流处理	寄存器 AES_DIN[127]存放 1 比特待运算数据，其余比特无效。
	8 比特位流处理	寄存器 AES_DIN[127:120]存放 8 比特待运算数据，其余比特无效。
	128 比特位流处理	寄存器 AES_DIN[127:0]存放 128 比特待运算数据。

- 操作方式：R/W
- 复位值：0x0
- 复位方式：h/s

偏移地址	名称	描述
0x0	AES_DIN[31:0]	AES 模块的 128 位分组输入的 0~31 位
0x4	AES_DIN[63:32]	AES 模块的 128 位分组输入的 32~63 位



偏移地址	名称	描述
0x8	AES_DIN[95:64]	AES 模块的 128 位分组输入的 64~95 位
0xC	AES_DIN[127:96]	AES 模块的 128 位分组输入的 96~127 位

向量 IV 输入分组寄存器 (AES_IVIN)

该寄存器是模块的初始向量分组的输入寄存器。根据工作模式和位宽的选择，可以确定是否需要进行配置，如表 19-12 所示。

表19-12 AES 单元中工作模式和配置 AES_IVIN 的关系

工作模式	配置 AES_IVIN 情况
CBC、CFB、OFB 或 CTR 工作模式	需要配置寄存器 AES_IVIN
ECB 工作模式	不需要配置 AES_IVIN

- 操作方式：R/W
- 复位值：0x0
- 复位方式：h/s

偏移地址	名称	描述
0x10	AES_IVIN [31:0]	AES 模块的向量 IV 分组输入（ECB 工作模式下无需配置）的 0~31 位
0x14	AES_IVIN [63:32]	AES 模块的向量 IV 分组输入（ECB 工作模式下无需配置）的 32~63 位
0x18	AES_IVIN [95:64]	AES 模块的向量 IV 分组输入（ECB 工作模式下无需配置）的 64~95 位
0x1C	AES_IVIN [127:96]	AES 模块的向量 IV 分组输入（ECB 工作模式下无需配置）的 96~127 位

密钥寄存器 (AES_KEY)

寄存器 AES_KEY 存放 AES 的密钥。

- 操作方式：R/W
- 复位值：0x0
- 复位方式：h/s



偏移地址	名称	描述
0x20	AES_KEY [31:0]	AES 模块的密钥输入的 0~31 位
0x24	AES_KEY [63:32]	AES 模块的密钥输入的 32~63 位
0x28	AES_KEY [95:64]	AES 模块的密钥输入的 64~95 位
0x2C	AES_KEY [127:96]	AES 模块的密钥输入的 96~127 位

分组输出寄存器 (AES_DOUT)

该寄存器是 128 位的结果输出寄存器，根据工作模式和位宽的选择，各位表示的意义有所不同，如表 19-13 所示。

表19-13 AES_DOUT 在不同的工作模式和位宽下的意义

工作模式	位宽	描述
当 AES 采用 ECB、CBC 或 CTR 工作模式	-	寄存器 AES_DOUT[127:0]存放 128 比特运算结果。
当 AES 采用 CFB 或 OFB 模式	1 比特位流处理	寄存器 AES_DOUT[127]存放 1 比特运算结果，其余比特为 0。
	8 比特位流处理	寄存器 AES_DOUT[127:120]存放 8 比特运算结果，其余比特为 0。
	128 比特位流处理	寄存器 AES_DOUT[127:0]存放 128 比特运算结果。

- 操作方式：R
- 复位值：0x0
- 复位方式：h/s

偏移地址	名称	描述
0x30	AES_DOUT [31:0]	AES 模块 128 位处理的结果输出的 0~31 位。
0x34	AES_DOUT [63:32]	AES 模块 128 位处理的结果输出的 32~63 位。
0x38	AES_DOUT [95:64]	AES 模块 128 位处理的结果输出的 64~95 位。
0x3C	AES_DOUT [127:96]	AES 模块 128 位处理的结果输出的 96~127 位。

向量 IV 输出分组寄存器 (AES_IVOUT)

该寄存器是 128 位的结果输出寄存器。



表19-14 AES_IVOUT 在不同的工作模式下的意义

工作模式	描述
AES 采用 ECB 或 CTR 工作模式	寄存器 AES_IVOUT 中的值无效。
其它工作模式	读取 AES_IVOUT 寄存器中的值，作为下一个分组运算时，即直接作为 AES_IVIN 寄存器的值。

- 操作方式：R
- 复位值：0x0
- 复位方式：h/s

偏移地址	名称	描述
0x40	AES_IVOUT[31:0]	AES 模块操作完成之后的向量 IV 输出的 31~0 位。
0x44	AES_IVOUT[63:32]	AES 模块操作完成之后的向量 IV 输出的 32~63 位。
0x48	AES_IVOUT[95:64]	AES 模块操作完成之后的向量 IV 输出的 64~95 位。
0x4C	AES_IVOUT[127:96]	AES 模块操作完成之后的向量 IV 输出的 96~127 位。

单元控制寄存器 (AES_CTRL)

该寄存器是完成对 AES 运算的各种寄存器的配置。

- 操作方式：R/W
- 复位值：4'h0
- 复位方式：h/s

偏移地址	名称	描述
0x50	AES_CTRL[3:1]	工作模式控制寄存器 000: ECB 模式; 100: 128-CFB 模式; 001: CBC 模式; 101: OFB 模式; 010: 1-CFB 模式; 110: CTR 模式; 011: 8-CFB 模式; 111: 保留, 做 ECB 处理。
0x50	AES_CTRL[0]	加、解密控制字: 0: 加密; 1: 解密。



中断控制寄存器 (INT_AES)

该寄存器是 AES 加密单元的中断寄存器，该寄存器是读清寄存器，一旦检测到 CPU 对该地址的读操作之后，即将该寄存器中的值清零。

- 操作方式：RC
- 复位值：0x0
- 复位方式：h/s

偏移地址	名称	描述
0x54	INT_AES[0]	AES 模块运算中断寄存器。

状态控制寄存器 (AES_BUSY)

AES_BUSY 寄存器是 AES 加密单元的状态寄存器。AES 的状态描述如表 19-15 所示。

表19-15 AES 的状态描述

操作	AES_BUSY[0]	描述
AES 模块启动之后	置 1	表示正在进行运算
当运算完成后	置 0	表示操作完成

- 操作方式：R
- 复位值：0x0
- 复位方式：h/s

偏移地址	名称	描述
0x58	AES_BUSY[0]	AES 加解密运算正在进行的状态指示寄存器。

启动控制寄存器 (START)

该寄存器是 AES 加密单元的启动控制寄存器，CPU 一旦往该寄存器中执行写操作之后，AES 单元将执行运算。

- 操作方式：W
- 复位值：0x0
- 复位方式：h/s

偏移地址	名称	描述
0x5C	START	AES 单元运算启动寄存器。



20 模式配置与接口调试

关于本章

本章描述内容如下表所示。

标题	内容
20.1 概述	概括介绍 Hi3510 管脚模式配置及接口调试。
20.2 信号描述	描述模式配置与接口调试时的接口信号。
20.3 工作方式	描述了模式配置、调试接口和调试模式。



20.1 概述

Hi3510 可通过管脚配置工作在不同模式，同时也可支持 ARM 子系统的单独调试、DSP 子系统的单独调试、ARM 与 DSP 子系统的联合调试和 ARM 子系统的实时跟踪调试功能。

20.2 信号描述

本节描述模式配置与接口调试时的接口信号，如表 20-1、表 20-2、表 20-3 所示。

表20-1 模式配置接口信号描述

管脚名	方向	描述
TESTMODE	I	测试或调试模式选择。
FUNCSEL2	I	功能选择管脚 2。
FUNCSEL1	I	功能选择管脚 1。
FUNCSEL0	I	功能选择管脚 0。
SCANEN	I	Hi3510 scan 测试使能，正常工作时必须将该管脚下拉。
BISTCLK	I	Hi3510 BIST 测试时钟输入，正常工作时必须将该管脚下拉。

表20-2 JTAG 接口信号描述

信号名	方向	描述
TCK	I	JTAG 时钟输入，必须外部下拉。
TDI	I	JTAG 数据输入，外部上拉。
TMS	I	JTAG 模式选择输入，外部上拉。
TRSTN	I	JTAG 复位输入，如果进行 ARM926EJ-S 调试，外部上拉；正常工作模式必须外部下拉。
TDO	O	JTAG 数据输出。
RTCK	O	进行 ARM926EJ-S 调试、连接 Multi-ICE 或 RealView-ICE 时的 JTAG 反馈时钟，建议下拉。



表20-3 ETM9 调试接口信号描述

信号名	方向	描述
TRACSEL	I	因部分 GPIO 口与 TRACE 输出信号管脚复用，在使用 ETM9 进行实时跟踪调试时，该管脚要设置为 1'h1。
GPIO3[4]	I/O	Trace 时钟输出（traceclk）。
GPIO3[3]	I/O	Trace 同步输出（tracesync）。
GPIO3[2:0]	I/O	Trace 流水状态输出（分别对应 pipestat2~pipestate0）。
GPIO2[7:0]	I/O	Trace 数据输出（分别对应 tracepkt7~tracepkt0）。

20.3 工作方式

本节描述了模式配置、调试接口和调试模式。

20.3.1 模式配置

Hi3510 有 8 种工作模式，可以通过专用管脚进行设置，如表 20-4 所示。

说明

推荐 Hi3510 正常工作模式配置为 0000。

表20-4 TEST_MODE 的含义

TEST MODE	FUNCSEL [2:0]	含义
0	X00	Hi3510 正常工作，整芯片的 JTAG 测试模式（板级测试）。其中 FUNCSEL2 为： 0: ARM926EJ-S 处理器时钟为 AHB 总线时钟频率的 2 倍。
0	X01	Hi3510 正常工作，可通过 JTAG 对 ARM926EJ-S 进行单独调试模式。其中 FUNCSEL2 为： 0: ARM926EJ-S 处理器时钟为 AHB 总线时钟的 2 倍。
0	X10	Hi3510 正常工作，可通过 JTAG 对 DSP 进行单独调试模式。其中 FUNCSEL2 为： 0: ARM926EJ-S 处理器时钟为 AHB 总线时钟的 2 倍。
0	X11	Hi3510 正常工作，可通过 JTAG 对 ARM926EJ-S、DSP 进行联合调试模式。其中 FUNCSEL2 为： 0: ARM926EJ-S 处理器时钟为 AHB 总线时钟的 2 倍。



TEST MODE	FUNCSEL [2:0]	含义
1	000	Hi3510 处于 ARM926EJ-S、USB、ETM9+SCAN 测试模式，不可正常工作。
1	001	Hi3510 处于 ARM926EJ-S BIST 测试模式，不可正常工作。
1	010	Hi3510 处于 Boundary SCAN 测试模式，不可正常工作。
1	100	Hi3510 处于 PLL 测试模式、不可正常工作。

注：X 在这版本芯片中只能设置为 0,为 1 时芯片不能工作。

20.3.2 调试接口

ETM9

Hi3510 集成了 ETM9 单元，提供 ARM 子系统的实时调试功能接口信号请参见表 20-3。ETM 接口具有以下特性：

- 支持 Medium PLUS 模式，FIFO 大小为 18×8 bit。当出现跟踪溢出时，可使处理器暂停或者放弃跟踪；
- ETM 采用 JTAG 端口进行调试配置、采用 TPA 端口捕获实时跟踪信息；
- 支持半速率；
- 不支持 Mux、DeMux 调试模式；
- 支持 RealView Trace 或者 Multi Trace 硬件调试。

说明

- RealView ICE V1.0.1 和 RealViewTrace（必须和 RealView ICE 搭配用）使用以太网口与 PC 相连，RealView Trace 支持跟踪的时钟频率达 250MHz。
- MultiICE V2.2 和 MultiTrace V1.0（需要和 MultiICE 搭配用）分别用并口和以太网口与 PC 相连，MultiTrace 支持跟踪的时钟频率达到 200MHz。

JTAG

Hi3510 提供的 JTAG 接口符合 IEEE 1149.1 标准，接口信号请参见表 20-2。

Hi3510 的 JTAG 接口可用于 ARM 软件调试、DSP 软件调试和板级测试。

20.3.3 调试模式

单独对 ARM 进行调试

对 ARM926EJ-S 的单独调试可以采用 2 种方式：

- 实时调试
在实时调试的情况下，需要使用到 Hi3510 内嵌的 ETM9+调试结构。
- 非实时调试

完全在主机软件的控制下，对目标软件进行非实时调试

ETM（Embedded Trace Module）是 ARM 提出的一种调试结构，它主要用于在实时条件下跟踪目标软件的运行状况，进而对目标软件进行调试。

ETM 调试结构主要包括 2 个组成部分：

- 跟踪端口（Trace Port）
用来连接主机调试软件，对 ETM 进行配置、信息分析等工作。
- 触发设施
ETM 的主体，用来根据设置的触发条件进行信息的采集、过滤、压缩等工作。

一个采用 ETM 进行调试的系统如图 20-1 所示。

图20-1 采用 ETM 进行调试的系统示例图

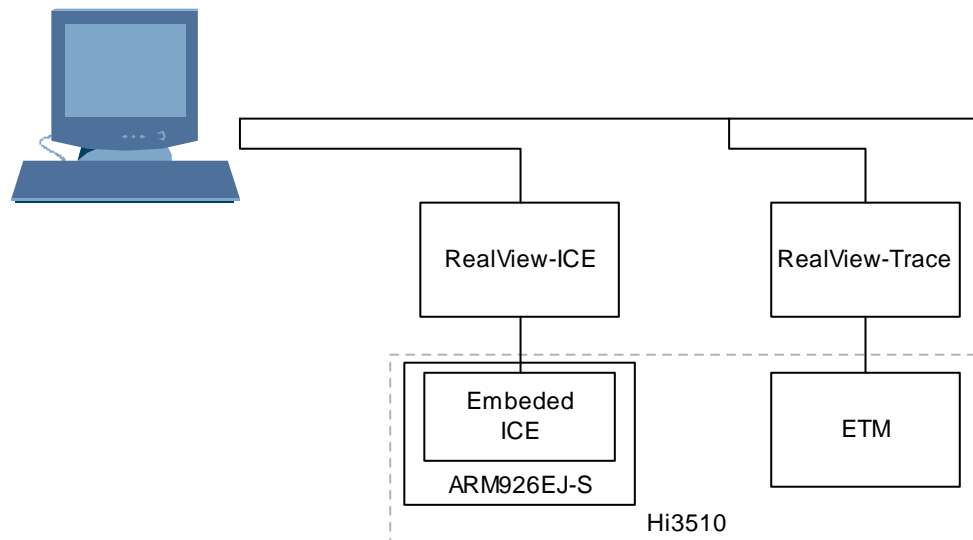


图 20-1 中通过 JTAG 接口单元（如 ARM RealView-ICE）对 ETM 进行初始设置，利用 TPA 将 ETM 捕获到的数据传到主机进行分析。Hi3510 内嵌了中等模式的 ETM9，数据位宽为 8 比特。在采用 ETM 进行调试时，需要将 TESTMODE、FUNCSEL1、FUNCSEL0 分别设置为外部下拉、外部下拉、外部上拉。

在非实时调试的情况下，只需要一个能够支持 JTAG 接口的 ICE 设备（如 RealView-ICE）和主机相连，并采用相应的调试软件即可。

单独对 DSP 进行调试

单独对 DSP 进行调试时，需要调试设备能够支持 DSP 的 JTAG 命令，如 Domain Tech. 公司提供的 USB 调试器、Corelis 公司提供的系列调试器等。采用 JTAG 接口设备对 DSP 程序进行单独调试的系统如图 20-2 所示。



图20-2 单独对 DSP 程序进行调试的系统示例图

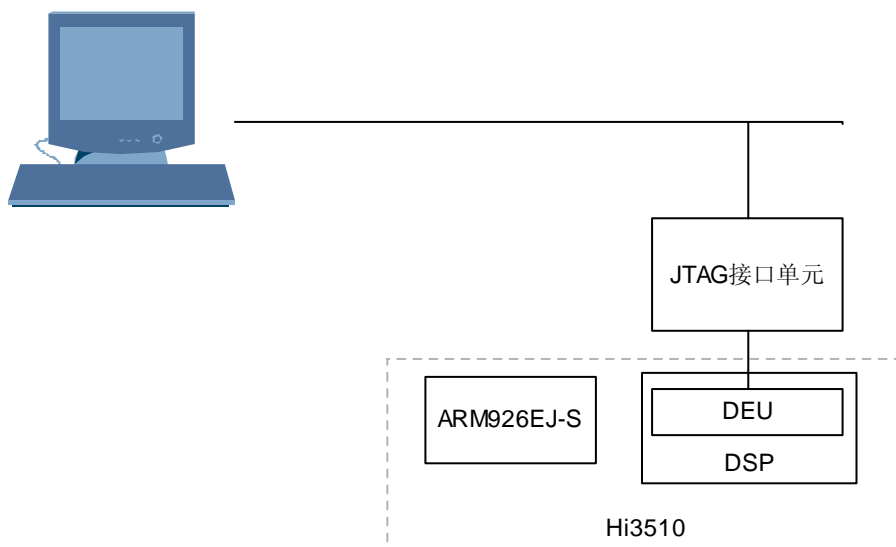


图 20-2 中的 DEU 是 DSP Core 提供的调试结构，它根据 JTAG 接口所传递的信息进行断点的设置和信息的传送等工作，与主机软件（如 LSI 公司的 SDK 软件）一起完成调试功能。在单独对 DSP 进行调试时，需要将 TESTMODE、FUNCSEL1、FUNCSEL0 分别设置为外部下拉、外部上拉、外部下拉。

对 ARM 和 DSP 进行联合调试

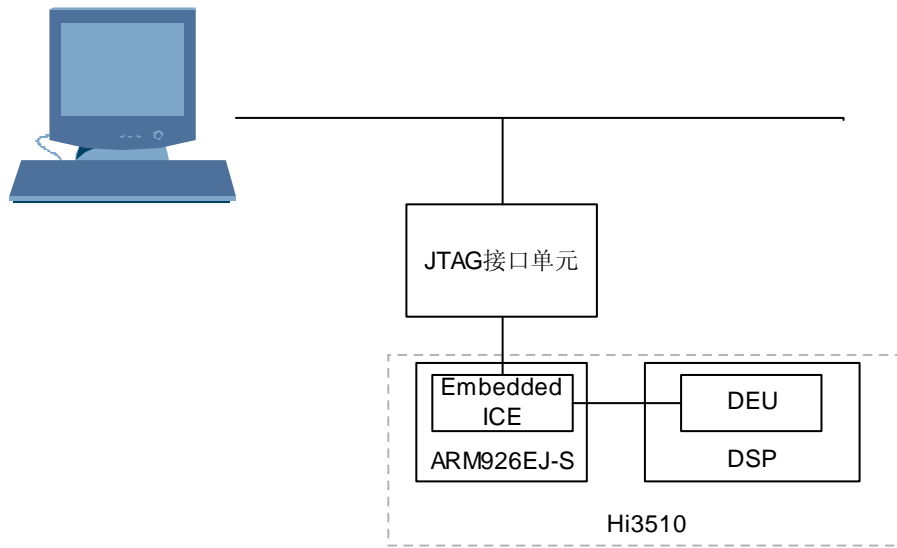
由于在软件模块中经常会涉及到多个 Core 之间的通信，因此，单独对其中的一个 Core 进行调试已经不能满足软件调试的要求。为了能够支持对复杂软件调试的支持，Hi3510 还支持多 Core 联合调试的方式（需要调试软件支持）。

在多 Core 联合调试的模式下，Hi3510 在芯片内部将 ARM926EJ-S 和 DSP 的调试结构通过 JTAG 接口有机的联系起来。通过软件（如 ARM 公司提供的 RVDS 软件），可以对 ARM926EJ-S、DSP 分别设置断点、观察点等，对目标软件进行调试。

对多 Core 进行联合调试的系统示意图如图 20-3 所示。在对 ARM 和 DSP 进行联合调试时，需要将 TESTMODE、FUNCSEL1、FUNCSEL0 分别设置为外部下拉、外部上拉、外部上拉。



图20-3 Dual-Core 调试系统示例图



板级测试模式

Hi3510 除了支持对软件进行调试以外，还支持在单板上的一些测试，如 Hi3510 与其它连接设备的连接性等。板级测试通过标准的 JTAG TAP 控制器实现。在进行板级互连测试时，需要将 TESTMODE、FUNCSEL1、FUNCSEL0 分别设置为外部下拉、外部下拉、外部下拉。



21 时序和参数

关于本章

本章描述内容如下表所示。

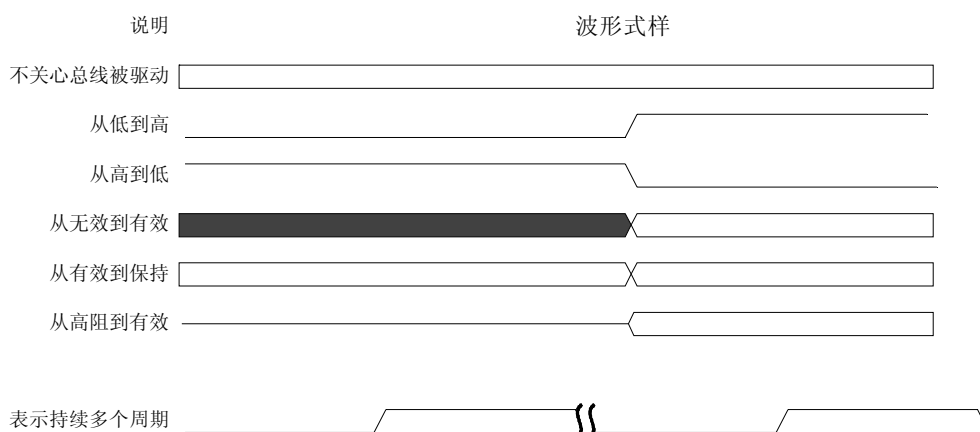
标题	内容
21.1 时序图例	概括介绍时序图例。
21.2 USB 时钟时序和参数	描述时钟及相关信号时序参数。
21.3 存储接口时序参数	描述存储接口时序。
21.4 串行口时序参数	描述串行口时序参数。
21.5 ETM 时序参数	描述 ETM 时序参数。
21.6 SIO 接口时序参数	描述 SIO 接口时序参数。
21.7 VI 视频输入接口时序	描述 VI 视频输入接口时序。
21.8 VO 视频输出接口时序	描述 VO 视频输出接口时序。
21.9 SF 以太网交换 RMII 接口时序 (50MHz)	描述以太网交换 RMII 接口时序参数。



21.1 时序图例

时序图例如图 21-1 所示。

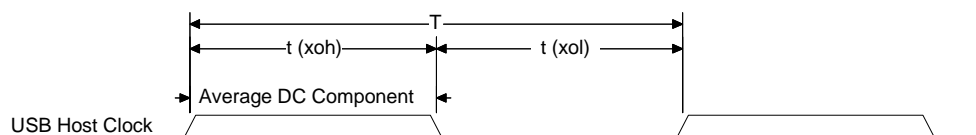
图21-1 时序图例



21.2 USB 时钟时序和参数

USB 时钟时序如图 21-2 所示。

图21-2 USB 时钟时序图



USB 时钟时序参数如表 21-1 所示。

表21-1 USB 时钟时序参数

参数	符号	最小值	典型值	最大值	单位
USB 时钟高电平时间	t (xoh)	TBD	-	TBD	ns
USB 时钟低电平时间	t (xol)	TBD	-	TBD	ns
USB 时钟周期	T (t)	-	20.833	-	ns

参数	符号	最小值	典型值	最大值	单位
USB 时钟频率	1/T (t)	-	48.000	-	MHz
USB 时钟频率稳定度	$\Delta f/f$	-	TBD	-	-

21.3 存储接口时序参数

21.3.1 SDRAM 接口时序参数

SDRAM 接口时序如图 21-3 所示。

图21-3 SDRAM 接口时序

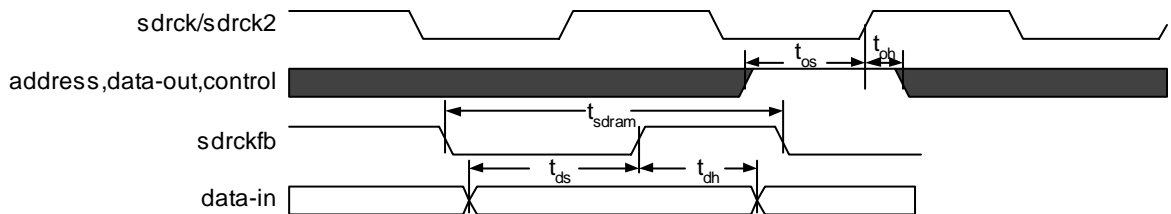


表21-2 SDRAM 接口时序参数列表

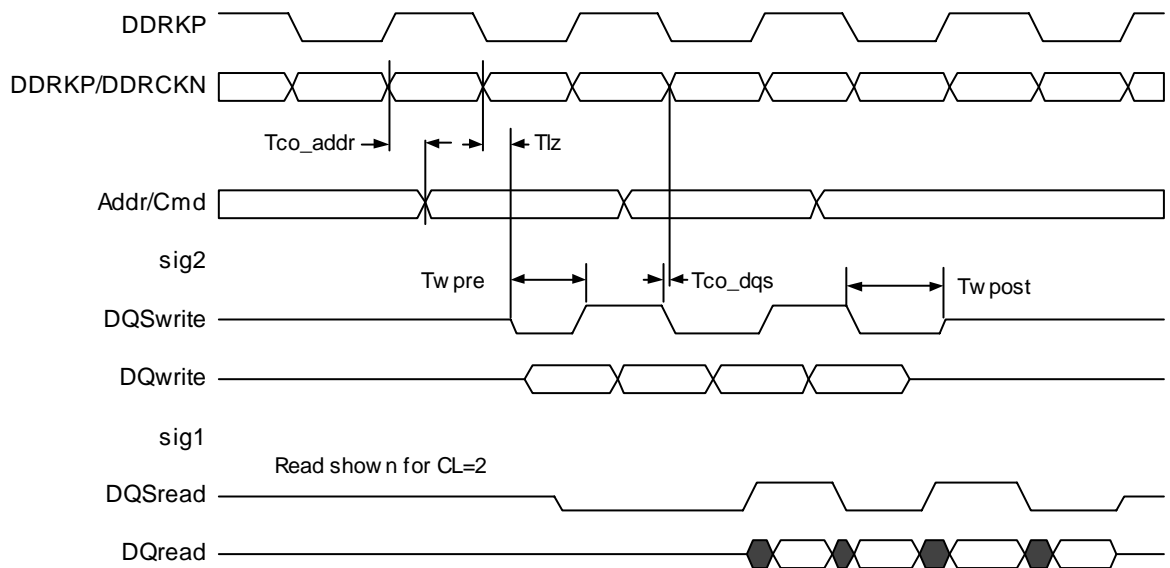
参数	符号	最小值	典型值	最大值	单位
SDRAM 时钟频率	t_{sdram}	-	10	-	ns
地址/数据/控制信号输出建立时间	t_{os}	-	-	6.5	ns
地址/数据/控制信号输出保持时间	t_{oh}	1.5	-	-	ns
输入数据建立时间	t_{ds}	2.0	-	-	ns
输入数据保持时间	t_{dh}	1.0	-	-	ns

21.3.2 DDR 接口时序参数

DDR 接口时序如图 21-4 所示。

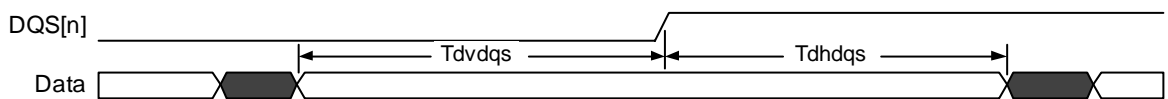


图21-4 DDR 接口时序



DDR 接口输出时序如图 21-5 所示。

图21-5 DDR 接口输出时序



DDR 接口输入时序如图 21-6 所示。

图21-6 DDR 接口输入时序

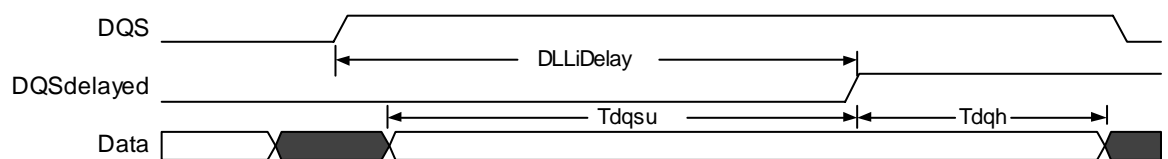


表21-3 DDR 接口时序参数列表

参数	符号	最小值	典型值	最大值	单位
DDRCK 下降沿到 DQS 使能或低电平	T_{lz}	0.2	0.2	TBD	Tck
DDRCK 上升沿到有效 ADDR、CONTRL 信号	T_{co_addr}	TBD	TBD	0.8	ns
DQS 上的写前导	T_{wpre}	TBD	0.5	TBD	Tck

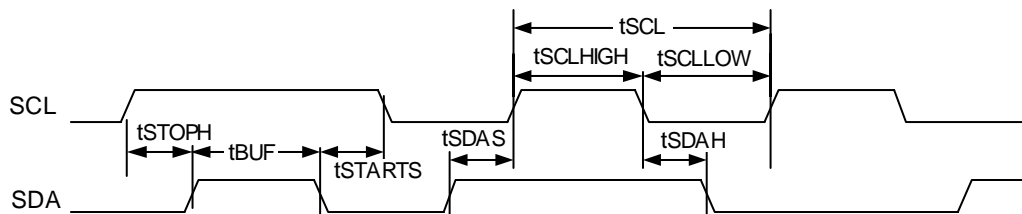
参数	符号	最小值	典型值	最大值	单位
DDRCK 到 DQS 输出	Tco_dqs	TBD	0.25	TBD	Tck
DQS 上的写后同步	Twpost	TBD	0.5	TBD	Tck
DQS 前的 DATA 建立	Tdvdqs	1	TBD	TBD	ns
DQS 后的 DATA 保持	Tdhdqs	1	TBD	TBD	ns
进入 DLL 的 DQS 延时	DLLiDelay	1.75	1.832	1.95	ns
延时的 DQS 前的 DATA 建立	Tdqsu	1.25	1.332	1.45	ns
延时的 DQS 前的 DATA 保持	Tdqh	1.25	1.332	1.45	ns

21.4 串行口时序参数

21.4.1 I²C 时序参数

I²C 传输时序如图 21-7 所示。

图21-7 I²C 传输时序



I²C 接口时序参数如表 21-4 所示。

表21-4 I²C 接口时序参数

参数	符号	最小值	典型值	最大值	单位
SCL 时钟宽度	t _{SCL}	TBD	TBD	TBD	ns
SCL 高电平脉冲宽度	t _{SCLHIGH}	TBD	TBD	TBD	ns
SCL 低电平脉冲宽度	t _{SCLLOW}	TBD	TBD	TBD	ns
STOP 建立时间	t _{STOPH}	TBD	TBD	TBD	ns
STOP 和 START 之间空余时间	t _{BUF}	TBD	TBD	TBD	ns
START 保持时间	t _{STARTS}	TBD	TBD	TBD	ns

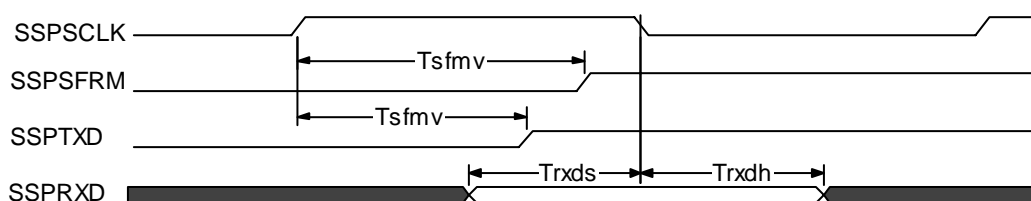


参数	符号	最小值	典型值	最大值	单位
SDA 建立时间	t_{SDAS}	TBD	TBD	TBD	ns
SDA 建立时间	t_{SDAH}	TBD	TBD	TBD	ns

21.4.2 SSP 时序参数

SSP 主模式时序如图 21-8 所示。

图21-8 SSP 主模式时序



SSP 主模式时序参数如表 21-5 所示。

表21-5 SSP 主模式时序参数

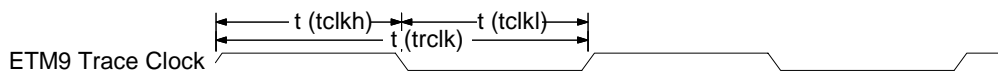
参数	符号	最小值	典型值	最大值	单位
SSPCLK 上升延到 SSPSRM 驱动有效	T_{sfmv}	TBD	TBD	TBD	TBD
SSPCLK 上升延到 SSPTXD 有效	T_{sfmv}	TBD	TBD	TBD	TBD
SSPRXD 建立时间	T_{rxds}	TBD	TBD	TBD	TBD
SSPRXD 保持时间	T_{rxdh}	TBD	TBD	TBD	TBD

详细内容可参见“第二部分 正文 16 SSP”。

21.5 ETM 时序参数

ETM 时钟时序如图 21-9 所示。

图21-9 ETM 时钟时序图





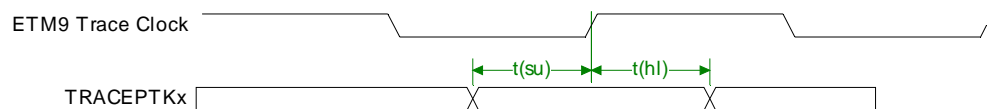
ETM 时钟时序参数如表 21-6 所示。

表21-6 ETM 时钟时序参数

参数	符号	最小值	典型值	最大值	单位
TRACECLK 时钟周期	t (trclk)	9.0	-	-	ns
TRACECLK 时钟频率	1/t (trclk)	-	-	110	MHz
TRACECLK 上升时间	t (er)	-	-	-	ns
TRACECLK 下降时间	t (ef)	-	-	-	ns
TRACECLK 高电平时间	t (tclkh)	-	-	-	ns
TRACECLK 低电平时间	t (trclk1)	-	-	-	ppm
TRACECLK 占空比	-	-	50	-	%

ETM 接口时序如图 21-10 所示。

图21-10 ETM 接口时序图



ETM 接口时序参数如表 21-7 所示。

表21-7 ETM 接口时序参数

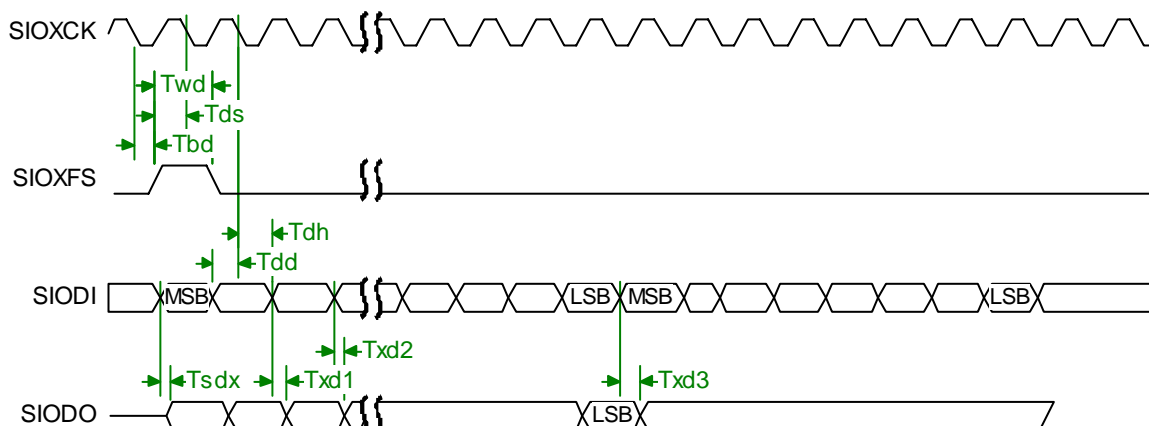
参数	符号	最小值	典型值	最大值	单位
TRACE 信号建立时间	t (su)	3	-	-	ns
TRACE 信号保持时间	t (hd)	2	-	-	ns

21.6 SIO 接口时序参数

SIO 接口时序如图 21-11 所示。



图21-11 SIO 接口时序



SIO 接口时序参数如表 21-8 所示。

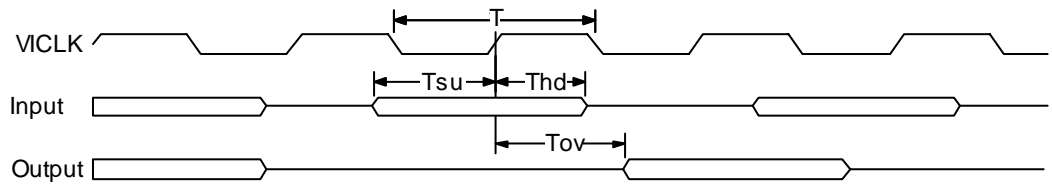
表21-8 SIO 接口时序参数

参数	符号	最小值	典型值	最大值	单位
SIOXFS 建立时间	t_{BS}	TBD	TBD	TBD	ns
SIOXFS 信号宽度	t_{WS}	TBD	TBD	TBD	ns
SIODI 建立时间	t_{DS}	TBD	TBD	TBD	ns
SIODI 保持时间	t_{DH}	TBD	TBD	TBD	ns
SIODO 延时时间	t_{SDX}	TBD	TBD	TBD	ns
SIODO 延时时间	t_{XD1}	TBD	TBD	TBD	ns
SIODO 保持时间	t_{XD2}	TBD	TBD	TBD	ns
SIODO 保持时间	t_{XD3}	TBD	TBD	TBD	ns

21.7 VI 视频输入接口时序参数

VI 视频输入接口时序如图 21-12 所示。

图21-12 VI 视频输入接口时序



VI 视频输入接口时序参数如表 21-9 所示。

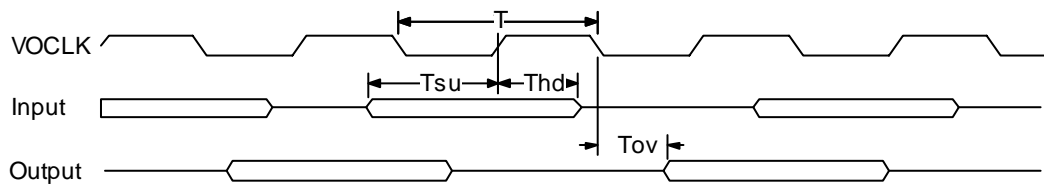
表21-9 VI 视频输入接口时序参数

参数	符号	最小值	典型值	最大值	单位
VICKL 时钟周期	t_{sdram}	-	37.03	-	ns
输入信号建立时间	t_{os}	5	-	-	ns
输入信号保持时间	t_{oh}	1	-	-	ns
输出信号延时	t_{ov}	-	-	-	ns

21.8 VO 视频输出接口时序参数

VO 视频输出接口时序如图 21-13 所示。

图21-13 VO 视频输出接口时序



VO 视频输出接口时序参数如表 21-10 所示。

表21-10 VO 视频输出接口时序参数

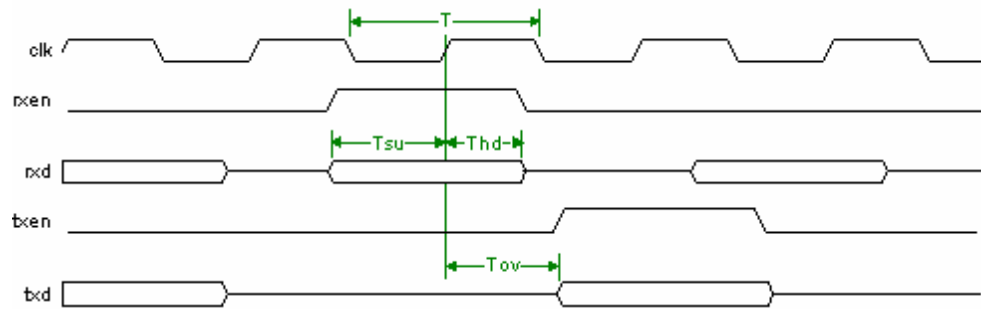
参数	符号	最小值	典型值	最大值	单位
VOCLK 时钟周期	t_{sdram}	18.51	37.03	-	ns
输入信号建立时间	t_{os}	4	-	-	ns
输入信号保持时间	t_{oh}	1	-	-	ns
输出信号延时	t_{ov}	0.5	-	8	ns



21.9 SF 以太网交换 RMII 接口时序（50MHz）参数

50MHz 时 RMII 接口时序如图 21-14 所示。

图21-14 50MHz 时 RMII 接口时序



50MHz 时 RMII 接口的时序参数如表 21-11 所示。

表21-11 50MHz 时 RMII 接口的时序参数

参数	符号	信号	最小值	最大值	单位
MII 信号建立时间 (relative to the clk_rmii)	T _{su} (RX)	RXER、 RXEN、 RXD[1:0]	4	-	ns
MII 信号保持时间 (relative to the clk_rmii)	T _{hd} (RX)	RXER、 RXEN、 RXD[1:0]	1	-	ns
MII 输出信号延时 (relative to the clk_rmii)	T _{ov} (MIITX)	TXD[1:0]、 TXEN	6	14	ns



22 电性能参数

关于本章

本章描述内容如下表所示。

标题	内容
22.1 DC 参数	介绍 DC 参数。
22.2 极限参数	介绍极限参数。
22.3 推荐工作条件	介绍推荐工作条件。



22.1 DC 参数

表22-1 DC 参数表 (VDDIO33=3.3V)

符号	参数	最小值	典型值	最大值	单位	说明
V _{IH}	高电平输入电压	1.7	-	5.5	V	-
V _{IL}	低电平输入电压	-0.3	-	0.7	V	-
I _L	输入漏电流	-	-	±1	μA	Input Buffer
I _{OZ}	三态输出漏电流	-	-	±1	μA	-
V _{OH}	高电平输出电压	2.4	-	-	V	I _{OL} =4、8、12、16mA
V _{OL}	低电平输出电压	-	-	0.4	V	I _{OL} =4、8、12、16mA
R _{PU}	上拉电阻	57	73	112	kΩ	-
R _{PD}	下拉电阻	57	82	160	kΩ	-

表22-2 DC 参数表 (VDDIO25=2.5V)

符号	参数	最小值	典型值	最大值	单位	说明
V _{IH(dc)}	高电平输入电压	V _{ref} +0.15	-	VDDIO25+0.3	V	-
V _{IL(dc)}	低电平输入电压	-0.3	-	V _{ref} -0.15	V	-
V _{IH(ac)}	高电平输入电压	V _{ref} +0.31	-	-	V	
V _{IL(ac)}	低电平输入电压	-	-	V _{ref} -0.31	V	
V _{OH(ac)}	高电平输出电压	V _{TTmax} +0.6075	-	-	V	
V _{OL(ac)}	低电平输出电压	-	-	V _{TTmin} -0.6075	V	
I _{OH(dc)}	高电平输出电流	-8.1	-	-	mA	
I _{OL(dc)}	低电平输出电流	8.1	-	-	mA	
V _{TT}	端接电压	V _{ref} -0.04	V _{ref}	V _{ref} +0.04	V	
V _{ref}	参考电压	1.13	1.25	1.38	V	



22.2 极限参数

表22-3 极限参数表

符号	参数	推荐使用范围	单位
T _{STG}	存储温度	-65~+150	°C
T _J	结温	0~125	°C
VDD	供电电压	请参考表 22-4	V

22.3 推荐工作条件

表22-4 推荐工作条件

符号	参数	最小值	典型值	最大值	单位
T _{OPT}	操作环境温度	-25	-	+85	°C
VDDCORE	内部 Core 电压	1.19	1.25	1.31	V
VDDIO25	I/O 电压	2.3	2.5	2.7	V
VDDIO33	I/O 电压	2.97	3.3	3.63	V
AVDD33USB	USB 工作电压	3.0	3.3	3.6	V
AVDD33PLL	PLL 模拟电压	2.97	3.3	3.63	V
VDD1PLL	PLL 数字电压	1.19	1.25	1.31	V
AVDDIOPLL	PLL 模拟电压	2.97	3.3	3.63	V



23 管脚描述

关于本章

本章描述内容如下表所示。

标题	内容
23.1 接口说明	说明管脚信号描述时用到的接口符号。
23.2 接口信号	描述 Hi3510 芯片的外部管脚的接口信号。
23.3 复用信号	描述 Hi3510 管脚复用信号。
23.4 电源、地和 NC 管脚	描述 Hi3510 芯片的电源、地和 NC 管脚。



23.1 接口说明

本节说明了管脚信号描述时用到的接口符号，如表 23-1 所示。

表23-1 接口符号说明

符号	说明
I/O	双向输入输出
I	输入
O	输出
OD	Open-Drain 输出
Hz	频率单位
mA	驱动电流单位
[N]	管脚序号，比如 nFUNCSEL[2]，表示第 3 个 nFUNCSEL 管脚
-	值为空

23.2 接口信号

本节描述了 Hi3510 芯片的外部管脚的接口信号，如表 23-2 所示。

表23-2 Hi3510 接口信号描述表

管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
时钟						
XIN1	I	Crystal	32.768k	-	-	32.768kHz 晶振时钟输入。不使用时，建议外部 1k 电阻下拉
XOUT1	O	Crystal	32.768k	-	-	32.768kHz 晶振时钟输出。
XIN2	I	Crystal	27M	-	-	27MHz 晶振时钟输入。
XOUT2	O	Crystal	27M	-	-	27MHz 晶振时钟输出。
XIN3	I	Crystal	48M	-	-	48MHz 晶振时钟输入。不使用时，建议外部 1k 电阻下拉。
XOUT3	O	Crystal	48M	-	-	48MHz 晶振时钟输出。



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
复位						
RSTN	I	LVTTL	<1M	-	PU	系统上电复位信号输入。全芯片复位，低电平有效。
RTCRTN	I	LVTTL	<1M	-	PU	RTC 上电复位输入，低电平有效。 建议在 PCB 板上与 RSTN 连接同一复位源。
WDGRST	OD	LVC MOS	<1M	8	Hi-Z	看门狗复位输出，低电平有效，OD 输出。
Mode & Debug						
TESTMODE	I	LVTTL	<1M	-	PD	测试或调试模式选择。
FUNCTIONSEL2	I	LVTTL	<1M	-	PD	功能选择管脚 2。
FUNCTIONSEL1	I	LVTTL	<1M	-	PD	功能选择管脚 1。
FUNCTIONSEL0	I	LVTTL	<1M	-	PD	功能选择管脚 0。
SCANEN	I	LVTTL	<1M	-	PD	Hi3510 scan 测试使能，正常工作时必须将该管脚连接为低电平。
BISTCLK	I	LVTTL	100M	-	PU	Hi3510 BIST 测试时钟输入，正常工作时必须将该管脚连接为低电平。
JTAG						
TCK	I	LVTTL	≤10M	-	PU	JTAG 时钟输入。必须外部下拉，建议 1k 电阻。
TDI	I	LVTTL	≤10M	-	PU	JTAG 数据输入。建议外部 4.7k 电阻上拉。
TMS	I	LVTTL	≤10M	-	PU	JTAG 模式选择输入。建议外部 4.7k 电阻上拉。
TRSTN	I	LVTTL	<1M	-	PU	JTAG 复位输入。如果进行 ARM926EJ-S 调试，建议外部 4.7k 电阻上拉；正常工作模式必须外部下拉，建议 10k 电阻。
TDO	O	LVC MOS	≤10M	8	Hi-Z	JTAG 数据输出。
RTCK	O	LVC MOS	≤10M	8	-	进行 ARM926EJ-S 调试、连接 Multi-ICE 或 RealView-ICE 时的 JTAG 反馈时钟。建议外部 1k 电阻下拉，。



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
ETM9						
TRACE SEL	I	LVTTL	<1M	-	PU	因部分 GPIO 口与 TRACE 输出信号管脚复用，在使用 ETM9 进行实时跟踪调试时，该管脚要设置为 1。 0: GPIO 接口； 1: ETM 接口。
GPIO3[4]	I/O	LVTTL	≤200M	16	PU	当 GPIO7[5]=0，作为 GPIO3[4]； 当 GPIO7[5]=1，且 pTRACESEL=0，作为 nVOCLK； 当 GPIO7[5]=1，且 pTRACESEL=1，作为 Trace 时钟输出 (traceclk)。 不使用时，可悬空处理。
GPIO3[3]	I/O	LVTTL	≤200M	12	PU Input	当 pTRACESEL=0，作为 GPIO3[3]； 当 pTRACESEL=1，作为 Trace 同步输出 (tracesync)。 不使用时，必须外部 1k 电阻下拉。
GPIO3[2:0]	I/O	LVTTL	≤200M	12	PU Input	当 pTRACESEL=0，作为 GPIO3[2:0]； 当 pTRACESEL=1，作为 Trace 流水状态输出 (pipestat[2:0])。 不使用时，必须外部 1k 电阻下拉。
GPIO2[7:0]	I/O	LVC MOS	≤200M	12	PU Input	GPIO2[7:0]与 Trace 数据输出 tracepkt[7:0]复用。 GPIO2[0]还可以复用作第 1 个音频接口同步信号 SIORFS0，GPIO2[1]可以复用为 DSP 时钟输出 ZSPCLK，用于观测 DSP 时钟的输出。 不使用时，可悬空处理。 说明 GPIO2[1]、GPIO2[0]只能同时做为输入或者同时做为输出使用。
MEMC						
BOOTS EL1	I	LVTTL	-	-	PD	静态 RAM 接口片选 1 上电复位时对应的 memory 数据位宽设置。
BOOTS EL0	I	LVTTL	-	-	PU	静态 RAM 接口片选 0 上电复位时对应的 memory 数据位宽设置。



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
SDRCK FB	I	LVTTL	100M	-	PU	SDRAM 反馈时钟输入。 不使用时，建议外部 1k 电阻下拉。
SDRCK 1	O	LVCMS	100M	12	-	SDRAM 接口时钟输出 1。
SDRCK 2	O	LVCMS	100M	12	-	SDRAM 接口时钟输出 2。
SDRRA SN	O	LVCMS	100M	8	-	SDRAM 接口行选通信号。
SDRCA SN	O	LVCMS	100M	8	-	SDRAM 接口列选通信号。
SDRCS N	O	LVCMS	100M	8	-	SDRAM 片选信号。
SDRDM [3:0]	O	LVCMS	100M	8	-	SDRAM 接口字节使能。
EBIWE N	O	LVCMS	100M	12	-	SDRAM 和静态 RAM 写使能。
EBIDQ3 1	I/O	LVTTL / LVCMS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0，作为 EBIDQ31； 当 GPIO7[0]=1，作为 GPIO5[4]。 不使用时，可悬空处理。
EBIDQ3 0	I/O	LVTTL / LVCMS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0，作为 EBIDQ30； 当 GPIO7[0]=1，作为 GPIO5[3]。 不使用时，可悬空处理。
EBIDQ2 9	I/O	LVTTL / LVCMS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0，作为 EBIDQ29； 当 GPIO7[0]=1，作为 GPIO5[2]。 不使用时，可悬空处理。
EBIDQ2 8	I/O	LVTTL / LVCMS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0，作为 EBIDQ28； 当 GPIO7[0]=1，作为 GPIO5[1]。 不使用时，可悬空处理。



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
EBIDQ2 7	I/O	LVTTL / LVCM OS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0, 作为 EBIDQ27; 当 GPIO7[0]=1, 作为 GPIO5[0]。 不使用时, 可悬空处理。
EBIDQ2 6	I/O	LVTTL / LVCM OS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0, 作为 EBIDQ26; 当 GPIO7[0]=1, 作为 GPIO4[7]。 不使用时, 可悬空处理。
EBIDQ2 5	I/O	LVTTL / LVCM OS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0, 作为 EBIDQ25; 当 GPIO7[0]=1, 作为 GPIO4[6]。 不使用时, 可悬空处理。
EBIDQ2 4	I/O	LVTTL / LVCM OS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0, 作为 EBIDQ24; 当 GPIO7[0]=1, 作为 GPIO4[5]。 不使用时, 可悬空处理。
EBIDQ2 3	I/O	LVTTL / LVCM OS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0, 作为 EBIDQ23; 当 GPIO7[0]=1, 作为 GPIO4[4]。 不使用时, 可悬空处理。
EBIDQ2 2	I/O	LVTTL / LVCM OS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0, 作为 EBIDQ22; 当 GPIO7[0]=1, 作为 GPIO4[3]。 不使用时, 可悬空处理。
EBIDQ2 1	I/O	LVTTL / LVCM OS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0, 作为 EBIDQ21; 当 GPIO7[0]=1, 作为 GPIO4[2]。 不使用时, 可悬空处理。
EBIDQ2 0	I/O	LVTTL / LVCM OS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0, 作为 EBIDQ20; 当 GPIO7[0]=1, 作为 GPIO4[1]。 不使用时, 可悬空处理。



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
EBIDQ19	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0, 作为 EBIDQ19; 当 GPIO7[0]=1, 作为 GPIO4[0]。 不使用时, 可悬空处理。
EBIDQ18	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0, 作为 EBIDQ18; 当 GPIO7[0]=1, 作为 GPIO3[7]。 不使用时, 可悬空处理。
EBIDQ17	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0, 作为 EBIDQ17; 当 GPIO7[0]=1, 作为 GPIO3[6]。 不使用时, 可悬空处理。
EBIDQ16	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。 当 GPIO7[0]=0, 作为 EBIDQ16; 当 GPIO7[0]=1, 作为 GPIO3[5]。 不使用时, 可悬空处理。
EBIDQ15	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。
EBIDQ14	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。
EBIDQ13	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。
EBIDQ12	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。
EBIDQ11	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
EBIDQ10	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。
EBIDQ9	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。
EBIDQ8	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。
EBIDQ7	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。
EBIDQ6	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。
EBIDQ5	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。
EBIDQ4	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。
EBIDQ3	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。
EBIDQ2	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。
EBIDQ1	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。
EBIDQ0	I/O	LVTTL / LVCMOS	100M	8	PU Input	SDRAM 和静态 RAM 数据总线。



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
SDRCKE	O	LVCMSOS	100M	8	-	SDRAM 时钟使能输出。
EBIADR [25:15]	O	LVCMSOS	100M	8	-	静态 RAM 地址总线。
EBIADR 14	O	LVCMSOS	100M	12	-	静态 RAM 地址总线、SDRAM bank 1 选择信号。
EBIADR 13	O	LVCMSOS	100M	12	-	静态 RAM 地址总线、SDRAM bank 0 选择信号。
EBIADR [12:0]	O	LVCMSOS	100M	12	-	静态 RAM 地址总线、SDRAM 行、列地址信号。
EBICS3N	O	LVCMSOS	100M	8	-	静态 RAM 片选 3。
EBICS2N	O	LVCMSOS	100M	8	-	静态 RAM 片选 2。
EBICS1N	O	LVCMSOS	100M	8	-	静态 RAM 片选 1，一般用于连接 boot FLASH 器件。
EBICS0N	O	LVCMSOS	100M	8	-	静态 RAM 片选 0。
EBIBLS 3	O	LVCMSOS	100M	8	-	静态 RAM 接口数据字节选择 3。
EBIBLS 2	O	LVCMSOS	100M	8	-	静态 RAM 接口数据字节选择 2。
EBIBLS 1	O	LVCMSOS	100M	8	-	静态 RAM 接口数据字节选择 1。
EBIBLS 0	O	LVCMSOS	100M	8	-	静态 RAM 接口数据字节选择 0。
EBIOEN	O	LVCMSOS	100M	8	-	静态 RAM 接口数据输出使能。
DREQ	I	LVCMSOS	100M	-	PD	外部 DMA 请求。 不使用时，建议外部 4.7k 电阻上拉。
DACK	O	LVTTL	100M	8	-	外部 DMA 响应。
DDRC						
DDRCKP	O	SSTL2	100M	class I	-	DDR SDRAM 接口正向时钟输出。
DDRCKN	O	SSTL2	100M	class I	-	DDR SDRAM 接口反向时钟输出。



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
DDRRA SN	O	SSTL2	100M	class I	-	DDR SDRAM 接口行选通信号。
DDRCASN	O	SSTL2	100M	class I	-	DDR SDRAM 接口列选通信号。
DDRC SN	O	SSTL2	100M	class I	-	DDR SDRAM 接口片选信号。
DDRDM1	O	SSTL2	100M	class I	-	DDR SDRAM 接口高字节 Mask 信号。
DDRDM0	O	SSTL2	100M	class I	-	DDR SDRAM 接口低字节 Mask 信号。
DDRWE N	O	SSTL2	100M	class I	-	DDR SDRAM 接口写使能信号。
DDR DQ [15:0]	I/O	SSTL2	100M	class I	-, Input	DDR SDRAM 接口数据总线。 不使用时, 可悬空处理。
DDRCK E	O	SSTL2	100M	class I	-	DDR SDRAM 接口时钟使能信号。
DDRBA1	O	SSTL2	100M	class I	-	DDR SDRAM 接口 bank 1 选择信号。
DDRBA0	O	SSTL2	100M	class I	-	DDR SDRAM 接口 bank 0 选择信号。
DDRAD R[12:0]	O	SSTL2	100M	class I	-	DDR SDRAM 接口行、列地址信号。
DDR DQ S1	I/O	SSTL2	100M	class I	-, Input	DDR SDRAM 接口数据 Strobe 信号 1。 不使用时, 可悬空处理。
DDR DQ S0	I/O	SSTL2	100M	class I	-, Input	DDR SDRAM 接口片选 Strobe 信号 0。 不使用时, 可悬空处理。
DDRRC VENI	I	SSTL2	100M	class I	-	DDRC 接收数据使能输入。 不使用时, 可悬空处理。
DDRRC VENO	O	SSTL2	100M	class I	-	DDRC 接收数据使能输出。
USB						
USB DP0	I/O	模拟信号	12M	-	-, Input	USB 端口 0, 差分正相数据。 不使用时, 建议 15k 电阻下拉。
USB DN0	I/O	模拟信号	12M	-	-, Input	USB 端口 0, 差分负相数据。 不使用时, 建议 15k 电阻下拉。



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
USBDP1	I/O	模拟信号	12M	-	-, Input	USB 端口 1, 差分正相数据。 不使用时, 建议 15k 电阻下拉。
USBDN1	I/O	模拟信号	12M	-	-, Input	USB 端口 1, 差分负相数据。 不使用时, 建议 15k 电阻下拉。
I²C						
SDA	I/O	LVTTL /LVCMOS	400k	8	PU, Input	I ² C 总线数据/地址, 需要外部 4.7k 电阻上拉。
SCL	I/O	LVTTL /LVCMOS	400k	8	PU, Input	I ² C 总线时钟, 需要外部 4.7k 电阻上拉。
SSP						
SSPSCLK	O	LVTTL /LVCMOS	<10M	4	-	SSP 总线时钟。
SSPRXD	I	LVTTL	<10M	-	PU	SSP 总线数据接收。 不使用时, 可悬空处理。
SSPTXD	O	LVTTL /LVCMOS	<10M	4	-	SSP 总线数据发送。
SSPSFRM	O	LVTTL /LVCMOS	<10M	4	-	SSP 帧或从设备选择输出信号。
UART						
URXD0	I	LVTTL	<1M	-	PU	UART0 数据接收。 不使用时, 可悬空处理。
UTXD0	O	LVCMOS	<1M	4	-	UART0 数据发送。
URXD1	I	LVTTL	<1M	-	PU	UART1 数据接收。 不使用时, 可悬空处理。
UTXD1	O	LVCMOS	<1M	4	-	UART1 数据发送。
UCTS1	I	LVTTL	<1M	-	PU	UART1 清除发送。 不使用时, 可悬空处理。



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
URTS1	O	LVCMS OS	<1M	4	-	UART1 请求发送。
GPIO						
INTRN	I	LVTTL	<10M	-	PU	全局中断。 不使用时, 建议 4.7k 电阻上拉。
GPIO0[7:0]	I/O	LVTTL /LVCMS OS	<10M	8	PU Input ,	通用目的输入输出接口。 不使用时, 可悬空处理。
GPIO1[7:0]	I/O	LVTTL /LVCMS OS	<10M	8	PU Input ,	通用目的输入输出接口。 不使用时, 可悬空处理。
VIU						
VIHS	I/O	LVTTL /LVCMS OS	27M	8	PU Input ,	水平同步或数据有效。 当工作在主模式时, 作为输出; 工作在从模式时, 作为输入。 当 hsync=0 时, 工作在数据有效模式下; 当 hsync=1 时, 工作在水平同步脉冲模式。 不使用时, 可悬空处理。
VIVS	I/O	LVTTL /LVCMS OS	27M	8	PU Input ,	垂直同步或场信号。 当工作在主模式时, 作为输出; 工作在从模式时, 作为输入。 当 vsync=0 时, 工作在场号模式下; 当 vsync=1 时, 工作在垂直脉冲模式。 不使用时, 可悬空处理。
VICK	I	LVTTL	27M	-	PU	像素位流时钟。 不使用时, 必须外部 1k 电阻下拉。
VIDAT[7:0]	I	LVTTL	27M	-	PU	视频数据输入或原始数据输入。 不使用时, 可悬空处理。
VOU						



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
VOHS	I/O	LVTTL / LVCMOS	27M	8	PU Input	行同步脉冲，可以配置成输入输出。 <ul style="list-style-type: none"> 在主模式下，配置成为输出。表示同步行脉冲信号，脉冲宽度和脉冲起始位置可配置。 在从模式下，配置成为输入，适合 BT.601 标准。 不使用时，可悬空处理。
VOVS	I/O	LVTTL / LVCMOS	27M	8	PU Input	帧同步脉冲，可以配置成输入输出。 <ul style="list-style-type: none"> 在主模式下，配置成为输出。表示同步场信号脉冲，脉冲宽度和位置可配置。 在从模式下，配置成为输入，适合 BT.601 标准。 不使用时，可悬空处理。
VOCK	I/O	LVTTL / LVCMOS	27M	12	PU Input	VOU 时钟输入输出。 <ul style="list-style-type: none"> 当作为输入时，Hi3510 的 VOU 模块 27MHz 时钟由外部提供； 当不接外部钟振时钟输入时，可配置为输出时钟，为 VDAC 器件提供时钟。 不使用时，建议外部 1k 电阻下拉。
LCDCB	I/O	LVTTL / LVCMOS	27M	8	PU Input	数据有效信号或者场号信号。 <ul style="list-style-type: none"> 当工作在主模式时，为输出； 当工作在从模式时，为输入（在从模式下，只能够表示场号信号）。 在 LCDCB=0 时，表示场号信号； 在 LCDCB=1 时，表示数据有效信号。 不使用时，可悬空处理。
VODAT [7:0]	O	LVCMOS	27M	8	-	输出像素值，TV 或 LCD_B[7:0]输出。
LCDP[14:8]	O	LVCMOS	27M	8	-	输出像素值，LCD_G[6:0]输出。
LCDP15	I/O	LVCMOS	27M	8	PU Input	输出像素值，LCD_G7 输出。 当 GPIO7[2]=0，作为 LCDP15； 当 GPIO7[2]=1，作为 SIORFS1。 不使用时，可悬空处理。



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
LCDP16	I/O	LVCMS OS	27M	8	PU Input	输出像素值, LCD_R0 输出。 当 GPIO7[2]=0, 作为 LCDP16; 当 GPIO7[2]=1, 作为 SIODI1。 不使用时, 可悬空处理。
LCDP17	I/O	LVCMS OS	27M	8	PU Input	输出像素值, LCD_R1 输出。 当 GPIO7[2]=0, 作为 LCDP17; 当 GPIO7[2]=1, 作为 SIOXCK1。 不使用时, 可悬空处理。
LCDP18	O	LVCMS OS	27M	8	-	输出像素值, LCD_R2 输出。 当 GPIO7[2]=0, 作为 LCDP18; 当 GPIO7[2]=1, 作为 SIODO1。 不使用时, 可悬空处理。
LCDP19	I/O	LVCMS OS	27M	8	PU Input	输出像素值, LCD_R3 输出。 当 GPIO7[2]=0, 作为 LCDP19; 当 GPIO7[2]=1, 作为 SIOXFS1。 不使用时, 可悬空处理。
LCDP20	O	LVCMS OS	27M	8	-	输出像素值, LCD_R4 输出。 当 GPIO7[1]=0, 作为 LCDP20; 当 GPIO7[1]=1, 作为 GPIO5[5]。 不使用时, 可悬空处理。
LCDP21	O	LVCMS OS	27M	8	-	输出像素值, LCD_R5 输出。 当 GPIO7[1]=0, 作为 LCDP21; 当 GPIO7[1]=1, 作为 GPIO5[6]。 不使用时, 可悬空处理。
LCDP22	O	LVCMS OS	27M	8	-	输出像素值, LCD_R6 输出。 当 GPIO7[1]=0, 作为 LCDP22; 当 GPIO7[1]=1, 作为 GPIO5[7]。 不使用时, 可悬空处理。
LCDP23	O	LVCMS OS	27M	8	-	输出像素值, LCD_R7 输出。 当 GPIO7[1]=0, 作为 LCDP23; 当 GPIO7[1]=1, 作为 GPIOOUT6[0]。 不使用时, 可悬空处理。
SIO						



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
SIODI0	I	LVTTL	<1M	-	PU	I ² S 或 PCM 接口数据输入。 不使用时，可悬空处理。
SIODO0	O	LVCMSOS	<1M	4	-	I ² S 或 PCM 接口数据输出。
SIOXFS0	I/O	LVTTL/LVCMSOS	<1M	4	PU Input	I ² S 左右通道标示或 PCM 帧同步信号。 不使用时，可悬空处理。
SIOXCK0	I/O	LVTTL/LVCMSOS	<1M	4	PU Input	I ² S 或 PCM 接口位流时钟。 不使用时，可悬空处理。
SIORFS0	I/O	LVCMSOS	<1M	12	PU Input	复用关系请参考上文 ETM9 管脚 GPIO2[0] 的描述。 不使用时，可悬空处理。
SF						
RMIICK	I	LVTTL	50M	-	PU	RMII 接口时钟输入。 不使用时，建议外部 1k 电阻下拉。
ATXD0	O	LVCMSOS	50M (RMII) 25M (MII)	4	-	(端口 0, 下行端口) MII0 发送数据 0 或 RMII0 发送数据 0。
ATXD1	O	LVCMSOS	50M (RMII) 25M (MII)	4	-	(端口 0, 下行端口) MII0 发送数据 1 或 RMII0 发送数据 1。
ATXD2	O	LVCMSOS	25M	4	-	(端口 0, 下行端口) MII0 发送数据 2; RMII0 模式下，不连接。
ATXD3	O	LVCMSOS	25M	4	-	(端口 0, 下行端口) MII0 发送数据 3; RMII0 模式下，不连接。
ATXEN	O	LVCMSOS	50M (RMII) 25M (MII)	4	-	(端口 0, 下行端口) MII0/RMII0 发送数据使能。
ATXERR	O	LVCMSOS	25M	4	-	(端口 0, 下行端口) MII0 发送数据错误; RMII0 模式下，不连接。



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
ATXCK	I	LVTTL	25M	-	PU	(端口 0, 下行端口) MII0 发送数据时钟; RMIIO 模式下, 不连接。 不使用时, 可悬空处理。
ARXD0	I	LVTTL	50M (RMII) 25M (MII)	-	PU	(端口 0, 下行端口) MII0 接收数据 0; 或 RMIIO 接收数据 0。 不使用时, 可悬空处理。
ARXD1	I	LVTTL	50M (RMII) 25M (MII)	-	PU	(端口 0, 下行端口) MII0 接收数据 1; 或 RMIIO 接收数据 1。 不使用时, 可悬空处理。
ARXD2	I	LVTTL	25M	-	PU	(端口 0, 下行端口) MII0 接收数据 2; RMIIO 模式下, 连接 GND。 不使用时, 可悬空处理。
ARXD3	I	LVTTL	25M	-	PU	(端口 0, 下行端口) MII0 接收数据 3; RMIIO 模式下, 连接 GND。 不使用时, 可悬空处理。
ARXDV	I	LVTTL	50M (RMII) 25M (MII)	-	PU	(端口 0, 下行端口) MII0/RMIIO 接收数据有效。 不使用时, 可悬空处理。
ARXERR	I	LVTTL	50M (RMII) 25M (MII)	-	PU	(端口 0, 下行端口) MII0/RMIIO 接收数据错误。 不使用时, 可悬空处理。
ARXCK	I	LVTTL	25M	-	PU	(端口 0, 下行端口) MII0 接收数据时钟; RMIIO 模式下, 连接 GND。 不使用时, 可悬空处理。
ACRS	I	LVTTL	25M	-	PU	(端口 0, 下行端口) MII0 载波侦听; RMIIO 模式下, 连接 GND。 不使用时, 可悬空处理。
ACOL	I	LVTTL	25M	-	PU	(端口 0, 下行端口) MII0 碰撞指示; RMIIO 模式下, 连接 GND。 不使用时, 可悬空处理。



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
BTXD0	O	LVCMSOS	50M (RMII) 25M (MII)	4	-	(端口 1, 上行端口) MII1 发送数据 0 或 RMII1 发送数据 0。
BTXD1	O	LVCMSOS	50M (RMII) 25M (MII)	4	-	(端口 1, 上行端口) MII1 发送数据 1 或 RMII1 发送数据 1。
BTXD2	O	LVCMSOS	25M	4	-	(端口 1, 上行端口) MII1 发送数据 2; RMII1 模式下, 不连接。
BTXD3	O	LVCMSOS	25M	4	-	(端口 1, 上行端口) MII1 发送数据 3; RMII1 模式下, 不连接。
BTXEN	O	LVCMSOS	50M (RMII) 25M (MII)	4	-	(端口 1, 上行端口) MII1/RMII1 发送数据使能。
BTXERR	O	LVCMSOS	25M	4	-	(端口 1, 上行端口) MII1 发送数据错误; RMII1 模式下, 不连接。
BTXCK	I	LVTTL	25M	-	PU	(端口 1, 上行端口) MII1 发送数据时钟; RMII1 模式下, 不连接。 不使用时, 可悬空处理。
BRXD0	I	LVTTL	50M (RMII) 25M (MII)	-	PU	(端口 1, 上行端口) MII1 接收数据 0 或 RMII1 接收数据 0。 不使用时, 可悬空处理。
BRXD1	I	LVTTL	50M (RMII) 25M (MII)	-	PU	(端口 1, 上行端口) MII1 接收数据 1 或 RMII1 接收数据 1。 不使用时, 可悬空处理。
BRXD2	I	LVTTL	25M	-	PU	(端口 1, 上行端口) MII1 接收数据 2; RMII1 模式下, 连接 GND。 不使用时, 可悬空处理。



管脚名	方向	类型	频率 (Hz)	驱动 (mA)	复位状态	功能描述
BRXD3	I	LVTTL	25M	-	PU	(端口 1, 上行端口) MII1 接收数据 3; RMI1 模式下, 连接 GND。 不使用时, 可悬空处理。
BRXDV	I	LVTTL	50M (RMI1) 25M (MII)	-	PU	(端口 1, 上行端口) MII1/RMI1 接收数据有效。 不使用时, 可悬空处理。
BRXERR	I	LVTTL	50M (RMI1) 25M (MII)	-	PU	(端口 1, 上行端口) MII1/RMI1 接收数据错误。 不使用时, 可悬空处理。
BRXCK	I	LVTTL	25M	-	PU	(端口 1, 上行端口) MII1 接收数据时钟; RMI1 模式下, 连接 GND。 不使用时, 可悬空处理。
BCRS	I	LVTTL	25M	-	PU	(端口 1, 上行端口) MII1 载波侦听; RMI1 模式下, 连接 GND。 不使用时, 可悬空处理。
BCOL	I	LVTTL	25M	-	PU	(端口 1, 上行端口) MII1 碰撞指示; RMI1 模式下, 连接 GND。 不使用时, 可悬空处理。
MDCK	O	LVTTL	2M	4	-	MII 管理接口时钟输出。
MDIO	I/O	LVTTL / LVCMOS	2M	4	PU Input	MII 管理接口数据输入输出, 需要外部 4.7k 电阻上拉。

23.3 复用信号

Hi3510 管脚复用信号由寄存器 GPIO7[5:0]控制, 复用说明如表 23-3 所示。



表23-3 管脚复用说明

位置	默认信号	复用信号 1	复用信号 2	复用说明
A10	GPIO3[4]	traceclk	nVOCLK	当 GPIO7[5]=0, 作为 GPIO3[4]; 当 GPIO7[5]=1, 且 pTRACESEL=0, 作为 nVOCLK; 当 GPIO7[5]=1, 且 pTRACESEL=1, 作为 Trace 时钟输出 (traceclk)。
C10	GPIO3[3]	Tracsync	-	当 pTRACESEL=0, 作为 GPIO3[3]; 当 pTRACESEL=1, 作为 Trace 同步输出 (tracsync)。
B10	GPIO3[2]	pipestat[2]	-	当 pTRACESEL=0, 作为 GPIO3[2]; 当 pTRACESEL=1, 作为 Trace 流水状态输出 (pipestat[2])。
D11	GPIO3[1]	pipestat[1]	-	当 pTRACESEL=0, 作为 GPIO3[1]; 当 pTRACESEL=1, 作为 Trace 流水状态输出 (pipestat[1])。
A9	GPIO3[0]	pipestat[0]	-	当 pTRACESEL=0, 作为 GPIO3[0]; 当 pTRACESEL=1, 作为 Trace 流水状态输出 (pipestat[0])。
B9	GPIO2[7]	tracepkt[7]	-	当 pTRACESEL=0, 作为 GPIO2[7]; 当 pTRACESEL=1, 作为 Trace 数据输出 (tracepkt[7])。
C9	GPIO2[6]	tracepkt[6]	-	当 pTRACESEL=0, 作为 GPIO2[6]; 当 pTRACESEL=1, 作为 Trace 数据输出 (tracepkt[6])。
D10	GPIO2[5]	tracepkt[5]	-	当 pTRACESEL=0, 作为 GPIO2[5]; 当 pTRACESEL=1, 作为 Trace 数据输出 (tracepkt[5])。



位置	默认信号	复用信号 1	复用信号 2	复用说明
D9	GPIO2[4]	tracepkt[4]	-	当 pTRACESEL=0, 作为 GPIO2[4]; 当 pTRACESEL=1, 作为 Trace 数据输出 (tracepkt[4])。
A8	GPIO2[3]	tracepkt[3]	-	当 pTRACESEL=0, 作为 GPIO2[3]; 当 pTRACESEL=1, 作为 Trace 数据输出 (tracepkt[3])。
B8	GPIO2[2]	tracepkt[2]	-	当 pTRACESEL=0, 作为 GPIO2[2]; 当 pTRACESEL=1, 作为 Trace 数据输出 (tracepkt[2])。
C8	GPIO2[1]	tracepkt[1]	ZSPCLK	当 GPIO7[4]=0, 作为 GPIO2[1]; 当 GPIO7[4]=1, 且 pTRACESEL=0, 作为 DSP 时钟输出 (ZSPCLK); 当 GPIO7[4]=1, 且 pTRACESEL=1, 作为 Trace 数据输出 (tracepkt[1])。 说明 GPIO2[1]、GPIO2[0]只能同时做为输入或者同时做为输出使用。
D8	GPIO2[0]	Tracepkt[0]	SIORFS0	当 GPIO7[3]=0, 作为 GPIO2[0]; 当 GPIO7[3]=1, 且 pTRACESEL=0, 作为 SIORFS0; 当 GPIO7[3]=1, 且 pTRACESEL=1, 作为 tracepkt[0]。 说明 GPIO2[1]、GPIO2[0]只能同时做为输入或者同时做为输出使用。
W14	EBIDQ31	GPIO5[4]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ31; 当 GPIO7[0] = 1, 作为 GPIO5[4]。
W13	EBIDQ30	GPIO5[3]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ30; 当 GPIO7[0] = 1, 作为 GPIO5[3]。



位置	默认信号	复用信号 1	复用信号 2	复用说明
AA13	EBIDQ29	GPIO5[2]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ29; 当 GPIO7[0] = 1, 作为 GPIO5[2]。
AB14	EBIDQ28	GPIO5[1]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ28; 当 GPIO7[0] = 1, 作为 GPIO5[1]。
W12	EBIDQ27	GPIO5[0]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ27; 当 GPIO7[0] = 1, 作为 GPIO5[0]。
Y12	EBIDQ26	GPIO4[7]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ26; 当 GPIO7[0] = 1, 作为 GPIO4[7]。
AA12	EBIDQ25	GPIO4[6]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ25; 当 GPIO7[0] = 1, 作为 GPIO4[6]。
AB13	EBIDQ24	GPIO4[5]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ24; 当 GPIO7[0] = 1, 作为 GPIO4[5]。
Y11	EBIDQ23	GPIO4[4]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ23; 当 GPIO7[0] = 1, 作为 GPIO4[4]。
AA11	EBIDQ22	GPIO4[3]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ22; 当 GPIO7[0] = 1, 作为 GPIO4[3]。
AB11	EBIDQ21	GPIO4[2]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ21; 当 GPIO7[0] = 1, 作为 GPIO4[2]。
AB10	EBIDQ20	GPIO4[1]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ20; 当 GPIO7[0] = 1, 作为 GPIO4[1]。
W10	EBIDQ19	GPIO4[0]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ19; 当 GPIO7[0] = 1, 作为 GPIO4[0]。



位置	默认信号	复用信号 1	复用信号 2	复用说明
Y10	EBIDQ18	GPIO3[7]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ18; 当 GPIO7[0] = 1, 作为 GPIO3[7]。
AA10	EBIDQ17	GPIO3[6]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ17; 当 GPIO7[0] = 1, 作为 GPIO3[6]。
AB9	EBIDQ16	GPIO3[5]	-	SDRAM 和静态 RAM 数据信号。 当 GPIO7[0] = 0, 作为 EBIDQ16; 当 GPIO7[0] = 1, 作为 GPIO3[5]。
L2	LCDP15	SIORFS1	-	输出像素值, LCD_G7 输出。 当 GPIO7[2] = 0, 作为 LCDP15; 当 GPIO7[2] = 1, 作为 SIORFS1。 同时必须满足 SC_PERCTRL1[LcdpOen] = 1。
L1	LCDP16	SIODI1	-	输出像素值, LCD_R0 输出。 当 GPIO7[2] = 0, 作为 LCDP16; 当 GPIO7[2] = 1, 作为 SIODI1。 同时必须满足 SC_PERCTRL1[LcdpOen] = 1。
K4	LCDP17	SIOXCK1	-	输出像素值, LCD_R1 输出。 当 GPIO7[2] = 0, 作为 LCDP17; 当 GPIO7[2] = 1, 作为 SIOXCK1。 同时必须满足 SC_PERCTRL1[LcdpOen] = 1。
K3	LCDP18	SIODO1	-	输出像素值, LCD_R2 输出。 当 GPIO7[2] = 0, 作为 LCDP18; 当 GPIO7[2] = 1, 作为 SIODO1。 同时必须满足 SC_PERCTRL1[LcdpOen] = 1。
K2	LCDP19	SIOXFS1	-	输出像素值, LCD_R3 输出。 当 GPIO7[2] = 0, 作为 LCDP19; 当 GPIO7[2] = 1, 作为 SIOXFS1。 同时必须满足 SC_PERCTRL1[LcdpOen] = 1。



位置	默认信号	复用信号 1	复用信号 2	复用说明
K1	LCDP20	GPIO5[5]	-	输出像素值，LCD_R4 输出。 当 GPIO7[1] = 0，作为 LCDP20； 当 GPIO7[1] = 1，作为 GPOUT5[5]。同时必须满足 SC_PERCTRL1[LcdpOen] = 1。
K5	LCDP21	GPIO5[6]	-	输出像素值，LCD_R5 输出。 当 GPIO7[1] = 0，作为 LCDP21； 当 GPIO7[1] = 1，作为 GPOUT5[6]。同时必须满足 SC_PERCTRL1[LcdpOen] = 1。
J3	LCDP22	GPIO5[7]	-	输出像素值，LCD_R6 输出。 当 GPIO7[1] = 0，作为 LCDP22； 当 GPIO7[1] = 1，作为 GPOUT5[7]。同时必须满足 SC_PERCTRL1[LcdpOen] = 1。
J2	LCDP23	GPIO6[0]	-	输出像素值，LCD_R7 输出。 当 GPIO7[1] = 0，作为 LCDP23； 当 GPIO7[1] = 1，作为 GPOUT6[0]。同时必须满足 SC_PERCTRL1[LcdpOen] = 1。

23.4 电源、地和 NC 管脚

本节描述了 Hi3510 芯片的电源、地和 NC (Not Connected) 管脚。具体描述如表 23-4 所示。

表23-4 电源和地管脚描述

管脚名称	类型	位置	功能描述
VDDIO33	数字电源	E14、E10、E12、V8、E17、E6、 H5、L5、M5、P5、U5、V7、 V9、V12、V14、V16、V11	接口工作电源，连接 3.3V。
VDDCORE	数字电源	V15、V10、E8、J5、R5、E15、 K18、P18	内核工作电源，连接 1.25V。



管脚名称	类型	位置	功能描述
VDDIO25	数字电源	P19、T18、U18、T21、N21、 J21、G21、L21、F18、G18、 G19、L19、V20、H18、J18、 L18、M18、N18、R18	DDR 工作电源， 连接 2.5V。 不使用 DDR 时， 可悬空处理。
VREF2	数字电源	P20	DDR 参考电源， 连接 1.25V。 不使用 DDR 时， 可悬空处理。
VREF1	数字电源	H20	
AVDD33USB0	模拟电源	E13	USB 端口 0 电 源，连接模拟 3.3V。 不使用时，建议连 接数字 3.3V。
AVDD33USB1	模拟电源	D14	USB 端口 1 电 源，连接模拟 3.3V。 不使用时，建议连 接数字 3.3V。
AVSSUSB0	模拟地	D13	模拟地 (AGND)。 不使用时，建议连 接 GND。
AVSSUSB1	模拟地	E16	模拟地 (AGND)。 不使用时，建议连 接 GND。
AVDD33PLL	模拟电源	W20	PLL 电源，连接模 拟 3.3V。
AVSS33PLL	模拟地	Y19	模拟地 (AGND)。
VDD1PLL	数字电源	W19	PLL 电源，连接数 字 1.25V。
VSS1PLL	数字地	V17	数字地 (DGND)。
AVDDIOPLL	模拟电源	V19	PLL 电源，连接模 拟 3.3V。
AVSSIOPLL	模拟地	V18	模拟地 (AGND)。



管脚名称	类型	位置	功能描述
DVSS	数字地	F6、F14、F15、F16、F17、G6、G17、H6、H17、J6、J17、K6、K17、L6、L17、M6、M17、N6、N17、P6、P17、R6、R17、T6、T17、U6、U7、U8、U9、U10、U11、U12、U13、U14、U15、U16、U17、W3、W21、F7、F8、F9、F10、F11、F12、F13、K10、K11、K12、K13、L10、L11、L12、L13、M10、M11、M12、M13、N10、N11、N12、N13	数字地 (DGND)。
NC	悬空	E18、W22、E9、V13、E7、E5、T5、F5、E11	无需连接。



24 封装、管脚分布

关于本章

本章描述内容如下表所示。

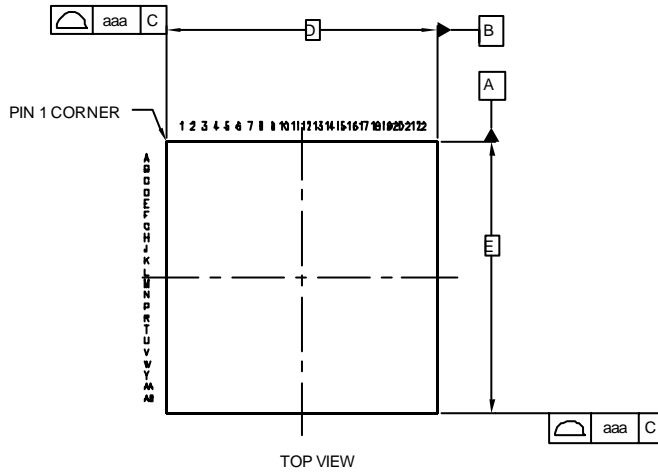
标题	内容
24.1 封装	介绍 Hi3510 芯片的封装。
24.2 管脚分布	介绍 Hi3510 芯片的管脚分布。



24.1 封装

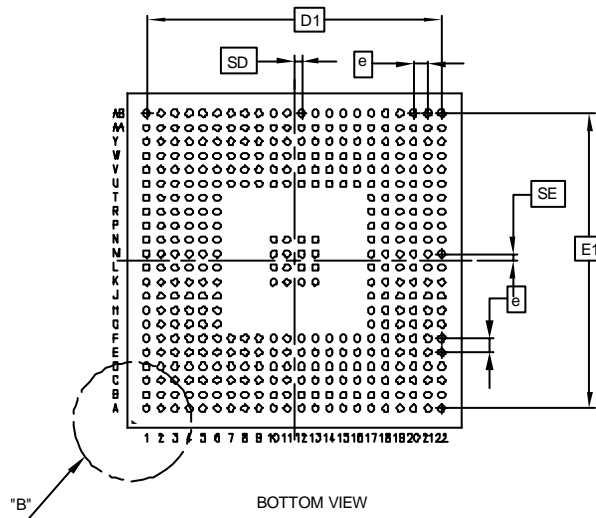
Hi3510 芯片采用 0.13 μm 工艺、LFBGA400 封装，大小为 19 \times 19mm，管脚间距为 0.8mm。具体封装尺寸如图 24-1 到图 24-5 所示，尺寸参数请参见表 24-1。

图24-1 芯片尺寸视图（俯视图）



注：此图中的尺寸单位是：MILLIMETER。

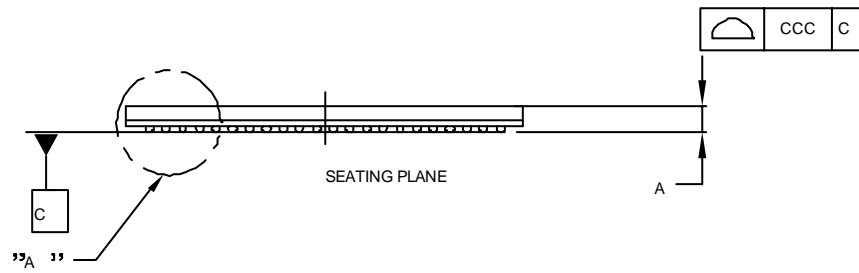
图24-2 芯片尺寸视图（仰视图）



注：此图中的尺寸单位是：MILLIMETER。

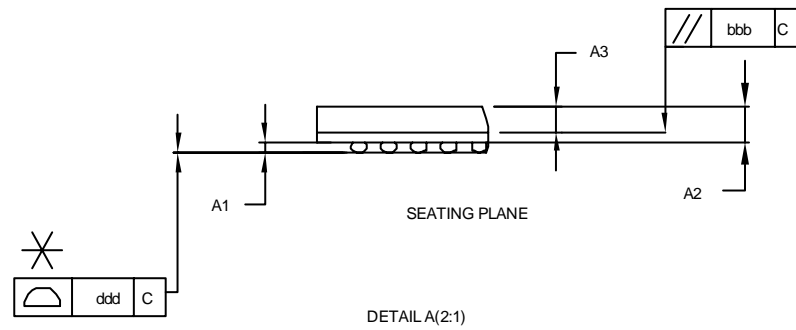


图24-3 芯片尺寸视图（侧视图）



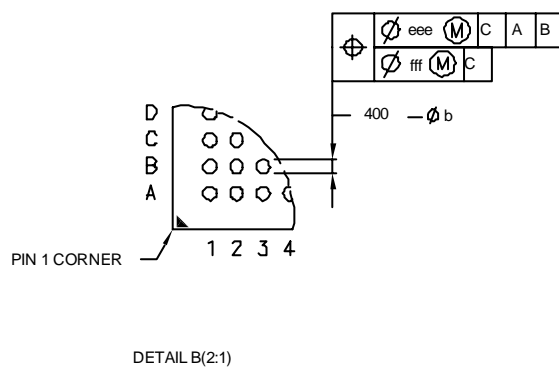
注：此图中的尺寸单位是：MILLIMETER。

图24-4 Detail A 的放大图



注：此图中的尺寸单位是：MILLIMETER。

图24-5 Detail B 的放大图



注：此图中的尺寸单位是：MILLIMETER。



表24-1 封装参数说明表

SYMBOL	MILLIMETER			INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	1.36	-	-	0.0535
A1	0.25	0.30	0.35	0.0098	0.0118	0.0138
A2	0.91	0.96	1.01	0.0358	0.0378	0.0398
A3	0.70 BASIC			0.0276 BASIC		
D	18.90	19.00	19.10	0.7441	0.7480	0.7520
D1	16.80 BASIC			0.6614 BASIC		
E	18.90	19.00	19.10	0.7441	0.7480	0.7520
E1	16.80 BASIC			0.6614 BASIC		
SD	0.40 BASIC			0.0157 BASIC		
SE	0.40 BASIC			0.0157 BASIC		
e	0.80 BASIC			0.0315 BASIC		
b	0.35	0.40	0.45	0.0138	0.0157	0.0177
aaa	0.15			0.0059		
bbb	0.20			0.0079		
ccc	0.20			0.0079		
ddd	0.12			0.0047		
eee	0.15			0.0059		
fff	0.08			0.0031		

24.2 管脚分布

Hi3510 V100 的管脚有 400 个，如表 24-2 所示。管脚按管脚名顺序排列如表 24-3 所示，按位置顺序如表 24-4 所示。

表24-2 Hi3510 V100 管脚数目统计表

管脚名	数量	管脚名	数量
VDDIO33	17	AVSSUSB0	1
VDDIO25	19	AVSSUSB1	1
VDDCORE	8	AVDD33PLL	1



管脚名	数量	管脚名	数量
VREF2	1	AVSS33PLL	1
VREF1	1	VDD1PLL	1
AVDD33USB0	1	VSS1PLL	1
AVDD33USB1	1	AVDDIOPLL	1
NC	9	AVSSIOPLL	1
DVSS	62	Total IO	273
总计	400	-	-



图24-6 Hi3510 管脚分布图（俯视图）

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
A	RTCSTN	GPIO1[2]	GPIO1[3]	GPIO1[6]	UTXD0	URTS1	TRACES EL	GPIO2[3]	GPIO3[0]	GPIO3[4]	ATXD2	ATXCK	ATXD3	ACRS	ATXEN	ARXD3	BTXD1	BTXCK	BTXERR	BRXD3	BRXDV	TCK	A																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
B	GPIO1[1]	XIN1	XOUT1	GPIO1[5]	URXD0	UCTS1	SSPRXD	GPIO2[2]	GPIO2[7]	GPIO3[2]	ATXD1	RMICLK	USBDP0	USBDN0	ARXERR	ARXD2	BTXD0	BTXD3	BTXEN	BRXD2	RTCK	TRSTN	B																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
C	GPIO1[0]	GPIO0[7]	GPIO1[7]	GPIO1[4]	MDIO	UTXD1	SSPTXD	GPIO2[1]	GPIO2[6]	GPIO3[3]	ATXD0	XIN3	USBDP1	USBDN1	ACOL	ARXD1	ARXDV	BTXD2	BRXCK	BRXD1	BCRS	TMS	C																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
D	GPIO0[6]	GPIO0[5]	GPIO0[4]	MDCK	SSPSCLK	URXD1	SSPSFRM	GPIO2[0]	GPIO2[4]	GPIO2[5]	GPIO3[1]	XOUT3	AVSSUSB0	AVDD33USB1	ARXD0	ATXERR	ARXCK	BCOL	BRXERR	BRXD0	TDO	TDI	D																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
E	SDA	SCL	BISTCLK	SCANEN	NC	VDDIO33	NC	VDDCORE	NC	VDDIO33	NC	VDDIO33	AVDD33USB0	VDDIO33	VDDCORE	AVSSUSB1	VDDIO33	NC	DDRCSEN	DDRRASN	DDRCASN	DDRWEN	E																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
F	FUNCSEL2	FUNCSEL1	FUNCSEL0	RSTN	NC	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	VDDIO25	DDRCKE	DDRCVENI	DDRCVENO	DDRQ15	F																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
G	GPIO0[3]	GPIO0[2]	GPIO0[1]	GPIO0[0]	TESTMODE	DVSS	<table border="1" style="margin: auto;"> <tr> <td>DVSS</td> <td>DVSS</td> <td>DVSS</td> <td>DVSS</td> </tr> <tr> <td>DVSS</td> <td>DVSS</td> <td>DVSS</td> <td>DVSS</td> </tr> <tr> <td>DVSS</td> <td>DVSS</td> <td>DVSS</td> <td>DVSS</td> </tr> <tr> <td>DVSS</td> <td>DVSS</td> <td>DVSS</td> <td>DVSS</td> </tr> </table>										DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	VDDIO25	VDDIO25	DDRQ13	VDDIO25	DDRQ14	DVSS	VDDIO25	DDRQ15	DVSS	VDDIO25	DDRQ16	DVSS	VDDIO25	DDRQ17	DVSS	VDDIO25	DDRQ18	DVSS	VDDIO25	DDRQ19	DVSS	VDDIO25	DDRQ20	DVSS	VDDIO25	DDRQ21	DVSS	VDDIO25	DDRQ22	DVSS	VDDIO25	DDRQ23	DVSS	VDDIO25	DDRQ24	DVSS	VDDIO25	DDRQ25	DVSS	VDDIO25	DDRQ26	DVSS	VDDIO25	DDRQ27	DVSS	VDDIO25	DDRQ28	DVSS	VDDIO25	DDRQ29	DVSS	VDDIO25	DDRQ30	DVSS	VDDIO25	DDRQ31	DVSS	VDDIO25	DDRQ32	DVSS	VDDIO25	DDRQ33	DVSS	VDDIO25	DDRQ34	DVSS	VDDIO25	DDRQ35	DVSS	VDDIO25	DDRQ36	DVSS	VDDIO25	DDRQ37	DVSS	VDDIO25	DDRQ38	DVSS	VDDIO25	DDRQ39	DVSS	VDDIO25	DDRQ40	DVSS	VDDIO25	DDRQ41	DVSS	VDDIO25	DDRQ42	DVSS	VDDIO25	DDRQ43	DVSS	VDDIO25	DDRQ44	DVSS	VDDIO25	DDRQ45	DVSS	VDDIO25	DDRQ46	DVSS	VDDIO25	DDRQ47	DVSS	VDDIO25	DDRQ48	DVSS	VDDIO25	DDRQ49	DVSS	VDDIO25	DDRQ50	DVSS	VDDIO25	DDRQ51	DVSS	VDDIO25	DDRQ52	DVSS	VDDIO25	DDRQ53	DVSS	VDDIO25	DDRQ54	DVSS	VDDIO25	DDRQ55	DVSS	VDDIO25	DDRQ56	DVSS	VDDIO25	DDRQ57	DVSS	VDDIO25	DDRQ58	DVSS	VDDIO25	DDRQ59	DVSS	VDDIO25	DDRQ60	DVSS	VDDIO25	DDRQ61	DVSS	VDDIO25	DDRQ62	DVSS	VDDIO25	DDRQ63	DVSS	VDDIO25	DDRQ64	DVSS	VDDIO25	DDRQ65	DVSS	VDDIO25	DDRQ66	DVSS	VDDIO25	DDRQ67	DVSS	VDDIO25	DDRQ68	DVSS	VDDIO25	DDRQ69	DVSS	VDDIO25	DDRQ70	DVSS	VDDIO25	DDRQ71	DVSS	VDDIO25	DDRQ72	DVSS	VDDIO25	DDRQ73	DVSS	VDDIO25	DDRQ74	DVSS	VDDIO25	DDRQ75	DVSS	VDDIO25	DDRQ76	DVSS	VDDIO25	DDRQ77	DVSS	VDDIO25	DDRQ78	DVSS	VDDIO25	DDRQ79	DVSS	VDDIO25	DDRQ80	DVSS	VDDIO25	DDRQ81	DVSS	VDDIO25	DDRQ82	DVSS	VDDIO25	DDRQ83	DVSS	VDDIO25	DDRQ84	DVSS	VDDIO25	DDRQ85	DVSS	VDDIO25	DDRQ86	DVSS	VDDIO25	DDRQ87	DVSS	VDDIO25	DDRQ88	DVSS	VDDIO25	DDRQ89	DVSS	VDDIO25	DDRQ90	DVSS	VDDIO25	DDRQ91	DVSS	VDDIO25	DDRQ92	DVSS	VDDIO25	DDRQ93	DVSS	VDDIO25	DDRQ94	DVSS	VDDIO25	DDRQ95	DVSS	VDDIO25	DDRQ96	DVSS	VDDIO25	DDRQ97	DVSS	VDDIO25	DDRQ98	DVSS	VDDIO25	DDRQ99	DVSS	VDDIO25	DDRQ100	DVSS	VDDIO25	DDRQ101	DVSS	VDDIO25	DDRQ102	DVSS	VDDIO25	DDRQ103	DVSS	VDDIO25	DDRQ104	DVSS	VDDIO25	DDRQ105	DVSS	VDDIO25	DDRQ106	DVSS	VDDIO25	DDRQ107	DVSS	VDDIO25	DDRQ108	DVSS	VDDIO25	DDRQ109	DVSS	VDDIO25	DDRQ110	DVSS	VDDIO25	DDRQ111	DVSS	VDDIO25	DDRQ112	DVSS	VDDIO25	DDRQ113	DVSS	VDDIO25	DDRQ114	DVSS	VDDIO25	DDRQ115	DVSS	VDDIO25	DDRQ116	DVSS	VDDIO25	DDRQ117	DVSS	VDDIO25	DDRQ118	DVSS	VDDIO25	DDRQ119	DVSS	VDDIO25	DDRQ120	DVSS	VDDIO25	DDRQ121	DVSS	VDDIO25	DDRQ122	DVSS	VDDIO25	DDRQ123	DVSS	VDDIO25	DDRQ124	DVSS	VDDIO25	DDRQ125	DVSS	VDDIO25	DDRQ126	DVSS	VDDIO25	DDRQ127	DVSS	VDDIO25	DDRQ128	DVSS	VDDIO25	DDRQ129	DVSS	VDDIO25	DDRQ130	DVSS	VDDIO25	DDRQ131	DVSS	VDDIO25	DDRQ132	DVSS	VDDIO25	DDRQ133	DVSS	VDDIO25	DDRQ134	DVSS	VDDIO25	DDRQ135	DVSS	VDDIO25	DDRQ136	DVSS	VDDIO25	DDRQ137	DVSS	VDDIO25	DDRQ138	DVSS	VDDIO25	DDRQ139	DVSS	VDDIO25	DDRQ140	DVSS	VDDIO25	DDRQ141	DVSS	VDDIO25	DDRQ142	DVSS	VDDIO25	DDRQ143	DVSS	VDDIO25	DDRQ144	DVSS	VDDIO25	DDRQ145	DVSS	VDDIO25	DDRQ146	DVSS	VDDIO25	DDRQ147	DVSS	VDDIO25	DDRQ148	DVSS	VDDIO25	DDRQ149	DVSS	VDDIO25	DDRQ150	DVSS	VDDIO25	DDRQ151	DVSS	VDDIO25	DDRQ152	DVSS	VDDIO25	DDRQ153	DVSS	VDDIO25	DDRQ154	DVSS	VDDIO25	DDRQ155	DVSS	VDDIO25	DDRQ156	DVSS	VDDIO25	DDRQ157	DVSS	VDDIO25	DDRQ158	DVSS	VDDIO25	DDRQ159	DVSS	VDDIO25	DDRQ160	DVSS	VDDIO25	DDRQ161	DVSS	VDDIO25	DDRQ162	DVSS	VDDIO25	DDRQ163	DVSS	VDDIO25	DDRQ164	DVSS	VDDIO25	DDRQ165	DVSS	VDDIO25	DDRQ166	DVSS	VDDIO25	DDRQ167	DVSS	VDDIO25	DDRQ168	DVSS	VDDIO25	DDRQ169	DVSS	VDDIO25	DDRQ170	DVSS	VDDIO25	DDRQ171	DVSS	VDDIO25	DDRQ172	DVSS	VDDIO25	DDRQ173	DVSS	VDDIO25	DDRQ174	DVSS	VDDIO25	DDRQ175	DVSS	VDDIO25	DDRQ176	DVSS	VDDIO25	DDRQ177	DVSS	VDDIO25	DDRQ178	DVSS	VDDIO25	DDRQ179	DVSS	VDDIO25	DDRQ180	DVSS	VDDIO25	DDRQ181	DVSS	VDDIO25	DDRQ182	DVSS	VDDIO25	DDRQ183	DVSS	VDDIO25	DDRQ184	DVSS	VDDIO25	DDRQ185	DVSS	VDDIO25	DDRQ186	DVSS	VDDIO25	DDRQ187	DVSS	VDDIO25	DDRQ188	DVSS	VDDIO25	DDRQ189	DVSS	VDDIO25	DDRQ190	DVSS	VDDIO25	DDRQ191	DVSS	VDDIO25	DDRQ192	DVSS	VDDIO25	DDRQ193	DVSS	VDDIO25	DDRQ194	DVSS	VDDIO25	DDRQ195	DVSS	VDDIO25	DDRQ196	DVSS	VDDIO25	DDRQ197	DVSS	VDDIO25	DDRQ198	DVSS	VDDIO25	DDRQ199	DVSS	VDDIO25	DDRQ200	DVSS	VDDIO25	DDRQ201	DVSS	VDDIO25	DDRQ202	DVSS	VDDIO25	DDRQ203	DVSS	VDDIO25	DDRQ204	DVSS	VDDIO25	DDRQ205	DVSS	VDDIO25	DDRQ206	DVSS	VDDIO25	DDRQ207	DVSS	VDDIO25	DDRQ208	DVSS	VDDIO25	DDRQ209	DVSS	VDDIO25	DDRQ210	DVSS	VDDIO25	DDRQ211	DVSS	VDDIO25	DDRQ212	DVSS	VDDIO25	DDRQ213	DVSS	VDDIO25	DDRQ214	DVSS	VDDIO25	DDRQ215	DVSS	VDDIO25	DDRQ216	DVSS	VDDIO25	DDRQ217	DVSS	VDDIO25	DDRQ218	DVSS	VDDIO25	DDRQ219	DVSS	VDDIO25	DDRQ220	DVSS	VDDIO25	DDRQ221	DVSS	VDDIO25	DDRQ222	DVSS	VDDIO25	DDRQ223	DVSS	VDDIO25	DDRQ224	DVSS	VDDIO25	DDRQ225	DVSS	VDDIO25	DDRQ226	DVSS	VDDIO25	DDRQ227	DVSS	VDDIO25	DDRQ228	DVSS	VDDIO25	DDRQ229	DVSS	VDDIO25	DDRQ230	DVSS	VDDIO25	DDRQ231	DVSS	VDDIO25	DDRQ232	DVSS	VDDIO25	DDRQ233	DVSS	VDDIO25	DDRQ234	DVSS	VDDIO25	DDRQ235	DVSS	VDDIO25	DDRQ236	DVSS	VDDIO25	DDRQ237	DVSS	VDDIO25	DDRQ238	DVSS	VDDIO25	DDRQ239	DVSS	VDDIO25	DDRQ240	DVSS	VDDIO25	DDRQ241	DVSS	VDDIO25	DDRQ242	DVSS	VDDIO25	DDRQ243	DVSS	VDDIO25	DDRQ244	DVSS	VDDIO25	DDRQ245	DVSS	VDDIO25	DDRQ246	DVSS	VDDIO25	DDRQ247	DVSS	VDDIO25	DDRQ248	DVSS	VDDIO25	DDRQ249	DVSS	VDDIO25	DDRQ250	DVSS	VDDIO25	DDRQ251	DVSS	VDDIO25	DDRQ252	DVSS	VDDIO25	DDRQ253	DVSS	VDDIO25	DDRQ254	DVSS	VDDIO25	DDRQ255	DVSS	VDDIO25	DDRQ256	DVSS	VDDIO25	DDRQ257	DVSS	VDDIO25	DDRQ258	DVSS	VDDIO25	DDRQ259	DVSS	VDDIO25	DDRQ260	DVSS	VDDIO25	DDRQ261	DVSS	VDDIO25	DDRQ262	DVSS	VDDIO25	DDRQ263	DVSS	VDDIO25	DDRQ264	DVSS	VDDIO25	DDRQ265	DVSS	VDDIO25	DDRQ266	DVSS	VDDIO25	DDRQ267	DVSS	VDDIO25	DDRQ268	DVSS	VDDIO25	DDRQ269	DVSS	VDDIO25	DDRQ270	DVSS	VDDIO25	DDRQ271	DVSS	VDDIO25	DDRQ272	DVSS	VDDIO25	DDRQ273	DVSS	VDDIO25	DDRQ274	DVSS	VDDIO25	DDRQ275	DVSS	VDDIO25	DDRQ276	DVSS	VDDIO25	DDRQ277	DVSS	VDDIO25	DDRQ278	DVSS	VDDIO25	DDRQ279	DVSS	VDDIO25	DDRQ280	DVSS	VDDIO25	DDRQ281	DVSS	VDDIO25	DDRQ282	DVSS	VDDIO25	DDRQ283	DVSS	VDDIO25	DDRQ284	DVSS	VDDIO25	DDRQ285	DVSS	VDDIO25	DDRQ286	DVSS	VDDIO25	DDRQ287	DVSS	VDDIO25	DDRQ288	DVSS	VDDIO25	DDRQ289	DVSS	VDDIO25	DDRQ290	DVSS	VDDIO25	DDRQ291	DVSS	VDDIO25	DDRQ292	DVSS	VDDIO25	DDRQ293	DVSS	VDDIO25	DDRQ294	DVSS	VDDIO25	DDRQ295	DVSS	VDDIO25	DDRQ296	DVSS	VDDIO25	DDRQ297	DVSS	VDDIO25	DDRQ298	DVSS	VDDIO25	DDRQ299	DVSS	VDDIO25	DDRQ300	DVSS	VDDIO25	DDRQ301	DVSS	VDDIO25	DDRQ302	DVSS	VDDIO25	DDRQ303	DVSS	VDDIO25	DDRQ304	DVSS	VDDIO25	DDRQ305	DVSS	VDDIO25	DDRQ306	DVSS	VDDIO25	DDRQ307	DVSS	VDDIO25	DDRQ308	DVSS	VDDIO25	DDRQ309	DVSS	VDDIO25	DDRQ310	DVSS	VDDIO25	DDRQ311	DVSS	VDDIO25	DDRQ312	DVSS	VDDIO25	DDRQ313	DVSS	VDDIO25	DDRQ314	DVSS	VDDIO25	DDRQ315	DVSS	VDDIO25	DDRQ316	DVSS	VDDIO25	DDRQ317	DVSS	VDDIO25	DDRQ318	DVSS	VDDIO25	DDRQ319	DVSS	VDDIO25	DDRQ320	DVSS	VDDIO25	DDRQ321	DVSS	VDDIO25	DDRQ322	DVSS	VDDIO25	DDRQ323	DVSS	VDDIO25	DDRQ324	DVSS	VDDIO25	DDRQ325	DVSS	VDDIO25	DDRQ326	DVSS	VDDIO25	DDRQ327	DVSS	VDDIO25	DDRQ328	DVSS	VDDIO25	DDRQ329	DVSS	VDDIO25	DDRQ330	DVSS	VDDIO25	DDRQ331	DVSS	VDDIO25	DDRQ332	DVSS	VDDIO25	DDRQ333	DVSS	VDDIO25	DDRQ334	DVSS	VDDIO25	DDRQ335	DVSS	VDDIO25	DDRQ336	DVSS	VDDIO25	DDRQ337	DVSS	VDDIO25	DDRQ338	DVSS	VDDIO25	DDRQ339	DVSS	VDDIO25	DDRQ340	DVSS	VDDIO25	DDRQ341	DVSS	VDDIO25	DDRQ342	DVSS	VDDIO25	DDRQ343	DVSS	VDDIO25	DDRQ344	DVSS	VDDIO25	DDRQ345	DVSS	VDDIO25	DDRQ346	DVSS	VDDIO25	DDRQ347	DVSS	VDDIO25	DDRQ348	DVSS	VDDIO25	DDRQ349	DVSS	VDDIO25	DDRQ350	DVSS	VDDIO25	DDRQ351	DVSS	VDDIO25	DDRQ352	DVSS	VDDIO25	DDRQ353	DVSS	VDDIO25	DDRQ354	DVSS	VDDIO25	DDRQ355	DVSS	VDDIO25	DDRQ356	DVSS	VDDIO25	DDRQ357	DVSS	VDDIO25	DDRQ358	DVSS	VDDIO25	DDRQ359	DVSS	VDDIO25	DDRQ360	DVSS	VDDIO25	DDRQ361	DVSS	VDDIO25	DDRQ362	DVSS	VDDIO25	DDRQ363	DVSS	VDDIO25	DDRQ364	DVSS	VDDIO25	DDRQ365	DVSS	VDDIO25	DDRQ366	DVSS	VDDIO25	DDRQ367	DVSS	VDDIO25	DDRQ368	DVSS	VDDIO25	DDRQ369	DVSS	VDDIO25	DDRQ370	DVSS	VDDIO25	DDRQ371	DVSS	VDDIO25	DDRQ372	DVSS	VDDIO25	DDRQ373	DVSS	VDDIO25
DVSS	DVSS	DVSS	DVSS																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
DVSS	DVSS	DVSS	DVSS																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
DVSS	DVSS	DVSS	DVSS																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
DVSS	DVSS	DVSS	DVSS																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										



表24-3 管脚排列表（按管脚名排序）

管脚名	位置	管脚名	位置
ACOL	C15	EBIDQ13	AB7
ACRS	A14	EBIDQ14	W9
ARXCK	D17	EBIDQ15	Y9
ARXD0	D15	EBIDQ16/GPIO3[5]	AB9
ARXD1	C16	EBIDQ17/GPIO3[6]	AA10
ARXD2	B16	EBIDQ18/GPIO3[7]	Y10
ARXD3	A16	EBIDQ19/GPIO4[0]	W10
ARXDV	C17	EBIDQ20/GPIO4[1]	AB10
ARXERR	B15	EBIDQ21/GPIO4[2]	AB11
ATXCK	A12	EBIDQ22/GPIO4[3]	AA11
ATXD0	C11	EBIDQ23/GPIO4[4]	Y11
ATXD1	B11	EBIDQ24/GPIO4[5]	AB13
ATXD2	A11	EBIDQ25/GPIO4[6]	AA12
ATXD3	A13	EBIDQ26/GPIO4[7]	Y12
ATXEN	A15	EBIDQ27/GPIO5[0]	W12
ATXERR	D16	EBIDQ28/GPIO5[1]	AB14
VDD1PLL	W19	EBIDQ29/GPIO5[2]	AA13
AVDD33PLL	W20	EBIDQ30/GPIO5[3]	W13
AVDD33USB0	E13	EBIDQ31/GPIO5[4]	W14
AVDD33USB1	D14	EBIOEN	AB2
AVDDIOPLL	V19	EBIWEN	AA4
VSS1PLL	V17	FUNCSEL0	F3
AVSS33PLL	Y19	FUNCSEL1	F2
AVSSIOPLL	V18	FUNCSEL2	F1
AVSSUSB0	D13	GPIO0[0]	G4
AVSSUSB1	E16	GPIO0[1]	G3
BCOL	D18	GPIO0[2]	G2
BCRS	C21	GPIO0[3]	G1
BISTCLK	E3	GPIO0[4]	D3



管脚名	位置	管脚名	位置
BOOTSEL0	V6	GPIO0[5]	D2
BOOTSEL1	V5	GPIO0[6]	D1
BRXCK	C19	GPIO0[7]	C2
BRXD0	D20	GPIO1[0]	C1
BRXD1	C20	GPIO1[1]	B1
BRXD2	B20	GPIO1[2]	A2
BRXD3	A20	GPIO1[3]	A3
BRXDV	A21	GPIO1[4]	C4
BRXERR	D19	GPIO1[5]	B4
BTXCK	A18	GPIO1[6]	A4
BTXD0	B17	GPIO1[7]	C3
BTXD1	A17	GPIO2[0]/tracepkt[0]/SIORFS0	D8
BTXD2	C18	GPIO2[1]/tracepkt[1]	C8
BTXD3	B18	GPIO2[2]/tracepkt[2]	B8
BTXEN	B19	GPIO2[3]/tracepkt[3]	A8
BTXERR	A19	GPIO2[4]/tracepkt[4]	D9
DACK	AB3	GPIO2[5]/tracepkt[5]	D10
DDRADR0	U19	GPIO2[6]/tracepkt[6]	C9
DDRADR1	U20	GPIO2[7]/tracepkt[7]	B9
DDRADR2	U21	GPIO3[0]/pipestat[0]	A9
DDRADR3	U22	GPIO3[1]/pipestat[1]	D11
DDRADR4	T19	GPIO3[2]/pipestat[2]	B10
DDRADR5	T20	GPIO3[3]/Tracsync	C10
DDRADR6	T22	GPIO3[4]/traceclk	A10
DDRADR7	R19	INTRN	H1
DDRADR8	R20	LCDCB	R2
DDRADR9	R21	LCDP8	N1
DDRADR10	R22	LCDP9	M2
DDRADR11	P21	LCDP10	N5
DDRADR12	P22	LCDP11	M3
DDRBA0	V21	LCDP12	M1



管脚名	位置	管脚名	位置
DDRBA1	V22	LCDP13	L4
DDRCASN	E21	LCDP14	L3
DDRCKE	F19	LCDP15/SIORFS1	L2
DDRCKN	M19	LCDP16/SIODI1	L1
DDRCKP	N19	LCDP17/SIOXCK1	K4
DDRCASN	E19	LCDP18/SIODO1	K3
DDRDM0	N20	LCDP19/SIOXFS1	K2
DDRDM1	K22	LCDP20/GPIO5[5]	K1
DDRQ0	N22	LCDP21/GPIO5[6]	K5
DDRQ1	M20	LCDP22/GPIO5[7]	J3
DDRQ2	M21	LCDP23/GPIO6[0]	J2
DDRQ3	M22	MDCK	D4
DDRQ4	L22	MDIO	C5
DDRQ5	K19	NC	E5
DDRQ6	K20	NC	E7
DDRQ7	K21	NC	E9
DDRQ8	J19	NC	E11
DDRQ9	J20	NC	E18
DDRQ10	J22	NC	F5
DDRQ11	H21	NC	T5
DDRQ12	H22	NC	V13
DDRQ13	G20	NC	W22
DDRQ14	G22	RMIICK	B12
DDRQ15	F22	RSTN	F4
DDRQS0	L20	RTCK	B21
DDRQS1	H19	RTCRSTN	A1
DDRRASN	E20	SCANEN	E4
DDRCVENI	F20	SCL	E2
DDRCVENO	F21	SDA	E1
DDRWEN	E22	SDRCASN	W1
DREQ	AB1	SDRCK	Y3



管脚名	位置	管脚名	位置
DVSS	F6	SDRCK2	V3
DVSS	F7	SDRCKE	AA2
DVSS	F8	SDRCKFB	Y2
DVSS	F9	SDRCSN	Y4
DVSS	F10	SDRDM0	Y5
DVSS	F11	SDRDM1	AB8
DVSS	F12	SDRDM2	AB12
DVSS	F13	SDRDM3	AB15
DVSS	F14	SDRRASN	W2
DVSS	F15	SIODI0	J1
DVSS	F16	SIODO0	H3
DVSS	F17	SIOXCK0	H4
DVSS	G6	SIOXFS0	H2
DVSS	G17	SSPRXD	B7
DVSS	H6	SSPSCLK	D5
DVSS	H17	SSPSFRM	D7
DVSS	J6	SSPTXD	C7
DVSS	J17	TCK	A22
DVSS	K6	TDI	D22
DVSS	K10	TDO	D21
DVSS	K11	TESTMODE	G5
DVSS	K12	TMS	C22
DVSS	K13	TRACESEL	A7
DVSS	K17	TRSTN	B22
DVSS	L6	UCTS1	B6
DVSS	L10	URTS1	A6
DVSS	L11	URXD0	B5
DVSS	L12	URXD1	D6
DVSS	L13	USBDN0	B14
DVSS	L17	USBDN1	C14
DVSS	M6	USBDP0	B13



管脚名	位置	管脚名	位置
DVSS	M10	USBDP1	C13
DVSS	M11	UTXD0	A5
DVSS	M12	UTXD1	C6
DVSS	M13	VDDCORE	E8
DVSS	M17	VDDCORE	E15
DVSS	N6	VDDCORE	J5
DVSS	N10	VDDCORE	K18
DVSS	N11	VDDCORE	P18
DVSS	N12	VDDCORE	R5
DVSS	N13	VDDCORE	V10
DVSS	N17	VDDCORE	V15
DVSS	P6	VDDIO25	F18
DVSS	P17	VDDIO25	G18
DVSS	R6	VDDIO25	G19
DVSS	R17	VDDIO25	G21
DVSS	T6	VDDIO25	H18
DVSS	T17	VDDIO25	J18
DVSS	U6	VDDIO25	J21
DVSS	U7	VDDIO25	L18
DVSS	U8	VDDIO25	L19
DVSS	U9	VDDIO25	L21
DVSS	U10	VDDIO25	M18
DVSS	U11	VDDIO25	N18
DVSS	U12	VDDIO25	N21
DVSS	U13	VDDIO25	P19
DVSS	U14	VDDIO25	R18
DVSS	U15	VDDIO25	T18
DVSS	U16	VDDIO25	T21
DVSS	U17	VDDIO25	U18
DVSS	W3	VDDIO25	V20
DVSS	W21	VDDIO33	E6



管脚名	位置	管脚名	位置
EBIADR0	Y14	VDDIO33	E10
EBIADR1	Y13	VDDIO33	E12
EBIADR2	AB16	VDDIO33	E14
EBIADR3	AA15	VDDIO33	E17
EBIADR4	Y15	VDDIO33	H5
EBIADR5	W15	VDDIO33	L5
EBIADR6	AB17	VDDIO33	M5
EBIADR7	AA16	VDDIO33	P5
EBIADR8	Y16	VDDIO33	U5
EBIADR9	W16	VDDIO33	V7
EBIADR10	AB18	VDDIO33	V8
EBIADR11	AA17	VDDIO33	V9
EBIADR12	Y17	VDDIO33	V11
EBIADR13	W17	VDDIO33	V12
EBIADR14	AB19	VDDIO33	V14
EBIADR15	AA18	VDDIO33	V16
EBIADR16	Y18	VICK	U4
EBIADR17	W18	VIDAT0	V1
EBIADR18	AB20	VIDAT1	U3
EBIADR19	AA19	VIDAT2	U2
EBIADR20	AB21	VIDAT3	U1
EBIADR21	AA20	VIDAT4	T4
EBIADR22	AB22	VIDAT5	T3
EBIADR23	AA21	VIDAT6	R3
EBIADR24	AA22	VIDAT7	T2
EBIADR25	Y22	VIHS	V4
EBIBLS0	W5	VIVS	V2
EBIBLS1	AA9	VOCK	R4
EBIBLS2	W11	VODAT0	R1
EBIBLS3	AA14	VODAT1	P4
EBICS0N	AA1	VODAT2	N2



管脚名	位置	管脚名	位置
EBICS1N	AA3	VODAT3	P2
EBICS2N	Y1	VODAT4	P1
EBICS3N	W4	VODAT5	N4
EBIDQ0	AA5	VODAT6	M4
EBIDQ1	AB4	VODAT7	N3
EBIDQ2	W6	VOHS	T1
EBIDQ3	Y6	VOVS	P3
EBIDQ4	AA6	VREF1	H20
EBIDQ5	AB5	VREF2	P20
EBIDQ6	W7	WDGRST	J4
EBIDQ7	Y7	XIN1	B2
EBIDQ8	AA7	XIN2	Y21
EBIDQ9	AB6	XIN3	C12
EBIDQ10	W8	XOUT1	B3
EBIDQ11	Y8	XOUT2	Y20
EBIDQ12	AA8	XOUT3	D12

表24-4 管脚排列表（按位置排序）

位置	管脚名	位置	管脚名
A1	RTCRSTN	M1	LCDP12
A2	GPIO1[2]	M2	LCDP9
A3	GPIO1[3]	M3	LCDP11
A4	GPIO1[6]	M4	VODAT6
A5	UTXD0	M5	VDDIO33
A6	URTS1	M6	DVSS
A7	TRACESEL	M10	DVSS
A8	GPIO2[3]/tracepkt[3]	M11	DVSS
A9	GPIO3[0]/pipestat[0]	M12	DVSS
A10	GPIO3[4]/traceclk	M13	DVSS
A11	ATXD2	M17	DVSS



位置	管脚名	位置	管脚名
A12	ATXCK	M18	VDDIO25
A13	ATXD3	M19	DDRCKN
A14	ACRS	M20	DDRDQ1
A15	ATXEN	M21	DDRDQ2
A16	ARXD3	M22	DDRDQ3
A17	BTXD1	N1	LCDP8
A18	BTXCK	N2	VODAT2
A19	BTXERR	N3	VODAT7
A20	BRXD3	N4	VODAT5
A21	BRXDV	N5	LCDP10
A22	TCK	N6	DVSS
B1	GPIO1[1]	N10	DVSS
B2	XIN1	N11	DVSS
B3	XOUT1	N12	DVSS
B4	GPIO1[5]	N13	DVSS
B5	URXD0	N17	DVSS
B6	UCTS1	N18	VDDIO25
B7	SSPRXD	N19	DDRCKP
B8	GPIO2[2]/tracepkt[2]	N20	DDRDM0
B9	GPIO2[7]/tracepkt[7]	N21	VDDIO25
B10	GPIO3[2]/pipestat[2]	N22	DDRDQ0
B11	ATXD1	P1	VODAT4
B12	RMICLK	P2	VODAT3
B13	USBDP0	P3	VOVS
B14	USBDN0	P4	VODAT1
B15	ARXERR	P5	VDDIO33
B16	ARXD2	P6	DVSS
B17	BTXD0	P17	DVSS
B18	BTXD3	P18	VDDCORE
B19	BTXEN	P19	VDDIO25
B20	BRXD2	P20	VREF2



位置	管脚名	位置	管脚名
B21	RTCK	P21	DDRADR11
B22	TRSTN	P22	DDRADR12
C1	GPIO1[0]	R1	VODAT0
C2	GPIO0[7]	R2	LCDCB
C3	GPIO1[7]	R3	VIDAT6
C4	GPIO1[4]	R4	VOCK
C5	MDIO	R5	VDDCORE
C6	UTXD1	R6	DVSS
C7	SSPTXD	R17	DVSS
C8	GPIO2[1]/tracepkt[1]	R18	VDDIO25
C9	GPIO2[6]/tracepkt[6]	R19	DDRADR7
C10	GPIO3[3]/Tracsync	R20	DDRADR8
C11	ATXD0	R21	DDRADR9
C12	XIN3	R22	DDRADR10
C13	USBDP1	T1	VOHS
C14	USBDN1	T2	VIDAT7
C15	ACOL	T3	VIDAT5
C16	ARXD1	T4	VIDAT4
C17	ARXDV	T5	NC
C18	BTXD2	T6	DVSS
C19	BRXCK	T17	DVSS
C20	BRXD1	T18	VDDIO25
C21	BCRS	T19	DDRADR4
C22	TMS	T20	DDRADR5
D1	GPIO0[6]	T21	VDDIO25
D2	GPIO0[5]	T22	DDRADR6
D3	GPIO0[4]	U1	VIDAT3
D4	MDCK	U2	VIDAT2
D5	SSPSCLK	U3	VIDAT1
D6	URXD1	U4	VICK
D7	SSPSFRM	U5	VDDIO33



位置	管脚名	位置	管脚名
D8	GPIO2[0]/tracepkt[0]/SIORFS0	U6	DVSS
D9	GPIO2[4]/tracepkt[4]	U7	DVSS
D10	GPIO2[5]/tracepkt[5]	U8	DVSS
D11	GPIO3[1]/pipestat[1]	U9	DVSS
D12	XOUT3	U10	DVSS
D13	AVSSUSB0	U11	DVSS
D14	AVDD33USB1	U12	DVSS
D15	ARXD0	U13	DVSS
D16	ATXERR	U14	DVSS
D17	ARXCK	U15	DVSS
D18	BCOL	U16	DVSS
D19	BRXERR	U17	DVSS
D20	BRXD0	U18	VDDIO25
D21	TDO	U19	DDRADR0
D22	TDI	U20	DDRADR1
E1	SDA	U21	DDRADR2
E2	SCL	U22	DDRADR3
E3	BISTCLK	V1	VIDAT0
E4	SCANEN	V2	VIVS
E5	NC	V3	SDRCK2
E6	VDDIO33	V4	VIHS
E7	NC	V5	BOOTSEL1
E8	VDDCORE	V6	BOOTSEL0
E9	NC	V7	VDDIO33
E10	VDDIO33	V8	VDDIO33
E11	NC	V9	VDDIO33
E12	VDDIO33	V10	VDDCORE
E13	AVDD33USB0	V11	VDDIO33
E14	VDDIO33	V12	VDDIO33
E15	VDDCORE	V13	NC
E16	AVSSUSB1	V14	VDDIO33



位置	管脚名	位置	管脚名
E17	VDDIO33	V15	VDDCORE
E18	NC	V16	VDDIO33
E19	DDRCASN	V17	VSS1PLL
E20	DDRRASN	V18	AVSSIOPLL
E21	DDRCASN	V19	AVDDIOPLL
E22	DDRWEN	V20	VDDIO25
F1	FUNCSEL2	V21	DDRBA0
F2	FUNCSEL1	V22	DDRBA1
F3	FUNCSEL0	W1	SDRCASN
F4	RSTN	W2	SDRRASN
F5	NC	W3	DVSS
F6	DVSS	W4	EBICS3N
F7	DVSS	W5	EBIBLS0
F8	DVSS	W6	EBIDQ2
F9	DVSS	W7	EBIDQ6
F10	DVSS	W8	EBIDQ10
F11	DVSS	W9	EBIDQ14
F12	DVSS	W10	EBIDQ19/GPIO4[0]
F13	DVSS	W11	EBIBLS2
F14	DVSS	W12	EBIDQ27/GPIO5[0]
F15	DVSS	W13	EBIDQ30/GPIO5[3]
F16	DVSS	W14	EBIDQ31/GPIO5[4]
F17	DVSS	W15	EBIADR5
F18	VDDIO25	W16	EBIADR9
F19	DDRCKE	W17	EBIADR13
F20	DDRCVENI	W18	EBIADR17
F21	DDRCVENO	W19	VDD1PLL
F22	DDRDQ15	W20	AVDD33PLL
G1	GPIO0[3]	W21	DVSS
G2	GPIO0[2]	W22	NC



位置	管脚名	位置	管脚名
G3	GPIO0[1]	Y1	EBICS2N
G4	GPIO0[0]	Y2	SDRCKFB
G5	TESTMODE	Y3	SDRCK1
G6	DVSS	Y4	SDRCSN
G17	DVSS	Y5	SDRDM0
G18	VDDIO25	Y6	EBIDQ3
G19	VDDIO25	Y7	EBIDQ7
G20	DDRDQ13	Y8	EBIDQ11
G21	VDDIO25	Y9	EBIDQ15
G22	DDRDQ14	Y10	EBIDQ18/GPIO3[7]
H1	INTRN	Y11	EBIDQ23/GPIO4[4]
H2	SIOXFS0	Y12	EBIDQ26/GPIO4[7]
H3	SIODO0	Y13	EBIADR1
H4	SIOXCK0	Y14	EBIADR0
H5	VDDIO33	Y15	EBIADR4
H6	DVSS	Y16	EBIADR8
H17	DVSS	Y17	EBIADR12
H18	VDDIO25	Y18	EBIADR16
H19	DDRDQS1	Y19	AVSS33PLL
H20	VREF1	Y20	XOUT2
H21	DDRDQ11	Y21	XIN2
H22	DDRDQ12	Y22	EBIADR25
J1	SIODI0	AA1	EBICS0N
J2	LCDP23/GPIO6[0]	AA2	SDRCKE
J3	LCDP22/GPIO5[7]	AA3	EBICS1N
J4	WDGRST	AA4	EBIWEN
J5	VDDCORE	AA5	EBIDQ0
J6	DVSS	AA6	EBIDQ4
J17	DVSS	AA7	EBIDQ8
J18	VDDIO25	AA8	EBIDQ12
J19	DDRDQ8	AA9	EBIBLS1



位置	管脚名	位置	管脚名
J20	DDRDQ9	AA10	EBIDQ17/GPIO3[6]
J21	VDDIO25	AA11	EBIDQ22/GPIO4[3]
J22	DDRDQ10	AA12	EBIDQ25/GPIO4[6]
K1	LCDP20/GPIO5[5]	AA13	EBIDQ29/GPIO5[2]
K2	LCDP19/SIOXFS1	AA14	EBIBLS3
K3	LCDP18/SIODO1	AA15	EBIADR3
K4	LCDP17/SIOXCK1	AA16	EBIADR7
K5	LCDP21/GPIO5[6]	AA17	EBIADR11
K6	DVSS	AA18	EBIADR15
K10	DVSS	AA19	EBIADR19
K11	DVSS	AA20	EBIADR21
K12	DVSS	AA21	EBIADR23
K13	DVSS	AA22	EBIADR24
K17	DVSS	AB1	DREQ
K18	VDDCORE	AB2	EBIOEN
K19	DDRDQ5	AB3	DACK
K20	DDRDQ6	AB4	EBIDQ1
K21	DDRDQ7	AB5	EBIDQ5
K22	DDRDM1	AB6	EBIDQ9
L1	LCDP16/SIODI1	AB7	EBIDQ13
L2	LCDP15/SIORFS1	AB8	SDRDM1
L3	LCDP14	AB9	EBIDQ16/GPIO3[5]
L4	LCDP13	AB10	EBIDQ20/GPIO4[1]
L5	VDDIO33	AB11	EBIDQ21/GPIO4[2]
L6	DVSS	AB12	SDRDM2
L10	DVSS	AB13	EBIDQ24/GPIO4[5]
L11	DVSS	AB14	EBIDQ28/GPIO5[1]
L12	DVSS	AB15	SDRDM3
L13	DVSS	AB16	EBIADR2



位置	管脚名	位置	管脚名
L17	DVSS	AB17	EBIADR6
L18	VDDIO25	AB18	EBIADR10
L19	VDDIO25	AB19	EBIADR14
L20	DDRDQS0	AB20	EBIADR18
L21	VDDIO25	AB21	EBIADR20
L22	DDRDQ4	AB22	EBIADR22



A 管脚功能速查表

表 A-1 简要列出了 Hi3510 的管脚功能，详细功能请参考正文中的相关描述。

表A-1 管脚功能速查表

位置	管脚名称	功能描述
A1	RTCRSTN	RTC上电复位输入，低有效。建议在PCB板上与RSTN连接同一复位源。
A2	GPIO1[2]	通用目的输入输出接口。
A3	GPIO1[3]	通用目的输入输出接口。
A4	GPIO1[6]	通用目的输入输出接口。
A5	UTXD0	UART0数据发送。
A6	URTS1	UART1请求发送。
A7	TRACESEL	因部分GPIO口与TRACE输出信号管脚复用，在使用ETM9进行实时跟踪调试时，该管脚要设置为1。为0时，为GPIO接口；为1时，为ETM接口。
A8	GPIO2[3]	Trace数据输出（对应tracepkt[3]）。
A9	GPIO3[0]	Trace流水状态输出（对应pipestat[0]）。
A10	GPIO3[4]	Trace时钟输出（traceclk）。
A11	ATXD2	（端口0）MII0发送数据2；RMII0模式下，不连接。
A12	ATXCK	（端口0）MII0发送数据时钟；RMII0模式下，不连接。
A13	ATXD3	（端口0）MII0发送数据3；RMII0模式下，不连接。
A14	ACRS	（端口0）MII0载波侦听；RMII0模式下，连接GND。
A15	ATXEN	（端口0）MII0/RMII0发送数据使能。
A16	ARXD3	（端口0）MII0接收数据3；RMII0模式下，连接GND。
A17	BTXD1	（端口1）MII1发送数据1或RMII1发送数据1。
A18	BTXCK	（端口1）MII1发送数据时钟；RMII1模式下，不连接。



位置	管脚名称	功能描述
A19	BTXERR	(端口1) MII1发送数据错误; RMI1模式下, 不连接。
A20	BRXD3	(端口1) MII1接收数据3; RMI1模式下, 连接GND。
A21	BRXDV	(端口1) MII1/RMI1接收数据有效。
A22	TCK	JTAG时钟输入。
B1	GPIO1[1]	通用目的输入输出接口。
B2	XIN1	32.768kHz晶振时钟输入。
B3	XOUT1	32.768kHz晶振时钟输出。
B4	GPIO1[5]	通用目的输入输出接口。
B5	URXD0	UART0数据接收。
B6	UCTS1	UART1清除发送。
B7	SSPRXD	SSP总线数据接收。
B8	GPIO2[2]	Trace数据输出(对应tracepkt[2])。
B9	GPIO2[7]	Trace数据输出(对应tracepkt[7])。
B10	GPIO3[2]	Trace流水状态输出(对应pipestat[2])。
B11	ATXD1	(端口0) MII0发送数据1或RMI0发送数据1。
B12	RMIICLK	RMI接口时钟输入。
B13	USBDP0	USB端口0, 差分正相数据。
B14	USBDN0	USB端口0, 差分负相数据。
B15	ARXERR	(端口0) MII0/RMI0接收数据错误。
B16	ARXD2	(端口0) MII0接收数据2; RMI0模式下, 连接GND。
B17	BTXD0	(端口1) MII1发送数据0或RMI1发送数据0。
B18	BTXD3	(端口1) MII1发送数据3; RMI1模式下, 不连接。
B19	BTXEN	(端口1) MII1/RMI1发送数据使能。
B20	BRXD2	(端口1) MII1接收数据2; RMI1模式下, 连接GND。
B21	RTCK	进行ARM926EJ-S调试、连接Multi-ICE或RealView-ICE时的JTAG反馈时钟。
B22	TRSTN	JTAG复位输入。
C1	GPIO1[0]	通用目的输入输出接口。
C2	GPIO0[7]	通用目的输入输出接口。
C3	GPIO1[7]	通用目的输入输出接口。
C4	GPIO1[4]	通用目的输入输出接口。



位置	管脚名称	功能描述
C5	MDIO	MII管理接收数据输入输出，需上拉处理。
C6	UTXD1	UART1数据发送。
C7	SSPTXD	SSP总线数据发送。
C8	GPIO2[1]	Trace数据输出（对应tracepkt[1]）。
C9	GPIO2[6]	Trace数据输出（对应tracepkt[6]）。
C10	GPIO3[3]	Trace同步输出（tracesync）。
C11	ATXD0	（端口0）MII0发送数据0或RMII0发送数据0。
C12	XIN3	48MHz晶振时钟输入。
C13	USBDP1	USB端口1，差分正相数据。
C14	USBDN1	USB端口1，差分负相数据。
C15	ACOL	（端口0）MII0碰撞指示；RMII0模式下，连接GND。
C16	ARXD1	（端口0）MII0接收数据1；或RMII0接收数据1。
C17	ARXDV	（端口0）MII0/RMII0接收数据有效。
C18	BTXD2	（端口1）MII1发送数据2；RMII1模式下，不连接。
C19	BRXCK	（端口1）MII1接收数据时钟；RMII1模式下，连接GND。
C20	BRXD1	（端口1）MII1接收数据1或RMII1接收数据1。
C21	BCRS	（端口1）MII1载波侦听；RMII1模式下，连接GND。
C22	TMS	JTAG模式选择输入。
D1	GPIO0[6]	通用目的输入输出接口。
D2	GPIO0[5]	通用目的输入输出接口。
D3	GPIO0[4]	通用目的输入输出接口。
D4	MDCK	MII管理接口时钟输出。
D5	SSPSCLK	SSP总线时钟。
D6	URXD1	UART1数据接收。
D7	SSPSFRM	SSP帧或从设备选择输出信号。
D8	GPIO2[0]	Tracesel=0时，为GPIO2[0]；Tracesel=1且GPIO7[3]=0时，为tracepkt[0]；Tracesel=0且GPIO7[3]=1时，为SIORFS0。
D8	SIORFS0	复用关系请参考上文ETM9管脚GPIO2[0]的描述。
D9	GPIO2[4]	Trace数据输出（对应tracepkt[4]）。
D10	GPIO2[5]	Trace数据输出（对应tracepkt[5]）。
D11	GPIO3[1]	Trace流水状态输出（对应pipestat[1]）。



位置	管脚名称	功能描述
D12	XOUT3	48MHz晶振时钟输出。
D13	AVSSUSB0	模拟地（AGND）。
D14	AVDD33USB1	USB端口1电源，连接模拟3.3V。
D15	ARXD0	（端口0）MII0接收数据0；或RMII0接收数据0。
D16	ATXERR	（端口0）MII0发送数据错误；RMII0模式下，不连接。
D17	ARXCK	（端口0）MII0接收数据时钟；RMII0模式下，连接GND。
D18	BCOL	（端口1）MII1碰撞指示；RMII1模式下，连接GND。
D19	BRXERR	（端口1）MII1/RMII1接收数据错误。
D20	BRXD0	（端口1）MII1接收数据0或RMII1接收数据0。
D21	TDO	JTAG数据输出。
D22	TDI	JTAG数据输入。
E1	SDA	I ² C总线数据/地址。
E2	SCL	I ² C总线时钟。
E3	BISTCLK	Hi3510 BIST测试时钟输入，正常工作时必须将该管脚连接为低电平。
E4	SCANEN	Hi3510 scan测试使能，正常工作时必须将该管脚连接为低电平。
E5	NC	悬空，无需连接。
E6	VDDIO33	接口工作电源，连接3.3V。
E7	NC	悬空，无需连接。
E8	VDDCORE	内核工作电源，连接1.25V。
E9	NC	悬空，无需连接。
E10	VDDIO33	接口工作电源，连接3.3V。
E11	NC	悬空，无需连接。
E12	VDDIO33	接口工作电源，连接3.3V。
E13	AVDD33USB0	USB端口0电源，连接模拟3.3V。
E14	VDDIO33	接口工作电源，连接3.3V。
E15	VDDCORE	内核工作电源，连接1.25V。
E16	AVSSUSB1	模拟地（AGND）。
E17	VDDIO33	接口工作电源，连接3.3V。



位置	管脚名称	功能描述
E18	NC	悬空，无需连接。
E19	DDRC SN	DDR SDRAM接口片选信号。
E20	DDRRASN	DDR SDRAM接口行选通信号。
E21	DDRCASN	DDR SDRAM接口列选通信号。
E22	DDRWEN	DDR SDRAM接口写使能信号。
F1	FUNCSEL2	功能选择管脚2。
F2	FUNCSEL1	功能选择管脚1。
F3	FUNCSEL0	功能选择管脚0。
F4	RSTN	系统上电复位信号输入。全芯片复位，低有效。
F5	NC	悬空，无需连接。
F6	DVSS	数字地（DGND）。
F7	DVSS	数字地（DGND）。
F8	DVSS	数字地（DGND）。
F9	DVSS	数字地（DGND）。
F10	DVSS	数字地（DGND）。
F11	DVSS	数字地（DGND）。
F12	DVSS	数字地（DGND）。
F13	DVSS	数字地（DGND）。
F14	DVSS	数字地（DGND）。
F15	DVSS	数字地（DGND）。
F16	DVSS	数字地（DGND）。
F17	DVSS	数字地（DGND）。
F18	VDDIO25	DDR工作电源，连接2.5V。
F19	DDRCKE	DDR SDRAM接口时钟使能信号。
F20	DDRCVENI	DDRC接收数据使能输入。
F21	DDRCVENO	DDRC接收数据使能输出。
F22	DDRDQ15	DDR SDRAM接口数据总线。
G1	GPIO0[3]	通用目的输入输出接口。
G2	GPIO0[2]	通用目的输入输出接口。
G3	GPIO0[1]	通用目的输入输出接口。
G4	GPIO0[0]	通用目的输入输出接口。



位置	管脚名称	功能描述
G5	TESTMODE	测试获调试模式选择。
G6	DVSS	数字地 (DGND)。
G17	DVSS	数字地 (DGND)。
G18	VDDIO25	DDR工作电源, 连接2.5V。
G19	VDDIO25	DDR工作电源, 连接2.5V。
G20	DDRDQ13	DDR SDRAM接口数据总线。
G21	VDDIO25	DDR工作电源, 连接2.5V。
G22	DDRDQ14	DDR SDRAM接口数据总线。
H1	INTRN	全局中断。
H2	SIOXFS0	I ² S左右通道标示或PCM帧同步信号。
H3	SIODO0	I ² S或PCM接口数据输出。
H4	SIOXCK0	I ² S或PCM接口位流时钟。
H5	VDDIO33	接口工作电源, 连接3.3V。
H6	DVSS	数字地 (DGND)。
H17	DVSS	数字地 (DGND)。
H18	VDDIO25	DDR工作电源, 连接2.5V。
H19	DDRDQS1	DDR SDRAM接口数据Strobe信号1。
H20	VREF1	DDR参考电源, 连接1.25V。
H21	DDRDQ11	DDR SDRAM接口数据总线。
H22	DDRDQ12	DDR SDRAM接口数据总线。
J1	SIODI0	I ² S或PCM接口数据输入。
J2	LCDP23	输出像素值, LCD_R7输出。 当GPIO7[1]=0, 作为LCDP23; 当GPIO7[1]=1, 作为GPIO6[0]。
J3	LCDP22	输出像素值, LCD_R6输出。 当GPIO7[1]=0, 作为LCDP22; 当GPIO7[1]=1, 作为GPIO5[7]。
J4	WDGRST	看门狗复位输出, 低有效, OD输出。
J5	VDDCORE	内核工作电源, 连接1.25V。
J6	DVSS	数字地 (DGND)。
J17	DVSS	数字地 (DGND)。



位置	管脚名称	功能描述
J18	VDDIO25	DDR工作电源，连接2.5V。
J19	DDRDQ8	DDR SDRAM接口数据总线。
J20	DDRDQ9	DDR SDRAM接口数据总线。
J21	VDDIO25	DDR工作电源，连接2.5V。
J22	DDRDQ10	DDR SDRAM接口数据总线。
K1	LCDP20	输出像素值，LCD_R4输出。 当GPIO7[1]=0，作为LCDP20； 当GPIO7[1]=1，作为GPIO5[5]。
K2	LCDP19	输出像素值，LCD_R3输出。 当GPIO7[2]=0，作为LCDP19； 当GPIO7[2]=1，作为SIOXFS2。
K3	LCDP18	输出像素值，LCD_R2输出。 当GPIO7[2]=0，作为LCDP18； 当GPIO7[2]=1，作为SIODO2。
K4	LCDP17	输出像素值，LCD_R1输出。 当GPIO7[2]=0，作为LCDP17； 当GPIO7[2]=1，作为SIOXCK2。
K5	LCDP21	输出像素值，LCD_R5输出。 当GPIO7[1]=0，作为LCDP21； 当GPIO7[1]=1，作为GPIO5[6]。
K6	DVSS	数字地（DGND）。
K10	DVSS	数字地（DGND）。
K11	DVSS	数字地（DGND）。
K12	DVSS	数字地（DGND）。
K13	DVSS	数字地（DGND）。
K17	DVSS	数字地（DGND）。
K18	VDDCORE	内核工作电源，连接1.25V。
K19	DDRDQ5	DDR SDRAM接口数据总线。
K20	DDRDQ6	DDR SDRAM接口数据总线。
K21	DDRDQ7	DDR SDRAM接口数据总线。
K22	DDRDM1	DDR SDRAM接口高字节Mask信号。



位置	管脚名称	功能描述
L1	LCDP16	输出象素值, LCD_R0输出。 当GPIO7[2]=0, 作为LCDP16; 当GPIO7[2]=1, 作为SIODI2。
L2	LCDP15	输出象素值, LCD_G7输出。 当GPIO7[2] =0, 作为LCDP15; 当GPIO7[2] =1, 作为SIORFS2。
L3	LCDP14	输出象素值, LCD_G6输出。
L4	LCDP13	输出象素值, LCD_G5输出。
L5	VDDIO33	接口工作电源, 连接3.3V。
L6	DVSS	数字地 (DGND)。
L10	DVSS	数字地 (DGND)。
L11	DVSS	数字地 (DGND)。
L12	DVSS	数字地 (DGND)。
L13	DVSS	数字地 (DGND)。
L17	DVSS	数字地 (DGND)。
L18	VDDIO25	DDR工作电源, 连接2.5V。
L19	VDDIO25	DDR工作电源, 连接2.5V。
L20	DDRDQS0	DDR SDRAM接口片选Strobe信号0。
L21	VDDIO25	DDR工作电源, 连接2.5V。
L22	DDRDQ4	DDR SDRAM接口数据总线。
M1	LCDP12	输出象素值, LCD_G4输出。
M2	LCDP9	输出象素值, LCD_G1输出。
M3	LCDP11	输出象素值, LCD_G3输出。
M4	VODAT6	输出象素值, TV或LCD_B6输出。
M5	VDDIO33	接口工作电源, 连接3.3V。
M6	DVSS	数字地 (DGND)。
M10	DVSS	数字地 (DGND)。
M11	DVSS	数字地 (DGND)。
M12	DVSS	数字地 (DGND)。
M13	DVSS	数字地 (DGND)。
M17	DVSS	数字地 (DGND)。
M18	VDDIO25	DDR工作电源, 连接2.5V。



位置	管脚名称	功能描述
M19	DDRCKN	DDR SDRAM接口反向时钟输出。
M20	DDRDQ1	DDR SDRAM接口数据总线。
M21	DDRDQ2	DDR SDRAM接口数据总线。
M22	DDRDQ3	DDR SDRAM接口数据总线。
N1	LCDP8	输出像素值，LCD_G0输出。
N2	VODAT2	输出像素值，TV或LCD_B2输出。
N3	VODAT7	输出像素值，TV或LCD_B7输出。
N4	VODAT5	输出像素值，TV或LCD_B5输出。
N5	LCDP10	输出像素值，LCD_G2输出。
N6	DVSS	数字地（DGND）。
N10	DVSS	数字地（DGND）。
N11	DVSS	数字地（DGND）。
N12	DVSS	数字地（DGND）。
N13	DVSS	数字地（DGND）。
N17	DVSS	数字地（DGND）。
N18	VDDIO25	DDR工作电源，连接2.5V。
N19	DDRCKP	DDR SDRAM接口正向时钟输出。
N20	DDRDM0	DDR SDRAM接口低字节Mask信号。
N21	VDDIO25	DDR工作电源，连接2.5V。
N22	DDRDQ0	DDR SDRAM接口数据总线。
P1	VODAT4	输出像素值，TV或LCD_B4输出。
P2	VODAT3	输出像素值，TV或LCD_B3输出。
P3	VOVS	帧同步脉冲，可以配置成输入输出。 在主模式下，配置成为输出。表示同步场信号脉冲，脉冲宽度和位置可配置。 在从模式下，配置成为输入，适合BT.601标准。
P4	VODAT1	输出像素值，TV或LCD_B1输出。
P5	VDDIO33	接口工作电源，连接3.3V。
P6	DVSS	数字地（DGND）。
P17	DVSS	数字地（DGND）。
P18	VDDCORE	内核工作电源，连接1.25V。
P19	VDDIO25	DDR工作电源，连接2.5V。



位置	管脚名称	功能描述
P20	VREF2	DDR参考电源, 连接1.25V。
P21	DDRADR11	DDR SDRAM接口行、列地址信号。
P22	DDRADR12	DDR SDRAM接口行、列地址信号。
R1	VODAT0	输出象素值, TV或LCD_B0输出。
R2	LCDCB	数据有效信号或者场号信号。 当工作在主模式时, 为输出; 当工作在从模式时, 为输入 (在从模式下, 只能表示场号信号)。 在LCDCB=0时, 表示场号信号; 在LCDCB=1时, 表示数据有效信号。
R3	VIDAT6	视频数据输入或原始数据输入。
R4	VOCK	象素时钟。视频数据为时钟的上升沿同步触发输出。
R5	VDDCORE	内核工作电源, 连接1.25V。
R6	DVSS	数字地 (DGND)。
R17	DVSS	数字地 (DGND)。
R18	VDDIO25	DDR工作电源, 连接2.5V。
R19	DDRADR7	DDR SDRAM接口行、列地址信号。
R20	DDRADR8	DDR SDRAM接口行、列地址信号。
R21	DDRADR9	DDR SDRAM接口行、列地址信号。
R22	DDRADR10	DDR SDRAM接口行、列地址信号。
T1	VOHS	行同步脉冲, 可以配置成输入输出。 在主模式下, 配置成为输出。表示同步行脉冲信号, 脉冲宽度和脉冲起始位置可配置。 在从模式下, 配置成为输入, 适合BT.601标准。
T2	VIDAT7	视频数据输入或原始数据输入。
T3	VIDAT5	视频数据输入或原始数据输入。
T4	VIDAT4	视频数据输入或原始数据输入。
T5	NC	悬空, 无需连接。
T6	DVSS	数字地 (DGND)。
T17	DVSS	数字地 (DGND)。
T18	VDDIO25	DDR工作电源, 连接2.5V。
T19	DDRADR4	DDR SDRAM接口行、列地址信号。
T20	DDRADR5	DDR SDRAM接口行、列地址信号。
T21	VDDIO25	DDR工作电源, 连接2.5V。



位置	管脚名称	功能描述
T22	DDRADR6	DDR SDRAM接口行、列地址信号。
U1	VIDAT3	视频数据输入或原始数据输入。
U2	VIDAT2	视频数据输入或原始数据输入。
U3	VIDAT1	视频数据输入或原始数据输入。
U4	VICK	像素位流时钟。
U5	VDDIO33	接口工作电源，连接3.3V。
U6	DVSS	数字地（DGND）。
U7	DVSS	数字地（DGND）。
U8	DVSS	数字地（DGND）。
U9	DVSS	数字地（DGND）。
U10	DVSS	数字地（DGND）。
U11	DVSS	数字地（DGND）。
U12	DVSS	数字地（DGND）。
U13	DVSS	数字地（DGND）。
U14	DVSS	数字地（DGND）。
U15	DVSS	数字地（DGND）。
U16	DVSS	数字地（DGND）。
U17	DVSS	数字地（DGND）。
U18	VDDIO25	DDR工作电源，连接2.5V。
U19	DDRADR0	DDR SDRAM接口行、列地址信号。
U20	DDRADR1	DDR SDRAM接口行、列地址信号。
U21	DDRADR2	DDR SDRAM接口行、列地址信号。
U22	DDRADR3	DDR SDRAM接口行、列地址信号。
V1	VIDAT0	视频数据输入或原始数据输入。
V2	VIVS	垂直同步或场信号。 当工作在主模式时，作为输出； 工作在从模式时，作为输入。 当vsync=0时，工作在场号模式下； 当vsync=1时，工作在垂直脉冲模式。 不使用时，必须连接GND。
V3	SDRCK2	SDRAM接口时钟输出2。



位置	管脚名称	功能描述
V4	VIHS	水平同步或数据有效。 当工作在主模式时，作为输出； 工作在从模式时，作为输入。 当hsync=0时，工作在数据有效模式下； 当hsync=1时，工作在水平同步脉冲模式。 不使用时，必须连接GND。
V5	BOOTSEL1	静态RAM接口片选1上电复位时对应的memory数据位宽设置。
V6	BOOTSEL0	静态RAM接口片选0上电复位时对应的memory数据位宽设置。
V7	VDDIO33	接口工作电源，连接3.3V。
V8	VDDIO33	接口工作电源，连接3.3V。
V9	VDDIO33	接口工作电源，连接3.3V。
V10	VDDCORE	内核工作电源，连接1.25V。
V11	VDDIO33	接口工作电源，连接3.3V。
V12	VDDIO33	接口工作电源，连接3.3V。
V13	NC	悬空，无需连接。
V14	VDDIO33	接口工作电源，连接3.3V。
V15	VDDCORE	内核工作电源，连接1.25V。
V16	VDDIO33	接口工作电源，连接3.3V。
V17	VSS1PLL	数字地（DGND）。
V18	AVSSIOPLL	模拟地（AGND）。
V19	AVDDIOPLL	PLL电源，连接模拟3.3V。
V20	VDDIO25	DDR工作电源，连接2.5V。
V21	DDRBA0	DDR SDRAM接口bank 0选择信号。
V22	DDRBA1	DDR SDRAM接口bank 1选择信号。
W1	SDRCASN	SDRAM接口列选通信号。
W2	SDRRASN	SDRAM接口行选通信号。
W3	DVSS	数字地（DGND）。
W4	EBICS3N	静态RAM片选3。
W5	EBIBLS0	静态RAM接口数据字节选择0。
W6	EBIDQ2	SDRAM和静态RAM数据总线。
W7	EBIDQ6	SDRAM和静态RAM数据总线。



位置	管脚名称	功能描述
W8	EBIDQ10	SDRAM和静态RAM数据总线。
W9	EBIDQ14	SDRAM和静态RAM数据总线。
W10	EBIDQ19	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ19; 当GPIO7[0]=1, 作为GPIO4[0]。
W11	EBIBLS2	静态RAM接口数据字节选择2。
W12	EBIDQ27	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ27; 当GPIO7[0]=1, 作为GPIO5[0]。
W13	EBIDQ30	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ30; 当GPIO7[0]=1, 作为GPIO5[3]。
W14	EBIDQ31	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ31; 当GPIO7[0]=1, 作为GPIO5[4]。
W15	EBIADR5	静态RAM地址总线、SDRAM行、列地址信号。
W16	EBIADR9	静态RAM地址总线、SDRAM行、列地址信号。
W17	EBIADR13	静态RAM地址总线、SDRAM bank 0选择信号。
W18	EBIADR17	静态RAM地址总线。
W19	VDD1PLL	PLL电源, 连接数字 1.25V。
W20	AVDD33PLL	PLL电源, 连接模拟3.3V。
W21	DVSS	数字地 (DGND)。
W22	NC	悬空, 无需连接。
Y1	EBICS2N	静态RAM片选2。
Y2	SDRCKFB	SDRAM反馈时钟输入。
Y3	SDRCK1	SDRAM接口时钟输出1。
Y4	SDRCSN	SDRAM片选信号。
Y5	SDRDM0	SDRAM接口字节使能。
Y6	EBIDQ3	SDRAM和静态RAM数据总线。
Y7	EBIDQ7	SDRAM和静态RAM数据总线。
Y8	EBIDQ11	SDRAM和静态RAM数据总线。
Y9	EBIDQ15	SDRAM和静态RAM数据总线。



位置	管脚名称	功能描述
Y10	EBIDQ18	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ18; 当GPIO7[0]=1, 作为GPIO3[7]。
Y11	EBIDQ23	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ23; 当GPIO7[0]=1, 作为GPIO4[4]。
Y12	EBIDQ26	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ26; 当GPIO7[0]=1, 作为GPIO4[7]。
Y13	EBIADR1	静态RAM地址总线、SDRAM行、列地址信号。
Y14	EBIADR0	静态RAM地址总线、SDRAM行、列地址信号。
Y15	EBIADR4	静态RAM地址总线、SDRAM行、列地址信号。
Y16	EBIADR8	静态RAM地址总线、SDRAM行、列地址信号。
Y17	EBIADR12	静态RAM地址总线、SDRAM行、列地址信号。
Y18	EBIADR16	静态RAM地址总线。
Y19	AVSS33PLL	模拟地 (AGND)。
Y20	XOUT2	27MHz晶振时钟输出。
Y21	XIN2	27MHz晶振时钟输入。
Y22	EBIADR25	静态RAM地址总线。
AA1	EBICS0N	静态RAM片选0。
AA2	SDRCKE	SDRAM时钟使能输出。
AA3	EBICS1N	静态RAM片选1, 一般用于连接boot FLASH器件。
AA4	EBIWEN	SDRAM和静态RAM写使能。
AA5	EBIDQ0	SDRAM和静态RAM数据总线。
AA6	EBIDQ4	SDRAM和静态RAM数据总线。
AA7	EBIDQ8	SDRAM和静态RAM数据总线。
AA8	EBIDQ12	SDRAM和静态RAM数据总线。
AA9	EBIBLS1	静态RAM接口数据字节选择1。
AA10	EBIDQ17	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ17; 当GPIO7[0]=1, 作为GPIO3[6]。



位置	管脚名称	功能描述
AA11	EBIDQ22	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ22; 当GPIO7[0]=1, 作为GPIO4[3]。
AA12	EBIDQ25	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ25; 当GPIO7[0]=1, 作为GPIO4[6]。
AA13	EBIDQ29	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ29; 当GPIO7[0]=1, 作为GPIO5[2]。
AA14	EBIBLS3	静态RAM接口数据字节选择3。
AA15	EBIADR3	静态RAM地址总线、SDRAM行、列地址信号。
AA16	EBIADR7	静态RAM地址总线、SDRAM行、列地址信号。
AA17	EBIADR11	静态RAM地址总线、SDRAM行、列地址信号。
AA18	EBIADR15	静态RAM地址总线。
AA19	EBIADR19	静态RAM地址总线。
AA20	EBIADR21	静态RAM地址总线。
AA21	EBIADR23	静态RAM地址总线。
AA22	EBIADR24	静态RAM地址总线。
AB1	DREQ	外部DMA请求。
AB2	EBIOEN	静态RAM接口数据输出使能。
AB3	DACK	外部DMA响应。
AB4	EBIDQ1	SDRAM和静态RAM数据总线。
AB5	EBIDQ5	SDRAM和静态RAM数据总线。
AB6	EBIDQ9	SDRAM和静态RAM数据总线。
AB7	EBIDQ13	SDRAM和静态RAM数据总线。
AB8	SDRDM1	SDRAM接口字节使能。
AB9	EBIDQ16	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ16; 当GPIO7[0]=1, 作为GPIO3[5]。
AB10	EBIDQ20	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ20; 当GPIO7[0]=1, 作为GPIO4[1]。
AB11	EBIDQ21	SDRAM和静态RAM数据总线。



位置	管脚名称	功能描述
AB12	SDRDM2	SDRAM接口字节使能。
AB13	EBIDQ24	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ24; 当GPIO7[0]=1, 作为GPIO4[5]。
AB14	EBIDQ28	SDRAM和静态RAM数据总线。 当GPIO7[0]=0, 作为EBIDQ28; 当GPIO7[0]=1, 作为GPIO5[1]。
AB15	SDRDM3	SDRAM接口字节使能。
AB16	EBIADR2	静态RAM地址总线、SDRAM行、列地址信号。
AB17	EBIADR6	静态RAM地址总线、SDRAM行、列地址信号。
AB18	EBIADR10	静态RAM地址总线、SDRAM行、列地址信号。
AB19	EBIADR14	静态RAM地址总线、SDRAM bank 1选择信号。
AB20	EBIADR18	静态RAM地址总线。
AB21	EBIADR20	静态RAM地址总线。
AB22	EBIADR22	静态RAM地址总线。



B 术语

A

- Alpha Blending** 将多个像素层按照 Alpha 值进行加权求和，得到一个 Alpha 混和后的输出位图。
- AMBA2.0 总线架构** 由英国 ARM 公司开发、开放的高性能片上总线架构，包括连接高速外设的 AHB 总线架构和连接低速外设的 APB 总线架构。
- ARM926EJ-S BIST 测试模式** 对 Hi3510 ARM926EJ-S 内部存储器进行测试的模式。

B

- baseline profile** H.264 协议分为 baseline, main, extended 三个 profile，即三个档次。profile 指定了用于编解码的技术点。
- blit** 位块传送，将某一内存块的数据传送到另一内存块。
- Boundary SCAN 测试模式** 对 Hi3510 进行边界扫描测试的模式。
- 半像素插值** 采用插值算法计算图像中相邻采样像素点中间位置的采样像素点的值

C

- capture_area** 视频捕获区域。
- Cb、Cr、Y** 彩色图像由 Y/U/V 三个分量来表示。其中 Y 分量为亮度分量，U/V 分量为色差分量。Cb 为蓝色色差，Cr 为红色色差分量。Y/U/V、Y/Cb/Cr 是业界约定的表示方法。
- clip** 剪裁，将超出输出图像大小范围的源位图、目标位图或者 Pattern 裁掉的一种操作。
- Colorspace** 关键色。在 Colorspace_min 和 Colorspace_max 之间的颜色为关键色，在 BLT 操作中，将视具有这种颜色的区域为透明。

**D**

- DC 系数** DCT 系数后的直流分量。
- de-blocking** 去块效应，视频编码器中用于消除块效应的环路滤波功能。de-blocking 功能为 H.263 协议的 J 选项。
- DMV** MV 为运动矢量。
- Down Scaling** 亚采样。

E

- ETM9+** External Trace Module Plus。外部实时跟踪模块，一般与 ARM9 系列处理器连接。
- ETM9+ SCAN 测试模式** 对 Hi3510 中的 ETM9+ 进行扫描测试的模式。

F

- Fixed-priority** 固定优先级
- Free-running** 自由运行（模式）。

I

- I 宏块** 按照帧内预测编码的宏块。
- I 帧** 按照帧内预测编码的图象，图象中的所有宏块全部为 I 宏块。

M

- MultiICE V2.2** ARM 公司提供的 JTAG 调试工具，与 Hi3510 的 JTAG 接口连接使用。
- MultiTrace V1.0** ARM 公司提供的实时跟踪调试工具，与 Hi3510 的 ETM+ 连接使用。

O

- one-shot** 一次触发（模式）。

P

- Pattern** 是一个固定大小的位图（例如 32×32 像素），用来平铺填充设备环境的某一区域。
- periodic** 周期触发（模式）。
- PLL 测试模式** 对锁相环进行测试的模式。



P-skip	P-skip 宏块为 P 帧中的“透明块”，即与上一帧相同的宏块。在帧间预测编码过程中 P-skip 的纹理数据不用编码。P-skip 为 H.264 协议中定义的语法。
P 宏块	按照帧间预测编码的宏块。
P 帧	按照帧间预测编码的帧，图象中既包含 P 宏块，也可能包含 I 宏块。
Q	
桥读	APB 桥读取。
桥写	APB 桥写入。
R	
RealView ICE V1.0.1	ARM 公司提供的 JTAG 调试工具，与 Hi3510 的 JTAG 接口连接使用。
RealViewTrace	ARM 公司提供的实时跟踪调试工具，与 Hi3510 的 ETM+连接使用。
Remap	地址重映射。
Round-robin	轮转优先级
Run、Level	VLC 编码为一种游程编码。run 表示当前的非零系数前面的 0 的个数，level 为非零系数编码后的码字。
S	
Slice group	一副图象划分为若干 slice，这些 slice 又可组成为 1 到多个组，即 slice group。可翻译为 slice 组。slice 是 H.264 中定义的一种语法成分。
stride 属性	行间距，描述了本行起始地址与下一行起始地址之间的距离。
水线值	FIFO 的水线值。
水平（垂直）消隐	水平（垂直）消隐区域。
Z	
Zig-zag 扫描	之字型扫描，这个也是所有图象编码都通用的一种扫描（从二维的图象块中抽取出字节串的过程）方式。



C 缩略语

A

AEC	Acoustic Echo Cancellation	声学回波抵消
AGC	Automatic Gain Control	自动增益控制
ALU	Arithmetic Logic Unit	算术逻辑单元

D

DEU	Device Emulation Unit	设备模拟单元
DMA	Direct Memory Access	直接存储器访问
DMAC	DMA Controller	DMA 控制器
DSP	Digital Signal Processor	数字信号处理器
DSU	Dedicated Scaling Unit	专用缩放单元
DTCM	Data Tight Coupled Memory	数据紧耦合存储器

E

ESD	Electro-Static Discharge	静电放电
ETM	External Trace Module	外部跟踪模块
EXP ARBITER	Expansion Arbiter	扩展总线仲裁器

F

FIQ	Fast Interrupt reQuest	快速中断请求
FLASH	Flash memory	闪存存储器/闪存

**I**

I2C	The Inter-Integrated Circuit	一种串行总线协议标准
I2S	Inter-IC Sound	I ² S 音频输入输出接口
ICD	Interface Clock Division	接口时钟分频系数
IrDA	Infrared Data Association	红外线数据连接
ITCM	Instruction Tight Coupled Memory	指令紧耦合存储器

L

LAN	Local Area Network	局域网
LCD	Liquid Crystal Display	液晶显示屏
LSU	Load Store Unit	数据访问单元

M

MAC	Media Access Control	媒体接入控制
MAU	Multiply Accumulate Unit	乘积单元
MDIO	Management Data Input / Output	管理数据输入输出接口
MEMC	MEMory Controller	存储控制器
MII	Media Independent Interface	介质无关接口
MMU	Memory Management Unit	存储控制单元
MQ	Modified Quantization mode	指色度系数可采用于亮度系数不同的量化步长。MQ 功能为 H.263 协议的 T 选项
MSL	Moisture Sensitivity Level	潮湿等级
MSS	Memory Sub System	Memory 子系统

N

NAT	Net Address Translation	网络地址转换
NTSC	National Television System Committee	国家电视制式委员会



O

OFB Output Feedback 输出反馈

P

PAL Phase-Alternation Line 逐行倒相制

PCB Printed Circuit Board 印刷电路板

PCM Pulse Code Modulation 脉冲编码调制

PFU Pre-fetch Unit 取指单元

PLL Phase-Locked Loop 锁相环

PPP Point to Point Protocol 点对点协议

PPPOE PPP over Ethernet 在以太网上传输的点对点协议

R

RH Relative Humidity 相对湿度

RISC Reduced Instruction Set Computer 精简指令集计算机

RMII Reduced MII 简化的 MII 接口

ROP Raster Operation 指对 2 种位图的颜色值进行布尔运算

RTC Real-time Clock 实时时钟

S

SDRAM Synchronous Dynamic Random Access Memory 同步动态随机存储器

SF Switch Fabric 以太网交换

SIO Serial Input / Output 串行输入输出接口

SIP Session Initiation Protocol 会话发起协议

SSP Synchronous Serial Protocol 同步串口

**T**

TBD	To Be Determined	待定
TCM	Tight Coupled Memory	紧耦合存储器
TDE	Two Dimensional Engine	2D 加速引擎
TV	Television	电视机

U

UART	Universal Asynchronous Receiver Transmitter	通用异步收发器
-------------	---	---------

V

VAD	Voice Activity Detection	语音激活检测
VGA	Video Graphics Array	视频图象阵列
VIC	Vector Interrupt Controller	向量中断控制器
VIU	Video Input Unit	视频输入单元
VLAN	Virtual LAN	虚拟局域网
VoIP	Voice over IP	基于 IP 的语音传输
VOU	Video Output Unit	视频输出单元